AWS Whitepaper

AWS Outposts High Availability Design and Architecture Considerations

AWS Outposts High Availability Design and Architecture Considerations: AWS Whitepaper

Table of Contents

AWS Outposts High Availability Design and Architecture Considerations

Publication date: **August 12, 2021** (*Document history*)

This whitepaper discusses architecture considerations and recommended practices that IT managers and system architects can apply to build highly available on-premises application environments with AWS Outposts.

Are you Well-Architected?

The AWS Well-Architected Framework helps you understand the pros and cons of the decisions you make when building systems in the cloud. The six pillars of the Framework allow you to learn architectural best practices for designing and operating reliable, secure, efficient, cost-effective, and sustainable systems. Using the AWS Well-Architected Tool, available at no charge in the AWS Management Console, you can review your workloads against these best practices by answering a set of questions for each pillar.

For more expert guidance and best practices for your cloud architecture—reference architecture deployments, diagrams, and whitepapers—refer to the AWS Architecture Center.

Introduction

This paper is intended for IT managers and system architects looking to deploy, migrate, and operate applications using the AWS cloud platform and run those applications on premises with AWS Outposts rack, the 42U rack form factor of AWS Outposts.

It introduces the architecture patterns, anti-patterns, and recommended practices for building highly available systems that include AWS Outposts rack. You will learn how to manage your AWS Outposts rack capacity and use networking and data center facility services to set up highly available AWS Outposts rack infrastructure solutions.

AWS Outposts rack is a fully managed service that provides a logical pool of cloud compute, storage, and networking capabilities. With Outposts racks, customers can use supported AWS managed services in their on-premises environments, including: Amazon Elastic Compute Cloud

(Amazon EC2), Amazon Elastic Block Store (Amazon EBS), Amazon S3 on Outposts, Amazon Elastic Kubernetes Service (Amazon EKS), Amazon Elastic Container Service (Amazon ECS), Amazon Relational Database Service (Amazon RDS), and other AWS services on Outposts. Services on Outposts are delivered on the same AWS Nitro System used in the AWS Regions.

By leveraging AWS Outposts rack, you can build, manage, and scale highly available on-premises applications using familiar AWS cloud services and tools. AWS Outposts rack is ideal for workloads that require low latency access to on-premises systems, local data processing, data residency, and migration of applications with local system interdependencies.

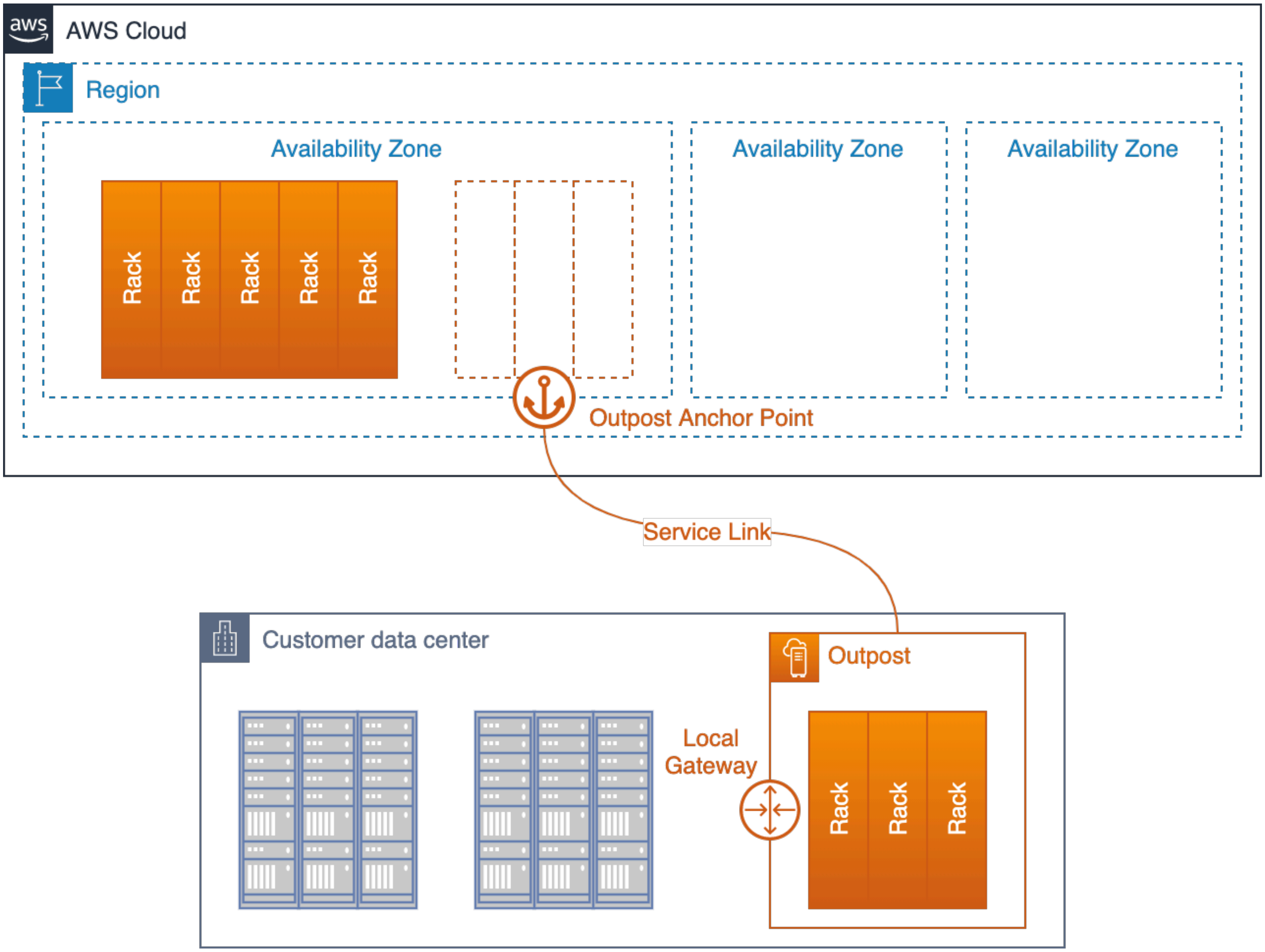
Extending AWS infrastructure and services to on-premises locations

The AWS Outposts service delivers AWS infrastructure and services to on-premises locations in more than 50 countries and territories, giving customers the ability to deploy the same AWS infrastructure, AWS services, APIs, and tools to virtually any datacenter, co-location space, or on-premises facility for a truly consistent hybrid experience. To understand how to design with Outposts, you should understand the different tiers that make up the AWS cloud.

An AWS Region is a geographical area of the world. Each AWS Region is a collection of data centers that are logically grouped into Availability Zones (AZs). AWS Regions provide multiple (at least two) physically separated and isolated Availability Zones which are connected with low latency, high throughput, and redundant network connectivity. Each AZ consists of one or more physical data centers.

A logical Outpost (hereafter referred to as an Outpost) is a deployment of one or more physically connected AWS Outposts racks managed as a single entity. An Outpost provides a pool of AWS compute and storage capacity at one of your sites as a private extension of an AZ in an AWS Region.

Perhaps the best conceptual model for AWS Outposts is to think of unplugging one or more racks from a data center in an AZ of an AWS Region, and installing it in your own data center or colocation facility. You roll the racks from the AZ data center to your data center. You then plug the racks into anchor points in the AZ data center with a (very) long cable so that the racks continue to function as a part of the AWS Region. You also plug them into your local network to provide low latency connectivity between your on-premises networks and the workloads running on those racks. This gives you the operational and API consistency of the AWS Cloud, while keeping your workload local.



An Outpost deployed in a customer data center and connected back to its anchor AZ and parent Region

The Outpost functions as an extension of the AZ where it is anchored. AWS operates, monitors, and manages AWS Outposts infrastructure as part of the AWS Region. Instead of a very long physical cable, an Outpost connects back to its parent Region through a set of encrypted VPN tunnels called the Service Link.

The Service Link terminates on a set of anchor points in an Availability Zone (AZ) in the Outpost's parent Region.

You choose where your content is stored. You can replicate and back up your content to the AWS Region or other locations. Your content will not be moved or copied outside of your chosen locations without your agreement, except as necessary to comply with the law or a binding order of a governmental body. For more information, see AWS Data Privacy FAQ.

The workloads that you deploy on those racks run locally. And, while the compute and storage capacity available in those racks is finite and cannot accommodate running the cloud-scale services of an AWS Region, the resources deployed on the rack (your instances and their local storage) receive the benefits of running locally while the management plane continues to operate in the AWS Region.

To deploy workloads on an Outpost, you add subnets to your Virtual Private Cloud (VPC) environments and specify an Outpost as the location for the subnets. Then, you select the desired subnet when deploying supported AWS resources through the AWS Management Console, CLI, APIs, CDK, or infrastructure as code (IaC) tools. Instances in Outpost subnets communicate with other instances on the Outpost or in the Region through VPC networking.

The Outpost Service Link carries both Outpost management traffic and customer VPC traffic (VPC traffic between the subnets on the Outpost and the subnets in the Region).

Important terms:

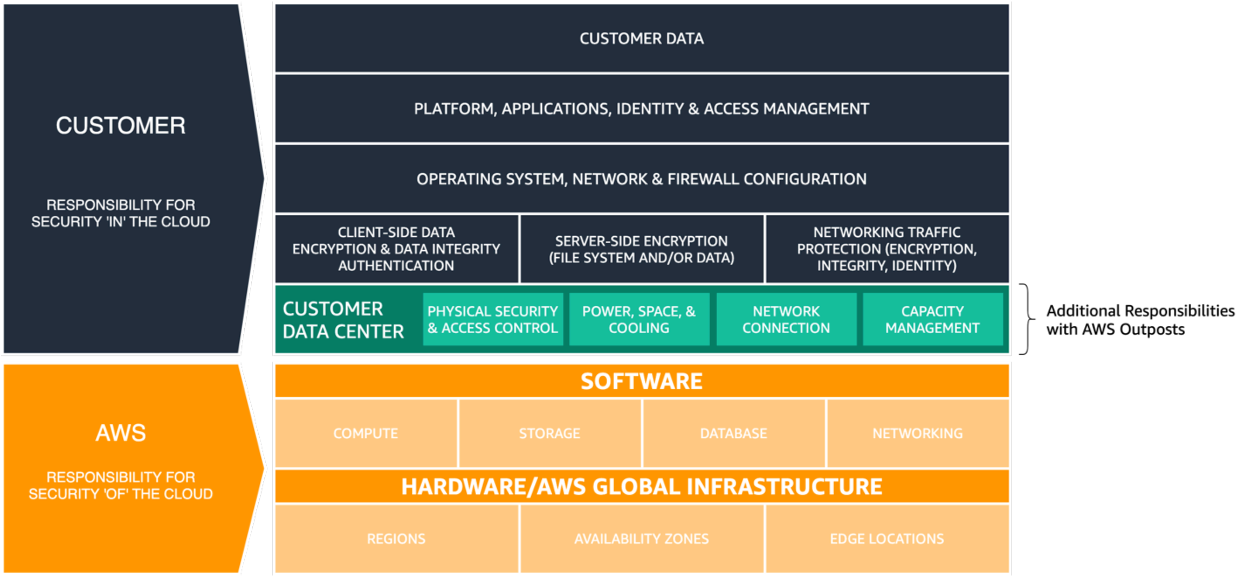
- **AWS Outposts –** is a fully managed *service* that offers the same AWS infrastructure, AWS services, APIs, and tools to virtually any datacenter, co-location space, or on-premises facility for a truly consistent hybrid experience.
- **Outpost –** is a *deployment* of one or more physically connected AWS Outposts racks that is managed as a single logical entity and pool of AWS compute, storage, and networking deployed at a customer site.
- **Parent Region –** the AWS Region which provides the management, control plane services, and regional AWS services for an Outpost deployment.
- **Anchor Availability Zone (anchor AZ) –** the Availability Zone in the parent Region that hosts the anchor points for an Outpost. An Outpost functions as an extension of its anchor AZ. The anchor AZ is chosen by the customer when the Outposts order is placed. After an anchor AZ is chosen, it cannot be changed for the duration of the AWS Outposts subscription term.
- **Anchor Points –** endpoints in the anchor AZ that receive the connections from remotely deployed Outposts.
- **Service Link –** a set of encrypted VPN tunnels that connect an Outpost to its anchor Availability Zone in its parent Region.

- **Local Gateway (LGW)** – A logical interconnect virtual router that enables communication between your Outpost and your on-premises network.

Understanding the AWS Outposts Shared Responsibility Model

When you deploy AWS Outposts infrastructure into your data centers or co-location facilities, you take on additional responsibilities in the AWS Shared Responsibility model. For example, in the Region, AWS provides diverse power sources, redundant core networking, and resilient Wide Area Network (WAN) connectivity to ensure services are available in the event of one or more component failures.

With Outposts, you are responsible for providing resilient power and network connectivity to the Outpost racks to meet your availability requirements for workloads running on Outposts.


CUSTOMER
RESPONSIBILITY FOR SECURITY 'IN' THE CLOUD
CUSTOMER DATA
PLATFORM, APPLICATIONS, IDENTITY & ACCESS MANAGEMENT
OPERATING SYSTEM, NETWORK & FIREWALL CONFIGURATION
CLIENT-SIDE DATA ENCRYPTION & DATA INTEGRITY AUTHENTICATION
SERVER-SIDE ENCRYPTION (FILE SYSTEM AND/OR DATA)
NETWORKING TRAFFIC PROTECTION (ENCRYPTION, INTEGRITY, IDENTITY)
CUSTOMER DATA CENTER
PHYSICAL SECURITY & ACCESS CONTROL
POWER, SPACE, & COOLING
NETWORK CONNECTION
CAPACITY MANAGEMENT
Additional Responsibilities with AWS Outposts
AWS
RESPONSIBILITY FOR SECURITY 'OF' THE CLOUD
SOFTWARE
COMPUTE
STORAGE
DATABASE
NETWORKING
HARDWARE/AWS GLOBAL INFRASTRUCTURE
REGIONS
AVAILABILITY ZONES
EDGE LOCATIONS

AWS Shared Responsibility Model updated for AWS Outposts

With AWS Outposts, you are responsible for the physical security and access controls of the data center environment. You must provide sufficient power, space, and cooling to keep the Outpost operational and network connections to connect the Outpost back to the Region.

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

AWS is responsible for the availability of the Outposts infrastructure including the power supplies, servers, and networking equipment within the AWS Outposts racks. AWS also manages the virtualization hypervisor, storage systems, and the AWS services that run on Outposts.

A central power shelf in each Outposts rack converts from AC to DC power and supplies power to servers in the rack via a bus bar architecture. With the bus bar architecture, half the power supplies in the rack can fail and all the servers will continue to run uninterrupted.

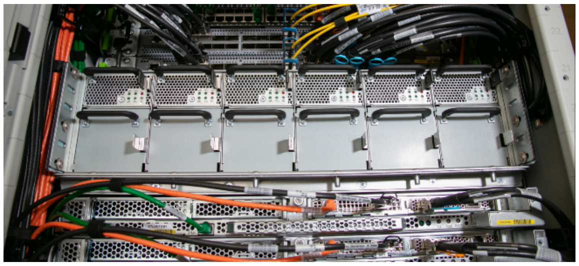


Figure 3 - AWS Outposts AC-to-DC power supplies and bus bar power distribution

The network switches and cabling within and between the Outposts racks are also fully redundant. A fiber patch panel provides connectivity between an Outpost rack and the on-premises network and serves as the demarcation point between the customer-managed data center environment and the managed AWS Outposts environment.

Just like in the Region, AWS is responsible for the cloud services offered on Outposts and takes on additional responsibilities as you select and deploy higher-level managed services like Amazon RDS on Outposts. You should review the AWS Shared Responsibility Model and the Frequently Asked Questions (FAQ) pages for individual services as you consider and select services to deploy on Outposts. These resources provide additional details on the division of responsibilities between you and AWS.

Thinking in terms of failure modes

When designing a highly available application or system you must consider what components might fail, what impact component failures will have on the system as well as your application RPO/RTO goals, and what mechanisms you can implement to mitigate or eliminate the impact of component failures. Does your application run on a single server, in a single rack, or in a single data center? What will happen when a server, rack, or data center experiences a temporary or permanent failure? What happens when there is a failure in a critical sub-system like networking or within the application itself? These are failure modes.

You should consider the failure modes in this section when planning your Outposts and application deployments. The sections that follow will review how to mitigate these failure modes to provide an increased level of high availability for your application environment.

Failure mode 1: Network

An Outpost deployment depends on a resilient connection to its parent Region for management and monitoring. Network disruptions may be caused by a variety of failures such as operator errors, equipment failures, and service provider outages. An Outpost, which may be comprised of one or more racks connected together at the site, is considered disconnected when it cannot communicate with the Region via the Service Link.

Redundant network paths can help mitigate the risk of disconnect events. You should map application dependencies and network traffic to understand the impact disconnect events will have on workload operations. Plan sufficient network redundancy to meet your application availability requirements.

During a disconnect event, instances running on an Outpost continue to run and are accessible from on-premises networks through the Outpost Local Gateway (LGW). Local workloads and services may be impaired or fail if they rely on services in the Region. Mutating requests (like starting or stopping instances on the Outpost), control plane operations, and service telemetry (for example, CloudWatch metrics) will fail while the Outpost is disconnected from the Region. CloudWatch metrics will be spooled locally on your Outpost for short periods of network disconnect, and will be sent to the Region for review when the service link connection is re-established.

Failure mode 2: Instances

Amazon EC2 instances may become impaired or fail if the server they are running on has an issue or if the instance experiences an operating system or application failure. How applications handle these types of failures depends on the application architecture. Monolithic applications typically use application or system features for recovery while modular service oriented or microservices architectures typically replace failed components to maintain service availability.

You can replace failed instances with new instances using automated mechanisms like Amazon EC2 Auto Scaling groups. Instance auto recovery can restart instances that fail due to server failures provided there is sufficient spare capacity available on the remaining servers and the service link is still connected.

Failure mode 3: Compute

Servers can fail or become impaired and may need to be taken out of operation (temporarily or permanently) for a variety of reasons, such as component failures and scheduled maintenance operations. How services on Outposts rack handle server failures and impairments varies and can depend on how customers configure high availability options.

