
AWS Outposts

User Guide for racks



AWS Outposts: User Guide for racks

Copyright © 2022 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?	1
Key concepts	1
AWS resources on Outposts	1
Pricing	3
How AWS Outposts works	4
Network components	4
VPCs and subnets	5
Routing	5
DNS	6
Service link	6
Local gateways	6
Local network interfaces	7
Requirements	8
Facility	8
Networking	9
Network readiness checklist	10
Power	13
Order fulfillment	14
Get started	16
Create an Outpost and order capacity	16
Order fulfillment for rack	14
Launch an instance	17
Step 1: Create a subnet	18
Step 2: Launch an instance on the Outpost	18
Step 3: Configure connectivity	21
Step 4: Test the connectivity	21
Service link	25
Connectivity through service links	25
Service link private connectivity using VPC	27
Prerequisites	27
Redundant internet connections	28
Outposts and sites	29
Outposts	29
Sites	30
Local gateway	33
Local gateway basics	33
Internet connectivity through the local gateway	34
Working with the local gateway	34
View and tag local gateway	34
Local gateway route tables	35
Direct VPC routing for AWS Outposts	35
Customer-owned IP addresses	38
Working with local gateway route tables	42
Rack local connectivity	50
Physical connectivity	51
Link aggregation	51
Virtual LANs	52
Network layer connectivity	53
Service link BGP connectivity	54
Service link infrastructure subnet advertisement and IP range	55
Local gateway BGP connectivity	55
Local gateway customer-owned IP subnet advertisement	56
Working with shared resources	58
Shareable Outpost resources	58

Prerequisites for sharing Outposts resources	59
Related services	59
Sharing across Availability Zones	59
Sharing an Outpost resource	60
Unsharing a shared Outpost resource	61
Identifying a shared Outpost resource	61
Shared Outpost resource permissions	61
Permissions for owners	61
Permissions for consumers	62
Billing and metering	62
Limitations	62
Security	63
Data protection	63
Encryption at Rest	63
Encryption in transit	64
Data deletion	64
Identity and access management	64
Policy structure	64
Example policies	65
Using temporary credentials with AWS Outposts	66
Service-linked roles	66
Using service-linked roles	67
Infrastructure security	69
Resilience	69
Compliance validation	70
Monitoring	71
CloudWatch metrics	71
Outpost metrics	72
Outpost metric dimensions	75
View CloudWatch metrics for your outpost	75
Logging AWS Outposts API calls with AWS CloudTrail	76
AWS Outposts information in CloudTrail	76
Understanding AWS Outposts log file entries	77
Maintenance	79
Hardware maintenance	79
Firmware updates	79
Planned and unplanned power down	80
Optimization	80
Amazon EC2 Dedicated Hosts on Outpost	80
Setup instance recovery or auto scaling	81
Placement groups on Outpost	81
Rack network troubleshooting	85
Connectivity with Outpost network devices	86
AWS Direct Connect public virtual interface connectivity to AWS Region	87
AWS Direct Connect private virtual interface connectivity to AWS Region	88
ISP public internet connectivity to AWS Region	88
Quotas	90
AWS Outposts and other services Service Quotas	90
Document history	91

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.

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 configurations** – Configurations of Amazon EC2 compute capacity, Amazon EBS storage capacity, and networking support. Each configuration has unique power, cooling, and weight support requirements.
- **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.
- **Outpost 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.
- **Outpost 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** – 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:

Resource type	Racks	Servers
Amazon EC2 instances – Launch an instance on your Outpost rack (p. 17)	✔ Yes	✔ Yes
Amazon ECS clusters – Amazon Elastic Container Service on AWS Outposts	✔ Yes	✔ Yes
Amazon EKS nodes – Amazon Elastic Kubernetes Service on AWS Outposts	✔ Yes	
AWS App Mesh Envoy proxy – AWS App Mesh on AWS Outposts	✔ Yes	✔ Yes
Storage		
Amazon EC2 instance block storage – Amazon EC2 instance store in the <i>Amazon EC2 User Guide for Linux Instances</i> and Amazon EC2 instance store in the <i>Amazon EC2 User Guide for Windows Instances</i>	✔ Yes	✔ Yes
EBS volumes – Launch an instance on your Outpost rack (p. 17)	✔ Yes	
Amazon S3 buckets – Using Amazon S3 on AWS Outposts	✔ Yes	
Analytics and Database		
Amazon EMR clusters – EMR Clusters on AWS Outposts	✔ Yes	
Amazon ElastiCache instances – Using Outposts in the <i>Amazon ElastiCache for Redis User Guide</i> , Using Outposts in the <i>Amazon ElastiCache for Memcached User Guide</i>	✔ Yes	
Amazon RDS DB instances – Amazon RDS on AWS Outposts	✔ Yes	
Networking, AWS IoT, and Amazon Machine Learning		
Amazon VPC – Subnets in AWS Outposts	✔ Yes	✔ Yes
Application Load Balancers – Subnets for your load balancer	✔ Yes	
AWS IoT Greengrass	✔ Yes	✔ Yes

Resource type	Racks	Servers
Amazon SageMaker Neo	 Yes	 Yes

Pricing

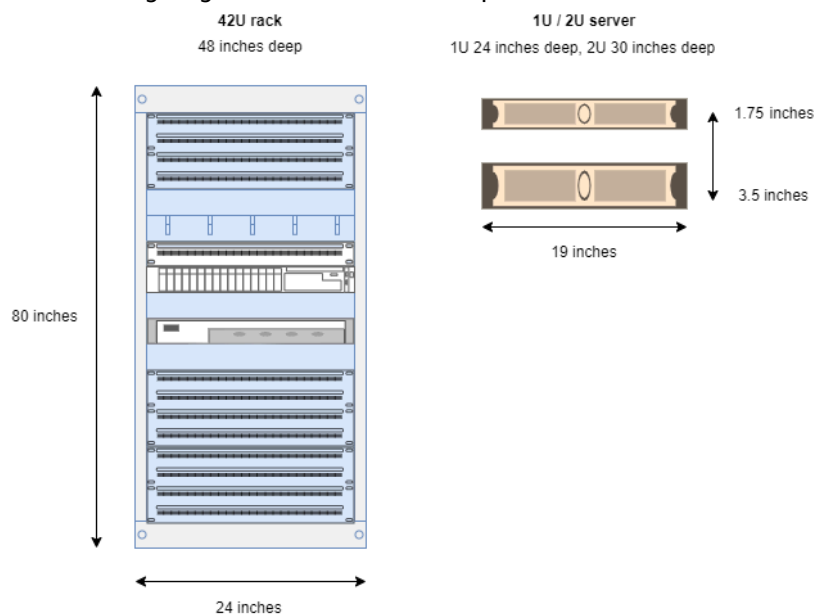
You can choose from a variety of Outpost configurations, each providing a combination of EC2 instance types and storage options. The price for rack configurations includes installation, removal, and maintenance. For servers, you must install and maintain the equipment.

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 the [AWS Outposts pricing page](#).

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

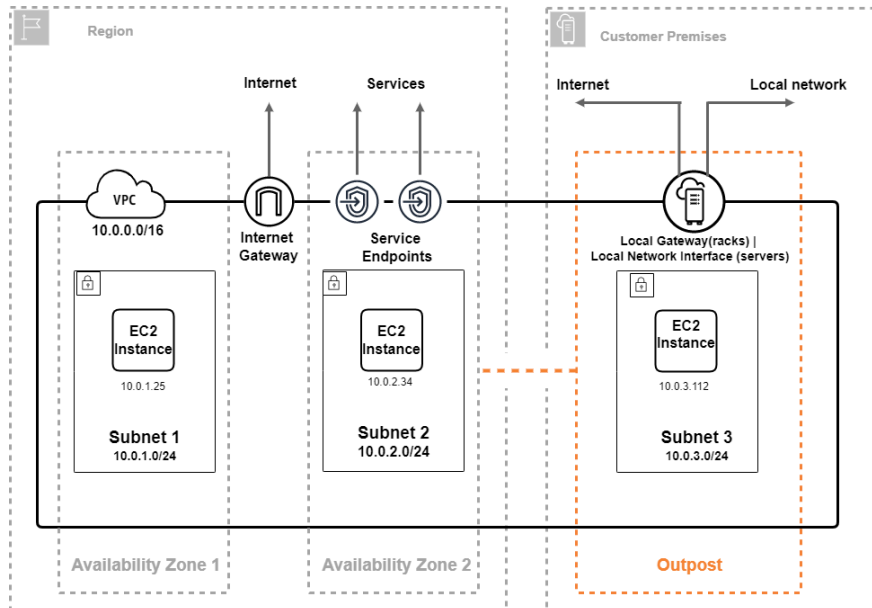
- [Network components \(p. 4\)](#)
- [VPCs and subnets \(p. 5\)](#)
- [Routing \(p. 5\)](#)
- [DNS \(p. 6\)](#)
- [Service link \(p. 6\)](#)
- [Local gateways \(p. 6\)](#)
- [Local network interfaces \(p. 7\)](#)

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
- A local gateway for [racks](#) (p. 6), or a local network interface for [servers](#) (p. 7)



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. You cannot configure a more specific range than the VPC CIDR local route on the Outpost for Outpost subnets.

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.

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 [Service link \(p. 25\)](#).

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 \(p. 33\)](#).

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 Outpost servers.

Site requirements for Outpost 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 are Outposts racks available?*

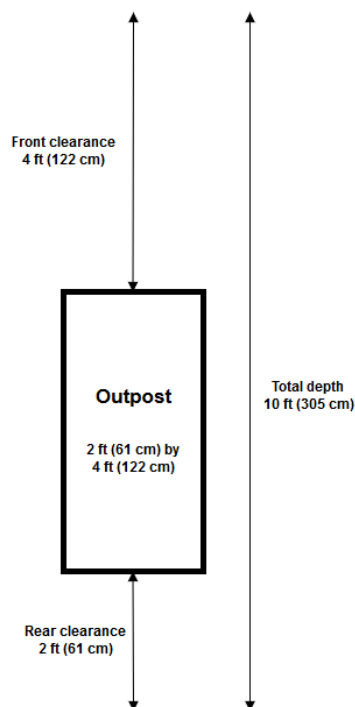
This topic offers Outpost rack requirements. For server requirements, see [Site requirements for Outpost servers](#) in the AWS Outposts User Guide for Outpost servers.

