CLOUD TRAIN

ACCELERATE YOUR GROWTH

MODULE 4

NETWORKING

AWS Workshop

Contact us

TO ACCELERATE YOUR CAREER GROWTH

For questions and more details:

please call @ +91 98712 72900, or

visit https://www.thecloudtrain.com/, or

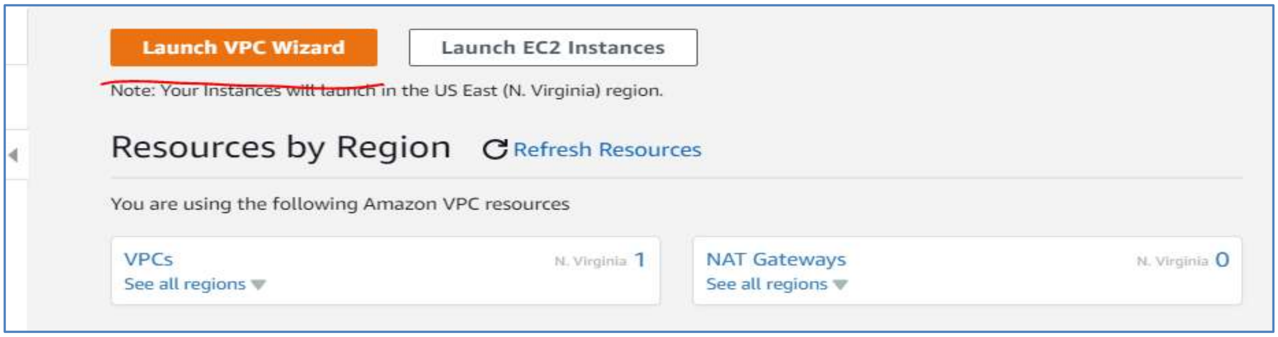
email at support@thecloudtrain.com, or

WhatsApp us @ +91 98712 72900

Create a Non-default VPC and attach it to an EC2 instance

Step 1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

Step 2. In the navigation pane, click **VPC Dashboard**. If you do not already have any VPC resources, locate the Your Virtual Private Cloud area of the dashboard and click **Get started creating a VPC**. Otherwise, click **LaunchVPC Wizard**.

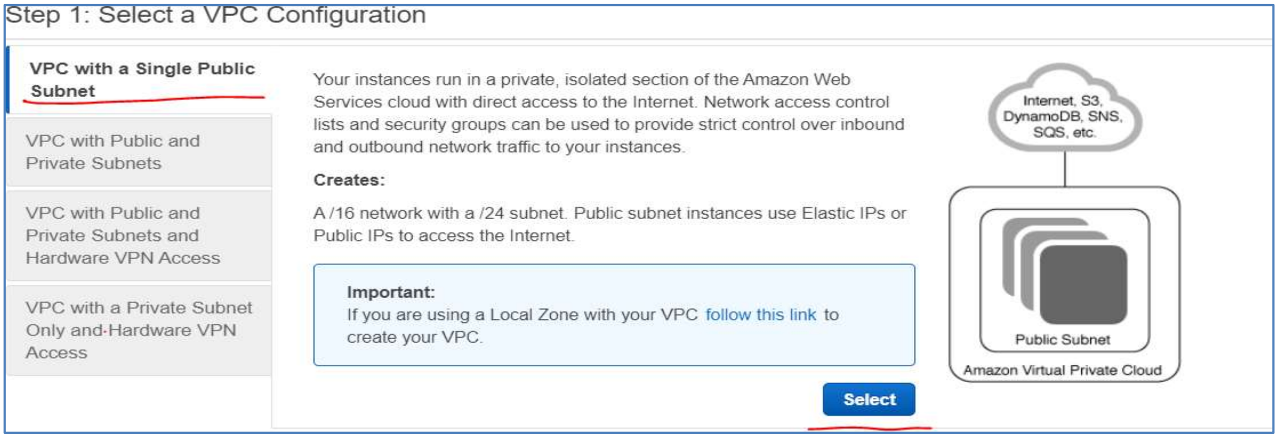


Note: Your Instances will launch in the US East (N. Virginia) region.

Resources by Region

You are using the following Amazon VPC resources

Step 3. Select the option, **VPC with a Single Public Subnet**, and then click **Select**.


VPC with a Single Public Subnet
VPC with Public and Private Subnets
VPC with Public and Private Subnets and Hardware VPN Access
VPC with a Private Subnet Only and Hardware VPN Access
Internet, S3, DynamoDB, SNS, SQS, etc.
Public Subnet
Amazon Virtual Private Cloud
Select

Step 1: Select a VPC Configuration

Your instances run in a private, isolated section of the Amazon Web Services cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.

Creates:

A /16 network with a /24 subnet. Public subnet instances use Elastic IPs or Public IPs to access the Internet.

Important:
If you are using a Local Zone with your VPC follow this link to create your VPC.

Step 4. Enter the following information into the wizard and click **Create VPC**.

IP CIDR block

10.0.0.0/16

VPC name

Custom VPC

Public subnet

10.0.0.0/24

Availability Zone

No Preference

Subnet name

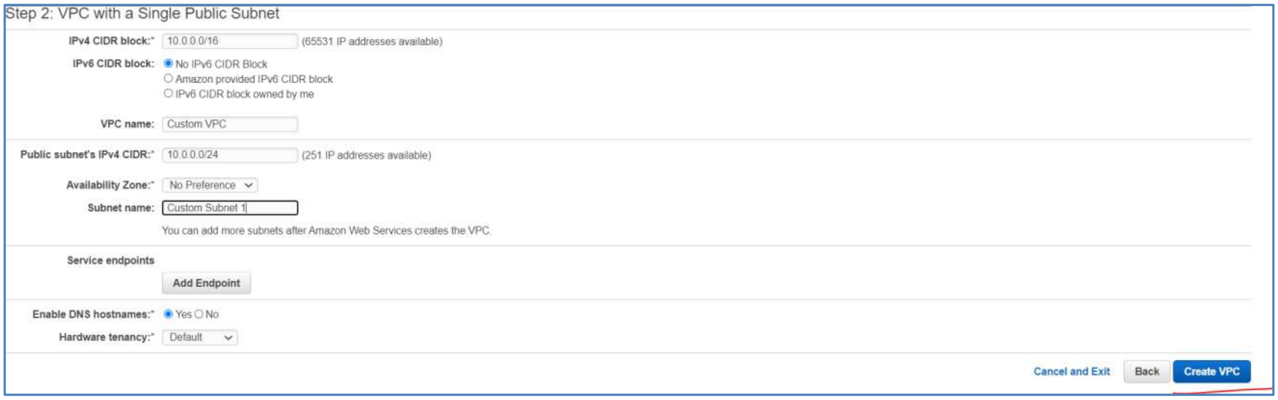
Custom Subnet 1

Enable DNS hostnames

Leave default selection

Hardware tenancy

Default



Step 2: VPC with a Single Public Subnet

IPv4 CIDR block:* 10.0.0.0/16 (65531 IP addresses available)

IPv6 CIDR block: ● No IPv6 CIDR Block ○ Amazon provided IPv6 CIDR block ○ IPv6 CIDR block owned by me

VPC name: Custom VPC

Public subnet's IPv4 CIDR:* 10.0.0.0/24 (251 IP addresses available)

Availability Zone:* No Preference

Subnet name: Custom Subnet 1

You can add more subnets after Amazon Web Services creates the VPC.

Service endpoints

Enable DNS hostnames:* ● Yes ○ No

Hardware tenancy:* Default

Step 5. It takes several minutes for the VPC to be created. After the VPC is created, proceed to the following section to add a second subnet.

