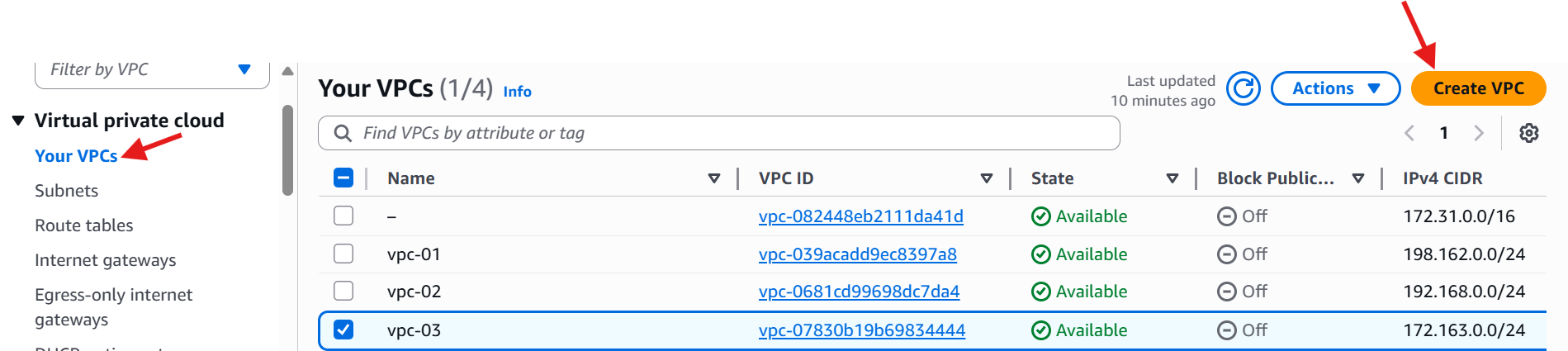
**VPC Challenges**

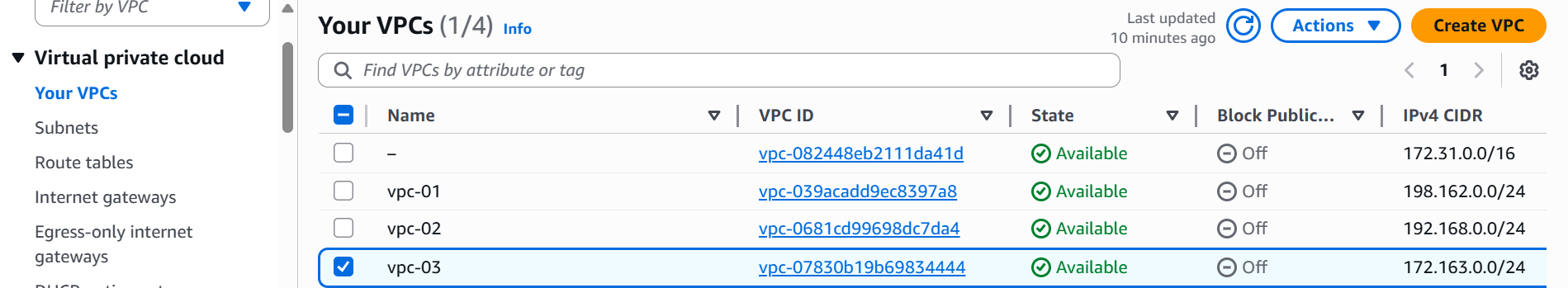
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Use Case: Setting up Transit Gateway and VPC Endpoints for a Multi-VPC Architecture  
Scenario:   
A large organization is migrating its on-premises infrastructure to the AWS cloud.  
The organization's architecture involves multiple VPCs for different departments and applications, each requiring secure communication with centralized services and external resources.  
The IT team needs to design and implement scalable and efficient network architecture to accommodate the organization's growth and ensure robust connectivity between VPCs and external services.**

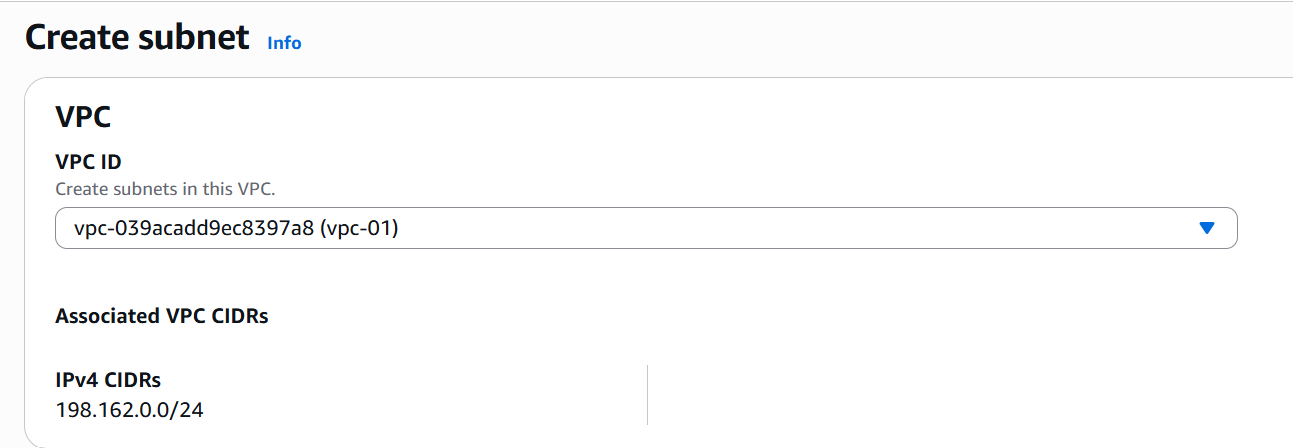
**Objectives:**

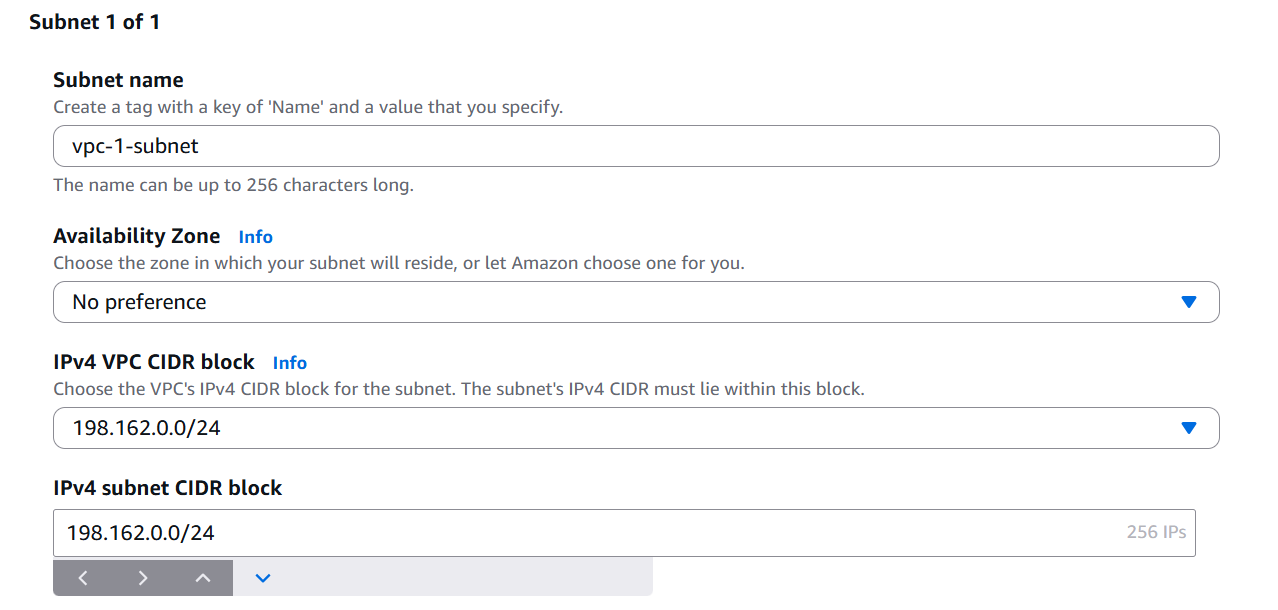
* **Design and deploy a scalable network architecture using AWS Transit Gateway to simplify network connectivity between multiple VPCs.**
  + - Go ec2 console and in search bar search **VPC**
    - Click on VPC , Vpc console open
    - Click on **vpc** and click on **create VPC**
    - Create 2 public vpc’s and 1 private vpc

