

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.

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only ☐ VPC and more

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

VPC-A

IPv4 CIDR block [Info](#)
☒ IPv4 CIDR manual input
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.100.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block [Info](#)
☒ No IPv6 CIDR block
☐ IPAM-allocated IPv6 CIDR block

Activate Windows
Go to Settings to activate Windows.

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

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

VPC > Internet gateways > Create internet gateway

Create internet gateway [Info](#)

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

Internet gateway settings

Name tag
Creates a tag with a key of 'Name' and a value that you specify.

VPC-A-IGW

Tags - optional

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	VPC-A-IGW	Remove

[Add new tag](#)

You can add 49 more tags.

[Cancel](#) [Create internet gateway](#)

Activate Windows
Go to Settings to activate Windows.

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

VPC > Internet gateways > Attach to VPC (igw-08630ec7896160ed0)

✓ The following internet gateway was created: igw-08630ec7896160ed0 - VPC-A-IGW. You can now attach to a VPC to enable the VPC to communicate with the internet. [Attach to a VPC](#)

Attach to VPC (igw-08630ec7896160ed0) [Info](#)

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

vpc-05699c0fc87f09a1a

[AWS Command Line Interface command](#)

[Cancel](#) [Attach internet gateway](#)

Activate Windows
Go to Settings to activate Windows.

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.

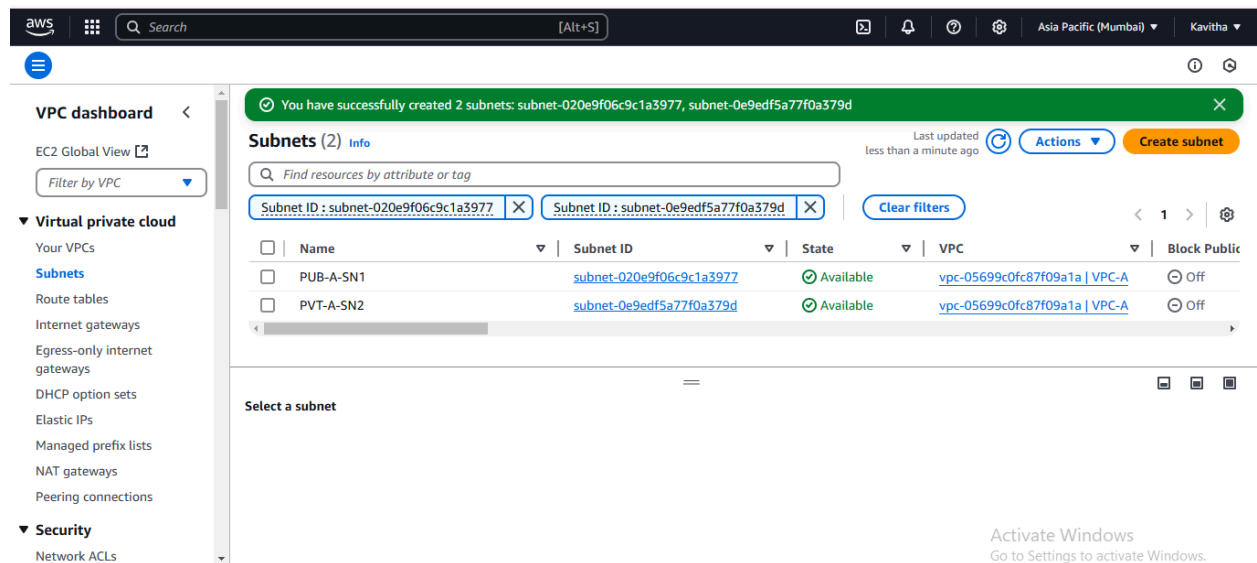
- **Private Subnet A:**

- 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.

