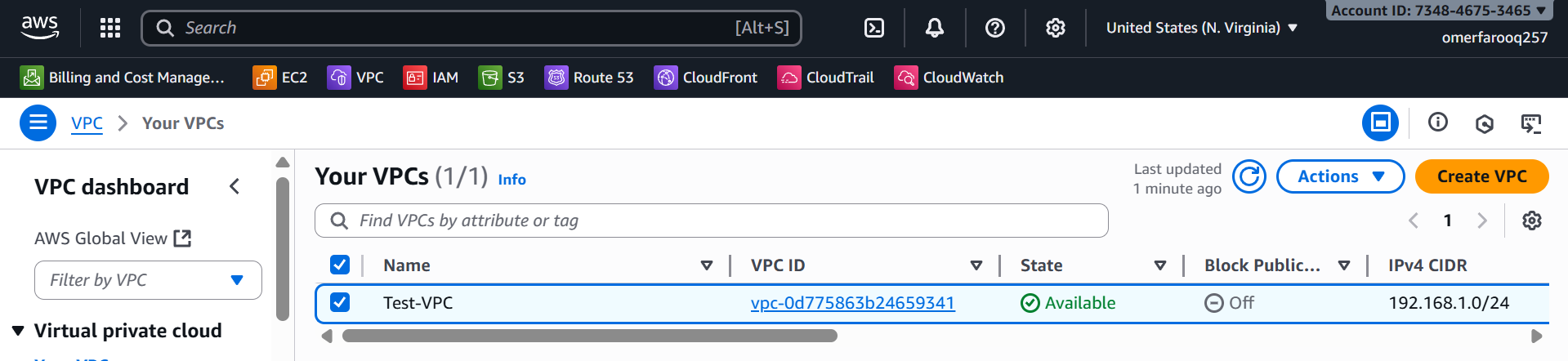
# VPC Tasks – Peering, Transit Gateway & Endpoint

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

**Purpose:** To establish a basic multi-tier architecture with public-facing and private resources

*STEPS TO CREATE A VPC WITH ONE PUBLIC AND ONE PRIVATE SUBNET:*



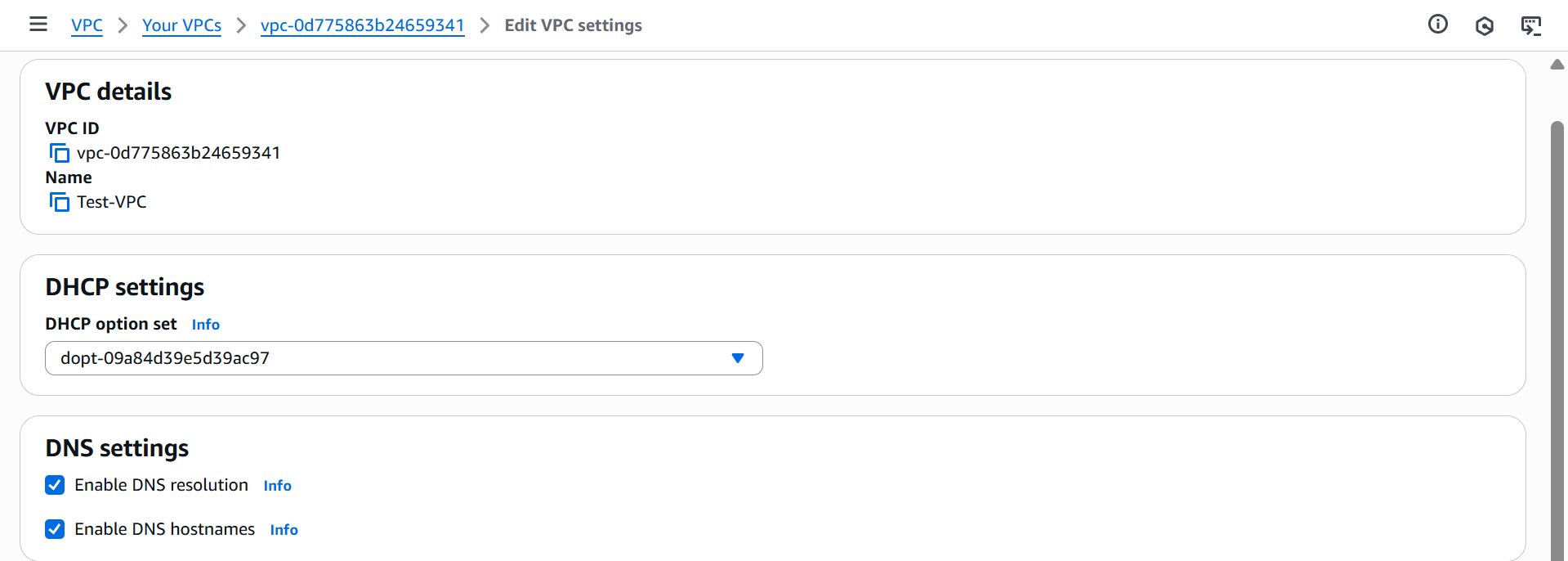
**Creating a VPC:**

* Navigate to **VPC Dashboard** → **VPCs**
* Click **Create VPC**
* Configurations:  
  Name: Test-VPC

IPv4 CIDR block: 192.168.1.0/24  
IPv6 CIDR block: No IPv6 CIDR block  
Tenancy: Default

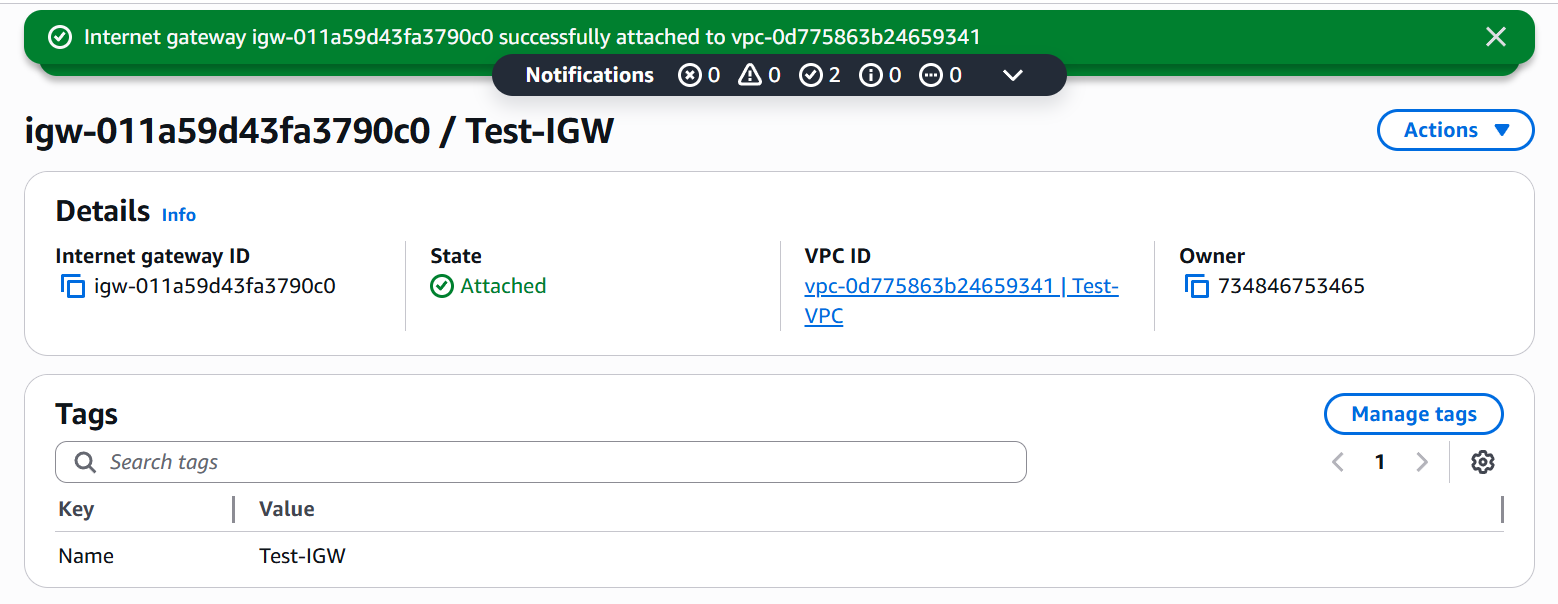
* Click **Create VPC**

**Enable DNS Hostname and DNS Resolution:**

****

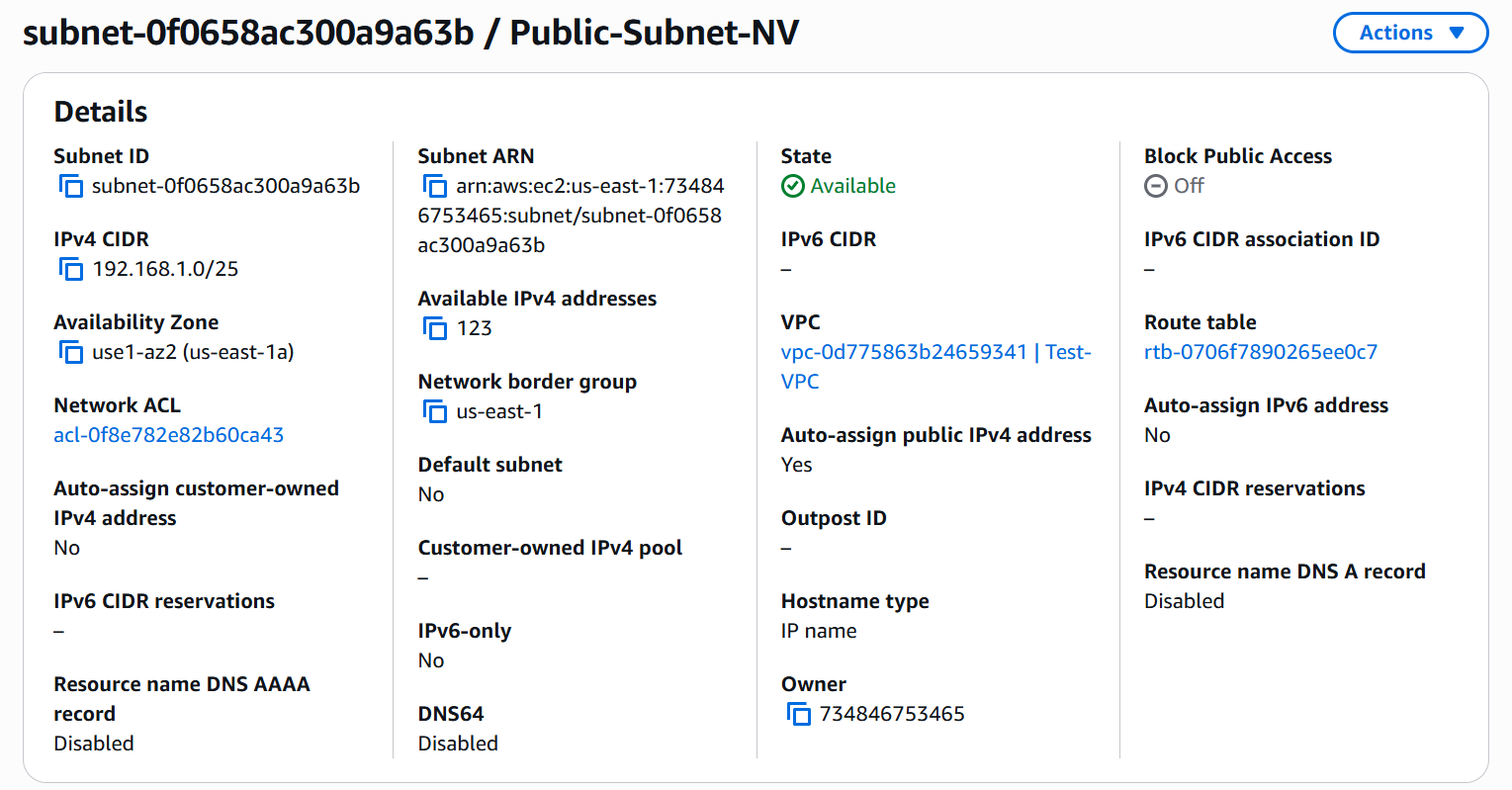
* Select your VPC (Test-VPC)
* Click **Actions** → **Edit VPC settings**
* Check both:
  + Enable DNS hostnames
  + Enable DNS resolution
* Click **Save**

**Create Internet Gateway:**

****

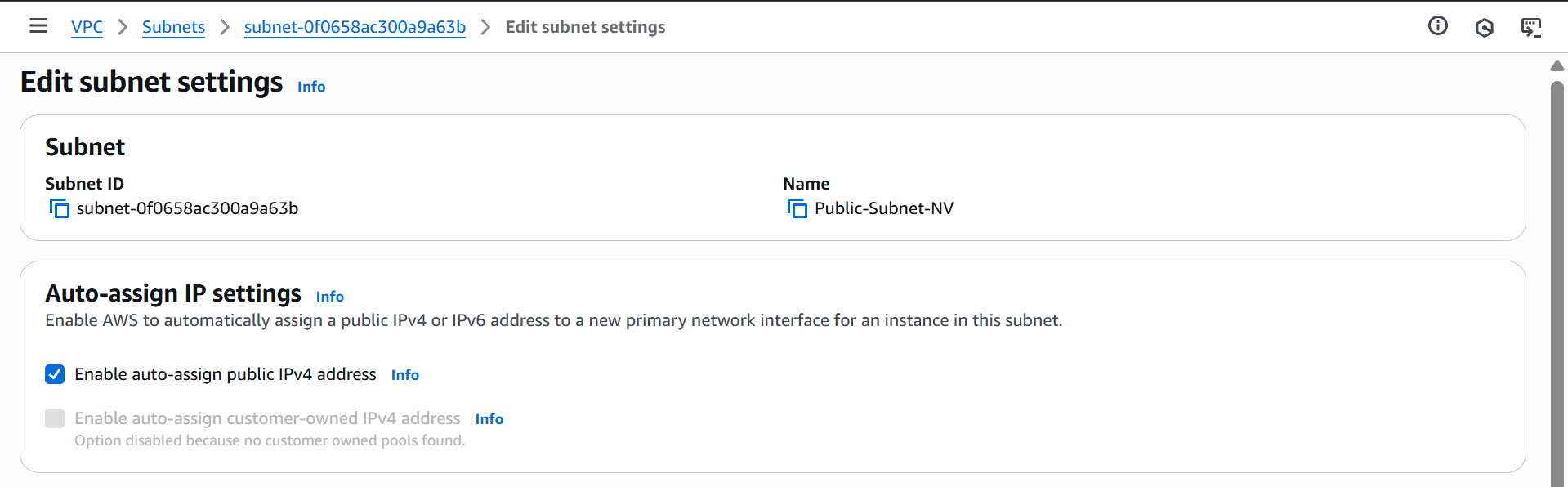
* Navigate to **VPC → Internet Gateways**
* Click **Create Internet Gateway**
* Name: **Test-IGW**
* Click **Create Internet gateway**
* Select the **IGW → Actions → Attach to VPC**
* Select **Test-VPC**
* Click **Attach Internet gateway**

**Creating Public Subnet:**

****

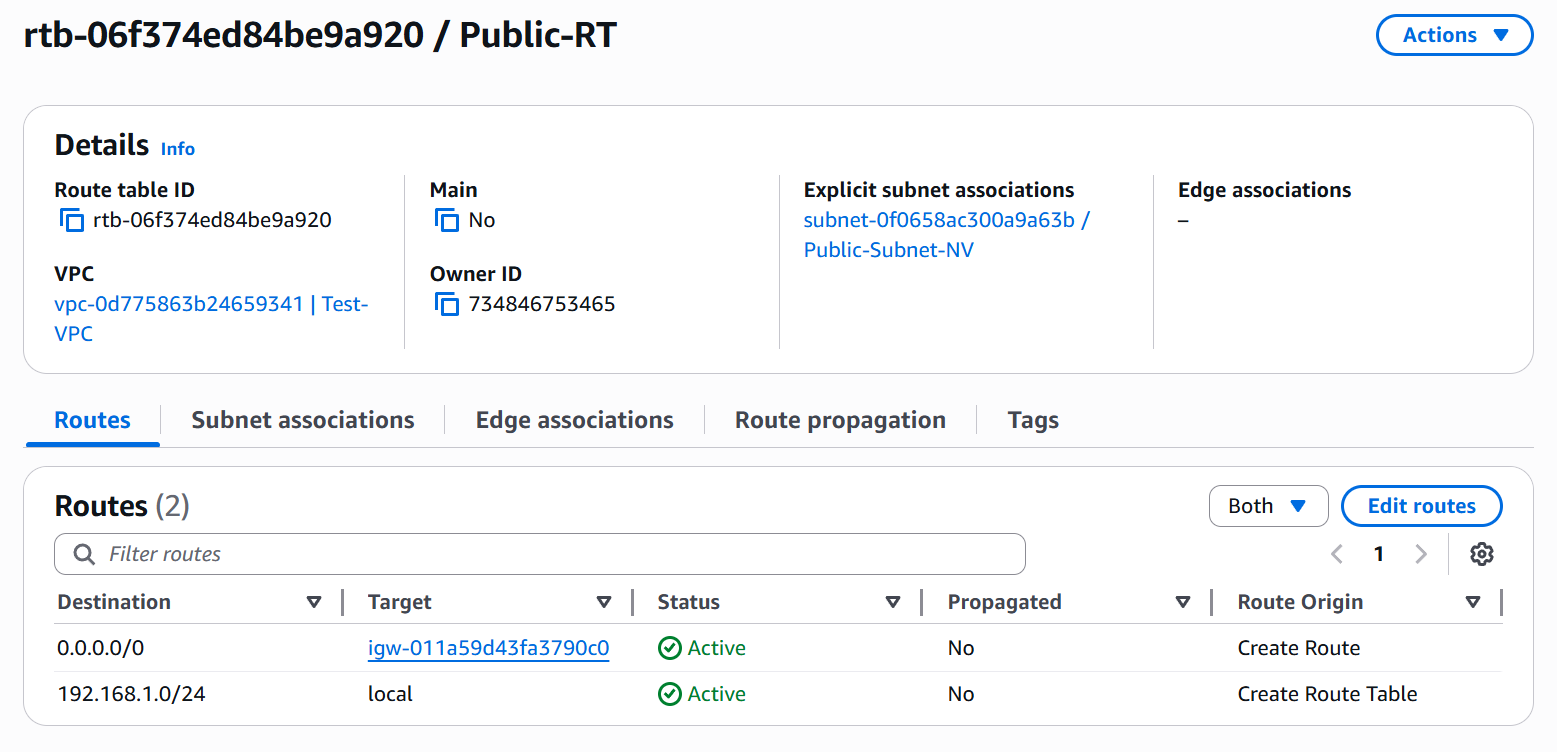
* Navigate to **VPC → Subnets**
* Click **Create Subnet**
* Configuration:
* VPC ID: **Test-VPC**
* Subnet name: **Public-Subnet**
* Availability Zone: **us-east-1a**
* IPv4 CIDR block: **192.168.1.0/25**
* Click **Create Subnet**

