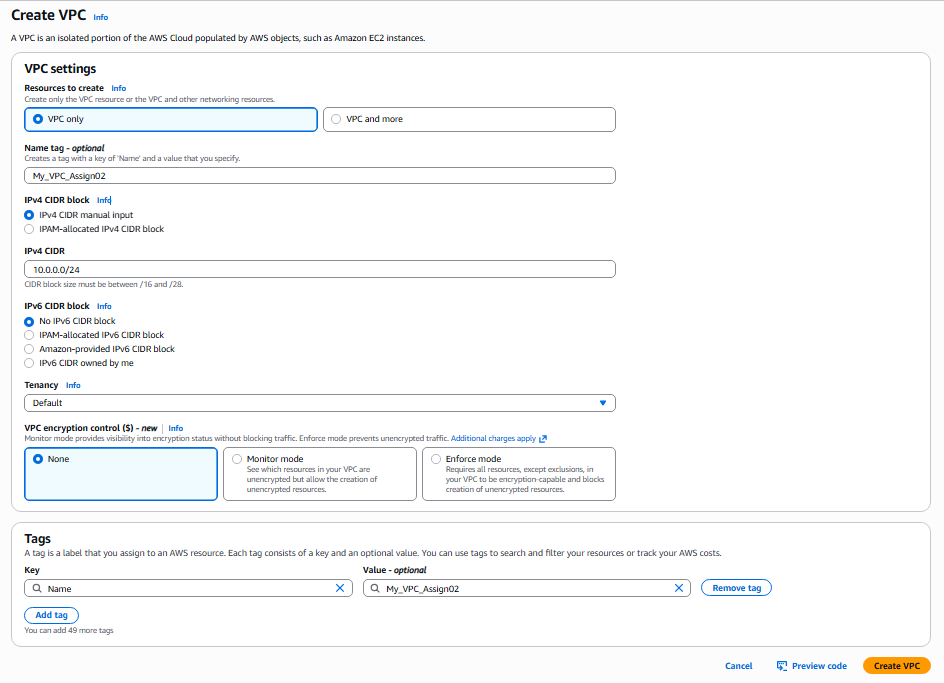
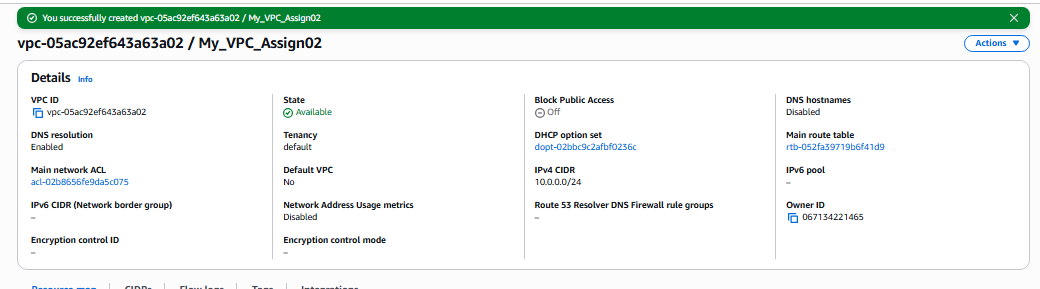
VPC-Assignment-02

1. Create one VPC, with 1 public subnet and 1 private subnet.

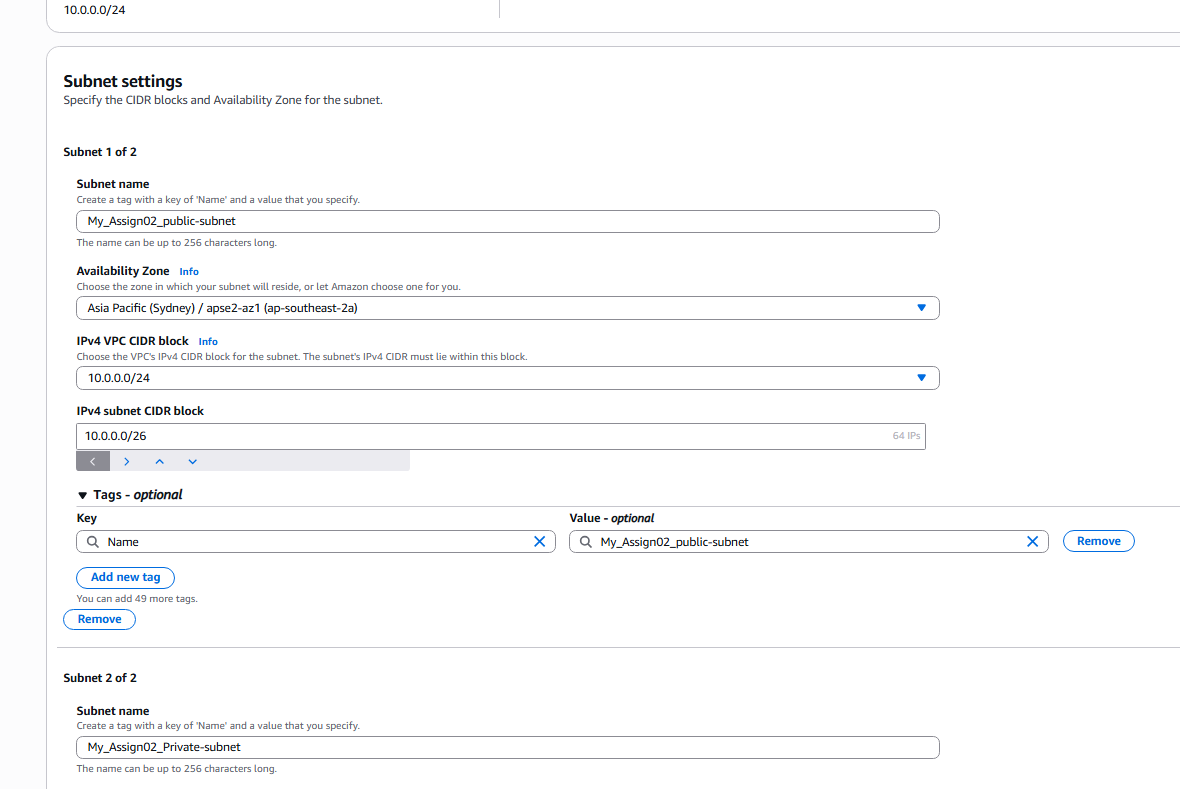
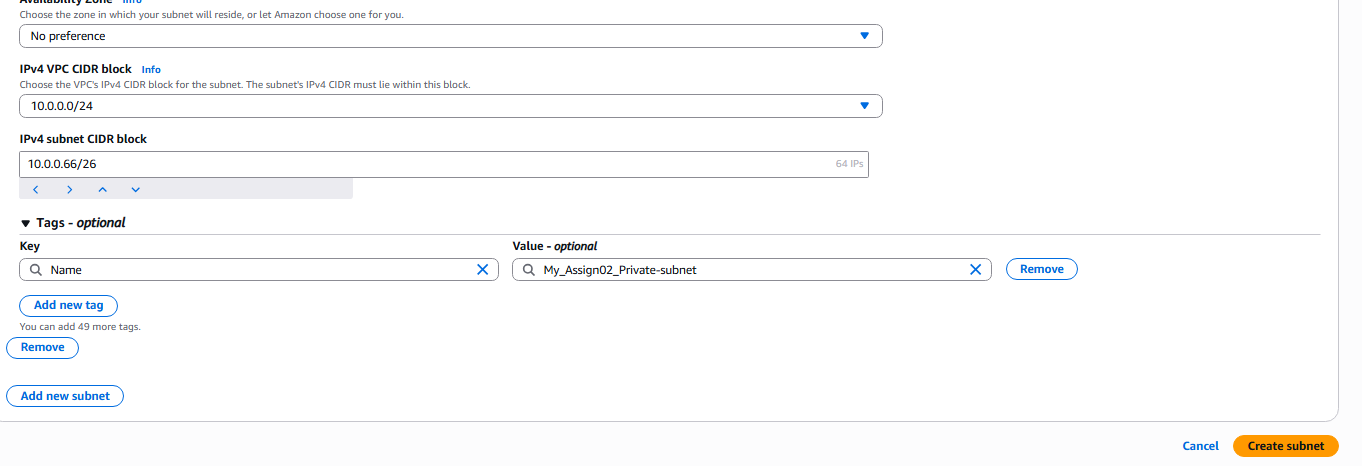
Go to VPC service on the aws console, click VPC and click ‘create VPC’, Select VPC only option in the resources to create, define the name an the IPV4 CIDT rande, leave the rest of the options to default and create VPC



Below is the image showing the the create VPC (My\_VPC \_Assign02)



Now Create the subnets inside the VPC’s by clicking the subnets option and click create subnet option, Assign the name of the subnet(My\_Assign02\_Public\_subnet), Assign the CIDR block 10.0.0.0/26 , clicked add subnet at the bottom to give more subnets give subnet name (My\_Assign02\_Private\_subnet), Assign the CIDR block 10.0.0.66/26 and click create subnet

a 

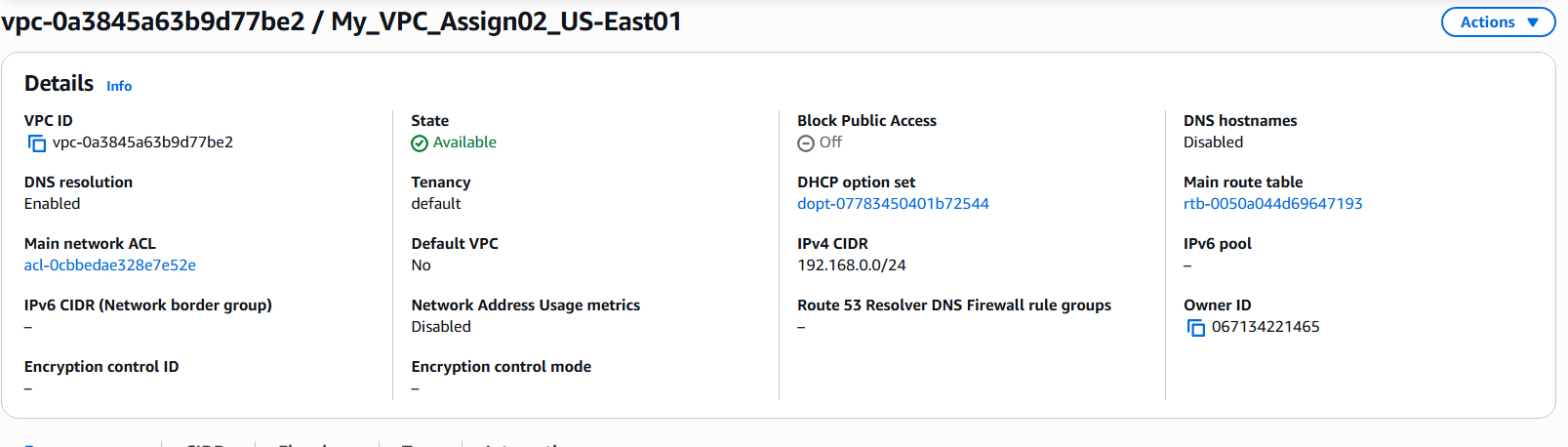
The subnets created are as follows.



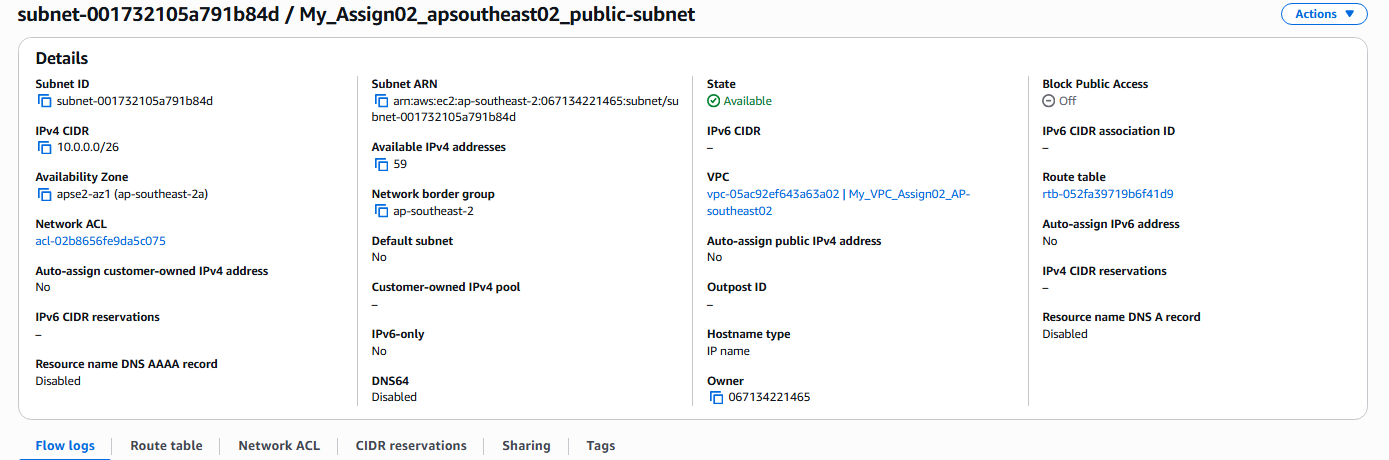
1. Enable VPC peering for cross-region.

