😊 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:

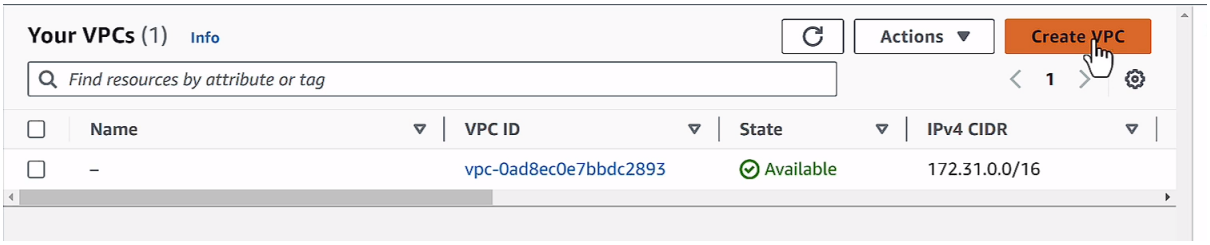
1. **Cross-VPC Communication:** VPC Peering allows resources in one VPC to communicate with resources in another VPC using private IP addresses.
2. **Security:** VPC Peering does not involve a gateway device or VPN connection. It is a direct connection between VPCs over the cloud provider's network infrastructure, ensuring secure communication.
3. **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.
4. **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

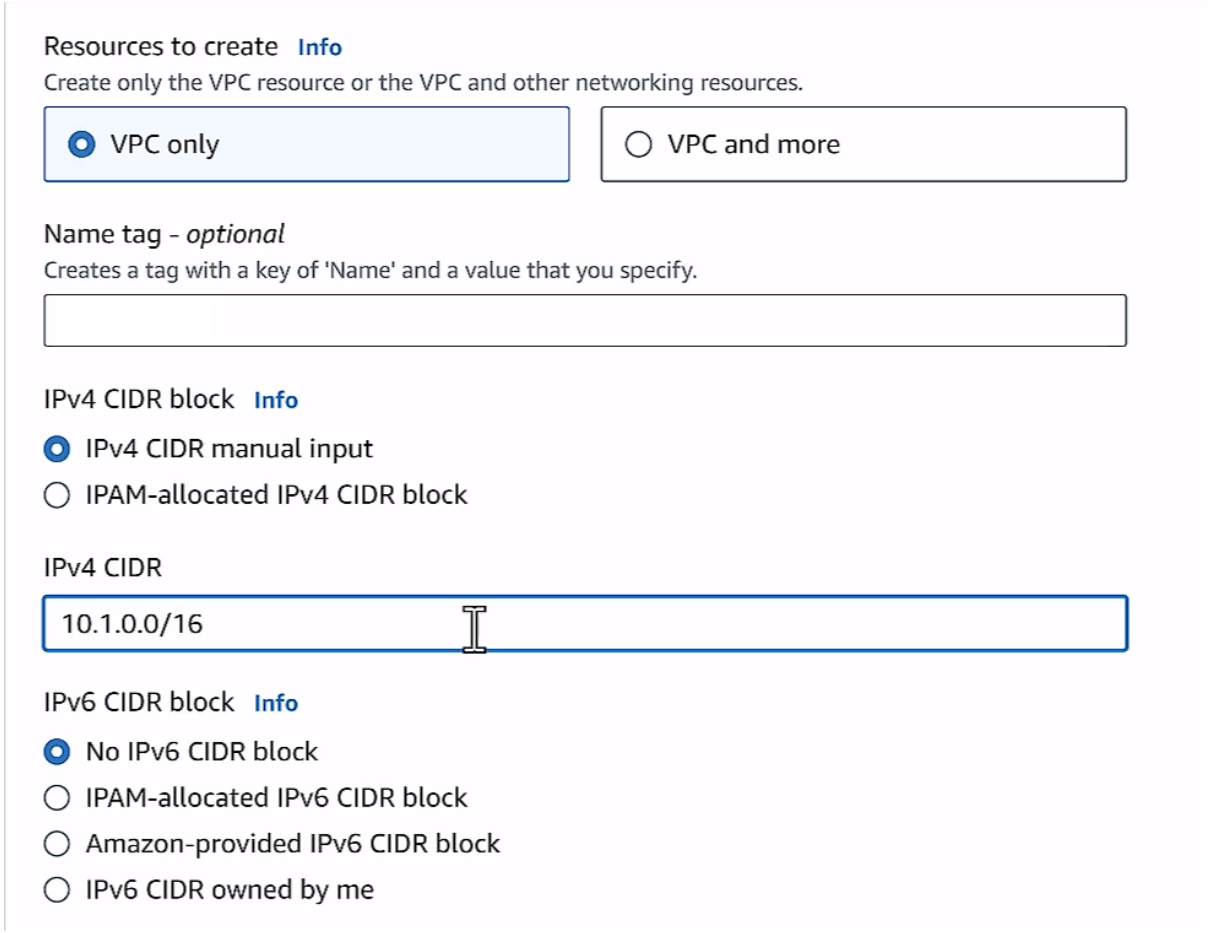
😁 Step 1: Create VPC

1. 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.
2. Switch to Singapore region and then navigate to VPC and click on create VPC.



