Project Title : Establishing VPC Peering Between Two VPCs in AWS

---

1. Introduction

This project outlines the process of establishing a VPC peering connection between two Virtual Private Clouds (VPCs) in AWS, enabling network traffic between instances in different VPCs.

---

2. Objectives

- Create two VPCs with distinct CIDR blocks.
- Establish a VPC peering connection between them.
- Configure route tables to allow communication.
- Verify connectivity between instances in the peered VPCs.

---

3. Prerequisites

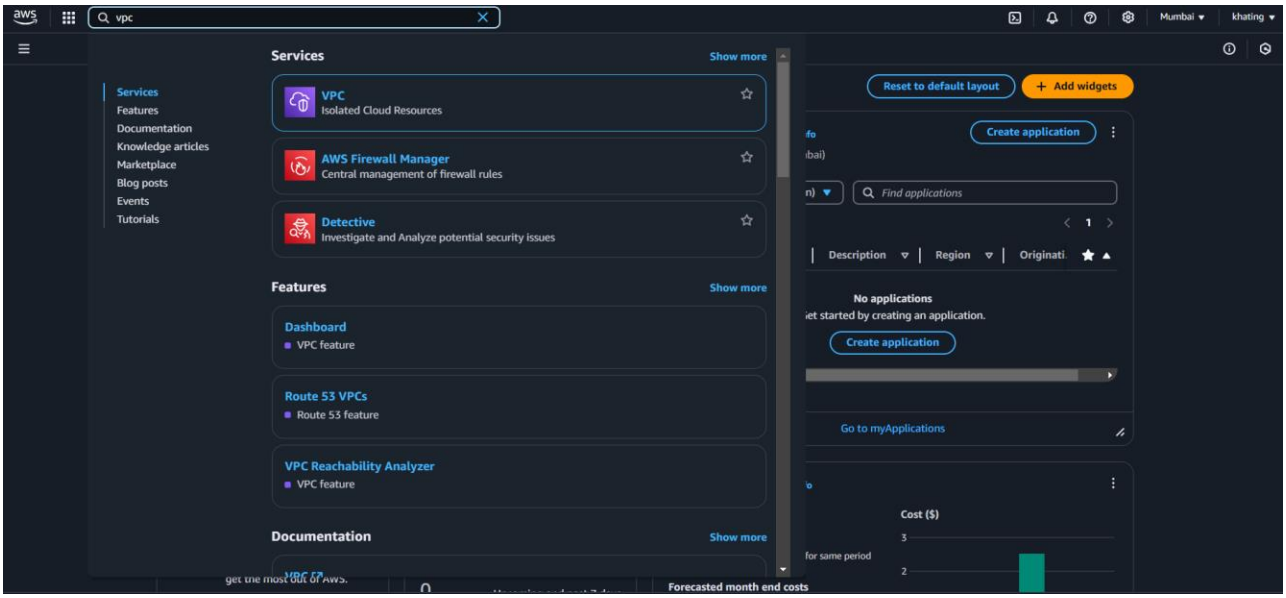
- An AWS account with necessary permissions.
- Basic understanding of AWS networking concepts.
- Access to the AWS Management Console.

---

4. Steps to Implement

Step 1: Create VPCs

Navigate to the VPC Dashboard : In the AWS Management Console, go to the VPC service.



Create the First VPC : Click on "Create VPC".

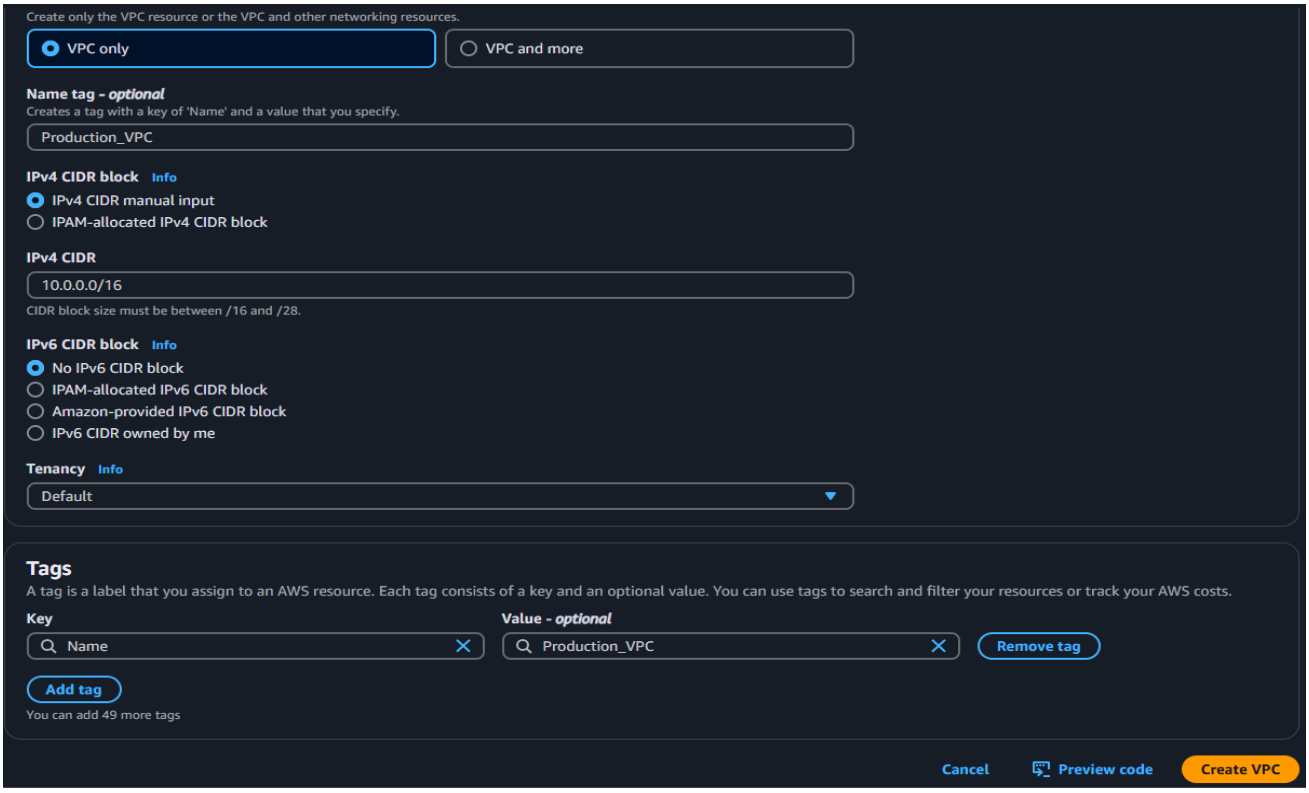
Set the following details :

- Name tag: Production_VPC
- IPv4 CIDR block: 10.0.0.0/16
- Leave other settings as default and click "Create VPC".

