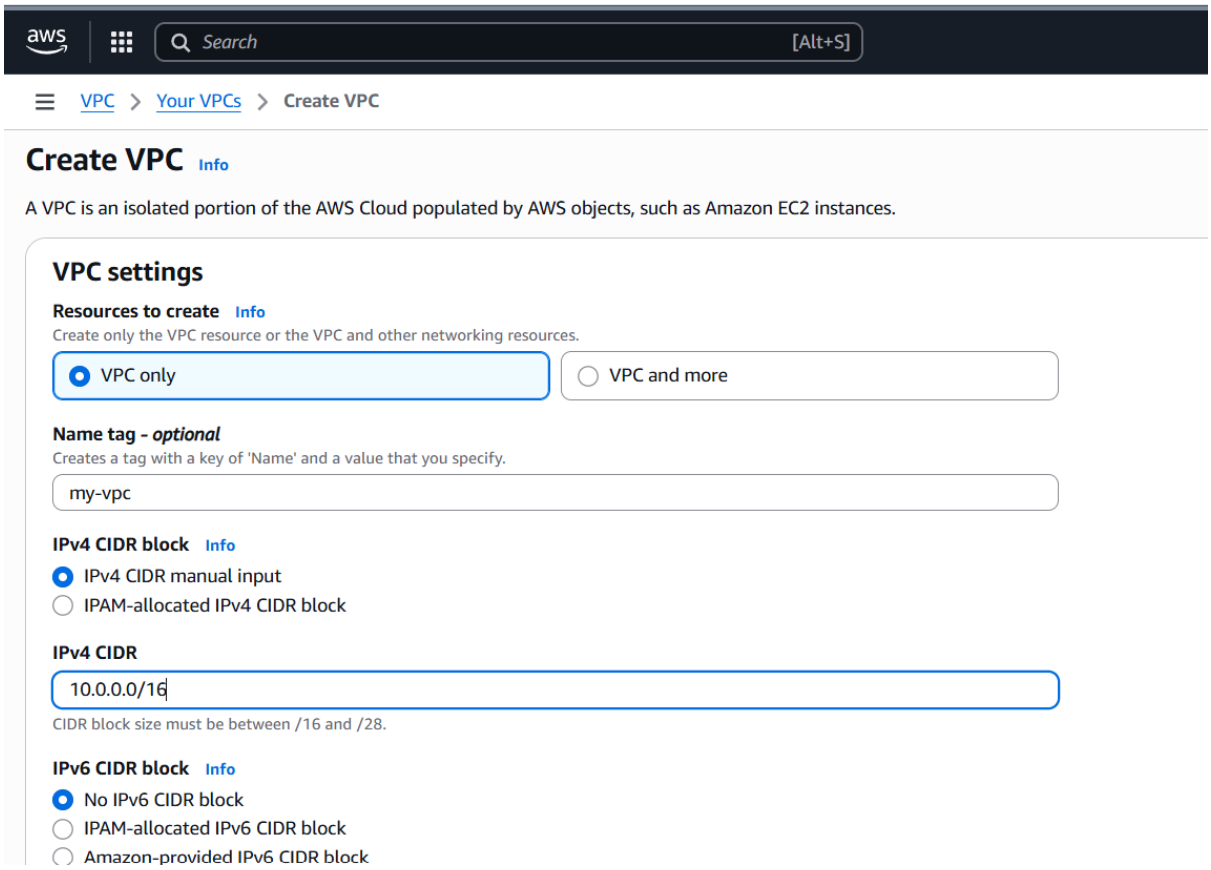
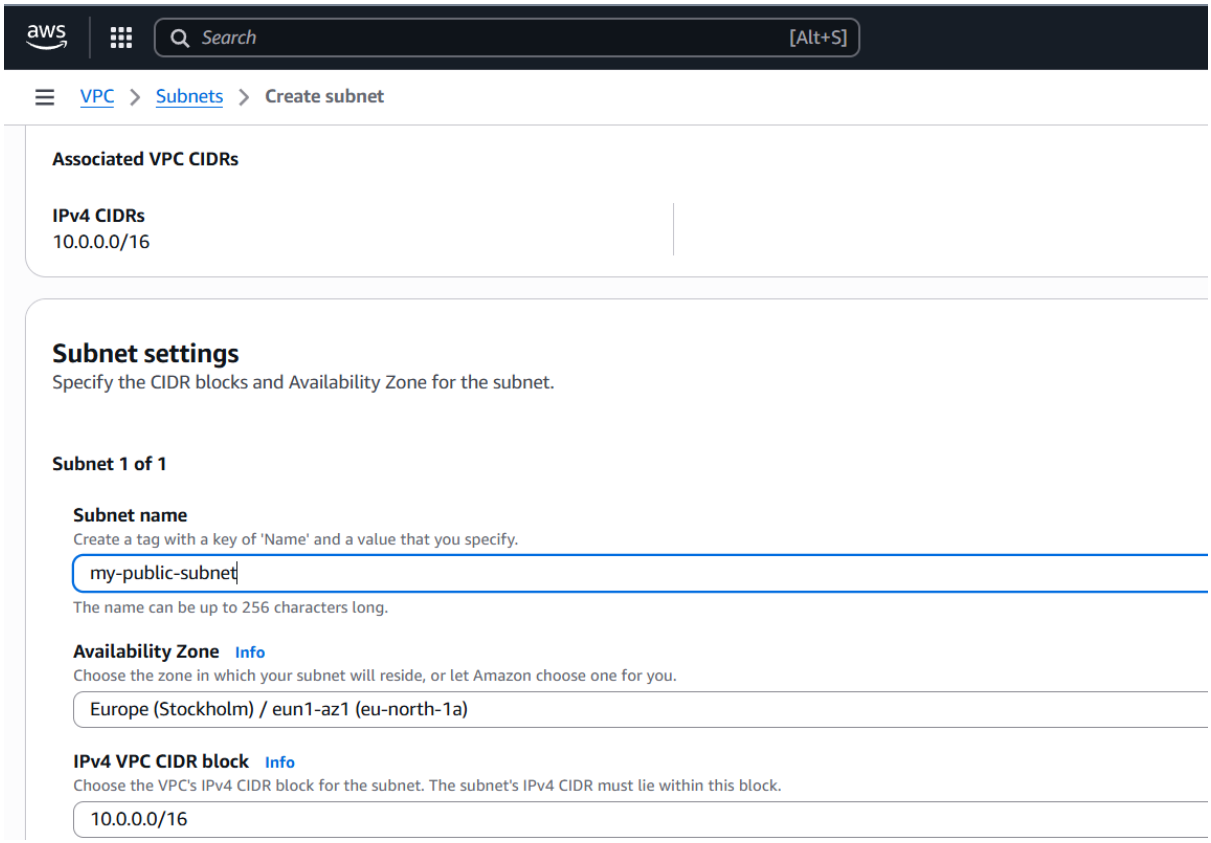
1. **Create one VPC, with 1 public subnet and 1 private subnet.**

