AWS – GCP
VPN CONNECTION

AUGUST 17

Devamadhav Kattimuttathu

Creating a hybrid network and routes between two clouds

Introduction

In this lab, we will set up AWS and Google Cloud environments to configure a hybrid cloud connection between them. We will create and isolate one Virtual Private Cloud (VPC) in AWS and one in Google Cloud Platform (GCP). After configuring the VPCs, we will establish routing to manage traffic between them. Additionally, we will set up a site-to-site VPN to ensure secure communication between both cloud environments.

Observations and Screenshots

We began the lab by creating a VPC in AWS. First, we navigated to the VPC section in the AWS Management Console and selected the option to create a new VPC. We named it dk5197-VPC1 and set the IP address range to 10.30.0.0/20. The VPC was successfully created.

Next, we created two subnets within this VPC. We named them subnet01 and subnet02, assigning them IP ranges 10.30.0.0/24 and 10.30.0.128/24, respectively. We then created a new Route Table named dk5197-RT to manage network traffic, associating it with dk5197-VPC1. In the subnet association of the Route Table, we explicitly associated the two subnets. This configuration isolates the networks and allows for more efficient routing.

Following this, we created an Internet Gateway named dk5197-IG and successfully attached it to our VPC. Finally, we updated the Route Table to include a new route with a destination of 0.0.0.0/0, which allows all outbound traffic to pass through the Internet Gateway.

