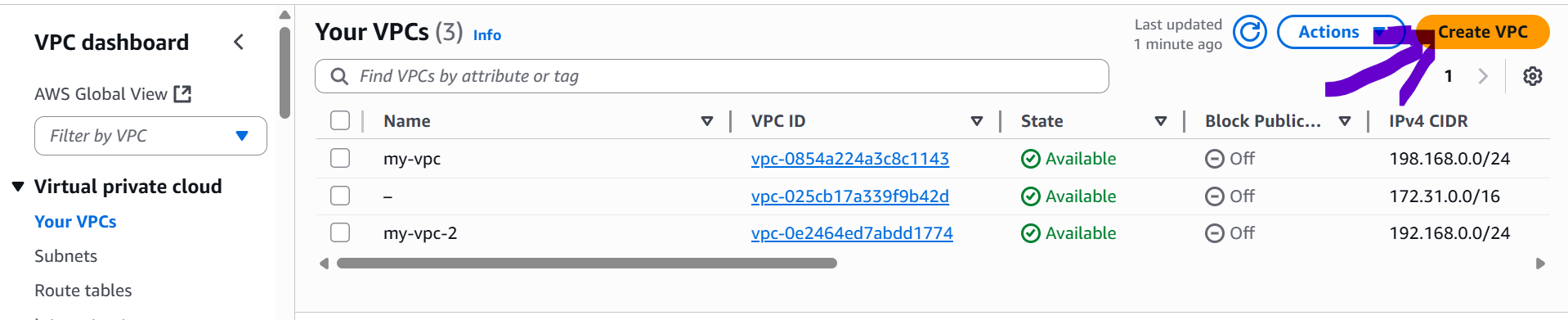
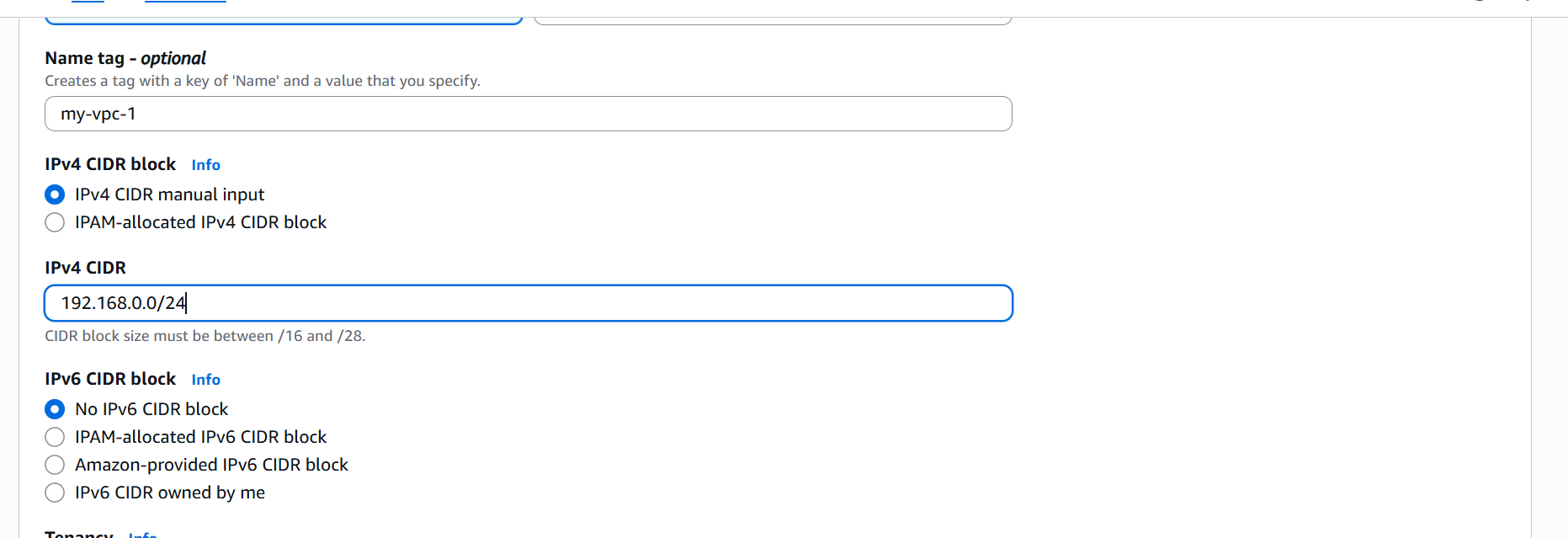
**VPC ASSIGNMENT -2**

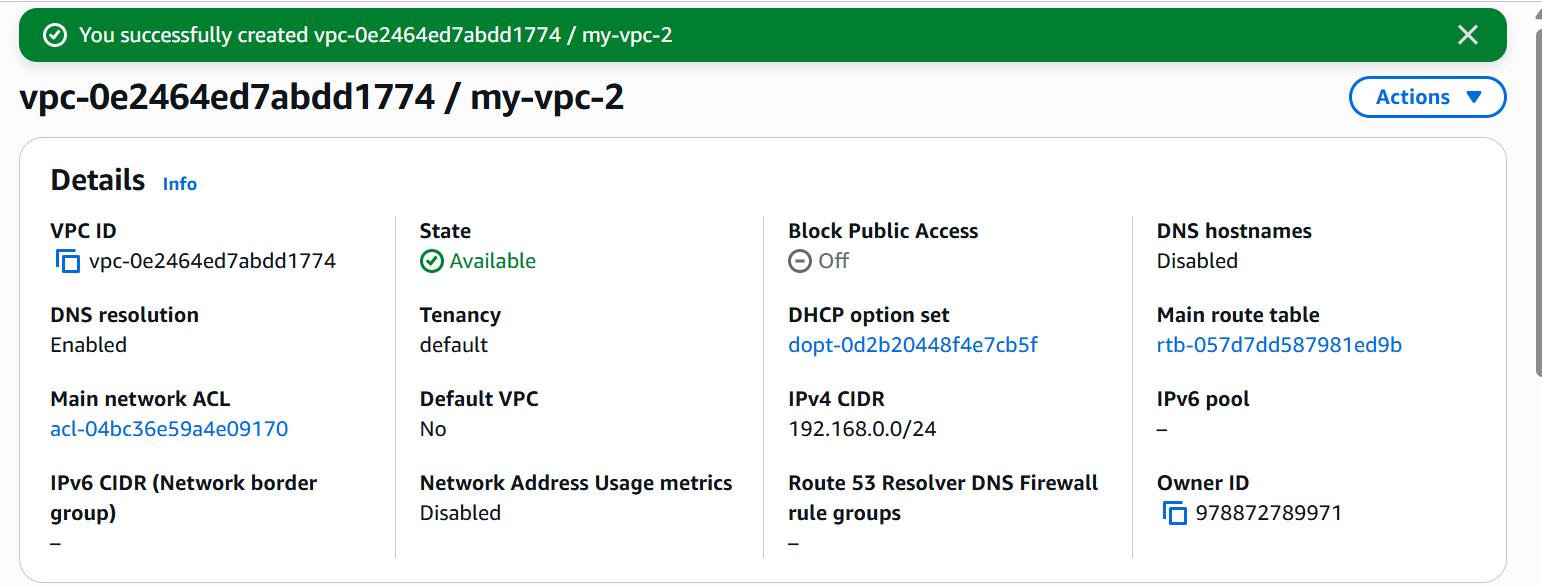
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

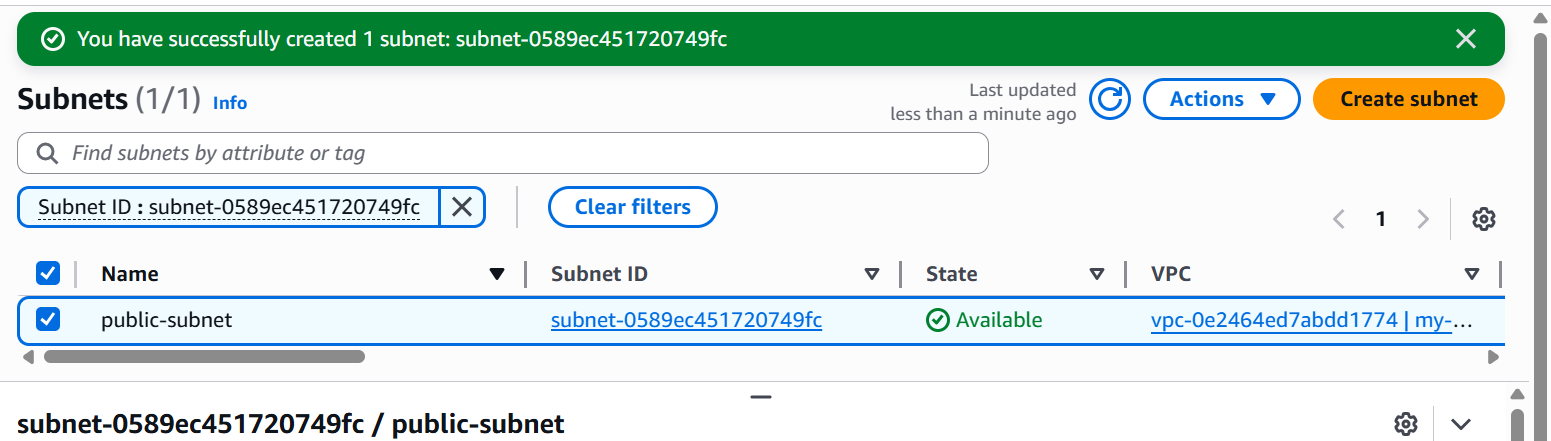
* Go to **AWS Console**
* **Go to VPC → Your VPCs**
* **Click on Create VPC**.
* **Name:** My-VPC
* **IPv4 CIDR block:** 10.0.0.0/16 (large enough for subnets)
* **IPv6 CIDR:** Optional (skip for now)
* **Tenancy:** Default
* Click **Create VPC**



