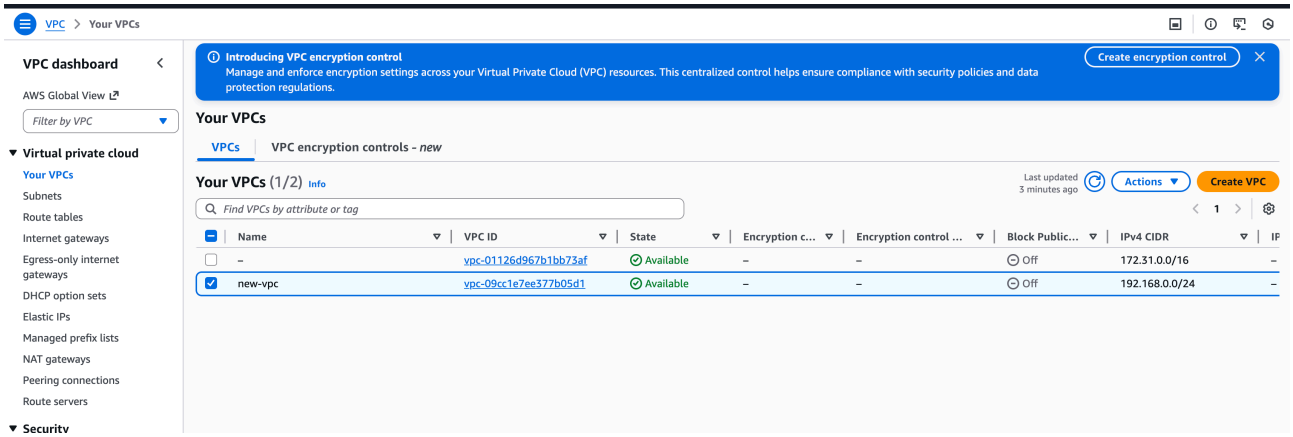
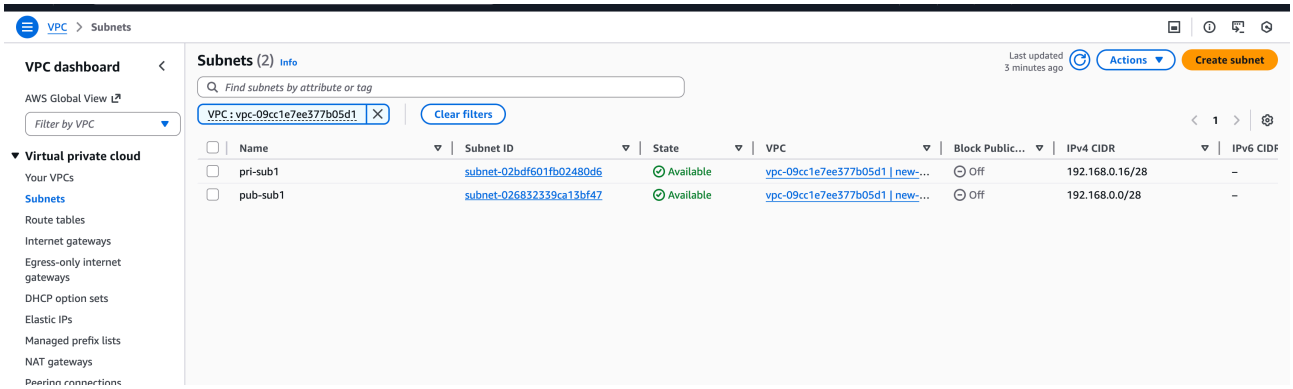
Vpc - 2 task

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

—> created a vpc with ipv4 cidr 192.168.0.0/24



—> created two subnets 1 priv and 1 public

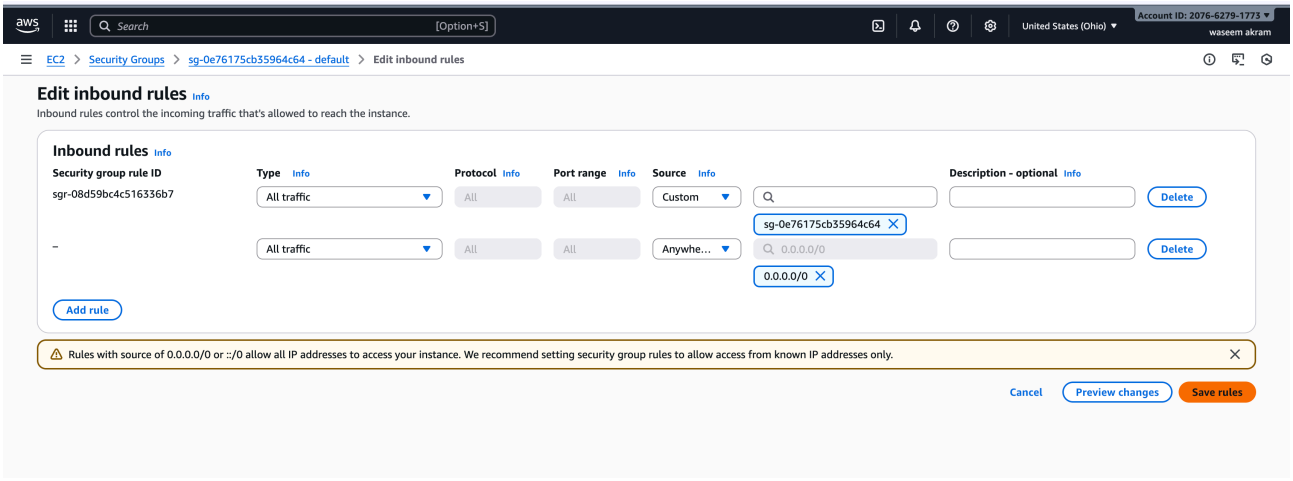


2)Enable VPC peering for cross-region.

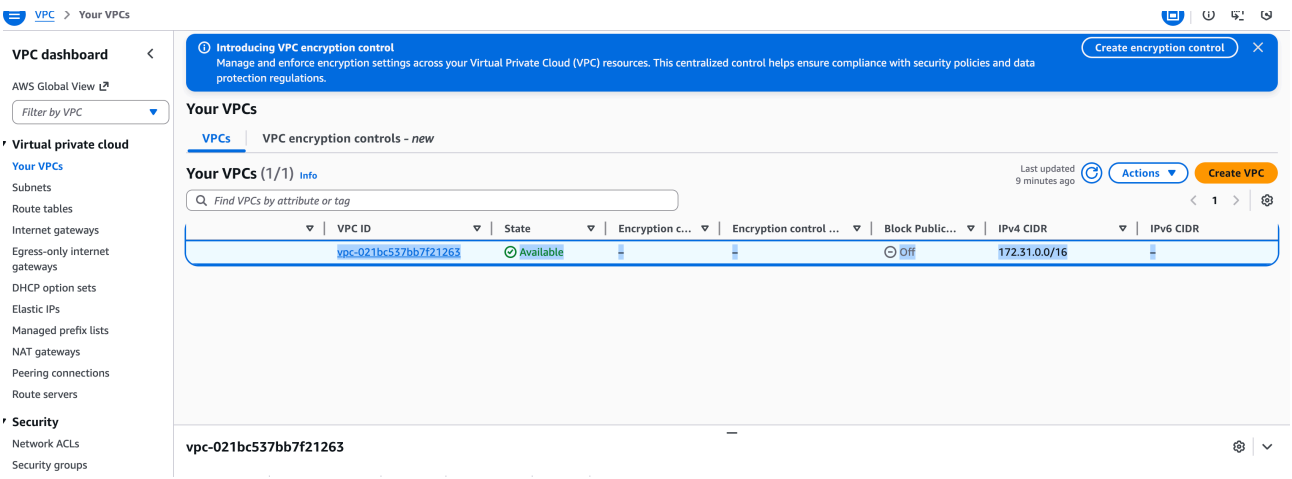
—> open your current region Stockholm and from above created vpc launch one instance using public subnet

—> and from other region Ohio launch a instance with default vpc and subnet
—> and give all traffic inbound in SG