The screenshot shows the 'Create subnet' page in the AWS Management Console. The breadcrumb navigation is 'VPC > Subnets > Create subnet'. The page title is 'Subnet 1 of 1'. The 'Subnet name' field is 'PUB-A-SN1'. The 'Availability Zone' is 'Asia Pacific (Mumbai) / ap-south-1a'. The 'IPv4 VPC CIDR block' is '10.100.0.0/16'. The 'IPv4 subnet CIDR block' is '10.100.0.0/24', with a note '256 IPs'. The 'Tags - optional' section shows a tag with key 'Name' and value 'PUB-A-SN1'. A 'Remove' button is next to the tag. A watermark 'Activate Windows' is visible in the bottom right corner.

The screenshot shows the 'Create subnet' page in the AWS Management Console for 'Subnet 2 of 2'. The breadcrumb navigation is 'VPC > Subnets > Create subnet'. The 'Subnet name' field is 'PVT-A-SN2'. The 'Availability Zone' is 'Asia Pacific (Mumbai) / ap-south-1b'. The 'IPv4 VPC CIDR block' is '10.100.0.0/16'. The 'IPv4 subnet CIDR block' is '10.100.1.0/24'. The 'Tags - optional' section shows a tag with key 'Name' and value 'PVT-A-SN2'. A 'Remove' button is next to the tag. A watermark 'Activate Windows' is visible in the bottom right corner.

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

Q Name X

Value - optional

Q PUB-A-RT X Remove

Add new tag

You can add 49 more tags.

Activate Windows
Go to Settings to activate Windows.

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
Q 0.0.0.0/0 X	Q local X		
	Internet Gateway	-	No
	Q igw-08630ec7896160ed0 X		

Add route

Remove

Cancel Preview Save changes

Activate Windows
Go to Settings to activate Windows.

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
<input checked="" type="checkbox"/>	PUB-A-SN1	subnet-020e9f06c9c1a3977	10.100.0.0/24	-	Main (rtb-08e9f792fe09224aa)
<input type="checkbox"/>	PVT-A-SN2	subnet-0e9edf5a77f0a379d	10.100.1.0/24	-	Main (rtb-08e9f792fe09224aa)

Selected subnets

subnet-020e9f06c9c1a3977 / PUB-A-SN1 X

Cancel Save associations

aws

Search

[Alt+S]

Asia Pacific (Mumbai)

Kavitha

[VPC](#) > [Route tables](#) > Create route table

Create route table

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.

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**

Remove

Add new tag

You can add 49 more tags.

Activate Windows

Go to Settings to activate Windows.

aws

Search

[Alt+S]

Asia Pacific (Mumbai)

Kavitha

[VPC](#) > [Route tables](#) > [rtb-044b2eda4f9b2c83e](#) > 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
<input type="checkbox"/>	PUB-A-SN1	subnet-020e9f06c9c1a3977	10.100.0.0/24	-	rtb-0a403f8d35fcd8a31 / PUB-A-RT
<input checked="" type="checkbox"/>	PVT-A-SN2	subnet-0e9edf5a77f0a379d	10.100.1.0/24	-	Main (rtb-08e9f792fe09224aa)

Selected subnets

subnet-0e9edf5a77f0a379d / PVT-A-SN2

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](#) > [Your VPCs](#) > Create VPC

Create VPC

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create **Info**
Create only the VPC resource or the VPC and other networking resources.

☒ VPC only ☐ VPC and more

Name tag - optional
Creates a tag with a key of 'Name' and a value that you specify.

IPv4 CIDR block **Info**

☒ IPv4 CIDR manual input ☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR

CIDR block size must be between /16 and /28.

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.

Activate Windows
Go to Settings to activate Windows.

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

< > ^ v

Activate Windows
Go to Settings to activate Windows.

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.

PVT-B-RT

VPC
The VPC to use for this route table.

vpc-099d6796d5c7e9b35 (VPC-B)

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**

Q Name X Q PVT-B-RT X Remove

Add new tag

You can add 49 more tags.