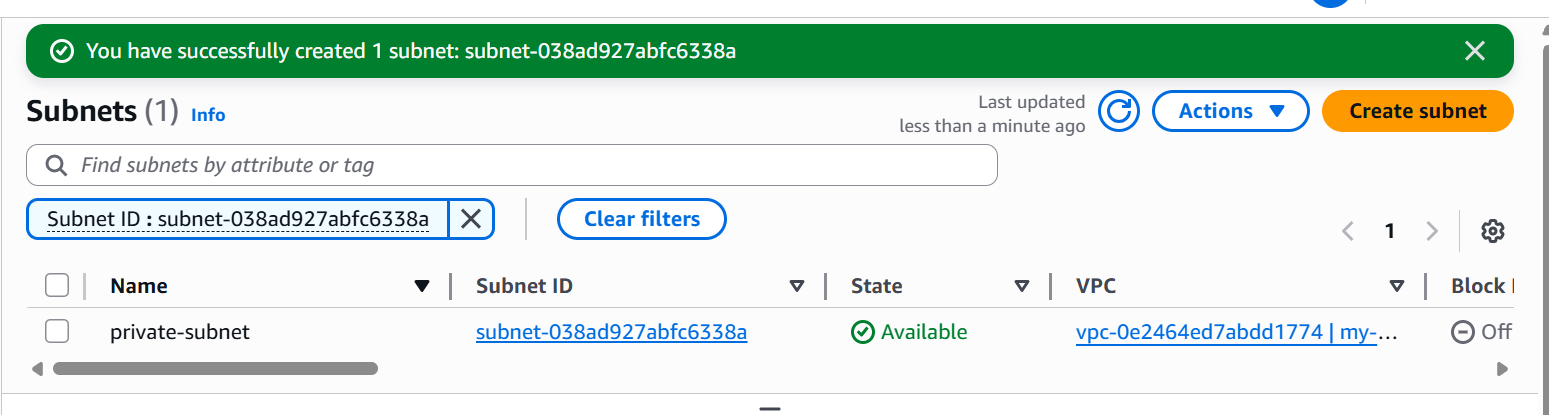




* Go to **Subnets → Create Subnet**.
* Select your VPC My-VPC.
* Name it Public Subnet.
* Choose **Availability Zone** (any one)
* CIDR block: 10.0.1.0/24
* Click **Create Subnet**

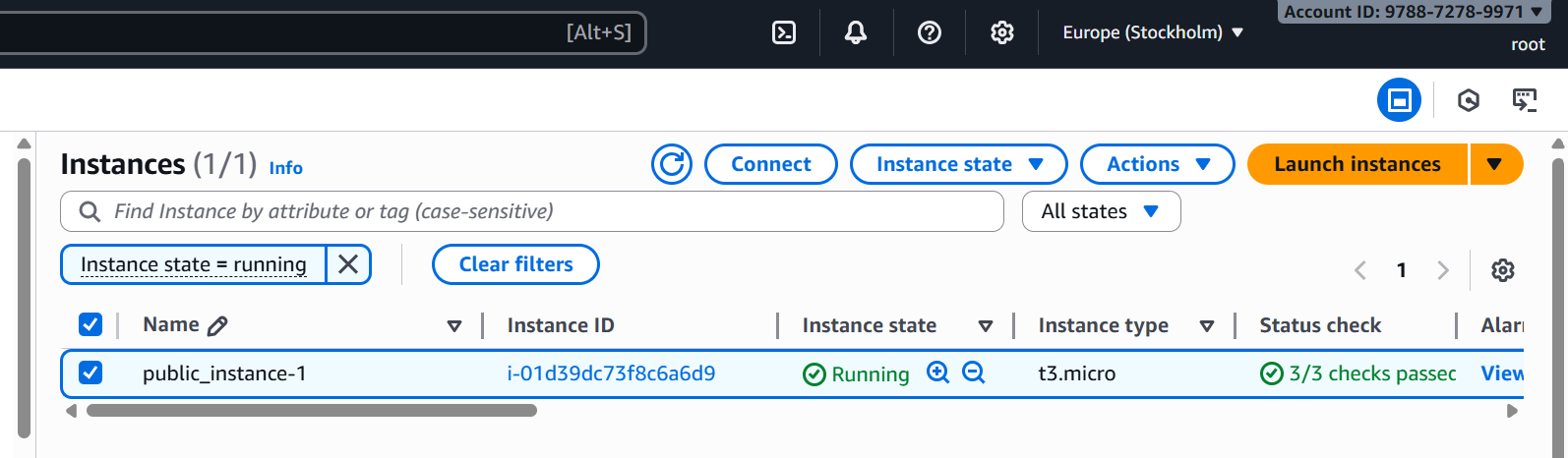


* Again, **Create Subnet**
* Name it Private Subnet
* Select same VPC My VPC
* CIDR block: 10.0.2.0/24
* Click on **Create Subnet**

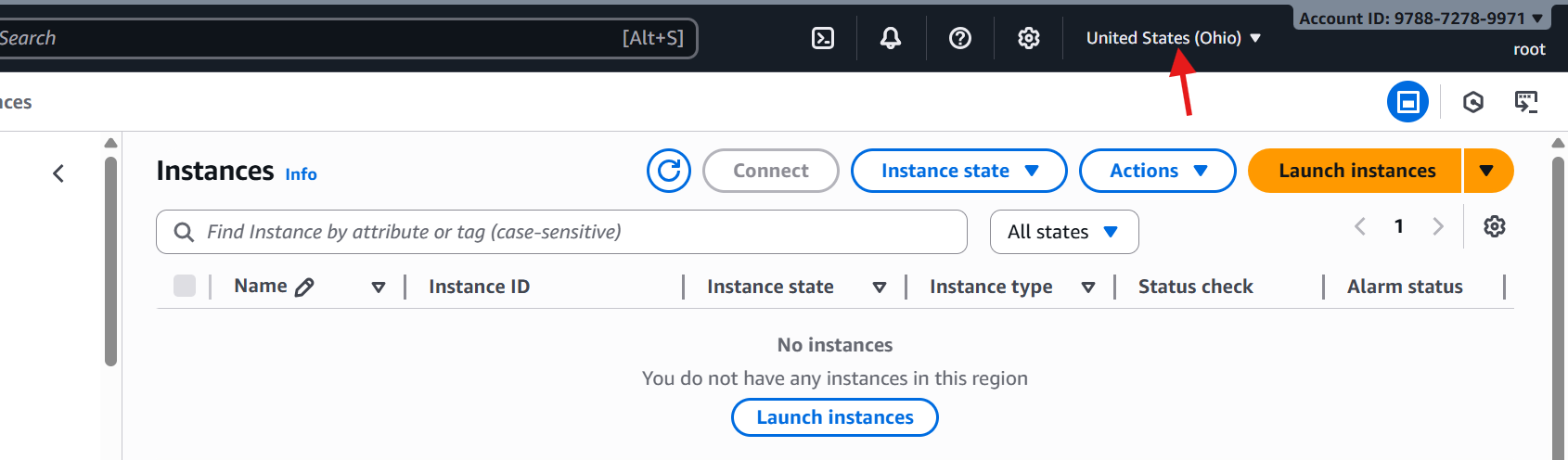


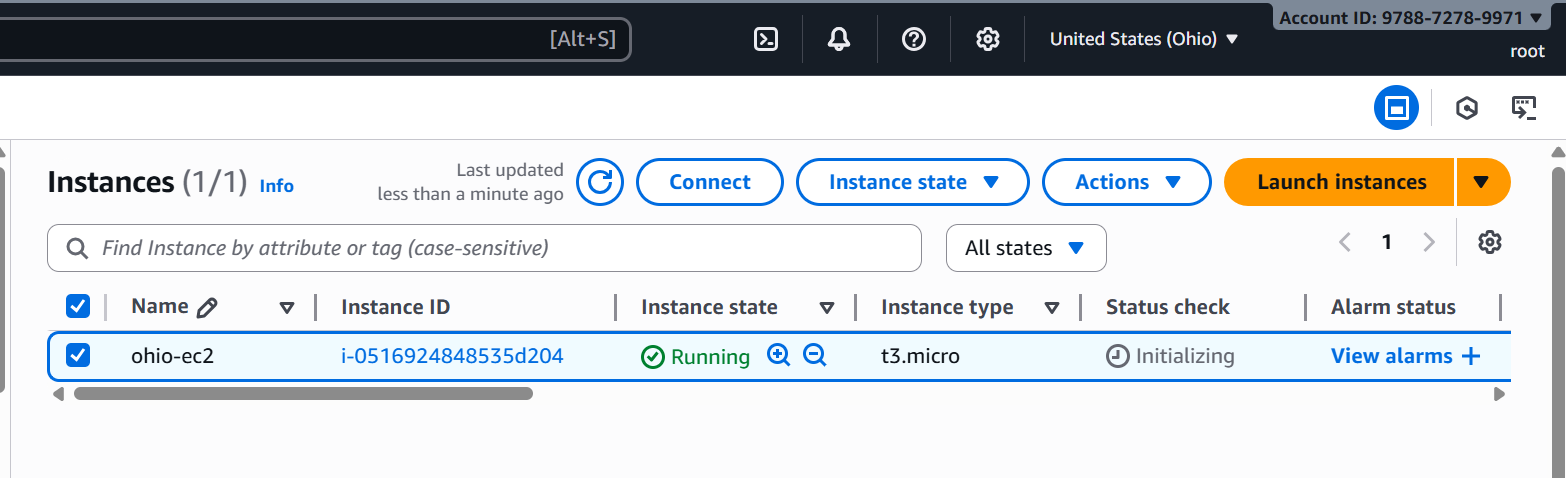
1. Enable VPC peering for cross-region

* Open Aws console and lauch one instance in Europe(stockholm)