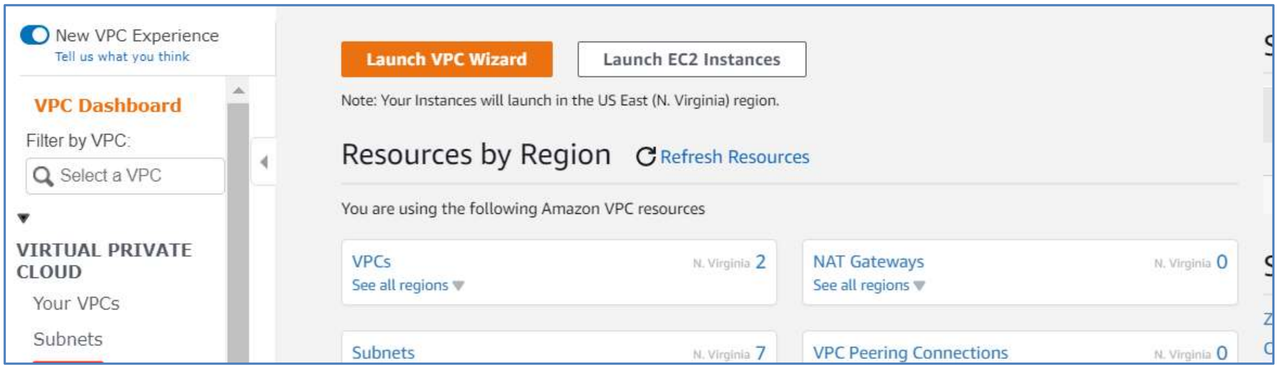
Add a second subnet

AWS Directory Service requires two subnets in your VPC, and each subnet must be in a different Availability Zone. The VPC wizard only creates one subnet, so you must manually create the second subnet, and specify a different Availability Zone than the first subnet. Create the second subnet by performing the following steps.

To create a subnet

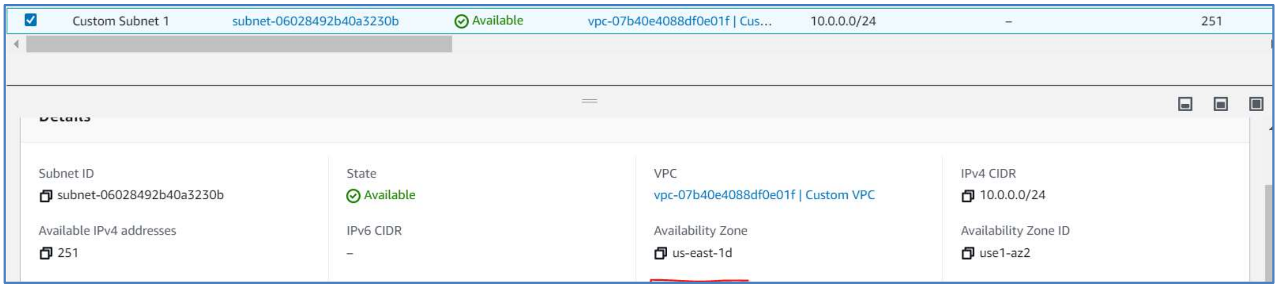
Step 1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

Step 2. In the navigation pane, select **Subnets**, select the subnet with the name Custom Subnet 1, and select the **Summary** tab at the bottom of the page. Make a note of the Availability Zone of this subnet.



Note: Your Instances will launch in the US East (N. Virginia) region.

You are using the following Amazon VPC resources

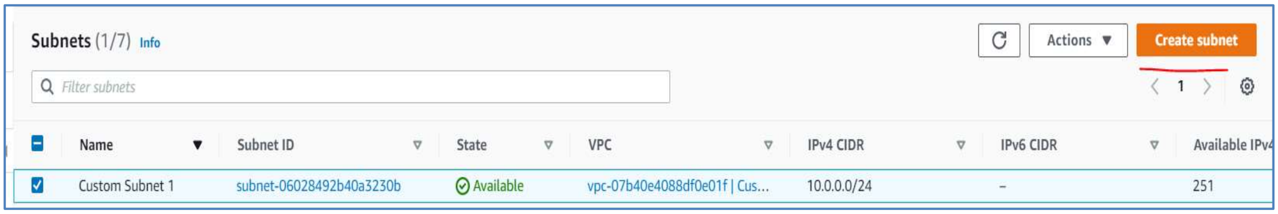


☑	Custom Subnet 1	subnet-06028492b40a3230b	Available	vpc-07b40e4088df0e01f | Cus...	10.0.0.0/24	–	251

Details

Subnet ID: subnet-06028492b40a3230b
State: Available
VPC: vpc-07b40e4088df0e01f | Custom VPC
IPv4 CIDR: 10.0.0.0/24
Available IPv4 addresses: 251
IPv6 CIDR: –
Availability Zone: us-east-1d
Availability Zone ID: use1-az2

Step 3. Click **Create Subnet** and enter the following information in the **Create Subnet** dialog box and click **Yes, Create**.



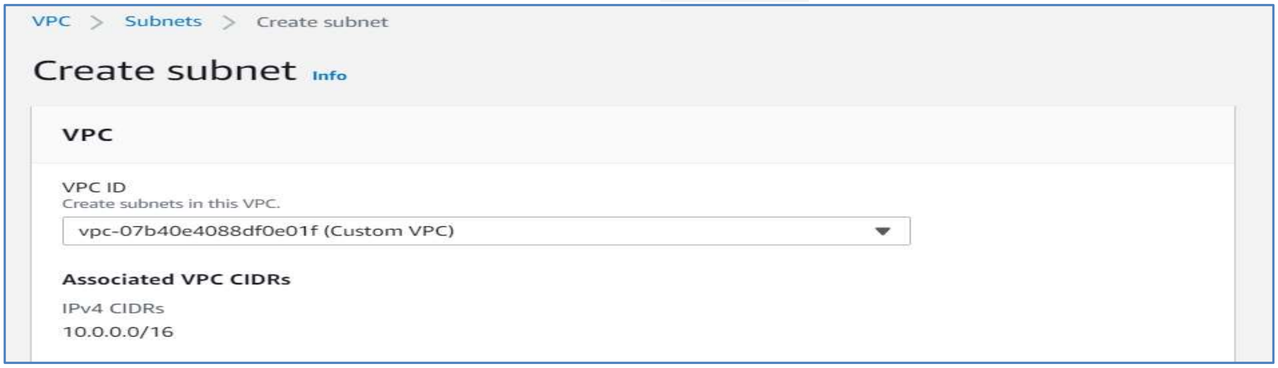
☐	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv4
☑	Custom Subnet 1	subnet-06028492b40a3230b	Available	vpc-07b40e4088df0e01f | Cus...	10.0.0.0/24	–	251

Name tag

Custom Subnet 2

VPC

Select your VPC. This is the VPC with the name **Custom VPC.**



VPC > Subnets > Create subnet

Create subnet Info

VPC

VPC ID
Create subnets in this VPC.
vpc-07b40e4088df0e01f (Custom VPC)

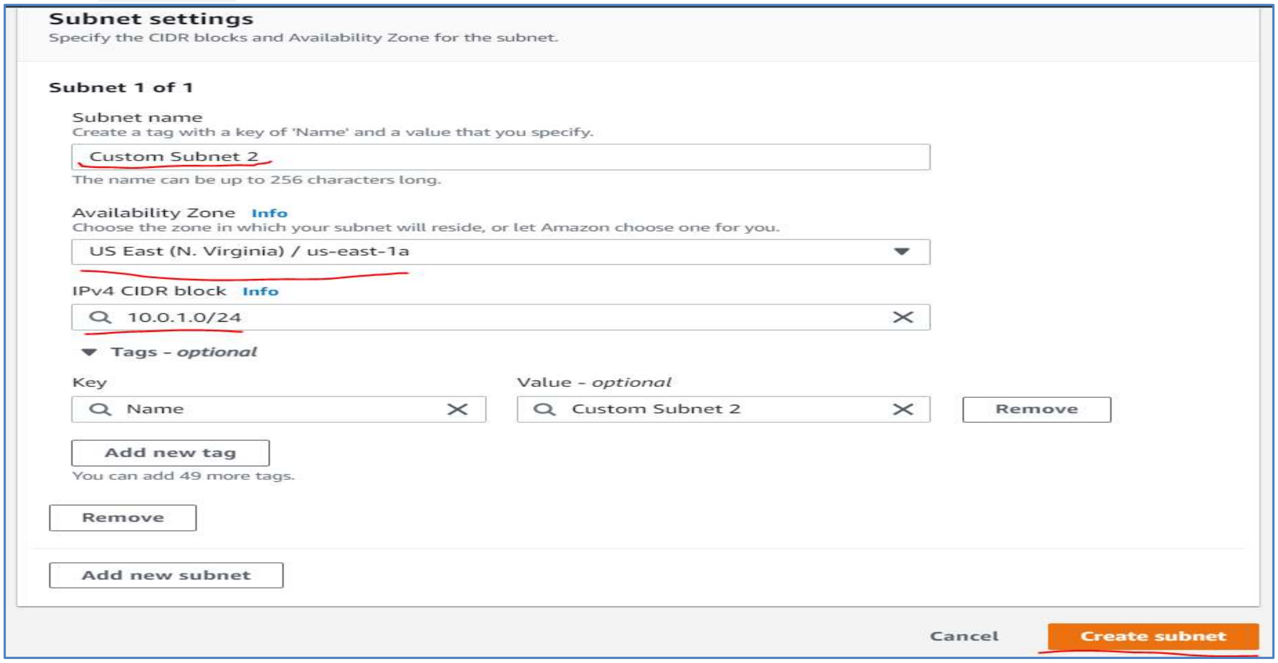
Associated VPC CIDRs
IPv4 CIDRs
10.0.0.0/16

Availability Zone

Select any Availability Zone other than the one noted in step 2. The two subnets used by AWS Directory Service must reside in different Availability Zones.