You should order sufficient compute capacity to support an `N+M` availability model, where `N` is the required capacity and `M` is the spare capacity provisioned to accommodate server failures.

Hardware replacements for failed servers are provided as part of the fully managed AWS Outposts rack service. AWS actively monitors the health of all servers and networking devices in an Outpost deployment. If there is a need to perform physical maintenance, AWS will schedule a time to visit your site to replace failed components. Provisioning spare capacity allows you to keep your workloads resilient against host failures while unhealthy servers are taken out of service and replaced.

Failure mode 4: Racks or data centers

Rack failures may occur due to a total loss of power to racks or due to environmental failures like loss of cooling or physical damage to the data center from a flood or earthquake. Deficiencies in data center power distribution architectures or errors during standard data center power maintenance can result in loss of power to one or more racks or even the entire data center.

These scenarios can be mitigated by deploying infrastructure to multiple data center floors or locations that are independent from one another within the same campus or metro area.

Taking this approach with AWS Outposts rack will require careful consideration for how applications are architected and distributed to run across multiple separate logical Outposts to maintain application availability.

Failure mode 5: AWS Availability Zone or Region

Each Outpost is anchored to a specific Availability Zone (AZ) within an AWS Region. Failures within the anchor AZ or parent Region could cause the loss of Outpost management and mutability and may disrupt network communication between the Outpost and the Region.

Similar to network failures, AZ or Region failures may cause the Outpost to become disconnected from the Region. The instances running on an Outpost continue to run and are accessible from on-premises networks through the Outpost Local Gateway (LGW) and may be impaired or fail if they rely on services in the Region, as described previously.

To mitigate the impact of AWS AZ and Region failures, you can deploy multiple Outposts each anchored to a different AZ or Region. You may then design your workload to operate in a distributed multi-Outpost deployment model using many of the similar mechanisms and architectural patterns that you use to design and deploy on AWS today.

The control plane of the services that run on AWS Outposts resides in the Region to which it is anchored, generating a dependency both on Zonal services such as Amazon EC2 and Amazon EBS and on Regional services such as Amazon RDS, Elastic Load Balancing and Amazon EKS. In Outposts, applications can be deployed under the concept of static stability to help improve resilience to control plane impairments.

Building HA applications and infrastructure solutions with AWS Outposts rack

With AWS Outposts rack, you can build, manage, and scale highly available on-premises applications using familiar AWS cloud services and tools. It's important to understand cloud HA architectures and approaches are generally different from traditional on-premises HA architectures you may be running in your datacenter today.

With traditional on-premises HA application deployments, applications are deployed in virtual machines (VMs). Complex IT systems and infrastructure are deployed and maintained to keep those virtual machines running and healthy. The VMs often have specific identities and each VM may play a critical role in the total application architecture.

Architectural roles are tightly coupled to VM identities. Systems architects leverage IT infrastructure features to provide highly available VM runtime environments that provide each VM with reliable access to compute capacity, storage volumes, and network services. If a VM fails, automated or manual recovery processes are run to restore the failed VM to a healthy state, often on other infrastructure or in another datacenter entirely.

Cloud HA architectures take a different approach. AWS cloud services provide reliable compute, storage, and networking capabilities. Application components are deployed to EC2 instances, containers, serverless functions, or other managed services.

An instance is an instantiation of an application component – perhaps one of many performing that role. Application components are loosely coupled to one another and to the role they play in the total application architecture. The individual identity of an instance is generally not important. Additional instances may be created or destroyed to scale up or scale down in response to demand. Failed instances or unhealthy instances are simply replaced with new healthy instances.

AWS Outposts rack is a fully managed service that extends AWS compute, storage, networking, database, and other *cloud services* to on-premises locations for a truly consistent hybrid experience. You should not think of the Outposts rack service as a drop-in replacement for IT infrastructure systems with traditional on-premises HA mechanisms. Attempting to use AWS services and Outposts to support a traditional on-premises HA architecture is an anti-pattern.

Workloads running on AWS Outposts rack *use cloud HA mechanisms* like Amazon EC2 Auto Scaling (to scale horizontally to meet workload demands), EC2 health checks (to detect and remove unhealthy instances), and Application Load Balancers (to redirect incoming workload traffic to scaled or replaced instances). When migrating applications to the cloud, whether to an AWS Region or AWS Outposts rack, you should update your HA application architecture to begin taking advantage of managed cloud services and cloud HA mechanisms.

The following sections introduce architecture patterns, anti-patterns, and recommended practices for deploying AWS Outposts rack in your on-premises environments to run workloads with high availability requirements. These sections introduce patterns and practices; however, they do not provide configuration and implementation details. You should read and become familiar with the AWS Outposts rack FAQs and User Guide and the FAQs and service documentation for the services that run on Outposts rack as you prepare your environment for Outposts rack and your applications for migration to AWS services.

Topics

- Networking
- Compute
- Storage
- Databases
- Larger failure modes

Networking

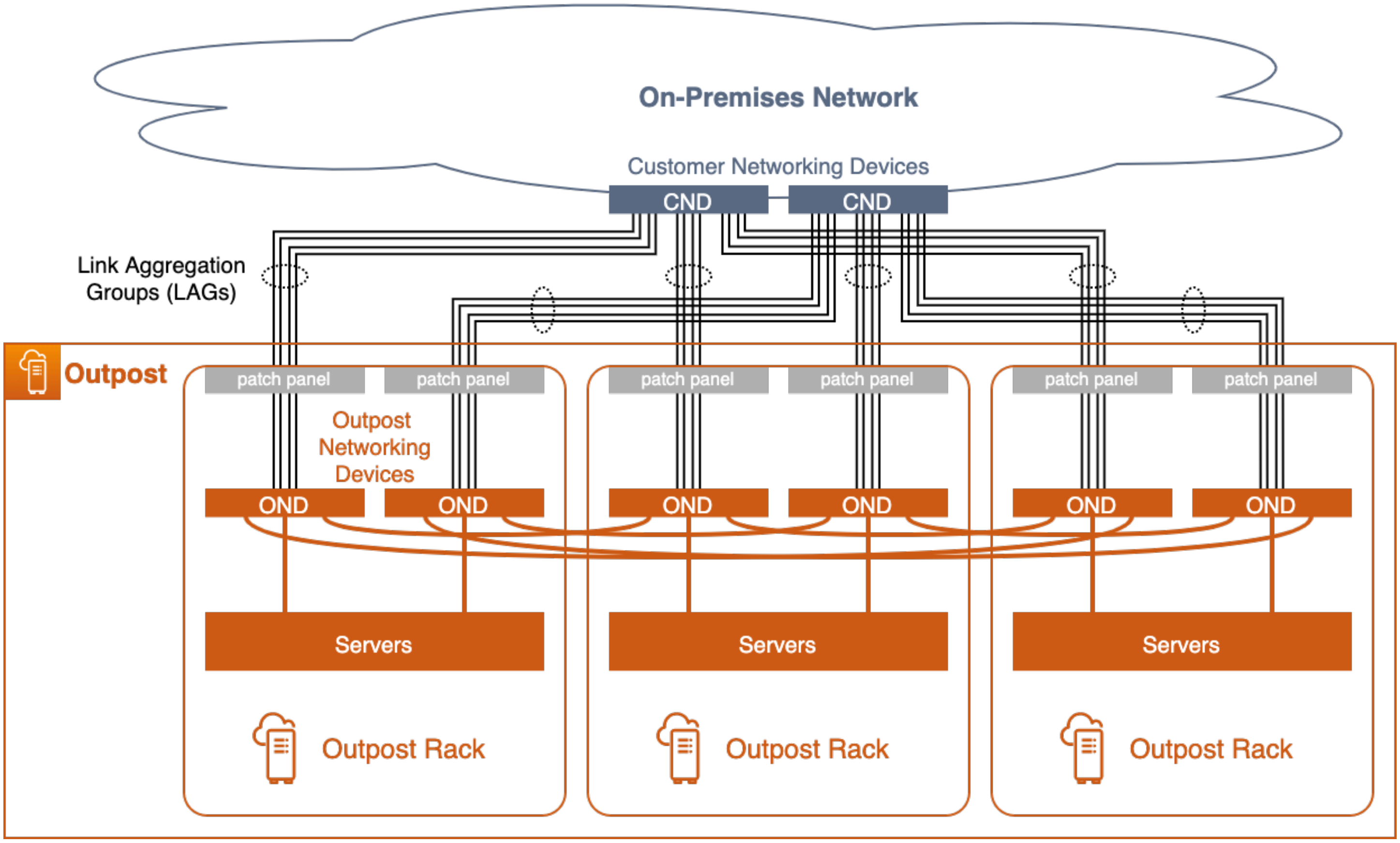
An Outpost deployment depends on a resilient connection to its anchor AZ for management, monitoring, and service operations to function properly. You should provision your on-premises network to provide redundant network connections for each Outpost rack and reliable connectivity back to the anchor points in the AWS cloud. Also consider network paths between the application workloads running on the Outpost and the other on-premises and cloud systems they communicate with – how will you route this traffic in your network?

Topics

- Network attachment
- Anchor connectivity
- Application/workload routing

Network attachment

Each AWS Outposts rack is configured with redundant top-of-rack switches called Outpost Networking Devices (ONDs). The compute and storage servers in each rack connect to both ONDs. You should connect each OND to a separate switch called a Customer Networking Device (CND) in your data center to provide diverse physical and logical paths for each Outpost rack. ONDs connect to your CNDs with one or more physical connections using fiber optic cables and optical transceivers. The physical connections are configured in logical link aggregation group (LAG) links.



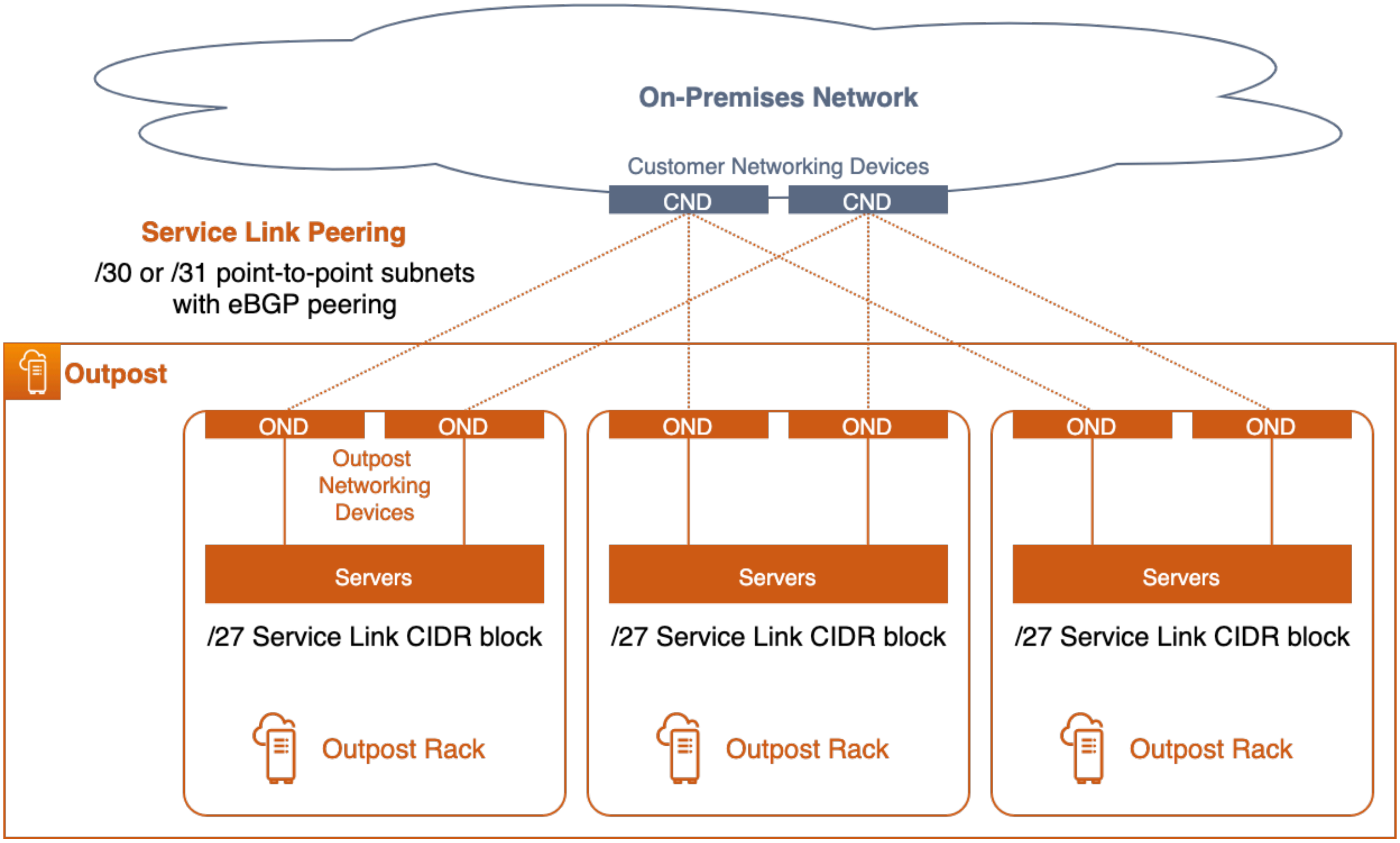
Multi-rack Outpost with redundant network attachments

The OND to CND links are always configured in a LAG – even if the physical connection is a single fiber optic cable. Configuring the links as LAG groups allow you to increase the link bandwidth by adding additional physical connections to the logical group. The LAG links are configured as IEEE 802.1q Ethernet trunks to enable segregated networking between the Outpost and the on-premises network.

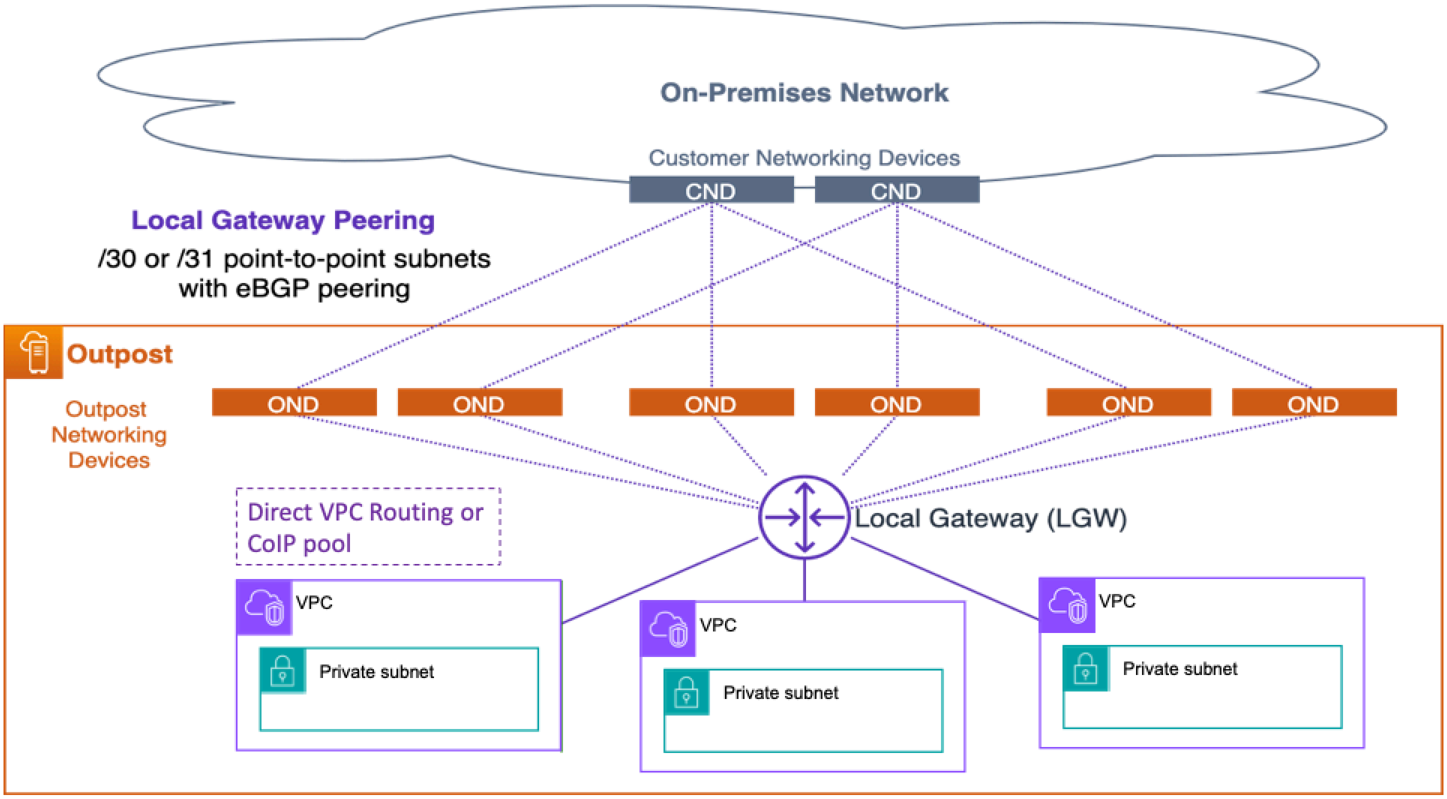
Every Outpost has at least two logically segregated networks that need to communicate with or across the customer network:

- **Service link network** – allocates the service link IP addresses to the Outpost servers and facilitates communication with the on-premises network to allow the servers to connect back to the Outpost anchor points in the Region. When you have multiple rack implementations in a single logical Outposts, you need to assign a Service Link /26 CIDR for each Rack.
- **Local Gateway network** – enables communication between the VPC subnets on the Outpost and the on-premises network via the Outpost Local Gateway (LGW).

These segregated networks attach to the on-premises network by a set of point-to-point IP connections over the LAG links. Each OND to CND LAG link is configured with VLAN IDs, point-to-point (/30 or /31) IP subnets, and eBGP peering for each segregated network (service link and LGW). You should consider the LAG links, with their point-to-point VLANs and subnets, as layer-2 segmented, routed layer-3 connections. The routed IP connections provide redundant logical paths that facilitate communication between the segregated networks on the Outpost and the on-premises network.


