Shaikh Ateeb Ahmed
07-08-2025

AMI-Based Apache Web Server Deployment on AWS EC2

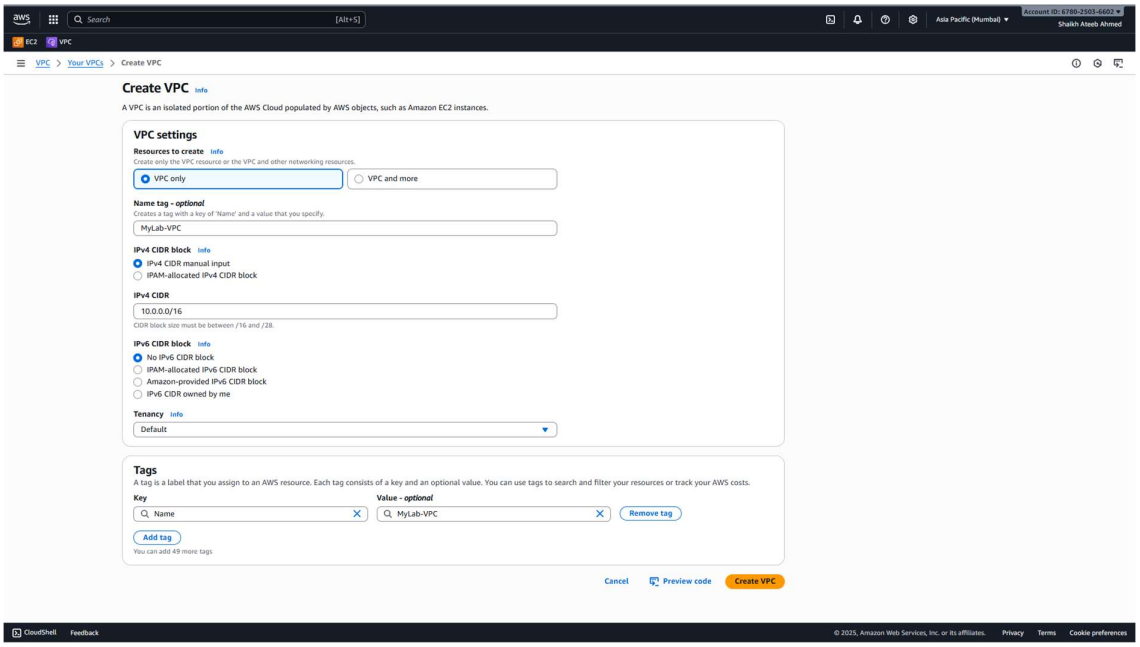
Create a VPC

1. Go to **VPC Console** (search “VPC” in AWS Console).
2. Left menu → **Your VPCs** → **Create VPC**.
3. Choose **VPC only**.
4. Fill in:

   Name tag: MyLab-VPC

   IPv4 CIDR block: 10.0.0.0/16 (gives 65k IPs, fine for labs)

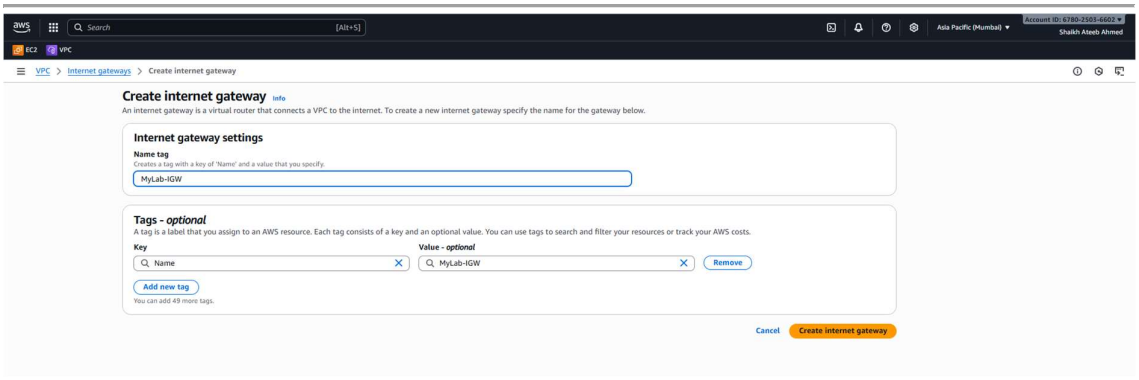
   Leave other defaults.
5. Click **Create VPC**.



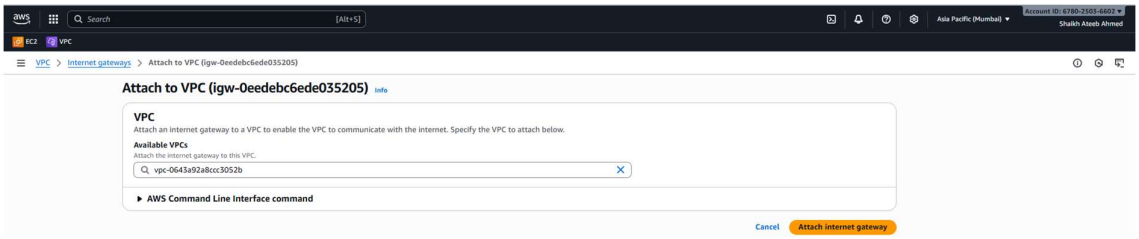
Create an Internet Gateway (IGW)

