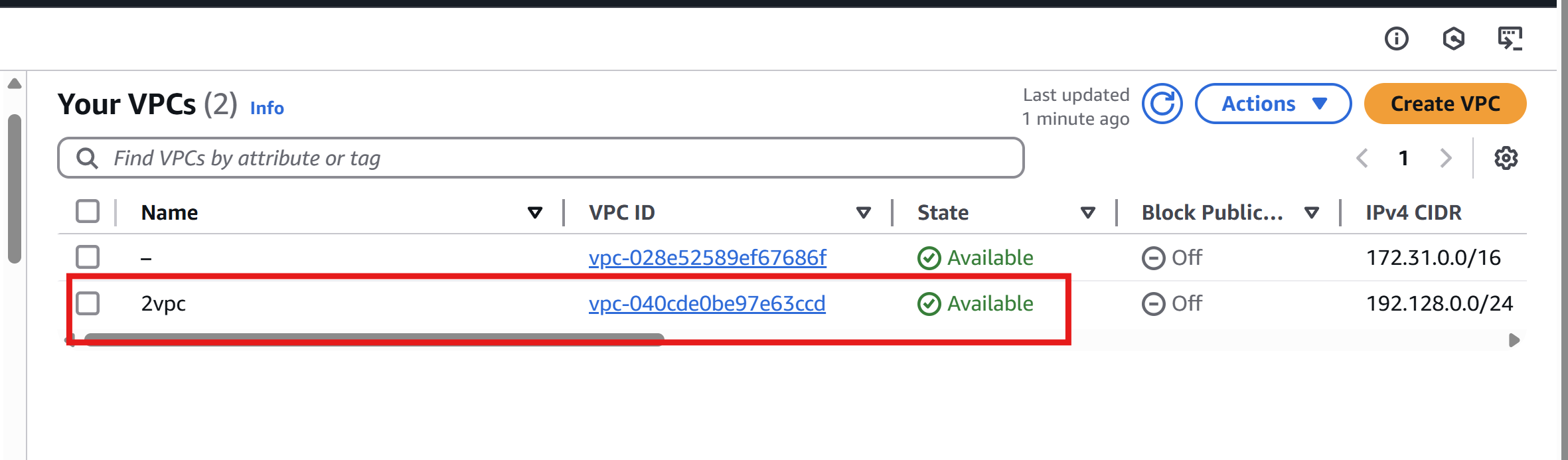
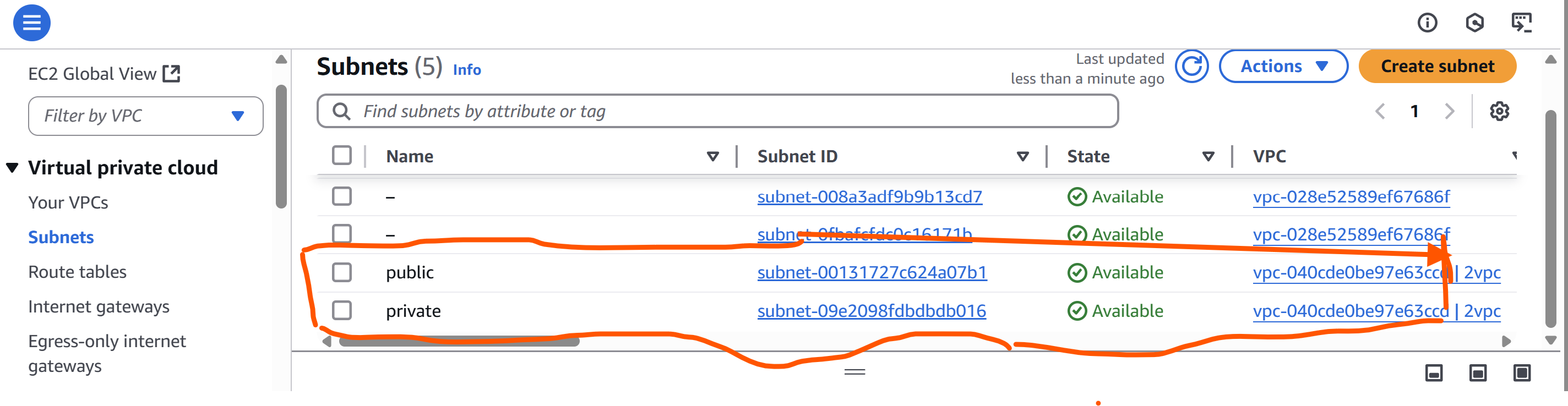
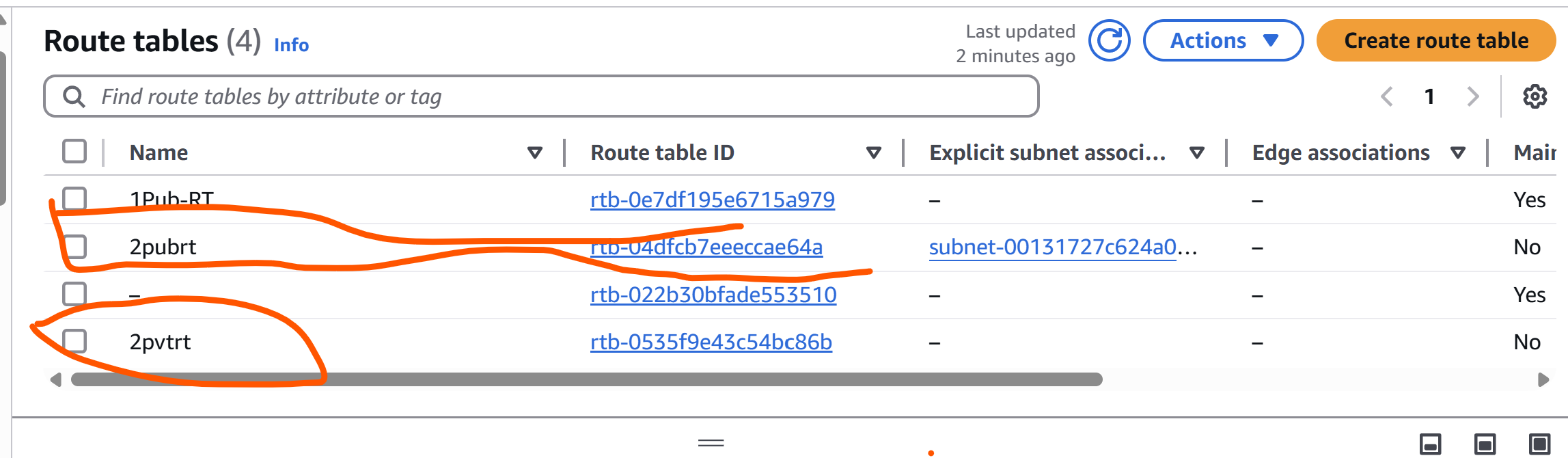
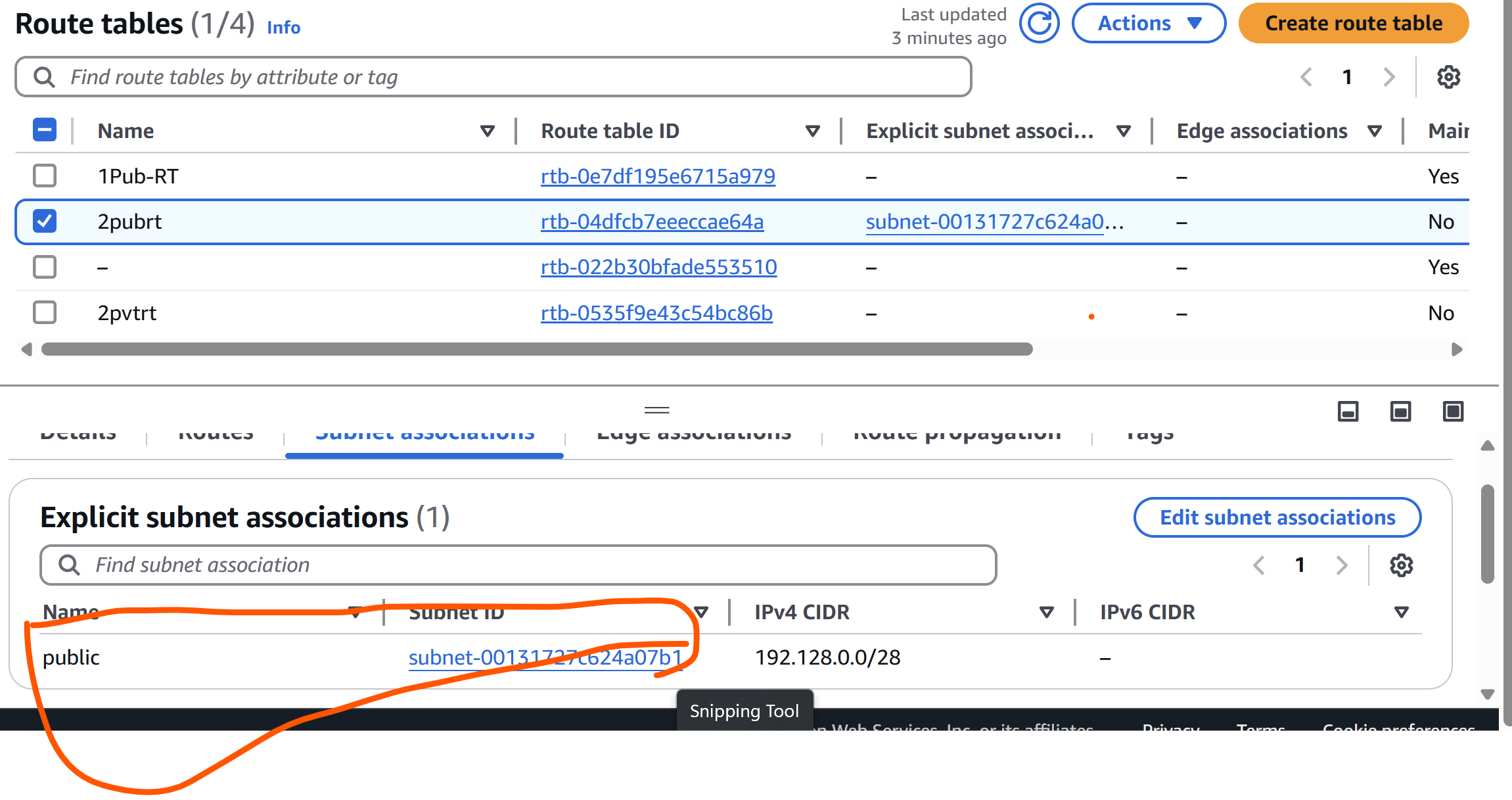
VPC Task-2