CIDR Block

10.0.1.0/24



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.
Custom Subnet 2
The name can be up to 256 characters long.

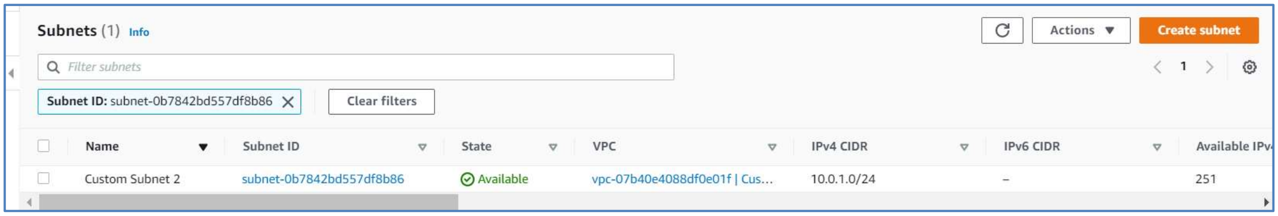
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a

IPv4 CIDR block Info
10.0.1.0/24

Tags - *optional*

Key	Value - *optional*
Name	Custom Subnet 2

Step 4. Check the subnet creating successfully:



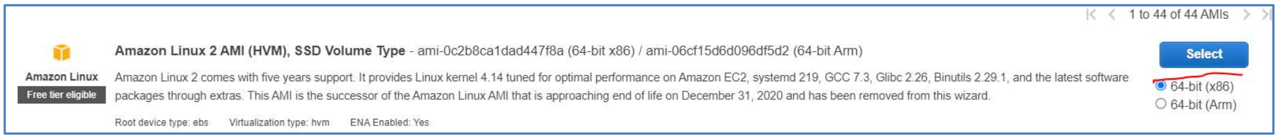
Subnets (1) Info

Subnet ID: subnet-0b7842bd557df8b86

	Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR	Available IPv
□	Custom Subnet 2	subnet-0b7842bd557df8b86	Available	vpc-07b40e4088df0e01f | Cus...	10.0.1.0/24	–	251

Attach Custom VPC to EC2 instance

Step 1. Go to Launch Instances and create a new instance using Amazon Linux:



1 to 44 of 44 AMIs

Amazon Linux
Free tier eligible

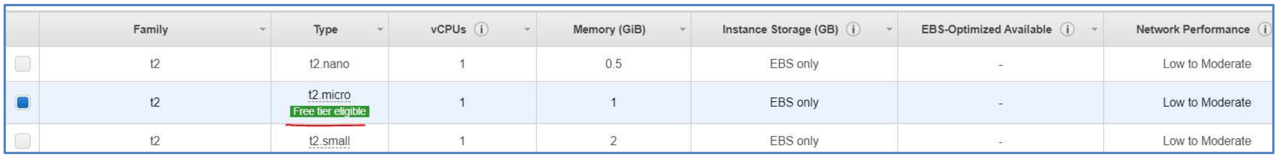
Amazon Linux 2 AMI (HVM), SSD Volume Type - ami-0c2b8ca1dad447f8a (64-bit x86) / ami-06cf15d6d096df5d2 (64-bit Arm)

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is approaching end of life on December 31, 2020 and has been removed from this wizard.

Root device type: ebs Virtualization type: hvm ENA Enabled: Yes

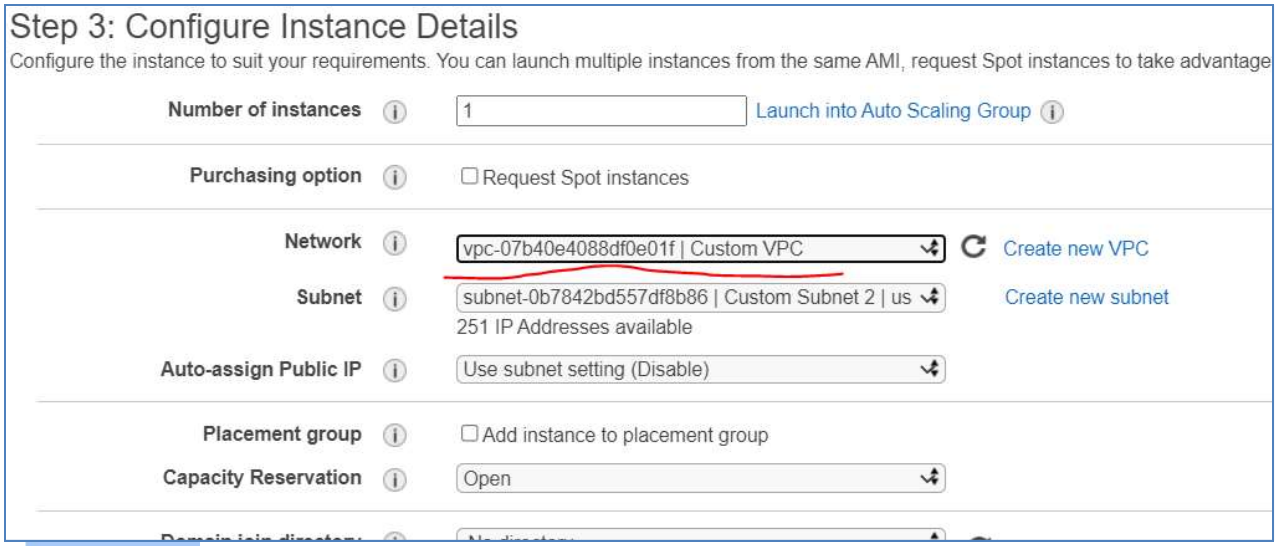
● 64-bit (x86)
○ 64-bit (Arm)

Step 2. Choose instance type as below and click Next to Configure instance details:



	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
□	t2	t2.nano	1	0.5	EBS only	-	Low to Moderate
■	t2	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate
□	t2	t2.small	1	2	EBS only	-	Low to Moderate

Step 3. Here, Select the Custom VPC that you created in previous steps. Leave other values default:



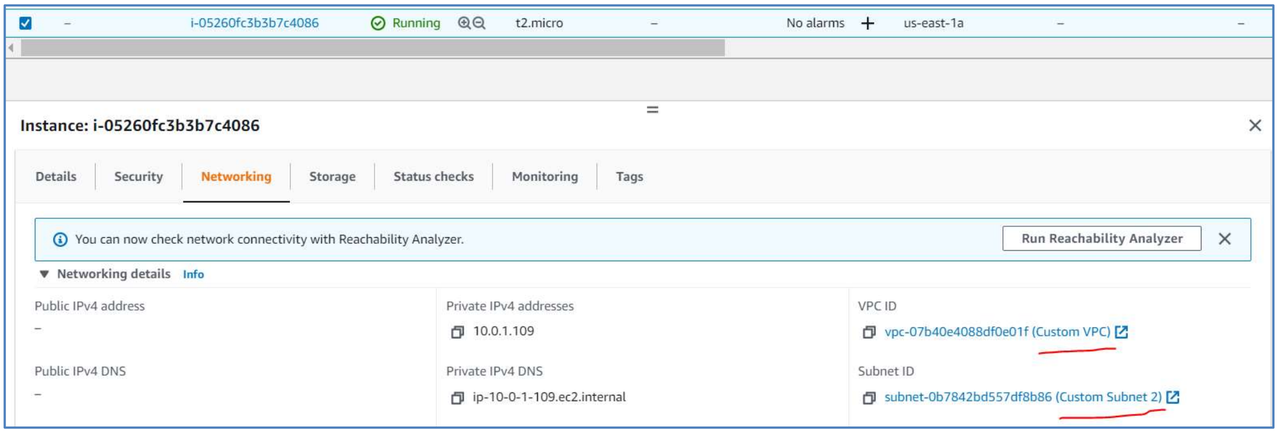
Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage

Number of instances	1 Launch into Auto Scaling Group
Purchasing option	□ Request Spot instances
Network	vpc-07b40e4088df0e01f | Custom VPC Create new VPC
Subnet	subnet-0b7842bd557df8b86 | Custom Subnet 2 | us Create new subnet 251 IP Addresses available
Auto-assign Public IP	Use subnet setting (Disable)
Placement group	□ Add instance to placement group
Capacity Reservation	Open

Step 4. Proceed to next pages with default values and at the end, click **Launch** with the existing key pair.

