Best practices for building a hybrid cloud architecture with AWS services

AWS Prescriptive Guidance

AWS Prescriptive Guidance: Best practices for building a hybrid cloud architecture with AWS services

Table of Contents

Best practices for building a hybrid cloud architecture with AWS services

Amazon Web Services (contributors)

June 2025 (document history)

Many businesses and organizations have adopted cloud computing as a key aspect of their technology strategy. They typically migrate their workloads to the AWS Cloud to increase agility, cost savings, performance, availability, resilience, and scalability. Most applications can be easily migrated, but some applications must remain on premises to take advantage of the low latency and local data processing of the on-premises environment, to avoid high data transfer costs, or for regulatory compliance. Furthermore, a subset of applications might need to be re-architected or modernized before they can be moved to the cloud. This leads many organizations to seek hybrid cloud architectures to integrate their on-premises and cloud operations to support a broad spectrum of use cases. This hybrid approach can provide the benefits of both on-premises and cloud-based computing, and can be particularly useful for edge computing scenarios.

When you build a hybrid cloud with AWS, we recommend that you determine your hybrid cloud strategy and your technical strategy:

- A **hybrid cloud strategy** provides guidelines that govern the consumption of cloud and on-premises resources to support your business objectives. This guidance describes common use cases for building a hybrid cloud, such as supporting ongoing migration to the cloud, ensuring business continuity during disasters, extending cloud infrastructure to the on-premises environment to support low-latency applications, or expanding your international presence on AWS. Defining this strategy helps you identify and define your business objectives for building a hybrid cloud, and provides guidelines for workload placements on the hybrid cloud.
- A **technical strategy** for the hybrid cloud identifies the guiding tenets of the hybrid cloud architecture and defines an implementation framework. This guidance outlines common requirements for a consistently deployed and managed hybrid cloud architecture to help you define principles for a planned hybrid cloud implementation. These requirements include standardized interfaces for resource provisioning and management across your cloud infrastructure.

This guide describes an operations and management framework to help solutions architects and operators identify the building blocks, best practices, and AWS hybrid cloud and in-Region services to implement a hybrid cloud with AWS.

Many organizations have used the solutions described in this guide to successfully deploy hybrid cloud environments that take advantage of the scale, agility, innovation, and global footprint provided by the AWS Cloud. (See case studies.) AWS hybrid cloud services deliver a consistent AWS experience from the cloud to on premises, and at the edge. Services such as AWS Outposts and AWS Local Zones place compute, storage, database, and other select AWS services close to large population and industry centers when you need low latency between end-user devices or existing on-premises data centers and workload servers.

In this guide:

Overview

This guide classifies AWS recommendations for the hybrid cloud into five pillars: networking, security, resiliency, capacity planning, and infrastructure management. It provides guidelines to help you improve your readiness and to develop a migration strategy by using an AWS hybrid edge service such as AWS Outposts or AWS Local Zones. We strongly recommend that you work with your AWS account team or AWS Partner to ensure that an AWS hybrid cloud specialist is available to assist you as you follow this guide and develop your process.

> **Note**
>
> Although AWS Outposts and Local Zones address similar problems, we recommend that you review use cases as well as the services and features available to decide which offering best suits your needs. For more information, see the AWS blog post AWS Local Zones and AWS Outposts, choosing the right technology for your edge workload.

Hybrid cloud workshops

With the assistance of an AWS hybrid cloud subject matter expert (SME), you can run a hybrid cloud workshop to assess your company's maturity level in relation to the five pillars discussed in this guide.

The workshop focuses on internal areas within your organization, such as networking, security, compliance, DevOps, virtualization, and business units. It helps you design a hybrid cloud architecture that meets your organization's requirements and defines implementation details, following the steps in the Hybrid cloud adoption process section of this guide.

PoCs

If you have specific requirements, you can use proofs of concept (PoCs) to validate functionality in Local Zones and AWS Outposts against those requirements.

AWS uses PoCs to help you test the workloads you want to move to an Outpost or Local Zone, to determine whether the workloads will be functional under the test architectures. To access a Local Zone for testing, follow the instructions in the Local Zones documentation. To test your

workload on AWS Outposts, work with your AWS account team or AWS Partner to access an AWS Outposts test laboratory and receive guidance from AWS solutions architects. In all scenarios, the development of a PoC requires you to generate a test document that contains:

- AWS services to use, such as Amazon Elastic Compute Cloud (Amazon EC2), Amazon Elastic Block Store (Amazon EBS), Amazon Virtual Private Cloud (Amazon VPC), and Amazon Elastic Kubernetes Service (Amazon EKS)
- Size and number of instances to consume (for example, `m5.xlarge` or `c5.2xlarge`)
- Test architecture diagram
- Test success criteria
- Details and objectives of each test to run

Pillars

The next section covers prerequisites and limitations for using the architectures discussed in this guide. The sections after that cover the details of each pillar so that the recommendations document that you create during the hybrid cloud workshop can reflect the design details required for implementation.

- Networking at the edge
- Security at the edge
- Resiliency at the edge
- Capacity planning at the edge
- Edge infrastructure management

Prerequisites and limitations

Before you follow this guide, work with your AWS account team or AWS Partner to review the prerequisites and limitations for implementing edge architectures with AWS Outposts and Local Zones.

Prerequisites

AWS Outposts

- Your existing data center must meet the AWS Outposts requirements for facilities, networking, and power. AWS Outposts is designed to operate in a data center environment that has 5-15 kVA redundant power inputs, 145.8 times the kVA of cubic feet per minute (CFM) airflow, and an ambient temperature between 41° F (5° C) and 95° F (35° C), among other requirements.
- Confirm that the AWS Outposts service is available in your country by consulting the AWS Outposts rack FAQs. See the question: *In which countries and territories is Outposts rack available?*
- If your organization requires four or more AWS Outposts racks, your data center must meet the Aggregation, Core, Edge (ACE) rack requirements.
- An internet or AWS Direct Connect link of at least 500 Mbps (1 Gbps is better) must be provided and sustained to connect AWS Outposts to the AWS Region, with appropriate backup connectivity if your use case requires it. The round-trip time latency from AWS Outposts to the Region must be 175 milliseconds at the maximum.
- You must have an active contract for AWS Enterprise Support or AWS Enterprise On-Ramp.

AWS Local Zones

- An AWS Local Zone must be available close to your data centers or users. See AWS Local Zones locations.
- Confirm that you have network connectivity from your on-premises infrastructure to the Local Zone:
  - Option 1: An AWS Direct Connect link from your data center to the AWS Direct Connect point of presence (PoP) that's closest to the Local Zone. For more information, see Direct Connect in the Local Zones documentation.

- Option 2: An internet link in addition to an on-premises virtual private network (VPN) appliance and the necessary licensing to launch a software-based VPN appliance on Amazon EC2 in the Local Zone. For more information, see VPN connection in the Local Zones documentation.

For additional connectivity options, see the Local Zones documentation.

Limitations

AWS Outposts

- Amazon Relational Database Service (Amazon RDS) on AWS Outposts Multi-AZ deployments require customer-owned IP (CoIP) address pools. For more information, see Customer-owned IP addresses for Amazon RDS on AWS Outposts.
- Multi-AZ on AWS Outposts is available for all supported versions of MySQL and PostgreSQL on Amazon RDS on AWS Outposts. For more information, see Amazon RDS on AWS Outposts support for Amazon RDS features. Amazon RDS on AWS Outposts supports SQL Server, Amazon RDS for MySQL, and Amazon RDS for PostgreSQL databases.
- AWS Outposts isn't designed to operate when it's disconnected from an AWS Region. For more information, see the Thinking in terms of failure modes section in the AWS whitepaper *AWS Outposts High Availability Design and Architecture Considerations*.
- Amazon Simple Storage Service (Amazon S3) on AWS Outposts has some limitations. These are discussed in the How is Amazon S3 on Outposts different from Amazon S3? section of the *Amazon S3 on Outposts User Guide*.
- Application Load Balancers on AWS Outposts don't support mutual TLS (mTLS) or sticky sessions.
- The ACE racks aren't fully enclosed and don't include front or rear doors.
- The instance capacity tool is applicable only for new orders.

AWS Local Zones

- Local Zones don't have an AWS Site-to-Site VPN endpoint. Instead, use a software-based VPN on Amazon EC2.
- Local Zones don't support AWS Transit Gateway. Instead, connect to the Local Zone by using a AWS Direct Connect Private virtual interface (VIF).

- Not all Local Zones support services such as Amazon RDS, Amazon FSx, Amazon EMR, or Amazon ElastiCache, or NAT gateways. For more information, see AWS Local Zones features.
- Application Load Balancers in Local Zones don't support mTLS or sticky sessions.

Hybrid cloud adoption process

The following sections discuss architectures and design details for each pillar of the AWS hybrid cloud:

- Networking at the edge
- Security at the edge
- Resiliency at the edge
- Capacity planning at the edge
- Edge infrastructure management

Networking at the edge

When you design solutions that use AWS edge infrastructure, such as AWS Outposts or Local Zones, you must carefully consider the network design. The network forms the foundation of connectivity for reaching workloads that are deployed in these edge locations, and is critical for ensuring low latency. This section outlines various aspects of hybrid edge connectivity.

VPC architecture

A virtual private cloud (VPC) spans all Availability Zones in its AWS Region. You can seamlessly extend any VPC in the Region to Outposts or Local Zones by using the AWS console or the AWS Command Line Interface (AWS CLI) to add an Outpost or Local Zone subnet. The following examples show how to create subnets in AWS Outposts and Local Zones by using the AWS CLI:

- **AWS Outposts**: To add an Outpost subnet to a VPC, specify the Amazon Resource Name (ARN) of the Outpost.

```
aws ec2 create-subnet --vpc-id vpc-081ec835f3EXAMPLE \
--cidr-block 10.0.0.0/24 \
--outpost-arn arn:aws:outposts:us-west-2:11111111111:outpost/op-0e32example1 \
--tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-only-subnet}]
```

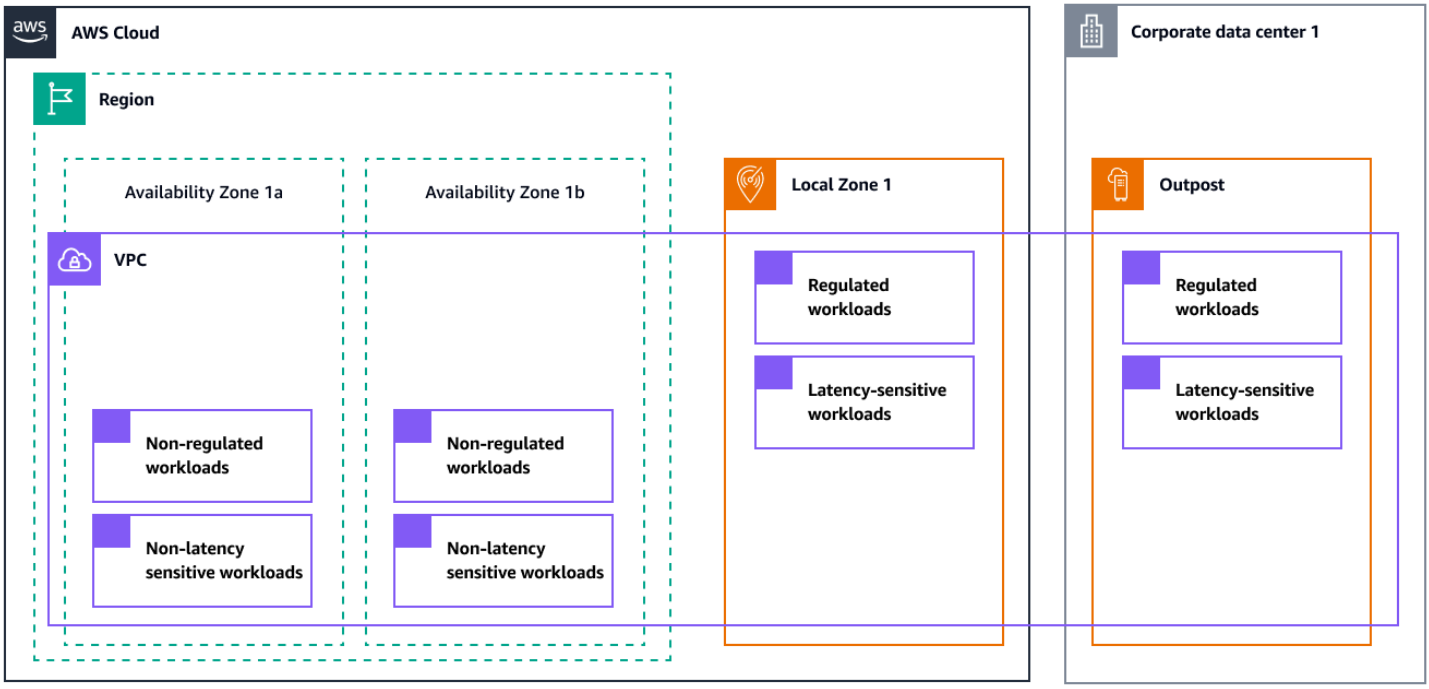
  For more information, see the AWS Outposts documentation.

- **Local Zones**: To add a Local Zone subnet to a VPC, follow the same procedure that you use with Availability Zones, but specify the Local Zone ID (`<local-zone-name>` in the following example).

```
aws ec2 create-subnet --vpc-id vpc-081ec835f3EXAMPLE \
--cidr-block 10.0.1.0/24 \
--availability-zone <local-zone-name> \
--tag-specifications ResourceType=subnet,Tags=[{Key=Name,Value=my-ipv4-only-subnet}]
```

  For more information, see the Local Zones documentation.

The following diagram shows an AWS architecture that includes Outpost and Local Zone subnets.

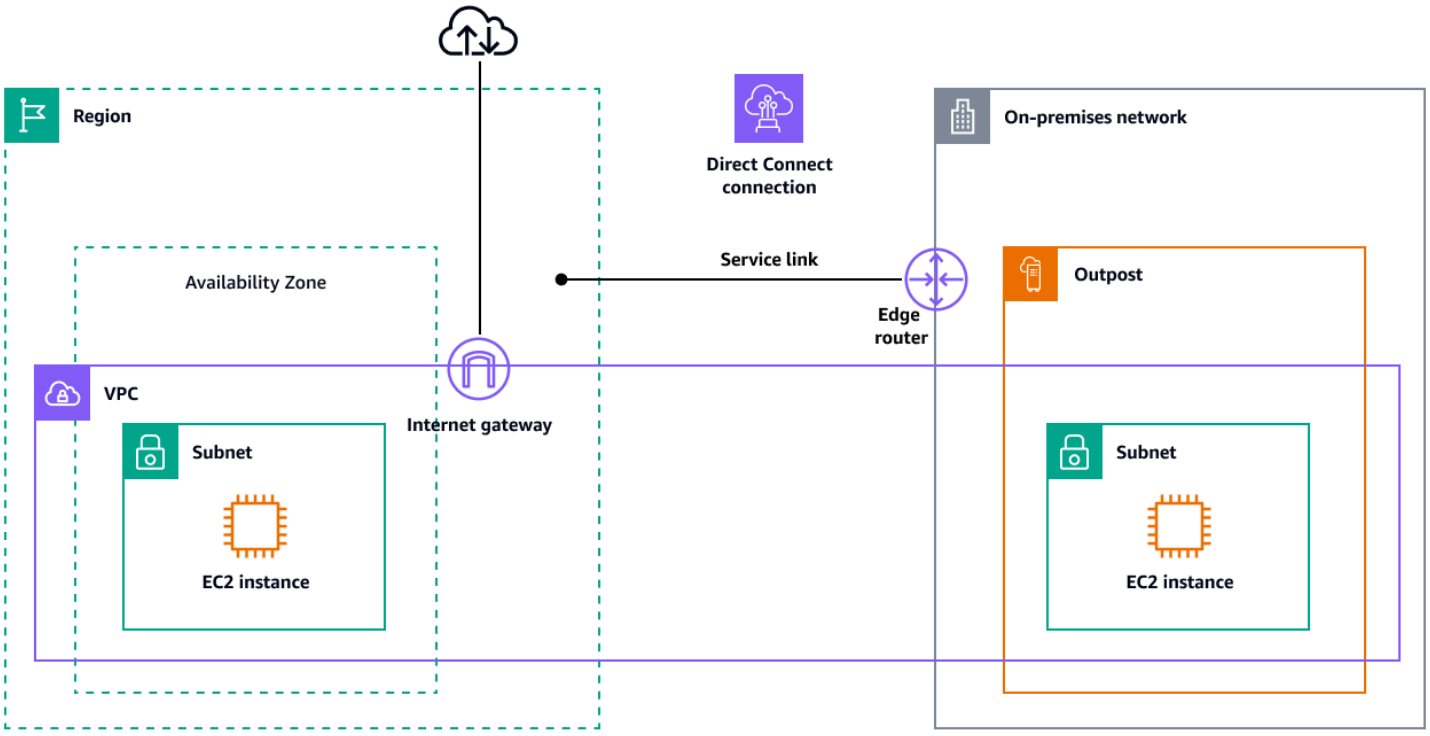


Edge to Region traffic

When you design a hybrid architecture by using services such as Local Zones and AWS Outposts, consider both control flows and data traffic flows between the edge infrastructures and AWS Regions. Depending on the type of edge infrastructure, your responsibility might vary: Some infrastructures require you to manage the connection to the parent Region, whereas others handle this through the AWS global infrastructure. This section explores the control plane and data plane connectivity implications for Local Zones and AWS Outposts.

AWS Outposts control plane

AWS Outposts provides a networking construct called a *service link*. The service link is a required connection between AWS Outposts and the selected AWS Region or parent Region (also referred to as *home Region*). It enables the management of the Outpost and the exchange of traffic between the Outpost and AWS Region. The service link uses an encrypted set of VPN connections to communicate with the home Region. You must provide connectivity between AWS Outposts and the AWS Region either through an internet link or an AWS Direct Connect public virtual interface (public VIF), or through an AWS Direct Connect private virtual interface (private VIF). For an optimal experience and resiliency, AWS recommends that you use redundant connectivity of at least 500 Mbps (1 Gbps is better) for the service link connection to the AWS Region. The minimum 500 Mbps service link connection allows you to launch Amazon EC2 instances, attach Amazon EBS volumes, and access AWS services such as Amazon EKS, Amazon EMR, and Amazon CloudWatch metrics. The network must support a maximum transmission unit (MTU) of 1,500 bytes between the Outpost and the service link endpoints in the parent AWS Region. For more information, see AWS Outposts connectivity to AWS Regions in the Outposts documentation.



For information about creating resilient architectures for service links that use AWS Direct Connect and the public internet, see the Anchor connectivity section in the AWS whitepaper *AWS Outposts High Availability Design and Architecture Considerations*.

AWS Outposts data plane

The data plane between AWS Outposts and the AWS Region is supported by the same service link architecture that is used by the control plane. The bandwidth of the data plane service link between AWS Outposts and the AWS Region should correlate with the amount of data that must be exchanged: The greater the data dependence, the greater the link bandwidth should be.

The bandwidth requirements vary depending on the following characteristics:

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

The traffic between EC2 instances in AWS Outposts and EC2 instances in the AWS Region has an MTU of 1,300 bytes. We recommend that you discuss these requirements with an AWS hybrid cloud specialist before you propose an architecture that has co-dependencies between the Region and AWS Outposts.

Local Zones data plane

The data plane between Local Zones and the AWS Region is supported through the AWS global infrastructure. The data plane is extended through a VPC from the AWS Region to a Local Zone. Local Zones also provide a high-bandwidth, secure connection to the AWS Region, and enables you to seamlessly connect to the full range of Regional services through the same APIs and tool sets.

The following table shows the connection options and associated MTUs.

