Azad Chauhan

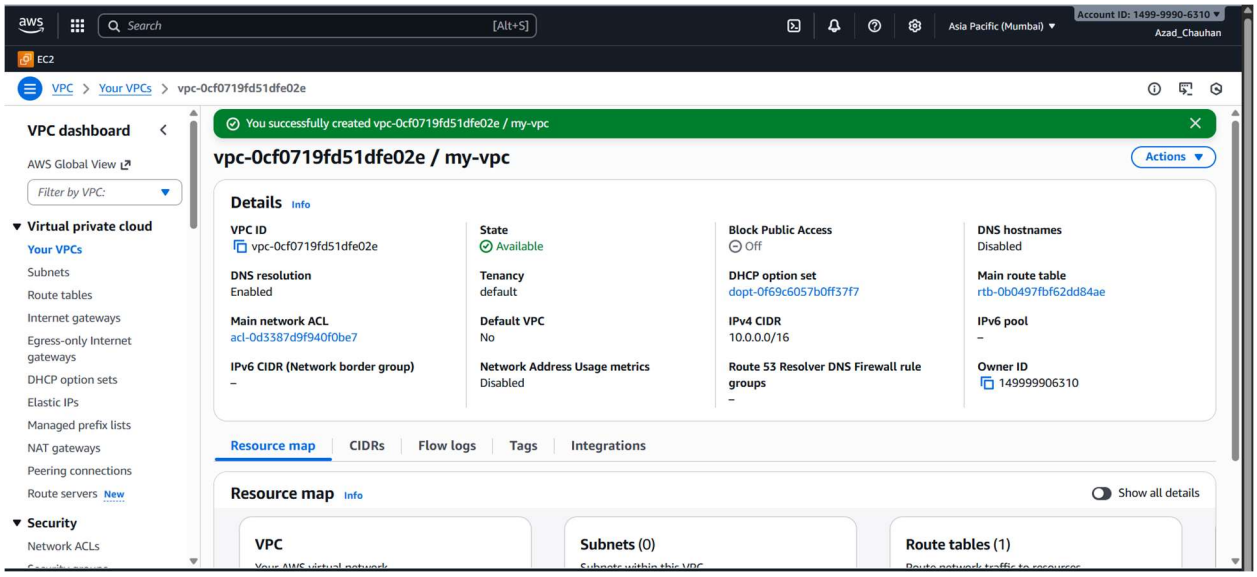
VPC with Public & Private Subnets + Bastion Host Setup

&

Private EC2 Access via Bastion Host (Jump Server)

STEP 1: Create VPC

1. AWS Console → **VPC**
2. **Create VPC**
3. Choose: **VPC Only**
4. **Name:** my-vpc
5. **IPv4 CIDR:** 10.0.0.0/16
6. **Tenancy :** Default
7. Create



