Assignment - VPC

Problem Statement:

You work for XYZ Corporation and based on the expansion requirements of your corporation you have been asked to create and set up a distinct Amazon VPC for the production and development team. You are expected to perform the following tasks for the respective VPCs.

Production Network:

1. Design and build a 4-tier architecture.
2. Create 5 subnets out of which 4 should be private named app1, app2, dbcache and db and one should be public, named web.
3. Launch instances in all subnets and name them as per the subnet that they have been launched in.
4. Allow dbcache instance and app1 subnet to send internet requests.
5. Manage security groups and NACLs.

Development Network:

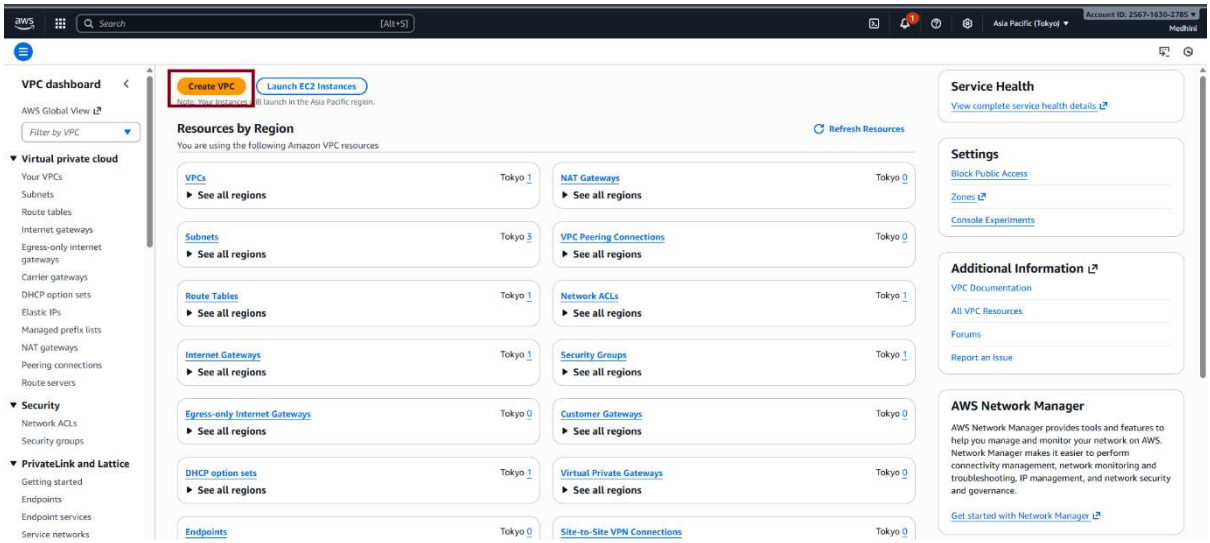
1. Design and build 2-tier architecture with two subnets named web and db and launch instances in both subnets and name them as per the subnet names.
2. Make sure only the web subnet can send internet requests.
3. Create peering connection between production network and development network.
4. Setup connection between db subnets of both production network and development network respectively.

Assignment - VPC

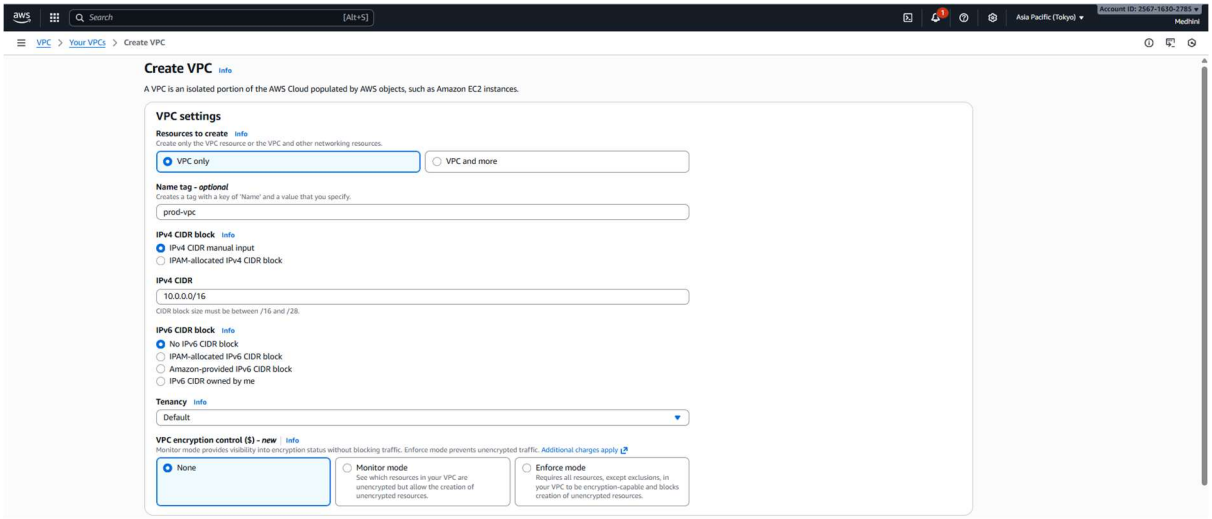
Production Network

Design and build a 4-tier architecture

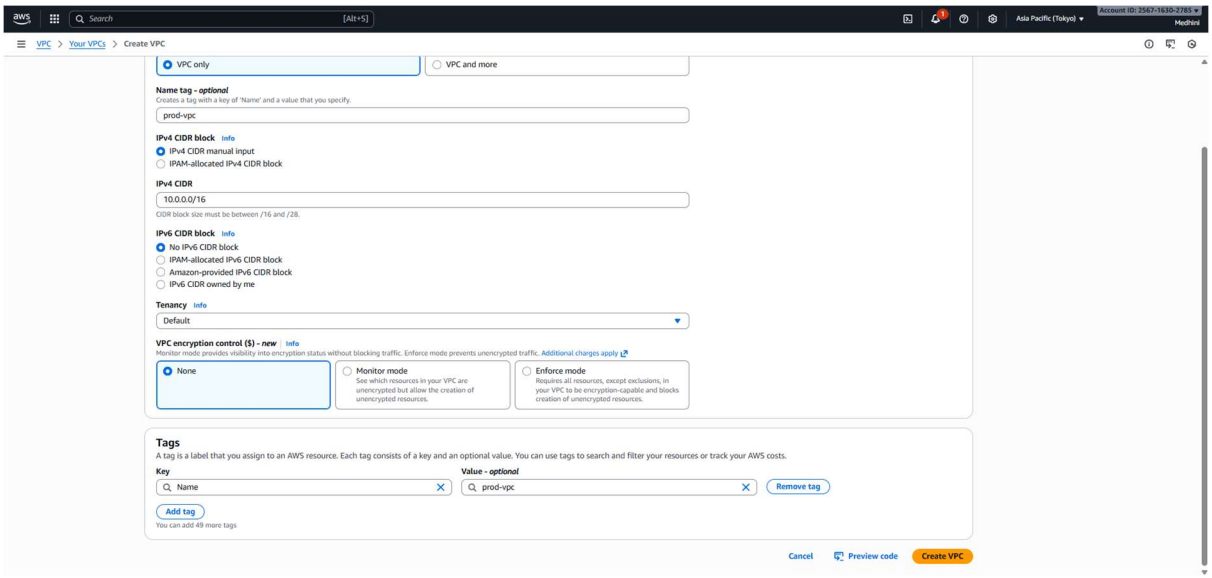
Step 1: Login to account, and Go to VPC, click on **“Create VPC”**

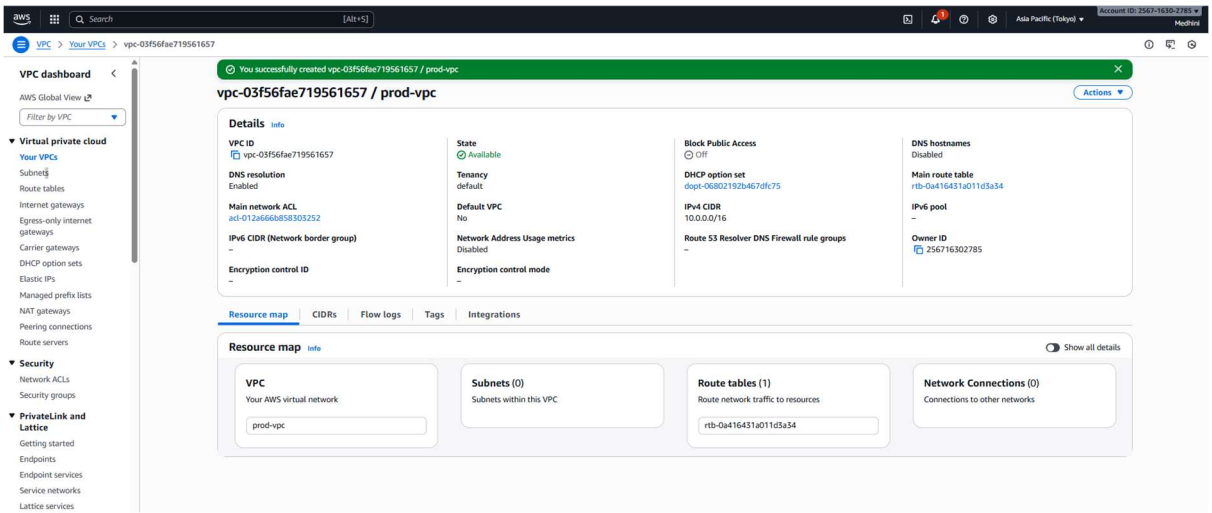


Step 2: Give a name and **IPv4 CIDR block: 10.0.0.0/16. Tenancy: default.**



Assignment - VPC

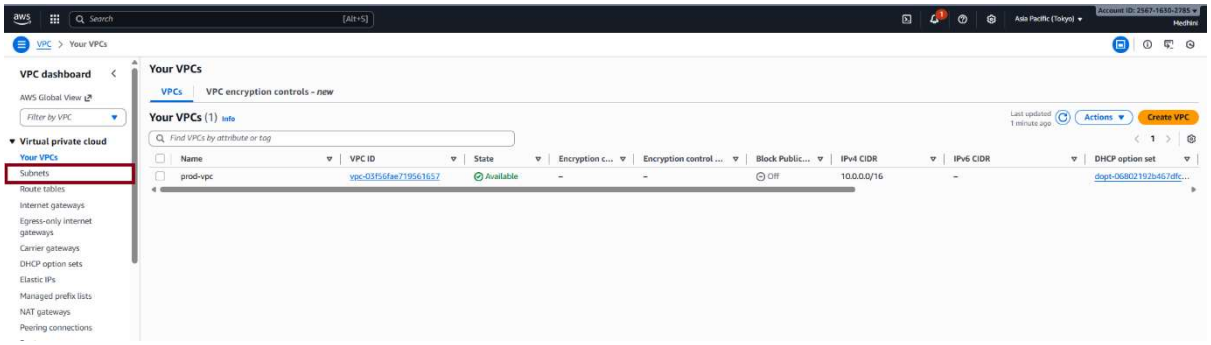




VPC has been created

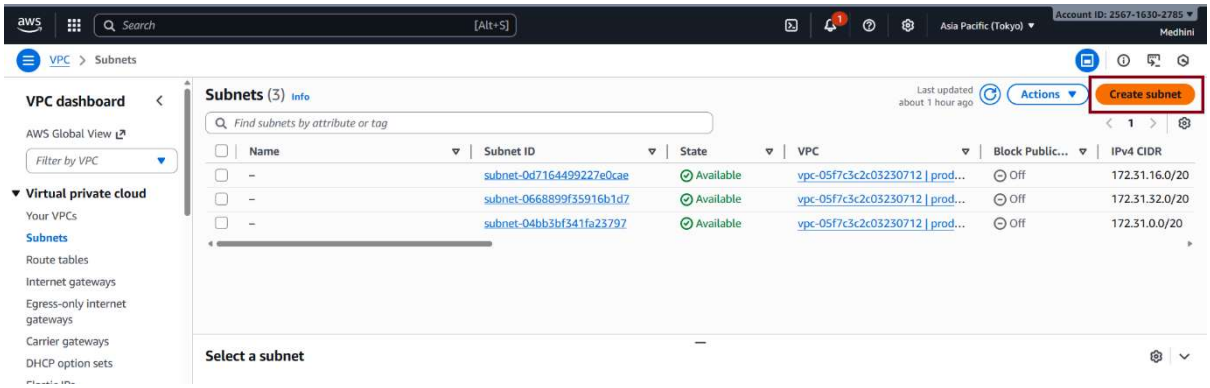
Create 5 subnets out of which 4 should be private named app1, app2, dbcache and db and one should be public, named web.

