Engineering
Simplicity

Network Configuration Example

Integrating AWS Outposts with QFX Series Devices in an IP Fabric Data Center

Published
2023-10-16

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

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.

Network Configuration Example Integrating AWS Outposts with QFX Series Devices in an IP Fabric Data Center

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

YEAR 2000 NOTICE

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

END USER LICENSE AGREEMENT

Table of Contents

About This Guide

This network configuration example (NCE) shows how to configure a QFX Series device to interoperate with the Amazon Web Services (AWS) Outposts solution. AWS Outposts is a new fully managed service that brings AWS infrastructure, services, APIs, and tools on premise to support applications that run on premise. AWS compute, storage, database, and other services run locally on Outposts, and you can access the full range of AWS services available in the Region to build, manage, and scale your on-premise applications using familiar AWS services and tools.

1

CHAPTER

AWS Outposts in an IP Fabric Data Center

Integrating AWS Outposts with QFX Series Devices in an IP Fabric Data Center

IN THIS SECTION

About This Network Configuration Example

This network configuration example (NCE) shows how to configure a QFX Series device to interoperate with the Amazon Web Services (AWS) Outposts solution. The AWS Outposts solution is designed to bring AWS services (Elastic Compute Cloud, Elastic Block Storage, etc) to an enterprise's on-premise data center. The ability to locally host AWS resources in a QFX-based data center and have these services mix seamlessly with cloud-based AWS workloads opens up new opportunities for efficiency, resiliency, and performance.

Outpost racks (one or more) are delivered to a customer site. Outpost racks are fully assembled and preconfigured, requiring only power and network connectivity to begin providing service in the data center. This NCE shows how to configure a QFX-based data center fabric to support the AWS Outposts solution.

