AWS Outposts

User Guide

AWS Outposts: User Guide

Table of Contents

What is AWS Outposts?

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

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

For more information, see the AWS Outposts product page.

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 configurations** – Mixes of Amazon EC2 compute capacity, Amazon EBS storage capacity, and networking support. Each configuration has unique power, cooling, and weight support requirements.
- **Outpost capacity** – Compute and storage resources available on the Outpost. You can view and manage the capacity for your Outpost from the AWS Outposts console.
- **Outpost equipment** – Physical hardware that provides access to the AWS Outposts service, including racks, servers, switches, and cabling owned and managed by AWS.
- **Service link** – Network route that enables communication between your Outpost and its associated AWS Region. Each Outpost is an extension of an Availability Zone and its associated Region.
- **Local gateway** – A logical interconnect virtual router that enables communication between your Outpost and your on-premises network.

AWS resources on Outposts

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

- Amazon EC2 instances and EBS volumes – Launch an instance on your Outpost (p. 22)
- Amazon ECS clusters – Amazon Elastic Container Service on AWS Outposts
- Amazon EKS nodes – Amazon Elastic Kubernetes Service on AWS Outposts
- Amazon EMR clusters – EMR Clusters on AWS Outposts
- Amazon RDS DB instances – Amazon RDS on AWS Outposts
- Amazon S3 buckets – Using Amazon S3 on AWS Outposts
- Application Load Balancers – Subnets for your load balancer
- AWS App Mesh Envoy proxy – AWS App Mesh on AWS Outposts

Pricing

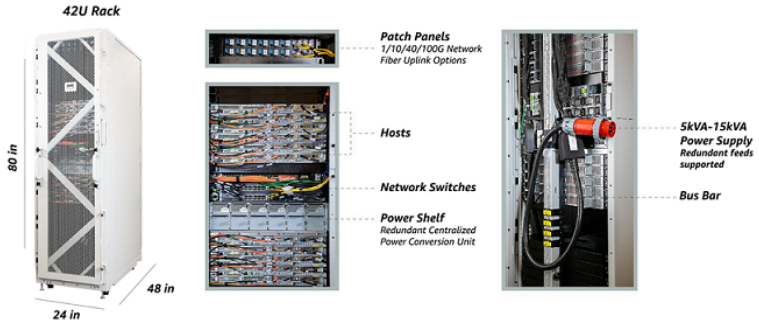
You can choose from a variety of Outpost configurations, each providing a combination of EC2 instance types and EBS volumes. The pricing for these configurations includes the EC2 instances and EBS volumes, plus delivery, installation, and maintenance of the Outpost equipment. You can also increase your compute and storage capacity over time by upgrading your configuration.

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

How AWS Outposts works

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

The following image shows an Outpost rack. The 42U rack is 80 inches high by 24 inches wide by 48 inches deep.



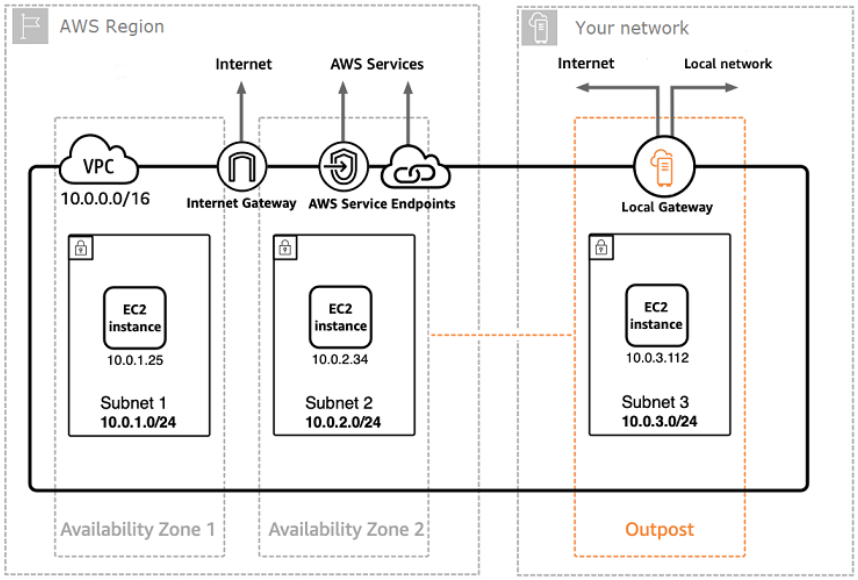
Contents

AWS Outposts network components

An 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.

- VPCs and subnets
- A local gateway
- A customer-owned IP address pool
- Route tables



Contents

VPCs and subnets

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

Outposts support multiple subnets. You can specify the EC2 instance subnet when you launch the EC2 instance in your Outpost. You cannot restrict the hardware server 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. AWS does not directly advertise the VPC CIDR, or the Outpost subnet range to your on-premises location.

Local Gateway

A local gateway serves two purposes. It provides a target in your VPC route tables for on-premises destined traffic, and performs network address translation (NAT) for instances that have been assigned addresses from your customer-owned IP pool.. You can also use the local gateway for communication for internet-bound traffic . Each Outpost supports a single local gateway. You can associate multiple VPCs with the local gateway. For more information, see Working with local gateways (p. 29) and Outpost connectivity to the local network (p. 9).

You can attach the local gateway to a VPC to connect to your on-premises network. The local gateway provides connectivity between your local network, or your local gateway VLAN, and the VPC. The local gateway performs NAT of the Outpost instances' IP addresses to Elastic IP addresses from a pool that is assigned to the local gateway. The local gateway NAT function is similar to how an internet gateway functions in an AWS Region.

The local gateway for your Outpost enables connectivity from your Outpost subnets to all AWS services that are available in the parent Region, in the same way that you access them from an Availability Zone subnet. For example, you can access the Regional service endpoints over the public internet, or you can use interface VPC endpoints (AWS PrivateLink) to access them without going over the public internet. For more information, see Outpost connectivity to AWS Regions (p. 15).

Customer-owned IP addresses

During the installation process, AWS uses information that you provide about your on-premises network to create an address pool, known as a *customer-owned IP address pool* (CoIP pool). AWS then assigns it to the local gateway for use and advertisement back to your customer network through BGP. Customer-owned IP addresses provide local or external connectivity to resources in your Outpost subnets through 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 from the customer-owned IP pool, and then assigning this new Elastic IP to your EC2 instance. The following requirements apply to the customer-owned IP address pool:

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

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

You can optionally share your customer-owned pool with multiple AWS accounts in your AWS Organizations using the AWS Resource Access Manager. After you share the pool, participants can allocate and associate Elastic IPs from the customer owned IP pool. For more information see, the section called “Step 3: Allocate and associate an Elastic IP address with the instance” (p. 23). For information about how to share a customer-owned IPv4 addresses, see Sharing Your Resources in the *AWS RAM User Guide*. You use the customer-owned pool after you launch the Outpost instance.

