**Design and deploy a scalable network architecture using AWS Transit Gateway to simplify network connectivity between multiple VPCs.**

**Step 1: Plan Your IP Addressing**

**non-overlapping CIDRs from different classes:**

**1.) VPC-A (N. Virginia, us-east-1) → 10.0.0.0/24**

**2.) VPC-B (OHIO -us-east-2) → 172.168.0.0/24**

**3). VPC-C (California, us-west-1) → 192.168.0.0/24**

**4.) VPC-D (Mumbai, ap-south-1) → 172.168.1.0/24**

**Step 2: Create 4 VPCs (Non-default)**

**Create VPCs (in 4 different regions)**

**Go to VPC Console → Your VPCs → Create VPC.**

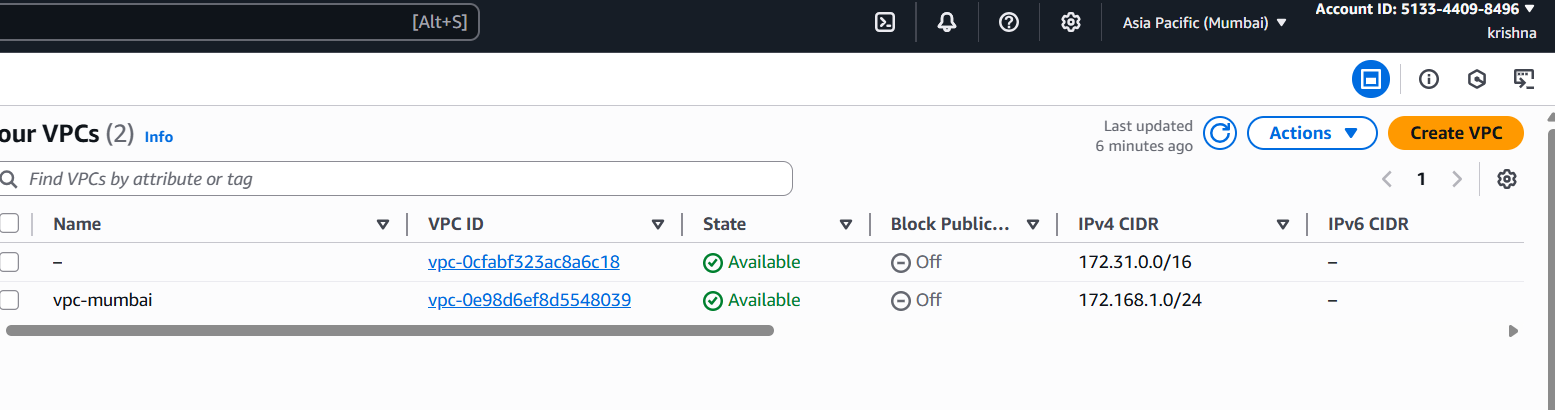
**Create 1 private subnets in each VPC (in different AZs).**

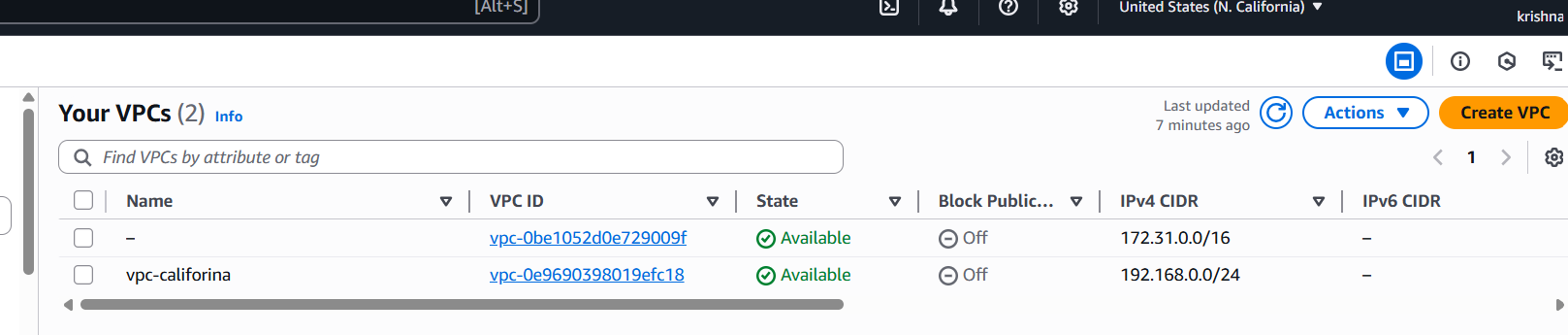
**No public subnets needed (since no IGW/NAT).**

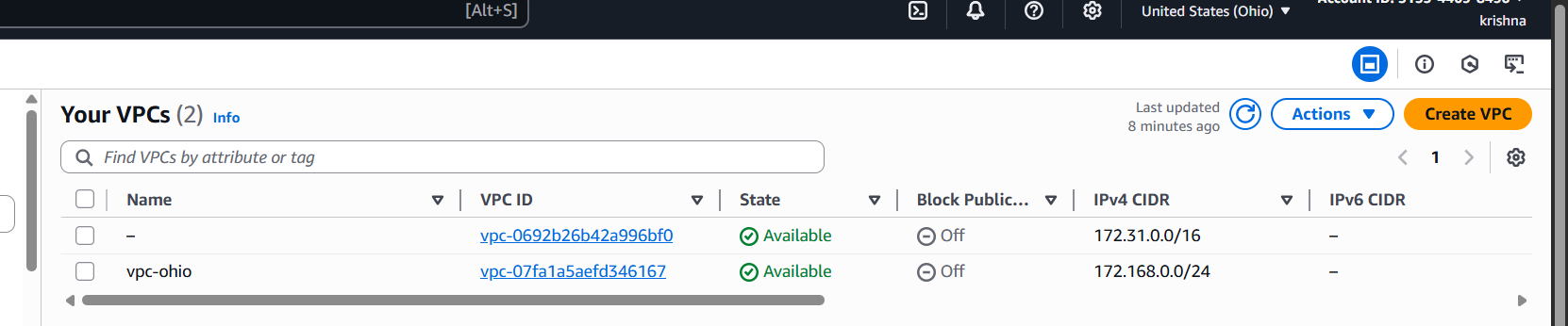
**Attach IGW TO only one VPC to (VPC-A (N. Virginia, us-east-1) → 10.0.0.0/24)**

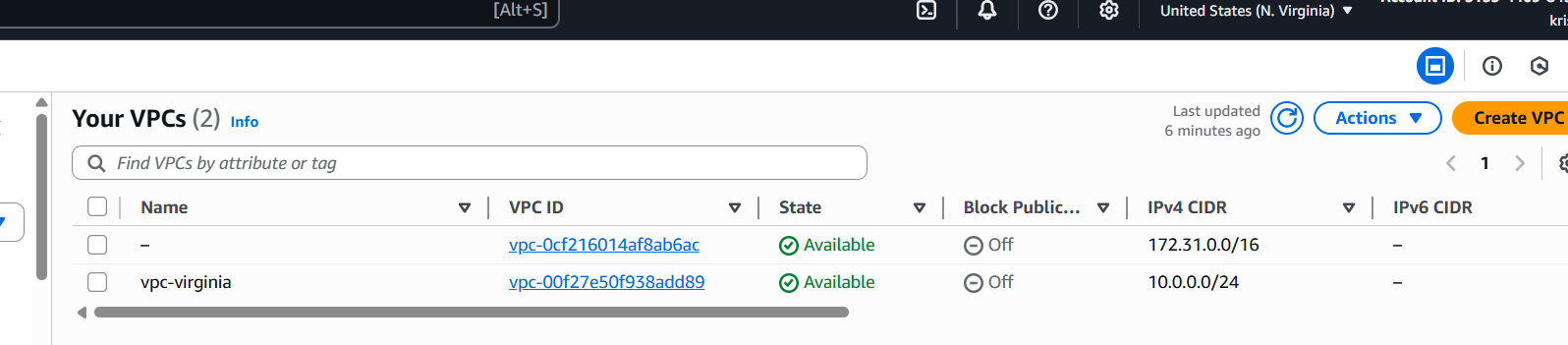
**Create a Route Table for each set of subnets and associate them.**