SEE ALSO

AWS Outposts

Use Case Overview

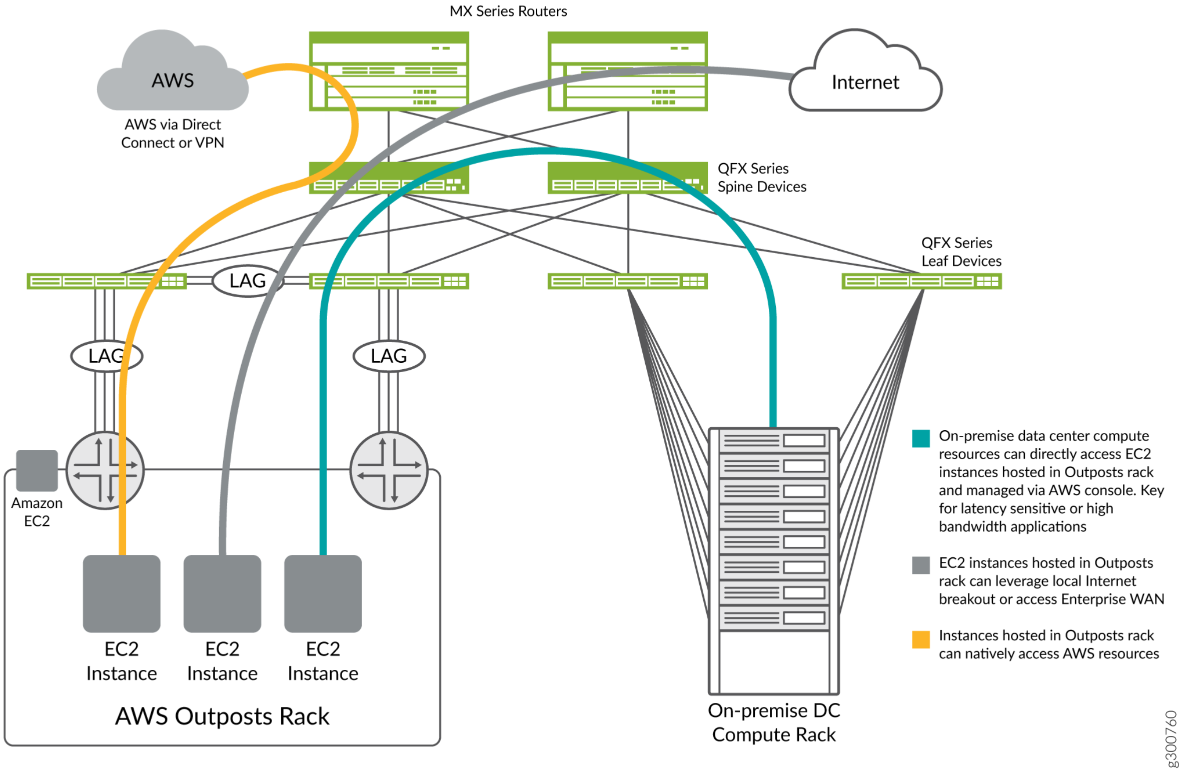
IN THIS SECTION

AWS Outposts in a QFX based Data Center

Enterprises are increasingly migrating business workloads to public clouds like Amazon Web Services. Hosting services in the cloud provides new options for scalability, resiliency, and cost optimization. However, there are cases where you may still need to host a workload in an on-premise data center. Application latency requirements, throughput requirements, and regulatory constraints are all reasons a workload may need to be hosted on-premise. AWS Outposts offers customers the same AWS hardware infrastructure, services, APIs, and tools to build and run their applications on-premise and in the cloud for a truly consistent hybrid experience. AWS compute, storage, database, and other services run locally on Outposts, and customers can access the full range of AWS services available in the Region to build, manage, and scale their on-premise applications using familiar AWS services and tools.

QFX Series switches from Juniper Networks are widely deployed in data centers for their speed, high quality, feature richness and extensive automation capabilities. A modern data center IP fabric typically leverages the QFX5000 Series devices in the leaf (or top-of-rack) role and either the QFX10000 or QFX5000 Series devices in the spine role. The flexibility of the QFX family and the Junos OS provides many architectural options for how a customer can build their data center fabric. This NCE focuses on a simple spine and leaf IP fabric design, which is well suited to accommodate the introduction of the AWS Outposts into the data center fabric.

Example use cases for AWS Outposts integration with IP fabric:



🟦 On-premise data center compute resources can directly access EC2 instances hosted in Outposts rack and managed via AWS console. Key for latency sensitive or high bandwidth applications

⬜ EC2 instances hosted in Outposts rack can leverage local Internet breakout or access Enterprise WAN

🟨 Instances hosted in Outposts rack can natively access AWS resources

Low Latency Access to AWS Workloads

With AWS Outposts integrated into the data center fabric, the workloads hosted in the Outposts solution become a full participant in the data center. This is important for applications where local non-AWS workloads need to interact with AWS workloads. This direct access via the data center fabric provides high bandwidth options (up to 400G) and lower latency paths for applications to interact with AWS resources. Resources in the data center will have direct IP connectivity to resources in the Outpost rack. Research applications requiring large amounts of data or financial applications requiring low latency are good examples where this direct access can provide a significant performance improvement.

On-premise Workloads Leveraging Local Internet Breakout

There are many types of workloads that require Internet connectivity to perform their function or to reach their enterprise users. When migrating these workloads to the cloud, enterprises have traditionally hosted these applications in the AWS cloud and leveraged AWS Internet gateways for access. The integration of AWS Outposts into an on-premise data center provides new and more flexible options for using local data center Internet connectivity for these applications. Doing so can provide faster speeds, lower cost, and different proximity options for Internet destined traffic.

Regulatory Requirements for Data Sovereignty

In various countries, regions, states, etc. there are regulatory limits on how and where data is stored. Enterprises operating in these environments can integrate the AWS Outposts solution into a data center that can meet these regulatory requirements. This allows enterprises to leverage the operational benefits of the AWS cloud, while staying compliant with local data sovereignty requirements.