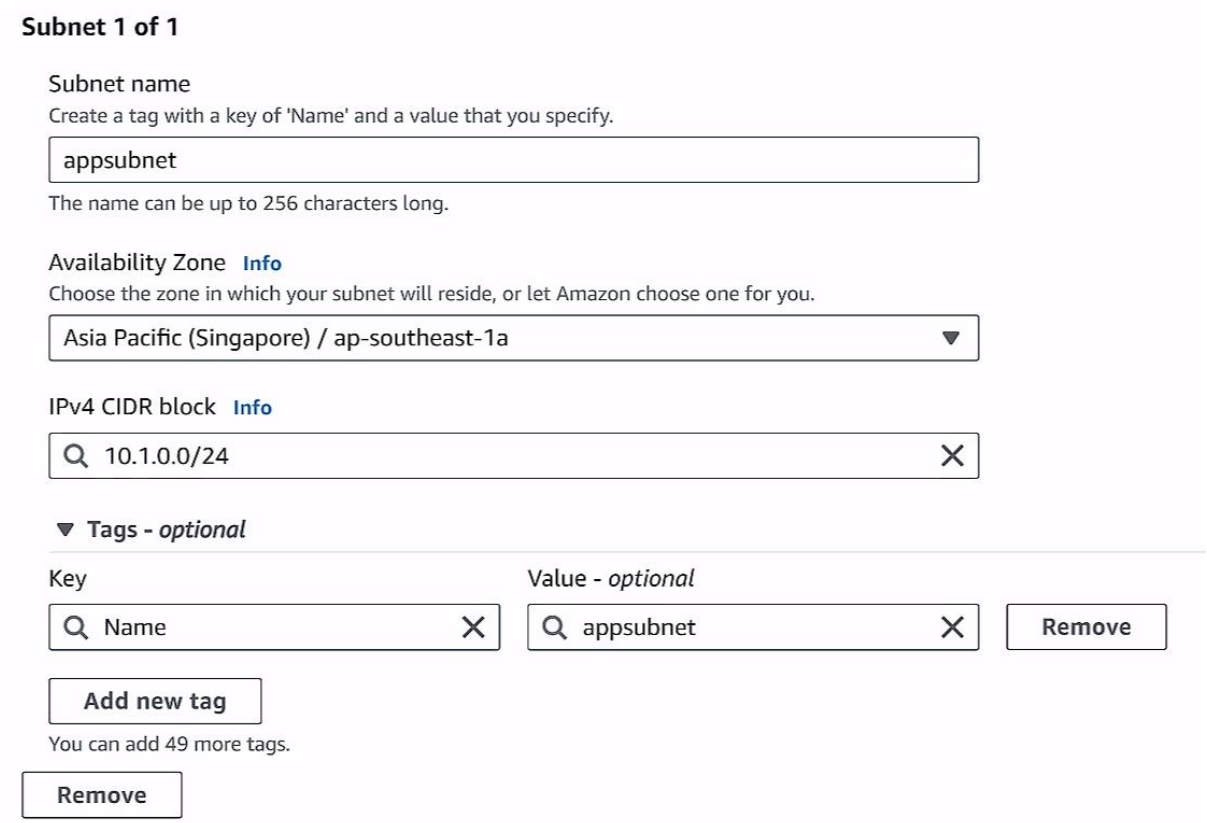
3. Select VPC only and give it a name, then IPv4 as your CIDR block.
4. Give a different CIDR for this region. The CIDR cannot be same for both of the regions.

