8/5/2022

Sumit Mishra

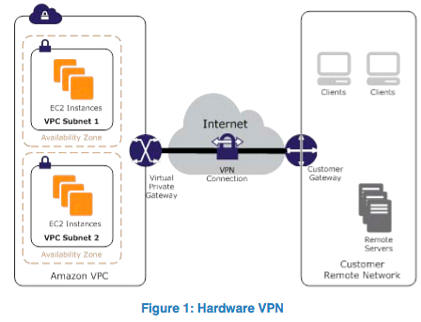
SIC: 190310286



**Project-03**

1.Configuring Site to Site Connectivity on AWS. 2. Configuring Point to Site Connectivity on AWS. 3. Configuring Transit Gateway.

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



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

Graphical user interface, text, application

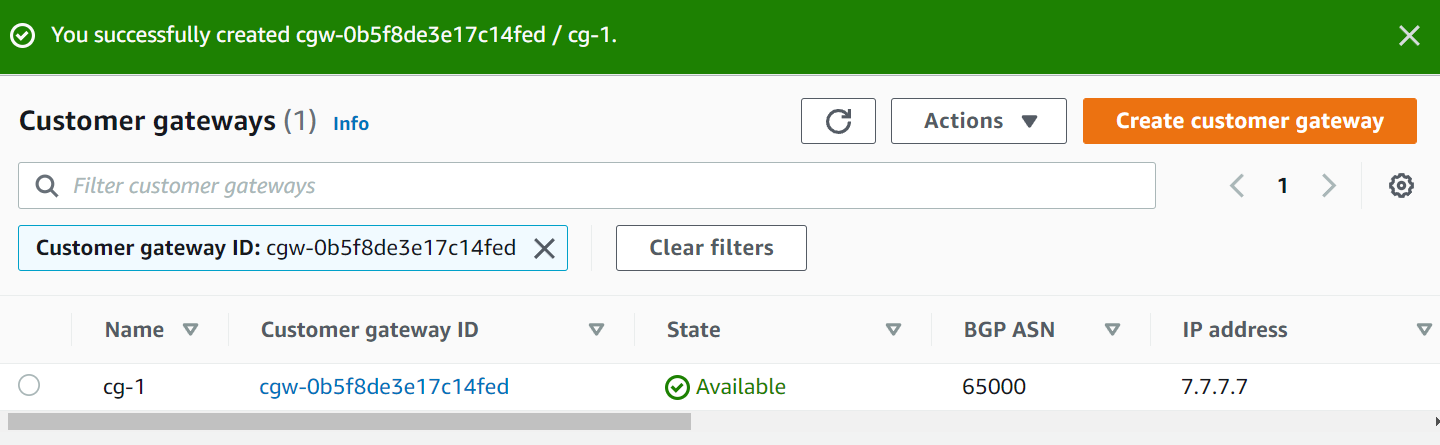
Description automatically generated

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

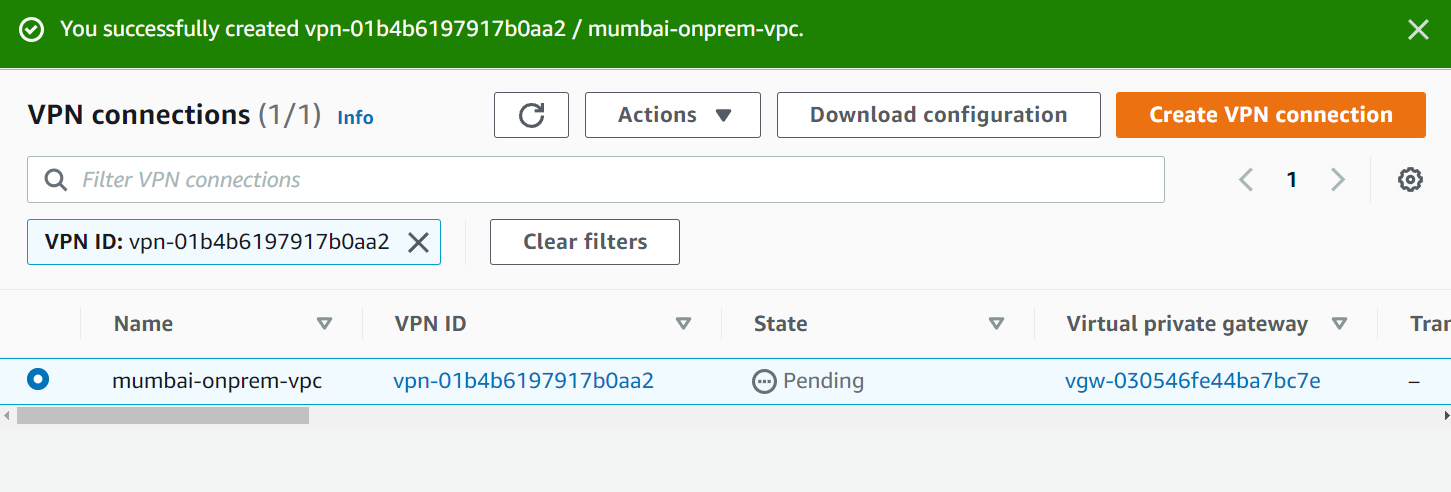
Graphical user interface, text, application

Description automatically generated

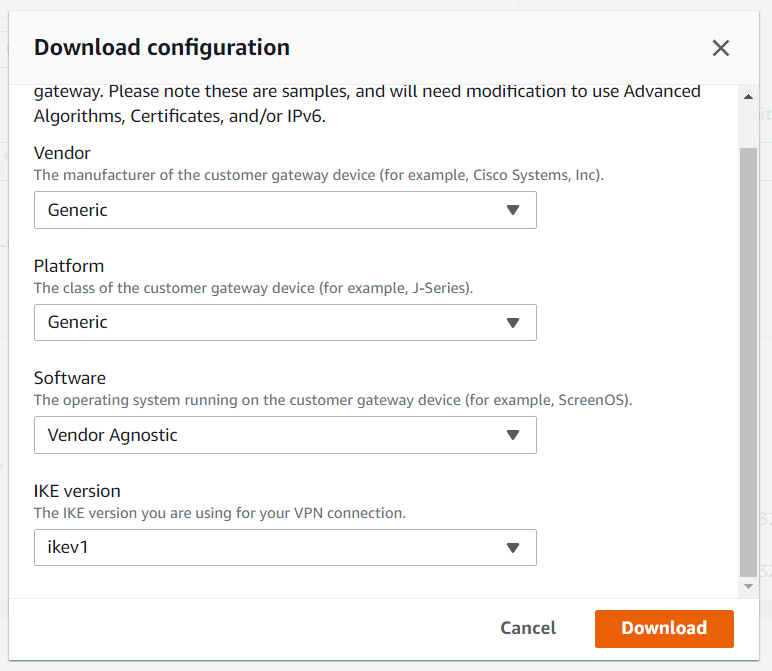
1. Created a Customer Gateway to attach to on-premises network acting as site 2.



