**Task on VPC-02**

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

Create a VPC

- Sign in to the AWS Management Console.

- Navigate to VPC service.

- Click Create VPC.

- Enter:

- Name tag (e.g., MyVPC).

- IPv4 CIDR block (e.g., 192.168.0.0/24).

- IPv6 CIDR block (optional).

- Tenancy (default).

- Click Create VPC.

Create Public and Private Subnets

- Go to Subnets in the VPC dashboard.

- Click Create Subnet.

- Select your VPC.

- Enter:

- Subnet name (e.g., PublicSubnet).

- Availability Zone (choose one).

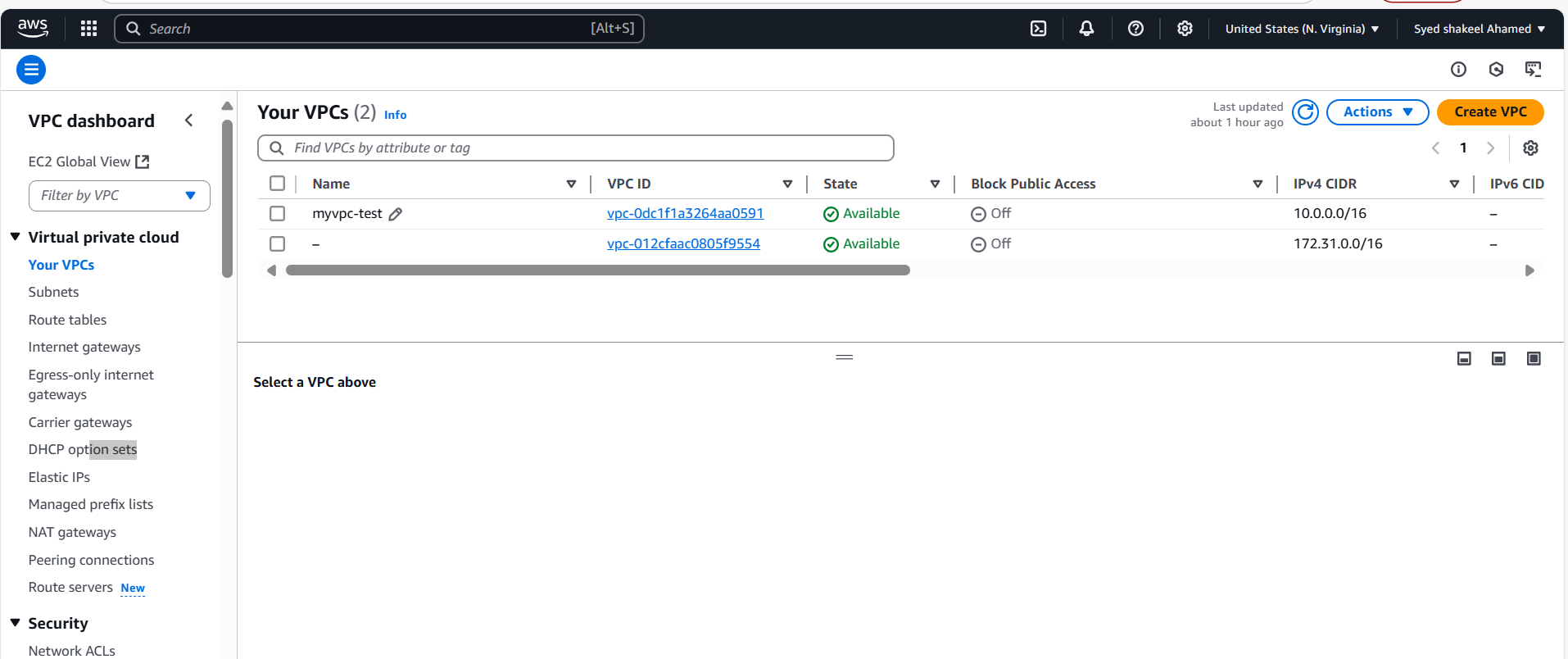
- IPv4 CIDR block (e.g., 192.168.0.0/26 for public subnet).

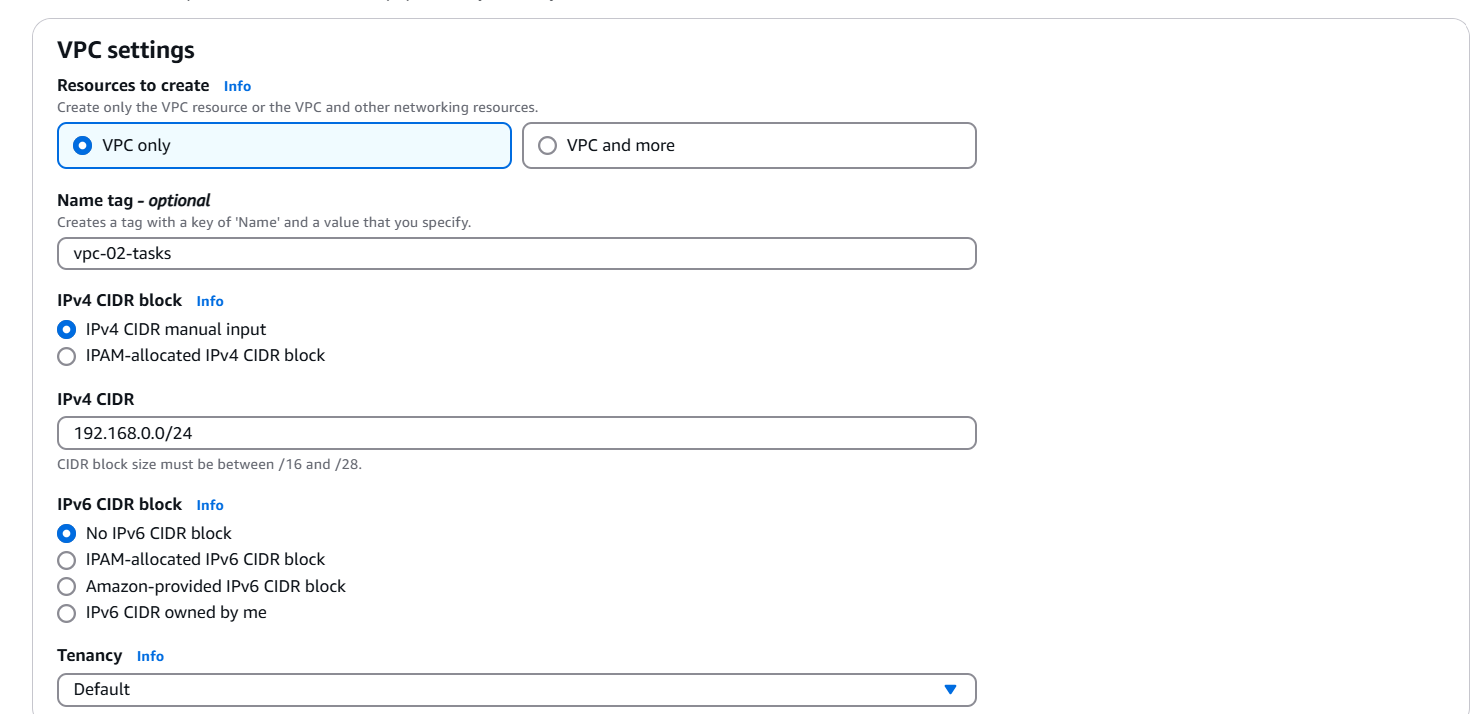
- Click Create Subnet.

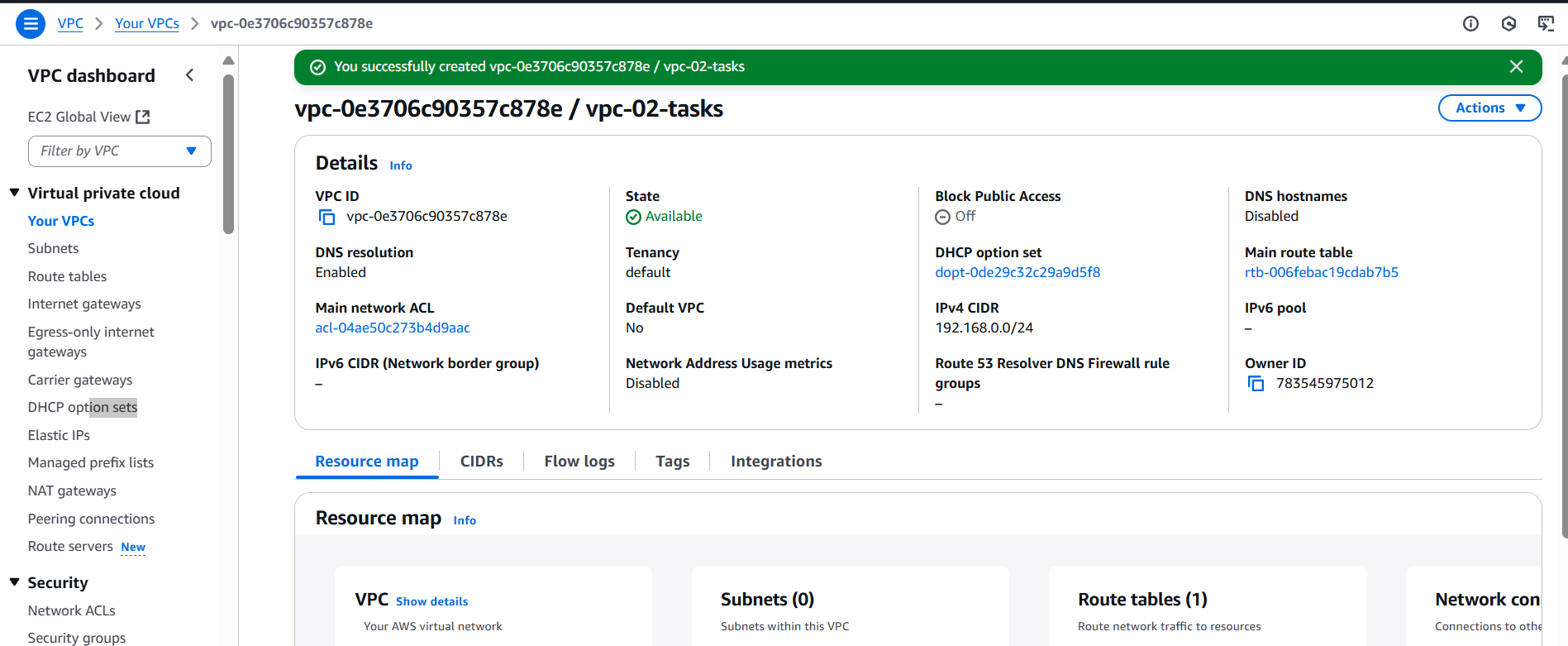
- Repeat the process for the private subnet:

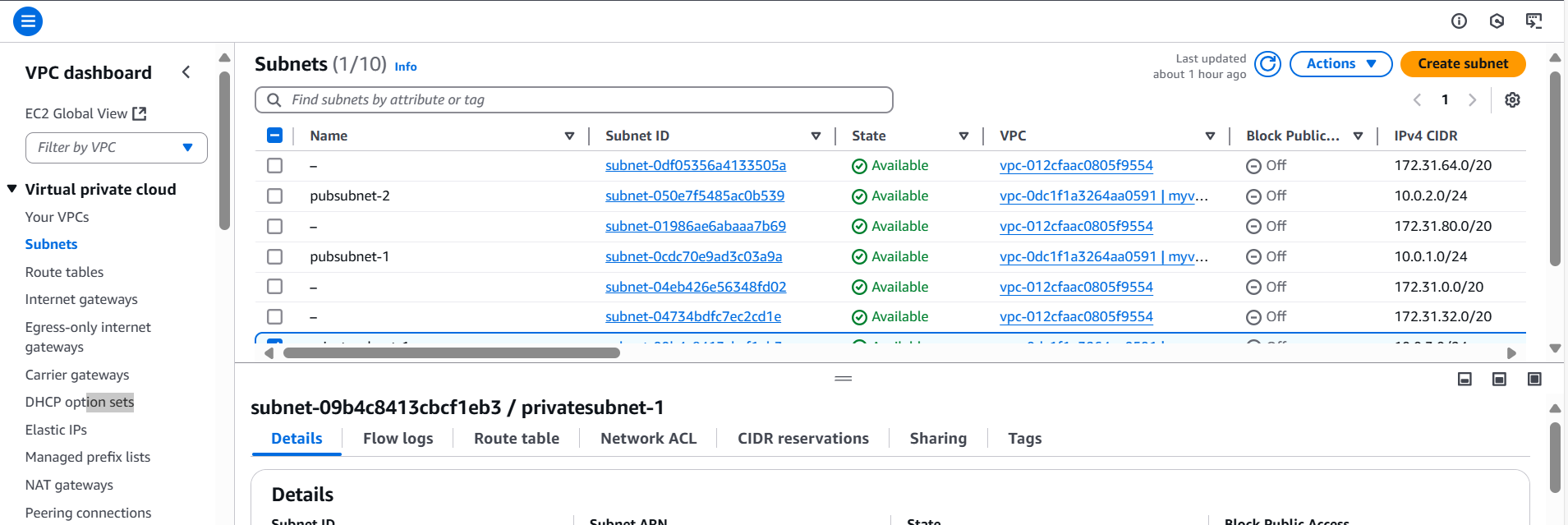
- Subnet name (e.g., PrivateSubnet).