STEP 2: Create Subnets

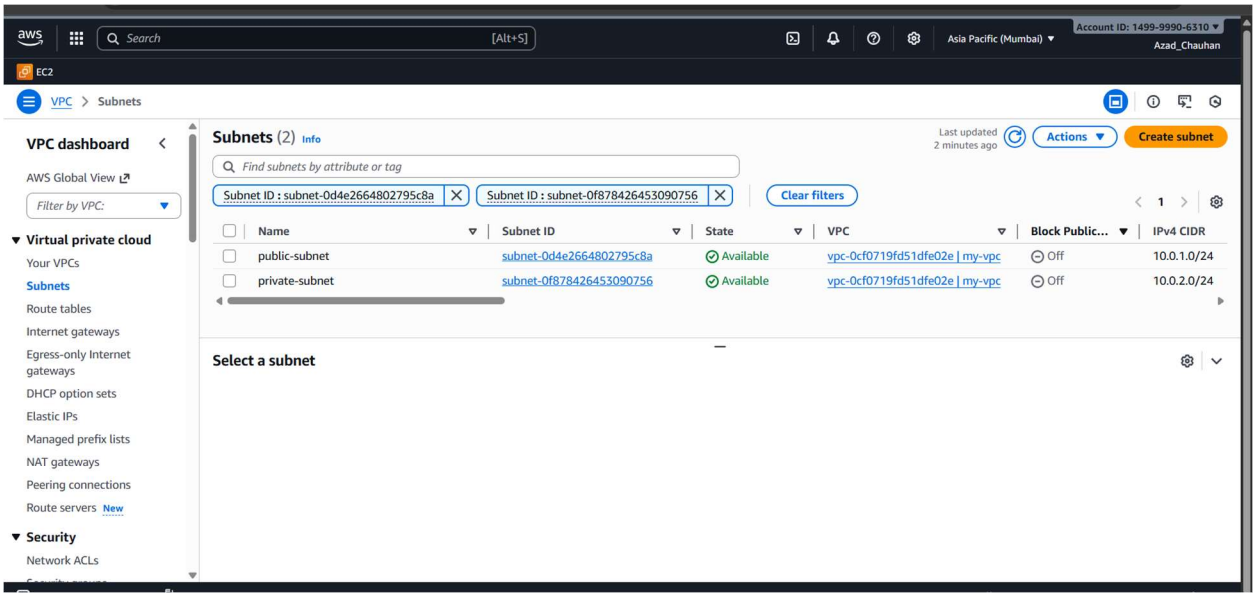
👉 Public Subnet

1. Subnets → Create subnet

2. **Name:** public-subnet
3. **VPC:** my-vpc
4. **CIDR:** 10.0.1.0/24
5. **Availability Zone:** ap-south-1a
6. **Enable Auto-assign Public IP:** Turn ON

👉 Private Subnet

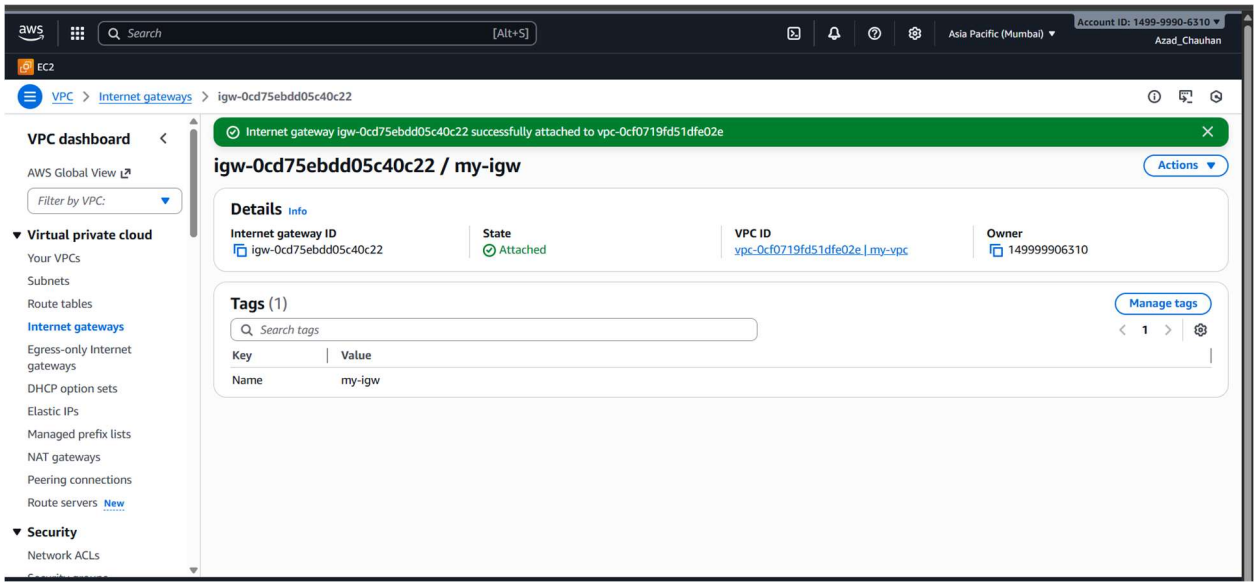
1. Subnets → Create subnet
2. **Name:** private-subnet
3. **CIDR:** 10.0.2.0/24
4. **AZ:** ap-south-1a
5. **Auto-assign Public IP:** OFF



STEP 3: Create & Attach Internet Gateway (IGW)