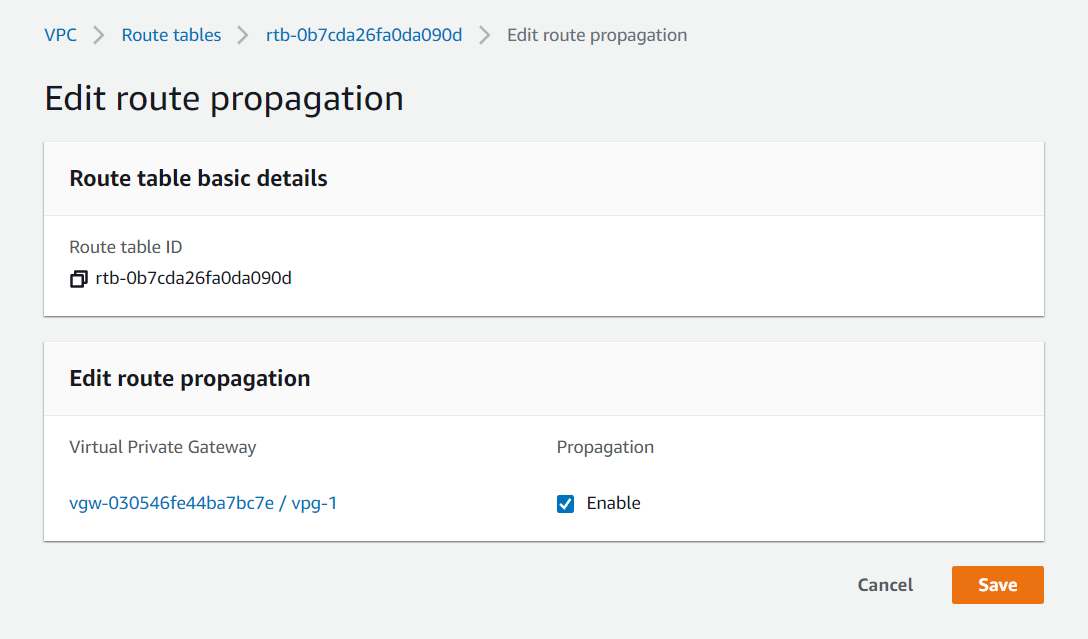
1. Creating a Site to site connection using Site-to-Site VPN gateway.



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



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



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

# Set up Point to site connectivity on AWS.


   Client VPN accessing the internet
  

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).
   1. Navigated to the location where Easy-RSA folder was pasted.



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

Text

Description automatically generated

* 1. Initialized a new PKI environment.

Text

Description automatically generated

* 1. Ran the following commands to build a new certificate authority (CA).
     1. Ran the following command, specified common name as test.

Text

Description automatically generated

* + 1. Generated server certificate and key.

Text

Description automatically generated

Text

Description automatically generated

* + 1. Generated client certificate and key.

