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

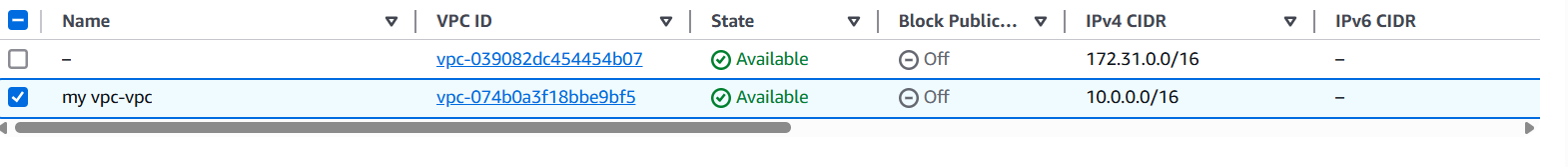
🡺Navigate to the amazon console sign in

🡺Click **Create VPC**.

🡺Choose **VPC and more**.

🡺Configure the VPC:

* **Name tag auto-generation**: Enter a name for your VPC.
* **IPv4 CIDR block**: Specify a CIDR block, such as 10.0.0.0/16.
* **IPv6 CIDR block**: Optional; select if needed.
* **Tenancy**: Choose **Default**.



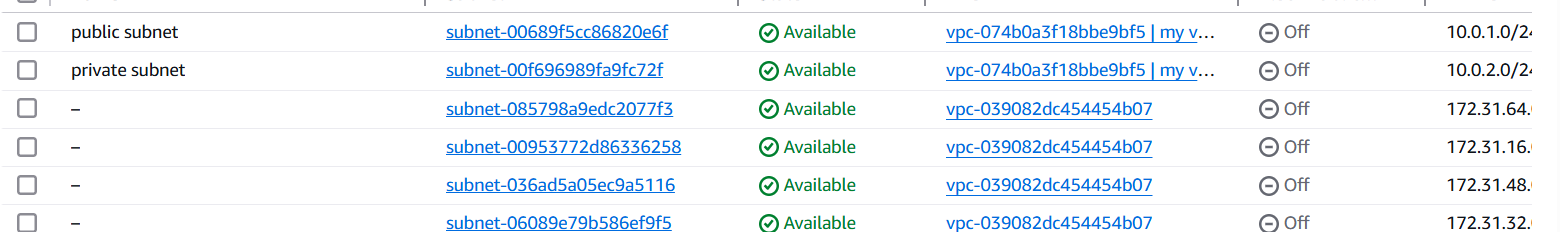
**2. Create Subnets**

1. In the VPC dashboard, select **Subnets**, then click **Create subnet**.
2. For the **Public Subnet**:

* **Name tag**: Enter a name (e.g., PublicSubnet).
* **Availability Zone**: Choose an AZ (e.g., us-east-1a).
* **IPv4 CIDR block**: Specify a CIDR block within the VPC range (e.g., 10.0.1.0/24).

1. For the **Private Subnet**:

* **Name tag**: Enter a name (e.g., PrivateSubnet).
* **Availability Zone**: Choose an AZ (e.g., us-east-1b).
* **IPv4 CIDR block**: Specify a CIDR block within the VPC range (e.g., 10.0.2.0/24).



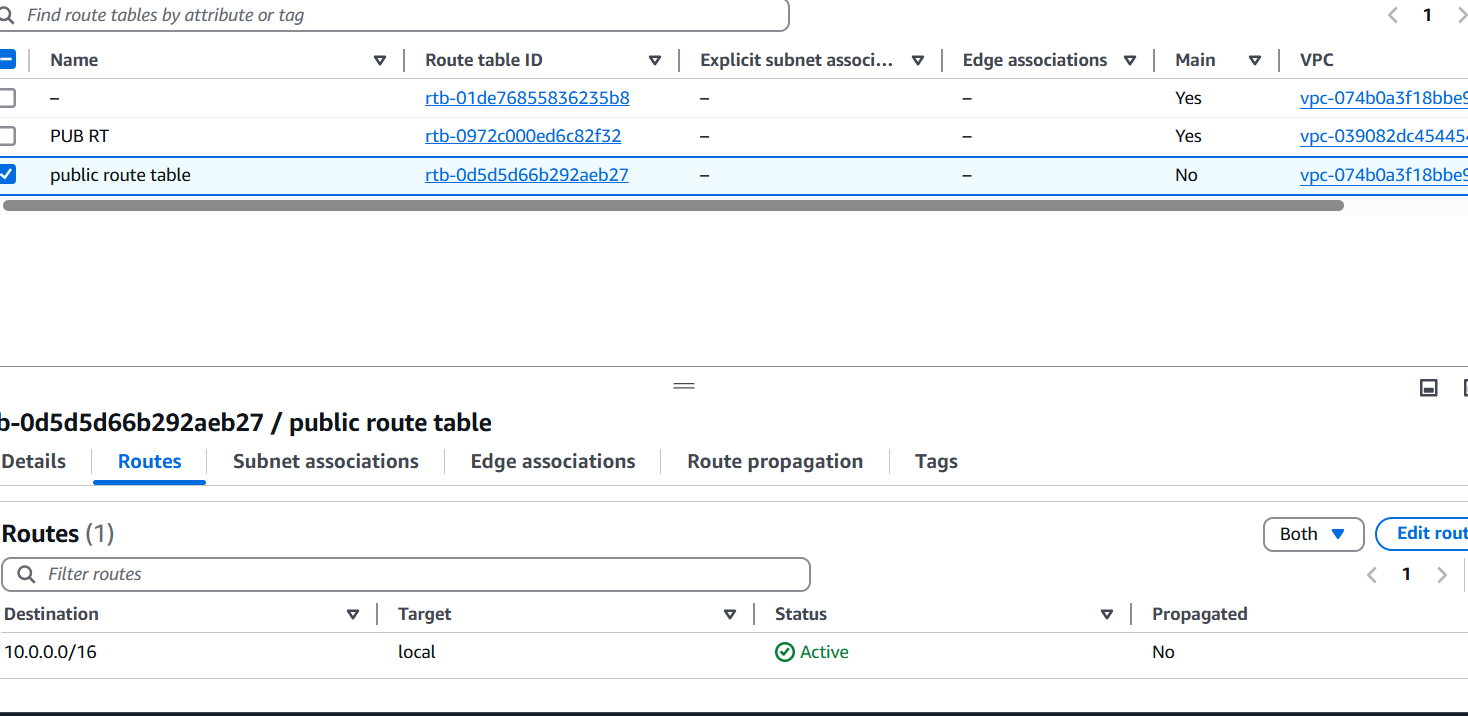
**3.Create and Attach an Internet Gateway**

