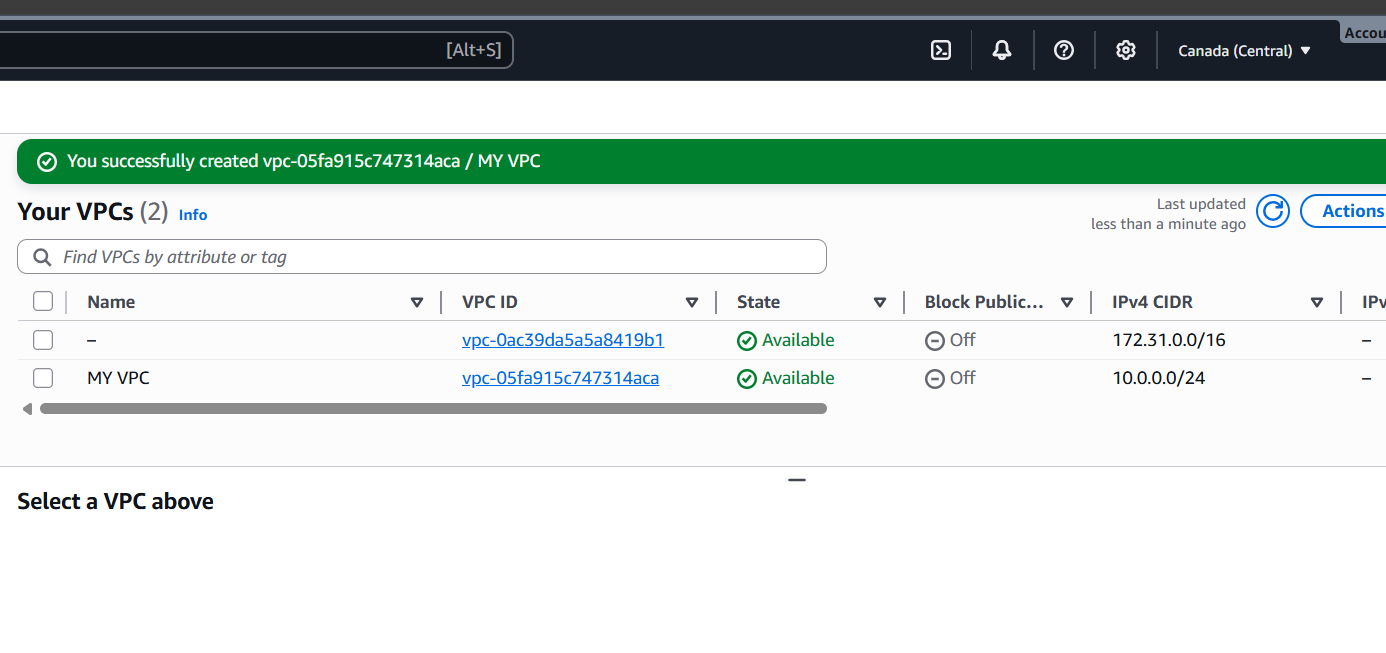
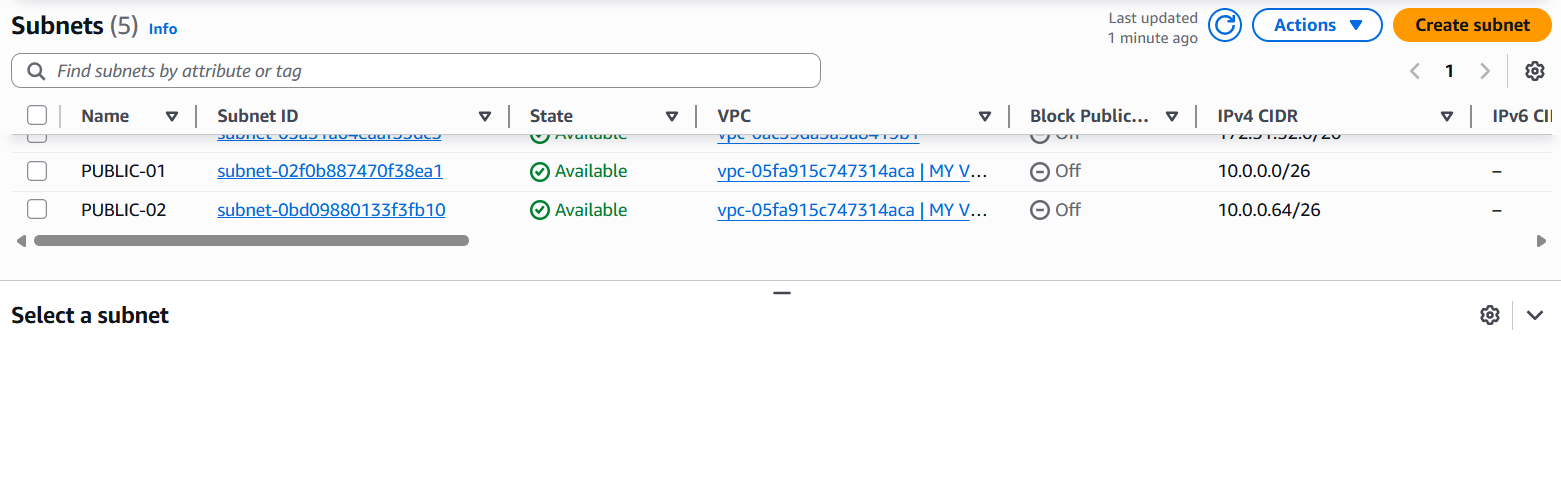
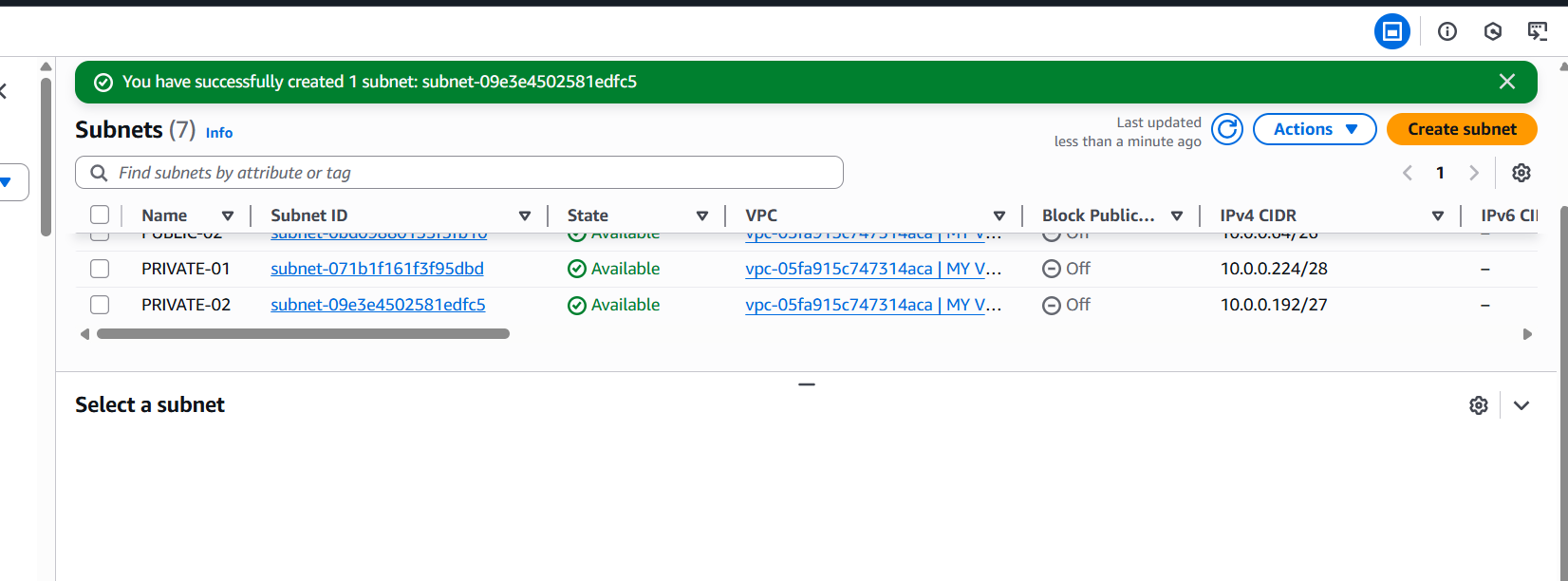
1. Create VPC with 2 private and 2 public subnets.

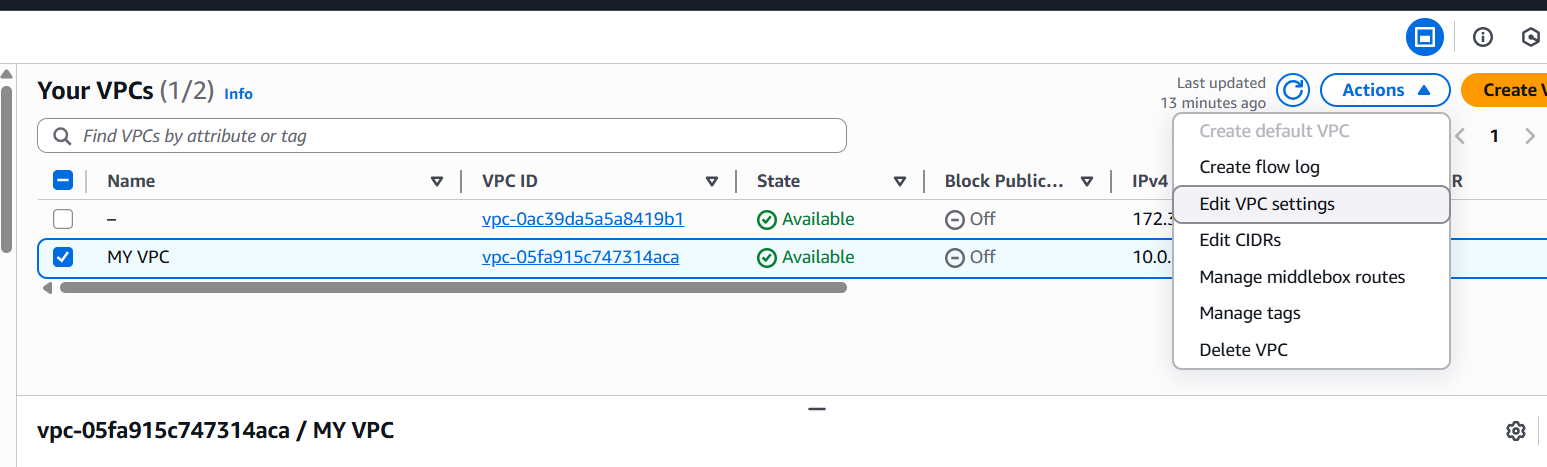


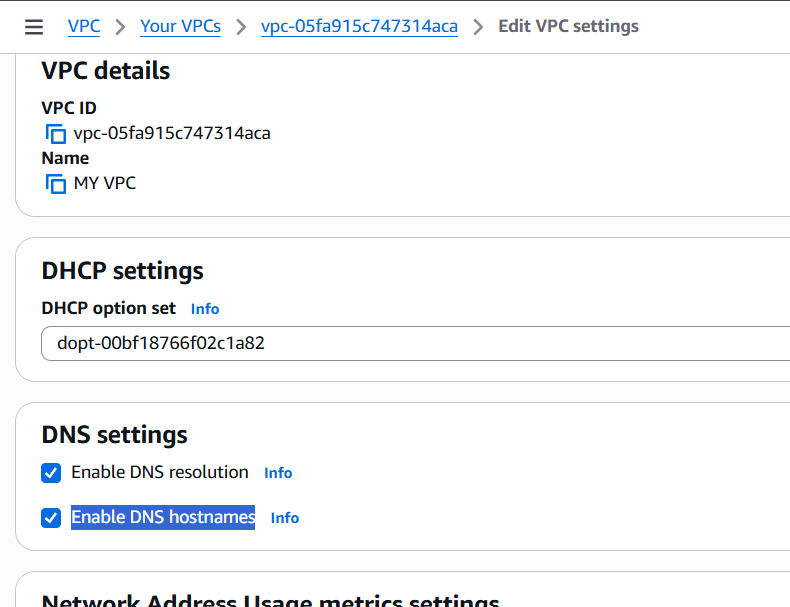




* Go to Aws console and in Search bar search vpc ,Click on vpc
* And click on create vpc
* And give name and
* Select ipv4 CIDR block
* And click on create vpc
* Here option called subnets ,Click on subnets
* And click on create subnet
* Select vpc (which vpc)
* Create subnet name (pub-sub)
* Click on IPV4 vpc CIDR block and select ip
* And give subnet CIDR block
* And click on create subnet
* Created 2 public subnets

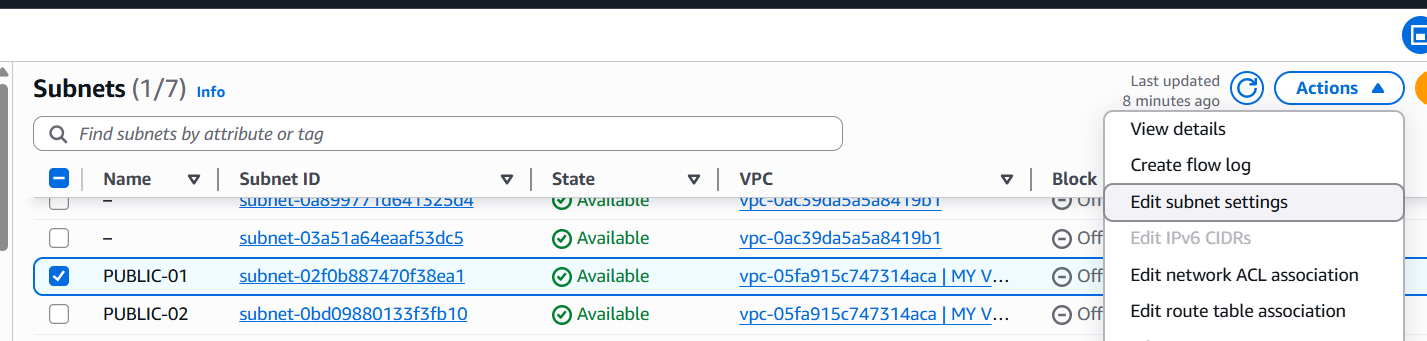
1. Enable DNS Hostname in VPC.