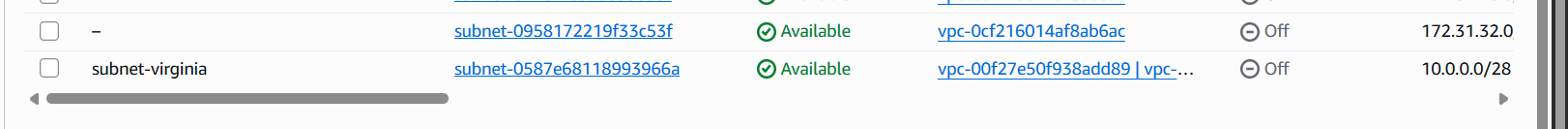
**Repeat for all 4 regions.**

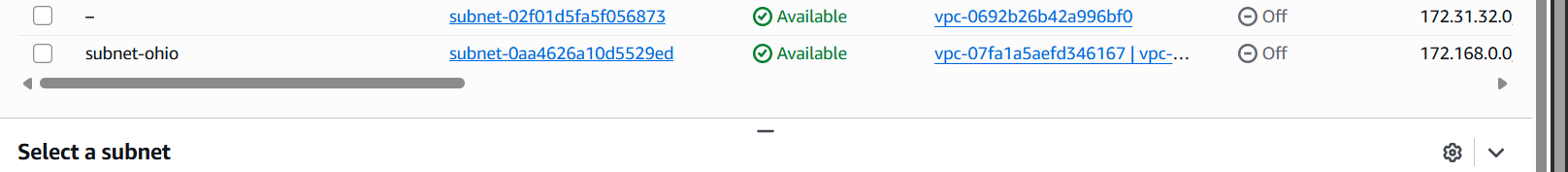


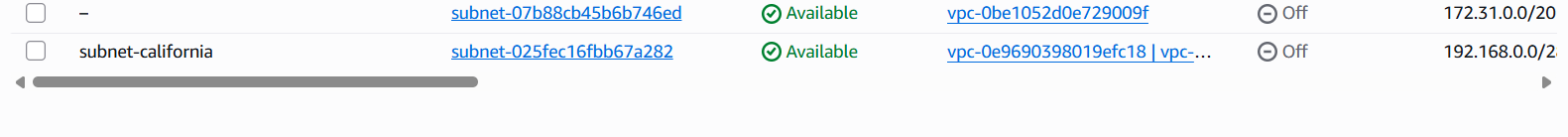


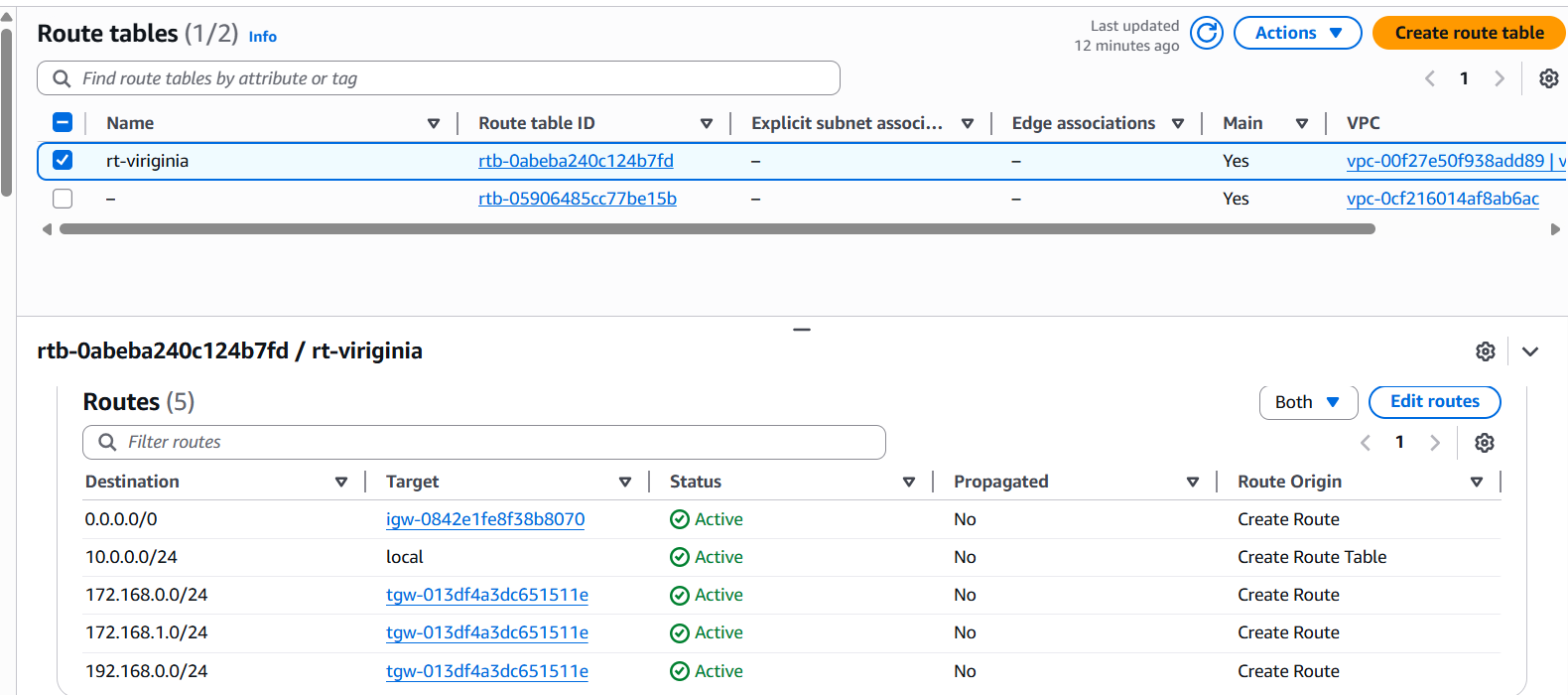


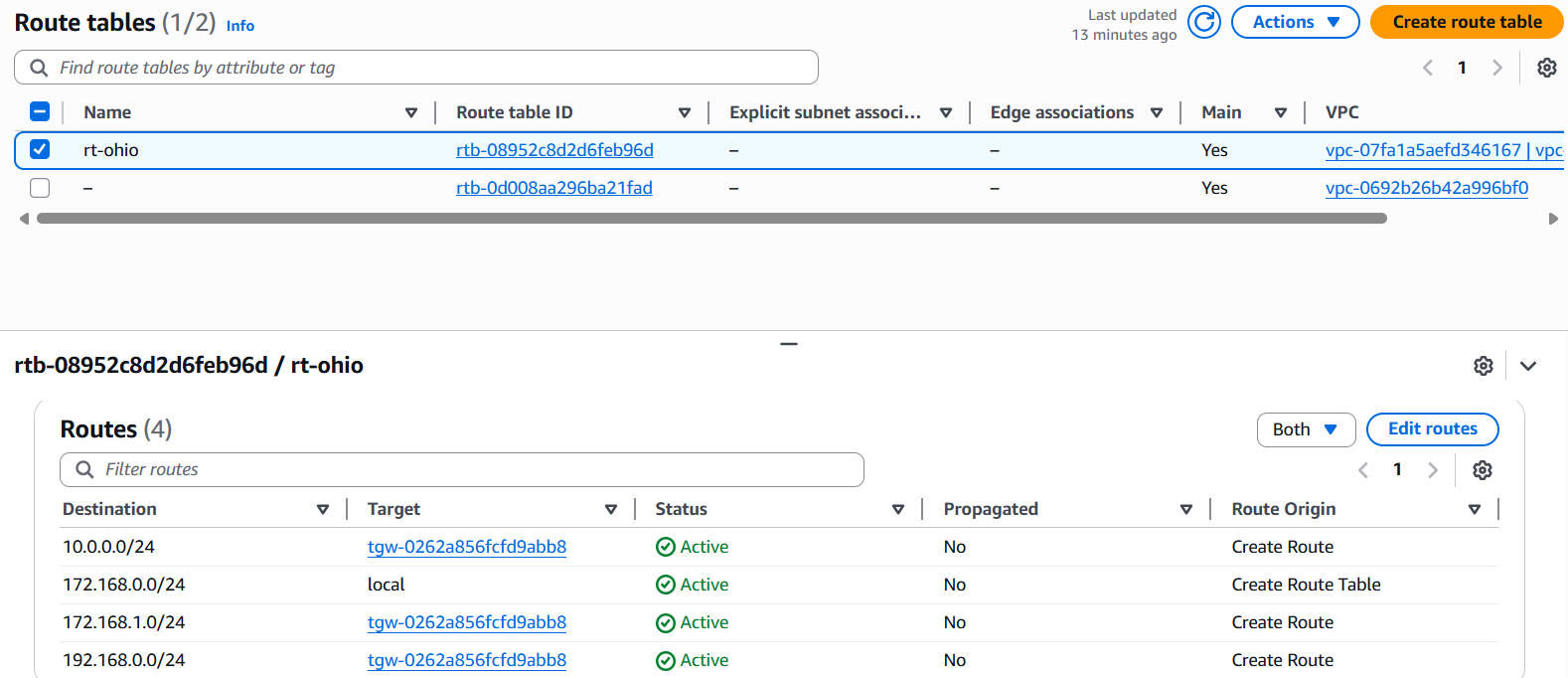


