

VPC Peering

VPC Peering refers to the networking connection between two Virtual Private Clouds (VPCs) in cloud computing environments, such as those provided by Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure. VPCs are isolated network environments within the cloud that allow users to launch and run resources, such as virtual machines, in a logically isolated section of the cloud.

VPC Peering enables the direct communication between instances in different VPCs as if they were on the same network. This connection is established over the internal network of the cloud provider, allowing for secure and efficient communication between resources in separate VPCs.

Key points about VPC Peering:

- Cross-VPC Communication:** VPC Peering allows resources in one VPC to communicate with resources in another VPC using private IP addresses.
- Security:** VPC Peering does not involve a gateway device or VPN connection. It is a direct connection between VPCs over the cloud provider's network infrastructure, ensuring secure communication.
- Transitive Peering:** In some cloud providers, VPC Peering can be transitive, meaning if VPC A is peered with VPC B and VPC B is peered with VPC C, then VPC A can communicate with VPC C through the peering connections.
- Limitations:** There might be limitations on certain configurations, such as overlapping IP address ranges or restrictions imposed by the cloud provider.

VPC Peering is commonly used to facilitate collaboration between different business units, share resources across projects, or enable communication between applications hosted in separate VPCs while maintaining a level of network isolation and security.

To Begin with the Lab

Step 1: Create VPC

- In this lab you are going to create a new VPC. This VPC will be created in a new region let say Singapore. You can choose any region of your choice.
- Switch to Singapore region and then navigate to VPC and click on create VPC.



The screenshot shows the AWS VPC console interface. At the top, it says "Your VPCs (1)" and has an "Info" link. Below that is a search bar with the placeholder "Find resources by attribute or tag". Underneath is a table with columns: Name, VPC ID, State, and IPv4 CIDR. One row is visible: "vpc-0ad8ec0e7bbdc2893" with state "Available" and CIDR "172.31.0.0/16". At the bottom right of the table, there is a "Create VPC" button with a hand cursor icon pointing at it.

- Select VPC only and give it a name, then IPv4 as your CIDR block.
- Give a different CIDR for this region. The CIDR cannot be same for both of the regions.

5. Keep the tendency to default and then click on create VPC.

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

10.1.0.0/16



IPv6 CIDR block [Info](#)

No IPv6 CIDR block

IPAM-allocated IPv6 CIDR block

Amazon-provided IPv6 CIDR block

IPv6 CIDR owned by me

6. Now go onto subnets and create a new subnet.

7. Then select your VPC.

VPC ID

Create subnets in this VPC.

vhc-0ad513a33a2627f03



Associated VPC CIDRs

IPv4 CIDRs

10.1.0.0/16

8. Now give a name to your subnet then select your availability.
9. Now give it a CIDR block. Then choose to create a subnet.

Subnet 1 of 1

Subnet name

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

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.



IPv4 CIDR block [Info](#)



▼ Tags - optional

Key

Value - optional



You can add 49 more tags.

10. When the subnet is created then go to Internet gateways (IGT) and create an Internet gateway.
11. Now give it a name and create it.
12. Then you need to attach this internet gateway to your newly created VPC.

Internet gateways (1/1) [Info](#)

<input checked="" type="checkbox"/>	Name	Internet gateway ID	State	VPC ID
<input checked="" type="checkbox"/>	-	igw-0809722ec99f6a2d5	<input checked="" type="checkbox"/> Attached	vpc-0ad8ec0e7bbdc2893

Attach to VPC (igw-081e0739a919b8f00) [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.



▶ AWS Command Line Interface command

13. Then go to the route table and create a new route table.
14. Give it a name then select your VPC and create your route table.

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.

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
<input type="text" value="Name"/>	<input type="text" value="stagingvpc-public-route-table"/>	<input type="button" value="Remove"/>

You can add 49 more tags.