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.
- **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.

Tip

For bandwidth recommendations for the service link connection, see [Bandwidth recommendations](#) (p. 25).

- 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 \(p. 51\)](#).

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
40 Gbps	MMF	– 40GBASE-ESR4 – 40GBASE-SR4

Uplink speed	Fiber type	Optical standard
100 Gbps	SMF	<ul style="list-style-type: none"> – 100G PSM4 MSA – 100GBASE-CWDM4 – 100GBASE-LR4
100 Gbps	MMF	<ul style="list-style-type: none"> – 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 \(p. 52\)](#).

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.

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 \(p. 53\)](#).

Outpost network device	Service link requirements	Local gateway requirements
#1	<ul style="list-style-type: none"> – Service link CIDR (/30 or /31) – Service link IP address 	<ul style="list-style-type: none"> – Local gateway CIDR (/30 or /31) – Local gateway IP address
#2	<ul style="list-style-type: none"> – Service link CIDR (/30 or /31) – Service link IP address 	<ul style="list-style-type: none"> – Local gateway CIDR (/30 or /31) – Local gateway IP address

Service link maximum transmission unit (MTU)

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. AWS Outposts requires a minimum of 1500 bytes across your on-premises network. The Outpost service link supports a maximum packet

size of 1300 bytes. For more information about the service link, see [Outpost connectivity to AWS Regions \(p. 25\)](#).

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 \(p. 54\)](#).

Outpost	Service link BGP requirements
Your Outpost	<ul style="list-style-type: none"> – 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	<ul style="list-style-type: none"> – Service link BGP peer IP address. – Service link BGP peer ASN. 2-byte (16-bit) or 4-byte (32-bit).
#2	<ul style="list-style-type: none"> – 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 \(p. 55\)](#).

Outpost	Local gateway BGP requirements
Your Outpost	<ul style="list-style-type: none"> – 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	<ul style="list-style-type: none"> – Local gateway BGP peer IP address. – Local gateway BGP peer ASN. 2-byte (16-bit) or 4-byte (32-bit).
#2	<ul style="list-style-type: none"> – 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/>.

AC line voltage	Single-phase 208 to 277 VAC (50 or 60 Hz) Three-phase 346 to 480 VAC (50 to 60 Hz)
Power consumption	5 kVA (4 kW), 10 kVA (9 kW), or 15 kVA (13 kW)
AC protection (upstream power breakers)	30 A or 32 A with D-curve circuit breaker
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 or AH532P6W, 1 drop to S1 and 1 drop to S2	1 x L6-30P or IEC309, 1 drop to S1	1 x AH530P7W or AH532P6W, 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	
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. The team will roll the rack to 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 Outpost hardware.

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.

Get started with AWS Outposts

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

Tasks

- [Create an Outpost and order Outpost capacity](#) (p. 16)
- [Launch an instance on your Outpost rack](#) (p. 17)

Create an Outpost and order Outpost capacity

To begin using AWS Outposts, you must create an Outpost and order Outpost capacity. For more information about Outposts configurations, see [our catalog](#).

Prerequisites

- An Outpost site is the physical location for your Outpost equipment. Before ordering capacity, verify that your site meets the requirements for AWS Outposts. For more information, see [Site requirements for Outpost rack](#) (p. 8).
- You must have an AWS Enterprise Support plan.

To create an Outpost and order capacity

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. Choose **Place order**.
4. Choose **Racks**.
5. To add capacity, choose a capacity configuration. If the available rack capacity configurations do not meet your needs, you can [request a custom capacity configuration](#) instead.
6. Choose **Next**.
7. Do one of the following:
 - To select an existing Outpost and site, choose **Use existing Outpost**, and select an Outpost.
 - To create a new Outpost at a new or existing site, choose **Create new Outpost** and perform the following steps:
 - Enter a name and description for your Outpost.
 - Select an Availability Zone for your Outpost.
 - (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. For more information about VPC requirements, see [the section called "Prerequisites"](#) (p. 27).

Note

If you need to undo the private connectivity for your Outpost, you must contact AWS Enterprise Support. For more information, see [Service link private connectivity using VPC](#) (p. 27).

- 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 name, description, and operating address for your site.

In **Site details**, enter the information listed below about the rack, read the facility requirements, and then choose, **I have read the facility requirements**.

- **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.

8. Choose **Next**.
9. Select a payment option and delivery address.
10. Choose **Next**.
11. For **Review and order**, review the information, and choose **Place order**.

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.

Order fulfillment for rack

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 Linux Instances*. For more information applicable to Windows, see [Local Amazon EBS snapshots on AWS Outposts](#) in the *Amazon EC2 User Guide for Windows Instances*.

Prerequisite

You must have an Outpost installed at your site. For more information, see [Create an Outpost and order Outpost capacity](#) (p. 16).

Tasks

- [Step 1: Create a subnet](#) (p. 18)
- [Step 2: Launch an instance on the Outpost](#) (p. 18)
- [Step 3: Configure connectivity](#) (p. 21)
- [Step 4: Test the connectivity](#) (p. 21)

Step 1: Create a subnet

You can add Outpost subnets to any VPC in the AWS Region for the Outpost. When you do so, the VPC also spans the Outpost. For more information, see [Network components](#) (p. 4).

Note

If you are launching an instance in an Outpost subnet that has been shared with you, skip to [Step 2: Launch an instance on the Outpost](#) (p. 18). For more information about sharing subnets, see [Sharing a subnet](#) in the *Amazon Virtual Private Cloud User Guide*.

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**.
4. Select the VPC and specify an IP address range for the subnet.
5. Choose **Create**.

Step 2: 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.

You can use placement groups and customer-owned IP (CoIP) address pools. If your Outpost has been configured to use CoIP address pool, you *must* map an Elastic IP address to any instance you launch before you configure local connectivity. For more information, see [the section called "Working with customer-owned IP address pools"](#) (p. 19).

You can launch an instance as follows:

AWS Outposts console

To launch an instance 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 Amazon EC2 console.

5. Follow the steps in the Amazon EC2 Launch Instance Wizard to launch the instance in your Outpost subnet. If want to add additional volumes of instance store, use step 4 in the wizard. You can only add instance store during instance launch.

For more information applicable to Linux, see [Launching an instance using the Launch Instance Wizard](#) in the *Amazon EC2 User Guide for Linux Instances*. For more information applicable to Windows, see [Launching an instance using the Launch Instance Wizard](#) in the *Amazon EC2 User Guide for Windows Instances*.

AWS CLI

To launch an instance in your Outpost subnet

- Use `run-instances` to launch an instance in your Outpost subnet. For more information about launching an instance, see [run-instances](#) in the AWS CLI Command Reference.

Example

```
aws ec2 run-instances --image-id ami-abc12345 --count 1 --instance-type c5.large  
--key-name MyKeyPair --security-group-ids sg-1a2b3c4d --subnet-id subnet-6e7f829e
```

You can use placement groups on the Outpost. If your Outpost uses a customer-owned IP (CoIP) address pool, you must map an Elastic IP address to any instance you launch before you configure local connectivity.

Contents

- [Working with placement groups \(p. 19\)](#)
- [Working with customer-owned IP address pools \(p. 19\)](#)

Working with placement groups

Outpost racks support placement groups. Use placement groups to influence how the Amazon EC2 service should attempt to place groups of *interdependent* instances you launch on underlying hardware. You can use different strategies to meet the needs of different workloads. In Outposts, you can use cluster, partition, or spread strategies just as you would in the Region. If you have a single-rack Outpost, you can take advantage of a host spread strategy to place instances across hosts instead of racks.

For more information about working with placement groups, see [Placement groups](#) and [Placement groups on AWS Outposts](#) in the *Amazon EC2 User Guide for Linux Instances*. For Windows, see [Placement groups](#) and [Placement groups on AWS Outposts](#) in the *Amazon EC2 User Guide for Windows Instances*.

Working with customer-owned IP address pools

If your Outpost uses a customer-owned IP (CoIP) address pool, you must map an Elastic IP address to any instance you launch before you move to the next step, configure local connectivity. For more information about CoIP, see [Customer-owned IP addresses \(p. 38\)](#).

You can allocate an Elastic IP address and assign it to the instance as follows:

Amazon EC2 console

To allocate and associate an Elastic IP address with the instance

1. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.

2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate new 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**, and close the confirmation screen.
8. In the navigation pane, choose **Elastic IPs**.
9. Select an Elastic IP address, and choose **Actions, Associate address**.
10. Select the instance from **Instance**, and then choose **Associate**.

AWS CLI

To allocate and associate an Elastic IP address with the instance

1. Use `describe-coip-pools` to retrieve information about your specified customer-owned address pools. For more information, see [describe-coip-pools](#) in the *AWS CLI Command Reference*.

Note the `PoolId` return value.

Example

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

Output

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

2. Use `allocate-address` to allocate an Elastic IP address. For more information, see [allocate-address](#) in the *AWS CLI Command Reference*.

Use the `customer-owned-ipv4-pool` option with the `PoolId` returned in the previous step.

Example

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

Output

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


3. Use `associate-address` to associate the Elastic IP address with the Outpost instance. For more information, see [associate-address](#) in the *AWS CLI Command Reference*.

Example

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

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 Resources](#) in the *AWS RAM User Guide*.

Step 3: Configure connectivity

You must explicitly associate a VPC with the local gateway route table to provide connectivity between the VPC and your local network. When you create a route, you can specify IP addresses, internet gateways, local gateways, virtual private gateways, and peering connections as destinations.

