User Guide for racks

AWS Outposts

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Outposts: User Guide for racks

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is AWS Outposts?

AWS Outposts is a fully managed service that extends AWS infrastructure, services, APIs, and tools to customer premises. By providing local access to AWS managed infrastructure, AWS Outposts enables customers to build and run applications on premises using the same programming interfaces as in AWS Regions, while using local compute and storage resources for lower latency and local data processing needs.

An Outpost is a pool of AWS compute and storage capacity deployed at a customer site. AWS operates, monitors, and manages this capacity as part of an AWS Region. You can create subnets on your Outpost and specify them when you create AWS resources such as EC2 instances, EBS volumes, ECS clusters, and RDS instances. Instances in Outpost subnets communicate with other instances in the AWS Region using private IP addresses, all within the same VPC.

> **Note**
>
> You cannot connect an Outpost to another Outpost or Local Zone that is within the same VPC.

For more information, see the AWS Outposts product page.

Key concepts

These are the key concepts for AWS Outposts.

- **Outpost site** – The customer-managed physical buildings where AWS will install your Outpost. A site must meet the facility, networking, and power requirements for your Outpost.
- **Outpost capacity** – Compute and storage resources available on the Outpost. You can view and manage the capacity for your Outpost from the AWS Outposts console.
- **Outpost equipment** – Physical hardware that provides access to the AWS Outposts service. The hardware includes racks, servers, switches, and cabling owned and managed by AWS.
- **Outposts racks** – An Outpost form factor that is an industry-standard 42U rack. Outpost racks include rack-mountable servers, switches, a network patch panel, a power shelf and blank panels.
- **Outposts ACE racks** – The Aggregation, Core, Edge (ACE) rack acts as a network aggregation point for multi-rack Outpost deployments. The ACE rack reduces the number of physical

networking port and logical interface requirements by providing connectivity between multiple Outpost compute racks in your logical Outposts and your on-premise network.

You must install an ACE rack if you have five or more compute racks. If you have less than five compute racks but plan to expand to five or more racks in the future, we recommend that you install an ACE rack at the earliest.

For additional information on ACE racks, see Scaling AWS Outposts rack deployments with ACE racks.

- **Outposts servers** – An Outpost form factor that is an industry-standard 1U or 2U server, which can be installed in a standard EIA-310D 19 compliant 4 post rack. Outpost servers provide local compute and networking services to sites that have limited space or smaller capacity requirements.
- **Service link** – Network route that enables communication between your Outpost and its associated AWS Region. Each Outpost is an extension of an Availability Zone and its associated Region.
- **Local gateway (LGW)** – A logical interconnect virtual router that enables communication between an Outpost rack and your on-premises network.
- **Local network interface** – A network interface that enables communication from an Outpost server and your on-premises network.

AWS resources on Outposts

You can create the following resources on your Outpost to support low-latency workloads that must run in close proximity to on-premises data and applications:

Compute

Resource type	Racks	Servers
Amazon EC2 instances	 Y	 Yes
Amazon ECS clusters	 Y	 Yes

Resource type	Racks	Servers
Amazon EKS nodes	Y	No

Database and analytics

Resource type	Racks	Servers
Amazon ElastiCache nodes (Redis cluster, Memcached cluster)	Y	No
Amazon EMR clusters	Y	No
Amazon RDS DB instances	Y	No

Networking

Resource type	Racks	Servers
App Mesh Envoy proxy	Y	Yes
Application Load Balancers	Y	No

Resource type	Racks	Servers
Amazon VPC subnets	 Y	 Yes
Amazon Route 53	 Y	 No

Storage

Resource type	Racks	Servers
Amazon EBS volumes	 Y	 No
Amazon S3 buckets	 Y	 No

Other AWS services

Service	Racks	Servers
AWS IoT Greengrass	 Y	 Yes
Amazon SageMaker Edge Manager	 Y	 Yes

Pricing

You can choose from a variety of Outpost configurations, each providing a combination of EC2 instance types and storage options. The price for rack configurations includes installation, removal, and maintenance. For servers, you must install and maintain the equipment.

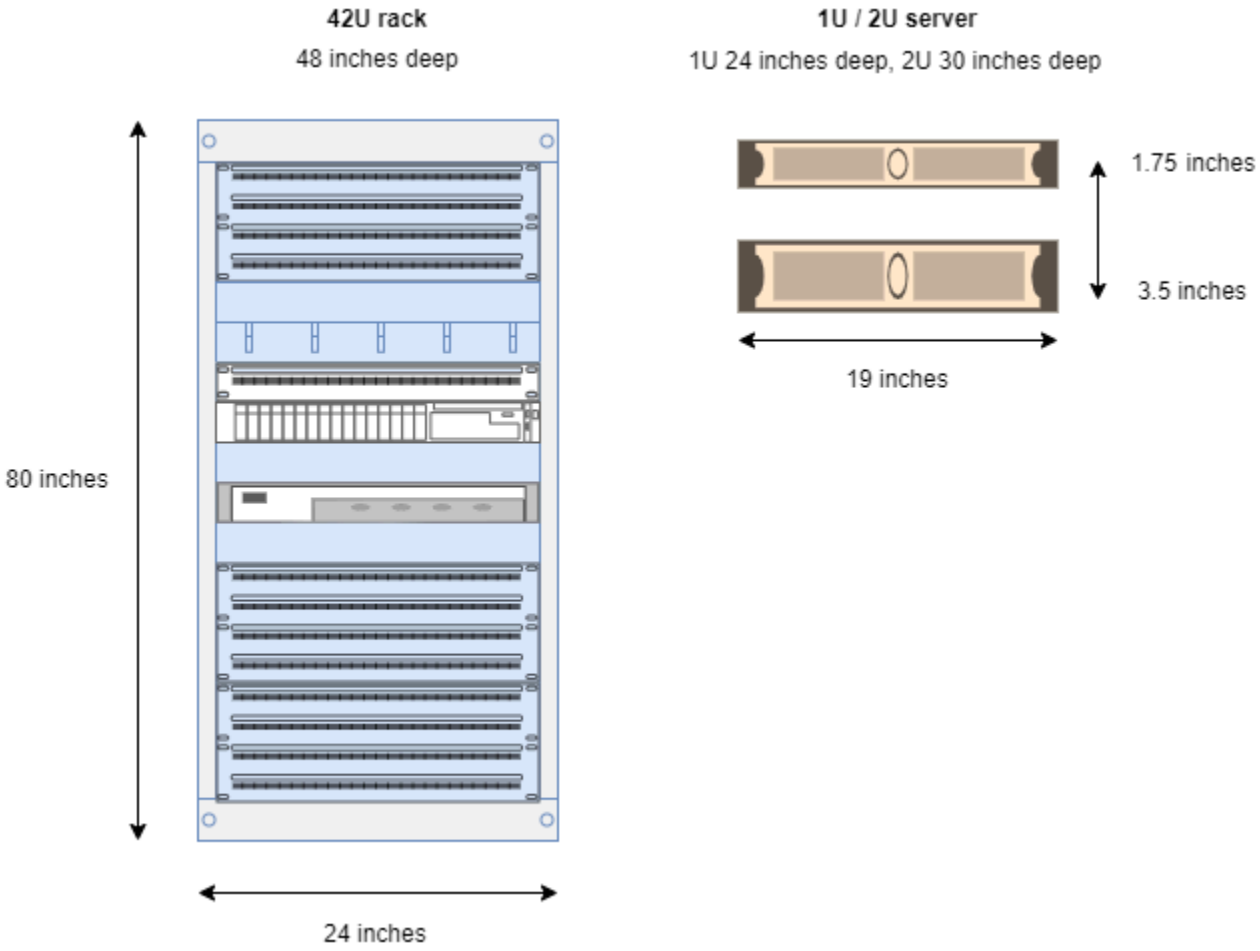
You purchase a configuration for a 3-year term and can choose from three payment options: All Upfront, Partial Upfront, and No Upfront. If you choose the Partial option or the No Upfront payment option, monthly charges will apply. Any upfront charges apply 24 hours after your Outpost is installed and the compute and storage capacity is available for use. For more information, see:

- AWS Outposts rack pricing
- AWS Outposts servers pricing

How AWS Outposts works

AWS Outposts is designed to operate with a constant and consistent connection between your Outpost and an AWS Region. To achieve this connection to the Region, and to the local workloads in your on-premises environment, you must connect your Outpost to your on-premises network. Your on-premises network must provide wide area network (WAN) access back to the Region and to the internet. It must also provide LAN or WAN access to the local network where your on-premises workloads or applications reside.

The following diagram illustrates both Outpost form factors.



Contents

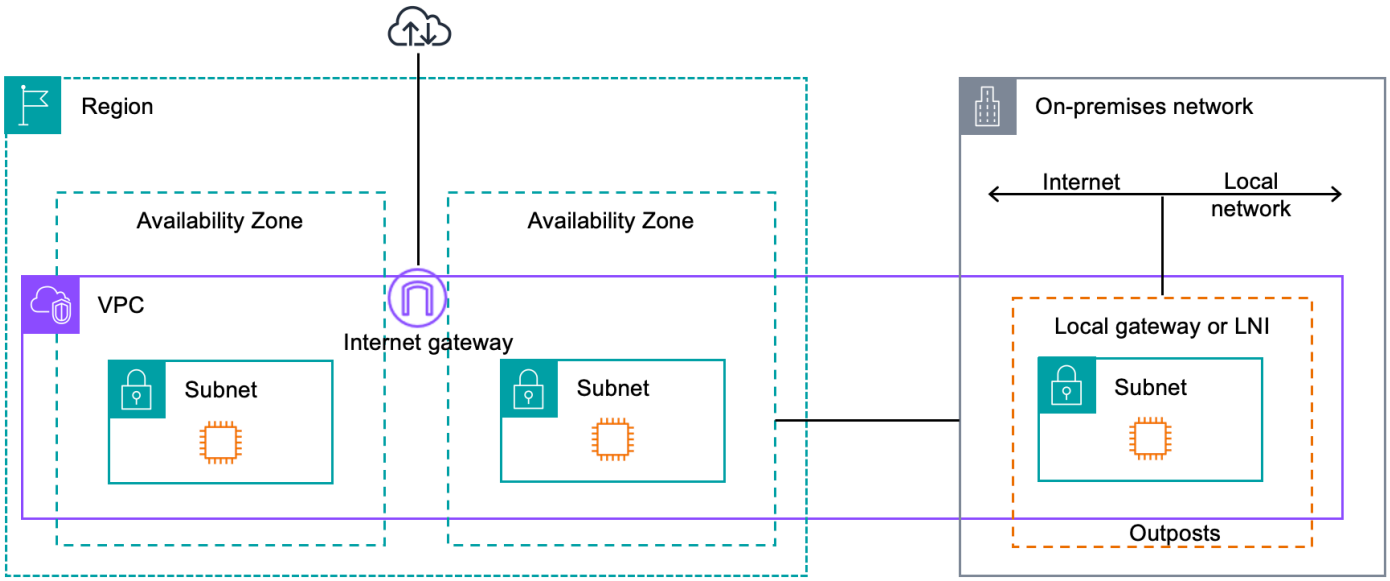
- DNS
- Service link
- Local gateways
- Local network interfaces

Network components

AWS Outposts extends an Amazon VPC from an AWS Region to an Outpost with the VPC components that are accessible in the Region, including internet gateways, virtual private gateways, Amazon VPC Transit Gateways, and VPC endpoints. An Outpost is homed to an Availability Zone in the Region and is an extension of that Availability Zone that you can use for resiliency.

The following diagram shows the network components for your Outpost.

- An AWS Region and an on-premises network
- A VPC with multiple subnets in the Region
- An Outpost in the on-premises network
- Connectivity between the Outpost and local network provided by either a local gateway (racks) or a local network interface (servers)



VPCs and subnets

A virtual private cloud (VPC) spans all Availability Zones in its AWS Region. You can extend any VPC in the Region to your Outpost by adding an Outpost subnet. To add an Outpost subnet to a VPC, specify the Amazon Resource Name (ARN) of the Outpost when you create the subnet.

Outposts support multiple subnets. You can specify the EC2 instance subnet when you launch the EC2 instance in your Outpost. You cannot specify the underlying hardware where the instance is deployed, because the Outpost is a pool of AWS compute and storage capacity.

Each Outpost can support multiple VPCs that can have one or more Outpost subnets. For information about VPC quotas, see Amazon VPC Quotas in the *Amazon VPC User Guide*.

You create Outpost subnets from the VPC CIDR range of the VPC where you created the Outpost. You can use the Outpost address ranges for resources, such as EC2 instances that reside in the Outpost subnet.

Routing

By default, every Outpost subnet inherits the main route table from its VPC. You can create a custom route table and associate it with an Outpost subnet.

The route tables for Outpost subnets work as they do for Availability Zone subnets. You can specify IP addresses, internet gateways, local gateways, virtual private gateways, and peering connections as destinations. For example, each Outpost subnet, either through the inherited main route table, or a custom table, inherits the VPC local route. This means that all traffic in the VPC, including the Outpost subnet with a destination in the VPC CIDR remains routed in the VPC.

Outpost subnet route tables can include the following destinations:

- **VPC CIDR range** – AWS defines this at installation. This is the local route and applies to all VPC routing, including traffic between Outpost instances in the same VPC.
- **AWS Region destinations** – This includes prefix lists for Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB gateway endpoint, AWS Transit Gateways, virtual private gateways, internet gateways, and VPC peering.

  If you have a peering connection with multiple VPCs on the same Outpost, the traffic between the VPCs remains in the Outpost and does not use the service link back to the Region.

- **Intra-VPC communication across Outposts with local gateway** – You can establish communication between subnets in the same VPC across different Outposts with local gateways using direct VPC routing. For more information, see:
  - Direct VPC routing
  - Routing to an AWS Outposts local gateway

DNS

For network interfaces connected to a VPC, EC2 instances in Outposts subnets can use the Amazon Route 53 DNS Service to resolve domain names to IP addresses. Route 53 supports DNS features, such as domain registration, DNS routing, and health checks for instances running in your Outpost. Both public and private hosted Availability Zones are supported for routing traffic to specific domains. Route 53 resolvers are hosted in the AWS Region. Therefore, service link connectivity from the Outpost back to the AWS Region must be up and running for these DNS features to work.

You might encounter longer DNS resolution times with Route 53, depending on the path latency between your Outpost and the AWS Region. In such cases, you can use the DNS servers installed locally in your on-premises environment. To use your own DNS servers, you must create DHCP option sets for your on-premises DNS servers and associate them with the VPC. You must also ensure that there is IP connectivity to these DNS servers. You might also need to add routes to the local gateway routing table for reachability but this is only an option for Outpost racks with local gateway. Because DHCP option sets have a VPC scope, instances in both the Outpost subnets and the Availability Zone subnets for the VPC will try to use the specified DNS servers for DNS name resolution.

Query logging is not supported for DNS queries originating from an Outpost.

Service link

The service link is a connection from your Outpost back to your chosen AWS Region or Outposts home Region. The service link is an encrypted set of VPN connections that are used whenever the Outpost communicates with your chosen home Region. You use a virtual LAN (VLAN) to segment traffic on the service link. The service link VLAN enables communication between the Outpost and the AWS Region for both management of the Outpost and intra-VPC traffic between the AWS Region and Outpost.

Your service link is created when your Outpost is provisioned. If you have a server form factor, you create the connection. If you have a rack, AWS creates the service link. For more information, see:

- Outpost connectivity to AWS Regions
- Application/workload routing in the *AWS Outposts High Availability Design and Architecture Considerations* AWS Whitepaper

Local gateways

Outpost racks include a local gateway to provide connectivity to your on-premises network. If you have an Outpost rack, you can include a local gateway as target where the destination is your on-premises network. Local gateways are only available for Outpost racks and can only be used in VPC and subnet route tables that are associated with an Outpost rack. For more information, see:

- Local gateway
- Application/workload routing in the *AWS Outposts High Availability Design and Architecture Considerations* AWS Whitepaper

Local network interfaces

Outpost servers include a local network interface to provide connectivity to your on-premises network. A local network interface is available only for Outposts servers running on an Outpost subnet. You cannot use a local network interface from an EC2 instance on an Outpost rack or in the AWS Region. The local network interface is meant only for on-premises locations. For more information, see Local network interface in the *AWS Outposts User Guide for Outposts servers*.

Site requirements for Outposts rack

An Outpost site is the physical location where your Outpost operates. Sites are only available in select countries and territories. For more information, see, AWS Outposts rack FAQs. Refer to the question: **In which countries and territories is Outposts rack available?**

This page covers the requirements for Outposts rack. If you are installing an Aggregation, Core, Edge (ACE) rack, your site must also meet the requirements listed in Site requirements for Outposts ACE racks.

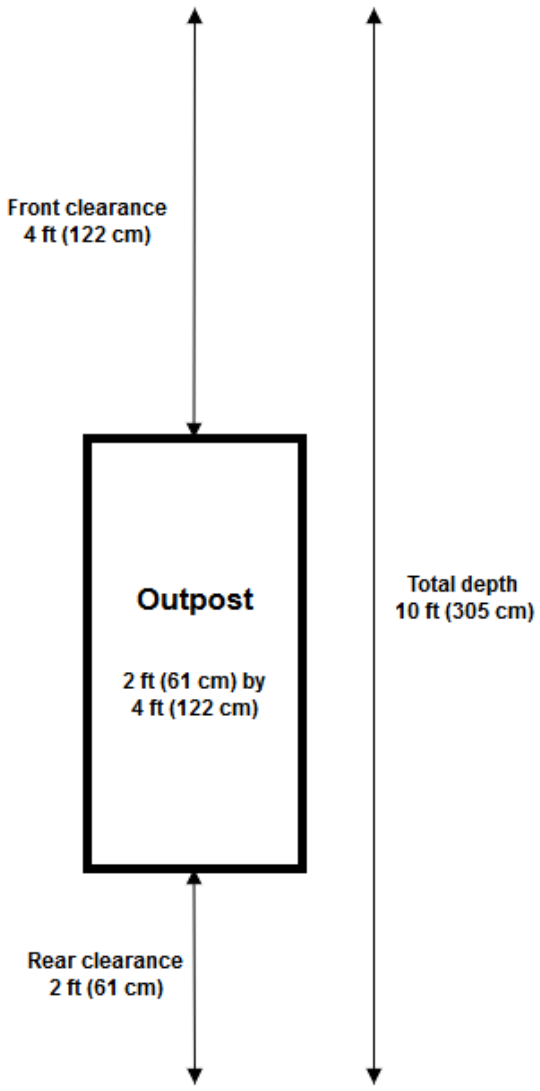
For the requirements for Outposts servers, see Site requirements for Outposts servers in the *AWS Outposts User Guide for Outposts servers*.

Facility

These are the facility requirements for racks.

- **Temperature and humidity** – The ambient temperature must be between 41° F (5° C) and 95° F (35° C). The relative humidity must be between 8 percent and 80 percent with no condensation.
- **Airflow** – Racks draw cold air from the front aisle and exhaust hot air to the back aisle. The rack position must provide at least 145.8 times the kVA of cubic feet per minute (CFM) airflow.
- **Loading dock** – Your loading dock must accommodate a rack crate that is 94 inches (239 cm) high by 54 inches (138 cm) wide by 51 inches (130 cm) deep.
- **Weight support** – Weight varies by configuration. You can find the weight for your configuration specified in the order summary at the rack point loads. The location where the rack is installed and the path to that location must support the specified weight. This includes any freight and standard elevators along the path.
- **Clearance** – The rack is 80 inches (203 cm) high by 24 inches (61 cm) wide by 48 inches (122 cm) deep. Any doorways, hallways, turns, ramps, and elevators must provide sufficient clearance. At the final resting position, there must be a 24 inch (61 cm) wide by 48 inch (122 cm) deep area for the Outpost, with an additional 48 inches (122 cm) of front clearance and 24 inches (61 cm) of rear clearance. The total minimum area required for the Outpost is 24 inch (61 cm) wide by 10 feet (305 cm) deep.

  The following diagram shows the total minimum area required for the Outpost, including clearance.



- **Seismic bracing** – To the extent required by regulation or code, you will install and maintain appropriate seismic anchorage and bracing for the rack while it is in your facility. AWS provides floor brackets that provide protection for up to 2.0G of seismic activity with all Outposts racks.
- **Bonding point** – We recommend that you provide a bonding wire / point at the rack position so that the AWS-certified technician can bond the racks during installation.
- **Facility access** – You will not change the facility in a way that negatively affects the ability of AWS to access, service, or remove the Outpost.
- **Elevation** – The elevation of the room where the rack is installed must be below 10,005 feet (3,050 meters).

Networking

These are the networking requirements for racks.

- Provide uplinks with speeds of 1 Gbps, 10 Gbps, 40 Gbps, or 100 Gbps.

  For bandwidth recommendations for the service link connection, see Bandwidth recommendations.
- Provide either single-mode fiber (SMF) with Lucent Connector (LC), multimode fiber (MMF), or MMF OM4 with LC.
- Provide one or two upstream devices, which can be switches or routers. We recommend two devices to provide high availability.

Network readiness checklist

Use this checklist when you are gathering the information for your Outpost configuration. This includes the LAN, WAN, and any devices between the Outpost and local traffic destinations, and the destination in the AWS Region.

Uplink speed, ports, and fiber

Uplink speed and ports

An Outpost has two Outpost network devices that attach to your local network. The number of uplinks each device can support depends on your bandwidth needs and what your router can support. For more information, see Physical connectivity.

The following list shows how many uplink ports are supported for each Outpost network device, based on the uplink speed.

1 Gbps

1, 2, 4, 6, or 8 uplinks

10 Gbps

1, 2, 4, 8, 12, or 16 uplinks

40 Gbps or 100 Gbps

1, 2, or 4 uplinks

Fiber

The following fiber types are supported:

- Single-mode fiber (SMF) with Lucent Connector (LC)
- Multi-mode fiber (MMF) or MMF OM4 with LC

Depending on the uplink speed and the type of fiber that you choose, the following optical standards are supported.

Uplink speed	Fiber type	Optical standard
1 Gbps	SMF	– 1000Base-LX
1 Gbps	MMF	– 1000Base-SX
10 Gbps	SMF	– 10GBASE-IR – 10GBASE-LR
10 Gbps	MMF	– 10GBASE-SR
40 Gbps	SMF	– 40GBASE-IR4 (LR4L) – 40GBASE-LR4
4 x 10 Gbps breakout application	MMF	– 40GBASE-ESR4 – 40GBASE-SR4
100 Gbps	SMF	– 100G PSM4 MSA – 100GBASE-CWDM4 – 100GBASE-LR4
4 x 25 Gbps breakout application	MMF	– 100GBASE-SR4

Outpost link aggregation and VLANs

Link aggregation control protocol (LACP) is required between the Outpost and your network. You must use dynamic LAG with LACP.

The following VLANs are required for each Outpost network device. For more information, see Virtual LANs.

Outpost network device	Service link VLAN	Local gateway VLAN
#1	Valid values: 1-4094	Valid values: 1-4094
#2	Valid values: 1-4094	Valid values: 1-4094

For each Outpost network device, you can choose whether to use the same VLANs or different VLANs for the service link and local gateway. However, we recommend that each Outpost network device have a different VLAN from the other Outpost network device. For more information, see Link aggregation and Virtual LANs.

We also recommend redundant layer 2 connectivity. LACP is used for link aggregation and is not used for high availability. LACP between the Outpost network devices is not supported.

Outpost network device IP connectivity

Each of the two Outpost network devices requires a CIDR and IP address for the service link and local gateway VLANs. We recommend allocating a dedicated subnet for each network device with a /30 or /31 CIDR. Specify a subnet and an IP address from the subnet for the Outpost to use. For more information, see Network layer connectivity.

Outpost network device	Service link requirements	Local gateway requirements
#1	– Service link CIDR (/30 or /31) – Service link IP address	– Local gateway CIDR (/30 or /31) – Local gateway IP address
#2	– Service link CIDR (/30 or /31)	– Local gateway CIDR (/30 or /31)

Outpost network device	Service link requirements	Local gateway requirements
	– Service link IP address	– Local gateway IP address

Service link maximum transmission unit (MTU)

The network must support 1500-bytes MTU between the Outpost and the service link endpoints in the parent AWS Region. For more information about the service link, see AWS Outposts connectivity to AWS Regions.

Service link Border Gateway Protocol

The Outpost establishes an external BGP (eBGP) peering session between each Outpost network device and your local network device for service link connectivity over the service link VLAN. For more information, see Service link BGP connectivity.

Outpost	Service link BGP requirements
Your Outpost	– Outpost BGP Autonomous System Number (ASN). 2-byte (16-bit) or 4-byte (32-bit). From your private ASN range (64512-65534 or 4200000000-4294967294). – Infrastructure CIDR (/26 required, advertised as two contiguous /27s).

Local network device	Service link BGP requirements
#1	– Service link BGP peer IP address. – Service link BGP peer ASN. 2-byte (16-bit) or 4-byte (32-bit).
#2	– Service link BGP peer IP address. – Service link BGP peer ASN. 2-byte (16-bit) or 4-byte (32-bit).

Service link firewall

UDP and TCP 443 must be statefully listed in the firewall.

Protocol	Source Port	Source Address	Destination Port	Destination Address
UDP	443	Outpost service link /26	443	Outpost Region's public routes
TCP	1025-65535	Outpost service link /26	443	Outpost Region's public routes

You can use an AWS Direct Connect connection or a public internet connection to connect the Outpost back to the AWS Region. For Outpost service link connectivity, you can use NAT or PAT at your firewall or edge router. Service link establishment is always initiated from the Outpost.

Local gateway Border Gateway Protocol

The Outpost establishes an eBGP peering session from each Outpost network device to a local network device for connectivity from your local network to the local gateway. For more information, see Local gateway BGP connectivity.

Outpost	Local gateway BGP requirements
Your Outpost	– Outpost BGP Autonomous System Number (ASN). 2-byte (16-bit) or 4-byte (32-bit). From your private ASN range (64512-65534 or 4200000000-4294967294). – CoIP CIDR to advertise (public or private, /26 minimum).

Local network devices	Local gateway BGP requirements
#1	– Local gateway BGP peer IP address.

Local network devices	Local gateway BGP requirements
	– Local gateway BGP peer ASN. 2-byte (16-bit) or 4-byte (32-bit).
#2	– Local gateway BGP peer IP address. – Local gateway BGP peer ASN. 2-byte (16-bit) or 4-byte (32-bit).

Power

The Outposts power shelf supports three power configurations: 5 kVA, 10 kVA, or 15 kVA. The configuration of the power shelf depends on the total power draw of the Outpost capacity. For example, if your Outpost resource has a maximum power draw of 9.7 kVA, you must provide the power configurations for 10 kVA: 4 x L6-30P or IEC309, 2 drops to S1, and 2 drops to S2 for redundant, single-phase power. The three power configurations are described in the following second table.

To see the power draw requirements for different Outpost resources, choose **Browse catalog** in the AWS Outposts console at https://console.aws.amazon.com/outposts/.

Requirement	Specification
AC line voltage	**Single-phase** 208 to 277 VAC; 50 or 60 Hz **Three-phase**: • 208 to 250 VAC (Delta); 50 to 60 Hz • 346 to 480 VAC (Wye); 50 to 60 Hz
Power consumption	5 kVA (4 kW), 10 kVA (9 kW), or 15 kVA (13 kW)
AC protection (upstream power breakers)	For both 1N input (non-redundant) and 2N input (redundant): 30 A, 32 A, or 50 A with D-curve or K-curve circuit breaker. For 2N input (redundant) only: C-curve, D-curve, or K-curve circuit breaker.

Requirement	Specification
	B-curve or lower is not supported.
AC inlet type (receptacle)	**Single-phase** 3xL6-30P, P+P+E, 30A or 3xIEC60309 P+N+E, IP67, 32A plugs **Three-phase, Wye** 1xIEC60309, 3P+N+E, IP67, clock position 7, 30A plug or 1xIEC60309, 3P+N+E, IP67, clock position 6, 32A plug **Three-phase, Delta** 1xNon-NEMA twistlock Hubbell CS8365C, 3P+E, center ground, 50A plug **Note** The best practice is to mate an IP67 plug with an IP67 receptacle. If that isn't possible, the IP67 plug will mate with an IP44 receptacle. The rating of the combined plug and socket will become the lower rating (IP44).
Whip length	10.25 ft (3 m)
Whip - Rack cabling input	From above or below the rack

The power shelf has two inputs, S1 and S2, that can be configured as follows.

	Redundant, single-phase	Redundant, three-phase	Single-phase	Three-phase
5 kVA	2 x L6-30P or IEC309; 1 drop to S1 and 1 drop to S2	2 x AH530P7W, AH532P6W, or CS8365C; 1 drop to S1 and 1 drop to S2	Not offered	1 x AH530P7W, AH532P6W or CS8365C; 1 drop to S1
10 kVA	4 x L6-30P or IEC309; 2 drops to S1 and 2 drops to S2		2 x L6-30P or IEC309; 2 drops to S1	

	Redundant, single-phase	Redundant, three-phase	Single-phase	Three-phase
15 kVA	6 x L6-30P or IEC309; 3 drops to S1 and 3 drops to S2		3 x L6-30P or IEC309; 3 drops to S1	

If the AC whips that AWS provides as previously described must be fitted with an alternate power plug, consider the following:

- Only a certified customer-provided electrician should modify the AC whip to fit a new plug type.
- The installation should comply with all applicable national, state, and local safety requirements, and be inspected as required for electrical safety.
- You, the customer, should notify your AWS representative of modifications to the AC whip plug. Upon request, you will provide information about the modifications to AWS. You'll also include any safety inspection records issued by the authority having jurisdiction. This is a requirement to validate safety of the installation before having AWS employees perform work on the equipment.

Order fulfillment

To fulfill the order, AWS will schedule a date and time with you. You will also receive a checklist of items to verify or provide before the installation.

The AWS installation team will arrive at your site at the scheduled date and time. They will place the rack at the identified position. You and your electrician are responsible for performing the electrical connection and installation to the rack.

You must ensure that electrical installations, and any changes to those installations, are performed by a certified electrician in accordance with all applicable laws, codes, and best practices. You must obtain approval from AWS in writing prior to making any changes to the Outpost hardware or the electrical installations. You agree to provide AWS with documentation verifying compliance and the safety of any changes. AWS is not responsible for any risks created by the Outpost electrical installation or facility electrical wiring or any changes. You must not make any other changes to the Outposts hardware.

The team will establish network connectivity for the Outposts rack over the uplink that you provide, and will configure the rack's capacity.

The installation is complete when you confirm that the Amazon EC2 and Amazon EBS capacity for your Outposts rack is available from your AWS account.

Site requirements for Outposts ACE racks

> **Note**
>
> Skip this section if you don't need an ACE rack.

An Aggregation, Core, Edge (ACE) rack acts as a network aggregation point for multi-rack Outpost deployments. You must install an ACE rack if you have five or more compute racks. If you have less than five compute racks but plan to expand to five or more racks in the future, we recommend that you install an ACE rack at the earliest.

To install an ACE rack, you must meet the requirements in this section in addition to the requirements listed in Site requirements for Outposts rack.

Facility

These are the facility requirements for an ACE rack.

- **Power** – All racks are shipped with 10kVA single phase (AA+BB; IEC60309 or L6-30P Whip connector types).
- **Weight support** – Rack weight is 705 lbs; 320 kg.
- **Clearance/Size dimension** – Rack height is 80 inches; 203 cm.

> **Note**
>
> ACE racks are not fully enclosed and don’t include a front or a rear door.

Networking

These are the networking requirements for an ACE rack. To understand how the ACE rack connects the Outposts networking devices, your on-premises networking devices, and your Outpost racks, see ACE rack connectivity.