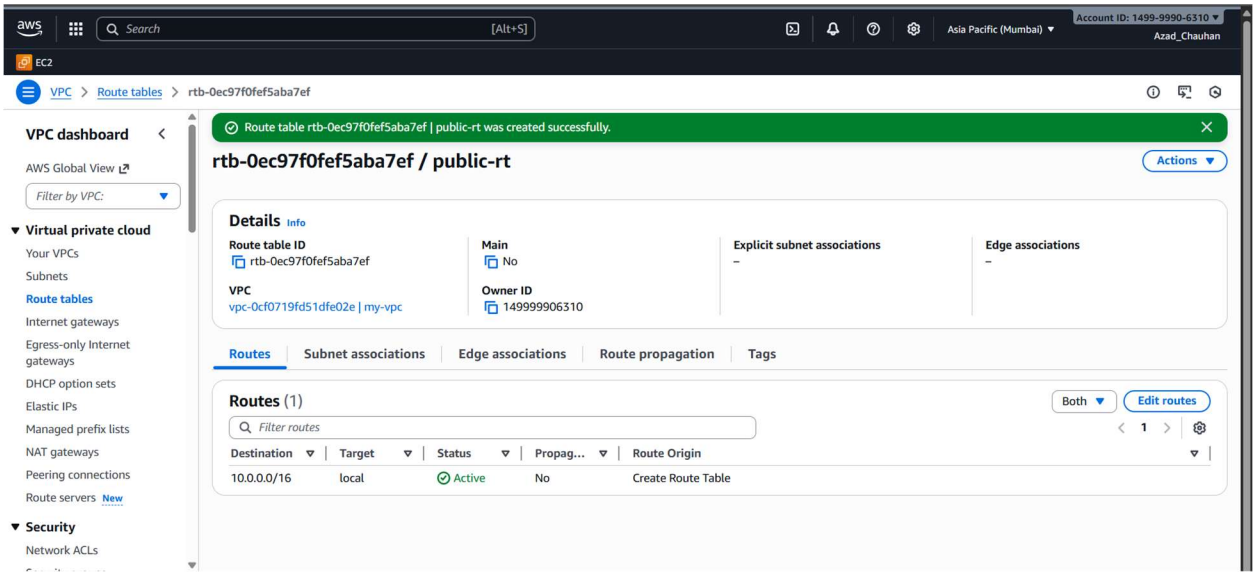
1. Internet Gateways → Create
2. Name: my-igw