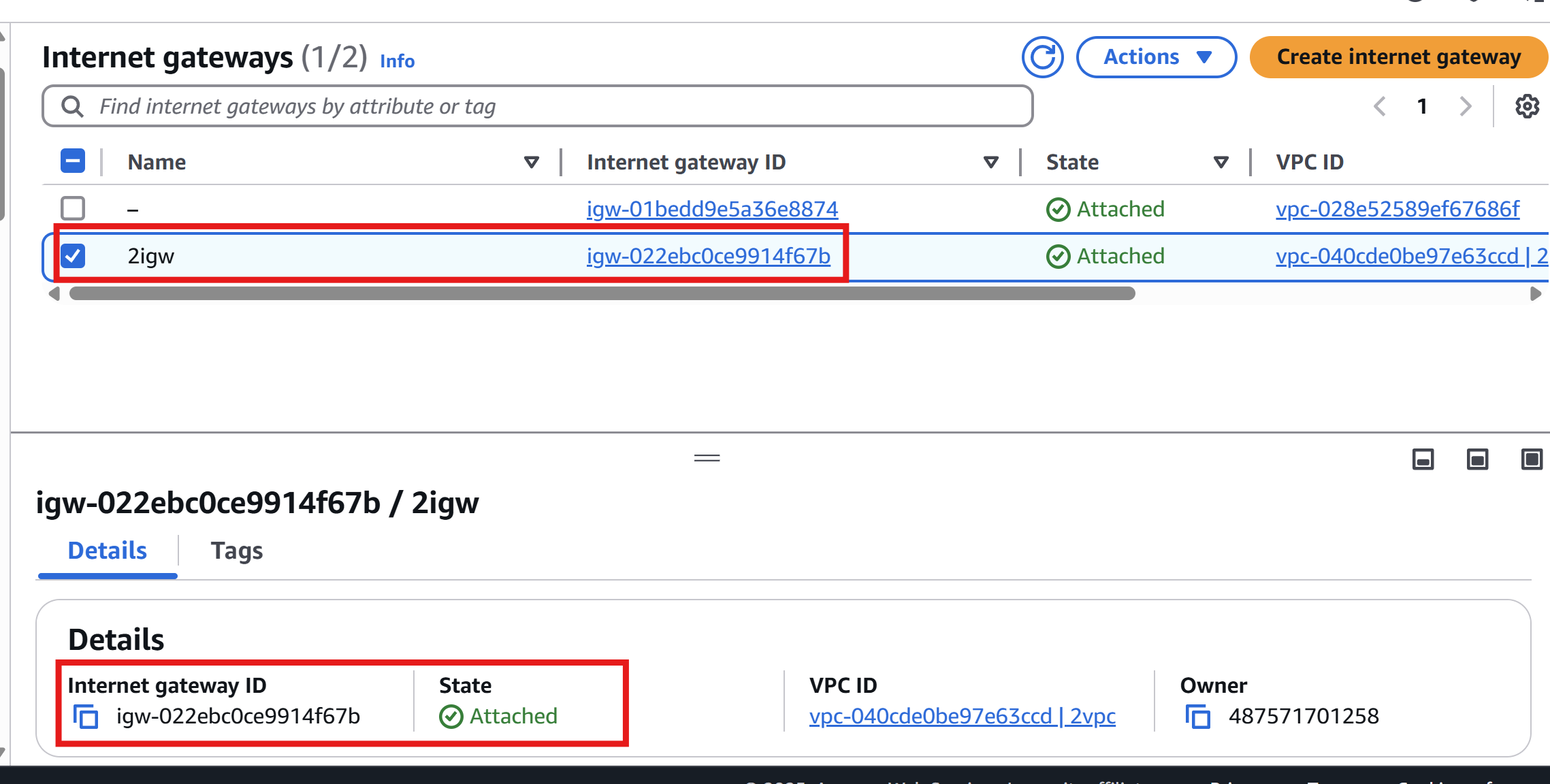
1) Create one VPC,with 1 one public subnet and 1 private subnet.











2) Enable VPC peering for cross region.

Step 1: Create VPC Peering Connection

* Go to **VPC Dashboard** in **Region A (us-north-1)**.
* In the left menu, click **"Peering Connections"** → Click **"Create Peering Connection"**.
* Fill in the details:
* **Name tag**: cross-region-peer
* **VPC Requester**: Select **VPC-A**
* **Account**: My account (assuming same AWS account)
* **Region**: Select **us-east-2**
* **VPC Accepter**: Select **VPC-B** from dropdown
* Click **Create Peering Connection**

Step 2: Accept the Peering Request

1. Switch to Region B (**us-east-2**).
2. Go to **VPC Dashboard** → **Peering Connections**
3. Select the pending connection → Click **"Actions"** → **Accept Request**

Now the VPCs are peered, but they **cannot talk yet** without route and security settings.

Step 3: Update Route Tables

**In Region A (us-north-1):**

1. Go to **Route Tables**
2. Find the route table for **VPC-A**
3. Click **"Routes"** tab → **Edit Routes** → **Add Route**
   * **Destination**: 172.168.0.0/16 (CIDR of VPC-B)
   * **Target**: Select the **Peering Connection (pcx-xxxx)**
4. Click **Save changes**

**In Region B (us-east-2):**

1. Repeat the same steps for **VPC-B’s** route table:
   * Destination: 192.128.0.0/24
   * Target: **Peering Connection**

**Step 4: Update Security Groups**

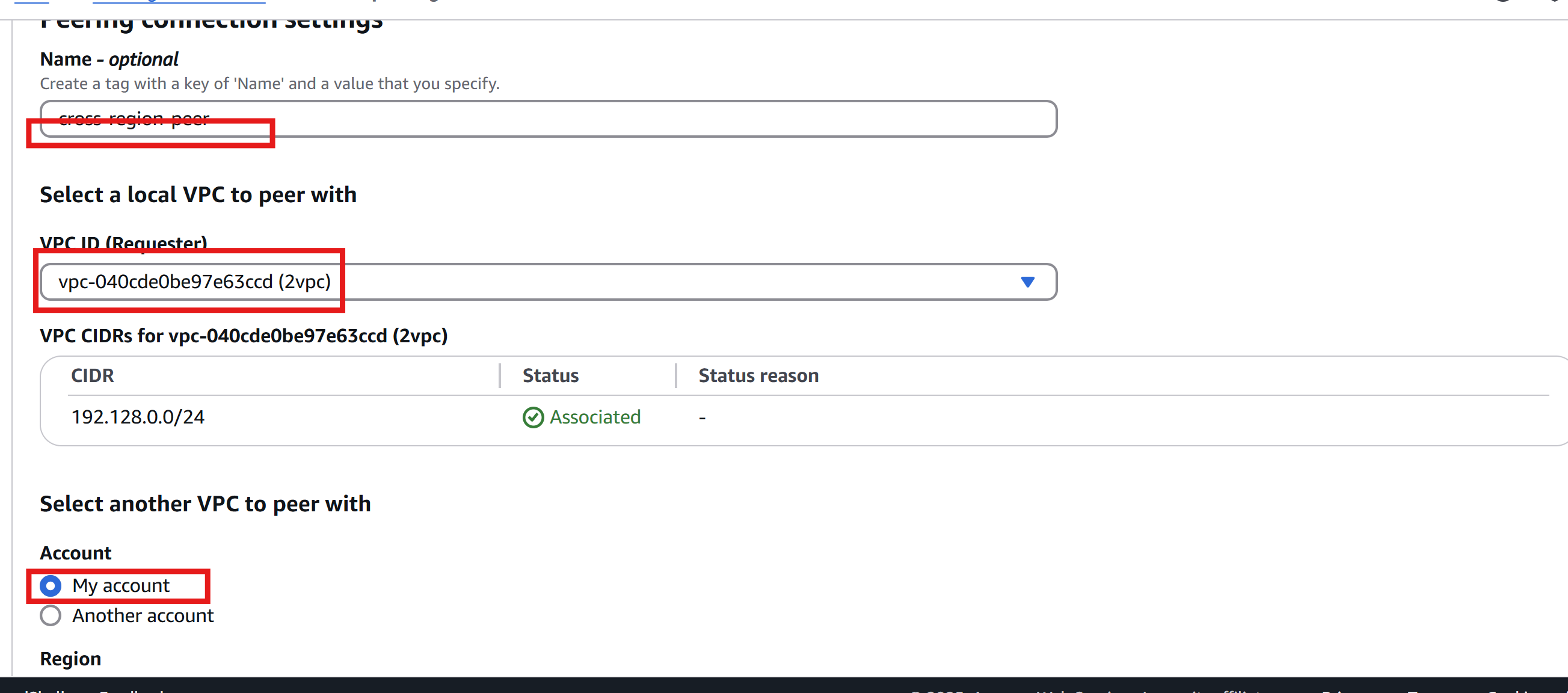
You need to allow traffic **from the other VPC’s CIDR**.