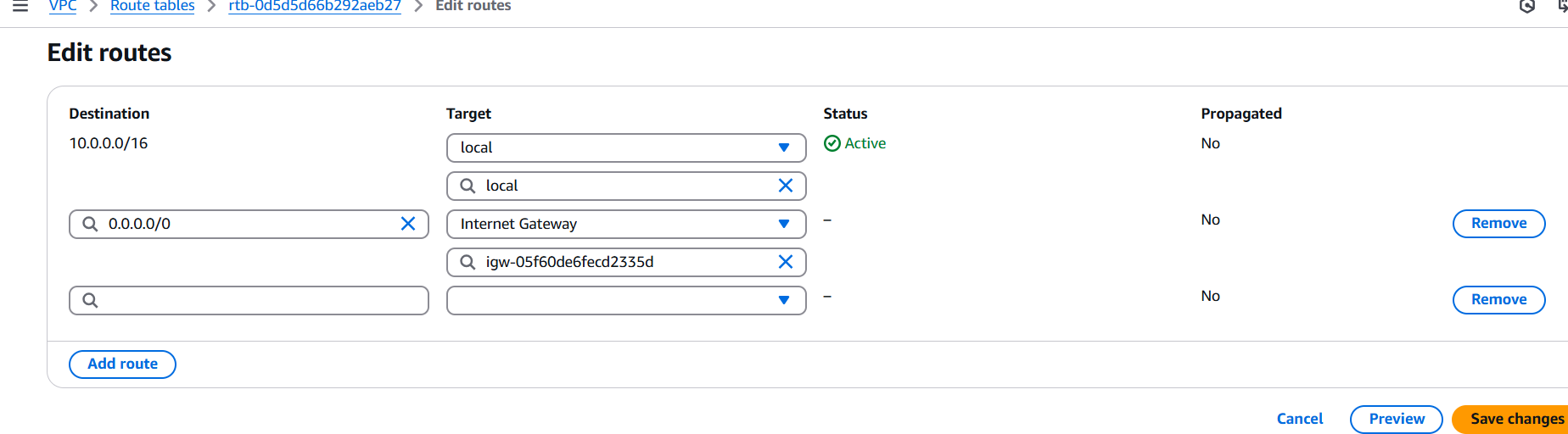
1. In the VPC dashboard, select **Internet Gateways**, then click **Create internet gateway**.
2. Enter a name (e.g., MyInternetGateway) and create the gateway.
3. Select the newly created internet gateway, click **Actions**, then **Attach to VPC**.
4. Choose your VPC and attach the gateway.



**4. Configure Route Tables**

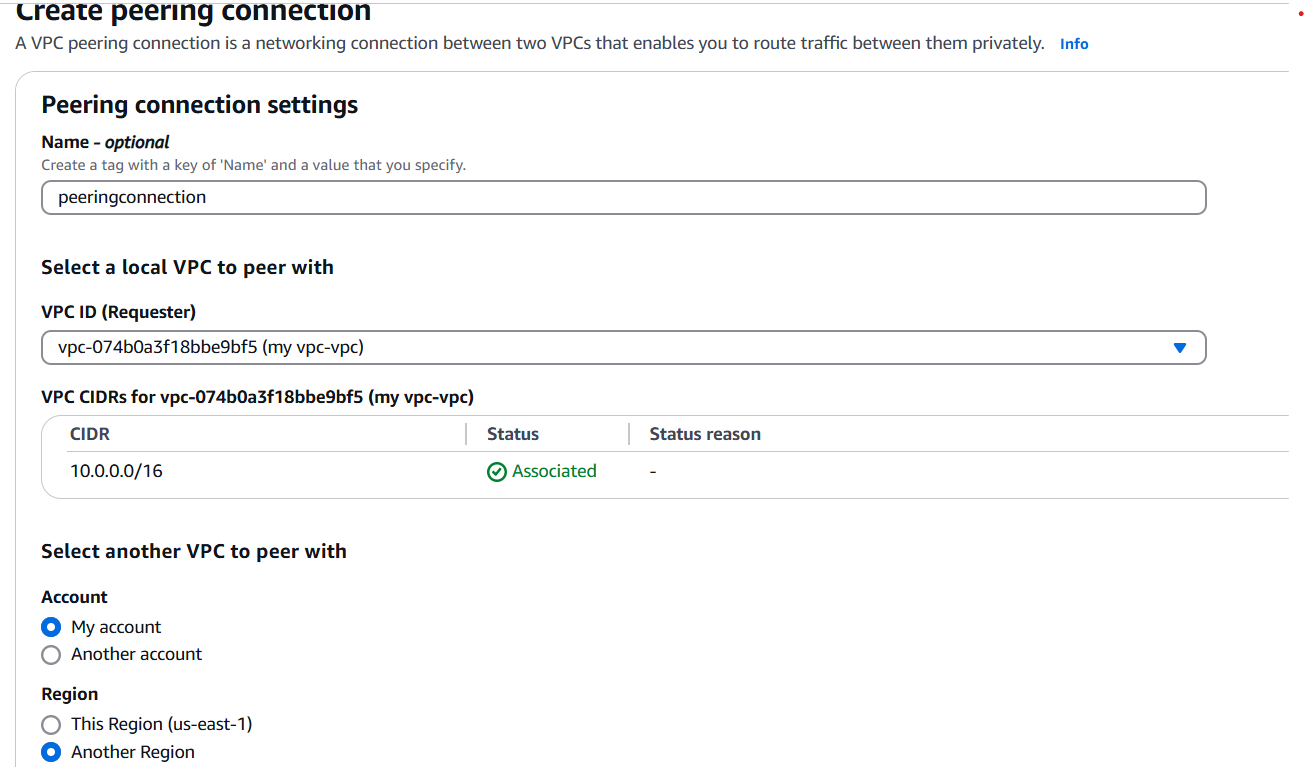
1. In the VPC dashboard, select **Route Tables**, then click **Create route table**.
2. Enter a name (e.g., PublicRouteTable) and select your VPC.
3. After creation, select the route table, go to the **Routes** tab, and click **Edit routes**.
4. Add a route:
   * **Destination**: 0.0.0.0/0
   * **Target**: Select the internet gateway you created.