Create the Second VPC Repeat the above steps with:

Name tag: Operations_VPC

IPv4 CIDR block: 192.168.0.0/16



Step 2: Create Subnets

1. **Add a Subnet to Production_VPC**

   Select PRODUCTION_VPC and click on "Create Subnet".

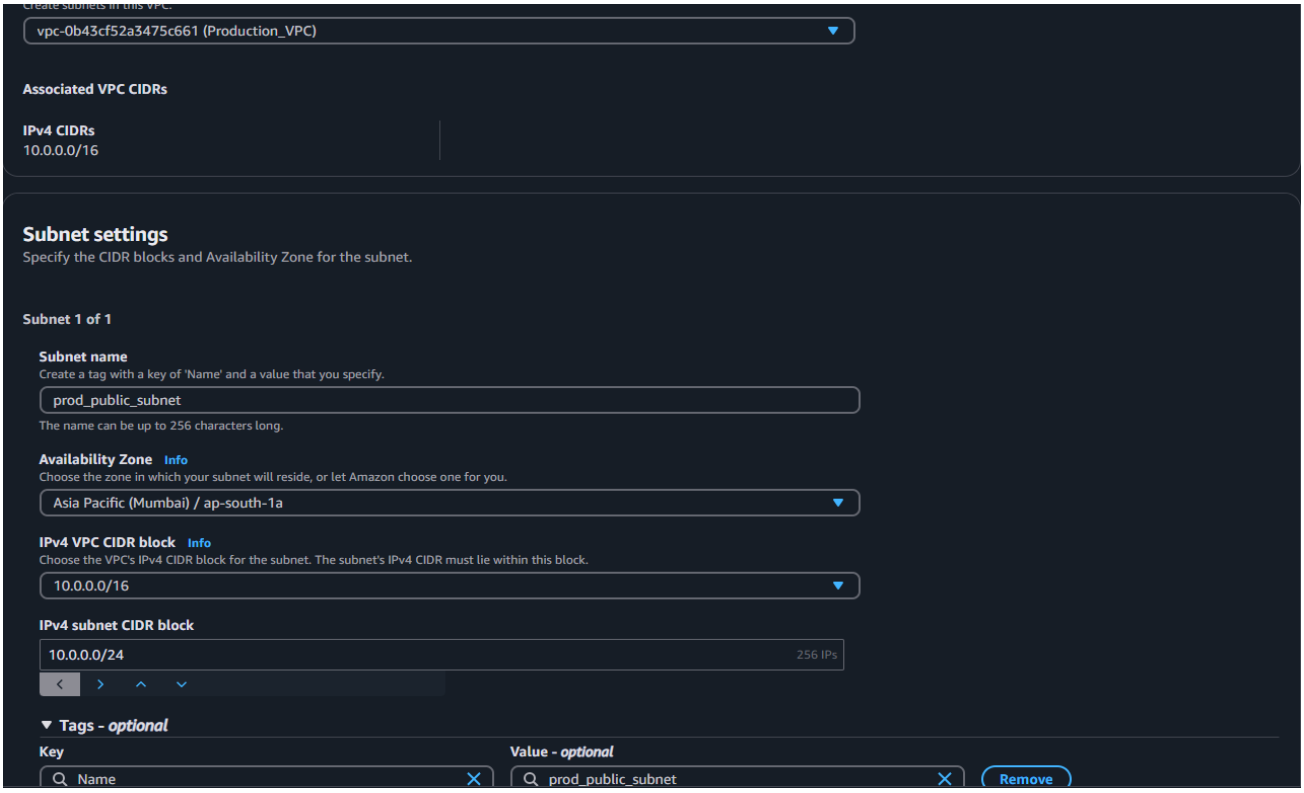
   Configure:

   - Subnet name: prod_public_subnet
   - Availability Zone: Choose one (e.g., ap-south-1a).
   - IPv4 CIDR block: 10.0.0.0/24

   Click "Create Subnet".

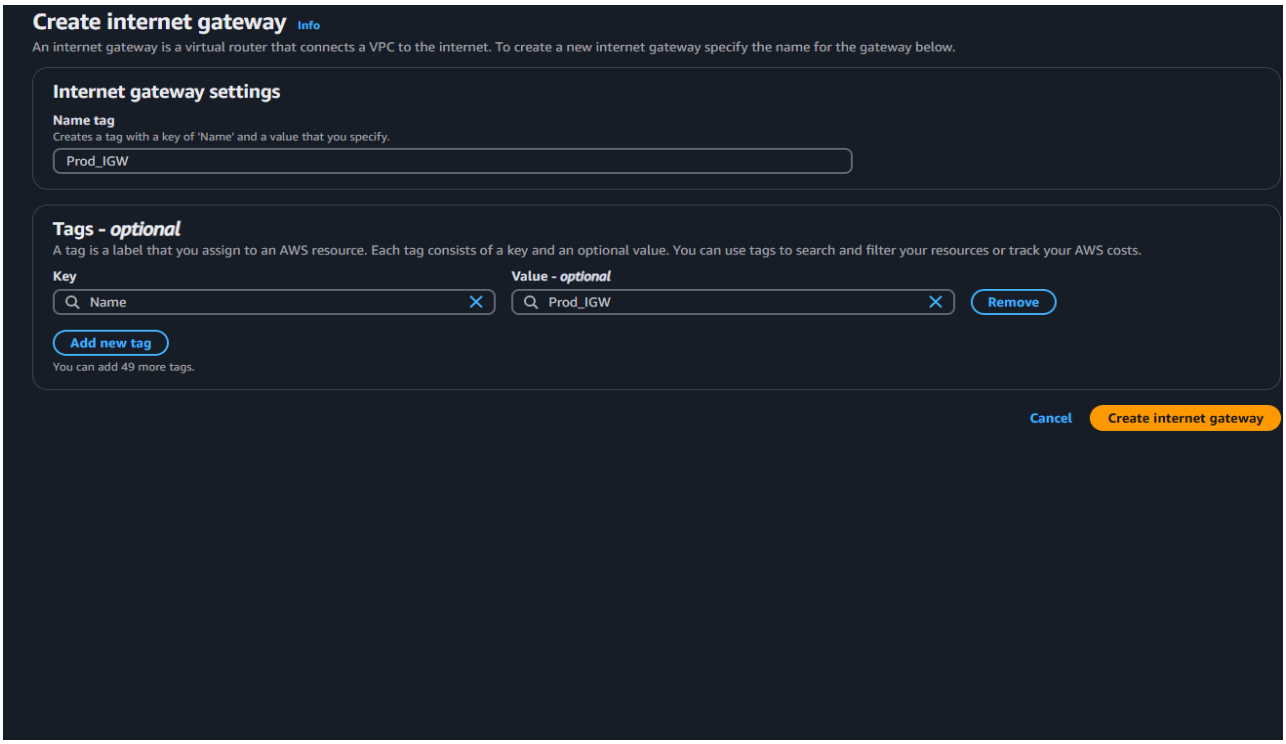
2. **Add a Subnet to OPERATIONS_VPC**

   - Repeat the above steps for OPERATIONS_VPC with:
     - Subnet name: Ops_public_Subnet
     - Availability Zone: Choose one (e.g., ap-south-1a).
     - IPv4 CIDR block: 192.168.0.0/24