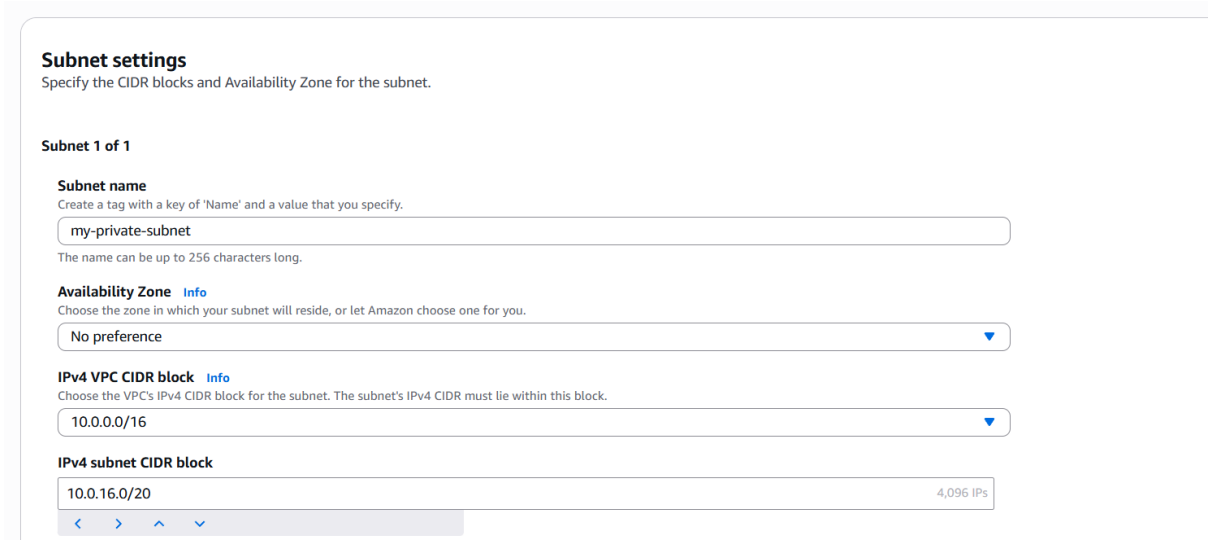
Create a vpc with the name my-vpc.



Create a public subnet .

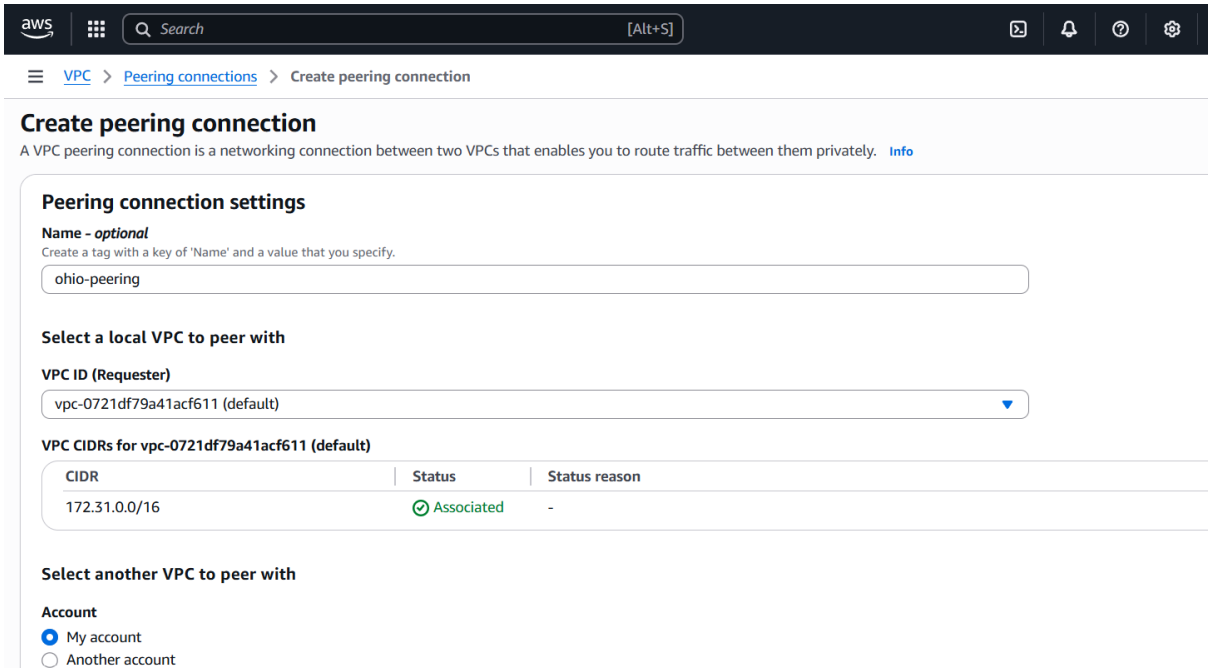


Create a private subnet.

