

What is an Instance in AWS?

An **instance** in AWS refers to a **virtual server** running in the cloud. It is created using **Amazon EC2 (Elastic Compute Cloud)** and can be used to host applications, databases, or any computing workloads.

Key Features of an AWS EC2 Instance

- Scalable** – Can increase or decrease compute capacity as needed.
- Flexible Configurations** – Choose different CPU, memory, and storage options.
- Pay-as-you-go Pricing** – Pay only for what you use.
- Secure** – Control access using IAM roles, security groups, and key pairs.
- Automated Backup & Recovery** – Snapshots, AMIs, and auto-scaling ensure data protection.

What is an AMI (Amazon Machine Image)?

An **AMI (Amazon Machine Image)** is a **pre-configured template** used to launch **EC2 instances** in AWS. It contains:

- Operating System (OS)** – Linux, Windows, Ubuntu, RHEL, etc.
- Pre-installed Software** – Web servers, databases, security tools.
- Application Code & Configurations** – Custom settings for deployments.
- EBS Snapshot** – Root volume and additional storage.

Types of AMIs in AWS

1. AWS-Provided AMIs

- Official images from AWS with common OS.
- Examples:
 - Amazon Linux AMI
 - Ubuntu, RHEL, Windows Server

2. Custom AMIs

- Created by users for specific applications.
- Useful for pre-installed software & security configurations.

3. AWS Marketplace AMIs

- Third-party AMIs with specialized software.
- Examples:
 - WordPress AMI
 - AI/ML-optimized AMI (NVIDIA, TensorFlow)

4. Shared & Community AMIs

- Public AMIs shared by AWS users.
- Can be free or paid.

What is an Instance Type in AWS?

An **Instance Type** in AWS defines the **compute power** of an **EC2 instance**, including:

- CPU (vCPUs)** – Processing power
- RAM (Memory)** – Available memory
- Storage** – SSD, HDD, or EBS-backed volumes
- Network Performance** – Bandwidth capacity

Categories of AWS EC2 Instance Types

AWS offers different instance families optimized for various workloads:

1. General Purpose

- ◆ Balanced CPU, memory, and storage.
- ◆ **Examples:** t3.micro, m5.large
- ◆ **Use Case:** Web servers, small applications, development & testing.

2. Compute Optimized

- ◆ More powerful CPUs for high-performance computing.
- ◆ **Examples:** c5.large, c6g.xlarge
- ◆ **Use Case:** Gaming, data processing, machine learning.

3. Memory Optimized

- ◆ High RAM for memory-intensive applications.
- ◆ **Examples:** r5.large, x1e.xlarge
- ◆ **Use Case:** Large databases, in-memory caching, analytics.

4. Storage Optimized

- ◆ High-speed, low-latency storage for large datasets.
- ◆ **Examples:** i3.large, d2.xlarge
- ◆ **Use Case:** Big Data, high IOPS workloads, NoSQL databases.

5. GPU (Graphics Processing Unit) Instances

- ◆ Designed for AI, ML, deep learning, and rendering.
- ◆ **Examples:** p4d.24xlarge, g5.xlarge
- ◆ **Use Case:** Video processing, 3D modeling, AI training.

AWS EC2 Instance Naming Convention

Each instance type follows this format:

 **Example:** t3.micro

- t → Instance family (General Purpose)
- 3 → Generation (newer = better performance)
- micro → Size (CPU & RAM capacity)

Comparison of Instance Types

Instance Type	vCPU	Memory (RAM)	Storage	Use Case
t3.micro	2	1 GB	EBS	Small websites, test environments
m5.large	2	8 GB	EBS	Web apps, databases
c5.large	2	4 GB	EBS	High-performance computing
r5.large	2	16 GB	EBS	In-memory databases
i3.large	2	15.25 GB	NVMe SSD	Big Data, NoSQL DBs
p4d.24xlarge	96	1.1 TB	NVMe SSD	AI, ML, Deep Learning

What is a Key Pair in AWS?

A **Key Pair** in AWS is a combination of:

- ✓ **Public Key** – Stored in AWS and used to encrypt data.
- ✓ **Private Key** – Downloaded by the user and used to decrypt data & log in.

How Key Pairs Work?

- 1 You create a key pair in AWS.
- 2 AWS keeps the public key and embeds it into the EC2 instance.
- 3 You download the private key (.pem file) and store it securely.

What is a Volume in AWS?

A **Volume** in AWS refers to a **virtual disk** that provides **persistent storage** for EC2 instances. It acts like a hard drive and stores **data, OS files, and applications**.

- 📌 **Key Points:**
- ✓ **Persistent Storage** – Data remains even after instance termination (except for ephemeral storage).
- ✓ **Flexible Size** – Can be increased without stopping the instance.
- ✓ **Attach & Detach** – Can be attached to multiple EC2 instances.
- ✓ **Backup & Restore** – Supports snapshots for data protection.

Types of Volumes in AWS

AWS provides different types of storage volumes under **Elastic Block Store (EBS)** and **Instance Store**:

1. Elastic Block Store (EBS) – Persistent Storage

- **Block-level storage** that attaches to EC2 instances.
- **Data is retained** even if the instance stops.

What is a Snapshot in AWS?

A **Snapshot** in AWS is a **point-in-time backup** of an **EBS (Elastic Block Store) volume**. It captures the **entire volume's state** and can be used to **restore, duplicate, or create new volumes**.

📌 Key Features:

- ✓ **Incremental Backup** – Stores only changed data after the first full snapshot.
 - ✓ **Stored in S3** – Even though snapshots are not directly visible in S3, AWS stores them efficiently.
 - ✓ **Used for Disaster Recovery** – Helps restore lost or corrupted data.
 - ✓ **Cross-Region & Cross-Account Sharing** – Can be copied to another AWS region or shared with other AWS accounts.
-

How Snapshots Work? 

- 1 **Take a Snapshot** of an EBS volume.
- 2 **AWS stores a full copy** during the first snapshot.
- 3 Future snapshots **only save changed data** (incremental).
- 4 **Restore Snapshot** to create a new volume if needed.