**Enable Auto-Assign Public IP for Public Subnet:**

****

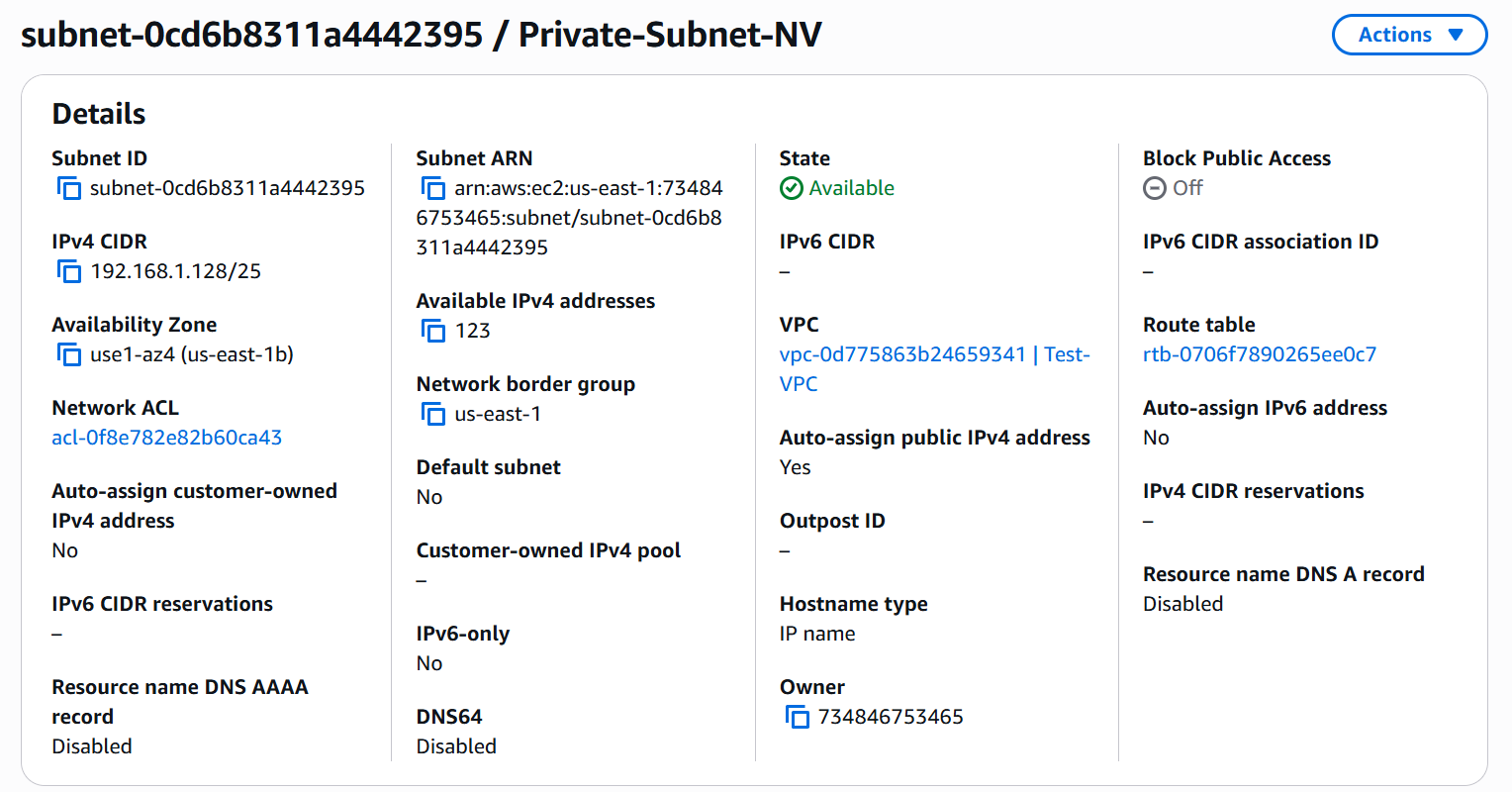
* Select **Public-Subnet-NV**
* Click **Actions → Edit subnet settings**
* Check**: Enable auto-assign public IPv4 address**
* Click **Save**

**Create Public Route Table**

****

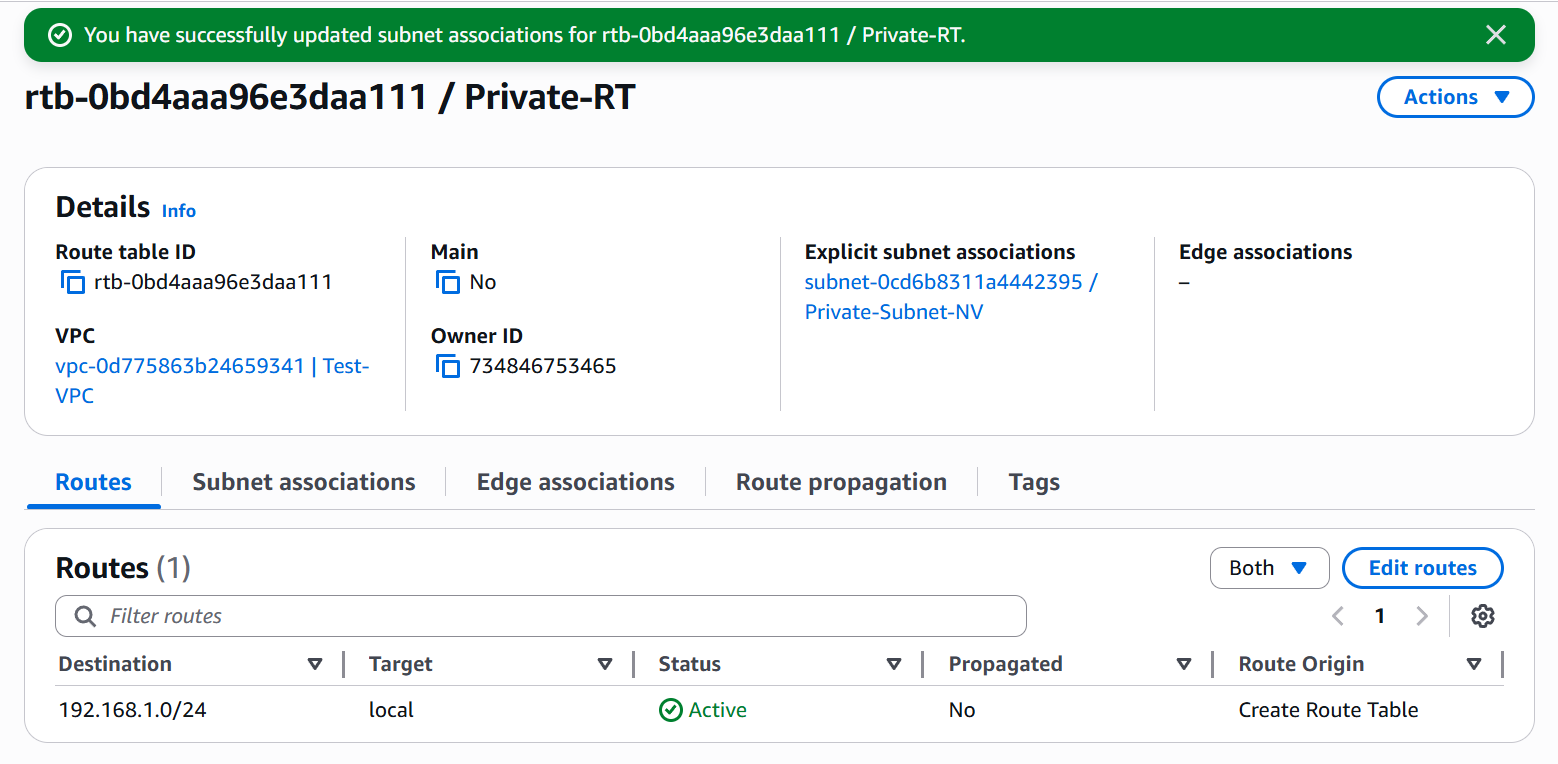
* Navigate to **VPC → Route Tables**
* Click **Create Route Table**
* Name: **Public-RT**
* VPC: **Test-VPC**
* Click **Create Route Table**
* Select the **RT → Routes tab**
* Click **Edit Routes**
  + Add route:
* Destination: **0.0.0.0/0**
* Target: **Internet Gateway → Test-IGW**
* **Save** and associate with Public-Subnet

**Creating Private Subnet:**

****

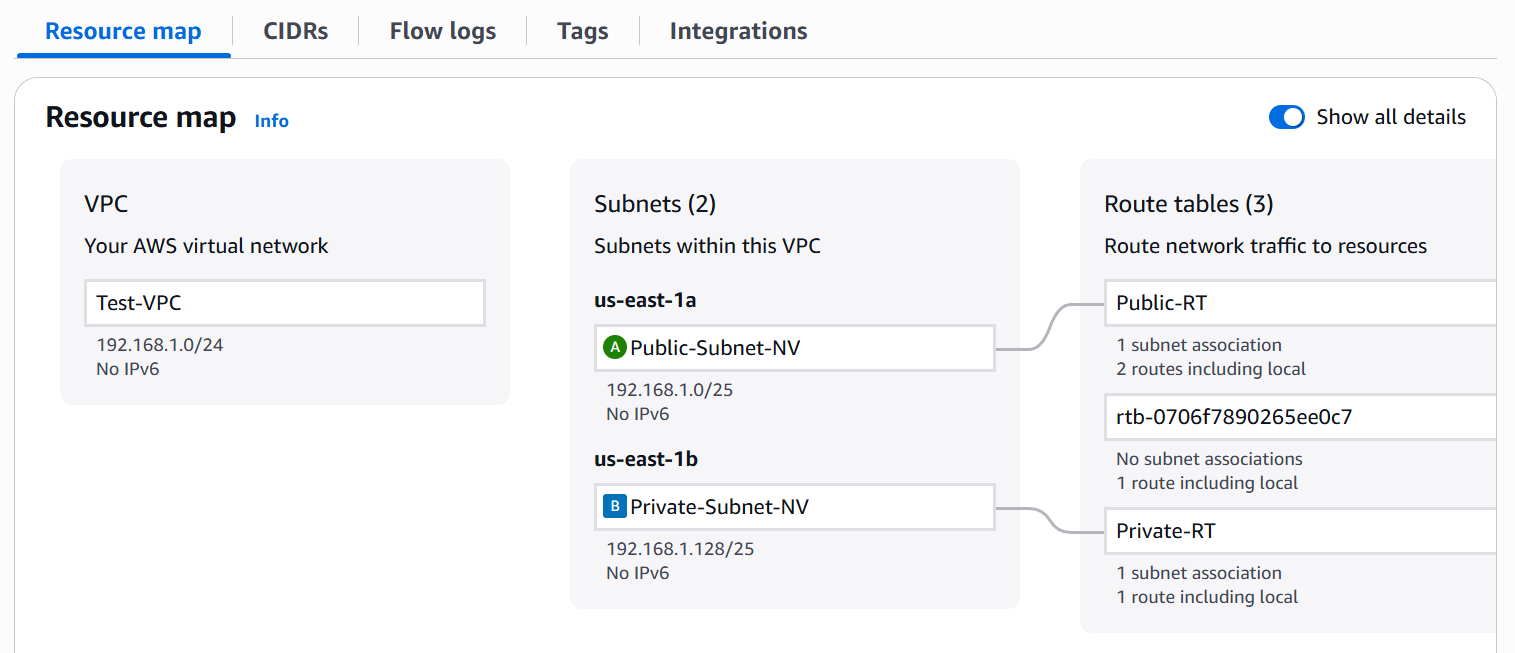
* Navigate to **VPC → Subnets**
* Click **Create Subnet**
* Configuration:
* VPC ID: **Test-VPC**
* Subnet name: **Private-Subnet**
* Availability Zone: **us-east-1b**
* IPv4 CIDR block: **192.168.1.128/25**
* Click **Create Subnet**

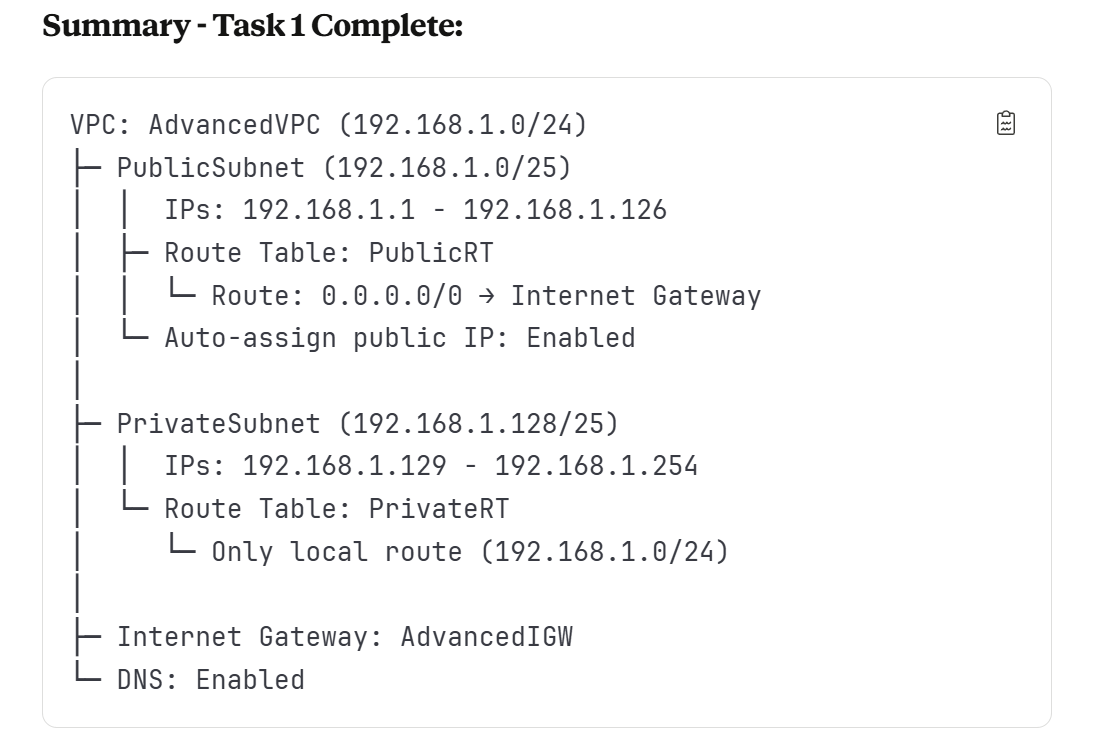
**Create Private Route Table**



* Navigate to **VPC → Route Tables**
* Click **Create Route Table**
* Name: **Private-RT**
* VPC: **Test-VPC**
* Click **Create Route Table**
* Select the **RT → Subnet Associations tab**
* Click **Edit Subnet Associations**
* Select **Private-Subnet**
* **Save**

