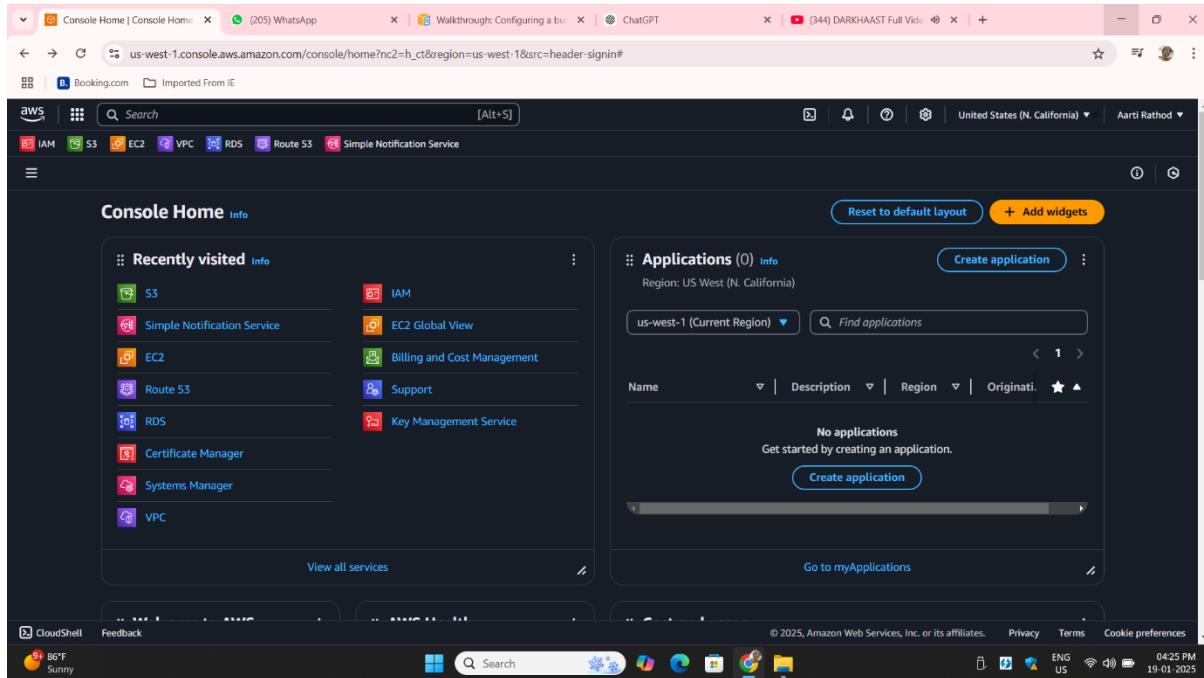


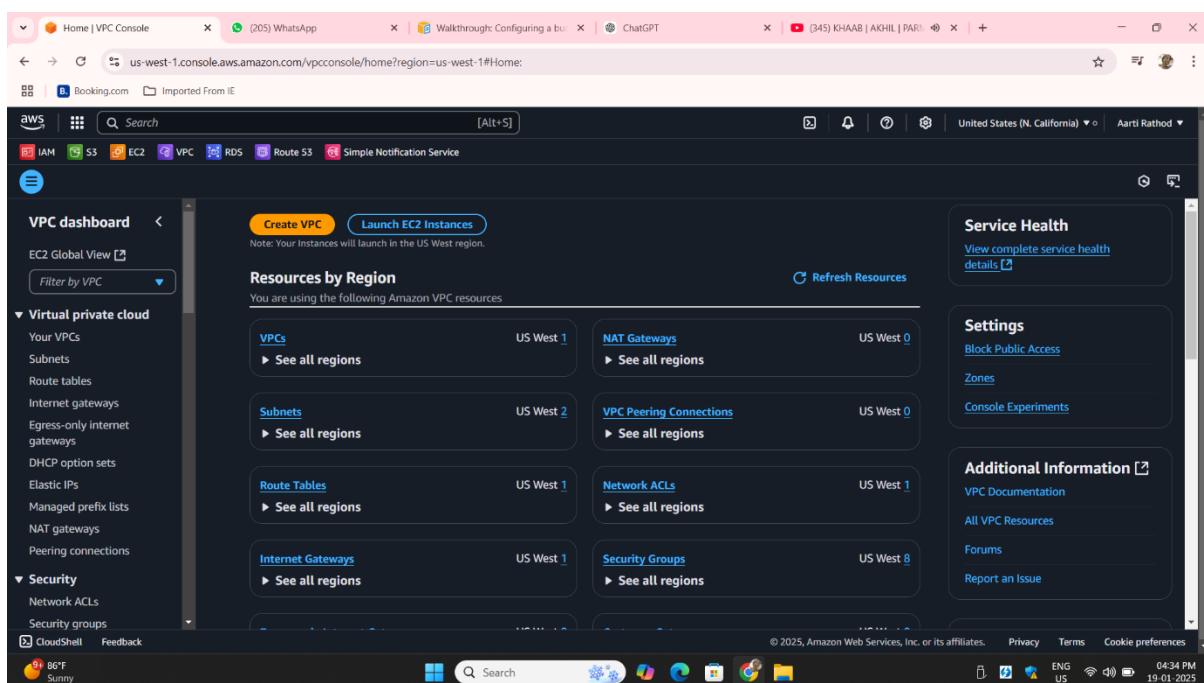
Steps for Setting Up VPC Peering and Instance Access via Jump Server

Step1: Log In to AWS Console:

Open your web browser and navigate to the [AWS Management Console](#).



Step2: Mumbai Region Setup: Select the VPC Service:From the search results, click on VPC (Virtual Private Cloud).



Step3 : Go to your VPC click on create VPC

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with options like 'Virtual private cloud' (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections), 'Security' (Network ACLs, Security groups), and 'CloudShell'. The main area is titled 'Your VPCs (1) Info' and shows a table with one row:

Name	VPC ID	State	Block Public...	IPv4 CIDR	IPv6 CIDR
vpc-0a3427c350f4857a	Available	Off	172.31.0.0/16	-	-

At the top right, there are 'Actions' and 'Create VPC' buttons. Below the table, it says 'Select a VPC above'. The bottom of the screen shows the Windows taskbar with various icons.

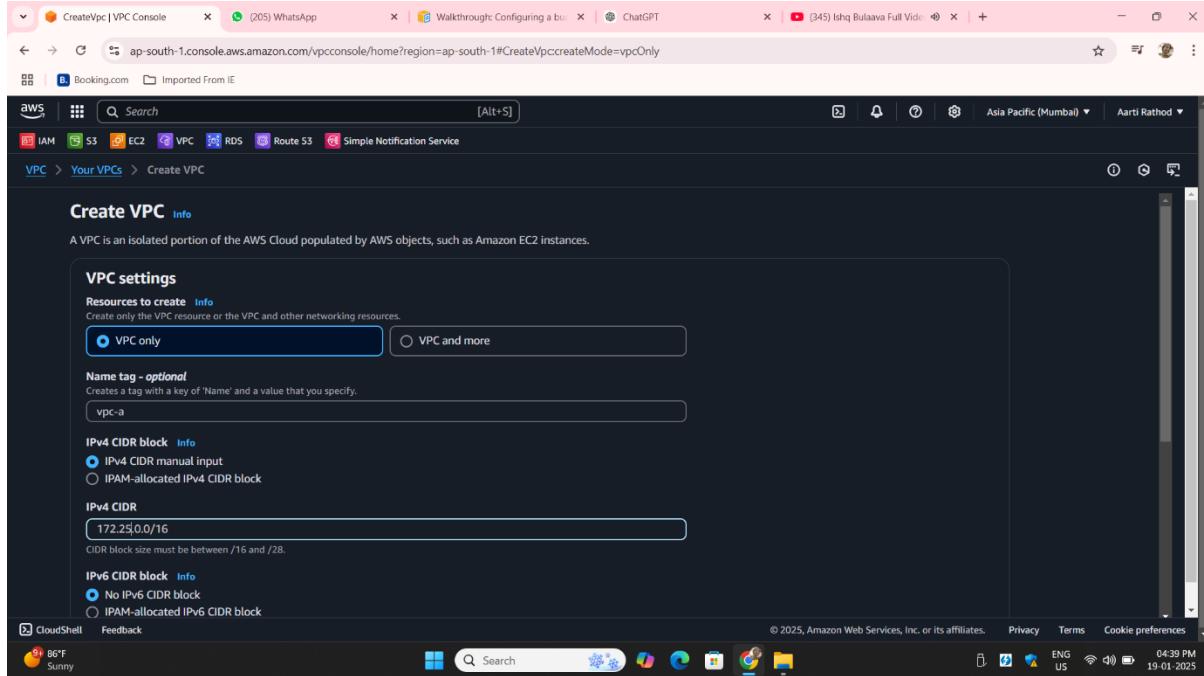
Step5: Select VPC only or VPC and more based on your needs. For simplicity, choose VPC only.

The screenshot shows the 'Create VPC' configuration page. At the top, it says 'CreateVpc | VPC Console' and 'CreateVpccreateMode=vpcOnly'. The main section is titled 'Create VPC Info' and contains the following fields:

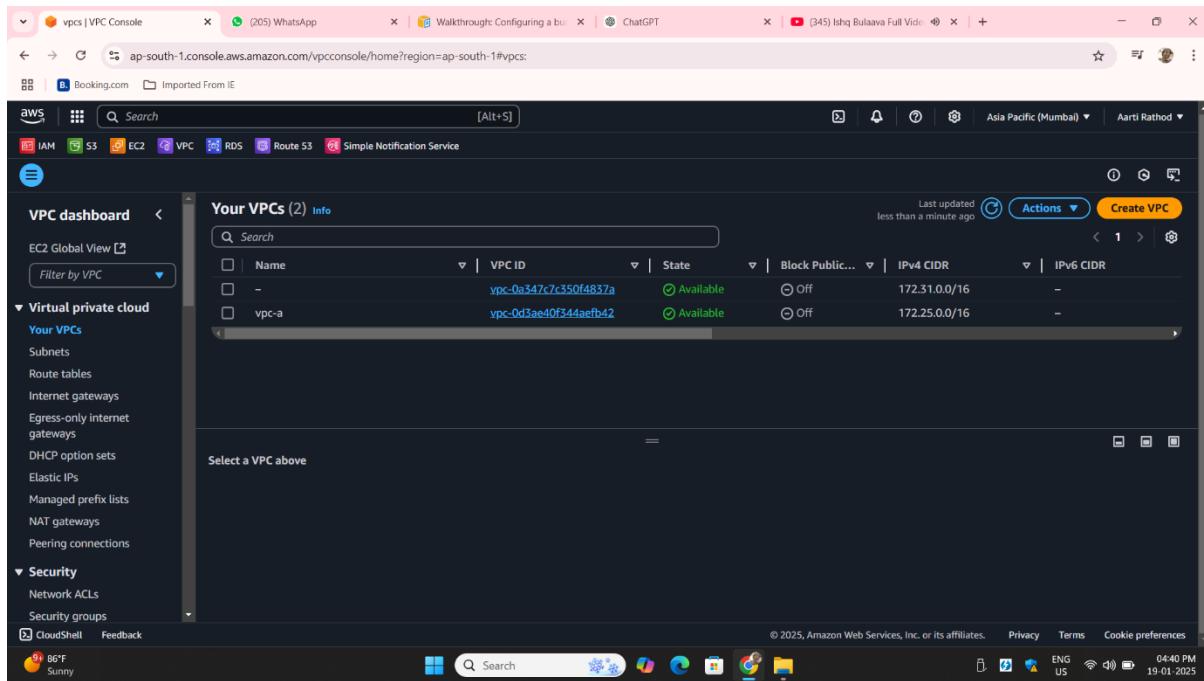
- VPC settings**: A radio button group for 'Resources to create' with 'VPC only' selected (highlighted in blue).
- Name tag - optional**: A text input field containing 'my-vpc-01'.
- IPv4 CIDR block**: A radio button group for 'IPv4 CIDR manual input' (selected) and 'IPAM-allocated IPv4 CIDR block'.
- IPv4 CIDR**: An input field containing '10.0.0.0/24'.
- IPv6 CIDR block**: A radio button group for 'No IPv6 CIDR block' (selected) and 'IPAM-allocated IPv6 CIDR block'.

At the bottom, there are 'CloudShell' and 'Feedback' buttons, and the Windows taskbar is visible at the bottom.