—> sg of Ohio instanced

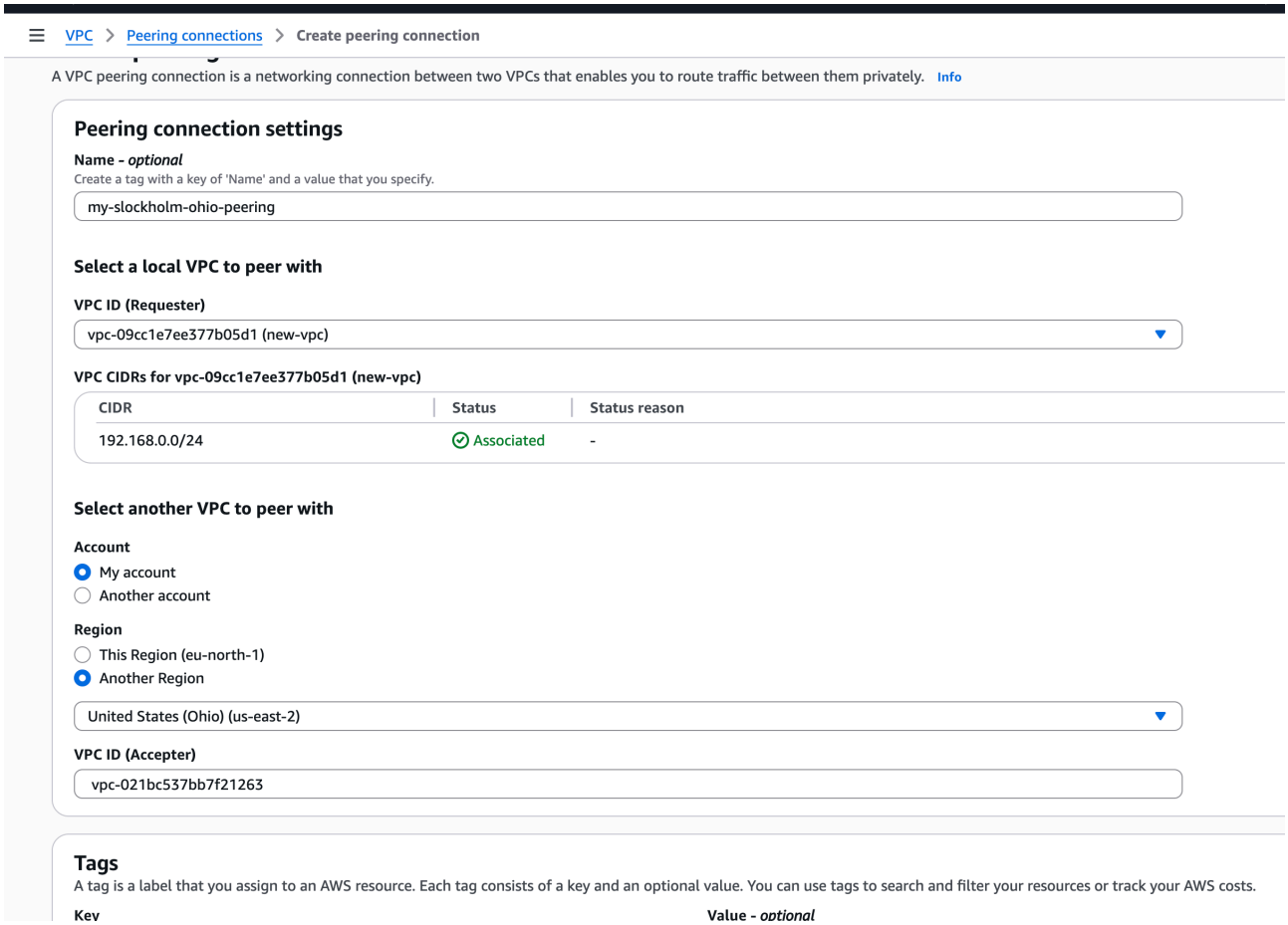


—> vpc of ohio



—> now lets start peering this two regions Stockholm and Ohio vpc's

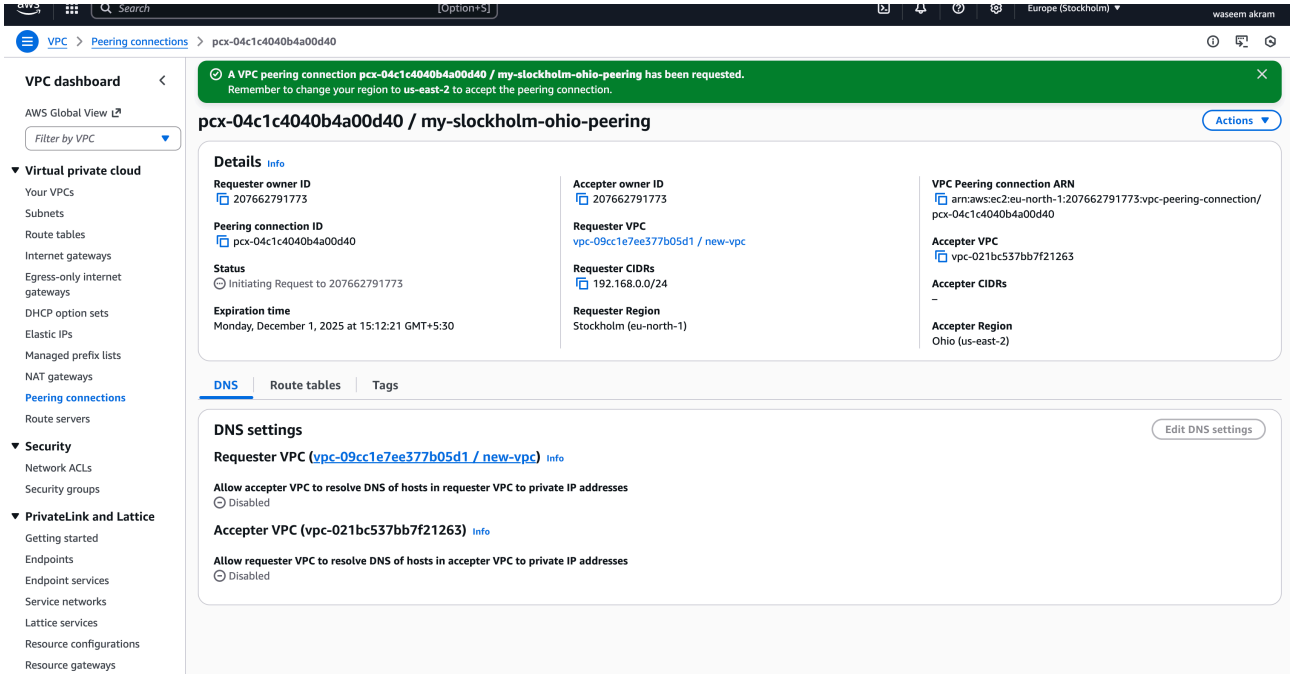
—> current region is Stockholm



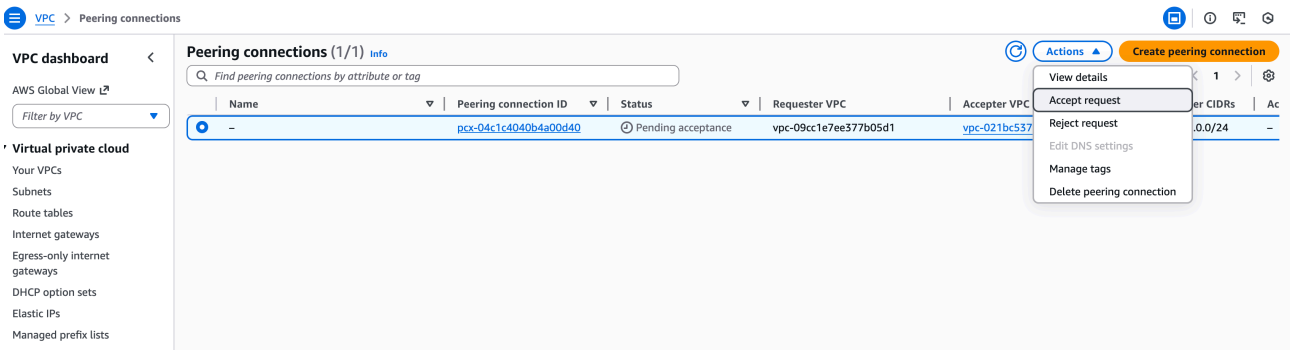
—> go to vpc
—> peering connections
—> create peering connection

Add details of vpc’s and

—> peering request send to Ohio region

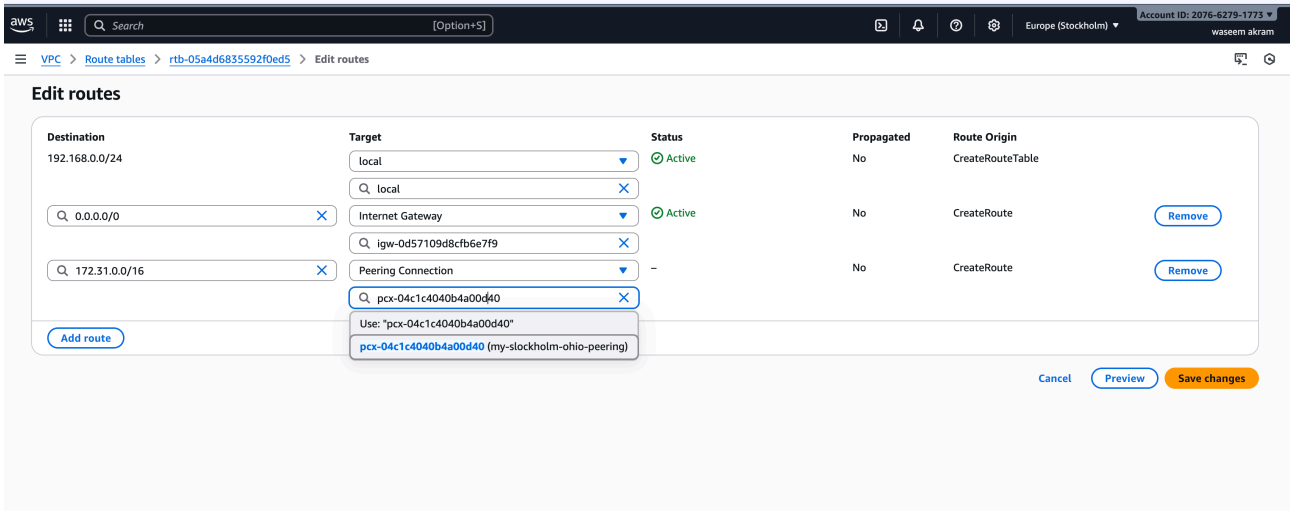


—> open your Ohio region vpc —> peering connections

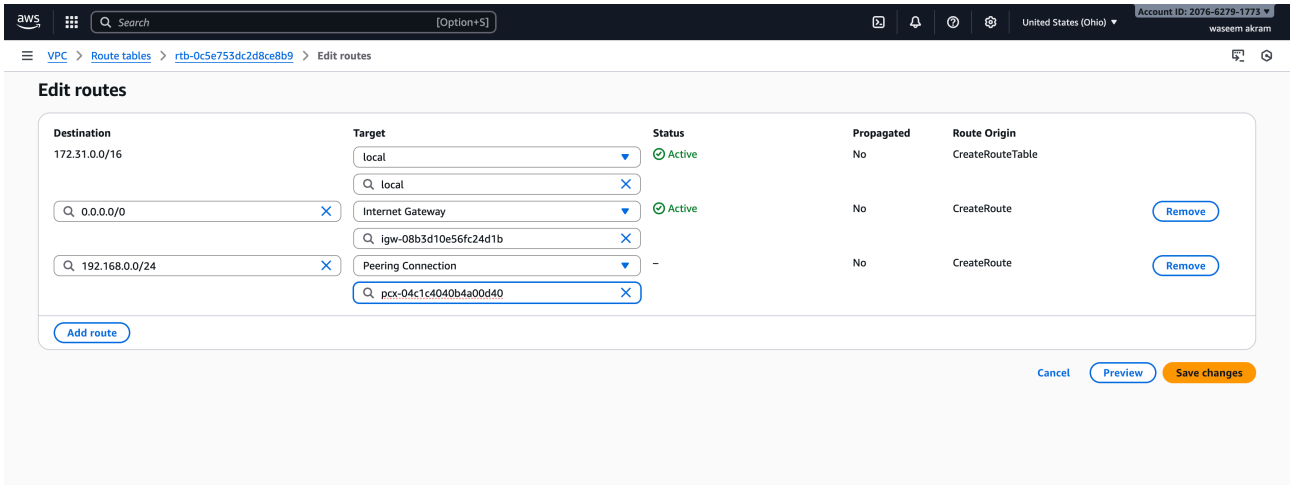


—> we can see peering request from Stockholm is received

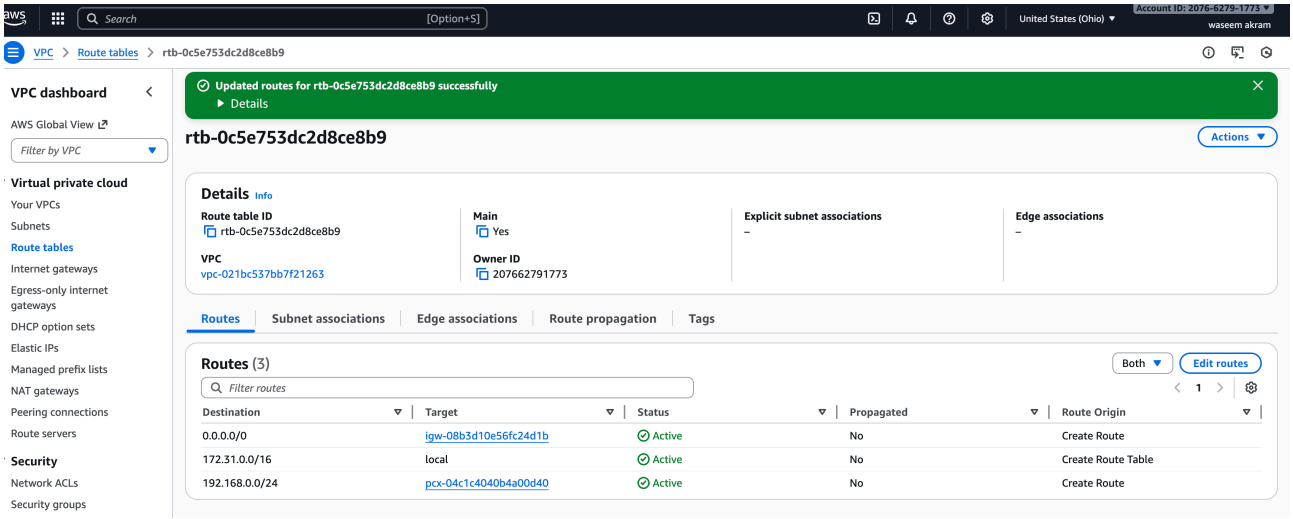
—> now go to both regions public routes and add Ohio ipv4 rang for Stockholm and stockholm ipv4 for ohio in public routes
—> below is Stockholm public route