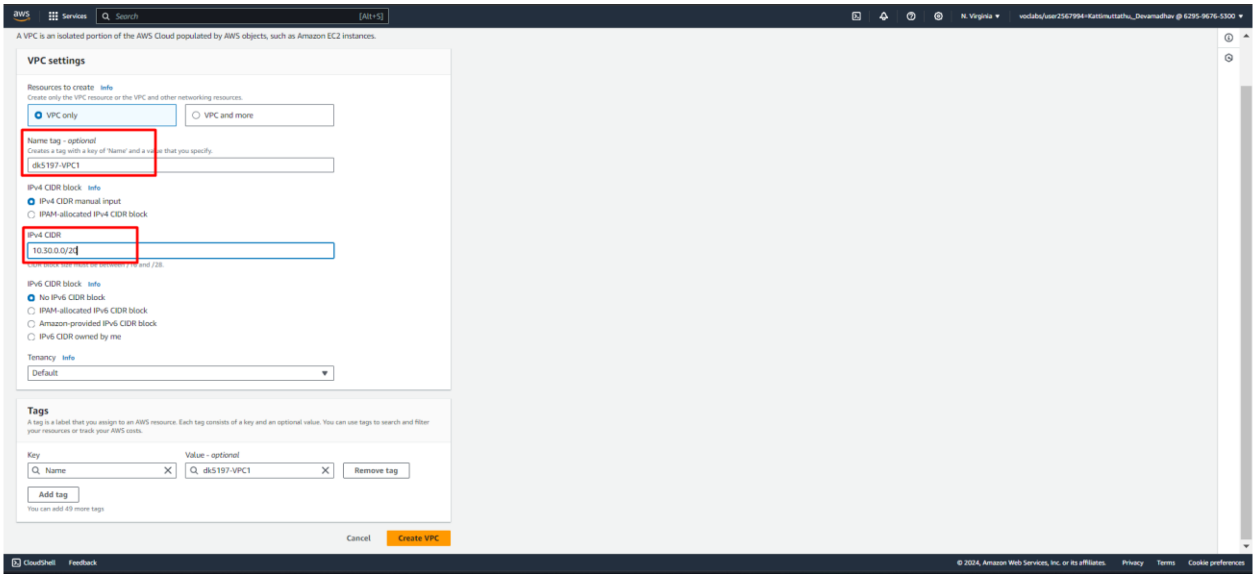

Figure 1 - Creating a new VPC in AWS

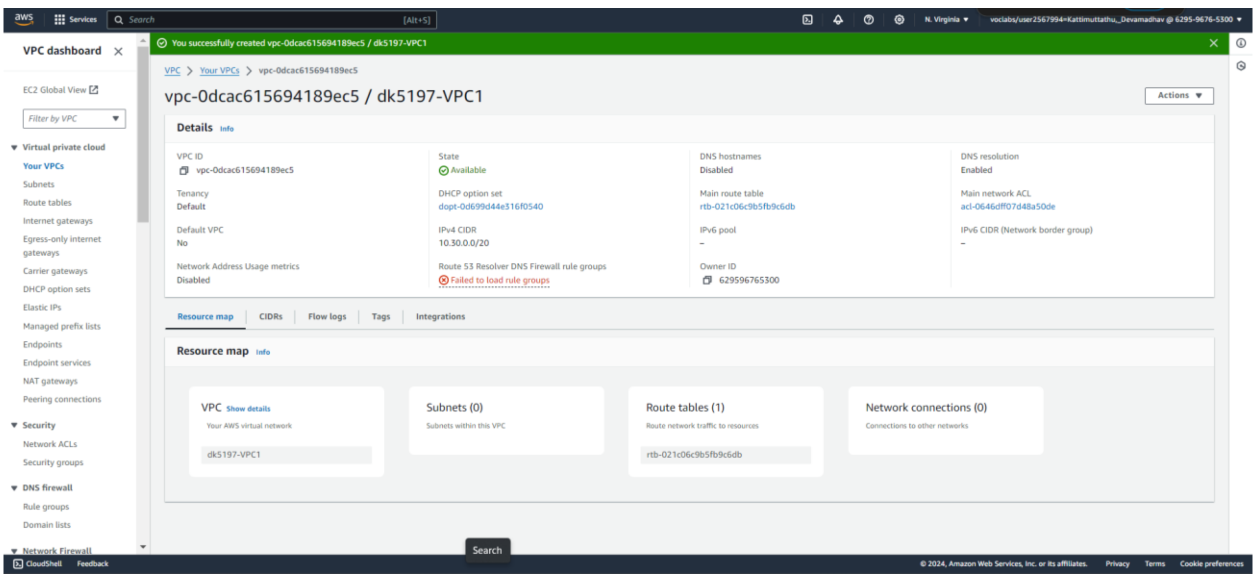

Figure 2 - Successfully created a new VPC named dk5197-VPC1