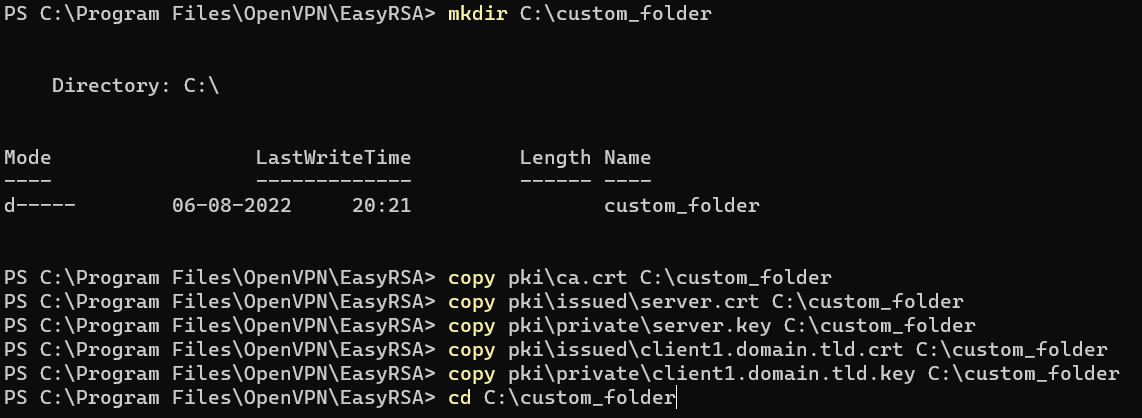
Text

Description automatically generated

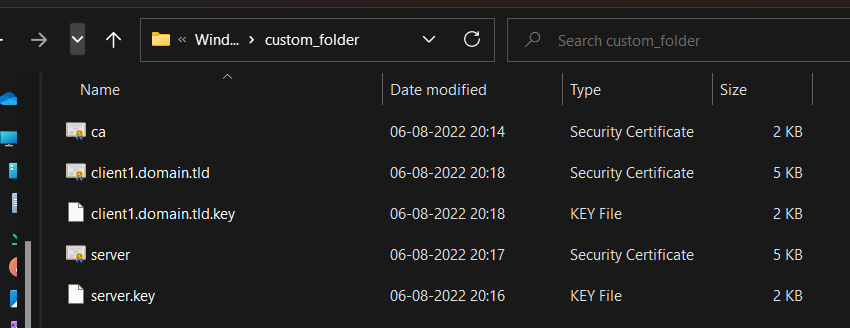
Text

Description automatically generated

* + 1. Exited the EasyRSA3 shell.
  1. Copied the server certificate and key and the client certificate and key to a custom folder by running the following commands.



We can see the generated certificates in the file explorer, inside the custom\_folder.



* 1. Generated an IAM user for point-to-site connectivity and allowed it Administrator Access, then configured the user inside the custom folder.
  2. Uploaded the server certificate and key and client certificate and key to ACM using the following commands.