**Verify the Setup:**

****

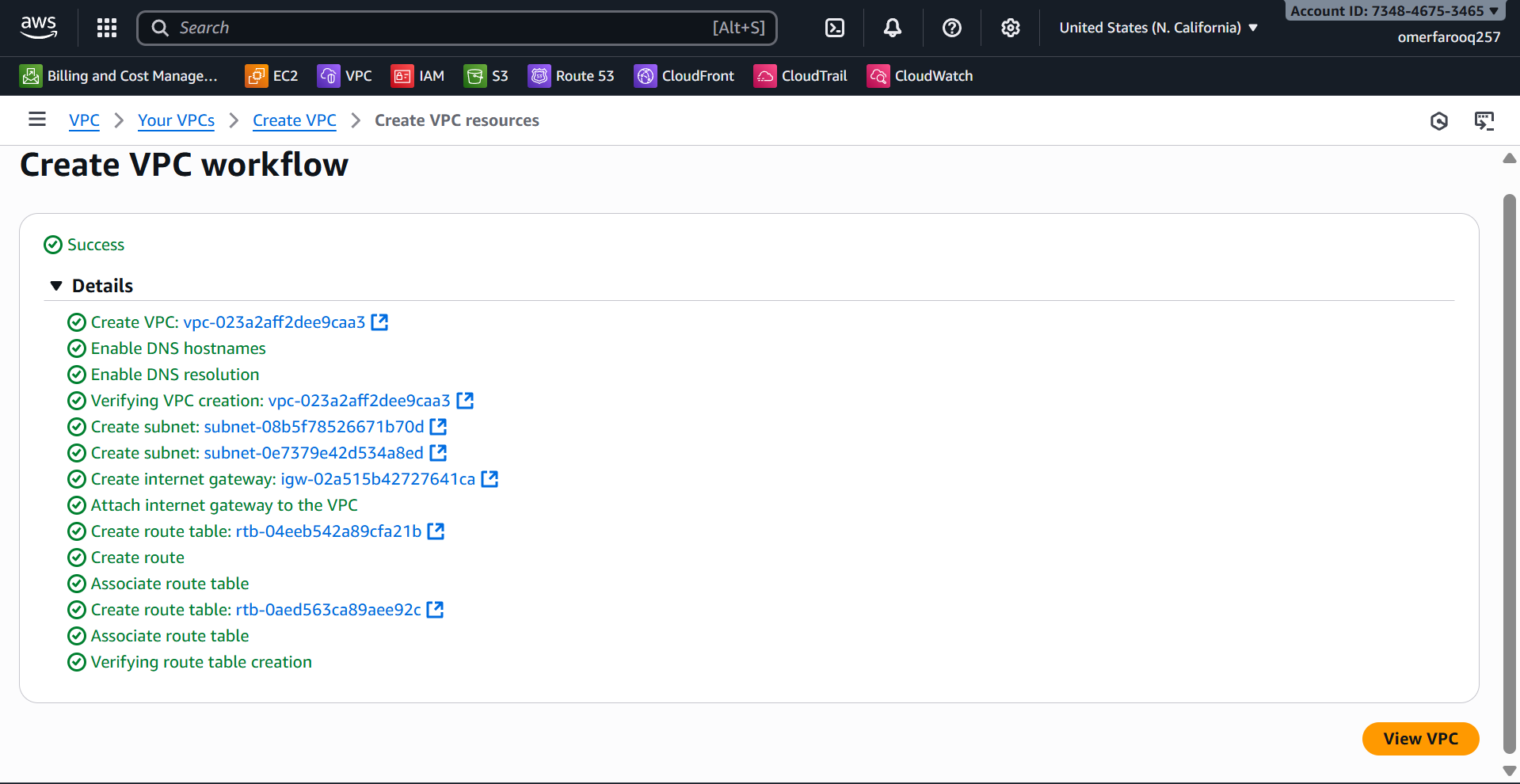
****

1. **Enable VPC peering for cross-region.**

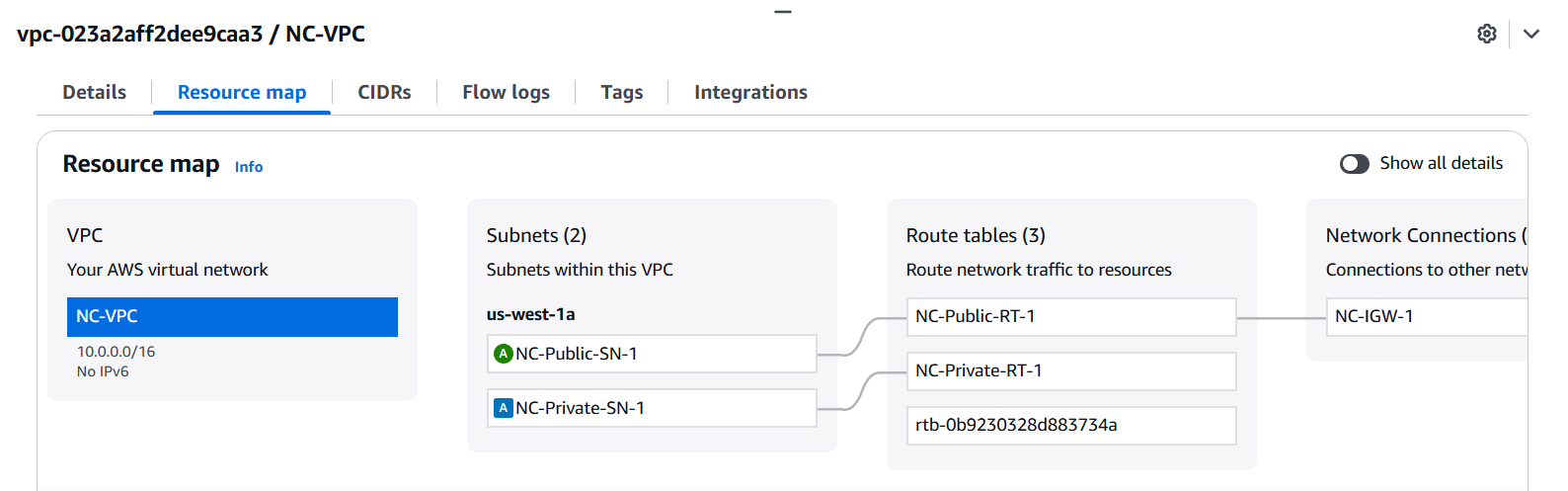
**Purpose: T**o enable private network connectivity between VPCs without internet exposure.

*STEP TO ENABLE VPC PEERING FOR CROSS-REGION:*

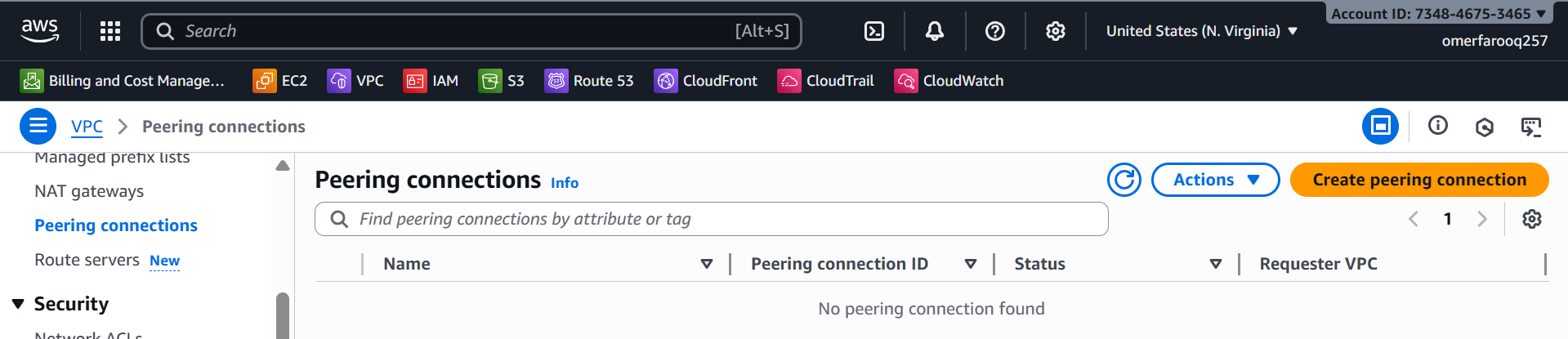
**Create a Second VPC in a Different Region – N. California**



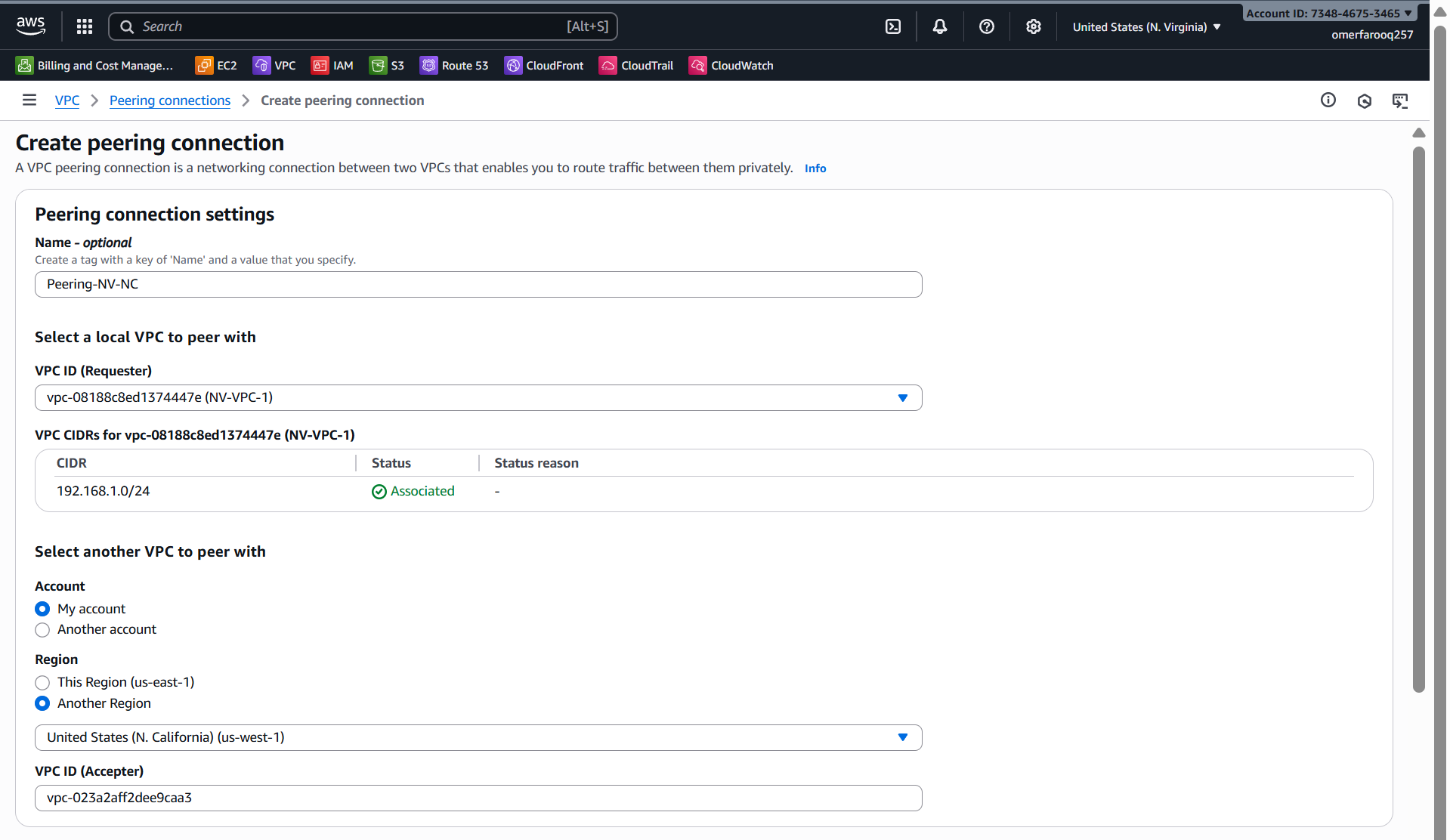
* The architecture of the VPC is shown in the image below.



**Create VPC Peering Connection (Requester Side)**

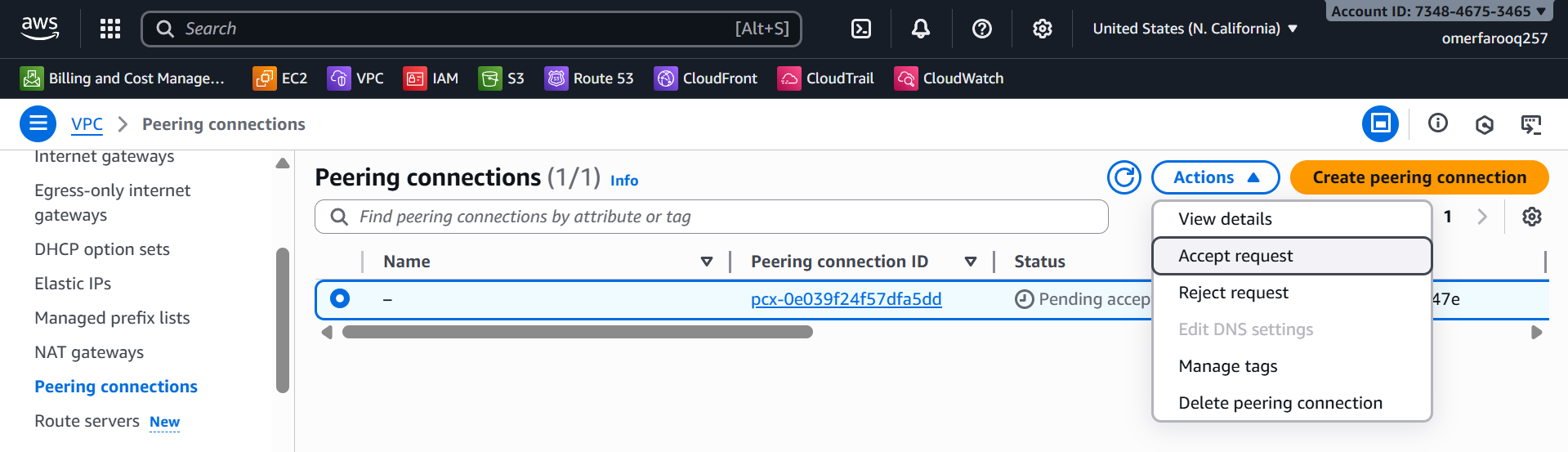


* Navigate to **VPC** → **Peering Connections**
* Click **Create Peering Connection**
* Configuration:
  + Peering connection name tag: Peering-NV-NC
  + **VPC ID (Requester):** vpc-0a1b2c3d4e5f6g7h8 (NV-VPC-1) - Region: us-east-1
  + VPC to accept peering: - Account: [Your AWS Account ID]
  + **VPC ID (Accepter)**: vpc-0i1j2k3l4m5n6o7p8 (NC-VPC) - Region: us-west-2
* Click **Create Peering Connection**



**Accept VPC Peering Connection (Accepter Side)**

* Switch to the us-west-2 region in the AWS Console
* Navigate to **VPC → Peering Connections**
* Find the pending peering connection
* Select it. **Actions → Accept Request**
* Click **Accept Peering Connection**





**Enable DNS Resolution Over Peering**

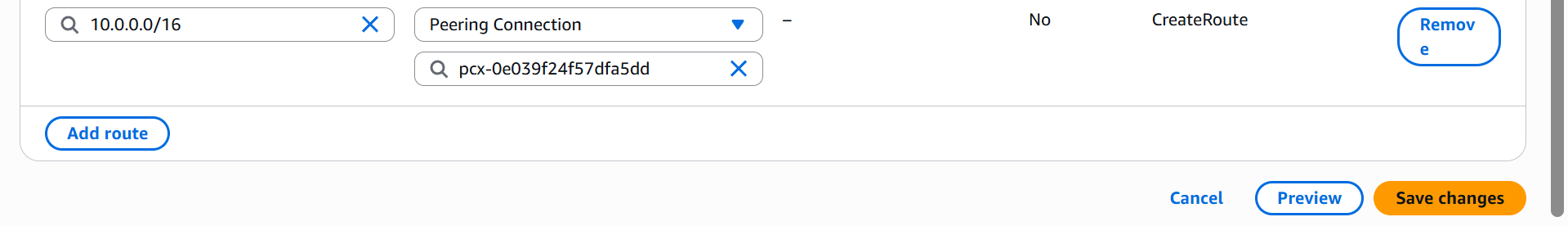
* Navigate to **VPC → Peering Connections**
* Select the peering connection
* Click **Actions → Edit Peering Connection Options**
* Check: ✅ **Allow DNS resolution from remote VPC endpoint**
* Click **Save**

**Update Route Tables (Requester Side - us-east-1)**

* Navigate to **VPC → Route Tables**
* Select **NV-Public-RT**
* **Routes tab → Edit Routes**

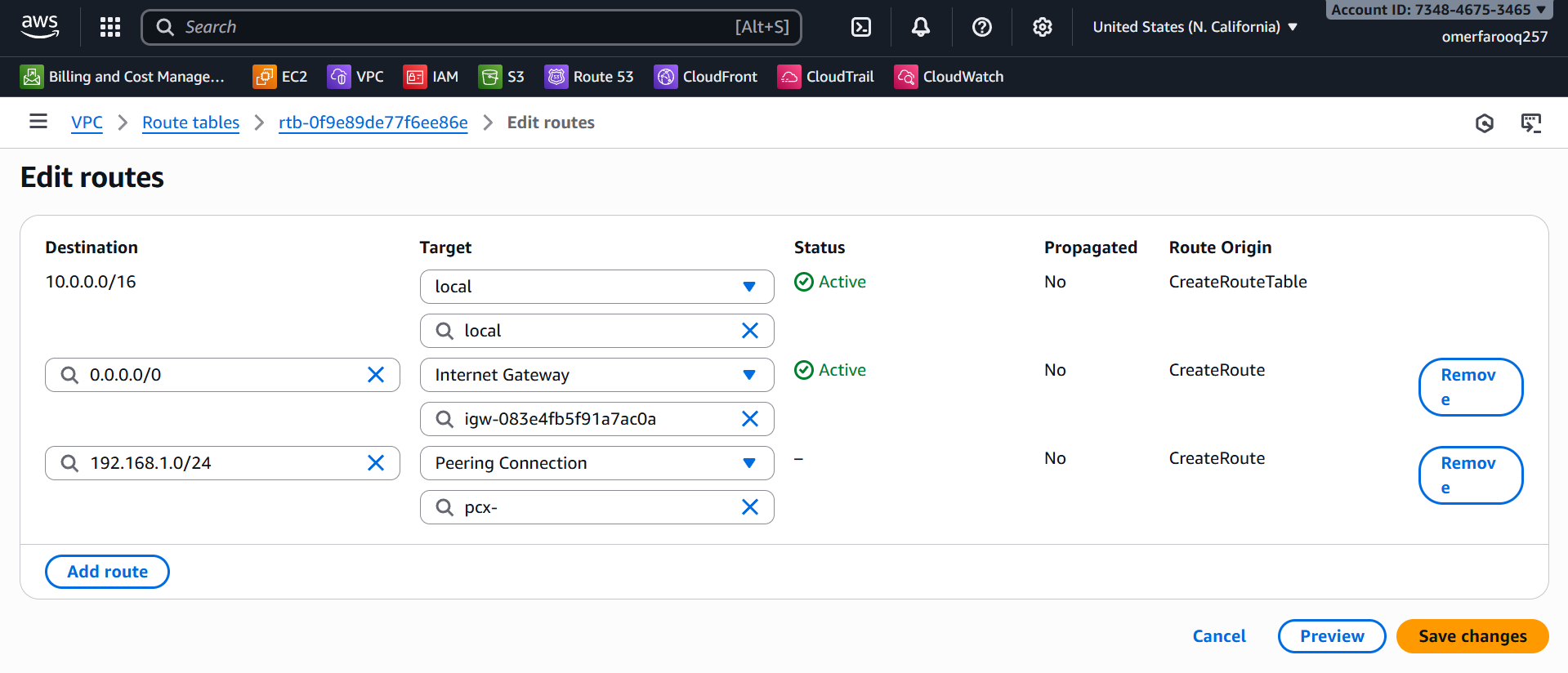


* Click **Add Route**
  + Destination: **10.0.0.0/16**
  + Target: **Peering Connection → Peering-NV-NC**
* Save
* Repeat for NV-Private-RT



