8/5/2022

Project-03

1.Configuring Site to Site Connectivity on AWS.

2. Configuring Point to Site Connectivity on AWS.

3. Configuring Transit Gateway.

Sumit Mishra
SIC: 190310286

1. Configuring Site to Site connectivity on AWS. (AWS site only)

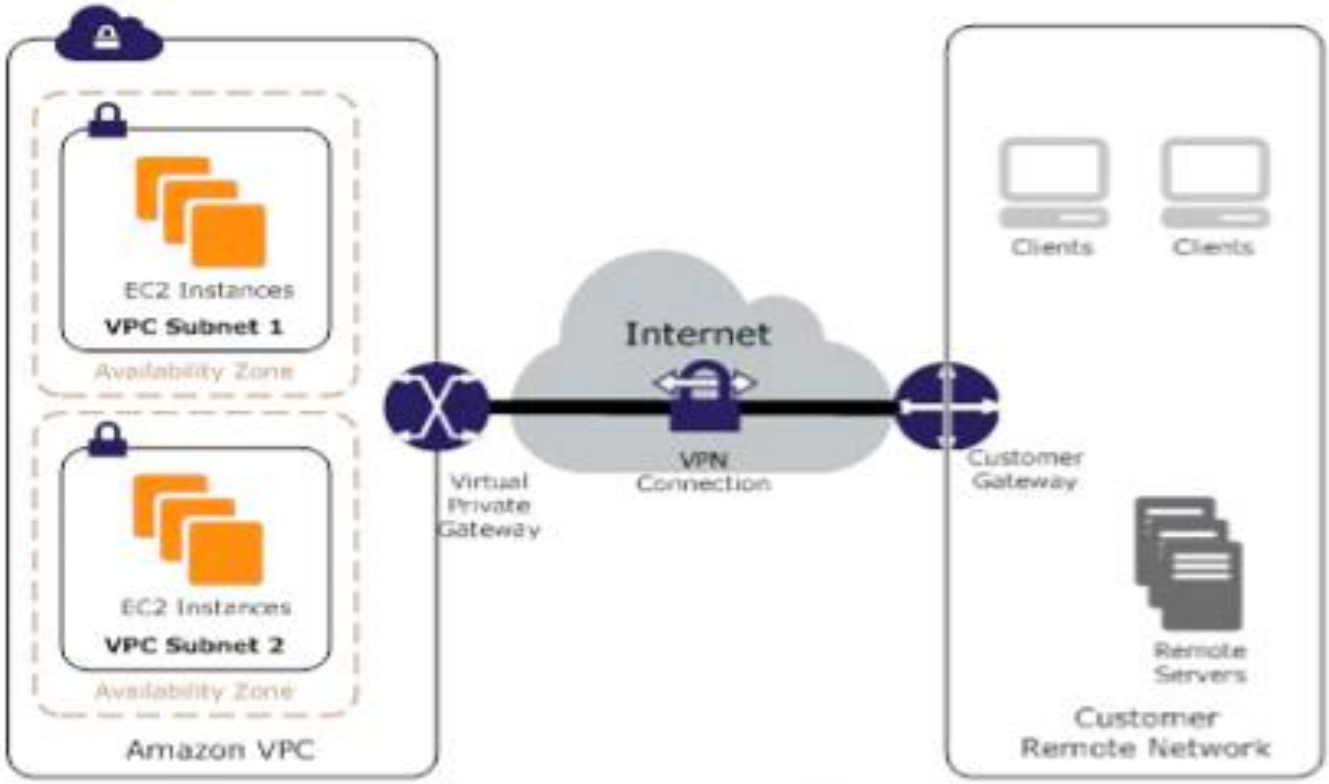
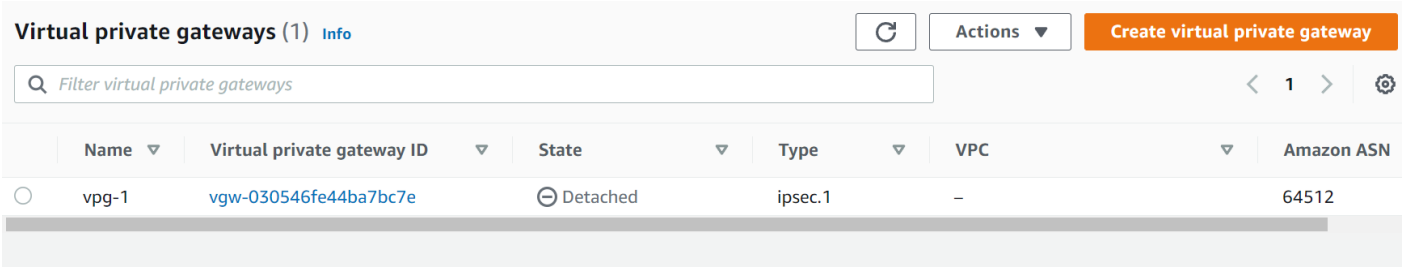
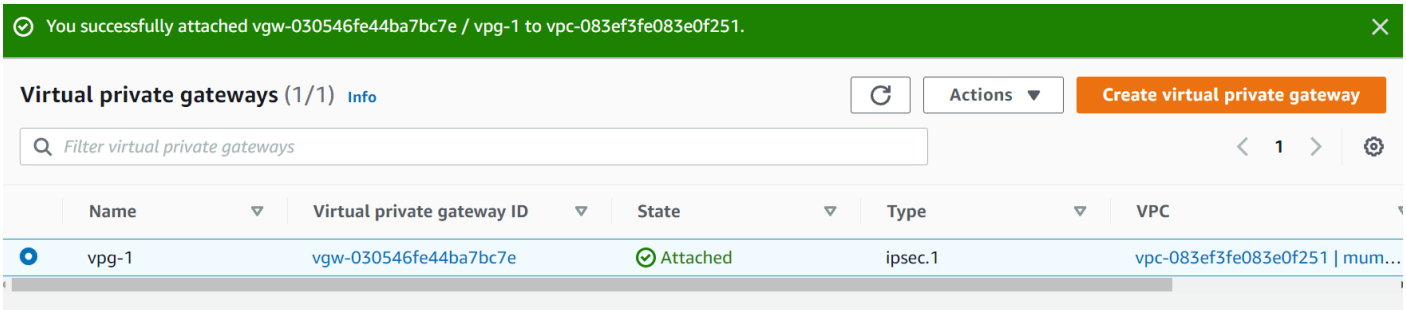