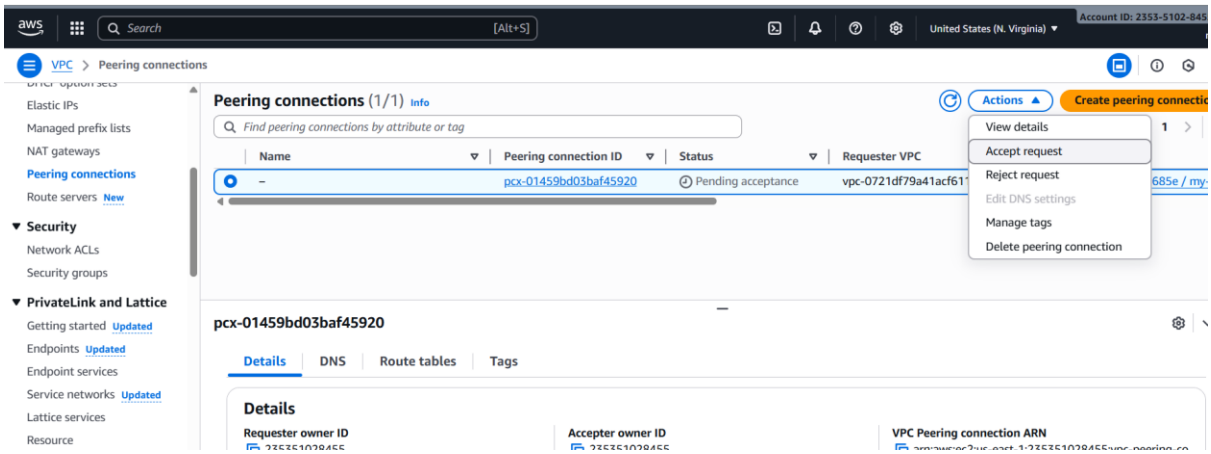


2. Enable VPC peering for cross-region.

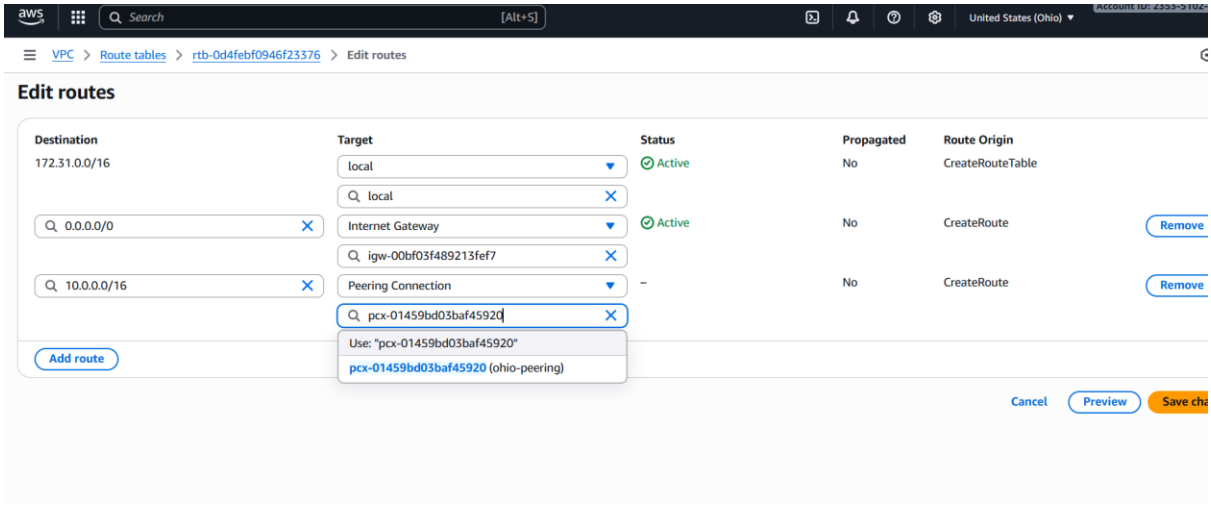
Click on peering connetions and give the details and give cross-region and vpc id of accepter and create peering.



Accept the peering request in the another region.



Go to routetable and edit with another region's cidr range.do same process in the another region also.

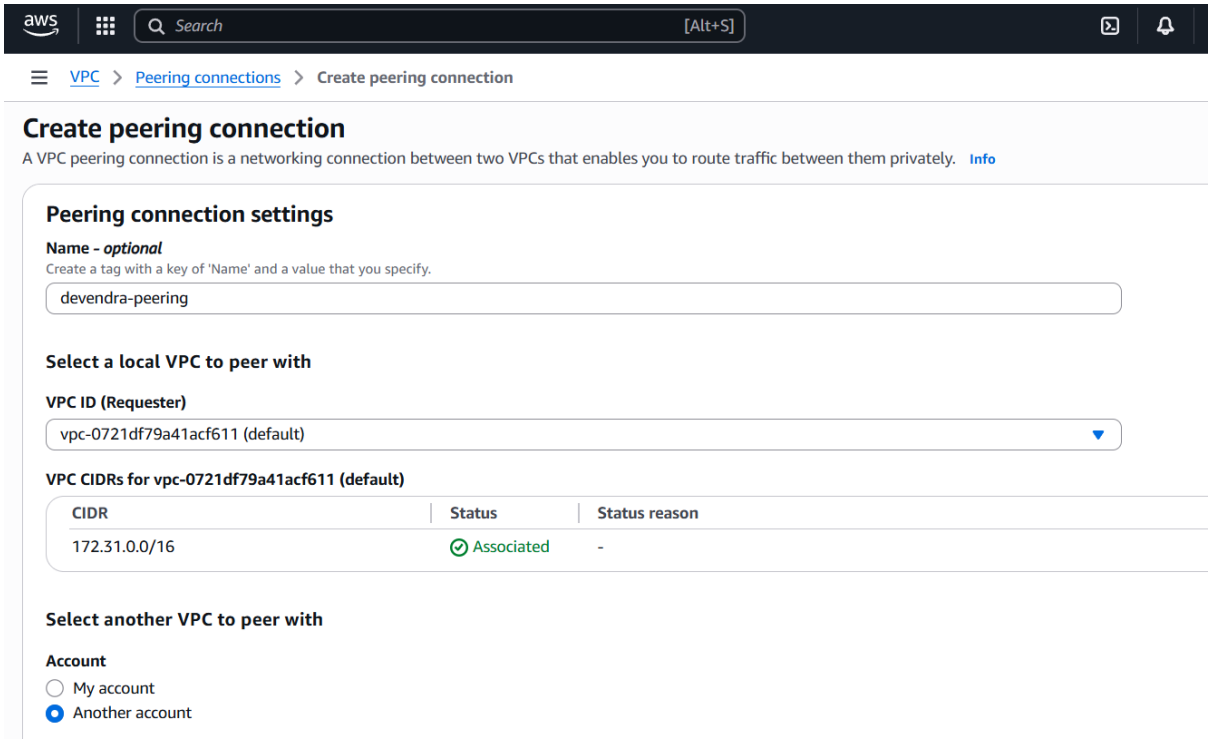


Connect with public ip and ping another region's private ip to check the connection.