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

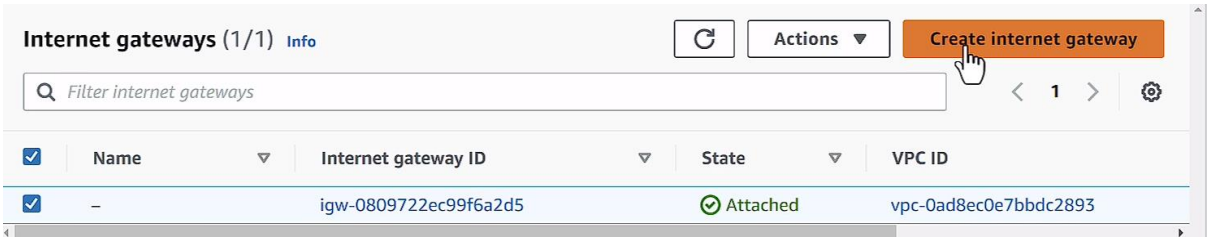


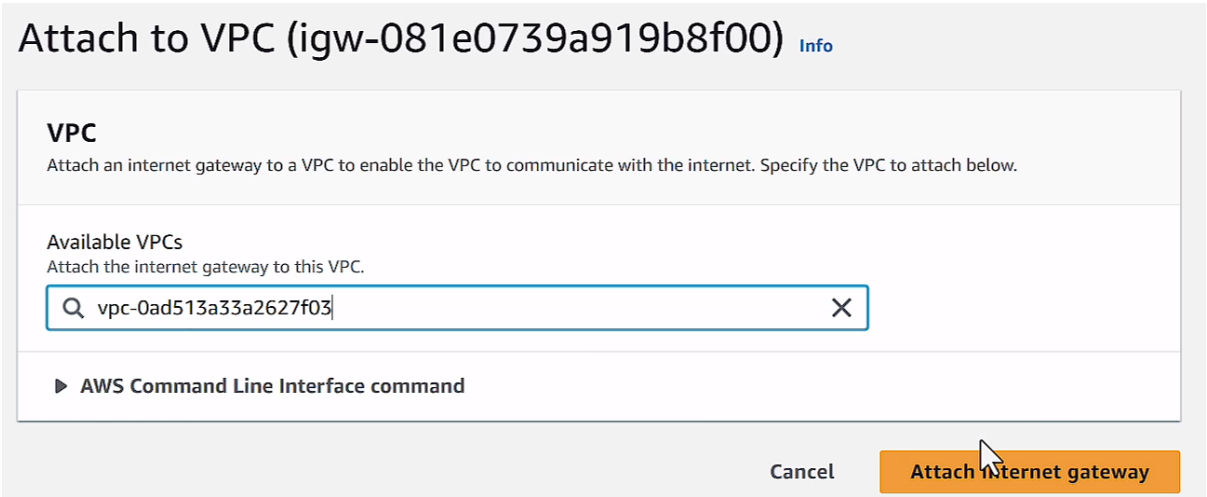
6. Now go onto subnets and create a new subnet.
7. Then select your VPC.

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.

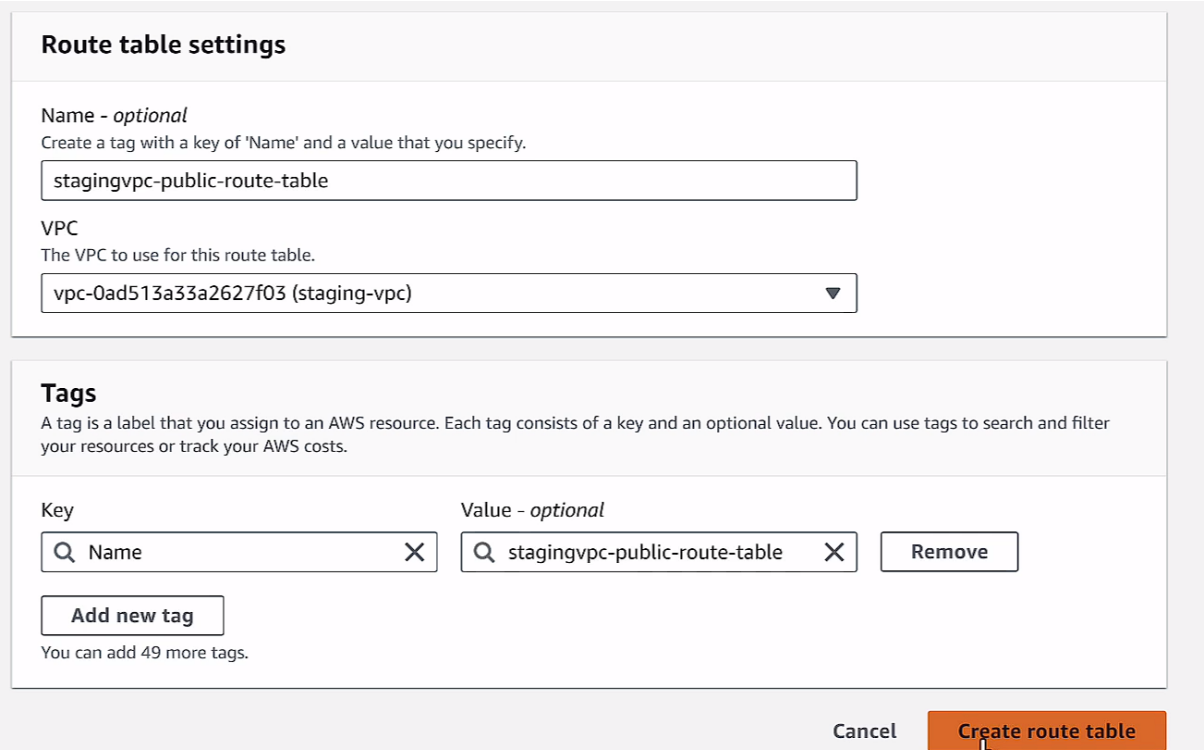


10. When the subnet is created then go to Internet gateways (IGT) and create an Internet gateway.
11. Now give it a name and create it.
12. Then you need to attach this internet gateway to your newly created VPC.