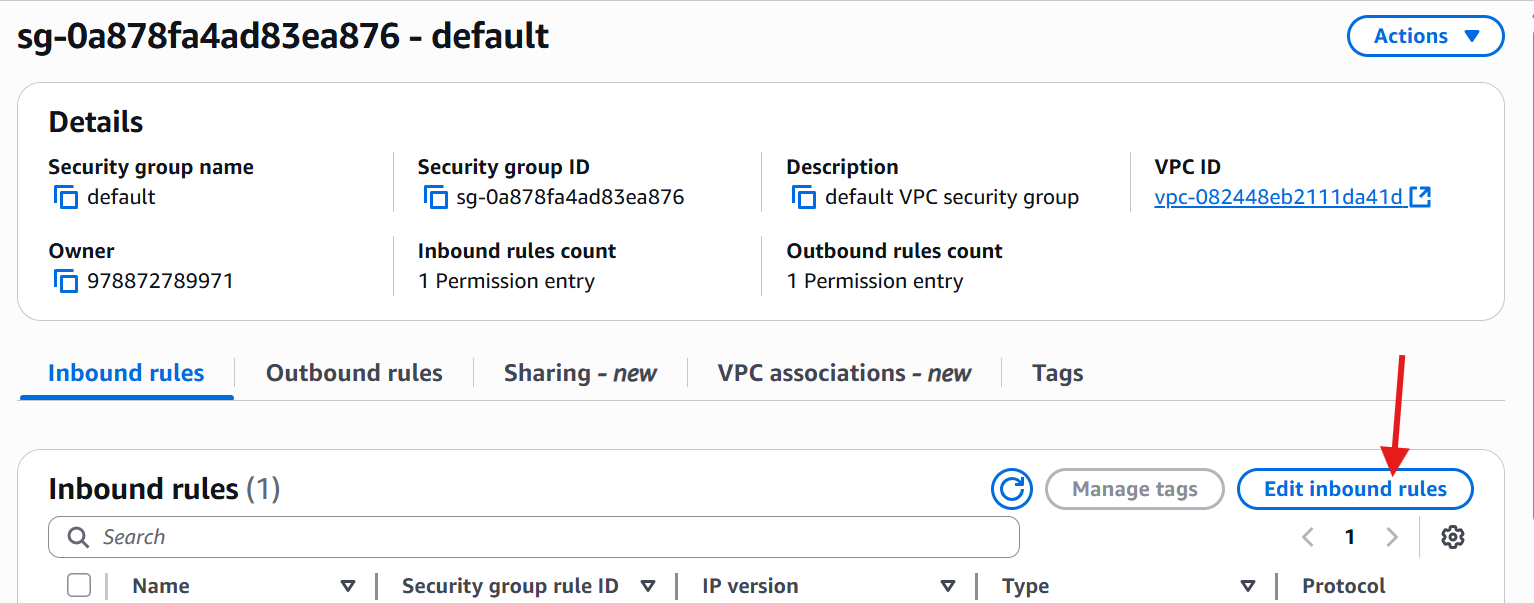


* And duplicate the tab and put the region as ohio
* Lauch one instance in ohio region

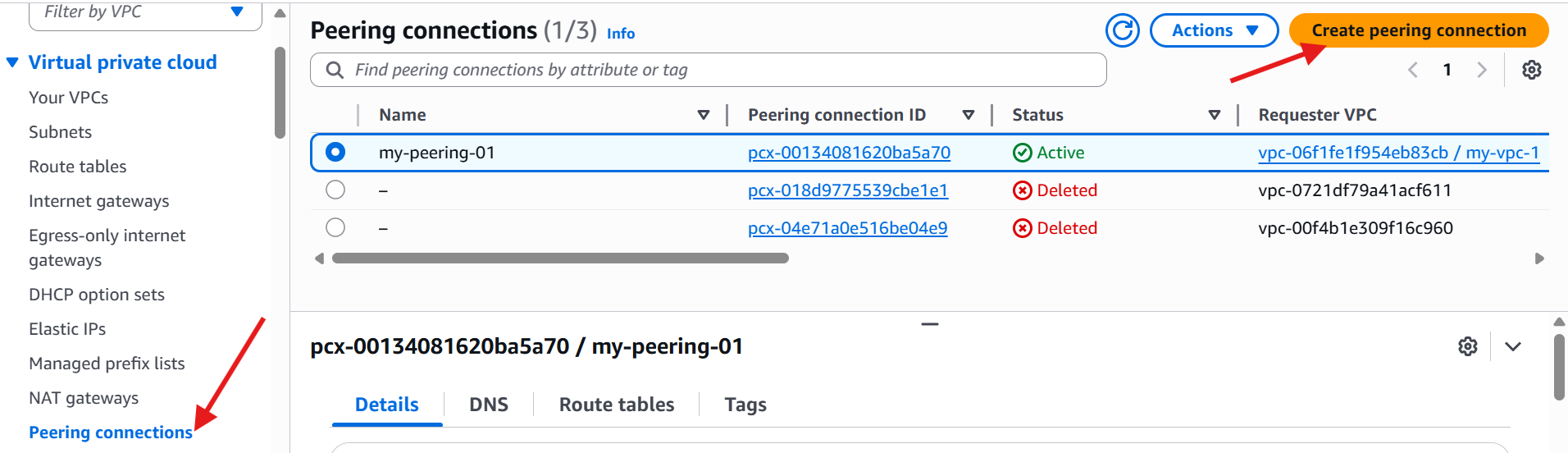




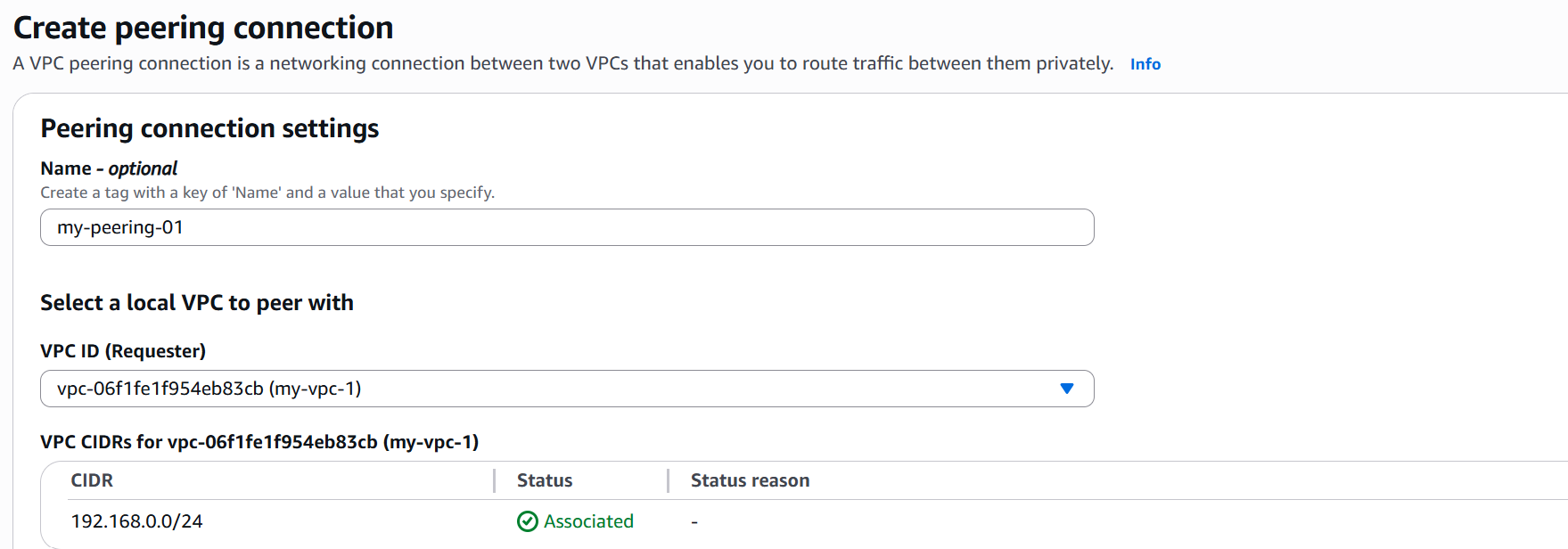
* Go to security and add the inbound rules in ohio region instance



* Go to Europe(stockholm) region
* Go to vpc and click on peering connection
* Create peering connection

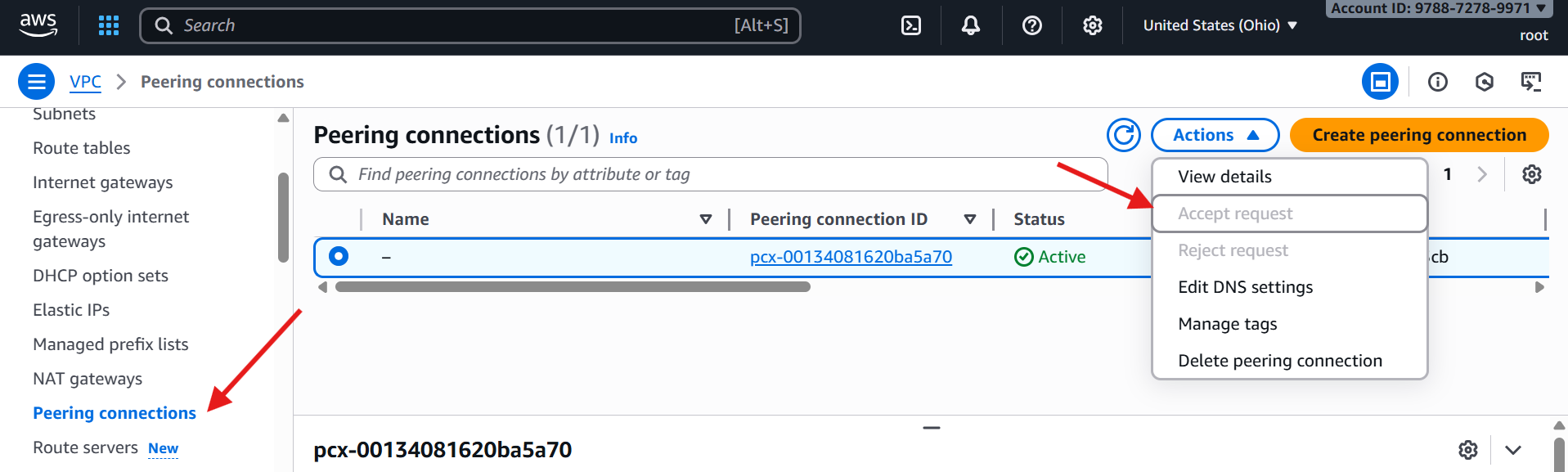


* Give name
* Select your local vpc
* Select another VPC to peer with 🡪 My account 🡪 another region
* Select the region **Ohio**
* VPC ID (Accepter) copy the Ohio vpc id and paste
* Click on create peering connection