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. You can include a local gateway as target when the destination is your on-premises network. A local gateway can only be used in VPC and subnet route tables that are associated with an Outpost.

The route tables for Outpost subnets work as they do for Availability Zone subnets. You can specify IP addresses, internet gateways, local gateways, virtual private gateways, and peering connections as destinations. For example, each Outpost subnet, either through the inherited main route table, or a custom table, inherits the VPC local route. This means that all traffic in the VPC, including the Outpost subnet with a destination in the VPC CIDR remains routed in the VPC. You cannot configure a more specific range than the VPC CIDR local route on the Outpost for Outpost subnets.

Outpost subnet route tables can include the following destinations:

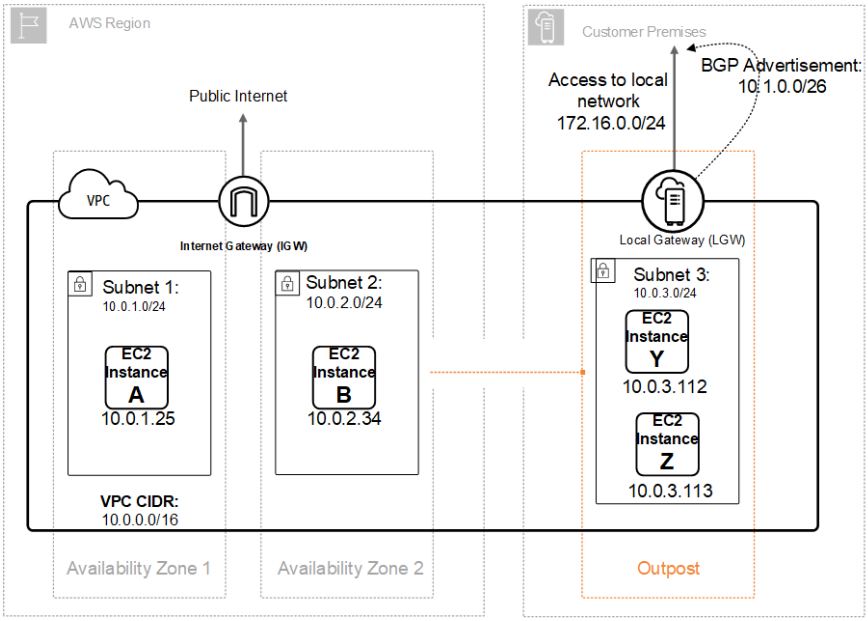
- **VPC CIDR range** – AWS defines this at installation. This is the local route and applies to all VPC routing, including traffic between Outpost instances in the same VPC.

- **Customer on-premises network** – The local gateway routes this traffic for low latency routing to the on-premises network.
- **AWS Region destinations** – This includes prefix lists for Amazon Simple Storage Service (Amazon S3), Amazon DynamoDB gateway endpoint, AWS Transit Gateways, virtual private gateways, internet gateways, and VPC peering.

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

Consider a scenario with the following configuration:

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



You need the following entries in the Outpost subnet route table.

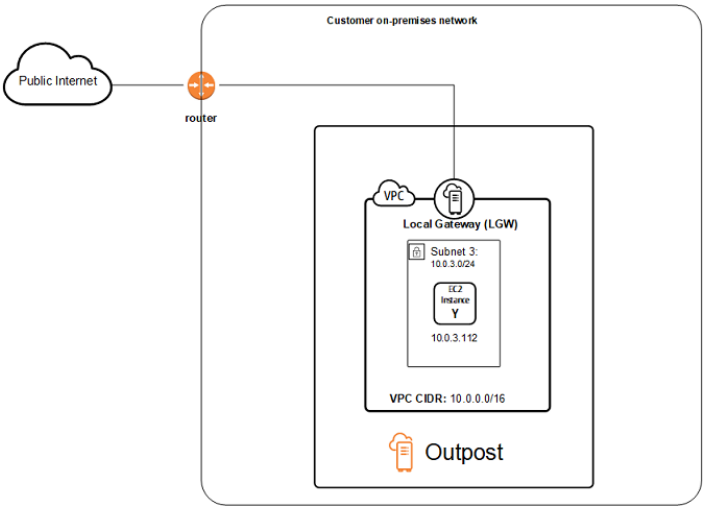
Destination	Target	Type	Notes
10.0.0.0/16	Local	Defined by AWS	The local VPC route. This route allows for intra-VPC connectivity,

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

Example: Local gateway routing

Consider a scenario with the following configuration:

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



You need the following entries in the Outpost subnet route table.

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

Local gateway access to the internet

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

Traffic initiated from the EC2 instance for the internet uses the 0.0.0.0/0 route to route traffic to the local gateway. The local gateway maps the EC2 instance Elastic IP address to the customer-owned IP address (10.1.0.2), and then sends the traffic to the customer router. The router uses NAT to translate the customer-owned IP address to a public IP address on the router, and then sends the traffic to the destination. If you do not want the instance to natively connect to the internet, create a NAT instance.

Outbound instance traffic to the on-premises network

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

Inbound traffic from the on-premises network to the instance

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

DNS

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

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

Outpost connectivity to the local network

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

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

For more information about AWS Outposts hardware and connectivity information, see AWS Outposts Now Available – Order Yours Today.

Contents

Physical connectivity

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

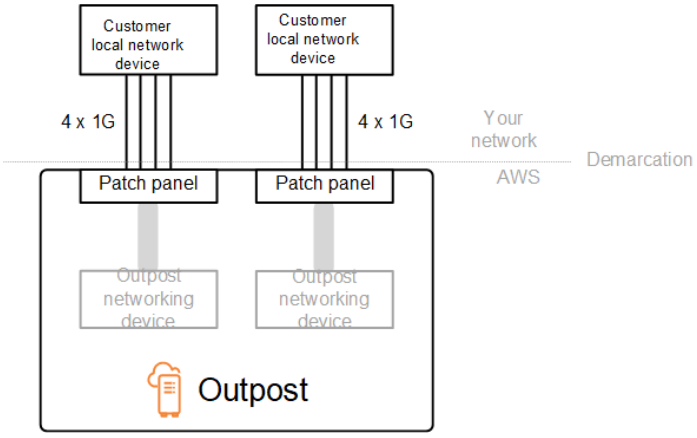
An Outpost requires a minimum of two physical links between these Outpost network devices and your local network devices. An Outpost supports the following uplink speeds and quantities for each Outpost network device.