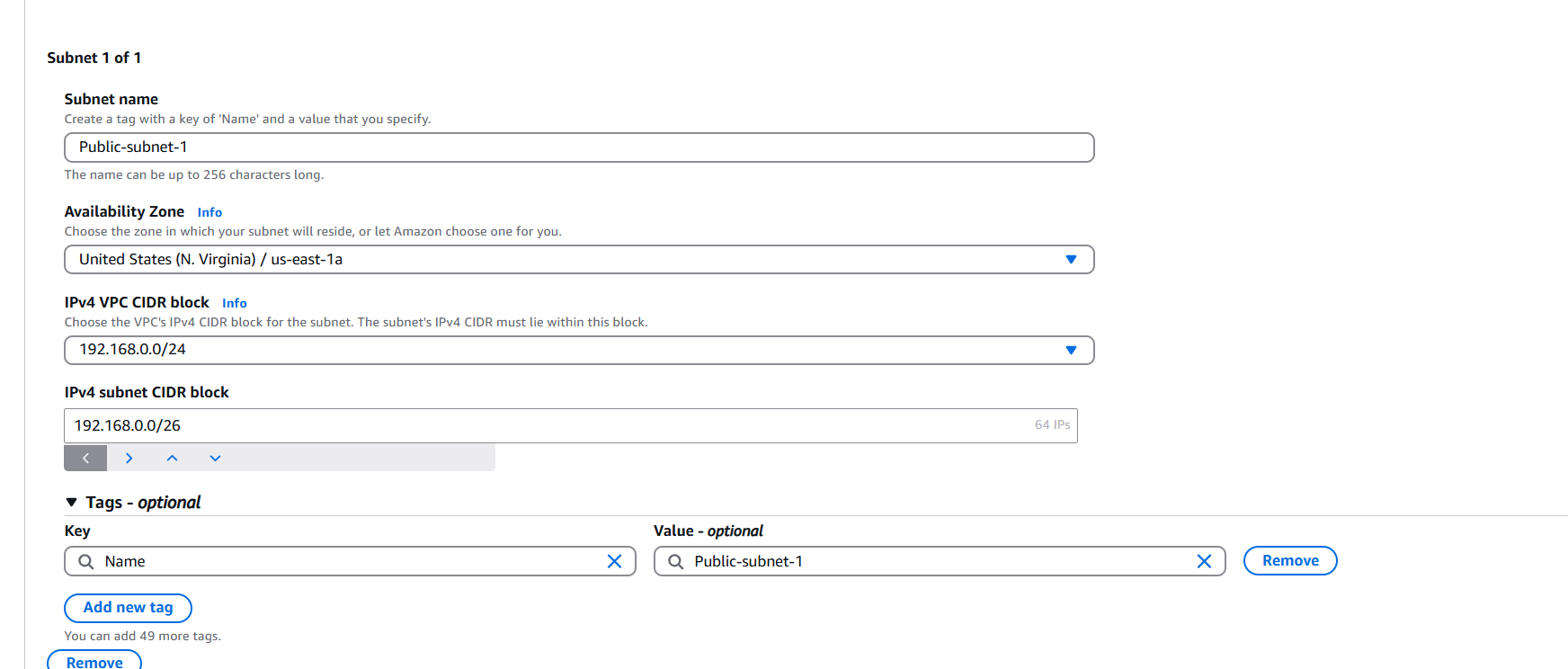
- IPv4 CIDR block (e.g., 192.168.0.128/26 for private subnet).

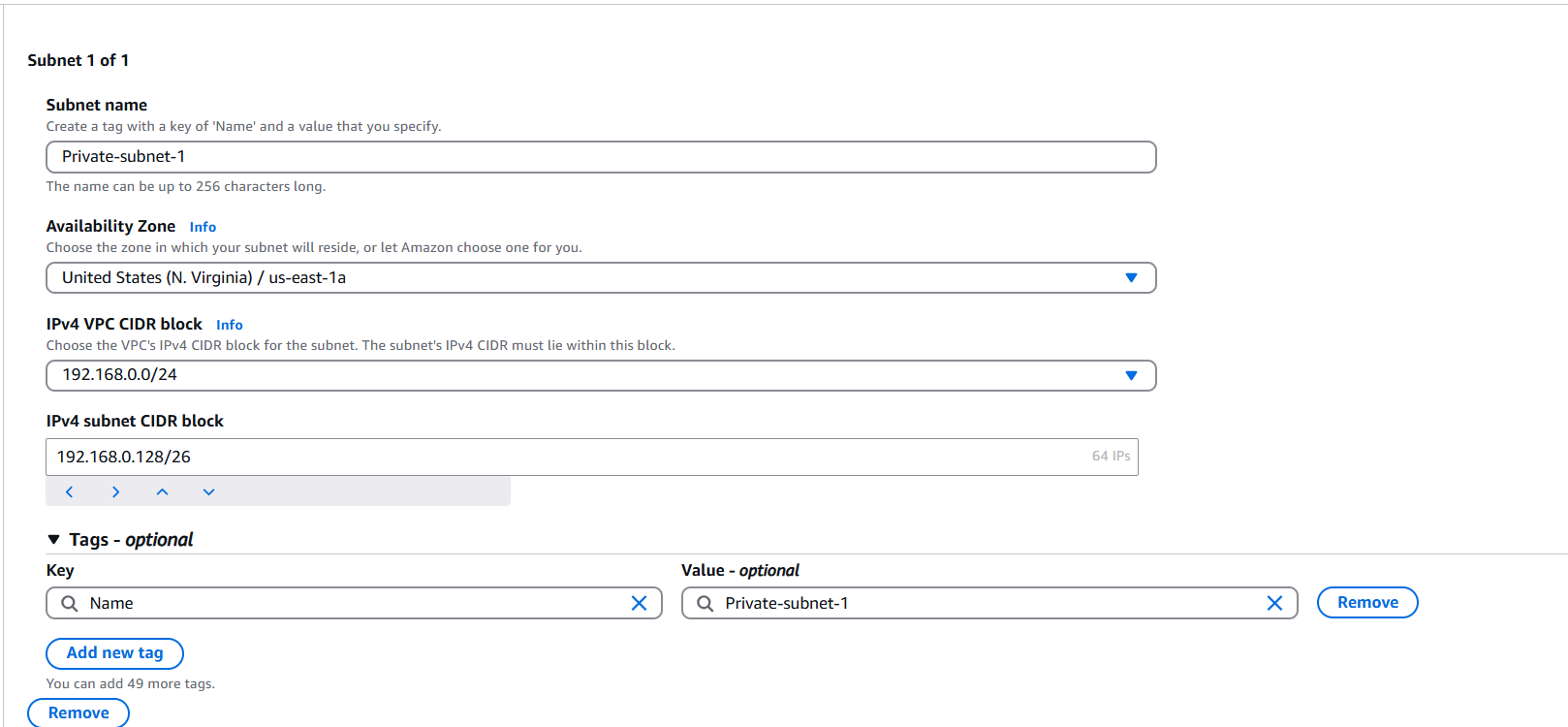












**2) Enable VPC peering for cross region.**

Create VPCs in Both Regions

- Sign in to the AWS Management Console.

- Navigate to VPC service.

- Create a VPC in North Virginia (us-east-1):

- Name: VPC-NVirginia

- CIDR Block: 192.168.0.0/24

- Create a VPC in Ohio (us-east-2):

- Name: VPC-Ohio

- CIDR Block: 172.31.0.0/16

Step 2: Create VPC Peering Connection

- Go to VPC Peering Connections in the AWS Console.

- Click Create Peering Connection.

- Enter:

- Name: NVirginia-Ohio-Peering

- Requester VPC: VPC-NVirginia

- Accepter VPC: VPC-Ohio

- Region: Select Another Region (us-east-2).

- Click Create Peering Connection.

- Accept the peering request in Ohio (us-east-2):

- Navigate to VPC Peering Connections in Ohio.

- Select the request and click Accept.

pcx-00abb237ca6d7d4d9

Update Route Tables

- Go to Route Tables in North Virginia (us-east-1).

- Select the route table associated with VPC-NVirginia.

- Add a new route:

- Destination: 172.31.0.0/16

- Target: Select the VPC Peering Connection (NVirginia-Ohio-Peering).

- Repeat the process for Ohio (us-east-2):

- Destination: 192.168.0.0/24

- Target: Select the VPC Peering Connection.

Update Security Groups

- Go to Security Groups in both regions.

- Allow inbound traffic for:

- Protocol: All Traffic or specific ports (SSH - 22, HTTP - 80, etc.).

- Source: 172.31.0.0/16 (for Ohio) and 192.168.0.0/24 (for North Virginia).

Test Connectivity

- Launch EC2 instances in both VPCs.

- Try pinging or SSH between instances using private IPs.