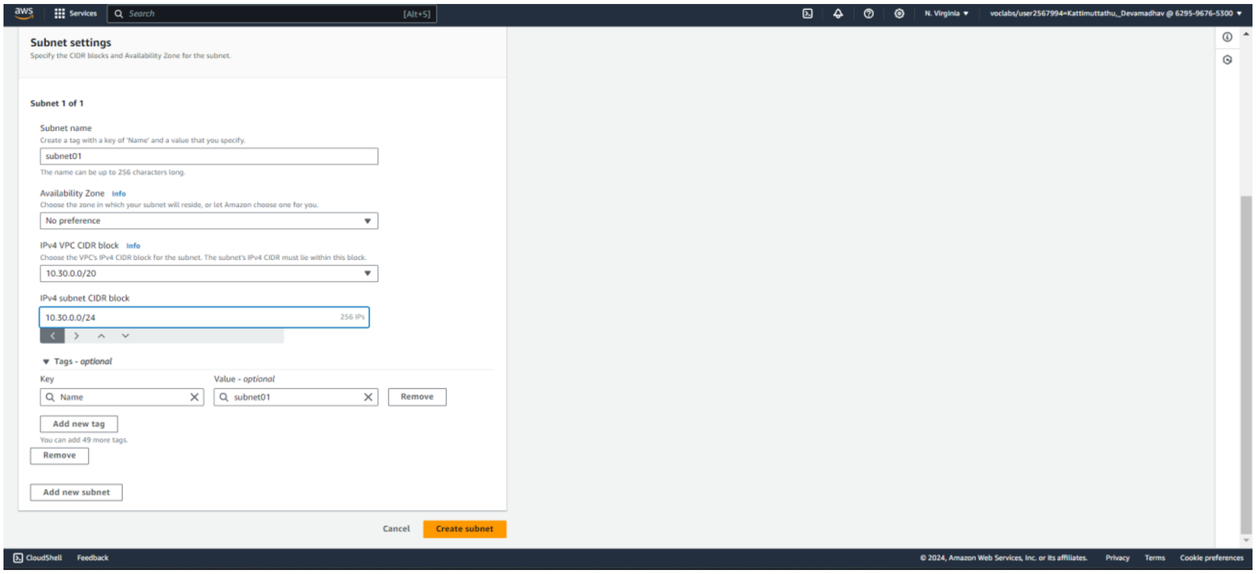

Figure 3 - Creating the first subnet for the VPC

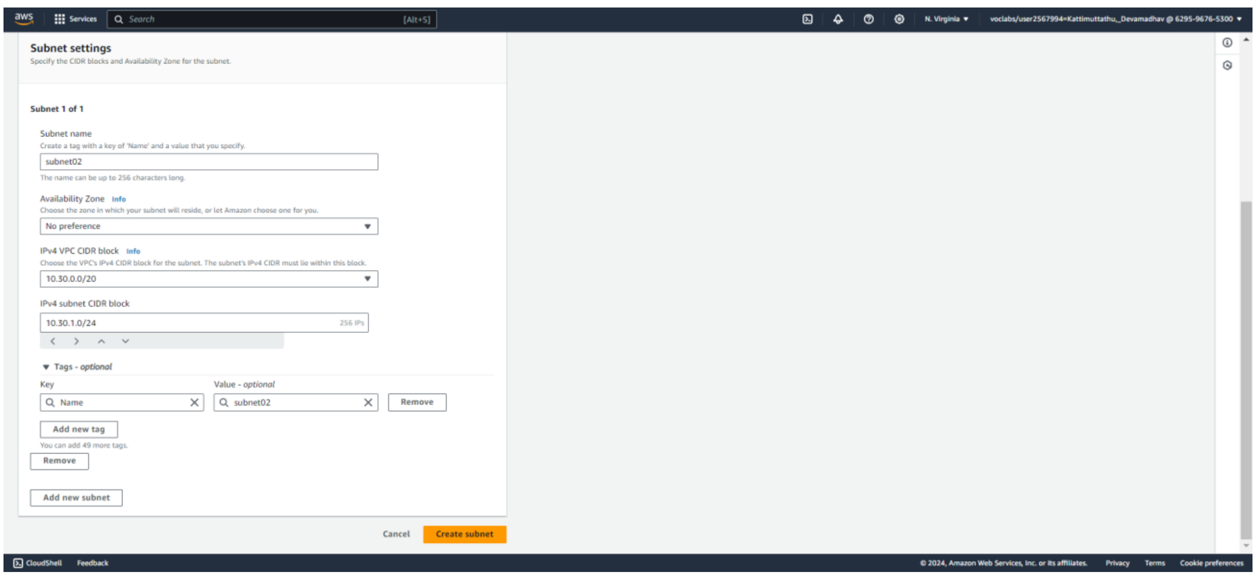

Figure 4 - Creating the second subnet for the VPC