Now create two VPC with the name (My\_VPC\_Assign02\_AP-southeast02, My\_VPC\_Assign02\_US-east01) since we are creating the two regions I have chosen the region names as the suffix for the identification.

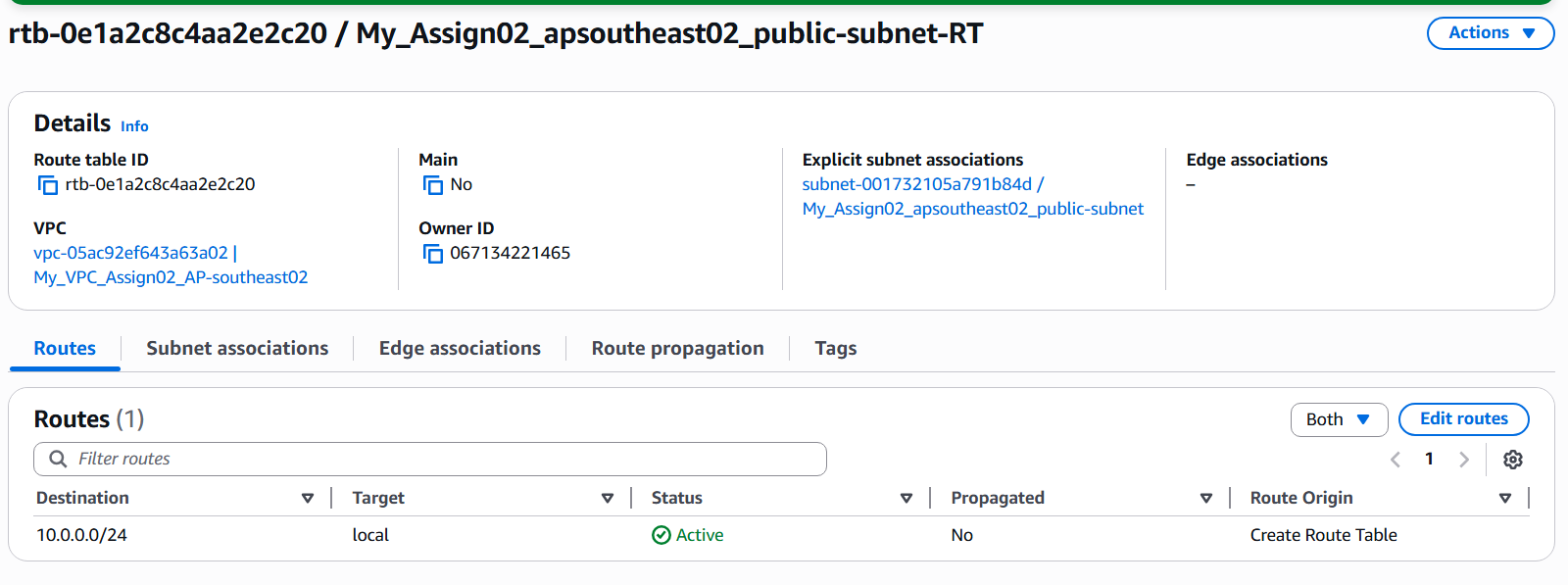




Now create the subnets public and private subnets in both the VPC’s and I have given the name (My\_VPC\_Assign02\_AP-southeast02\_public-subnet, My\_VPC\_Assign02\_AP-southeast02-private-subnet), (My\_VPC\_Assign02\_US-East01-public-subnet, My\_VPC\_Assign02\_US-East01-private-subnet)

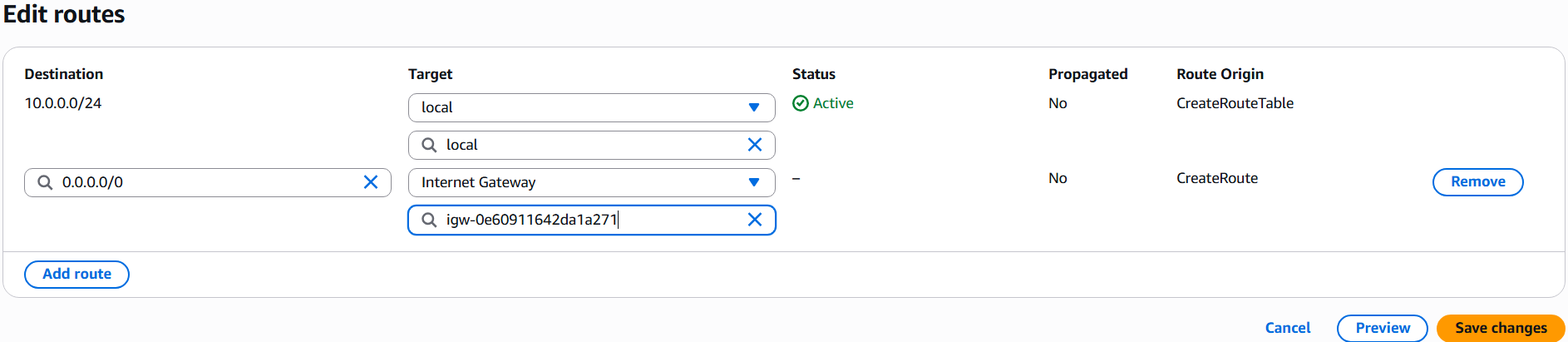
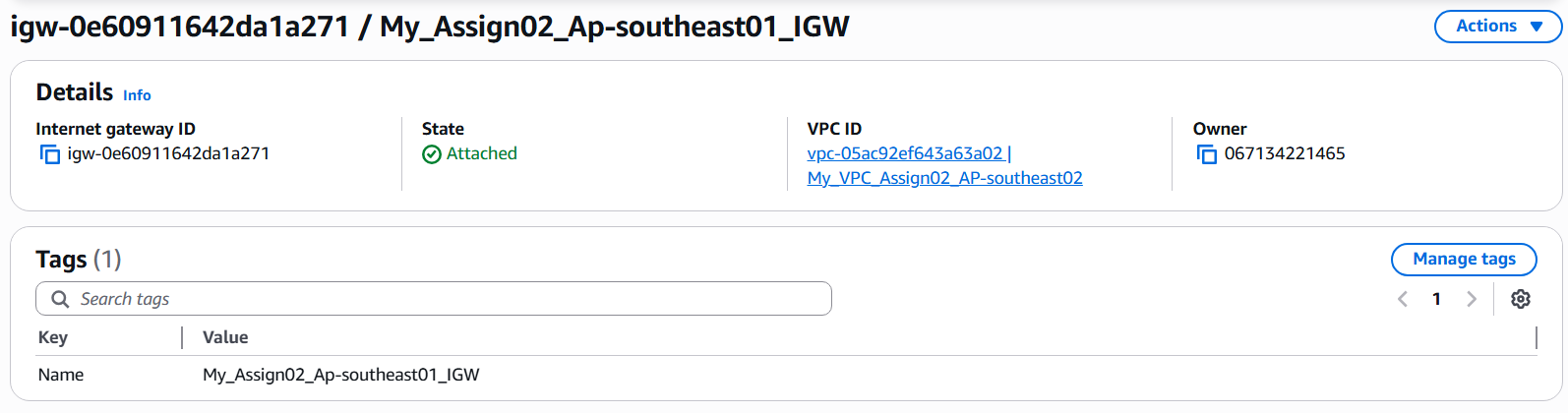


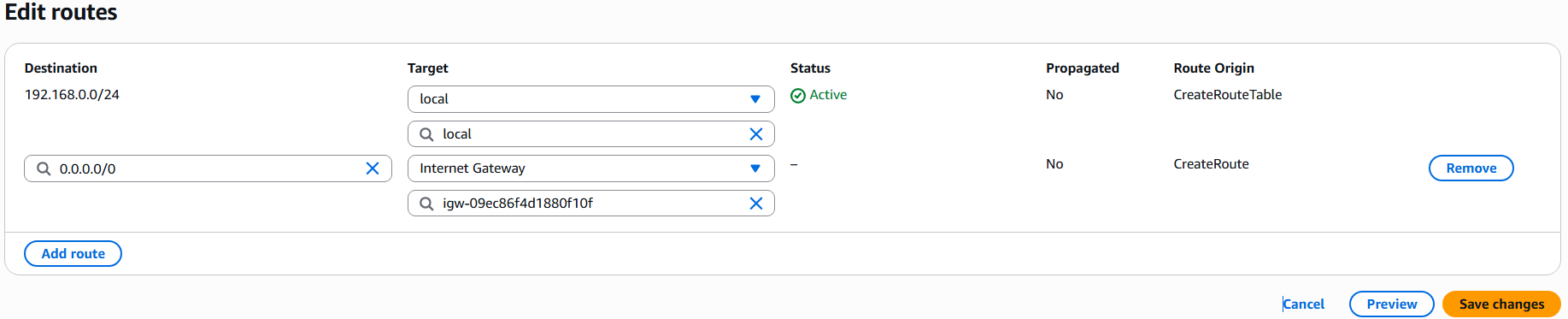
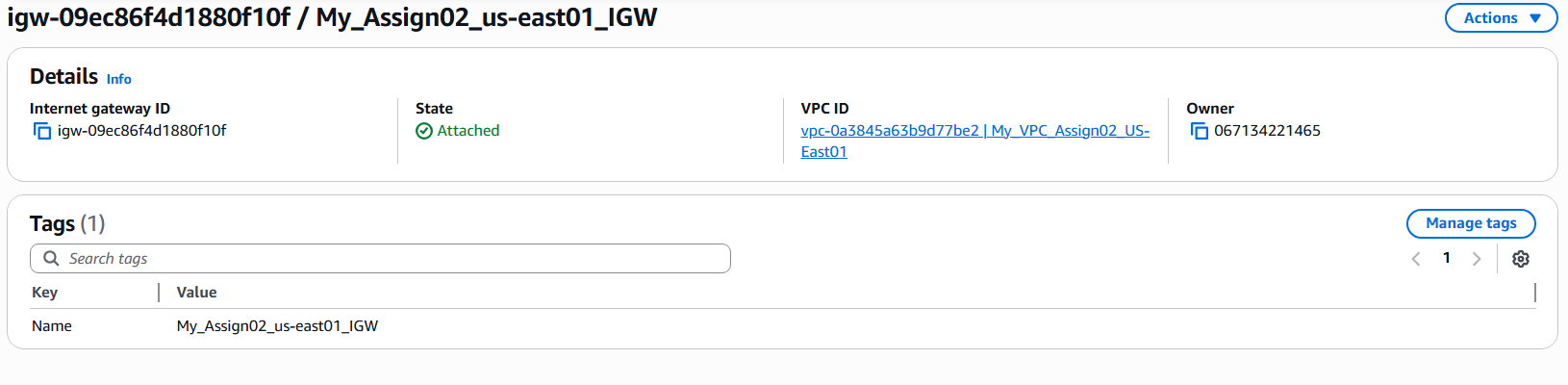


Create the Route table and associate the route table to the created private and public subnet

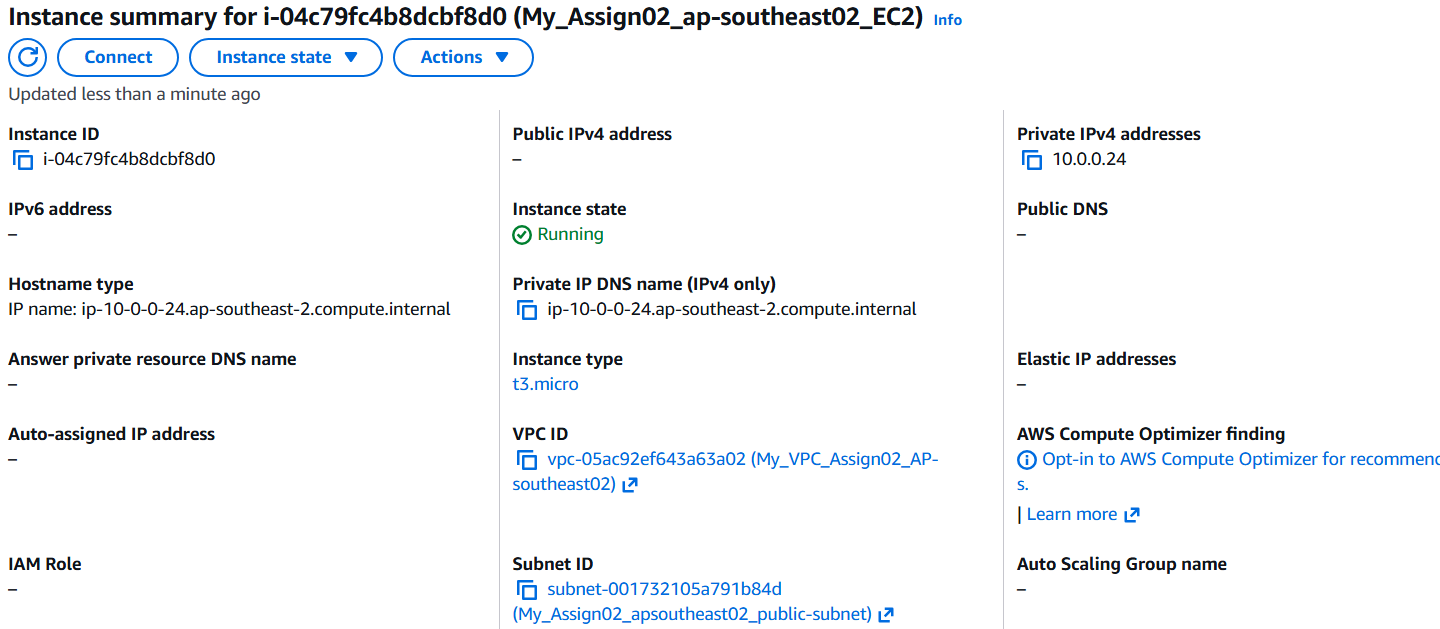


Create Internet gateway for the public subnet and attach the IGW to the public subnet in the route table