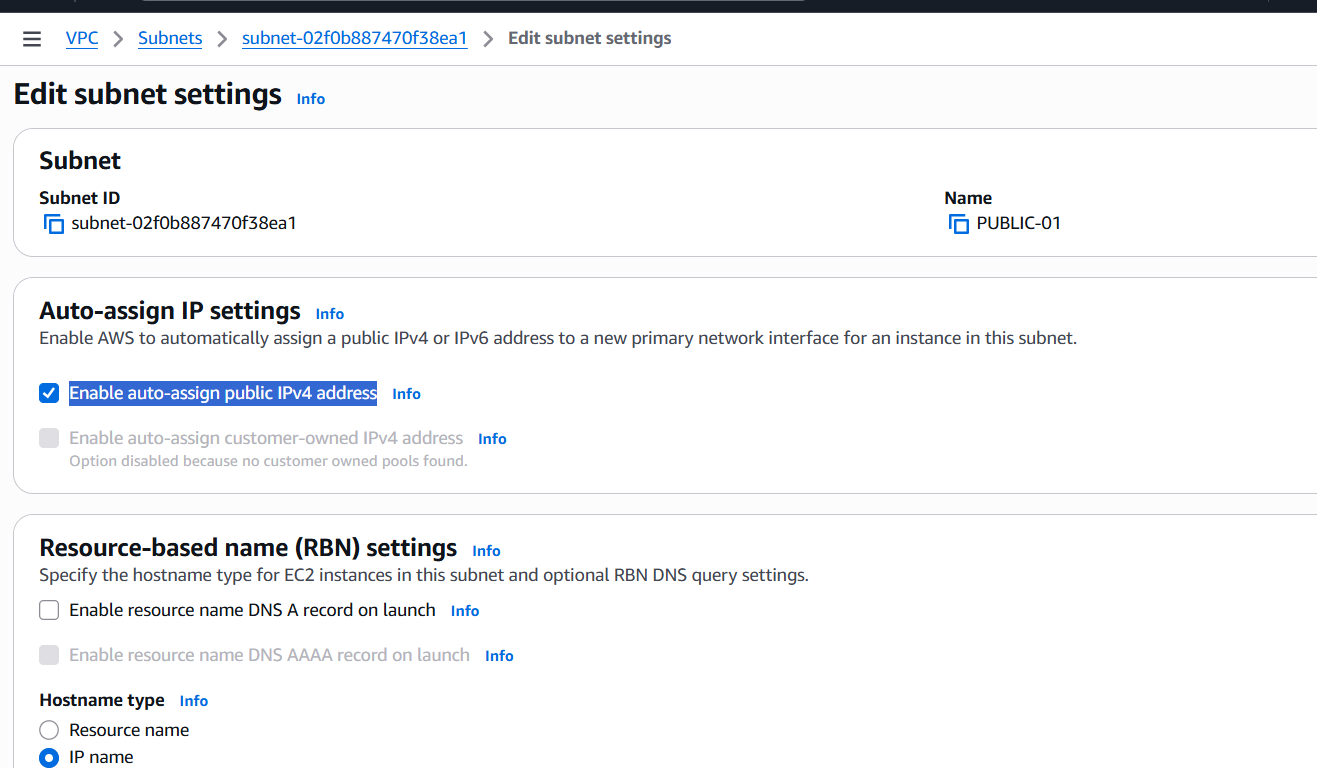


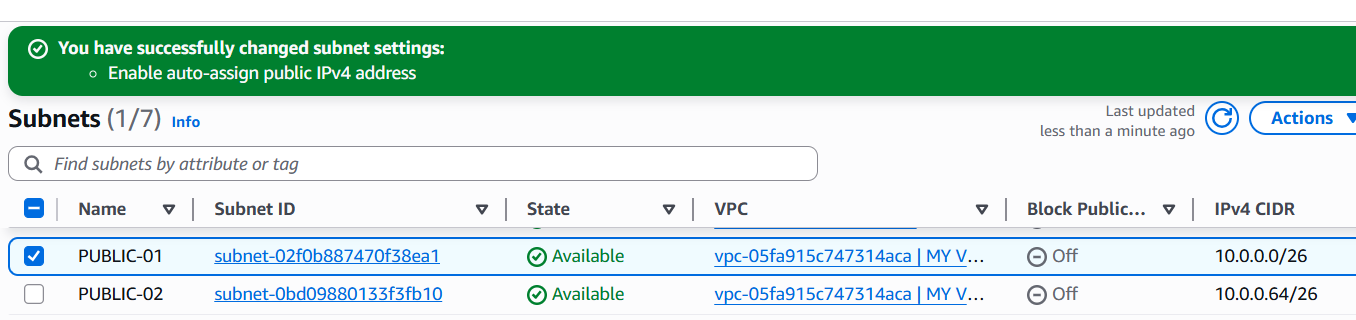


* Go to edit vpc settings.
* In the bottom we will find enable DNS hostnames.

1. Enable Auto Assign Public IP in 2 public subnets.

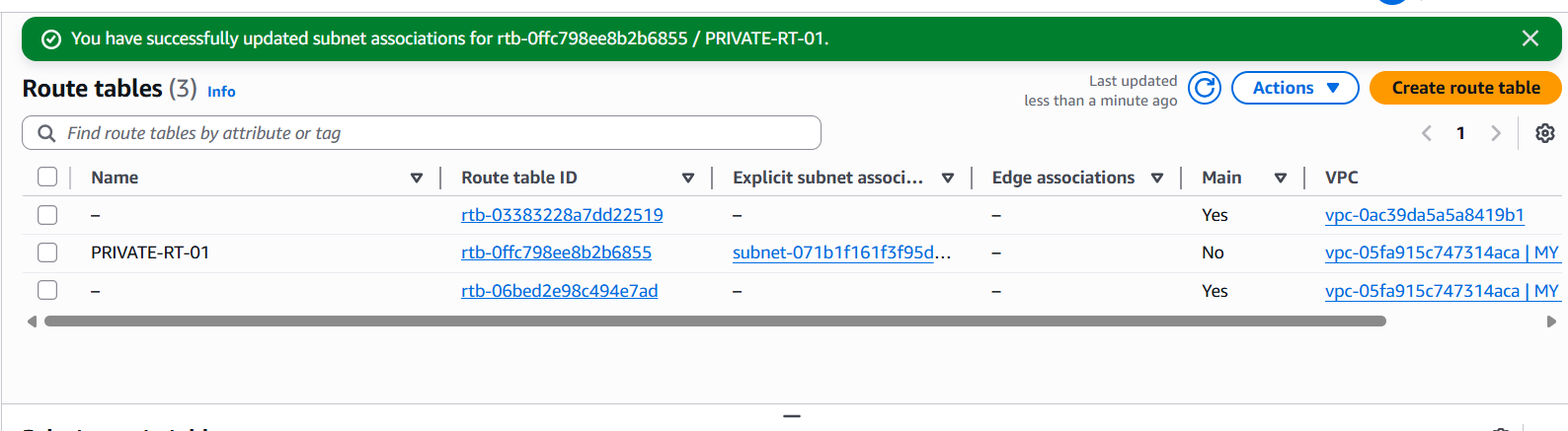


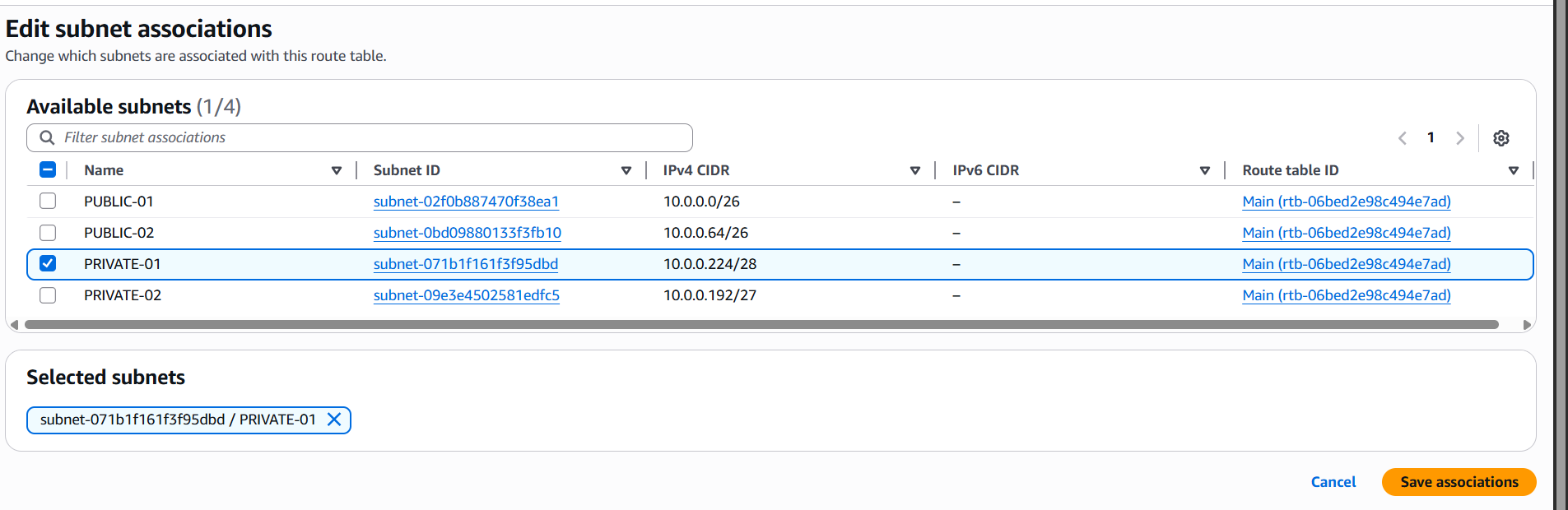


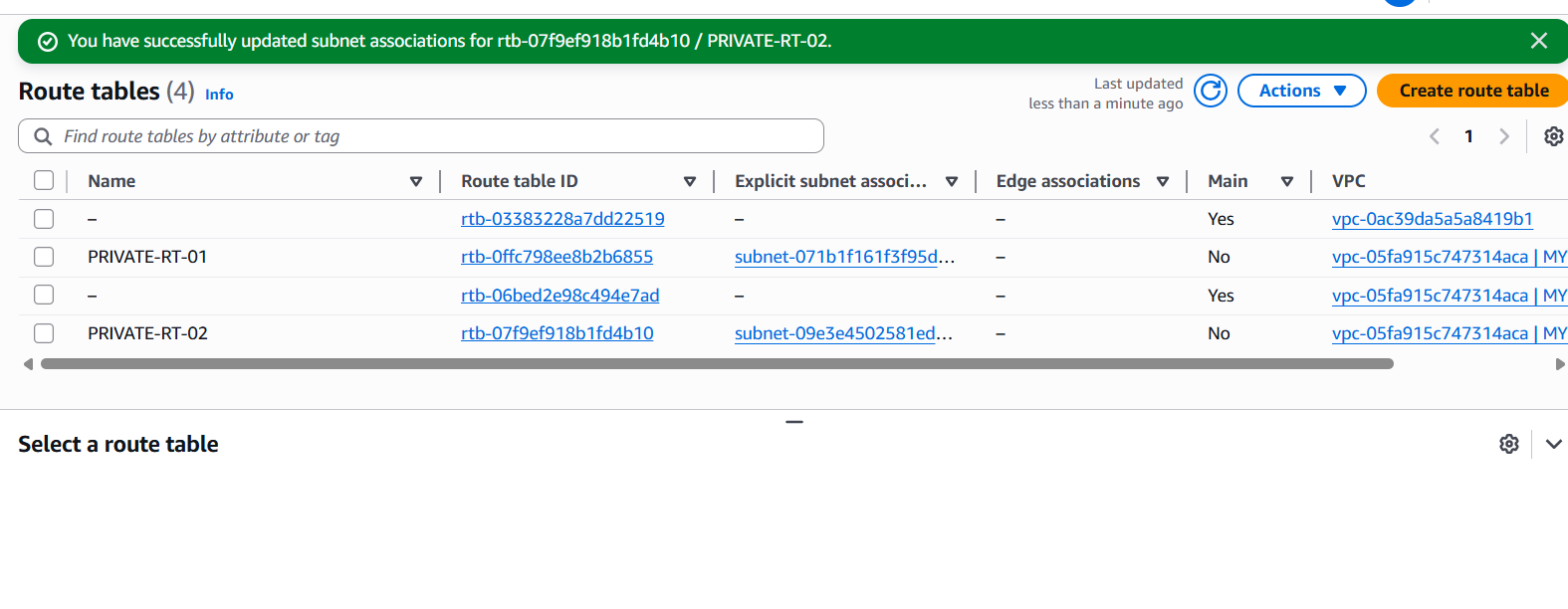


* Go to VPC Dashboard , click on Subnets
* Select your Public Subnet
* Click Actions → Edit subnet settings
* Enable Auto-assign IP settings → Auto-assign public IPv4 address
* Save changes
* Repeat the same steps for your second public subnet

