

1)VPC CREATION:

Step 1: Go to the VPC Dashboard

- In the AWS Console, go to VPC.
- Click “Your VPCs” in the left-hand menu.
- Click the “Create VPC” button.

Step 2: Choose VPC Creation Method

- Choose “VPC only”

Step 3: Configure VPC Settings

- Name tag: (e.g., myvp)
- IPv4 CIDR block: (e.g., 10.0.0.0/16)

Click Create VPC.

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
myvp	vpc-0fbff48d46c80each	Available	Off	10.0.0.0/16	-

2) SUBNET CREATION:

Step 1: go to **Subnets** → Click **Create Subnet**.

Step 2: In the Create Subnet page:

- **Select your VPC.**
- **Create your first subnet:**
 - **Name:** (e.g., Public Subnet 1)
 - **AZ:** Choose (eu-north-1a)
 - **CIDR block:** e.g., 10.0.1.0/24
- Click **Add another subnet** (on same page):
 - **Name:** (e.g., Private Subnet 1)
 - **AZ:** Choose (eu-north-1a)
 - **CIDR block:** e.g., 10.0.2.0/24
- Click **Add another subnet** (on same page):
 - **Name:** (e.g., Public Subnet 2)
 - **AZ:** Choose (eu-north-1b)
 - **CIDR block:** e.g., 10.0.3.0/24
- Click **Add another subnet** (on same page):
 - **Name:** (e.g., Private Subnet 2)
 - **AZ:** Choose (eu-north-1b)
 - **CIDR block:** e.g., 10.0.4.0/24

The screenshot shows the AWS VPC dashboard with the Subnets section selected. The table lists four subnets:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
Public Subnet 2	subnet-034a8643286391232	Available	vpc-0fbff48d46c80eacb myvp	Off	10.0.3.0/24
Private Subnet 2	subnet-0e678deed603a856a	Available	vpc-0fbff48d46c80eacb myvp	Off	10.0.4.0/24
Public Subnet 1	subnet-0a3a77ffcc827e815	Available	vpc-0fbff48d46c80eacb myvp	Off	10.0.1.0/24
Private Subnet 1	subnet-00da7fac75f8d53d8	Available	vpc-0fbff48d46c80eacb myvp	Off	10.0.2.0/24

3)EC2 INSTANCE CREATION:

Step 1: Go to the EC2 Dashboard

- In the AWS Console, go to **EC2**.
- Click “**Instances**” in the left menu.
- Click the “**Launch Instance**” button.

Step 2: Configure Basic Settings

1. Name:

- Give your instance a name (e.g., myec2)

2. Application and OS Image (AMI):

- Choose Ubuntu

3. Instance Type:

- Select t3.micro (free tier eligible)

4. Key Pair (Login):

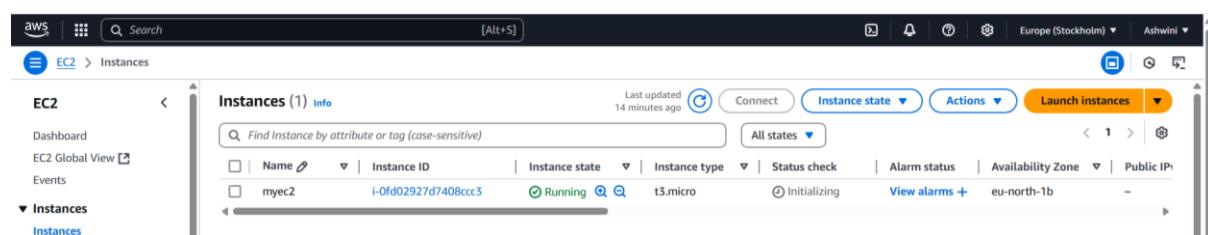
- Create or choose an existing key pair(e.g, webserver)

5. Network Settings:

- **VPC:** Choose the VPC you created (myvp)
- **Subnet:** Pick a subnet within that VPC
- **Auto-assign Public IP:** Set to **Disable** if you don't want internet access
- **Firewall (security group):**
 - Select an existing group(e.g, default)

Step 3: Launch

- Click “**Launch Instance**”



4) VPC ENDPOINT FOR S3:

Step 1: Go to the VPC Dashboard:

- In the AWS Console, go to **VPC**.
- Click on **Endpoints** in the left menu.
- Click **Create Endpoint**.

2: name tag (e.g, myendpoint)

3: Choose Service Category:

- Select **AWS services**.

4: Find the S3 Service:

- In the **Service Name** section, search for the **S3 service** specific to your region.
- In the **Service Name** list, it should look like:
com.amazonaws.<region>.s3
(Example: com.amazonaws.eu-north-1.s3)

5: Configure the Endpoint:

- **VPC**: Choose the VPC that your EC2 instance resides in.
- **Route Tables**: Choose the route tables for the subnets that need access to S3.
- **Policy**: Either choose the default full access or create a custom policy.