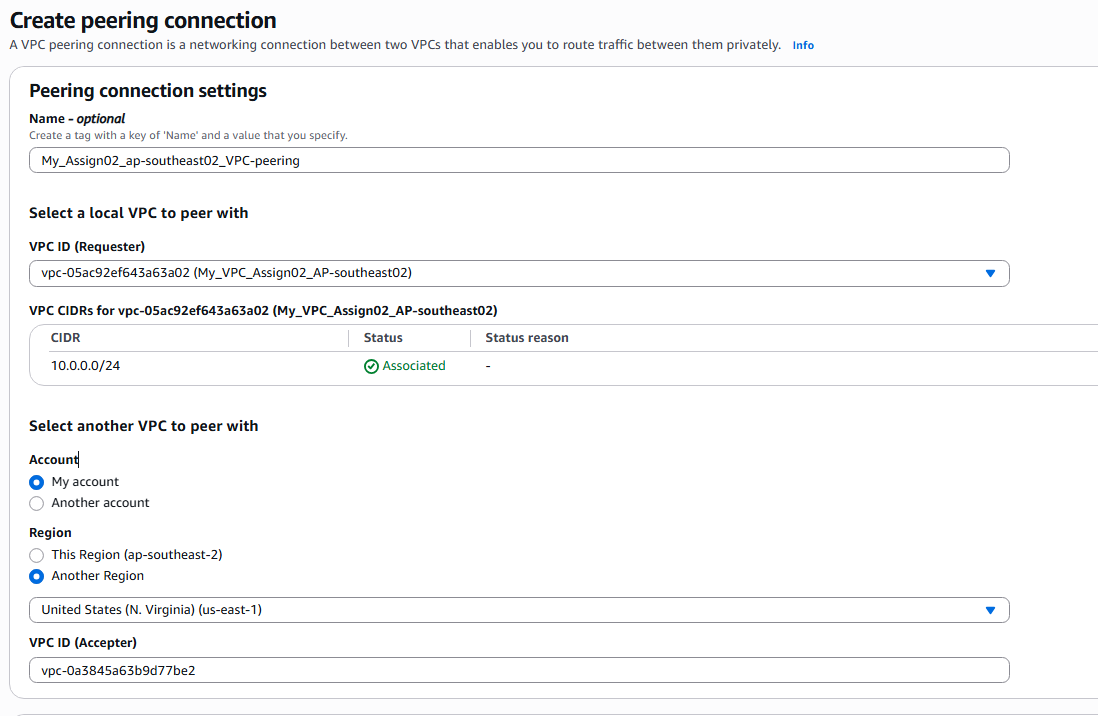




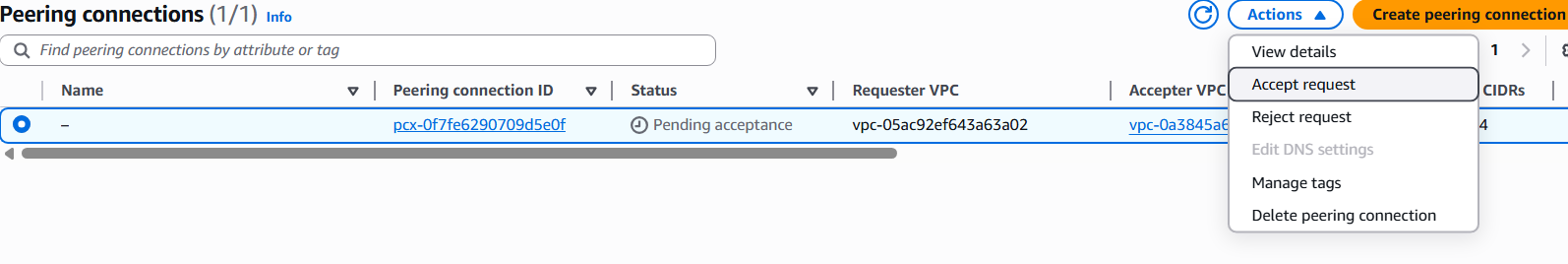
Provision a instance in the public and private subnets in the both regions and lauch the instances.

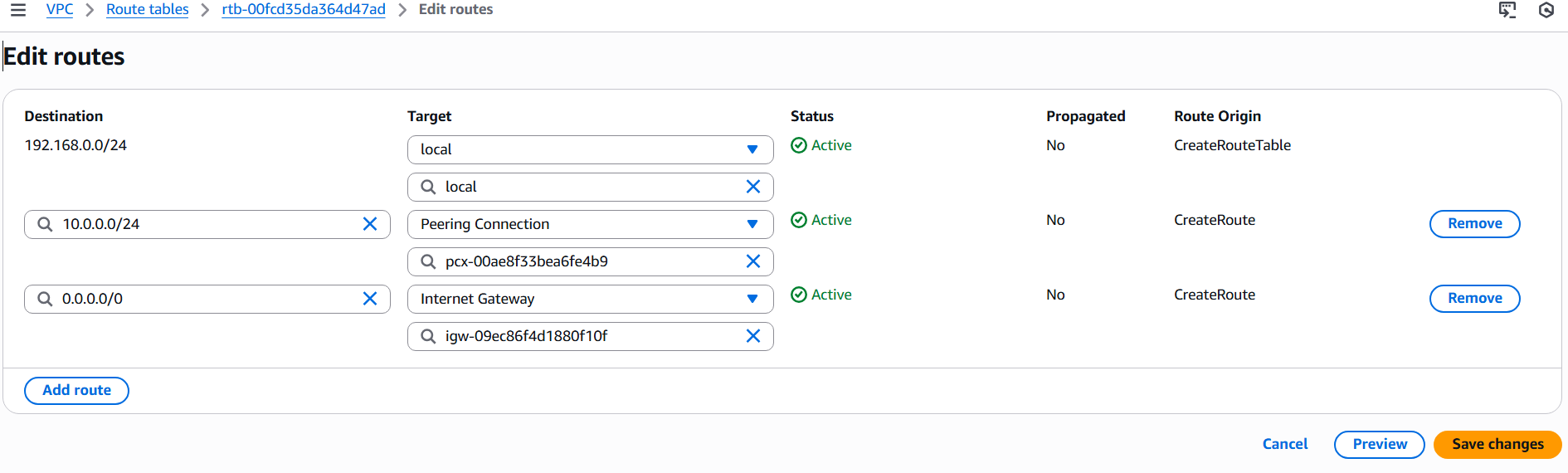
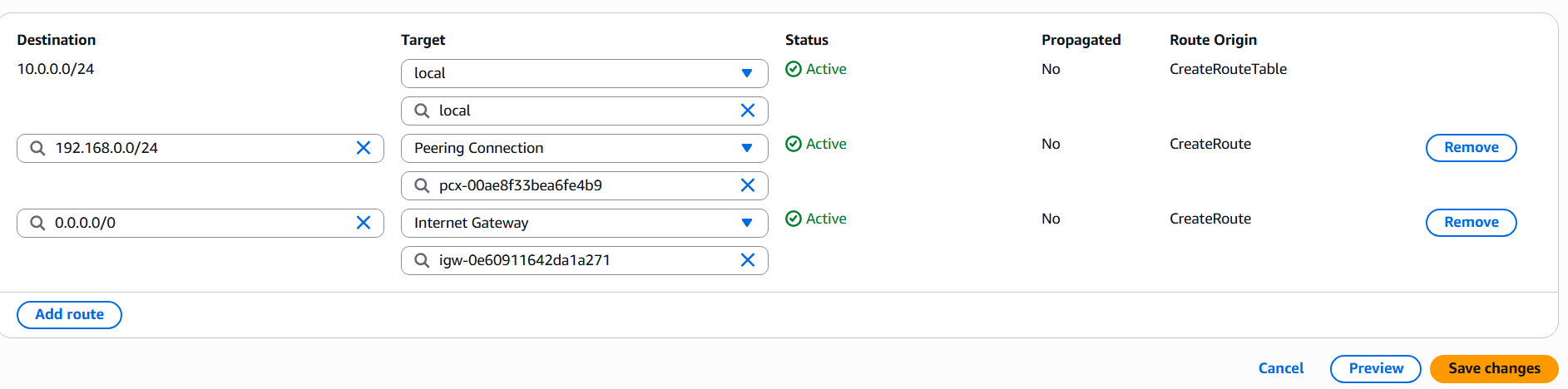


Now raise the peering request to the VPC in the another region by creating the peering request by giving the name of the connection, click the peering with ‘My account’ , specifiy the ‘region’ of the VPC you want to peer with and also specify the peering ID of the another region’s VPC.

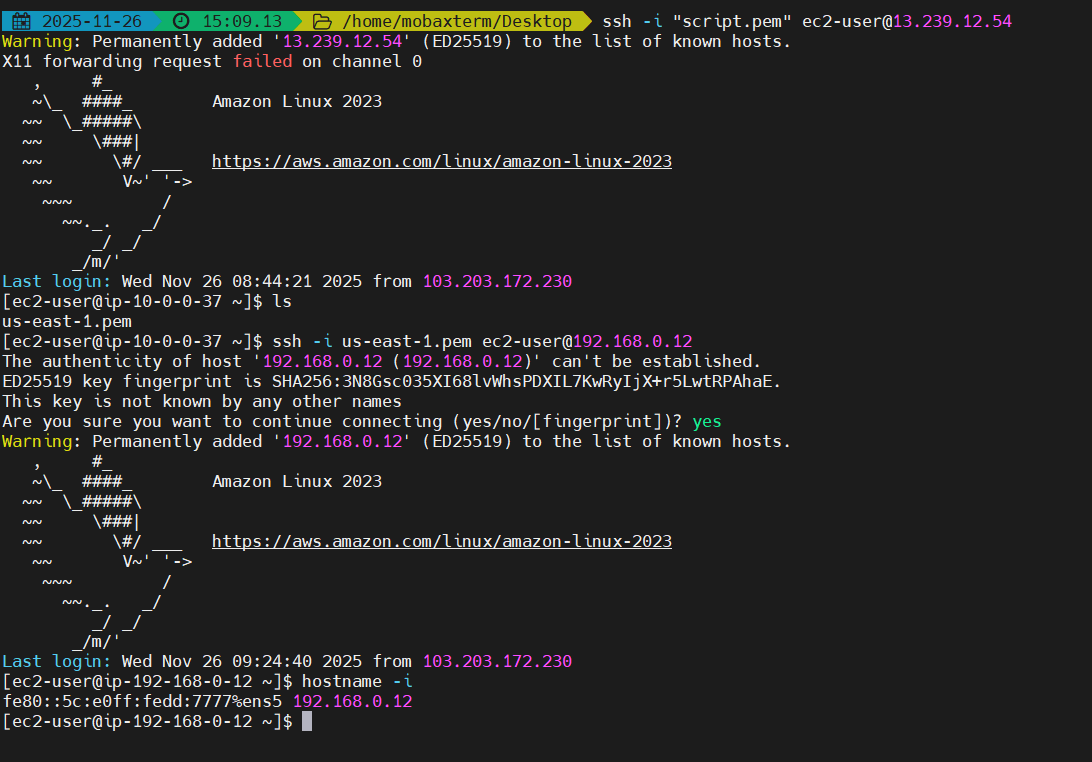


Once the peering request is made, Go to the VPC Peering option and you can see the peering request pending on the console page, select and click the action button to accept the request



Once the peering request is successful Update routing tables by selecting the target option to peering connection

