

Network Configuration Example

Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks

Release
11.2



Published: 2012-01-27

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. Gated is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of Gated has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., copyright © 1996, 1997, Maker Communications, Inc.

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

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

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

Network Configuration Example Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks

Release 11.2

Copyright © 2012, Juniper Networks, Inc.

All rights reserved.

Revision History

April 2011—R1 Junos OS 11.2

The information in this document is current as of the date listed in the revision history.

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

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

Table of Contents

Introduction	1
Requirements of a Two-Tiered Virtualized Data Center for Large Enterprise	
Networks	1
Network Traffic Segmentation	1
Flexibility	1
Security	2
Access and Availability	2
Cost-Effective Incremental Scaling	2
Orchestration and Automation	2
Two-Tiered Virtualized Data Center Solution for Large Enterprise Networks	2
Network Traffic Segmentation	3
Flexibility	3
Security	3
Access and Availability	3
Cost-Effective Incremental Scaling	4
Orchestration and Automation	4
Example: Configuring a Two-Tiered Virtualized Data Center for Large Enterprise	
Networks	5

Introduction

This network configuration example provides a step-by-step procedure to configure a two-tiered virtualized data center using Juniper Networks MX Series 3D Universal Edge Routers, EX Series Ethernet Switches, and SRX Series Services Gateways.

Requirements of a Two-Tiered Virtualized Data Center for Large Enterprise Networks

Large enterprises have certain specific needs for the hosting environment that the design of their data center must meet. This section describes the requirements of a company that operates as a service provider to its individual business units (BUs).

One of the primary requirements of a virtualized data center (VDC) for a large enterprise is the ability to segment the network by business unit. This includes traffic segmentation and administrative control segmentation.

Other requirements include security controls between business units, security controls between the company and the outside world, flexibility to grow and adapt the network, and a robust and cost-effective way to manage the entire network.

Network Traffic Segmentation

The requirement described here is for network resources to be isolated in several ways. Traffic must be segmented by business units. Traffic flows between network segments must be prohibited except where specifically allowed. Traffic isolation must be controlled at designated policy enforcement points. Network resources must be dedicated to a segment, but the network must have the flexibility to change the allocation of resources.

Segmented resources must be logically grouped according to policies. For example, test traffic must be isolated from production traffic. Traffic must also be isolated according to business entities, contractual requirements, legal or regulatory requirements, risk rating, and corporate standards.

The network segmentation design must not be disruptive to the business, must be integrated with the larger data center and cloud network design, must allow business units to access network resources globally, and must support new business capabilities.

Flexibility

The network design must be flexible enough to react to business and environment changes with minimal design and re-engineering efforts. The VDC design must be flexible in terms of isolating business unit workloads from other business units and general data center services and applications. The network solution must ensure that the business is minimally impacted when network and segmentation changes take place.

The VDC must be flexible enough to be implemented:

- Within a single data center
- Within a data hall
- Across two or more data centers

- Across two or more data halls within or between data centers
- Between a data center and an external cloud service provider

Security

The network design must allow business units to be isolated within the hosting environment. In the event of a network security incident, business units must be isolated from the hosting environment and other business units.

Traffic flow between business unit segments must be denied by default and must be explicitly permitted only at policy enforcement points owned and controlled by the data center service provider.

The policy enforcement point must include access control capabilities and might include threat protection capabilities.

Access and Availability

The VDC must provide access to common data center services such as computation, storage, security, traffic management, operations, and applications. The network must operate across multiple global service providers and must deliver optimal, predictable, and consistent performance across the network. The VDC must be implemented across data center business units.

The network solution must meet business unit availability requirements as defined in service-level agreements.

Cost-Effective Incremental Scaling

The VDC design must be cost effective for the business to run and must enable new business capabilities. It must be possible to implement the network solution in an incremental manner with minimal impact to the business.

Orchestration and Automation

The VDC design must include a management system that supports automation for provisioning, availability and workload monitoring, and reporting. Workload and availability reports must be available by business unit.

Related Documentation

- [Two-Tiered Virtualized Data Center Solution for Large Enterprise Networks on page 2](#)
- [Example: Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks on page 5](#)

Two-Tiered Virtualized Data Center Solution for Large Enterprise Networks

The following describes a Juniper Networks two-tiered, high-speed, multiservice virtualized data center (VDC). A two-tiered architecture meets the low latency requirements of a virtualized server environment and supports the overlying security mandate to maintain controlled segmentation between various business units.

Network Traffic Segmentation

Juniper Networks VDC design uses virtualization technologies such as virtual LANs (VLANs), virtual routers, virtual route forwarders, inter-virtual route forwarding, and logical systems to provide flexible traffic isolation.

A fully redundant two-tiered data center design consists of Juniper Networks EX Series Ethernet Switches at the access layer for server connectivity, MX Series 3D Universal Edge Routers as a collapsed LAN aggregation/core layer, and clustered SRX Series Services Gateways to provide firewall security services across the data center trust boundaries.

Flexibility

The Juniper Networks VDC design uses 802.1Q VLANs, MPLS, BGP, Virtual Router Redundancy Protocol (VRRP), Traffic Engineering, and Fast Reroute to provide design flexibility while maintaining a standards-based approach. The design can also support a virtual private LAN service (VPLS).

Security

The Juniper Networks VDC design uses security zones to implement the policy enforcement points. The SRX cluster is responsible for all stateful packet inspection for traffic that crosses business unit trust boundaries as well as all ingress and egress traffic for the data center.

The Juniper Networks Junos operating system is configured with different administrator accounts for each logical system that supports confined access to network resources and can be customized for individual business units.

Access and Availability

In the Juniper Networks VDC design, described in [“Example: Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks” on page 5](#), top-of-rack (TOR) EX Series switches provide access to the servers and provide redundancy.

All uplinks from the TOR switches are 802.1Q trunk links that are terminated directly into each of the MX Series devices that make up the Point of Delivery (POD) at the aggregation/core layer.

A VRRP instance is defined on each VLAN within the MX Series device to act as the default router for all server hosts in a given VLAN. To allow for VRRP to work properly, each bridge domain is extended between each MX Series device through an interconnection link. The MX Series device uses an integrated routing and bridging (IRB) interface as the Layer 3 interface for each bridge domain, with VRRP configured for redundancy.

A pair of 802.3ad aggregated Ethernet bundles are used between the MX Series devices. Each MX Series device is divided into a number of Logical Systems. Logical systems in the MX Series device are used to define logical trust boundaries within the data center itself and between respective business units.

A clustered pair of SRX Series devices acting as firewalls provide security services across the data center trust boundaries. Virtual routers on the SRX Series devices act as customer edge (CE) routers for each business unit.

A single redundancy group for the data plane is defined on the SRX Series Services Gateways with two redundant Ethernet interfaces as member interfaces. This redundancy group handles the data plane failover of the SRX Series firewall and is configured such that any loss of either northbound or southbound SRX Series interfaces forces a full failover to the secondary node. This failover is essentially a Layer 1 failover, which means that it occurs quickly and does not disrupt the routing topology above it.

Cost-Effective Incremental Scaling

The Juniper Networks VDC design supports incremental scaling of the network. This allows the VDC to be created with minimum cost to meet the current need.

The access layer can be expanded by adding EX Series switches at the top of rack.

The aggregation/core layer can be expanded by adding additional MX Series devices within a given POD.

The security services can be expanded by adding 4-port 10-Gigabit Ethernet I/O cards (IOCs) and services processing cards (SPCs) in the SRX Series devices. The addition of IOCs increases the 10-Gigabit Ethernet port density. The addition of each SPC card to the chassis adds another 10 Gbps (5 Gbps Internet mix (IMIX)), 2 million sessions, and 100,000 connections per second (CPS) up to a maximum rated capacity for the platform of 150 Gbps (47.5 Gbps IMIX), 10 million sessions, and 350,000 CPS (as measured in Junos OS Release 10.2).

Orchestration and Automation

The Juniper Networks VDC design uses the Juniper Networks Junos Space management platform. Junos Space includes a portfolio of applications for scaling services, simplifying network operations, and automating support for complex network environments.

In addition, the network devices are configured to support background Secure Copy Protocol (SCP) file transfers, commit scripts, and a file archive site.

Related Documentation

- [Requirements of a Two-Tiered Virtualized Data Center for Large Enterprise Networks on page 1](#)
- [Example: Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks on page 5](#)

Example: Configuring a Two-Tiered Virtualized Data Center for Large Enterprise Networks

This example provides a step-by-step procedure for configuring a two-tiered virtualized data center for large enterprise networks.

- [Requirements on page 5](#)
- [Configuring a Two-Tiered Virtualized Data Center Overview on page 5](#)
- [Configuring the Access Layer on page 8](#)
- [Configuring the Aggregation Layer in the Trusted Logical Systems on page 12](#)
- [Configuring the Core Layer in the Untrusted Logical Systems on page 20](#)
- [Configuring the Security Device on page 25](#)

Requirements

This example uses the following hardware and software components:

- Two MX Series 3D Universal Edge Routers running Junos OS Release 10.2 or later
- Six EX Series Ethernet Switches running Junos OS Release 10.2 or later
- Two SRX Series Services Gateways running Junos OS Release 10.4 or later

Configuring a Two-Tiered Virtualized Data Center Overview

This example provides a step-by-step procedure for configuring a two-tiered virtualized data center for large enterprises. The steps in the example follow the data path from an interface connected to a server in BU2 using VLAN 17, to Logical System Trust1, through Virtual Router MX-VR2, through Virtual Router SRX-VR2, through VRF2 in the Logical System Untrust, and out to the core network.