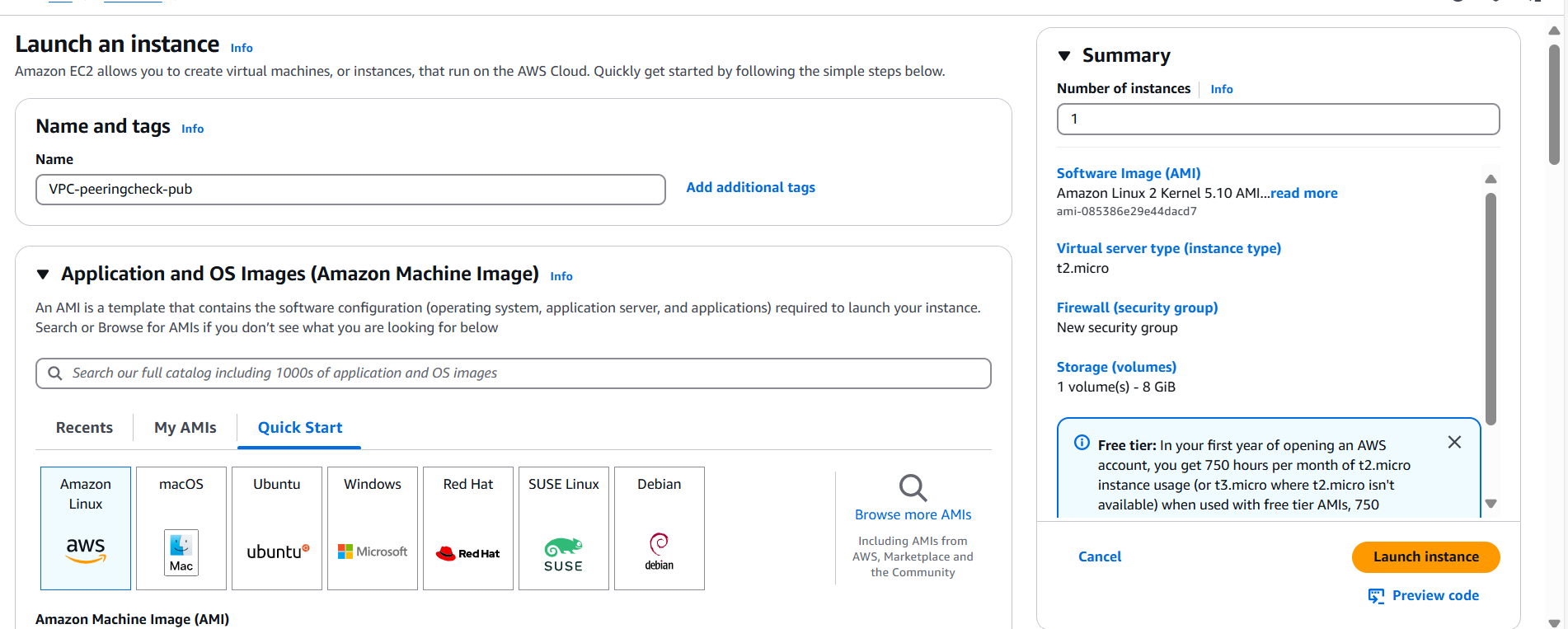
- If connectivity fails, check:

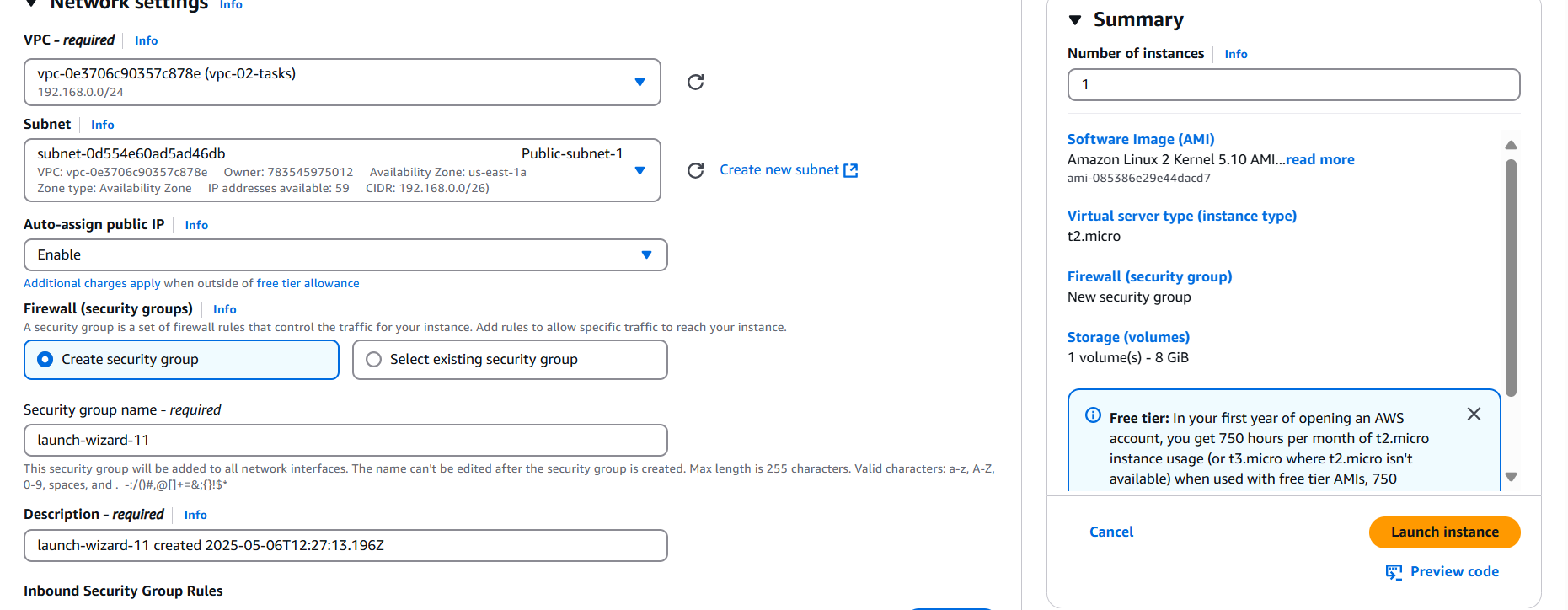
- Route tables.

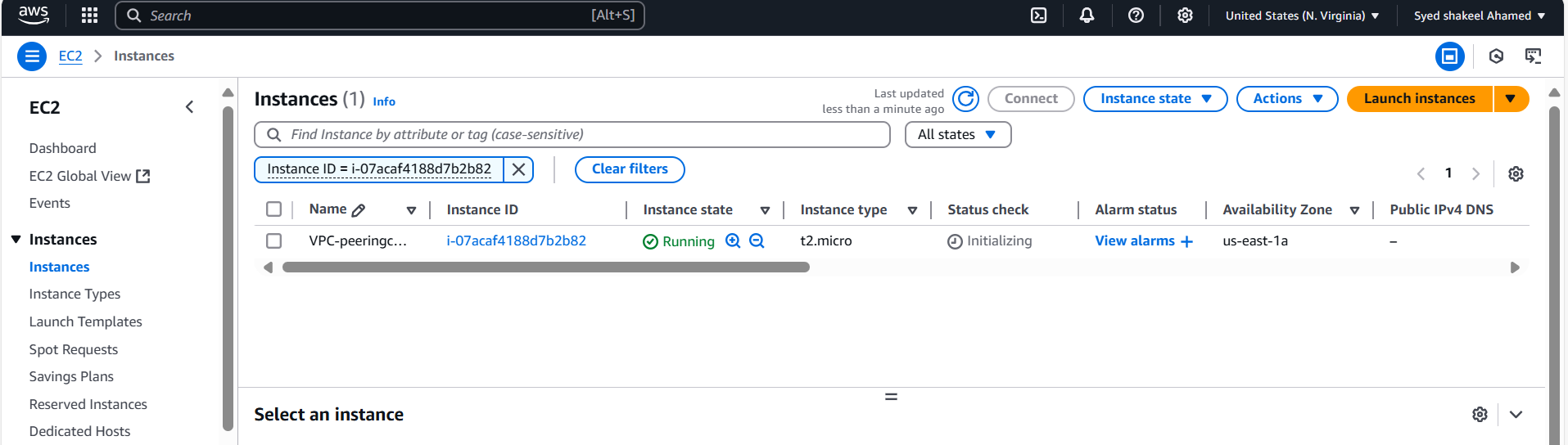
- Security group rules.

- Peering connection status.