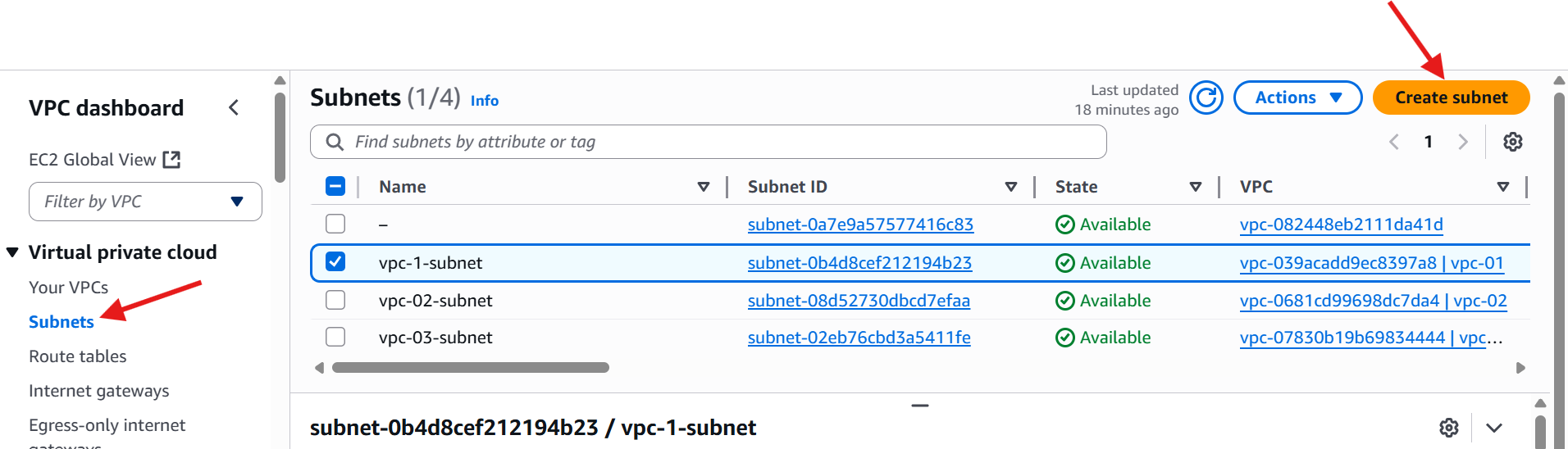


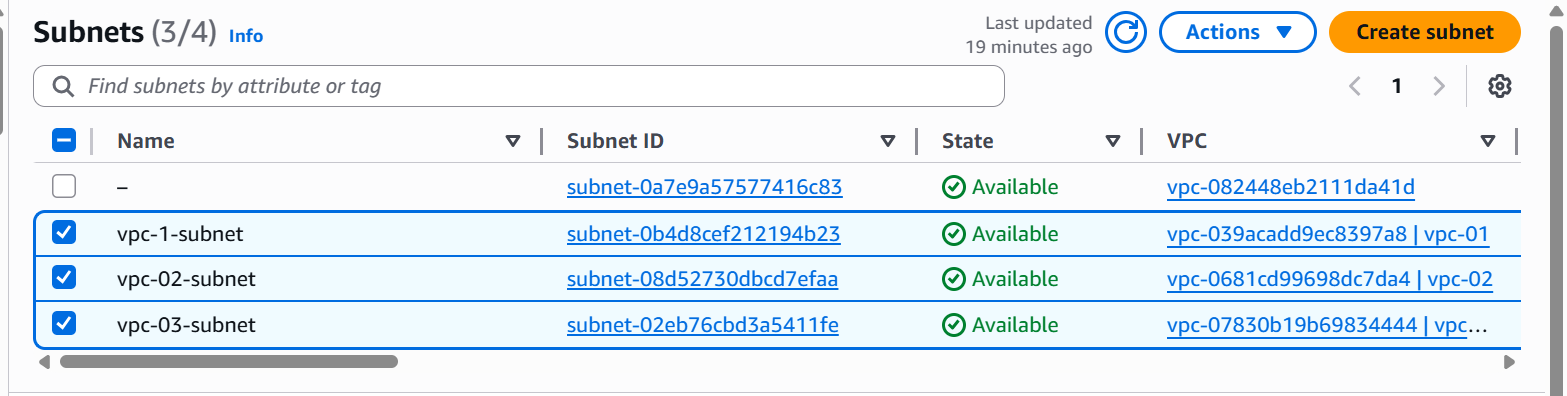


* + Click on subnets
  + And click on create subnets
  + Select **VPC ID**
  + And give subnet name
  + Give **IPV4 Vpc CIDR Block**
  + Click on **create subnet**

