INTEGRATING SRX SERIES SERVICES GATEWAYS INTO A QFABRIC™ SWITCH-BASED DATA CENTER IMPLEMENTATION GUIDE
Table of Contents

Introduction .................................................................................................... 4
Scope ............................................................................................................. 4
Target Audience .......................................................................................... 4
SRX Technical Concepts .............................................................................. 4
  Security Zones and Interfaces ..................................................................... 5
  Security Policies ........................................................................................ 5
  HA Chassis Clusters .................................................................................. 5
  Redundancy Groups ................................................................................... 5
  Redundant Ethernet Interfaces ................................................................. 6
  Redundancy Group Failover ....................................................................... 6
QFabric System Hardware Architecture Overview ........................................ 6
QFabric and SRX Integration Design Overview ............................................ 7
Use Case 1: QFabric in L2 Mode, SRX As the FHR ...................................... 8
Use Case 2: QFabric in L2 or L3, VR-based Traffic Steering to the SRX ........ 9
Use Case 3: SRX As a Low-Latitude Firewall .............................................. 10
Physical and Logical Topologies .................................................................. 11
  QFabric in L2 Mode, SRX As FHR .......................................................... 12
  QFabric in L3 Mode, VR-based Traffic Steering ......................................... 13
Implementation ............................................................................................ 14
  QFabric Configuration .............................................................................. 14
  Interface Naming Conventions for QFabric Technology ............................ 15
  Interface Type Configuration ..................................................................... 15
  SRX Series Configuration ........................................................................ 16
  Configuring Inter-VR Routing on the SRX Series ....................................... 18
  Connecting the SRX Cluster with the QFabric Solution ........................... 20
  IP Routing Between the SRX Series and QFabric Solution ....................... 20
Enabling Services Offload (Low-Latitude Firewall) .................................... 23
AppSecure ................................................................................................ 25
  Configuring AppSecure Intrusion Policies ............................................. 25
Validation Testing and Convergence Testing .............................................. 27
Summary ................................................................................................. 27
Appendix A: Configuration Files ............................................................... 28
  Use Case 1: QFabric in Layer 2, SRX As FHR—Active/Passive and Active/Active Cluster Modes ...................... 28
  QFabric Layer 2 Configuration (QF-2PS.conf) ........................................ 28
  SRX Delta Configuration to Convert an Active/Passive Cluster to Active/Active ............................................. 52
  Use Case 2: QFabric in Layer 3 Mode, VR-based Traffic Steering to SRX As Needed .......................................... 52
    QFabric Configuration for Layer 3 Mode with Static Routing (QF-3PS-static.conf) ........................................... 52
    SRX Configuration—Active/Passive Cluster (SRX-3PS-static) .................. 68
  Use Case 3: SRX with Services Offload ................................................... 82
    SRX Delta Configuration for Enabling Services Offload ........................ 82
Appendix B: QFabric Interfaces for Servers .............................................. 83
  QFabric Interfaces for EX Switches ........................................................ 83
About Juniper Networks ............................................................................ 84
List of Figures
Figure 1: QFabric in L2 mode, SRX Series as the FHR ............................................................... 8
Figure 2: QFabric in L2 or L3 mode, SRX Series used as firewall .................................................... 9
Figure 3: Example of data center architecture ..................................................................... 11
Figure 4: QFabric as L2 switch and SRX as FHR ................................................................. 12
Figure 5: QFabric in L3 mode, SRX for security services ............................................................. 13
Figure 6: Network security zones ................................................................................. 18
Figure 7: QFabric and SRX links, reth interface and LAG ........................................................... 20

List of Tables
Table 1: Juniper Networks Products Used .......................................................................... 11
Table 2: Servers ................................................................................................. 12
Table 3: Test Equipment: Traffic Generators ......................................................................... 12
Table 4: IP and Network Address Scheme ........................................................................... 14
Introduction

The data center is one of the most intensive deployment locations for networking equipment, consisting of thousands of servers that are accessed by tens of thousands of client systems. The need for large-scale access creates a complex set of data flows and makes it challenging to define the specific profile of network traffic. Moreover, because determining firewall deployment and sizing in a data center is a massive effort, firewalls are often deployed in a limited fashion, if deployed at all. Another driving factor in the limited use of firewalls is the lack of performance. Raw bandwidth limitations, connections per second, and sustained connections have caused architects to distribute security or dispense with it altogether.

To address these challenges, Juniper Networks has created a new class of security products. Juniper Networks SRX Series Services Gateways provide the ability to scale in ways that were not thought possible in the past. Juniper has leveraged key technologies, such as the Juniper Networks Junos® operating system and proven hardware architectures, to create the SRX Series of products. The SRX Series meets the demanding needs of data centers today, with the ability to expand and address customers’ needs of tomorrow. With the high-end SRX Series gateway products, firewall deployment in the data center becomes natural and reasonable. This document describes various SRX deployment options in a Juniper Networks QFabric technology-based data center.

Scope

This document provides various design considerations and implementation guidelines to deploy firewall services in the QFabric switch-based data center using high-end SRX Series Services Gateways. The firewall services in the data center core can provide additional security and help meet compliance requirements by segmenting deployed server networks and securing traffic within server networks.

This document briefly reviews the technical concepts of the SRX Series Services Gateways related to design and implementation of firewall services. Deployment scenarios are based on a single logical switch design.

The deployment scenarios and design considerations include:

- Active/passive firewall cluster deployment with SRX acting as the first-hop router (FHR) for QFabric
- Active/active firewall cluster deployment with SRX acting as the FHR for QFabric
- Active/passive firewall cluster deployment with QFabric acting as the FHR for intravirtual router traffic
- Active/active firewall cluster deployment with QFabric acting as the FHR for intravirtual router traffic
- Active/passive low-latency firewall deployment for latency-sensitive applications

In addition, the implementation steps and validated configuration details for the deployment scenarios are presented.

The designs were tested for transit traffic latency over various paths that network traffic might take during normal operations in the event of a failure. The designs were validated for convergence for various failure scenarios, such as link failure, network infrastructure device failure, and firewall node failure, to ensure required resiliency. The SRX Series Services Gateways support all major dynamic routing protocols.

For network integration, we used OSPF as the dynamic routing protocol. For performance and scalability details concerning the SRX Series Services Gateways and QFabric, contact your local Juniper account representative.

Note: The Juniper Networks MX Series 3D Universal Edge Routers are represented in the diagrams for completeness, however, the MX Series configuration is out of the scope of this document.

Target Audience

This document is intended for architects, network engineers and operators, and those who require technical knowledge regarding integrating the SRX Series with QFabric technology. It is assumed that the reader is familiar with the Junos operating system, knowledgeable about the QFabric family of products, and has basic familiarity with firewall products.

SRX Technical Concepts

The following technical concepts are related to the SRX Series and relevant to firewall deployment:

- Security zones and interfaces
- Security policies
- High availability (HA) chassis clusters
- Redundancy groups
- Redundant Ethernet interfaces
- Redundancy group failover
Security Zones and Interfaces
Interfaces act as a doorway through which traffic enters and exits an SRX Series gateway appliance. Many interfaces can share the same security requirements; however, different interfaces can also have different security requirements for inbound and outbound data packets. You can group interfaces with identical security requirements together into a single security zone.

Security zones are logical entities to which one or more interfaces are bound. On a single device, you can configure multiple security zones, dividing the network into segments to which you can apply various security options to satisfy the needs of each segment. Security zones also allow administrators to group network addresses in an abstract construct of security zones and define security policies for inter-zone traffic. This approach reduces or eliminates the need for maintaining address lists for security policy. To achieve this, you must define at least two security zones to protect one area of the network from the other. If all interfaces on the firewall share a single security zone, the security policy is defined as an intra-zone policy, and administrators must maintain an address list to specify the source and destination for the security policies. This can be helpful for migrating existing security policies from another firewall platform.

Security Policies
Security zones are the building blocks for security policies. A security zone is a logical entity to which one or more interfaces are bound. A security zone provides a means of distinguishing groups of hosts (user systems and other hosts, such as servers) and their resources from one another to apply different security measures to them. Active security policies enforce rules for the transit traffic in terms of which traffic can pass through the firewall, and the actions that must occur on the traffic as it passes through the firewall. By default, a device denies all traffic in all directions. Through the creation of policies, you can control the traffic flow from zone to zone by defining the kinds of traffic permitted to pass from specified sources to specified destinations at scheduled times. At the broadest level, you can allow all kinds of traffic from any source in one zone to any destination in all other zones without any scheduling restrictions. At the narrowest level, you can allow only one kind of traffic between a specified host in one zone and another specified host in another zone during a scheduled time interval.

HA Chassis Clusters
To form a chassis cluster, a pair of identical SRX Series devices are combined to act as a single system that enforces the same overall security. For Juniper Networks SRX5600 Services Gateway and Juniper Networks SRX5800 Services Gateway chassis clusters, the placement and type of Services Processing Cards (SPCs) must match in the two clusters. 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.

Redundancy Groups
Chassis clustering provides HA of interfaces and services through redundancy groups and primacy within groups. A redundancy group is an abstract construct that includes and manages a collection of objects. It contains objects on both nodes, acting as the primary on one node and backup on the other at all times. When a redundancy group is primary, its objects on that node are active. A redundancy group can be primary on only one node at a time.

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

Three characteristics determine the primacy of a redundancy group:
- Priority configured for the node
- Node ID (in case of tied priorities)
- Order in which the node comes up

Note: If a lower priority node comes up first, it assumes the primacy for a redundancy group and remains as primary if preempt is not enabled.

A chassis cluster can include many redundancy groups, some of which might be primary on one node and some primary on another node. Alternatively, all redundancy groups can be primary on a single node. One redundancy group’s primacy does not affect another redundancy group’s primacy.

When you initialize a device in chassis cluster mode, the system creates a redundancy group referred to in this document as redundancy group 0. Redundancy group 0 manages the primary and failover between the routing engines on each node of the cluster. The node on which redundancy group 0 is primary determines which routing engine is active in the cluster. A node is considered the primary node of the cluster if its routing engine is the active one.
You can configure up to 128 redundancy groups, numbered 1 through 128. Each redundancy group contains one or more redundant Ethernet interfaces. A redundant Ethernet interface is a pseudointerface that contains a pair of physical Gigabit Ethernet or Fast Ethernet interfaces. If a redundancy group is active on node 0, the child links of all associated redundant Ethernet interfaces on node 0 are active. If the redundancy group fails over to node 1, the child links of all redundant Ethernet interfaces on node 1 become active. You can configure multiple redundancy groups to load-share traffic across the cluster.

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

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

**Redundant Ethernet Interfaces**

A redundant Ethernet interface is a pseudointerface that includes a physical interface from each node of the cluster. The interface can contain either a pair of Fast Ethernet or Gigabit Ethernet interfaces that are referred to as child interfaces of the redundant Ethernet interface (the redundant parent). Each redundant Ethernet interface can contain only two interfaces, because a cluster contains only two nodes. A redundant Ethernet interface's child interface is associated with the redundant Ethernet interface as part of the child interface configuration. The redundant Ethernet interface's child interface inherits most of its configuration from its parent. A redundant Ethernet interface inherits its failover property from the redundancy group to which it belongs. A redundant Ethernet interface remains active as long as its primary child interface is available and active.

**Redundancy Group Failover**

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

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

If the monitored interfaces of a redundancy group on both nodes reach their thresholds at the same time, the redundancy group is primary on the node with the lower node ID, in this case node 0.

**QFabric System Hardware Architecture Overview**

The QFabric system is a single-layer networking tier that connects servers and storage devices to one another across a high-speed, unified core fabric. You can view the QFabric system as a single, extremely large, nonblocking, high-performance Layer 2 and Layer 3 switching system. The Director software running on the Director group allows the main QFabric system administrator to access and configure every device and port in the QFabric system from a single location. Although you configure the system as a single entity, the fabric contains four major hardware components. The hardware components can be chassis-based, group-based, or a hybrid of the two. It is important to understand the components and their functions.

**Director group**—A management platform that establishes, monitors, and maintains all components in the QFabric system. A set of Director devices run the Junos OS on top of a CentOS foundation. The Director group handles tasks such as QFabric system network topology discovery, node and interconnect device configuration and startup, and Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), and Network File System (NFS) services. The Director group also runs the software for management applications, hosts and load-balances internal processes for the QFabric system, and starts additional QFabric system processes as requested.
**Node device**—A hardware system located on the ingress of the QFabric system that connects to endpoints (such as servers or storage devices) or external networks, and is connected to the heart of the QFabric system through an interconnect device. You can use a node device similar to how a top-of-rack switch is implemented. By default, node devices connect to servers or storage devices. However, when you group node devices together to connect to a network that is external to the QFabric system, the formation is called a network node group.

**Interconnect device**—The primary fabric for data plane traffic traversing the QFabric system between node devices. To reduce latency to a minimum, the interconnect device implements multistage Clos switching (Clos refers to a specific type of circuit switching network, first formalized by Charles Clos in 1953) to provide nonblocking interconnections between any node device in the system.

**Control plane network**—An out-of-band Gigabit Ethernet management network that connects all QFabric system components. For example, you can use a group of EX4200 Ethernet switches configured as a Virtual Chassis to enable the control plane network. The control plane network connects the Director group to the management ports of the node and interconnect devices. By keeping the control plane network separate from the data plane, the QFabric system can scale to support thousands of servers and storage devices.

For more information about the QFabric hardware and software architecture, see:  

**QFabric and SRX Integration Design Overview**

The following use cases show scenarios for a QFabric and SRX integration:

- QFabric technology in Layer 2 (L2) mode, SRX as the FHR
- QFabric technology in L2 or Layer 3 (L3) mode, virtual router-based traffic steering
- SRX Series services offload feature providing a low-latency firewall

The SRX Series, which is deployed as a one-arm solution in the three use cases, is sometimes referred to as a “firewall on a stick” configuration. Administrators have the flexibility to selectively define which traffic gets protection through the firewall or bypasses the firewall for certain types of traffic using routing policy configuration. L3 termination can be at the firewall, and inter-VLAN traffic is routed by the firewall. This configuration ensures that traffic to and from VLANs is always protected using firewall services. For applications that do not require firewall protection, or in cases where sending traffic through firewalls breaks the application (legacy application), VLANs hosting these applications can be terminated on the QFabric solution or on the MX Series, and communication among the VLANs can bypass the firewalls.