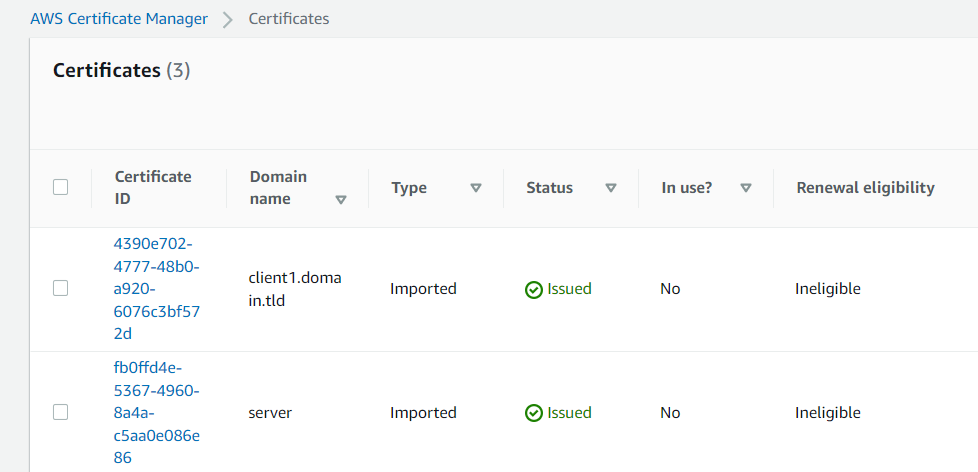
Text

Description automatically generated

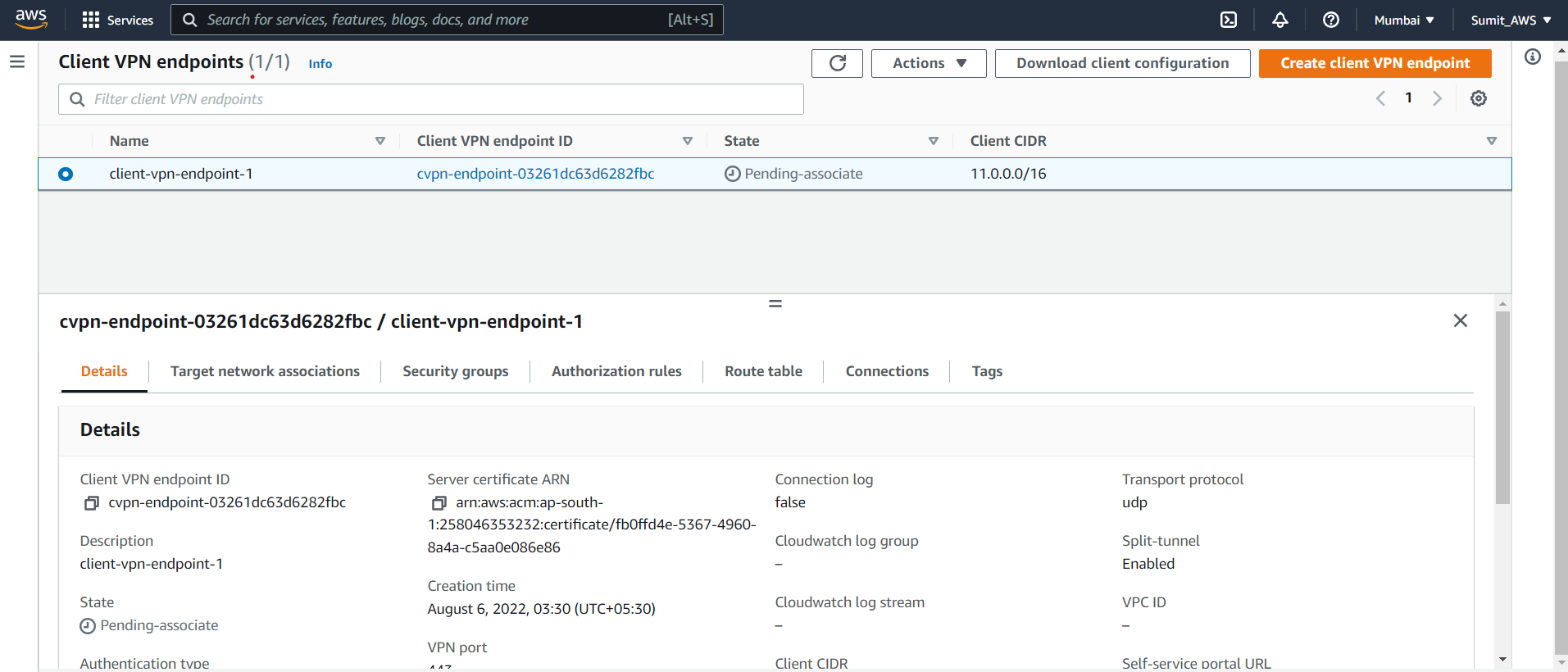
Text

Description automatically generated

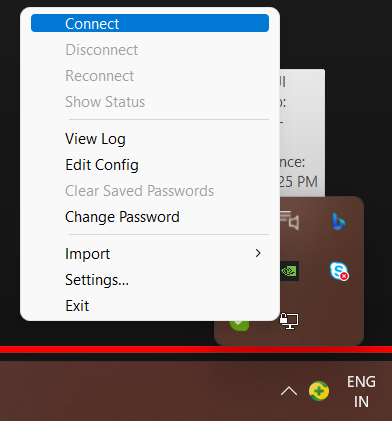
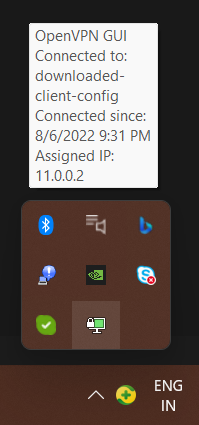
We can now see the certificates in the ACM section in AWS console.



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



1. Associated a target network to Mumbai-VPC.
2. Added an authorization rule to the Client VPN endpoint.
3. 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.
4. When the client-VPN-endpoint became active, I downloaded the client configuration file and performed the following steps.
   1. Opened the configuration file using notepad and inserted the following at the end.
   2. Inserted the client certificate file’s path.
   3. Inserted the client key file’s path.
5. Saved the file and moved the configuration file to config folder inside OpenVPN folder.
6. Connected to the server using OpenVPN client GUI present in hidden icons in the taskbar.