Figure 1: Hardware VPN

1. Configured Virtual Private Gateway for a VPC which acts as site 1.

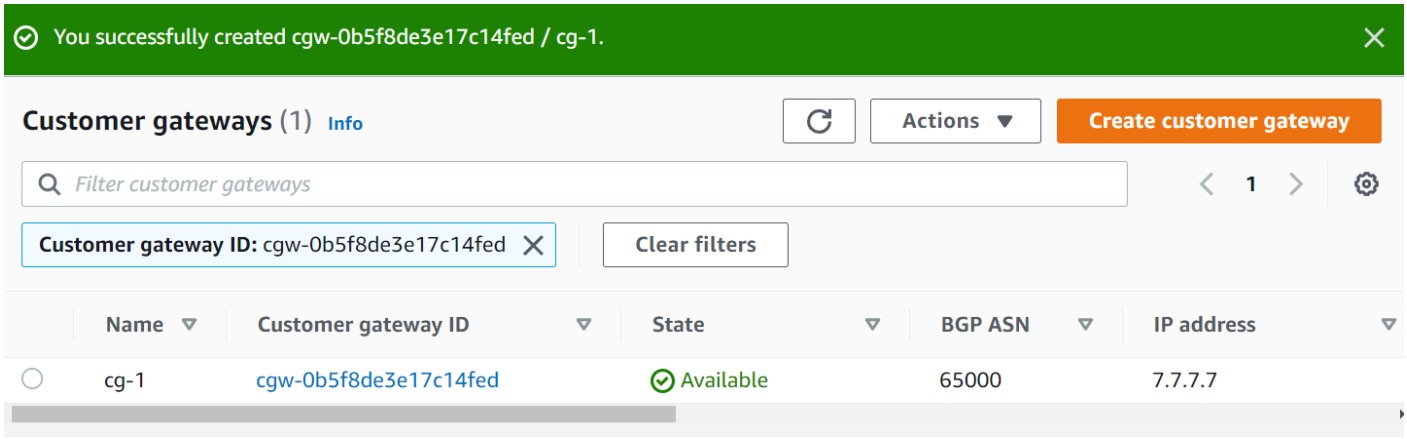


2. Attached the vpg-1 to Mumbai-vpc.

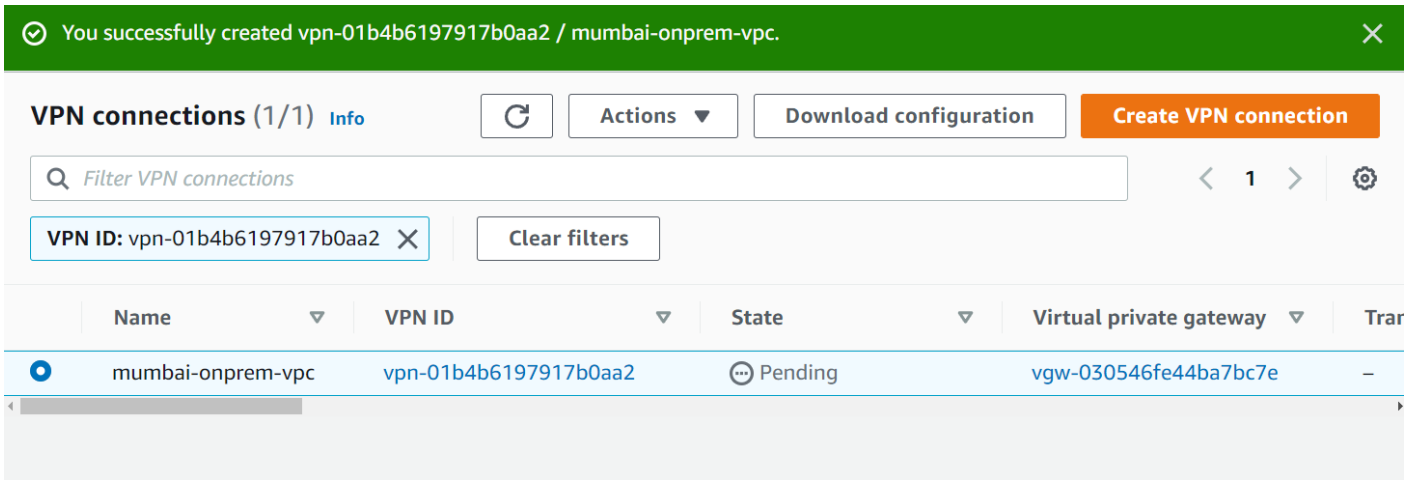


3. Created a Customer Gateway to attach to on-premises network acting as site 2.

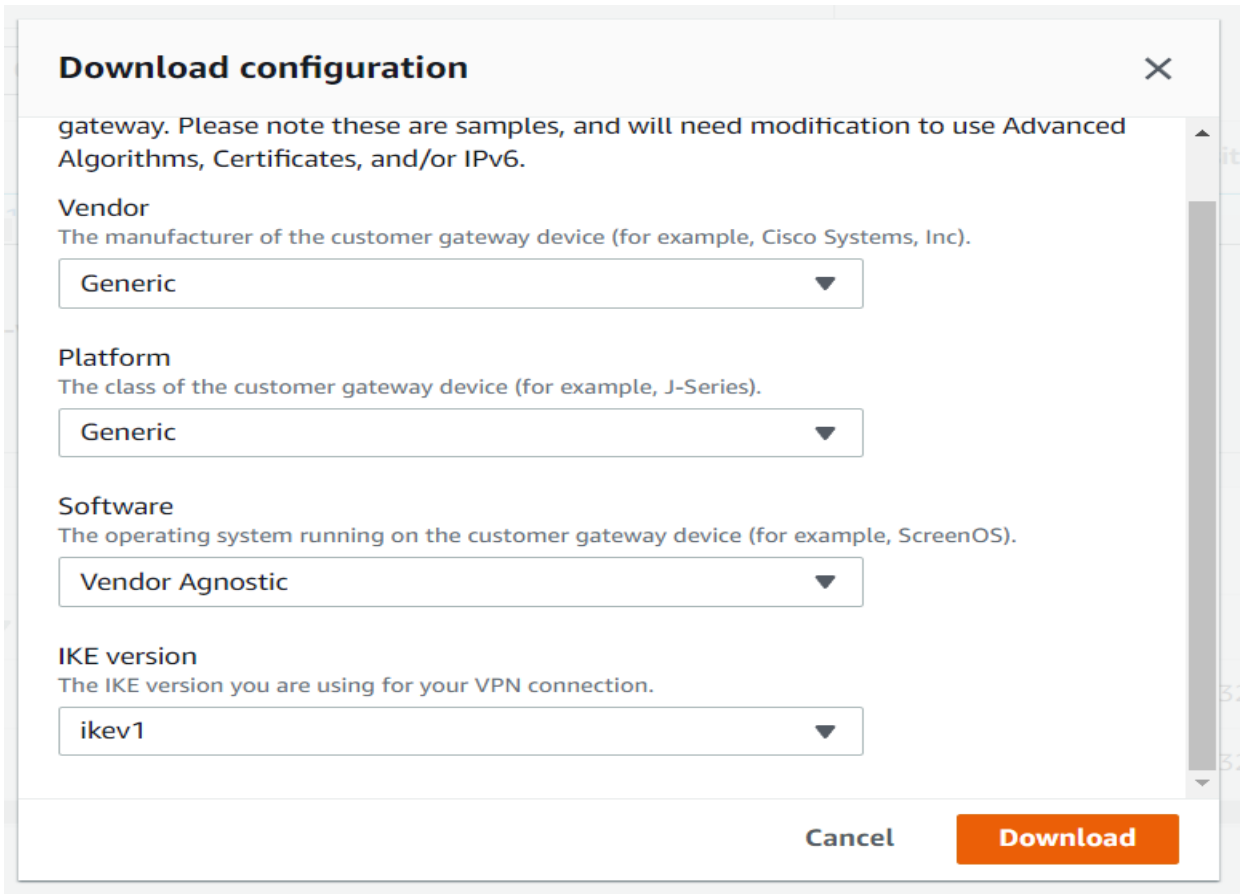
Name- Sumit Mishra
SIC: 190310286