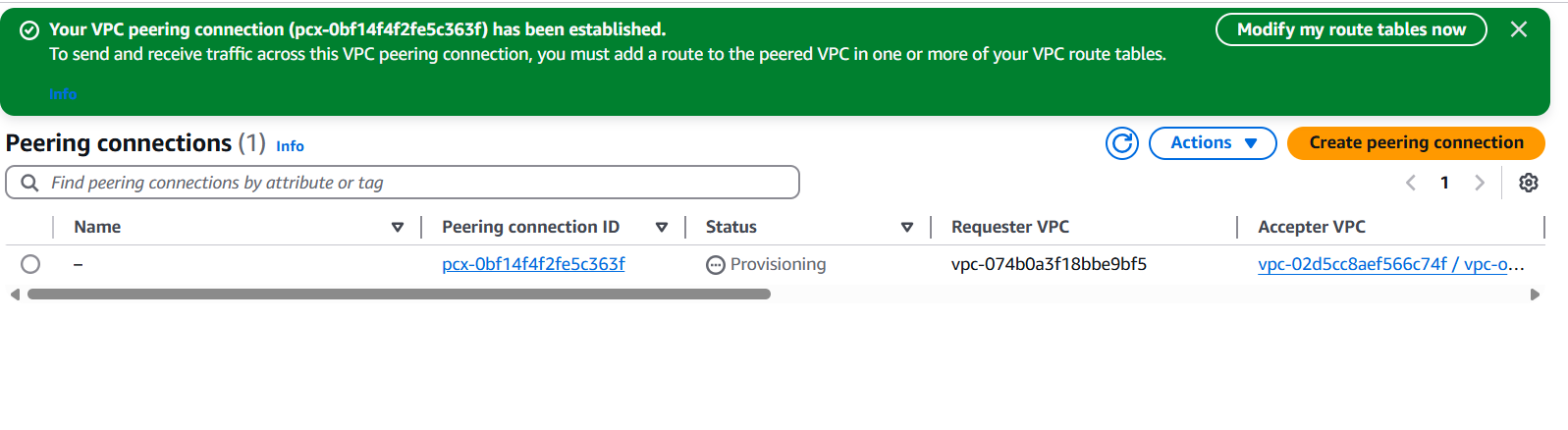




1. Enable VPC peering for cross region.

**Step 1: Create a VPC Peering Connection**  
1. Go to the VPC console.  
2. Click on "Peering Connections" in the left-hand menu.  
3. Click on "Create peering connection".  
4. Select "Another AWS account" or "My account" depending on your use case.  
5. Choose the VPC in the current region (Requester VPC).  
6. Enter the VPC ID and region of the VPC you want to peer with (Accepter VPC).  
7. Click "Create peering connection".  
**Step 2: Accept the VPC Peering Connection**1. Log in to the AWS account that owns the Accepter VPC.  
2. Go to the VPC console in the region of the Accepter VPC.  
3. Click on "Peering Connections" in the left-hand menu.  
4. Find the pending peering connection and select it.  
5. Click on "Actions" and then select "Accept request".