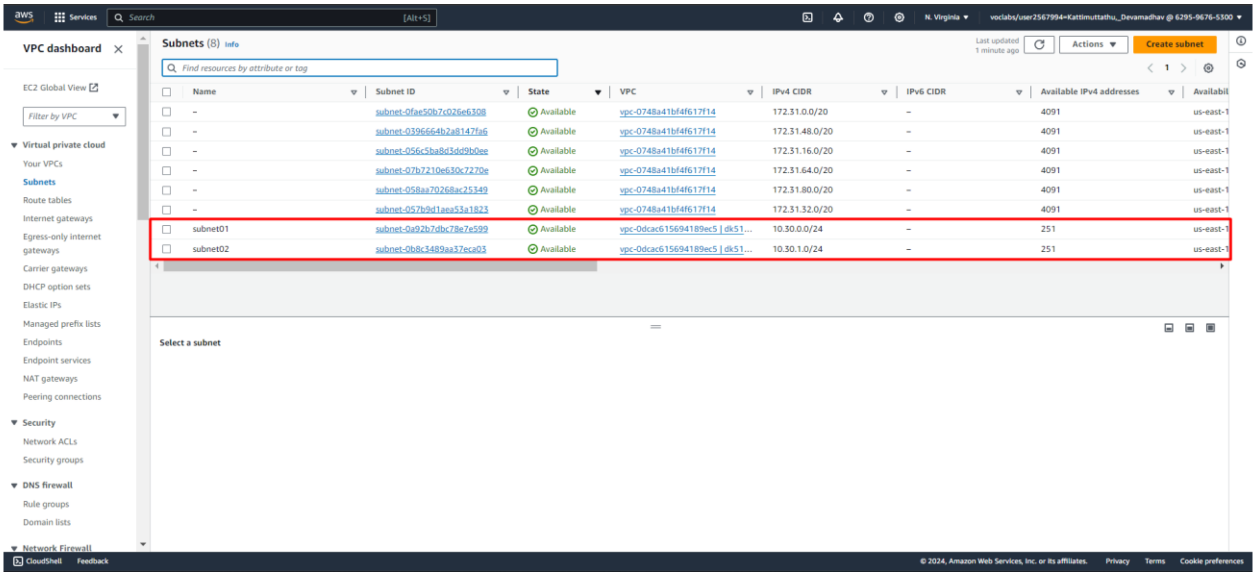

Figure 5 - Successfully created two subnets for the VPC in AWS

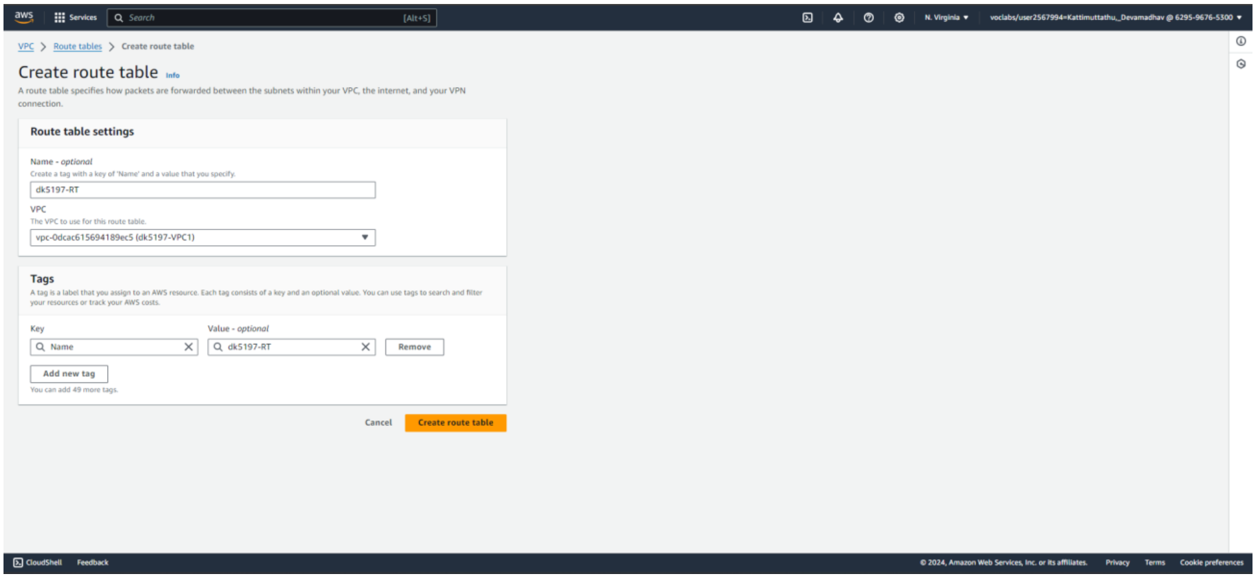