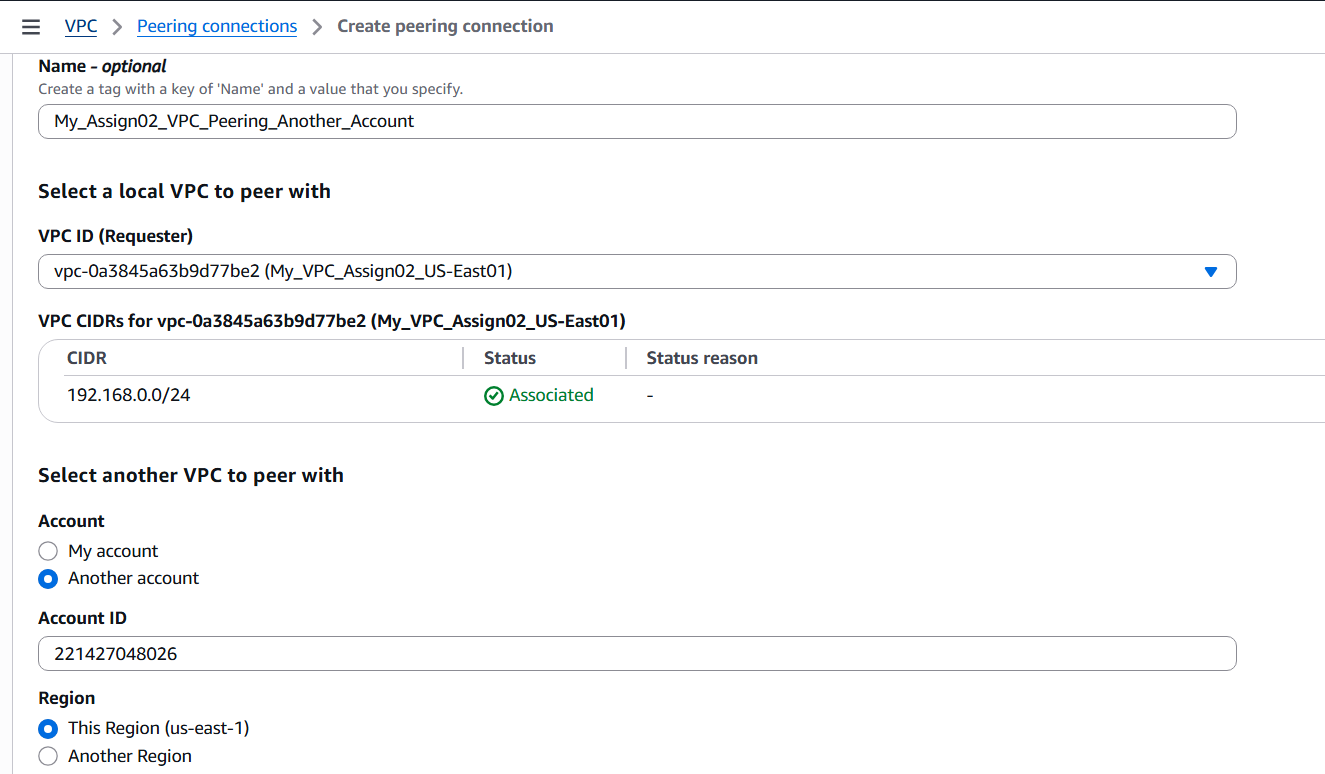
Logon to the instance provisioned in the public subnet earlier and create the pem file by copying the contents of the pem file in the local machine and give the proper permissions to the pem file to initiate a ssh connection to the instance in the private subnet, here we are using this instance as the jump server to the private subnet instance. As we can see that the connection is established between the public instance in Sydney region to the private instance in the us-east-1 region.



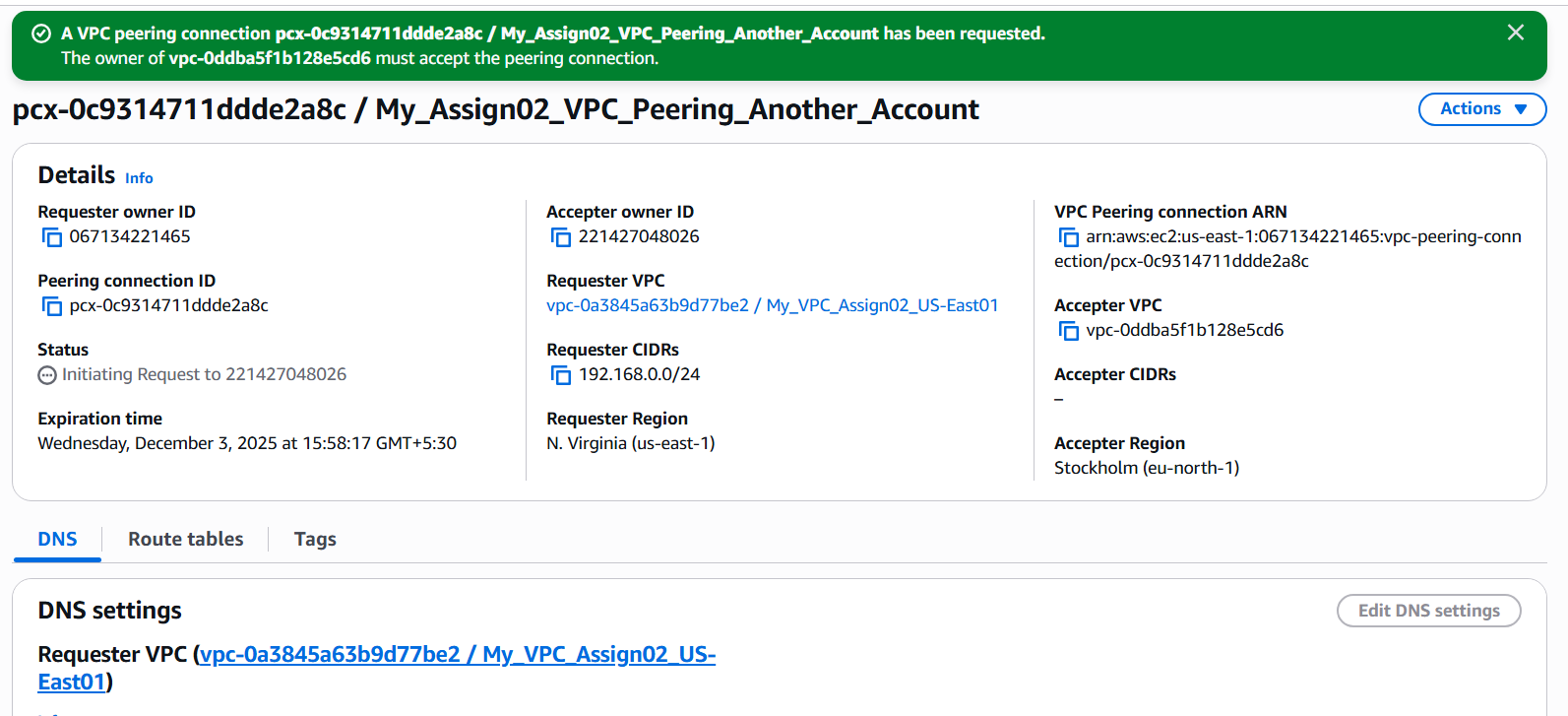
1. Enable VPC peering for cross-account (you can collaborate with your friend to do this task).

I have kept the VPC and instance’s create in the Us-East-1 region intact and made the VPC connection peering with the another account using the create peering connection option in the VPC peering.

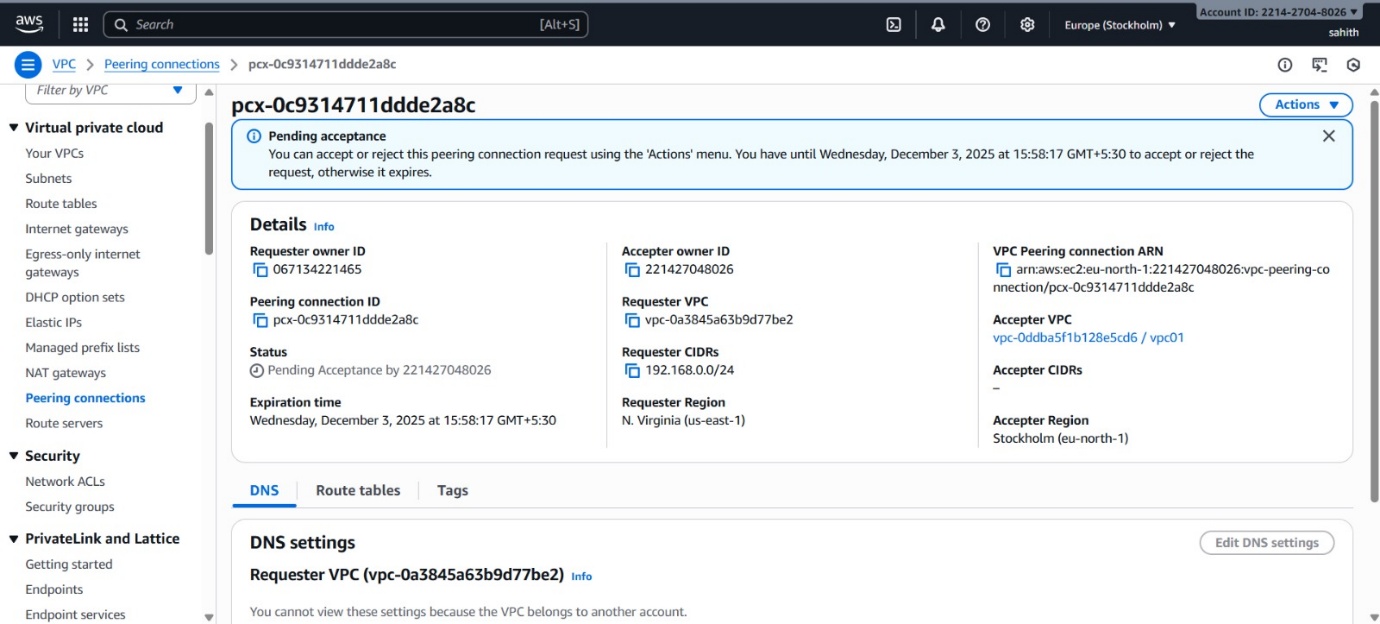
Entered the name of the connection as (My\_Assign02-VPC\_Peering\_Another\_Account), Choose the vpc and select Another account in the peer with option and provide the account ID, select the region of the another account VPC and create the request



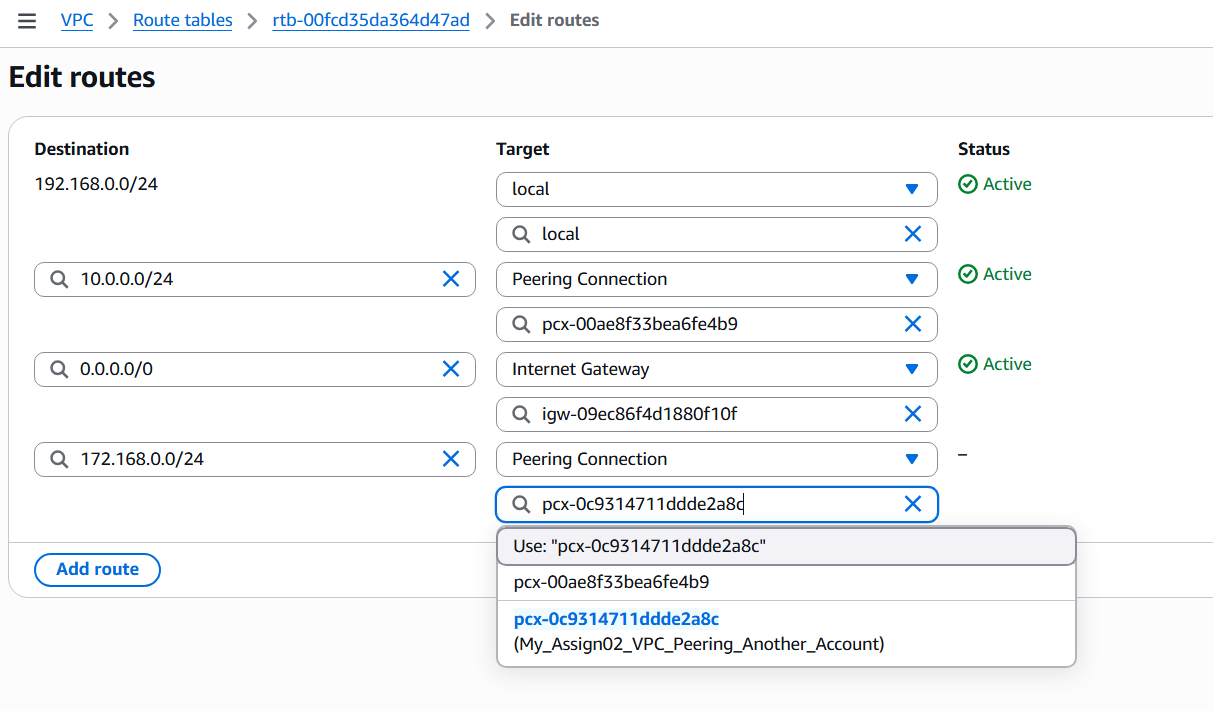
The request made is as follows.



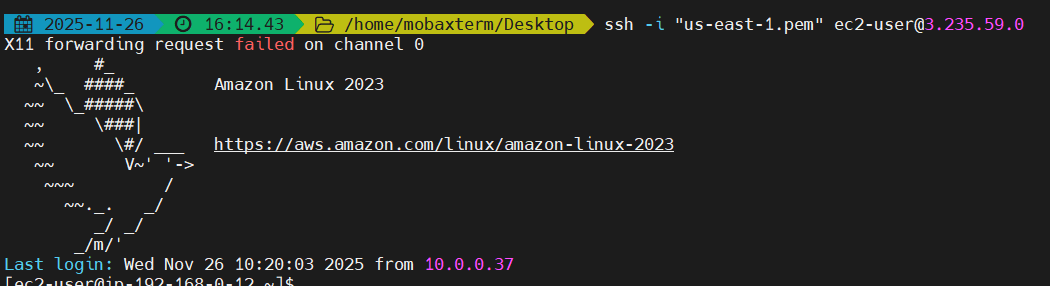
This screenshot is of the other account and pending acceptance is shown, go to actions and accept the request



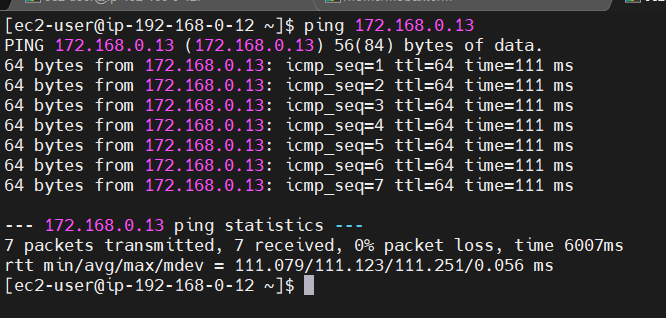
After the request is accepted, update the route tables in both accounts by selecting the peering connection option



Now logon to the public subnet EC2 instance and make the ping request to check if the connection is established with the EC2 instance in the private subnet of the another account.