With a one-arm firewall deployment, administrators can virtually keep the firewall inline by terminating the VLAN at the firewall or configuring the routing on the QFabric solution to always send inter-VLAN routing traffic through the firewall. Administrators can also selectively bypass traffic using routing policy control. Because this configuration provides flexible deployment options, we will explore the design options with one-arm firewall deployment solutions.
Use Case 1: QFabric in L2 Mode, SRX As the FHR

In this deployment scenario, the first-hop router (FHR) is either the SRX Series or the MX Series, depending on which traffic needs firewall policies (or any other L3 equipment). The QFabric solution is strictly an L2 connection, as shown in Figure 1.

In this scenario, only finance servers require the SRX Series security services, so the routing for finance resides on the SRX Series, and the routing for engineering and marketing resides on the MX Series. The SRX Series serves as the FHR for the finance servers, and the MX Series is the FHR for the engineering and marketing servers. All traffic from and to finance servers must go through the SRX Series for security policy enforcement.

The advantages of this deployment are:

- Large table scale, such as host tables and media access control (MAC) address table. Using a QFabric solution in L2 mode allows you to use an external routing layer with higher table scales, such as the MX Series.
- Simple configuration model. Each appliance has a dedicated role. The MX Series and the SRX Series handle the routing, the SRX Series handles security, and the QFabric solution handles L2 processing. Only a VLAN configuration is required.
- Appropriate design when VLANs must span different geographic data centers.

This deployment also has some drawbacks. Traffic that must be routed (inside the QFabric solution) needs to go across the MX Series or SRX Series before returning to the QFabric solution, thus increasing latency and bandwidth consumption.
Use Case 2: QFabric in L2 or L3, VR-based Traffic Steering to the SRX

In this scenario, the MX Series is the gateway to the outside world, and the SRX Series is used for services. The QFabric solution is the default gateway for the servers.

The advantages of this deployment are:

- Low and consistent latency for routing because routing happens closer to the endpoints (servers) inside the QFabric solution if no SRX Series services are needed.
- Full separation of the data center from the outside world, with the MX Series as the gateway between them, and dedicated to Internet-bound traffic only and leveraging advanced features (MPLS/VPN).
- Only intervirtual router (VR) traffic that needs security services goes to the SRX Series for security policy enforcement.
- Only traffic that needs to reach the outside world exits the QFabric system.

The drawbacks to this deployment are:

- Splitting the routing functionality between three devices makes the configuration more complex than Use Case 1.
- VRs must be used on the QFabric side for traffic isolation, with a correct mapping of the traffic to the appropriate security zone on the SRX Series or to the appropriate integrated routing and bridging (IRB) on the MX Series.
- Limited table scale, such as host tables.

If a customer wants to create a security zone on a per VR basis and apply those security policies to inter-VR traffic, the QFabric solution must act as the FHR, and the SRX Series is used for firewall and other security services only. To avoid capacity or scaling problems, administrators should take into account that the QFabric solution must route a significantly higher volume of traffic that does not require services.
**Note:** In traditional data center architectures, Virtual Router Redundancy Protocol (VRRP) is required to secure gateway redundancy for L3 devices. However, with the QFabric architecture, VRRP is not necessary because a QFabric solution is a single logical switch, therefore multiple devices running as gateways are not required. Within a network node group, high availability of a gateway is built in already. For example, in Figure 6, the SRX Series cluster connects to two QFabric nodes, which are part of a network node group. To the SRX Series cluster, it is the same as connecting to different ports on different line cards on a single switch. These line cards and ports are fully synchronized at the QFabric Director level. Because protocols are not needed to ensure switchover between devices, users do not have to configure VRRP among network node groups.

**Use Case 3: SRX As a Low-Latency Firewall**

Implementing firewalls within the data center has always been a challenge because firewalls usually introduce considerable latency because they have to analyze the packets for security processing. Often times, latency-sensitive applications are bypassed for this reason. Juniper Networks offers a low-latency firewall solution that works in conjunction with QFabric technology, which has interface-to-interface latency of approximately 2 microseconds (ms) at small scale, growing slowly to about 10 ms at the largest scale.

This use case deploys the SRX Series as a low-latency firewall with QFabric technology in customer environments in which business applications require segmentation while still maintaining extremely low latency. The SRX Series offers the licensed software feature, services offloading, which provides a mechanism for achieving low latency where it is desired. Services offloading processes fast-path packets in the network processor instead of in the Services Processing Unit (SPU). This method reduces the latency that arises when packets are forwarded from network processors to SPUs for processing and back to I/O cards (IOCs) for transmission.

Services offloading reduces packet-processing latency by 500% to 600%. When the first packet arrives at the interface, the network processor forwards it to the SPU. If the SPU verifies that the traffic is qualified for services offloading, a services-offload session is created on the network processor, and subsequent fast-path packets are processed in the network processor. If the traffic does not qualify for services offloading, a normal session is created on the network processor.

**Note:** Services offloading is supported only on the SRX1400 Services Gateway and the SRX3000 and SRX5000 lines with Flex IOC in the Junos OS Release 11.4. Services offloading has certain limitations. For more information, see the Junos OS Release Notes Limitations section.
Physical and Logical Topologies

Figure 3 shows the high-level physical topology of the lab validation test configuration. The tables list the products and software used.

![Example of data center architecture](image-url)

Table 1: Juniper Networks Products Used

<table>
<thead>
<tr>
<th>Device</th>
<th>Description and Quantity</th>
<th>Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>QFX3500</td>
<td>QFabric Node (8)</td>
<td>Junos 11.3X30</td>
</tr>
<tr>
<td>QFX3008-I</td>
<td>QFabric Interconnect (2)</td>
<td>Junos 11.3X30</td>
</tr>
<tr>
<td>QFX3100</td>
<td>QFabric Director (2)</td>
<td>Junos 11.3X30</td>
</tr>
<tr>
<td>EX4200-48T</td>
<td>Virtual chassis for management connections of QFabric components (8) and high-density servers with 1G connections (2)</td>
<td>Junos 12.1R2.9</td>
</tr>
<tr>
<td>SRX5800</td>
<td>High-end appliance (2 chassis in the cluster)</td>
<td>Junos 12.1R2</td>
</tr>
<tr>
<td>MX480</td>
<td>MX Series 3D Universal Edge Router</td>
<td>Junos 12.1R2</td>
</tr>
<tr>
<td></td>
<td>• RE-S-2000 (quantity 2+2, total four in test bed)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• MPC 3D 16x 10GE (quantity 1+1, total 2 in test bed)</td>
<td></td>
</tr>
</tbody>
</table>
Table 2: Servers

<table>
<thead>
<tr>
<th>Device</th>
<th>Description and Quantity</th>
<th>Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM x3550 M4</td>
<td>Rack mount x Series server with 10G NIC/CNAs (Emulex, Intel)</td>
<td>MS Windows Server 2008R2 x64 SP1 CentOS Linux</td>
</tr>
<tr>
<td>IBM HS22 blade in BC type H</td>
<td>Blade server with Xeon CPU and 10G CFF NIC/CNA (Emulex/Intel/QLogic)</td>
<td>MS Windows Server 2008R2 x64 SP1</td>
</tr>
</tbody>
</table>

Table 3: Test Equipment: Traffic Generators

<table>
<thead>
<tr>
<th>Device</th>
<th>Description and Quantity</th>
<th>Software Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spirent Test Center</td>
<td>Spirent Test Center with 1G and 10G ports, for L2/L3 and L4/L7 traffic</td>
<td>3.90 for L4/L7 3.95 for L2/L3</td>
</tr>
</tbody>
</table>

QFabric in L2 Mode, SRX As FHR

In this configuration, the QFabric solution is in L2 mode, and the server VLANs (111-113, 121-123, 211-213, and 221-223) are terminated on two virtual routers (A and B) on the SRX Series. As shown in Figure 4, VR-C on the SRX Series represents the Internet virtual router that is used for communicating with the MX Series for the Internet and connections to other data centers. VR-A has two zones: A1 and A2. VR-B has two zones: B1 and B2.

![Figure 4: QFabric as L2 switch and SRX as FHR](image_url)

---

Copyright © 2013, Juniper Networks, Inc.
QFabric in L3 Mode, VR-based Traffic Steering

In this configuration, the QFabric solution is in L3 mode, and the server VLANs (111-113,121-123, 211-213, and 221-223) are terminated on four virtual routers (A1, A2, B1, and B2) on the QFabric solution, as shown in Figure 5. Intravirtual routing is handled by the QFabric solution. When a server from one virtual router wants to communicate with another server in another virtual router, it must go through the SRX appliance for security policy evaluation and enforcement. VR-A on the SRX Series provides the routing for the inter-VR traffic between VR-A1 and VR-A2. VR-A has security zones (A1, A2) on the SRX Series. Similarly, VR-B provides the routing for intervirtual router traffic between VR-B1 and VR-B2. VR-C on the SRX Series represents the Internet virtual router used for communicating with the MX Series for the Internet and connections to other data centers. VR-A and VR-B on the SRX Series never communicate directly to the outside world. They go through VR-C for security purposes.

Table 4 lists the IP addresses and network addressing schemes used for both the validation configuration and the logical diagrams (Figures 4 and 5). Appendix A provides the QFabric and SRX Series configurations for the three use cases.

Note: The networks in Table 4 were used for the QFabric solution as the L2 switch (Use Case 1). This information helps understand the logical topologies depicted in Figures 4 and 5 and the device configurations in Appendix A.
Table 4: IP and Network Address Scheme

<table>
<thead>
<tr>
<th>Network</th>
<th>IPv4 Space</th>
<th>VLAN</th>
<th>Virtual Router on SRX Series</th>
<th>Security Zone on SRX Series</th>
<th>Virtual Router on QFabric</th>
<th>For Hosts or Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>net110</td>
<td>10.210.110.0/24</td>
<td>110</td>
<td>VRA</td>
<td>A1</td>
<td>VRA1</td>
<td>Infrastructure. (SRX-QF)</td>
</tr>
<tr>
<td>net111</td>
<td>10.210.111.0/24</td>
<td>111</td>
<td>VRA</td>
<td>A1</td>
<td>VRA1</td>
<td>Hosts</td>
</tr>
<tr>
<td>net112</td>
<td>10.210.112.0/24</td>
<td>112</td>
<td>VRA</td>
<td>A1</td>
<td>VRA1</td>
<td>Hosts</td>
</tr>
<tr>
<td>net113</td>
<td>10.210.113.0/24</td>
<td>113</td>
<td>VRA</td>
<td>A1</td>
<td>VRA1</td>
<td>Hosts</td>
</tr>
<tr>
<td>net120</td>
<td>10.210.120.0/24</td>
<td>120</td>
<td>VRA</td>
<td>A2</td>
<td>VRA2</td>
<td>Infrastructure. (SRX-QF)</td>
</tr>
<tr>
<td>net121</td>
<td>10.210.121.0/24</td>
<td>121</td>
<td>VRA</td>
<td>A2</td>
<td>VRA2</td>
<td>Hosts</td>
</tr>
<tr>
<td>net122</td>
<td>10.210.122.0/24</td>
<td>122</td>
<td>VRA</td>
<td>A2</td>
<td>VRA2</td>
<td>Hosts</td>
</tr>
<tr>
<td>net123</td>
<td>10.210.123.0/24</td>
<td>123</td>
<td>VRA</td>
<td>A2</td>
<td>VRA2</td>
<td>Hosts</td>
</tr>
<tr>
<td>net211</td>
<td>10.210.211.0/24</td>
<td>211</td>
<td>VRB</td>
<td>B1</td>
<td>VRB1</td>
<td>Hosts</td>
</tr>
<tr>
<td>net212</td>
<td>10.210.212.0/24</td>
<td>212</td>
<td>VRB</td>
<td>B1</td>
<td>VRB1</td>
<td>Hosts</td>
</tr>
<tr>
<td>net213</td>
<td>10.210.213.0/24</td>
<td>213</td>
<td>VRB</td>
<td>B1</td>
<td>VRB1</td>
<td>Hosts</td>
</tr>
<tr>
<td>net221</td>
<td>10.210.221.0/24</td>
<td>221</td>
<td>VRB</td>
<td>B2</td>
<td>VRB2</td>
<td>Hosts</td>
</tr>
<tr>
<td>net222</td>
<td>10.210.222.0/24</td>
<td>222</td>
<td>VRB</td>
<td>B2</td>
<td>VRB2</td>
<td>Hosts</td>
</tr>
<tr>
<td>net223</td>
<td>10.210.223.0/24</td>
<td>223</td>
<td>VRB</td>
<td>B2</td>
<td>VRB2</td>
<td>Hosts</td>
</tr>
<tr>
<td>net301</td>
<td>10.210.1.0/24</td>
<td>301</td>
<td>VRC</td>
<td>C1</td>
<td>-</td>
<td>Infrastructure. (SRX-QF-MX)</td>
</tr>
<tr>
<td>net302</td>
<td>10.210.2.0/24</td>
<td>302</td>
<td>VRC</td>
<td>C1</td>
<td>-</td>
<td>Infrastructure. (SRX-QF-MX)</td>
</tr>
<tr>
<td>net303</td>
<td>10.210.3.0/24</td>
<td>303</td>
<td>VRC</td>
<td>C1</td>
<td>-</td>
<td>Infrastructure. (SRX-QF-MX)</td>
</tr>
</tbody>
</table>

Implementation

This section reviews the implementation steps for deploying the SRX Series and QFabric solution in the data center.

QFabric Configuration

Deploying or bring up the QFabric system is not described. We are assuming that the QFabric architecture has been initialized by a certified specialist and is ready to be configured.

A QFabric switch has three types of node groups:
- Server node group (SNG)—By default, every node device that joins the QFabric switch is placed within an automatically generated SNG that contains one node device. SNGs connect to servers and storage devices.
- Redundant SNG (RSNG)—You can assign two node devices in an RSNG. When grouped together, you can create link aggregation groups (LAGs) that span the interfaces on both node devices to provide resiliency and redundancy.
- Network node group (NNG)—You can assign up to eight node devices in an NNG. When grouped together, the node devices within an NNG connect to other routers running routing protocols such as OSPF and BGP.