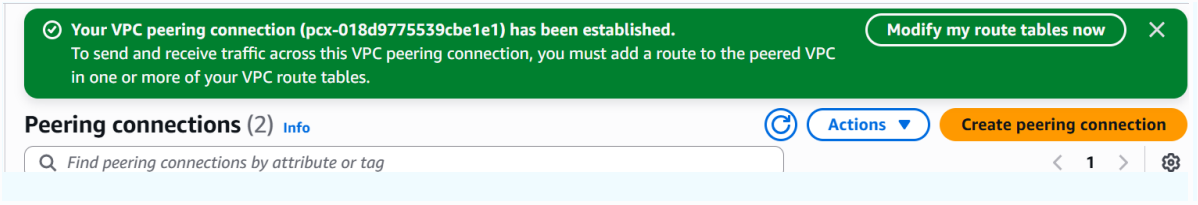
```
[root@ip-172-31-32-120 ~]# ping 10.0.15.89
PING 10.0.15.89 (10.0.15.89) 56(84) bytes of data.
64 bytes from 10.0.15.89: icmp_seq=416 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=417 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=418 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=419 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=420 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=421 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=422 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=423 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=424 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=425 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=426 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=427 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=428 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=429 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=430 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=431 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=432 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=433 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=434 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=435 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=436 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=437 ttl=127 time=13.7 ms
64 bytes from 10.0.15.89: icmp_seq=438 ttl=127 time=13.7 ms
```

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

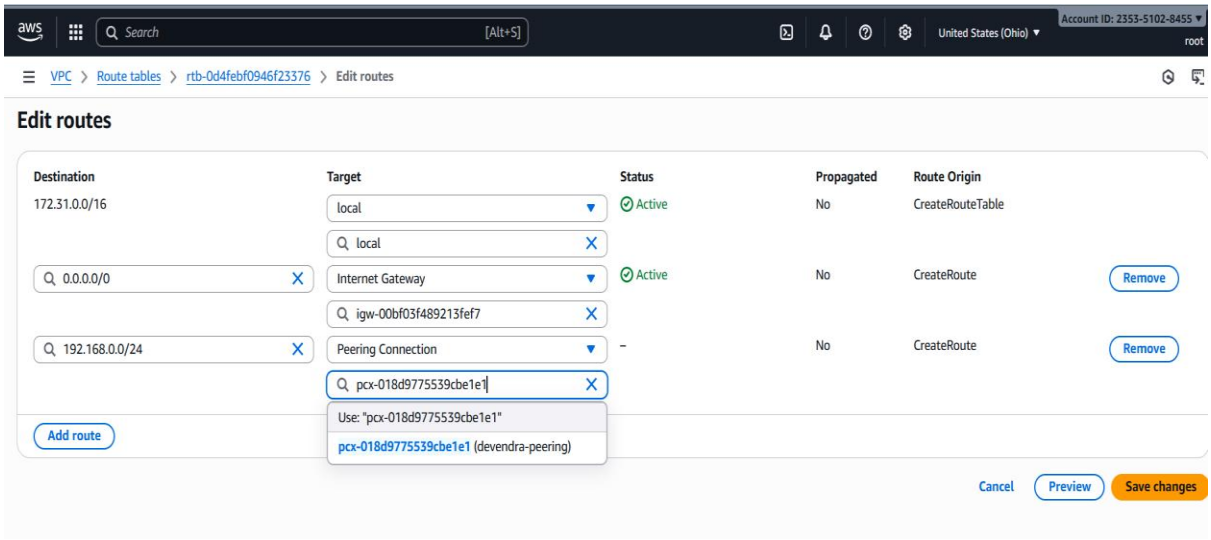
Create peering connection and give details of another region of different account.



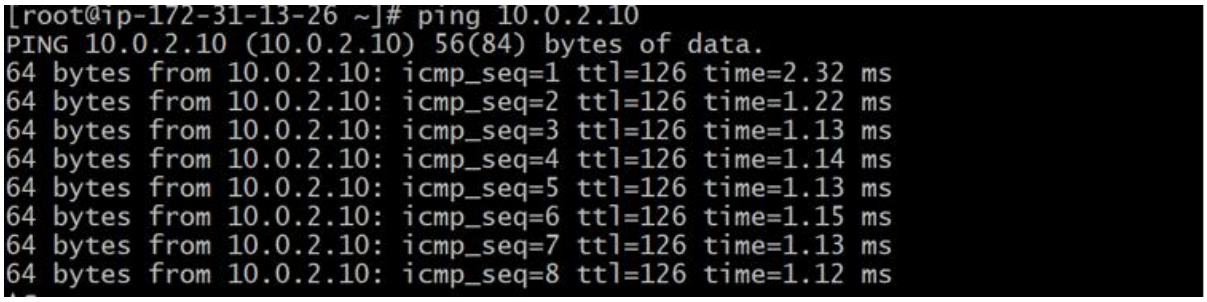
Accept the request from recivers account.



Add routes in sender’s account and reciver’s account.

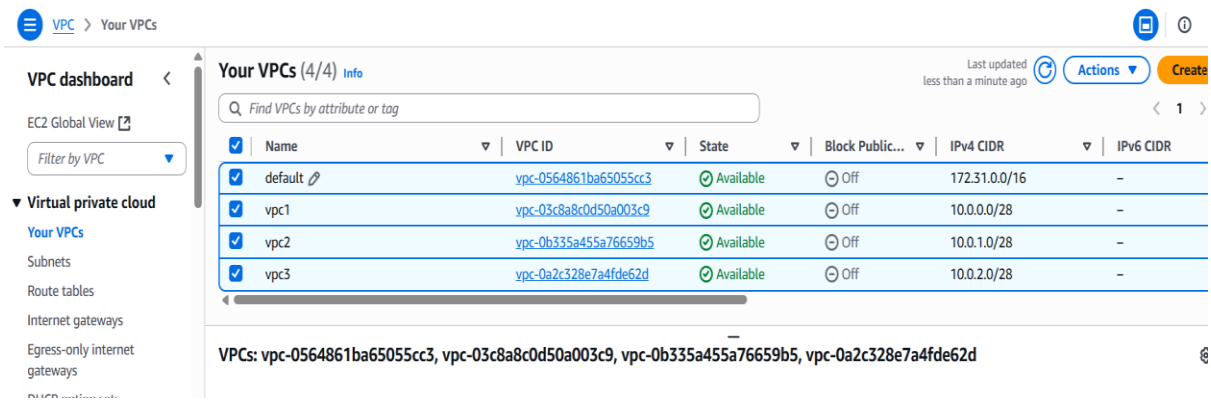


login with your public ip and ping<reciver's private I'd> to check the connectivity.