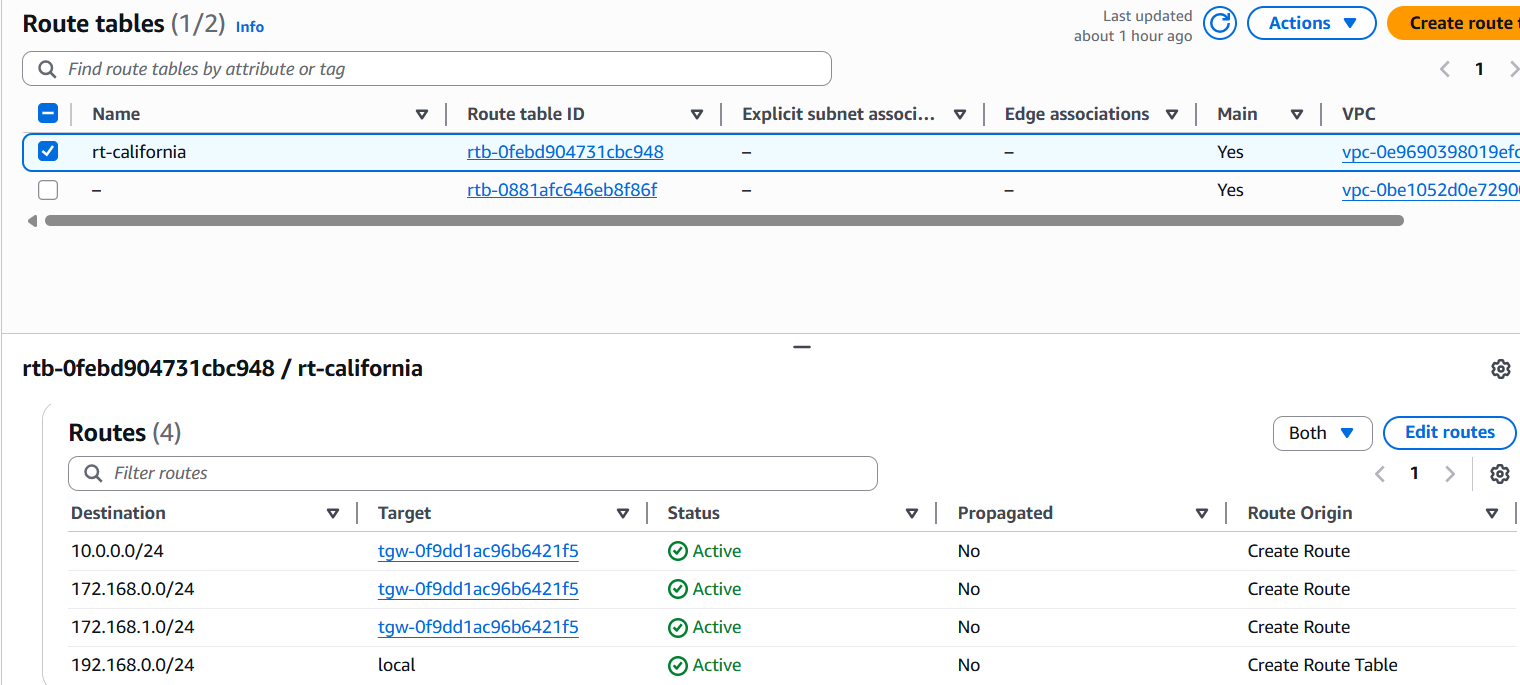


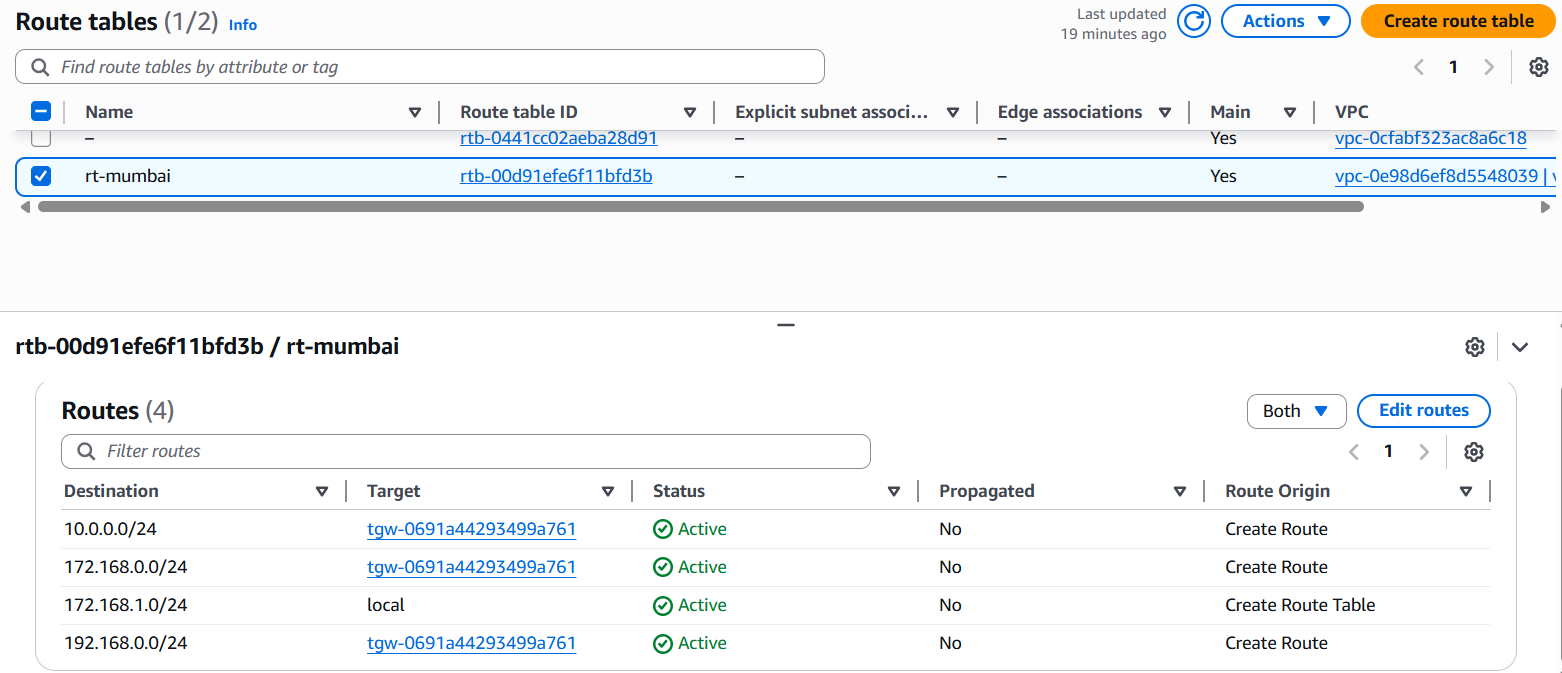


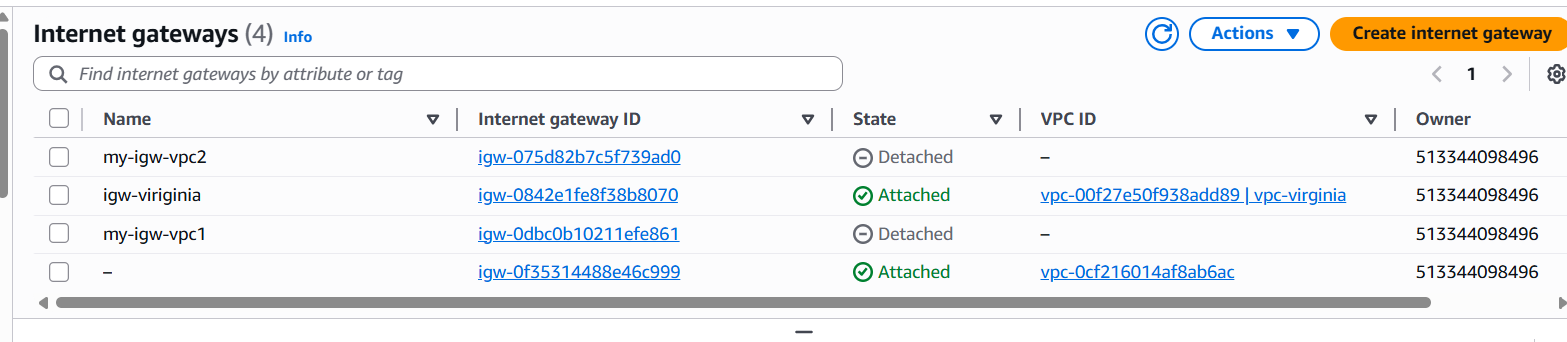












**Step 3: Create Transit Gateways**

**Create Transit Gateways (1 per region)**

Repeat per region.

