Configure Private App-to-DB Connectivity Across Two VPCs (Using NAT + VPC Peering)

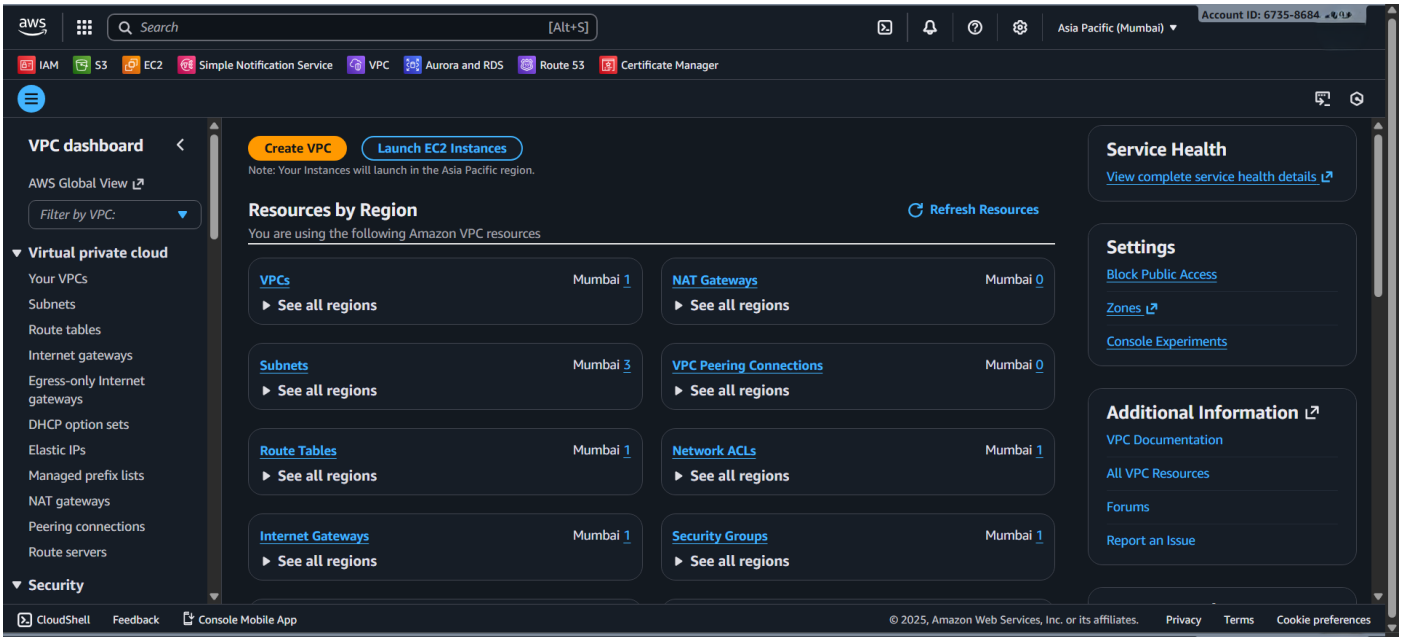
By: **Nikhilesh Sakhare**

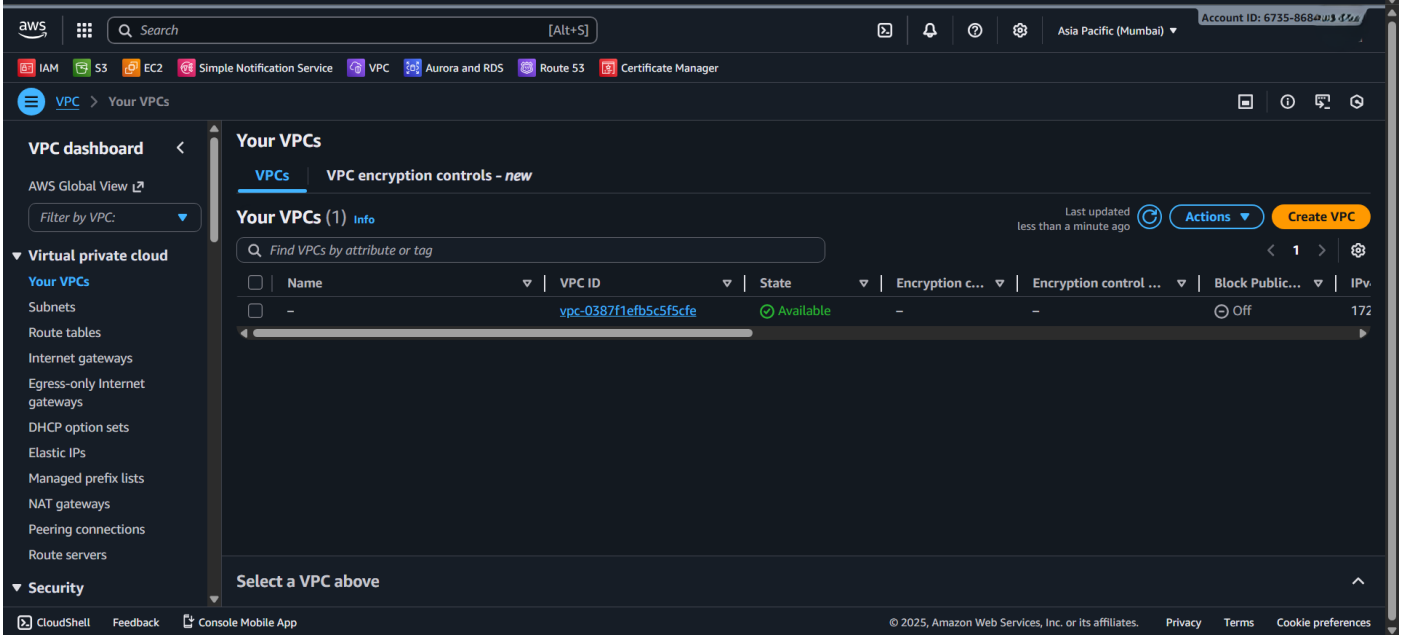
Prerequisites

- AWS account access to VPC and EC2 services.
- Region used: **Asia Pacific (Mumbai)**.
- Key pair available: **sample-2-ind.pem**.

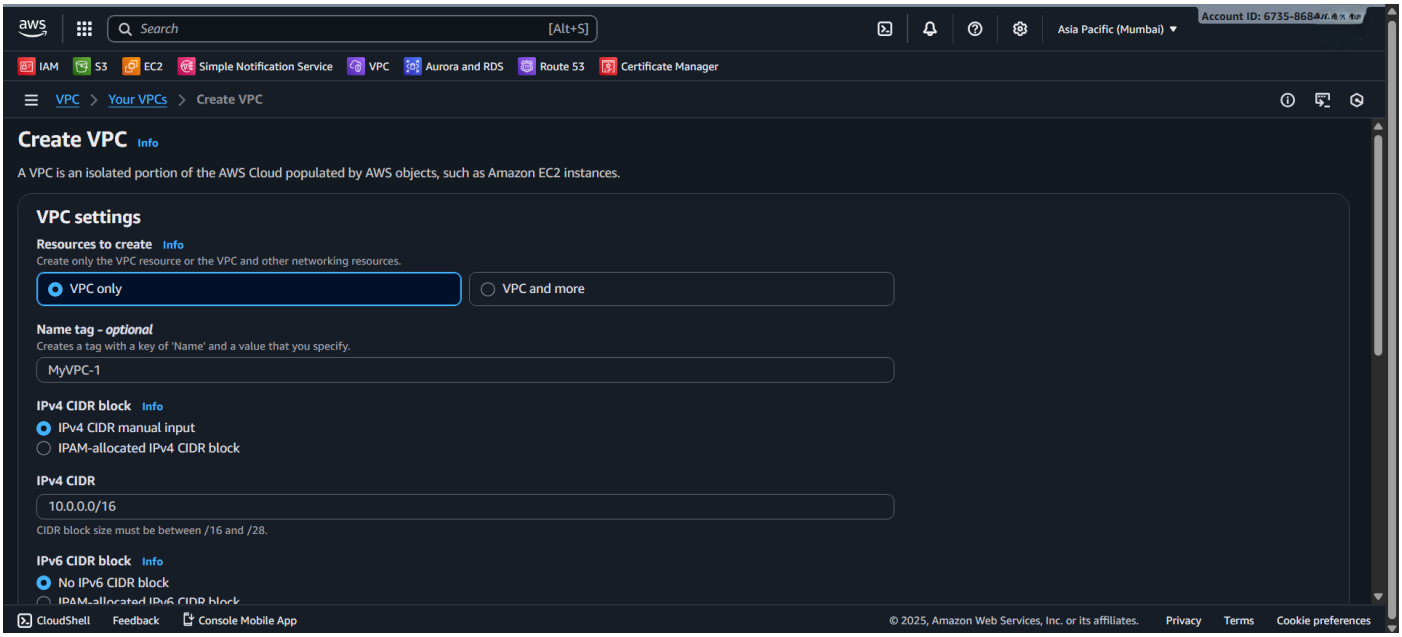
Step 1 — Create VPC-1 (MyVPC-1)

1. Open **VPC → Your VPCs → Create VPC** and select **VPC only**.

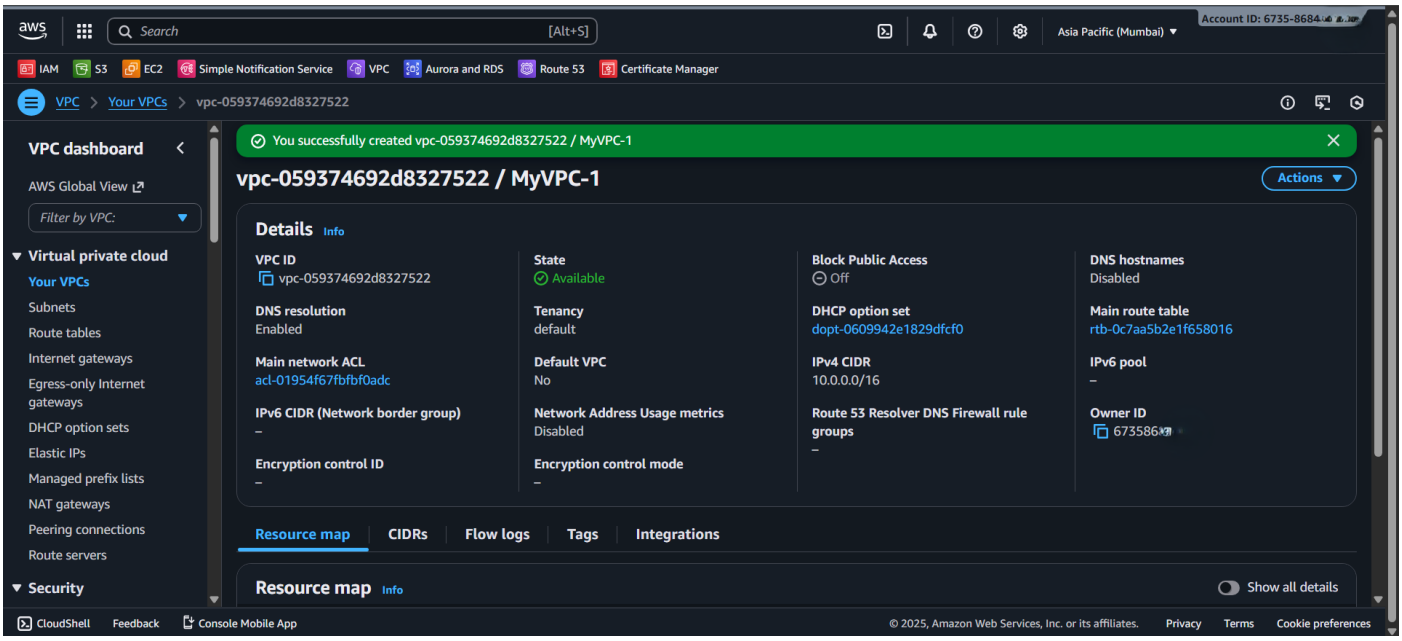




2. Set Name: **MyVPC-1** and IPv4 CIDR: **10.0.0.0/16**.

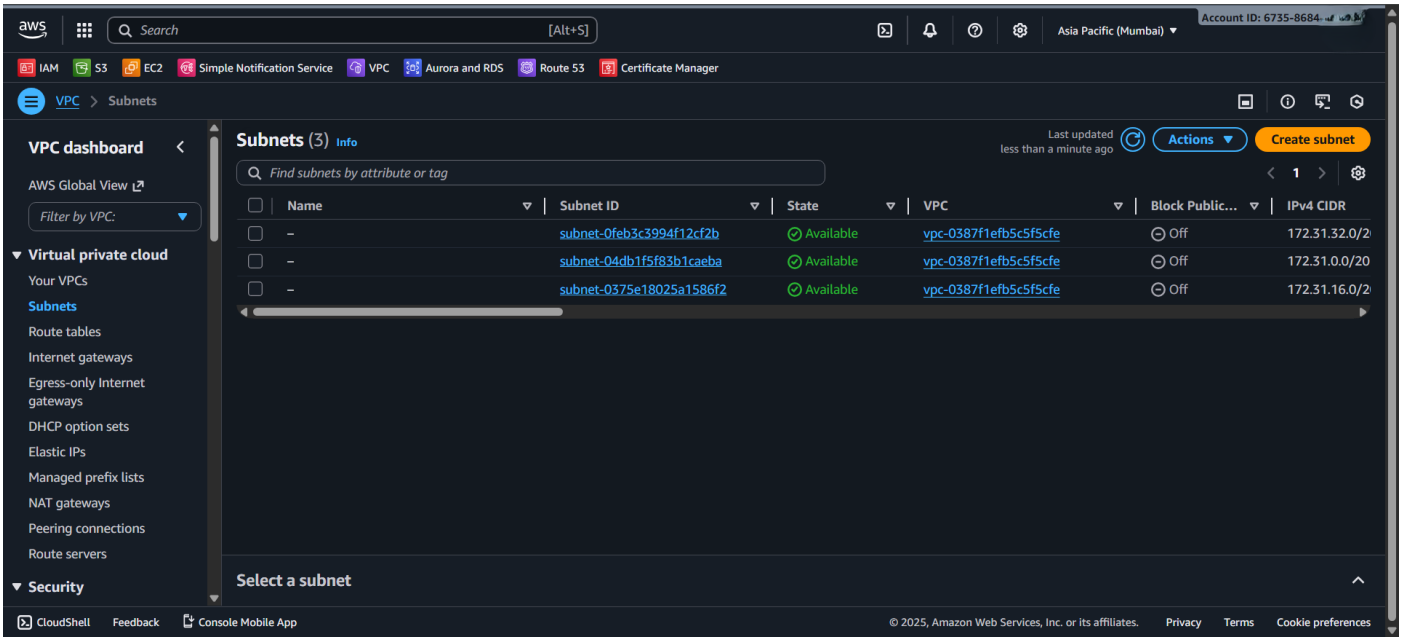


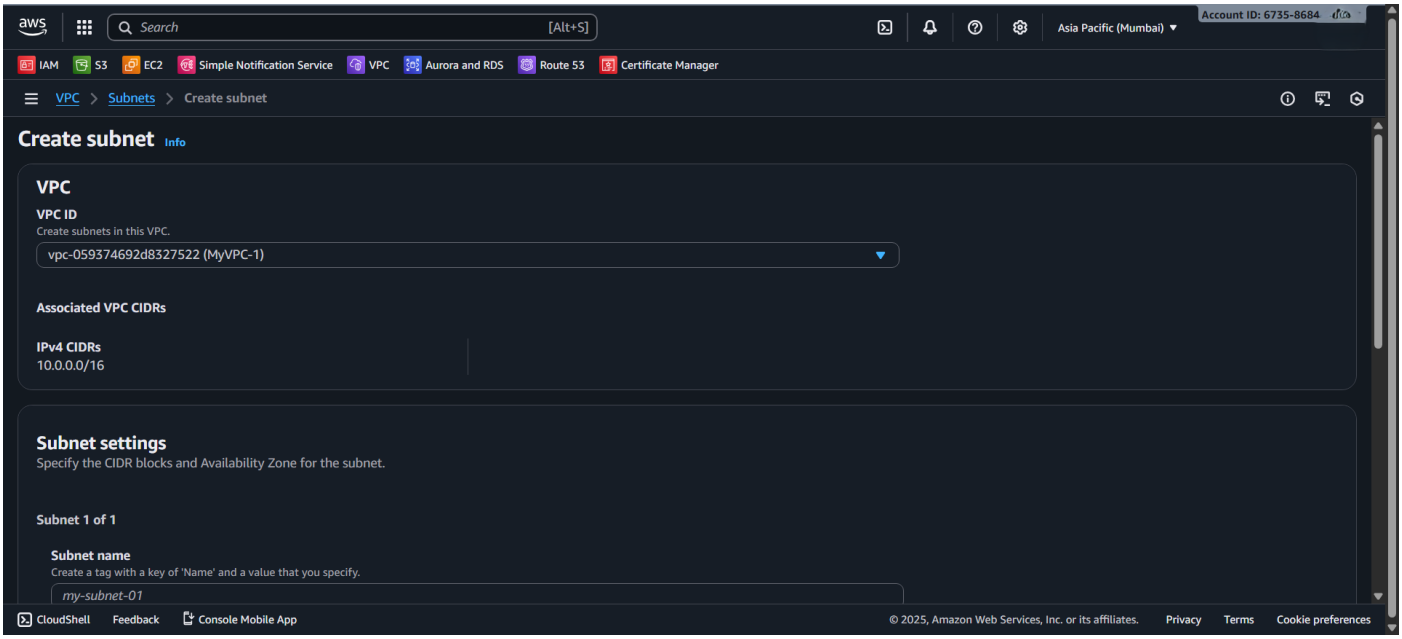
3. Create the VPC and verify VPC ID is created.