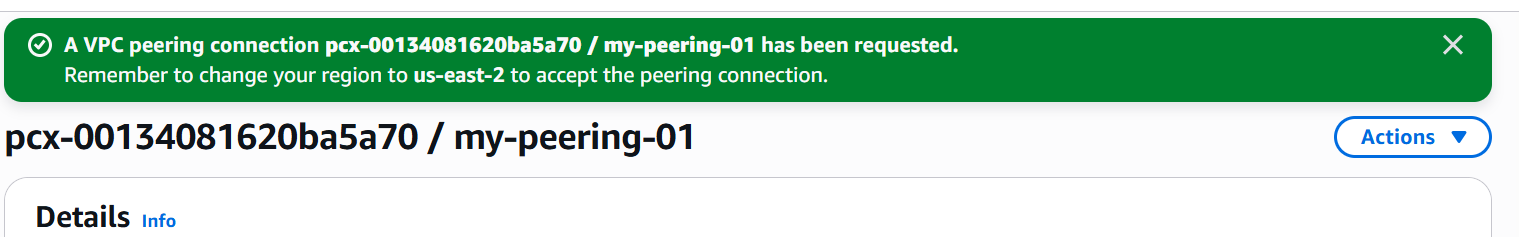


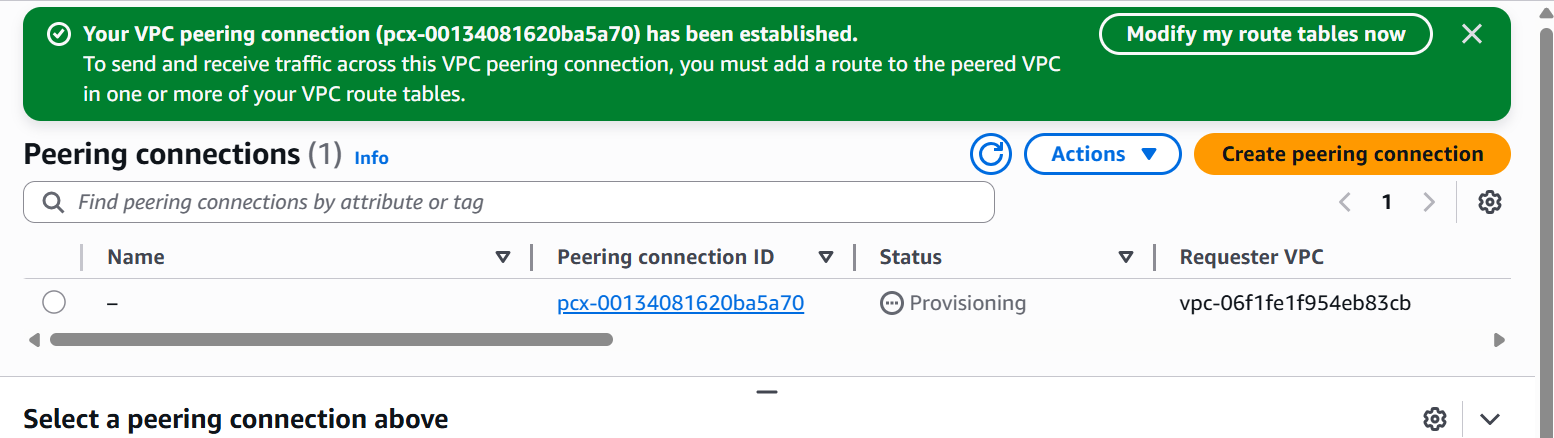


* Go to Ohio region vpc
* Click on peering connection
* You got one request from another region(Europe)
* Accept the request

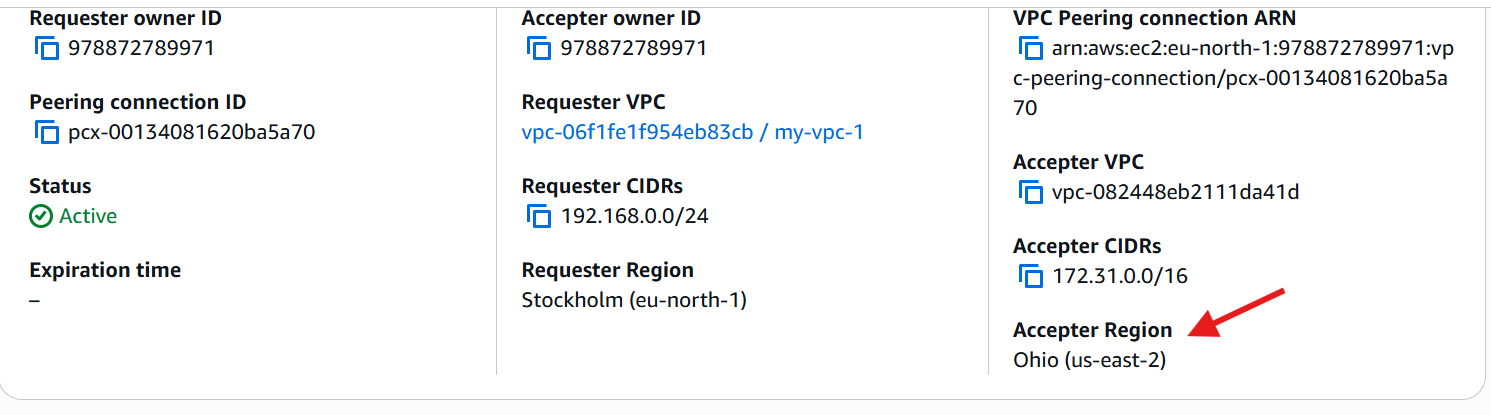


* Here the results



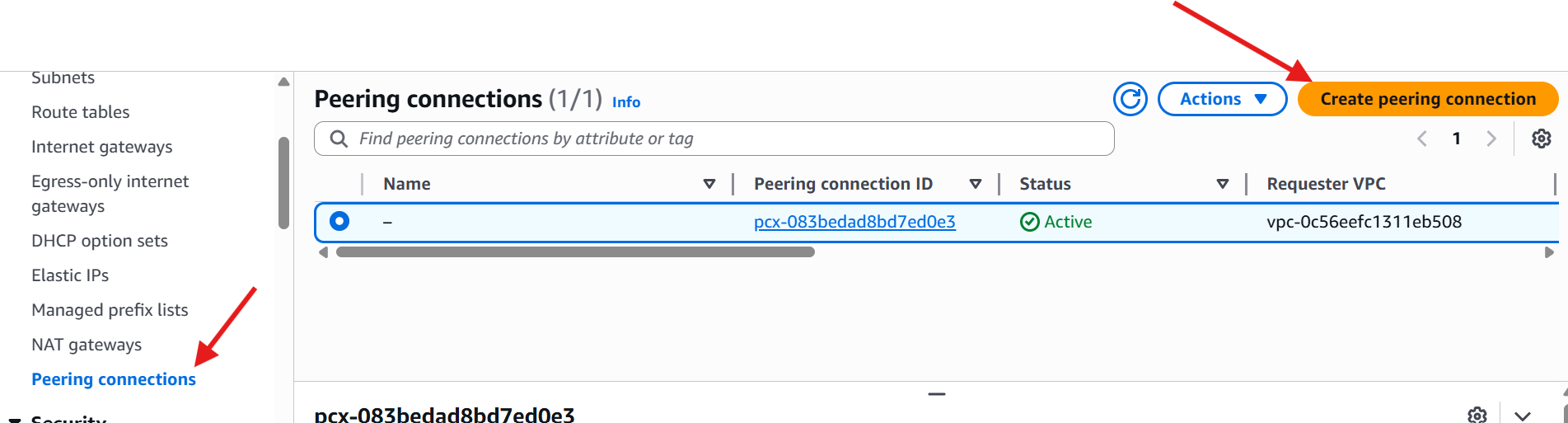


* Check in Europe region it shows like Acceptor (ohio region)