1. Left menu → **Internet gateways** → **Create internet gateway**.
2. Name: MyLab-IGW → Create.

Shaikh Ateeb Ahmed
07-08-2025



3. Select it → **Actions → Attach to VPC** → choose MyLab-VPC.



Create a Public Subnet

1. Left menu → **Subnets** → **Create subnet**.
2. Choose:

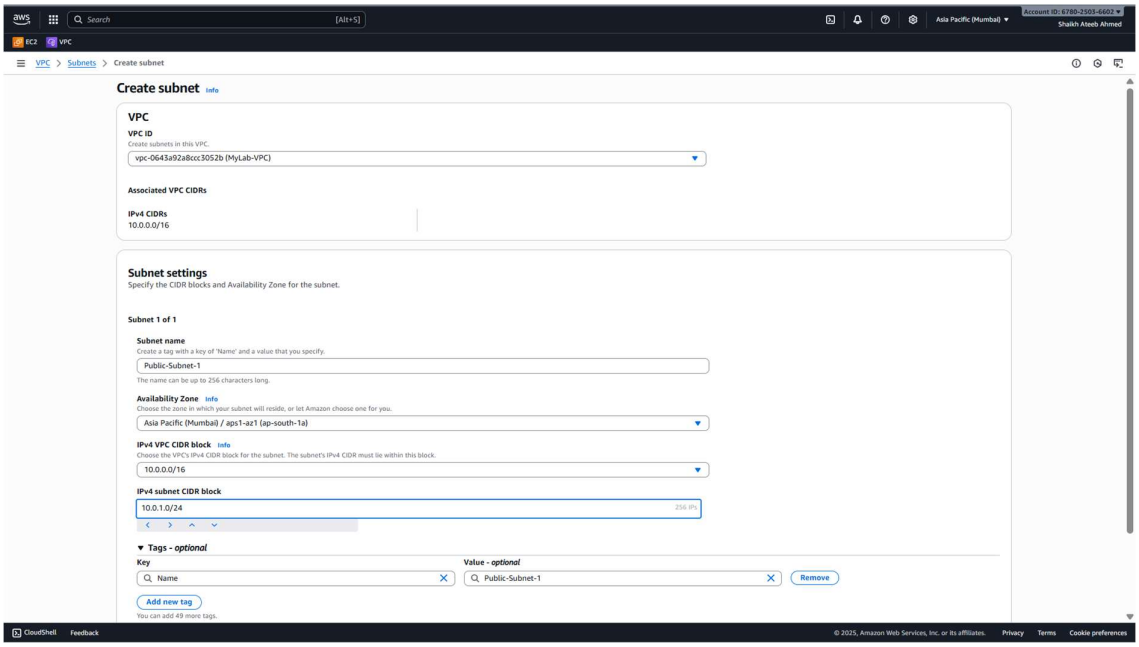
   VPC: MyLab-VPC

   Subnet name: Public-Subnet-1

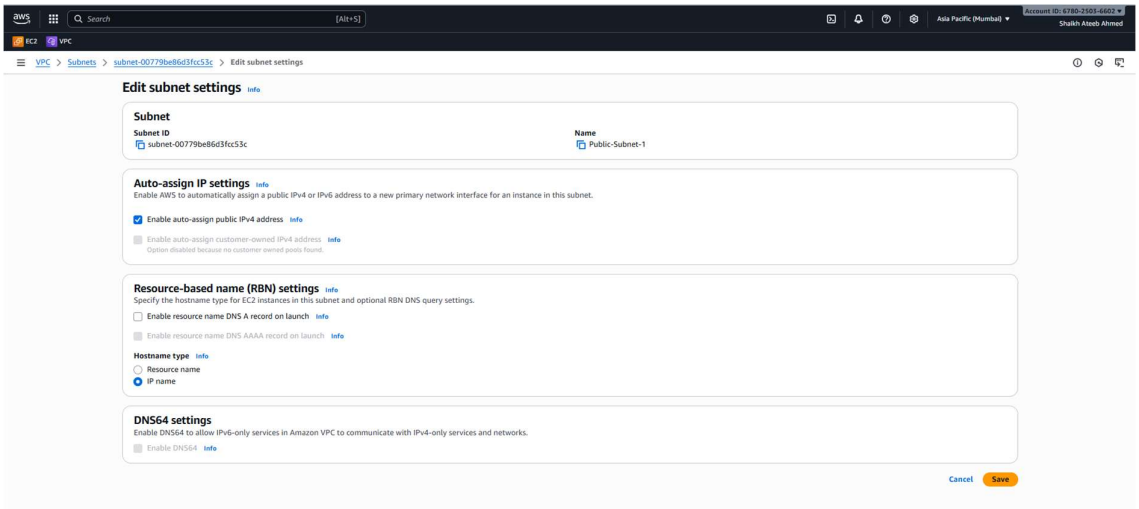
   Availability Zone: e.g. ap-south-1a

   IPv4 CIDR block: 10.0.1.0/24 (gives 256 IPs)
3. Click **Create subnet**.

Shaikh Ateeb Ahmed
07-08-2025



4. Go to **Subnets → select Public-Subnet-1**.
5. Actions → **Modify subnet settings**.
6. Check **Enable auto-assign public IPv4 address** → Save.