4. Creating a Site to site connection using Site-to-Site VPN gateway.

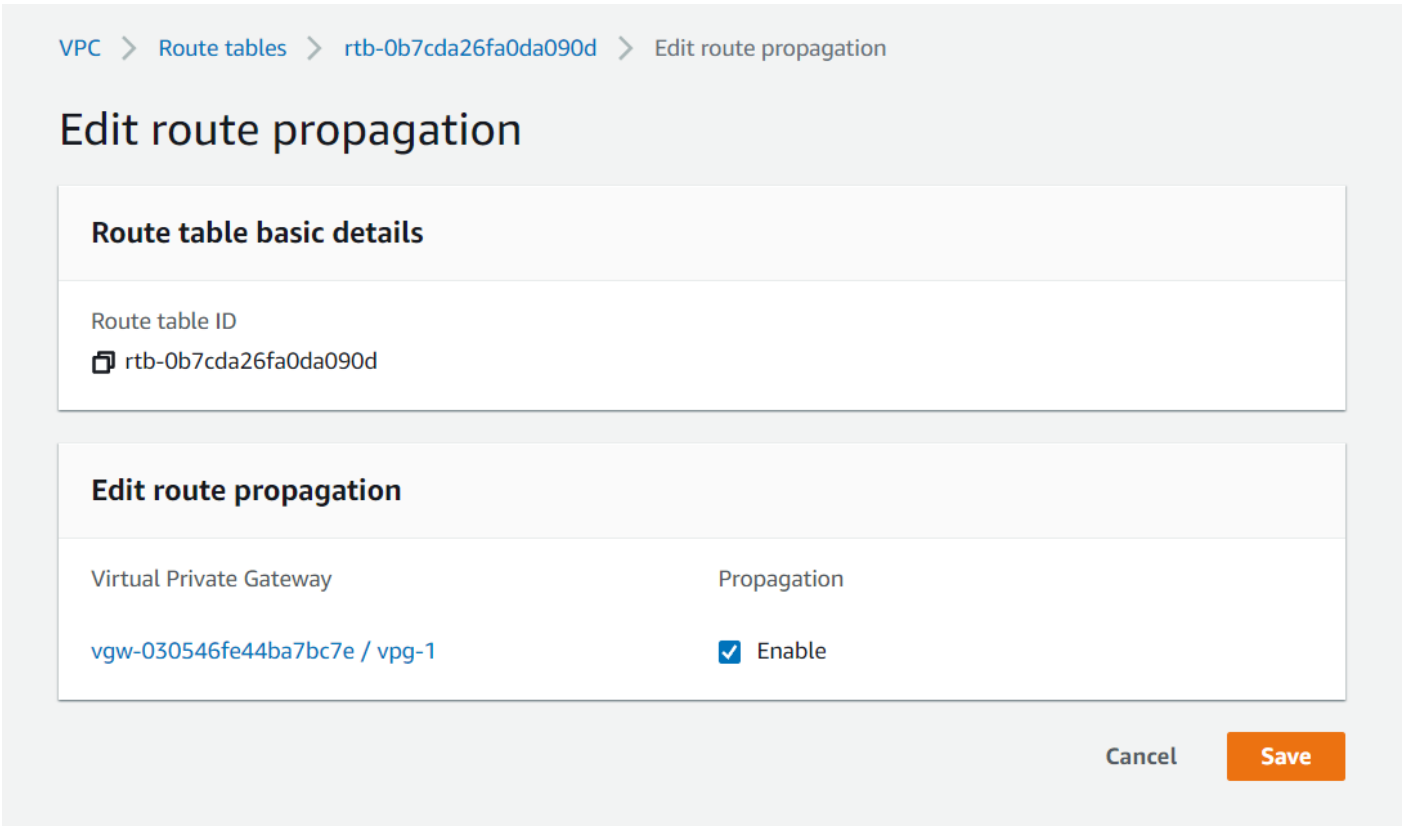


5. Downloading the Configuration file to be shared with the client.



6. Also enabled the route propagation in the Mumbai-vpc route table.

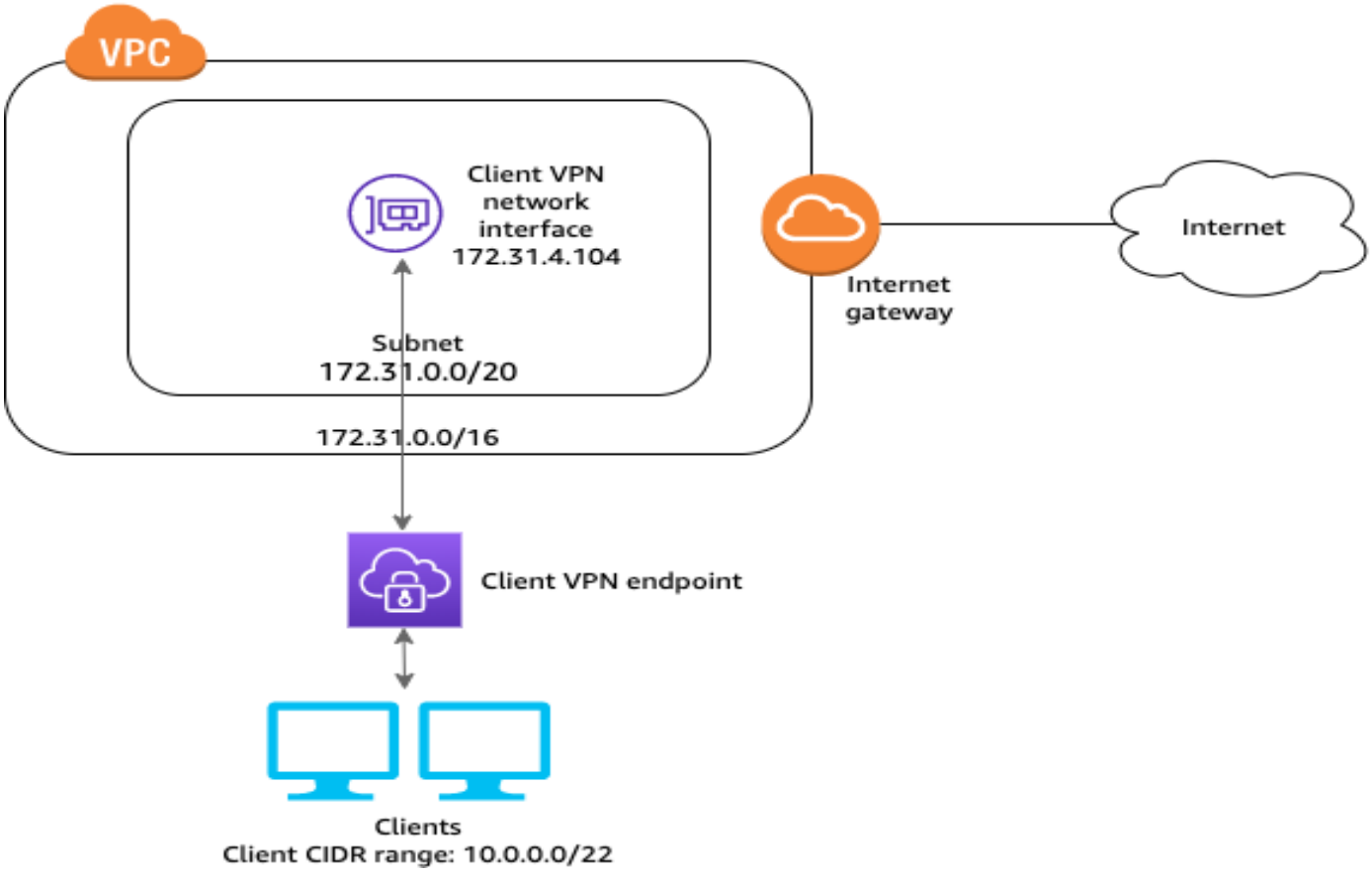
Name- Sumit Mishra
SIC: 190310286



This completed the Site-to-site connectivity on AWS.

