AWS Regions

AWS regions refer to the geographical locations where Amazon Web Services (AWS) has established data centers. Each AWS region is a separate geographic area, isolated from other regions for fault tolerance, compliance, and low latency. AWS currently has 25 regions globally, which allows customers to choose the region that best suits their requirements.

Each AWS region is identified by a code, such as us-east-1 for the US East (N. Virginia) region or eu-west-1 for the EU (Ireland) region. These codes are used to specify the location of your AWS resources.

Availability Zones:

Within each AWS region, there are multiple Availability Zones (AZs). Availability Zones are essentially data centers with redundant power, cooling, and networking infrastructure. They are physically separate from one another within a single region but are connected through low-latency links.

The purpose of Availability Zones is to provide fault tolerance and high availability. By deploying your applications across multiple Availability Zones, you can better protect your workload from common types of failures, such as power outages, network failures, or disasters affecting a single zone.

Each Availability Zone in a region typically consists of one or more data centers. Deploying resources across different Availability Zones ensures that if one zone goes down or experiences service disruptions, your applications and data remain available in other zones.

It's important to note that each Availability Zone within a region is designed to be independent, fault-tolerant, and has its own infrastructure. This allows you to build highly reliable and scalable applications by distributing your resources across multiple Availability Zones.

AWS Edge Locations:

AWS Edge Locations are a part of the AWS global infrastructure and play a vital role in content delivery and edge computing. Unlike AWS regions and availability zones, which are data centers in specific geographic locations, Edge Locations are distributed points of presence (PoPs) in different locations worldwide.

Edge Locations are deployed in major cities and metropolitan areas, bringing AWS services closer to end-users and reducing latency for delivering content and applications. They act as caching endpoints for content delivery and improve the performance of AWS services like Amazon CloudFront, Amazon Route 53, and AWS Global Accelerator.

These Edge Locations store cached copies of data and content closer to end-users, reducing the time it takes to retrieve data and respond to user requests. This helps in delivering a better user experience, especially for content-heavy applications, media streaming, and dynamic websites.

It's important to note that Edge Locations do not store the full range of AWS services and resources like regions do. Instead, they primarily focus on content delivery and caching. The global network of Edge Locations is constantly growing to expand the reach and improve performance for AWS services.

EC2 Key Pairs

An EC2 key pair is a security credential provided by AWS that allows you to securely connect to your EC2 (Elastic Compute Cloud) instances. Here's how an EC2 key pair works:

1. Key Generation: When you create an EC2 key pair, AWS generates a pair of cryptographic keys: a public key and a private key. The key pair uses asymmetric encryption, where the public key is used to encrypt data and the private key is used to decrypt it.
2. Public Key Assignment: The public key of the key pair is associated with the EC2 instance during instance creation. When the instance is started, the public key is placed on the instance within the operating system. This key is used to encrypt login credentials and establish a secure connection.
3. Private Key Management: The private key of the key pair is downloaded and saved securely on your local machine. It should never be shared or exposed to others. The private key is required to authenticate and establish a secure connection to your EC2 instance.
4. Secure Connection: To connect to your EC2 instance, you use SSH (Secure Shell) or RDP (Remote Desktop Protocol) depending on the operating system of the instance. When you initiate the connection with the private key, AWS verifies your identity by decrypting the login credentials using the corresponding public key on the instance.
5. Access Control: By controlling access to the private key, you effectively control who can connect to your EC2 instance. It is vital to keep the private key secure and protected from unauthorized access.

EC2 key pairs provide a secure way to access your EC2 instances and ensure that only authorized users can connect to them. It is essential to handle the private key with care and follow security best practices when managing key pairs.

EC2 Instance Types

EC2 (Elastic Compute Cloud) offers a wide range of instance types to cater to different computing needs. Each instance type is optimized for specific workloads and offers varying combinations of CPU, memory, storage, and network resources. Here is an overview of EC2 instance types:

1. General Purpose Instances:

- Examples: T4g, T3, T3a
- Balanced compute, memory, and network resources.
- Suitable for a wide range of workloads, including web servers, small databases, and development environments.

2. Compute-Optimized Instances:

- Examples: C6g, C5, C5a
- Excellent performance for compute-intensive applications.
- Ideal for high-performance computing (HPC), scientific modeling, batch processing, gaming, and video encoding.