The private instance IP\_Address is 172.168.0.13 and we can see the Ping request is successful

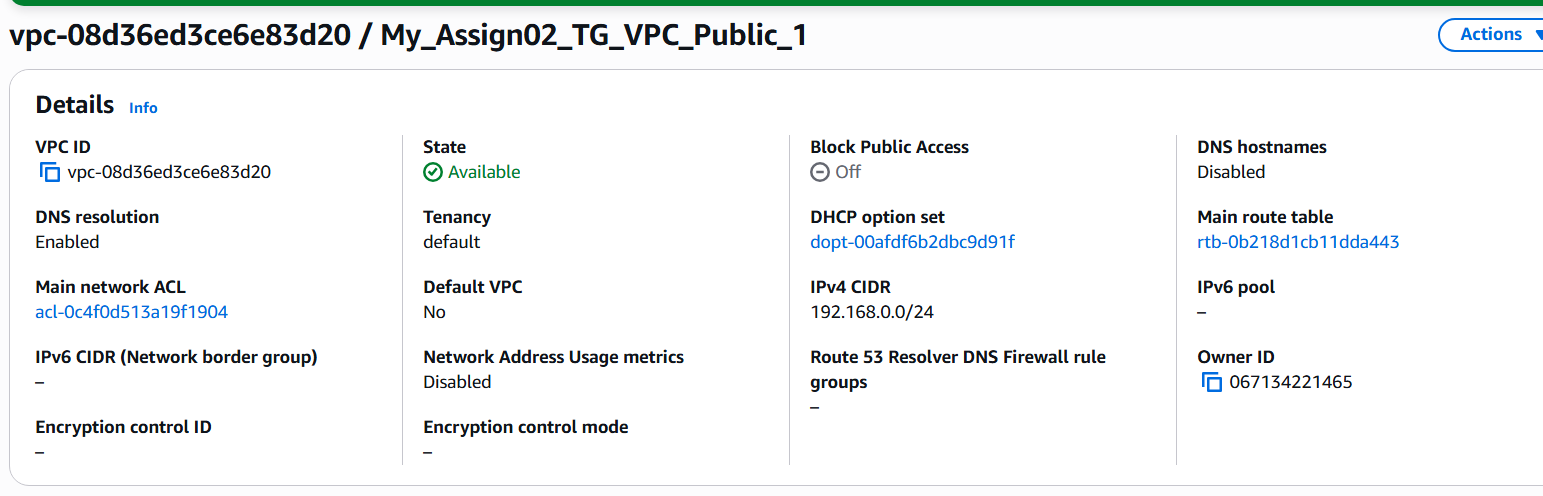


1. Set up a VPC Transit Gateway.

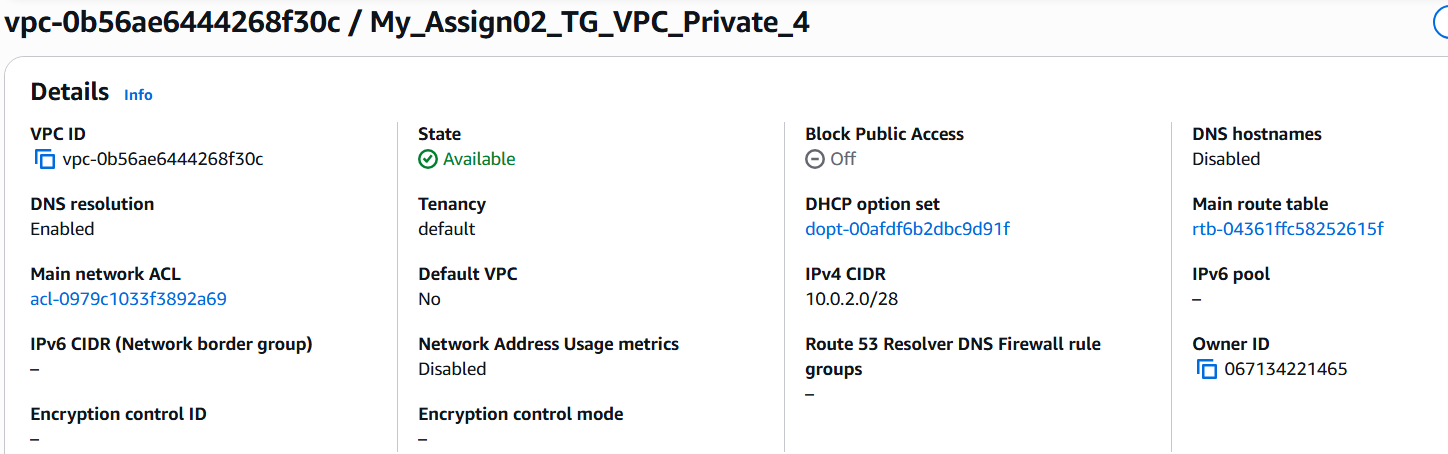
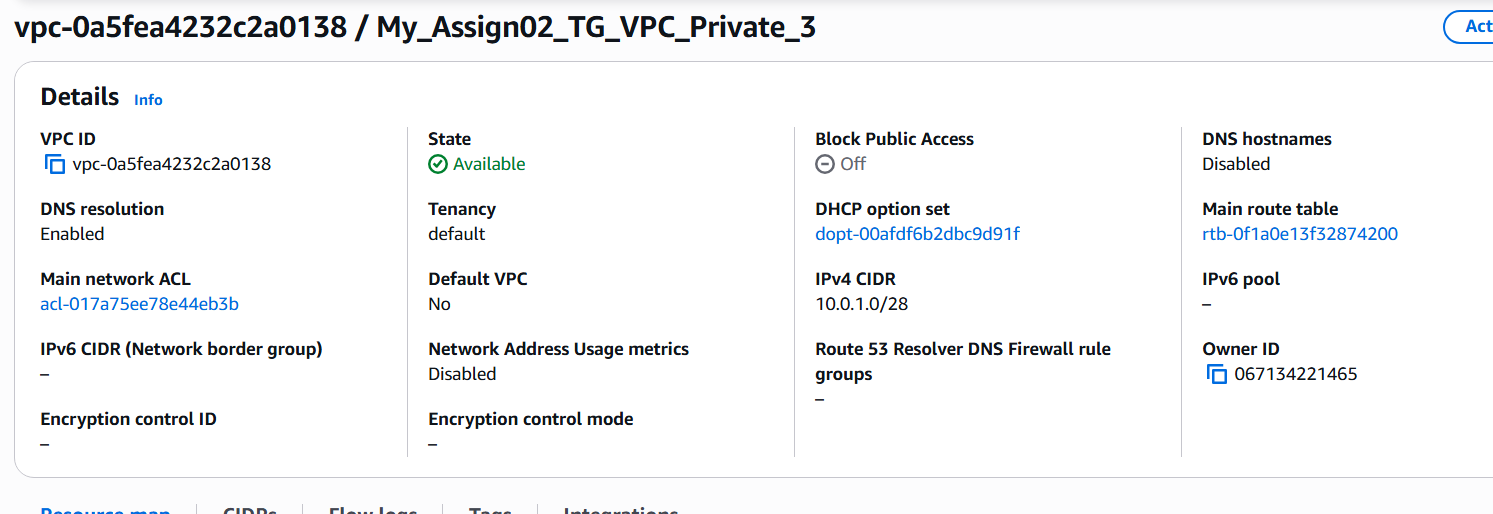
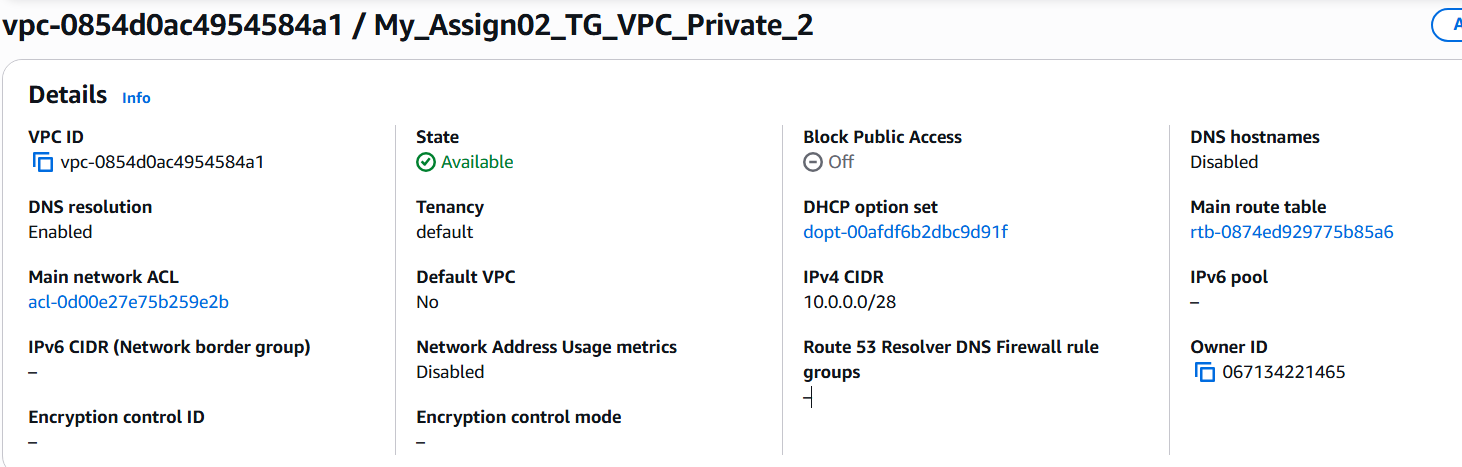
Transit Gateway is a network hub that acts as a virtual router, allowing you to connect everything through a single gateway. It helps us to connect number of VPC’s on cloud and also physical data centres.

All the VPC’s connected to Transit Gateway need to connect and update the routing tables.

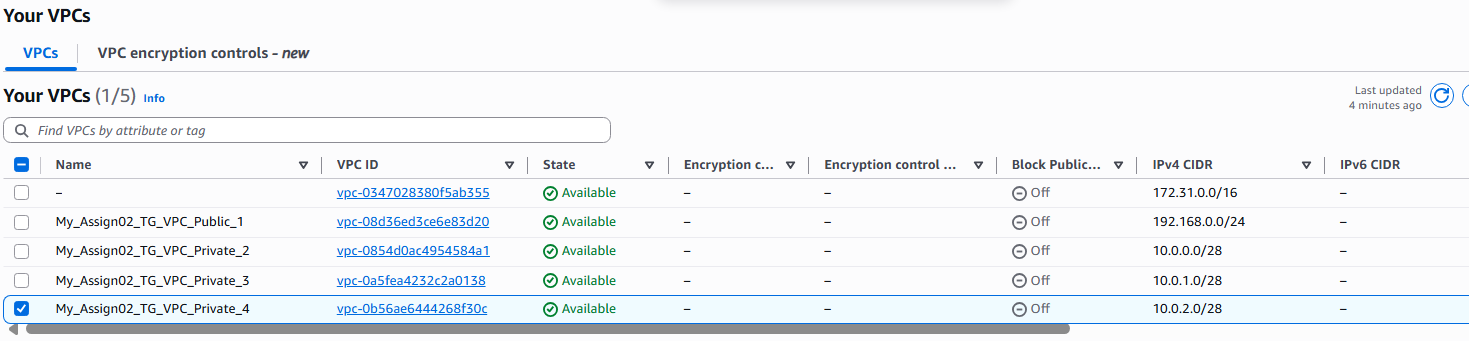
Here we are creating four VPC’s, one public VPC and Three Private VPC’s.