1. Add 2 private subnets in private route table.

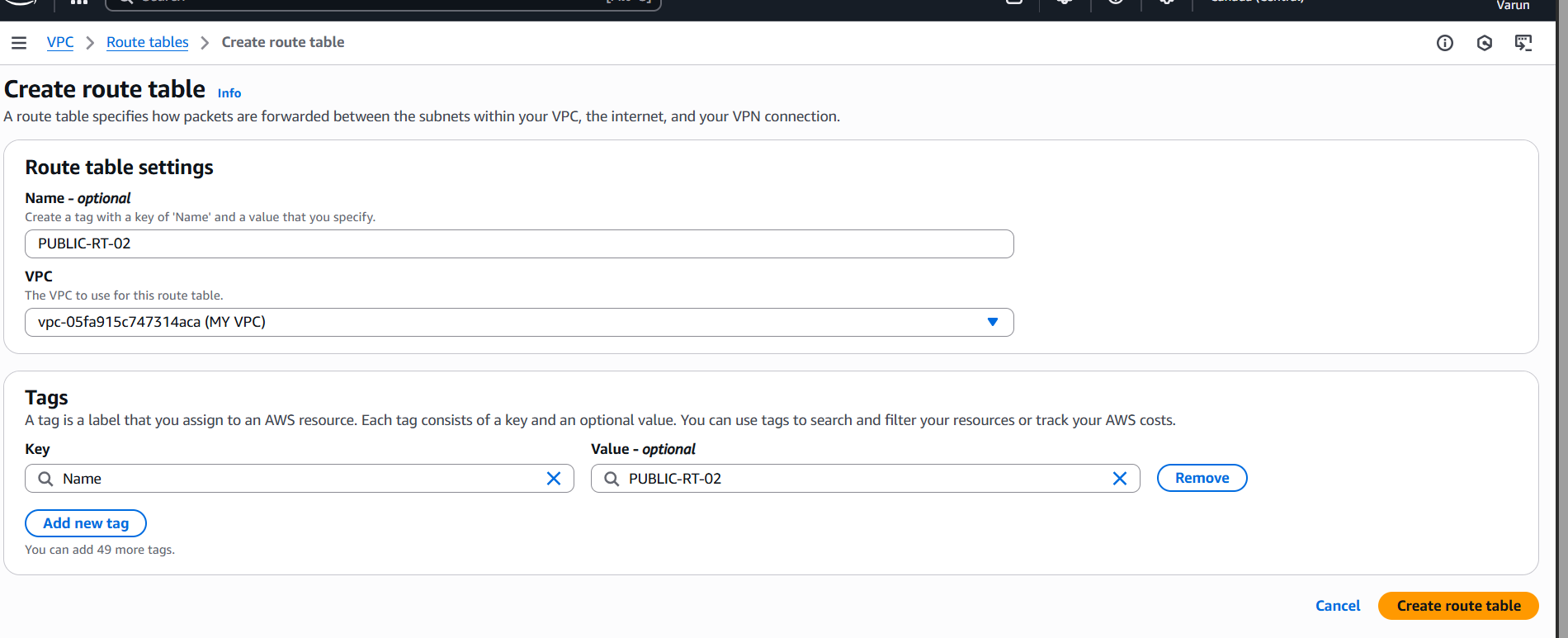


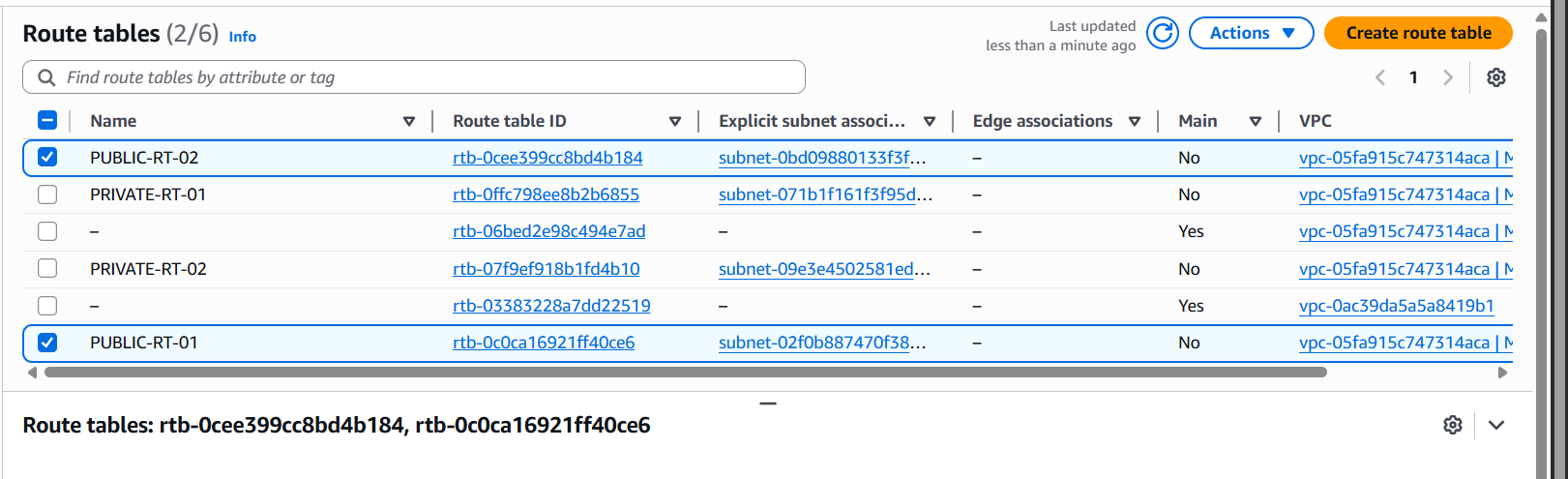




* Go to VPC →click on Route Tables → click on Create route table
* Give name like PRIVATE-RT-01,PRIVATE-RT-02
* Attach it to the my-VPC
* Add Private Subnets to the Route Table
* Select your new Private Route Table
* Go to Subnet associations → Edit subnet associations
* Select the two private subnets you created ,Save.