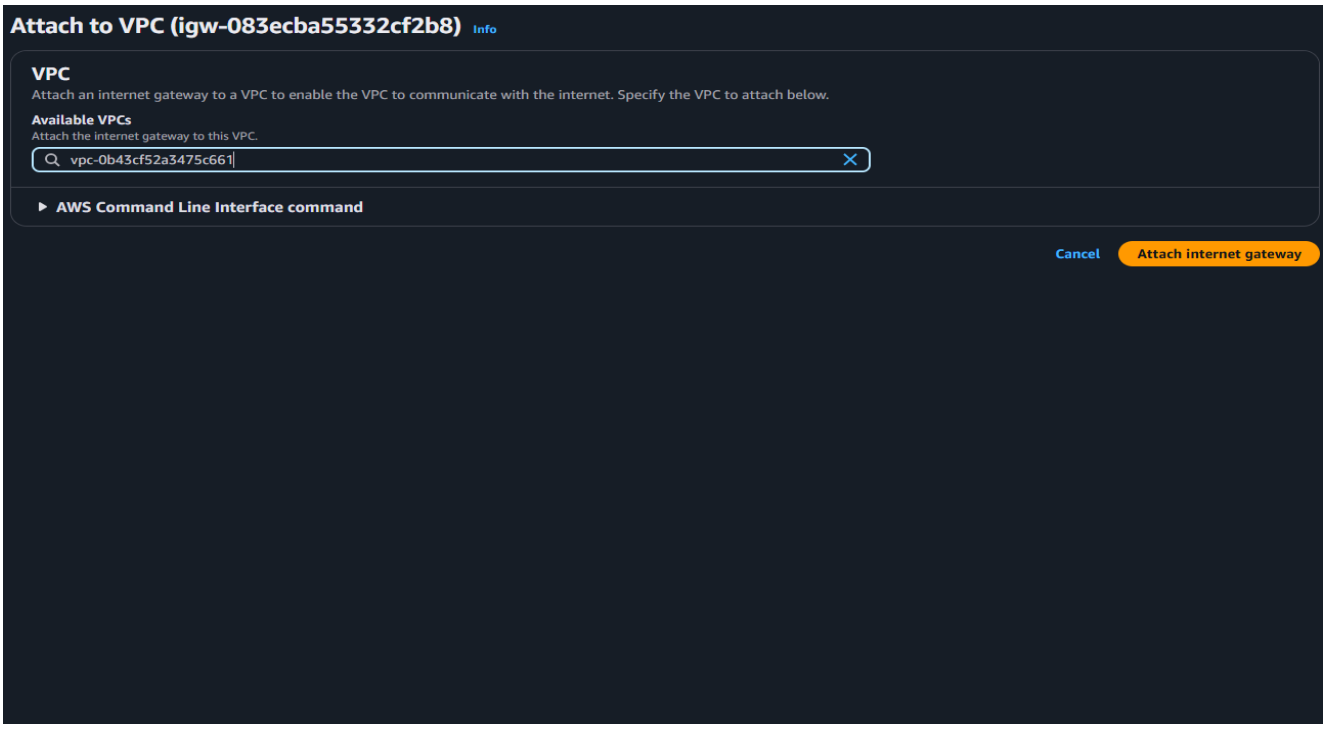
Step 3: Create and Attach Internet Gateway to your VPC

- Click on Create internet gateway.
- Provide a Name tag for easier identification, e.g., Prod_IGW.
- Click Create internet gateway to proceed.



- In the dialog box, select the VPC you want to attach the IGW to.
- Click on Actions, then choose Attach to VPC.

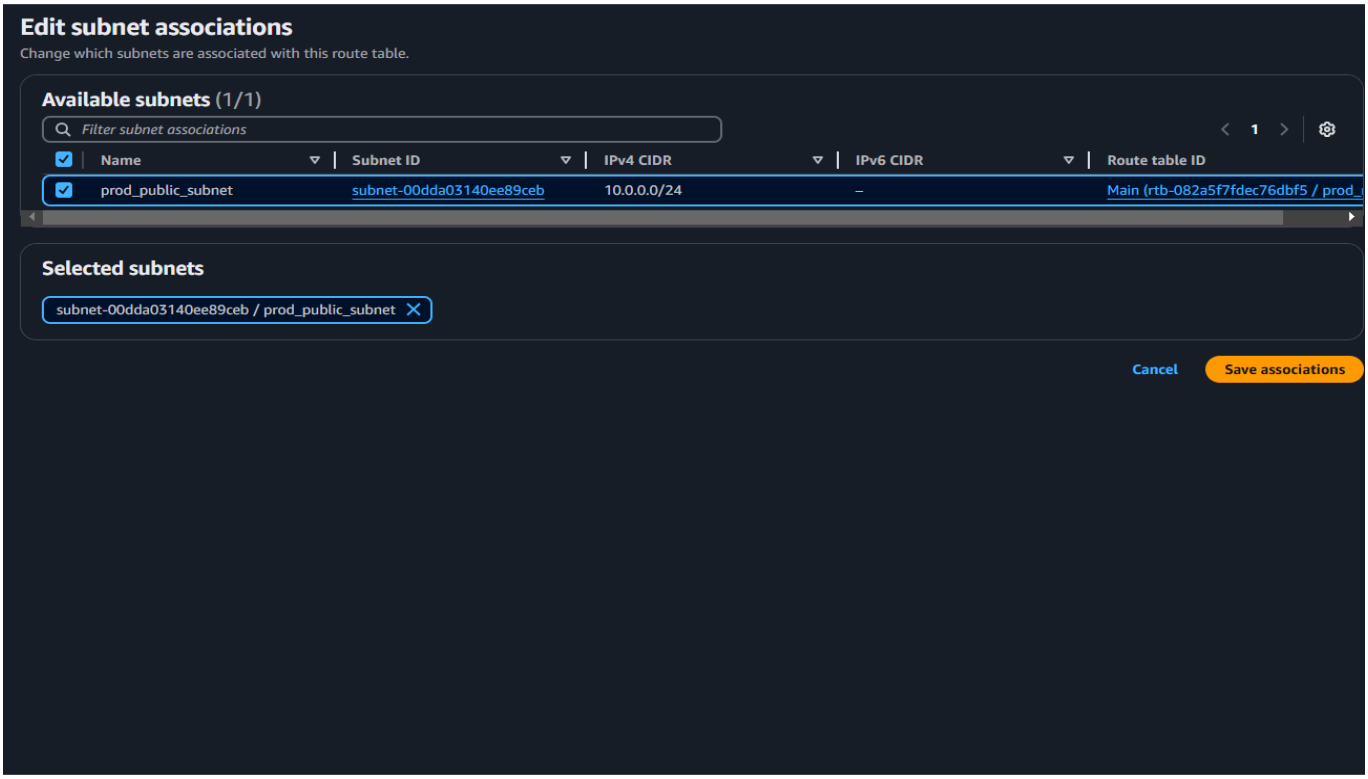
- Click Attach internet gateway to complete the attachment.