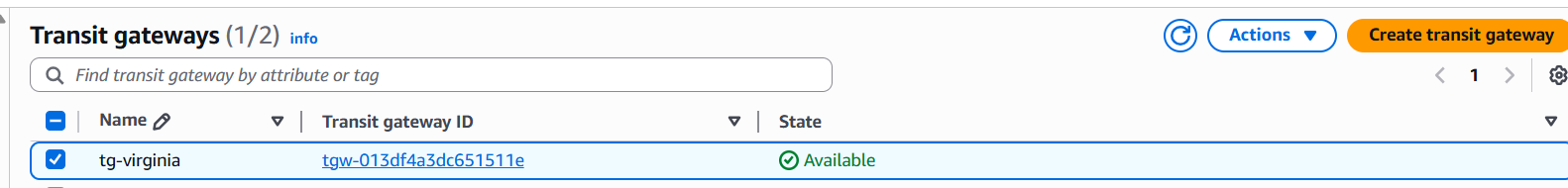
1. **VPC Console → Transit Gateways → Create transit gateway**

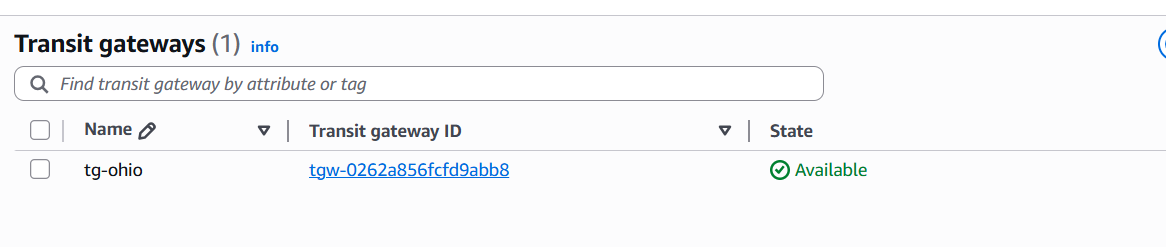
Attach each VPC to its regional TGW

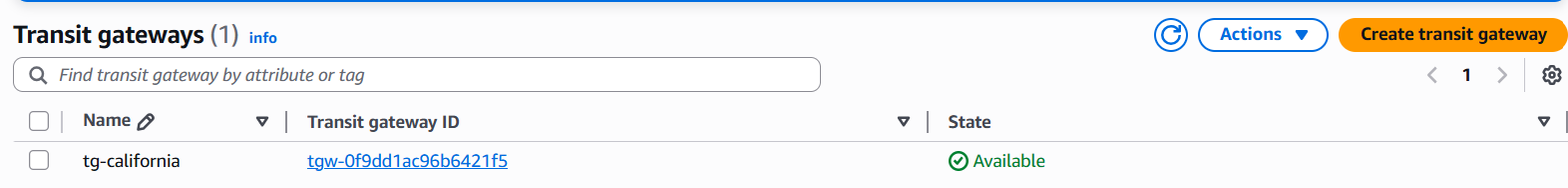
**TGW VPC attachment** with the **private subnets**.

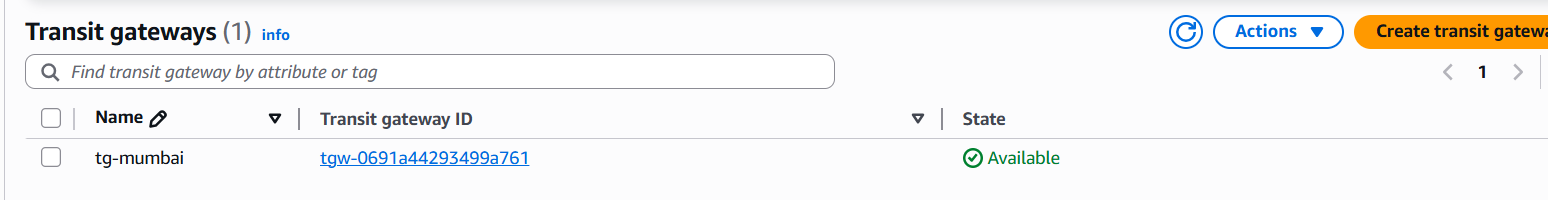
**Associate** the attachment with the TGW route table in that region.

**Enable route propagation** from that attachment into the TGW route table (or add static routes).









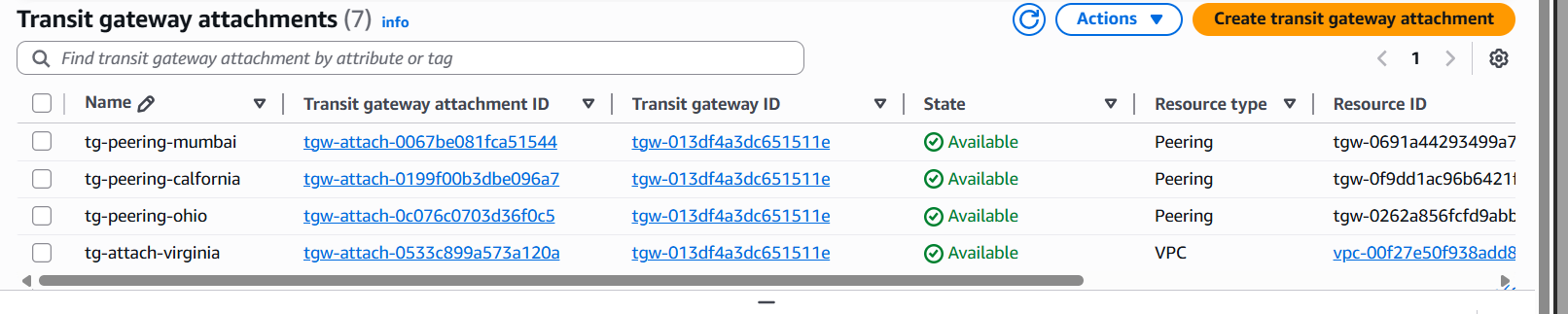
**Step-4 Attach Each VPC to its Regional TGW**

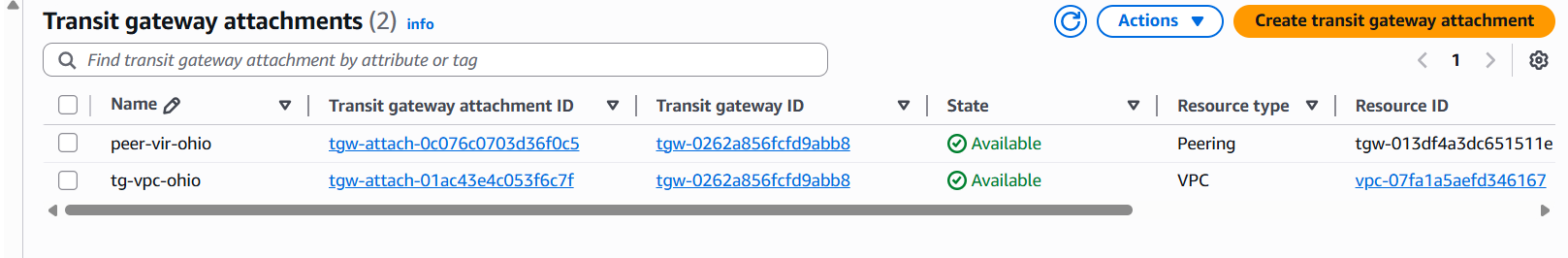
**STEP-5:Establish Inter‑Region TGW Peering (Hub‑and‑Spoke)**

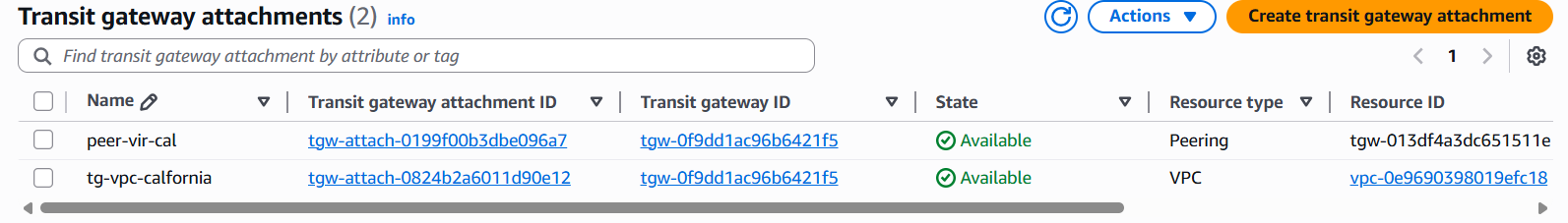
**Create Peering Attachment (Requester Side)**