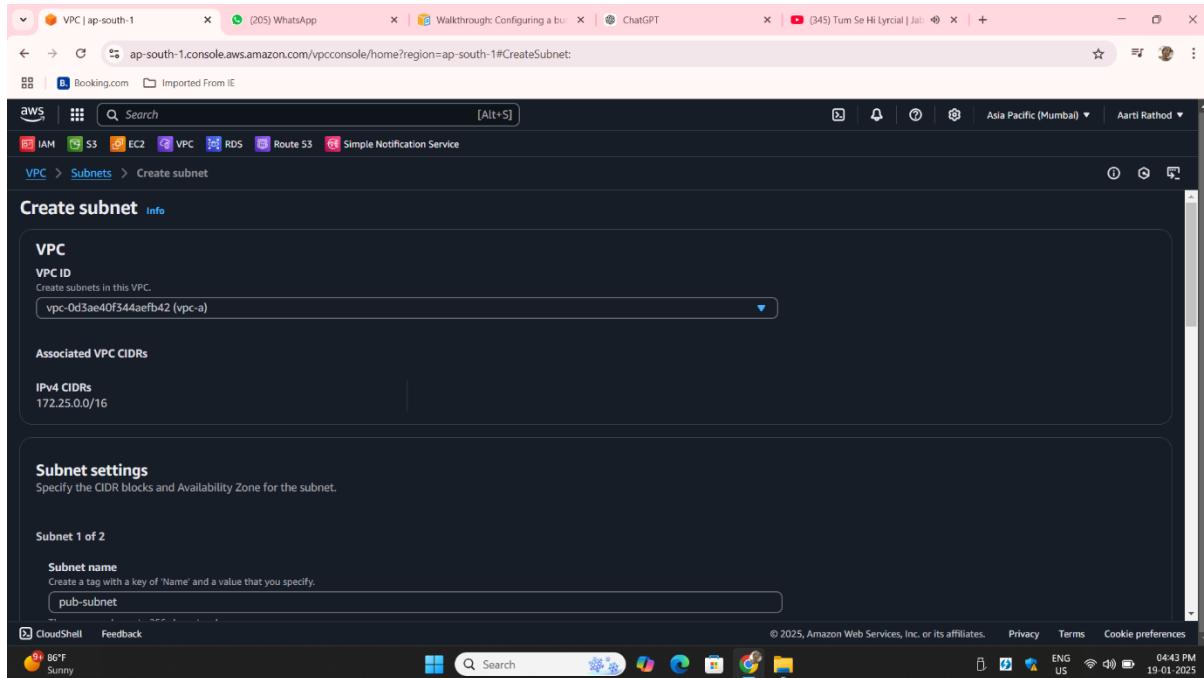
Step6: Name tag: Enter a name for your VPC vpc a IPv4 CIDR block: Define the CIDR range for your VPC 172.25.0.0/16 Click on create VPC



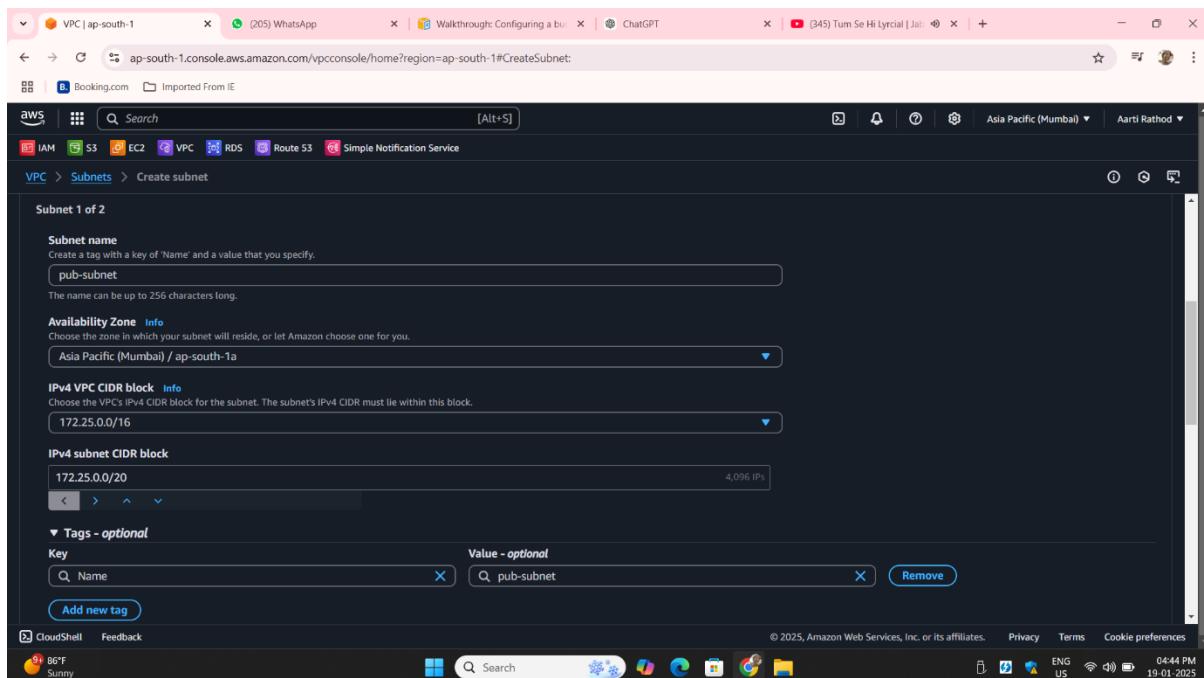
Step7: Here your vpc is created



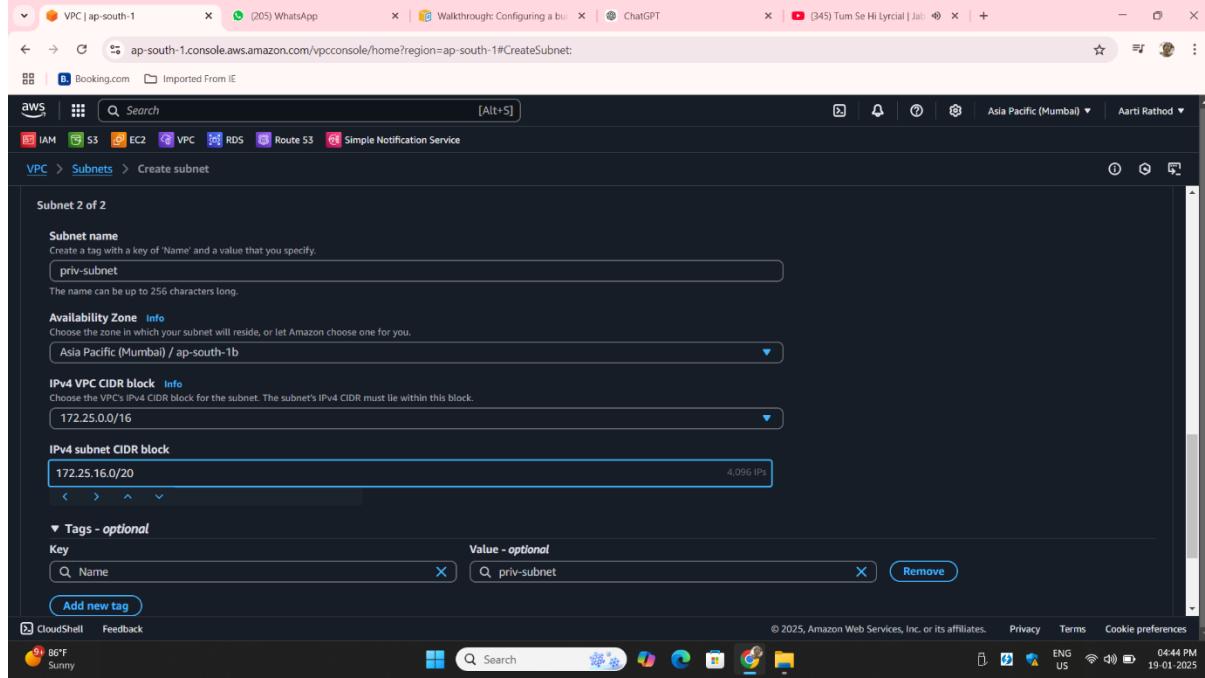
Step 8: Create Subnets: Public and private subnets within the VPC. Select the VPC you want to create the subnet in give a subnet name pub-subnet.



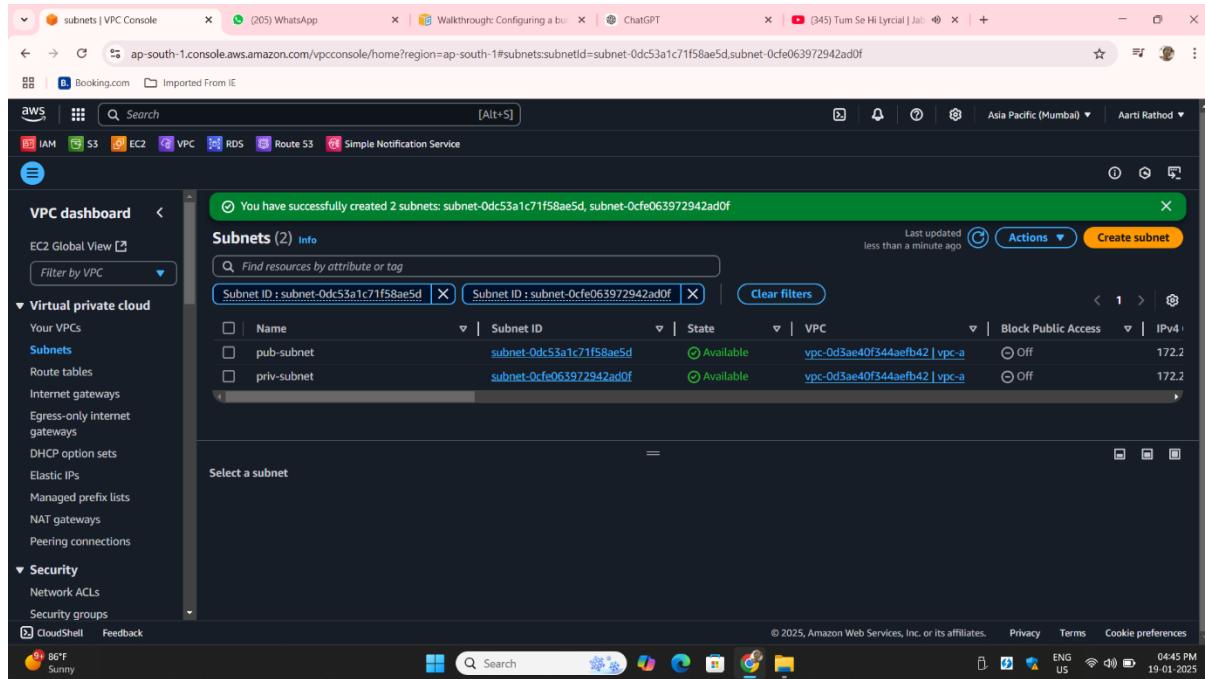
**Step 9 : Availability Zone: Choose an availability zone ap-south-1a
IPv4 CIDR block: Assign a CIDR block for the subnet 172.25.0.0/20**



Step10: give a second subnet name priv-subnet Availability Zone: Choose an availability zone ap-south-1b IPv4 CIDR block: Assign a CIDR block for the subnet 172.25.16.0/20



Step10 : Here your two subnet is created public and private



Step 11: Go to EC2 Dashboard in the AWS Management Console.

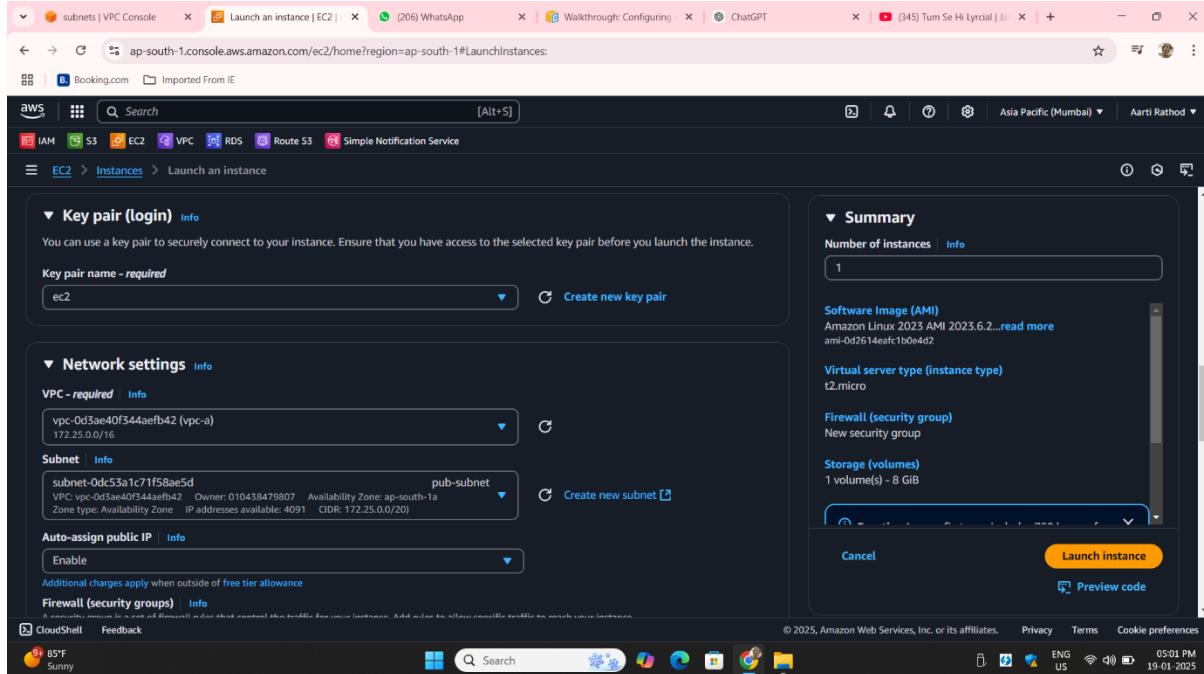
Click on Launch Instance

The screenshot shows the AWS Management Console EC2 Dashboard. The left sidebar includes sections for Instances, Images, and Elastic Block Store. The main area displays EC2 resources in the Asia Pacific (Mumbai) Region, such as 0 instances running, 0 auto scaling groups, 0 capacity reservations, etc. Below this are sections for Launch instance, Service health, and Offer usage (monthly). A note at the bottom states: "Note: Your instances will launch in the Asia Pacific (Mumbai) Region". The status bar at the bottom right shows the date as 19-01-2025 and the time as 04:55 PM.

