Start working on the below challenge.  
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 a 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.  
Configure VPC endpoints to securely access AWS services   
without internet gateways or NAT gateways, ensuring data  
privacy and minimizing exposure to external threats.  
Note:  
Create 4 Vpc's in 4 different regions and setup transit   
gateway.  
Do not use default vpc's

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

Create 4 vpc in 4 different regions

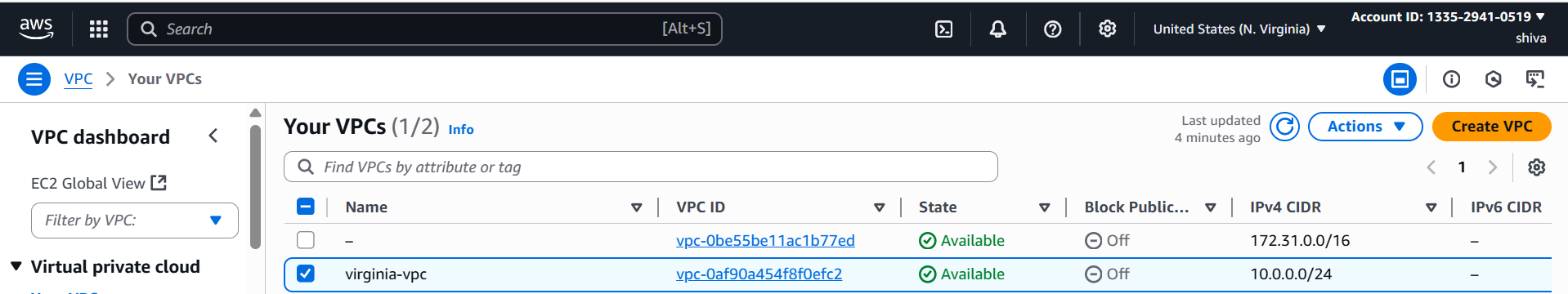
Go to VPC console – create vpc

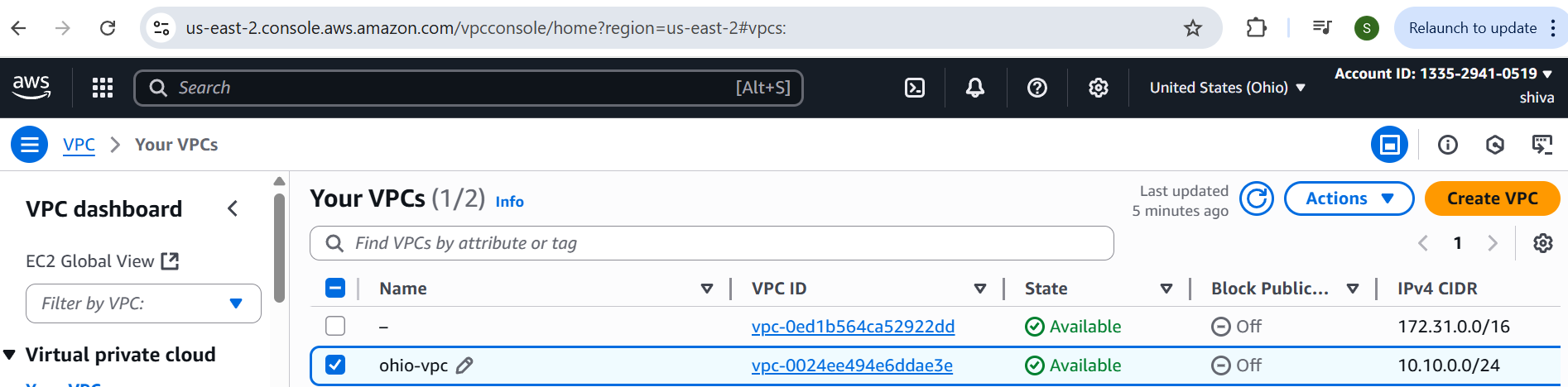
Virginia – 10.0.0.0/24

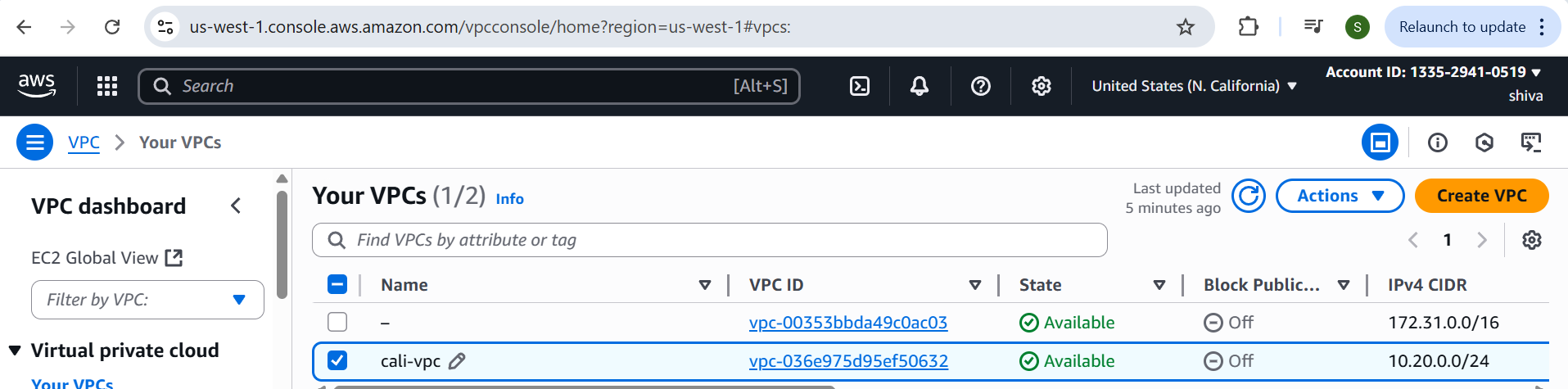
Ohio – 10.10.0.0/24

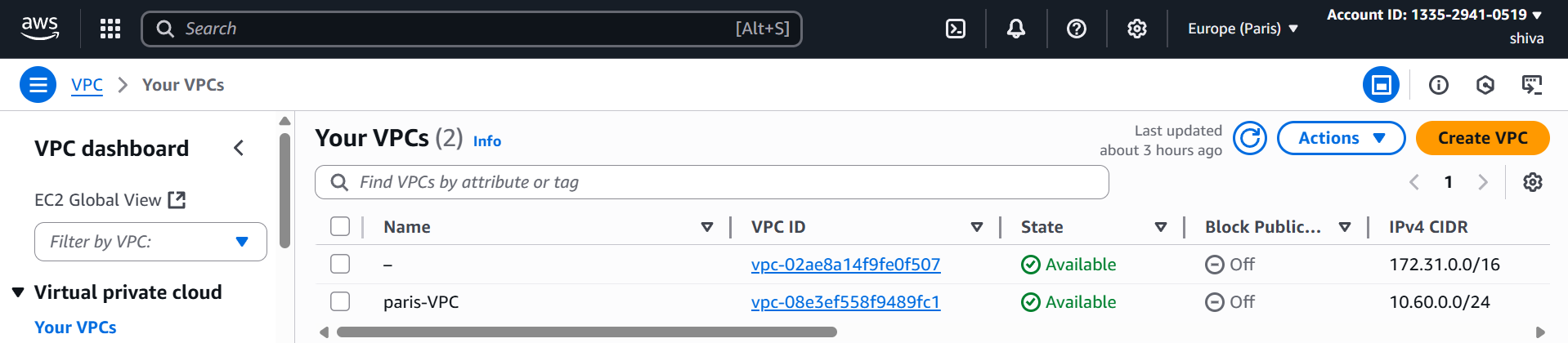
California – 10.20.0.0/24

paris – 10.60.0.0/24









Create subnets in all four regions

Attach IGW in region A Route table(Virginia).