Activate Windows
Go to Settings to activate Windows.

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

VPC > Route tables > rtb-0a4fa9935f4f155db > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/1)

Filter subnet associations

<input checked="" type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/>	PVT-B-SN	subnet-01596ed1493e60921	10.200.1.0/24	-	rtb-0a4fa9935f4f155db / PVT-B-RT

Selected subnets

subnet-01596ed1493e60921 / PVT-B-SN X

Cancel Save associations

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

EC2 > Instances > Launch an instance

Network settings [Info](#)

VPC - required [Info](#)

vpc-05699c0fc87f09a1a (VPC-A)
10.100.0.0/16

Subnet [Info](#)

subnet-020e9f06c9c1a3977 PUB-A-SN1
VPC: vpc-05699c0fc87f09a1a Owner: 476114123923
Availability Zone: ap-south-1a Zone type: Availability Zone
IP addresses available: 251 CIDR: 10.100.0.0/24

[Create new subnet](#)

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

Security group name - required

launch-wizard-1

Summary

Number of instances [Info](#)

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...[read more](#)
ami-0ddfb243cbee3768

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel Launch instance

Activate Windows
Go to Settings to activate Windows.

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

Search

[Alt+S]

Asia Pacific (Mumbai)

Kavitha

< >

EC2 > Instances > Launch an instance

?

▼ Network settings Info

VPC - required Info

vpc-05699c0fc87f09a1a (VPC-A)
10.100.0.0/16

Subnet Info

subnet-0e9edf5a77f0a379d PVT-A-SN2
VPC: vpc-05699c0fc87f09a1a Owner: 476114123923
Availability Zone: ap-south-1b Zone type: Availability Zone
IP addresses available: 251 CIDR: 10.100.1.0/24

Auto-assign public IP Info

Disable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Security group name - required

launch-wizard-2

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _./@#,%&!+~\$*

▼ Summary

Number of instances Info

1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-0dddba243cbec3768

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Cancel

Launch instance

Activate Windows Go to Settings to activate Windows.

Preview code

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

[illegible]

EC2 Instance created successfully

aws

Search

[Alt+S]

Asia Pacific (Mumbai)

Kavitha

EC2

Instances

EC2

Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Save as Plan

Instances (1/3) Info

Last updated 2 minutes ago

Connect

Instance state

Actions

Launch instances

Find Instance by attribute or tag (case-sensitive)

All states

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
<input checked="" type="checkbox"/>	EC2-A-PRIVATE	i-0ffe5cd9660812b1a	Running	t2.micro	Initializing	View alarms +	ap-south-1l
<input type="checkbox"/>	EC2-B PRIVATE	i-0ef08e5c5bdd303cf	Running	t2.micro	Initializing	View alarms +	ap-south-1l
<input type="checkbox"/>	EC2-A-PUBLIC	i-08a2a8bdc65d57243	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1z

i-0ffe5cd9660812b1a (EC2-A-PRIVATE)