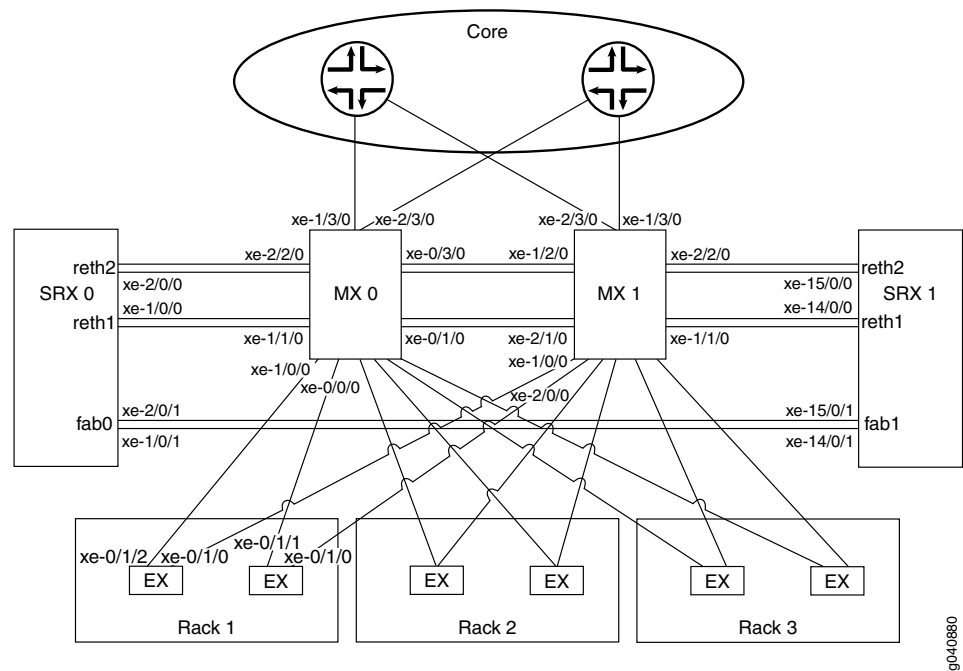
The core network in this example simultaneously supports IP-based routing and MPLS-based label switching. The virtual routers on the SRX Series device perform the functions of customer edge (CE) routers. The VPN routing and forwarding (VRF) routing instances on the MX Series devices perform the functions of service provider edge (PE) routers. The OSPF protocol serves as the interior gateway protocol to carry routes to the PE router loopback addresses that are used as the BGP next-hop address for the IP-based and MPLS-based networks supported by this example.



NOTE: The steps in this example are representative of the entire network configuration. The example does not show every step for every virtual device.

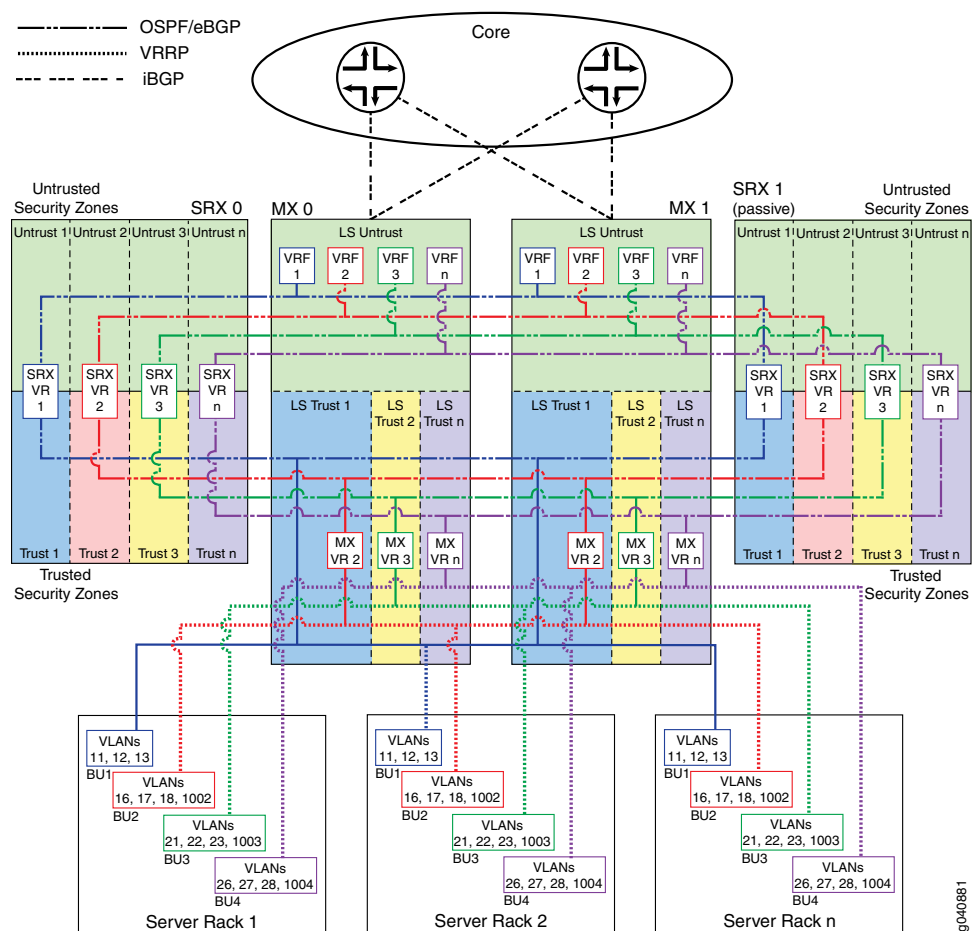
The physical connections used in this example are shown in [Figure 1 on page 6](#).

Figure 1: Virtualized Data Center Physical Topology



The logical connections used in this example are shown in [Figure 2 on page 7](#).

Figure 2: Virtualized Data Center Logical Topology



In the logical topology illustration:

- Users access the data center across the enterprise core network shown at the top.
- Virtual routers configured in Logical System Untrust on the MX Series devices forward the traffic to separate virtual routers configured in the Untrusted security zone on the SRX Series devices. These virtual routers act as edge routers for the various business units.
- Virtual routers configured on the active SRX Series device forward the traffic to the Trusted security zones.
- Virtual routers configured in separate logical systems on the MX Series devices forward the traffic to a bridge domain of VLANs configured on the EX Series devices.
- Business unit 1 requires additional separation. In this case, the virtual router (VR) configured on the SRX Series device forwards the traffic directly to the bridge domain on the EX Series devices.
- The EX Series devices switch the traffic to the data center server.

- The SRX Series devices apply security policy to all traffic traversing the untrust to trust boundary and all traffic forwarded between logical systems.
- The SRX Series devices are configured in an active/passive cluster so that only one node in the cluster is active on the data forwarding plane at a time.
- The SRX Series devices are configured with a single redundancy group for the data plane. The redundancy group uses two Ethernet interfaces (**reth1** and **reth2** in [Figure 1 on page 6](#)) as member interfaces.

Configuring the Access Layer

Configure the access layer by doing the following:

- [Configuring Interfaces on page 8](#)
- [Configuring VLANs in the Access Layer on page 9](#)
- [Configuring a Redundant Trunk Group and Disabling the Spanning Tree Protocol for the Trunk Interfaces on page 10](#)
- [Configuring Management Automation on page 11](#)

Configuring Interfaces

Step-by-Step Procedure

This procedure explains how to configure the physical, logical, and network management interfaces for the access layer devices. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Configure the access layer server-facing 10-Gigabit Ethernet interfaces.

This example configures the **ge-0/0/17** interface with VLAN ID 17.

Include the **member** statement and specify VLAN ID 17 at the **[edit interfaces ge-0/0/17 unit 0 family ethernet-switching vlan]** hierarchy level.

```
[edit interfaces ge-0/0/17 unit 0]
user@ex# set family ethernet-switching vlan members 17
```

Repeat this step for every server-facing interface by using the appropriate interface name and VLAN number.

2. Configure the 10-Gigabit Ethernet trunk interfaces from the EX Series device to the two MX Series devices.

This example configures the **xe-0/1/2** and **xe-0/1/0** interfaces.

Include the **port-mode** statement and specify the **trunk** option at the **[edit interfaces xe-0/1/2 unit 0 family ethernet-switching]** and **[edit interfaces xe-0/1/0 unit 0 family ethernet-switching]** hierarchy levels.

Include the **members** statement and specify the **all** option at the **[edit interfaces xe-0/1/2 unit 0 family ethernet-switching vlan]** and **[edit interfaces xe-0/1/0 unit 0 family ethernet-switching]** hierarchy levels.

```
[edit interfaces xe-0/1/2 unit 0]
user@ex# set family ethernet-switching port-mode trunk
user@ex# set family ethernet-switching vlan members all
```

```
[edit interfaces xe-0/1/0 unit 0]
user@ex# set family ethernet-switching port-mode trunk
user@ex# set family ethernet-switching vlan members all
```

Repeat this step for every 10-Gigabit Ethernet trunk interface by using the appropriate interface name.

3. Enable the IPv4 address family for the loopback logical interface.

Include the **family** statement and specify the **inet** option to enable IPv4 at the **[edit interfaces lo0 unit 0]** hierarchy level.

```
[edit interfaces lo0 unit 0]
user@ex# set family inet
```

Repeat this step for every EX Series device by using the appropriate address for that device.

4. Configure the EX Series device management Ethernet interface.

This example configures the **unit 0** logical interface.

Include the **family** statement and specify the **inet** option at the **[edit me0 unit 0]** hierarchy level.

Include the **address** statement and specify **10.8.108.19/24** as the IPv4 address at the **[edit interfaces me0 unit 0 family inet]** hierarchy level.

```
[edit interfaces me0 unit 0]
user@ex# set family inet address 10.8.108.19/24
```

Repeat this step for every EX Series device by using the appropriate management interface address for that device.

Configuring VLANs in the Access Layer

Step-by-Step Procedure

This procedure explains how to configure the VLAN names and tag IDs and associate trunk interfaces with one of the access layer devices. This procedure shows a representative sample of the configuration. The example does not show the configuration for every VLAN.

1. Configure the VLAN name and tag ID (number) for each VLAN on the EX Series device.

This example configures a VLAN with the name **vlan17** and tag ID 17.

Include the **vlan-id** statement and specify **17** as the VLAN tag ID at the **[edit vlans vlan17]** hierarchy level.

```
[edit vlans vlan17]
user@ex# set vlan-id 17
```

Repeat this step for every VLAN on each EX Series device by using the appropriate VLAN names and tag IDs.

2. Associate the logical trunk interfaces with each VLAN on the EX Series device.

This example associates logical interfaces **xe-0/1/0.0** and **xe-0/1/2.0** with **vlan17**.

Include the **interface** statement and specify **xe-0/1/0.0** at the **[edit vlans vlan17]** hierarchy level.

Include the **interface** statement and specify **xe-0/1/2.0** at the **[edit vlans vlan17]** hierarchy level.

```
[edit vlans vlan17]
user@ex# set interface xe-0/1/0.0
user@ex# set interface xe-0/1/2.0
```

Repeat this step for every VLAN on each EX Series device by using the appropriate trunk interface names.

Configuring a Redundant Trunk Group and Disabling the Spanning Tree Protocol for the Trunk Interfaces

Step-by-Step Procedure This procedure explains how to configure a redundant trunk group and disable the Rapid Spanning Tree Protocol (RSTP) on the trunk interfaces.