Uplink speed	Number of uplinks
1 Gbps	1, 2, 3, 4, 5, 6, 7, 8
10 Gbps	1, 2, 3, 4, 8, 12, 16
40 Gbps	1, 2, 3, 4
100 Gbps	1, 2, 3, 4

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

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

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



Link aggregation

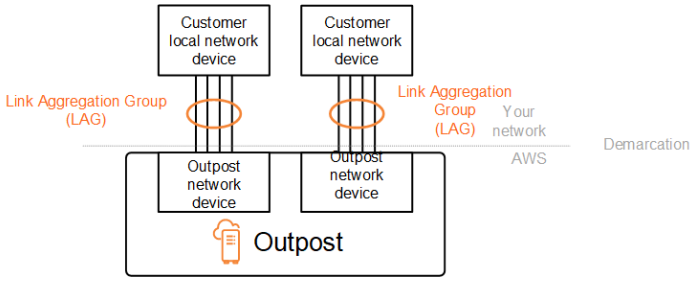
AWS Outposts uses the Link Aggregation Control Protocol (LACP) to establish two link aggregation group (LAG) connections to your Outpost network devices and to your local network devices. The links from each Outpost network device are aggregated into an Ethernet LAG to represent a single network connection. These LAGs use LACP with standard fast timers.

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

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

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

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



Virtual LANs

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

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

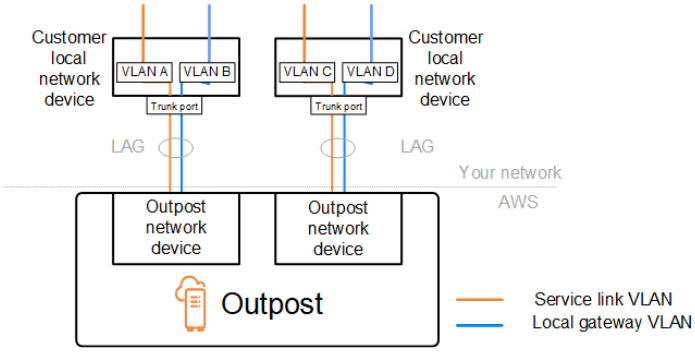
- **Service link VLAN** – Enables communication between the Outpost and the AWS Region for both management of the Outpost and intra-VPC between the AWS Region and Outpost. This VLAN

provides access to the AWS Region, which enables the service link connection from the Outpost to be established back to the Region. The service link is a custom VPN or VPNs from the Outpost to the Region. It is connected to the Outpost that is configured in the Availability Zone when your purchase the Outpost.

- **Local gateway VLAN** – Enables VPC traffic from your VPC to your local LAN. This VLAN enables instances running on the Outpost to communicate with your on-premises network. It also enables them to communicate with the internet through your on-premises network.

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

An Outpost is designed to segregate 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.

Network layer connectivity

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

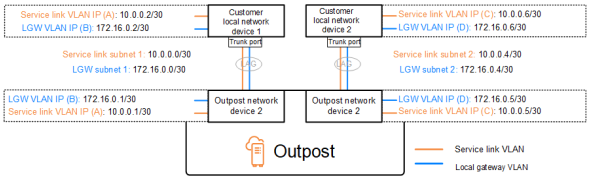
You need to establish two paths:

- **Service link path** - To establish this path, specify a VLAN subnet with a range of /30 or /31 and an IP address for the service link VLAN on the Outpost network device.
- **Local gateway path** - To establish this path, specify a VLAN subnet with a range of /30 or /31 and an IP address for the local gateway VLAN on the Outpost network device.

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

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

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



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

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

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

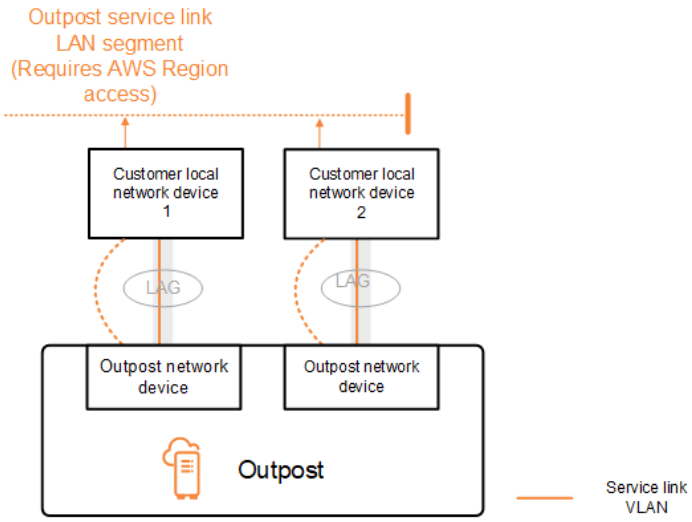
VLAN	Subnet	Customer Device 2 IP	AWS OND 2 IP
C	172.16.0.4/30	172.16.0.6	172.16.0.5
D	10.0.0.4/30	10.0.0.6	10.0.0.5

Service link BGP connectivity

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

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

- The service link BGP ASN. The valid values are 64512-65535.
- The infrastructure CIDR. This must be a /26 CIDR.
- The customer local network device 1 service link BGP peer IP address.
- The customer local network device 1 service link BGP peer ASN. The valid values are 0-65535.
- The customer local network device 2 service link BGP peer IP address.
- The customer local network device 2 service link BGP peer ASN. The valid values are 0-65535. For more information, see RFC4893.



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

1. Each Outpost network device uses the ASN to establish a BGP peering session with its connected local network device.
2. Outpost network devices advertise the /26 CIDR range as two /27 CIDR blocks to support link and device failures.
3. The subnet is used for connectivity from the Outpost to the AWS Region.

Service link infrastructure subnet advertisement and IP range

The service link infrastructure subnet is a /26 CIDR range that you provide during the pre-installation process. The service link range is used by the Outpost infrastructure to establish connectivity to the Region through the service link. The service link subnet is the Outpost source, which initiates the connectivity.