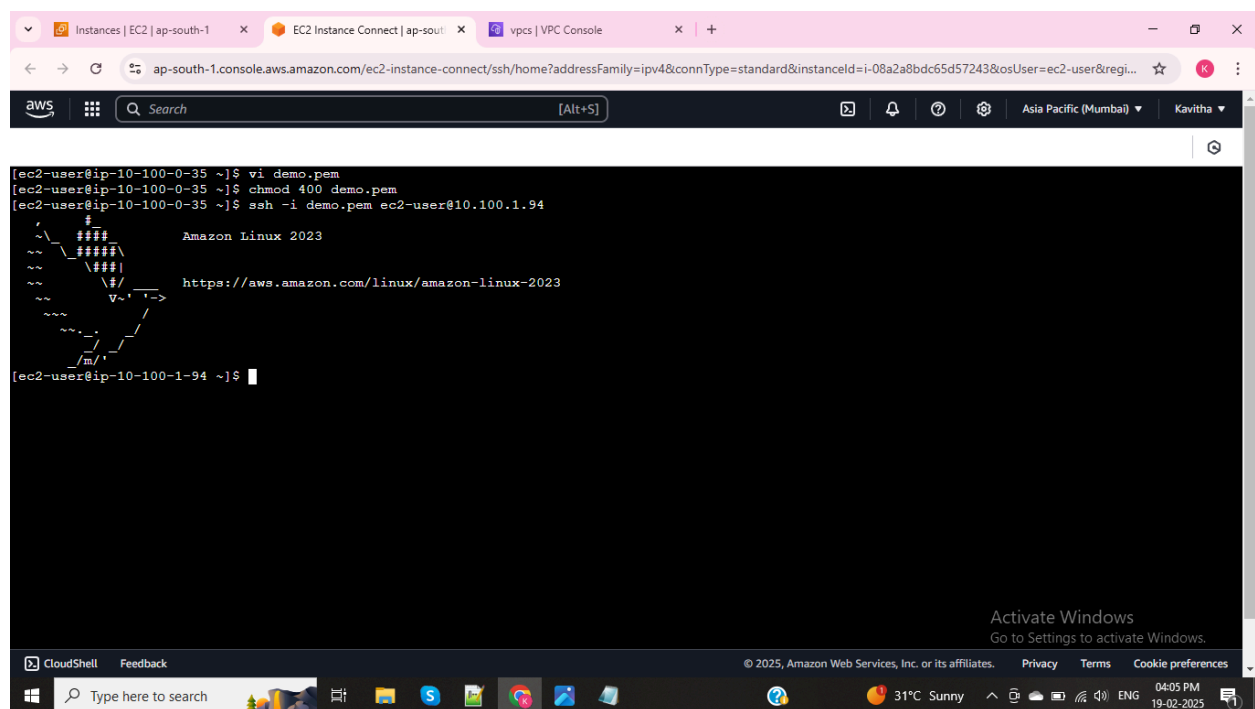
To enable secure communication between instances in a EC2 A Public to EC2 A private:

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)**

A screenshot of the AWS CloudShell interface. The terminal window shows a series of commands and their outputs. The user is logged in as 'ec2-user' on an instance with IP '10.100.0-35'. They create a file 'demo.pem', set permissions to '400', and then execute an SSH command to connect to another instance with IP '10.100.1-94'. The terminal output shows the SSH connection process, including the display of the Amazon Linux 2023 logo and the URL 'https://aws.amazon.com/linux/amazon-linux-2023'. The terminal window is titled 'ec2-user@ip-10-100-0-35 ~]' and the command prompt is '~]\$. The browser tabs at the top show 'Instances | EC2 | ap-south-1', 'EC2 Instance Connect | ap-south-1', and 'vpcs | VPC Console'. The AWS logo and search bar are visible in the top left of the terminal window. The bottom of the terminal window shows the 'Activate Windows' watermark and the 'CloudShell' logo.

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

Create peering connection

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. [Info](#)

Peering connection settings

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

MY VPC A - B

Select a local VPC to peer with

VPC ID (Requester)
vpc-05699c0fc87f09a1a (VPC-A)

VPC CIDRs for vpc-05699c0fc87f09a1a (VPC-A)

CIDR	Status	Status reason
10.100.0.0/16	Associated	-

Select another VPC to peer with

Account
My account

Activate Windows
Go to Settings to activate Windows.