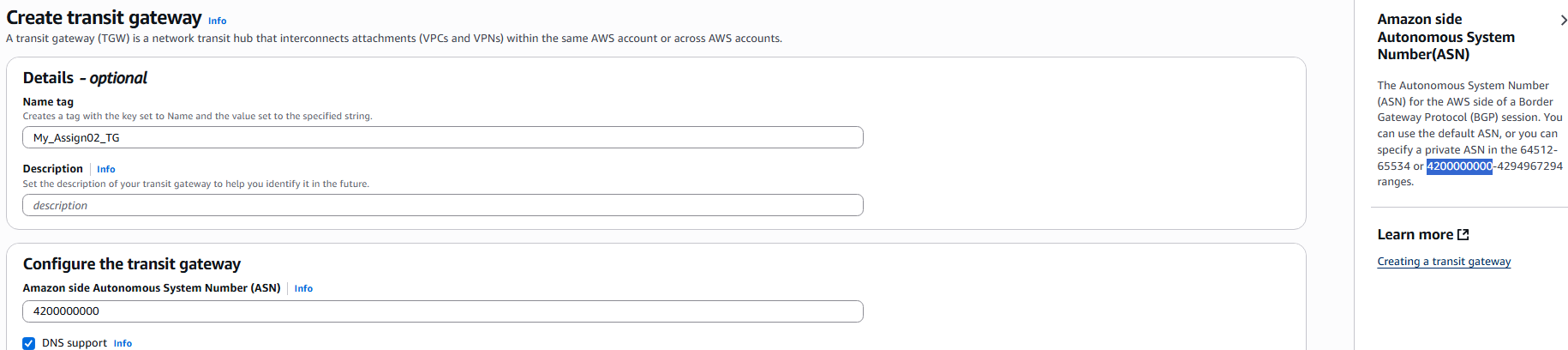
After creating the VPC1 I have associated the public route table and added the Internet Gateway to the VPC1

\

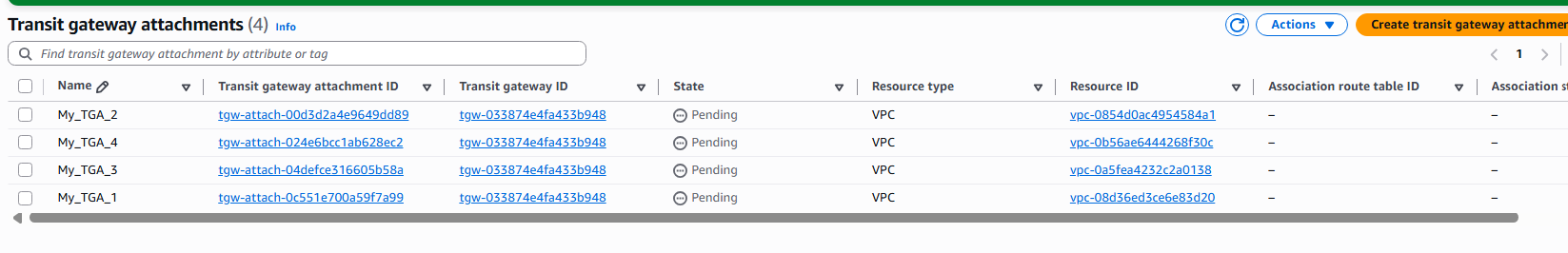
Create the private VPC2, VPC3, VPC4 with private subnets associated



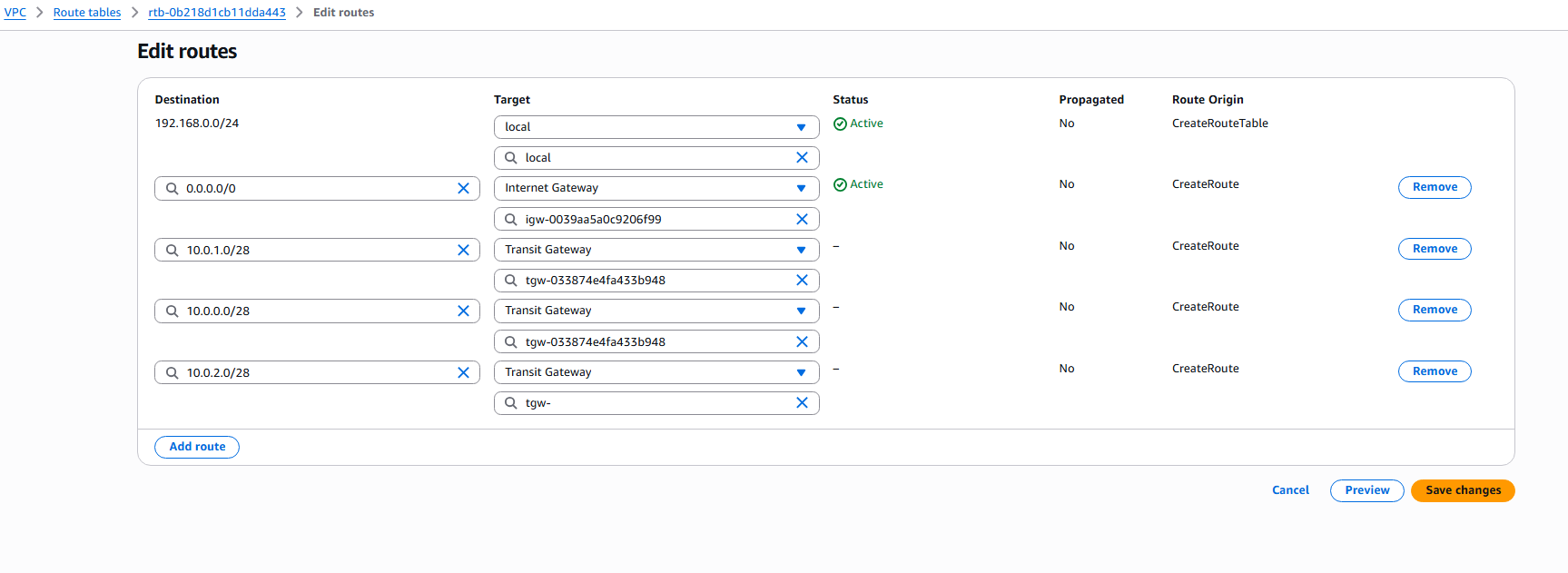
Now I have created a transit gateway by giving the name My\_Assign02\_TG and give the ASN number to 4200000000



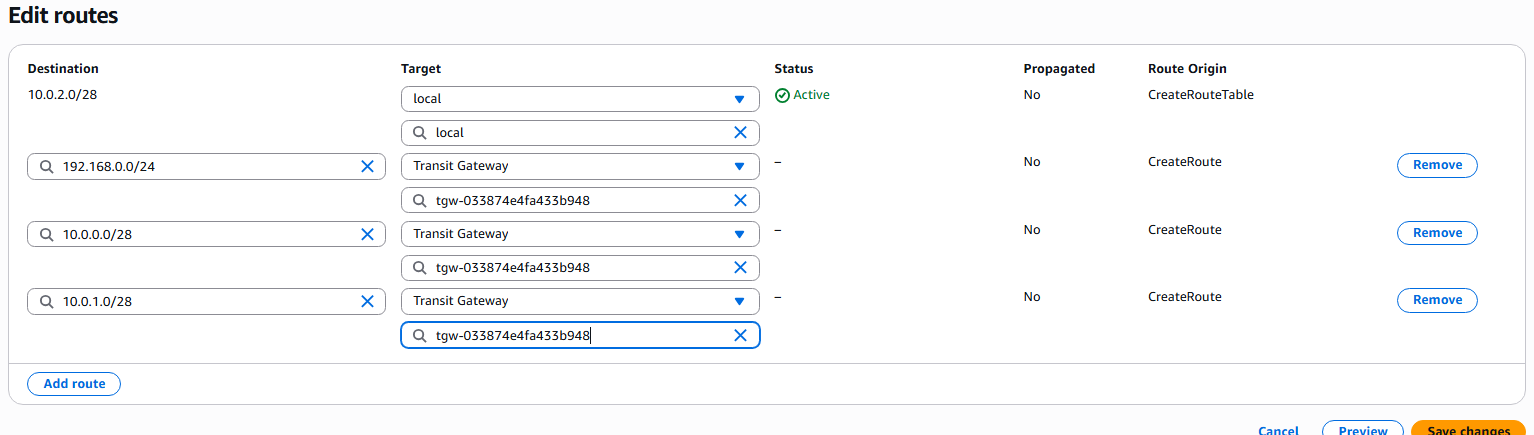
Create the transit attachment for the VPC1, VPC2, VPC3, VPC4 with name My\_TG\_1, My\_TG\_2, My\_TG\_3, My\_TG\_4



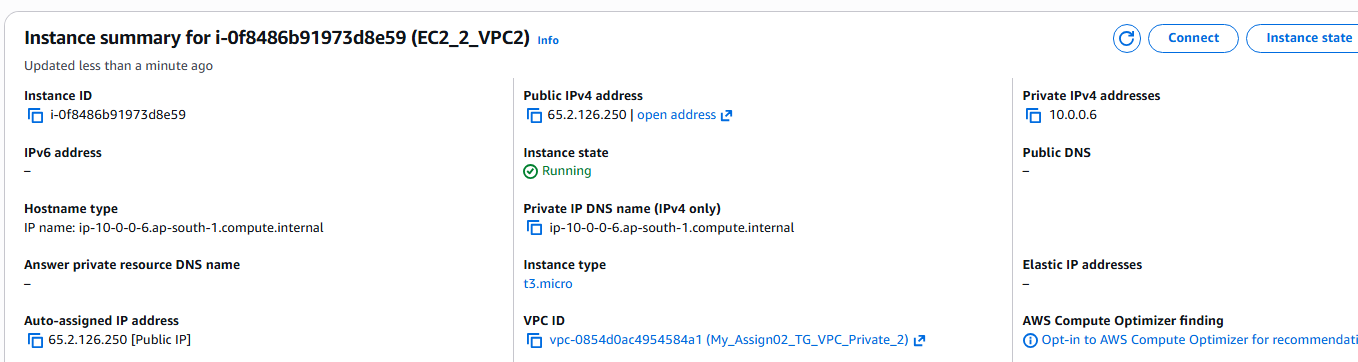
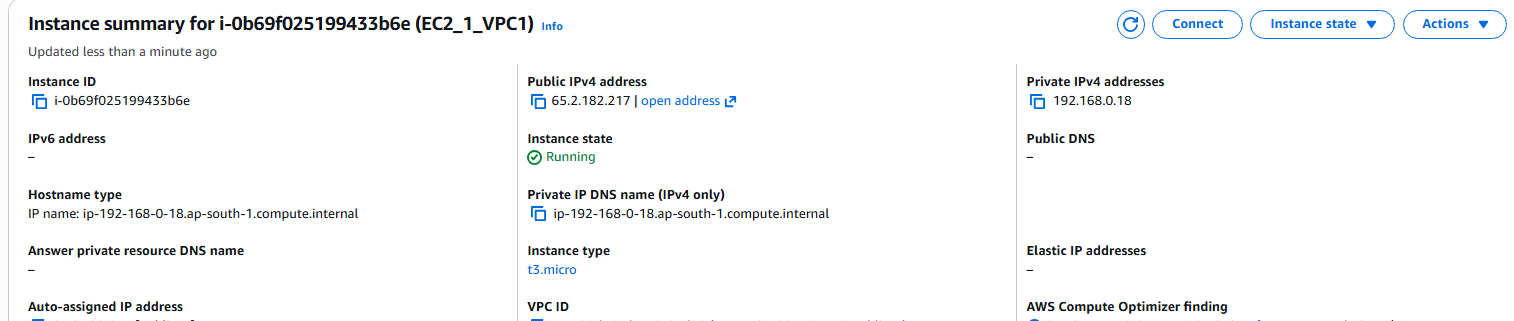
Now Update the Routing tables in the respective subnets of the VPC1 with the VPC2 Destination address with 10.0.0.0/28, VPC3 Destination address with 10.0.1.0/28, VPC4 Destination address with 10.0.2.0/28



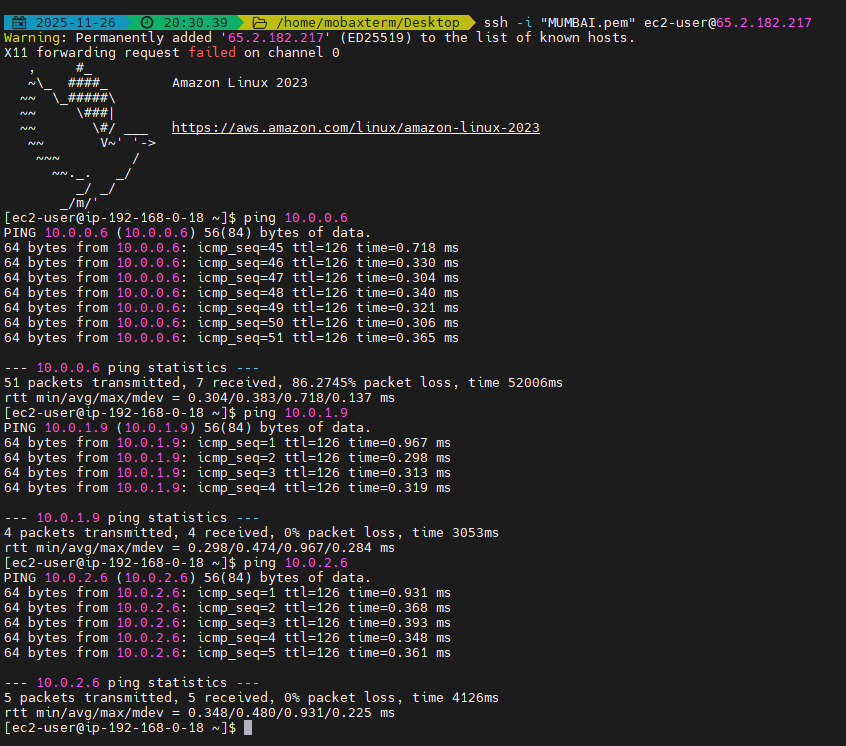
Now Update the Routing tables in the respective subnets of the VPC2 with the VPC2 Destination address with 10.0.0.0/28, VPC3 Destination address with 10.0.1.0/28, VPC4 Destination address with 10.0.2.0/28