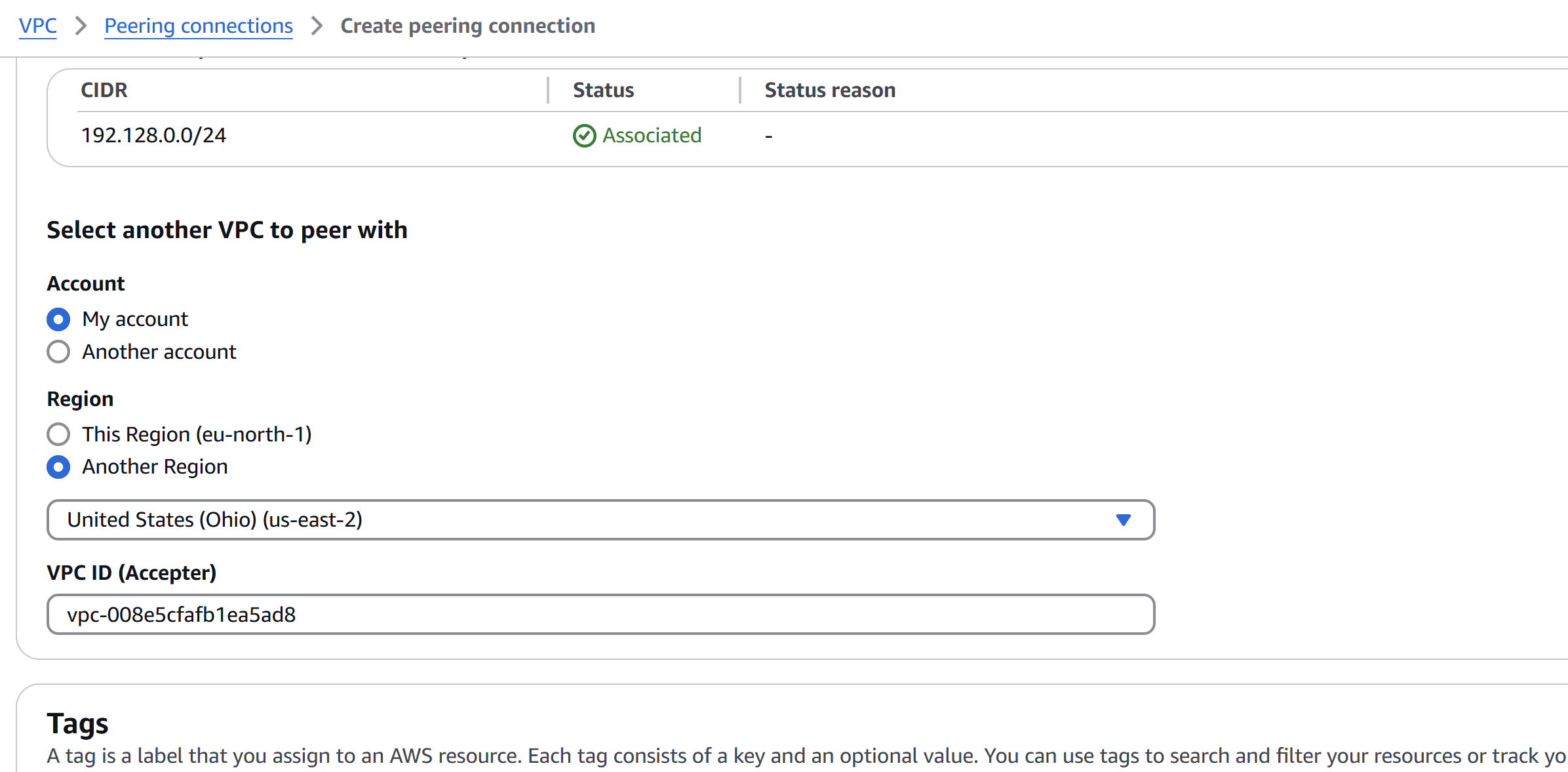
**In VPC-A:**

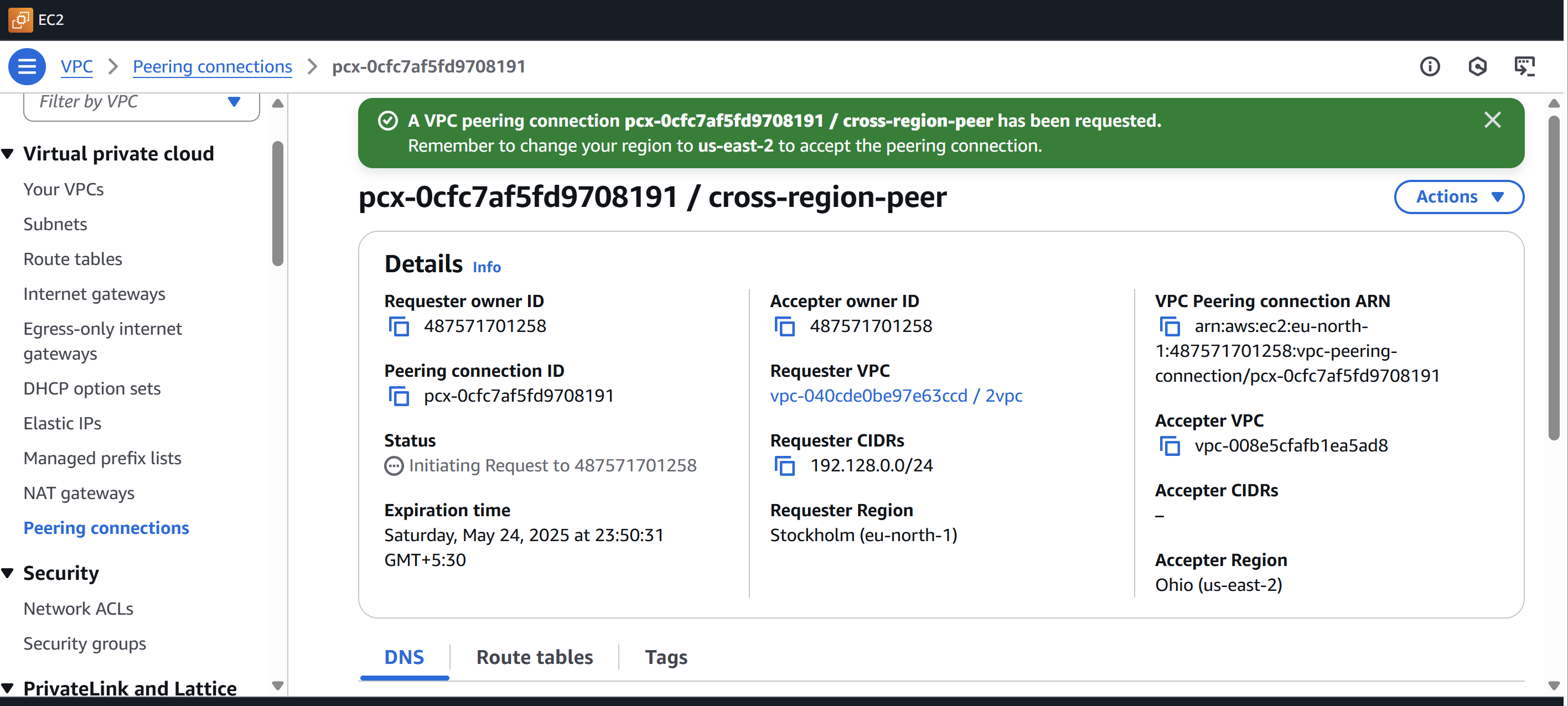
1. Go to **EC2** → **Security Groups**
2. Select the group attached to your instance
3. Click **Inbound rules** → **Edit**
4. Add rule:
   * Type: All traffic (or specific ports like SSH, HTTP)
   * Source: 172.168.0.0/16

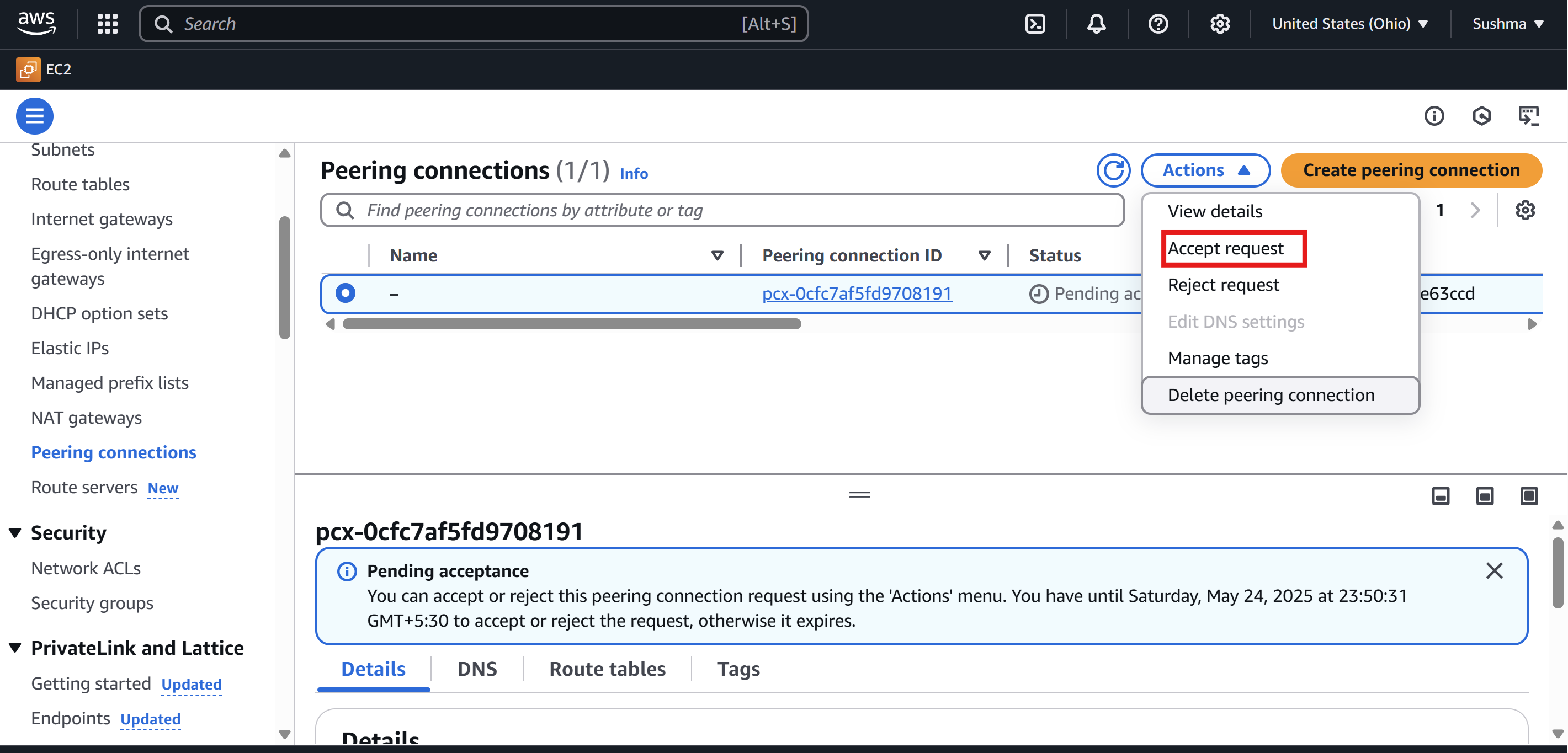
**In VPC-B:**

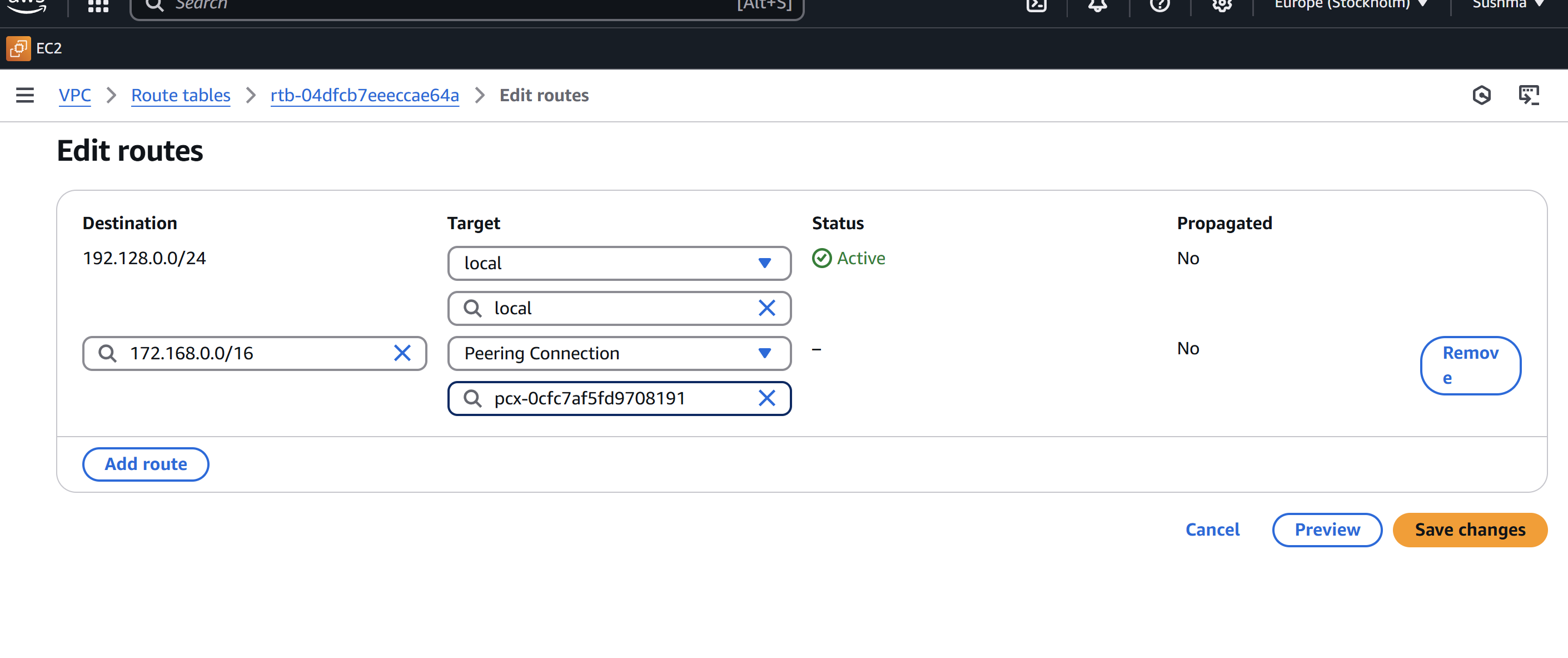
Repeat with 192.128.0.0/24 as the source.