1. Configure the trunk interfaces as a redundant trunk group.

This example configures the **xe-0/1/0.0** and **xe-0/1/2.0** trunk interfaces in a redundant trunk group named **rtgroup1**.

Include the **interface** statement at the **[edit ethernet-switching-options redundant-trunk-group group rtgroup1]** hierarchy level and specify each trunk interface name.

Include the **primary** statement at the **[edit ethernet-switching-options redundant-trunk-group group rtgroup1 xe-0/1/2.0]** hierarchy level.

```
[edit ethernet-switching-options redundant-trunk-group group rtgroup1]
user@ex# set interface xe-0/1/0.0
user@ex# set interface xe-0/1/2.0 primary
```

Repeat this step for every redundant trunk group by using the appropriate interface names.

2. Disable RSTP on the trunk interfaces.

On an EX Series device, RSTP is enabled by default. RSTP cannot be enabled on the same interface as routing.

This example disables RSTP on the **xe-0/1/0.0** and **xe-0/1/2.0** trunk interfaces.

Include the **disable** statement at the **[edit protocols rstp interface xe-0/1/0.0]** and **[edit protocols rstp interface xe-0/1/2.0]** hierarchy levels.

```
[edit protocols rstp]
user@ex# set interface xe-0/1/0.0 disable
user@ex# set interface xe-0/1/2.0 disable
```

Repeat this step for every core-facing trunk interface by using the appropriate interface name.

Configuring Management Automation

Step-by-Step Procedure This procedure explains how to configure static routes to the management network, a known host to support background Secure Copy Protocol (SCP) file transfers, a commit script, and an event archive site.

1. Configure static routes so the Ethernet management interface can reach the management network.

Include the **route** statement, and specify **10.8.0.0/16** as the IPv4 subnet address of the management network at the **[edit routing-options static]** hierarchy level.

Include the **next-hop** statement, and specify the IPv4 host address of the next-hop router at the **[edit routing-options static route 10.8.0.0/16]** hierarchy level.

```
[edit routing-options static]
user@ex# set route 10.8.0.0/16 next-hop 10.8.108.254
```

Repeat this step for every Ethernet management interface on the EX Series devices.

2. Configure an SSH known host.

Include the **host** statement, and specify the IPv4 address and RSA host key options for trusted servers at the **[edit security ssh-known-hosts]** hierarchy level. In this example, the RSA host key is truncated to make it easier to read.

```
[edit security ssh-known-hosts]
user@ex# set host 127.0.0.1 rsa-key AAAAB3NzaC1yc2
```

Repeat this step for every EX Series device.

3. Configure outbound SSH to support Juniper Message Bundle (JMB) transfers to Juniper Support Systems (JSS).

In this example, the client ID is configured as **00187D0B670D**.

Include the **client** statement, specify **00187D0B670D** as the client ID, and specify **10.8.7.32** as the IPv4 address at the **[edit system services outbound-ssh]** hierarchy level.

Include the **port** statement and specify **7804** as the TCP port at the **[edit system services outbound-ssh client 00187D0B670D 10.8.7.32]** hierarchy level.

Include the **device-id** statement and specify **FA022D** as the device ID at the **[edit system services outbound-ssh client 00187D0B670D]** hierarchy level.

Include the **secret** statement at the **[edit system services outbound-ssh client 00187D0B670D]** hierarchy level.

Include the **services** statement and specify **netconf** as the available service at the **[edit system services outbound-ssh client 00187D0B670D]** hierarchy level.

```
[edit system services outbound-ssh client 00187D0B670D]
user@ex# set 10.8.7.32 port 7804
user@ex# set device-id FA022D
user@ex# set secret "$9$-9w4aik.QznDj9A0BEhrlKMxN"
user@ex# set services netconf
```

Repeat this step for every EX Series device.

4. Configure a commit script.

In this example the script file name is **jais-activate-scripts.slax**.

Include the **allow-transients** statement at the **[edit system scripts commit]** hierarchy level.

Include the **optional** statement at the **[edit system scripts commit file jais-activate-scripts.slax]** hierarchy level.

```
[edit system scripts commit]
user@ex# set allow-transients
user@ex# set file jais-activate-scripts.slax optional
```

Repeat this step for every EX Series devices.

5. Configure an event archive site.

In this example, the archive URL is the local **/var/tmp/** directory, and the name given to the destination is **juniper-aim**.

Include the **archive-sites** statement and specify the archive URL at the **[edit event-options destinations juniper-aim]** hierarchy level.

```
[edit event-options destinations juniper-aim]
user@ex# set archive-sites "scp://admin@127.0.0.1:/var/tmp" password "12345"
```

Repeat this step for every EX Series device.

Configuring the Aggregation Layer in the Trusted Logical Systems

Configure the aggregation layer by doing the following:

- [Configuring Interfaces in the Trusted Logical Systems on page 12](#)
- [Configuring VLANs in the Aggregation Layer on page 15](#)
- [Configuring the Virtual Router Routing Instance on page 16](#)
- [Configuring Management Interfaces on page 18](#)
- [Configuring Logical System Administrator Accounts on page 19](#)
- [Configuring Management Automation on page 19](#)

Configuring Interfaces in the Trusted Logical Systems

Step-by-Step Procedure

This procedure explains how to configure the physical, logical, and Layer 3 routing interfaces for the logical system in the trusted security zone of the aggregation layer. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Enable flexible VLAN tagging on the physical interfaces.

This example configures physical interface **xe-1/0/0**.

Include the **encapsulation** statement and specify the **flexible-ethernet-services** option at the **[edit interfaces xe-1/0/0]** hierarchy level.

Include the **flexible-vlan-tagging** statement at the **[edit interfaces xe-1/0/0]** hierarchy level.

```
[edit interfaces xe-1/0/0]
user@mx# set encapsulation flexible-ethernet-services
user@mx# set flexible-vlan-tagging
```

Repeat this step for every physical interface connected to the EX series, SRX Series, and MX Series devices using the appropriate interface name.

2. Configure the 10-Gigabit Ethernet interfaces connected to the EX Series access layer device.

This example configures logical interface 17 on the **xe-1/0/0** interface under the logical system named **Trust1**.

Include the **encapsulation** statement and specify the **vlan-bridge** option at the **[edit logical-systems Trust1 interfaces xe-1/0/0 unit 17]** hierarchy level.

Include the **vlan-id** statement and specify 17 as the VLAN ID at the **[edit logical-systems Trust1 interfaces xe-1/0/0 unit 17]** hierarchy level.

```
[edit logical-systems Trust1 interfaces xe-1/0/0 unit 17]
user@mx# set encapsulation vlan-bridge
user@mx# set vlan-id 17
```

Repeat this step for every interface connected to the access layer devices by using the appropriate interface name, logical interface number, VLAN ID, and logical system name.

3. Configure the 10-Gigabit Ethernet interfaces connected to the other MX Series device shown in [Figure 1 on page 6](#).

This example configures logical interface 17 on the **xe-0/1/0** interface.

Include the **encapsulation** statement and specify the **vlan-bridge** option at the **[edit logical-systems Trust1 interfaces xe-0/1/0 unit 17]** hierarchy level.

Include the **vlan-id** statement and specify 17 as the VLAN tag ID at the **[edit logical-systems Trust1 interfaces xe-0/1/0 unit 17]** hierarchy level.

```
[edit logical-systems Trust1 interfaces xe-0/1/0 unit 17]
user@mx# set encapsulation vlan-bridge
user@mx# set vlan-id 17
```

Repeat this step for every interface connected to the other MX Series device shown in [Figure 1 on page 6](#) by using the appropriate interface name, logical interface number, VLAN ID, and logical system name.

4. Configure the 10-Gigabit Ethernet interface connected to the SRX Series device.

This example configures logical interface 15 on the **xe-1/1/0** interface. Include the **encapsulation** statement and specify the **vlan-bridge** option at the **[edit logical-systems Trust1 interfaces xe-1/1/0 unit 15]** hierarchy level.

Include the **vlan-id** statement and specify 15 as the VLAN tag ID at the **[edit logical-systems Trust1 interfaces xe-1/1/0 unit 15]** hierarchy level.

```
[edit logical-systems Trust1 interfaces xe-1/1/0 unit 15]
user@mx# set encapsulation vlan-bridge
user@mx# set vlan-id 15
```

Repeat this step for every interface connected to the SRX Series device by using the appropriate interface name, logical interface number, VLAN ID, and logical system name.

5. Configure the Layer 3 integrated routing and bridging (IRB) interface address.

This example configures the **unit 17** logical interface with **10.17.2.2/24** as the IPv4 address under the logical system named **Trust1**. Include the **address** statement and specify **10.17.2.2/24** as the IPv4 address at the **[edit logical-systems Trust1 interfaces irb unit 17 family inet]** hierarchy level.

```
[edit logical-systems Trust1 interfaces irb unit 17 family inet]
user@mx# set address 10.17.2.2/24
```

Repeat this step for every Layer 3 IBR by using the appropriate logical interface name and IPv4 address.

6. Configure the IRB interface to participate in Virtual Router Redundancy Protocol (VRRP).

This example configures the **unit 17** logical interface with **17** as the VRRP group name.

Include the **virtual-address** statement and specify **10.17.2.1** as the IPv4 address of the virtual router at the **[edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24 vrrp-group 17]** hierarchy level.

Include the **accept-data** statement at the **[edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24 vrrp-group 17]** hierarchy level so the interface will accept packets destined for the virtual IP address.

Include the **priority** statement and specify **200** as the router's priority at the **[edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24 vrrp-group 17]** hierarchy level.

Include the **fast-interval** statement and specify **200** as the interval between VRRP advertisements at the **[edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24 vrrp-group 17]** hierarchy level.

Include the **preempt** statement at the **[edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24 vrrp-group 17]** hierarchy level.

```
[edit logical-systems Trust1 interfaces irb unit 17 family inet address 10.17.2.2/24
vrrp-group 17]
user@mx# set virtual-address 10.17.2.1
user@mx# set accept-data
user@mx# set priority 200
user@mx# set fast-interval 200
user@mx# set preempt
```

Repeat this step for every Layer 3 IBR interface by using the appropriate logical interface name, IPv4 address, VRRP group name, and priority.

Configuring VLANs in the Aggregation Layer

Step-by-Step Procedure

This procedure explains how to configure the VLAN names and tag IDs and associate trunk interfaces and Layer 3 routing interfaces with each VLAN. This procedure shows a representative sample of the configuration. The example does not show the configuration for every VLAN.