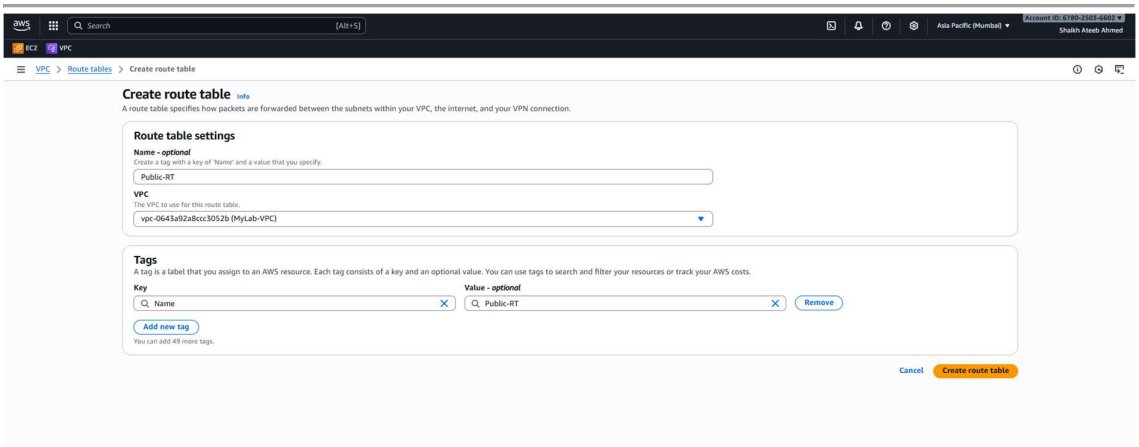
Create and Configure a Route Table

1. Left menu → **Route tables** → **Create route table**.

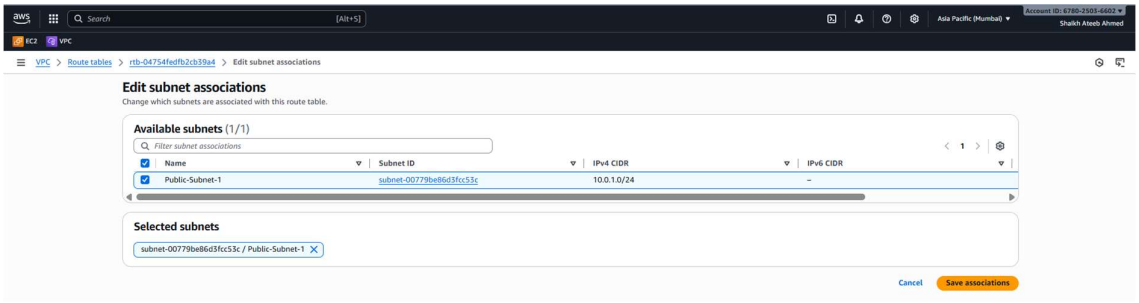
   Name: Public-RT

   VPC: MyLab-VPC → Create.

Shaikh Ateeb Ahmed
07-08-2025



2. Select Public-RT → **Subnet associations** → **Edit subnet associations** → check Public-Subnet-1 → Save.

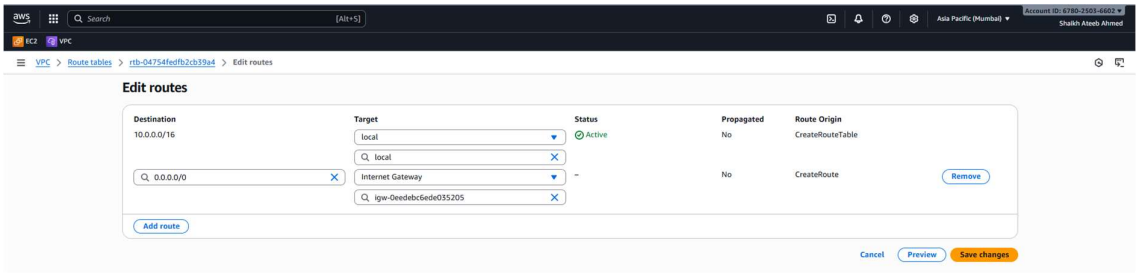


3. Go to **Routes tab** → **Edit routes** → Add route:

   Destination: 0.0.0.0/0

   Target: your MyLab-IGW

   Save changes.



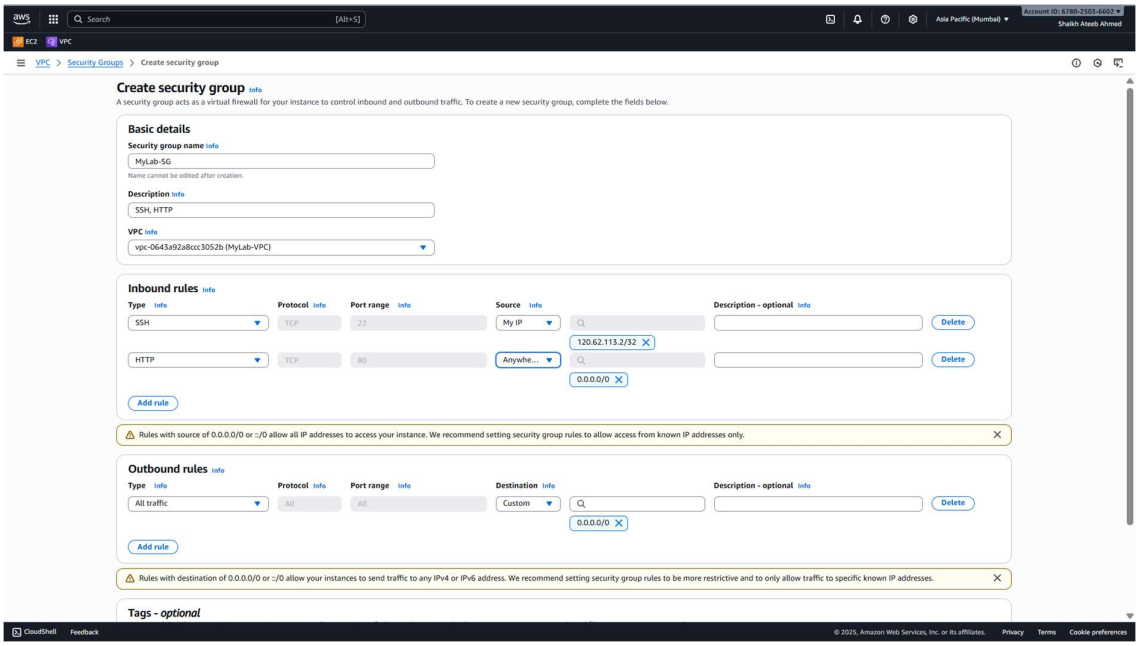
Now your subnet is “public” (it routes traffic to the Internet).