Name- Sumit Mishra
SIC: 190310286

2. Set up Point to site connectivity on AWS.



1. Downloaded and installed open VPN connect.
2. Downloaded an installed Easy-RSA.
3. Renamed the extracted folder to EasyRSA3 then cut and pasted it in the folder of OpenVPN in Local Disk C:/ Program Files.
4. Opened Windows Terminal as Administrator and did the following to set up mutual authentication (server and client certificate).
    a. Navigated to the location where Easy-RSA folder was pasted.

```
PS C:\WINDOWS\system32> cd 'C:\Program Files\OpenVPN'
PS C:\Program Files\OpenVPN>
```

    b. Ran the following command on Command Prompt to activate the Easy-RSA Shell.

```
PS C:\Program Files\OpenVPN\EasyRSA> .\EasyRSA-Start.bat

Welcome to the EasyRSA 3 Shell for Windows.
Easy-RSA 3 is available under a GNU GPLv2 license.

Invoke './easyrsa' to call the program. Without commands, help is displayed.

EasyRSA Shell
#
```

    c. Initialized a new PKI environment.

```
# ./easyrsa init-pki
* Notice:

  init-pki complete; you may now create a CA or requests.

  Your newly created PKI dir is:
  * C:/Program Files/OpenVPN/EasyRSA/pki

* Notice:
  IMPORTANT: Easy-RSA 'vars' file has now been moved to your PKI above.
```

d. Ran the following commands to build a new certificate authority (CA).

i. Ran the following command, specified common name as test.

```
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]:
                                                              test

* Notice:

CA creation complete and you may now import and sign cert requests.
```

ii. Generated server certificate and key.

```
EasyRSA Shell
# ./easyrsa build-server-full server nopass
* Notice:
Using Easy-RSA configuration from: C:/Program Files/OpenVPN/EasyRSA/pki/vars

* Notice:
Using SSL: openssl OpenSSL 3.0.3 3 May 2022 (Library: OpenSSL 3.0.3 3 May 20
22)
```

```
* Notice:

Keypair and certificate request completed. Your files are:
req: C:/Program Files/OpenVPN/EasyRSA/pki/reqs/server.req
key: C:/Program Files/OpenVPN/EasyRSA/pki/private/server.key
```

iii. Generated client certificate and key.

```
EasyRSA Shell
# ./easyrsa build-client-full client1.domain.tld nopass
* Notice:
Using Easy-RSA configuration from: C:/Program Files/OpenVPN/EasyRSA/pki/vars

* Notice:
Using SSL: openssl OpenSSL 3.0.3 3 May 2022 (Library: OpenSSL 3.0.3 3 May 20
22)
```

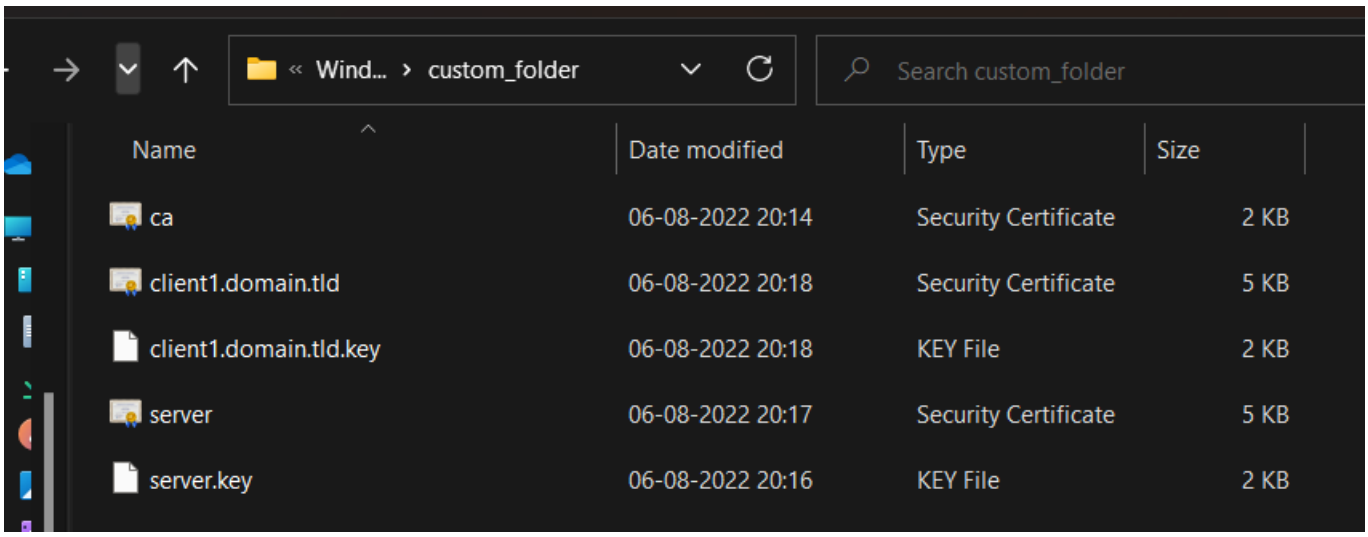
```
* Notice:

Keypair and certificate request completed. Your files are:
req: C:/Program Files/OpenVPN/EasyRSA/pki/reqs/client1.domain.tld.req
key: C:/Program Files/OpenVPN/EasyRSA/pki/private/client1.domain.tld.key
```

iv. Exited the EasyRSA3 shell.

e. Copied the server certificate and key and the client certificate and key to a custom folder by running the following commands.

```
PS C:\Program Files\OpenVPN\EasyRSA> mkdir C:\custom_folder


    Directory: C:\


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        06-08-2022     20:21                custom_folder


PS C:\Program Files\OpenVPN\EasyRSA> copy pki\ca.crt C:\custom_folder
PS C:\Program Files\OpenVPN\EasyRSA> copy pki\issued\server.crt C:\custom_folder
PS C:\Program Files\OpenVPN\EasyRSA> copy pki\private\server.key C:\custom_folder
PS C:\Program Files\OpenVPN\EasyRSA> copy pki\issued\client1.domain.tld.crt C:\custom_folder
PS C:\Program Files\OpenVPN\EasyRSA> copy pki\private\client1.domain.tld.key C:\custom_folder
PS C:\Program Files\OpenVPN\EasyRSA> cd C:\custom_folder
```

We can see the generated certificates in the file explorer, inside the custom_folder.