Figure 6 - Creating a route table in AWS

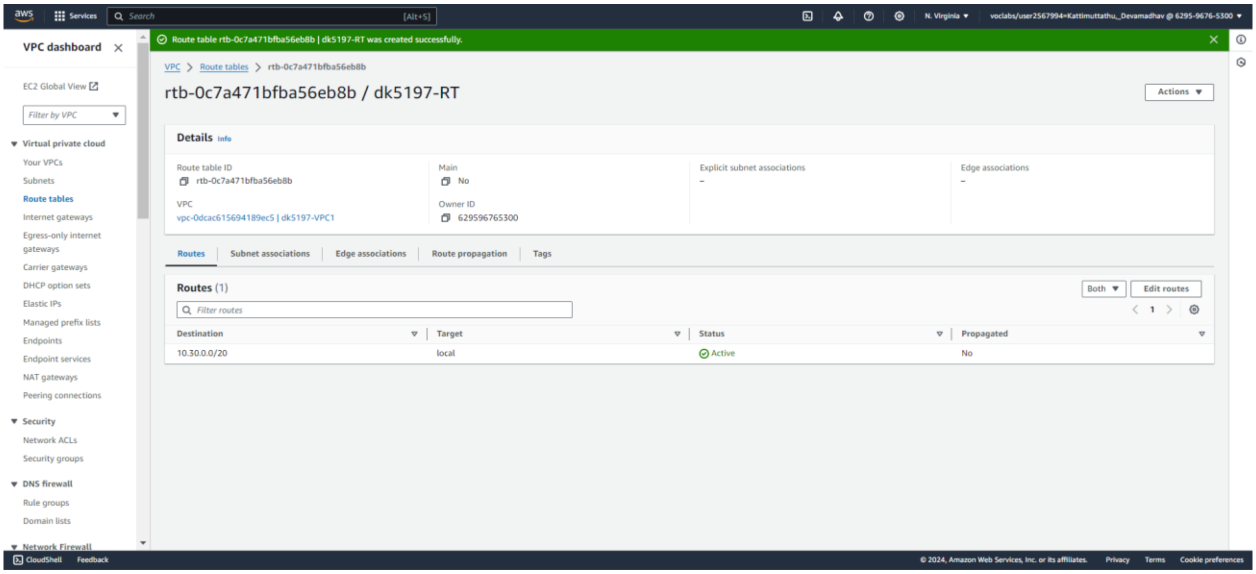

Figure 7 - Successfully created new route table for VPC

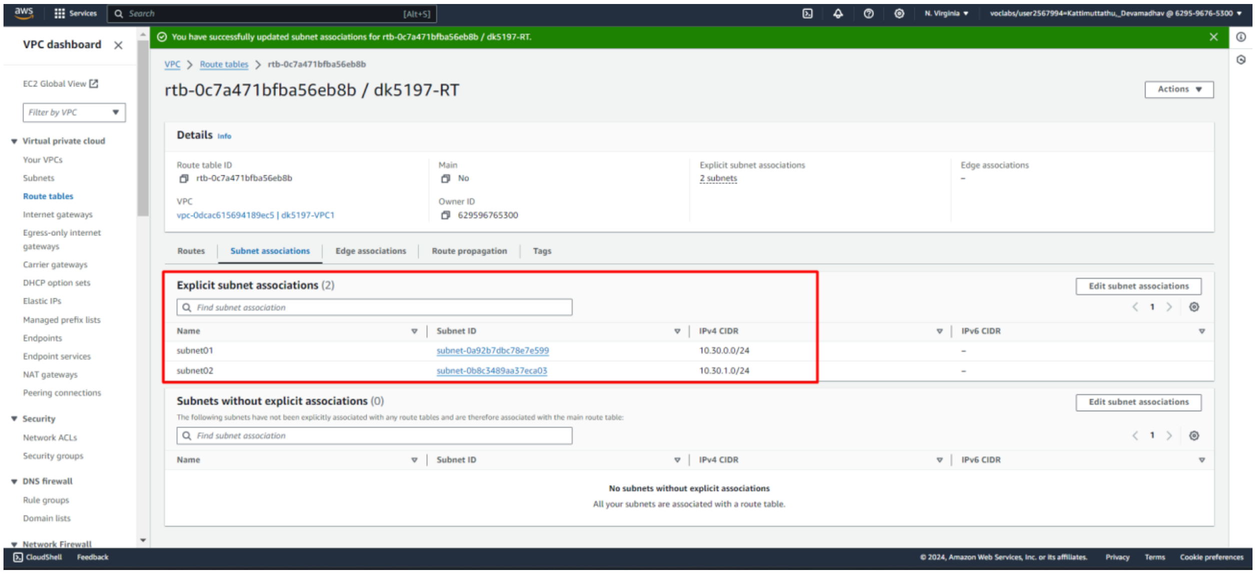