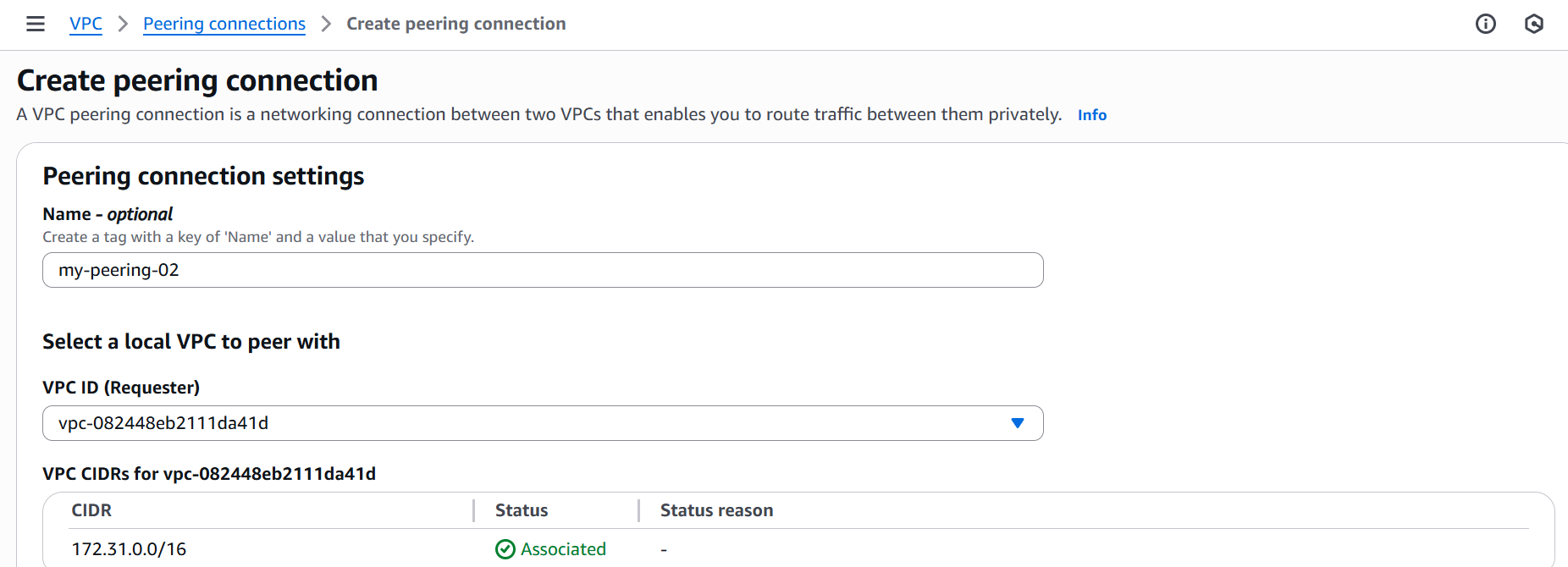


1. Enable VPC peering for cross-account (you can collaborate with your friend to do this task)

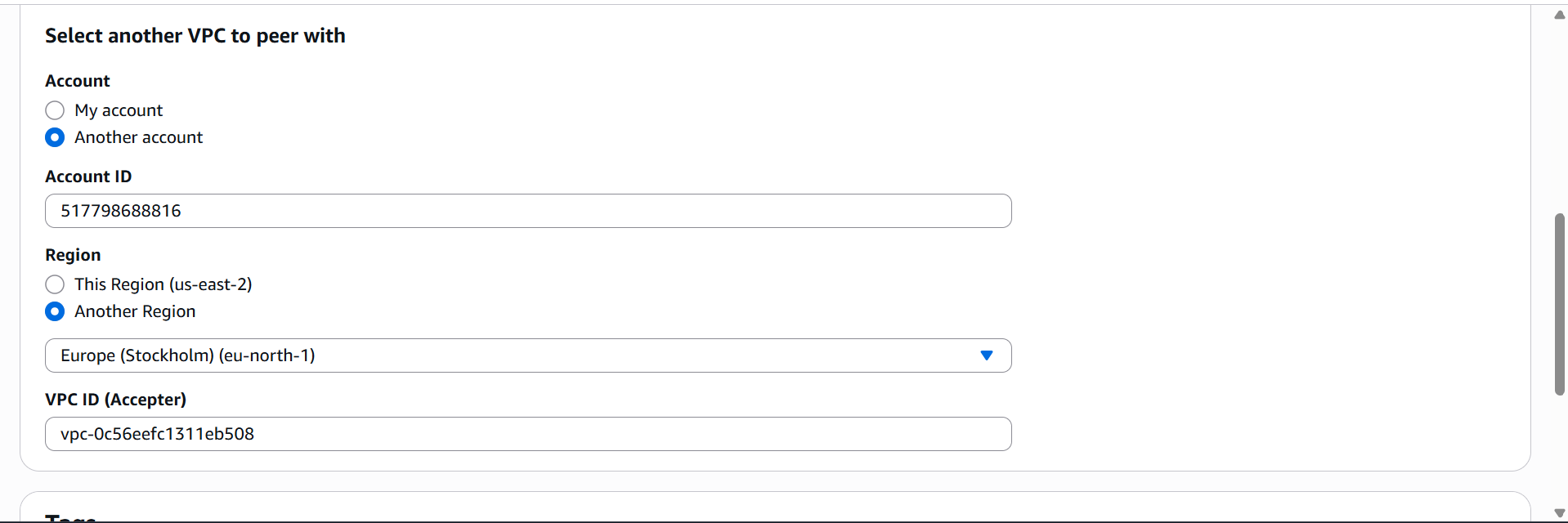
* Go to ec2 console
* Open search bar and enter **vpc** and then check your **VPC** in which region(ohio)
* Click on **peering connection** and click on **create peering connection**



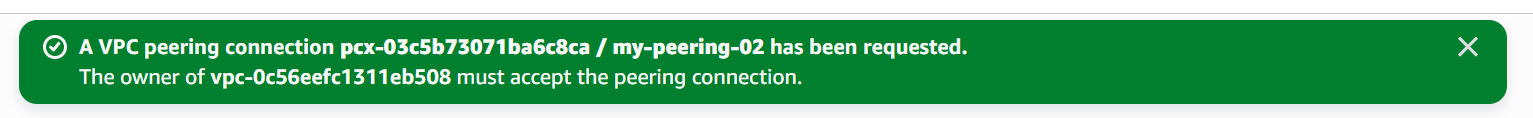
* Create peering name
* Give your vpc



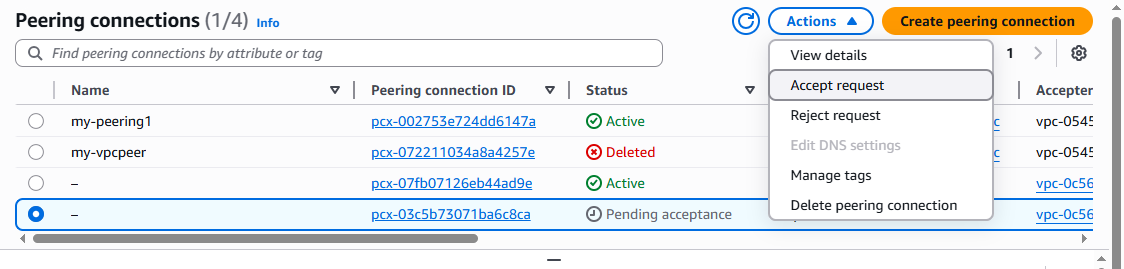
* **Account:** select (Another account) 🡪enter friend account id
* **Region:** select another region **🡪**enter friend region
* **VPC ID :** Enter friend VPC id
* Before of this **choose different account & Different region.**