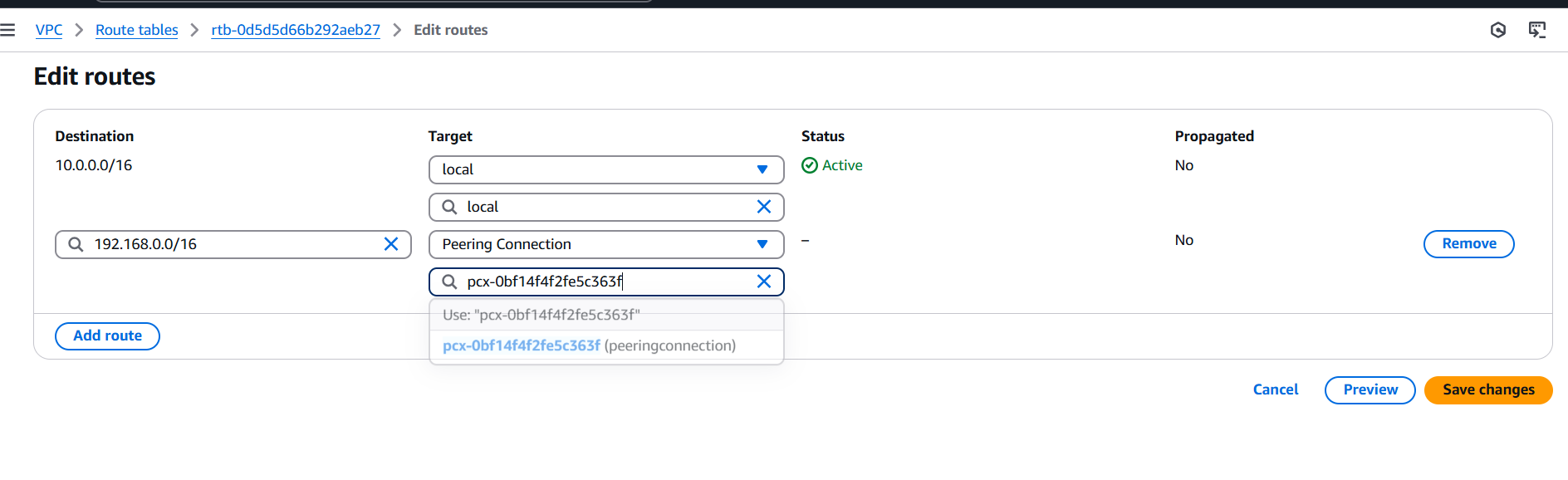


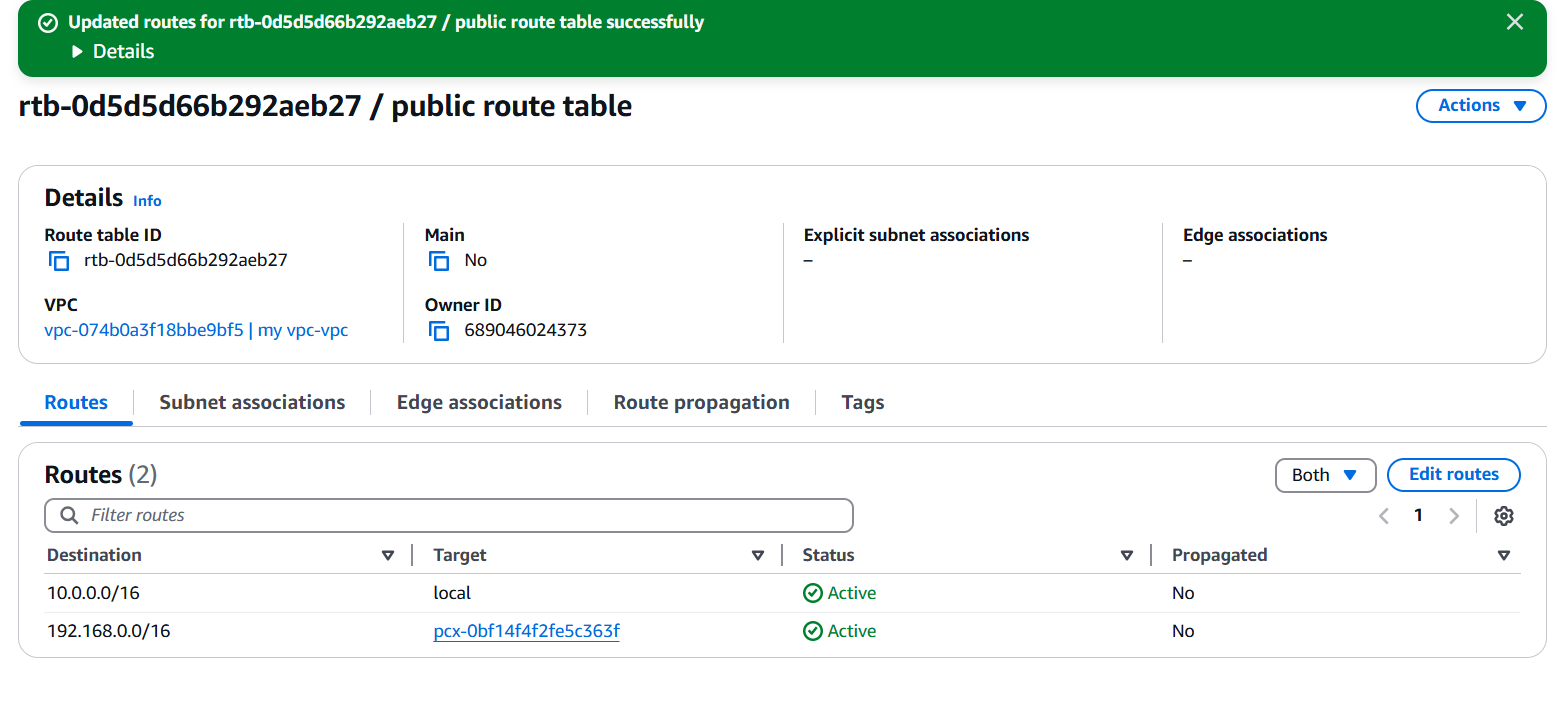
**Step 3: Add Route Table Entries**

To allow communication between VPCs:

**In Region A:**

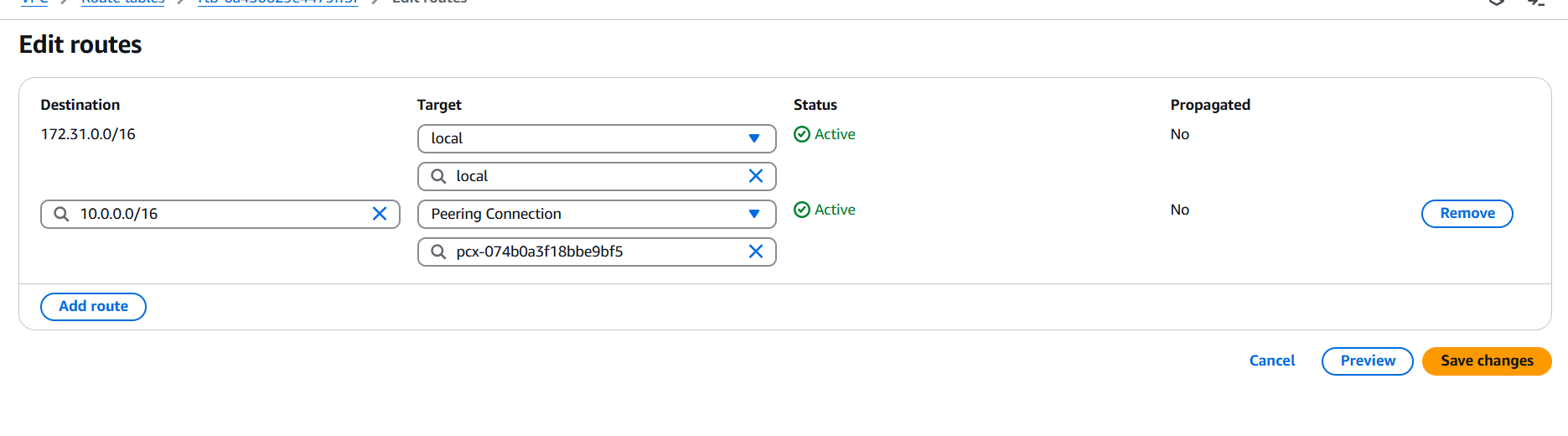
1. Go to **Route Tables**.
2. Select the route table associated with the subnet in my vpc N.virgina.
3. Click **Edit routes**.
4. Add route:
   * **Destination**: CIDR of VPC in Ohio (192.168.0.0/16)
   * **Target**: Select the Peering Connection.