1. Add 2 public subnets in public route table.

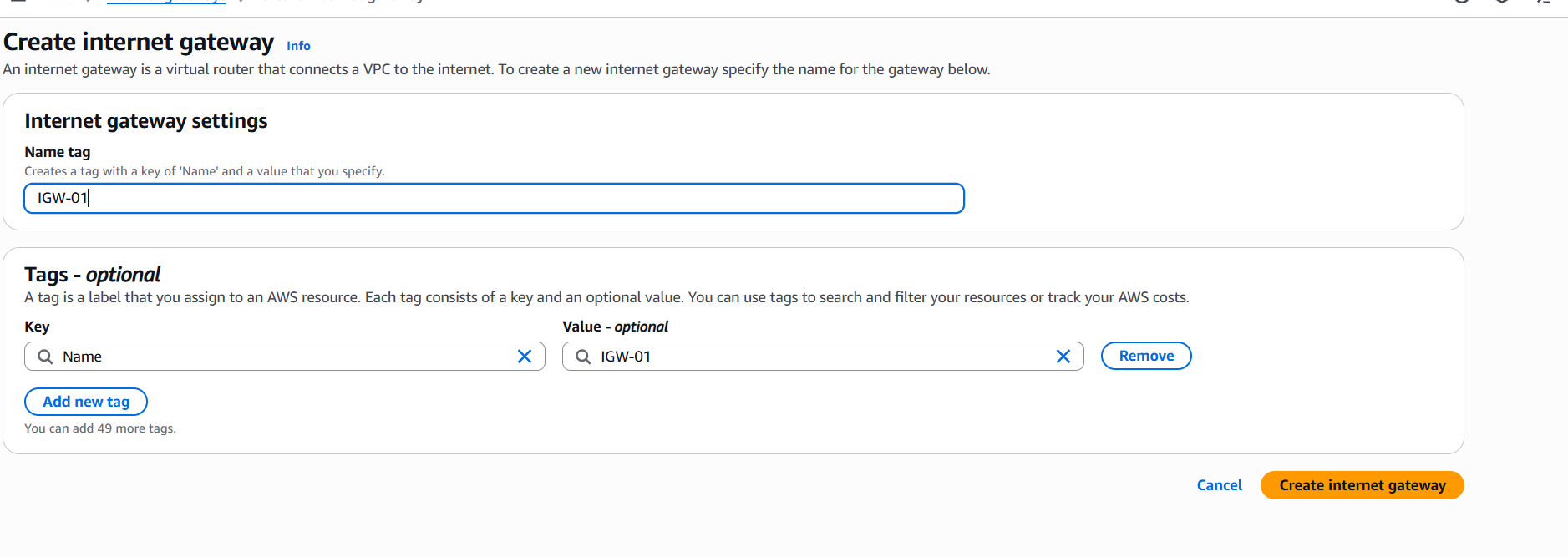


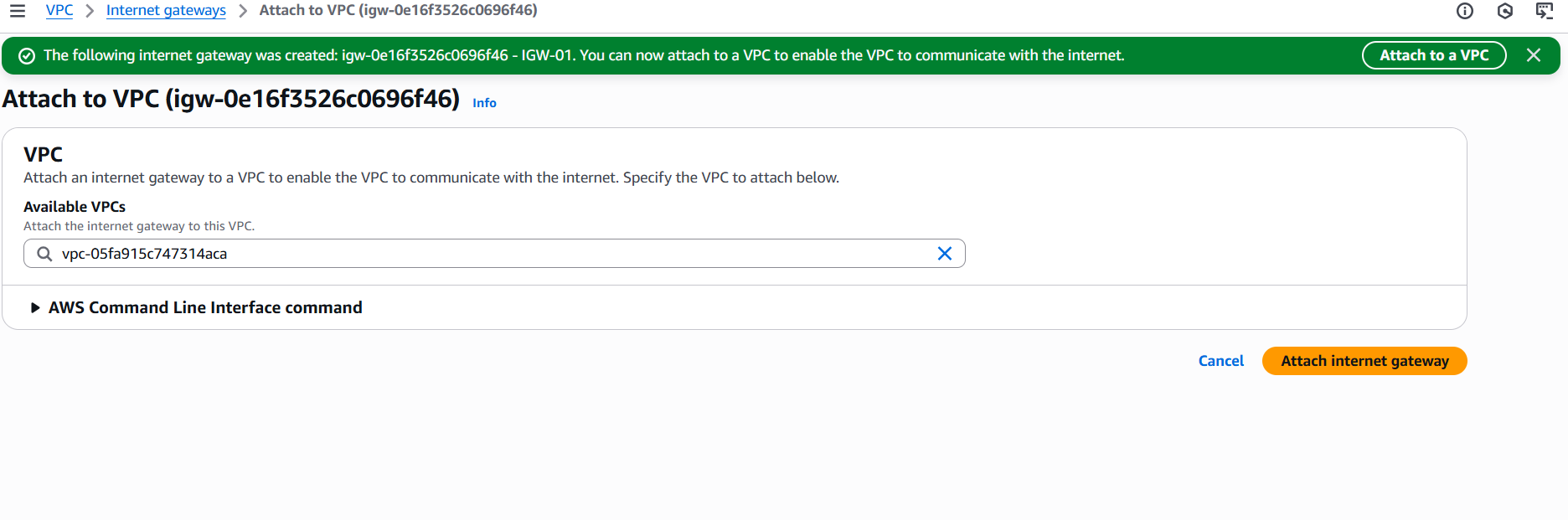


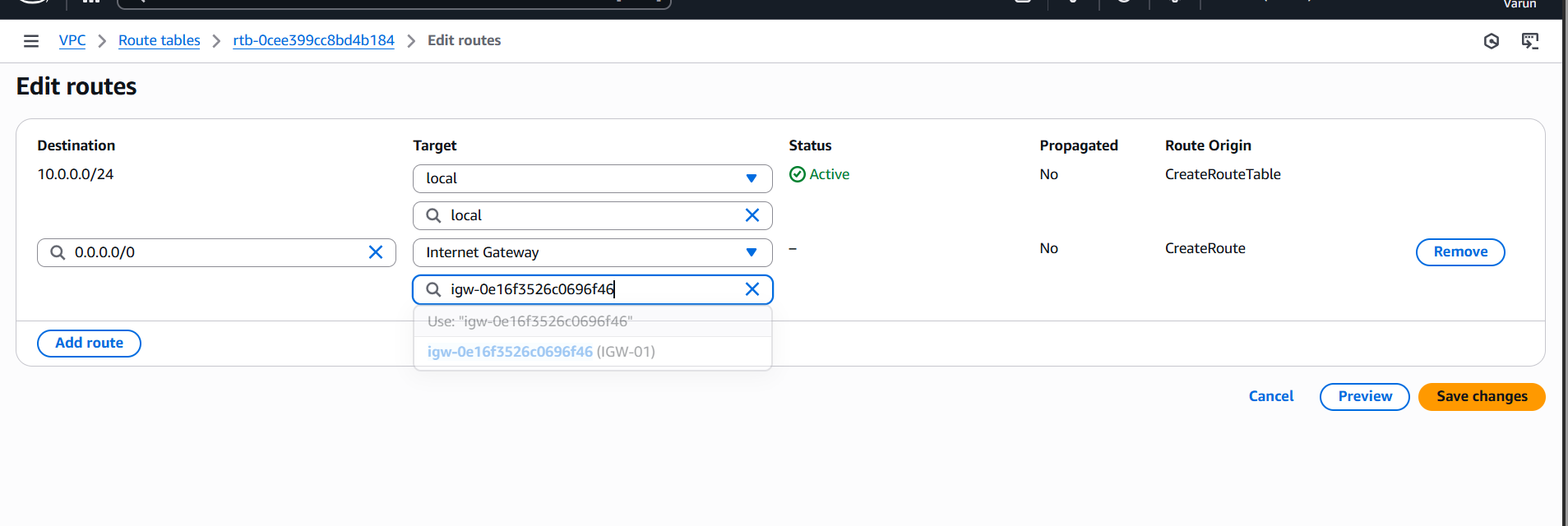
* Go to VPC →click on Route Tables → click on Create route table
* Give name like PUBLIC-RT-01
* Attach it to the my-VPC
* Add public Subnets to the Route Table
* Select your new Public Route Table
* Go to Subnet associations → Edit subnet associations
* Select the two public subnets you created,Save.

1. Public route table will have the routes to internet and local.



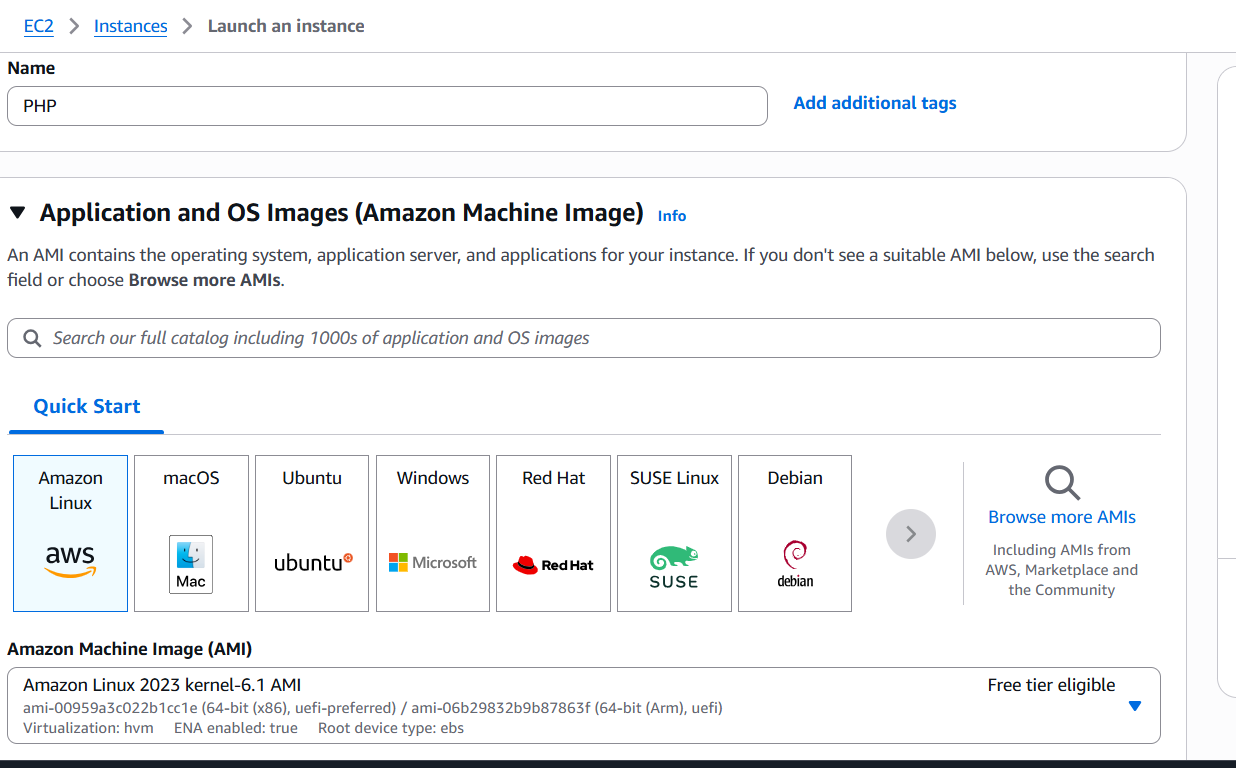


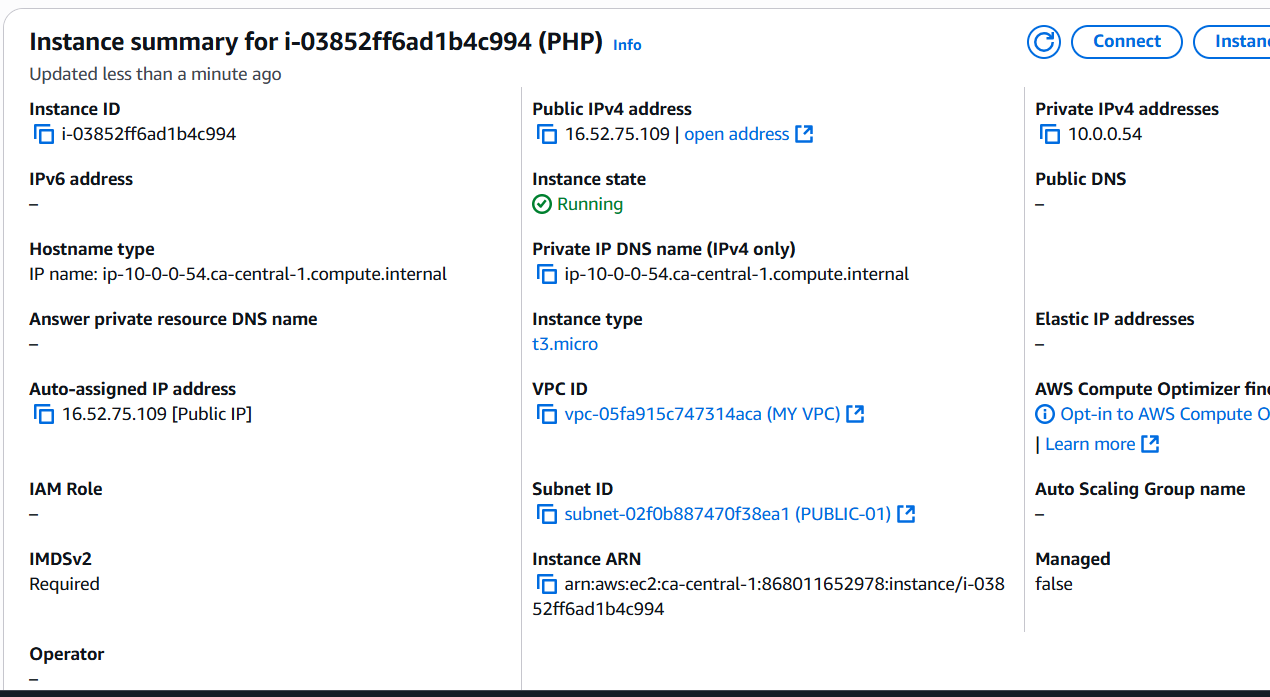


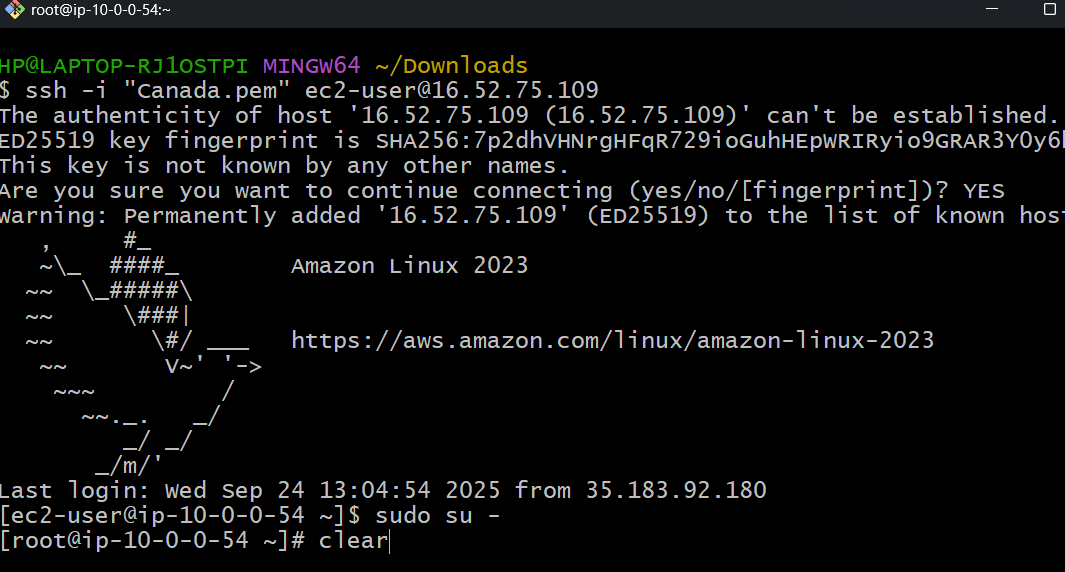


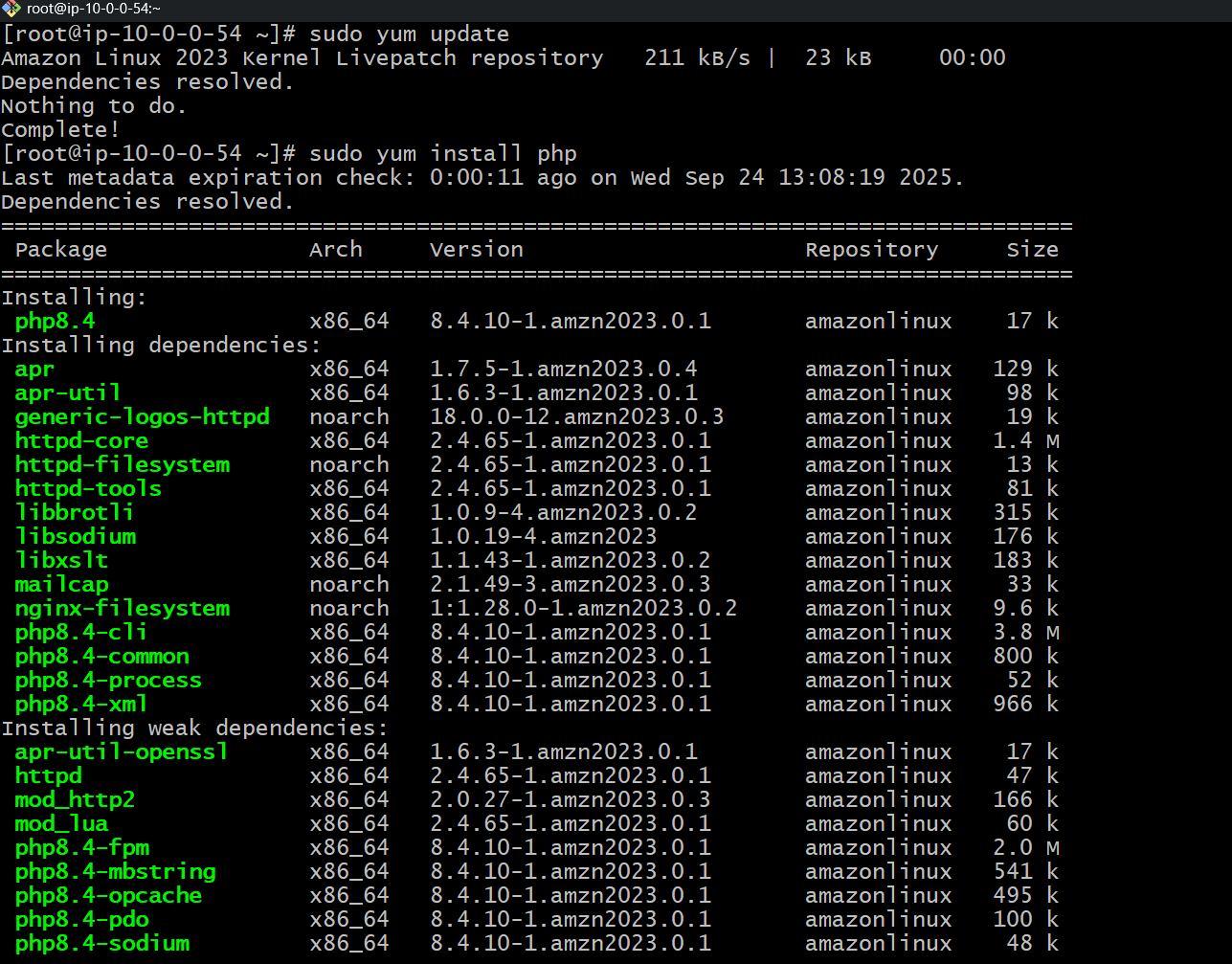
* Go to VPC click on Internet Gateways
* Create internet gateway and name it.
* Select the created IGW
* Click on Actions
* Attach to VPC and choose your VPC
* Edit routes by adding IGW.

