🌐 Amazon EC2 (Elastic Compute Cloud)

Amazon EC2 is one of the **core AWS services** that provides resizable, secure, and scalable virtual servers in the cloud.

Definition: Amazon EC2 is a web service that allows you to run applications on virtual servers (called *instances*) in the AWS Cloud.

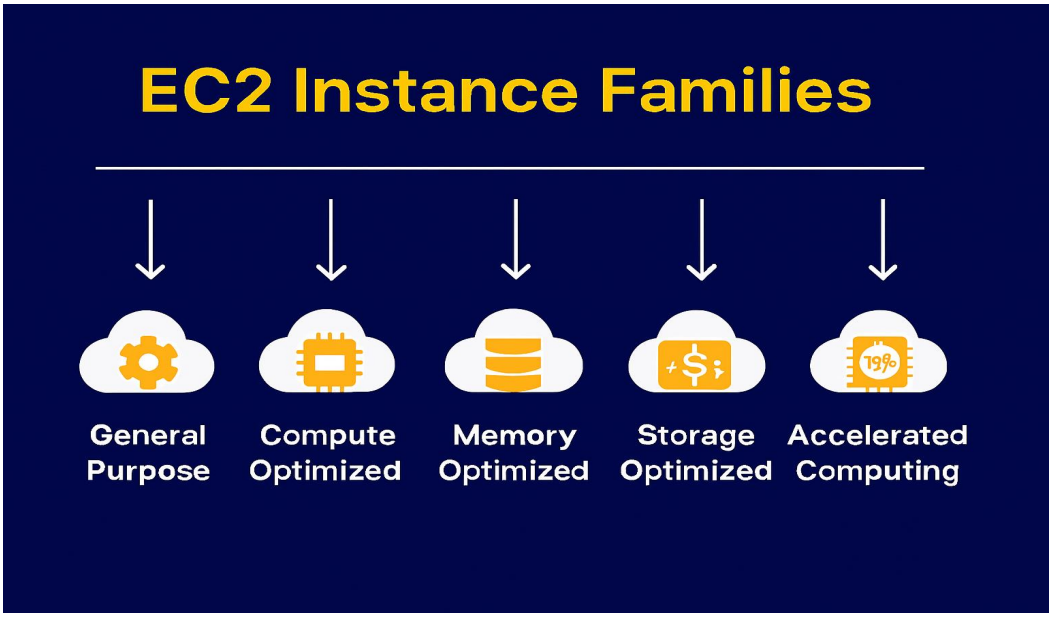
Why Use It?

- No need to buy physical servers.
- Scale up/down quickly.
- Pay only for what you use.

EC2 Instance Basics

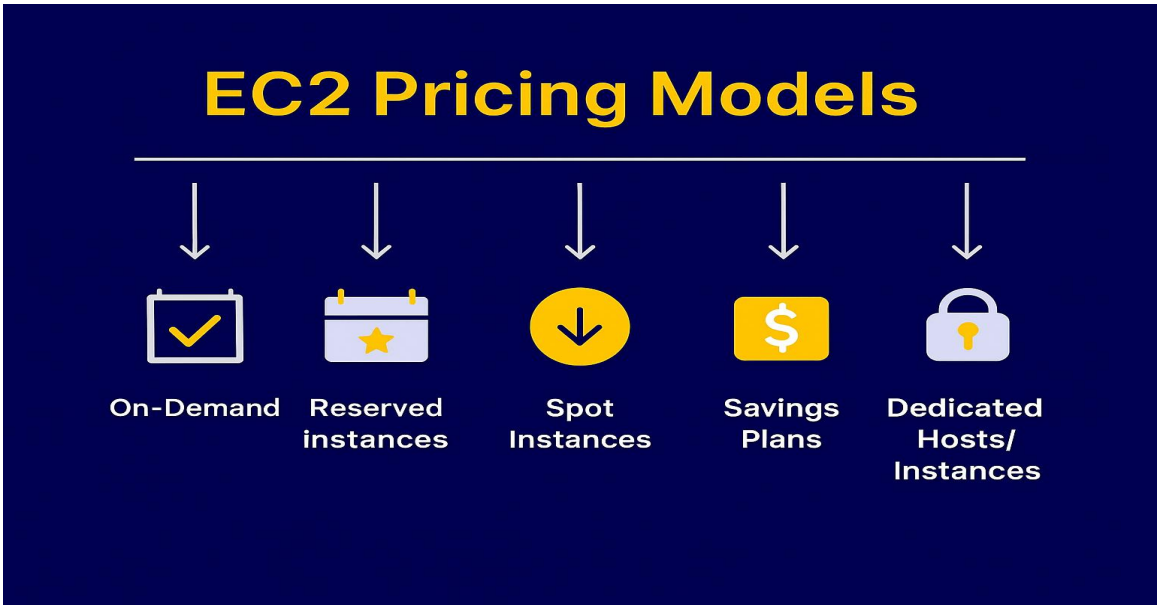
- **Instance**: A virtual server in AWS.
- **AMI (Amazon Machine Image)**: A template that contains the OS + software for your instance (e.g., Amazon Linux, Ubuntu, Windows).

