User guide for second-generation Outposts racks

AWS Outposts

AWS Outposts: User guide for second-generation Outposts racks

Table of Contents

What is AWS Outposts?

AWS Outposts is a family of fully managed solutions delivering AWS infrastructure, AWS services, APIs, and tools to customer premises. Outposts is available in a variety of form factors, from 1U and 2U Outposts servers to 42U Outposts racks. With Outposts, you can run the supported AWS services locally and connect to a broad range of services available in the AWS Region. AWS Outposts supports workloads that require low latency, local data processing, data residency, and migration with local network and system interdependencies.

Second-generation Outposts racks are configured for faster processing, higher memory capacity, and increased network bandwidth than AWS Outposts racks. You start with a single compute rack paired with a network rack and then add compute racks according to business needs.

The second-generation Outposts racks support:

- Network rack with built-in resiliency and independent scaling of compute and networking
- Specialized Amazon EC2 instances with accelerated networking for ultra-low latency and high throughput workloads

Amazon EC2 instances supported

Second-generation Outposts racks currently support the following Amazon EC2 instances for a broad range of on-premises workloads:

- General purpose **M7i**,**M8i**
- Compute optimized **C7i**,**C8i**
- Memory optimized **R7i**,**R8i**

Simplified network scaling and configuration

Second-generation Outposts racks have a network rack that serves as a traffic aggregation layer for all connected compute and storage racks allowing you to:

- Scale your compute resources independently from your networking infrastructure, giving you more flexibility and cost efficiency as your workloads grow.

- Easily architect for high availability using the second-generation rack's built-in resiliency features that handles device failures.
- Define your local gateway (LGW) network configurations, including IP addresses, Virtual LAN (VLAN) and Border Gateway Protocol (BGP) settings, through the API and AWS Outposts console.

Specialized Amazon EC2 instances with accelerated networking

Second-generation Outposts racks support specialized Amazon EC2 instances with accelerated networking. These instances are built for the latency-sensitive, compute-intensive, and throughput-intensive mission-critical workloads on-premises. In addition to the Outpost logical network, these instances feature a secondary physical network with network accelerator cards connected to top of rack (TOR) switches.

The second-generation racks support the following specialized, bare-metal Amazon EC2 instances:

- Ultra-low latency with deterministic performance **bmn-sf2e**
- High throughput and low latency **bmn-cx2**

AWS Outposts racks generations

The following table lists the differences between the first-generation and second-generation Outposts racks:

	First-generation Outposts racks	Second-generation Outposts racks
Compute	M5, C5, R5, G4dn	M7i, C7i, R7i, Bmn-sf2e, Bmn-cx2
Networking	• Coupled scaling of compute and networking • User-managed scaling and resiliency setup	• Decoupled scaling of compute from networking • Built-in resiliency to handle network device failures
Locally supported services	Amazon EC2, Amazon EBS, Amazon S3, Amazon EBS	Amazon EC2, Amazon EBS, Amazon EKS, Amazon ECS,

	First-generation Outposts racks	Second-generation Outposts racks
	snapshots, Amazon EKS, Amazon ECS, Route 53 Resolver, Amazon RDS, Amazon EMR, AWS IoT Greengrass, Application Load Balancers, Amazon ElastiCache, Elastic Disaster Recovery	Amazon RDS, Amazon EMR, AWS IoT Greengrass, Application Load Balancers
Power	Supported power configurations: 5 kVA, 10 kVA, or 15 kVA	Supported power configurations: 10 kVA, 15 kVA, 30 kVA
Minimum footprint	One compute rack	Two racks: one compute and one network

Key concepts

These are the key concepts:

- **Outpost site** – The customer-managed physical buildings where AWS will install your Outpost. A site must meet the facility, networking, and power requirements for your Outpost.
- **Outpost capacity** – Compute and storage resources available on the Outpost. You can view and manage the capacity for your Outpost from the AWS Outposts console.
- **Outpost equipment** – Physical hardware that provides access to the AWS Outposts service. The hardware includes racks, servers, switches, and cabling owned and managed by AWS.
- **Outposts racks** – An Outpost form factor that is an industry-standard 42U rack. Outposts racks include rack-mountable servers, switches, a network patch panel, a power shelf and blank panels.
- **Network racks** – Designed for networking, a network rack has a traffic aggregation layer for all connected compute and storage racks allowing you to decouple compute scaling from networking. This enables cost-efficient scaling of your on-premises workloads based on specific workload needs. The network rack also comes with built-in resiliency to handle network device failures, making it easier for you to architect for high availability of your Outposts network. In

addition, you can define the local gateway (LGW) network configurations, including IP addresses, Virtual LAN (VLAN) and Border Gateway Protocol (BGP) settings, through the API and console.

- **Outposts servers** – An Outpost form factor that is an industry-standard 1U or 2U server, which can be installed in a standard EIA-310D 19 compliant 4 post rack. Outposts servers provide local compute and networking services to sites that have limited space or smaller capacity requirements.
- **Outpost owner** – The account owner for the account that places the AWS Outposts order. After AWS engages with the customer, the owner may include additional points of contact. AWS will communicate with the contacts to clarify orders, installation appointments, and hardware maintenance and replacement. Contact AWS Support Center if the contact information changes.
- **Service link** – Network route that enables communication between your Outpost and its associated AWS Region. Each Outpost is an extension of an Availability Zone and its associated Region.
- **Local gateway (LGW)** – A logical interconnect virtual router that enables communication between an Outposts rack and your on-premises network.

Supported AWS services by AWS Region

AWS Outposts supports AWS services based on the AWS Region in which your Outpost operates. To determine the supported services, view your Region in the respective geographic area:

Areas

- North America
- Asia Pacific
- Europe

North America

The following table indicates AWS Outposts support for AWS services in North America Regions.

AWS Region	Application Load Balancer	Amazon EBS	Amazon EC2	Amazon ECS	Amazon EKS	Amazon EMR	AWS IoT Greengrass	Amazon RDS SQL, MySQL and PostgreSQL
US East (N. Virginia)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
US East (Ohio)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
US West (N. California)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
US West (Oregon)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Canada (Central)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Asia Pacific

The following table indicates AWS Outposts support for AWS services in Asia Pacific Regions.

AWS Region	Application Load Balancer	Amazon EBS	Amazon EC2	Amazon ECS	Amazon EKS	Amazon EMR	AWS IoT Greengrass	Amazon RDS SQL, MySQL and PostgreSQL
Asia Pacific (Singapore)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Europe

The following table indicates AWS Outposts support for AWS services in Europe Regions.

AWS Region	Application Load Balancer	Amazon EBS	Amazon EC2	Amazon ECS	Amazon EKS	Amazon EMR	AWS IoT Greengrass	Amazon RDS SQL, MySQL and PostgreSQL
Europe (London)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Europe (Paris)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Pricing

Pricing is based on your order details. When you place an order, you can choose from a variety of Outpost configurations, each providing a combination of Amazon EC2 instance types and storage options. You also choose a contract term and a payment option. Pricing includes delivery, installation, infrastructure service maintenance, software patches and upgrades, and rack removal.

For pricing based on location, configuration, and payment option, see: Outposts racks pricing

You are billed for shared resources and any data transfer from the AWS Region to the Outpost. You are also billed for data transfers that AWS performs to maintain availability and security.

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. It must also provide LAN or WAN access to the local network where your on-premises workloads or applications reside.

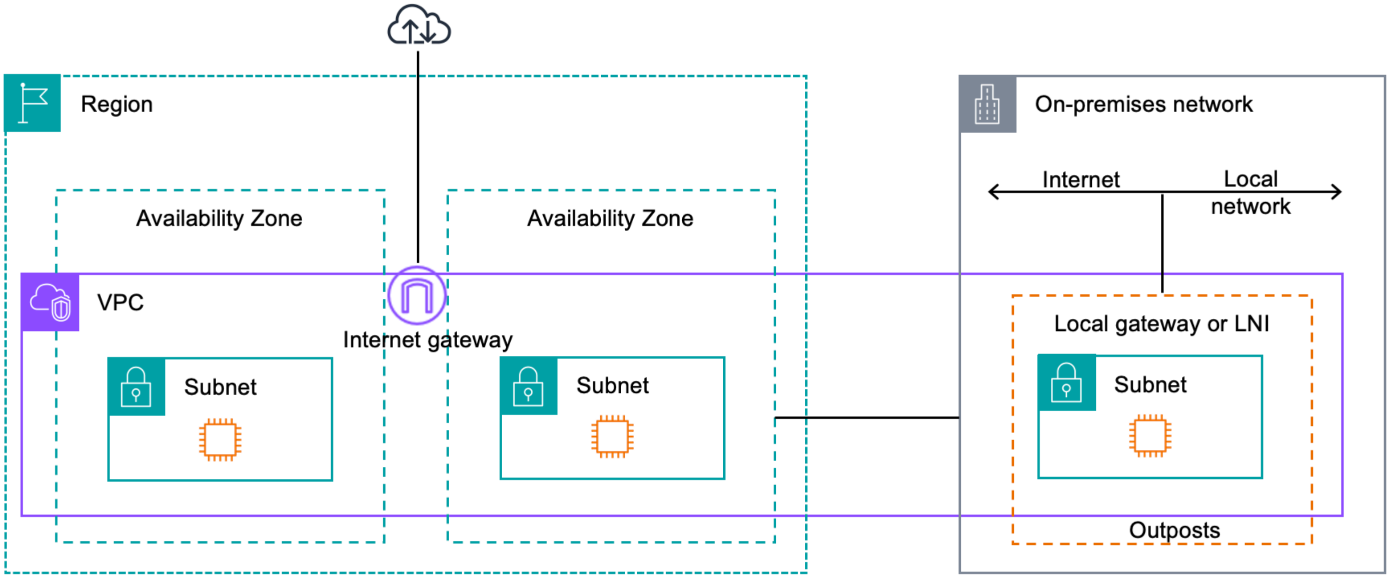
Contents

Network components

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

The following diagram shows the network components for your Outpost.

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



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 can't specify the underlying hardware where the instance is deployed, because the Outpost is a pool of AWS compute and storage capacity.

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

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

Routing

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

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

Outpost subnet route tables can include the following destinations:

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

  If you have a peering connection with multiple VPCs on the same Outpost, the traffic between the VPCs remains in the Outpost and does not use the service link back to the Region.
- **Intra-VPC communication across Outposts with local gateway** – You can establish communication between subnets in the same VPC across different Outposts with local gateways using direct VPC routing. For more information, see:
  - Direct VPC routing
  - Routing to an AWS Outposts local gateway

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:

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

Local gateways

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

- Local gateways for your Outposts racks
- Application/workload routing in the *AWS Outposts High Availability Design and Architecture Considerations* AWS Whitepaper

Bare-metal networking interfaces on Bmn instances

Instances from the Bmn family include one or more low latency and high throughput bare-metal local networking interfaces. For specialized mission-critical use cases, you can connect to your on-premises network through these high performance interfaces.

Site requirements for second-generation Outposts racks

This page covers the requirements for second-generation Outposts network and compute racks.

Contents

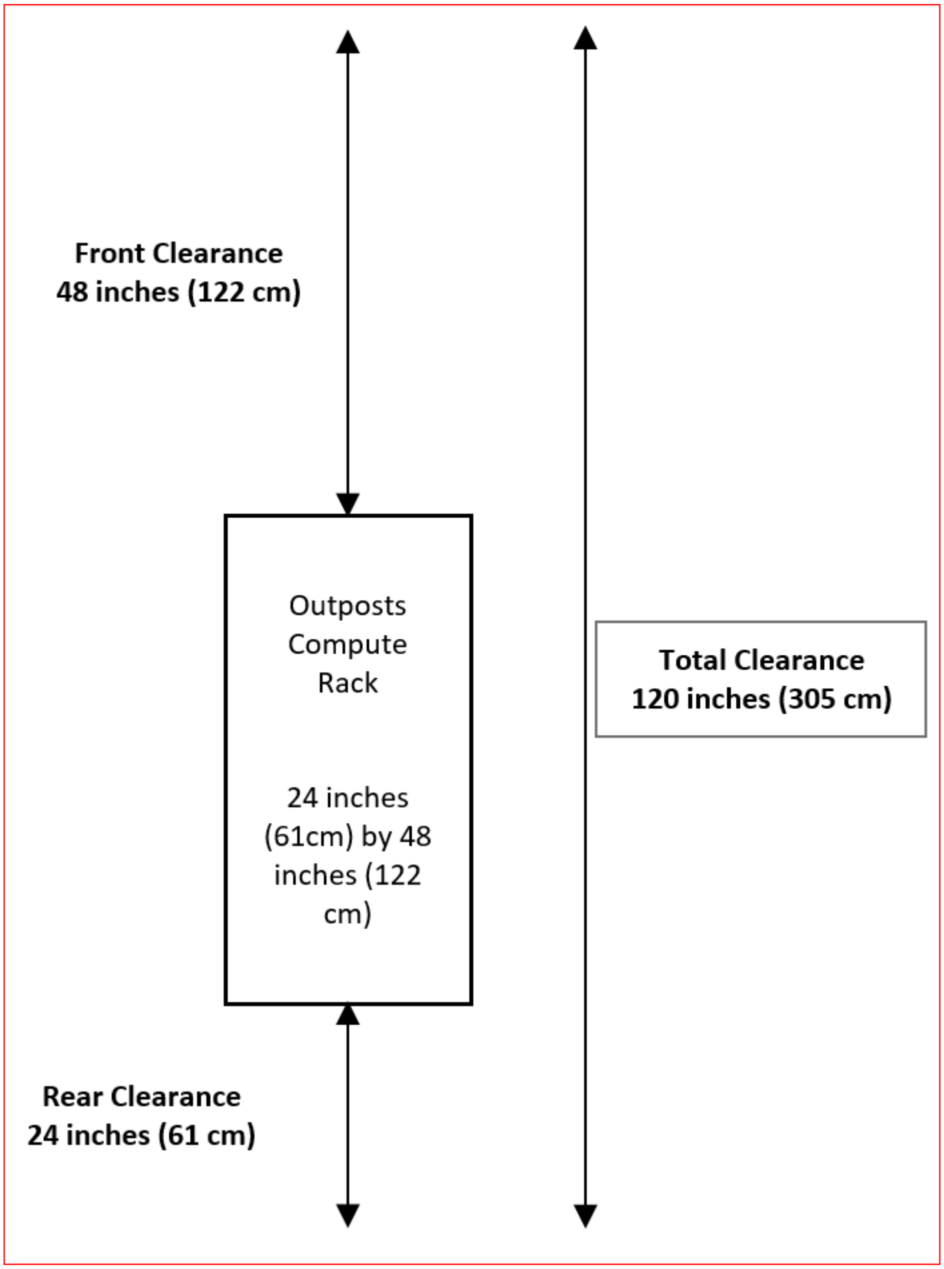
Compute rack requirements

Facility

These are the facility requirements for compute 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 Outposts compute rack, including clearance.



- **Seismic bracing –** To the extent required by regulation or code, you will install and maintain appropriate seismic anchorage and bracing for the rack while it is in your facility. AWS provides floor brackets that provide protection for up to 2.0G of seismic activity with all Outposts racks.
- **Bonding point –** We recommend that you provide a bonding wire/point at the rack position so that your electrician can bond the racks during installation which will be validated by the AWS-certified technician.
- **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).

Network rack requirements

An Outposts network rack acts as a network aggregation point for one or more Outpost compute racks and is a required component of every logical Outposts deployment.

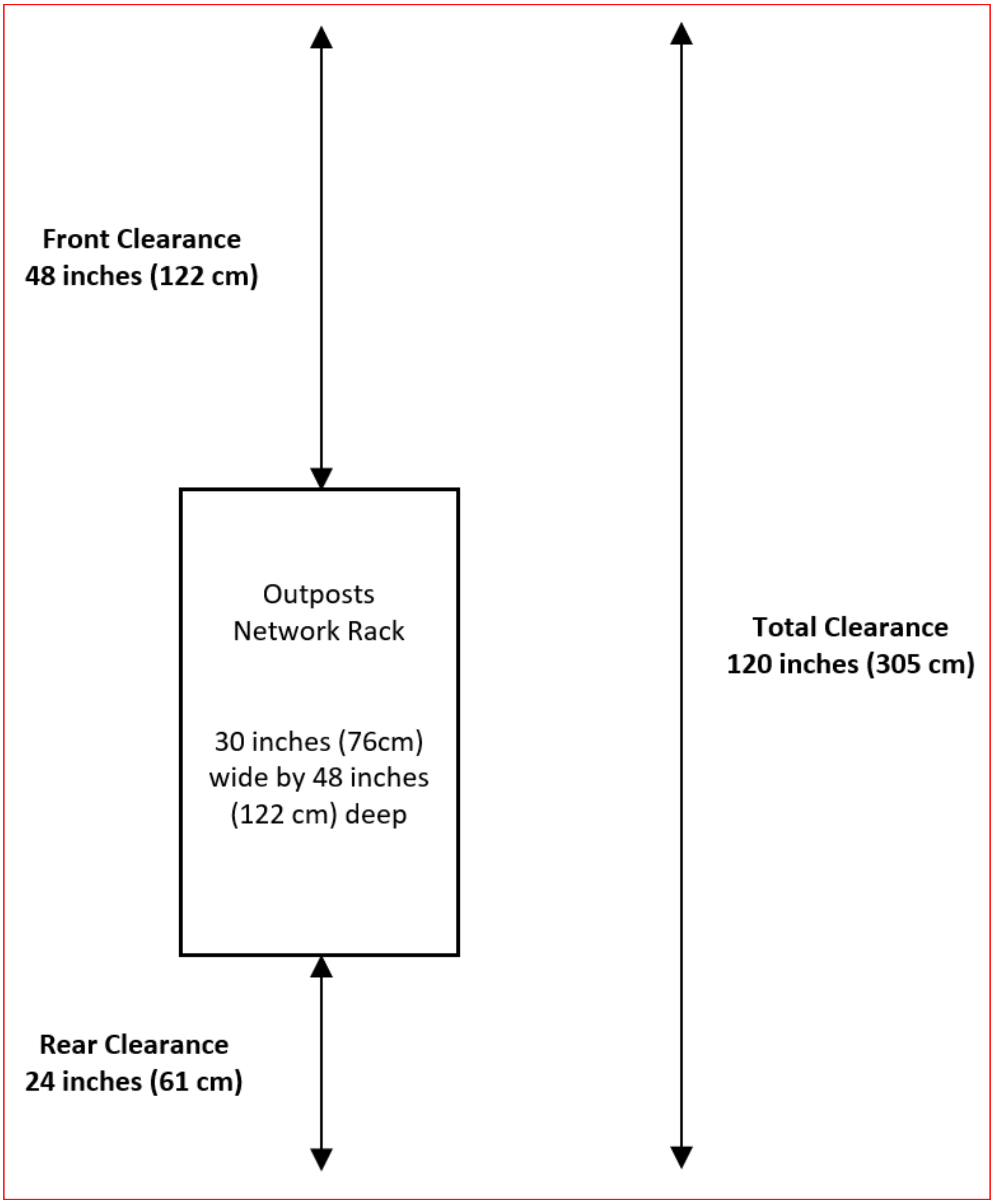
Contents

Facility

These are the facility requirements for network 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** – The Outposts network racks weighs 1975 lbs (896 kg).
- **Clearance** – The Outposts network racks is 80 inches (203 cm) high, 30 inches (76 cm) wide, and 48 inches (122 cm) deep.

  The following diagram shows the total minimum area required for the Outposts network rack, including clearance.



Network connectivity requirements

A network rack acts as a network aggregation point for one or more Outposts compute/storage racks, and includes four physical networking devices which must be connected to two or four upstream customer devices with a minimum of four physical links (one for each networking device).

- **Rack network requirements** – Ensure that you meet the requirements listed in the Network readiness checklist and Local network connectivity for Outposts racks sections.

- **Uplink speed** – Provide uplinks with speeds of 10 Gbps, 40 Gbps, or 100 Gbps. For bandwidth recommendations for the service link connection, see Service link bandwidth recommendations.
- **Fiber** – Provide single-mode fiber (SMF) with Lucent Connector (LC), or multi-mode fiber (MMF) with Lucent Connector (LC). For the full list of supported fiber types and optical standards, see **Uplink speed, ports, and fiber** in Network readiness checklist.
- **Upstream device** – Provide two or four upstream devices, which can be switches or routers. To understand how the network devices in an Outposts Network rack connects to your on-premises networking devices see **Physical connectivity** in Network readiness checklist.
- **Service link VLAN and a local Gateway VLAN** – For each of the four Outposts Networking devices you must provide a Service VLAN and a different Local Gateway VLAN. You can choose to provide only two distinct VLANs, one for the Service VLAN and one for the Local gateway VLAN, or have different VLANs in each Outposts networking device for both Service VLAN and LGW VLAN for a total of 8 different VLANs. For more information on how link aggregation groups (LAGs) and VLAN are used, see Link aggregation and Virtual LANs.
- **CIDR and IP address requirements** – Outposts service link requires /24 block for service link infrastructure subnets. For the interfaces on the Outposts networking devices that connect to customers on-premises networking devices, Outposts requires a 4x /31 CIDR for Service Link VLAN and 4x /31 CIDR for Local Gateway VLAN. For the interface connectivity, you must specify the IP addresses for the four Outposts networking devices to use. For more information, see Network layer connectivity.
- **Customer and Outpost BGP Autonomous System Number (ASN) for service link VLAN and a Local Gateway VLAN** – The Outpost establishes an external BGP (eBGP) peering session between each networking device in the Outposts network rack and your local network device for service link connectivity over the service link VLAN. In addition, it establishes an eBGP peering session from each networking device in the Outposts network rack to a local network device for connectivity to the local gateway. For more information, see Service link BGP connectivity and Local gateway BGP connectivity.

⚠ **Important**

Service Link infrastructure subnets – A service link infrastructure subnet (must be /24) is required for each Outposts installation.

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.

The following requirements are for Outposts network racks:

- Provide uplinks with speeds of 10 Gbps, 40 Gbps, or 100 Gbps.
- Provide either single-mode fiber (SMF) with Lucent Connector (LC), multimode fiber (MMF), or MMF OM4 with LC.
- Provide two or four upstream devices, which can be switches or routers.

Physical Connectivity

An Outposts network racks has four 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.

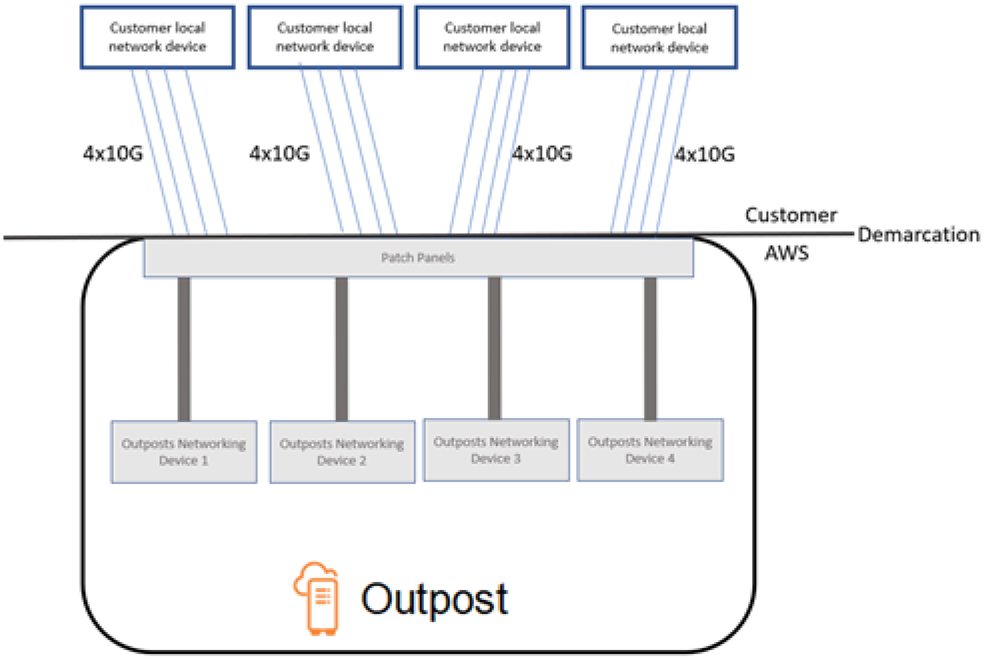
The following table shows how many uplink ports are supported for each Outpost network device, based on the uplink speed.

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

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).