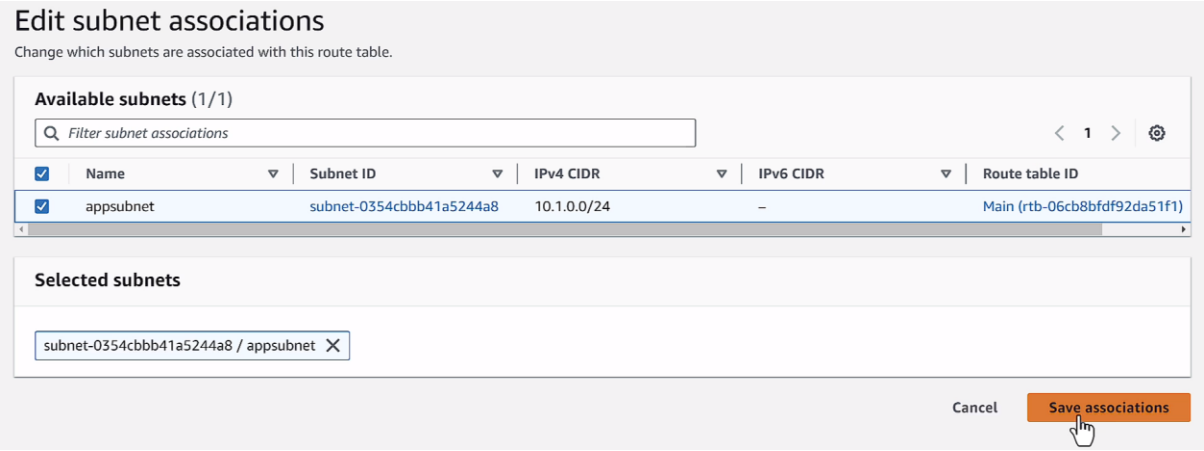




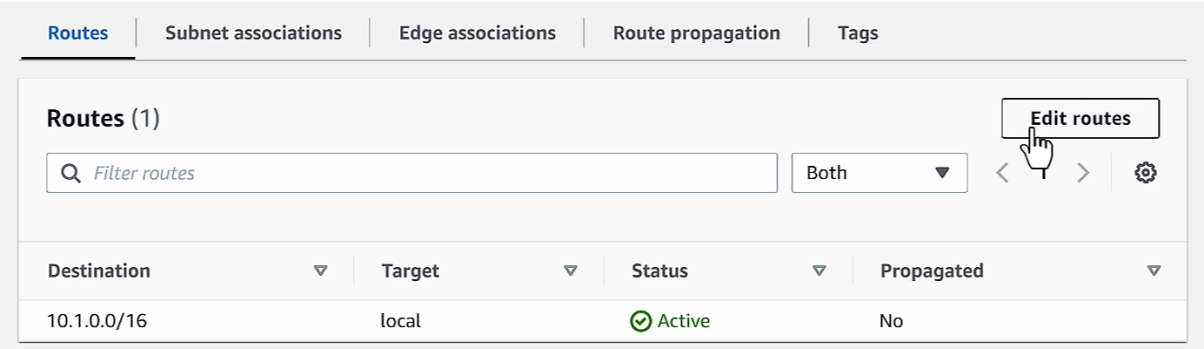
13. Then go to the route table and create a new route table.
14. Give it a name then select your VPC and create your route table.



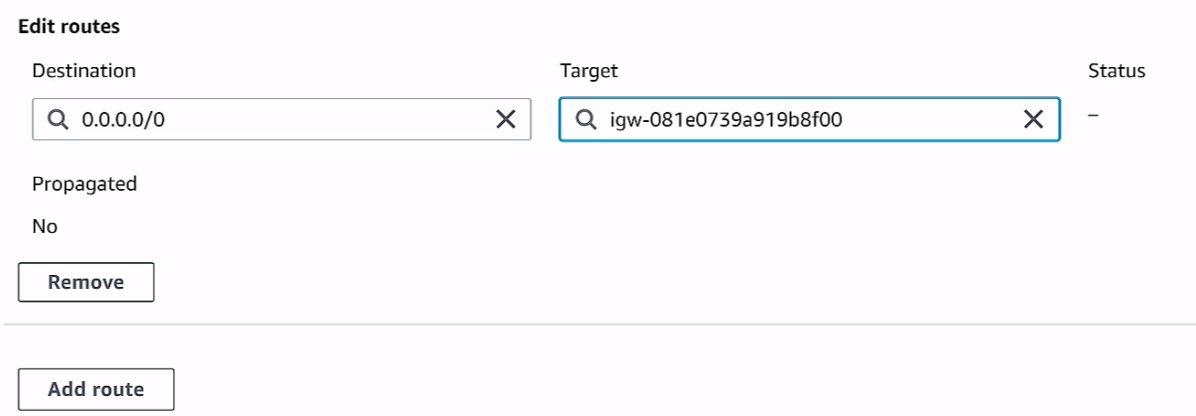
15. Once your route table is created then navigate to subnet association and associate your public subnet to it.