- **Instance Types**: Different combinations of CPU, memory, storage, networking. Categories:
    1. **General Purpose** (e.g., t3.micro) → balanced workloads.
    2. **Compute Optimized** (c5.large) → high performance computing.
    3. **Memory Optimized** (r5.large) → databases, big data.
    4. **Storage Optimized** (i3.large) → heavy I/O apps.
    5. **Accelerated Computing** (p3, g4) → ML/AI, GPUs.



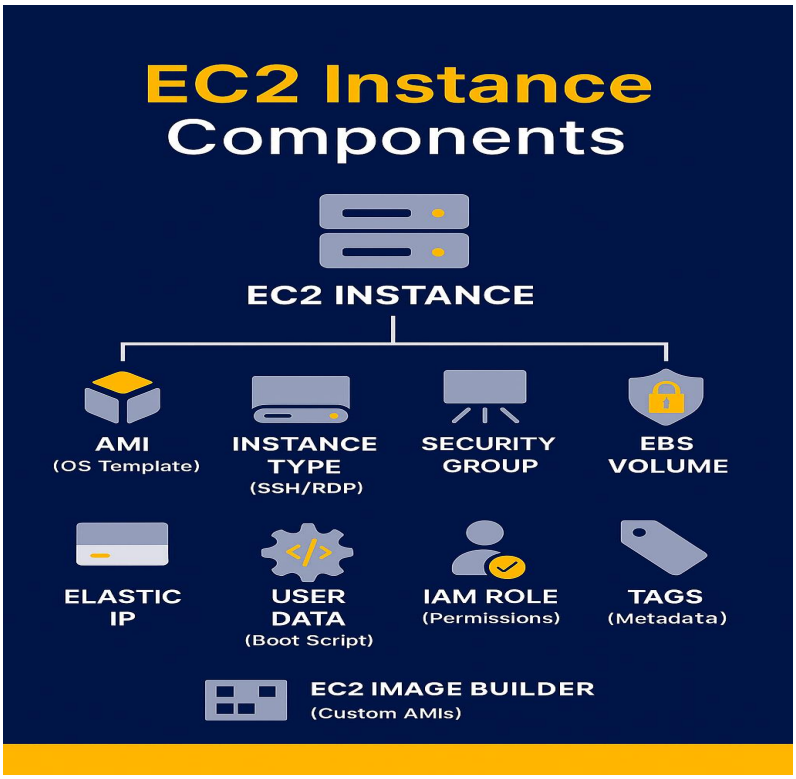
EC2 Pricing Models

1. **On-Demand** → Pay per hour/second, no commitment. Best for short-term use.
2. **Reserved Instances** → 1 or 3-year commitment, cheaper than on-demand.
3. **Spot Instances** → Up to 90% discount but can be interrupted anytime.
4. **Savings Plans** → Commit to $/hr usage, flexible across instance families.
5. **Dedicated Hosts/Instances** → Physical servers for compliance or licensing.