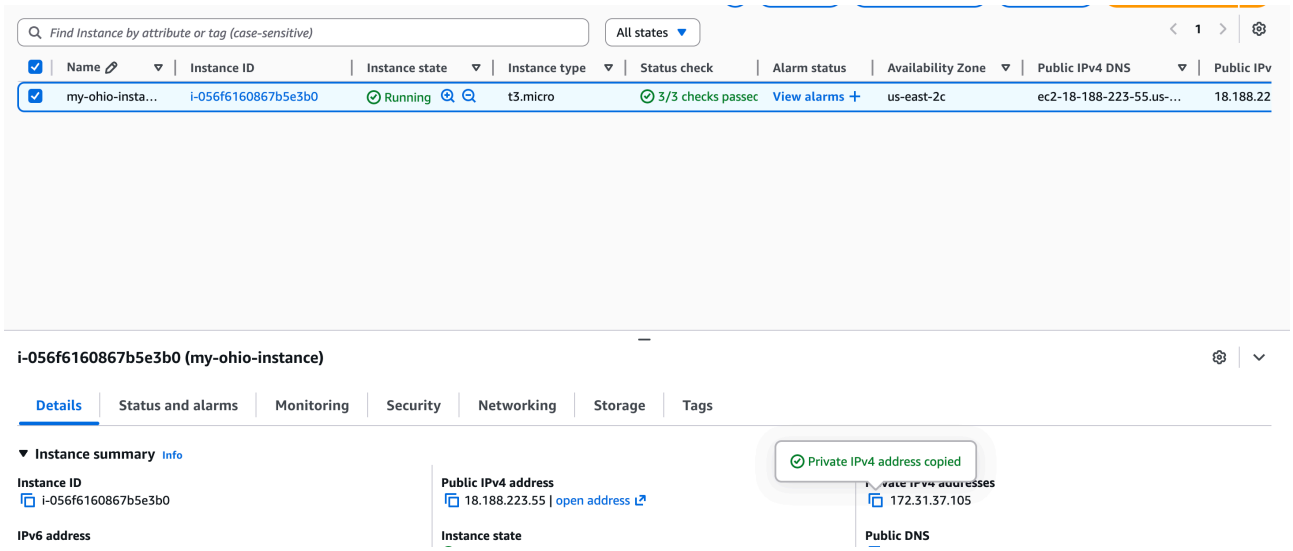
—> below is Ohio public route



—> we can see peering target added in both route

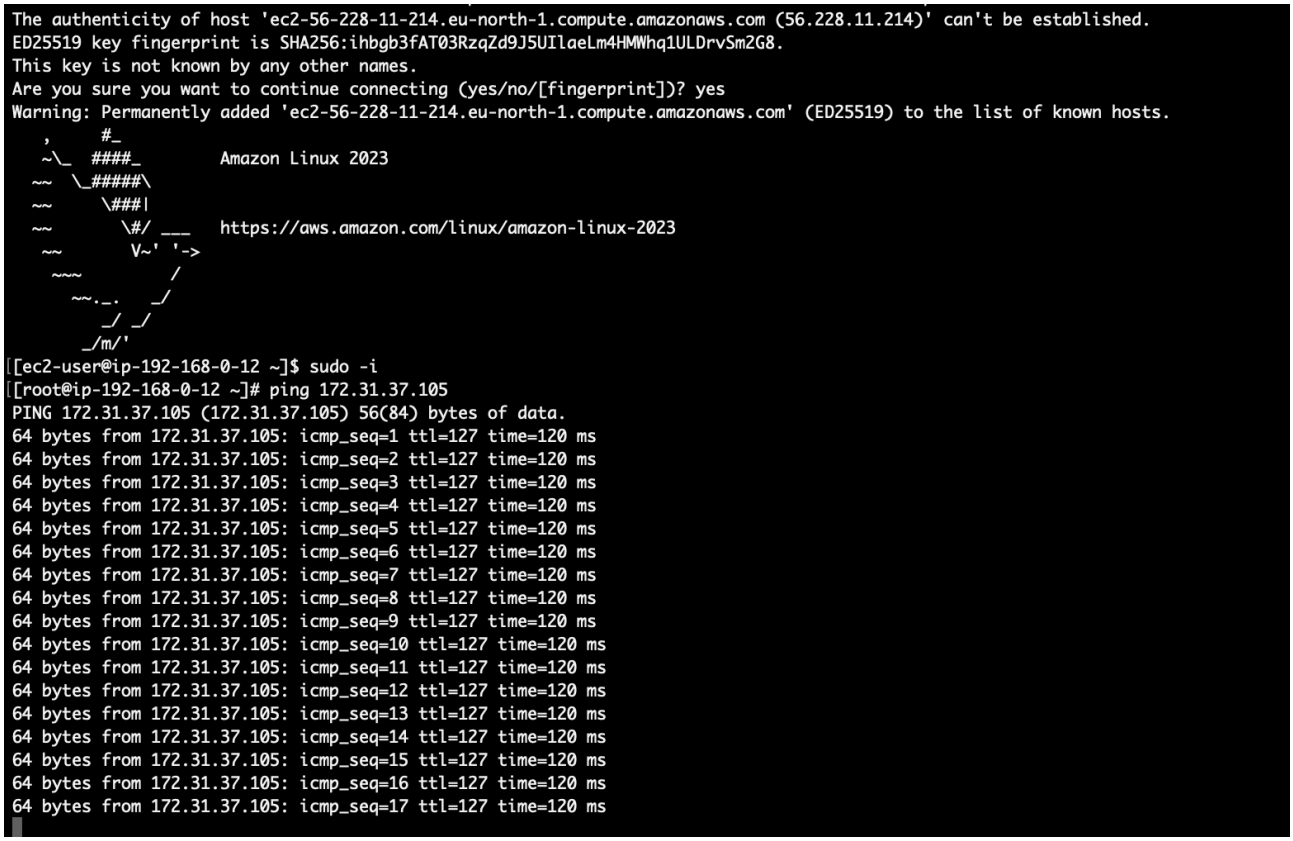


—> now copy the Ohio instance private ip and



—> now open the server of Stockholm and ping the private ip of 172.31.37.105

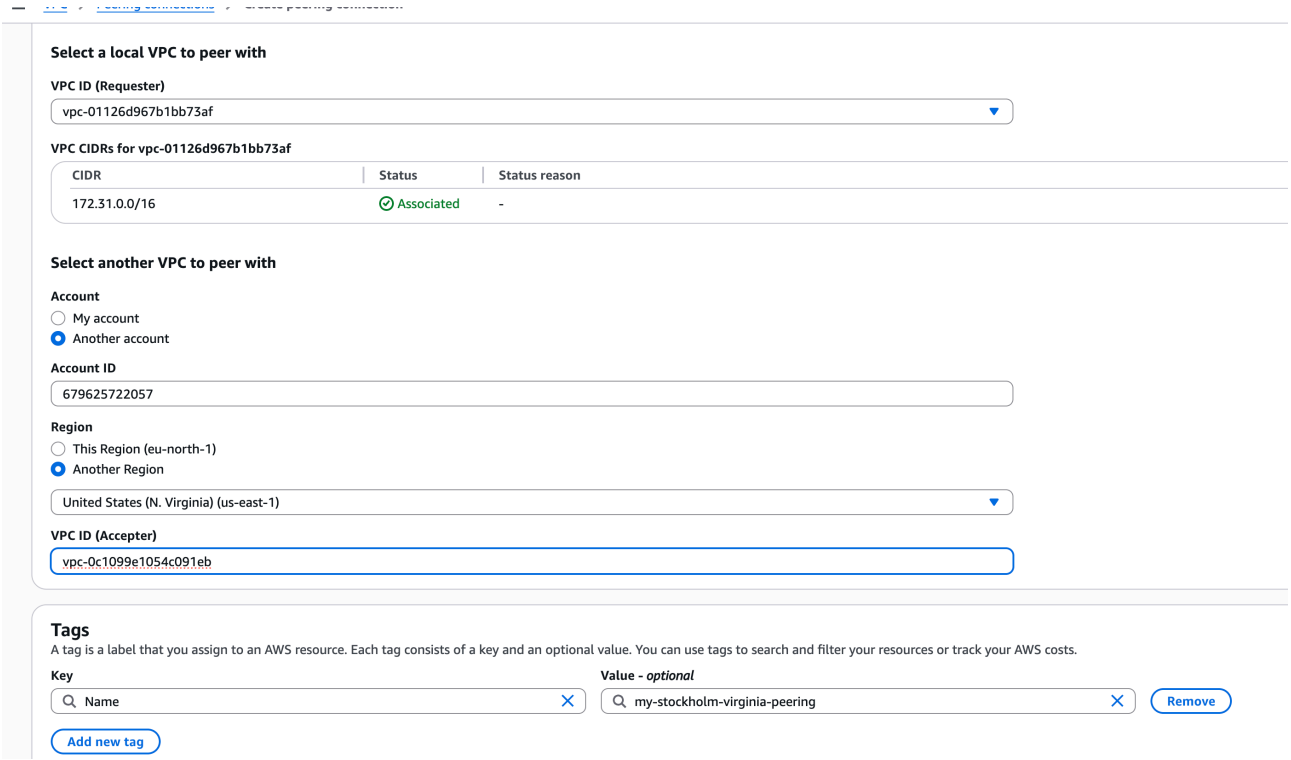
—> we can see the two instance are peering the private data of Ohio region instance



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

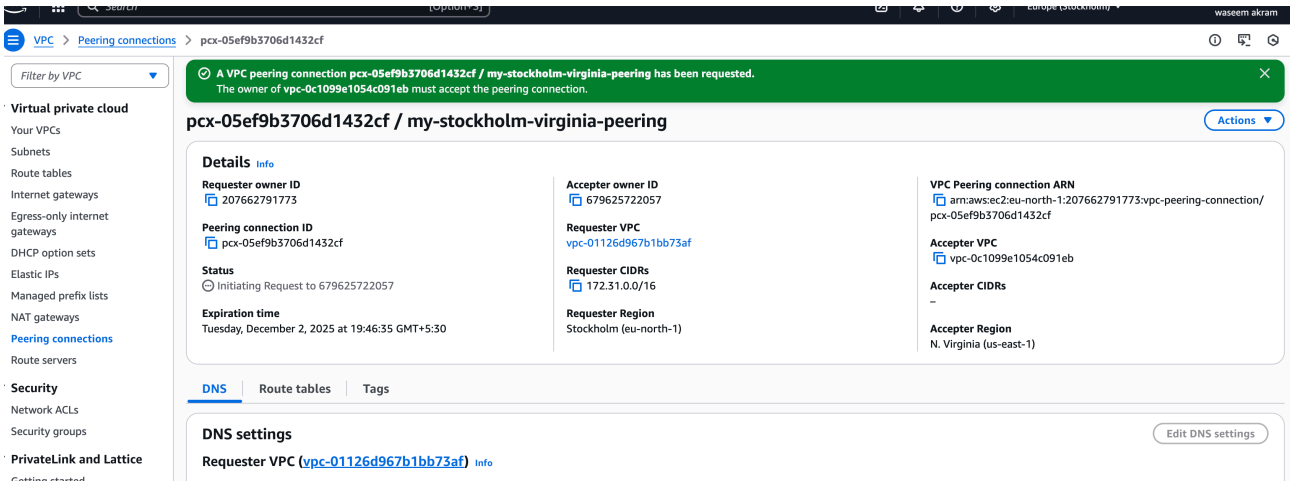
—> created a vpc with 172.31.0.0/16 range
—> and created public subnet
—> created a route and added internet gate to it