**Update Route Tables (Accepter Side - us-west-2)**

* Switch to us-west-2 region
* Navigate to **VPC → Route Tables**
* Select the west subnet's route table
* Edit Routes
* Add route:
  + Destination: **192.168.1.0/24**
  + Target: **Peering Connection → East-West-Peering**

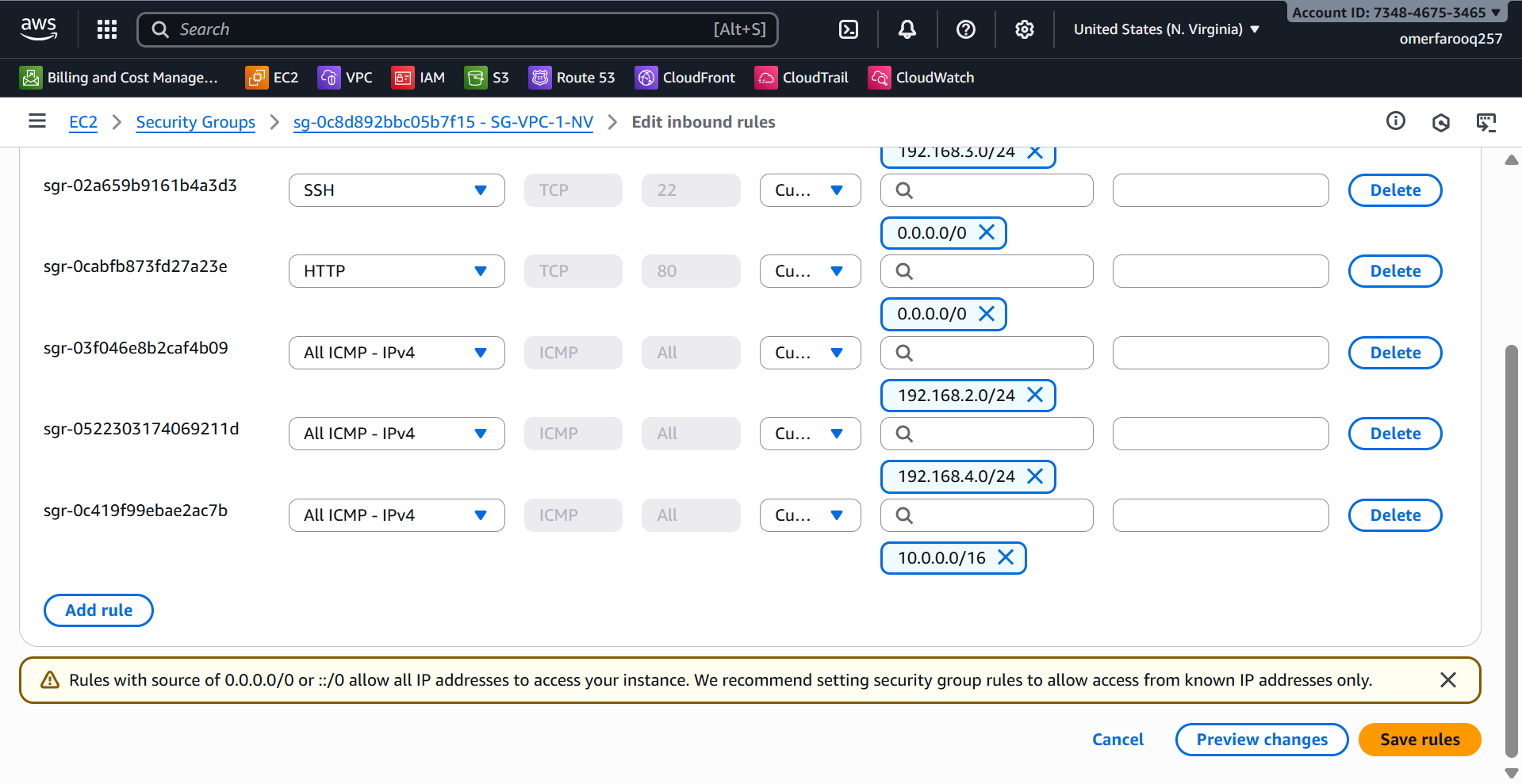
****

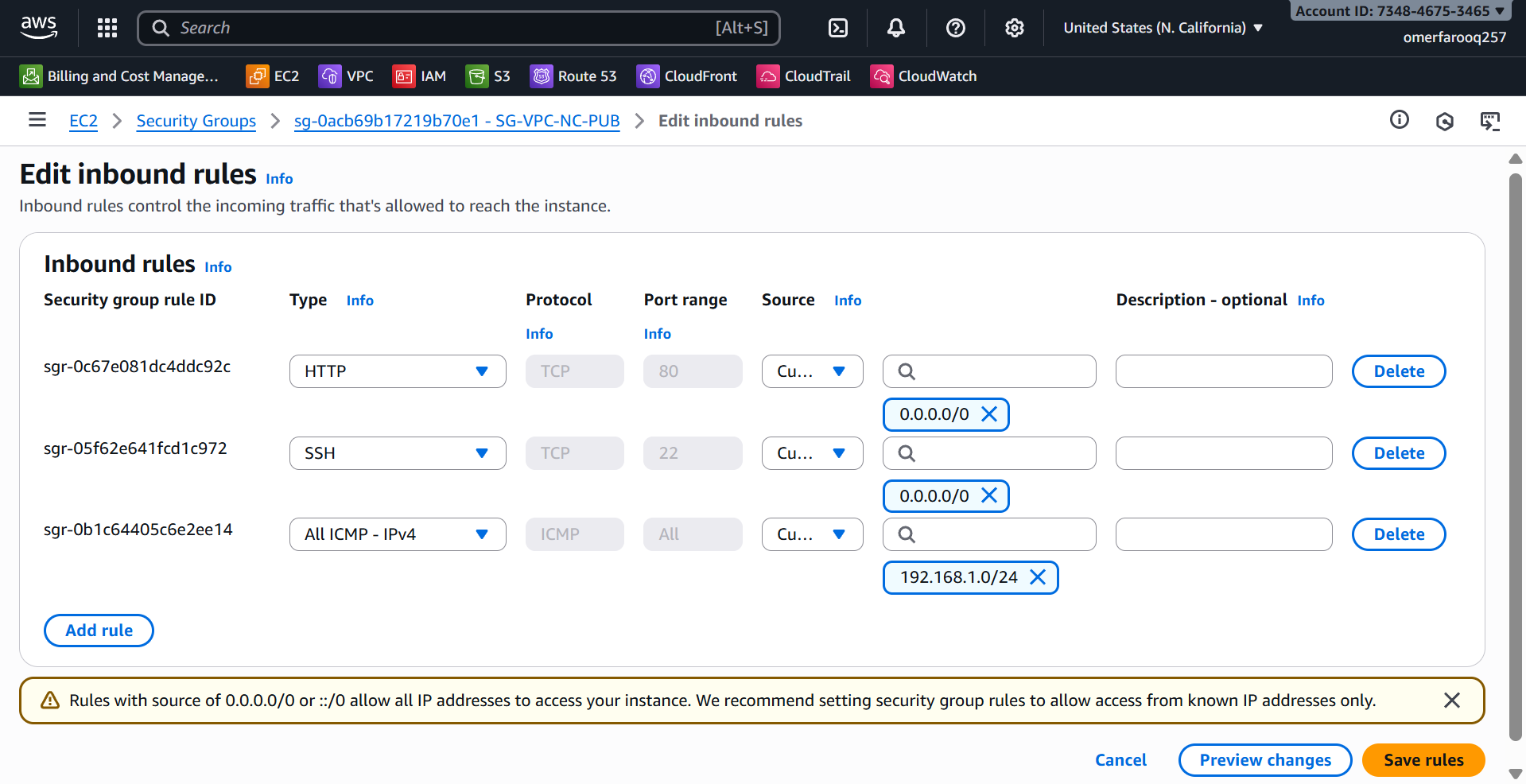
**Update Security Groups for Peering for instances in NV-VPC (us-east-1):**

* Click on **Add rule**
  + Protocol: **All ICMP – IPv4**
  + Custom CIDR: **10.0.0.0/16**

**Update Security Groups for Peering for instances in NC-VPC (us-west-1):**

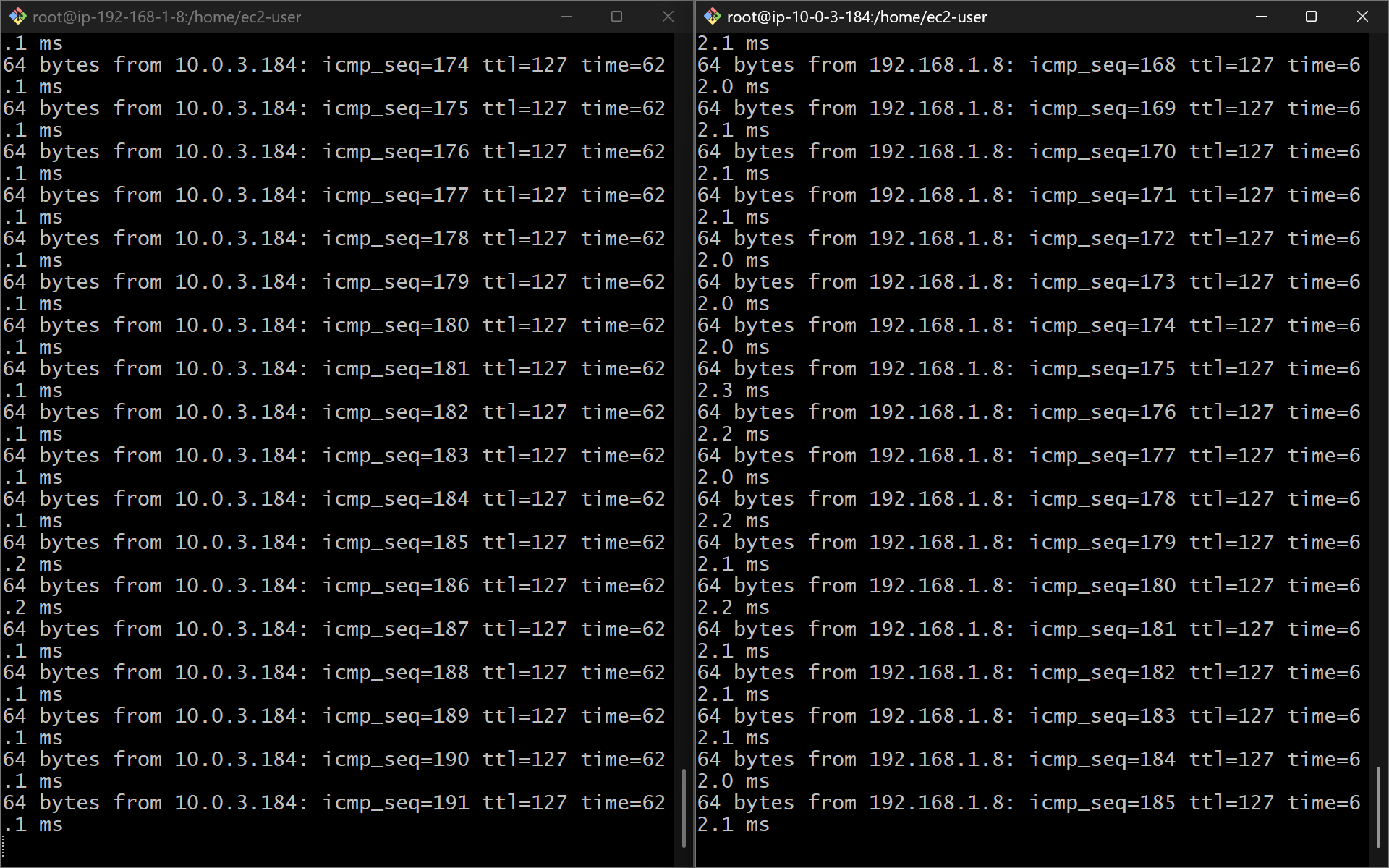
* Click on **Add rule**
  + Protocol: **All ICMP – IPv4**
  + Custom CIDR: **192.168.1.0/16**





**Test the connection between two VPCs in different regions:**

* Launch an EC2 instance in N. Virginia and another in N. California.
* Configure the instances and assign the appropriate:
  + VPC
  + Subnet
  + Key
  + Security Group
* Launch the instances in Git Bash and check the connectivity between the two servers.
  + # ping 10.0.3.184
  + # ping 192.168.1.8

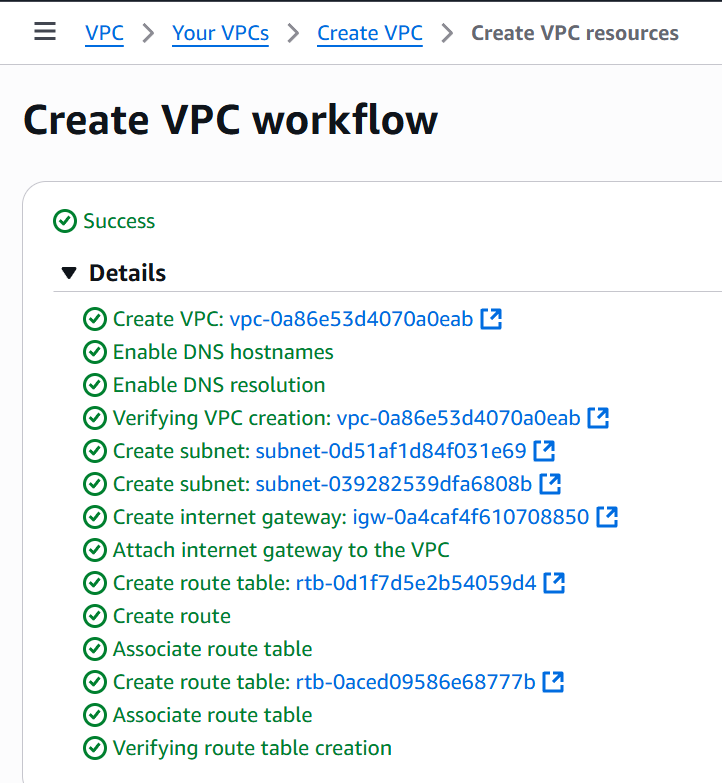


1. **Enable VPC peering for cross-account. (You can collaborate with your friend and do this task.)**
2. **Set up a VPC Transit gateway.**

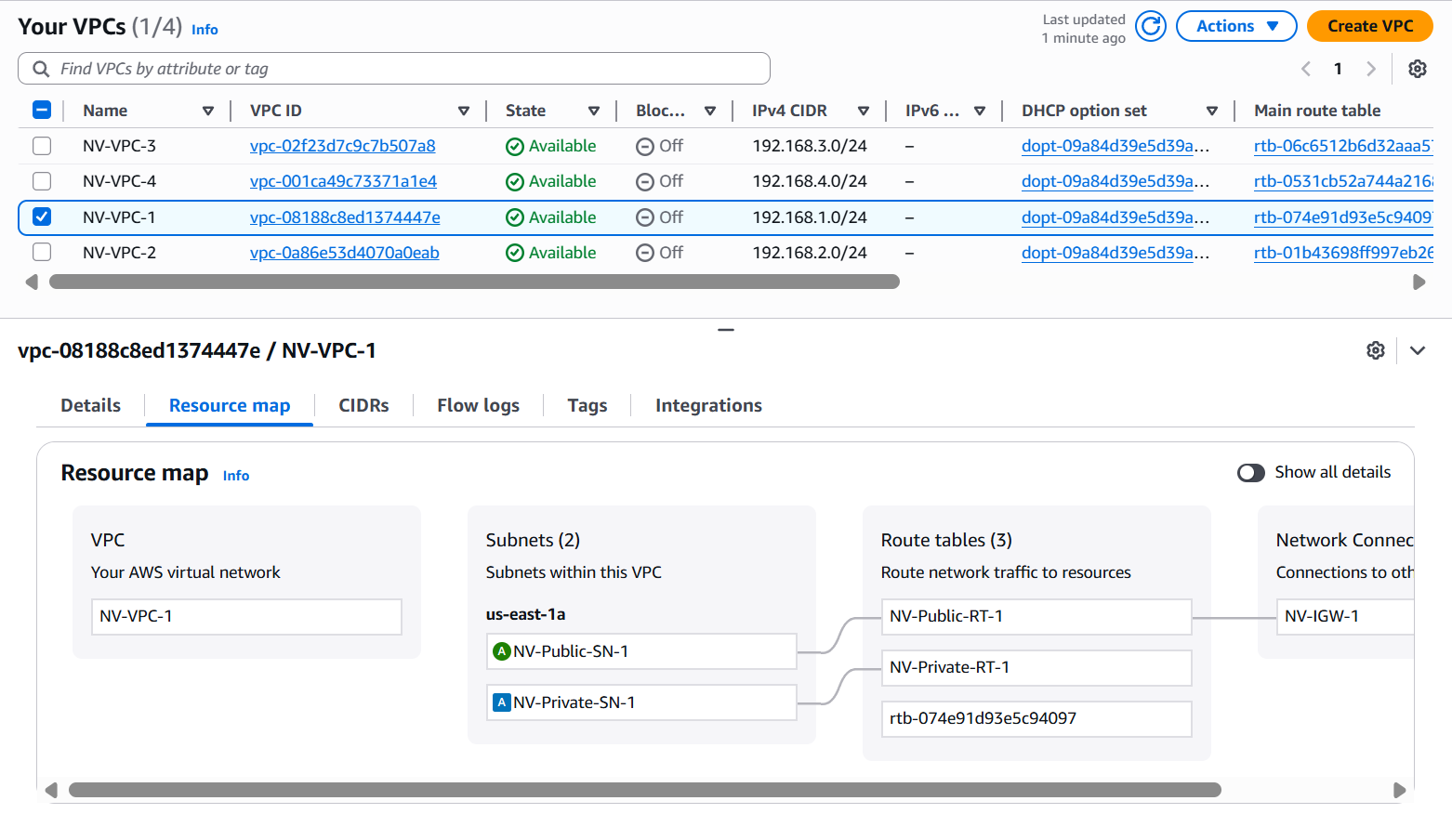
**Purpose:** To simplify VPC-to-VPC and VPC-to-on-premises connectivity. Alternative to VPC Peering, and when planning to add more VPCs. Easier than managing multiple peering connections.