- **Rack network requirements** – Ensure that you meet the requirements listed in the Network readiness checklist and Local network connectivity for racks sections except for the following changes:
  - The ACE rack has four networking devices that connect to the upstream devices, not two as in the case of a single Outposts rack.
  - ACE racks do not support 1 Gbps uplinks.
- **Uplink speed** – Provide uplinks with speeds of 10 Gbps, 40 Gbps, or 100 Gbps. For bandwidth recommendations for the service link connection, Service link bandwidth recommendations.

  > **Important**
  >
  > ACE racks do not support 1 Gbps uplinks.

- **Fiber** – Provide single-mode fiber (SMF) with Lucent Connector (LC), or multi-mode fiber (MMF) with Lucent Connector (LC). For the full list of supported fiber types and optical standards, see Uplink speed, ports, and fiber.
- **Upstream device** – Provide two or four upstream devices, which can be switches or routers.
- **Service VLAN and a Local Gateway VLAN** – For each of the four ACE networking device you must provide a Service VLAN and a different Local Gateway VLAN. You can choose to provide only two distinct VLANs, one for the Service VLAN and one for the Local gateway VLAN, or have different VLANs in each ACE networking device for both Service VLAN and LGW VLAN for a total of 8 different VLANs. For more information on how link aggregation groups (LAGs) and VLAN are used, see Link aggregation and Virtual LANs.
- **CIDR and IP address for the service link and local gateway VLANs** – We recommend allocating a dedicated subnet for each ACE networking device with a /30 or /31 CIDR. Alternatively, it is possible to allocate a single /29 subnet in each Service and Local Gateway VLAN. In both cases, you must specify the IP addresses for the ACE networking devices to use. For more information, see Network layer connectivity.
- **Customer and Outpost BGP Autonomous System Number (ASN) for service link VLAN and a Local Gateway VLAN** – The Outpost establishes an external BGP (eBGP) peering session between

each ACE rack device and your local network device for service link connectivity over the service link VLAN. In addition, it establishes an eBGP peering session from each ACE networking device to a local network device for connectivity from your local network to the local gateway. For more information, see Service link BGP connectivity and Local gateway BGP connectivity.

Important

Service link infrastructure subnets – A service link infrastructure subnet (must be /26) is required for each compute rack included in your Outposts installation.

Power

These are the power requirements for an ACE rack.

Requirement	Specification
AC line voltage	Single-phase 200 to 240 VAC; 50 or 60 Hz
Power consumption	10 kVA single phase (AA+BB)
AC protection (upstream power breakers)	For 2N input (redundant) only: C-curve, D-curve, or K-curve circuit breaker. B-curve or lower is not supported.
AC inlet type (receptacle)	IEC60309 or L6-30P whip connector types.

Get started with AWS Outposts

Order an Outpost to get started. After installation of your Outpost equipment, launch Amazon EC2 instances and access your on-premises network.

Tasks

- Create an Outpost and order Outpost capacity
- Launch an instance on your Outpost rack

Create an Outpost and order Outpost capacity

To begin using AWS Outposts, you must create an Outpost and order Outpost capacity.

Prerequisites

- Review the available configurations for your Outposts racks.
- An Outpost site is the physical location for your Outpost equipment. Before ordering capacity, verify that your site meets the requirements. For more information, see Site requirements for Outposts rack.
- You must have an AWS Enterprise Support plan or an AWS Enterprise On-Ramp Support plan.
- Determine which AWS account will own the Outpost. Use this account to create the Outposts site, create the Outpost, and place the order. Monitor the email associated with this account for information from AWS.

Tasks

- Step 1: Create a site
- Step 2: Create an Outpost
- Step 3: Place the order
- Step 4: Modify instance capacity
- Next steps

Step 1: Create a site

Create a site to specify the operating address. The operating address is the physical location for your Outposts racks.

Prerequisites

- Determine the operating address.

To create a site

1. Sign in to AWS using the AWS account that will own the Outpost.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. To select the parent AWS Region, use the Region selector in the upper-right corner of the page.
4. In the navigation pane, choose **Sites**.
5. Choose **Create site**.
6. For **Supported hardware type**, choose **Racks and servers**.
7. Enter a name, description, and operating address for your site.
8. For **Site details**, provide the requested information about the site.

   - **Max weight** – The maximum rack weight that this site can support, in lbs.
   - **Power draw** – The power draw available at the hardware placement position for the rack, in kVA.
   - **Power option** – The power option that you can provide for the hardware.
   - **Power connector** – The power connector that AWS should plan to provide for connections to the hardware.
   - **Power feed drop** – Indicate whether the power feed comes above or below the rack.
   - **Uplink speed** – The uplink speed the rack should support for the connection to the Region, in Gbps.
   - **Number of uplinks** – The number of uplinks for each Outpost networking device that you intend to use to connect the rack to your network.
   - **Fiber type** – The type of fiber that you will use to attach the rack to your network.
   - **Optical standard** – The type of optical standard that you will use to attach the rack to your network.

9. (Optional) For **Site notes**, enter any other information that might be useful for AWS to know about the site.
10. Read the facility requirements, and then select **I have read the facility requirements**.
11. Choose **Create site**.

Step 2: Create an Outpost

Create an Outpost for your racks. Then, specify this Outpost when you place your order.

Prerequisites

- Determine the AWS Availability Zone to associate with your site.

To create an Outpost

1. In the navigation pane, choose **Outposts**.
2. Choose **Create Outpost**.
3. Choose **Racks**.
4. Enter a name and description for your Outpost.
5. Choose an Availability Zone for your Outpost.
6. (Optional) To configure private connectivity, select **Use Private connectivity**. Choose a VPC and subnet in the same AWS account and Availability Zone as your Outpost. For more information, see the section called “Prerequisites”.
7. For **Site ID**, choose your site.
8. Choose **Create Outpost**.

Step 3: Place the order

Place an order for the Outposts racks that you need. After you submit the order, an AWS Outposts representative will contact you.

> **Important**
>
> You can't edit an order after you submit it so review all details carefully before submission. If you need to change an order, contact your AWS Account Manager.

Prerequisites

- Determine how you will pay for the order. You can pay all upfront, partially upfront, or nothing upfront. If you do not choose to pay all upfront, you'll pay monthly charges over the three-year term.

  The pricing includes delivery, installation, infrastructure service maintenance, and software patches and upgrades.
- Determine whether the delivery address is different than the operating address that you specified for the site.

To place an order

1. In the navigation pane, choose **Orders**.
2. Choose **Place order**.
3. For **Supported hardware type**, choose **Racks**.
4. To add capacity, choose a configuration. If the available configurations do not meet your needs, you can contact AWS to request a custom capacity configuration instead.
5. Choose **Next**.
6. Choose **Use an existing Outpost** and select your Outpost.
7. Choose **Next**.
8. Select a contract term and payment option.
9. Specify the shipping address. You can specify a new address or select the site's operating address. If you select the operating address, be aware that any future change to the site's operating address will not propagate to existing orders. If you need to change the shipping address on an existing order, contact your AWS Account Manager.
10. Choose **Next**.
11. On the **Review and order** page, verify that your information is correct and edit as needed. You will not be able to edit the order after you submit it.
12. Choose **Place order**.

Step 4: Modify instance capacity

An Outpost provides a pool of AWS compute and storage capacity at your site as a private extension of an Availability Zone in an AWS Region. Because the compute and storage capacity

available in the Outpost is finite and determined by the size and number of racks that AWS installs at your site, you get to decide how much Amazon EC2, Amazon EBS, and Amazon S3 on AWS Outposts capacity you need to run your initial workloads, accommodate future growth, and to provide extra capacity to mitigate server failures and maintenance events.

The capacity of each new Outpost order is configured with a default capacity configuration. You can convert the default configuration to create various instances to meet your business needs. To do so, you create a capacity task, specify the instance sizes and quantity, and run the capacity task to implement the changes.

Note

- You can change the quantity of instance sizes after you place the order for your Outposts.
- Instances sizes and quantities are defined at the Outpost level.
- Instances are placed automatically based on best practices.

To modify instance capacity

1. From the AWS Outposts console'sAWS Outposts left navigation pane, choose **Capacity tasks**.
2. On the **Capacity tasks** page, choose **Create capacity task**.
3. On the **Getting started** page, choose the order.
4. To modify capacity, you can use the steps in the console or upload a JSON file.

Console steps

1. Choose **Modify a new Outpost capacity configuration**.
2. Choose **Next**.
3. On the **Configure instance capacity** page, each instance type shows one instance size with the maximum quantity preselected. To add more instance sizes, choose **Add instance size**.
4. Specify the instance quantity and note the capacity that is displayed for that instance size.
5. View the message at the end of each instance-type section that informs you if you are over or under capacity. Make adjustments at the instance size or quantity level to optimize your total available capacity.

6. You can also request AWS Outposts to optimize the instance quantity for a specific instance size. To do so:

   a. Choose the instance size.

   b. Choose **Auto-balance** at the end of the related instance-type section.

7. For each instance type, ensure that the instance quantity is specified for at least one instance size.
8. Choose **Next**.
9. On the **Review and create** page, verify the updates that you are requesting.
10. Choose **Create**. AWS Outposts creates a capacity task.
11. On the capacity task page, monitor the status of the task.

> **Note**
>
> - AWS Outposts might request you to stop one or more running instances to enable running the capacity task. After you stop these instances, AWS Outposts will run the task.
> - If you need to change your capacity after you complete your order, contact AWS Support to make the changes.

Upload JSON file

1. Choose **Upload a capacity configuration**.
2. Choose **Next**.
3. On the **Upload capacity configuration plan** page, upload the JSON file that specifies the instance type, size, and quantity.

   Example

   Example JSON file:

```
{
    "RequestedInstancePools": [
        {
            "InstanceType": "c5.24xlarge",
            "Count": 1
```

```
        },
        {
            "InstanceType": "m5.24xlarge",
            "Count": 2
        }
    ]
}
```

4. Review the contents of the JSON file in the **Capacity configuration plan** section.
5. Choose **Next**.
6. On the **Review and create** page, verify the updates that you are requesting.
7. Choose **Create**. AWS Outposts creates a capacity task.
8. On the capacity task page, monitor the status of the task.

> **Note**
>
> - AWS Outposts might request you to stop one or more running instances to enable running the capacity task. After you stop these instances, AWS Outposts will run the task.
> - If you need to change your capacity after you complete your order, contact AWS Support to make the changes.

Next steps

You can view the status of your order using the AWS Outposts console. The initial status of your order is **Order received**. An AWS representative will contact you within three business days. You will receive an email confirmation when the status of your order changes to **Order processing**. An AWS representative may contact you to get any additional information that AWS requires.

If you have any questions about your order, contact AWS Support.

To fulfill the order, AWS will schedule a date and time with you.

You will also receive a checklist of items to verify or provide before the installation. The AWS installation team will arrive at your site at the scheduled date and time. The team will roll the rack to the identified position and your electrician can power the rack. The team will establish network connectivity for the rack over the uplink that you provide, and will configure the rack's capacity.

The installation is complete when you confirm that the Amazon EC2 and Amazon EBS capacity for your Outpost is available from your AWS account.

Launch an instance on your Outpost rack

After your Outpost is installed and the compute and storage capacity is available for use, you can get started by creating resources. Launch Amazon EC2 instances and create Amazon EBS volumes on your Outpost using an Outpost subnet. You can also create snapshots of Amazon EBS volumes on your Outpost. For more information applicable to Linux, see Local Amazon EBS snapshots on AWS Outposts in the *Amazon EC2 User Guide*. For more information applicable to Windows, see Local Amazon EBS snapshots on AWS Outposts in the *Amazon EC2 User Guide*.

Prerequisite

You must have an Outpost installed at your site. For more information, see Create an Outpost and order Outpost capacity.

Tasks

- Step 1: Create a VPC
- Step 2: Create a subnet and custom route table
- Step 3: Configure local gateway connectivity
- Step 4: Configure the on-premises network
- Step 5: Launch an instance on the Outpost
- Step 6: Test the connectivity

Step 1: Create a VPC

You can extend any VPC in the AWS Region to your Outpost. Skip this step if you already have a VPC that you can use.

To create a VPC for your Outpost

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. Choose the same Region as the Outposts rack.
3. On the navigation pane, choose **Your VPCs** and then choose **Create VPC**.

4. Choose **VPC only**.
5. (Optional) for **Name tag** enter a name for the VPC.
6. For **IPv4 CIDR block**, choose **IPv4 CIDR manual input** and enter the IPv4 address range for the VPC in the **IPv4 CIDR** text box.

> **Note**
>
> If you want to use Direct VPC routing, specify a CIDR range that does not overlap with the IP range that you use in your on-premises network.

7. For **IPv6 CIDR block**, choose **No IPv6 CIDR block**.
8. For **Tenancy**, choose **Default**.
9. (Optional) To add a tag to your VPC, choose **Add tag**, and enter a key and a value.
10. Choose **Create VPC**.

Step 2: Create a subnet and custom route table

You can create and add an Outpost subnet to any VPC in the AWS Region that the Outpost is homed to. When you do so, the VPC includes the Outpost. For more information, see Network components.

> **Note**
>
> If you are launching an instance in an Outpost subnet that has been shared with you by another AWS account, skip to Step 5: Launch an instance on the Outpost.

2a: Create an Outpost subnet

To create an Outpost subnet

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions**, **Create subnet**. You are redirected to create a subnet in the Amazon VPC console. We select the Outpost for you and the Availability Zone that the Outpost is homed to.

4. Select a VPC.
5. In **Subnet settings**, optionally name your subnet and specify an IP address range for the subnet.
6. Choose **Create subnet**.
7. (Optional)To make it easier to identify Outpost subnets, enable the **Outpost ID** column on the **Subnets** page. To enable the column, choose the **Preferences** icon, select **Outpost ID**, and choose **Confirm**.

2b: Create a custom route table

Use the following procedure to create a custom route table with a route to the local gateway. You can't use the same route table as the Availability Zone subnets.

To create a custom route table

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. On the navigation pane, choose **Route tables**.
3. Choose **Create route table**.
4. (Optional) For **Name**, enter a name for your route table.
5. For **VPC**, choose your VPC.
6. (Optional) To add a tag, choose **Add new tag** and enter the tag key and tag value.
7. Choose **Create route table**.

2c: Associate the Outpost subnet and custom route table

To apply route table routes to a particular subnet, you must associate the route table with the subnet. A route table can be associated with multiple subnets. However, a subnet can only be associated with one route table at a time. Any subnet not explicitly associated with a table is implicitly associated with the main route table by default.

To associate the Outpost subnet and custom route table

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the navigation pane, choose **Route tables**.
3. On the **Subnet associations** tab, choose **Edit subnet associations**.

4. Select the check box for the subnet to associate with the route table.
5. Choose **Save associations**.

Step 3: Configure local gateway connectivity

The local gateway (LGW) enables connectivity between your Outpost subnets and your on-premises network. For more information about the LGW, see Local gateway.

To provide connectivity between an instance in the Outposts subnet and your local network, you must complete the following tasks.

3a. Create a custom local gateway route table

You can create a custom route table for your local gateway (LGW) using the AWS Outposts console.

To create a custom LGW route table using the console

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route table**.
4. Choose **Create local gateway route table**.
5. (Optional) For **Name**, enter a name for your LGW route table.
6. For **Local gateway**, choose your local gateway.
7. For **Mode**, choose a mode for communication with your on-premises network.

   - Choose **Direct VPC routing** to use the private IP address of an instance.
   - Choose **CoIP** to use the customer-owned IP address.
     - (Optional) Add or remove CoIP pools and additional CIDR blocks

       [Add a CoIP pool] Choose **Add new pool** and do the following:
       - For **Name**, enter a name for your CoIP pool.
       - For **CIDR**, enter a CIDR block of customer-owned IP addresses.
     - [Add CIDR blocks] Choose **Add new CIDR** and enter a range of customer-owned IP addresses.
     - [Remove a CoIP pool or an additional CIDR block] Choose **Remove** to the right of a CIDR block or below the CoIP pool.

You can specify up to 10 CoIP pools and 100 CIDR blocks.

8. (Optional) Add or remove a tag.

   [Add a tag] Choose **Add new tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Choose **Remove** to the right of the tag's key and value.

9. Choose **Create local gateway route table**.

3b: Associate the VPC with the custom LGW route table

You must associate the VPCs with your LGW route table. They are not associated by default.

Use the following procedure to associate a VPC with a LGW route table.

You can optionally tag your association to help you identify it or categorize it according to your organization's needs.

AWS Outposts console

To associate a VPC with the custom LGW route table

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.
4. Select the route table, and then choose **Actions**, **Associate VPC**.
5. For **VPC ID**, select the VPC to associate with the local gateway route table.
6. (Optional) Add or remove a tag.

   To add a tag, choose **Add new tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   To remove a tag, choose **Remove** to the right of the tag's key and value.

7. Choose **Associate VPC**.

AWS CLI

To associate a VPC with the custom LGW route table

Use the create-local-gateway-route-table-vpc-association command.

Example

```
aws ec2 create-local-gateway-route-table-vpc-association \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
    --vpc-id vpc-07ef66ac71EXAMPLE
```

Output

```
{
    "LocalGatewayRouteTableVpcAssociation": {
        "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0ee765bcc8EXAMPLE",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
        "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
        "VpcId": "vpc-07ef66ac71EXAMPLE",
        "State": "associated"
    }
}
```

3c: Add a route entry in the Outpost subnet route table

Add a route entry in the Outpost subnet route table to enable traffic between the Outpost subnets and LGW.

Outpost subnets within a VPC, which is associated with Outpost LGW route tables, can have an additional target type of a Outpost Local gateway ID for their route tables. Consider the case where you want route traffic with a destination address of 172.16.100.0/24 to the customer network through the LGW. To do this, edit the Outpost subnet route table and add the following route with the destination network and a target of the LGW (`lgw-xxxx`).

Destination	Target
172.16.100.0/24	`lgw-id`

To add a route entry with `lgw-id` as a target in the Outpost subnet route table:

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Route tables**, and select the route table you created in 2b: Create a custom route table.
3. Choose **Actions** and then **Edit routes**.
4. To add a route, choose **Add route**.
5. For **Destination** enter the destination CIDR block to the customer network.
6. For **Target**, choose **Outpost local gateway ID**.
7. Choose **Save changes**.

3d: Associate the custom LGW route table with the LGW VIF groups

VIF groups are logical groupings of virtual interfaces (VIFs). Associate the local gateway route table with the VIF group.

To associate the custom LGW route table with the LGW VIF groups

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.
4. Choose the route table.
5. Choose the **VIF group association** tab in the details pane, and then choose **Edit VIF group association**.
6. For **VIF group settings**, select **Associate VIF group**, and choose a VIF group.
7. Choose **Save changes**.

3e: Add a route entry in the LGW route table

Edit the local gateway route table to add a static route that has the VIF Group as the target and your on-premise subnet CIDR range (or 0.0.0.0/0) as the destination.

Destination	Target
172.16.100.0/24	`VIF-Group-ID`

To add a route entry in the LGW route table

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Local gateway route table**.
3. Select the local gateway route table, and then choose **Actions**, **Edit routes**.
4. Choose **Add route**.
5. For **Destination**, enter the destination CIDR block, a single IP address, or the ID of a prefix list.
6. For **Target**, select the ID of the local gateway.
7. Choose **Save routes**.

3f: (Optional) Assign a customer-owned IP address to the instance

If you configured your Outposts in the 3a. Create a custom local gateway route table to use a customer-owned IP (CoIP) address pool, you must allocate an Elastic IP address from the CoIP address pool and associate the Elastic IP address with the instance. For more information about CoIP, see Customer-owned IP addresses.

If you configured your Outposts to use Direct VPC routing (DVR), skip this step.

Amazon VPC console

To assign a CoIP address to the instance

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate Elastic IP address**.
4. For **Network Border Group**, select the location from which the IP address is advertised.
5. For **Public IPv4 address pool**, choose **Customer owned IPv4 address pool**.
6. For **Customer owned IPv4 address pool**, select the pool that you configured.
7. Choose **Allocate**.
8. Select the Elastic IP address, and choose **Actions**, **Associate Elastic IP address**.
9. Select the instance from **Instance**, and then choose **Associate**.

AWS CLI

To assign a CoIP address to the instance

1. Use the describe-coip-pools command to retrieve information about your customer-owned address pools.

```
aws ec2 describe-coip-pools
```

The following is example output.

```
{
    "CoipPools": [
        {
            "PoolId": "ipv4pool-coip-0abcdef0123456789",
            "PoolCidrs":  [
                "192.168.0.0/16"
            ],
            "LocalGatewayRouteTableId": "lgw-rtb-0abcdef0123456789"
        }
    ]
}
```

2. Use the allocate-address command to allocate an Elastic IP address. Use the pool ID returned in the previous step.

```
aws ec2 allocate-address--address 192.0.2.128 --customer-owned-ipv4-
pool ipv4pool-coip-0abcdef0123456789
```

The following is example output.

```
{
    "CustomerOwnedIp": "192.0.2.128",
    "AllocationId": "eipalloc-02463d08ceEXAMPLE",
    "CustomerOwnedIpv4Pool": "ipv4pool-coip-0abcdef0123456789",
}
```

3. Use the associate-address command to associate the Elastic IP address with the Outpost instance. Use the allocation ID returned in the previous step.

```
aws ec2 associate-address --allocation-id eipalloc-02463d08ceEXAMPLE --network-
interface-id eni-1a2b3c4d
```

The following is example output.

```
{
    "AssociationId": "eipassoc-02463d08ceEXAMPLE",
}
```

Shared customer-owned IP address pools

If you want to use a shared customer-owned IP address pool, the pool must be shared before you start the configuration. For information about how to share a customer-owned IPv4 address, see Sharing your AWS resources in the *AWS RAM User Guide*.

Step 4: Configure the on-premises network

The Outpost establishes an external BGP peering from each Outpost Networking Device (OND) to a Customer Local Network Device (CND) to send and receive traffic from your on-premise network to the Outposts. For more information, see Local gateway BGP connectivity.

To send and receive traffic from your on-premises network to the Outpost, ensure that:

- On your customer network devices, the BGP session on the Local gateway VLAN is in an ACTIVE state from your network devices.
- For traffic going from on-premises to Outposts, ensure that you are receiving in your CND the BGP advertisements from Outposts. These BGP advertisements contain the routes that your on-premises network must use to route traffic from the on-premises to Outpost. Hence, ensure that your network has the right routing between Outposts and the on-prem resources.
- For traffic going from Outposts to on-premises network, ensure that your CNDs are sending the BGP route advertisements of on-premises network subnets to Outposts (or 0.0.0.0/0). As an alternative, you can advertise a default route (e.g. 0.0.0.0/0) to Outposts. The on-premises subnets advertised by the CNDs must have a CIDR range that is equal to or included in the CIDR range that you configured in 3e: Add a route entry in the LGW route table.

Example: BGP advertisements in Direct VPC mode

Consider the scenario where you have an Outpost, configured in Direct VPC mode, with two Outposts rack network devices connected by a local gateway VLAN to two customer local network devices. The following is configured:

- A VPC with a CIDR block 10.0.0.0/16.
- An Outpost subnet in the VPC with a CIDR block 10.0.3.0/24.
- A subnet in the on-premises network with a CIDR block 172.16.100.0/24
- Outposts uses the private IP address of the instances on the Outpost subnet, for example 10.0.3.0/24, to communicate with your on-premises network.

In this scenario, the route advertised by:

- The local gateway to your customer devices is 10.0.3.0/24.
- Your customer devices to the Outpost local gateway is 172.16.100.0/24.

As a result, the local gateway will send outbound traffic with destination network 172.16.100.0/24 to your customer devices. Ensure that your network has the correct routing configuration to deliver traffic to the destination host within your network.

For the specific commands and configuration required to check the state of the BGP sessions and the advertised routes within those sessions, see the documentation from your networking vendor. For troubleshooting, see AWS Outposts rack network troubleshooting checklist.

Example: BGP advertisements in CoIP mode

Consider the scenario where you have an Outpost with two Outposts rack network devices connected by a local gateway VLAN to two customer local network devices. The following is configured:

- A VPC with a CIDR block 10.0.0.0/16.
- A subnet in the VPC with a CIDR block 10.0.3.0/24.
- A customer-owned IP pool (10.1.0.0/26).
- An Elastic IP address association that associates 10.0.3.112 to 10.1.0.2.
- A subnet in the on-premises network with a CIDR block 172.16.100.0/24

- Communication between your Outpost and on-premises network will use the CoIP Elastic IPs to address instances in the Outpost, the VPC CIDR range is not used.

In this scenario the route advertised by:

- The local gateway to your customer devices is 10.1.0.0/26.
- Your customer devices to the Outpost local gateway is 172.16.100.0/24.

As a result the local gateway will send outbound traffic with destination network 172.16.100.0/24 to your customer devices. Ensure that your network has the right routing configuration to deliver traffic to the destination host within your network.

For the specific commands and configuration required to check the state of the BGP sessions and the advertised routes within those sessions, see the documentation from your networking vendor. For troubleshooting, see AWS Outposts rack network troubleshooting checklist.

Step 5: Launch an instance on the Outpost

You can launch EC2 instances in the Outpost subnet that you created, or in an Outpost subnet that has been shared with you. Security groups control inbound and outbound VPC traffic for instances in an Outpost subnet, just as they do for instances in an Availability Zone subnet. To connect to an EC2 instance in an Outpost subnet, you can specify a key pair when you launch the instance, just as you do for instances in an Availability Zone subnet.

Considerations

- You can create a placement groups to influence how Amazon EC2 should attempt to place groups of interdependent instances on the Outposts hardware. You can choose the placement group strategy that meets the needs of your workload.
- If your Outpost has been configured to use a customer-owned IP (CoIP) address pool, you must assign a customer-owned IP address to any instances that you launch.

To launch instances in your Outpost subnet

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions, View details**.

4. On the **Outpost summary** page, choose **Launch instance**. You are redirected to the instance launch wizard in the Amazon EC2 console. We select the Outpost subnet for you, and show you only the instance types that are supported by your Outposts rack.
5. Choose an instance type that is supported by your Outposts rack. Note that instances that appear greyed out are not available for your Outpost.
6. (Optional) To launch the instances into a placement group, expand **Advanced details** and scroll to **Placement group**. You can either select an existing placement group or create a new placement group.
7. Complete the wizard to launch the instance in your Outpost subnet. For more information, see the following in the *Amazon EC2 User Guide*:

   - Linux – Launch an instance using the new launch instance wizard
   - Windows – Launch an instance using the new launch instance wizard

> **Note**
>
> If you are creating an Amazon EBS volume, you must use the gp2 volume type or the wizard will fail.

Step 6: Test the connectivity

You can test connectivity by using the appropriate use cases.

Test connectivity from your local network to the Outpost

From a computer in your local network, run the `ping` command to the Outpost instance's private IP address.

```
ping 10.0.3.128
```

The following is example output.

```
Pinging 10.0.3.128

Reply from 10.0.3.128:  bytes=32 time=<1ms TTL=128
Reply from 10.0.3.128:  bytes=32 time=<1ms TTL=128
```

```
Reply from 10.0.3.128:  bytes=32 time=<1ms TTL=128

Ping statistics for 10.0.3.128
Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

Approximate round trip time in milliseconds
Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
```

Test the connectivity from an Outpost instance to your local network

Depending on your operating system, use **ssh** or **rdp** to connect to the private IP address of your Outpost instance. For information about connecting to a Linux instance, see Connect to your Linux instance in the *Amazon EC2 User Guide*. For information about connecting to a Windows instance, see Connect to your Windows instance in the *Amazon EC2 User Guide*.

After the instance is running, run the `ping` command to an IP address of a computer in your local network. In the following example, the IP address is 172.16.0.130.

```
ping 172.16.0.130
```

The following is example output.

```
Pinging 172.16.0.130

Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128
Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128
Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128

Ping statistics for 172.16.0.130
Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

Approximate round trip time in milliseconds
Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
```

Test connectivity between the AWS Region and the Outpost

Launch an instance in the subnet in the AWS Region. For example, use the run-instances command.

```
aws ec2 run-instances \
    --image-id ami-abcdefghi1234567898 \
    --instance-type c5.large \
```

```
    --key-name MyKeyPair \
    --security-group-ids sg-1a2b3c4d123456787 \
    --subnet-id subnet-6e7f829e123445678
```

After the instance is running, perform the following operations:

1. Get the private IP address of the instance in the AWS Region. This information is available in the Amazon EC2 console on the instance detail page.
2. Depending on your operating system, use **ssh** or **rdp** to connect to the private IP address of your Outpost instance.
3. Run the **ping** command from your Outpost instance, specifying the IP address of the instance in the AWS Region.

   ```
   ping 10.0.1.5
   ```

   The following is example output.

   ```
   Pinging 10.0.1.5

   Reply from 10.0.1.5:  bytes=32 time=<1ms TTL=128
   Reply from 10.0.1.5:  bytes=32 time=<1ms TTL=128
   Reply from 10.0.1.5:  bytes=32 time=<1ms TTL=128

   Ping statistics for 10.0.1.5
   Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

   Approximate round trip time in milliseconds
   Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
   ```

Customer-owned IP address connectivity examples

Test the connectivity from your local network to the Outpost

From a computer in your local network, run the `ping` command to the Outpost instance's customer-owned IP address.

```
ping 172.16.0.128
```

The following is example output.

```
Pinging 172.16.0.128

Reply from 172.16.0.128:  bytes=32 time=<1ms TTL=128
Reply from 172.16.0.128:  bytes=32 time=<1ms TTL=128
Reply from 172.16.0.128:  bytes=32 time=<1ms TTL=128

Ping statistics for 172.16.0.128
Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

Approximate round trip time in milliseconds
Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
```

Test the connectivity from an Outpost instance to your local network

Depending on your operating system, use **ssh** or **rdp** to connect to the private IP address of your Outpost instance. For information about connecting to a Linux instance, see Connect to your Linux instance in the *Amazon EC2 User Guide*. For information about connecting to a Windows instance, see Connect to your Windows instance in the *Amazon EC2 User Guide*.

After the Outpost instance is running, run the `ping` command to an IP address of a computer in your local network.

```
ping 172.16.0.130
```

The following is example output.

```
Pinging 172.16.0.130

Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128
Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128
Reply from 172.16.0.130:  bytes=32 time=<1ms TTL=128

Ping statistics for 172.16.0.130
Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

Approximate round trip time in milliseconds
Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
```

Test connectivity between the AWS Region and the Outpost

Launch an instance in the subnet in the AWS Region. For example, use the run-instances command.

```
aws ec2 run-instances \
    --image-id ami-abcdefghi1234567898 \
    --instance-type c5.large \
    --key-name MyKeyPair \
    --security-group-ids sg-1a2b3c4d123456787 \
    --subnet-id subnet-6e7f829e123445678
```

After the instance is running, perform the following operations:

1. Get the AWS Region instance private IP address, for example 10.0.0.5. This information is available in the Amazon EC2 console on the instance detail page.
2. Depending on your operating system, use **ssh** or **rdp** to connect to the private IP address of your Outpost instance.
3. Run the `ping` command from your Outpost instance to the AWS Region instance IP address.

   ```
   ping 10.0.0.5
   ```

   The following is example output.

   ```
   Pinging 10.0.0.5

   Reply from 10.0.0.5:  bytes=32 time=<1ms TTL=128
   Reply from 10.0.0.5:  bytes=32 time=<1ms TTL=128
   Reply from 10.0.0.5:  bytes=32 time=<1ms TTL=128

   Ping statistics for 10.0.0.5
   Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

   Approximate round trip time in milliseconds
   Minimum = 0ms,  Maximum = 0ms,  Average = 0ms
   ```

AWS Outposts connectivity to AWS Regions

AWS Outposts supports wide area network (WAN) connectivity through the service link connection.