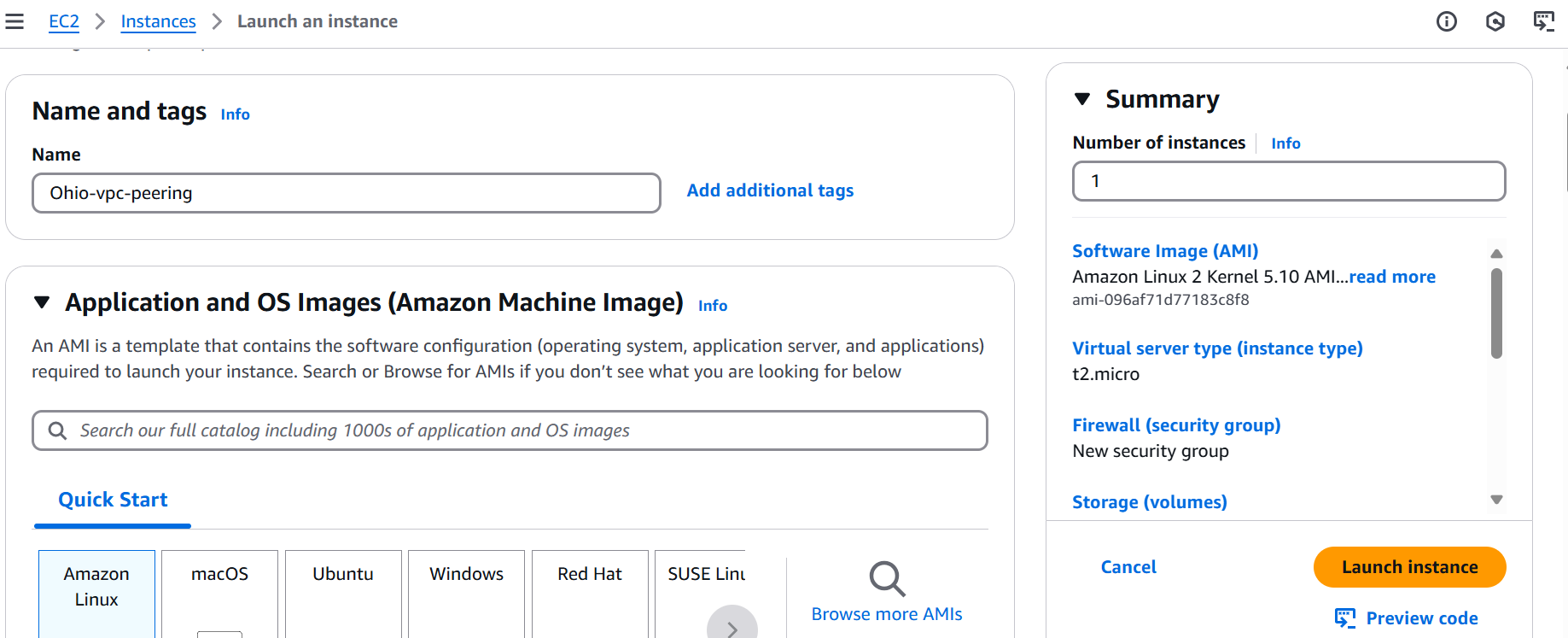
N.virginia-Public Ec2-instance

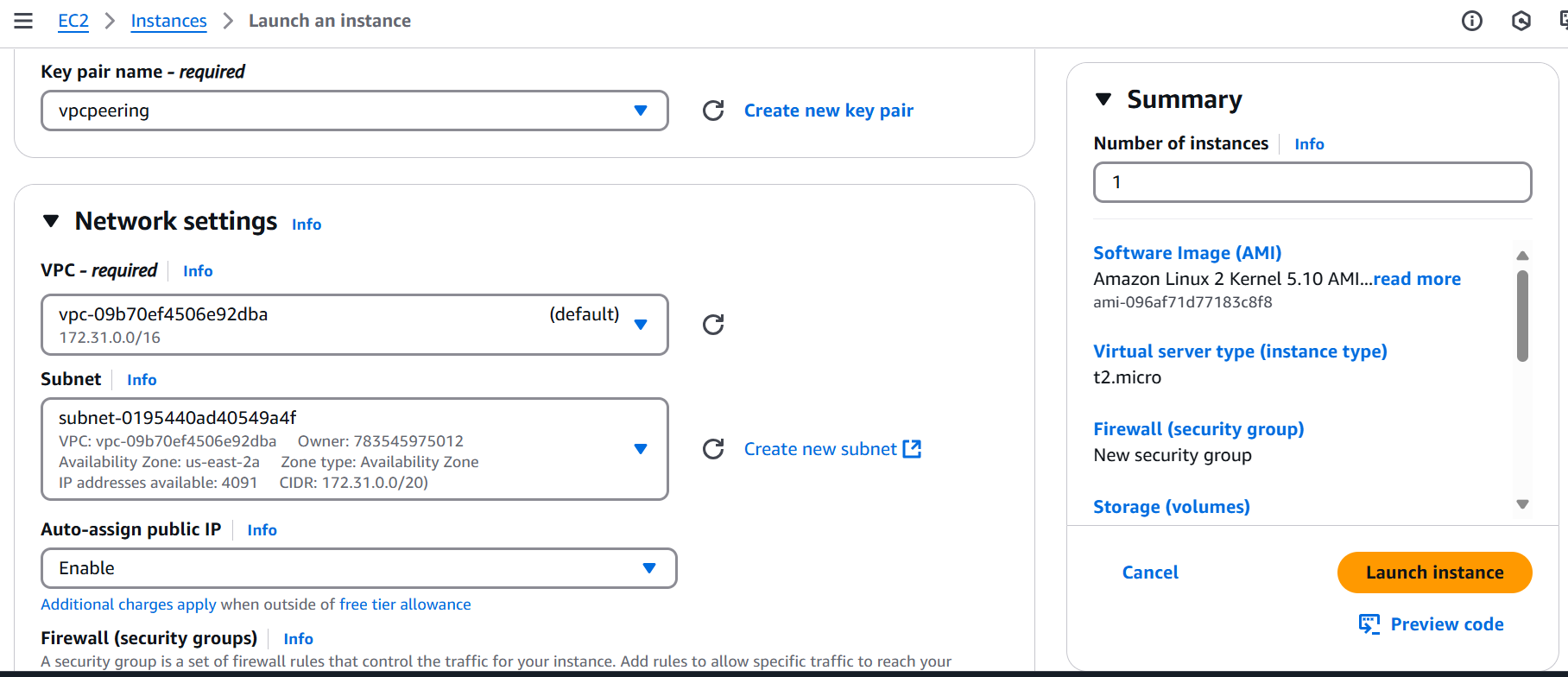


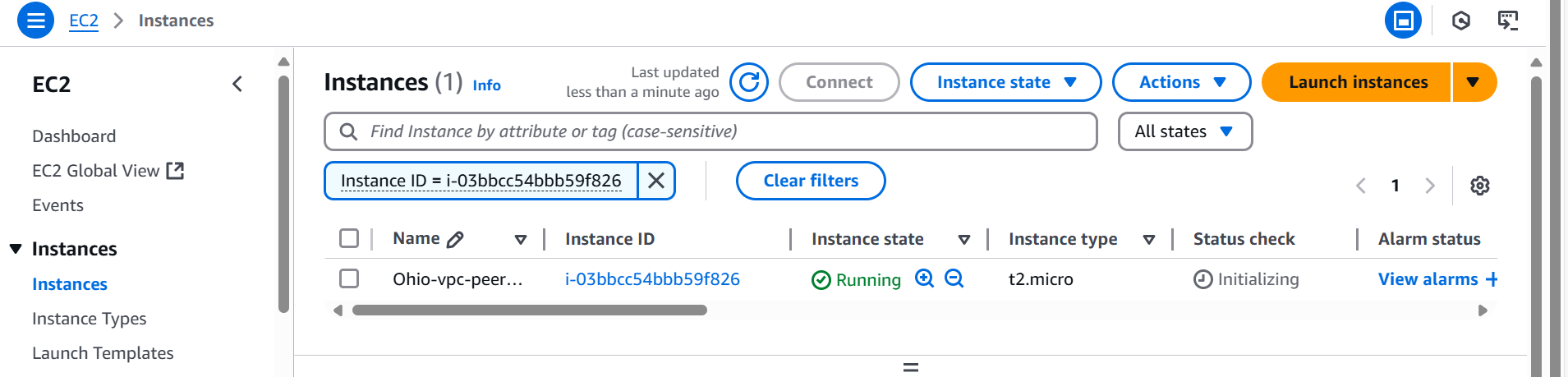




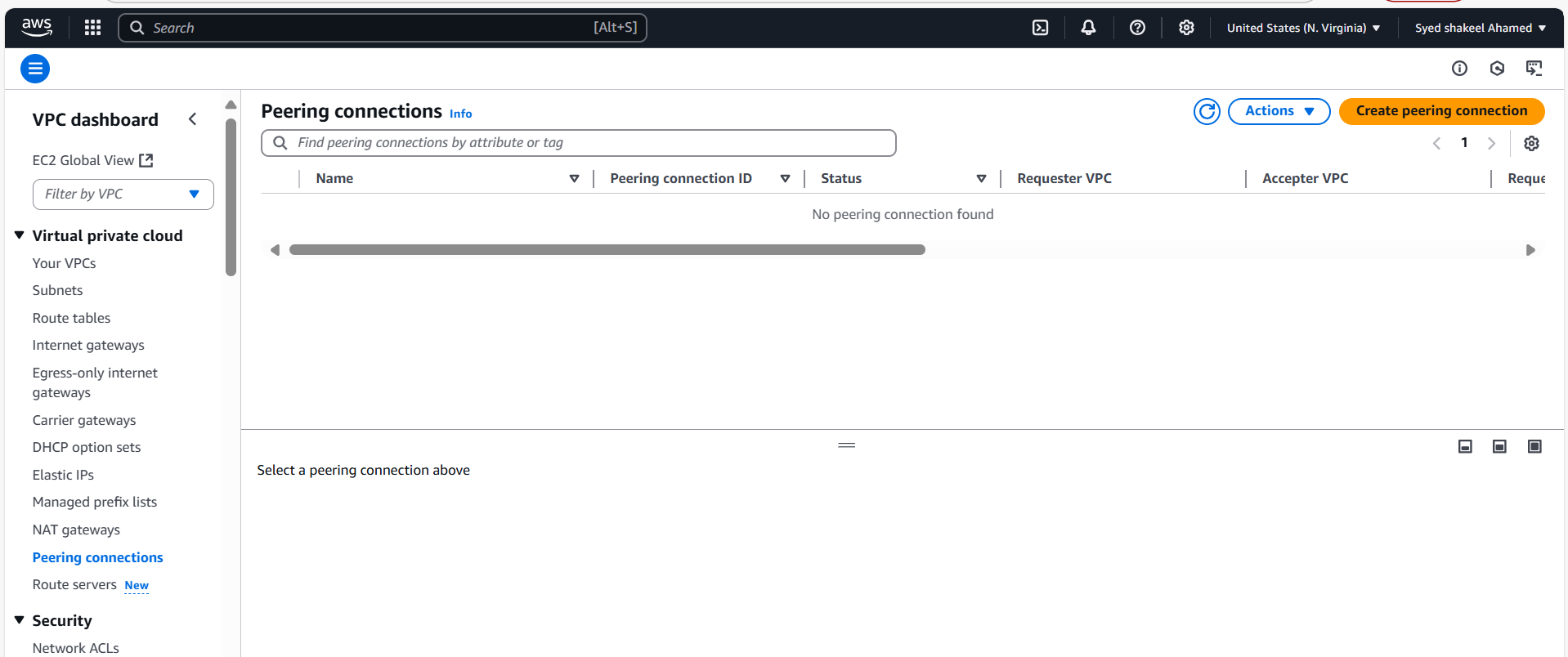
Ohio-Ec2 instance

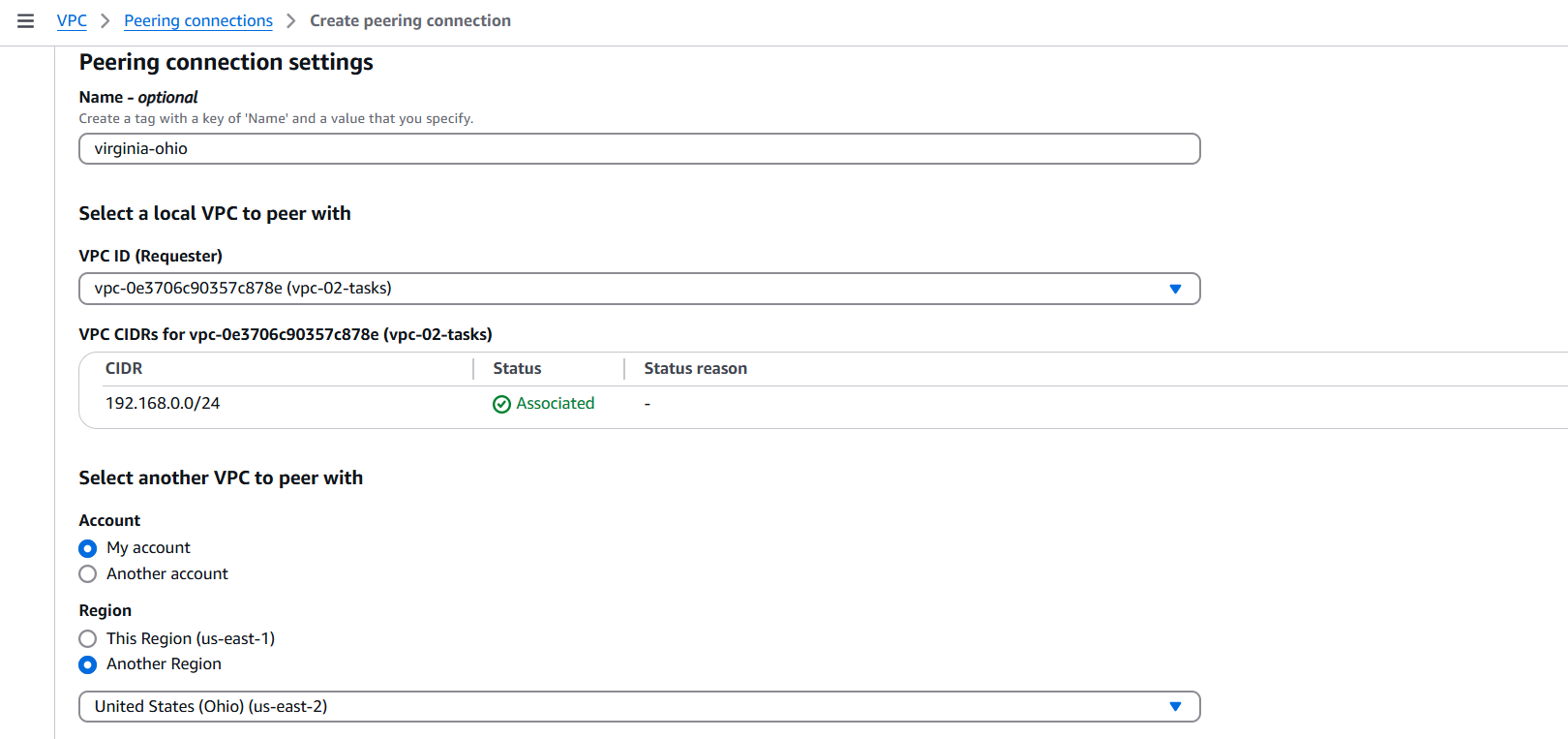


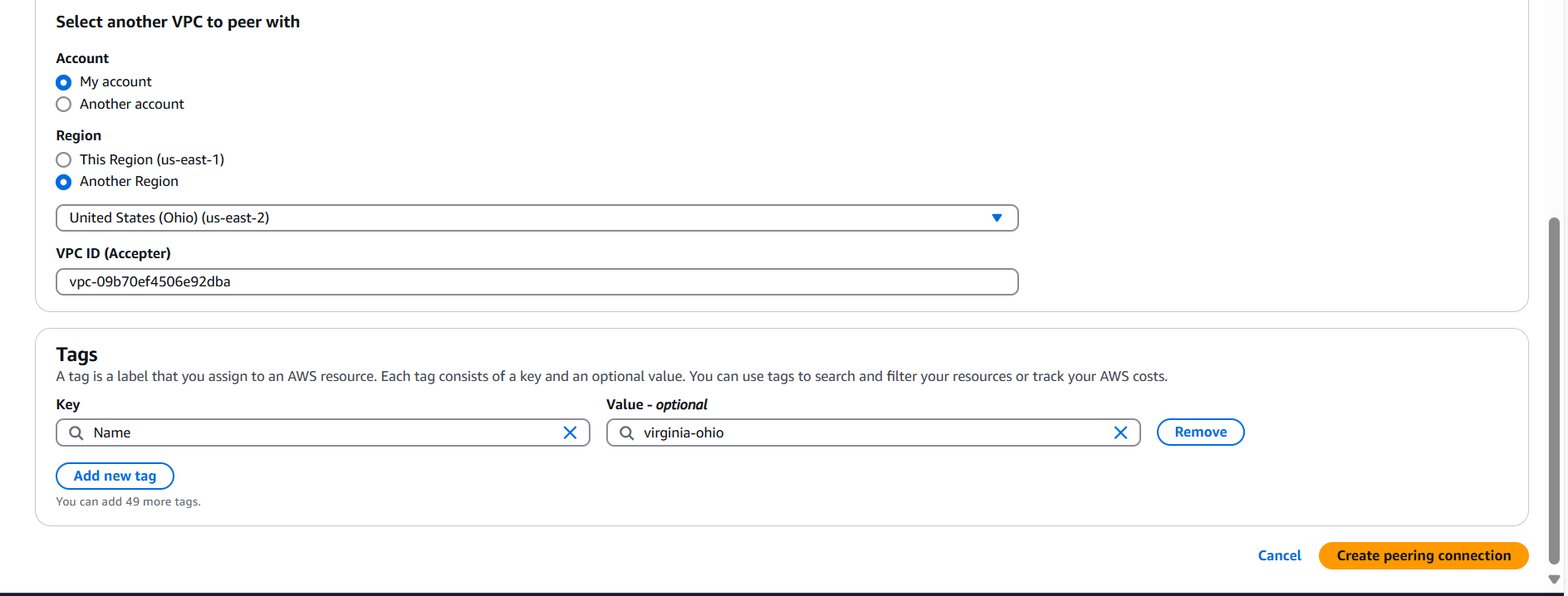


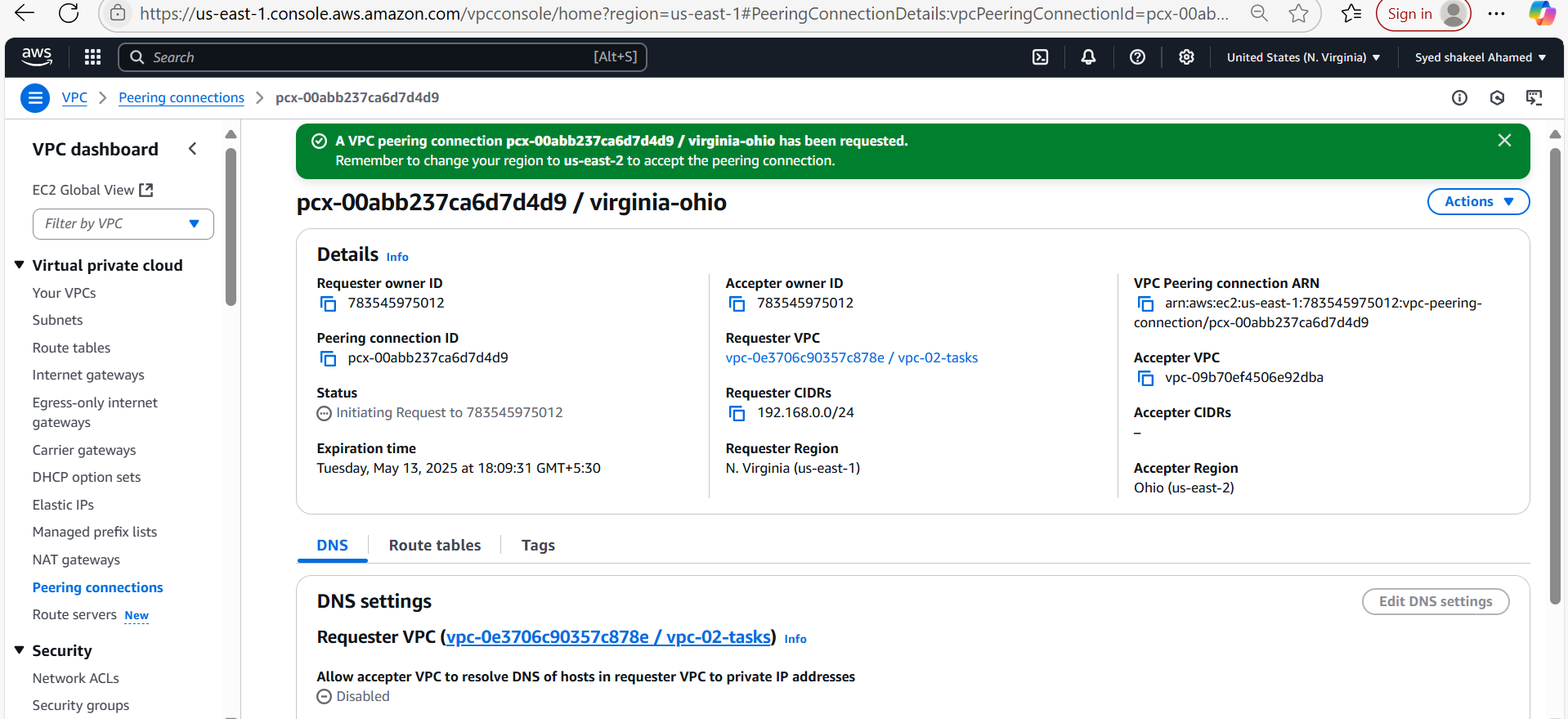


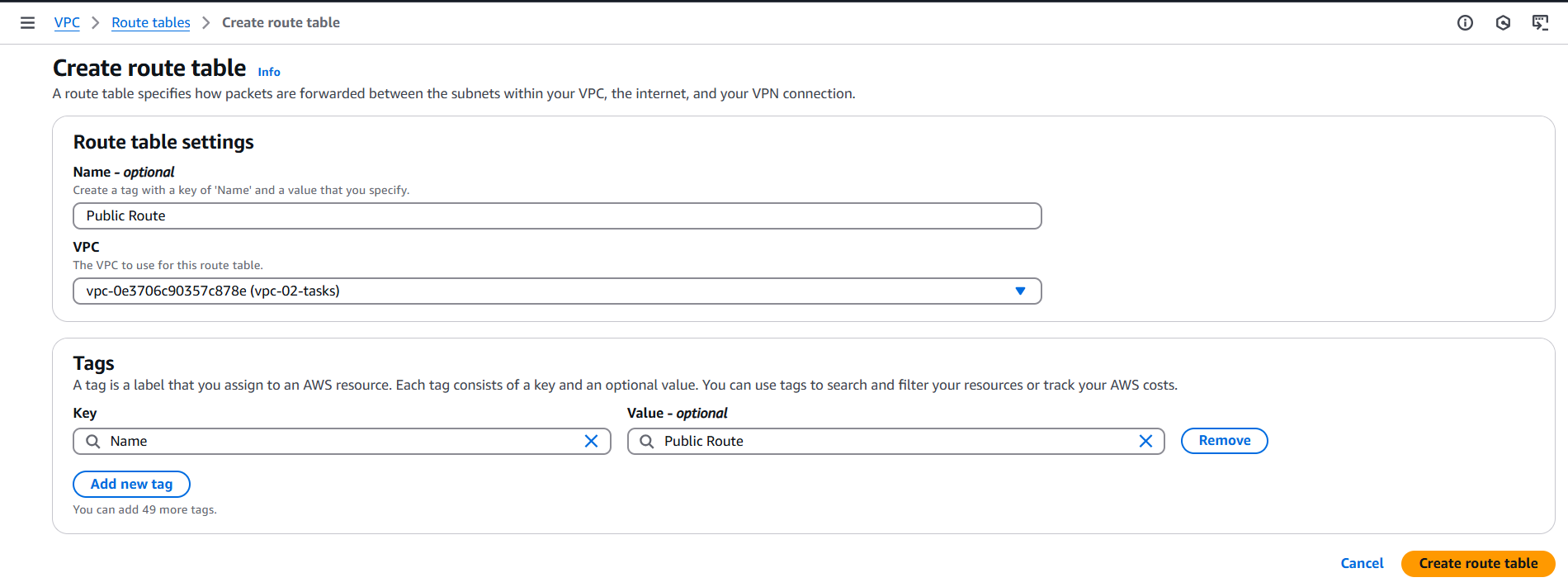
VPC-Peering

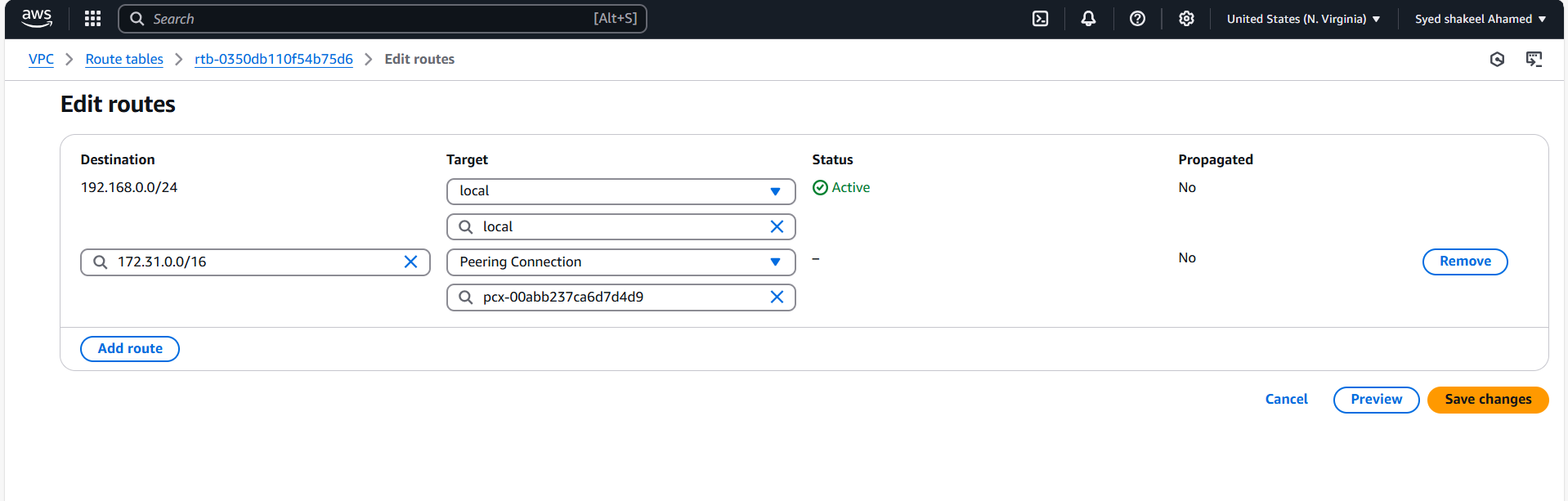


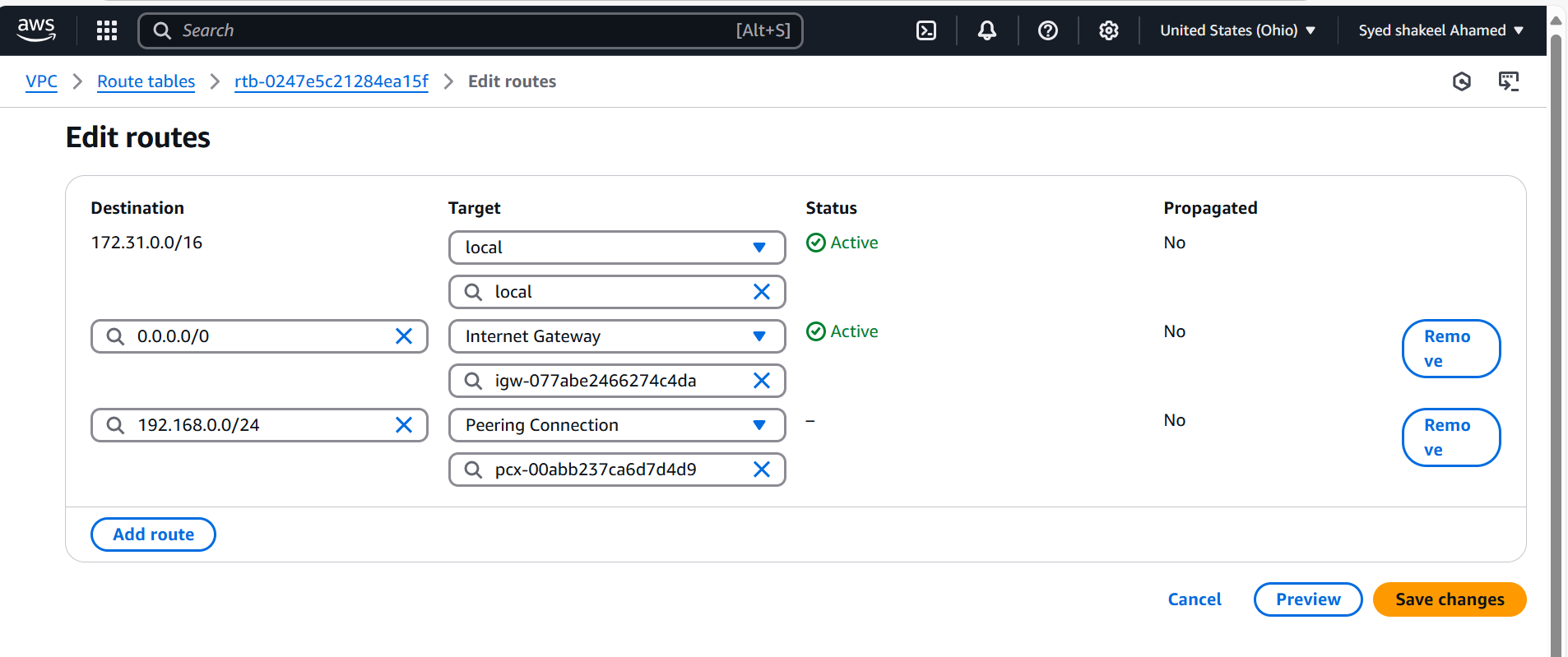


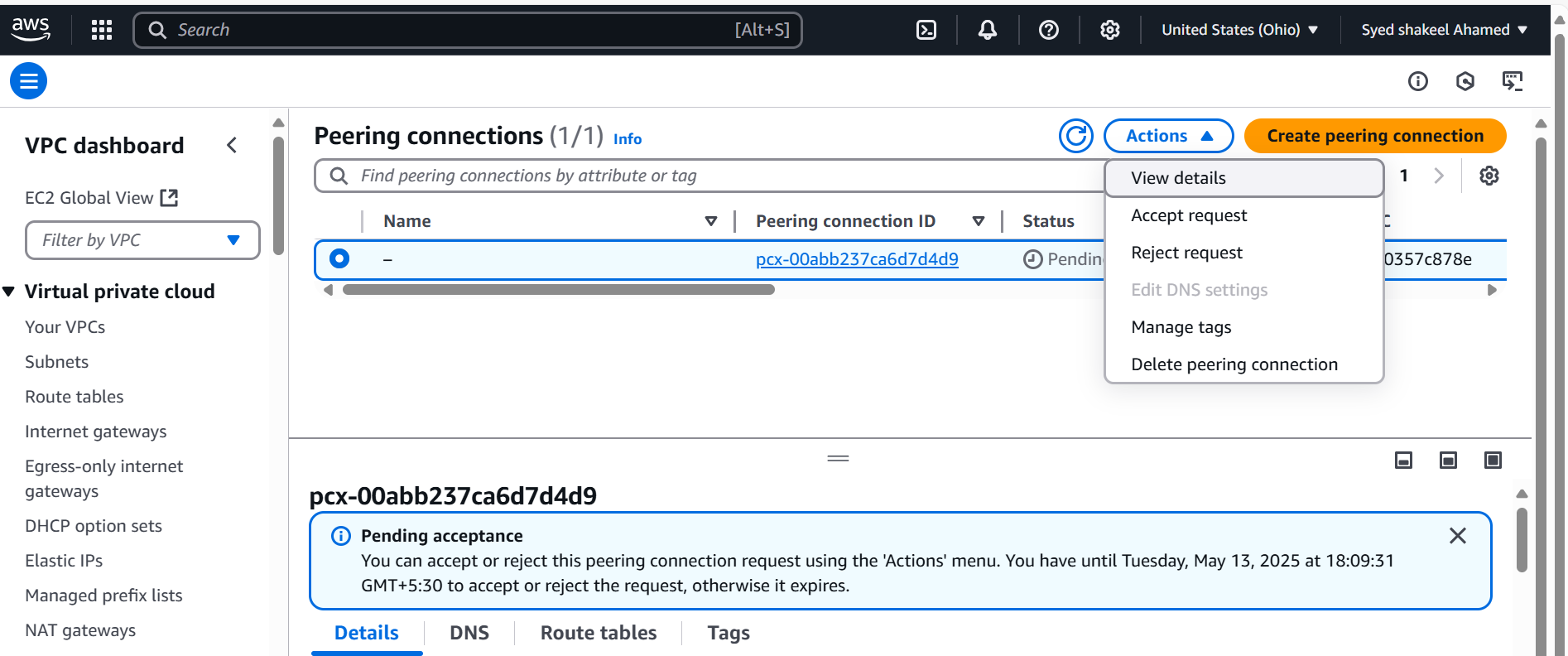


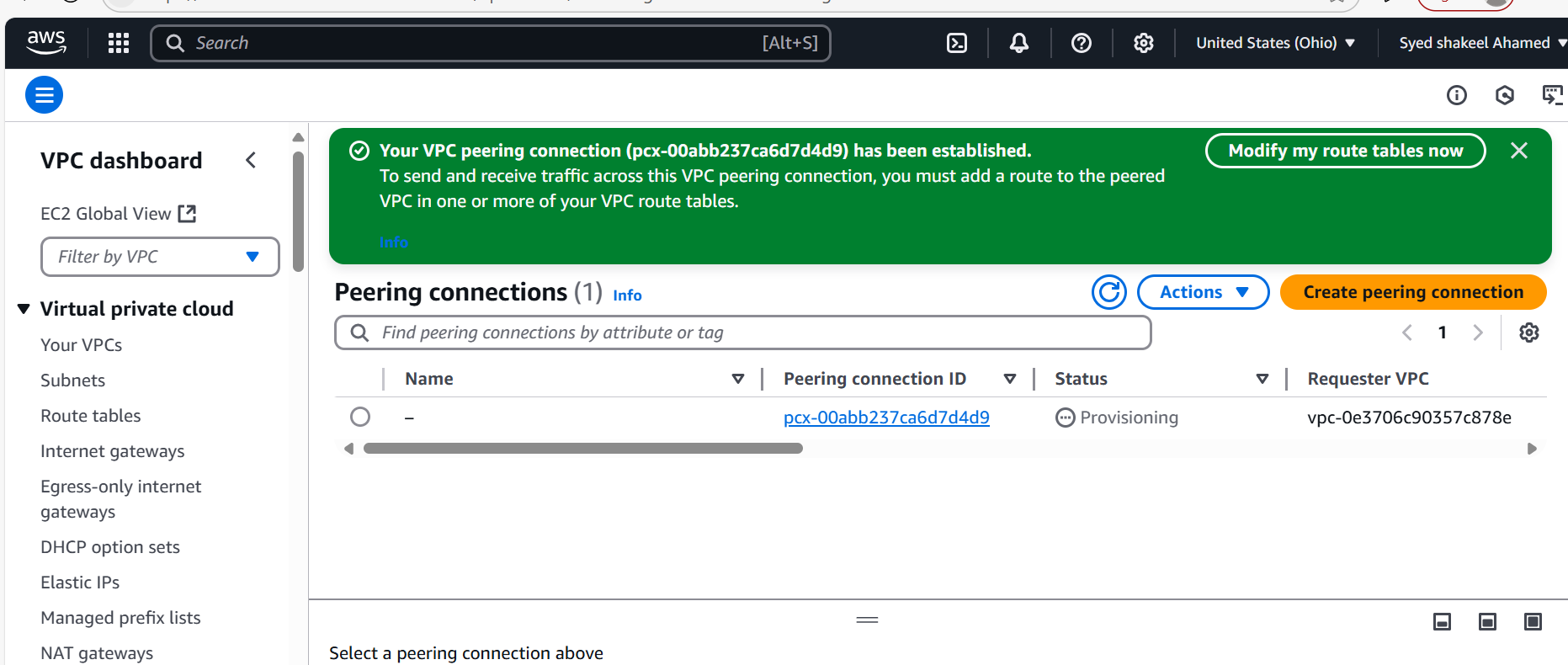


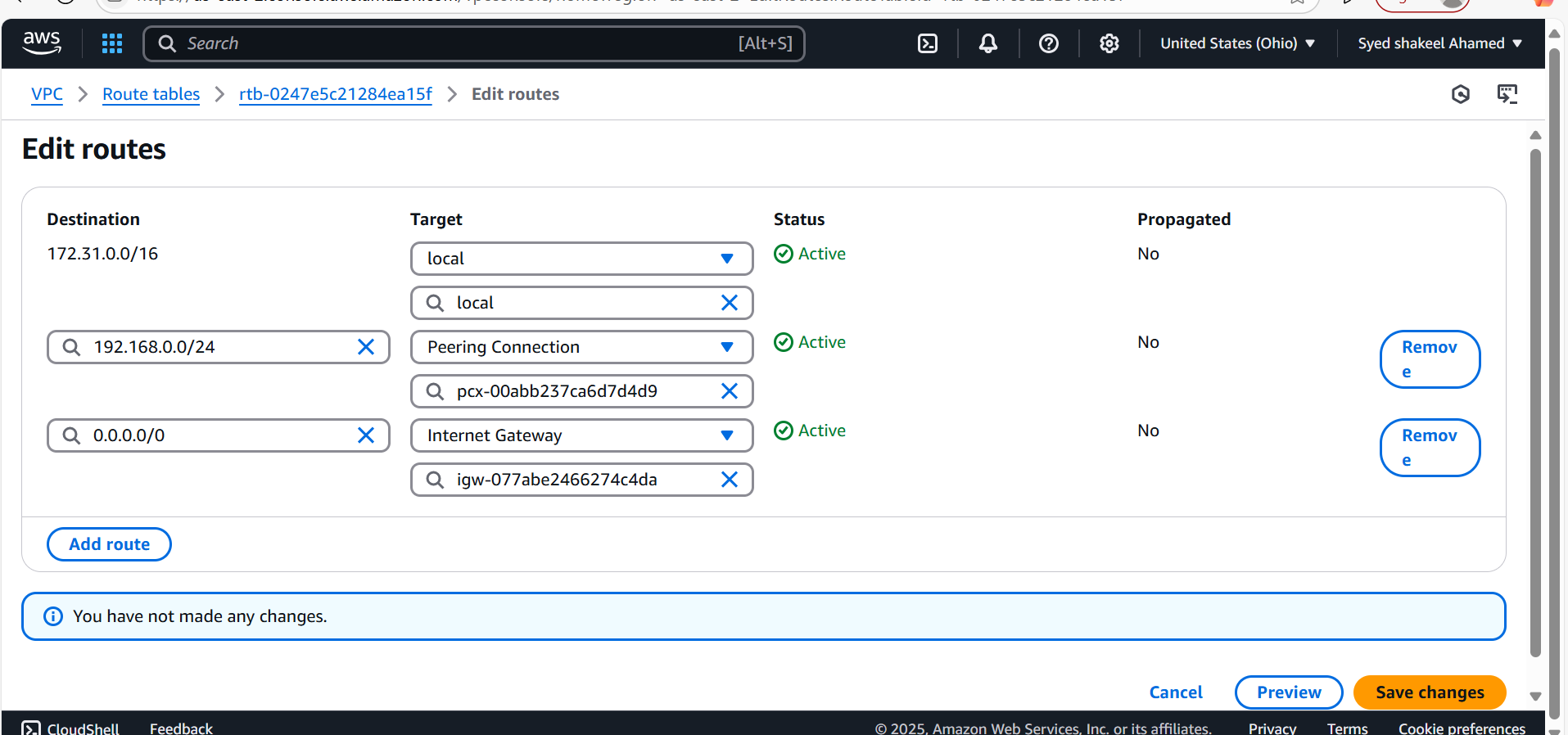


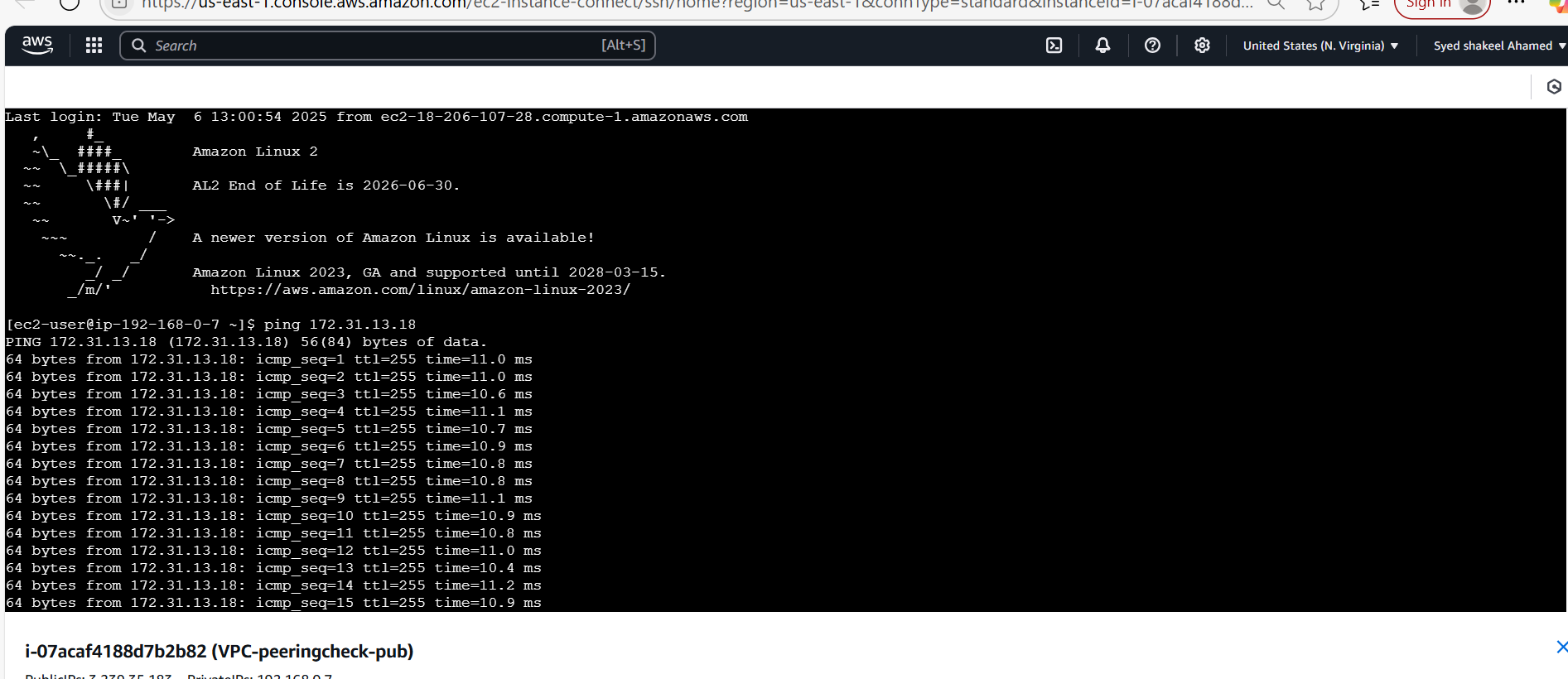












**3) Enable VPC peering for cross account. (You can collaborate with your friend and do this task).**

**4) Setup VPC Transits gateway.**

Amazon VPC Transit Gateways is a network transit hub used to interconnect virtual private clouds (VPCs) and on-premises networks. As your cloud infrastructure expands globally, inter-Region peering connects transit gateways together using the AWS Global Infrastructure. All network traffic between AWS data centers is automatically encrypted at the physical layer.

