

EC2

Elastic compute cloud

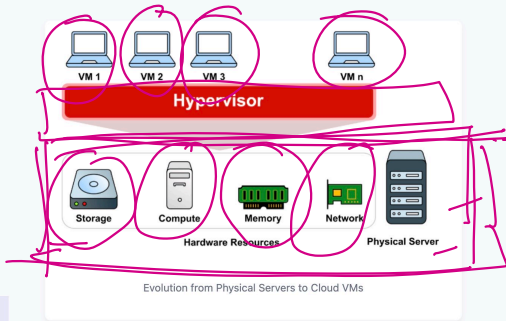
Understanding Virtual Machines in AWS

Evolution of Computing

The journey from physical servers to cloud-based virtual machines represents a fundamental shift in computing infrastructure:

- ✓ **Traditional Physical Servers:** Dedicated hardware for specific applications
- ✓ **On-premises Virtualization:** Multiple VMs on a single physical server
- ✓ **Cloud-based VMs (EC2):** Virtualized compute resources on demand
- ✓ **Containerization:** Lightweight application packaging
- ✓ **Serverless Computing:** Function-based execution without managing servers

Amazon EC2 represents the evolution of virtual machines into the cloud, offering greater flexibility, scalability, and cost efficiency compared to traditional virtualization.

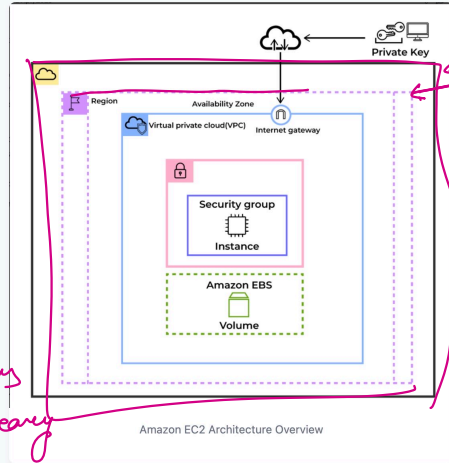


AWS Offerings of EC2

What is Amazon EC2?

Amazon Elastic Compute Cloud (EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.














- ✔ Virtual computing environments, known as instances
- ✔ Pre-configured templates for instances, called Amazon Machine Images (AMIs)
- ✔ Various configurations of CPU, memory, storage, and networking capacity
- ✔ Secure login information for instances with key pairs
- ✔ Storage volumes for temporary data, known as instance store volumes
- ✔ Persistent storage volumes using Amazon EBS
- ✔ Multiple physical locations for resources, known as Regions and Availability Zones



② → offerings ← shared tenancy
← Dedicated tenancy

US-east-1
Region

Types of EC2 Instance Types

General Purpose	Compute Optimised	Memory Optimised	Accelerated Computing	Storage Optimised
 A1 ARM based core and custom silicon	 C4 Compute - CPU intensive apps and DBs	 R4 RAM - Memory intensive apps and DB's	 P2 Processing optimised - Machine Learning	 H1 High Disk Throughput - Big data clusters
 T2 Tiny - Web servers and small DBs		 X1 Xtreme RAM - For SAP/Spark	 G3 Graphics Intensive - Video and streaming	 I3 IOPS - NoSQL DBs
 M4 Main - App servers and general purpose		 z1d High Compute and High Memory - Gaming	 F1 Field Programmable - Hardware acceleration	 D2 Dense Storage - Data Warehousing

Instance Families

Family	Use Case	Description	Example Types
General Purpose	Balanced resources	Ideal for applications that use resources in equal proportions	t3, m5, m6g
Compute Optimized	High performance computing	Ideal for compute-bound applications that benefit from high-performance processors	c5, c6g
Memory Optimized	Memory-intensive workloads	Designed for memory-intensive applications like in-memory databases	r5, r6g, x1
Storage Optimized	High storage throughput	Designed for workloads that require high, sequential read/write access to large datasets	d2, i3, h1
Accelerated Computing	Hardware accelerators	Uses hardware accelerators to perform functions like graphics processing more efficiently	p3, g4, inf1

**Instance types are regularly updated by AWS. Always check the AWS console for the latest options.*

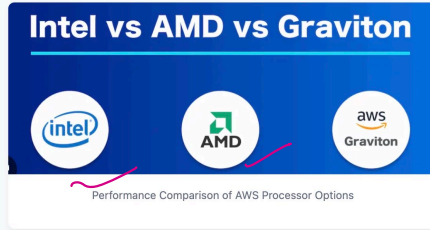
Processors of AWS

Processor Options for EC2

AWS offers multiple processor options to meet diverse workload requirements:

- ✓ **Intel Xeon:** Traditional x86 processors with proven reliability
- ✓ **AMD EPYC:** High core count and competitive price-performance ratio
- ✓ **AWS Graviton:** ARM-based processors designed by AWS for better price-performance
- ✓ **NVIDIA GPUs:** Accelerators for graphics and machine learning workloads
- ✓ **AWS Inferentia:** Custom chips designed for machine learning inference

Choosing the right processor can significantly impact both performance and cost. AWS Graviton processors can provide up to 40% better price-performance compared to x86 processors for many workloads.