Contents

Connectivity through service links

The service link is a necessary connection between your Outposts and your chosen AWS Region (or home Region) and allows for the management of the Outposts and the exchange of traffic to and from the AWS Region. The service link leverages an encrypted set of VPN connections to communicate with the home Region.

To set up the service link connectivity, you or AWS must configure the service link physical, virtual LAN (VLAN), and network layer connectivity with your local network devices during the Outpost provisioning. For more information, see Local network connectivity for racks and Site requirements for Outposts rack.

For the wide area network (WAN) connectivity to the AWS Region, AWS Outposts can establish service link VPN connections through the AWS Region's public connectivity. This requires the Outposts to have access to the Region's public IP ranges, which can be through the public internet or AWS Direct Connect public virtual interfaces. For the current IP address ranges, see AWS IP address ranges in the *Amazon VPC user guide*. This connectivity can be enabled by configuring specific or default (0.0.0.0/0) routes in the service link network layer path. For more information, see Service link BGP connectivity and Service link infrastructure subnet advertisement and IP range.

Alternatively, you can select the private connectivity option for your Outpost. For more information, see Service link private connectivity using VPC.

After the service link connection is established, your Outpost becomes operational and is managed by AWS. The service link is used for the following traffic:

- Customer VPC traffic between the Outpost and any associated VPCs.

- Outposts management traffic, such as resource management, resource monitoring and firmware and software updates.

Service link maximum transmission unit (MTU) requirements

The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The network must support 1500-bytes MTU between the Outpost and the service link endpoints in the parent AWS Region. For information on the required MTU between an instance in the Outpost and an instance in the AWS Region through the service link, see Network maximum transmission unit (MTU) for your Amazon EC2 instance in the *Amazon EC2 User Guide*.

Service link bandwidth recommendations

For an optimal experience and resiliency, AWS recommends that you use redundant connectivity of at least 500 Mbps (1 Gbps is better) for the service link connection to the AWS Region. You can use AWS Direct Connect or an internet connection for the service link. The minimum 500 Mbps service link connection allows you to launch Amazon EC2 instances, attach Amazon EBS volumes, and access AWS services, such as Amazon EKS, Amazon EMR, and CloudWatch metrics.

Your Outposts service link bandwidth requirements vary depending on the following characteristics:

- Number of AWS Outposts racks and capacity configurations
- Workload characteristics, such as AMI size, application elasticity, burst speed needs, and Amazon VPC traffic to the Region

To receive a custom recommendation about the service link bandwidth required for your needs, contact your AWS sales representative or APN partner.

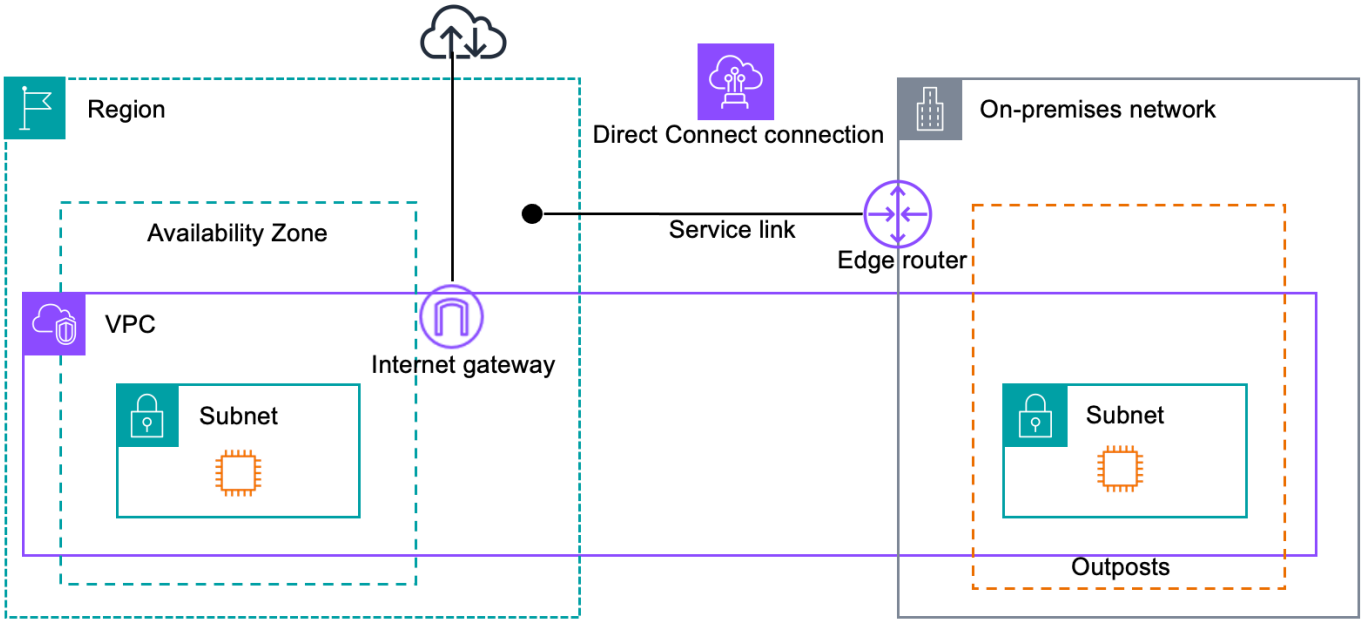
Firewalls and the service link

This section discusses firewall configurations and the service link connection.

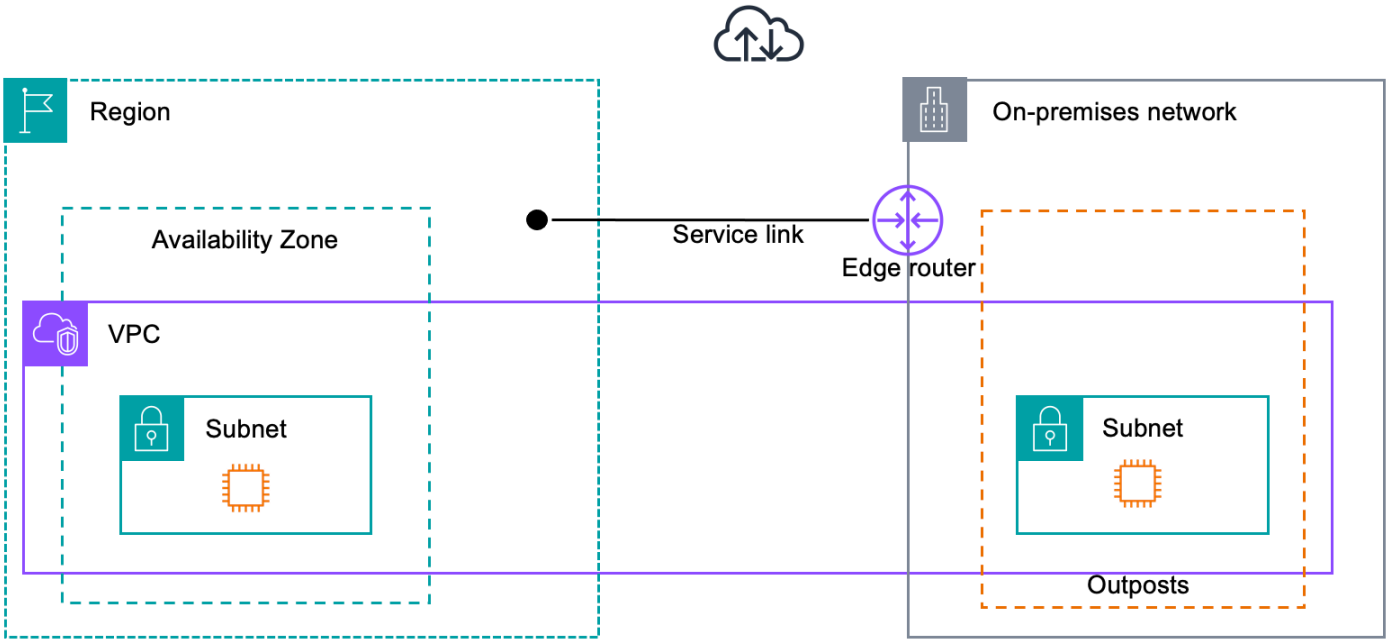
In the following diagram, the configuration extends the Amazon VPC from the AWS Region to the Outpost. An AWS Direct Connect public virtual interface is the service link connection. The following traffic goes over the service link and the AWS Direct Connect connection:

- Management traffic to the Outpost through the service link

- Traffic between the Outpost and any associated VPCs



If you are using a stateful firewall with your internet connection to limit connectivity from the public internet to the service link VLAN, you can block all inbound connections that initiate from the internet. This is because the service link VPN initiates only from the Outpost to the Region, not from the Region to the Outpost.



If you use a firewall to limit the connectivity from the service link VLAN, you can block all inbound connections. You must allow outbound connections back to the Outpost from the AWS Region as per the following table. If the firewall is stateful, outbound connections from the Outpost that are allowed, meaning that they were initiated from the Outpost, should be allowed back inbound.

Protocol	Source Port	Source Address	Destination Port	Destination Address
UDP	443	AWS Outposts service link /26	443	AWS Outposts Region's public routes
TCP	1025-65535	AWS Outposts service link /26	443	AWS Outposts Region's public routes

Note

Instances in an Outpost cannot use the service link to communicate with instances in another Outposts. Leverage routing through the local gateway or local network interface to communicate between Outposts.
AWS Outposts racks are also designed with redundant power and networking equipment, including local gateway components. For more information, see Resilience in AWS Outposts.

Service link private connectivity using VPC

You can select the private connectivity option in the console when you create your Outpost. When you do so, a service link VPN connection is established after the Outpost is installed using a VPC and subnet that you specify. This allows private connectivity by way of the VPC and minimizes public internet exposure.

Prerequisites

The following prerequisites are required before you can configure private connectivity for your Outpost:

- You must configure permissions for an IAM entity (user or role) to allow the user or role to create the service-linked role for private connectivity. The IAM entity needs permission to access the following actions:
  - `iam:CreateServiceLinkedRole` on `arn:aws:iam::*:role/aws-service-role/outposts.amazonaws.com/AWSServiceRoleForOutposts*`
  - `iam:PutRolePolicy` on `arn:aws:iam::*:role/aws-service-role/outposts.amazonaws.com/AWSServiceRoleForOutposts*`
  - `ec2:DescribeVpcs`
  - `ec2:DescribeSubnets`

  For more information, see Identity and access management (IAM) for AWS Outposts and Using service-linked roles for AWS Outposts.
- In the same AWS account and Availability Zone as your Outpost, create a VPC for the sole purpose of Outpost private connectivity with a subnet /25 or larger that does not conflict with 10.1.0.0/16. For example, you might use 10.2.0.0/16.
- Create an AWS Direct Connect connection, private virtual interface, and virtual private gateway to allow your on-premises Outpost to access the VPC. If the AWS Direct Connect connection is in a different AWS account from your VPC, see Associating a virtual private gateway across accounts in the *AWS Direct Connect User Guide*.
- Advertise the subnet CIDR to your on-premises network. You can use AWS Direct Connect to do so. For more information, see AWS Direct Connect virtual interfaces and Working with AWS Direct Connect gateways in the *AWS Direct Connect User Guide*.

You can select the private connectivity option when you create your Outpost in the AWS Outposts console. For instructions, see Create an Outpost and order Outpost capacity.

> **Note**
>
> To select the private connectivity option when your Outpost is in **PENDING** status, choose **Outposts** from the console and select your Outpost. Choose **Actions**, **Add private connectivity** and follow the steps.

After you select the private connectivity option for your Outpost, AWS Outposts automatically creates a service-linked role in your account that enables it to complete the following tasks on your behalf:

- Creates network interfaces in the subnet and VPC that you specify, and creates a security group for the network interfaces.
- Grants permission to the AWS Outposts service to attach the network interfaces to a service link endpoint instance in the account.
- Attaches the network interfaces to the service link endpoint instances from the account.

For more information about the service-linked role, see Using service-linked roles for AWS Outposts.

> **Important**
>
> After your Outpost is installed, confirm connectivity to the private IPs in your subnet from your Outpost.

Redundant internet connections

When you build connectivity from your Outpost to the AWS Region, we recommend that you create multiple connections for higher availability and resiliency. For more information, see AWS Direct Connect Resiliency Recommendations.

If you need connectivity to the public internet, you can use redundant internet connections and diverse internet providers, just as you would with your existing on-premises workloads.

Outposts and sites

Manage Outposts and sites for AWS Outposts.

You can tag Outposts and sites to help you identify them or categorize them according to your organization's needs. For more information about tagging, see Tagging AWS Resources in the *AWS General Reference Guide*.

Topics

Manage Outposts

AWS Outposts includes hardware and virtual resources known as Outposts. Use this section to create and manage Outposts, including changing the name, and adding or viewing details or tags.

To create an Outpost

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Outposts**.
4. Choose **Create Outpost**.
5. Choose a hardware type for this Outpost.
6. Enter a name and description for your Outpost.
7. Select an Availability Zone for your Outpost.
8. (Optional) Choose **Private connectivity option**. For **VPC** and **Subnet**, select a VPC and subnet in the same AWS account and Availability Zone as your Outpost.

   > **Note**
   >
   > If you need to undo the private connectivity for your Outpost, you must contact AWS Enterprise Support.

9. From **Site ID**, do one of the following:

- To select an existing site, choose the site.
- To create a new site, choose **Create site**, click **Next**, and enter the information about your site in the new window.

  After you create the site, return to this window to select the site. You may need to refresh the site list to see the new site. To refresh your data, choose the refresh icon ().

  For more information, see the section called “Sites”.

10. Choose **Create Outpost**.

> **Tip**
>
> To add capacity to your new Outpost, you must place an order.

Use the following steps to edit the name and description of an Outpost.

To edit the Outpost name and description

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Outposts**.
4. Select the Outpost, and then choose **Actions**, **Edit Outpost**.
5. Modify the name and description.

   For **Name**, enter the name.

   For **Description**, enter the description.
6. Choose **Save changes**.

Use the following steps to view the details of an Outpost.

To view the Outpost details

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.

2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Outposts**.
4. Select the Outpost, and then choose **Actions, View details**.

You can also use the AWS CLI to view Outpost details.

To view Outpost details with the AWS CLI

- Use the get-outpost AWS CLI command.

Use the following steps to manage tags on an Outpost.

To manage the Outpost tags

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Outposts**.
4. Select the Outpost, and then choose **Actions**, **Manage tags**.
5. Add or remove a tag.

   To add a tag, choose **Add new tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   To remove a tag, choose **Remove** to the right of the tag's key and value.
6. Choose **Save changes**.

Manage Outpost sites

The customer-managed physical buildings where AWS will install your Outpost. A site must meet the facility, networking, and power requirements for your Outpost. For more information, see *Requirements for Outposts racks*.

To create an Outpost site

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Sites**.
4. Choose **Create site**.
5. Choose a supported hardware type for the site.
6. Enter a name, description, and operating address for your site. If you chose to support racks at the site, enter the following information:
   - **Max weight** – Specify the maximum rack weight that this site can support.
   - **Power draw** – Specify in kVA the power draw available at the hardware placement position for the rack.
   - **Power option** – Specify the power option that you can provide for hardware.
   - **Power connector** – Specify the power connector that AWS should plan to provide for connections to the hardware.
   - **Power feed drop** – Specify whether the power feed comes above or below the rack.
   - **Uplink speed** – Specify the uplink speed the rack should support for the connection to the Region.
   - **Number of uplinks** – Specify the number of uplinks for each Outpost network device that you intend to use to connect the rack to your network.
   - **Fiber type** – Specify the type of fiber that you will use to attach the Outpost to your network.
   - **Optical standard** – Specify the type of optical standard that you will use to attach the Outpost to your network.
   - **Notes** – Specify notes about a site.
7. Read the facility requirements and choose **I have read the facility requirements**.
8. Choose **Create site**.

Use the following steps to edit an Outpost site.

To edit a site

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.

2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Sites**.
4. Select the site, and then select **Actions**, **Edit site**.
5. You can modify the name, description, operating address, and site details.

   If you change the operating address, be aware that the changes will not propagate to existing orders.
6. Choose **Save changes**.

Use the following steps to view the details of an Outpost site.

To view the site details

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Sites**.
4. Select the site, and then choose **Actions**, **View details**.

Use the following steps to manage tags on an Outpost site.

To manage the site tags

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Sites**.
4. Select the site, and then choose **Actions**, **Manage tags**.
5. Add or remove a tag.

   To add a tag, choose **Add new tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   To remove a tag, choose **Remove** to the right of the tag's key and value.
6. Choose **Save changes**.

Local gateway

The local gateway is a core component of the Outposts architecture. The local gateway enables connectivity between your Outpost subnets and your on-premises network. If the on-premise infrastructure provides an internet access, workloads running on Outposts can also leverage the local gateway to communicate with regional services or regional workloads. This connectivity can be achieved either by using a public connection (internet) or using Direct Connect. For more information, see AWS Outposts connectivity to AWS Regions.

Contents

Local gateway basics

Each Outpost supports a single local gateway. A local gateway has the following components:

- **Route tables** – You use to create local gateway route tables. For more information, see the section called “Local gateway route tables”.
- **CoIP pools** – (Optional) You can use IP address ranges that you own to facilitate communication between the on-premises network and instances in your VPC. For more information, see the section called “Customer-owned IP addresses”.
- **Virtual interfaces (VIFs)** – AWS creates one VIF for each LAG and adds both VIFs to a VIF group. The local gateway route table must have a default route to the two VIFs for local network connectivity. For more information, see *Local network connectivity*.
- **VIF group associations** – AWS adds the VIFs it creates to a VIF group. VIF groups are logical groupings of VIFs. For more information, see the section called “VIF group associations”.
- **VPC associations** – You use to create VPC associations with your VPCs and the local gateway route table. VPC route tables associated with subnets that reside on an Outpost can use the local gateway as a route target. For more information, see the section called “VPC associations”.

When AWS provisions your Outpost rack, we create some components and you are responsible for creating others.

AWS responsibilities

- Delivers the hardware.
- Creates the local gateway.
- Creates the virtual interfaces (VIFs) and a VIF group.

Your responsibilities

- Create the local gateway route table.
- Associate a VPC with the local gateway route table.
- Associate a VIF group with the local gateway route table.

Routing

The instances in your Outpost subnet can use one of the following options for communication with your on-premises network through the local gateway:

- Private IP addresses – The local gateway uses the private IP addresses of instances in your Outpost subnet to facilitate communication with your on-premises network. This is the default.
- Customer-owned IP addresses – The local gateway performs network address translation (NAT) for the customer-owned IP addresses that you assign to the instances in the Outpost subnet. This option supports overlapping CIDR ranges and other network topologies.

For more information, see the section called “Local gateway route tables”.

Connectivity through the local gateway

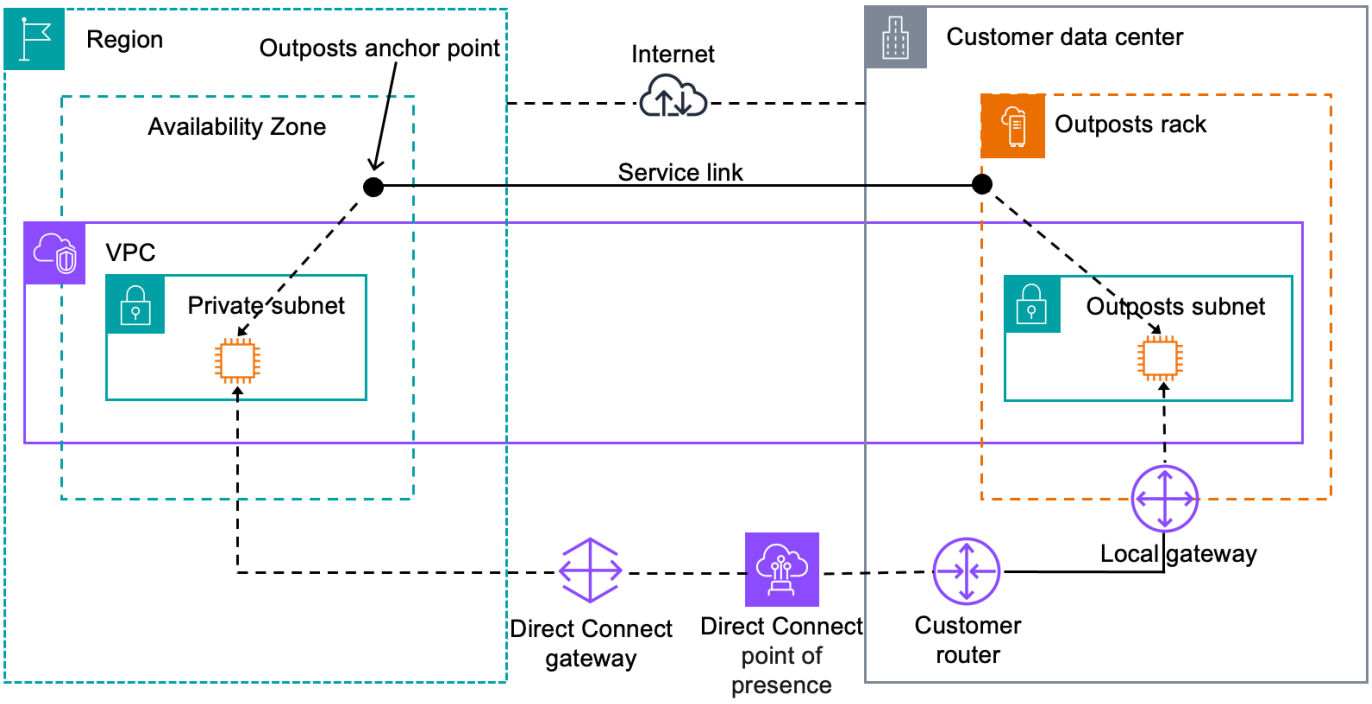
The primary role of a local gateway is to provide connectivity from an Outpost to your local on-premises network. It also provides connectivity to the internet through your on-premises network. For examples, see the section called “Direct VPC routing” and the section called “Customer-owned IP addresses”.

The local gateway can also provide a data plane path back to the AWS Region. The data plane path for the local gateway traverses from the Outpost, through the local gateway, and to your private

local gateway LAN segment. It would then follow a private path back to the AWS service endpoints in the Region. Note that the control plane path always uses the service link connectivity, regardless of the data plane path that you use.

You can connect your on-premises Outposts infrastructure to AWS services in the Region privately over AWS Direct Connect. For more information, see AWS Outposts private connectivity.

The following image shows the connectivity through the local gateway:



Local gateway route tables

Outpost subnet route tables on a rack can include a route to your on-premises network. The local gateway routes this traffic for low latency routing to the on-premises network.

By default, Outposts uses the private IP address of the instances on the Outpost to communicate with your on-premises network. This is known as *direct VPC routing for AWS Outposts* (or direct VPC routing). However, you can provide an address range, known as a *customer-owned IP address pool* (CoIP), and have instances on your network use those addresses to communicate with your on-premises network. Direct VPC routing and CoIP are mutually exclusive options and routing works differently based on your choice.

Contents

- Direct VPC routing
- Customer-owned IP addresses
- Work with local gateway route tables

Direct VPC routing

Direct VPC routing uses the private IP address of the instances in your VPC to facilitate communication with your on-premises network. These addresses are advertised to your on-premises network with BGP. Advertisement to BGP is only for the private IP addresses that belong to the subnets on your Outpost rack. This type of routing is the default mode for Outposts. In this mode, the local gateway does not perform NAT for instances, and you do not need to assign Elastic IP addresses to your EC2 instances. You have the option to use your own address space instead of direct VPC routing mode. For more information, see Customer-owned IP addresses.

Direct VPC routing is supported only for instance network interfaces. With network interfaces that AWS creates on your behalf (known as requester-managed network interfaces), their private IP addresses are not reachable from your on-premises network. For example, VPC endpoints are not directly reachable from your on-premises network.

The following examples illustrate direct VPC routing.

Examples

- Example: Internet connectivity through the VPC
- Example: Internet connectivity through the on-premises network

Example: Internet connectivity through the VPC

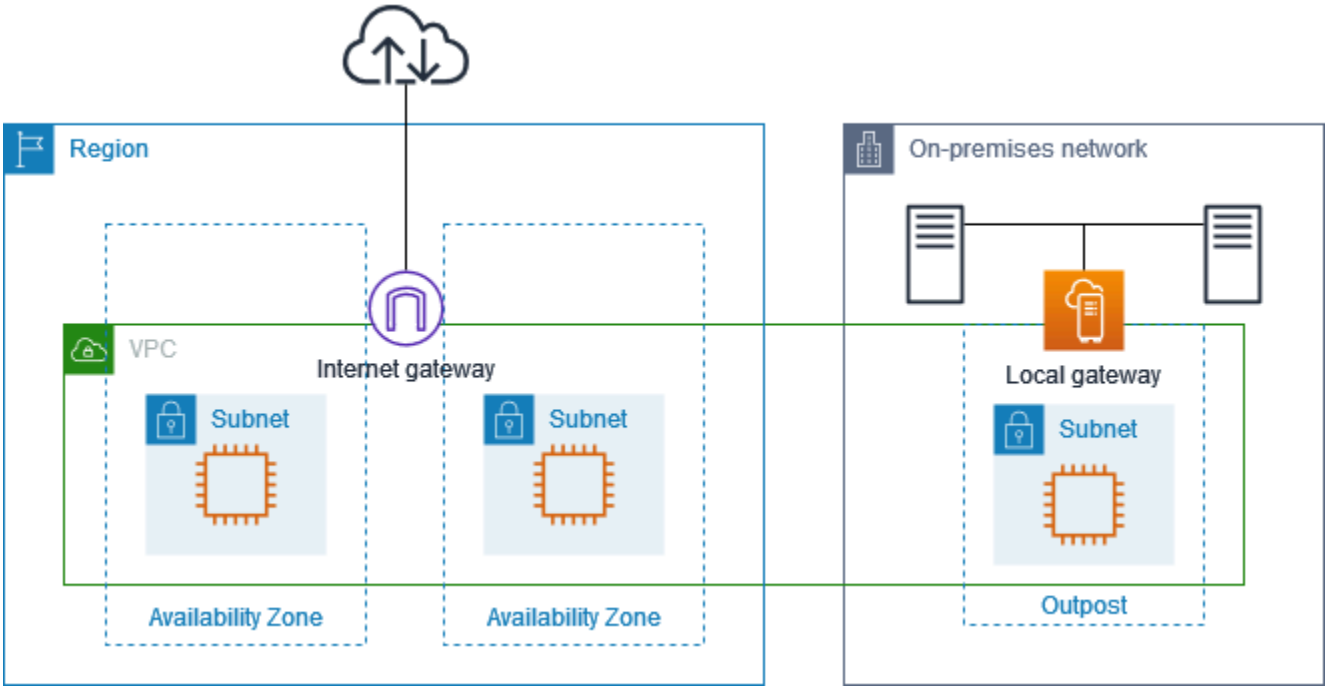
Instances in an Outpost subnet can access the internet through the internet gateway attached to the VPC.

Consider the following configuration:

- The parent VPC spans two Availability Zones and has a subnet in each Availability Zone.
- The Outpost has one subnet.
- Each subnet has an EC2 instance.
- The local gateway uses BGP advertisement to advertise the private IP addresses of the Outpost subnet to the on-premises network.

> **Note**
>
> BGP advertisement is supported only for subnets on an Outpost that have a route with the local gateway as the destination. Any other subnets are not advertised through BGP.

In the following diagram, traffic from the instance in the Outpost subnet can use the internet gateway for the VPC to access the internet.



To achieve internet connectivity through the parent Region, the route table for the Outpost subnet must have the following routes.

Destination	Target	Comments
VPC CIDR	Local	Provides connectivity between the subnets in the VPC.
0.0.0.0	*internet-gateway-id*	Sends traffic destined for the internet to the internet gateway.
on-premises network CIDR	*local-gateway-id*	Sends traffic destined for the on-premises network to the local gateway.

Example: Internet connectivity through the on-premises network

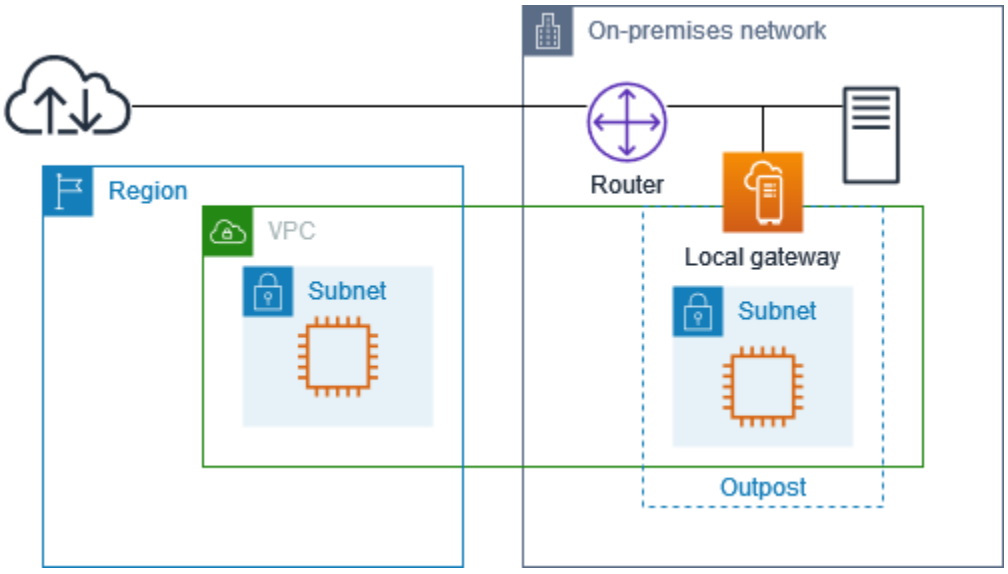
Instances in an Outpost subnet can access the internet through the on-premises network. Instances in the Outpost subnet do not need a public IP address or Elastic IP address.

Consider the following configuration:

- The Outpost subnet has an EC2 instance.
- The router in the on-premises network performs network address translation (NAT).
- The local gateway uses BGP advertisement to advertise the private IP addresses of the Outpost subnet to the on-premises network.

> **Note**
>
> BGP advertisement is supported only for subnets on an Outpost that have a route with the local gateway as the destination. Any other subnets are not advertised through BGP.

In the following diagram, traffic from the instance in the Outpost subnet can use the local gateway to access the internet or the on-premises network. Traffic from the on-premises network uses the local gateway to access the instance in the Outpost subnet.



To achieve internet connectivity through the on-premises network, the route table for the Outpost subnet must have the following routes.

Destination	Target	Comments
VPC CIDR	Local	Provides connectivity between the subnets in the VPC.
0.0.0.0/0	*local-gateway-id*	Sends traffic destined for the internet to the local gateway.

Outbound access to the internet

Traffic initiated from the instance in the Outpost subnet with a destination of the internet uses the route for 0.0.0.0/0 to route traffic to the local gateway. The local gateway sends the traffic to the router. The router uses NAT to translate the private IP address to a public IP address on the router, and then sends the traffic to the destination.

Outbound access to the on-premises network

Traffic initiated from the instance in the Outpost subnet with a destination of the on-premises network uses the route for 0.0.0.0/0 to route traffic to the local gateway. The local gateway sends the traffic to the destination in the on-premises network.

Inbound access from the on-premises network

Traffic from the on-premises network with a destination of the instance in the Outpost subnet uses the private IP address of the instance. When the traffic reaches the local gateway, the local gateway sends the traffic to the destination in the VPC.

Customer-owned IP addresses

By default, the local gateway uses the private IP addresses of instances in your VPC to facilitate communication with your on-premises network. However, you can provide an address range, known as a *customer-owned IP address pool* (CoIP), which supports overlapping CIDR ranges and other network topologies.