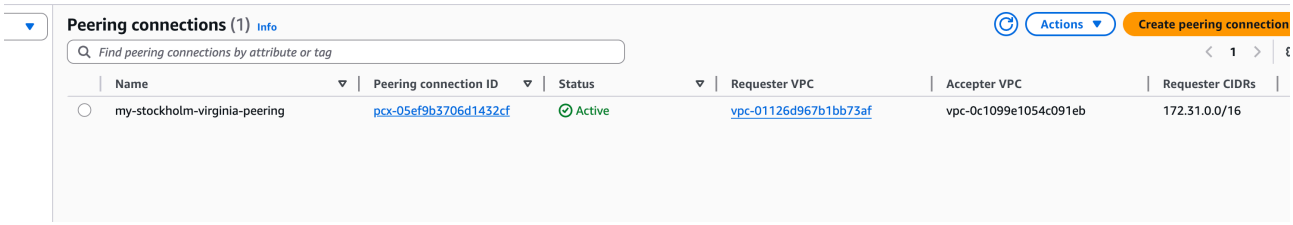
—> now go to vpc
—> peering connections
—> give your vpc details
—>select another vpc to peer with
—> friends account id
—> region select according to friends region
—> add friends vpc id



—> request send to the friends vpc peering

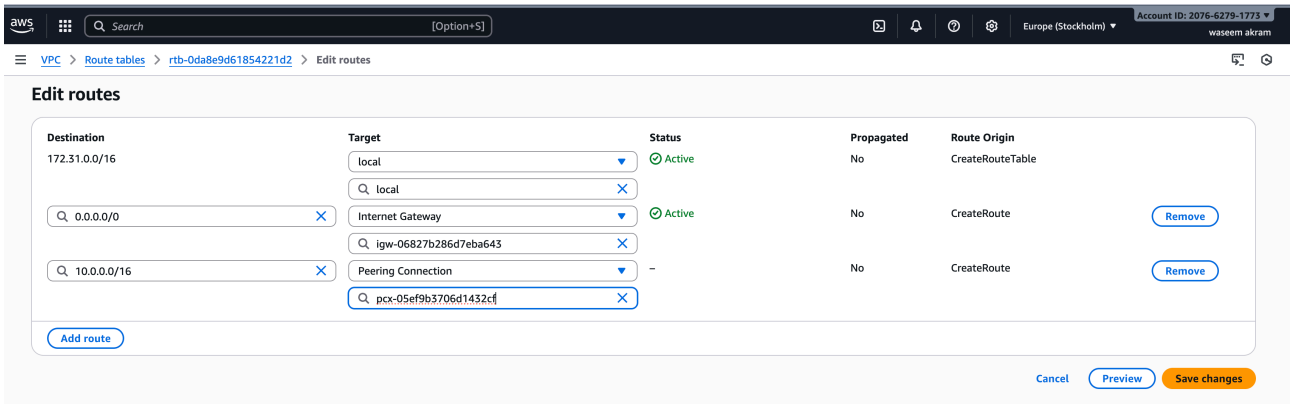
—> later after friend accepting the peer request





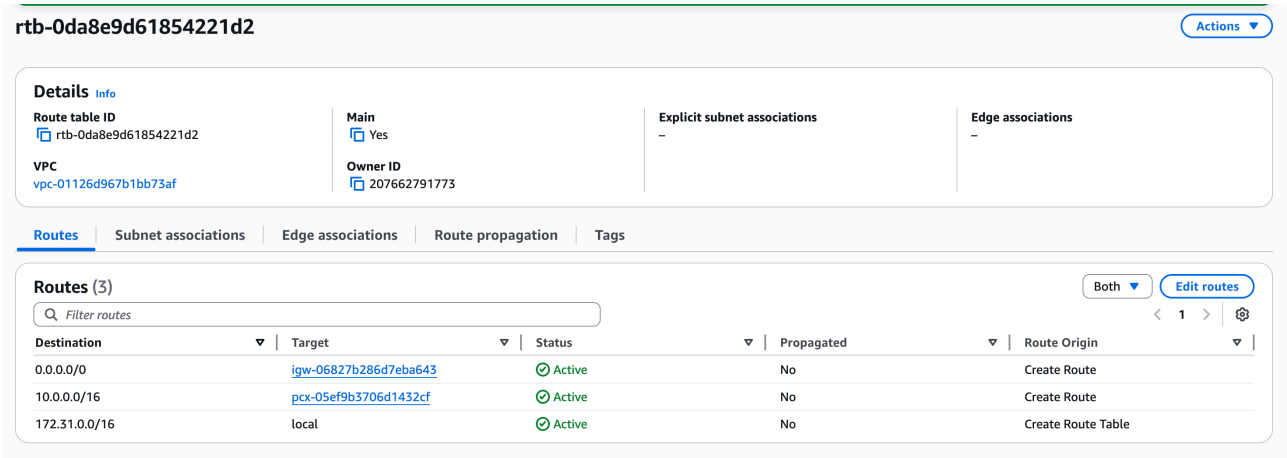
—> go and add route of the peering request
—> now after my friend accept the peer request we can see there is active status means peering is now successfully running and active

next
—> my public route

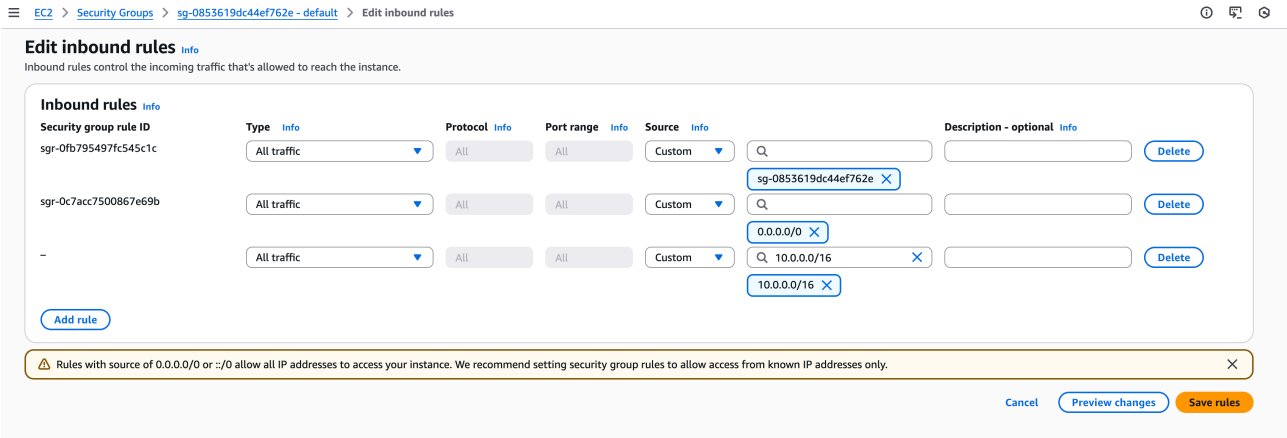


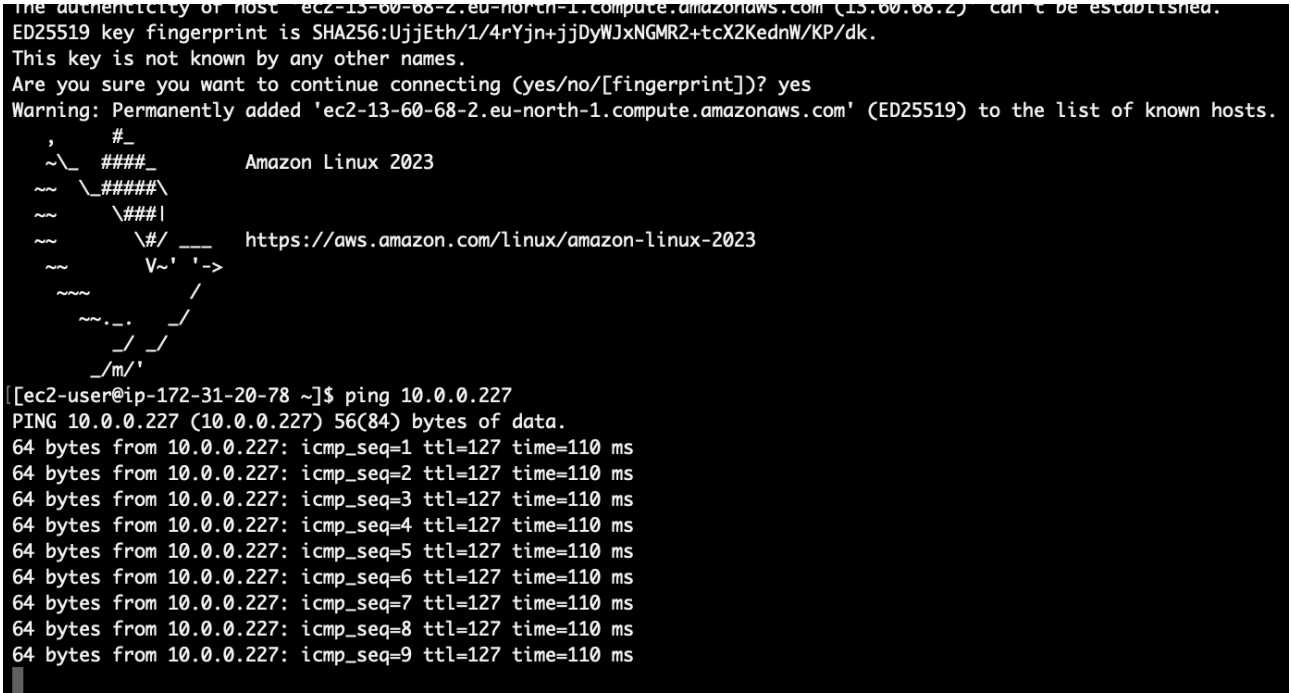
—> added friends ipv4 range and added peering connectivity

And saved



—> on other side friends has also added the same my ipv4 172.31.0.0/16 as peering connectivity
—> edit the secutity group and add all traffic for it



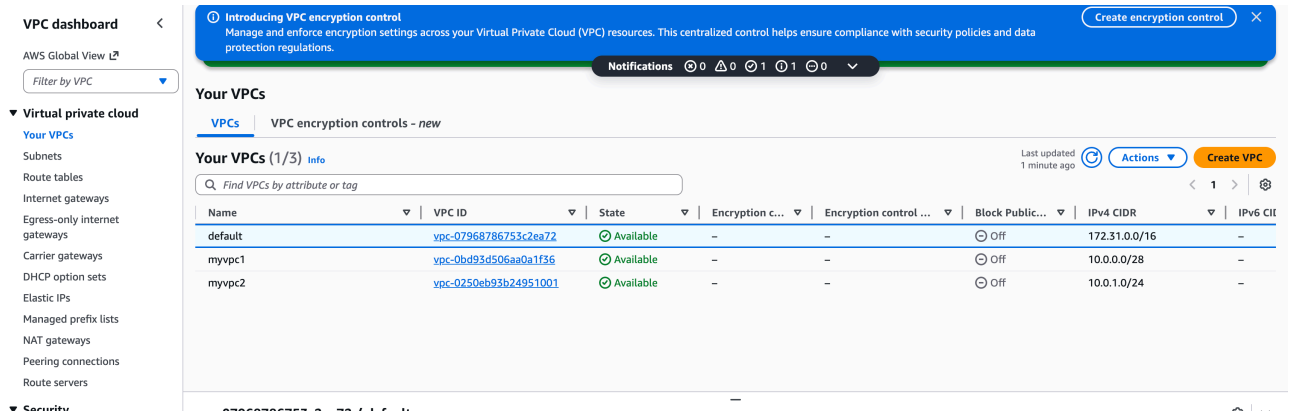


—> add the same range in your inbound security group also
—> ping the friends instance private ip 10.0.0.227
—> its working

4)Set up a VPC Transit Gateway.