1. Configure the VLAN name and tag ID (number) for each VLAN on the MX Series device.

This example configures a VLAN with the name **vlan17** and tag ID **17** in the Logical System **Trust1**. Include the **vlan-id** statement and specify **17** as the VLAN ID at the **[edit logical-systems Trust1 bridge-domains vlan17]** hierarchy level.

```
[edit logical-systems Trust1 bridge-domains vlan17]
user@mx# set vlan-id 17
```

Repeat this step for every VLAN on each MX Series device by using the appropriate VLAN names and tag IDs.

2. Associate the logical trunk interfaces with each VLAN on the MX Series device.

This example associates logical interface **xe-1/0/0.17** that is connected to the EX Series device and logical interface **xe-0/1/0.17** that is connected to the other MX Series device with **vlan17**.

Include the **interface** statement and specify **xe-1/0/0.17** at the **[edit logical-systems Trust1 bridge-domains vlan17]** hierarchy level.

Include the **interface** statement and specify **xe-0/1/0.17** at the **[edit logical-systems Trust1 bridge-domains vlan17]** hierarchy level.

```
[edit logical-systems Trust1 bridge-domains vlan17]
user@mx# set interface xe-1/0/0.17
user@mx# set interface xe-0/1/0.17
```

Repeat this step for every server-facing VLAN on each MX Series device by using the appropriate trunk interface names.

3. Associate a Layer 3 interface with each VLAN on the MX Series device.

This example associates the **irb.17** interface with **vlan17**.

Include the **routing-interface** statement and specify **irb.17** at the **[edit logical-systems Trust1 bridge-domains vlan17]** hierarchy level.

```
[edit logical-systems Trust1 bridge-domains vlan17]
user@mx# set routing-interface irb.17
```

Repeat this step for every server-facing VLAN on each MX Series device by using the appropriate Layer 3 interface name.

4. Associate the logical interfaces with each interconnection VLAN on the MX Series device.

This example associates logical interface **xe-1/1/0.15** that is connected to the SRX Series device and logical interface **xe-0/1/0.15** that is connected to the other MX Series device with **vlan15**.

Include the **interface** statement and specify **xe-1/1/0.15** at the **[edit logical-systems Trust1 bridge-domains vlan15]** hierarchy level.

Include the **interface** statement and specify **xe-0/1/0.15** at the **[edit logical-systems Trust1 bridge-domains vlan15]** hierarchy level.

```
[edit logical-systems Trust1 bridge-domains vlan15]
user@mx# set interface xe-1/1/0.15
user@mx# set interface xe-0/1/0.15
```

Repeat this step for every interconnect VLAN on each MX Series device by using the appropriate interconnect interface names.

5. Associate a Layer 3 interface with each interconnection VLAN on the MX Series device to support active participation in the OSPF protocol.

This example associates the **irb.15** interface with **vlan15**.

Include the **routing-interface** statement and specify **irb.15** at the **[edit logical-systems Trust1 bridge-domains vlan15]** hierarchy level.

```
[edit logical-systems Trust1 bridge-domains vlan15]
user@mx# set routing-interface irb.15
```

Repeat this step for every server-facing VLAN on each MX Series device by using the appropriate Layer 3 interface name.

Configuring the Virtual Router Routing Instance

Step-by-Step Procedure

This procedure explains how to configure a single virtual router routing instance. This procedure shows a representative sample of the example configuration. The example does not show the configuration for every device.

1. Configure the routing instance type.

This example configures the routing instance with the name **MX-VR2**. Include the **instance-type** statement and specify **virtual-router** as the type at the **[edit logical-systems Trust1 routing-instances MX-VR2]** hierarchy level.

```
[edit logical-systems Trust1 routing-instances MX-VR2]
user@mx# set instance-type virtual-router
```

Repeat this step for every virtual router in each MX Series device by using the appropriate virtual router name.

2. Add the IRB interfaces used by the virtual router routing instance.

Include the **interface** statement and specify the name of each IRB interface at the **[edit routing-instances MX-VR2]** hierarchy level.

```
[edit logical-systems Trust1 routing-instances MX-VR2]
user@mx# set interface irb.15
user@mx# set interface irb.16
user@mx# set interface irb.17
```

```
user@mx# set interface irb.18
user@mx# set interface irb.1002
```

Repeat this step for every virtual router in each MX Series device by using the appropriate interface names.

3. Configure the IGP protocol active interface used by the virtual router routing instance so the routing tables can be populated with the routes to the servers.

This example configures one IRB interface to actively participate in the OSPF protocol area 0.0.0.0.

Include the **interface** statement and specify the name of the IRB interface at the **[edit logical-systems Trust1 routing-instances MX-VR2 protocols ospf area 0.0.0.0]** hierarchy level.

```
[edit routing-instances MX-VR2 protocols ospf area 0.0.0.0]
user@mx# set interface irb.15
```

Repeat this step for every virtual router in each MX Series device by using the appropriate virtual router name.

4. Configure the interior gateway protocol passive interfaces that are associated with each VLAN within the virtual router routing instance.

This example configures the IRB interfaces to passively participate in the OSPF protocol area 0.0.0.0.

Include the **passive** statement at the **[edit logical-systems Trust1 routing-instances MX-VR2 protocols ospf area 0.0.0.0 interface *irb-name*]** hierarchy level.

```
[edit logical-systems Trust1 routing-instances MX-VR2 protocols ospf area 0.0.0.0]
user@mx# set interface irb.16 passive
user@mx# set interface irb.17 passive
user@mx# set interface irb.18 passive
user@mx# set interface irb.1002 passive
```

Repeat this step for every virtual router in each MX Series device by using the appropriate virtual router name.

5. Configure the logical system router identifier.

Include the **router-id** statement and specify 10.200.11.101 as the router identifier at the **[edit logical-systems Trust1 routing-instances MX-VR2 routing-options]** hierarchy level.

```
[edit logical-systems Trust1 routing-instances MX-VR2 routing-options]
user@mx# set router-id 10.200.11.101
```

Repeat this step for every virtual router in each MX Series device by using the appropriate router identifier.

Configuring Management Interfaces

Step-by-Step Procedure This procedure explains how to configure static routes to the management network and the IPv4 address family for the loopback logical interface. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Configure static routes so the Ethernet management interface can reach the management network.

Include the **route** statement and specify **10.0.0.0/8** as the IPv4 subnet address of the management network at the **[edit routing-options static]** hierarchy level.

Include the **next-hop** statement, and specify the IPv4 host address of the next-hop router at the **[edit routing-options static route 10.0.0.0/8]** hierarchy level.

Include the **retain** and **no-readvertise** statements at the **[edit routing-options static route 10.0.0.0/8]** hierarchy level.

```
[edit routing-options static]
user@mx# set route 10.0.0.0/8 next-hop 10.8.3.254
user@mx# set route 10.0.0.0/8 retain
user@mx# set route 10.0.0.0/8 no-readvertise
```

Repeat this step for every MX Series device.

2. Configure the MX Series device management Ethernet interface. This example configures the **unit 0** logical interface.

Include the **family** statement and specify the **inet** option at the **[edit fxp0 unit 0]** hierarchy level.

Include the **address** statement and specify **10.8.3.212/24** as the IPv4 address at the **[edit interfaces fxp0 unit 0]** hierarchy level.

```
[edit interfaces fxp0 unit 0]
user@mx# set family inet address 10.8.3.212/24
```

Repeat this step for every MX Series device by using the appropriate management interface address for that device.

3. Configure the loopback logical interface.

Include the **family** statement and specify the **inet** option at the **[edit interfaces lo0 unit 0]** hierarchy level.

```
[edit interfaces lo0 unit 0]
user@mx# set family inet
```

Repeat this step for every MX Series device.

Configuring Logical System Administrator Accounts

Step-by-Step Procedure This procedure explains how to configure administrator account classes that are confined to the context of the logical system to which they are assigned and administrator accounts for each logical system.

1. Create administrator account classes.

In this example, the **trust1-admin** user class is created with **all** permissions for the **Trust1** logical system.

Include the **class** statement and specify **trust1-admin** as the class name at the **[edit system login]** hierarchy level.

Include the **logical-system** statement and specify **Trust1** as the logical system name at the **[edit system login class trust1-admin]** hierarchy level.

Include the **permissions** statement and specify the **all** option at the **[edit system login class trust1-admin]** hierarchy level.

```
[edit system]
user@mx# set login class trust1-admin logical-system Trust1
user@mx# set login class trust1-admin permissions all
```

Repeat this step for the **trust2-admin** and **untrust-admin** classes on each MX Series device by using the appropriate logical-system name.

2. Create administrator accounts that correspond to each logical system in the MX Series device.

In this example, the **trust1** user account is created and assigned the **trust1-admin** class.

Include the **class** statement and specify **trust1-admin** as the user class at the **[edit system login user trust1]** hierarchy level.

Include the **encrypted-password** statement and enter the encrypted password string at the **[edit system login user trust1 authentication]** hierarchy level.

```
[edit system]
user@mx# set login user trust1 class trust1-admin
user@mx# set login user trust1 authentication encrypted-password 12345
```

Repeat this step for the **trust2** and **untrust** user accounts on each MX Series device.

Configuring Management Automation

Step-by-Step Procedure This procedure explains how to configure a known host to support background SCP file transfers, a commit script, and an archive site.

1. Configure a commit script.

In this example, the script file name is **jais-activate-scripts.slax**.

Include the **allow-transients** statement at the **[edit system scripts commit]** hierarchy level.

Include the **optional** statement at the **[edit system scripts commit file jais-activate-scripts.slax]** hierarchy level.

```
[edit system scripts commit]
user@mx# set allow-transients
user@mx# set file jais-activate-scripts.slax optional
```

2. Configure an event archive site.

In this example the archive URL is the local **/var/tmp/** directory, and the name given to the destination is **juniper-aim**.

Include the **archive-sites** statement and specify the archive URL at the **[edit event-options destinations juniper-aim]** hierarchy level.

```
[edit event-options destinations juniper-aim]
user@mx# set archive-sites "scp://admin@127.0.0.1://var/tmp" password "12345"
```

Configuring the Core Layer in the Untrusted Logical Systems

Configure the core layer by doing the following:

- [Configuring Interfaces in the Untrusted Logical Systems on page 20](#)
- [Configuring VLANs in the Core Layer on page 22](#)
- [Configuring Protocols in the Untrusted Logical System on page 23](#)

Configuring Interfaces in the Untrusted Logical Systems

Step-by-Step Procedure

This procedure explains how to configure the physical, logical, and Layer 3 routing interfaces for the logical system in the untrusted security zone of the core layer. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Configure the 10-Gigabit redundant Ethernet interfaces connected to the other MX Series device shown in [Figure 1 on page 6](#).