If you choose CoIP, you must create an address pool, assign it to the local gateway route table, and advertise these addresses back to your customer network through BGP. Any customer-owned IP Addresses associated with your local gateway route table show in the route table as propagated routes.

Customer-owned IP addresses provide local or external connectivity to resources in your on-premises network. You can assign these IP addresses to resources on your Outpost, such as EC2 instances, by allocating a new Elastic IP address from the customer-owned IP pool, and then assigning it to your resource. For more information, see the section called "3f: (Optional) Assign a customer-owned IP address to the instance".

The following requirements apply to the customer-owned IP address pool:

- You must be able to route the address in your network
- The CIDR block must be a minimum of /26

When you allocate an Elastic IP address from your customer-owned IP address pool, you continue to own the IP addresses in your customer-owned IP address pool. You are responsible for advertising them as needed on your internal networks or WAN.

You can optionally share your customer-owned pool with multiple AWS accounts in your organization using AWS Resource Access Manager. After you share the pool, participants can allocate an Elastic IP address from the customer owned IP address pool, and then assign it to an EC2 instance on the Outpost. For more information, see Sharing your AWS resources in the *AWS RAM User Guide*.

Examples

- Example: Internet connectivity through the VPC
- Example: Internet connectivity through the on-premises network

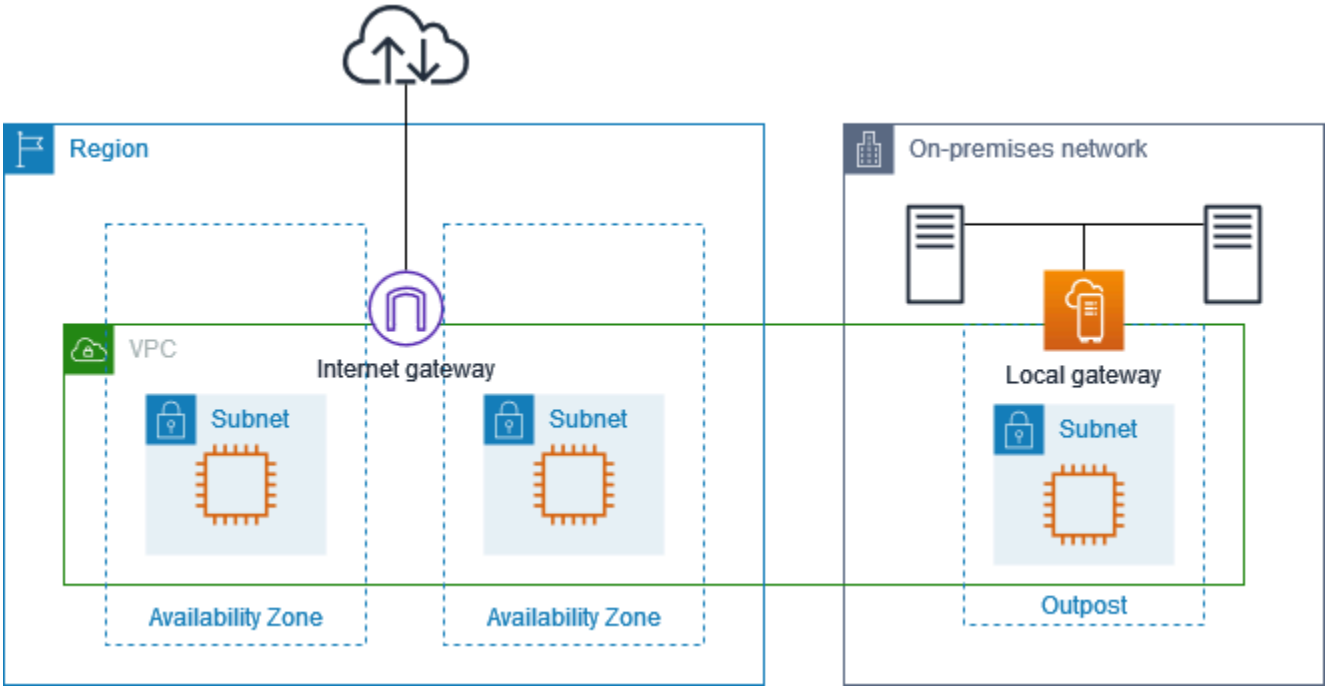
Example: Internet connectivity through the VPC

Instances in an Outpost subnet can access the internet through the internet gateway attached to the VPC.

Consider the following configuration:

- The parent VPC spans two Availability Zones and has a subnet in each Availability Zone.
- The Outpost has one subnet.
- Each subnet has an EC2 instance.
- There is a customer-owned IP address pool.

- The instance in the Outpost subnet has an Elastic IP address from the customer-owned IP address pool.
- The local gateway uses BGP advertisement to advertise the customer-owned IP address pool to the on-premises network.



To achieve internet connectivity through the Region, the route table for the Outpost subnet must have the following routes.

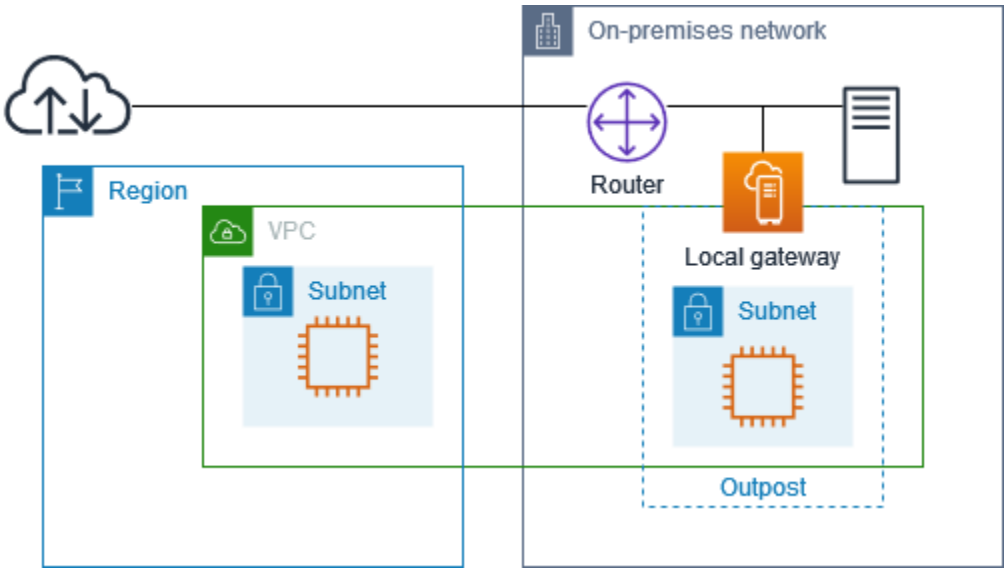
Destination	Target	Comments
VPC CIDR	Local	Provides connectivity between the subnets in the VPC.
0.0.0.0	*internet-gateway-id*	Sends traffic destined for the public internet to the internet gateway.
On-premises network CIDR	*local-gateway-id*	Sends traffic destined for the on-premises network to the local gateway.

Example: Internet connectivity through the on-premises network

Instances in an Outpost subnet can access the internet through the on-premises network.

Consider the following configuration:

- The Outpost subnet has an EC2 instance.
- There is a customer-owned IP address pool.
- The local gateway uses BGP advertisement to advertise the customer-owned IP address pool to the on-premises network.
- An Elastic IP address association that maps 10.0.3.112 to 10.1.0.2.
- The router in the customer on-premises network performs NAT.



To achieve internet connectivity through the local gateway, the route table for the Outpost subnet must have the following routes.

Destination	Target	Comments
VPC CIDR	Local	Provides connectivity between the subnets in the VPC.
0.0.0.0/0	*local-gateway-id*	Sends traffic destined for the internet to the local gateway.

Outbound access to the internet

Traffic initiated from the EC2 instance in the Outpost subnet with a destination of the internet uses the route for 0.0.0.0/0 to route traffic to the local gateway. The local gateway maps the private IP address of the instance to the customer-owned IP address, and then sends the traffic to the router. The router uses NAT to translate the customer-owned IP address to a public IP address on the router, and then sends the traffic to the destination.

Outbound access to the on-premises network

Traffic initiated from the EC2 instance in the Outpost subnet with a destination of the on-premises network uses the route for 0.0.0.0/0 to route traffic to the local gateway. The local gateway translates the IP address of the EC2 instance to the customer-owned IP address (Elastic IP address), and then sends the traffic to the destination.

Inbound access from the on-premises network

Traffic from the on-premises network with a destination of the instance in the Outpost subnet uses the customer-owned IP address (Elastic IP address) of the instance. When the traffic reaches the local gateway, the local gateway maps the customer-owned IP address (Elastic IP address) to the instance IP address, and then sends the traffic to the destination in the VPC. In addition, the local gateway route table evaluates any routes that target elastic network interfaces. If the destination address matches any static route's destination CIDR, traffic is sent to that elastic network interface. When traffic follows a static route to an elastic network interface, the destination address is preserved and is not translated to the private IP address of the network interface.

Work with local gateway route tables

As part of the rack installation, AWS creates the local gateway, configures VIFs and a VIF group. You create the local gateway route table. A local gateway route table must have an association to VIF group and a VPC. You create and manage the association of the VIF group and the VPC. Consider the following information about local gateway route tables:

- VIF groups and local gateway route tables must have a one-to-one relationship.
- The local gateway is owned by the AWS account associated with the Outpost and only the owner can modify the local gateway route table.
- You can share the local gateway route table with other AWS accounts or organizational units using AWS Resource Access Manager. For more information, see Working with shared AWS Outposts resources.

- Local gateway route tables have a mode that determines whether to use the private IP address of instances to communicate with your on-premises network (direct VPC routing) or a customer-owned IP address pool (CoIP). Direct VPC routing and CoIP are mutually exclusive options and routing works differently based on your choice. For more information, see ???.
- Direct VPC routing mode does not support overlapping CIDR ranges.

Tasks

- View local gateway route table details
- Create custom local gateway route tables
- Manage local gateway route table routes
- Manage local gateway route table tags
- Switch local gateway route table modes or delete a local gateway route table
- Manage CoIP pools
- VIF group associations
- VPC associations

View local gateway route table details

You can view the details of your local gateway route tables using the console or the AWS CLI.

AWS Outposts console

To view the local gateway route table details

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route table**.
4. Select the local gateway route table, and then choose **Actions, View details**.

AWS CLI

To view the local gateway route table details

Use the describe-local-gateway-route-tables AWS CLI command.

Example

```
aws ec2 describe-local-gateway-route-tables --region us-west-2
```

Output

```
{
    "LocalGatewayRouteTables": [
        {
            "LocalGatewayRouteTableId": "lgw-rtb-059615ef7deEXAMPLE",
            "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
            "OutpostArn": "arn:aws:outposts:us-west-2:111122223333:outpost/
op-0dc11b66edEXAMPLE",
            "State": "available",
            "Tags": []
        }
    ]
}
```

Note

If the default local gateway route table that you are viewing is using CoIP mode, then the local gateway route table is configured with a default route to each of the VIFs, and a propagated route to each associated customer-owned IP address in the pool CoIP pool.

Create custom local gateway route tables

You can create a custom route table for your local gateway using the AWS Outposts console.

To create a custom local gateway route table using the console

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route table**.
4. Choose **Create local gateway route table**.
5. (Optional) For **Name**, enter a name for your local gateway route table.
6. For **Local gateway**, choose your local gateway.

7. (Optional) Choose **Associate VIF group** and choose your **VIF group**.
8. For **Mode**, choose a mode for communication with your on-premises network.

   - Choose **Direct VPC routing** to use the private IP address of an instance.
   - Choose **CoIP** to use the customer-owned IP address.
     - (Optional) Add or remove CoIP pools and additional CIDR blocks

       [Add a CoIP pool] Choose **Add new pool** and do the following:
       - For **Name**, enter a name for your CoIP pool.
       - For **CIDR**, enter a CIDR block of customer-owned IP addresses.
     - [Add CIDR blocks] Choose **Add new CIDR** and enter a range of customer-owned IP addresses.
     - [Remove a CoIP pool or an additional CIDR block] Choose **Remove** to the right of a CIDR block or below the CoIP pool.

       You can specify up to 10 CoIP pools and 100 CIDR blocks.
9. (Optional) Add or remove a tag.

   [Add a tag] Choose **Add new tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   [Remove a tag] Choose **Remove** to the right of the tag's key and value.
10. Choose **Create local gateway route table**.

Manage local gateway route table routes

You can create local gateway route tables and inbound routes to elastic network interfaces on your Outpost. You can also modify an existing local gateway inbound route to change the target elastic network interface.

A route is in **active** status only when its target elastic network interface is attached to a running instance. If the instance is stopped or the interface is detached, the route goes from **active** to **blackhole** status.

The following requirements and limitations apply to a local gateway:

- The target elastic network interface must belong to a subnet on your Outpost and must be attached to an instance in that Outpost. A local gateway route cannot target an Amazon EC2 instance on a different Outpost or in the parent AWS Region.
- The subnet must belong to a VPC that is associated to the local gateway route table.
- You must not exceed more than 100 elastic network interface routes in the same route table.
- AWS prioritizes the most specific route, and if the routes match, we prioritize static routes over propagated routes.
- Interface VPC endpoints are not supported.
- BGP advertisement is only for subnets on an Outpost that have a route in the route table that targets the local gateway. If subnets do not have a route in the route table that targets the local gateway, then those subnets are not advertised with BGP.
- Only ENIs that are attached to Outpost instances can communicate through the local gateway for that Outpost. ENIs that belong to the Outpost subnet but attached to an instance in the Region cannot communicate through the local gateway for that Outpost.
- Managed interfaces such as VPCE endpoints or interfaces cannot be reached from on-premise through the local gateway. They can be reached only from instances that are within the Outpost.

The following NAT considerations apply.

- The local gateway does not perform NAT on traffic that matches an elastic network interface route. Instead, the destination IP address is preserved.
- Turn off source/destination checking for the target elastic network interface. For more information, see Network interface basics in the *Amazon EC2 User Guide*.
- Configure the operating system to allow traffic from the destination CIDR to be accepted on the network interface.

AWS Outposts console

To edit a local gateway route table route

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route table**.
4. Select the local gateway route table, and then choose **Actions, Edit routes**.

5. To add a route, choose **Add route**. For **Destination**, enter the destination CIDR block, a single IP address, or the ID of a prefix list.
6. To modify an existing route, for **Destination**, replace the destination CIDR block or single IP address. For **Target**, choose a target.
7. Choose **Save routes**.

AWS CLI

To create a local gateway route table route

- Use the create-local-gateway-route AWS CLI command.

Example

```
aws ec2 create-local-gateway-route \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
    --network-interface-id eni-03e612f0a1EXAMPLE \
    --destination-cidr-block 192.0.2.0/24
```

Output

```
{
    "Route": {
        "DestinationCidrBlock": "192.0.2.0/24",
        "NetworkInterfaceId": "eni-03e612f0a1EXAMPLE",
        "Type": "static",
        "State": "active",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-059615ef7dEXAMPLE",
        "OwnerId": "111122223333"
    }
}
```

To modify a local gateway route table route

You can modify the elastic network interface targeted by an existing route. To use the modify operation, the route table must already have a route with the specified destination CIDR block.

- Use the modify-local-gateway-route AWS CLI command.

Example

```
aws ec2 modify-local-gateway-route \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
    --network-interface-id eni-12a345b6c7EXAMPLE \
    --destination-cidr-block 192.0.2.0/24
```

Output

```
{
    "Route": {
        "DestinationCidrBlock": "192.0.2.0/24",
        "NetworkInterfaceId": "eni-12a345b6c7EXAMPLE",
        "Type": "static",
        "State": "active",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
        "LocalGatewayRouteTableArn": "arn:aws:ec2:us-west-2:111122223333:local-
gateway-route-table/lgw-rtb-059615ef7dEXAMPLE",
        "OwnerId": "111122223333"
    }
}
```

Manage local gateway route table tags

You can tag your local gateway route tables to help you identify them or categorize them according to your organization's needs.

To manage the local gateway route table tags

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.
4. Select the local gateway route table, and then choose **Actions, Manage tags**.
5. Add or remove a tag.

   To add a tag, choose **Add new tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

To remove a tag, choose **Remove** to the right of the tag's key and value.

6. Choose **Save changes**.

Switch local gateway route table modes or delete a local gateway route table

You must delete and recreate the local gateway route table to switch modes. Deleting the local gateway route table causes network traffic interruption.

To switch modes or delete a local gateway route table

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. Verify that you are in the correct AWS Region.

   To change the Region, use the Region selector in the top-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.
4. Verify if the local gateway route table is associated with a VIF group. If it is associated, you must remove the association between the local gateway route table and the VIF group.
   a. Choose the ID of the local gateway route table.
   b. Choose the **VIF group association** tab.
   c. If one or more VIF groups are associated with the local gateway route table, choose **Edit VIF group association**.
   d. Clear the **Associate VIF group** checkbox.
   e. Choose **Save changes**.
5. Choose **Delete local gateway route table**.
6. In the confirmation dialog box, type **`delete`** and then choose **Delete**.
7. (Optional) Create a local gateway route table with a new mode.
   a. On the navigation pane, choose **Local gateway route tables**.
   b. Choose **Create local gateway route table**.
   c. Configure the local gateway route table using the new mode. For more information, see Create custom local gateway route tables.

Manage CoIP pools

You can provide IP address ranges to facilitate communication between your on-premises network and instances in your VPC. For more information, see Customer-owned IP addresses.

Customer-owned IP pools are available for local gateway route tables in CoIP mode. To switch between local gateway route table modes, see Switch local gateway route table modes.

Use the following procedure to create a CoIP pool.

To create a CoIP pool

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.
4. Choose the route table.
5. Choose the **CoIP pools** tab in the details pane, and then choose **Create CoIP pool**.
6. (Optional) For **Name**, enter a name for your CoIP pool.
7. Choose **Add new CIDR** and enter a range of customer-owned IP addresses.
8. (Optional) Add or remove CIDR blocks

   [Add CIDR block] Choose **Add new CIDR** and enter a range of customer-owned IP addresses.

   [Remove CIDR block] Choose **Remove** to the right of a CIDR block.
9. Choose **Create CoIP pool**.

Use the following procedure to edit a CoIP pool.

To edit a CoIP pool

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.
4. Choose the route table.
5. Choose the **CoIP pools** tab in the details pane, and then choose a CoIP pool.
6. Choose **Actions, Edit CoIP pool**.

7. Choose **Add new CIDR** and enter a range of customer-owned IP addresses.
8. (Optional) Add or remove CIDR blocks

   [Add CIDR block] Choose **Add new CIDR** and enter a range of customer-owned IP addresses.

   [Remove CIDR block] Choose **Remove** to the right of a CIDR block.
9. Choose **Save changes**.

Use the following procedure to manage tags or add a name tag to a CoIP pool.

To manage tags on a CoIP pool

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.
4. Choose the route table.
5. Choose the **CoIP pools** tab in the details pane, and then choose a CoIP pool.
6. Choose **Actions, Manage tags**.
7. Add or remove a tag.

   To add a tag, choose **Add new tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   To remove a tag, choose **Remove** to the right of the tag's key and value.
8. Choose **Save changes**.

Use the following procedure to delete a CoIP pool.

To delete a CoIP pool

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.

4. Choose the route table.
5. Choose the **CoIP pools** tab in the details pane, and then choose a CoIP pool.
6. Choose **Actions, Delete CoIP pool**.
7. In the confirmation dialog box, type **`delete`** and then choose **Delete**.

VIF group associations

VIF groups are logical groupings of virtual interfaces (VIFs). You can change the local gateway route table the VIF group is associated with. When you disassociate a VIF group from a local gateway route table, you delete all routes from the route table and interrupt network traffic.

To change the association of a VIF group

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.
4. Choose the route table.
5. Choose the **VIF group association** tab in the details pane, and then choose **Edit VIF group association**.
6. For **VIF group settings**, take one of the following actions:
   - To associate the VIF group with the local gateway route table, select **Associate VIF group**, and choose a VIF group.
   - To disassociate the VIF group from the local gateway route table, clear **Associate VIF group**.

   > **Important**
   >
   > Disassociating a VIF group from the local gateway route table automatically deletes all routes and interrupts network traffic.

7. Choose **Save changes**.

VPC associations

You must associate the VPCs with your local gateway route table. They are not associated by default.

Create a VPC association

Use the following procedure to associate a VPC with a local gateway route table.

You can optionally tag your association to help you identify it or categorize it according to your organization's needs.

AWS Outposts console

To associate a VPC

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.
4. Select the route table, and then choose **Actions**, **Associate VPC**.
5. For **VPC ID**, select the VPC to associate with the local gateway route table.
6. (Optional) Add or remove a tag.

   To add a tag, choose **Add new tag** and do the following:

   - For **Key**, enter the key name.
   - For **Value**, enter the key value.

   To remove a tag, choose **Remove** to the right of the tag's key and value.
7. Choose **Associate VPC**.

AWS CLI

To associate a VPC

Use the create-local-gateway-route-table-vpc-association command.

Example

```
aws ec2 create-local-gateway-route-table-vpc-association \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
    --vpc-id vpc-07ef66ac71EXAMPLE
```

Output

```
{
    "LocalGatewayRouteTableVpcAssociation": {
        "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0ee765bcc8EXAMPLE",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
        "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
        "VpcId": "vpc-07ef66ac71EXAMPLE",
        "State": "associated"
    }
}
```

Delete a VPC association

Use the following procedure to disassociate a VPC from a local gateway route table.

AWS Outposts console

To disassociate a VPC

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.
4. Select the route table, and then choose **Actions**, **View details**.
5. In **VPC associations**, select the VPC to dissociate, and then choose **Disassociate**.
6. Choose **Disassociate**.

AWS CLI

To disassociate a VPC

Use the delete-local-gateway-route-table-vpc-association command.

Example

```
aws ec2 delete-local-gateway-route-table-vpc-association \
    --local-gateway-route-table-id lgw-rtb-059615ef7dEXAMPLE \
    --vpc-id vpc-07ef66ac71EXAMPLE
```

Output

```
{
    "LocalGatewayRouteTableVpcAssociation": {
        "LocalGatewayRouteTableVpcAssociationId": "lgw-vpc-assoc-0ee765bcc8EXAMPLE",
        "LocalGatewayRouteTableId": "lgw-rtb-059615ef7dEXAMPLE",
        "LocalGatewayId": "lgw-09b493aa7cEXAMPLE",
        "VpcId": "vpc-07ef66ac71EXAMPLE",
        "State": "associated"
    }
}
```

Local network connectivity for racks

You need the following components to connect your Outpost rack to your on-premises network:

- Physical connectivity from the Outpost patch panel to your customer local network devices.
- Link Aggregation Control Protocol (LACP) to establish two link aggregation group (LAG) connections to your Outpost network devices and to your local network devices.
- Virtual LAN (VLAN) connectivity between the Outpost and your customer local network devices.
- Layer 3 point-to-point connectivity for each VLAN.
- Border Gateway Protocol (BGP) for the route advertisement between the Outpost and your on-premises service link.
- BGP for the route advertisement between the Outpost and your on-premises local network device for connectivity to the local gateway.

Contents

Physical connectivity

An Outpost rack has two physical network devices that attach to your local network.

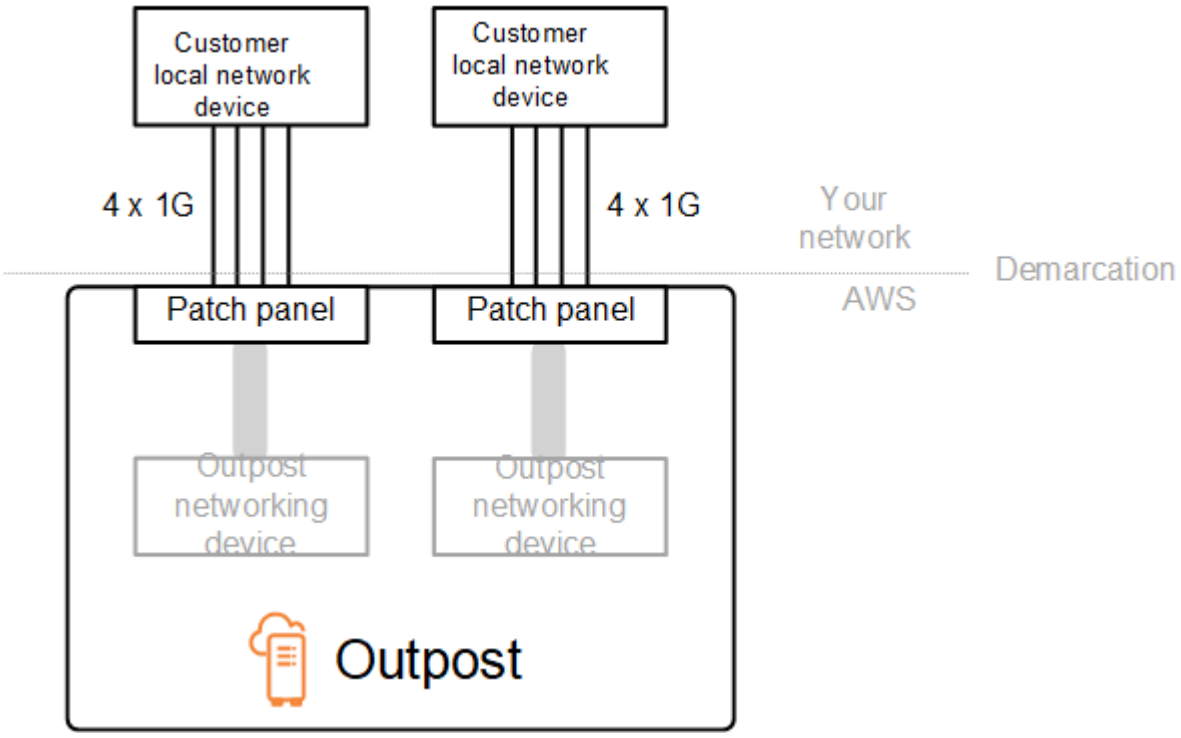
An Outpost requires a minimum of two physical links between these Outpost network devices and your local network devices. An Outpost supports the following uplink speeds and quantities for each Outpost network device.

Uplink speed	Number of uplinks
1 Gbps	1, 2, 4, 6, or 8
10 Gbps	1, 2, 4, 8, 12, or 16
40 Gbps or 100 Gbps	1, 2, or 4

The uplink speed and quantity are symmetrical on each Outpost network device. If you use 100 Gbps as the uplink speed, you must configure the link with forward error correction (FEC CL91).

Outpost racks can support single-mode fiber (SMF) with Lucent Connector (LC), multimode fiber (MMF), or MMF OM4 with LC. AWS provides the optics that are compatible with the fiber that you provide at the rack position.

In the following diagram, the physical demarcation is the fiber patch panel in each Outpost. You provide the fiber cables that are required to connect the Outpost to the patch panel.



Link aggregation

AWS Outposts uses the Link Aggregation Control Protocol (LACP) to establish two link aggregation group (LAG) connections, one from each Outpost network device to each local network device. The

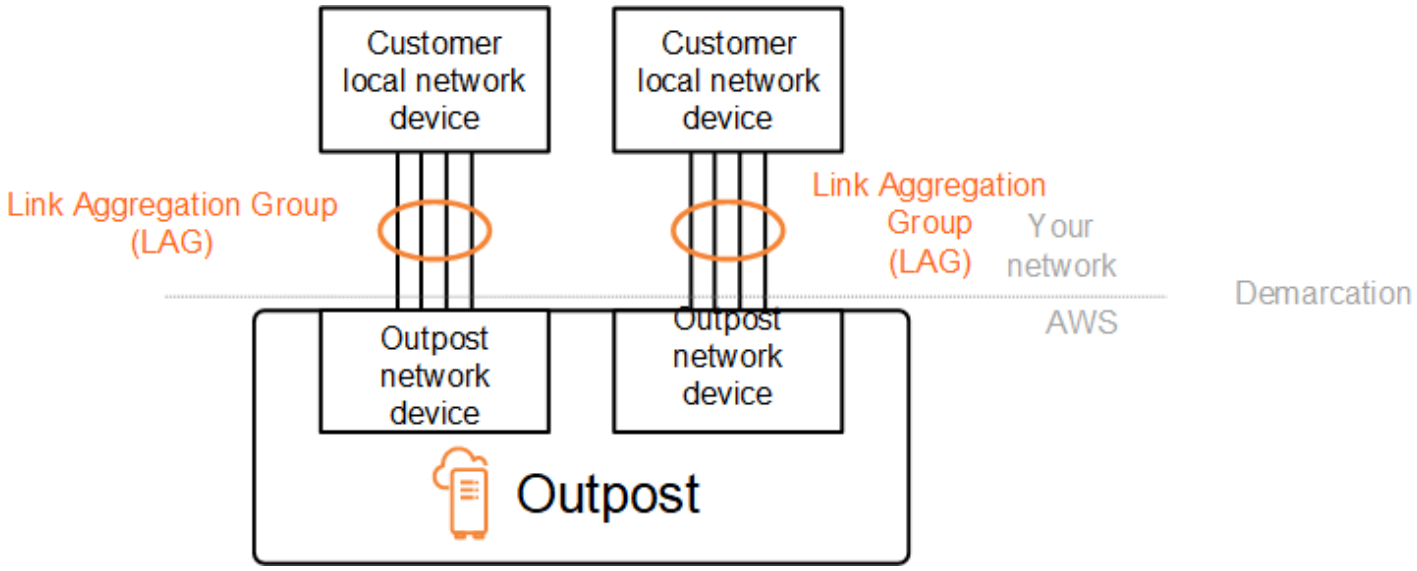
links from each Outpost network device are aggregated into an Ethernet LAG to represent a single network connection. These LAGs use LACP with standard fast timers. You can't configure LAGs to use slow timers.

To enable an Outpost installation at your site, you must configure your side of the LAG connections on your network devices.

From a logical perspective, ignore the Outpost patch panels as the demarcation point and use the Outpost networking devices.

For deployments that have multiple racks, an Outpost must have four LAGs between the aggregation layer of the Outpost network devices and your local network devices.

The following diagram shows four physical connections between each Outpost network device and its connected local network device. We use Ethernet LAGs to aggregate the physical links connecting the Outpost network devices and the customer local network devices.



Virtual LANs

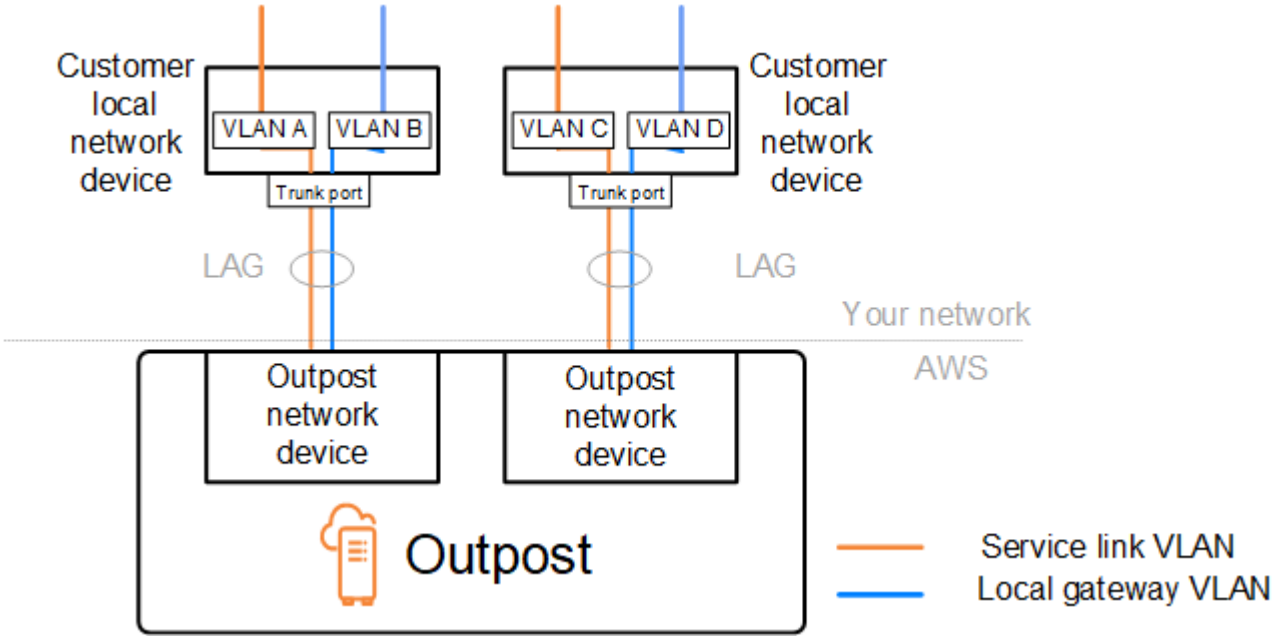
Each LAG between an Outpost network device and a local network device must be configured as an IEEE 802.1q Ethernet trunk. This enables the use of multiple VLANs for network segregation between data paths.