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

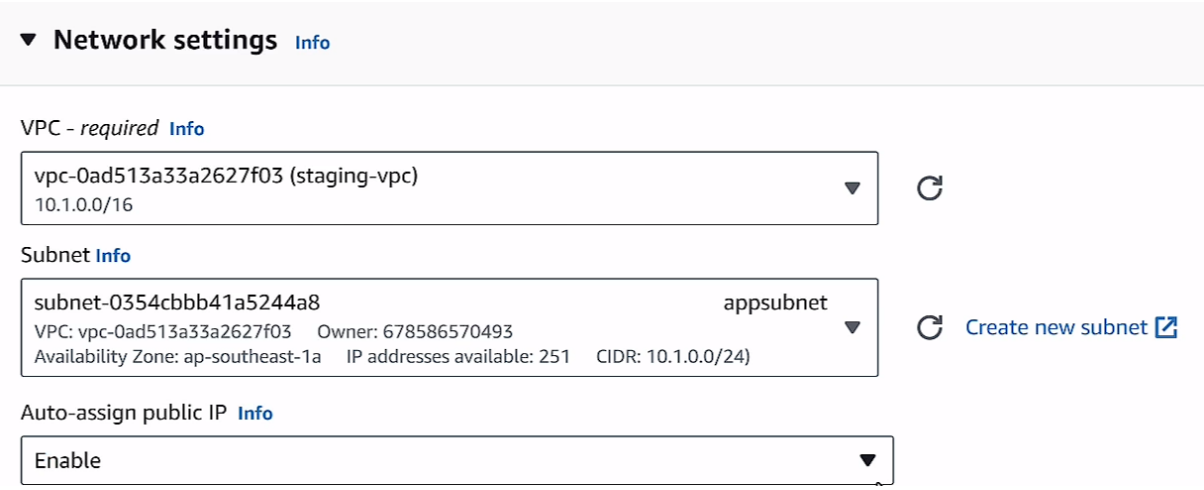


17. Select your destination as the internet and target as your Internet gateway.

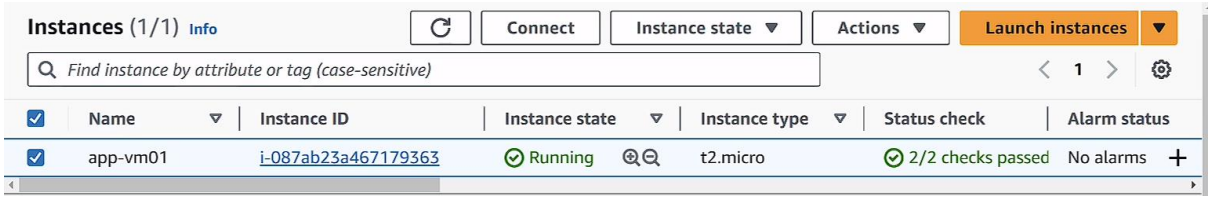


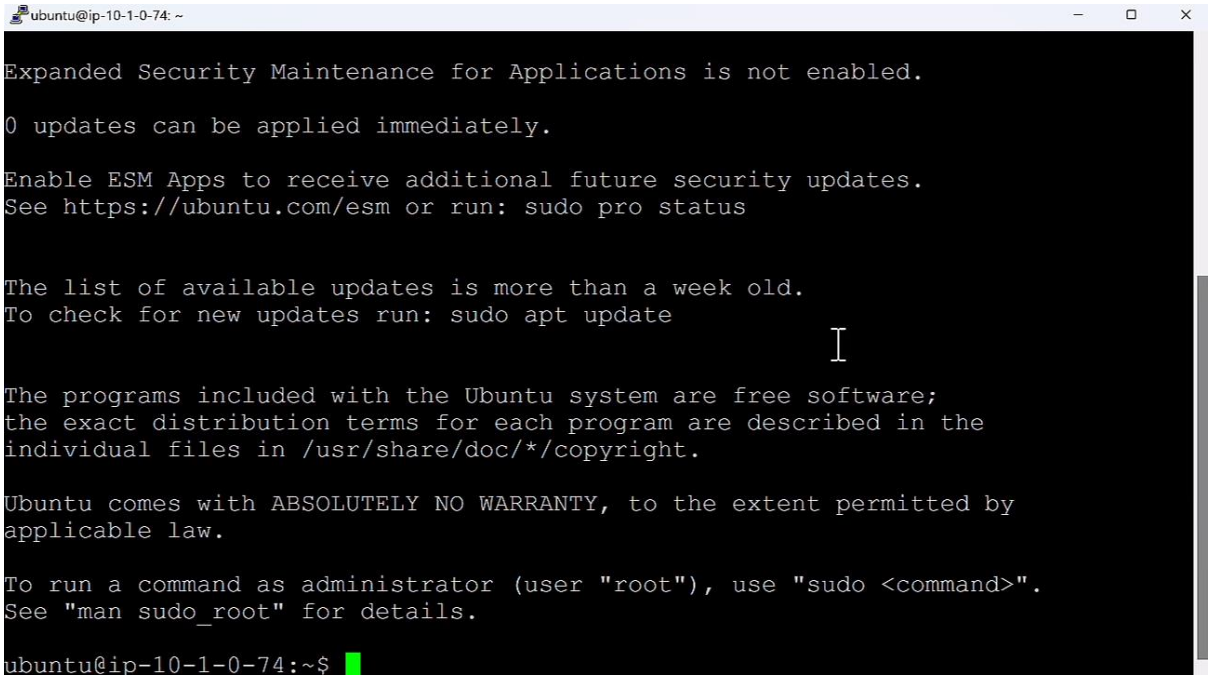
😁 Step 2: Create EC2

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

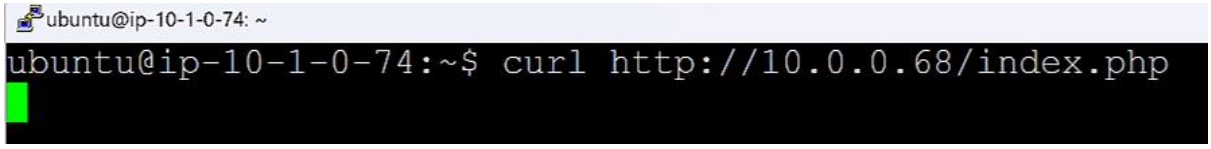


3. Then just create your VPC. Once your instance is in running state.
4. Login into your instance using Putty tool.





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



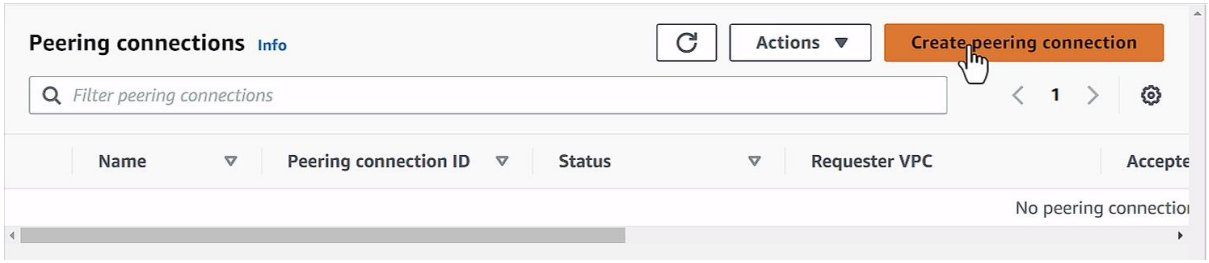