3. **Attach to VPC:** my-vpc

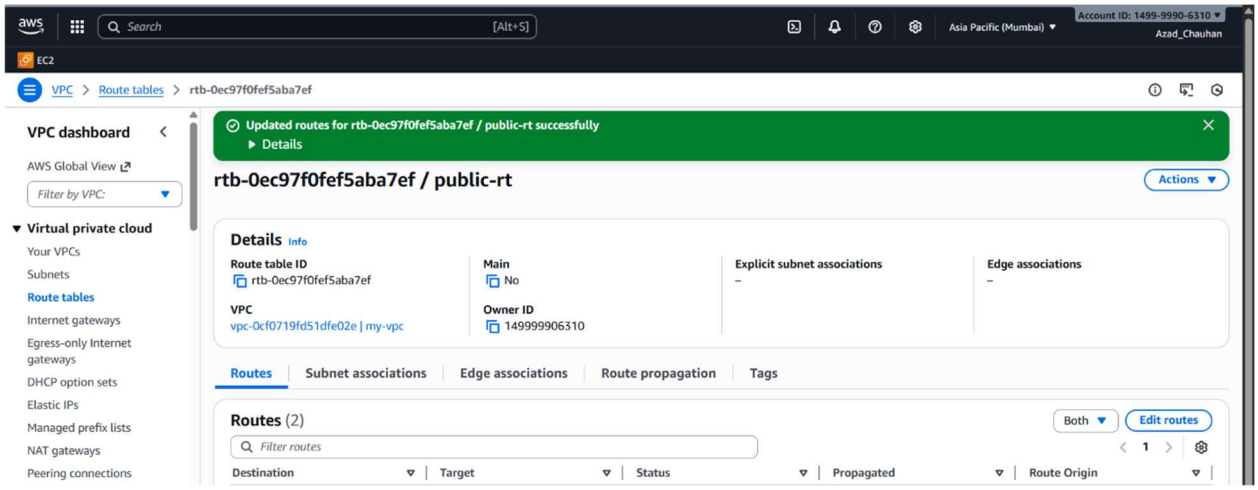


STEP 4: Route Table for Public Subnet

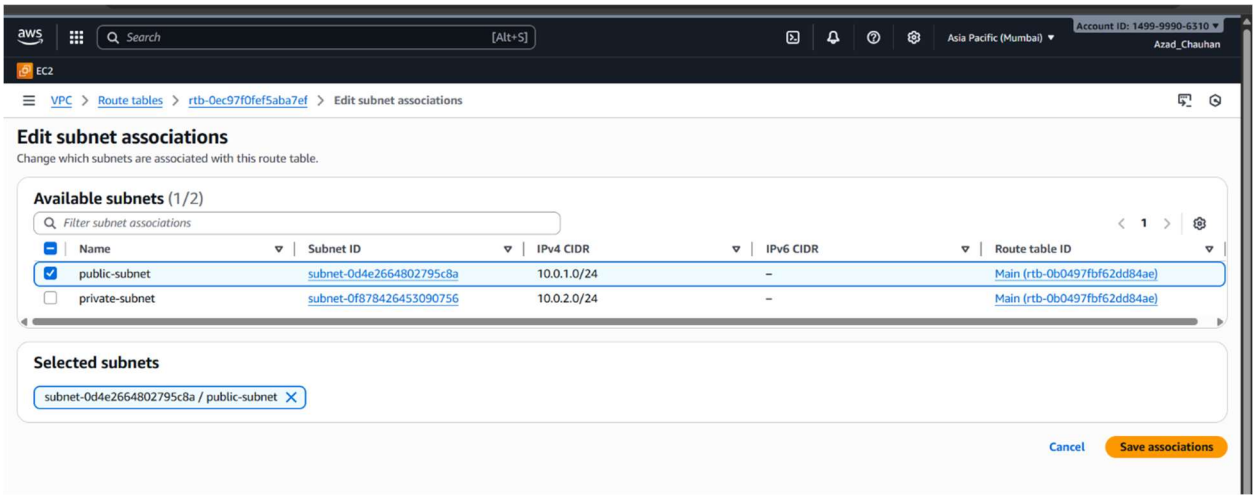
1. Route Tables → Create
2. Name: public-rt
3. VPC: my-vpc
4. Create



5. Edit routes:
    - Destination: 0.0.0.0/0
    - Target: **Internet Gateway (my-igw)**
    - **Save Changes**



6. Subnet associations:
    - Select **public-subnet**



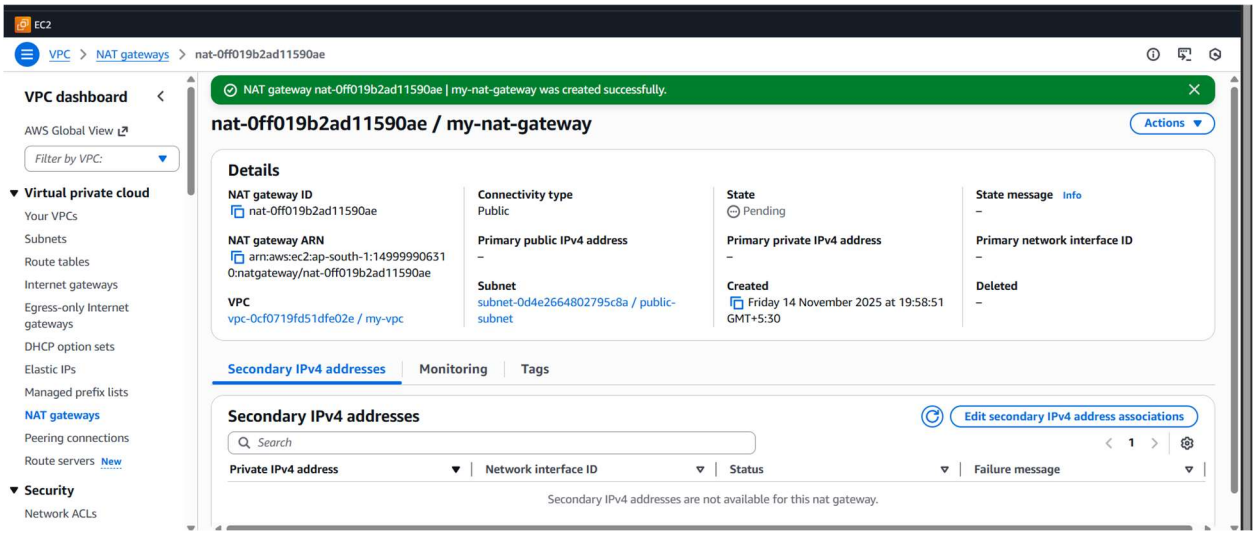
⭐ **Public subnet now has internet access.**