3. Memory-Optimized Instances:

- Examples: R6g, R5, R5a
- Designed for memory-intensive workloads and applications that require high memory per vCPU.
- Well-suited for data processing, real-time analytics, and in-memory databases.

4. GPU Instances:

- Examples: P4, G4, G3
- Equipped with powerful GPUs (Graphics Processing Units).
- Ideal for workloads like deep learning, machine learning, and graphics-intensive applications.

5. Storage-Optimized Instances:

- Examples: I3, D3
- High storage density and fast storage performance.
- Optimized for applications that require high I/O performance or large local storage capacity.

6. Burstable Performance Instances:

- Examples: T3, T3a, T2
- Provide a baseline performance level with the ability to burst to higher levels when needed.
- Suitable for applications with variable workloads and periodic bursts of traffic.

EC2 Security Groups

EC2 Security Groups act as virtual firewalls that control inbound and outbound traffic for EC2 instances. They allow you to define fine-grained access rules to specify which network traffic is allowed or denied. Here is some information about EC2 Security Groups:

1. Basic Functionality:

- Each EC2 instance is associated with one or more security groups.
- Security groups act at the instance level, controlling inbound and outbound traffic for the specific instance.
- You can define rules to allow or deny traffic based on protocols, ports, and IP ranges.
- Security group rules can be added, modified, or removed at any time.

2. Inbound and Outbound Rules:

- Inbound Rules: Control incoming traffic to the instance. You can specify the source IP or range, the protocol (such as SSH, HTTP, or RDP), and the destination port.
- Outbound Rules: Control outgoing traffic from the instance. By default, all outbound traffic is allowed but can be restricted as per the desired requirements.

3. Port Configuration:

- Security groups use rules based on protocols and ports. For example, you can open port 80 for HTTP traffic or port 22 for SSH access.
- You can configure both TCP and UDP protocols, as well as specify custom port ranges.

4. Security Group Types:

- Default Security Group: Every VPC (Virtual Private Cloud) has a default security group that allows all outbound traffic and permits inbound traffic from other instances within the same security group.
- Custom Security Group: You can create custom security groups with specific rules tailored to your application or workload.

5. Security Group Behavior:

- Instances associated with the same security group can communicate with each other by default.
- The "deny by default" principle is followed, meaning that if a rule is not explicitly defined to allow traffic, it is denied by default.

6. Modifying Security Groups:

- You can modify the rules of a security group to allow or deny specific types of traffic. Changes take effect immediately.

EC2 Security Groups play a crucial role in protecting your EC2 instances and controlling network traffic. They provide an additional layer of security by allowing administrators to specify which traffic is permitted to access the instances.

EC2 Instance Tenancy

EC2 Dedicated Instances and Dedicated Hosts are two additional options for managing instance tenancy in AWS, providing even greater isolation and control over your EC2 instances.

A Dedicated Instance is a single-tenant environment where your instance runs on physical hardware that is dedicated to your AWS account. This means your instance will be the only one running on that hardware, ensuring that you have exclusive access to the underlying resources. Dedicated Instances can be launched using the EC2 Launch Wizard or the AWS Management Console, and they are charged per hour at an additional cost compared to instances launched in the default multi-tenant environment.

On the other hand, a Dedicated Host allows you to have even more control over the underlying hardware by giving you direct access to an entire physical server. With a Dedicated Host, you can launch multiple EC2 instances on the same physical server while retaining the benefits of a dedicated environment. This gives you greater control over instance placement, instance affinity, and the ability to bring your own licenses. Dedicated Hosts can be managed using the EC2 CLI, API, or the AWS Management Console, and they are charged based on the instance hours used on the host.

Both Dedicated Instances and Dedicated Hosts provide enhanced isolation and security, making them suitable for use cases with strict compliance requirements or for workloads that require complete control over underlying hardware resources. It's important to note that choosing Dedicated Instances or Dedicated Hosts will incur additional costs compared to the default multi-tenant environment. You can refer to the AWS documentation for more detailed information on how to manage instance tenancy and use Dedicated Instances and Dedicated Hosts effectively.