Once instance is launched, then try to connect to it using the given aws key pair:



– i-05260fc3b3b7c4086 Running t2.micro – No alarms + us-east-1a – –

Instance: i-05260fc3b3b7c4086

You can now check network connectivity with Reachability Analyzer.

Public IPv4 address: –

Public IPv4 DNS: –

Private IPv4 addresses: 10.0.1.109

Private IPv4 DNS: ip-10-0-1-109.ec2.internal

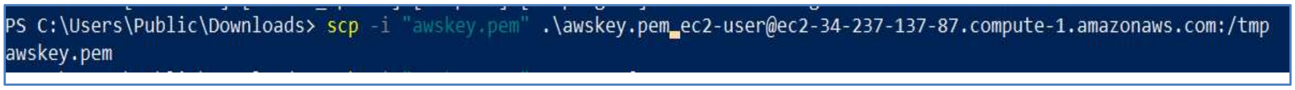
VPC ID: vpc-07b40e4088df0e01f (Custom VPC)

Subnet ID: subnet-0b7842bd557df8b86 (Custom Subnet 2)

Step 5. Since, we have not assigned an Internet Gateway to our VPC, we cannot directly connect to this instance directly from public server like our laptop. You can see above screenshot, there is no Public IP address assigned. So we need to connect this instance from an EC2 instance having public IP. Use existing EC2 instance that you created earlier in default VPC or create a new one in Default VPC only, so that we can connect to it from outside.

Step 6. To connect from the public instance, we need aws key pair also inside it. Lets copy it from our laptop to the public instance in default VPC using scp in /tmp folder. Make sure you use your server name here in the command. Do not just copy paste:

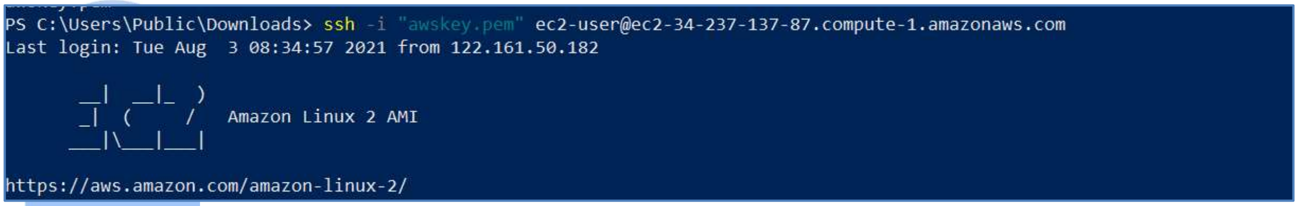
```
scp -i "awskey.pem" .\awskey.pem ec2-user@ec2-34-237-137-87.compute-1.ama
zonaws.com:/tmp
```



```
PS C:\Users\Public\Downloads> scp -i "awskey.pem" .\awskey.pem ec2-user@ec2-34-237-137-87.compute-1.amazonaws.com:/tmp
awskey.pem
```

Step 7. Connect to the public instance.:

```
ssh -i "awskey.pem" ec2-user@ec2-34-237-137-87.compute-1.amazonaws.com
```



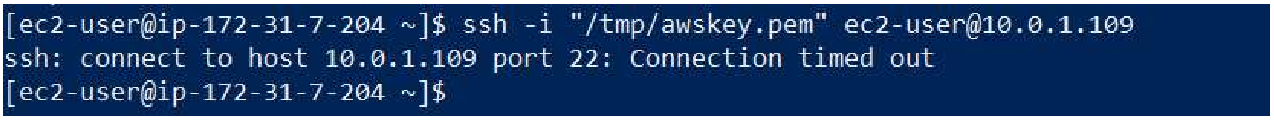
```
PS C:\Users\Public\Downloads> ssh -i "awskey.pem" ec2-user@ec2-34-237-137-87.compute-1.amazonaws.com
Last login: Tue Aug  3 08:34:57 2021 from 122.161.50.182

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
```

Step 8. Let's try to connect to the private instance with Custom VPC from inside the public instance. We already have the aws key pair copied in /tmp, so lets use it. Again, change the server IP as per your server details:

```
ssh -i "/tmp/awskey.pem" ec2-user@10.0.1.109
```



```
[ec2-user@ip-172-31-7-204 ~]$ ssh -i "/tmp/awskey.pem" ec2-user@10.0.1.109
ssh: connect to host 10.0.1.109 port 22: Connection timed out
[ec2-user@ip-172-31-7-204 ~]$
```

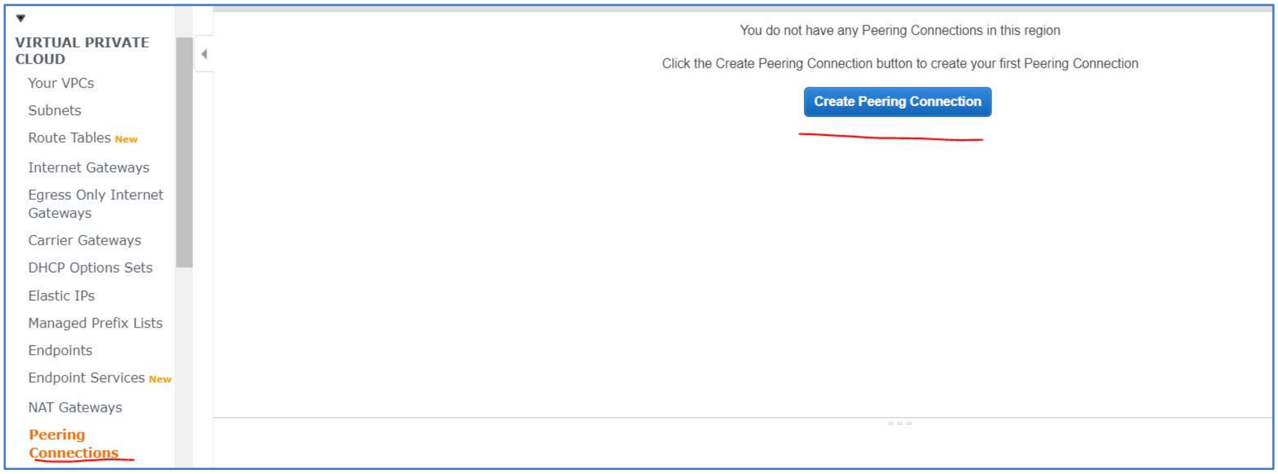
Step 9. Here, we can see connection timed out, because the instance are in different VPC which has no connection enable by default between them. We can do it by VPC peering which is our next task.

Connect two instances in different VPC's using VPC peering

To create a VPC peering connection with a VPC in the same Region

Step 1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/.

Step 2. In the navigation pane, choose **Peering Connections, Create Peering Connection**.