Azad Chauhan

STEP 5: NAT Gateway for Private Subnet

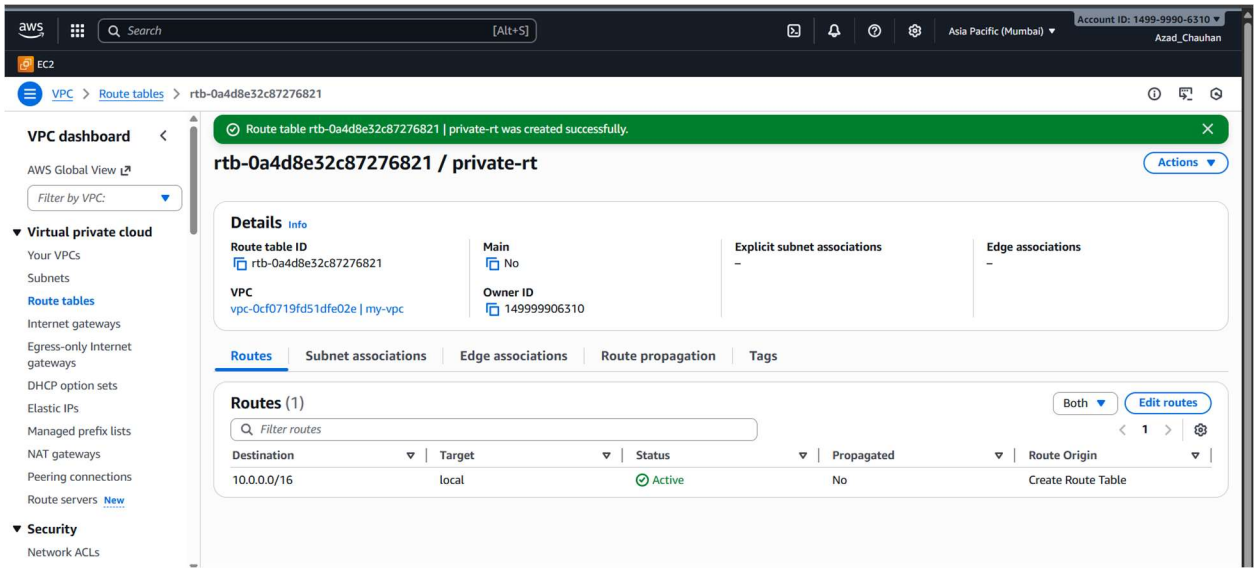
Private subnet ko internet dene ke liye NAT Gateway use hota hai.