**Create Transit Gateways and attachments(peering and VPC):**

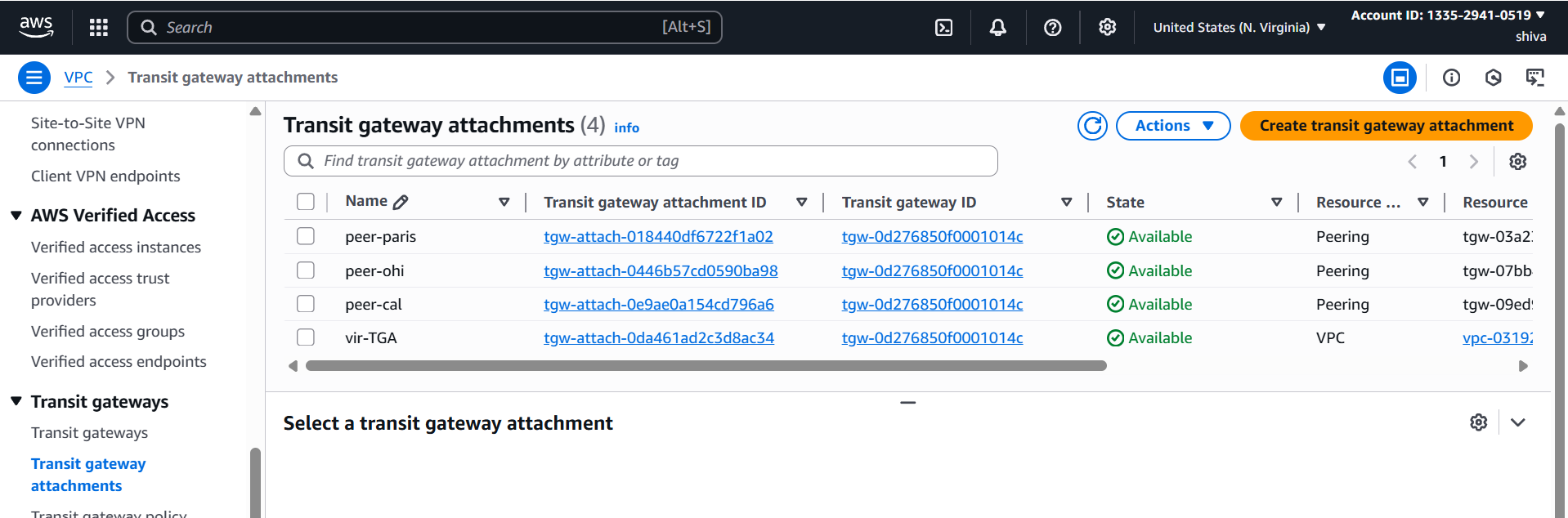
Create transit gateway in all four regions

### Attach VPCs to Their Local Transit Gateways

1. Go to VPC Console → Transit Gateways → Create Transit Gateway
2. Go to transit gateway attachments
3. For VPC
4. Go to Transit Gateway Attachments → Create Attachment
5. Choose VPC
6. Select vpc and create
7. For peering
8. Go to Transit Gateway Attachments → Create Attachment
9. Choose "Peering Attachment"
10. Select your account and Region B's Transit Gateway (transit gateway id)
11. Create