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

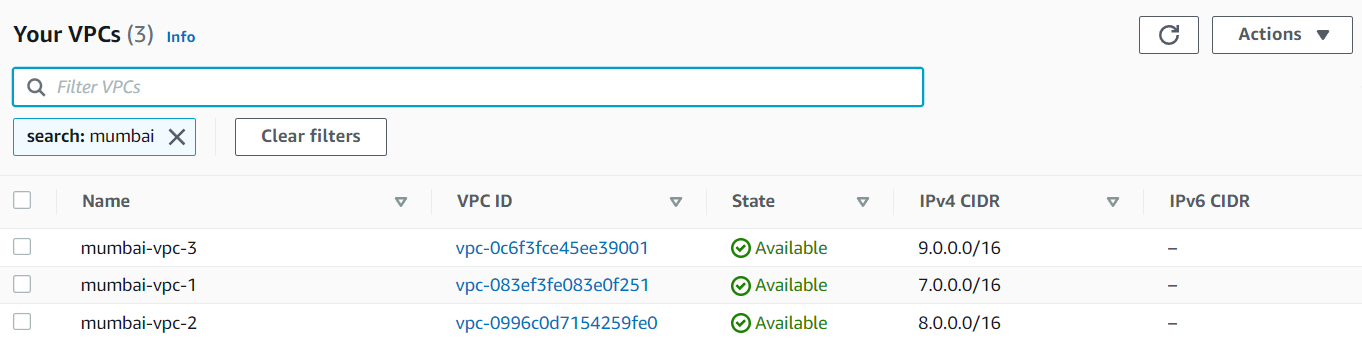
Text

Description automatically generated

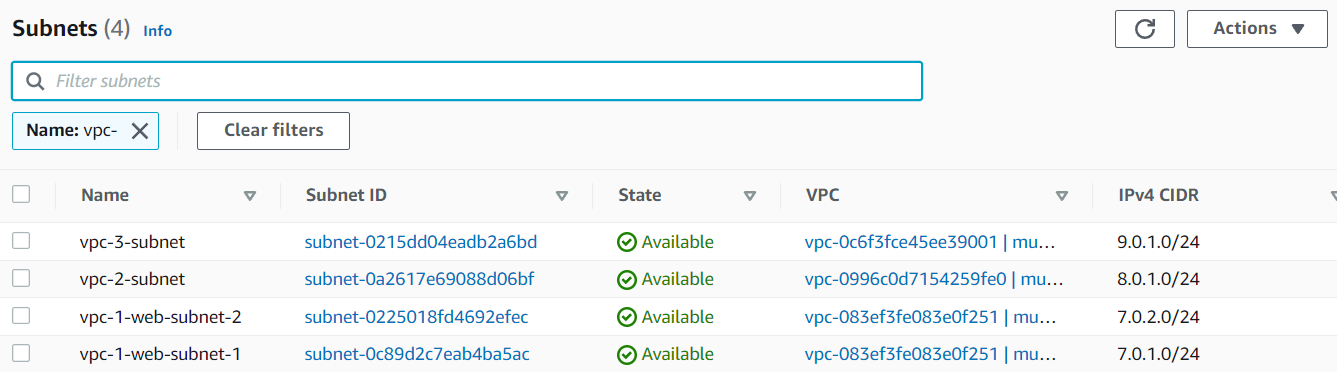
1. So, I can say that I successfully configured Point-to-Site connectivity using AWS.