Types of Purchasing Options

Flexible Pricing Models

Amazon EC2 offers several pricing models to suit different workloads and budget requirements:

- ✓ **On-Demand Instances:** Pay by the hour with no long-term commitments
- ✓ **Reserved Instances:** Up to 72% discount for 1 or 3-year commitments
- ✓ **Spot Instances:** Bid on unused EC2 capacity for up to 90% discount
- ✓ **Dedicated Hosts:** Physical servers dedicated for your use
- ✓ **Savings Plans:** Flexible pricing model with savings up to 72%

Choosing the right pricing model can significantly reduce your cloud computing costs while meeting your application's requirements.

Purchasing Method	Commitment Structure	Exchanging Options	% Savings over On-Demand
On-Demand	Charged on a \$/second basis for instances that are deployed	Not Applicable	Not Applicable
Compute Savings Plans	Commitments are made on \$/hour basis for term lengths of 1 or 3 years	Not Applicable	Up to 66%
EC2 Savings Plans	Commitments are made on \$/hour basis for term lengths of 1 or 3 years	Not Applicable	Up to 72%
Standard RI	Commitments are made to specific instance configurations for term lengths of 1 or 3 years	Can be bought and sold in the AWS RI Marketplace	Up to 72%*
Convertible RI	Commitments are made to specific instance configurations for term lengths of 1 or 3 years	Can be exchanged for other instances of same or different properties	Up to 66%*

* RIs save 5-10% over their Savings Plan counterparts

Comparison of EC2 Pricing Models

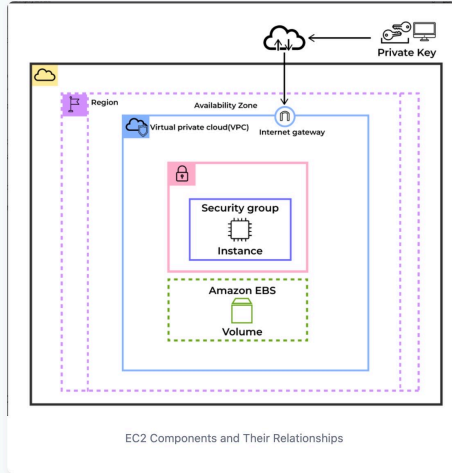
Compute Instance
Volta

Different Components of EC2

Essential EC2 Components

An EC2 instance consists of several key components that work together:

- ✓ **Amazon Machine Images (AMIs):** Templates containing the OS and application software
- ✓ **Instance Types:** Hardware configuration of your virtual server
- ✓ **Elastic Block Store (EBS):** Persistent storage volumes for your data
- ✓ **Key Pairs:** Secure login credentials for your instances
- ✓ **Security Groups:** Virtual firewalls controlling traffic to instances
- ✓ **Elastic IP Addresses:** Static IPv4 addresses for dynamic cloud computing
- ✓ **Tags:** Metadata assigned to your EC2 resources



Security Groups and NACL

Securing Your EC2 Instances

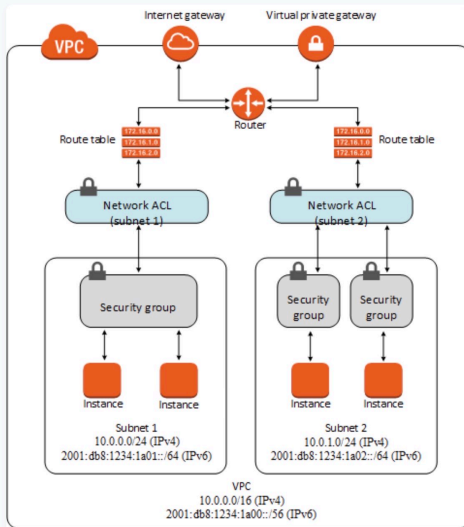
AWS provides multiple layers of security for your EC2 instances:

- ✓ **Security Groups:** Instance-level firewall that controls inbound and outbound traffic
- ✓ **Network ACLs:** Subnet-level firewall that provides an additional layer of security

Security Groups vs. Network ACLs

Feature	Security Group	Network ACL
Scope	Instance level	Subnet level
State	Stateful (return traffic automatically allowed)	Stateless (return traffic must be explicitly allowed)
Rules	Allow rules only	Allow and deny rules
Rule Processing	All rules evaluated before decision	Rules processed in number order

Best practice is to use both Security Groups and NACLs as complementary security layers for your EC2 instances.



Security Groups and NACLs in AWS VPC

Launching an EC2 Instance

Step-by-Step Guide

1

Choose an Amazon Machine Image (AMI)

Select a template that contains the operating system and applications you need.

```
aws ec2 describe-images --owners amazon --filters "Name=name,Values=amzn2-ami-hvm-*-x86_64-gp2" --query "Images[0].ImageId"
```

2

Select an Instance Type

Choose the appropriate CPU, memory, storage, and networking capacity for your workload.

```
aws ec2 describe-instance-types --filters "Name=instance-type,Values=t2.micro" --query "InstanceTypes[0]"
```

3

Configure Security Groups

Set up firewall rules to control inbound and outbound traffic to your instance.

```
aws ec2 create-security-group --group-name MySecurityGroup --description "My security group" --vpc-id vpc-1a2b3c4d
aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port 22 --cidr 203.0.113.0/24
aws ec2 authorize-security-group-ingress --group-id sg-903004f8 --protocol tcp --port 80 --cidr 0.0.0.0/0
```

4

Launch the Instance

Create and start your EC2 instance with the selected configuration.

```
aws ec2 run-instances --image-id ami-0abcdef1234567890 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
```

5

Connect to Your Instance

Use SSH (for Linux) or RDP (for Windows) to connect to your running instance.

```
ssh -i "MyKeyPair.pem" ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```