From	To	MTU
Amazon EC2 in Region	Amazon EC2 in Local Zones	1,300 bytes
AWS Direct Connect	Local Zones	1,468 bytes
Internet gateway	Local Zones	1,500 bytes
Amazon EC2 in Local Zones	Amazon EC2 in Local Zones	9,001 bytes

Local Zones use the AWS global infrastructure to connect with AWS Regions. The infrastructure is fully managed by AWS, so you don't have to set up this connectivity. We recommend that you discuss your Local Zones requirements and considerations with an AWS hybrid cloud specialist before you design any architecture that has co-dependencies between the Region and Local Zones.

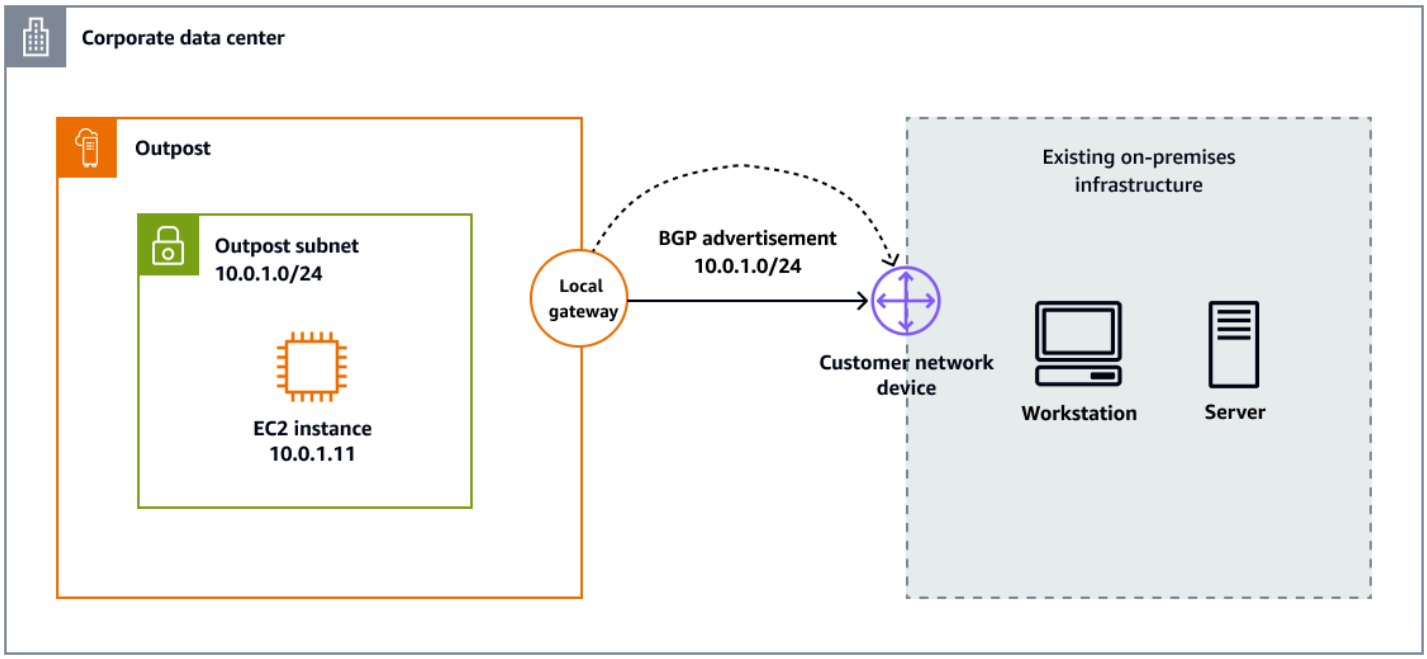
Edge to on-premises traffic

AWS hybrid cloud services are designed to address use cases that require low latency, local data processing, or data residency compliance. The network architecture for accessing this data is important, and it depends on whether your workload is running in AWS Outposts or Local Zones. Local connectivity also requires a well-defined scope, as discussed in the following sections.

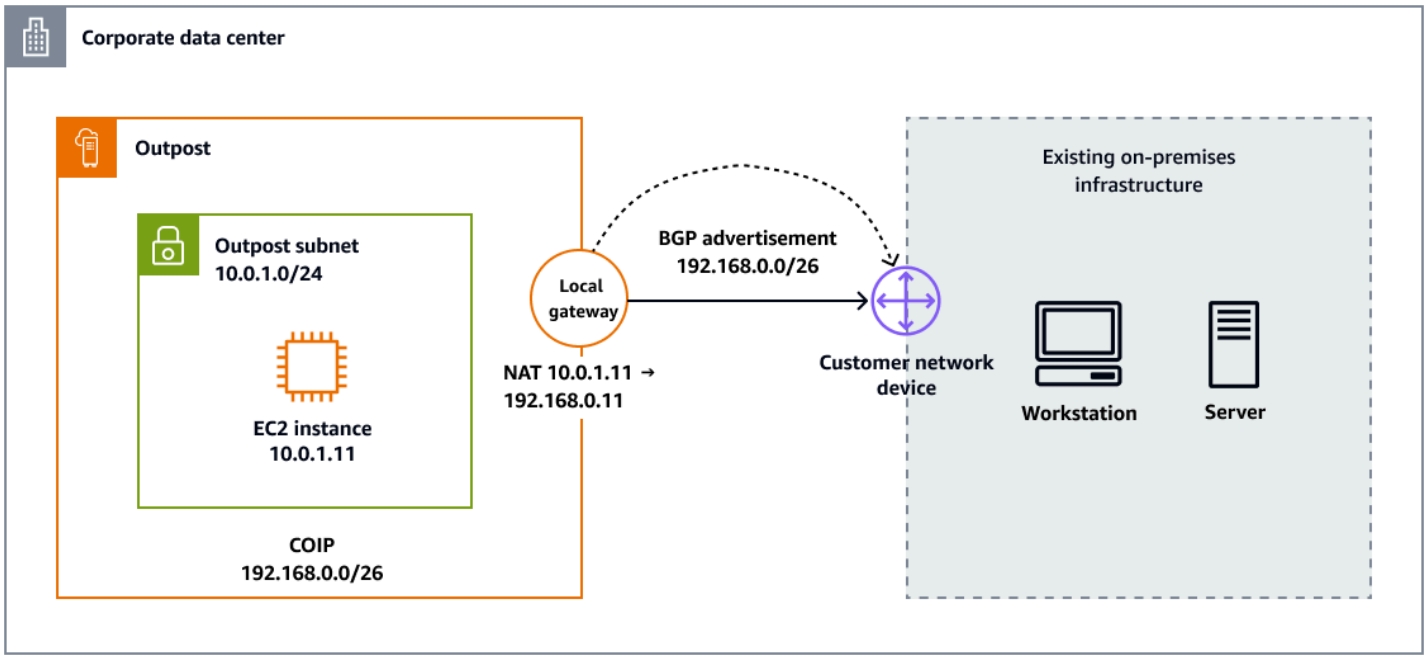
AWS Outposts local gateway

The local gateway (LGW) is a core component of the AWS Outposts architecture. The local gateway enables connectivity between your Outpost subnets and your on-premises network. The primary role of an LGW is to provide connectivity from an Outpost to your local on-premises network. It also provides connectivity to the internet through your on-premises network through either direct VPC routing or customer-owned IP addresses.

- **Direct VPC routing** uses the private IP address of the instances in your VPC to facilitate communication with your on-premises network. These addresses are advertised to your on-premises network with Border Gateway Protocol (BGP). Advertisement to BGP is only for the private IP addresses that belong to the subnets on your Outpost rack. This type of routing is the default mode for AWS Outposts. In this mode, the local gateway doesn't perform NAT for instances, and you do not need to assign Elastic IP addresses to your EC2 instances. The following diagram shows an AWS Outposts local gateway that uses direct VPC routing.



- With **customer-owned IP** addresses, you can provide an address range, known as a *customer-owned IP (CoIP) address pool*, which supports overlapping CIDR ranges and other network topologies. When you choose a CoIP, you must create an address pool, assign it to the local gateway route table, and advertise these addresses back to your network through BGP. CoIP addresses provide local or external connectivity to resources in your on-premises network. You can assign these IP addresses to resources on your Outpost, such as EC2 instances, by allocating a new Elastic IP address from the CoIP, and then assigning it to your resource. The following diagram shows an AWS Outposts local gateway that uses CoIP mode.



Local connectivity from AWS Outposts to a local network requires some parameter configurations, such as enabling the BGP routing protocol and advertising prefixes between the BGP peers. The MTU that can be supported between your Outpost and local gateway is 1,500 bytes. For more information, contact an AWS hybrid cloud specialist or review the AWS Outposts documentation.

Local Zones and the internet

Industries that require low-latency or local data residency (examples include gaming, live streaming, financial services, and the government) can use Local Zones to deploy and provide their applications to end users over the internet. During the deployment of a Local Zone, you must allocate public IP addresses for use in a Local Zone. When you allocate Elastic IP addresses, you can specify the location from which the IP address is advertised. This location is called a *network border group*. A network border group is a collection of Availability Zones, Local Zones, or AWS Wavelength Zones from which AWS advertises a public IP address. This helps ensure minimum latency or physical distance between the AWS network and the users who access the resources in these Zones. To see all the network border groups for Local Zones, see Available Local Zones in the Local Zones documentation.

To expose an Amazon EC2-hosted workload in a Local Zone to the internet, you can enable the **Auto-assign Public IP** option when you launch the EC2 instance. If you use an Application Load Balancer, you can define it as internet-facing so that public IP addresses assigned to the Local Zone can be propagated by the border network that's associated with the Local Zone. In addition, when

you use Elastic IP addresses, you can associate one of these resources with an EC2 instance after its launch. When you send traffic through an internet gateway in Local Zones, the same instance bandwidth specifications used by the Region are applied. Local Zone network traffic goes directly to the internet or to points of presence (PoPs) without traversing the Local Zone's parent Region, to enable access to low-latency computing.

Local Zones provide the following connectivity options over the internet:

- Public access: Connects workloads or virtual appliances to the internet by using Elastic IP addresses through an internet gateway.
- Outbound internet access: Enables resources to reach public endpoints through network address translation (NAT) instances or virtual appliances with associated Elastic IP addresses, without direct internet exposure.
- VPN connectivity: Establishes private connections by using Internet Protocol Security (IPsec) VPN through virtual appliances with associated Elastic IP addresses.

For more information, see Connectivity options for Local Zones in the Local Zones documentation.

Local Zones and AWS Direct Connect

Local Zones also support AWS Direct Connect, which lets you route your traffic over a private network connection. For more information, see Direct Connect in Local Zones in the Local Zones documentation.

Local Zones and transit gateways

AWS Transit Gateway doesn't support direct VPC attachments to Local Zone subnets. However, you can connect to Local Zone workloads by creating Transit Gateway attachments in the parent Availability Zone subnets of the same VPC. This configuration enables interconnectivity between multiple VPCs and your Local Zone workloads. For more information, see Transit gateway connection between Local Zones in the Local Zones documentation.

Local Zones and VPC peering

You can extend any VPC from a parent Region into a Local Zone by creating a new subnet and assigning it to the Local Zone. VPC peering can be established between VPCs that are extended to Local Zones. When the peered VPCs are in the same Local Zone, traffic stays within the Local Zone and does not hairpin through the parent Region.

Security at the edge

In the AWS Cloud, security is the top priority. As organizations adopt the scalability and flexibility of the cloud, AWS helps them adopt security, identity, and compliance as key business factors. AWS integrates security into its core infrastructure and offers services to help you meet your unique cloud security requirements. When you expand the scope of your architecture into the AWS Cloud, you benefit from the integration of infrastructures such as Local Zones and Outposts into AWS Regions. This integration enables AWS to extend a select group of core security services to the edge.

Security is a shared responsibility between AWS and you. The AWS shared responsibility model differentiates between the 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 AWS security as part of AWS compliance programs.
- **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.

Data protection

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

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

Encryption at rest

Encryption in EBS volumes

With AWS Outposts, all data is encrypted at rest. The key material is wrapped with an external key, the Nitro Security Key (NSK), which is stored in a removable device. The NSK is required to decrypt the data on your Outpost rack. You can use Amazon EBS encryption for your EBS volumes and snapshots. Amazon EBS encryption uses AWS Key Management Service (AWS KMS) and KMS keys.

In the case of Local Zones, all EBS volumes are encrypted by default in all Local Zones, except to for the list documented in the AWS Local Zones FAQ (see the question: *What's the default encryption behavior of EBS volumes in Local Zones?*), unless encryption is enabled for the account.

Encryption in Amazon S3 on Outposts

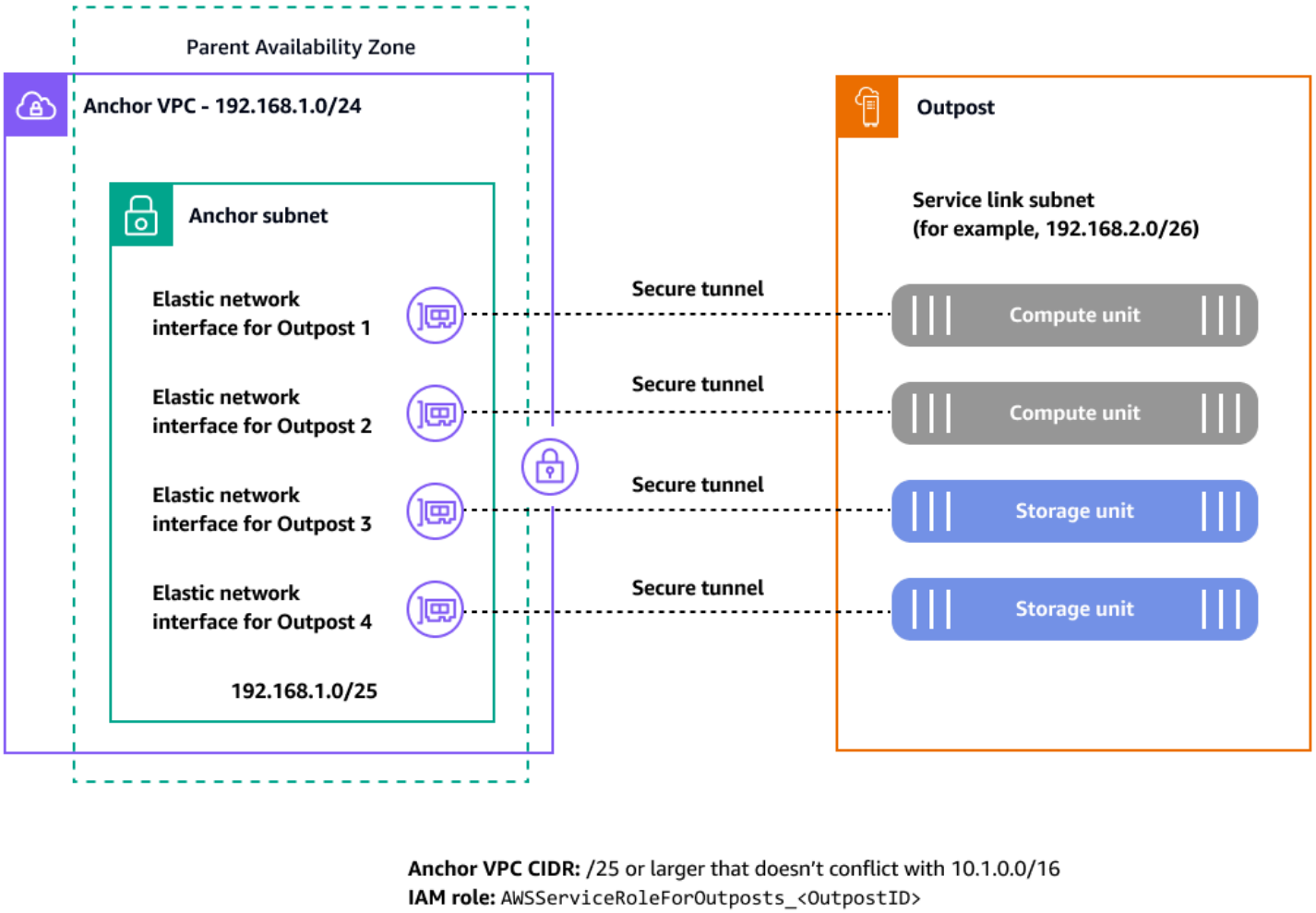
By default, all data stored in Amazon S3 on Outposts is encrypted by using server-side encryption with Amazon S3 managed encryption keys (SSE-S3). You can optionally use server-side encryption with customer-provided encryption keys (SSE-C). To use SSE-C, specify an encryption key as part of your object API requests. Server-side encryption encrypts only the object data, not the object metadata.

> **Note**
>
> Amazon S3 on Outposts doesn't support server-side encryption with KMS keys (SSE-KMS).

Encryption in transit

For AWS Outposts, the service link is a necessary connection between your Outposts server and your chosen AWS Region (or home Region) and allows for the management of the Outpost and the exchange of traffic to and from the AWS Region. The service link uses an AWS managed VPN to communicate with the home Region. Each host inside AWS Outposts creates a set of VPN tunnels to split control plane traffic and VPC traffic. Depending on the service link connectivity (internet or AWS Direct Connect) for AWS Outposts, those tunnels require firewall ports to be opened for the service link to create the overlay on top of it. For detailed technical information about the security of AWS Outposts and the service link, see Connectivity through service link and Infrastructure security in AWS Outposts in the AWS Outposts documentation.

The AWS Outposts service link creates encrypted tunnels that establish control plane and data plane connectivity to the parent AWS Region, as illustrated in the following diagram.



Each AWS Outposts host (compute and storage) requires these encrypted tunnels over well-known TCP and UDP ports to communicate with its parent Region. The following table shows the source and destination ports and addresses for the UDP and TCP protocols.

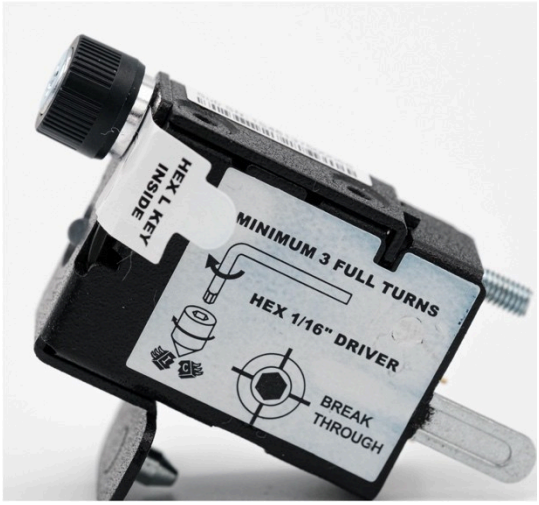
Protocol	Source port	Source address	Destination port	Destination address
UDP	443	AWS Outposts service link /26	443	AWS Outposts Region's public routes or anchor VPC CIDR
TCP	1025-65535	AWS Outposts service link /26	443	AWS Outposts Region's public

Protocol	Source port	Source address	Destination port	Destination address
				routes or anchor VPC CIDR

Local Zones are also connected to the parent Region through the redundant and very high-bandwidth global private backbone of Amazon. This connection gives applications that are running in Local Zones fast, secure, and seamless access to other AWS services. As long as Local Zones are part of the AWS global infrastructure, all data flowing over the AWS global network is automatically encrypted at the physical layer before it leaves AWS secured facilities. If you have specific requirements to encrypt the data in transit between your on-premises locations and AWS Direct Connect PoPs to access a Local Zone, you can enable MAC Security (MACsec) between your on-premises router or switch and the AWS Direct Connect endpoint. For more information, see the AWS blog post Adding MACsec security to AWS Direct Connect connections.