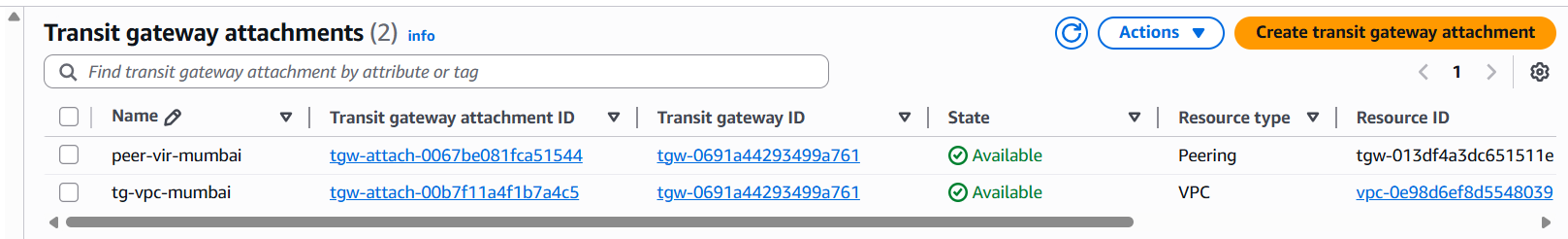
1. **In the AWS Console, go to the VPC Dashboard in Region A (e.g., us-east-1).**
2. **On the left menu, choose Transit Gateway Attachments → Create transit gateway attachment.**
3. **Select:**
   * **Attachment type → *Peering connection*.**
   * **Transit Gateway ID → TGW-US-E1.-Virginia**
   * **Region → target region (e.g., us-west-2 ohio).**
   * **Transit Gateway ID of the peer (TGW-of-Ohio)**