On-Premises Network
Customer Networking Devices
CND
CND
Service Link Peering
/30 or /31 point-to-point subnets with eBGP peering
Outpost
OND
OND
Outpost Networking Devices
Servers
/27 Service Link CIDR block
Outpost Rack
OND
OND
Servers
/27 Service Link CIDR block
Outpost Rack
OND
OND
Servers
/27 Service Link CIDR block
Outpost Rack

Service link peering



Local Gateway peering

You should terminate the layer-2 LAG links (and their VLANs) on the directly attached CND switches and configure the IP interfaces and BGP peering on the CND switches. You should not bridge the LAG VLANs between your data center switches. For more information, see Network layer connectivity in the *AWS Outposts User Guide*.

Inside a logical multi-rack Outpost, the ONDs are redundantly interconnected to provide highly available network connectivity between the racks and the workloads running on the servers. AWS is responsible for network availability within the Outpost.

Recommended practices for highly available network attachment without ACE

- Connect each Outpost Networking Device (OND) in an Outpost rack to a separate Customer Networking Device (CND) in the data center.
- Terminate the layer-2 links, VLANs, layer-3 IP subnets, and BGP peering on the directly attached Customer Networking Device (CND) switches. Do not bridge the OND to CND VLANs between the CNDs or across the on-premises network.

- Add links to the Link Aggregation Groups (LAGs) to increase the available bandwidth between the Outpost and the data center. Do not rely on the aggregate bandwidth of the diverse paths through both ONDs.
- Use the diverse paths through the redundant ONDs to provide resilient connectivity between the Outpost networks and the on-premises network.
- To achieve optimal redundancy and allow for non-disruptive OND maintenance, we recommend that customers configure BGP advertisements and policies as follows:
  - Customer network equipment should receive BGP advertisements from Outpost without changing BGP attributes and to enable BGP multipath/load-balancing to achieve optimal inbound traffic flows (from customer towards Outpost). AS-Path prepending is used for Outpost BGP prefixes to shift traffic away from a particular OND/uplink in case maintenance is required. The customer network should prefer routes from Outpost with AS-Path length 1 over routes with AS-Path length 4, that is, react to AS-Path prepending.
  - The customer network should advertise equal BGP prefixes with the same attributes towards all ONDs in Outpost. By default, the Outpost network load balances outbound traffic (towards the customer) between all uplinks. Routing policies are used on the Outpost side to shift traffic away from a particular OND in case maintenance is required. Equal BGP prefixes from the customer side on all ONDs are required to perform this traffic shift, and perform maintenance in a non-disruptive way. When maintenance is required on the customer's network, we recommend using AS-Path prepending to temporarily shift away traffic from particular uplink or device.

Recommended practices for highly available network attachment with ACE

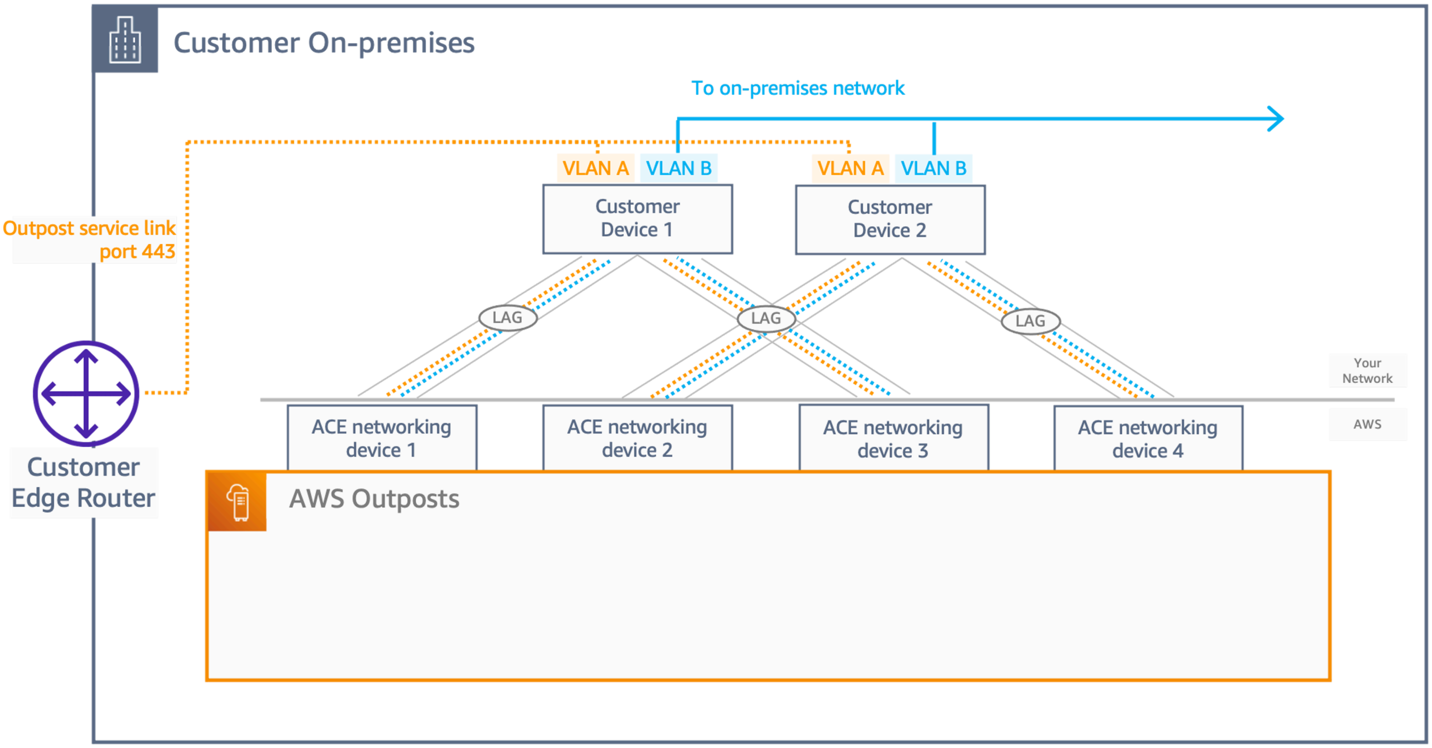
For a multi-rack deployment with four or more compute racks, you must use the Aggregation, Core, Edge (ACE) rack, which will act as a network aggregation point to reduce the number of fiber links to your on-premises networking devices. The ACE rack provides the connectivity to the ONDs in each Outposts rack, so AWS will own the VLAN interface allocation and configuration between ONDs and ACE networking devices.

Isolated network layers for Service Link and Local Gateway networks are still required regardless of whether or not an ACE rack is used, which aim to have a VLAN point-to-point (/30 or /31) IP subnets, and eBGP peering configuration for each segregated network. The proposed architectures should follow any of two architectures:

Topics

- Two customer networking devices
- Four customer networking devices

Two customer networking devices

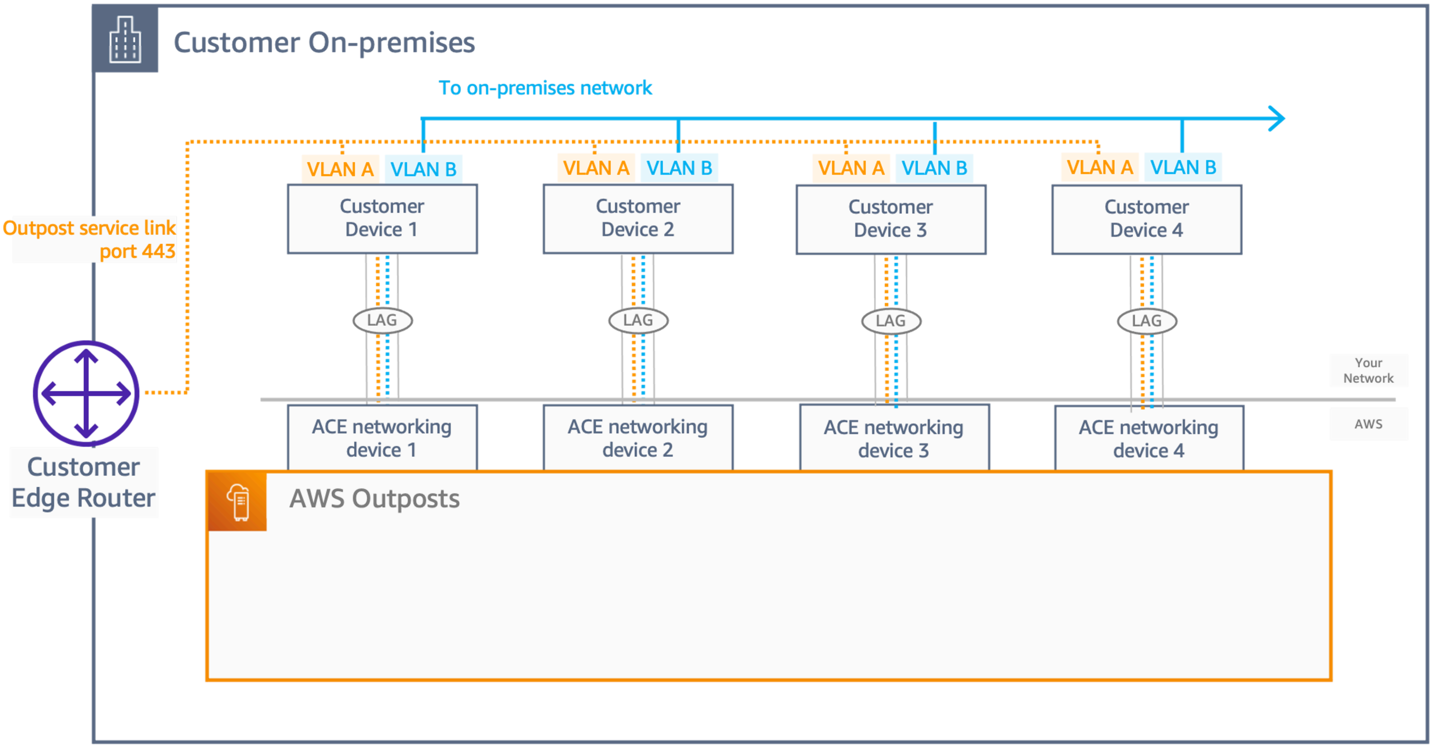


Two-customer networking devices

- With this architecture, the customer should have two networking devices (CND) to interconnect the ACE networking devices, providing redundancy.
- For each physical connections, you must enable a LAG (to increase the available bandwidth between the Outpost and the data center), even if it is a single physical port, and it will carry two network segments, having 2 point-to-point VLANs (/30 or /31), and eBGP configurations between ACEs and CNDs.
- In a steady state the traffic is load balanced following Equal-cost multipath (ECMP) pattern to/from the customer network from the ACE layer, 25% traffic distribution across the ACE to customer. In order to allow this behavior, the eBGP peering's between ACEs and CNDs must have BGP multipath/load-balancing enabled, and announced the customer prefixes with the same BGP metric on the 4 eBGP peering connections.
- To achieve optimal redundancy and allow for non-disruptive OND maintenance, we recommend that customers to follow these recommendations:

- Customer networking device should advertise equal BGP prefixes with the same attributes towards all ONDs in Outpost.
- Customer networking device should receive BGP advertisements from Outpost without changing BGP attributes and to enable BGP multipath/load-balancing.

Four customer networking devices



Four-customer network devices

With this architecture, the customer will have four networking devices (CND) to interconnect the ACE networking devices, providing redundancy and the same networking logic, including VLANs, eBGP, and ECMP applicable to a 2 CND architecture.

Anchor connectivity

An Outpost service link connects to either public or private anchors (not both) in a specific Availability Zone (AZ) in the Outpost's parent Region. Outpost servers initiate outbound service link VPN connections from their service link IP addresses to the anchor points in the anchor AZ. These connections use UDP and TCP port 443. AWS is responsible for the availability of the anchor points in the Region.

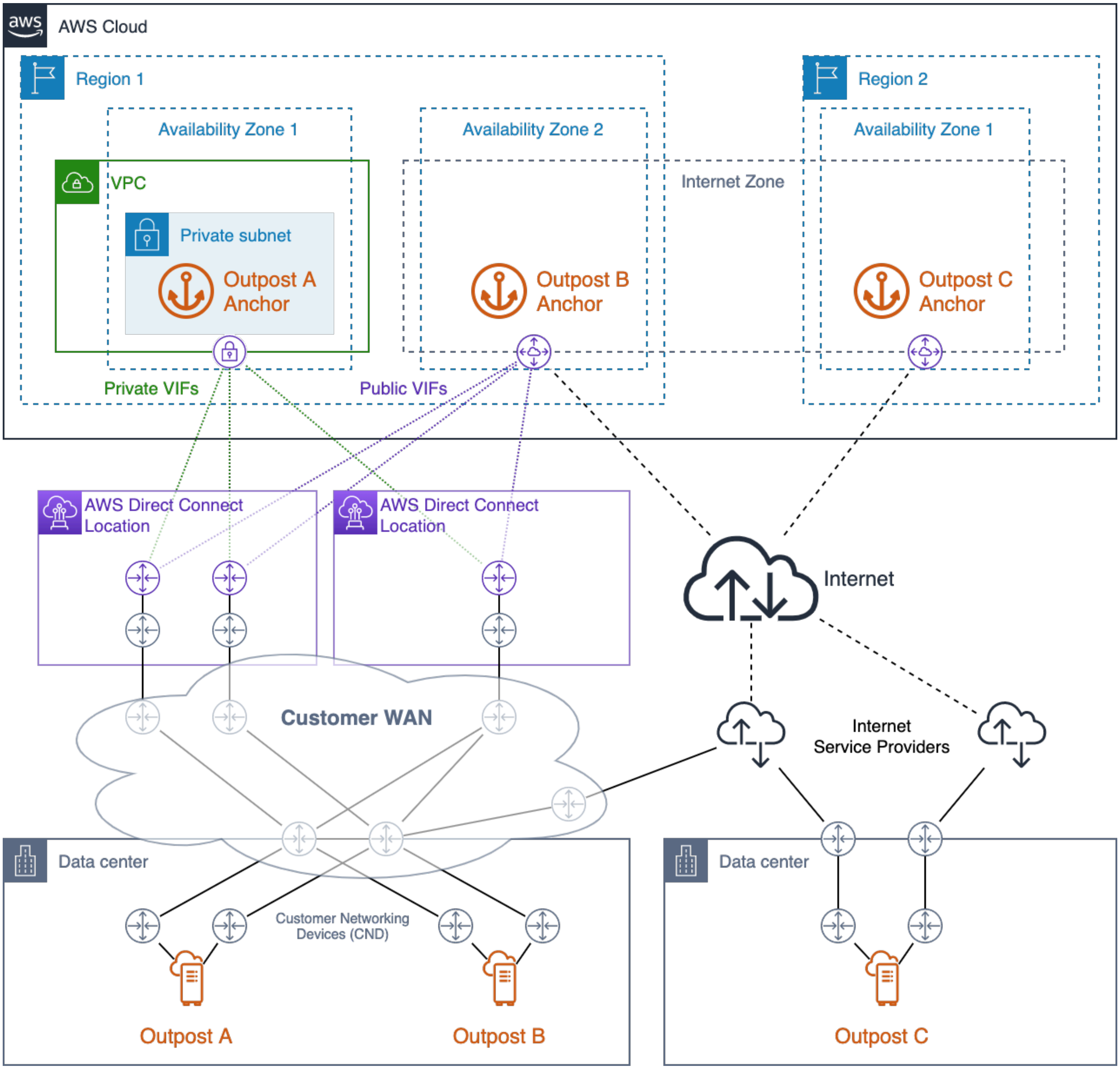
You must ensure the Outpost service link IP addresses can connect through your network to the anchor points in the anchor AZ. The service link IP addresses do not need to communicate with other hosts on your on-premises network.

Public anchor points reside in the Region's public IP ranges (in the EC2 service CIDR blocks) and may be accessed via the internet or AWS Direct Connect (DX) public virtual interfaces (VIFs). The use of public anchor points allows for more flexible path selection as service link traffic may be routed over any available path that can successfully reach the anchor points on the public internet.

Private anchor points allow you to use your IP address ranges for anchor connectivity. Private anchor points are created in a private subnet within a dedicated VPC using customer-assigned IP addresses. The VPC is created in the AWS account that owns the Outpost resource and you are responsible for ensuring the VPC is available and properly configured. Use a Security Control Policy (SCP) in AWSOrigamiServiceGateway Organizations to prevent users from deleting that Virtual Private Cloud (VPC).Private anchor points must be accessed using Direct Connect private VIFs.

You should provision redundant network paths between the Outpost and the anchor points in the Region with connections terminating on separate devices in more than one location. Dynamic routing should be configured to automatically reroute traffic to alternate paths when connections or networking devices fail. You should provision sufficient network capacity to ensure that the failure of one WAN path does not overwhelm the remaining paths.

The following diagram shows three Outposts with redundant network paths to their anchor AZs using AWS Direct Connect as well as public internet connectivity. Outpost A and Outpost B are anchored to different Availability Zones in the same Region. Outpost A connects to private anchor points in AZ 1 of region 1. Outpost B connects to public anchor points in AZ 2 of region 1. Outpost C connects to public anchors in AZ 1 of region 2.



Highly available anchor connectivity with AWS Direct Connect and public internet access

Outpost A has three redundant network paths to reach its private anchor point. Two paths are available through redundant Direct Connect circuits at a single Direct Connect location. The third path is available through a Direct Connect circuit at a second Direct Connect location. This design keeps Outpost A's service link traffic on private networks and provides path redundancy that allows for failure of any one of the Direct Connect circuits or failure of an entire Direct Connect location.

Outpost B has four redundant network paths to reach its public anchor point. Three paths are available through public VIFs provisioned on the Direct Connect circuits and locations used by Outpost A. The fourth path is available through the customer WAN and the public internet. Outpost B's service link traffic may be routed over any available path that can successfully reach the anchor points on the public internet. Using the Direct Connect paths may provide more consistent latency and higher bandwidth availability, while the public internet path may be used for Disaster Recovery (DR) or bandwidth augmentation scenarios.