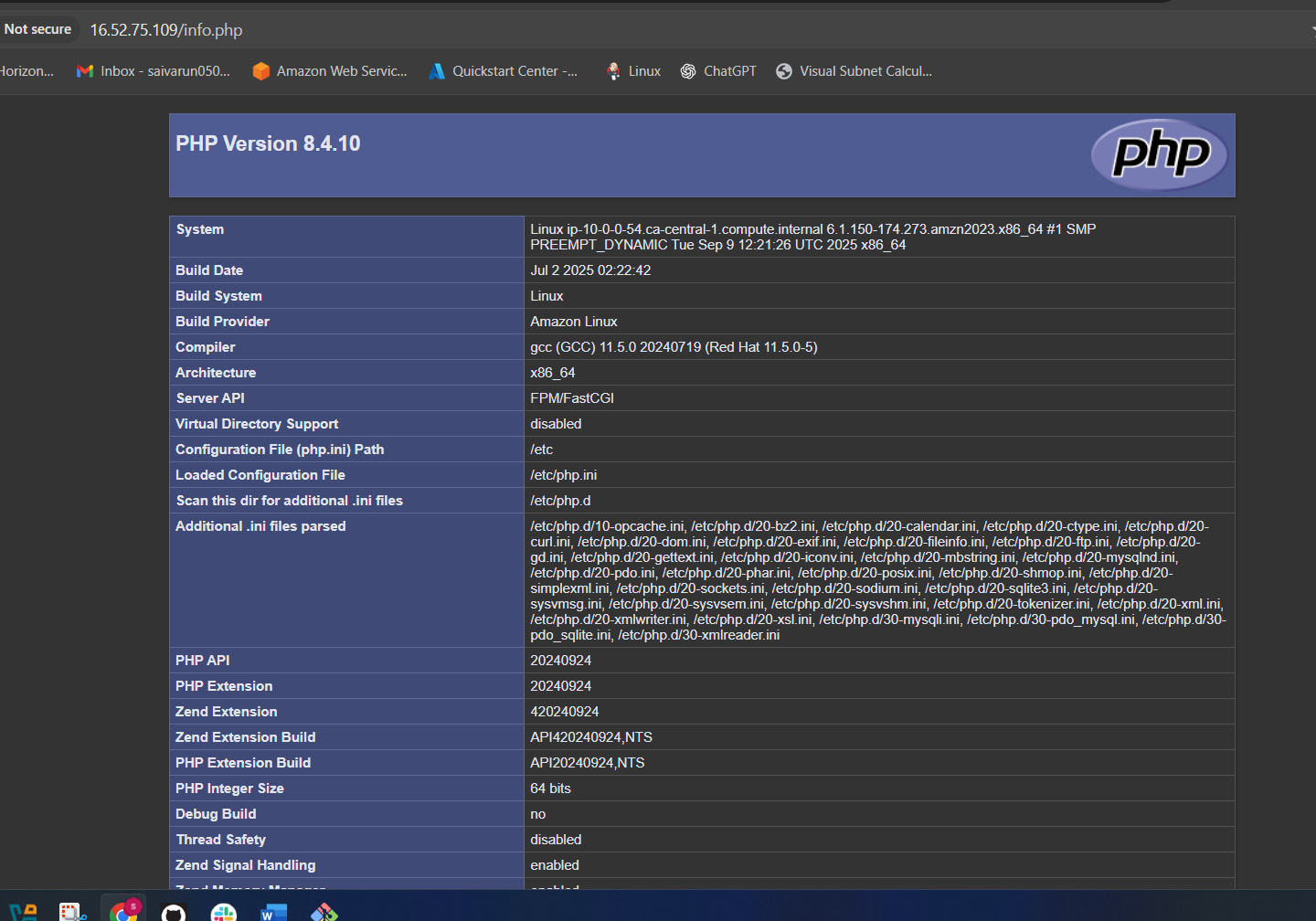
1. Create EC2 in public subnet with t2.micro and install PHP.











•Go to EC2 → Click on Instances  
• Click on Launch Instance  
•Give name Public-EC2  
•Select AMI Amazon Linux 2  
•Instance type: t2.micro (Free tier eligible).  
•Key Pair - Create or use an existing one.  
•Click on Network settings:  
•VPC: Select your VPC.  
•Subnet: Select your Public Subnet.  
•Auto-assign Public IP: Enable  
•Security Group - select existing one and give default security group

sudo yum update -y

sudo yum install httpd -y

sudo amazon-linux-extras install php8.2 -y # Or desired PHP version

sudo yum install php-cli php-mysqlnd php-gd php-mbstring php-xml -y

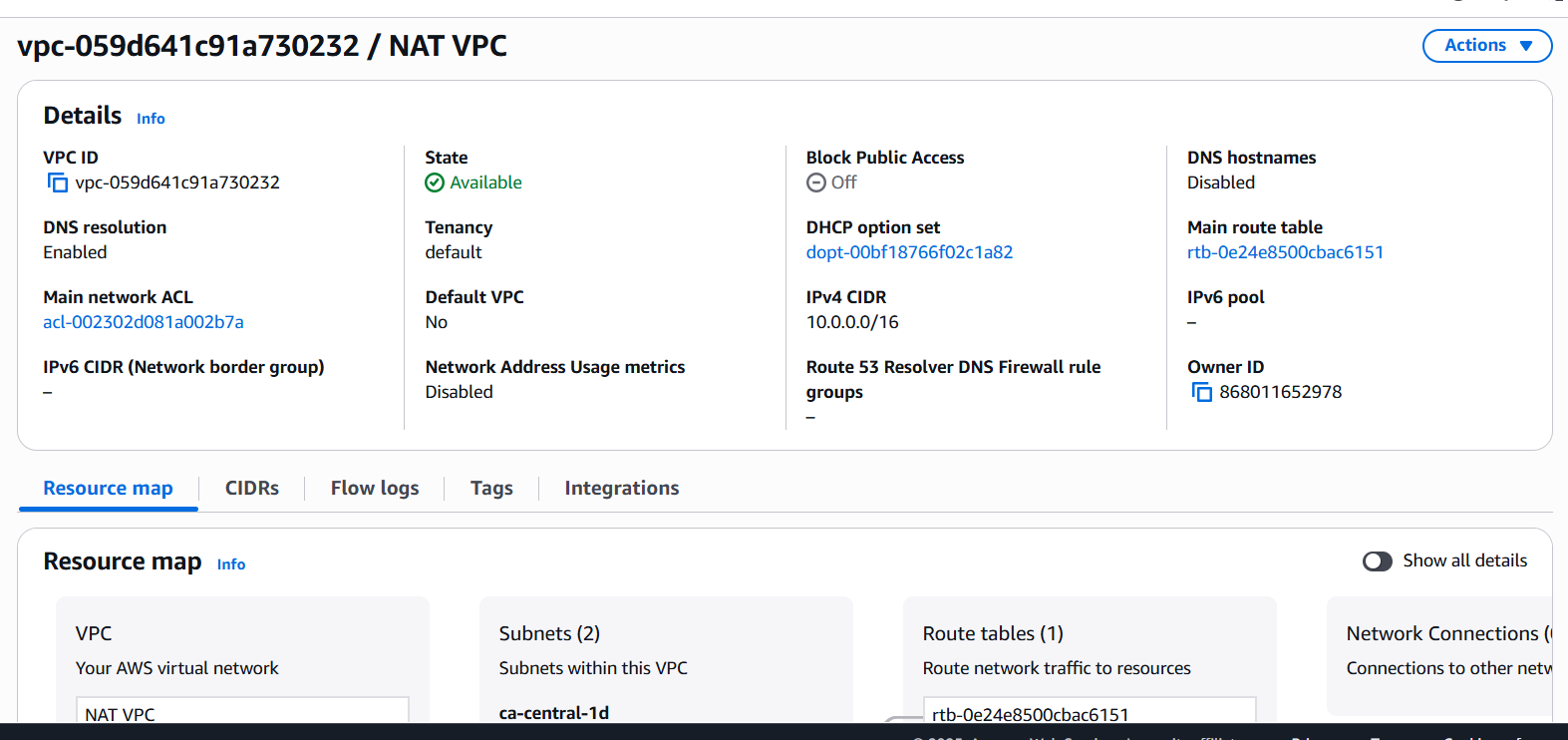
sudo yum install php php-mysql php-gd php-mbstring php-xml -y