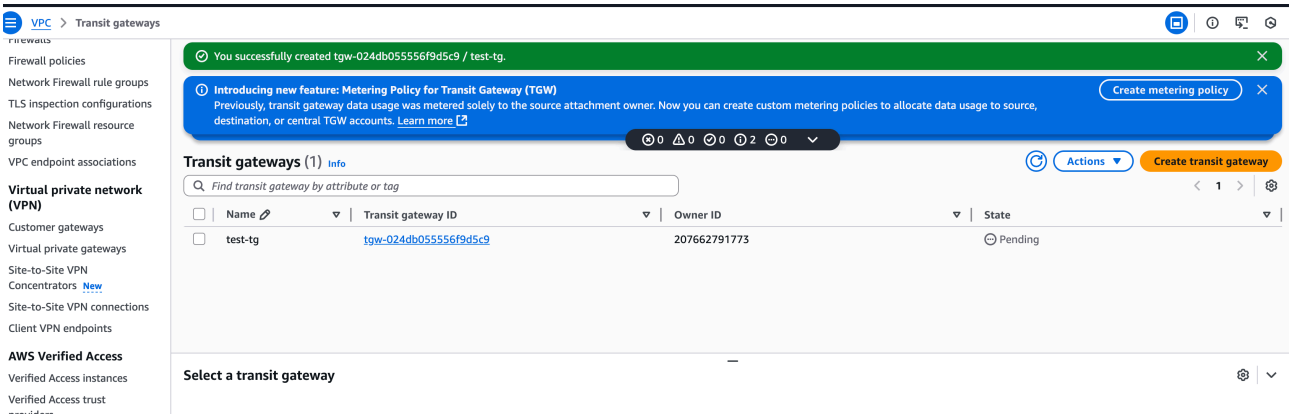
—> create a no of vpc’s
Given below ranges for three vpc’s

10.0.1.0/28
172.31.0.0/16
10.0.0.0/28

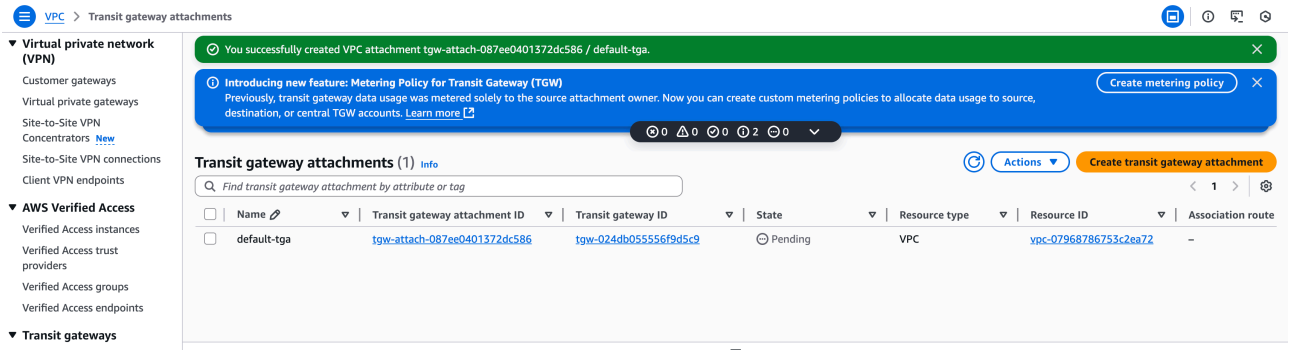


—> create three subnets with with one subnet for one vpc

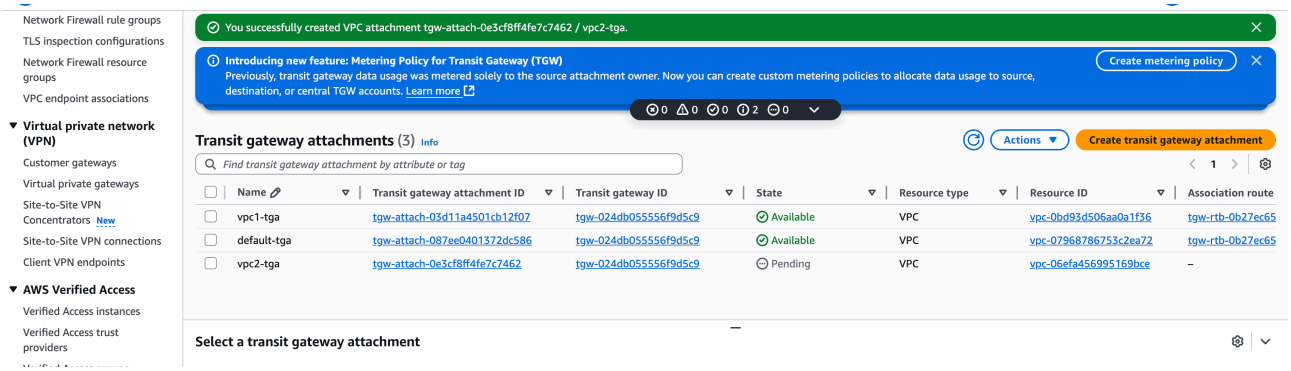
Go in vpc and check for transit gateway
—> created a transit gateway



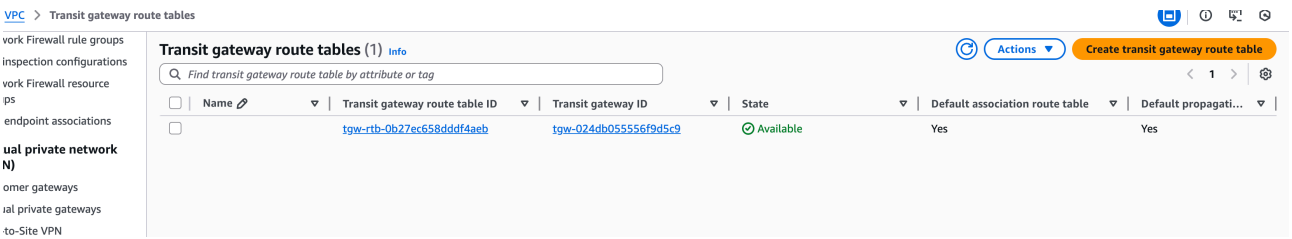
—> now attach the all vpc’s with the created transit gateway



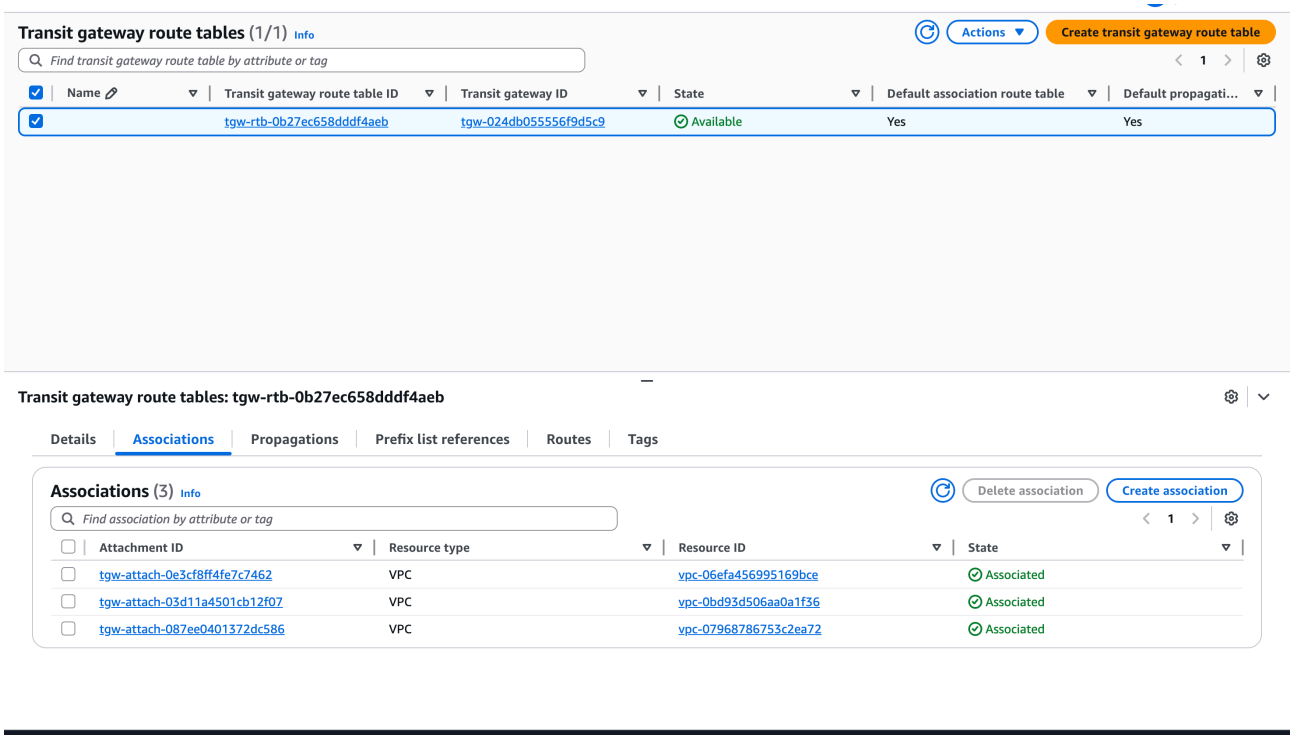
—> attached the three vpc’s



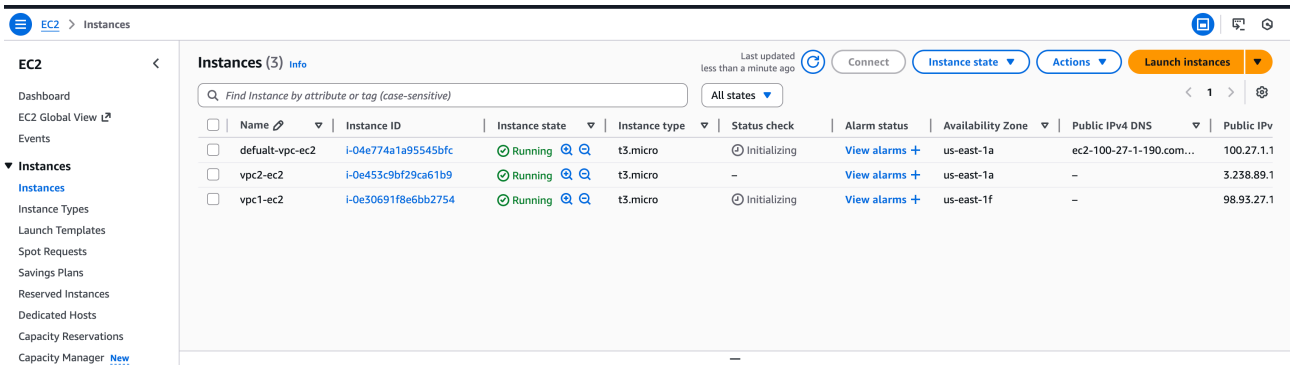
—> this is our transit gateway route table in which we have to add all the ranges in each route of vpc’s