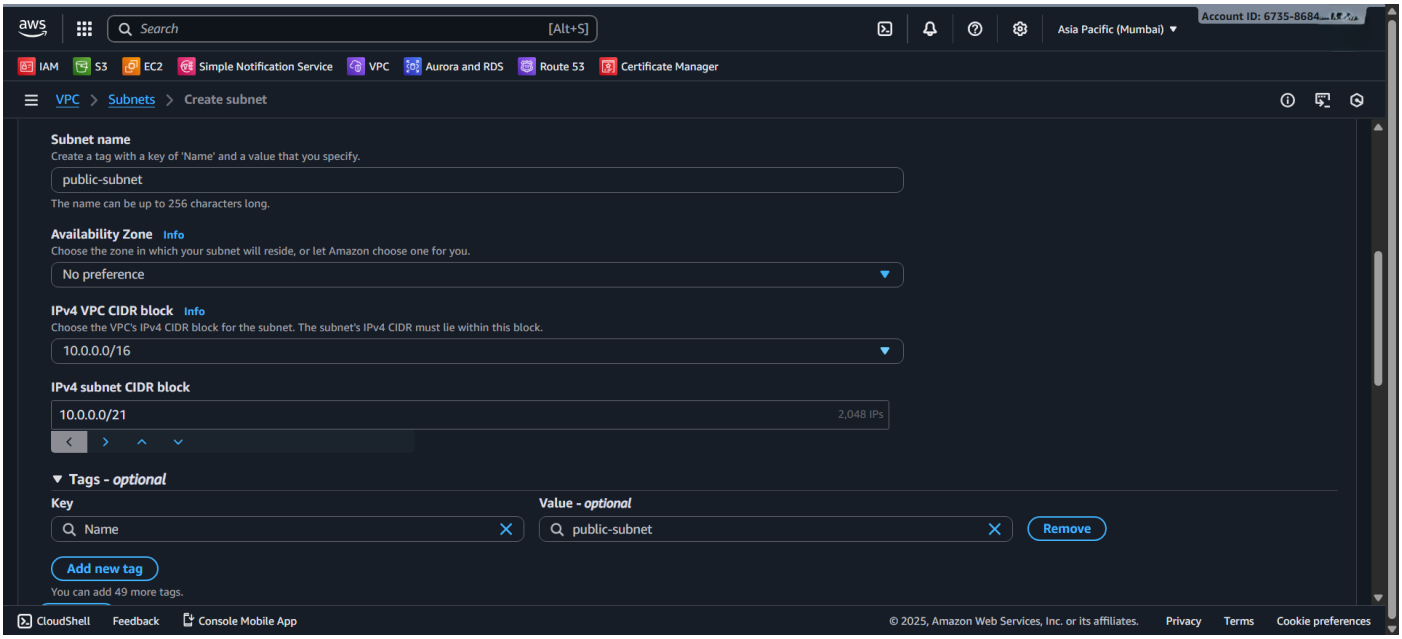
Step 2 — Create public and private subnets in VPC-1

1. Go to **VPC → Subnets → Create subnet** and choose VPC **MyVPC-1**.

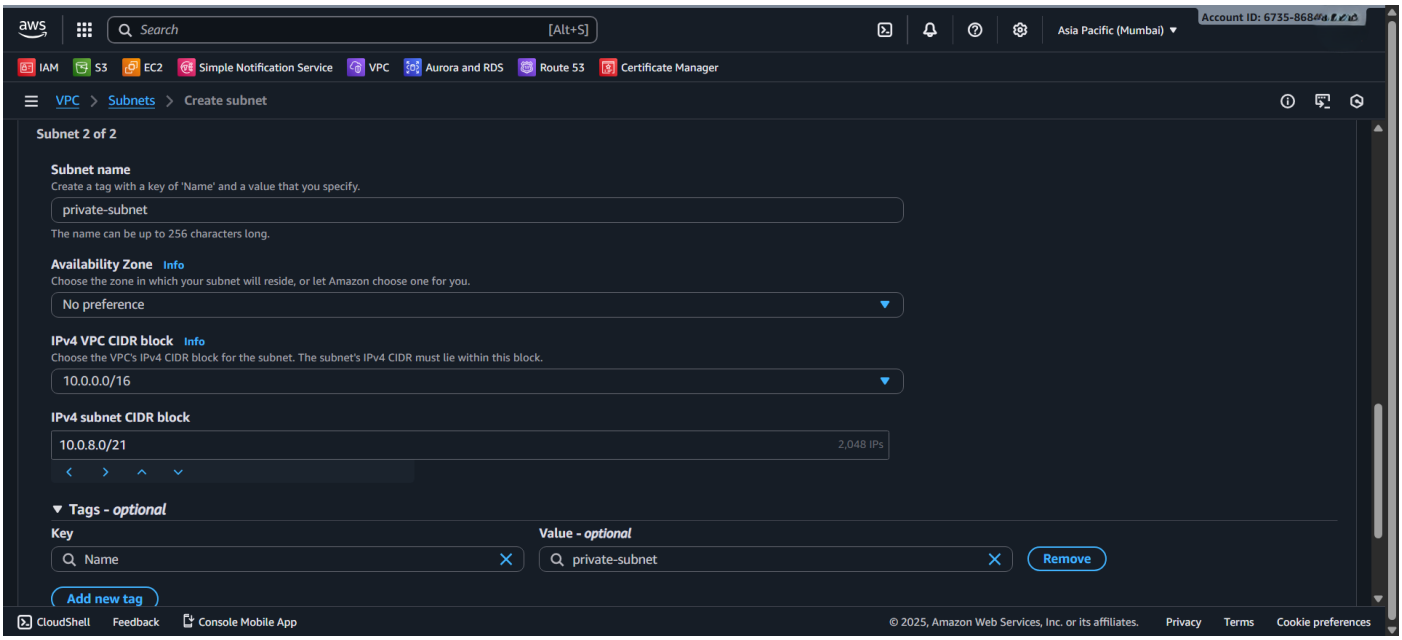




2. Create **public-subnet** with CIDR **10.0.0.0/21**.



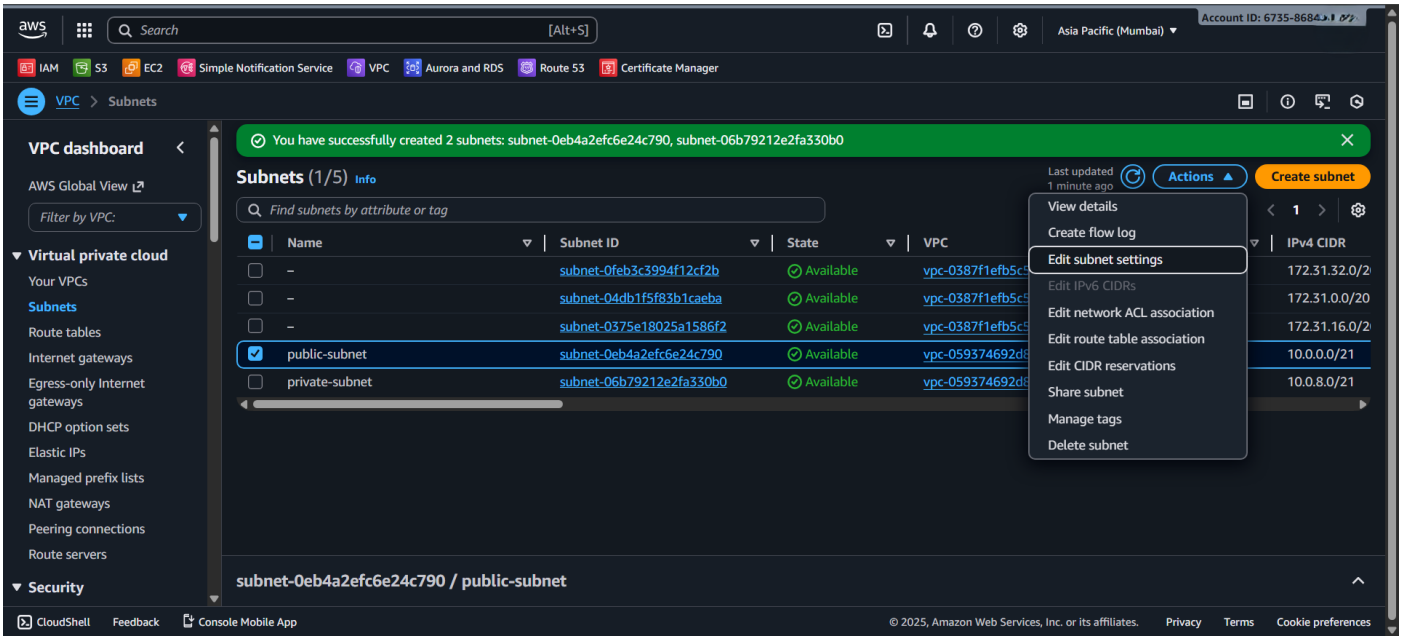
3. Create **private-subnet** with CIDR **10.0.8.0/21**.



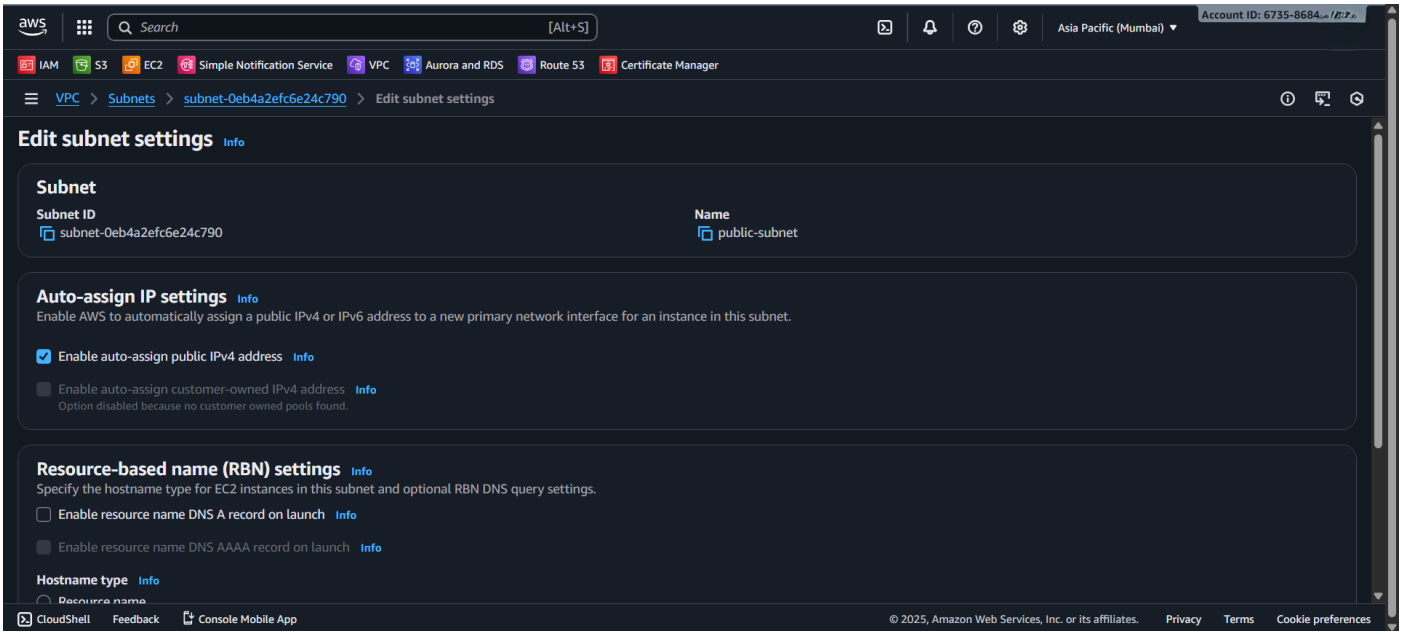
4. Verify both subnets exist in the subnet list.

Step 3 — Enable public IP auto-assign for the public subnet

1. Select **public-subnet → Actions → Edit subnet settings**.

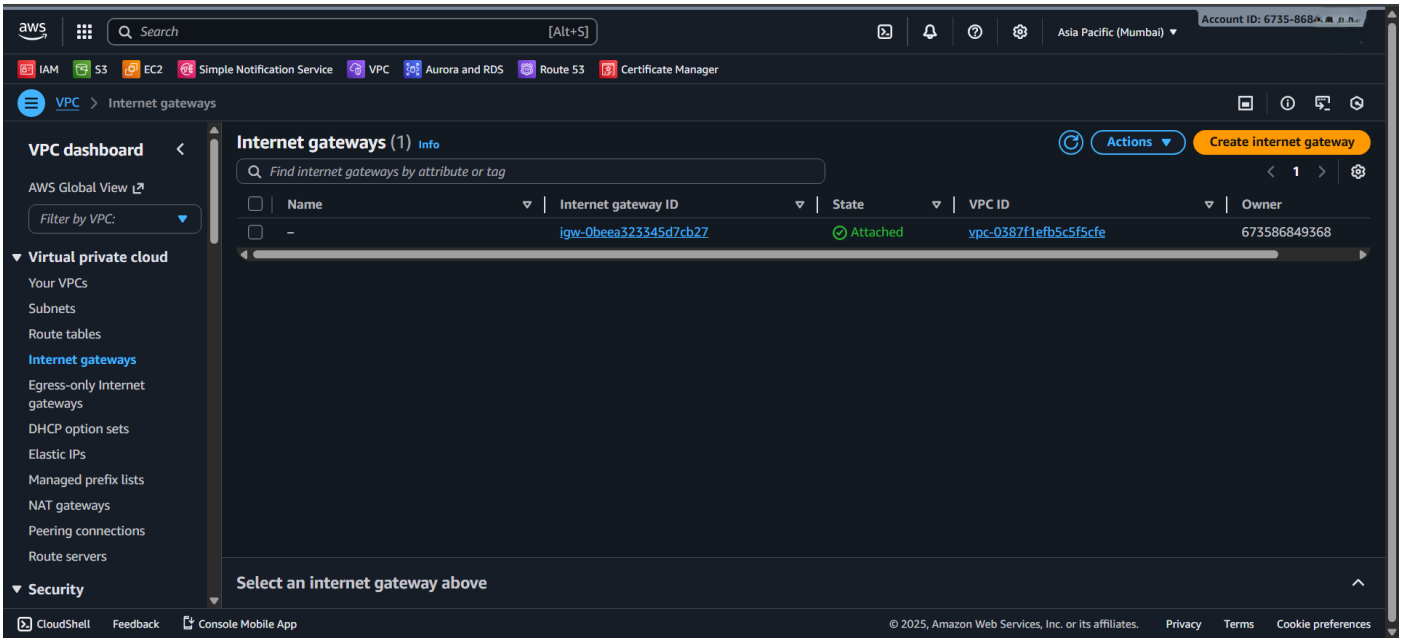


2. Enable **Auto-assign public IPv4 address** and save.

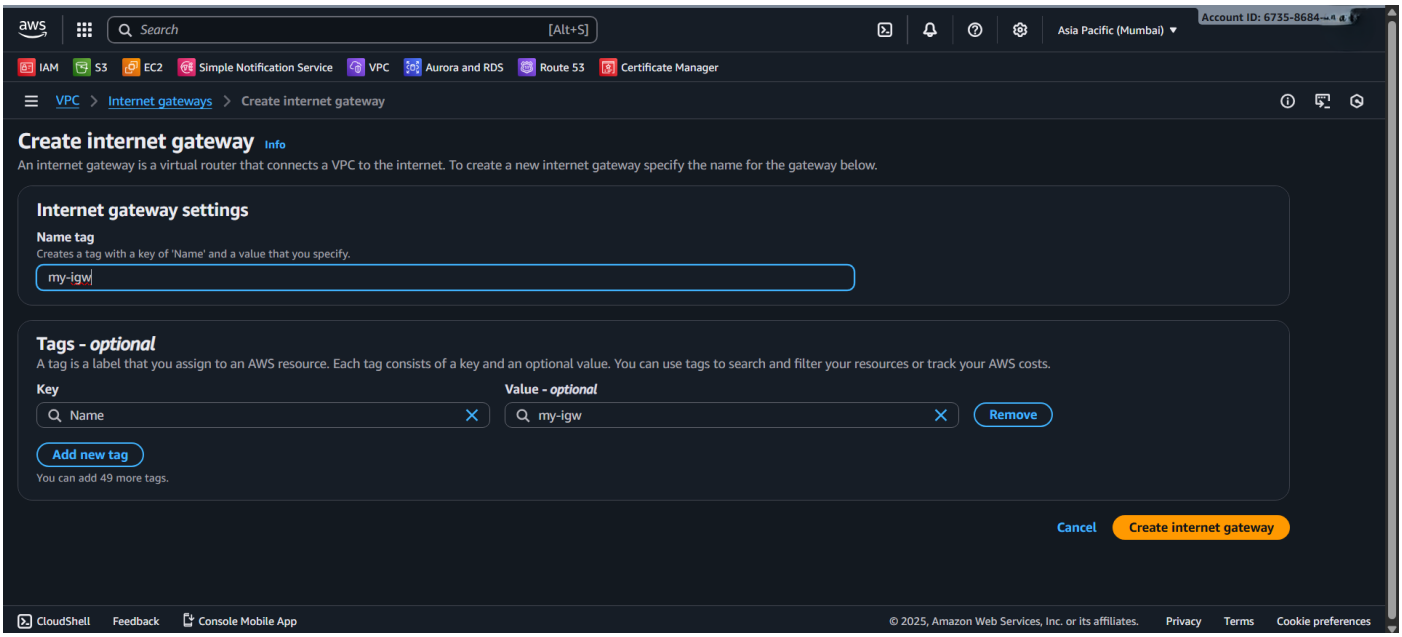


Step 4 — Create and attach Internet Gateway (my-igw) to VPC-1

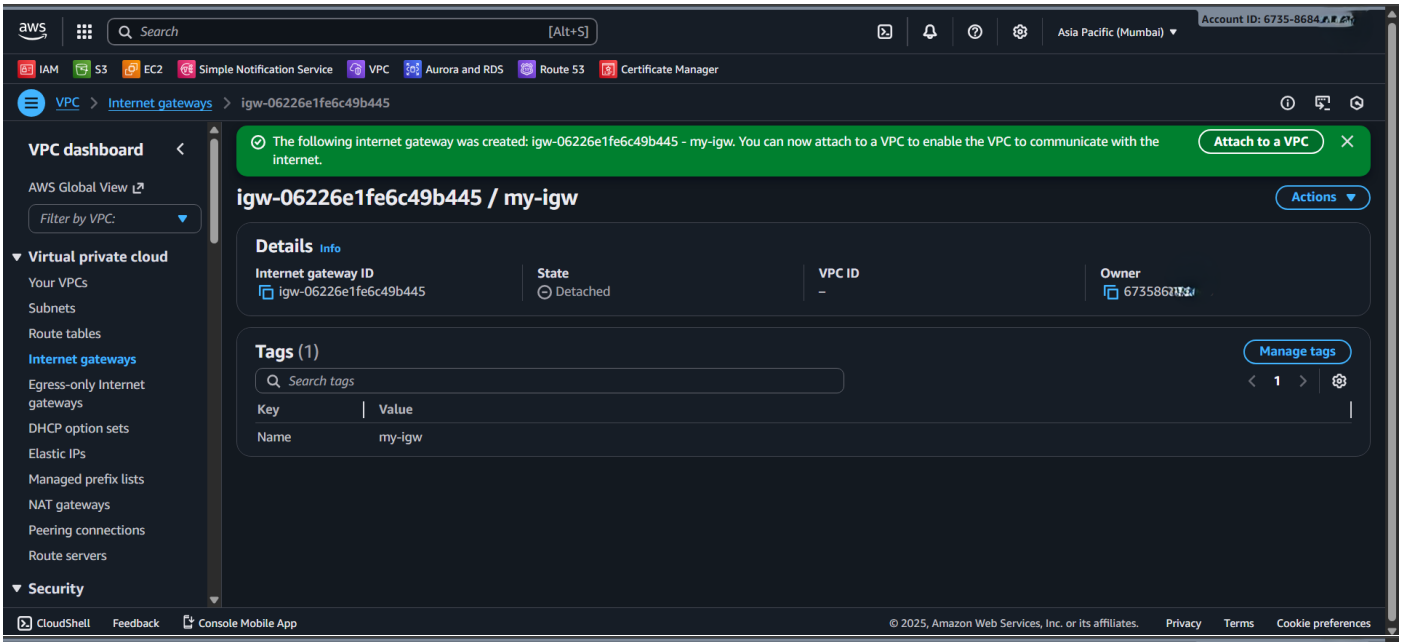
1. Go to **VPC → Internet gateways → Create internet gateway**.