Step 1: Go to subnets as shown in below picture



Step 2: Click on **“Create Subnet”**

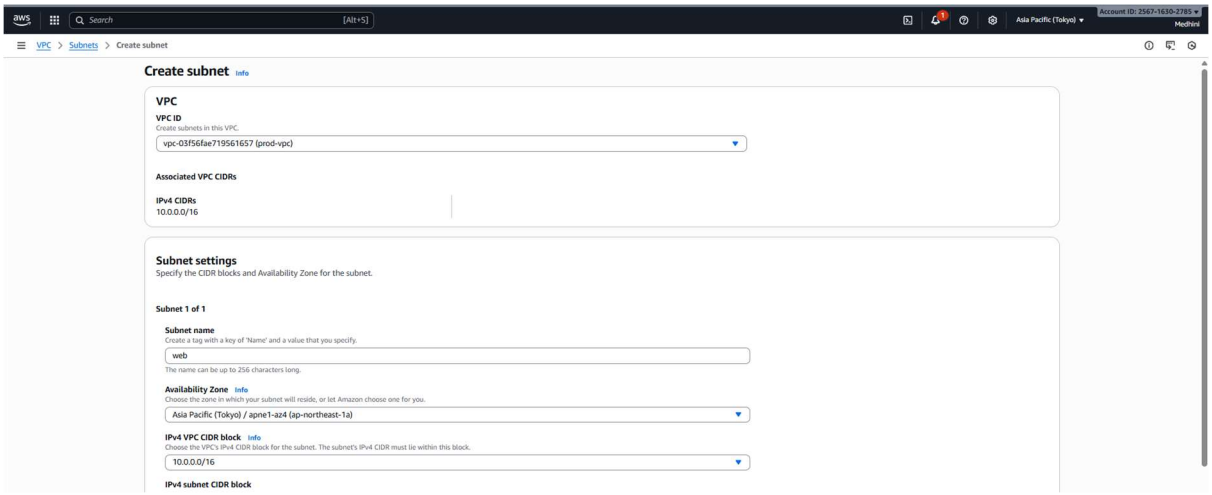
Assignment - VPC



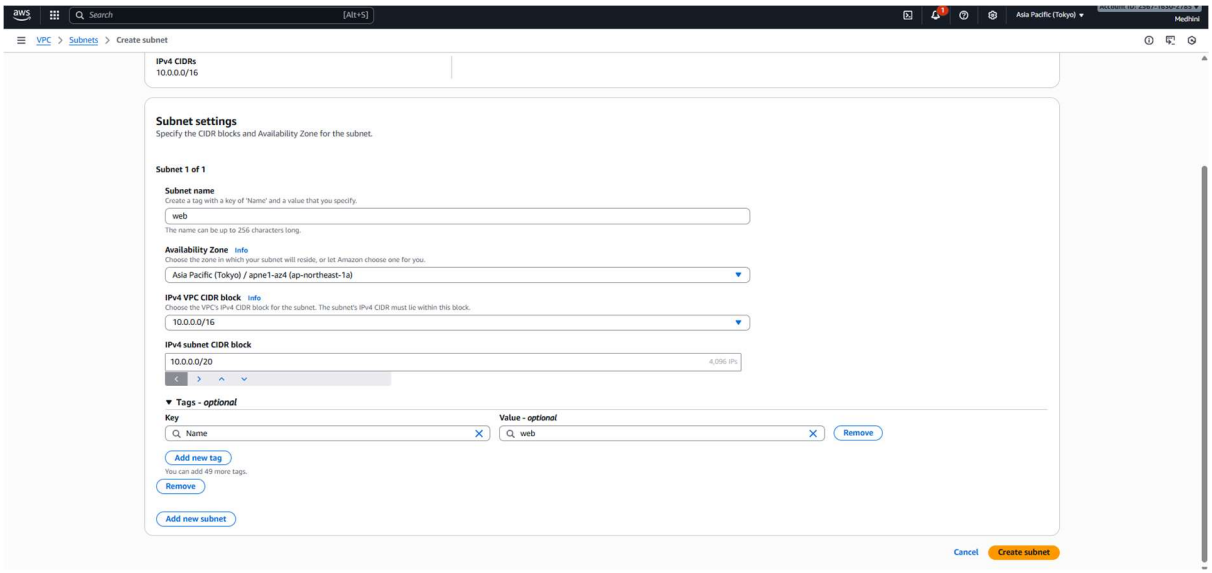
Step 4: Create 5 subnets in prod-vpc each in a different AZ where possible:

- **web (public): 10.0.0.0/24 (AZ-a)**
- **app1 (private with internet): 10.0.1.0/24 (AZ-a or b)**
- **app2 (private): 10.0.2.0/24 (AZ-b)**
- **dbcache (private with internet access for *that* instance): 10.0.3.0/24 (AZ-c)**
- **db (private): 10.0.4.0/24 (AZ-c)**

Choose AZs to distribute availability.

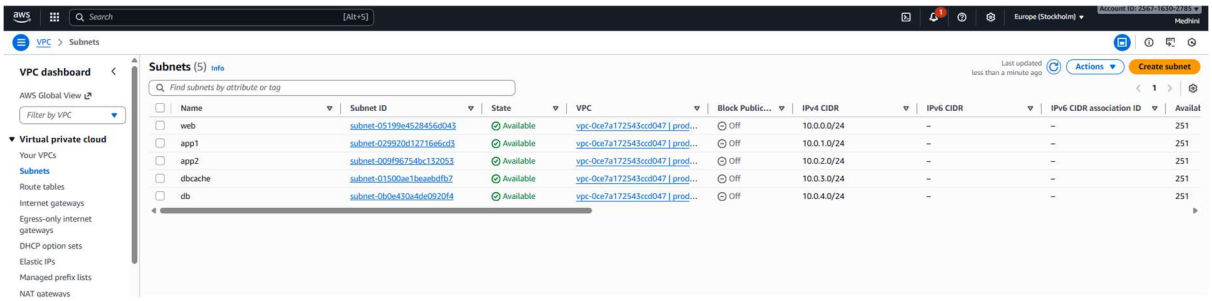


Assignment - VPC



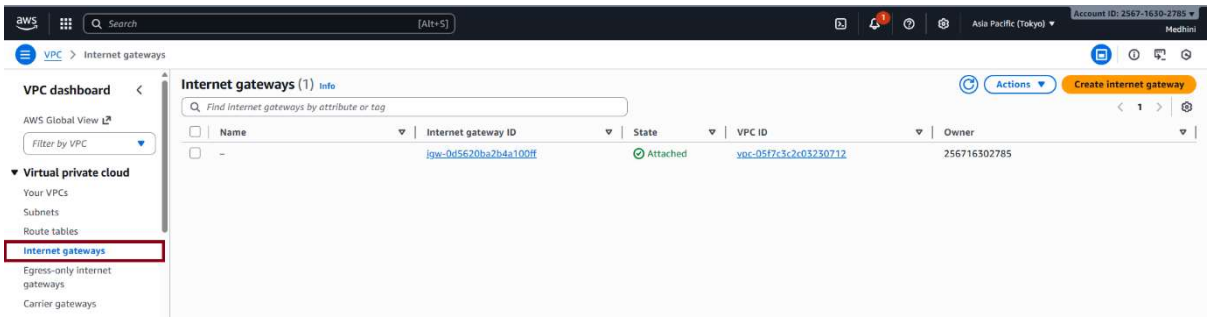
Step 5: Created 5 different subnets

- web (already) = 10.0.0.0/20 (covers 10.0.0.0 - 10.0.15.255)
- app1 = 10.0.16.0/24
- app2 = 10.0.17.0/24 ← use this one
- dbcache = 10.0.18.0/24
- db = 10.0.19.0/24



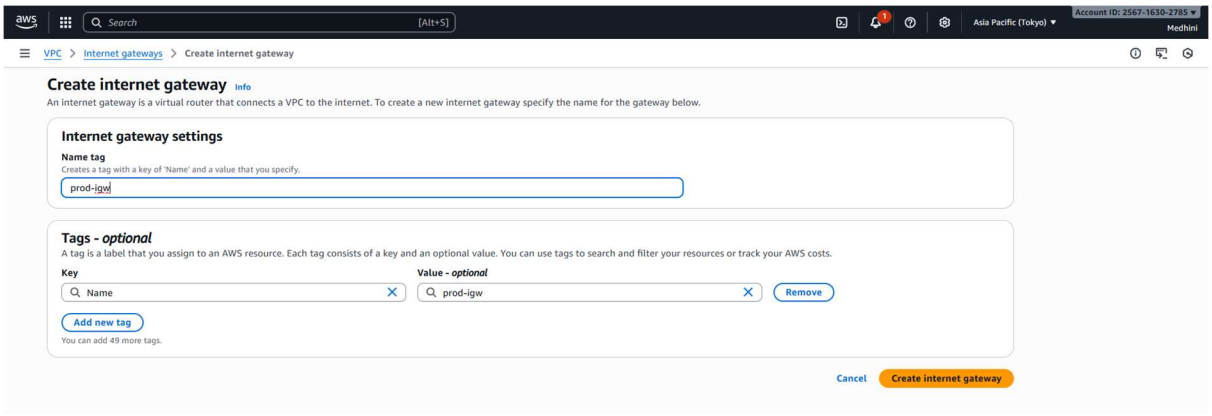
Create an Internet Gateway and attach it

Step 1: Click on Internet gateways

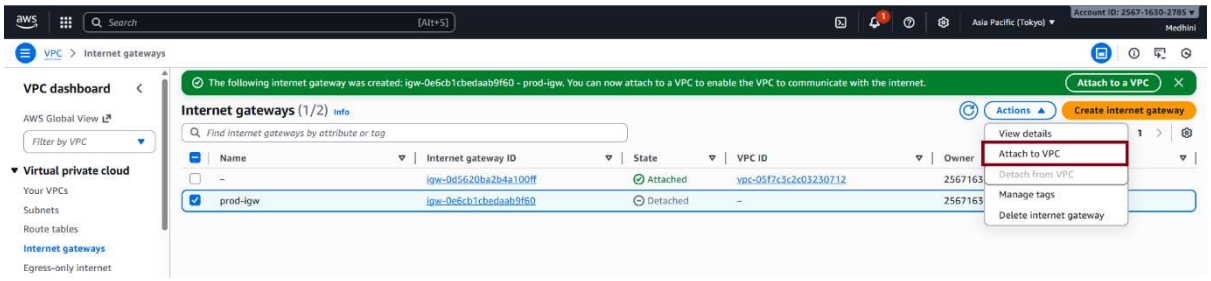


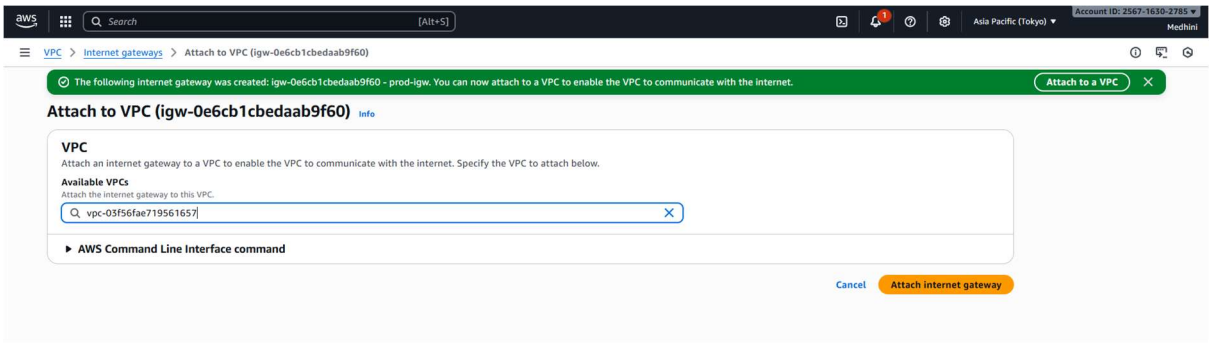
Step 2: Give a name

Assignment - VPC



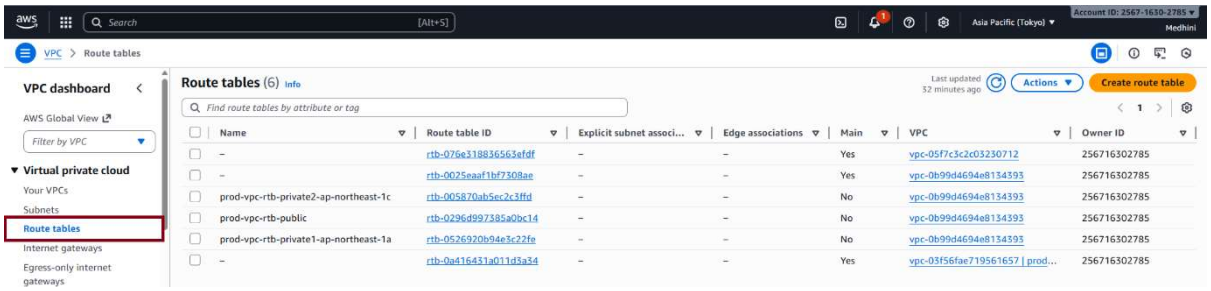
Step 3: Attach **prod-igw to prod-vpc.**