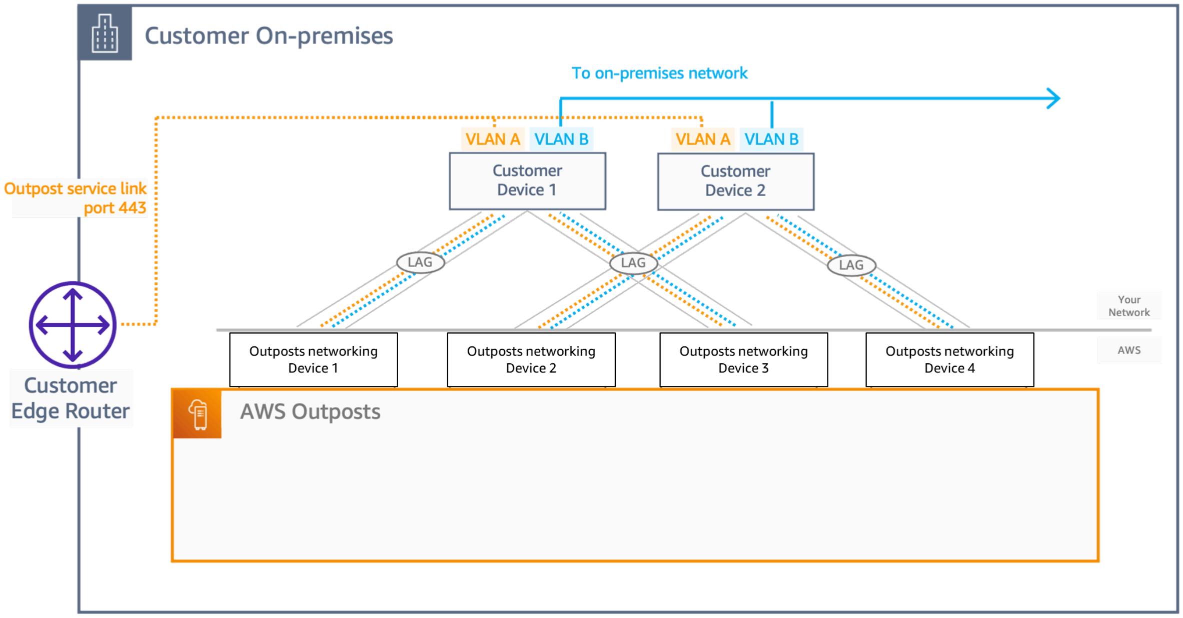
Provide either a single-mode fiber (SMF) with Lucent Connector (LC), multimode fiber (MMF), or MMF OM4 with LC. AWS provides the optics that are compatible with the fiber that you provide at the rack position.

In the following image, 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.

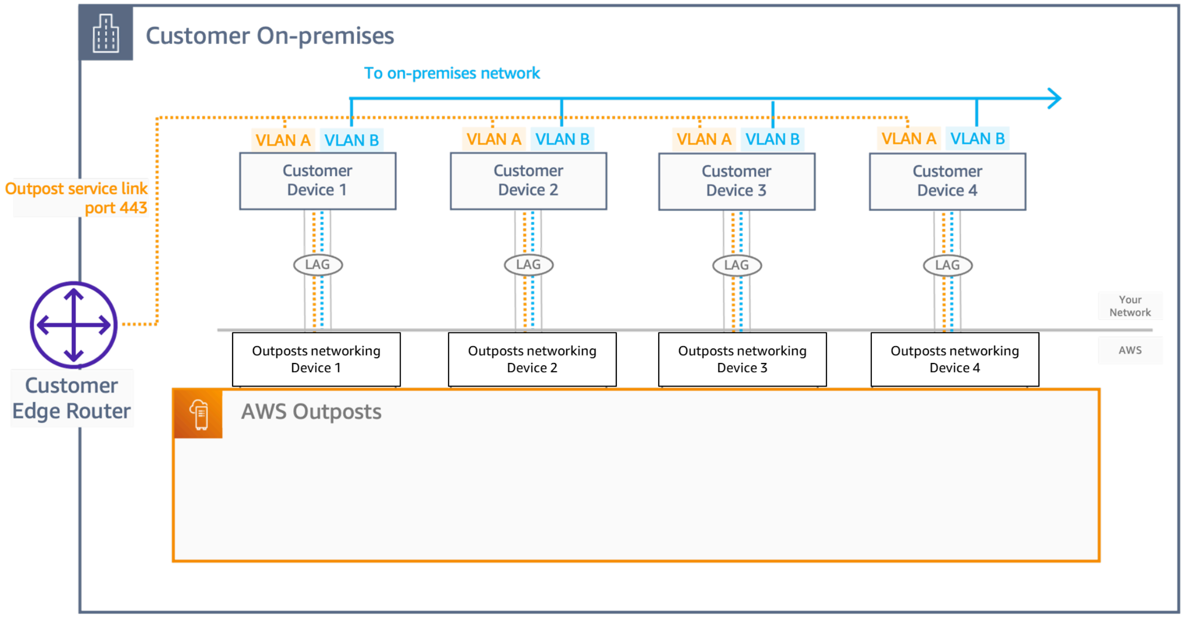

Customer local network device
Customer local network device
Customer local network device
Customer local network device
4x10G
4x10G
4x10G
4x10G
Customer
Demarcation
AWS
Patch Panels
Outposts Networking Device 1
Outposts Networking Device 2
Outposts Networking Device 3
Outposts Networking Device 4
Outpost

The following images show the networking connection topologies supported on Outposts network racks.

The following image shows the four Outposts networking devices in the Outposts network rack connected to two upstream customer devices:



The following image shows the four Outposts networking devices in the Outposts network rack connected to four upstream customer devices:



Uplink speed, ports, and fiber

Uplink speed and ports

An Outpost has four 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.

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

10 Gbps

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

40 Gbps or 100 Gbps

1, 2, or 4 uplinks

Fiber

AWS Outposts requires fibers with Lucent Connectors (LC).

The following table lists the supported optical standards and the corresponding fiber type required. If the optical standard uses the Multi-fiber Push-On (MPO) connector, you will need a MPO to 4 x LC Type-B breakout cable where the 4 x LC connectors attach to the Outpost to establish one link.

Uplink speed	Optical standard	Fiber type
10 Gbps	– 10GBASE-IR – 10GBASE-LR	SMF (LC)
10 Gbps	– 10GBASE-SR	MMF (LC)
40 Gbps	– 40GBASE-IR4 (LR4L) – 40GBASE-LR4	SMF (LC)
40 Gbps	– 40GBASE-ESR4 – 40GBASE-SR4	MMF (MPO to 4 x LC Type-B breakout)
100 Gbps	– 100GBASE-CWDM4 – 100GBASE-LR4	SMF (LC)
100 Gbps	– 100G PSM4 MSA	SMF (MPO to 4 x LC Type-B breakout)
100 Gbps	– 100GBASE-SR4	MMF (MPO to 4 x LC Type-B breakout)

Outpost link aggregation and VLANs

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

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

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

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

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

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

Outpost network device IP connectivity

Each of the four 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. Specify a subnet and an IP address from the subnet for the Outpost to use. For more information, see Network layer connectivity.

Outpost network device	Service link interface requirements	Local gateway interface requirements
#1	– Service link CIDR (/31) – Service link IP address	– Local gateway CIDR (/31) – Local gateway IP address
#2	– Service link CIDR (/31) – Service link IP address	– Local gateway CIDR (/31) – Local gateway IP address
#3	– Service link CIDR (/31) – Service link IP address	– Local gateway CIDR (/31) – Local gateway IP address
#4	– Service link CIDR (/31) – Service link IP address	– Local gateway CIDR (/31) – Local gateway IP address

Service link maximum transmission unit (MTU)

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

Service link Border Gateway Protocol

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

Outpost	Service link BGP requirements
Your Outpost	– Outpost BGP Autonomous System Number (ASN). 2-byte (16-bit) or 4-byte (32-bit). From your private ASN range (64512-65534 or 4200000000-4294967294). – Must be the same ASN for all devices – Infrastructure CIDR (/31 required)

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

Local network device	Service link BGP requirements
#4	– 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 /24	443	Outpost Region's public routes
TCP	1025-65535	Outpost service link /24	443	Outpost Region's public routes

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

Local gateway Border Gateway Protocol

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

Outpost	Local gateway BGP requirements
Your Outpost	– Outpost BGP Autonomous System Number (ASN). 2-byte (16-bit) or 4-byte (32-bit). From your private ASN range (64512-65534 or 4200000000-4294967294).

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

Power requirements for compute and network racks

The Outposts power shelf supports 10 kVA, 15 kVA, and 30 kVA power positions. Contact your AWS representative if you need a custom configuration for your unique application requirements.

The following table lists the power requirements for Outposts compute and network racks:

Requirement	Specification
AC line voltage	**Single-phase** 208 to 277 VAC; 50 or 60 Hz **Three-phase**: • 208 to 250 VAC (Delta); 50 to 60 Hz • 346 to 480 VAC (Wye); 50 to 60 Hz
Power consumption	15 kVA (13 kW) or 30 kVA (26 kW)

Requirement	Specification
AC protection (upstream power breakers)	30 A, 32 A, or 50 A
AC inlet type (receptacle)	Single phase L6-30P (30A) or IEC309 P+N+E, 6 hour (32 A), three phase AH530P7W 3P+N+E, 7 hour (30A), or three phase AH532P6W 3P+N+E 6 hour (32 A), or three phase Non-NEMA twistlock Hubbell CS8365C, 3P+E, center ground (50A)
Whip length	10.25 ft (3 m)
Whip - Rack cabling input	From above or below the rack

Outposts compute racks support redundant single and three phase power drops. The following table shows the supported power and connectors.

	Redundant, single-ph ase	Redundant, three-phase	Single-phase	Three-phase
15 kVA	6 x L6-30P or IEC309; 3 drops to S1 and 3 drops to S2	2 x AH530P7W or AH532P6W, 1 drop to S1 and 1 drop to S2	3 x L6-30P or IEC309; 3 drops to S1	1 x AH530P7W or AH532P6W, 1 drop to S1
30 kVA	12 x L6-30P or IEC309; 3 drops to S1 and 3 drops to S2 on each of the two power shelves	4 x AH530P7W or AH532P6W or CS8365C, 1 drop to S1 and 1 drop to S2 on each of the two power shelves	6 x L6-30P or IEC309; 3 drops to S1 on each of the two power shelves	2 x AH530P7W or AH532P6W or CS8365C, 1 drop to S1 on each of the two power shelves

Outposts network racks draws 8.89 kVA and supports redundant single and three phase power drops. The following table shows the supported power and connectors.

	Redundant, single-phase	Redundant, three-phase	Single-phase	Three-phase
10 kVA	4 x L6-30P or IEC309; 2 drops to S1 and 2 drops to S2	2 x AH530P7W, AH532P6W, or CS8365C, 1 drop to S1 and 1 drop to S2	2 x L6-30P or IEC309; 2 drops to S1	1x AH530P7W, AH532P6W, or CS8365C, 1 drop to S1

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

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

Order fulfillment

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

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

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

The team will establish network connectivity for the 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 rack is available from your AWS account.

Get started with Outposts racks

Order an Outposts rack to get started. After installation of your Outpost equipment, launch an Amazon EC2 instance and configure connectivity to your on-premises network.

Tasks

- Create an order for an Outposts rack
- Launch an instance on your Outposts rack
- Optimize Amazon EC2 for AWS Outposts

Create an order for an Outposts rack

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

Prerequisites

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

Tasks

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

Step 1: Create a site

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

Prerequisites

- Determine the operating address.

To create a site

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

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

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

Step 2: Create an Outpost

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

Prerequisites

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

To create an Outpost

1. In the navigation pane, choose **Outposts**.
2. Choose **Create Outpost**.
3. Choose **Racks**.
4. Enter a name and description for your Outpost.
5. Choose an Availability Zone for your Outpost.
6. (Optional) To configure private connectivity, select **Use Private connectivity**. Choose a VPC and subnet in the same AWS account and Availability Zone as your Outpost. For more information, see Service link private connectivity options.

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

7. For **Site ID**, choose your site.
8. Choose **Create Outpost**.

ⓘ **Note**

You won't be able to modify the AZ anchor or physical location of your Outpost after you complete the order.

Step 3: Place the order

Place an order for the Outposts racks that you need.

⚠ **Important**

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

Prerequisites

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

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

To place an order

1. From the navigation pane, choose **Orders**.
2. Choose **Place order**.
3. For **Supported hardware type**, choose **Racks**.
4. For **Configurations**, specify the quantity for each resource that you need. If the available configurations do not meet your capacity needs, contact AWS Support Center to request a custom capacity configuration.
5. For **Storage**:

- Choose an Amazon EBS storage tier.
- (Optional) Choose an Amazon S3 storage tier.

6. Choose **Next**.
7. Choose **Use an existing Outpost** and select your Outpost.
8. Choose **Next**.
9. Enter the name and number of the contact person at the operating site.
10. Specify the shipping address. You can specify a new address or select the site's operating address. If you select the operating address, be aware that any future change to the site's operating address will not propagate to existing orders. If you need to change the name and address of the shipping location on an existing order, contact your AWS Account Manager.
11. For **Site details**, specify your site information for each field.
12. Review the **Facility requirements**.
13. Select **I have read the facility requirements**.
14. Choose **Next**.
15. Select a contract term and payment option.
16. Choose **Next**.
17. On the **Review and order** page, verify that your information is correct and edit as needed. You will not be able to edit the order after you submit it.
18. Choose **Place order**.

Step 4: Modify instance capacity

An Outpost provides a pool of AWS compute and storage capacity at your site as a private extension of an Availability Zone in an AWS Region. Because the compute and storage capacity available in the Outpost is finite and determined by the size and number of racks that AWS installs at your site, you get to decide how much Amazon EC2 and Amazon EBS on AWS Outposts capacity you need to run your initial workloads, accommodate future growth, and to provide extra capacity to mitigate server failures and maintenance events.

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

Note

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

To modify instance capacity

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

Console steps

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

   a. Choose the instance size.

   b. Choose **Auto-balance** at the end of the related instance-type section.
7. For each instance type, ensure that the instance quantity is specified for at least one instance size.
8. Choose **Next**.
9. On the **Review and create** page, verify the updates that you are requesting.

10. Choose **Create**. AWS Outposts creates a capacity task.
11. On the capacity task page, monitor the status of the task.

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

Upload a JSON file

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

   Example

   Example JSON file:

   ```
   {
       "InstancePools": [
           {
               "InstanceType": "c5.24xlarge",
               "Count": 1
           },
           {
               "InstanceType": "m5.24xlarge",
               "Count": 2
           }
       ]
   }
   ```
4. Review the contents of the JSON file in the **Capacity configuration plan** section.
5. Choose **Next**.

6. On the **Review and create** page, verify the updates that you are requesting.
7. Choose **Create**. AWS Outposts creates a capacity task.
8. On the capacity task page, monitor the status of the task.

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

Next steps

You can view the status of your order using the AWS Outposts console. The initial status of your order is **Order received**. If you have any questions about your order, contact AWS Support Center.

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 Outposts 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.

Prerequisite

You must have an Outpost installed at your site.

Tasks

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

Step 1: Create a VPC

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

To create a VPC for your Outpost

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. Choose the same Region as the Outposts rack.
3. On the navigation pane, choose **Your VPCs** and then choose **Create VPC**.
4. Choose **VPC only**.
5. (Optional) for **Name tag** enter a name for the VPC.
6. For **IPv4 CIDR block**, choose **IPv4 CIDR manual input** and enter the IPv4 address range for the VPC in the **IPv4 CIDR** text box.

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

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

Step 2: Create a subnet and custom route table

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

2a: Create an Outpost subnet

To create an Outpost subnet

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions**, **Create subnet**. You are redirected to create a subnet in the Amazon VPC console. We select the Outpost for you and the Availability Zone that the Outpost is homed to.
4. Select a VPC.
5. In **Subnet settings**, optionally name your subnet and specify an IP address range for the subnet.
6. Choose **Create subnet**.
7. (Optional) To make it easier to identify Outpost subnets, enable the **Outpost ID** column on the **Subnets** page. To enable the column, choose the **Preferences** icon, select **Outpost ID**, and choose **Confirm**.

2b: Create a custom route table

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

To create a custom route table

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

7. Choose **Create route table**.

2c: Associate the Outpost subnet and custom route table

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

To associate the Outpost subnet and custom route table

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.
2. From the navigation pane, choose **Route tables**.
3. On the **Subnet associations** tab, choose **Edit subnet associations**.
4. Select the check box for the subnet to associate with the route table.
5. Choose **Save associations**.

Step 3: Configure local gateway connectivity

The local gateway (LGW) enables connectivity between your Outpost subnets and your on-premises network.

For more information about the LGW, see Local gateways.

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

3a. Create a custom local gateway route table

Use the following procedure to create a custom route table for your local gateway.

To create a custom 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 table**.
4. Choose **Create local gateway route table**.

5. (Optional) For **Name**, enter a name for your route table.
6. For **Local gateway**, choose your local gateway.
7. For **Mode**, choose a mode for communication with your on-premises network.
   - Choose **Direct VPC routing** to use the private IP addresses of your instances.
   - Choose **CoIP** to use addresses from your customer-owned IP address pools. For more information, see Create a CoIP pool.
8. (Optional) To add a tag, choose **Add new tag** and enter a tag key and a tag value.
9. Choose **Create local gateway route table**.

3b: Associate the VPC with the custom route table

Use the following procedure to associate a VPC with your local gateway route table. They are not associated by default.

To associate a VPC with the custom 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 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) To add a tag, choose **Add new tag** and enter a tag key and a tag value.
7. Choose **Associate VPC**.

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

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

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

Destination	Target
172.16.100.0/24	`lgw-id`

To add a route entry with the local gateway as a target in the subnet route table

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

3d: Create a local gateway routing domain by associating the custom route table with the VIF groups

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

To associate the custom route table with the VIF groups

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. To change the AWS Region, use the Region selector in the upper-right corner of the page.
3. On the navigation pane, choose **Networking** and then **LGW routing domain**.
4. Choose **Create LGW routing domain**.
5. Enter a name for the local gateway routing domain.
6. Choose the local gateway, the local gateway VIF group, and the local gateway route table.

> **Note**
>
> If you do not have a VIF group, create one. For information on how to create a VIF group, see [VIF and VIF groups](#).

7. Choose **Create LGW routing domain**.

3e: Add a route entry in the route table

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

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

To add a route entry in the LGW route table

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

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

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

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

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 the section called “Sharing an Outpost resource”.

Step 4: Configure the on-premises network

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

For more information, see Local gateway BGP connectivity.

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

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

Example: BGP advertisements in Direct VPC mode

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

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

In this scenario, the route advertised by:

- The local gateway to your customer devices is 10.0.3.0/24.

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

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

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

For troubleshooting, see AWS Outposts rack network troubleshooting checklist.

Example: BGP advertisements in CoIP mode

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

- A VPC with a CIDR block 10.0.0.0/16.
- A subnet in the VPC with a CIDR block 10.0.3.0/24.
- A customer-owned IP pool (10.1.0.0/26).
- An Elastic IP address association that associates 10.0.3.112 to 10.1.0.2.
- A subnet in the on-premises network with a CIDR block 172.16.100.0/24
- Communication between your Outpost and on-premises network will use the CoIP Elastic IPs to address instances in the Outpost, the VPC CIDR range is not used.

In this scenario the route advertised by:

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

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

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

Step 5: Launch an instance on the Outpost

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

Considerations

- To use block data or boot volumes backed by compatible third-party storage, you must provision and configure these volumes for use with EC2 instances on Outposts. For more information, see *Third-party block storage*.
- You can create a placement group to influence how Amazon EC2 should attempt to place groups of interdependent instances on the Outposts hardware. You can choose the placement group strategy that meets the needs of your workload.
- If you add Amazon EBS volumes, you must use the gp2 and gp3 volume types.
- If your Outpost has been configured to use a customer-owned IP (CoIP) address pool, you must assign a customer-owned IP address to any instances that you launch.

To launch instances in your Outpost subnet

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions, View details**.
4. On the **Outpost summary** page, choose **Launch instance**. You are redirected to the instance launch wizard in the Amazon EC2 console. We select the Outpost subnet for you, and show you only the instance types that are supported by your Outposts rack.
5. Choose an instance type that is supported by your Outposts rack. Note that instances that appear greyed out are not available.
6. (Optional) To launch the instances into a placement group, expand **Advanced details** and scroll to **Placement group**. You can either select an existing placement group or create a new placement group.
7. (Optional) You can add a third-party data volume.

    a. Expand **Configure storage**. Next to **External storage volume**, choose **Edit**.

   b. For **Storage Network Protocol**, choose **iSCSI**.
   c. Enter the Initiator IQN, then add the target IP address, the port, and the IQN of the external storage array.

8. Complete the wizard to launch the instance in your Outpost subnet. For more information, see Launch an EC2 instance in the *Amazon EC2 User Guide*:

Step 6: Test the connectivity

You can test connectivity by using the appropriate use cases.

Test connectivity from your local network to the Outpost

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

```
ping 10.0.3.128
```

The following is example output.

```
Pinging 10.0.3.128

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

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

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

Test the connectivity from an Outpost instance to your local network

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

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

```
ping 172.16.0.130
```

The following is example output.

```
Pinging 172.16.0.130

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

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

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

Test connectivity between the AWS Region and the Outpost

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

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

After the instance is running, perform the following operations:

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

   ```
   ping 10.0.1.5
   ```

   The following is example output.

```
Pinging 10.0.1.5

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

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

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

Customer-owned IP address connectivity examples

Test the connectivity from your local network to the Outpost

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

```
ping 172.16.0.128
```

The following is example output.

```
Pinging 172.16.0.128

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

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

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

Test the connectivity from an Outpost instance to your local network

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

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

```
ping 172.16.0.130
```

The following is example output.

```
Pinging 172.16.0.130

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

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

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

Test connectivity between the AWS Region and the Outpost

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

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

After the instance is running, perform the following operations:

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

   ```
   ping 10.0.0.5
   ```

The following is example output.

```
Pinging 10.0.0.5

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

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

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

Optimize Amazon EC2 for AWS Outposts

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

Contents

Dedicated Hosts on Outposts

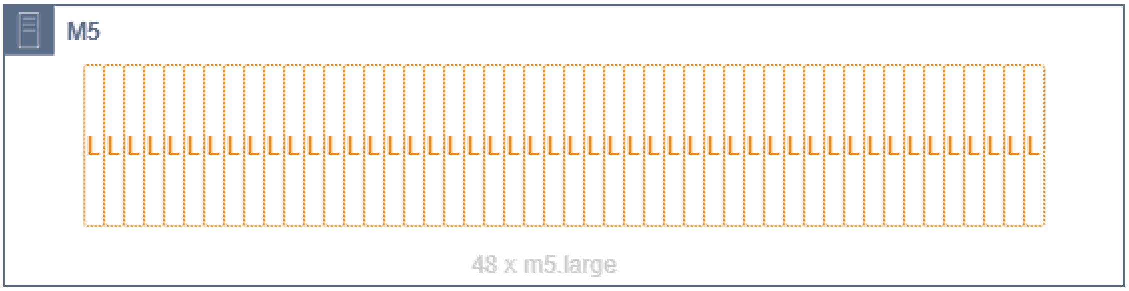
An Amazon EC2 Dedicated Host is a physical server with EC2 instance capacity fully dedicated to your use. Your Outpost already provides you with dedicated hardware, but Dedicated Hosts allows you to use existing software licenses with per-socket, per-core, or per-VM license restrictions against a single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

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

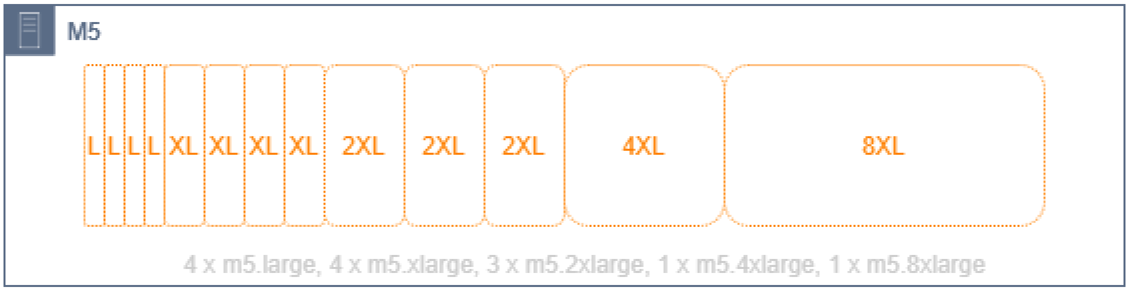
- Alter the capacity layout of a server
- Control instance placement at the hardware level

Alter the capacity layout of a server

Dedicated Hosts offers you the capability to alter the layout of servers in your Outpost deployment without contacting 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:


M5
L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L L
48 x m5.large

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:


M5
L
L
L
L
XL
XL
XL
XL
2XL
2XL
2XL
4XL
8XL
4 x m5.large, 4 x m5.xlarge, 3 x m5.2xlarge, 1 x m5.4xlarge, 1 x m5.8xlarge

For more information, see Allocate a Dedicated Host in the *Amazon EC2 User Guide*.