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

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

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

Local gateway BGP connectivity

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

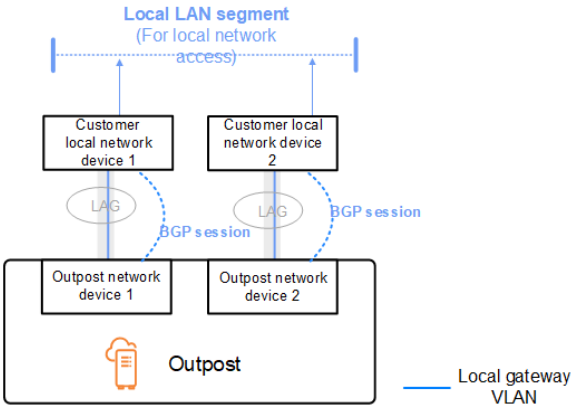
The Outpost establishes an external BGP peering session over the local gateway VLAN between each Outpost network device and it's connected customer local network device. The peering session is established between the /30 or /31 IPs that you provided when you set up network connectivity and

uses point-to-point connectivity between the Outpost network devices and customer local network devices. For more information, see the section called “Network layer connectivity” (p. 11).

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

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

- AWS provides the local gateway BGP ASN. The valid values are 64512-65535.
- You provide the customer owned CIDR that gets advertised.
- You provide the customer local network device 1 local gateway BGP peer IP address.
- You provide the customer local network device 1 local gateway BGP peer ASN. The valid values are 0-65535. For more information, see RFC4893.
- You provide the customer local network device 2 local gateway BGP peer IP address.
- You provide the customer local network device 2 local gateway BGP peer ASN. The valid values are 0-65535. For more information, see RFC4893.



Local gateway customer-owned IP subnet advertisement

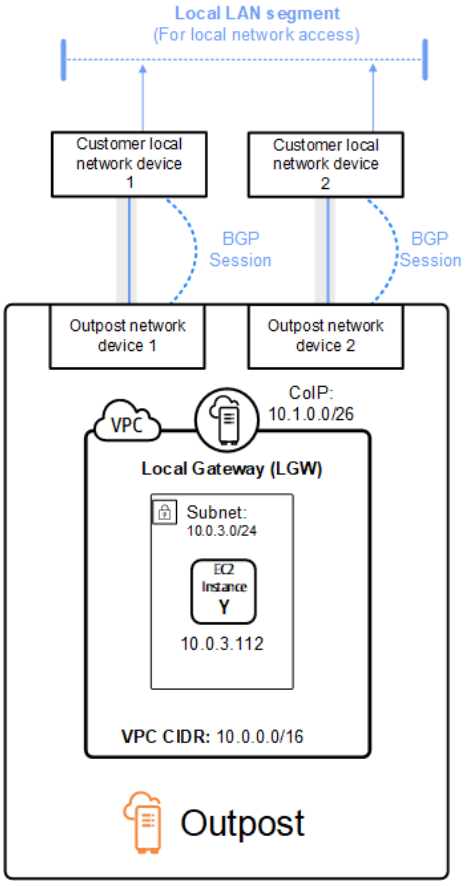
During the pre-installation process, AWS creates an address pool, known as a customer-owned IP address pool. It is created based on information that you provide about your on-premises network. You can create Elastic IP addresses from this pool, and then assign the addresses to resources on your Outpost, 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.

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

- A VPC with a CIDR block 10.0.0.0/16.
- A subnet in the VPC with a CIDR block 10.0.3.0/24.
- An EC2 instance in the subnet with a private IP address 10.0.3.112.

- A customer-owned IP pool (10.1.0.0/26).
- An Elastic IP address association that associates 10.0.3.112 to 10.1.0.2.
- A local gateway that uses BGP to advertise 10.1.0.0/26 to the on-premises network through the local devices.
- Communication between your Outpost and on-premises network will use the COIP Elastic IPs to address instances in the Outpost, the VPC CIDR range is not used.



Outpost connectivity to AWS Regions

AWS Outposts supports two types of wide area network (WAN) connectivity: service links and local gateways.

Contents

Connectivity through service links

When an Outpost is provisioned, it initiates the service link VPN back to the AWS Region, and builds the service link VPN connection. The Outpost must be able to reach the public AWS ranges, either through

the public internet or AWS Direct Connect public virtual interface. This connectivity can be through specific routes in the service link VLAN, or through a default route of 0.0.0.0/0. For more information about the public ranges for AWS, see AWS IP Address Ranges.

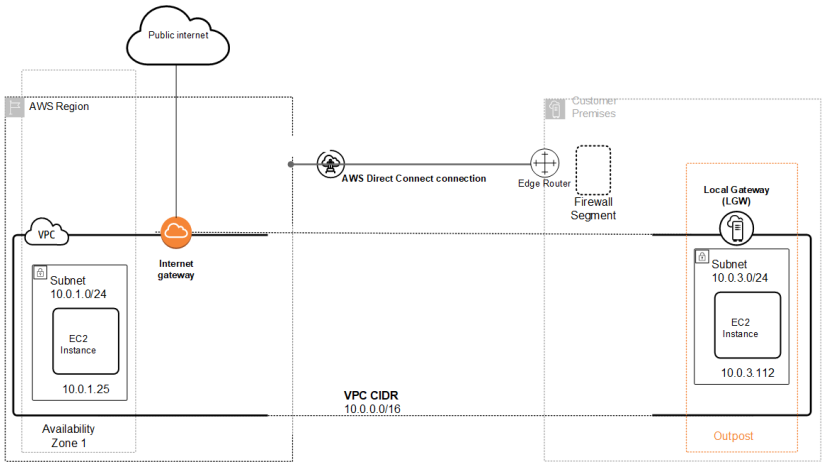
After the service link is established, the Outpost is in service and managed by AWS. The service link is used for the following traffic:

- Management traffic to the Outpost through the service link
- Traffic between the Outpost and any associated VPCs

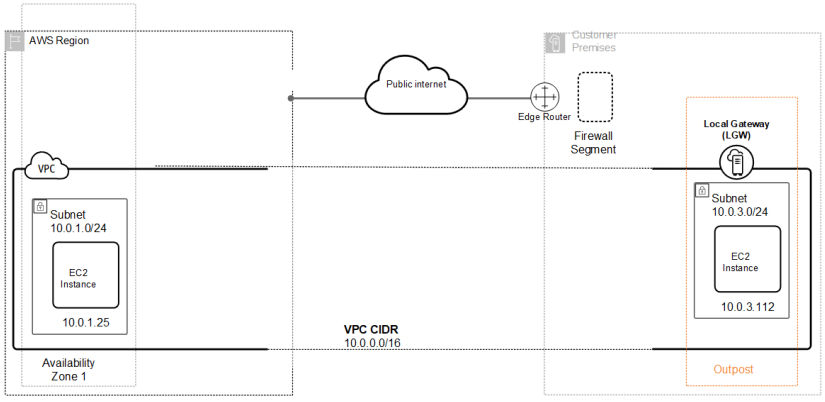
Outpost service links support an MTU of 1300 bytes. You can use AWS Direct Connect, or an internet connection to connect to the AWS Region. For an optimal experience and resiliency, AWS recommends that you use dual 1Gbps connections to the AWS Region.

