VIRTUAL PRIVATE CLOUD PEERING

VPC Peering in AWS is a networking connection between two Virtual Private Clouds (VPCs) that allows traffic to flow between them using private IP addresses. It's commonly used to connect VPCs in the same or different AWS regions.

Here are the key aspects:

1. **Private Communication**: Once VPC peering is established, instances in one VPC can communicate with instances in the other VPC as if they were in the same network. This communication happens over private IP addresses.
2. **No Transitive Peering**: VPC peering is a one-to-one connection. If VPC-A is peered with VPC-B, and VPC-B is peered with VPC-C, instances in VPC-A cannot directly communicate with instances in VPC-C, unless you create an additional peering connection between VPC-A and VPC-C.
3. **Regional & Inter-Region Peering**: VPC peering can occur within the same region (Intra-Region Peering) or across different AWS regions (Inter-Region Peering).
4. **No Overlapping CIDRs**: The IP address ranges (CIDRs) of the two VPCs involved in the peering connection must not overlap.
5. **Routing**: For traffic to flow between the VPCs, you must update the route tables in both VPCs to allow the communication.
6. **Security**: Security groups and Network Access Control Lists (NACLs) still apply, so communication between the VPCs can be restricted based on your configurations.

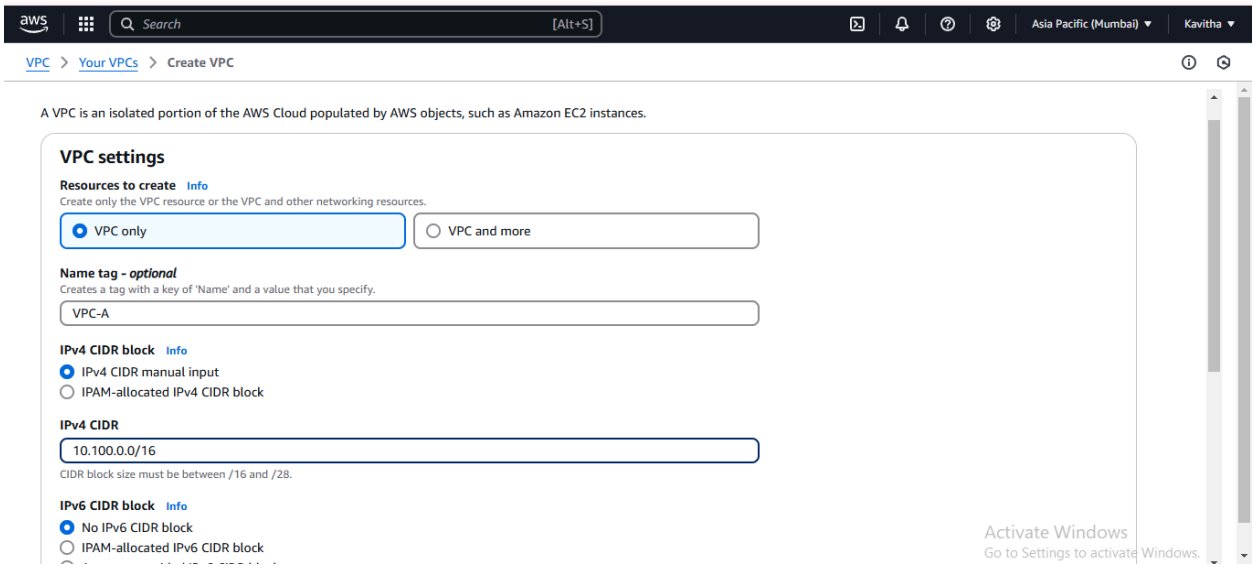
TASK: To enable communication between instances in two Virtual Private Clouds (VPCs) using VPC Peering.

Create 2 VPC'S, Internet gateway & attach to VPC, create 2 public & private subnets for VPC A and private subnet for VPC B, 2 route tables for VPC A then one route table for VPC B.

Go to **AWS console Management**:

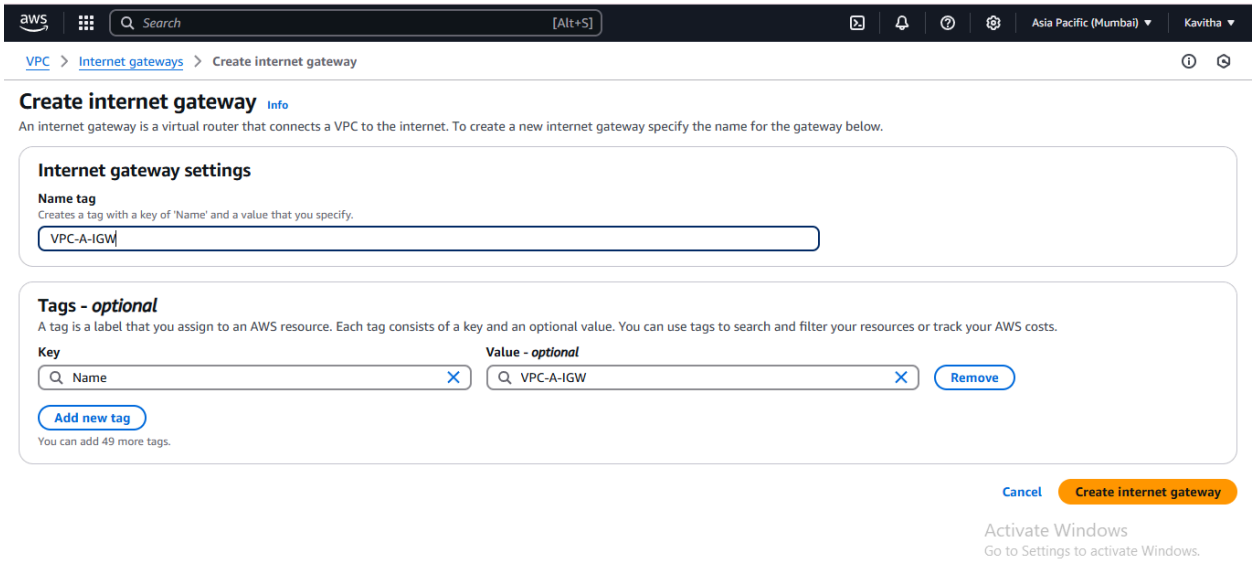
1. Create a VPC:

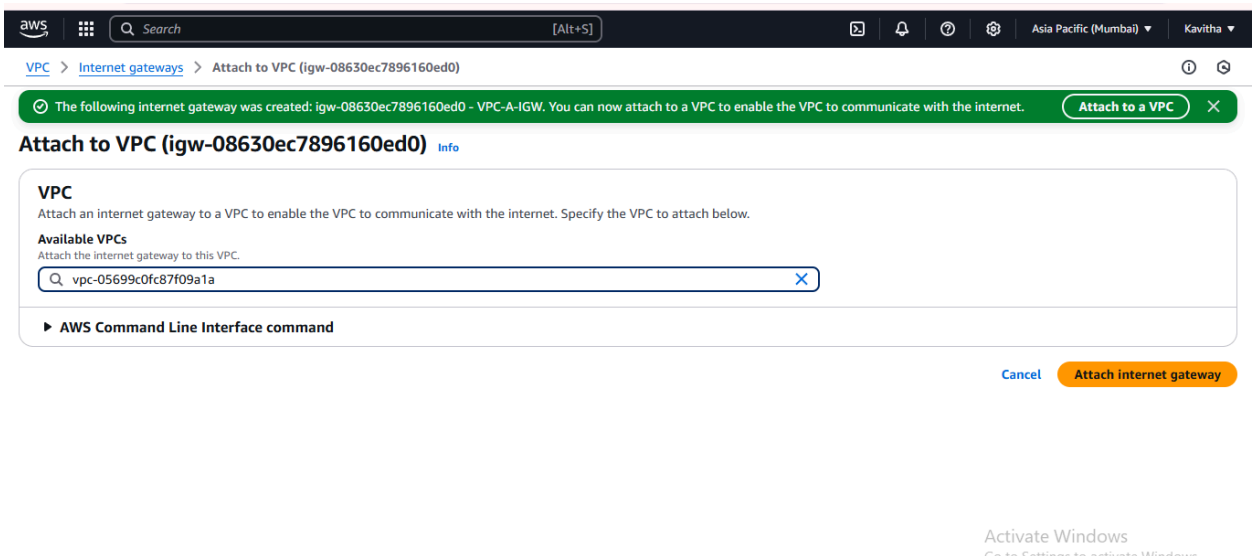
- Navigate to the **VPC Dashboard** in the AWS Management Console.
- Click on **"Create VPC "**.
- Provide a **Name** and specify an **IPv4 CIDR block** (e.g., `10.100.0.0/16`).
- Choose **"Create"** to establish the VPC A.



2. Create and Attach an Internet Gateway:

- In the VPC Dashboard, select **"Internet Gateways"** and click **"Create Internet Gateway"**.
- Provide a **Name** and choose **"Create"**.
- Select the newly created Internet Gateway, click **"Actions"**, and choose **"Attach to VPC"**.
- Select VPC A and confirm the attachment