To configure routing for racks

1. Associate the VPC with the local gateway route table as follows:
 - a. On the navigation pane, choose **Local gateway route tables**.
 - b. Select the route table, and then choose **Actions, Associate VPC**.
 - c. For **VPC**, select the VPC to associate with the local gateway route table.
 - d. Choose **Associate VPC**.
2. For the instance in your Outpost subnets to communicate with the local network, you must add a route with the local gateway as the next hop target to your Outpost's VPC subnet route table.
 - a. Open the Amazon VPC console at <https://console.aws.amazon.com/vpc/>.
 - b. In the navigation pane, choose **Route Tables**.
 - c. Select the route table associated with the subnet, and then choose **Actions, Edit routes**.
 - d. Choose **Add route**.
 - e. For **Destination**, enter the CIDR for the local network.
 - f. For **Target**, select the ID of the local gateway.
 - g. Choose **Save routes**.

For more information, applicable to Linux, see [Work with network interfaces](#) in the *Amazon EC2 User Guide for Linux Instances*. For more information, applicable to Windows, see [Work with network interfaces](#) in the *Amazon EC2 User Guide for Windows Instances*.

Step 4: Test the connectivity

You can test connectivity by using the appropriate use cases, as follows:

- 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 private IP address.

```
ping 10.0.3.128
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 Linux Instances*. For information about connecting to a Windows instance, see [Connect to your Windows instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

After the Outpost 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
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. Use **run-instance** to launch an instance in the subnet in the AWS Region. For more information, see [run-instances](#) in the *AWS CLI Command Reference*.

Example

```
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 10.0.1.5. 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. In the following example, the IP address is 10.0.1.5.

```
ping 10.0.1.5
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

You can test connectivity by using the appropriate use cases, as follows:

- 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 (that you created in [the section called "Step 2: Launch an instance on the Outpost" \(p. 18\)](#)). In the following example, the COIP is 172.16.0.128.

```
ping 172.16.0.128
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 Linux Instances*. For information about connecting to a Windows instance, see [Connect to your Windows instance](#) in the *Amazon EC2 User Guide for Windows Instances*.

After the Outpost 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
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. Use `run-instance` to launch an instance in the subnet in the AWS Region. For more information, see [run-instances](#) in the *AWS CLI Command Reference*.

Example

```
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. In the following example, the IP address is 10.0.0.5.