Step 4: Associate the Route Table with the Subnet & Edit the Route Table to Use the Internet Gateway

1. Navigate to Subnet Associations:

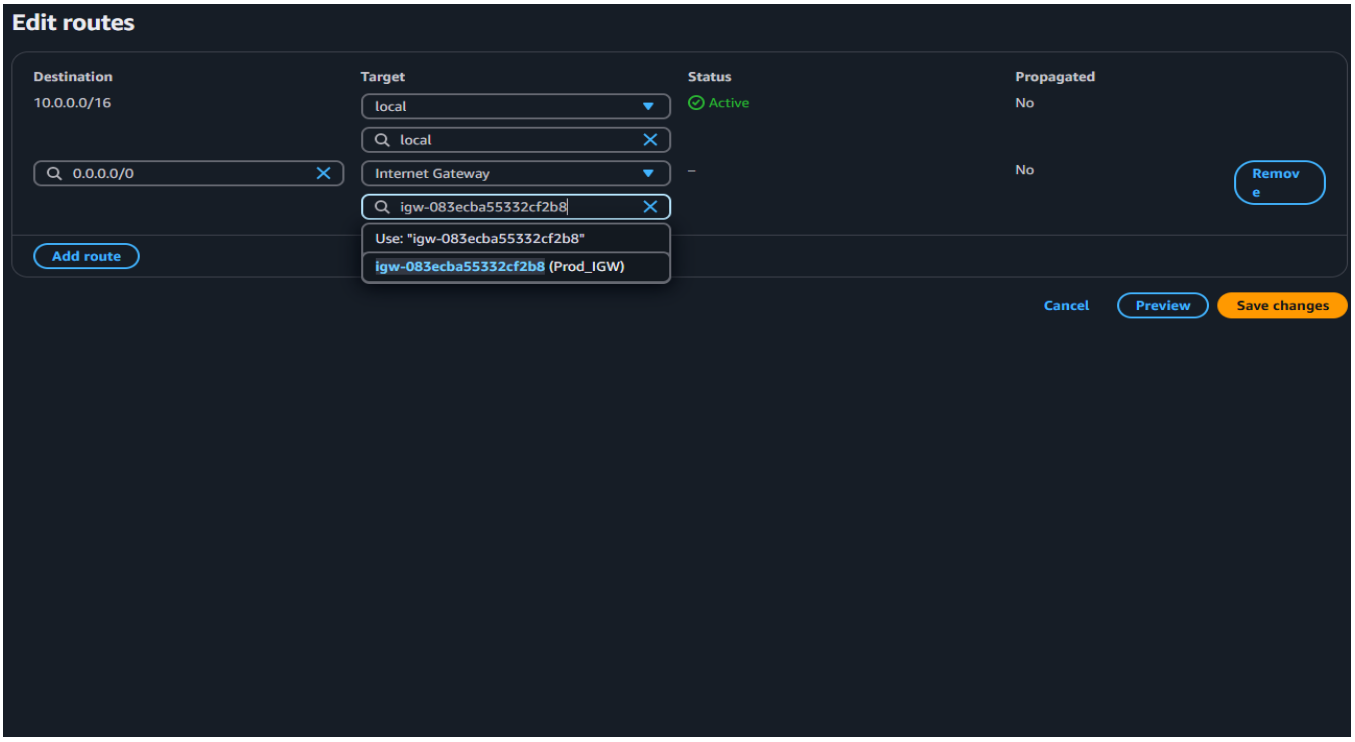
    - In the Route Tables section, select the route table you've just updated.
    - Go to the Subnet associations tab.
    - Click Edit subnet associations.



- Select the subnet(s) you want to associate with this route table.
- Click Save associations to confirm.
-

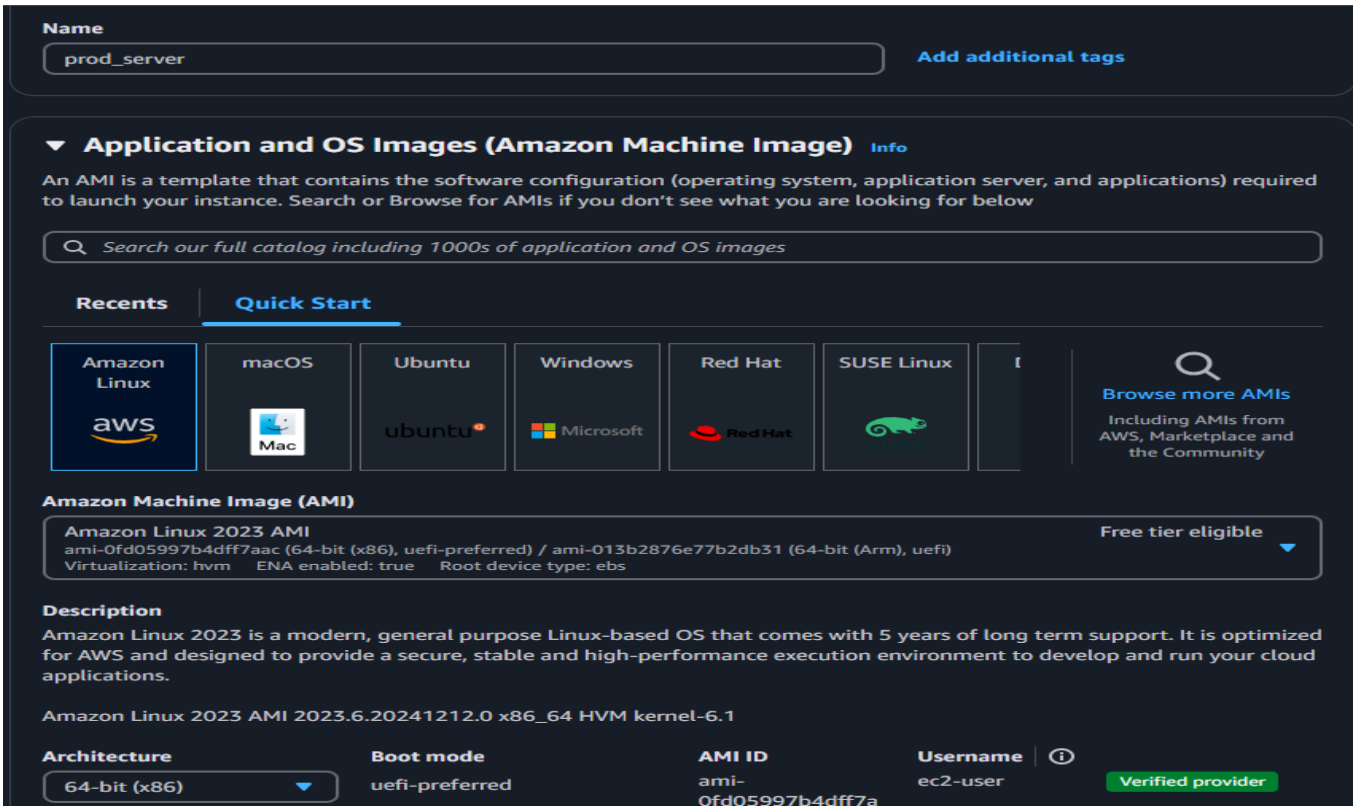
1. **Edit Route Tables to use the internet gateway**