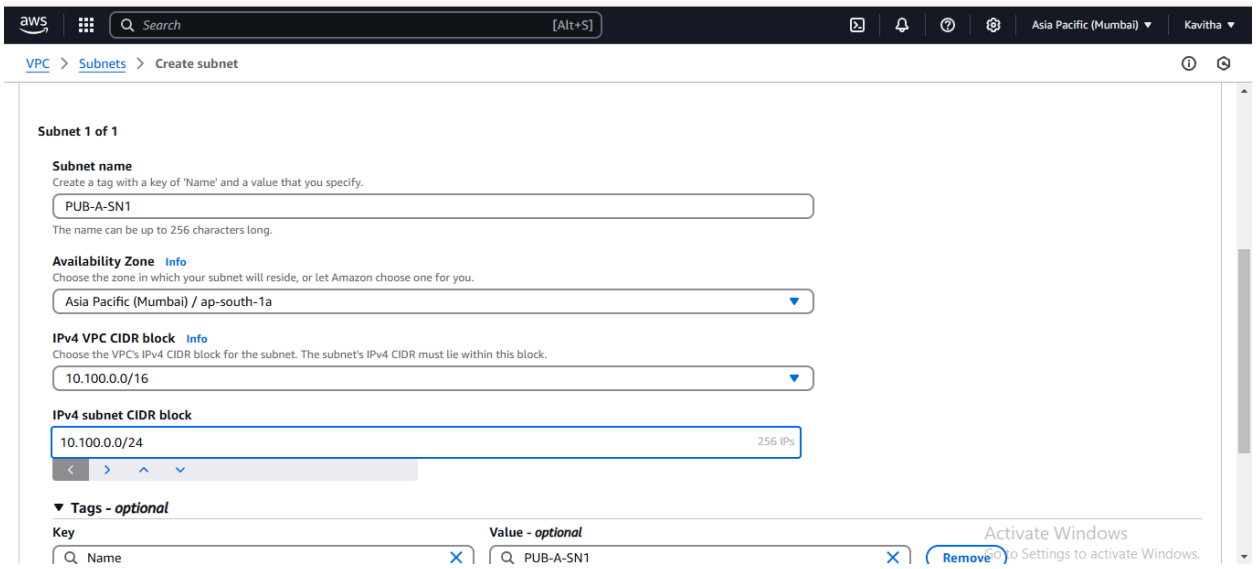


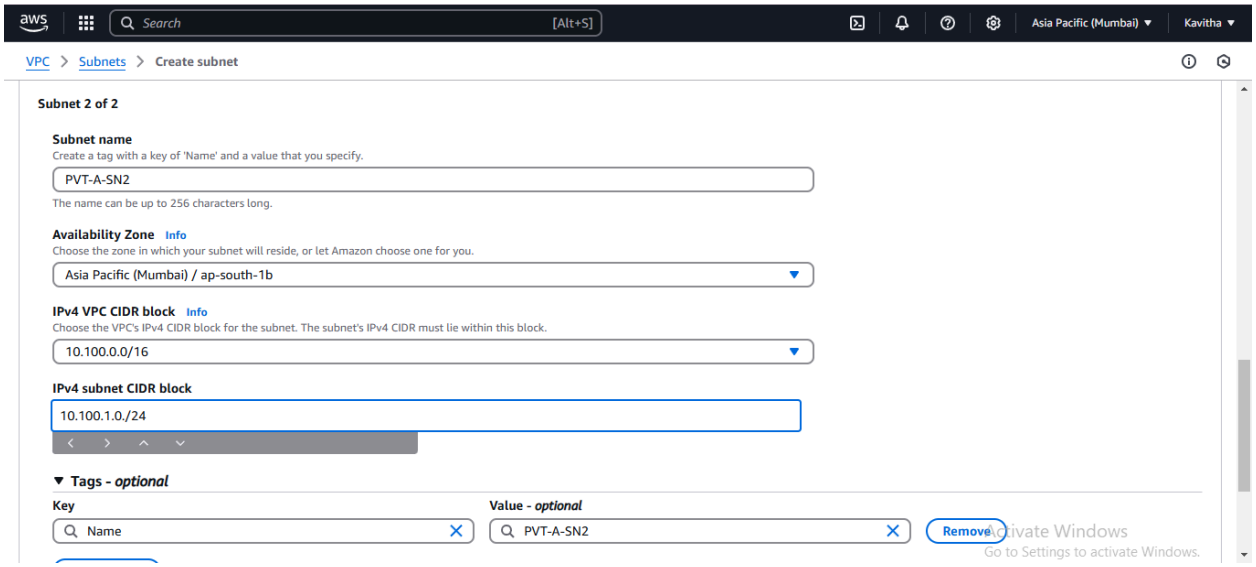


3. Create Subnets:

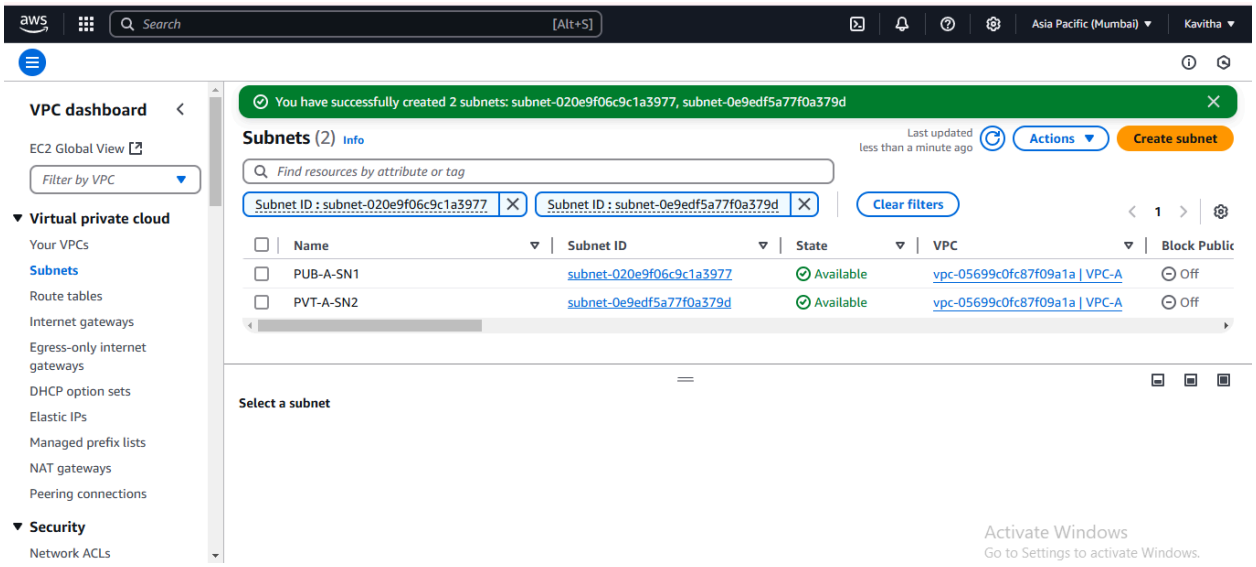
- **Public Subnet A:**
  - In the VPC Dashboard, select **"Subnets"** and click **"Create Subnet"**.
  - Assign a **Name**, select the **VPC A** created earlier, and specify an **IPv4 CIDR block** (e.g., `10.100.0.0/24`).
  - Designate an **Availability Zone** as 1a.
  - Ensure **Auto-assign public IPv4 address** is **enabled**.

- **Private Subnet A:**
  - Repeat the steps above to create another subnet.
  - Assign a **Name**, select the **same VPC A**, and specify a different **IPv4 CIDR block** (e.g., `10.100.1.0/24`).
  - Ensure **Auto-assign public IPv4 address** is **disabled**.





Subnets created successfully



4. Configure Route Tables:

- **Public Route Table:**
  - In the VPC Dashboard, select **"Route Tables"** and click **"Create Route Table"**.
  - Assign a **Name**, select **VPC A**, and choose **"Create"**.
  - With the new route table selected, navigate to the **"Routes"** tab and click **"Edit routes"**.
  - Add a route with **Destination** `0.0.0.0/0` and
  - Save the changes.
  - Navigate to the **"Subnet Associations"** tab, click **"Edit subnet associations"**, and associate the **public subnet**.

- **Private Route Table:**
  - Repeat the steps to create another route table for the private subnet.
  - No routes need to be added at this stage.
  - Associate this route table with the **private subnet**.

aws | Search [Alt+S] | Asia Pacific (Mumbai) | Kavitha

VPC > Route tables > Create route table

Create route table Info

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
PUB-A-RT

VPC
The VPC to use for this route table.
vpc-05699c0fc87f09a1a (VPC-A)

Tags

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - *optional*	
Name	PUB-A-RT	Remove

Add new tag
You can add 49 more tags.

aws | Search [Alt+S] | Asia Pacific (Mumbai) | Kavitha

VPC > Route tables > rtb-0a403f8d35fcd8a31 > Edit routes

Edit routes

Destination	Target	Status	Propagated	
10.100.0.0/16	local	Active	No	
	local			
0.0.0.0/0	Internet Gateway	–	No	Remove
	igw-08630ec7896160ed0			

Add route

Cancel | Preview | Save changes

aws | Search [Alt+S] | Asia Pacific (Mumbai) | Kavitha

VPC > Route tables > rtb-0a403f8d35fcd8a31 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)

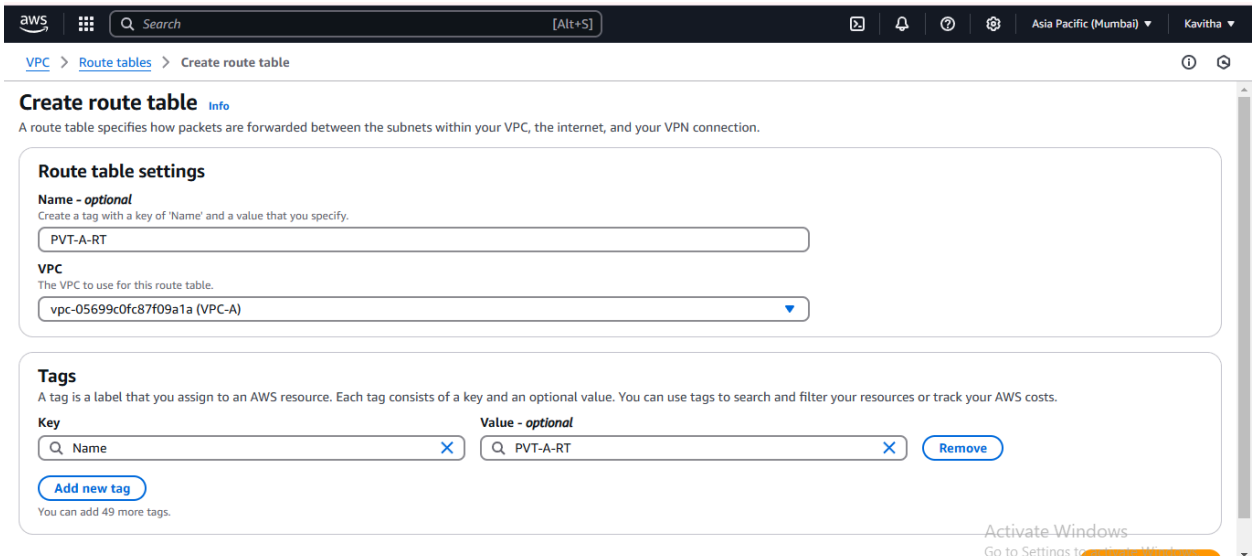
Filter subnet associations

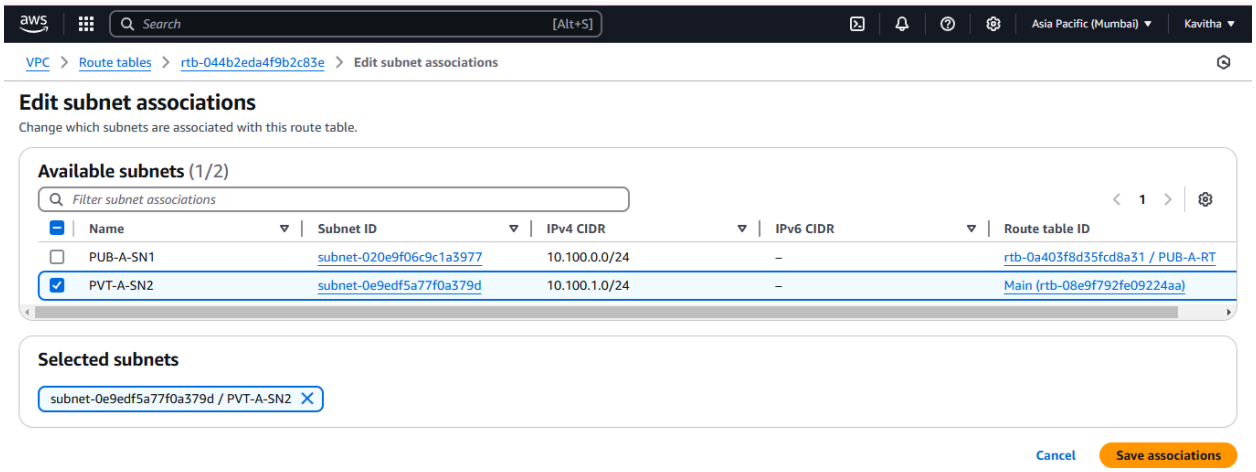
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
PUB-A-SN1	subnet-020e9f06c9c1a3977	10.100.0.0/24	–	Main (rtb-08e9f792fe09224aa)
PVT-A-SN2	subnet-0e9edf5a77f0a379d	10.100.1.0/24	–	Main (rtb-08e9f792fe09224aa)