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

- Control plane
- Intra-VPC traffic between the Outpost and the AWS Region



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



If you use a firewall to limit the connectivity from the service link VLAN, you can block all inbound and outbound connections. This is because the service link VPN initiates from the Outpost to the AWS Region. You must configure the following outbound rules on any firewall between the AWS Region and the service link VLAN. If the firewall is stateful, outbound connections from the Outpost that are allowed, meaning that they were initiated from the Outpost, should be allowed back inbound. If the firewall is stateless, you must allow connections back inbound.

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

Note
An Outpost VPC cannot use the service link to communicate with or within the same VPC extended in other Outposts. Use the local gateways to communicate between VPCs across Outposts. Outpost racks are also designed with redundant power and networking equipment, including local gateway components. For more information, see Resilience in AWS Outposts (p. 41).

Connectivity through the local gateway

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

The data plane path for the local gateway traverses from the Outpost, through the local gateway, and to your private local gateway LAN segment. It would then follow a private path back to the AWS service endpoints in the Region.

Redundant internet connections

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

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

Outpost site requirements

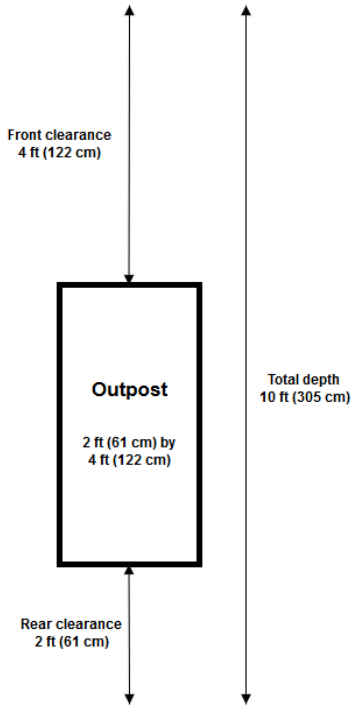
An Outpost site is the physical location where AWS will install your Outpost. Before you order an Outpost, verify that your site meets the following requirements.

Requirements

Facility

- **Temperature and humidity** – The ambient temperature must be between 41° F (5° C) and 104° F (40° C). The relative humidity must be between 8 percent and 80 percent with no condensation.
- **Airflow** – 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 location where the rack will be installed and the path to that location must support the weight specified in the order summary at the rack point loads. This includes freight and standard elevators.
- **Clearance** – The equipment is 80 inches (203 cm) high by 24 inches (61 cm) wide by 48 inches (122 cm) deep. Any doorways, hallways, turns, ramps, and elevators must provide sufficient clearance. At the final resting position, there must be a 24 inch (61 cm) wide by 48 inch (122 cm) deep area for the Outpost, with an additional 48 inches (122 cm) of front clearance and 24 inches (61 cm) of rear clearance. The total minimum area required for the Outpost is 24 inch (61 cm) wide by 10 feet (305 cm) deep.

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



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

Networking

- Provide uplinks with speeds of 1, 10, 40, or 100 Gbps.
- Provide either single-mode fiber (SMF) with Lucent Connector (LC), multimode fiber (MMF), or MMF OM4 with LC.
- Provide one or two upstream devices. This can be switches or routers. We recommend two devices to provide high availability.

Power

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

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

AC line voltage	Single phase 200 to 277 VAC (50 or 60 Hz) or three phase 346 to 480 VAC (50 to 60 Hz)
Power consumption	5 kVA (4 kW), 10 kVA (9 kW), or 15 kVA (13 kW)
AC protection (upstream power breakers)	30 or 32 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 (30 A), or three-phase AH532P6W 3P+N+E 6 hour (32 A)
Whip length	10.25 ft (3 m)
Whip - Rack cabling input	From above or below the rack

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

	Redundant, single phase	**Redundant, three phase**	**Single phase**	**Three phase**
5 kVA	2 x L6-30P or IEC309, 1 drop to S1 and 1 drop to S2	2 x AH530P7W or AH532P6W, 1 drop to S1 and 1 drop to S2	1 x L6-30P or IEC309, 1 drop to S1	1 x AH530P7W or AH532P6W, 1 drop to S1
10 kVA	4 x L6-30P or IEC309, 2 drops to S1 and 2 drops to S2		2 x L6-30P or IEC309, 2 drops to S1	
15 kVA	6 x L6-30P or IEC309, 3 drops to S1 and 3 drops to S2		3 x L6-30P or IEC309, 3 drops to S1	

Get started with AWS Outposts

To begin using AWS Outposts, you must create an Outpost and order Outpost capacity. For more information about Outposts configurations, see our catalog. After your Outpost equipment is installed, the compute and storage capacity is available for you when you launch Amazon Elastic Compute Cloud (Amazon EC2) instances and create Amazon Elastic Block Store (Amazon EBS) volumes on your Outpost.

Tasks

- Create an Outpost and order Outpost capacity (p. 21)
- Launch an instance on your Outpost (p. 22)

Create an Outpost and order Outpost capacity

When you order Outpost capacity, you can choose from a variety of Outpost configurations. Each configuration provides a mix of EC2 instance types and EBS volumes.

Prerequisites

- An Outpost site is the physical location where AWS will install your Outpost equipment. Before ordering capacity, verify that your site meets the requirements for AWS Outposts. For more information, see Outpost site requirements (p. 18).
- You must have an AWS Enterprise Support plan.

To create an outpost

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. Choose **Create Outpost**.
3. Provide a name and description for your Outpost.
4. Choose **Create site**. Complete and submit the form, then select the site that you created.
5. Select an Availability Zone for your Outpost.
6. Choose **Create Outpost**.

To order outpost capacity

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. In the navigation pane, choose **Outposts catalog**, and then do the following:
    a. Select a capacity configuration for your Outpost. If the available capacity configurations do not meet your needs, you can request a custom capacity configuration instead.
    b. Choose **Place order**.
    c. Select the Outpost that you created.
    d. Choose **Place order**.
3. You can view the status of your order using the AWS Outposts console. The initial status of your order is **Order received**. An AWS representative will contact you within three business days. You will receive an email confirmation when the status of your order changes to **Order processing**. An AWS representative may contact you to get any additional information that is required by the AWS installation team.