Now launch the instances in VPC1, VPC2, VPC3, VPC4



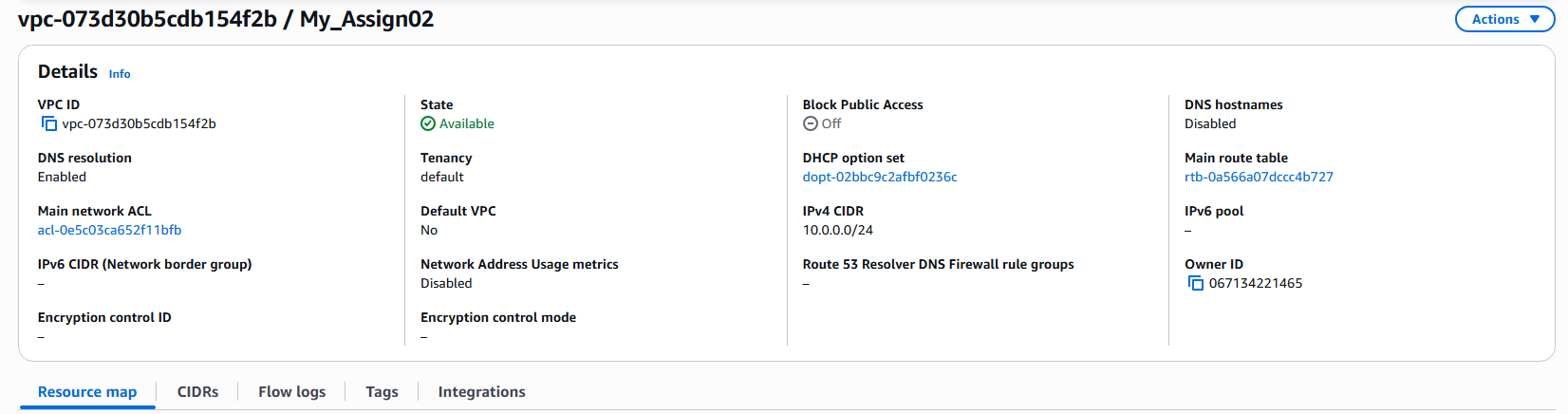
Now launch the instance in VPC1 and try pinging the private ip addresses of VPC2, VPC3, VPC4 and we can see that all the ping request’s are successful proving that transit gateway is successfully configured.



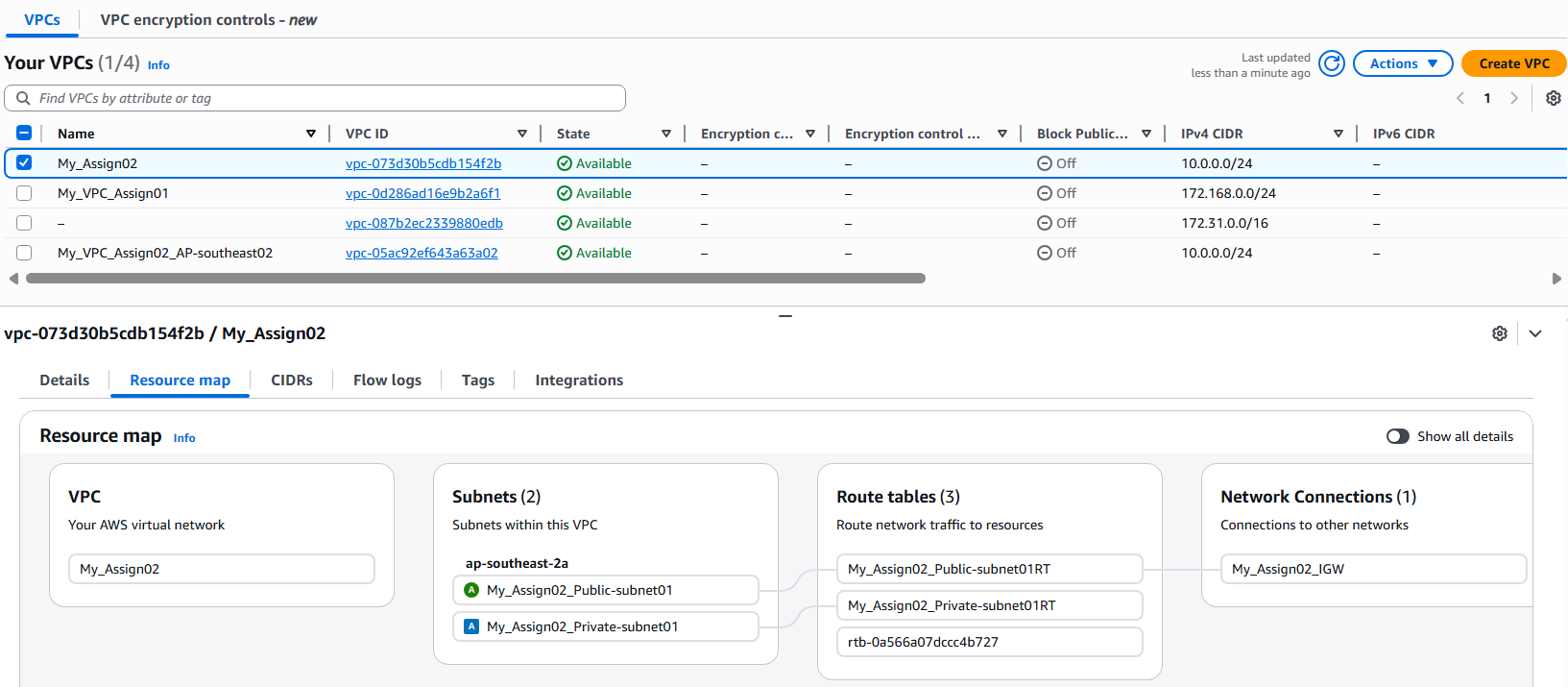
1. Set up a VPC Endpoint

VPC Endpoint service will help us to communicate with aws services without need for internet, meaning communication will be private and will be within VPC.

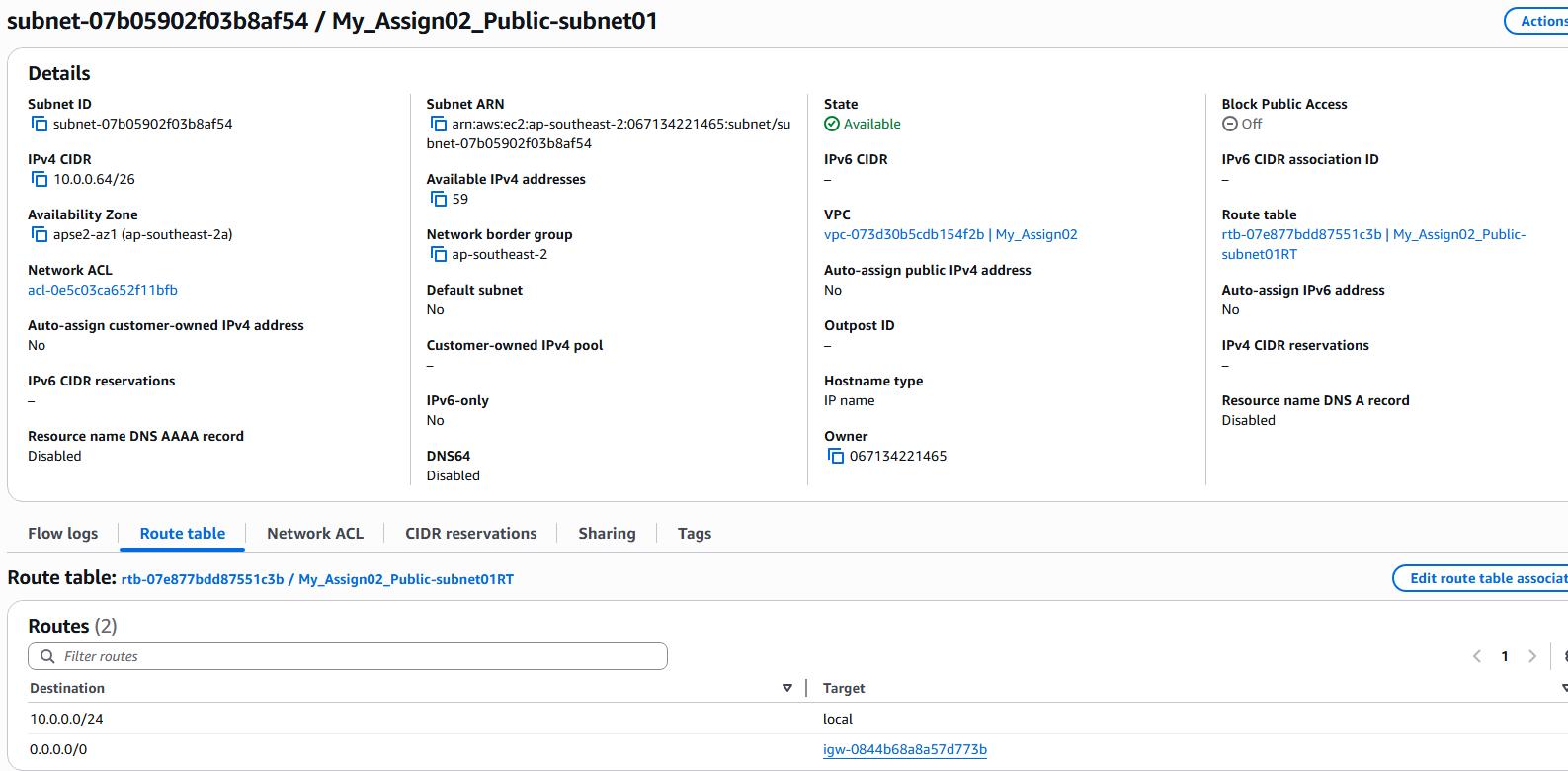
First, we will configure the vpc with one private and one public



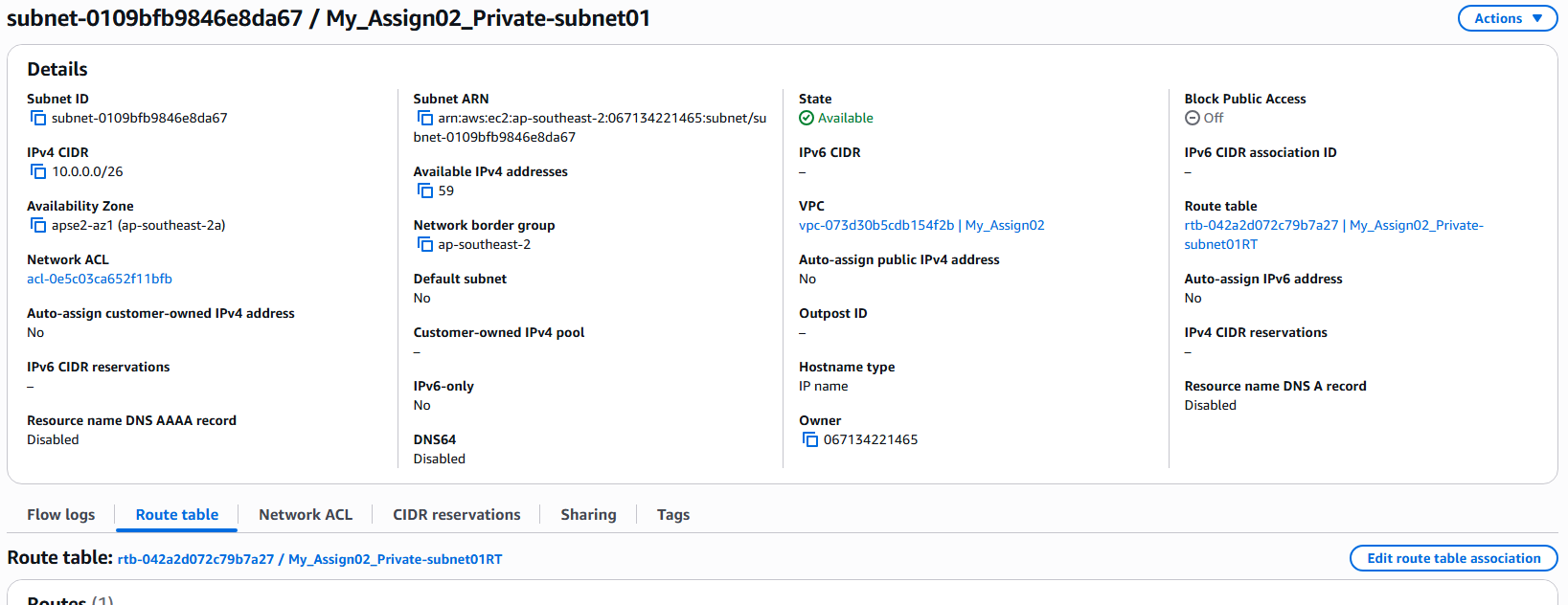
I have configured the VPC as follows as seen in the resource map with one public and one private and internet gateway connected to the public subnet