Control instance placement at the hardware level

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

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

Set up instance recovery

Instances on your Outpost that go into an unhealthy state because of hardware failure must be migrated to a healthy host. You can set up auto-recovery to have this migration done automatically based on instance status checks. For more information, see Instance resiliency.

Placement groups on Outposts

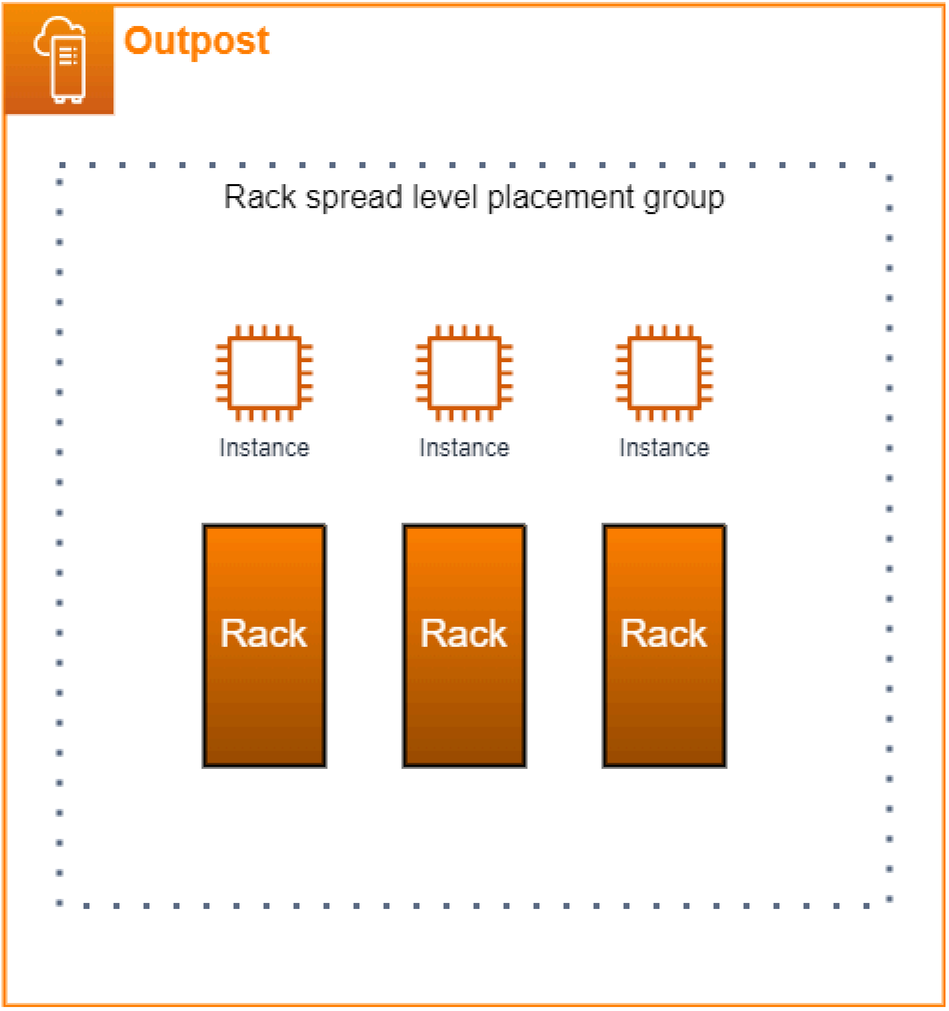
AWS Outposts supports placement groups. Use placement groups to influence how Amazon EC2 should attempt to place groups of interdependent instances that you launch on the underlying hardware. You can use different strategies (cluster, partition, or spread) to meet the needs of different workloads. If you have a single-rack Outpost, you can use the spread strategy to place instances across hosts instead of racks.

Spread placement groups

Use a spread placement group to distribute a single instance across distinct hardware. Launching instances in a spread placement group reduces the risk of simultaneous failures that might occur when instances share the same equipment. Placement groups can spread instances across racks or hosts. You can use host level spread placement groups only with AWS Outposts.

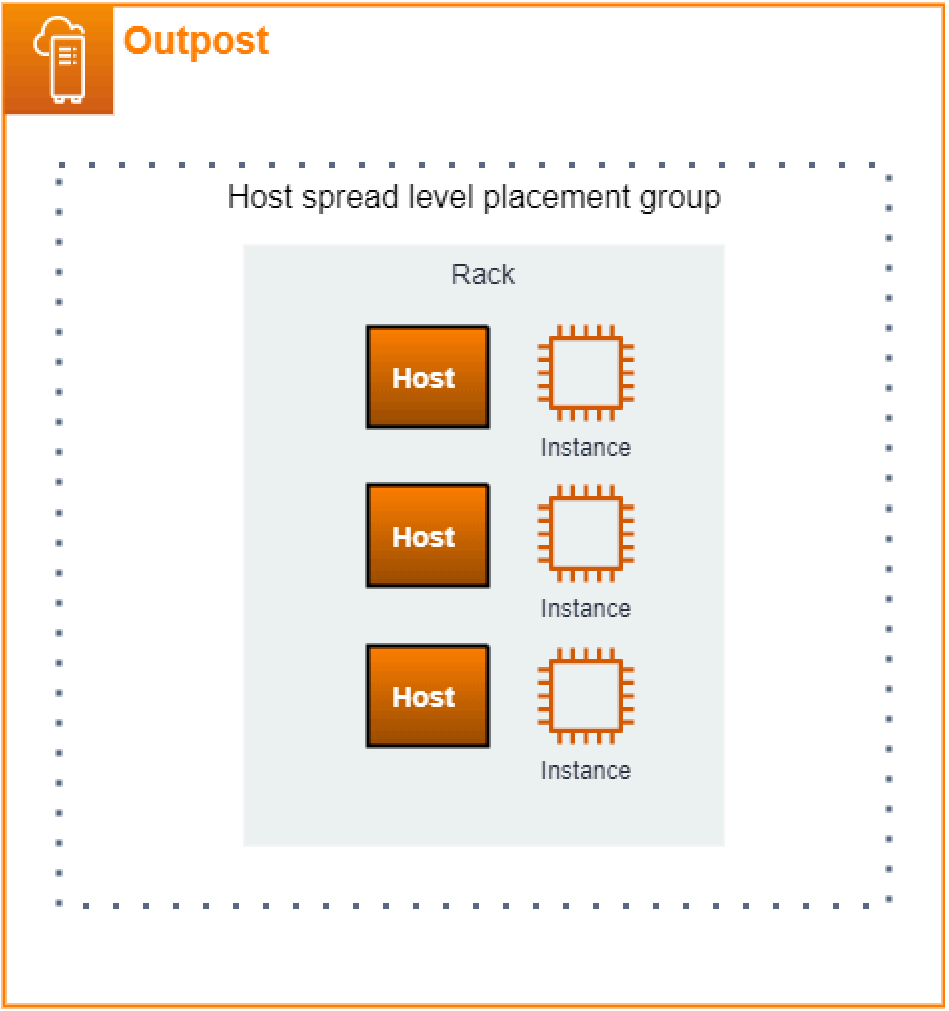
Rack spread level placement groups

Your rack spread level placement group can hold as many instances as you have racks in your Outpost deployment. The following illustration shows a three-rack Outpost deployment running three instances in a rack spread level placement group.



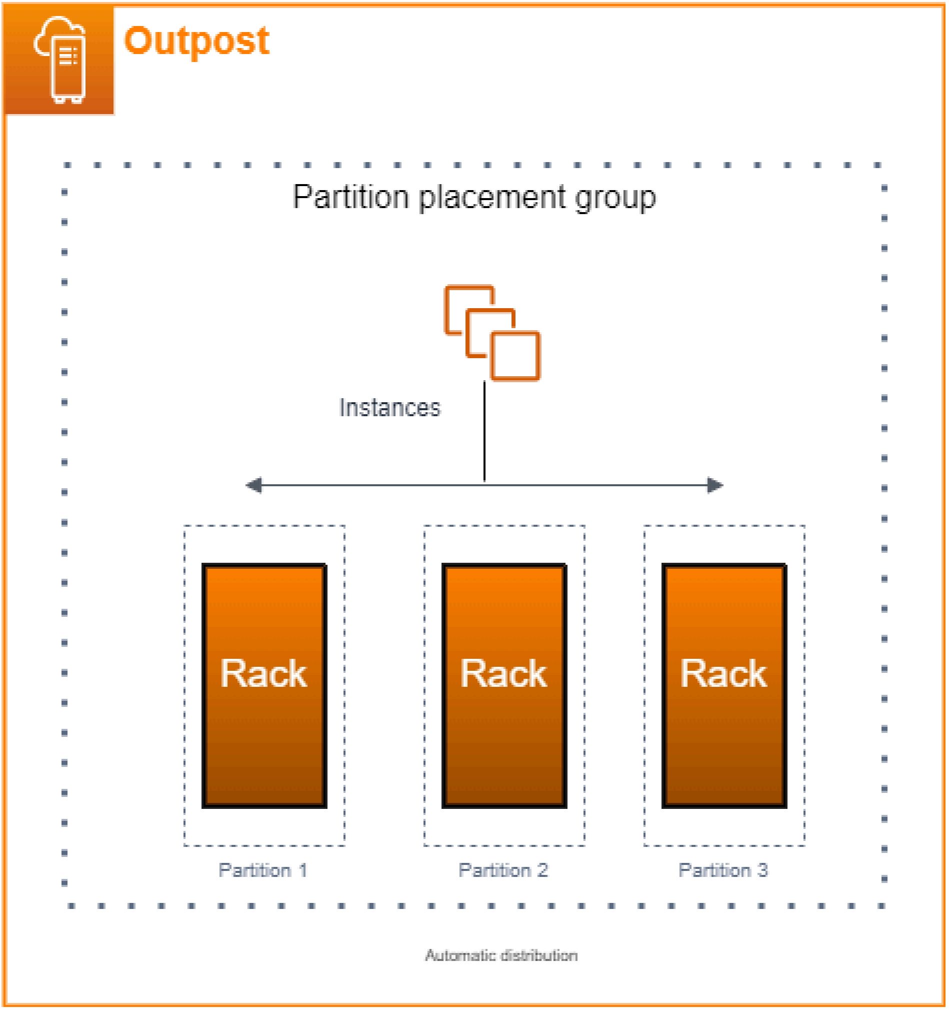
Host spread level placement groups

Your host spread level placement group can hold as many instances as you have hosts in your Outpost deployment. The following illustration shows a single-rack Outpost deployment running three instances in a host spread level placement group.

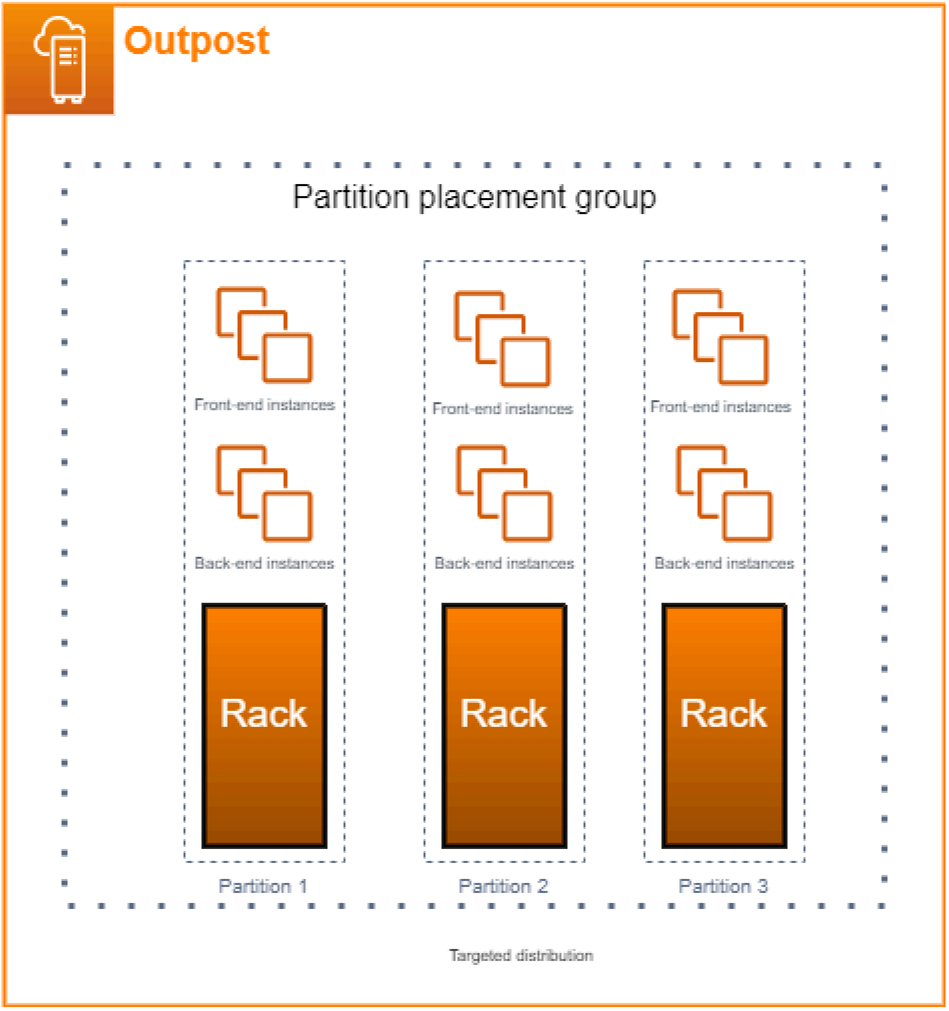


Partition placement groups

Use a partition placement group to distribute multiple instances across racks with partitions. Each partition can hold multiple instances. You can use automatic distribution to spread instances across partitions or deploy instances to target partitions. The following illustration shows a partition placement group with automatic distribution.



You can also deploy instances to target partitions. The following illustration shows a partition placement group with targeted distribution.



For more information about working with placement groups, see Placement groups and Placement groups on AWS Outposts in the *Amazon EC2 User Guide*.

For more information about AWS Outposts high availability, see AWS Outposts High Availability Design and Architecture Considerations.

Local network connectivity for Outposts network racks

You need the following components to connect your Outposts network racks to your on-premises network:

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

Contents

Link aggregation

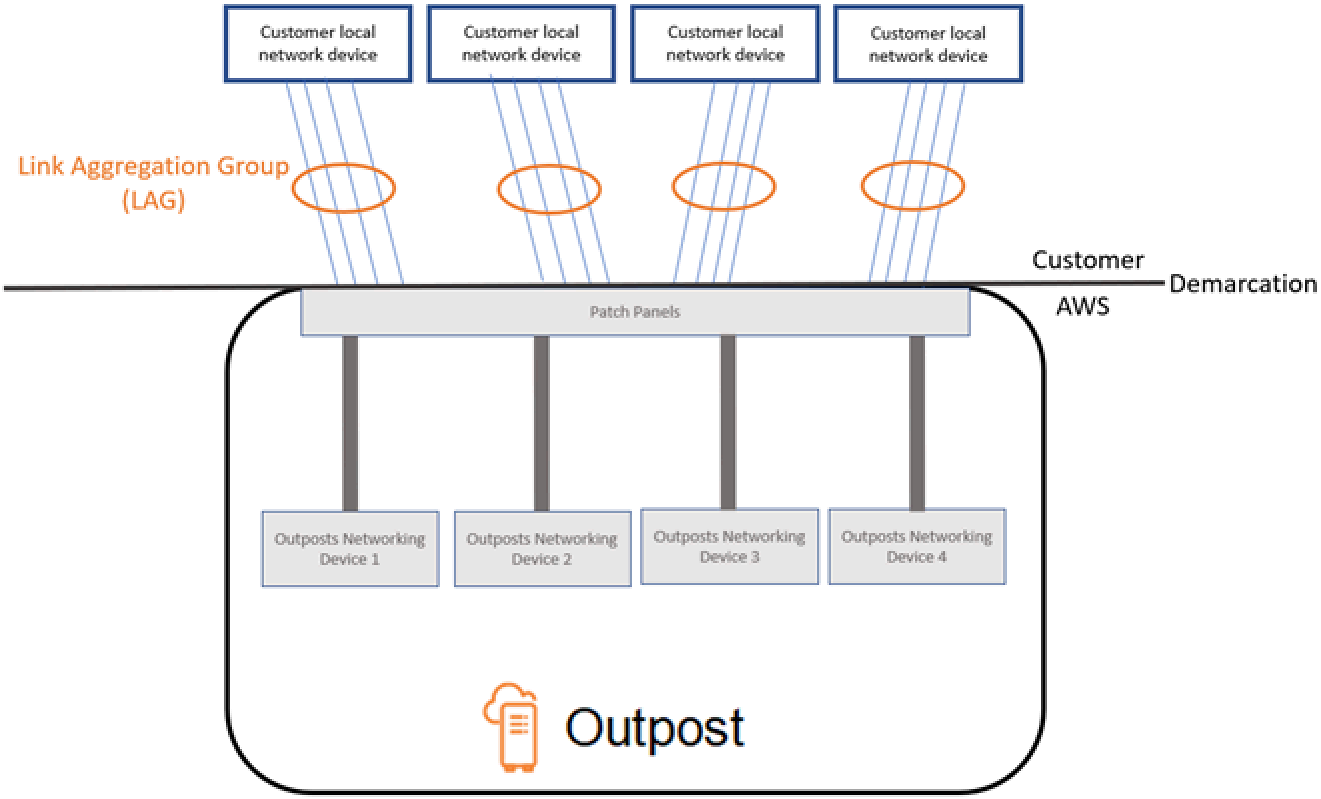
AWS Outposts uses the Link Aggregation Control Protocol (LACP) to establish four link aggregation group (LAG) connections, one from each Outpost network device within the Outposts Network rack to each customer upstream network device. The links from each Outpost network device are aggregated into an 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.

You can review your LAG details on the AWS Outposts console: Choose **Networking** and then **Link aggregation groups (LAGs)** from the left pane.

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.


Customer local network device
Customer local network device
Customer local network device
Customer local network device
Link Aggregation Group (LAG)
Customer
Demarcation
AWS
Patch Panels
Outposts Networking Device 1
Outposts Networking Device 2
Outposts Networking Device 3
Outposts Networking Device 4
Outpost

Virtual LANs

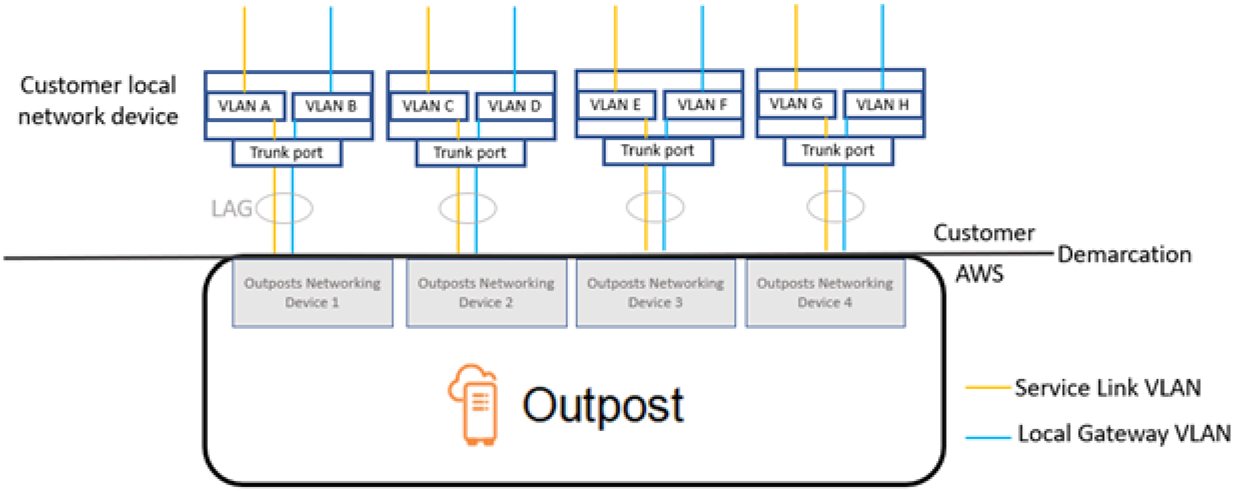
Each LAG between an Outpost network device and a local network device must be configured as an IEEE 802.1q Ethernet trunk. This enables the use of multiple VLANs for network segregation between data paths.

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

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

You can configure the service link VLAN and local gateway VLAN only between the Outpost and your customer local network devices. You can review your service link and LGW VLAN information on the AWS Outposts console: choose **Networking** and then **Link aggregation groups (LAGs)** from the navigation pane. Select the link aggregation group. Choose the **LGW virtual interfaces (VIFs)** and **Service link virtual interfaces (VIFs)** tabs to see the **VLAN** value.

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



To integrate your Outpost with your on-premises network during the installation and on-going operation, you must allocate the VLANs used between the Outpost network devices and the customer local network devices. You need to provide this information to AWS before the installation. For more information, see the section called “Network readiness checklist”.

Network layer connectivity

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

We recommend the following:

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

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

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

You can review your service link and LGW IP connectivity information on the AWS Outposts console: Choose **Networking** and then **Link aggregation groups (LAGs)** from the left pane. Select the link aggregation group. Choose the **LGW virtual interfaces (VIFs)** and **Service link virtual interfaces (VIFs)** tabs to see the IP values.

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 /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. As part of the installation process, AWS configures the attributes that you provided.

You can review your BGP information on the AWS Outposts console: Choose **Networking** and then **Link aggregation groups (LAGs)** from the navigation pane. Select the link aggregation group. Choose the **LGW virtual interfaces (VIFs)** and **Service link virtual interfaces (VIFs)** tabs to see the BGP values.

Service link infrastructure subnet advertisement and IP range

You provide a /24 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.

Local gateway BGP connectivity

The Outpost 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 /31 IPs that you provided when you set up network connectivity and uses point-to-point connectivity between each Outpost network device and customer local network device. For more information, see the section called “Network layer connectivity”.

Each BGP session uses the private ASN on the Outpost network device side, and an ASN that you choose on the customer local network device side.

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

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

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

Local gateway customer-owned IP subnet advertisement

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

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.

AWS Outposts connectivity to AWS Regions

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

Contents

Connectivity through service link

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

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

- Customer VPC traffic between the Outpost and any associated VPCs.
- Outposts management traffic, such as resource management, resource monitoring, and firmware and software updates.

Service link maximum transmission unit (MTU) requirements

The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection.

Note the following:

- The network **must** support 1500-bytes MTU between the Outpost and the service link endpoints in the parent AWS Region.
- The traffic that goes from an instance in Outposts to an instance in the Region has an MTU of 1300 bytes, which is lower than the required MTU of 1500 bytes due to packet overheads.

Service link bandwidth recommendations

For an optimal experience and resiliency, AWS requires that you use redundant connectivity of at least 500 Mbps for each compute rack and a maximum of 175 ms round trip latency 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 and maximum round trip time requirements for the 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 with optimal performance.

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

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

We strongly recommend consulting with your AWS sales representative or APN partner to evaluate available home Region options in your geography and seek a custom recommendation about the service link bandwidth and latency requirements for your workloads.

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 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.

Set up your service link

The following steps explain the service link setup process.

1. Choose a connection option between your Outposts and the home AWS Region. You can choose either a public or private connection.
2. After you order your Outposts racks, AWS contacts you to collect VLAN, IP, BGP, and infrastructure subnet IPs. For more information, see Local network connectivity.

