**VPC Peering**

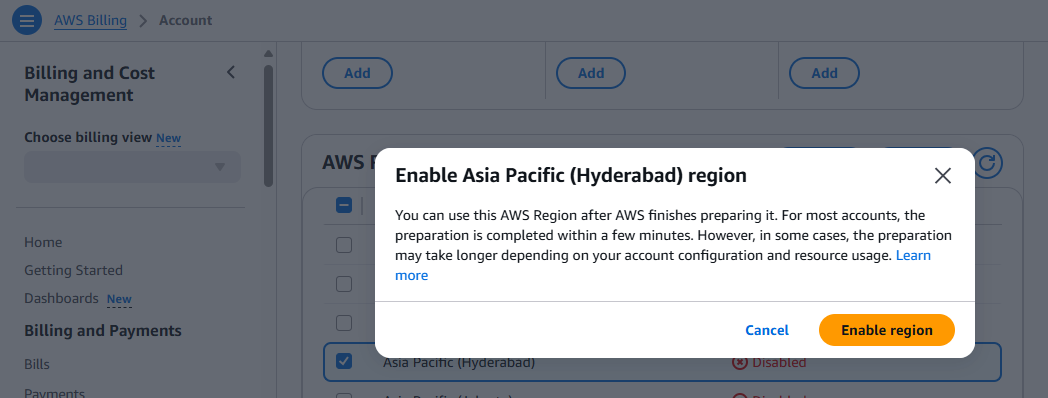
## Step by step implementation

### Create two VPCs

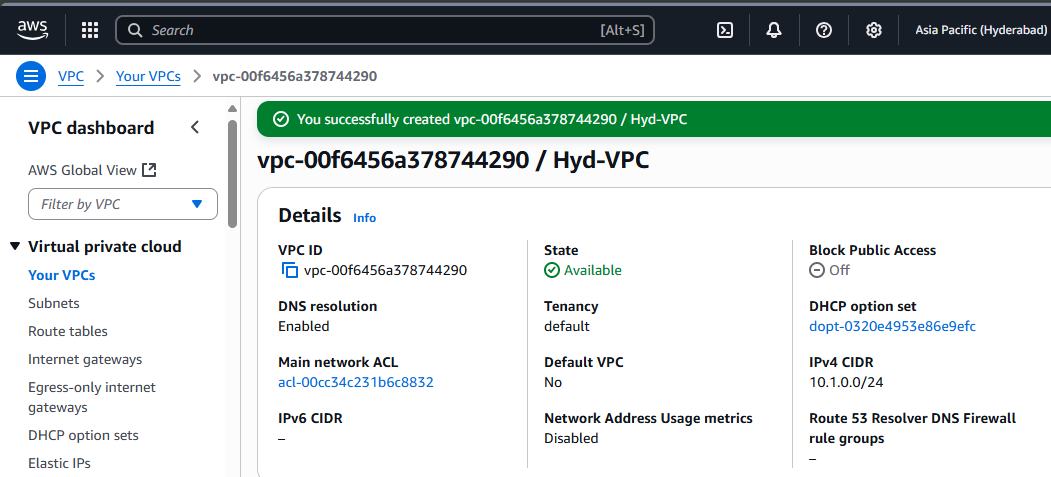
Go to the AWS Management Console → VPC service.

Switch to Mumbai (ap-south-1) region → Create a VPC with the following settings:  
Name: Mumbai-VPC  
IPv4 CIDR block: 10.0.0.0/24  
Keep defaults, no IPv6.  
Create.

In order to use the Hyderabad region, I needed to enable it -



Once done, switch to Hyderabad (ap-south-2) region → Create another VPC with CIDR block 10.1.0.0/24 and name : Hyd-VPC.