This example configures logical interface **19** on the **xe-0/3/0** interface under the logical system named **Untrust** to participate in VLAN 19. Include the **encapsulation** statement and specify the **vlan-bridge** option at the **[edit logical-systems Untrust interfaces xe-0/3/0 unit 19]** hierarchy level.

Include the **vlan-id** statement and specify **19** as the VLAN tag ID at the **[edit logical-systems Untrust interfaces xe-0/3/0 unit 19]** hierarchy level.

```
[edit logical-systems Untrust interfaces xe-0/3/0 unit 19]
user@mx# set encapsulation vlan-bridge
user@mx# set vlan-id 19
```

Repeat this step for every redundant Ethernet interface connected to the other MX Series device by using the appropriate interface name, logical interface number, VLAN ID, and logical system name.

2. Configure the 10-Gigabit Ethernet interfaces connected to the SRX Series device.

This example configures logical interface **19** on the **xe-2/2/0** interface under the logical system named **Untrust** to participate in VLAN 19.

Include the **encapsulation** statement and specify the **vlan-bridge** option at the **[edit logical-systems Untrust interfaces xe-2/2/0 unit 19]** hierarchy level.

Include the **vlan-id** statement and specify **19** as the VLAN tag ID at the **[edit logical-systems Untrust interfaces xe-2/2/0 unit 19]** hierarchy level.

```
[edit logical-systems Untrust interfaces xe-2/2/0 unit 19]
user@mx# set encapsulation vlan-bridge
user@mx# set vlan-id 19
```

Repeat this step for every redundant Ethernet interface connected to the SRX Series device by using the appropriate interface name, logical interface number, VLAN ID, and logical system name.

3. Configure the 10-Gigabit Ethernet interfaces connected to the IP-based/MPLS-based core network.

This example configures logical interface **0** on the **xe-1/3/0** interface under the logical system named **Untrust**.

Include the **address** statement and specify **10.200.4.1/30** as the IPv4 address at the **[edit logical-systems Untrust interfaces xe-1/3/0 unit 0 family inet]** hierarchy level.

Include the **family** statement and specify the **mpls** option at the **[edit logical-systems Untrust interfaces xe-1/3/0 unit 0]** hierarchy level.

```
[edit logical-systems Untrust interfaces xe-1/3/0 unit 0]
user@mx# set family inet address 10.200.4.1/30
user@mx# set family mpls
```

Repeat this step for every 10-Gigabit Ethernet interface connected to the service provider network by using the appropriate interface name, logical interface number, IPv4 address, and logical system name.

4. Configure the Layer 3 IRB interface address.

This example configures the **unit 19** logical interface that participates in VLAN 19 with **10.19.2.1/24** as the IPv4 address under the logical system named **Untrust**.

Include the **address** statement and specify **10.19.2.1/24** as the IPv4 address at the **[edit logical-systems Untrust interfaces irb unit 19 family inet]** hierarchy level.

```
[edit logical-systems Untrust interfaces irb unit 19 family inet]
user@mx# set address 10.19.2.1/24
```

Repeat this step for every Layer 3 IRB interface by using the appropriate logical interface name and IPv4 address.

5. Configure an IP address for the loopback logical interface of the Logical System **Untrust**.

Include the **address** statement and specify **10.200.11.1/32** as the IPv4 address at the **[edit logical-systems Untrust interfaces lo0 unit 1 family inet]** hierarchy level.

```
[edit logical-systems Untrust interfaces lo0 unit 1 family inet]
user@mx# set address 10.200.11.1/32
```

Repeat this step for every MX Series device by using the appropriate IPv4 address.

Configuring VLANs in the Core Layer

Step-by-Step Procedure

This procedure explains how to configure the VLAN names and tag IDs and associate interfaces and Layer 3 routing interfaces with each core interconnect VLAN. This procedure shows a representative sample of the configuration. The example does not show the configuration for every VLAN.

1. Configure the VLAN name and tag ID (number) for each core interconnect VLAN on the MX Series device.

This example configures a VLAN with the name **vlan14** and tag ID **14** in the Logical System **Untrust**.

Include the **vlan-id** statement and specify **14** as the VLAN ID at the **[edit logical-systems Untrust bridge-domains vlan14]** hierarchy level.

```
[edit logical-systems Untrust bridge-domains vlan14]
user@mx# set vlan-id 14
```

Repeat this step for every VLAN on each MX Series device by using the appropriate VLAN names and tag IDs.

2. Associate the logical interfaces with each VLAN on the MX Series device.

This example associates logical interface **xe-0/3/0.14** that is connected to the other MX Series device and **xe-2/2/0.14** that is connected to the SRX Series device with **vlan14**.

Include the **interface** statement and specify **xe-0/3/0.14** at the **[edit logical-systems Untrust bridge-domains vlan14]** hierarchy level.

Include the **interface** statement and specify **xe-2/2/0.14** at the **[edit logical-systems Untrust bridge-domains vlan14]** hierarchy level.

```
[edit logical-systems Untrust bridge-domains vlan14]
user@mx# set interface xe-0/3/0.14
user@mx# set interface xe-2/2/0.14
```

Repeat this step for every core interconnect VLAN on each MX Series device by using the appropriate interface names.

3. Associate a Layer 3 interface with each VLAN on the MX Series device.

This example associates the **irb.14** interface with **vlan14**.

Include the **routing-interface** statement and specify **irb.14** at the **[edit logical-systems Untrust bridge-domains vlan14]** hierarchy level.

```
[edit logical-systems Untrust bridge-domains vlan14]
user@mx# set routing-interface irb.14
```

Repeat this step for every core interconnect VLAN on each MX Series device by using the appropriate Layer 3 interface name.

Configuring Protocols in the Untrusted Logical System

Step-by-Step Procedure This procedure explains how to configure the BGP, MPLS, RSVP, and OSPF protocols for the Logical System Untrust. This procedure shows a representative sample of the configuration. The example does not show the configuration for every device.

1. Add interfaces to the OSPF protocol on the MX Series device.

This example adds logical interfaces **xe-1/3/0.0** and **lo0.1** to the OSPF protocol used in the core network.

Include the **interface** statement and specify the **xe-1/3/0.0** and **lo0.1** interfaces at the **[edit logical-systems Untrust protocols ospf area 0.0.0.0]** hierarchy level.

```
[edit logical-systems Untrust protocols ospf area 0.0.0.0]
user@mx# set interface xe-1/3/0.0
user@mx# set interface lo0.1
```

Repeat this step for every 10-Gigabit Ethernet interface connected to the core layer devices by using the appropriate interface name.

2. Configure the Generic Router Encapsulation (GRE) tunnel.

This example enables a dynamic GRE tunnel named **GRE1**.

Include the **gre** statement to specify the tunnel type at the **[edit logical-systems Untrust routing-options dynamic-tunnel GRE1]** hierarchy level.

Include the **source-address** statement and specify **10.200.11.1** as the IPv4 source address at the **[edit logical-systems Untrust routing-options dynamic-tunnel GRE1]** hierarchy level.

Include the **destination-networks** statement and specify **0.0.0.0/0** as the destination prefix at the **[edit logical-systems Untrust routing-options dynamic-tunnel GRE1]** hierarchy level.

```
[edit logical-systems Untrust routing-options dynamic-tunnel GRE1]
user@mx# set source-address 10.200.11.1
user@mx# set gre
user@mx# set destination-networks 0.0.0.0/0
```

Repeat this step for each MX Series device by using the appropriate source address.

3. Configure the Logical System local autonomous system number and router identifier.

Include the **autonomous-system** statement and specify **65000** as the autonomous system number at the **[edit logical-systems Untrust routing-options]** hierarchy level.

Include the **router-id** statement and specify **10.200.11.101** as the router identifier at the **[edit logical-systems Untrust routing-options]** hierarchy level.

```
[edit logical-systems Untrust]
user@mx# set routing-options autonomous-system 65000
user@mx# set routing-options router-id 10.200.11.101
```

Repeat this step for each MX Series device by using the appropriate router identifier and autonomous system number 65000.

4. Configure the internal BGP peer group.

Include the **type** statement and specify the **internal** option at the **[edit logical-systems Untrust protocols bgp group int]** hierarchy level.

Include the **local-address** statement and specify the router ID (10.200.11.1) of Logical System Untrust as the local address at the **[edit logical-systems Untrust protocols bgp group int]** hierarchy level.

Include the **unicast** statement at the **[edit logical-systems Untrust protocols bgp group int family inet]** and **[edit logical-systems Untrust protocols bgp group int family inet-vpn]** hierarchy levels.

Include the **local-as** statement and specify **65000** as the local autonomous system number at the **[edit logical-systems Untrust protocols bgp group int]** hierarchy level.

Include the **peer-as** statement and specify **65000** as the peer autonomous system number at the **[edit logical-systems Untrust protocols bgp group int]** hierarchy level.

Include the **neighbor** statement and specify the neighbor IPv4 addresses at the **[edit logical-systems Untrust protocols bgp group int]** hierarchy level.

The neighbor addresses are the router ID addresses of the other MX Series device in the local data center, MX Series devices in a remote data center, and routers located in the IP-based/MPLS-based core network.

```
[edit logical-systems Untrust protocols bgp group int]
user@mx# set type internal
user@mx# set local-address 10.200.11.1
user@mx# set family inet unicast
user@mx# set family inet-vpn unicast
user@mx# set local-as 65000
user@mx# set peer-as 65000
user@mx# set neighbor 10.200.11.2
user@mx# set neighbor 10.200.11.3
user@mx# set neighbor 10.200.11.4
```

Repeat this step for every MX Series device.

5. Add interfaces to the MPLS protocol used in the service provider core network.

This example adds the **xe-1/3/0.0** and **xe-2/3/0.0** interfaces that are connected to the service provider core network.

Include the **interface** statement and specify the **xe-1/3/0.0** and **xe-2/3/0.0** interfaces at the **[edit logical-systems Untrust protocols mpls]** hierarchy level.

```
[edit logical-systems Untrust protocols mpls]
user@mx# set interface xe-1/3/0.0
user@mx# set interface xe-2/3/0.0
```

Repeat this step for every MX Series device.

6. Create an MPLS LSP to the router that is located in the MPLS-based core network.

This example creates an LSP named **to-core-router**.

Include the **to** statement and specify **10.200.11.3** as the IPv4 address of the core router at the **[edit logical-systems Untrust protocols mpls label-switched-path to-core-router]** hierarchy level.