Each Outpost has the following VLANs to communicate with your local network devices:

- **Service link VLAN** – Enables communication between your Outpost and your local network devices in order to establish a service link path for the service link connectivity. For more information, see AWS Outposts connectivity to AWS Regions.
- **Local gateway VLAN** – Enables communication between your Outpost and your local network devices in order to establish a local gateway path to connect your Outpost subnets and your local area network. Outpost local gateway leverages this VLAN to provide your instances the connectivity to your on-premise network, which might include internet access through your network. For more information, see Local gateway.

You can configure the service link VLAN and local gateway VLAN only between the Outpost and your customer local network devices.

An Outpost is designed to separate the service link and local gateway data paths into two isolated networks. This enables you to choose which of your networks can communicate with services running on the Outpost. It also enables you to make the service link an isolated network from the local gateway network by using multiple route table on your customer local network device, commonly known as Virtual Routing and Forwarding instances (VRF). The demarcation line exists at the port of the Outpost network devices. AWS manages any infrastructure on the AWS side of the connection, and you manage any infrastructure on your side of the line.



To integrate your Outpost with your on-premises network during the installation and on-going operation, you must allocate the VLANs used between the Outpost network devices and

the customer local network devices. You need to provide this information to AWS before the installation. For more information, see the section called “Network readiness checklist”.

Network layer connectivity

To establish network layer connectivity, each Outpost network device is configured with Virtual Interfaces (VIFs) that include the IP address for each VLAN. Through these VIFs, AWS Outposts network devices can set up IP connectivity and BGP sessions with your local network equipment.

We recommend the following:

- Use a dedicated subnet, with a /30 or /31 CIDR, to represent this logical point-to-point connectivity.
- Do not bridge the VLANs between your local network devices.

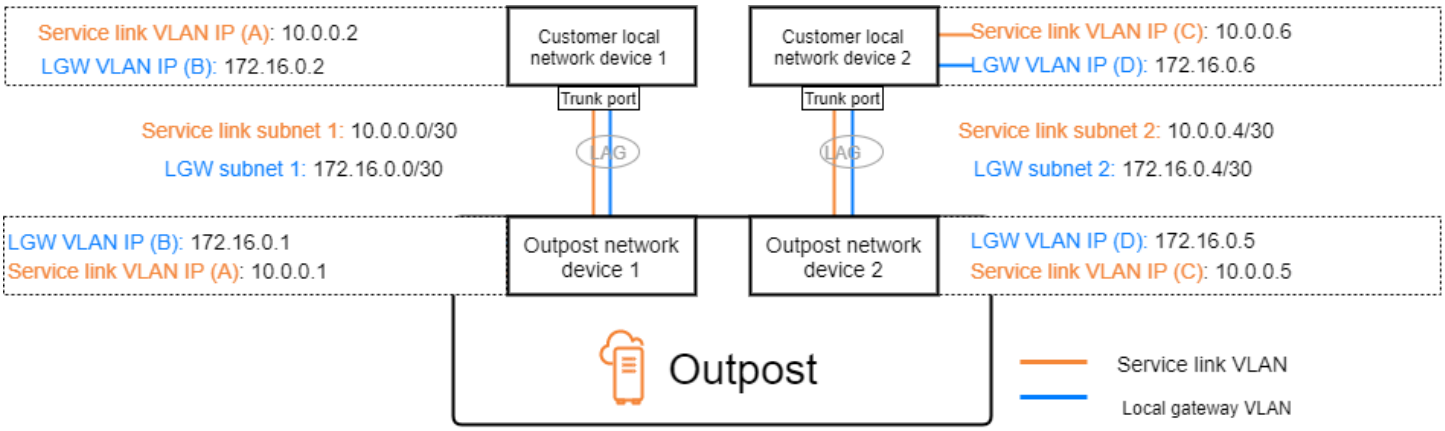
For the network layer connectivity, you must establish two paths:

- **Service link path** - To establish this path, specify a VLAN subnet with a range of /30 or /31 and an IP address for each service link VLAN on the AWS Outposts network device. Service link Virtual Interfaces (VIFs) are used for this path to establish IP connectivity and BGP sessions between your Outpost and your local network devices for service link connectivity. For more information, see AWS Outposts connectivity to AWS Regions.
- **Local gateway path** - To establish this path, specify a VLAN subnet with a range of /30 or /31 and an IP address for the local gateway VLAN on the AWS Outposts network device. Local gateway VIFs are used on this path to establish IP connectivity and BGP sessions between your Outpost and your local network devices for your local resource connectivity.

The following diagram shows the connections from each Outpost network device to the customer local network device for the service link path and the local gateway path. There are four VLANs for this example:

- VLAN A is for the service link path that connects the Outpost network device 1 with the customer local network device 1.
- VLAN B is for the local gateway path that connects the Outpost network device 1 with the customer local network device 1.
- VLAN C is for the service link path that connects the Outpost network device 2 with the customer local network device 2.

- VLAN D is for the local gateway path that connects the Outpost network device 2 with the customer local network device 2.



The following table shows example values for the subnets that connect the Outpost network device 1 with the customer local network device 1.

VLAN	Subnet	Customer Device 1 IP	AWS OND 1 IP
A	10.0.0.0/30	10.0.0.2	10.0.0.1
B	172.16.0.0/30	172.16.0.2	172.16.0.1

The following table shows example values for the subnets that connect the Outpost network device 2 with the customer local network device 2.

VLAN	Subnet	Customer Device 2 IP	AWS OND 2 IP
C	10.0.0.4/30	10.0.0.6	10.0.0.5
D	172.16.0.4/30	172.16.0.6	172.16.0.5

ACE rack connectivity

> **Note**
>
> Skip this section if you don't need an ACE rack.

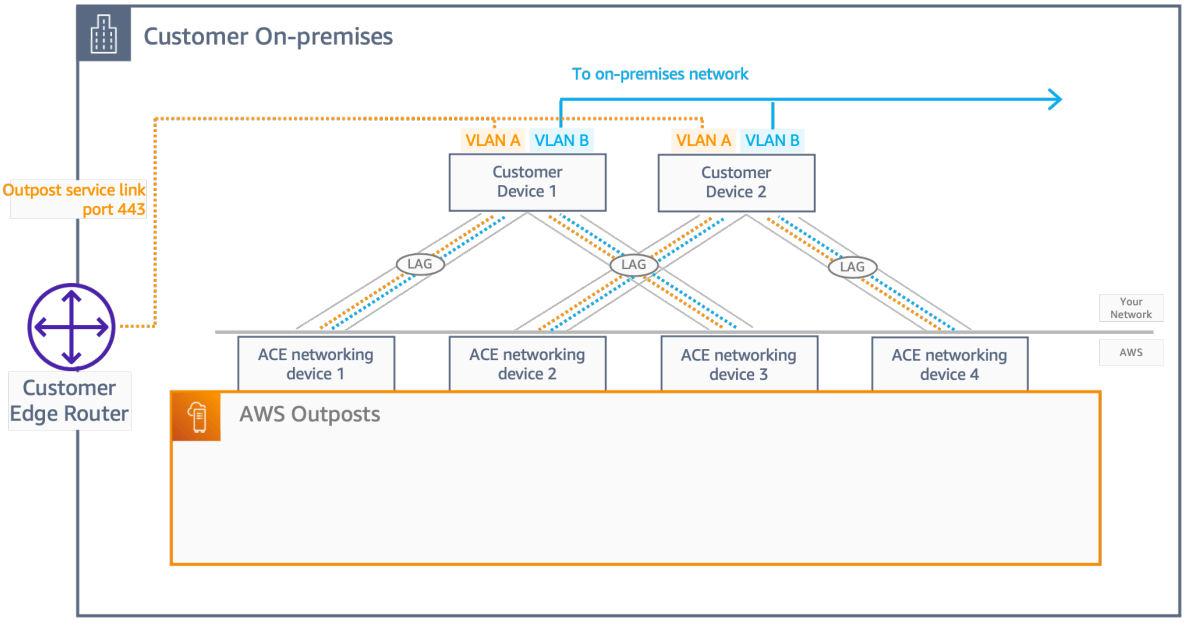
An Aggregation, Core, Edge (ACE) rack acts as a network aggregation point for multi-rack Outpost deployments. You must use an ACE rack if you have five or more compute racks. If you have less than five compute racks but plan to expand to five or more racks in the future, we recommend that you install an ACE rack at the earliest.

With an ACE rack, the Outposts networking devices are no longer directly attached to your on-premises networking devices. Instead, they are connected to the ACE rack, which provides connectivity to the Outpost racks. In this topology, AWS owns the VLAN interface allocation and configuration between Outposts networking devices and the ACE networking devices.

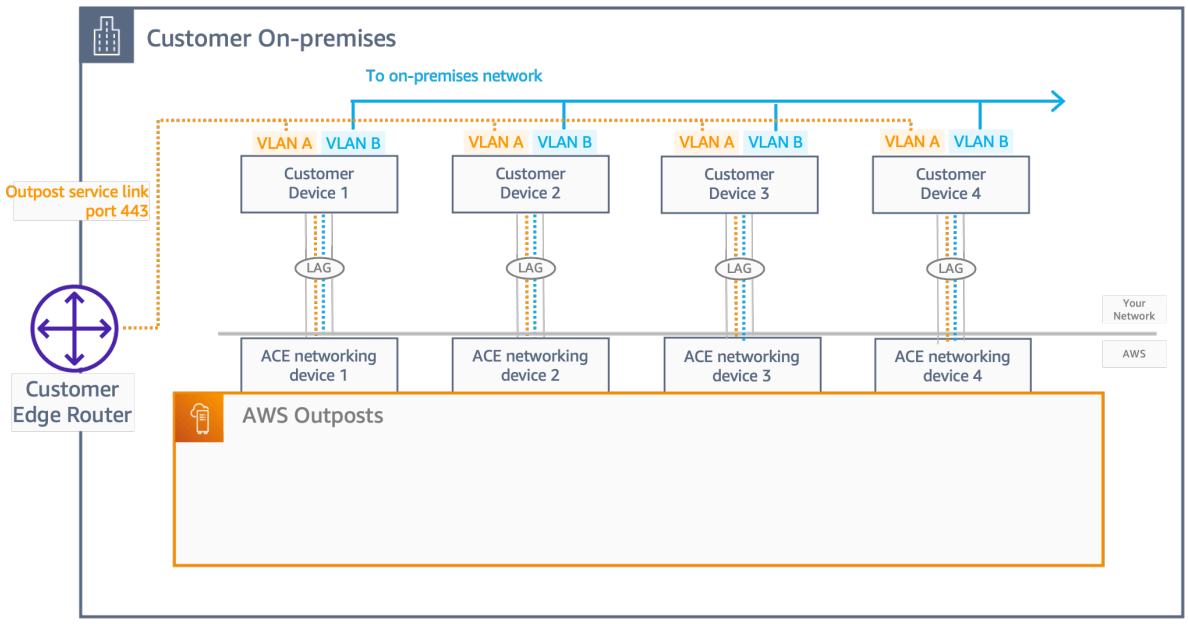
An ACE rack includes four networking devices which can be connected to two upstream customer devices in a customer on-premises network or four upstream customer devices for maximum resiliency.

The following images show the two networking topologies.

The following image shows the four ACE networking devices of the ACE rack connected to two upstream customer devices:



The following image shows the four ACE networking devices of the ACE rack connected to four upstream customer devices:



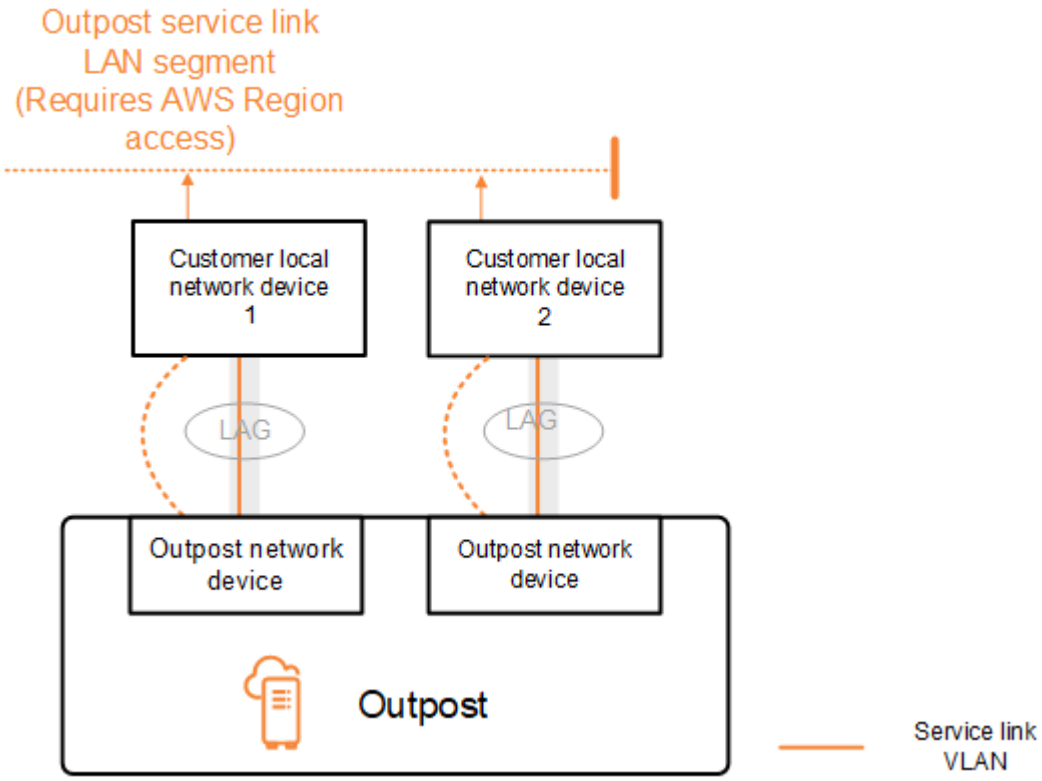
Service link BGP connectivity

The Outpost establishes an external BGP peering session between each Outpost network device and the customer local network device for service link connectivity over the service link VLAN. The BGP peering session is established between the /30 or /31 IP addresses provided for the point-to-point VLAN. Each BGP peering session uses a private Autonomous System Number (ASN) on the Outpost network device and an ASN that you choose for your customer local network devices. AWS provides the attributes as part of the installation process.

Consider the scenario where you have an Outpost with two Outpost network devices connected by a service link VLAN to two customer local network devices. You configure the following infrastructure, and customer local network device BGP ASN attributes for each service link:

- The service link BGP ASN. 2-byte (16-bit) or 4-byte (32-bit). The valid values are 64512-65535 or 4200000000-4294967294.
- The infrastructure CIDR. This must be a /26 CIDR per rack.
- The customer local network device 1 service link BGP peer IP address.
- The customer local network device 1 service link BGP peer ASN. The valid values are 1-4294967294.
- The customer local network device 2 service link BGP peer IP address.

- The customer local network device 2 service link BGP peer ASN. The valid values are 1-4294967294. For more information, see RFC4893.



The Outpost establishes an external BGP peering session over the service link VLAN using the following process:

1. Each Outpost network device uses the ASN to establish a BGP peering session with its connected local network device.
2. Outpost network devices advertise the /26 CIDR range as two /27 CIDR ranges to support link and device failures. Each OND advertises its own /27 prefix with an AS-Path length of 1, plus the /27 prefixes of all other ONDs with an AS-Path length of 4 as a backup.
3. The subnet is used for connectivity from the Outpost to the AWS Region.

We recommend that you configure customer network equipment to receive BGP advertisements from Outposts without changing the BGP attributes. The customer network should prefer routes from Outposts with an AS-Path length of 1 over routes with an AS-Path length of 4.

The customer network should advertise equal BGP prefixes with the same attributes to all ONDs. The Outpost network load balances outbound traffic between all uplinks by default. Routing

policies are used on the Outpost side to shift traffic away from an OND if maintenance is required. This traffic shift requires equal BGP prefixes from the customer side on all ONDs. If maintenance is required on the customer network, we recommend that you use AS-Path prepending to temporarily shift traffic array from specific uplinks.

Service link infrastructure subnet advertisement and IP range

You provide a /26 CIDR range during the pre-installation process for the *service link infrastructure subnet*. The Outpost infrastructure uses this range to establish connectivity to the Region through the service link. The service link subnet is the Outpost source, which initiates the connectivity.

Outpost network devices advertise the /26 CIDR range as two /27 CIDR blocks to support link and device failures.

You must provide a service link BGP ASN and an infrastructure subnet CIDR (/26) for the Outpost. For each Outpost network device, provide the BGP peering IP address on the VLAN of the local network device and the BGP ASN of the local network device.

If you have a multiple rack deployment, you must have one /26 subnet per rack.

Local gateway BGP connectivity

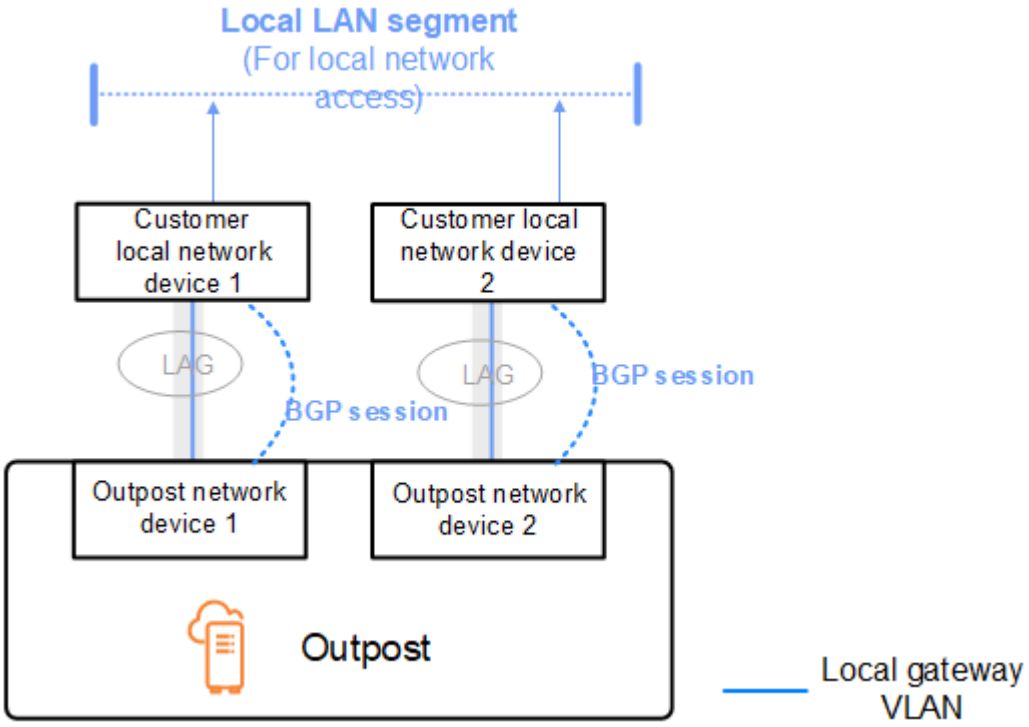
The Outpost establishes an external BGP peering from each Outpost network device to a local network device for connectivity to the local gateway. It uses a private Autonomous System Number (ASN) that you assign in order to establish the external BGP sessions. Each Outpost network device has a single external BGP peering to a local network device using its local gateway VLAN.

The Outpost establishes an external BGP peering session over the local gateway VLAN between each Outpost network device and its connected customer local network device. The peering session is established between the /30 or /31 IPs that you provided when you set up network connectivity and uses point-to-point connectivity between the Outpost network devices and customer local network devices. For more information, see the section called “Network layer connectivity”.

Each BGP session uses the private ASN on the Outpost network device side, and an ASN that you choose on the customer local network device side. AWS provides the attributes as part of the pre-installation process.

Consider the scenario where you have an Outpost with two Outpost network devices connected by a service link VLAN to two customer local network devices. You configure the following local gateway and customer local network device BGP ASN attributes for each service link:

- AWS provides the local gateway BGP ASN. 2-byte (16-bit) or 4-byte (32-bit). The valid values are 64512-65535 or 4200000000-4294967294.
- (Optional) You provide the customer owned CIDR that gets advertised (public or private, /26 minimum).
- You provide the customer local network device 1 local gateway BGP peer IP address.
- You provide the customer local network device 1 local gateway BGP peer ASN. The valid values are 1-4294967294. For more information, see RFC4893.
- You provide the customer local network device 2 local gateway BGP peer IP address.
- You provide the customer local network device 2 local gateway BGP peer ASN. The valid values are 1-4294967294. For more information, see RFC4893.



We recommend that you configure customer network equipment to receive BGP advertisements from Outposts without changing the BGP attributes, and enable BGP multipath/load balancing to achieve optimal inbound traffic flows. AS-Path prepending is used for local gateway prefixes to shift traffic away from ONDs if maintenance is required. The customer network should prefer routes from Outposts with an AS-Path length of 1 over routes with an AS-Path length of 4.

The customer network should advertise equal BGP prefixes with the same attributes to all ONDs. The Outpost network load balances outbound traffic between all uplinks by default. Routing

policies are used on the Outpost side to shift traffic away from an OND if maintenance is required. This traffic shift requires equal BGP prefixes from the customer side on all ONDs. If maintenance is required on the customer network, we recommend that you use AS-Path prepending to temporarily shift traffic array from specific uplinks.

Local gateway customer-owned IP subnet advertisement

By default, the local gateway uses the private IP addresses of instances in your VPC to facilitate communication with your on-premise network. However, you can provide a customer-owned IP address pool (CoIP).

If you choose CoIP, AWS creates the pool from information you provide during the installation process. You can create Elastic IP addresses from this pool, and then assign the addresses to resources on your Outpost, such as EC2 instances.

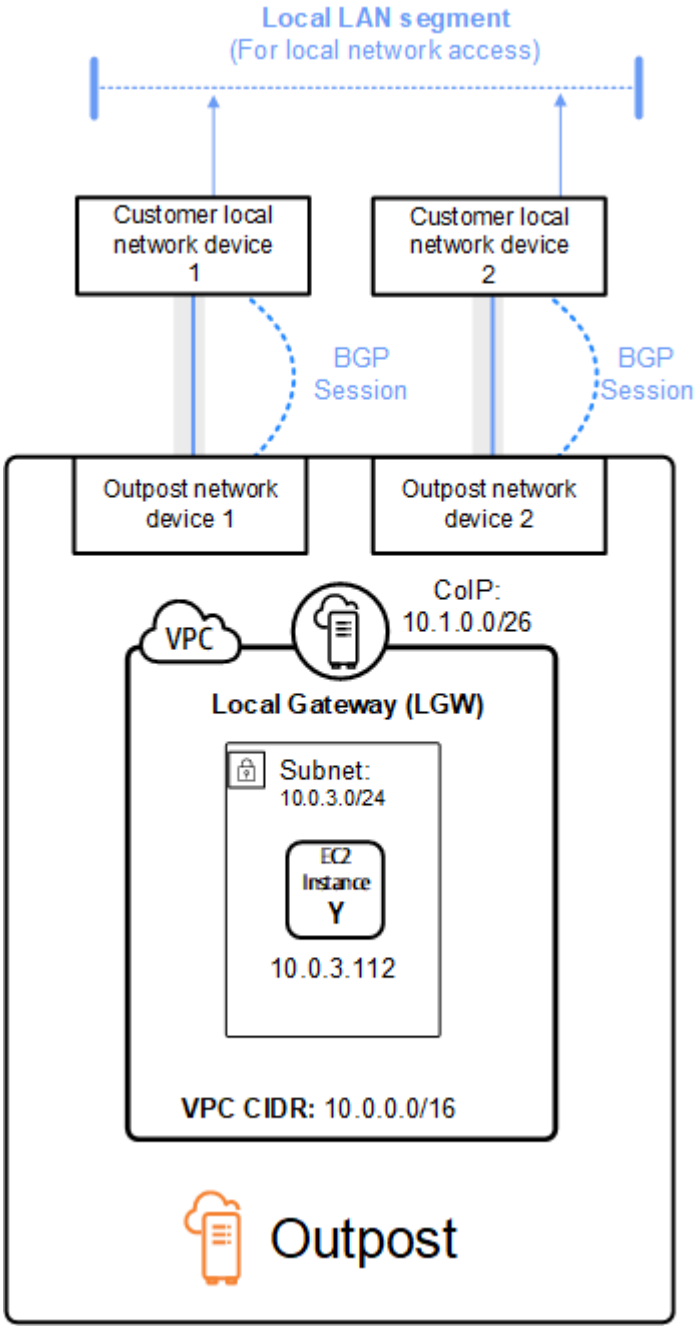
The local gateway translates the Elastic IP address to an address in the customer-owned pool. The local gateway advertises the translated address to your on-premises network, and any other network that communicates with the Outpost. The addresses are advertised on both local gateway BGP sessions to the local network devices.

> **Tip**
>
> If you are not using CoIP, then BGP advertises the private IP addresses of any subnets on your Outpost that have a route in the route table that targets the local gateway.

Consider the scenario where you have an Outpost with two Outpost network devices connected by a service link VLAN to two customer local network devices. The following is configured:

- A VPC with a CIDR block 10.0.0.0/16.
- A subnet in the VPC with a CIDR block 10.0.3.0/24.
- An EC2 instance in the subnet with a private IP address 10.0.3.112.
- A customer-owned IP pool (10.1.0.0/26).
- An Elastic IP address association that associates 10.0.3.112 to 10.1.0.2.
- A local gateway that uses BGP to advertise 10.1.0.0/26 to the on-premises network through the local devices.

- Communication between your Outpost and on-premises network will use the CoIP Elastic IPs to address instances in the Outpost, the VPC CIDR range is not used.



Working with shared AWS Outposts resources

With Outpost sharing, Outpost owners can share their Outposts and Outpost resources, including Outpost sites and subnets, with other AWS accounts under the same AWS organization. As an Outpost owner, you can create and manage Outpost resources centrally, and share the resources across multiple AWS accounts within your AWS organization. This allows other consumers to use Outpost sites, configure VPCs, and launch and run instances on the shared Outpost.

In this model, the AWS account that owns the Outpost resources (*owner*) shares the resources with other AWS accounts (*consumers*) in the same organization. Consumers can create resources on Outposts that are shared with them in the same way that they would create resources on Outposts that they create in their own account. The owner is responsible for managing the Outpost and resources that they create in it. Owners can change or revoke shared access at any time. With the exception of instances that consume Capacity Reservations, owners can also view, modify, and delete resources that consumers create on shared Outposts. Owners cannot modify instances that consumers launch into Capacity Reservations that they have shared.

Consumers are responsible for managing the resources that they create on Outposts that are shared with them, including any resources that consume Capacity Reservations. Consumers can't view or modify resources owned by other consumers or by the Outpost owner. They also can't modify Outposts that are shared with them.

An Outpost owner can share Outpost resources with:

- Specific AWS accounts inside of its organization in AWS Organizations.
- An organizational unit inside of its organization in AWS Organizations.
- Its entire organization in AWS Organizations.

Contents

- Identifying a shared Outpost resource
- Shared Outpost resource permissions
- Billing and metering
- Limitations

Shareable Outpost resources

An Outpost owner can share the Outpost resources listed in this section with consumers.

These are the resources available for Outpost rack. For server resources, see Working with shared AWS Outposts resources in the AWS Outposts User Guide for Outposts servers.

- **Allocated Dedicated Hosts** – Consumers with access to this resource can:
  - Launch and run EC2 instances on a Dedicated Host.
- **Capacity Reservations** – Consumers with access to this resource can:
  - Identify Capacity Reservations shared with them.
  - Launch and manage instances that consume Capacity Reservations.
- **Customer-owned IP address (CoIP) pools** – Consumers with access to this resource can:
  - Allocate and associate customer-owned IP addresses with instances.
- **Local gateway route tables** – Consumers with access to this resource can:
  - Create and manage VPC associations to a local gateway.
  - View configurations of local gateway route tables and virtual interfaces.
- **Outposts** – Consumers with access to this resource can:
  - Create and manage subnets on the Outpost.
  - Create and manage EBS volumes on the Outpost.
  - Use the AWS Outposts API to view information about the Outpost.
- **S3 on Outposts** – Consumers with access to this resource can:
  - Create and manage S3 buckets, access points, and endpoints on the Outpost.
- **Sites** – Consumers with access to this resource can:
  - Create, manage, and control an Outpost at the site.
- **Subnets** – Consumers with access to this resource can:
  - View information about subnets.

- Launch and run EC2 instances in subnets.

Use the Amazon VPC console to share an Outpost subnet. For more information, see Sharing a subnet in the *Amazon VPC User Guide*.

Prerequisites for sharing Outposts resources

- To share an Outpost resource with your organization or an organizational unit in AWS Organizations, you must enable sharing with AWS Organizations. For more information, see Enable Sharing with AWS Organizations in the *AWS RAM User Guide*.
- To share an Outpost resource, you must own it in your AWS account. You cannot share an Outpost resource that has been shared with you.
- To share an Outpost resource, you must share it with an account that is within your organization.

Related services

Outpost resource sharing integrates with AWS Resource Access Manager (AWS RAM). AWS RAM is a service that enables you to share your AWS resources with any AWS account or through AWS Organizations. With AWS RAM, you share resources that you own by creating a *resource share*. A resource share specifies the resources to share, and the consumers with whom to share them. Consumers can be individual AWS accounts, organizational units, or an entire organization in AWS Organizations.

For more information about AWS RAM, see the *AWS RAM User Guide*.

Sharing across Availability Zones

To ensure that resources are distributed across the Availability Zones for a Region, we independently map Availability Zones to names for each account. This could lead to Availability Zone naming differences across accounts. For example, the Availability Zone `us-east-1a` for your AWS account might not have the same location as `us-east-1a` for another AWS account.

To identify the location of your Outpost resource relative to your accounts, you must use the *Availability Zone ID* (AZ ID). The AZ ID is a unique and consistent identifier for an Availability Zone across all AWS accounts. For example, `use1-az1` is an AZ ID for the `us-east-1` Region and it is the same location in every AWS account.

To view the AZ IDs for the Availability Zones in your account

1. Open the AWS RAM console at https://console.aws.amazon.com/ram.
2. The AZ IDs for the current Region are displayed in the **Your AZ ID** panel on the right-hand side of the screen.

> **Note**
>
> Local gateway route tables are in the same AZ as their Outpost, so you do not need to specify an AZ ID for route tables.