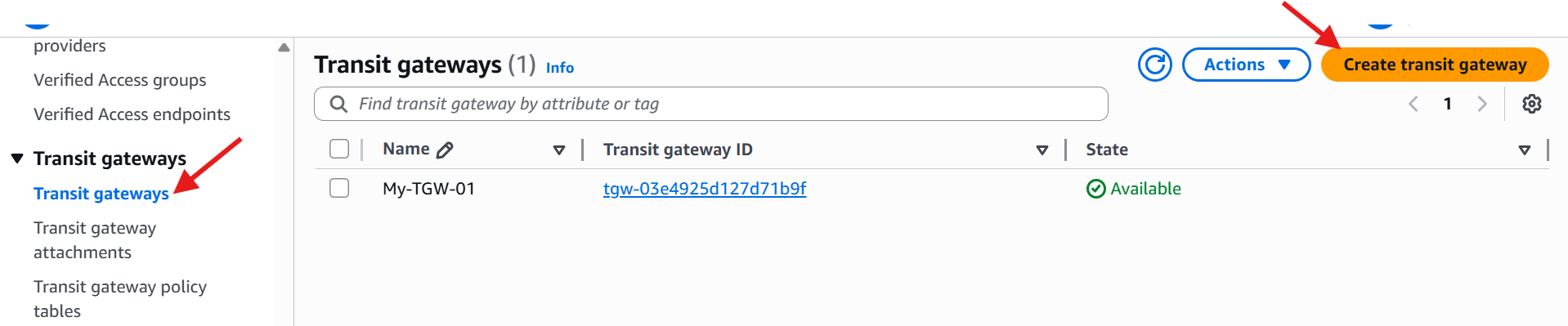


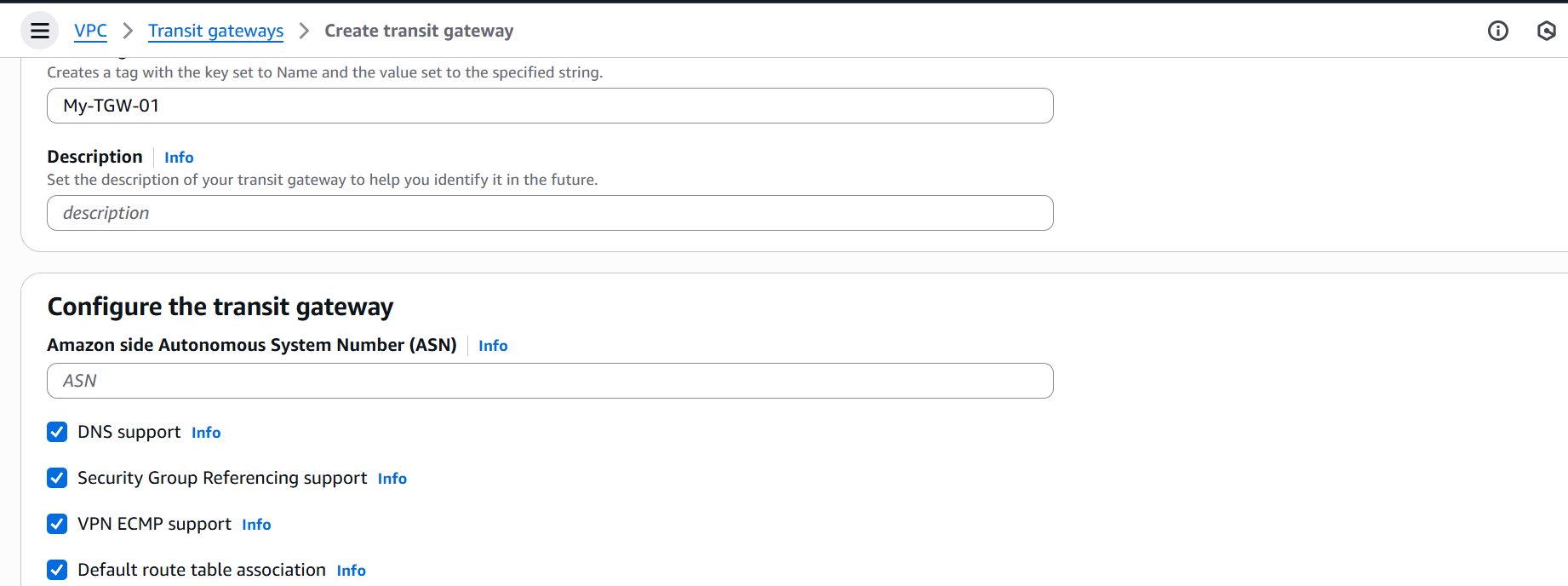


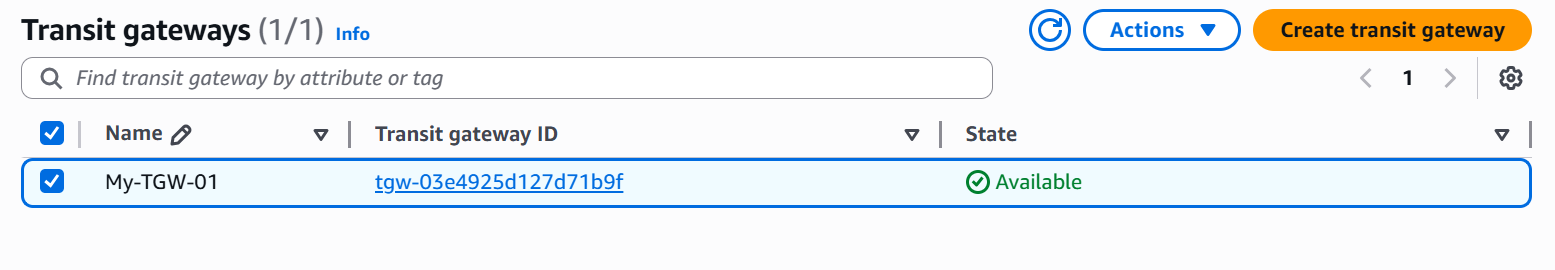




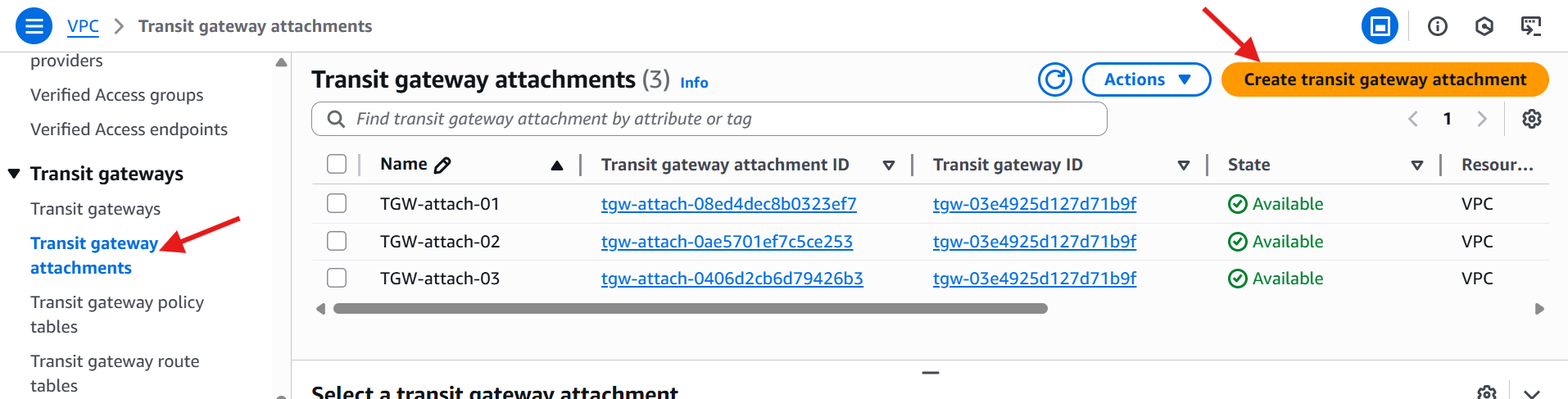
* + Click on **Transit gateway**
  + Click on crate **Transit gateway**



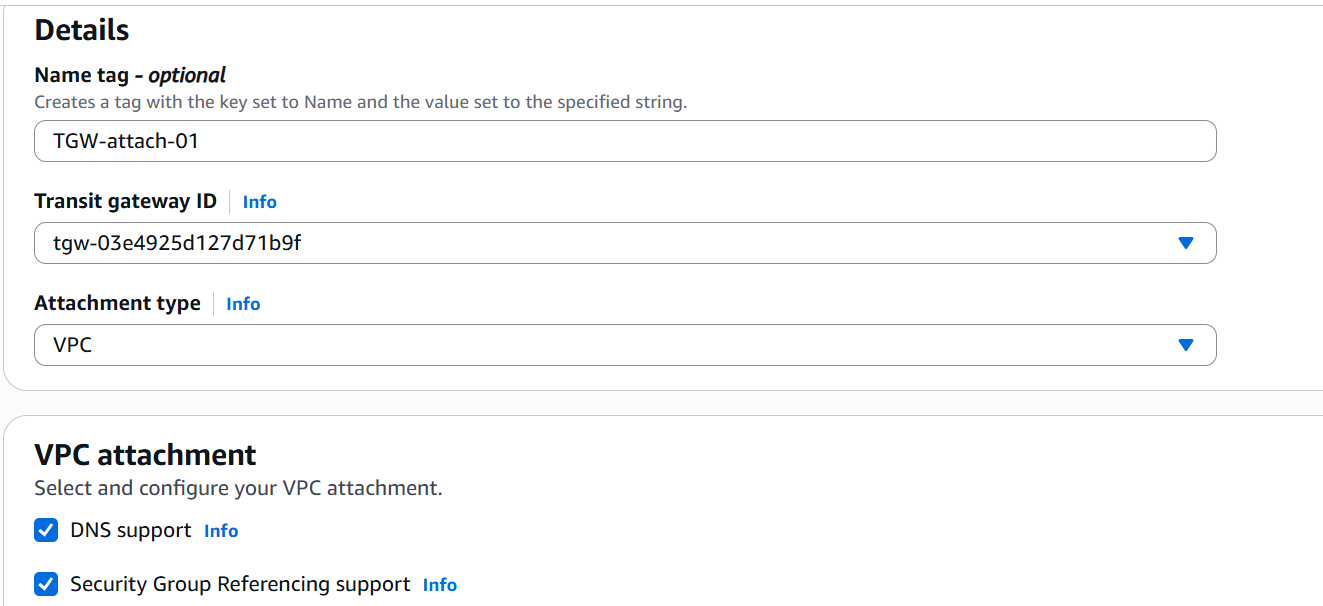




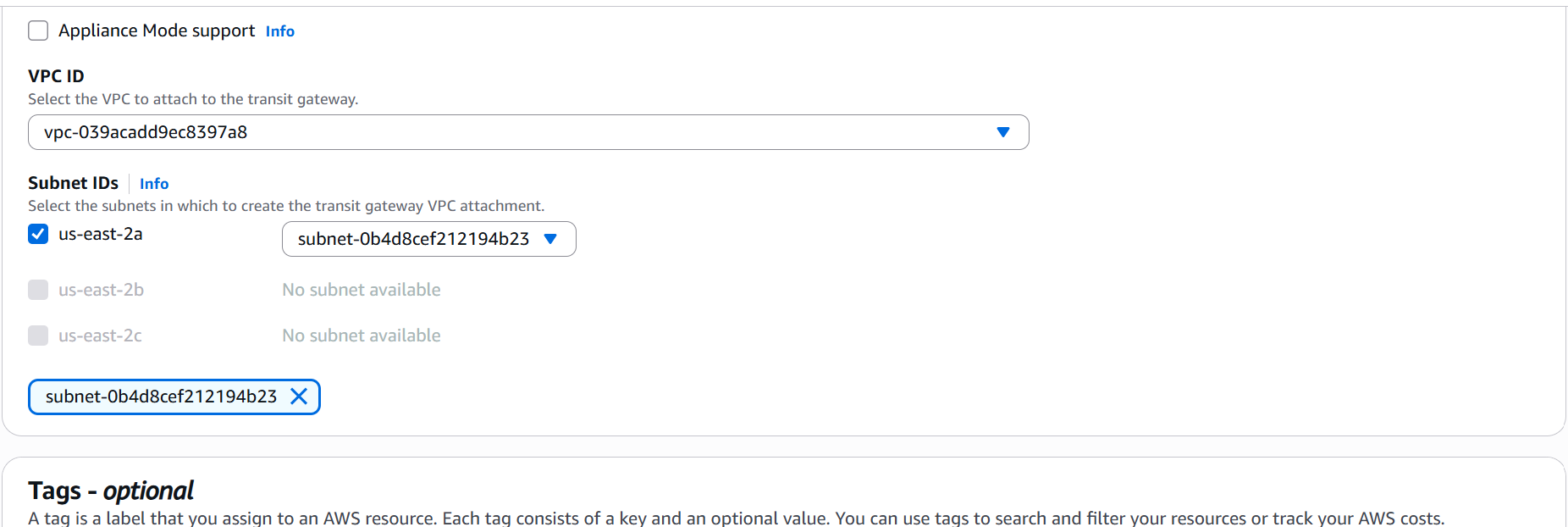
* + Click on **transit gateway attachments**
  + Click on **create transit gateway attachments**

****

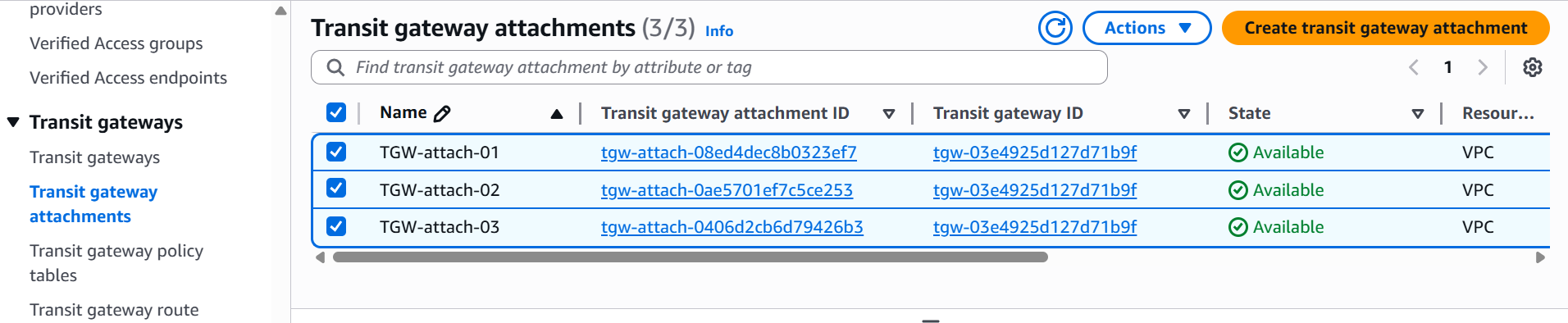
* + Give name **TGW-attach-01**
  + Give transit gateway id