Security Group

1. Choose **Create a new security group**.
2. Creating new, add rules:

Shaikh Ateeb Ahmed
07-08-2025

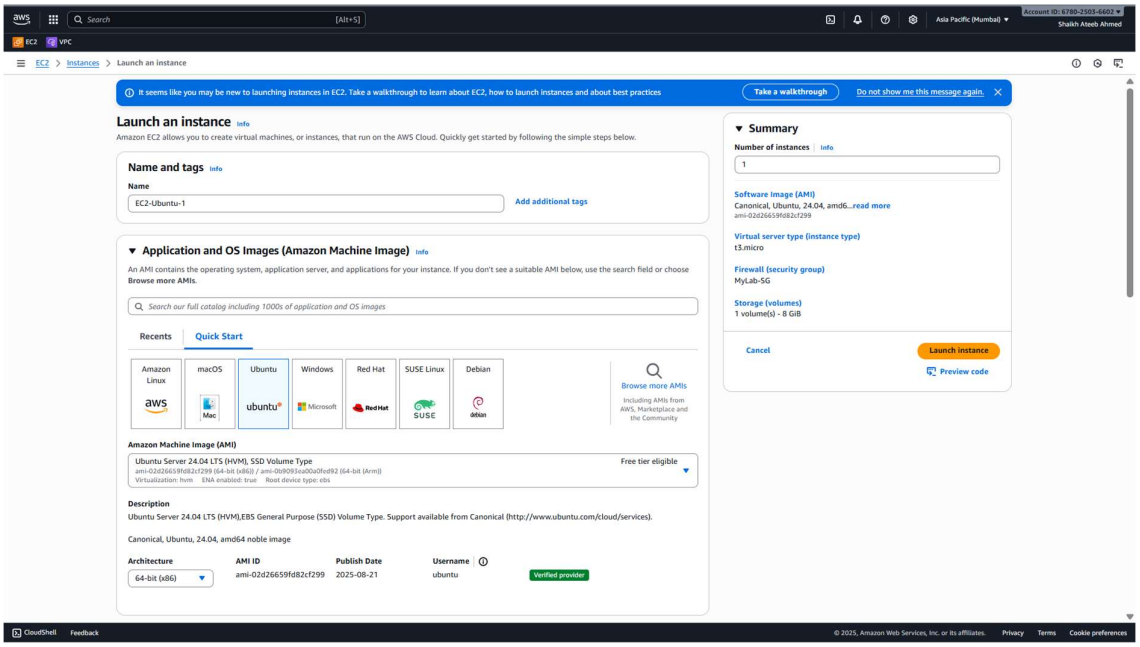
SSH → Type: **SSH**, Protocol: TCP, Port: 22, Source: **My IP** (choose “My IP” to restrict).
HTTP → Type: **HTTP**, Protocol: TCP, Port: 80, Source: **0.0.0.0/0** (allows public access to web server).

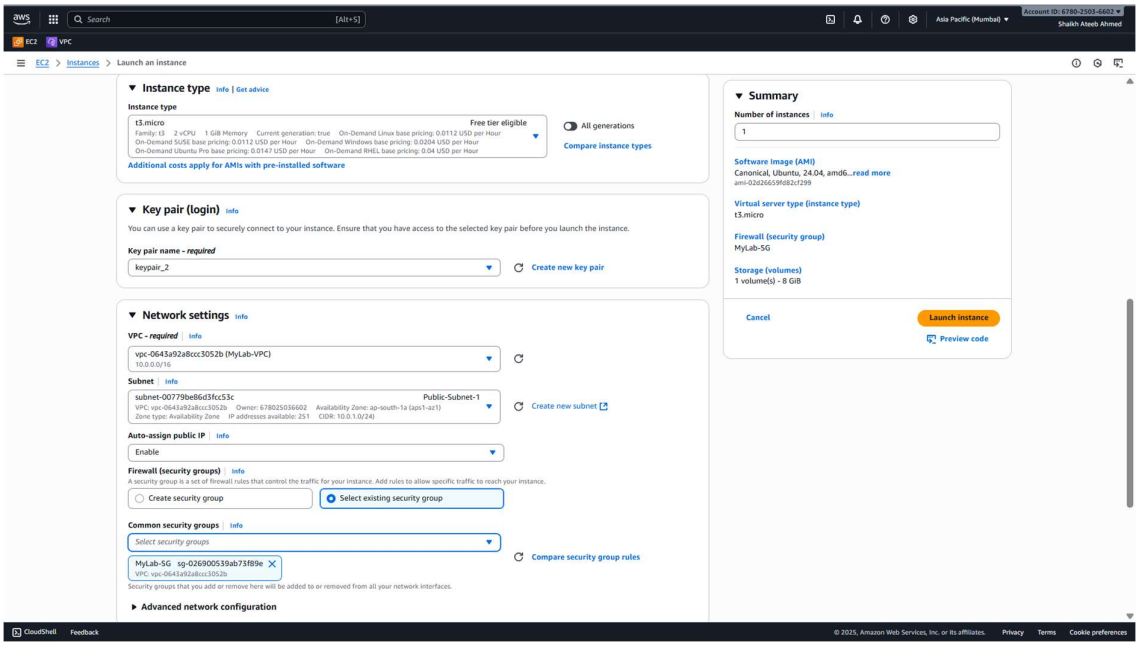


Launch Ubuntu EC2 (Console)

1. EC2 → Instances → **Launch instances**.
2. Give name eg. EC2-Ubuntu-1
3. *Image*: choose **Ubuntu Server**.
4. *Instance type*: e.g. t3.micro (or whatever).
5. *Key pair*: select existing.
6. *Network / subnet*: pick a public subnet (ensure **Auto-assign Public IP = Enable**).
7. *Security group*: choose that we have created above.
8. Launch.