Data deletion

When you stop or terminate an EC2 instance in AWS Outposts, 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. Deleting data from the Outpost hardware involves the use of specialized hardware. The NSK is a small device, illustrated in the following photograph, that attaches to the front of every compute or storage unit in an Outpost. It is designed to provide a mechanism to prevent your data from being exposed from your data center or colocation site. Data on the Outpost device is protected by wrapping keying material used to encrypt the device and storing the wrapped material on the NSK. When you return an Outpost host, you destroy the NSK by turning a small screw on the chip that crushes the NSK and physically destroys the chip. Destroying the NSK shreds the data cryptographically on your Outpost.



Identity and access management

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

The following table lists the IAM features that you can use with AWS Outposts.

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

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

* In addition to IAM identity-based policies, Amazon S3 on Outposts supports both bucket and access point policies. These are resource-based policies that are attached to the Amazon S3 on Outposts resource.

For more information about how these features are supported in AWS Outposts, see the AWS Outposts user guide.

Infrastructure security

Infrastructure protection is a key part of an information security program. It ensures that workload systems and services are protected against unintended and unauthorized access, and potential vulnerabilities. For example, you define trust boundaries (for example, network and account boundaries), system security configuration and maintenance (for example, hardening, minimization, and patching), operating system authentication and authorizations (for example, users, keys, and access levels), and other appropriate policy-enforcement points (for example, web application firewalls or API gateways).

AWS provides a number of approaches to infrastructure protection, as discussed in the following sections.

Protecting networks

Your users might be part of your workforce or your customers, and can be located anywhere. For this reason, you can't trust everyone who has access to your network. When you follow the principle of applying security at all layers, you employ a zero trust approach. In the zero trust security model, application components or microservices are considered discrete, and no component or microservice trusts any other component or microservice. To achieve zero trust security, follow these recommendations:

- Create network layers. Layered networks help logically group similar networking components. They also shrink the potential scope of impact of unauthorized network access.

- Control traffic layers. Apply multiple controls with a defense-in-depth approach for both inbound and outbound traffic. This includes the use of security groups (stateful inspection firewalls), network ACLs, subnets, and route tables.
- Implement inspection and protection. Inspect and filter your traffic at each layer. You can inspect your VPC configurations for potential unintended access by using Network Access Analyzer. You can specify your network access requirements and identify potential network paths that do not meet them.

Protecting compute resources

Compute resources include EC2 instances, containers, AWS Lambda functions, database services, IoT devices, and more. Each compute resource type requires a different approach to security. However, these resources do share common strategies that you need to consider: *defense in depth*, *vulnerability management*, *reduction in attack surface*, *automation of configuration and operation*, and *performing actions at a distance*.

Here's general guidance for protecting your compute resources for key services:

- Create and maintain a vulnerability management program. Regularly scan and patch resources such as EC2 instances, Amazon Elastic Container Service (Amazon ECS) containers, and Amazon Elastic Kubernetes Service (Amazon EKS) workloads.
- Automate compute protection. Automate your protective compute mechanisms, including vulnerability management, reduction in attack surface, and management of resources. This automation frees up time that you can use to secure other aspects of your workload, and helps reduce the risk of human error.
- Reduce the attack surface. Reduce your exposure to unintended access by hardening your operating systems and minimizing the components, libraries, and externally consumable services that you use.

In addition, for each AWS service that you use, check the specific security recommendations in the service documentation.

Internet access

Both AWS Outposts and Local Zones provide architectural patterns that give your workloads access to and from the internet. When you use these patterns, consider internet consumption from the

Region a viable option only if you use it for patching, updating, accessing Git repositories that are external to AWS, and similar scenarios. For this architectural pattern, the concepts of centralized inbound inspection and centralized internet egress apply. These access patterns use AWS Transit Gateway, NAT gateways, network firewalls, and other components that reside in AWS Regions, but are connected to AWS Outposts or Local Zones through the data path between the Region and the edge.

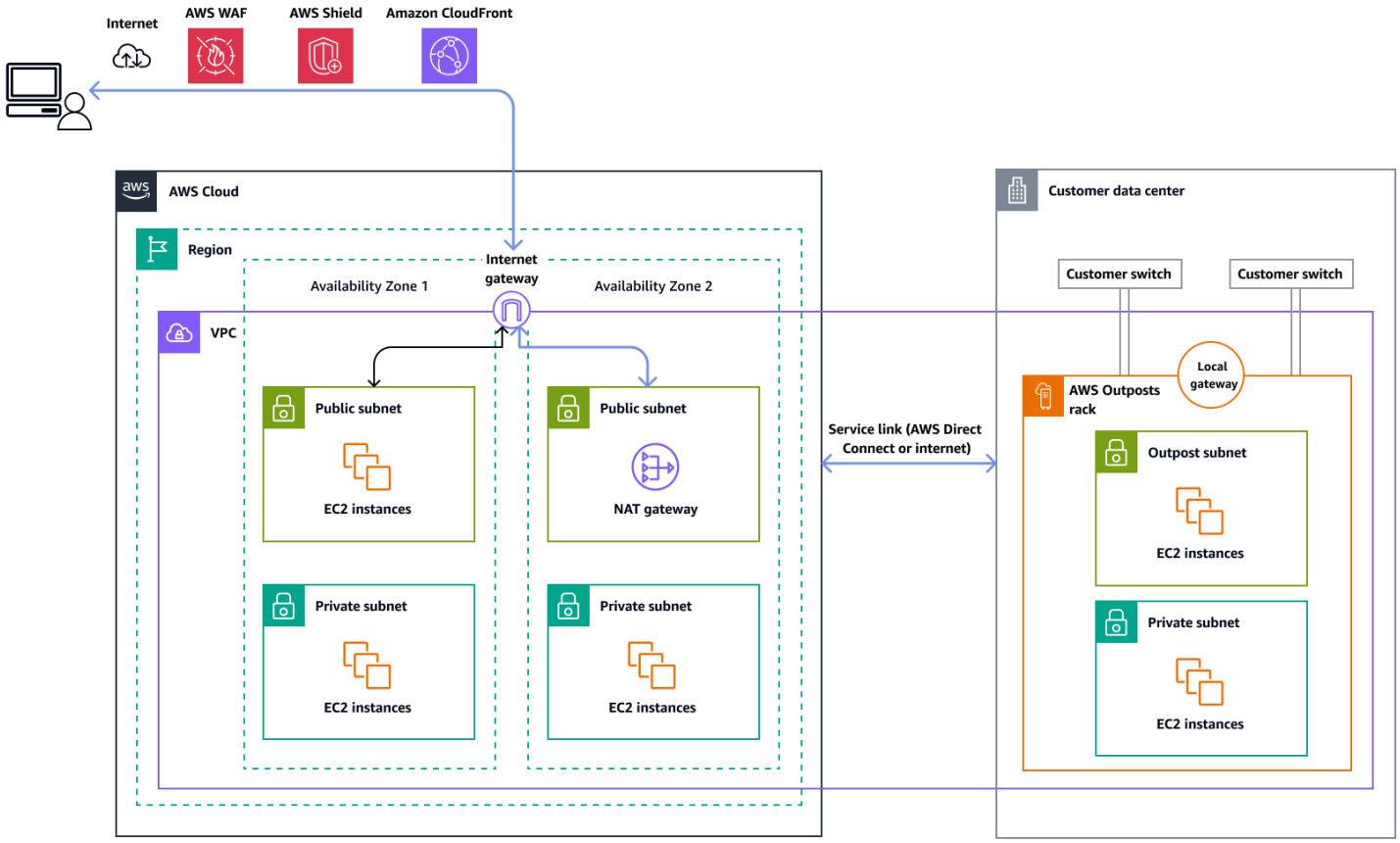
Local Zones adopts a network construct called a *network border group*, which is used in AWS Regions. AWS advertises public IP addresses from these unique groups. A network border group consists of Availability Zones, Local Zones, or Wavelength Zones. You can explicitly allocate a pool of public IP addresses for use in a network border group. You can use a network border group to extend the internet gateway to Local Zones by allowing Elastic IP addresses to be served from the group. This option requires that you deploy other components to complement the core services available in Local Zones. Those components might come from ISVs and help you build inspection layers in your Local Zone, as described in the AWS blog post Hybrid inspection architectures with AWS Local Zones.

In AWS Outposts, if you want to use the local gateway (LGW) to reach the internet from your network, you must modify the custom route table that's associated with the AWS Outposts subnet. The route table must have a default route entry (0.0.0.0/0) that uses the LGW as the next hop. You are responsible for implementing the remaining security controls in your local network, including perimeter defenses such as firewalls and intrusion prevention systems or intrusion detection systems (IPS/IDS). This aligns with the shared responsibility model, which divides security duties between you and the cloud provider.

Internet access through the parent AWS Region

In this option, the workloads in the Outpost access the internet through the service link and the internet gateway in the parent AWS Region. Outbound traffic to the internet can be routed through the NAT gateway that's instantiated in your VPC. For additional security for your ingress and egress traffic, you can use AWS security services such as AWS WAF, AWS Shield, and Amazon CloudFront in the AWS Region.

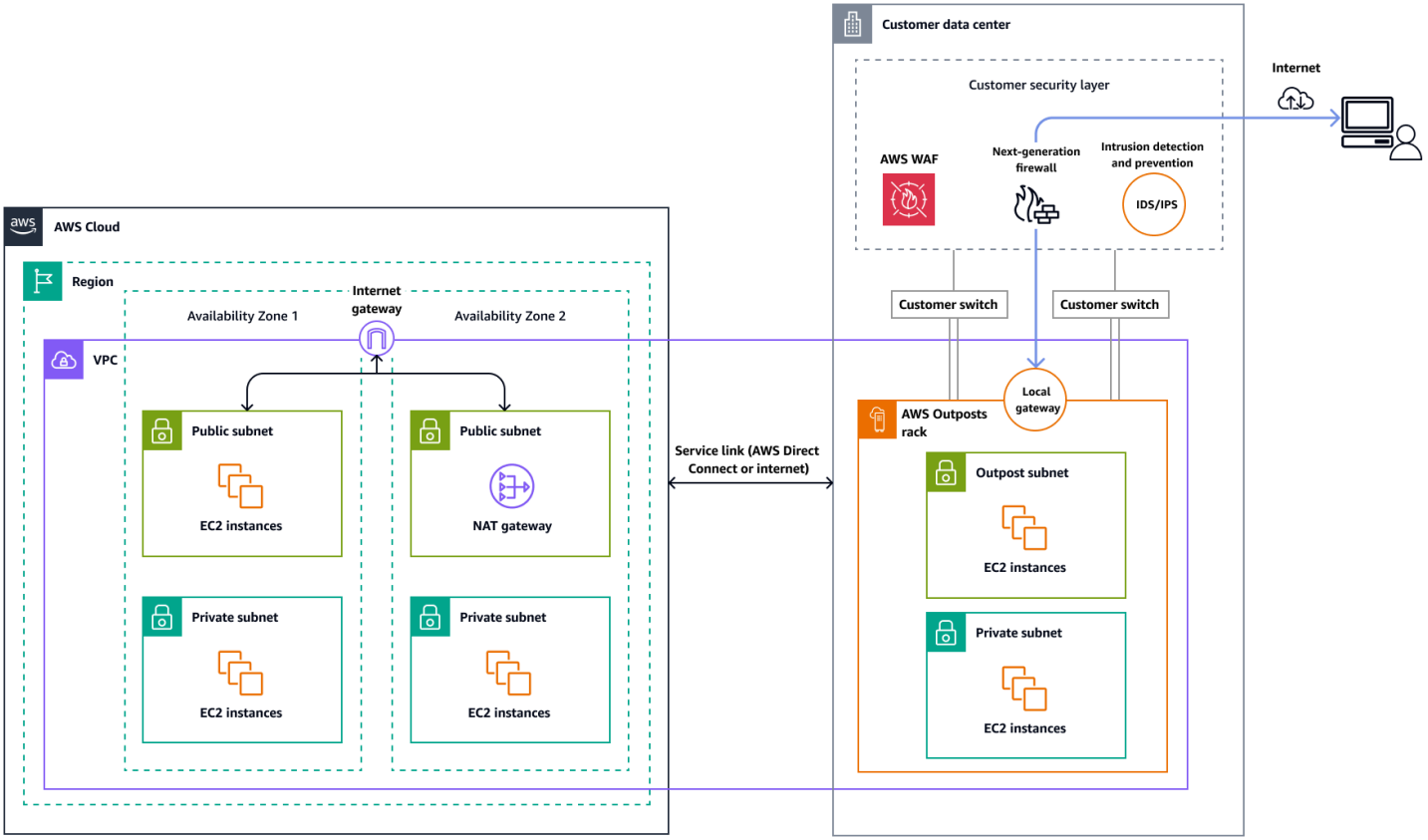
The following diagram shows traffic between the workload in the AWS Outposts instance and the internet going through the parent AWS Region.



Internet access through your local data center's network

In this option, the workloads in the Outpost access the internet through your local data center. The workload traffic that accesses the internet traverses through your local internet point of presence and egresses locally. In this case, your local data center's network security infrastructure is responsible for securing the AWS Outposts workload traffic.

The following image shows traffic between a workload in the AWS Outposts subnet and the internet going through a data center.



Infrastructure governance

Regardless of whether your workloads are deployed in an AWS Region, Local Zone, or Outpost, you can use AWS Control Tower for infrastructure governance. AWS Control Tower offers a straightforward way to set up and govern an AWS multi-account environment, following prescriptive best practices. AWS Control Tower orchestrates the capabilities of several other AWS services, including AWS Organizations, AWS Service Catalog, and IAM Identity Center(see all integrated services) to build a landing zone in less than an hour. Resources are set up and managed on your behalf.

AWS Control Tower provides unified governance across all AWS environments, including Regions, Local Zones (low-latency extensions), and Outposts (on-premises infrastructure). This helps ensure consistent security and compliance across your entire hybrid cloud architecture. For more information, see the AWS Control Tower documentation.

You can configure AWS Control Tower and capabilities such as guardrails to comply with data residency requirements in governments and regulated industries such as Financial Services

Institutions (FSIs). To understand how to deploy guardrails for data residency at the edge, see the following:

- Best practices for managing data residency in AWS Local Zones using landing zone controls (AWS blog post)
- Architecting for data residency with AWS Outposts rack and landing zone guardrails (AWS blog post)
- Data Residency with Hybrid Cloud Services Lens (AWS Well-Architected Framework documentation)

Sharing Outposts resources

Because an Outpost is a finite infrastructure that lives in your data center or in a co-location space, for centralized governance of AWS Outposts, you need to centrally control which accounts AWS Outposts resources are shared with.

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

Shareable resources in AWS Outposts are:

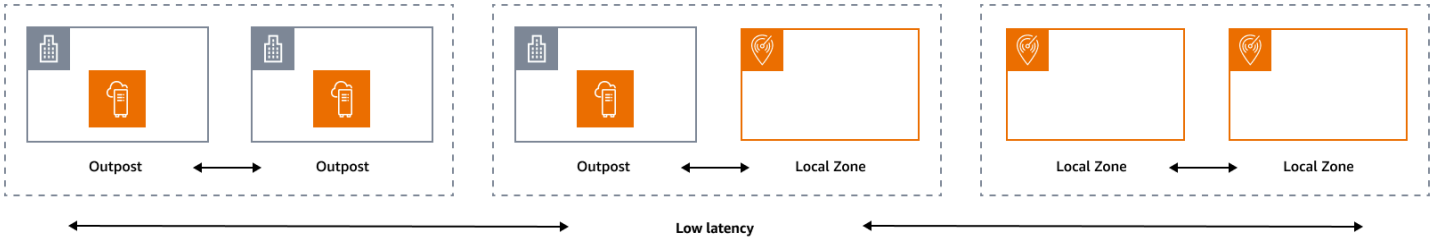
- Allocated dedicated hosts
- Capacity reservations
- Customer-owned IP (CoIP) address pools
- Local gateway route tables
- Outposts
- Amazon S3 on Outposts
- Sites
- Subnets

To follow the best practices for sharing Outposts resources in a multi-account environment, see the following AWS blog posts:

- Sharing AWS Outposts in a multi-account AWS environment: Part 1
- Sharing AWS Outposts in a multi-account AWS environment: Part 2

Resiliency at the edge

The reliability pillar encompasses the ability of a workload to perform its intended function correctly and consistently when it is expected to. This includes the ability to operate and test the workload through its lifecycle. In this sense, when you design a resilient architecture at the edge, you must first consider which infrastructures you will use to deploy that architecture. There are three possible combinations to implement by using AWS Local Zones and AWS Outposts: *Outpost to Outpost*, *Outpost to Local Zone*, and *Local Zone to Local Zone*, as illustrated in the following diagram. Although there are other possibilities for resilient architectures, such as combining AWS edge services with traditional on-premises infrastructure or AWS Regions, this guide focuses on these three combinations that apply to the design of hybrid cloud services



Infrastructure considerations

At AWS, one of the core principles of service design is to avoid single points of failure in the underlying physical infrastructure. Because of this principle, AWS software and systems use multiple Availability Zones and are resilient to the failure of a single zone. At the edge, AWS offers infrastructures that are based on Local Zones and Outposts. Therefore, a critical factor in ensuring resilience in infrastructure design is defining where an application's resources are deployed.

Local Zones

Local Zones act similarly to Availability Zones within their AWS Region, because they can be selected as a placement location for zonal AWS resources such as subnets and EC2 instances. However, they aren't located in an AWS Region, but near large population, industrial, and IT centers where no AWS Region exists today. Despite this, they still retain high-bandwidth, secure connections between local workloads in the Local Zone and workloads that are running in the AWS

Region. Therefore, you should use Local Zones to deploy workloads closer to your users for low-latency requirements.

Outposts

AWS Outposts is a fully managed service that extends AWS infrastructure, AWS services, APIs, and tools to your data center. The same hardware infrastructure that's used in the AWS Cloud is installed in your data center. Outposts are then connected to the nearest AWS Region. You can use Outposts to support your workloads that have low latency or local data processing requirements.

Parent Availability Zones

Each Local Zone or Outpost has a parent Region (also referred to as *home Region*). The parent Region is where the control plane of the AWS edge infrastructure (Outpost or Local Zone) is anchored. In the case of Local Zones, the parent Region is a fundamental architectural component of a Local Zone and cannot be modified by customers. AWS Outposts extends the AWS Cloud to your on-premises environment, so you must select a specific Region and Availability Zone during the ordering process. This selection anchors the control plane of your Outposts deployment to the chosen AWS infrastructure.