Transit gateway concepts The following are the key concepts for transit gateways: • Attachments — You can attach the following: • One or more VPCs • A Connect SD-WAN/third-party network appliance • An AWS Direct Connect gateway • A peering connection with another transit gateway • A VPN connection to a transit gateway • Transit gateway Maximum Transmission Unit (MTU) — The maximum transmission unit (MTU) of a network connection is the size, in bytes, of the largest permissible packet that can be passed over the connection. The larger the MTU of a connection, the more data that can be passed in a single packet. A transit gateway supports an MTU of 8500 bytes for traffic between VPCs, AWS Direct Connect, Transit Gateway Connect, and peering attachments (intra-Region, inter-Region, and Cloud WAN peering attachments). Traffic over VPN connections can have an MTU of 1500 bytes. • Transit gateway route table — A transit gateway has a default route table and can optionally have additional route tables. A route table includes dynamic and static routes that decide the next hop based on the destination IP address of the packet. The target of these routes could be any transit gateway attachment. By default, transit gateway attachments are associated with the default transit gateway route table. • Associations — Each attachment is associated with exactly one route table. Each route table can be associated with zero to many attachments. • Route propagation — A VPC, VPN connection, or Direct Connect gateway can dynamically propagate routes to a transit gateway route table. With a Connect attachment, the routes are Transit gateway concepts 1 Amazon VPC AWS Transit Gateway propagated to a transit gateway route table by default. With a VPC, you must create static routes to send traffic to the transit gateway. With a VPN connection, routes are propagated from the transit gateway to your on-premises router using Border Gateway Protocol (BGP). With a Direct Connect gateway, allowed prefixes are originated to your on-premises router using BGP. With a peering attachment, you must create a static route in the transit gateway route table to point to the peering attachment