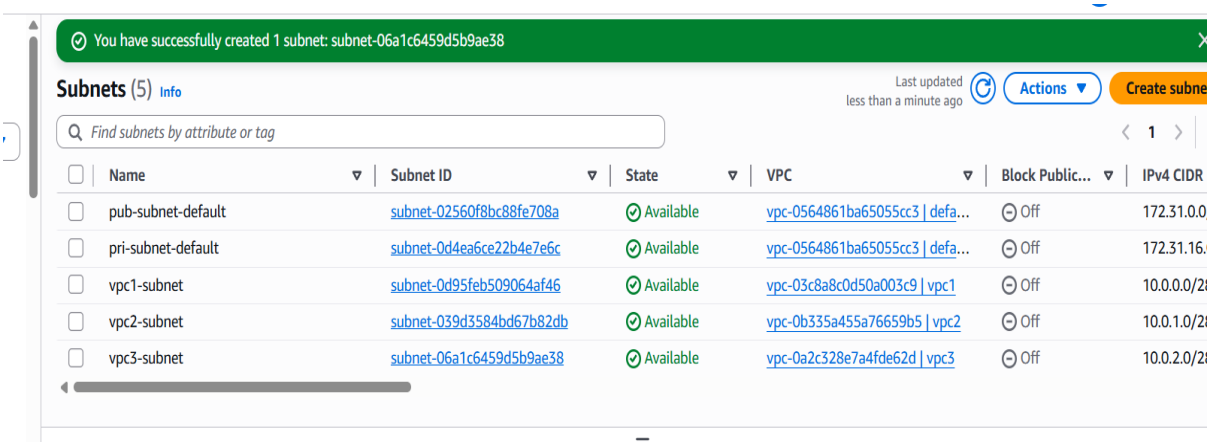


4. Set up a VPC Transit Gateway.

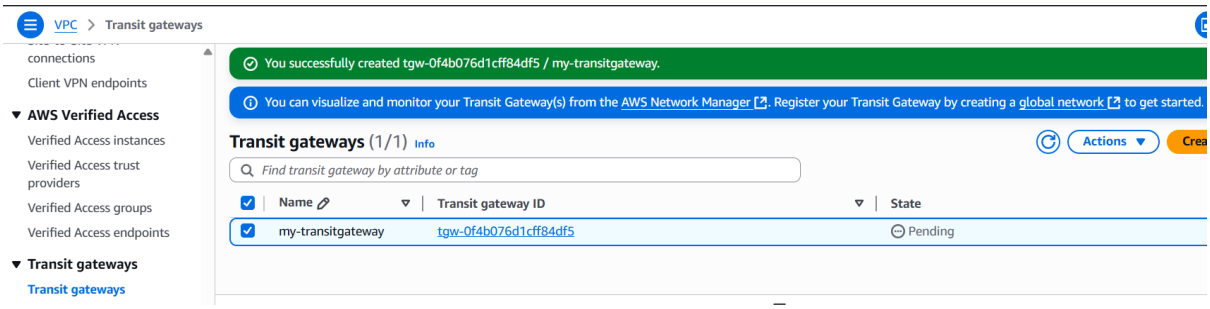
Create 4 vpc's.



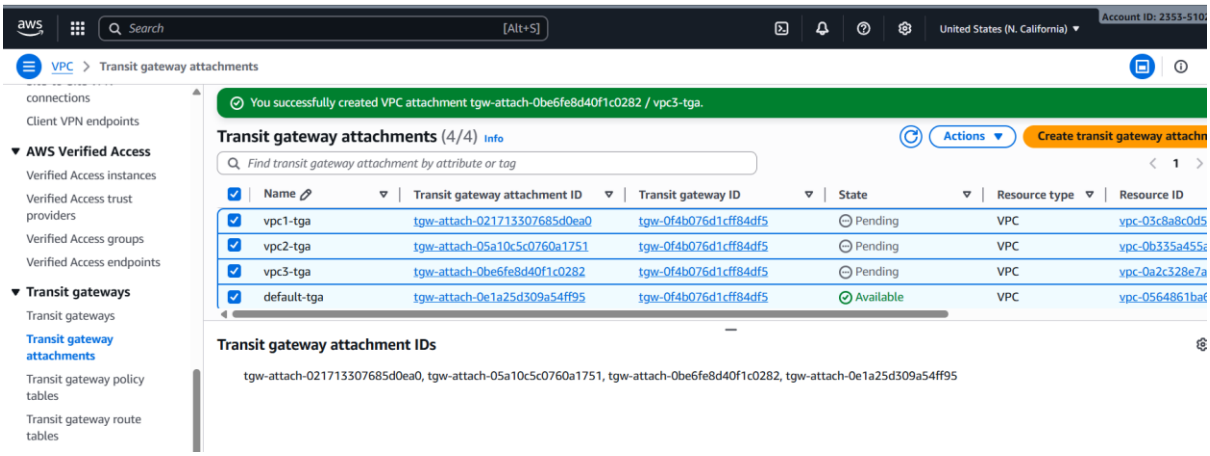
Create subnets to the vpc's.