When you develop high availability architectures in the edge, the parent Region of these infrastructures, such as Outposts or Local Zones, must be the same, so that a VPC can be extended between them. This extended VPC is the basis for creating these high-availability architectures. When you define a highly resilient architecture, this is why you must validate the parent Region and the Availability Zone of the Region where the service will be (or is) anchored. As illustrated in the following diagram, if you want to deploy a high availability solution between two Outposts, you must choose two different Availability Zones to anchor the Outposts. This allows for a Multi-AZ architecture from a control plane perspective. If you want to deploy a highly available solution that includes one or more Local Zones, you must first validate the parent Availability Zone where the infrastructure is anchored. For this purpose, use the following AWS CLI command:

```
aws ec2 describe-availability-zones --zone-ids use1-mia1-az1
```

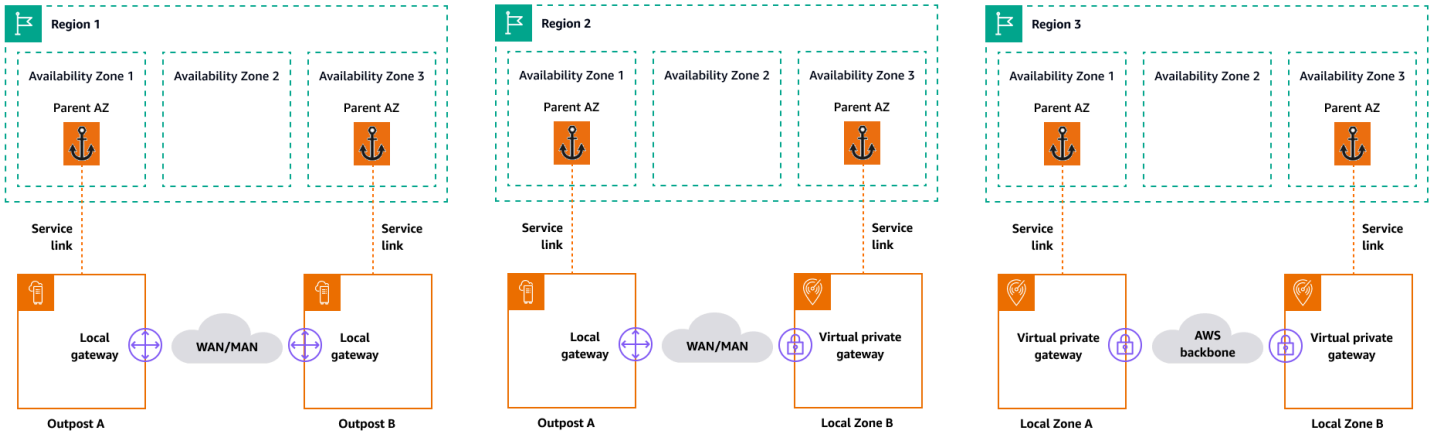
Output of the previous command:

```
{     "AvailabilityZones": [
          {
             "State": "available",
             "OptInStatus": "opted-in",
```

```
            "Messages": [],
            "RegionName": "us-east-1",
            "ZoneName": "us-east-1-mia-1a",
            "ZoneId": "use1-mia1-az1",
            "GroupName": "us-east-1-mia-1",
            "NetworkBorderGroup": "us-east-1-mia-1",
            "ZoneType": "local-zone",
            "ParentZoneName": "us-east-1d",
            "ParentZoneId": "use1-az2"
        }
    ]
}
```

In this example, the Miami Local Zone (`us-east-1d-mia-1a1`) is anchored in the `us-east-1d-az2` Availability Zone. Therefore, if you need to create a resilient architecture at the edge, you must ensure that the secondary infrastructure (either Outposts or Local Zones) is anchored to an Availability Zone other than `us-east-1d-az2`**.** For example, `us-east-1d-az1` would be valid.

The following diagram provides examples of highly available edge infrastructures.



Networking considerations

This section discusses initial considerations for networking at the edge, mainly for connections to access the edge infrastructure. It reviews valid architectures that provide a resilient network for the service link.

Resiliency networking for Local Zones

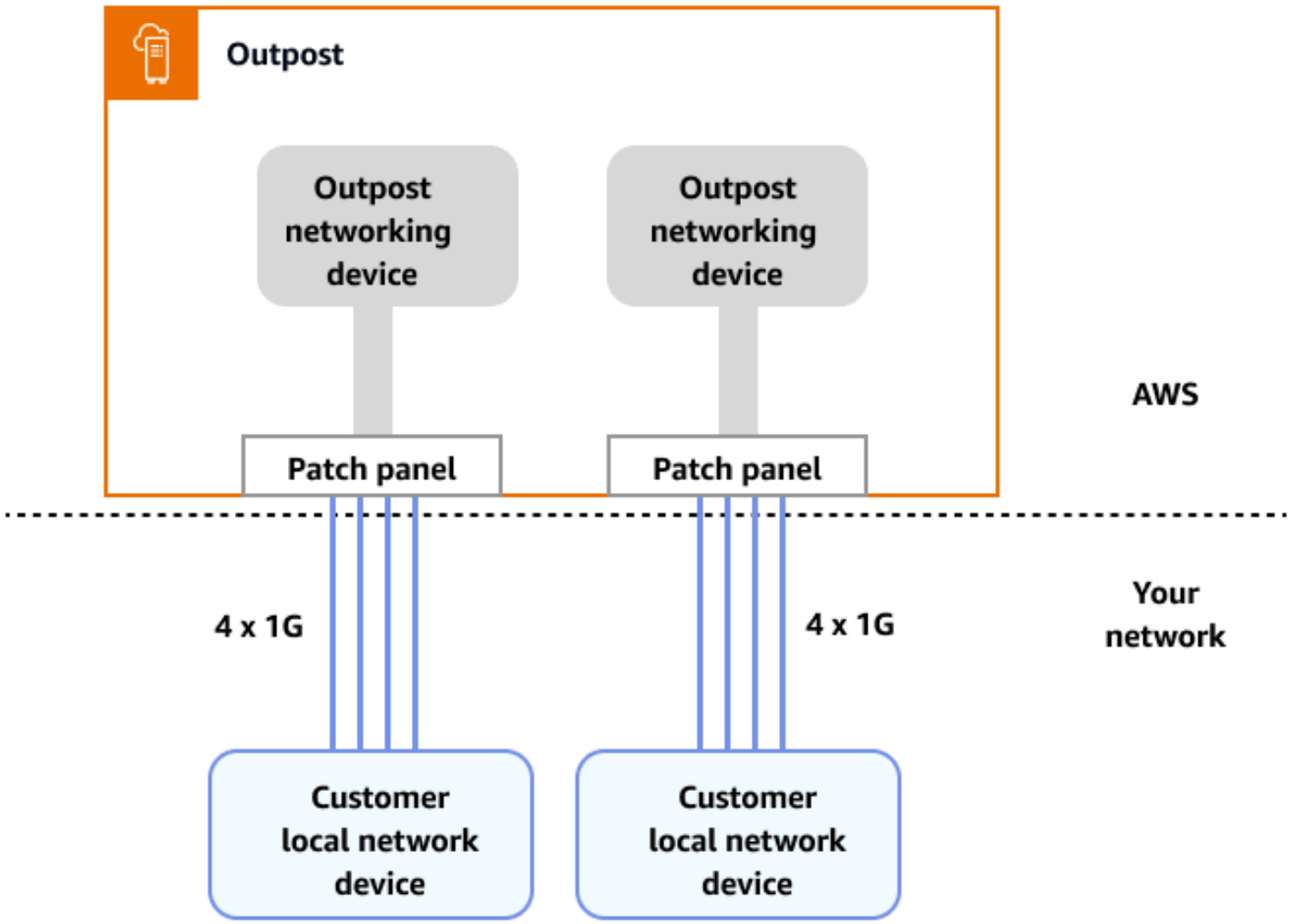
Local Zones are connected to the parent Region with multiple, redundant, secure, high-speed links that enable you to consume any Regional service, such as Amazon S3 and Amazon RDS,

seamlessly. You are responsible for providing connectivity from your on-premises environment or users to the Local Zone. Regardless of the connectivity architecture you choose (for example, VPN or AWS Direct Connect), the latency that must be achieved through the network links must be equivalent to avoid any impact on application performance in the event of a failure in a main link. If you're using AWS Direct Connect, the applicable resilience architectures are the same as those for accessing an AWS Region, as documented in AWS Direct Connect resiliency recommendations. However, there are scenarios that apply mostly to international Local Zones. In the country where the Local Zone is enabled, having only a single AWS Direct Connect PoP makes it impossible to create the architectures recommended for AWS Direct Connect resilience. If you have access to only a single AWS Direct Connect location or require resiliency beyond a single connection, you can create a VPN appliance on Amazon EC2 and AWS Direct Connect, as illustrated and discussed in the AWS blog post Enabling highly available connectivity from on premises to AWS Local Zones.

Resiliency networking for Outposts

In contrast to Local Zones, Outposts have redundant connectivity for accessing workloads deployed in Outposts from your local network. This redundancy is achieved through two Outposts network devices (ONDs). Each OND requires at least two fiber connections at 1 Gbps, 10 Gbps, 40 Gbps, or 100 Gbps to your local network. These connections must be configured as a link aggregation group (LAG) to allow for the scalable addition of more links.

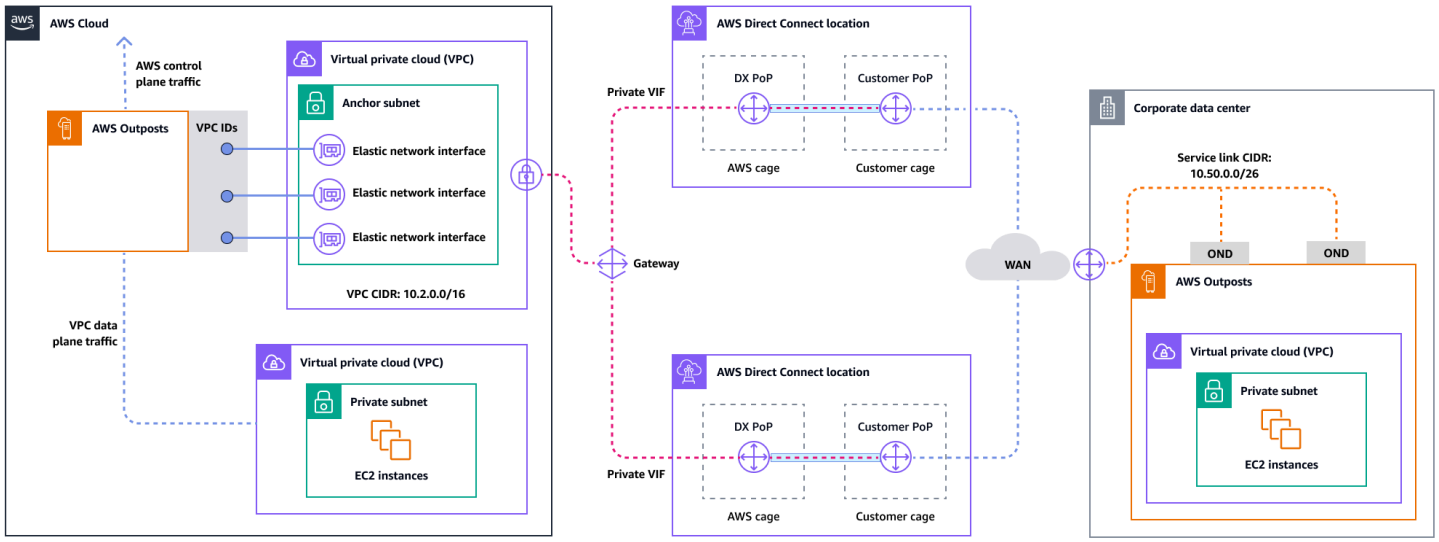
Uplink speed	Number of uplinks
1 Gbps	1, 2, 4, 6, or 8
10 Gbps	1, 2, 4, 8, 12, or 16
40 or 100 Gbps	1, 2, or 4



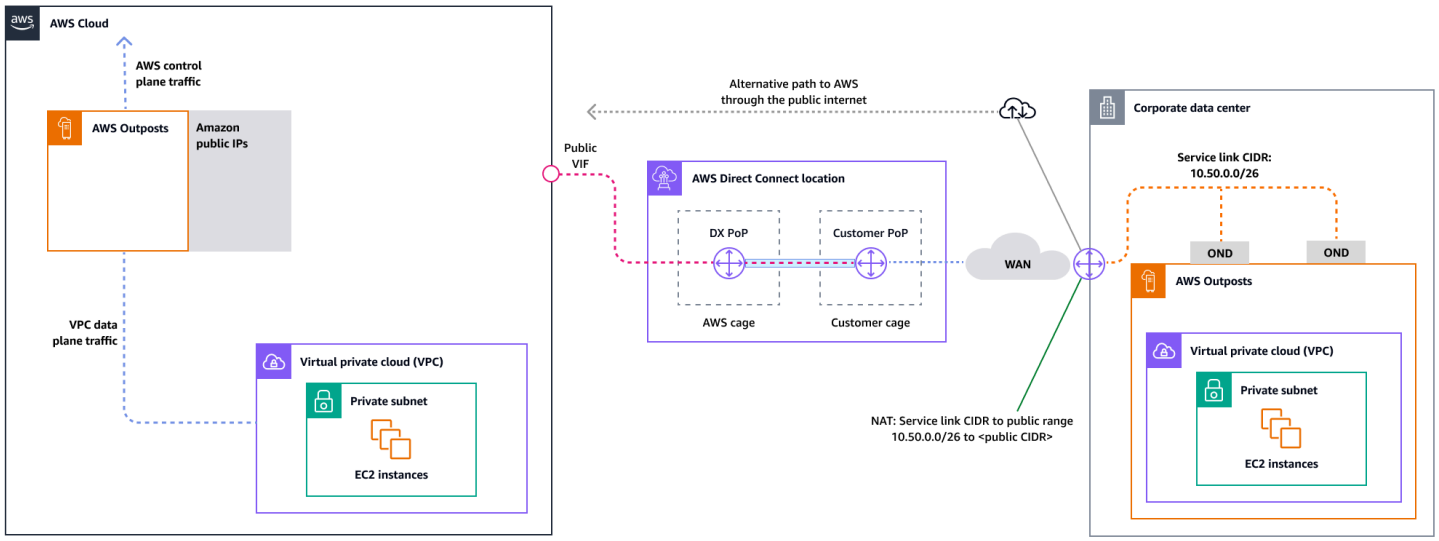
For more information about this connectivity, see Local network connectivity for Outposts Racks in the AWS Outposts documentation.

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

The following diagram illustrates this architecture for a highly available private connection.



The following diagram illustrates this architecture for a highly available public connection.



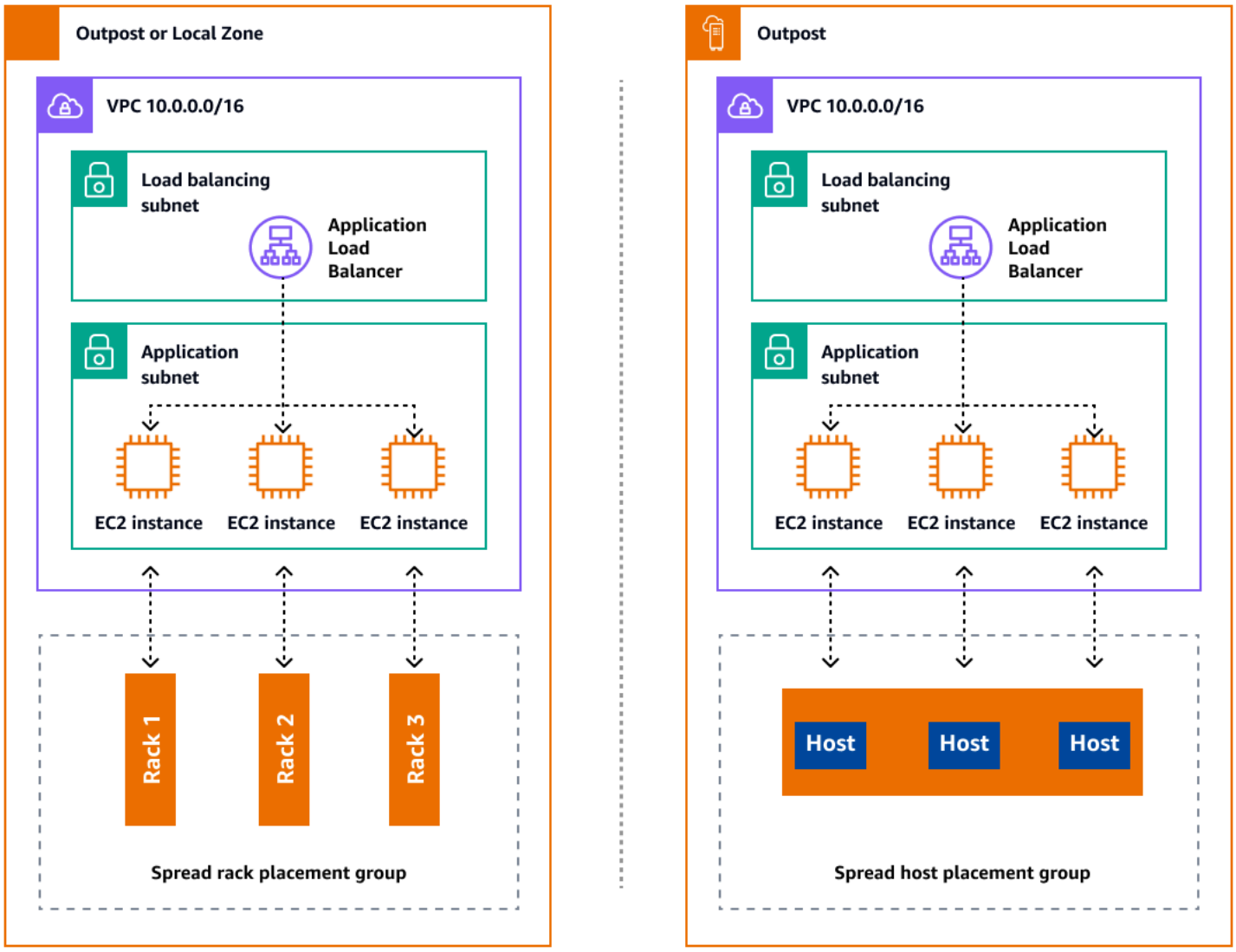
Scaling Outposts rack deployments with ACE racks

The Aggregation, Core, Edge (ACE) rack serves as a critical aggregation point for AWS Outposts multi-rack deployments, and is primarily recommended for installations that exceed three racks or for planning future expansion. Each ACE rack features four routers that support 10 Gbps, 40 Gbps, and 100 Gbps connections (100 Gbps is optimal). Each rack can connect to up to four upstream customer devices for maximum redundancy. ACE racks consume up to 10 kVA of power and weigh up to 705 lbs. Key benefits include reduced physical networking requirements, fewer fiber cabling uplinks, and decreased VLAN virtual interfaces. AWS monitors these racks through telemetry data via VPN tunnels and works closely with customers during installation to ensure proper power availability, network configuration, and optimal placement. The ACE rack architecture provides

increasing value as deployments scale, and effectively simplifies connectivity while reducing complexity and physical port requirements in larger installations. For more information, see the AWS blog post Scaling AWS Outposts rack deployments with ACE Rack.

Distributing instances across Outposts and Local Zones

Outposts and Local Zones have a finite number of compute servers. If your application deploys multiple related instances, these instances might deploy on the same server or on servers in the same rack unless they are configured differently. In addition to the default options, you can distribute instances across servers to mitigate the risk of running related instances on the same infrastructure. You can also distribute instances across multiple racks by using partition placement groups. This is called the *spread rack* distribution model. Use automatic distribution to spread instances across partitions in the group, or deploy instances to selected target partitions. By deploying instances to target partitions, you can deploy selected resources to the same rack while distributing other resources across racks. Outposts also provides another option called *spread host* that lets you distribute your workload at the host level. The following diagram shows the spread rack and spread host distribution options.