In this guide, we are connecting the SRX Series to the NNG.

By default, all QFabric nodes are identified by serial number. However, managing devices by serial number can be a challenge. To simplify the management process, you can identify QFabric nodes by more user-friendly descriptions, such as the physical location (row and rack) or node numbers, as shown in the following code snippet.

Configuring Aliases for the Node Devices

```
set fabric aliases node-device P4336-C Node15
set fabric aliases node-device P4649-C Node16
```

Configuring the Role of the Node Devices

```
set fabric resources node-group NW-NG-0 network-domain
set fabric resources node-group NW-NG-0 node-device Node15
set fabric resources node-group NW-NG-0 node-device Node16
```
Interface Naming Conventions for QFabric Technology

The standard Junos OS port naming convention is a three-level identifier: `interface_name-fpc/pic/port_no`. The Flexible PIC Concentrator (FPC) is the first level, and it provides the slot location within the chassis. For QFabric architecture, the three-level identification poses a significant challenge for management because QFabric technology can scale to include up to 128 QFabric nodes, and there is no concept of a slot with QFabric nodes. Therefore, the interface naming convention has been enhanced for QFabric technology to four levels, with an added chassis-level identifier. The new interface naming scheme is `QFabric Node:interface_name-fpc/pic/port`. The QFabric node can be either the serial number or the assigned alias name.

**Note:** This interface naming convention applies only to physical interfaces. For logical interfaces such as LAGs, use `node-group:interface_name-fpc/pic/slot`. Routed VLAN interfaces (rVIs) follow the standard naming convention used by Juniper Networks EX Series Ethernet Switches, which is `vlan.x`.

```
netadmin@qfabric> show interfaces Node15:xe-0/0/10
Physical interface: row1-rack1:xe-0/0/10, Enabled, Physical link is Up
   Interface index: 49182, SNMP ifIndex: 7340572
   Link-level type: Ethernet, MTU: 1514, Speed: 10Gbps, Duplex: Full-Duplex,
   BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
   Source filtering: Disabled, Flow control: Disabled
   Interface flags: Internal: 0x0
   CoS queues : 12 supported, 12 maximum usable queues
   Last flapped
   Input rate
     : 2011-09-06 21:10:51 UTC (04:20:44 ago)
     : 0 bps (0 pps)
   Output rate
     : 0 bps (0 pps)
```

Interface Type Configuration

The following sections cover common configurations for ports and VLANs. QFabric architecture follows the same configuration context as EX Series switches. If you are familiar with configuring EX Series switches, the only difference is the interface naming convention.

There are three different interface types: access, trunk, and routed interface. Just as with any other Junos OS platform, interface configurations are performed under the interface stanza. You can configure the access and trunk ports on any node groups. Routed interfaces are limited to routed VLAN interface (RVI) or NNG ports.

**Note:** Configuring post-mode access is optional. If the port mode is not defined, the default port mode is access. Routed interfaces can be either RVI or L3 ports on an NNG. RVI is a logical L3 interface that provides routing between different networks. The following example shows physical L3 interface configurations on both an NNG and RVI.

**L3 Routed Port on an NNG**

```
set Node15:xe-0/0/0.0 family ethernet-switching port-mode access
set Node15:xe-0/0/1.0 family ethernet-switching port-mode trunk
set Node15:xe-0/0/0.0 family inet address 1.1.1.1/24
```

Configuring an RVI and Binding It to the VLAN

```
set vlan.1250 family inet address 10.83.100.1/24
set vlans v1250 l3-interface vlan.1250
```

For more information about L3 implementations for QFabric deployment options, see:


For more information on how to configure node groups, see:

SRX Series Configuration

Configuring for High Availability

Devices deployed in the data center must ensure consistent service delivery, and the accessibility of the network dictates the availability of the data center’s services. For an HA security infrastructure, one word describes the greatest challenge: state. Most modern devices track the state of the traffic going through the device. When a failure occurs between two active security devices, the state must be shared between them. If the state is not shared, the secondary firewall drops the existing sessions because it is not aware of the state. When a stateful device is deployed, it is important to ensure service continuity so that state can be shared between devices.

The primary goal is to ensure that the SRX Series remains operational despite data plane or control plane loss if failure occurs. The SRX Series platform introduces a new idea to HA design that allows it to fail over the control plane or data plane between chassis. In this hybrid design, the two devices act as one large chassis. The two different systems distribute across the two units as compared to a traditional active/backup cluster in which one device does all the operations while the other device remains idle.

The control plane portion of the cluster is the Routing Engine (RE). The RE can fail over between the two chassis, with the first node passing traffic while the second node maintains the active RE. In the event of a failure, the system that is running on the failed chassis fails over to the second chassis. This is done in a stateful manner for all traffic passing through the device. The only traffic that is lost is what is in the device or wires that fail. In the data center, this provides ease of deployment of active/backup, with the flexibility that the second chassis can provide some backup services.

To configure two SRX Series nodes as an HA chassis cluster, physically connect the two devices (back-to-back for the fabric and control ports) and ensure that they are the same models.

Configuring a Cluster on the SRX Series Device

To configure a chassis cluster, you specify the following:

- Cluster ID and node ID
- Control port
- Fabric port

The following steps are common for all deployment scenarios:

1. Configuring the chassis cluster
2. Configuring redundancy groups
3. Connecting between SRX Series nodes for the cluster
4. Configuring redundant Ethernet interfaces
5. Configuring security zones
6. Configuring security policies

Configuring SRX Series Nodes for the Chassis Cluster

The cluster ID must be between 1 and 15. In each cluster, the node ID must be between 0 and 1. Cluster ID 0 unsets the cluster. You must reboot after the cluster IDs are configured.

```
set chassis cluster cluster-id <0 and 15> node <0 and 1> reboot
set chassis cluster control-ports fpc 6 port 0
set chassis cluster control-ports fpc0 port0
set interfaces fab0 fabric-options member-interfaces xe-3/0/0
set interfaces fab1 fabric-options member-interfaces xe-9/0/0
```
Configuring Redundancy Groups
Redundancy group 0 is reserved for the routing engine. Numbers 1 to 127 are reserved for the interfaces. The node priority determines which node is active for any given redundancy group. The active/passive and active/active SRX Series chassis cluster configuration is achieved by setting appropriate node priority for different redundancy groups and the state of monitored interfaces. The following configuration monitors all interfaces that are part of redundancy group 1.

```bash
set chassis cluster redundancy-group 0 node 0 priority 100
set chassis cluster redundancy-group 0 node 1 priority 200
set chassis cluster redundancy-group 1 node 0 priority 100
set chassis cluster redundancy-group 1 node 1 priority 200
set chassis cluster redundancy-group 1 interface-monitor xe-4/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-4/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-4/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-4/0/3 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-16/0/0 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-16/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-16/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor xe-16/0/3 weight 255
```

Configuring Redundant Ethernet Interfaces
You next configure the redundant Ethernet (reth) interfaces and assign the child interfaces (physical interfaces) to the parent redundant Ethernet interface. The reth interface is assigned to a redundancy group. The active node of the redundancy group determines which interface forwards traffic between the pair of physical interfaces that belong to the reth interface.

```bash
set interfaces xe-4/0/0 gigether-options redundant-parent reth1
set interfaces xe-4/0/1 gigether-options redundant-parent reth1
set interfaces reth1 redundant-ether-options redundancy-group 1
```

Then you configure the logical interface properties for the reth interface. The logical interface is assigned an IP address and runs dynamic routing protocols for network integration. The child interfaces inherit this configuration.

```bash
set interfaces reth1 vlan-tagging
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 111 vlan-id 111
set interfaces reth1 unit 111 family inet address 10.210.111.1/24
set interfaces reth1 unit 112 vlan-id 112
set interfaces reth1 unit 112 family inet address 10.210.112.1/24
```

Configuring Security Zones
Security zones distinguish groups of hosts and their resources from one another. You can then apply different security measures to the interfaces bound to the security zone. Here we create the security zones and assign the logical interfaces to each zone. You also must enable the protocols supported on these interfaces.

```bash
set security zones security-zone A1 address-book address net111 10.210.111.0/24
set security zones security-zone A1 address-book address net112 10.210.112.0/24
set security zones security-zone A1 address-book address net113 10.210.113.0/24
set security zones security-zone A1 host-inbound-traffic system-services all
set security zones security-zone A1 interfaces reth1.111
set security zones security-zone A1 interfaces reth1.112
set security zones security-zone A1 interfaces reth1.113
```
Configuring Security Policies

After the security zones are configured, you define the security policies between the zones. Security policies enforce a set of rules for transit traffic, identifying which traffic can pass through the firewall and the actions taken on the traffic as it passes through the firewall.

```plaintext
set security policies from-zone A1 to-zone A2 policy A1_to_A2-all-permit
description "Policy to allow all traffic from A1 to A2 (test 112A)"
set security policies from-zone A1 to-zone A2 policy A1_to_A2-all-permit match
source-address any
set security policies from-zone A1 to-zone A2 policy A1_to_A2-all-permit match
destination-address any
set security policies from-zone A1 to-zone A2 policy A1_to_A2-all-permit match
application any
set security policies from-zone A1 to-zone A2 policy A1_to_A2-all-permit then
permit
set security policies from-zone A1 to-zone A2 policy A1_to_A2-all-permit then
count
```

**Note:** By default, the Junos OS denies all traffic through an SRX Series device. You can change this behavior by configuring a standard security policy that permits certain types of traffic.

Configuring Inter-VR Routing on the SRX Series

In our test configuration, we created the routing configuration with the following assumptions:

- Routing instances on the SRX Series allow for easy routing separation between virtual routers. We use three virtual routers: VRA, VRB, and VR C.
- VRA and VRB know little about the rest of the world and have simple configurations.
- VRC communicates with the rest of the world (public Internet, other data centers) and learns more than VRA and VRB.
- The default route for VRA and VRB is to point traffic to VRC (static route 0.0.0.0/0 next-table VRC.inet.0).
- VRC learns about networks in VRA and VRB through route leaking using a routing information base (RIB) group configuration.

In large cloud or hosted data center deployments in which multi-tenancy is a requirement, each VR on the QFabric solution can represent a customer, and each VR on the SRX Series can be assigned to an individual customer. This configuration maintains separation throughout the network, although the VRs are using the same physical infrastructure. Customers can choose not to enable inter-VR routing on the SRX Series to maintain strict separation of the routing tables.

**Note:** Creating a static route to place packets into the other routing table works only in one direction because the Junos OS prevents the creation of routing loops if someone attempts to add another static route in the opposite direction.
A RIB group is used for route leaks between VRs. RIB groups are created under the main routing instance. The order of the routing table’s RIB group statements is important. Routes from the first table are imported to other tables.

To configure RIB groups:
1. Under interface-routes, enable importing interface routes through the RIB group.
2. Under static, enable importing static routes through the RIB group.
3. Under ospf, enable importing OSPF learned routes (from QFabric) through the RIB group (to VRC).

```
root@srx001-0# show routing-options
rib-groups {
  VRA-in {
    import-rib [ VRA.inet.0 VRC.inet.0 ];
  }
  VRB-in {
    import-rib [ VRB.inet.0 VRC.inet.0 ];
  }
}
root@srx001-0# show routing-instances
VRA {
  instance-type virtual-router;
  interface reth1.110;
  interface reth1.120;
  routing-options {
    interface-routes {
      rib-group inet VRA-in;
    }
    static {
      rib-group VRA-in;
      route 0.0.0.0/0 next-table VRC.inet.0;
      route 10.210.111.0/24 next-hop 10.210.110.2;
      ... (part of config omitted)
    }
  }
}
protocols {
  ospf {
    rib-group VRA-in;
    area 0.0.0.0 {
      interface reth1.110;
      interface reth1.120;
    }
  }
}
VRC {
  instance-type virtual-router;
  ... (part of config omitted)
}
Connecting the SRX Cluster with the QFabric Solution

We configured the links between the QFabric solution and the SRX Series as reth interfaces on the SRX side and LAG on the QFabric side. In active/active mode, two reth interfaces are configured (reth1 and reth2). In active/passive mode, only reth1 is configured.

Four 10GbE links on the SRX side are part of the reth interface with a reported capacity 20 Gbps. Only two links are used at a time (from the active SRX chassis, one at a time). The links on the QFabric side are configured as two LAG links, each with two 10G ports terminated to different nodes in the NNG.

Figure 7 shows two links with the same color (gray or black); the color denotes that they belong to the same QFabric LAG. The SRX Series uses only one of these at a time.

The resiliency tests show no signs of traffic loss after link recovery between the SRX Series and the QFabric side because of the nonrevertive behavior of the reth interfaces. After a member link fails and failover to another SRX chassis occurs, even if the original link recovers, the reth interface does not switch traffic back to prevent traffic losses in case of link flapping.

You can view the reth interface on the SRX Series as two tied LAGs, one terminating on node 0 and the other on node 1, with the rule that only one of them passes traffic at a time. In Figure 7, only one reth member on the active SRX cluster node is used to pass the traffic.

For more information, see:

http://kb.juniper.net/InfoCenter/index?page=content&id=TN10

IP Routing Between the SRX Series and QFabric Solution

When the QFabric solution acts as an L3 switch, it is routing IP traffic between networks (isolated using VLANs). You can create routing instances on the QFabric side to limit IP visibility between parts of the network. The SRX Series routes the traffic between hosts belonging to the different routing instances on the QFabric side.

On the SRX Series, we need to define security zones to match every routing instance on the QFabric side. You can group the security zones using VRs on the SRX Series. The SRX Series needs to know about all the networks that are terminated on the QFabric side. You can do this by provisioning static routes on the SRX Series or by running a dynamic routing protocol between the SRX and QFabric sides. Both routing configurations were tested.

Static Routing

Static routes route traffic from the SRX Series to the QFabric side and vice versa. The SRX Series must be configured with static routes for all networks configured on the QFabric solution.

In the following configuration, VLANs 110–113 connect the QFabric side with the SRX Series, with default routing pointing to the SRX (10.210.11.1). Two other routing instances are omitted in the following code snippet.
Configuring Routing on QFabric

```bash
root@QFabric# show routing-instances
A1 {
    instance-type virtual-router;
    interface vlan.110;
    interface vlan.111;
    interface vlan.112;
    interface vlan.113;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop 10.210.110.1;
        }
    }
}
A2 {
    instance-type virtual-router;
    interface vlan.120;
    interface vlan.121;
    interface vlan.122;
    interface vlan.123;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop 10.210.120.1;
        }
    }
}
```

Configuring Static Routing on SRX

We want to import RIB groups from inet VRA-in. For every network on the QFabric side, we have added a corresponding static route on the SRX Series.