Selected subnets

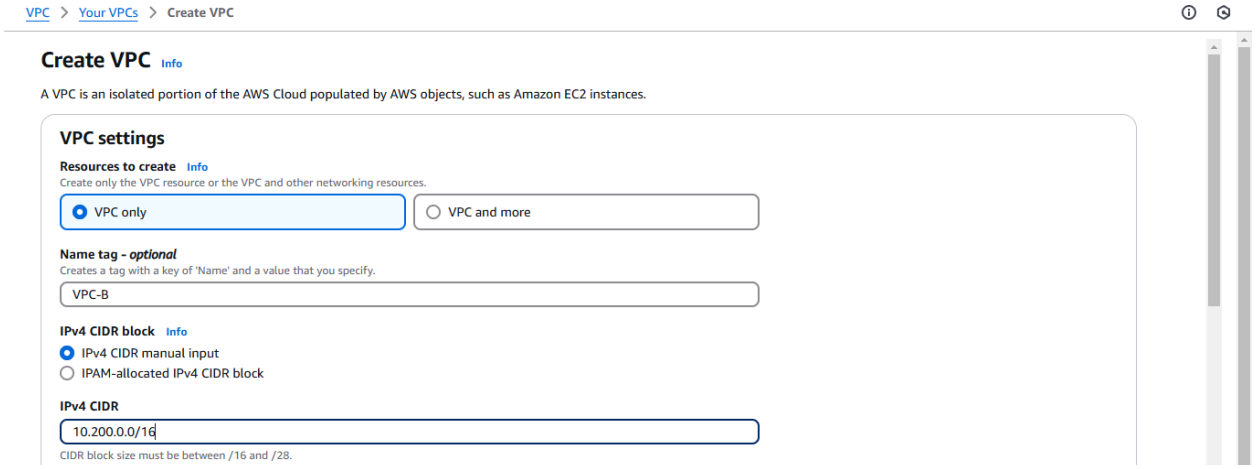
subnet-020e9f06c9c1a3977 / PUB-A-SN1

Cancel | Save associations





Same configuration is repeated for VPC B and creating Private VPC B one private subnet and Route table



aws | Search [Alt+S] | Asia Pacific (Mumbai) | Kavitha

VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.

vpc-099d6796d5c7e9b35 (VPC-B)

Associated VPC CIDRs

IPv4 CIDRs
10.200.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

aws | Search [Alt+S] | Asia Pacific (Mumbai) | Kavitha

VPC > Subnets > Create subnet

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

PVT-B-SN

The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

Asia Pacific (Mumbai) / ap-south-1b

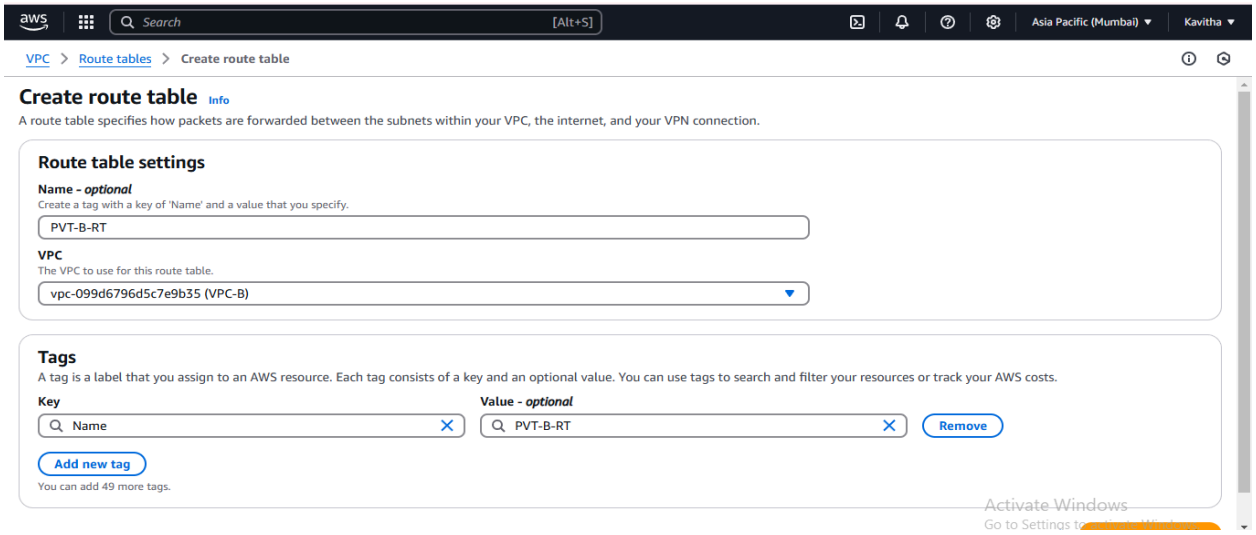
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