### Create Subnets in each VPC

In Mumbai, Go to VPC → Subnets → Create subnet

VPC: Mumbai-VPC

Subnet name: Mumbai-subnet

Availability Zone: pick any (e.g., ap-south-1a)

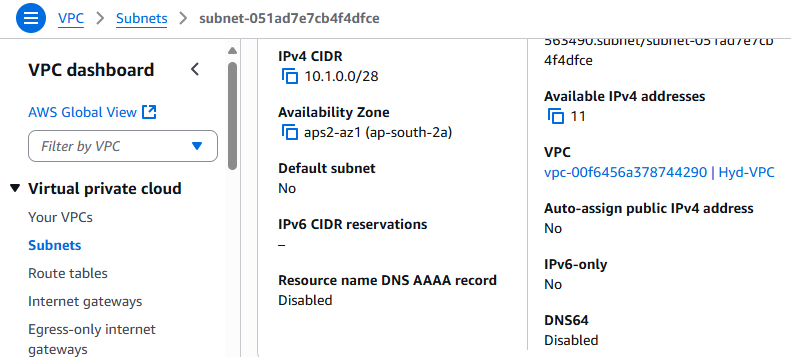
IPv4 CIDR block: 10.0.0.0/28

Create subnet

Similarly create one for Hyderabad region.

Subnet Name - Hyd-subnet

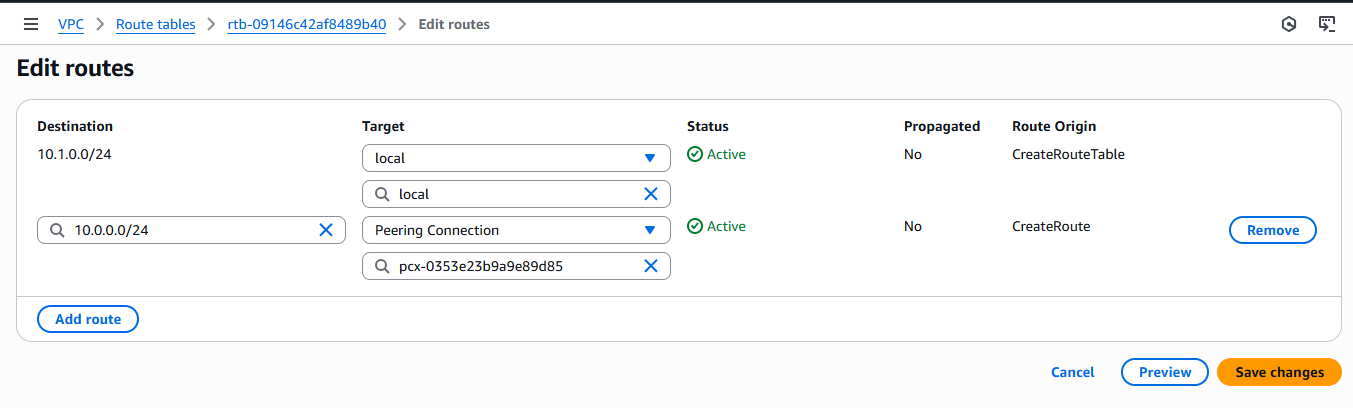
IPv4 CIDR block: 10.1.0.0/28



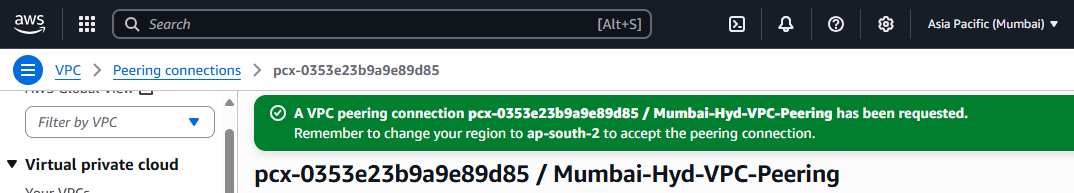
### Create a VPC Peering Connection

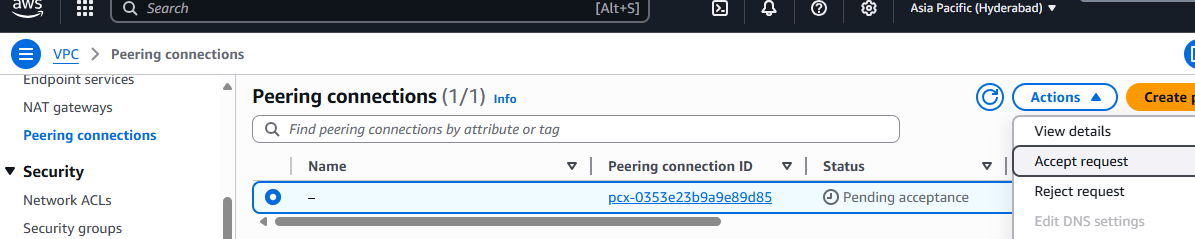
We’ll make Mumbai-VPC the requester and Hyd-VPC the accepter.