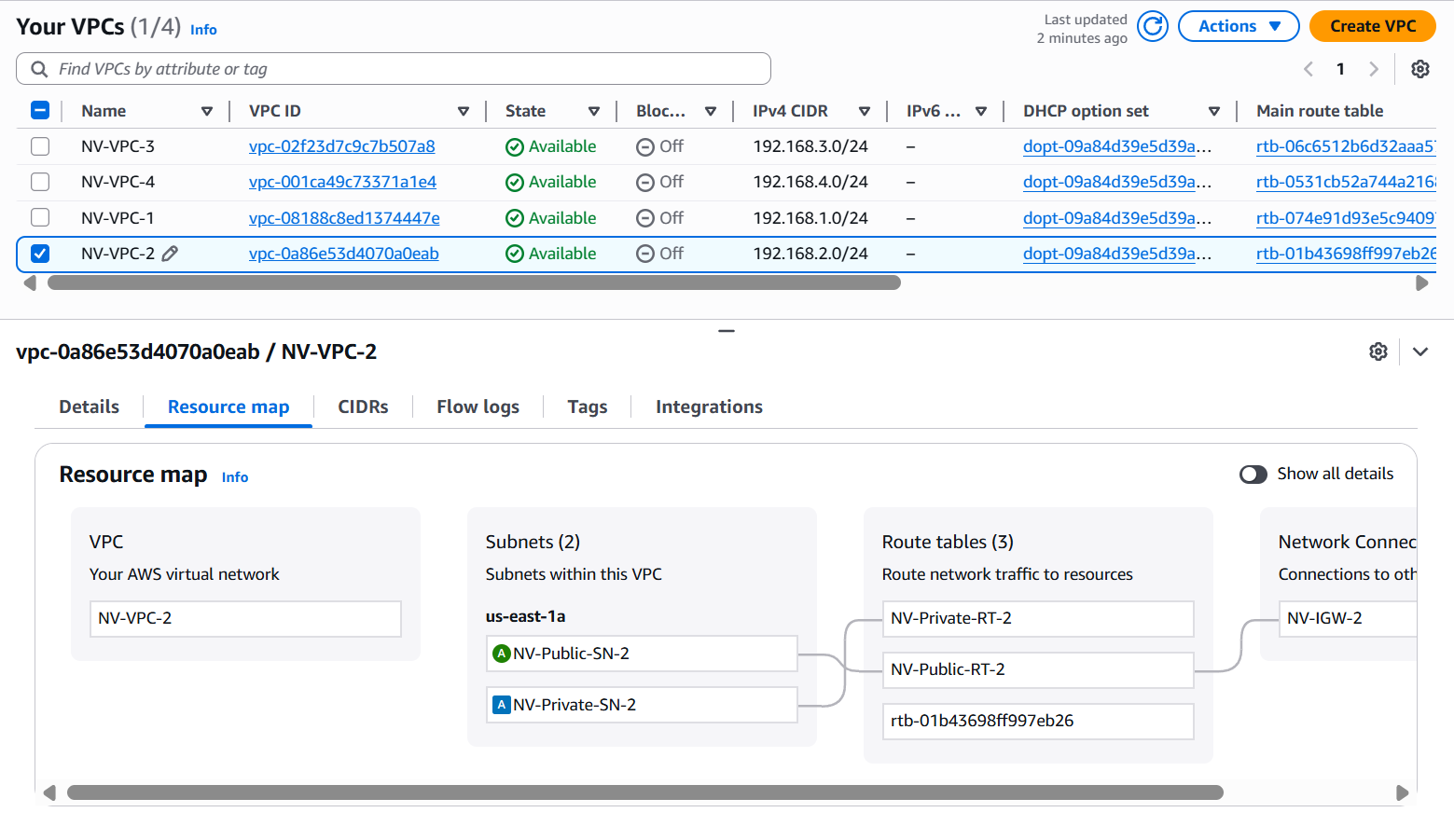
*STEPS TO CREATE A TRANSIT GATEWAY:*

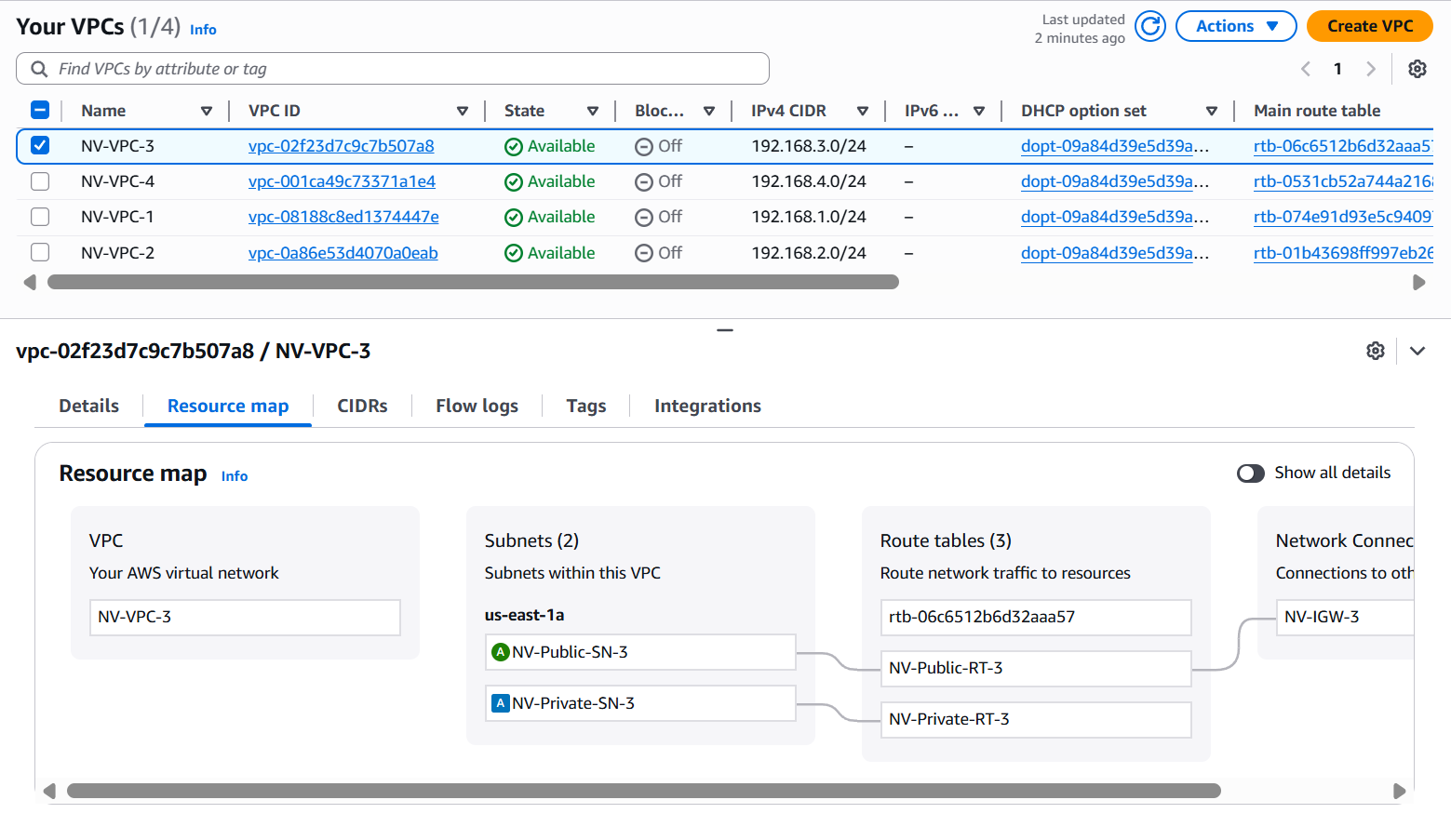
**Create 4 VPCs in any region. In this task, I have created four VPCs in the North Virginia region.**