3. During installation, AWS configures service link on the Outpost based on the information you provided.
4. You configure your local networking devices, such as routers, to connect to each Outpost network device through BGP connectivity. For information on service link VLAN, IP, and BGP connectivity, see Network connectivity requirements.
5. You configure your networking devices, such as firewalls, to enable your Outposts to access to the AWS Region or home Region. AWS Outposts utilizes the service link infrastructure subnet IPs to set up VPN connections and exchange control and data traffic with the Region. Service link establishment is always initiated from the Outpost.

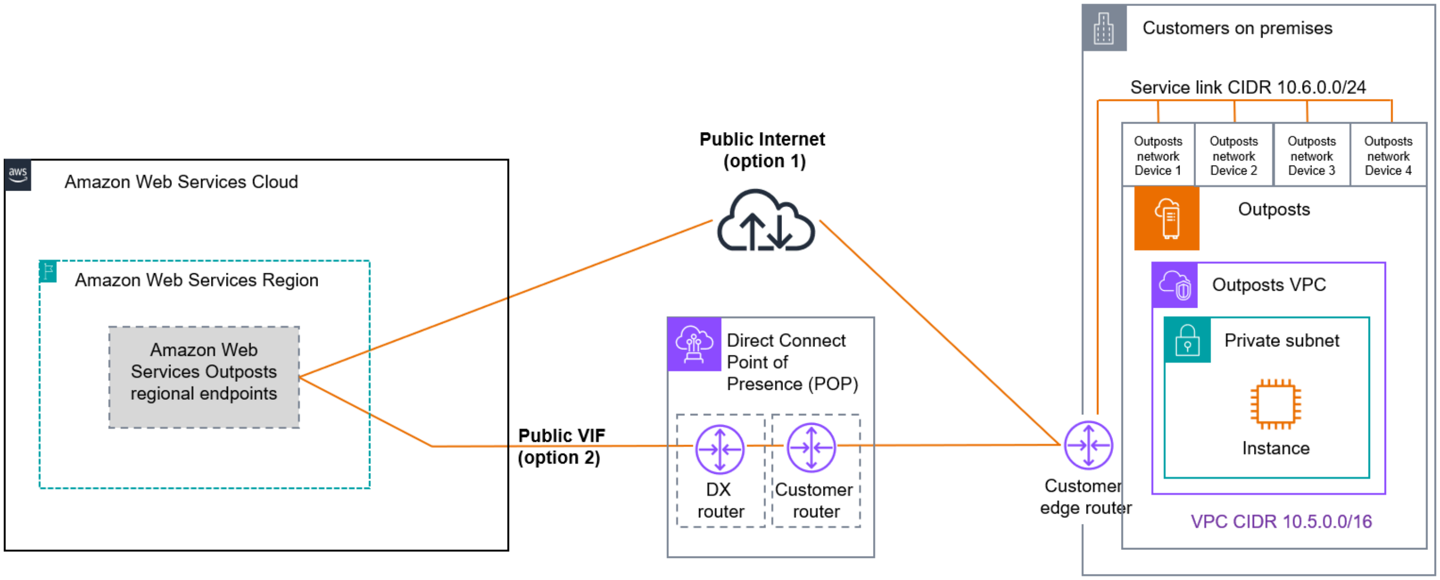
> **Note**
>
> You won't be able to modify the service link configuration or connectivity type after you complete the order.

Service link public connectivity options

You can configure the service link with a public connection for the traffic between the Outposts and home AWS Region. You can choose to use the public internet or Direct Connect public VIFs.

If you plan on allow-listing only AWS Region public IPs (instead of 0.0.0.0/0) on your firewalls, you must ensure that your firewall rules are up-to-date with the current IP address ranges. For more information, see AWS IP address ranges in the *Amazon VPC User Guide*.

The following image shows both options to establish a service link public connection between your Outposts and the AWS Region:


Customers on premises
Service link CIDR 10.6.0.0/24
Outposts network Device 1
Outposts network Device 2
Outposts network Device 3
Outposts network Device 4
Outposts
Outposts VPC
Private subnet
Instance
VPC CIDR 10.5.0.0/16
Customer edge router
Public Internet (option 1)
Amazon Web Services Cloud
Amazon Web Services Region
Amazon Web Services Outposts regional endpoints
Public VIF (option 2)
Direct Connect Point of Presence (POP)
DX router
Customer router

> **Note**
>
> Second-generation Outposts racks require a /24 or larger subnet for the service link infrastructure. This subnet is customer-provided IP address space used by Outpost networking devices to establish connectivity to AWS Region endpoints.

Option 1. Public connectivity through the internet

This option requires the AWS Outposts service link infrastructure subnet IPs to have access to the public IP ranges of your AWS Region or home Region. You must allow-list AWS Region public IPs or 0.0.0.0/0 on networking devices such as your firewall.

Option 2. Public connectivity through Direct Connect public VIFs

This option requires the AWS Outposts service link infrastructure subnet IPs to have access to the public IP ranges of your AWS Region or home Region over DX service. You must allow-list AWS Region public IPs or 0.0.0.0/0 on networking devices such as your firewall.

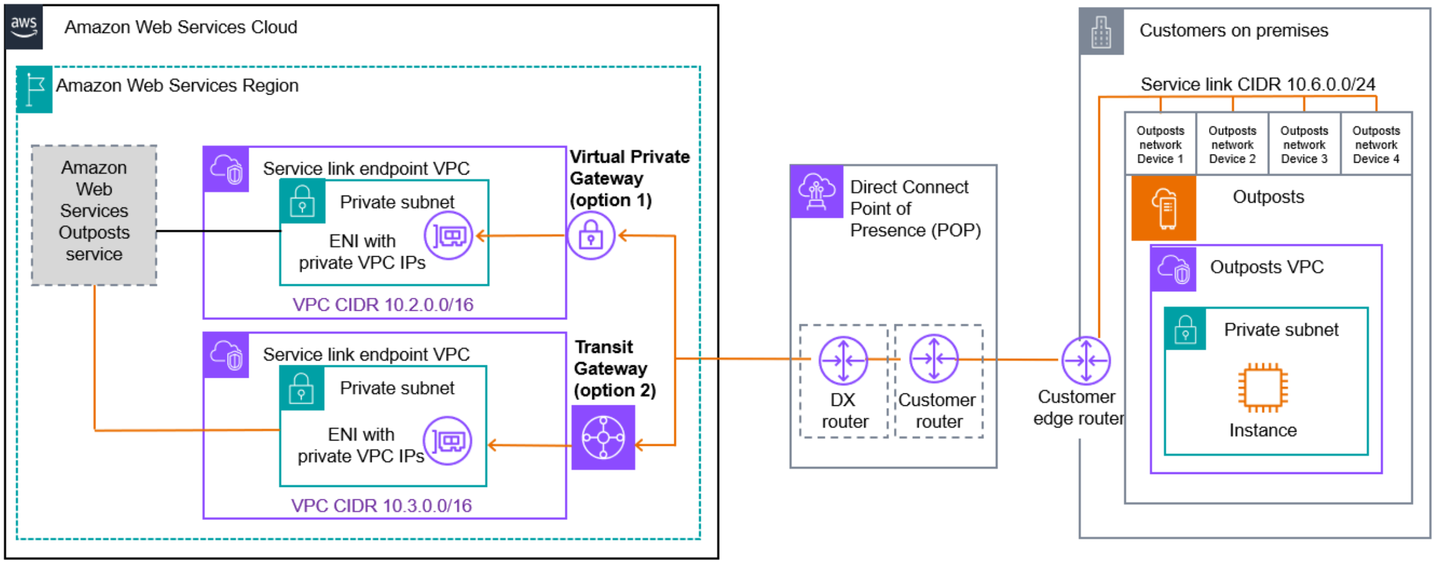
Service link private connectivity options

You can configure the service link with a private connection for the traffic between the Outposts and home AWS Region. You can choose to use Direct Connect private or transit VIFs.

Select the private connectivity option when you create your Outpost in the AWS Outposts console. For instructions, see Create an Outpost.

When you select the private connectivity option, a service link VPN connection is established after the Outpost is installed, using a VPC and subnet that you specify. This allows private connectivity through the VPC and minimizes public internet exposure.

The following image shows both options to establish a service link VPN private connection between your Outposts and the AWS Region:



Amazon Web Services Cloud
Amazon Web Services Region
Amazon Web Services Outposts service
Service link endpoint VPC
Private subnet
ENI with private VPC IPs
VPC CIDR 10.2.0.0/16
Virtual Private Gateway (option 1)
Service link endpoint VPC
Private subnet
ENI with private VPC IPs
VPC CIDR 10.3.0.0/16
Transit Gateway (option 2)
Direct Connect Point of Presence (POP)
DX router
Customer router
Customer edge router
Customers on premises
Service link CIDR 10.6.0.0/24
Outposts network Device 1
Outposts network Device 2
Outposts network Device 3
Outposts network Device 4
Outposts
Outposts VPC
Private subnet
Instance

Note

Second-generation Outposts racks require a larger subnet size (/24 or larger) and a VPC Endpoint for the Outposts service.

IP Address Planning for Private Connectivity

When configuring private connectivity for the Outposts service link, plan your IP addressing carefully to avoid future conflicts. Service Link VIFs are immutable. You cannot create CoIP pools or DVR subnet ranges assigned to the Local Gateway (LGW) that overlap with existing Service Link address ranges or VPC CIDR ranges used for the dedicated private connectivity VPC, as they will cause BGP routing conflicts and affect Service Link functionality.

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 AWS Identity and Access Management for AWS Outposts
- In the same AWS account and Availability Zone as your Outpost, create a VPC for the sole purpose of Outpost private connectivity with a subnet /24 or larger that does not conflict with 10.1.0.0/16. For example, you might use 10.3.0.0/16.

  > ⚠ **Important**
  >
  > Do not delete this VPC as it maintains the connection to your Outposts.
- Configure the subnet security group to allow traffic for UDP 443 inbound and outbound directions.
- Advertise the subnet CIDR to your on-premises network. You can use AWS Direct Connect to do so. For more information, see Direct Connect virtual interfaces and Working with Direct Connect gateways in the *Direct Connect User Guide*.
- Create a new VPC endpoint for AWS Outposts in your private connectivity VPC and subnet.

  Use the following VPC endpoint settings:
  - **Service**: Outposts (com.amazonaws.*region*.outposts)

    Example: `com.amazonaws.`*`us-west-2`*`.outposts`
  - **Endpoint type**: Interface

- **Private DNS Enabled**: set to false (disabled)
- **VPC**: the VPC you created for private connectivity
- **Subnet**: the subnet you created for private connectivity