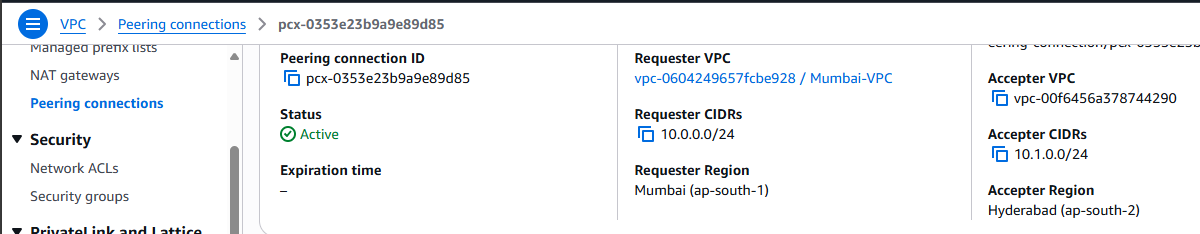
Switch to Mumbai (ap-south-1) region.  
In the VPC console → Peering Connections → Create peering connection.

* + Name tag: Mumbai-Hyd-Peering
  + VPC (requester): Mumbai-VPC
  + VPC (accepter): Another region → select ap-south-2. From ap-south-2 region, copy the VPC ID of Hyd-VPC and paste it in VPC ID (Accepter)
  + Leave other defaults.
  + Create a peering connection.

Now switch to Hyderabad (ap-south-2) region. Go to Peering Connections, we’ll see a pending request.  
Select it → Accept request.



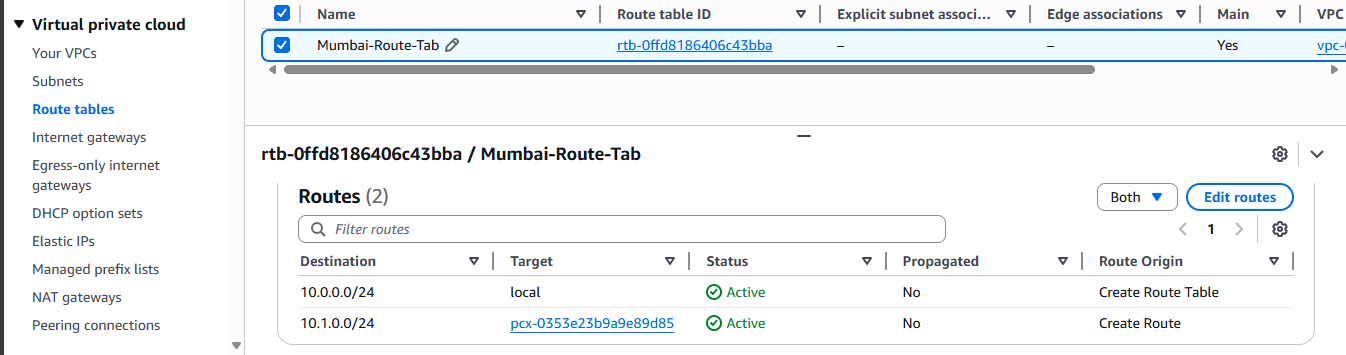




### Update Route Tables for Peering

In the Mumbai region, Go to VPC → Route Tables.  
Find the route table associated with your Mumbai-Subnet (if only one exists, it’s usually the “Main” route table).  
Edit routes → Add route:

* + Destination: 10.1.0.0/24 (CIDR of Hyd-VPC)
  + Target: Select ‘Peering Connection’ and then your Mumbai-Hyd-VPC-Peering connection.Save changes.



Similarly we’ll do it for ap-south-2.