```bash
root@srx001-0# show routing-instances
VRA {
    instance-type virtual-router;
    interface reth1.110;
    interface reth1.120;
    routing-options {
        interface-routes {
            rib-group inet VRA-in;
        }
        static {
            rib-group VRA-in;
            route 0.0.0.0/0 next-table VRC.inet.0;
            route 10.210.111.0/24 next-hop 10.210.110.2;
            route 10.210.112.0/24 next-hop 10.210.110.2;
            route 10.210.121.0/24 next-hop 10.210.120.2;
            route 10.210.122.0/24 next-hop 10.210.120.2;
            route 10.210.123.0/24 next-hop 10.210.120.2;
        }
    }
}
```
Dynamic Routing
OSPF was used to exchange routing information between the SRX and QFabric sides. The QFabric solution still uses a
default route to point to the SRX Series because it is the only router in this context.

Note: It is important to enable OSPF in the main routing instance, even if ‘other critical configuration’ is in other routing
instances; otherwise, the Junos OSPF process does not start. Security policy for the zone must enable OSPF for inbound traffic.

Configuring OSPF Routing on QFabric

```bash
root@QFabric# show routing-instances
A1 {
  instance-type virtual-router;
  interface vlan.110;
  interface vlan.111;
  interface vlan.112;
  interface vlan.113;
  routing-options {
    static {
      route 0.0.0.0/0 next-hop 10.210.110.1;
    }
    router-id 10.210.110.2;
  }
  protocols {
    ospf {
      domain-id 10.210.110.2;
      area 0.0.0.0 {
        interface vlan.110;
        interface vlan.111 {
          passive;
        }
        interface vlan.112 {
          passive;
        }
        interface vlan.113 {
          passive;
        }
      }
    }
  }
}
A2 {
  instance-type virtual-router;
  interface vlan.120;
  interface vlan.121;
  interface vlan.122;
  interface vlan.123;
  routing-options {
    static {
      route 0.0.0.0/0 next-hop 10.210.120.1;
    }
    router-id 10.210.120.2;
  }
  protocols {
    ospf {
      domain-id 10.210.120.2;
      area 0.0.0.0 {
        interface vlan.120;
      }
    }
  }
}
```
Configuring OSPF Routing on SRX

```
interface vlan.121 {
    passive;
}
interface vlan.122 {
    passive;
}
interface vlan.123 {
    passive;
}
```

Enabling Services Offload (Low-Latency Firewall)

To enable the service-offloading feature:

1. Use IOCs on the SRX Series that have more than one interface per network processor. Only intra-network-processing flows can be offloaded. For the SRX5000 platform, the choice is a flexible IOC (SRX5K-FPC-IOC) and matching port modules (for example, SRX-IOC-4XGE-XFP).
2. Add a license for the Services Offload feature.
3. Configure the cluster to work in active/passive mode. Active/active is not supported.
4. Enable services offloading for each PIC that you want to use, and reboot to apply the changes. For example:
   ```
   set chassis node 0 fpc 0 pic 0 services-offload
   ```
5. Configure services offloading for specific flows in the security policy. For example:
   ```
   set security policies from-zone A1 to-zone C1 policy A1_to_C1-permit2 then permit services-offload
   ```
Some useful commands for troubleshooting services offload include:

- show chassis cluster status
- show chassis fpc pic-status
- show system license usage
- show security flow session
- show security flow session services-offload
- show security policies from-zone A1 to-zone C1 policy-name A1_to_C1-permit2 detail
- show configuration security policies from-zone A1 to-zone C1 policy A1_to_C1-permit2

The following code snippet enables the services offload feature.

```plaintext
node 0 {
    fpc 4 {
        pic 0 {
            services-offload;
        }
        pic 1 {
            services-offload;
        }
    }
}
node 1 {
    fpc 4 {
        pic 0 {
            services-offload;
        }
        pic 1 {
            services-offload;
        }
    }
}
root@srx001-0# show security policies from-zone A1 to-zone C1 policy A1_to_C1-permit2
description "Policy to specifically permit some traffic from A1 to C1 for testing Service Offload";
match {
    source-address net111;
    destination-address net303;
    application [ junos-http junos-udp-any ];
}
root@srx001-0# show chassis
then {
    permit {
        services-offload;
    }
    count;
}
```
AppSecure
Beyond basic network segmentation and security policy control, the SRX Series offers multiple security features to protect data center assets. AppSecure is a suite of next-generation security capabilities that utilize advanced application identification and classification to deliver greater visibility, enforcement, control, and protection over the network.

Working in conjunction with the other security services offered by the SRX Series, AppSecure provides a deep understanding of application behaviors and weaknesses to prevent application-borne threats that are difficult to detect and stop.

For more information on AppSecure and its features, see:

Configuring AppSecure Intrusion Policies
Testing intrusion prevention system (IPS) and intrusion detection and prevention (IDP) features on the SRX Series is done by targeting specific FTP commands. The following are the four major steps involved:

1. Update the security package at https://services.netscreen.com/cgi-bin/index.cgi.

2. Create the IDP policy.

3. Set the IDP policy as an active policy. Only one policy can be active at a time.

4. Apply the IDP policy for traffic that matches the security policy.

Verification was done by initiating an FTP session from zone C1 to the server in zone A1. The server responds and executes commands, except for file transfers where the filename matches "*.exe". The SRX Series drops the session (flow) because it is enforcing the IDP policy.

The following snippet shows the IDP configuration.

```
root@srx001-0# show security idp
idp-policy C1A1-ftp1 {
  rulebase-ips {
    rule 1 {
      match {
        from-zone C1;
        source-address any;
        to-zone A1;
        destination-address any;
        application default;
        attacks {
          custom-attacks ftp1;
        }
      }
      then {
        action {
          drop-connection;
        }
        notification {
          log-attacks;
        }
      }
    }
  }
  active-policy C1A1-ftp1;
  custom-attack ftp1 {
    severity major;
    attack-type {
      signature {
        context ftp-get-filename;
        pattern ".*\.[exe]";
        direction any
      }
    }
  }
}
```
Configuring AppSecure Application Firewalls

The AppSecure AppFw feature is configured on the SRX Series. In the following code snippet, we configure AppFw to block traffic for Dropbox while not blocking similar traffic (TCP port 80) to other services.

```
root@srx001-0# show security application-firewall
rule-sets FileSync-deny {
  rule Dropbox {
    match {
      dynamic-application [ junos:DROPBOX junos:DROPBOX-LAN-SYNC ];
    }
    then {
      deny;
    }
  }
  default-rule {
    permit;
  }
}
root@srx001-0# show security policies from-zone A1 to-zone C1 policy A1_to_C1-permit3
description “Policy to specifically permit some traffic from A1 to C1 (including public Internet) and test app-services”;
match {
  source-address net111;
  destination-address any;
  application [ junos-http junos-https ];
}
then {
  permit {
    application-services {
      application-firewall {
        rule-set FileSync-deny;
      }
    }
    count;
  }
}
Validation Testing and Convergence Testing
The three use case designs have been successfully validated. Convergence testing was performed for the following traffic flows to test resiliency and HA for the firewall deployment.

- Traffic between various VLANs on the QFabric switch
- Traffic between various VRs for permitting and denying traffic on the SRX Series
- Traffic between various security zones within the VRs to check security policy enforcement
- AppSecure IPS and AppFW features on the SRX Series

The following failure scenarios have been tested for full convergence of the network:

- Access link failure
- QFabric to active SRX Series link failure
- Active SRX Series device failure
- QFabric node failure
- SRX Series chassis cluster control link failure
- SRX Series chassis cluster fabric link failure

For details concerning configuration and convergence test results, contact your local Juniper account representative.

Summary
This document explores the different options for integrating a high-speed firewall in the QFabric switch-based data center network. The data center network presents multiple firewall integration points and possibilities for seamlessly fitting in to specific customer environments.

Regardless of the deployment model, the need for a high-speed, low-latency, service-ready firewall system in the data center is paramount. More data centers are consolidating. As more applications are being hosted within these data centers and more complex traffic patterns evolve, additional security and less compromise are required. The Juniper Networks SRX5000 product line offers systems that are data center ready and are architected to support the data centers of today, as well as fulfill specific and individual customer requirements as the data center network transitions and evolves to meet future demands and challenges.
Appendix A: Configuration Files
The following is a complete configuration for the deployment featured in this guide.

Use Case 1: QFabric in Layer 2, SRX As FHR—Active/Passive and Active/Active Cluster Modes

QFabric Layer 2 Configuration (QF-2PS.conf)

```plaintext
system {
    root-authentication {
        encrypted-password "$1$pKp3TsxD$wKH8888alH1Yz6SaSUEW9."; ## SECRET-DATA
    }
}
chassis {
    node-group NW-NG-0 {
        aggregated-devices {
            ethernet {
                device-count 48;
            }
        }
    }
    node-group RSNG-1 {
        aggregated-devices {
            ethernet {
                device-count 48;
            }
        }
    }
    node-group RSNG-2 {
        aggregated-devices {
            ethernet {
                device-count 48;
            }
        }
    }
}
interfaces {
    interface-range LAG1-ae0 {
        member Node01:xe-0/0/0;
        member Node02:xe-0/0/0;
        ether-options {
            802.3ad RSNG-1:ae0;
        }
    }
    interface-range LAG1-ae1 {
        member Node01:xe-0/0/1;
        member Node02:xe-0/0/1;
        ether-options {
            802.3ad RSNG-1:ae1;
        }
    }
    interface-range LAG1-ae2 {
        member Node01:xe-0/0/2;
        member Node02:xe-0/0/2;
        ether-options {
            802.3ad RSNG-1:ae2;
        }
    }
}
```
interface-range LAG1-ae11 {
    member Node01:ge-0/0/11;
    member Node02:ge-0/0/11;
    ether-options {
        802.3ad RSNG-1:ae11;
    }
}
interface-range LAG1-ae12 {
    member Node01:ge-0/0/12;
    member Node02:ge-0/0/12;
    ether-options {
        802.3ad RSNG-1:ae12;
    }
}
interface-range LAG1-ae13 {
    member Node01:ge-0/0/13;
    member Node02:ge-0/0/13;
    ether-options {
        802.3ad RSNG-1:ae13;
    }
}
interface-range LAG1-ae14 {
    member Node01:ge-0/0/14;
    member Node02:ge-0/0/14;
    ether-options {
        802.3ad RSNG-1:ae14;
    }
}
interface-range LAG1-ae15 {
    member Node01:ge-0/0/15;
    member Node02:ge-0/0/15;
    ether-options {
        802.3ad RSNG-1:ae15;
    }
}
interface-range LAG1-ae16 {
    member Node01:ge-0/0/16;
    member Node02:ge-0/0/16;
    ether-options {
        802.3ad RSNG-1:ae16;
    }
}
interface-range LAG1-ae17 {
    member Node01:ge-0/0/17;
    member Node02:ge-0/0/17;
    ether-options {
        802.3ad RSNG-1:ae17;
    }
}
interface-range LAG1-ae18 {
    member Node01:ge-0/0/18;
    member Node02:ge-0/0/18;
    ether-options {
        802.3ad RSNG-1:ae18;
    }
}
interface-range LAG1-ae19 {
    member Node01:ge-0/0/19;
    member Node02:ge-0/0/19;
    ether-options {
        802.3ad RSNG-1:ae19;
    }
}
interface-range LAGR1-ae0 {
    member Node15:xe-0/0/0;
    member Node16:xe-0/0/0;
    ether-options {
        802.3ad NW-NG-0:ae0;
    }
}
interface-range LAGR1-ae1 {
    member Node15:xe-0/0/1;
    member Node16:xe-0/0/1;
    ether-options {
        802.3ad NW-NG-0:ae1;
    }
}
interface-range LAGR1-ae2 {
    member Node15:xe-0/0/2;
    member Node16:xe-0/0/2;
    ether-options {
        802.3ad NW-NG-0:ae2;
    }
}
interface-range LAGR1-ae3 {
    member Node15:xe-0/0/3;
    member Node16:xe-0/0/3;
    ether-options {
        802.3ad NW-NG-0:ae3;
    }
}
interface-range LAGR1-ae8 {
    member "Node09:xe-0/0/[0-1]";
    member "Node10:xe-0/0/[0-1]";
    ether-options {
        802.3ad NW-NG-0:ae8;
    }
}
interface-range LAG2-ae0 {
    member "Node17:xe-0/0/[0-1]";
    member "Node18:xe-0/0/[0-1]";
    ether-options {
        802.3ad RSNG-2:ae0;
    }
}
interface-range LAG2-ae4 {
    member "Node17:xe-0/0/[8-9]";
    member "Node18:xe-0/0/[8-9]";
    ether-options {
        802.3ad RSNG-2:ae4;
    }
}
interface-range LAG1-ae20 {
member Node01:ge-0/0/20;
member Node02:ge-0/0/20;
ether-options {
  802.3ad RSNG-1:ae20;
}
}
interface-range LAG1-ae21 {
  member Node01:ge-0/0/21;
  member Node02:ge-0/0/21;
  ether-options {
    802.3ad RSNG-1:ae21;
  }
}
interface-range LAG1-ae22 {
  member Node01:ge-0/0/22;
  member Node02:ge-0/0/22;
  ether-options {
    802.3ad RSNG-1:ae22;
  }
}
interface-range LAG1-ae23 {
  member Node01:ge-0/0/23;
  member Node02:ge-0/0/23;
  ether-options {
    802.3ad RSNG-1:ae23;
  }
}
interface-range LAG1-ae24 {
  member Node01:ge-0/0/24;
  member Node02:ge-0/0/24;
  ether-options {
    802.3ad RSNG-1:ae24;
  }
}
interface-range LAG1-ae25 {
  member Node01:ge-0/0/25;
  member Node02:ge-0/0/25;
  ether-options {
    802.3ad RSNG-1:ae25;
  }
}
interface-range LAG2-ae5 {
  member "Node17:xe-0/0/[10-11]";
  member "Node18:xe-0/0/[10-11]";
  ether-options {
    802.3ad RSNG-2:ae5;
  }
}
NW-NG-0:ae0 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 111-113 121-123 ];
      }
    }
  }
}
NW-NG-0:ae1 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 111-113 121-123 ];
      }
    }
  }
}
NW-NG-0:ae2 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 211-213 221-223 301-303 ];
      }
    }
  }
}
NW-NG-0:ae3 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 211-213 221-223 301-303 ];
      }
    }
  }
}
NW-NG-0:ae8 {
  aggregated-ether-options {
    lACP {
      active;
    }
  }
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members 301-303;
      }
    }
  }
}
Node02:xe-0/0/1 {
  disable;
}
RSNG-1:ae0 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 111-113 121-123 211-213 221-223 ];
      }
    }
  }
}
RSNG-1:ae1 {
    description "Server with VMs (srv131)";
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 301-303 ];
            }
        }
    }
}