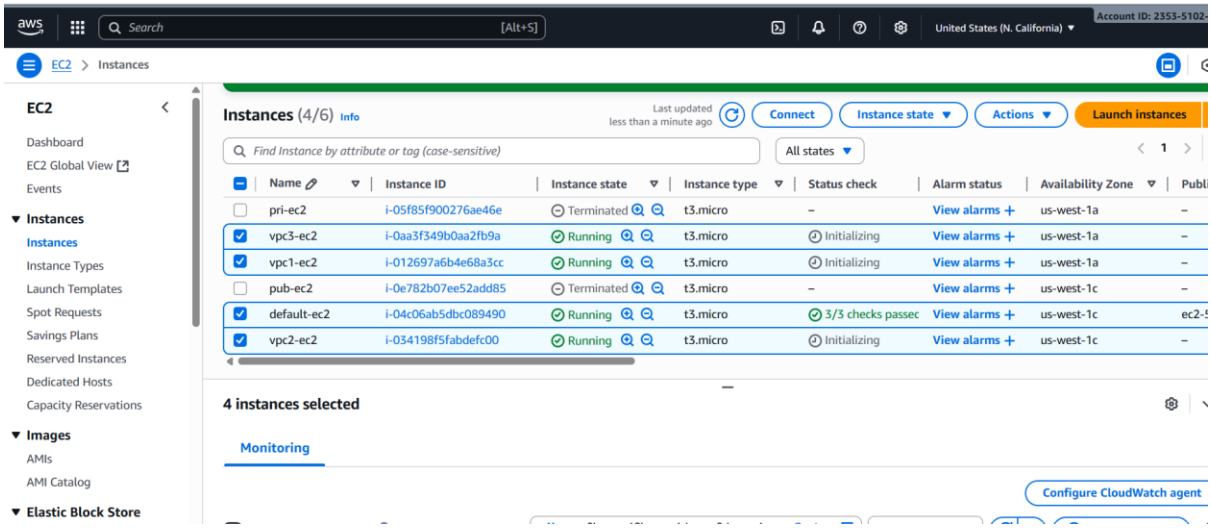
Create a transit gateway.



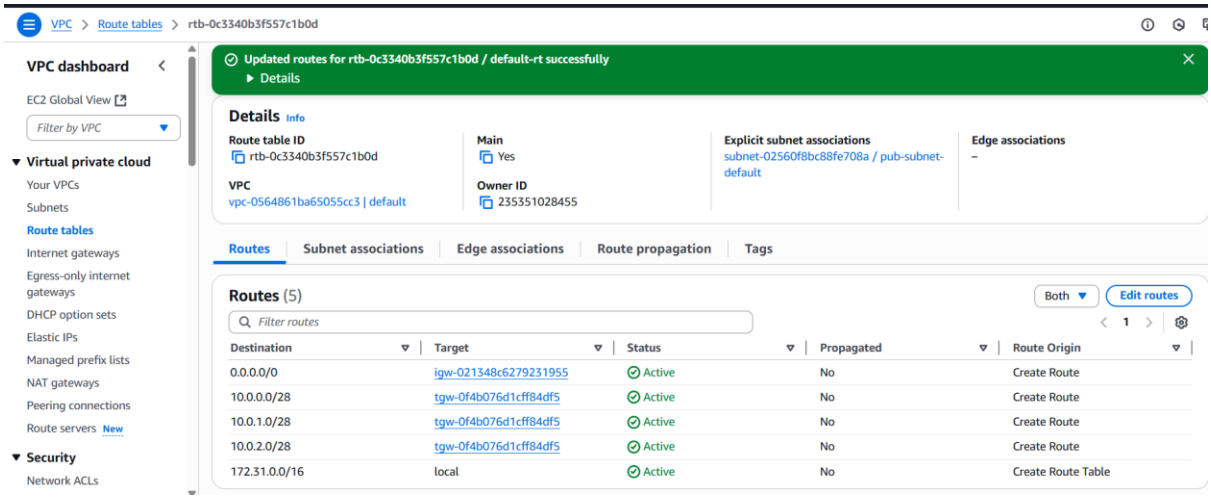
Create 4 transit gateway attachments and attach to 4 vpc’s.



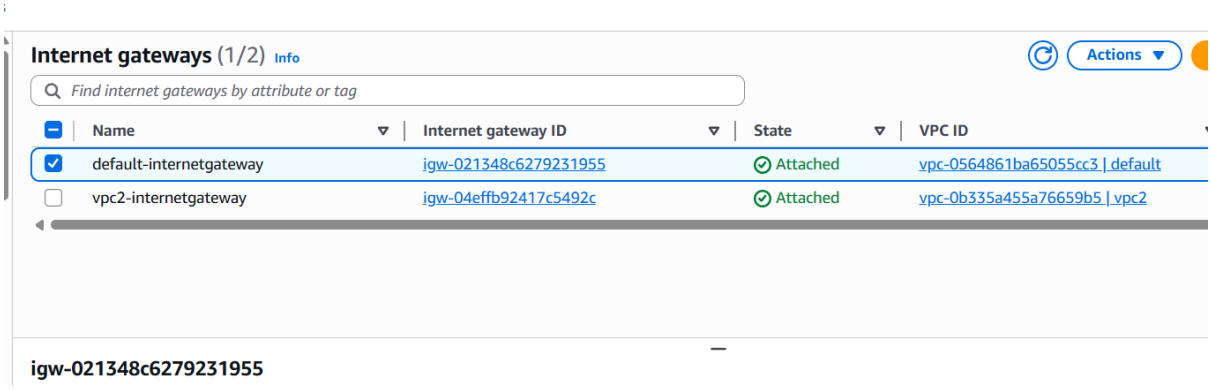
Create 4 instances for 4 vpc’s.



Go to route table and edit the routes give permissions for all vpc’s.



We need to create internet gateways for 4 vpc’s.



Login with public ip address and ping any instance private ip address.