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

4. To fulfill the order, AWS will schedule a date and time with you. You will also receive a checklist of items to verify or provide before the installation. The AWS installation team will arrive at your site at the scheduled date and time. The team will roll the rack to the identified position and your electrician can power the rack. The team will establish network connectivity for the rack over the uplink that you provide, and will configure the rack's capacity. The installation is complete when you confirm that the Amazon EC2 and Amazon EBS capacity for your Outpost is available from your AWS account.

Launch an instance on your Outpost

After your Outpost is installed and the compute and storage capacity is available for use, you can launch EC2 instances and create EBS volumes on your Outpost using an Outpost subnet.

Prerequisite

You must have an Outpost installed at your site. For more information, see Create an Outpost and order Outpost capacity (p. 21).

Tasks

Step 1: Create a subnet

You can add Outpost subnets to any VPC in the AWS Region for the Outpost. When you do so, the VPC also spans the Outpost. For more information, see AWS Outposts network components (p. 3).

> **Note**
> If you are launching an instance in an Outpost subnet that has been shared with you, skip to Step 2: Launch an instance on the Outpost (p. 22). For more information about sharing subnets, see Sharing a subnet in the *Amazon Virtual Private Cloud User Guide*.

To create an outpost subnet

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions**, **Create subnet**.
4. Select the VPC and specify an IP address range for the subnet.
5. Choose **Create**.

Step 2: Launch an instance on the Outpost

You can launch EC2 instances in the Outpost subnet that you created, or in an Outpost subnet that has been shared with you. Security groups control inbound and outbound 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.

AWS Outposts console

To launch an instance in your Outpost subnet

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions, View details**.
4. On the **Outpost summary** page, choose **Launch instance**. You are redirected to the Amazon EC2 console.
5. Follow the steps in the Amazon EC2 Launch Instance Wizard to launch the instance in your Outpost subnet. For more information, see Launching an instance using the Launch Instance Wizard.

AWS CLI

To launch an instance in your Outpost subnet

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

  Example

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

Step 3: Allocate and associate an Elastic IP address with the instance

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

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

Amazon EC2 console

To allocate and associate an Elastic IP address with the instance

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Elastic IPs**.
3. Choose **Allocate new address**.
4. For **Network Border Group**, select the location from which the IP address is advertised.
5. For **Public IPv4 address pool**, choose **Customer owned IPv4 address pool**.
6. For **Customer owned IPv4 address pool**, select the pool that you configured.
7. Choose **Allocate**, and close the confirmation screen.
8. In the navigation pane, choose **Elastic IPs**.
9. Select an Elastic IP address, and choose **Actions**, **Associate address**.
10. Select the instance from **Instance**, and then choose **Associate**.

AWS CLI

To allocate and associate an Elastic IP address with the instance

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

   Note the `PoolId` return value.

   Example

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

   Output

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

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

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

   Example

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

   Output

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

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

   Example

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

   Output

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

```
}
```

Step 4: Configure local connectivity

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

To configure routing

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

Step 5: Test the connectivity

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

- Test the connectivity from your local network to the Outpost. From a computer in your local network, run the `ping` command to the Outpost instance's customer-owned IP address (that you created in the section called “Step 2: Launch an instance on the Outpost” (p. 22)). In the following example, the COIP is 192.0.2.128.

```
ping 192.0.2.128
Pinging 192.0.2.128

Reply from 192.0.2.128:  bytes=32 time=<1ms TTL=128
Reply from 192.0.2.128:  bytes=32 time=<1ms TTL=128
Reply from 192.0.2.128:  bytes=32 time=<1ms TTL=128

Ping statistics for 192.0.2.128
Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

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

- Test the connectivity from an Outpost instance to your local network. Depending on your operating system, use **ssh** or **rdp** to connect to the private IP address of your Outpost instance. For information about connecting to a Linux instance, see Connect to your Linux instance in the *Amazon EC2 User Guide for Linux Instances*. For information about connecting to a Windows instance, see Connect to your Windows instance in the *Amazon EC2 User Guide for Windows Instances*.

After the Outpost instance is running, run the ping command to an IP address of a computer in your local network. In the following example, the IP address is 192.0.2.130.

```
ping 192.0.2.130
Pinging 192.0.2.130

Reply from 192.0.2.130:  bytes=32 time=<1ms TTL=128
Reply from 192.0.2.130:  bytes=32 time=<1ms TTL=128
Reply from 192.0.2.130:  bytes=32 time=<1ms TTL=128

Ping statistics for 192.0.2.130
Packets:  Sent = 3,  Received = 3,  Lost = 0 (0% lost)

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

- Test connectivity between the AWS Region and the Outpost. Use `run-instance` to launch an instance in the subnet in the AWS Region. For more information, see run-instances in the *AWS CLI Command Reference*.

  Example

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

  After the instance is running, perform the following operations:

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

     ```
     ping 10.0.0.5
     Pinging 10.0.0.5

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

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

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

Working with Outposts

You can manage Outposts manually to complete tasks such as adding tags and updating names and descriptions.

Manage Outpost tags

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

AWS Outposts console

To manage the Outpost tags

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, Manage tags**.
4. Add or remove a tag.

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

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

   To remove a tag, choose **Remove** to the right of the tag's Key and Value.
5. Choose **Save changes**.

Manage the Outpost name and description

You can modify the Outpost name and description.

AWS Outposts console

To manage the Outpost tags

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, Edit Outpost**.
4. Modify the name and description.

   For **Name**, enter the name.

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

View Outpost details

You can view the Outpost details using the AWS Outposts console, or the AWS CLI;

AWS Outposts console

To view the Outpost details

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**.

AWS CLI

To view the Outpost details

Use the GetOutpost AWS CLI command.

Working with local gateways

A local gateway serves two purposes. It provides a target in your VPC route tables for on-premises destined traffic, and it performs network address translation (NAT) for instances that have been assigned addresses from your customer-owned IP pool. You can also use the local gateway for communication between your Outpost and its parent AWS Region.

Each Outpost supports one local gateway. The AWS account associated with the Outpost owns the local gateway.

A local gateway has the following components:

- **Route tables** - AWS creates a local gateway for your Outpost, and a local gateway route table as part of the installation process. VPC route tables associated with subnets that reside on an Outpost can use the local gateway as a route target.
- **Virtual interfaces (VIFs)** - AWS creates one VIF for each LAG, and then associates the VIF with the default local gateway route table. The local gateway route table has a default route to the two VIFs for local network connectivity.

AWS configures a local gateway for your Outpost and a local gateway route table during the installation process. Each Outpost supports a single local gateway. The local gateway is owned by the AWS account associated with the Outpost. You can share the local gateway with other AWS accounts or organizational units using AWS Resource Access Manager.