Outpost C has two redundant network paths to reach its public anchor point. Outpost C is deployed in a different data center than Outposts A and B. Outpost C's data center does not have dedicated circuits connecting to the customer WAN. Instead, the data center has redundant internet connections provided by two different Internet Service Providers (ISPs). Outpost C's service link traffic may be routed over either of the ISP networks to reach the anchor points on the public internet. This design allows flexibility to route service link traffic over any available public internet connection. However, the end-to-end path is dependent on public third-party networks where bandwidth availability and network latency fluctuate.

The network path between an Outpost and its service link anchor points must meet the following bandwidth specification:

- 500 Mbps - 1 Gbps of available bandwidth per Outpost rack (for example, 3 racks: 1.5 – 3 Gbps available bandwidth)

Recommended practices for highly available anchor connectivity

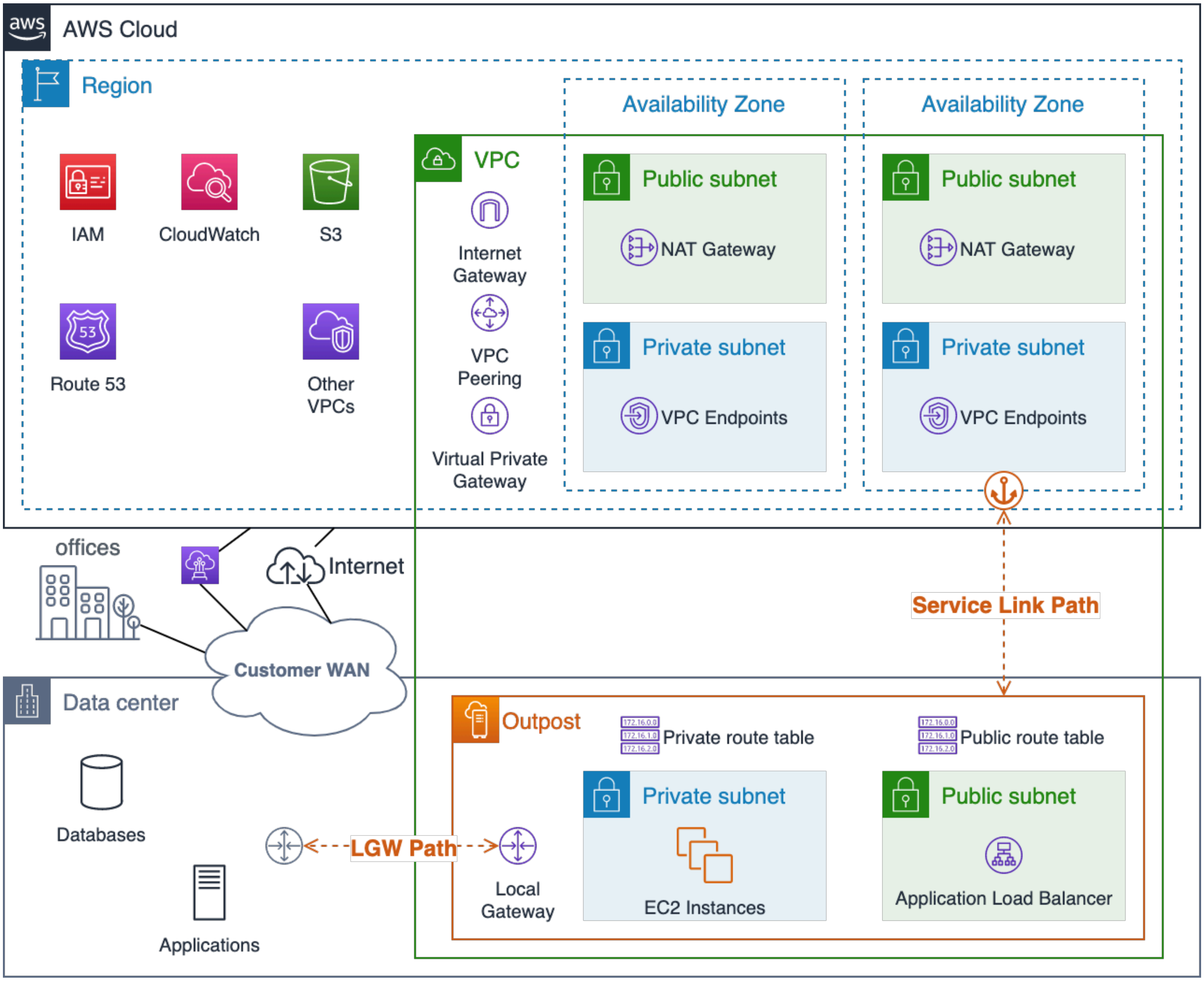
- Provision redundant network paths between each Outpost and its anchor points in the Region.
- Use Direct Connect (DX) paths to control latency and bandwidth availability.
- Ensure that TCP and UDP port 443 are open (outbound) from the Outpost Service Link CIDR blocks to the EC2 IP address ranges in the parent Region. Ensure the ports are open on all network paths.
- Keep track of the Amazon EC2 IP address ranges on your firewall if you are using a subset of CIDR ranges for the Region.
- Ensure each path meets the bandwidth availability and latency requirements.
- Use dynamic routing to automate traffic redirection around network failures.
- Test routing the service link traffic over each planned network path to ensure the path functions as expected.

Application/workload routing

There are two paths out of the Outpost for application workloads:

- The service link path: Consider that application traffic will compete with Outposts control plane traffic, in addition to limiting the MTU to 1300 bytes.
- The local gateway (LGW) path: Consider that the customer's local network allows access to both applications that are on-premises and also in the AWS Region.

You configure the Outpost subnet route tables to control which path to take to reach destination networks. Routes pointed to the LGW will direct traffic out the Local Gateway and to the on-premises network. Routes pointed to the services and resources in the Region, such as Internet Gateway, NAT Gateway, Virtual Private Gateway, and TGW, will use service link to reach these targets. If you have a VPC peering connection with multiple VPCs on the same Outpost, the traffic between the VPCs remains on the Outpost and doesn't use the service link back to the Region. For information on VPC peering, see Connect VPCs using VPC peering in the *Amazon VPC User Guide*.



Visualization of the Outpost service link and LGW network paths

You should take care when planning application routing to consider both normal operation and limited routing and service availability during network failures. The Service Link path is not available when an Outpost is disconnected from the Region.

You should provision diverse paths and configure dynamic routing between the Outpost LGW and your critical on-premises applications, systems and users. Redundant network paths allow the network to route traffic around failures and ensure that on-premises resources will be able to communicate with workloads running on the Outpost during partial network failures.

Outpost VPC route configurations are static. You configure subnet routing tables through the AWS Management Console, CLI, APIs, and other Infrastructure as Code (IaC) tools; however, you will not be able modify the subnet routing tables during a disconnect event. You will have to reestablish connectivity between the Outpost and the Region to update the route tables. Use the same routes for normal operations as you plan to use during disconnect events.

Resources on the Outpost can reach the internet via the service link and an Internet Gateway (IGW) in the Region or via the Local Gateway (LGW) path. Routing internet traffic over the LGW path and the on-premises network allows you to use existing on-premises internet ingress/egress points and may provide lower latency, higher MTUs, and reduced AWS data egress charges when compared to using the service link path to an IGW in the Region.

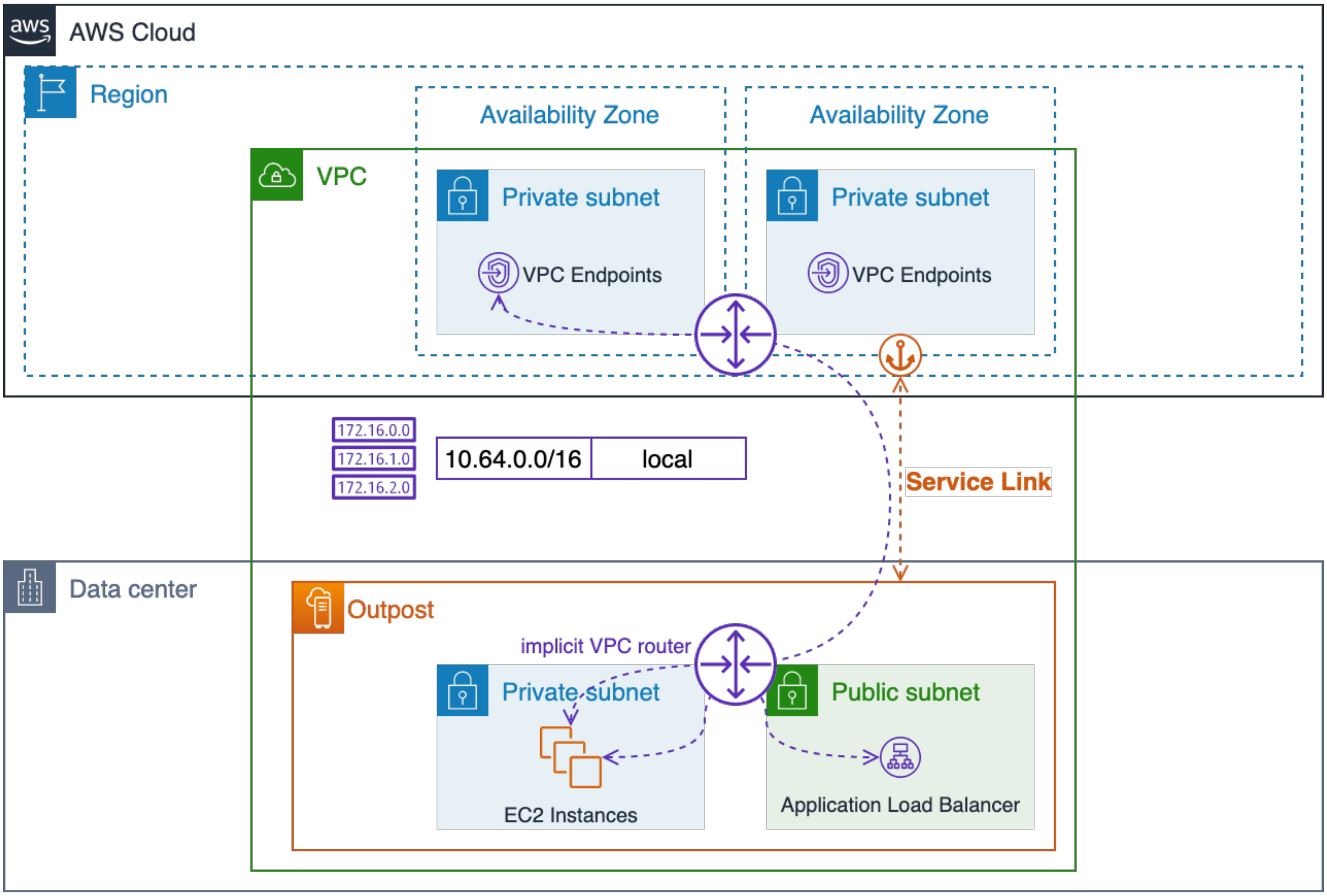
If your application must run on-premises and it needs to be accessible from the public internet, you should route the application traffic over your on-premises internet connection(s) to the LGW to reach the resources on the Outpost.

While you can configure subnets on an Outpost like public subnets in the Region, this may be an undesirable practice for most use cases. Inbound internet traffic will come in through the AWS Region and be routed over the service link to the resources running on the Outpost.

The response traffic will in turn be routed over the service link and back out through the AWS Region’s internet connections. This traffic pattern may add latency and will incur data egress charges as traffic leaves the Region on its way to the Outpost and as return traffic comes back through the Region and egresses out to the internet. If your application can run in the Region, the Region is the best place to run it.

Traffic between VPC resources (in the same VPC) will always follow the local VPC CIDR route and be routed between subnets by the implicit VPC routers.

For example, traffic between an EC2 instance running on the Outpost and a VPC Endpoint in the Region will always be routed over the service link.



Local VPC routing through the implicit routers

Recommended practices for application/workload routing

- Use the Local Gateway (LGW) path instead of the service link path where possible.
- Route internet traffic over the LGW path.
- Configure the Outpost subnet routing tables with a standard set of routes – they will be used for both normal operations and during disconnect events.
- Provision redundant network paths between the Outpost LGW and critical on-premises application resources. Use dynamic routing to automate traffic redirection around on-premises network failures.

Compute

While Amazon EC2 capacity in AWS Regions is seemingly infinite, capacity on Outposts is finite. You are responsible for planning and managing the compute capacity of your Outposts deployments.

Topics

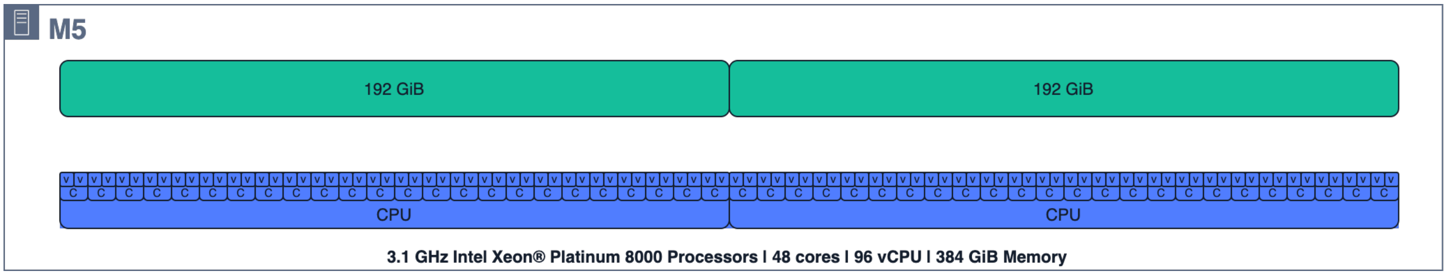
- Capacity planning
- Capacity management
- Instance placement

Capacity planning

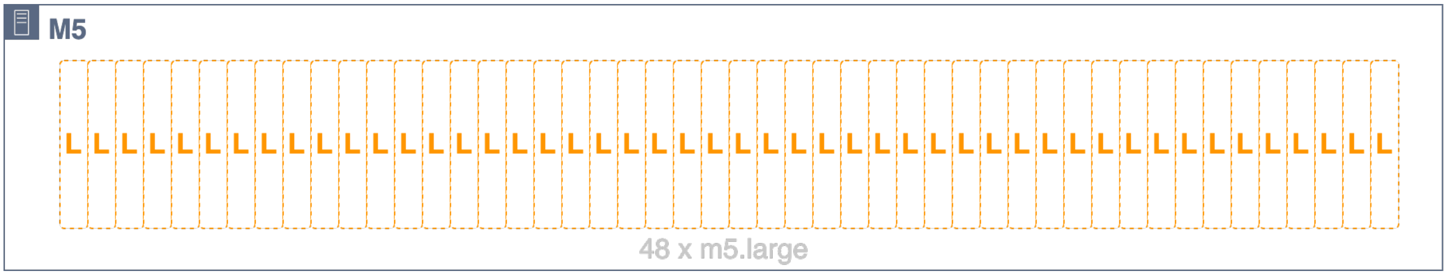
While Amazon EC2 capacity in AWS Regions is seemingly infinite, capacity on Outposts is finite – constrained by the total volume of compute capacity ordered. You are responsible for planning and managing the compute capacity of your Outposts deployments. You should order sufficient compute capacity to support an N+M availability model, where N is the required number of servers and M is the number of spare servers provisioned to accommodate server failures. N+1 and N+2 are the most common availability levels.

Each host (`C5`, `M5`, `R5`, etc.) supports a single family of EC2 instances. Before you can launch instances on EC2 compute servers, you must provide slotting layouts that specify the EC2 instance sizes you want each server to provide. AWS configures each server with the requested slotting layout.

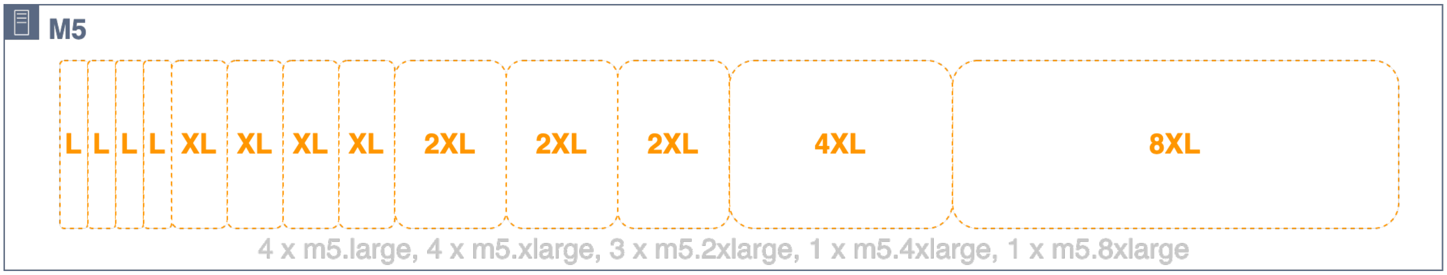
Hosts may be homogeneously slotted where all slots are the same instance size (for example, 48 `m5.large` slots) or heterogeneously slotted with a mixture of instances types (for example, 4 `m5.large`, 4 `m5.xlarge`, 3 `m5.2xlarge`, 1 `m5.4xlarge`, and 1 `m5.8xlarge`) – see the next three figures for visualizations of these slotting configurations.


M5
192 GiB
192 GiB
CPU
CPU
3.1 GHz Intel Xeon® Platinum 8000 Processors I 48 cores I 96 vCPU I 384 GiB Memory

`m5.24xlarge` host compute resources


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

`m5.24xlarge` host homogenously slotted into 48 `m5.large` slots


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

`m5.24xlarge` host heterogeneously slotted into 4 `m5.large`, 4 `m5.xlarge`, 3 `m5.2xlarge`, 1 `m5.4xlarge`, and 1 `m5.8xlarge` slots

The full host capacity does not have to be slotted. Slots may be added to a host that has available unallocated capacity. You can modify a slotting layout by using the Capacity Management APIs or UIs for AWS Outposts and creating a new capacity task. For more information, see Capacity management for AWS Outposts in the *AWS Outposts user guide for racks*. You may be required to shut down or restart certain instances to complete a new capacity task if the new slotting layout cannot be applied while certain slots are occupied by running instances. The `CreateCapacityTask` API allows you to express the number of each instance size that should be present on the Outpost ID indicated, and in the event that a task cannot be completed due to running instances, returns instances that must be stopped in order to satisfy the request. At this point, you can optionally indicate that you want to see “N” additional options in the event that you would prefer not to stop one of the instances returned, and you can also indicate an EC2 instance ID, EC2 instance tag, account, or service that should not be suggested as an instance to shut down in order to satisfy the capacity task request. After making your selection of the option you’d like to go with, we recommend using the Dry Run parameter to validate the proposed changes and understand the potential impact before implementing.