RSNG-1:ae2 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}

RSNG-1:ae3 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}

RSNG-1:ae4 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 301-303 ];
            }
        }
    }
}

RSNG-1:ae5 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}

RSNG-1:ae6 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}
{  
  RSNG-1:ae13 {  
    unit 0 {  
      family ethernet-switching {  
        port-mode trunk;  
        vlan {  
          members [ 111-113 121-123 211-213 221-223 301-303 ];  
        }  
      }  
    }  
  }  
  RSNG-1:ae14 {  
    unit 0 {  
      family ethernet-switching {  
        port-mode trunk;  
        vlan {  
          members 301-303;  
        }  
      }  
    }  
  }  
  RSNG-1:ae15 {  
    unit 0 {  
      family ethernet-switching {  
        port-mode trunk;  
        vlan {  
          members [ 111-113 121-123 211-213 221-223 301-303 ];  
        }  
      }  
    }  
  }  
  RSNG-1:ae16 {  
    unit 0 {  
      family ethernet-switching {  
        port-mode trunk;  
        native-vlan-id 121;  
      }  
    }  
  }  
  RSNG-1:ae17 {  
    unit 0 {  
      family ethernet-switching {  
        port-mode trunk;  
        vlan {  
          members [ 111-113 121-123 211-213 221-223 301-303 ];  
        }  
      }  
    }  
  }  
  RSNG-1:ae18 {  
    unit 0 {  
      family ethernet-switching {  
        port-mode trunk;  
        native-vlan-id 122;  
      }  
    }  
  }  
}
RSNG-1:ae19 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 111-113 121-123 211-213 221-223 301-303 ];
      }
    }
  }
}

RSNG-1:ae20 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      native-vlan-id 111;
    }
  }
}

RSNG-1:ae21 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 111-113 121-123 211-213 221-223 ];
      }
    }
  }
}

RSNG-1:ae22 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      native-vlan-id 303;
    }
  }
}

RSNG-1:ae23 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 111-113 121-123 211-213 221-223 ];
      }
    }
  }
}

RSNG-1:ae24 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      native-vlan-id 303;
    }
  }
}

RSNG-1:ae25 {
  unit 0 {

}
family ethernet-switching {
    port-mode trunk;
    vlan {
        members [ 111-113 121-123 211-213 221-223 ];
    }
}
RSNG-2:ae0 {
    description “EX4200 VC”;
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}
RSNG-2:ae4 {
    description “EX4200VC (ex056 & ex006)”;
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}
RSNG-2:ae5 {
    description STC;
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 301-303 ];
            }
        }
    }
}
protocols {
    lldp {
        interface all;
    }
}
fabric {
    resources {
        node-group NW-NG-0 {
            network-domain;
            node-device Node09;
            node-device Node10;
            node-device Node15;
            node-device Node16;
        }
        node-group RSNG-1 {

node-device Node01;
node-device Node02;
}
node-group RSNG-2 {
  node-device Node17;
  node-device Node18;
}
}

aliases {
  node-device BBAK3172 {
    Node01;
  }
  node-device BBAK3152 {
    Node02;
  }
  node-device P4560-C {
    Node09;
  }
  node-device P4585-C {
    Node10;
  }
  node-device P4336-C {
    Node15;
  }
  node-device P4649-C {
    Node16;
  }
  node-device P4303-C {
    Node17;
  }
  node-device P3333-C {
    Node18;
  }
}