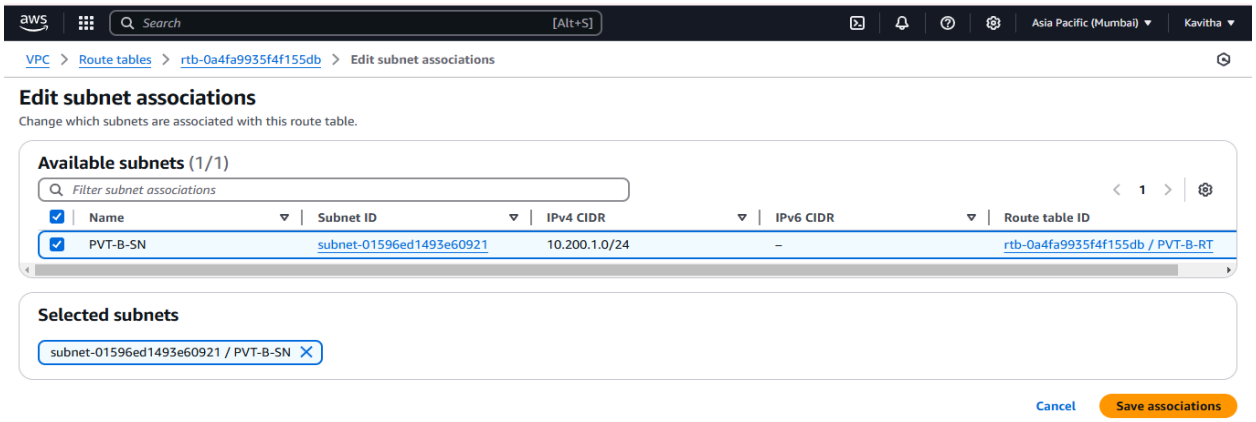
10.200.0.0/16

IPv4 subnet CIDR block

10.200.1.0/24 256 IPs

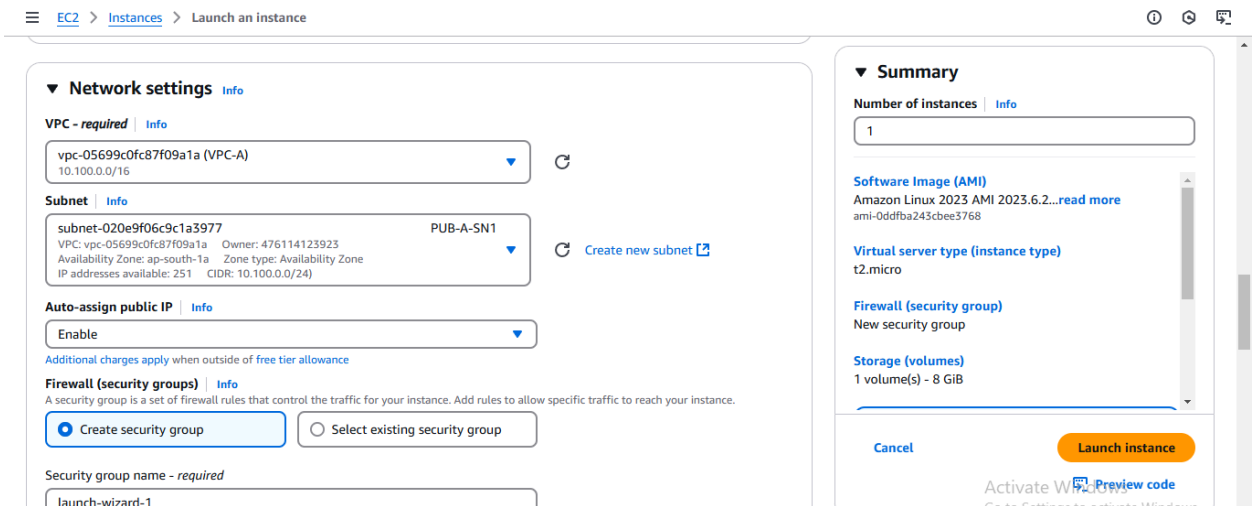
Activate Windows
Go to Settings to activate Windows.



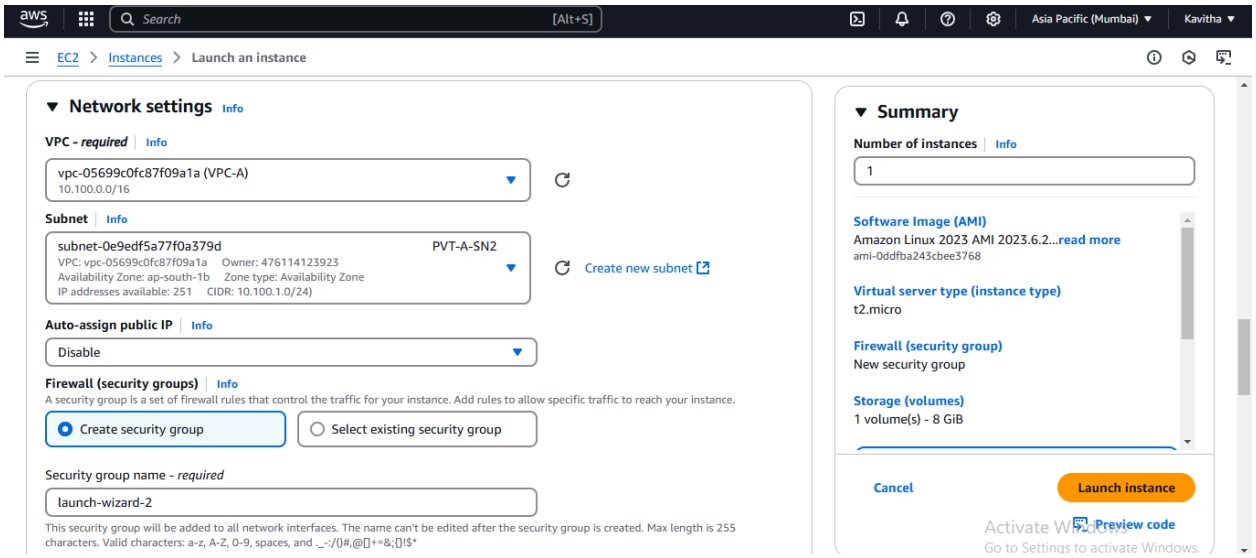


Creating 3 EC2 Instance - Public and private for VPC A & Private for VPC B

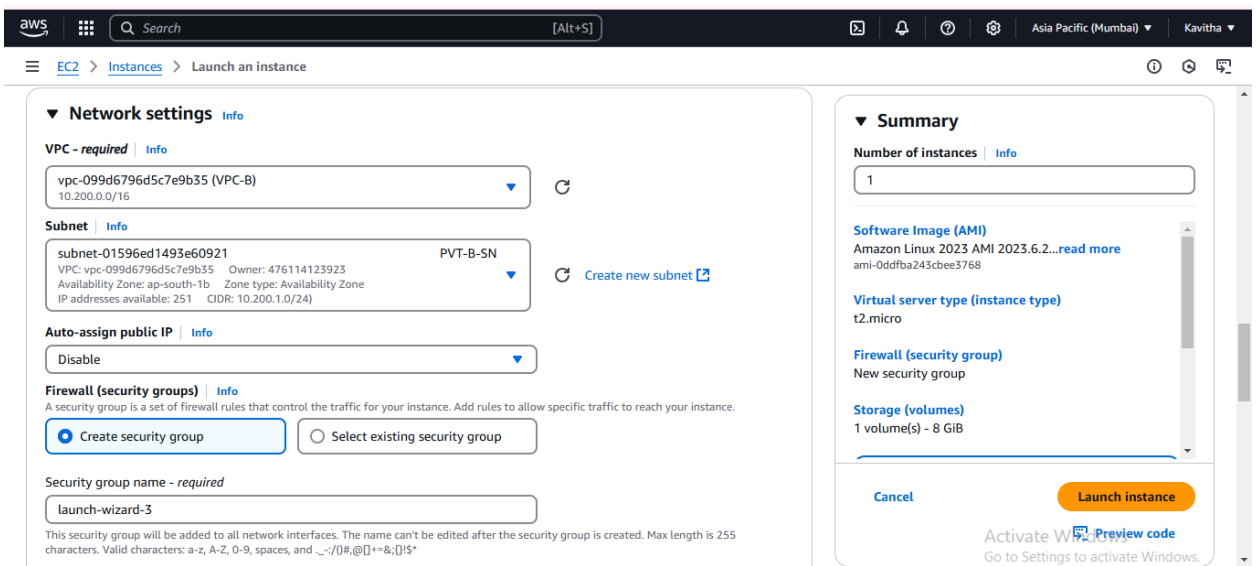
Launch Instance >EC2 A edit the Network settings choose VPC A > PUB-SN