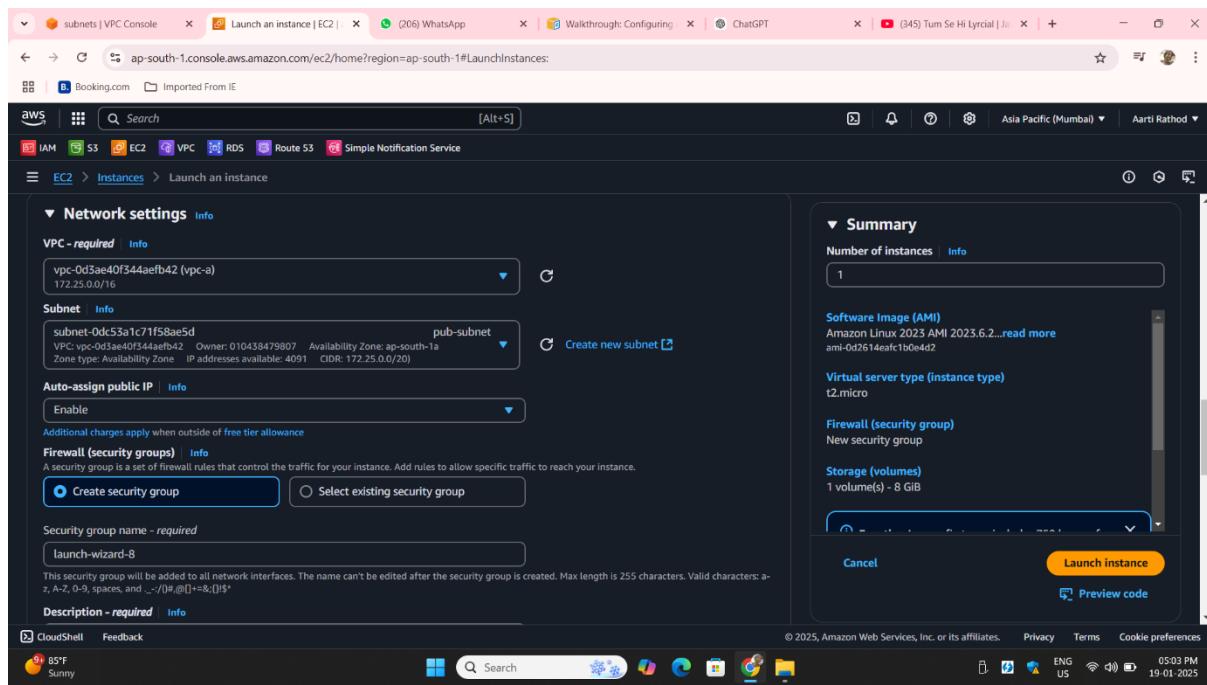
Give a name jump server(public instance)

The screenshot shows the "Launch an instance" wizard. In the "Name and tags" step, the name "jump server" is entered. In the "Application and OS Images (Amazon Machine Image)" step, the search bar is empty. On the right, the "Summary" section shows 1 instance selected, with options for Software Image (AMI), Virtual server type (instance type), Firewall (security group), and Storage (volumes). A message indicates a free tier of 750 hours is included. At the bottom right is a "Launch instance" button.

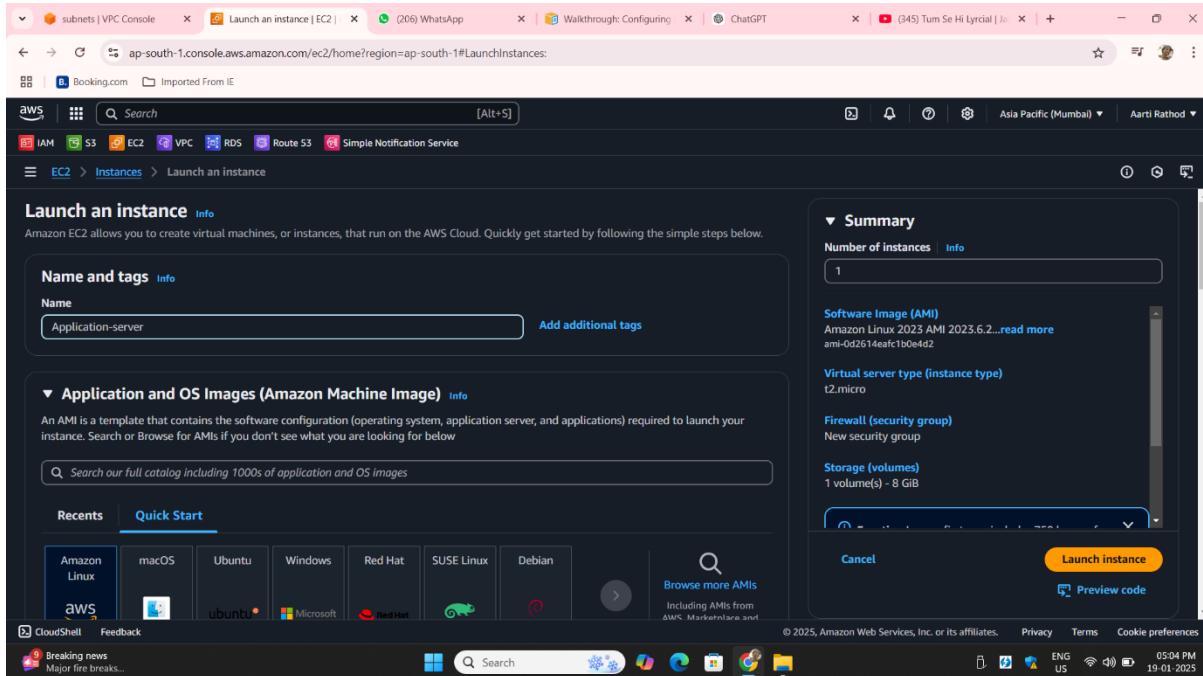
Step: Select key pair go to network setting select your vpc public subnet then enable auto assing ipcreta



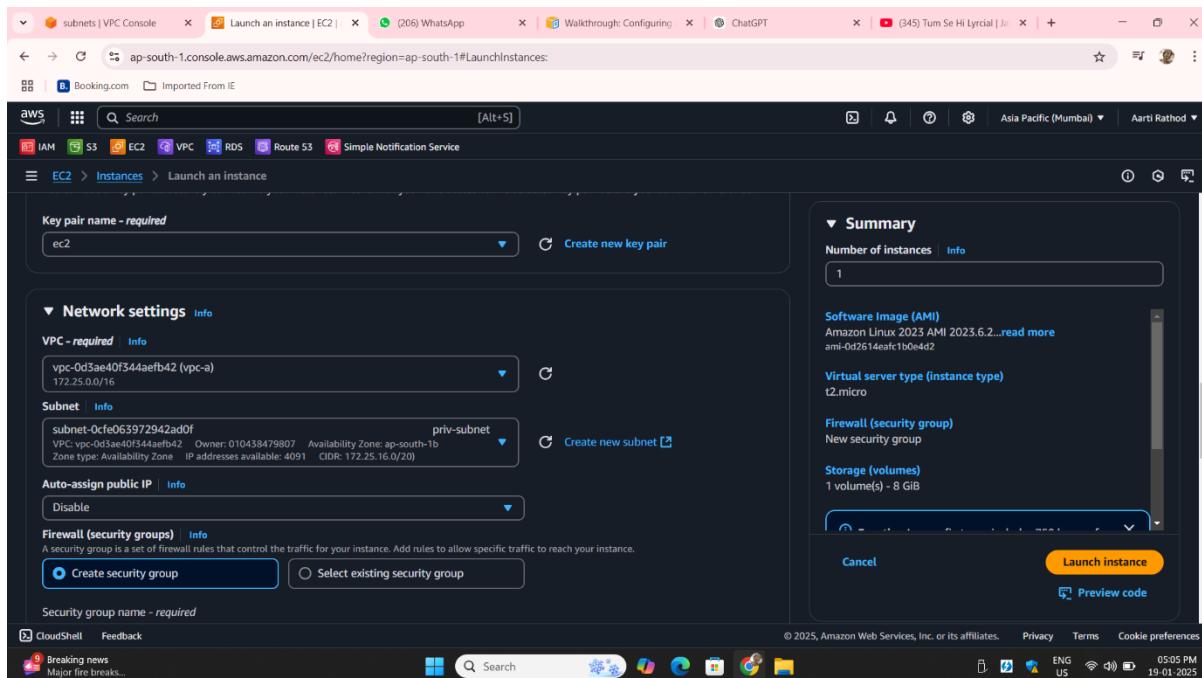
Create security group and click on Launch Instance



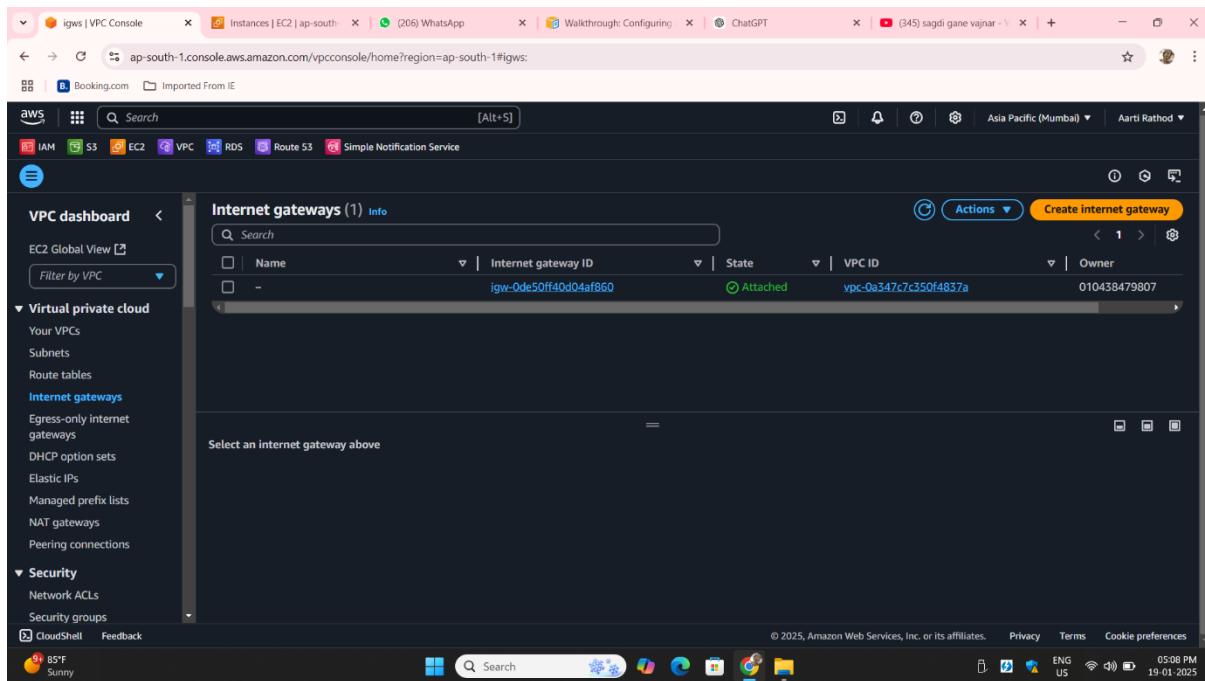
Create another instance give a name application server (private instance)



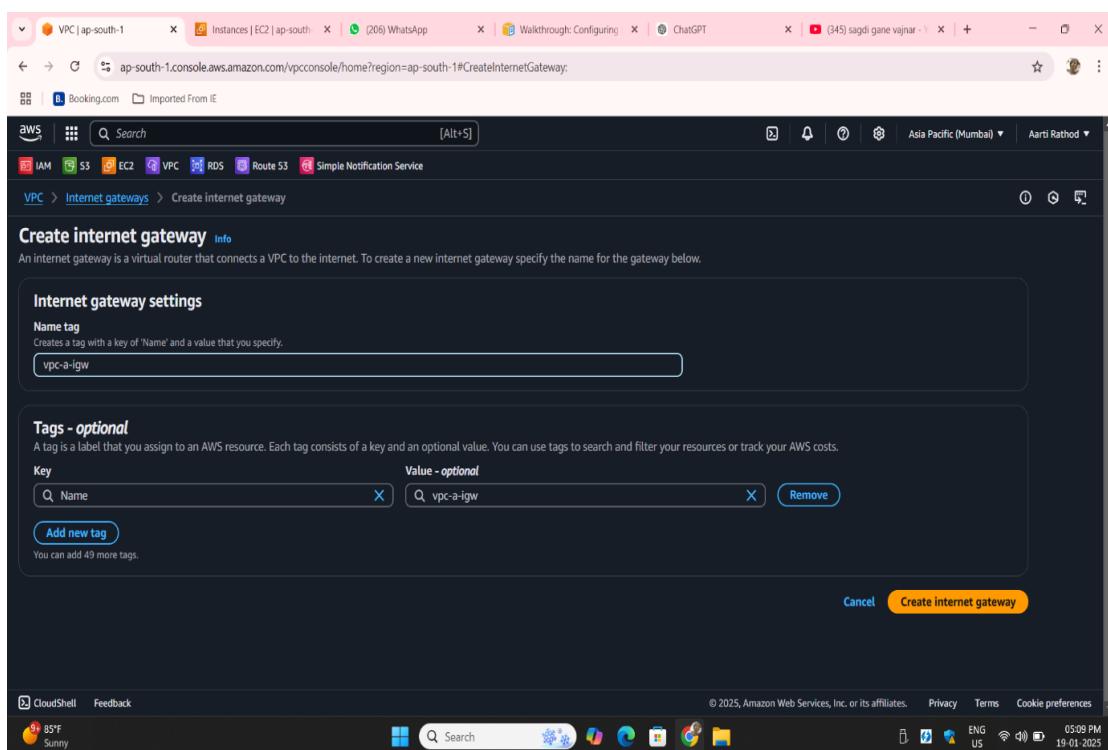
Select key pair go to network settings edit select vpc private subnet create security group and click on launch instance