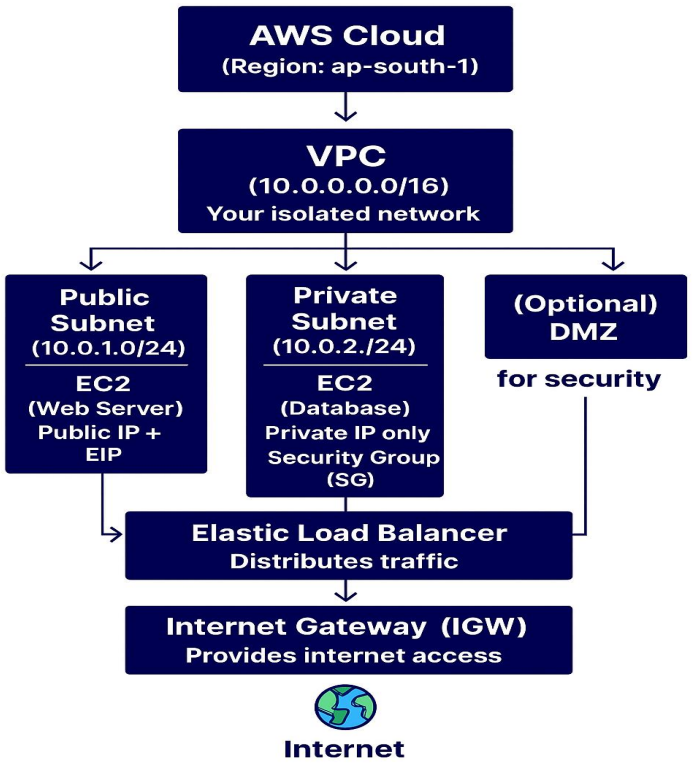
EC2 Key Components

- **Key Pairs** → SSH login to Linux/Windows instance (private key kept safe).
- **Security Groups** → Virtual firewalls controlling inbound/outbound traffic.
- **Elastic IP** → Static public IP address attached to your instance.
- **User Data** → Scripts that run at boot time (e.g., install Apache).



EC2 Networking

- **VPC (Virtual Private Cloud)** → Every EC2 runs inside a VPC.
- **Subnet** → A range of IP addresses in your VPC.
- **Public vs Private Subnets**:
  Public → Accessible from internet.
  Private → Internal apps/databases.
- **ENI (Elastic Network Interface)** → A virtual network card for EC2.
- **EIP (Elastic IP Address)** → Static IP tied to your AWS account.
- **Elastic Load Balancer (ELB)** → Distributes traffic across multiple EC2s.



EC2 Storage Options

- **EBS (Elastic Block Store)**: Persistent storage, like a virtual hard drive.
  Types: SSD (gp3, io2), HDD (st1, sc1).
- **Instance Store**: Temporary storage physically attached to host.

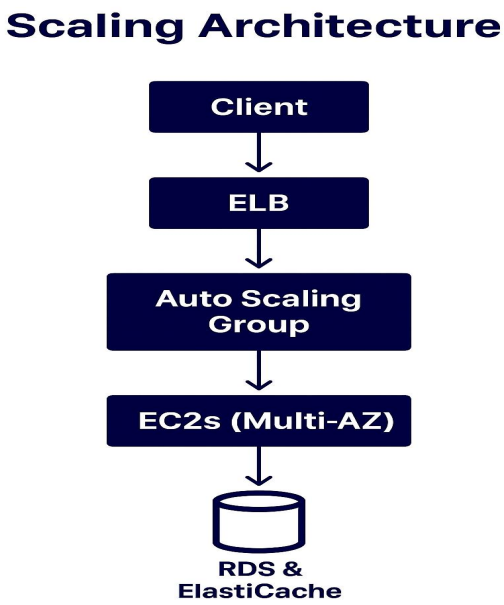
Data lost when instance stops/terminates.

- **EFS (Elastic File System)**: Shared, scalable storage for multiple EC2s.
- **S3 Integration**: Store backups/data in Amazon S3.