Technical Overview

IN THIS SECTION

AWS Outposts in a QFX Based Data Center Fabric for Locally Hosted Workloads

This NCE covers the integration of a single AWS Outpost rack into a QFX based IP fabric data center.

The AWS Outposts solution integrates with existing data centers using a standard Layer 3 IP handoff. Consistent with other AWS connectivity models, there is no direct Layer 2 access to the Outpost resources. The Outpost rack uses two Outpost Network Devices integrated into the rack. As a customer, there is no access to the rack devices and all configuration of the Outpost is performed using the AWS Console.

The Outpost rack provides one uplink connection from each Outpost Network Device. These uplinks can be 1G, 10G, 40G, or 100G and can be organized into LAG bundles for increased throughput and resiliency. Each connection terminates into a dedicated QFX device, most likely a QFX5000 Series device acting as a fabric leaf.

The QFX family includes a number of platforms that can provide connectivity for the Outposts solution. This NCE assumes that we are using two QFX5120-48Ys for the handoff to the AWS Outpost. Unlike other data center connectivity models, we connect each Outpost Network Device to only one of the leaf devices. This results in each QFX5120-48Y being connected to one of the Outpost Network Devices. We also use a link between the two QFX devices to provide resilient connectivity to the Outpost rack in the event of device failure.

Table 1 on page 6 shows QFX devices that we recommend using to integrate with AWS Outposts.

Table 1: QFX Device Family Options for AWS Outposts

QFX Device	Attributes	AWS Outposts Recommendation
QFX5120-48Y	48 x 10/25G access ports with 8 100G fabric uplink ports	For general purpose data center fabric access. Provides flexibility for 10G or 25G server access for on-premise workloads and connections to AWS Outposts.
QFX5120-32C	32 x 100G access ports	For high speed data center fabrics, the QFX5120-32C provides 100G ports for AWS Outposts or server access.
QFX5110-48S	48 x 10G access ports with 4 100G uplink ports	For data center fabrics that have standardized on 10G, the QFX5110-48S provides 10G ports for AWS Outposts interconnect with 100G uplinks for the data center fabric.

When you order an AWS Outposts solution from AWS, you need to provide the following information about the on-premise data center that is used to build the initial configuration for the rack.

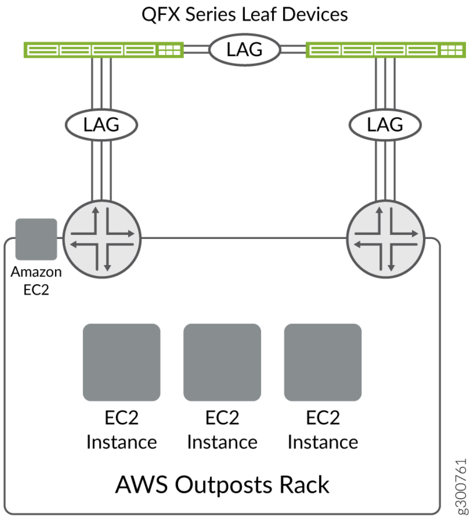
- Two /27 subnets for rack connectivity. One subnet is used for the control and management channel to the AWS Cloud. Both subnets are used by the Outposts infrastructure elements to connect back to the Region. The traffic destined to these subnets can be control, management or VPC data traffic.

  The subnets are configured on the Outpost prior to installation.

- Two BGP AS numbers (ASNs): one for the data center (likely an existing ASN) and a second (likely new) ASN for the Outpost, to provide peering between the data center fabric and the Outpost. These can be existing public ASNs or private ASNs. This example uses private ASNs.

  The routing between the QFX devices and the Outpost system is needed to exchange connectivity information with the rack. The Outpost needs reachability to the AWS cloud as well as local data center resources. The existing data center fabric is likely using an IGP, such as OSPF or ISIS, for current operations and the Outpost learns this routing information using the BGP peerings we establish with the Outpost. While you could use statics routes between the data center and the rack, we recommend that you use a dynamic routing protocol.