Below is the public subnet created and the see that the igw is also associated with this public subnet

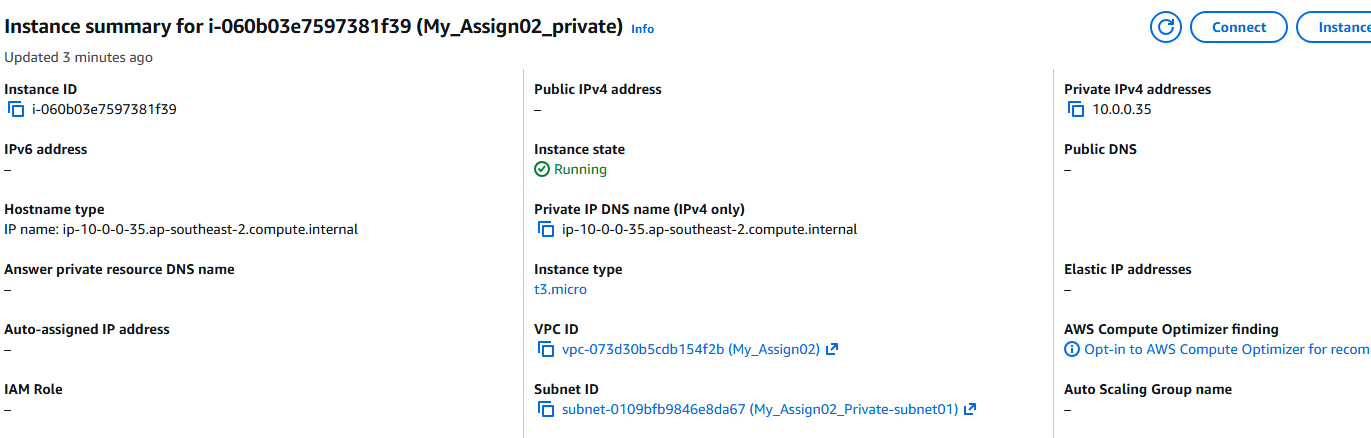


Below is the private subnet created and also the Routing table associated



I have provisioned EC2 instance in public and private subnets as follows

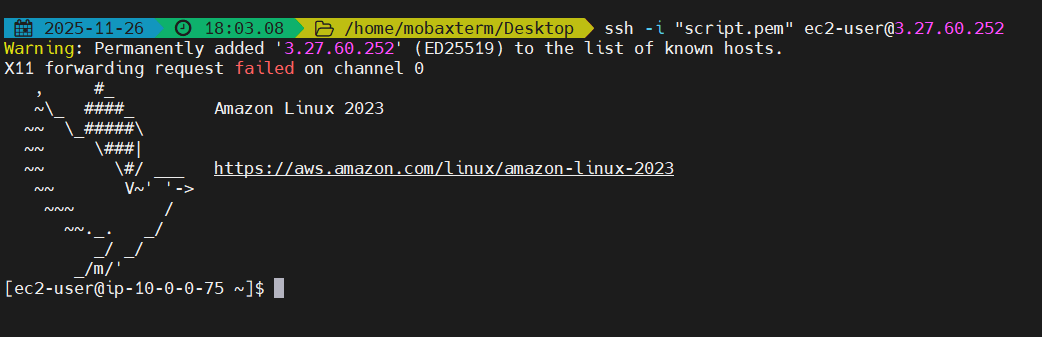




I have created a s3 bucket named myassigntechie



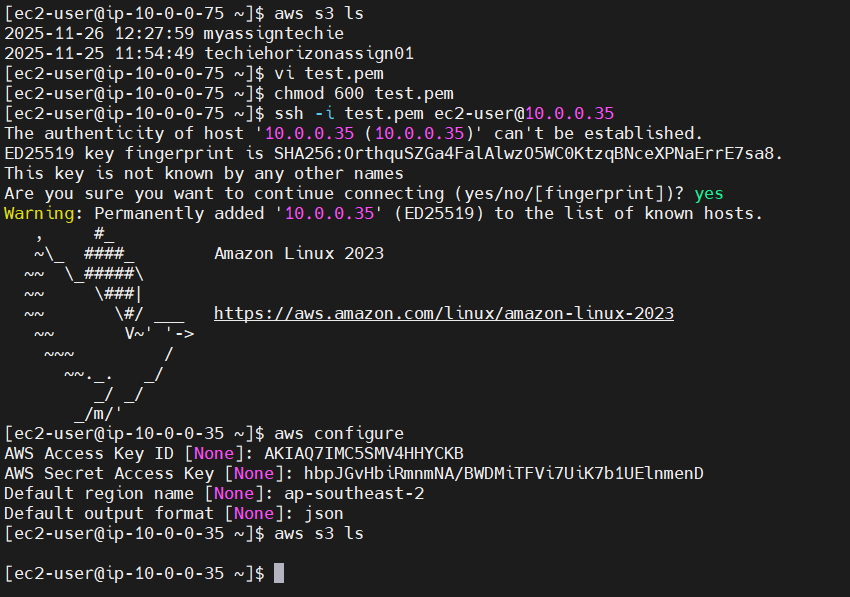
Now I have logged on to the private instance and given the access and secret key of the account in the aws configure menu.



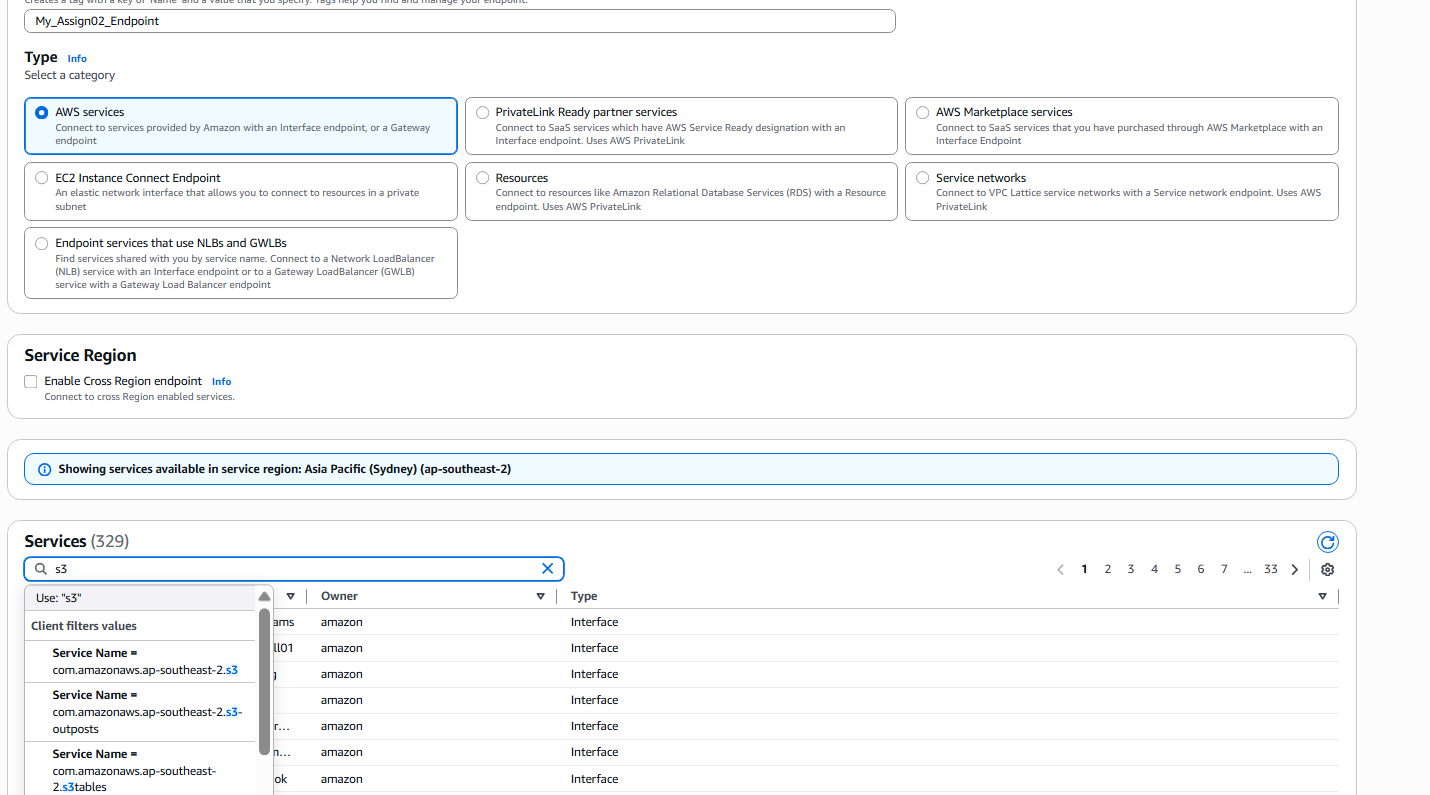
After giving the key and value. Check if we can able to see if any s3 buckets are accessible from the public instance and we can see the bucket names are shown .

Now jump to the private instance by copying the pem file content to the pem file created on the public instance and establish the connection to the private instance.

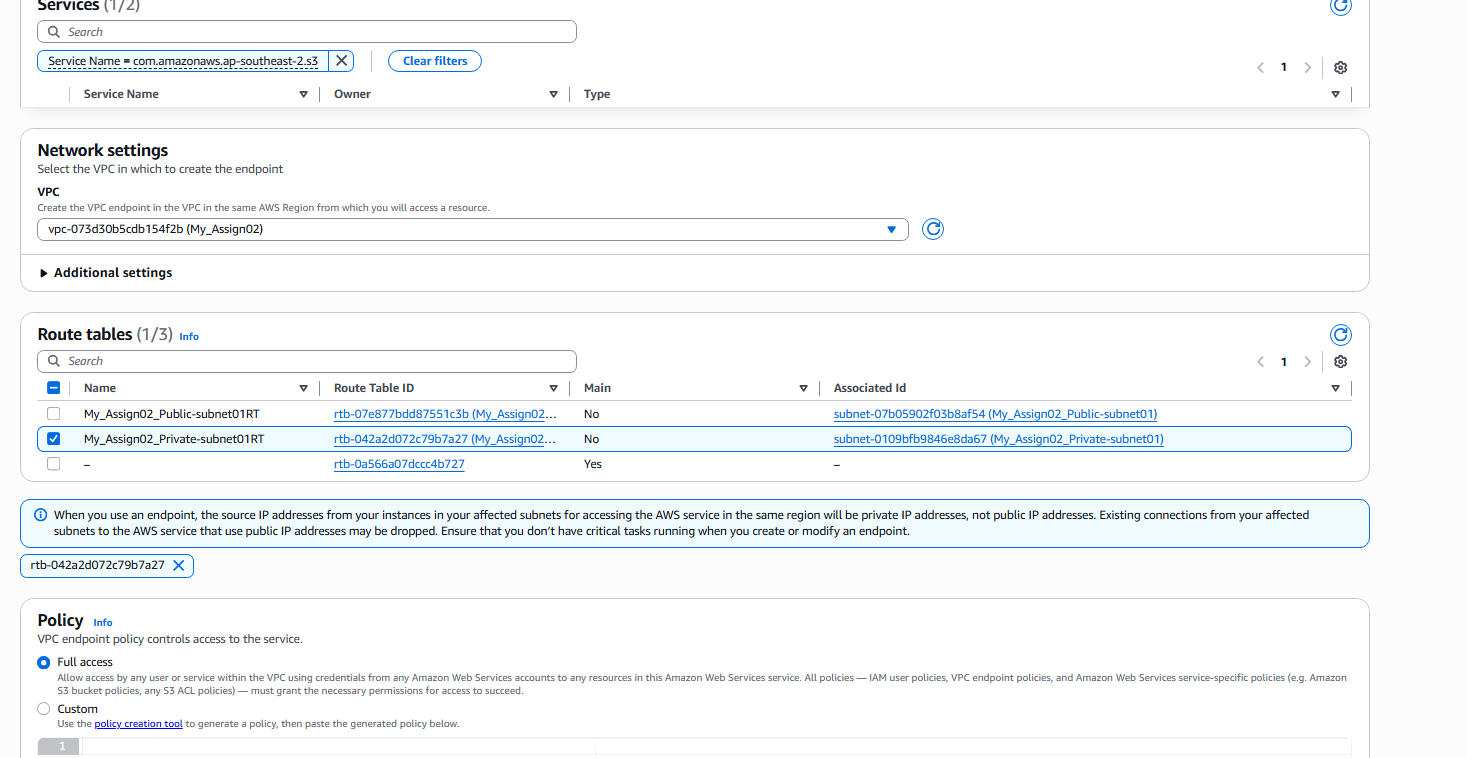
In the private instance, give the aws access and secret key values and check if the buckets in our region are visible or not. On executing aws s3 ls command we can see no s3 buckets are shown because there is neither the internet gateway or the nat gateway is configured to list the buckets.



Now create a endpoint connection by clicking create endpoint option, specify the name of the endpoint along with the Type option to AWS services. Select the service name=com.amazonaws-ap-southeast-2.s3



Select the vpc and also the route table ie.. private subnet and give policy access to full



Now after creating the endpoint execute the aws s3 ls, now we can see the bucket list as follow , end point had created a private connection access to the aws s3 service.