All hosts contribute their provisioned slots to the EC2 capacity pools on the Outpost, and all slots of a given instance type and size are managed as a single EC2 capacity pool. For example, the previous heterogeneously slotted host with `m5.large`, `m5.xlarge`, `m5.2xlarge`, `m5.4xlarge`, and m5.8xlarge slots would contribute these slots to five EC2 capacity pools – one pool for each instance type and size. These pools may be spread across multiple hosts, and instance placement should be a consideration to achieve workload high availability.

It is important to consider host slotting and EC2 capacity pools when planning spare capacity for N+M host availability. AWS detects when a host fails or is degraded and schedules a site visit to replace the failed host. You should design your EC2 capacity pools to tolerate the failure of at least one server of each instance family (N+1) in an Outpost. With this minimum level of host availability, when a host fails or needs to be taken out of service, you can restart failed or degraded instances on the spare slots of the remaining hosts of the same family.

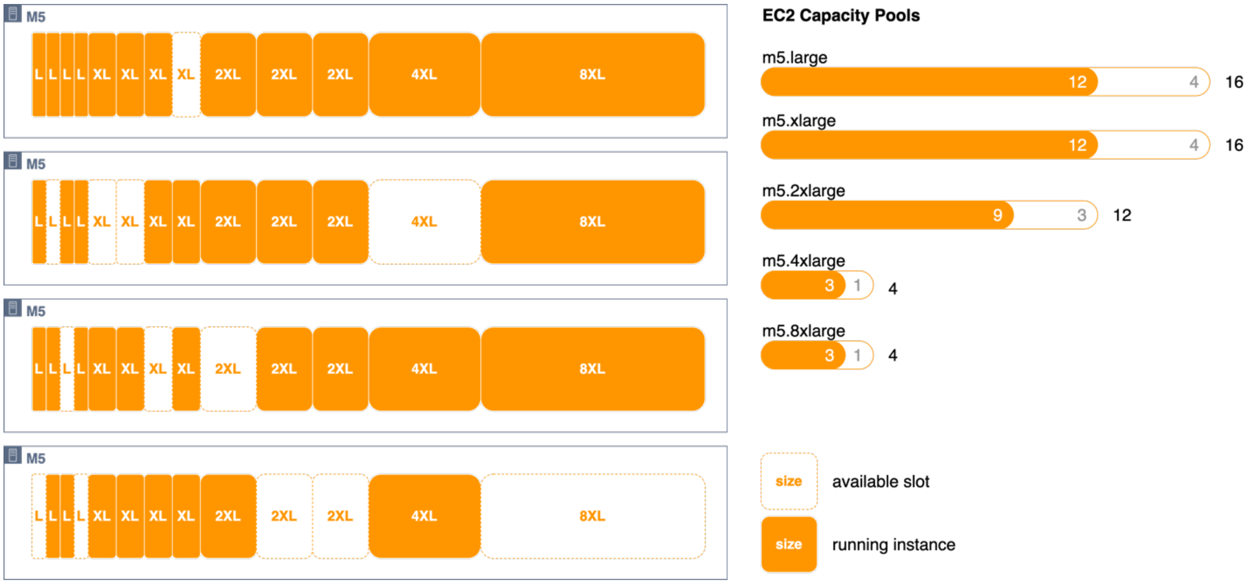
Planning for N+M availability is simple when you have homogenously slotted hosts or groups of heterogeneously slotted hosts with identical slotting layouts. You simply calculate the number of hosts (N) your need to run all your workloads and then add (M) additional hosts to meet your requirements for server availability during failure and maintenance events.

The following slotting configurations are not usable due to NUMA boundaries:

- 3 m5.8xlarge
- 1 m5.16xlarge and 1 m5.8xlarge

Consult your AWS account team to validate your planned AWS Outposts rack slotting configuration.

In the following figure, four m5.24xlarge hosts are heterogeneously slotted with an identical slotting layout. The four hosts create five EC2 capacity pools. Each pool is running at maximum utilization (75%) to maintain N+1 availability for the instances running on these four hosts. If any host fails, there is sufficient room to restart the failed instances on the remaining hosts.


M5
L L L L XL XL XL XL 2XL 2XL 2XL 4XL 8XL
M5
L L L L XL XL XL XL 2XL 2XL 2XL 4XL 8XL
M5
L L L L XL XL XL XL 2XL 2XL 2XL 4XL 8XL
M5
L L L L XL XL XL XL 2XL 2XL 2XL 4XL 8XL
EC2 Capacity Pools
m5.large 12 4 16
m5.xlarge 12 4 16
m5.2xlarge 9 3 12
m5.4xlarge 3 1 4
m5.8xlarge 3 1 4
size available slot
size running instance

Visualization of EC2 host slots, running instances, and slot pools

For more complex slotting layouts, where hosts are not identically slotted, you will need to calculate N+M availability for each EC2 capacity pool. You can use the following formula to calculate how many hosts (that contribute slots to a given EC2 capacity pool) can fail and still allow the remaining hosts to carry the running instances:

$$M = \left\lfloor \frac{poolSlots_{available}}{serverSlots_{max}} \right\rfloor$$

Where:

- **poolSlots$_{\textbf{available}}$** is the number of available slots in the given EC2 capacity pool (total number of slots in the pool minus the number of running instances)
- **serverSlots$_{\textbf{max}}$** is the maximum number of slots contributed by any host to the given EC2 capacity pool
- **M** is the number of hosts that can fail and still allow the remaining hosts to carry the running instances

Example: An Outpost has three hosts that contribute slots to an `m5.2xlarge` capacity pool. The first contributes 4 slots, the second contributes 3 slots, and the third host contributes 2 slots. The `m5.2xlarge` instance pool on the Outpost has a total capacity of 9 slots (4 + 3 + 2). The Outpost has 4 running `m5.2xlarge` instances. How many hosts may fail and still allow the remaining hosts to carry the running instances?

$$poolSlots_{available} = total\ capacity - running\ instances = 9 - 4 = 5$$

$$serverSlots_{max} = \max([4, 3, 2]) = 4$$

$$M = \left[\frac{poolSlots_{available}}{serverSlots_{max}}\right] = \left[\frac{5}{4}\right] = [1.25] = 1$$

Answer: You can lose any one of the hosts and still carry the running instances on the remaining hosts.

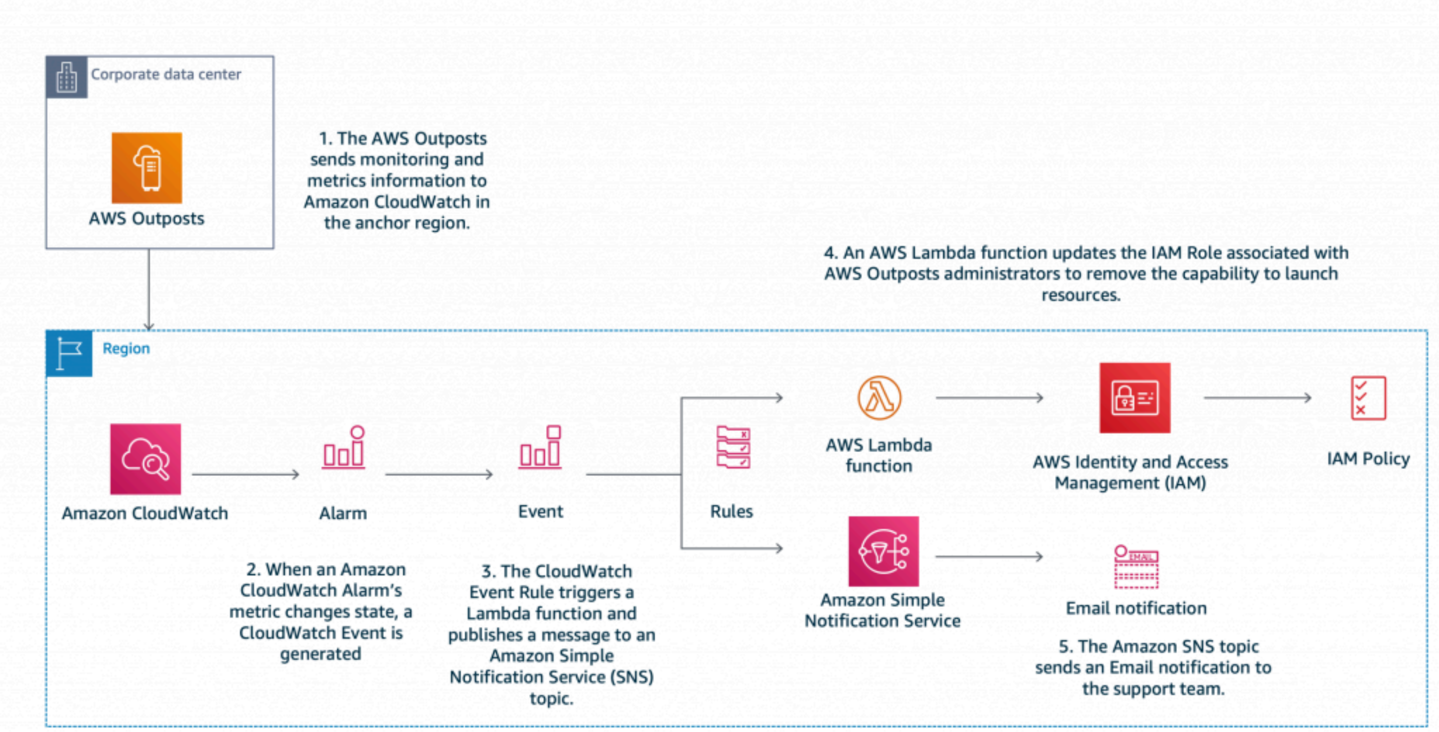
Recommended practices for compute capacity planning

- Size your compute capacity to provide N+M redundancy for each EC2 capacity pool on an Outpost.
- Deploy N+M servers for homogeneously or identical heterogeneously slotted servers.
- Calculate the N+M availability for each EC2 capacity pool and ensure that each pool meets your availability requirements.
- Consider size compute capacity for use cases that require ALB on AWS Outposts and Amazon Route 53 local Resolver.

Capacity management

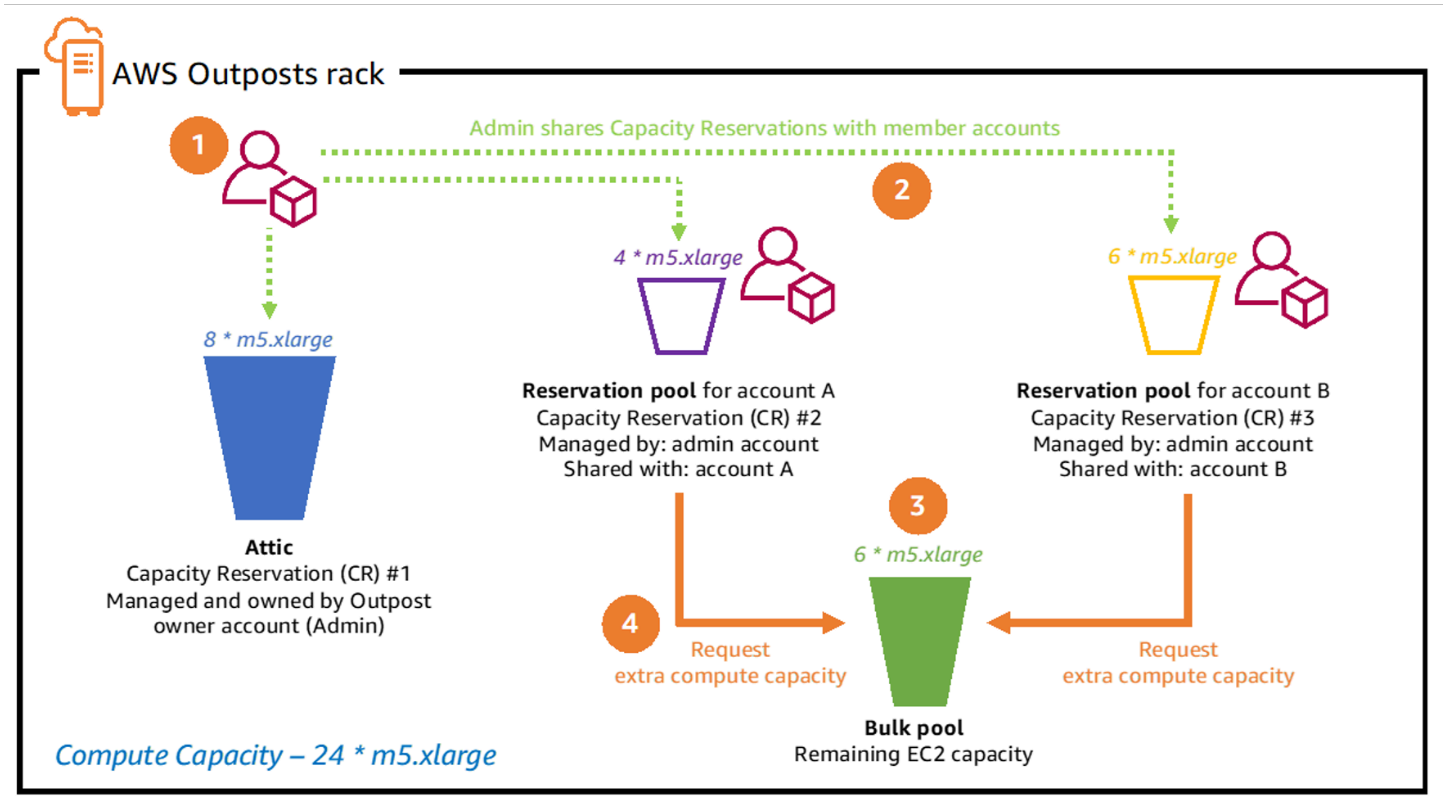
You can monitor Outpost EC2 instance pool utilization in the AWS Management Console and via Amazon CloudWatch metrics. Contact Enterprise Support to retrieve or change the slotting layouts for your Outposts.

You use the same instance auto recovery and EC2 Auto Scaling mechanisms to recover or replace instances impacted by server failures and maintenance events. You must monitor and manage your Outpost capacity to ensure sufficient spare capacity is always available to accommodate server failures. The Managing your AWS Outposts capacity using Amazon CloudWatch and AWS Lambda blog post provides a hands-on tutorial showing you how to combine AWS CloudWatch and AWS Lambda to manage your Outpost capacity to maintain instance availability.

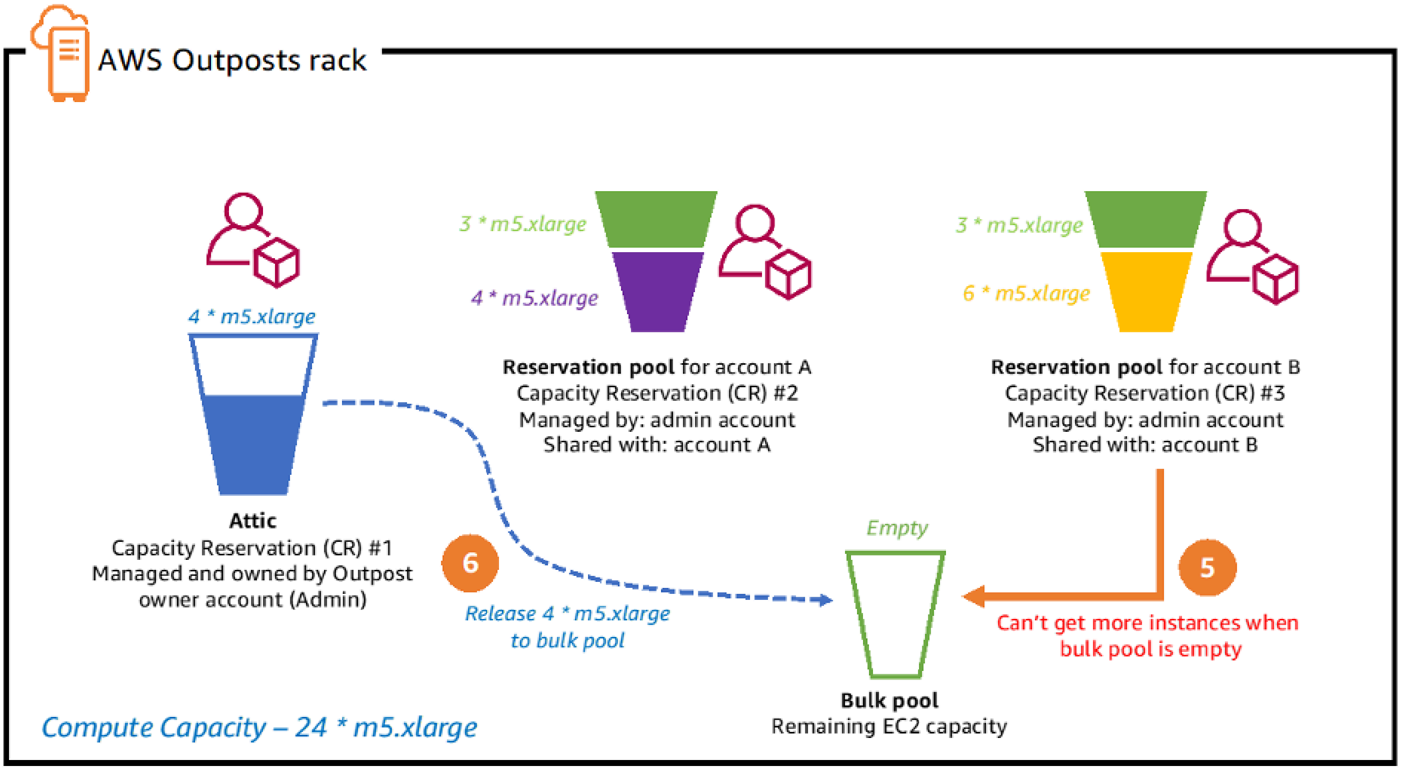

Corporate data center
AWS Outposts
1. The AWS Outposts sends monitoring and metrics information to Amazon CloudWatch in the anchor region.
4. An AWS Lambda function updates the IAM Role associated with AWS Outposts administrators to remove the capability to launch resources.
Region
Amazon CloudWatch
Alarm
Event
Rules
AWS Lambda function
AWS Identity and Access Management (IAM)
IAM Policy
Amazon Simple Notification Service
Email notification
2. When an Amazon CloudWatch Alarm's metric changes state, a CloudWatch Event is generated
3. The CloudWatch Event Rule triggers a Lambda function and publishes a message to an Amazon Simple Notification Service (SNS) topic.
5. The Amazon SNS topic sends an Email notification to the support team.