Local gateways

Manage local gateway tags

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

AWS Outposts console

To manage the local gateway tags

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Local gateways**.
3. Select the local gateway route table, and then choose **Manage tags**.
4. Add or remove a tag.

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

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

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

Local gateway route tables

AWS creates a local gateway for your Outpost, and a local gateway route table as part of the installation process. AWS configures two VIFs, one for each of the Outpost network devices (ONDs) that are physically connected to the local network.

View local gateway route table routes

The local gateway route table is configured with a default route to each of the VIFs. After the route tables are provisioned, you can view the local gateway route tables using the AWS Outposts console, or the AWS CLI;

AWS Outposts console

To view the local gateway route table routes

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Local gateways**, and then select the local gateway.
3. Select the local gateway route table.
4. Choose **Routes** to see the route for each VIF.

AWS CLI

To view the local gateway route table routes

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

Example

```
aws ec2 describe-local-gateway-route-tables
```

Output

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

Manage local gateway route table tags

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

AWS Outposts console

To manage the local gateway route table tags

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

2. On the navigation pane, choose **Local gateway route tables**.
3. Select the local gateway route table, and then choose **Actions, Manage tags**.
4. Add or remove a tag.

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

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

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

VPC associations

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

Create a VPC association

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

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

AWS Outposts console

To associate a VPC

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Local gateway route tables**.
3. Select the route table, and then choose **Actions**, **Associate VPC**.
4. For **VPC**, select the VPC to associate with the local gateway route table.
5. (Optional) Add or remove a tag.

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

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

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

AWS CLI

To associate a VPC

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

```
aws ec2 create-local-gateway-route-table-vpc-association --local-gateway-route-table-
id local gateway route table id --vpc-id vpc id
```

Example

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

Output

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

Delete a VPC association

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

AWS Outposts console

To disassociate a VPC

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Local gateway route tables**.
3. Select the route table.
4. Select the VPC, and then choose **Disassociate**.

AWS CLI

To disassociate a VPC

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

```
aws ec2 delete-local-gateway-route-table-vpc-association --local-gateway-route-table-
id local gateway route table id --vpc-id vpc id
```

Example

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

Output

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

Working with shared AWS Outposts resources

With Outpost sharing, Outpost owners can share their Outposts and Outpost resources, including local gateway route tables, with other AWS accounts under the same AWS organization. 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 configure VPCs, launch and run instances, and create EBS volumes 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. They can also view, modify, and delete resources that consumers create on shared Outposts.

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

An Outpost owner can share Outpost resources with:

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

Contents

Shareable Outpost resources

An Outpost owner can share the following Outpost resources with consumers.

- **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.

- **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.
- **Subnets** – Consumers with access to this resource can:
  - View information about subnets.
  - Launch and run EC2 instances in subnets.

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

Prerequisites for sharing Outposts resources

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

Related services

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

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

Sharing across Availability Zones

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

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

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

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

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

Sharing an Outpost resource

When an owner shares an Outpost with a consumer, the consumer can create resources on the Outpost in the same way that they would create resources on Outposts that they create in their own account. Consumers with access to shared local gateway route tables can create and manage VPC associations. For more information, see Shareable Outpost resources (p. 33).

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

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

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

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

1. Open the AWS Outposts console at https://console.aws.amazon.com/outposts/.
2. On the navigation pane, choose **Outposts**.
3. Select the Outpost, and then choose **Actions**, **View details**.
4. On the **Outpost summary** page, choose **Resource shares**.
5. Choose **Create resource share**.

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

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

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

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

Use the create-resource-share command.

Unsharing a shared Outpost resource

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

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

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

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

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

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

Use the disassociate-resource-share command.

Identifying a shared Outpost resource

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

To identify a shared Outpost using the AWS Outposts console

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

To identify a shared Outpost resource using the AWS CLI

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

Shared Outpost resource permissions

Permissions for owners

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

Permissions for consumers

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

Billing and metering

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

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

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

Resource quotas

Outposts are purchased as a pre-validated capacity configuration. There are no Outposts-specific limits or quotas for Outposts resources.

Security in AWS Outposts

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

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

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

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

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

Contents

Data protection in AWS Outposts

AWS Outposts conforms to the AWS shared responsibility model, which includes regulations and guidelines for data protection. AWS is responsible for protecting the global infrastructure that runs all AWS services. AWS maintains control over data hosted on this infrastructure, including the security configuration controls for handling customer content and personal data. AWS customers and APN Partners, acting either as data controllers or data processors, are responsible for any personal data that they put in the AWS Cloud.

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

For more information about data protection, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

Encryption at Rest

Amazon EBS encryption is an encryption solution for your EBS volumes and snapshots. It uses AWS Key Management Service (AWS KMS) customer master keys (CMK). With Outposts, encryption is enabled by default. For more information, see Amazon EBS Encryption in the *Amazon EC2 User Guide*.

Encryption in transit

AWS encrypts in-transit data between your Outpost and its AWS Region. For more information, see Connectivity through service links (p. 15).

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

Data deletion

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

For information about data deletion during required hardware maintenance, see Hardware maintenance (p. 49).

Identity and Access Management for AWS Outposts

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be authenticated (signed in) and authorized (have permissions) to use AWS resources. IAM enables you to create users and groups under your AWS account. You control the permissions that users have to perform tasks using AWS resources. You can use IAM for no additional charge.

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

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

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

Policy structure

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

```
{
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"*",
    "Condition":{
      "condition":{
        "key":"value"
        }
      }
    }
```

```
  ]
}
```

There are various elements that make up a statement:

- **Effect:** The effect can be `Allow` or `Deny`. By default, IAM users don't have permission to use resources and API actions, so all requests are denied. An explicit allow overrides the default. An explicit deny overrides any allows.
- **Action**: The action is the specific API action for which you are granting or denying permission.
- **Resource**: The resource that's affected by the action. Some API actions allow you to include specific resources in your policy that can be created or modified by the action. To specify a resource in the statement, you need to use its Amazon Resource Name (ARN).
- **Condition**: Conditions are optional. Use them to control when your policy is in effect.

Example policies

In an IAM policy statement, you can specify any API action from any service that supports IAM. For AWS Outposts, use the following prefix with the name of the API action: `outposts:`. For example:

- `outposts:CreateOutpost`
- `outposts:DescribeOutposts`

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

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

You can also specify multiple actions using wildcards. For example, you can specify all AWS Outposts API actions whose name begins with the word "Get".

```
"Action": "outposts:Get*"
```

To specify all AWS Outposts API actions, use the * wildcard.