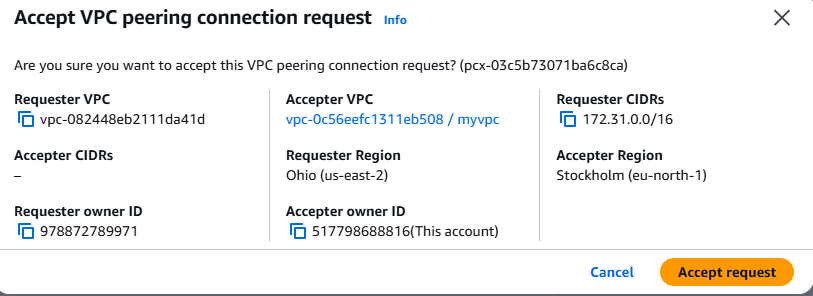


* **This is my page where I send a request**

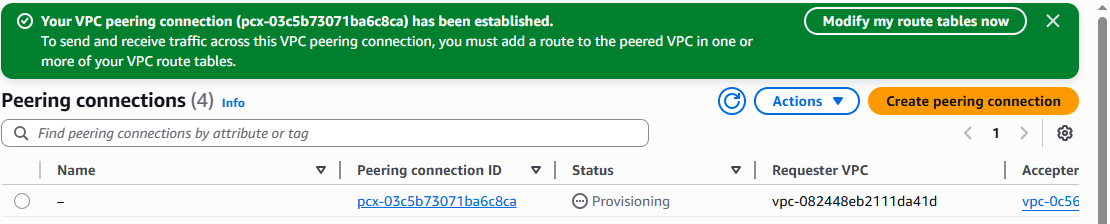


* **Here the results :other region other account**

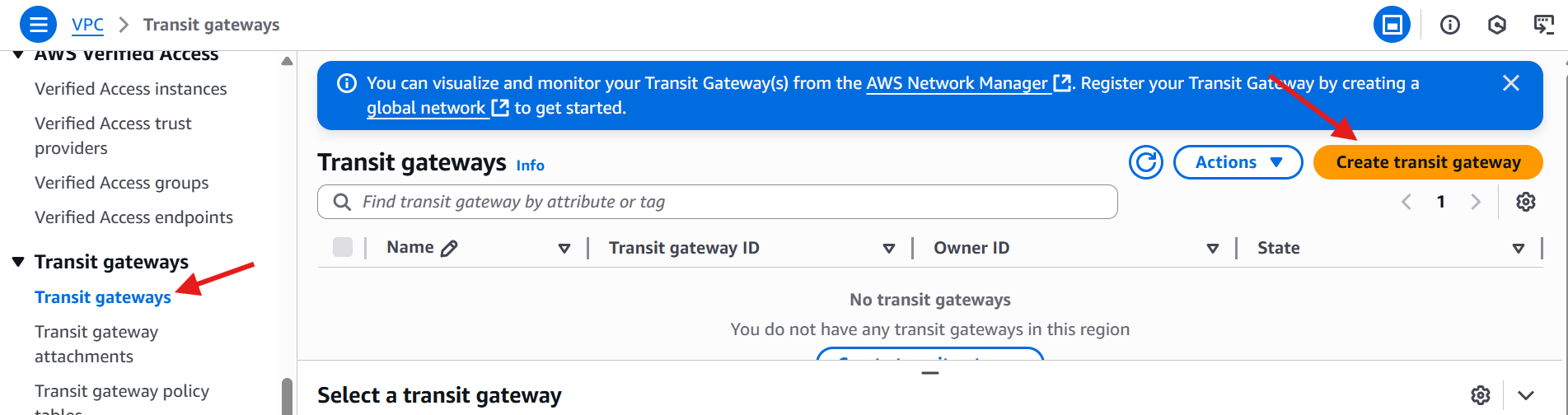


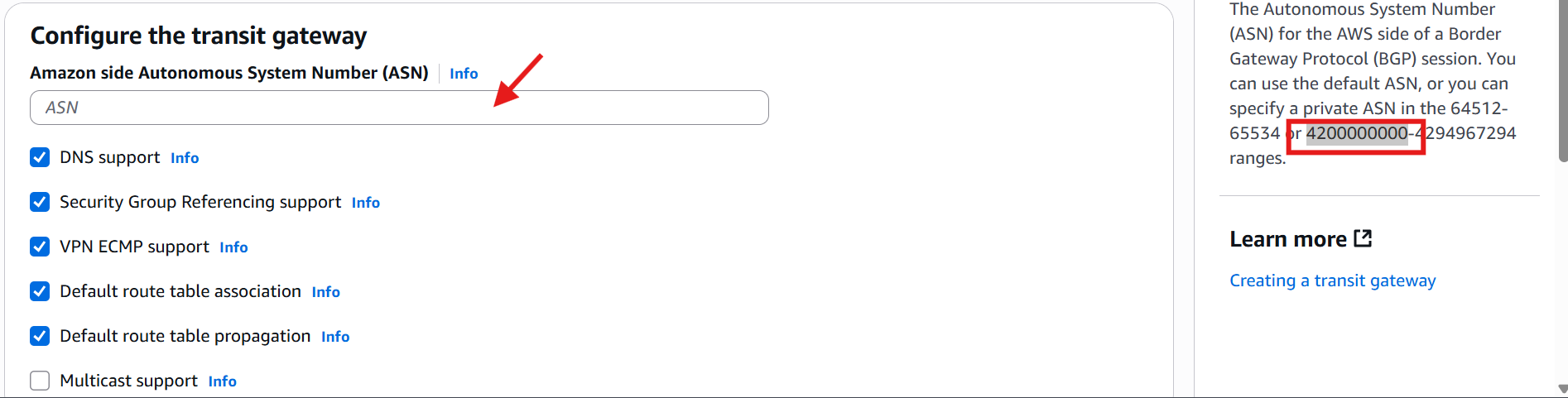


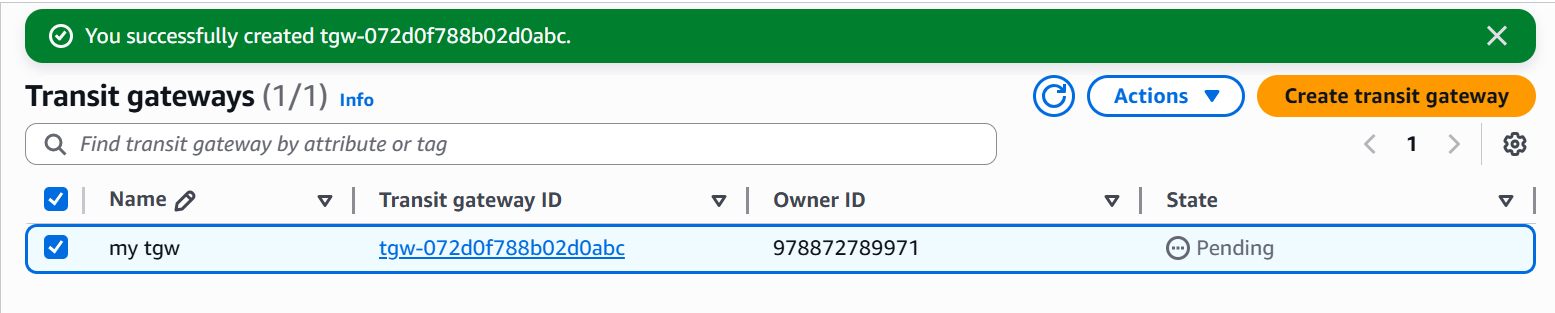
* **Here vpc peering connection has been established.**



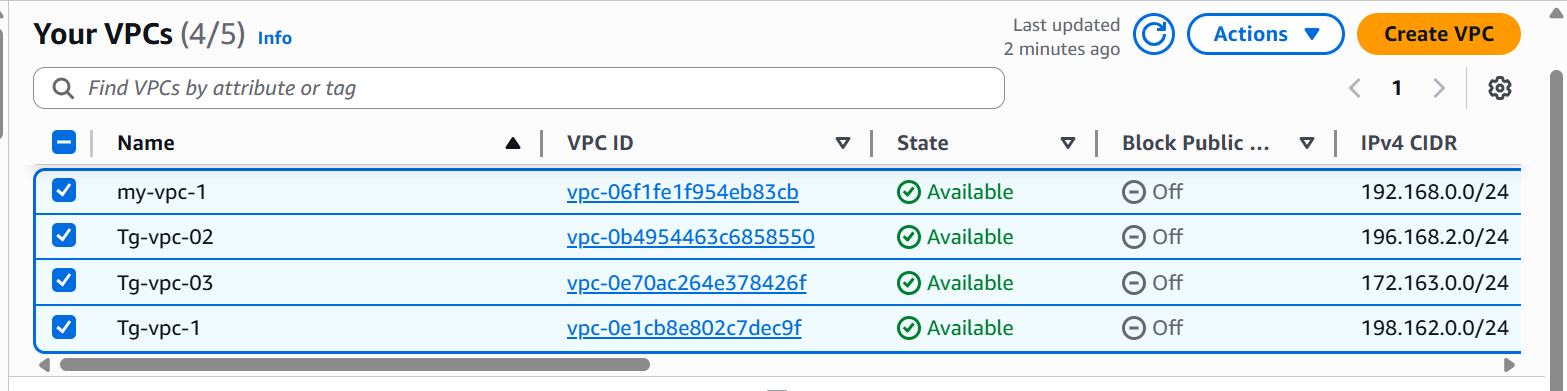
1. Set up a VPC Transit Gateway  
   * Go to **VPC console → Transit Gateways → Create Transit Gateway**.
   * Give it a name, set **Amazon ASN** (default is fine), and create it.