    - In the VPC Dashboard, select Route Tables from the left-hand menu.
    - Identify and select the route table associated with the subnet you wish to provide internet access to.
    - Click Edit routes, then Add route.
    - In the Destination field, enter 0.0.0.0/0 to represent all IPv4 addresses.
    - In the Target field, select the Internet Gateway ID (e.g., igw-xxxxxxxx).
    - Click Save routes to apply the changes

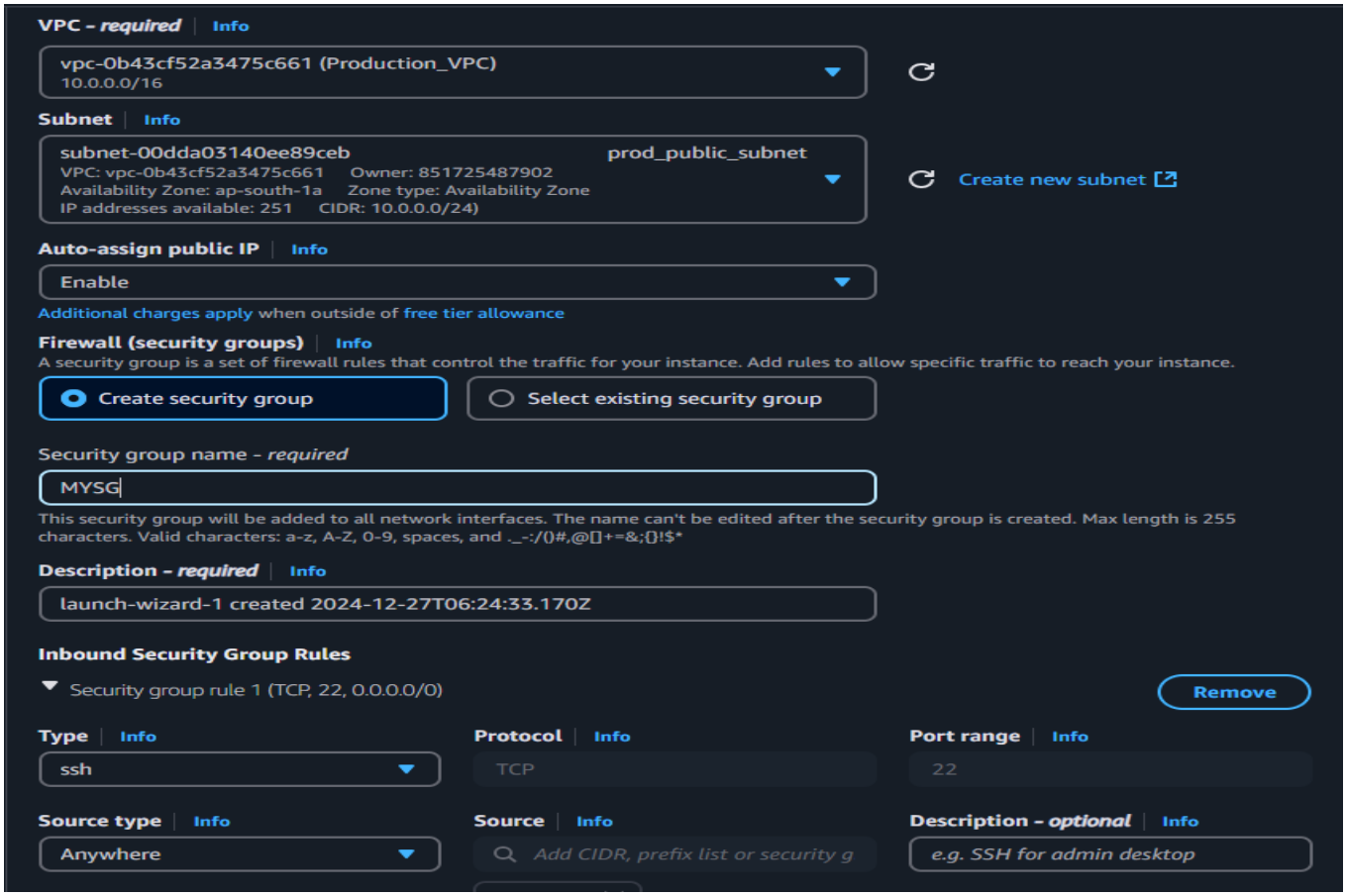


Step 5 : Launch an EC2 Instance in the Public Subnet of Both the VPC.

1. Open the Amazon EC2 Console.
2. Click Launch Instance.
3. Provide an instance name (e.g., My_EC2_Instance).
4. Select an Amazon Machine Image (AMI) (e.g., Amazon Linux 2 AMI)
5. Choose an instance type (e.g., t2.micro).
6. Under Key pair (login), select an existing key pair or create a new one.
7. In Network settings, select your VPC (e.g., Production_VPC) and the public subnet (e.g., prod_public_subnet).