```
ping 10.0.0.5  
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
```

Outpost connectivity to AWS Regions

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

Contents

- [Connectivity through service links \(p. 25\)](#)
- [Service link private connectivity using VPC \(p. 27\)](#)
- [Redundant internet connections \(p. 28\)](#)

Connectivity through service links

During AWS Outposts provisioning, you or AWS creates a service link connection that connects 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.

If you select the private connectivity option for your Outpost, the service link VPN connection is established using an existing VPC and subnet that you specify. For more information, see [Service link private connectivity using VPC \(p. 27\)](#).

Alternatively, the Outpost is able to create the service link VPN back to the AWS Region through public Region connectivity. To do so, the Outpost needs connectivity to the AWS Region's public IP ranges, either through the public internet or AWS Direct Connect public virtual interface. This connectivity can be through specific routes in the service link VLAN, or through a default route of 0.0.0.0/0. For more information about the public ranges for AWS, see [AWS IP Address Ranges](#).

After the service link is established, the Outpost is in service and managed by AWS. The service link is used for the following traffic:

- Management traffic to the Outpost through the service link, including internal control plane traffic, internal resource monitoring, and updates to firmware and software.
- Traffic between the Outpost and any associated VPCs, including customer data plane traffic.

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. AWS Outposts requires a minimum of 1500 bytes across your on-premises network. Outpost service links support a maximum packet size of 1300 bytes.

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

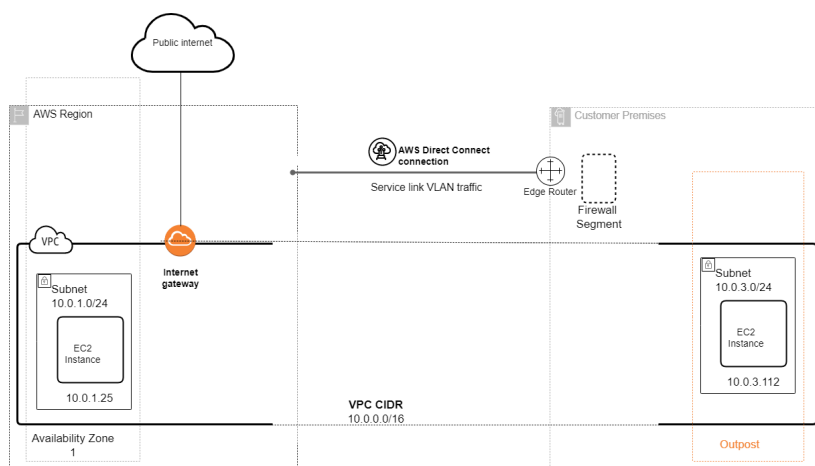
To receive a custom recommendation about the service link bandwidth required for your needs, contact your AWS sales representative or APN partner.

Firewalls and the service link

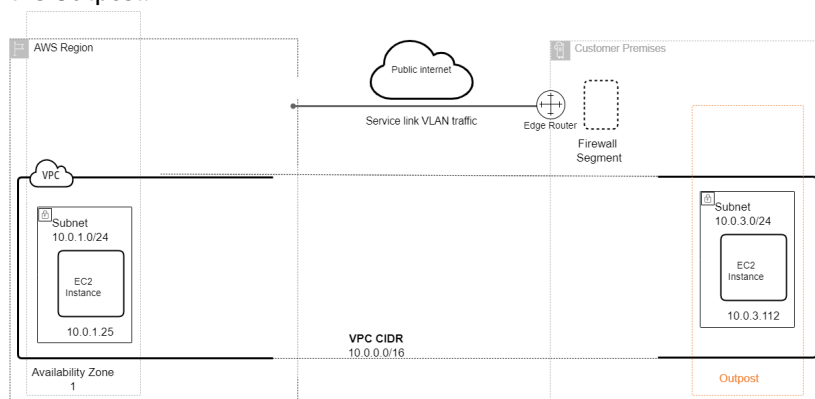
This section discusses firewall configurations and the service link connection.

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	Outpost service link /26	443	Outpost Region's public routes
TCP	1025-65535	Outpost service link /26	443	Outpost Region's public routes

Note

Instances in an Outpost cannot use the service link to communicate with instances in another Outposts if both instances are in the same VPC. Use the local gateway or local network interface to communicate between Outposts in the same VPC. Outpost racks are also designed with redundant power and networking equipment, including local gateway components. For more information, see [Resilience in AWS Outposts \(p. 69\)](#).

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.

Note

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

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 \(p. 64\)](#) and [Using service-linked roles for AWS Outposts \(p. 67\)](#).

- 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*. For other options besides AWS Direct Connect, see the [Introduction](#) to *Amazon Virtual Private Cloud Connectivity Options*.

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](#) (p. 16).

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](#) (p. 67).

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 \(p. 29\)](#)
- [Manage Outpost sites \(p. 30\)](#)

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. For more information, see [Service link private connectivity using VPC \(p. 27\)](#).

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" \(p. 30\)](#).

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 \(p. 8\)](#).

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 the name and description of on an Outpost site.

To edit the site 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 **Sites**.
4. Select the Outpost, and then choose **Actions, Edit site**.
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 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 for your Outpost rack enables connectivity from your Outpost subnets to all AWS services that are available in the parent Region, in the same way that you access them from an Availability Zone subnet. For example, you can access the Regional service endpoints over the public internet, or you can use interface VPC endpoints (AWS PrivateLink) to access them without going over the public internet. For more information, see [Outpost connectivity to AWS Regions \(p. 25\)](#).

Topics

- [Local gateway basics \(p. 33\)](#)
- [Working with the local gateway \(p. 34\)](#)
- [Local gateway route tables \(p. 35\)](#)
- [Local network connectivity for racks \(p. 50\)](#)

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” \(p. 35\)](#).
- **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 [Customer-owned IP addresses \(p. 38\)](#).
- **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 [the section called “Rack local connectivity” \(p. 50\)](#).
- **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” \(p. 48\)](#).
- **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” \(p. 48\)](#).

When AWS provisions your Outpost rack, we create some components and you are responsible for creating others. The following list summarizes the breakdown of responsibilities:

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

To create the local gateway route table, you must decide how your local gateway will connect to your on-premises network, including whether to use private IP addresses or 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-premise network. However, you can use a customer-owned IP address pool

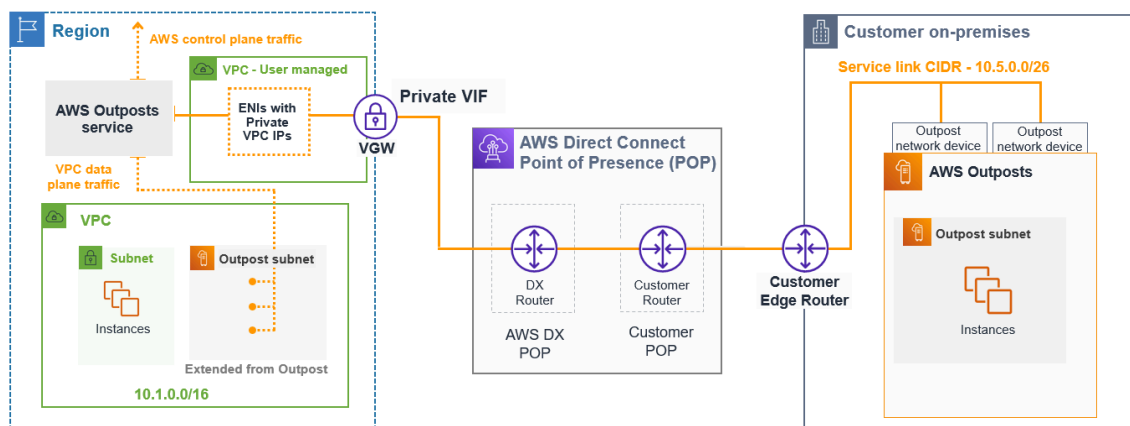
(CoIP) configuration, which supports overlapping CIDR ranges and other network topologies. To use this configuration, you must create a CoIP address pool, and the local gateway performs network address translation (NAT) for instances that have been assigned addresses from this pool. The local gateway NAT function is similar to how an internet gateway functions in an AWS Region. For more information, see [the section called "Local gateway route tables"](#) (p. 35).

Internet connectivity through the local gateway

The primary role of a local gateway is to provide connectivity from an Outpost to your local on-premises LAN. It also provides connectivity to the internet through your on-premises network. The local gateway can also provide a data plane path back to the AWS Region. If you already have connectivity between your LAN and the Region through AWS Site-to-Site VPN or AWS Direct Connect, you can use the same path to connect from the Outpost to the AWS Region privately.

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.

The following diagram shows a private connectivity configuration that uses an AWS Direct Connect connection, virtual interface, and virtual private gateway.



Working with the local gateway

The local gateway connects an Outpost *rack* to your on-premises network. Outpost servers use a different approach. For more information, see [Local network interface](#) in the AWS Outposts User Guide for Outpost servers.

View and tag local gateway

You can view the details and tag your local gateway. Tags help you identify or categorize the local gateway according to your organization's needs.

To view the details of a local gateway

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 gateways**.
4. Select the local gateway and then choose **View details**.

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

To manage the local gateway 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 gateways**.
4. Select the local gateway and then choose **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 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.

To illustrate the impact on routing, the following sections contain routing examples.

Contents

- [Direct VPC routing for AWS Outposts \(p. 35\)](#)
- [Customer-owned IP addresses \(p. 38\)](#)
- [Working with local gateway route tables \(p. 42\)](#)

Direct VPC routing for AWS Outposts