- Two subnets and two VLAN IDs are required for each of the links between the QFX devices and the Outpost. The VLANs are used to carry the two point-to-point routed subnets (either /30 or /31) on the same connection. Each of the links between the QFX and the Outpost will be an Ethernet trunk and will carry both subnets.



Example: Connecting an AWS Outpost to QFX5120-48Y Leaf Nodes

IN THIS SECTION

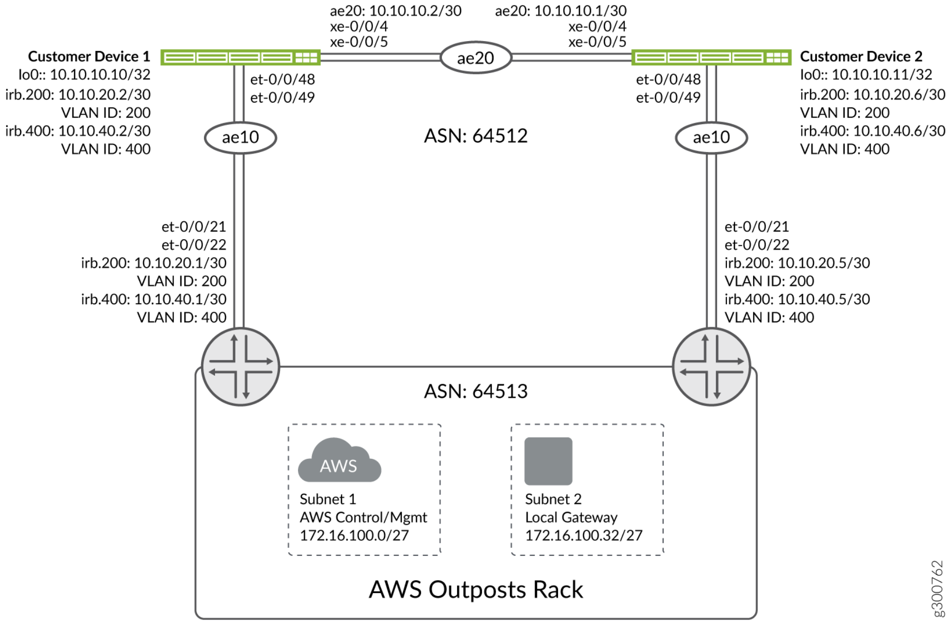
Requirements

This example uses the following:

- One operational data center IP fabric running OSPF
- One AWS Outpost rack
- Two QFX5120-48Y switches running Junos OS Release 18.4R2, acting as leaf devices

Overview

From the data center fabric there are no special requirements for the AWS Outposts solution. You can leverage fabric architectures like EVPN/VXLAN, IP fabric, or Virtual Chassis to integrate the Outpost into your network. For this example, we will use a simple IP fabric with OSPF as the IGP for internal route propagation. You should review your own fabric requirements and make required modifications to the steps below.



Configuration

IN THIS SECTION

Procedure

Step-by-Step Procedure

1. Before you begin to configure your devices for the Outpost connections, make sure your QFX devices have the following basic configuration.

   This configuration should be applied to both Customer Device 1 and Customer Device 2:

   ```
   set system host-name hostname
   set system services ssh root-login allow
   set system services netconf ssh
   ```

```
set protocols lldp interface all
set protocols lldp-med interface all
set chassis aggregated-devices ethernet device-count 10
set forwarding-options storm-control-profiles default all
```

2. The AWS Outpost supports 1/10/40/100G interface speeds and the rack expects these interfaces to be put into a LAG group. In this example, we are using a LAG bundle named `ae10` towards the Outpost.