—> route table of transit gate way which is attached to three vpc’s

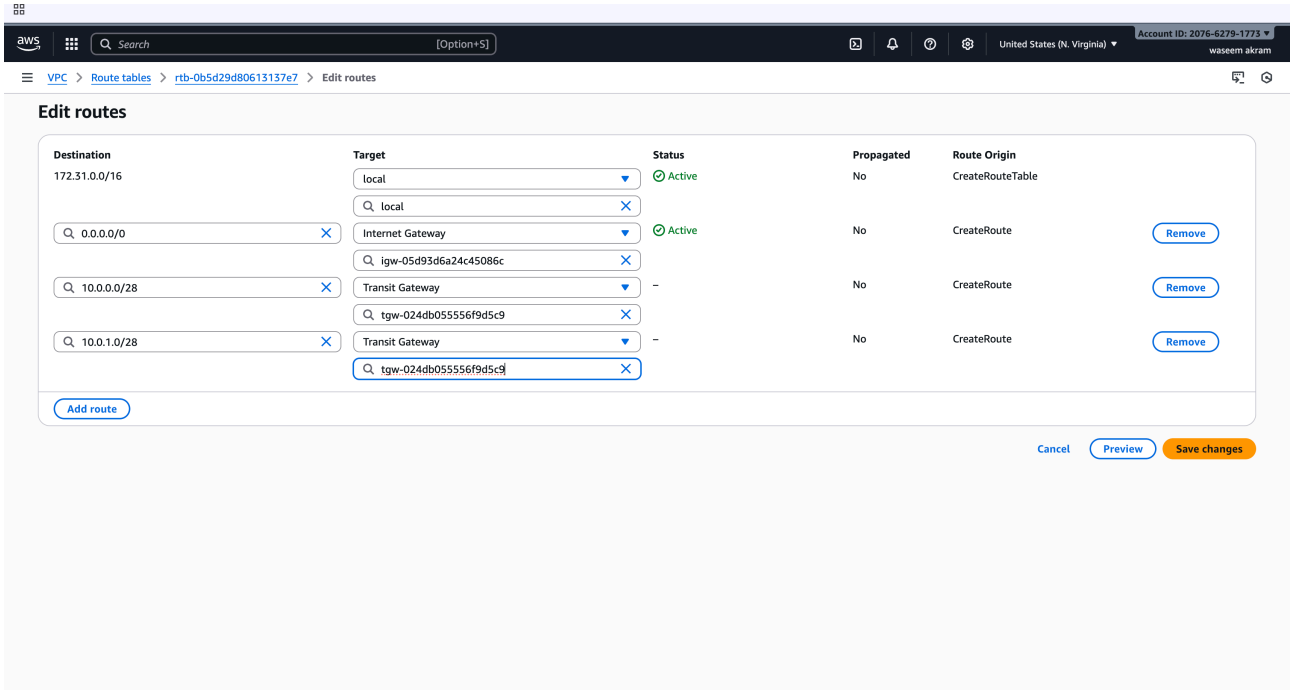


—> launch the three instances from each each vpc’s



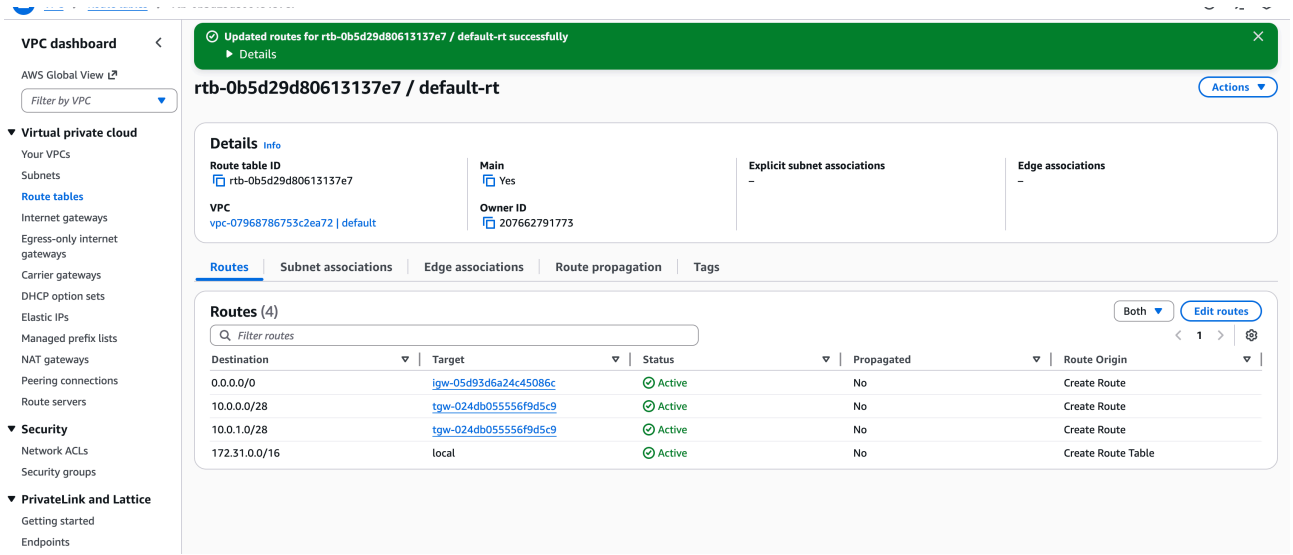
—>and choose that subnet which we created after creating vpc’s

—>now go inside the route table of each each vpc's and add other two vpc ipv4 range with transit gateway

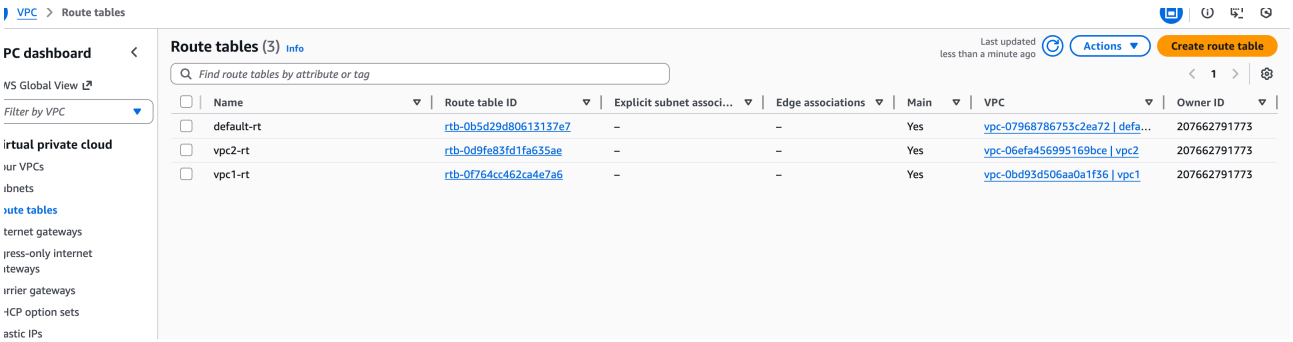


—> like we are in default vpc with 172.31.0.0/16
We are adding other vpc range of 10.0.0.0/16 (vpc1)
And vpc range of 10.0.1.0/16 (vpc2)

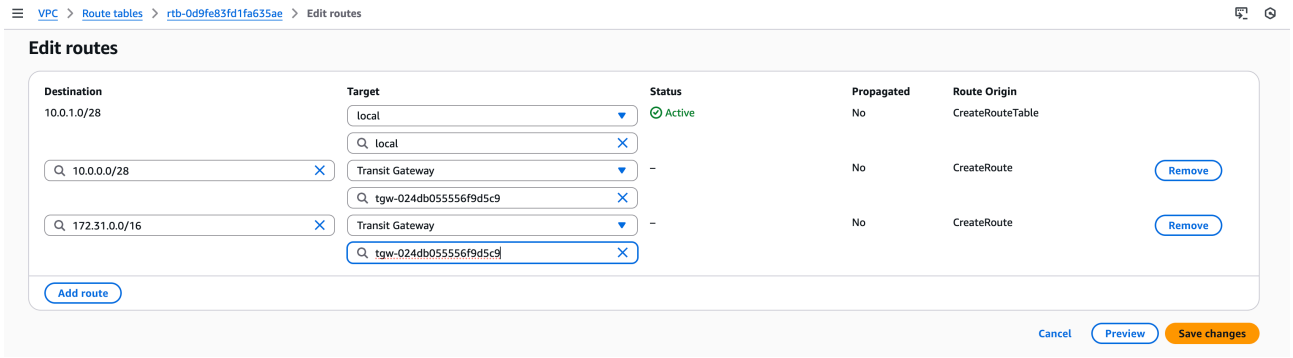
—> similarly add other two route tables of other vpc's just like above



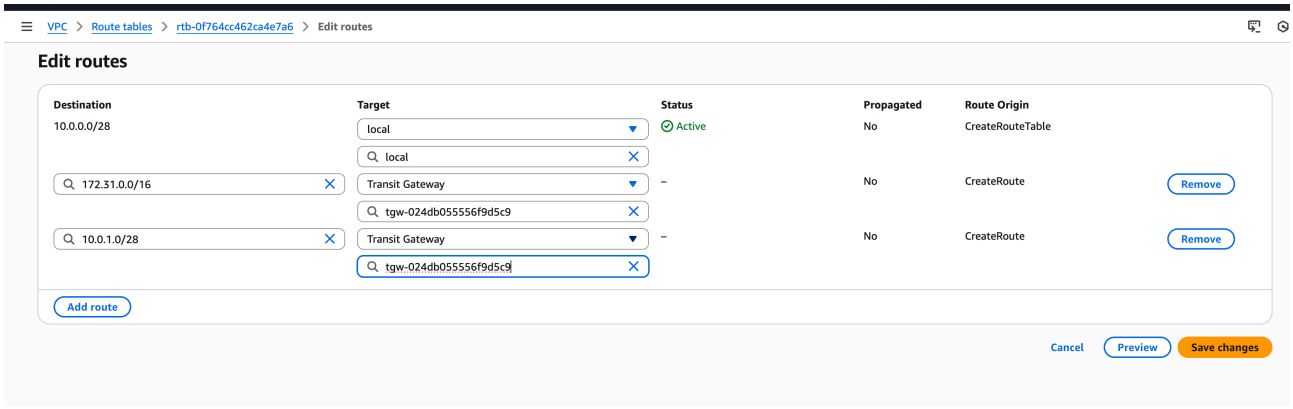
—> these are your route tables of defualt vpc ,vpc1,vpc2