Shaikh Ateeb Ahmed
07-08-2025





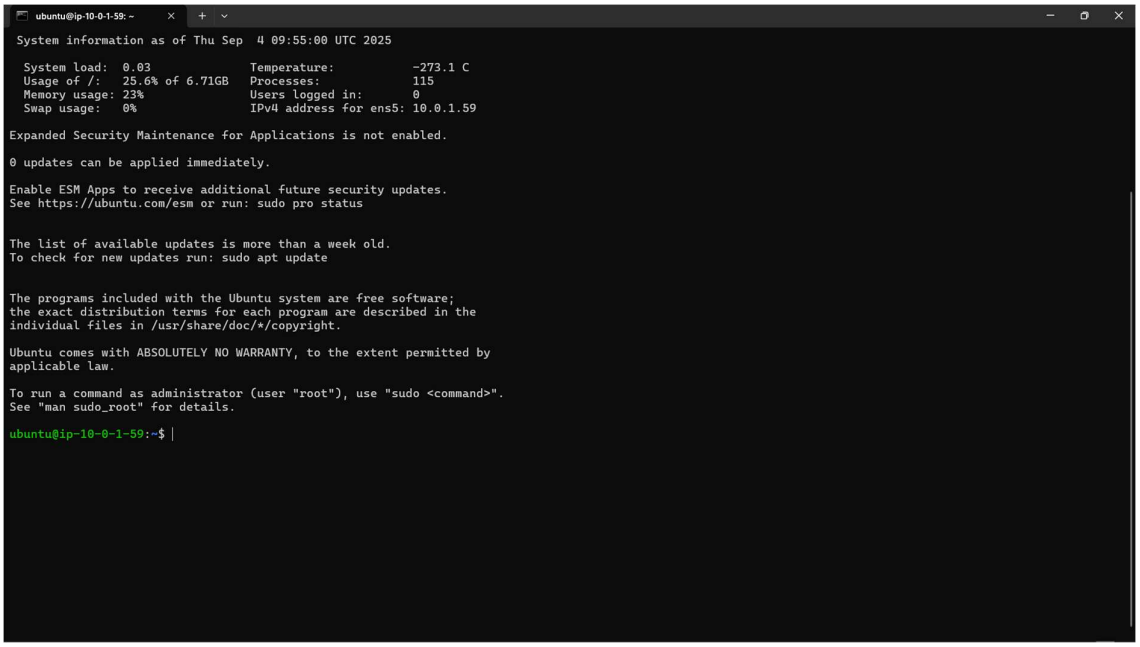
SSH in, install Apache, verify (commands to run on your machine after instance is running)

Replace <KEY.pem> and <PUBLIC_IP> with your values.

chmod 400 <KEY.pem>

ssh -i "<KEY.pem>" ubuntu@<PUBLIC_IP>

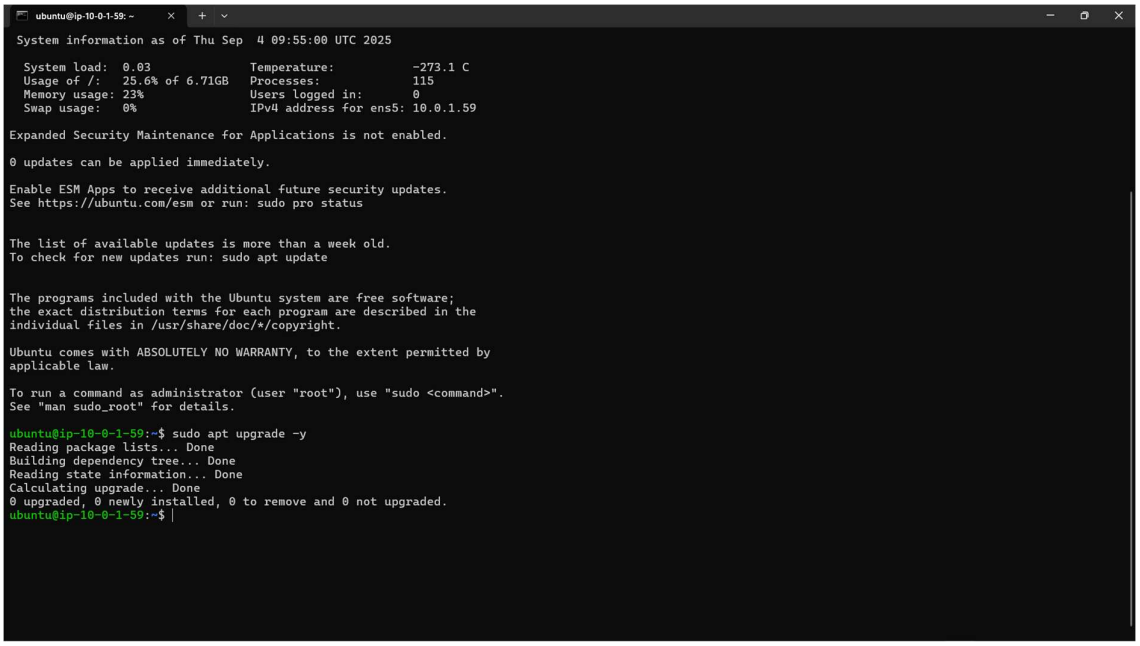
Shaikh Ateeb Ahmed
07-08-2025



Once on the instance:

update packages

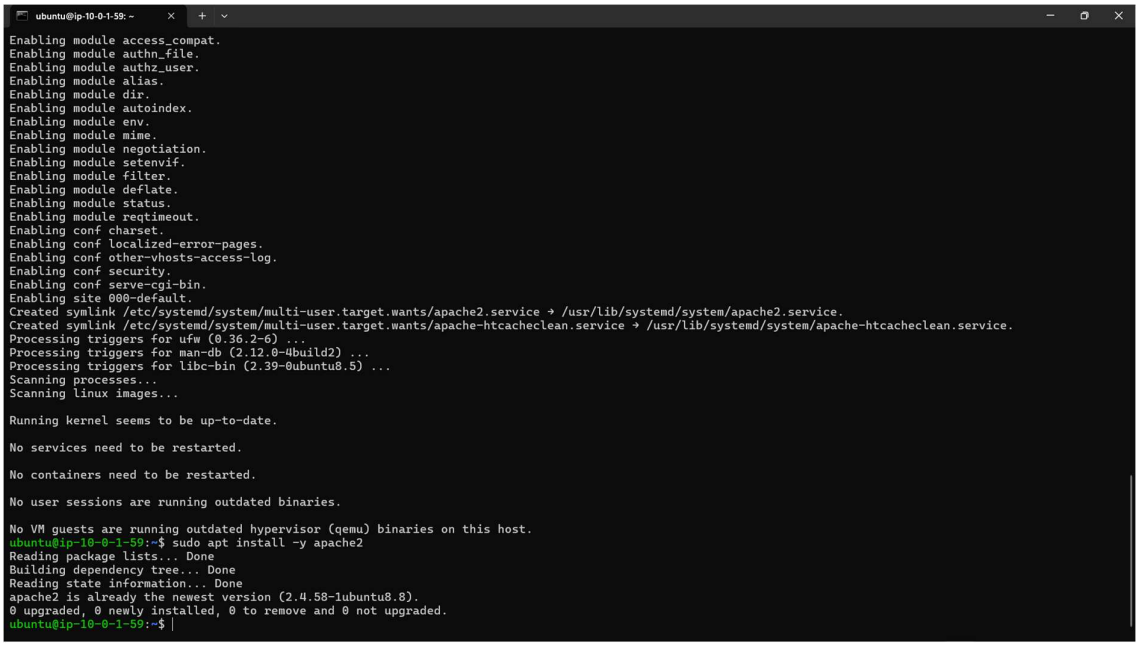
sudo apt upgrade -y



install Apache

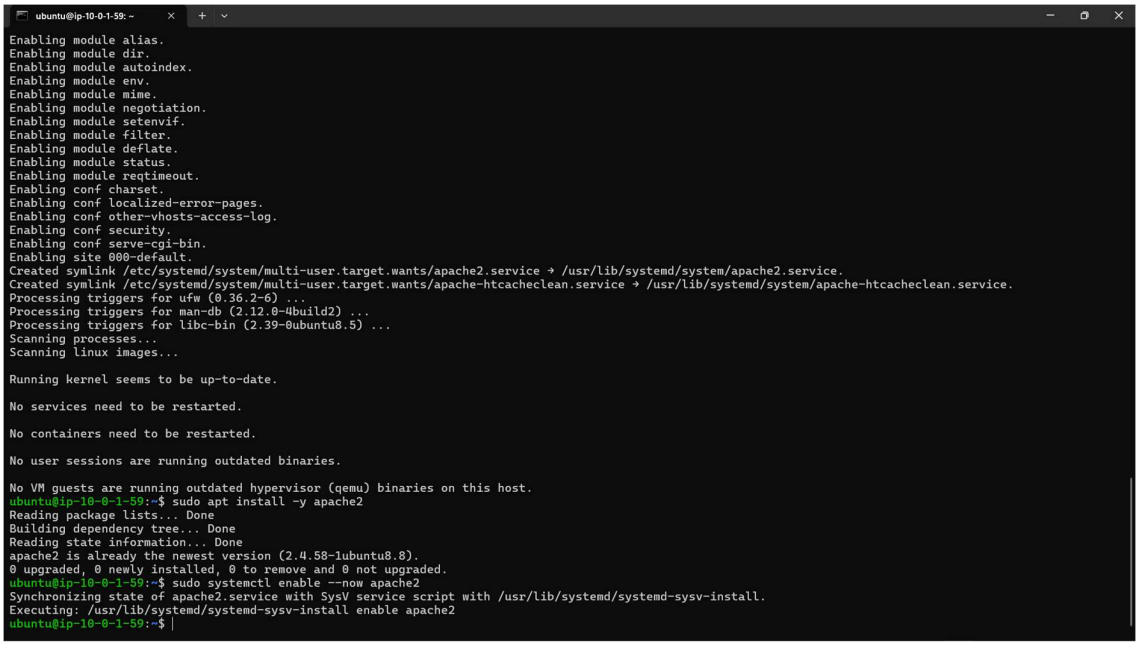
sudo apt install -y apache2

Shaikh Ateeb Ahmed
07-08-2025



enable and start Apache

sudo systemctl enable --now apache2



From your laptop/browser: open http://<PUBLIC_IP>/ — you should see the default Apache2 Ubuntu page.