For Outpost racks, 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 have to assign an Elastic IP addresses to your EC2 instance. You have the option to use your own address space instead of direct VPC routing mode. For more information, see [Customer-owned IP addresses \(p. 38\)](#).

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.

Example topologies with direct VPC routing

The routing tables for two example network topologies using direct VPC routing on an Outpost rack.

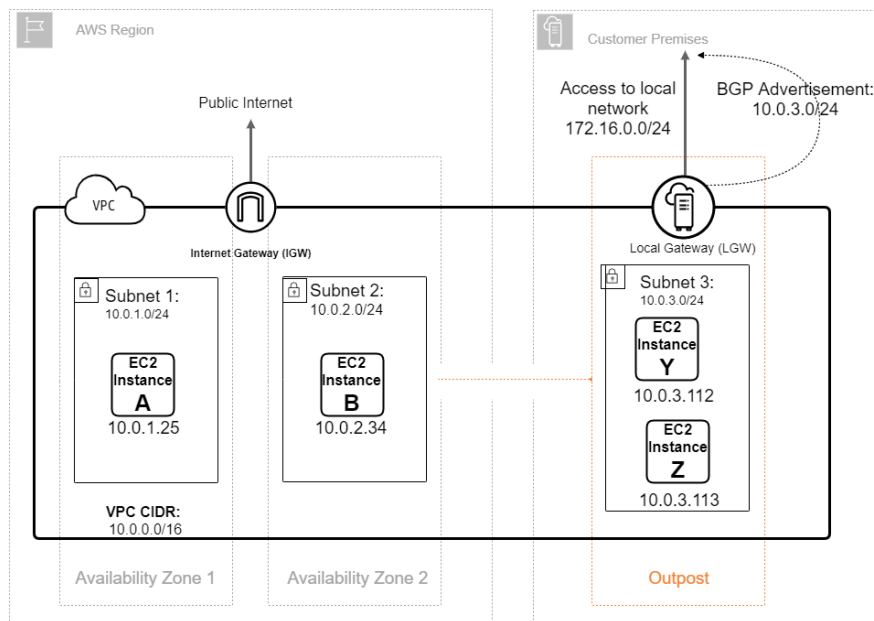
Internet connectivity through Region

Consider a scenario with the following configuration:

- A VPC in the Region with a CIDR block 10.0.0.0/16 that spans Availability Zone 1 and Availability Zone 2.
- Three subnets in the VPC, Subnet 1 in Availability Zone 1 (10.0.1.0/24), Subnet 2 in Availability Zone 2 (10.0.2.0/24), and Subnet 3 in the Outpost (10.0.3.0/24). The Outpost is homed to Availability Zone 2.
- An EC2 instance in Subnet 1 with an IP address of 10.0.1.25.
- An EC2 instance in Subnet 2 with an IP address of 10.0.2.34.
- Two EC2 instance in Subnet 3 with private IP addresses 10.0.3.112 and 10.0.3.113.
- An on-premises network CIDR of 172.16.0.0/24.
- A local gateway that uses BGP advertisement (10.0.3.0/24) to advertise the private IP addresses to the on-premises network.

Note

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.



To achieve internet connectivity through the Region, you need the following entries in the Outpost subnet route table.

Destination	Target	Type	Notes
10.0.0.0/16	Local	Defined by AWS	The local VPC route. This route allows for intra-VPC connectivity,

Destination	Target	Type	Notes
			including subnets in the AWS Region.
0.0.0.0	internet-gateway-id	Defined by the user	This route allows instances to connect to the public internet.
172.16.0.0/24	local-gateway-id	Defined by the user	This route allows the instances in Subnet 3 to connect to the on-premises network through the local gateway.

Internet connectivity through the local gateway

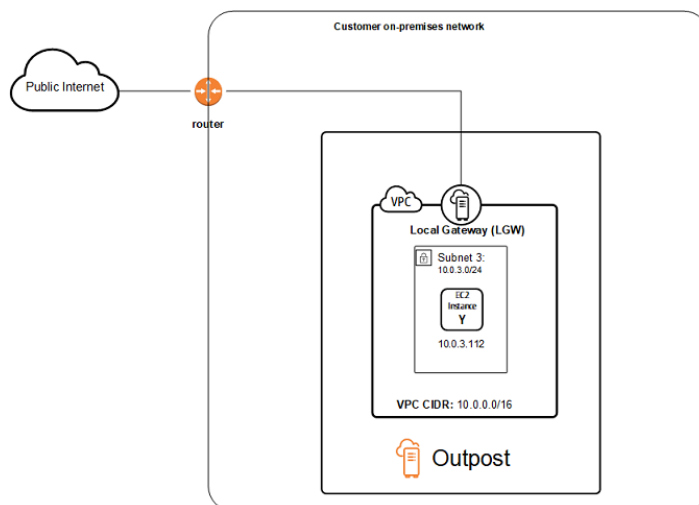
Consider a scenario with the following configuration:

- 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 local gateway that uses BGP advertisement (10.0.3.0/24) to advertise the private IP addresses to the on-premises network.

Note

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.

- A router on the customer on-premises network that performs NAT. If you route traffic from the Outpost directly to the internet, you must use NAT from the Outpost.



To achieve internet connectivity through the local gateway, you need the following entries in the Outpost subnet route table.

Destination	Target	Type	Notes
10.0.0.0/16	Local	Defined by AWS	This route allows for intra-VPC connectivity, including subnets in the Region.
0.0.0.0/0	<i>local-gateway-id</i>	Defined by the user	Instances in the subnet do not need an Elastic IP address assigned to allow for internet connectivity.

Local gateway access to the internet

The local gateway can provide access to the internet to your Outpost subnets. You configure the route table so that the local gateway routes traffic to the public internet.

Traffic initiated from the EC2 instance for the internet uses the 0.0.0.0/0 route to route traffic to the local gateway. The local gateway sends the traffic to the customer 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 instance traffic to the on-premises network

Traffic initiated from the EC2 instance with a destination of the on-premises network uses the Outpost subnet route table. The traffic routes to the local gateway, and the local gateway sends the traffic to the destination.

Inbound traffic from the on-premises network to the instance

Traffic from the on-premises network with the EC2 instance as the destination uses the private IP address. When the traffic reaches the local gateway, the local gateway sends the traffic to 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 “Working with customer-owned IP address pools” \(p. 19\)](#).

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*.

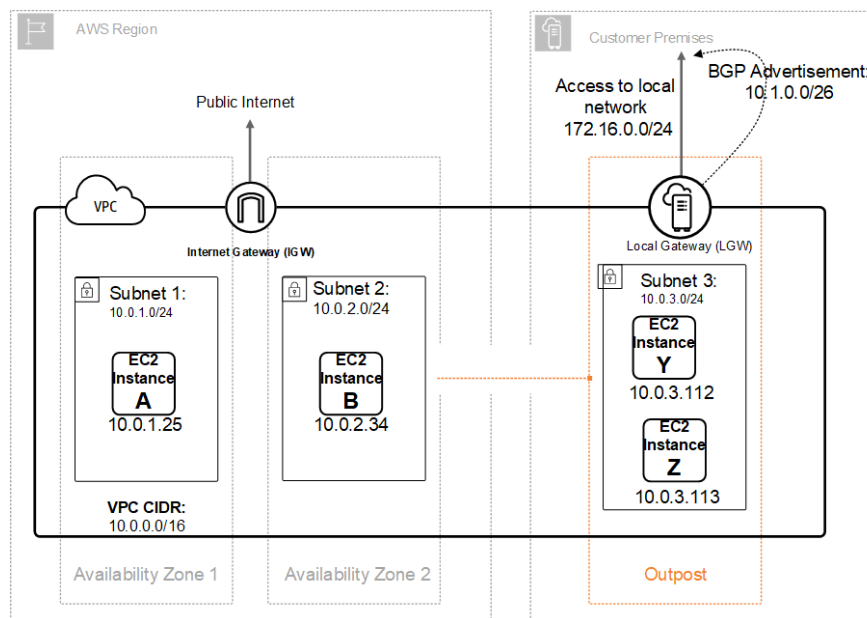
Example topologies with ColP address space

The routing tables for two example network topologies using ColP address space.

Internet connectivity through Region

Consider a scenario with the following configuration:

- A VPC with a CIDR block 10.0.0.0/16 that spans Availability Zone 1 and Availability Zone 2.
- Three subnets in the VPC, Subnet 1 in Availability Zone 1 (10.0.1.0/24), Subnet 2 in Availability Zone 2 (10.0.2.0/24), and Subnet 3 in the Outpost (10.0.3.0/24). The Outpost is homed to Availability Zone 2.
- An EC2 instance in Subnet 1 with an IP address of 10.0.1.25.
- An EC2 instance in Subnet 2 with an IP address of 10.0.2.34.
- Two EC2 instance in Subnet 3 with private IP addresses 10.0.3.112 and 10.0.3.113.
- An on-premises network CIDR of 172.16.0.0/24.
- A customer-owned IP pool (10.1.0.0/26).
- A local gateway that uses BGP advertisement (10.1.0.0/26) to advertise the customer-owned IP pool to the on-premises network.
- An Elastic IP address association that maps 10.0.3.112 to 10.1.0.2 and 10.0.3.113 to 10.1.0.3.



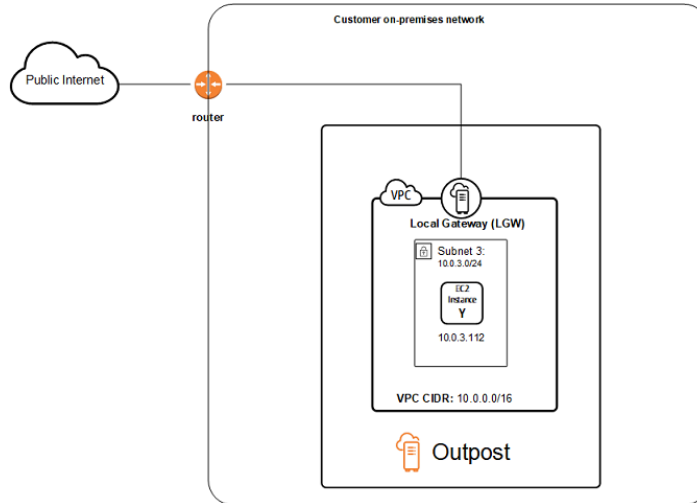
To achieve internet connectivity through the Region, you need the following entries in the Outpost subnet route table.

Destination	Target	Type	Notes
10.0.0.0/16	Local	Defined by AWS	The local VPC route. This route allows for intra-VPC connectivity, including subnets in the AWS Region.
0.0.0.0	internet-gateway-id	Defined by the user	This route allows instances to connect to the public internet. Instances in Subnet 3 need an Elastic IP address assigned to allow for internet connectivity.
172.16.0.0/24	local-gateway-id	Defined by the user	This route allows the instances in Subnet 3 to connect to the on-premises network through the local gateway.

Internet connectivity through the local gateway

Consider a scenario with the following configuration:

- 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).
- A local gateway that uses BGP advertisement (10.1.0.0/26) to advertise the customer-owned IP pool to the on-premises network.
- An Elastic IP address association that maps 10.0.3.112 to 10.1.0.2.
- A router on the customer on-premises network that performs NAT.



To achieve internet connectivity through the local gateway, you need the following entries in the Outpost subnet route table.

Destination	Target	Type	Notes
10.0.0.0/16	Local	Defined by AWS	This route allows for intra-VPC connectivity, including subnets in the Region.
0.0.0.0/0	<i>local-gateway-id</i>	Defined by the user	Instances in the subnet need an Elastic IP address assigned to allow for internet connectivity.

Local gateway access to the internet

The local gateway can provide access to the internet to your Outpost subnets. You configure the route table so that the local gateway routes traffic to the public internet.

Traffic initiated from the EC2 instance for the internet uses the 0.0.0.0/0 route to route traffic to the local gateway. The local gateway maps the EC2 instance private IP address to the customer-owned IP address (10.1.0.2), and then sends the traffic to the customer 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 instance traffic to the on-premises network

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

Inbound traffic from the on-premises network to the instance

Traffic from the on-premises network with the EC2 instance as the destination uses the customer-owned IP address (Elastic IP address). When the traffic reaches the local gateway, the local gateway maps the customer-owned IP address (Elastic IP address) to the EC2 instance IP address, and then sends the

traffic to 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 network interface's private address.

Working 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 \(p. 58\)](#).
- 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 [??? \(p. 35\)](#).
- Direct VPC routing mode does not support overlapping CIDR ranges.

Topics

- [View local gateway route table details \(p. 42\)](#)
- [Create custom local gateway route tables \(p. 43\)](#)
- [Manage local gateway route table routes \(p. 44\)](#)
- [Manage local gateway route table tags \(p. 46\)](#)
- [Switch local gateway route table modes or delete a local gateway route table \(p. 46\)](#)
- [Manage CoIP pools \(p. 46\)](#)