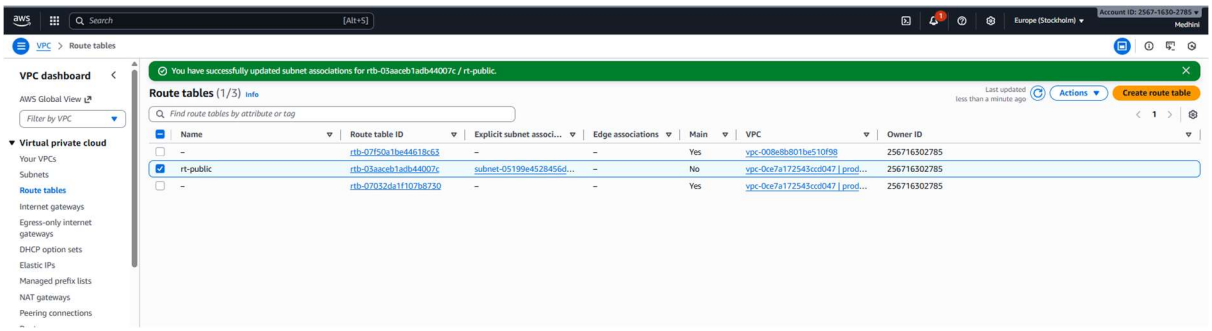
Route tables for Prod

Step 1: Click on **Route Tables**

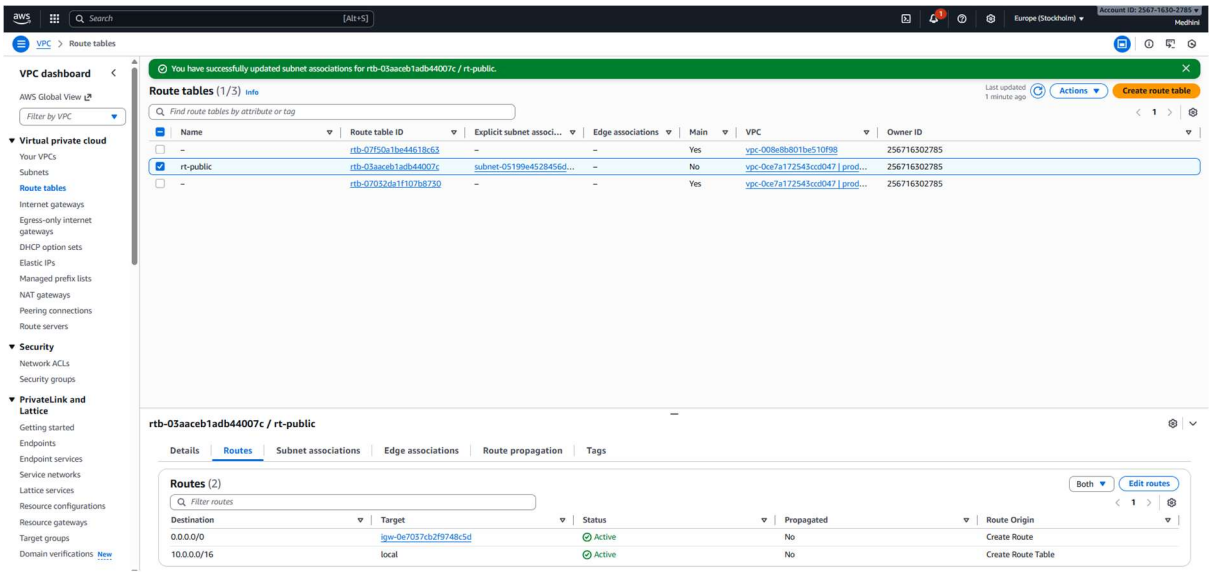


Step 2: Create route table

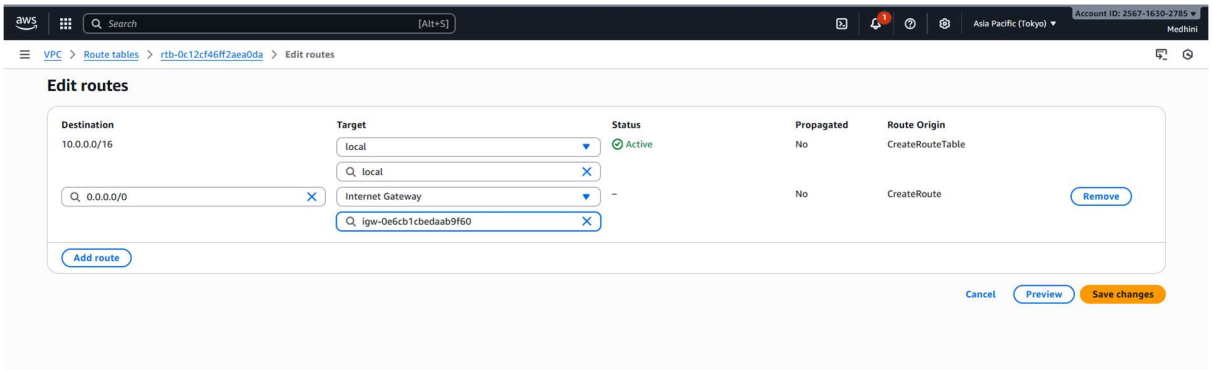
Assignment - VPC



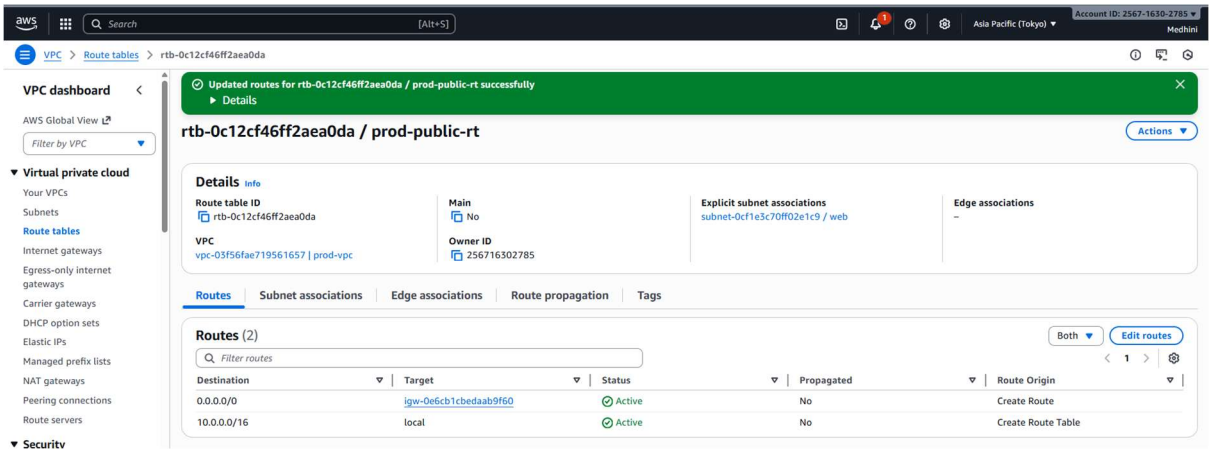
Step 3: Associate the **web** subnet to new route table



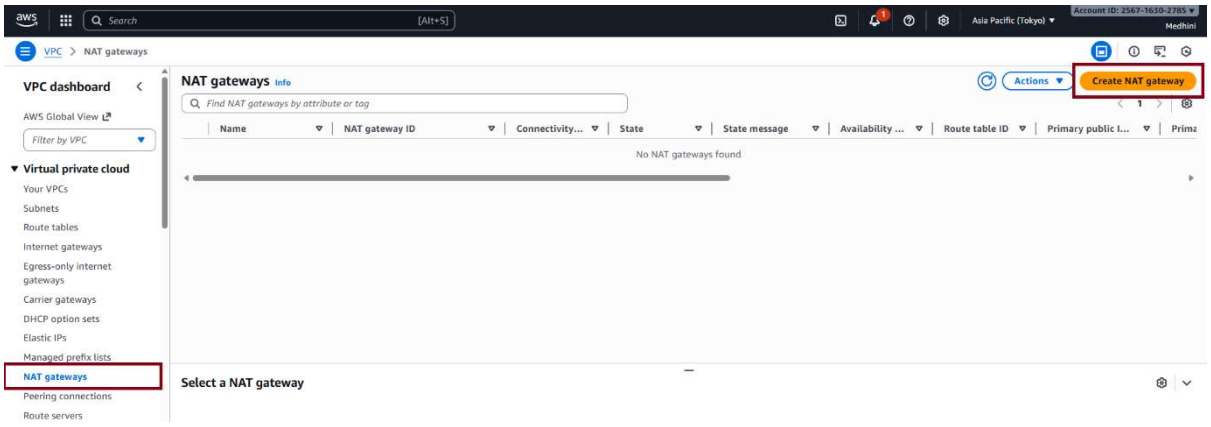
Step 4: Add route 0.0.0.0/0 → prod-igw

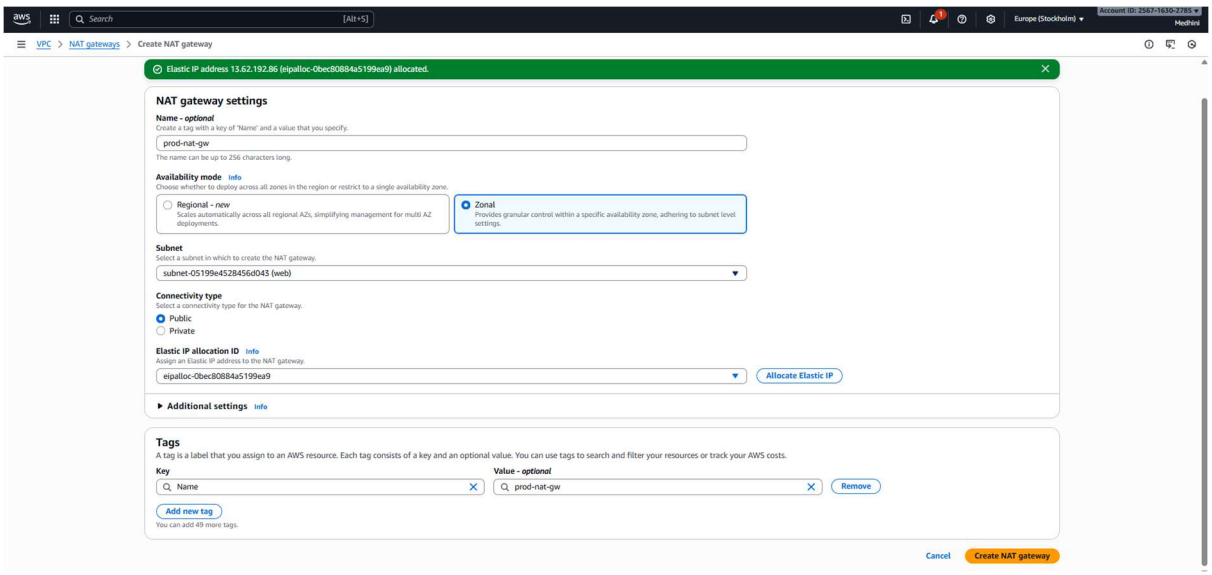


Assignment - VPC



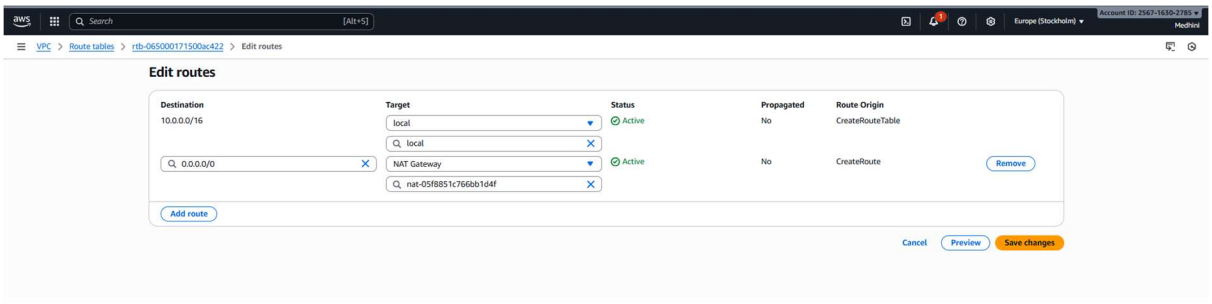
Step 5: Create Nat gateway



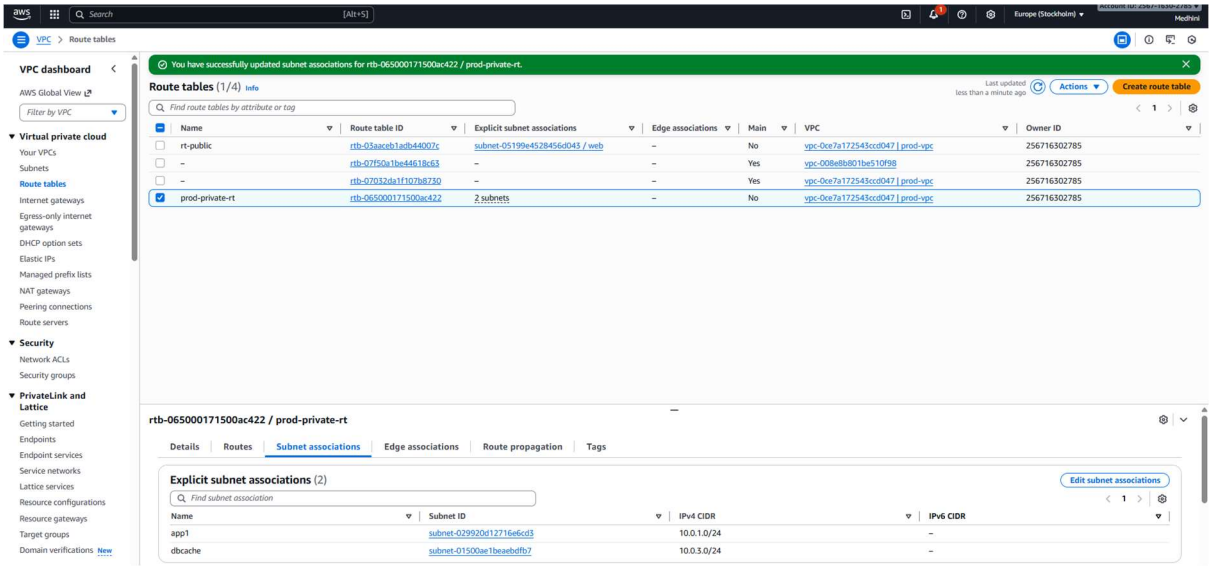


Edit routes

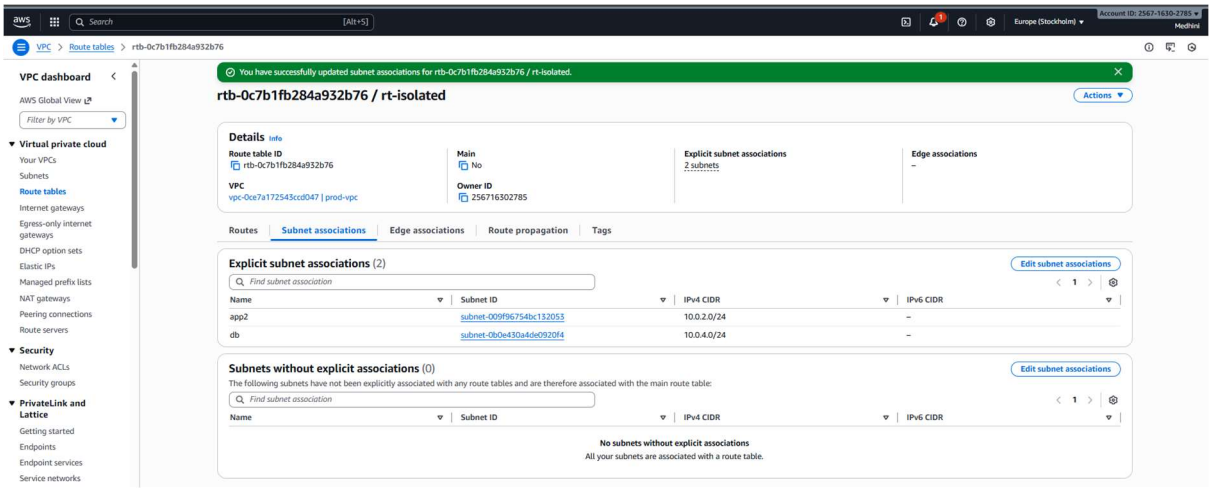
Assignment - VPC



Associate app1 and dbcache to



Isolated route table is associated with app2 and db

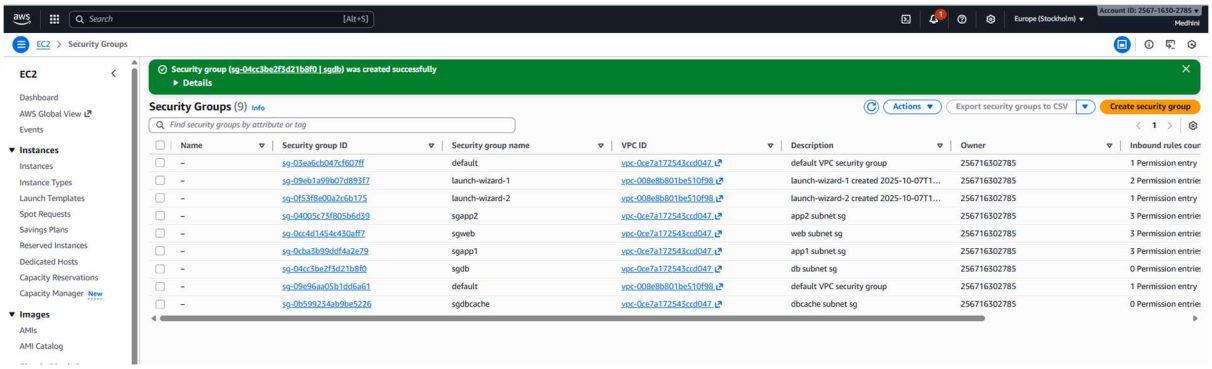


Assignment - VPC

Launch instances in all subnets and name them as per the subnet that they have been launched in