1. NAT Gateway → Create
2. Subnet: **public-subnet**
3. Elastic IP: Allocate + attach
4. Create

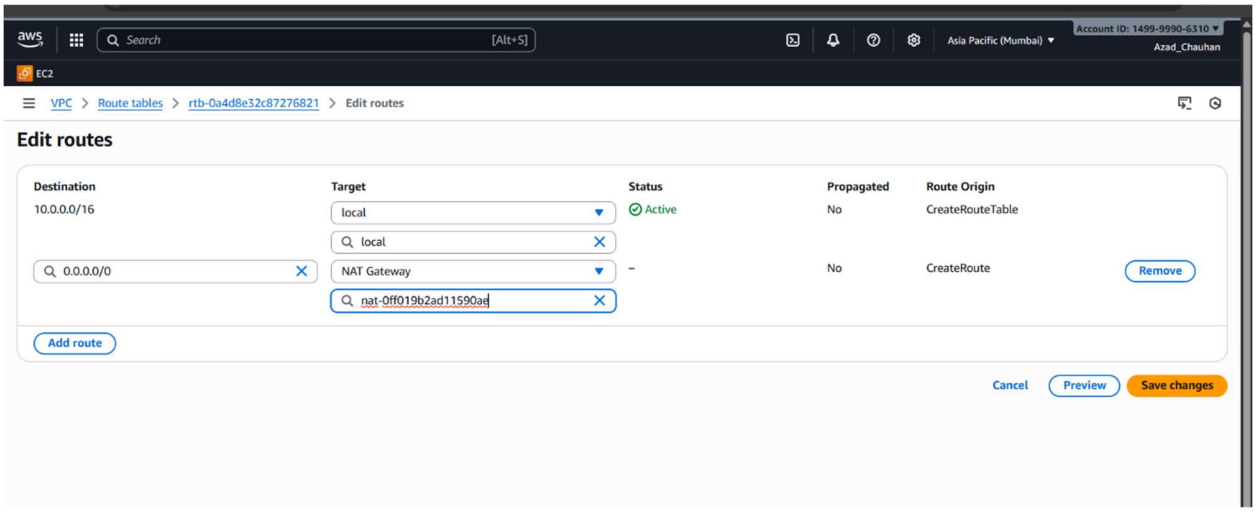


STEP 6: Route Table for Private Subnet

1. Route Tables → Create
2. Name: private-rt
3. VPC: my-vpc
4. Create

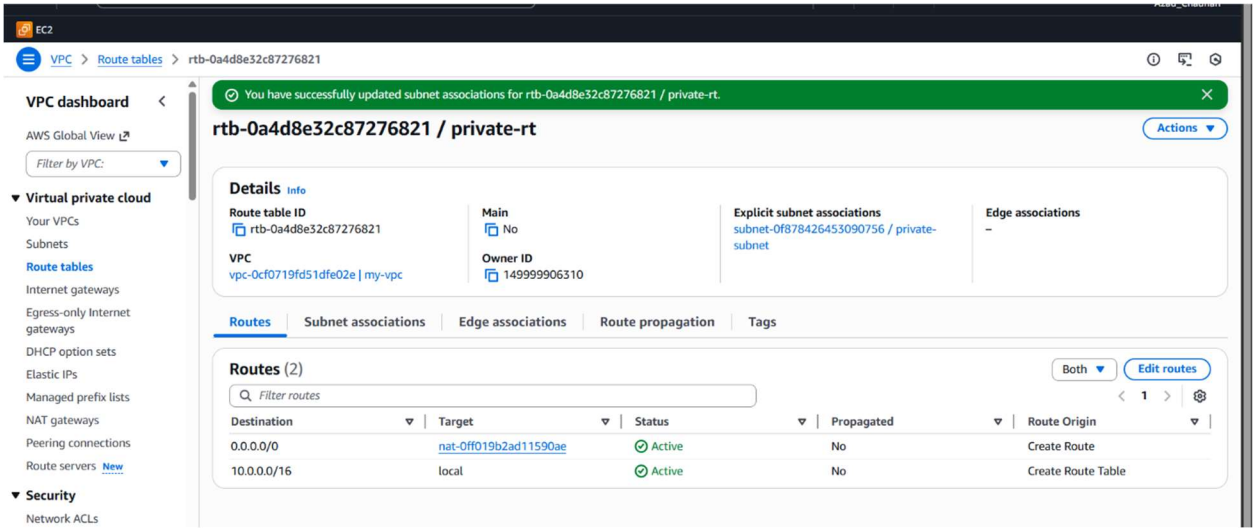


5. Edit routes:
    - Destination: 0.0.0.0/0
    - Target: **NAT Gateway**



6. Subnet Association:
    - Select **private-subnet**

Azad Chauhan



⭐ Ab private subnet ke resources **internet access le sakte hain**, but inbound internet se hidden rahenge.