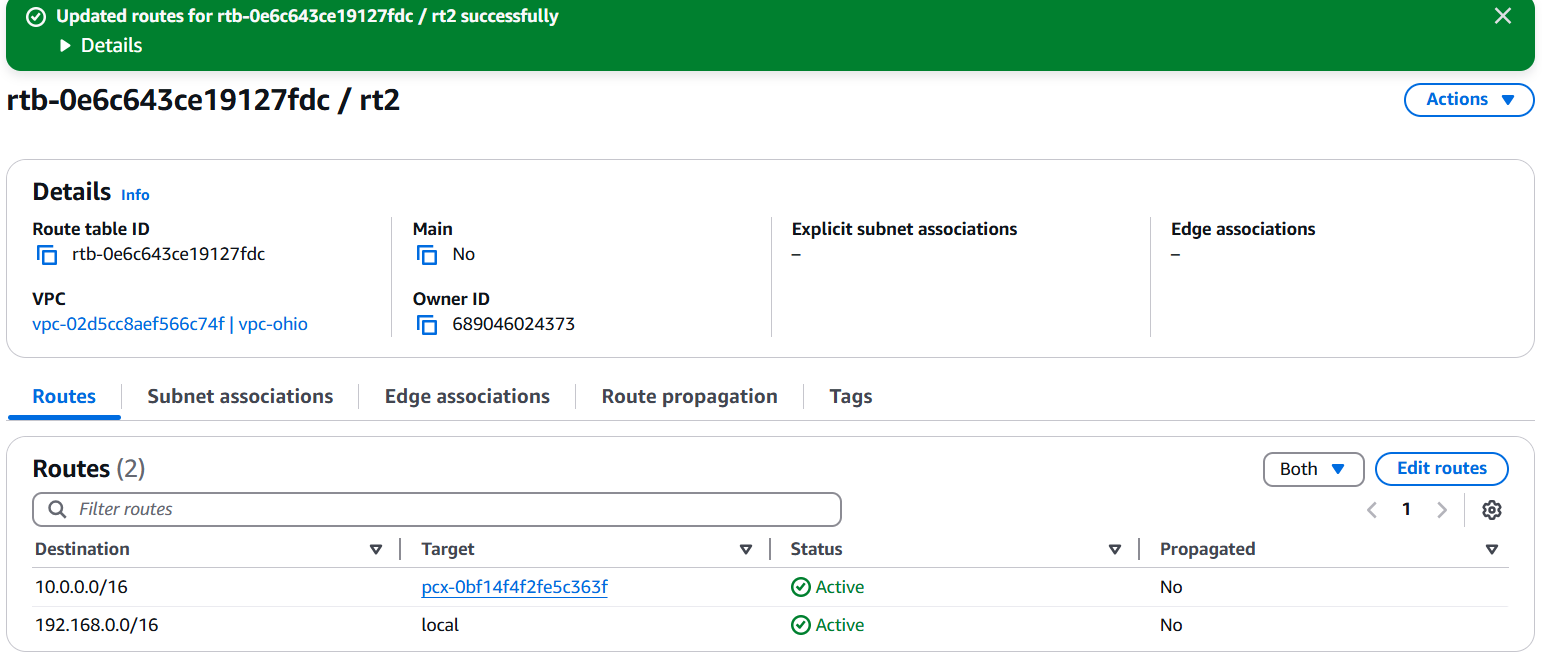


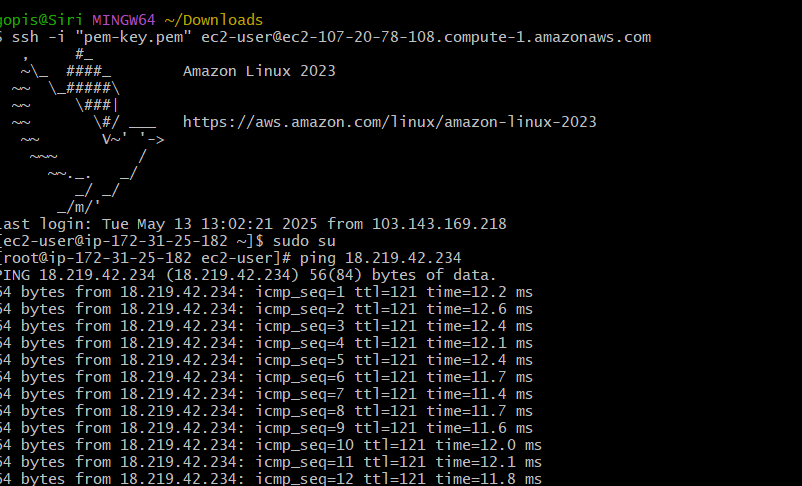


**In Region B:**

1. Go to **Route Tables**.
2. Select the route table associated with the subnet in VPC Ohio.
3. Click **Edit routes**.
4. Add route:
   * **Destination**: CIDR of myVPC in N.virgina ( 10.0.0.0/16)
   * **Target**: Select the Peering Connection.