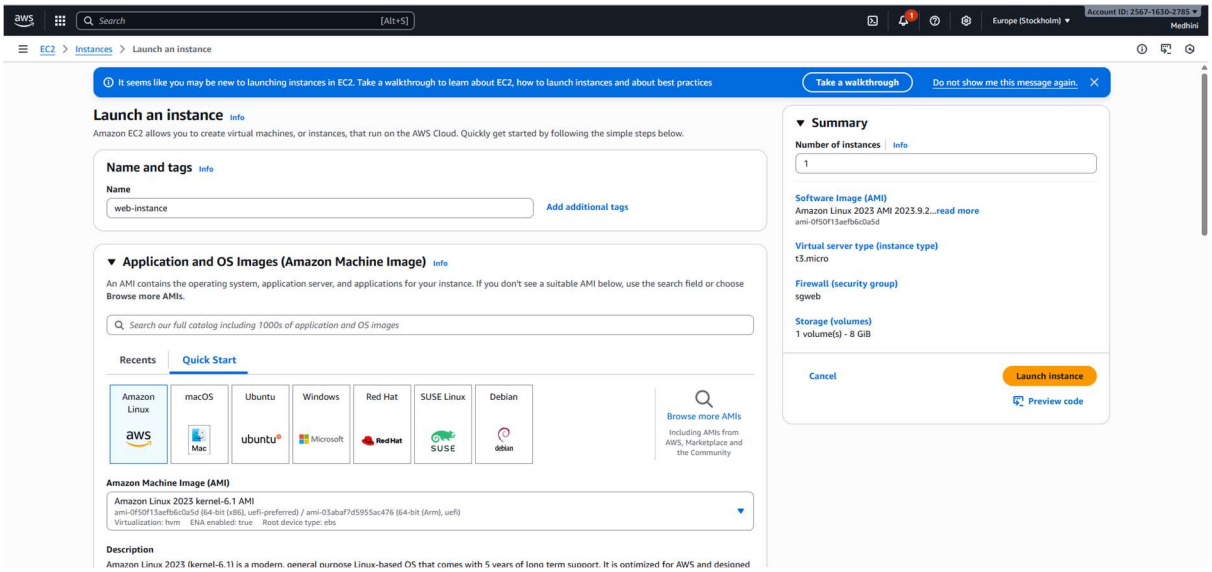
Instance	Subnet	Public IP?	Notes
web-instance	Public	Yes	Internet-facing
app1-instance	Private	No	NAT gateway gives outbound internet
dbcache-instance	Private	No	NAT gateway gives outbound internet
app2-instance	Private	No	Must stay internal
db-instance	Private	No	Never expose DB

Step 1: Create Security groups

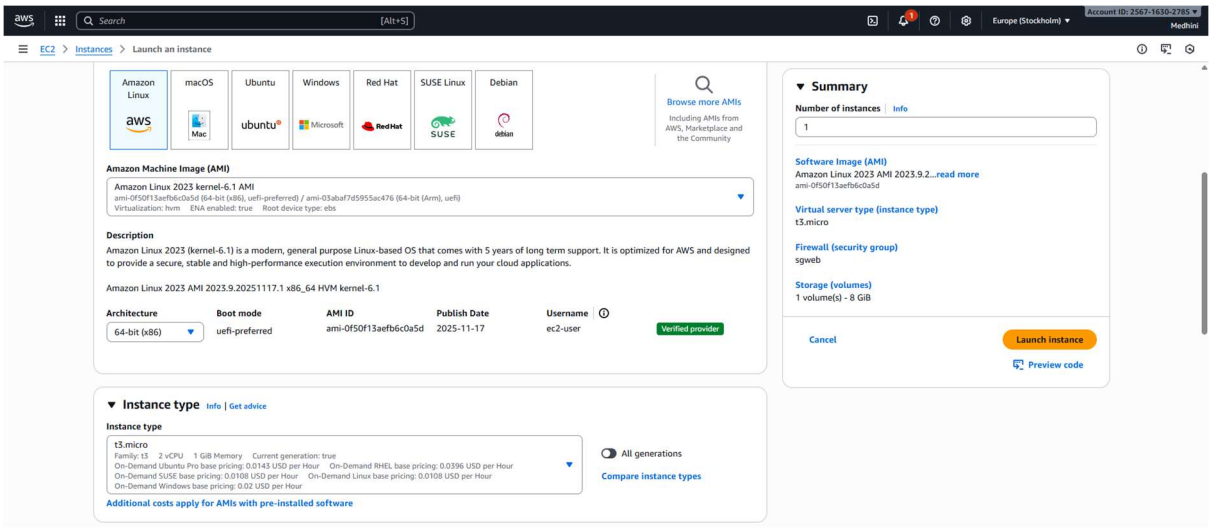


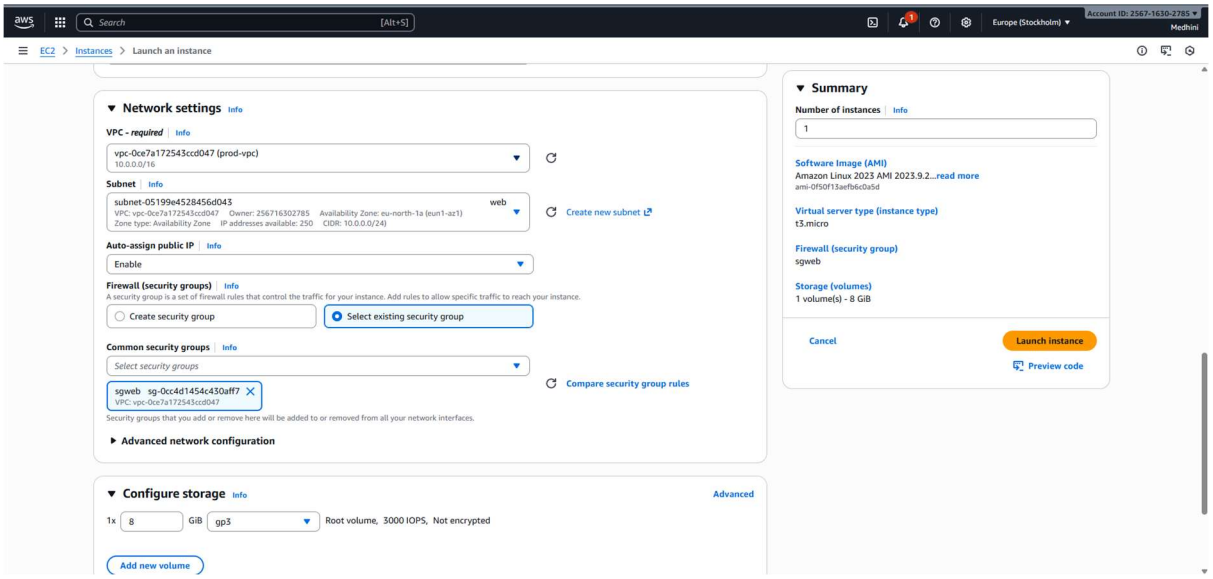
Step 2: Create keypairs and instances for all subnets