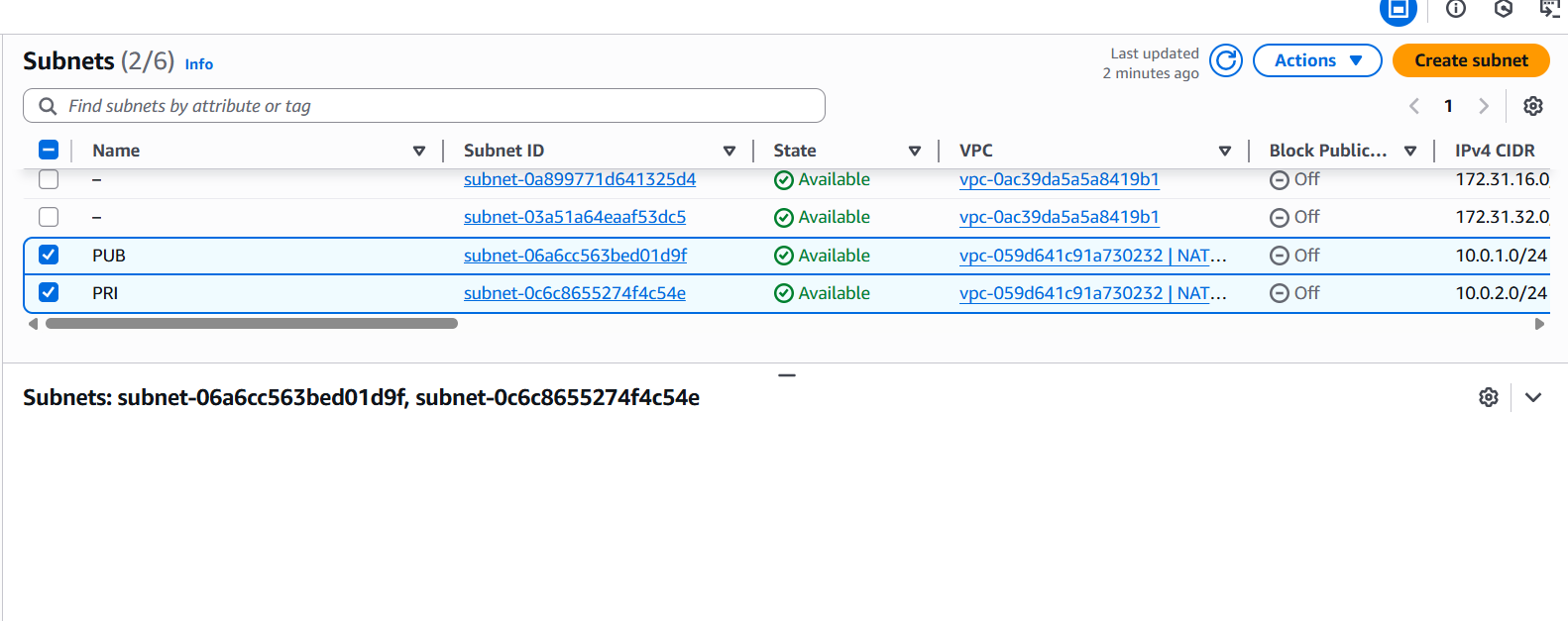
sudo systemctl start httpd

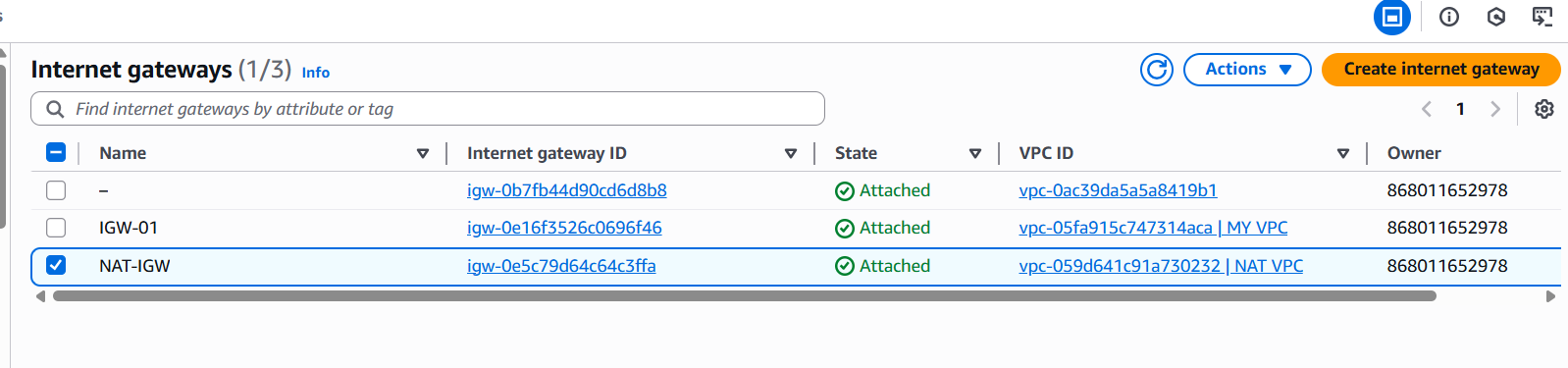
sudo systemctl enable httpd

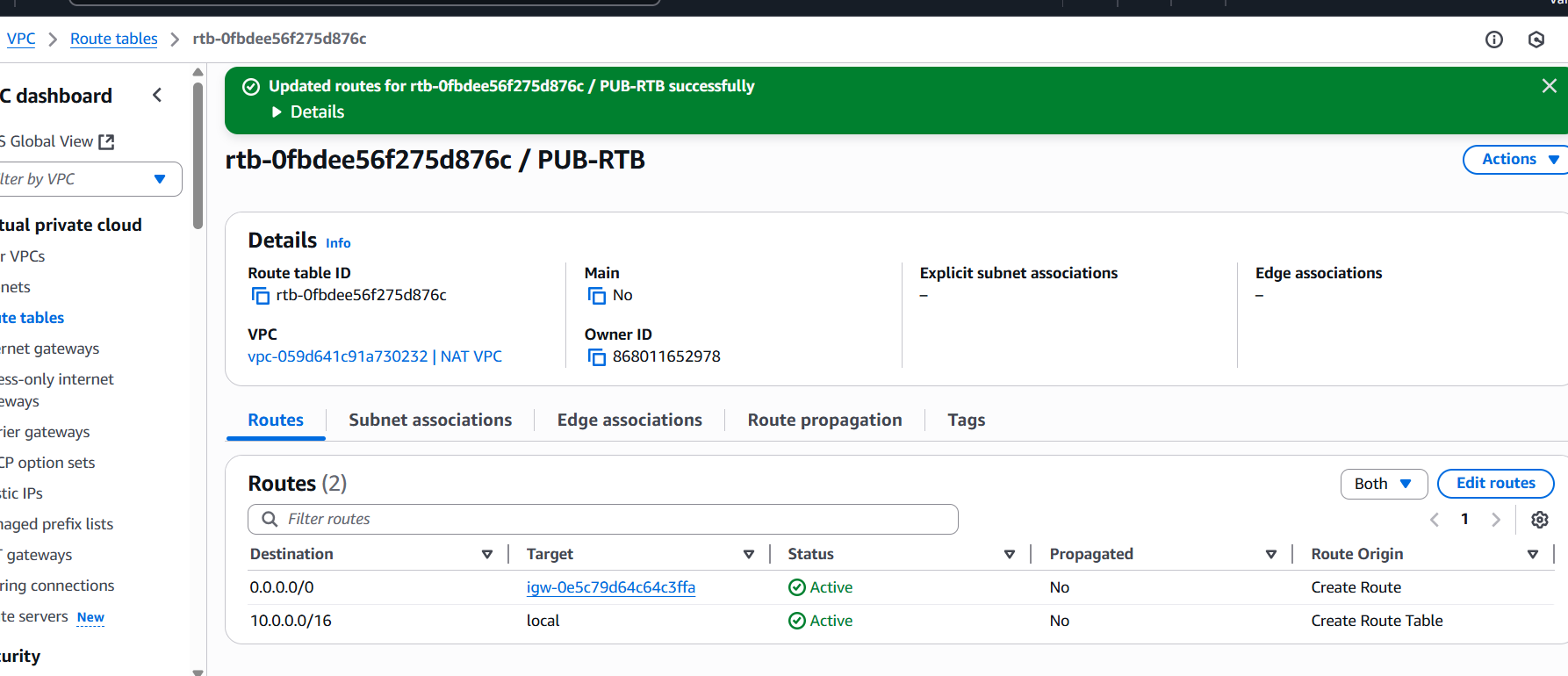
sudo echo "<?php phpinfo(); ?>" | sudo tee /var/www/html/info.php

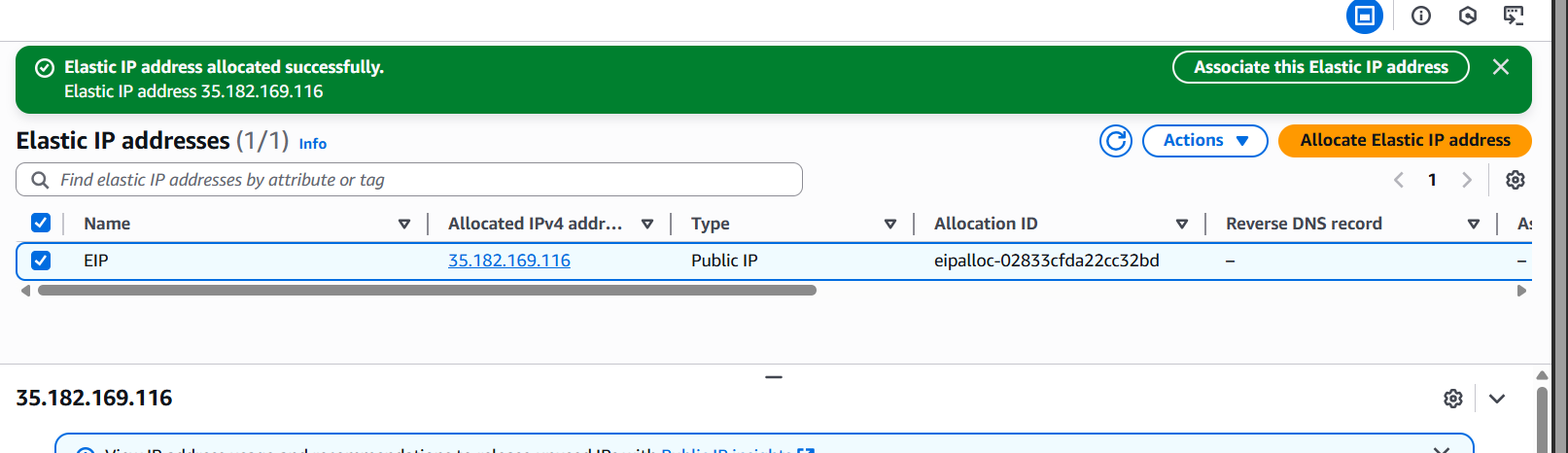
1. Configure NAT gateway in public subnet and connect to private instance.