Create peering connection

Account
☒ My account
☐ Another account

Region
☒ This Region (ap-south-1)
☐ Another Region

VPC ID (Acceptor)
vpc-099d6796d5c7e9b35 (VPC-B)

VPC CIDRs for vpc-099d6796d5c7e9b35 (VPC-B)

CIDR	Status	Status reason
10.200.0.0/16	Associated	-

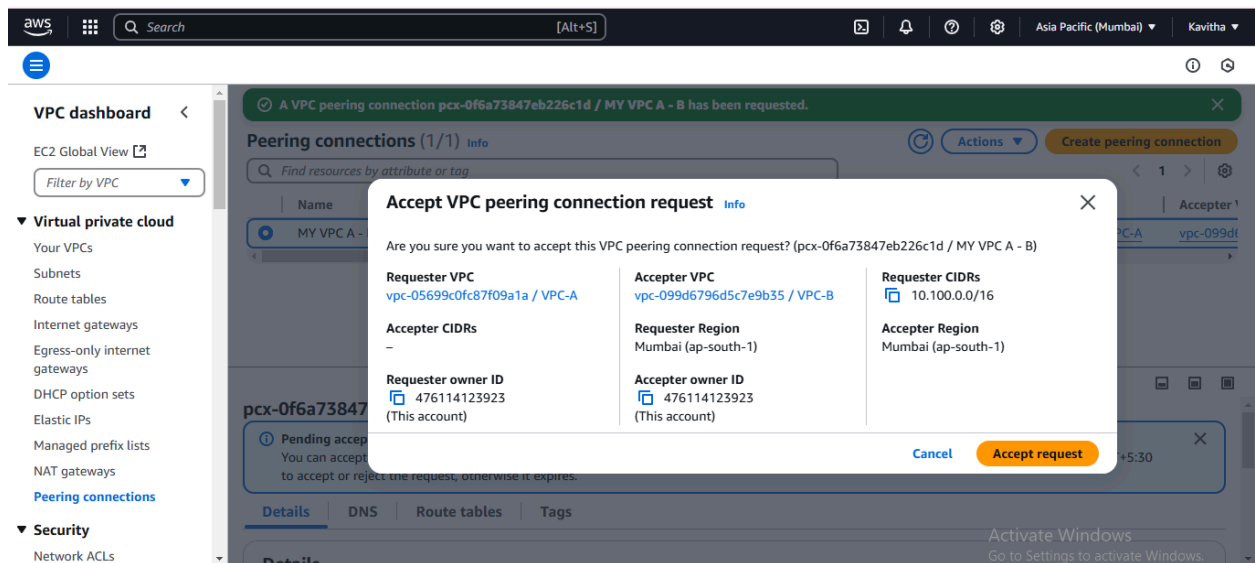
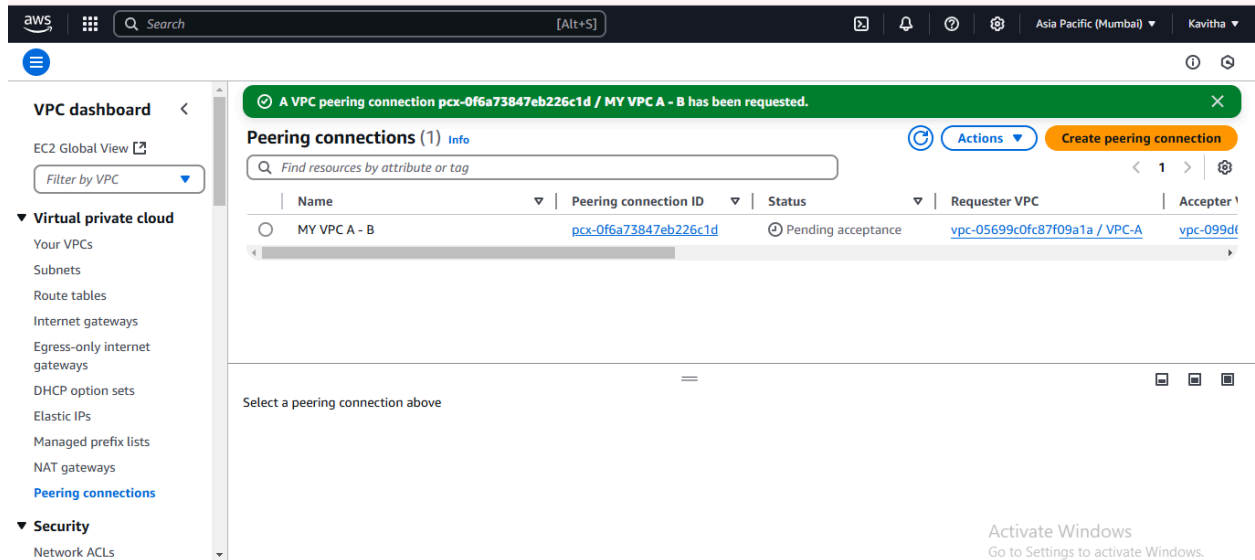
Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key
Name