Managing AWS Outposts capacity with Amazon CloudWatch and AWS Lambda

Capacity Reservations can be used in a multi-account environment to control how much of your Outpost compute capacity is used by a single account, or an AWS Organization unit (OU) containing multiple accounts. You can create a capacity reservation for Amazon EC2 on Outposts, as well as supported Outposts AWS services such as Amazon Elastic Kubernetes Service (EKS, Amazon Elastic Container Service (ECS), and Amazon Elastic Map Reduce (EMR). Capacity reservations are created and shared to accounts through AWS Resource Access Manager (AWS RAM) in the Outpost owner account. The Creating computing quotas on AWS Outposts rack with EC2 Capacity Reservations sharing provides a hands-on tutorial and additional guidance for implementing capacity reservations with your Outpost for the purpose of capacity management.


AWS Outposts rack
1
Admin shares Capacity Reservations with member accounts
2
8 * m5.xlarge
4 * m5.xlarge
6 * m5.xlarge
Reservation pool for account A
Capacity Reservation (CR) #2
Managed by: admin account
Shared with: account A
Reservation pool for account B
Capacity Reservation (CR) #3
Managed by: admin account
Shared with: account B
3
6 * m5.xlarge
Attic
Capacity Reservation (CR) #1
Managed and owned by Outpost owner account (Admin)
4
Request extra compute capacity
Request extra compute capacity
Bulk pool
Remaining EC2 capacity
Compute Capacity – 24 * m5.xlarge

Capacity Reservation sharing process steps 1-4


AWS Outposts rack
3 * m5.xlarge
4 * m5.xlarge
3 * m5.xlarge
6 * m5.xlarge
4 * m5.xlarge
Reservation pool for account A
Capacity Reservation (CR) #2
Managed by: admin account
Shared with: account A
Reservation pool for account B
Capacity Reservation (CR) #3
Managed by: admin account
Shared with: account B
Attic
Capacity Reservation (CR) #1
Managed and owned by Outpost owner account (Admin)
6
Release 4 * m5.xlarge to bulk pool
Empty
5
Can't get more instances when bulk pool is empty
Bulk pool
Remaining EC2 capacity
Compute Capacity – 24 * m5.xlarge

Capacity Reservation sharing process steps 5-6

Recommended practices for compute capacity management

- Configure your EC2 instances in Auto Scaling groups or use instance auto recovery to restart failed instances.
- Automate capacity monitoring for your Outpost deployments and configure notifications and (optionally) automated responses for capacity alarms.
- Use Capacity Reservations to have granular control over how much compute capacity is shared to other accounts within your AWS Organization.

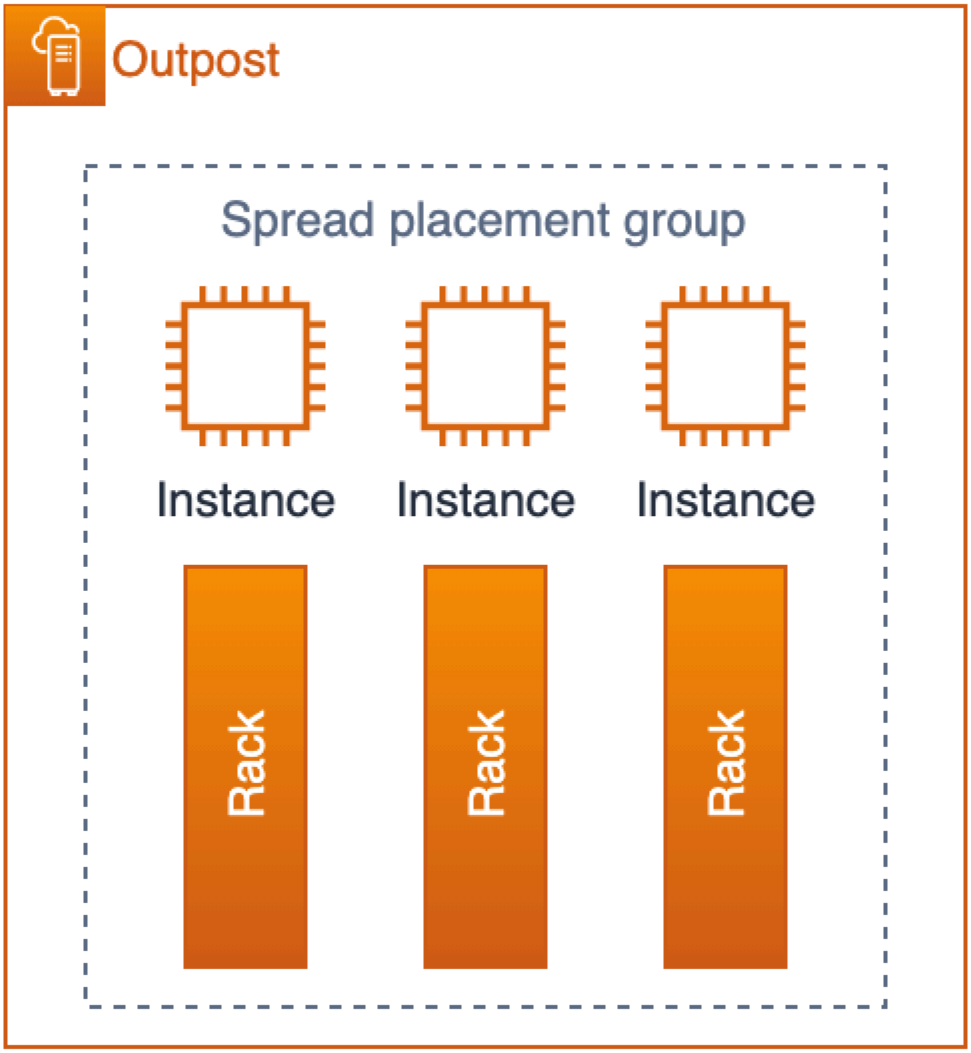
Instance placement

Outposts have a finite number of compute hosts. If your application deploys multiple related instances on Outposts; without additional configuration, the instances may deploy on the same hosts or on hosts in the same rack. Today, there are three mechanisms you may use to distribute instances to mitigate the risk of running related instances on the same infrastructure:

Multi-Outpost deployment – similar to a multi-AZ strategy in the Region, you can deploy Outposts to separate data centers and deploy application resources to specific Outposts. This allows you to run instances on the desired Outpost (a logical set of racks). Intra-VPC Communication Across Multiple Outposts with Direct VPC Routing is another strategy that can be used to spread workloads across multiple Outposts within the same VPC using the Outpost local gateways (LGW) to create routes between the subnets on the Outposts. A multi-Outpost strategy may be employed to protect against rack and data center failure modes and, if the Outposts are anchored to separate AZs or Regions, may also provide protection against AZ or Region failure modes. For more information about multi-Outpost architectures, see the Larger Failure Modes.

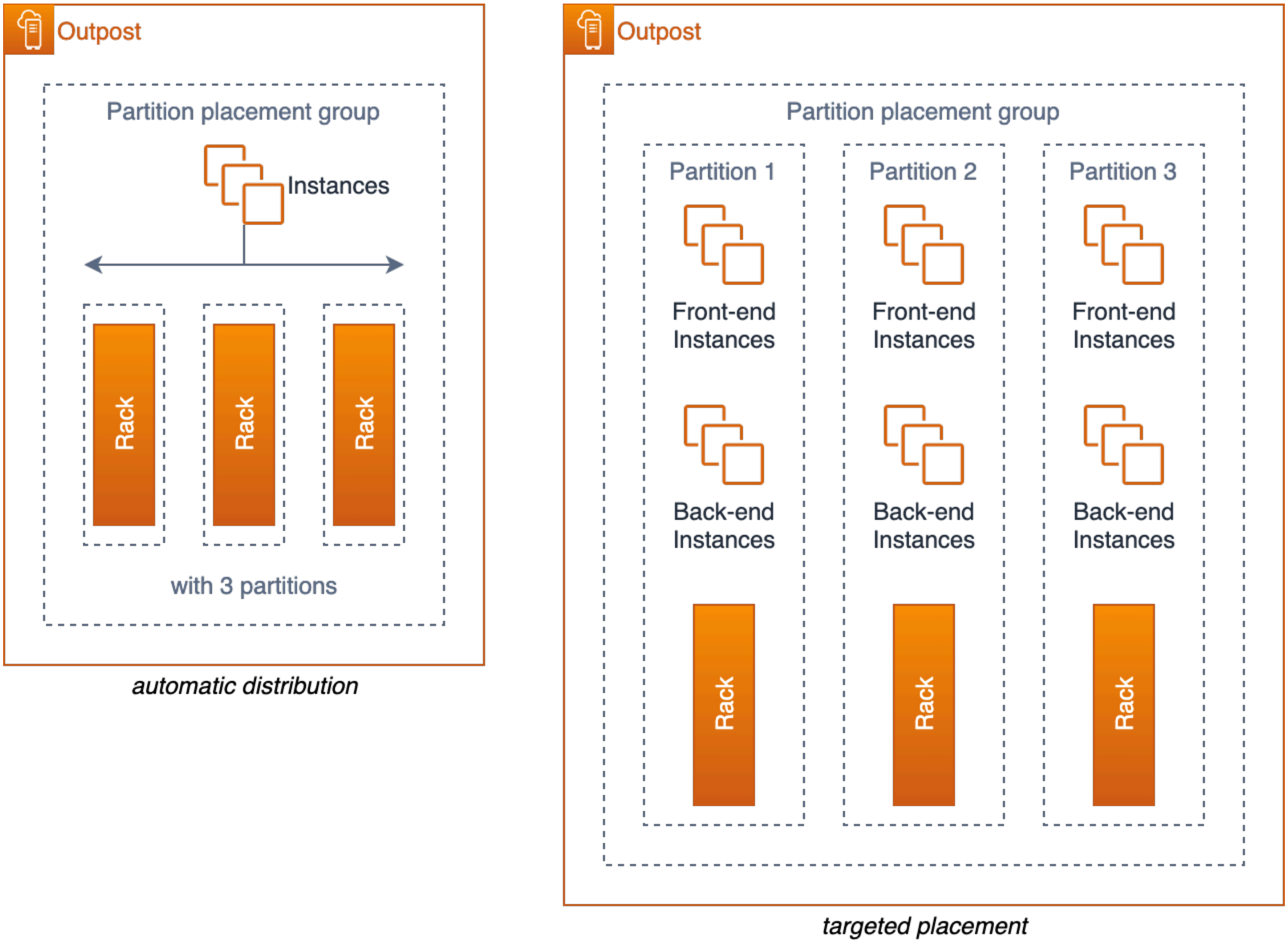
Amazon EC2 placement groups on Outposts (single-Outpost multi-rack instance placement) – You can create placement groups on Outposts that you have created in your account. This allows you to spread out instances across underlying hardware on an Outpost at your site. When you create a placement group with a spread strategy on an Outpost, you can choose to have the placement group spread instances across hosts or racks.

A spread placement group provides a simple way distribute single instances across racks or hosts to reduce the potential for correlated failures. You may only deploy into the group as many instances as you have hosts in your Outpost.



EC2 spread placement group on an Outpost with three racks

You can also distribute instances across multiple racks with partition placement groups. Use automatic distribution to spread instances across partitions in the group or deploy instances to selected target partitions. Deploying instances to target partitions allows you to deploy selected resources to the same rack while distributing other resources across racks. For example, if you have a logical Outpost with three racks, creating a partition placement group with three partitions allows you to distribute resources across the racks.



automatic distribution

targeted placement

EC2 partition placement groups on an Outpost with three racks

Creative server slotting – if you have a single-rack Outpost or if the service you are using on Outposts does not support placement groups, you may be able to use creative slotting to ensure your instances do not deploy on the same physical server. If the related instances are the same EC2 instance size, you may be able to slot your servers to limit the number of slots of that size configured on each server – spreading the slots across the servers. Server slotting will limit the number of instances (of that size) that can run on a single server.

As an example, consider the slotting layout shown previously in Figure 13. If your application needed to deploy three `m5.4xlarge` instances on the Outpost configured with this slotting layout, EC2 would place each instance on a separate server and there would be no possibility that these instances could run on the same server – as long as the slotting configuration does not change to open additional `m5.4xlarge` slots on the servers.

Recommended practices for compute instance placement

- Use Amazon EC2 placement groups on Outposts to control placement of instances across racks within a single logical Outpost.
- Instead of ordering an Outpost with a single medium or large Outpost rack, consider splitting the capacity into two small or medium racks to allow you to take advantage of the EC2 placement groups ability to distribute instances across racks.
- Amazon EC2 Placement group on Outposts can be used to influence the placement of EKS nodegroups, Control Plane Nodes for EKS Local Cluster and ECS Task.
- Use Intra-VPC Communication to spread workloads across multiple Outposts within the same VPC.

Storage

The AWS Outposts rack service provides three storage types:

- Instance storage on supported EC2 instance types
- Amazon Elastic Block Store (EBS) gp2 volumes for persistent block storage
- Amazon Simple Storage Service on Outposts (S3 on Outposts) for local object storage

Instance storage is provided on supported servers (`C5d`, `M5d`, `R5d`, `G4dn`, and `I3en`). Just like in the Region, the data in an instance store persists only for the (running) lifetime of the instance.

Outposts EBS volumes and S3 on Outposts object storage are provided as part of the AWS Outposts rack managed services. Customers are responsible for capacity management of the Outpost storage pools. Customers specify their storage requirements for EBS and S3 storage when ordering an Outpost. AWS configures the Outpost with the number of storage servers required to provide the requested storage capacity. AWS is responsible for the availability of the EBS and S3 on Outposts storage services. Sufficient storage servers are provisioned to provide highly available storage services to the Outpost. Loss of a single storage server should not disrupt the services nor result in data loss.

You can use the AWS Management Console and CloudWatch metrics to monitor Outpost EBS and S3 on Outposts capacity utilization.

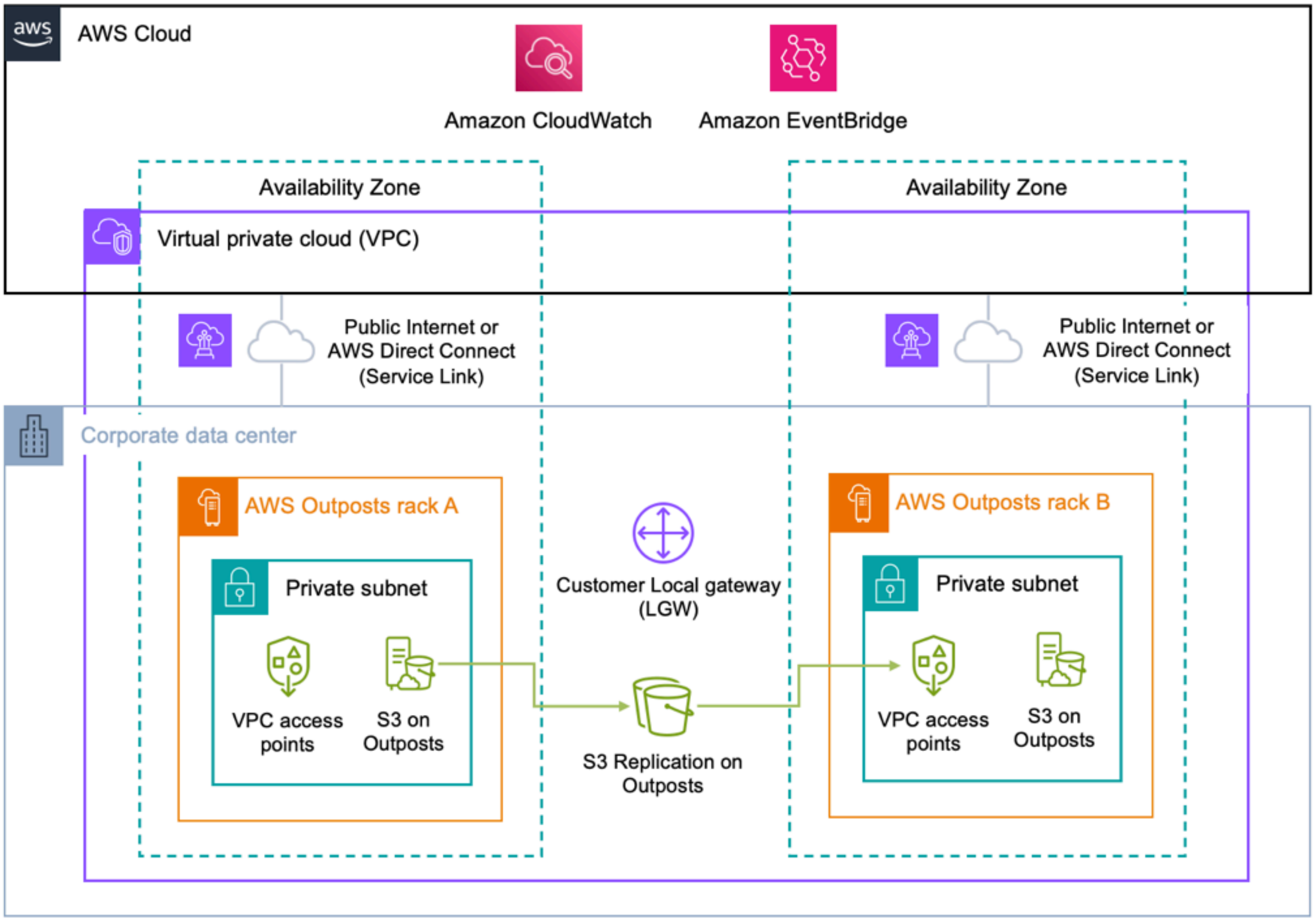
Data protection

For EBS Volumes: AWS Outposts rack supports EBS volume snapshots to provide a simple and secure data protection mechanism to protect your block storage data. Snapshots are point-in-time incremental backups of your EBS volumes. By default, snapshots of Amazon EBS volumes on your Outpost are stored on Amazon S3 in the Region. If your Outposts have been configured with S3 on Outposts capacity, you can use EBS Local Snapshots on Outposts to store snapshots locally on your Outpost using S3 on Outposts storage.