6. If you do control Z to stop this request. Then if you do a curl command for the public IP address of the machine. So, the public IP address and slash index dot PHP. You will see that you are getting the information accordingly.
7. So, you are just getting what is the HTML, you know, part of that particular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

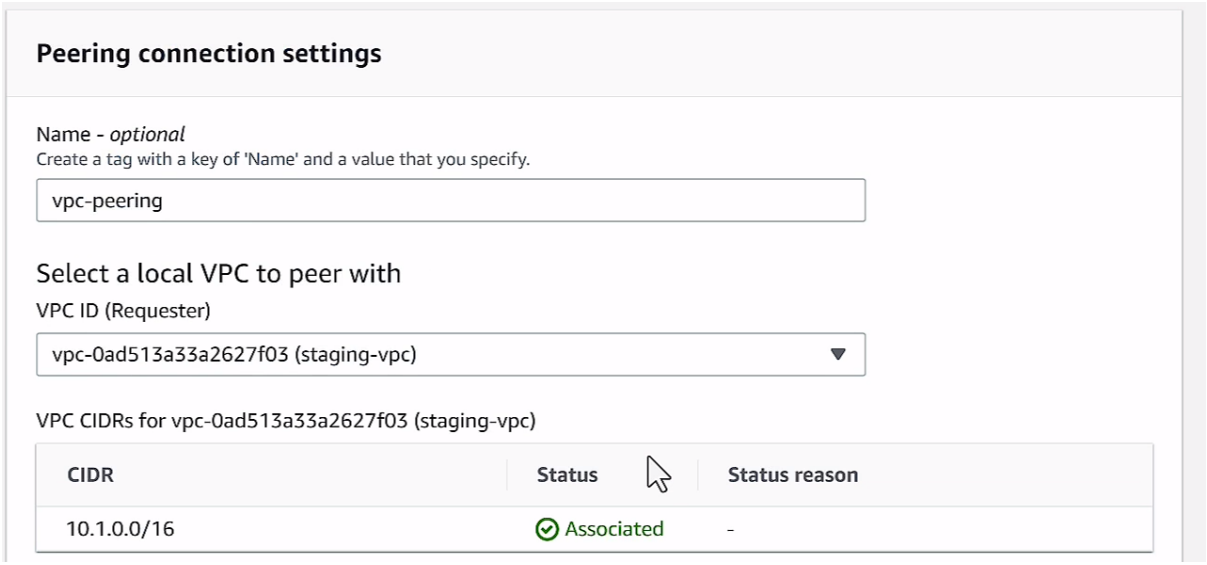
8. So, you want to get this same information using the private IP address. You want internal communication to be possible between the machines across these VPCs. For this you have to create a VPC peering connection.

