Assignment 1

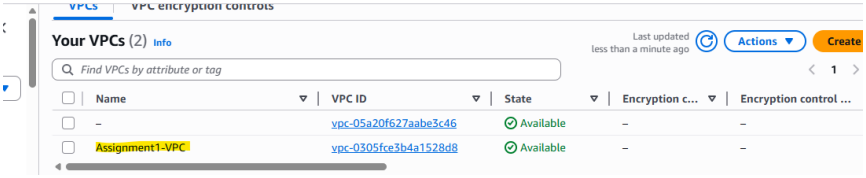
Working for an organization, you are required to provide them with a safe and secure environment for the deployment of their resources. They might require different types of connectivity. Implement the following to fulfill the requirements of the company.

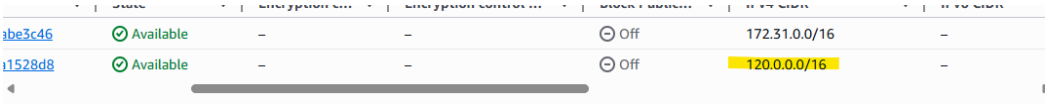
Tasks To Be Performed:

1. Create a VPC with 120.0.0.0/16 CIDR block.

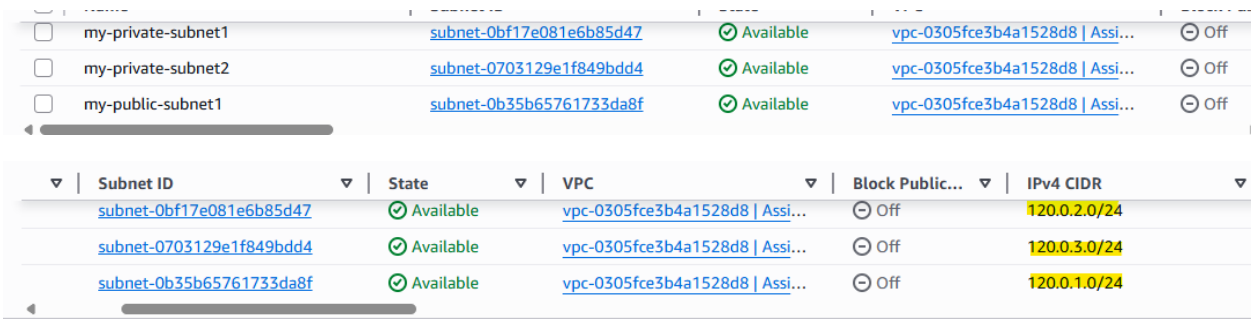
2. Create 1 public subnet and 2 private subnets and make sure you connect a NAT gateway for internet connectivity to a private subnet

Step 1: Create VPC with the CIDR block 172.31.0.0/16

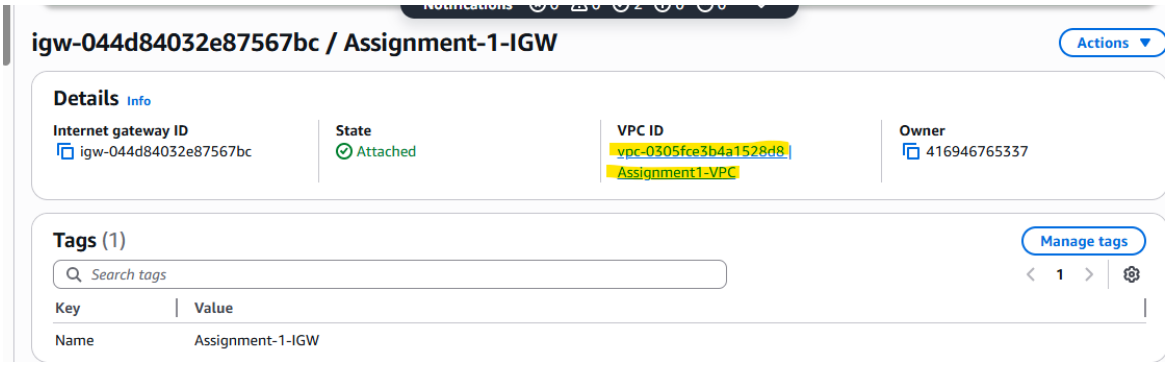




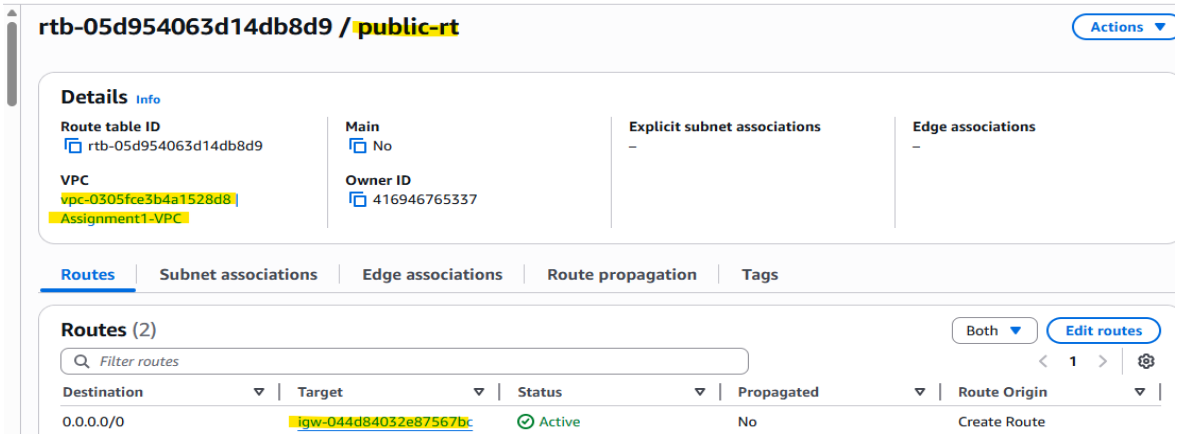
Step 2: Create 1 public subnet and 2 private subnets

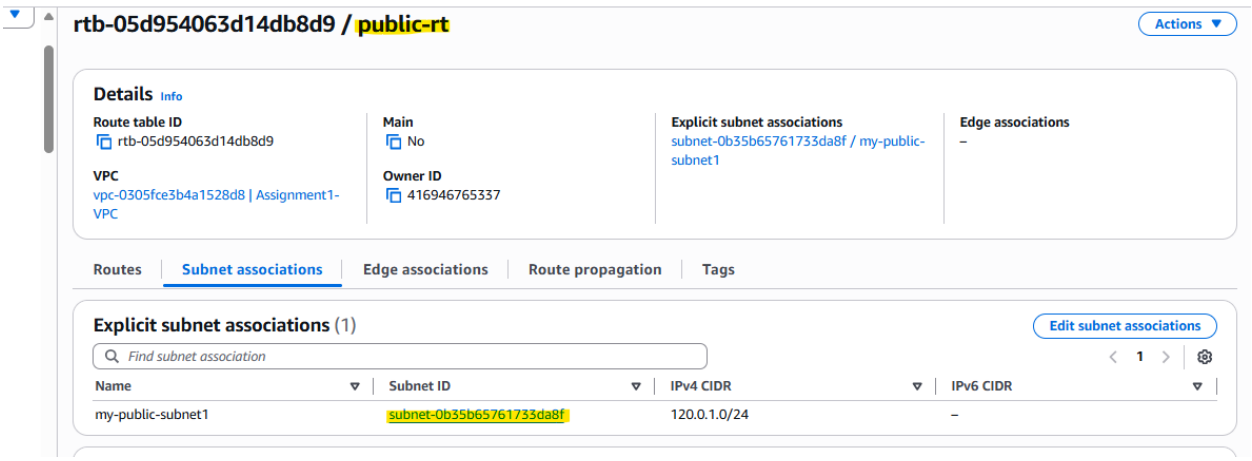


Step 3: Create the Internet Gateway and attach it to VPC

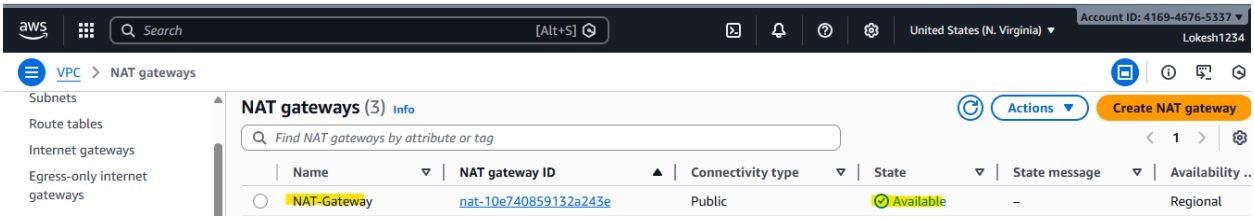


Step 4: Create the route table for public subnet

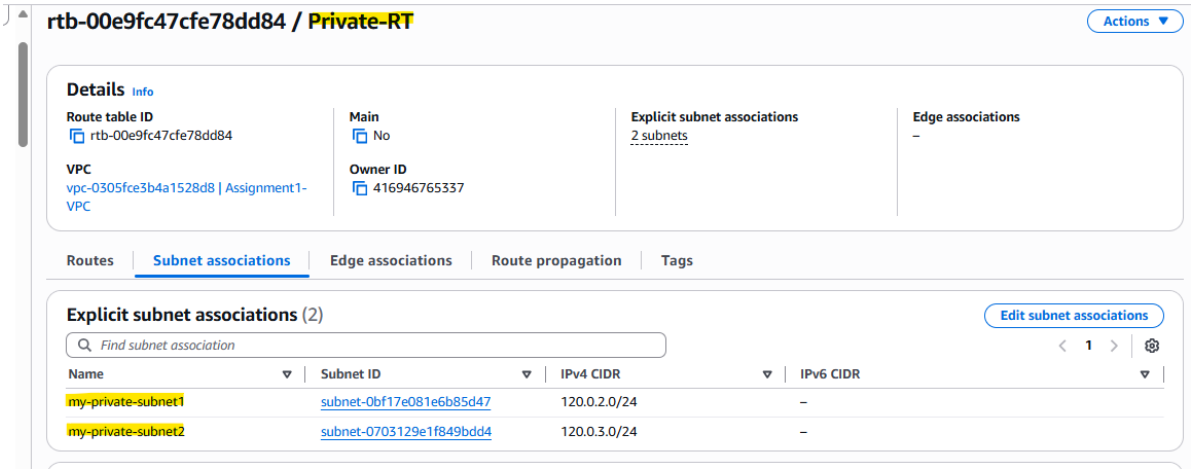




Step 5: Create a NAT gateway

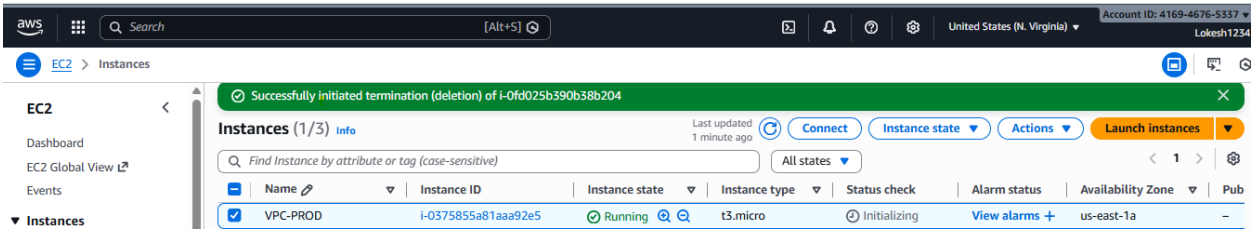


Step 6: Create private route table and associate them with private subnets to it.



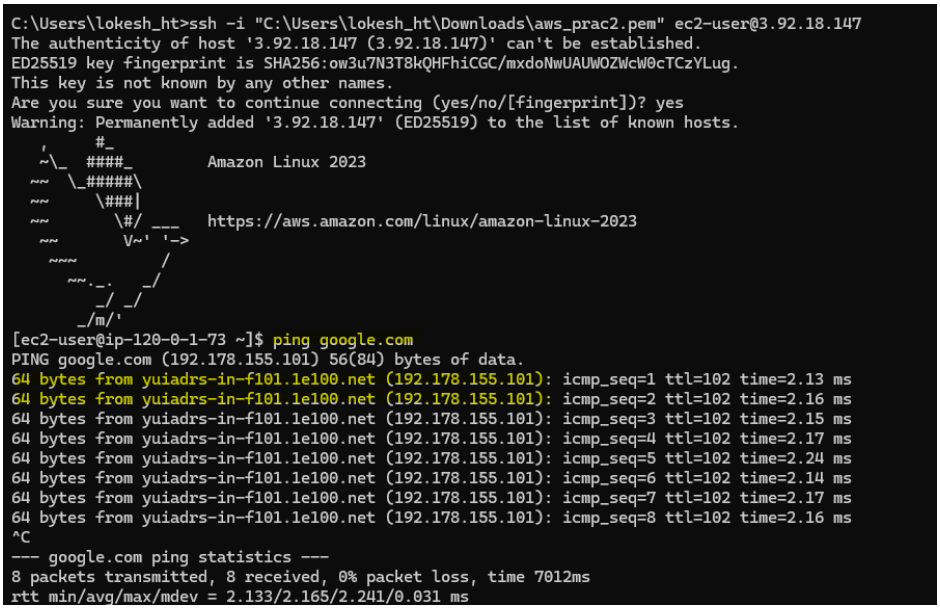
Testing whether we can connect to the Internet through the public subnet

Create a EC2 instance and assign the VPC created use the public subnet assign auto assign public IP and launch the Ec2 instance



Connect from the outside using the ssh command to the Ec2 instance

And ping google.com



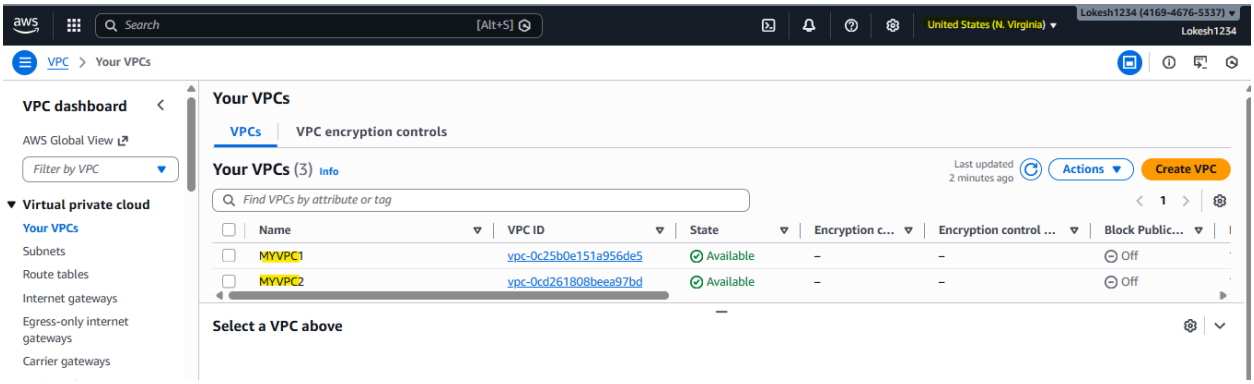
Assignment-2

Problem Statement:

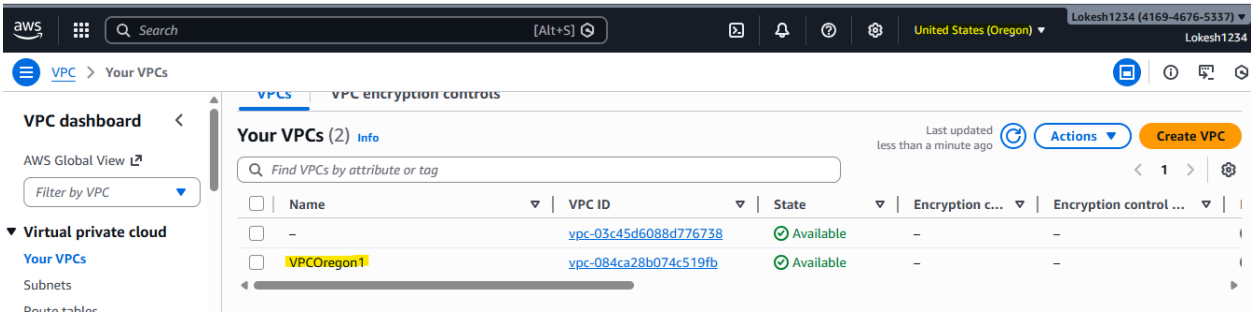
Working for an organization, you are required to provide them a safe and secure environment for the deployment of their resources. They might require different types of connectivity. Implement the following to fulfill the requirements of the company.

Tasks To Be Performed:

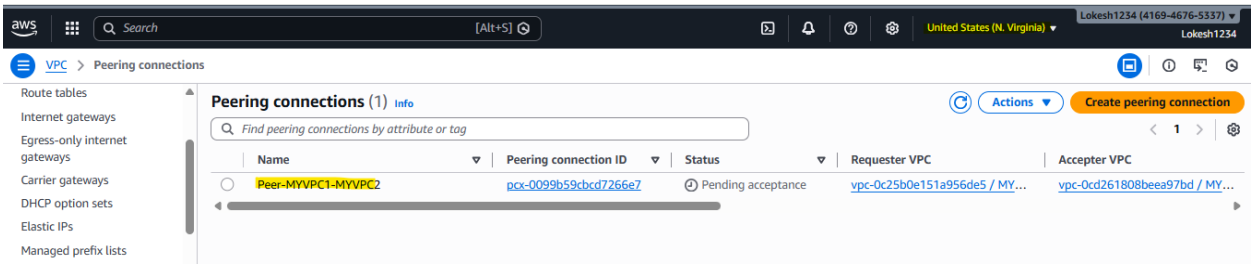
1. Create 2 VPCs in the North Virginia region named MYVPC1 and MYVPC2

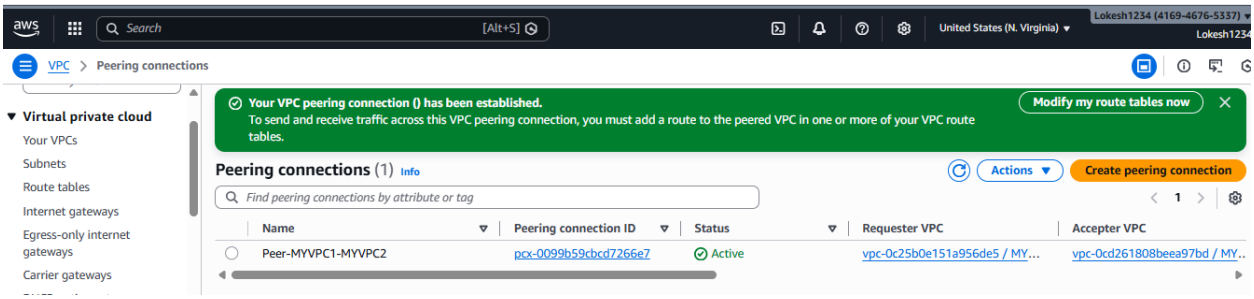


2. Create one VPC in the Oregon region named VPCOregon1

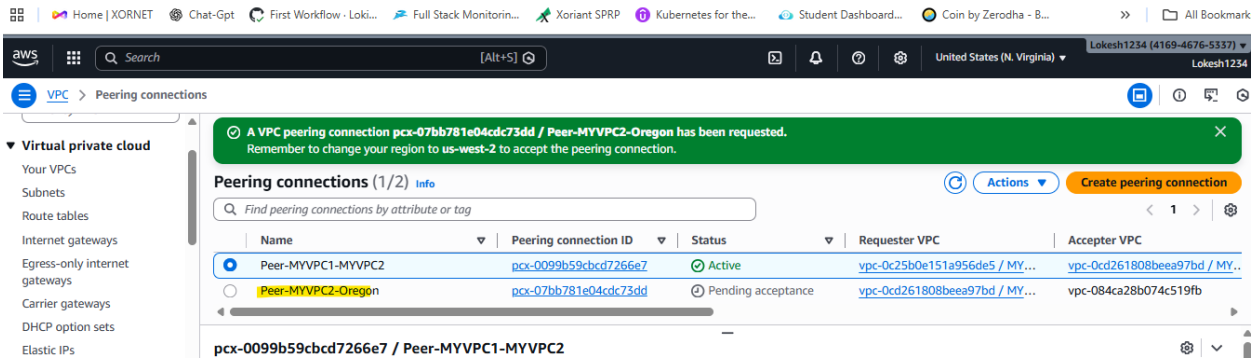


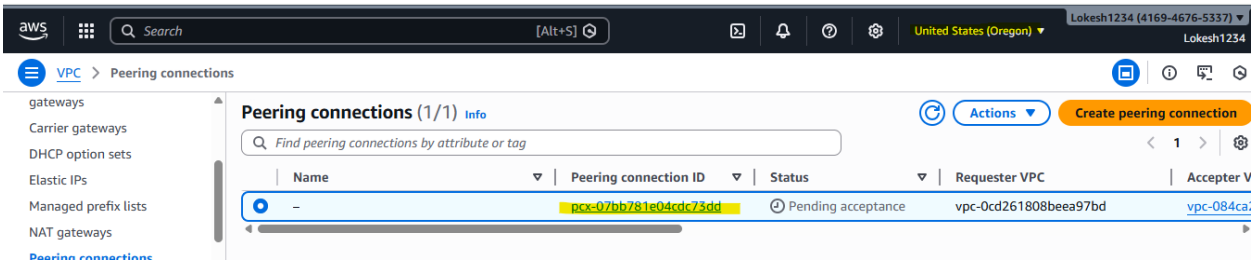
3. Create a peering connection between MYVPC1 and MYVPC2

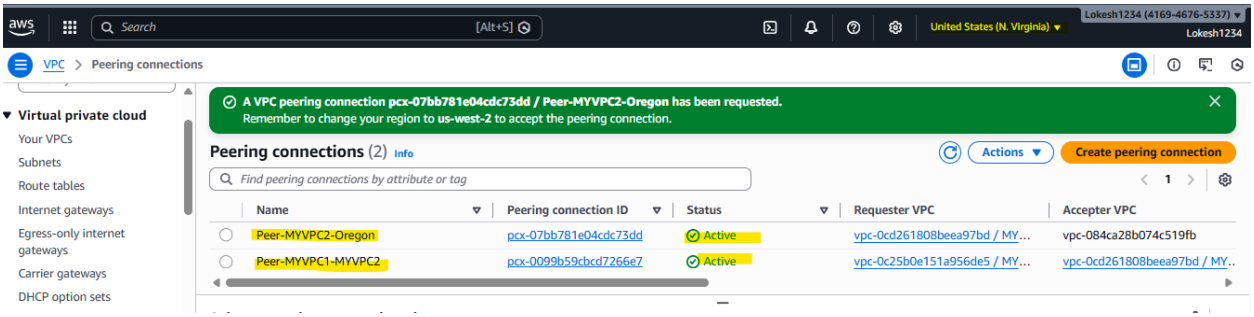




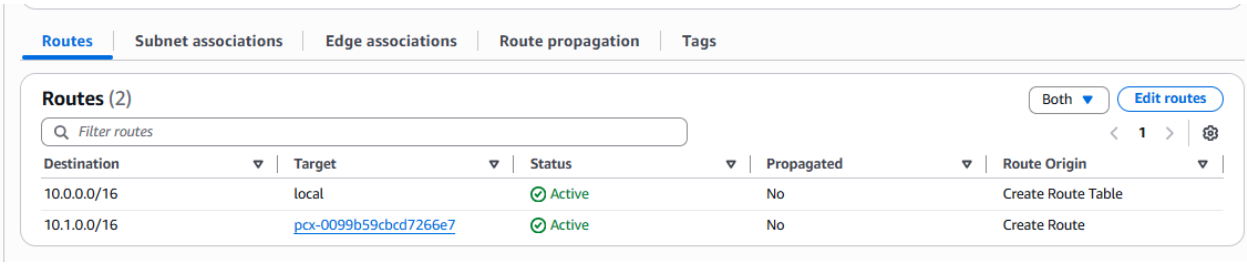
4. Create a peering connection between MYVPC2 and VPCOregon1



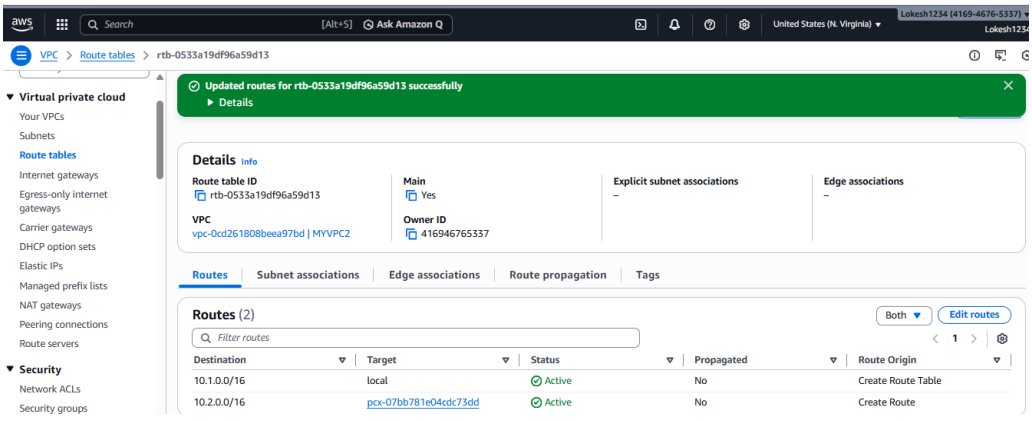


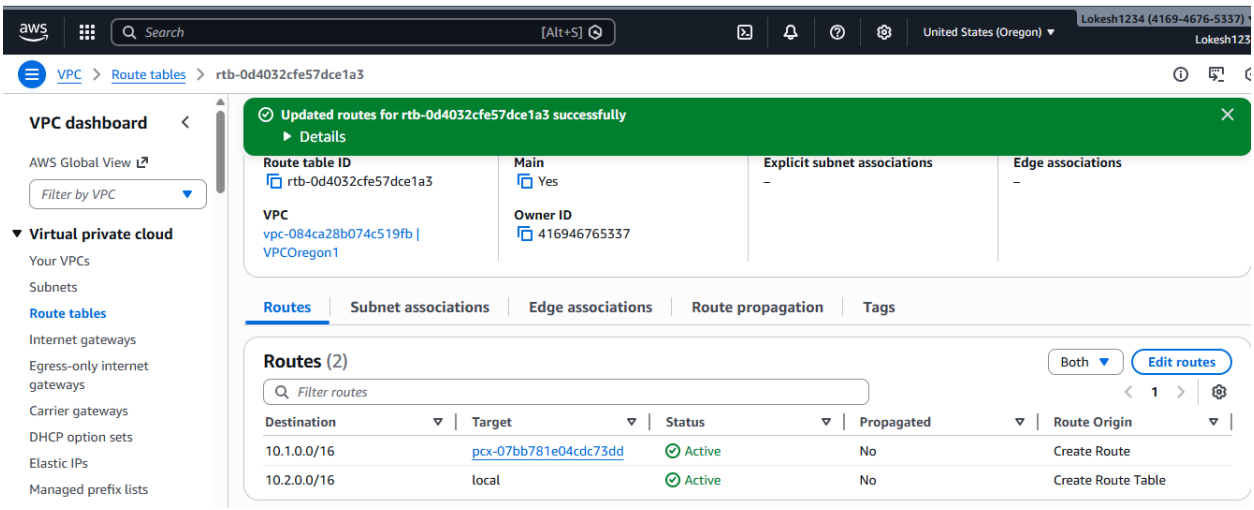


Create route table for the traffic flow on VPC1 and VPC2



Create route table to Oregon VPC from VPc2





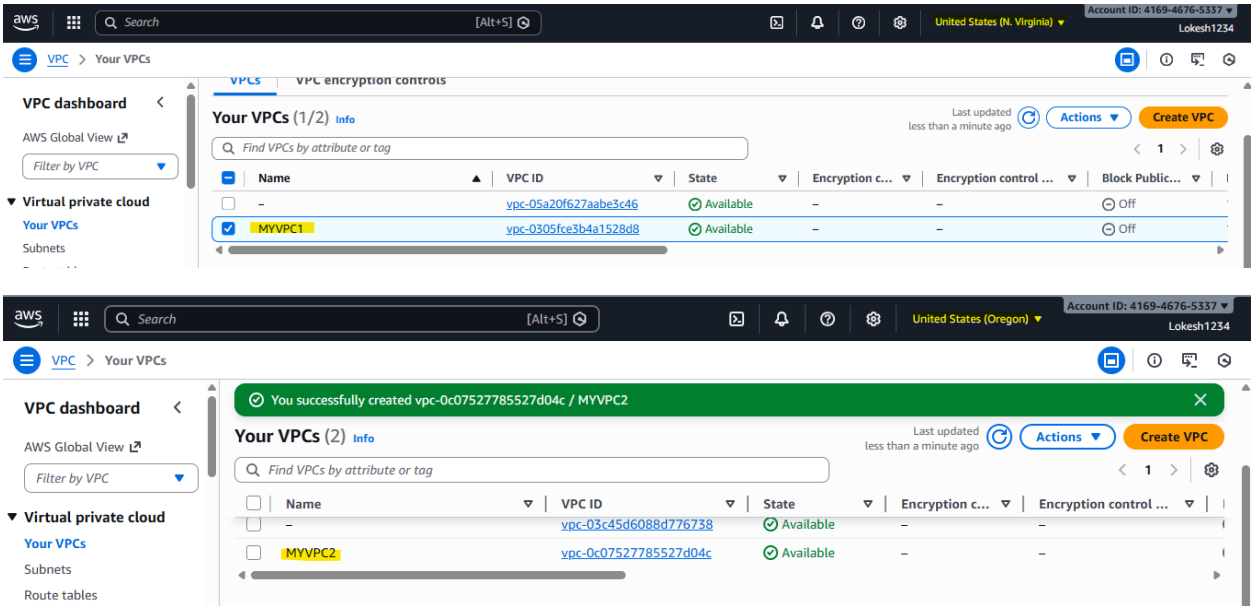
Assignment 2

Working for an organization, you are required to provide them with a safe and secure environment for the deployment of their resources. They might require different types of connectivity. Implement the following to fulfill the requirements of the company.

Tasks To Be Performed:

1. Create 2 VPCs in the North Virginia region named MYVPC1 and MYVPC2
2. Create one VPC in the Oregon region named VPCOregon1
3. Create a peering connection between MYVPC1 and MYVPC2
4. Create a peering connection between MYVPC2 and VPCOregon1

Step1 : Create two VPC's one in N virginia and one in oregon



Step 2: Create peering b/w VPC1 and VPC2

VPC → Peering Connections → Create

Name: Peering-VPC1-VPC2

Requester VPC: MYVPC1

Accepter VPC: MYVPC2

Once the peering request is done from the N Virginia region accept the request for peering same on Oregon region.

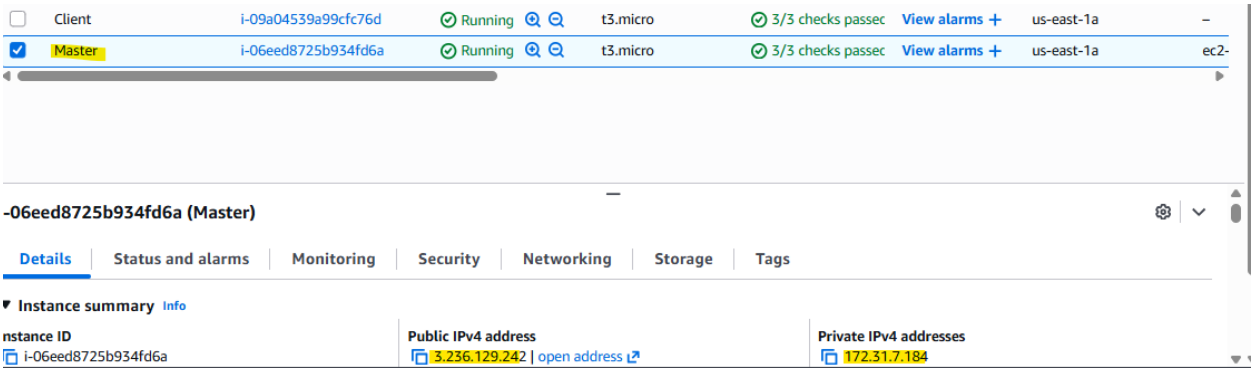
Assignment 3

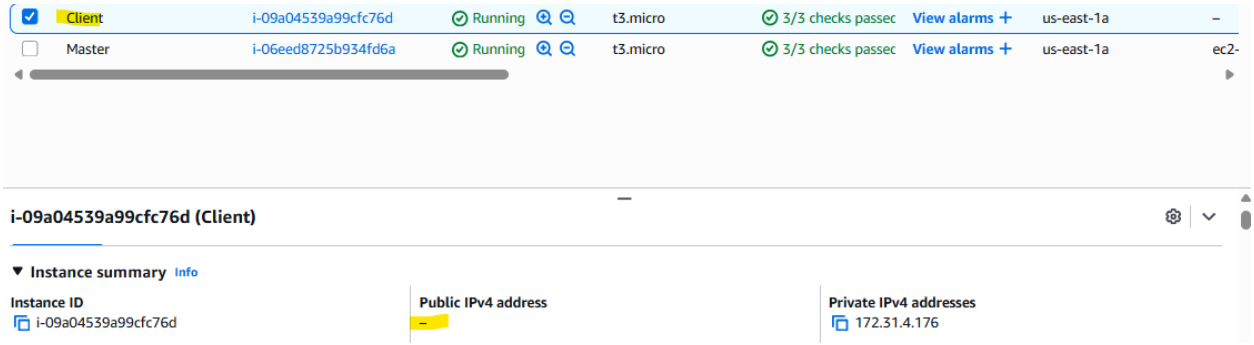
Problem Statement:

Working for an organization, you are required to provide them a safe and secure environment for the deployment of their resources. They might require different types of connectivity. Implement the following to fulfill the requirements of the company.

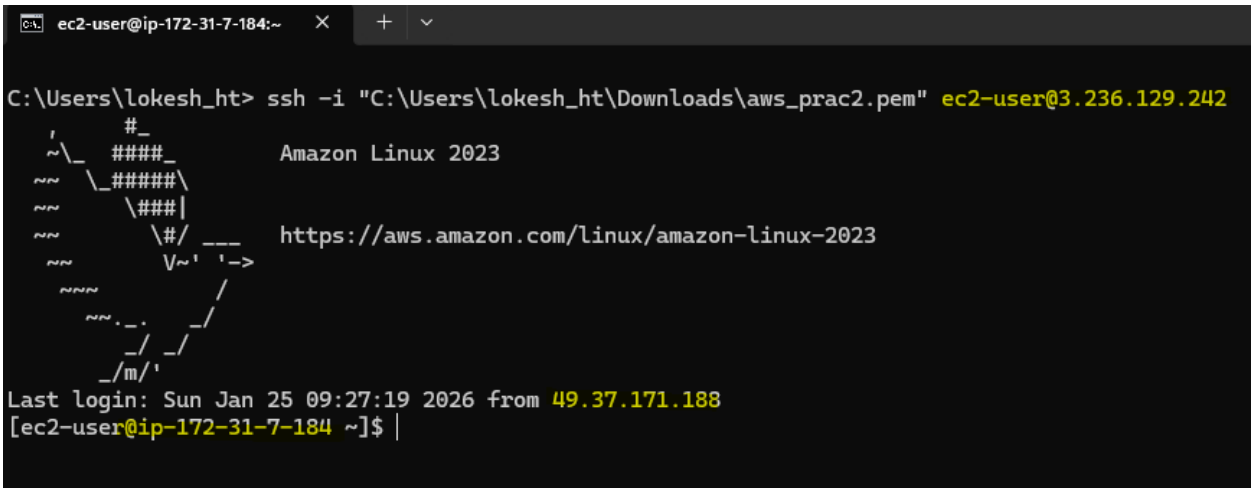
Tasks To Be Performed:

1. Create 2 EC2 instances in any public subnet of any VPC and name them Master and Client.
2. Using security groups, make sure that the Client instance can only be accessed (SSH) through the Master instance.