vlans {
  v111 {
    vlan-id 111;
  }
  v112 {
    vlan-id 112;
  }
  v113 {
    vlan-id 113;
  }
  v121 {
    vlan-id 121;
  }
  v122 {
    vlan-id 122;
  }
  v123 {
    vlan-id 123;
  }
  v211 {
    vlan-id 211;
  }
}
SRX Configuration—Active Passive Cluster Mode (SRX-2PS.conf)

```conf
## Last changed: 2012-09-26 19:26:31 EDT
version 12.1R2.9;
groups {
    node0 {
        system {
            host-name srx001-0;
            backup-router 10.13.97.1 destination 0.0.0.0/0;
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 10.13.97.62/24;
                    }
                }
            }
        }
    }
    node1 {
        system {
            host-name srx001-1;
            backup-router 10.13.97.1 destination 0.0.0.0/0;
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 10.13.97.42/24;
                    }
                }
            }
        }
    }
}```
apply-groups "${node}";

system {
  time-zone America/New_York;
  authentication-order [ radius password ];
  root-authentication {
    encrypted-password "$1$ElUDAN5V$C/4LOvuGWfVnM1S4JH8vF."; ## SECRET-DATA
  }
  name-server {
    172.28.113.111;
    172.28.114.111;
  }
  services {
    ssh;
    xnm-clear-text;
    web-management {
      https {
        system-generated-certificate;
      }
    }
  }
  syslog {
    user * {
      any emergency;
    }
    file messages {
      any notice;
      authorization info;
    }
    file interactive-commands {
      interactive-commands any;
    }
  }
  license {
    autoupdate {
      url https://ael.juniper.net/junos/key_retrieval;
    }
  }
  ntp {
    server 172.28.113.123;
    server 172.28.114.123;
  }
}

chassis {
  cluster {
    control-link-recovery;
    reth-count 3;
    control-ports {
      fpc 9 port 0;
      fpc 21 port 0;
      fpc 11 port 1;
      fpc 23 port 1;
    }
    redundancy-group 0 {
node 0 priority 100;
node 1 priority 200;
}
redundancy-group 1 {
  node 0 priority 100;
  node 1 priority 200;
  interface-monitor {
    xe-4/0/0 weight 255;
    xe-4/0/1 weight 255;
    xe-4/0/2 weight 255;
    xe-4/0/3 weight 255;
    xe-16/0/0 weight 255;
    xe-16/0/1 weight 255;
    xe-16/0/2 weight 255;
    xe-16/0/3 weight 255;
  }
}
}
)
}
}

interfaces {
  xe-4/0/0 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  xe-4/0/1 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  xe-4/0/2 {
    gigether-options {
      redundant-parent reth2;
    }
  }
  xe-4/0/3 {
    gigether-options {
      redundant-parent reth2;
    }
  }
  xe-16/0/0 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  xe-16/0/1 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  xe-16/0/2 {
    gigether-options {
      redundant-parent reth2;
    }
  }
  xe-16/0/3 {
    gigether-options {
redundant-parent reth2;
}
}
fab0 {
  fabric-options {
    member-interfaces {
      xe-0/2/0;
      xe-1/2/0;
    }
  }
}
}
fab1 {
  fabric-options {
    member-interfaces {
      xe-12/2/0;
      xe-13/2/0;
    }
  }
}
}
reth1 {
  vlan-tagging;
  redundant-ether-options {
    redundancy-group 1;
  }
}
unit 111 {
  vlan-id 111;
  family inet {
    address 10.210.111.1/24;
  }
}
}
unit 112 {
  vlan-id 112;
  family inet {
    address 10.210.112.1/24;
  }
}
}
unit 113 {
  vlan-id 113;
  family inet {
    address 10.210.113.1/24;
  }
}
}
unit 121 {
  vlan-id 121;
  family inet {
    address 10.210.121.1/24;
  }
}
}
unit 122 {
  vlan-id 122;
  family inet {
    address 10.210.122.1/24;
  }
}
}
unit 123 {
  vlan-id 123;
  family inet {
address 10.210.123.1/24;
}
}
}
reth2 {
  vlan-tagging;
  redundant-ether-options {
    redundancy-group 1;
  }
  unit 211 {
    vlan-id 211;
    family inet {
      address 10.210.211.1/24;
    }
  }
  unit 212 {
    vlan-id 212;
    family inet {
      address 10.210.212.1/24;
    }
  }
  unit 213 {
    vlan-id 213;
    family inet {
      address 10.210.213.1/24;
    }
  }
  unit 221 {
    vlan-id 221;
    family inet {
      address 10.210.221.1/24;
    }
  }
  unit 222 {
    vlan-id 222;
    family inet {
      address 10.210.222.1/24;
    }
  }
  unit 223 {
    vlan-id 223;
    family inet {
      address 10.210.223.1/24;
    }
  }
  unit 301 {
    vlan-id 301;
    family inet {
      address 10.210.1.1/24;
    }
  }
  unit 302 {
    vlan-id 302;
    family inet {
      address 10.210.2.1/24;
    }
  }
}
unit 303 {
    vlan-id 303;
    family inet {
        address 10.210.3.1/24;
    }
}
}
snmp {
    community solutions;
}
}
routing-options {
    static {
        route 10.0.0.0/8 next-hop 10.13.97.1;
        route 172.16.0.0/12 next-hop 10.13.97.1;
        route 0.0.0.0/0 next-hop 10.13.97.1;
    }
    rib-groups {
        VRA-in {
            import-rib [ VRA.inet.0 VRC.inet.0 ];
        }
        VRB-in {
            import-rib [ VRB.inet.0 VRC.inet.0 ];
        }
    }
}
security {
    idp {
        idp-policy C1A1-ftp1 {
            rulebase-ips {
                rule 1 {
                    match {
                        from-zone C1;
                        source-address any;
                        to-zone A1;
                        destination-address any;
                        application default;
                        attacks {
                            custom-attacks ftpl;
                        }
                    }
                    then {
                        action {
                            drop-connection;
                        }
                        notification {
                            log-attacks;
                        }
                    }
                }
            }
        }
        active-policy C1A1-ftp1;
        custom-attack ftpl {
            severity major;
            attack-type {
                signature {
context ftp-get-file;  
    pattern ".*\.[exe]$";  
    direction any;  
}  
}  
security-package {  
    url https://services.netscreen.com/cgi-bin/index.cgi;  
}  
}  
application-firewall {  
    rule-sets FileSync-deny {  
        rule Dropbox {  
            match {  
                dynamic-application [ junos:DROPBOX junos:DROPBOX-LAN-SYNC ];  
            }  
            then {  
                deny;  
            }  
        }  
        default-rule {  
            permit;  
        }  
    }  
}  
screen {  
    ids-option untrusted {  
        icmp {  
            ip-sweep threshold 1000000;  
            fragment;  
            large;  
            flood threshold 10000;  
            ping-death;  
        }  
        ip {  
            bad-option;  
            record-route-option;  
            timestamp-option;  
            security-option;  
            stream-option;  
            spoofing;  
            source-route-option;  
            loose-source-route-option;  
            strict-source-route-option;  
            unknown-protocol;  
            tear-drop;  
        }  
        tcp {  
            syn-fin;  
            fin-no-ack;  
            tcp-no-flag;  
            syn-frag;  
            port-scan threshold 1000000;  
            syn-ack-ack-proxy threshold 10;  
            syn-flood {  
                attack-threshold 100;  
                source-threshold 100;  
            }  
        }  
    }  
}
destination-threshold 100;
   timeout 15;
}
land;
winnuke;
tcp-sweep threshold 1000;
}
udp {
   flood threshold 10000;
   udp-sweep threshold 1000;
}
limit-session {
   source-ip-based 100;
   destination-ip-based 100;
}
policies {
   from-zone A1 to-zone A1 {
      policy A1-deny1 {
         description “Policy to specifically block HTTP traffic between two networks (test 111B)”;
         match {
            source-address net113;
            destination-address net111;
            application junos-http;
         }
         then {
            deny;
            count;
         }
      }
      policy A1-all-permit {
         description “Default intrazone policy to allow all traffic (test 111A)”;
         match {
            source-address any;
            destination-address any;
            application any;
         }
         then {
            permit;
            count;
         }
      }
   }
   from-zone A2 to-zone A2 {
      policy A2-all-permit {
         description “Default intrazone policy to allow all traffic”;
         match {
            source-address any;
            destination-address any;
            application any;
         }
         then {
            permit;
            count;
         }
      }
   }
}
from-zone A1 to-zone A2 {
    policy A1_to_A2-deny1 {
        description "Policy to specifically block traffic from A1 to A2 for two networks (test 112B)";
        match {
            source-address net113;
            destination-address net123;
            application junos-http;
        }
        then {
            deny;
            count;
        }
    }
    policy A1_to_A2-all-permit {
        description "Policy to allow all traffic from A1 to A2 (test 112A)";
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
            count;
        }
    }
}

from-zone A2 to-zone A1 {
    policy A2_to_A1-permit1 {
        description "Policy to specifically permit some traffic from A2 to A1 for two networks (test 121A)";
        match {
            source-address net122;
            destination-address net112;
            application junos-http;
        }
        then {
            permit;
            count;
        }
    }
    policy A2_to_A1-all-deny {
        description "Policy to block all (not-specifically permitted) traffic from A2 to A1 (test 121B)";
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            deny;
            count;
        }
    }
}
from-zone B1 to-zone B1 {
    policy B1-all-permit {
        description "Default intrazone policy to allow all traffic";
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
            count;
        }
    }
}

from-zone B2 to-zone B2 {
    policy B2-all-permit {
        description "Default intrazone policy to allow all traffic";
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
            count;
        }
    }
}

from-zone B1 to-zone B2 {
    policy B1_to_B2 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}

from-zone B2 to-zone B1 {
    policy B2_to_B1 {
        match {
            source-address any;
            destination-address any;
            application junos-http;
        }
        then {
            permit;
            count;
        }
    }
}

from-zone C1 to-zone C1 {
policy C1-all-permit {
    description "Default intrazone policy to allow all traffic";
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        permit;
        count;
    }
}

from-zone C1 to-zone A1 {
    policy C1_to_A1-permit1 {
        description "Policy to specifically permit some traffic from C1 to A1 for two networks (test 151A)";
        match {
            source-address any;
            destination-address [ net111 net112 ];
            application [ junos-http junos-ftp ];
        }
        then {
            permit {
                application-services {
                    idp;
                }
            }
            count;
        }
    }
    policy C1_to_A1-deny1 {
        description "Policy to specifically deny some traffic from C1 to A1 for two networks (test 151B), general rule is to block anyway";
        match {
            source-address net301;
            destination-address net113;
            application junos-http;
        }
        then {
            deny;
            count;
        }
    }
}

from-zone A1 to-zone C1 {
    policy A1_to_C1-deny1 {
        description "Policy to specifically deny some traffic from A1 to C1 for two networks (test 115B), general rule is to block anyway";
        match {
            source-address net111;
            destination-address net302;
            application junos-http;
        }
        then {
            deny;
            count;
        }
    }
}
policy A1_to_C1-permit1 {
    description “Policy to specifically permit some traffic from A1 to C1 for two networks (test 115A)”;
    match {
        source-address net111;
        destination-address net301;
        application junos-http;
    }
    then {
        permit;
        count;
    }
}
policy A1_to_C1-permit2 {
    description “Policy to specifically permit some traffic from A1 to C1”; 
    match {
        source-address net111;
        destination-address net303;
        application [ junos-http junos-udp-any ];
    }
    then {
        permit;
        count;
    }
}
policy A1_to_C1-permit3 {
    description “Policy to specifically permit some traffic from A1 to C1 (including public Internet) and test app-services”; 
    match {
        source-address net111;
        destination-address any;
        application [ junos-http junos-https ];
    }
    then {
        permit {
            application-services {
                application-firewall {
                    rule-set FileSync-deny;
                }
            }
        }
        count;
    }
}
from-zone B1 to-zone C1 {
    policy B1_to_C1-permit2 {
        description “Policy to specifically permit some traffic from B1 to C1”;
        match {
            source-address net211;
            destination-address net303;
            application [ junos-http junos-udp-any ];
        }
    }
}
then {
    permit;
    count;
}

zones {
    security-zone A1 {
        address-book {
            address net111 10.210.111.0/24;
            address net112 10.210.112.0/24;
            address net113 10.210.113.0/24;
        }
        host-inbound-traffic {
            system-services {
                all;
            }
        }
        interfaces {
            reth1.111;
            reth1.112;
            reth1.113;
        }
    }
    security-zone A2 {
        address-book {
            address net121 10.210.121.0/24;
            address net122 10.210.122.0/24;
            address net123 10.210.123.0/24;
        }
        interfaces {
            reth1.121;
            reth1.122;
            reth1.123;
        }
    }
    security-zone B1 {
        address-book {
            address net211 10.210.211.0/24;
            address net212 10.210.212.0/24;
            address net213 10.210.213.0/24;
        }
        interfaces {
            reth2.211;
            reth2.212;
            reth2.213;
        }
    }
    security-zone B2 {
        address-book {
            address net221 10.210.221.0/24;
            address net222 10.210.222.0/24;
            address net223 10.210.223.0/24;
        }
        interfaces {
            reth2.221;
            reth2.222;
            reth2.223;
security-zone C1 {
    address-book {
        address net301 10.210.1.0/24;
        address net302 10.210.2.0/24;
        address net303 10.210.3.0/24;
    }
    screen untrusted;
    interfaces {
        reth2.301;
        reth2.302;
        reth2.303;
    }
}
}
}
routing-instances {
    VRA {
        instance-type virtual-router;
        interface reth1.111;
        interface reth1.112;
        interface reth1.113;
        interface reth1.121;
        interface reth1.122;
        interface reth1.123;
        routing-options {
            interface-routes {
                rib-group inet VRA-in;
            }
            static {
                rib-group VRA-in;
                route 0.0.0.0/0 next-table VRC.inet.0;
            }
        }
    }
    VRB {
        instance-type virtual-router;
        interface reth2.211;
        interface reth2.212;
        interface reth2.213;
        interface reth2.221;
        interface reth2.222;
        interface reth2.223;
        routing-options {
            interface-routes {
                rib-group inet VRB-in;
            }
            static {
                rib-group VRB-in;
                route 0.0.0.0/0 next-table VRC.inet.0;
            }
        }
    }
    VRC {
        instance-type virtual-router;
        interface reth2.301;
interface reth2.302;
interface reth2.303;
routing-options {
  static {
    route 0.0.0.0/0 next-hop 10.210.3.21;
  }
}

SRX Delta Configuration to Convert an Active/Passive Cluster to Active/Active

del chassis cluster redundancy-group 1 interface-monitor xe-4/0/2 weight 255
del chassis cluster redundancy-group 1 interface-monitor xe-4/0/3 weight 255
del chassis cluster redundancy-group 1 interface-monitor xe-16/0/2 weight 255
del chassis cluster redundancy-group 1 interface-monitor xe-16/0/3 weight 255
set chassis cluster redundancy-group 2 node 0 priority 200
set chassis cluster redundancy-group 2 node 1 priority 100
set chassis cluster redundancy-group 2 interface-monitor xe-4/0/2 weight 255
set chassis cluster redundancy-group 2 interface-monitor xe-4/0/3 weight 255
set chassis cluster redundancy-group 2 interface-monitor xe-16/0/2 weight 255
set chassis cluster redundancy-group 2 interface-monitor xe-16/0/3 weight 255
set interfaces reth2 redundant-ether-options redundancy-group 2

Use Case 2: QFabric in Layer 3 Mode, VR-based Traffic Steering to SRX As Needed

QFabric Configuration for Layer 3 Mode with Static Routing (QF-3PS-static.conf)

version "10.4I0 [daveu]";
system {
  root-authentication {
    encrypted-password "$1$pKp3TsxD$wKH8888a1H1Yz6SaSQEW9."; ## SECRET-DATA
  }
}
chassis {
  node-group NW-NG-0 {
    aggregated-devices {
      ethernet {
        device-count 48;
      }
    }
  }
  node-group RSNG-1 {
    aggregated-devices {
      ethernet {
        device-count 48;
      }
    }
  }
  node-group RSNG-2 {
    aggregated-devices {
      ethernet {
        device-count 48;
      }
    }
  }
}
interfaces {
  interface-range LAG1-ae0 {
    member Node01:xe-0/0/0;
    member Node02:xe-0/0/0;
    ether-options {
      802.3ad RSNG-1:ae0;
    }
  }
  interface-range LAG1-ae1 {
    member Node01:xe-0/0/1;
    member Node02:xe-0/0/1;
    ether-options {
      802.3ad RSNG-1:ae1;
    }
  }
  interface-range LAG1-ae2 {
    member Node01:xe-0/0/2;
    member Node02:xe-0/0/2;
    ether-options {
      802.3ad RSNG-1:ae2;
    }
  }
  interface-range LAG1-ae3 {
    member Node01:xe-0/0/3;
    member Node02:xe-0/0/3;
    ether-options {
      802.3ad RSNG-1:ae3;
    }
  }
  interface-range LAG1-ae4 {
    member Node01:xe-0/0/4;
    member Node02:xe-0/0/4;
    ether-options {
      802.3ad RSNG-1:ae4;
    }
  }
  interface-range LAG1-ae5 {
    member Node01:xe-0/0/5;
    member Node02:xe-0/0/5;
    ether-options {
      802.3ad RSNG-1:ae5;
    }
  }
  interface-range LAG1-ae6 {
    member Node01:xe-0/0/6;
    member Node02:xe-0/0/6;
    ether-options {
      802.3ad RSNG-1:ae6;
    }
  }
  interface-range LAG1-ae7 {
    member Node01:xe-0/0/7;
    member Node02:xe-0/0/7;
    ether-options {
      802.3ad RSNG-1:ae7;
    }
  }
}
interface-range LAG1-ae8 {
member Node01:ge-0/0/8;
member Node02:ge-0/0/8;
ether-options {
  802.3ad RSNG-1:ae8;
}

interface-range LAG1-ae9 {
  member Node01:ge-0/0/9;
  member Node02:ge-0/0/9;
  ether-options {
    802.3ad RSNG-1:ae9;
  }
}

interface-range LAG1-ae10 {
  member Node01:ge-0/0/10;
  member Node02:ge-0/0/10;
  ether-options {
    802.3ad RSNG-1:ae10;
  }
}

interface-range LAG1-ae11 {
  member Node01:ge-0/0/11;
  member Node02:ge-0/0/11;
  ether-options {
    802.3ad RSNG-1:ae11;
  }
}

interface-range LAG1-ae12 {
  member Node01:ge-0/0/12;
  member Node02:ge-0/0/12;
  ether-options {
    802.3ad RSNG-1:ae12;
  }
}

interface-range LAG1-ae13 {
  member Node01:ge-0/0/13;
  member Node02:ge-0/0/13;
  ether-options {
    802.3ad RSNG-1:ae13;
  }
}

interface-range LAG1-ae14 {
  member Node01:ge-0/0/14;
  member Node02:ge-0/0/14;
  ether-options {
    802.3ad RSNG-1:ae14;
  }
}

interface-range LAG1-ae15 {
  member Node01:ge-0/0/15;
  member Node02:ge-0/0/15;
  ether-options {
    802.3ad RSNG-1:ae15;
  }
}

interface-range LAG1-ae16 {
member Node01:ge-0/0/16;
member Node02:ge-0/0/16;
ether-options {
  802.3ad RSNG-1:ae16;
}
}
interface-range LAG1-ae17 {
  member Node01:ge-0/0/17;
  member Node02:ge-0/0/17;
  ether-options {
    802.3ad RSNG-1:ae17;
  }
}
interface-range LAG1-ae18 {
  member Node01:ge-0/0/18;
  member Node02:ge-0/0/18;
  ether-options {
    802.3ad RSNG-1:ae18;
  }
}
interface-range LAG1-ae19 {
  member Node01:ge-0/0/19;
  member Node02:ge-0/0/19;
  ether-options {
    802.3ad RSNG-1:ae19;
  }
}
interface-range LAGR1-ae0 {
  member Node15:xe-0/0/0;
  member Node16:xe-0/0/0;
  ether-options {
    802.3ad NW-NG-0:ae0;
  }
}
interface-range LAGR1-ae1 {
  member Node15:xe-0/0/1;
  member Node16:xe-0/0/1;
  ether-options {
    802.3ad NW-NG-0:ae1;
  }
}
interface-range LAGR1-ae2 {
  member Node15:xe-0/0/2;
  member Node16:xe-0/0/2;
  ether-options {
    802.3ad NW-NG-0:ae2;
  }
}
interface-range LAGR1-ae3 {
  member Node15:xe-0/0/3;
  member Node16:xe-0/0/3;
  ether-options {
    802.3ad NW-NG-0:ae3;
  }
}
interface-range LAGR1-ae8 {
member "Node09:xe-0/0/[0-1]"
member "Node10:xe-0/0/[0-1]"
ether-options {
  802.3ad NW-NG-0:ae8;
}
}
interface-range LAG2-ae0 {
  member "Node17:xe-0/0/[0-1]"
  member "Node18:xe-0/0/[0-1]"
  ether-options {
    802.3ad RSNG-2:ae0;
  }
}
interface-range LAG2-ae4 {
  member "Node17:xe-0/0/[8-9]"
  member "Node18:xe-0/0/[8-9]"
  ether-options {
    802.3ad RSNG-2:ae4;
  }
}
interface-range LAG1-ae20 {
  member Node01:ge-0/0/20;
  member Node02:ge-0/0/20;
  ether-options {
    802.3ad RSNG-1:ae20;
  }
}
interface-range LAG1-ae21 {
  member Node01:ge-0/0/21;
  member Node02:ge-0/0/21;
  ether-options {
    802.3ad RSNG-1:ae21;
  }
}
interface-range LAG1-ae22 {
  member Node01:ge-0/0/22;
  member Node02:ge-0/0/22;
  ether-options {
    802.3ad RSNG-1:ae22;
  }
}
interface-range LAG1-ae23 {
  member Node01:ge-0/0/23;
  member Node02:ge-0/0/23;
  ether-options {
    802.3ad RSNG-1:ae23;
  }
}
interface-range LAG1-ae24 {
  member Node01:ge-0/0/24;
  member Node02:ge-0/0/24;
  ether-options {
    802.3ad RSNG-1:ae24;
  }
}
interface-range LAG1-ae25 {
member Node01:ge-0/0/25;
member Node02:ge-0/0/25;
ether-options {
    802.3ad RSNG-1:ae25;
}
}

interface-range LAG2-ae5 {
    member "Node17:xe-0/0/[10-11]";
    member "Node18:xe-0/0/[10-11]";
    ether-options {
        802.3ad RSNG-2:ae5;
    }
}

NW-NG-0:ae0 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 110 120 ];
            }
        }
    }
}

NW-NG-0:ae1 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 110 120 ];
            }
        }
    }
}

NW-NG-0:ae2 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 210 220 301-303 ];
            }
        }
    }
}

NW-NG-0:ae3 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 210 220 301-303 ];
            }
        }
    }
}

NW-NG-0:ae8 {
    aggregated-ether-options {
        lACP {
        ...
active;
}
}
unit 0 {
    family ethernet-switching {
        port-mode trunk;
        vlan {
            members 301-303;
        }
    }
}
RSNG-1:ae0 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}
RSNG-1:ae1 {
    description “Server with VMs (srv131)”;
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 301-303 ];
            }
        }
    }
}
RSNG-1:ae2 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}
RSNG-1:ae3 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}
RSNG-1:ae4 {
port-mode trunk;
  vlan {
    members [ 111-113 121-123 211-213 221-223 301-303 ];
  }
}
RSNG-1:ae5 {
    unit 0 {
      family ethernet-switching {
        port-mode trunk;
        vlan {
          members [ 111-113 121-123 211-213 221-223 ];
        }
      }
    }
}
RSNG-1:ae6 {
    unit 0 {
      family ethernet-switching {
        port-mode trunk;
        vlan {
          members [ 111-113 121-123 211-213 221-223 ];
        }
      }
    }
}
RSNG-1:ae7 {
    unit 0 {
      family ethernet-switching {
        port-mode trunk;
        vlan {
          members [ 111-113 121-123 211-213 221-223 301-303 ];
        }
      }
    }
}
RSNG-1:ae8 {
    unit 0 {
      family ethernet-switching {
        port-mode trunk;
        native-vlan-id 111;
      }
    }
}
RSNG-1:ae9 {
    unit 0 {
      family ethernet-switching {
        port-mode trunk;
        vlan {
          members [ 111-113 121-123 211-213 221-223 301-303 ];
        }
      }
    }
}
RSNG-1:ae10 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            native-vlan-id 112;
        }
    }
}

RSNG-1:ae11 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 301-303 ];
            }
        }
    }
}

RSNG-1:ae12 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            native-vlan-id 113;
        }
    }
}

RSNG-1:ae13 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 301-303 ];
            }
        }
    }
}

RSNG-1:ae14 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members 301-303;
            }
        }
    }
}

RSNG-1:ae15 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 301-303 ];
            }
        }
    }
}

RSNG-1:ae16 {
```yaml
unit 0 {
  family ethernet-switching {
    port-mode trunk;
    native-vlan-id 121;
  }
}
RSNG-1:ae17 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 111-113 121-123 211-213 221-223 301-303 ];
      }
    }
  }
}
RSNG-1:ae18 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      native-vlan-id 122;
    }
  }
}
RSNG-1:ae19 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 111-113 121-123 211-213 221-223 301-303 ];
      }
    }
  }
}
RSNG-1:ae20 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      native-vlan-id 111;
    }
  }
}
RSNG-1:ae21 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
      vlan {
        members [ 111-113 121-123 211-213 221-223 ];
      }
    }
  }
}
RSNG-1:ae22 {
  unit 0 {
    family ethernet-switching {
      port-mode trunk;
```
native-vlan-id 303;
    
}  

RSNG-1:ae23 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}

RSNG-1:ae24 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            native-vlan-id 303;
        }
    }
}

RSNG-1:ae25 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}

RSNG-2:ae0 {
    description "EX4200 VC";
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}

RSNG-2:ae4 {
    description "EX4200VC (ex056 & ex006)";
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 211-213 221-223 ];
            }
        }
    }
}

RSNG-2:ae5 {
    description STC;
    unit 0 {

family ethernet-switching {
    port-mode trunk;
    vlan {
        members [ 111-113 121-123 211-213 221-223 301-303 ];
    }
}

family inet {
    address 10.210.110.2/24;
}

unit 111 {
    family inet {
        address 10.210.111.1/24;
    }
}

unit 112 {
    family inet {
        address 10.210.112.1/24;
    }
}

unit 113 {
    family inet {
        address 10.210.113.1/24;
    }
}

unit 120 {
    family inet {
        address 10.210.120.2/24;
    }
}

unit 121 {
    family inet {
        address 10.210.121.1/24;
    }
}

unit 122 {
    family inet {
        address 10.210.122.1/24;
    }
}

unit 123 {
    family inet {
        address 10.210.123.1/24;
    }
}

unit 210 {
    family inet {
    }
}

unit 211 {
    family inet {
        address 10.210.211.1/24;
    }
}
unit 212 {
    family inet {
        address 10.210.212.1/24;
    }
}

unit 213 {
    family inet {
        address 10.210.213.1/24;
    }
}

unit 220 {
    family inet {
        address 10.210.220.2/24;
    }
}

unit 221 {
    family inet {
        address 10.210.221.1/24;
    }
}

unit 222 {
    family inet {
        address 10.210.222.1/24;
    }
}

unit 223 {
    family inet {
        address 10.210.223.1/24;
    }
}

protocols {
    lldp {
        interface all;
    }
}

routing-instances {
    A1 {
        instance-type virtual-router;
        interface vlan.110;
        interface vlan.111;
        interface vlan.112;
        interface vlan.113;
        routing-options {
            static {
                route 0.0.0.0/0 next-hop 10.210.110.1;
            }
            router-id 10.210.110.2;
        }
    }
    A2 {
        instance-type virtual-router;
        interface vlan.120;
        interface vlan.121;
interface vlan.122;
interface vlan.123;
routing-options {
    static {
        route 0.0.0.0/0 next-hop 10.210.120.1;
    }
    router-id 10.210.120.2;
}
}
B1 {
    instance-type virtual-router;
    interface vlan.210;
    interface vlan.211;
    interface vlan.212;
    interface vlan.213;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop 10.210.210.1;
        }
    }
}
}
B2 {
    instance-type virtual-router;
    interface vlan.220;
    interface vlan.221;
    interface vlan.222;
    interface vlan.223;
    routing-options {
        static {
            route 0.0.0.0/0 next-hop 10.210.220.1;
        }
        router-id 10.210.220.2;
    }
}
}
fabric {
    resources {
        node-group NW-NG-0 {
            network-domain;
            node-device Node09;
            node-device Node10;
            node-device Node15;
            node-device Node16;
        }
        node-group RSNG-1 {
            node-device Node01;
            node-device Node02;
        }
        node-group RSNG-2 {
            node-device Node17;
            node-device Node18;
        }
    }
    aliases {
        node-device BBAK3172 {

Node01;
    }
node-device BBAK3152 {
    Node02;
    }
node-device P4560-C {
    Node09;
    }
node-device P4585-C {
    Node10;
    }
node-device P4336-C {
    Node15;
    }
node-device P4649-C {
    Node16;
    }
node-device P4303-C {
    Node17;
    }
node-device P3333-C {
    Node18;
    }
}
}
vlans {
    v110 {
        vlan-id 110;
        l3-interface vlan.110;
    }
    v111 {
        vlan-id 111;
        l3-interface vlan.111;
    }
    v112 {
        vlan-id 112;
        l3-interface vlan.112;
    }
    v113 {
        vlan-id 113;
        l3-interface vlan.113;
    }
    v120 {
        vlan-id 120;
        l3-interface vlan.120;
    }
    v121 {
        vlan-id 121;
        l3-interface vlan.121;
    }
    v122 {
        vlan-id 122;
        l3-interface vlan.122;
    }
    v123 {
        vlan-id 123;
    }
l3-interface vlan.123;
}
v210 {
  vlan-id 210;
  l3-interface vlan.210;
}
v211 {
  vlan-id 211;
  l3-interface vlan.211;
}
v212 {
  vlan-id 212;
  l3-interface vlan.212;
}
v213 {
  vlan-id 213;
  l3-interface vlan.213;
}
v220 {
  vlan-id 220;
  l3-interface vlan.220;
}
v221 {
  vlan-id 221;
  l3-interface vlan.221;
}
v222 {
  vlan-id 222;
  l3-interface vlan.222;
}
v223 {
  vlan-id 223;
  l3-interface vlan.223;
}
v301 {
  vlan-id 301;
}
v302 {
  vlan-id 302;
}
v303 {
  vlan-id 303;
}
Configuration for Enabling OSPF