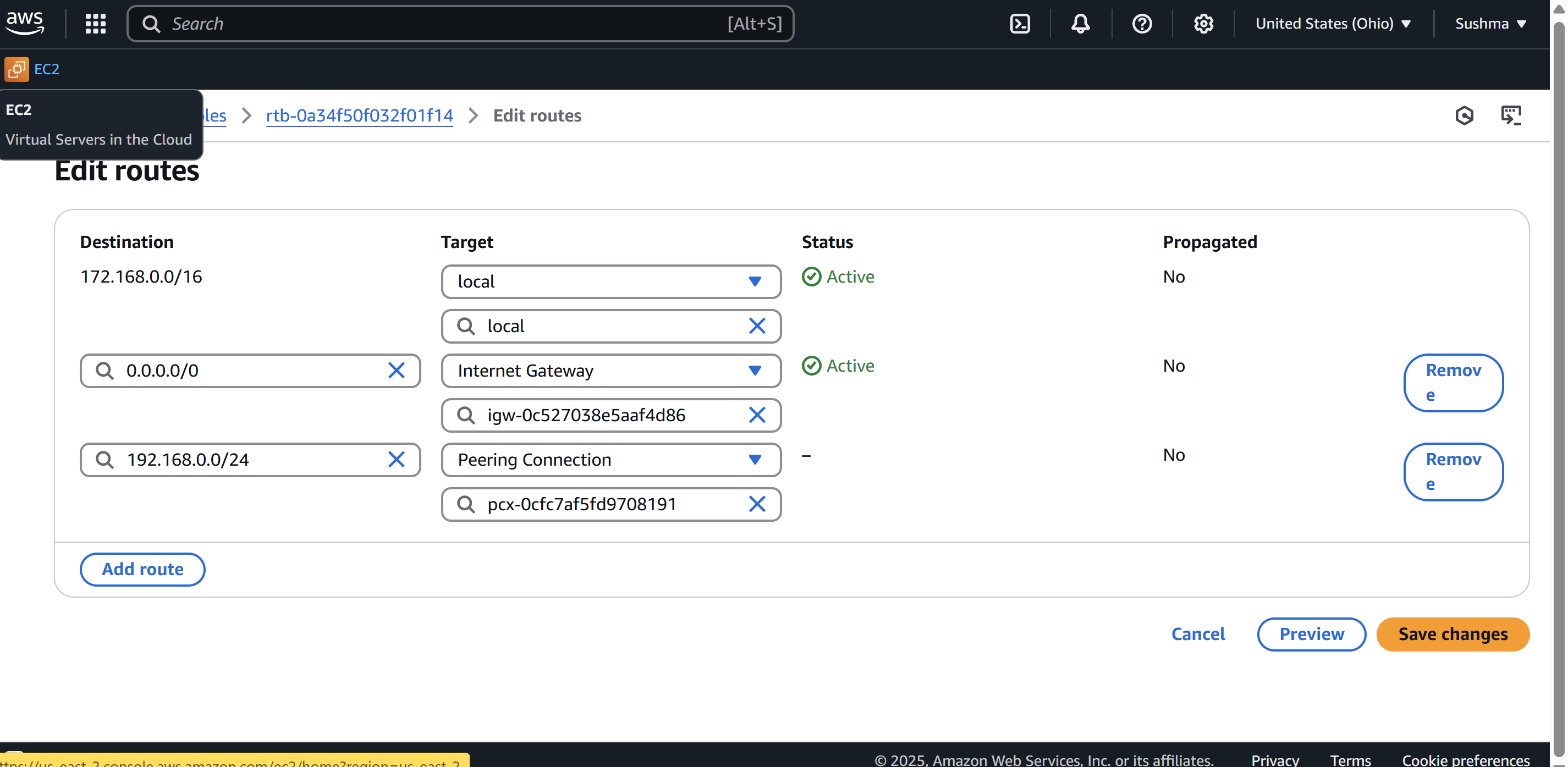


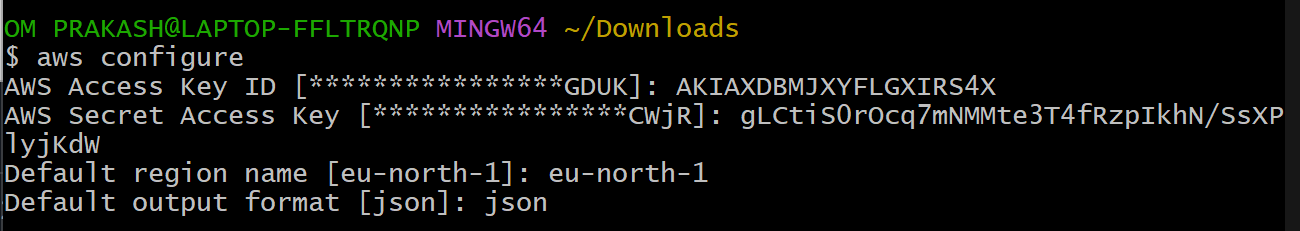


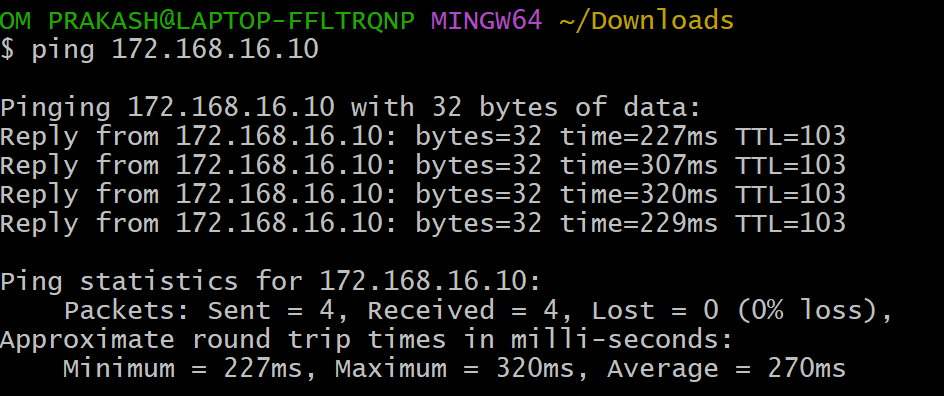






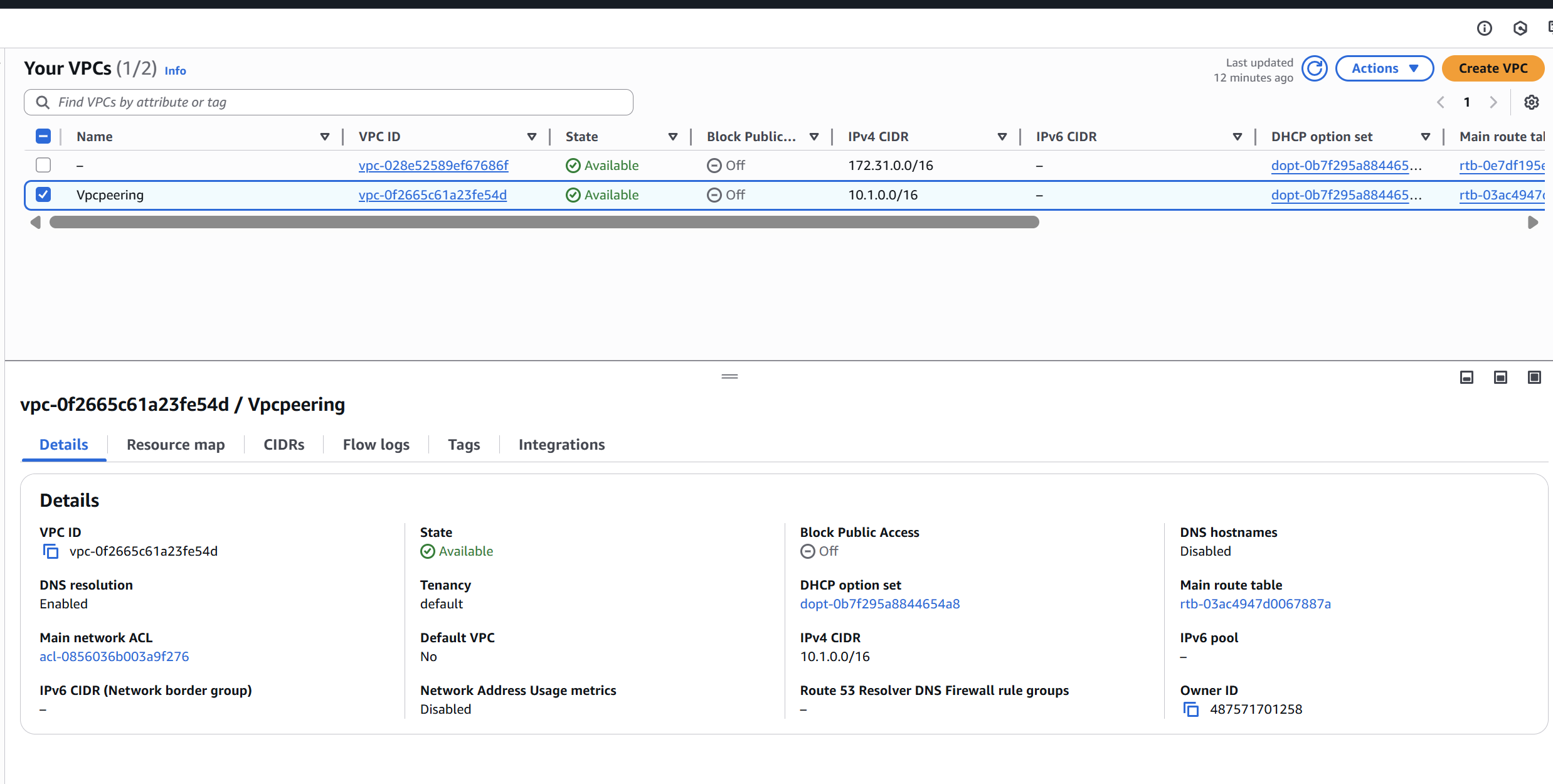




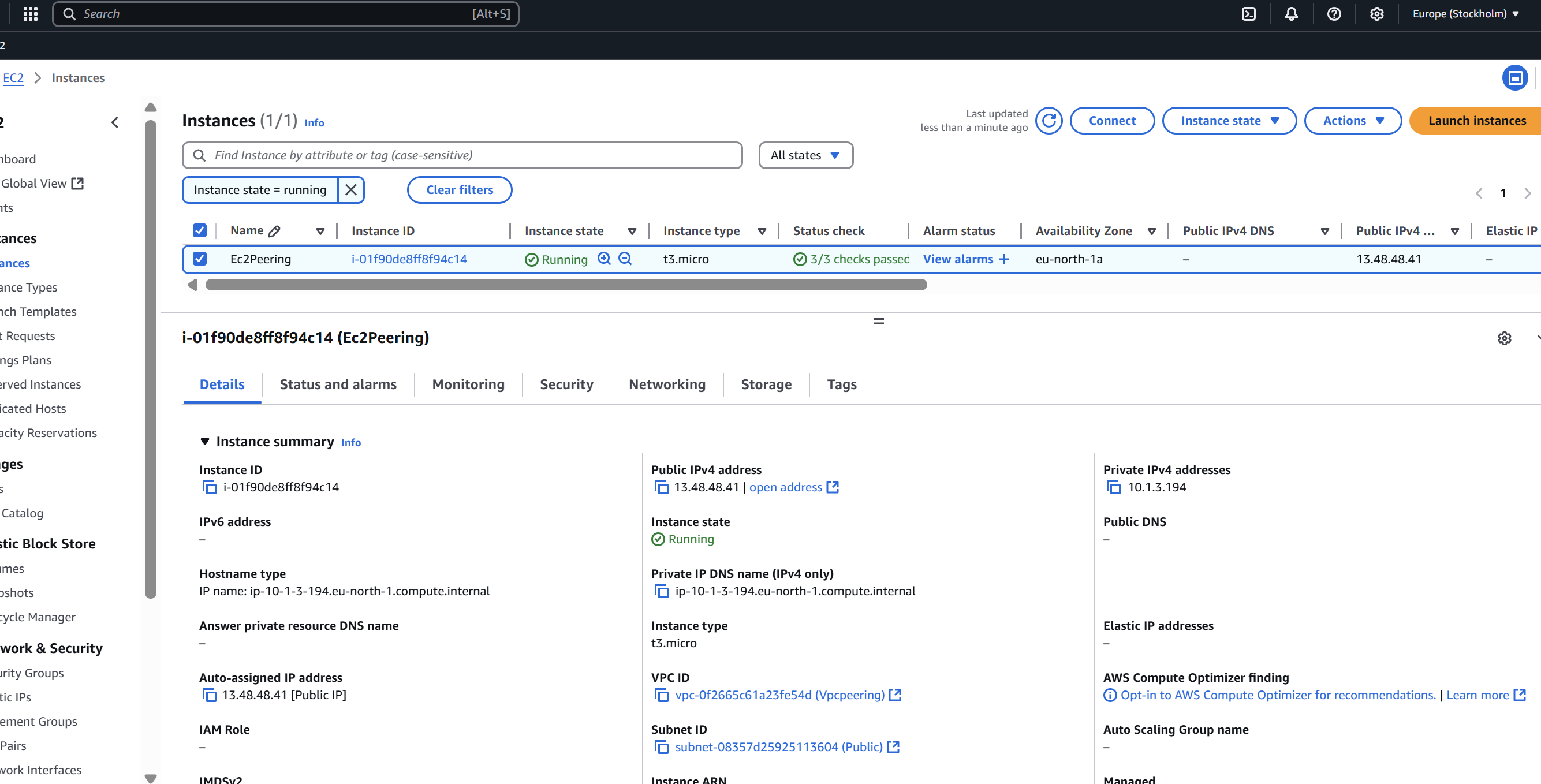


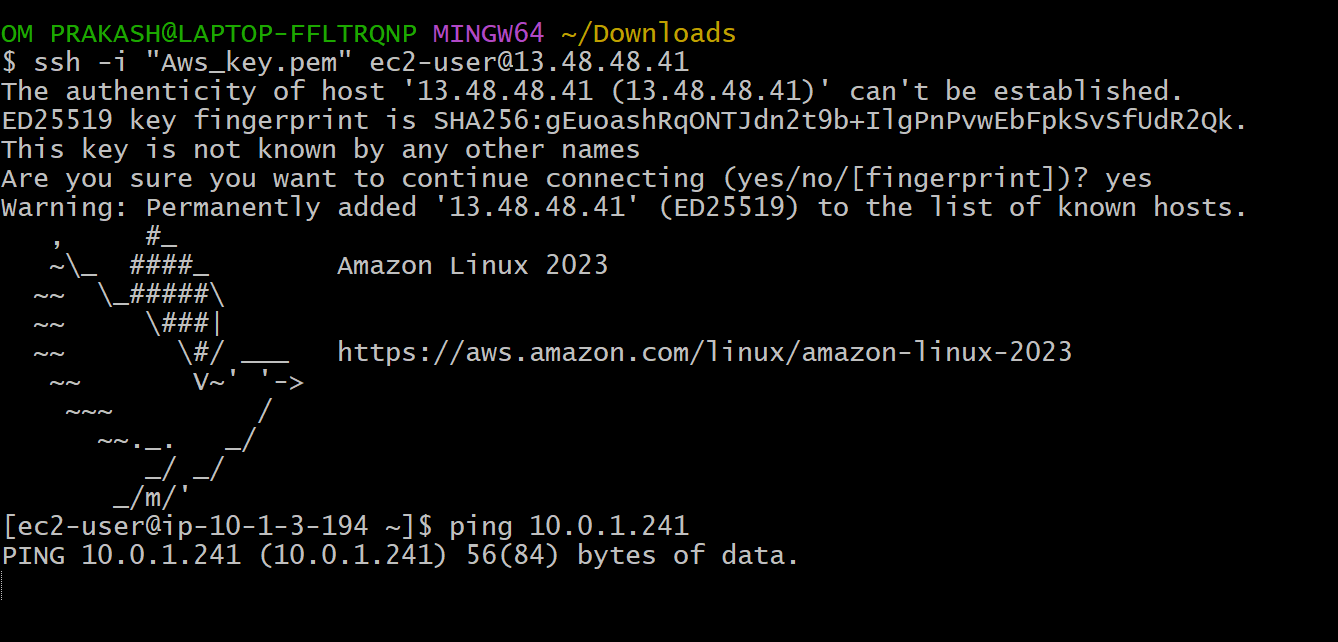
3) Enable VPC peering for cross account. (You can collaborate with your friend and do this task).

* Create one vpc with 2 subnets(private and public)with CIDR
* Attach one subnet with IGW
* Then create route table and associte with subnets (public in Pub-RT and private in private RT)
* Then set public as main
* Go to VPC peering and provide tag name ,rquester vpc ID,another account ,another account region,accepter vpc and click create
* Accepter received request and accept the request
* Then provides private Ip address of ec2 in instances
* In this account will create one ec2 instance and set security groups
* All traffic and ssh
* Go to git bash and connect with ssh and the ping <private ip of another account>