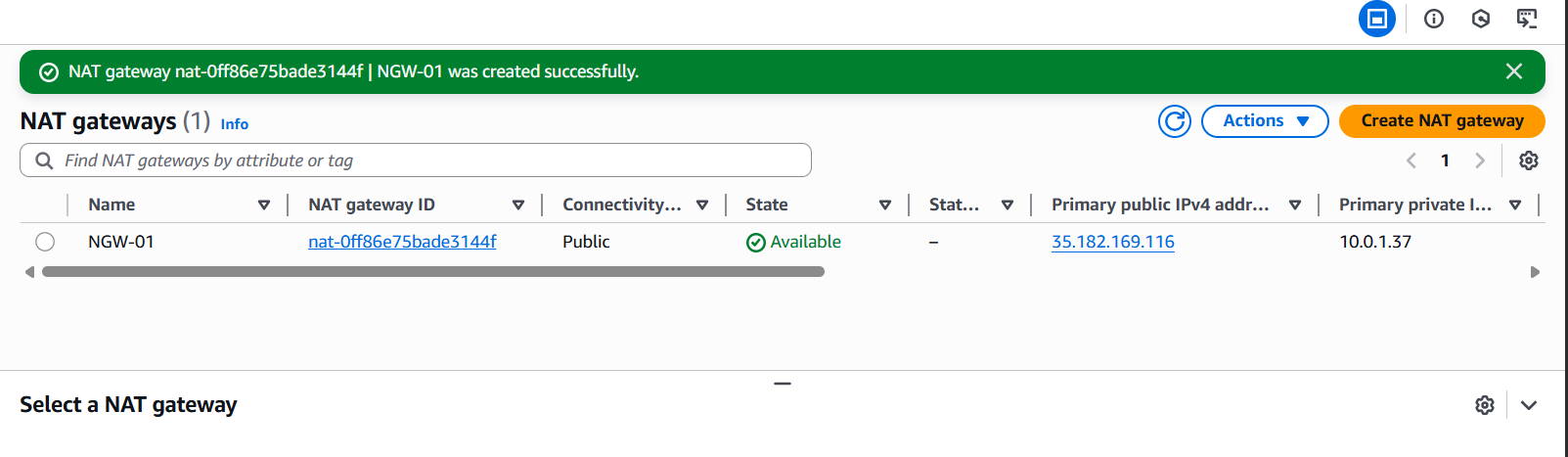


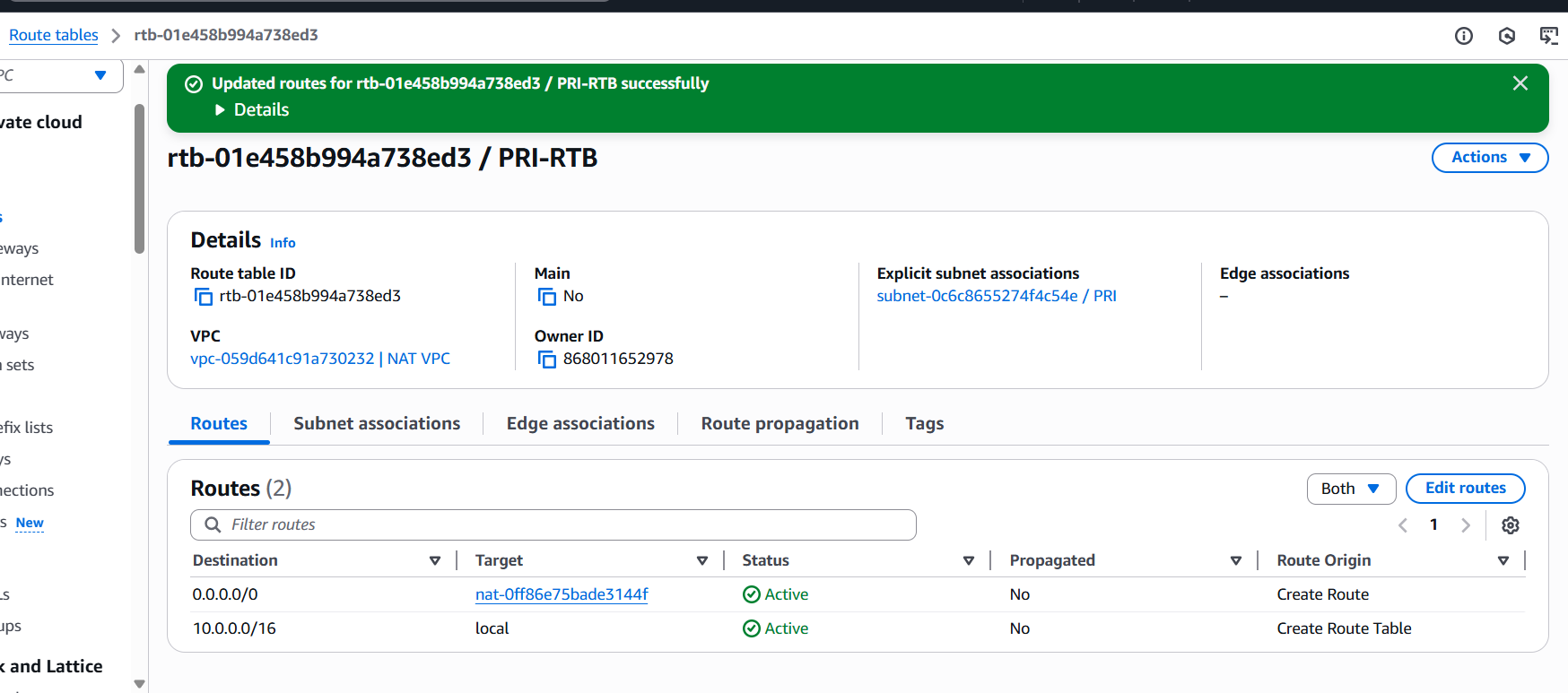


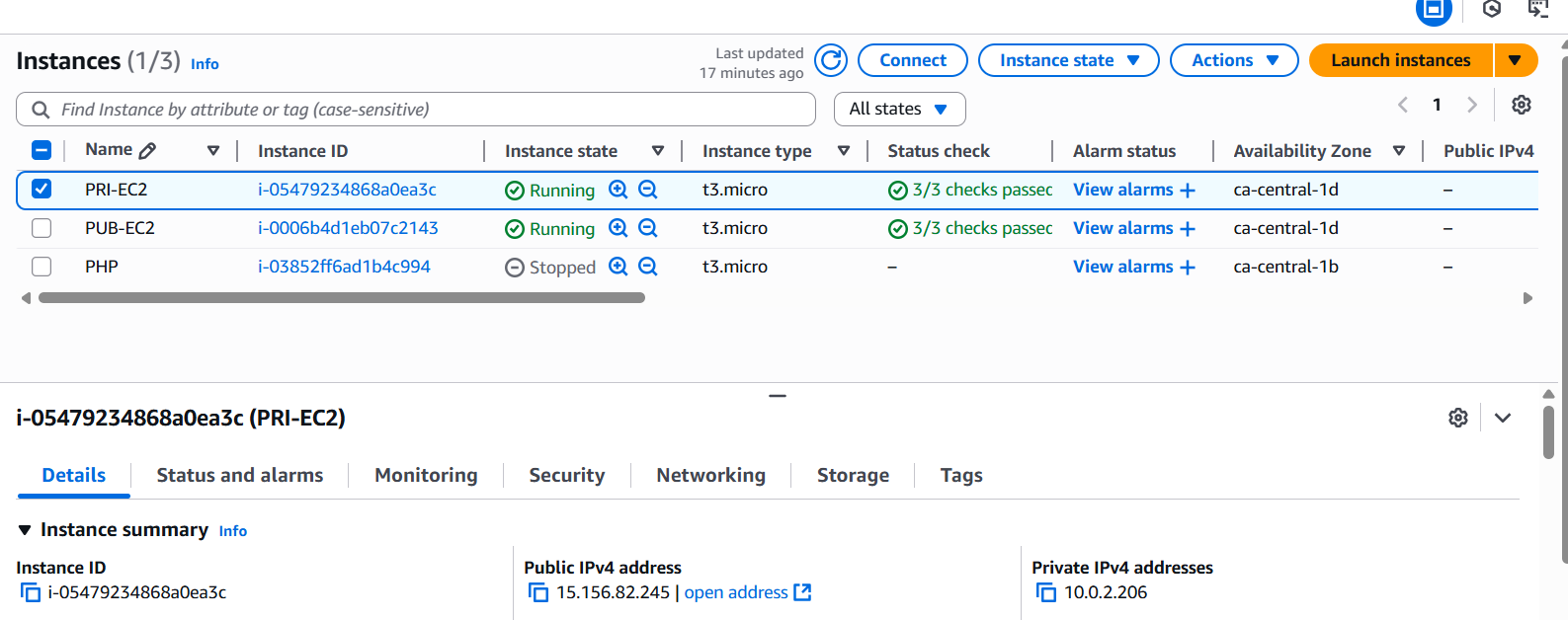


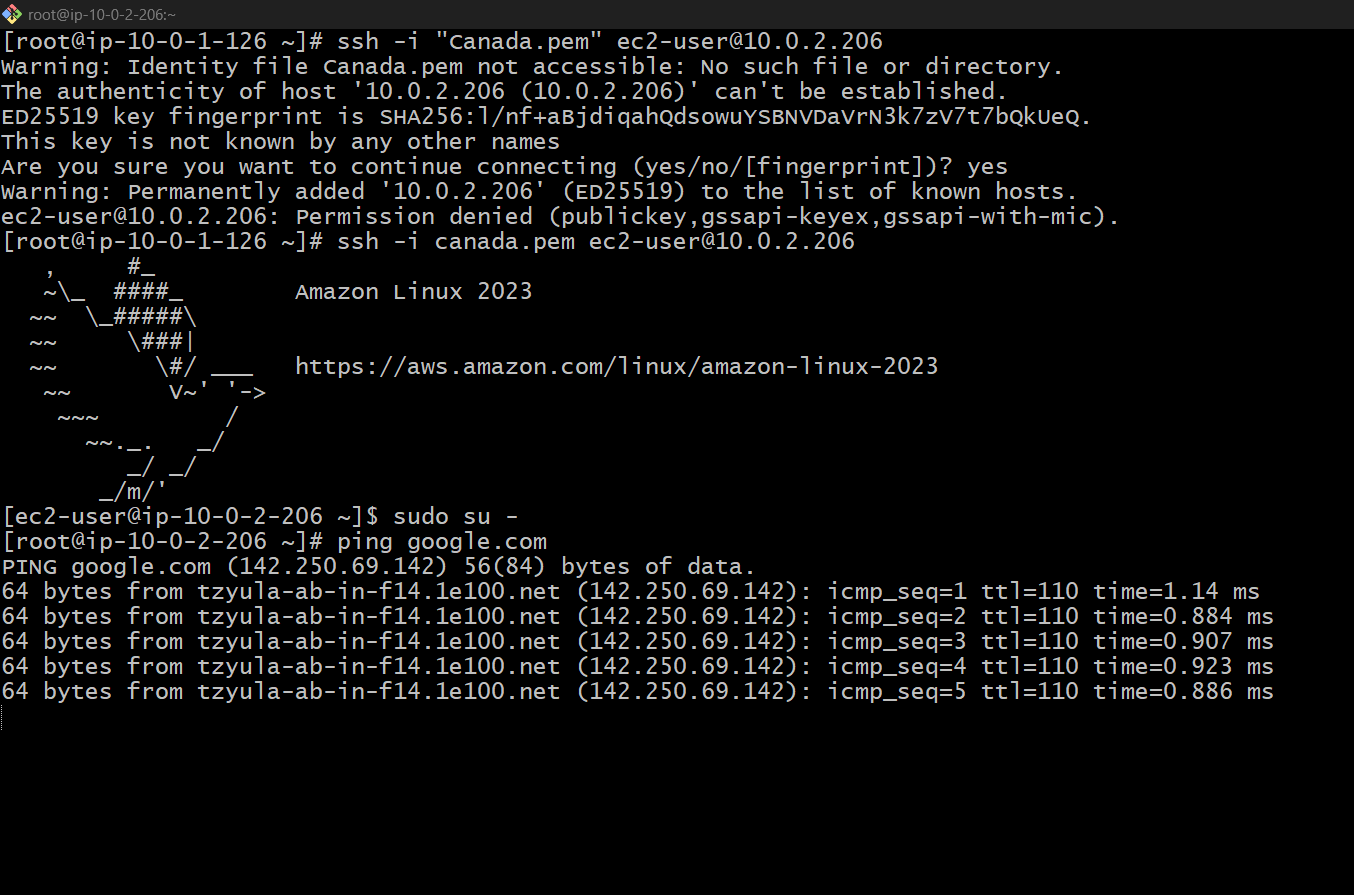












**1) Create (or choose) the VPC + subnets + IGW**

1. Console: **Services → VPC → Your VPCs → Create VPC**.
   * Name: my-vpc, IPv4 CIDR: 10.0.0.0/16.
2. **Subnets → Create subnet** (Public):
   * VPC: my-vpc, Name: public-subnet-1, AZ: your chosen AZ, CIDR: 10.0.1.0/24.
   * In subnet settings turn **Auto-assign IPv4 public IP** ON (so public EC2 get public IP).
3. **Subnets → Create subnet** (Private):
   * Name: private-subnet-1, CIDR: 10.0.2.0/24. Auto-assign public IP **OFF**.
4. **Internet Gateways → Create internet gateway**, name it my-igw → **Attach to VPC** my-vpc.

**2) Public route table (for the public subnet)**

Route Tables → Create route table, name rtb-public, VPC my-vpc.

Select rtb-public → Routes → Edit routes → Add route:

Destination 0.0.0.0/0 → Target: Internet gateway my-igw.

Subnet Associations → Edit subnet associations → associate public-subnet-1 with rtb-public.

**3) Allocate an Elastic IP and create the NAT Gateway**

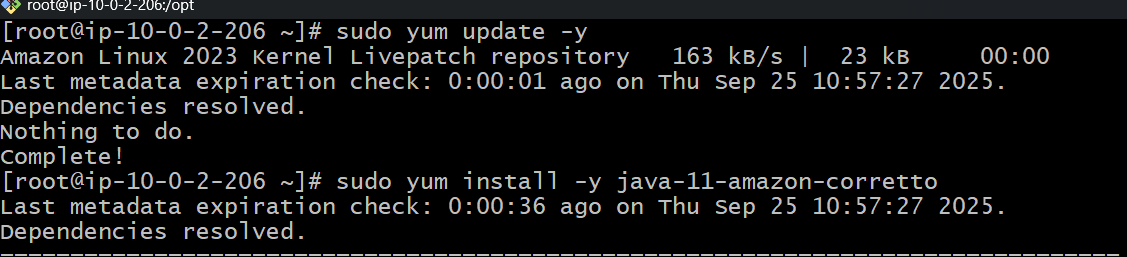
1. Console: **VPC → Elastic IPs → Allocate Elastic IP address**. Note the EIP (you’ll use it for testing).
2. **VPC → NAT Gateways → Create NAT gateway**:
   * Subnet: choose **public-subnet-1** (NAT must be in a public subnet).
   * Elastic IP allocation: choose the EIP you allocated.
   * Name it nat-gw-1. Click **Create NAT gateway**.
3. Wait a minute until status becomes **Available**.

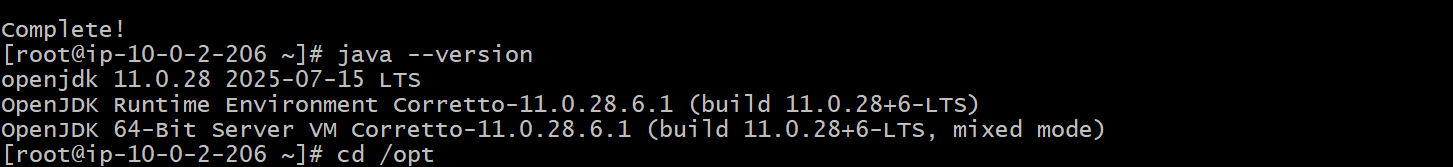
**4) Private route table → route 0.0.0.0/0 to NAT**

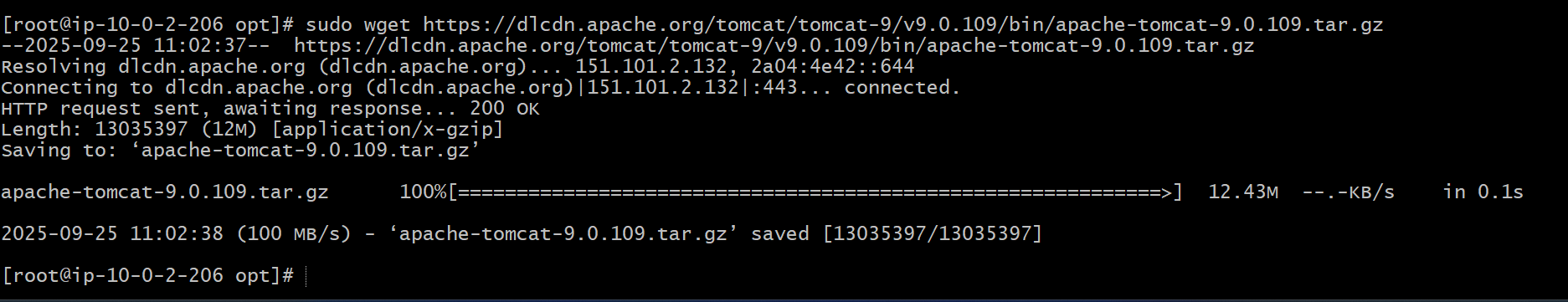
1. **Route Tables → Create route table**, name rtb-private, VPC my-vpc.
2. **Subnet Associations → Edit subnet associations** → select private-subnet-1.
3. Select rtb-private → Routes → Edit routes → Add route:
   * Destination 0.0.0.0/0 → Target: **NAT Gateway** → choose nat-gw-1.
4. Save. Now any EC2 in private-subnet-1 will send Internet-bound traffic to the NAT gateway and use the NAT EIP.
5. Create 2 ec2 instances public and private.
6. Connect to public instances and ping to private ip address now.