EC2 Instance Placement Groups

When you launch a new EC2 instance, the EC2 service attempts to place the instance in such a way that all of your instances are spread out across underlying hardware to minimize correlated failures. You can use *placement groups* to influence the placement of a group of *interdependent* instances to meet the needs of your workload. Depending on the type of workload, you can create a placement group using one of the following placement strategies:

- **Cluster** – packs instances close together inside an Availability Zone. This strategy enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communication that is typical of high-performance computing (HPC) applications.
- **Partition** – spreads your instances across logical partitions such that groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions. This strategy is typically used by large distributed and replicated workloads, such as Hadoop, Cassandra, and Kafka.
- **Spread** – strictly places a small group of instances across distinct underlying hardware to reduce correlated failures.

Cluster placement groups

A cluster placement group is a logical grouping of instances within a single Availability Zone. A cluster placement group can span peered virtual private networks (VPCs) in the same Region. Instances in the same cluster placement group enjoy a higher per-flow throughput limit for TCP/IP traffic and are placed in the same high-bisection bandwidth segment of the network.

The following image shows instances that are placed into a cluster placement group.

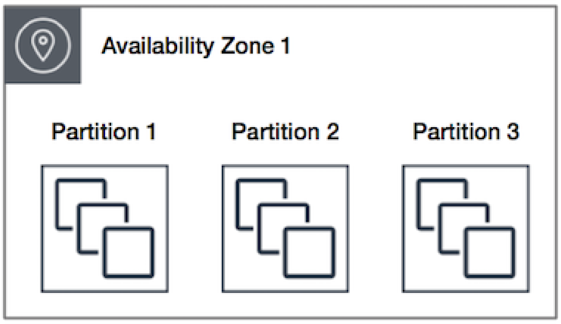
Availability Zone

Cluster placement groups are recommended for applications that benefit from low network latency, high network throughput, or both. They are also recommended when the majority of the network traffic is between the instances in the group.

Partition placement groups

Partition placement groups help reduce the likelihood of correlated hardware failures for your application. When using partition placement groups, Amazon EC2 divides each group into logical segments called partitions. Amazon EC2 ensures that each partition within a placement group has its own set of racks. Each rack has its own network and power source. No two partitions within a placement group share the same racks, allowing you to isolate the impact of hardware failure within your application.

The following image is a simple visual representation of a partition placement group in a single Availability Zone. It shows instances that are placed into a partition placement group with three partitions—Partition 1, Partition 2, and Partition 3. Each partition comprises multiple instances. The instances in a partition do not share racks with the instances in the other partitions, allowing you to contain the impact of a single hardware failure to only the associated partition.



Partition placement groups can be used to deploy large distributed and replicated workloads, such as HDFS, HBase, and Cassandra, across distinct racks. When you launch instances into a partition placement group, Amazon EC2 tries to distribute the instances evenly across the number of partitions that you specify. You can also launch instances into a specific partition to have more control over where the instances are placed.

A partition placement group can have partitions in multiple Availability Zones in the same Region. A partition placement group can have a maximum of seven partitions per Availability Zone. The number of instances that can be launched into a partition placement group is limited only by the limits of your account.

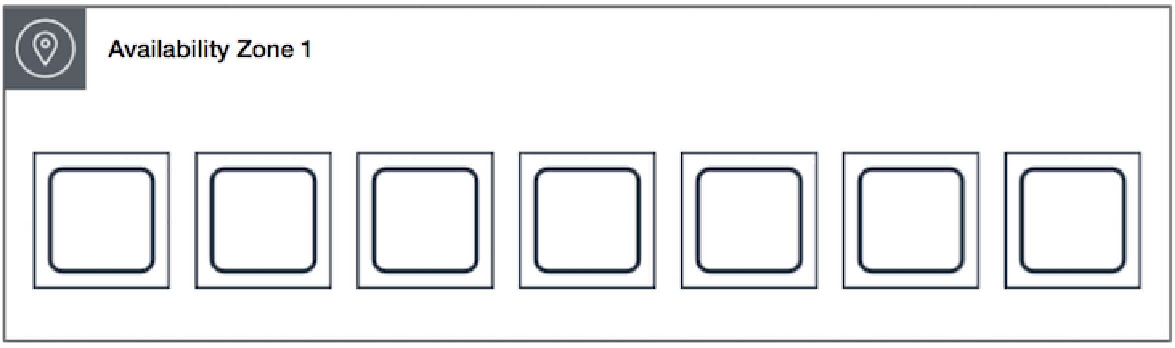
In addition, partition placement groups offer visibility into the partitions — you can see which instances are in which partitions. You can share this information with topology-aware applications, such as HDFS, HBase, and Cassandra. These applications use this information to make intelligent data replication decisions for increasing data availability and durability.