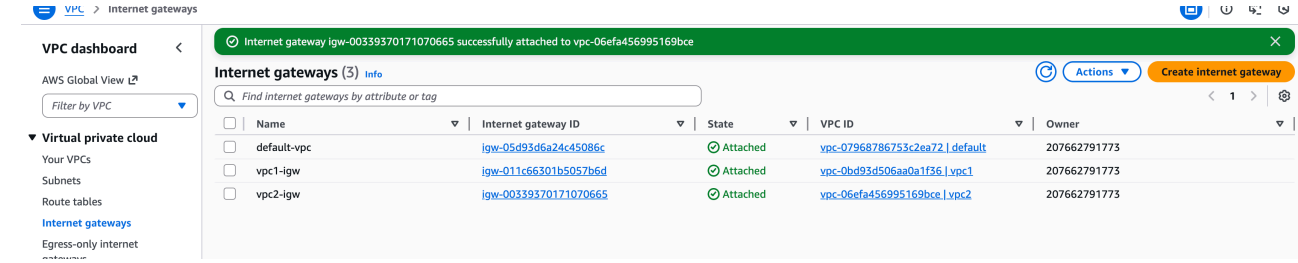
vpc2 route



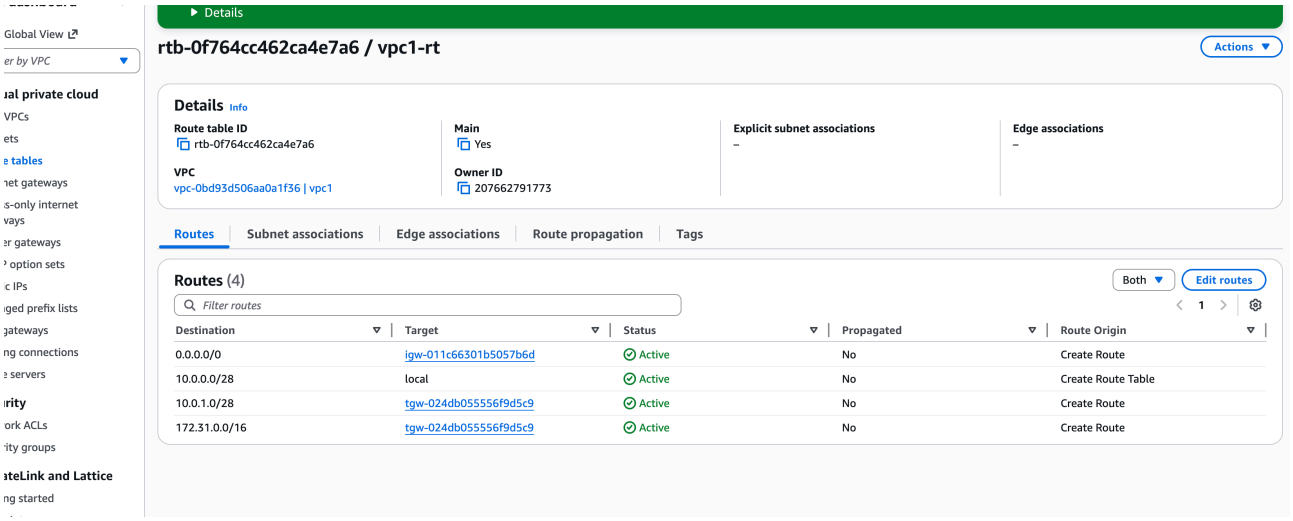
vpc1 route



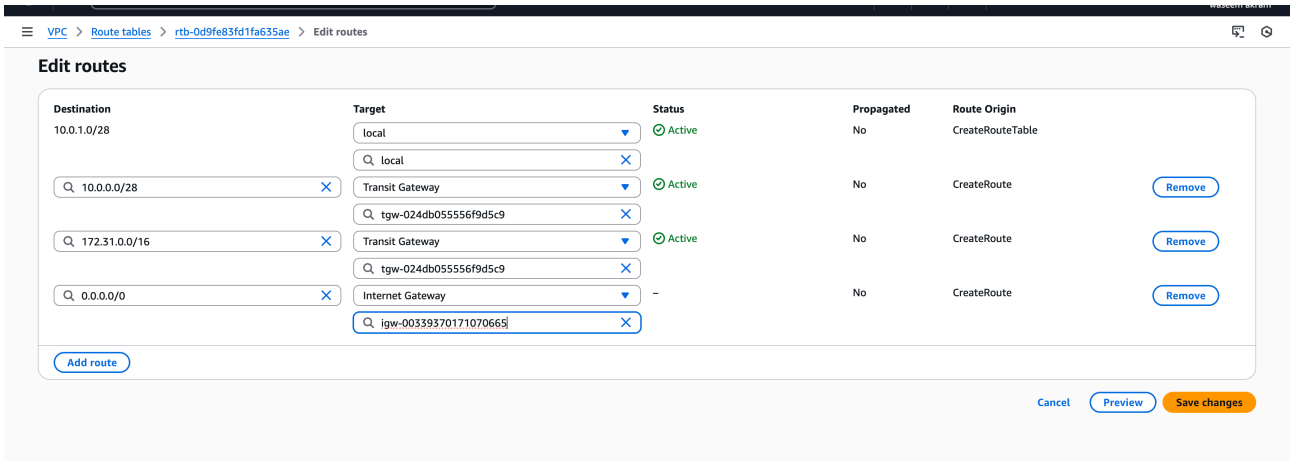
—> and create internet gateway for each route which is attached to its vpc’s



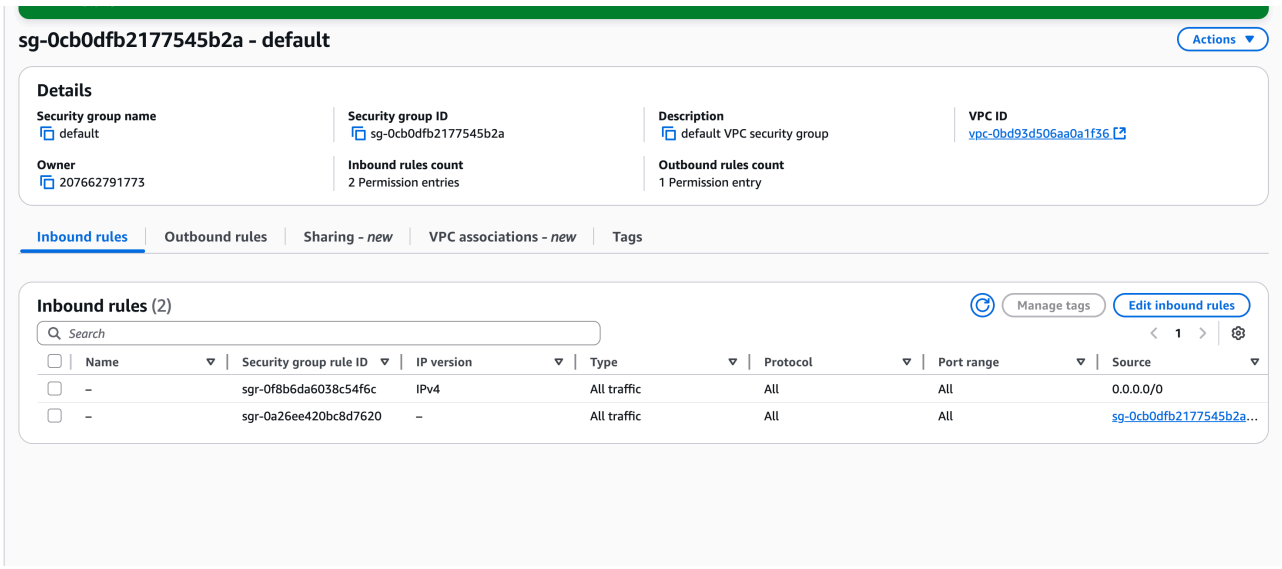
—> and add the internet gate way to other two route’s also



—> add internet gate way to all routes



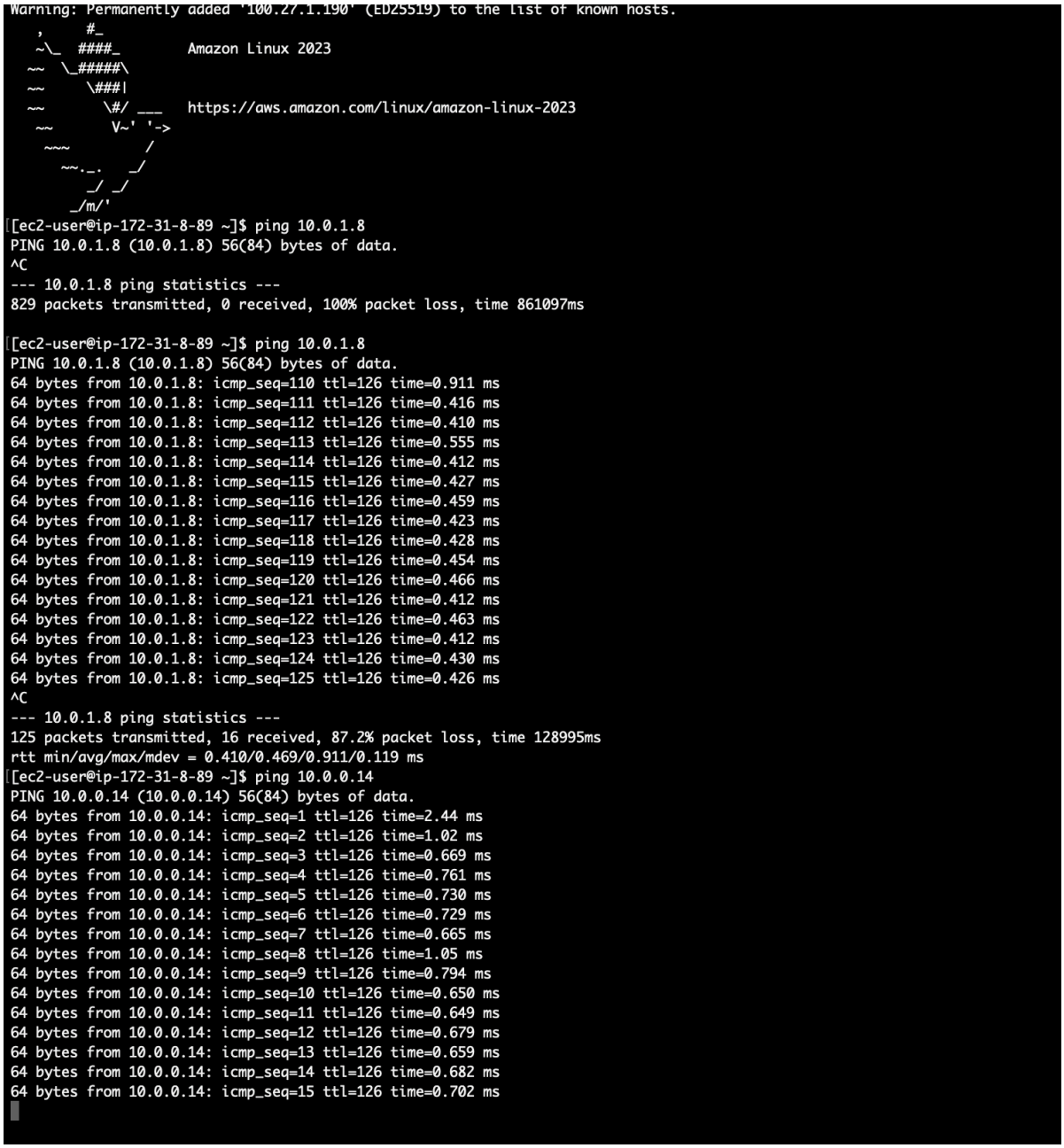
—> add all traffic to all security groups of each instances