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

Using Temporary Credentials with AWS Outposts

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

AWS Outposts supports using temporary credentials.

Considerations

AWS Outposts does not support specifying resource ARNs in an IAM policy, tagging resources, or controlling access based on tags.

AWS Outposts does not provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see AWS Global Condition Context Keys in the *IAM User Guide*.

Infrastructure security in AWS Outposts

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

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

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

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

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

Resilience in AWS Outposts

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

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

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

Compliance validation for AWS Outposts

The existing compliance certifications for AWS Services apply to services running entirely in an AWS Region. AWS Outposts and services on an Outpost require a separate evaluation for certifications.

Under the shared responsibility model, AWS is responsible for the hardware and software that run AWS services. This applies to AWS Outposts, just as it does to an AWS Region. This includes patching the infrastructure software and configuring infrastructure devices. As a customer, you are responsible for implementing best practices for data encryption, patching their guest operating system and applications, identity and access management, and operating system, network, and firewall configurations.

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

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

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

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

Monitor your Outpost

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

CloudWatch metrics

You can use Amazon CloudWatch to retrieve statistics about data points for your Outposts as an ordered set of time series data, known as *metrics*. You can use these metrics to verify that your system is performing as expected. For more information, see CloudWatch metrics for AWS Outposts (p. 43).

CloudTrail logs

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

The CloudTrail logs contain information about the calls to API actions for AWS Outposts. They also contain information for calls to API actions from services on an Outpost, such as Amazon EC2 and Amazon EBS. For more information, see AWS Outposts information in CloudTrail (p. 47).

VPC Flow Logs

You can 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*.

CloudWatch metrics for AWS Outposts

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

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

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

Contents

Outpost metrics

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

Metric	Description
`ConnectedStatus`	The status of an Outpost's service link connection. If the average statistic is less than `1`, the connection is impaired. **Unit**: Count **Maximum resolution**: 1 minute **Statistics**: The most useful statistic is `Average`. **Dimensions** • `OutpostId`
`CapacityExceptions`	The number of insufficient capacity errors for instance launches. **Unit**: Count **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Maximum` and `Minimum`. **Dimensions** • `OutpostId` • `InstanceType`, `OutpostId`
`InstanceFamilyCapacityAvailability`	The percentage of instance capacity available. **Unit**: Percent **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Average` and `pNN.NN` (percentiles). **Dimensions** • `InstanceFamily`, `OutpostId`
`InstanceFamilyCapacityUtilization`	The percentage of instance capacity in use. **Unit**: Percent **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Average` and `pNN.NN` (percentiles). **Dimensions** • `InstanceFamily`, `OutpostId`
`InstanceTypeCapacityAvailability`	The percentage of instance capacity available.

Metric	Description
	Unit: Percent **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Average` and `pNN.NN` (percentiles). **Dimensions** • `InstanceType`, `OutpostId`
`InstanceTypeCapacityUtilization`	The percentage of instance capacity in use. **Unit**: Percent **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Average` and `pNN.NN` (percentiles). **Dimensions** • `InstanceType`, `OutpostId`
`UsedInstanceType_Count`	The number of instance types that are currently in use. **Unit**: Count **Maximum resolution**: 5 minutes **Dimensions** • `InstanceType`, `OutpostId`
`AvailableInstanceType_Count`	The number of available instance types. **Unit**: Count **Maximum resolution**: 5 minutes **Dimensions** • `InstanceType`, `OutpostId`
`EBSVolumeTypeCapacityUtilization`	The percentage of EBS volume type capacity in use. **Unit**: Percent **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Average` and `pNN.NN` (percentiles). **Dimensions** • `VolumeType`, `OutpostId`

Metric	Description
EBSVolumeTypeCapacityAvailablePercentage	The percentage of EBS volume type capacity available. **Unit**: Percent **Maximum resolution**: 5 minutes **Statistics**: The most useful statistics are `Average` and `pNN.NN` (percentiles). **Dimensions** • `VolumeType`, `OutpostId`

Outpost metric dimensions

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

Dimension	Description
`InstanceFamily`	The instance family.
`InstanceType`	The instance type.
`OutpostId`	The ID of the Outpost.
`VolumeType`	The EBS volume type.

View CloudWatch metrics for your outpost

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

To view metrics using the CloudWatch console

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

To view metrics using the AWS CLI

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

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

To get the statistics for a metric using the AWS CLI

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

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

Logging AWS Outposts API calls with AWS CloudTrail

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

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

AWS Outposts information in CloudTrail

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

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

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

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

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

- With root or AWS Identity and Access Management (IAM) user credentials.
- With temporary security credentials for a role or federated user.
- By another AWS service.

For more information, see the CloudTrail userIdentity element.

Understanding AWS Outposts log file entries

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

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


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

Outpost maintenance

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

Hardware maintenance

If AWS detects an irreparable issue with hardware hosting EC2 instances running on your Outpost, we will send you instance retirement notices for the affected instances. If you stop and start an affected instance, this migrates it to the available capacity, and the hypervisor scrubs (sets to zero) all data that was allocated to the instance from the hardware. If you do not stop and start an affected instance, AWS stops and starts it for you when it reaches its scheduled retirement date. For more information, see Instance Retirement in the *Amazon EC2 User Guide*.

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

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

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.

Document history

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

update-history-change	update-history-description	update-history-date
Shared AWS Outposts resources (p. 50)	With Outpost sharing, Outpost owners can share their Outposts and Outpost resources, including local gateway route tables, with other AWS accounts under the same AWS organization. For more information, see Working with shared AWS Outposts resources in the *AWS Outposts User Guide*.	October 15, 2020
Additional CloudWatch metrics (p. 50)	Additional CloudWatch metrics for instance type counts are available. For more information, see CloudWatch metrics for AWS Outposts in the *AWS Outposts User Guide*.	September 21, 2020
Additional CloudWatch metric (p. 50)	An additional CloudWatch metric for service link connected status is available. For more information, see CloudWatch metrics for AWS Outposts in the *AWS Outposts User Guide*.	September 11, 2020
Support for sharing customer-owned IPv4 addresses (p. 50)	Use AWS Resource Access Manager to share customer-owned IPv4 addresses. For more information, see Customer-owned IP addresses in the *AWS Outposts User Guide*.	April 20, 2020
Additional CloudWatch metrics (p. 50)	Additional CloudWatch metrics for EBS volumes are available. For more information, see CloudWatch metrics for AWS Outposts in the *AWS Outposts User Guide*.	April 4, 2020
Initial release (p. 50)	This is the initial release of AWS Outposts.	December 3, 2019