After that need to accept peering connection from that region

Repeat for other regions

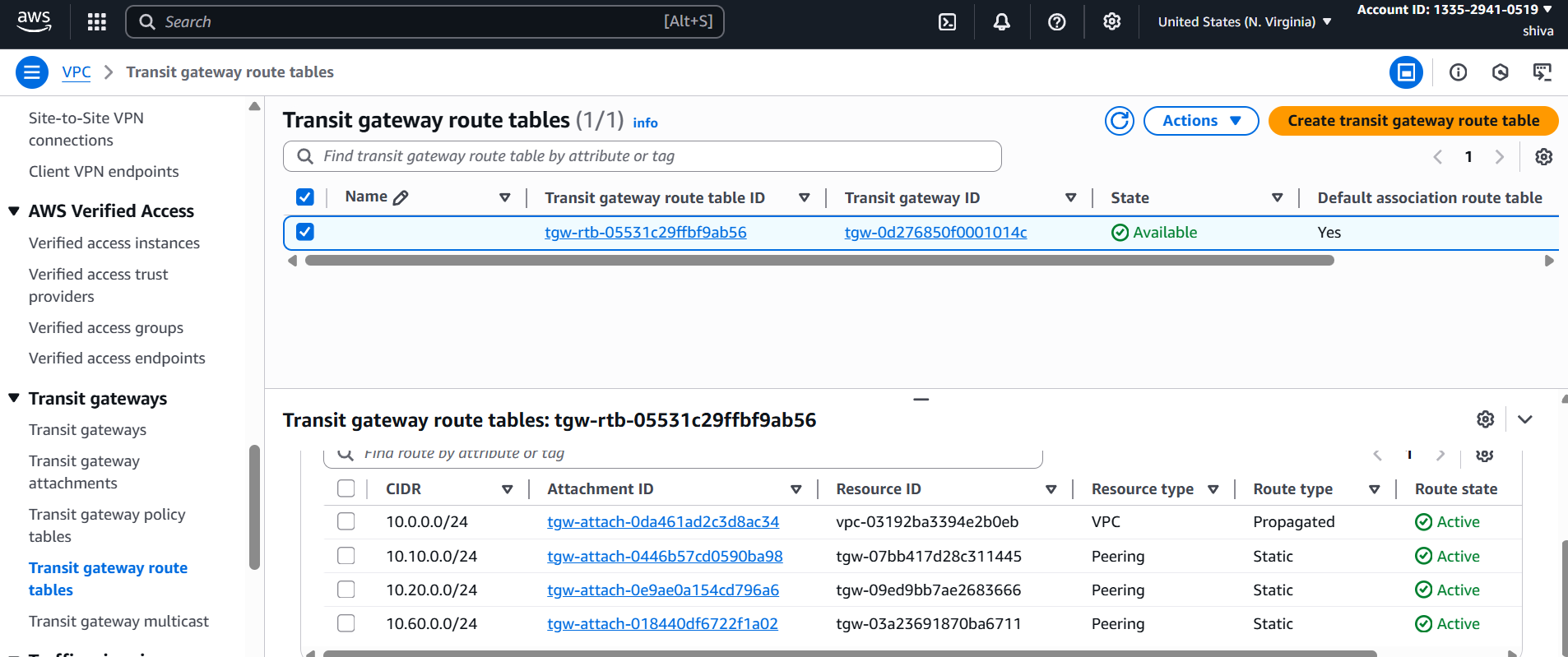


### Configure Routes (Most Important Step!):

### In all the regions do these two things

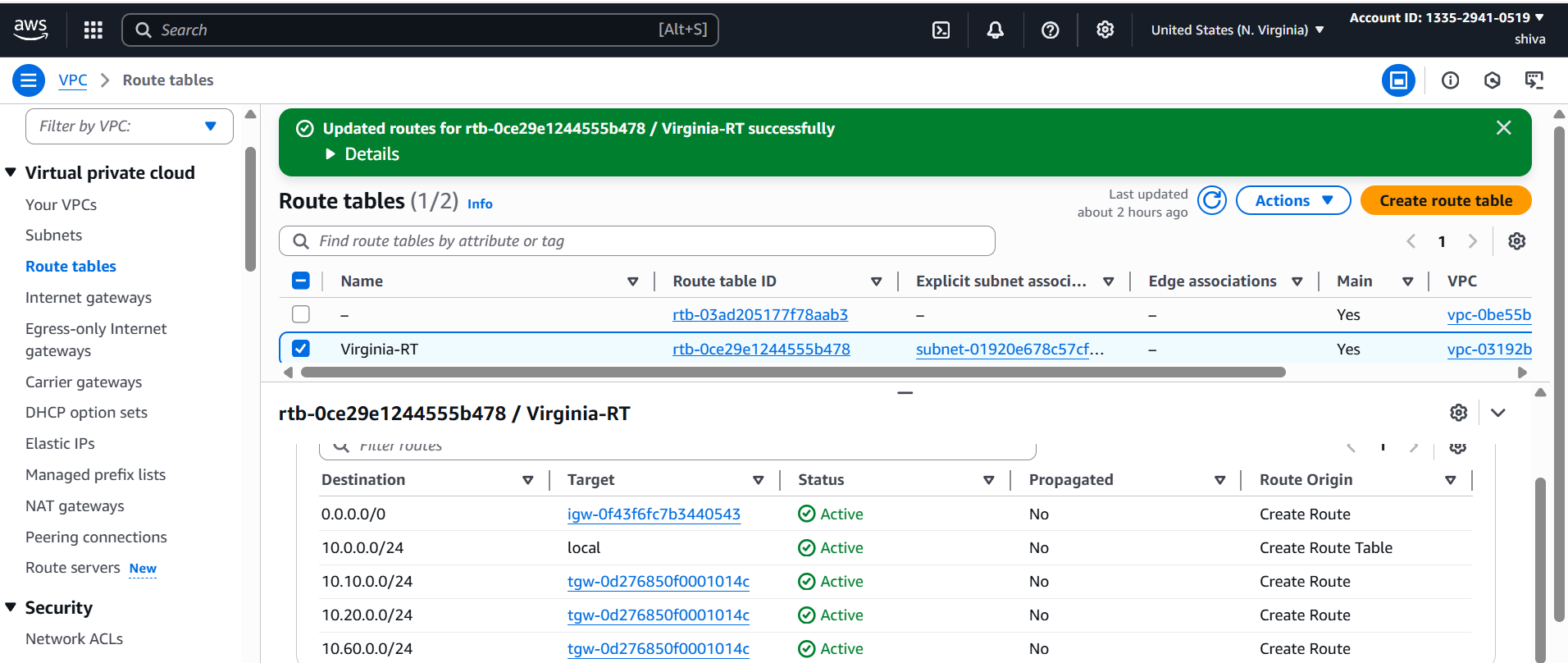
1. **Transit Gateway Route Table**:
   * Add route pointing to the other VPC's CIDR via the peering attachment  
     (e.g., in Region A: 10.10.0.0/24→ Peering Attachment to Region B (repeat for other regions also))

Go to transit gateway route tables -> routes



1. **VPC Route Table**:
   * Add route pointing to the other VPC's CIDR via the Transit Gateway  
     (e.g., in VPC A: 10.10.0.0/24 → TGW attachment) (repeat for other regions also)