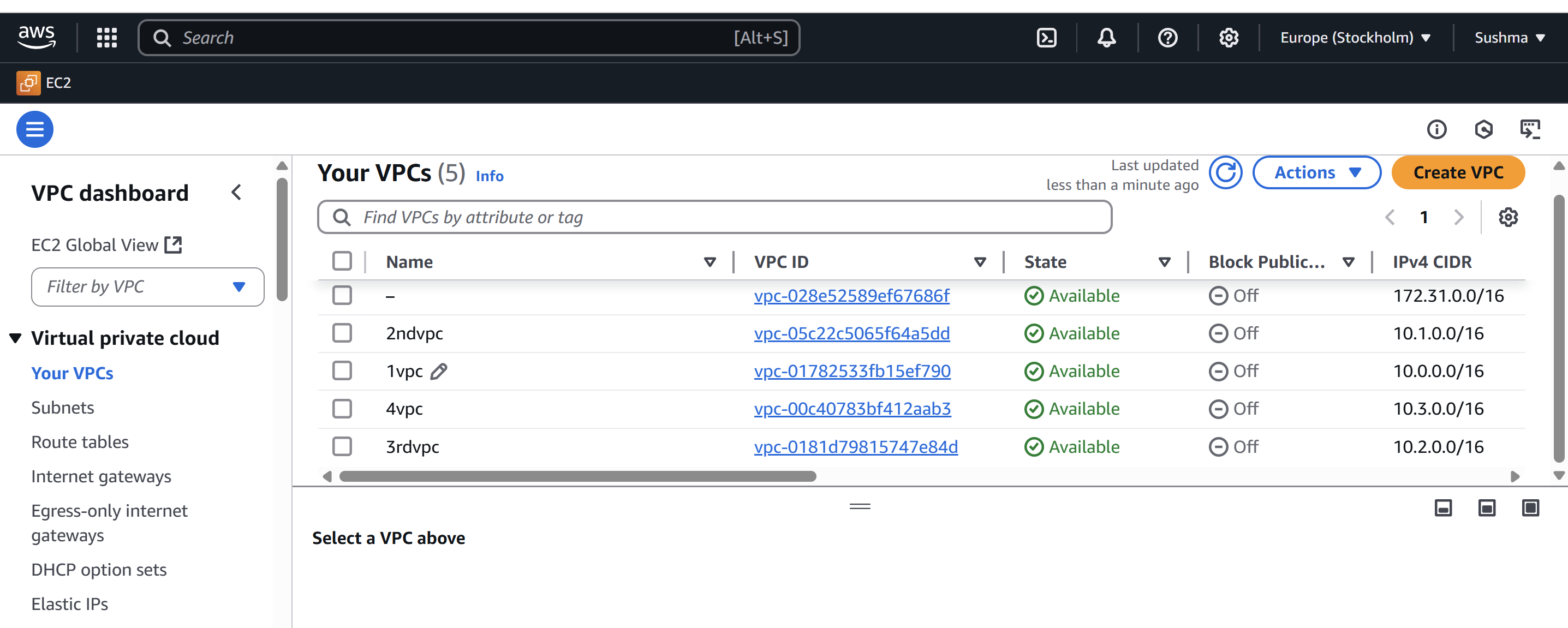


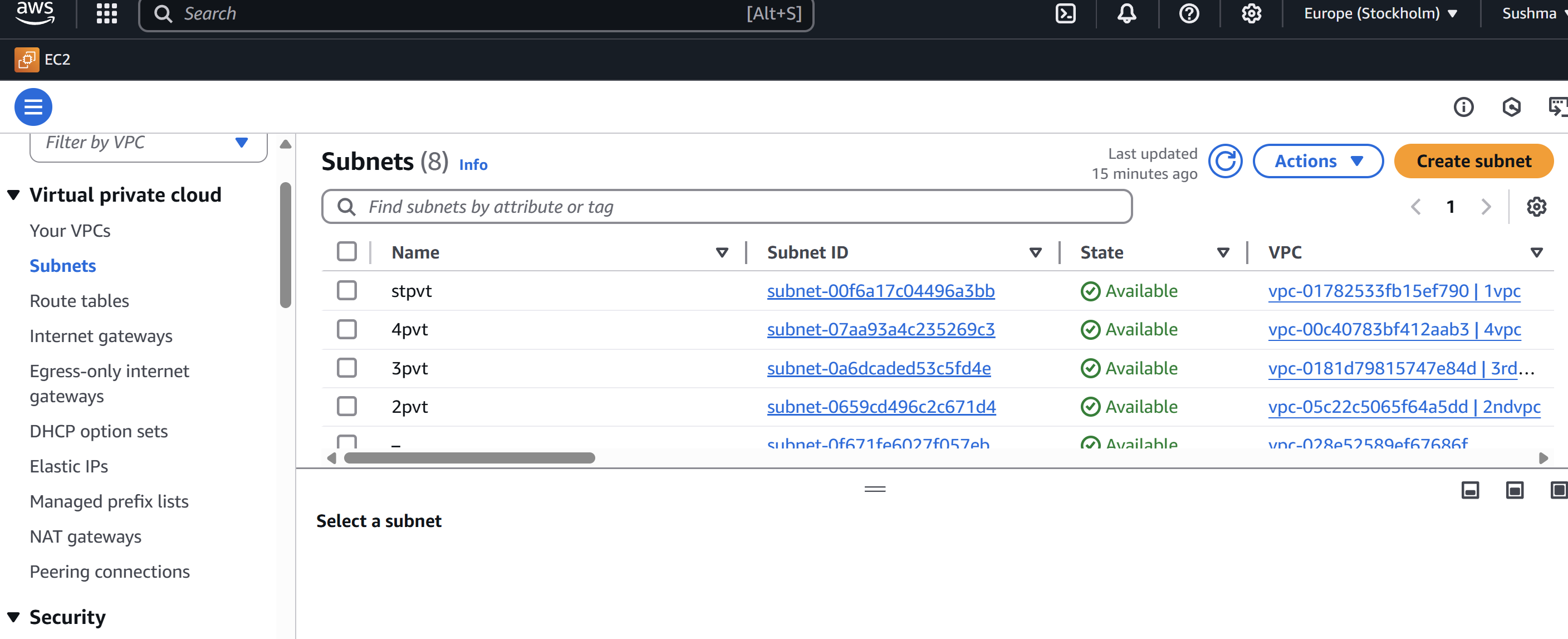


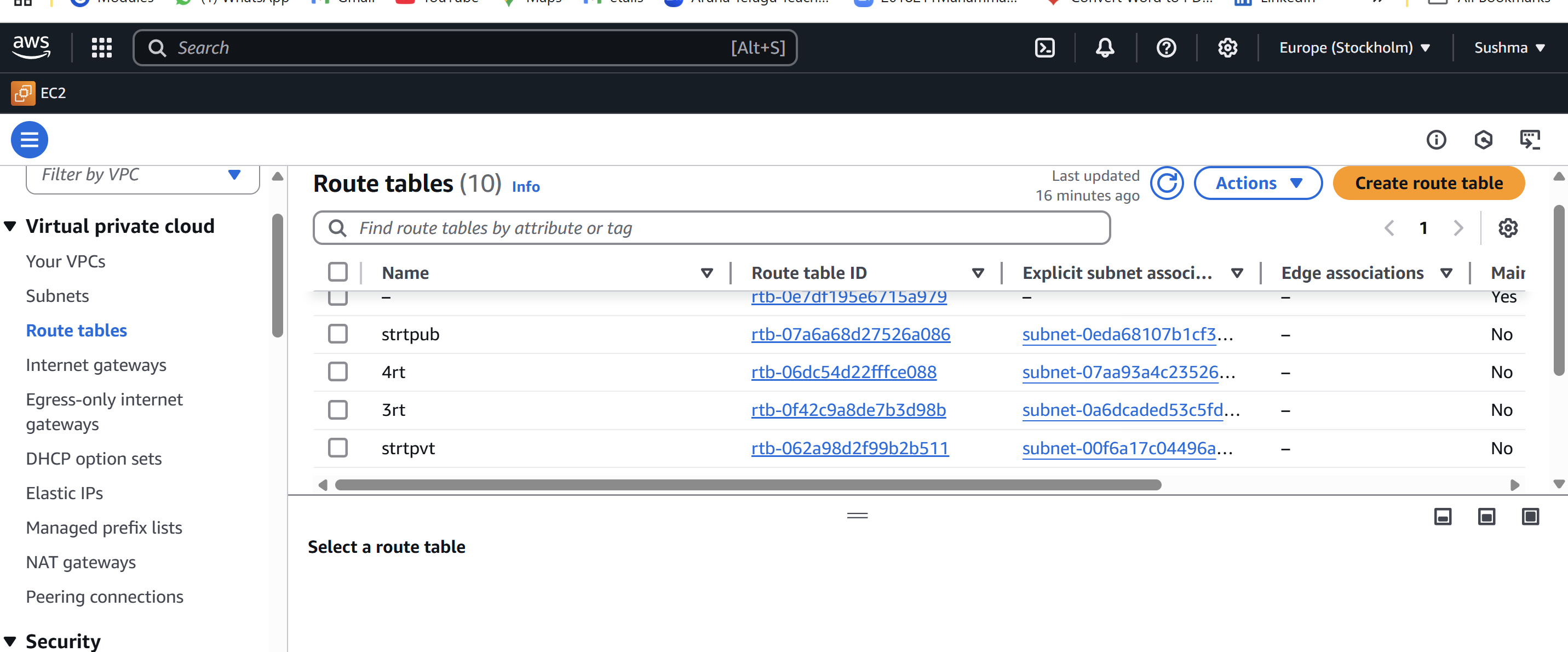
4) Setup VPC Transist gateway.

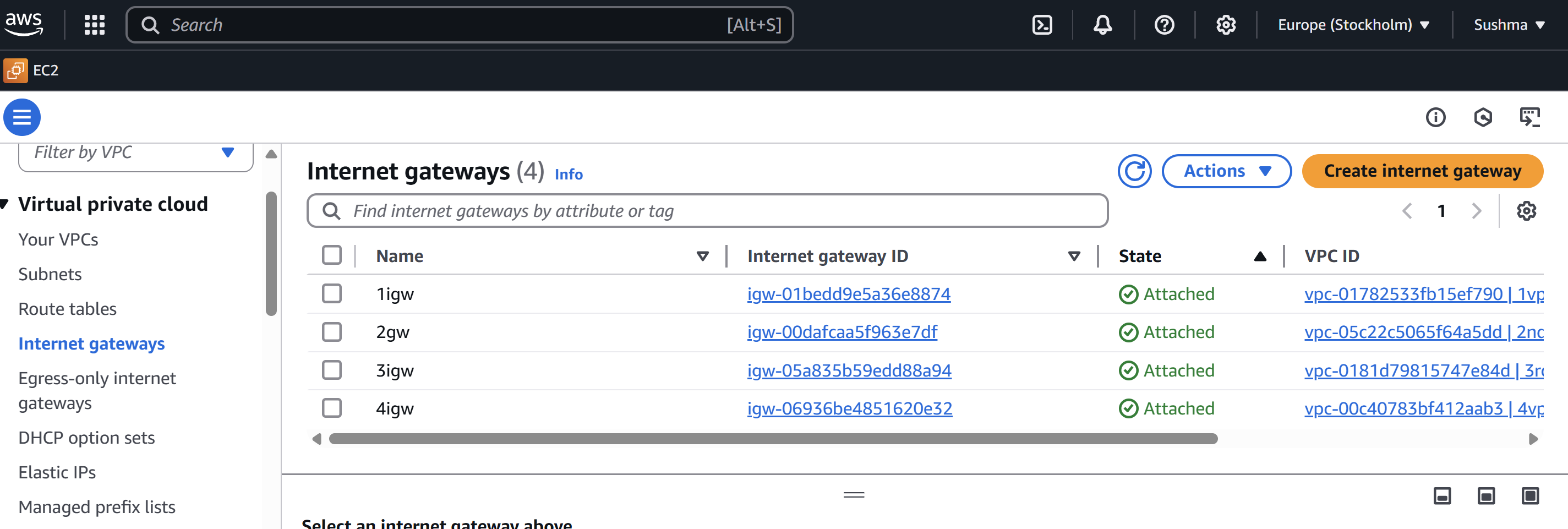
Go to VPC and create 4 vpc’s with CIDR

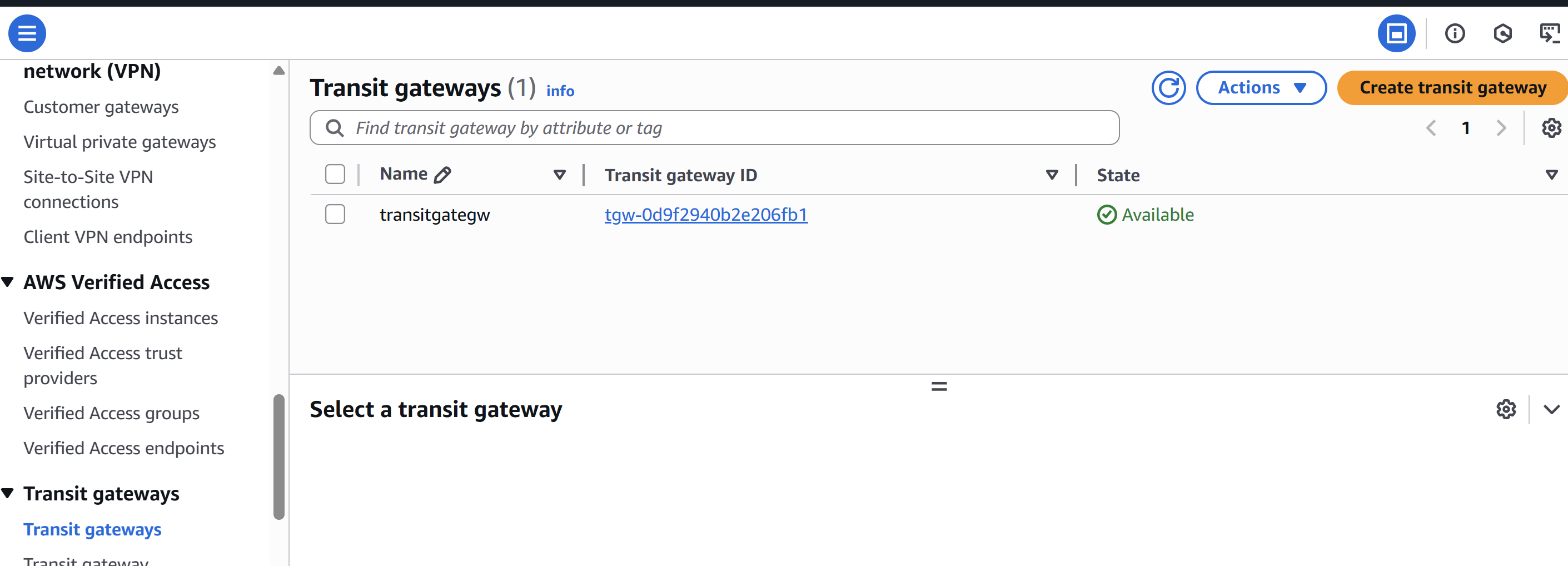
* Vpc1 with 10.0.0.0/16 and one public subnet with 10.0.0.0/20
  + - Vpc2 with 10.1.0.0/16 and one private subnet with 10.1.0.0/20
    - Vpc2 with 10.2.0.0/16 and one private subnet with 10.2.0.0/20
    - Vpc2 with 10.3.0.0/16 and one private subnet with 10.3.0.0/20
* Now go to transit gateway provide name tag,ASN as default,enable everything and click on create(TG will not ask for any vpc or subnets)
* Now click on Transit gateway attachment and click on **create Transit gateway attachment provide name** tag,transit gateway id,attach type(eg:vpc)
* VPC1 id then click on create
* Similarly,create attchment for all the VPC’s 2,3,and 4
* After creating TG attachment it will by default create TG route table by default
* Do not make any changes in TG RT
* Go to transit gateway Route table and create one route table click on check box in association we can see vp’s added
* Now go to routes they are automaticlly created
* Now launch 4 ec2 instance with 4vpc and login to public ec2 vpc1 and ping with vpc2 ec2instance private id (no internet access)
* Because all the request whivh have been redirecting to these vpc’s we haven't mentioned our vpc where this traffic need to be redirected.
* We know that we have Tc created which is help us to distribute with subnet and all
* But our vpc this request has been redirected to our vpc routing table and our routing table is not aware where this traffic need to be sent.
* Now go to RT click on vpc1-RT
* Edit routes and add routes copy vpc 2,vpc3,vpc4 CIDR and paste ,then select Tg and TG id ---->save
* Similarly do this to all Rt
* Now Go to git bash and try to ping (still it is not connected)
* In vpc1 internet gate way attched and now create IGW all the vpc’s
* Similarly connect all the vpc’s to IGW --->create IGW for all the vpc’s
* Go to add--->IGW all the RT
* Check security group to all the instance (all traffic--->save)
* Now check vpc1 with vpc2,vpc3,vpc4 (private ip)
* Similarly check vpc2 with vpc3,vpc4and vpc1
* Check vpc3 with vpc1,2,4
* Check vpc4 with vpc1,2,3
* All these are communicate