****

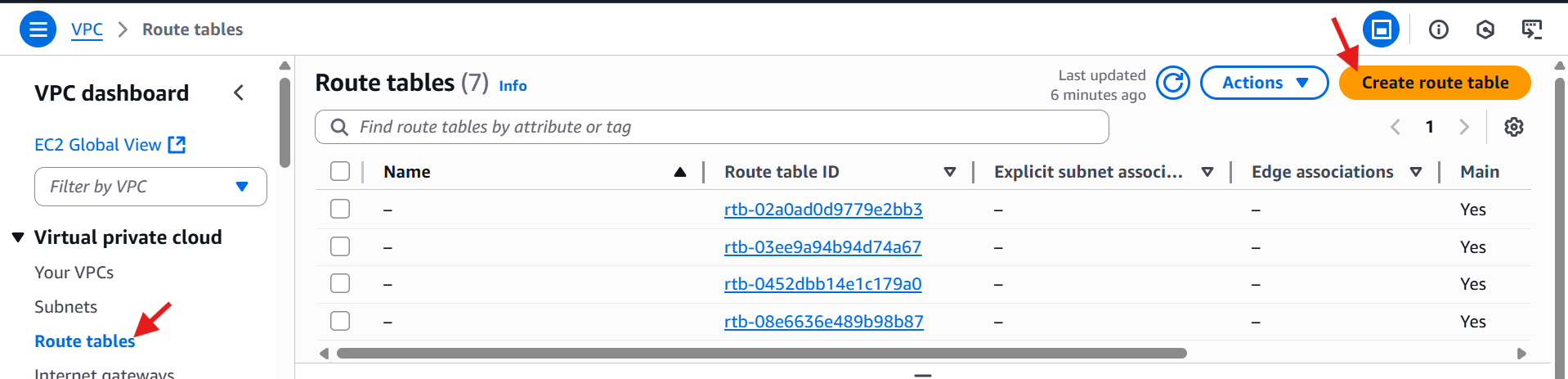
* Select the vpc for which vpc you want to attach
* And Click on **create transit gateway attachment**

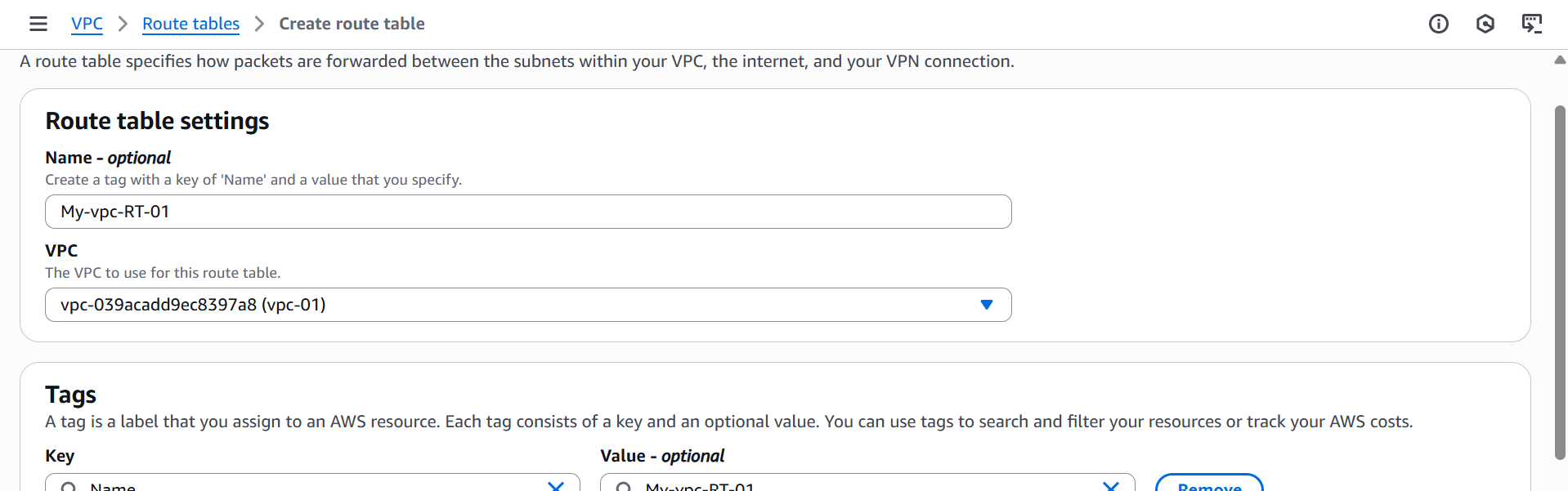


* In this way we have to create transit gateway attachments and give **transit gateway ID** and **VPC ID**  For other vpc’s.

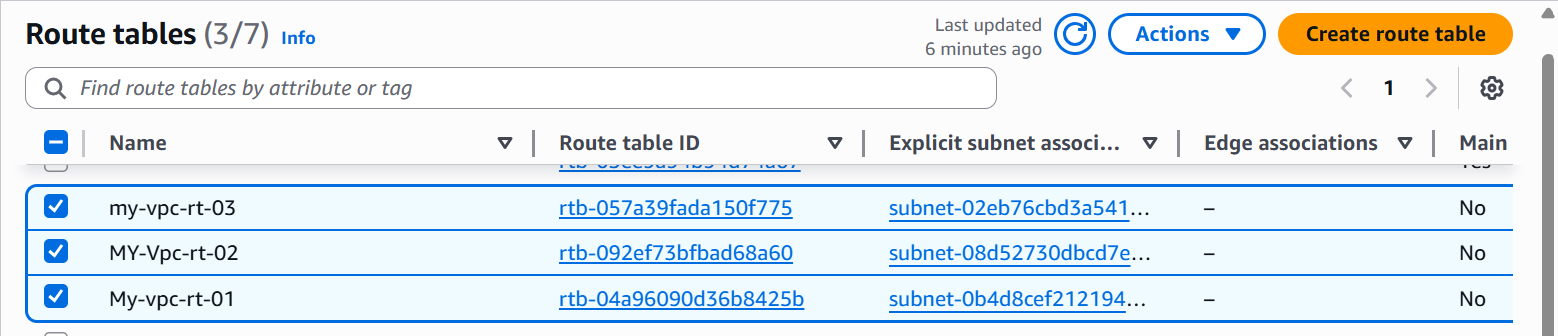
****

* Click on **Route tables**
* And click on **create route table**
* Give route table name and select vpc
* And click on create route table