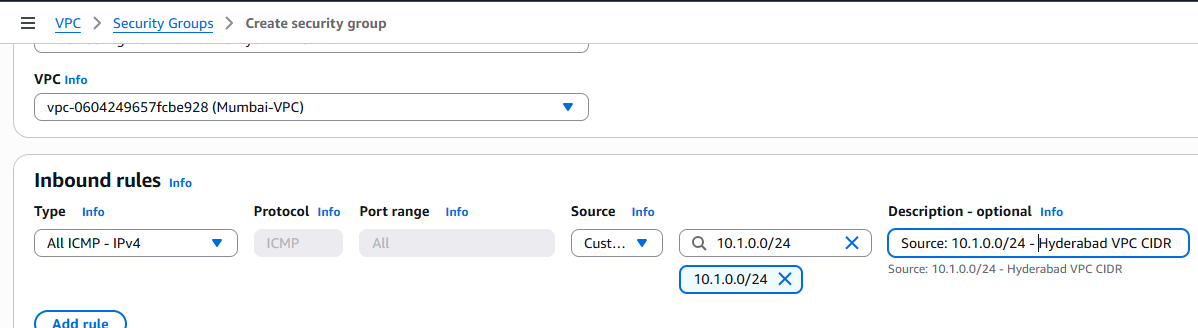
### Configure Security Groups

In Mumbai, Go to VPC → Security Groups → **Create security group**

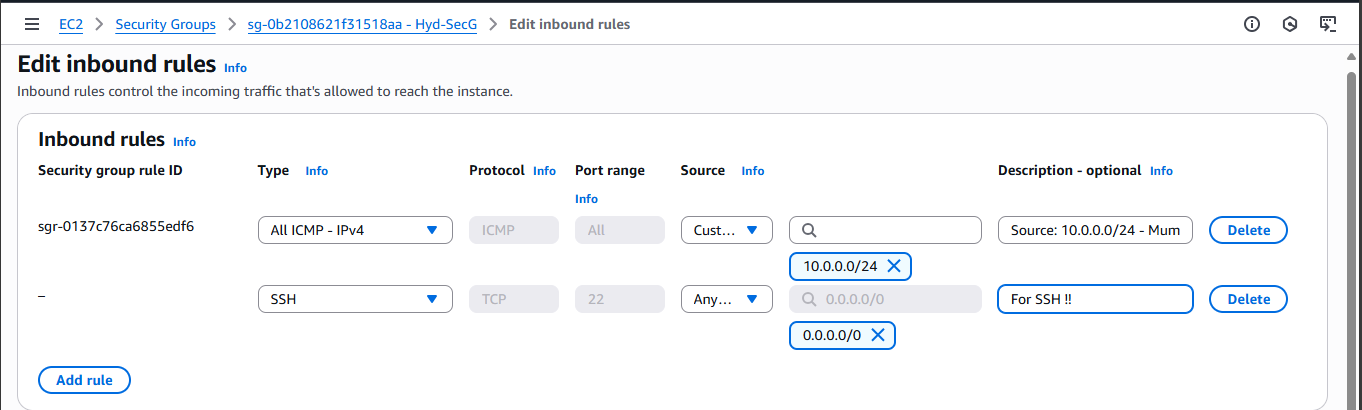
* + Name: Mumbai-SG
  + VPC: Mumbai-VPC

**Add Inbound Rule**:

* + Type: All ICMP - IPv4
  + Source: 10.1.0.0/24 (Hyderabad VPC CIDR) Save.



Similarly create Hyd-SecG.



### Network Setup -

**1. Create Internet Gateways (IGW)**

* Navigate to VPC → Internet Gateways.
* Click Create internet gateway, give it a name (e.g., Hyd-IGW, Mumbai-IGW).
* The IGW will be created in a detached state.

**2. Attach IGW to the VPC**

* Select the newly created IGW.
* Click Actions → Attach to VPC.
* Choose the respective VPC (Hyd-VPC or Mumbai-VPC) and attach.
* Verify that the IGW State is now Attached.