Feature	EBS	Instance Store	EFS	S3
Type	Block	Ephemeral Block	File	Object
Durability	Persistent	Ephemeral	Persistent	Extremely High
Performance	High	Very High	Moderate	High
Shared Access	No	No	Yes	Yes (via API)
Mountable	Yes	Yes	Yes	No
Use Case	Boot volumes	Cache/scratch	Shared content	Backups, assets

EC2 Scaling & Availability

- **Auto Scaling Groups (ASG)**: Add/remove instances automatically based on demand.
- **Elastic Load Balancing (ELB)**: Distributes traffic.
- **High Availability**: Deploy across multiple **Availability Zones**
- **Elastic Beanstalk**: Simplifies deployment by managing EC2 + scaling for you.



EC2 Monitoring & Security

- **CloudWatch** → Monitor CPU, memory, disk, network.
- **CloudTrail** → Logs API activity (who launched/stopped EC2).
- **Systems Manager (SSM)** → Remote management without SSH.
- **IAM Roles for EC2** → Give EC2 permissions (e.g., read from S3).
- **Security Best Practices**:
  - Use least privilege IAM roles.
  - Restrict Security Group rules.
  - Rotate keys/passwords.

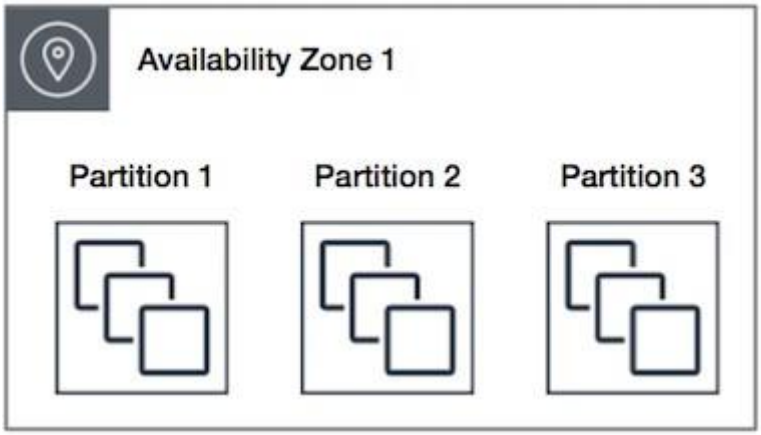
EC2 Advanced Features

- **Placement Groups**:
  - Cluster → Low latency, high bandwidth.

← All packed close together

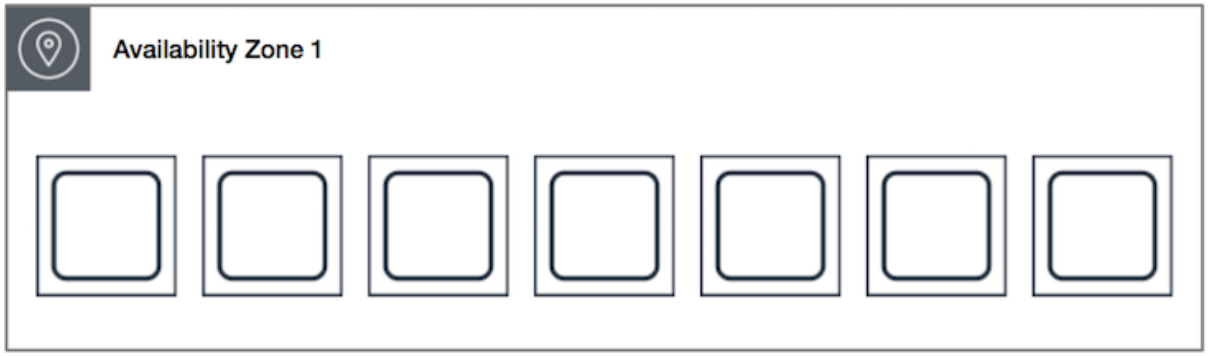
Purpose: High performance, low latency

  - Spread → Spread across hardware (HA).



Purpose: Fault isolation across partitions

  - Partition → Isolated partitions for big data clusters.



← Spread across hardware

Purpose: Maximize availability for critical instances

- **Elastic GPU / Elastic Inference** → Attach GPU power.
- **Hibernate** → Save RAM state, restart faster.
- **Nitro System** → Latest hypervisor for performance/security.
- **Bare Metal Instances** → Direct access to physical hardware.