Shaikh Ateeb Ahmed
07-08-2025

Create an AMI of that instance

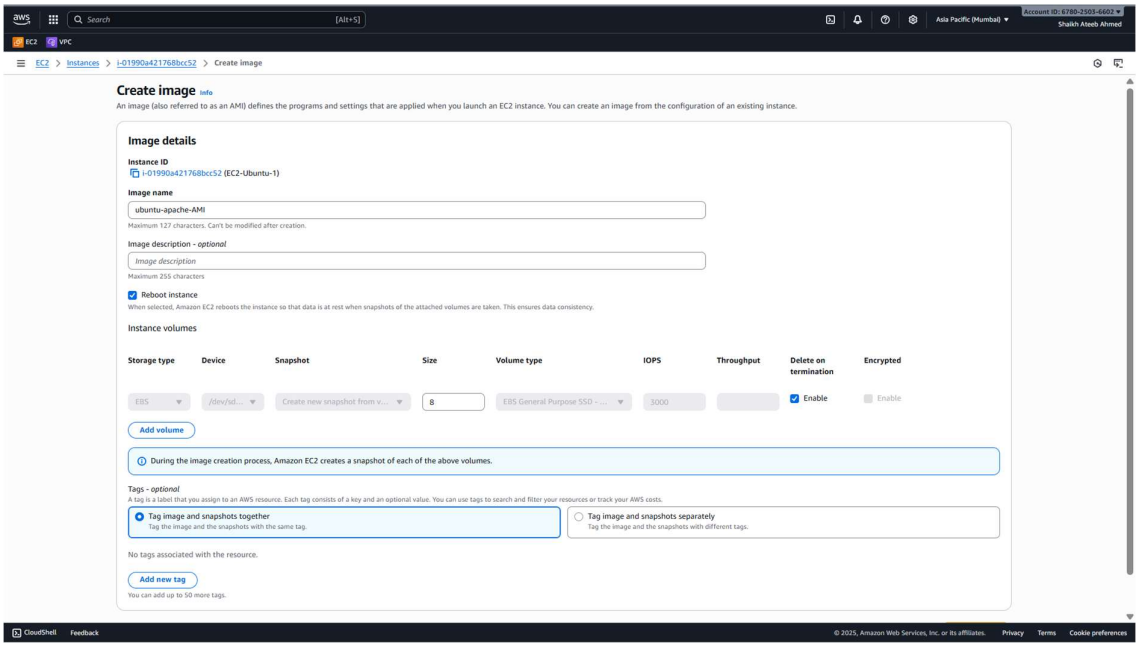
Console (simplest)

1. EC2 → Instances → select the Ubuntu instance that u have create above.
2. **Actions → Image and templates → Create image**.

   Name: e.g. ubuntu-apache-AMI

   Leave volumes as-is (root volume included).
3. Click **Create image**.

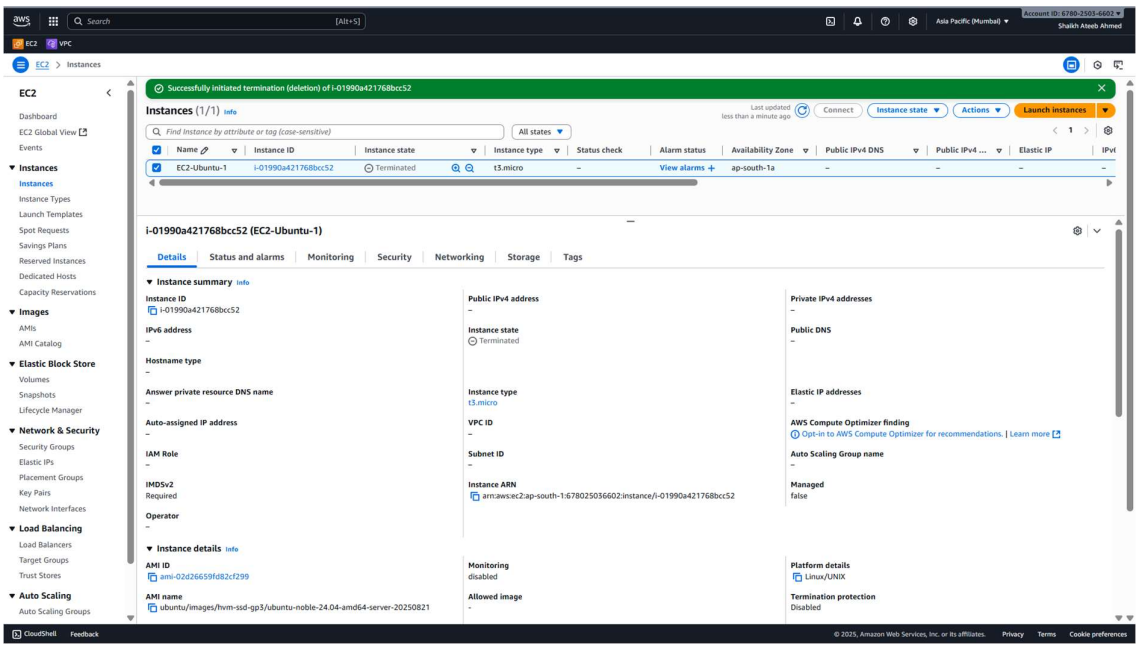
Shaikh Ateeb Ahmed
07-08-2025



4. EC2 → Images → AMIs → wait until the AMI state becomes **available**.

Terminate the original instance (Console)

1. EC2 → Instances → select the instance → **Instance state → Terminate instance** → confirm.
2. Wait for state to change to **terminated**.