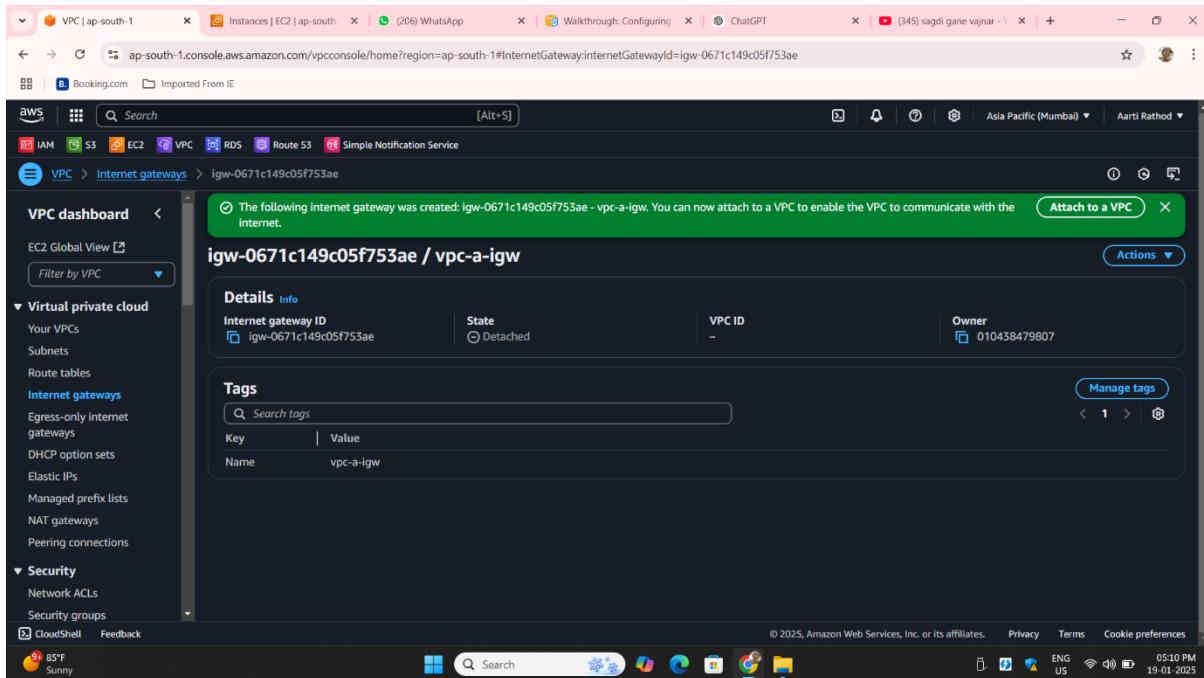
Go to vpc In the left-hand menu, click on Internet Gateways.Click the Create Internet Gateway button.



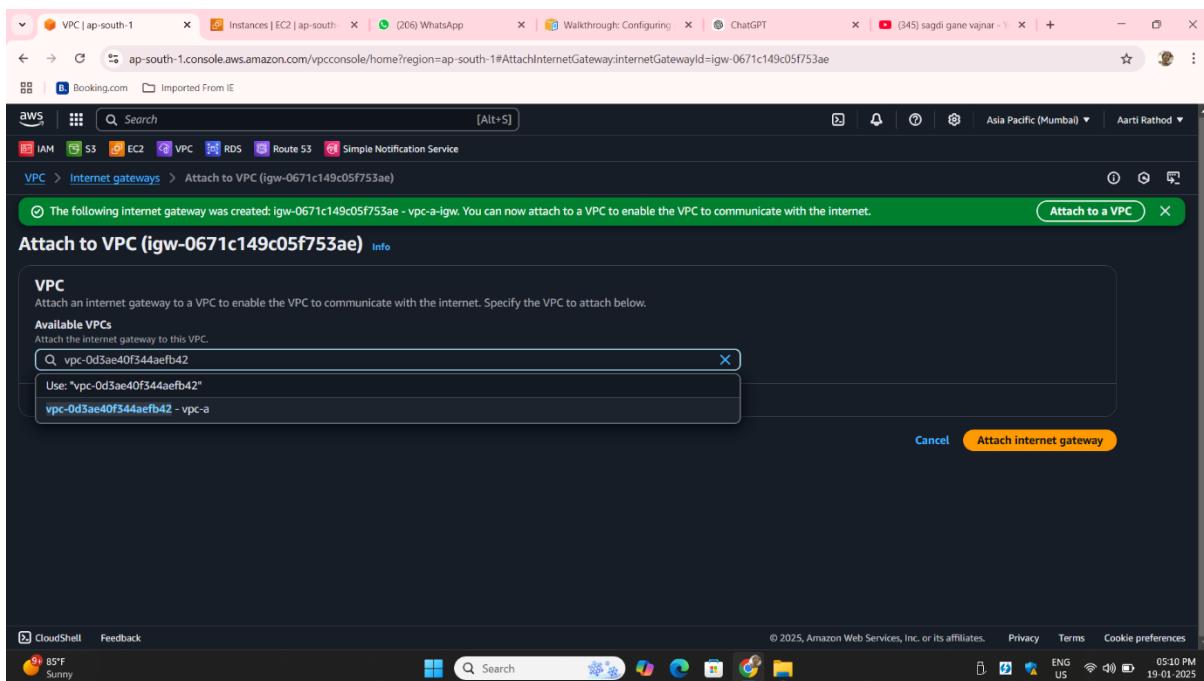
Step 14: Provide a Name tag `vpc-a-igw` Click Create Internet Gateway.



Step 15: Attach the Internet Gateway to a VPC Once the IGW is created, click the Actions dropdown next to the IGW. Select Attach to VPC.

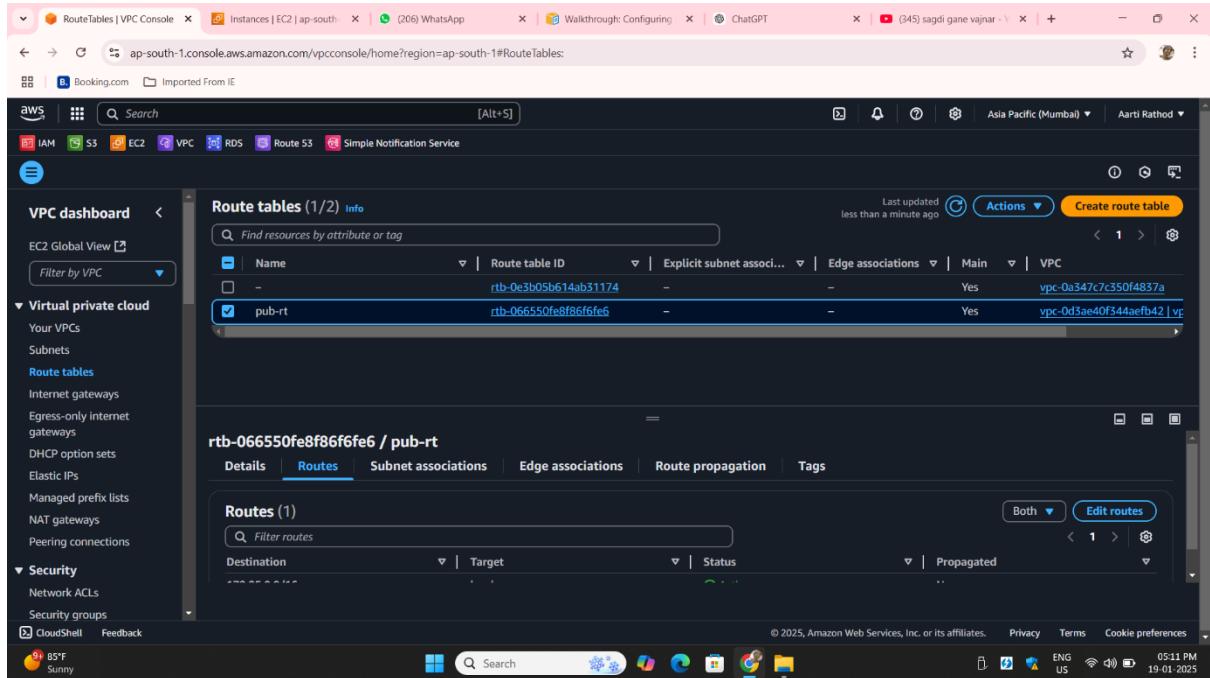


Step 16: Choose the VPC you want to attach the IGW to. Click Attach Internet Gateway. Go to the Route Tables section in the VPC Dashboard.



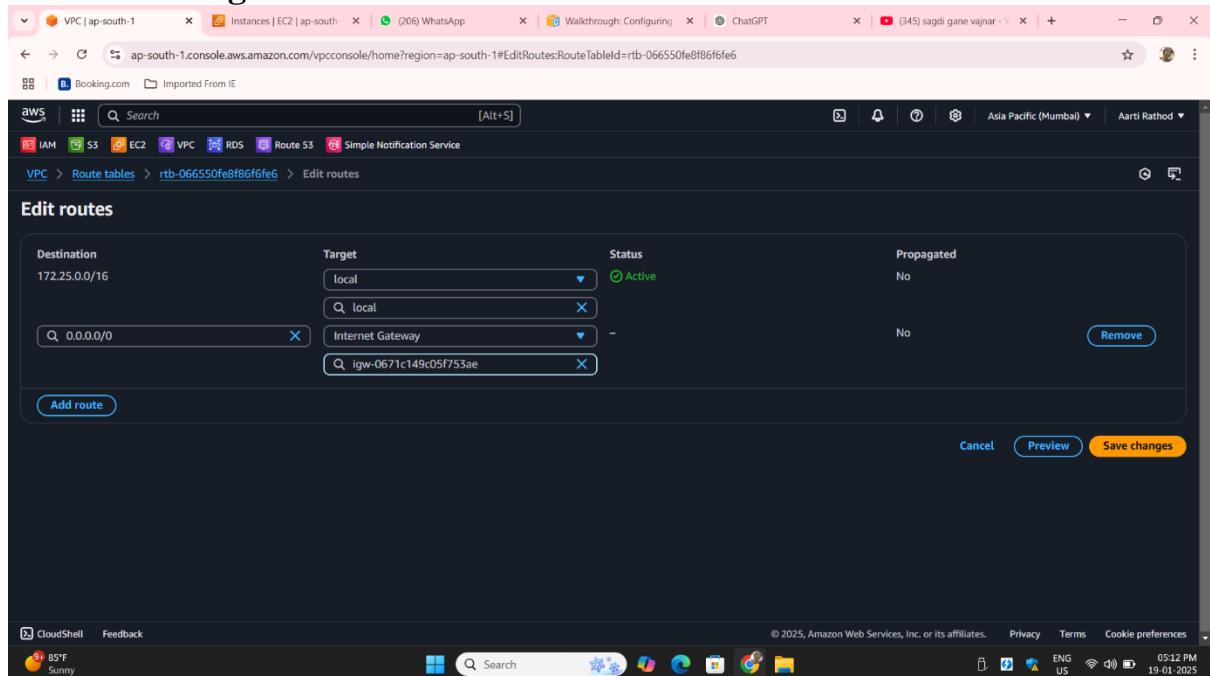
By

default one route table is created giving a name pub-rt. Select the route table and go to route.



The screenshot shows the AWS VPC Route Tables page. On the left, there's a sidebar with 'Virtual private cloud' expanded, showing 'Route tables' selected. The main area displays a table titled 'Route tables (1/2)'. It has columns for Name, Route table ID, Explicit subnet associa..., Edge associations, Main, and VPC. A row for 'pub-rt' is selected, showing its Route table ID as 'rtb-066550fe8f86f6fe6'. Below this, a detailed view for 'rtb-066550fe8f86f6fe6 / pub-rt' is shown with tabs for Details, Routes, Subnet associations, Edge associations, Route propagation, and Tags. The 'Routes (1)' tab is selected, showing a single route with Destination '0.0.0.0/0', Target 'local', Status 'Active', and Propagated status 'No'. The bottom of the screen shows the AWS navigation bar and system status.

Add a new route Destination: 0.0.0.0/0 Target: Select the Internet Gateway Save the changes.



The screenshot shows the 'Edit routes' dialog for the 'rtb-066550fe8f86f6fe6' route table. It has a table with columns for Destination, Target, Status, and Propagated. There are two rows: one for '172.25.0.0/16' with Target 'local' and Status 'Active'; and another for '0.0.0.0/0' with Target 'Internet Gateway' and Status '-' (indicating it's being edited). The 'Add route' button is at the bottom left, and 'Cancel', 'Preview', and 'Save changes' buttons are at the bottom right. The bottom of the screen shows the AWS navigation bar and system status.

After that go to subnet association edit subnet association select public subnet and save changes.