Use the following **IAM policy document**:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "outposts:StartConnection",
                "outposts:GetConnection"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Principal": "*"
        }
    ]
}
```

- Create a security group for the endpoint and authorize **inbound** TCP port 443 and ICMP traffic with addresses of `0.0.0.0/0` and with no **outbound** rules.

> **Note**
>
> To select the private connectivity option when your Outpost is in **PENDING** status, choose **Outposts** from the AWS Outposts 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 Service-linked roles for AWS Outposts.

⚠ **Important**

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

ⓘ **Note**

VPC configuration cannot be changed after order placement. If incorrect VPC specifications are provided during ordering, the Outpost must be decommissioned and a new order placed.

Option 1. Private connectivity through Direct Connect private VIFs

Create an AWS Direct Connect connection, private virtual interface, and virtual private gateway to allow your on-premises Outpost to access the VPC.

For more information, see the following sections in the *Direct Connect User Guide*:

- Dedicated and hosted connections
- Create a private virtual interface
- Virtual private gateway associations

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 *Direct Connect User Guide*.

Option 2. Private connectivity through Direct Connect transit VIFs

Create an AWS Direct Connect connection, transit virtual interface, and transit gateway to allow your on-premises Outpost to access the VPC.

For more information, see the following sections in the *Direct Connect User Guide*:

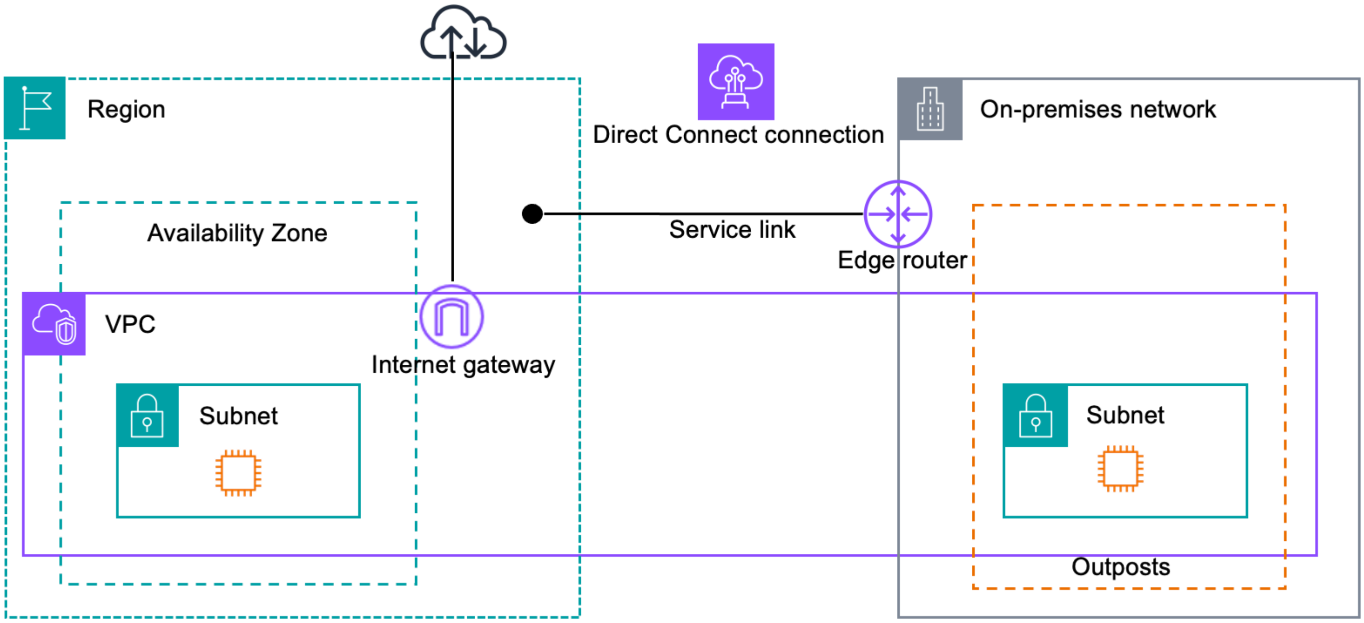
- Dedicated and hosted connections
- Create a transit virtual interface to the Direct Connect gateway
- Transit gateway associations

Firewalls and the service link

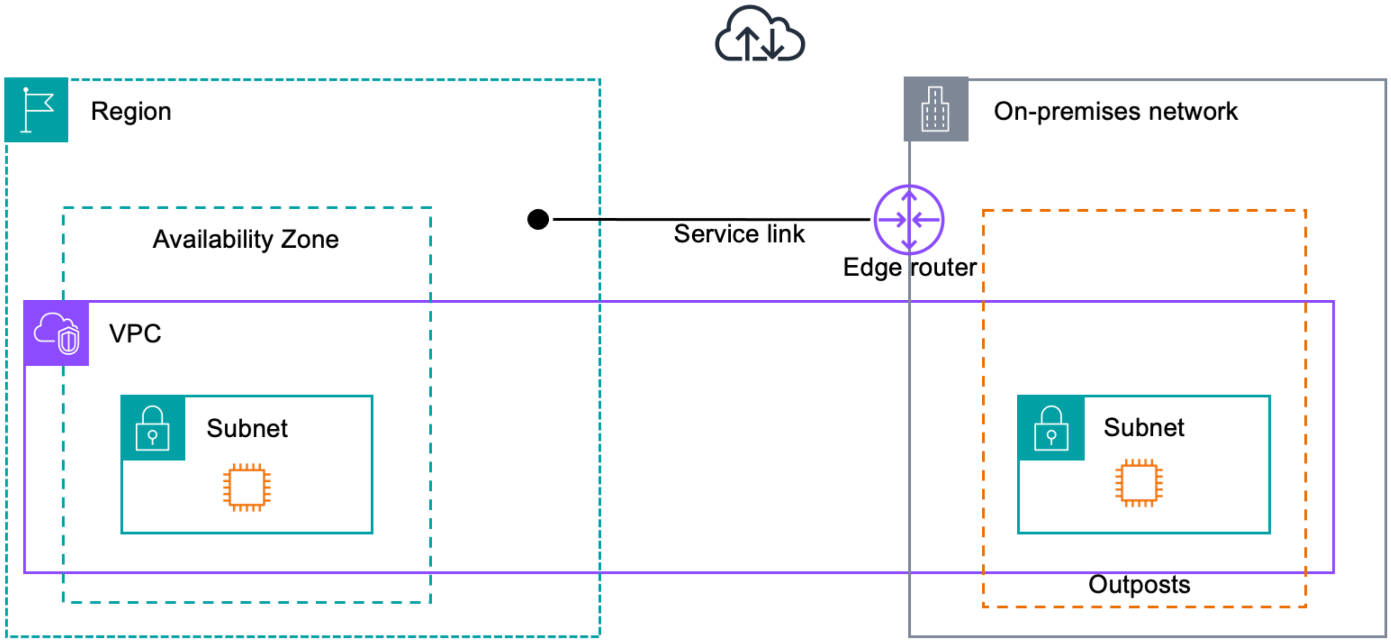
This section discusses firewall configurations and the service link connection.

In the following diagram, the configuration extends the Amazon VPC from the AWS Region to the Outpost. An Direct Connect public virtual interface is the service link connection. The following traffic goes over the service link and the 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 stateful firewall that is both UDP and TCP-aware to limit connectivity regarding the service link VLAN, you can deny all inbound connections. If the firewall is acting in a stateful manner, allowed outbound connections from the Outposts service link should automatically allow reply traffic back in without explicit rule configuration. Only outbound connections initiated from the Outpost service link need to be configured as allowed.

Protocol	Source Port	Source Address	Destination Port	Destination Address
UDP	443	AWS Outposts service link /24	443	AWS Outposts Region's public networks
TCP	1025-65535	AWS Outposts service link /24	443	AWS Outposts Region's public networks

If you use a non-stateful firewall to limit connectivity regarding the service link VLAN, you must allow outbound connections initiated from the Outposts service link to the AWS Outposts Region's public networks. You must also explicitly allow reply traffic in from the Outposts Region’s public networks inbound to the service link VLAN. Connectivity is always initiated outbound from the Outposts service link, but reply traffic must be allowed back into the service link VLAN.

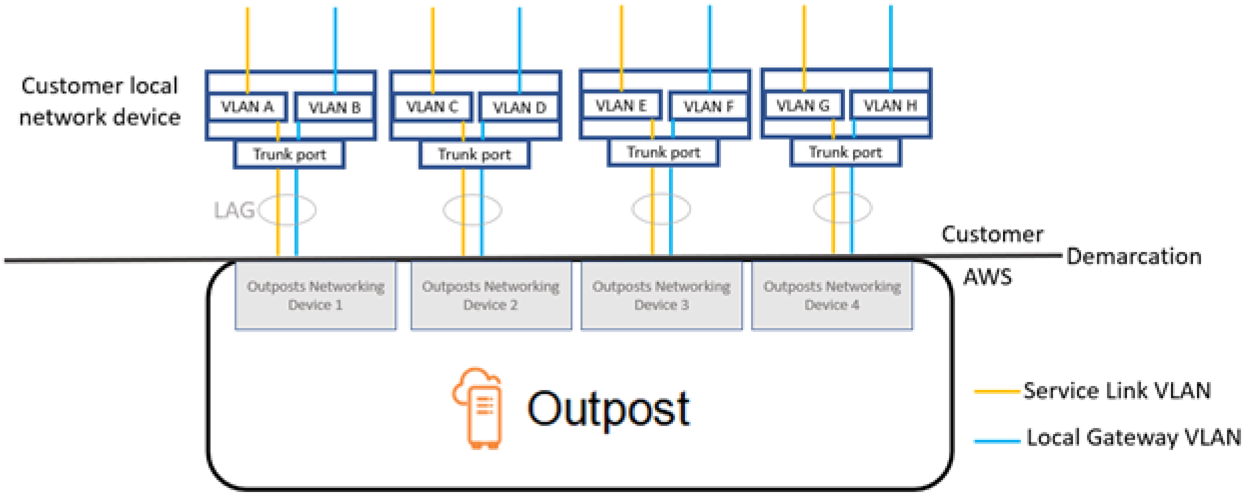
Protocol	Source Port	Source Address	Destination Port	Destination Address
UDP	443	AWS Outposts service link /24	443	AWS Outposts Region's public networks
TCP	1025-65535	AWS Outposts service link /24	443	AWS Outposts Region's public networks
UDP	443	AWS Outposts Region's public networks	443	AWS Outposts service link /24
TCP	443	AWS Outposts Region's public networks	1025-65535	AWS Outposts service link /24

Note

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

Outposts rack network troubleshooting checklist

Use this checklist to help troubleshoot a service link that has a status of `DOWN`.



Connectivity with Outpost network devices

Check the BGP peering status on the customer local network devices that are connected to the Outpost network devices. If the BGP peering status is `DOWN`, follow these steps:

1. Ping the remote peer IP address on the Outpost network devices from the customer devices. You can find the peer IP address in the BGP configuration of your device. You can also refer to the the section called “Network readiness checklist” provided to you at the time of installation.
2. If pinging is unsuccessful, check the physical connection and ensure that connectivity status is UP.
   a. Confirm the LACP status of the customer local network devices.
   b. Check the interface status on the device. If the status is UP, skip to step 3.
   c. Check the customer local network devices and confirm that the optical module is working.
   d. Replace faulty fibers and ensure the lights (Tx/Rx) are within acceptable range.
3. If pinging is successful, check the customer local network devices and ensure that the following BGP configurations are correct.
   a. Confirm that the local Autonomous System Number (Customer ASN) is correctly configured.
   b. Confirm that the remote Autonomous System Number (Outpost ASN) is correctly configured.
   c. Confirm that the interface IP and remote peer IP addresses are correctly configured.
   d. Confirm that the advertised and received routes are correct.
4. If your BGP session is flapping between active and connect states, verify that TCP port 179 and other relevant ephemeral ports are not blocked on the customer local network devices.
5. If you need to troubleshoot further, check the following on the customer local network devices:

   a. BGP and TCP debug logs
   b. BGP logs
   c. Packet capture

6. If the issue persists, perform MTR / traceroute / packet captures from your Outpost connected router to the Outpost network device peer IP addresses. Share the test results with AWS Support, using your Enterprise support plan.

If BGP peering status is `UP` between the customer local network devices and the Outpost network devices, but the service link is still `DOWN`, you can troubleshoot further by checking the following devices on your customer local network devices. Use one of the following checklists, depending on how your service link connectivity is provisioned.

- Edge routers connected with Direct Connect – Public virtual interface in use for service link connectivity. For more information, see Direct Connect public virtual interface connectivity to AWS Region.
- Edge routers connected with Direct Connect – Private virtual interface in use for service link connectivity. For more information, see Direct Connect private virtual interface connectivity to AWS Region.
- Edge routers connected with Internet Service Providers (ISPs) – Public internet in use for service link connectivity. For more information, see ISP public internet connectivity to AWS Region.

Direct Connect public virtual interface connectivity to AWS Region

Use the following checklist to troubleshoot edge routers connected with Direct Connect when a public virtual interface is in use for service link connectivity.

1. Confirm that the devices connecting directly with the Outpost network devices are receiving the service link IP address ranges through BGP.
   a. Confirm the routes that are being received through BGP from your device.
   b. Check the route table of the service link Virtual Routing and Forwarding instance (VRF). It should show that it is using the IP address range.
2. To ensure Region connectivity, check the route table for the service link VRF. It should include the AWS Public IP address ranges or the default route.
3. If you are not receiving the AWS public IP address ranges in the service link VRF, check the following items.

   a. Check the Direct Connect link status from the edge router or the AWS Management Console.

   b. If the physical link is UP, check the BGP peering status from the edge router.

   c. If the BGP peering status is `DOWN`, ping the peer AWS IP address and check the BGP configuration in the edge router. For more information, see Troubleshooting Direct Connect in the *Direct Connect User Guide* and My virtual interface BGP status is down in the AWS console. What should I do?.

   d. If BGP is established and you are not seeing the default route or AWS public IP address ranges in the VRF, contact AWS Support, using your Enterprise support plan.

4. If you have an on-premises firewall, check the following items.

   a. Confirm that the required ports for service link connectivity are allowed in the network firewalls. Use traceroute on port 443 or any other network troubleshooting tool to confirm the connectivity through the firewalls and your network devices. The following ports are required to be configured in the firewall policies for the service link connectivity.

      - **TCP protocol** – Source port: TCP 1025-65535, Destination port: 443.
      - **UDP protocol** – Source port: TCP 1025-65535, Destination port: 443.

   b. If the firewall is stateful, ensure that the outbound rules allow the Outpost's service link IP address range to the AWS public IP address ranges. For more information, see AWS Outposts connectivity to AWS Regions.

   c. If the firewall is not stateful, make sure to allow the inbound flow also (from the AWS public IP address ranges to the service link IP address range).

   d. If you have configured a virtual router in the firewalls, ensure that the appropriate routing is configured for traffic between the Outpost and the AWS Region.

5. If you have configured NAT in the on-premises network to translate the Outpost's service link IP address ranges to your own public IP addresses, check the following items.

   a. Confirm that the NAT device is not overloaded and has free ports to allocate for new sessions.

   b. Confirm that the NAT device is correctly configured to perform the address translation.

6. If the issue persists, perform MTR / traceroute / packet captures from your edge router to the Direct Connect peer IP addresses. Share the test results with AWS Support, using your Enterprise support plan.

Direct Connect private virtual interface connectivity to AWS Region

Use the following checklist to troubleshoot edge routers connected with Direct Connect when a private virtual interface is in use for service link connectivity.

1. If connectivity between the Outposts rack and the AWS Region is using the AWS Outposts private connectivity feature, check the following items.
    a. Ping the remote peering AWS IP address from the edge router and confirm the BGP peering status.
    b. Ensure that BGP peering over the 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 Direct Connect in the *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 Direct Connect private virtual interface is a private connection to your edge router in your chosen Direct Connect location, and it uses BGP to exchange routes. Your private virtual private cloud (VPC) CIDR range is advertised through this BGP session to your edge router. Similarly, the IP address range for the Outpost service link is advertised to the region through BGP from your edge router.
    d. Confirm that the network ACLs associated with the service link private endpoint in your VPC allow the relevant traffic. For more information, see Network readiness checklist.
    e. If you have an on-premises firewall, ensure that the firewall has outbound rules that allow the service link IP address ranges and the Outpost service endpoints (the network interface IP addresses) located in the VPC or the VPC CIDR. Ensure that the TCP 1025-65535 and UDP 443 ports are not blocked. For more information, see Introducing AWS Outposts private connectivity.
    f. If the firewall is not stateful, ensure that the firewall has rules and policies to allow inbound traffic to the Outpost from the Outpost service endpoints in the VPC.
2. If you have more than 100 networks in your on-premises network, you can advertise a default route over the BGP session to AWS on your private virtual interface. If you don't want to advertise a default route, summarize the routes so that the number of advertised routes is less than 100.
3. If the issue persists, perform MTR / traceroute / packet captures from your edge router to the Direct Connect peer IP addresses. Share the test results with AWS Support, using your Enterprise support plan.

ISP public internet connectivity to AWS Region

Use the following checklist to troubleshoot edge routers connected through an ISP when using the public internet for service link connectivity.

- Confirm that the internet link is up.
- Confirm that the public servers are accessible from your edge devices connected through an ISP.

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

1. Check whether BGP peering status with the ISP routers is established.
    a. Confirm that the BGP is not flapping.
    b. Confirm that the BGP is receiving and advertising the required routes from the ISP.
2. In case of static route configuration, check that the default route is properly configured on the edge device.
3. Confirm whether you can reach the internet using another ISP connection.
4. If the issue persists, perform MTR / traceroute / packet captures on your edge router. Share the results with your ISP's technical support team for further troubleshooting.

If the internet and public servers are accessible through the ISP links, complete the following steps.

1. Confirm whether any of your publicly accessible EC2 instances or load balancers in the Outpost home Region are accessible from your edge device. You can use ping or telnet to confirm the connectivity, and then use traceroute to confirm the network path.
2. If you use VRFs to separate traffic in your network, confirm that the service link VRF has routes or policies that direct traffic to and from the ISP (internet) and VRF. See the following checkpoints.
    a. Edge routers connecting with the ISP. Check the edge router's ISP VRF route table to confirm that the service link IP address range is present.
    b. Customer local network devices connecting with the Outpost. Check the configurations of the VRFs and ensure that the routing and policies required for connectivity between the service link VRF and the ISP VRF are configured properly. Usually, a default route is sent from the ISP VRF into the service link VRF for traffic to the internet.

   c. If you configured source-based routing in the routers connected to your Outpost, confirm that the configuration is correct.

3. Ensure that the on-premises firewalls are configured to allow outbound connectivity (TCP 1025-65535 and UDP 443 ports) from the Outpost service link IP address ranges to the public AWS IP address ranges. If the firewalls are not stateful, ensure that inbound connectivity to the Outpost is also configured.
4. Ensure that NAT is configured in the on-premises network to translate the Outpost’s service link IP address ranges to public IP addresses. In addition, confirm the following items.
   a. The NAT device is not overloaded and has free ports to allocate for new sessions.
   b. The NAT device is correctly configured to perform the address translation.

If the issue persists, perform MTR / traceroute / packet captures.

- If the results show that packets are dropping or blocked at the on-premises network, check with your network or technical team for additional guidance.
- If the results show that the packets are dropping or blocked at the ISP's network, contact the ISP’s technical support team.
- If the results do not show any issues, collect the results from all tests (such as MTR, telnet, traceroute, packet captures, and BGP logs) and contact AWS Support using your Enterprise support plan.

Outposts is behind two firewall devices

If you have placed your Outpost behind a high-availability pair of synced firewalls or two stand-alone firewalls, asymmetric routing of the service link might occur. This means that inbound traffic could pass through firewall-1, while outbound traffic goes through firewall-2. Use the following checklist to identify potential asymmetric routing of the service link especially if it was functioning correctly before.

- Verify if there were any recent changes or ongoing maintenance in your corporate network’s routing setup that might have led to asymmetric routing of the service link through the firewalls.
  - Use firewall traffic graphs to check for changes to traffic patterns that line up with the start of the service link issue.
  - Check for a partial firewall failure or a split-brained firewall-pair scenario that might have caused your firewalls to no longer sync their connection tables between each other.

  - Check for links down or recent changes to routing (OSPF/ISIS/EIGRP metric changes, BGP route-map changes) in your corporate network that line up with the start of the service link issue.
- If you are using public Internet connectivity for the service link to the home region, a service provider maintenance could have given rise to asymmetric routing of the service link through the firewalls.
  - Check traffic graphs for links to your ISP(s) for changes to traffic patterns that line up with the start of the service link issue.
- If you are using Direct Connect connectivity for the service link, it is possible that an AWS planned maintenance triggered asymmetric routing of the service link.
  - Check for notifications of planned maintenance on your Direct Connect service(s).
  - Note that if you have redundant Direct Connect services, you can proactively test the routing of the Outposts service link over each likely network path under maintenance conditions. This allows you to test if an interruption to one of your Direct Connect services could lead to asymmetric routing of the service link. The resiliency of the Direct Connect portion of the end-to-end network connectivity can be tested by the Direct Connect Resiliency with Resiliency Toolkit. For more information, see Testing Direct Connect Resiliency with Resiliency Toolkit – Failover Testing.

After you have gone through the preceding checklist and pinpointed asymmetric routing of the service link as a possible root cause, there are a number of further actions you can take:

- Restore symmetric routing by reverting any corporate network changes or waiting for a provider planned maintenance to complete.
- Log in to one or both firewalls and clear all flow state information for all flows from the command-line (if supported by the firewall vendor).
- Temporarily filter out BGP announcements through one of the firewalls or shut the interfaces on one firewall in order to force symmetric routing through the other firewall.
- Reboot each firewall in turn to eliminate potential corruption in the flow-state tracking of the service link traffic in the firewall’s memory.
- Engage your firewall vendor to either verify or relax the tracking of UDP flow-state for UDP connections sourced on port 443 and destined to port 443.

Local gateways for your Outposts racks

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

Contents

Local gateway basics

AWS creates a local gateway for each Outposts rack as part of the installation process. An Outposts rack supports a single local gateway. The local gateway is owned by the AWS account associated with the Outposts rack.

> **Note**
>
> To understand instance bandwidth limitations for traffic going through a local gateway, see Amazon EC2 instance network bandwidth in the *Amazon EC2 User Guide*.

> **IP Address Planning for Local Gateway**
>
> For Outposts using private connectivity, do not create CoIP pools or DVR subnet ranges that overlap with existing Service Link VIF address ranges or VPC CIDR ranges used for

the dedicated private connectivity VPC, as they will cause BGP routing conflicts and affect Service Link functionality.

A local gateway has the following components:

- **Route tables –** Only the owner of a local gateway can create local gateway route tables. For more information, see the section called "Route table routes".
- **CoIP pools –** (Optional) You can use IP address ranges that you own to facilitate communication between the on-premises network and instances in your VPC. For more information, see the section called "Customer-owned IP addresses".
- **Local gateway virtual interfaces (VIFs) and VIF groups –** Local gateway VIFs is a logical interface component of Outposts racks that sets up VLAN, IP, and BGP connectivity between an Outposts networking device and an on-premise networking device for local gateway connectivity. VIF groups are logical groupings of VIFs. You must create four local gateway VIFs within each VIF group for second-generation Outposts racks.
- **Local gateway route table and VPC associations –** Local gateway route table and VPC associations allow you to connect your VPCs to local gateway route tables. With this association, you can add a route targeted to the local gateway within your Outposts subnet route table. This enables communication between your Outposts subnet resources and your on-premises network through the local gateway.
- **Local gateway routing domains –** A local gateway routing domain is the association of a local gateway route table and local gateway VIF group. With this association, you can add a route targeted to a local gateway VIF group within your local gateway route table. This enables communication between your Outposts subnet resources and your on-premises network through the selected VIF group.

When AWS provisions your second-generation Outposts rack, we create some components and you are responsible for creating others.

AWS responsibilities

- Delivers the hardware.
- Creates the local gateway.

Your responsibilities

- Create the local gateway route table.
- Associate a VPC with the local gateway route table.
- Create the local gateway VIF and VIF groups.
- Associate a local gateway route table with a local gateway VIF group to create a local gateway routing domain.

Local gateway routing

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

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

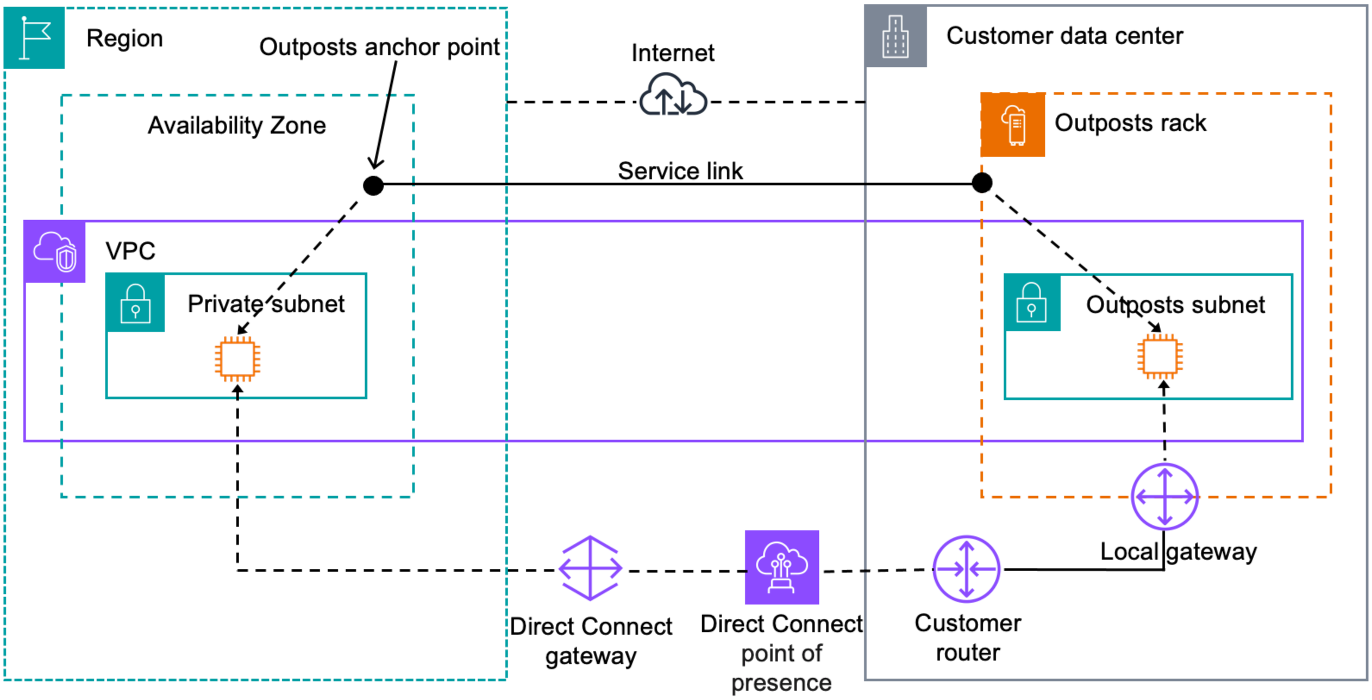
Connectivity through a local gateway

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

The data plane path for the local gateway traverses from the Outpost, through the local gateway, and to your private local gateway LAN segment. The local gateway can also provide a data plane path back to the AWS Region, for example for AWS service endpoints in the Region. Note that the control plane path always uses the service link connectivity, regardless of the data plane path that you use.

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

The following image shows the connectivity through the local gateway:



Local gateway route tables

The local gateway is owned by the AWS account associated with the Outpost. 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. Only the owner of the local gateway can modify the local gateway route table.

Outpost subnet route tables can include a route to local gateway VIF groups in order to provide connectivity to your on-premises network.

Local gateway route tables have a mode that determines how instances in the Outposts subnet communicate with your on-premises network. The default option is direct VPC routing, which uses the private IP addresses of the instances. The other option is to use addresses from a customer-owned IP address pool (CoIP) that you provide. Direct VPC routing and CoIP are mutually exclusive options that control how routing works. To determine the best option for your Outpost, see How to choose between CoIP and Direct VPC routing modes on AWS Outposts rack.

You can share the local gateway route table with other AWS accounts or organizational units using AWS Resource Access Manager. For more information, see Share your AWS Outposts resources.

Contents

- Direct VPC routing
- Customer-owned IP addresses
- Custom route tables

Direct VPC routing

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

Direct VPC routing mode does not support overlapping CIDR ranges.

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

The following examples illustrate direct VPC routing.

Examples

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

Example: Internet connectivity through the VPC

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

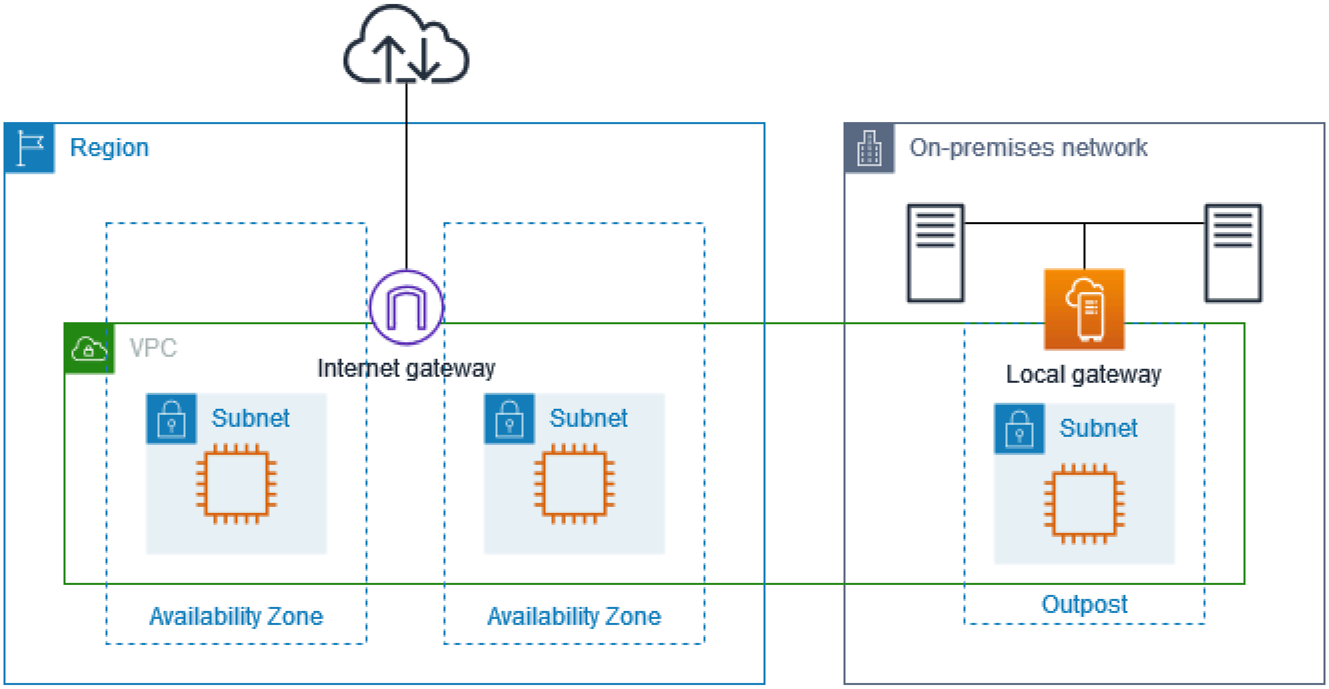
Consider the following configuration:

- The parent VPC spans two Availability Zones and has a subnet in each Availability Zone.
- The Outpost has one subnet.
- Each subnet has an EC2 instance.

- The local gateway uses BGP advertisement to advertise the private IP addresses of the Outpost subnet to the on-premises network.

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

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



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

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

Destination	Target	Comments
on-premises network CIDR	*local-gateway-id*	Sends traffic destined for the on-premises network to the local gateway.

Example: Internet connectivity through the on-premises network

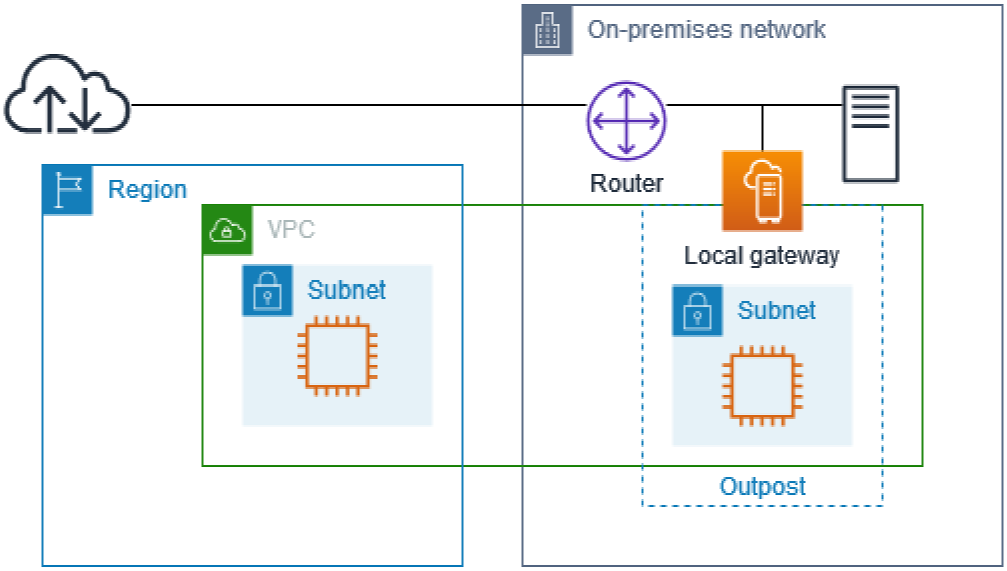
Instances in an Outpost subnet can access the internet through the on-premises network. Instances in the Outpost subnet do not need a public IP address or Elastic IP address.

Consider the following configuration:

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

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

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



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

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

Outbound access to the internet

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

Outbound access to the on-premises network

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

Inbound access from the on-premises network

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

Customer-owned IP addresses

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

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

Customer-owned IP addresses provide local or external connectivity to resources in your on-premises network. You can assign these IP addresses to resources on your Outpost, such as EC2 instances, by allocating a new Elastic IP address from the customer-owned IP pool, and then assigning it to your resource. For more information, see CoIP pools.

> **Note**
>
> For a customer-owned IP address pool, you must be able to route the address in your network.

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.

Examples

- Example: Internet connectivity through the VPC

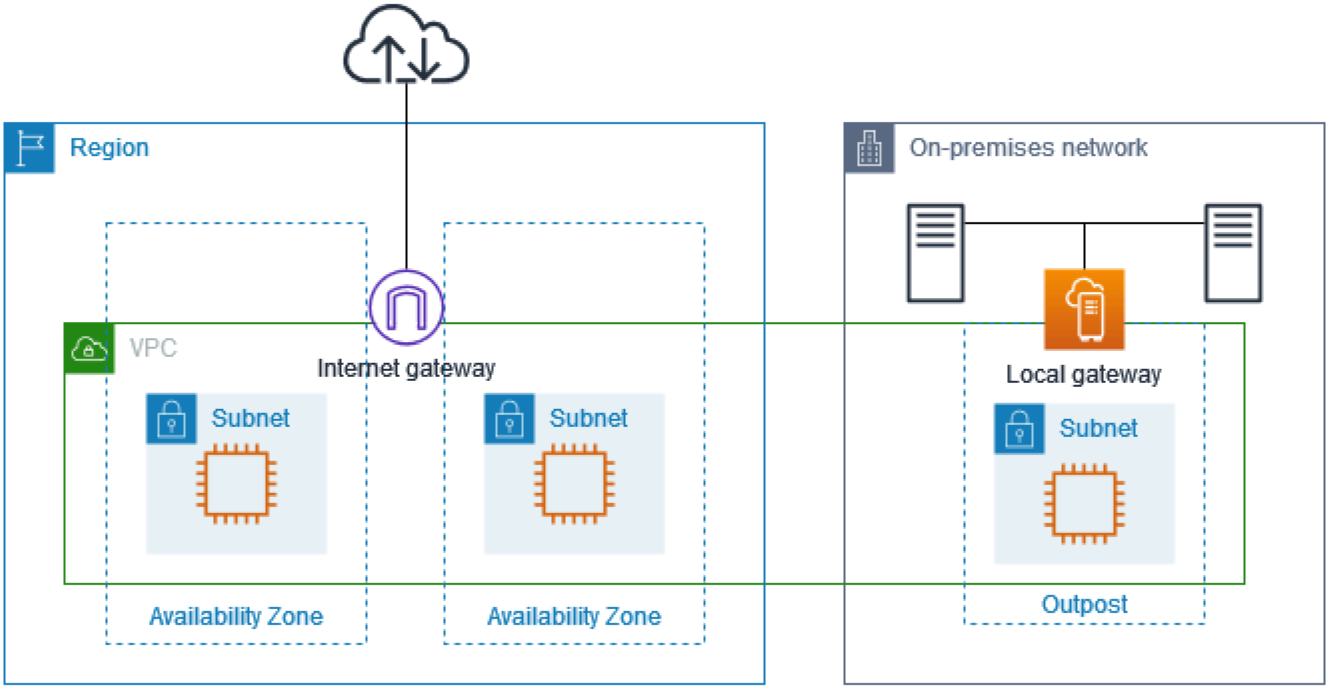
- Example: Internet connectivity through the on-premises network

Example: Internet connectivity through the VPC

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

Consider the following configuration:

- The parent VPC spans two Availability Zones and has a subnet in each Availability Zone.
- The Outpost has one subnet.
- Each subnet has an EC2 instance.
- There is a customer-owned IP address pool.
- The instance in the Outpost subnet has an Elastic IP address from the customer-owned IP address pool.
- The local gateway uses BGP advertisement to advertise the customer-owned IP address pool to the on-premises network.



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

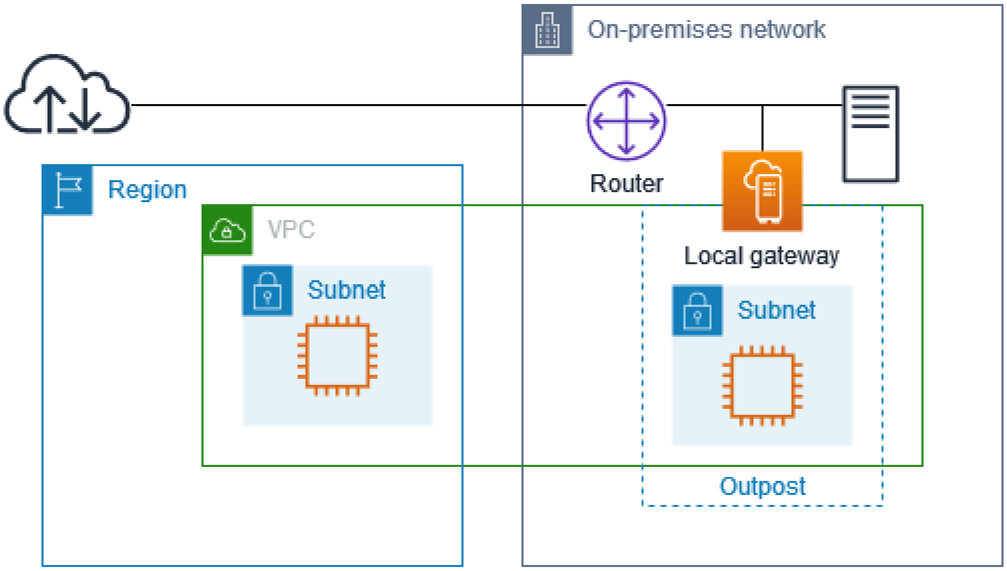
Destination	Target	Comments
VPC CIDR	Local	Provides connectivity between the subnets in the VPC.
0.0.0.0	*internet-gateway-id*	Sends traffic destined for the public internet to the internet gateway.
On-premises network CIDR	*local-gateway-id*	Sends traffic destined for the on-premises network to the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

Example: Internet connectivity through the on-premises network

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

Consider the following configuration:

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



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

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

Outbound access to the internet

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

Outbound access to the on-premises network

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

Inbound access from the on-premises network

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

Custom route tables

You can create a custom route table for your local gateway. The local gateway route table must have an association to a VIF group and a VPC. For step-by-step directions, see Configure local gateway connectivity.

Local gateway route table routes

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

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

Contents

Requirements and limitations

The following requirements and limitations apply:

- The target network interface must belong to a subnet on your Outpost and must be attached to an instance in that Outpost. A local gateway route can't target an Amazon EC2 instance on a different Outpost or in the parent AWS Region.
- The subnet must belong to a VPC that is associated to the local gateway route table.
- You must not exceed more than 100 network interface routes in the same route table.
- AWS prioritizes the most specific route, and if the routes match, we prioritize static routes over propagated routes.
- Interface VPC endpoints are not supported.
- BGP advertisement is only for subnets on an Outpost that have a route in the route table that targets the local gateway. If subnets do not have a route in the route table that targets the local gateway, then those subnets are not advertised with BGP.

- Only network interfaces that are attached to Outpost instances can communicate through the local gateway for that Outpost. Network interfaces that belong to the Outpost subnet but attached to an instance in the Region can't communicate through the local gateway for that Outpost.
- Requester-managed interfaces, such as those created for VPC endpoints, can't be reached from the on-premises network through the local gateway. They can be reached only from instances that are in the Outpost subnet.

The following NAT considerations apply:

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

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**.

   Edit the local gateway route table to add a static route that has the VIF Group as the target.
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**.

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

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

To switch modes or delete a local gateway route table

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

   To change the Region, use the Region selector in the top-right corner of the page.
3. On the navigation pane, choose **Local gateway route tables**.

4. Verify if the local gateway route table is associated with a VIF group. If it is associated, you must remove the association between the local gateway route table and the VIF group.

    a. Choose the ID of the local gateway route table.

    b. Choose the **VIF group association** tab.

    c. If one or more VIF groups are associated with the local gateway route table, choose **Edit VIF group association**.

    d. Clear the **Associate VIF group** checkbox.

    e. Choose **Save changes**.

5. Choose **Delete local gateway route table**.

6. In the confirmation dialog box, type **`delete`** and then choose **Delete**.

7. (Optional) Create a local gateway route table with a new mode.

    a. On the navigation pane, choose **Local gateway route tables**.

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

    c. Configure the local gateway route table using the new mode. For more information, see Create custom local gateway route tables.

VIF and VIF groups

Local gateway virtual interfaces (VIFs) is a logical interface component of Outposts racks that sets up VLAN, IP, and BGP connectivity between your Outposts networking devices and an on-premise networking device for local gateway connectivity. VIFs are created within VIF groups. VIF groups are logical groupings of VIFs and VIFs are created within VIF groups. You must create four local gateway VIFs within each VIF group.

To create a local gateway VIF group and VIFs

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. From the navigation pane, choose **LGW virtual interface (VIF) groups**.
4. Choose **Create VIF group**.
5. In the **LGW VIF group settings** section:

    - Enter a name for the VIF group.

- Choose the local gateway.
- Add your BGP ASN.

6. In the **LGW virtual interface** section:

   - Enter a name for the VIF.
   - Choose the link aggregation group (LAG).
   - Add a virtual local area network (VLAN).
   - Add local IP address.
   - Add a peer IP address.
   - Add the peer BGP ASN.

   > **Note**
   >
   > - You must create four local gateway virtual interfaces (LGW VIFs). Each VIF must be associated to a link aggregation group (LAG) within the VIF group. This ensures complete connectivity between your Outpost and on-premise network devices. Incomplete VIF groups can not be associated with local gateway route tables to create a routing domain.
   > - To ensure proper point-to-point connectivity between your Outpost and on-premises router, it's essential to match each LAG with its corresponding local gateway VIF. You can review these configurations in the **Link Aggregation Groups (LAGs)** section of the AWS Outposts console, where you'll find details about your LAGs and their associated service link VIFs networking configurations. This information helps you verify the correct mapping of your network connections between your Outpost and on-premises infrastructure.
   > - The local gateway IP addresses can’t overlap with the service link IP addresses that are associated with the same LAG. You can review your service link IP information in the **Link Aggregation Groups (LAGs)** section of the AWS Outposts console.
   > - A local gateway VIF is ready to transfer local gateway traffic when its state is available.

7. Choose **Create a LGW VIF Group**.

To delete a local gateway VIF

Before deleting a virtual interfaces (VIF) from a VIF group, ensure that the local gateway VIF group is not associated with a local gateway routing domain and is disconnected from local gateway route tables. Deleting a local gateway routing domain can impact your local gateway local network connectivity.

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. From the navigation pane, choose **LGW virtual interface (VIF) groups**.
4. Choose the VIF group that contains the VIF you want to delete.
5. Choose **Manage LGW VIFs**.
6. Select the VIF to be deleted.
7. Choose **Delete**.

To add a local gateway VIF

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. From the navigation pane, choose **LGW virtual interface (VIF) groups**.
4. Choose the VIF group that you want to add the VIF to.
5. Choose **Manage LGW VIFs**.
6. Choose **Add VIF**.
7. Provide the following information:

   - Enter a name for the VIF.
   - Choose the link aggregation group (LAG).
   - Add a virtual local area network (VLAN).
   - Add local IP address.
   - Add a peer IP address.
   - Add the peer BGP ASN.

Note

- You must create four local gateway virtual interfaces (LGW VIFs). Each VIF must be associated to a link aggregation group (LAG) within the VIF group. This ensures complete connectivity between your Outpost and on-premise network devices. Incomplete VIF groups can not be associated with local gateway route tables to create a routing domain.
- To ensure proper point-to-point connectivity between your Outpost and on-premises router, it's essential to match each LAG with its corresponding local gateway VIF. You can review these configurations in the **Link Aggregation Groups (LAGs)** section of the AWS Outposts console, where you'll find details about your LAGs and their associated service link VIFs networking configurations. This information helps you verify the correct mapping of your network connections between your Outpost and on-premises infrastructure.
- The local gateway IP addresses can’t overlap with the service link IP addresses that are associated with the same LAG. You can review your service link IP information in the **Link Aggregation Groups (LAGs)** section of the AWS Outposts console.
- A local gateway VIF is ready to transfer local gateway traffic when its state is available.

8. Choose **Save changes**.

To delete a VIF group

You can delete a VIF group if it is not associated with a local gateway routing 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. From the navigation pane, choose **Networking** and **LGW virtual interface (VIF) groups**.
4. Choose the VIF group that you want to delete.
5. On the VIF group page, choose **Disassociate LGW routing domain**.

> **Note**
>
> Deleting a local gateway routing domain can impact your local gateway local network connectivity.

6. On the **Delete LGW routing domain** window that appears, choose **Delete LGW routing domain**.
7. On the VIF group page, choose **Delete**.
8. On the **Delete LGW VIF group** window that appears, choose **Delete LGW VIF group**.

> **Note**
>
> Deleting a VIF group will delete all the VIFs in the group. You cannot undo this action.

Create a CoIP pool

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

Customer-owned IP pools are available for local gateway route tables in CoIP mode.

Use the following procedure to create a CoIP pool.

Console

To create a CoIP pool 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 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) To add a CIDR block, choose **Add new CIDR** and enter a range of customer-owned IP addresses.
9. Choose **Create CoIP pool**.

AWS CLI

To create a CoIP pool using the AWS CLI

1. Use the create-coip-pool command to create a pool of CoIP addresses for the specified local gateway route table.