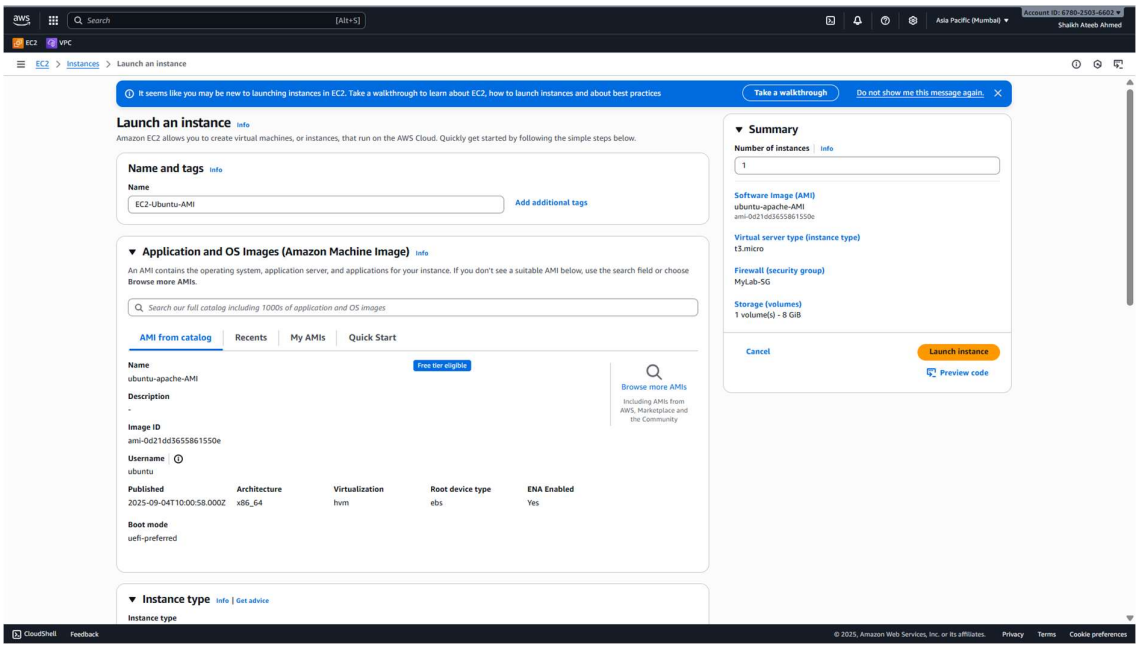


Terminating is permanent for that instance ID; AMI remains available in **Images → AMIs**.

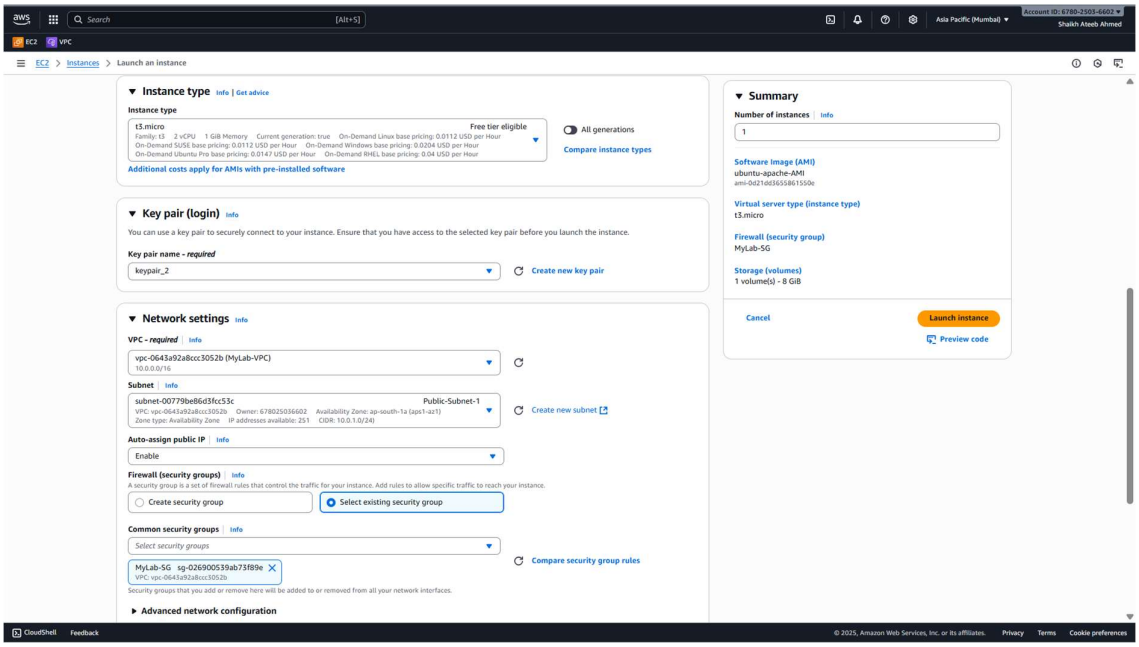
Shaikh Ateeb Ahmed
07-08-2025

Launch new instance from your AMI

1. EC2 → **Launch instances**.
2. Under **My AMIs** (left) → select the AMI you created.
3. Give name eg. EC2-Ubuntu-AMI
4. Set *Instance type* (e.g. t3.micro).
5. Configure network/subnet: choose as before (public subnet + auto-assign public IP if you want browser access).
6. Select or create Security Group (must allow SSH and HTTP).
7. Choose key pair (same or different).
8. Launch. EC2 will create instances from the same AMI (It will have Apache already installed and your default page present).



Shaikh Ateeb Ahmed
07-08-2025



Then test in browser: http://<new-instance-public-ip>/ — you should see the same Apache default page.