😁 Step 3: Create Peering Connection

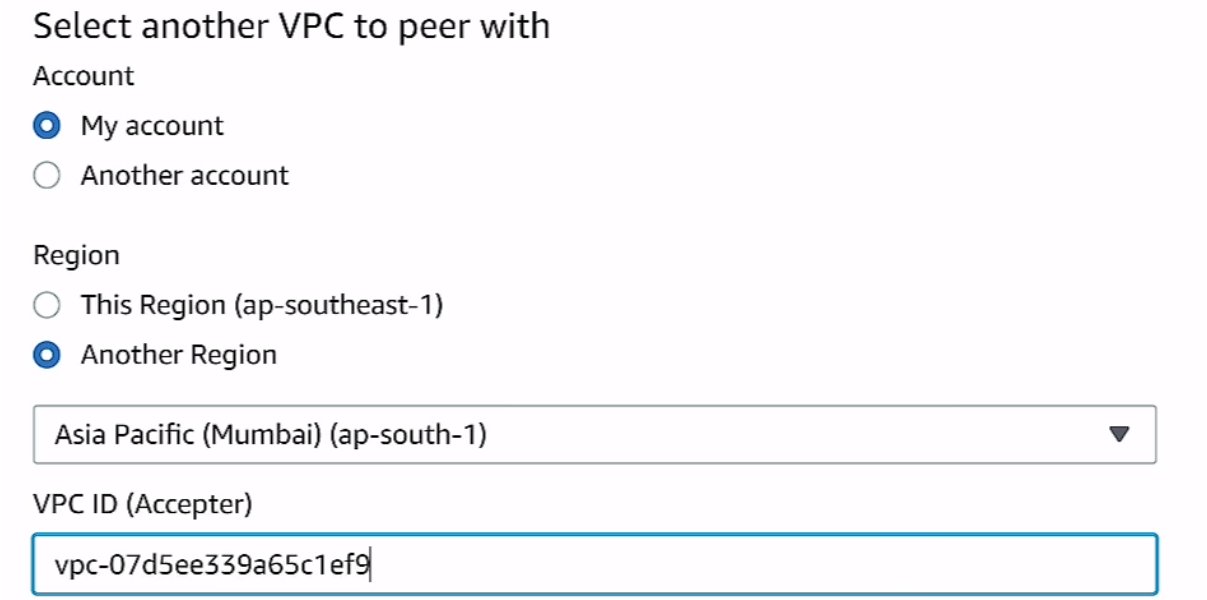
1. Now to create VPC peering connection you need to navigate to VPC then look for Peering Connection.
2. So, in the Singapore region, create your peering connection.



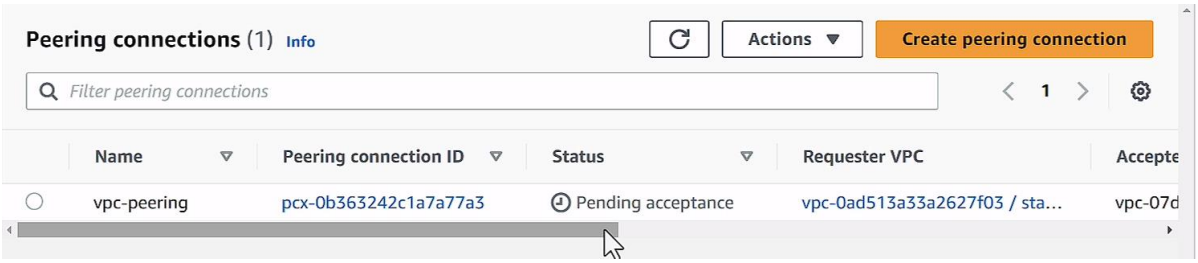
3. Now give a name to your connection then select you VPC.



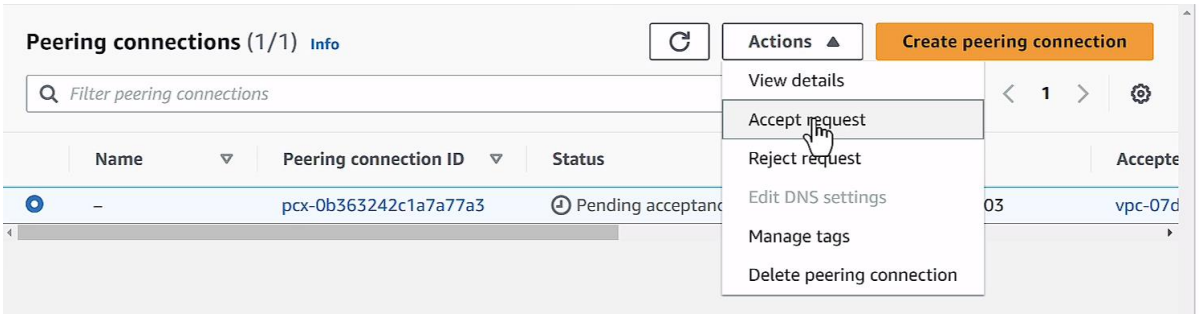
4. Now select My account and for region select another region.
5. Then you need to add the VPC ID of the Mumbai region VPC. For that navigate to VPC of Mumbai region and copy the ID then come back and paste it here.
6. After that choose to peering connection.