Sharing an Outpost resource

When an owner shares an Outpost with a consumer, the consumer can create resources on the Outpost in the same way that they would create resources on Outposts that they create in their own account. Consumers with access to shared local gateway route tables can create and manage VPC associations. For more information, see Shareable Outpost resources.

To share an Outpost resource, you must add it to a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. A resource share specifies the resources to share, and the consumers with whom they are shared. When you share an Outpost resource using the AWS Outposts console, you add it to an existing resource share. To add the Outpost resource to a new resource share, you must first create the resource share using the AWS RAM console.

If you are part of an organization in AWS Organizations and sharing within your organization is enabled, you can grant consumers in your organization access from the AWS RAM console to the shared Outpost resource. Otherwise, consumers receive an invitation to join the resource share and are granted access to the shared Outpost resource after accepting the invitation.

You can share an Outpost resource that you own using the AWS Outposts console, AWS RAM console, or the AWS CLI.

To share an Outpost that you own using the AWS Outposts console

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.

3. Select the Outpost, and then choose **Actions**, **View details**.
4. On the **Outpost summary** page, choose **Resource shares**.
5. Choose **Create resource share**.

You are redirected to the AWS RAM console to finish sharing the Outpost using the following procedure. To share a local gateway route table that you own, use the following procedure as well.

To share an Outpost or local gateway route table that you own using the AWS RAM console

See Creating a Resource Share in the *AWS RAM User Guide*.

To share an Outpost or local gateway route table that you own using the AWS CLI

Use the create-resource-share command.

Unsharing a shared Outpost resource

When a shared Outpost is unshared, consumers can no longer view the Outpost in the AWS Outposts console. They cannot create new subnets on the Outpost, create new EBS volumes on the Outpost, or view the Outpost details and instance types using the AWS Outposts console or the AWS CLI. Existing subnets, volumes, or instances created by consumers are not deleted. Any existing subnets consumers created on the Outpost can still be used to launch new instances.

When a shared local gateway route table is unshared, consumers can no longer create new VPC associations to it. Any existing VPC associations consumers created remain associated with the route table. Resources in these VPCs can continue to route traffic to the local gateway.

To unshare a shared Outpost resource that you own, you must remove it from the resource share. You can do this using the AWS RAM console or the AWS CLI.

To unshare a shared Outpost resource that you own using the AWS RAM console

See Updating a Resource Share in the *AWS RAM User Guide*.

To unshare a shared Outpost resource that you own using the AWS CLI

Use the disassociate-resource-share command.

Identifying a shared Outpost resource

Owners and consumers can identify shared Outposts using the AWS Outposts console and AWS CLI. They can identify shared local gateway route tables using the AWS CLI.

To identify a shared Outpost using the AWS Outposts console

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions**, **View details**.
4. On the **Outpost summary** page, view the **Owner ID** to identify the AWS account ID of the Outpost owner.

To identify a shared Outpost resource using the AWS CLI

Use the list-outposts and describe-local-gateway-route-tables commands. These commands return the Outpost resources that you own and Outpost resources that are shared with you. `OwnerId` shows the AWS account ID of the Outpost resource owner.

Shared Outpost resource permissions

Permissions for owners

Owners are responsible for managing the Outpost and resources that they create in it. Owners can change or revoke shared access at any time. They can use AWS Organizations to view, modify, and delete resources that consumers create on shared Outposts.

Permissions for consumers

Consumers can create resources on Outposts that are shared with them in the same way that they would create resources on Outposts that they create in their own account. Consumers are responsible for managing the resources that they launch onto Outposts that are shared with them. Consumers can't view or modify resources owned by other consumers or by the Outpost owner, and they can't modify Outposts that are shared with them.

Billing and metering

Owners are billed for Outposts and Outpost resources that they share. They are also billed for any data transfer charges associated with their Outpost's service link VPN traffic from the AWS Region.

There are no additional charges for sharing local gateway route tables. For shared subnets, the VPC owner is billed for VPC-level resources such as AWS Direct Connect and VPN connections, NAT gateways, and Private Link connections.

Consumers are billed for application resources that they create on shared Outposts, such as load balancers and Amazon RDS databases. Consumers are also billed for chargeable data transfers from the AWS Region.

Limitations

The following limitations apply to working with AWS Outposts sharing:

- Limitations for shared subnets apply to working with AWS Outposts sharing. For more information about VPC sharing limits, see Limitations in the *Amazon Virtual Private Cloud User Guide*.
- Service quotas apply per individual account.

Security in AWS Outposts

Security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security of the cloud and security in the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to AWS Outposts, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You are also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

For more information about security and compliance for AWS Outposts, see the AWS Outposts rack FAQ.

This documentation helps you understand how to apply the shared responsibility model when using AWS Outposts. It shows you how to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your resources.

Contents

Data protection in AWS Outposts

The AWS shared responsibility model applies to data protection in AWS Outposts. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. This content includes the security configuration and management tasks for the AWS services that you use.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties.

For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

Encryption at rest

With AWS Outposts, all data is encrypted at rest. The key material is wrapped to an external key stored in a removable device, the Nitro Security Key (NSK). The NSK is required to decrypt the data on your Outpost rack.

You can use Amazon EBS encryption for your EBS volumes and snapshots. Amazon EBS encryption uses AWS Key Management Service (AWS KMS) and KMS keys. For more information, see Amazon EBS Encryption in the *Amazon EC2 User Guide*.

Encryption in transit

AWS encrypts in-transit data between your Outpost and its AWS Region. For more information, see Connectivity through service links.

You can use an encryption protocol, such as Transport Layer Security (TLS), to encrypt sensitive data in transit through the local gateway to your local network.

Data deletion

When you stop or terminate an EC2 instance, the memory allocated to it is scrubbed (set to zero) by the hypervisor before it is allocated to a new instance, and every block of storage is reset.

Destroying the Nitro Security Key cryptographically shreds the data on your Outpost.

Identity and access management (IAM) for AWS Outposts

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be authenticated (signed in) and authorized (have permissions) to use AWS Outposts resources. You can use IAM for no additional charge.

Contents

How AWS Outposts works with IAM

Before you use IAM to manage access to AWS Outposts, learn what IAM features are available to use with AWS Outposts.

IAM features you can use with AWS Outposts

IAM feature	AWS Outposts support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys (service-specific)	Yes
ACLs	No
ABAC (tags in policies)	Yes
Temporary credentials	Yes

IAM feature	AWS Outposts support
Principal permissions	Yes
Service roles	No
Service-linked roles	Yes

Identity-based policies for AWS Outposts

Supports identity-based policies	Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for AWS Outposts

To view examples of AWS Outposts identity-based policies, see AWS Outposts policy examples.

Resource-based policies within AWS Outposts

Supports resource-based policies	No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal

in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for AWS Outposts

Supports policy actions	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of AWS Outposts actions, see Actions defined by AWS Outposts in the *Service Authorization Reference*.

Policy actions in AWS Outposts use the following prefix before the action:

```
outposts
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
```

```
    "outposts:action1",
    "outposts:action2"
]
```

You can specify multiple actions using wildcards (*). For example, to specify all actions that begin with the word `List`, include the following action:

```
"Action": "outposts:List*"
```

Policy resources for AWS Outposts

Supports policy resources	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

Some AWS Outposts API actions support multiple resources. To specify multiple resources in a single statement, separate the ARNs with commas.

```
"Resource": [
    "resource1",
    "resource2"
]
```

To see a list of AWS Outposts resource types and their ARNs, see Resource types defined by AWS Outposts in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by AWS Outposts.

Policy condition keys for AWS Outposts

Supports service-specific policy condition keys	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of AWS Outposts condition keys, see Condition keys for AWS Outposts in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by AWS Outposts.

To view examples of AWS Outposts identity-based policies, see AWS Outposts policy examples.

ACLs in AWS Outposts

Supports ACLs	No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

ABAC with AWS Outposts

Supports ABAC (tags in policies)	Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see What is ABAC? in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using temporary credentials with AWS Outposts

Supports temporary credentials	Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then

switch roles. For more information about switching roles, see Switching to a role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for AWS Outposts

Supports forward access sessions (FAS)	Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service roles for AWS Outposts

Supports service roles	No

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

Service-linked roles for AWS Outposts

Supports service-linked roles	Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS

account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing AWS Outposts service-linked roles, see Using service-linked roles for AWS Outposts.

AWS Outposts policy examples

By default, users and roles don't have permission to create or modify AWS Outposts resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Creating IAM policies in the *IAM User Guide*.

For details about actions and resource types defined by AWS Outposts, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for AWS Outposts in the *Service Authorization Reference*.

Contents

Policy best practices

Identity-based policies determine whether someone can create, access, or delete AWS Outposts resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on

specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.

- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Example: Using resource-level permissions

The following example uses resource-level permissions to grant permission to get information about the specified Outpost.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "outposts:GetOutpost",
      "Resource": "arn:aws:outposts:region:12345678012:outpost/op-1234567890abcdef0"
    }
  ]
}
```

The following example uses resource-level permissions to grant permission to get information about the specified site.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "outposts:GetSite",
      "Resource": "arn:aws:outposts:region:12345678012:site/os-0abcdef1234567890"
    }
  ]
}
```

Using service-linked roles for AWS Outposts

AWS Outposts uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to AWS Outposts. Service-linked roles are predefined by AWS Outposts and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes setting up your AWS Outposts more efficient because you don't have to manually add the necessary permissions. AWS Outposts defines the permissions of its service-linked roles, and unless defined otherwise, only AWS Outposts can assume its roles. The defined permissions include the trust policy and the permissions policy, and that permissions policy cannot be attached to any other IAM entity.

You can delete a service-linked role only after first deleting the related resources. This protects your AWS Outposts resources because you can't inadvertently remove permission to access the resources.

For information about other services that support service-linked roles, see AWS Services That Work with IAM and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

Service-linked role permissions for AWS Outposts

AWS Outposts uses the service-linked role named **AWSServiceRoleForOutposts_*OutpostID*** – Allows Outposts to access AWS resources for private connectivity on your behalf. This service-

linked role allows private connectivity configuration, creates network interfaces, and attaches them to service link endpoint instances.

The AWSServiceRoleForOutposts_*OutpostID* service-linked role trusts the following services to assume the role:

- `outposts.amazonaws.com`

The **AWSServiceRoleForOutposts_*OutpostID*** service-linked role includes the following policies:

- **AWSOutpostsServiceRolePolicy**
- **AWSOutpostsPrivateConnectivityPolicy_*OutpostID***

The **AWSOutpostsServiceRolePolicy** policy is a service-linked role policy to enable access to AWS resources managed by AWS Outposts.

This policy allows AWS Outposts to complete the following actions on the specified resources:

- Action: `ec2:DescribeNetworkInterfaces` on `all AWS resources`
- Action: `ec2:DescribeSecurityGroups` on `all AWS resources`
- Action: `ec2:CreateSecurityGroup` on `all AWS resources`
- Action: `ec2:CreateNetworkInterface` on `all AWS resources`

The **AWSOutpostsPrivateConnectivityPolicy_*OutpostID*** policy allows AWS Outposts to complete the following actions on the specified resources:

- Action: `ec2:AuthorizeSecurityGroupIngress` on `all AWS resources that match the following Condition:`

```
{ "StringLike" : { "ec2:ResourceTag/outposts:private-connectivity-resourceId" :
 "OutpostID" }} and { "StringEquals" : { "ec2:Vpc" : "vpcArn" }}
```

- Action: `ec2:AuthorizeSecurityGroupEgress` on `all AWS resources that match the following Condition:`

```
{ "StringLike" : { "ec2:ResourceTag/outposts:private-connectivity-resourceId" :
 "OutpostID" }} and { "StringEquals" : { "ec2:Vpc" : "vpcArn" }}
```

- Action: `ec2:CreateNetworkInterfacePermission` on `all AWS resources that match the following Condition:`

```
{ "StringLike" : { "ec2:ResourceTag/outposts:private-connectivity-resourceId" :
 "OutpostID" }} and { "StringEquals" : { "ec2:Vpc" : "vpcArn" }}
```

- Action: `ec2:CreateTags` on `all AWS resources that match the following Condition:`

```
{ "StringLike" : { "aws:RequestTag/outposts:private-connectivity-resourceId" :
 "{{OutpostId}}*"}}
```

You must configure permissions to allow an IAM entity (such as a user, group, or role) to create, edit, or delete a service-linked role. For more information, see Service-Linked Role Permissions in the *IAM User Guide*.

Creating a service-linked role for AWS Outposts

You don't need to manually create a service-linked role. When you configure private connectivity for your Outpost in the AWS Management Console, AWS Outposts creates the service-linked role for you.

For more information, see Service link private connectivity using VPC.

Editing a service-linked role for AWS Outposts

AWS Outposts does not allow you to edit the AWSServiceRoleForOutposts_*OutpostID* service-linked role. After you create a service-linked role, you cannot change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a Service-Linked Role in the *IAM User Guide*.

Deleting a service-linked role for AWS Outposts

If you no longer require a feature or service that requires a service-linked role, we recommend that you delete that role. That way you avoid having an unused entity that is not actively monitored or maintained. However, you must clean up the resources for your service-linked role before you can manually delete it.

> **Note**
>
> If the AWS Outposts service is using the role when you try to delete the resources, then the deletion might fail. If that happens, wait for a few minutes and try the operation again.

> **Warning**
>
> You must delete your Outpost before you can delete the AWSServiceRoleForOutposts_*OutpostID* service-linked role. The following procedure deletes your Outpost.

Before you begin, make sure that your Outpost is not being shared using AWS Resource Access Manager (AWS RAM). For more information, see Unsharing a shared Outpost resource.

To delete AWS Outposts resources used by the AWSServiceRoleForOutposts_*OutpostID*

- Contact AWS Enterprise Support to delete your Outpost.

To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the AWSServiceRoleForOutposts_*OutpostID* service-linked role. For more information, see Deleting a Service-Linked Role in the *IAM User Guide*.

Supported Regions for AWS Outposts service-linked roles

AWS Outposts supports using service-linked roles in all of the Regions where the service is available. For more information, see AWS Outposts endpoints and quotas.

AWS managed policies for AWS Outposts

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you

reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

AWS managed policy: AWSOutpostsServiceRolePolicy

This policy is attached to a service-linked role that allows AWS Outposts to perform actions on your behalf. For more information, see Using service-linked roles.

AWS managed policy: AWSOutpostsPrivateConnectivityPolicy

This policy is attached to a service-linked role that allows AWS Outposts to perform actions on your behalf. For more information, see Using service-linked roles.

AWS Outposts updates to AWS managed policies

View details about updates to AWS managed policies for AWS Outposts since this service began tracking these changes.

Change	Description	Date
AWS Outposts started tracking changes	AWS Outposts started tracking changes for its AWS managed policies.	December 03, 2019

Infrastructure security in AWS Outposts

As a managed service, AWS Outposts is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access AWS Outposts through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

For more information about the infrastructure security provided for the EC2 instances and EBS volumes running on your Outpost, see Infrastructure Security in Amazon EC2.

VPC Flow Logs function the same way as they do in an AWS Region. This means that they can be published to CloudWatch Logs, Amazon S3, or to Amazon GuardDuty for analysis. Data needs to be sent back to the Region for publication to these services, so it is not visible from CloudWatch or other services when the Outpost is in a disconnected state.

Tamper monitoring on AWS Outposts equipment

Ensure that no one modifies, alters, reverse engineers, or tampers with the AWS Outposts equipment. AWS Outposts equipment may be equipped with tamper monitoring to ensure compliance with the AWS Service Terms.

Resilience in AWS Outposts

AWS Outposts is designed to be highly available. Outpost racks are designed with redundant power and networking equipment. For additional resilience, we recommend that you provide dual power sources and redundant network connectivity for your Outpost.

For high availability, you can provision additional built-in and always active capacity on Outposts rack. Outpost capacity configurations are designed to operate in production environments, and support N+1 instances for each instance family when you provision the capacity to do so. AWS recommends that you allocate sufficient additional capacity for your mission-critical applications to enable recovery and failover if there is an underlying host issue. You can use the Amazon

CloudWatch capacity availability metrics and set alarms to monitor the health of your applications, create CloudWatch actions to configure automatic recovery options, and monitor the capacity utilization of your Outposts over time.

When you create an Outpost, you select an Availability Zone from an AWS Region. This Availability Zone supports control plane operations such as responding to API calls, monitoring the Outpost, and updating the Outpost. To benefit from the resiliency provided by Availability Zones, you can deploy applications on multiple Outposts, each attached to a different Availability Zone. This enables you to build additional application resilience and avoid a dependence on a single Availability Zone. For more information about Regions and Availability Zones, see AWS Global Infrastructure.

You can use a placement group with a spread strategy to ensure that instances are placed on distinct Outposts racks. By doing so, this can help reduce correlated failures. For more information, see Placement groups on Outposts.

You can launch instances in Outposts using Amazon EC2 Auto Scaling and create an Application Load Balancer to distribute traffic between the instances. For more information, see Configuring an Application Load Balancer on AWS Outposts.

Compliance validation for AWS Outposts

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

> **Note**
>
> Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.

- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Internet access for AWS Outposts workloads

This section explains how AWS Outposts workloads can access the internet in the following ways:

- Through the parent AWS Region
- Through your local data center's network

Internet access through the parent AWS Region

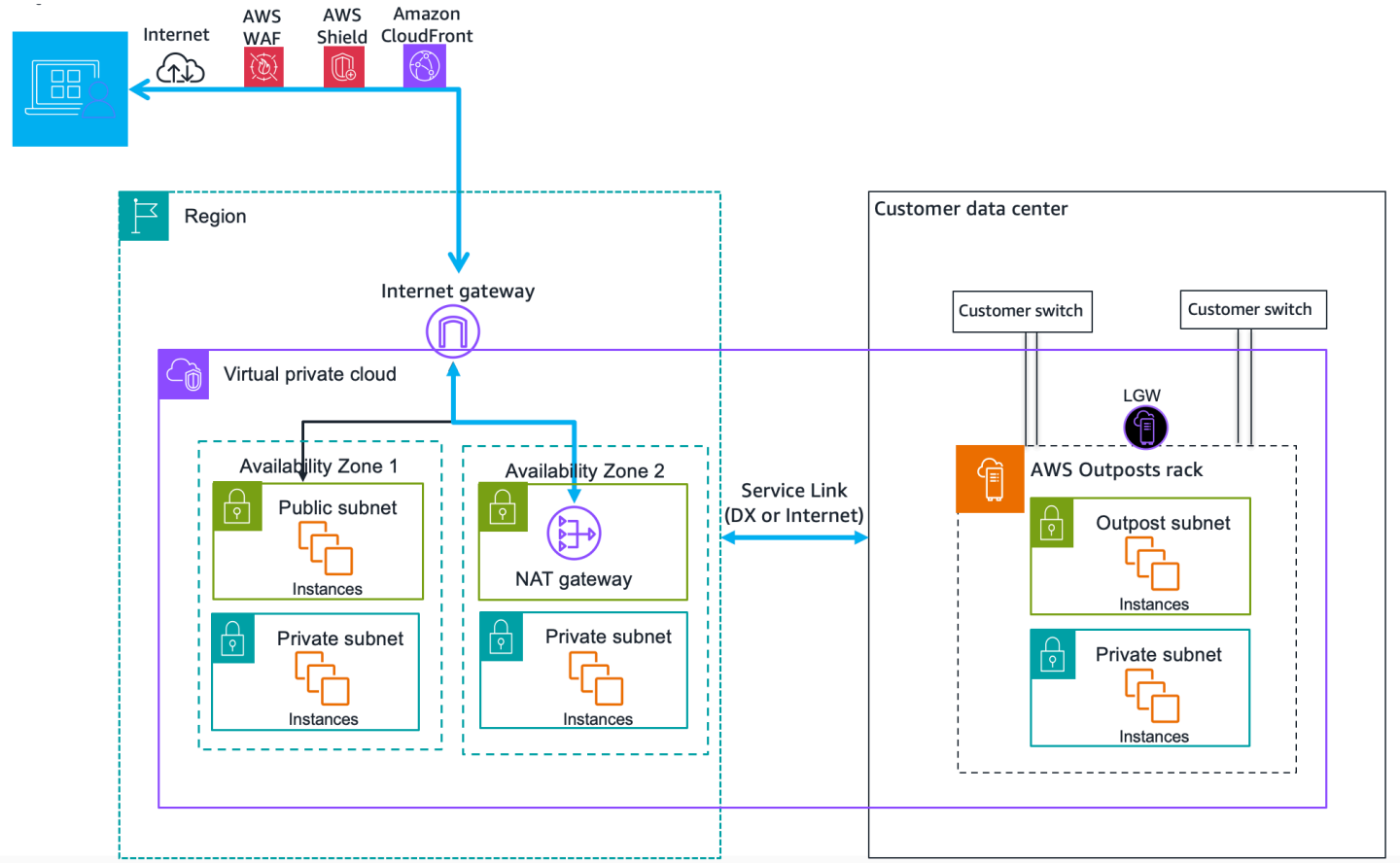
In this option, the workloads in the Outposts access the internet through the service link and then through the internet gateway (IGW) in the parent AWS Region. The outbound traffic to the internet can be through the NAT gateway instantiated in your VPC. For additional security for your ingress and egress traffic, you can use AWS security services such as AWS WAF, AWS Shield, and Amazon CloudFront in the AWS Region.

For the route table setting on the Outposts subnet, see Local gateway route tables.

Considerations

- Use this option when:
  - You need flexibility in securing the internet traffic with multiple AWS services in the AWS Region.
  - You do not have an internet point of presence in your data center or co-location facility.
- In this option, the traffic must traverse through the parent AWS Region, which introduces latency.
- Similar to data transfer charges in AWS Regions, data transfer out from the parent Availability Zone to the Outpost incurs charges. To learn more about data transfer, see Amazon EC2 On-Demand Pricing.
- The utilization of the service link bandwidth will increase.

The following image shows traffic between the workload in the Outposts instance and the internet going through the parent AWS Region.



Internet access through your local data center's network

In this option, the workloads residing in the Outposts access the internet through your local data center. The workload traffic accessing the internet traverses through your local internet point of presence and egress locally. The security layer of your local data center's network is responsible for securing the Outposts workload traffic.

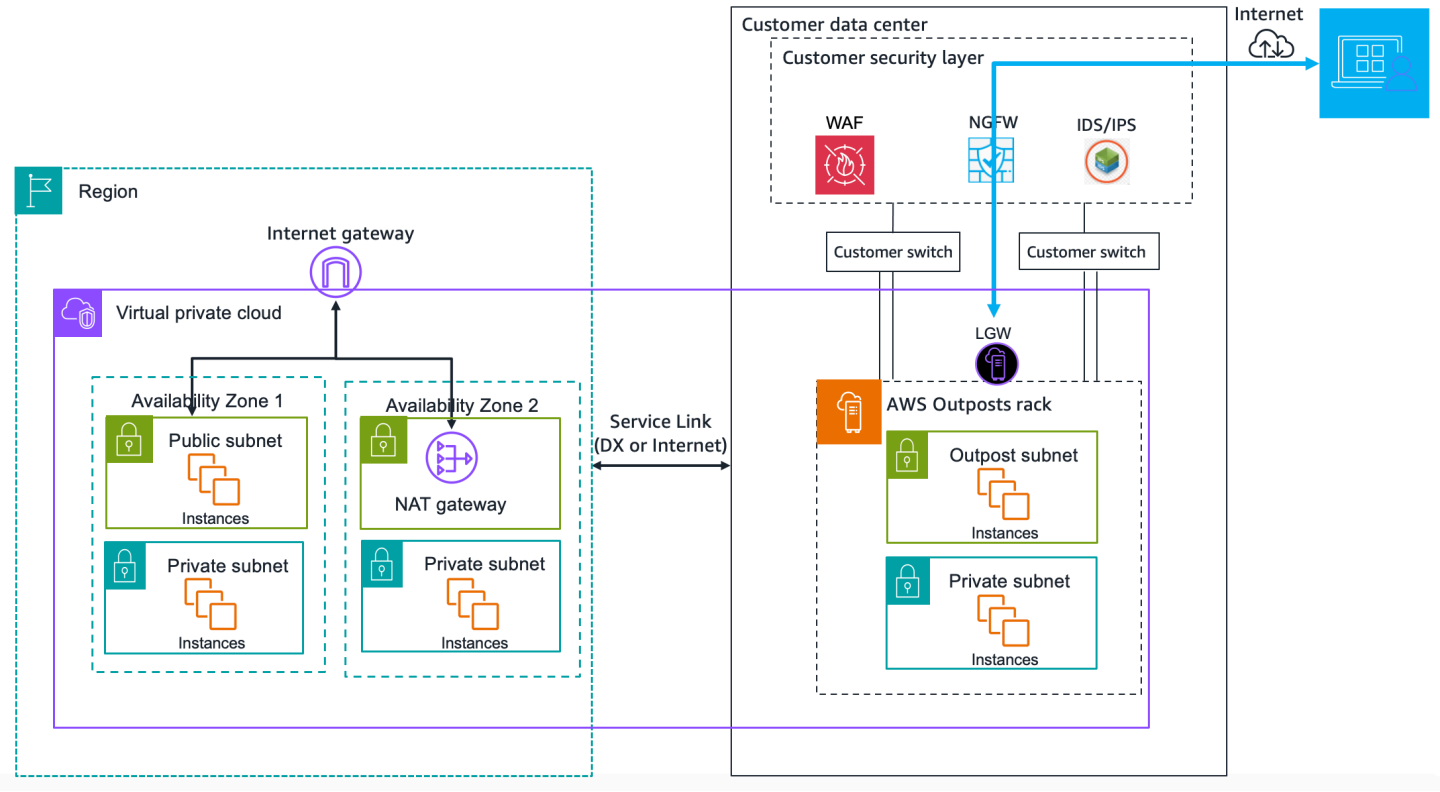
For the route table setting on the Outposts subnet, see Local gateway route tables.

Considerations

- Use this option when:
  - Your workloads require low latency access to internet services.
  - You prefer to avoid incurring Data Transfer Out (DTO) charges.
  - You want to preserve the service link bandwidth for control plane traffic.
- Your security layer is responsible for securing Outposts workload traffic.

- If you opt for Direct VPC Routing (DVR), then you must ensure that the Outposts CIDRs do not conflict with the on-premises CIDRs.
- If the default route (0/0) is propagated through the local gateway (LGW), then instances may not be able to get to the service endpoints. Alternatively, you can choose VPC endpoints to reach the desired service.

The following image shows traffic between the workload in the Outposts instance and the internet going through your local data center.



Monitor your Outpost

AWS Outposts integrates with the following services that offer monitoring and logging capabilities:

CloudWatch metrics

Use Amazon CloudWatch to retrieve statistics about data points for your Outposts as an ordered set of time series data, known as *metrics*. You can use these metrics to verify that your system is performing as expected. For more information, see CloudWatch metrics for AWS Outposts.

CloudTrail logs

Use AWS CloudTrail to capture detailed information about the calls made to AWS APIs. You can store these calls as log files in Amazon S3. You can use these CloudTrail logs to determine such information as which call was made, the source IP address where the call came from, who made the call, and when the call was made.

The CloudTrail logs contain information about the calls to API actions for AWS Outposts. They also contain information for calls to API actions from services on an Outpost, such as Amazon EC2 and Amazon EBS. For more information, see AWS Outposts information in CloudTrail.

VPC Flow Logs

Use VPC Flow Logs to capture detailed information about the traffic going to and from your Outpost and within your Outpost. For more information, see VPC Flow Logs in the *Amazon VPC User Guide*.

Traffic Mirroring

Use Traffic Mirroring to copy and forward network traffic from Outpost to out-of-band security and monitoring appliances in Outpost. You can use the mirrored traffic for content inspection, threat monitoring, or troubleshooting. For more information, see *Traffic Mirroring Guide* for Amazon Virtual Private Cloud.

AWS Health Dashboard

The AWS Health Dashboard displays information and notifications that are initiated by changes in the health of AWS resources. The information is presented in two ways: on a dashboard that shows recent and upcoming events organized by category, and in a full event log that shows all events from the past 90 days. For example, a connectivity issue on the service link would initiate

an event that would appear on the dashboard and event log, and remain in the event log for 90 days. A part of the AWS Health service, AWS Health Dashboard requires no setup and can be viewed by any user that is authenticated in your account. For more information, see Getting started with the AWS Health Dashboard.

CloudWatch metrics for AWS Outposts

AWS Outposts publishes data points to Amazon CloudWatch for your Outposts. CloudWatch enables you to retrieve statistics about those data points as an ordered set of time series data, known as *metrics*. Think of a metric as a variable to monitor, and the data points as the values of that variable over time. For example, you can monitor the instance capacity available to your Outpost over a specified time period. Each data point has an associated timestamp and an optional unit of measurement.

You can use metrics to verify that your system is performing as expected. For example, you can create a CloudWatch alarm to monitor the `ConnectedStatus` metric. If the average metric is less than 1, CloudWatch can initiate an action, such as sending a notification to an email address. You can then investigate potential on-premises or uplink networking issues that might be impacting the operations of your Outpost. Common issues include recent on-premises network configuration changes to firewall and NAT rules, or internet connection issues. For `ConnectedStatus` issues, we recommend verifying connectivity to the AWS Region from within your on-premises network, and contacting AWS Support if the problem persists.

For more information about creating a CloudWatch alarm, see Using Amazon CloudWatch Alarms in the *Amazon CloudWatch User Guide*. For more information about CloudWatch, see the Amazon CloudWatch User Guide.

Contents

Outpost metrics

The `AWS/Outposts` namespace includes the following metrics.

`ConnectedStatus`

The status of an Outpost's service link connection. If the average statistic is less than 1, the connection is impaired.

Unit: Count

Maximum resolution: 1 minute

Statistics: The most useful statistic is `Average`.

Dimensions: `OutpostId`

`CapacityExceptions`

The number of insufficient capacity errors for instance launches.

Unit: Count

Maximum resolution: 5 minutes

Statistics: The most useful statistics are `Maximum` and `Minimum`.

Dimensions: `InstanceType` and `OutpostId`

`IfTrafficIn`

The bitrate of data that the Outposts Virtual Interfaces (VIFs) receive from the connected local network devices.

Unit: Bits per second

Maximum resolution: 5 minutes

Statistics: The most useful statistics are `Max` and `Min`.

Dimensions for local gateway VIFs (lgw-vif): `OutpostsId`, `VirtualInterfaceGroupId`, and `VirtualInterfaceId`

Dimensions for service link VIFs (sl-vif): `OutpostsId` and `VirtualInterfaceId`

`IfTrafficOut`

The bitrate of data that the Outposts Virtual Interfaces (VIFs) transfer to the connected local network devices.

Unit: Bits per second

Maximum resolution: 5 minutes

Statistics: The most useful statistics are `Max` and `Min`.

Dimensions for local gateway VIFs (lgw-vif): `OutpostsId`, `VirtualInterfaceGroupId`, and `VirtualInterfaceId`

Dimensions for service link VIFs (sl-vif): `OutpostsId` and `VirtualInterfaceId`

`InstanceFamilyCapacityAvailability`

The percentage of instance capacity available. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

Unit: Percent

Maximum resolution: 5 minutes

Statistics: The most useful statistics are `Average` and `pNN.NN` (percentiles).

Dimensions: `InstanceFamily` and `OutpostId`

`InstanceFamilyCapacityUtilization`

The percentage of instance capacity in use. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

Unit: Percent

Maximum resolution: 5 minutes

Statistics: The most useful statistics are `Average` and `pNN.NN` (percentiles).

Dimensions: `Account`, `InstanceFamily`, and `OutpostId`

`InstanceTypeCapacityAvailability`

The percentage of instance capacity available. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

Unit: Percent

Maximum resolution: 5 minutes

Statistics: The most useful statistics are `Average` and `pNN.NN` (percentiles).