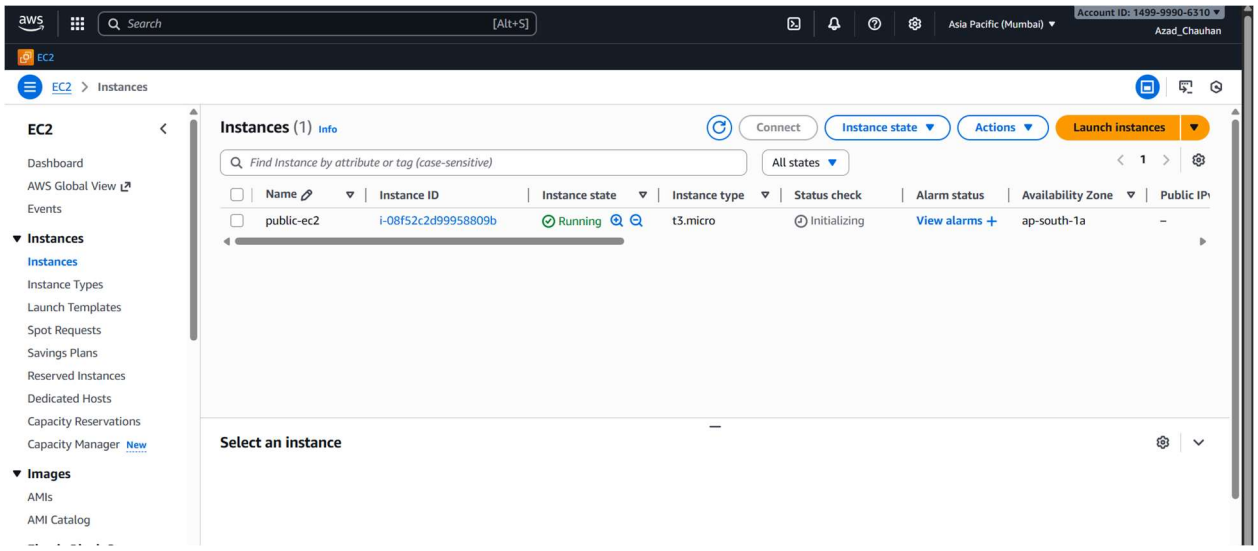
Azad Chauhan

sMETHOD: Connect Private EC2 Using Public EC2 (Bastion Host)

(Requires 2 EC2 instances: 1 Public + 1 Private)

✅ STEP 1 — Launch Public EC2

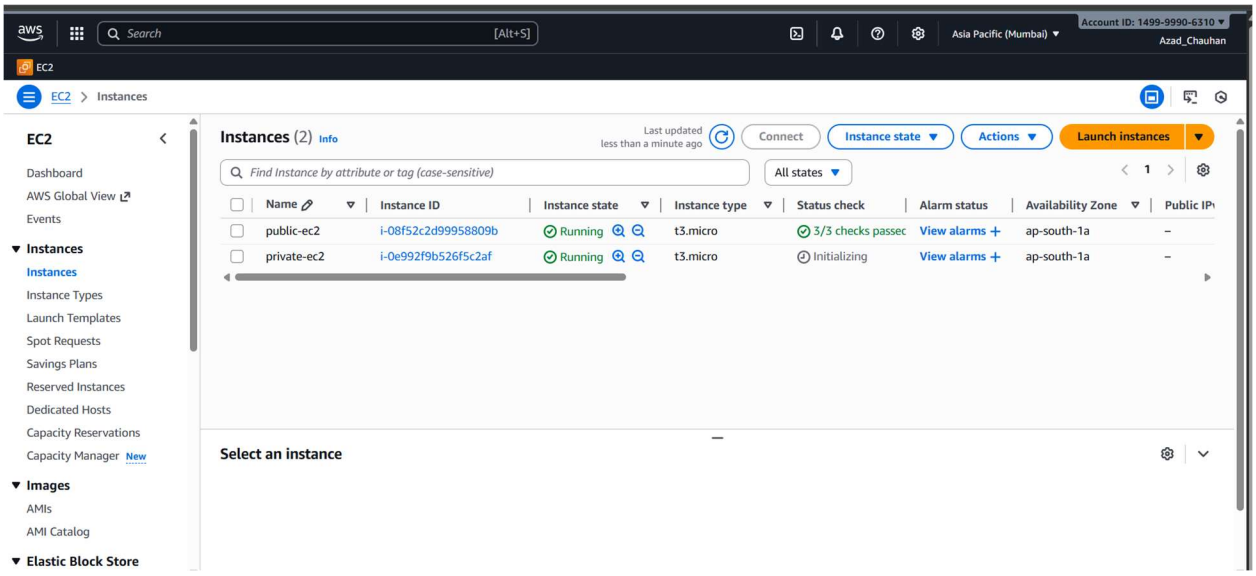
1. **Go to EC2 → Launch instance**
2. **Name: public-ec2**
3. **Choose AMI: Ubuntu**
4. **Network settings:**
    - **VPC: *your VPC***
    - **Subnet: public-subnet**
    - **Auto-assign Public IP: Enabled**
5. **Security Group:**
    - **Allow SSH (22) from My IP**
6. **Select / Create Key Pair**
7. **Launch instance**



STEP 2 — Launch Private EC2

Azad Chauhan

1. **EC2 → Launch instance**
2. **Name: private-ec2**
3. **AMI: Ubuntu**
4. **Network settings:**
    - **VPC: same**
    - **Subnet: private-subnet**
    - **Auto-assign Public IP: Disabled**
5. **Security Group:**
    - **Allow SSH (22)**
    - **Source: Public EC2's security group (important)**
6. **Launch instance**



STEP 3 — Connect to Public EC2

Open your terminal and run:

scp -i "john.pem" john.pem ubuntu@<PUBLIC_EC2_IP>:/home/ubuntu/

(This command securely copies the john.pem key file from your local machine to the /home/ubuntu/ directory on the public EC2 instance using the provided SSH key.)