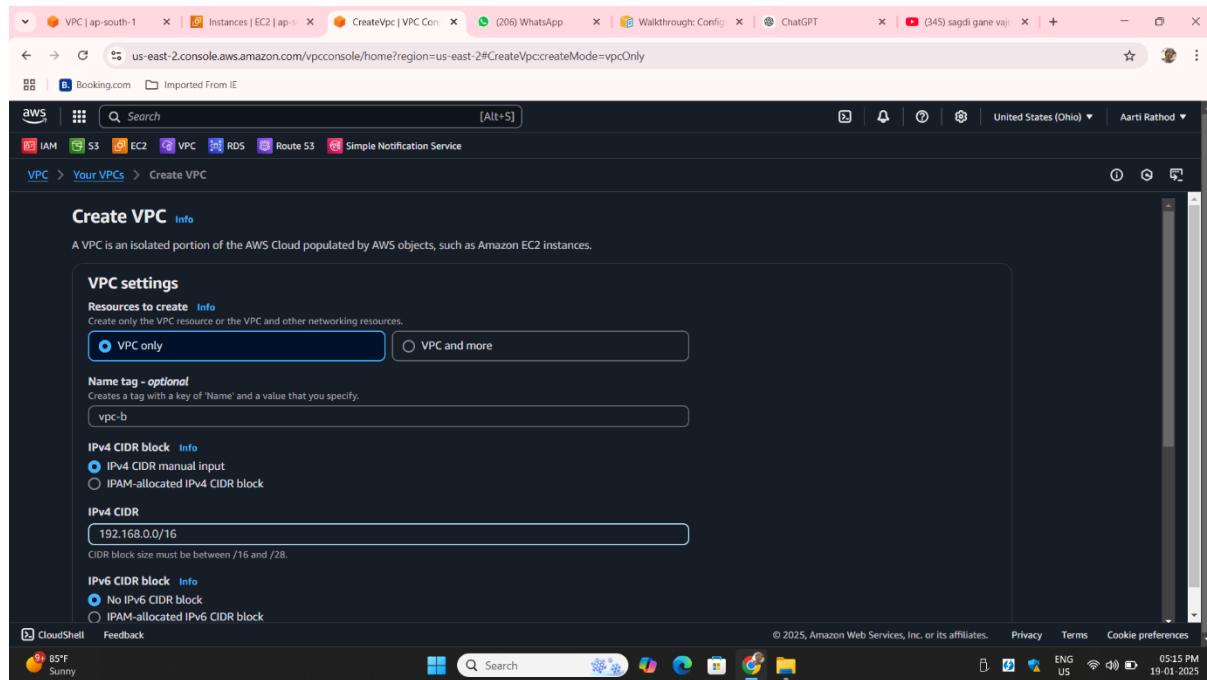
The screenshot shows the AWS VPC Route Tables page. A green success message at the top states: "Updated routes for rtb-066550fe8f86f6fe6 / pub-rt successfully". Below this, the route table details are shown: Route table ID (rtb-066550fe8f86f6fe6), Main (Yes), VPC (vpc-0d3ae40f344aefb42 | vpc-a), and Owner ID (01043847907). The Subnet associations tab is selected, showing "Explicit subnet associations (0)". The bottom right corner of the browser window shows the date and time as 19-01-2025.

Ohio Region Setup: Select the VPC Service:

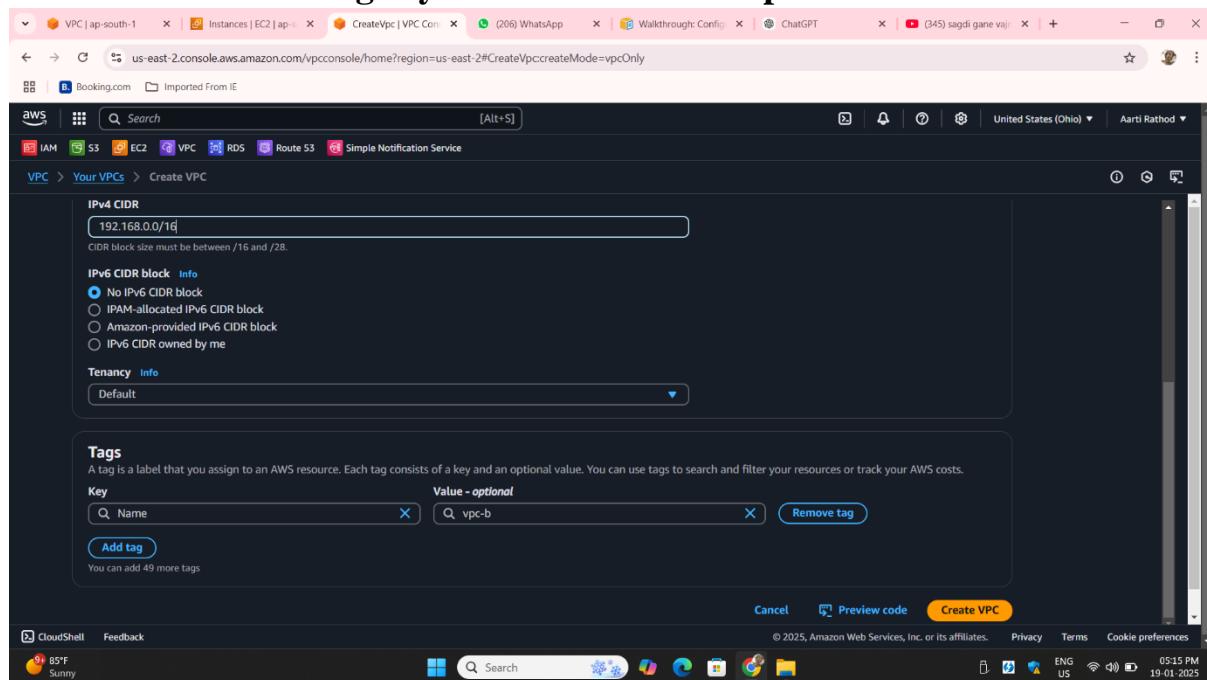
The screenshot shows the AWS VPC dashboard. The "Your VPCs (1) Info" section displays a single VPC entry: Name (vpc-02af05c4a2e06e31), VPC ID (vpc-02af05c4a2e06e31), State (Available), Block Public (Off), IPv4 CIDR (172.31.0.0/16), and IPv6 CIDR (not specified). The "Actions" button is highlighted in orange. The bottom right corner of the browser window shows the date and time as 19-01-2025.

Click on create VPC

Select VPC only or VPC and more based on your needs. For simplicity,



**choose VPC only. Give a name tag vpc-b give a CIDR range 192.168.0.0/16
Leave the other settings by default and create VPC.**



Go to subnet click on create subnet

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with 'VPC dashboard' and 'Virtual private cloud' sections. Under 'Subnets', it lists 'Route tables', 'Internet gateways', 'Egress-only internet gateways', 'DHCP option sets', 'Elastic IPs', 'Managed prefix lists', 'NAT gateways', and 'Peering connections'. Below that is a 'Security' section with 'Network ACLs' and 'Security groups'. At the bottom of the sidebar are 'CloudShell' and 'Feedback' buttons. The main area has a title 'Subnets (3) Info' with a search bar. A table lists three subnets:

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR
-	subnet-0237e6796f348cd5d	Available	vpc-02af05c4a2e06e31	Off	172.31.32.0/
-	subnet-0d91bf1b9de4a7cbb	Available	vpc-02af05c4a2e06e31	Off	172.31.0.0/2
-	subnet-0dafaec724fef1d22	Available	vpc-02af05c4a2e06e31	Off	172.31.16.0/

A 'Create subnet' button is located at the top right of the table area. Below the table, there's a section titled 'Select a subnet'.

Select your vpc give a name to your vpc database subnet give the CIDR range 192.168.0.0/20

The screenshot shows the 'Create subnet' wizard. The first step, 'VPC', is selected. It shows a dropdown menu for 'VPC ID' containing 'vpc-072e9631fd1692336 (vpc-b)'. Below that is a section for 'Associated VPC CIDRs' with 'IPv4 CIDRs' set to '192.168.0.0/16'. The next section is 'Subnet settings', which asks to 'Specify the CIDR blocks and Availability Zone for the subnet'. Under 'Subnet 1 of 1', there's a 'Subnet name' field containing 'database subnet'. At the bottom, there are 'CloudShell' and 'Feedback' buttons.

Here your subnet is created

The screenshot shows the AWS VPC dashboard. A green success message at the top states: "You have successfully created 1 subnet: subnet-0880c0dedbf15ac6e". The main table displays one subnet entry:

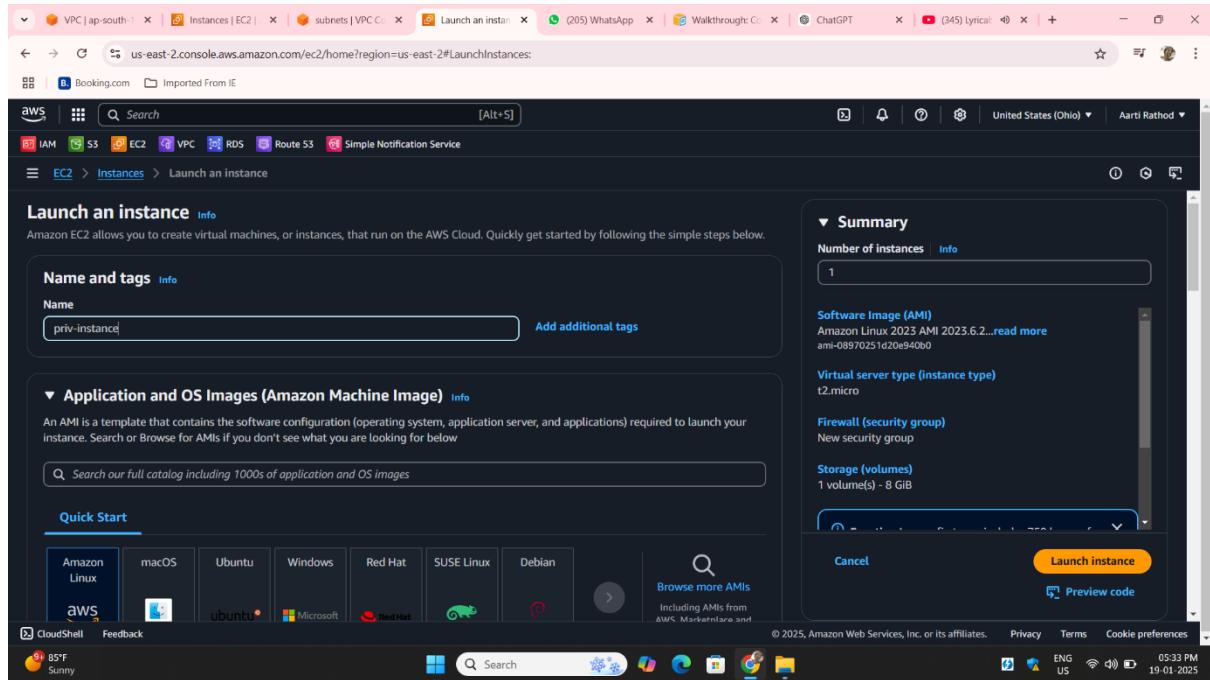
Name	Subnet ID	State	VPC	Block Public...	IPv4 CDR
database subnet	subnet-0880c0dedbf15ac6e	Available	vpc-072e9631fd1692336 vpc-b	Off	192.168.0.0/16

The sidebar on the left includes sections for Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections), Security (Network ACLs, Security groups), and CloudShell.

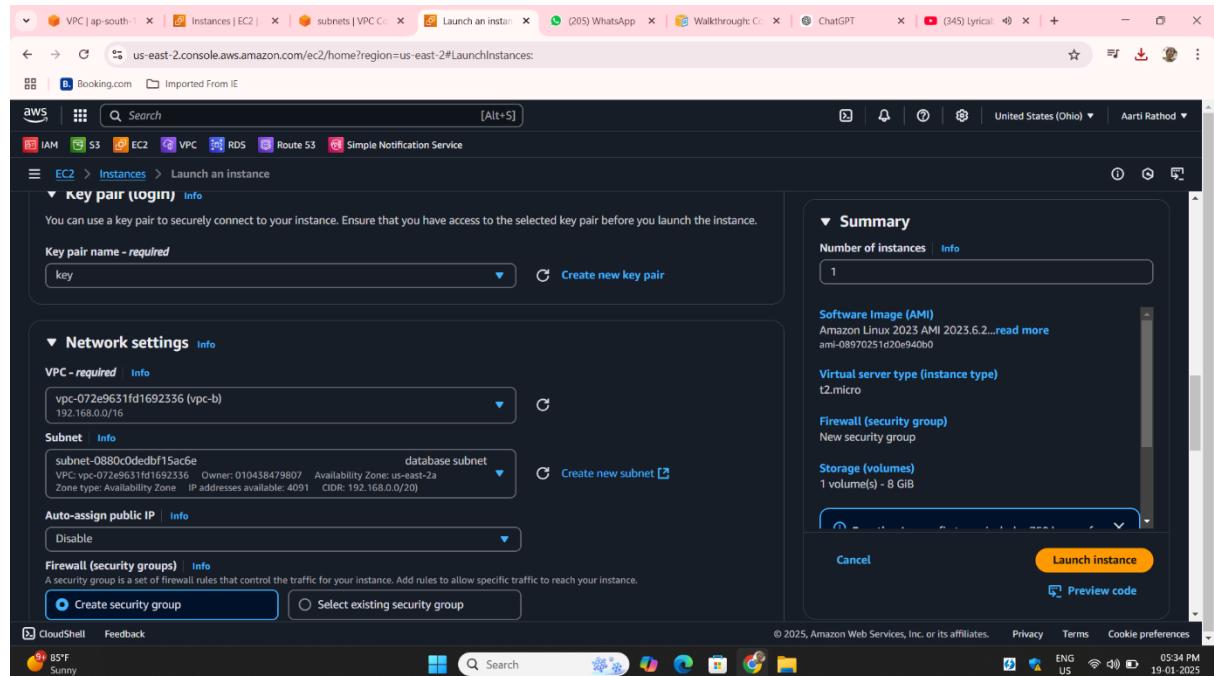
Go to ohio ec2 service click on launch instance

The screenshot shows the AWS EC2 Instances page. The main message says: "No instances. You do not have any instances in this region". A "Launch instances" button is visible. The sidebar on the left includes sections for Instances (Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes, Snapshots).