Include the **no-cspf** statement at the **[edit logical-systems Untrust protocols mpls]** hierarchy level.

```
[edit logical-systems Untrust protocols mpls]
user@mx# set label-switched-path to-core-router to 10.200.11.3
user@mx# set no-cspf
```

Repeat this step for every MX Series device.

7. Add interfaces to the RSVP protocol used in the MPLS-based core network.

Include the **interface** statement and specify the **xe-1/3/0.0** and **xe-2/3/0.0** interfaces at the **[edit logical-systems Untrust protocols rsvp]** hierarchy level.

```
[edit logical-systems Untrust protocols rsvp]
user@mx# set interface xe-1/3/0.0
user@mx# set interface xe-2/3/0.0
```

Repeat this step for every MX Series device.

Configuring the Security Device

The following procedures explain how to configure the redundant Ethernet interfaces, node cluster, security zones, security policies, and routing policies for the trusted security zone of the access layer.

- [Configuring the Redundant Ethernet Interface Link Aggregation Group on page 25](#)
- [Configuring the SRX Series Cluster on page 26](#)
- [Creating Security Zones and Configuring the In-Bound Traffic Policy Action on page 28](#)
- [Configuring the Security Zone Policies on page 29](#)
- [Creating the Routing Policies on page 30](#)
- [Configuring the Virtual Router Routing Instance on page 33](#)

Configuring the Redundant Ethernet Interface Link Aggregation Group

Step-by-Step Procedure

This procedure explains how to configure the redundant Ethernet interface link aggregation group. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Configure the number of aggregated Ethernet interfaces supported on the node.

This example enables support for two interfaces.

Include the **device-count** statement and specify **2** as the number of interfaces supported at the **[edit chassis aggregated-devices ethernet]** hierarchy level.

```
[edit chassis aggregated-devices ethernet]
user@srx# set device-count 2
```

Repeat this step for every SRX Series device by using the appropriate device count.

2. Assign 10-Gigabit Ethernet child interfaces to the redundant Ethernet (reth) parent interface.

This example assigns the **xe-1/0/0** 10-Gigabit Ethernet child interface to the **reth1** parent interface on Node0.

Include the **redundant-parent** statement and specify **reth1** as the parent interface at the **[edit interfaces xe-1/0/0 gigether-options]** hierarchy level.

```
[edit interfaces xe-1/0/0 gigether-options]
user@srx# set redundant-parent reth1
```

Repeat this step for every redundant Ethernet interface by using the appropriate interface name and redundant parent name.

3. Configure the redundant Ethernet parent interface options.

This example configures the **reth1** redundant parent interface.

Include the **redundancy-group** statement and specify **1** as the group number at the **[edit interfaces reth1 redundant-ether-options]** hierarchy level.

Include the **vlan-tagging** statement at the **[edit interfaces reth1]** hierarchy level.

```
[edit interfaces reth1]
user@srx# set redundant-ether-options redundancy-group 1
user@srx# set vlan-tagging
```

Repeat this step for every redundant parent interface by using the appropriate redundant parent name and redundancy group number.

4. Configure the redundant Ethernet parent logical interfaces.

This example configures the **unit 15** logical interface.

Include the **address** statement and specify **10.15.2.2/24** as the IPv4 address at the **[edit interfaces reth1 unit 15 family inet]** hierarchy level.

Include the **vlan-id** statement and specify **15** as the VLAN identifier at the **[edit interfaces reth1 unit 15]** hierarchy level.

```
[edit interfaces reth1 unit 15]
user@srx# set family inet address 10.15.2.2/24
user@srx# set vlan-id 15
```

Repeat this step for every redundant parent interface by using the appropriate redundant parent name, IPv4 address, and VLAN identifier.

Configuring the SRX Series Cluster

Step-by-Step Procedure

This procedure explains how to configure fabric connections between the nodes in the cluster. This procedure shows a representative sample of the configuration. The example does not show the configuration for every interface.

1. Configure the 10-Gigabit Ethernet interface to serve as the fabric between the cluster nodes.

This example configures **xe-1/0/1** as the child fabric interface and **fab0** as the parent fabric interface. The connection is from SRX0 to SRX1.

Include the **member-interfaces** statement and specify the **xe-1/0/1** interface at the **[edit interfaces fab0 fabric-options]** hierarchy level.

```
[edit interfaces fab0 fabric-options]
user@srx# set member-interfaces xe-1/0/1
```

Repeat this step for every 10-Gigabit Ethernet interface that is part of the cluster fabric by using the appropriate child interface name and parent interface name.

2. Configure the number of redundant Ethernet interfaces that the cluster supports.

This example configures **4** as the number of interfaces.

Include the **reth-count** statement and specify **4** as the number of interfaces at the **[edit chassis cluster]** hierarchy level.

```
[edit chassis cluster]
user@srx# set reth-count 4
```

Repeat this step for every SRX Series device in the cluster.

3. Configure the node priority for the redundancy group to determine which node is primary and which is secondary.

This example configures **node 0** with a higher priority.

Include the **priority** statement and specify **200** at the **[edit chassis cluster redundancy-group 1 node 0]** hierarchy level.

Include the **priority** statement and specify **100** at the **[edit chassis cluster redundancy-group 1 node 1]** hierarchy level.

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

Repeat this step for every redundancy group on every SRX Series device in the cluster.

4. Allow a node with a higher priority to initiate a failover to become the primary node for the redundancy group.

Include the **preempt** statement at the **[edit chassis cluster redundancy-group 1]** hierarchy level.

```
[edit chassis cluster redundancy-group 1]
user@srx# set preempt
```

Repeat this step for every redundancy group on every SRX Series device in the cluster.

5. Enable control link recovery to be done automatically.

Include the **control-link-recovery** statement at the **[edit chassis cluster]** hierarchy level.

```
[edit chassis cluster]
user@srx# set control-link-recovery
```

Repeat this step for every redundancy group on every SRX Series device in the cluster.

6. Enable interface monitoring to monitor the health of the interfaces and trigger redundancy group failover.

This example configures the **xe-1/0/0** interface with a weight of **255**.

Include the **weight** statement at the **[edit chassis cluster redundancy-group 1 interface-monitor xe-1/0/0]** hierarchy level.

```
[edit chassis cluster redundancy-group 1 interface-monitor xe-1/0/0]
user@srx# set weight 255
```

Repeat this step for every redundancy group interface on every SRX Series device in the cluster.

Creating Security Zones and Configuring the In-Bound Traffic Policy Action

Step-by-Step Procedure

This procedure explains how to configure the trusted and untrusted security zones on the SRX Series device. This procedure shows a representative sample of the configuration. The example does not show the configuration for every zone.

1. Assign a redundant Ethernet logical interface to a trusted zones.

This example assigns the **reth1.15** interface to the **Trust2** zone.

Include the **interfaces** statement and specify **reth1.15** as the interface in the zone at the **[edit security zones security-zone Trust2]** hierarchy level.

```
[edit security zones security-zone Trust2]
user@srx# set interfaces reth1.15
```

Repeat this step for every trusted security zone by using the appropriate zone name and redundant Ethernet logical interface name.

2. Assign a redundant Ethernet logical interface to the untrusted zones.

This example assigns the **reth2.19** interface to the **Untrust2** zone.

Include the **interfaces** statement and specify **reth2.19** as the interface in the zone at the **[edit security zones security-zone Untrust2]** hierarchy level.

```
[edit security zones security-zone Untrust2]
user@srx# set interfaces reth2.19
```

Repeat this step for every untrusted security zone by using the appropriate zone name and redundant Ethernet logical interface name.

3. Enable all inbound system services traffic in the trusted security zone.

This example enables all services for the **Trust2** zone.

Include the **system-services** statement and specify the **all** option at the **[edit security zones security-zone Trust2 host-inbound-traffic]** hierarchy level.

```
[edit security zones security-zone Trust2 host-inbound-traffic]
user@srx# set system-services all
```

Repeat this step for every security zone on the SRX Series device where system services are allowed.

4. Enable all protocols for inbound traffic in the trusted security zone.

This example enables all protocols for the **Trust2** zone.

Include the **protocols** statement and specify the **all** option at the **[edit security zones security-zone Trust2 host-inbound-traffic]** hierarchy level.

```
[edit security zones security-zone Trust2 host-inbound-traffic]
user@srx# set protocols all
```

Repeat this step for every security zone on the SRX Series device where all protocols are allowed for inbound traffic.

Configuring the Security Zone Policies

Step-by-Step Procedure

This procedure explains how to configure the security zone policies on the SRX Series device. This procedure shows a representative sample of the configuration. The example does not show the configuration for every policy.

1. Define which zone traffic is coming from and which zone traffic is going to for the policy being created.

This example defines the from zone as **Trust2** and the to zone as **Untrust2**.

On a single command line, include the **from-zone** statement and specify **Trust2**, include the **to-zone** statement and specify **Untrust2**, include the **policy** statement and specify **denyftp** as the policy name, and included the **match** statement at the **[edit security policies]** hierarchy level.

```
[edit security policies]
user@srx# set from-zone Trust2 to-zone Untrust2 policy denyftp match
```

Repeat this step for every policy that controls traffic between zones.

2. Configure the policy match criteria for denying traffic.

This example matches the Junos OS FTP application from any source to any destination address in a policy named **denyftp**.

Include the **source-address** statement and specify **any** as the IPv4 address at the **[edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp match]** hierarchy level.

Include the **destination-address** statement and specify **any** as the IPv4 address at the **[edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp match]** hierarchy level.

Include the **application** statement and specify **junos-ftp** as the application at the **[edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp match]** hierarchy level.

```
[edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp match]
user@srx# set source-address any
user@srx# set destination-address any
user@srx# set application junos-ftp
```

Repeat this step for every protocol matching policy by using the correct protocol.

3. Block specific applications from passing from the Trust2 zone to the Untrust2 zone.

This example denies the Junos OS FTP application from the **Trust2** zone to the **Untrust2** zone.

Include the **deny** statement at the **[edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp then]** hierarchy level.

```
[edit security policies from-zone Trust2 to-zone Untrust2 policy denyftp then]
user@srx# set deny
```

Repeat this step for every deny policy.

4. Configure the policy match criteria for allowing traffic.

This example matches any application from any source to any destination address in a policy named **allow_all**.

Include the **source-address** statement and specify **any** as the IPv4 address at the **[edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all match]** hierarchy level.

Include the **destination-address** statement and specify **any** as the IPv4 address at the **[edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all match]** hierarchy level.

Include the **application** statement and specify **any** as the application at the **[edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all match]** hierarchy level.

```
[edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all match]
user@srx# set source-address any
user@srx# set destination-address any
user@srx# set application any
```

Repeat this step for every application matching policy.