**Click Create transit gateway attachment.**





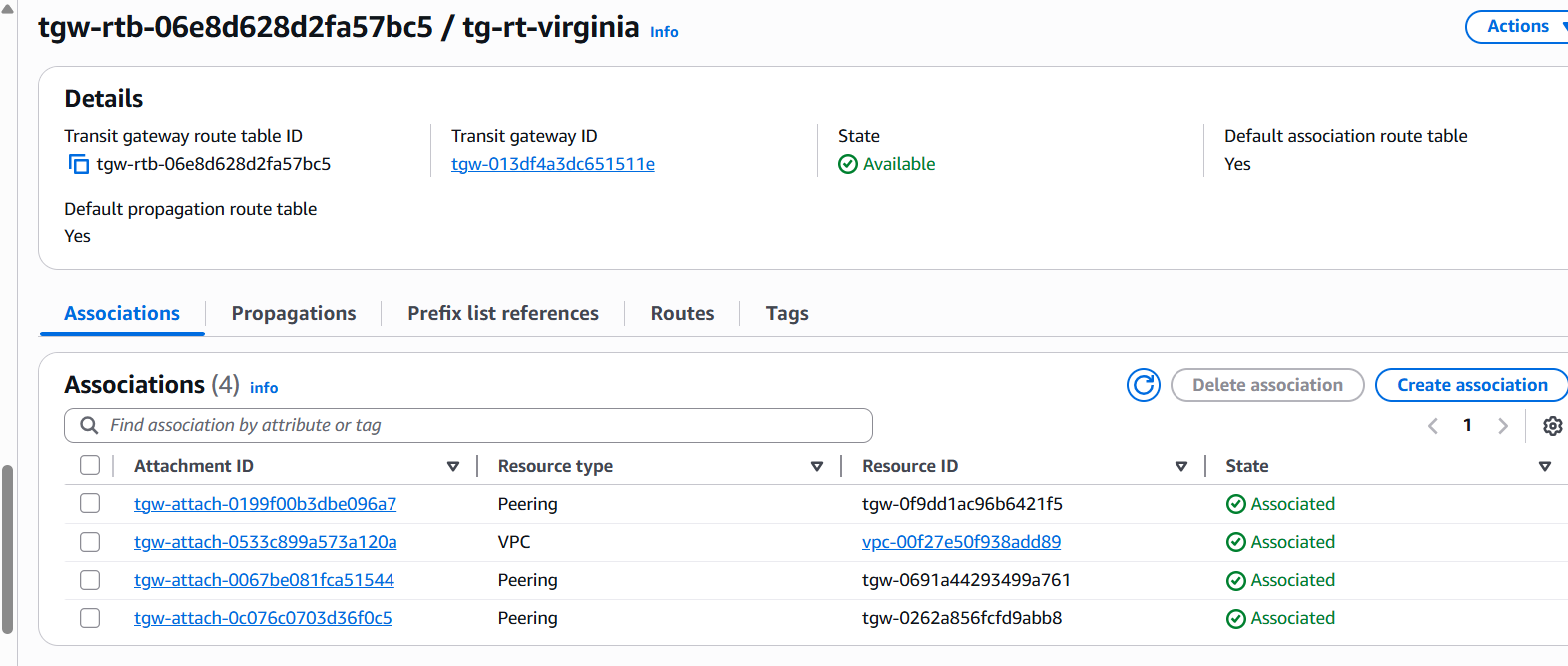


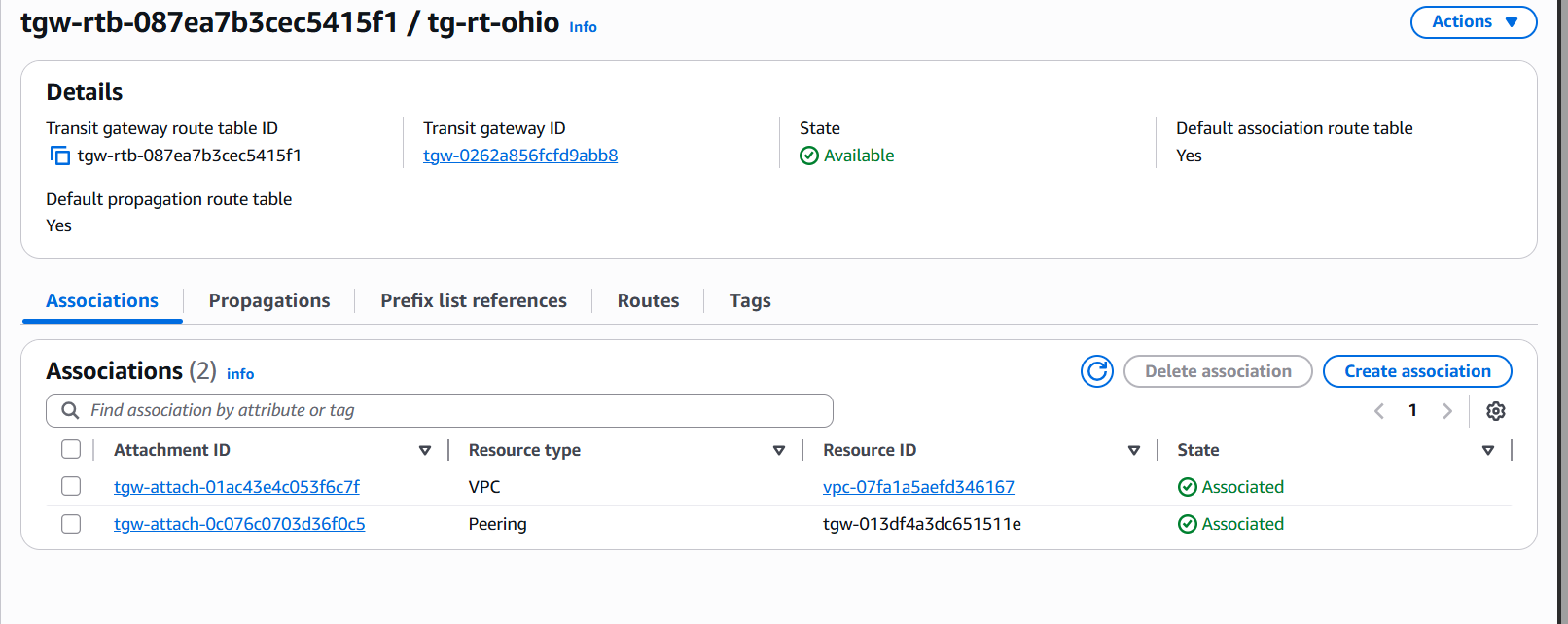


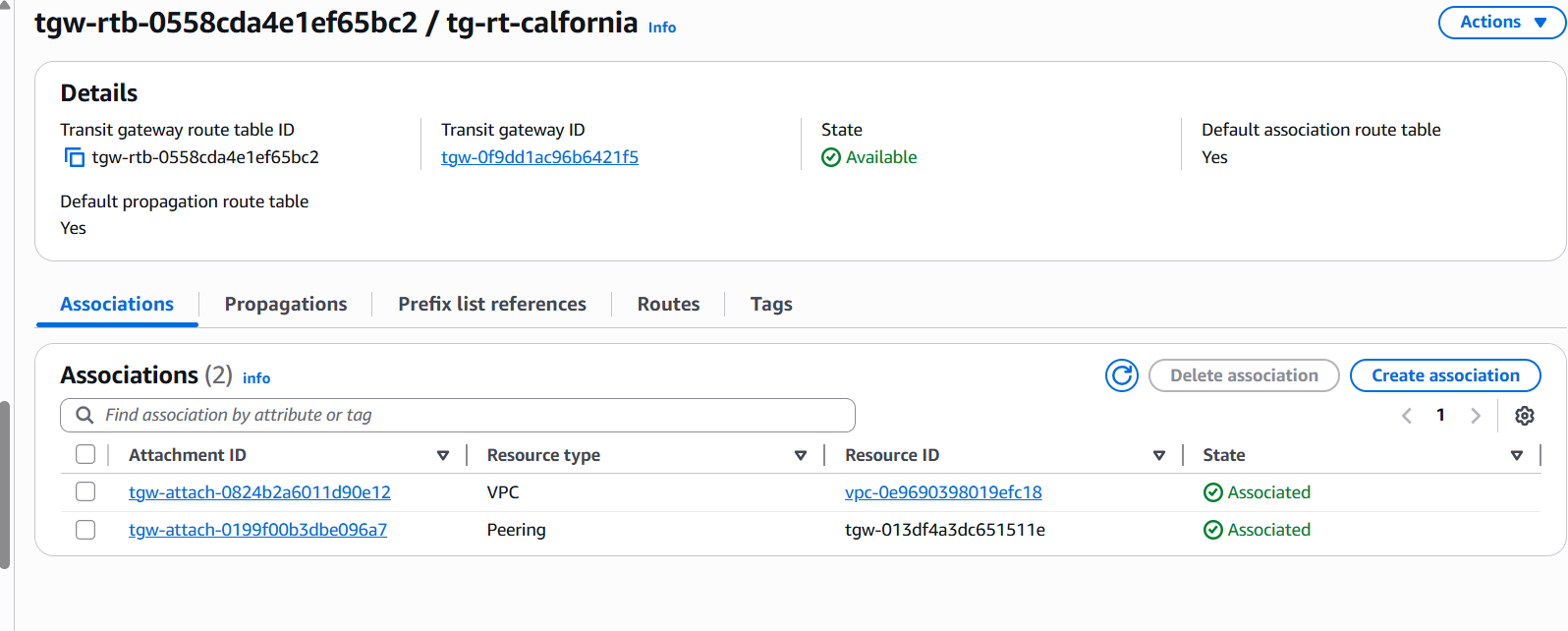
**Step-6:**

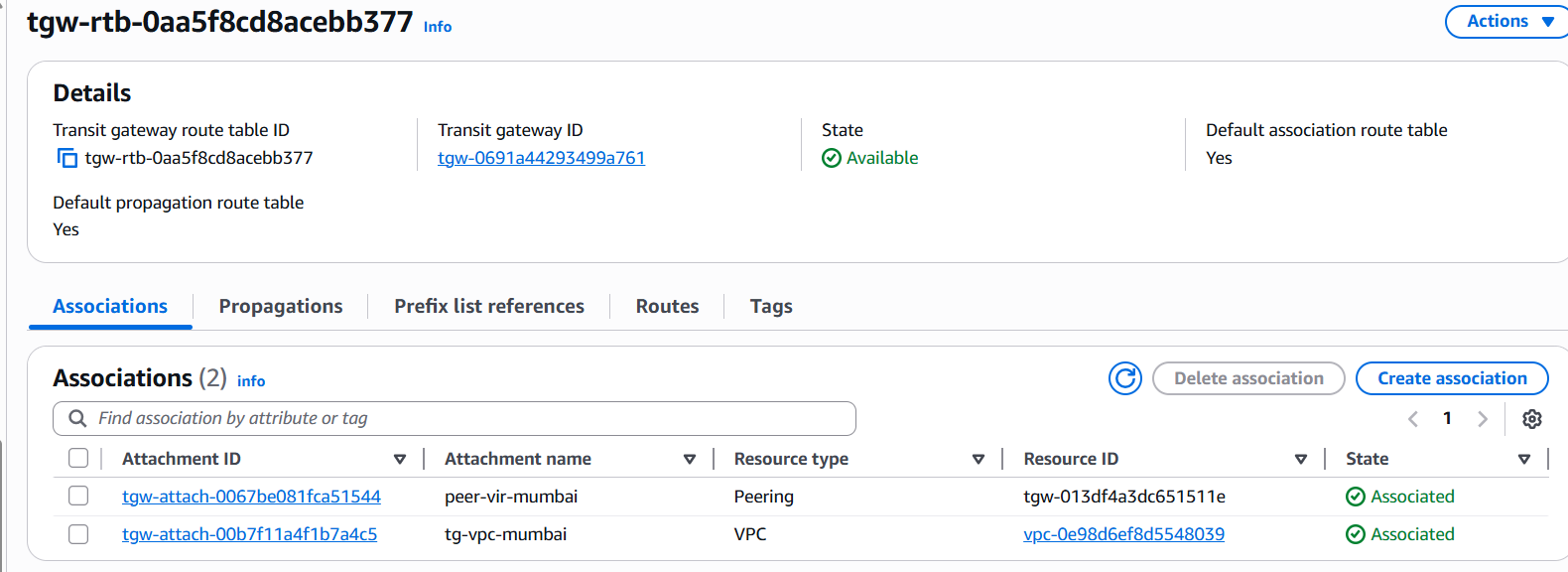
**Configure TGW Route Tables**

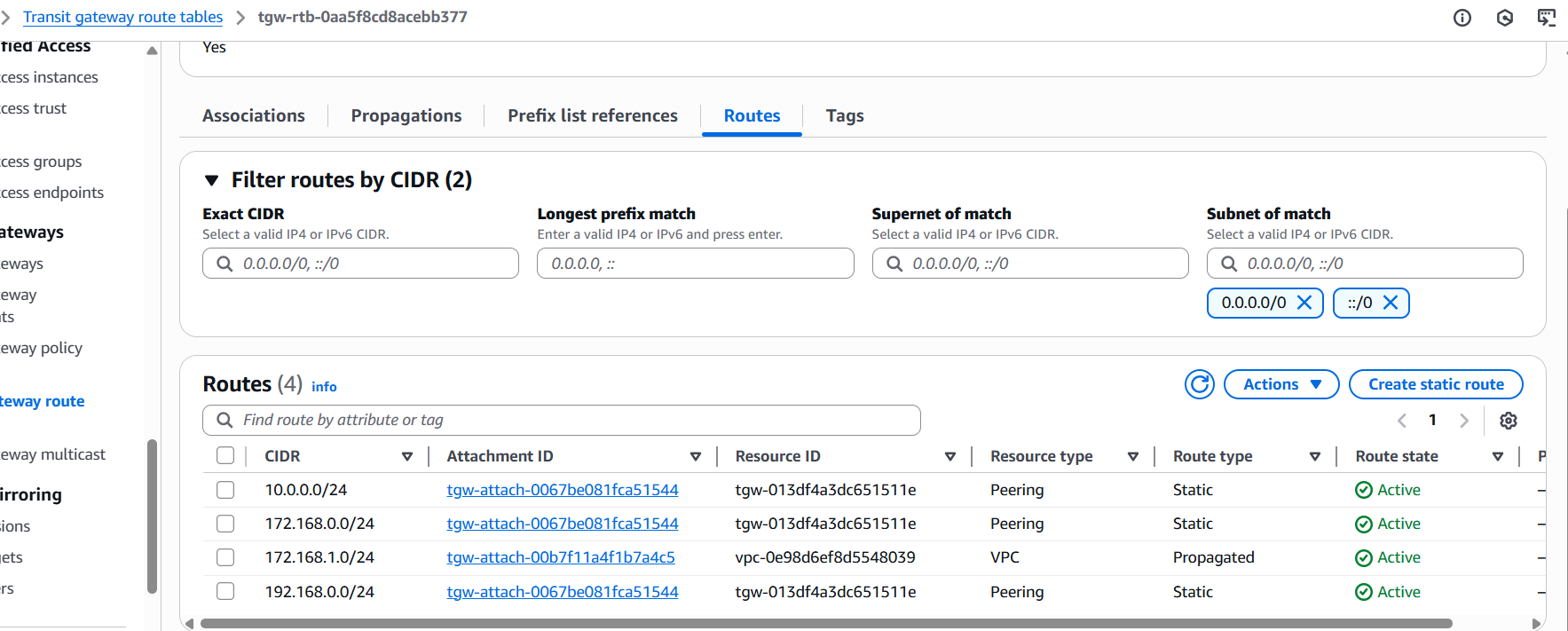
* After the peering is active, you must **update TGW route tables** so traffic can flow.