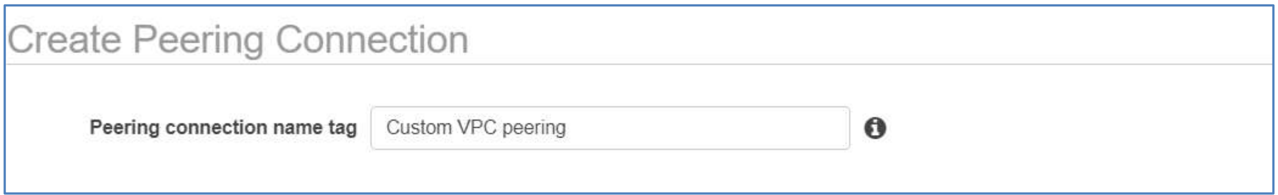


You do not have any Peering Connections in this region

Click the Create Peering Connection button to create your first Peering Connection

Step 3. Configure the following information, and choose **Create Peering Connection** when you are done:

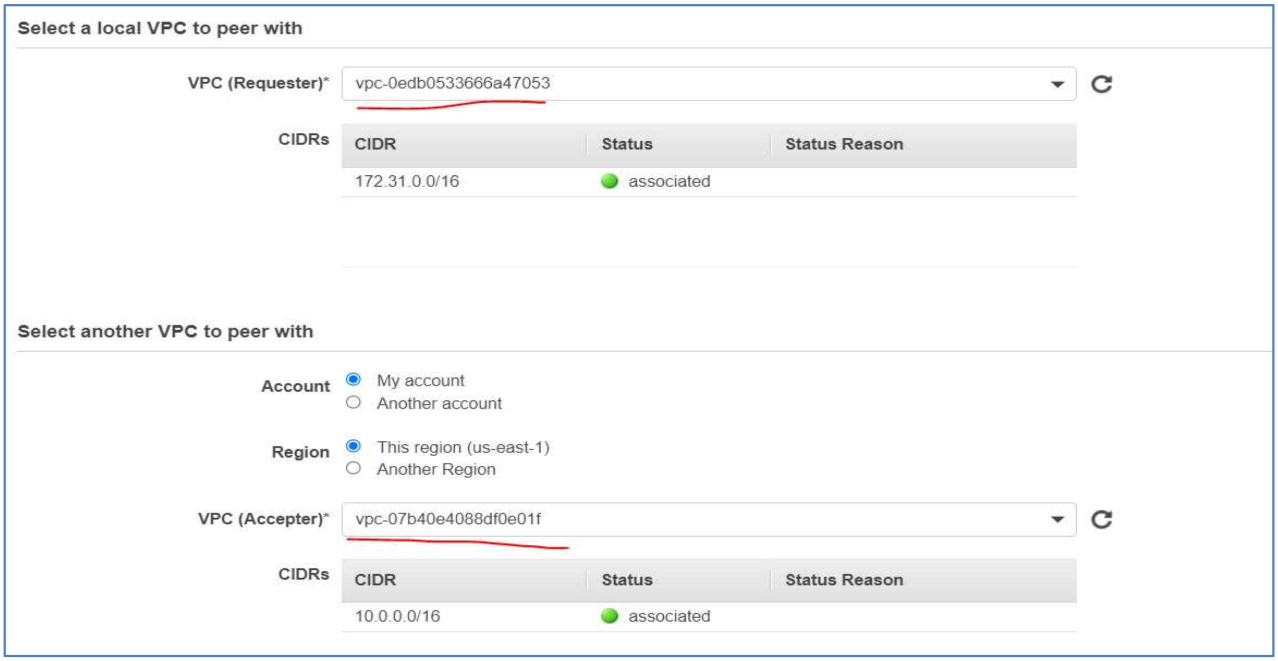
- **Peering connection name tag**: You can optionally name your VPC peering connection, say **Custom VPC peering**



Create Peering Connection

Peering connection name tag: Custom VPC peering

- **VPC (Requester)**: Select the VPC in your account with which you want to create the VPC peering connection. Lets select default VPC here.
- Under **Select another VPC to peer with**: Ensure **My account** is selected and select another of your VPCs. Lets select Custom VPC here.



Select a local VPC to peer with

VPC (Requester)* vpc-0edb0533666a47053

CIDRs

CIDR	Status	Status Reason
172.31.0.0/16	● associated	

Select another VPC to peer with

Account ● My account ○ Another account

Region ● This region (us-east-1) ○ Another Region

VPC (Accepter)* vpc-07b40e4088df0e01f

CIDRs

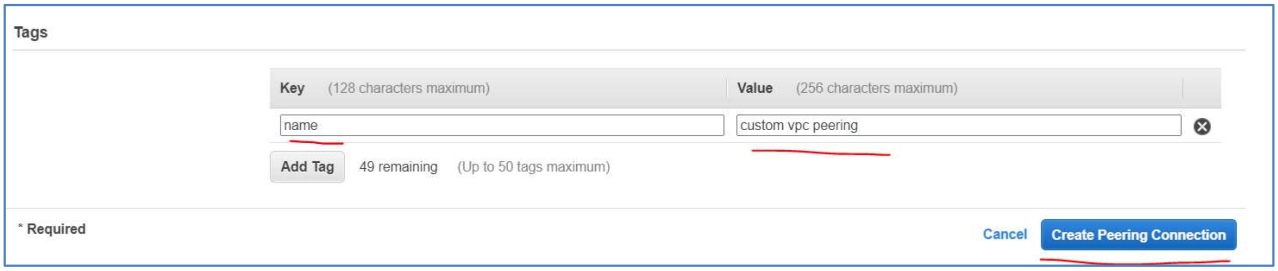
CIDR	Status	Status Reason
10.0.0.0/16	● associated	

- (Optionally add or remove a tag.

  [Add a tag] Choose **Add tag** and do the following:

  - For **Key**, enter the key name.
  - For **Value**, enter the key value.

  [Remove a tag] Choose the Delete button ("X") to the right of the tag's Key and Value.

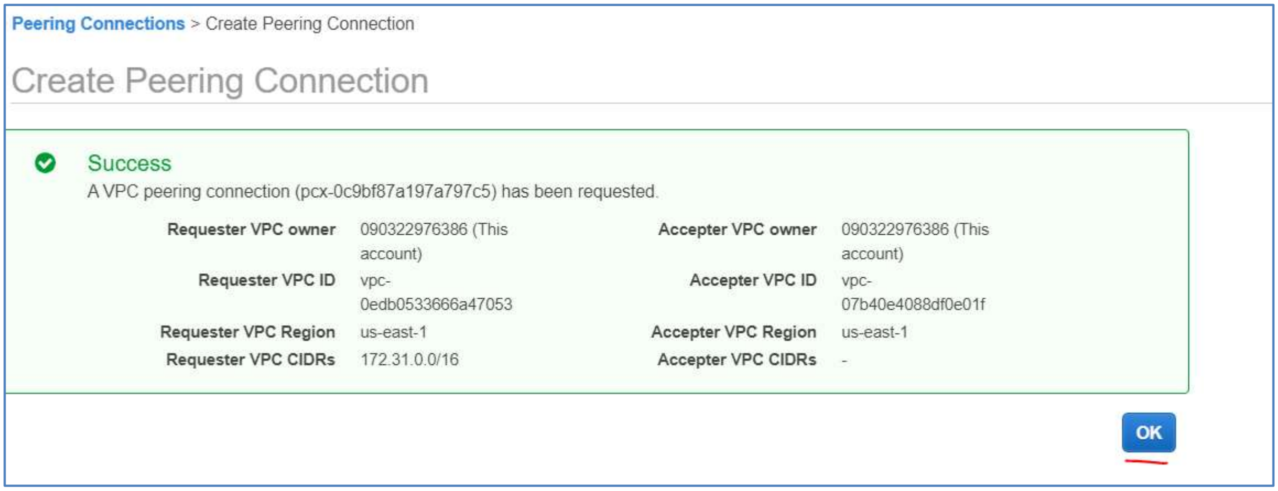


Tags

Key (128 characters maximum)	Value (256 characters maximum)
name	custom vpc peering

Add Tag
49 remaining (Up to 50 tags maximum)
* Required
Cancel
Create Peering Connection

Step 4. In the confirmation dialog box, choose **OK**.



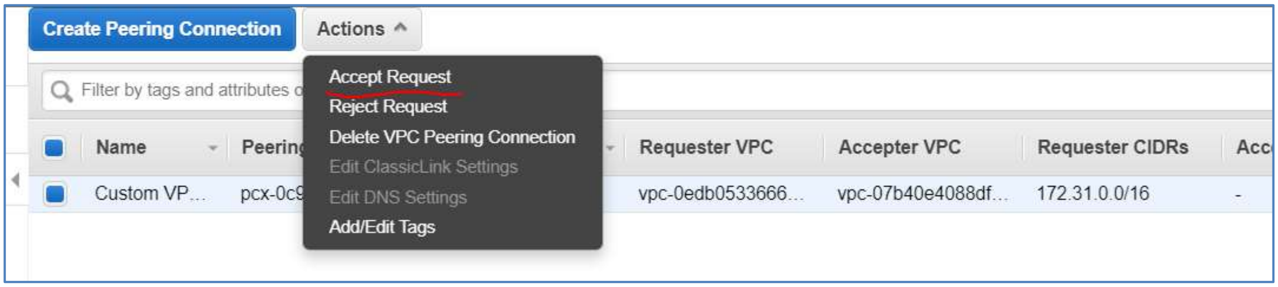
Peering Connections > Create Peering Connection

Create Peering Connection

Success
A VPC peering connection (pcx-0c9bf87a197a797c5) has been requested.