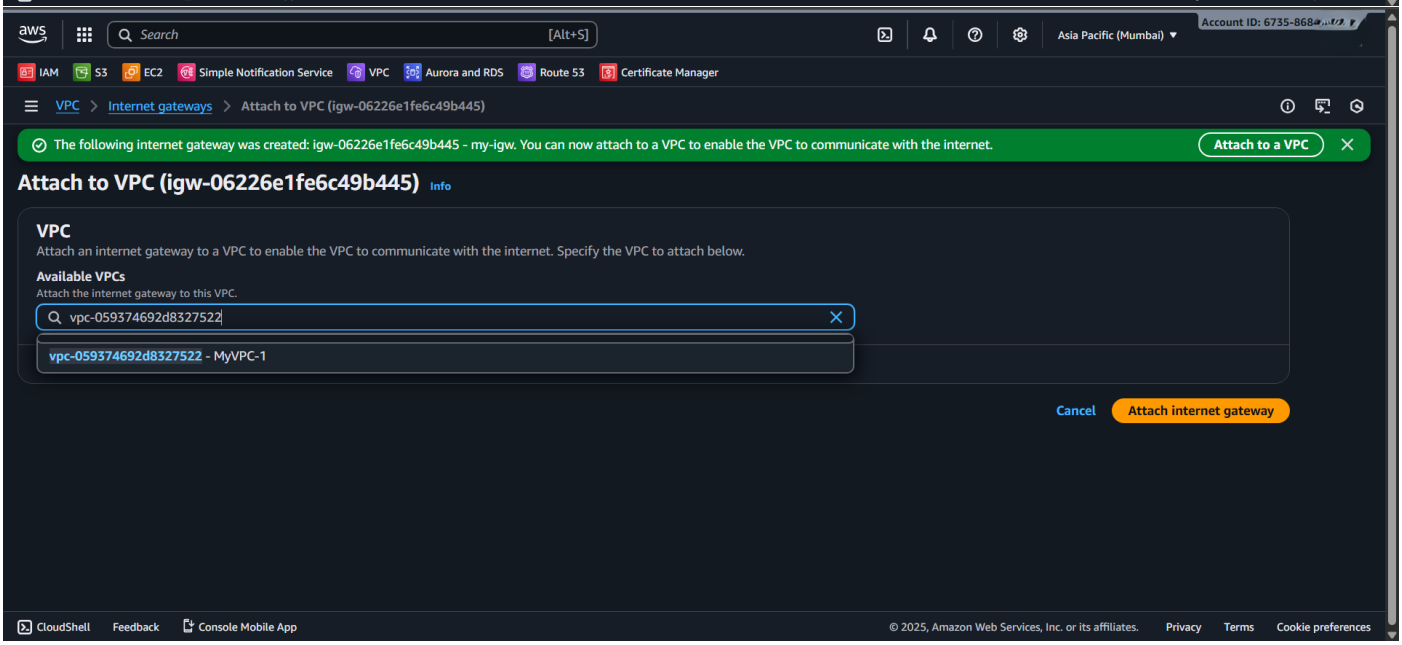


2. Name it **my-igw** and create.



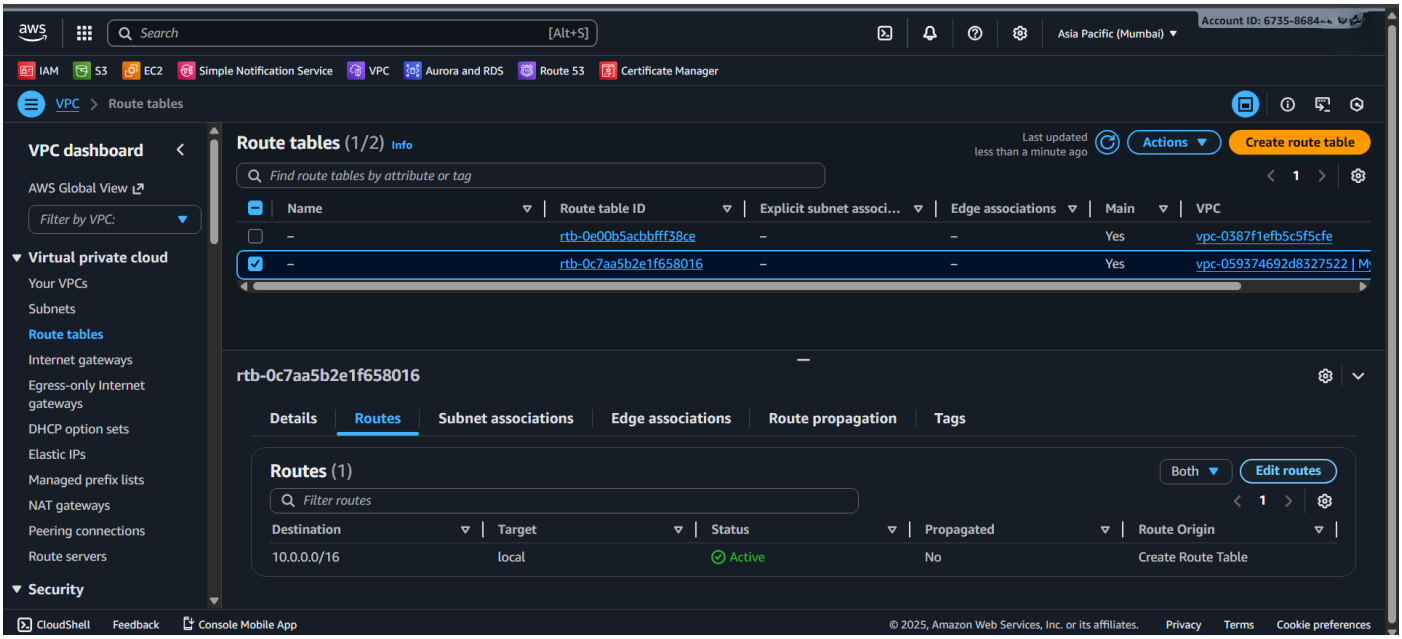
3. Attach it to **MyVPC-1** (vpc-059374692d8327522).



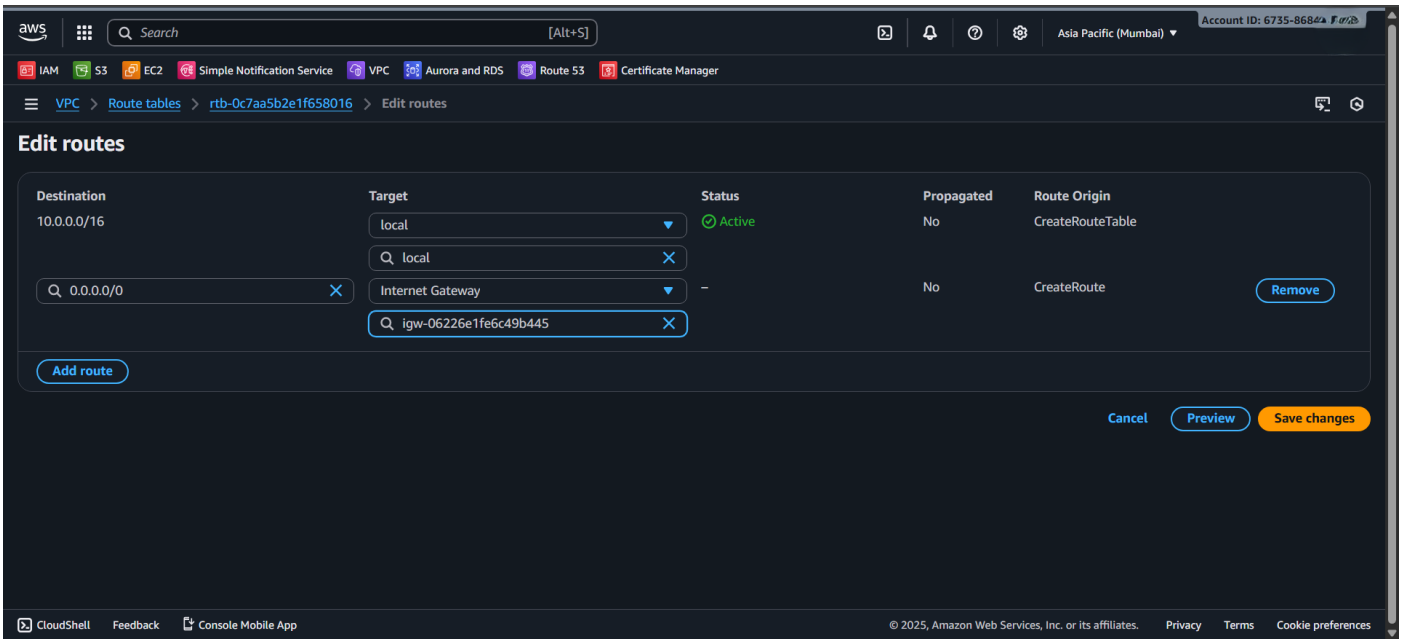


Step 5 — Configure public route table (0.0.0.0/0 → IGW) and associate public-subnet

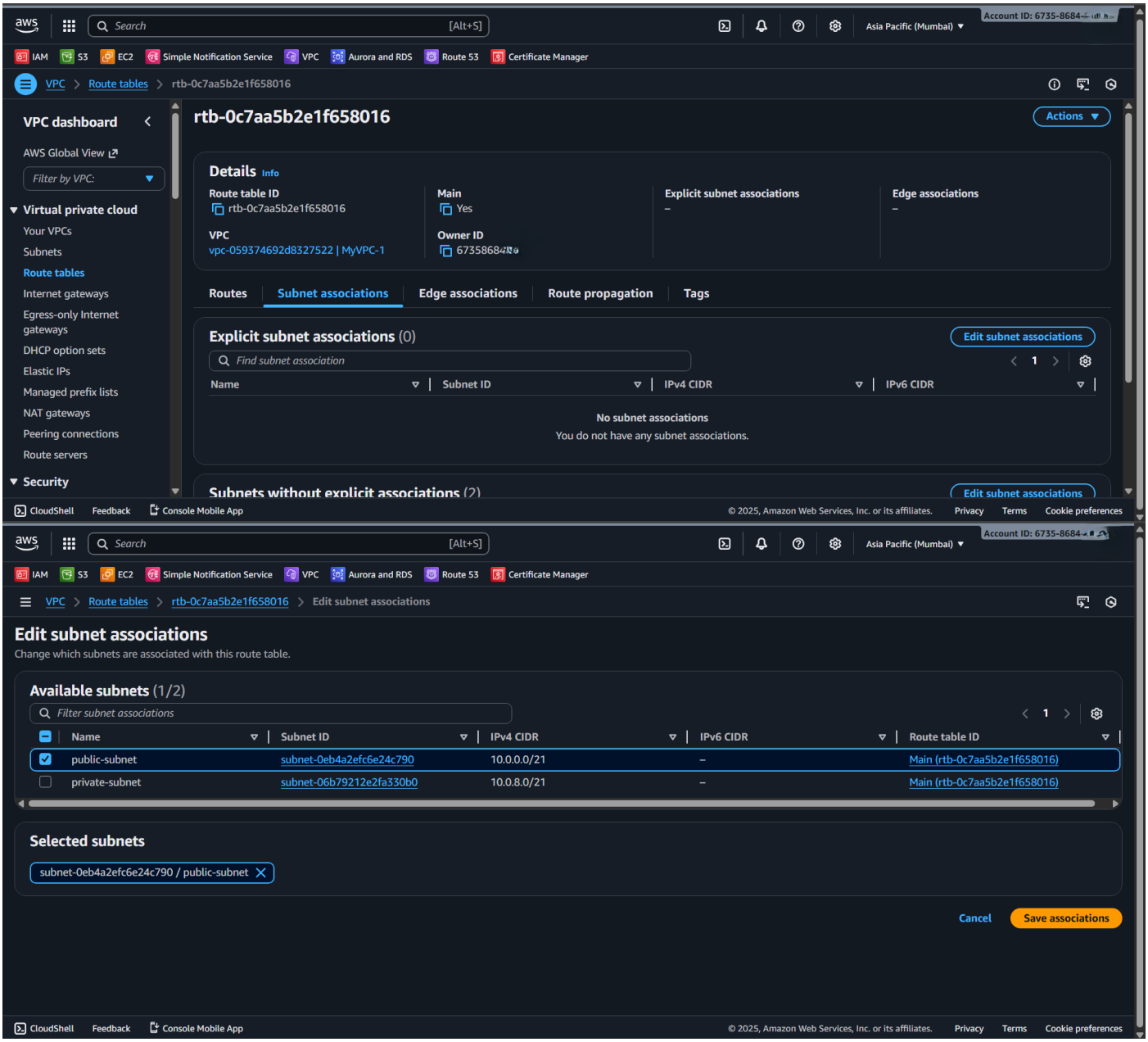
1. Go to **VPC → Route tables**, select the route table for VPC-1 (example shown: rtb-0c7aa5b2e1f658016).