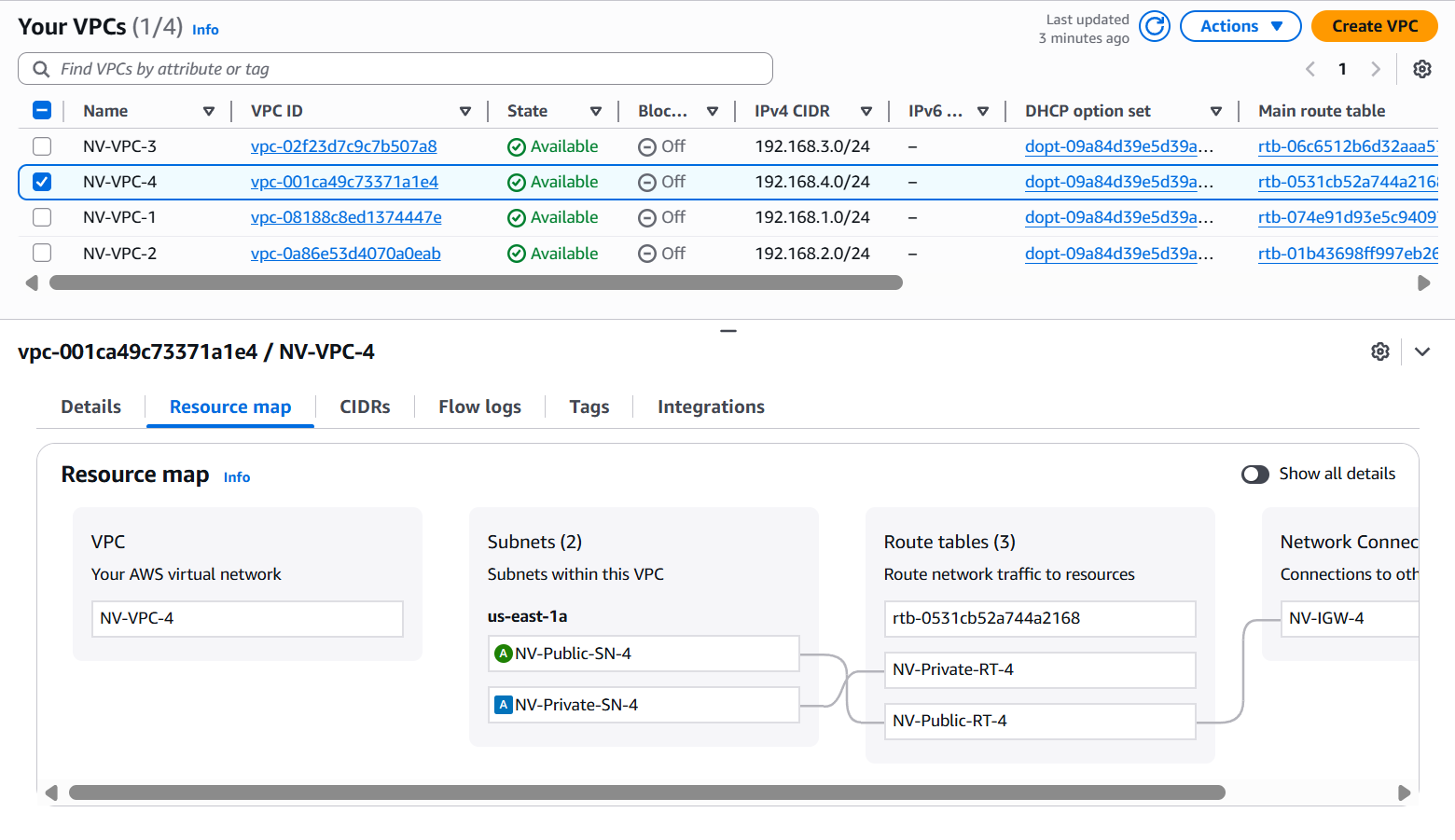


**The resource map of all four VPCs is attached below.**



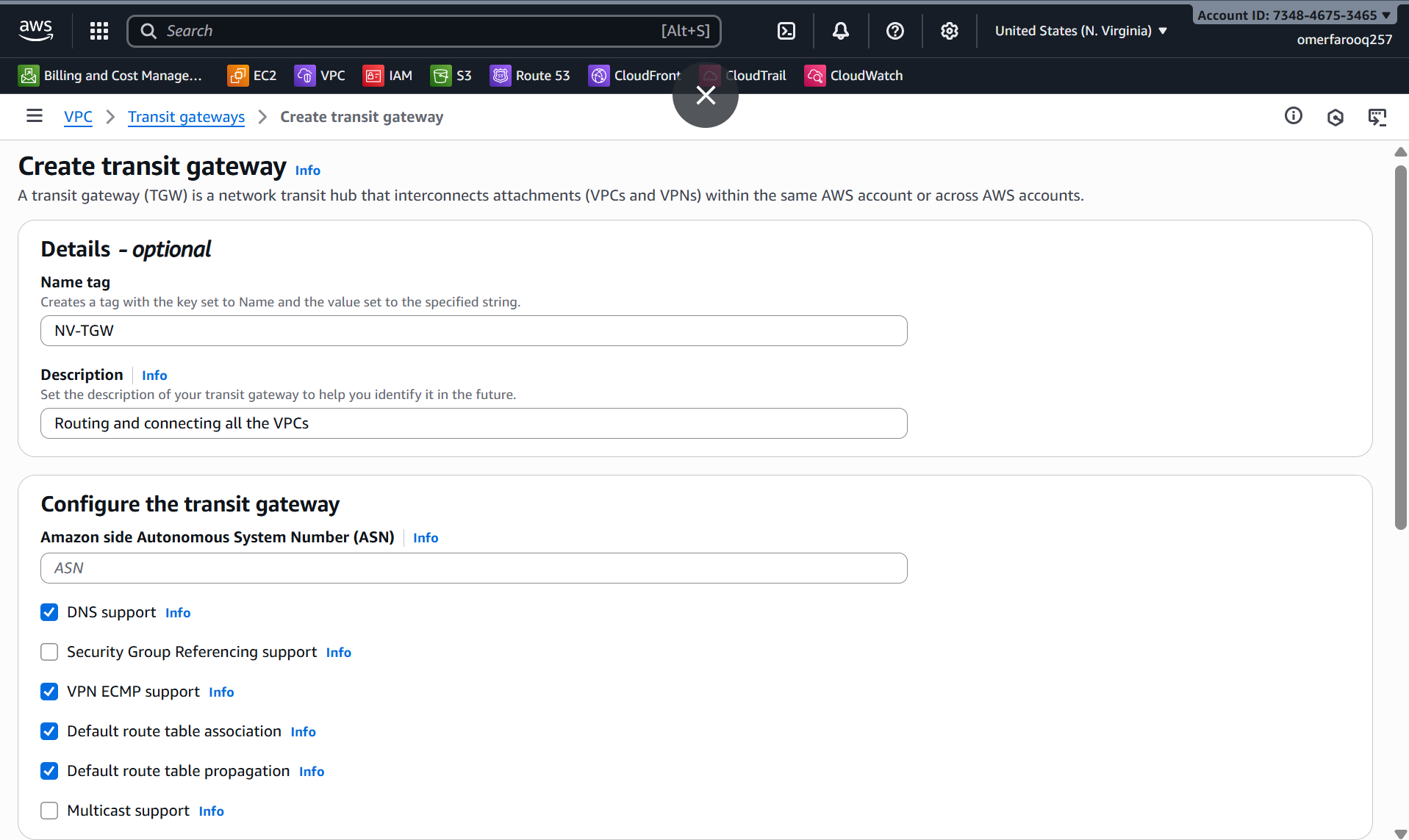




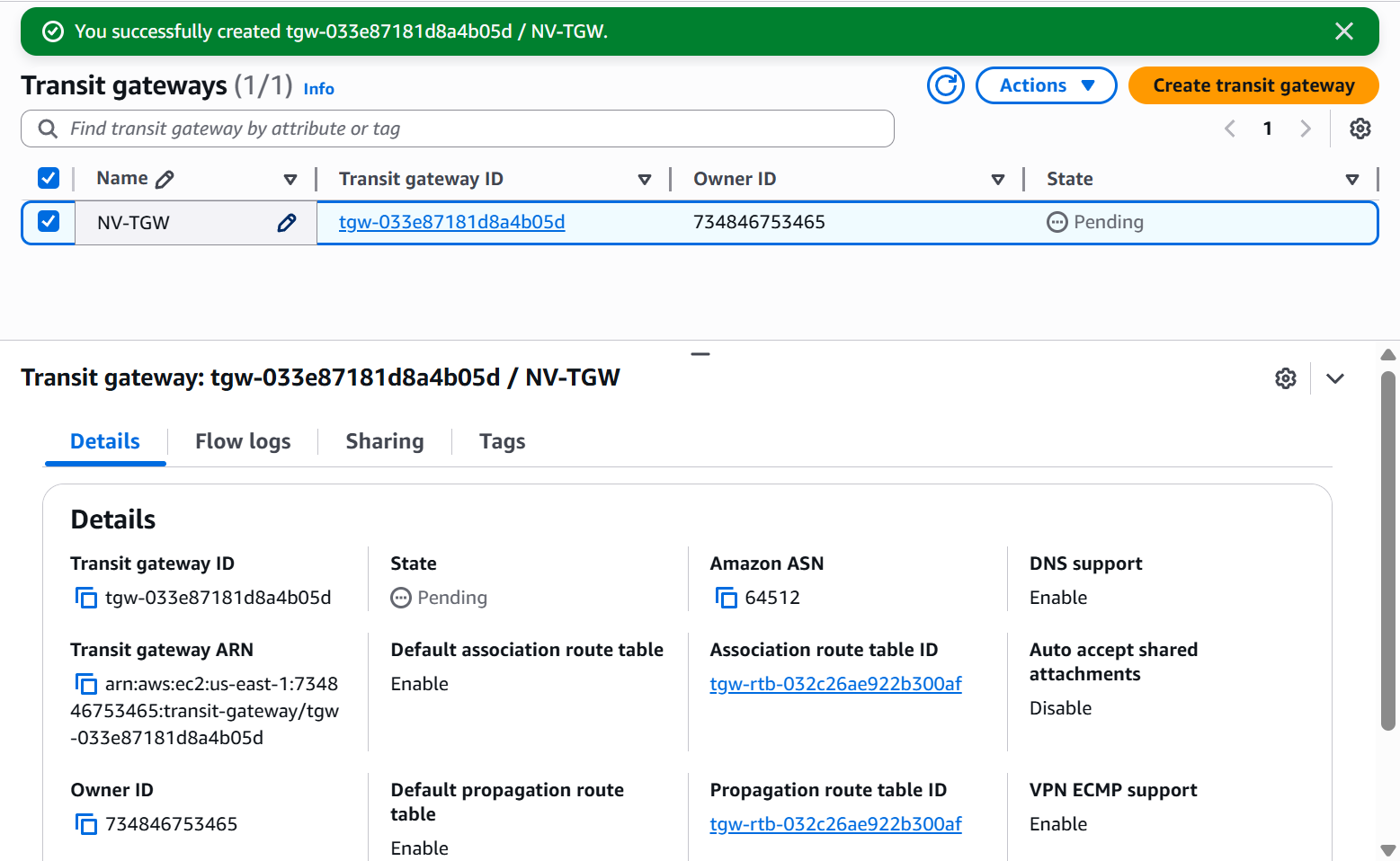


**Creating a Transit Gateway:**

* Region: United States (N. Virginia)
* Navigate to **VPC → Transit Gateway**
* Click on **Create transit gateway**
* **Configuration:**
  + DNS support: ✓Enable
  + VPN ECMP support: ✓ Enable
  + Default route table association: ✓ Enable
  + Default route table propagation: ✓ Enable

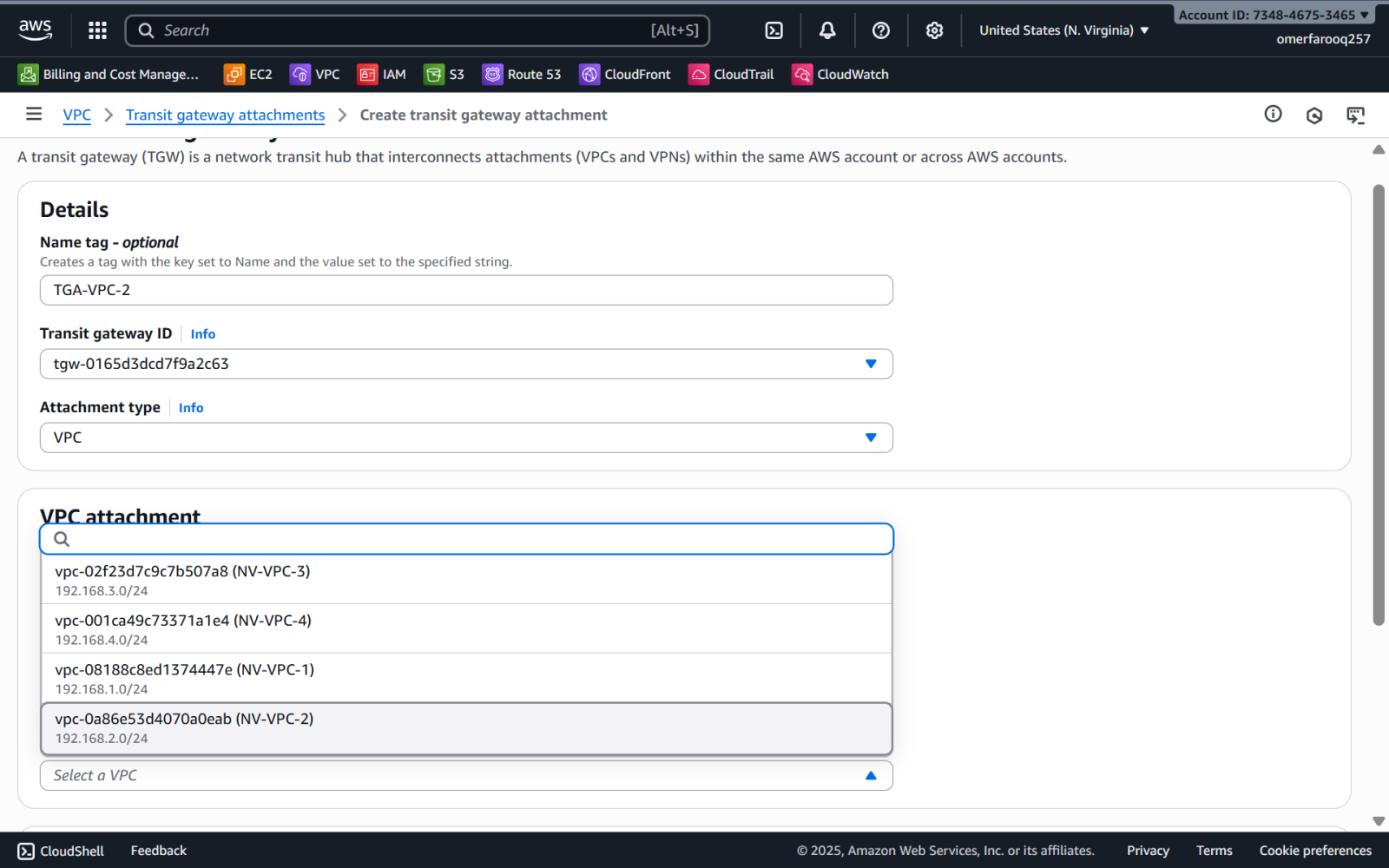


* Click **"Create transit gateway"**
* **Wait 5-10 minutes** for state: **"available"**
* **Note Transit Gateway ID**: tgw-california-xxxxx



**Create VPC Attachments:**

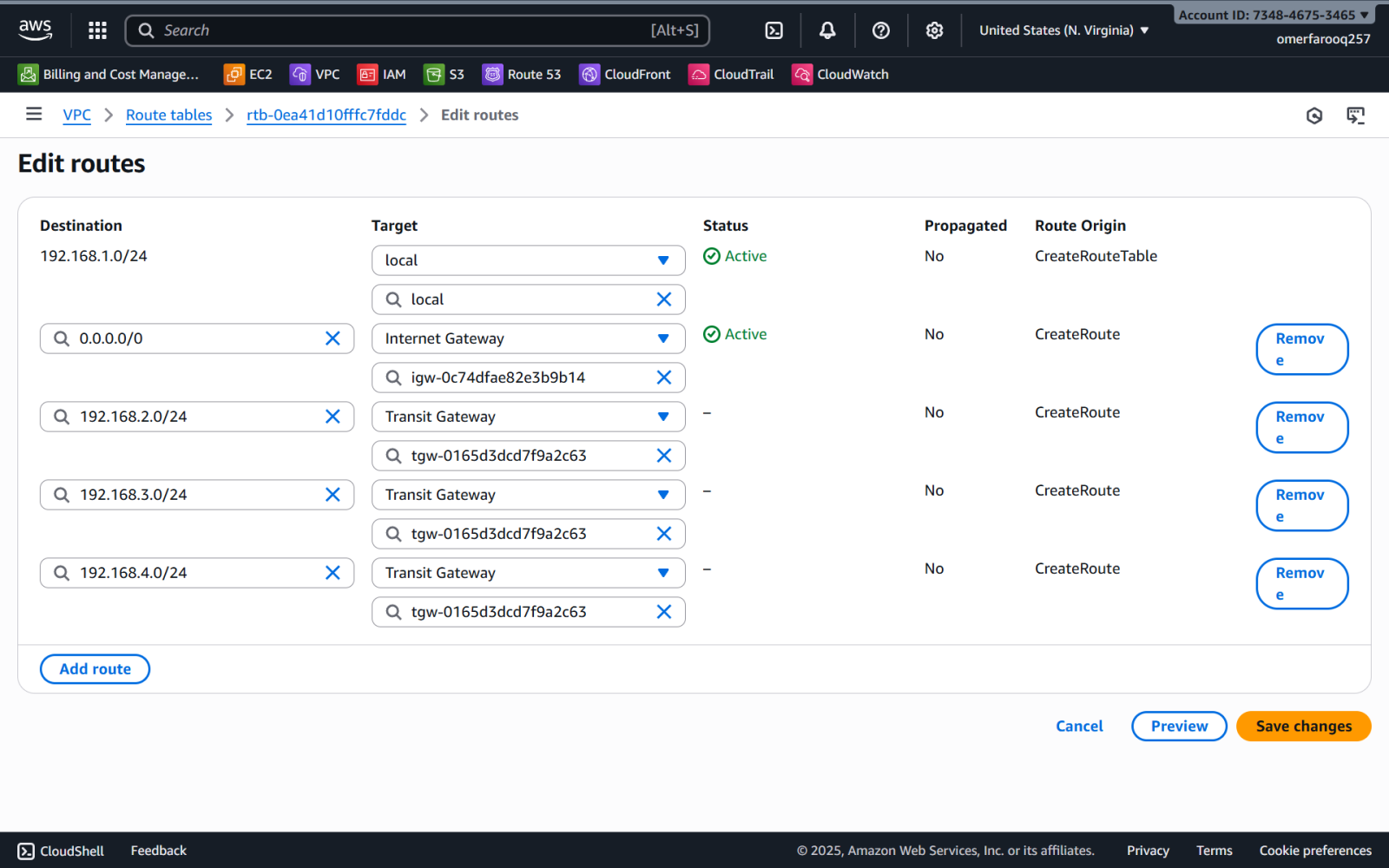
* VPC Dashboard → **"Transit gateway attachments"**
* Click **"Create transit gateway attachment"**
* Configure Attachment
  + Name tag: **TGA-VPC-1**
  + Transit gateway ID: **tgw-0165d3dcd7f9a2c63**
  + Attachment type: **VPC**
  + DNS support: ✓ **Enable**
  + VPC ID: **NV-VPC-1 (192.168.1.0/24)**



* Repeat the same steps for the remaining VPCs to create transit gateway attachments.
* Configure the attachment with the appropriate VPCs.

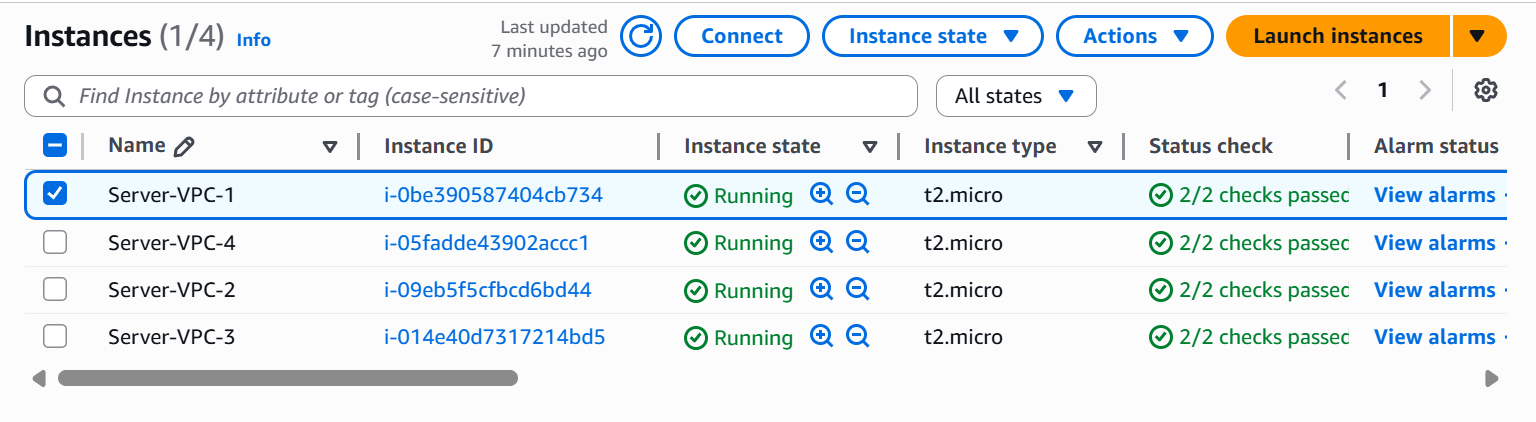
**Configure the route tables for each VPC.**

* Navigate to **"Route tables"**
* Select **NV-Public-RT-1 & NV-Private-RT-1**
* **"Routes"** → **"Edit routes"**
* Configure:
  + Destination: 192.168.2.0/24 Target: Transit Gateway → TGW-NV (VPC-2)
  + Destination: 192.168.3.0/24 Target: Transit Gateway → TGW-NV (VPC-3)
  + Destination: 192.168.4.0/24 Target: Transit Gateway → TGW-NV (VPC-4)

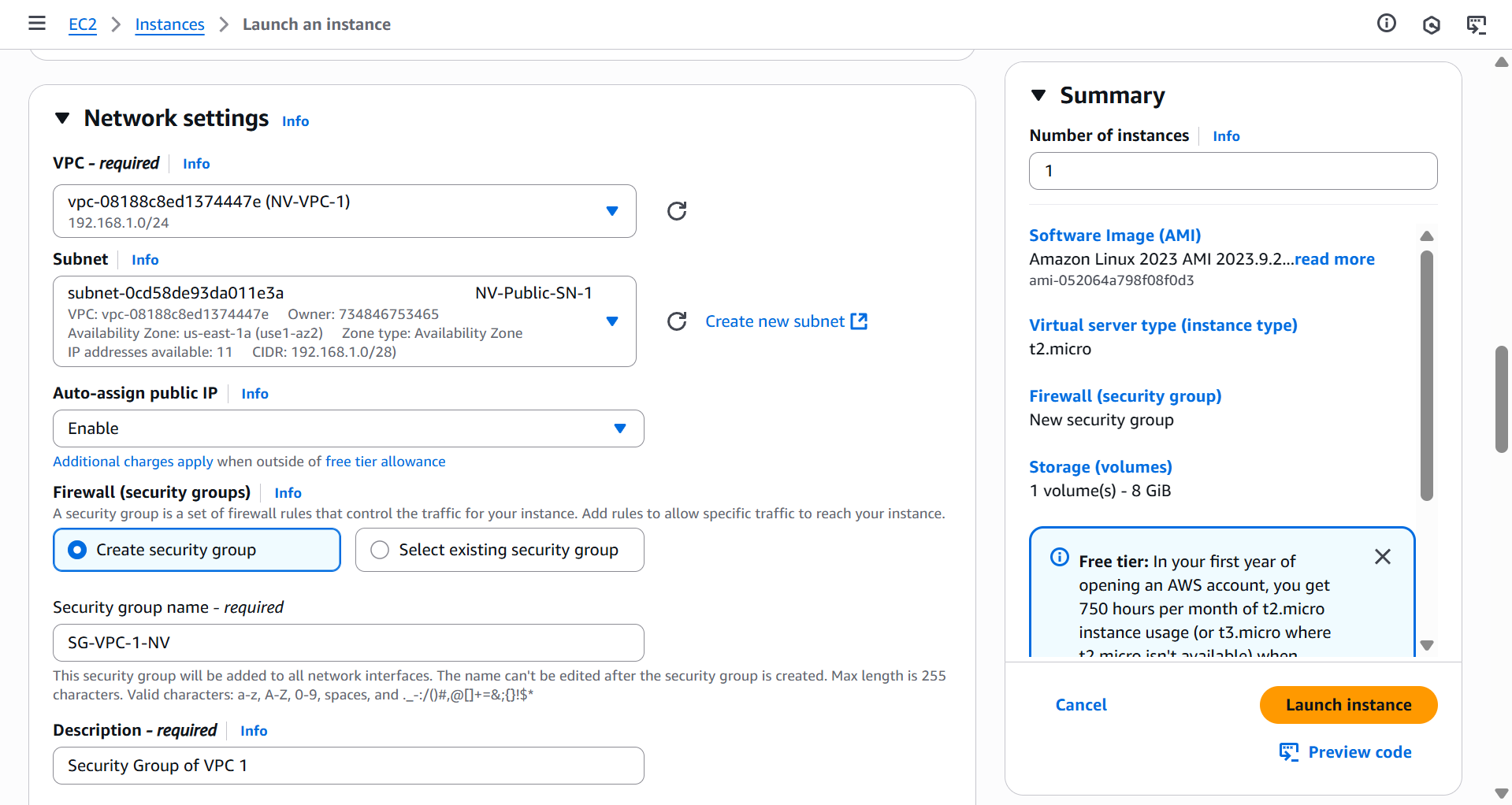


* Repeat the above steps for the remaining VPCs’ Route Tables.
* Make sure to use the appropriate CIDR Range.