Requester VPC owner	090322976386 (This account)	Accepter VPC owner	090322976386 (This account)
Requester VPC ID	vpc-0edb0533666a47053	Accepter VPC ID	vpc-07b40e4088df0e01f
Requester VPC Region	us-east-1	Accepter VPC Region	us-east-1
Requester VPC CIDRs	172.31.0.0/16	Accepter VPC CIDRs	-

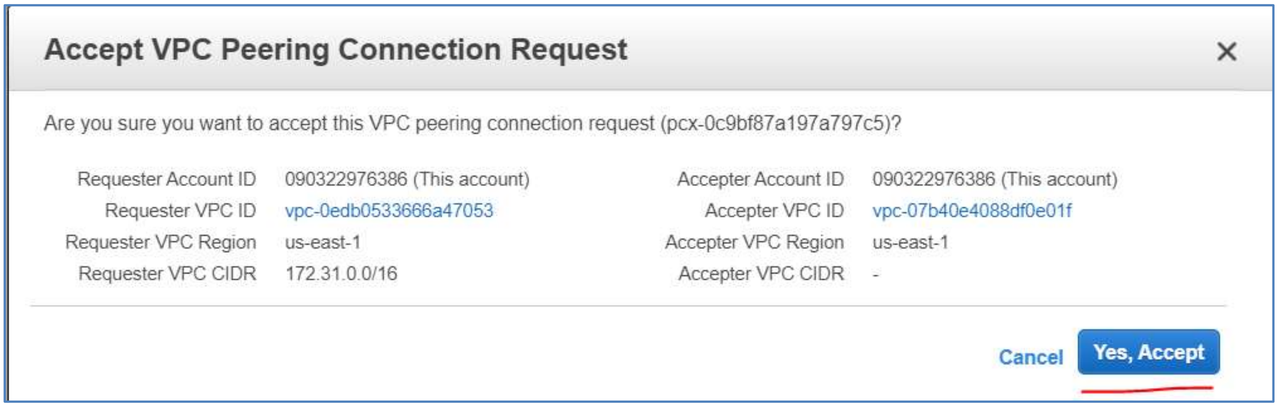
Step 5. Select the VPC peering connection that you've created, and choose **Actions**, **Accept Request**



Create Peering Connection
Actions
Filter by tags and attributes o
Accept Request
Reject Request
Delete VPC Peering Connection
Edit ClassicLink Settings
Edit DNS Settings
Add/Edit Tags

Name	Peering	Requester VPC	Accepter VPC	Requester CIDRs	Acc
Custom VP...	pcx-0c9	vpc-0edb0533666...	vpc-07b40e4088df...	172.31.0.0/16	-

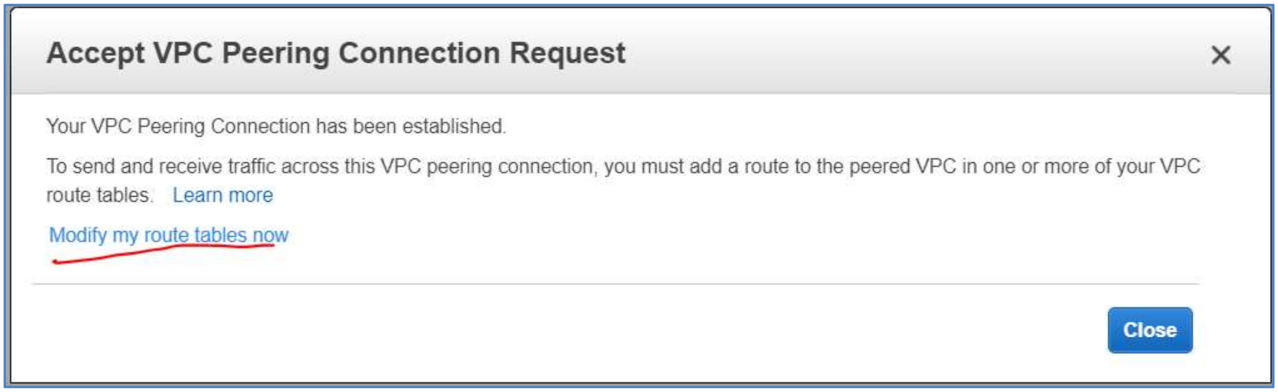
Step 6. In the confirmation dialog, choose **Yes, Accept**. A second confirmation dialog displays; choose **Modify my route tables now** to go directly to the route tables page



Accept VPC Peering Connection Request

Are you sure you want to accept this VPC peering connection request (pcx-0c9bf87a197a797c5)?

Requester Account ID	090322976386 (This account)	Accepter Account ID	090322976386 (This account)
Requester VPC ID	vpc-0edb0533666a47053	Accepter VPC ID	vpc-07b40e4088df0e01f
Requester VPC Region	us-east-1	Accepter VPC Region	us-east-1
Requester VPC CIDR	172.31.0.0/16	Accepter VPC CIDR	-



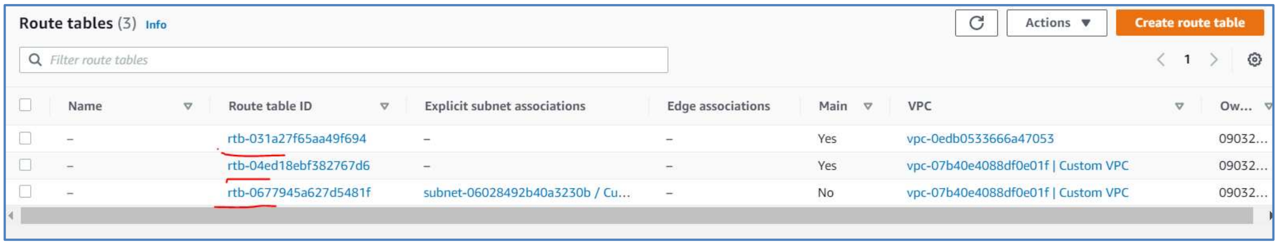
Accept VPC Peering Connection Request

Your VPC Peering Connection 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. Learn more

Modify my route tables now

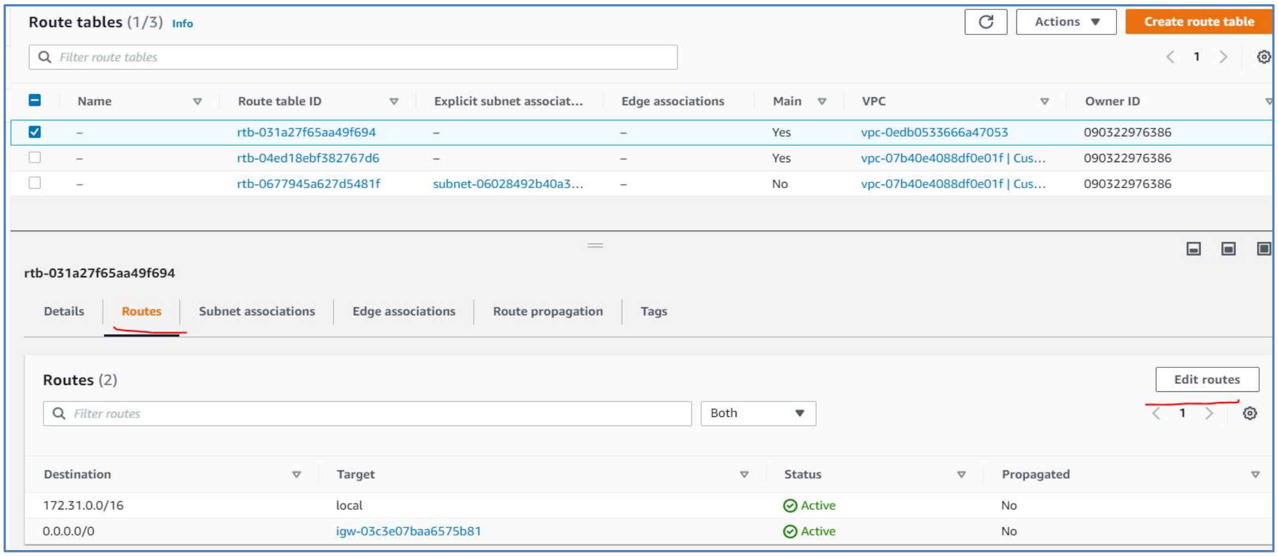
Step 7. Now, open Route tables of both the VPCs and edit them to add routes to each other by adding each other's subnet CIDR block. Here, default VPC has one subnet and custom VPC has 2 subnets that we created.