Work with transit gateways

You can create, access, and manage your transit gateways using any of the following interfaces:

• AWS Management Console — Provides a web interface that you can use to access your transit gateways.

• AWS Command Line Interface (AWS CLI) — Provides commands for a broad set of AWS services, including Amazon VPC, and is supported on Windows, macOS, and Linux. For more information, see AWS Command Line Interface.

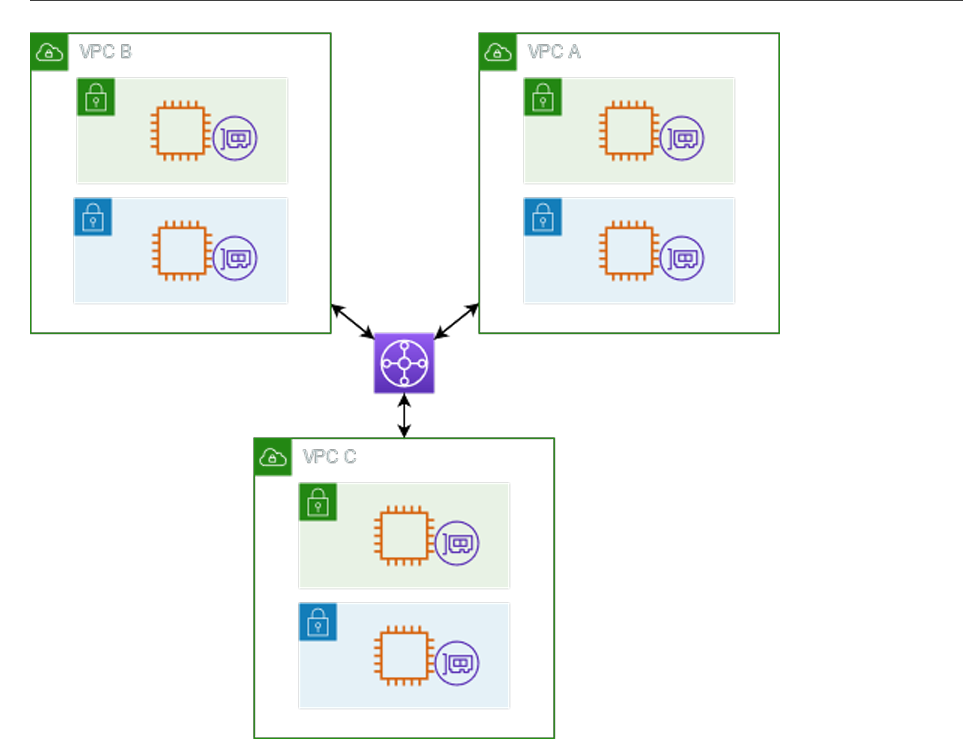
• AWS SDKs — Provides language-specific API operations and takes care of many of the connection details, such as calculating signatures, handling request retries, and handling errors.

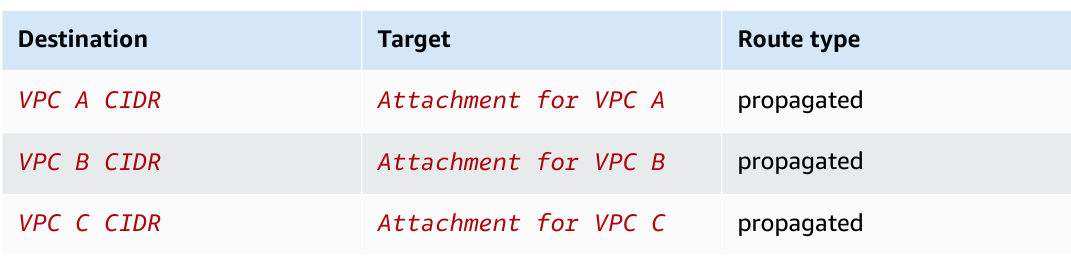
• Query API — Provides low-level API actions that you call using HTTPS requests. Using the Query API is the most direct way to access Amazon VPC, but it requires that your application handle low-level details such as generating the hash to sign the request, and handling errors.