For S3 on Outposts buckets (data residency use cases):

- You can use S3 Versioning on Outposts, to save all changes, and history of objects. When enabled, S3 Versioning saves multiple distinct copies of an object in the same bucket. You can use S3 Versioning to preserve, retrieve, and restore every version of every object stored in your Outposts buckets. S3 Versioning helps you recover from unintended user actions and application failures.
- You can use S3 Replication on Outposts, to create and configure replication rules to automatically replicate your S3 objects to another Outpost, or to another bucket on the same Outpost. During replication, S3 on Outposts objects are sent over the customer's local gateway (LGW), and objects do not travel back to the AWS Region. S3 Replication on Outposts provides an easy and flexible way to automatically replicate data within a specific data perimeter to address data redundancy and compliance requirements.

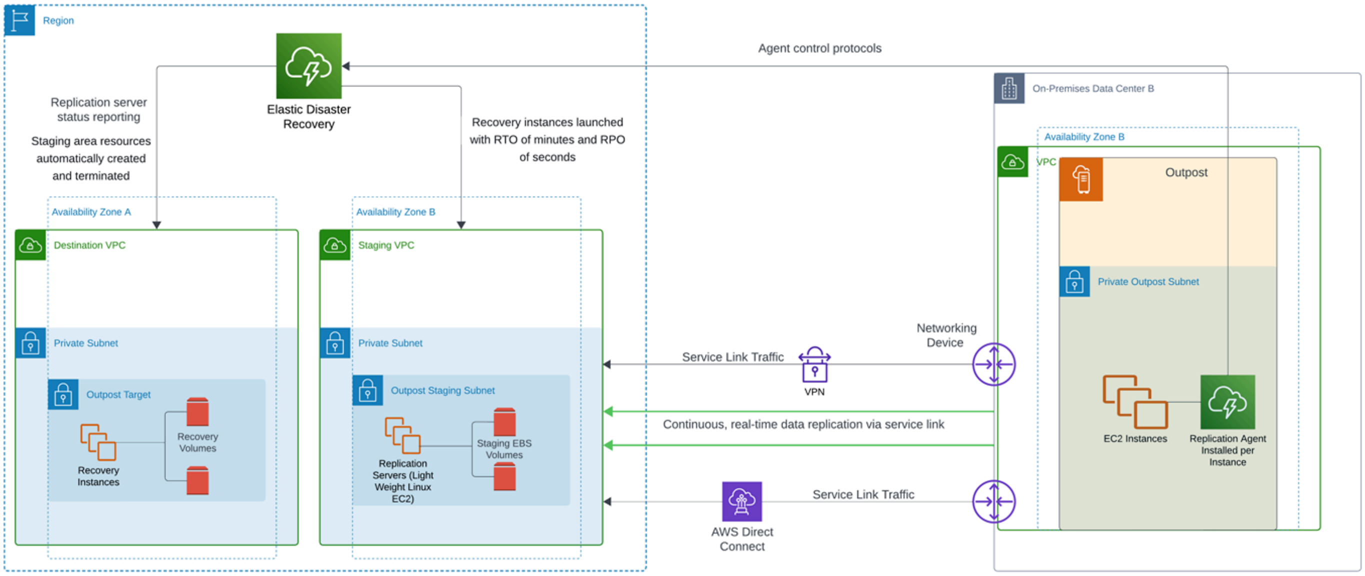
  S3 Replication on Outposts also provides detailed metrics and notifications to monitor the status of your object replication. You can monitor replication progress by tracking bytes pending, operations pending, and replication latency between your source and destination Outposts buckets using Amazon CloudWatch. You can also set up Amazon EventBridge rules to receive replication failure events to quickly diagnose and correct configuration issues. See the Amazon S3 Replication on Outposts YouTube video for additional details on how to configure.



Amazon S3 Replication on Outposts

For S3 on Outposts buckets (non-data residency use cases) to AWS Regions: You can use AWS DataSync to automate Amazon S3 on Outposts data transfers between your Outpost and the Region. DataSync allows you to choose what to transfer, when to transfer, and how much bandwidth to use. Backing up your on-premises S3 on Outposts buckets to S3 buckets in the AWS Region allows you to leverage the 99.999999999% (11 9's) of data durability and additional storage tiers (Standard, Infrequent Access, and Glacier) for cost optimization available with the regional S3 service.

Instance replication: You can use use AWS Elastic Disaster Recovery (AWS DRS) to replicate individual instances and attached block storage from on-premises systems to an Outpost, from an Outpost to the Region, from the Region to an Outpost, or from one Outpost to another Outpost. The Architecting for Disaster Recovery on AWS Outposts Racks with AWS Elastic Disaster Recovery blog post describes each of these scenarios and how to design a solution with AWS DRS.



Disaster recovery (DR) from an Outpost to the Region

Using AWS Outposts rack as an AWS DRS destination (replication target) requires S3 on Outposts storage, which is used for the purpose of storing replicated Amazon EBS snapshots. S3 on Outposts storage is also required on the source Outposts for failback. The Outposts rack must be using Direct VPC Routing (DVR) to use AWS DRS. AWS DRS cannot be used to protect managed service instances on Outposts, it is only supported for disaster recovery of EC2 instances and their attached EBS volumes.

Recommended practices for data protection:

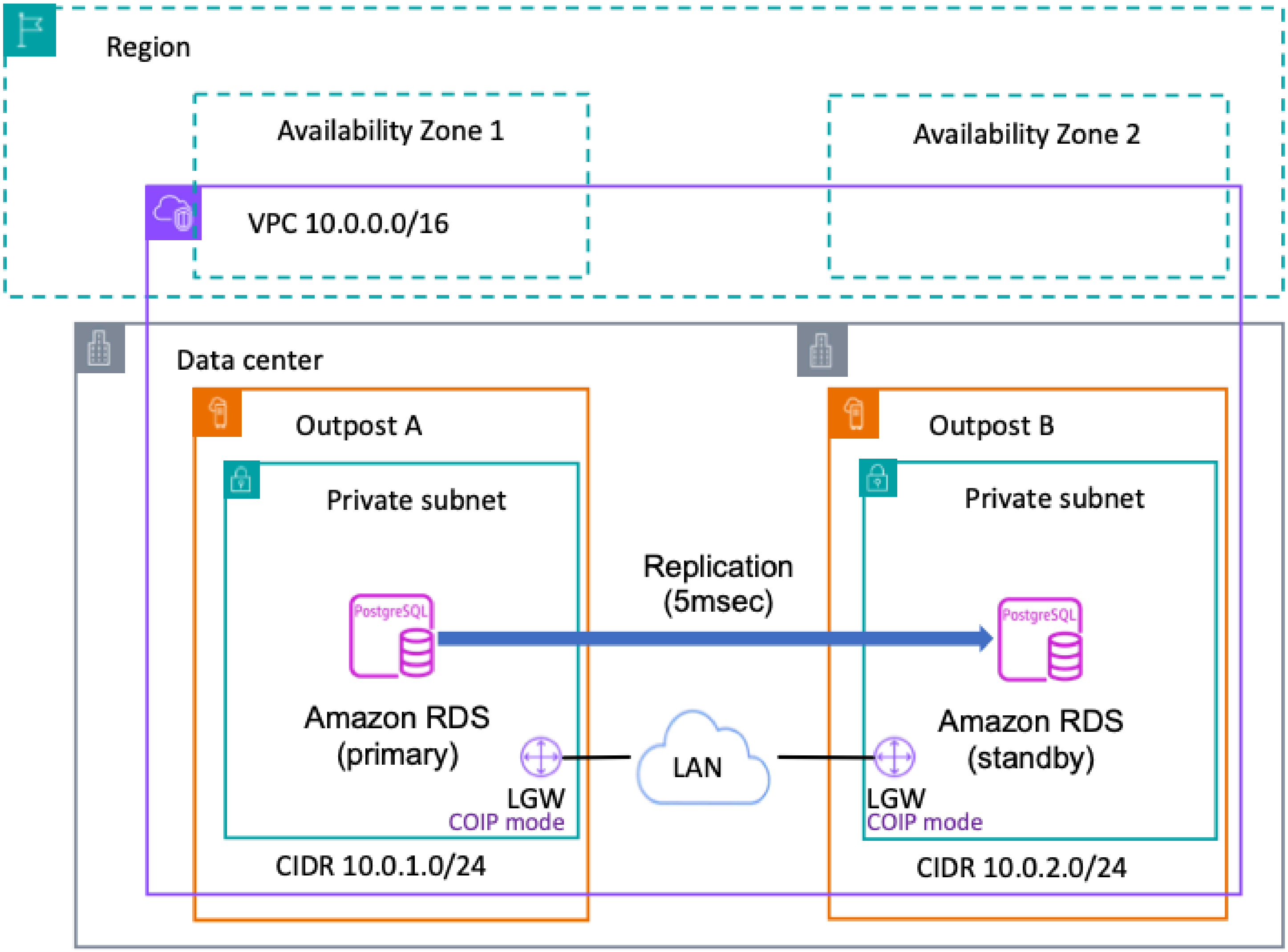
- Use EBS snapshots to create point-in-time backups of block storage volumes to Amazon S3 in the Region or S3 on Outposts.
- Use S3 on Outposts object versioning to maintain multiple versions and history of your objects.
- Use S3 Replication on Outposts to automatically replicate your object data to another Outpost.
- For non-data residency use cases, use AWS DataSync to back up objects stored in S3 on Outpost to Amazon S3 in the Region.
- Use AWS DRS to replicate instances between on-premises systems, logical Outposts, and the Region.

Databases

Amazon Relational Database Service (RDS) on AWS Outposts extends RDS for SQL Server, RDS for MySQL and RDS for PostgreSQL databases to AWS Outposts deployments. For those deployments where a highly available architecture must be provided, Amazon RDS support Multi-AZ instances deployment for PostgreSQL and MySQL on AWS Outposts.

Amazon RDS on Outposts with Multi-AZ

In Multi-AZ deployments, Amazon RDS creates a primary DB instance on one AWS Outposts and RDS synchronously replicates the data to a standby DB instance on a different Outposts. In order to provide a resilient architecture, the two AWS Outposts must be anchored to different Availability Zones in a given region and must be operating on Customer-owned IP (CoIP) model. In order to allow the replication between the primary instance and the standby, there must be a network link between the two Outposts with a round-trip time (RTT) latency of single-digit milliseconds. We recommend 5 milliseconds or less. Also consider sizing the replication link between Outposts with sufficient bandwidth to avoid queuing replication jobs.



Amazon RDS on Outpost with multi AZ

Considerations for Amazon RDS on Outposts with Multi-AZ

Review the following considerations for Amazon RDS on Outposts deployments in Multi-AZ:

- Have at least two Outposts deployments anchored to different Availability Zones in the same AWS Region.
- Both primary and standby instance requires a single VPC and one subnet per Outposts deployment.
- Associate your DB instance's VPC with all of your local gateway route tables.
- Make sure that your Outposts use customer-owned IP routing.

- Your local network must allow outbound and related inbound traffic between Outposts for Internet Security Association and Key Management Protocol (ISAKAMP) which use UPD port 500 and IPsec Network Address Translation Traversal (NAT-T) using UDP port 4500.
- Local RDS backups are not supported for Multi-AZ deployments.
- If your workload must abide by data residency regulations for your industry or geography, consult with regulators to determine if Multi-AZ RDS will meet your requirements.

For more details see Working with Multi-AZ deployments for Amazon RDS on AWS Outposts.

Amazon RDS on AWS Outposts Read Replicas

Amazon RDS Read Replicas provide enhanced performance and durability for Amazon RDS database (DB) instances. They make it easy to elastically scale out beyond the capacity constraints of a single DB instance for read-heavy database workloads. Amazon RDS on AWS Outposts uses the MySQL and PostgreSQL DB engines' built-in replication functionality to create a read replica from a source DB instance. The source DB instance becomes the primary DB instance. Updates made to the primary DB instance are asynchronously copied to the read replica. Read replica use Customer-owned IP (CoIP) model and the replications runs on your local network.

Considerations for Amazon RDS on Outposts Read Replicas

Review the following considerations for Amazon RDS on Outposts deployments for Read Replicas:

- You can't create read replicas for RDS for SQL Server on RDS on Outposts DB instances.
- Cross-Region read replicas aren't supported on RDS on Outposts.
- Cascading read replicas aren't supported on RDS on Outposts.
- The source RDS on Outposts DB instance can't have local backups. The backup target for the source DB instance must be your AWS Region. Ensure you have a redundant service link connection of at least 500 mbps to send your RDS backups to AWS Region databases with frequently changing data or heavy write traffic.
- Read replicas require customer-owned IP (CoIP) pools.
- Read replicas on RDS on Outposts can only be created in the same virtual private cloud (VPC) as the source DB instance.
- Read replicas on RDS on Outposts can be located on the same Outpost or another Outpost in the same VPC as the source DB instance.

- You can't create read replicas for DB instances encrypted with AWS KMS External Key Store (XKS).
- Creating your read replica as a Multi-AZ DB instance is independent of whether the source database is a Multi-AZ DB instance.

Amazon RDS storage autoscaling on AWS Outposts

If your workload is unpredictable, you can enable storage autoscaling for an Amazon RDS DB instance. Amazon Relational Database Service (Amazon RDS) on AWS Outposts supports manual and automatic storage scaling. With storage autoscaling enabled, when Amazon RDS detects that your DB instance is running out of free database space it automatically scales up your storage which is based on the EBS capacity sized for your Outposts deployment. The feature provide the same capabilities that have at Regions where there are some specific factors that apply for autoscaling what can be found in the *Amazon RDS Autoscaling guide*. It’s important to carefully manage the maximum storage allocated for RDS instances on Outposts, as EBS resources are restricted to the capacity provisioned in the Outpost. Amazon RDS storage autoscaling allows you to set a maximum storage limit, ensuring that your deployment stays within the available EBS capacity. For more information on managing your Outposts capacity, see the Capacity management section of this whitepaper.

Amazon RDS on AWS Outposts local backup

Amazon RDS local backups on AWS Outposts enable you to recover an RDS DB instance directly from S3 stored locally on your Outposts. This allows you to meet data residency requirements and reduces latency compared to recovering from an AWS Region. With Amazon RDS on AWS Outposts, you have the following restore options:

- From a manual DB snapshot stored in the parent Region or locally on your Outposts.
- From an automated backup (point-in-time recovery):
  - If restoring from the parent AWS Region, you can store backups either in the AWS Region or on your Outposts.
  - If restoring from your Outposts, backups must be stored locally on Outposts with S3 support.

Considerations for Amazon RDS local backup on AWS Outposts

Refer to the following considerations to take advantage of Amazon RDS local backups on AWS Outposts:

- You need S3 on Outposts capacity to store the backups locally.
- Local backups are supported on MySQL and PostgreSQL DB instances.
- Local backups are not supported for Multi-AZ instance deployments or read replicas.

Snapshot Exporting and Restoring for RDS on AWS Outposts

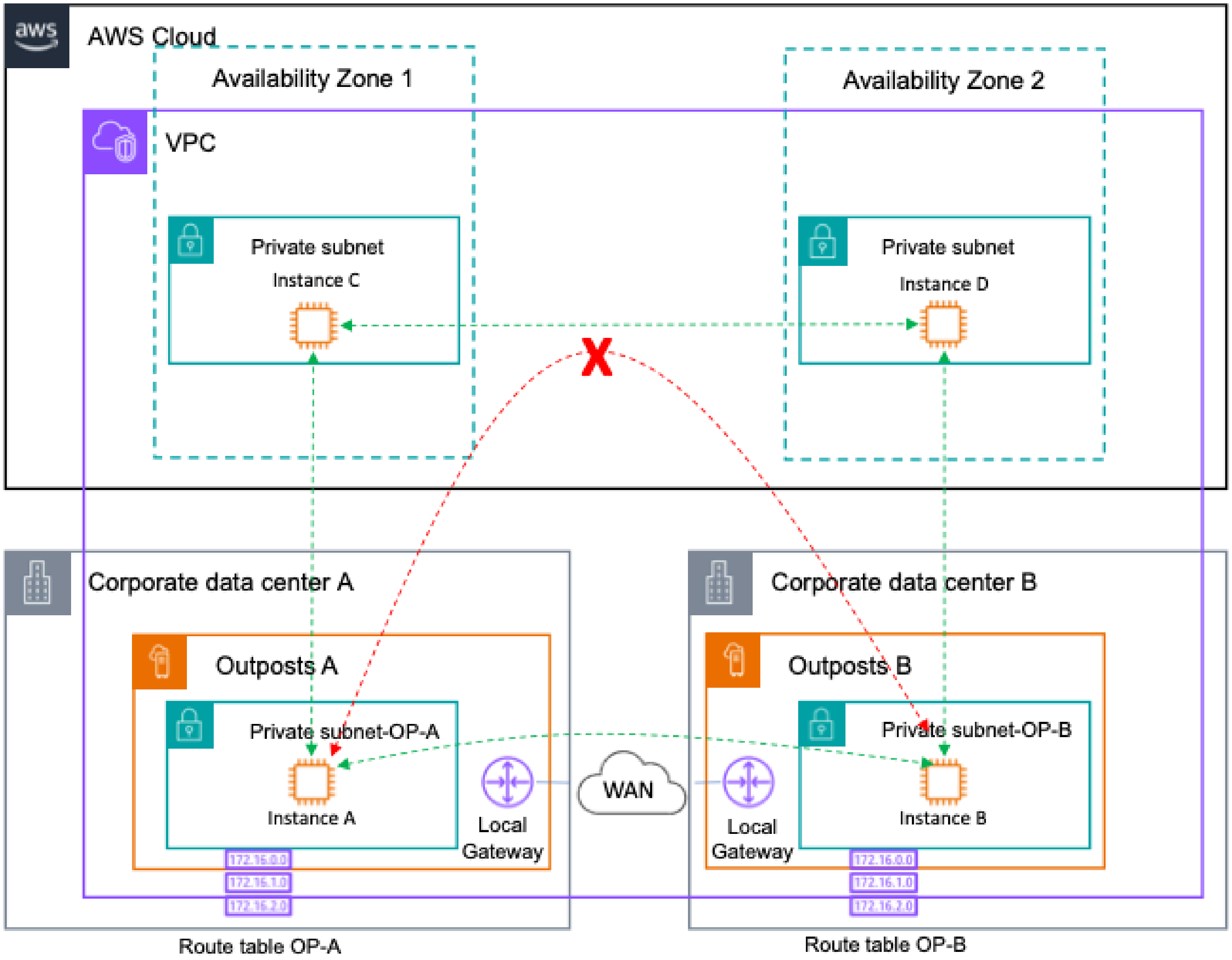
Exporting Snapshots to S3 and restoring a DB instance from Amazon S3: While RDS snapshots can be exported or restored directly from Amazon S3 in the AWS Region, this is not supported within AWS Outposts environments.