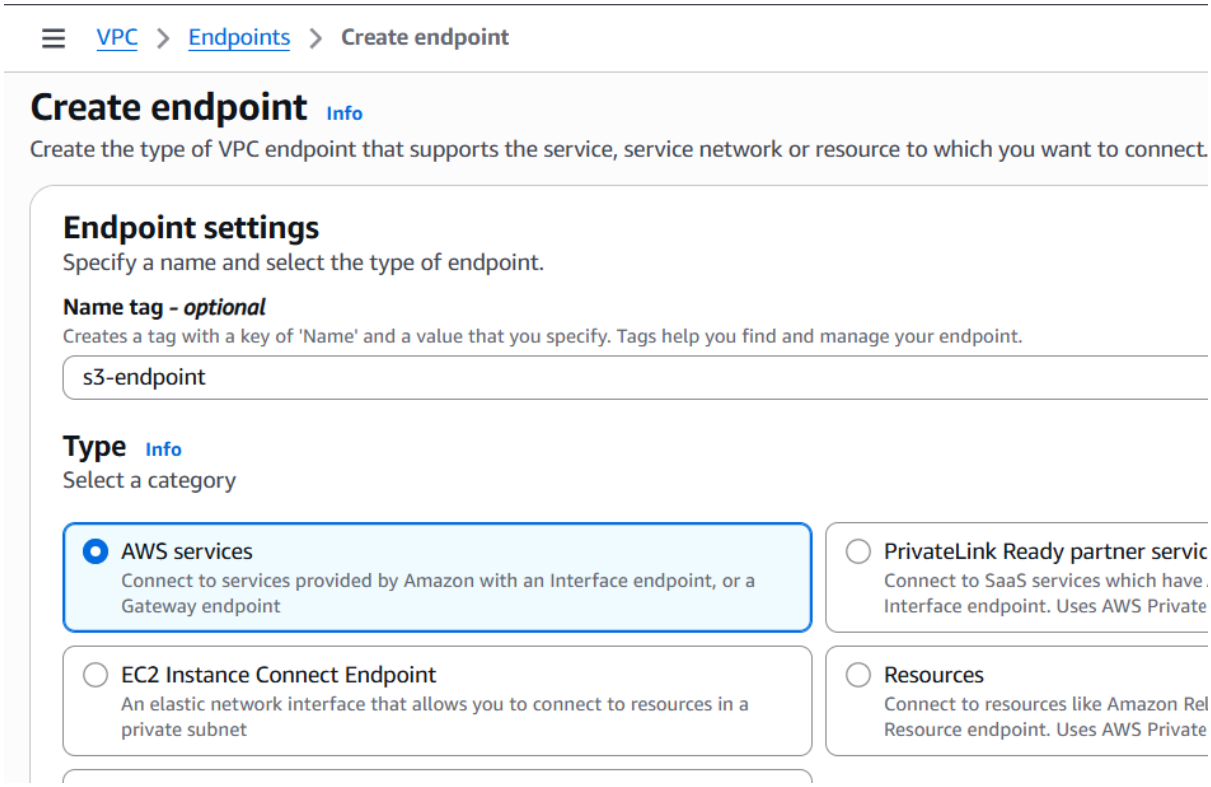
```
[root@ip-172-31-13-26 ~]# ping 10.0.2.10
PING 10.0.2.10 (10.0.2.10) 56(84) bytes of data.
64 bytes from 10.0.2.10: icmp_seq=1 ttl=126 time=2.32 ms
64 bytes from 10.0.2.10: icmp_seq=2 ttl=126 time=1.22 ms
64 bytes from 10.0.2.10: icmp_seq=3 ttl=126 time=1.13 ms
64 bytes from 10.0.2.10: icmp_seq=4 ttl=126 time=1.14 ms
64 bytes from 10.0.2.10: icmp_seq=5 ttl=126 time=1.13 ms
64 bytes from 10.0.2.10: icmp_seq=6 ttl=126 time=1.15 ms
64 bytes from 10.0.2.10: icmp_seq=7 ttl=126 time=1.13 ms
64 bytes from 10.0.2.10: icmp_seq=8 ttl=126 time=1.12 ms
^C
--- 10.0.2.10 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7009ms
rtt min/avg/max/mdev = 1.121/1.292/2.321/0.389 ms
[root@ip-172-31-13-26 ~]# ping 10.0.0.6
PING 10.0.0.6 (10.0.0.6) 56(84) bytes of data.
64 bytes from 10.0.0.6: icmp_seq=1 ttl=126 time=1.93 ms
64 bytes from 10.0.0.6: icmp_seq=2 ttl=126 time=1.08 ms
64 bytes from 10.0.0.6: icmp_seq=3 ttl=126 time=1.08 ms
64 bytes from 10.0.0.6: icmp_seq=4 ttl=126 time=1.09 ms
64 bytes from 10.0.0.6: icmp_seq=5 ttl=126 time=1.05 ms
64 bytes from 10.0.0.6: icmp_seq=6 ttl=126 time=1.07 ms
^C
--- 10.0.0.6 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 1.048/1.217/1.931/0.319 ms
```

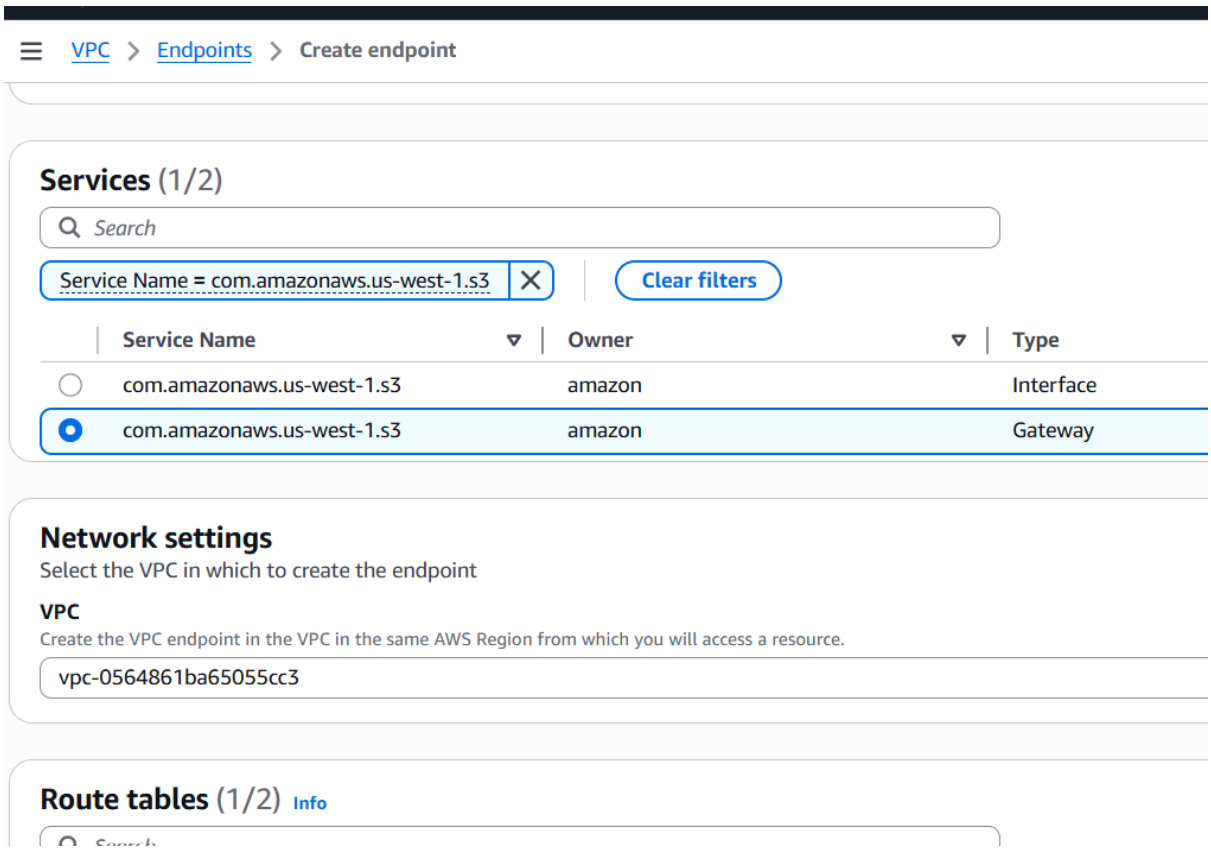
5. Set up a VPC Endpoint.