2. Open **Routes → Edit routes** and add: **0.0.0.0/0 → Internet Gateway (igw-06226e1fe6c49b445)**, then save.

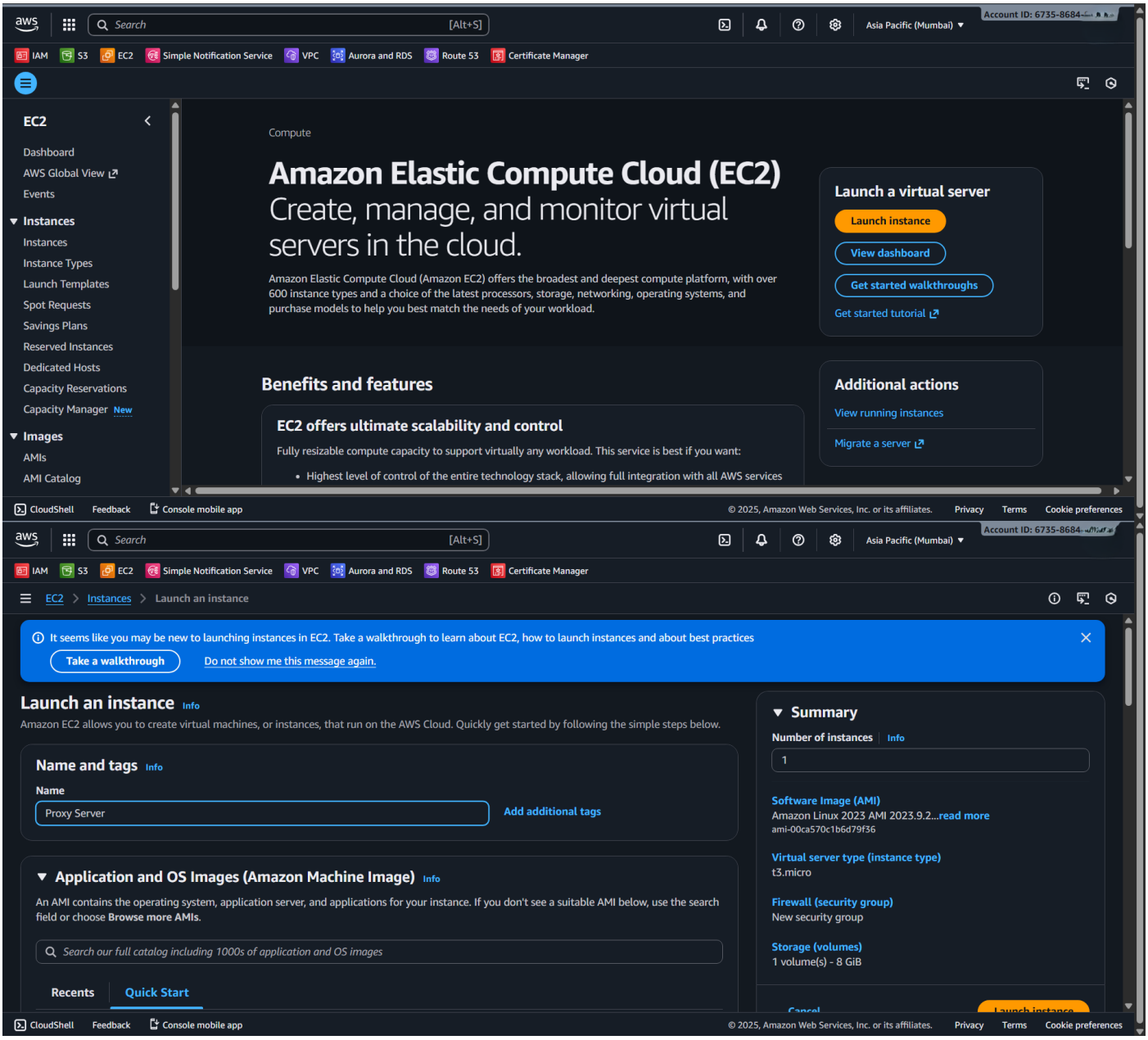


3. Open **Subnet associations → Edit subnet associations**, select **public-subnet**, and save.

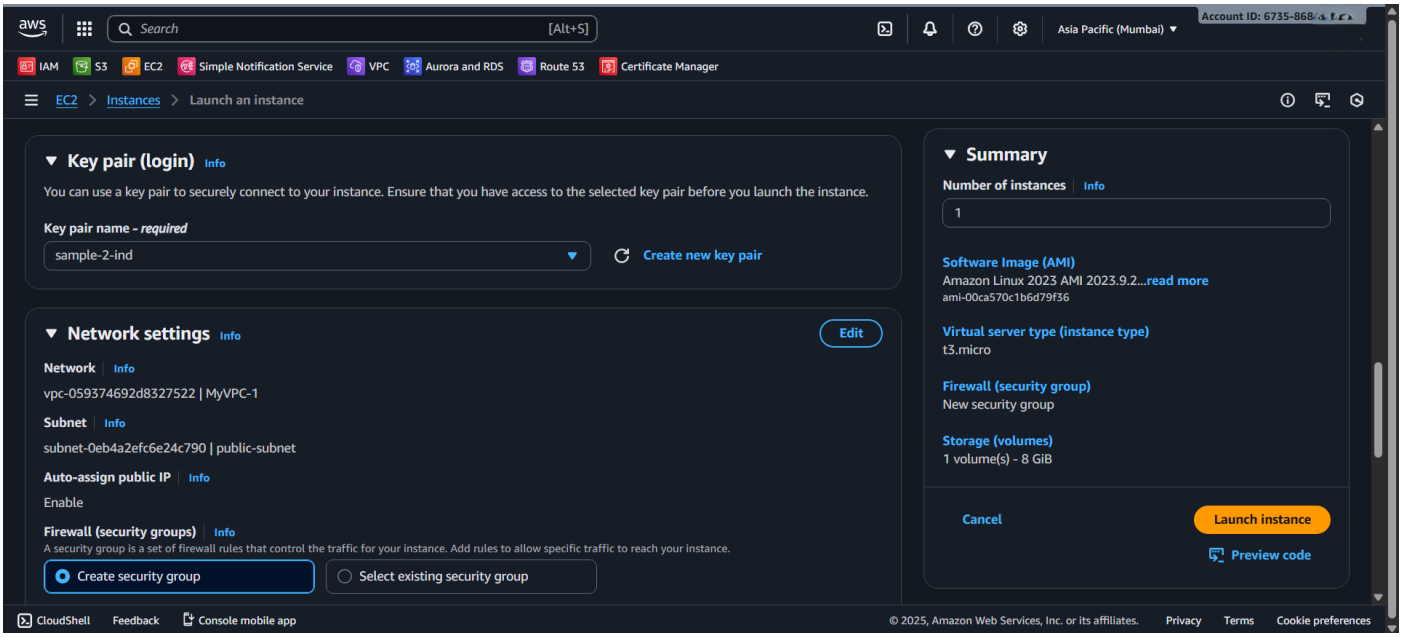


Step 6 — Launch Proxy Server EC2 in the public subnet

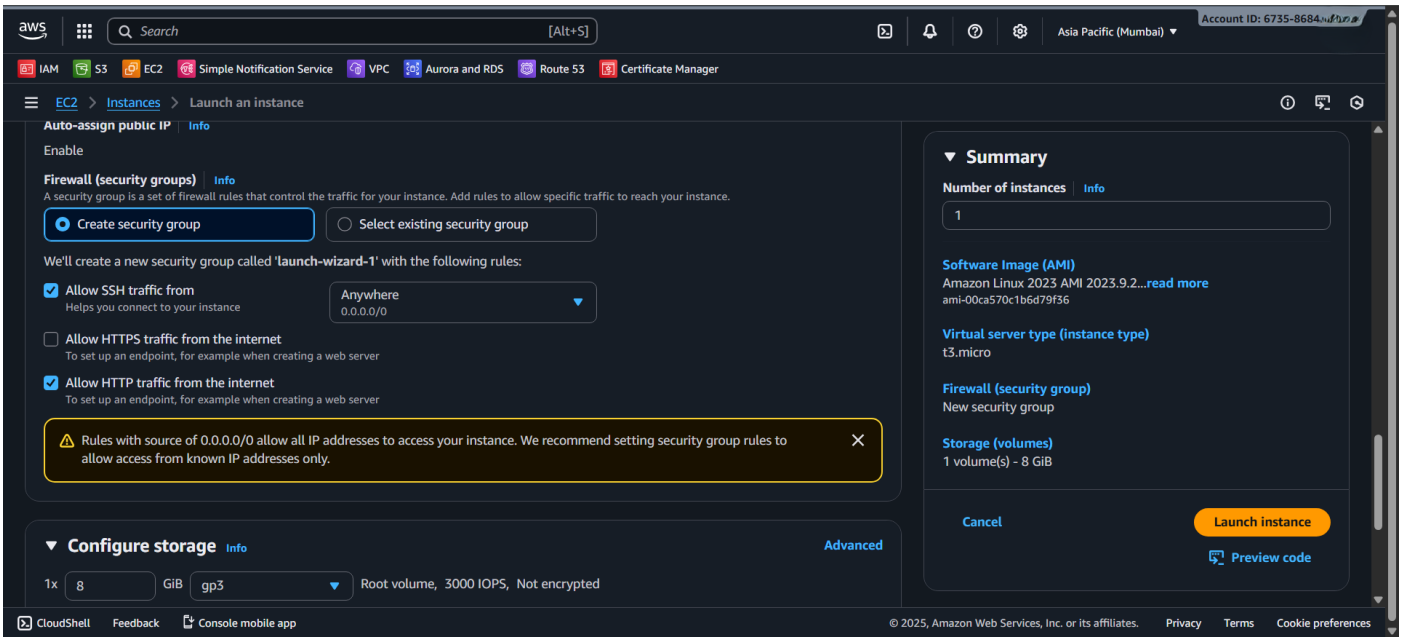
1. Go to **EC2 → Launch instance** and set Name: **Proxy Server**.



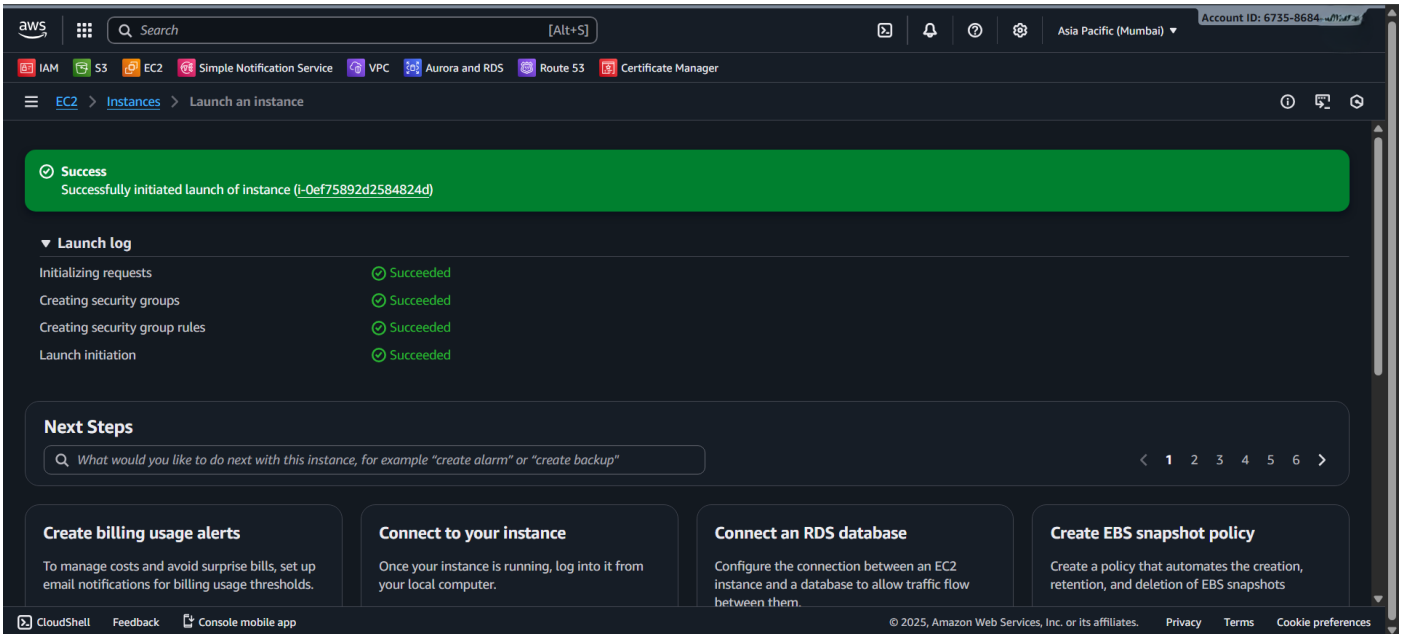
2. Network settings:
    - VPC: **MyVPC-1**
    - Subnet: **public-subnet**
    - Auto-assign public IP: **Enable**



3. Configure security group to allow **SSH (22)** (and HTTP shown enabled in screenshots).

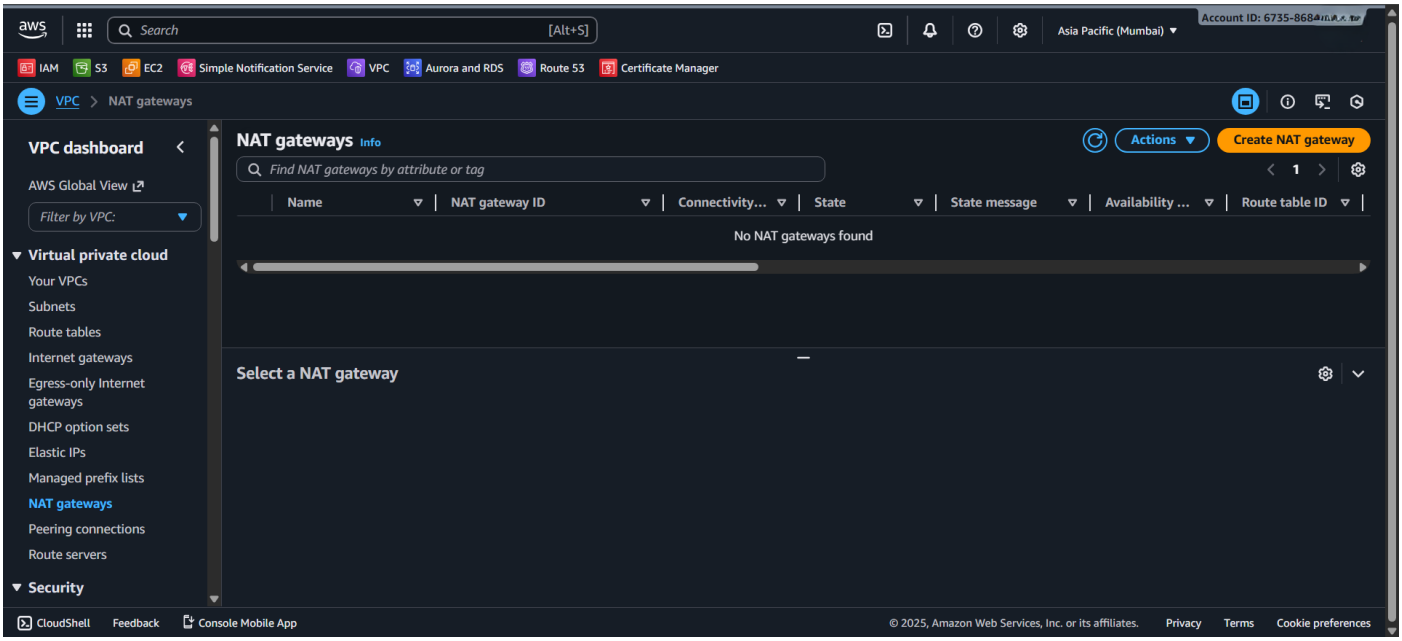


4. Launch and verify Proxy Server is running (example shown: i-0ef75892d2584824d).

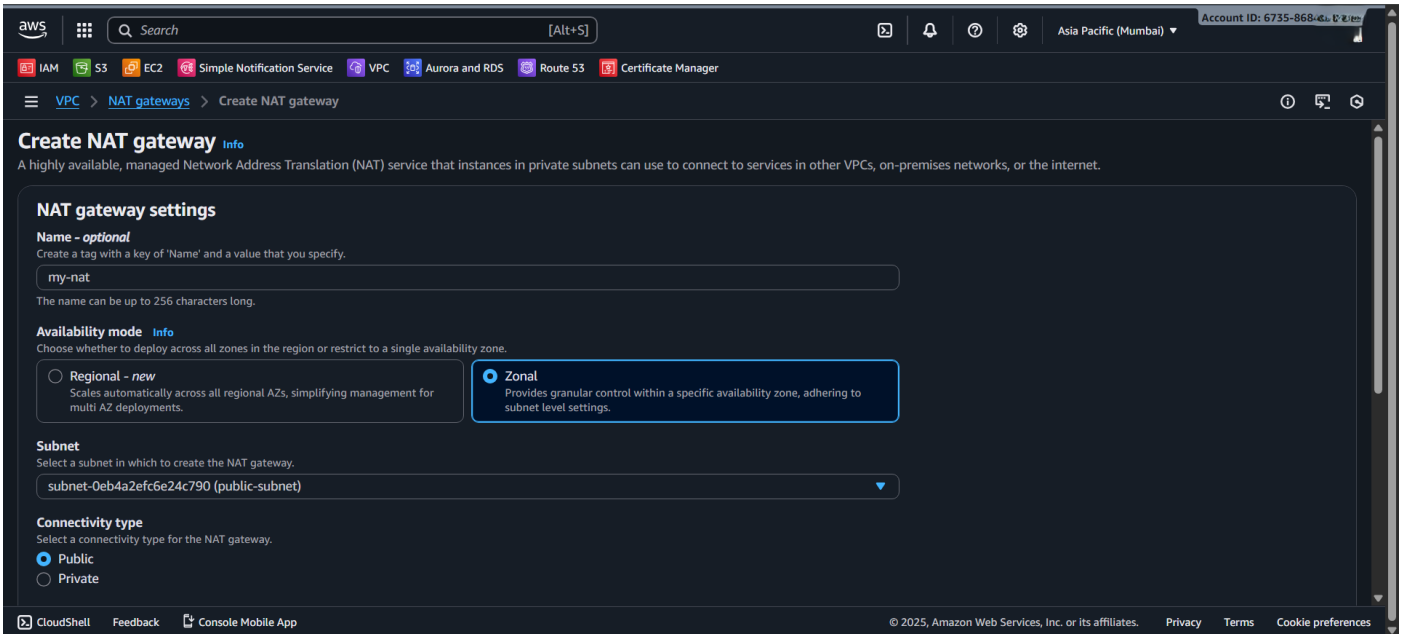


Step 7 — Create NAT Gateway (my-nat) in the public subnet

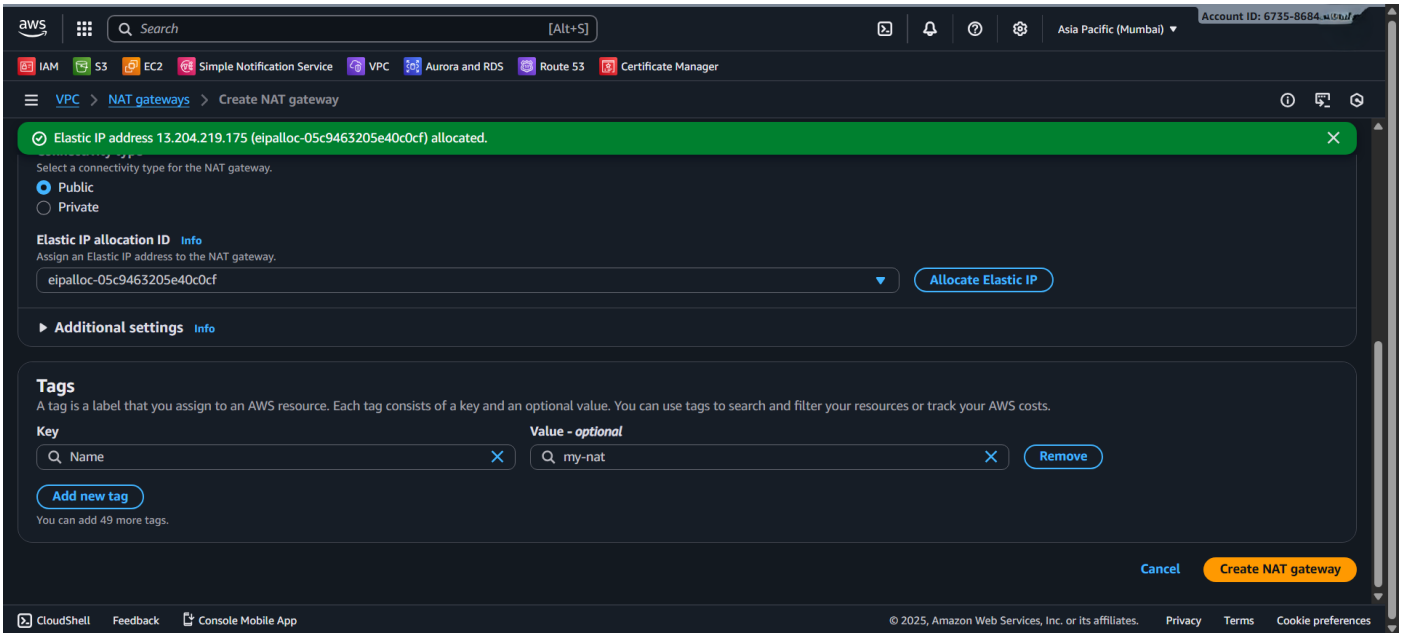
1. Go to **VPC → NAT gateways → Create NAT gateway**.