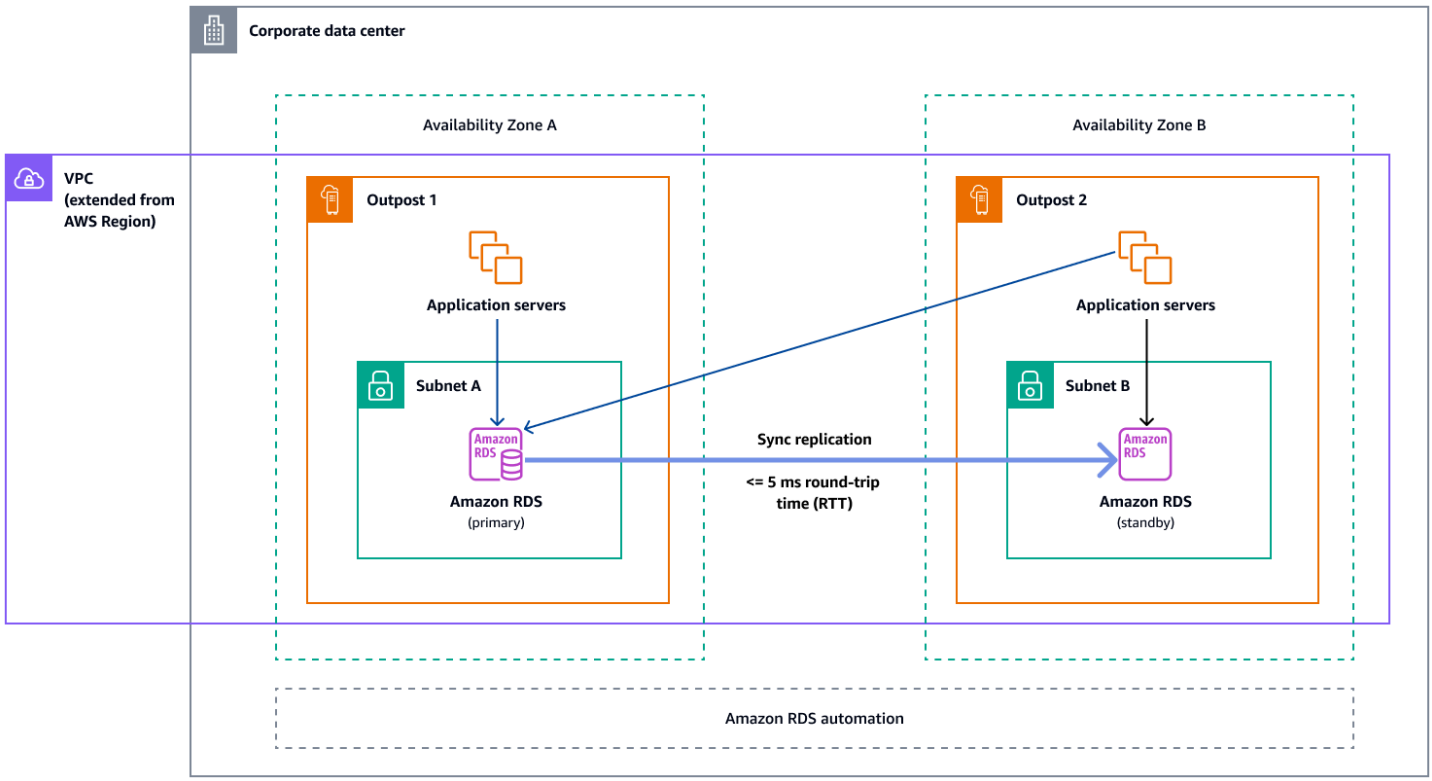
Amazon RDS Multi-AZ in AWS Outposts

When you use Multi-AZ instance deployments on Outposts, Amazon RDS creates two database instances across two Outposts. Each Outpost runs on its own physical infrastructure and connects to different Availability Zones in a Region for high availability. When two Outposts are connected through a customer-managed local connection, Amazon RDS manages synchronous replication between the primary and standby database instances. In case of a software or infrastructure failure, Amazon RDS automatically promotes the standby instance to the primary role and updates the DNS record to point to the new primary instance. For Multi-AZ deployments, Amazon RDS creates a primary DB instance on one Outpost and synchronously replicates the data to a standby DB instance on a different Outpost. Multi-AZ deployments on Outposts operate like Multi-AZ deployments in AWS Regions, with the following differences:

- They require a local connection between two or more Outposts.
- They require customer-owned IP (CoIP) address pools. For more information, see Customer-owned IP addresses for Amazon RDS on AWS Outposts in the Amazon RDS documentation.
- Replication runs on your local network.

Multi-AZ deployments are available for all supported versions of MySQL and PostgreSQL on Amazon RDS on Outposts. Local backups are not supported for Multi-AZ deployments.

The following diagram shows the architecture for Amazon RDS on Outposts Multi-AZ configurations.



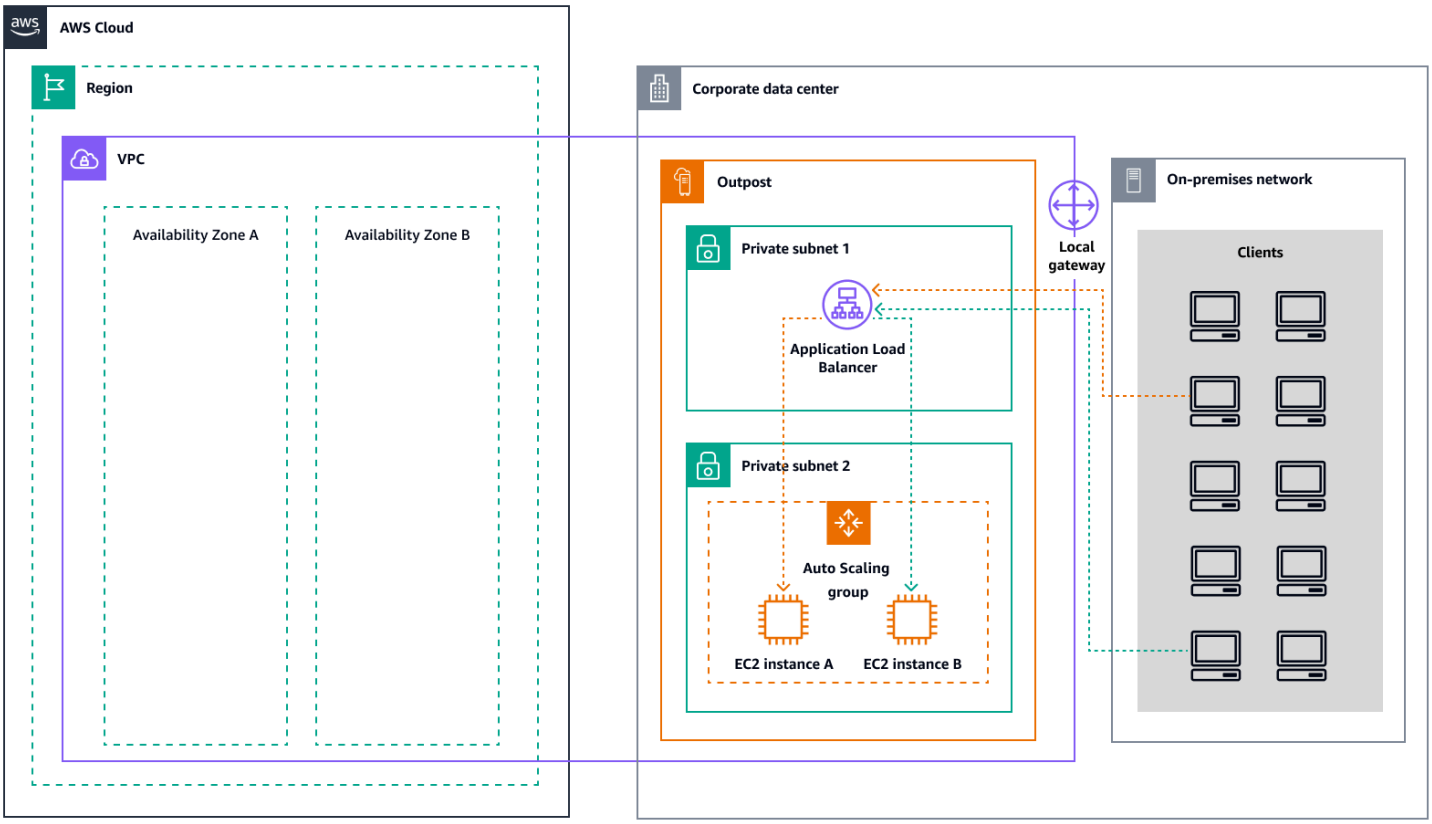
Failover mechanisms

Load balancing and automatic scaling

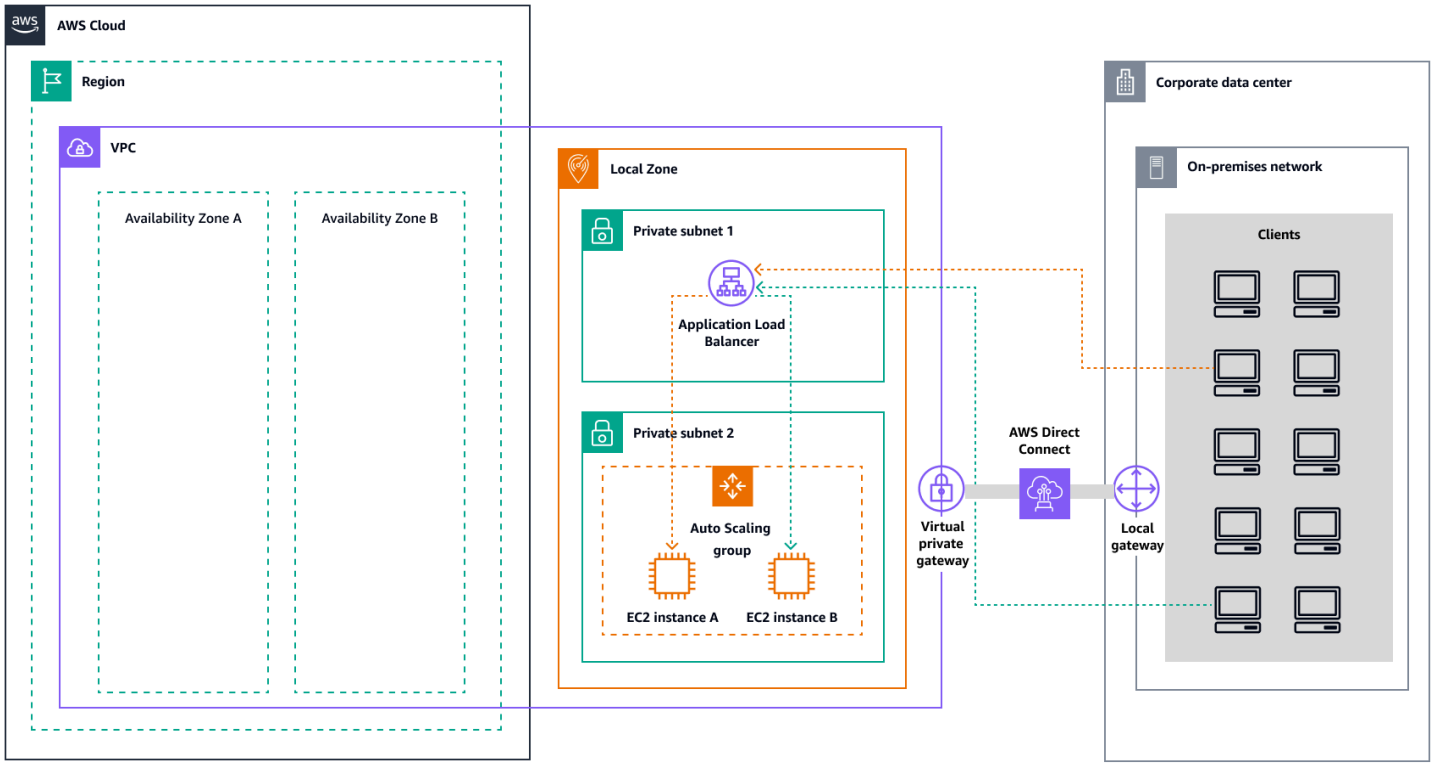
Elastic Load Balancing (ELB) automatically distributes your incoming application traffic across all the EC2 instances that you are running. ELB helps manage incoming requests by optimally routing traffic so that no single instance is overwhelmed. To use ELB with your Amazon EC2 Auto Scaling group, attach the load balancer to your Auto Scaling group. This registers the group with the load

balancer, which acts as a single point of contact for all incoming web traffic to your group. When you use ELB with your Auto Scaling group, it is not necessary to register individual EC2 instances with the load balancer. Instances that are launched by your Auto Scaling group are automatically registered with the load balancer. Similarly, instances that are terminated by your Auto Scaling group are automatically deregistered from the load balancer. After you attach a load balancer to your Auto Scaling group, you can configure your group to use ELB metrics (such as the Application Load Balancer request count per target) to scale the number of instances in the group as demand fluctuates. Optionally, you can add ELB health checks to your Auto Scaling group so that Amazon EC2 Auto Scaling can identify and replace unhealthy instances based on these health checks. You can also create an Amazon CloudWatch alarm that notifies you if the healthy host count of the target group is lower than allowed.

The following diagram illustrates how an Application Load Balancer manages workloads on Amazon EC2 in AWS Outposts.



The following diagram illustrates a similar architecture for Amazon EC2 in Local Zones.



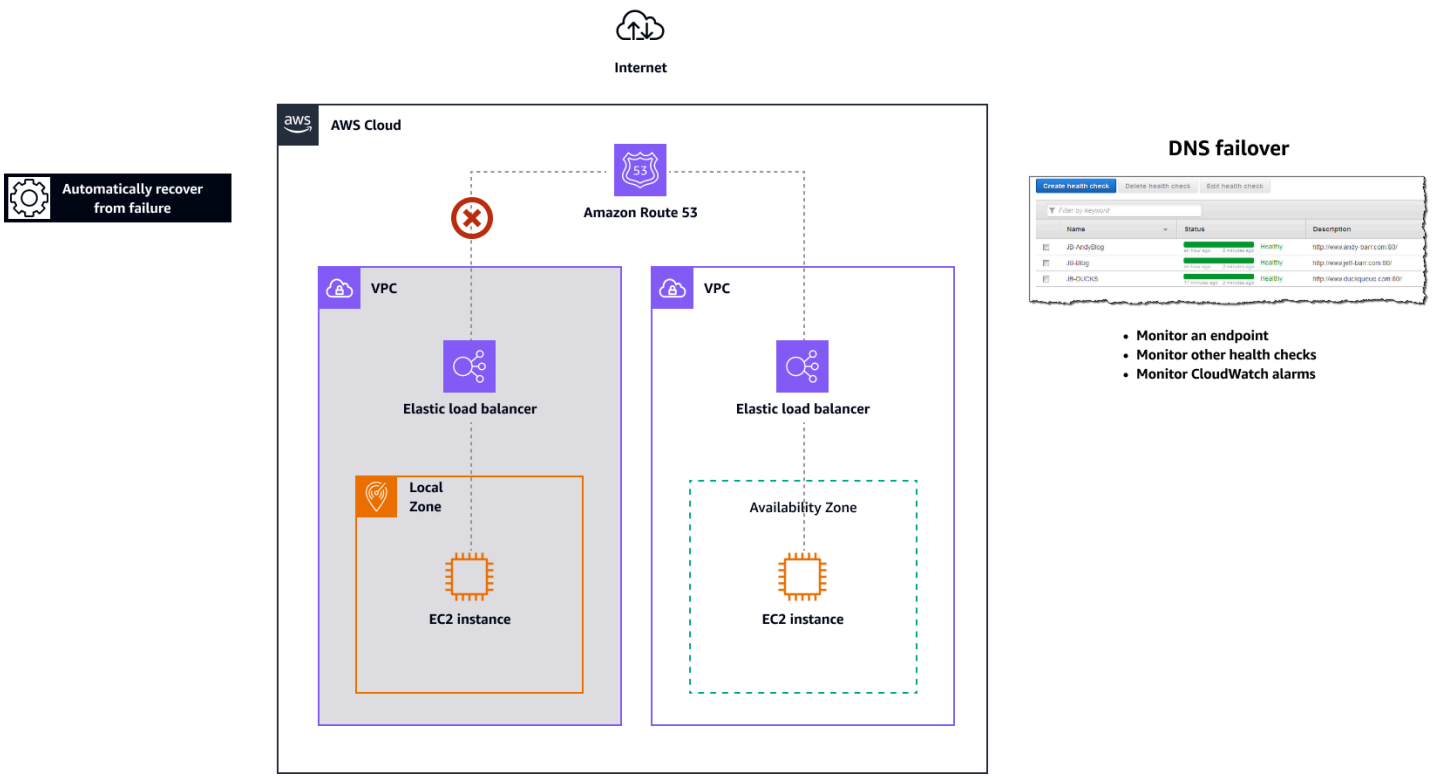
> **Note**
>
> Application Load Balancers are available in both AWS Outposts and Local Zones. However, to use an Application Load Balancer in AWS Outposts, you need to size the Amazon EC2 capacity to provide the scalability that the load balancer requires. For more information about sizing a load balancer in AWS Outposts, see the AWS blog post Configuring an Application Load Balancer on AWS Outposts.

Amazon Route 53 for DNS failover

When you have more than one resource performing the same function—for example, multiple HTTP or mail servers—you can configure Amazon Route 53 to check the health of your resources and respond to DNS queries by using only the healthy resources. For example, let's assume that your website, `example.com`, is hosted on two servers. One server is in a Local Zone and the other server is in an Outpost. You can configure Route 53 to check the health of those servers and to respond to DNS queries for `example.com` by using only the servers that are currently healthy. If you're using alias records to route traffic to selected AWS resources, such as ELB load balancers, you can configure Route 53 to evaluate the health of the resource and route traffic only to resources

that are healthy. When you configure an alias record to evaluate the health of a resource, you don't need to create a health check for that resource.

The following diagram illustrates Route 53 failover mechanisms.



Notes

- If you're creating failover records in a private hosted zone, you can create a CloudWatch metric, associate an alarm with the metric, and then create a health check that is based on the data stream for the alarm.
- To make an application publicly accessible in AWS Outposts by using an Application Load Balancer, set up networking configurations that enable Destination Network Address Translation (DNAT) from public IPs to the load balancer's fully qualified domain name (FQDN), and create a Route 53 failover rule with health checks that point to the exposed public IP. This combination ensures reliable public access to your Outposts-hosted application.

Amazon Route 53 Resolver on AWS Outposts

Amazon Route 53 Resolver is available on Outposts racks. It provides your on-premises services and applications with local DNS resolution directly from Outposts. Local Route 53 Resolver endpoints also enable DNS resolution between Outposts and your on-premises DNS server. Route 53 Resolver on Outposts helps improve the availability and performance of your on-premises applications.

One of the typical use cases for Outposts is to deploy applications that require low-latency access to on-premises systems, such as factory equipment, high-frequency trading applications, and medical diagnosis systems.

When you opt in to use local Route 53 Resolvers on Outposts, applications and services will continue to benefit from local DNS resolution to discover other services, even if connectivity to a parent AWS Region is lost. Local Resolvers also help reduce latency for DNS resolutions because query results are cached and served locally from the Outposts, which eliminates unnecessary round-trips to the parent AWS Region. All DNS resolutions for applications in Outposts VPCs that use private DNS are served locally.

In addition to enabling local Resolvers, this launch also enables local Resolver endpoints. Route 53 Resolver outbound endpoints enable Route 53 Resolvers to forward DNS queries to DNS resolvers that you manage—for example, on your on-premises network. In contrast, Route 53 Resolver inbound endpoints forward the DNS queries they receive from outside the VPC to the Resolver that's running on Outposts. It allows you to send DNS queries for services deployed on a private Outposts VPC from outside that VPC. For more information about inbound and outbound endpoints, see Resolving DNS queries between VPCs and your network in the Route 53 documentation.

Capacity planning at the edge

The capacity planning phase involves collecting the vCPU, memory, and storage requirements to deploy your architecture. In the cost optimization pillar of the AWS Well-Architected Framework, right-sizing is an ongoing process that starts with planning. You can use AWS tools to define optimizations based on resource consumption within AWS.

Edge capacity planning in Local Zones is the same as in AWS Regions. You should check to make sure that your instances are available in each Local Zone, because some instance types might differ from the types in AWS Regions. For Outposts, you should plan for capacity based on your workload

requirements. Outposts are slotted with fixed numbers of instances per host and can be reslotted as needed. If your workloads require spare capacity, take that into consideration when you plan your capacity needs.