Give a instance name priv-instance



Create new key pair and go to network setting edit select vpc ,subnet leave other settings by default and click on launch instance



Here your instance is created

The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar shows navigation links for VPC, IAM, S3, EC2, RDS, Route 53, and Simple Notification Service. The main content area displays a table titled 'Instances (1) Info' with one row. The row details a single instance named 'priv-instance' with the ID 'i-01b5c8fa5004acfad'. The instance is in the 'Running' state, t2.micro type, and Initializing status check. It is located in the 'us-east-2a' availability zone. A 'View alarms +' button is present. The top right of the table has buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. The bottom of the table has a 'Select an instance' dropdown.

Go to Mumbai region

The screenshot shows the AWS Management Console with the EC2 service selected. The left sidebar shows navigation links for VPC, IAM, S3, EC2, RDS, Route 53, and Simple Notification Service. The main content area displays a table titled 'Instances (1/2) Info' with two rows. The first row is selected and labeled 'jump server' with the ID 'i-051ee17c04018f5fa'. This instance is in the 'Running' state, t2.micro type, and 2/2 checks passed status check. It is located in the 'ap-south-1a' availability zone. The second row is labeled 'Application-se...' with the ID 'i-005b5749b13ce256f'. This instance is also in the 'Running' state, t2.micro type, and 2/2 checks passed status check. It is located in the 'ap-south-1b' availability zone. A 'View alarms +' button is present. The top right of the table has buttons for 'Connect', 'Instance state', 'Actions', and 'Launch instances'. The bottom of the table has a 'Select an instance' dropdown. Below the table, a detailed view for the selected instance 'i-051ee17c04018f5fa (jump server)' is shown, including tabs for Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags. The 'Details' tab is active, showing sections for Instance summary, Public IPv4 address (13.234.119.236), Instance state (Running), and Private IP DNS name (IPv4 only) (ip-172-25-3-253.ap-south-1.compute.internal).

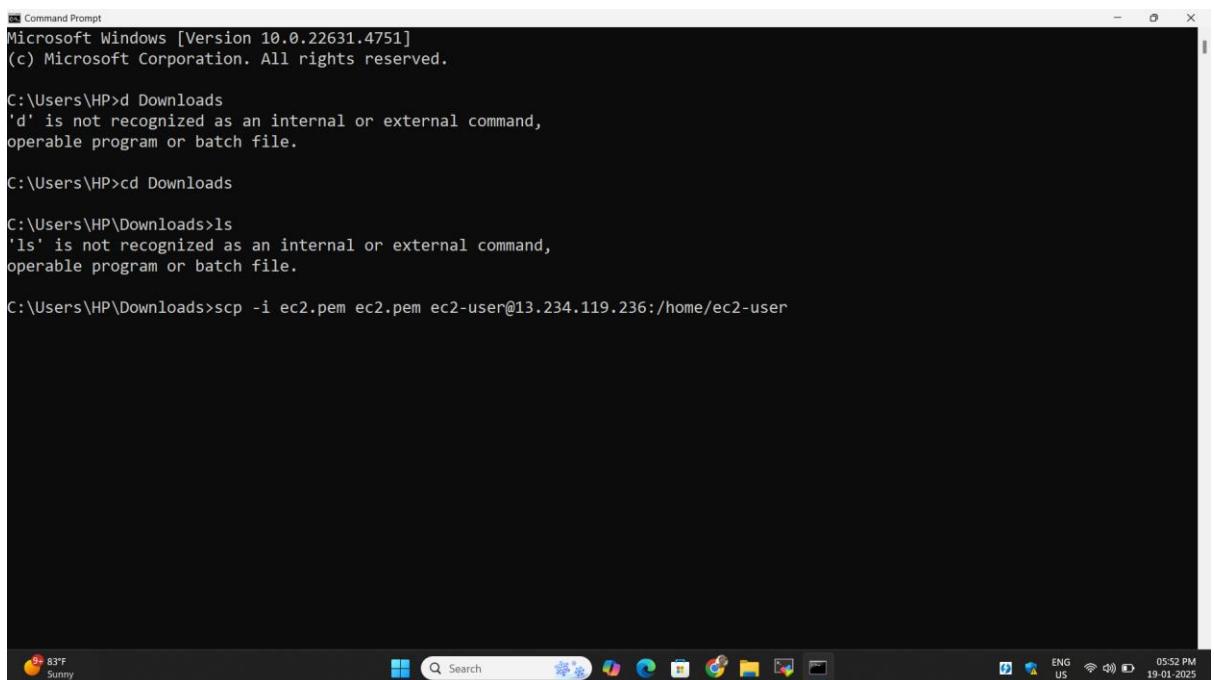
Now we will transfer the key file from your laptop to your public EC2 instance using Command Prompt

Navigate to the Key File Location Use the cd command to navigate to the folder where your key file

Use SCP to Transfer the Key File Run the following command to securely copy the key file to your public instance

```
scp -i "your-key.pem" private-key.pem ec2-user@<Public-Instance-Public-IP>:/home/ec2-user/
```

enter the command which will ask if you want to transfer file write yes/no write yes



```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

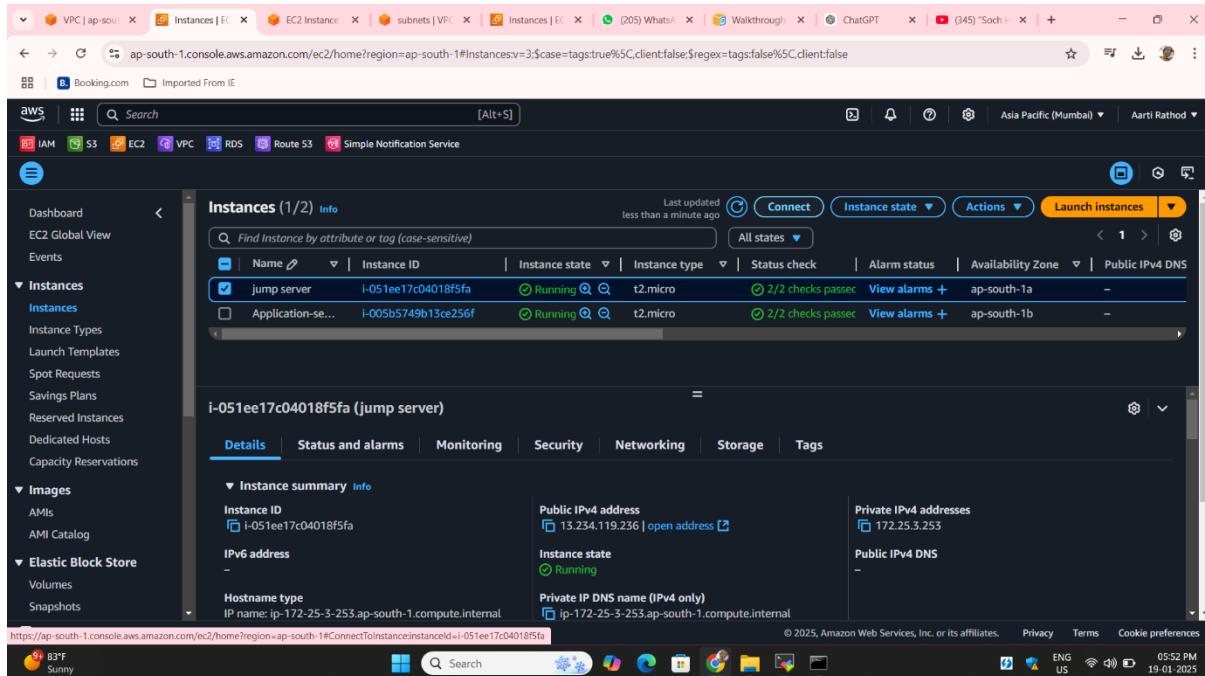
C:\Users\HP>d Downloads
'd' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\HP>cd Downloads

C:\Users\HP\Downloads>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\HP\Downloads>scp -i ec2.pem ec2.pem ec2-user@13.234.119.236:/home/ec2-user
```

Now Go to your instance select public instance click on connect

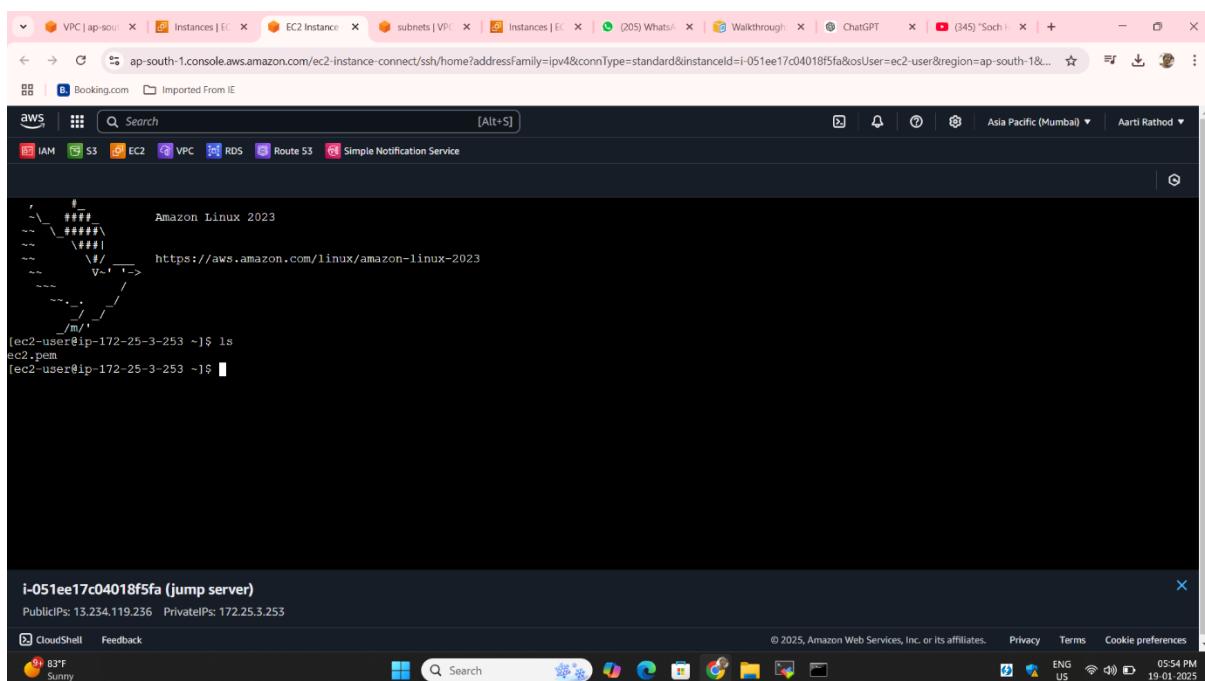


The screenshot shows the AWS EC2 Instances page. There are two instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
jump server	i-051ee17c04018f5fa	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1a	-
Application-se...	i-005b5749b13ce256f	Running	t2.micro	2/2 checks passed	View alarms	ap-south-1b	-

The 'jump server' instance is selected. Below the table, the instance details for 'i-051ee17c04018f5fa (jump server)' are shown. The 'Details' tab is selected, displaying information such as Public IP address (13.234.119.236), Instance state (Running), and Private IP DNS name (ip-172-25-3-253.ap-south-1.compute.internal).

Navigate to the /home/ec2-user/ directory and list the files to verify the transfer: ls You should see the transferred key file ec2.pem



The screenshot shows an AWS CloudShell terminal session. The terminal output shows the user navigating to the home directory of the ec2-user account and listing files:

```
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

(ec2-user@ip-172-25-3-253 ~]$ ls
ec2.pem
(ec2-user@ip-172-25-3-253 ~]$
```

At the bottom of the terminal window, it says "PublicIPs: 13.234.119.236 PrivateIPs: 172.25.3.253".