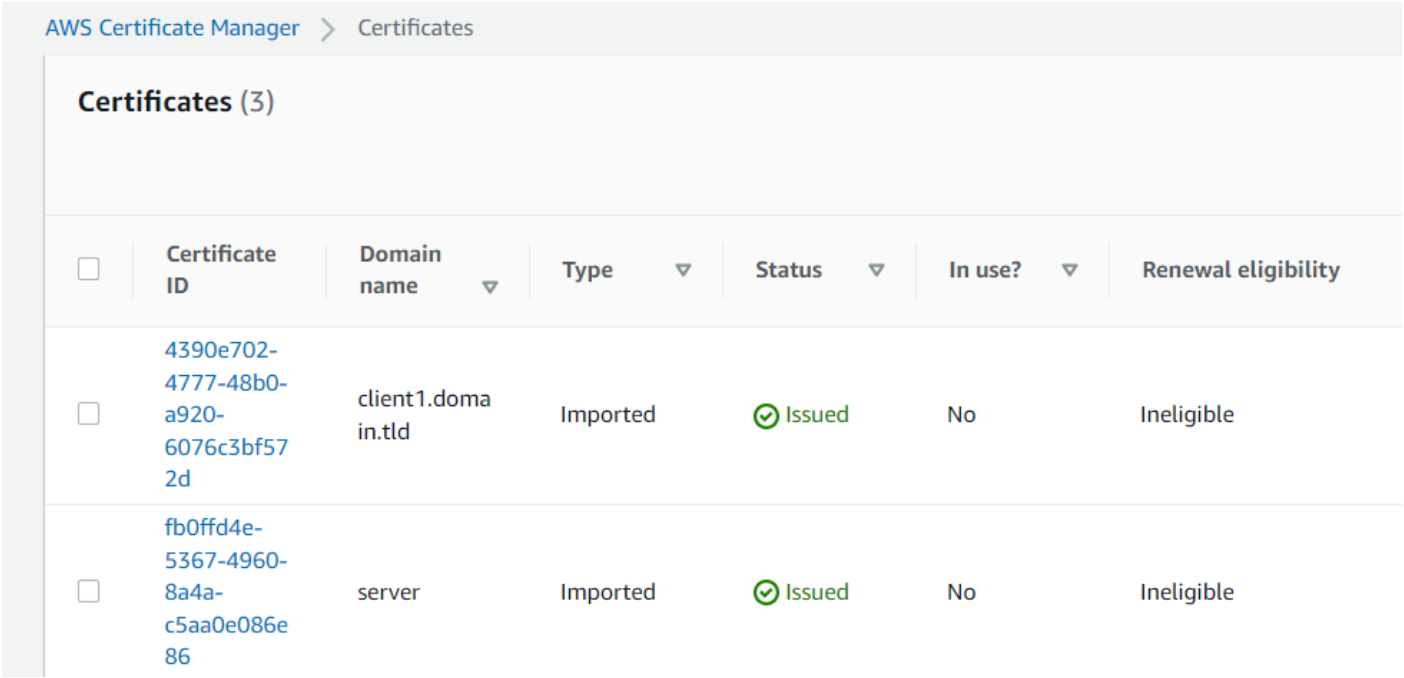
f. Generated an IAM user for point-to-site connectivity and allowed it Administrator Access, then configured the user inside the custom folder.

g. Uploaded the server certificate and key and client certificate and key to ACM using the following commands.

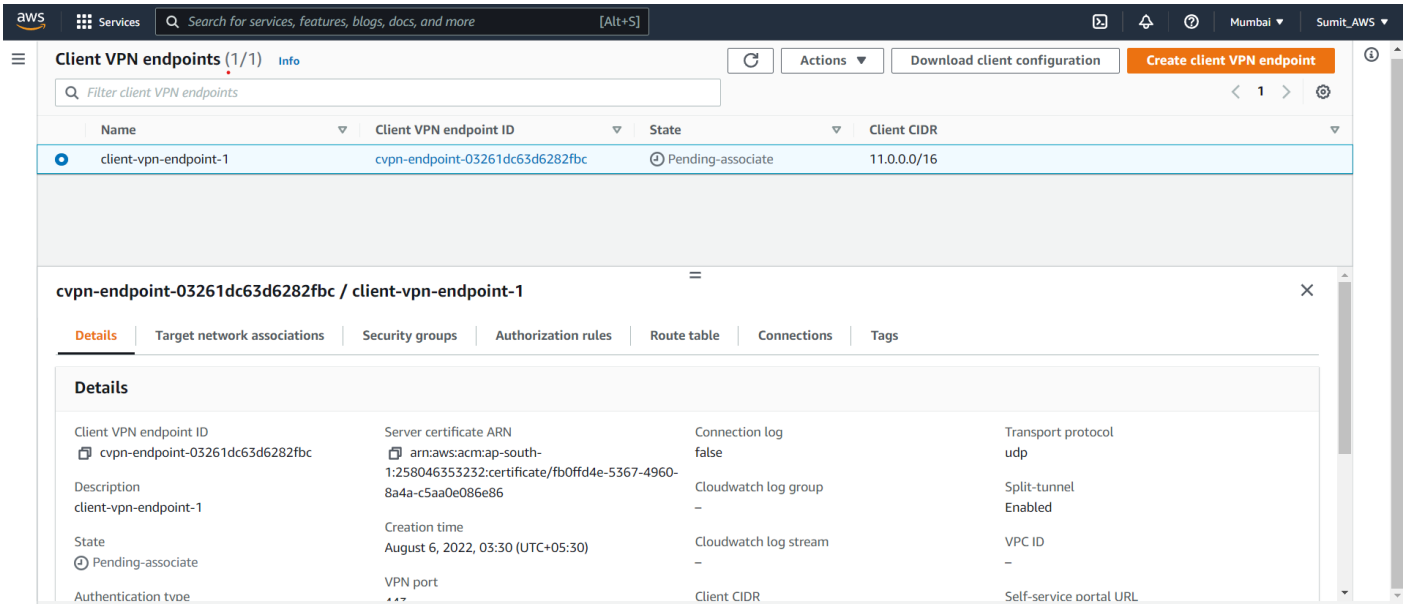
```
PS C:\custom_folder> aws acm import-certificate --certificate fileb://server
.crt --private-key fileb://server.key --certificate-chain fileb://ca.crt
{
    "CertificateArn": "arn:aws:acm:ap-south-1:258046353232:certificate/fb0ff
d4e-5367-4960-8a4a-c5aa0e086e86"
}

PS C:\custom_folder>
```

```
PS C:\custom_folder> aws acm import-certificate --certificate fileb://client
1.domain.tld.crt --private-key fileb://client1.domain.tld.key --certificate-
chain fileb://ca.crt
{
    "CertificateArn": "arn:aws:acm:ap-south-1:258046353232:certificate/4390e
702-4777-48b0-a920-6076c3bf572d"
}

PS C:\custom_folder>
```

We can now see the certificates in the ACM section in AWS console.