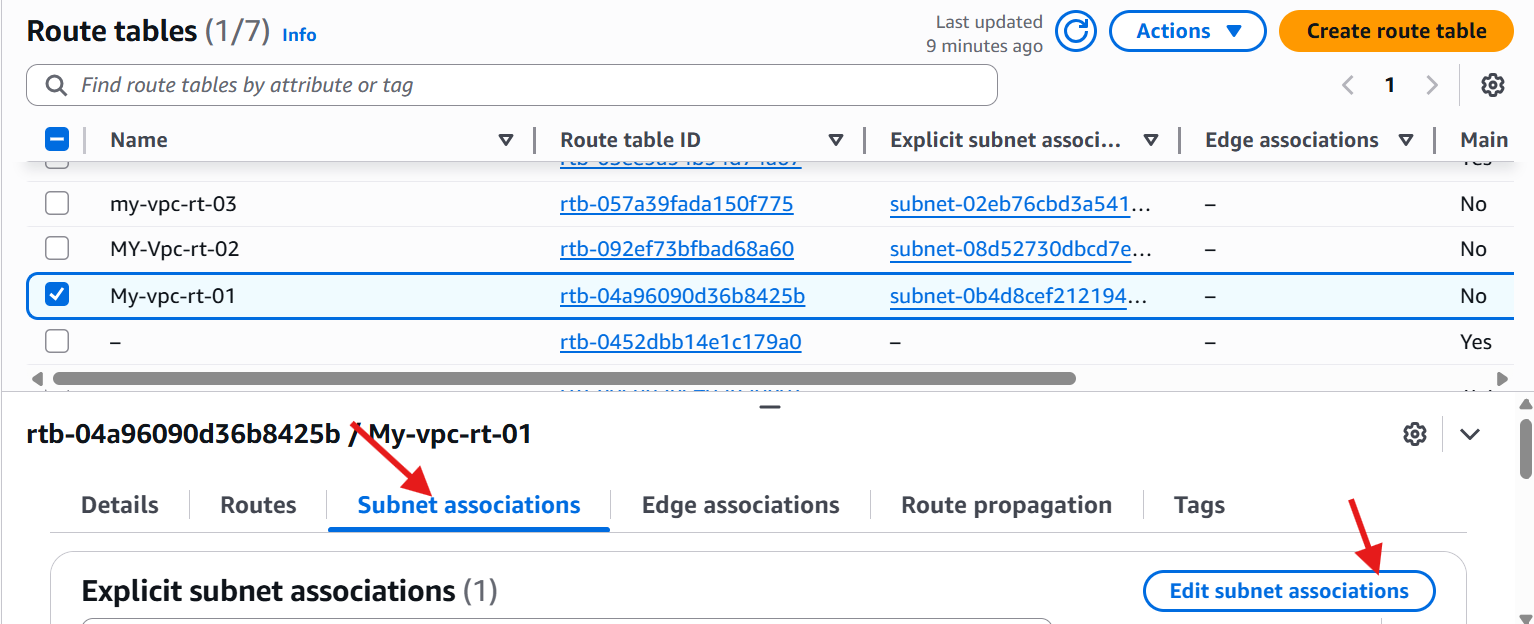
****

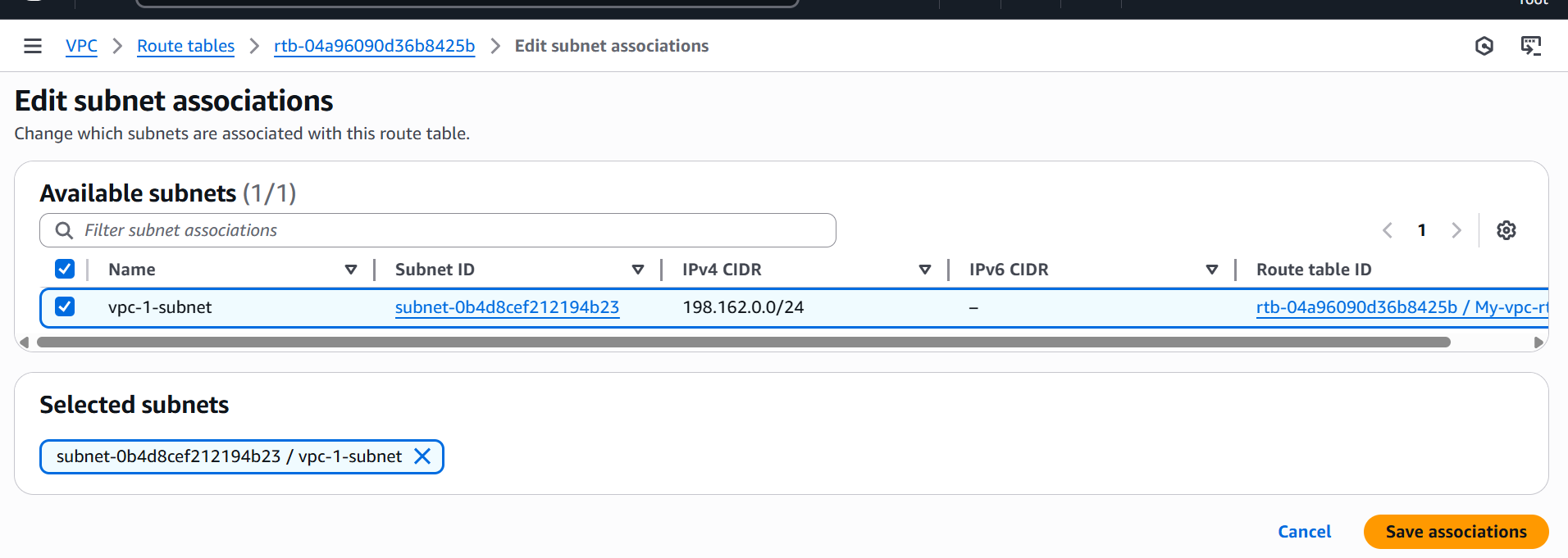
****

* In this way we have to create 3 Route tables for 3 vpc’s

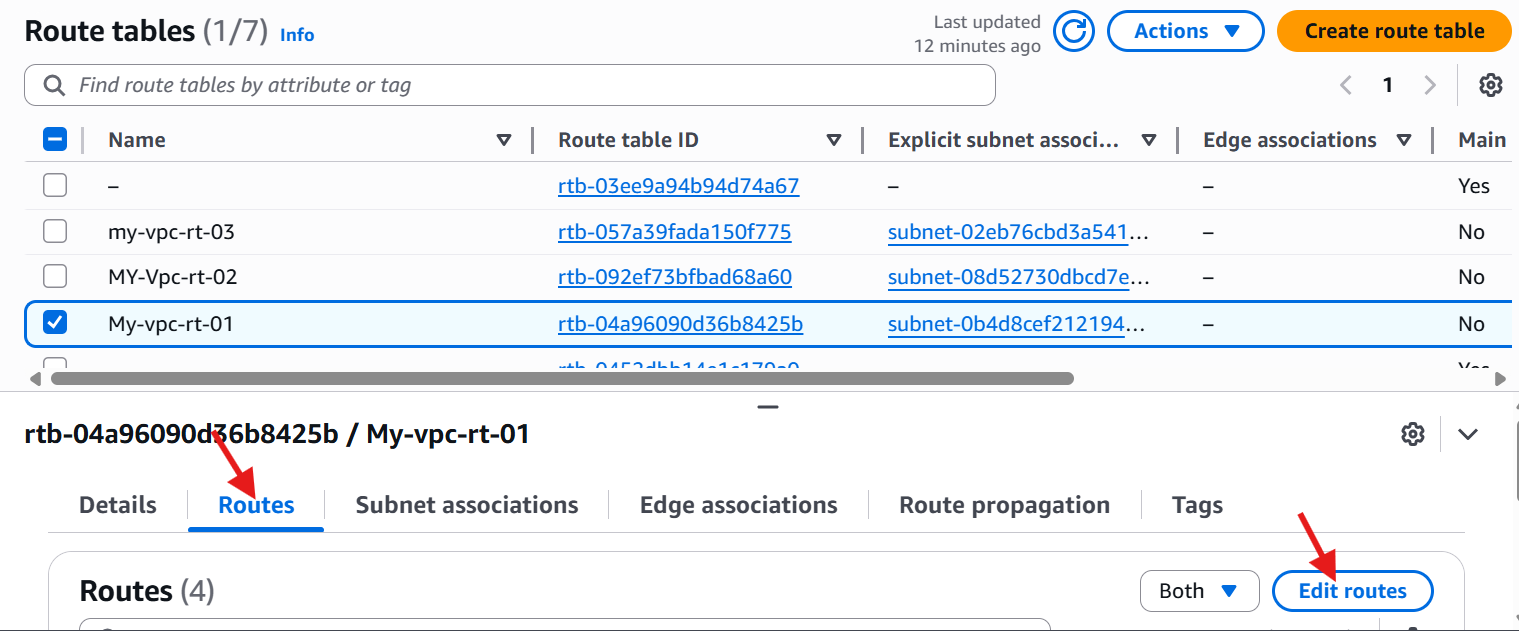
****

* And select route table and click on subnet associations, and add the subnets for each Route table.