Capacity planning on Outposts

AWS Outposts capacity planning requires specific inputs for Regional right-sizing, plus edge-specific factors that affect application availability, performance, and growth. For detailed guidance, see Capacity planning in the AWS whitepaper *AWS Outposts High Availability Design and Architecture Considerations*.

Capacity planning for Local Zones

A Local Zone is an extension of an AWS Region that is geographically close to your users. Resources that are created in a Local Zone can serve local users with very low-latency communications. To enable a Local Zone in your AWS account, review Getting started with AWS Local Zones in the AWS documentation. Each Local Zone has different slotting available for families of EC2 instances. Validate the instances available in each Local Zone before you use them. To confirm the available EC2 instances, run the following AWS CLI command:

```
aws ec2 describe-instance-type-offerings \
--location-type "availability-zone" \
--filters Name=location,Values=<local-zone-name>
```

Expected output:

```
{
  "InstanceTypeOfferings": [
      {
          "InstanceType": "m5.2xlarge",
          "LocationType": "availability-zone",
          "Location": "<local-zone-name>"
      },
      {
          "InstanceType": "t3.micro",
          "LocationType": "availability-zone",
          "Location": "local.zone-name"
      },
      ...
  ]
```

```
}
```

Edge infrastructure management

AWS provides fully managed services that extend AWS infrastructure, services, APIs, and tools closer to your end users and data centers. The services that are available in Outposts and Local Zones are the same as those available in AWS Regions, so you can manage those services by using the same AWS console, AWS CLI, or AWS APIs. For supported services, see the AWS Outposts feature comparison table and AWS Local Zones features.

Deploying services at the edge

You can configure the available services in Local Zones and Outposts in the same way you configure them in AWS Regions: by using the AWS console, AWS CLI, or AWS APIs. The primary difference between Regional and edge deployments is the subnets where resources will be provisioned. The Networking at the edge section described how subnets are deployed in Outposts and Local Zones. After you identify the edge subnets, you use the edge subnet ID as a parameter to deploy the service in Outposts or Local Zones. The following sections provide examples of deploying edge services.

Amazon EC2 at the edge

The following `run-instances` example launches a single instance of type `m5.2xlarge` into the edge subnet for the current Region. The key pair is optional if you do not plan to connect to your instance by using SSH on Linux or remote desktop protocol (RDP) on Windows.

```
aws ec2 run-instances \
    --image-id ami-id \
    --instance-type m5.2xlarge \
    --subnet-id <subnet-edge-id> \
    --key-name MyKeyPair
```

Application Load Balancers at the edge

The following `create-load-balancer` example creates an internal Application Load Balancer and enables the Local Zones or Outposts for the specified subnets.

```
aws elbv2 create-load-balancer \
```

```
    --name my-internal-load-balancer \
    --scheme internal \
    --subnets <subnet-edge-id>
```

To deploy an internet-facing Application Load Balancer to a subnet on an Outpost, you set the `internet-facing` flag in the `--scheme` option and provide a CoIP pool ID, as shown in this example:

```
aws elbv2 create-load-balancer \
    --name my-internal-load-balancer \
    --scheme internet-facing \
    --customer-owned-ipv4-pool <coip-pool-id>
    --subnets <subnet-edge-id>
```

For information about deploying other services at the edge, follow these links:

Service	AWS Outposts	AWS Local Zones
Amazon EKS	Deploy Amazon EKS on-premises with AWS Outposts	Launch low-latency EKS clusters with AWS Local Zones
Amazon ECS	Amazon ECS on AWS Outposts	Amazon ECS applications in shared subnets, Local Zones, and Wavelength Zones
Amazon RDS	Amazon RDS on AWS Outposts	Select the Local Zone subnet
Amazon S3	Getting started with Amazon S3 on Outposts	Not available
Amazon ElastiCache	Using Outposts with ElastiCache	Using Local Zones with ElastiCache
Amazon EMR	EMR clusters on AWS Outposts	EMR clusters on AWS Local Zones
Amazon FSx	Not available	Select the Local Zone subnet

Service	AWS Outposts	AWS Local Zones
AWS Elastic Disaster Recovery	Working with AWS Elastic Disaster Recovery and AWS Outposts	Not available
AWS Application Migration Service	Not available	Select the Local Zone subnet as the staging subnet

Outposts-specific CLI and SDK

AWS Outposts has two groups of commands and APIs for creating a service order or manipulating the routing tables between the local gateway and your local network.

Outposts ordering process

You can use the AWS CLI or the Outposts APIs to create an Outposts site, to create an Outpost, and to create an Outposts order. We recommend that you work with a hybrid cloud specialist during your AWS Outposts ordering process to ensure proper selection of resource IDs and optimal configuration for your implementation needs. For a complete resource ID list, see the AWS Outposts racks pricing page.

Local gateway management

The management and operation of the local gateway (LGW) in Outposts requires knowledge of the AWS CLI and SDK commands available for this task. You can use the AWS CLI and AWS SDKs to create and modify LGW routes, among other tasks. For more information about managing the LGW, see these resources:

- AWS CLI for Amazon EC2
- EC2.Client in the AWS SDK for Python (Boto)
- Ec2Client in the AWS SDK for Java

CloudWatch metrics and logs

For AWS services that are available in both Outposts and Local Zones, metrics and logs are managed in the same way as in Regions. Amazon CloudWatch provides metrics that are dedicated to monitoring Outposts in the following dimensions:

Dimension	Description
`Account`	The account or service using the capacity
`InstanceFamily`	The instance family
`InstanceType`	The instance type
`OutpostId`	The ID of the Outpost
`VolumeType`	The EBS volume type
`VirtualInterfaceId`	The ID of the local gateway or service link virtual interface (VIF)
`VirtualInterfaceGroupId`	The ID of the VIF group for the local gateway VIF

For more information, see CloudWatch metrics for Outposts racks in the Outposts documentation.

Resources

AWS references

- Hybrid Cloud with AWS
- AWS Outposts User Guide for Outposts racks
- AWS Local Zones User Guide
- AWS Outposts Family
- AWS Local Zones
- Extend a VPC to a Local Zone, Wavelength Zone, or Outpost (Amazon VPC documentation)
- Linux instances in Local Zones (Amazon EC2 documentation)
- Linux instances in Outposts (Amazon EC2 documentation)
- Get Started Deploying Low Latency Applications with AWS Local Zones (tutorial)

AWS blog posts

- Running AWS infrastructure on premises with Amazon EC2
- Building modern applications with Amazon EKS on Amazon EC2
- How to choose between CoIP and direct VPC routing modes on Amazon EC2 rack
- Selecting network switches for your Amazon EC2
- Maintaining a local copy of your data in AWS Local Zones
- Amazon ECS on Amazon EC2
- Managing edge-aware service mesh with Amazon EKS for AWS Local Zones
- Deploying local gateway ingress routing on Amazon EC2
- Automating your workload deployments in AWS Local Zones
- Sharing Amazon EC2 in a multi account AWS environment: Part 1
- Sharing Amazon EC2 in a multi account AWS environment: Part 2
- AWS Direct Connect and AWS Local Zones interoperability patterns
- Deploy Amazon RDS on Amazon EC2 with Multi-AZ high availability

Contributors

The following individuals contributed to this guide.

Authoring

- Leonardo Solano, Principal Hybrid Cloud Solutions Architect, AWS
- Len Gomes, Partner Solutions Architect, AWS
- Matt Price, Senior Enterprise Support Engineer, AWS
- Tom Gadomski, Solutions Architect, AWS
- Obed Gutierrez, Solutions Architect, AWS
- Dionysios Kakaletris, Technical Account Manager, AWS
- Vamsi Krishna, Principal Outposts Specialist, AWS

Reviewing

- David Filiatrault, Delivery Consultant, AWS

Technical writing

- Handan Selamoglu, Sr. Documentation Manager, AWS

Document history

The following table describes significant changes to this guide. If you want to be notified about future updates, you can subscribe to an RSS feed.

Change	Description	Date
Initial publication	—	June 10, 2025

AWS Prescriptive Guidance glossary

The following are commonly used terms in strategies, guides, and patterns provided by AWS Prescriptive Guidance. To suggest entries, please use the **Provide feedback** link at the end of the glossary.

Numbers

7 Rs

Seven common migration strategies for moving applications to the cloud. These strategies build upon the 5 Rs that Gartner identified in 2011 and consist of the following:

- Refactor/re-architect – Move an application and modify its architecture by taking full advantage of cloud-native features to improve agility, performance, and scalability. This typically involves porting the operating system and database. Example: Migrate your on-premises Oracle database to the Amazon Aurora PostgreSQL-Compatible Edition.
- Replatform (lift and reshape) – Move an application to the cloud, and introduce some level of optimization to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Amazon Relational Database Service (Amazon RDS) for Oracle in the AWS Cloud.
- Repurchase (drop and shop) – Switch to a different product, typically by moving from a traditional license to a SaaS model. Example: Migrate your customer relationship management (CRM) system to Salesforce.com.
- Rehost (lift and shift) – Move an application to the cloud without making any changes to take advantage of cloud capabilities. Example: Migrate your on-premises Oracle database to Oracle on an EC2 instance in the AWS Cloud.
- Relocate (hypervisor-level lift and shift) – Move infrastructure to the cloud without purchasing new hardware, rewriting applications, or modifying your existing operations. You migrate servers from an on-premises platform to a cloud service for the same platform. Example: Migrate a Microsoft Hyper-V application to AWS.
- Retain (revisit) – Keep applications in your source environment. These might include applications that require major refactoring, and you want to postpone that work until a later time, and legacy applications that you want to retain, because there's no business justification for migrating them.

- Retire – Decommission or remove applications that are no longer needed in your source environment.

A

ABAC

See attribute-based access control.

abstracted services

See managed services.

ACID

See atomicity, consistency, isolation, durability.

active-active migration

A database migration method in which the source and target databases are kept in sync (by using a bidirectional replication tool or dual write operations), and both databases handle transactions from connecting applications during migration. This method supports migration in small, controlled batches instead of requiring a one-time cutover. It's more flexible but requires more work than active-passive migration.

active-passive migration

A database migration method in which the source and target databases are kept in sync, but only the source database handles transactions from connecting applications while data is replicated to the target database. The target database doesn't accept any transactions during migration.

aggregate function

A SQL function that operates on a group of rows and calculates a single return value for the group. Examples of aggregate functions include `SUM` and `MAX`.

AI

See artificial intelligence.

AIOps

See artificial intelligence operations.

anonymization

The process of permanently deleting personal information in a dataset. Anonymization can help protect personal privacy. Anonymized data is no longer considered to be personal data.

anti-pattern

A frequently used solution for a recurring issue where the solution is counter-productive, ineffective, or less effective than an alternative.

application control

A security approach that allows the use of only approved applications in order to help protect a system from malware.

application portfolio

A collection of detailed information about each application used by an organization, including the cost to build and maintain the application, and its business value. This information is key to the portfolio discovery and analysis process and helps identify and prioritize the applications to be migrated, modernized, and optimized.

artificial intelligence (AI)

The field of computer science that is dedicated to using computing technologies to perform cognitive functions that are typically associated with humans, such as learning, solving problems, and recognizing patterns. For more information, see What is Artificial Intelligence?

artificial intelligence operations (AIOps)

The process of using machine learning techniques to solve operational problems, reduce operational incidents and human intervention, and increase service quality. For more information about how AIOps is used in the AWS migration strategy, see the operations integration guide.

asymmetric encryption

An encryption algorithm that uses a pair of keys, a public key for encryption and a private key for decryption. You can share the public key because it isn't used for decryption, but access to the private key should be highly restricted.

atomicity, consistency, isolation, durability (ACID)

A set of software properties that guarantee the data validity and operational reliability of a database, even in the case of errors, power failures, or other problems.

attribute-based access control (ABAC)

The practice of creating fine-grained permissions based on user attributes, such as department, job role, and team name. For more information, see ABAC for AWS in the AWS Identity and Access Management (IAM) documentation.

authoritative data source

A location where you store the primary version of data, which is considered to be the most reliable source of information. You can copy data from the authoritative data source to other locations for the purposes of processing or modifying the data, such as anonymizing, redacting, or pseudonymizing it.

Availability Zone

A distinct location within an AWS Region that is insulated from failures in other Availability Zones and provides inexpensive, low-latency network connectivity to other Availability Zones in the same Region.

AWS Cloud Adoption Framework (AWS CAF)

A framework of guidelines and best practices from AWS to help organizations develop an efficient and effective plan to move successfully to the cloud. AWS CAF organizes guidance into six focus areas called perspectives: business, people, governance, platform, security, and operations. The business, people, and governance perspectives focus on business skills and processes; the platform, security, and operations perspectives focus on technical skills and processes. For example, the people perspective targets stakeholders who handle human resources (HR), staffing functions, and people management. For this perspective, AWS CAF provides guidance for people development, training, and communications to help ready the organization for successful cloud adoption. For more information, see the AWS CAF website and the AWS CAF whitepaper.

AWS Workload Qualification Framework (AWS WQF)

A tool that evaluates database migration workloads, recommends migration strategies, and provides work estimates. AWS WQF is included with AWS Schema Conversion Tool (AWS SCT). It analyzes database schemas and code objects, application code, dependencies, and performance characteristics, and provides assessment reports.

B

bad bot

A bot that is intended to disrupt or cause harm to individuals or organizations.

BCP

See business continuity planning.

behavior graph

A unified, interactive view of resource behavior and interactions over time. You can use a behavior graph with Amazon Detective to examine failed logon attempts, suspicious API calls, and similar actions. For more information, see Data in a behavior graph in the Detective documentation.

big-endian system

A system that stores the most significant byte first. See also endianness.

binary classification

A process that predicts a binary outcome (one of two possible classes). For example, your ML model might need to predict problems such as “Is this email spam or not spam?" or "Is this product a book or a car?"

bloom filter

A probabilistic, memory-efficient data structure that is used to test whether an element is a member of a set.

blue/green deployment

A deployment strategy where you create two separate but identical environments. You run the current application version in one environment (blue) and the new application version in the other environment (green). This strategy helps you quickly roll back with minimal impact.

bot

A software application that runs automated tasks over the internet and simulates human activity or interaction. Some bots are useful or beneficial, such as web crawlers that index information on the internet. Some other bots, known as *bad bots*, are intended to disrupt or cause harm to individuals or organizations.

botnet

Networks of bots that are infected by malware and are under the control of a single party, known as a *bot herder* or *bot operator*. Botnets are the best-known mechanism to scale bots and their impact.

branch

A contained area of a code repository. The first branch created in a repository is the *main branch*. You can create a new branch from an existing branch, and you can then develop features or fix bugs in the new branch. A branch you create to build a feature is commonly referred to as a *feature branch*. When the feature is ready for release, you merge the feature branch back into the main branch. For more information, see About branches (GitHub documentation).

break-glass access

In exceptional circumstances and through an approved process, a quick means for a user to gain access to an AWS account that they don't typically have permissions to access. For more information, see the Implement break-glass procedures indicator in the AWS Well-Architected guidance.

brownfield strategy

The existing infrastructure in your environment. When adopting a brownfield strategy for a system architecture, you design the architecture around the constraints of the current systems and infrastructure. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

buffer cache

The memory area where the most frequently accessed data is stored.

business capability

What a business does to generate value (for example, sales, customer service, or marketing). Microservices architectures and development decisions can be driven by business capabilities. For more information, see the Organized around business capabilities section of the Running containerized microservices on AWS whitepaper.

business continuity planning (BCP)

A plan that addresses the potential impact of a disruptive event, such as a large-scale migration, on operations and enables a business to resume operations quickly.

C

CAF

See AWS Cloud Adoption Framework.

canary deployment

The slow and incremental release of a version to end users. When you are confident, you deploy the new version and replace the current version in its entirety.

CCoE

See Cloud Center of Excellence.

CDC

See change data capture.

change data capture (CDC)

The process of tracking changes to a data source, such as a database table, and recording metadata about the change. You can use CDC for various purposes, such as auditing or replicating changes in a target system to maintain synchronization.

chaos engineering

Intentionally introducing failures or disruptive events to test a system's resilience. You can use AWS Fault Injection Service (AWS FIS) to perform experiments that stress your AWS workloads and evaluate their response.

CI/CD

See continuous integration and continuous delivery.

classification

A categorization process that helps generate predictions. ML models for classification problems predict a discrete value. Discrete values are always distinct from one another. For example, a model might need to evaluate whether or not there is a car in an image.

client-side encryption

Encryption of data locally, before the target AWS service receives it.

Cloud Center of Excellence (CCoE)

A multi-disciplinary team that drives cloud adoption efforts across an organization, including developing cloud best practices, mobilizing resources, establishing migration timelines, and leading the organization through large-scale transformations. For more information, see the CCoE posts on the AWS Cloud Enterprise Strategy Blog.

cloud computing

The cloud technology that is typically used for remote data storage and IoT device management. Cloud computing is commonly connected to edge computing technology.

cloud operating model

In an IT organization, the operating model that is used to build, mature, and optimize one or more cloud environments. For more information, see Building your Cloud Operating Model.

cloud stages of adoption

The four phases that organizations typically go through when they migrate to the AWS Cloud:

- Project – Running a few cloud-related projects for proof of concept and learning purposes
- Foundation – Making foundational investments to scale your cloud adoption (e.g., creating a landing zone, defining a CCoE, establishing an operations model)
- Migration – Migrating individual applications
- Re-invention – Optimizing products and services, and innovating in the cloud

These stages were defined by Stephen Orban in the blog post The Journey Toward Cloud-First & the Stages of Adoption on the AWS Cloud Enterprise Strategy blog. For information about how they relate to the AWS migration strategy, see the migration readiness guide.

CMDB

See configuration management database.

code repository

A location where source code and other assets, such as documentation, samples, and scripts, are stored and updated through version control processes. Common cloud repositories include GitHub or Bitbucket Cloud. Each version of the code is called a *branch*. In a microservice structure, each repository is devoted to a single piece of functionality. A single CI/CD pipeline can use multiple repositories.

cold cache

A buffer cache that is empty, not well populated, or contains stale or irrelevant data. This affects performance because the database instance must read from the main memory or disk, which is slower than reading from the buffer cache.

cold data

Data that is rarely accessed and is typically historical. When querying this kind of data, slow queries are typically acceptable. Moving this data to lower-performing and less expensive storage tiers or classes can reduce costs.

computer vision (CV)

A field of AI that uses machine learning to analyze and extract information from visual formats such as digital images and videos. For example, Amazon SageMaker AI provides image processing algorithms for CV.

configuration drift

For a workload, a configuration change from the expected state. It might cause the workload to become noncompliant, and it's typically gradual and unintentional.

configuration management database (CMDB)

A repository that stores and manages information about a database and its IT environment, including both hardware and software components and their configurations. You typically use data from a CMDB in the portfolio discovery and analysis stage of migration.

conformance pack

A collection of AWS Config rules and remediation actions that you can assemble to customize your compliance and security checks. You can deploy a conformance pack as a single entity in an AWS account and Region, or across an organization, by using a YAML template. For more information, see Conformance packs in the AWS Config documentation.

continuous integration and continuous delivery (CI/CD)

The process of automating the source, build, test, staging, and production stages of the software release process. CI/CD is commonly described as a pipeline. CI/CD can help you automate processes, improve productivity, improve code quality, and deliver faster. For more information, see Benefits of continuous delivery. CD can also stand for *continuous deployment*. For more information, see Continuous Delivery vs. Continuous Deployment.

CV

See computer vision.