# Transit Gateway setup.

1. Created 3 VPCs in the Mumbai region.



1. Created subnets in the 3 VPCs.



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

Graphical user interface, text, application, email

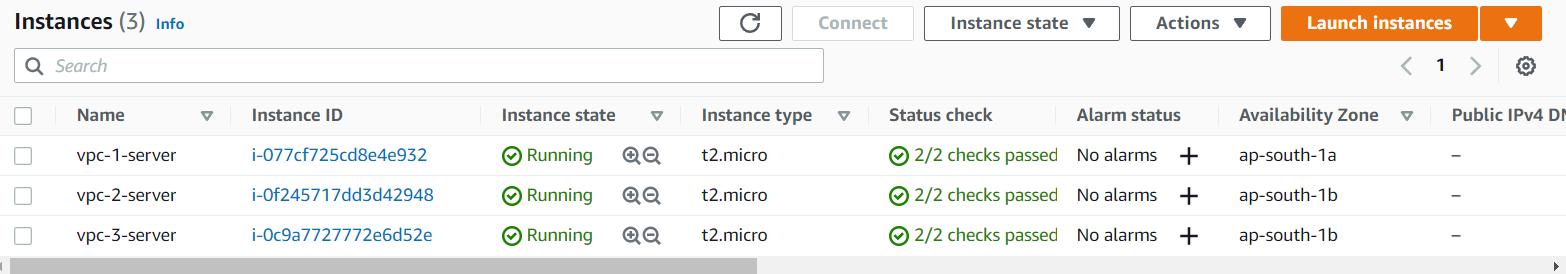
Description automatically generated

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

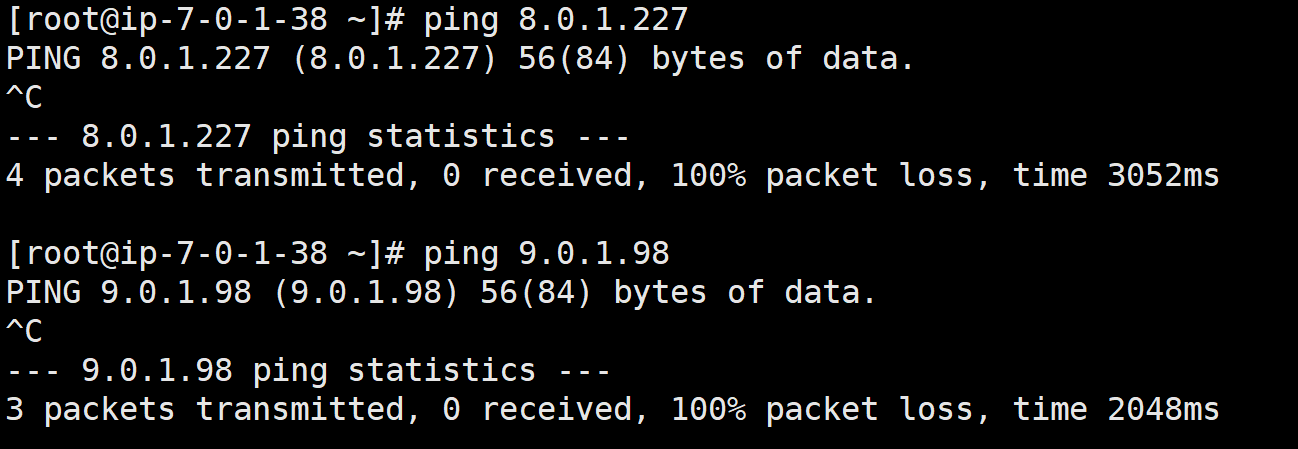
Graphical user interface, text, application

Description automatically generated

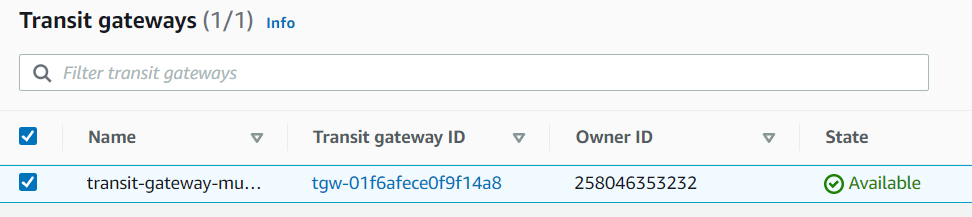
1. Created an instance inside each VPC, enabling public IP access to instance inside VPC-1 only.