—> this will allow to run the connectivity

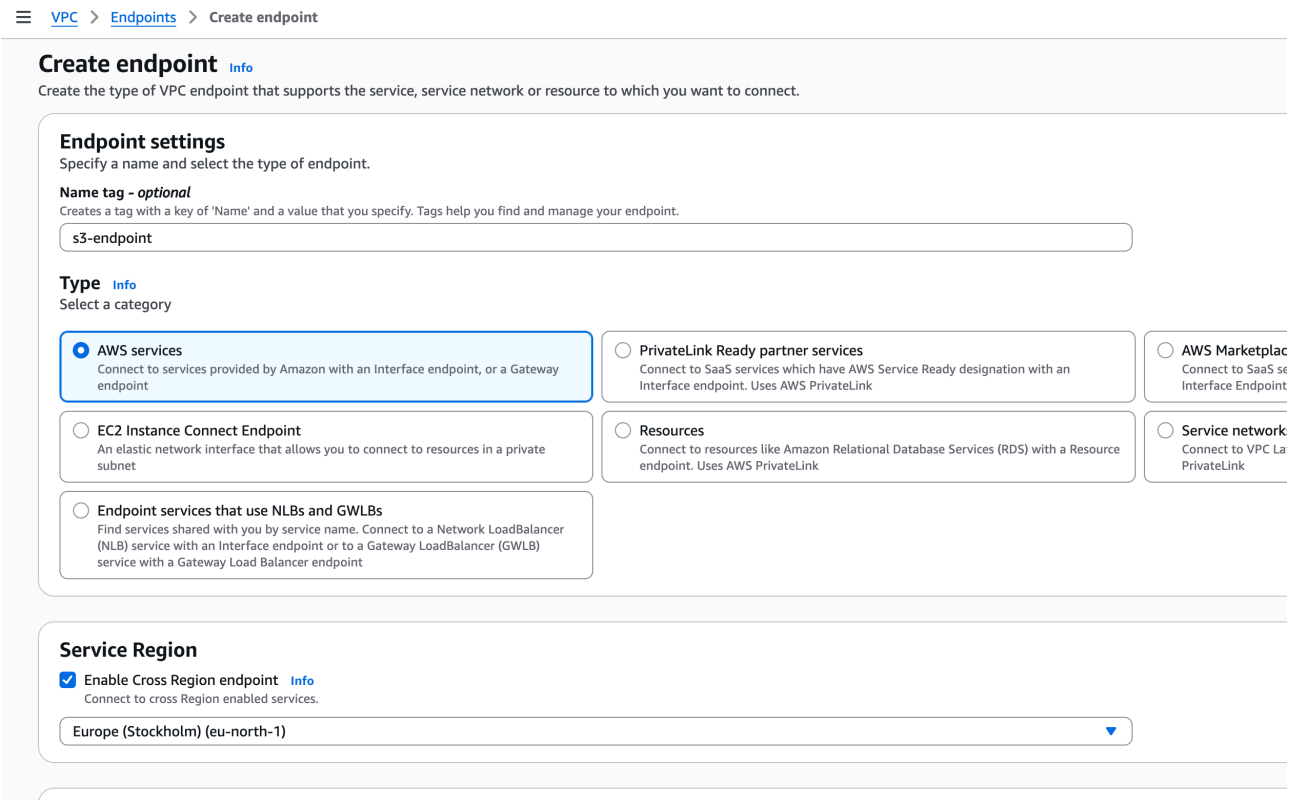
```
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-8-89 ~]$ ping 10.0.1.8
PING 10.0.1.8 (10.0.1.8) 56(84) bytes of data.
^C
--- 10.0.1.8 ping statistics ---
829 packets transmitted, 0 received, 100% packet loss, time 861097ms

[ec2-user@ip-172-31-8-89 ~]$ ping 10.0.1.8
PING 10.0.1.8 (10.0.1.8) 56(84) bytes of data.
64 bytes from 10.0.1.8: icmp_seq=110 ttl=126 time=0.911 ms
64 bytes from 10.0.1.8: icmp_seq=111 ttl=126 time=0.416 ms
64 bytes from 10.0.1.8: icmp_seq=112 ttl=126 time=0.410 ms
64 bytes from 10.0.1.8: icmp_seq=113 ttl=126 time=0.555 ms
64 bytes from 10.0.1.8: icmp_seq=114 ttl=126 time=0.412 ms
64 bytes from 10.0.1.8: icmp_seq=115 ttl=126 time=0.427 ms
64 bytes from 10.0.1.8: icmp_seq=116 ttl=126 time=0.459 ms
64 bytes from 10.0.1.8: icmp_seq=117 ttl=126 time=0.423 ms
64 bytes from 10.0.1.8: icmp_seq=118 ttl=126 time=0.428 ms
64 bytes from 10.0.1.8: icmp_seq=119 ttl=126 time=0.454 ms
64 bytes from 10.0.1.8: icmp_seq=120 ttl=126 time=0.466 ms
64 bytes from 10.0.1.8: icmp_seq=121 ttl=126 time=0.412 ms
```

—> I have pinged the vpc 1 instance and also pinged the vpc 2 instance with default vpc

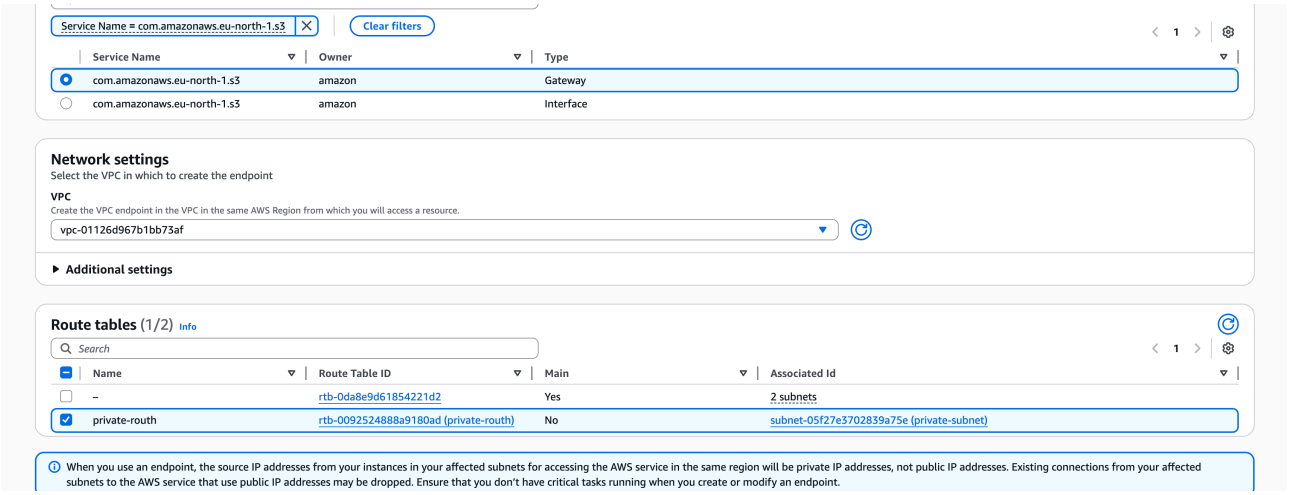


5)Set up a VPC Endpoint

—> go to vpc
—> search for endpoint

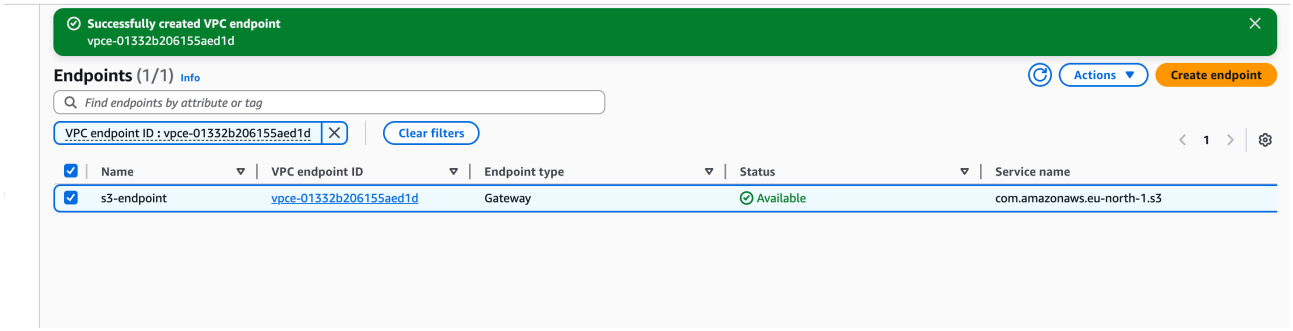


—> Endpoint is a service which helps us to communicate with aws services without internet

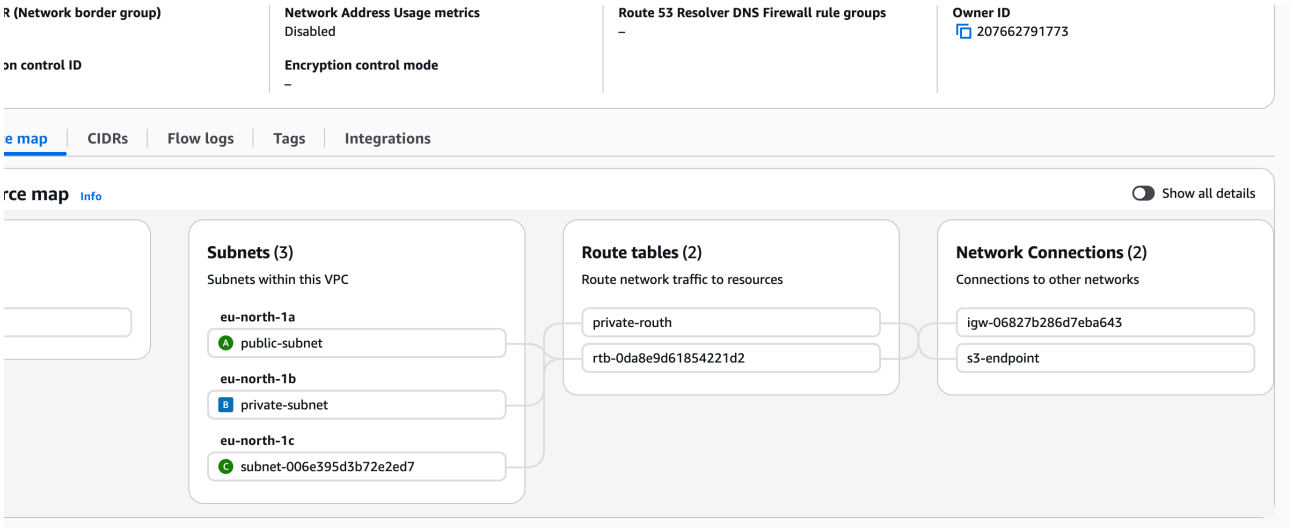


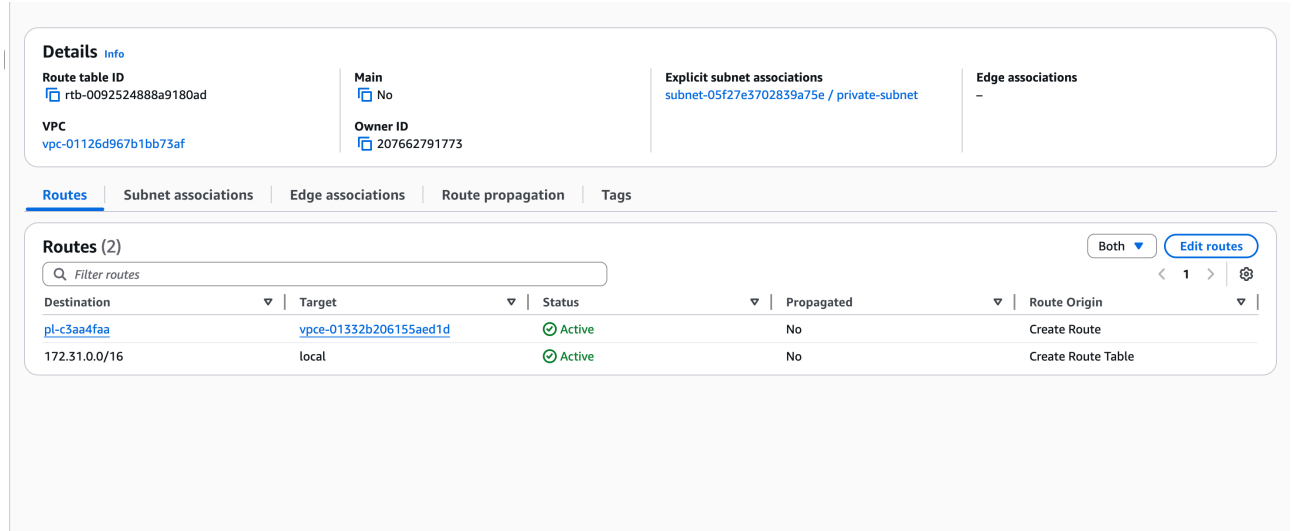
—> select s3 as services add vpc details ,select the private route
—> and create

—> endpoint is created



—> check the private route we can see the endpoint is connected to private route





—> this will communicate with aws services without internet