```
aws ec2 create-coip-pool --local-gateway-route-table-id lgw-rtb-
abcdefg1234567890
```

   The following is example output.

```
{
    "CoipPool": {
        "PoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890",
        "PoolArn": "arn:aws:ec2:us-west-2:123456789012:coip-pool/ipv4pool-
coip-1234567890abcdefg"
    }
}
```

2. Use the create-coip-cidr command to create a range of CoIP addresses in the specified CoIP pool.

```
aws ec2 create-coip-cidr --cidr 15.0.0.0/24 --coip-pool-id ipv4pool-
coip-1234567890abcdefg
```

   The following is example output.

```
{
    "CoipCidr": {
        "Cidr": "15.0.0.0/24",
        "CoipPoolId": "ipv4pool-coip-1234567890abcdefg",
        "LocalGatewayRouteTableId": "lgw-rtb-abcdefg1234567890"
    }
}
```

After you create a CoIP pool, use the following procedure to assign an address to your instance.

Console

To assign a CoIP address to an instance using the console

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

AWS CLI

To assign a CoIP address to an instance using the AWS CLI

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

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

   The following is example output.

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

```
        }
    ]
}
```

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

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

   The following is example output.

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

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

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

   The following is example output.

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

Multiple local gateway routing domains

Multiple local gateway (LGW) routing domains is a feature available on second-generation AWS Outposts racks that enables you to create up to 10 isolated routing domains with independent network paths to your on-premises network. This capability enables both customer-owned IP (CoIP) and direct VPC routing (DVR) configurations to coexist on the same Outpost, with each routing domain configured independently for either CoIP or DVR mode.

A Virtual Interface (VIF) represents a network interface connecting your Outpost to your on-premises network. VIFs are grouped into **VIF Groups**, which are then associated with route tables to create routing domains. An LGW routing domain is the association of an **LGW Route Table** and **LGW VIF Group**. Each LGW routing domain operates as an independent network segment with its own route table, **VIF Group**, associated VPCs, and VLAN configurations.

> **Note**
>
> Multiple LGW routing domains feature is not available on first-generation AWS Outposts racks.

Traffic isolation with routing domains

IP address ranges must not overlap within a routing domain to prevent routing conflicts. Multiple on-premises VLANs can be configured within a single domain. Traffic isolation across routing domains works through multiple layers of separation:

- **Logical level** – Each routing domain maintains its own BGP sessions and independent route tables. VPCs are exclusively associated with one LGW routing domain per Outpost, preventing unintended cross-domain communication. This association is agnostic of the routing domain mode (DVR or CoIP).
- **Data plane level** – Traffic remains confined within assigned VLANs and VIFs, ensuring complete isolation.

Each VIF within a **VIF Group** requires specific configuration including:

- IP addresses for local and peer endpoints
- VLAN tags for traffic segregation
- BGP parameters for route exchange
- Association with the appropriate Link Aggregation Group (LAG)

Creating routing domains

You can create and manage routing domains through the AWS Management Console or AWS CLI.

To create a routing domain:

1. **Create an LGW route table**

   Create an LGW route table in DVR or CoIP routing mode, and associate your VPC with it.

   The route table will show as inactive until both a VPC and VIF group are associated. A VPC can be associated with 1 LGW route table per Outpost.

2. **Create a VIF group**

   Create a VIF group for your routing domain.

   Every LGW routing domain requires a unique VIF group and LGW route table for network traffic isolation. Second-generation Outposts racks require 4 VIFs per VIF group. Each VIF maps to: 1 VLAN → 1 LAG. VIF group and individual VIF status will show as "Pending" for up to 10 minutes.

3. **Configure each VIF**

   Configure each VIF within the group with the required network parameters.

   Assign each VIF:

   - IP addresses
   - VLAN tags (can be unique or the same as existing VLANs, as long as there's no VIF IP conflict on that LAG)
   - BGP parameters
   - LAG assignment

4. **Create the routing domain**

   Create the LGW routing domain by associating the route table and VIF group.

   Your VIF group and route table can only be associated to 1 routing domain. This creates the 1:1 mapping between the VIF group and route table.

5. **Add a route to the VIF group**

   Add a route in the LGW route table that directs traffic to the VIF group.

   This route enables traffic to flow from your Outpost through the local gateway to your on-premises network via the VIF group. Without this route, the routing domain will not forward traffic even though the VIF group and route table are associated.

6. **Update your Customer Networking Devices (CNDs)**

Update your Customer Networking Devices (CNDs) to allow the VIF VLANs to send traffic across the physical uplinks that correspond to each LAG.

Note: This step requires configuration on your on-premises CNDs. Each VLAN must be configured as a trunk port on the physical interface(s) that connect to the Outpost Networking Devices (ONDs) to complete BGP peering setup. Consult your network equipment documentation for VLAN trunking configuration specific to your hardware.

Monitoring routing domains

You can monitor the following:

- VIF status and connectivity
- BGP session health
- Route propagation
- Traffic flow through each routing domain

Best practices and considerations

Network planning

- Maintain non-overlapping IP ranges between routing domains.
- Plan for sufficient VIFs per **VIF group** based on your configuration.
- Verify VIF requirements based on your Outpost generation and rack configuration.

Documentation and organization

- Use clear naming conventions for **VIF groups** and routing domains.
- Tag LAG IDs for easier identification.
- Review service link VIF information to select appropriate LAG IDs.

Configuration restrictions

Routing domains

- Each **VIF Group** can be associated with only one routing domain at a time.
- To associate a **VIF Group** with a different routing domain, you must first delete the existing **LGW Route Table** and routing domain association.
- Each routing domain must be configured for either CoIP or DVR mode during **LGW Route Table** creation.

IP addressing

- Local addresses cannot overlap with peer and local addresses of any local gateway VIF or service link VIF on the same LAG.
- Local addresses cannot overlap with peer and local addresses of any local gateway VIF in the same **VIF group**.

VLAN configuration

- VLAN IDs cannot overlap with service link VIF on the same LAG.
- VLAN IDs can overlap with other local gateway VIFs on the same **VIF group**/LAG.

BGP

- Local BGP ASN matches the BGP for the **VIF group**.
- Peer BGP ASN has no overlapping prevention.

Capacity management for AWS Outposts

An Outpost provides a pool of AWS compute and storage capacity at your site as a private extension of an Availability Zone in an AWS Region. Because the compute and storage capacity available in the Outpost is finite and determined by the size and number of assets that AWS installs at your site, you get to decide how much Amazon EC2 and Amazon EBS on AWS Outposts capacity you need to run your initial workloads, accommodate future growth, and to provide extra capacity to mitigate server failures and maintenance events.

Topics

- View AWS Outposts capacity
- Modify AWS Outposts instance capacity
- Troubleshooting capacity task issues

View AWS Outposts capacity

You can view the capacity configuration at the instance or Outpost level.

To view capacity configuration for your Outpost using the console

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. From the left navigation pane, choose **Outposts**.
3. Choose the Outpost.
4. On the Outpost details page select either **Instance view** or **Rack view**.

   - **Instance view** - Provides information on the instances configured on the Outposts and the distribution of instances by size and family.
   - **Rack view** - Provides visualization of the instances on each asset within each Outpost and allows you to select **Modify instance capacity** to make changes to instance capacity.

Modify AWS Outposts instance capacity

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

Considerations

Consider the following before modifying instance capacity:

- Capacity tasks can be run only by the AWS account that owns the Outpost resources (owner). Consumers cannot run capacity tasks. For more information about owners and consumers, see Share your AWS Outposts resources.
- Instances sizes and quantities can be defined at the Outpost level or at an individual asset level.
- Capacity is configured automatically across an asset or all the assets in an Outpost based on possible configurations and best practices.
- While a capacity task is running, the assets associated with the selected outpost may be isolated. For this reason we recommend creating a capacity task only when you don't expect to launch new instances on your Outposts.
- You can choose to run the capacity task instantly or to keep attempting periodically over the next 48 hours. Choosing to run instantly requires less asset isolation time, but the task might fail if instances need to be stopped to run the task. Choosing to run periodically allows more time to stop instances before the task would fail, but assets may be isolated for longer.
- It is possible for valid capacity configurations to not utilize all of the available vCPU on an asset. When this is the case, a message at the end of the **Instance type** section will inform you that you are under capacity, but will allow the configuration to be applied as requested.
- When you modify an Outpost in the console, not all supported instances are shown because mixing disk-backed instances with non-disk-backed instances is not fully supported in console. To access all possible instances, utilize the StartCapacityTask API.
- When defining capacity for an Outpost, all instance families and types will be included in the re-configuration unless they are listed as instances to avoid.
- You can only modify your existing Outposts capacity configuration to use valid Amazon EC2 instance sizes from instance families supported on your respective asset model.
- If you have instances running on your Outpost that you do not want to stop to run a capacity task, select their respective Instance ID under the section **Instances to keep as-is – *optional*** and make sure to retain the necessary quantity of this instance size in your updated capacity configuration. This will retain instances being used to support production workloads while a capacity task runs.

- When configuring an asset with multiple instance sizes within an instance family, use **Auto-balance** to make sure you aren't attempting to over or under-provision your droplet. Over-provisioning is not supported, and will cause a capacity task failure.
- If you want to completely reconfigure an instance family on your Outpost without retaining any of the instance sizes from the original capacity configuration, you must stop any running instances of that family on your Outpost before executing the capacity task. If the instance is owned by another account or is used by a layered service running on the Outpost, you must use the instance owner account to stop the instance or service instance.
- Several capacity tasks can run in parallel as long as they apply to mutually exclusive sets of AssetIDs. For example, you can create several asset-level capacity tasks for different AssetIDs at the same time. However, if there is a running Outpost-level task, you cannot create another Outpost or asset-level task at the same time. Similarly, if there is a running asset-level task, you cannot create an Outpost-level task or an asset-level task on the same AssetID at the same time.

To modify capacity configuration for your Outpost using the console

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. From the left navigation pane, choose **Capacity tasks**.
3. On the **Capacity tasks** page, choose **Create capacity task**.
4. On the **Getting started** page, choose the order, Outpost, or asset to configure.
5. To modify capacity, specify an option for **Method of modification**: e steps in the console or upload a JSON file.

   - **Modify capacity configuration plan** to use the steps in the console
   - **Upload a capacity configuration plan** to upload a JSON file

> **Note**
>
> - To prevent capacity management from recommending specific instances to stop, specify the instances that should not be stopped. These instances will be excluded from the list of instances to stop.

Console steps

1. Choose **Instance view** or **Rack view**.
2. Choose **Modify an Outpost capacity configuration** or **Modify** on a single asset.
3. Choose an Outpost or asset if different than the current selection.
4. Choose to either run this capacity task immediately or periodically over 48 hours.
5. Choose **Next**.
6. On the **Configure instance capacity** page, each instance type shows one instance size with the maximum quantity preselected. To add more instance sizes, choose **Add instance size**.
7. Specify the instance quantity and note the capacity that is displayed for that instance size.
8. View the message at the end of each instance-type section that informs you if you are over or under capacity. Make adjustments at the instance size or quantity level to optimize your total available capacity.
9. You can also request AWS Outposts to optimize the instance quantity for a specific instance size. To do so:

    a. Choose the instance size.

    b. Choose **Auto-balance** at the end of the related instance-type section.
10. For each instance type, ensure that the instance quantity is specified for at least one instance size.
11. Optionally, choose instances to keep as-is.
12. Choose **Next**.
13. On the **Review and create** page, verify the updates that you are requesting.
14. Choose **Create**. AWS Outposts creates a capacity task.
15. On the capacity task page, monitor the status of the task.

Upload a JSON file

1. Choose **Upload a capacity configuration**.
2. Choose **Next**.
3. On the **Upload capacity configuration plan** page, upload the JSON file that specifies the instance type, size, and quantity. Optionally, you can specify the InstancesToExclude, and TaskActionOnBlockingInstances parameters in the JSON file.

Example

Example JSON file:

```
{
  "InstancePools": [
    {
      "InstanceType": "c5.24xlarge",
      "Count": 1
    },
    {
      "InstanceType": "m5.24xlarge",
      "Count": 2
    }
  ],
  "InstancesToExclude": {
    "AccountIds": [
      "111122223333"
    ],
    "Instances": [
      "i-1234567890abcdef0"
    ],
    "Services": [
      "ALB"
    ]
  },
  "TaskActionOnBlockingInstances": "WAIT_FOR_EVACUATION"
}
```

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

Troubleshooting capacity task issues

Review the following known issues to resolve an issue related to capacity management in a new order. If you do not see your issue listed, contact Support.

Order *oo-xxxxxx* is not associated with Outpost ID *op-xxxxx*

This issue occurs when you use the AWS CLI or API to run the `StartCapacityTask` and the Outpost ID in the request does not match the Outpost ID in the order.

To resolve this issue:

1. Sign in to AWS.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. From the navigation pane, choose **Orders**.
4. Select the order and verify that the order status is one of the following: `PREPARING`, `IN_PROGRESS`, or `ACTIVE`.
5. Note the Outpost ID in the order.
6. Enter the correct Outpost ID in the `StartCapacityTask` API request.

The capacity plan includes instance types that are not supported

This issue occurs when you use the AWS CLI or API to create or modify the capacity task and the request contains unsupported instances types.

To resolve this issue, use the console or CLI.

Use the console

1. Sign in to AWS.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. From the navigation pane, choose **Capacity task**.
4. Use the **Upload a capacity configuration** option to upload a JSON with the same list of instance types.
5. The console displays an error message with the list of supported instance types.
6. Correct the request to remove the unsupported instance types.
7. Create or modify the capacity task on the console using the corrected JSON or use the CLI or API with this corrected list of instance types.

Use the CLI

1. Use the GetOutpostSupportedInstanceTypes command to see the list of supported instance types.
2. Create or modify the capacity task with the correct list of instance types.

No Outpost with Outpost ID *op-xxxxx*

This issue occurs when you use the AWS CLI or API to run the `StartCapacityTask` and the request contains an Outpost ID that is not valid for one of the following reasons:

- The Outpost is in a different AWS Region.
- You do not have permissions to this Outpost.
- The Outpost ID is incorrect.

To resolve this issue:

1. Note the AWS Region that you used in the `StartCapacityTask` API request.
2. Use the `ListOutposts` API action to get a list of Outposts that you own in the AWS Region.
3. Check if the Outpost ID is listed.
4. Enter the correct Outpost ID in the `StartCapacityTask` request.
5. If you do not find the Outpost ID, use the `ListOutposts` API action again to check if the Outpost exists in a different AWS Region.

Active CapacityTask cap-*XXXX* already found for Outpost op-*XXXX*

This issue occurs when you use the AWS Outposts console or API to run StartCapacityTask on an Outpost and there is already a running capacity task for the Outpost. A capacity task is considered running if it has any of the following statuses: `REQUESTED`, `IN_PROGRESS`, `WAITING_FOR_EVACUATION`, or `CANCELLATION_IN_PROGRESS`.

To resolve this issue, use the AWS Outposts console or CLI.

Use the console

1. Sign in to AWS.

2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. From the navigation pane, choose **Capacity tasks**.
4. Ensure that there are no running capacity tasks for the OutpostId.
5. If there are running capacity tasks for the OutpostId, wait for them to terminate, or cancel them if desired.
6. When there no running capacity tasks for the requested OutpostId, retry your request to create the capacity task.

Use the CLI

1. Use the ListCapacityTasks command to find running capacity tasks for the Outpost.
2. Wait for all running capacity tasks to terminate, or cancel them if desired.
3. When there no running capacity tasks for the requested OutpostId, retry your request to create the capacity task.

Active CapacityTask cap-*XXXX* already found for Asset *XXXX* on Outpost op-XXXX

This issue occurs when you use the AWS Outposts console or API to run StartCapacityTask on an asset and there is already a running capacity task for the asset. A capacity task is considered running if it has any of the following statuses: `REQUESTED`, `IN_PROGRESS`, `WAITING_FOR_EVACUATION`, or `CANCELLATION_IN_PROGRESS`.

To resolve this issue, use the AWS Outposts console or CLI.

Use the console

1. Sign in to AWS.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. From the navigation pane, choose **Capacity tasks**.
4. Ensure that there are no running capacity tasks for the OutpostId and no running asset-level capacity Tasks for the AssetId.
5. If there are running capacity tasks, wait for them to terminate, or cancel them if desired.
6. When there no running capacity tasks, retry your request to create the capacity task.

Use the CLI

1. Use the ListCapacityTasks command to find running capacity tasks for the OutpostID and AssetID.
2. Ensure that there are no running Outpost-level capacity tasks for the OutpostId, and no running asset-level capacity Tasks for the AssetId.
3. If there are running capacity tasks, wait for them to terminate, or cancel them if desired.
4. Retry your request to create the capacity task.

AssetId=*XXXX* is not valid for Outpost=op-*XXXX*

This issue occurs when you use the AWS Outposts console or API to run StartCapacityTask on an asset and the AssetID is not valid for one of the following reasons:

- The asset is not associated with the Outpost.
- The asset is isolated.

To resolve this issue, use the AWS Outposts console or CLI.

Use the console

1. Sign in to AWS.
2. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
3. Choose **Rack view** for the Outpost.
4. Verify that the requested AssetId is associated with the Outpost, and that it is not marked as an Isolated Host.
    a. If the Asset is isolated, this may be because a capacity task is running on it. You can navigate to the capacity tasks panel and check if there are any running Outpost or asset-level tasks for the OutpostId and AssetId. If there are, then wait for the task to terminate and for the asset to become available again.
    b. If there are no running capacity tasks for an isolated asset, then the asset may be degraded.
5. After you verify that the asset exists and is in a valid state, retry your request to create the capacity task.

Use the CLI

1. Use the ListAssets command to find the assets associated with the OutpostID.
2. Verify that the requested AssetId is associated with the Outpost, and that its State is `ACTIVE`.

    a. If the asset State is not ACTIVE, this may be because a capacity task is running on it. Use the ListCapacityTasks command to determine if there are running Outpost or asset-level tasks for the OutpostId and AssetId. If there are, then wait for the task to terminate and for the asset to become ACTIVE again.

    b. If there are no running capacity tasks for an isolated asset, then the asset may be degraded.

3. After you verify that the asset exists and is in a valid state, retry your request to create the capacity task.

Share your 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 can't modify instances that consumers launch into Capacity Reservations that they have shared.

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

An Outpost owner can share Outpost resources with:

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

Contents

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

Shareable Outpost resources

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

These are the resources available for Outposts racks.

- **Allocated Dedicated Hosts** – Consumers with access to this resource can:
  - Launch and run EC2 instances on a Dedicated Host.
- **Capacity Reservations** – Consumers with access to this resource can:
  - Identify Capacity Reservations shared with them.
  - Launch and manage instances that consume Capacity Reservations.
- **Customer-owned IP address (CoIP) pools** – Consumers with access to this resource can:
  - Allocate and associate customer-owned IP addresses with instances.
- **Local gateway route tables** – Consumers with access to this resource can:
  - Create and manage VPC associations to a local gateway.
  - View configurations of local gateway route tables and virtual interfaces.
  - Create a VPC subnet route where the target is a local gateway.
- **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.
- **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 can't 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 IDs for the Availability Zones in your account

1. Navigate to the AWS RAM console in the AWS RAM console.
2. The AZ IDs for the current Region are displayed in the **Your AZ ID** panel on the right-hand side of the screen.

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

Sharing an Outpost resource

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

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

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

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

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

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

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

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

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

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

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

Use the create-resource-share command.

Unsharing a shared Outpost resource

When you unshare your Outpost with a consumer, the consumer can no longer do the following:

- View the Outpost in the AWS Outposts console.
- Create new subnets on the Outpost.
- Create new Amazon EBS volumes on the Outpost.
- View the Outpost details and instance types using the AWS Outposts console or the AWS CLI.

Subnets, volumes, or instances that the consumer created during the shared period are not deleted and the consumer can continue to do the following:

- Access and modify these resources.
- Launch new instances on an existing subnet that the consumer created.

To prevent the consumer from accessing their resources and launching new instances on your Outpost, request that the consumer delete their resources.

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

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 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.

Third-party block storage on Outposts racks

With Outposts racks, you can leverage existing data you're stored on third-party storage arrays. You can specify external block data volumes and external block boot volumes for your EC2 instances on Outposts. Using this integration, you can use external block data and boot volumes backed by third-party vendors such as, Dell PowerStore, HPE Alletra Storage MP B10000, NetApp on-premises enterprise storage arrays, and Pure Storage FlashArray storage systems.

Considerations

- Available on Outposts racks and Outposts 2U servers. Not available on Outposts 1U servers.
- Available in all AWS Regions where second-generation Outposts racks are supported.
- Available at no extra charge.
- You are responsible for the configuration and day-to-day management of the storage array. You also create and manage the external block volumes on the storage array. If you have issues with the hardware, software, or connectivity for the storage array, contact the third-party storage vendor.

Note

The block volume stored on your external storage array contains the operating system that will be booted into an EC2 instance on Outposts. Launching an AMI that is backed by external storage arrays is not supported. To launch an AMI, EBS or instance storage is required for Outposts racks.

External block data volumes

After you provision and configure block data volumes backed by a compatible third-party storage system, you can attach the volumes to your EC2 instances when you launch them. If you configure the volumes for multi-attach on the storage array, you can attach a volume to multiple EC2 instances.

Key steps

- AWS technicians configure the local gateway to ensure connectivity between the Outpost subnets and the local network.

- You use the management interface for the external storage array to create the volume. Then, you'll configure the initiator mapping by created a new Initiator Group and adding the iSCSI Qualified Name (IQN) of the target EC2 instance to this group. This associates the external block data volume with the EC2 instance.
- You add the external data volume when you launch the instance. You'll need the Initiator IQN, the target IP address, the port, and the IQN of the external storage array. For more information, see Launch an instance on the Outpost.

For more information, see Simplifying the use of third-party block storage with AWS Outposts.

External block boot volumes

Booting an EC2 instance on Outposts from external storage arrays provides a centralized, cost-effective, and efficient solution for on-premises workloads that depend on third-party storage. You can choose between the following options:

iSCSI SAN boot

Provides direct booting from the external storage array. Utilizes an AWS-provided iPXE helper AMI so that the instances can boot from a network location. When iPXE is combined with iSCSI, the EC2 instance treats the remote iSCSI target (the storage array) as a local disk. All read and write operations from the operating system are performed on the external storage array.

iSCSI or NVMe-over-TCP LocalBoot

Launches EC2 instances using a copy of the boot volume retrieved from the storage array, leaving the original source image unmodified. We launch a helper instance using a LocalBoot AMI. This helper instance copies the boot volume from the storage array to the instance store of the EC2 instance, and acts as an iSCSI initiator or NVMe-over-TCP host. Finally, the EC2 instance reboots using the local instance store volume.

Because instance store is temporary storage, the boot volume is deleted when the EC2 instance is terminated. Therefore, this option is suitable for read-only boot volumes, such as those used in virtual desktop infrastructure (VDI).

You can't boot EC2 Windows instances using NVMe-over-TCP LocalBoot. This is only supported using EC2 Linux instances.

For more information, see Deploying external boot volumes for use with AWS Outposts.

Security in AWS Outposts

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

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

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

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

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

Contents

Data protection in AWS Outposts

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

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

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

Encryption at rest

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

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

Encryption in transit

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

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

Data deletion

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

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

Identity and access management (IAM) for AWS Outposts

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

Contents

How AWS Outposts works with IAM

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

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

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

Identity-based policies for AWS Outposts

Supports identity-based policies: Yes

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

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for AWS Outposts

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

Policy actions for AWS Outposts

Supports policy actions: Yes

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

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Include actions in a policy to grant permissions to perform the associated operation.

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

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

```
outposts
```

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

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

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

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

Policy resources for AWS Outposts

Supports policy resources: Yes

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

The `Resource` JSON policy element specifies the object or objects to which the action applies. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, use a wildcard (*) to indicate that the statement applies to all resources.

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

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

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

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

Policy condition keys for AWS Outposts

Supports service-specific policy condition keys: Yes

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

The `Condition` element specifies when statements execute based on defined criteria. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

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

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

ABAC with AWS Outposts

Supports ABAC (tags in policies): Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes called tags. You can attach tags to IAM entities and AWS resources, then design ABAC policies to allow operations when the principal's tag matches the tag on the resource.

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

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

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

Using temporary credentials with AWS Outposts

Supports temporary credentials: Yes

Temporary credentials provide short-term access to AWS resources and are automatically created when you use federation or switch roles. AWS recommends that you dynamically generate temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM and AWS services that work with IAM in the *IAM User Guide*.

Cross-service principal permissions for AWS Outposts

Supports forward access sessions (FAS): Yes

Forward access sessions (FAS) use the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. For policy details when making FAS requests, see Forward access sessions.

Service-linked roles for AWS Outposts

Supports service-linked roles: Yes

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

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

AWS Outposts policy examples

By default, users and roles don't have permission to create or modify AWS Outposts resources. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies.

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

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

Contents

- Policy best practices
- Example: Using resource-level permissions

Policy best practices

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

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.
- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see Validate policies with IAM Access Analyzer in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Secure API access with MFA in the *IAM User Guide*.

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

Example: Using resource-level permissions

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

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "outposts:GetOutpost",
            "Resource": "arn:aws:outposts:us-east-1:111122223333:outpost/
op-1234567890abcdef0"
        }
    ]
}
```

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

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "outposts:GetSite",
            "Resource": "arn:aws:outposts:us-east-1:111122223333:site/
os-0abcdef1234567890"
        }
    ]
}
```

Service-linked roles for AWS Outposts

AWS Outposts uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a type of service role that is linked directly to AWS Outposts. AWS Outposts defines service-linked roles and includes all the permissions that it 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 can't 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.

Service-linked role permissions for AWS Outposts

AWS Outposts uses the service-linked role named **AWSServiceRoleForOutposts_*`OutpostID`***. This role grants Outposts permissions to manage networking resources to enable private connectivity on your behalf. This role also allows Outposts to create and configure network interfaces, manage security groups, and attach interfaces to service link endpoint instances. These permissions are necessary for establishing and maintaining the secure, private connection between your on-premises Outpost and AWS services, ensuring reliable operation of your Outpost deployment.

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

- `outposts.amazonaws.com`

Service-linked role policies

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

- AWSOutpostsServiceRolePolicy
- AWSOutpostsPrivateConnectivityPolicy_*`OutpostID`*

AWSOutpostsServiceRolePolicy

The `AWSOutpostsServiceRolePolicy` policy enables 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:DescribeSubnets` on all AWS resources
- Action: `ec2:DescribeVpcEndpoints` on all AWS resources
- Action: `ec2:CreateNetworkInterface` on the following AWS resources:

  ```
  "arn:*:ec2:*:*:vpc/*",
  "arn:*:ec2:*:*:subnet/*",
  "arn:*:ec2:*:*:security-group/*"
  ```

- Action: `ec2:CreateNetworkInterface` on the AWS resource `"arn:*:ec2:*:*:network-interface/*"` that match the following condition:

  ```
  "ForAnyValue:StringEquals" : { "aws:TagKeys": [ "outposts:private-
  connectivity-resourceId" ] }
  ```

- Action: `ec2:CreateSecurityGroup` on the following AWS resources:

  ```
  "arn:*:ec2:*:*:vpc/*"
  ```

- Action: `ec2:CreateSecurityGroup` on the AWS resource `"arn:*:ec2:*:*:security-group/*"` that match the following condition:

  ```
  "ForAnyValue:StringEquals": { "aws:TagKeys": [ "outposts:private-
  connectivity-resourceId" ] }
  ```

AWSOutpostsPrivateConnectivityPolicy_OutpostID

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}}*"}},
"StringEquals": {"ec2:CreateAction" : ["CreateSecurityGroup",
 "CreateNetworkInterface"]}}
```

- Action: `ec2:RevokeSecurityGroupIngress` on all AWS resources that match the following condition:

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

- Action: `ec2:RevokeSecurityGroupEgress` on all AWS resources that match the following condition:

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

- Action: `ec2:DeleteNetworkInterface` on all AWS resources that match the following condition:

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

- Action: `ec2:DeleteSecurityGroup` on all AWS resources that match the following condition:

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

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*.

Create 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 options.

Edit 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 can't 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 Update a service-linked role in the *IAM User Guide*.

Delete 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.

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.

You must delete your Outpost before you can delete the AWSServiceRoleForOutposts_*OutpostID* service-linked role.

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

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

Contact AWS Enterprise Support to delete your Outpost.

To manually delete the service-linked role using IAM

For more information, see Delete 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 the FAQs for Outposts racks.

AWS managed policies for AWS Outposts

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

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

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

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

AWS managed policy: AWSOutpostsServiceRolePolicy

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

AWS Outposts updates to AWS managed policies

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

Change	Description	Date
Updates to the AWS Identity and Access Management service-linked role AWSServiceRoleForOutposts_*OutpostID*	The **AWSServiceRoleForOutposts_*OutpostID*** service-linked role permissions are updated to refine how AWS Outposts manages networking resources for private connectivity, with more precise controls over network interface and security group operations needed for service link endpoint instances.	April 18, 2025
AWS Outposts started tracking changes	AWS Outposts started tracking changes for its AWS managed policies.	December 03, 2019

Infrastructure security in AWS Outposts

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

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

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.

- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

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 or to Amazon GuardDuty for analysis. Data needs to be sent back to the Region for publication to these services, so it is not visible from CloudWatch or other services when the Outpost is in a disconnected state.

Tamper monitoring on AWS Outposts equipment

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

Resilience in AWS Outposts

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

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

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

Availability Zone. For more information about Regions and Availability Zones, see AWS Global Infrastructure.

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

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

Compliance validation for AWS Outposts

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

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

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Internet access for AWS Outposts workloads

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

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

Internet access through the parent AWS Region

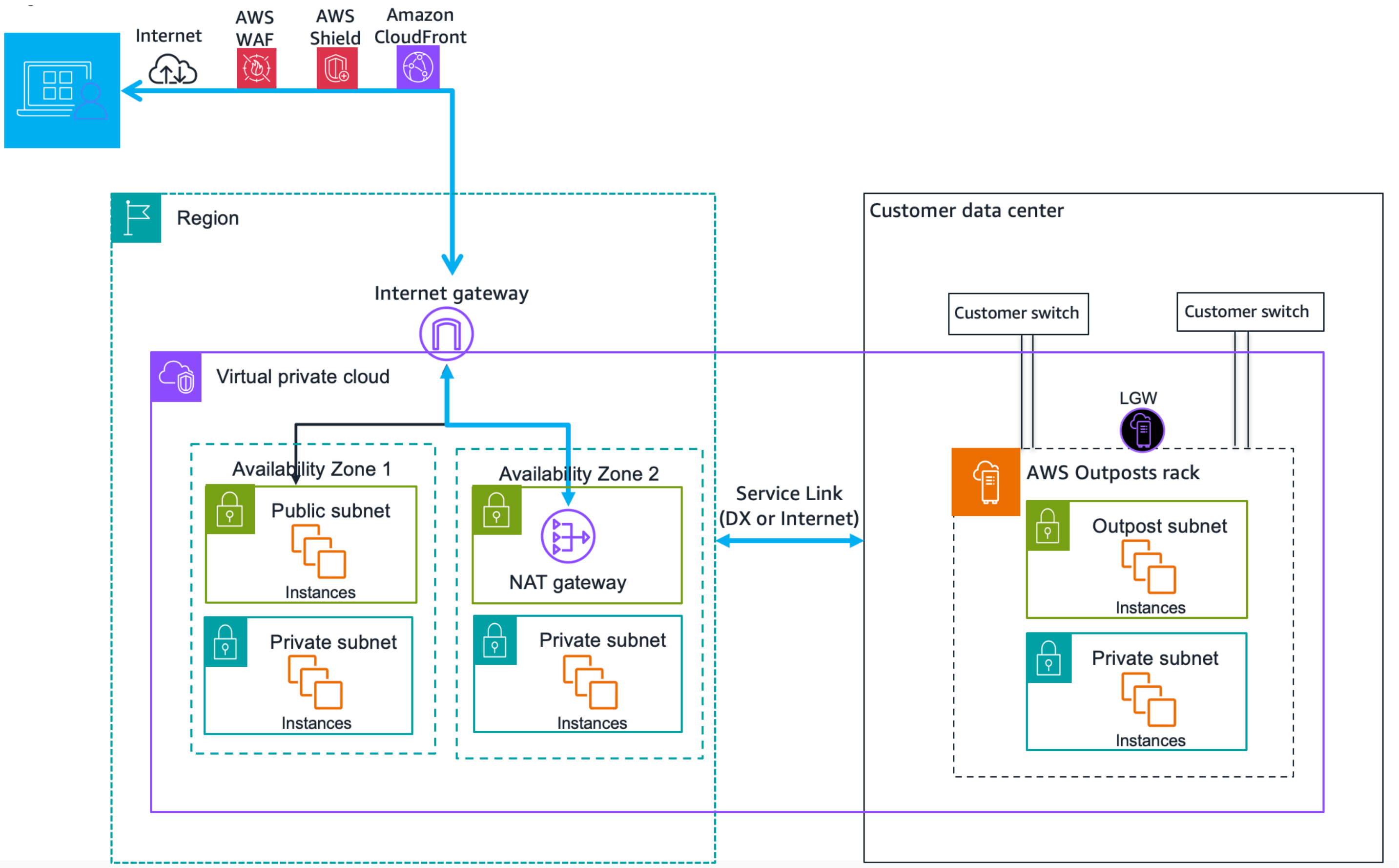
In this option, the workloads in the Outposts access the internet through the service link and then through the internet gateway (IGW) in the parent AWS Region. The outbound traffic to the internet can be through the NAT gateway instantiated in your VPC. For additional security for your ingress and egress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

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

Considerations

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

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



Internet access through your local data center's network

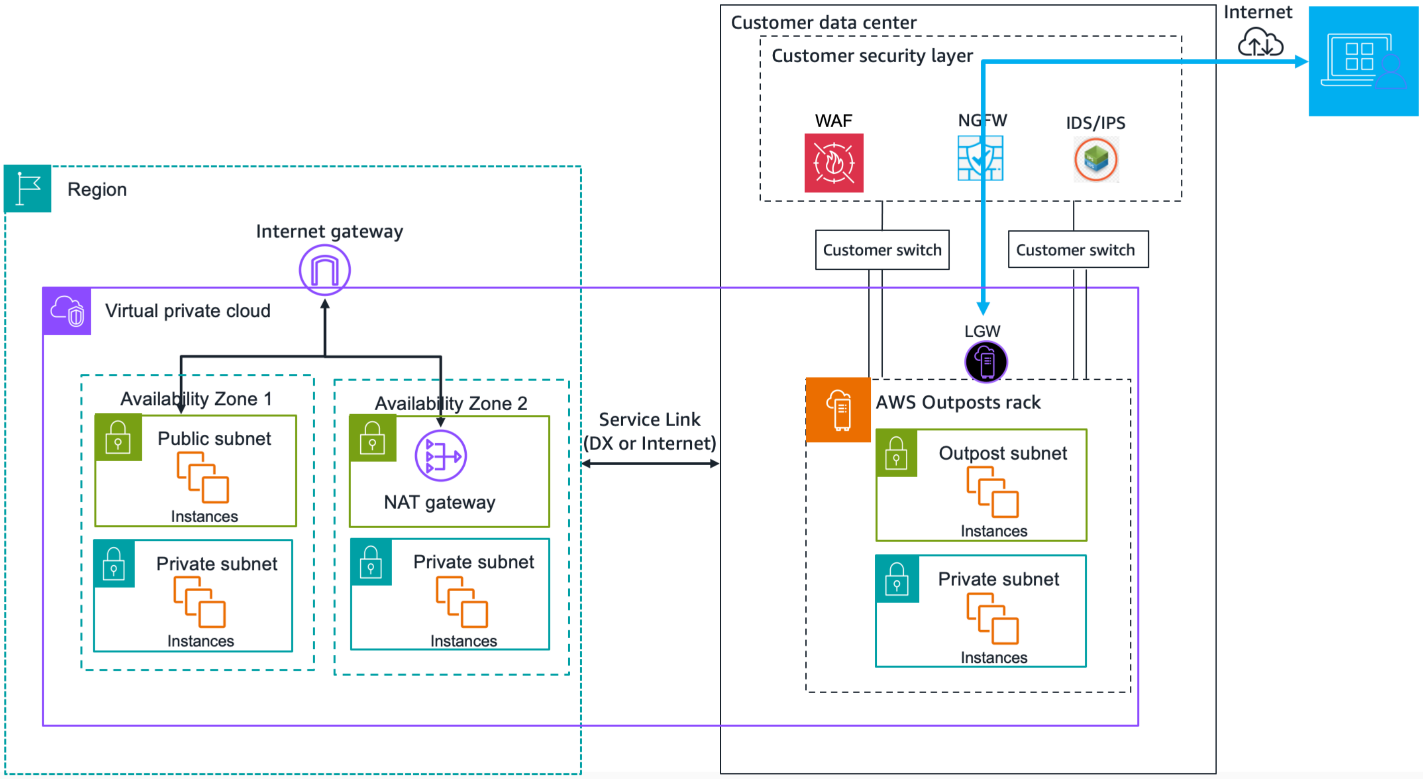
In this option, the workloads residing in the Outposts access the internet through your local data center. The workload traffic accessing the internet traverses through your local internet point of presence and egress locally. The security layer of your local data center’s network is responsible for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

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

Considerations

- Use this option when:
  - Your workloads require low latency access to internet services.
  - You prefer to avoid incurring Data Transfer Out (DTO) charges.
  - You want to preserve the service link bandwidth for control plane traffic.
- Your security layer is responsible for securing Outposts workload traffic.
- If you opt for Direct VPC Routing (DVR), then you must ensure that the Outposts CIDRs do not conflict with the on-premises CIDRs.
- If the default route (0/0) is propagated through the local gateway (LGW), then instances may not be able to get to the service endpoints. Alternatively, you can choose VPC endpoints to reach the desired service.

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


Customer data center
Customer security layer
WAF
NGFW
IDS/IPS
Internet
Customer switch
Customer switch
LGW
AWS Outposts rack
Outpost subnet
Instances
Private subnet
Instances
Region
Internet gateway
Virtual private cloud
Availability Zone 1
Public subnet
Instances
Private subnet
Instances
Availability Zone 2
NAT gateway
Private subnet
Instances
Service Link
(DX or Internet)

Monitor your Outposts rack

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 rack 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 Outposts racks.

CloudTrail logs

Use AWS CloudTrail to capture detailed information about the calls made to AWS APIs. 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 Log API calls using CloudTrail.

VPC Flow Logs

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

Traffic Mirroring

Use Traffic Mirroring to copy and forward network traffic from your Outposts rack to out-of-band security and monitoring appliances. You can use the mirrored traffic for content inspection, threat monitoring, or troubleshooting. For more information, see the Amazon VPC Traffic Mirroring Guide.

AWS Health Dashboard

The 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, 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 Outposts racks

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

Metrics

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

Contents

- Amazon EBS metrics
- Virtual interface metrics
- Outposts metrics

Instance metrics

The following metrics are available for Amazon EC2 instances.

Metric	Dimension	Description
`InstanceFamilyCapacityAvailability`	`InstanceFamily` and `OutpostId`	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).
`InstanceFamilyCapacityUtilization`	`Account`, `InstanceFamily` , and `OutpostId`	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

Metric	Dimension	Description
		Statistics: The most useful statistics are `Average` and `pNN.NN` (percentiles).
`InstanceTypeCapacityAvailability`	`InstanceType` and `OutpostId`	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).
`InstanceTypeCapacityUtilization`	`Account`, `InstanceType` , and `OutpostId`	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).