   We also need a link between the two fabric devices to provide resiliency. This example uses `ae20` as the link between the two devices. In some environments, this interconnect may not be necessary because the IP fabric can provide alternate paths.

   Configuration for Customer Device 1:

```
set interfaces et-0/0/48 ether-options 802.3ad ae10
set interfaces et-0/0/49 ether-options 802.3ad ae10
set interfaces xe-0/0/4 ether-options 802.3ad ae20
set interfaces xe-0/0/5 ether-options 802.3ad ae20
set interfaces ae10 aggregated-ether-options lacp active
set interfaces ae10 aggregated-ether-options lacp periodic fast
set interfaces ae20 aggregated-ether-options lacp active
set interfaces ae20 aggregated-ether-options lacp periodic fast
```

   Configuration for Customer Device 2:

```
set interfaces et-0/0/48 ether-options 802.3ad ae10
set interfaces et-0/0/49 ether-options 802.3ad ae10
set interfaces xe-0/0/4 ether-options 802.3ad ae20
set interfaces xe-0/0/5 ether-options 802.3ad ae20
set interfaces ae10 aggregated-ether-options lacp active
set interfaces ae10 aggregated-ether-options lacp periodic fast
set interfaces ae20 aggregated-ether-options lacp active
set interfaces ae20 aggregated-ether-options lacp periodic fast
```

3. Configure two subnets to connect to the AWS cloud and the local data center. The subnets are carried between the QFX devices and the Outpost using VLAN IDs provided at time of provisioning. This example uses VLAN ID 200, named `AWS-LGW`, for the local data center connection, and VLAN ID 400, named `AWS-SERVICE-LINK`, for the connection to the AWS cloud. The Outpost expects both VLAN tags on the links between the data center edge and the Outpost Network Devices.

These VLANs are only used between the two devices to carry the two Outposts subnets. The VLANs are not extended into the data center. All traffic into the Outpost will use the Layer 3 interfaces. Customers may wish to use Virtual Routers to separate the AWS service link traffic from the datacenter fabric. This example provides an optional step (step 7) which covers the creation of a dedicated VRF to create this isolation.

This configuration should be applied to both Customer Device 1 and Customer Device 2:

```
set vlans AWS-LGW vlan-id 200
set vlans AWS-SERVICE-LINK vlan-id 400
set interfaces ae10 unit 0 family ethernet-switching vlan members [ AWS-LGW AWS-SERVICE-LINK ]
set interfaces ae10 unit 0 family ethernet-switching interface-mode trunk
```

4. Add Layer 3 addressing to the interfaces. These are the point-to-point links between the two devices.

   We also configure a loopback address on the QFX devices to establish an IBGP peering between them.

   Configuration for Customer Device 1:

   ```
   set interfaces irb unit 200 family inet address 10.10.20.2/30
   set interfaces irb unit 400 family inet address 10.10.40.2/30
   set interfaces lo0 unit 0 family inet address 10.10.10.10/32
   set interfaces ae20 unit 0 family inet address 10.10.10.2/30
   set vlans AWS-LGW l3-interface irb.200
   set vlans AWS-SERVICE-LINK l3-interface irb.400
   ```

   Configuration for Customer Device 2:

   ```
   set interfaces irb unit 200 family inet address 10.10.20.6/30
   set interfaces irb unit 400 family inet address 10.10.40.6/30
   set interfaces lo0 unit 0 family inet address 10.10.10.11/32
   set interfaces ae20 unit 0 family inet address 10.10.10.1/30
   set vlans AWS-LGW l3-interface irb.200
   set vlans AWS-SERVICE-LINK l3-interface irb.400
   ```

5. Configure the dynamic routing protocols needed to exchange routes with the Outpost.

   We are using BGP and OSPF in this example. Start with a basic BGP configuration to set up logging and hold-time timers.