Spread placement groups

A spread placement group is a group of instances that are each placed on distinct hardware.

Spread placement groups are recommended for applications that have a small number of critical instances that should be kept separate from each other. Launching instances in a spread level placement group reduces the risk of simultaneous failures that might occur when instances share the same equipment. Spread level placement groups provide access to distinct hardware, and are therefore suitable for mixing instance types or launching instances over time.

The following image shows seven instances in a single Availability Zone that are placed into a spread placement group. The seven instances are placed on seven different racks, each rack has its own network and power source.



A spread placement group can span multiple Availability Zones in the same Region. For spread level placement groups, you can have a maximum of seven running instances per Availability Zone per group.

Spot Instances

A Spot Instance is an instance that uses spare EC2 capacity that is available for less than the On-Demand price. Because Spot Instances enable you to request unused EC2 instances at steep discounts, you can lower your Amazon EC2 costs significantly. The hourly price for a Spot Instance is called a Spot price. The Spot price of each instance type in each Availability Zone is set by Amazon EC2, and is adjusted gradually based on the long-term supply of and demand for Spot Instances. Your Spot Instance runs whenever capacity is available.

Spot Instances are a cost-effective choice if you can be flexible about when your applications run and if your applications can be interrupted. For example, Spot Instances are well-suited for data analysis, batch jobs, background processing, and optional tasks.

Key differences between Spot Instances and On-Demand Instances

	Spot Instances	**On-Demand Instances**
Launch time	Can only be launched immediately if the Spot Instance request is active and capacity is available.	Can only be launched immediately if you make a manual launch request and capacity is available.
Available capacity	If capacity is not available, the Spot Instance request continues to automatically make the launch request	If capacity is not available when you make a launch request, you get an insufficient capacity error (ICE).

	until capacity becomes available.	
Hourly price	The hourly price for Spot Instances varies based on long-term supply and demand.	The hourly price for On-Demand Instances is static.
Rebalance recommendation	The signal that Amazon EC2 emits for a running Spot Instance when the instance is at an elevated risk of interruption.	You determine when an On-Demand Instance is interrupted (stopped, hibernated, or terminated).
Instance interruption	You can stop and start an Amazon EBS-backed Spot Instance. In addition, Amazon EC2 can interrupt an individual Spot Instance if capacity is no longer available.	You determine when an On-Demand Instance is interrupted (stopped, hibernated, or terminated).

Amazon Machine Images (AMIs)

Amazon Machine Images (AMIs) are pre-configured templates that contain all the necessary information to launch instances (virtual servers) in the Amazon Web Services (AWS) environment. An AMI serves as the basic building block for creating virtual machines on AWS. Here's some key information about AMIs:

1. Definition: An AMI is a packaged system image that includes an operating system, application software, and any additional software required to run and configure the instance. It provides a blueprint for launching virtual machines in AWS.

2. Types: There are two main types of AMIs—public and private. Public AMIs are created and shared by the AWS community, while private AMIs are created and used within your own AWS account.

3. Components: AMIs consist of the following main components:

- Root volume: The root volume contains the operating system and the initial file system.

- Block device mapping: AMIs can include additional EBS volumes or instance store volumes, which are connected to the instance.

4. Usage: AMIs are used to launch instances in AWS. When you launch an instance, you can choose the desired AMI, configure the instance size, networking, and other parameters. Each instance launched from an AMI is a copy of the AMI at the time of launch.

5. Customization: AMIs can be customized by installing software, configuring settings, or making other modifications to the virtual machine. You can create your own AMIs from running instances or use existing public AMIs as a starting point.

6. Marketplace: The AWS Marketplace is a digital catalog where you can find various AMIs provided by AWS and third-party vendors. It offers a wide range of pre-configured software environments, including operating systems, databases, web servers, and more.

7. Sharing and Security: You can control the sharing and access permissions for your private AMIs. Additionally, AWS provides various security features, including encryption options for AMIs and secure access management using AWS Identity and Access Management (IAM).

AMIs are vital for deploying and scaling applications in the AWS environment. They enable you to quickly launch instances with specific software configurations, reducing the time and effort required in setting up servers from scratch.