- [VIF group associations \(p. 48\)](#)
- [VPC associations \(p. 48\)](#)

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.
- 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.

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 for Linux Instances*.
- 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. 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, Delete local gateway route table**.
5. In the confirmation dialog box, type **delete** and then choose **Delete**.
6. (Optional) Create a local gateway route table with a new mode.
 - a. Choose **Create local gateway route table**.
 - b. Configure the local gateway route table using the new mode. For more information, see [Create custom local gateway route tables \(p. 43\)](#).

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 \(p. 38\)](#).

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 \(p. 46\)](#).

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.

Topics

- [Physical connectivity \(p. 51\)](#)
- [Link aggregation \(p. 51\)](#)
- [Virtual LANs \(p. 52\)](#)
- [Network layer connectivity \(p. 53\)](#)
- [Service link BGP connectivity \(p. 54\)](#)
- [Service link infrastructure subnet advertisement and IP range \(p. 55\)](#)
- [Local gateway BGP connectivity \(p. 55\)](#)
- [Local gateway customer-owned IP subnet advertisement \(p. 56\)](#)

Physical connectivity

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

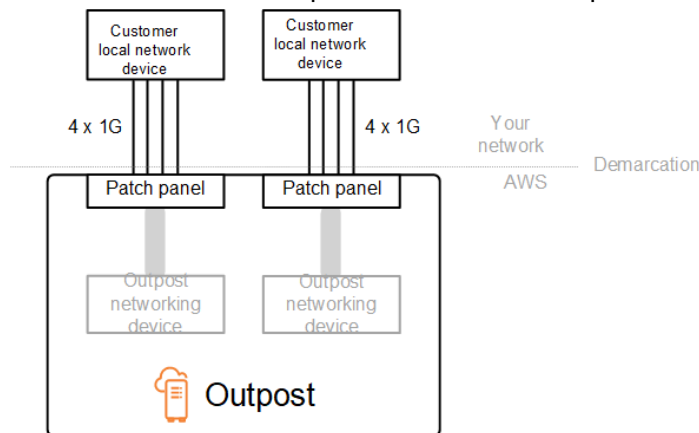
An Outpost requires a minimum of two physical links between these Outpost network devices and your local network devices. An Outpost supports the following uplink speeds and quantities for each Outpost network device.

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

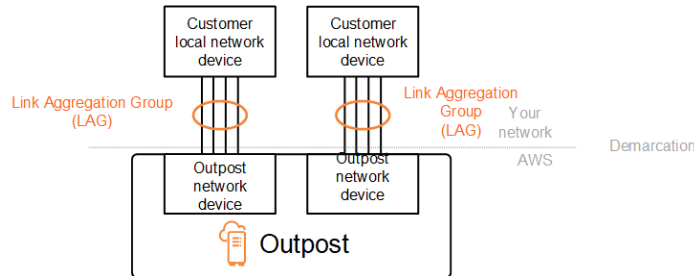
AWS Outposts uses the Link Aggregation Control Protocol (LACP) to establish two link aggregation group (LAG) connections, one from each Outpost network device to each local network device. The links from each Outpost network device are aggregated into an Ethernet LAG to represent a single network connection. 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

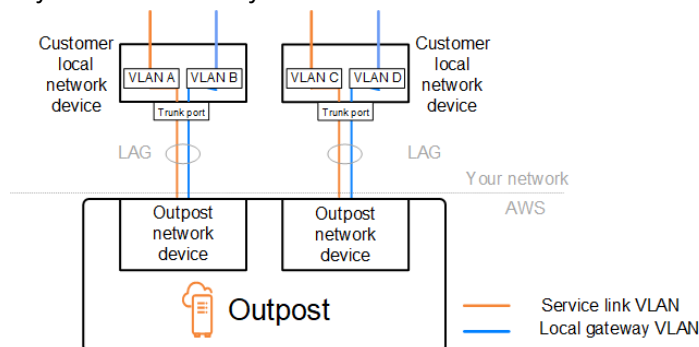
Each LAG between an Outpost network device and a local network device must be configured as an IEEE 802.1q Ethernet trunk. This enables the use of multiple VLANs for network segregation between data paths.

Each Outpost has the following data paths between the on-premises network and its network:

- **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. This VLAN provides access to the AWS Region, which enables the service link connection from the Outpost to be established back to the Region. The service link is a custom VPN or VPNs from the Outpost to the Region. It is connected to the Outpost that is configured in the Availability Zone when you purchase the Outpost.
- **Local gateway VLAN** – Enables VPC traffic from your VPC to your local LAN. This VLAN enables instances running on the Outpost to communicate with your on-premises network. It also enables them to communicate with the internet through your on-premises network.

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” \(p. 10\)](#).

Network layer connectivity

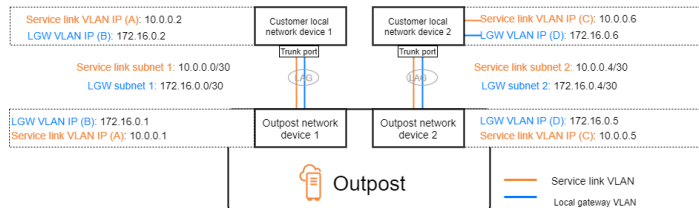
Each Outpost network device requires an IP address on each VLAN so they can communicate with the customer local network devices to establish a BGP session. We recommend that you use a dedicated subnet, with a /30 or /31 CIDR, to represent this logical point-to-point connectivity. We recommend that you do not bridge the VLANs between your customer local network devices.

You need to 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 the service link VLAN on the Outpost network device.
- **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 Outpost network device.

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

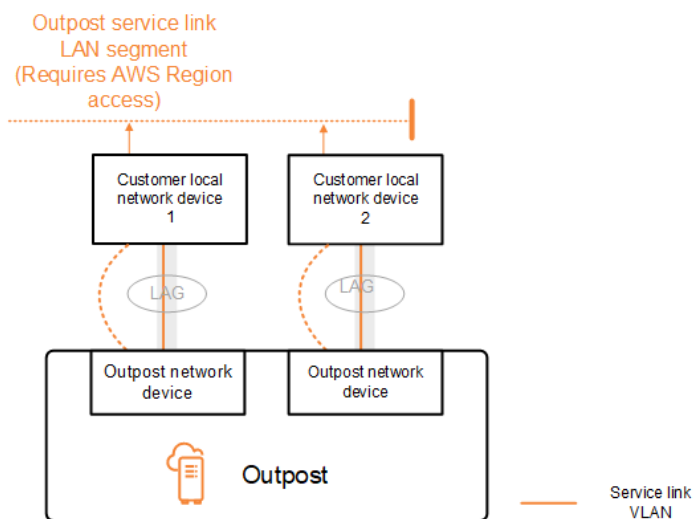
VLAN	Subnet	Customer Device 2 IP	AWS OND 2 IP
D	172.16.0.4/30	172.16.0.6	172.16.0.5

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 blocks to support link and device failures.
3. The subnet is used for connectivity from the Outpost to the AWS Region.

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

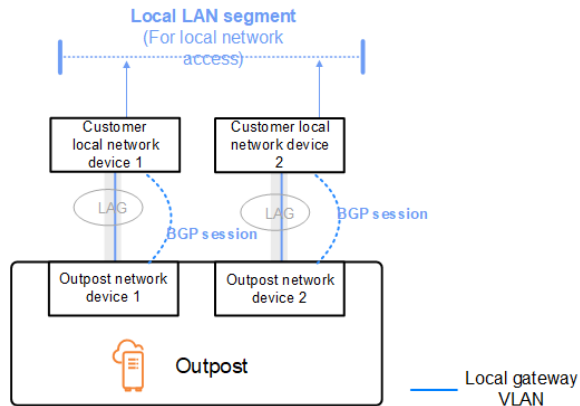
The Outpost establishes an external BGP peering from each Outpost network device to a local network device for connectivity to the local gateway. It uses a private Autonomous System Number (ASN) that you assign in order to establish the external BGP sessions. Each Outpost network device has a single 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" \(p. 53\)](#).

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](#).



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.

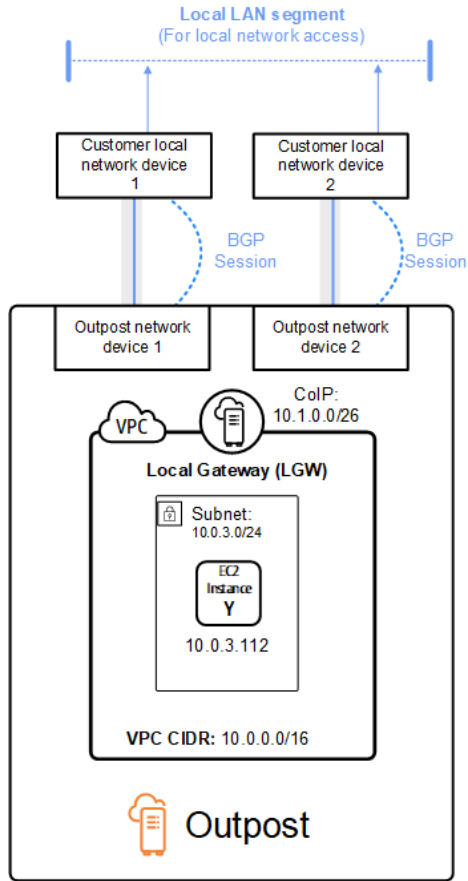
The local gateway translates the Elastic IP address to an address in the customer-owned pool. The local gateway advertises the translated address to your on-premises network, and any other network that communicates with the Outpost. The addresses are advertised on both local gateway BGP sessions to the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

- [Shareable Outpost resources \(p. 58\)](#)
- [Prerequisites for sharing Outposts resources \(p. 59\)](#)
- [Related services \(p. 59\)](#)
- [Sharing across Availability Zones \(p. 59\)](#)
- [Sharing an Outpost resource \(p. 60\)](#)
- [Unsharing a shared Outpost resource \(p. 61\)](#)
- [Identifying a shared Outpost resource \(p. 61\)](#)
- [Shared Outpost resource permissions \(p. 61\)](#)
- [Billing and metering \(p. 62\)](#)
- [Limitations \(p. 62\)](#)

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 Outpost 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 IPv4 addresses** – Consumers with access to this resource can:
 - Allocate and associate customer-owned IPv4 address 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 \(p. 58\)](#).

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 [AWS Outposts 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 \(p. 63\)](#)
- [Identity and Access Management \(IAM\) for AWS Outposts \(p. 64\)](#)
- [Infrastructure security in AWS Outposts \(p. 69\)](#)
- [Resilience in AWS Outposts \(p. 69\)](#)
- [Compliance validation for AWS Outposts \(p. 70\)](#)

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. AWS customers and APN Partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual user accounts with AWS Identity and Access Management (IAM), so that 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, encryption is enabled by default.

For Outpost racks, Amazon EBS encryption is an encryption solution for your EBS volumes and snapshots. Amazon EBS encryption uses AWS Key Management Service (AWS KMS) and KMS keys. For Outpost servers, Amazon EC2 instance store is encrypted by default.

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](#) (p. 25).

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.

For information about data deletion during required hardware maintenance, see [Hardware maintenance](#) (p. 79).

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 resources. IAM enables you to create users and groups under your AWS account. You control the permissions that users have to perform tasks using AWS resources. You can use IAM for no additional charge.

By default, IAM users don't have permissions for AWS Outposts resources and operations. To allow IAM users to manage AWS Outposts resources, you must create an IAM policy that explicitly grants them permissions, and attach the policy to the IAM users or groups that require those permissions.

When you attach a policy to a user or group of users, it allows or denies the users permission to perform the specified tasks on the specified resources. For more information, see [Policies and Permissions](#) in the *IAM User Guide*.

Before you use IAM to manage access to AWS Outposts, make sure that you understand what IAM features are available to use with AWS Outposts. To get a high-level view of how AWS Outposts and other AWS services work with IAM, see [AWS Services That Work with IAM](#) in the *IAM User Guide*.

Policy structure

An IAM policy is a JSON document that consists of one or more statements. Each statement is structured as follows.