Now from here you have to take access from your public instance to your private instance Use the transferred key file to SSH into the private instance from the public instance:

```
ssh -i "private-key.pem" ec2-user@<Private-Instance-Private-I
```

```
(ec2-user@ip-172-25-3-253 ~]$ ls
private-key.pem

[ec2-user@ip-172-25-3-253 ~]$ ssh -i ec2.pem ec2-user@172.25.19.159
The authenticity of host '172.25.19.159 (172.25.19.159)' can't be established.
ED25519 key fingerprint is SHA256:027qf00rK3lmF3lGdxh0aqExphffgHijqcyjHGB5yK+k.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.25.19.159' (ED25519) to the list of known hosts.

Permissions 0644 for 'ec2.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "ec2.pem": bad permissions
ec2-user@172.25.19.159: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-172-25-3-253 ~]$ ssh -i ec2.pem ec2-user@172.25.19.159
i-051ee17c04018f5fa (jump server)
PublicIPs: 13.234.119.236 PrivateIPs: 172.25.3.253

CloudShell Feedback
83°F Sunny
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
06:03 PM 19-01-2025
```

Now give a private instance private ip with command `ssh -I ec2.pem ec2-user@172.25.19.159` after this they ask confirmation write yes there

Here your private instance access is get

```
(ec2-user@ip-172-25-3-253 ~]$ ls
private-key.pem

[ec2-user@ip-172-25-3-253 ~]$ ssh -i ec2.pem ec2-user@172.25.19.159
The authenticity of host '172.25.19.159 (172.25.19.159)' can't be established.
ED25519 key fingerprint is SHA256:027qf00rK3lmF3lGdxh0aqExphffgHijqcyjHGB5yK+k.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.25.19.159' (ED25519) to the list of known hosts.

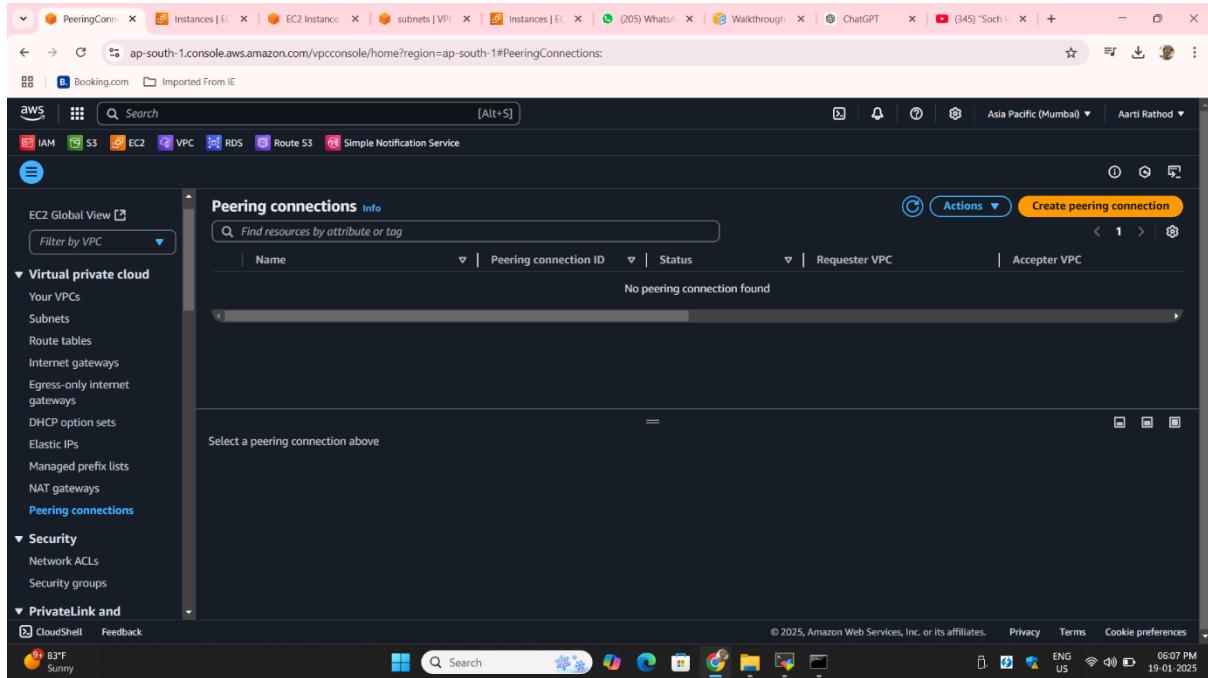
Permissions 0644 for 'ec2.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "ec2.pem": bad permissions
ec2-user@172.25.19.159: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-172-25-3-253 ~]$ ssh -i ec2.pem ec2-user@172.25.19.159
ssh: Could not resolve hostname ec2.pem: Name or service not known
[ec2-user@ip-172-25-3-253 ~]$ ssh -i ec2.pem ec2-user@172.25.19.159: /home/ec2-user
ssh: Could not resolve hostname 172.25.19.159: /home/ec2-user: Name or service not known
[ec2-user@ip-172-25-3-253 ~]$ sudo ssh -i ec2.pem ec2-user@172.25.19.159
The authenticity of host '172.25.19.159 (172.25.19.159)' can't be established.
ED25519 key fingerprint is SHA256:027qf00rK3lmF3lGdxh0aqExphffgHijqcyjHGB5yK+k.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.25.19.159' (ED25519) to the list of known hosts.

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

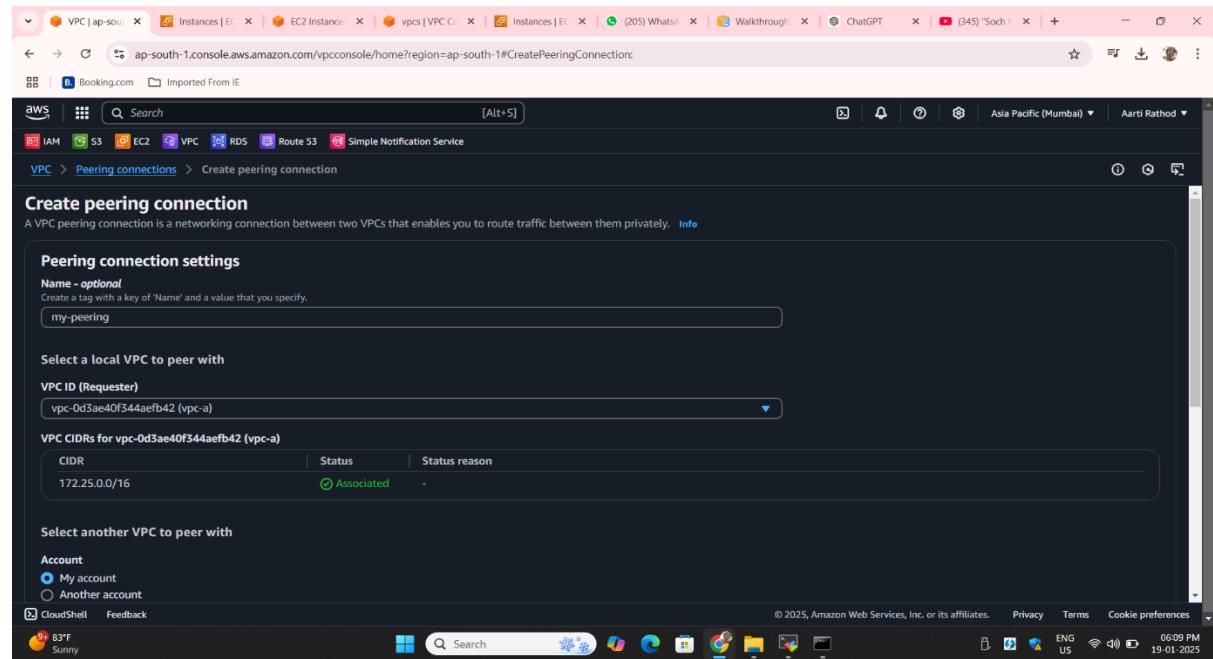
[ec2-user@ip-172-25-19-159 ~]$ i-051ee17c04018f5fa (jump server)
PublicIPs: 13.234.119.236 PrivateIPs: 172.25.3.253

CloudShell Feedback
83°F Sunny
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
06:04 PM 19-01-2025
```

VPC Peering is used to enable private communication between resources in two VPCs across the same or different regions. This allows instances in one VPC to access instances or services in another VPC without using public internet. On the left-hand menu, click Peering Connections. Click the Create Peering Connection button.



Name tag: Enter a name for the connection my- Peering Requester VPC:
Select the VPC you want to peer from vpc-a



Choose account we want to connect with ohio region so we will Enter the Account ID of the target VPC and select its VPC.Click Create Peering Connection.

Select another VPC to peer with

Account My account Another account

Region This Region (ap-south-1) Another Region
US East (Ohio) (us-east-2)

VPC ID (Acceptor)
vpc-072e9631fd1692336

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	my-peering

Add new tag

You can add 49 more tags.

Here you can see the status is pending

Peering connections (1) **Actions** **Create peering connection**

Name	Peering connection ID	Status	Requester VPC	Acceptor VPC
my-peering	pcc-086623e0cbd1f5778	Pending acceptance	vpc-0d3ae40f344aefb42 / vpc-a	vpc-072e9631fd1692336

Select a peering connection above

Go to ohio region peering connection

The screenshot shows the AWS VPC dashboard with the 'Peering connections' section selected. A single peering connection is listed:

Name	Peering connection ID	Status	Requester VPC	Acceptor VPC
-	pcc-086623e0cbd1f5778	Pending acceptance	vpc-0d3ae40f344aefb42	vpc-072e9631fd1692336 / vpc-b

The status is 'Pending acceptance'. The accepter VPC is 'vpc-072e9631fd1692336 / vpc-b'. The interface includes a search bar, a 'Create peering connection' button, and an 'Actions' dropdown.

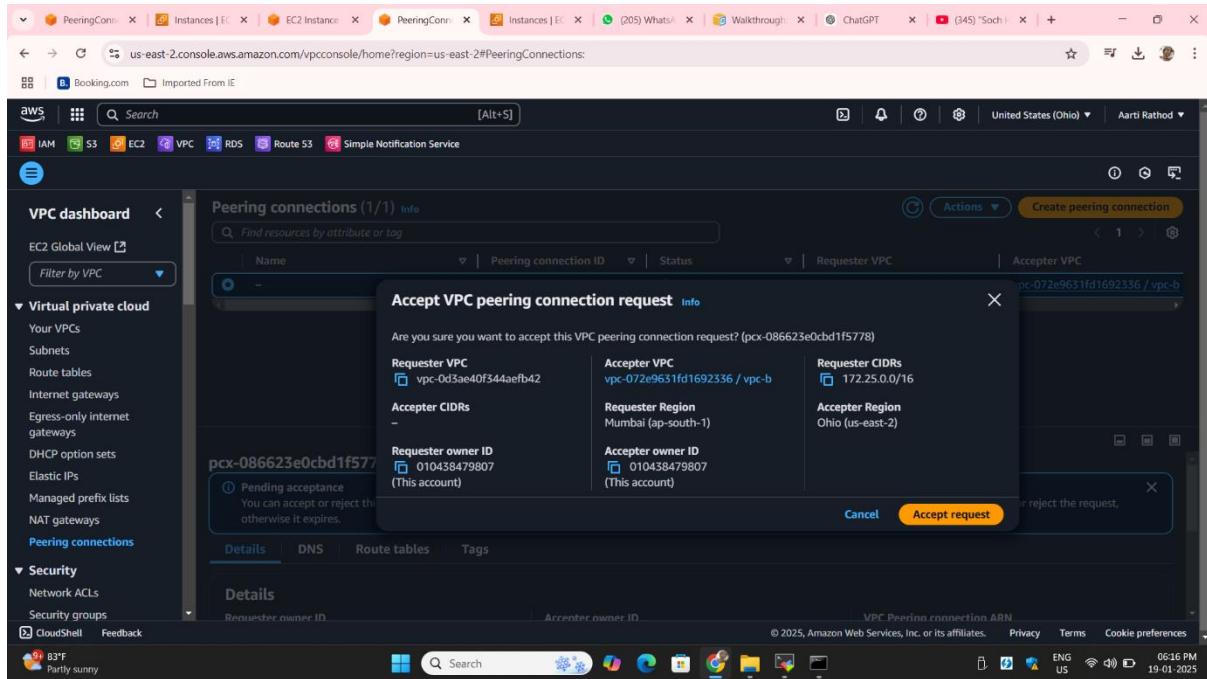
Select peering connection go to action click on accept request

The screenshot shows the same AWS VPC dashboard, but the context has shifted to a specific peering connection. The 'Actions' menu for the peering connection 'pcc-086623e0cbd1f5778' is open, displaying the following options:

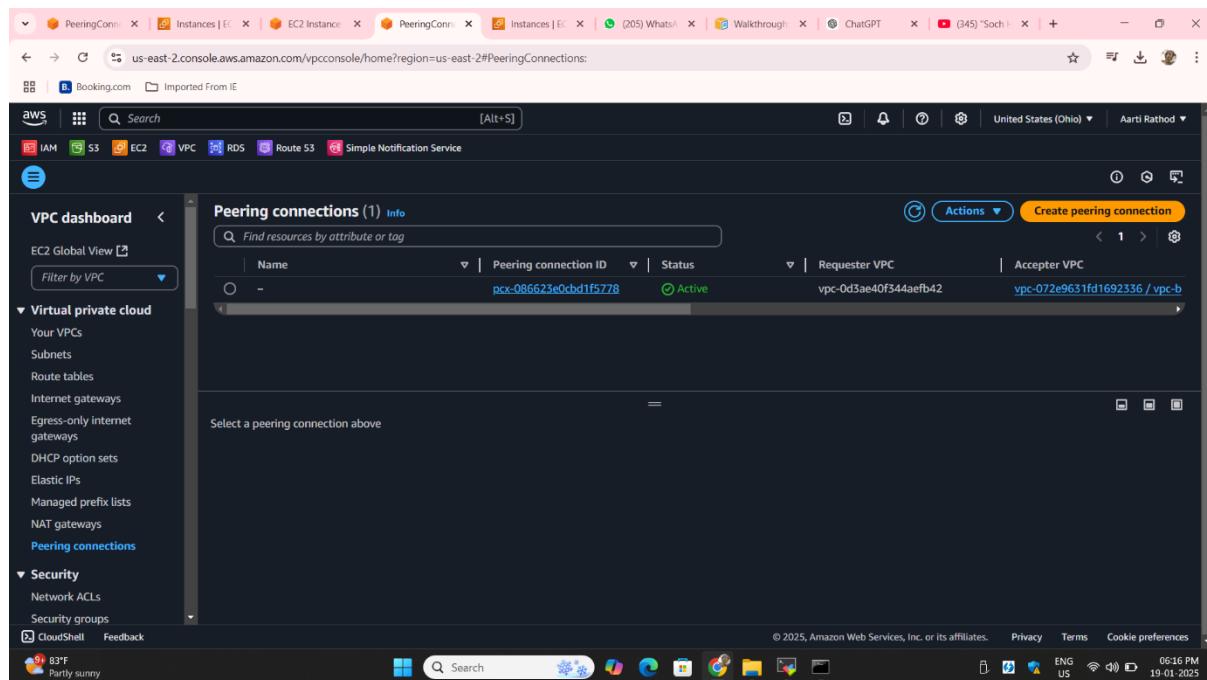
- View details
- Accept request** (highlighted)
- Reject request
- Edit DNS settings
- Manage tags
- Delete peering connection

A tooltip message indicates: 'You can accept or reject this peering connection request using the 'Actions' menu. You have until Sunday, January 26, 2025 at 18:09:41 GMT+5:30 to accept or reject the request; otherwise it expires.'