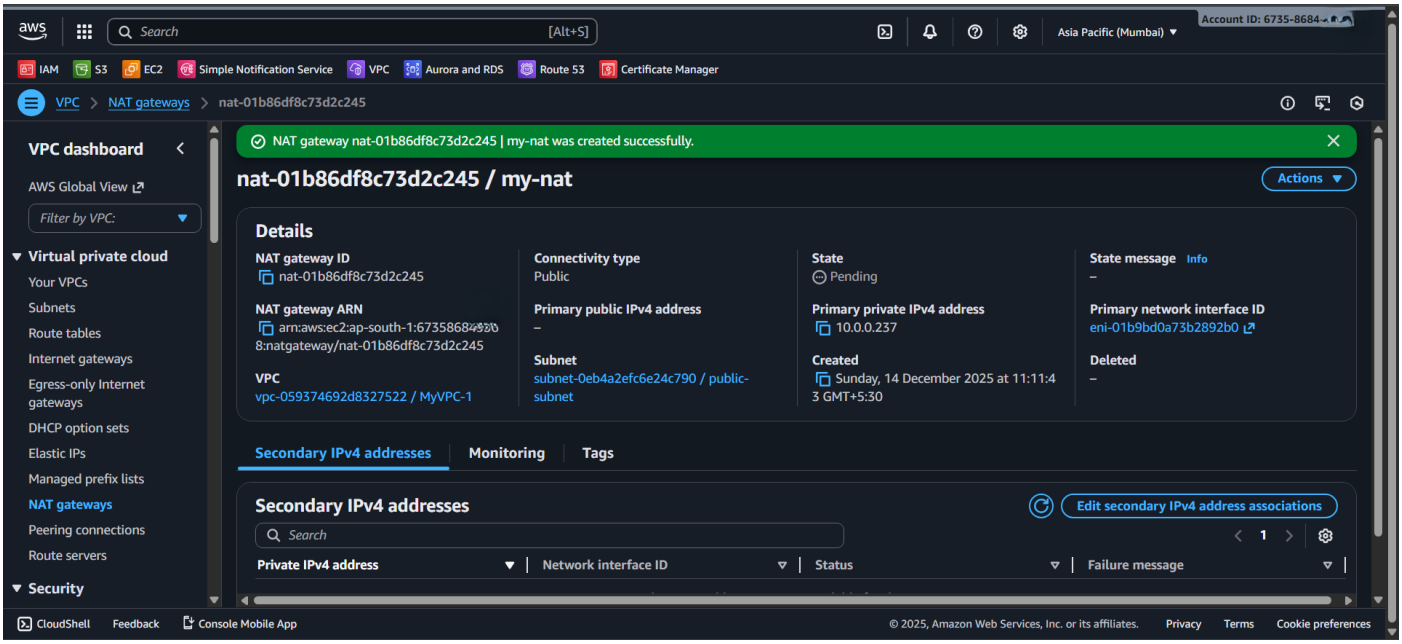
2. Set:
    - Name: **my-nat**
    - Subnet: **public-subnet**
    - Connectivity: **Public**



3. Allocate an Elastic IP (example shown: **13.204.219.175**) and create NAT gateway.

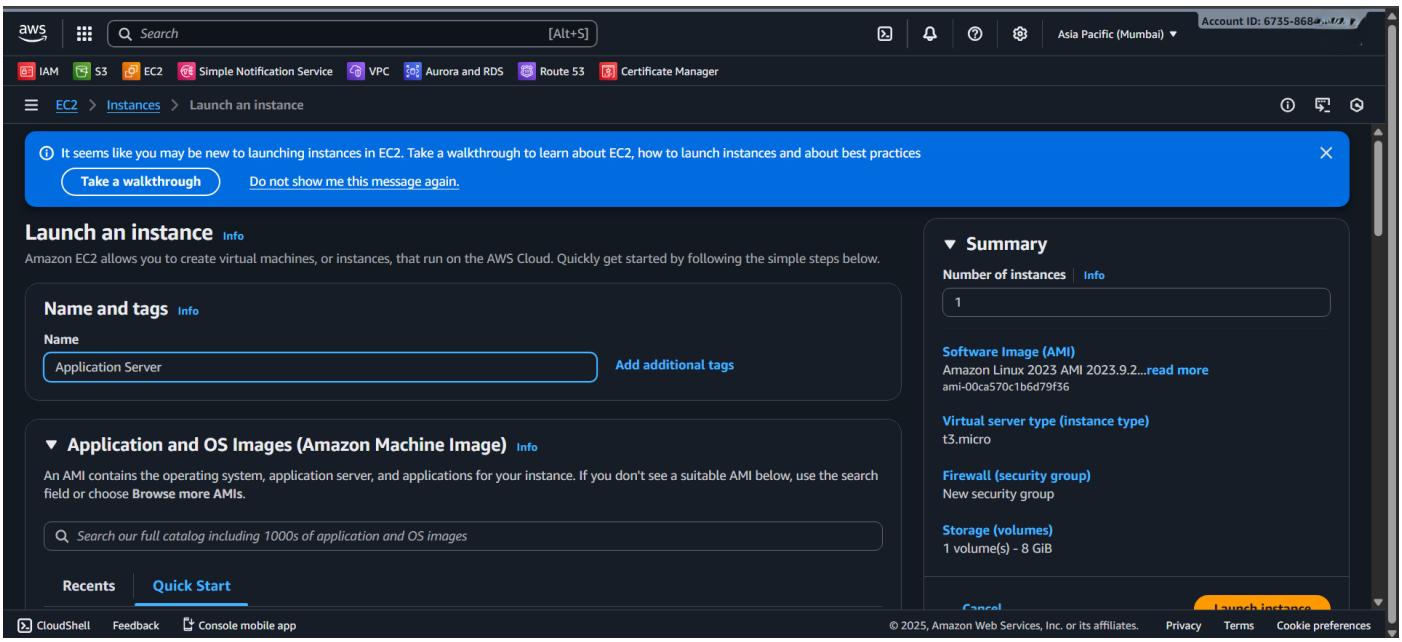


4. Verify NAT gateway is created (example shown: nat-01b86df8c73d2c245 / my-nat).

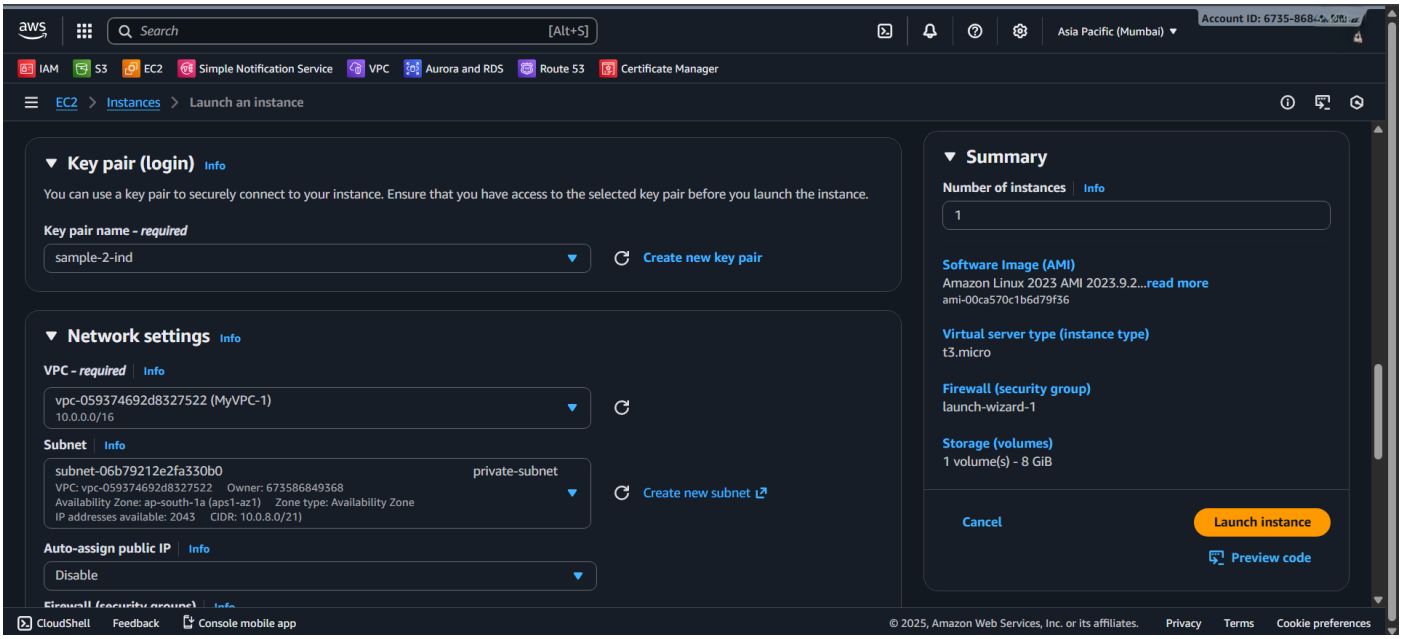


Step 8 — Launch Application Server EC2 in the private subnet

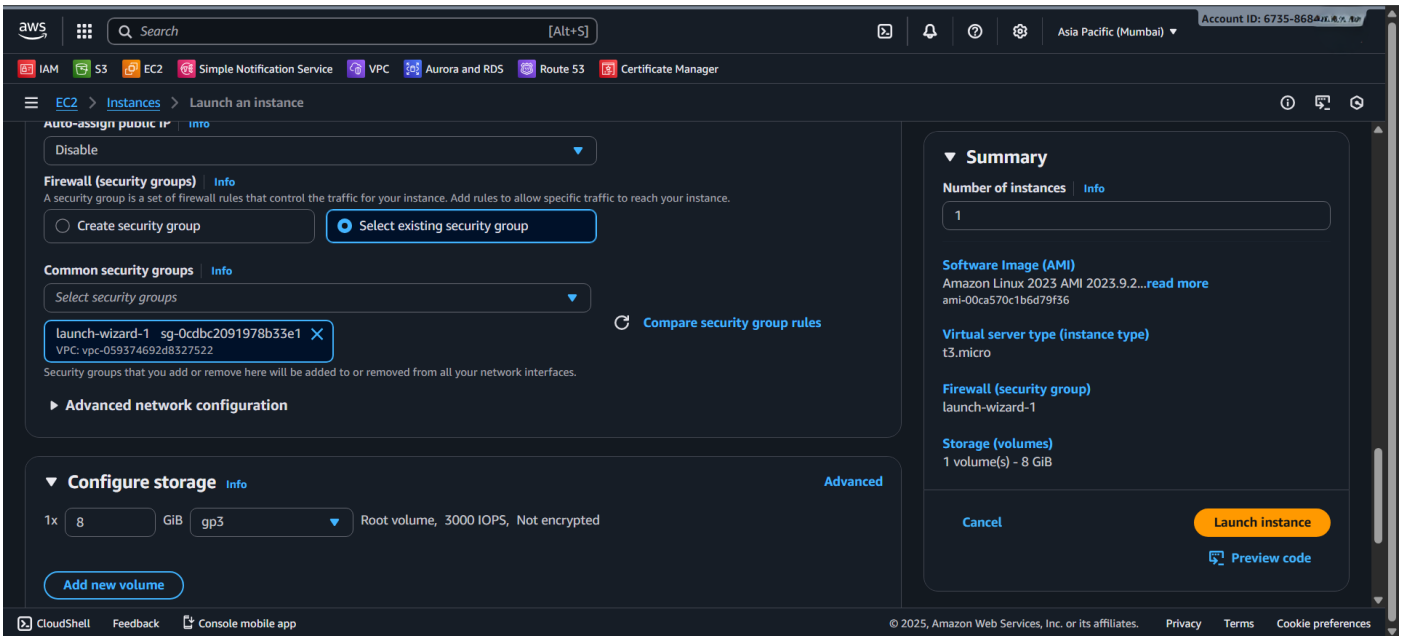
1. Go to **EC2 → Launch instance** and set Name: **Application Server**.



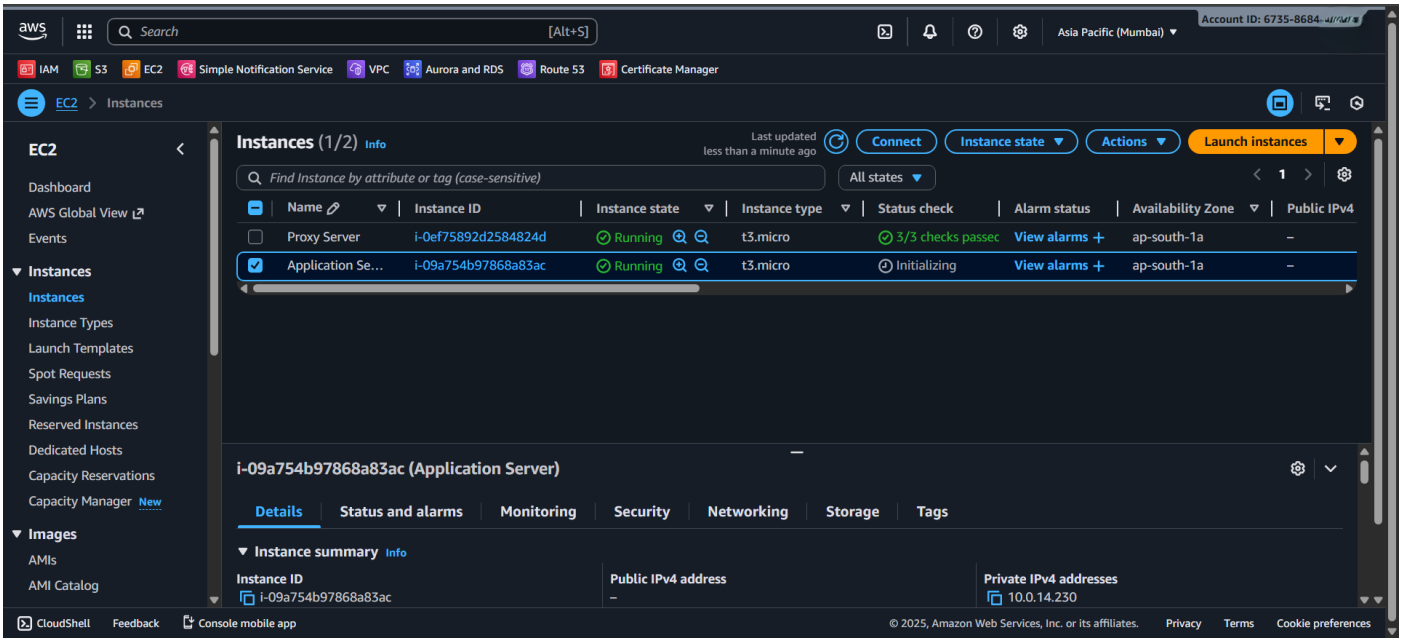
2. Network settings:
    - VPC: **MyVPC-1**
    - Subnet: **private-subnet**
    - Auto-assign public IP: **Disable**



3. Select existing security group **launch-wizard-1 (sg-0cdbc2091978b33e1)**.

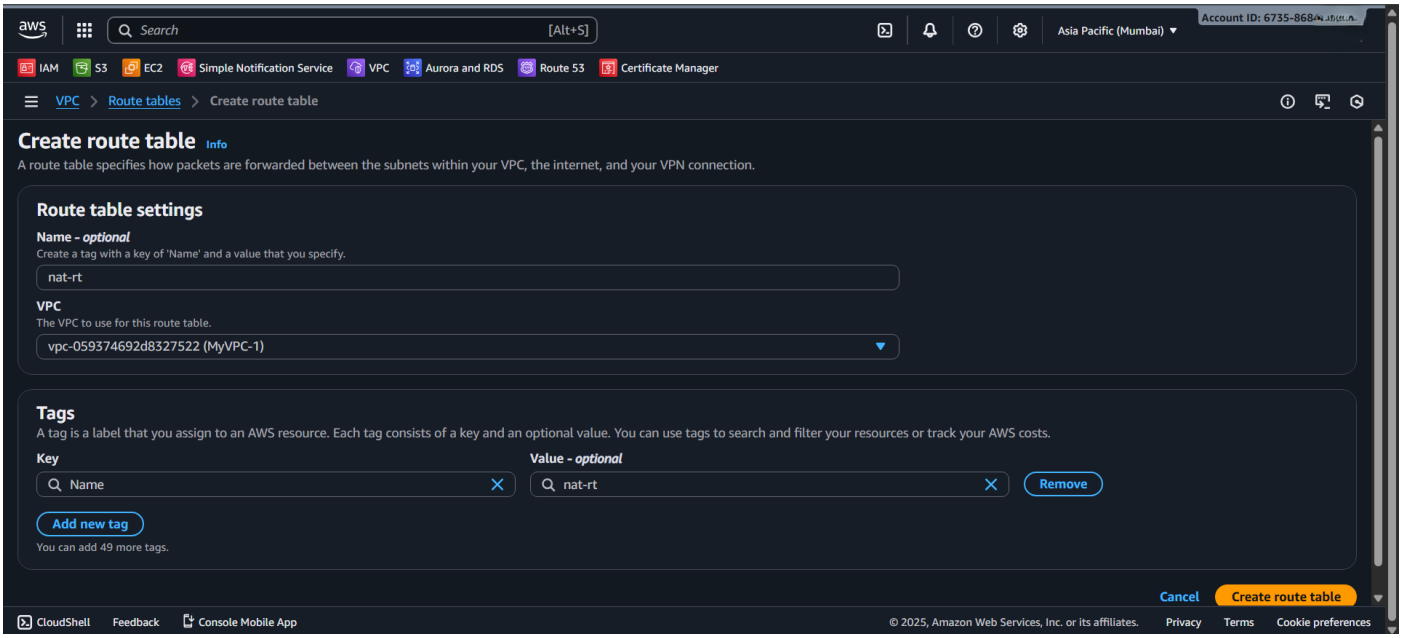


4. Verify Application Server is running and has private IP **10.0.14.230**.

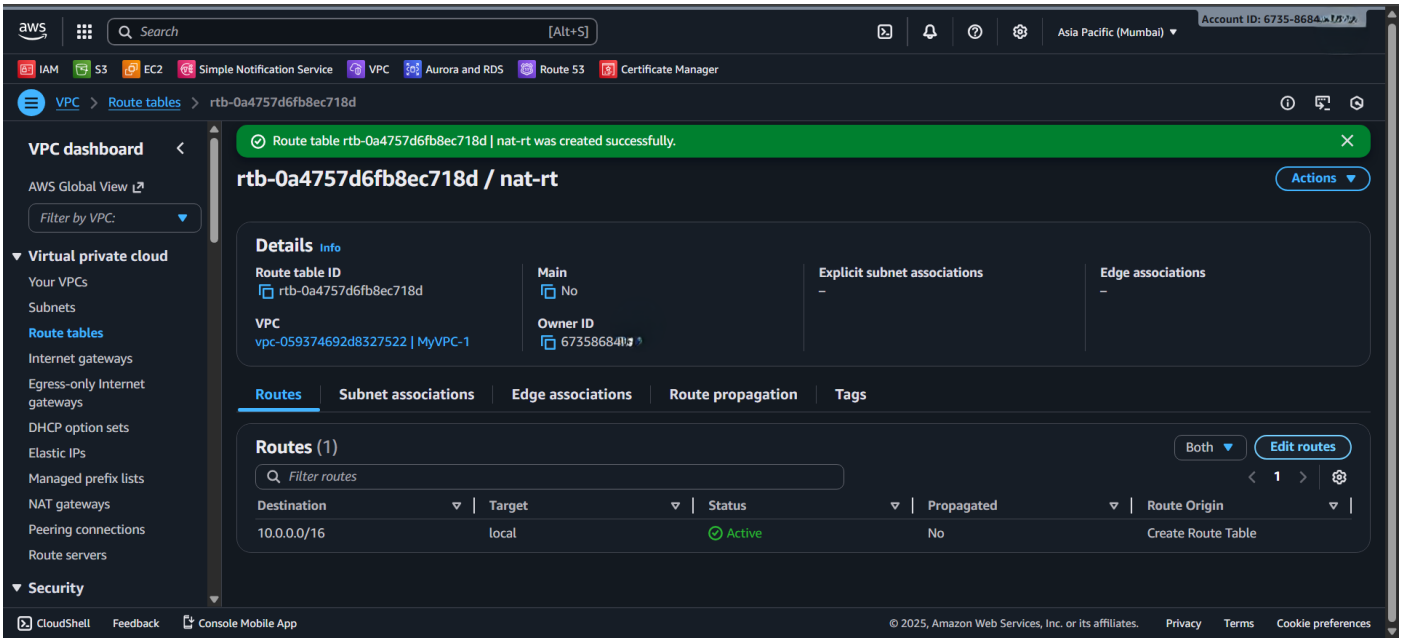


Step 9 — Create a private subnet route table (nat-rt) and route private traffic to NAT