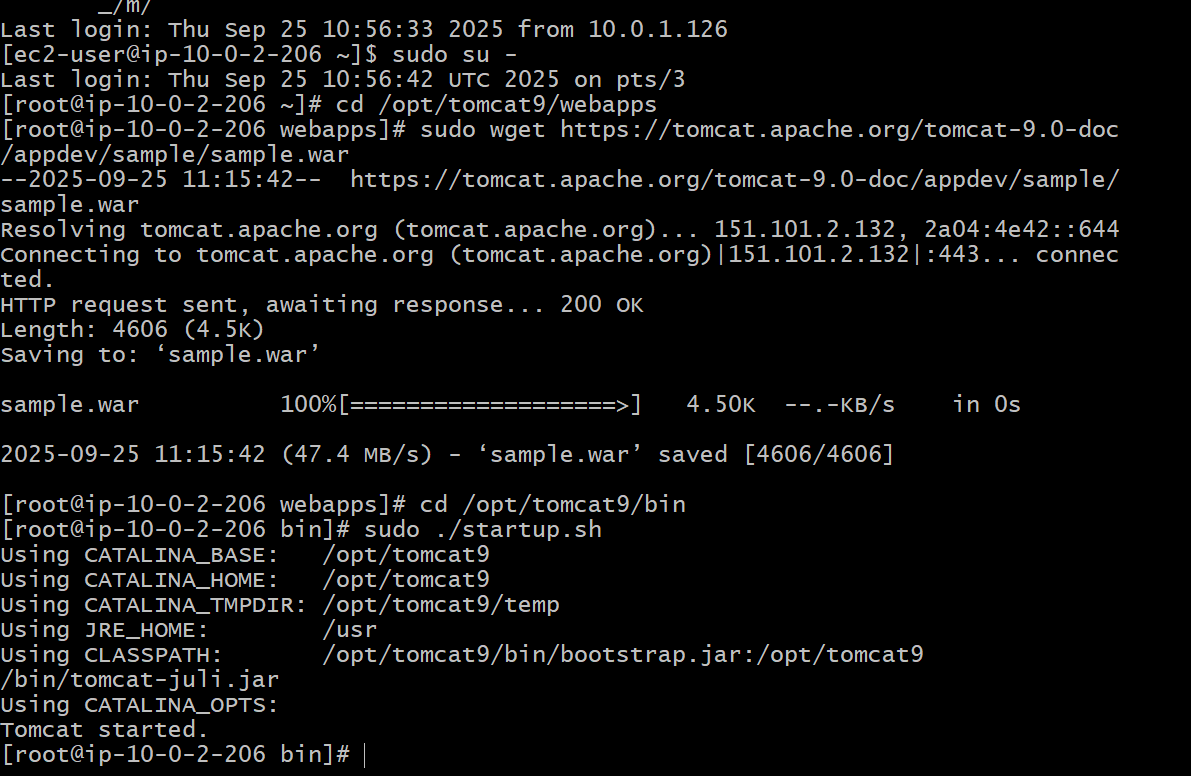
------------------------------END----------------------------------------------------------

1. **Install Apache Tomcat in private EC2 and deploy a sample app.**

****

****

****

****

**1.connect to private ec2**

ssh -i my-key.pem ec2-user@<private-ec2-private-ip>

**2.update and install java-11**

sudo yum update -y  
sudo yum install -y java-11-amazon-corretto  
Check version:  
java -version

**3.Download and install apache tomcat**

**1. Go to /opt directory:  
cd /opt  
sudo wget https://downloads.apache.org/tomcat/tomcat-9/v9.0.91/bin/apache-  
tomcat-9.0.91.tar.gz  
2. Extract:  
sudo tar -xvzf apache-tomcat-9.0.91.tar.gz  
sudo mv apache-tomcat-9.0.91 tomcat9**

**3. Make scripts executable:  
sudo chmod +x /opt/tomcat9/bin/\*.sh**

**4.start tomcat**

cd /opt/tomcat9/bin  
sudo ./startup.sh  
Check logs:  
tail -f /opt/tomcat9/logs/catalina.out

**5.Deploy a sample app**

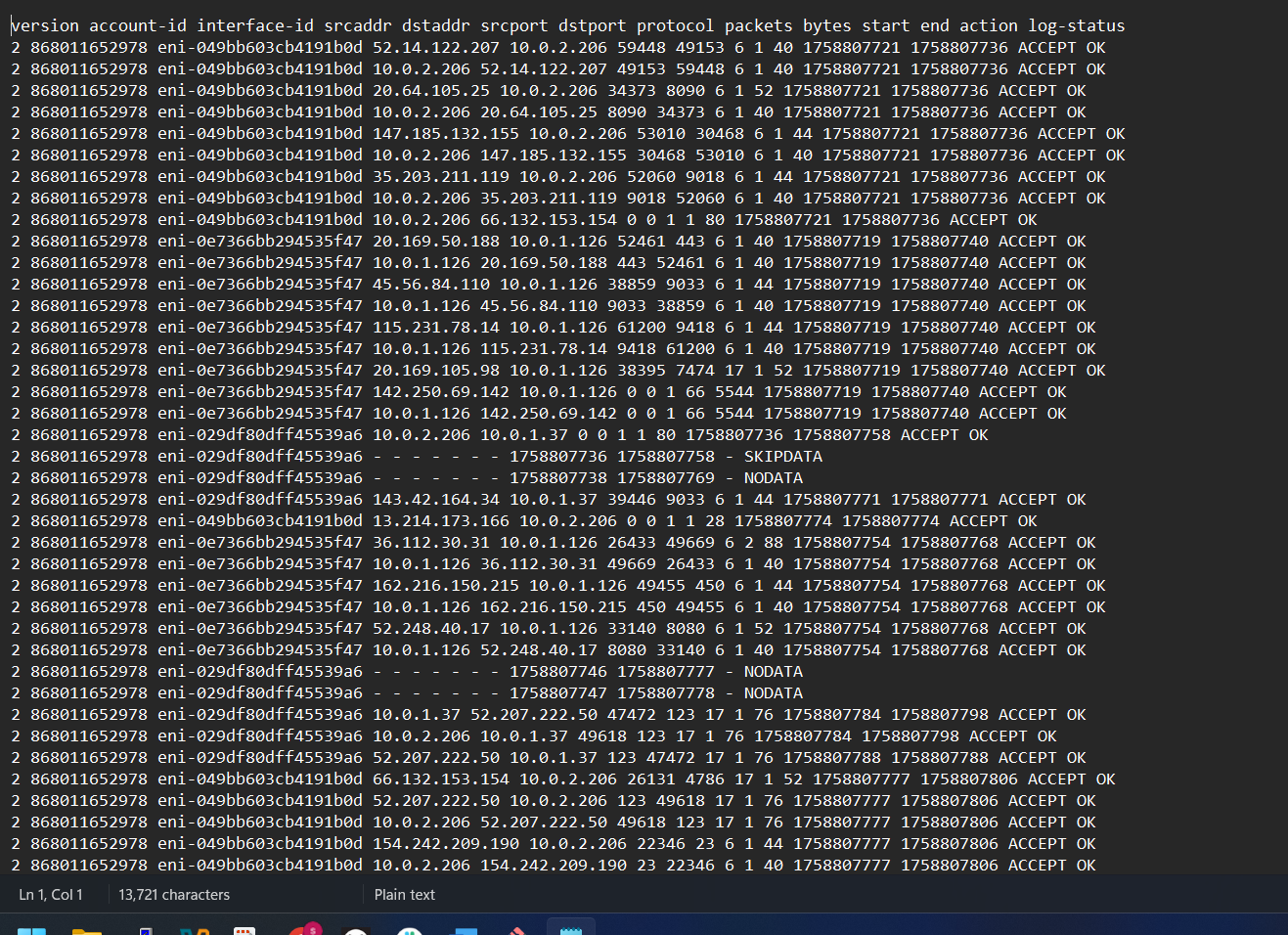
1. Download a sample .war file (Java web app):  
cd /opt/tomcat9/webapps  
sudo wget https://tomcat.apache.org/tomcat-9.0-doc/appdev/sample/sample.war  
2. Tomcat will automatically deploy it (extracts to sample/ folder)

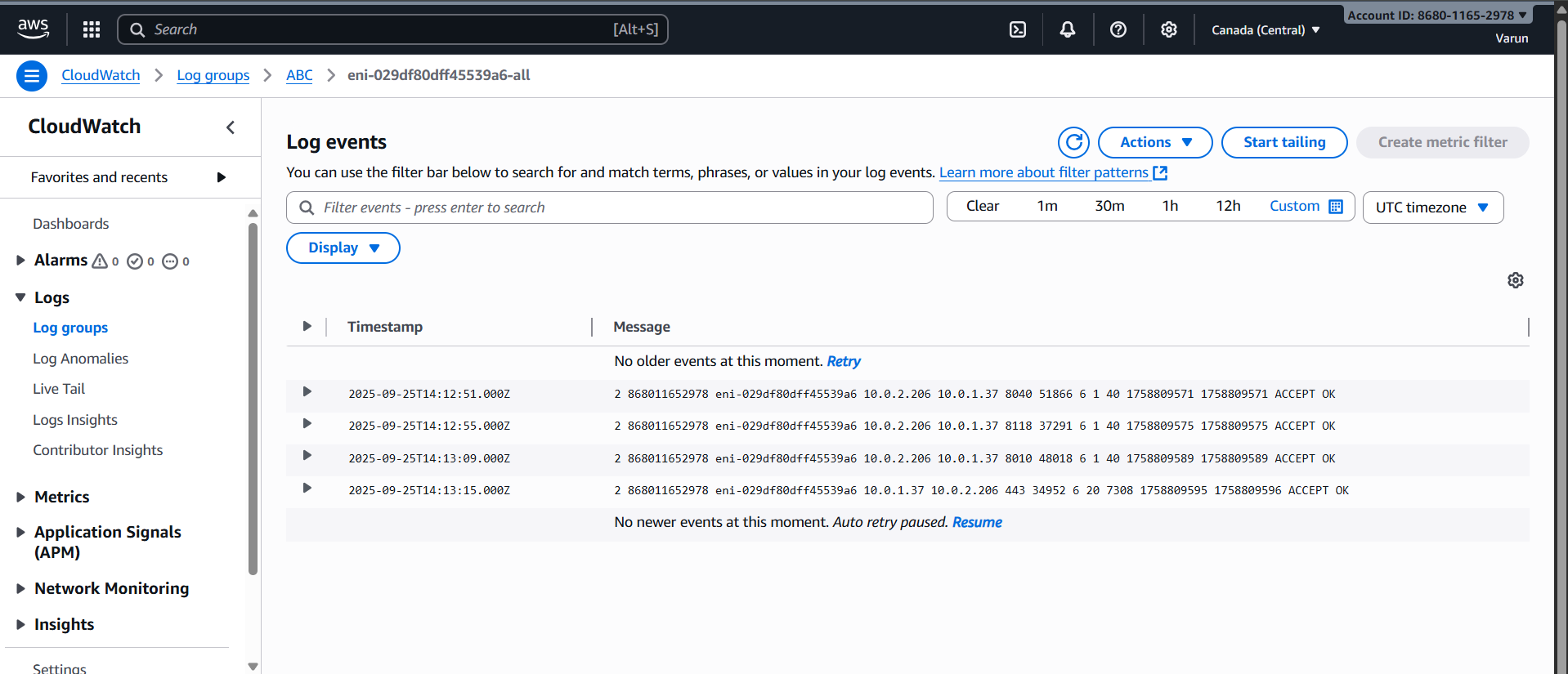
**6.Verify deployment.**

Use curl inside the private EC2

-------------------------------------------END-----------------------------------------------------------

1. **Configure VPC flow logs and store the logs in S3 and CloudWatch.**





* Go to vpc ->select vpc->create a flowlog->give a name->filter ALL-> Aggregation 1 minute->Send to s3 bucket
* Create s3 bucket->copy the ARN->paste->create flowlog
* Navigate to vpc flowlogs
* Wait for a while and refresh
* Go inside it and download the file and view it in your notepad.