15. Once your route table is created then navigate to subnet association and associate your public subnet to it.

Subnets without explicit associations (1)

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Edit subnet association		Find subnet association	<	1	>	⚙️
<input type="button" value="Edit subnet association"/>	<input type="text"/>					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR			
appsubnet	subnet-0354cbbb41a5244a8	10.1.0.0/24	-			

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/1)						Find subnet associations	<	1	>	⚙️
<input checked="" type="checkbox"/>	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID					
<input checked="" type="checkbox"/>	appsubnet	subnet-0354cbbb41a5244a8	10.1.0.0/24	-	Main (rtb-06cb8bfd92da51f1)					

Selected subnets		Save associations
<input type="text" value="subnet-0354cbbb41a5244a8 / appsubnet"/>		<input style="background-color: orange; color: white; border: 1px solid orange;" type="button" value="Save associations"/>

16. Once it is done then go to routes and click on edit routes.

Routes	Subnet associations	Edge associations	Route propagation	Tags
Routes (1)				
<input type="text"/> Filter routes	Both			
Destination	Target	Status	Propagated	
10.1.0.0/16	local	<input checked="" type="checkbox"/> Active	No	

17. Select your destination as the internet and target as your Internet gateway.

Edit routes

Destination	Target	Status
<input type="text"/> 0.0.0.0/0	<input type="text"/> igw-081e0739a919b8f00	-
Propagated		
No		
Remove		
Add route		

😊 Step 2: Create EC2

1. Now navigate to EC2 and create an instance based on Ubuntu OS.
2. There you need to select your VPC and subnet.

▼ Network settings [Info](#)

VPC - required [Info](#)

vpc-0ad513a33a2627f03 (staging-vpc)
10.1.0.0/16



Subnet [Info](#)

subnet-0354cbbb41a5244a8 appsubnet
VPC: vpc-0ad513a33a2627f03 Owner: 678586570493
Availability Zone: ap-southeast-1a IP addresses available: 251 CIDR: 10.1.0.0/24



[Create new subnet](#)

Auto-assign public IP [Info](#)

Enable

3. Then just create your VPC. Once your instance is in running state.
4. Login into your instance using Putty tool.

Instances (1/1) Info		C	Connect	Instance state ▾	Actions ▾	Launch instances ▾
<input type="text"/> Find instance by attribute or tag (case-sensitive)						
<input checked="" type="checkbox"/>	Name ▾	Instance ID	Instance state ▾	Instance type ▾	Status check	Alarm status
<input checked="" type="checkbox"/>	app-vm01	i-087ab23a467179363	Running	t2.micro	2/2 checks passed	No alarms

```
ubuntu@ip-10-1-0-74: ~
Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-1-0-74:~$
```

5. If you use the Curl command to go ahead and send the request onto the private IP address of Web instance of Mumbai region. And you'll put index dot PHP. It's not going to work. That's because we are trying to contact this machine via its private IP address. The IP used here is the IP of Web instance that you have running in Mumbai region.

Curl <http://10.0.0.68/index.php>

```
ubuntu@ip-10-1-0-74: ~
ubuntu@ip-10-1-0-74:~$ curl http://10.0.0.68/index.php
```

6. If you do control Z to stop this request. Then if you do a curl command for the public IP address of the machine. So, the public IP address and slash index dot PHP. You will see that you are getting the information accordingly.
7. So, you are just getting what is the HTML, you know, part of that particular page. But you are getting the information.

```

ubuntu@ip-10-1-0-74:~$ curl 15.206.100.230/index.php
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="description" content="Displaying AWS Certifications data">
    <link href=".//lib/bootstrap/dist/css/bootstrap.min.css" rel="stylesheet">
    <title>AWS Certifications</title>
  </head>
  <body>
    <div class="container-sm">
      <h1>AWS Certifications</h1>
      <p class="lead">This is a list of some of the AWS Certifications</p>

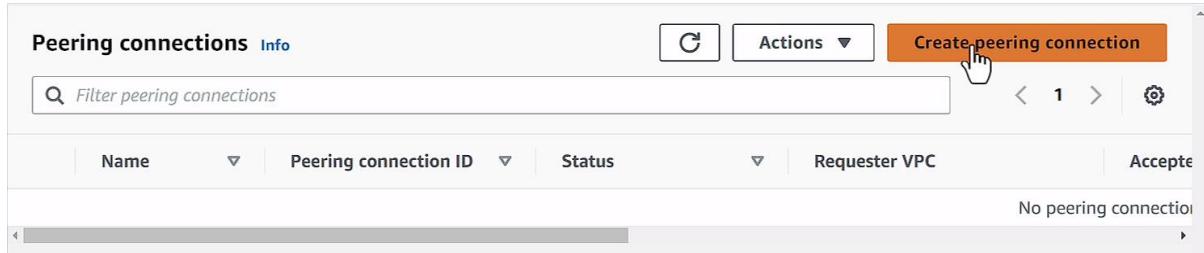
      <table class="table table-striped table-bordered">
        <thead class="table-dark">
          <tr>
            <th class="th-sm" scope="col">Course ID</th>
            <th class="th-sm" scope="col">Course Name</th>
            <th class="th-sm" scope="col">Rating</th>

```

8. So, you want to get this same information using the private IP address. You want internal communication to be possible between the machines across these VPCs. For this you have to create a VPC peering connection.