Connect to the Master Ec2 using the public Ip



Copy the pem key to the linux m/c of master instance

```
PS C:\Users\lokesh_ht> scp -i "C:\Users\lokesh_ht\Downloads\aws_prac2.pem" "C:\Users\lokesh_ht\Downloads\aws_prac2.pem"
ec2-user@3.236.129.242:~/
aws_prac2.pem                                                 100% 1834     7.5KB/s   00:00
PS C:\Users\lokesh_ht>
```

Change the permission for pem key

```
Last login: Sun Jan 25 09:45:50 2026 from 49.37.171.188
[ec2-user@ip-172-31-7-184 ~]$ chmod 400 aws_prac2.pem
[ec2-user@ip-172-31-7-184 ~]$
```

Connect to the Client Ec2 instance using the private key

```
[ec2-user@ip-172-31-7-184 ~]$ ssh -i aws_prac2.pem ec2-user@172.31.4.176
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Sun Jan 25 09:44:02 2026 from 172.31.7.184
[ec2-user@ip-172-31-4-176 ~]$
```

Assignment 4

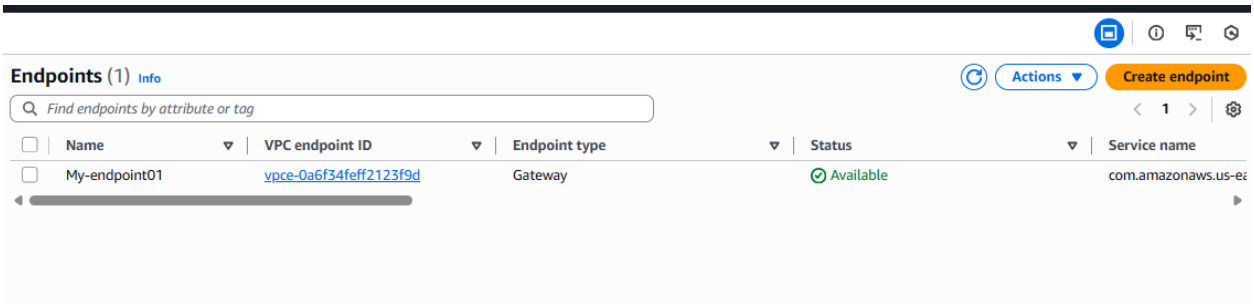
Problem Statement:

Working for an organization, you are required to provide them a safe and secure environment for the deployment of their resources. They might require different types of connectivity. Implement the following to fulfill the requirements of the company.

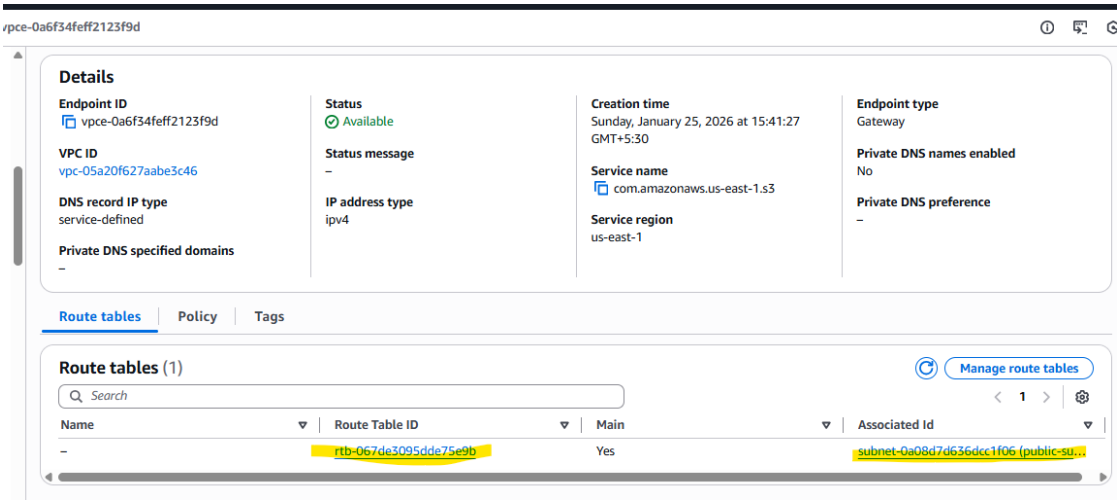
Tasks To Be Performed:

1. Create a VPC endpoint for a S3 bucket of your choice for secure access to the files.

Create a Gateway VPC Endpoint for S3



Choose the route tables associated with the subnets where your EC2 instances run.



Attach a policy (IAM-like JSON) to control access. Example: allow full S3 access:

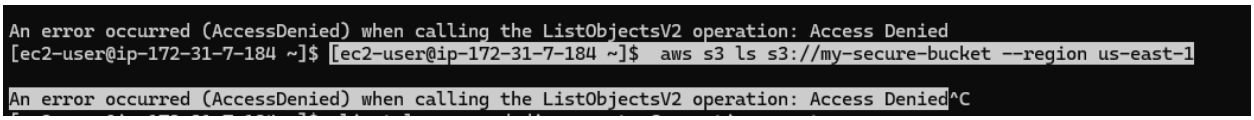
```
{
  "Statement": [
    {
      "Action": "*",
      "Effect": "Allow",
      "Resource": "*",
      "Principal": "*"
    }
  ]
}
```

In the S3 bucket policy, restrict access to only requests coming via the VPC endpoint.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Principal": "*",
```

```
      "Action": "s3:*",
      "Effect": "Allow",
      "Resource": [
        "arn:aws:s3:::my-secure-bucket",
        "arn:aws:s3:::my-secure-bucket/*"
      ],
      "Condition": {
        "StringEquals": {
          "aws:SourceVpce": "vpce-1234567890abcdef0"
        }
      }
    }
  ]
}
```

Test Connectivity



Assignment 5

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.