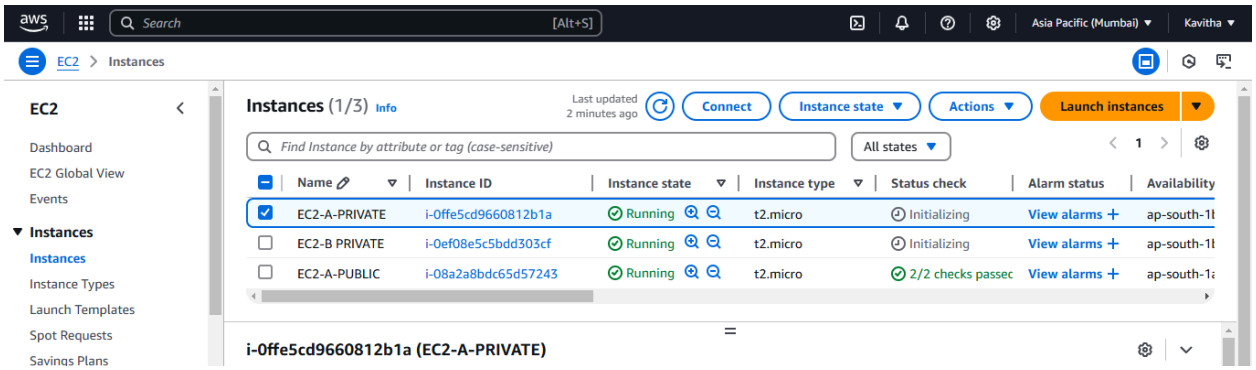
For EC2 Private Instance A In network setting choose VPC A > Private SN



For EC2 Instance B - choose VPC B and private subnet in network setting



EC2 Instance created successfully



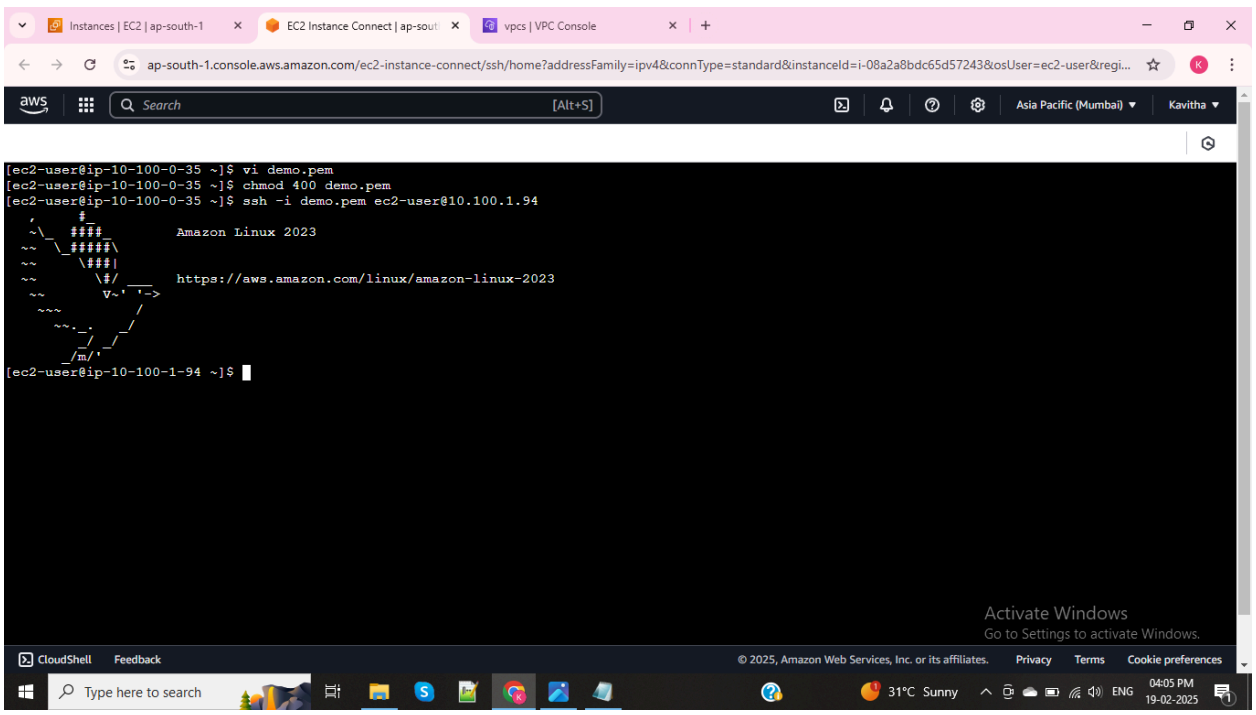
To enable secure communication between instances in a EC2 A Public to EC2 A private:

Launch an **EC2 instance in the public subnet of VPC A.**

Connect the **Public Instance** --> **Update the server** --> Create a **.pem file** and Key was pasted in that file and save.

Change permission for the key --> **chmod 400 demo.pem**

To connect private Instance give --> **ssh -i demo.pem ec2-user@(private Ip of private EC2 Instance A)**



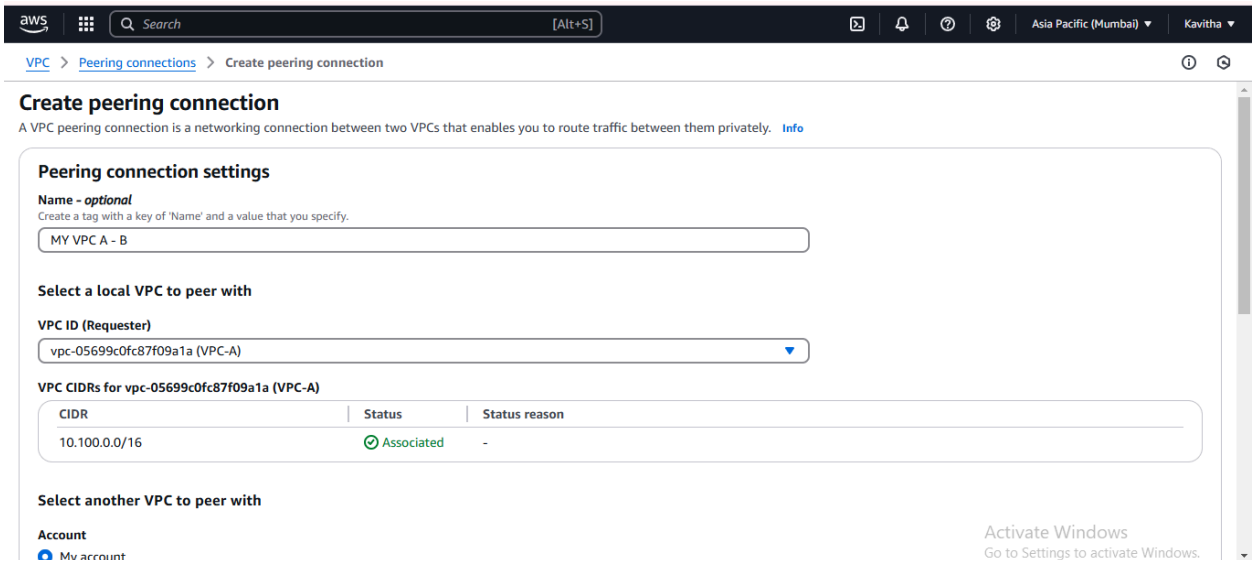
Now, Communication establish between instances in two Virtual Private Clouds (VPCs) using VPC Peering in AWS