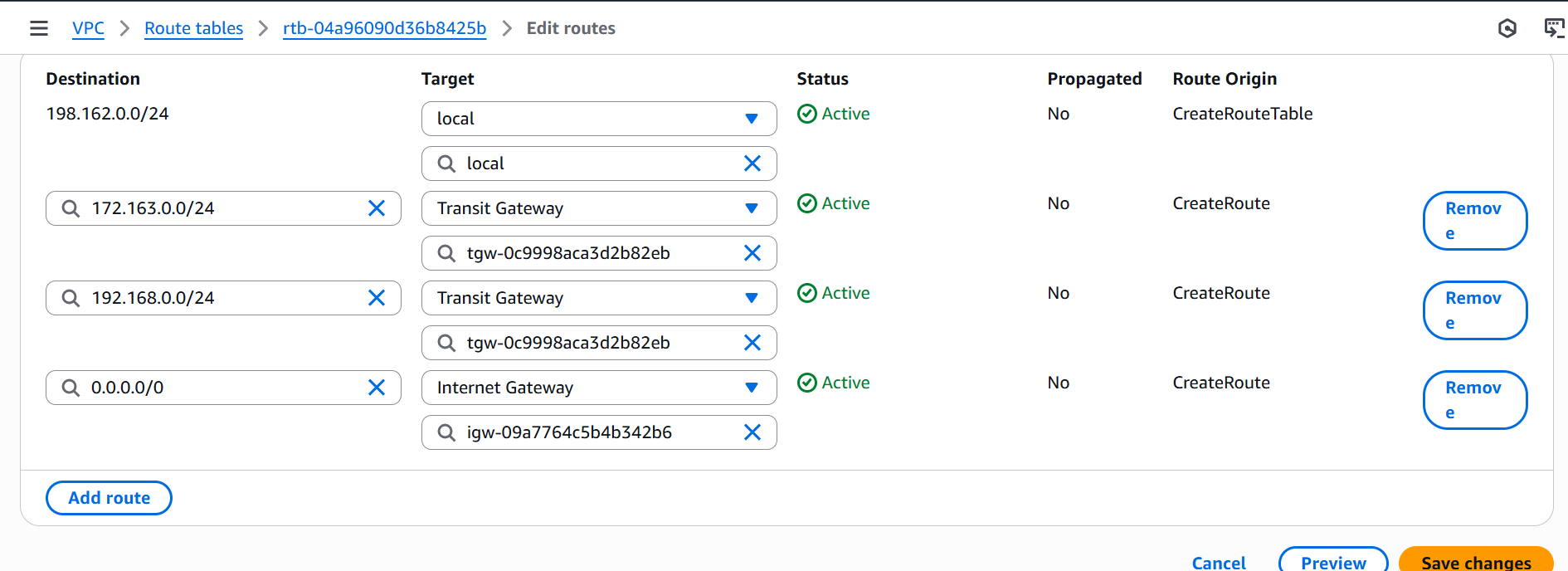
****

****

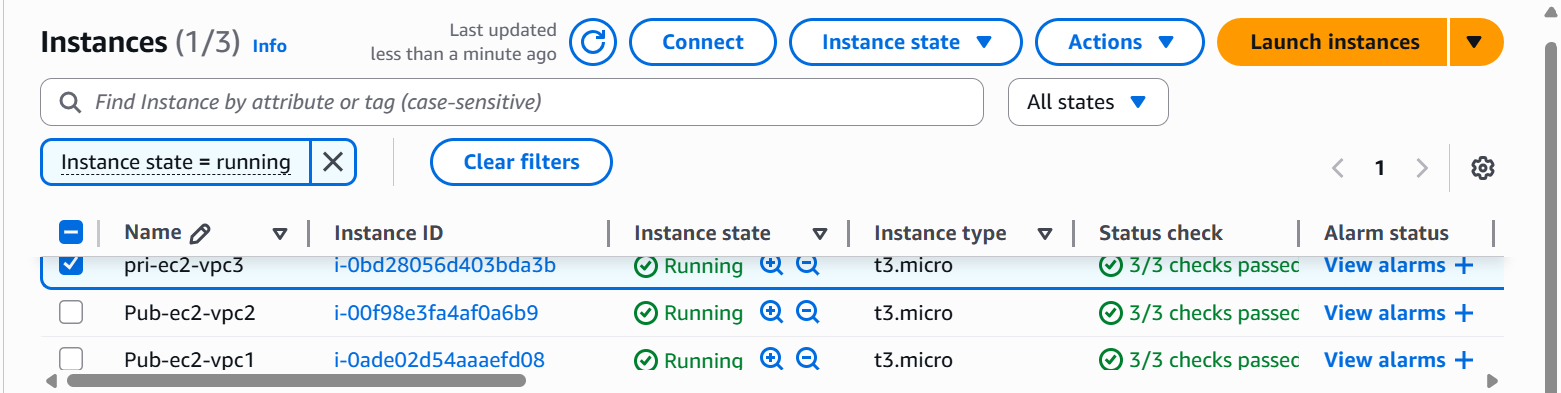
* In this way we have to add subnet associations for each route table.
* And click on **routes** and click on **edit routes**

****

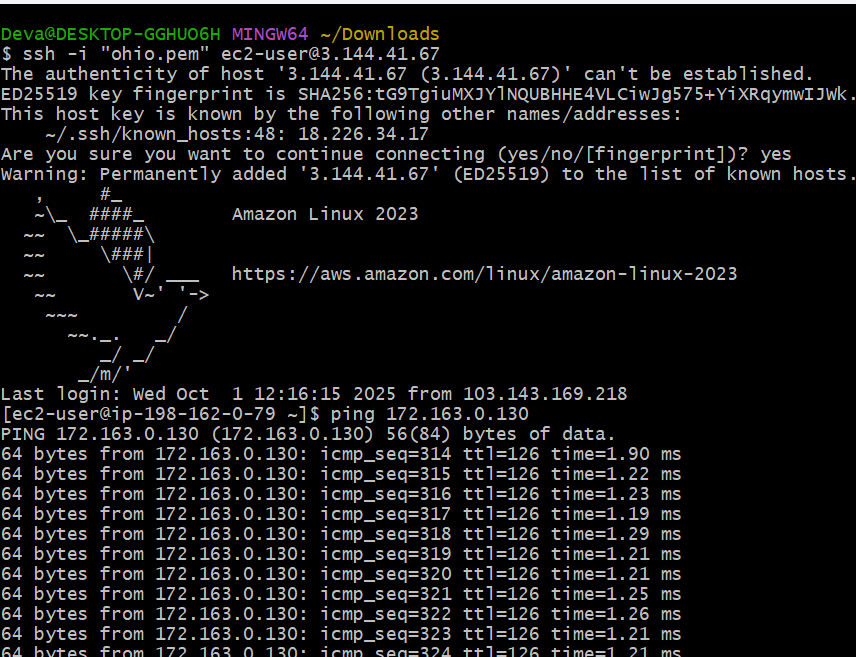
* Add internet gateway and add other two vpc,s CIDR range and attach transit gateway.
* Click on **save changes**

****

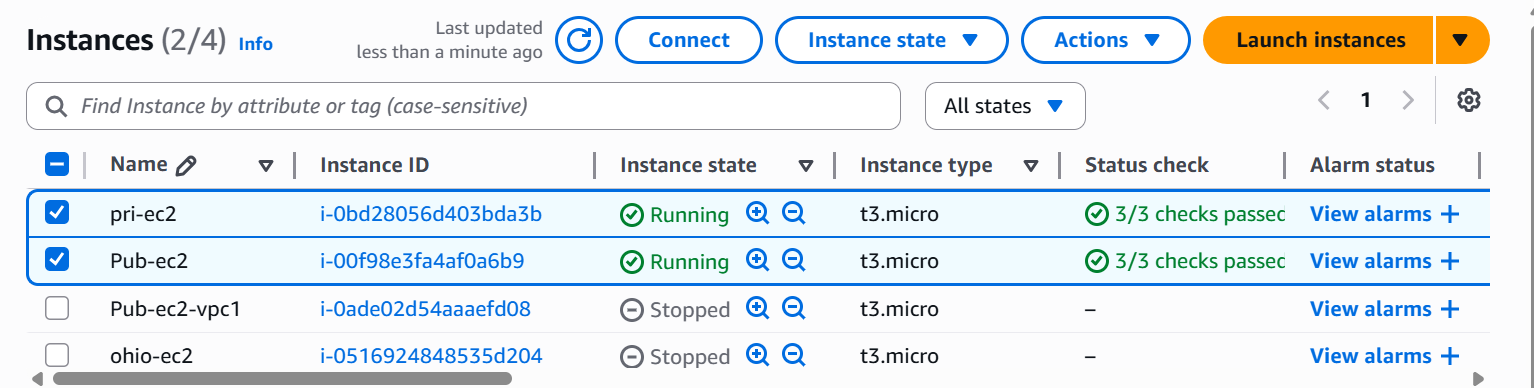
* In this way we have to **edit routes** for **RT-01 & RT-02** and **RT-03** route table is private, so we don’t need to attach internet gateway.
* Go to **ec2** and launch **2 public instances** and **1 private instance.**



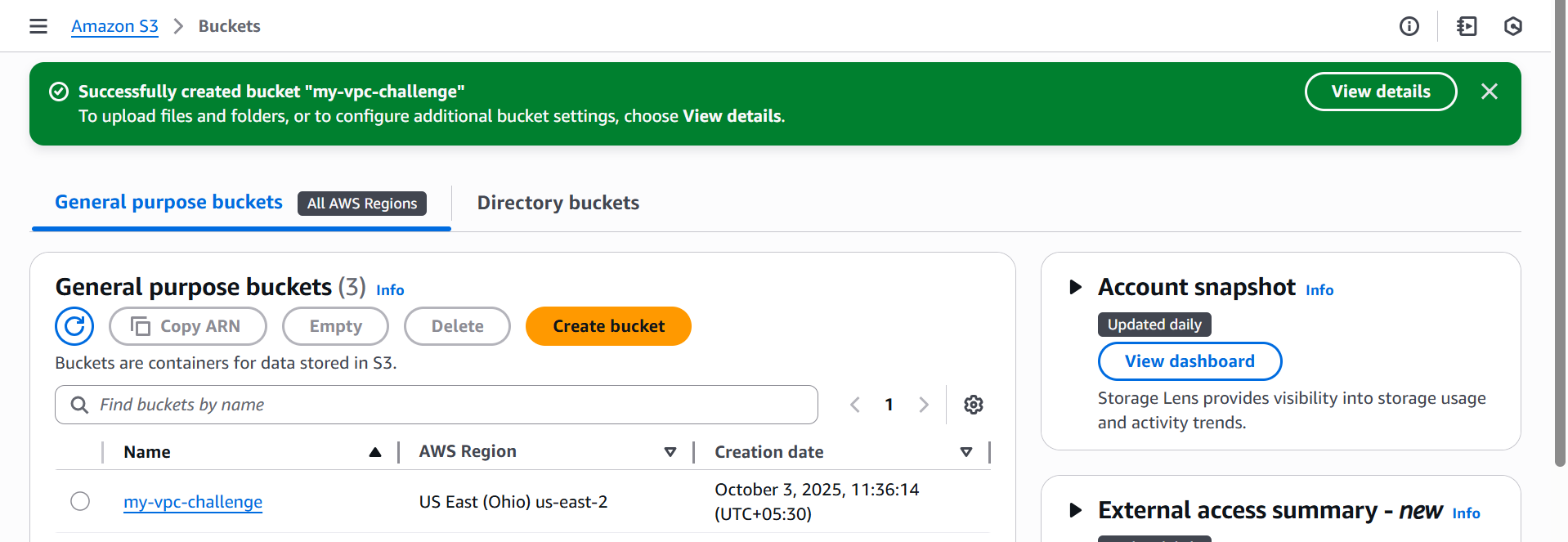
* Open gitbash and connect ssh with the public instance and ping private ip address of private instance .