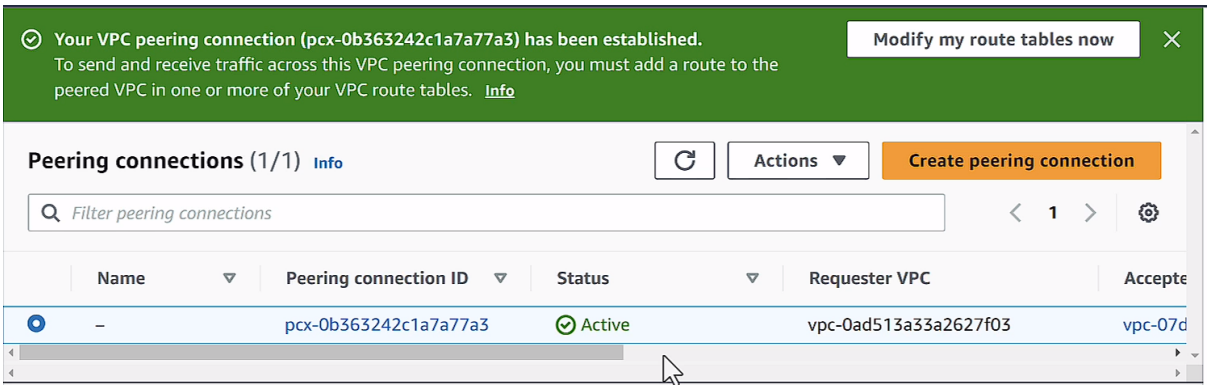
7. If you go on to peering connections, you will see that it is in the pending acceptance state. There is kind of a life cycle when it comes on to VPC peering connections. So first you create the connection from one side, then on the other side for the other VPC.
8. The owner of that account needs to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



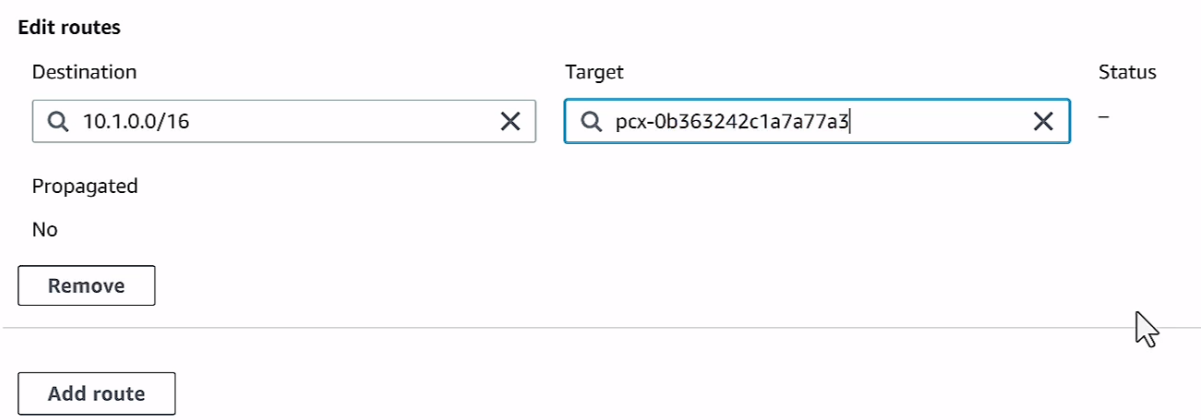
9. So, in a new tab open VPC peering connection of Mumbai region and there accept the request.
10. Now select you VPC and accept the request.



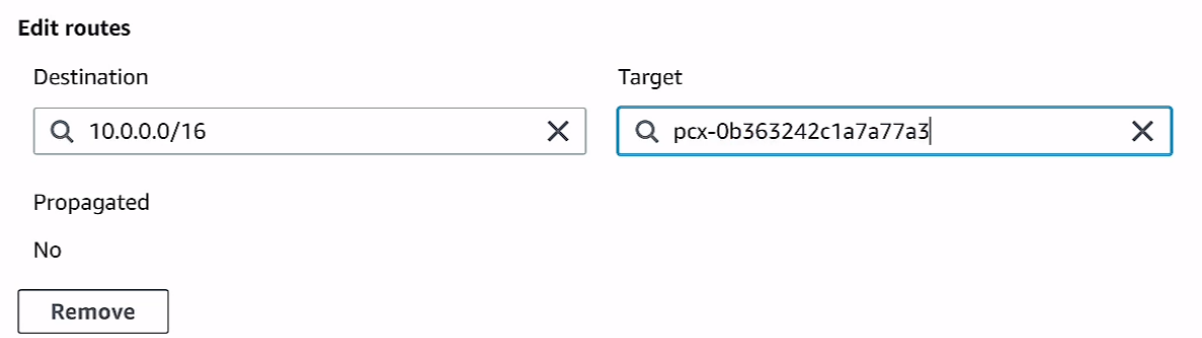
11. Here you can see that you peering connection is in active state now.



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.



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.



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.