Metric	Dimension	Description
`UsedInstanceType_Count`	`Account`, `InstanceType` , and `OutpostId`	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

Metric	Dimension	Description
`AvailableInstanceType_Count`	`InstanceType` and `OutpostId`	The number of available instance types. This metric includes the `AvailableReservedInstances` count. To determine the number of instances that you can reserve, subtract the `AvailableReservedInstances` count from the `AvailableInstanceType_Count` count. `Number of instances that you can reserve = AvailableInstanceType_Count - AvailableReservedInstances` This metric does not include capacity for any Dedicated Hosts configured on the Outpost. **Unit**: Count **Maximum resolution**: 5 minutes

Metric	Dimension	Description
`AvailableReservedInstances`	`InstanceType` and `OutpostId`	The number of instances that are available for launch into the compute capacity reserved using Capacity Reservations. This metric does not include Amazon EC2 Reserved Instances. This metric does not include the number of instances that you can reserve. To determine how many instances you can reserve, subtract the `AvailableReservedInstances` count from the `AvailableInstanceType_Count` count. `Number of instances that you can reserve = AvailableInstanceType_Count - AvailableReservedInstances` **Unit**: Count **Maximum resolution**: 5 minutes

Metric	Dimension	Description
`UsedReservedInstances`	`InstanceType` and `OutpostId`	The number of instances that are running in the compute capacity reserved using Capacity Reservations. This metric does not include Amazon EC2 Reserved Instances. **Unit**: Count **Maximum resolution**: 5 minutes
`TotalReservedInstances`	`InstanceType` and `OutpostId`	The total number of instances , running and available for launch, provided by the compute capacity reserved using Capacity Reservations. This metric does not include Amazon EC2 Reserved Instances. **Unit**: Count **Maximum resolution**: 5 minutes

Amazon EBS metrics

The following metrics are available for the EBS volume type capacity.

Metric	Dimension	Description
`EBSVolumeTypeCapacityUtilization`	`VolumeType` and `OutpostId`	The percentage of EBS volume type capacity in use.

Metric	Dimension	Description
		Unit: Percent **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Average` and `pNN.NN` (percentiles).
`EBSVolumeTypeCapacityAvailability`	`VolumeType` and `OutpostId`	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).
`EBSVolumeTypeCapacityUtilizationGB`	`VolumeType` and `OutpostId`	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).

Metric	Dimension	Description
EBSVolumeTypeCapacityAvailabilityGB	VolumeType and OutpostId	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).

Virtual interface metrics

The following metrics are available for the virtual interface (VIF).

Metric	Dimension	Description
VifBgpSessionState	**Dimensions for local gateway VIFs**: OutpostsId , VirtualInterfaceGroupId , VirtualInterfaceId . **Dimensions for service link VIFs**: OutpostsId , VirtualInterfaceId .	The Border Gateway Protocol (BGP) session state between the AWS Outposts of virtual interface (VIF) and on-premise devices. **Unit**: Values 1 through 6 where: • 1 – Idle. This is the initial state where the Outposts rack is waiting for a start event. • 2 – Connect. The Outposts rack is waiting for the TCP connection to be completed .

Metric	Dimension	Description
		• 3 – Active. The Outposts rack is trying to initiate a TCP connection. • 4 – OpenSent. The router has sent an OPEN message and is waiting for one in return. • 5 – OpenConfirm. The router has received an OPEN message and is waiting for a KEEPALIVE message. • 6 – Established. The BGP connection is fully establish ed and the Outposts rack and on-premise devices can exchange routing informati on. **Maximum resolution**: 5 minute **Statistics**: The most useful statistic is `Maximum`.

Metric	Dimension	Description
`VifConnectionStatus`	**Dimensions for local gateway VIFs**: `OutpostsId` , `VirtualInterfaceGroupId` , `VirtualInterfaceId` . **Dimensions for service link VIFs**: `OutpostsId` , `VirtualInterfaceId` .	Shows whether the virtual interfaces (VIFs) are ready to forward traffic. **Unit**: 1 or 0 where: • 1 – Indicates that the Outpost VIF is successfully connected to on-premise devices, configured, and ready to forward traffic. • `0` – Indicates that the Outpost VIF is not ready to forward traffic. **Maximum resolution**: 5 minute **Statistics**: The most useful statistic is `Maximum`.
`IfTrafficIn`	**Dimensions for local gateway VIFs (lgw-vif)**: `OutpostsId` , `VirtualInterfaceGroupId` , and `VirtualInterfaceId` **Dimensions for service link VIFs (sl-vif)**: `OutpostsId` and `VirtualInterfaceId`	The bitrate of data that the Outposts Virtual Interfaces (VIFs) receive from the connected local network devices. **Unit**: Bits per second **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Max` and `Min`.

Metric	Dimension	Description
`IfTrafficOut`	**Dimensions for local gateway VIFs (lgw-vif)**: `OutpostsId` , `VirtualInterfaceGroupId` , and `VirtualInterfaceId` **Dimensions for service link VIFs (sl-vif)**: `OutpostsId` and `VirtualInterfaceId`	The bitrate of data that the Outposts Virtual Interfaces (VIFs) transfer to the connected local network devices. **Unit**: Bits per second **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Max` and `Min`.

Outposts metrics

The following metrics are available for your Outposts.

Metric	Dimension	Description
`ConnectedStatus`	`OutpostId`	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`.
`CapacityExceptions`	`InstanceType` and `OutpostId`	The number of insufficient capacity errors for instance launches.

Metric	Dimension	Description
		Unit: Count **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Maximum` and `Minimum`.

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

View CloudWatch metrics for your Outposts rack

You can view the CloudWatch metrics for your Outposts rack 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 field.

To view metrics using the AWS CLI

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

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

To get the statistics for a metric using the AWS CLI

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

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

Log AWS Outposts API calls using AWS CloudTrail

AWS Outposts is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service. CloudTrail captures 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. 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, when it was made, and additional details.

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

- Whether the request was made with root user or user credentials.
- Whether the request was made on behalf of an IAM Identity Center user.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

CloudTrail is active in your AWS account when you create the account, and you automatically have access to the CloudTrail **Event history**. The CloudTrail **Event history** provides a viewable, searchable, downloadable, and immutable record of the past 90 days of recorded management events in an AWS Region. For more information, see Working with CloudTrail Event history in the *AWS CloudTrail User Guide*. There are no CloudTrail charges for viewing the **Event history**.

For an ongoing record of events in your AWS account past 90 days, create a CloudTrail Lake event data store.

CloudTrail Lake event data stores

> *CloudTrail Lake* lets you run SQL-based queries on your events. CloudTrail Lake converts existing events in row-based JSON format to Apache ORC format. ORC is a columnar storage format that is optimized for fast retrieval of data. Events are aggregated into *event data stores*, which are immutable collections of events based on criteria that you select by applying advanced event selectors. The selectors that you apply to an event data store control which events persist and are available for you to query. For more information about CloudTrail Lake, see Working with AWS CloudTrail Lake in the *AWS CloudTrail User Guide*.
>
> CloudTrail Lake event data stores and queries incur costs. When you create an event data store, you choose the pricing option you want to use for the event data store. The pricing option determines the cost for ingesting and storing events, and the default and maximum retention period for the event data store. For more information about CloudTrail pricing, see AWS CloudTrail Pricing.

AWS Outposts management events in CloudTrail

Management events provide information about management operations that are performed on resources in your AWS account. These are also known as control plane operations. By default, CloudTrail logs management events.

AWS Outposts logs all AWS Outposts control plane operations as management events. For a list of the AWS Outposts control plane operations that AWS Outposts logs to CloudTrail, see the AWS Outposts API Reference.

AWS Outposts event examples

The following example shows a CloudTrail event that demonstrates the `SetSiteAddress` operation.

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AKIAIOSFODNN7EXAMPLE:jdoe",
        "arn": "arn:aws:sts::111122223333:assumed-role/example/jdoe",
        "accountId": "111122223333",
        "accessKeyId": "AKIAI44QH8DHBEXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AKIAIOSFODNN7EXAMPLE",
                "arn": "arn:aws:iam::111122223333:role/example",
                "accountId": "111122223333",
                "userName": "example"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-08-14T16:28:16Z"
            }
        }
    },
    "eventTime": "2020-08-14T16:32:23Z",
    "eventSource": "outposts.amazonaws.com",
    "eventName": "SetSiteAddress",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "XXX.XXX.XXX.XXX",
    "userAgent": "userAgent",
    "requestParameters": {
        "SiteId": "os-123ab4c56789de01f",
        "Address": "***"
    },
    "responseElements": {
        "Address": "***",
        "SiteId": "os-123ab4c56789de01f"
    },
    "requestID": "1abcd23e-f4gh-567j-klm8-9np01q234r56",
    "eventID": "1234a56b-c78d-9e0f-g1h2-34jk56m7n890",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Outposts rack 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 Outposts rack and determines whether any maintenance is required.

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

Contents

Update contact details

If the Outpost owner changes, contact AWS Support Center with the new owner's name and contact information.

Hardware maintenance

If AWS detects an irreparable issue with hardware during the server-provisioning process or while hosting Amazon EC2 instances running on your Outposts rack, we will notify the owner of the instances that the affected instances are scheduled for retirement. For more information, see Instance retirement in the *Amazon EC2 User Guide*.

The Outpost owner and instance owner can work together to resolve the issue. The instance owner can stop and start an affected instance to migrate it to available capacity. Instance owners can stop and start the affected instances at a time that is convenient for them. Otherwise, AWS stops and starts the affected instances on the instance retirement date. If there is no additional capacity on the Outpost, the instance remains in the stopped state. The Outpost owner can try to free up used capacity or request additional capacity for the Outpost so that the migration can complete.

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

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

Firmware updates

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

Network equipment maintenance

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

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

The customer network should advertise equal BGP prefixes with the same attributes to all ONDs. The Outpost network load balances outbound traffic between all uplinks by default. Routing policies are used on the Outpost side to shift traffic away from an OND if maintenance is required. This traffic shift requires equal BGP prefixes from the customer side on all ONDs. If maintenance is required on the customer network, we recommend that you use AS-Path prepending to temporarily shift traffic array from specific uplinks.

Best practices for power and network events

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

Power events

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

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

- After power is restored, create a case at AWS Support Center to request verification that AWS Outposts and the related services are running.

Network connectivity events

The service link connection between your Outpost and the AWS Region or Outposts home Region will typically automatically recover from network interruptions or issues that may occur in your upstream corporate network devices or in the network of any third party connectivity provider once the network maintenance is completed. During the time the service link connection is down, your Outposts operations are limited to local network activities.

Amazon EC2 instances, Local gateway, and Amazon EBS volumes on the Outposts will continue to operate normally and can be accessed locally through the local network. Similarly, AWS service resources such as Amazon ECS worker nodes continue to run locally. However, API availability will be degraded. For example, the run, start, stop, and terminate APIs might not work. Instance metrics and logs will continue to be cached locally for up to 7 days, and will be pushed to the AWS Region when connectivity returns. Disconnection beyond 7 days might result in loss of metrics and logs.

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

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

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

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

  Capture a trace-route between:

  - The public IP addresses at the Outposts location and the IP address returned by the `outposts.`*`region`*`.amazonaws.com`. Replace *`region`* with the name of the parent AWS Region.

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

Resources

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

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

Outposts rack end-of-term options

At the end of your AWS Outposts term, you must choose between the following options:

- Renew your subscription and keep your existing Outposts racks.
- Prepare your Outposts racks for return.
- Convert to a month-to-month subscription and keep your existing Outposts racks.

Renew your subscription

You must complete the following steps at least **5 business days** before the current subscription for your Outposts racks ends. Failing to complete these steps at least 5 business days before the current subscription ends might result in unanticipated charges.

To renew your subscription and keep your existing Outposts racks:

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. In the navigation pane, choose **Outposts**.
3. Choose **Actions**.
4. Choose **Renew Outpost**.
5. Choose the subscription term length and payment option.

   For pricing, see AWS Outposts rack pricing. You can also request a price quote.
6. Choose **Submit support ticket**.

> **Note**
>
> If renewing before the current subscription for your Outposts racks ends, you will be charged immediately for any upfront fees.

Your new subscription will start the day after your current subscription ends.

If you do not indicate that you want to renew your subscription or return your Outposts rack, you will be converted to a month-to-month subscription automatically. Your Outposts rack will be renewed on a monthly basis at the rate of the **No Upfront** payment option that corresponds to your AWS Outposts configuration. Your new monthly subscription will start the day after your current subscription ends.

Return AWS Outposts racks

You must prepare your AWS Outposts rack for return and complete the decommission process at least **5 business days** before the current subscription for your Outposts rack ends. AWS can't start the return process until you do so. Failing to complete these steps at least 5 business days before the current subscription ends might result in delays in decommissioning and unanticipated charges.

You will not be charged a shipping fee when you return an Outposts rack. However, if you return a rack that is damaged, you might incur a cost.

To prepare your AWS Outposts rack for return:

> **Important**
>
> Do not power down the Outposts rack until AWS is on-site for the scheduled retrieval.

1. If the Outpost's resources are shared, you must unshare these resources.

   You can unshare a shared Outpost resource in one of the following ways:

   - Use the AWS RAM console. For more information, see Updating a resource share in the *AWS RAM User Guide*.
   - Use the AWS CLI to run the disassociate-resource-share command.

   For the list of Outpost resources that can be shared, see Shareable Outpost resources.
2. Terminate the active instances associated with subnets on your Outpost. To terminate the instances, follow the instructions in Terminate your instance in the *Amazon EC2 User Guide*.

   > **Note**
   >
   > Some AWS-managed services running on your Outpost, such as Application Load Balancers or Amazon Relational Database Service (RDS), consume EC2 capacity. However, their associated instances aren't visible on the Amazon EC2 dashboard.

You must terminate the resources tied to these services to free up capacity. For more information, see Why is some EC2 instance capacity missing on my Outpost?.

3. Verify the instance-capacity-availability of your Amazon EC2 instances in your AWS account.

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

    b. Choose **Outposts**.

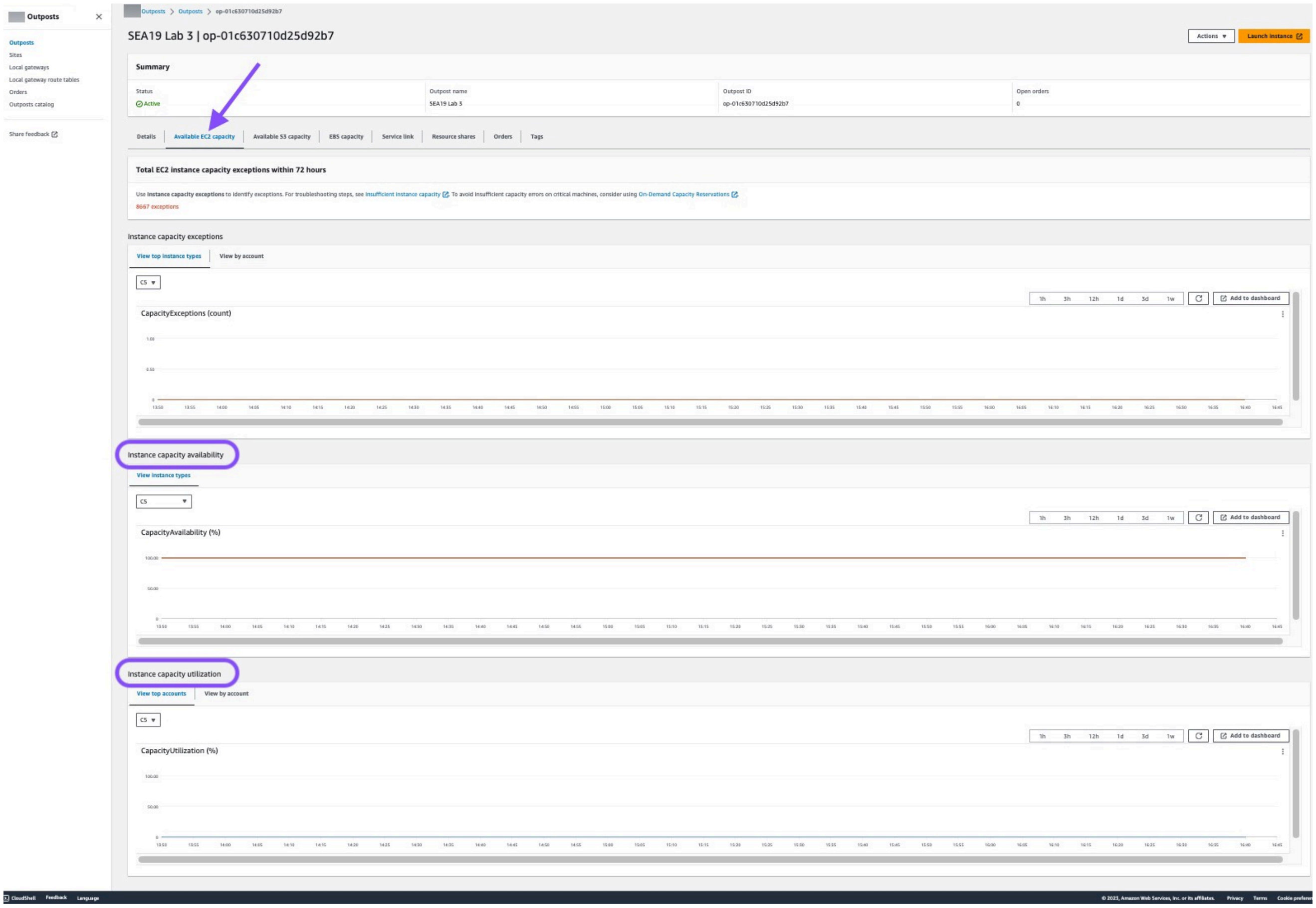
    c. Choose the specific Outpost you are returning.

    d. On the page for the Outpost, choose the **Available EC2 capacity** tab.

    e. Ensure that the **Instance capacity availability** is at 100% for each instance family.

    f. Ensure that the **Instance capacity utilization** is at 0% for each instance family.

    The following image shows the **Instance capacity availability** and **Instance capacity utilization** graphs on the **Available EC2 capacity** tab.



Outposts > Outposts > op-01c630710d25d92b7

SEA19 Lab 3 | op-01c630710d25d92b7

Summary

Status: Active

Outpost name: SEA19 Lab 3

Outpost ID: op-01c630710d25d92b7

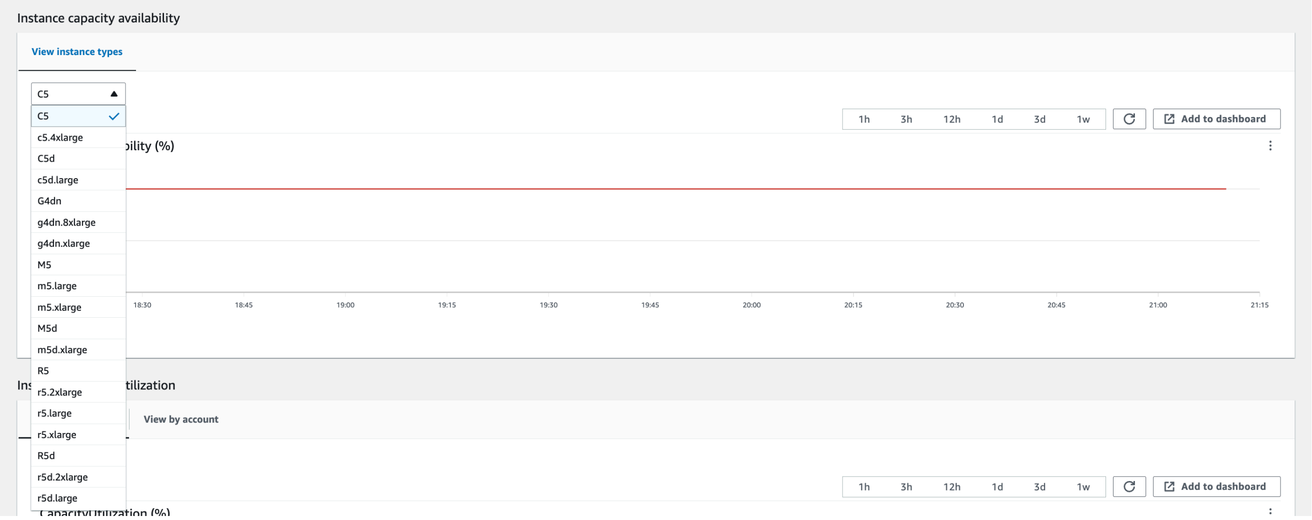
Open orders: 0

Total EC2 instance capacity exceptions within 72 hours

Use **Instance capacity exceptions** to identify exceptions. For troubleshooting steps, see Insufficient instance capacity. To avoid insufficient capacity errors on critical machines, consider using On-Demand Capacity Reservations.

8667 exceptions

The following image shows the list of instance types.



4. Create backups of your Amazon EC2 instances and server volumes. To create the backups, follow the instructions in Backup and recovery for Amazon EC2 with EBS volumes in the *AWS Prescriptive Guidance* guide.

5. Delete the Amazon EBS volumes associated with your Outpost.

   a. Open the Amazon EC2 console console at https://console.aws.amazon.com/ec2/.

   b. From the navigation pane, choose **Volumes**.

   c. Choose **Actions** and **Delete volume**.

   d. In the confirmation dialog box, choose **Delete**.

6. Delete any VPC associations and customer-owned IP address pool (CoIP) CIDRs associated with your Outpost.

An AWS retrieval team will power down the rack. After it's powered down, you can destroy the AWS Nitro Security Key or the AWS retrieval team can do so on your behalf.

To return your AWS Outposts racks

> ⚠ **Important**
>
> AWS can't stop the return process after you have submitted your decommission request.

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

2. In the navigation pane, choose **Outposts**.

3. Choose **Actions**.

4. Choose **Decommission Outpost** and follow the workflow to delete resources.
5. Choose **Submit request**.

   An AWS representative will contact you to begin the decommissioning process.

> ⓘ **Note**
>
> Returning your racks before the current subscription for your Outposts racks ends will not terminate any outstanding charges associated with this Outpost.

An AWS retrieval team will power down the rack. After it's powered down, you can destroy the AWS Nitro Security Key or the AWS retrieval team can do so on your behalf.

Convert to a month-to-month subscription

To convert to a month-to-month subscription and keep your existing Outposts racks, no action is needed. If you have questions, open a billing support case.

Your Outposts racks will be renewed on a monthly basis at the rate of the **No Upfront** payment option that corresponds to your Outposts configuration. Your new monthly subscription starts the day after your current subscription ends.

Quotas for AWS Outposts

Your AWS account has default quotas, formerly referred to as limits, for each AWS service. Unless otherwise noted, each quota is Region-specific. You can request increases for some quotas, but not for all quotas.

To view the quotas for AWS Outposts, open the Service Quotas console. In the navigation pane, choose **AWS services**, and select **AWS Outposts**.

To request a quota increase, see Requesting a quota increase in the *Service Quotas User Guide*.

Your AWS account has the following quotas related to AWS Outposts.

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

AWS Outposts and the quotas for other services

AWS Outposts relies on the resources of other services and those services may have their own default quotas. For example, your quota for local network interfaces comes from the Amazon VPC quota for network interfaces.

Document history for Outposts racks

The following table describes the documentation updates for Outposts racks.

Change	Description	Date
AWS Outposts supports external block volumes from Dell and HPE storage arrays	You can use external block data and boot volumes backed by third-party vendors such as, Dell PowerStore and HPE Alletra Storage MP B10000.	September 30, 2025
Support for AWS services	AWS Outposts supports AWS services based on the AWS Region in which your Outpost operates.	September 18, 2025
Metrics available for VIF connection status and BGP session state.	You can monitor the status of your AWS Outposts VIF connection and BGP session on the CloudWatch console with the **ConnectionStatus** and the **BGPSessionState** metrics.	July 31, 2025
Renewing your subscription and preparing racks for return	To renew a subscription or return a rack, you must complete the process at least 10 business days before the current subscription ends.	July 16, 2025
Support for gp3 volume type	Second-generation Outpost racks now supports Amazon EBS gp3 volumes.	June 30, 2025
Virtual interfaces (VIFs) and VIF Groups	Local gateway VIFs (Virtual Interface) is a logical interface	May 5, 2025

	component of Outposts racks that sets up VLAN, IP, and BGP connectivity between an Outposts networking device and an on-premise networking device for local gateway connectivity. You must create local gateway VIFs and VIF groups.	
Updates to static stability	In the event that your network is interrupted, instance metrics and logs will be cached locally for up to 7 days.	May 1, 2025
Initial release	This is the initial release of AWS Outposts racks second generation for accelerated networking.	April 29, 2025