Route tables -> routes ->edit routes ->add all other CIDRs with Transit Gateway



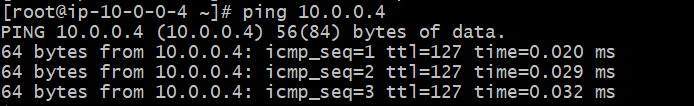
**Testing:**

Make sure security groups allow traffic between them

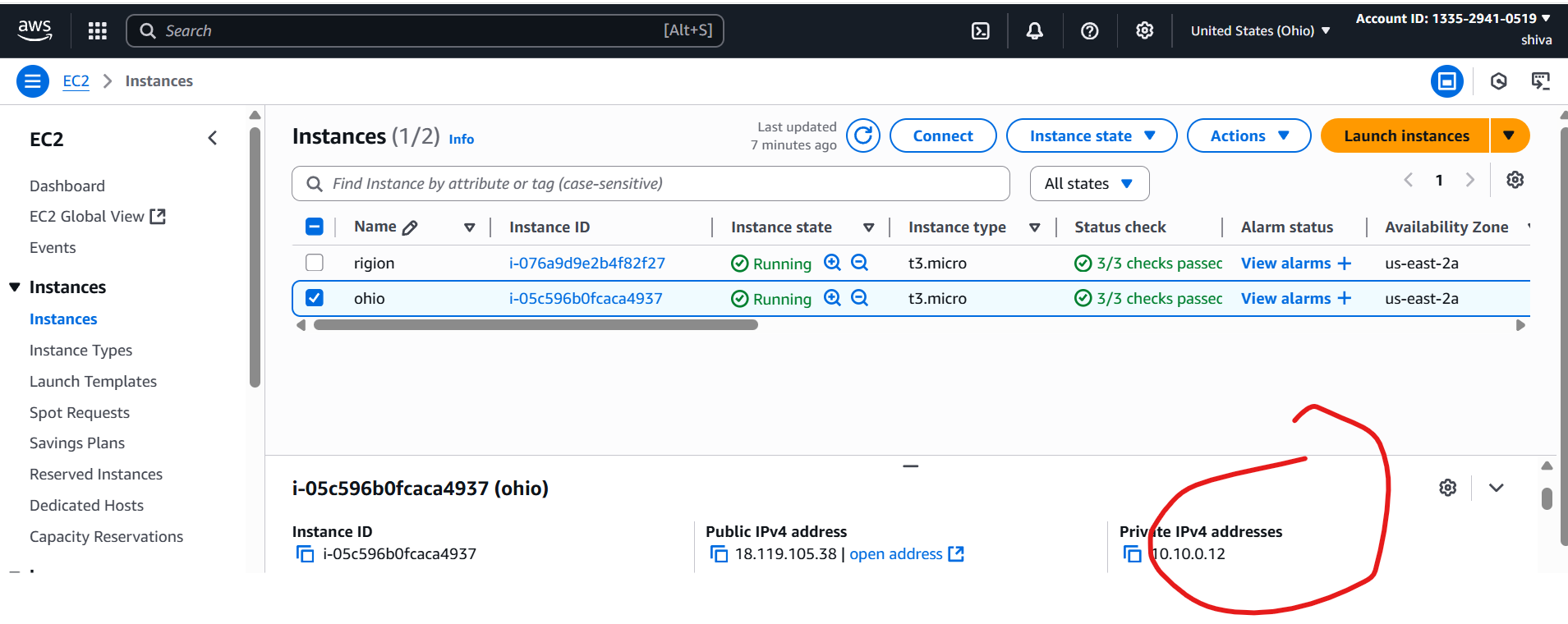
connect to region A (ex: Virginia ip)

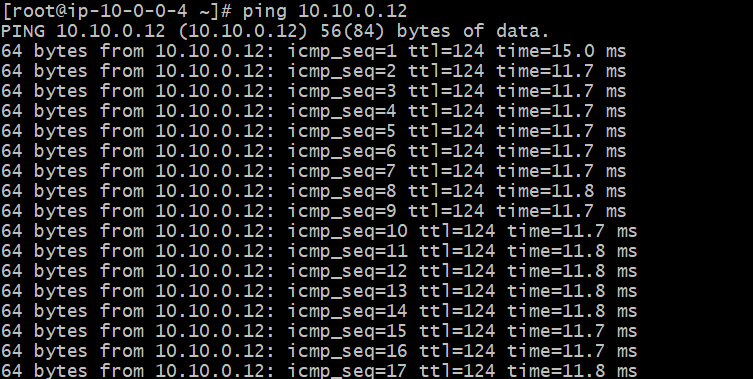
Check connection of other regions ip’s from Virginia region

Virginia connection:

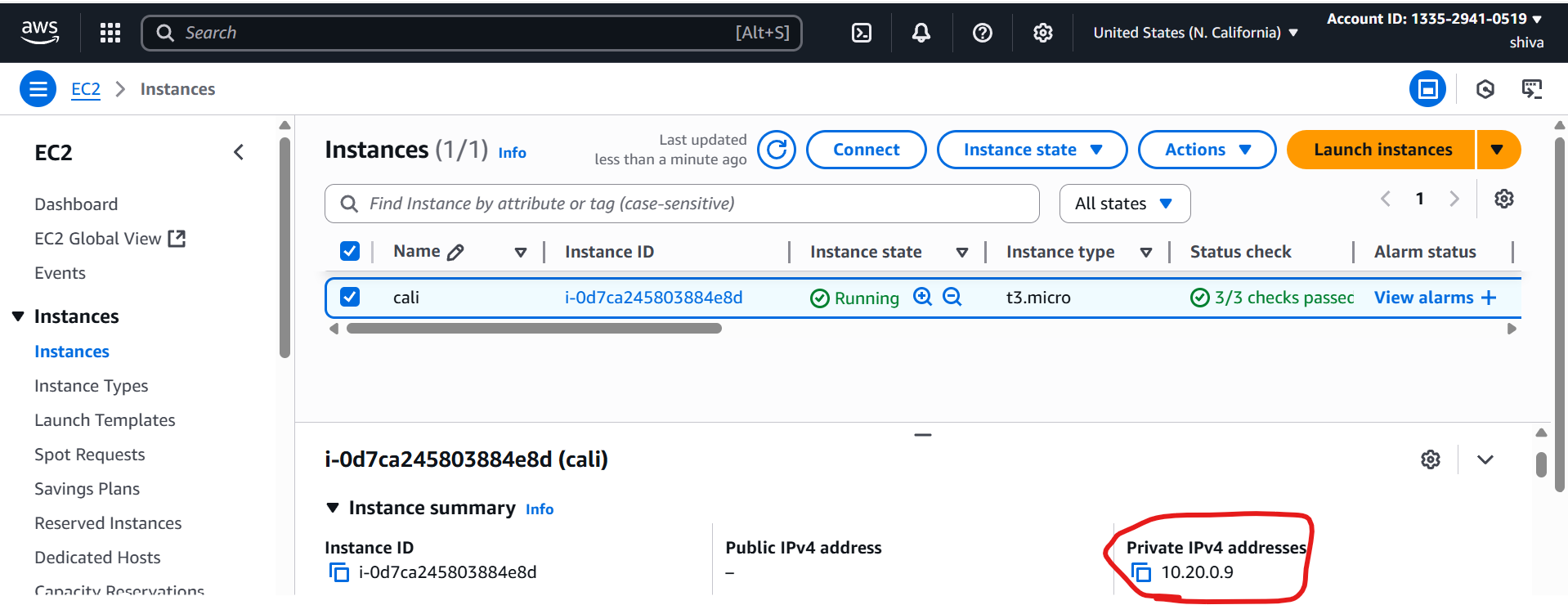


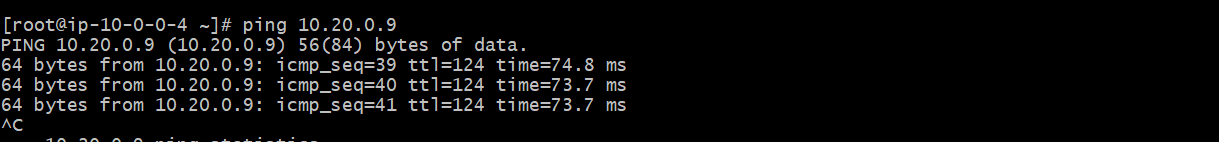
Ohio connection:



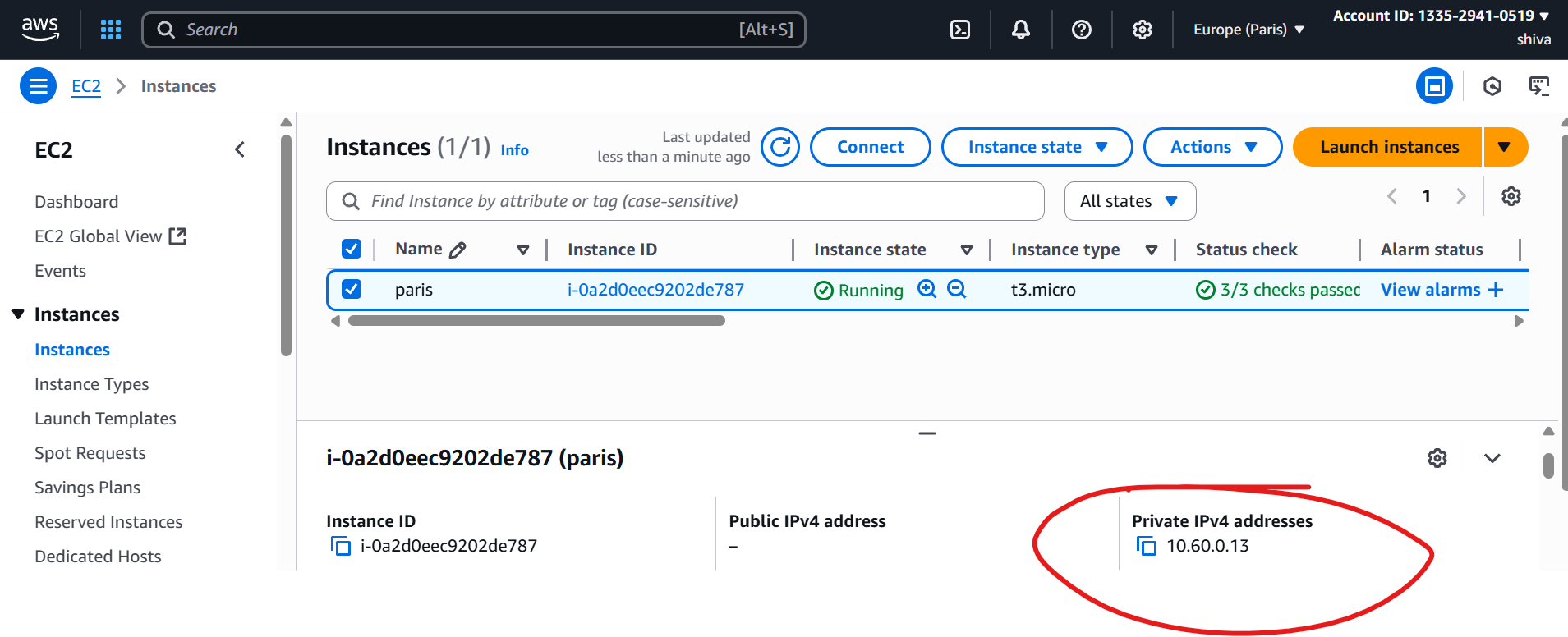


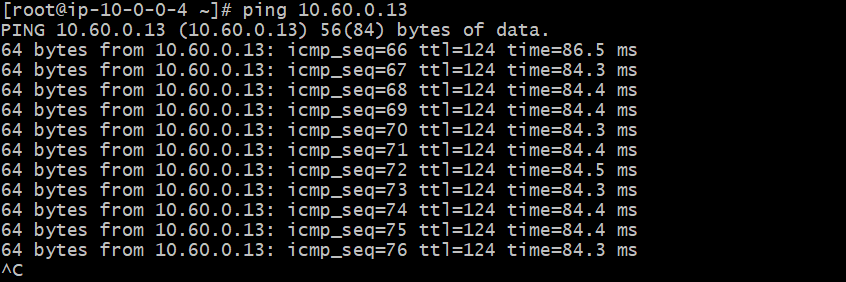
California connection:





Paris connection:





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

For each VPC → Go to Endpoints → Create   
Endpoint.  
 Choose services:  
•Gateway Endpoint: S3  
Create an S3 endpoint in Ohio and California, paris

From N.virgina Ec2--ssh to Ohio private ip

Configure aws