Web instance

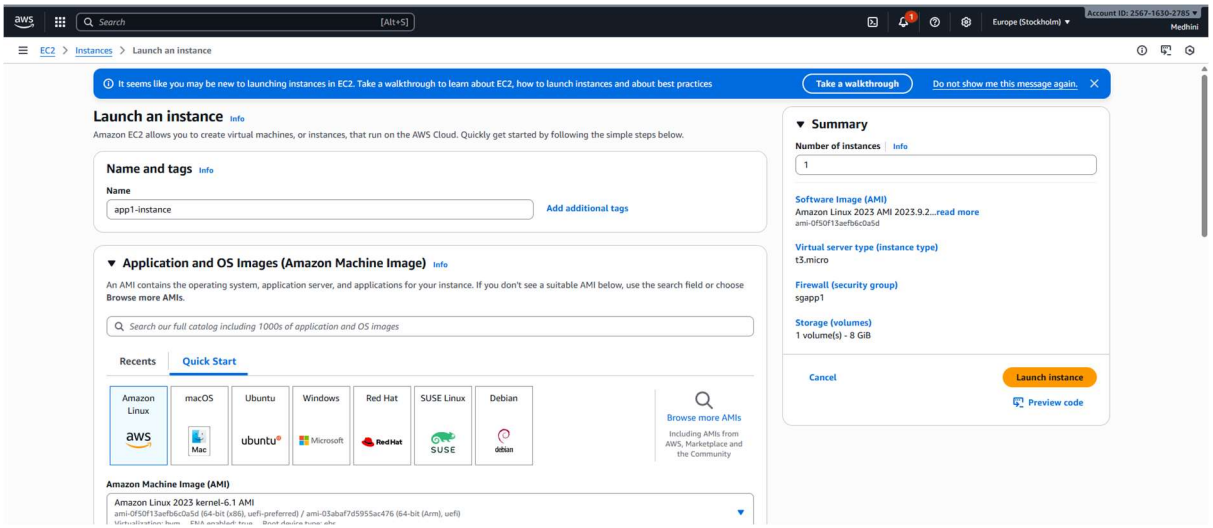


Assignment - VPC

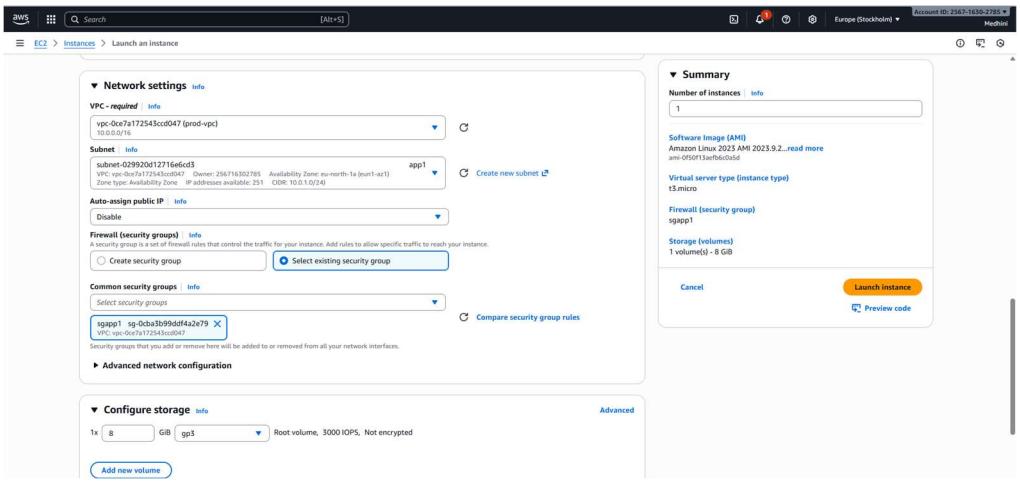




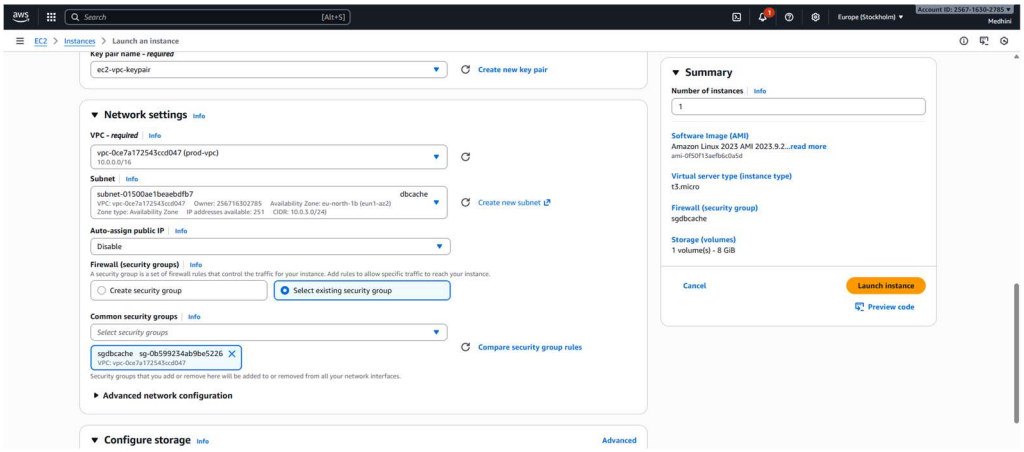
App1 Instance



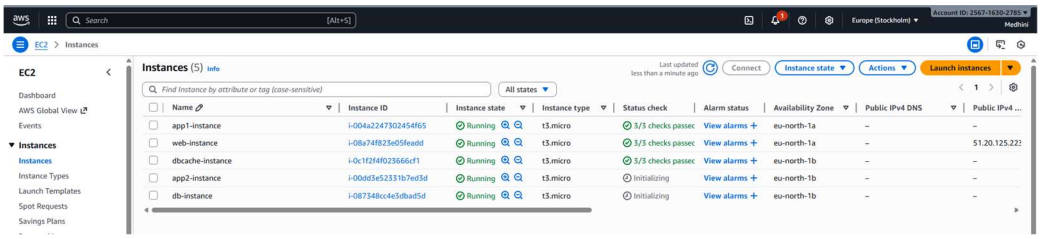
Assignment - VPC



Dbcache instance



All instances are created

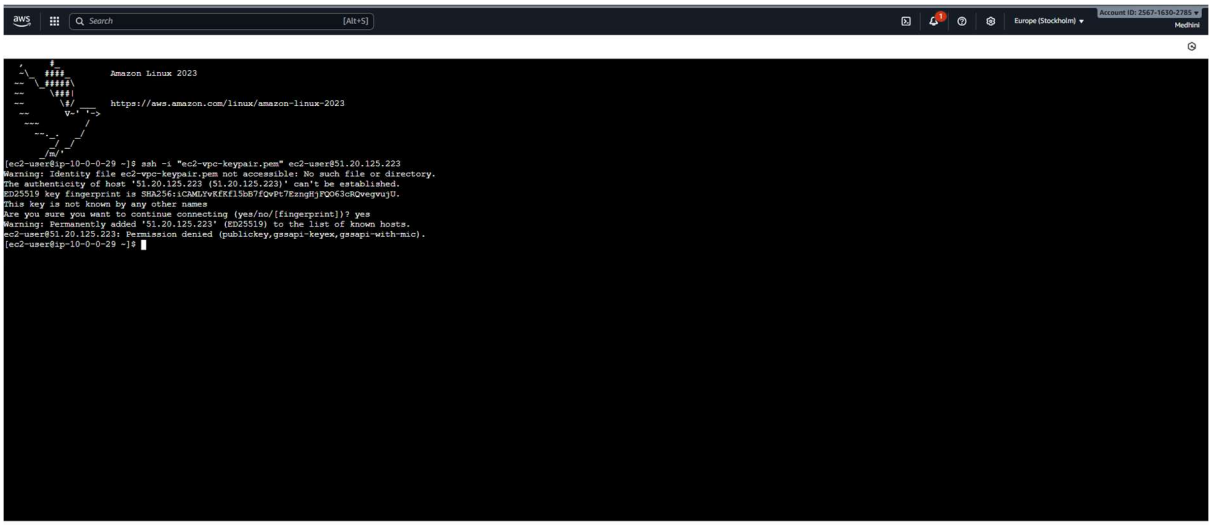


Result

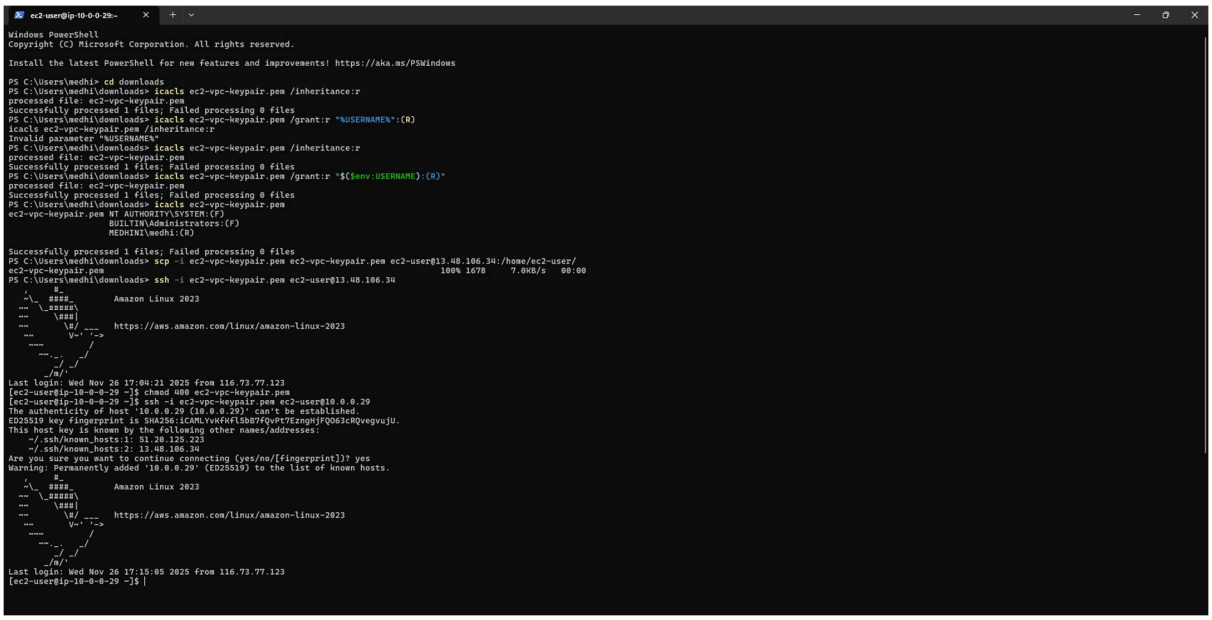
Instance	Subnet	Public IP	Internet?
web-instance	web	YES	Direct
app1	app1	NO	Through NAT
dbcache	dbcache	NO	Through NAT
app2	app2	NO	❌ (isolated)
db	db	NO	❌ (isolated)

Assignment - VPC

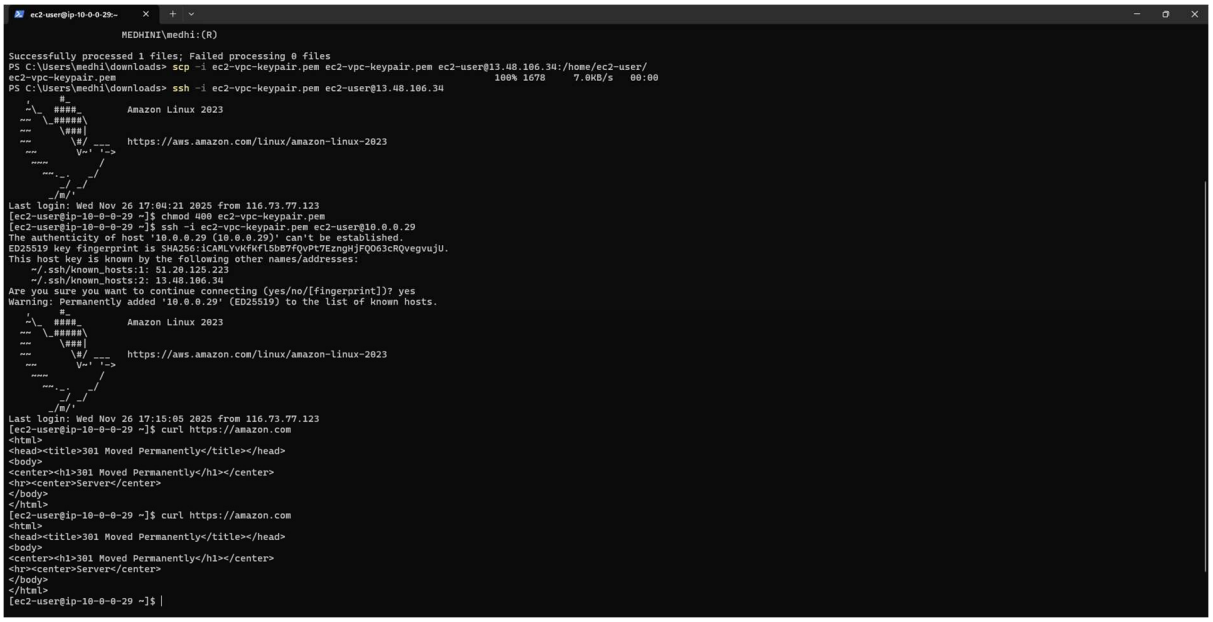
Web-instance



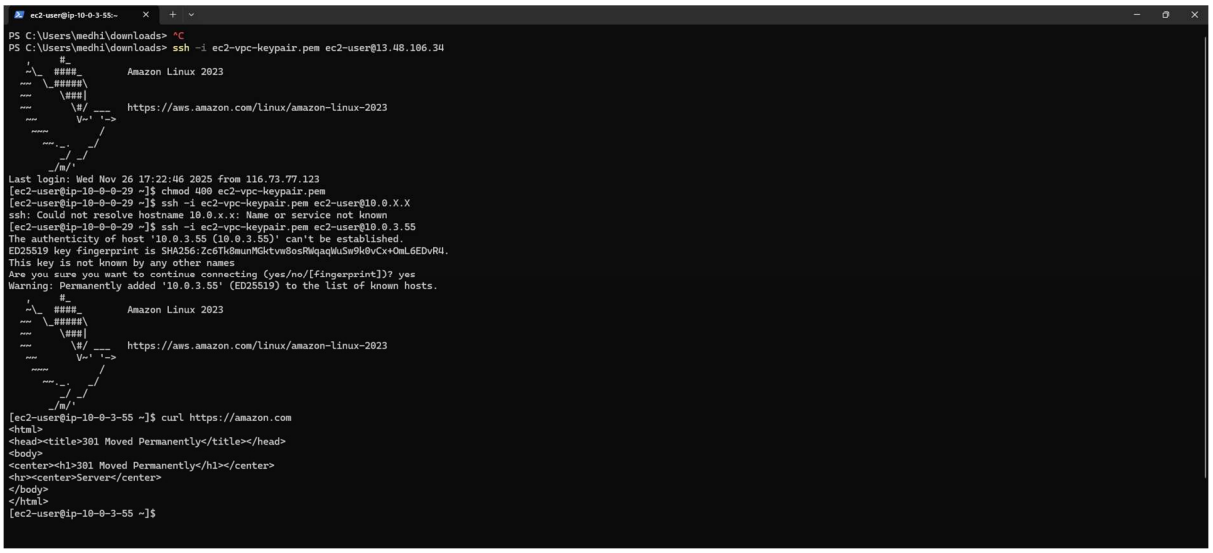
App1 verification is completed as shown in below picture *(app1 should connect from your laptop using keypair)*



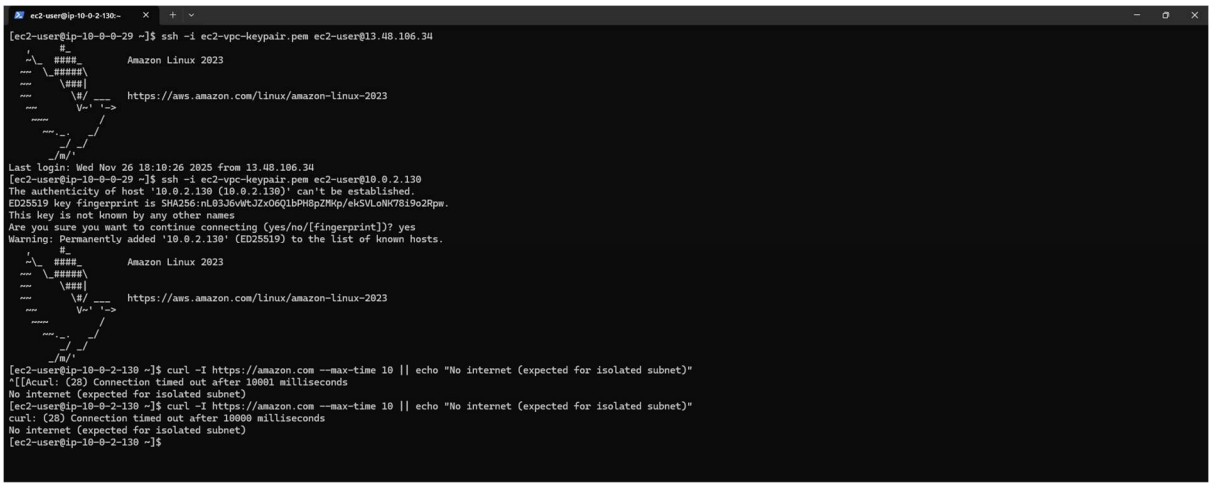
Assignment - VPC



Dbcache has been verified

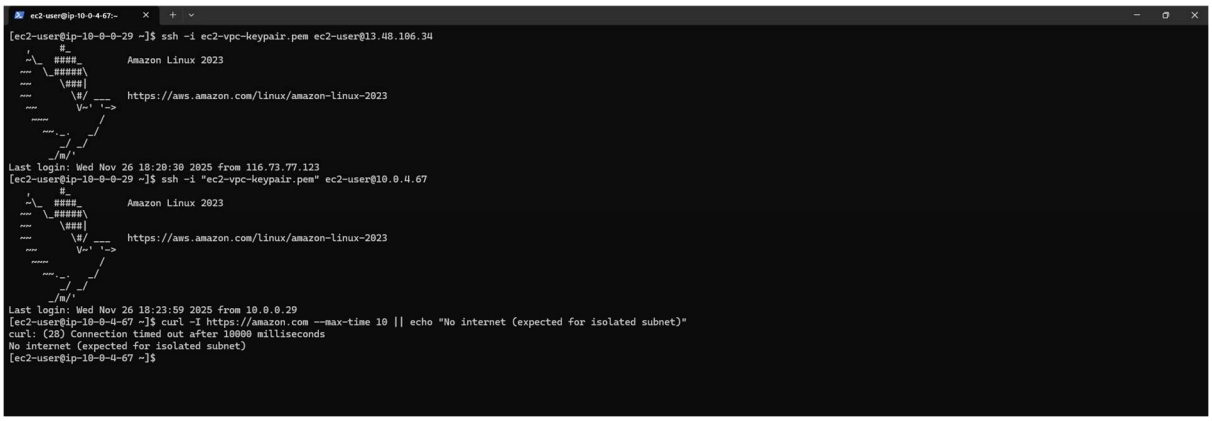


App2 verification completed



Assignment - VPC

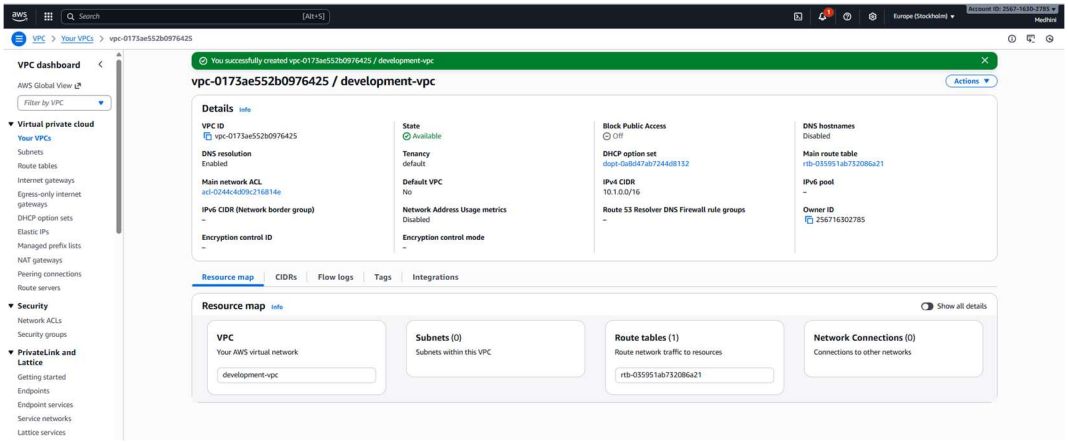
db verification has been completed



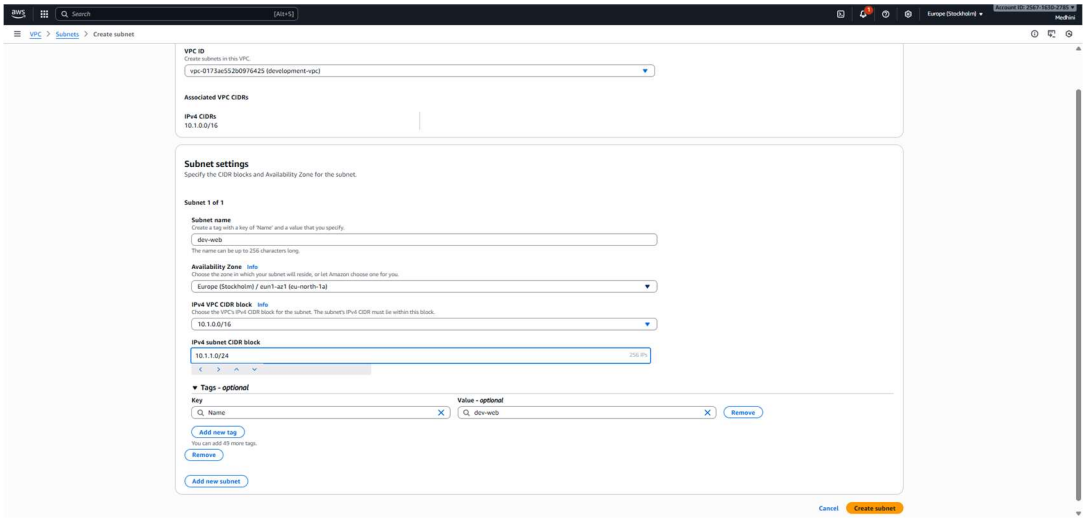
Development Network

Design and build 2-tier architecture with two subnets named web and db and launch instances in both subnets and name them as per the subnet names.