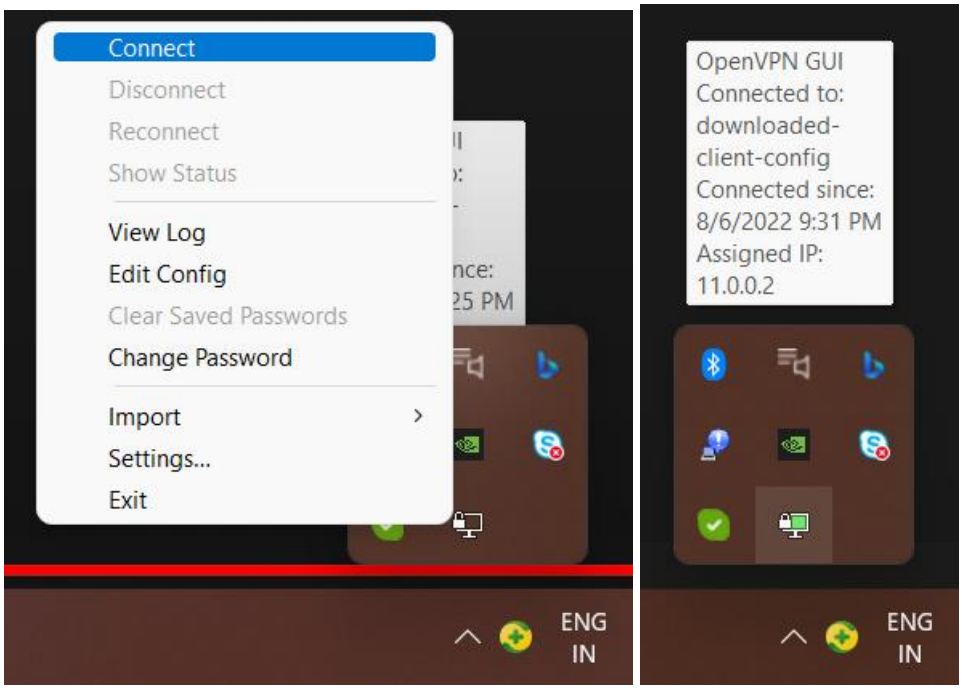


5. Created a client VPN endpoint using AWS console (while configuring make sure to enable split tunnelling).



6. Associated a target network to Mumbai-VPC.
7. Added an authorization rule to the Client VPN endpoint.

8. While the client VPN endpoint was configured, I launched an EC2 instance inside the Mumbai-VPC, I did not provide a public IP for connecting to the Instance.
9. When the client-VPN-endpoint became active, I downloaded the client configuration file and performed the following steps.
    a. Opened the configuration file using notepad and inserted the following at the end.
    b. Inserted the client certificate file's path.
    c. Inserted the client key file's path.
10. Saved the file and moved the configuration file to config folder inside OpenVPN folder.
11. Connected to the server using OpenVPN client GUI present in hidden icons in the taskbar.



12. After connection was successful, I connected to the EC2 instance using its private IP from my computer using xshell and was successful.

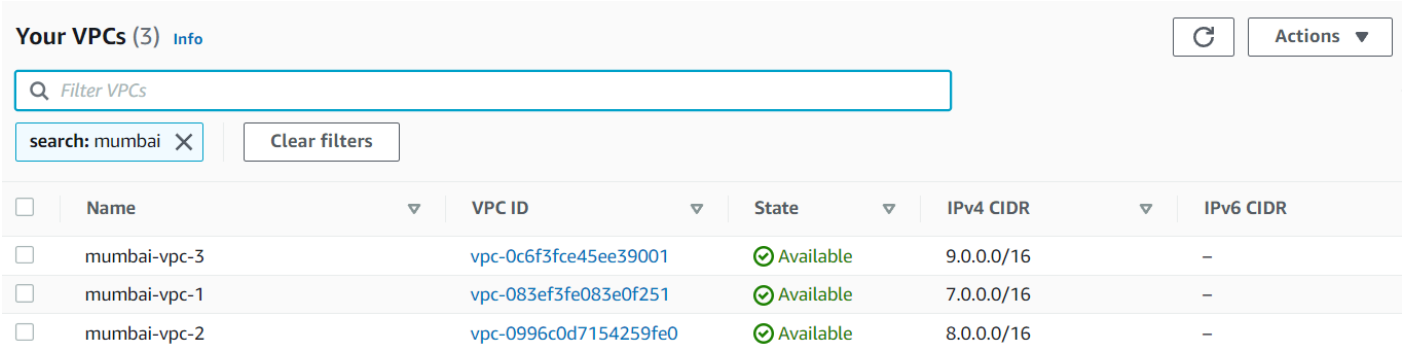
```
Xshell 7 (Build 0111)
Copyright (c) 2020 NetSarang Computer, Inc. All rights reserved.

Type `help' to learn how to use Xshell prompt.
[C:\~]$ ssh -i "VPC-server-sumit.pem" ec2-user@7.0.1.40


Connecting to 7.0.1.40:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

WARNING! The remote SSH server rejected X11 forwarding request.

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

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-7-0-1-40 ~]$
```

13. So, I can say that I successfully configured Point-to-Site connectivity using AWS.

Name- Sumit Mishra
SIC: 190310286

3. Transit Gateway setup.

1. Created 3 VPCs in the Mumbai region.



Your VPCs (3) Info

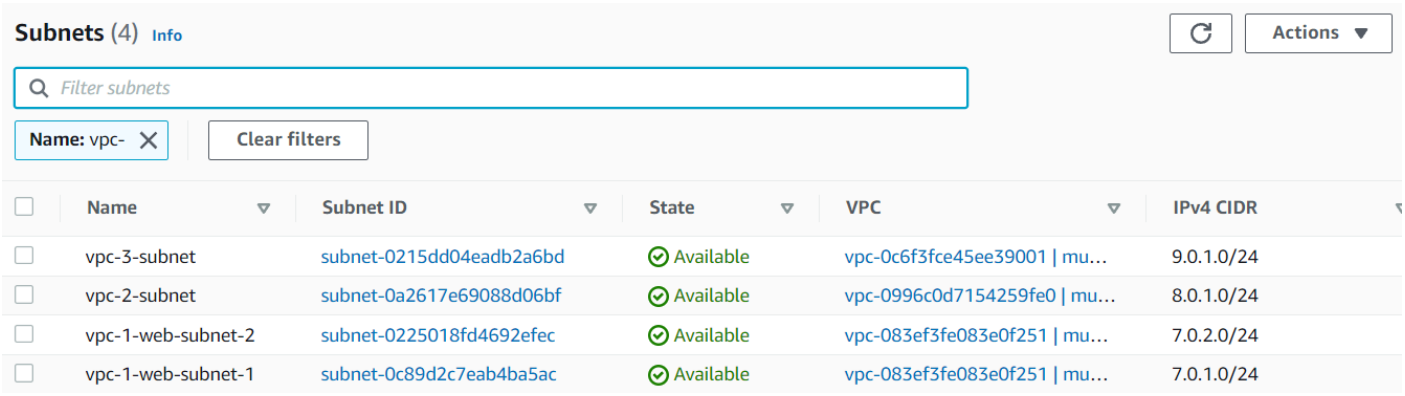
Actions

Filter VPCs

search: mumbai | Clear filters

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
mumbai-vpc-3	vpc-0c6f3fce45ee39001	Available	9.0.0.0/16	–
mumbai-vpc-1	vpc-083ef3fe083e0f251	Available	7.0.0.0/16	–
mumbai-vpc-2	vpc-0996c0d7154259fe0	Available	8.0.0.0/16	–