```plaintext
set protocols ospf area 0.0.0.0 interface NW-NG-0:ae5.0
set protocols ospf area 0.0.0.0 interface NW-NG-0:ae8.0
set routing-instances A1 protocols ospf domain-id 10.210.110.2
set routing-instances A1 protocols ospf area 0.0.0.0 interface vlan.110
set routing-instances A1 protocols ospf area 0.0.0.0 interface vlan.111 passive
set routing-instances A1 protocols ospf area 0.0.0.0 interface vlan.112 passive
set routing-instances A1 protocols ospf area 0.0.0.0 interface vlan.113 passive
set routing-instances A2 protocols ospf domain-id 10.210.120.2
set routing-instances A2 protocols ospf area 0.0.0.0 interface vlan.120
set routing-instances A2 protocols ospf area 0.0.0.0 interface vlan.121 passive
set routing-instances A2 protocols ospf area 0.0.0.0 interface vlan.122 passive
set routing-instances A2 protocols ospf area 0.0.0.0 interface vlan.123 passive
set routing-instances B1 protocols ospf area 0.0.0.0 interface vlan.210
set routing-instances B1 protocols ospf area 0.0.0.0 interface vlan.211 passive
set routing-instances B1 protocols ospf area 0.0.0.0 interface vlan.212 passive
set routing-instances B1 protocols ospf area 0.0.0.0 interface vlan.213 passive
set routing-instances B2 protocols ospf area 0.0.0.0 interface vlan.220
set routing-instances B2 protocols ospf area 0.0.0.0 interface vlan.221 passive
set routing-instances B2 protocols ospf area 0.0.0.0 interface vlan.222 passive
set routing-instances B2 protocols ospf area 0.0.0.0 interface vlan.223 passive
```

SRX Configuration—Active/Passive Cluster (SRX-3PS-static)