5. Permit any application traffic to pass from the Trust2 zone to the Untrust2 zone.

This example allows any application traffic from the **Trust2** zone to the **Untrust2** zone.

Include the **permit** statement at the **[edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all then]** hierarchy level.

```
[edit security policies from-zone Trust2 to-zone Untrust2 policy allow_all then]
user@srx# set permit
```

Repeat this step for every permit policy.

Creating the Routing Policies

Step-by-Step Procedure

This procedure explains how to create the routing policies on the SRX Series device that can be applied to the appropriate routing instances. This procedure shows a representative sample of the configuration. The example does not show the configuration for every policy.

1. Create a policy to set the local preference for BGP routes to 120.

This example creates a policy named **local-pref-120** that sets the BGP local preference value for received routes advertised by BGP to **120**.

Include the **protocol** statement and specify **bgp** as the value at the **[edit policy-options policy-statement local-pref-120 term term1 from]** hierarchy level.

Include the **local-preference** statement and specify **120** as the value at the **[edit policy-options policy-statement local-pref-120 term term1 then]** hierarchy level.

```
[edit policy-options policy-statement local-pref-120]
user@srx# set term term1 from protocol bgp
user@srx# set term term1 then local-preference 120
```

Repeat this step for each SRX Series device.

2. Configure the match criteria for a policy named **default-ospf** to accept all aggregate (generated) routes.

Include the **protocol** statement and specify **aggregate** as the protocol to match at the **[edit policy-options policy-statement default-ospf term term1 from]** hierarchy level.

Include the **route-filter** statement and specify **0.0.0.0/0 exact** as the match criteria at the **[edit policy-options policy-statement default-ospf term term1 from]** hierarchy level.

```
[edit policy-options policy-statement default-ospf term term1 from]
user@srx# set protocol aggregate
user@srx# set route-filter 0.0.0.0/0 exact
```

Repeat this step for each SRX Series device.

3. Configure the action for a policy to set the metric to **0**, and set the external route type to **1**.

This example configures a policy named **default-ospf** that sets the metric to **0**, sets the external route to type **1**, and accepts aggregate routes into the routing table.

Include the **metric** statement and specify **0** as the external type at the **[edit policy-options policy-statement default-ospf term term1 then]** hierarchy level.

Include the **type** statement and specify **1** as the external route type at the **[edit policy-options policy-statement default-ospf term term1 then external]** hierarchy level.

Include the **accept** statement at the **[edit policy-options policy-statement default-ospf term term1 then]** hierarchy level.

```
[edit policy-options policy-statement default-ospf term term1 then]
user@srx# set metric 0
user@srx# set external type 1
user@srx# set accept
```

Repeat this step for each SRX Series device.

4. Create a policy that accepts OSPF routes with specified prefixes.

This example creates a policy named **trust2-ebgp-out** that accepts OSPF routes with the route prefixes that correspond to the subnets for each trust VLAN.

Include the **protocol** statement and specify **ospf** as the protocol at the **[edit policy-options policy-statement trust2-ebgp-out term term1 from]** hierarchy level.

Include the **route-filter** statement and specify the VLAN subnet addresses and the **exact** match keyword at the **[edit policy-options policy-statement trust2-ebgp-out term term1 from]** hierarchy level.

Include the **accept** statement at the **[edit policy-options policy-statement trust2-ebgp-out term term1 then]** hierarchy level.

```
[edit policy-options policy-statement trust2-ebgp-out term term1]
user@srx# set from protocol ospf
user@srx# set from route-filter 10.16.2.0/24 exact
user@srx# set from route-filter 10.17.2.0/24 exact
user@srx# set from route-filter 10.18.2.0/24 exact
user@srx# set then accept
```

Repeat this step for each SRX Series device.

5. Create a policy that accepts BGP routes if the route type is external.

This example creates a policy named **check-bgp-routes** that accepts BGP routes only if the route type is external.

Include the **protocol** statement and specify **bgp** as the protocol at the **[edit policy-options policy-statement check-bgp-routes term term1 from]** hierarchy level.

Include the **route-type** statement and specify the **external** option at the **[edit policy-options policy-statement check-bgp-routes term term1 from]** hierarchy level.

Include the **accept** statement at the **[edit policy-options policy-statement check-bgp-routes term term1 then]** hierarchy level.

```
[edit policy-options policy-statement check-bgp-routes term term1]
user@srx# set from protocol bgp
user@srx# set from route-type external
user@srx# set then accept
```

Repeat this step for each SRX Series device.

6. Create a policy that accepts routes from other virtual router routing instances.

This example creates a policy named **from_srx_vr1** that accepts routes from routing instance **SRX-VR1**.

Include the **instance** statement and specify **SRX-VR1** as the routing instance name at the **[edit policy-options policy-statement from_srx_vr1 term term1 from]** hierarchy level.

Include the **accept** statement at the **[edit policy-options policy-statement from_srx_vr1 term term1 then]** hierarchy level.

```
[edit policy-options policy-statement from_srx_vr1 term term1]
user@srx# set from instance SRX-VR1
user@srx# set then accept
```

Repeat this step for each virtual router in each SRX Series device.

Configuring the Virtual Router Routing Instance

Step-by-Step Procedure

This procedure explains how to configure a single virtual router routing instance. This procedure shows a representative sample of the example configuration. The example does not show the configuration for every virtual router routing instance.

1. Configure the routing instance type.

This example configures the routing instance with the name **SRX-VR2**.

Include the **instance-type** statement and specify **virtual-router** as the type at the **[edit routing-instances SRX-VR2]** hierarchy level.

```
[edit routing-instances SRX-VR2]
user@srx# set instance-type virtual-router
```

Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name.

2. Add the redundant Ethernet interfaces used by the virtual router routing instance.

This example adds **reth1.15** and **reth2.19** interfaces to the **SRX-VR2** routing instance.

Include the **interface** statement and specify the name of the redundant Ethernet interface at the **[edit routing-instances SRX-VR2]** hierarchy level.

```
[edit routing-instances SRX-VR2]
user@srx# set interface reth1.15
user@srx# set interface reth2.19
```

Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name and interface names.

3. Configure the routing options used by the virtual router routing instance.

This example configures the autonomous system number and enables the graceful restart feature on the **SRX-VR2** routing instance.

Include the **autonomous-system** statement and specify **65019** as the autonomous system number at the **[edit routing-instances SRX-VR2 routing-options]** hierarchy level.

Include the **graceful-restart** statement at the **[edit routing-instances SRX-VR2 routing-options]** hierarchy level.

```
[edit routing-instances SRX-VR2 routing-options]
user@srx# set autonomous-system 65019
user@srx# set graceful-restart
```

Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name and interface names.

4. Apply the routing policy that accepts external BGP routes and uses them as generated routes for the routing instance.

This example applies the policy named **check-bgp-routes** to the **SRX-VR2** routing instance.

Include the **policy** statement and specify **check-bgp-routes** at the **[edit routing-instances SRX-VR2 routing-options generate route 0.0.0.0/0]** hierarchy level.

Include the **graceful-restart** statement at the **[edit routing-instances SRX-VR2 routing-options]** hierarchy level.

```
[edit routing-instances SRX-VR2 routing-options]
user@srx# set generate route 0.0.0.0/0 policy
user@srx# set graceful-restart
```

Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name and interface names.

5. Apply the routing policy that accepts routes from other routing instances.

This example applies the policy named **from_srx_vr1** to the **SRX-VR2** routing instance.

Include the **instance-import** statement and specify **from_srx_vr1** at the **[edit routing-instances SRX-VR2 routing-options]** hierarchy level.

```
[edit routing-instances SRX-VR2 routing-options]
user@srx# set instance-import from_srx_vr1
```

Repeat this step for every virtual router in each SRX Series device except the **SRX-VR1** instance.

6. Configure the IGP protocol export policy used by the virtual router routing instance in the trusted security zone.

This example configures the **default-ospf** policy.

Include the **export** statement and specify **default-ospf** as the policy name at the **[edit routing-instances SRX-VR2 protocols ospf]** hierarchy level.

```
[edit routing-instances SRX-VR2 protocols ospf]
user@srx# set export default-ospf
```

Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name and policy name.

7. Configure the IGP protocol active and passive interfaces used by the virtual router routing instance in the trusted security zone.

This example configures the **reth1.15** redundant Ethernet interface to actively participate in the OSPF protocol area 0.0.0.0, and the **reth2.19** redundant Ethernet interface to passively participate.

Include the **interface** statement and specify **reth1.15** at the **[edit routing-instances SRX-VR2 protocols ospf area 0.0.0.0]** hierarchy level.

Include the **interface** statement and specify **reth2.19** at the **[edit routing-instances SRX-VR2 protocols ospf area 0.0.0.0]** hierarchy level.

Include the **passive** statement at the **[edit routing-instances SRX-VR2 protocols ospf area 0.0.0.0 reth2.19]** hierarchy level.

```
[edit routing-instances SRX-VR2 protocols ospf area 0.0.0.0]
user@srx# set interface reth1.15
user@srx# set interface reth2.19 passive
```

Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name and interface names.

8. Configure the BGP protocol peer groups used by the virtual router routing instance in the untrusted security zone.

Include the **type** statement and specify the **external** option at the **[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]** hierarchy level.

Include the **peer-as** statement and specify **65000** as the peer autonomous system number at the **[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]** hierarchy level.

Include the **neighbor** statement and specify **10.19.2.1** as the IPv4 neighbor address at the **[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]** hierarchy level. The neighbor address is the IRB Logical interface address of the VRF routing instance on the MX Series device.

```
[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]
user@srx# set type external
user@srx# set peer-as 65000
user@srx# set neighbor 10.19.2.1
```

Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name, instance type, neighbor address, and peer AS number.

9. Configure the BGP protocol peer groups export and import policies used by the virtual router routing instance in the untrusted security zone.

Include the **export** statement and specify **trust2-ebgp-out** as the export policy name at the **[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]** hierarchy level.

Include the **import** statement and specify **local-pref-120** as the import policy name at the **[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]** hierarchy level.

```
[edit routing-instances SRX-VR2 protocols bgp group MX0-vrf]
user@srx# set export trust2-ebgp-out
user@srx# set import local-pref-120
```

Repeat this step for every virtual router in each SRX Series device by using the appropriate virtual router name, export policy, and import policy.

Results The configuration steps of this example have been completed. The following section is for your reference.