2. Created subnets in the 3 VPCs.



Subnets (4) Info

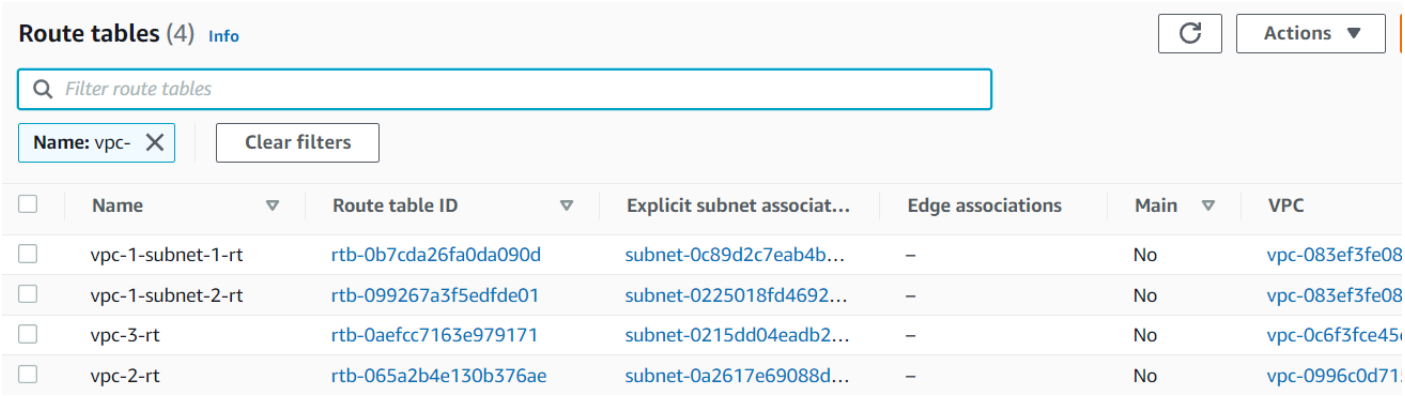
Actions

Filter subnets

Name: vpc- | Clear filters

Name	Subnet ID	State	VPC	IPv4 CIDR
vpc-3-subnet	subnet-0215dd04eadb2a6bd	Available	vpc-0c6f3fce45ee39001 | mu...	9.0.1.0/24
vpc-2-subnet	subnet-0a2617e69088d06bf	Available	vpc-0996c0d7154259fe0 | mu...	8.0.1.0/24
vpc-1-web-subnet-2	subnet-0225018fd4692efec	Available	vpc-083ef3fe083e0f251 | mu...	7.0.2.0/24
vpc-1-web-subnet-1	subnet-0c89d2c7eab4ba5ac	Available	vpc-083ef3fe083e0f251 | mu...	7.0.1.0/24

3. Created route tables for the 3 VPCs and associated them with their respective subnets.



Route tables (4) Info

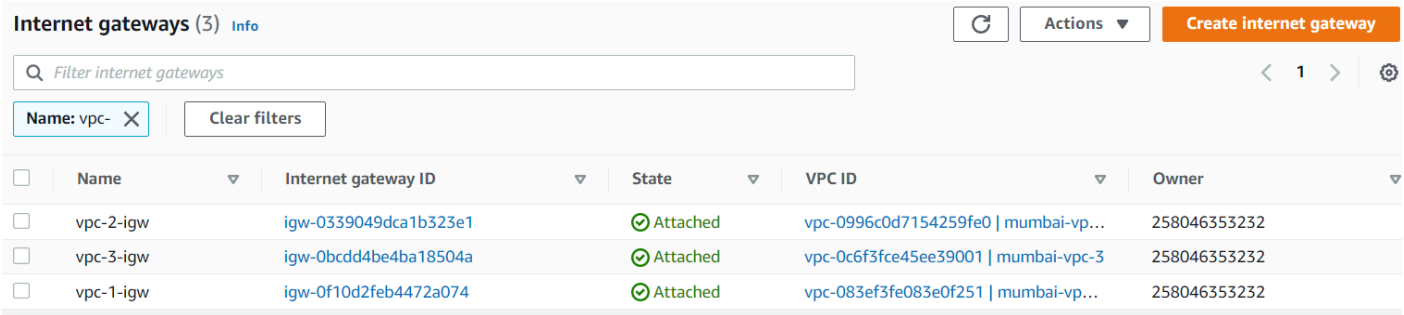
Actions

Filter route tables

Name: vpc- | Clear filters

Name	Route table ID	Explicit subnet associat...	Edge associations	Main	VPC
vpc-1-subnet-1-rt	rtb-0b7cda26fa0da090d	subnet-0c89d2c7eab4b...	–	No	vpc-083ef3fe08
vpc-1-subnet-2-rt	rtb-099267a3f5edfde01	subnet-0225018fd4692...	–	No	vpc-083ef3fe08
vpc-3-rt	rtb-0aefcc7163e979171	subnet-0215dd04eadb2...	–	No	vpc-0c6f3fce45
vpc-2-rt	rtb-065a2b4e130b376ae	subnet-0a2617e69088d...	–	No	vpc-0996c0d71

4. Created internet gateways for the 3 VPCs, attached them to their respective VPCs and added the respective internet gateways to the respective route tables.



Internet gateways (3) Info

Actions | Create internet gateway

Filter internet gateways

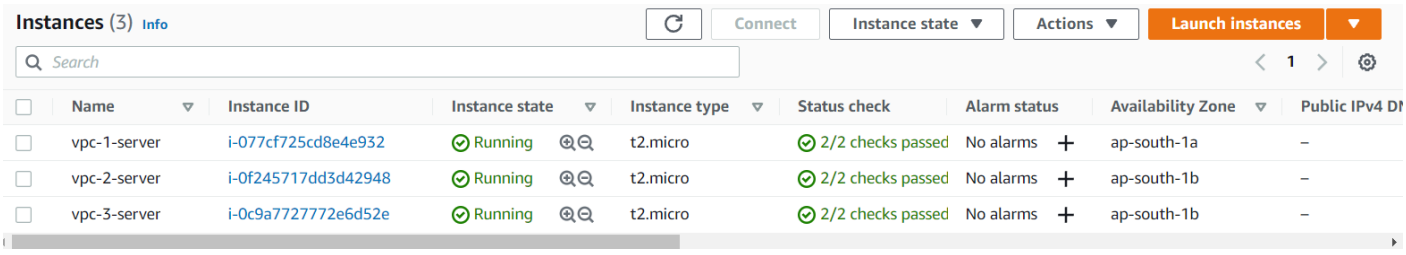
1

Name: vpc- | Clear filters

Name	Internet gateway ID	State	VPC ID	Owner
vpc-2-igw	igw-0339049dca1b323e1	Attached	vpc-0996c0d7154259fe0 | mumbai-vp...	258046353232
vpc-3-igw	igw-0bcdd4be4ba18504a	Attached	vpc-0c6f3fce45ee39001 | mumbai-vpc-3	258046353232
vpc-1-igw	igw-0f10d2feb4472a074	Attached	vpc-083ef3fe083e0f251 | mumbai-vp...	258046353232