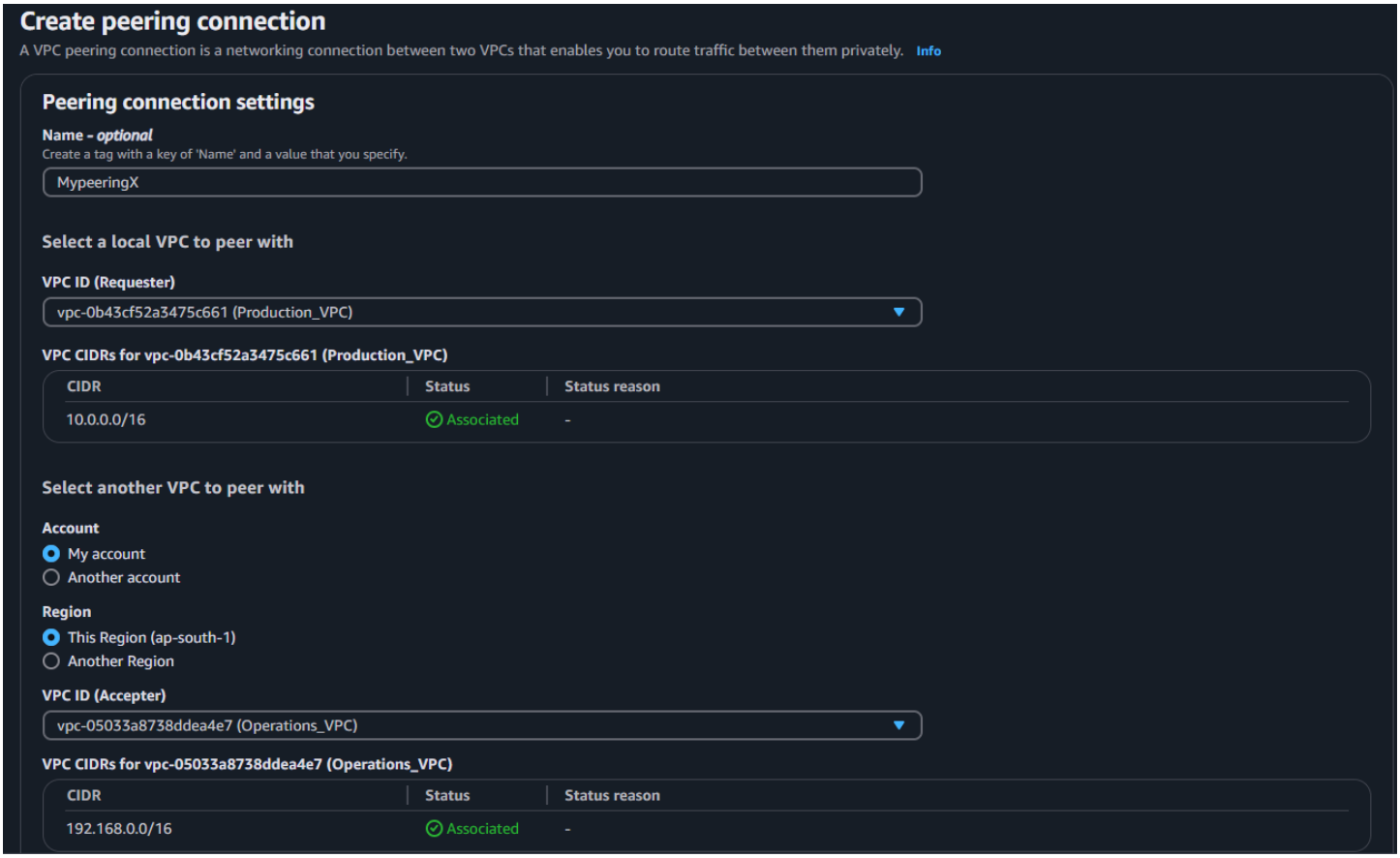


8. Ensure Auto-assign public IP is enabled.
9. Select an existing security group or create a new one with appropriate inbound rules (e.g., allowing SSH access).
10. Click Launch instance.

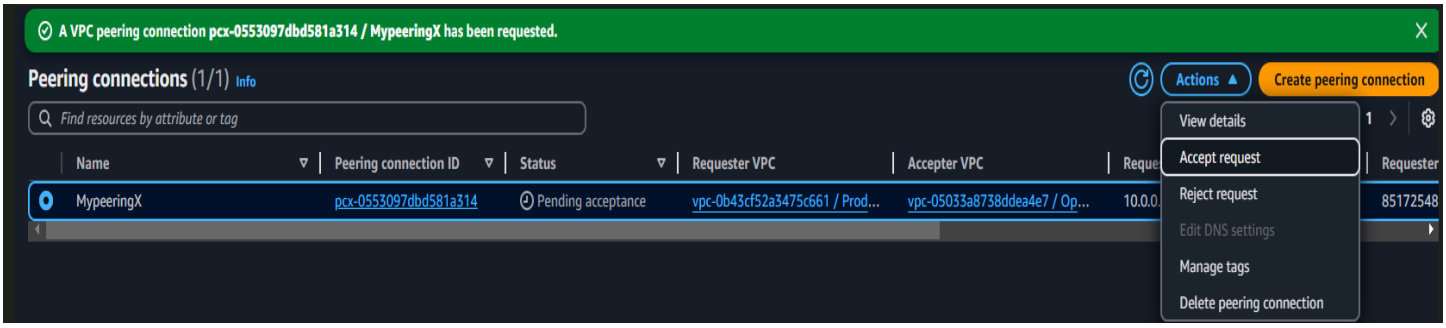


Step 6: Create a VPC Peering Connection

- In the VPC dashboard, select "Peering Connections".
- Click "Create Peering Connection".
- Set:
  - Peering connection name tag: PRODUCTION_VPC-to-OPERATIONS_VPC
  - Requester VPC: PRODUCTION_VPC
  - Accepter VPC: OPERATIONS_VPC
- Click "Create Peering Connection".

