Day 2: AWS Compute & Networking – Full Documentation

Topics Covered:

- EC2: Launching, Connecting, Managing Instances
- AMIs, Key Pairs, Security Groups
- ELB (Elastic Load Balancer)
- Auto Scaling
- AWS VPC: Subnets, Route Tables, IGW, NAT
- Elastic IP, ENI
- **Lab 1**: Launch EC2 instance in custom VPC
- **Lab 2**: Setup ELB and Auto Scaling Group

✅ 1. EC2 – Elastic Compute Cloud

📄 What is EC2?

EC2 provides scalable virtual servers (instances) in the cloud.

🔧 Key Concepts:

- **AMI (Amazon Machine Image):** Blueprint of the instance (OS + software).
- **Instance Type:** Hardware config (t2.micro = 1 vCPU, 1 GB RAM - free tier).
- **Key Pair:** For SSH connection. Keep `.pem` file safe.
- **Security Group:** Virtual firewall – control traffic via inbound/outbound rules.

🛠️ Lab 1: Launch EC2 in Mumbai Region

Step-by-step:

1. **Go to EC2 Console:**
   AWS Console > Services > EC2
2. **Click "Launch Instance"**
3. **Configure:**
   - Name: `My-EC2`
   - AMI: Amazon Linux 2023 (Free Tier eligible)
   - Instance Type: `t2.micro`
   - Key Pair: Create new OR use existing → Download `.pem` file
   - Network: Default VPC or your own VPC
   - Security Group: Create new → Add rule → Allow SSH (port 22), HTTP (80)
4. **Launch instance**
5. **Connect to EC2:**

Use SSH terminal:

```
ssh -i "your-key.pem" ec2-user@<public-ip>
```

   -
   - For Windows: Use PuTTY or Git Bash

✅ 2. VPC – Virtual Private Cloud

📄 What is VPC?

Private network isolated within AWS for secure resources.

🧱 Components:

- **Subnets** (Public & Private)
- **Route Tables**
- **Internet Gateway (IGW)**
- **NAT Gateway (optional)**
- **Elastic IPs**

🛠️ Lab 2: Create Custom VPC & Launch Public EC2

Step-by-step:

1. **Go to VPC Dashboard:**
   AWS Console > Services > VPC
2. **Click "Create VPC" → Select "VPC Only"**
   - Name: `My-Custom-VPC`
   - IPv4 CIDR block: `10.0.0.0/16`
3. **Create Subnet:**
   - Public Subnet: `10.0.1.0/24`, AZ: `ap-south-1a`
   - Private Subnet (optional): `10.0.2.0/24`, AZ: `ap-south-1b`
4. **Create Internet Gateway:**
   - Name: `My-IGW`
   - Attach to `My-Custom-VPC`
5. **Create Route Table:**
   - Associate with Public Subnet
   - Add route:
     - Destination: `0.0.0.0/0`

        - Target: `Internet Gateway`

6. **Launch EC2 in Custom VPC:**
    - Select public subnet
    - Enable Auto-assign public IP
7. **Security Group:**
    - Allow SSH and HTTP access
8. **Elastic IP:**
    - Allocate from EC2 dashboard
    - Associate with EC2 instance

✅ 3. ELB – Elastic Load Balancer

📄 What is ELB?

Distributes traffic across multiple EC2 instances.

🔧 Types:

- Application Load Balancer (ALB) – HTTP/HTTPS
- Network Load Balancer (NLB) – TCP
- Classic Load Balancer (legacy)

🛠️ Lab 3: Setup ELB + Auto Scaling Group

Step-by-step:

1. **Launch 2 EC2 Instances** (use same config, AMI)
2. **Create Target Group:**

   - Type: Instance
   - Protocol: HTTP:80
   - Register both instances

3. **Create Application Load Balancer:**
   - Name: `My-ALB`
   - Scheme: Internet-facing
   - Listeners: HTTP:80
   - VPC: Select your custom VPC
   - Subnets: Select at least 2 AZs
   - Security Group: Allow HTTP (80)
4. **Connect ALB to Target Group**
5. **Test Load Balancer Endpoint**
   - Visit: `http://<ALB-DNS-Name>`
   - It should load traffic from both EC2s

✅ 4. Auto Scaling Group (ASG)

📄 What is ASG?

Automatically increases/decreases EC2 instances based on demand.

🛠️ Lab 4: Setup Auto Scaling

1. **Create Launch Template**
   - AMI: Amazon Linux 2023
   - Instance Type: t2.micro

- Security Group: Allow HTTP

2. **Create Auto Scaling Group:**
   - Launch template: select above
   - VPC: Custom VPC
   - Subnets: at least 2 AZs
   - Attach to existing Target Group (from ELB)
   - Desired capacity: 2
     Min: 1
     Max: 3
3. **Scaling Policy:**
   - Use CPU utilization > 60% to trigger scale-up

🧾 Summary:

Service	Purpose
EC2	Virtual server in the cloud
VPC	Isolated network to run resources
ELB	Distribute load to multiple EC2s
Auto Scaling	Automatically scale EC2s
Subnets	Divide VPC into smaller segments
IGW	Allow internet access
Elastic IP	Static public IP for EC2