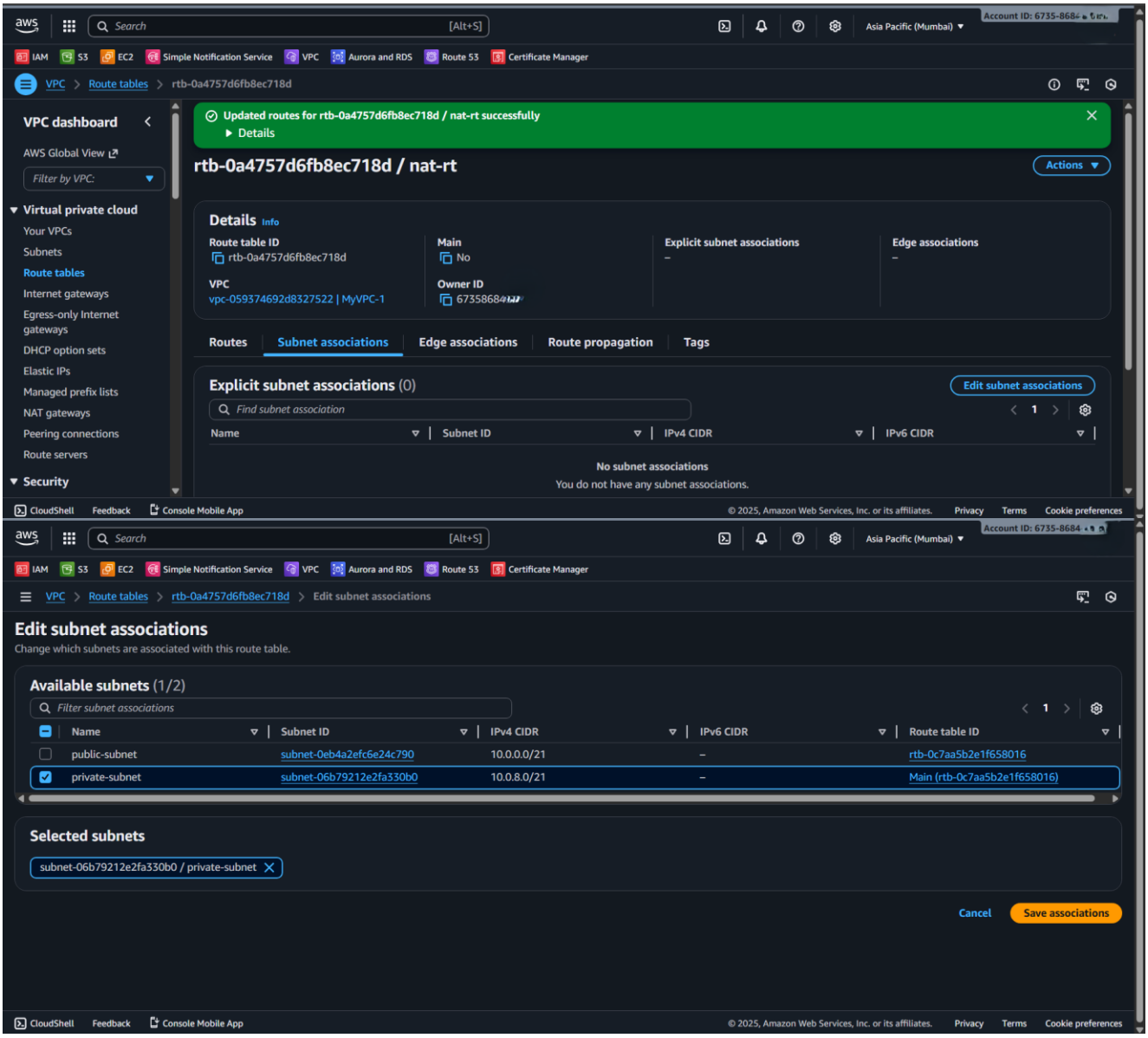
1. Go to **VPC → Route tables → Create route table**.
2. Name it **nat-rt** and select VPC **MyVPC-1**.



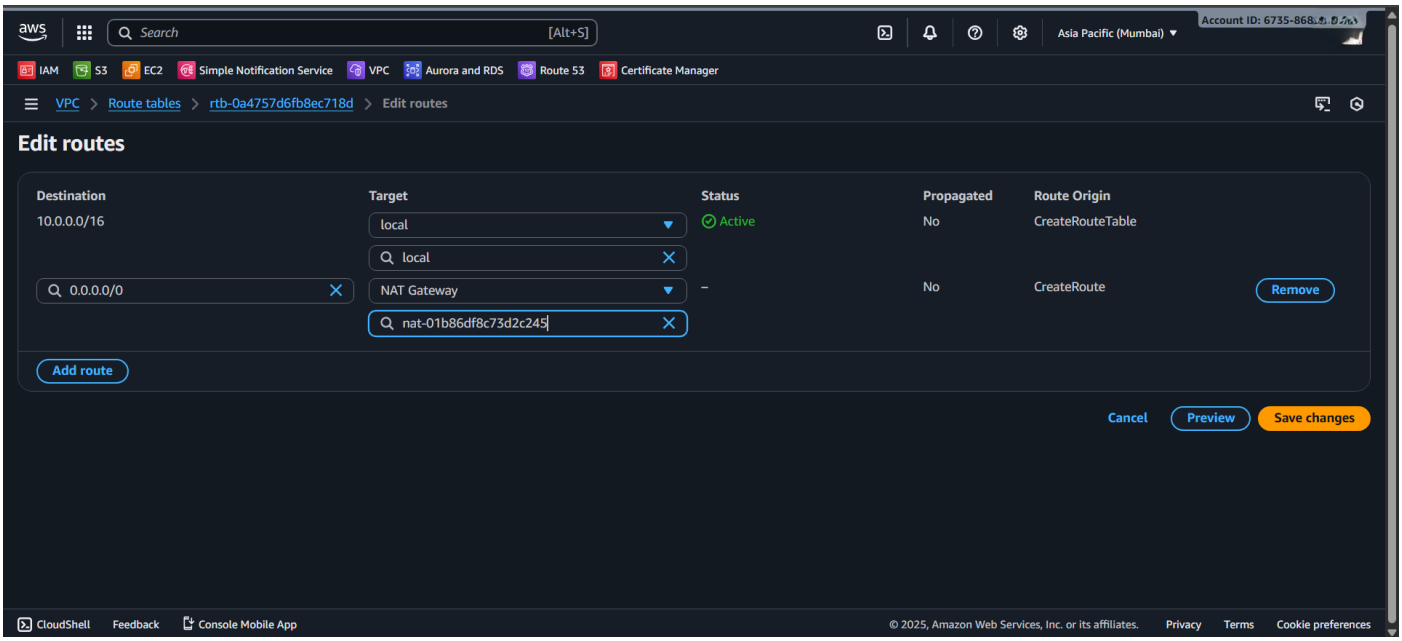
3. Verify route table created successfully (example shown: rtb-0a4757d6fb8ec718d / nat-rt).



4. Associate **private-subnet** with **nat-rt** (explicit association should show subnet-06b79212e2fa330b0 / private-subnet).

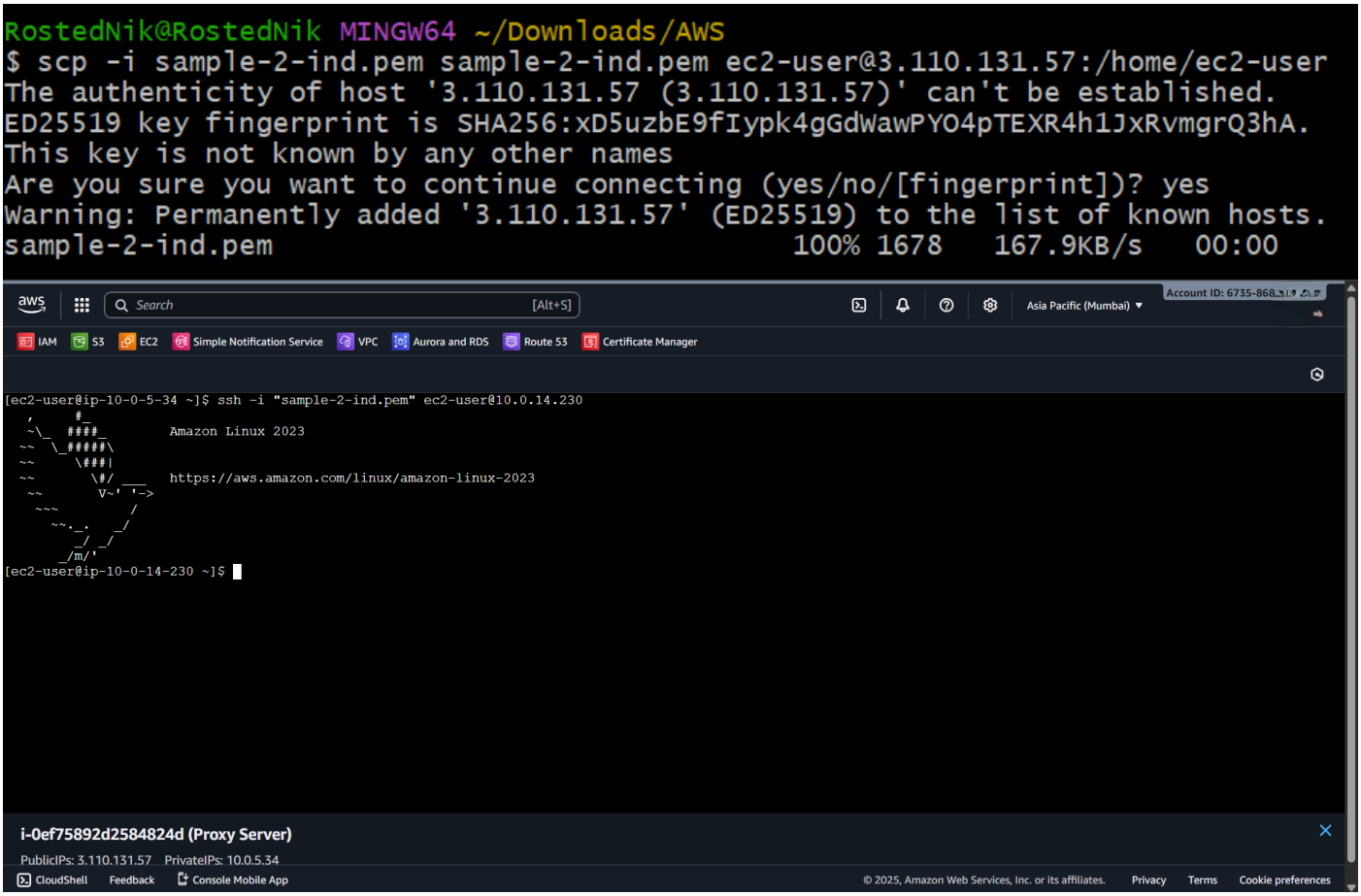


5. In **nat-rt → Routes → Edit routes**, add:
    - **0.0.0.0/0 → NAT Gateway (nat-01b86df8c73d2c245)**, then save.

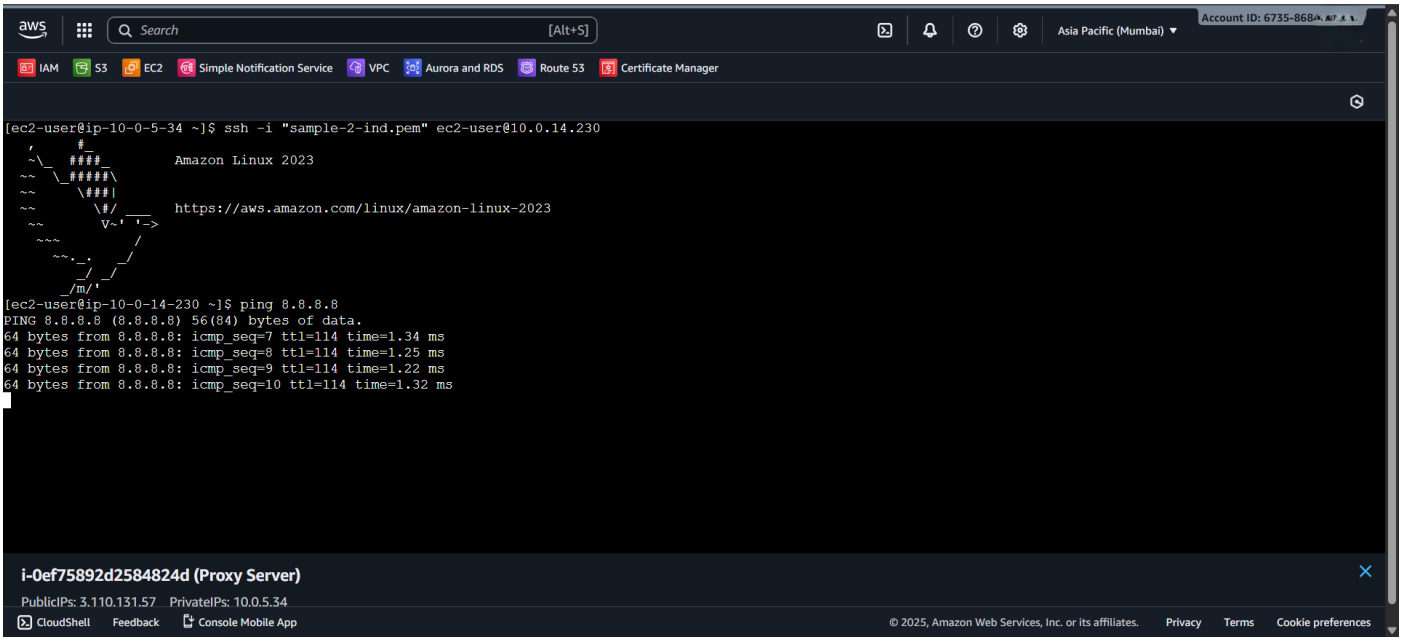


Step 10 — Validate private subnet outbound connectivity

1. Use SSH from the Proxy environment to connect to the Application Server private IP **10.0.14.230** using sample-2-ind.pem.

