🔥 AWS EC2 (Elastic Compute Cloud) – Full Guide

1️⃣ What is EC2?

- **Elastic Compute Cloud (EC2)** is a service that provides **resizable compute capacity** in the cloud.
- Simply → It's like renting a **virtual server (VM)** instead of buying physical hardware.
- You can launch, stop, scale, or terminate servers anytime.
- Highly **elastic** → you can increase or decrease resources within minutes.

2️⃣ Why use EC2?

- **Alternative to physical servers**:
  - Buying servers → expensive, fixed capacity, hard to scale.
  - EC2 → pay-as-you-go, scalable, configurable.
- **Cost-efficient** → you only pay for what you use.
- **Flexible** → Choose OS, storage, CPU, memory, networking.
- **Scalable** → integrates with Auto Scaling & Load Balancers.
- **Secure** → tightly integrated with AWS Identity & Access Management (IAM).

3️⃣ EC2 Key Components

🔹 Instance

- A running **virtual server** in AWS cloud.
- Example: t2.micro = 1 vCPU, 1GB RAM (Free Tier).

🔹 Amazon Machine Image (AMI)

- A template that contains OS + pre-installed software.
- Types:
  - AWS-provided AMIs (Amazon Linux, Ubuntu, Windows).
  - Marketplace AMIs (with software like WordPress, Jenkins).
  - Custom AMIs (your own configurations).

🔹 Instance Types

- Define **CPU, memory, storage, and networking capacity**.
- Categories:
    - **General Purpose** (t2, t3, m5) → balance of CPU/RAM.
    - **Compute Optimized** (c5, c6g) → high-performance CPU workloads.
    - **Memory Optimized** (r5, x1e) → large in-memory apps (DB, caching).
    - **Storage Optimized** (i3, d2) → big data, high IOPS storage.
    - **Accelerated Computing** (p2, g4) → ML, GPU workloads.

🔹 **Storage Options**

- **EBS (Elastic Block Store)** → network-attached storage (like SSD/HDD).
- **Instance Store** → temporary storage tied to the instance lifecycle.
- **S3** → object storage (not directly part of EC2, but used for backups).

🔹 **Networking**

- Each EC2 instance runs inside a **VPC (Virtual Private Cloud)**.
- You assign:
    - **Private IP** (inside VPC).
    - **Public IP / Elastic IP** (for internet access).
- **Elastic IP** = static public IP address (doesn't change when instance restarts).

🔹 **Security**

- Managed via **Security Groups (SGs)** → act like **virtual firewalls**.
- **Inbound Rules** → define what traffic is allowed into EC2.
- **Outbound Rules** → define what traffic can go out.

4️⃣ EC2 Setup (Step-by-Step)

Launching an Instance:

1. **Go to AWS EC2 Console** → choose a region.
2. Click **Launch Instance**.
3. Give **name** (server identifier).
4. **Select AMI** (e.g., Ubuntu 22.04).
5. **Choose Instance Type** (e.g., t2.micro).

6. **Create Key Pair**:
    - .pem file → for SSH login.
    - Must keep it safe, else you cannot access instance.
7. **Configure Network**:
    - Assign Public IP if you need internet access.
    - Place instance in a VPC subnet.
    - Attach **Security Group**.
8. **Add Storage** → define root volume size/type.
9. **Review and Launch**.
10. **Access Instance**:
11. ssh -i key.pem ubuntu@<public-ip>

5️⃣ Security Groups & Ports

Security Groups control **who can access your instance and how**.

Common Ports to Know:

Protocol	Port	Purpose
SSH	22	Access Linux instances
RDP	3389	Access Windows instances
HTTP	80	Serve web traffic
HTTPS	443	Secure web traffic
MySQL	3306	Database access
PostgreSQL	5432	Database access
MongoDB	27017	Database access
Custom TCP	8080	APIs, app servers
Custom UDP	1194	VPN connections

⚠️ Always restrict SSH/RDP to **your IP only** for security.

6️⃣ Pricing Models

- **On-Demand** → pay by the hour/second. Best for short-term workloads.
- **Reserved Instances** → commit for 1–3 years → cheaper.
- **Spot Instances** → bid for unused capacity (up to 90% discount).
- **Savings Plans** → flexible alternative to reserved pricing.