**3. Update Route Tables & Associate Subnets**

* Go to VPC → Route Tables.
* Select the route table associated with your subnet (e.g., Hyd-Subnet, Mumbai-Subnet).
* In the Routes tab:  
  Add a new route:  
  Destination: 0.0.0.0/0  
  Target: <IGW-ID>
* In the Subnet Associations tab:  
  Explicitly associate the route table with the correct subnet (this step is important - if your subnet shows up under *“Subnets without explicit associations”*, fix it here).
* After this step, your subnet becomes a public subnet with internet access via the IGW.

**4. Update Security Groups for SSH**

* Go to EC2 → Security Groups.
* Edit inbound rules of the Security Group attached to your EC2 instances.Add the following rule:  
  Type: SSH   
  Protocol: TCP  
  Port Range: 22
* Source: My IP (or 0.0.0.0/0 for testing only)

### Connecting to EC2 Instances via CloudShell

In CloudShell, click Actions → Upload file.Select the .pem file for your instance  
Run the following command to secure the key file (otherwise SSH will reject it):

>>> chmod 400 hyd-key-pair.pem

Go to EC2 → Instances → [Your EC2] → Networking tab.

Copy the Public IPv4 address of the instance.Use the command below:

>>> ssh -i hyd-key-pair.pem ec2-user@<Public-IP>

Repeat the same process for both Mumbai EC2 and Hyderabad EC2, each with their respective key pair and public IP.

### Testing the VPC Peering Connection

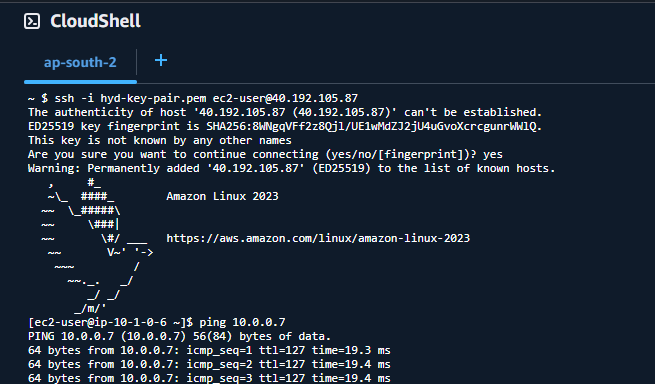
Get Private IPs from the EC2 Console → Instances → Networking tab.

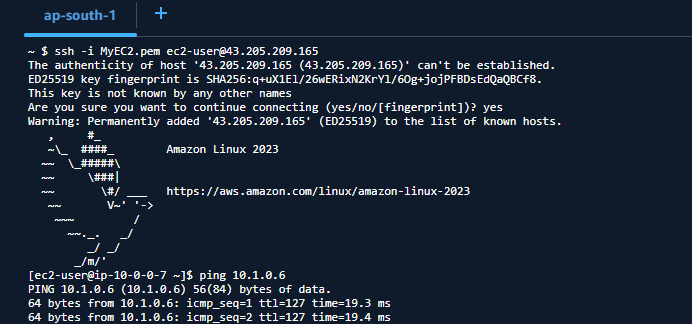
Example:  
Hyd-EC2 private IP → 10.1.0.123  
Mumbai-EC2 private IP → 10.0.0.45

In our Mumbai EC2 SSH session**:**  
>>> ping 10.1.0.123

Replace with the actual Hyd-EC2 private IP.  
You should see successful replies:  
64 bytes from 10.1.0.123: icmp\_seq=1 ttl=255 time=1.23 ms  
  
In our Hyd EC2 SSH session (CloudShell tab):  
>>> ping 10.0.0.45  
Replace with the actual Mumbai-EC2 private IP.  
  
If pings fail:

* Check Security Groups → Ensure ICMP is allowed from the peer VPC CIDR.
* Check Route Tables → Verify 0.0.0.0/0 → IGW (for internet access) and CIDR → Peering Connection (for cross-VPC access).
* Check Peering Status → In VPC → Peering Connections, status should be Active





## Conclusion

In this project, we set up **cross-region VPC peering** between Mumbai and Hyderabad using AWS Free Tier resources. After configuring Internet Gateways, Route Tables, Subnet Associations, and Security Groups, we launched EC2 instances in both VPCs and verified private communication via ping tests. This project provides a beginner-friendly introduction to secure multi-region networking on AWS.