Dimensions: `InstanceType` and `OutpostId`

`InstanceTypeCapacityUtilization`

The percentage of instance capacity in use. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

Unit: Percent

Maximum resolution: 5 minutes

Statistics: The most useful statistics are `Average` and `pNN.NN` (percentiles).

Dimensions: `Account`, `InstanceType`, and `OutpostId`

`UsedInstanceType_Count`

The number of instance types that are currently in use, including any instance types used by managed services such as Amazon Relational Database Service (Amazon RDS) or Application Load Balancer. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

Unit: Count

Maximum resolution: 5 minutes

Dimensions: `Account`, `InstanceType`, and `OutpostId`

`AvailableInstanceType_Count`

The number of available instance types. This metric does not include capacity for any Dedicated Hosts configured on the Outpost.

Unit: Count

Maximum resolution: 5 minutes

Dimensions: `InstanceType` and `OutpostId`

`AvailableReservedInstances`

The number of instances available on the Outpost for On-Demand Capacity Reservations (ODCR). This metric does not measure Amazon EC2 Reserved Instances.

Unit: Count

Maximum resolution: 5 minutes

Dimensions: `InstanceType` and `OutpostId`

`UsedReservedInstances`

The number of instances available on the Outpost for On-Demand Capacity Reservations (ODCR). This metric does not measure Amazon EC2 Reserved Instances.

Unit: Count

Maximum resolution: 5 minutes

Dimensions: `InstanceType` and `OutpostId`

`TotalReservedInstances`

The number of instances available on the Outpost for On-Demand Capacity Reservations (ODCR). This metric does not measure Amazon EC2 Reserved Instances.

Unit: Count

Maximum resolution: 5 minutes

Dimensions: `InstanceType` and `OutpostId`

`EBSVolumeTypeCapacityUtilization`

The percentage of EBS volume type capacity in use.

Unit: Percent

Maximum resolution: 5 minutes

Statistics: The most useful statistics are `Average` and `pNN.NN` (percentiles).

Dimensions: `VolumeType` and `OutpostId`

`EBSVolumeTypeCapacityAvailability`

The percentage of EBS volume type capacity available.

Unit: Percent

Maximum resolution: 5 minutes

Statistics: The most useful statistics are `Average` and `pNN.NN` (percentiles).

Dimensions: `VolumeType` and `OutpostId`

`EBSVolumeTypeCapacityUtilizationGB`

The number of gigabytes in use for the EBS volume type.

Unit: Gigabyte

Maximum resolution: 5 minutes

Statistics: The most useful statistics are `Average` and `pNN.NN` (percentiles).

Dimensions: `VolumeType` and `OutpostId`

`EBSVolumeTypeCapacityAvailabilityGB`

The number of gigabytes of available capacity for the EBS volume type.

Unit: Gigabyte

Maximum resolution: 5 minutes

Statistics: The most useful statistics are `Average` and `pNN.NN` (percentiles).

Dimensions: `VolumeType` and `OutpostId`

Outpost metric dimensions

To filter the metrics for your Outpost, use the following dimensions.

Dimension	Description
`Account`	The account or service using the capacity.
`InstanceFamily`	The instance family.
`InstanceType`	The instance type.
`OutpostId`	The ID of the Outpost.
`VolumeType`	The EBS volume type.

Dimension	Description
VirtualInterfaceId	The ID of the local gateway or service link Virtual Interface (VIF).
VirtualInterfaceGroupId	The ID of the virtual interface group for the local gateway Virtual Interface (VIF).

View CloudWatch metrics for your outpost

You can view the CloudWatch metrics for your load balancers using the CloudWatch console.

To view metrics using the CloudWatch console

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose **Metrics**.
3. Select the **Outposts** namespace.
4. (Optional) To view a metric across all dimensions, enter its name in the search box.

To view metrics using the AWS CLI

Use the following list-metrics command to list the available metrics.

```
aws cloudwatch list-metrics --namespace AWS/Outposts
```

To get the statistics for a metric using the AWS CLI

Use the following get-metric-statistics command to get statistics for the specified metric and dimension. CloudWatch treats each unique combination of dimensions as a separate metric. You can't retrieve statistics using combinations of dimensions that were not specially published. You must specify the same dimensions that were used when the metrics were created.

```
aws cloudwatch get-metric-statistics --namespace AWS/Outposts \
--metric-name InstanceTypeCapacityUtilization --statistics Average  --period 3600 \
--dimensions Name=OutpostId,Value=op-01234567890abcdef \
Name=InstanceType,Value=c5.xlarge \
--start-time 2019-12-01T00:00:00Z --end-time 2019-12-08T00:00:00Z
```

Log AWS Outposts API calls using AWS CloudTrail

AWS Outposts is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Outposts. CloudTrail captures all API calls for AWS Outposts as events. The calls captured include calls from the AWS Outposts console and code calls to the AWS Outposts API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an S3 bucket, including events for AWS Outposts. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS Outposts, the IP address from which the request was made, who made the request, when it was made, and additional details.

For more information about CloudTrail, see the AWS CloudTrail User Guide.

AWS Outposts information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS Outposts, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail event history.

For an ongoing record of events in your AWS account, including events for AWS Outposts, create a trail. A *trail* enables CloudTrail to deliver log files to an S3 bucket in the parent AWS Region. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail Supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple Regions and Receiving CloudTrail log files from multiple accounts

All AWS Outposts actions are logged by CloudTrail. They are documented in the *AWS Outposts API Reference*. For example, calls to the `CreateOutpost`, `GetOutpostInstanceTypes`, and `ListSites` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine whether the request was made:

- With root or user credentials.
- With temporary security credentials for a role or federated user.
- By another AWS service.

For more information, see the CloudTrail userIdentity element.

Understanding AWS Outposts log file entries

A trail is a configuration that enables delivery of events as log files to an S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source. It includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the `CreateOutpost` action.


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE:jdoe",
        "arn": "arn:aws:sts::111122223333:assumed-role/example/jdoe",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/example",
                "accountId": "111122223333",
                "userName": "example"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-08-14T16:28:16Z"
            }
```

```
        }
    },
    "eventTime": "2020-08-14T16:32:23Z",
    "eventSource": "outposts.amazonaws.com",
    "eventName": "SetSiteAddress",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "XXX.XXX.XXX.XXX",
    "userAgent": "userAgent",
    "requestParameters": {
        "SiteId": "os-123ab4c56789de01f",
        "Address": "***"
    },
    "responseElements": {
        "Address": "***",
        "SiteId": "os-123ab4c56789de01f"
    },
    "requestID": "1abcd23e-f4gh-567j-klm8-9np01q234r56",
    "eventID": "1234a56b-c78d-9e0f-g1h2-34jk56m7n890",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Outpost maintenance

Under the shared responsibility model, AWS is responsible for the hardware and software that run AWS services. This applies to AWS Outposts, just as it does to an AWS Region. For example, AWS manages security patches, updates firmware, and maintains the Outpost equipment. AWS also monitors the performance, health, and metrics for your Outpost and determines whether any maintenance is required.

> **Warning**
>
> Data on instance store volumes is lost if the underlying disk drive fails, or if the instance stops, hibernates, or terminates. To prevent data loss, we recommend that you back up your long-term data on instance store volumes to persistent storage, such as an Amazon S3 bucket, an Amazon EBS volume, or a network storage device in your on-premises network.

Contents

Hardware maintenance

If AWS detects an irreparable issue with hardware hosting Amazon EC2 instances running on your Outpost, we will notify the owner of the Outpost and the owner of the instances that the affected instances are scheduled for retirement. For more information, see Instance retirement in the *Amazon EC2 User Guide*.

The Outpost owner and instance owner can work together to resolve the issue. The instance owner can stop and start an affected instance to migrate it to available capacity. Instance owners can stop and start the affected instances at a time that is convenient for them. Otherwise, AWS stops and starts the affected instances on the instance retirement date. If there is no additional capacity on

the Outpost, the instance remains in the stopped state. The Outpost owner can try to free up used capacity or request additional capacity for the Outpost so that the migration can complete.

If hardware maintenance is required, AWS will contact the manager of the Outpost site to confirm a date and time for the AWS installation team to visit. Visits can be scheduled as soon as two business days from the time that the site manager speaks with the AWS team.

When the AWS installation team arrives on site, they will replace the unhealthy hosts, switches, or rack elements and bring the new capacity online. They will not perform any hardware diagnostics or repairs on site. If they replace a host, they will remove and destroy the NIST-compliant physical security key, effectively shredding any data that might remain on the hardware. This ensures that no data leaves your site. If they replace an Outpost networking device, network configuration information might be present on the device when it is removed from the site. This information might include IP addresses and ASNs used to establish virtual interfaces for configuring the path to your local network or back to the Region.

Firmware updates

Updating the Outpost firmware does not typically affect the instances on your Outpost. In the rare case that we need to reboot the Outpost equipment to install an update, you will receive an instance retirement notice for any instances running on that capacity.

Network equipment maintenance

Maintenance of Outpost Networking Devices (OND) is performed without affecting regular Outpost operations and traffic. If maintenance is required traffic is shifted away from the OND. You might notice temporary changes in BGP advertisements, such as AS-Path prepending, and corresponding changes in traffic patterns on Outpost uplinks. With OND firmware updates, you might notice BGP flapping.

We recommend that you configure customer network equipment to receive BGP advertisements from Outposts without changing the BGP attributes, and enable BGP multipath/load balancing to achieve optimal inbound traffic flows. AS-Path prepending is used for local gateway prefixes to shift traffic away from ONDs if maintenance is required. The customer network should prefer routes from Outposts with an AS-Path length of 1 over routes with an AS-Path length of 4.

The customer network should advertise equal BGP prefixes with the same attributes to all ONDs. The Outpost network load balances outbound traffic between all uplinks by default. Routing

policies are used on the Outpost side to shift traffic away from an OND if maintenance is required. This traffic shift requires equal BGP prefixes from the customer side on all ONDs. If maintenance is required on the customer network, we recommend that you use AS-Path prepending to temporarily shift traffic array from specific uplinks.

Best practices for AWS Outposts power and network events

As stated in the AWS Service Terms for AWS Outposts customers, the facility where the Outposts equipment is located must meet the minimum power and network requirements to support the installation, maintenance, and use of the Outposts equipment. An Outposts rack can operate correctly only when power and network connectivity is uninterrupted.

Power events

With complete power outages, there is an inherent risk that an AWS Outposts resource may not return to service automatically. In addition to deploying redundant power and backup power solutions, we recommend that you do the following in advance to mitigate the impact of some of the worst-case scenarios:

- Move your services and applications off the Outposts equipment in a controlled fashion, using DNS-based or off-rack load-balancing changes.
- Stop containers, instances, databases in an ordered incremental fashion and use the reverse order when restoring them.
- Test plans for the controlled moving or stopping of services.
- Back-up critical data and configurations and store it outside the Outposts.
- Keep power downtimes to a minimum.
- Avoid repeated switching of the power feeds (off-on-off-on) during the maintenance.
- Allow for extra time within the maintenance window to deal with the unexpected.
- Manage the expectations of your users and customers by communicating a wider maintenance window time-frame than you would normally need.

Network connectivity events

The service link connection between your Outpost and the AWS Region or Outposts home Region will typically automatically recover from network interruptions or issues that may occur in your upstream corporate network devices or in the network of any third party connectivity provider

once the network maintenance is completed. During the time the service link connection is down, your Outposts operations are limited to local network activities.

For more information, see the question *What happens when my facility's network connection goes down?* on the AWS Outposts rack FAQs page.

If the service link is down because of an on-site power issue or the loss of network connectivity, the AWS Health Dashboard sends a notification to the account that owns the Outposts. Neither you nor AWS can suppress the notification of a service link interruption, even if the interruption is expected. For more information, see Getting started with your AWS Health Dashboard in the *AWS Health User Guide*.

In the case of a planned service maintenance that will affect network connectivity, take the following proactive steps to limit the impact of potential problematic scenarios:

- If your Outposts rack connects to the parent AWS Region through Internet or public Direct Connect, then in advance of a planned maintenance, capture a trace-route. Having a working (pre-network-maintenance) network path and a problematic (post-network-maintenance) network path to identify the differences would help in troubleshooting. If you escalate a post-maintenance issue to AWS or your ISP, you can include this information.

  Capture a trace-route between:

  - The public IP addresses at the Outposts location and the IP address returned by the `outposts.region.amazonaws.com`. Replace *`region`* with the name of the parent AWS Region.
  - Any instance in the parent Region with public Internet connectivity and the public IP addresses at the Outposts location.
- If you are in control of the network maintenance, limit the duration of downtime for the service link. Include a step in your maintenance process that verifies that the network has recovered.
- If you are not in control of the network maintenance, monitor the service link downtime with respect to the announced maintenance window and escalate early to the party in charge of the planned network maintenance if the service link is not back up at the end of the announced maintenance window.

Resources

Here are some monitoring related resources that can provide reassurance that the Outposts is operating normally after a planned or unplanned power or network event:

- The AWS blog Monitoring best practices for AWS Outposts covers observability and event management best practices specific to Outposts.
- The AWS blog Debugging tool for network connectivity from Amazon VPC explains the *AWSSupport-SetupIPMonitoringFromVPC* tool. This tool is an AWS Systems Manager document (SSM document) that creates an Amazon EC2 Monitor Instance in a subnet specified by you and monitors target IP addresses. The document runs ping, MTR, TCP trace-route and trace-path diagnostic tests and stores the results in Amazon CloudWatch Logs which can be visualized in a CloudWatch dashboard (e.g. latency, packet loss). For Outposts monitoring, the Monitor Instance should be in one subnet of the parent AWS Region and configured to monitor one or more of your Outpost instances using its private IP(s) - this will provide packet loss graphs and latency between AWS Outposts and the parent AWS Region.
- The AWS blog Deploying an automated Amazon CloudWatch dashboard for AWS Outposts using AWS CDK describes the steps involved in deploying an automated dashboard.
- If you have questions or need more information, see Creating a support case in the *AWS Support User Guide*.

Optimize Amazon EC2 for AWS Outposts

In contrast to the AWS Region, Amazon Elastic Compute Cloud (Amazon EC2) capacity on an Outpost is finite. You are constrained by the total volume of compute capacity that you ordered. This topic offers best practices and optimization strategies to help you get the most out of your Amazon EC2 capacity in AWS Outposts.

Contents

Dedicated Hosts on Outposts

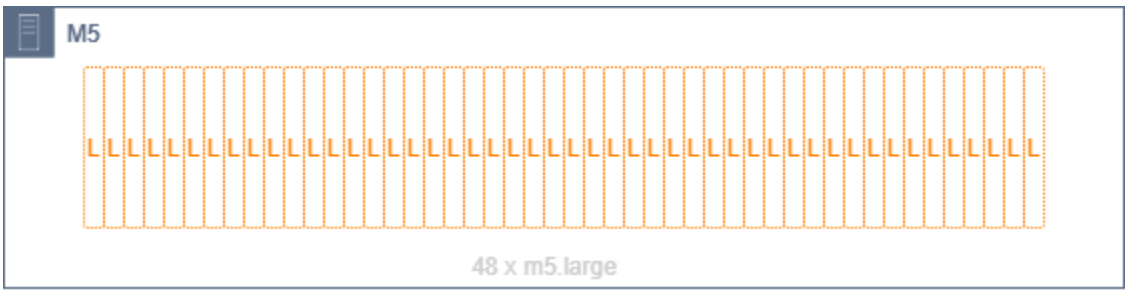
An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Your Outpost already provides you with dedicated hardware, but Dedicated Hosts allows you to use existing software licenses with per-socket, per-core, or per-VM license restrictions against a single host. For more information, see Dedicated Host on AWS Outposts in the *Amazon EC2 User Guide*. For Windows, see Dedicated Host on AWS Outposts in the *Amazon EC2 User Guide*.

Beyond licensing, Outpost owners can use Dedicated Hosts to optimize the servers in their Outpost deployments in two ways:

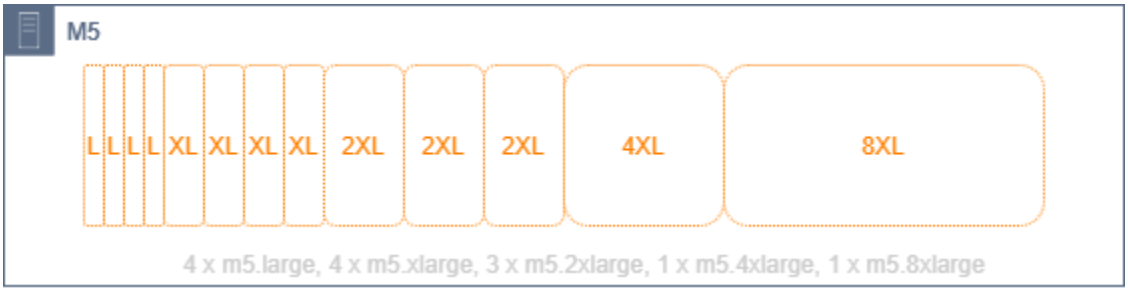
- Alter the capacity layout of a server
- Control instance placement at the hardware level

Alter the capacity layout of a server

Dedicated Hosts offers you the capability to alter the layout of servers in your Outpost deployment without contacting AWS Support. When you purchase capacity for your Outpost, you specify an EC2 capacity layout that each server provides. Each server supports a single family of instance types. A layout can offer a single instance type or multiple instance types. Dedicated Hosts allows you to alter whatever you chose for that initial layout. If you allocate a host to support a single instance type for the entire capacity, you can only launch a single instance type from that host. The following illustration presents an m5.24xlarge server with a homogeneous layout:



You can allocate the same capacity for multiple instance types. When you allocate a host to support multiple instance types, you get a heterogeneous layout that doesn't require an explicit capacity layout. The following illustration presents an m5.24xlarge server with a heterogeneous layout at full capacity:



For more information, see Allocate Dedicated Hosts in the *Amazon EC2 User Guide* or Allocate Dedicated Hosts *Amazon EC2 User Guide*.

Control instance placement at the hardware level

You can use Dedicated Hosts to control instance placement at the hardware level. Use auto-placement for Dedicated Hosts to manage whether instances you launch are launched onto a specific host, or onto any available host that has matching configurations. Use host affinity to establish a relationship between an instance and a Dedicated Host. If you have an Outpost rack, you can use these Dedicated Hosts features to minimize the impact of correlated hardware failures. For more information about instance recovery, see Understand auto-placement and affinity in the *Amazon EC2 User Guide* or Understand auto-placement and affinity *Amazon EC2 User Guide*.

You can share Dedicated Hosts using AWS Resource Access Manager. Sharing Dedicated Hosts allows you to distribute hosts in an Outpost deployment across AWS accounts. For more information, see *Working with shared resources*.

Set up instance recovery

Instances on your Outpost that go into an unhealthy state because of hardware failure must be migrated to a healthy host. You can set up auto-recovery to have this migration done automatically based on instance status checks. For more information, see Recover your Linux instance or Recover your Windows instance.

Placement groups on Outposts

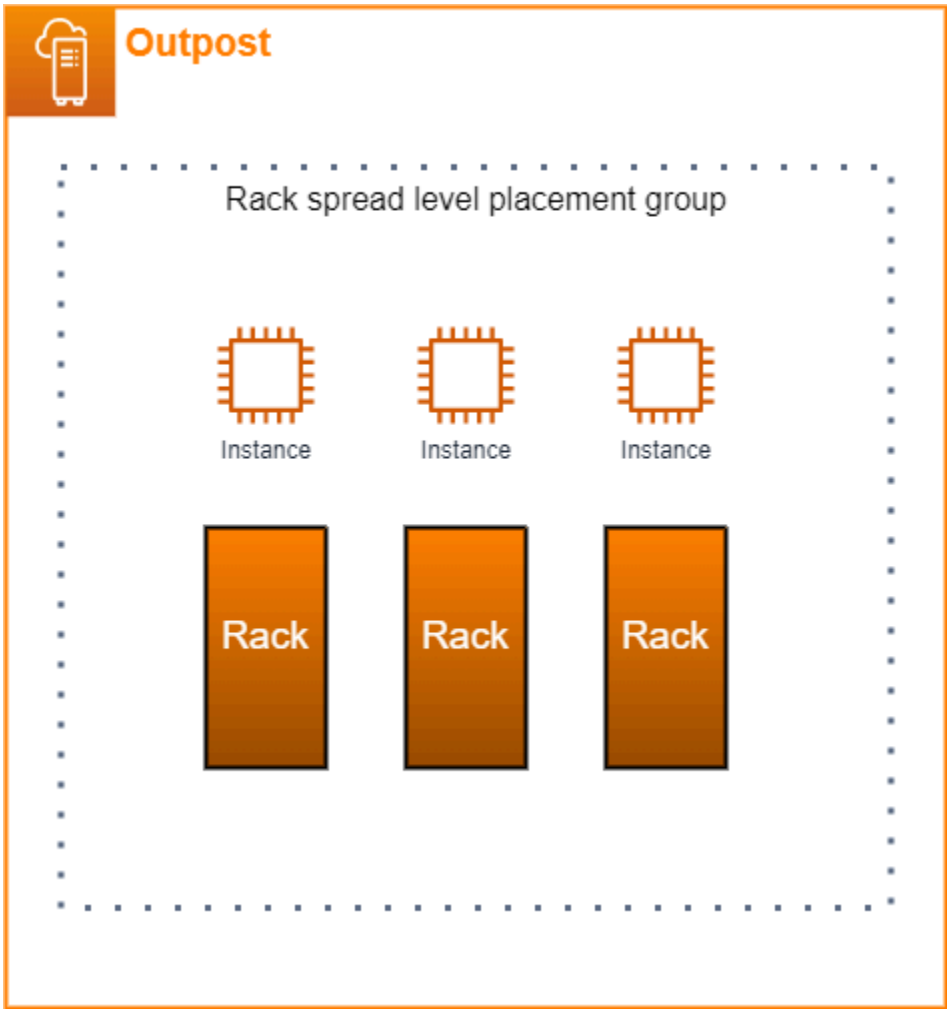
AWS Outposts supports placement groups. Use placement groups to influence how Amazon EC2 should attempt to place groups of interdependent instances that you launch on the underlying hardware. You can use different strategies (cluster, partition, or spread) to meet the needs of different workloads. If you have a single-rack Outpost, you can use the spread strategy to place instances across hosts instead of racks.

Spread placement groups

Use a spread placement group to distribute a single instance across distinct hardware. Launching instances in a spread placement group reduces the risk of simultaneous failures that might occur when instances share the same equipment. Placement groups can spread instances across racks or hosts. You can use host level spread placement groups only with AWS Outposts.

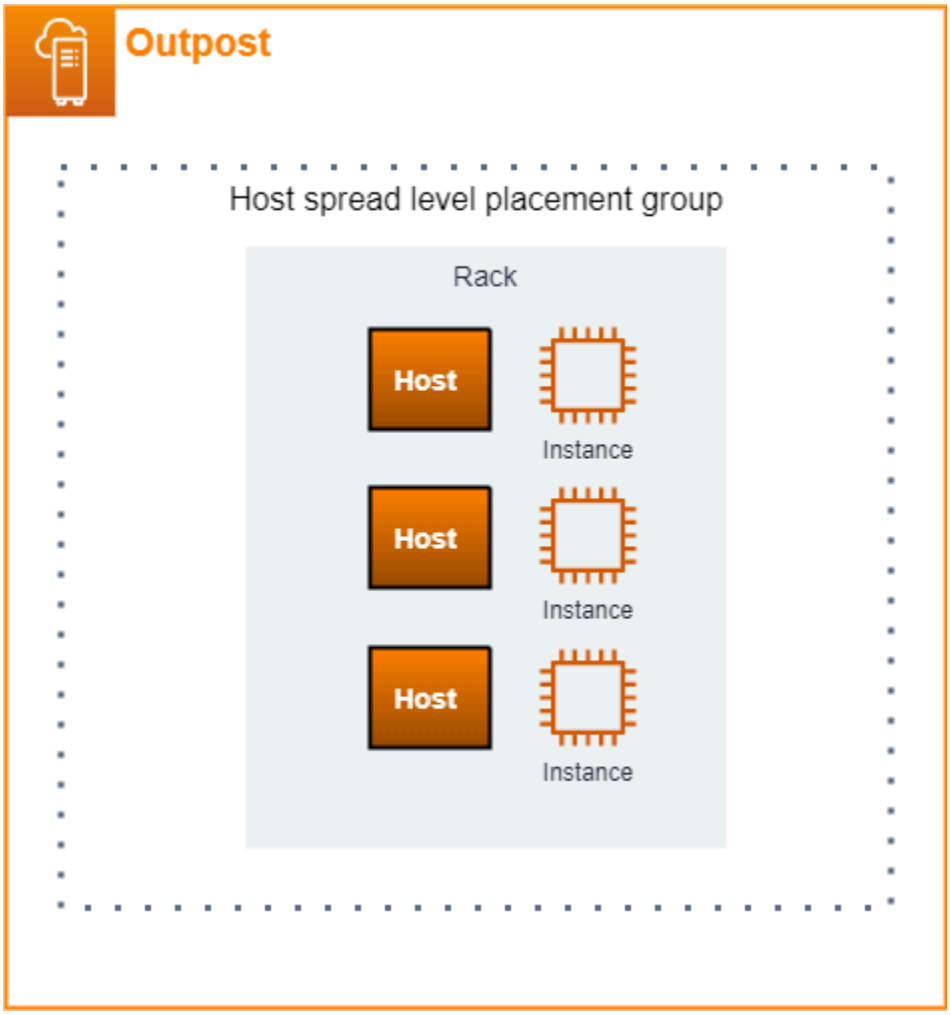
Rack spread level placement groups

Your rack spread level placement group can hold as many instances as you have racks in your Outpost deployment. The following illustration shows a three-rack Outpost deployment running three instances in a rack spread level placement group.



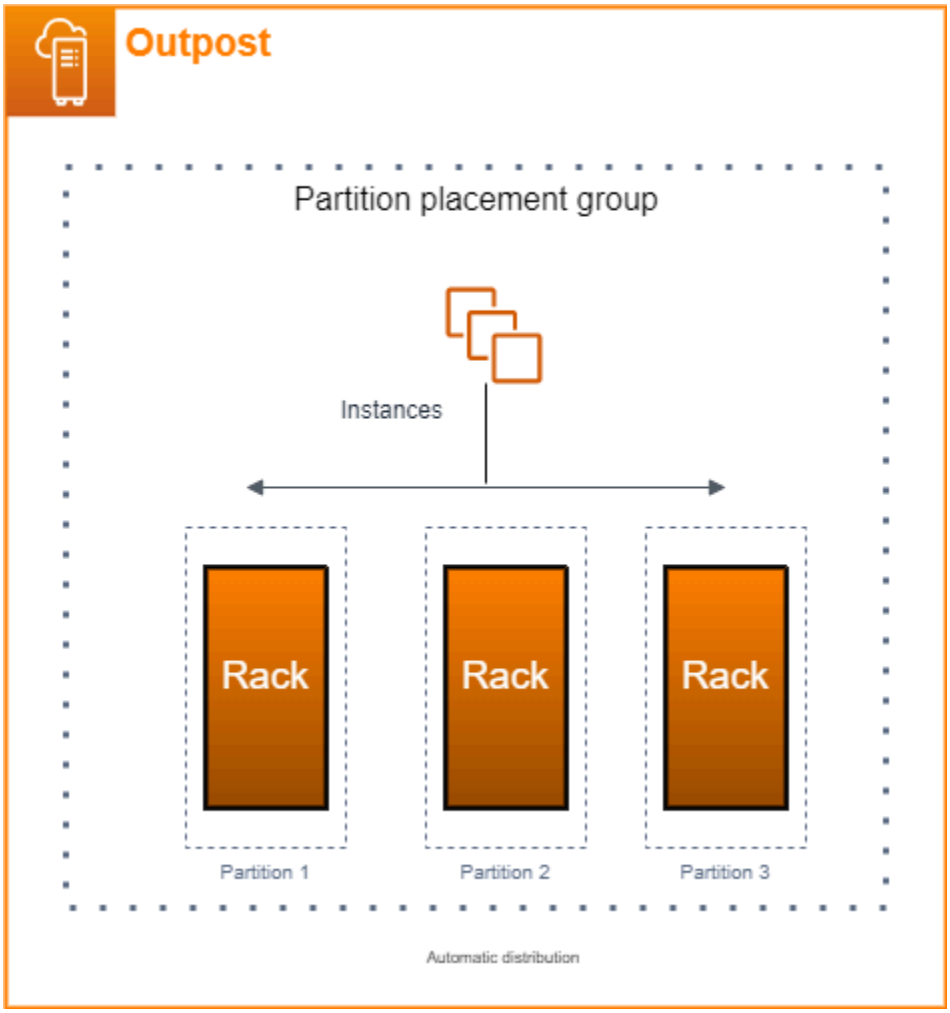
Host spread level placement groups

Your host spread level placement group can hold as many instances as you have hosts in your Outpost deployment. The following illustration shows a single-rack Outpost deployment running three instances in a host spread level placement group.

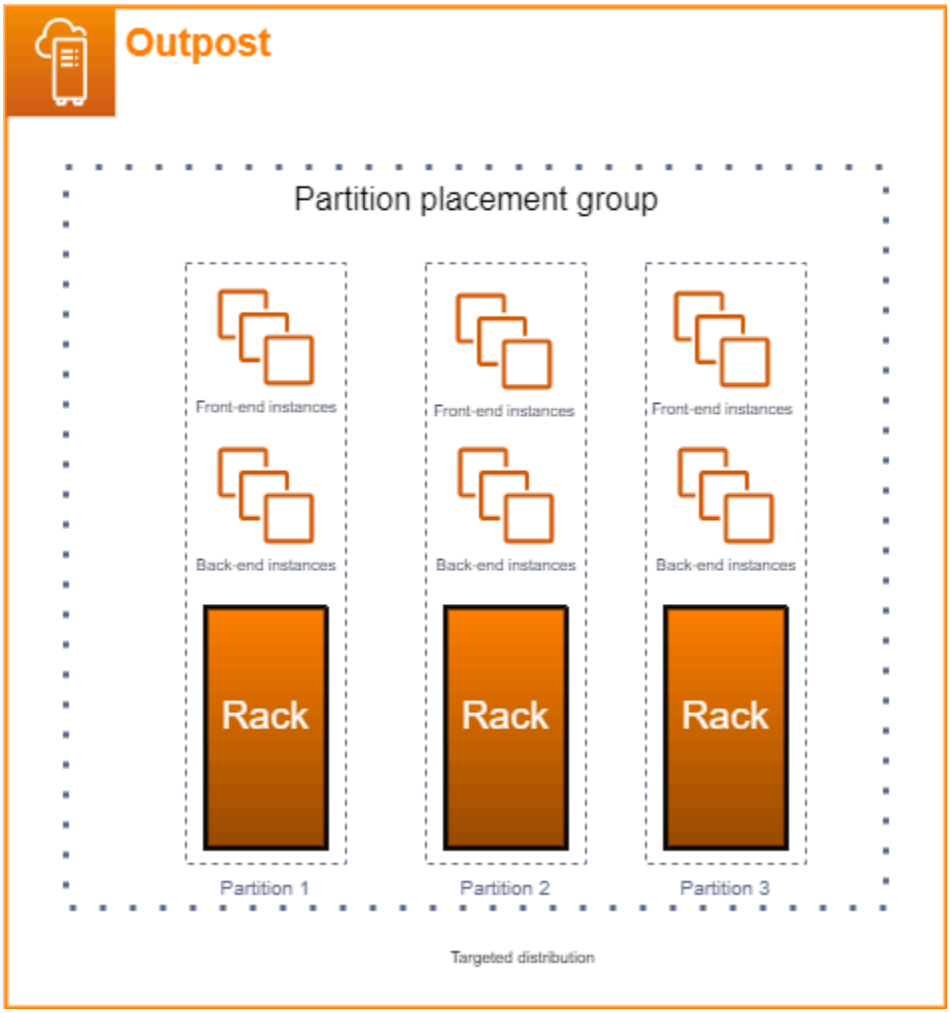


Partition placement groups

Use a partition placement group to distribute multiple instances across racks with partitions. Each partition can hold multiple instances. You can use automatic distribution to spread instances across partitions or deploy instances to target partitions. The following illustration shows a partition placement group with automatic distribution.