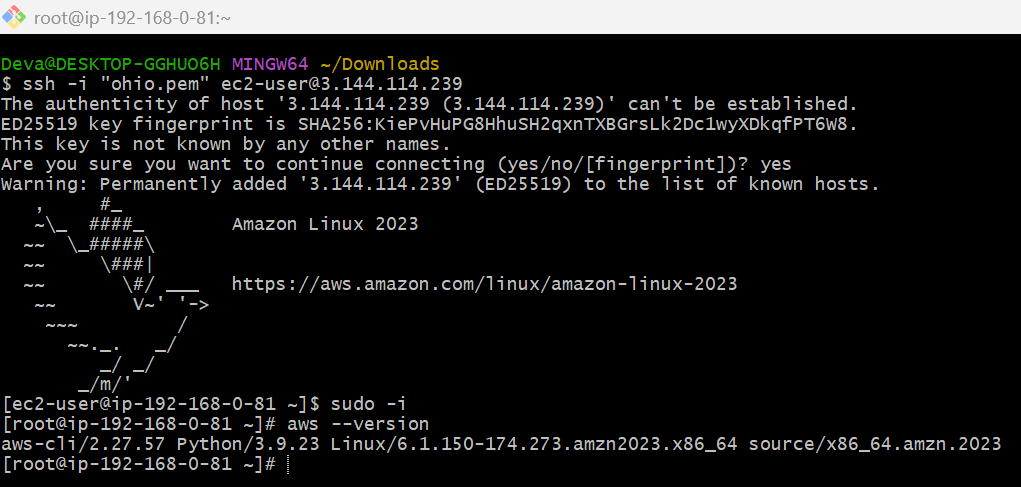
* Configure VPC endpoints to securely access AWS services without internet gateways or NAT gateways, ensuring data privacy and minimizing exposure to external threats.
* Launch one public instance and private instance with public subnet and private subnet.



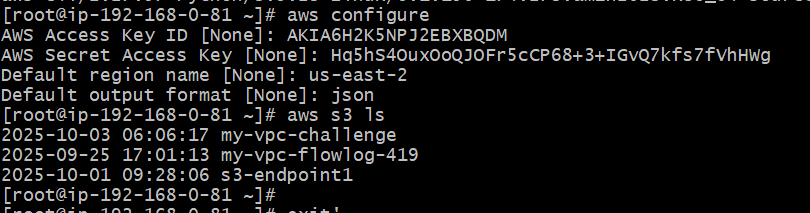
* Go to s3 and click on create bucket
* Create one bucket (**my-vpc-challenge**)



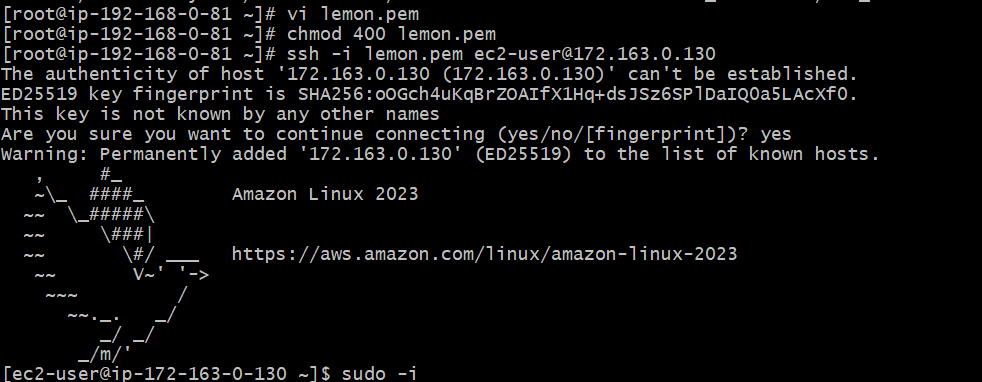
* Connect ssh with the public instance



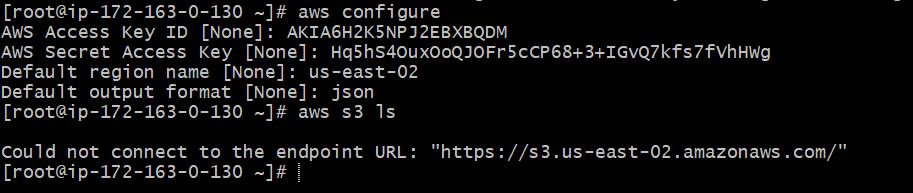
* Go to aws configure and give security key and password



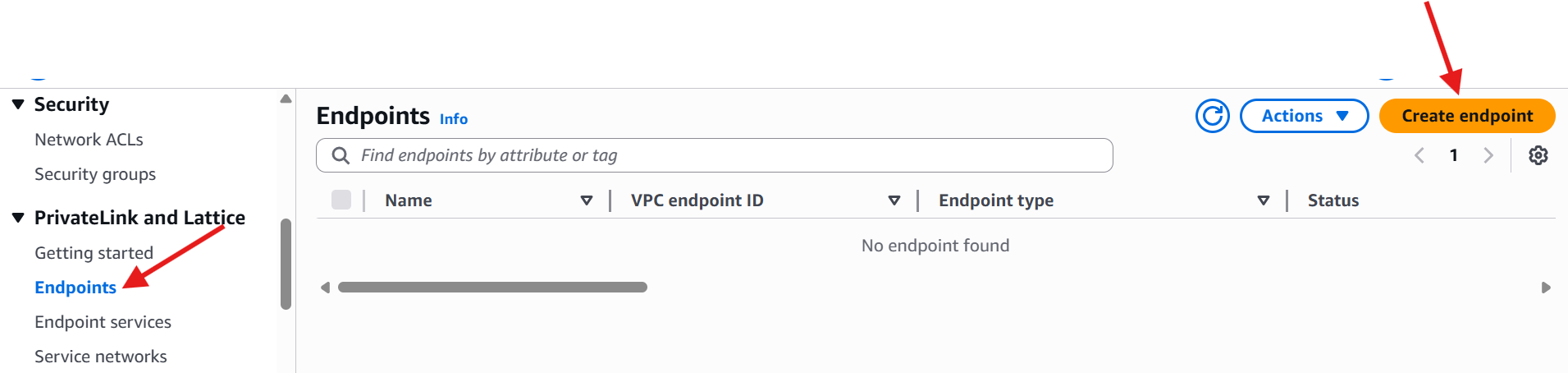
* Open another gitbash tab and give cat pemkey name , it gives secret key.
* Copy the key
* Come to 1st gitbash tab and create and open a file (**vi lemon.pem**) and paste it in that file and **save**.
* Give permissions to that file **chmod 400 file name**.
* And check **ssh -i lemon.pem ec2-user@<private ip>**