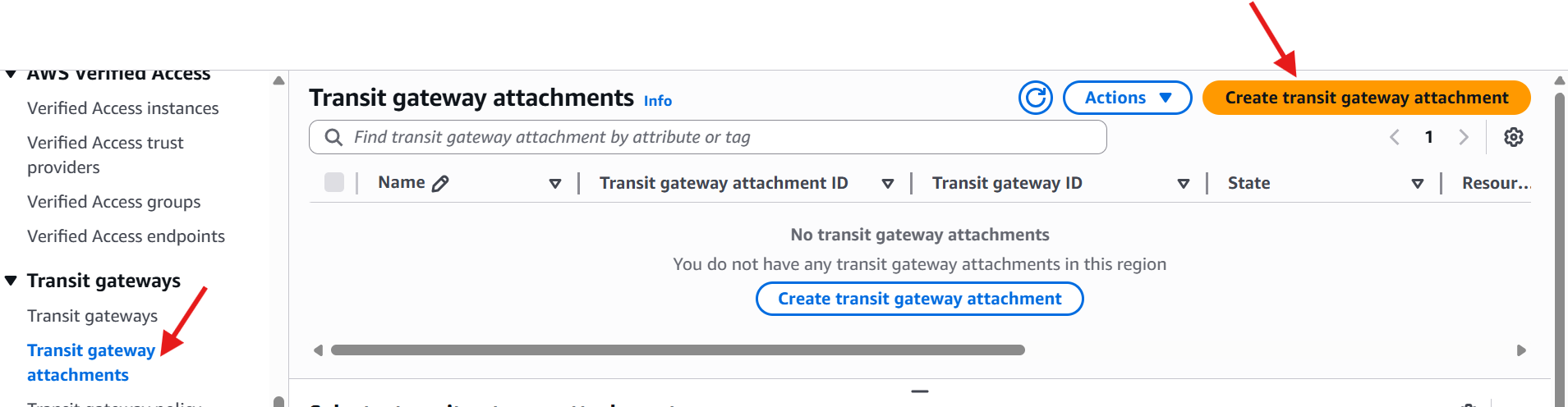


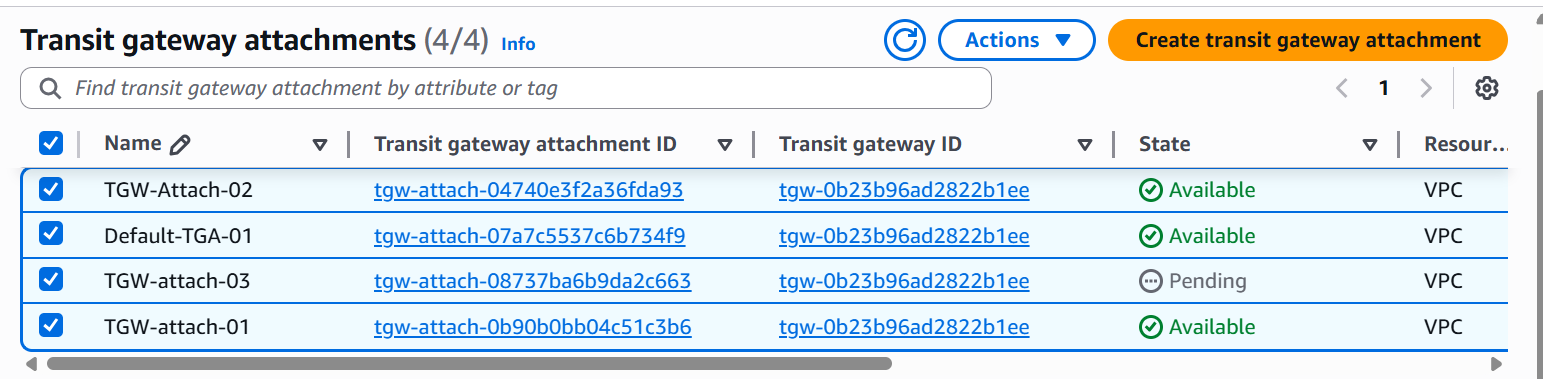


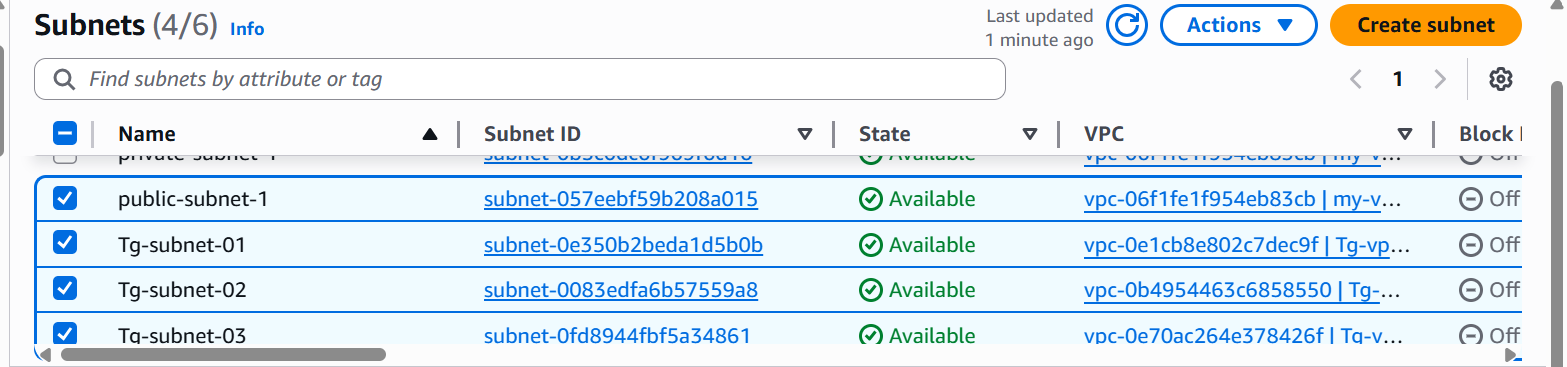
* + Create 3 VPC’s and one default



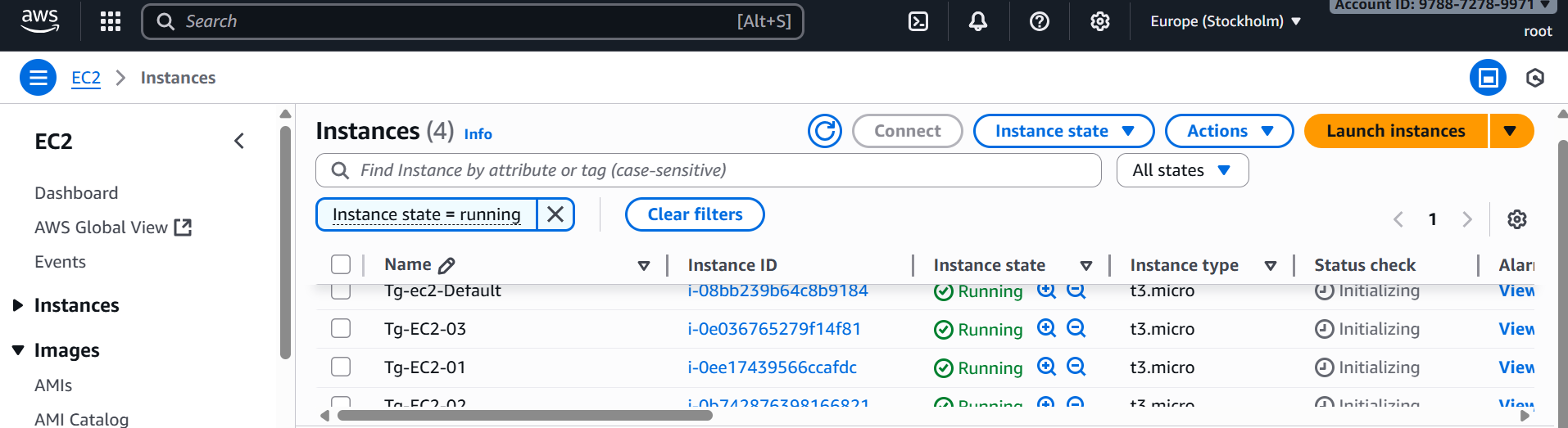
* + Attach each VPC you want to connect.
  + Go to **Transit Gateway Attachments → Create attachment**.
  + Choose the **Transit Gateway** you created and select the **VPC + subnets**.



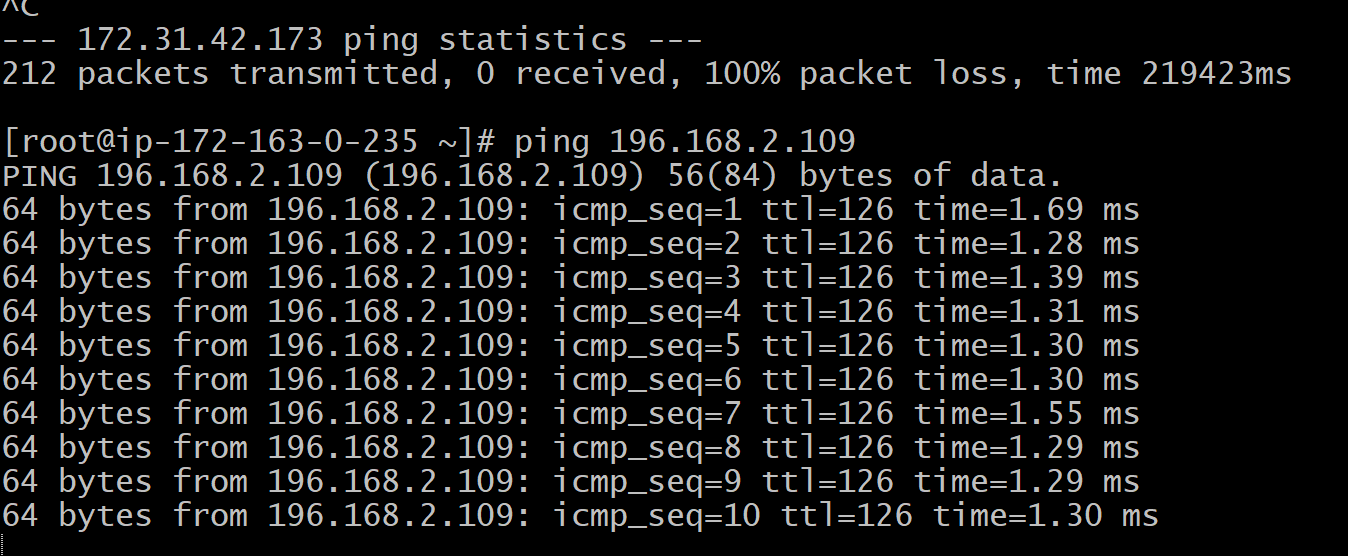




* + Launch 4 EC2 instances in both VPCs

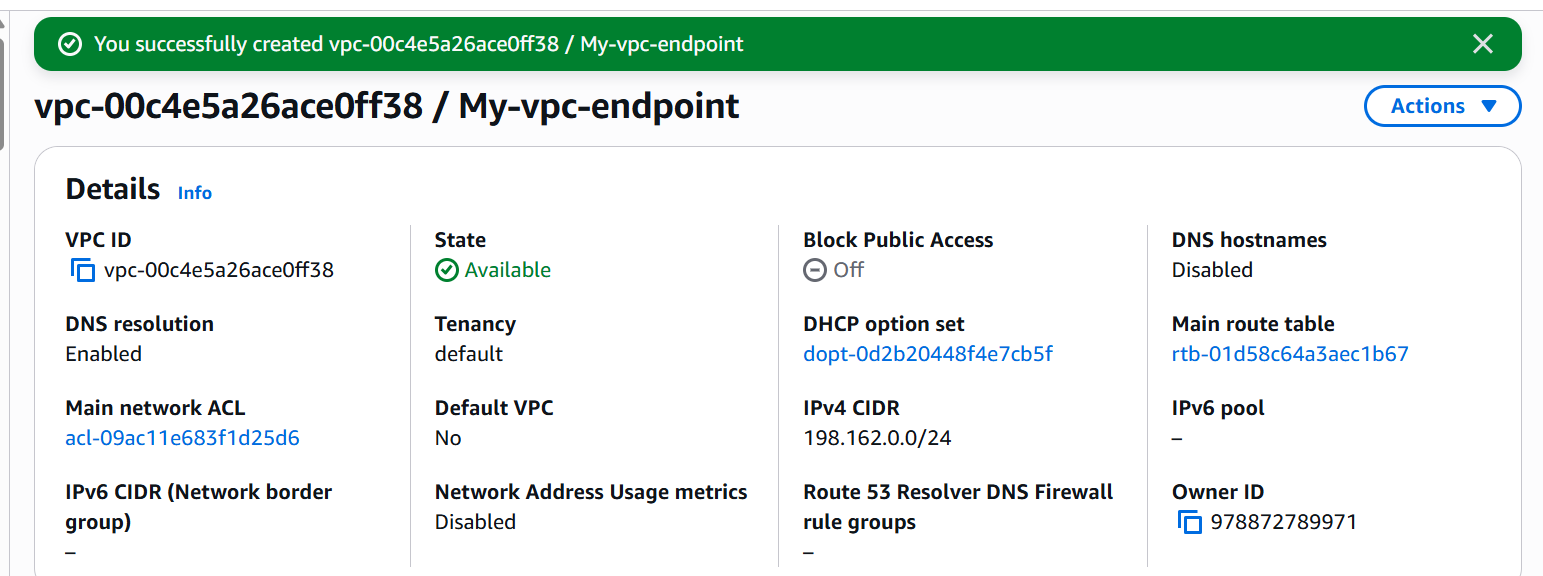


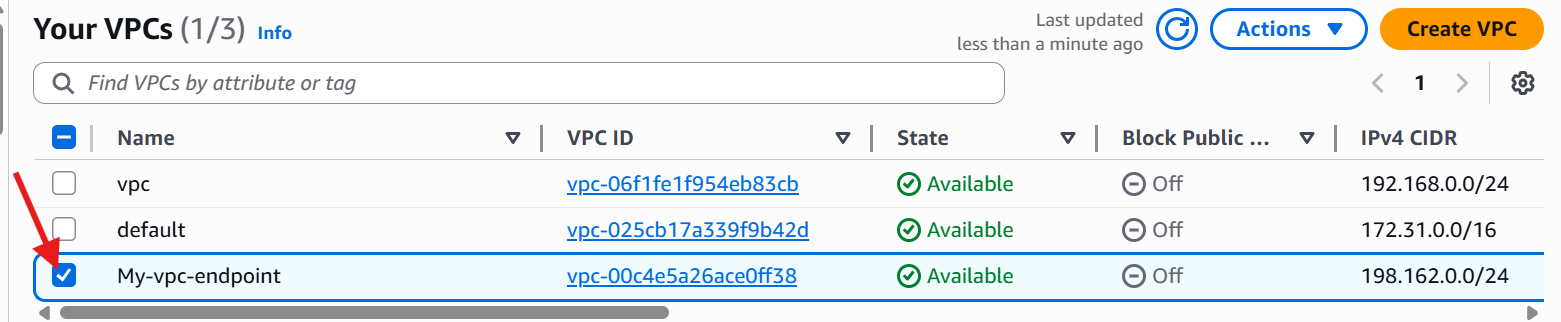
* + Try pinging or connecting using private IPs to confirm the setup