You can also deploy instances to target partitions. The following illustration shows a partition placement group with targeted distribution.

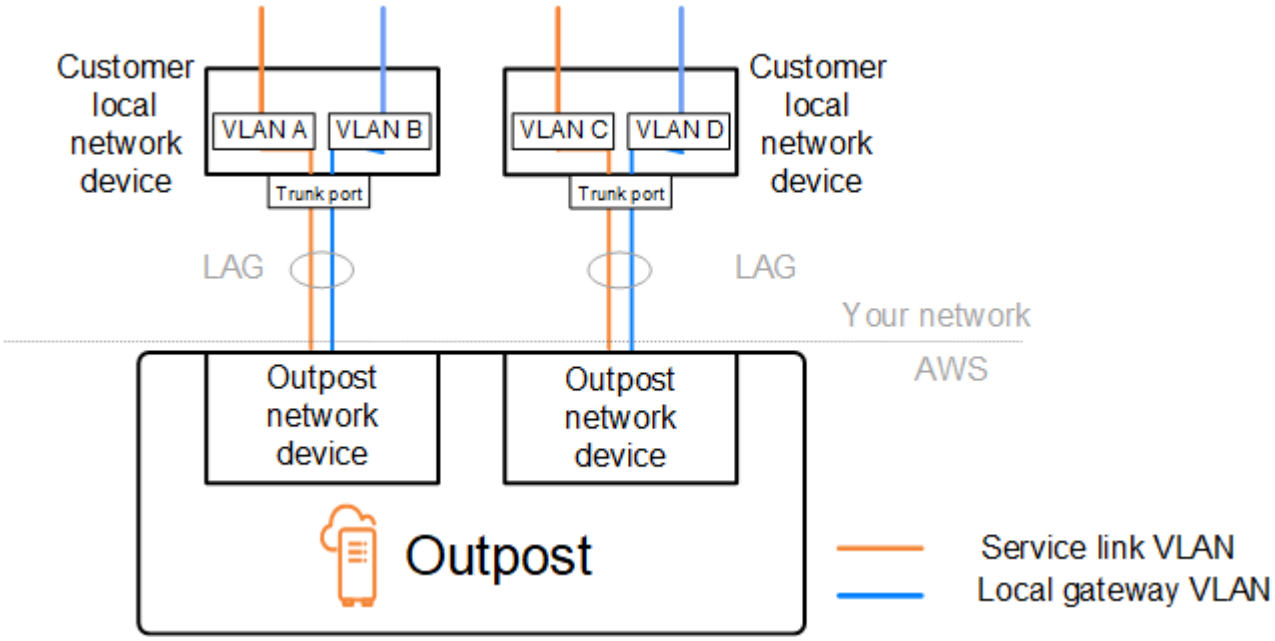


For more information about working with placement groups, see Placement groups and Placement groups on AWS Outposts in the *Amazon EC2 User Guide*. For Windows, see Placement groups and Placement groups on AWS Outposts in the *Amazon EC2 User Guide*.

For more information about AWS Outposts high availability, see AWS Outposts High Availability Design and Architecture Considerations.

AWS Outposts rack network troubleshooting checklist

Use this checklist to help troubleshoot a service link that has a status of `DOWN`.



Connectivity with Outpost network devices

Check the BGP peering status on the customer local network devices that are connected to the Outpost network devices. If the BGP peering status is `DOWN`, follow these steps:

1. Ping the remote peer IP address on the Outpost network devices from the customer devices. You can find the peer IP address in the BGP configuration of your device. You can also refer to the Network readiness checklist provided to you at the time of installation.
2. If pinging is unsuccessful, check the physical connection and ensure that connectivity status is UP.
   a. Confirm the LACP status of the customer local network devices.
   b. Check the interface status on the device. If the status is UP, skip to step 3.
   c. Check the customer local network devices and confirm that the optical module is working.
   d. Replace faulty fibers and ensure the lights (Tx/Rx) are within acceptable range.
3. If pinging is successful, check the customer local network devices and ensure that the following BGP configurations are correct.
   a. Confirm that the local Autonomous System Number (Customer ASN) is correctly configured.
   b. Confirm that the remote Autonomous System Number (Outpost ASN) is correctly configured.
   c. Confirm that the interface IP and remote peer IP addresses are correctly configured.
   d. Confirm that the advertised and received routes are correct.

4. If your BGP session is flapping between active and connect states, verify that TCP port 179 and other relevant ephemeral ports are not blocked on the customer local network devices.
5. If you need to troubleshoot further, check the following on the customer local network devices:
   a. BGP and TCP debug logs
   b. BGP logs
   c. Packet capture
6. If the issue persists, perform MTR / traceroute / packet captures from your Outpost connected router to the Outpost network device peer IP addresses. Share the test results with AWS Support, using your Enterprise support plan.

If BGP peering status is UP between the customer local network devices and the Outpost network devices, but the service link is still DOWN, you can troubleshoot further by checking the following devices on your customer local network devices. Use one of the following checklists, depending on how your service link connectivity is provisioned.

- Edge routers connected with AWS Direct Connect – Public virtual interface in use for service link connectivity. For more information, see AWS Direct Connect public virtual interface connectivity to AWS Region.
- Edge routers connected with AWS Direct Connect – Private virtual interface in use for service link connectivity. For more information, see AWS Direct Connect private virtual interface connectivity to AWS Region.
- Edge routers connected with Internet Service Providers (ISPs) – Public internet in use for service link connectivity. For more information, see ISP public internet connectivity to AWS Region.

AWS Direct Connect public virtual interface connectivity to AWS Region

Use the following checklist to troubleshoot edge routers connected with AWS Direct Connect when a public virtual interface is in use for service link connectivity.

1. Confirm that the devices connecting directly with the Outpost network devices are receiving the service link IP address ranges through BGP.
   a. Confirm the routes that are being received through BGP from your device.
   b. Check the route table of the service link Virtual Routing and Forwarding instance (VRF). It should show that it is using the IP address range.

2. To ensure Region connectivity, check the route table for the service link VRF. It should include the AWS Public IP address ranges or the default route.
3. If you are not receiving the AWS public IP address ranges in the service link VRF, check the following items.
   a. Check the AWS Direct Connect link status from the edge router or the AWS Management Console.
   b. If the physical link is `UP`, check the BGP peering status from the edge router.
   c. If the BGP peering status is `DOWN`, ping the peer AWS IP address and check the BGP configuration in the edge router. For more information, see Troubleshooting AWS Direct Connect in the *AWS Direct Connect User Guide* and My virtual interface BGP status is down in the AWS console. What should I do?.
   d. If BGP is established and you are not seeing the default route or AWS public IP address ranges in the VRF, contact AWS Support, using your Enterprise support plan.
4. If you have an on-premises firewall, check the following items.
   a. Confirm that the required ports for service link connectivity are allowed in the network firewalls. Use traceroute on port 443 or any other network troubleshooting tool to confirm the connectivity through the firewalls and your network devices. The following ports are required to be configured in the firewall policies for the service link connectivity.
      - **TCP protocol** – Source port: TCP 1025-65535, Destination port: 443.
      - **UDP protocol** – Source port: TCP 1025-65535, Destination port: 443.
   b. If the firewall is stateful, ensure that the outbound rules allow the Outpost's service link IP address range to the AWS public IP address ranges. For more information, see AWS Outposts connectivity to AWS Regions.
   c. If the firewall is not stateful, make sure to allow the inbound flow also (from the AWS public IP address ranges to the service link IP address range).
   d. If you have configured a virtual router in the firewalls, ensure that the appropriate routing is configured for traffic between the Outpost and the AWS Region.
5. If you have configured NAT in the on-premises network to translate the Outpost's service link IP address ranges to your own public IP addresses, check the following items.
   a. Confirm that the NAT device is not overloaded and has free ports to allocate for new sessions.
   b. Confirm that the NAT device is correctly configured to perform the address translation.

6. If the issue persists, perform MTR / traceroute / packet captures from your edge router to the AWS Direct Connect peer IP addresses. Share the test results with AWS Support, using your Enterprise support plan.

AWS Direct Connect private virtual interface connectivity to AWS Region

Use the following checklist to troubleshoot edge routers connected with AWS Direct Connect when a private virtual interface is in use for service link connectivity.

1. If connectivity between the Outpost rack and the AWS Region is using the AWS Outposts private connectivity feature, check the following items.
   a. Ping the remote peering AWS IP address from the edge router and confirm the BGP peering status.
   b. Ensure that BGP peering over the AWS Direct Connect private virtual interface between your service link endpoint VPC and the Outpost installed on your premises is UP. For more information, see Troubleshooting AWS Direct Connect in the *AWS Direct Connect User Guide*, My virtual interface BGP status is down in the AWS console. What should I do?, and How can I troubleshoot BGP connection issues over Direct Connect?.
   c. The AWS Direct Connect private virtual interface is a private connection to your edge router in your chosen AWS Direct Connect location, and it uses BGP to exchange routes. Your private virtual private cloud (VPC) CIDR range is advertised through this BGP session to your edge router. Similarly, the IP address range for the Outpost service link is advertised to the region through BGP from your edge router.
   d. Confirm that the network ACLs associated with the service link private endpoint in your VPC allow the relevant traffic. For more information, see Network readiness checklist.
   e. If you have an on-premises firewall, ensure that the firewall has outbound rules that allow the service link IP address ranges and the Outpost service endpoints (the network interface IP addresses) located in the VPC or the VPC CIDR. Ensure that the TCP 1025-65535 and UDP 443 ports are not blocked. For more information, see Introducing AWS Outposts private connectivity.
   f. If the firewall is not stateful, ensure that the firewall has rules and policies to allow inbound traffic to the Outpost from the Outpost service endpoints in the VPC.
2. If you have more than 100 networks in your on-premises network, you can advertise a default route over the BGP session to AWS on your private virtual interface. If you don't want to

advertise a default route, summarize the routes so that the number of advertised routes is less than 100.

3. If the issue persists, perform MTR / traceroute / packet captures from your edge router to the AWS Direct Connect peer IP addresses. Share the test results with AWS Support, using your Enterprise support plan.

ISP public internet connectivity to AWS Region

Use the following checklist to troubleshoot edge routers connected through an ISP when using the public internet for service link connectivity.

- Confirm that the internet link is up.
- Confirm that the public servers are accessible from your edge devices connected through an ISP.

If the internet or public servers are not accessible through the ISP links, complete the following steps.

1. Check whether BGP peering status with the ISP routers is established.
   a. Confirm that the BGP is not flapping.
   b. Confirm that the BGP is receiving and advertising the required routes from the ISP.
2. In case of static route configuration, check that the default route is properly configured on the edge device.
3. Confirm whether you can reach the internet using another ISP connection.
4. If the issue persists, perform MTR / traceroute / packet captures on your edge router. Share the results with your ISP's technical support team for further troubleshooting.

If the internet and public servers are accessible through the ISP links, complete the following steps.

1. Confirm whether any of your publicly accessible EC2 instances or load balancers in the Outpost home Region are accessible from your edge device. You can use ping or telnet to confirm the connectivity, and then use traceroute to confirm the network path.
2. If you use VRFs to separate traffic in your network, confirm that the service link VRF has routes or policies that direct traffic to and from the ISP (internet) and VRF. See the following checkpoints.

   a. Edge routers connecting with the ISP. Check the edge router's ISP VRF route table to confirm that the service link IP address range is present.

   b. Customer local network devices connecting with the Outpost. Check the configurations of the VRFs and ensure that the routing and policies required for connectivity between the service link VRF and the ISP VRF are configured properly. Usually, a default route is sent from the ISP VRF into the service link VRF for traffic to the internet.

   c. If you configured source-based routing in the routers connected to your Outpost, confirm that the configuration is correct.

3. Ensure that the on-premises firewalls are configured to allow outbound connectivity (TCP 1025-65535 and UDP 443 ports) from the Outpost service link IP address ranges to the public AWS IP address ranges. If the firewalls are not stateful, ensure that inbound connectivity to the Outpost is also configured.
4. Ensure that NAT is configured in the on-premises network to translate the Outpost's service link IP address ranges to public IP addresses. In addition, confirm the following items.

   a. The NAT device is not overloaded and has free ports to allocate for new sessions.

   b. The NAT device is correctly configured to perform the address translation.

If the issue persists, perform MTR / traceroute / packet captures.

- If the results show that packets are dropping or blocked at the on-premises network, check with your network or technical team for additional guidance.
- If the results show that the packets are dropping or blocked at the ISP's network, contact the ISP's technical support team.
- If the results do not show any issues, collect the results from all tests (such as MTR, telnet, traceroute, packet captures, and BGP logs) and contact AWS Support using your Enterprise support plan.

Outposts is behind two firewall devices

If you have placed your Outpost behind a high-availability pair of synced firewalls or two stand-alone firewalls, asymmetric routing of the service link might occur. This means that inbound traffic could pass through firewall-1, while outbound traffic goes through firewall-2. Use the following checklist to identify potential asymmetric routing of the service link especially if it was functioning correctly before.

- Verify if there were any recent changes or ongoing maintenance in your corporate network's routing setup that might have led to asymmetric routing of the service link through the firewalls.
  - Use firewall traffic graphs to check for changes to traffic patterns that line up with the start of the service link issue.
  - Check for a partial firewall failure or a split-brained firewall-pair scenario that might have caused your firewalls to no longer sync their connection tables between each other.
  - Check for links down or recent changes to routing (OSPF/ISIS/EIGRP metric changes, BGP route-map changes) in your corporate network that line up with the start of the service link issue.
- If you are using public Internet connectivity for the service link to the home region, a service provider maintenance could have given rise to asymmetric routing of the service link through the firewalls.
  - Check traffic graphs for links to your ISP(s) for changes to traffic patterns that line up with the start of the service link issue.
- If you are using AWS Direct Connect connectivity for the service link, it is possible that an AWS planned maintenance triggered asymmetric routing of the service link.
  - Check for notifications of planned maintenance on your AWS Direct Connect service(s).
  - Note that if you have redundant AWS Direct Connect services, you can proactively test the routing of the Outposts service link over each likely network path under maintenance conditions. This allows you to test if an interruption to one of your AWS Direct Connect services could lead to asymmetric routing of the service link. The resiliency of the AWS Direct Connect portion of the end-to-end network connectivity can be tested by the AWS Direct Connect Resiliency with Resiliency Toolkit. For more information, see Testing AWS Direct Connect Resiliency with Resiliency Toolkit – Failover Testing.

After you have gone through the preceding checklist and pinpointed asymmetric routing of the service link as a possible root cause, there are a number of further actions you can take:

- Restore symmetric routing by reverting any corporate network changes or waiting for a provider planned maintenance to complete.
- Log in to one or both firewalls and clear all flow state information for all flows from the command-line (if supported by the firewall vendor).
- Temporarily filter out BGP announcements through one of the firewalls or shut the interfaces on one firewall in order to force symmetric routing through the other firewall.

- Reboot each firewall in turn to eliminate potential corruption in the flow-state tracking of the service link traffic in the firewall's memory.
- Engage your firewall vendor to either verify or relax the tracking of UDP flow-state for UDP connections sourced on port 443 and destined to port 443.

AWS Outposts end-of-term options

At the end of your AWS Outposts term, you have three options:

- Renew your subscription and keep your existing Outpost.
- End your subscription and prepare your Outpost racks for return.
- Convert to a month-to-month subscription and keep your existing Outpost.

Topics

Renew your subscription

To renew your subscription and keep your existing Outpost:

Complete the following steps at least **30 days** before your Outpost's term ends:

1. Sign in to the AWS Support Center Console.
2. Choose **Create case**.
3. Choose **Account and billing**.
4. For **Service**, choose **Billing**.
5. For **Category**, choose **Other Billing Questions**.
6. For **Severity**, choose **Important question**.
7. Choose **Next step: Additional information**.
8. On the **Additional information** page, for **Subject**, enter your request to renew such as **`Renew my Outpost subscription`**.
9. For **Description**, enter one of the following payment options:
    - No upfront
    - Partial upfront

- All upfront

For pricing, see AWS Outposts rack pricing. You can also request a price quote.

10. Choose **Next step: Solve now or contact us**.
11. On the **Contact us** page, choose your preferred language.
12. Choose your preferred contact method.
13. Review your case details and then choose **Submit**. Your case ID number and summary appear.

AWS Customer Support will initiate the subscription renewal process. Your new subscription will start the day after your current subscription ends.

If you do not indicate that you want to renew your subscription or return your Outpost rack, you will be converted to a month-to-month subscription automatically. Your Outpost will be renewed on a monthly basis at the rate of the **No Upfront** payment option that corresponds to your AWS Outposts configuration. Your new monthly subscription will start the day after your current subscription ends.

End your subscription and prepare racks for return

> **Important**
>
> AWS cannot begin the return process until you have completed the following procedures. We cannot stop the return process after you have opened a support case to end your subscription.

To end your subscription:

Complete the following steps at least **30 days** before your Outpost's term ends:

1. Sign in to the AWS Support Center Console.
2. Choose **Create case**.
3. Choose **Account and billing**.
4. For **Service**, choose **Billing**.
5. For **Category**, choose **Other Billing Questions**.

6. For **Severity**, choose **Important question**.
7. Choose **Next step: Additional information**.
8. On the **Additional information** page, for **Subject**, enter a clear request, such as **`End my Outpost subscription`**.
9. For **Description**, enter the date that you prefer to have the Outpost retrieved.
10. Choose **Next step: Solve now or contact us**.
11. On the **Contact us** page, choose your preferred language.
12. Choose your preferred contact method.
13. Review your case details and then choose **Submit**. Your case ID number and summary appear.

AWS Customer Support will contact you to coordinate the retrieval.

To prepare your AWS Outposts racks for return:

> **Important**
>
> Do not power down the Outpost rack until AWS is on-site for the scheduled retrieval.

1. If the Outpost's resources are shared, you must unshare these resources.

   You can unshare a shared Outpost resource in one of the following ways:

   - Use the AWS RAM console. For more information, see Updating a resource share in the *AWS RAM User Guide*.
   - Use the AWS CLI to run the disassociate-resource-share command.

   For the list of Outpost resources that can be shared, see Shareable Outpost resources.
2. Terminate the active instances associated with subnets on your Outpost. To terminate the instances, follow the instructions in Terminate your instance in the *Amazon EC2 User Guide*.

   > **Note**
   >
   > Some AWS-managed services running on your Outpost, such as Application Load Balancers or Amazon Relational Database Service (RDS), consume EC2 capacity. However, their associated instances aren't visible on the Amazon EC2 dashboard.

You must terminate the resources tied to these services to free up capacity. For more information, see [Why is some EC2 instance capacity missing on my Outpost?].

3. Verify the instance-capacity-availability of your Amazon EC2 instances in your AWS account.

   a. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.

   b. Choose **Outposts**.

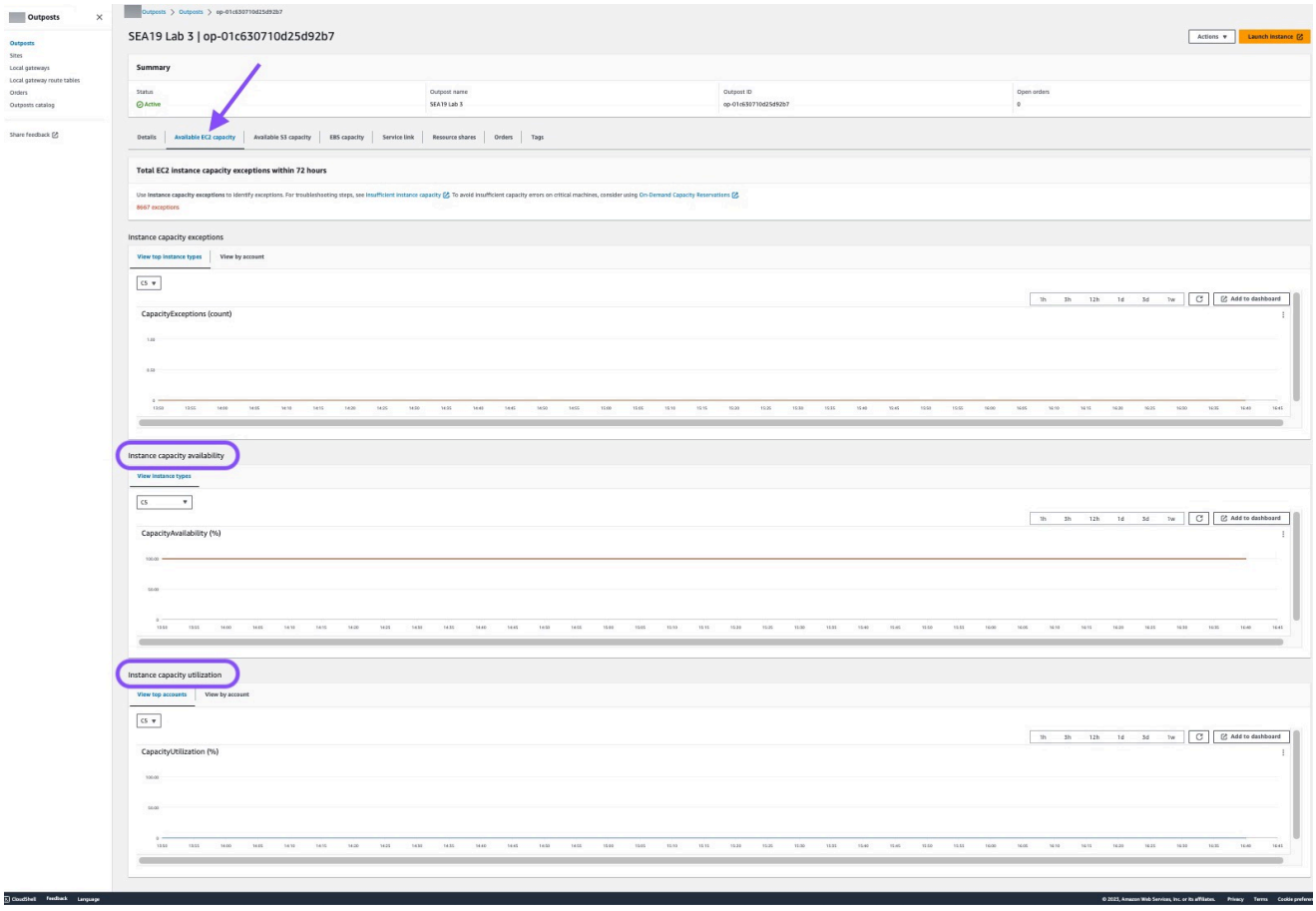
   c. Choose the specific Outpost you are returning.

   d. On the page for the Outpost, choose the **Available EC2 capacity** tab.

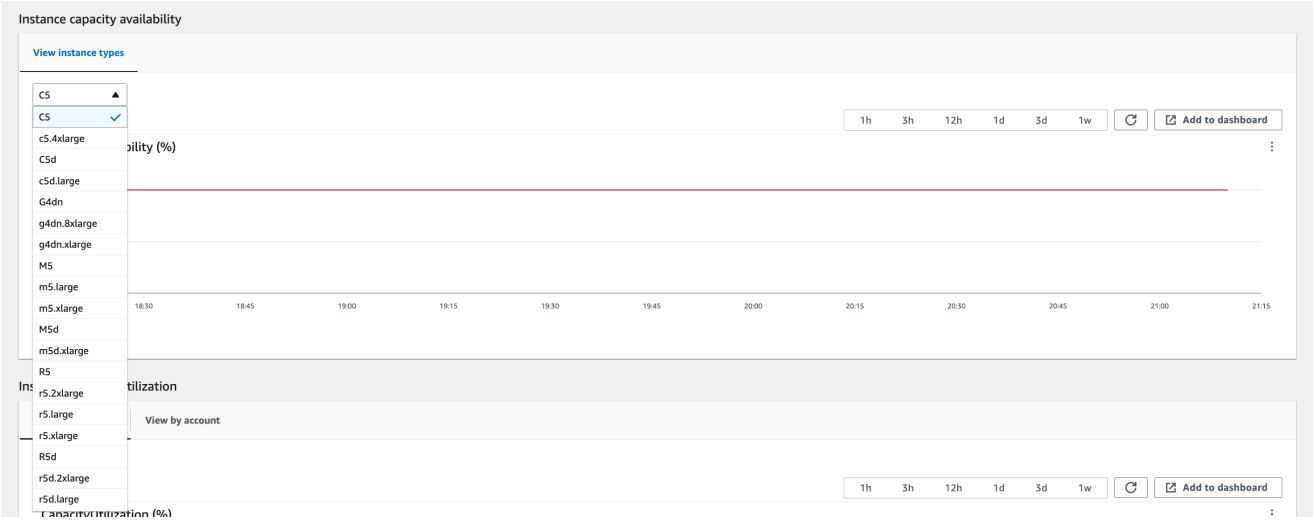
   e. Ensure that the **Instance capacity availability** is at 100% for each instance family.

   f. Ensure that the **Instance capacity utilization** is at 0% for each instance family.

   The following image shows the **Instance capacity availability** and **Instance capacity utilization** graphs on the **Available EC2 capacity** tab.



   The following image shows the list of instance types.



4. Create backups of your Amazon EC2 instances and server volumes. To create the backups, follow the instructions in Backup and recovery for Amazon EC2 with EBS volumes in the *AWS Prescriptive Guidance* guide.
5. Delete the Amazon EBS volumes associated with your Outpost.
   a. Open the Amazon EC2 console console at https://console.aws.amazon.com/ec2/.
   b. From the navigation pane, choose **Volumes**.
   c. Choose **Actions** and **Delete volume**.
   d. In the confirmation dialog box, choose **Delete**.
6. If you have Amazon S3 on Outposts, delete any local snapshots on the Outposts.
   a. Open the Amazon EC2 console console at https://console.aws.amazon.com/ec2/.
   b. From the navigation pane, choose **Snapshots**.
   c. Select the snapshots with an Outpost ARN.
   d. Choose **Actions** and **Delete snapshots**.
   e. In the confirmation dialog box, choose **Delete**.
7. Delete any Amazon S3 buckets associated with your Outpost. To delete the buckets, follow the instructions in Deleting your Amazon S3 on Outposts bucket in the *Amazon Simple Storage Service User Guide*.
8. Delete any VPC associations and customer-owned IP address pool (CoIP) CIDRs associated with your Outpost.

An AWS retrieval team will power down the rack. After it's powered down, you can destroy the AWS Nitro Security Key or the AWS retrieval team can do so on your behalf.

Convert to a month-to-month subscription

To convert to a month-to-month subscription and keep your existing Outpost, no action is needed. If you have questions, open a billing support case.

Your Outpost will be renewed on a monthly basis at the rate of the **No Upfront** payment option that corresponds to your AWS Outposts configuration. Your new monthly subscription will start the day after your current subscription ends.

Quotas for AWS Outposts

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, but not for all quotas.

To view the quotas for AWS Outposts, open the Service Quotas console. In the navigation pane, choose **AWS services**, and select **AWS Outposts**.

To request a quota increase, see Requesting a quota increase in the *Service Quotas User Guide*.

Your AWS account has the following quotas related to AWS Outposts.

Resource	Default	Adjustable	Comments
Outpost sites	100	Yes	An Outpost site is the customer managed physical building where you power and attach your Outpost equipment to the network. You can have 100 Outposts sites in each Region of your AWS account.
Outposts per site	10	Yes	AWS Outposts includes hardware and virtual resources, known as Outposts. This quota limits your Outpost virtual resources. You can have 10 Outposts in each Outpost site.

AWS Outposts and the quotas for other services

AWS Outposts relies on the resources of other services and those services may have their own default quotas. For example, your quota for local network interfaces comes from the Amazon VPC quota for network interfaces.

Document history

The following table describes important changes to the *AWS Outposts User Guide*.

Change	Description	Date
Capacity management	You can modify the default capacity configuration for your new Outposts order.	April 16, 2024
AWS Outposts rack supports service link interface throughput metrics	You can now monitor throughput usage between your Outpost rack service link virtual interfaces (VIFs) and your local network devices, by leveraging `IfTrafficIn` and `IfTrafficOut` Amazon CloudWatch metrics.	November 17, 2023
Intra-VPC communication across AWS Outposts with local gateway	You can establish communication between subnets in the same VPC across different Outposts with local gateways.	August 30, 2023
End-of-term options for AWS Outposts racks	At the end of your AWS Outposts term, you can renew, end, or convert your subscription.	August 1, 2023
Amazon Route 53 on Outposts is available on AWS Outposts racks.	Amazon Route 53 on Outposts includes a Resolver that caches all DNS queries that originate from the AWS Outposts. You can also set up hybrid connectivity between an Outpost and an on-premises DNS resolver	July 20, 2023

	when you deploy inbound and outbound endpoints.	
Local gateway inbound routes	You can create and modify local gateway inbound routes to elastic network interfaces on your Outpost.	September 15, 2022
Introducing direct VPC routing for AWS Outposts	Uses the private IP address of instances in your VPC to facilitate communication with your on-premises network.	September 14, 2022
Created AWS Outposts User Guide for Outposts rack	AWS Outposts User Guide broke into separate guides for rack and servers.	September 14, 2022
Create and manage local gateway route tables	Create and modify local gateway route tables and CoIP pools. Manage VIF group associations.	September 14, 2022
Placement groups on AWS Outposts	Placement groups that use a spread strategy can distribute instances across hosts.	June 30, 2022
Dedicated Hosts on AWS Outposts	You can now use Dedicated Hosts on Outposts.	May 31, 2022
Shared Outpost sites	Create and manage Outpost sites and share them with other AWS accounts in your organization.	October 18, 2021
New CloudWatch dimension	A new CloudWatch dimension for metrics in the AWS Outposts namespace.	October 13, 2021

Share S3 buckets	Share and manage S3 buckets on your Outpost.	August 5, 2021
Support for some placement groups	You can use cluster, partition, or spread placement strategie s just as you would in a Region.	July 28, 2021
Additional CloudWatch metrics	Additional CloudWatch metrics are available for Reserved Instances.	May 24, 2021
Network troubleshooting checklist	A network troubleshooting checklist is available.	February 22, 2021
Additional CloudWatch metrics	Additional CloudWatch metrics for EBS volumes are available.	February 2, 2021
Console ordering updates	The console ordering process is updated.	January 14, 2021
Private connectivity	You can configure private connectivity for your Outpost when you create it in the AWS Outposts console.	December 21, 2020
Network readiness checklist	Use the network readiness checklist when you are gathering the information for your Outpost configuration.	October 28, 2020

Shared AWS Outposts resources	With Outpost sharing, Outpost owners can share their Outposts and Outpost resources, including local gateway route tables, with other AWS accounts under the same AWS organization.	October 15, 2020
Additional CloudWatch metrics	Additional CloudWatch metrics for instance type counts are available.	September 21, 2020
Additional CloudWatch metric	An additional CloudWatc h metric for service link connected status is available.	September 11, 2020
Support for sharing customer-owned IPv4 addresses	Use AWS Resource Access Manager to share customer-owned IPv4 addresses.	April 20, 2020
Additional CloudWatch metrics	Additional CloudWatch metrics for EBS volumes are available.	April 4, 2020
Initial release	This is the initial release of AWS Outposts.	December 3, 2019