7️⃣ EC2 CLI Commands (Interview Bonus)

```
# List instances
aws ec2 describe-instances

# Start instance
aws ec2 start-instances --instance-ids <id>

# Stop instance
aws ec2 stop-instances --instance-ids <id>

# Terminate instance
aws ec2 terminate-instances --instance-ids <id>
```

8️⃣ Common Interview Questions (Internship Level)

Basics

- What is AWS EC2?
- How is EC2 different from a physical server?
- What is an AMI?

Configuration

- What is an instance type?
- Difference between EBS and Instance Store?
- What are reserved vs spot vs on-demand instances?

Security & Networking

- What is a Security Group?

- Difference between Security Group and NACL?
- What ports do you open for a web server?

Troubleshooting

- You launched EC2 but cannot SSH. What will you check?
    - Key pair permissions (chmod 400 key.pem).
    - Security group inbound rule for port 22.
    - Public IP attached.
- How to make sure an IP doesn't change when you restart?
    - Use **Elastic IP**.

Advanced

- How do you scale EC2 automatically?
    - Use **Auto Scaling Groups (ASG)**.
- How to connect EC2 with DynamoDB/Lambda?
    - Use **IAM roles** attached to EC2.

1️⃣ **Inbound Rules (Traffic into EC2)**

- Control **what type of traffic is allowed to reach your instance**.
- Example:
    - If you're running a web server, you need to allow:
        - **Port 80 (HTTP)** → for normal web traffic.
        - **Port 443 (HTTPS)** → for secure web traffic.
    - If you want to log into the server:
        - **Port 22 (SSH)** → for Linux.
        - **Port 3389 (RDP)** → for Windows.

✅ If you don't allow it in **Inbound Rules**, traffic from outside **cannot reach** your instance.

2️⃣ **Outbound Rules (Traffic out of EC2)**

- Control **what type of traffic your instance is allowed to send out**.
- By default → AWS allows **all outbound traffic** (so your instance can download updates, call APIs, connect to DBs).

- Example:
    - If your instance needs to:
        - Download OS updates → needs outbound access to the internet (port 80/443).
        - Call an external API → needs outbound HTTPS (443).
        - Connect to an RDS database in another VPC → outbound rule for DB port (e.g., 3306 for MySQL).

✅ If you block outbound traffic, your server **cannot reach outside** (internet, APIs, databases).

3️⃣ Stateful Behavior (Important Interview Point)

- Security Groups are **stateful**:
    - If you allow inbound traffic on a port, the **response traffic is automatically allowed outbound**, even if outbound rules don't explicitly allow it.
    - Same for outbound → if instance sends a request out, the return response is allowed back in.

👉 Example:

- If you allow inbound HTTP (port 80), and someone visits your site:
    - Request comes **inbound on port 80**.
    - Your server sends a **response outbound on port 80**.
    - That response is automatically allowed due to stateful behavior.

4️⃣ Example Scenario

Imagine you launched a **web server on EC2** (Ubuntu + Apache):

🔹 Inbound Rules:

- Allow **SSH (22)** only from your IP → so only you can log in.
- Allow **HTTP (80)** from 0.0.0.0/0 → so anyone on the internet can access your website.
- Allow **HTTPS (443)** from 0.0.0.0/0 → secure access.

🔹 Outbound Rules:

- Allow **all traffic** (default) → server can download security patches, fetch data from APIs, etc.

5️⃣ Interview Trick Question

If I remove all outbound rules, but keep inbound rules for HTTP/SSH, will my EC2 still work?

- Answer:
    - You will **still be able to connect via SSH and HTTP requests will reach the server**, because inbound allows them.
    - BUT your EC2 instance won't be able to:
        - Download software updates.
        - Call external APIs.
        - Push logs to external services.
    - So inbound access works, but the instance is **isolated** outbound.

✅ Summary:

- **Inbound rules → who can talk to my server?**
- **Outbound rules → who can my server talk to?**
- **Stateful → once a connection is allowed in one direction, the response is automatically allowed in the opposite direction.**