Route tables (3) Info

□	Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC	Ow...
□	–	rtb-031a27f65aa49f694	–	–	Yes	vpc-0edb0533666a47053	09032...
□	–	rtb-04ed18ebf382767d6	–	–	Yes	vpc-07b40e4088df0e01f | Custom VPC	09032...
□	–	rtb-0677945a627d5481f	subnet-06028492b40a3230b / Cu...	–	No	vpc-07b40e4088df0e01f | Custom VPC	09032...

Step 8. Let's edit the default VPC Route table and add CIDR block of both the subnets in Route table with target and **Peering connection -> Custom VPC Peering:**



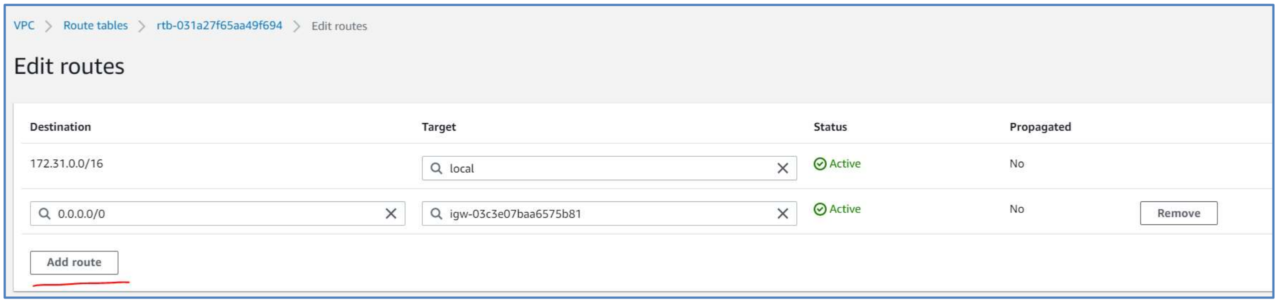
Route tables (1/3) Info

☐	Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC	Owner ID
☑	–	rtb-031a27f65aa49f694	–	–	Yes	vpc-0edb0533666a47053	090322976386
□	–	rtb-04ed18ebf382767d6	–	–	Yes	vpc-07b40e4088df0e01f | Cus...	090322976386
□	–	rtb-0677945a627d5481f	subnet-06028492b40a3...	–	No	vpc-07b40e4088df0e01f | Cus...	090322976386

rtb-031a27f65aa49f694

Routes (2)

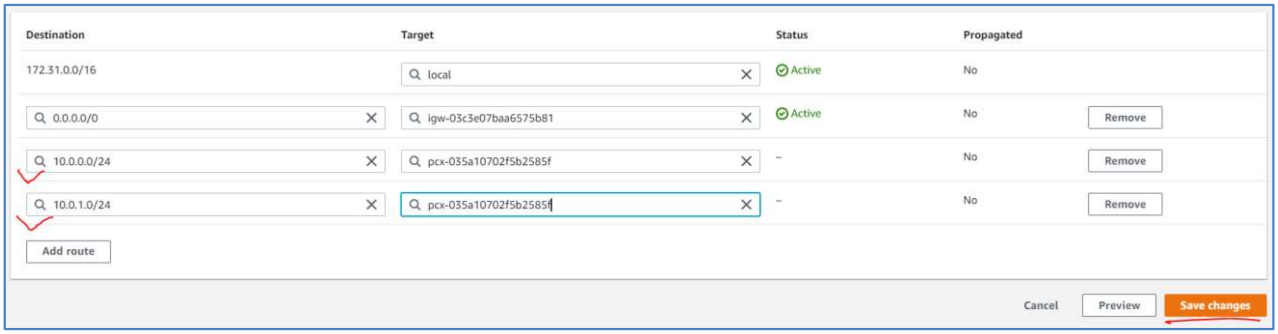
Destination	Target	Status	Propagated
172.31.0.0/16	local	✓ Active	No
0.0.0.0/0	igw-03c3e07baa6575b81	✓ Active	No



VPC > Route tables > rtb-031a27f65aa49f694 > Edit routes

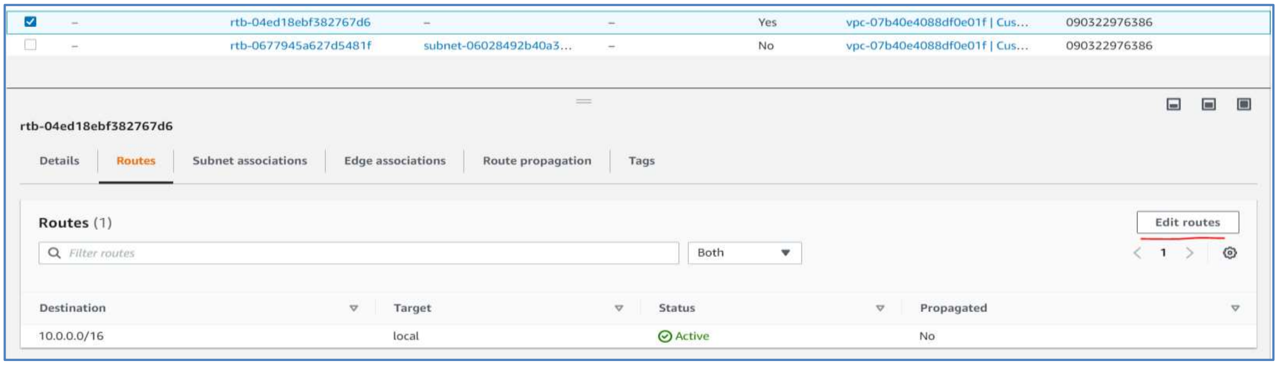
Edit routes

Destination	Target	Status	Propagated	
172.31.0.0/16	local	Active	No	
0.0.0.0/0	igw-03c3e07baa6575b81	Active	No	Remove



Destination	Target	Status	Propagated	
172.31.0.0/16	local	Active	No	
0.0.0.0/0	igw-03c3e07baa6575b81	Active	No	Remove
10.0.0.0/24	pcx-035a10702f5b2585f	-	No	Remove
10.0.1.0/24	pcx-035a10702f5b2585f	-	No	Remove

Step 9. Similarly add for route of default VPC subnet to both the subnets of Custom VPCs. Add target as **Peering Connection and** enter your **Custom VPC Peering:**

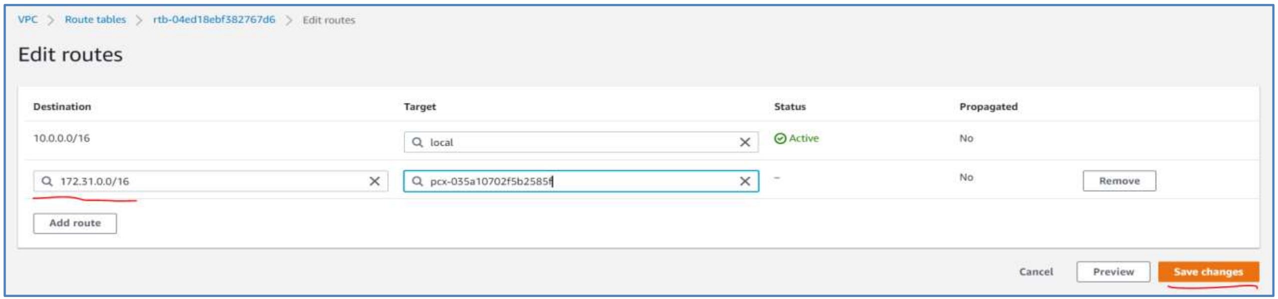


☑	–	rtb-04ed18ebf382767d6	–	–	Yes	vpc-07b40e4088df0e01f | Cus...	090322976386
□	–	rtb-0677945a627d5481f	subnet-06028492b40a3...	–	No	vpc-07b40e4088df0e01f | Cus...	090322976386

rtb-04ed18ebf382767d6

Routes (1)