Click on accept request



Here the status is active of peering connection the connection is established between them.



In Mumbai region also status is active

The screenshot shows the AWS VPC Peering connections page. A single peering connection named "my-peering" is listed with the status "Active". The connection connects the Requester VPC (vpc-0d3ae40f344aefb42) and the Acceptor VPC (vpc-072e9631fd1692336).

Here you can see when I ping to private instance its ping

```
ssh: Could not resolve hostname 172.25.19.159: Name or service not known
[ec2-user@ip-172-25-3-253 ~]$ sudo ssh -i ec2.pem ec2-user@172.25.19.159
The authenticity of host '172.25.19.159 (172.25.19.159)' can't be established.
ED25519 key fingerprint is SHA256:027gf00rK3Dmf31GdjhOaqSYpHfqHjqjcyjHGB5yK+k.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '172.25.19.159' (ED25519) to the list of known hosts.

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-25-19-159 ~]$ ping 172.25.19.159
PING 172.25.19.159 (172.25.19.159) 56(84) bytes of data.
64 bytes from 172.25.19.159: icmp_seq=1 ttl=127 time=0.016 ms
64 bytes from 172.25.19.159: icmp_seq=2 ttl=127 time=0.026 ms
64 bytes from 172.25.19.159: icmp_seq=3 ttl=127 time=0.029 ms
^C
--- 172.25.19.159 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2095ms
rtt min/avg/max/mdev = 0.016/0.023/0.029/0.005 ms
[ec2-user@ip-172-25-19-159 ~]$ i-051ee17c04018f5fa (jump server)
PublicIPs: 13.234.119.236 PrivateIPs: 172.25.3.253
```

Now go to routes table select private rt go to routes then click edit routes.

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
priv-rt	rtb-0133580ced100ceac	-	-	No	vpc-0d3ae40f544aefb42 vp
	rtb-0e3b05b614ab31174	-	-	Yes	vpc-0a347c7c350f4837a
pub-rt	rtb-066550fe8f86f6fe5	subnet-0dc53a1c71f58ae...	-	Yes	vpc-0d3ae40f544aefb42 vp

Select the range of ohio region subnet paste here then select peering connection select peering connection click on save changes

Destination	Target	Status	Propagated
172.25.0.0/16	local	Active	No
192.168.0.0/20	Peering Connection ppx-086623e0cbd1f5778	-	No

Go to ohio region route table

The screenshot shows the AWS VPC Route Tables page. The left sidebar is titled 'VPC dashboard' and includes sections for EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables), Security (Network ACLs, Security groups), and CloudShell/Feedback. The main content area is titled 'Route tables (1/2) Info'. It lists two route tables: 'rtb-023a03b83d4dbfd04' and 'rtb-02a65acd342452b15'. The first route table is selected. Below it, the details for 'rtb-023a03b83d4dbfd04' are shown, including its Route Table ID, Main status (Yes), and Owner ID. The bottom navigation bar includes links for Details, Routes, Subnet associations, Edge associations, Route propagation, and Tags.

Select route and go to edit routes

This screenshot is identical to the one above, showing the AWS VPC Route Tables page. The 'Routes' tab is now selected under the 'rtb-023a03b83d4dbfd04' details view. A single route entry is listed: 'Destination: 192.168.0/16, Target: local, Status: Active, Propagated: No'. To the right of this table, there are buttons for 'Both' and 'Edit routes'.

Cpy private subnet cidr range of Mumbai subnet destination select peering connection select your peering connection and click on save changes

Now go to Mumbai ec2 instance go to security select security group

Click on edit inbound rules

The screenshot shows the AWS EC2 Security Groups console. The left sidebar has sections for Instances, Images, Elastic Block Store, and Network & Security. The main area displays the details for the security group 'sg-00902e728dc217d3e - launch-wizard-9'. The 'Inbound rules' tab is selected, showing a table with one rule:

Name	Security group rule ID	IP version	Type	Protocol	Port range
-	sgr-0f5dcc08416cdcf0	IPv4	SSH	TCP	22

Click add rules select ALL ICMP-IPV4 select source anywhere click on save rules

The screenshot shows the 'Edit inbound rules' dialog box. It lists two rules:

- SSH rule: Type: SSH, Protocol: TCP, Port range: 22, Source: Custom (0.0.0.0/0)
- ICMP rule: Type: All ICMP - IPv4, Protocol: ICMP, Port range: All, Source: Anywhere (0.0.0.0/0)

At the bottom right, there are 'Cancel', 'Preview changes', and 'Save rules' buttons. The 'Save rules' button is highlighted.

Same thing we have to do with ohio region instance select the instance go to security

The screenshot shows the AWS EC2 Instances page. The left sidebar includes options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and Elastic Block Store. The main content area displays a table for 'Instances (1/1)'. The table has columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. One row is selected, showing 'priv-instance' with Instance ID i-01b5c8fa5004acfad, State Running, Type t2.micro, Status Initializing, Availability Zone us-east-2a, and Public IP 192.168.5.47. Below the table, the instance details for 'i-01b5c8fa5004acfad (priv-instance)' are shown, including Details, Status and alarms, Monitoring, Security, Networking, Storage, and Tags tabs. Under the Details tab, the Instance summary section shows Instance ID i-01b5c8fa5004acfad, Public IPv4 address -, Instance state Running, and Private IPv4 addresses 192.168.5.47.

Edit inbound rules

The screenshot shows the AWS EC2 Security Groups page. The left sidebar includes EC2 (selected) and Security Groups. The main content area shows the details for 'sg-09357e4ec7e51a02d - launch-wizard-1'. It lists the Security group name (launch-wizard-1), Security group ID (sg-09357e4ec7e51a02d), Owner (010438479807), Description (launch-wizard-1 created 2025-01-19T12:02:55.42Z), and VPC ID (vpc-072e9631fd1692336). Below this, there are tabs for Inbound rules, Outbound rules, Sharing - new, VPC associations - new, and Tags. The Inbound rules section shows a table with one entry: Name -, Security group rule ID sgr-087e4916525888cc1, IP version IPv4, Type SSH, Protocol TCP, and Port range 22.

Click add rule type select All ICMP –IPV4 source anywhere and then click on save rules

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-087e4916525888cc1	SSH	TCP	22	Custom 0.0.0.0/0	
-	All ICMP - IPv4	ICMP	All	Anywhere 0.0.0.0/0	

Add rule

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel Preview changes Save rules

Go to Mumbai route table pub-rt

Name	Route table ID	Explicit subnet assoc...	Main	VPC
priv-rt	rtb-0133580ced100ceak	-	-	vpce-0d5ae40f544aefb42 vp...
-	rtb-0e3b05b614ab31174	-	-	vpce-0a347c7c550f4837a
<input checked="" type="checkbox"/> pub-rt	rtb-066550fe8f86f6fe6	subnet-0dc53a1c71f58ae5d	-	vpce-0d3ae40f544aefb42 vp...

rtb-066550fe8f86f6fe6 / pub-rt

Details Routes Subnet associations Edge associations Route propagation Tags

Details

Route table ID rtb-066550fe8f86f6fe6	Main <input checked="" type="checkbox"/> Yes	Explicit subnet associations subnet-0dc53a1c71f58ae5d / pub-subnet	Edge associations -
VPC	Owner ID		

**Go to route add one route which is ohio region subnet CIDR range
sores peering connection and save changes**

Name	Route table ID	Explicit subnet associations	Main	VPC
priv-rt	rtb-0133580ced100ce4	-	No	vpc-0d3ae40f344aefb42 vpc-a
-	rtb-0e3b05b614pb31174	-	Yes	vpc-0a347c7c350f4837a
pub-rt	rtb-066550fe8f86f6fe6	subnet-0dc53a1c71f58ae...	-	vpc-0d3ae40f344aefb42 vpc-a

Here you have successfully route the subnet

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0671c149c05f753ae	Active	No
172.25.0.0/16	local	Active	No
192.168.0.0/20	pcv-086623e0cbd1f5778	Active	No

Here what we have to copy ohio region subnet Cidr range 192.168.0.0/20

The screenshot shows the AWS VPC dashboard with the 'Subnets' section selected. There are four subnets listed:

Subnet ID	State	VPC	Block Public Access	IPv4 CIDR	IPv6 CIDR
subnet-0237e6796f548c05d	Available	vpc-02af05c4a2e06e31	Off	172.31.32.0/20	-
subnet-0d91bf1b9de4a7ccb	Available	vpc-02af05c4a2e06e31	Off	172.31.0/20	-
subnet-0dafaec724fe1d22	Available	vpc-02af05c4a2e06e31	Off	172.31.16.0/20	-
subnet-0880c0dedbf15ac6e	Available	vpc-072e9631fd692335 vpc-b	Off	192.168.0.0/20	-

Below the table, the details for the subnet-0880c0dedbf15ac6e are shown. The CIDR range is explicitly mentioned as 192.168.0.0/20.

Here I take access of database instance which we have present in ohio region with ssh command after that we take access from Mumbai region private instace this way your database is secure

```

64 bytes from 172.25.19.159: icmp_seq=1 ttl=127 time=0.017 ms
64 bytes from 172.25.19.159: icmp_seq=2 ttl=127 time=0.028 ms
64 bytes from 172.25.19.159: icmp_seq=3 ttl=127 time=0.027 ms
64 bytes from 172.25.19.159: icmp_seq=4 ttl=127 time=0.029 ms
^C
--- 172.25.19.159 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3085ms
rtt min/avg/max/mdev = 0.017/0.025/0.029/0.004 ms
[ec2-user@ip-172-25-19-159 ~]$ ping 192.168.5.47
PING 192.168.5.47 (192.168.5.47) 56(84) bytes of data.
^C
--- 192.168.5.47 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3139ms
[ec2-user@ip-172-25-19-159 ~]$ ping 192.168.5.47
PING 192.168.5.47 (192.168.5.47) 56(84) bytes of data.
64 bytes from 192.168.5.47: icmp_seq=1 ttl=127 time=194 ms
64 bytes from 192.168.5.47: icmp_seq=2 ttl=127 time=194 ms
64 bytes from 192.168.5.47: icmp_seq=3 ttl=127 time=194 ms
64 bytes from 192.168.5.47: icmp_seq=4 ttl=127 time=193 ms
64 bytes from 192.168.5.47: icmp_seq=5 ttl=127 time=193 ms
64 bytes from 192.168.5.47: icmp_seq=6 ttl=127 time=194 ms
^C
--- 192.168.5.47 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5003ms
rtt min/avg/max/mdev = 193.196/193.604/193.964/0.265 ms
[ec2-user@ip-172-25-19-159 ~]$ 

```

I-051ee17c04018f5fa (jump server)

PublicIPs: 13.234.119.236 PrivateIPs: 172.25.3.253

Here when you pin the ip of database instance its ping

```
4 packets transmitted, 4 received, 0% packet loss, time 3085ms
rtt min/avg/max/mdev = 0.017/0.025/0.029/0.004 ms
[ec2-user@ip-172-25-19-159 ~]$ ping 192.168.5.47
PING 192.168.5.47 (192.168.5.47) 56(84) bytes of data.
^C
--- 192.168.5.47 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3139ms

[ec2-user@ip-172-25-19-159 ~]$ ping 192.168.5.47
PING 192.168.5.47 (192.168.5.47) 56(84) bytes of data.
64 bytes from 192.168.5.47: icmp_seq=1 ttl=127 time=194 ms
64 bytes from 192.168.5.47: icmp_seq=2 ttl=127 time=194 ms
64 bytes from 192.168.5.47: icmp_seq=3 ttl=127 time=194 ms
64 bytes from 192.168.5.47: icmp_seq=4 ttl=127 time=193 ms
64 bytes from 192.168.5.47: icmp_seq=5 ttl=127 time=193 ms
64 bytes from 192.168.5.47: icmp_seq=6 ttl=127 time=194 ms
^C
--- 192.168.5.47 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5003ms
rtt min/avg/max/mdev = 193.196/193.604/193.964/0.265 ms
[ec2-user@ip-172-25-19-159 ~]$ yum install httpd
Error: This command has to be run with superuser privileges (under the root user on most systems).
[ec2-user@ip-172-25-19-159 ~]$ sudo yum install httpd
*Amazon Linux 2023 repository
[1]+  Stopped                  sudo yum install httpd
[ec2-user@ip-172-25-19-159 ~]$ yum update httpd
Error: This command has to be run with superuser privileges (under the root user on most systems).
[ec2-user@ip-172-25-19-159 ~]$ sudo yum update httpd
Waiting for process with pid 5591 to finish.
^CKeyboardInterrupt: Terminated.
[ec2-user@ip-172-25-19-159 ~]$
```