Step 1: Create vpc

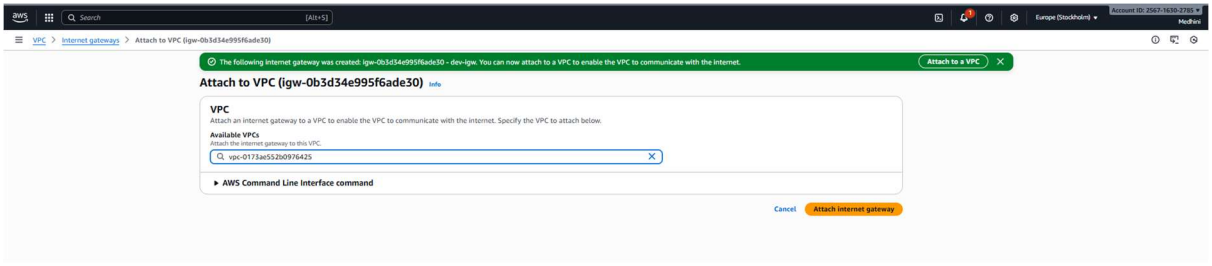


Step 2: Create public subnet and private subnet



Assignment - VPC

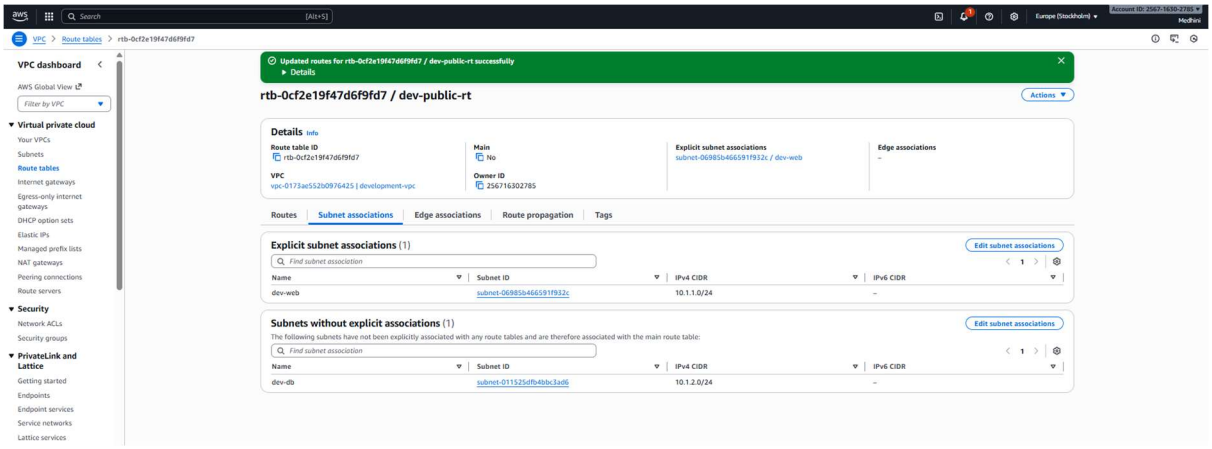
Step 3: Create Internet gateway and Attach it to dev vpc



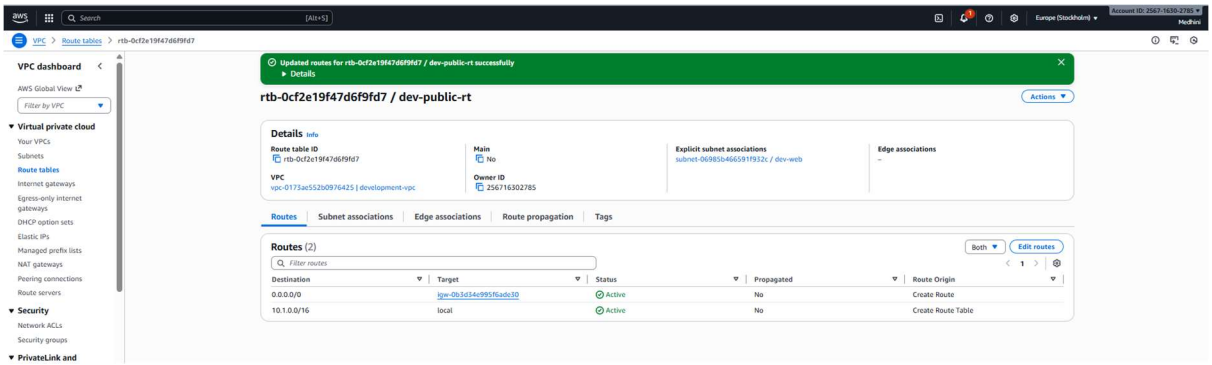
Step 4: Create **route table**

Public Route Table (for web subnet)

- *Name: dev-public-rt*
- *Associate subnets:*
  - *dev-web*
- *Add route:*
  - *0.0.0.0/0 → Internet Gateway (dev-igw)*



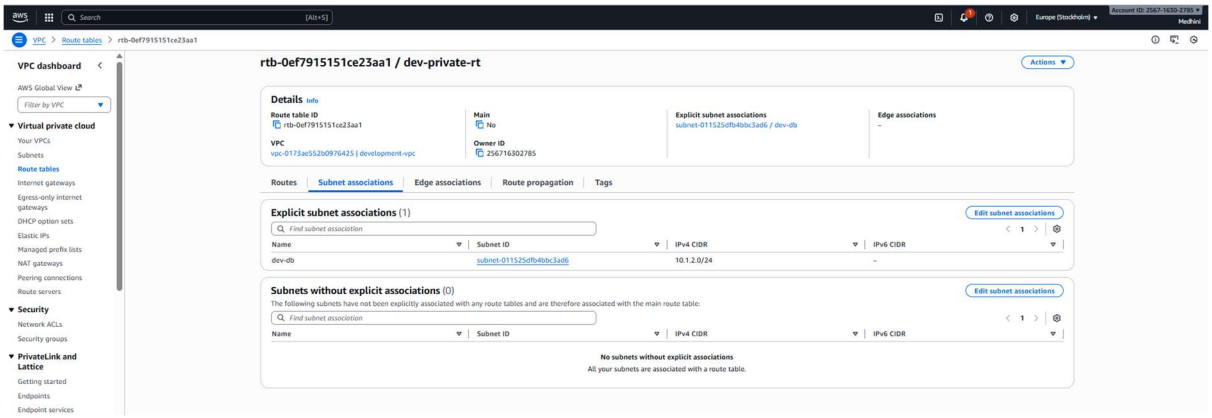
Add inbound rule

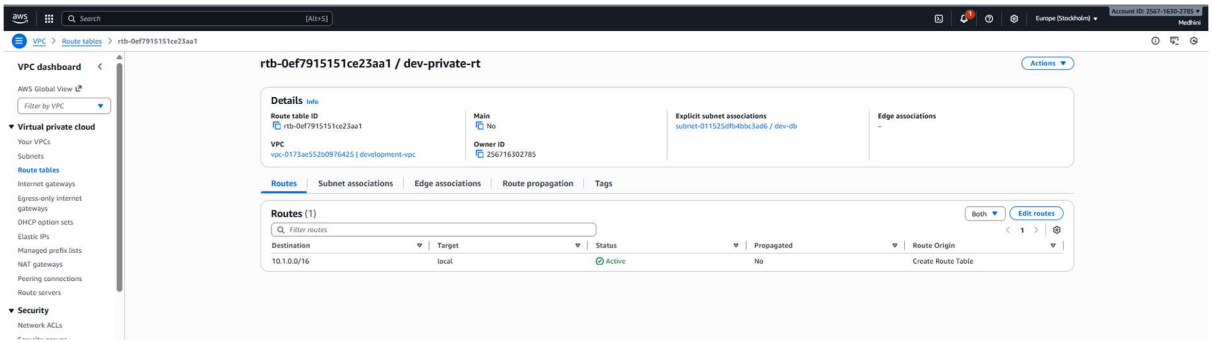


Assignment - VPC

Private Route Table (for db subnet)

- *Name: dev-private-rt*
- *Associate subnet:*
  - *dev-db*
- *No internet route*
  - *Only local VPC traffic*



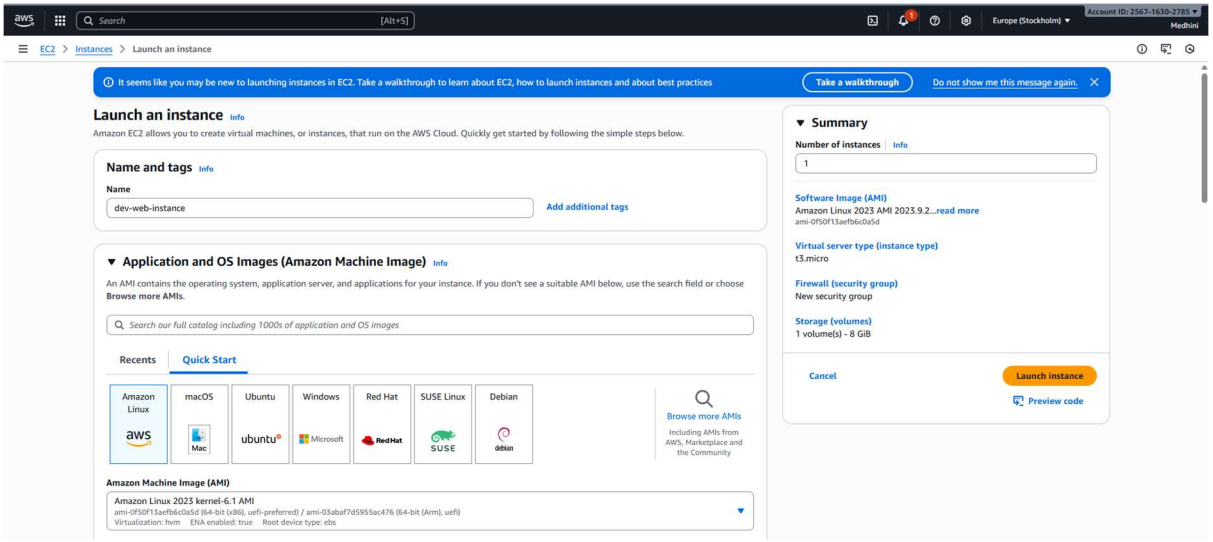


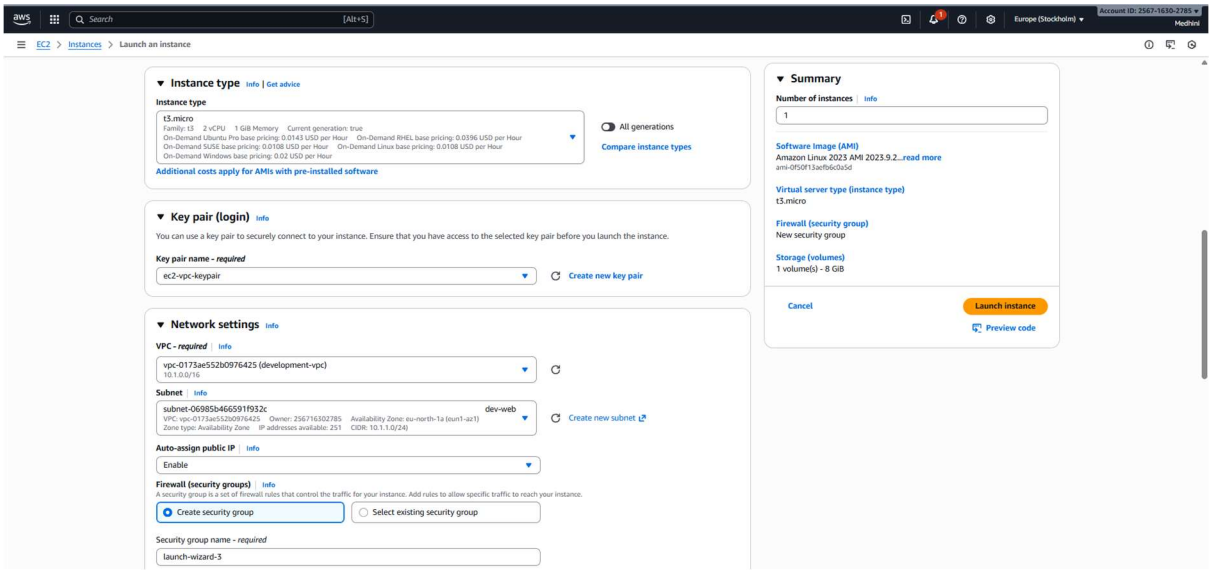
Step 4: Launch EC2 Instances

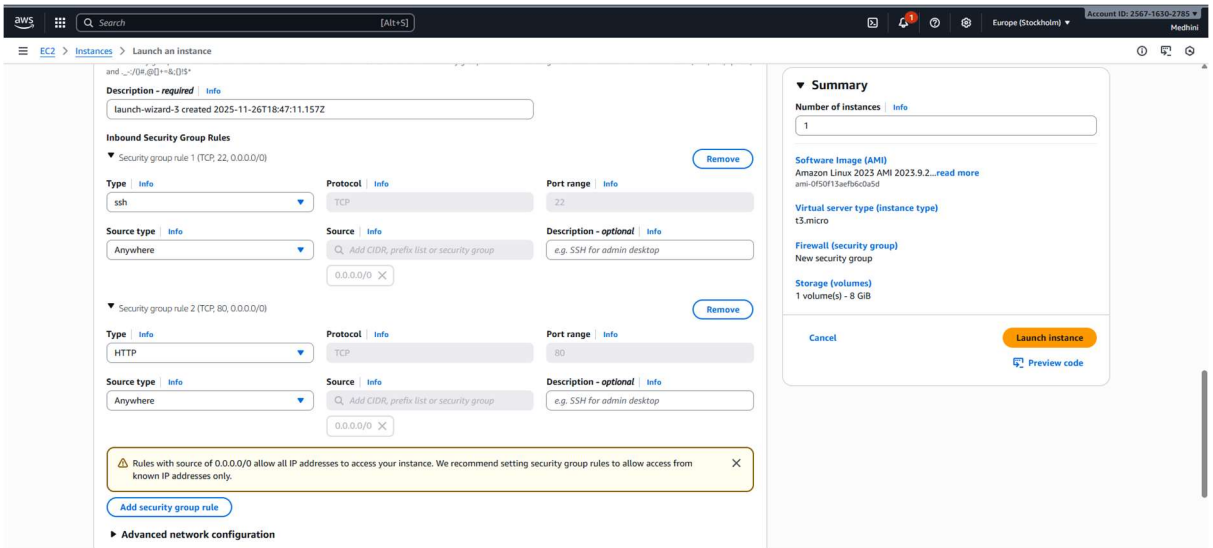
For web subnet

- ***Subnet: dev-web***
- ***Security Group: allow SSH + HTTP***
- ***Enable Auto-assign Public IP***
- ***Name: dev-web-instance***