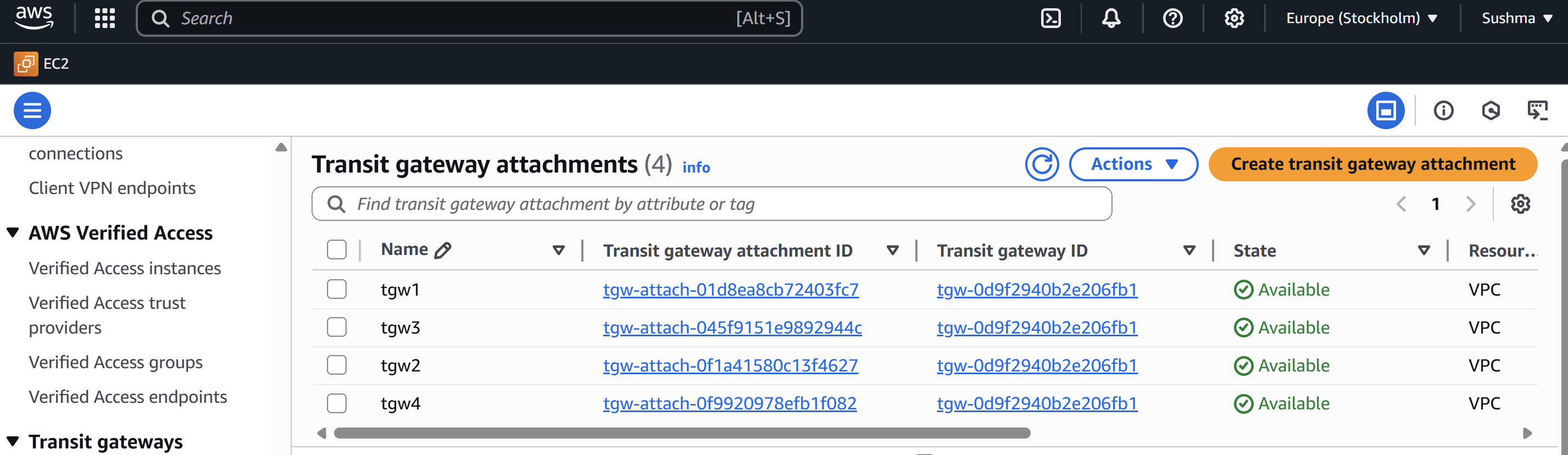


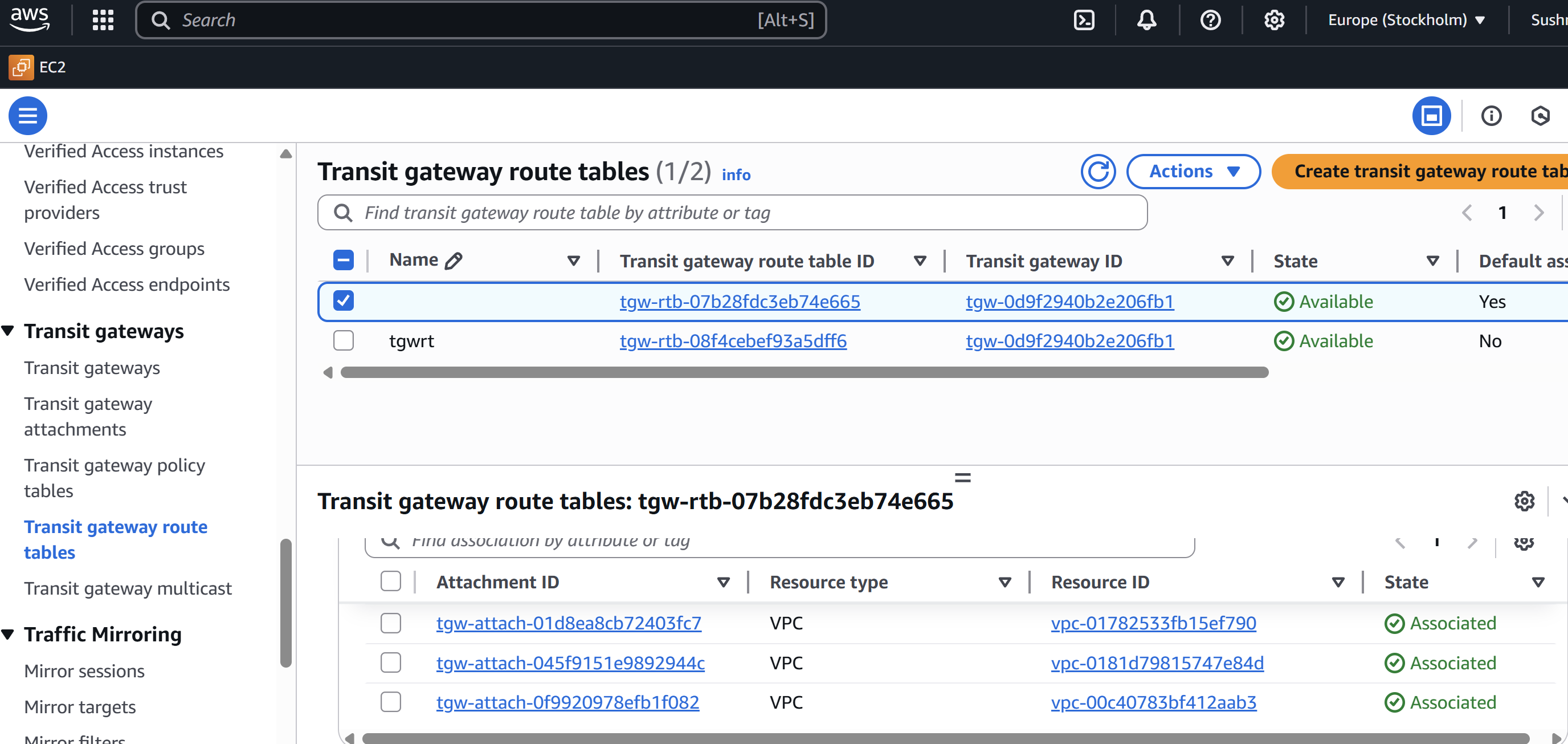




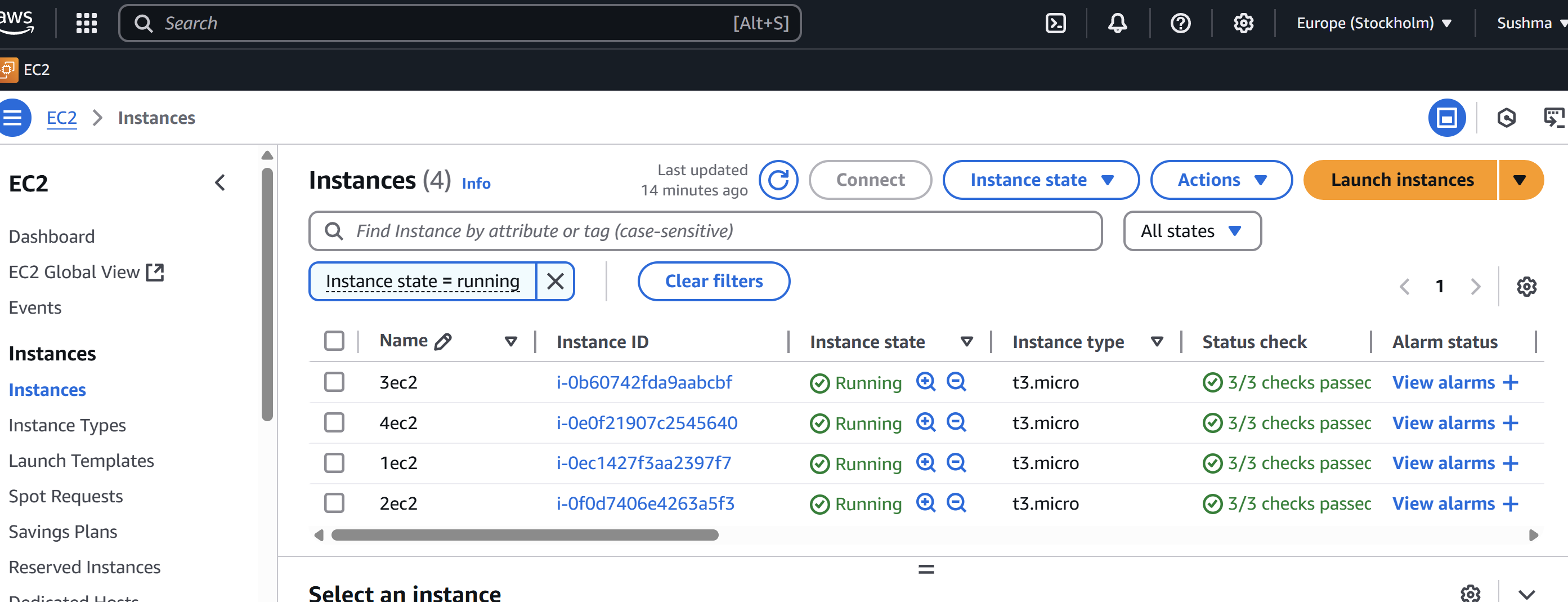


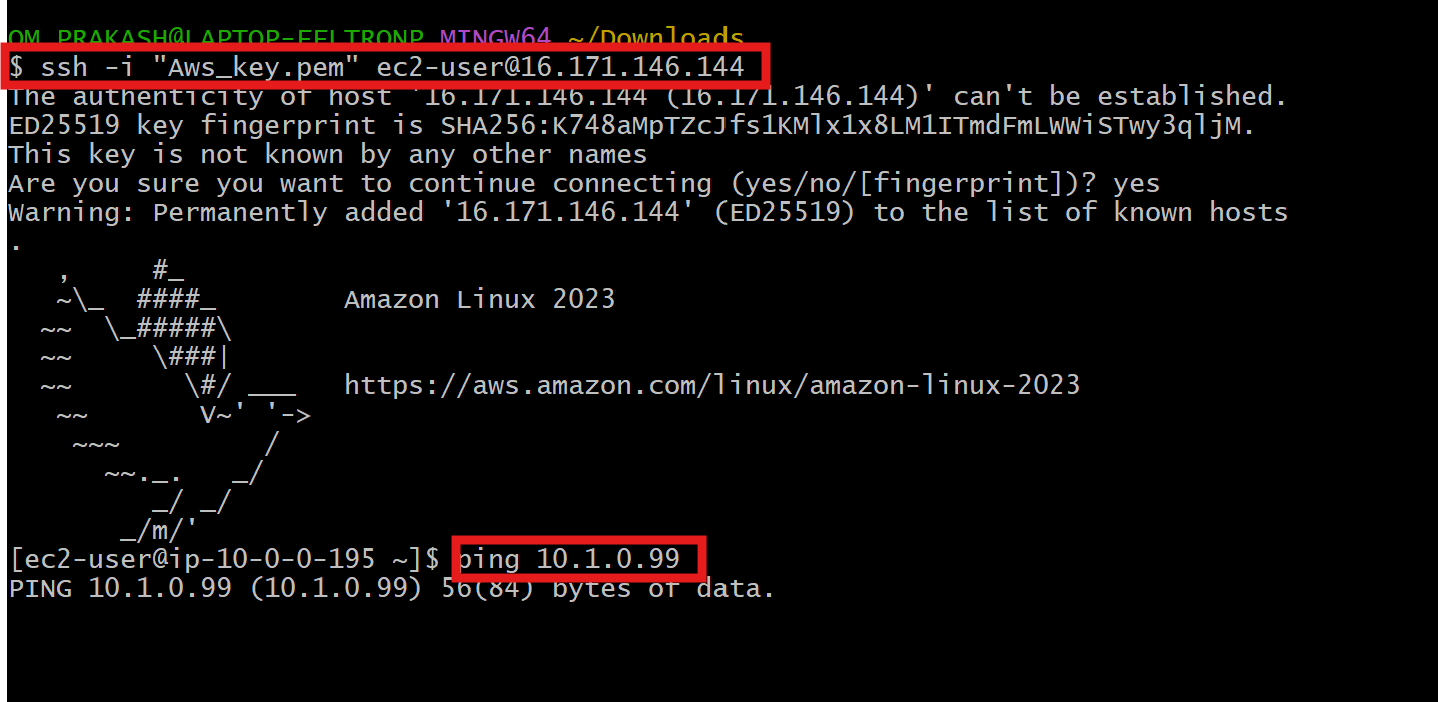


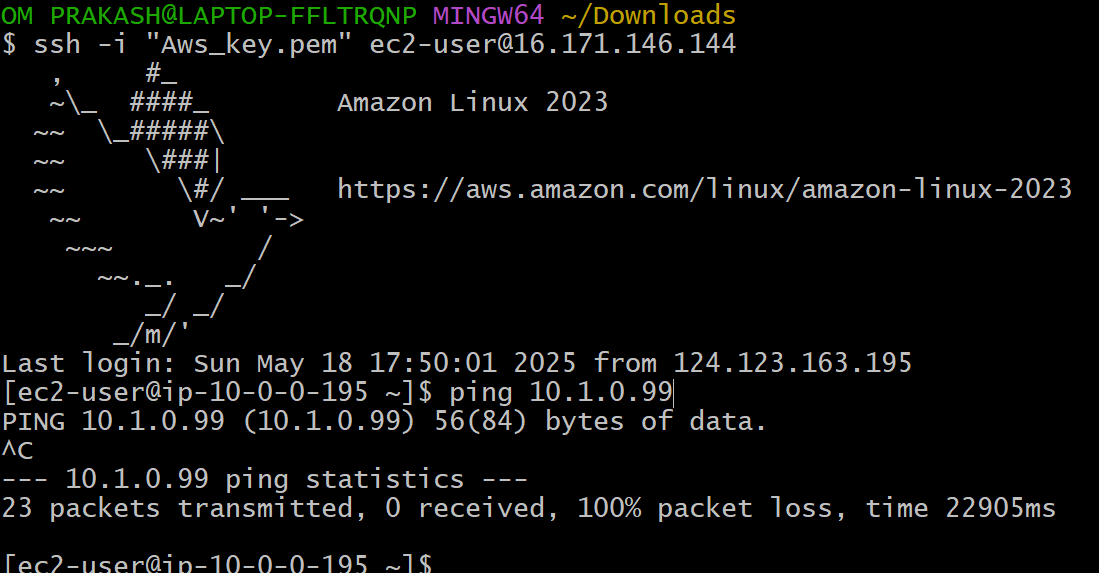








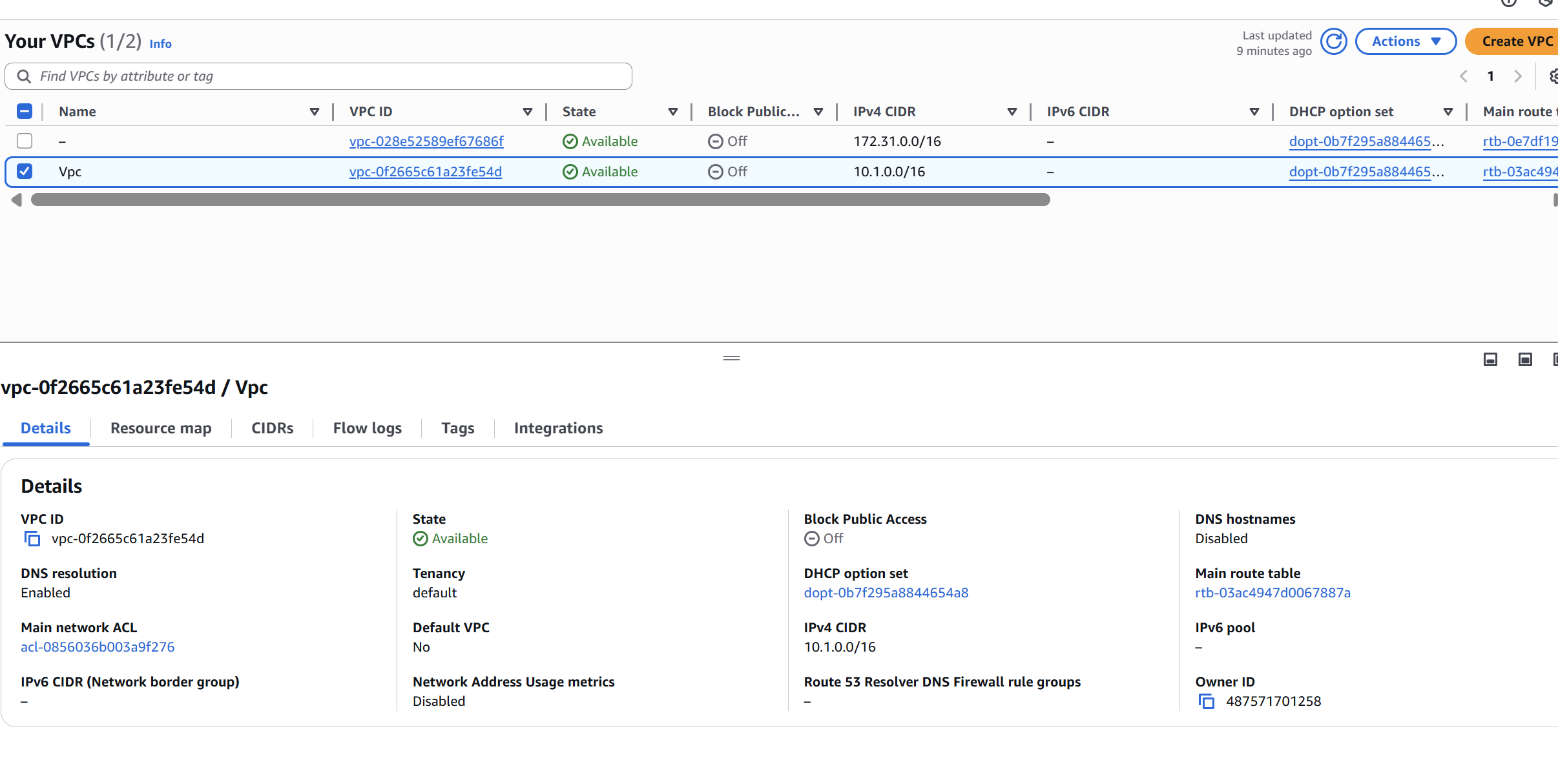


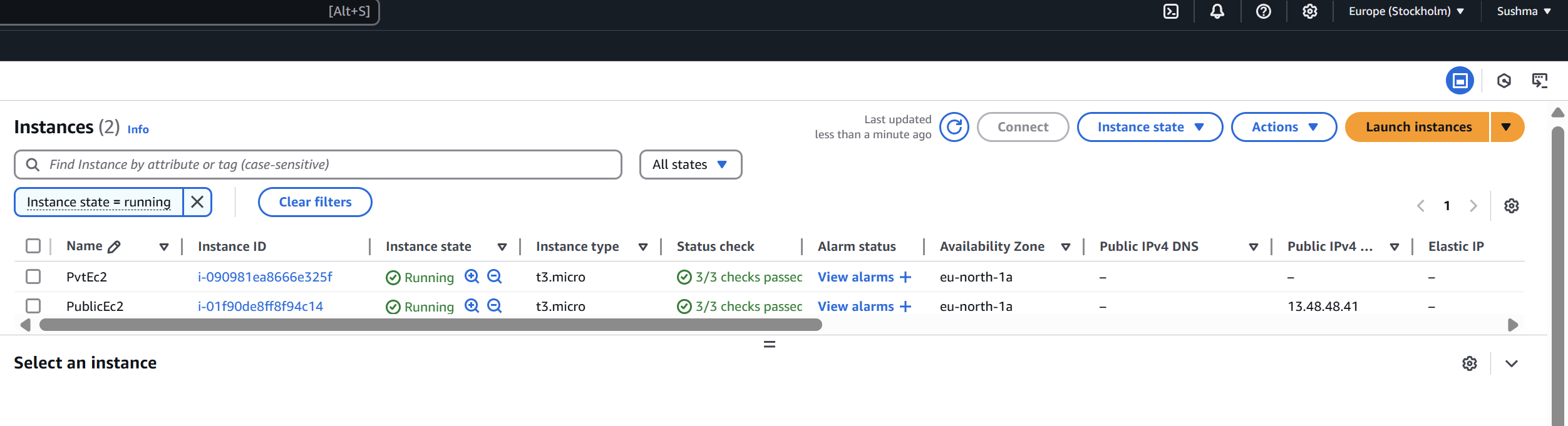


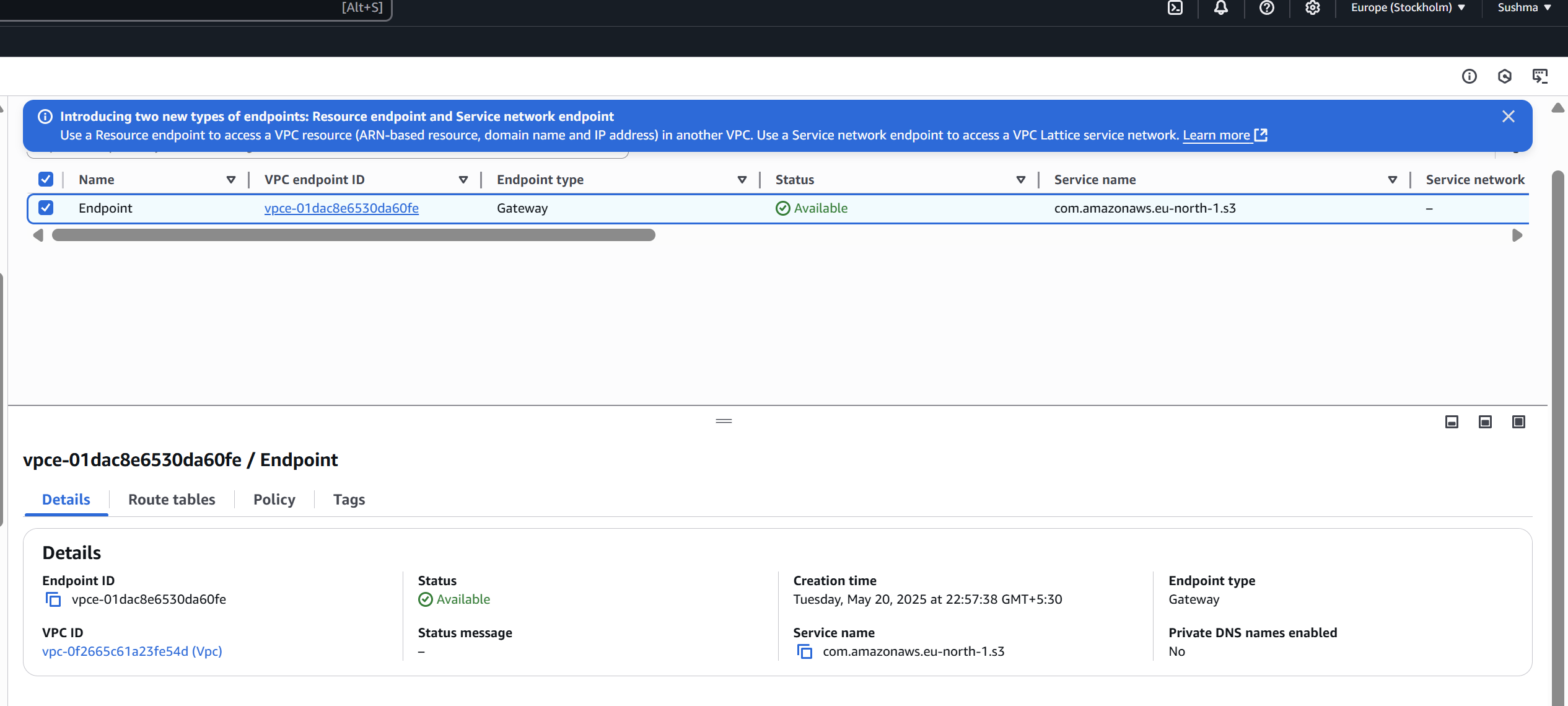
5) Setup VPC End Point.

**Go to**AWS console  
l Open VPC ---->go vpc--->click on subnet---->take on pub-subnet and private -subnet  
l Launch 2 instaces in ec2 with pubnet (enable) and private with (disable)  
l Now create S3 bucket ,Search S3 qnd use existing one  
l Go to ec2 public ec2 and connect to bash after connecting  
l Aws s3 ls(for public ec2 if it is not configured then configure)  
l Aws configure  
l Access key and security key provide Go to security crendentials in ec2  
l Aws s3 ls  
l We can access the bucket  
l Now we want to connect with private IP  
**l In**in publich ec2 create file with pem key and provide chmod 400 permission  
l Then connect to ec2(jump server)  
l Search foe aws s3 ls  
l It will not give any output because it is not connected to internet so **now creating end pint for this s3**

Now creating endponit for this S3 and attach endpoint to the RT of Private ec2.  
Go to aws -->-vpc--->end ponits  
Create endpoints--->name (s3-endponits)  
Choose any service categatoty(aws service)  
Choose s3  
Choose gateway  
Vpc(default)  
RT(private)  
Policy(full access)  
Create endpoints  
Now in git bash we get connection  
With help of end point we can communicate without any internet connection to check s3



ss

securely