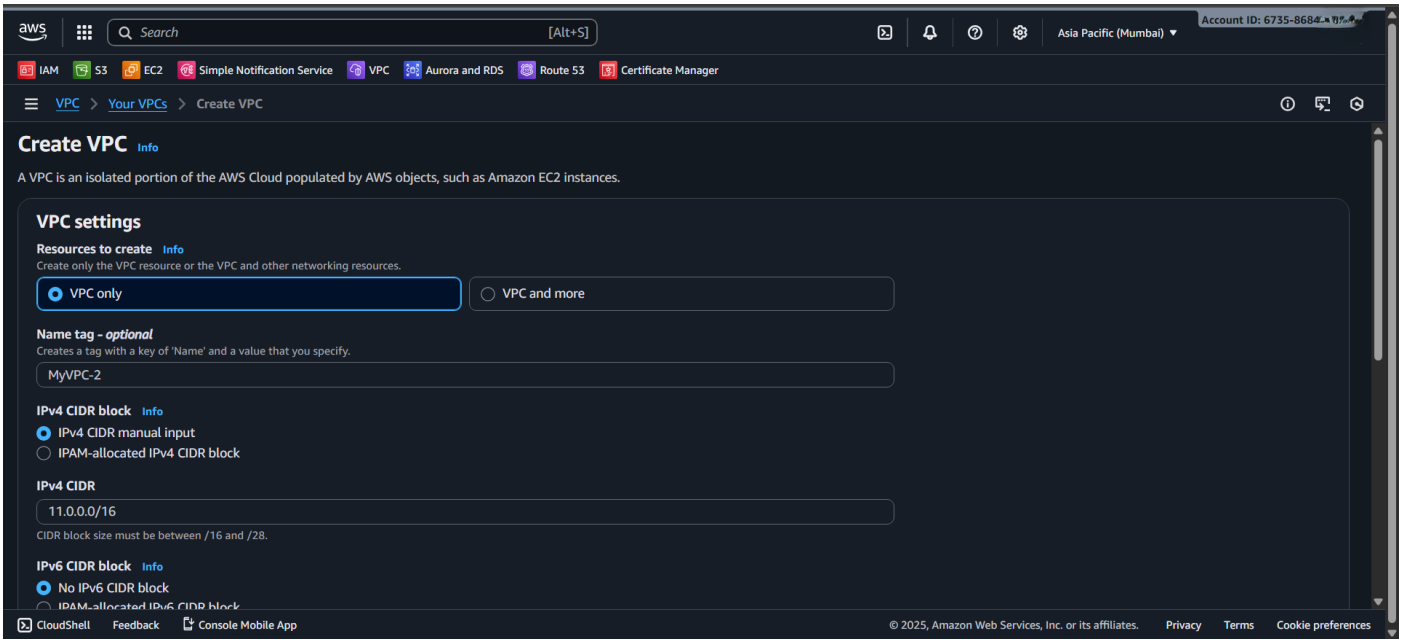


2. From the Application Server session, run ping 8.8.8.8 and verify responses are received.

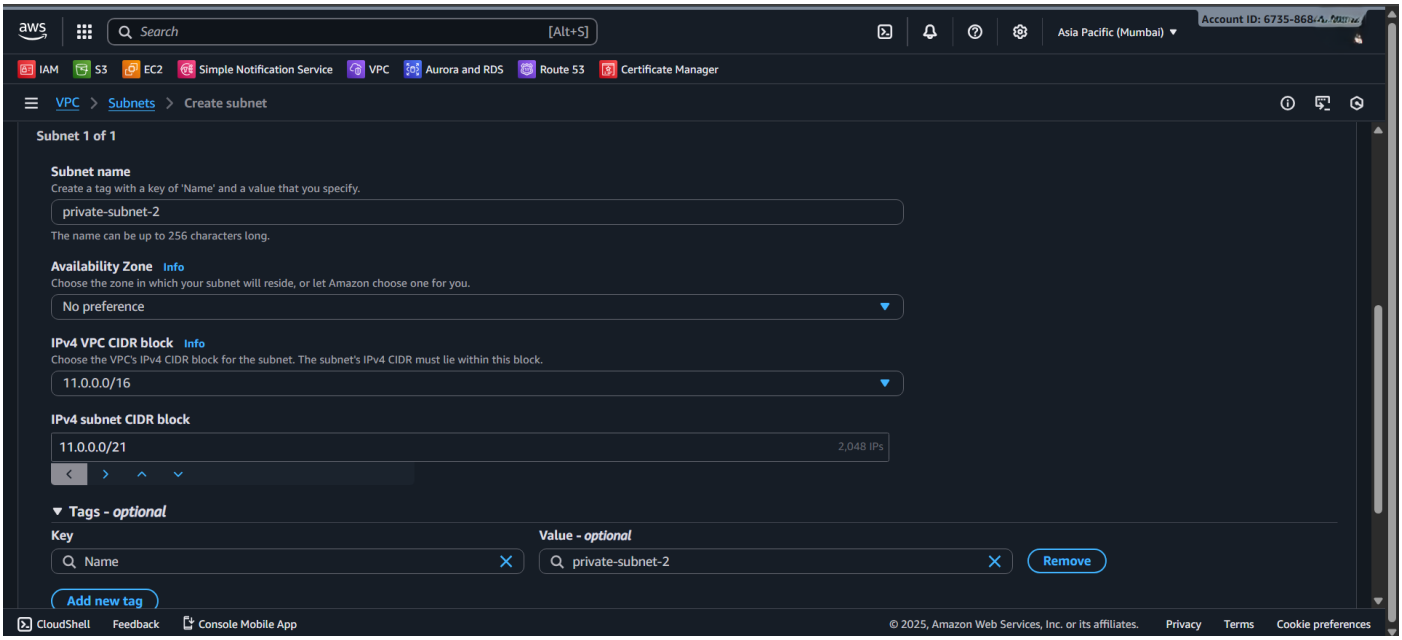


Step 11 — Create VPC-2 (MyVPC-2) and its private subnet

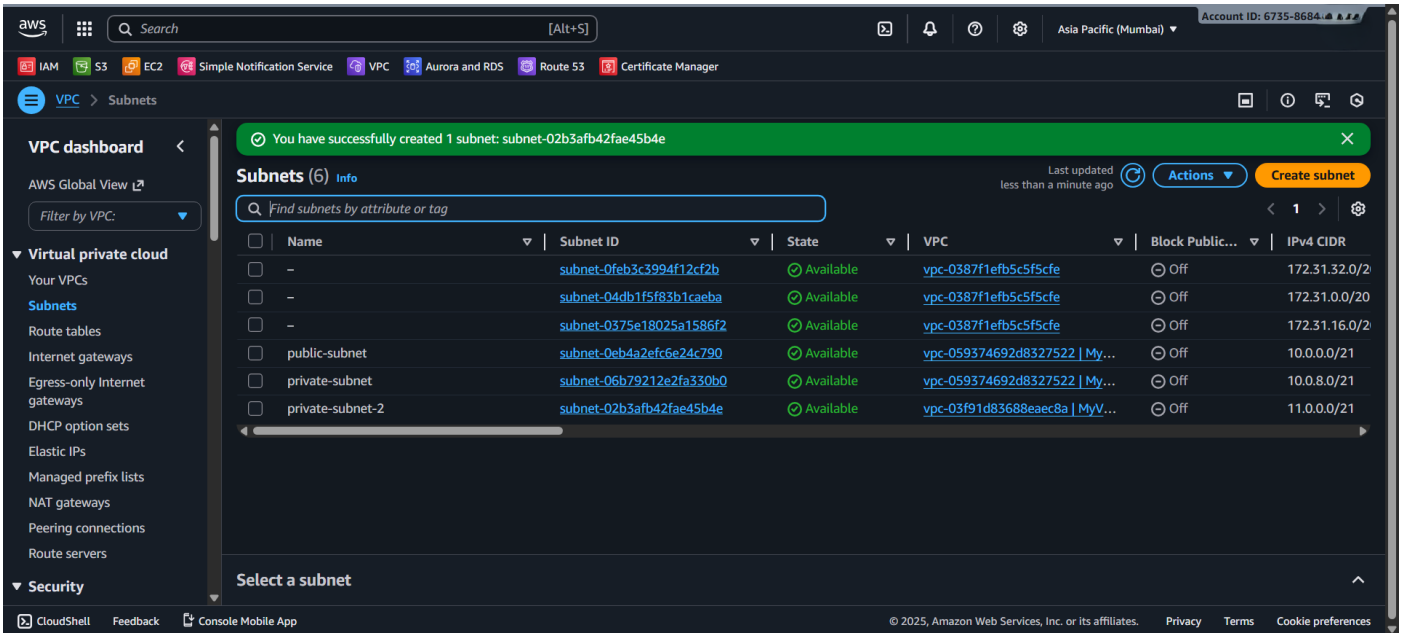
1. Go to **VPC → Your VPCs → Create VPC** and create:
    - Name: **MyVPC-2**
    - IPv4 CIDR: **11.0.0.0/16**



2. Go to **VPC → Subnets → Create subnet** and create:
    - Name: **private-subnet-2**
    - CIDR: **11.0.0.0/21**

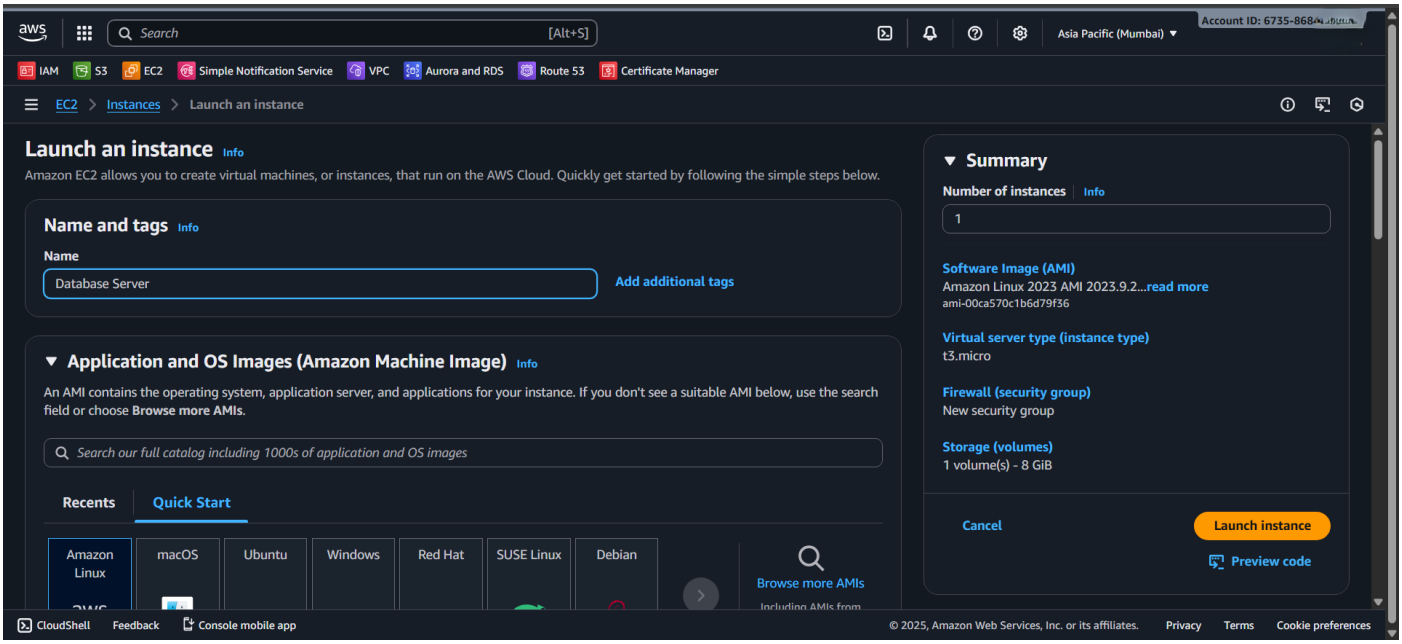


3. Verify subnet is created (example shown: subnet-02b3afb42fae45b4e).

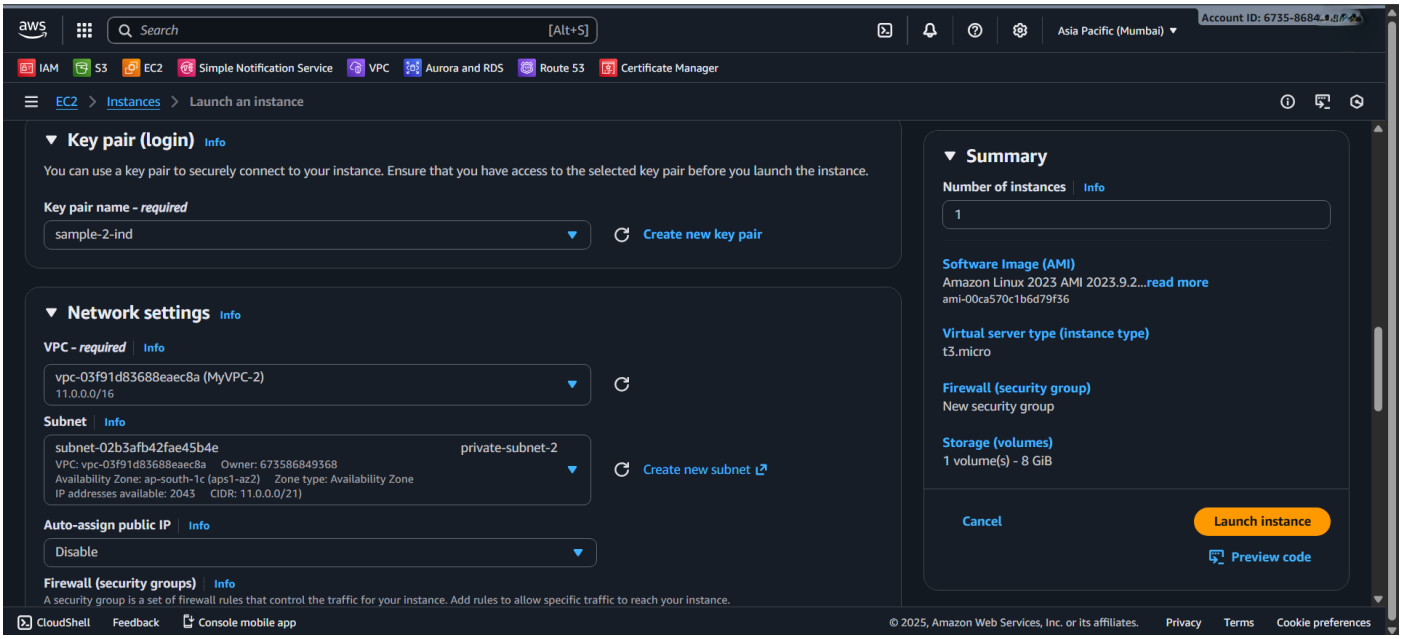


Step 12 — Launch Database Server EC2 in VPC-2 private subnet

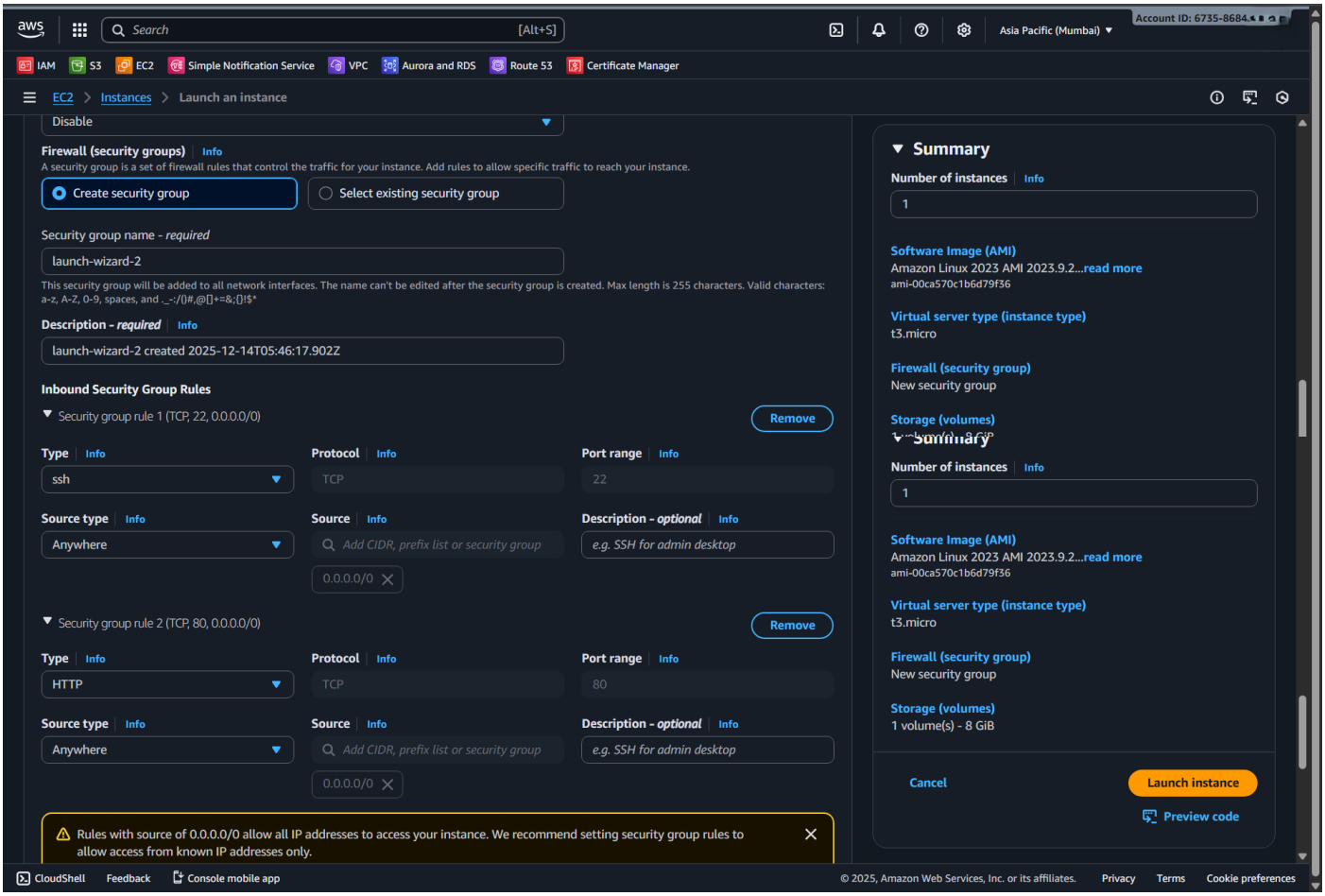
1. Go to **EC2 → Launch instance** and set Name: **Database Server**.