Assignment - VPC



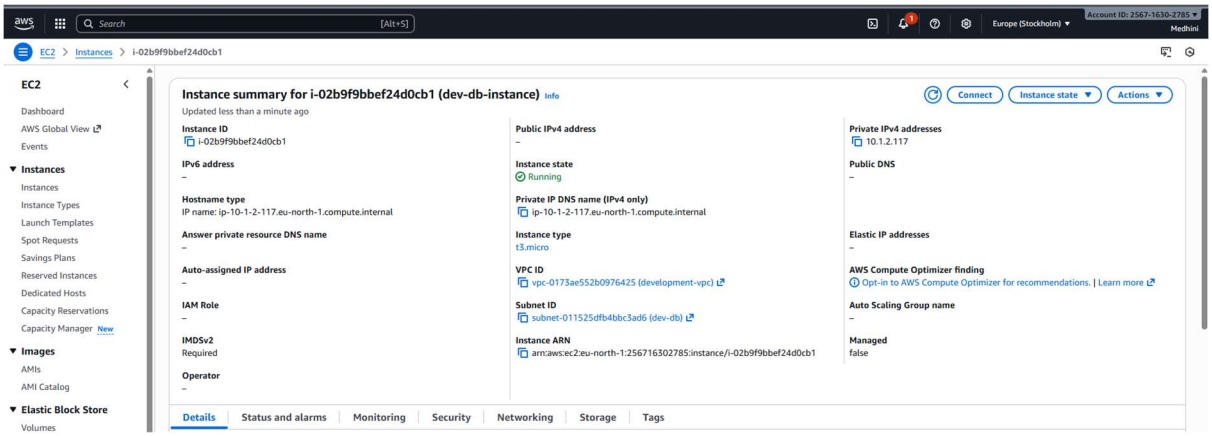




Assignment - VPC

For db subnet

- **Subnet: dev-db**
- **No Public IP**
- **Security Group:**
  - **Allow MySQL/Postgres from Production-DB subnet only**
- **Name: dev-db-instance**

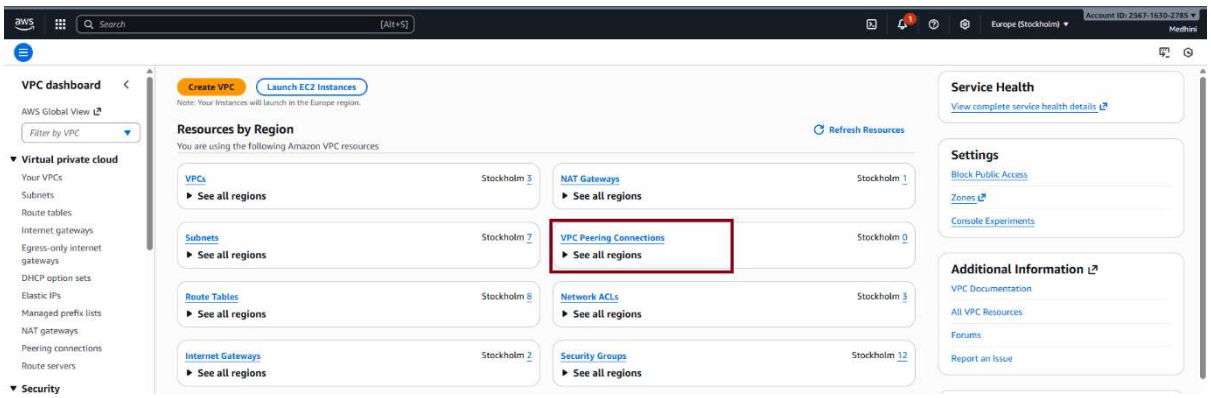


STEP 6 : VPC Peering (Prod ↔ Dev)

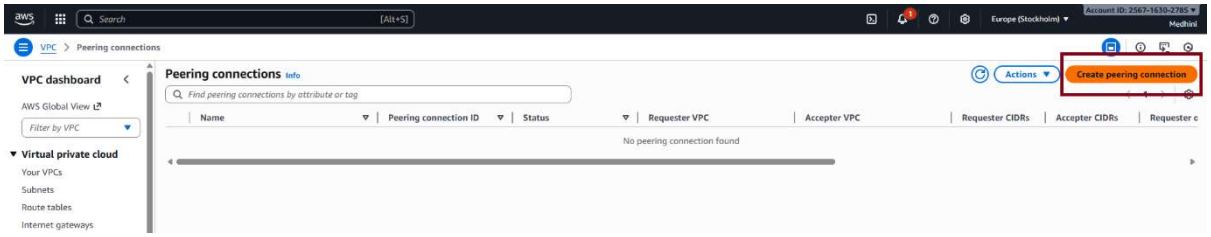
On the AWS console → VPC Peering

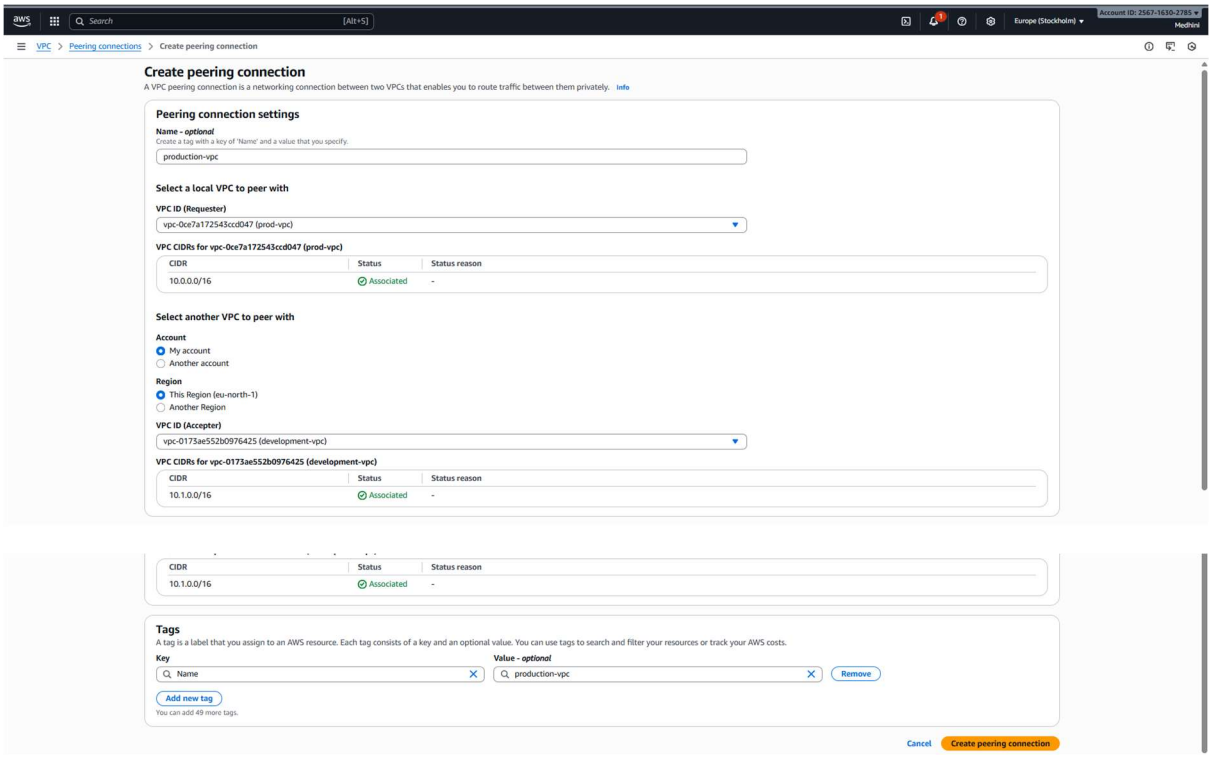
1. *Choose Production VPC as requester*
   → production-vpc
2. *Choose Development VPC as accepter*
   → development-vpc
3. *Accept the request in the Development VPC*

Click on **VPC Peering connections**

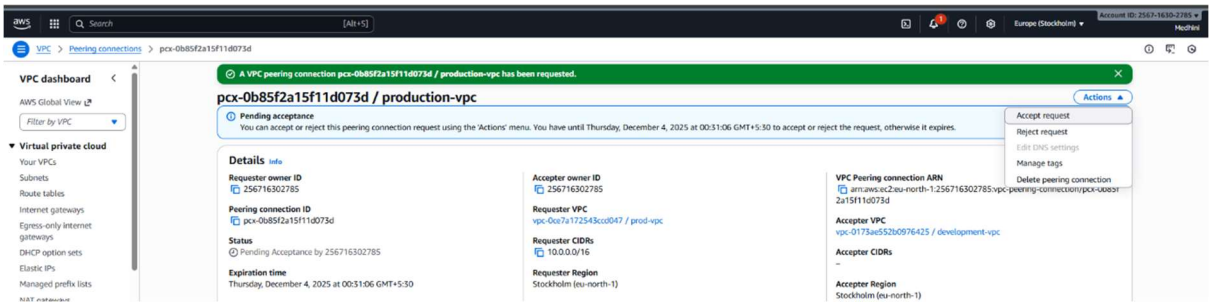


Assignment - VPC



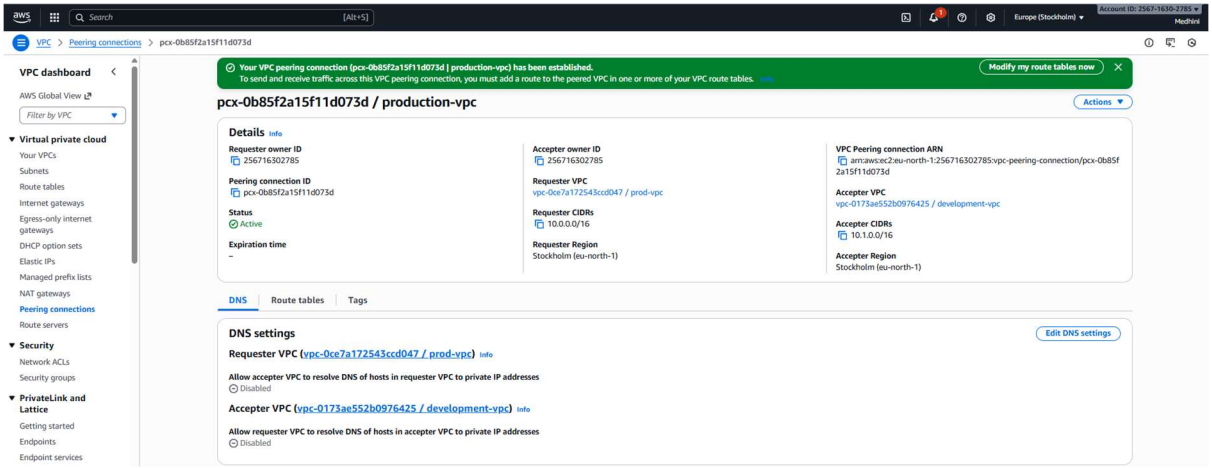


Accept the Request

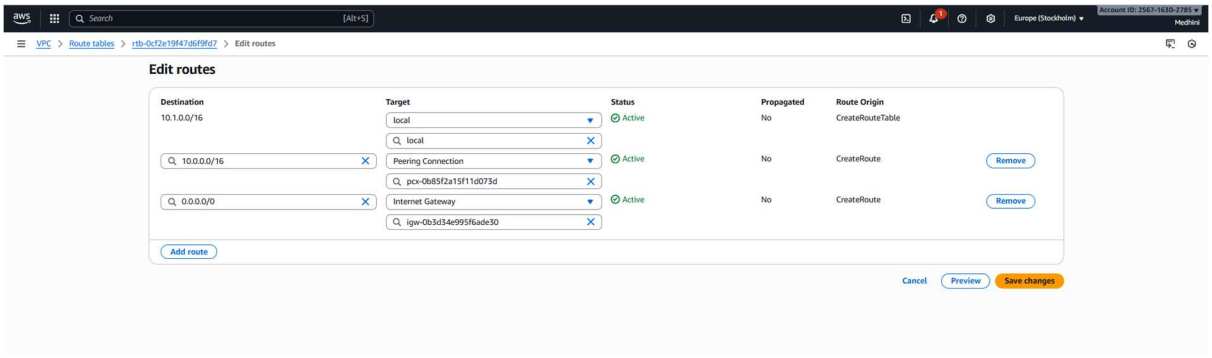


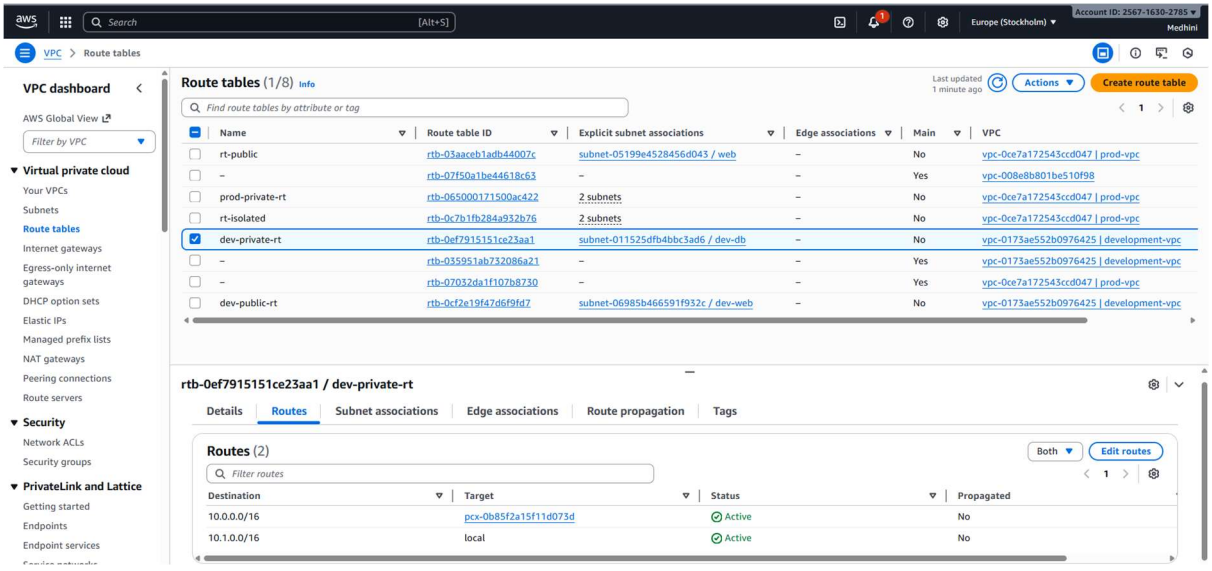
Connect has been established once you accept the request