D

data at rest

Data that is stationary in your network, such as data that is in storage.

data classification

A process for identifying and categorizing the data in your network based on its criticality and sensitivity. It is a critical component of any cybersecurity risk management strategy because it helps you determine the appropriate protection and retention controls for the data. Data classification is a component of the security pillar in the AWS Well-Architected Framework. For more information, see Data classification.

data drift

A meaningful variation between the production data and the data that was used to train an ML model, or a meaningful change in the input data over time. Data drift can reduce the overall quality, accuracy, and fairness in ML model predictions.

data in transit

Data that is actively moving through your network, such as between network resources.

data mesh

An architectural framework that provides distributed, decentralized data ownership with centralized management and governance.

data minimization

The principle of collecting and processing only the data that is strictly necessary. Practicing data minimization in the AWS Cloud can reduce privacy risks, costs, and your analytics carbon footprint.

data perimeter

A set of preventive guardrails in your AWS environment that help make sure that only trusted identities are accessing trusted resources from expected networks. For more information, see Building a data perimeter on AWS.

data preprocessing

To transform raw data into a format that is easily parsed by your ML model. Preprocessing data can mean removing certain columns or rows and addressing missing, inconsistent, or duplicate values.

data provenance

The process of tracking the origin and history of data throughout its lifecycle, such as how the data was generated, transmitted, and stored.

data subject

An individual whose data is being collected and processed.

data warehouse

A data management system that supports business intelligence, such as analytics. Data warehouses commonly contain large amounts of historical data, and they are typically used for queries and analysis.

database definition language (DDL)

Statements or commands for creating or modifying the structure of tables and objects in a database.

database manipulation language (DML)

Statements or commands for modifying (inserting, updating, and deleting) information in a database.

DDL

See database definition language.

deep ensemble

To combine multiple deep learning models for prediction. You can use deep ensembles to obtain a more accurate prediction or for estimating uncertainty in predictions.

deep learning

An ML subfield that uses multiple layers of artificial neural networks to identify mapping between input data and target variables of interest.

defense-in-depth

An information security approach in which a series of security mechanisms and controls are thoughtfully layered throughout a computer network to protect the confidentiality, integrity, and availability of the network and the data within. When you adopt this strategy on AWS, you add multiple controls at different layers of the AWS Organizations structure to help secure resources. For example, a defense-in-depth approach might combine multi-factor authentication, network segmentation, and encryption.

delegated administrator

In AWS Organizations, a compatible service can register an AWS member account to administer the organization's accounts and manage permissions for that service. This account is called the *delegated administrator* for that service. For more information and a list of compatible services, see Services that work with AWS Organizations in the AWS Organizations documentation.

deployment

The process of making an application, new features, or code fixes available in the target environment. Deployment involves implementing changes in a code base and then building and running that code base in the application's environments.

development environment

See environment.

detective control

A security control that is designed to detect, log, and alert after an event has occurred. These controls are a second line of defense, alerting you to security events that bypassed the preventative controls in place. For more information, see Detective controls in *Implementing security controls on AWS*.

development value stream mapping (DVSM)

A process used to identify and prioritize constraints that adversely affect speed and quality in a software development lifecycle. DVSM extends the value stream mapping process originally designed for lean manufacturing practices. It focuses on the steps and teams required to create and move value through the software development process.

digital twin

A virtual representation of a real-world system, such as a building, factory, industrial equipment, or production line. Digital twins support predictive maintenance, remote monitoring, and production optimization.

dimension table

In a star schema, a smaller table that contains data attributes about quantitative data in a fact table. Dimension table attributes are typically text fields or discrete numbers that behave like text. These attributes are commonly used for query constraining, filtering, and result set labeling.

disaster

An event that prevents a workload or system from fulfilling its business objectives in its primary deployed location. These events can be natural disasters, technical failures, or the result of human actions, such as unintentional misconfiguration or a malware attack.

disaster recovery (DR)

The strategy and process you use to minimize downtime and data loss caused by a disaster. For more information, see Disaster Recovery of Workloads on AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

See database manipulation language.

domain-driven design

An approach to developing a complex software system by connecting its components to evolving domains, or core business goals, that each component serves. This concept was introduced by Eric Evans in his book, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). For information about how you can use domain-driven design with the strangler fig pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

DR

See disaster recovery.

drift detection

Tracking deviations from a baselined configuration. For example, you can use AWS CloudFormation to detect drift in system resources, or you can use AWS Control Tower to detect changes in your landing zone that might affect compliance with governance requirements.

DVSM

See development value stream mapping.

E

EDA

See exploratory data analysis.

EDI

See electronic data interchange.

edge computing

The technology that increases the computing power for smart devices at the edges of an IoT network. When compared with cloud computing, edge computing can reduce communication latency and improve response time.

electronic data interchange (EDI)

The automated exchange of business documents between organizations. For more information, see What is Electronic Data Interchange.

encryption

A computing process that transforms plaintext data, which is human-readable, into ciphertext.

encryption key

A cryptographic string of randomized bits that is generated by an encryption algorithm. Keys can vary in length, and each key is designed to be unpredictable and unique.

endianness

The order in which bytes are stored in computer memory. Big-endian systems store the most significant byte first. Little-endian systems store the least significant byte first.

endpoint

See service endpoint.

endpoint service

A service that you can host in a virtual private cloud (VPC) to share with other users. You can create an endpoint service with AWS PrivateLink and grant permissions to other AWS accounts or to AWS Identity and Access Management (IAM) principals. These accounts or principals can connect to your endpoint service privately by creating interface VPC endpoints. For more information, see Create an endpoint service in the Amazon Virtual Private Cloud (Amazon VPC) documentation.

enterprise resource planning (ERP)

A system that automates and manages key business processes (such as accounting, MES, and project management) for an enterprise.

envelope encryption

The process of encrypting an encryption key with another encryption key. For more information, see Envelope encryption in the AWS Key Management Service (AWS KMS) documentation.

environment

An instance of a running application. The following are common types of environments in cloud computing:

- development environment – An instance of a running application that is available only to the core team responsible for maintaining the application. Development environments are used to test changes before promoting them to upper environments. This type of environment is sometimes referred to as a *test environment*.
- lower environments – All development environments for an application, such as those used for initial builds and tests.
- production environment – An instance of a running application that end users can access. In a CI/CD pipeline, the production environment is the last deployment environment.
- upper environments – All environments that can be accessed by users other than the core development team. This can include a production environment, preproduction environments, and environments for user acceptance testing.

epic

In agile methodologies, functional categories that help organize and prioritize your work. Epics provide a high-level description of requirements and implementation tasks. For example, AWS CAF security epics include identity and access management, detective controls, infrastructure security, data protection, and incident response. For more information about epics in the AWS migration strategy, see the program implementation guide.

ERP

See enterprise resource planning.

exploratory data analysis (EDA)

The process of analyzing a dataset to understand its main characteristics. You collect or aggregate data and then perform initial investigations to find patterns, detect anomalies, and check assumptions. EDA is performed by calculating summary statistics and creating data visualizations.

F

fact table

The central table in a star schema. It stores quantitative data about business operations. Typically, a fact table contains two types of columns: those that contain measures and those that contain a foreign key to a dimension table.

fail fast

A philosophy that uses frequent and incremental testing to reduce the development lifecycle. It is a critical part of an agile approach.

fault isolation boundary

In the AWS Cloud, a boundary such as an Availability Zone, AWS Region, control plane, or data plane that limits the effect of a failure and helps improve the resilience of workloads. For more information, see AWS Fault Isolation Boundaries.

feature branch

See branch.

features

The input data that you use to make a prediction. For example, in a manufacturing context, features could be images that are periodically captured from the manufacturing line.

feature importance

How significant a feature is for a model's predictions. This is usually expressed as a numerical score that can be calculated through various techniques, such as Shapley Additive Explanations (SHAP) and integrated gradients. For more information, see Machine learning model interpretability with AWS.

feature transformation

To optimize data for the ML process, including enriching data with additional sources, scaling values, or extracting multiple sets of information from a single data field. This enables the ML model to benefit from the data. For example, if you break down the "2021-05-27 00:15:37" date into "2021", "May", "Thu", and "15", you can help the learning algorithm learn nuanced patterns associated with different data components.

few-shot prompting

Providing an LLM with a small number of examples that demonstrate the task and desired output before asking it to perform a similar task. This technique is an application of in-context learning, where models learn from examples (*shots*) that are embedded in prompts. Few-shot prompting can be effective for tasks that require specific formatting, reasoning, or domain knowledge. See also zero-shot prompting.

FGAC

See fine-grained access control.

fine-grained access control (FGAC)

The use of multiple conditions to allow or deny an access request.

flash-cut migration

A database migration method that uses continuous data replication through change data capture to migrate data in the shortest time possible, instead of using a phased approach. The objective is to keep downtime to a minimum.

FM

See foundation model.

foundation model (FM)

A large deep-learning neural network that has been training on massive datasets of generalized and unlabeled data. FMs are capable of performing a wide variety of general tasks, such as understanding language, generating text and images, and conversing in natural language. For more information, see What are Foundation Models.

G

generative AI

A subset of AI models that have been trained on large amounts of data and that can use a simple text prompt to create new content and artifacts, such as images, videos, text, and audio. For more information, see What is Generative AI.

geo blocking

See geographic restrictions.

geographic restrictions (geo blocking)

In Amazon CloudFront, an option to prevent users in specific countries from accessing content distributions. You can use an allow list or block list to specify approved and banned countries. For more information, see Restricting the geographic distribution of your content in the CloudFront documentation.

Gitflow workflow

An approach in which lower and upper environments use different branches in a source code repository. The Gitflow workflow is considered legacy, and the trunk-based workflow is the modern, preferred approach.

golden image

A snapshot of a system or software that is used as a template to deploy new instances of that system or software. For example, in manufacturing, a golden image can be used to provision software on multiple devices and helps improve speed, scalability, and productivity in device manufacturing operations.

greenfield strategy

The absence of existing infrastructure in a new environment. When adopting a greenfield strategy for a system architecture, you can select all new technologies without the restriction

of compatibility with existing infrastructure, also known as brownfield. If you are expanding the existing infrastructure, you might blend brownfield and greenfield strategies.

guardrail

A high-level rule that helps govern resources, policies, and compliance across organizational units (OUs). *Preventive guardrails* enforce policies to ensure alignment to compliance standards. They are implemented by using service control policies and IAM permissions boundaries. *Detective guardrails* detect policy violations and compliance issues, and generate alerts for remediation. They are implemented by using AWS Config, AWS Security Hub, Amazon GuardDuty, AWS Trusted Advisor, Amazon Inspector, and custom AWS Lambda checks.

H

HA

See high availability.

heterogeneous database migration

Migrating your source database to a target database that uses a different database engine (for example, Oracle to Amazon Aurora). Heterogeneous migration is typically part of a re-architecting effort, and converting the schema can be a complex task. AWS provides AWS SCT that helps with schema conversions.

high availability (HA)

The ability of a workload to operate continuously, without intervention, in the event of challenges or disasters. HA systems are designed to automatically fail over, consistently deliver high-quality performance, and handle different loads and failures with minimal performance impact.

historian modernization

An approach used to modernize and upgrade operational technology (OT) systems to better serve the needs of the manufacturing industry. A *historian* is a type of database that is used to collect and store data from various sources in a factory.

holdout data

A portion of historical, labeled data that is withheld from a dataset that is used to train a machine learning model. You can use holdout data to evaluate the model performance by comparing the model predictions against the holdout data.

homogeneous database migration

Migrating your source database to a target database that shares the same database engine (for example, Microsoft SQL Server to Amazon RDS for SQL Server). Homogeneous migration is typically part of a rehosting or replatforming effort. You can use native database utilities to migrate the schema.

hot data

Data that is frequently accessed, such as real-time data or recent translational data. This data typically requires a high-performance storage tier or class to provide fast query responses.

hotfix

An urgent fix for a critical issue in a production environment. Due to its urgency, a hotfix is usually made outside of the typical DevOps release workflow.

hypercare period

Immediately following cutover, the period of time when a migration team manages and monitors the migrated applications in the cloud in order to address any issues. Typically, this period is 1–4 days in length. At the end of the hypercare period, the migration team typically transfers responsibility for the applications to the cloud operations team.

I

IaC

See infrastructure as code.

identity-based policy

A policy attached to one or more IAM principals that defines their permissions within the AWS Cloud environment.

idle application

An application that has an average CPU and memory usage between 5 and 20 percent over a period of 90 days. In a migration project, it is common to retire these applications or retain them on premises.

IIoT

See industrial Internet of Things.

immutable infrastructure

A model that deploys new infrastructure for production workloads instead of updating, patching, or modifying the existing infrastructure. Immutable infrastructures are inherently more consistent, reliable, and predictable than mutable infrastructure. For more information, see the Deploy using immutable infrastructure best practice in the AWS Well-Architected Framework.

inbound (ingress) VPC

In an AWS multi-account architecture, a VPC that accepts, inspects, and routes network connections from outside an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

incremental migration

A cutover strategy in which you migrate your application in small parts instead of performing a single, full cutover. For example, you might move only a few microservices or users to the new system initially. After you verify that everything is working properly, you can incrementally move additional microservices or users until you can decommission your legacy system. This strategy reduces the risks associated with large migrations.

Industry 4.0

A term that was introduced by Klaus Schwab in 2016 to refer to the modernization of manufacturing processes through advances in connectivity, real-time data, automation, analytics, and AI/ML.

infrastructure

All of the resources and assets contained within an application's environment.

infrastructure as code (IaC)

The process of provisioning and managing an application's infrastructure through a set of configuration files. IaC is designed to help you centralize infrastructure management, standardize resources, and scale quickly so that new environments are repeatable, reliable, and consistent.

industrial Internet of Things (IIoT)

The use of internet-connected sensors and devices in the industrial sectors, such as manufacturing, energy, automotive, healthcare, life sciences, and agriculture. For more information, see Building an industrial Internet of Things (IIoT) digital transformation strategy.

inspection VPC

In an AWS multi-account architecture, a centralized VPC that manages inspections of network traffic between VPCs (in the same or different AWS Regions), the internet, and on-premises networks. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

Internet of Things (IoT)

The network of connected physical objects with embedded sensors or processors that communicate with other devices and systems through the internet or over a local communication network. For more information, see What is IoT?

interpretability

A characteristic of a machine learning model that describes the degree to which a human can understand how the model's predictions depend on its inputs. For more information, see Machine learning model interpretability with AWS.

IoT

See Internet of Things.

IT information library (ITIL)

A set of best practices for delivering IT services and aligning these services with business requirements. ITIL provides the foundation for ITSM.

IT service management (ITSM)

Activities associated with designing, implementing, managing, and supporting IT services for an organization. For information about integrating cloud operations with ITSM tools, see the operations integration guide.

ITIL

See IT information library.

ITSM

See IT service management.

L

label-based access control (LBAC)

An implementation of mandatory access control (MAC) where the users and the data itself are each explicitly assigned a security label value. The intersection between the user security label and data security label determines which rows and columns can be seen by the user.

landing zone

A landing zone is a well-architected, multi-account AWS environment that is scalable and secure. This is a starting point from which your organizations can quickly launch and deploy workloads and applications with confidence in their security and infrastructure environment. For more information about landing zones, see Setting up a secure and scalable multi-account AWS environment.

large language model (LLM)

A deep learning AI model that is pretrained on a vast amount of data. An LLM can perform multiple tasks, such as answering questions, summarizing documents, translating text into other languages, and completing sentences. For more information, see What are LLMs.

large migration

A migration of 300 or more servers.

LBAC

See label-based access control.

least privilege

The security best practice of granting the minimum permissions required to perform a task. For more information, see Apply least-privilege permissions in the IAM documentation.

lift and shift

See 7 Rs.

little-endian system

A system that stores the least significant byte first. See also endianness.

LLM

See large language model.

lower environments

See environment.

M

machine learning (ML)

A type of artificial intelligence that uses algorithms and techniques for pattern recognition and learning. ML analyzes and learns from recorded data, such as Internet of Things (IoT) data, to generate a statistical model based on patterns. For more information, see Machine Learning.

main branch

See branch.

malware

Software that is designed to compromise computer security or privacy. Malware might disrupt computer systems, leak sensitive information, or gain unauthorized access. Examples of malware include viruses, worms, ransomware, Trojan horses, spyware, and keyloggers.

managed services

AWS services for which AWS operates the infrastructure layer, the operating system, and platforms, and you access the endpoints to store and retrieve data. Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB are examples of managed services. These are also known as *abstracted services*.

manufacturing execution system (MES)

A software system for tracking, monitoring, documenting, and controlling production processes that convert raw materials to finished products on the shop floor.

MAP

See Migration Acceleration Program.

mechanism

A complete process in which you create a tool, drive adoption of the tool, and then inspect the results in order to make adjustments. A mechanism is a cycle that reinforces and improves itself as it operates. For more information, see Building mechanisms in the AWS Well-Architected Framework.

member account

All AWS accounts other than the management account that are part of an organization in AWS Organizations. An account can be a member of only one organization at a time.

MES

See manufacturing execution system.

Message Queuing Telemetry Transport (MQTT)

A lightweight, machine-to-machine (M2M) communication protocol, based on the publish/subscribe pattern, for resource-constrained IoT devices.

microservice

A small, independent service that communicates over well-defined APIs and is typically owned by small, self-contained teams. For example, an insurance system might include microservices that map to business capabilities, such as sales or marketing, or subdomains, such as purchasing, claims, or analytics. The benefits of microservices include agility, flexible scaling, easy deployment, reusable code, and resilience. For more information, see Integrating microservices by using AWS serverless services.

microservices architecture

An approach to building an application with independent components that run each application process as a microservice. These microservices communicate through a well-defined interface by using lightweight APIs. Each microservice in this architecture can be updated, deployed,

and scaled to meet demand for specific functions of an application. For more information, see Implementing microservices on AWS.

Migration Acceleration Program (MAP)

An AWS program that provides consulting support, training, and services to help organizations build a strong operational foundation for moving to the cloud, and to help offset the initial cost of migrations. MAP includes a migration methodology for executing legacy migrations in a methodical way and a set of tools to automate and accelerate common migration scenarios.

migration at scale

The process of moving the majority of the application portfolio to the cloud in waves, with more applications moved at a faster rate in each wave. This phase uses the best practices and lessons learned from the earlier phases to implement a *migration factory* of teams, tools, and processes to streamline the migration of workloads through automation and agile delivery. This is the third phase of the AWS migration strategy.

migration factory

Cross-functional teams that streamline the migration of workloads through automated, agile approaches. Migration factory teams typically include operations, business analysts and owners, migration engineers, developers, and DevOps professionals working in sprints. Between 20 and 50 percent of an enterprise application portfolio consists of repeated patterns that can be optimized by a factory approach. For more information, see the discussion of migration factories and the Cloud Migration Factory guide in this content set.

migration metadata

The information about the application and server that is needed to complete the migration. Each migration pattern requires a different set of migration metadata. Examples of migration metadata include the target subnet, security group, and AWS account.

migration pattern

A repeatable migration task that details the migration strategy, the migration destination, and the migration application or service used. Example: Rehost migration to Amazon EC2 with AWS Application Migration Service.

Migration Portfolio Assessment (MPA)

An online tool that provides information for validating the business case for migrating to the AWS Cloud. MPA provides detailed portfolio assessment (server right-sizing, pricing, TCO

comparisons, migration cost analysis) as well as migration planning (application data analysis and data collection, application grouping, migration prioritization, and wave planning). The MPA tool (requires login) is available free of charge to all AWS consultants and APN Partner consultants.

Migration Readiness Assessment (MRA)

The process of gaining insights about an organization's cloud readiness status, identifying strengths and weaknesses, and building an action plan to close identified gaps, using the AWS CAF. For more information, see the migration readiness guide. MRA is the first phase of the AWS migration strategy.

migration strategy

The approach used to migrate a workload to the AWS Cloud. For more information, see the 7 Rs entry in this glossary and see Mobilize your organization to accelerate large-scale migrations.