```
{
  "Statement": [{
    "Effect": "effect",
    "Action": "action",
    "Resource": "*"
  ]
}
```



```
    "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 temporary credentials with AWS Outposts

You can use temporary credentials to sign in with federation, assume an IAM role, or assume a cross-account role. Obtain temporary security credentials by calling AWS STS API operations, such as [AssumeRole](#) or [GetFederationToken](#).

AWS Outposts supports using temporary credentials.

Service-linked roles

[Service-linked roles](#) allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

AWS Outposts supports service-linked roles. For information about creating or managing AWS Outposts service-linked roles, see [Using service-linked roles for AWS Outposts \(p. 67\)](#).

Services that require permission to manage AWS Outposts resources

Some AWS services require permissions to manage Outpost resources, such as the local gateway route table or customer owned IP (CoIP) address pools. These services can call *permission-only* actions to manage these resources. A permission-only action can be called only by an AWS service. To make these actions available, you assign a service-linked role to grant the calling service permission to manage these resources.

For example, if you assign an Amazon RDS service-linked role that adds one or more of these permissions to your DB instance, Amazon RDS can call these permission-only actions on your behalf. For more information, see [Working with Amazon RDS on AWS Outposts](#) and [Service-linked role permissions for Amazon RDS](#) in the Amazon RDS User Guide.

The following list contains permission-only actions that AWS services might call on your behalf. Consult the service-linked role of the service that you're using to determine if your service requires these actions.

CreateLocalGatewayRouteTablePermission

Grants permission to allow a service to access a local gateway route table.

DeleteLocalGatewayRouteTablePermission

Grants permission to deny a service from accessing a local gateway route table.

DescribeLocalGatewayRouteTablePermissions

Grants permission to allow a service to describe local gateway route table permissions.

CreateCoipPoolPermission

Grants permission to allow a service to access a customer owned IP (CoIP) pool.

DeleteCoipPoolPermission

Grants permission to deny a service from accessing a customer owned IP (CoIP) pool.

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 \(p. 27\)](#).

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 \(p. 61\)](#).

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](#).

Infrastructure security in AWS Outposts

As a managed service, AWS Outposts is protected by the AWS global network security procedures that are described in the [Amazon Web Services: Overview of Security Processes](#) whitepaper.

You use AWS published API calls to access AWS Outposts through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed 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.

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 the Outpost. 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 capacity utilization of your Outpost 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 AWS 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 AWS 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 Outpost racks. By doing so, this can help reduce correlated failures. For more information, see [the section called "Placement groups on Outpost" \(p. 81\)](#).

Compliance validation for AWS Outposts

AWS publishes a list of specific in scope compliance certifications for AWS Outposts. For more information, see [AWS Services in Scope by Compliance Program](#). However, these services are not in scope when running locally on AWS Outposts unless AWS Outposts is also separately listed for the specific compliance or assurance program.

Third-party auditors assess the security and compliance of AWS Outposts as part of multiple AWS compliance programs. These include ISO, PCI, HIPAA, and others.

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. This includes patching the infrastructure software and configuring infrastructure devices. As a customer, you are responsible for implementing best practices for data encryption, patching their guest operating system and applications, identity and access management, and operating system, network, and firewall configurations.

For more information about security and compliance for AWS Outposts, see [AWS Outposts FAQ](#).

AWS uses secure channels from manufacturing through installation and delivery of the Outpost equipment. When the Outpost equipment is on your site, any replacement parts are delivered through the same secure channels and are checked for tampering. No server or switch repairs occur on site.