Assignment - VPC

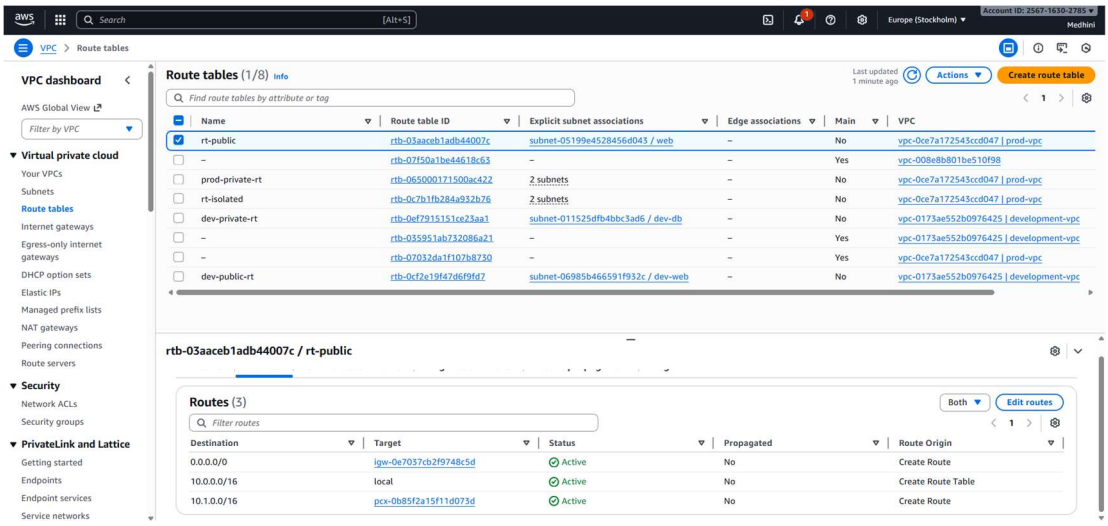


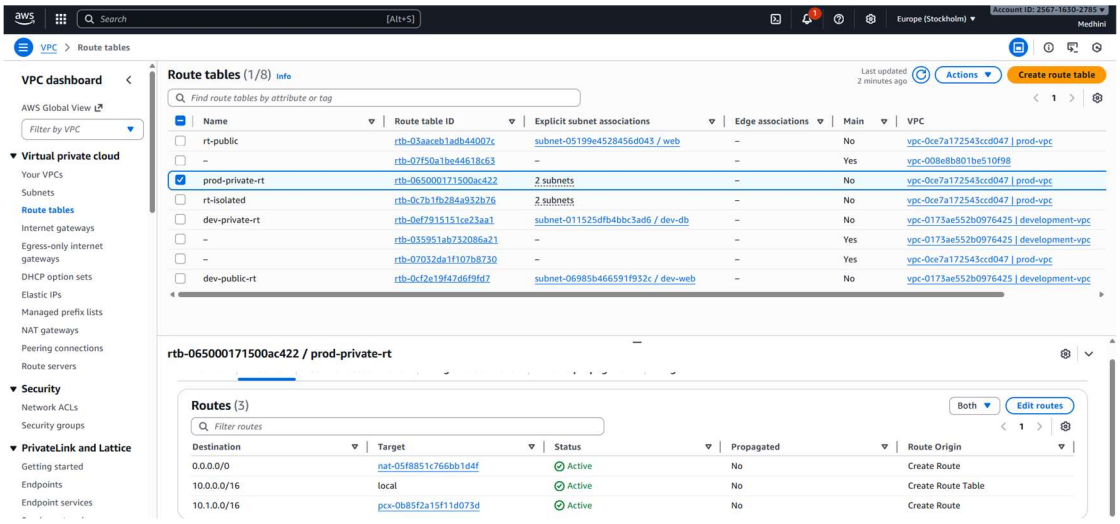
Set the route table

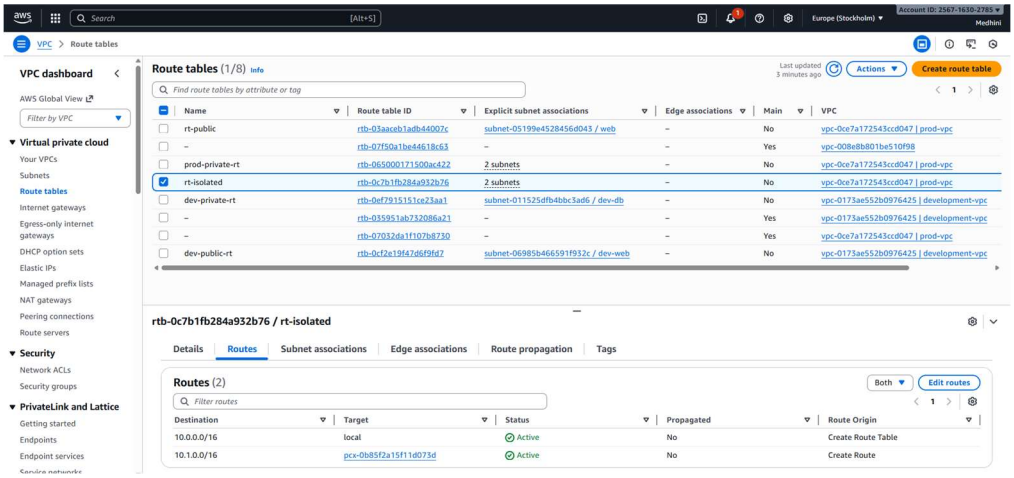




Assignment - VPC

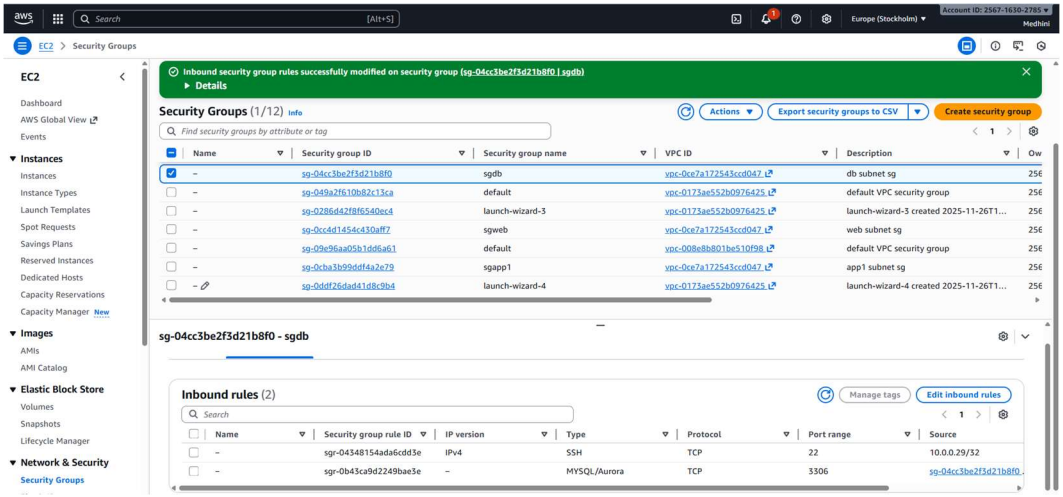




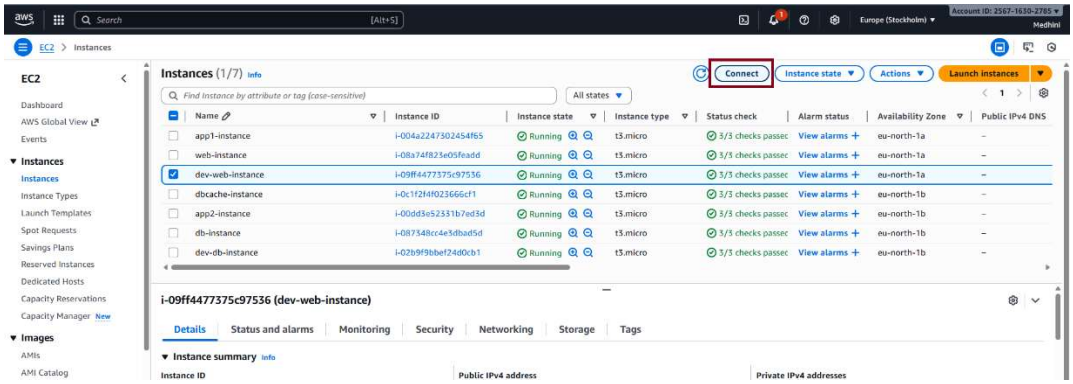


Assignment - VPC

Edit inbound rules for both production sg and development sg

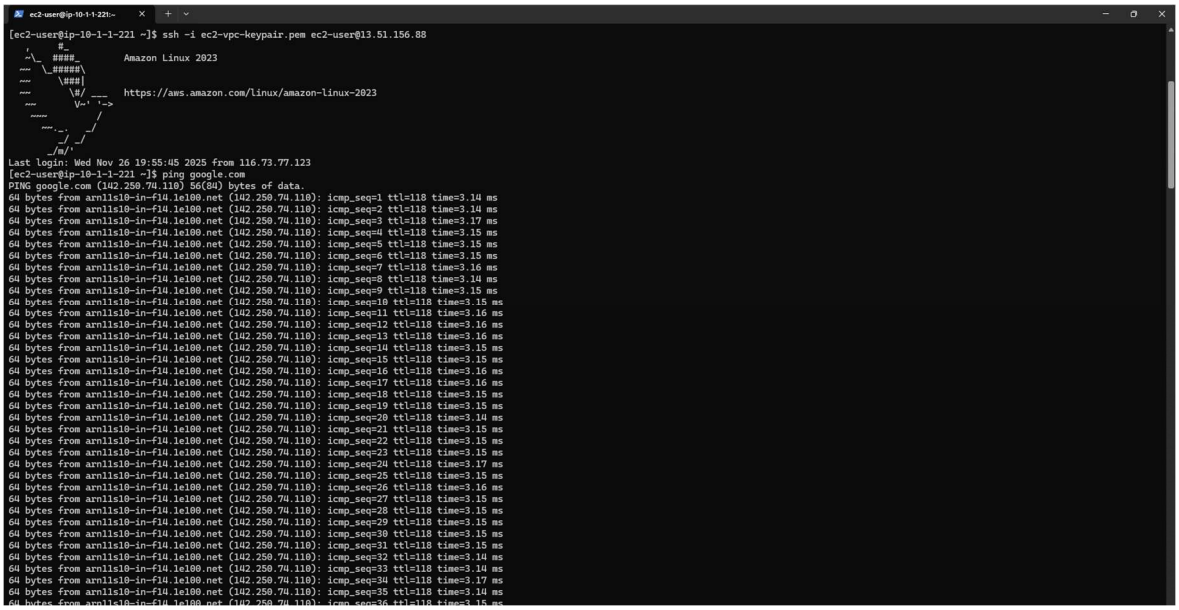


Run Instances



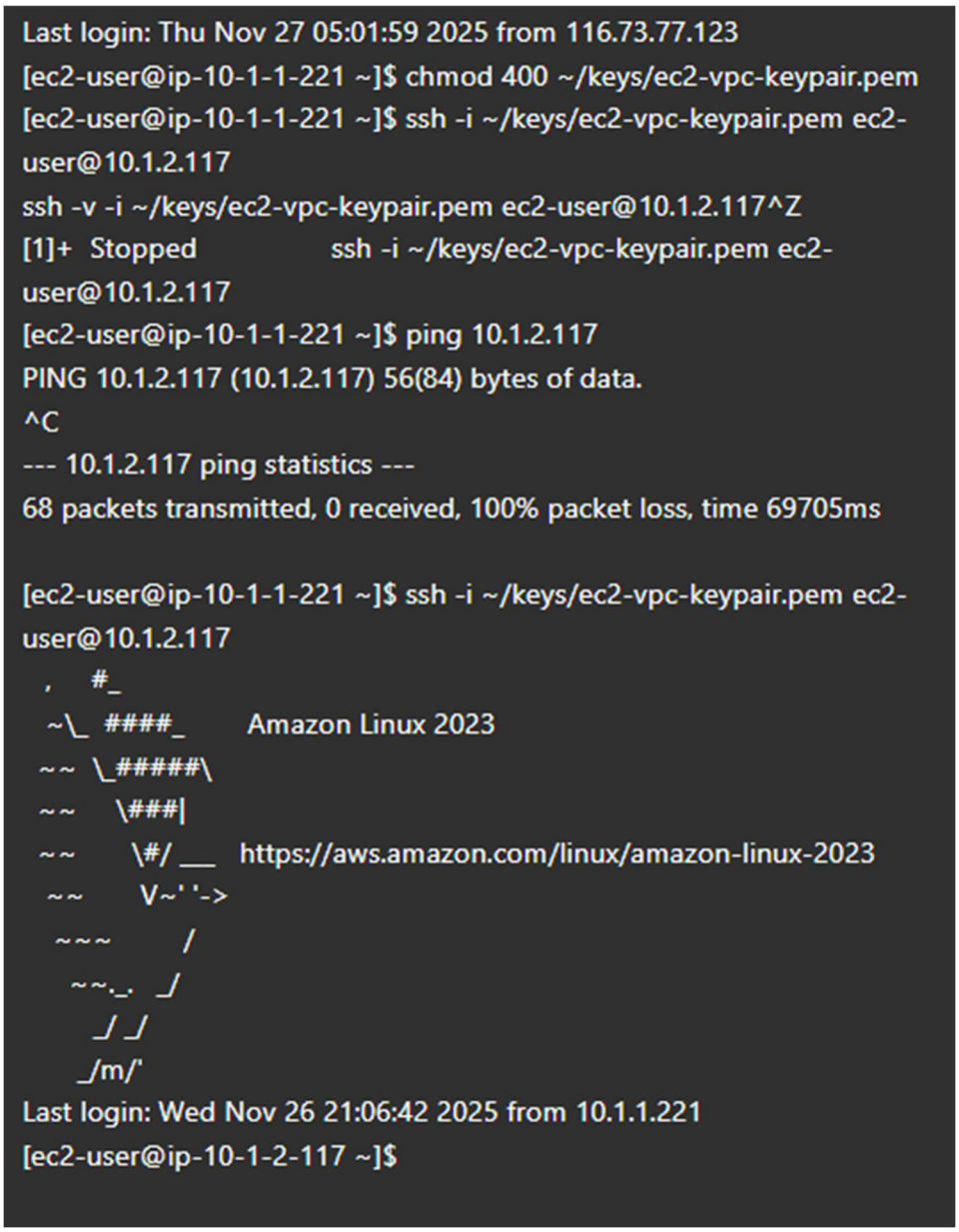
Logged into **dev-web EC2 (Amazon Linux)**

Verified it has **Internet access** (by pinging google.com)



Assignment - VPC

Verified the dev and prod instances



This confirms:

- SSH connection from bastion - private EC2 works.
- The network (SGs, ACLs, routing) between bastion and private EC2 is correctly configured.