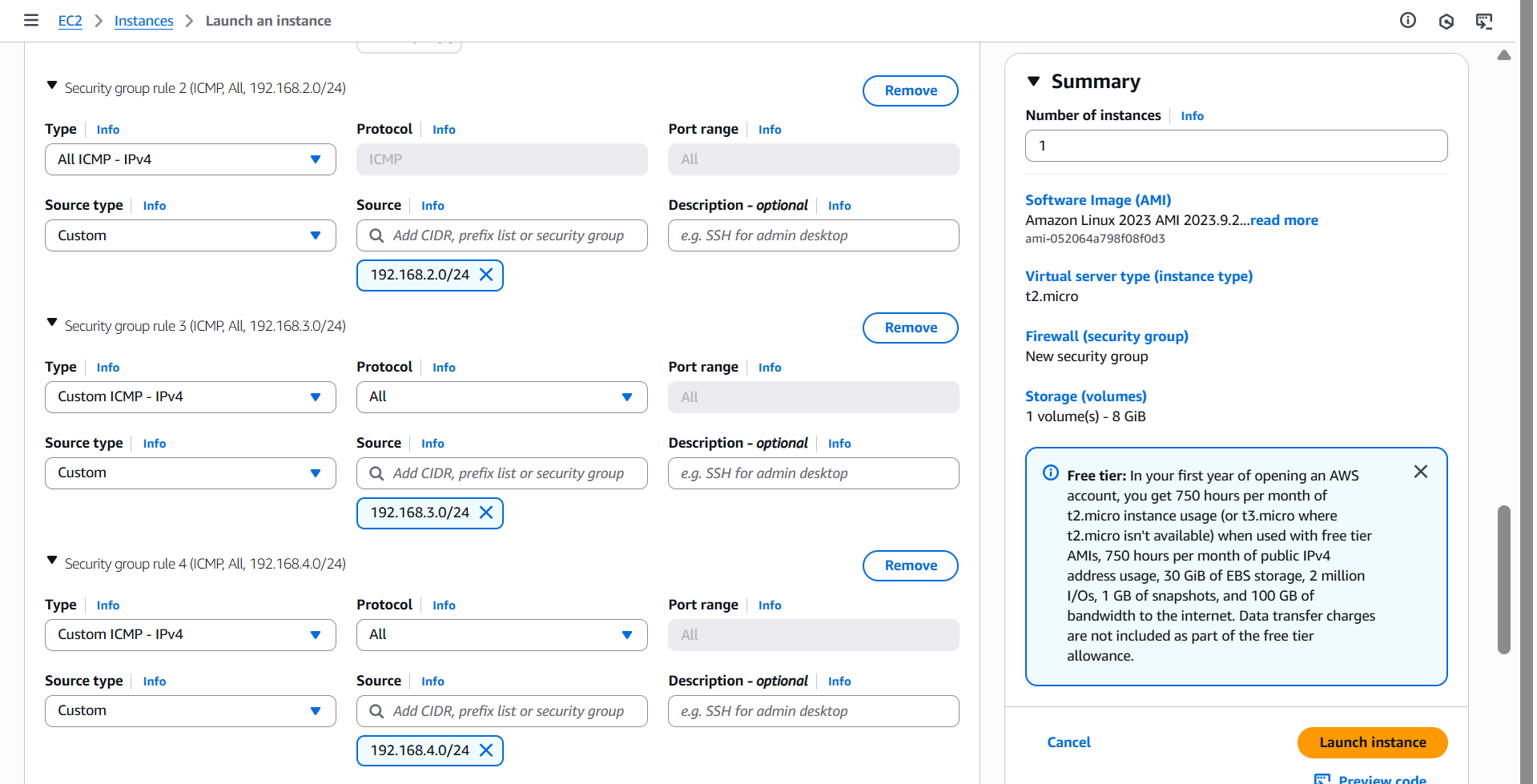
**Launch EC2 Instances and Update Security Groups:**



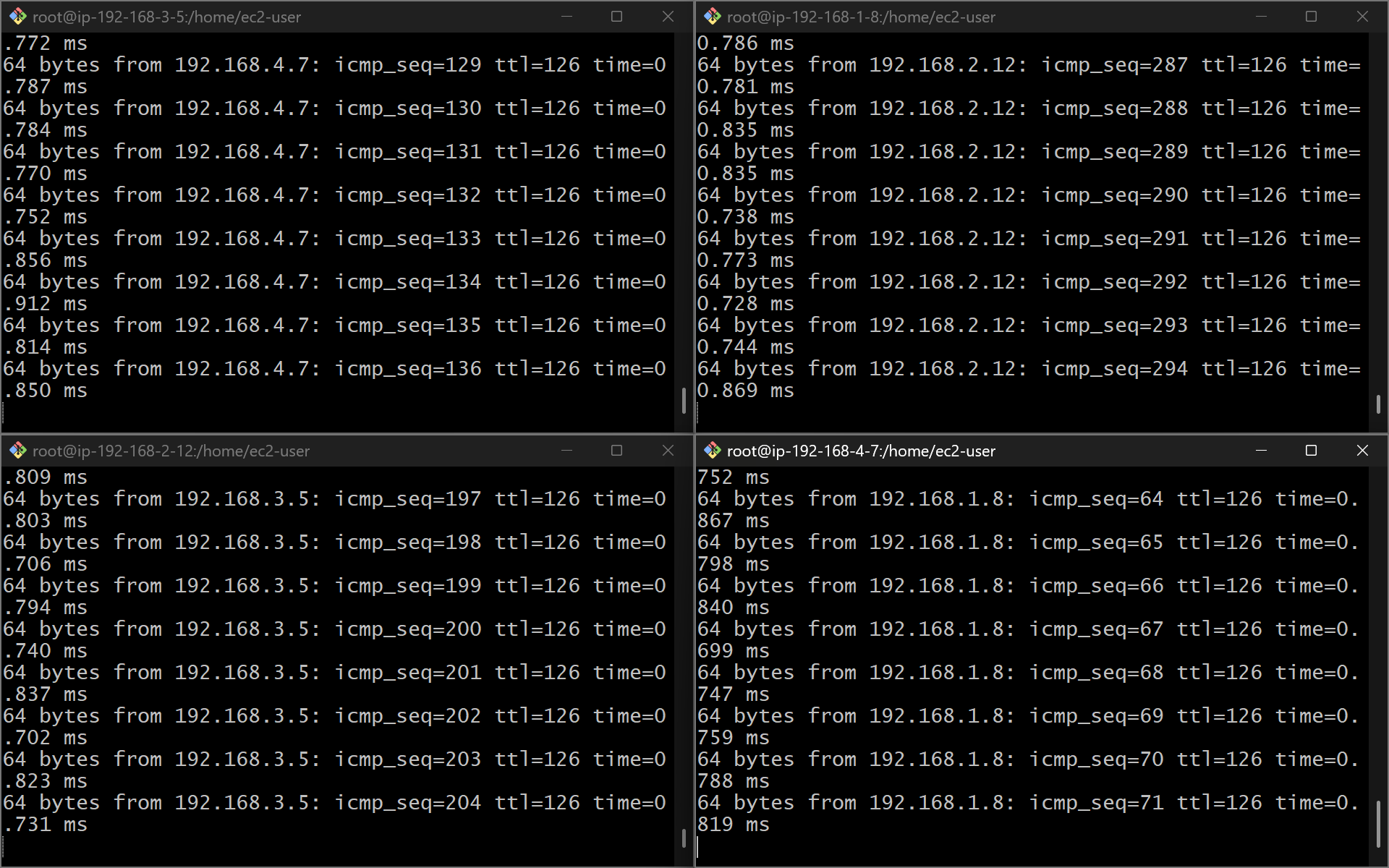
* EC2 **→** Instances **→ Launch Instance**
* Basic configurations:
  + Name: Server-VPC-1
  + Key pair: key.pem
  + VPC – NV-VPC-1
  + Subnet: NV-Public-SN-1



* Configure the **Network Settings** **→ Firewall (Security Groups)** as shown below and save the security group for future use.
* We can create the Security Groups separately, but to reduce a step, I have created and configured them here while launching an instance.



* Repeat the steps and launch a total of four instances, one in each VPC’s public subnet.
* Once all the configurations are done, test the connectivity by pinging each other.
* If everything goes right, the output will look like the image attached below.

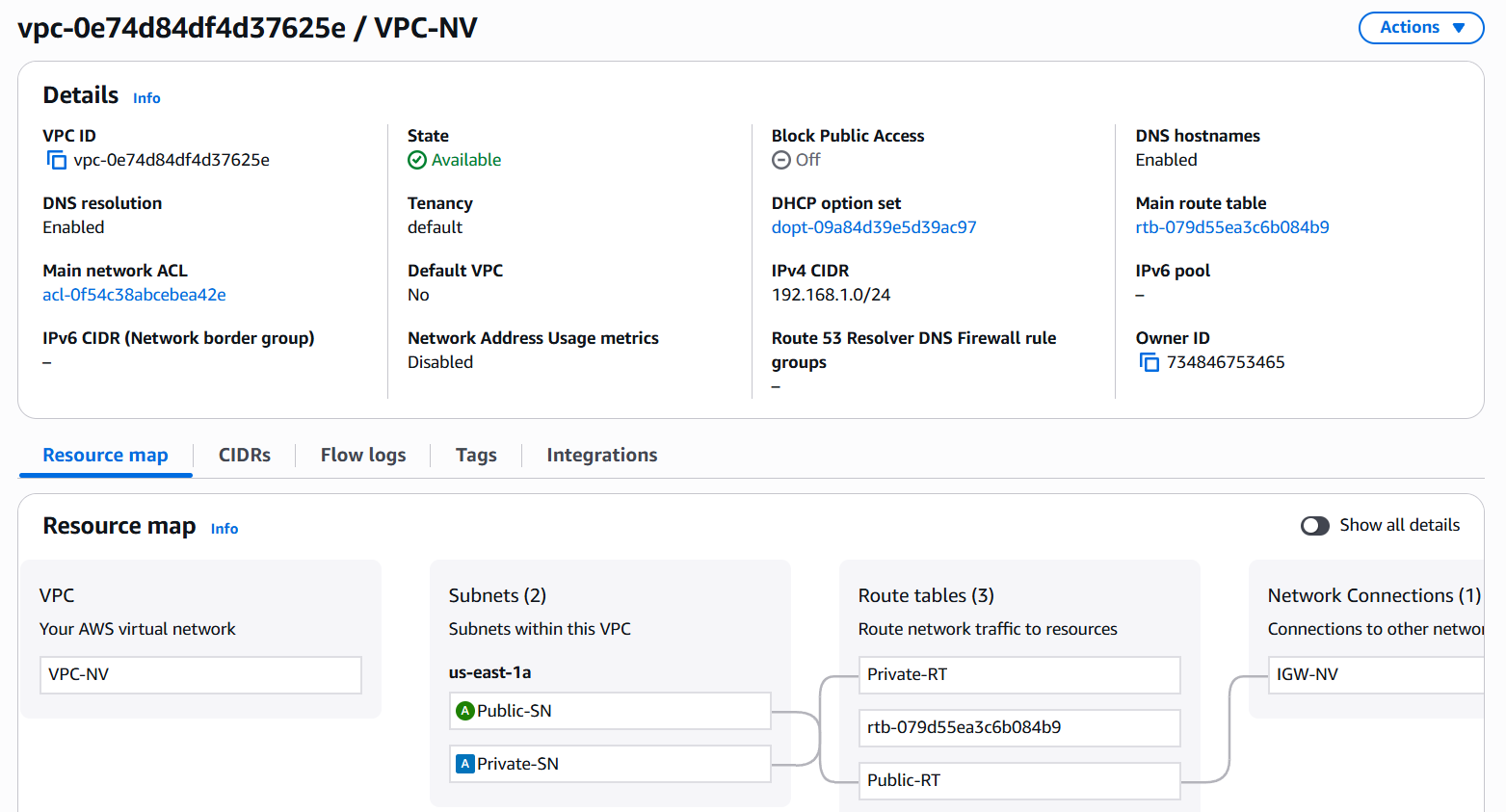


1. **Set up VPC End Point.**

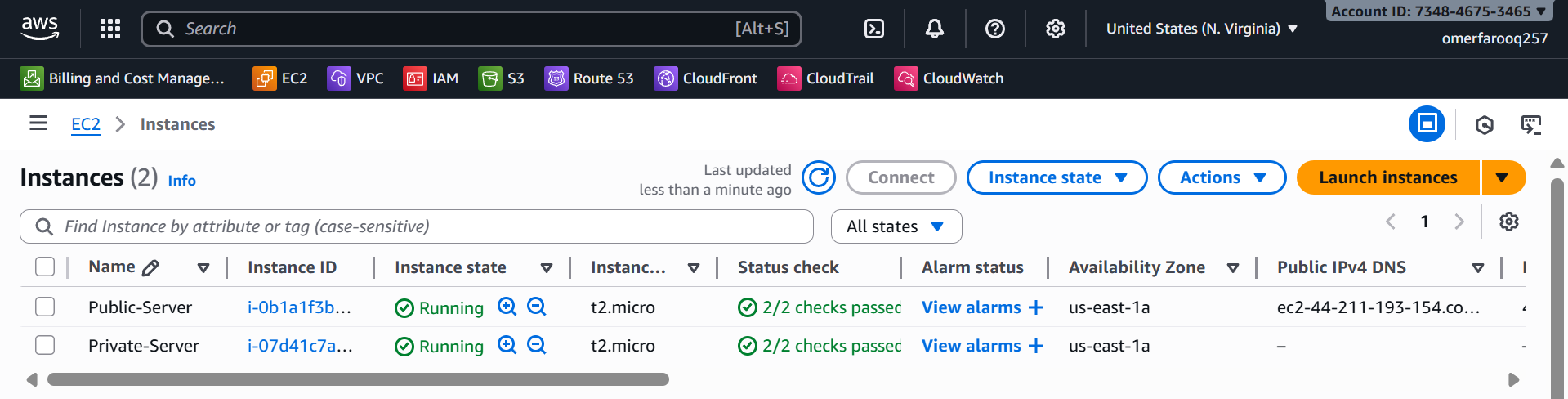
**Purpose:** To access AWS services privately, avoiding internet exposure and reducing data transfer costs. Eliminate NAT Gateway charges for AWS services and keep traffic within the AWS network.

*STEPS TO CREATE AND SET UP A VPC ENDPOINT:*

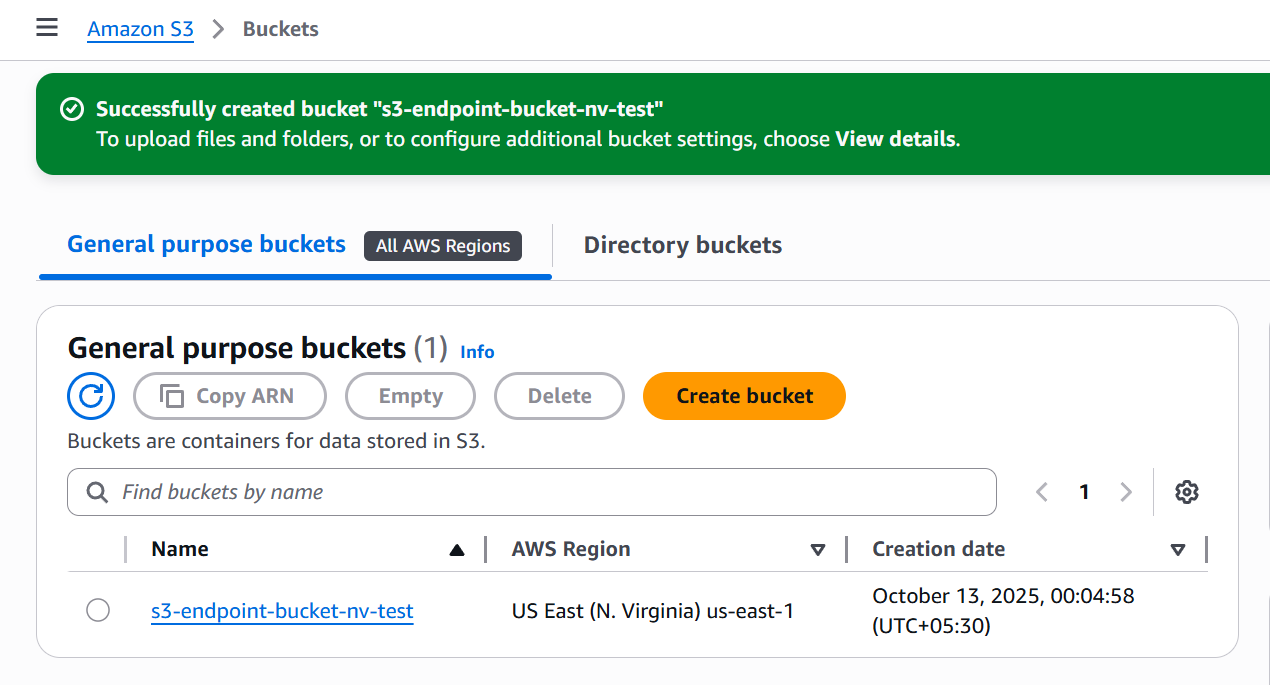
**Create a VPC, Subnets (Public & Private), a Routing Table, and an Internet Gateway.**