This configuration should be applied to both Customer Device 1 and Customer Device 2:

```
set protocols bgp hold-time 30
set protocols bgp log-updown
```

6. For BGP peering with the Outpost, set up a peering session for each of the VLANs used for the AWS connection. BGP learns the AWS control routes from one peering and the AWS local connection routes from the second peering.

   This example uses EBGP with private ASNs to build the peerings: 64512 is used for the data center side and 64513 for the Outpost side.

   Configuration for Customer Device 1:

   ```
   set protocols bgp group AWS-OUTPOST type external
   set protocols bgp group AWS-OUTPOST description "Connection to AWS Outpost"
   set protocols bgp group AWS-OUTPOST family inet unicast
   set protocols bgp group AWS-OUTPOST local-as 64512
   set protocols bgp group AWS-OUTPOST multipath multiple-as
   set protocols bgp group AWS-OUTPOST neighbor 10.10.20.1 peer-as 64513
   set protocols bgp group AWS-OUTPOST neighbor 10.10.40.1 peer-as 64513
   ```

   Configuration for Customer Device 2:

   ```
   set protocols bgp group AWS-OUTPOST type external
   set protocols bgp group AWS-OUTPOST description "Connection to AWS Outposts"
   set protocols bgp group AWS-OUTPOST family inet unicast
   set protocols bgp group AWS-OUTPOST local-as 64512
   set protocols bgp group AWS-OUTPOST multipath multiple-as
   set protocols bgp group AWS-OUTPOST neighbor 10.10.20.5 peer-as 64513
   set protocols bgp group AWS-OUTPOST neighbor 10.10.40.5 peer-as 64513
   ```

7. (Optional) You can create a virtual router VRF to separate routing for the AWS service link traffic from the main data center fabric, and to create BGP peering for the service link. Once the VRF is in place, you need to ensure that the service link maintains separation through the data center fabric to the AWS gateway.

   ```
   set protocols bgp group AWS-OUTPOST type external
   set protocols bgp group AWS-OUTPOST description "Connection to AWS Outposts"
   set protocols bgp group AWS-OUTPOST family inet unicast
   ```

```
set protocols bgp group AWS-OUTPOST local-as 64512
set protocols bgp group AWS-OUTPOST multipath multiple-as
set protocols bgp group AWS-OUTPOST neighbor 10.10.20.1 peer-as 64513
set routing-instances AWS-SERVICE instance-type virtual-router
set routing-instances AWS-SERVICE interface irb.400
set routing-instances AWS-SERVICE protocols bgp group outposts-service-link type external
set routing-instances AWS-SERVICE protocols bgp group outposts-service-link local-as 64512
set routing-instances AWS-SERVICE protocols bgp group outposts-service-link neighbor
10.10.40.1 peer-as 64513
```

8. With the EBGP peerings established between the data center and the Outpost, configure IBGP to propagate the learned routes between the two QFX devices.

   If a link fails, the QFX devices use this interconnect link to maintain reachability to the data center or the Outpost.

   Configuration for Customer Device 1:

   ```
   set protocols bgp group INTERCONNECT local-as 64512
   set protocols bgp group INTERCONNECT type internal
   set protocols bgp group INTERCONNECT local-address 10.10.10.10
   set protocols bgp group INTERCONNECT neighbor 10.10.10.11
   ```

   Configuration for Customer Device 2:

   ```
   set protocols bgp group INTERCONNECT local-as 64512
   set protocols bgp group INTERCONNECT type internal
   set protocols bgp group INTERCONNECT local-address 10.10.10.11
   set protocols bgp group INTERCONNECT neighbor 10.10.10.10
   ```

9. Define the routing policies to exchange routes between the Outpost and the data center fabric. In this example we assume that the data center fabric is running OSPF as the IGP. We want to learn routes from both the Outposts connections and advertise the data center fabric routes to the Outpost Configure policies to advertise OSPF and static routes from the fabric to the Outpost. Default behavior of BGP will import the routes learned from Outpost.