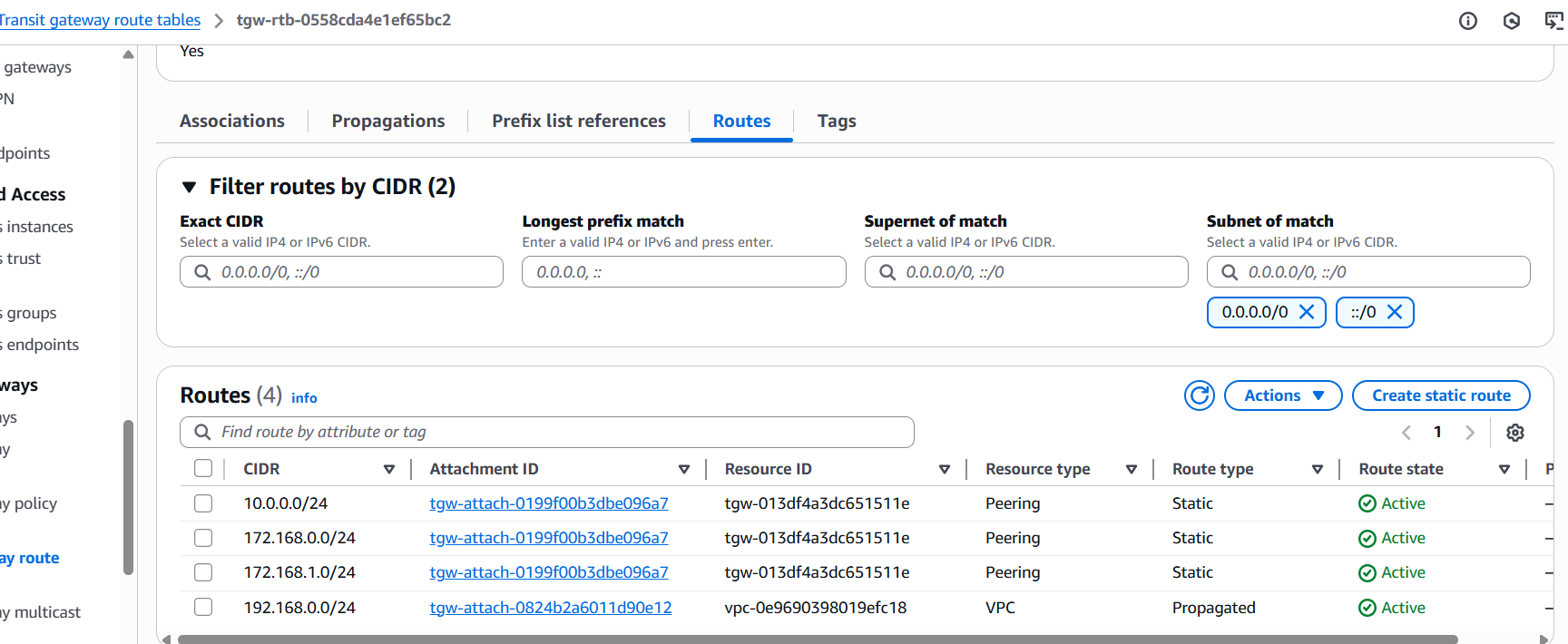


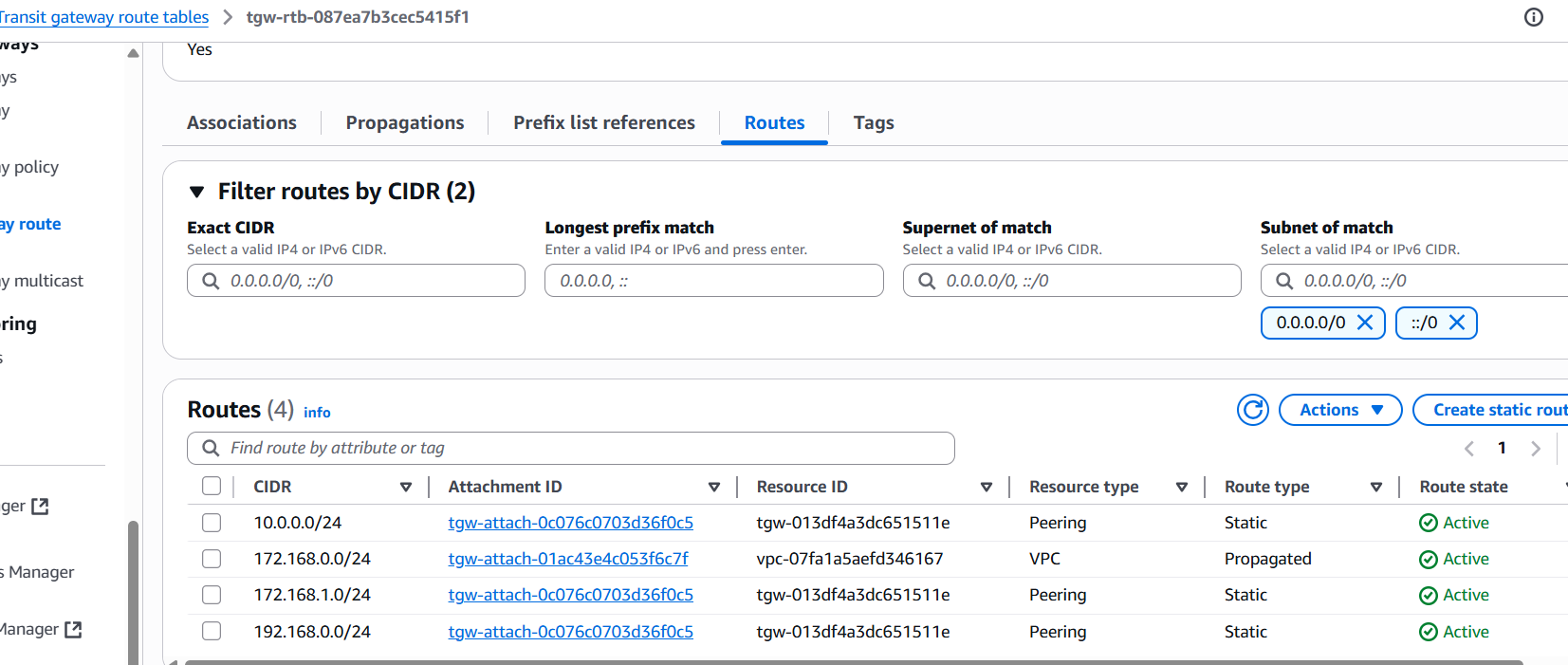


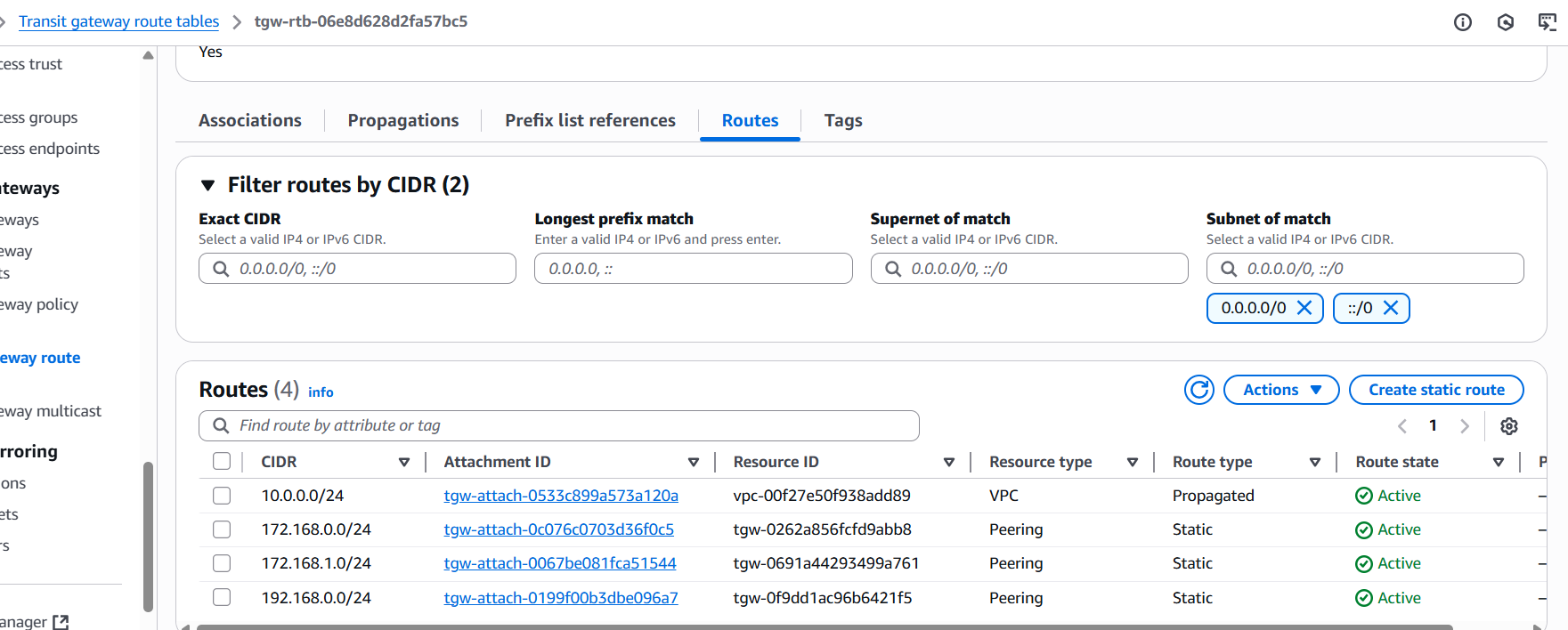












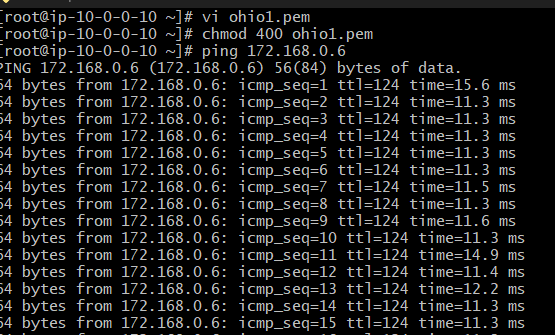
**Step-8 :Verification**

**EC2-to-EC2 connectivity:**

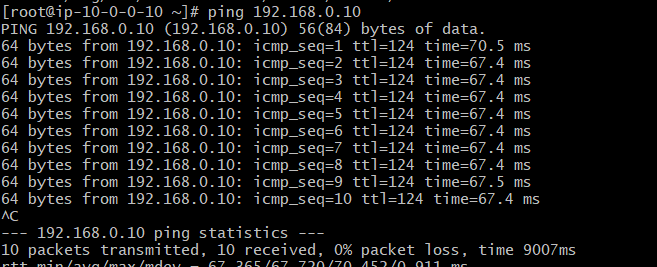
* **Launch 1 EC2 in each VPC in public subnet- - Vpc-1-( N.Virginia)**

**Remaining all Launch 1 EC2 in each VPC private subnet.**

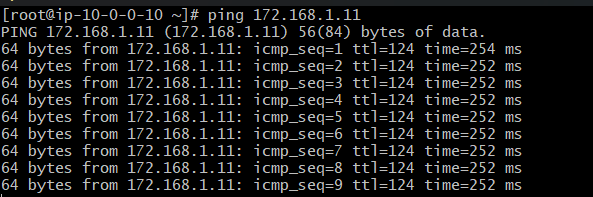
* **From Virginia EC2 → ping Ohio EC2’s private IP.**



**Virginia EC2 → ping California EC2’s private IP.**



**Virginia EC2 → ping Mumbai EC2’s private IP.**



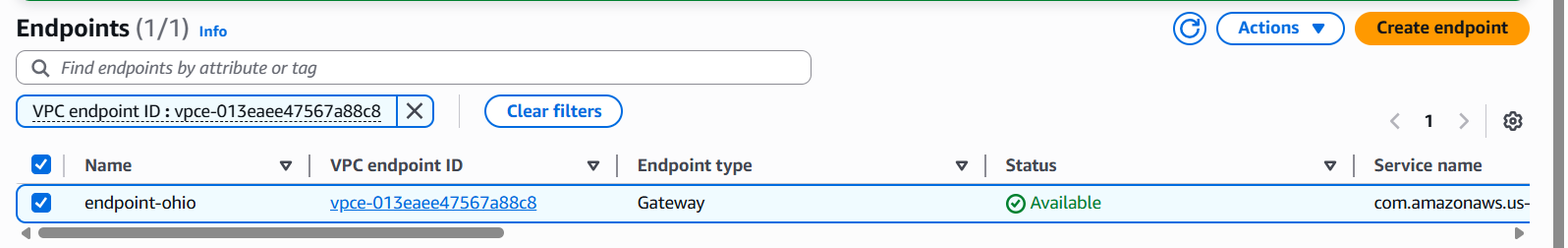