****

**Create two Instances – a Public and a Private.**

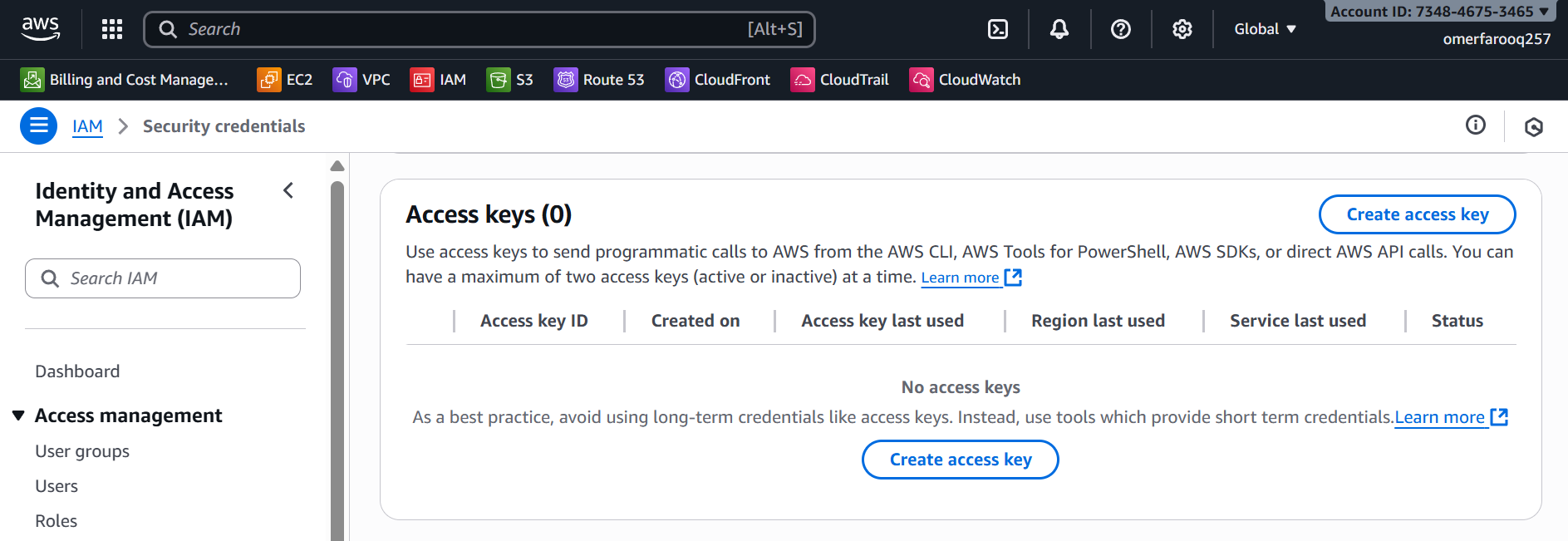
****

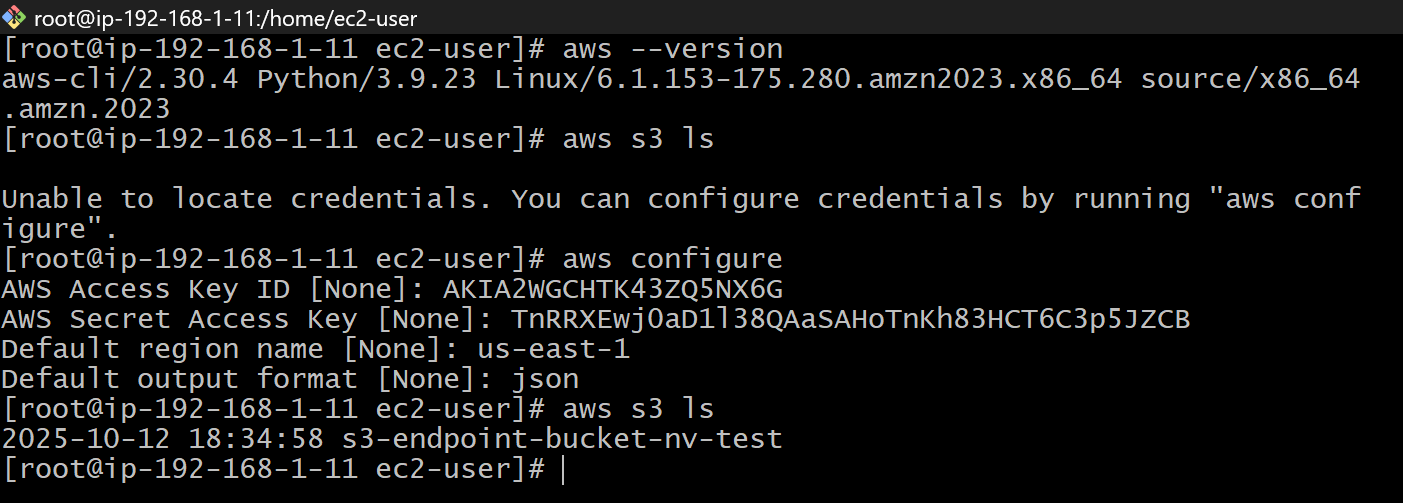
**Create an S3 bucket.**

****

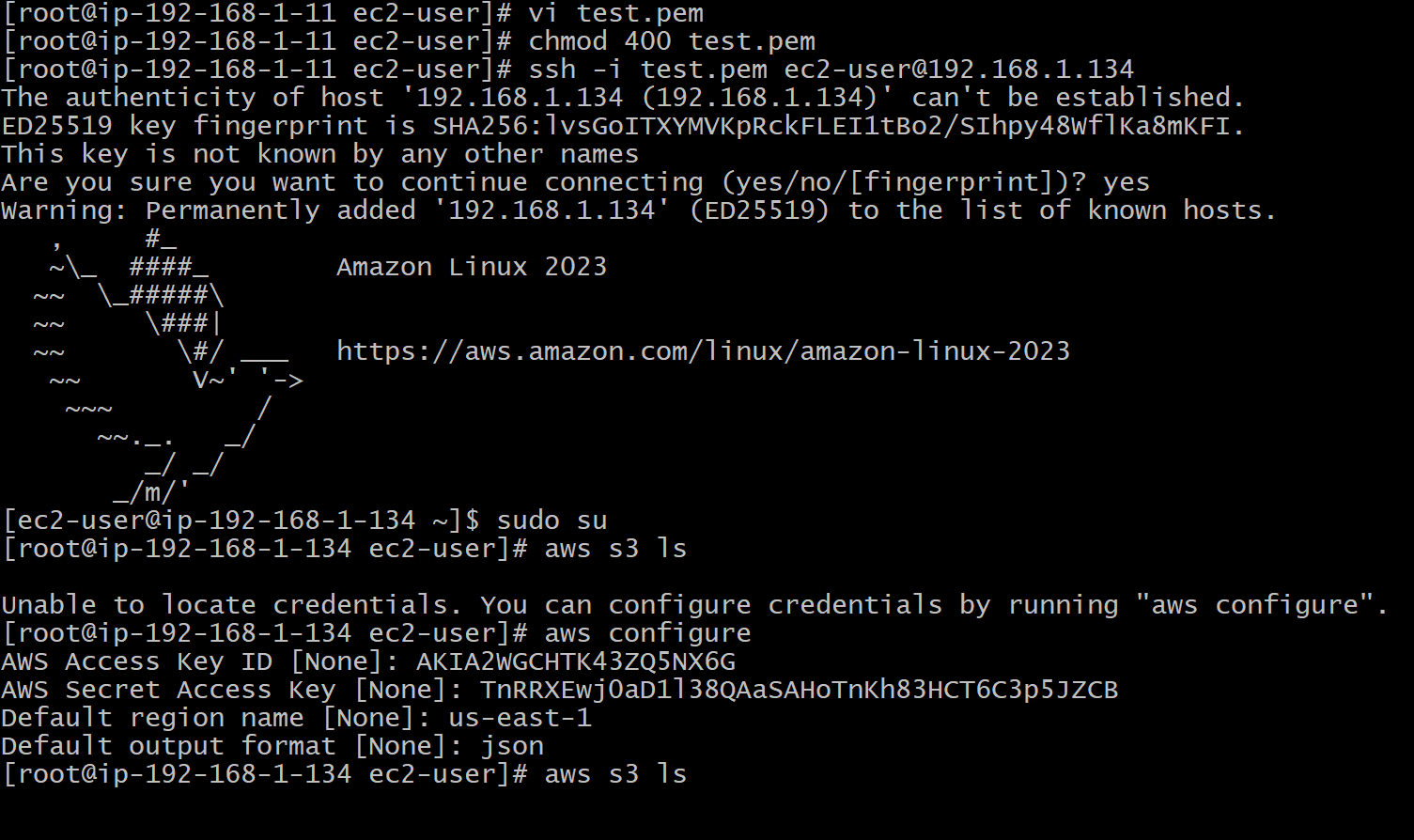
**Generate Access keys to configure AWS CLI in the Instances.**

* Navigate to **IAM → Access Management → Access keys → Create access key**

****

****

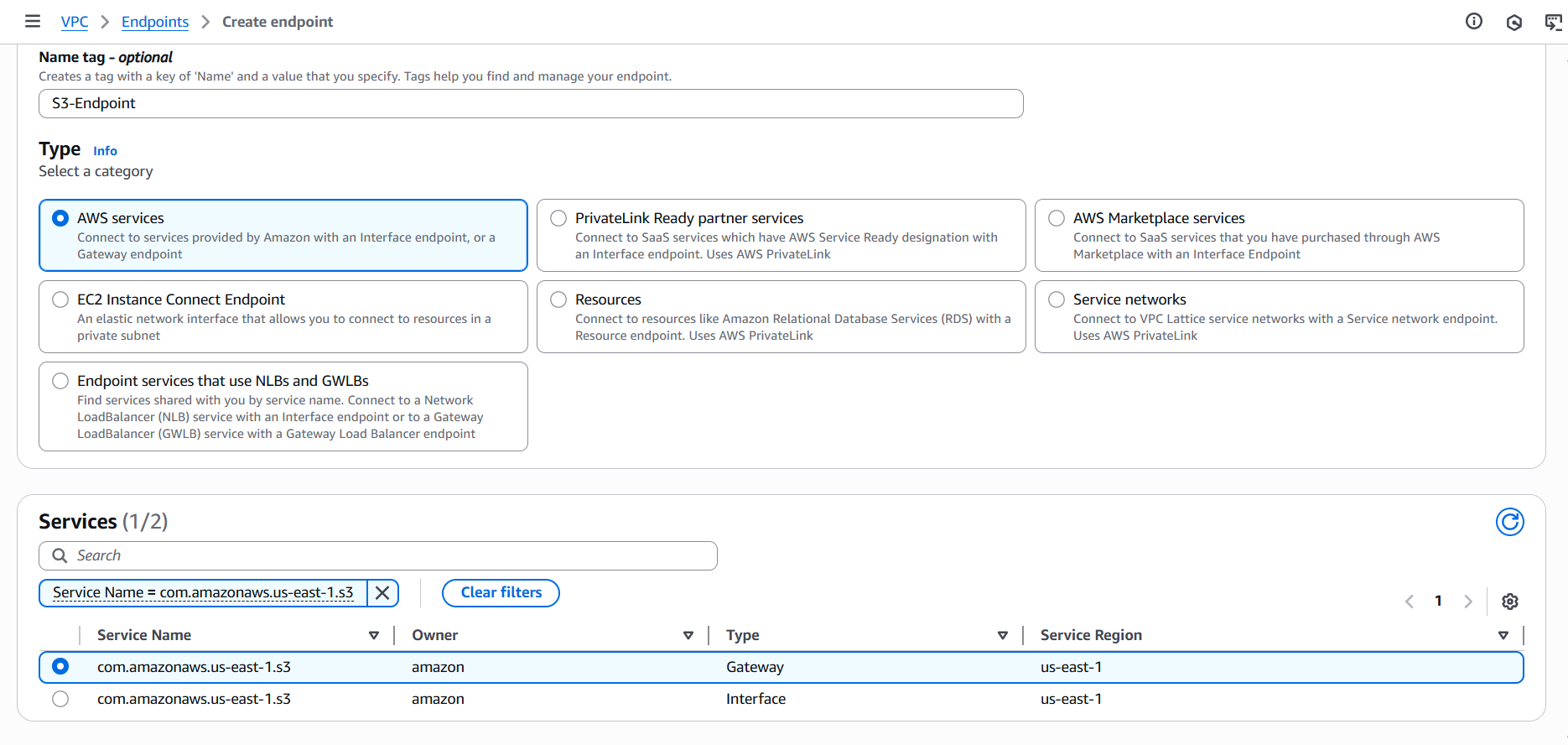
* Access **Private Instance** using **Bastion/Jump Server**.
* Configure AWS CLI in the Private Instance.

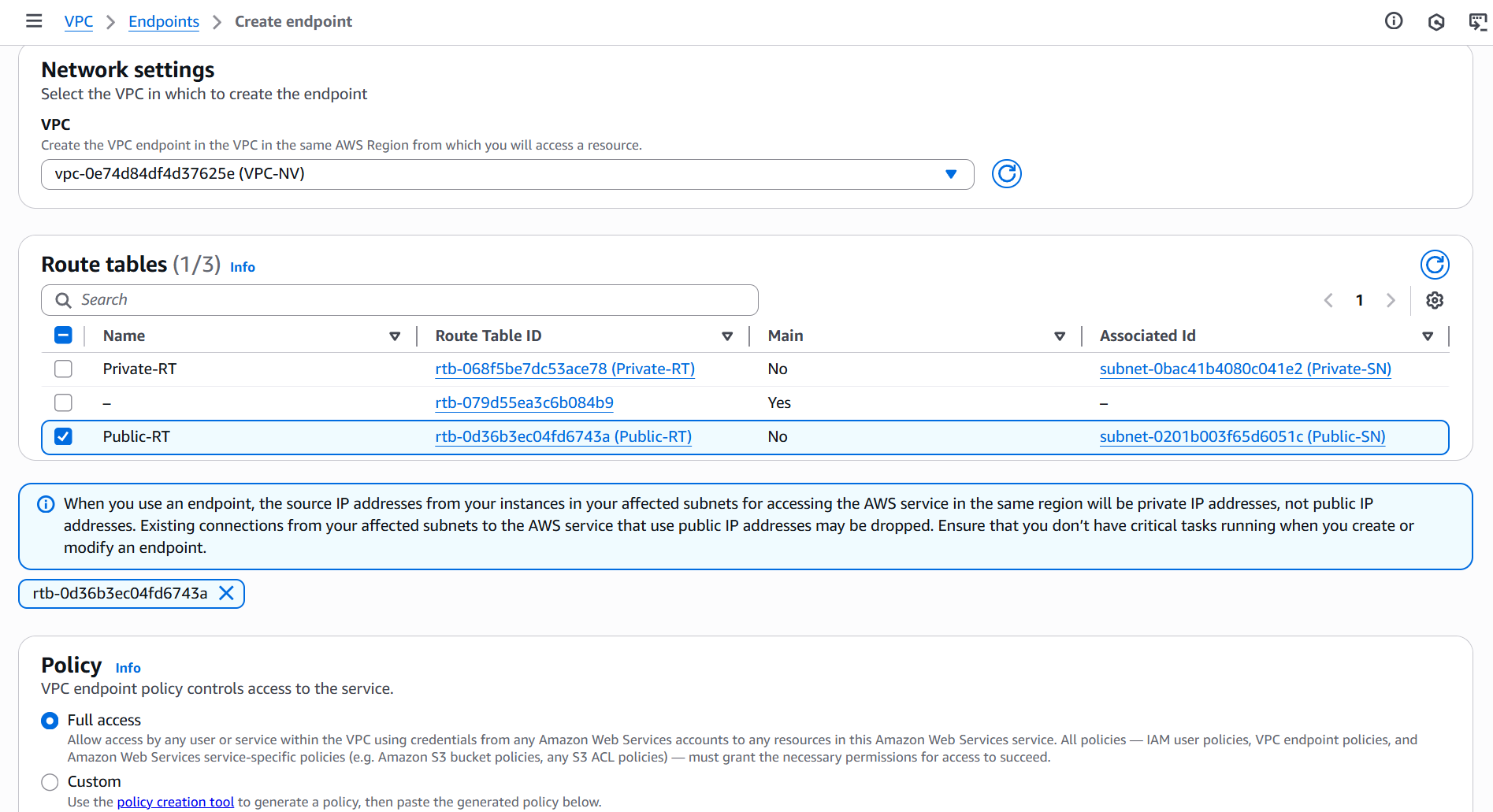
****

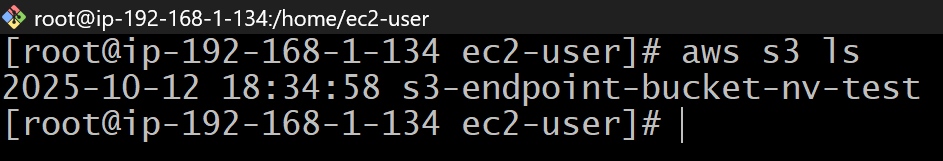
**Creating Endpoint:**

* Navigate to **VPC → Endpoints**
* Click **Create Endpoint**
* **Configuration:**
  + **Name tag:** S3-Endpoint
  + **Service category: AWS services**
  + **Service name: com.amazonaws.us-east-1.s3**
  + **VPC:** NV-VPC (vpc-0a1b2c3d4e5f6g7h8)
  + **Route tables:**
    - **☑** Public-RT (rtb-0x1y2z3a4b5c6d7e8)
    - **☑** Private-RT (rtb-0m1n2o3p4q5r6s7t8)
  + **Policy: Full Access (or custom)**

****

****

****

****

**Verification:**

* From the Private-Server, run the following command:  
  # aws s3 ls

**Output:**

* *2025-10-12 18:34:58 s3-endpoint-bucket-nv-t*