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



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

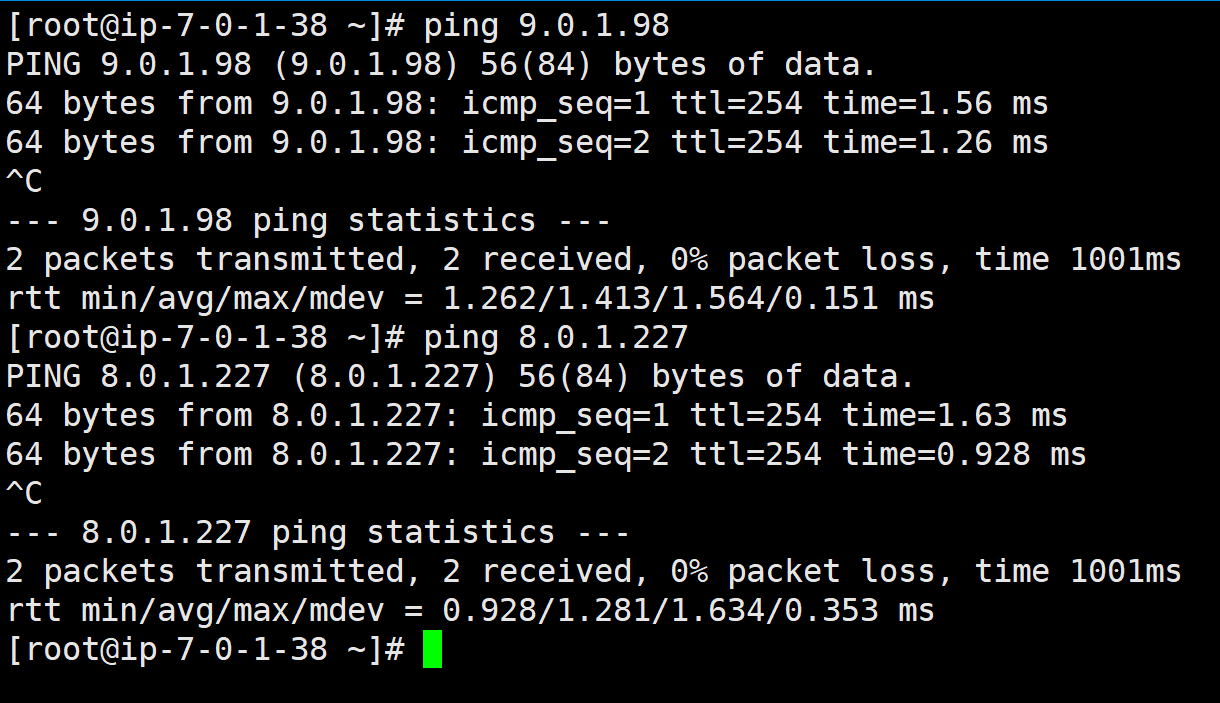


* 1. Created transit gateway attachments for each VPC.

Graphical user interface, text, application

Description automatically generated

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

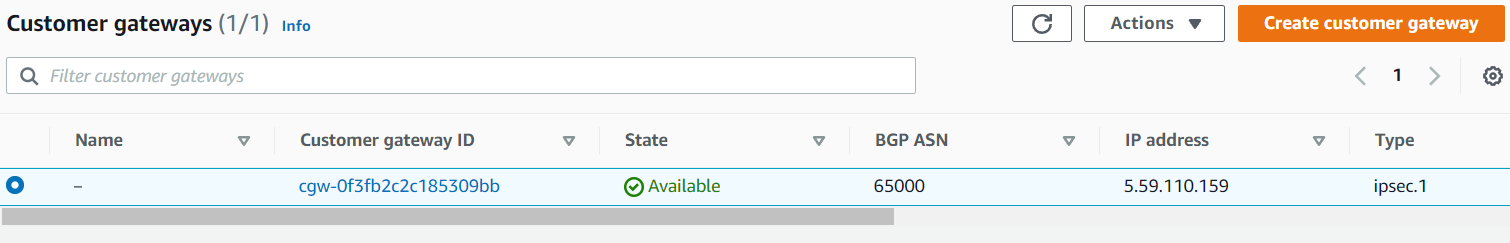


1. Created a new VPN attachment for site-to-site connectivity purpose.

Graphical user interface, text, application

Description automatically generated

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



Graphical user interface, text, application

Description automatically generated

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

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