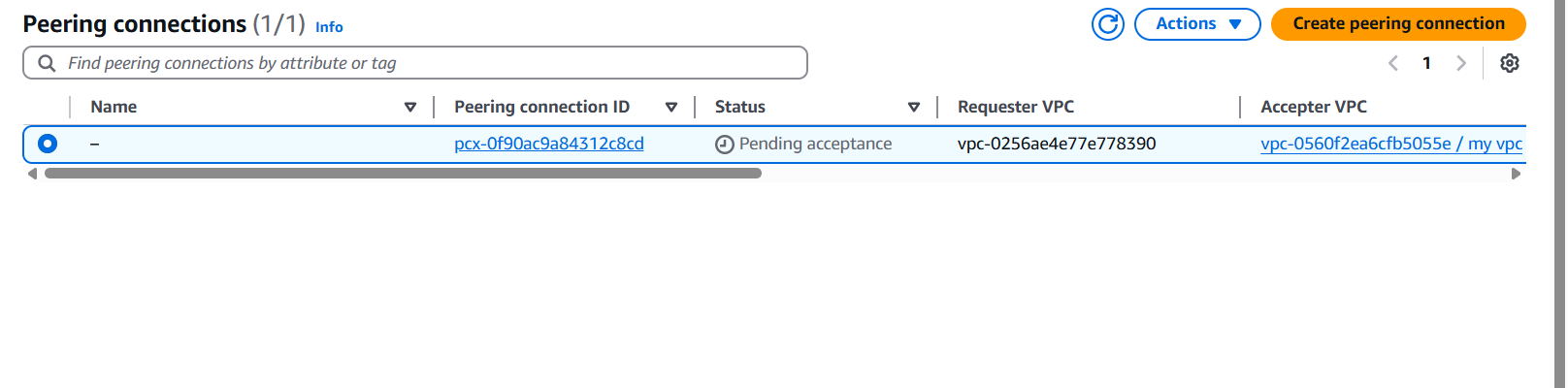


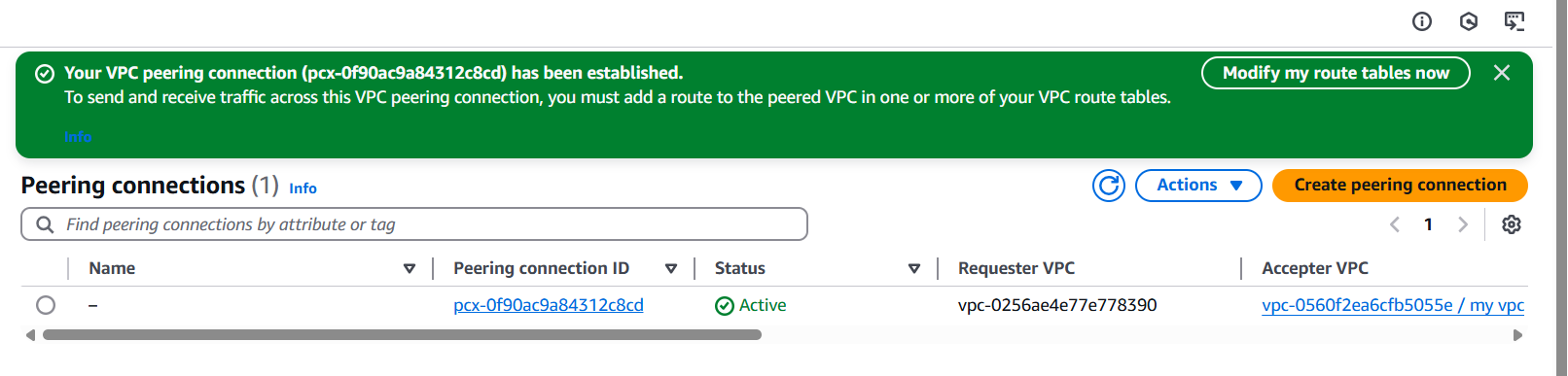


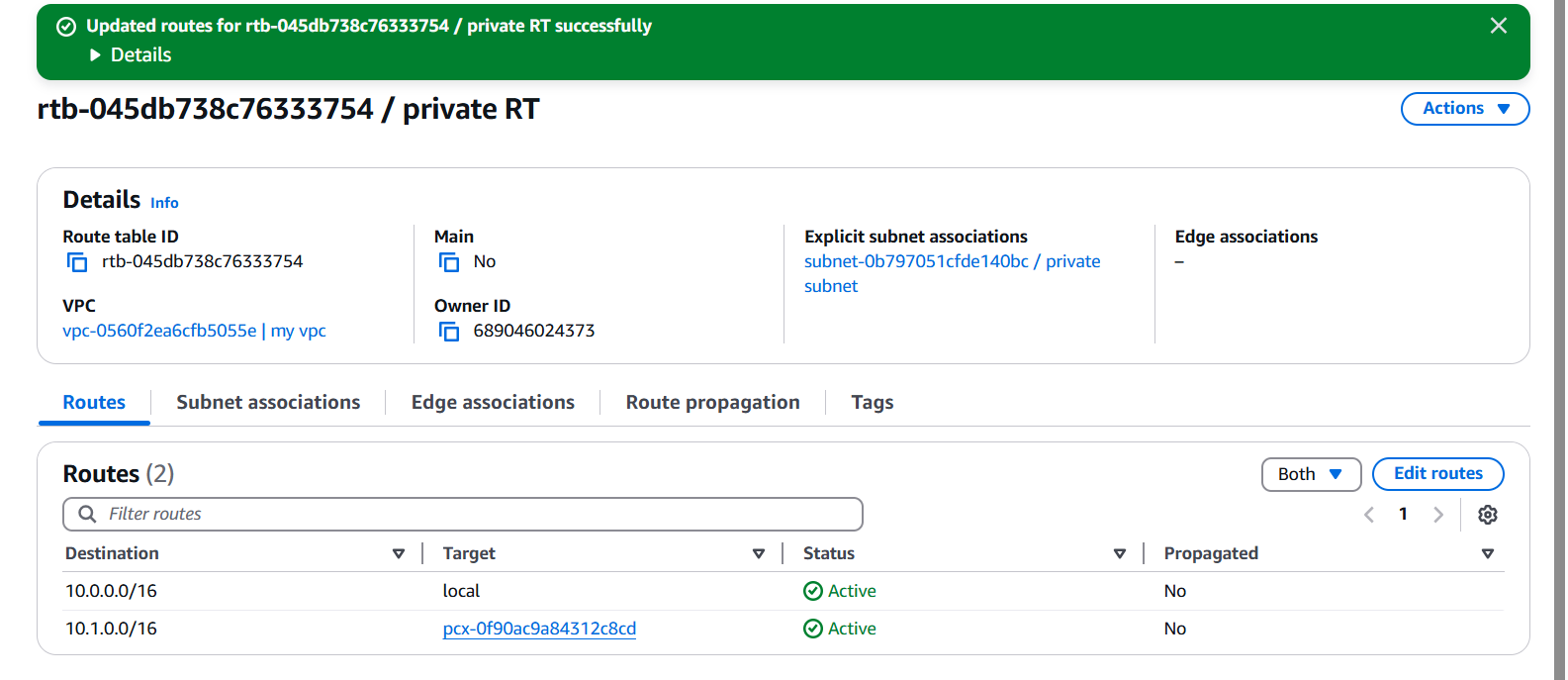


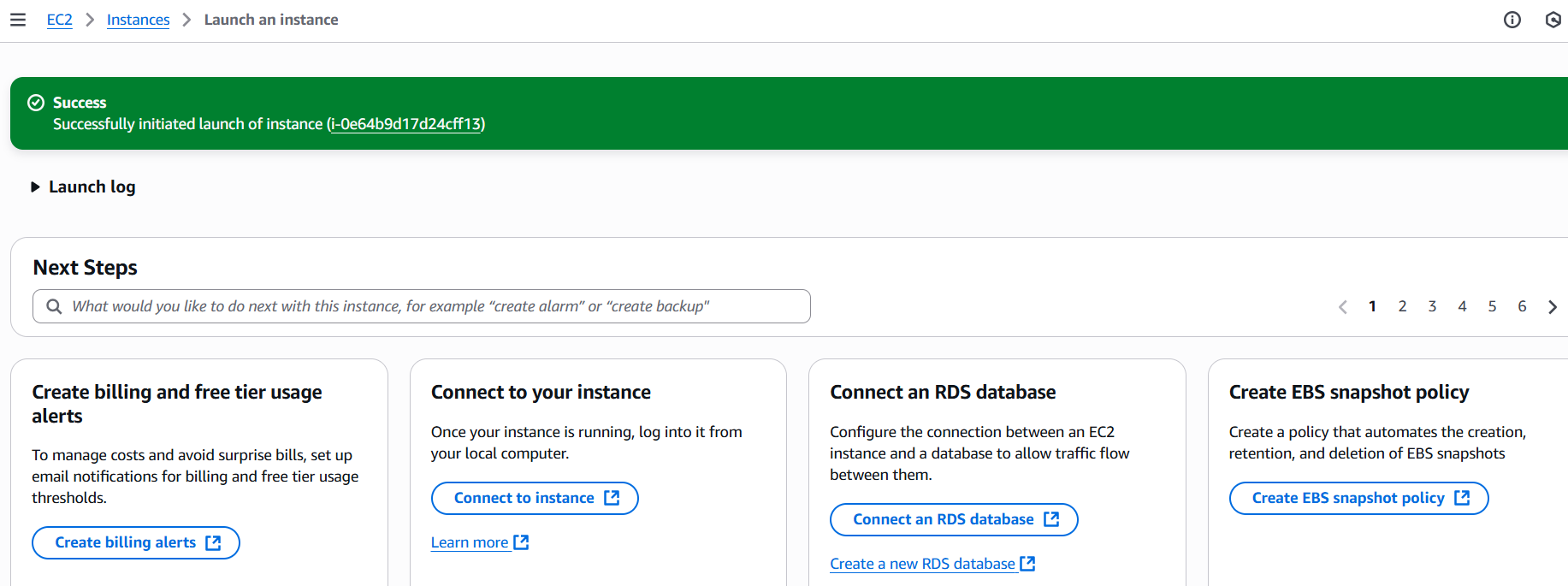
1. Enable VPC peering for cross account. (You can collaborate with your friend and do this task).