The relevant sample configuration for the EX Series device follows.

```
EX Series Device  system {
                  scripts {
                    commit {
                      allow-transients;
                      file jais-activate-scripts.slax {
                        optional;
                      }
                    }
                  }
                }
```

```
services {
  ftp;
  ssh;
  telnet;
  outbound-ssh {
    client 00187D0B670D {
      device-id FA022D;
      secret "$9$-9w4aik.QznDj9A0BEhrlKMxN"; ## SECRET-DATA
      services netconf;
      10.8.7.32 port 7804;
    }
  }
}
}
interfaces {
  ge-0/0/17 {
    unit 0 {
      family ethernet-switching {
        vlan {
          members 17;
        }
      }
    }
  }
  xe-0/1/0 {
    unit 0 {
      family ethernet-switching {
        port-mode trunk;
        vlan {
          members all;
        }
      }
    }
  }
  xe-0/1/2 {
    unit 0 {
      enable;
      family ethernet-switching {
        port-mode trunk;
        vlan {
          members all;
        }
      }
    }
  }
  lo0 {
    unit 0 {
      family inet {
        address 127.0.0.1/32;
      }
    }
  }
  me0 {
    unit 0 {
      family inet {
        address 10.8.108.19/24;
      }
    }
  }
}
```



```

    }
  }
}
event-options {
  destinations {
    juniper-aim {
      archive-sites {
        "scp://admin@127.0.0.1:/var/tmp" password "$9$u3KWOEcrevL7-eKaZ"; ##
        SECRET-DATA
      }
    }
  }
}
routing-options {
  static {
    route 10.8.0.0/16 next-hop 10.8.108.254;
  }
}
protocols {
  rstp {
    interface xe-0/1/0.0 {
      disable;
    }
    interface xe-0/1/2.0 {
      disable;
    }
  }
}
security {
  ssh-known-hosts {
    host 127.0.0.1 {
      rsa-key AAAAB3NzaC1yc2;
    }
  }
}
ethernet-switching-options {
  redundant-trunk-group {
    group rtgroup1 {
      interface xe-0/1/0.0;
      interface xe-0/1/2.0 {
        primary;
      }
    }
  }
}
vllans {
  vlan17 {
    vlan-id 17;
    interface {
      xe-0/1/0.0;
      xe-0/1/2.0;
    }
  }
}
}

```

The relevant sample configuration for the MX Series device follows.

```
MX Series Device  groups {
                    re0 {
                      system {
                        host-name MX0;
                      }
                    }
                    re1 {
                      system {
                        host-name MX0re1;
                      }
                    }
                  }
  apply-groups [re0 re1];
  system {
    scripts {
      commit {
        allow-transients;
        file jais-activate-scripts.slax {
          optional;
        }
      }
    }
    login {
      class trust1-admin {
        logical-system Trust1;
        permissions all;
      }
      user trust1 {
        uid 2000;
        class trust1-admin;
        authentication {
          encrypted-password 12345; ## SECRET-DATA
        }
      }
    }
  }
  logical-systems {
    Trust1 {
      interfaces {
        xe-0/1/0 {
          unit 17 {
            encapsulation vlan-bridge;
            vlan-id 17;
          }
        }
        xe-1/0/0 {
          unit 17 {
            encapsulation vlan-bridge;
            vlan-id 17;
          }
        }
        xe-1/1/0 {
          unit 15 {
            encapsulation vlan-bridge;
          }
        }
      }
    }
  }
```

```

        vlan-id 15;
    }
}
irb {
    unit 15 {
        family inet {
            address 10.15.2.3/24;
        }
    }
    unit 17 {
        family inet {
            address 10.17.2.2/24 {
                vrrp-group 17 {
                    virtual-address 10.17.2.1;
                    priority 200;
                    fast-interval 200;
                    preempt;
                    accept-data;
                }
            }
        }
    }
}
}
routing-instances {
    MX-VR2 {
        instance-type virtual-router;
        interface irb.15;
        interface irb.16;
        interface irb.17;
        interface irb.18;
        interface irb.1002;
        protocols {
            ospf {
                area 0.0.0.0 {
                    interface irb.16 {
                        passive;
                    }
                    interface irb.17 {
                        passive;
                    }
                    interface irb.18 {
                        passive;
                    }
                    interface irb.1002 {
                        passive;
                    }
                }
            }
        }
    }
}
}
bridge-domains {
    vlan15 {
        vlan-id 15;
        interface xe-1/1/0.15;
    }
}

```

```
        interface xe-0/1/0.15; ## 'xe-0/1/0.15' is not defined
        routing-interface irb.15;
    }
    vlan17 {
        vlan-id 17;
        interface xe-1/0/0.17;
        interface xe-0/1/0.17;
        routing-interface irb.17;
    }
}
Untrust {
    interfaces {
        xe-0/3/0 {
            unit 19 {
                encapsulation vlan-bridge;
                vlan-id 19;
            }
        }
        xe-1/3/0 {
            unit 0 {
                family inet {
                    address 10.200.4.1/30;
                }
                family mpls;
            }
        }
        xe-2/2/0 {
            unit 19 {
                encapsulation vlan-bridge;
                vlan-id 19;
            }
        }
    }
    irb {
        unit 19 {
            family inet {
                address 10.19.2.1/24;
            }
        }
    }
    lo0 {
        unit 1 {
            family inet {
                address 10.200.11.1/32;
            }
        }
    }
}
protocols {
    RSVP {
        interface xe-1/3/0.0;
        interface xe-2/3/0.0;
    }
    mpls {
        no-cspf;
        label-switched-path to-core-router {
```

```

        to 10.200.11.3;
    }
    interface xe-1/3/0.0;
    interface xe-2/3/0.0;
}
bgp {
    group int {
        type internal;
        local-address 10.200.11.1;
        family inet {
            unicast;
        }
        family inet-vpn {
            unicast;
        }
        peer-as 65000;
        local-as 65000;
        neighbor 10.200.11.2;
        neighbor 10.200.11.3;
        neighbor 10.200.11.4;
    }
}
ospf {
    area 0.0.0.0 {
        interface xe-1/3/0.0;
        interface lo0.1;
    }
}
}
routing-options {
    router-id 10.200.11.101;
    autonomous-system 65000;
    dynamic-tunnels {
        GRE1 {
            source-address 10.200.11.1;
            gre;
            destination-networks {
                0.0.0.0/0;
            }
        }
    }
}
}
bridge-domains {
    vlan14 {
        vlan-id 14;
        interface xe-0/3/0.14;
        interface xe-2/2/0.14;
        routing-interface irb.14;
    }
}
}
}
interfaces {
    xe-0/1/0 {
        flexible-vlan-tagging;
        encapsulation flexible-ethernet-services;
    }
}

```

```
}
xe-0/3/0 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
}
xe-1/0/0 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
}
xe-1/1/0 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
}
xe-2/2/0 {
    flexible-vlan-tagging;
    encapsulation flexible-ethernet-services;
}
fxp0 {
    unit 0 {
        family inet {
            address 10.8.3.212/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet;
    }
}
}
event-options {
    destinations {
        juniper-aim {
            archive-sites {
                "scp://admin@127.0.0.1:/var/tmp" password "$9$DyimfQFnCpOF3re"; ##
                SECRET-DATA
            }
        }
    }
}
}
routing-options {
    static {
        route 10.0.244.8/30 next-hop 10.0.134.10;
        route 10.0.0.0/8 {
            next-hop 10.8.3.254;
            retain;
            no-readvertise;
        }
    }
}
}
```

The relevant sample configuration for the SRX Series device follows.

```
SRX Series Device  system {
                    host-name srx0;
                    chassis {
```

```

cluster {
    control-link-recovery;
    reth-count 4;
    redundancy-group 1 {
        node 0 priority 200;
        node 1 priority 100;
        preempt;
        interface-monitor {
            xe-1/0/0 weight 255;
        }
    }
}

interfaces {
    xe-1/0/0 {
        gigether-options {
            redundant-parent reth1;
        }
    }
    fab0 {
        fabric-options {
            member-interfaces {
                xe-1/0/1;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 127.0.0.1/32;
            }
        }
    }
    reth1 {
        vlan-tagging;
        redundant-ether-options {
            redundancy-group 1;
        }
        unit 15 {
            vlan-id 15;
        }
    }
}

policy-options {
    policy-statement check-bgp-routes {
        term term1 {
            from {
                protocol bgp;
                route-type external;
            }
            then accept;
        }
    }
    policy-statement default-ospf {
        term term1 {

```

```
        from {
            protocol aggregate;
            route-filter 0.0.0.0/0 exact;
        }
        then {
            metric 0;
            external {
                type 1;
            }
            accept;
        }
    }
}
policy-statement from_srx_vr1 {
    term term1 {
        from instance SRX-VR1;
        then accept;
    }
}
policy-statement local-pref-120 {
    term term1 {
        from protocol bgp;
        then {
            local-preference 120;
        }
    }
}
policy-statement trust2-ebgp-out {
    term term1 {
        from {
            protocol ospf;
            route-filter 10.16.2.0/24 exact;
            route-filter 10.17.2.0/24 exact;
            route-filter 10.18.2.0/24 exact;
        }
        then accept;
    }
}
}
security {
    policies {
        from-zone Trust2 to-zone Untrust2 {
            policy denyftp {
                match {
                    source-address any;
                    destination-address any;
                    application junos-ftp;
                }
                then {
                    deny;
                }
            }
            policy allow_all {
                match {
                    source-address any;
                    destination-address any;
                }
            }
        }
    }
}
```



```

        application any;
    }
    then {
        permit;
    }
}
}
}
zones {
    security-zone Trust2 {
        host-inbound-traffic {
            system-services {
                all;
            }
            protocols {
                all;
            }
        }
        interfaces {
            reth1.15;
        }
    }
    security-zone Untrust2 {
        interfaces reth2.19;
    }
}
}
routing-instances {
    SRX-VR2 {
        instance-type virtual-router;
        interface reth1.15;
        interface reth2.19;
        routing-options {
            graceful-restart;
            autonomous-system 65019;
            instance-import from_srx_vr1;
        }
        protocols {
            bgp {
                group MX0-vrf {
                    import local-pref-120;
                    export trust2-ebgp-out;
                }
            }
            ospf {
                export default-ospf;
                area 0.0.0.0 {
                    interface reth1.15;
                    interface reth2.19 {
                        passive;
                    }
                }
            }
        }
    }
}
}
}
}
}

```

**Related
Documentation**

- [Requirements of a Two-Tiered Virtualized Data Center for Large Enterprise Networks on page 1](#)
- [Two-Tiered Virtualized Data Center Solution for Large Enterprise Networks on page 2](#)