As a customer, you are responsible for the physical security and environmental controls at the facility where the Outpost is located, and for providing networking between the Outpost and the AWS Region. Your responsibilities include the following:

- Physical and environmental security of the Outpost, starting from the moment that the Outpost equipment arrives at your facility to the point at which the Outpost equipment is removed at the end of the term or for repairs.
- Physical access controls around the Outpost equipment at your facility. This includes background checks and security training for facility staff.
- Data management policies, including terminating EC2 instances and deleting data volumes before the Outpost equipment is removed at the end of the term or for repairs.
- Configuring and maintaining a network connection between the Outpost and the AWS Region. Communication sent over this connection between the Outpost and the Region is encrypted by AWS.
- Encrypting any traffic traveling over your network to the local gateway.

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 \(p. 71\)](#).

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 \(p. 76\)](#).

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](#) (p. 72)
- [Outpost metric dimensions](#) (p. 75)
- [View CloudWatch metrics for your outpost](#) (p. 75)

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

- `OutpostId`
- `InstanceType`, `OutpostId`

`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`, `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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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, 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.

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
```

Logging AWS Outposts API calls with 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 AWS Identity and Access Management (IAM) 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:jdope",
    "arn": "arn:aws:sts::111122223333:assumed-role/example/jdope",
    "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.

Topics

- [Hardware maintenance \(p. 79\)](#)
- [Firmware updates \(p. 79\)](#)
- [Planned and unplanned power down \(p. 80\)](#)
- [Optimize Amazon EC2 for AWS Outposts \(p. 80\)](#)
- [AWS Outposts rack network troubleshooting checklist \(p. 85\)](#)

Hardware maintenance

If AWS detects an irreparable issue with hardware hosting EC2 instances running on your Outpost, we will send you instance retirement notices for the affected instances. If you stop or terminate an affected instance, the hypervisor scrubs (sets to zero) all data that was allocated to the instance from the hardware. Start the affected instance that you stopped to migrate the instance to available capacity.

If you do not stop and start an affected instance, AWS stops and starts it for you when it reaches its scheduled retirement date. For more information, see [Instance Retirement](#) in the *Amazon EC2 User Guide*.

If hardware maintenance is required, AWS will contact you to confirm a date and time for the AWS installation team to visit your Outpost site. You can schedule a visit as soon as two business days from the time that you speak 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.

Note

We recommend that you contact AWS Support before you make any changes to the physical facility where your Outpost is located that might impact the connection between your Outpost and the AWS Region. For more information, see [Creating a support case](#) in the *AWS Support User Guide*.

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.

Planned and unplanned power down

We recommend opening a case with AWS Support before making any power changes that would disrupt the connection between the Outpost site and the AWS Region. If you have an unplanned power loss, contact support. For 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.

Topics

- [Amazon EC2 Dedicated Hosts on Outpost](#) (p. 80)
- [Setup instance recovery or auto scaling](#) (p. 81)
- [Placement groups on Outpost](#) (p. 81)

Amazon EC2 Dedicated Hosts on Outpost

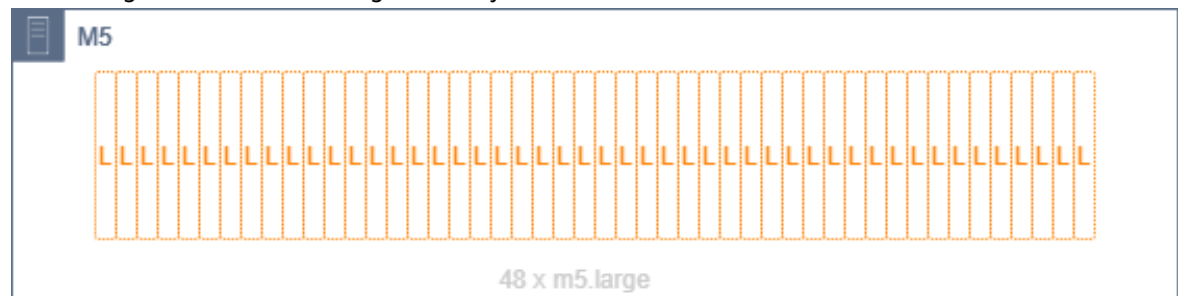
An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Your Outpost already provides you with dedicated hardware, but Dedicated Hosts allows you to use existing software licenses with per-socket, per-core, or per-VM license restrictions against a single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

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 for Linux Instances* or [Allocate Dedicated Hosts](#) *Amazon EC2 User Guide for Windows Instances*.

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 for Linux Instances* or [Understand auto-placement and affinity](#) *Amazon EC2 User Guide for Windows Instances*.

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](#) (p. 58).

Setup instance recovery or auto scaling

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 about auto-recovery, see [Recover your instance](#) in the *Amazon EC2 User Guide for Linux Instances* or [Recover your instance](#) *Amazon EC2 User Guide for Windows Instances*.

For workloads that require auto scaling, you can set up an auto scaling group to achieve the same effect. For more information about auto scaling, see [Health checks for Auto Scaling instances](#).

Placement groups on Outpost

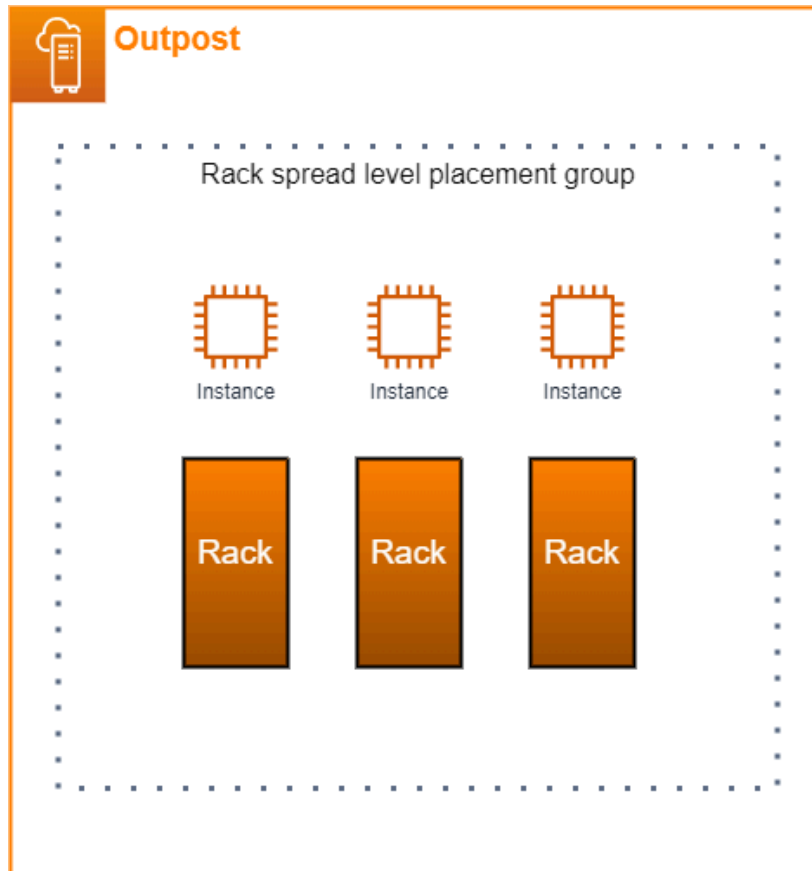
Outpost racks support placement groups. Use placement groups to influence how the Amazon EC2 service should attempt to place groups of *interdependent* instances you launch on underlying hardware. You can use different strategies to meet the needs of different workloads. In Outposts, you can use cluster, partition, or spread strategies just as you would in the Region. If you have a single-rack Outpost, you can take advantage of a host 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.

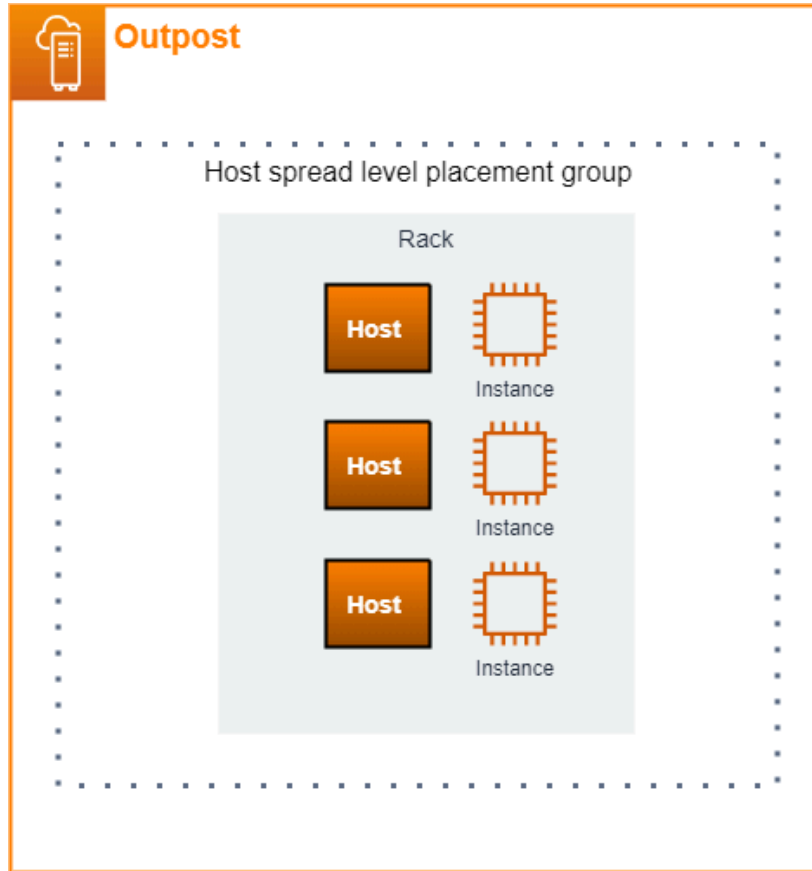
Rack spread level placement groups

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.



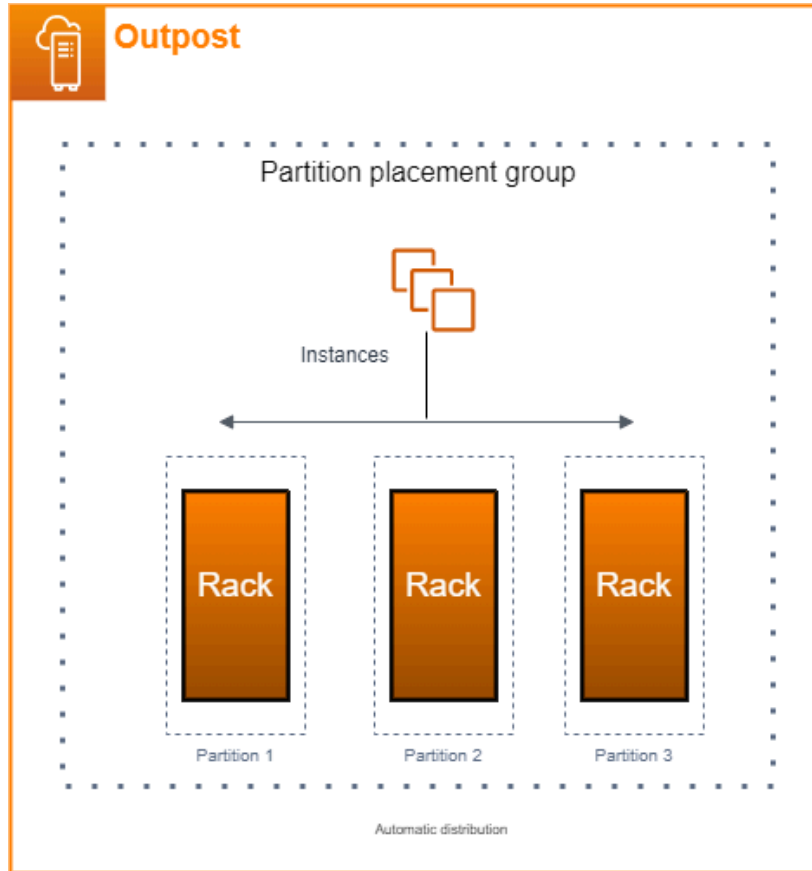
Host spread level placement groups

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.

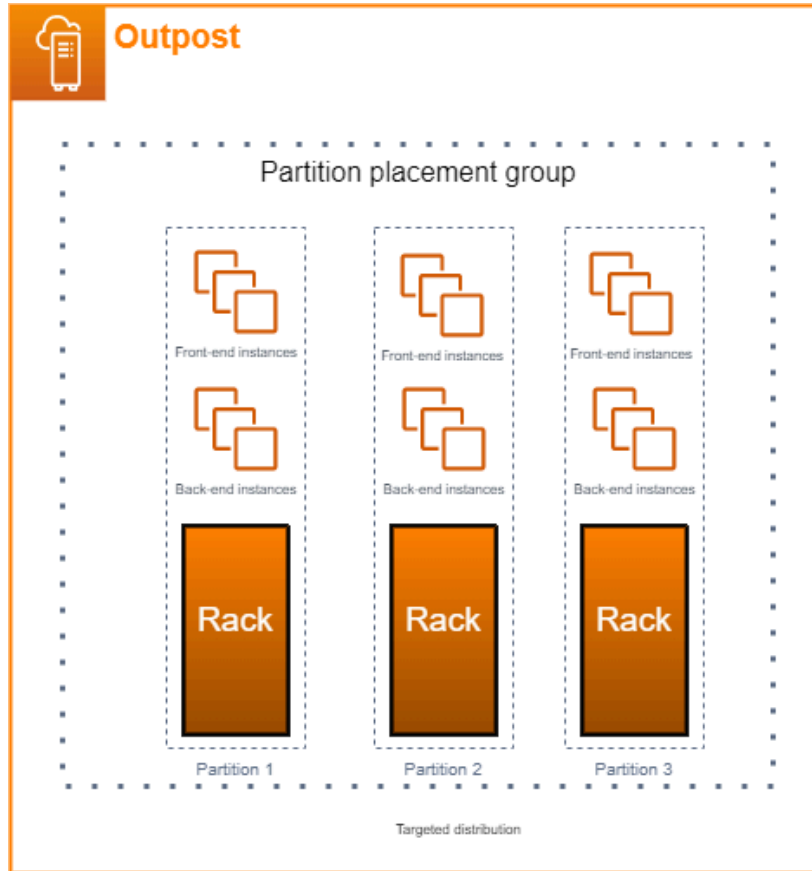


Partition placement groups

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.

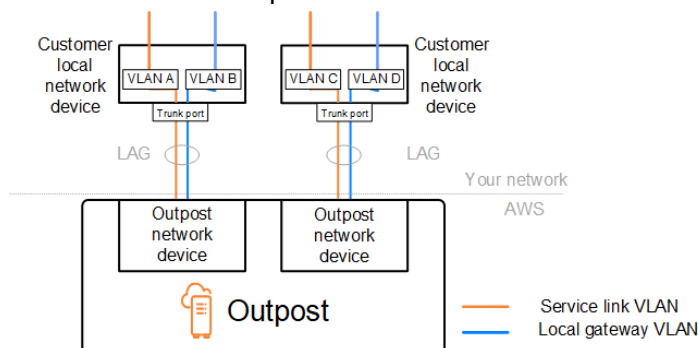


For more information about working with placement groups, see [Placement groups](#) and [Placement groups on AWS Outposts](#) in the *Amazon EC2 User Guide for Linux Instances*. For Windows, see [Placement groups](#) and [Placement groups on AWS Outposts](#) in the *Amazon EC2 User Guide for Windows Instances*.

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

If your service link is down, 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 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 \(p. 10\)](#) 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 DOWN, complete steps c and d. If 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 by confirming:
 - a. That the local Autonomous System Number (Customer ASN) is correctly configured.
 - b. That the remote Autonomous System Number (Outpost ASN) is correctly configured.
 - c. That the Interface IP and remote peer IP addresses are correctly configured.
 - d. 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 items:
 - a. BGP and TCP debugs on the customer local network devices
 - b. BGP logs on the customer local network devices
 - c. Packet capture for the customer local network devices
6. If the issue persists, perform MTR / traceroute / packet captures from your Outpost connected router to the Outpost network device peer IPs. Use the Enterprise support plan from the AWS Support console to share the test results with AWS Support.

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 \(p. 87\)](#).
- 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 \(p. 88\)](#).
- 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 \(p. 88\)](#).

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 /26 IP 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 /26 IP range.
2. To ensure Region connectivity, check the route table for the service link VRF. It should include the AWS Public IP ranges or default route.
3. If you are not receiving the AWS Public IP ranges in the service link VRF, check the following items.
 - a. Check the AWS Direct Connect link status from the edge router or 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 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 ranges in the VRF, contact AWS Support, using the 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 range (/26 – provided by the customer) to the AWS Public IP ranges. For more information, see [Outpost connectivity to AWS Regions \(p. 25\)](#).
 - c. If the firewall is not stateful, make sure to allow the inbound flow also (from the AWS Public IP ranges to the service link /26 IPs).
 - d. If you have configured a virtual router in the firewalls, ensure that the appropriate routing is configured for traffic to and from the Outpost and AWS Region.
5. If you have configured NAT in the on-premises network to translate the Outpost's service link /26 IP ranges to your owned public IPs, check the following items.
 - a. Confirm that the NAT device is not overloaded and has free ports to allocate for new session.
 - 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 IPs. Use the Enterprise support plan from the AWS Support console to share the test results with AWS Support.

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 AWS Region is using the AWS Outposts private connectivity feature, check the following items.
 - a. Ping the remote peering AWS IP 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 uses BGP to exchange routes. Your private Amazon VPC CIDR range is advertised through this BGP session to your edge router. Similarly, the /26 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 \(p. 10\)](#).
 - e. If you have an on-premises firewall, ensure that the firewall has outbound rules allowing the service link IP 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 IPs. Use the Enterprise support plan from AWS Support to share the test results with AWS Support.

ISP public internet connectivity to AWS Region

Use the following checklist to troubleshoot edge routers connected with an ISP when public internet is in use for service link connectivity.

- Confirm that the internet link is up.
- Confirm that the public servers are accessible from your edge devices connecting with the ISP.

If the internet or public servers are not accessible through the ISP links, complete the following steps.

1. Confirm whether BGP peering status with 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 via 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 ping or use telnet to confirm the connectivity, and 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 directing 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 /26 IP 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 routing 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 ranges to the public AWS IP 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 /26 IP ranges to Public IPs. 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 traceroute results show that packets are dropping or blocked at the on-premises network, check with your network or technical team for further guidance.
- If the traceroute results show that the packets are dropping or blocked at the ISP's network, contact the ISP's technical support team.
- If the traceroute results do not show any issues, collect the results from all tests (such as MTR, telnet, traceroute, packet captures, and BGP logs) and use the Enterprise support plan from the AWS Support console to contact AWS Support.

Service Quotas for AWS Outposts

This topic lists default quotas, formerly referred to as limits, for AWS Outposts. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, and other quotas cannot be increased.

- 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	<p>An Outpost site is the customer managed physical building where you power and attach your Outpost equipment to the network.</p> <p>You can have 100 Outposts sites in each Region of your AWS account.</p>
Outposts per site	10	Yes	<p>AWS Outposts includes hardware and virtual resources, known as Outposts. This quota limits your Outpost virtual resources.</p> <p>You can have 10 Outposts in each Outpost site.</p>

AWS Outposts and other services Service Quotas

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 come out of the Amazon VPC quota for network interfaces. For more information, see [AWS service quotas](#) in the Amazon Web Services General Reference.

Document history

The following table describes important changes to the *AWS Outposts User Guide*.

Change	Description	Date
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 Outpost 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 strategies 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 CloudWatch 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 (p. 91)	This is the initial release of AWS Outposts.	December 3, 2019