ML

See machine learning.

modernization

Transforming an outdated (legacy or monolithic) application and its infrastructure into an agile, elastic, and highly available system in the cloud to reduce costs, gain efficiencies, and take advantage of innovations. For more information, see Strategy for modernizing applications in the AWS Cloud.

modernization readiness assessment

An evaluation that helps determine the modernization readiness of an organization's applications; identifies benefits, risks, and dependencies; and determines how well the organization can support the future state of those applications. The outcome of the assessment is a blueprint of the target architecture, a roadmap that details development phases and milestones for the modernization process, and an action plan for addressing identified gaps. For more information, see Evaluating modernization readiness for applications in the AWS Cloud.

monolithic applications (monoliths)

Applications that run as a single service with tightly coupled processes. Monolithic applications have several drawbacks. If one application feature experiences a spike in demand, the entire architecture must be scaled. Adding or improving a monolithic application's features also becomes more complex when the code base grows. To address these issues, you can

use a microservices architecture. For more information, see Decomposing monoliths into microservices.

MPA

See Migration Portfolio Assessment.

MQTT

See Message Queuing Telemetry Transport.

multiclass classification

A process that helps generate predictions for multiple classes (predicting one of more than two outcomes). For example, an ML model might ask "Is this product a book, car, or phone?" or "Which product category is most interesting to this customer?"

mutable infrastructure

A model that updates and modifies the existing infrastructure for production workloads. For improved consistency, reliability, and predictability, the AWS Well-Architected Framework recommends the use of immutable infrastructure as a best practice.

O

OAC

See origin access control.

OAI

See origin access identity.

OCM

See organizational change management.

offline migration

A migration method in which the source workload is taken down during the migration process. This method involves extended downtime and is typically used for small, non-critical workloads.

OI

See operations integration.

OLA

See operational-level agreement.

online migration

A migration method in which the source workload is copied to the target system without being taken offline. Applications that are connected to the workload can continue to function during the migration. This method involves zero to minimal downtime and is typically used for critical production workloads.

OPC-UA

See Open Process Communications - Unified Architecture.

Open Process Communications - Unified Architecture (OPC-UA)

A machine-to-machine (M2M) communication protocol for industrial automation. OPC-UA provides an interoperability standard with data encryption, authentication, and authorization schemes.

operational-level agreement (OLA)

An agreement that clarifies what functional IT groups promise to deliver to each other, to support a service-level agreement (SLA).

operational readiness review (ORR)

A checklist of questions and associated best practices that help you understand, evaluate, prevent, or reduce the scope of incidents and possible failures. For more information, see Operational Readiness Reviews (ORR) in the AWS Well-Architected Framework.

operational technology (OT)

Hardware and software systems that work with the physical environment to control industrial operations, equipment, and infrastructure. In manufacturing, the integration of OT and information technology (IT) systems is a key focus for Industry 4.0 transformations.

operations integration (OI)

The process of modernizing operations in the cloud, which involves readiness planning, automation, and integration. For more information, see the operations integration guide.

organization trail

A trail that's created by AWS CloudTrail that logs all events for all AWS accounts in an organization in AWS Organizations. This trail is created in each AWS account that's part of the

organization and tracks the activity in each account. For more information, see Creating a trail for an organization in the CloudTrail documentation.

organizational change management (OCM)

A framework for managing major, disruptive business transformations from a people, culture, and leadership perspective. OCM helps organizations prepare for, and transition to, new systems and strategies by accelerating change adoption, addressing transitional issues, and driving cultural and organizational changes. In the AWS migration strategy, this framework is called *people acceleration*, because of the speed of change required in cloud adoption projects. For more information, see the OCM guide.

origin access control (OAC)

In CloudFront, an enhanced option for restricting access to secure your Amazon Simple Storage Service (Amazon S3) content. OAC supports all S3 buckets in all AWS Regions, server-side encryption with AWS KMS (SSE-KMS), and dynamic PUT and DELETE requests to the S3 bucket.

origin access identity (OAI)

In CloudFront, an option for restricting access to secure your Amazon S3 content. When you use OAI, CloudFront creates a principal that Amazon S3 can authenticate with. Authenticated principals can access content in an S3 bucket only through a specific CloudFront distribution. See also OAC, which provides more granular and enhanced access control.

ORR

See operational readiness review.

OT

See operational technology.

outbound (egress) VPC

In an AWS multi-account architecture, a VPC that handles network connections that are initiated from within an application. The AWS Security Reference Architecture recommends setting up your Network account with inbound, outbound, and inspection VPCs to protect the two-way interface between your application and the broader internet.

P

permissions boundary

An IAM management policy that is attached to IAM principals to set the maximum permissions that the user or role can have. For more information, see Permissions boundaries in the IAM documentation.

personally identifiable information (PII)

Information that, when viewed directly or paired with other related data, can be used to reasonably infer the identity of an individual. Examples of PII include names, addresses, and contact information.

PII

See personally identifiable information.

playbook

A set of predefined steps that capture the work associated with migrations, such as delivering core operations functions in the cloud. A playbook can take the form of scripts, automated runbooks, or a summary of processes or steps required to operate your modernized environment.

PLC

See programmable logic controller.

PLM

See product lifecycle management.

policy

An object that can define permissions (see identity-based policy), specify access conditions (see resource-based policy), or define the maximum permissions for all accounts in an organization in AWS Organizations (see service control policy).

polyglot persistence

Independently choosing a microservice's data storage technology based on data access patterns and other requirements. If your microservices have the same data storage technology, they can encounter implementation challenges or experience poor performance. Microservices are more easily implemented and achieve better performance and scalability if they use the data store

best adapted to their requirements. For more information, see Enabling data persistence in microservices.

portfolio assessment

A process of discovering, analyzing, and prioritizing the application portfolio in order to plan the migration. For more information, see Evaluating migration readiness.

predicate

A query condition that returns `true` or `false`, commonly located in a `WHERE` clause.

predicate pushdown

A database query optimization technique that filters the data in the query before transfer. This reduces the amount of data that must be retrieved and processed from the relational database, and it improves query performance.

preventative control

A security control that is designed to prevent an event from occurring. These controls are a first line of defense to help prevent unauthorized access or unwanted changes to your network. For more information, see Preventative controls in *Implementing security controls on AWS*.

principal

An entity in AWS that can perform actions and access resources. This entity is typically a root user for an AWS account, an IAM role, or a user. For more information, see *Principal* in Roles terms and concepts in the IAM documentation.

privacy by design

A system engineering approach that takes privacy into account through the whole development process.

private hosted zones

A container that holds information about how you want Amazon Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs. For more information, see Working with private hosted zones in the Route 53 documentation.

proactive control

A security control designed to prevent the deployment of noncompliant resources. These controls scan resources before they are provisioned. If the resource is not compliant with the control, then it isn't provisioned. For more information, see the Controls reference guide in the

AWS Control Tower documentation and see Proactive controls in *Implementing security controls on AWS*.

product lifecycle management (PLM)

The management of data and processes for a product throughout its entire lifecycle, from design, development, and launch, through growth and maturity, to decline and removal.

production environment

See environment.

programmable logic controller (PLC)

In manufacturing, a highly reliable, adaptable computer that monitors machines and automates manufacturing processes.

prompt chaining

Using the output of one LLM prompt as the input for the next prompt to generate better responses. This technique is used to break down a complex task into subtasks, or to iteratively refine or expand a preliminary response. It helps improve the accuracy and relevance of a model's responses and allows for more granular, personalized results.

pseudonymization

The process of replacing personal identifiers in a dataset with placeholder values. Pseudonymization can help protect personal privacy. Pseudonymized data is still considered to be personal data.

publish/subscribe (pub/sub)

A pattern that enables asynchronous communications among microservices to improve scalability and responsiveness. For example, in a microservices-based MES, a microservice can publish event messages to a channel that other microservices can subscribe to. The system can add new microservices without changing the publishing service.

Q

query plan

A series of steps, like instructions, that are used to access the data in a SQL relational database system.

query plan regression

When a database service optimizer chooses a less optimal plan than it did before a given change to the database environment. This can be caused by changes to statistics, constraints, environment settings, query parameter bindings, and updates to the database engine.

R

RACI matrix

See responsible, accountable, consulted, informed (RACI).

RAG

See Retrieval Augmented Generation.

ransomware

A malicious software that is designed to block access to a computer system or data until a payment is made.

RASCI matrix

See responsible, accountable, consulted, informed (RACI).

RCAC

See row and column access control.

read replica

A copy of a database that's used for read-only purposes. You can route queries to the read replica to reduce the load on your primary database.

re-architect

See 7 Rs.

recovery point objective (RPO)

The maximum acceptable amount of time since the last data recovery point. This determines what is considered an acceptable loss of data between the last recovery point and the interruption of service.

recovery time objective (RTO)

The maximum acceptable delay between the interruption of service and restoration of service.

refactor

See 7 Rs.

Region

A collection of AWS resources in a geographic area. Each AWS Region is isolated and independent of the others to provide fault tolerance, stability, and resilience. For more information, see Specify which AWS Regions your account can use.

regression

An ML technique that predicts a numeric value. For example, to solve the problem of "What price will this house sell for?" an ML model could use a linear regression model to predict a house's sale price based on known facts about the house (for example, the square footage).

rehost

See 7 Rs.

release

In a deployment process, the act of promoting changes to a production environment.

relocate

See 7 Rs.

replatform

See 7 Rs.

repurchase

See 7 Rs.

resiliency

An application's ability to resist or recover from disruptions. High availability and disaster recovery are common considerations when planning for resiliency in the AWS Cloud. For more information, see AWS Cloud Resilience.

resource-based policy

A policy attached to a resource, such as an Amazon S3 bucket, an endpoint, or an encryption key. This type of policy specifies which principals are allowed access, supported actions, and any other conditions that must be met.

responsible, accountable, consulted, informed (RACI) matrix

A matrix that defines the roles and responsibilities for all parties involved in migration activities and cloud operations. The matrix name is derived from the responsibility types defined in the matrix: responsible (R), accountable (A), consulted (C), and informed (I). The support (S) type is optional. If you include support, the matrix is called a *RASCI matrix*, and if you exclude it, it's called a *RACI matrix*.

responsive control

A security control that is designed to drive remediation of adverse events or deviations from your security baseline. For more information, see Responsive controls in *Implementing security controls on AWS*.

retain

See 7 Rs.

retire

See 7 Rs.

Retrieval Augmented Generation (RAG)

A generative AI technology in which an LLM references an authoritative data source that is outside of its training data sources before generating a response. For example, a RAG model might perform a semantic search of an organization's knowledge base or custom data. For more information, see What is RAG.

rotation

The process of periodically updating a secret to make it more difficult for an attacker to access the credentials.

row and column access control (RCAC)

The use of basic, flexible SQL expressions that have defined access rules. RCAC consists of row permissions and column masks.

RPO

See recovery point objective.

RTO

See recovery time objective.

runbook

A set of manual or automated procedures required to perform a specific task. These are typically built to streamline repetitive operations or procedures with high error rates.

S

SAML 2.0

An open standard that many identity providers (IdPs) use. This feature enables federated single sign-on (SSO), so users can log into the AWS Management Console or call the AWS API operations without you having to create user in IAM for everyone in your organization. For more information about SAML 2.0-based federation, see About SAML 2.0-based federation in the IAM documentation.

SCADA

See supervisory control and data acquisition.

SCP

See service control policy.

secret

In AWS Secrets Manager, confidential or restricted information, such as a password or user credentials, that you store in encrypted form. It consists of the secret value and its metadata. The secret value can be binary, a single string, or multiple strings. For more information, see What's in a Secrets Manager secret? in the Secrets Manager documentation.

security by design

A system engineering approach that takes security into account through the whole development process.

security control

A technical or administrative guardrail that prevents, detects, or reduces the ability of a threat actor to exploit a security vulnerability. There are four primary types of security controls: preventative, detective, responsive, and proactive.

security hardening

The process of reducing the attack surface to make it more resistant to attacks. This can include actions such as removing resources that are no longer needed, implementing the security best practice of granting least privilege, or deactivating unnecessary features in configuration files.

security information and event management (SIEM) system

Tools and services that combine security information management (SIM) and security event management (SEM) systems. A SIEM system collects, monitors, and analyzes data from servers, networks, devices, and other sources to detect threats and security breaches, and to generate alerts.

security response automation

A predefined and programmed action that is designed to automatically respond to or remediate a security event. These automations serve as detective or responsive security controls that help you implement AWS security best practices. Examples of automated response actions include modifying a VPC security group, patching an Amazon EC2 instance, or rotating credentials.

server-side encryption

Encryption of data at its destination, by the AWS service that receives it.

service control policy (SCP)

A policy that provides centralized control over permissions for all accounts in an organization in AWS Organizations. SCPs define guardrails or set limits on actions that an administrator can delegate to users or roles. You can use SCPs as allow lists or deny lists, to specify which services or actions are permitted or prohibited. For more information, see Service control policies in the AWS Organizations documentation.

service endpoint

The URL of the entry point for an AWS service. You can use the endpoint to connect programmatically to the target service. For more information, see AWS service endpoints in *AWS General Reference*.

service-level agreement (SLA)

An agreement that clarifies what an IT team promises to deliver to their customers, such as service uptime and performance.

service-level indicator (SLI)

A measurement of a performance aspect of a service, such as its error rate, availability, or throughput.

service-level objective (SLO)

A target metric that represents the health of a service, as measured by a service-level indicator.

shared responsibility model

A model describing the responsibility you share with AWS for cloud security and compliance. AWS is responsible for security *of* the cloud, whereas you are responsible for security *in* the cloud. For more information, see Shared responsibility model.

SIEM

See security information and event management system.

single point of failure (SPOF)

A failure in a single, critical component of an application that can disrupt the system.

SLA

See service-level agreement.

SLI

See service-level indicator.

SLO

See service-level objective.

split-and-seed model

A pattern for scaling and accelerating modernization projects. As new features and product releases are defined, the core team splits up to create new product teams. This helps scale your organization's capabilities and services, improves developer productivity, and supports rapid

innovation. For more information, see Phased approach to modernizing applications in the AWS Cloud.

SPOF

See single point of failure.

star schema

A database organizational structure that uses one large fact table to store transactional or measured data and uses one or more smaller dimensional tables to store data attributes. This structure is designed for use in a data warehouse or for business intelligence purposes.

strangler fig pattern

An approach to modernizing monolithic systems by incrementally rewriting and replacing system functionality until the legacy system can be decommissioned. This pattern uses the analogy of a fig vine that grows into an established tree and eventually overcomes and replaces its host. The pattern was introduced by Martin Fowler as a way to manage risk when rewriting monolithic systems. For an example of how to apply this pattern, see Modernizing legacy Microsoft ASP.NET (ASMX) web services incrementally by using containers and Amazon API Gateway.

subnet

A range of IP addresses in your VPC. A subnet must reside in a single Availability Zone.

supervisory control and data acquisition (SCADA)

In manufacturing, a system that uses hardware and software to monitor physical assets and production operations.

symmetric encryption

An encryption algorithm that uses the same key to encrypt and decrypt the data.

synthetic testing

Testing a system in a way that simulates user interactions to detect potential issues or to monitor performance. You can use Amazon CloudWatch Synthetics to create these tests.

system prompt

A technique for providing context, instructions, or guidelines to an LLM to direct its behavior. System prompts help set context and establish rules for interactions with users.

T

tags

Key-value pairs that act as metadata for organizing your AWS resources. Tags can help you manage, identify, organize, search for, and filter resources. For more information, see Tagging your AWS resources.

target variable

The value that you are trying to predict in supervised ML. This is also referred to as an *outcome variable*. For example, in a manufacturing setting the target variable could be a product defect.

task list

A tool that is used to track progress through a runbook. A task list contains an overview of the runbook and a list of general tasks to be completed. For each general task, it includes the estimated amount of time required, the owner, and the progress.

test environment

See environment.

training

To provide data for your ML model to learn from. The training data must contain the correct answer. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict). It outputs an ML model that captures these patterns. You can then use the ML model to make predictions on new data for which you don't know the target.

transit gateway

A network transit hub that you can use to interconnect your VPCs and on-premises networks. For more information, see What is a transit gateway in the AWS Transit Gateway documentation.

trunk-based workflow

An approach in which developers build and test features locally in a feature branch and then merge those changes into the main branch. The main branch is then built to the development, preproduction, and production environments, sequentially.

trusted access

Granting permissions to a service that you specify to perform tasks in your organization in AWS Organizations and in its accounts on your behalf. The trusted service creates a service-linked role in each account, when that role is needed, to perform management tasks for you. For more information, see Using AWS Organizations with other AWS services in the AWS Organizations documentation.

tuning

To change aspects of your training process to improve the ML model's accuracy. For example, you can train the ML model by generating a labeling set, adding labels, and then repeating these steps several times under different settings to optimize the model.

two-pizza team

A small DevOps team that you can feed with two pizzas. A two-pizza team size ensures the best possible opportunity for collaboration in software development.

U

uncertainty

A concept that refers to imprecise, incomplete, or unknown information that can undermine the reliability of predictive ML models. There are two types of uncertainty: *Epistemic uncertainty* is caused by limited, incomplete data, whereas *aleatoric uncertainty* is caused by the noise and randomness inherent in the data. For more information, see the Quantifying uncertainty in deep learning systems guide.

undifferentiated tasks

Also known as *heavy lifting*, work that is necessary to create and operate an application but that doesn't provide direct value to the end user or provide competitive advantage. Examples of undifferentiated tasks include procurement, maintenance, and capacity planning.

upper environments

See environment.

V

vacuuming

A database maintenance operation that involves cleaning up after incremental updates to reclaim storage and improve performance.

version control

Processes and tools that track changes, such as changes to source code in a repository.

VPC peering

A connection between two VPCs that allows you to route traffic by using private IP addresses. For more information, see What is VPC peering in the Amazon VPC documentation.

vulnerability

A software or hardware flaw that compromises the security of the system.

W

warm cache

A buffer cache that contains current, relevant data that is frequently accessed. The database instance can read from the buffer cache, which is faster than reading from the main memory or disk.

warm data

Data that is infrequently accessed. When querying this kind of data, moderately slow queries are typically acceptable.

window function

A SQL function that performs a calculation on a group of rows that relate in some way to the current record. Window functions are useful for processing tasks, such as calculating a moving average or accessing the value of rows based on the relative position of the current row.

workload

A collection of resources and code that delivers business value, such as a customer-facing application or backend process.

workstream

Functional groups in a migration project that are responsible for a specific set of tasks. Each workstream is independent but supports the other workstreams in the project. For example, the portfolio workstream is responsible for prioritizing applications, wave planning, and collecting migration metadata. The portfolio workstream delivers these assets to the migration workstream, which then migrates the servers and applications.

WORM

See write once, read many.

WQF

See AWS Workload Qualification Framework.

write once, read many (WORM)

A storage model that writes data a single time and prevents the data from being deleted or modified. Authorized users can read the data as many times as needed, but they cannot change it. This data storage infrastructure is considered immutable.

Z

zero-day exploit

An attack, typically malware, that takes advantage of a zero-day vulnerability.

zero-day vulnerability

An unmitigated flaw or vulnerability in a production system. Threat actors can use this type of vulnerability to attack the system. Developers frequently become aware of the vulnerability as a result of the attack.

zero-shot prompting

Providing an LLM with instructions for performing a task but no examples (*shots*) that can help guide it. The LLM must use its pre-trained knowledge to handle the task. The effectiveness of zero-shot prompting depends on the complexity of the task and the quality of the prompt. See also few-shot prompting.

zombie application

An application that has an average CPU and memory usage below 5 percent. In a migration project, it is common to retire these applications.