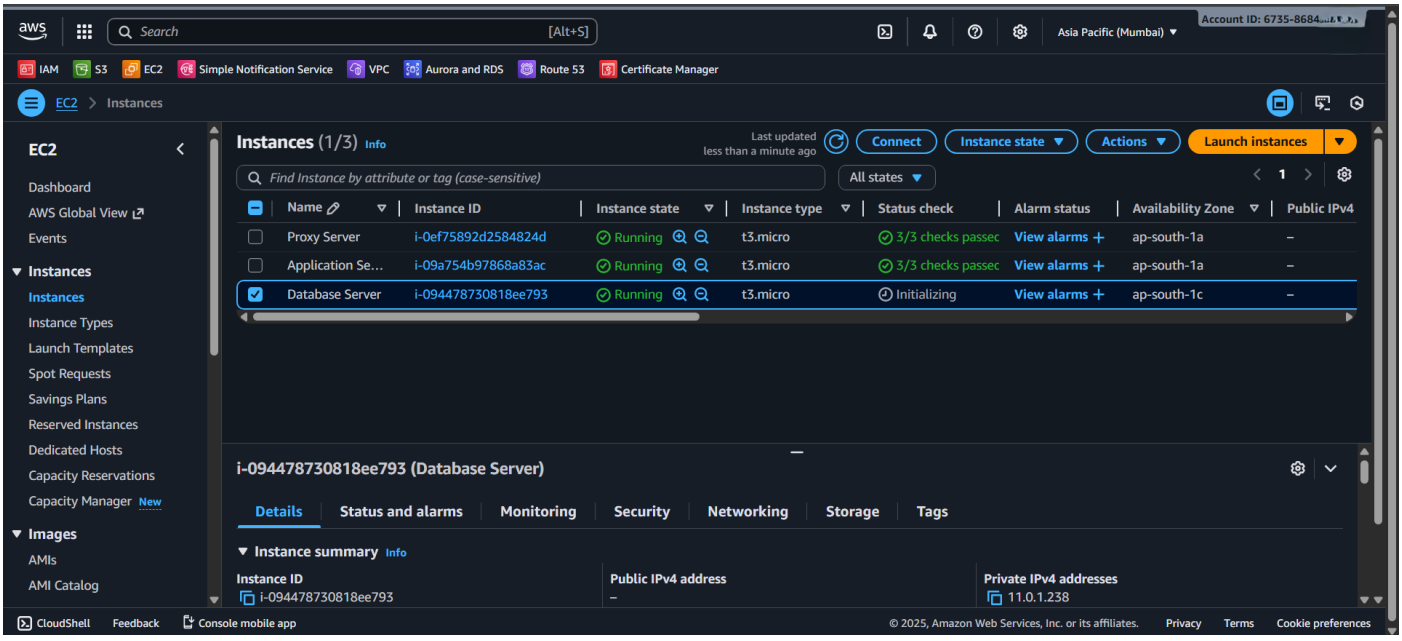
2. Network settings:
   - VPC: **MyVPC-2 (vpc-03f91d83688eaec8a)**
   - Subnet: **private-subnet-2**
   - Auto-assign public IP: **Disable**



3. Create a new security group **launch-wizard-2** (SSH inbound shown in screenshot).

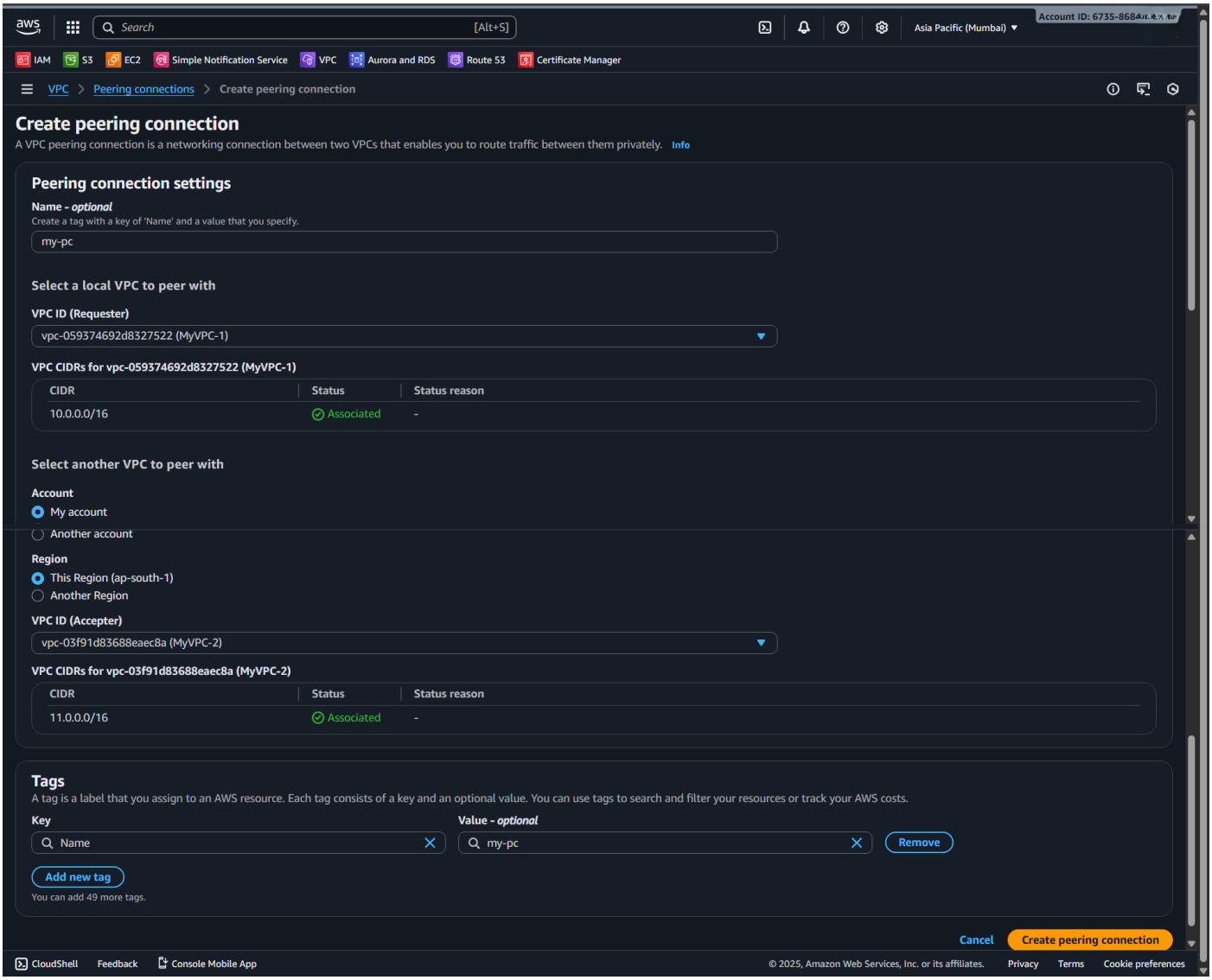


4. Verify Database Server is running and has private IP **11.0.1.238** (example shown).

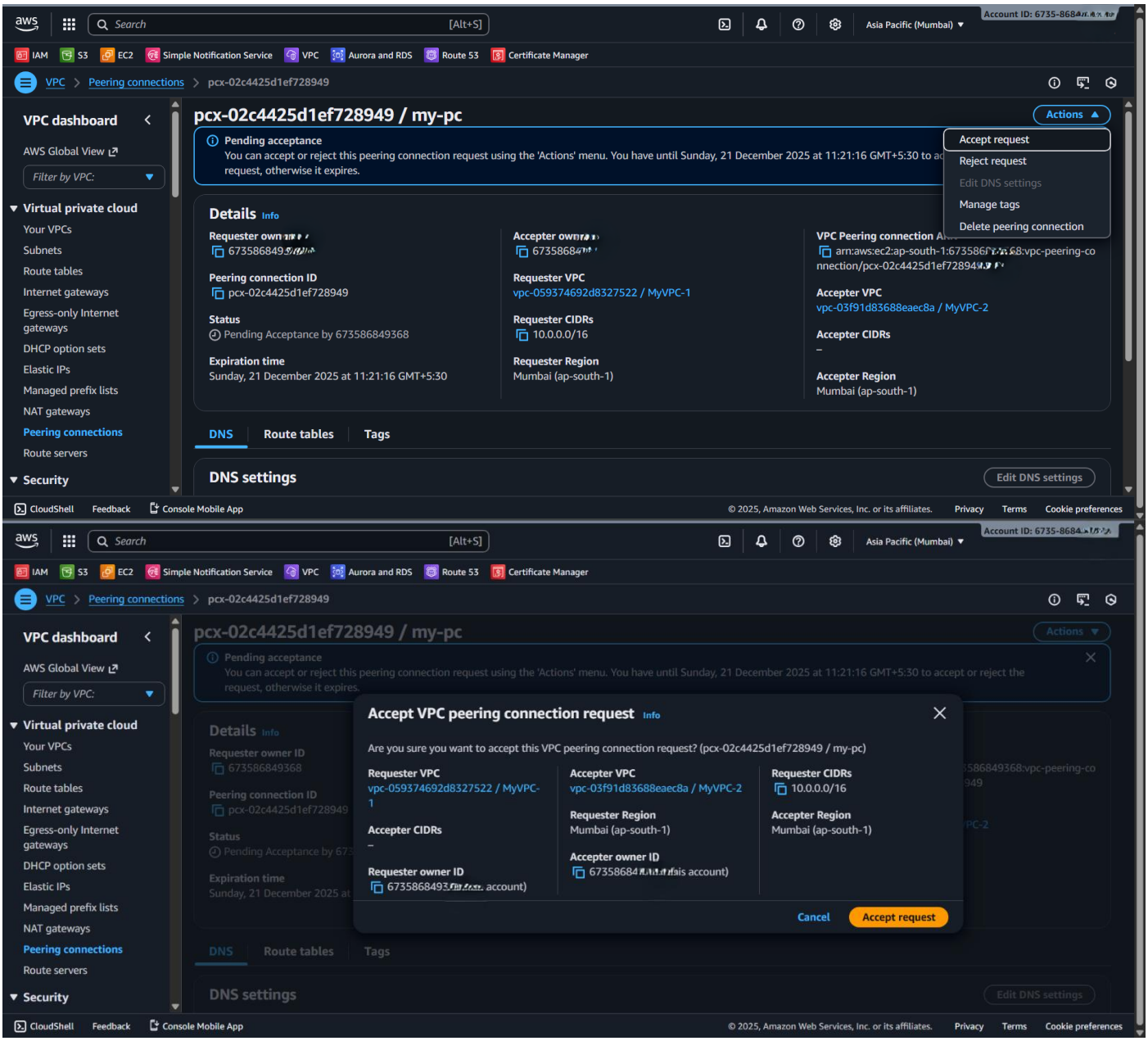


Step 13 — Create and accept VPC Peering (my-pc)

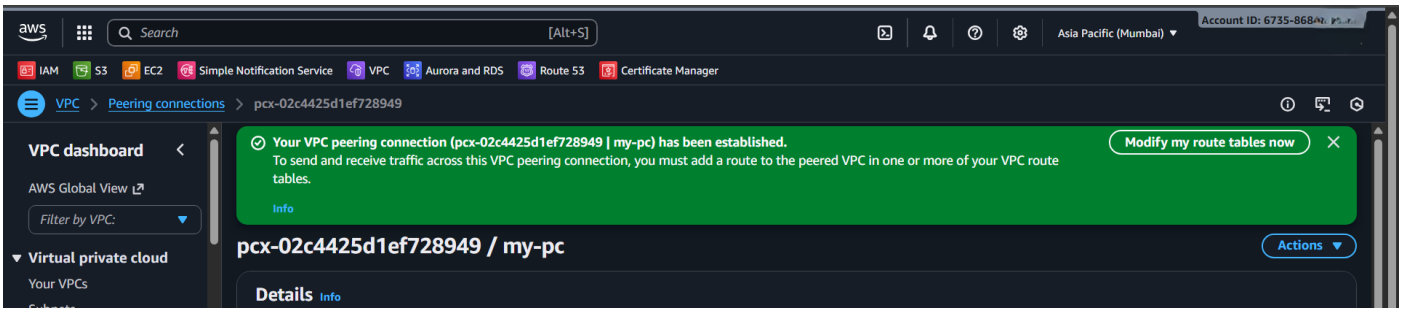
1. Go to **VPC → Peering connections → Create peering connection**.
2. Set:
    - Name: **my-pc**
    - Requester: **MyVPC-1 (vpc-059374692d8327522)**
    - Accepter: **MyVPC-2 (vpc-03f91d83688eaec8a)**



3. Open the created peering connection (example shown: pcx-02c4425d1ef728949) and select **Actions → Accept request**.

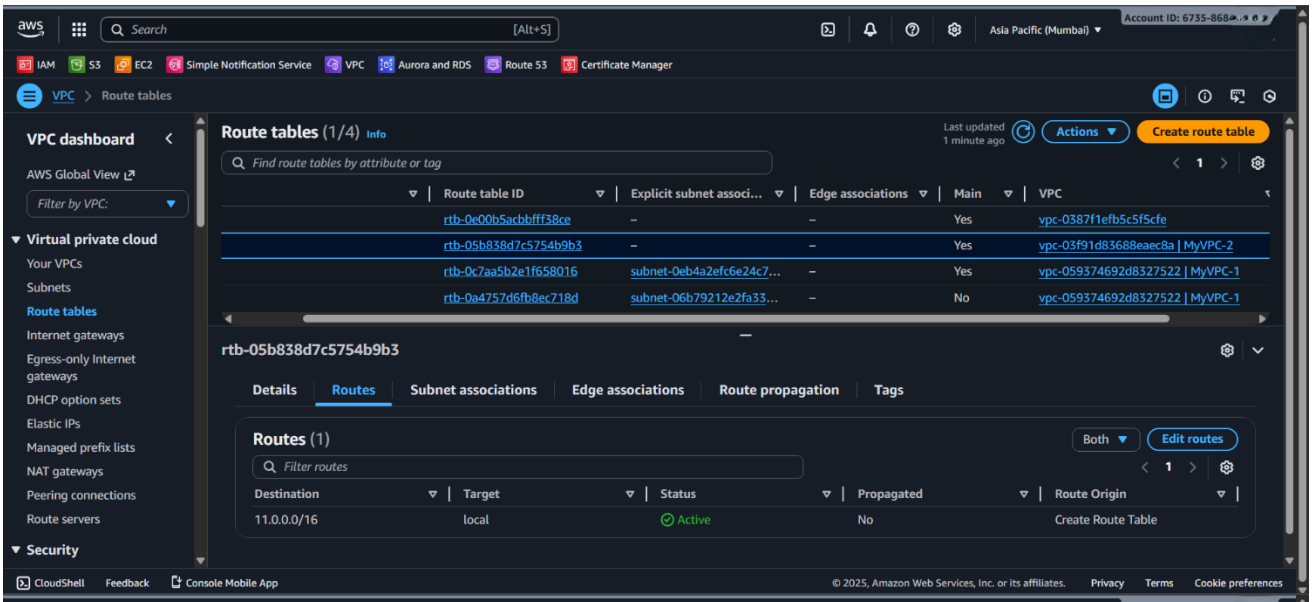


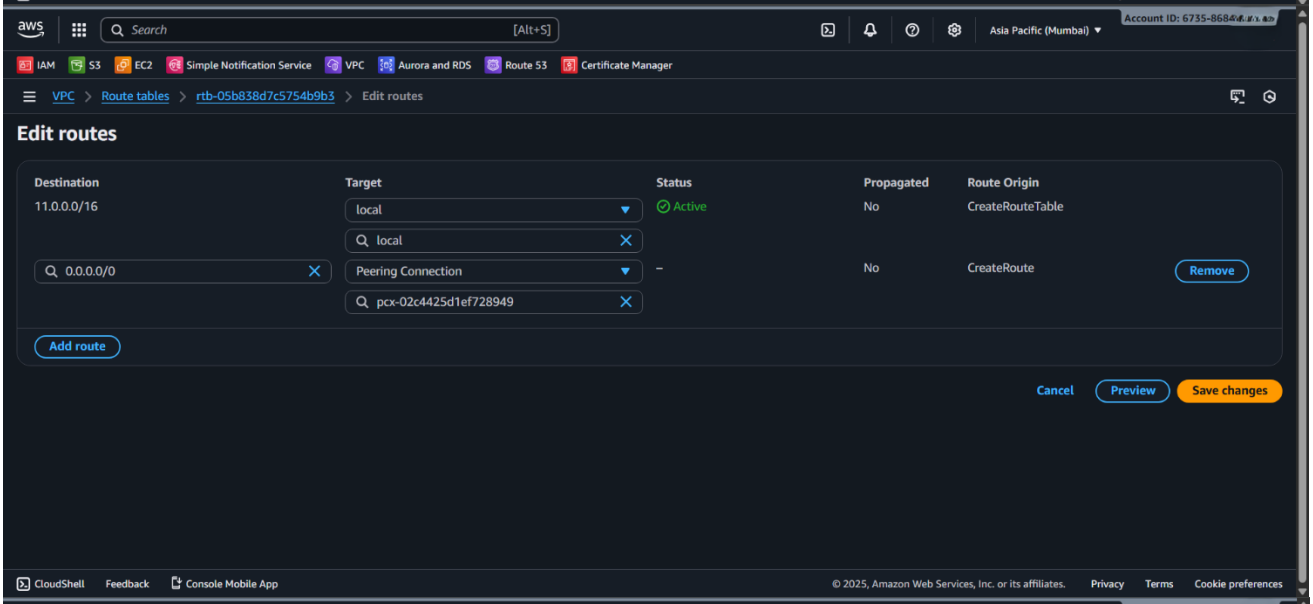
4. Acceptance is required for the peering connection to become active.

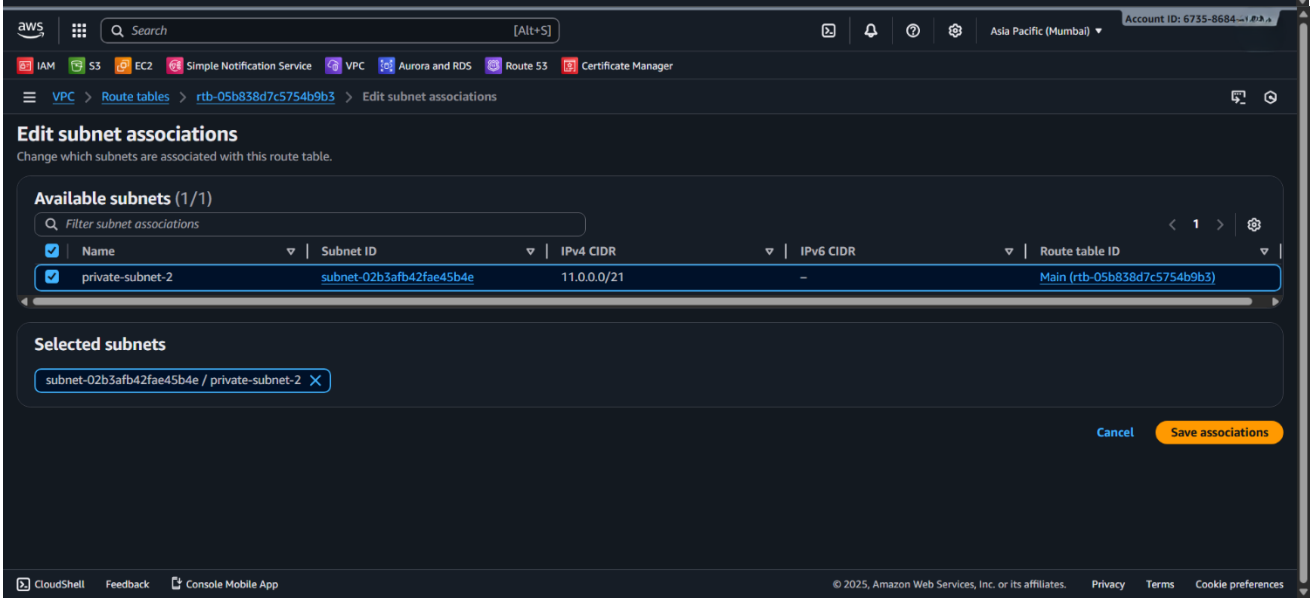


Step 14 - finalize end-to-end App → DB access

- To complete private App → DB communication, add route table entries on both sides so:
  - VPC-1 routes **11.0.0.0/16 → pcx-…**
  - VPC-2 routes **10.0.0.0/16 → pcx-…**







- After routes are added, validate connectivity from **Application Server** to **Database Server** private IP (11.0.1.238) using the required DB port or a network test.