Value - optional
MY VPC A - B

[Add new tag](#) [Remove](#)

Activate Windows
Go to Settings to activate Windows.



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.

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

VPC > Route tables > rtb-044b2eda4f9b2c83e > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.100.0.0/16	local	Active	No
10.200.1.0/24	local		
	Peering Connection	Active	No
	pcx-0f6a73847eb226c1d		

Add route

Cancel Preview Save changes

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

VPC > Route tables > rtb-0a4fa9935f4f155db > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.200.0.0/16	local	Active	No
10.100.1.0/24	local		
	Peering Connection	-	No
	pcx-0f6a73847eb226c1d		

Add route

Cancel Preview Save changes

Activate Windows
Go to Settings to activate Windows.

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.

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

EC2 > Security Groups > sg-0ec5d40adc821ef95 - launch-wizard-3 > Edit inbound rules

Edit inbound rules [Info](#)

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules [Info](#)

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-0eac2a18a5e0cbbcf	SSH	TCP	22	Cus... 10.100.1.0/24		Delete
-	All ICMP - IPv4	ICMP	All	Cus... 10.100.1.0/24		Delete

[Add rule](#)

Cancel [Preview changes](#) [Save rules](#)

Activate Windows

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 Acceptor 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)

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

~$ ssh ec2-user@10.200.1.113
The authenticity of host '10.200.1.113 (10.200.1.113)' can't be established.
ED25519 key fingerprint is SHA256:4sHfGSXw9aPNp2v8hmlnwfynox2AoITeg6xIAzH7E2w.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.200.1.113' (ED25519) to the list of known hosts.
ec2-user@10.200.1.113: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-100-1-94 ~]$ vi demo.pem
[ec2-user@ip-10-100-1-94 ~]$ vi demo.pem
[ec2-user@ip-10-100-1-94 ~]$ chmod 400 demo.pem
[ec2-user@ip-10-100-1-94 ~]$ ssh -i demo.pem ec2-user@10.200.1.113

~$ ssh ec2-user@10.200.1.113
The authenticity of host '10.200.1.113 (10.200.1.113)' can't be established.
ED25519 key fingerprint is SHA256:4sHfGSXw9aPNp2v8hmlnwfynox2AoITeg6xIAzH7E2w.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.200.1.113' (ED25519) to the list of known hosts.
ec2-user@10.200.1.113: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-100-1-94 ~]$ vi demo.pem
[ec2-user@ip-10-100-1-94 ~]$ vi demo.pem
[ec2-user@ip-10-100-1-94 ~]$ chmod 400 demo.pem
[ec2-user@ip-10-100-1-94 ~]$ ssh -i demo.pem ec2-user@10.200.1.113

~$ ssh ec2-user@10.200.1.113
The authenticity of host '10.200.1.113 (10.200.1.113)' can't be established.
ED25519 key fingerprint is SHA256:4sHfGSXw9aPNp2v8hmlnwfynox2AoITeg6xIAzH7E2w.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.200.1.113' (ED25519) to the list of known hosts.
ec2-user@10.200.1.113: Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
[ec2-user@ip-10-100-1-94 ~]$
```

Activate Windows
Go to Settings to activate Windows.