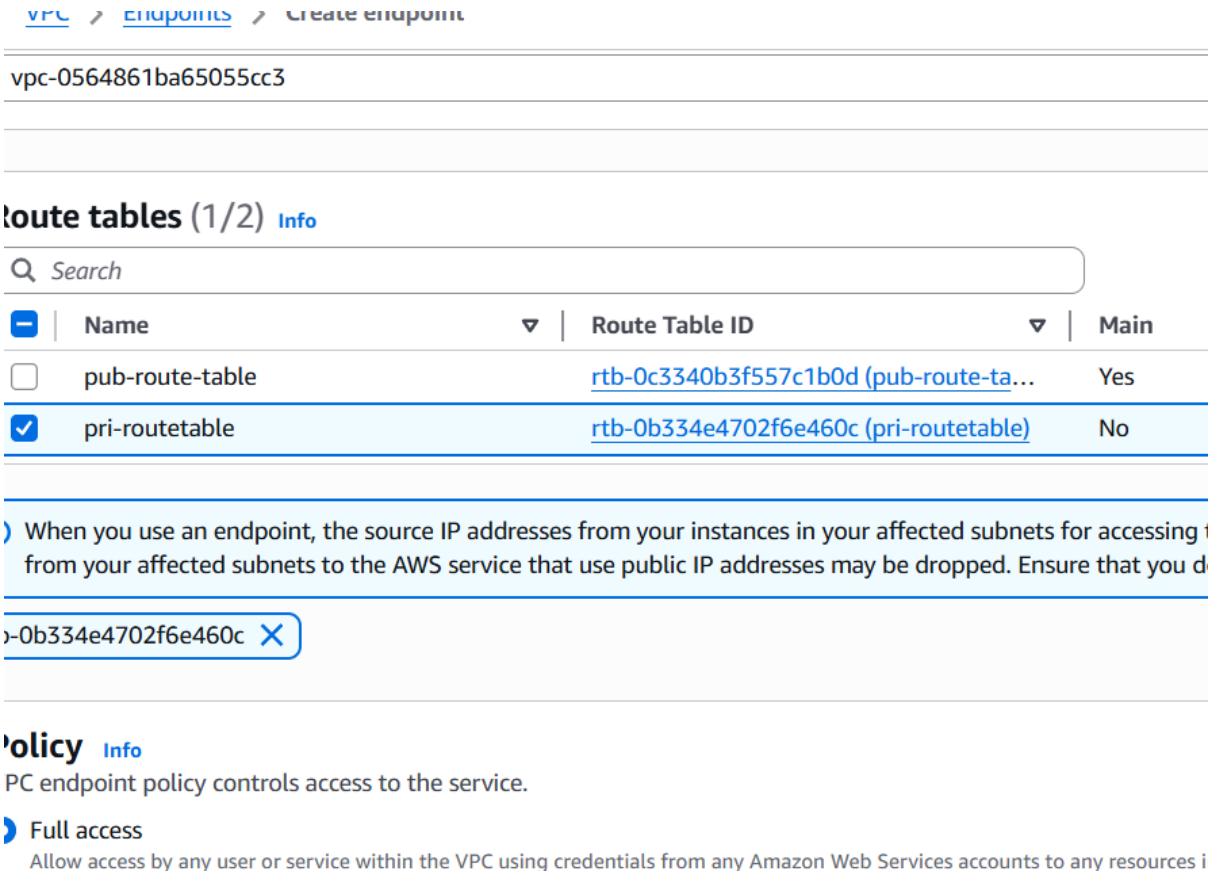
Open ec2 instance with public ip address.

```
MUJJU SK@DESKTOP-LU541U4 MINGW64 ~/Downloads
$ ssh -i california.pem ec2-user@54.183.26.154
The authenticity of host '54.183.26.154 (54.183.26.154)' can't be established.
ED25519 key fingerprint is SHA256:zLM1ROVXJkpgA5yITcCKbmqX5QeERCIiEu1lD3ZkVZk.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.183.26.154' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-6-12 ~]$ sudo su-
sudo: su-: command not found
[ec2-user@ip-172-31-6-12 ~]$ sudo su -
[root@ip-172-31-6-12 ~]# aws --version
aws-cli/2.27.57 Python/3.9.23 Linux/6.1.150-174.273.amzn2023.x86_64 source/x86_64.amzn.2023
[root@ip-172-31-6-12 ~]#
```

Create an end point for public instance.


VPC > Endpoints > Create endpoint
Create endpoint Info
Create the type of VPC endpoint that supports the service, service network or resource to which you want to connect.
Endpoint settings
Specify a name and select the type of endpoint.
Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify. Tags help you find and manage your endpoint.
s3-endpoint
Type Info
Select a category
AWS services
Connect to services provided by Amazon with an Interface endpoint, or a Gateway endpoint
PrivateLink Ready partner servic
Connect to SaaS services which have
Interface endpoint. Uses AWS Private
EC2 Instance Connect Endpoint
An elastic network interface that allows you to connect to resources in a private subnet
Resources
Connect to resources like Amazon Rel
Resource endpoint. Uses AWS Private


VPC > Endpoints > Create endpoint
Services (1/2)
Search
Service Name = com.amazonaws.us-west-1.s3
Clear filters
Service Name
Owner
Type
com.amazonaws.us-west-1.s3
amazon
Interface
com.amazonaws.us-west-1.s3
amazon
Gateway
Network settings
Select the VPC in which to create the endpoint
VPC
Create the VPC endpoint in the VPC in the same AWS Region from which you will access a resource.
vpc-0564861ba65055cc3
Route tables (1/2) Info



We can see endpoint is connected to bucket.