😊 Step 3: Create Peering Connection

1. Now to create VPC peering connection you need to navigate to VPC then look for Peering Connection.
2. So, in the Singapore region, create your peering connection.



3. Now give a name to your connection then select you VPC.

Peering connection settings

Name - *optional*

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

vpc-peering

Select a local VPC to peer with

VPC ID (Requester)

vpc-0ad513a33a2627f03 (staging-vpc)



VPC CIDRs for vpc-0ad513a33a2627f03 (staging-vpc)

CIDR	Status	Status reason
10.1.0.0/16	Associated	-

4. Now select My account and for region select another region.
5. Then you need to add the VPC ID of the Mumbai region VPC. For that navigate to VPC of Mumbai region and copy the ID then come back and paste it here.
6. After that choose to peering connection.

Select another VPC to peer with

Account

My account

Another account

Region

This Region (ap-southeast-1)

Another Region

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



VPC ID (Acceptor)

vpc-07d5ee339a65c1ef9

7. If you go on to peering connections, you will see that it is in the pending acceptance state. There is kind of a life cycle when it comes on to VPC peering connections. So first you create the connection from one side, then on the other side for the other VPC.
8. The owner of that account needs to go ahead and approve this VPC peering connection. Now, since this connection is within the same account itself.

Peering connections (1) Info				
Name	Peering connection ID	Status	Requester VPC	Accepter
vpc-peering	pcx-0b363242c1a7a77a3	Pending acceptance	vpc-0ad513a33a2627f03 / sta...	vpc-07d

9. So, in a new tab open VPC peering connection of Mumbai region and there accept the request.
10. Now select you VPC and accept the request.

Peering connections (1/1) Info				
Name	Peering connection ID	Status	Requester VPC	Accepter
-	pcx-0b363242c1a7a77a3	Pending acceptance	vpc-0ad513a33a2627f03	vpc-07d

11. Here you can see that you peering connection is in active state now.

ⓘ Your VPC peering connection (pcx-0b363242c1a7a77a3) has been established. To send and receive traffic across this VPC peering connection, you must add a route to the peered VPC in one or more of your VPC route tables. [Info](#) Modify my route tables now X

Peering connections (1/1) Info				
Name	Peering connection ID	Status	Requester VPC	Accepter
-	pcx-0b363242c1a7a77a3	Active	vpc-0ad513a33a2627f03	vpc-07d

12. Now, for the next steps? You now need to update the route tables for both of the VPCs.
13. You will go on to the route table for your public route table for now because you're trying to reach Web Instance.
14. Go on to routes. Click on edit routes. Add an additional route here. In the destination output is the CIDR block for your staging VPC which is in the Singapore region and the target is your peering connection.

Edit routes

Destination	Target	Status
<input type="text" value="10.1.0.0/16"/> <input type="button" value="X"/>	<input type="text" value="pcx-0b363242c1a7a77a3"/> <input type="button" value="X"/>	-
Propagated		
No		
<input type="button" value="Remove"/>		
<input type="button" value="Add route"/>		

15. Similarly, you have to go on to the Singapore region. Go on to the route tables, onto your public route table here. Go onto routes. Click on edit routes. Add a route. This is going to be for the CIDR block of your Mumbai VPC. The target is your Peering connection.

Edit routes

Destination	Target
<input type="text" value="10.0.0.0/16"/> <input type="button" value="X"/>	<input type="text" value="pcx-0b363242c1a7a77a3"/> <input type="button" value="X"/>
Propagated	
No	
<input type="button" value="Remove"/>	

16. Once it is done and the peering connection is set.
17. Now go back to Putty session or login again if it is expired.
18. Then again run the CURL command for the Private IP of your web instance. This time you will see that it is showing you the information accordingly.