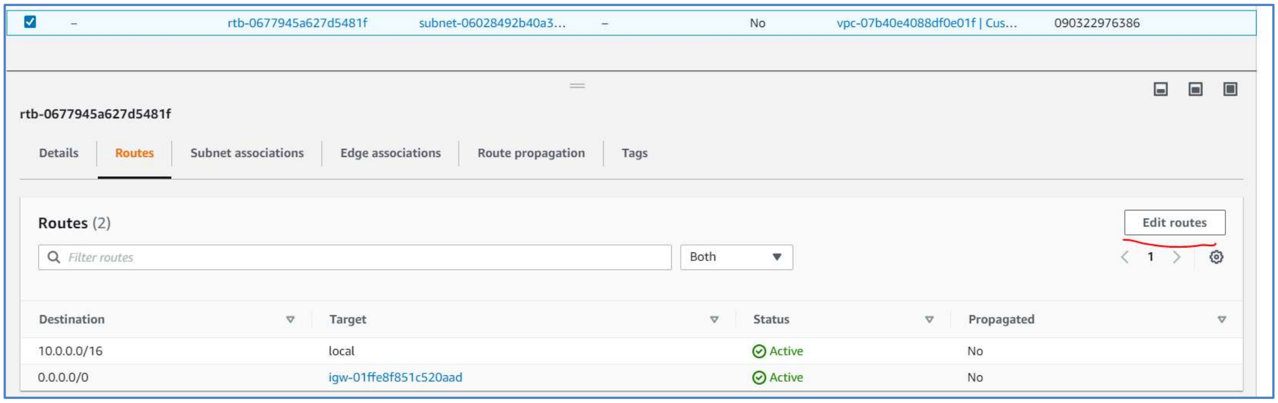
Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No



VPC > Route tables > rtb-04ed18ebf382767d6 > Edit routes

Edit routes

Destination	Target	Status	Propagated	
10.0.0.0/16	local	Active	No	
172.31.0.0/16	pcx-035a10702f5b2585f	-	No	Remove

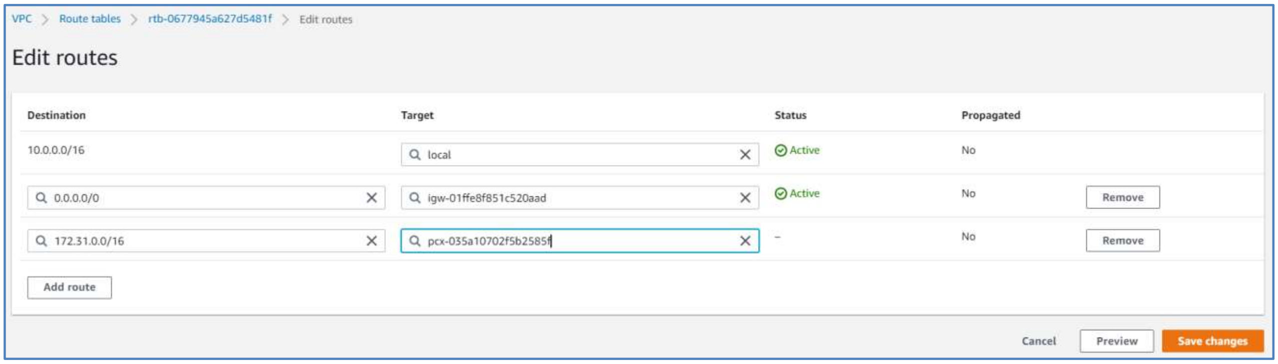


☑	–	rtb-0677945a627d5481f	subnet-06028492b40a3...	–	No	vpc-07b40e4088df0e01f | Cus...	090322976386

rtb-0677945a627d5481f

Routes (2)

Destination	Target	Status	Propagated
10.0.0.0/16	local	✓ Active	No
0.0.0.0/0	igw-01ffe8f851c520aad	✓ Active	No



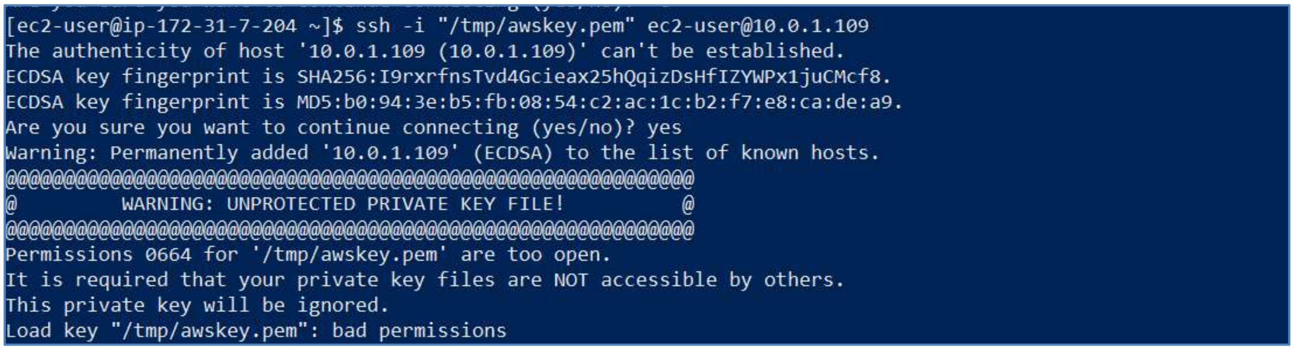
VPC > Route tables > rtb-0677945a627d5481f > Edit routes

Edit routes

Destination	Target	Status	Propagated	
10.0.0.0/16	local	✓ Active	No	
0.0.0.0/0	igw-01ffe8f851c520aad	✓ Active	No	Remove
172.31.0.0/16	pcx-035a10702f5b2585f	-	No	Remove

Step 10. Once all routes are added to both VPC's route tables, try to connect to the private instance from public instance now. You should be able to connect now. If you get the permission error like below:

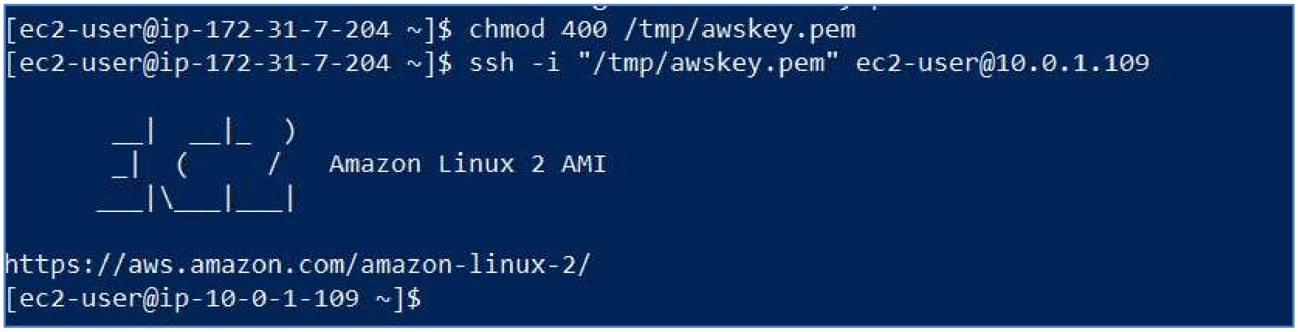
```
ssh -i "/tmp/awskey.pem" ec2-user@10.0.1.109
```



```
[ec2-user@ip-172-31-7-204 ~]$ ssh -i "/tmp/awskey.pem" ec2-user@10.0.1.109
The authenticity of host '10.0.1.109 (10.0.1.109)' can't be established.
ECDSA key fingerprint is SHA256:I9rxrfnsTvd4Gcieax25hQqizDsHfIZYWPx1juCMcf8.
ECDSA key fingerprint is MD5:b0:94:3e:b5:fb:08:54:c2:ac:1c:b2:f7:e8:ca:de:a9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.1.109' (ECDSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for '/tmp/awskey.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 "/tmp/awskey.pem": bad permissions
```

Step 11. Modify the permission of key file to 400 and try again:

```
chmod 400 /tmp/awskey.pem
```



```
[ec2-user@ip-172-31-7-204 ~]$ chmod 400 /tmp/awskey.pem
[ec2-user@ip-172-31-7-204 ~]$ ssh -i "/tmp/awskey.pem" ec2-user@10.0.1.109

       __|  __|_  )
       _|  (     /   Amazon Linux 2 AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-10-0-1-109 ~]$
```