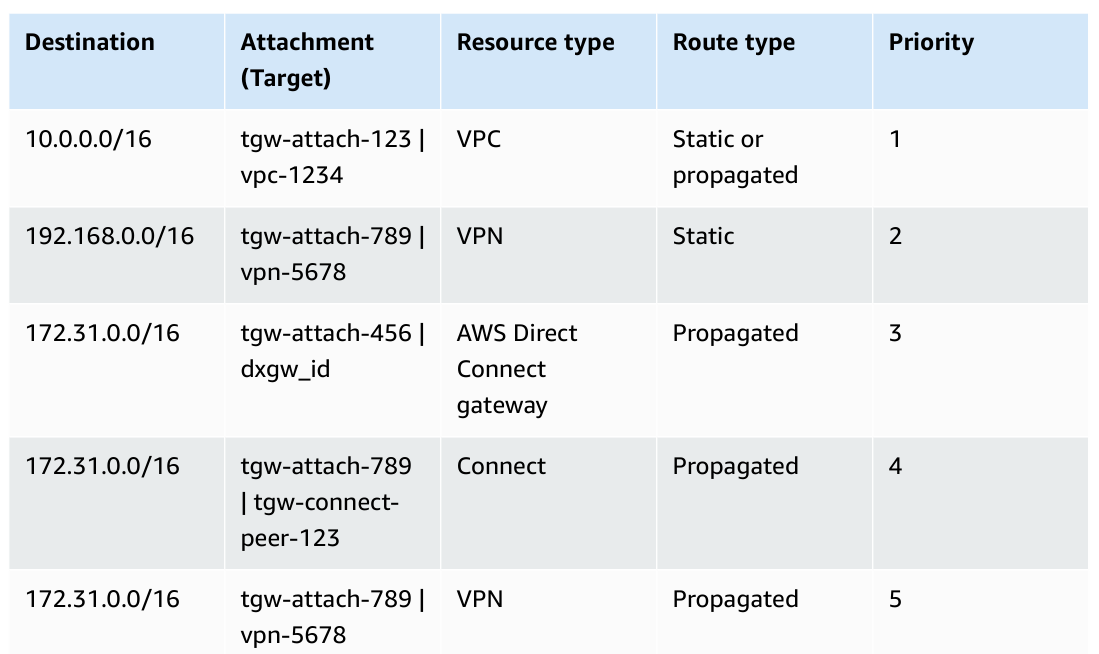
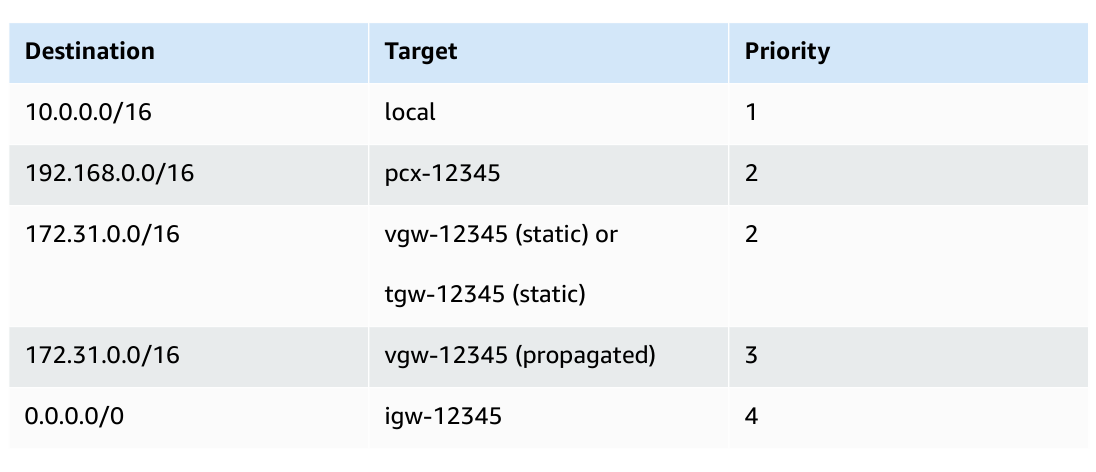
How Amazon VPC Transit Gateways work

In AWS Transit Gateway a transit gateway acts as a Regional virtual router for traffic flowing between your virtual private clouds (VPCs) and on-premises networks. A transit gateway scales elastically based on the volume of network traffic. Routing through a transit gateway operates at layer 3, where the packets are sent to a specific next-hop attachment, based on their destination IP addresses

Example architecture diagram The following diagram shows a transit gateway with three VPC attachments. The route table for each of these VPCs includes the local route and routes that send traffic destined for the other two VPCs to the transit gateway.







Create the transit gateway

When you create a transit gateway, we create a default transit gateway route table and use it as the default association route table and the default propagation route table. Prerequisites 35 Amazon VPC <https://console.aws.amazon.com/vpc/>.

AWS Transit Gateway To create a transit gateway

1. Open the Amazon VPC console at

2. In the Region selector, choose the Region that you used when you created the VPCs.

3. On the navigation pane, choose Transit Gateways.

4. Choose Create transit gateway.

5. (Optional) For Name tag, enter a name for the transit gateway. This creates a tag with "Name" as the key and the name that you specified as the value.

6. (Optional) For Description, enter a description for the transit gateway.

7. In Configure the transit gateway section, do the following:

1. For Amazon side Autonomous System Number (ASN), enter the private ASN for your transit gateway. This should be the ASN for the AWS side of a Border Gateway Protocol (BGP) session.

The range is from 64512 to 65534 for 16-bit ASNs. The range is from 4200000000 to 4294967294 for 32-bit ASNs.

If you have a multi-Region deployment, we recommend that you use a unique ASN for each of your transit gateways.

2. (Optional) Choose whether to enable any of the following:

• DNS support for VPCs attached to this transit gateway.

• VPN ECMP support for VPN connections attached to the transit gateway.

• Default route table association, which automatically associates transit gateway attachments with this transit gateway's default route table.

• Default route table propagation, which automatically propagates route table attachments to this transit gateway's default route table.

• Multicast support, which allows you to create multicast domains in this transit gateway.

8. (Optional) In the Configure-cross-account sharing options section, choose whether to Auto accept shared attachments. If enabled, attachments are automatically accepted. Otherwise, you must accept or reject attachment requests.

9. (Optional) In the Transit gateway CIDR blocks section, add a size /24 CIDR block or larger for IPv4 addresses or /64 block or larger CIDR block for IPv6 addresses. You can associate

Step 1: Create the transit gateway 36 Amazon VPC AWS Transit Gateway any public or private IP address range, except for addresses in the 169.254.0.0/16 range, and ranges that overlap with the addresses for your VPC attachments and on-premises networks. Note Transit gateway CIDR blocks are used if you are configuring Connect (GRE) attachments or PrivateIP VPNs. Transit Gateway assigns IPs for the Tunnel endpoints (GRE/PrivateIP VPN) from this range.

10. (Optional) Add key-value tags to this transit gateway to further help identify it.

1. Choose Add new tab.

2. Enter a Key name and associated Value.

3. Choose Add new tag to add additional tags, or skip to the next step.

11. Choose Create transit gateway. When the gateway is created, the initial state of the transit gateway is pending. Step 2: Attach your VPCs to your transit gateway Wait until the transit gateway you created in the previous section shows as available before proceeding with creating an attachment.

Create an attachment for each VPC. Confirm that you have created two VPCs and launched an EC2 instance in each, as described in Prerequisites. Create a transit gateway attachment to a VPC

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

2. On the navigation pane, choose Transit Gateway Attachments.

3. Choose Create transit gateway attachment.

4. (Optional) For Name tag, enter a name for the attachment.

5. For Transit gateway ID, choose the transit gateway to use for the attachment.

6. For Attachment type, choose VPC.

7. Choose whether to enable DNS support. For this exercise, do not enable IPv6 support.

8. For VPC ID, choose the VPC to attach to the transit gateway. Step 2: Attach your VPCs to your transit gateway 37 Amazon VPC AWS Transit Gateway

9. For Subnet IDs, select one subnet for each Availability Zone to be used by the transit gateway to route traffic. You must select at least one subnet. You can select only one subnet per Availability Zone.

10. Choose Create transit gateway attachment. Each attachment is always associated with exactly one route table. Route tables can be associated with zero to many attachments. To determine the routes to configure, decide on the use case for your transit gateway, and then configure the routes. For more information, see the section called “Example transit gateway scenarios”. Step 3: Add routes between the transit gateway and your VPCs A route table includes dynamic and static routes that determine the next hop for associated VPCs based on the destination IP address of the packet. Configure a route that has a destination for non-local routes and the target of the transit gateway attachment ID. For more information, see Routing for a transit gateway in the Amazon VPC User Guide.

To add a route to a VPC route table

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

2. On the navigation pane, choose Route Tables.

3. Choose the route table associated with your VPC.

4. Choose the Routes tab, then choose Edit routes.

5. Choose Add route.

6. In the Destination column, enter the destination IP address range. For Target, choose Transit Gateway, and then choose the transit gateway ID.

7. Choose Save changes.

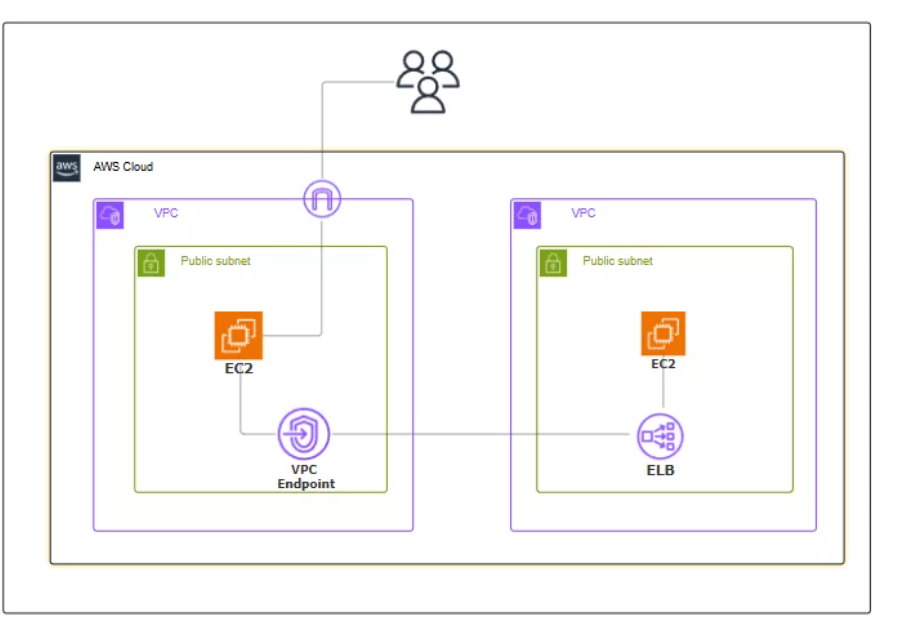
Test the transit gateway You can confirm that the transit gateway was successfully created by connecting to an Amazon EC2 instance in each VPC, and then sending data between them, such as a ping command.

**5) Setup VPC End Point**

In modern cloud architectures, ensuring secure and efficient communication between services is critical. One of the most effective ways to do this within AWS is by using VPC (Virtual Private Cloud) Endpoint services.

VPC endpoint’s role in connecting resources securely without requiring access over the public internet.

* **Interface Endpoints (PrivateLink)**: Direct connection to services like AWS Lambda, EC2, etc.
* **Gateway Endpoints**: For services like S3 or DynamoDB.



Navigate the VPC and Click on create VPC.

* Enter the VPC name.
* Enter Ipv4 CIDR.
* Click on create VPC.

Click on Internet gateway and create internet gateway.

* Enter the gate name.
* Click on create internet gateway.

Click on your created internet gateway and attach to your created VPC

Next, Click on rout tables and Click on create route tables.

Next, create subnet.

* Enter subnet name.
* Select AZ.
* Enter IPV4.

Go to your created route table.

* Select subnet associations.
* Select your subnet.
* Click on save associations.

Now, Navigate the EC2 instance and click on create instance.

* Enter the name and select AMI, instance type.
* Create keypair.
* Configure the security group with rules allowing **SSH, HTTP,** and **HTTPS** traffic from all IP addresses.
* Configure Apache HTTP server on the EC2 instance to serve a custom HTML page.
* Click on create instance.

Now you created network load balancer.

* Enter load balancer name.
* Select vpc, security group and target.
* TCP listener on port **80** and associate it with the EC2 instance. Ensure that the load balancer and the EC2 instance use the same security group for consistent security policies.
* Click on create.

Now, you go to create customer VPC.

* Enter the name.
* Enter IPV4 CIDR.
* Click on create VPC.

Create customer IGW and attach your created customer VPC

Create subnet and enter the name.

* Select VPC and IPV4.
* Click on create subnet.

Create route table.

* Enter the name.
* Select VPC.
* Click on create route table.

Select your created route table.

* Click on subnet association.
* Select your created subnet.

Create instance.

* Enter the instance name, AMI, instance type.
* Select your keypair.
* Configure the security group with rules allowing **SSH, HTTP,** and **HTTPS** traffic from all IP addresses.
* Click on create instance.

Create VPC End point.

* Enter the name.
* Select type.
* And select security group and VPC.

## **Test the End-to-End Connection.**

After creating the VPC Endpoint and configuring routing and security, test the connection to ensure that traffic from the consumer VPC can reach the service through the VPC Endpoint:

1. **From the consumer side**, use a tool like curl, wget, or any application to make a request to the private endpoint.
2. **Monitor logs** from your services and load balancer to ensure the requests are being routed correctly.
3. **Verify DNS resolution** if you enabled private DNS. Ensure the correct endpoint DNS resolves to the private IPs.