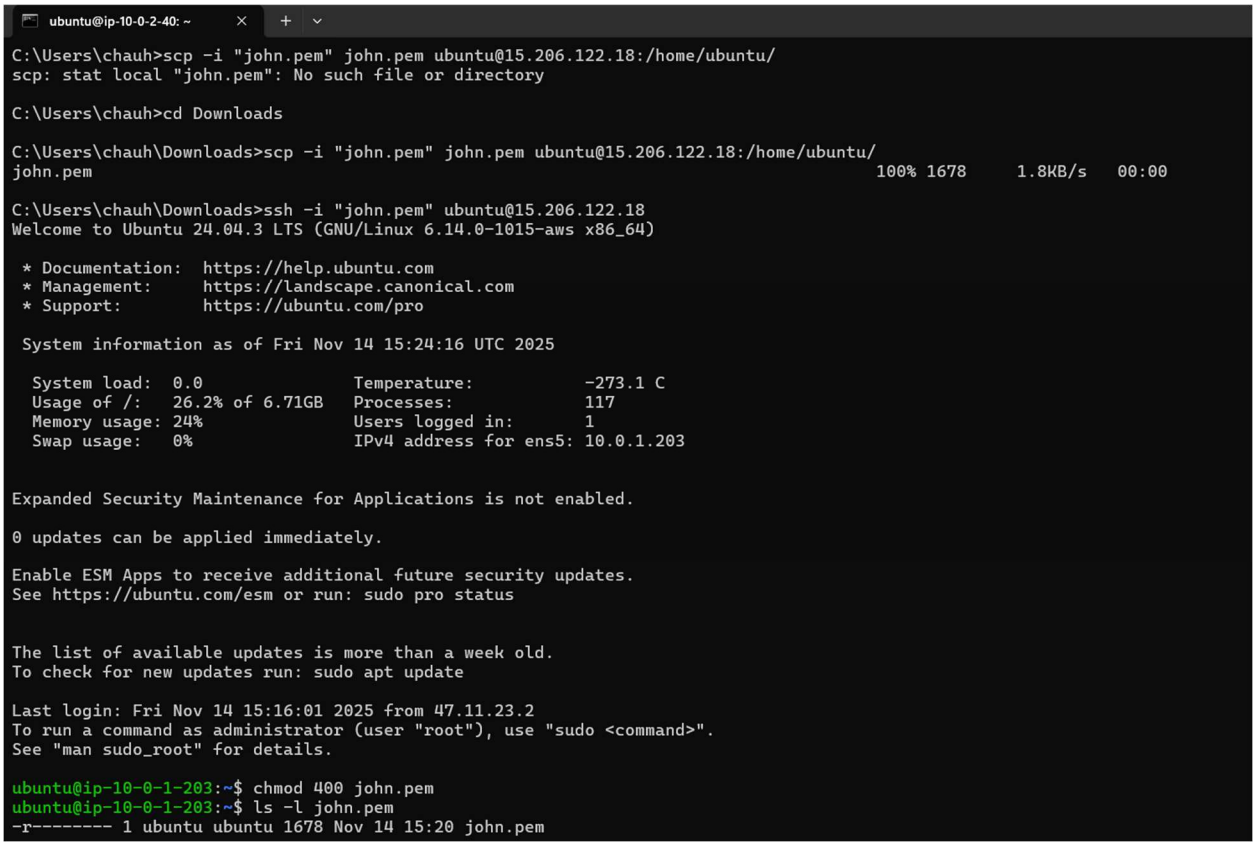
Azad Chauhan

```
C:\Users\chauh>cd Downloads

C:\Users\chauh\Downloads>scp -i "john.pem" john.pem ubuntu@15.206.122.18:/home/ubuntu/
john.pem                                                                      100% 1678     1.8KB/s   00:00
```

After this command you can connect your EC2 via ssh

ssh -i “john.pem” ubuntu@15.206.122.18



chmod 400 john.pem

(This command changes the file permissions of john.pem so that only the owner can read it.
SSH requires private key files to have strict permissions for security.)

ls -l john.pem

(This command lists the file details and shows the new permissions.
You can see -r-------- which means read-only for the owner.)

Azad Chauhan

STEP 4 — Find Private EC2 Private IP

EC2 → Instances → private-ec2 → Private IPv4 address

Example:
10.0.2.40

Run SSH command from inside Public EC2 terminal

Inside your public EC2, run: