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.

6.creating vpc end point for the s3 service and access the objects in any buckets from with in the vpc

Development Network:

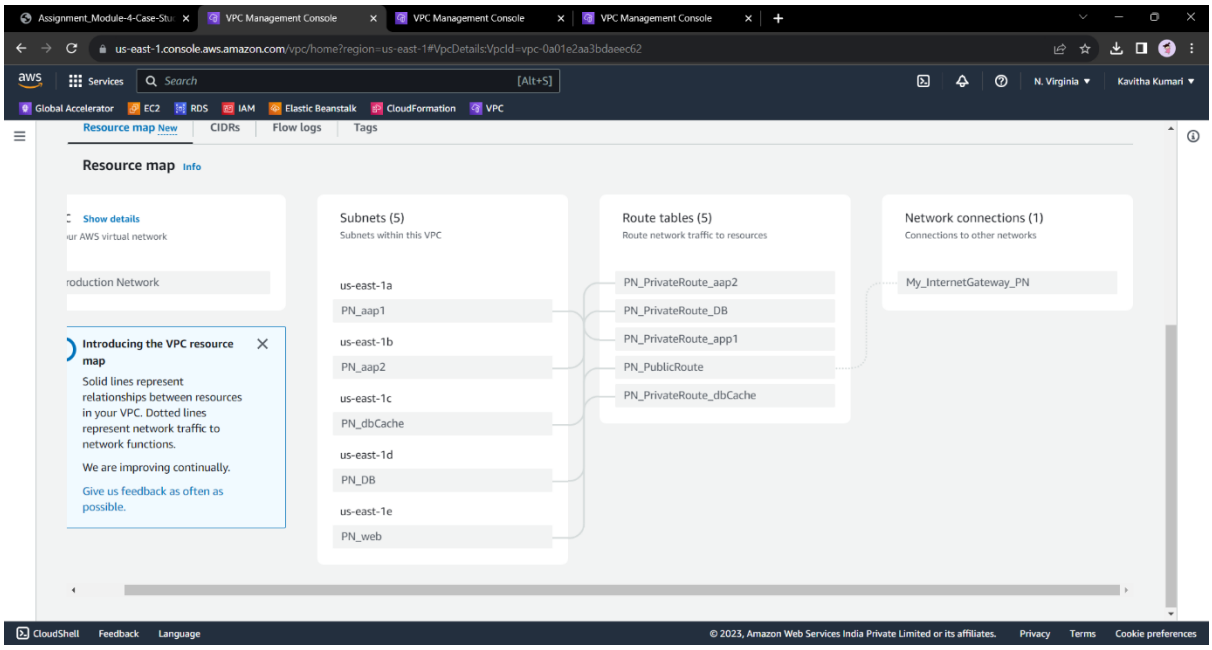
1. Design and build 2-tier architecture with two subnets named web and db and launch instances in both subnets and name them as per the subnet names

2.Make sure only the web subnet can send internet requests.

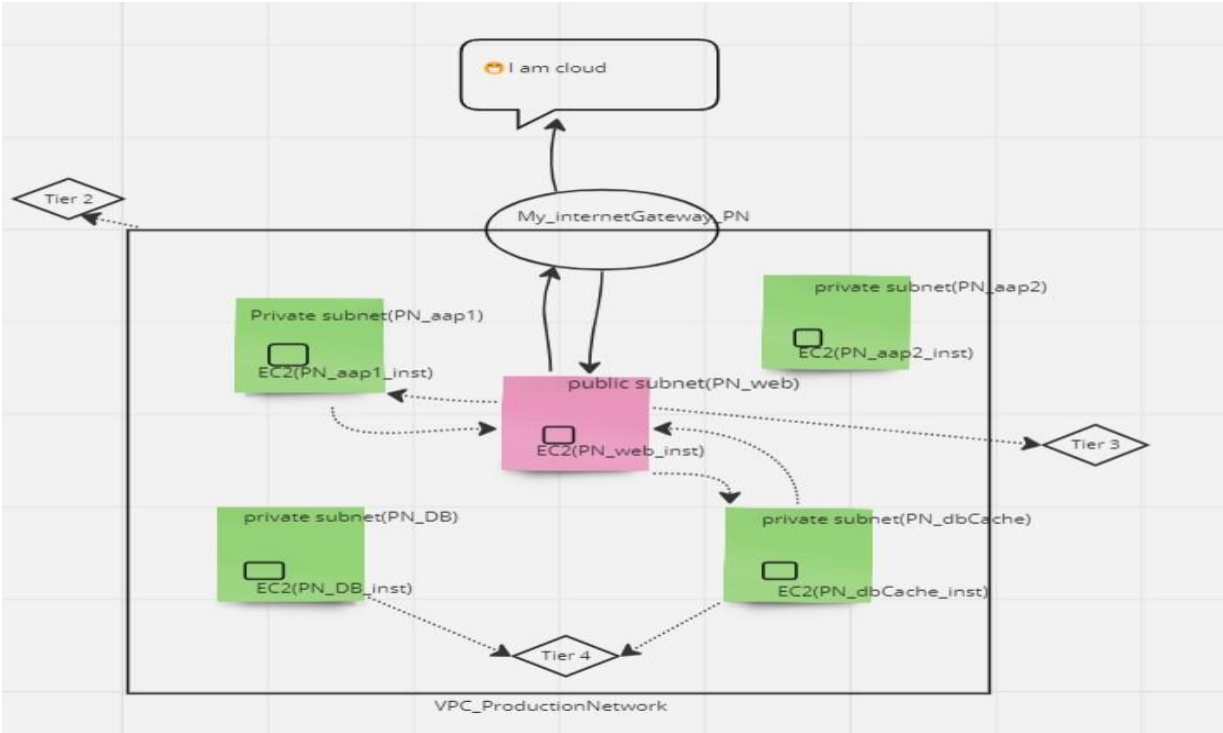
3.Create peering connection between production network and development network.
4.Setup connection between db subnets of both production network and development network respectively.

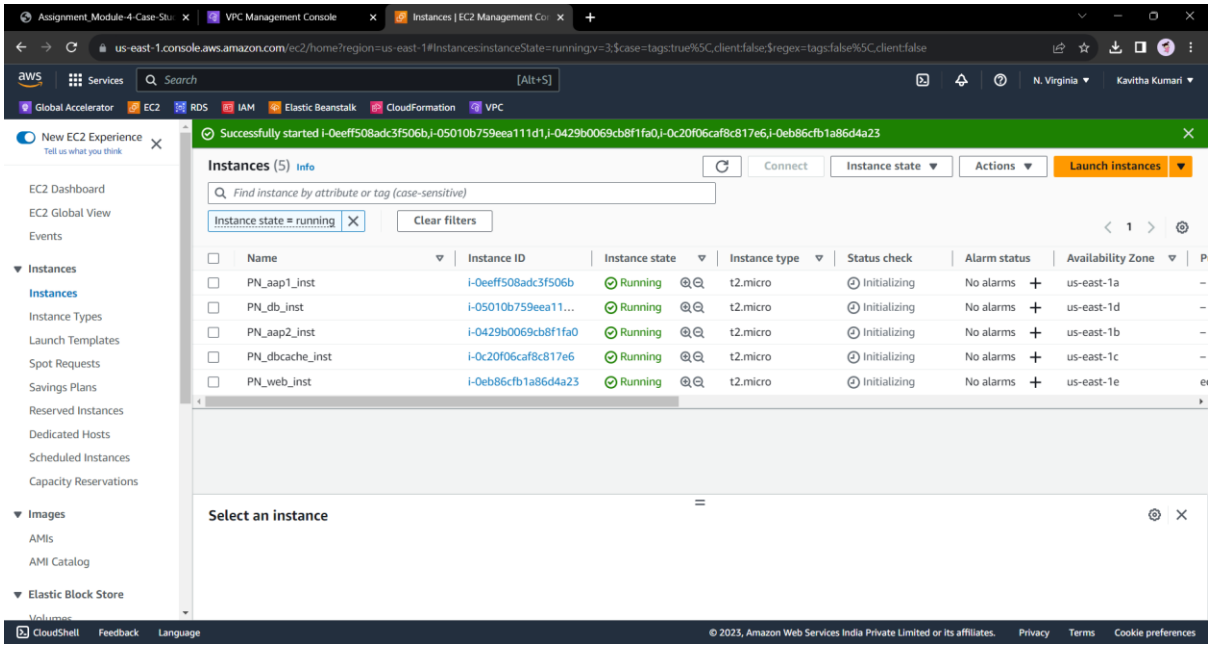
Procedure:

- Below is the 4-tier architecture.
- Create a VPC with the name Production Network. With the 5 subnets with the name as per given in the question.
- Create an internet gateway and attach it to the VPC.
- **Now we will create 5 Route table for each of the subnet. Note that for the public subnet that is web subnet we will give access to internet. Therefore, we will allow internet gateway in the PN_PublicRoute.**

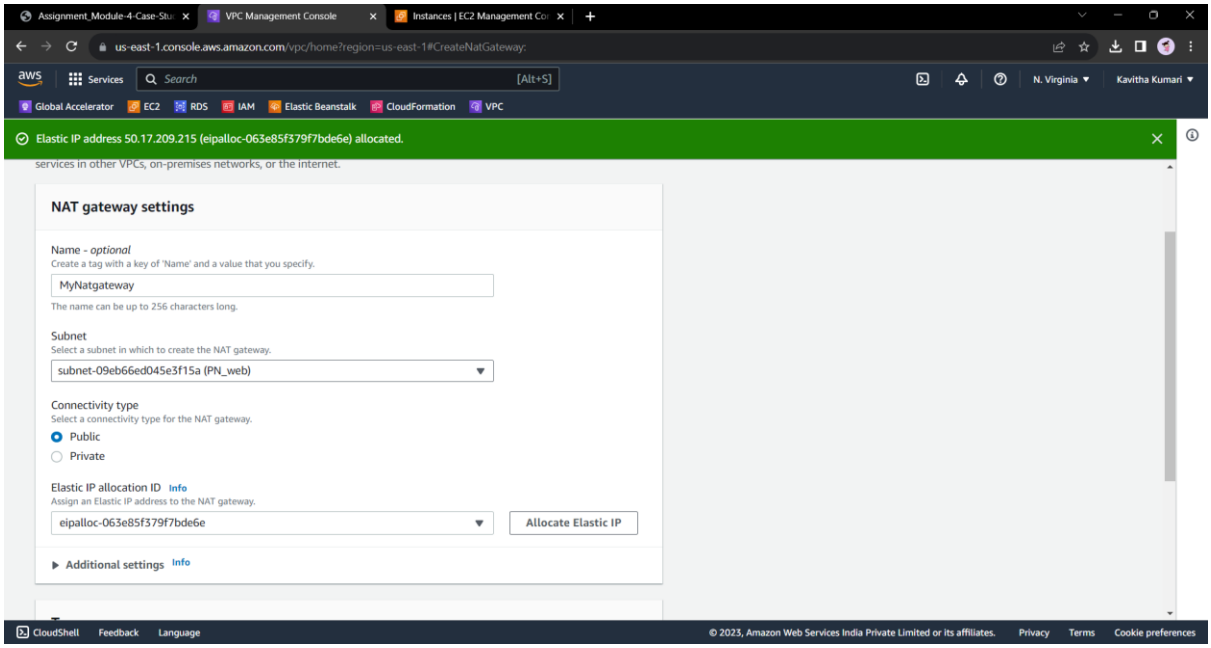


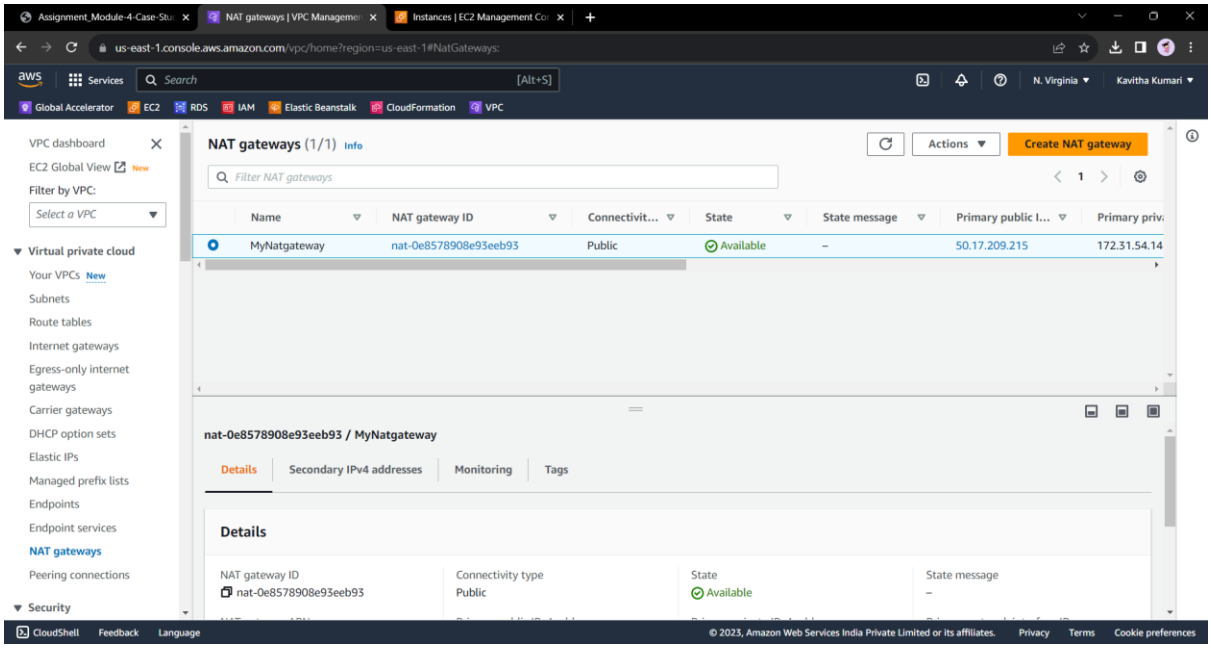
- Now we will create 4 Route table for the respective subnet.
- **Launch instances in all subnets and name them as per the subnet that they have been launched in.**



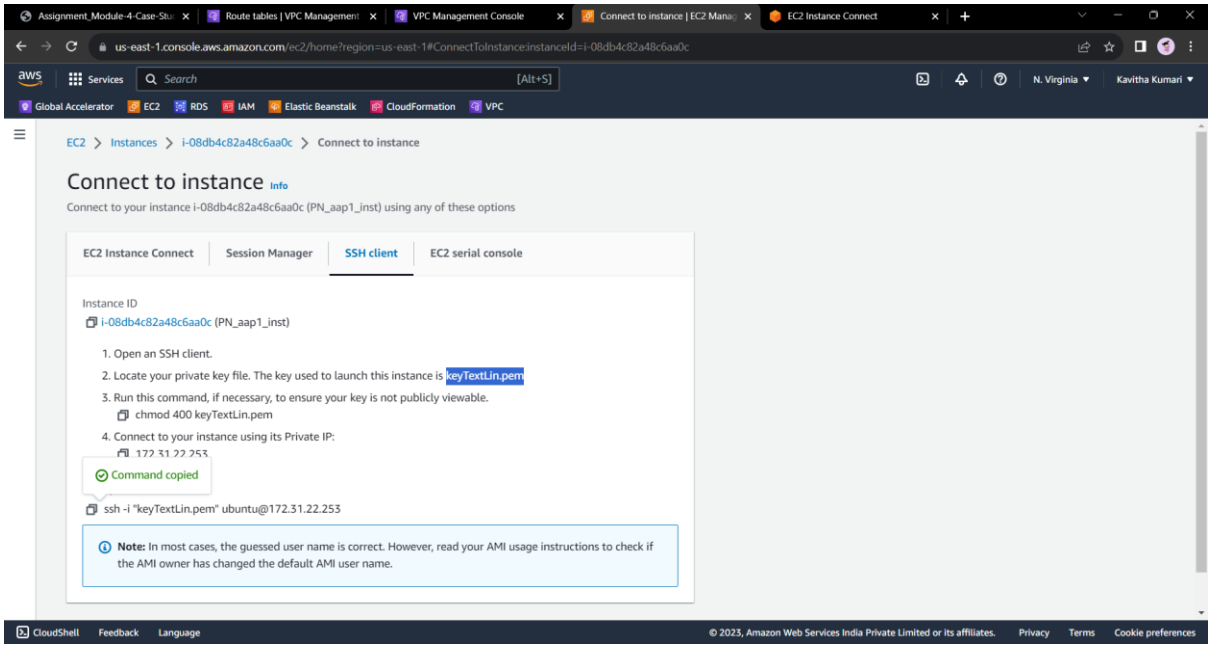


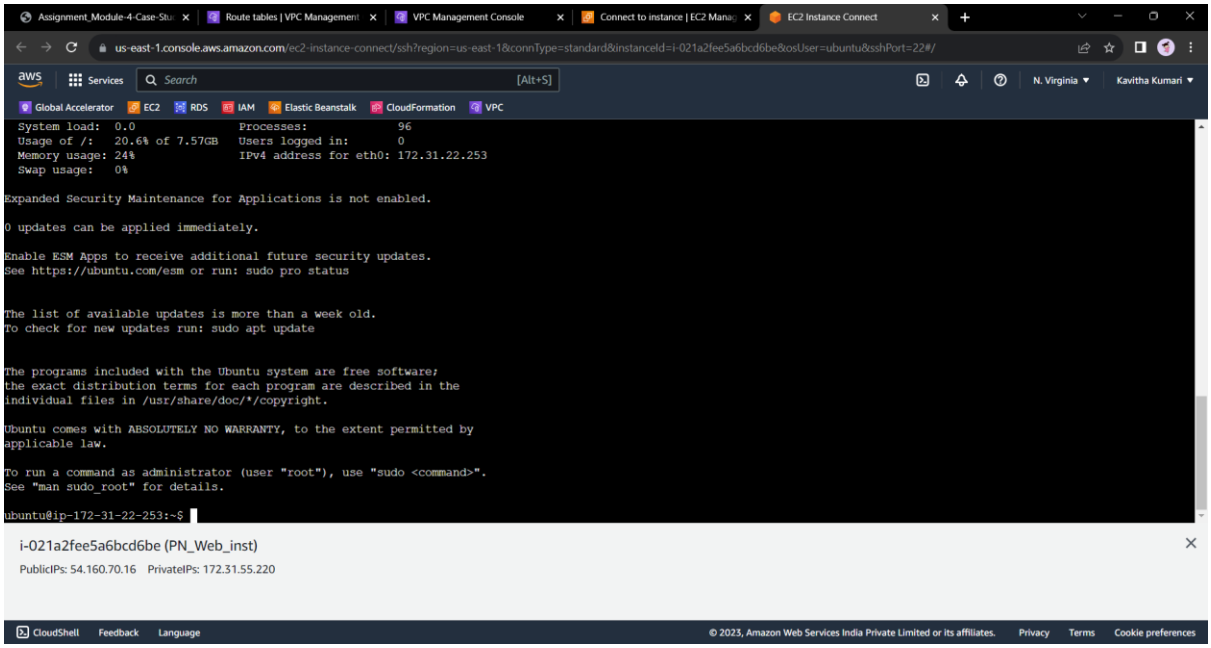
- **Now we need to allow the dbcache inst and the app1 to send internet request.**
- For this we need to create a NAT gateway so that our private instance can access the internet from the public instance in our network.
- In VPC goto Nat gateway and click on create NAT gateway.
- Enter the name of the NAT gateway and choose the public subnet and click on allocate the elastic IP and click on the create NAT gateway.



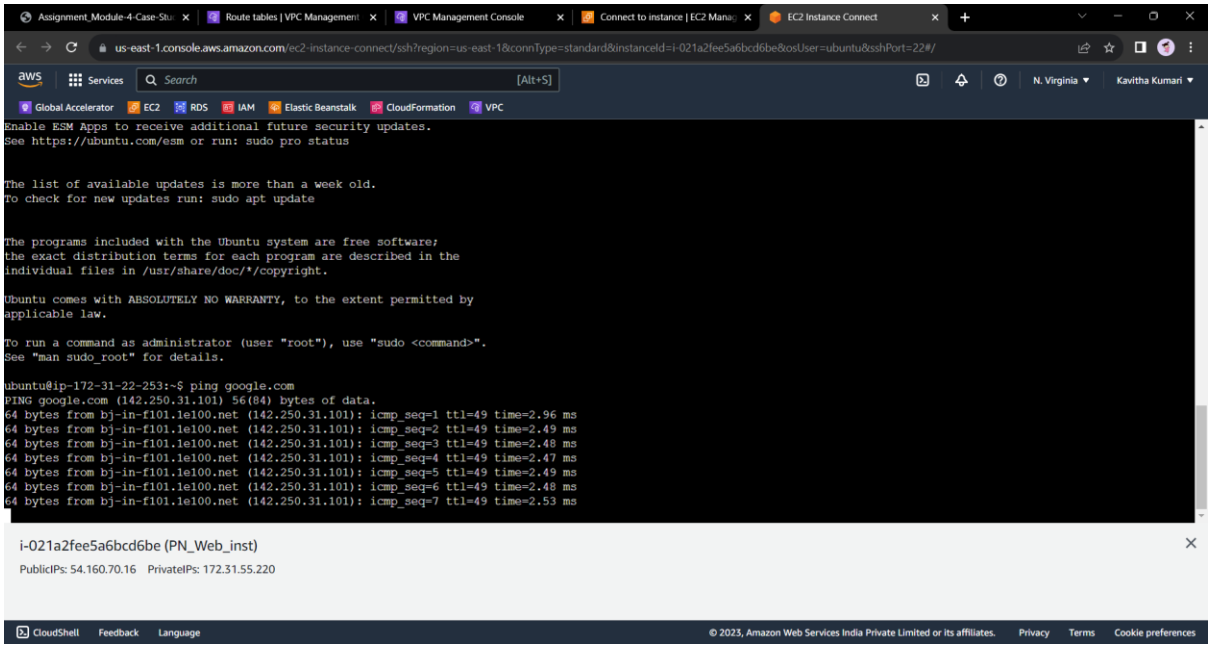


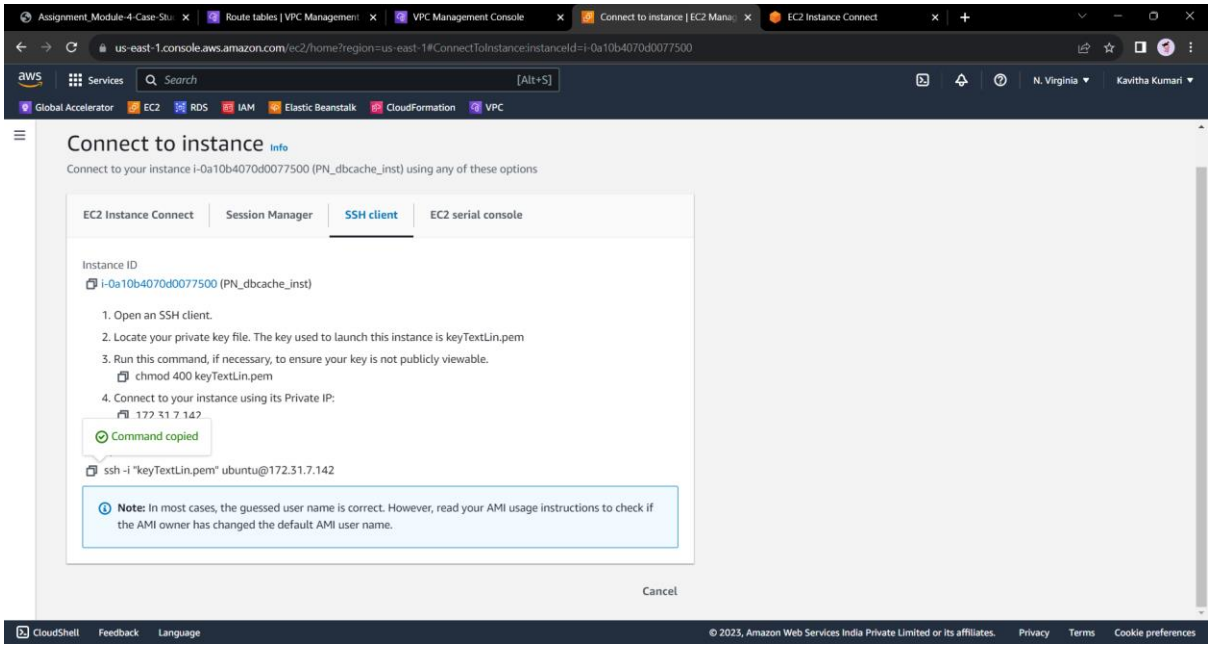
- Goto to the public instance and click on connect and there first we need to create the file with the name of the key which you have. Example we need to run the following command if I have the key ketTextLin.pem
- Sudo nano keyTextLin.pem
- Press ctrl + s and ctrl + x to save and exit.
- Now we need to change the access permission of the file. For this we will run the following command.
- Sudo chmod 400 ketTextLin.pem
- And then ssh in to private instance see the below steps:

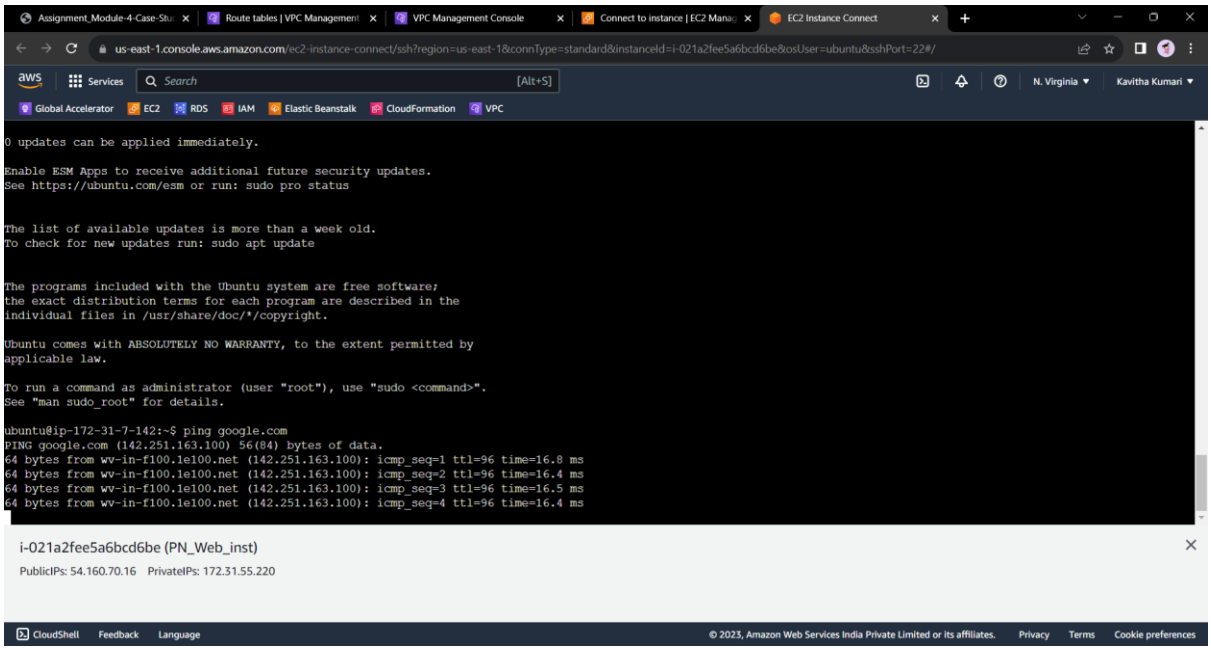




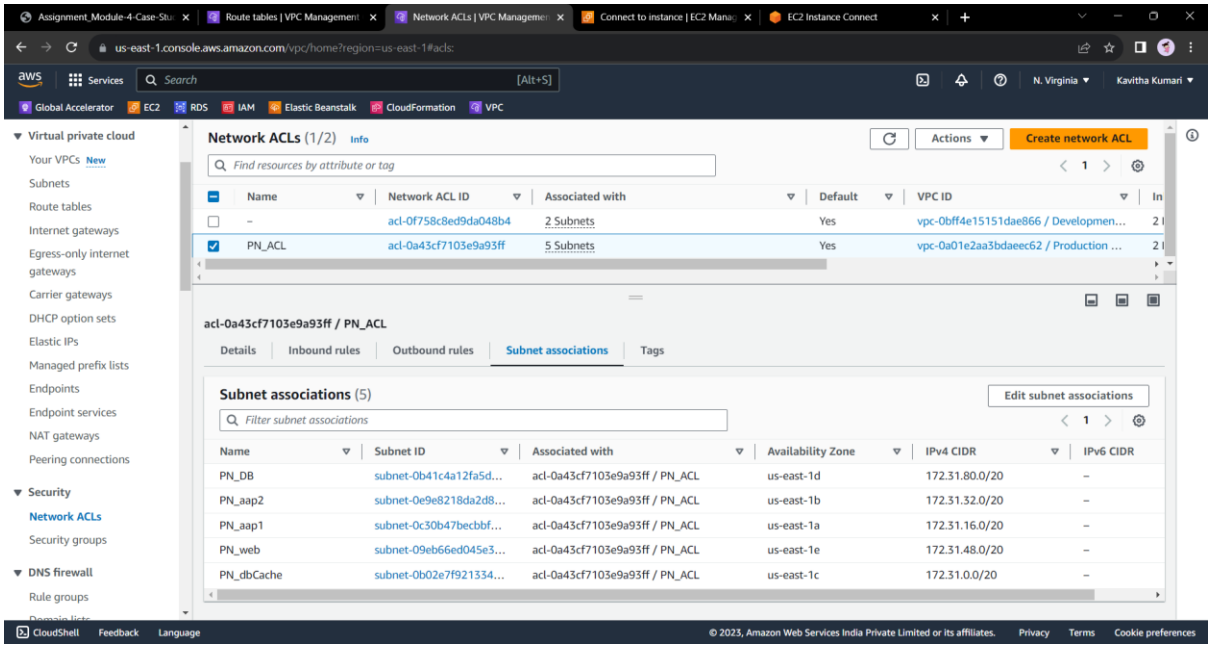
- Therefore, we can get the internet access to the private instance from the google.com with the help of NAT gateway.



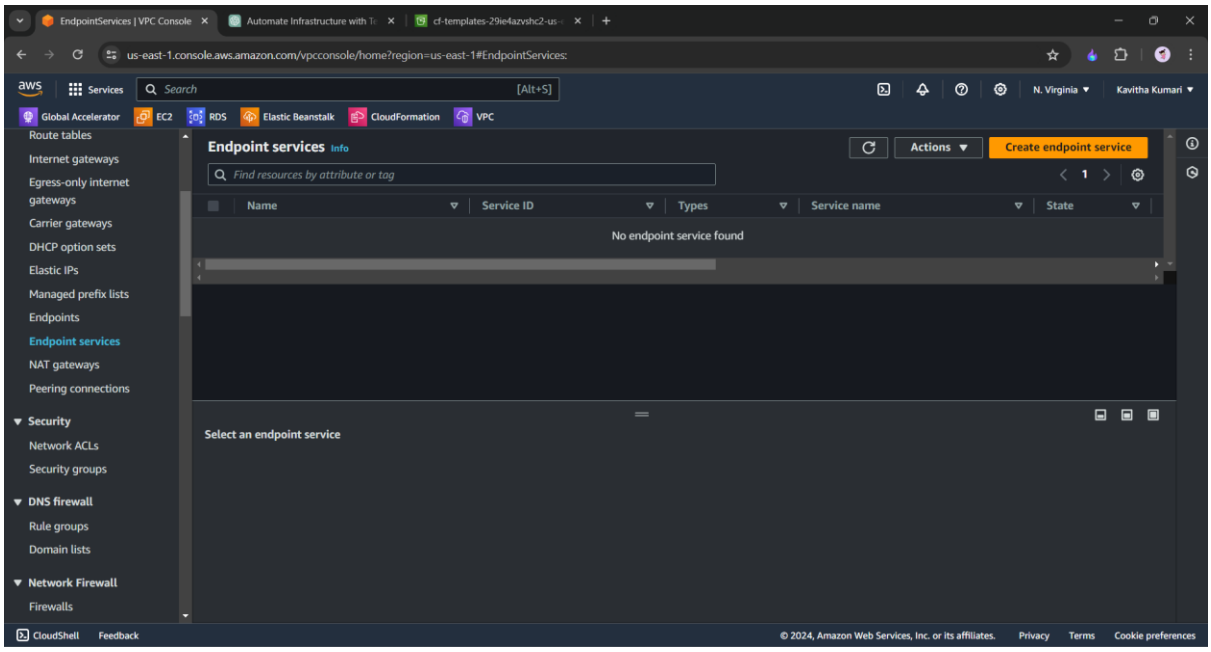




- **Below is how we managing the security groups and NACLs**

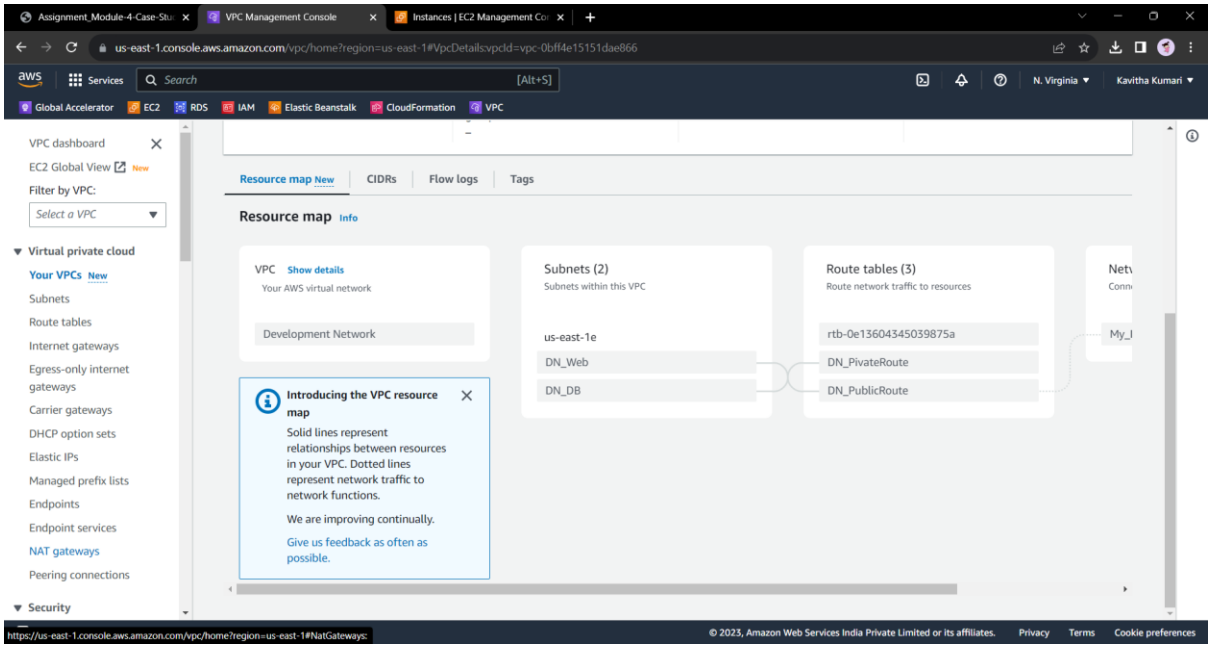


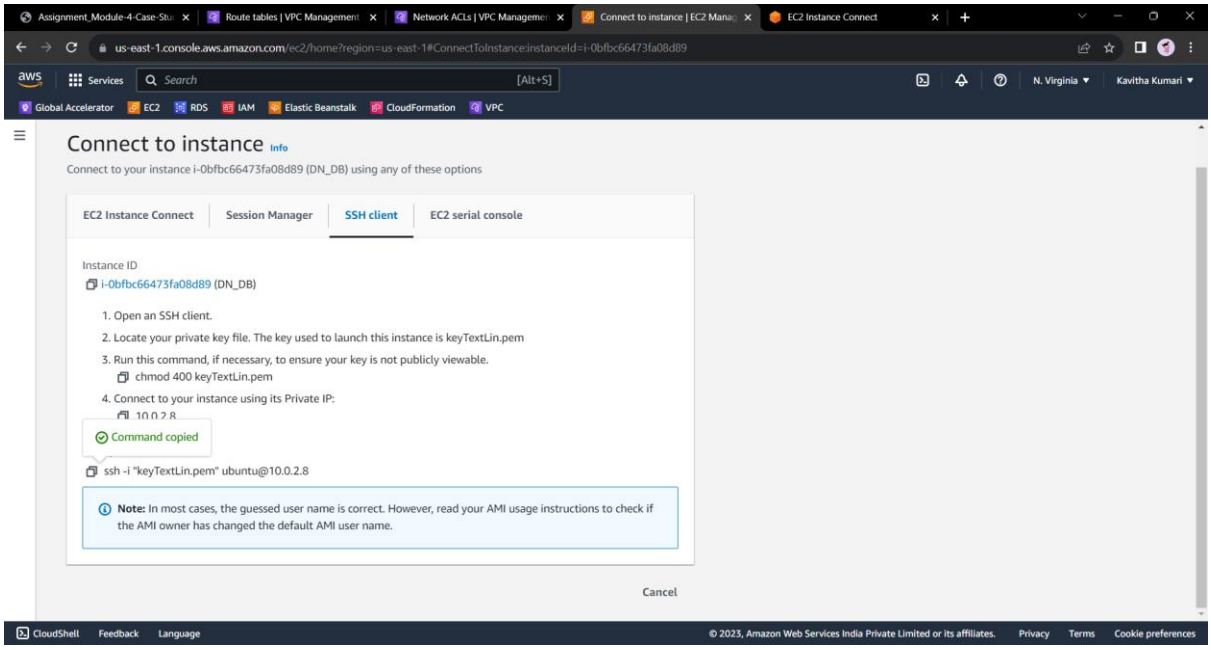
- creating vpc end point for the s3 service and access the objects in any buckets from with in the vpc: -
- **Go to the Amazon VPC console.**
- **In the navigation pane, choose "Endpoints" and then "Create Endpoint."**



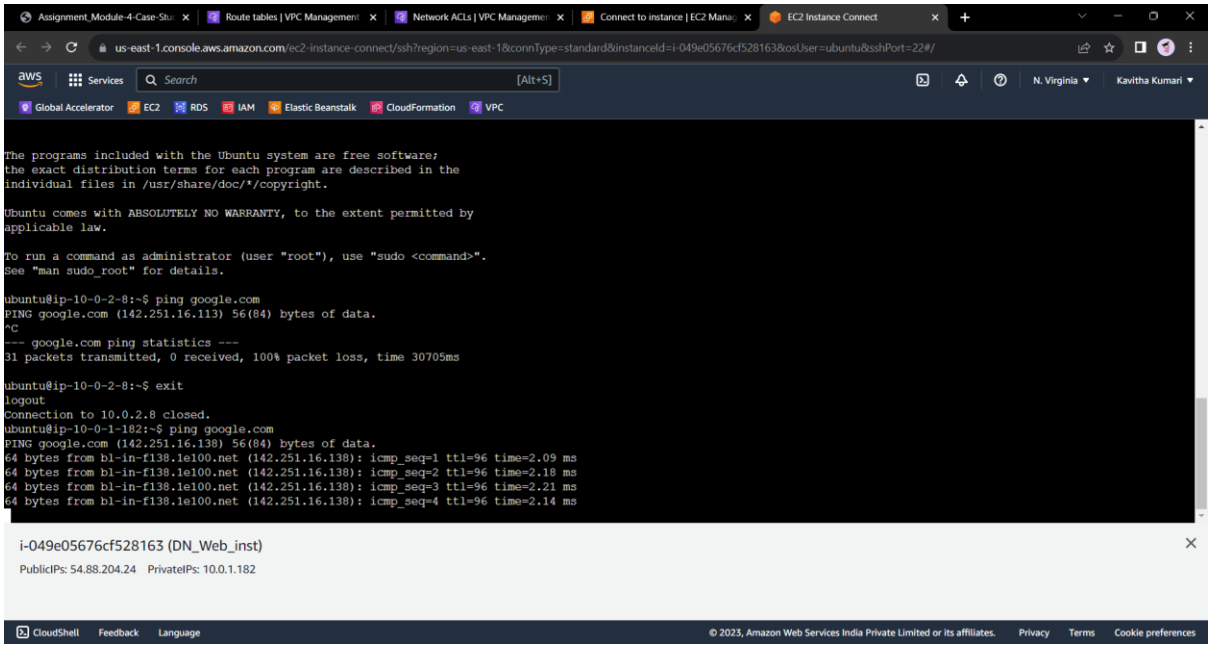
- **Select "AWS services" as the service category.**
- **Choose "com.amazonaws.region.s3" as the service name.**
- **Select the VPC and specify the route table for the endpoint.**
- **Optionally, you can restrict access to specific S3 buckets by specifying the bucket policy.**
- **Update Route Tables:**

- **Ensure that the route tables associated with your VPC have a route for the Amazon S3 endpoint to direct traffic destined for the S3 service to the VPC endpoint.**
- **Update Route Tables:**
- **Ensure that the route tables associated with your VPC have a route for the Amazon S3 endpoint to direct traffic destined for the S3 service to the VPC endpoint.**
- **Update Route Tables:**
- **Ensure that the route tables associated with your VPC have a route for the Amazon S3 endpoint to direct traffic destined for the S3 service to the VPC endpoint.**
- **----------**
- **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**

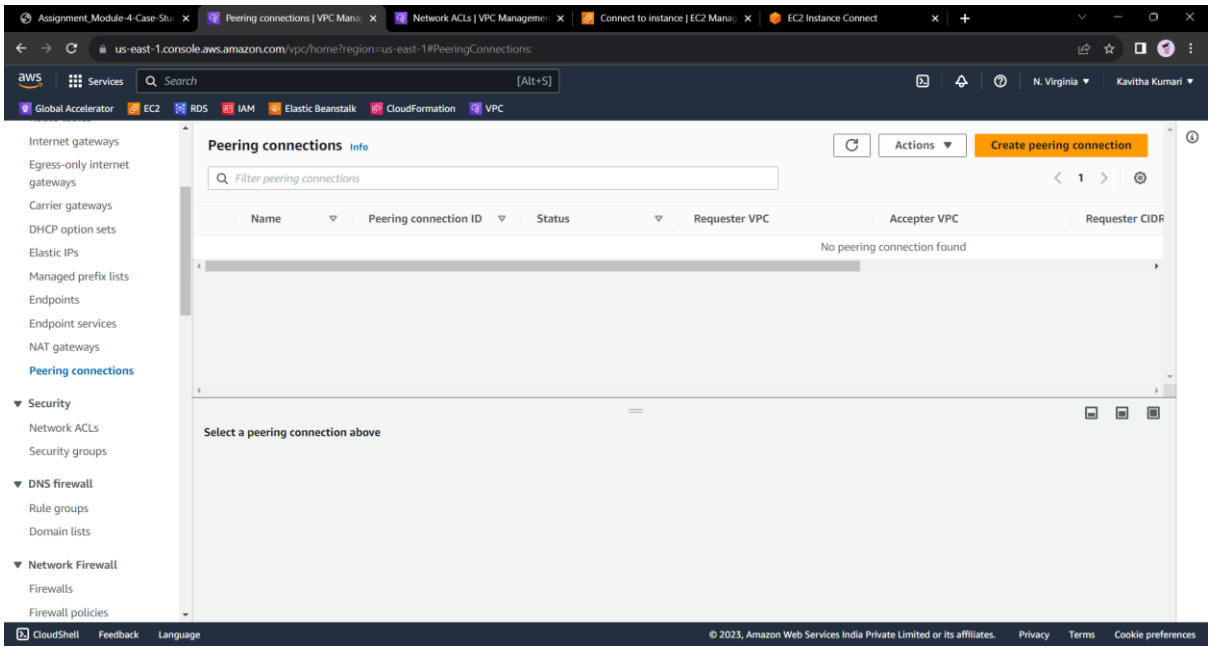




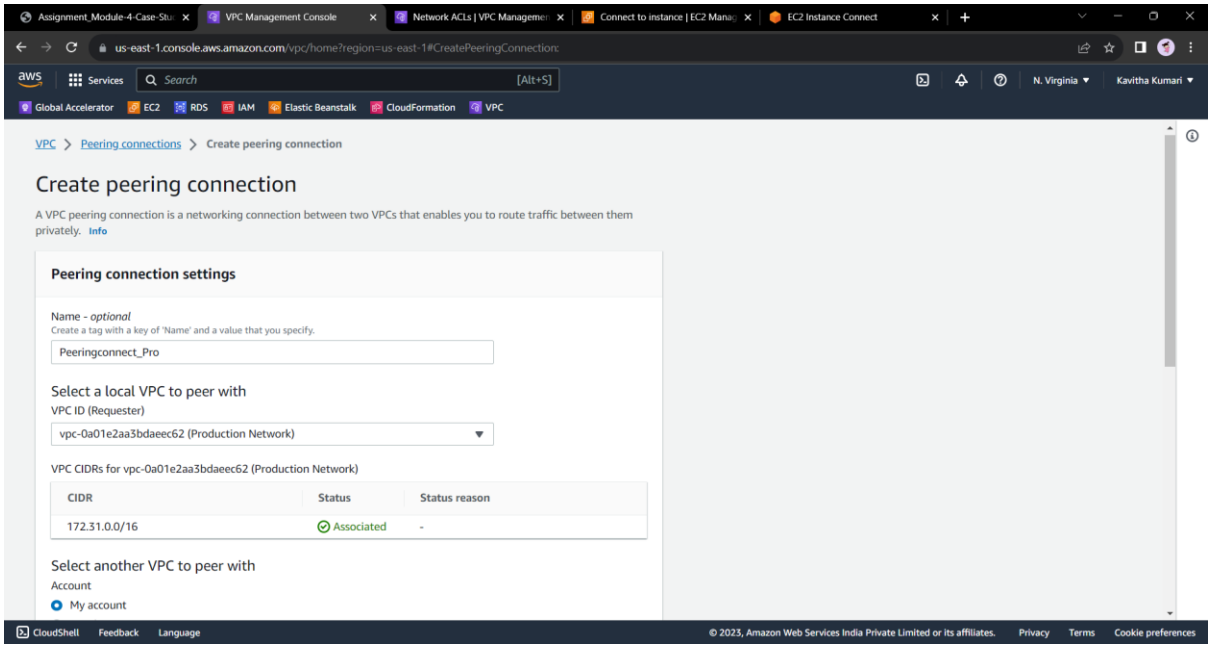
- **Make sure only the web subnet can send internet requests.**

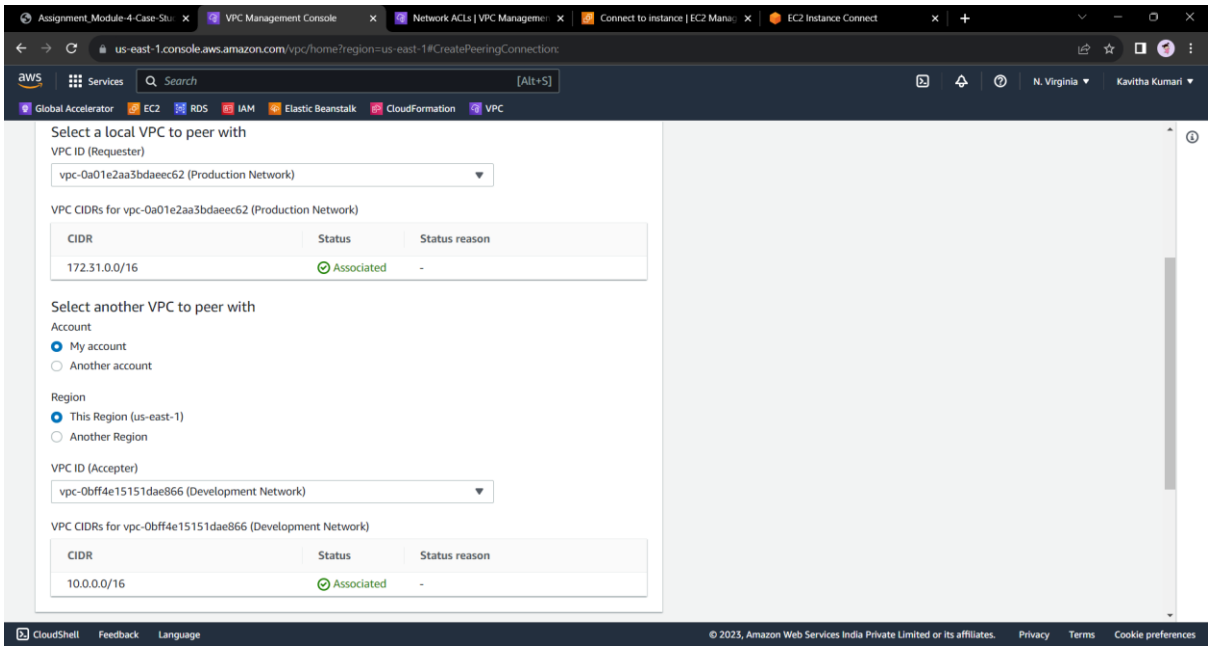


- Here we can see that we are able to ping google.com from the public instance.
- **Create peering connection between production network and development network**
- Here, we need to connect the ProductionNetwork VPC to DevelopmentNetwork through the VPC Peering.
- Click on **create peering connection.**

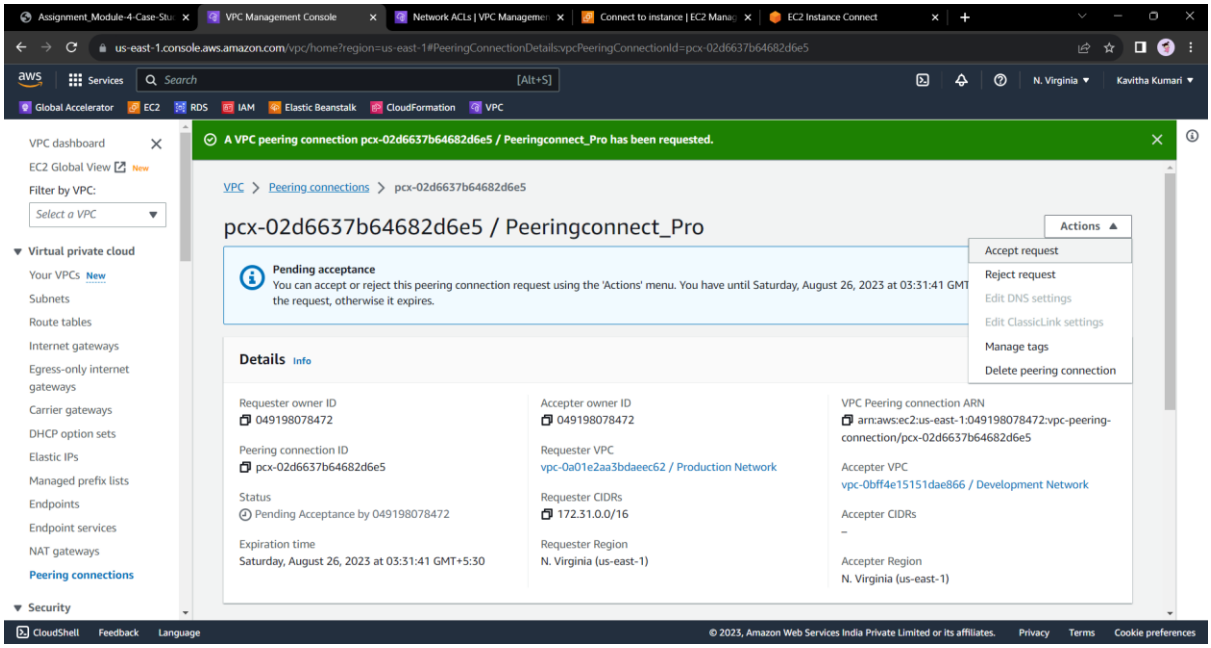


- Fill the details as shown in the following picture.

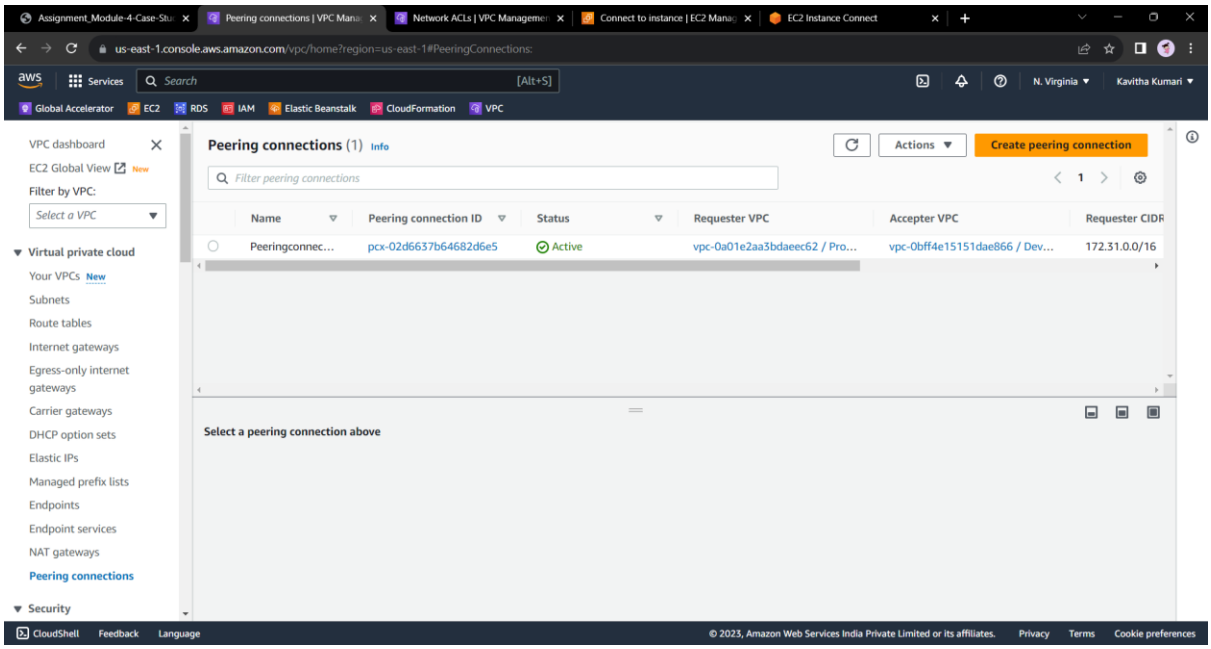




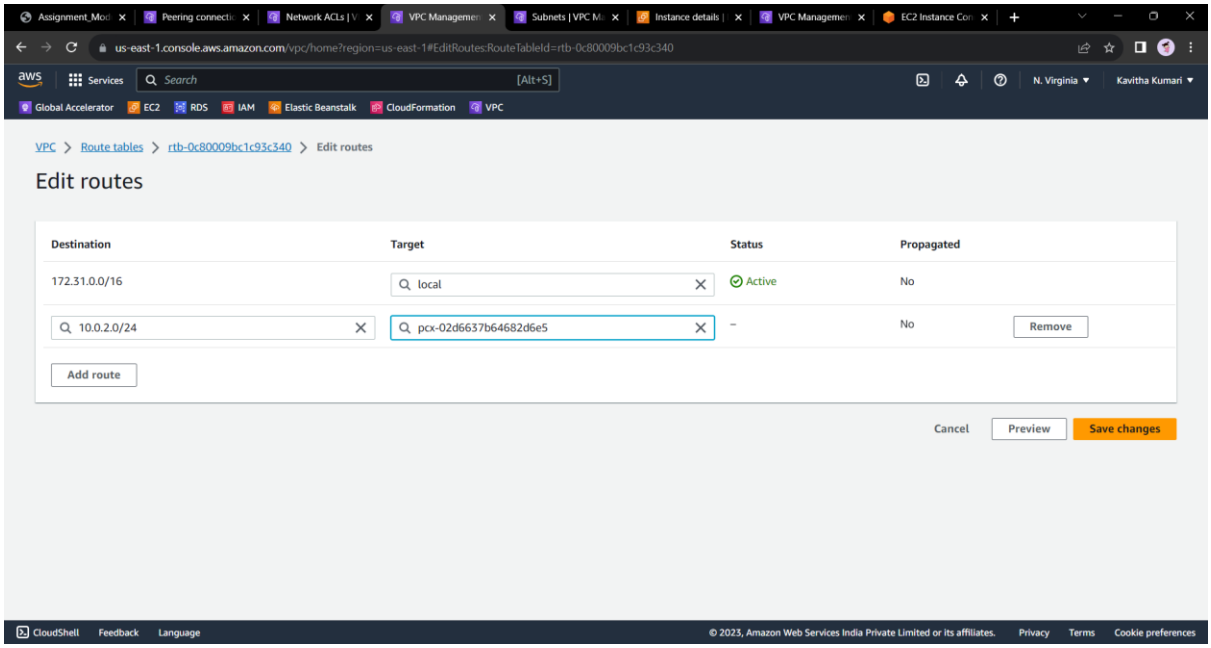
- Click on submit and and then it will be created. To establish the connection, we also need to accept the request as shown below.

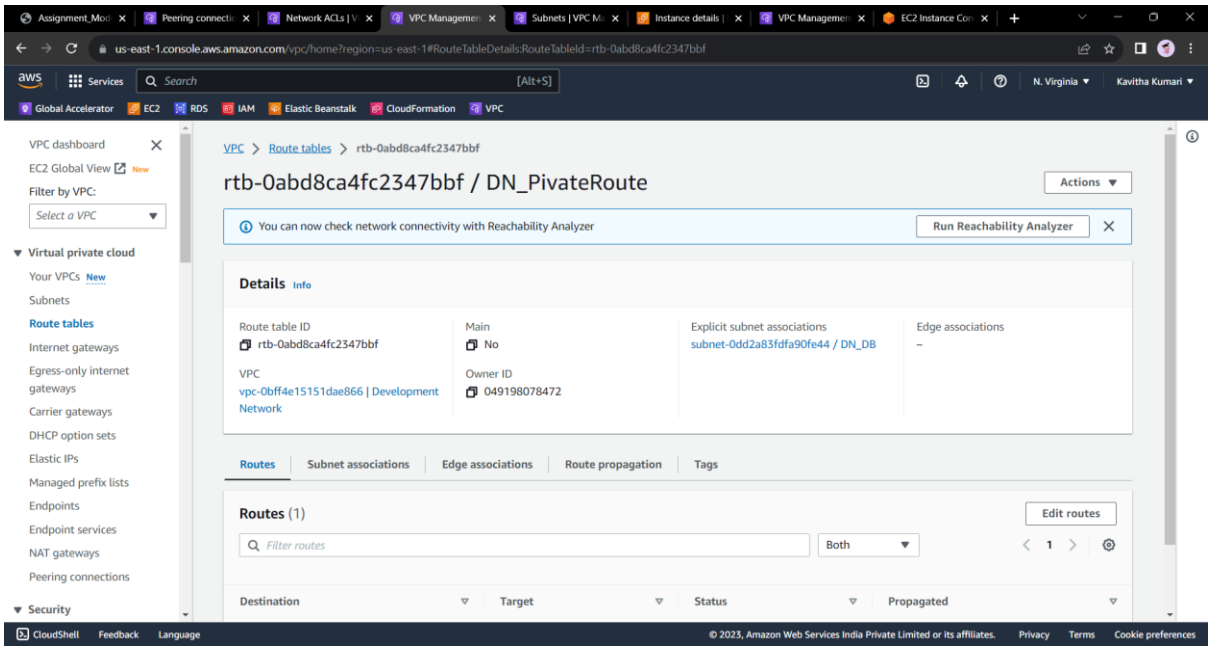


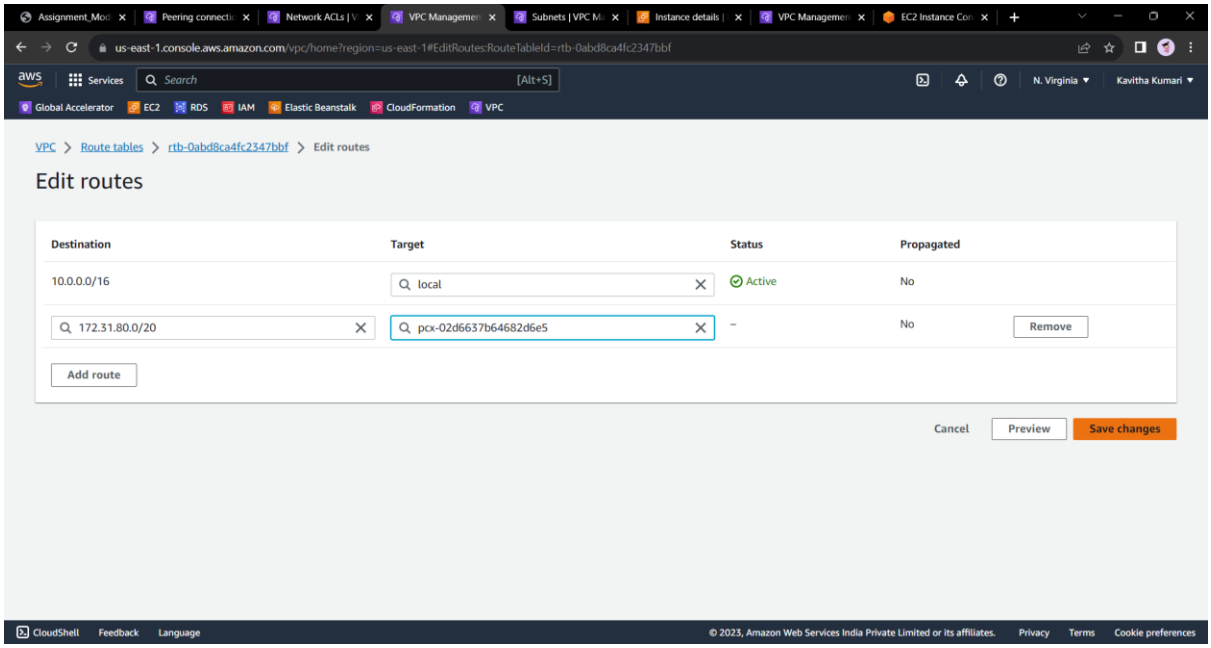
- Once we accept the request, we will see the status as Active as shown in the following picture.



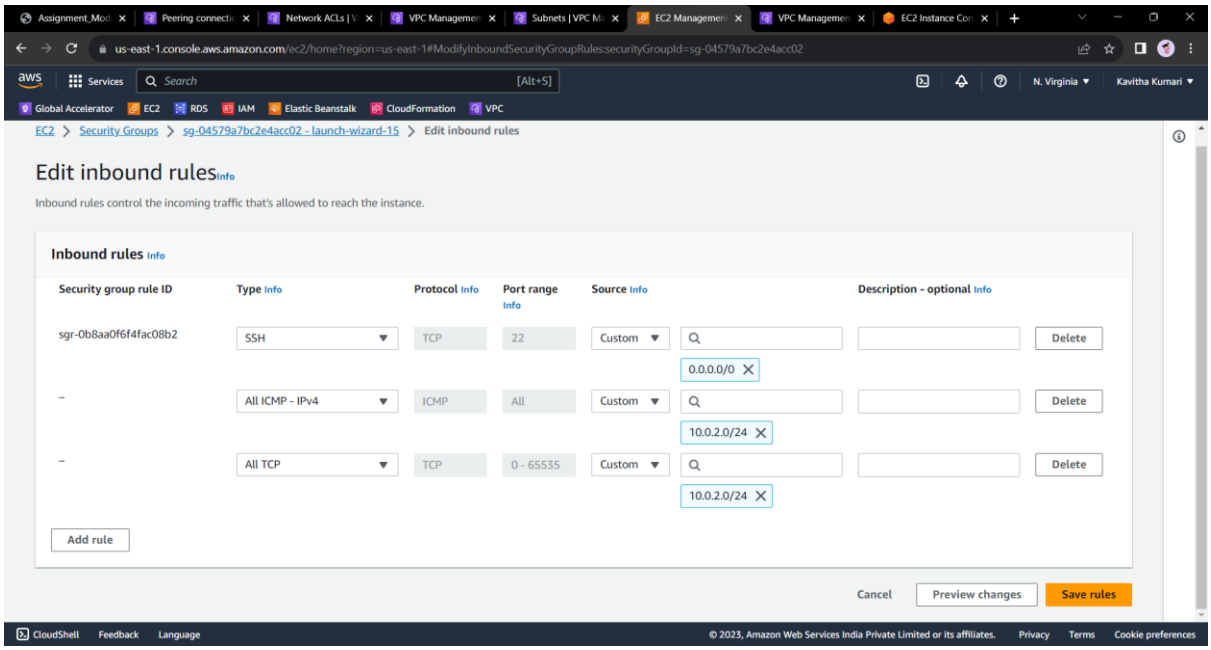
- **Setup connection between db subnets of both production network and development network respectively.**
- In the below figure we are adding the subnet ip address of the DB instance of the DevelopmentNetwork to the DB instance route table of the ProductionNetwork and target is set as peering connection which we have made now.