6: Create the Endpoint:

- Click **Create Endpoint**.
- The VPC endpoint will now be available and route traffic from EC2 to S3 without leaving the AWS network.

The screenshot shows the AWS VPC Dashboard. A green notification bar at the top center says "Successfully created VPC endpoint vpce-069a791a4412b5dfd". Below the notification, the main interface displays the "Endpoints (1/1)" section. A table lists one endpoint named "myendpoint" with the VPC endpoint ID "vpce-069a791a4412b5dfd", status "Available", and service name "com.amazonaws.eu-r". The left sidebar shows navigation options like "VPC dashboard", "EC2 Global View", and "Virtual private cloud".

Step 2: Configure Security Groups for EC2

1. Go to EC2 Dashboard:

- Navigate to **EC2 > Instances**.
- Select your EC2 instance.

2. Modify Security Group:

- In the **Security** tab, click on the **Security Group ID**.
- Go to **Inbound Rules** and ensure that **Port 80** (HTTP) or **Port 443** (HTTPS) is open for communication.

The screenshot shows the AWS EC2 Security Groups interface. A green success message at the top states: "Inbound security group rules successfully modified on security group (sg-03f174f1161e74762 | default)". Below it, the security group details are shown: Security group name (default), Security group ID (sg-03f174f1161e74762), Description (default VPC security group), and VPC ID (vpc-0fbfb48d46c80each). The Inbound rules section displays one rule: Name (sgr-09305434949022217), IP version (-), Type (HTTP), Protocol (TCP), and Port range (80).

3. Outbound Rules:

- Outbound traffic is allowed by default, but verify that the EC2 instance can communicate with the VPC endpoint.

Step 3: Configure S3 Bucket Policy

1. Go to S3 Dashboard:

- In the AWS Console, navigate to **S3**.
- Select the **S3 bucket** you want your EC2 instance to access.

2. Edit Bucket Policy:

- Click on the **Permissions** tab.
- Scroll down to **Bucket Policy** and click **Edit**.

3. Add a Bucket Policy for VPC Endpoint Access:

To ensure that only traffic from your VPC and VPC endpoint can access the S3 bucket, you need to create a policy restricting access.

Example bucket policy:

```
{
```

```
"Version": "2012-10-17",
```

```
"Statement": [
```

```
{
```

```
    "Effect": "Allow",
```

```
"Principal": "*",
"Action": "s3:GetObject",
"Resource": "arn:aws:s3:::buck.my/*",# REPLACE
"Condition": {
    "StringEquals": {
        "aws:SourceVpc": " vpc-0fbbf48d46c80eacb"# REPLACE
    }
}
}
]
```

4. Save the Policy:

- Click **Save** to apply the policy.

5) Set up ALB (Application Load Balancer)

1. Go to **EC2 Dashboard** → **Load Balancers** → **Create Load Balancer** → **Application Load Balancer**.

Fill details:

- **Name:** myalb.
- **Scheme:** Internet-facing
- **IP address type:** IPv4.
- **Listeners:** Create listener for HTTP on port 80.

2. Configure Availability Zones

- **Select your VPC(myvp)**
- **Tick both Public Subnets:**
 - Public Subnet 1 (AZ1)
 - Public Subnet 2 (AZ2)

3. Configure Security Group for ALB

- Allow **inbound HTTP (80)**
- Outbound can stay default (allow all).

4. Attach Target Group

- In "Listeners" section:
 - Forward traffic to your **Target Group** you created earlier.

The screenshot shows the AWS EC2 Target groups page. On the left, there's a sidebar with navigation links like AMI Catalog, Elastic Block Store, Network & Security, and Security Groups. The main area has a heading 'Target groups (1) Info'. A table lists one target group: 'mytarget' with ARN 'arn:aws:elasticloadbalancing:eu-north-1:123456789012:targetgroup/mytarget'. The table includes columns for Name, ARN, Port, Protocol, Target type, Load balancer (which is currently 'None associated'), and VPC ID ('vpc-0fbff4'). There are 'Actions' and 'Create target group' buttons at the top right.

5. Review and Create

- Review all settings.
- Click **Create Load Balancer**.

The screenshot shows the AWS EC2 Load balancers page. The sidebar is identical to the previous screenshot. The main area has a heading 'Load balancers (1/1)'. A table lists one load balancer: 'myalb' with ARN 'arn:aws:elasticloadbalancing:eu-north-1:123456789012:loadbalancer/myalb'. The table includes columns for Name, DNS name ('myalb-1785366354.eu-north-1'), State ('Active'), VPC ID ('vpc-0fbff48d46c80eacb'), Availability Zones ('2 Availability Zones'), Type ('application'), and Date create ('April 27, 2023'). There are 'Actions' and 'Create load balancer' buttons at the top right.