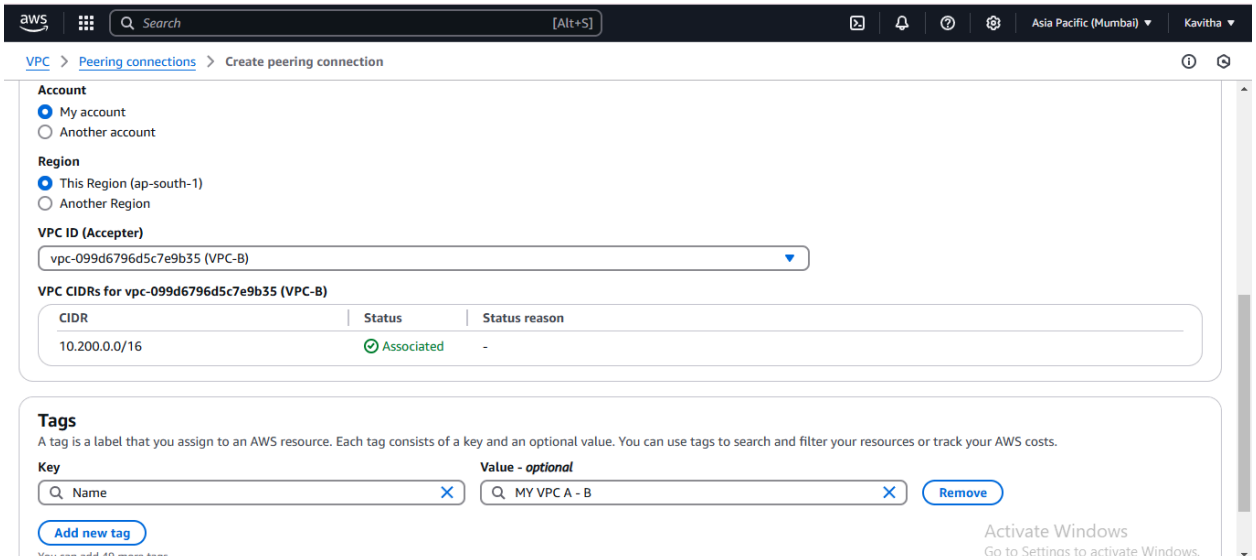
1. Establish a VPC Peering Connection:

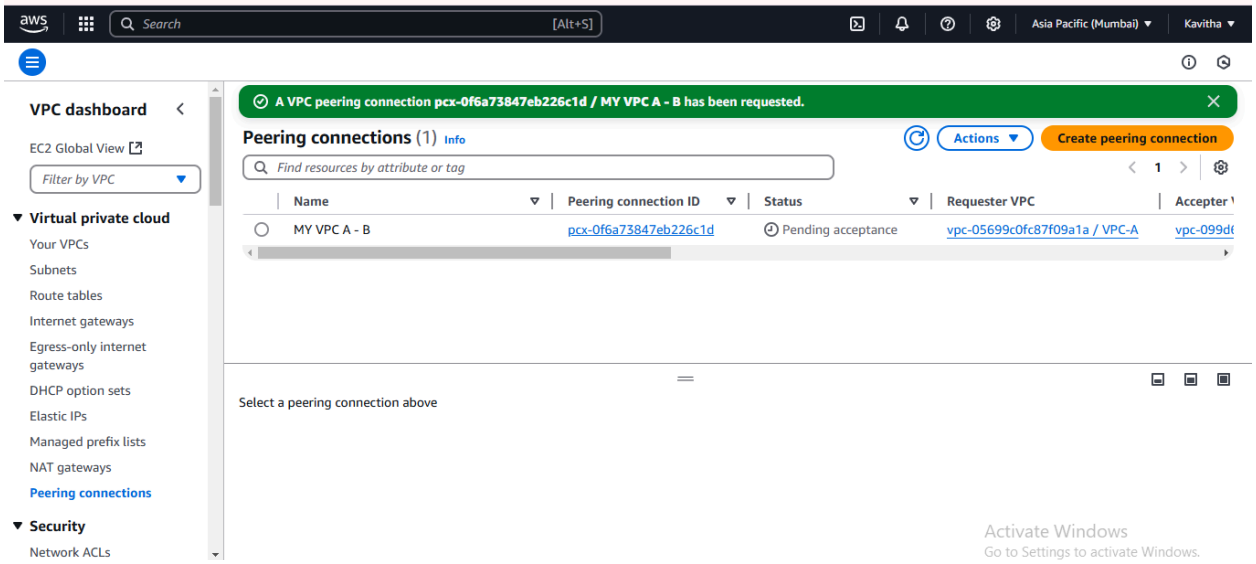
- **Initiate the Peering Request:**
  - In the AWS Management Console, navigate to the **VPC Dashboard**.
  - Select **"Peering Connections"** and click **"Create Peering Connection"**.

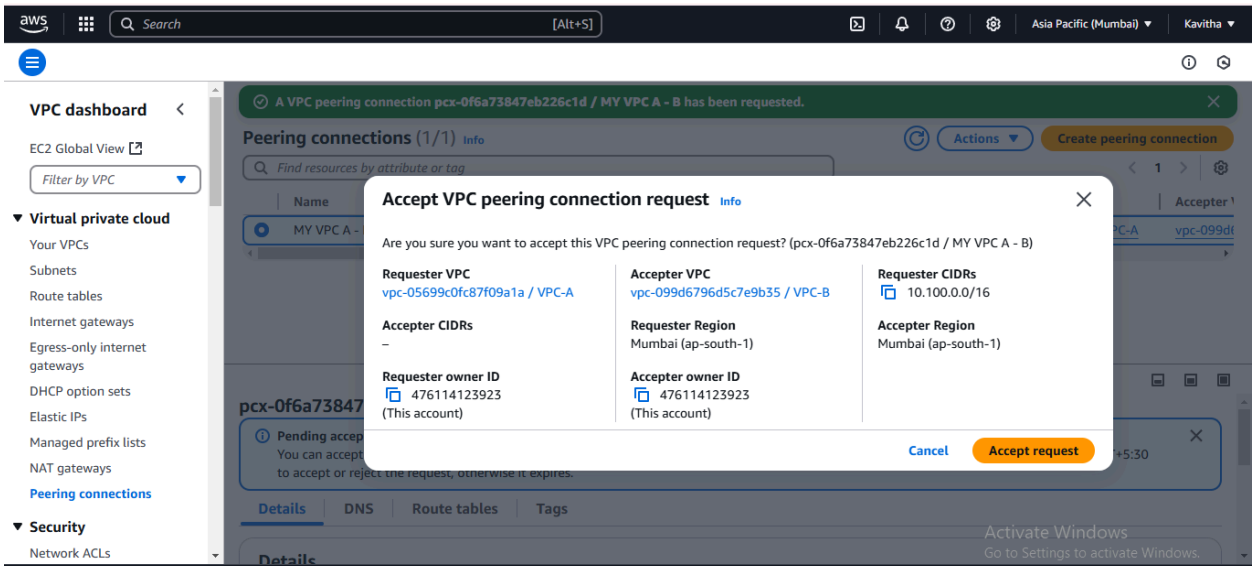
- Specify the **Requester VPC** and the **Accepter VPC**. The VPCs can be within the same AWS account or across different accounts and regions.
- Provide a **My VPC A TO B** (**Name)** for the peering connection.