```plaintext
## Last changed: 2012-09-26 20:20:17 EDT
version 12.1R2.9;
groups {
    node0 {
        system {
            host-name srx001-0;
            backup-router 10.13.97.1 destination 0.0.0.0/0;
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 10.13.97.62/24;
                    }
                }
            }
        }
    }
    node1 {
        system {
            host-name srx001-1;
            backup-router 10.13.97.1 destination 0.0.0.0/0;
        }
        interfaces {
            fxp0 {
                unit 0 {
                    family inet {
                        address 10.13.97.42/24;
                    }
                }
            }
        }
    }
}
```
apply-groups "${node}";

system {
  time-zone America/New_York;
  authentication-order [ radius password ];
  root-authentication {
    encrypted-password "$1$ElUDAN5V$C/4LOvuGWfVnM1S4JHHyF."; ## SECRET-DATA
  }
  name-server {
    172.28.113.111;
    172.28.114.111;
  }
  services {
    ssh;
    xnm-clear-text;
    web-management {
      https {
        system-generated-certificate;
      }
    }
  }
  syslog {
    user * {
      any emergency;
    }
    file messages {
      any notice;
      authorization info;
    }
    file interactive-commands {
      interactive-commands any;
    }
  }
  license {
    autoupdate {
      url https://ae1.juniper.net/junos/key_retrieval;
    }
  }
  ntp {
    server 172.28.113.123;
    server 172.28.114.123;
  }
}

chassis {
  cluster {
    control-link-recovery;
    reth-count 3;
    control-ports {
      fpc 9 port 0;
      fpc 21 port 0;
      fpc 11 port 1;
      fpc 23 port 1;
    }
    redundancy-group 0 {

node 0 priority 100;
node 1 priority 200;
}
redundancy-group 1 {
  node 0 priority 100;
  node 1 priority 200;
  interface-monitor {
    xe-4/0/0 weight 255;
    xe-4/0/1 weight 255;
    xe-4/0/2 weight 255;
    xe-4/0/3 weight 255;
    xe-16/0/0 weight 255;
    xe-16/0/1 weight 255;
    xe-16/0/2 weight 255;
    xe-16/0/3 weight 255;
  }
}
}
}

interfaces {
  xe-4/0/0 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  xe-4/0/1 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  xe-4/0/2 {
    gigether-options {
      redundant-parent reth2;
    }
  }
  xe-4/0/3 {
    gigether-options {
      redundant-parent reth2;
    }
  }
  xe-16/0/0 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  xe-16/0/1 {
    gigether-options {
      redundant-parent reth1;
    }
  }
  xe-16/0/2 {
    gigether-options {
      redundant-parent reth2;
    }
  }
  xe-16/0/3 {

gigether-options {
    redundant-parent reth2;
}
}

fab0 {
    fabric-options {
        member-interfaces {
            xe-0/2/0;
            xe-1/2/0;
        }
    }
}

fab1 {
    fabric-options {
        member-interfaces {
            xe-12/2/0;
            xe-13/2/0;
        }
    }
}

reth1 {
    vlan-tagging;
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 110 {
        vlan-id 110;
        family inet {
            address 10.210.110.1/24;
        }
    }
    unit 120 {
        vlan-id 120;
        family inet {
            address 10.210.120.1/24;
        }
    }
}

reth2 {
    vlan-tagging;
    redundant-ether-options {
        redundancy-group 1;
    }
    unit 210 {
        vlan-id 210;
        family inet {
        }
    }
    unit 220 {
        vlan-id 220;
        family inet {
            address 10.210.220.1/24;
        }
    }
    unit 301 {
vlan-id 301;
family inet {
    address 10.210.1.1/24;
}
unit 302 {
    vlan-id 302;
    family inet {
        address 10.210.2.1/24;
    }
}
unit 303 {
    vlan-id 303;
    family inet {
        address 10.210.3.1/24;
    }
}
snmp {
    community solutions;
}
routing-options {
    static {
        route 10.0.0.0/8 next-hop 10.13.97.1;
        route 172.16.0.0/12 next-hop 10.13.97.1;
        route 0.0.0.0/0 next-hop 10.13.97.1;
    }
    rib-groups {
        VRA-in {
            import-rib [ VRA.inet.0 VRC.inet.0 ];
        }
        VRB-in {
            import-rib [ VRB.inet.0 VRC.inet.0 ];
        }
    }
}
security {
    idp {
        idp-policy C1A1-ftp1 {
            rulebase-ips {
                rule 1 {
                    match {
                        from-zone C1;
                        source-address any;
                        to-zone A1;
                        destination-address any;
                        application default;
                        attacks {
                            custom-attacks ftp1;
                        }
                    }
                    then {
                        action {
                            drop-connection;
                        }
                    }
                    notification {
                    }
                }
            }
        }
    }
}
log-attacks;

active-policy C1A1-ftp1;
custom-attack ftp1 {
    severity major;
    attack-type {
        signature {
            context ftp-get-filename;
            pattern ".*\.\[exe\]";
            direction any;
        }
    }
}

security-package {
    url https://services.netscreen.com/cgi-bin/index.cgi;
}

application-firewall {
    rule-sets FileSync-deny {
        rule Dropbox {
            match {
                dynamic-application [ junos:DROPBOX junos:DROPBOX-LAN-SYNC ];
            }
            then {
                deny;
            }
        }
        default-rule {
            permit;
        }
    }
}

screen {
    ids-option untrusted {
        icmp {
            ip-sweep threshold 1000000;
            fragment;
            large;
            flood threshold 10000;
            ping-death;
        }
        ip {
            bad-option;
            record-route-option;
            timestamp-option;
            security-option;
            stream-option;
            spoofing;
            source-route-option;
            loose-source-route-option;
            strict-source-route-option;
            unknown-protocol;
        }
    }
}
tear-drop;
}

tcp {
    syn-fin;
    fin-no-ack;
    tcp-no-flag;
    syn-frag;
    port-scan threshold 1000000;
    syn-ack-ack-proxy threshold 10;
    syn-flood {
        attack-threshold 100;
        source-threshold 100;
        destination-threshold 100;
        timeout 15;
    }
    land;
    winnuke;
    tcp-sweep threshold 1000;
}

udp {
    flood threshold 10000;
    udp-sweep threshold 1000;
}

limit-session {
    source-ip-based 100;
    destination-ip-based 100;
}
}

policies {
    from-zone A1 to-zone A1 {
        policy A1-deny1 {
            description "Policy to specifically block HTTP traffic between two networks (test 111B)";
            match {
                source-address net113;
                destination-address net111;
                application junos-http;
            }
            then {
                deny;
                count;
            }
        }
        policy A1-all-permit {
            description "Default intrazone policy to allow all traffic (test 111A)";
            match {
                source-address any;
                destination-address any;
                application any;
            }
            then {
                permit;
                count;
            }
        }
    }
}
from-zone A2 to-zone A2 {
    policy A2-all-permit {
        description "Default intrazone policy to allow all traffic";
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
            count;
        }
    }
}

from-zone A1 to-zone A2 {
    policy A1_to_A2-deny1 {
        description "Policy to specifically block traffic from A1 to A2 for two networks (test 112B)";
        match {
            source-address net113;
            destination-address net123;
            application junos-http;
        }
        then {
            deny;
            count;
        }
    }
    policy A1_to_A2-all-permit {
        description "Policy to allow all traffic from A1 to A2 (test 112A)";
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
            count;
        }
    }
}

from-zone A2 to-zone A1 {
    policy A2_to_A1-permit1 {
        description "Policy to specifically permit some traffic from A2 to A1 for two networks (test 121A)";
        match {
            source-address net122;
            destination-address net112;
            application junos-http;
        }
        then {
            permit;
            count;
        }
    }
}
policy A2_to_A1-all-deny {
    description "Policy to block all (not-specifically permitted) traffic from A2 to A1 (test 121B)";
    match {
        source-address any;
        destination-address any;
        application any;
    }
    then {
        deny;
        count;
    }
}

from-zone B1 to-zone B1 {
    policy B1-all-permit {
        description "Default intrazone policy to allow all traffic";
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
            count;
        }
    }
}

from-zone B2 to-zone B2 {
    policy B2-all-permit {
        description "Default intrazone policy to allow all traffic";
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
            count;
        }
    }
}

from-zone B1 to-zone B2 {
    policy B1_to_B2 {
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
        }
    }
}

from-zone B2 to-zone B1 {
policy B2_to_B1 {
    match {
        source-address any;
        destination-address any;
        application junos-http;
    }
    then {
        permit;
        count;
    }
}
}
from-zone C1 to-zone C1 {
    policy C1-all-permit {
        description "Default intrazone policy to allow all traffic";
        match {
            source-address any;
            destination-address any;
            application any;
        }
        then {
            permit;
            count;
        }
    }
}
from-zone C1 to-zone A1 {
    policy C1_to_A1-permit1 {
        description "Policy to specifically permit some traffic from C1 to A1 for two networks (test 151A)";
        match {
            source-address any;
            destination-address [ net111 net112 ];
            application [ junos-http junos-ftp ];
        }
        then {
            permit {
                application-services {
                    idp;
                }
            }
            count;
        }
    }
    policy C1_to_A1-deny1 {
        description "Policy to specifically deny some traffic from C1 to A1 for two networks (test 151B), general rule is to block anyway";
        match {
            source-address net301;
            destination-address net113;
            application junos-http;
        }
        then {
            deny;
            count;
        }
    }
}
from-zone A1 to-zone C1 {
  policy A1_to_C1-deny1 {
    description "Policy to specifically deny some traffic from A1 to C1 for two networks (test 115B), general rule is to block anyway";
    match {
      source-address net111;
      destination-address net302;
      application junos-http;
    }
    then {
      deny;
      count;
    }
  }
  policy A1_to_C1-permit1 {
    description "Policy to specifically permit some traffic from A1 to C1 for two networks (test 115A)";
    match {
      source-address net111;
      destination-address net301;
      application junos-http;
    }
    then {
      permit;
      count;
    }
  }
  policy A1_to_C1-permit2 {
    description "Policy to specifically permit some traffic from A1 to C1";
    match {
      source-address net111;
      destination-address net303;
      application [ junos-http junos-udp-any ];
    }
    then {
      permit;
      count;
    }
  }
  policy A1_to_C1-permit3 {
    description "Policy to specifically permit some traffic from A1 to C1 (including public Internet) and test app-services";
    match {
      source-address net111;
      destination-address any;
      application [ junos-http junos-https ];
    }
    then {
      permit {
        application-services {
          application-firewall {
            rule-set FileSync-deny;
          }
        }
      }
    }
  }
}
from-zone B1 to-zone C1 {
  policy B1_to_C1-permit2 {
    description "Policy to specifically permit some traffic from B1 to C1";
    match {
      source-address net211;
      destination-address net303;
      application [junos-http junos-udp-any];
    }
    then {
      permit;
      count;
    }
  }
}

zones {
  security-zone A1 {
    address-book {
      address net111 10.210.111.0/24;
      address net112 10.210.112.0/24;
      address net113 10.210.113.0/24;
    }
    host-inbound-traffic {
      system-services {
        all;
      }
    }
    interfaces {
      reth1.110;
    }
  }
  security-zone A2 {
    address-book {
      address net121 10.210.121.0/24;
      address net122 10.210.122.0/24;
      address net123 10.210.123.0/24;
    }
    interfaces {
      reth1.120;
    }
  }
  security-zone B1 {
    address-book {
      address net211 10.210.211.0/24;
      address net212 10.210.212.0/24;
      address net213 10.210.213.0/24;
    }
    interfaces {
      reth2.210;
    }
  }
}
security-zone B2 {
    address-book {
        address net221 10.210.221.0/24;
        address net222 10.210.222.0/24;
        address net223 10.210.223.0/24;
    }
    interfaces {
        reth2.220;
    }
}

security-zone C1 {
    address-book {
        address net301 10.210.1.0/24;
        address net302 10.210.2.0/24;
        address net303 10.210.3.0/24;
    }
    screen untrusted;
    interfaces {
        reth2.301;
        reth2.302;
        reth2.303;
    }
}

routing-instances {
    VRA {
        instance-type virtual-router;
        interface reth1.110;
        interface reth1.120;
        routing-options {
            interface-routes {
                rib-group inet VRA-in;
            }
            static {
                rib-group VRA-in;
                route 0.0.0.0/0 next-table VRC.inet.0;
                route 10.210.111.0/24 next-hop 10.210.110.2;
                route 10.210.112.0/24 next-hop 10.210.110.2;
                route 10.210.121.0/24 next-hop 10.210.120.2;
                route 10.210.122.0/24 next-hop 10.210.120.2;
                route 10.210.123.0/24 next-hop 10.210.120.2;
            }
        }
    }
    VRB {
        instance-type virtual-router;
        interface reth2.210;
        interface reth2.220;
        routing-options {
            interface-routes {
                rib-group inet VRB-in;
            }
            static {
                rib-group VRB-in;
                route 0.0.0.0/0 next-table VRC.inet.0;
                route 10.210.111.0/24 next-hop 10.210.110.2;
                route 10.210.112.0/24 next-hop 10.210.110.2;
                route 10.210.121.0/24 next-hop 10.210.120.2;
                route 10.210.122.0/24 next-hop 10.210.120.2;
                route 10.210.123.0/24 next-hop 10.210.120.2;
            }
        }
    }
}
Delta Configuration to Enable OSPF Instead of Static Routing on the SRX

```plaintext
set protocols ospf area 0.0.0.0 interface ge-3/0/0.0
set security zones security-zone A1 host-inbound-traffic protocols ospf
set security zones security-zone A2 host-inbound-traffic protocols ospf
set security zones security-zone B1 host-inbound-traffic protocols ospf
set security zones security-zone B2 host-inbound-traffic protocols ospf
set security zones security-zone C1 host-inbound-traffic protocols ospf
set routing-instances VRA protocols ospf rib-group VRA-in
set routing-instances VRA protocols ospf area 0.0.0.0 interface reth1.110
set routing-instances VRA protocols ospf area 0.0.0.0 interface reth1.120
set routing-instances VRB protocols ospf rib-group VRB-in
set routing-instances VRB protocols ospf area 0.0.0.0 interface reth2.210
set routing-instances VRB protocols ospf area 0.0.0.0 interface reth2.220
set routing-instances VRB protocols ospf area 0.0.0.0 interface reth2.310
set routing-instances VRB protocols ospf area 0.0.0.0 interface reth2.320
set routing-instances VRB protocols ospf area 0.0.0.0 interface reth2.330
del routing-instances VRA routing-options static route 10.210.121.0/24 next-hop 10.210.120.2
del routing-instances VRA routing-options static route 10.210.122.0/24 next-hop 10.210.120.2
del routing-instances VRA routing-options static route 10.210.123.0/24 next-hop 10.210.120.2
```

![Delta Configuration to Enable OSPF Instead of Static Routing on the SRX](image-url)
Use Case 3: SRX with Services Offload
SRX Delta Configuration for Enabling Services Offload

set chassis node 0 fpc 4 pic 0 services-offload
set chassis node 0 fpc 4 pic 1 services-offload
set chassis node 0 fpc 9 pic 0 services-offload
set chassis node 0 fpc 9 pic 1 services-offload
set chassis node 0 fpc 11 pic 0 services-offload
set chassis node 0 fpc 11 pic 1 services-offload
set chassis node 1 fpc 4 pic 0 services-offload
set chassis node 1 fpc 4 pic 1 services-offload
set chassis node 1 fpc 9 pic 0 services-offload
set chassis node 1 fpc 9 pic 1 services-offload
set chassis node 1 fpc 11 pic 0 services-offload
set chassis node 1 fpc 11 pic 1 services-offload
set security policies from-zone A1 to-zone C1 policy A1_to_C1-permit2 description
“Policy to specifically permit some traffic from A1 to C1 for testing Service
Offload”
set security policies from-zone A1 to-zone C1 policy A1_to_C1-permit2 then permit
services-offload
set security policies from-zone B1 to-zone C1 policy B1_to_C1-permit2 description
“Policy to specifically permit some traffic from B1 to C1 for testing Service
Offload”
set security policies from-zone B1 to-zone C1 policy B1_to_C1-permit2 then permit
services-offload
Appendix B: QFabric Interfaces for Servers

In our configuration, QFabric nodes are configured in pairs to provide RSNGs. Interfaces for servers (hosts) are configured as LAG links with two members, one out of each node member for HA purposes. This configuration approach is well documented in the implementation guide *Designing a Layer 2 Data Center Network with the QFabric Architecture*.

For server ports, it makes sense to enable and use Link Layer Discovery Protocol (LLDP), which is globally enabled in our configuration. LLDP can be used as a troubleshooting tool as well as provide service for Data Center Bridging Capability Exchange (DCBx).

Configuration snippet for server interfaces on the QFabric side:

```plaintext
interface-range LAG1-ae0 {
    member Node01:xe-0/0/0;
    member Node02:xe-0/0/0;
    ether-options {
        802.3ad RSNG-1:ae0;
    }
}
RSNG-1:ae0 {
    unit 0 {
        family ethernet-switching {
            port-mode trunk;
            vlan {
                members [ 111-113 121-123 ];
            }
        }
    }
}

Or alternative, shorter CLI syntax

set interfaces interface-range LAG1-ae0 member Node01:xe-0/0/0
set interfaces interface-range LAG1-ae0 member Node02:xe-0/0/0
set interfaces interface-range LAG1-ae0 ether-options 802.3ad RSNG-1:ae0
set interfaces RSNG-1:ae0 unit 0 family ethernet-switching port-mode trunk
set interfaces RSNG-1:ae0 unit 0 family ethernet-switching vlan members [ 111-113 121-123 ]
```

QFabric Interfaces for EX Switches

To allow for higher density of 1G server connections in the data center, you can use the EX4200 line of switches. These switches are connected in the VC for HA setup. You can connect the switch to the QFabric side using 10G links, with each switch having one or two uplinks (a total of two or four per VC). This approach is documented in *Designing a Layer 2 Data Center Network with the QFabric Architecture*.

Bridge protocol data unit (BPDU) protect is enabled by default on the QFabric architecture. As soon as it sees the BPDU frame on the port, the port is flagged and disabled. If a timeout is not configured on the port for this type of error, the port remains disabled until administrator issues the clear command (clear ethernet-switching bpdu-error interface x). Be sure to disable all spanning-tree protocols on the EX ports facing QFabric.

Configuration Snippet for EX VC Interfaces on QFabric

```plaintext
interface-range LAG2-ae4 {
    member “Node17:xe-0/0/[8-9]”;  
    member “Node18:xe-0/0/[8-9]”; 
    ether-options {
        802.3ad RSNG-2:ae4;
    }
}
```
About Juniper Networks

Juniper Networks is in the business of network innovation. From devices to data centers, from consumers to cloud providers, Juniper Networks delivers the software, silicon and systems that transform the experience and economics of networking. The company serves customers and partners worldwide. Additional information can be found at www.juniper.net.