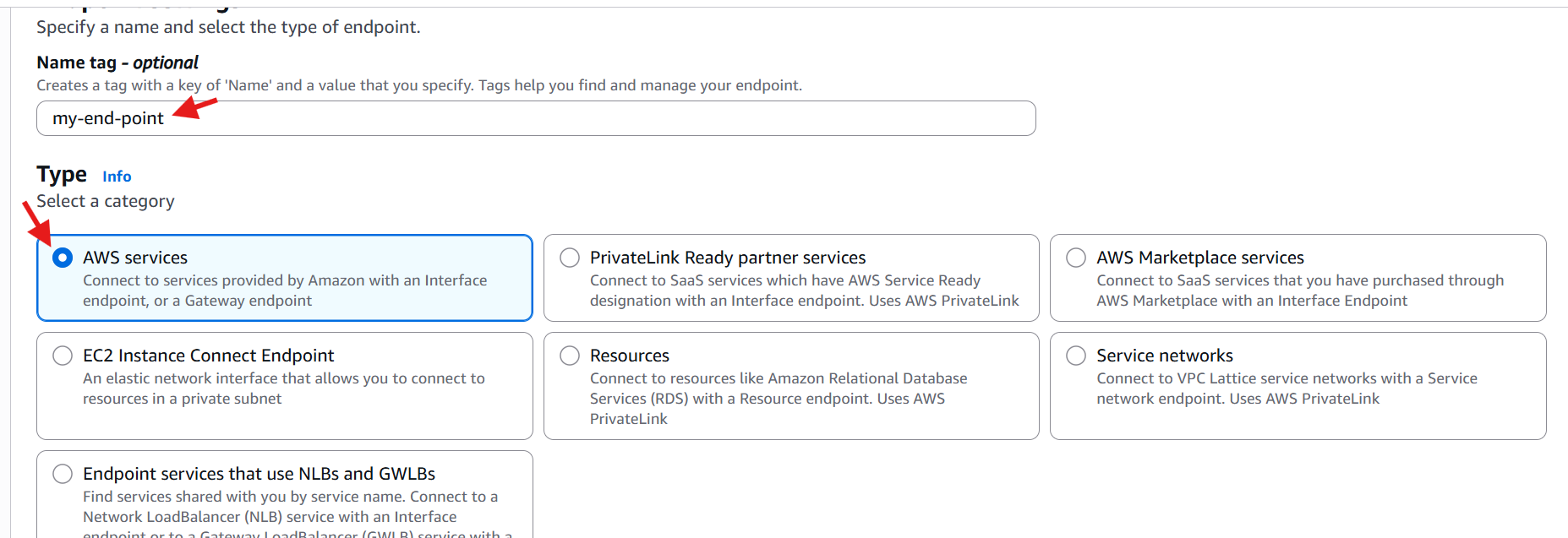


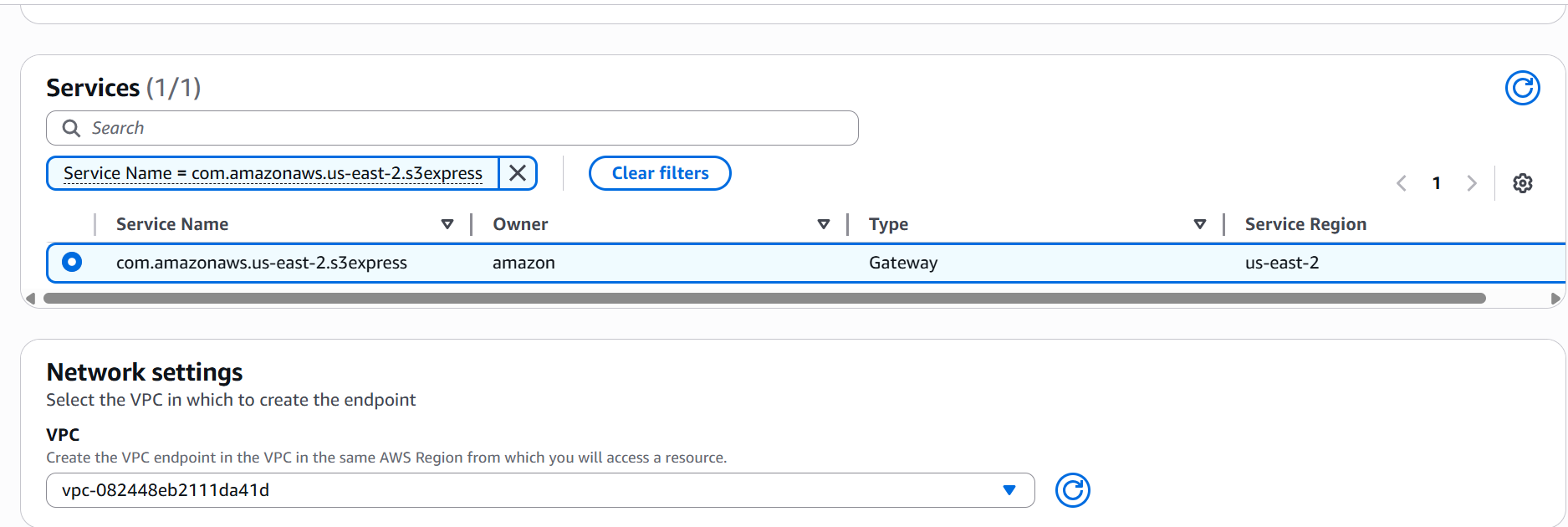
* After give aws configure
* Give access key 🡪secret access key 🡪region 🡪format:(json)

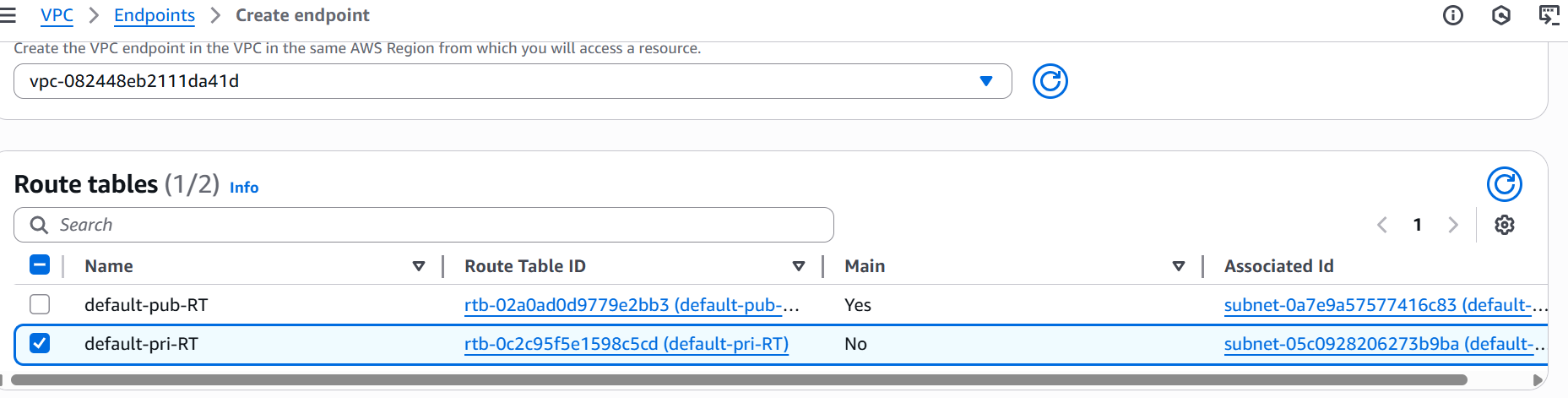


* Go to vpc console
* Click on **endpoints**🡪 click on **create end point**
* Give name **(my-end-point)**
* Select type **AWS services**
* And select services **s3 express** 🡪 **owner**: Amazon, **type**: Gateway , **Region** :us-east-2.
* Select **vpc** and select private route table(**default-pri-RT)**
* Click on **create endpoint**.

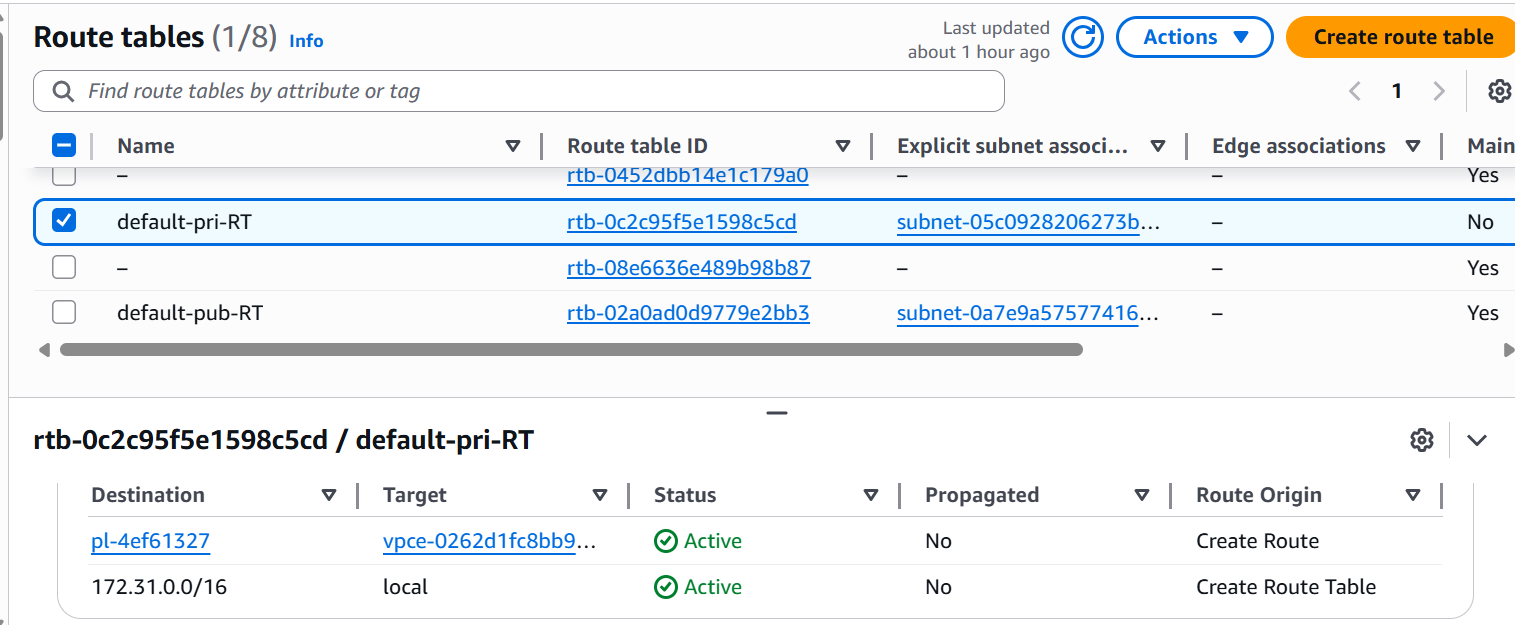








* You can check in private route table , the end point is created for private subnet.



* Now you can access s3 from the private instance: **aws s3 ls**