   This configuration should be applied to both Customer Device 1 and Customer Device 2:

   ```
   set policy-options policy-statement AWS-OUTPOST-TO-OSPF term redistribute-bgp from protocol
   bgp
   set policy-options policy-statement AWS-OUTPOST-TO-OSPF term redistribute-bgp from external
   ```

```
set policy-options policy-statement AWS-OUTPOST-TO-OSPF term redistribute-bgp then accept
set policy-options policy-statement INTERCONNECT-EXPORT term reject-loopback from route-
filter 10.10.10.10/32 exact
set policy-options policy-statement INTERCONNECT-EXPORT term reject-loopback from route-
filter 10.10.10.11/32 exact
set policy-options policy-statement INTERCONNECT-EXPORT term reject-loopback then reject
set policy-options policy-statement INTERCONNECT-EXPORT term redistribute-static from
protocol static
set policy-options policy-statement INTERCONNECT-EXPORT term redistribute-static then accept
set policy-options policy-statement INTERCONNECT-EXPORT term next-hop-self from protocol bgp
set policy-options policy-statement INTERCONNECT-EXPORT term next-hop-self from route-type
external
set policy-options policy-statement INTERCONNECT-EXPORT term next-hop-self then next-hop
self
set policy-options policy-statement INTERCONNECT-EXPORT term ospf from protocol ospf
set policy-options policy-statement INTERCONNECT-EXPORT term ospf then accept
set policy-options policy-statement INTERCONNECT-EXPORT term ospf-external from protocol
ospf
set policy-options policy-statement INTERCONNECT-EXPORT term ospf-external from external
set policy-options policy-statement INTERCONNECT-EXPORT term ospf-external then accept
set policy-options policy-statement OUTPOST-EXPORT term locals from protocol direct
set policy-options policy-statement OUTPOST-EXPORT term locals then accept
set policy-options policy-statement OUTPOST-EXPORT term statics from protocol static
set policy-options policy-statement OUTPOST-EXPORT term statics then accept
set policy-options policy-statement OUTPOST-EXPORT term ospf from protocol ospf
set policy-options policy-statement OUTPOST-EXPORT term ospf then accept
set policy-options policy-statement OUTPOST-EXPORT term ospf-external from protocol ospf
set policy-options policy-statement OUTPOST-EXPORT term ospf-external from external
set policy-options policy-statement OUTPOST-EXPORT term ospf-external then accept
set protocols bgp group AWS-OUTPOST export OUTPOST-EXPORT
set protocols bgp group INTERCONNECT export INTERCONNECT-EXPORT
```

10. (Optional) When using a dedicated VRF for the service link, apply the routing policy to the additional BGP group.

    ```
    set routing-instances AWS-SERVICE protocols bgp group outposts-service-link export OUTPOST-
    EXPORT
    ```

11. To exchange the QFX loopback addresses for the IBGP sessions, set up OSPF on the link between the two QFX devices. Attach a routing policy to distribute the Outpost learned routes into OSPF.

    Ensure the OSPF configuration is consistent with any existing OSPF fabric configurations.

This configuration should be applied to both Customer Device 1 and Customer Device 2:

```
set protocols ospf export AWS-OUTPOST-TO-OSPF
set protocols ospf area 0.0.0.0 interface lo0.0 passive
set protocols ospf area 0.0.0.0 interface ae20 interface-type p2p
```

12. The last step is to add routing options for graceful restart and enable per-flow load balancing.

    This configuration should be applied to both Customer Device 1 and Customer Device 2:

```
set routing-options graceful-restart
set policy-options policy-statement ECMP-POLICY then load-balance per-packet
set routing-options forwarding-table export ECMP-POLICY
set routing-options forwarding-table ecmp-fast-reroute
```

SEE ALSO

Network Management and Monitoring Guide

Data Center EVPN-VXLAN Fabric Architecture Guide