- **Accept the Peering Request:**
  - The owner of the Accepter VPC must navigate to **"Peering Connections"** in their VPC Dashboard.
  - Locate the pending peering request and choose **"Actions"** > **"Accept Request"**.



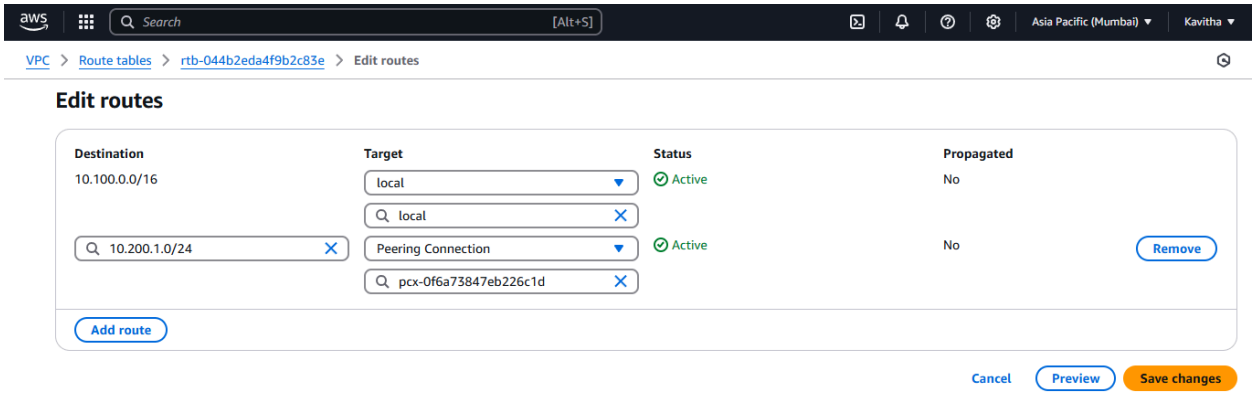


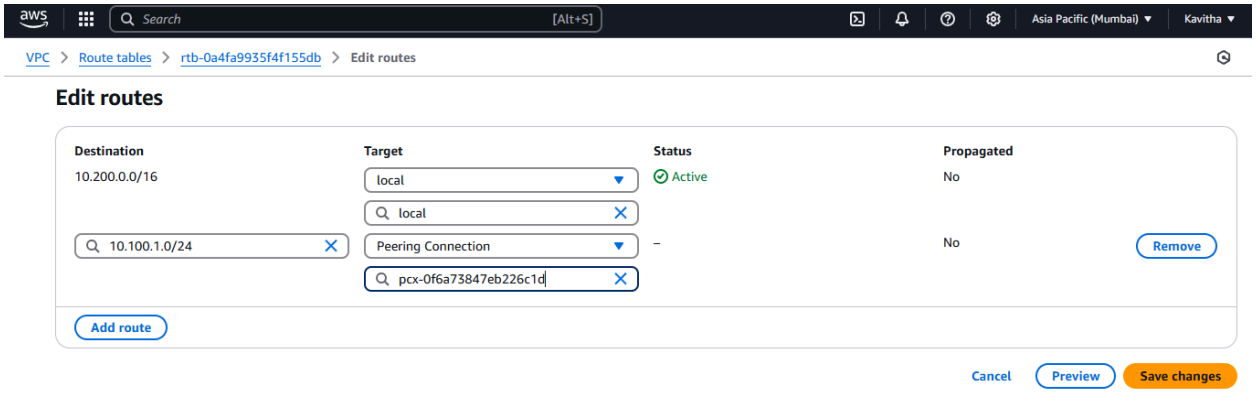




2. Update Route Tables:

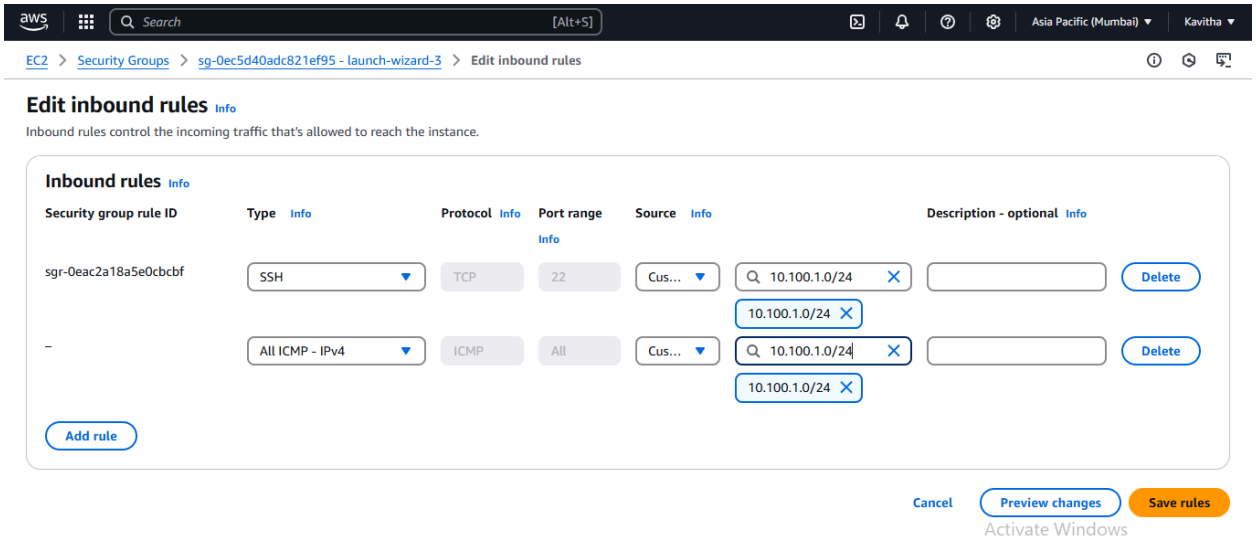
- **Modify Route Tables in Both VPCs:**
  - In each VPC, navigate to **"Route Tables"** in the VPC Dashboard.
  - Select the route table associated with the subnets containing your instances.
  - Click **"Edit Routes"** and add a new route:
  - **Destination:** The CIDR block of the peered VPC.
  - **Target:** The VPC Peering Connection ID.
  - Save the changes to enable routing between the VPCs.





3. Configure Security Groups:

- **Adjust Security Group Rules in EC2 B Private :**
  - For each instance that needs to communicate across the VPCs, modify its security group:
  - **Inbound Rules:** Add rules allowing traffic from the CIDR block of the peered VPC or specific IP addresses as needed.
  - This configuration ensures that only desired traffic is allowed between instances in the peered VPCs.



5. Testing the Connection:

- **Verify Connectivity:**
  - SSH into an instance in the Requester VPC.
  - Attempt to ping or SSH into the private IP address of an instance in the Accepter VPC.
  - Ensure that security group rules and network ACLs allow the necessary traffic for these tests.

Again create .pem file > paste the key & save > chmod 400 demo.pem >

ssh -i demo.pem ec2-user@(EC2 B Pvt IP)