What's the difference between Security Groups and NACLs?

Concept Explanation

- **Security Groups (SG):**
  - Act as a **virtual firewall** for EC2 instances.
  - Work at the **instance level**.
  - **Stateful** → if inbound traffic is allowed, the outbound response is automatically allowed.
  - Only **allow rules**, no explicit deny.
- **Network ACLs (NACLs):**
  - Control traffic at the **subnet level**.
  - **Stateless** → inbound and outbound rules must be defined separately.
  - Can have both **allow and deny** rules.
  - Useful for **extra layer of security**.

What's the difference between EBS and Instance Store?

Concept Explanation

- **EBS (Elastic Block Store):**
  - Persistent block storage.
  - Data survives stop/start of the instance.
  - Snapshots can be backed up to S3.
  - Suitable for databases, apps, and critical workloads.
- **Instance Store:**
  - Temporary block storage physically attached to the host.

    - Data is **lost** when the instance is stopped or terminated.
    - High I/O performance.
    - Suitable for caches, buffers, temporary data.

What is the difference between Elastic IP and Public IP?

Concept Explanation

- **Public IP:**
    - Assigned automatically when you launch an instance in a public subnet.
    - Changes if you stop and start the instance.
- **Elastic IP (EIP):**
    - Static public IP address allocated to your AWS account.
    - Can be remapped to another instance if needed.
    - Useful for long-lived workloads (e.g., DNS pointing).

EC2 Tenancy Types (Important for interviews)

1. Default (Shared Tenancy) – cheapest
2. Dedicated Instance – runs on single-tenant hardware
3. Dedicated Host – physical server visibility, for licensing
4. Host Reservations

Use Cases:

- Security-sensitive workloads → Dedicated
- BYOL (Bring Your Own License) → Dedicated Host

EC2 Metadata + IMDSv2

Metadata endpoint: http://169.254.169.254/latest/meta-data/

Used for:

- Instance details
- IAM Role temporary credentials

- Network info

IMDSv2 is mandatory for security.

EC2 Lifecycle (States)

- **Pending → Running → Stopping → Stopped → Terminated**.
- **Stop vs Terminate vs Hibernate**:
  - Stop = instance shuts down, restart possible.
  - Terminate = deleted permanently.
  - Hibernate = saves RAM state, resumes faster.

Automation & Customization

- **User Data:** Run bootstrap scripts at instance launch (install software, configure settings).
- **EC2 Image Builder:** Automate creation of custom AMIs.
- **Terraform/CloudFormation:** Launch and manage EC2 via IaC.

EC2 Billing Traps (Real-world important)

- EBS volumes keep billing even after instance stopped
- Unused Elastic IP costs money
- Snapshots stored in S3 (priced separately)
- Inter-AZ data transfer = charged

Default EC2 Limits (per Region)

- **Running On-Demand Instances:**
  - By default, **20 instances per Region** (across all instance families).
  - Some newer accounts may start with **32 vCPUs** as the limit.
- **Spot Instances:**
  - Default is **20 Spot Instances per Region**.

- **Dedicated Hosts / Instances:**
  - Separate limits (often lower).
- **EBS Volumes, Elastic IPs, etc.** also have their own limits.

Connecting ways to EC2:

1. **Mobaxterm:** Download mobaxterm and open → click on new session Remote host → public ip (instance id) → check specify ubuntu

   Advanced SSH setting → check use private key → browse private key in downloads → ok accept → connect to server

2. **Putty:** putty download (64 bit x86) → open → next … → repair → repair → install

   Putty → host name → public ip of EC2 → SSH → auth → cred → browser → ppk key pair → login as ubuntu ec2-user(linux)

   To increase font size, we have to right click → change settings → appearance → change 20 ok apply

3. **Git bash:** open location where we have key pair

   In AWS console click on connect select SSH copy and paste commands to git console then we see it was connected

4. **VS code:** open → … → extension → search → remote → SSH → opens a remote window → connect to SSH host file → save