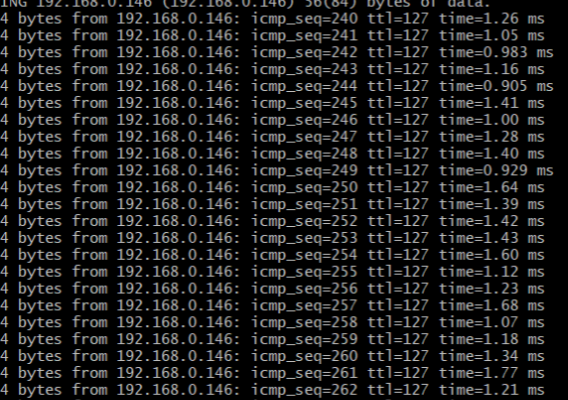
Step 1: Create a VPC Peering Connection (Requester Account)  
1. Go to the VPC console in the requester account.  
2. Click on "Peering Connections" and then "Create peering connection".  
3. Select "Another AWS account".  
4. Enter the AWS account ID and VPC ID of the accepter account.  
5. Choose the VPC in the requester account.  
6. Click "Create peering connection".  
Step 2: Accept the VPC Peering Connection (Accepter Account)











4) Setup VPC Transit gateway.

1. Go to VPC Dashboard > Your VPCs

2. Create VPC A:

CIDR block: 10.0.0.0/16

3. Create VPC B:

CIDR block: 10.1.0.0/16

Add subnets, route tables, and IGWs as needed.

Create Transit Gateway (TGW)

1. Go to VPC Dashboard > Transit Gateways

2. Click Create Transit Gateway

3. Fill in:

Name: My-TGW

Amazon ASN: leave default or set custom (e.g., 64512)

DNS Support: Enabled

Default Route Table Association: Enable

Default Route Table Propagation: Enable

4. Click Create Transit Gateway

Wait until status is Available

Attach VPCs to Transit Gateway

1. Go to Transit Gateway Attachments > Create Attachment

2. Choose:

Attachment Type: VPC

Transit Gateway: Select the one you created

VPC: Select VPC A

Subnets: Select one subnet from each AZ you want to connect

3. Click Create Attachment

Wait for attachments to become Available

1. Repeat the same steps to attach VPC B

Wait for attachments to become Available

Update Route Tables in VPCs

Go to Route Tables of each VPC and add the appropriate route:

In VPC A:

Destination: 10.1.0.0/16

Target: Transit Gateway

In VPC B:

Destination: 10.0.0.0/16

Target: Transit Gateway

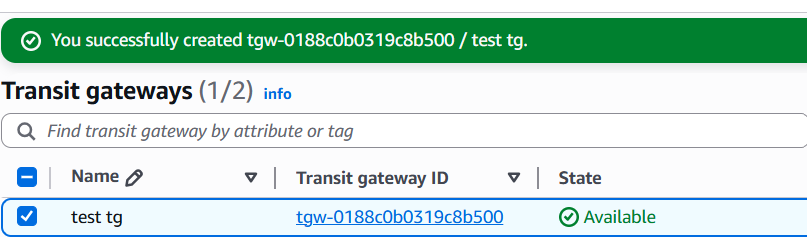
Launch EC2 Instances for Testing

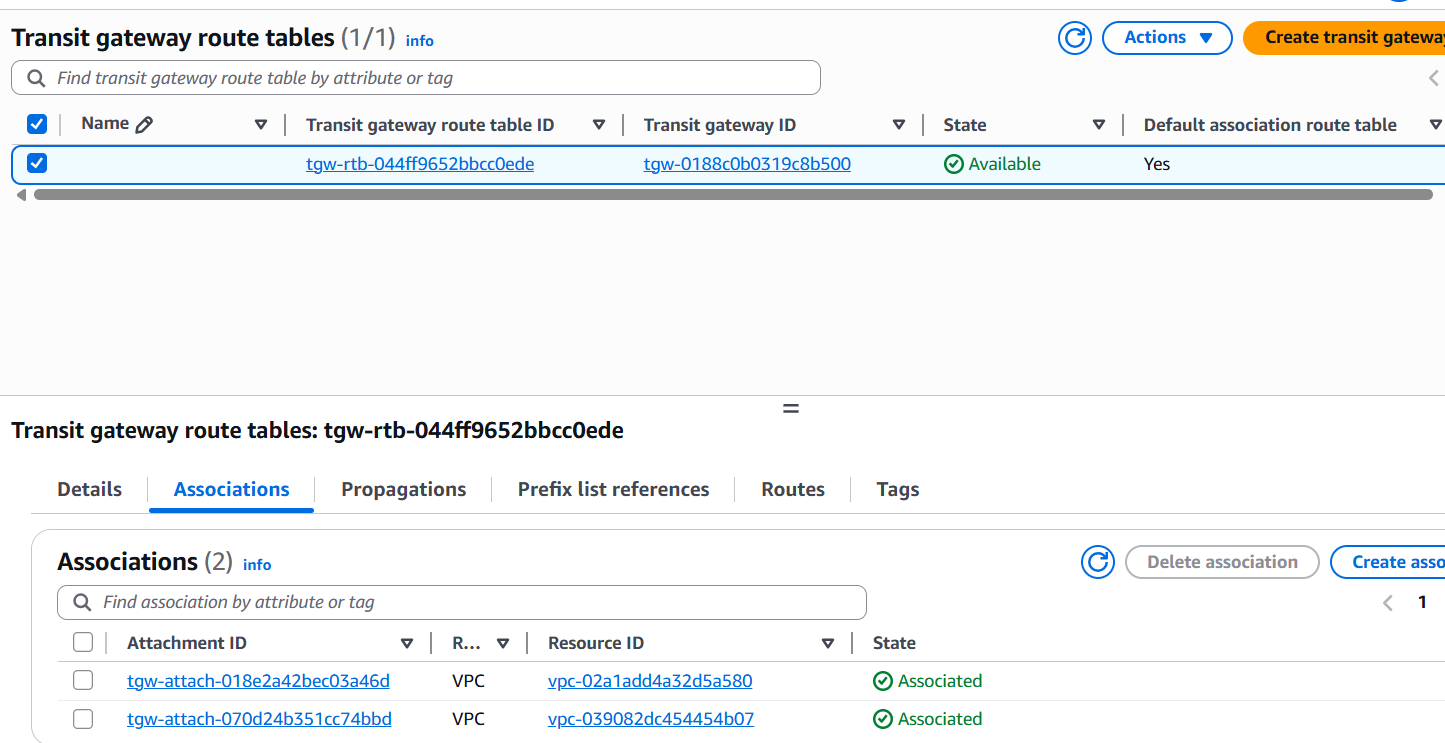
1. Launch one EC2 instance in each VPC

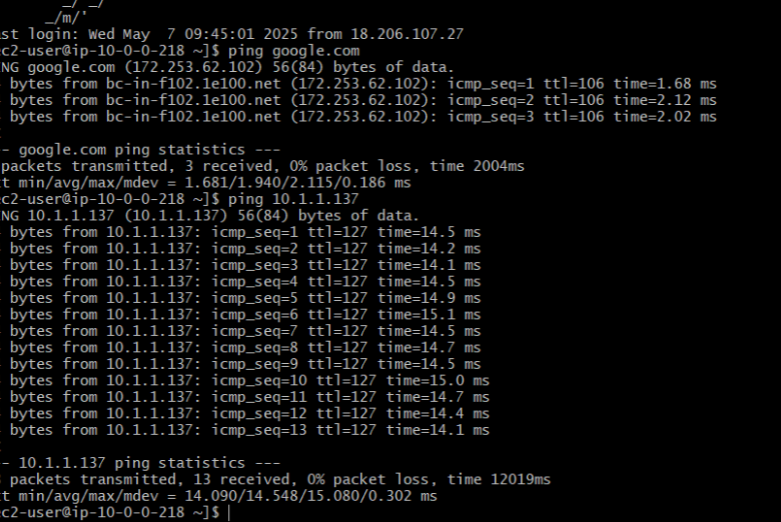
2. Use private IPs to pin

=>create vpc atleast create one subnet in vpc and route table will be associated

with subnet.

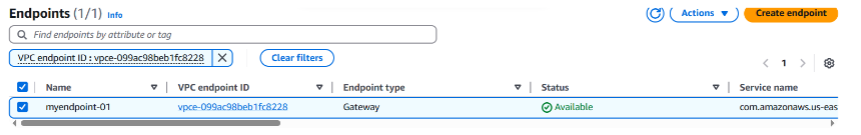


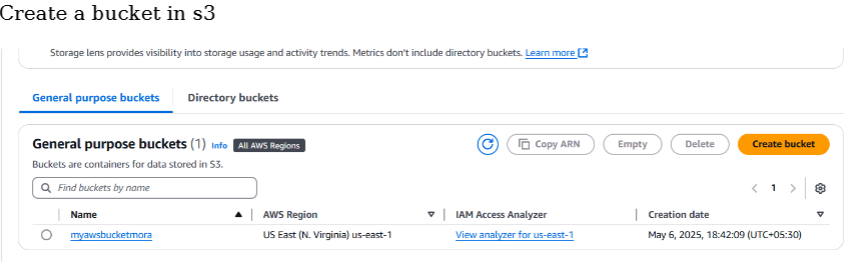


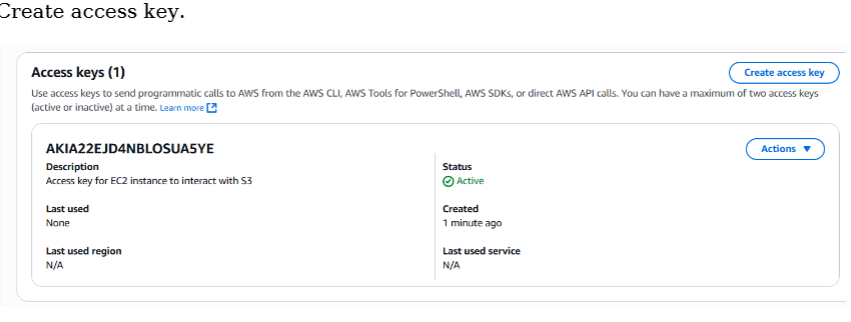


5) Setup VPC End Point.

create vpc atleast create one subnet in vpc and route table will be associated   
with subnet.  
1. Go to the VPC Dashboard  
•Sign in to AWS Management Console.  
•Open the VPC service.  
2. Create Endpoint  
•In the left panel, click Endpoints.  
•Click "Create Endpoint".  
3. Configure Endpoint  
•Service Category: Select AWS services.  
•Service Name:  
Search for s3 and choose com.amazonaws.<region>.s3  
Example: com.amazonaws.us-east-1.s3 for N. Virginia  
•VPC: Choose your VPC.  
4. Configure Route Tables  
•Select the route table(s) associated with your subnet(s).  
•This allows traffic to route to the endpoint.  
5. Policy  
•Choose "Full access" or custom policy depending on your security needs.  
6. Name tag (Optional)  
•Give your endpoint a name like MyS3Endpoint.  
7. Create Endpoint  
•Click Create endpoint.







Test the VPC Endpoint  
1. Launch an EC2 instance without a public IP in that VPC.  
2. Use the AWS CLI or curl to access an S3 bucket:  
bash  
CopyEdit  
aws s3 ls s3://<your-bucket-name>  
If this works from a private EC2, your VPC endpoint is functioning correctly