Larger failure modes

To design HA architectures to mitigate larger failure modes like rack, data center, Availability Zone (AZ), or Region failures, you should deploy multiple Outposts with sufficient infrastructure capacity in separate data centers with independent power and WAN connectivity. You anchor the Outposts to different Availability Zones (AZs) within an AWS Region or across multiple Regions. You should also provision resilient and sufficient site-to-site connectivity between the locations to support synchronous or asynchronous data replication and workload traffic redirection. Depending on your application architecture, you can use globally available Amazon Route 53 DNS and Amazon Route 53 on Outposts to direct traffic to the desired location, and automate traffic redirection to surviving locations in the event of large-scale failures.

Outposts Rack Intra-VPC routing

AWS Outposts rack supports intra-VPC communication across multiple Outposts. Resources on two separate logical Outposts can communicate with each other by routing traffic between subnets within the same VPC spanning across them using the Outpost local gateways (LGW). With intra-VPC communication across multiple Outposts, you can override the Local Route in your Outposts subnet associated route table by adding a more specific route to the other Outposts subnet using the local LGW as the next-hop. It can provide advantages to architecting applications that requires span a VPC between two logical Outposts as Amazon ECS across two Outposts racks or Amazon EKS cluster across AWS Outposts.

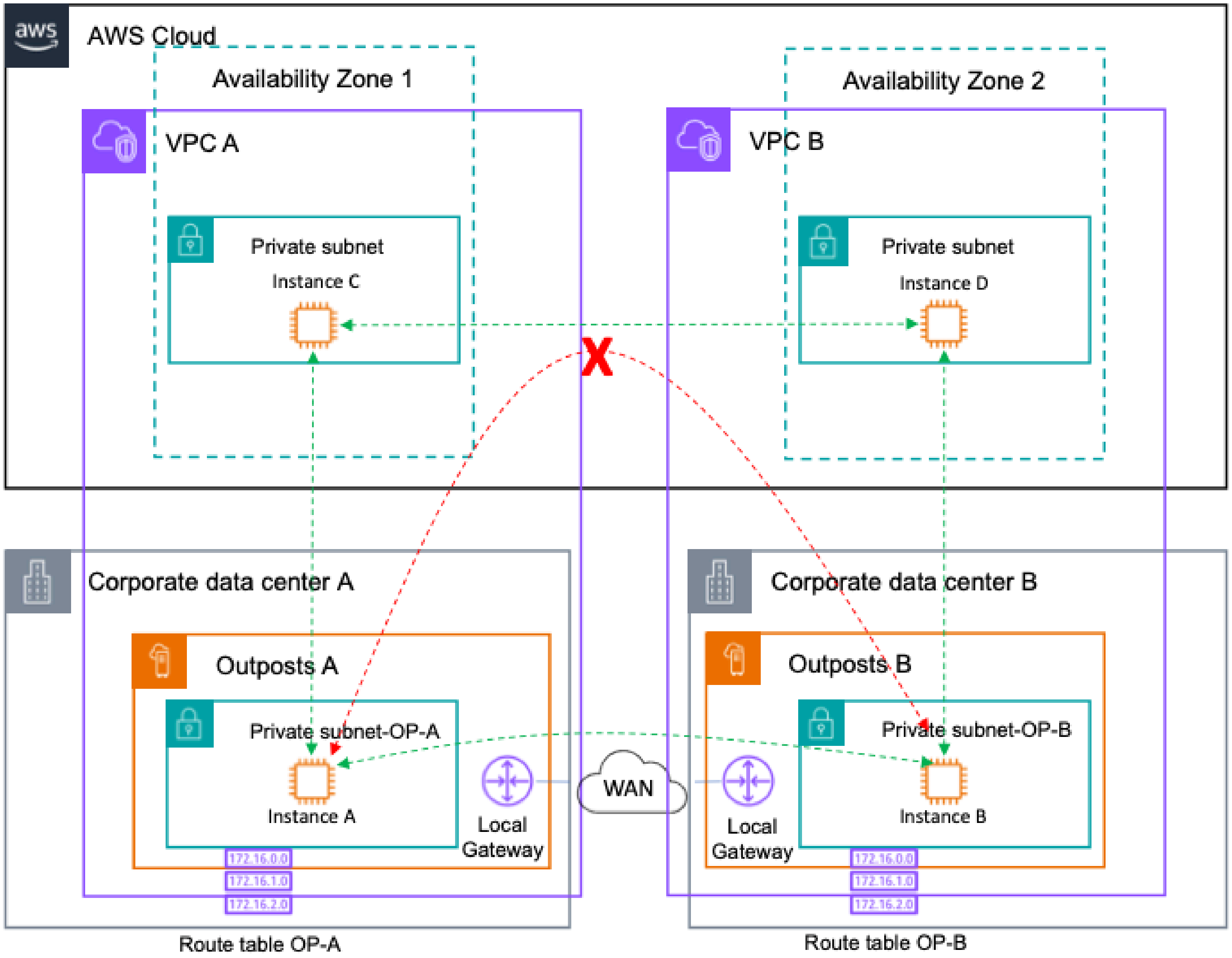


Network paths for single VPC with multiple logical Outposts

Outposts-to-Outposts traffic routing through the Region is blocked as this is an anti-pattern. Such traffic would incur egress charges in both directions and significantly higher latency than routing the traffic across the customer WAN.

Outposts Rack Inter-VPC routing

Resources on two separate Outposts deployed in different VPCs can communicate each other across the customer network. Deploying this architecture enable you to route traffic Outposts-to-Outposts by your local on-premises and WAN networks adding routes towards the counterpart Outposts/VPC subnets.



Network paths for multiple VPC with multiple logical Outposts

Recommended practices for protecting against larger failure modes:

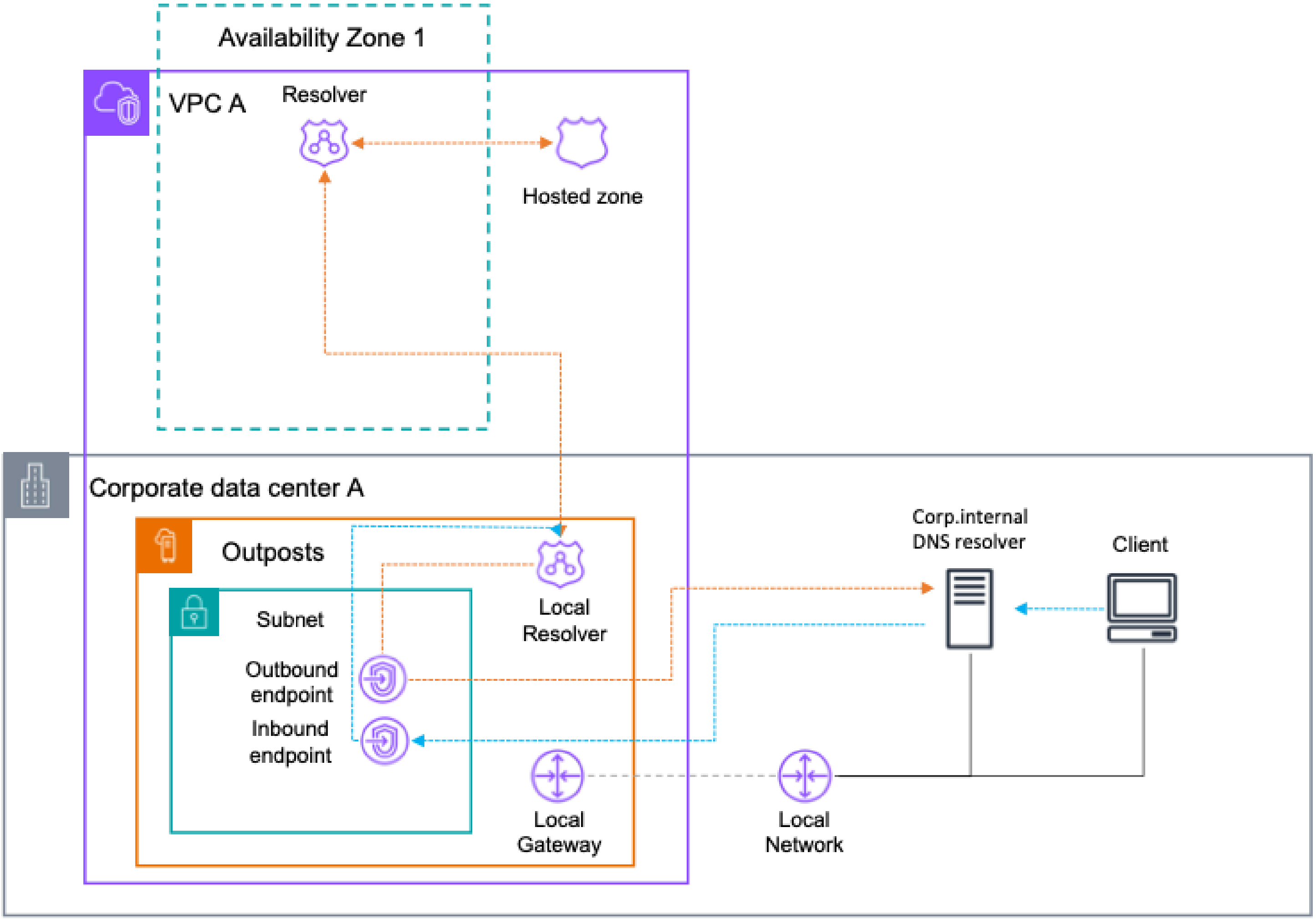
- Deploy multiple Outposts anchored to multiple AZs and Regions.
- Use separate VPCs for each Outpost in a multi-Outpost deployment.

Route 53 Local Resolver on Outposts

When the AWS Outposts service link gets impacted by a temporary disconnect, the local DNS resolution fails, making it difficult for applications and services to discover other services, even when they are running on the same Outposts rack. However, with Route 53 Resolver on AWS

Outposts, applications and services will continue to benefit from local DNS resolution to discover other services – even in the case of connectivity loss to the parent AWS Region. At the same time, for DNS resolution for on-premises host names, the Route 53 Resolver on Outposts helps to reduce latency as query results are cached and served locally, while being fully integrated with Route 53 Resolver endpoints.

Route 53 resolver Inbound endpoints forward DNS queries they receive from outside the VPC to the Resolver running in Outposts. In contrast, Route 53 Resolver Outbound enable Route 53 Resolvers to forward DNS queries to DNS resolvers that you manage on your on-premises network as is illustrated in the following diagram.



Route 53 resolver on Outposts

Route 53 Resolver on Outposts considerations

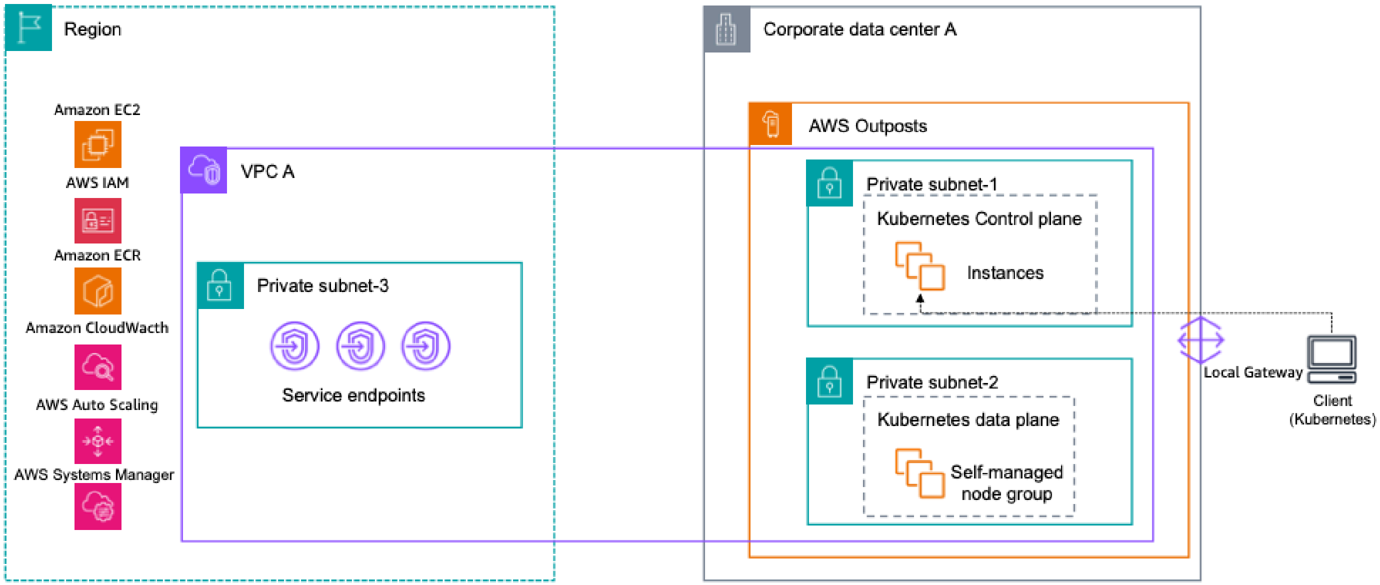
Consider the following:

- You must enable the Route 53 Resolver on Outposts, and it applies to the whole Outposts deployment, even if that involves multiple compute racks under a single Outposts ID.
- In order to enable this feature, your Outposts must have enough compute capacity to deploy the local resolver in the form of at least 4 EC2 instances of any c5.xlarge, m5.large or m5.xlarge.
- If you are using private DNS, you must share the Private Hosted Zone with the required Outposts VPCs' in order to cache the records locally in the Route 53 Resolver on Outposts.
- In order to enable integration with on-premises DNS with Inbound and Outbound endpoints, your Outposts must have enough compute capacity to deploy two EC2 instances per Route53 endpoint.

EKS Local Cluster on Outposts

When there are disconnections of Outposts service link from the parent region, there might be challenges with services as EKS Extended Cluster, where the control plane lives in the region. Among the challenges is the loss of communication between the EKS control plane and the worker nodes and PODs. Although both worker nodes and PODs can continue to operate and service applications that resides on Outposts locally, the Kubernetes control plane may consider them unhealthy and schedule their replacement when the connection to the control plane recovers. This may lead to application downtimes when connectivity is restored.

To simplify this, there is an option to host your entire EKS cluster on Outposts. In this configuration, both the Kubernetes control plane and your worker nodes run locally on premises on your Outposts compute capacity. That way, your cluster continues to operate even in the event of a temporary drop in your service link connection and after it is restored.



Amazon EKS local cluster on Outposts

Amazon EKS Local Cluster on Outposts considerations

Consider the following when you deploy an Amazon EKS local cluster in Outposts:

- During a disconnection there are not options to execute any change in the cluster itself that requires to add new worker nodes, or auto-scale a node group, as long as it depends on EC2 and ASG API calls toward the AWS parent Region.
- There are a set of unsupported features on local clusters listed on eksctl AWS Outposts support..

Conclusion

With AWS Outposts rack, you can build, manage, and scale highly available on-premises applications using familiar AWS tools and services like Amazon EC2, Amazon EBS, Amazon S3 on Outposts, Amazon ECS, Amazon EKS, and Amazon RDS. Workloads can run locally, serve clients, access applications and systems in your on-premises networks, and access the full set of services in the AWS Region. Outposts rack is ideal for workloads that require low latency access to on-premises systems, local data processing, data residency, and migration of applications with local system interdependencies.

When you provide an Outpost deployment with adequate power, space, and cooling and resilient connections to the AWS Region, you can build highly available single data center services. And, for higher levels of availability and resiliency, you can deploy multiple Outposts and distribute your applications across logical and geographic boundaries.

Outposts rack removes the undifferentiated heavy lifting of building on-premises compute, storage, and application networking pools and allow you to extend the reach of the AWS Global Infrastructure to your data centers and co-location facilities. Now, you can focus your time and energy towards modernizing your applications, streamlining your application deployments, and increasing the business impact of your IT services.

Contributors

Contributors to this document include:

- Jesus Federico, Principal Solutions Architect, Telco, Amazon Web Services
- Mallory Gershenfeld, S3 on Outposts, Amazon Web Services
- Rob Goodwin, Senior Solutions Architect, Hybrid Cloud, Amazon Web Services
- Chris Lunsford, Senior Specialist Solutions Architect, AWS Outposts, Amazon Web Services
- Rohan Mathews, Lead Architect, AWS Outposts, Amazon Web Services
- Brianna Rosentrater, Hybrid Edge Specialist Solutions Architect, Amazon Web Services
- Leonardo Solano, Principal Hybrid Edge Specialist Solutions Architect, Amazon Web Services
-

Document history

To be notified about updates to this whitepaper, subscribe to the RSS feed.

Change	Description	Date
Major update	Added updates about networking, DRS support, Amazon EKS Local Cluster, Placement Groups, and Amazon RDS on AWS Outposts	November 24, 2024
Minor update	Added additional slotting guidance in capacity planning.	February 9, 2024
Minor update	Updated to reflect feature launches since initial publicati on.	July 19, 2023
Minor update	Updated recommended practices for highly available network attachment.	June 29, 2023
Initial publication	Whitepaper first published.	August 12, 2021

Note

To subscribe to RSS updates, you must have an RSS plug-in enabled for the browser that you are using.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents current AWS product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

AWS Glossary

For the latest AWS terminology, see the AWS glossary in the *AWS Glossary Reference*.