5. Created an instance inside each VPC, enabling public IP access to instance inside VPC-1 only.

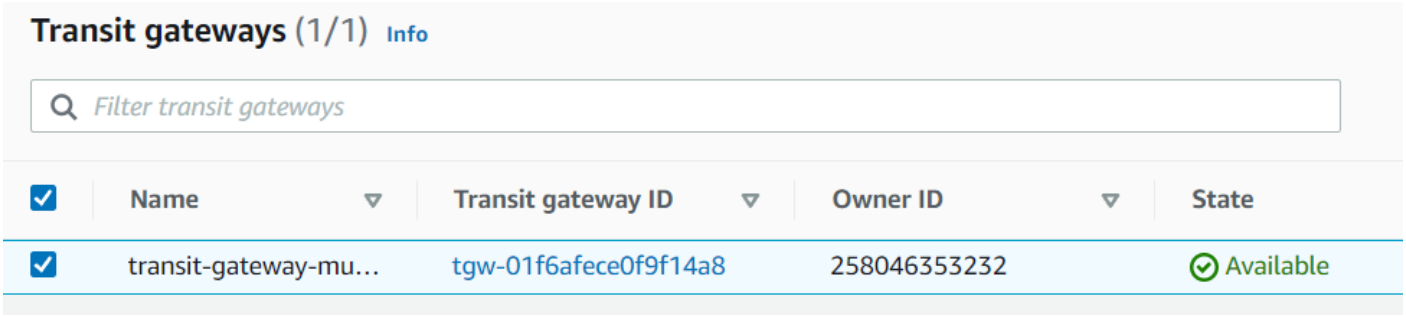
Name- Sumit Mishra
SIC: 190310286



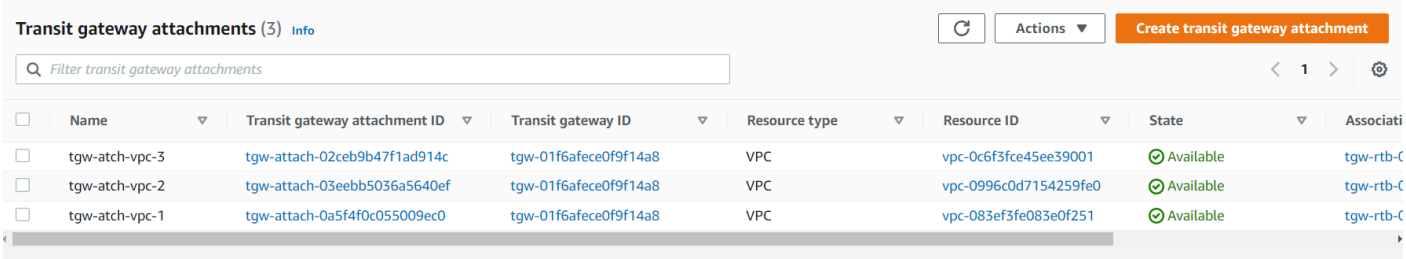
a. Now I connected to the instance inside Mumbai-vpc-1 and tried to ping the other two instances, but it failed.

```
[root@ip-7-0-1-38 ~]# ping 8.0.1.227
PING 8.0.1.227 (8.0.1.227) 56(84) bytes of data.
^C
--- 8.0.1.227 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3052ms

[root@ip-7-0-1-38 ~]# ping 9.0.1.98
PING 9.0.1.98 (9.0.1.98) 56(84) bytes of data.
^C
--- 9.0.1.98 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2048ms
```

6. Created a transit gateway and used it to create routes in each route table of the VPCs in order to access the other VPCs.
   a. Created a transit gateway.



b. Created transit gateway attachments for each VPC.

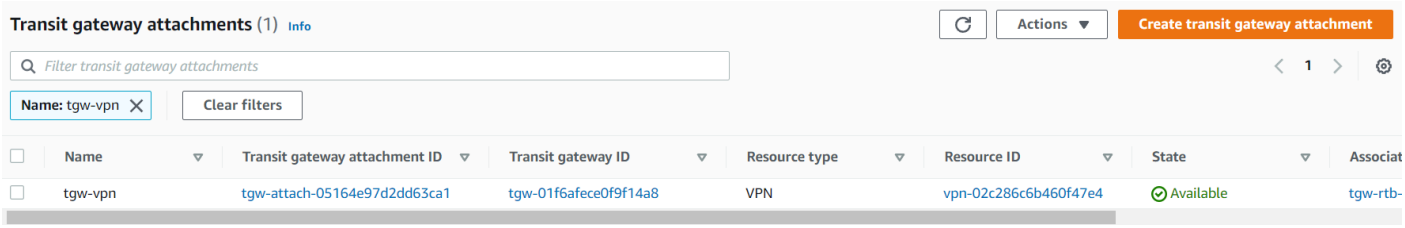


c. Attachments were attached to the individual route tables of the VPCs.
d. Now, I tried to ping the other two instances using their private IP, it successfully pinged the information.

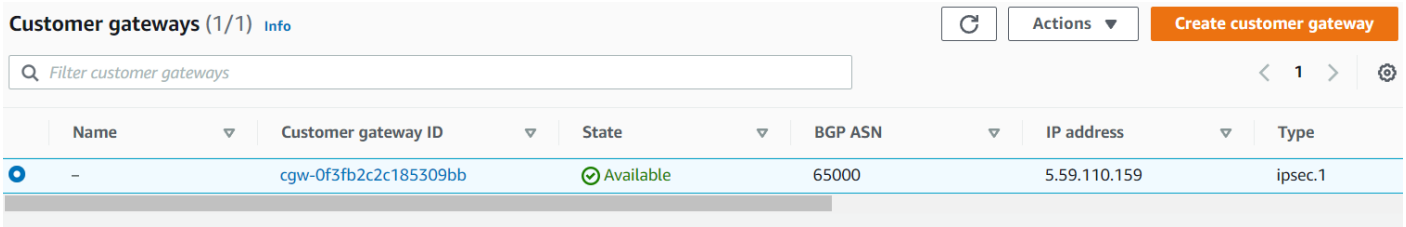
Name- Sumit Mishra
SIC: 190310286

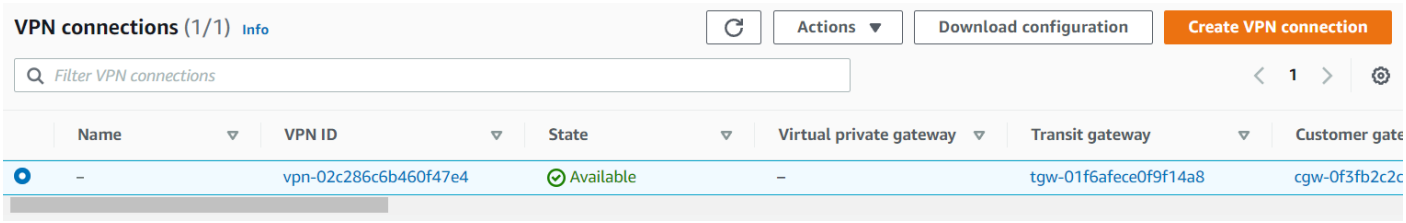
```
[root@ip-7-0-1-38 ~]# ping 9.0.1.98
PING 9.0.1.98 (9.0.1.98) 56(84) bytes of data.
64 bytes from 9.0.1.98: icmp_seq=1 ttl=254 time=1.56 ms
64 bytes from 9.0.1.98: icmp_seq=2 ttl=254 time=1.26 ms
^C
--- 9.0.1.98 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 1.262/1.413/1.564/0.151 ms
[root@ip-7-0-1-38 ~]# ping 8.0.1.227
PING 8.0.1.227 (8.0.1.227) 56(84) bytes of data.
64 bytes from 8.0.1.227: icmp_seq=1 ttl=254 time=1.63 ms
64 bytes from 8.0.1.227: icmp_seq=2 ttl=254 time=0.928 ms
^C
--- 8.0.1.227 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.928/1.281/1.634/0.353 ms
[root@ip-7-0-1-38 ~]#
```

7. Created a new VPN attachment for site-to-site connectivity purpose.



a. In the backend AWS will create a customer gateway and site-to-site gateway and enable routing.





b. We have to download the client configuration file and give it to the client side engineers to configure the connection.
c. We can do some CIDR changes in the transit gateway routing table for site-to-site connectivity.

8. Transit Gateway set up was configured for the 3 VPCs successfully.