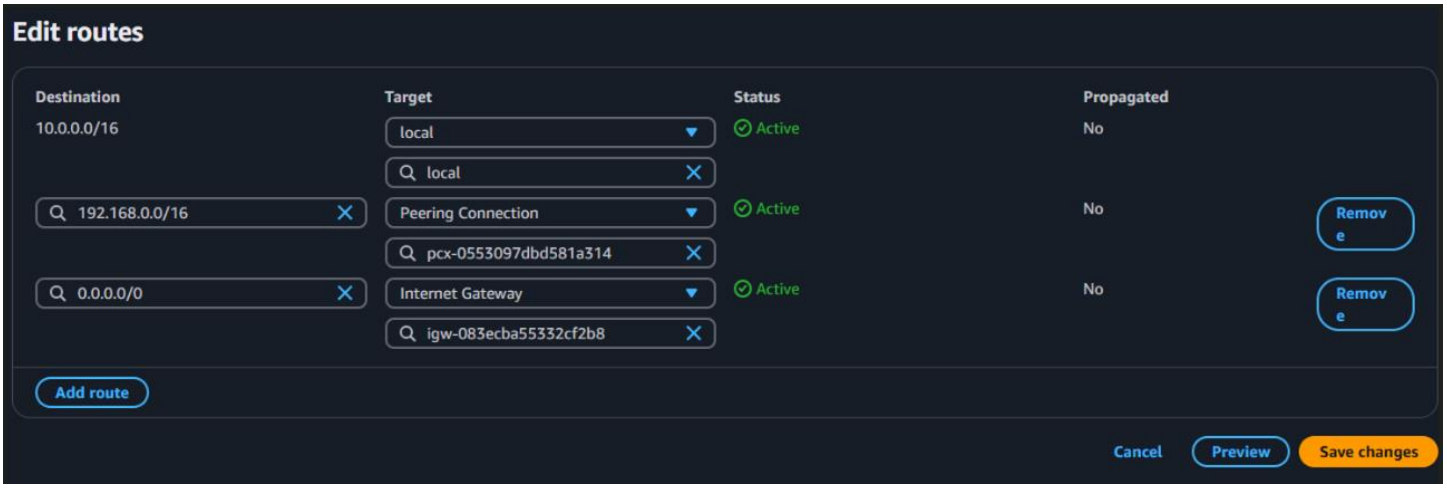


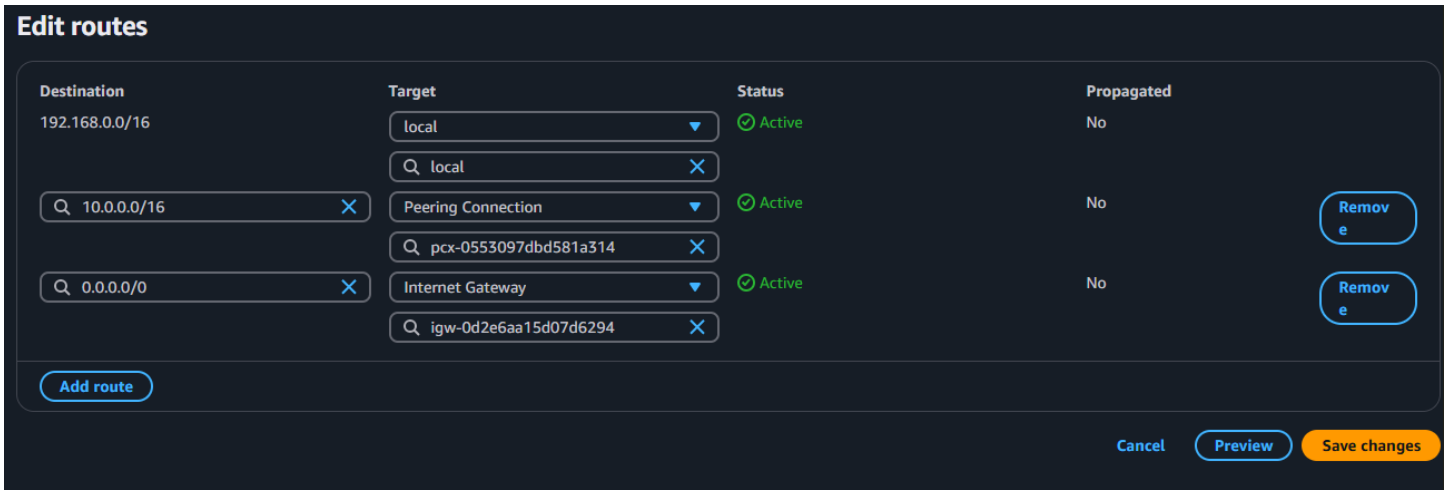
- In "Peering Connections", select the newly created connection.
- Click "Actions" > "Accept Request".
- Confirm acceptance.



Step 7: Update Route Tables

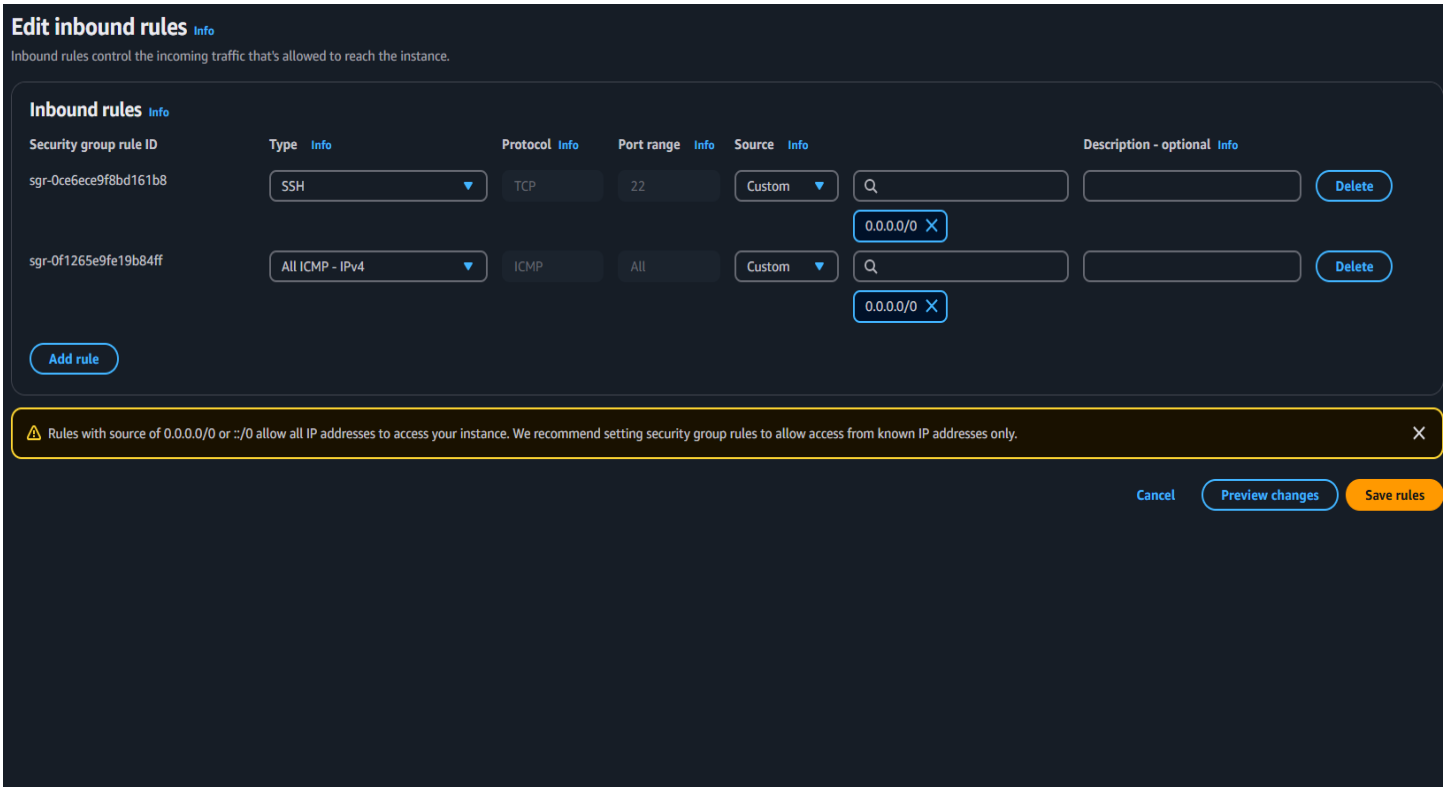
1. **Modify Route Table for PRODUCTION_VPC**
   - Navigate to "Route Tables".
   - Select the route table associated with PRODUCTION_VPC.
   - Click "Edit Routes" and add:
     - Destination: 192.168.0.0/16
     - Target: Select the peering connection PRODUCTION_VPC-to-OPERATIONS_VPC
   - Save changes.
2. **Modify Route Table for OPERATIONS_VPC**
   - Repeat the above steps for OPERATIONS_VPC with:
     - Destination: 10.0.0.0/16
     - Target: Select the peering connection PRODUCTION_VPC-to-OPERATIONS_VPC





Step 8 : Configure Security Groups

1. **Update Security Group for Instances in PRODUCTION_VPC**
    - Ensure the security group allows inbound traffic from SSH via port 22 from Anywhere and also Allows traffic from ICMP(Internet control messaging protocol) from Anywhere.
2. **Update Security Group for Instances in OPERATIONS_VPC**
    - Ensure the security group allows inbound traffic from SSH via port 22 from Anywhere and also Allows traffic from ICMP(Internet control messaging protocol) from Anywhere.