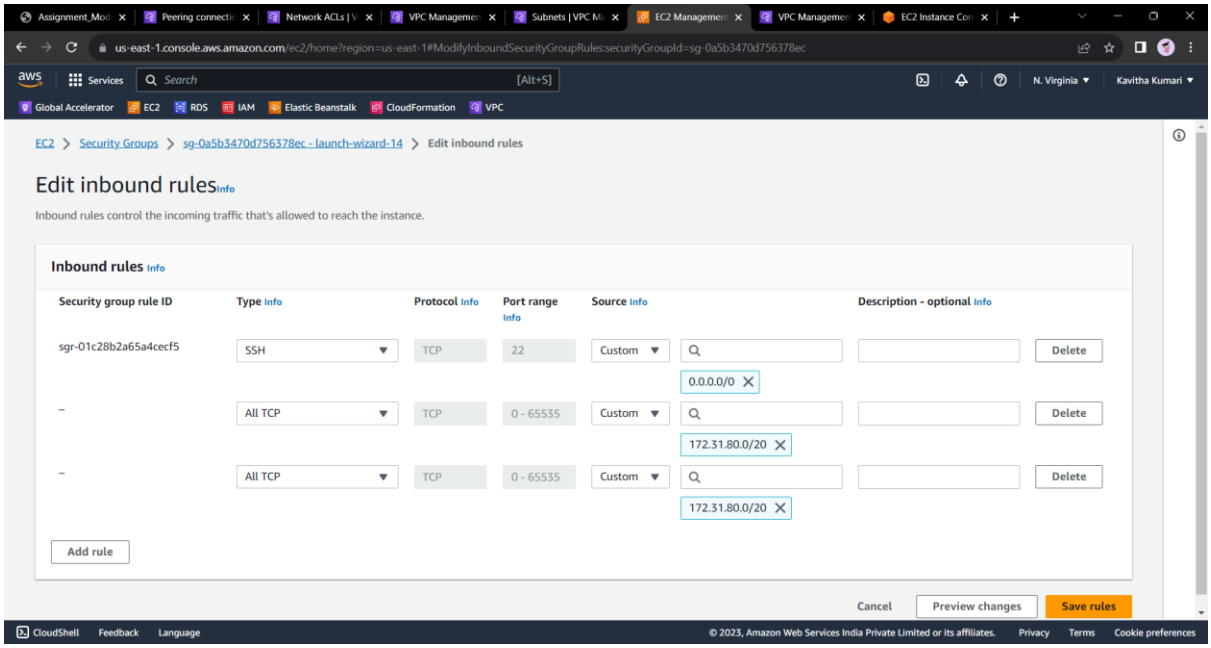

us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#EditRoutes:RouteTableId=rtb-0c80009bc1c93c340
VPC > Route tables > rtb-0c80009bc1c93c340 > Edit routes
Edit routes
Destination
Target
Status
Propagated
172.31.0.0/16
local
Active
No
10.0.2.0/24
pcx-02d6637b64682d6e5
No
Remove
Add route
Cancel
Preview
Save changes

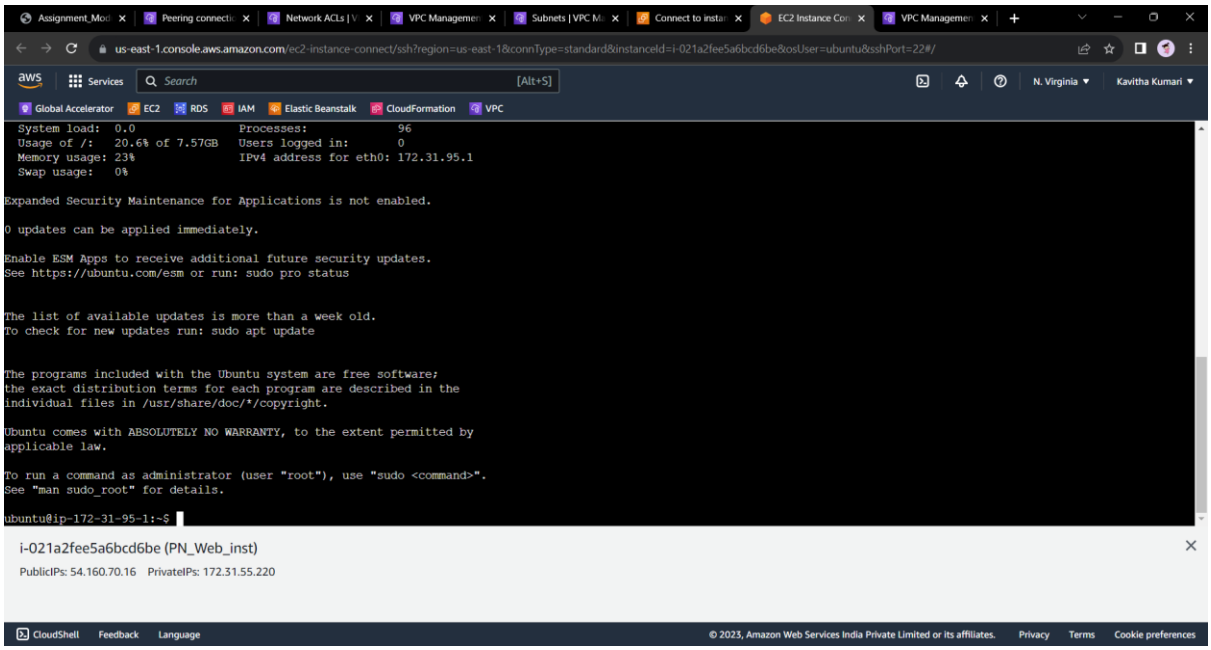

us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-0abd8ca4fc2347bbf
VPC dashboard
EC2 Global View
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
VPC > Route tables > rtb-0abd8ca4fc2347bbf
rtb-0abd8ca4fc2347bbf / DN_PivateRoute
Actions
You can now check network connectivity with Reachability Analyzer
Run Reachability Analyzer
Details
Route table ID
rtb-0abd8ca4fc2347bbf
Main
No
Explicit subnet associations
subnet-0dd2a83fdfa90fe44 / DN_DB
Edge associations
VPC
vpc-0bff4e15151dae866 | Development Network
Owner ID
049198078472
Routes
Subnet associations
Edge associations
Route propagation
Tags
Routes (1)
Edit routes
Filter routes
Both
Destination
Target
Status
Propagated



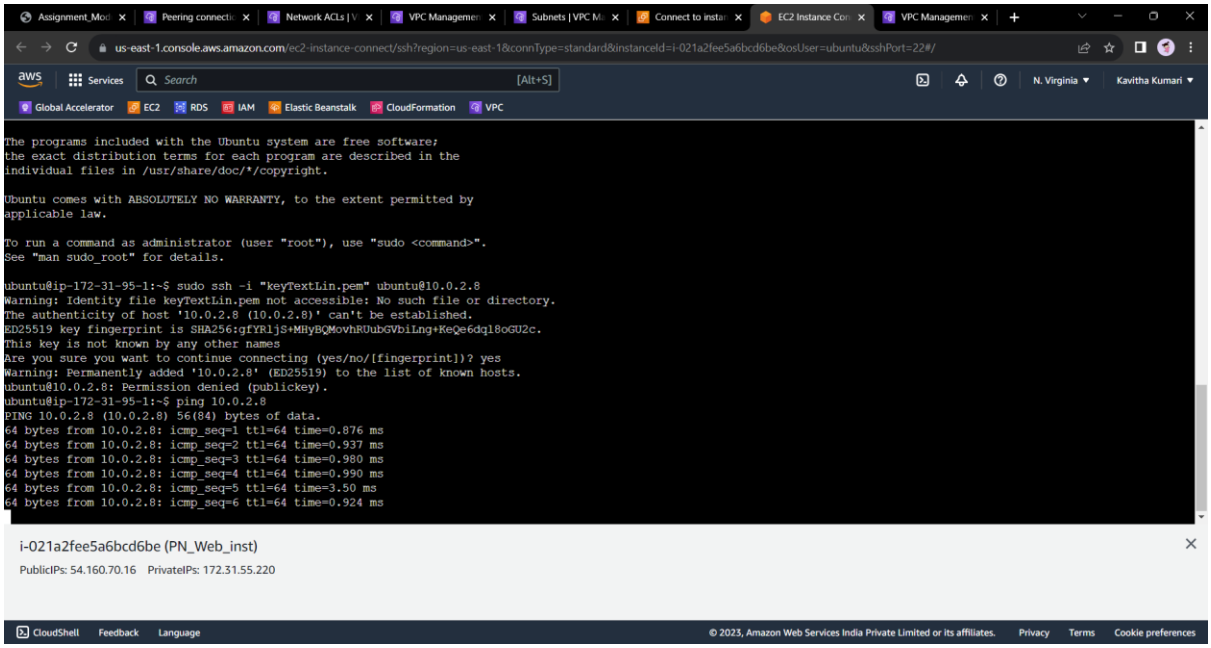
- Now edited the Security Group's inbound rule for both of these instances as per the following screenshot:







Now I am in my PN_DB instance. Let me try to ssh into my DN_DB



From here I can ping into my DN_DB instance.

------End----------