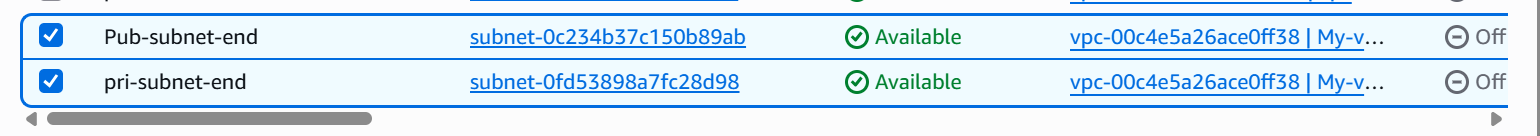


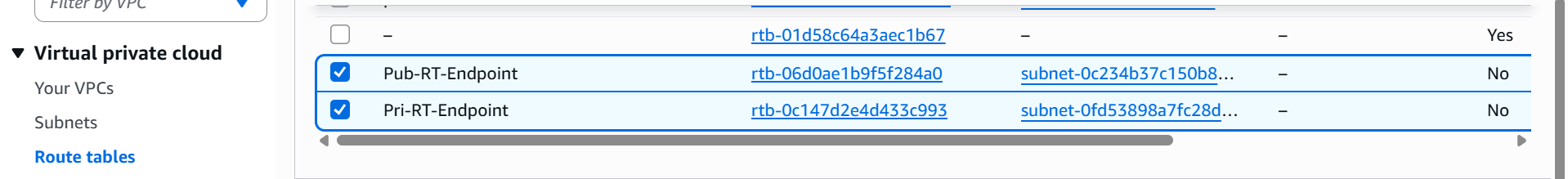
1. Set up a VPC Endpoint

* Create vpc **my-vpc-endpoint**
* Create subnets and attach to vpc
* Create route tables and added subnet associations and routes

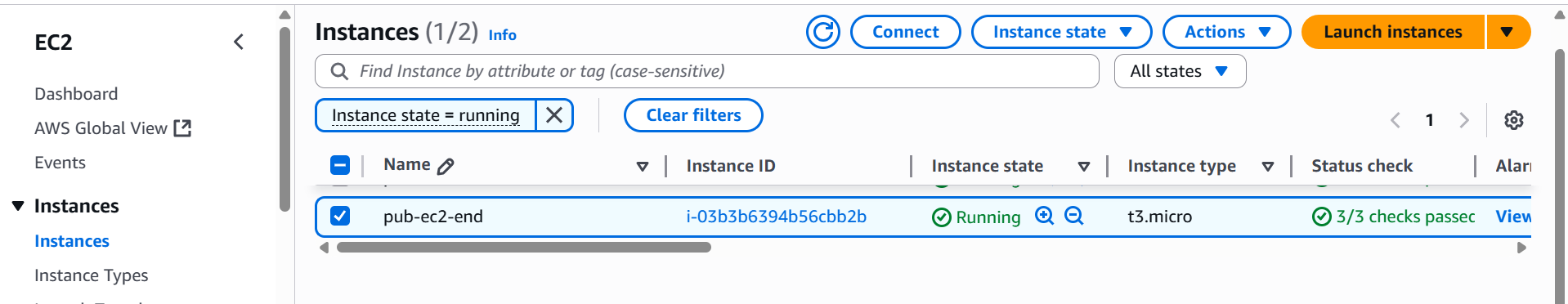


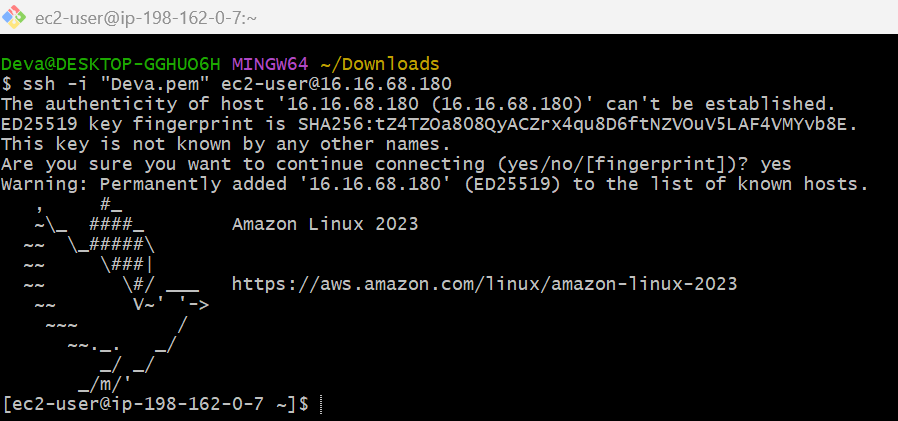




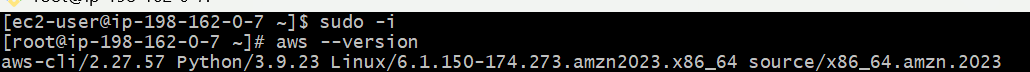


* Launch one instance
* Copy the public ip and connect to gitbash

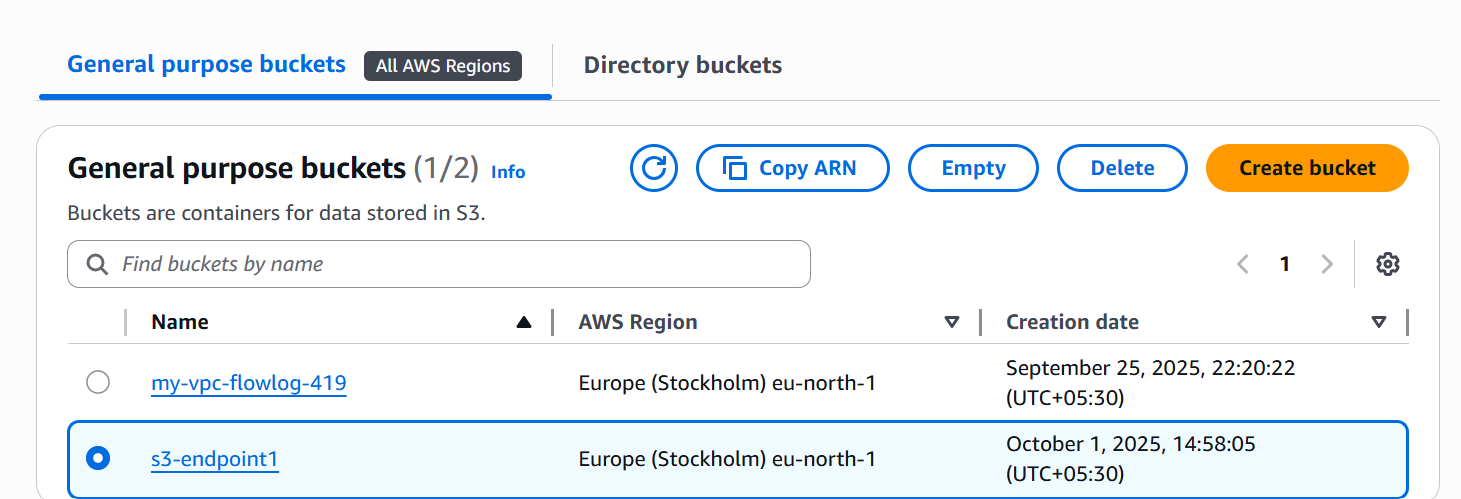




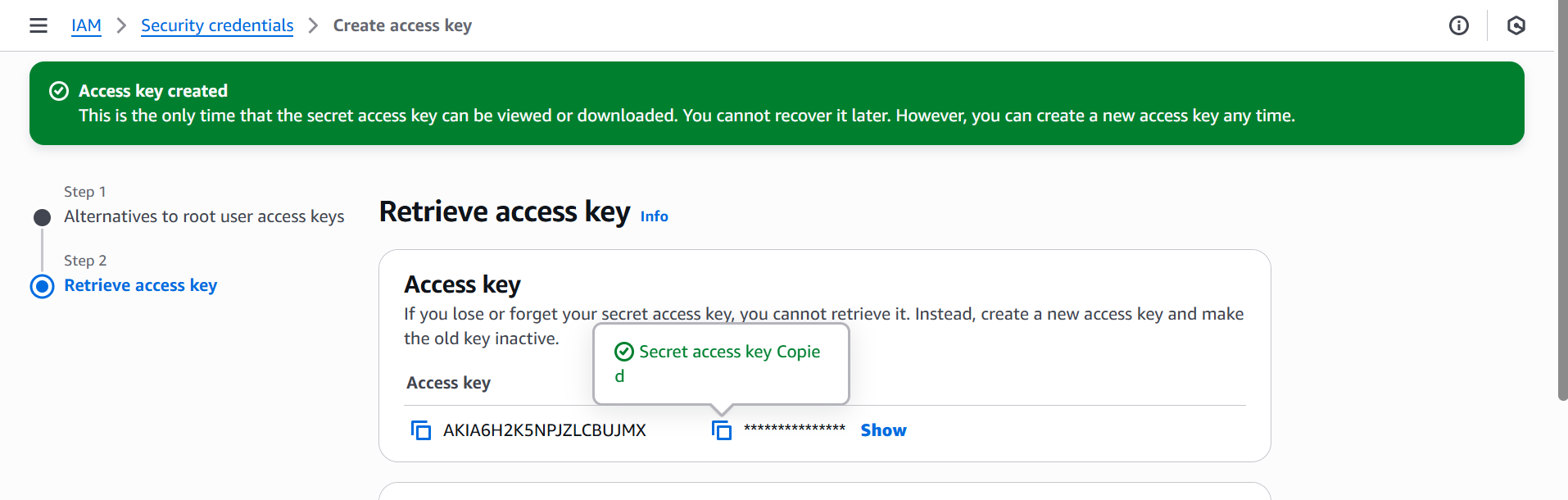
* Go to root user
* Give aws –version - It shows aws version



* Create s3 bucket



* Go to IAM and create access key



* Go to git bash and give aws configure
* Enter the access key and password
* Give region and output format**(json)**
* Change directory to home **cd ~**
* And give command aws s3 ls – it shows your buckets