Step 9 : Launch EC2 Instances and Test Connectivity

1. **Test Connectivity**
    - Use ping or SSH to test connectivity between the instances across VPCs.

    SSH into the Instances

2. Open a terminal on your local machine.
3. Connect to the instances using SSH: ssh -i mykey.pem ec2-user@<public ip address>
    -

```
ec2-user@ip-10-0-0-46:~

Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Arun shankar khating>cd downloads

C:\Users\Arun shankar khating\Downloads>ssh -i mykey.pem ec2-user@
52.66.52.173
The authenticity of host '52.66.52.173 (52.66.52.173)' can't be es
tablished.
ED25519 key fingerprint is SHA256:/iDHY1ANhnEpBGlORIoM8SJluVV/xPRh
1EnMMDdX8uo.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint]
)? yes
Warning: Permanently added '52.66.52.173' (ED25519) to the list of
 known hosts.
       #_
       ~\_  ####_        Amazon Linux 2023
      ~~  \_#####\
      ~~     \###|
      ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-202
3
       ~~       V~' '->
        ~~~         /
          ~~._.   _/
             _/ _/
           _/m/'
[ec2-user@ip-10-0-0-46 ~]$
```

```
ec2-user@ip-10-0-0-46:  ec2-user@ip-192-168-0-

Microsoft Windows [Version 10.0.22631.4602]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Arun shankar khating>cd downloads

C:\Users\Arun shankar khating\Downloads>ssh -i mykey.pem ec2-user@
13.201.123.214
The authenticity of host '13.201.123.214 (13.201.123.214)' can't b
e established.
ED25519 key fingerprint is SHA256:whUCFG+YLyNGofWO/DVdlQIS1z4/a/+P
VaOwjg79GkY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint]
)? yes
Warning: Permanently added '13.201.123.214' (ED25519) to the list
of known hosts.
       #_
       ~\_  ####_        Amazon Linux 2023
      ~~  \_#####\
      ~~     \###|
      ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-202
3
       ~~       V~' '->
        ~~~         /
          ~~._.   _/
             _/ _/
           _/m/'
[ec2-user@ip-192-168-0-246 ~]$
```

Step 10 : Ping instances private ip address to see a successful connection.

Use the ping command to test connectivity to the second instance using its private IP address:

ping <private-ip-of-instance-2>

```
ec2-user@ip-10-0-0-46:~

C:\Users\Arun shankar khating\Downloads>ssh -i mykey.pem ec2-user@
52.66.52.173
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-202
3
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Dec 27 06:38:04 2024 from 103.48.101.180
[ec2-user@ip-10-0-0-46 ~]$ ping 192.168.0.246
PING 192.168.0.246 (192.168.0.246) 56(84) bytes of data.
64 bytes from 192.168.0.246: icmp_seq=1 ttl=127 time=1.34 ms
64 bytes from 192.168.0.246: icmp_seq=2 ttl=127 time=0.576 ms
64 bytes from 192.168.0.246: icmp_seq=3 ttl=127 time=0.371 ms
64 bytes from 192.168.0.246: icmp_seq=4 ttl=127 time=0.764 ms
64 bytes from 192.168.0.246: icmp_seq=5 ttl=127 time=1.24 ms
64 bytes from 192.168.0.246: icmp_seq=6 ttl=127 time=0.546 ms
64 bytes from 192.168.0.246: icmp_seq=7 ttl=127 time=0.754 ms
64 bytes from 192.168.0.246: icmp_seq=8 ttl=127 time=0.390 ms
64 bytes from 192.168.0.246: icmp_seq=9 ttl=127 time=0.436 ms
64 bytes from 192.168.0.246: icmp_seq=10 ttl=127 time=0.351 ms
64 bytes from 192.168.0.246: icmp_seq=11 ttl=127 time=0.329 ms
64 bytes from 192.168.0.246: icmp_seq=12 ttl=127 time=0.739 ms
^C
--- 192.168.0.246 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11380ms
rtt min/avg/max/mdev = 0.329/0.653/1.341/0.323 ms
[ec2-user@ip-10-0-0-46 ~]$
```

```
ec2-user@ip-192-168-0-246:~

13.201.123.214
       #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-202
3
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Dec 27 06:50:18 2024 from 103.48.101.180
[ec2-user@ip-192-168-0-246 ~]$ ping 10.0.0.46
PING 10.0.0.46 (10.0.0.46) 56(84) bytes of data.
64 bytes from 10.0.0.46: icmp_seq=1 ttl=127 time=0.644 ms
64 bytes from 10.0.0.46: icmp_seq=2 ttl=127 time=0.412 ms
64 bytes from 10.0.0.46: icmp_seq=3 ttl=127 time=0.901 ms
64 bytes from 10.0.0.46: icmp_seq=4 ttl=127 time=1.32 ms
64 bytes from 10.0.0.46: icmp_seq=5 ttl=127 time=0.649 ms
64 bytes from 10.0.0.46: icmp_seq=6 ttl=127 time=0.394 ms
64 bytes from 10.0.0.46: icmp_seq=7 ttl=127 time=0.399 ms
64 bytes from 10.0.0.46: icmp_seq=8 ttl=127 time=0.407 ms
64 bytes from 10.0.0.46: icmp_seq=9 ttl=127 time=0.411 ms
64 bytes from 10.0.0.46: icmp_seq=10 ttl=127 time=0.492 ms
64 bytes from 10.0.0.46: icmp_seq=11 ttl=127 time=0.913 ms
64 bytes from 10.0.0.46: icmp_seq=12 ttl=127 time=0.660 ms
64 bytes from 10.0.0.46: icmp_seq=13 ttl=127 time=0.550 ms
^C
--- 10.0.0.46 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12426ms
rtt min/avg/max/mdev = 0.394/0.627/1.324/0.265 ms
[ec2-user@ip-192-168-0-246 ~]$
```

---

5. Verification

- Ensure the peering connection is active.
- Verify route tables have correct entries.
- Confirm security groups permit necessary traffic.
- Test instance connectivity to validate the setup.

---

6. Conclusion

By following these steps, you've successfully established a VPC peering connection between two VPCs, configured routing and security, and verified inter-VPC communication.