**Configure VPC endpoints to securely access AWS services without internet gateways or NAT gateways, ensuring data privacy and minimizing exposure to external threats.**

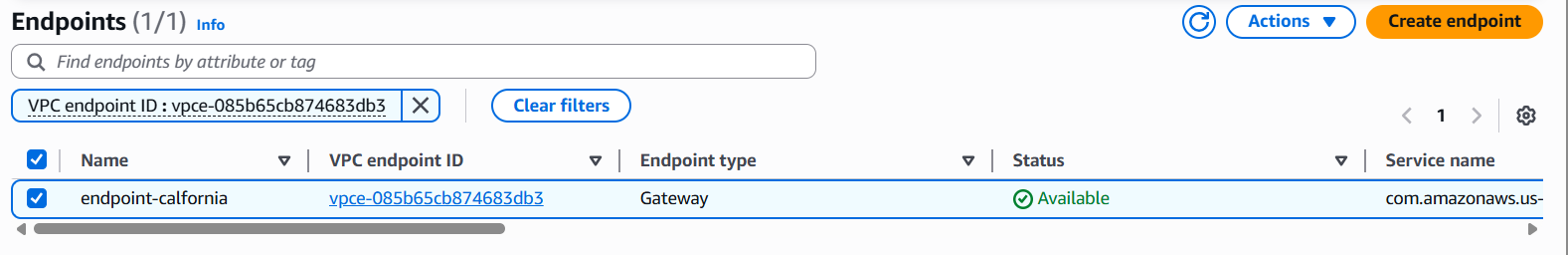
**Configure VPC Endpoints (Private access to AWS services)**

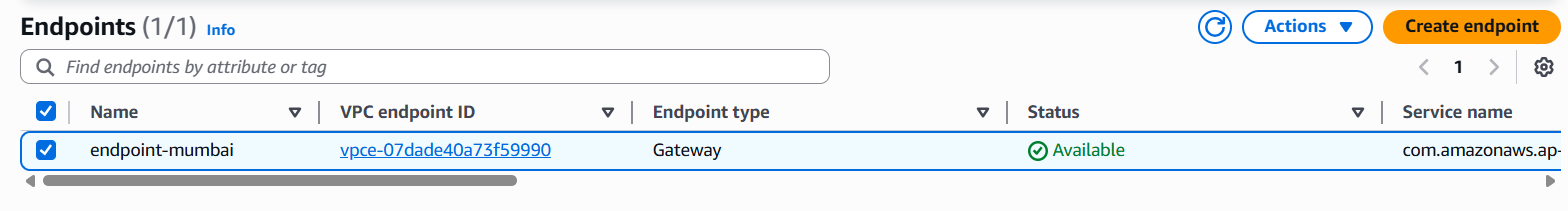
For **each VPC** → Go to **Endpoints → Create Endpoint**.

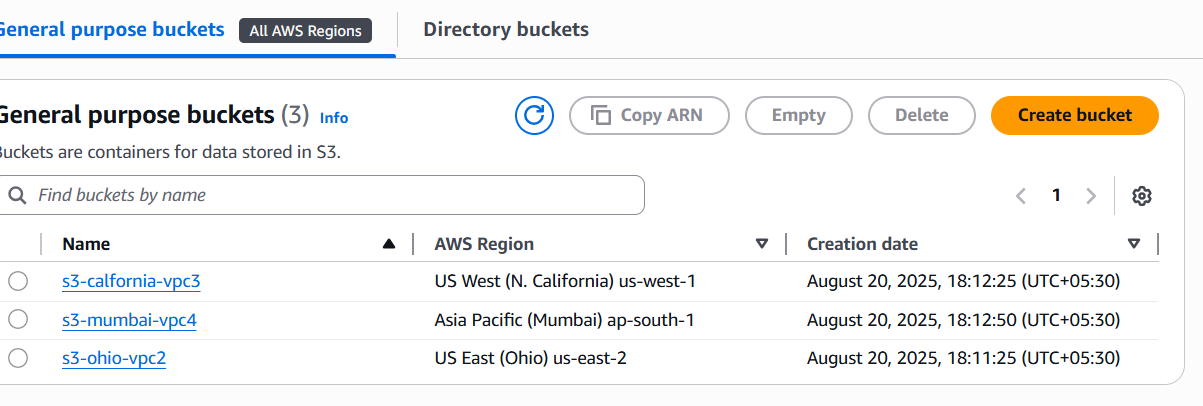
Choose services:

* Gateway Endpoint: **S3**









From N. virgnia Ec2­--🡪ssh to Ohio private ip

Login to Ohio ec2 and configure aws

and check the access to resources like s3

aws s3 ls