Figure 8 - Added our two subnets as explicit subnet associations

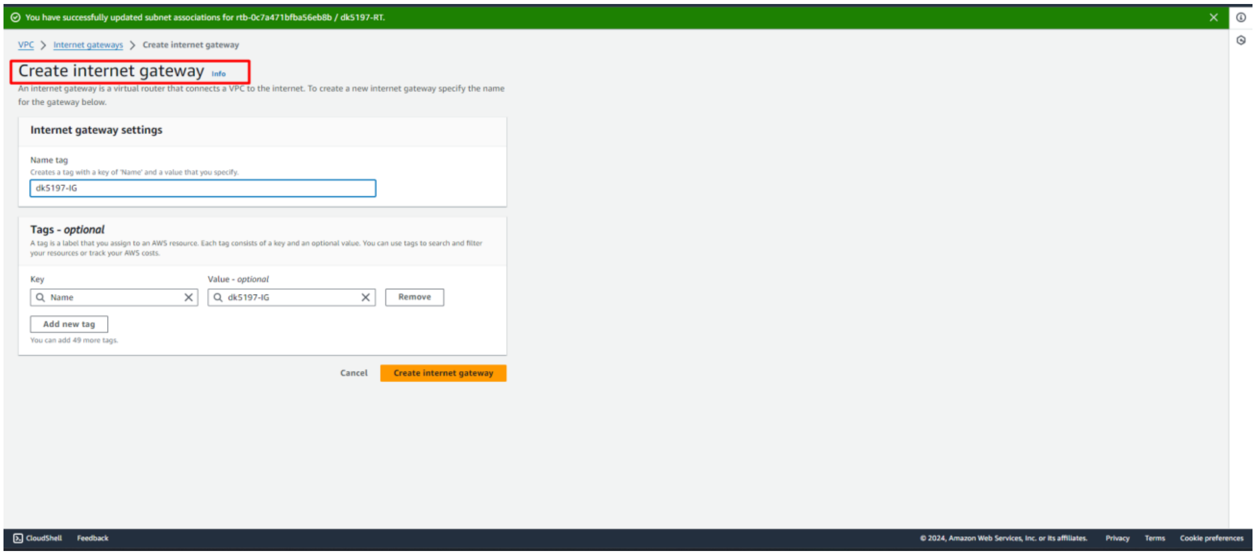

Figure 9 - Creating a new internet gateway for AWS VPC

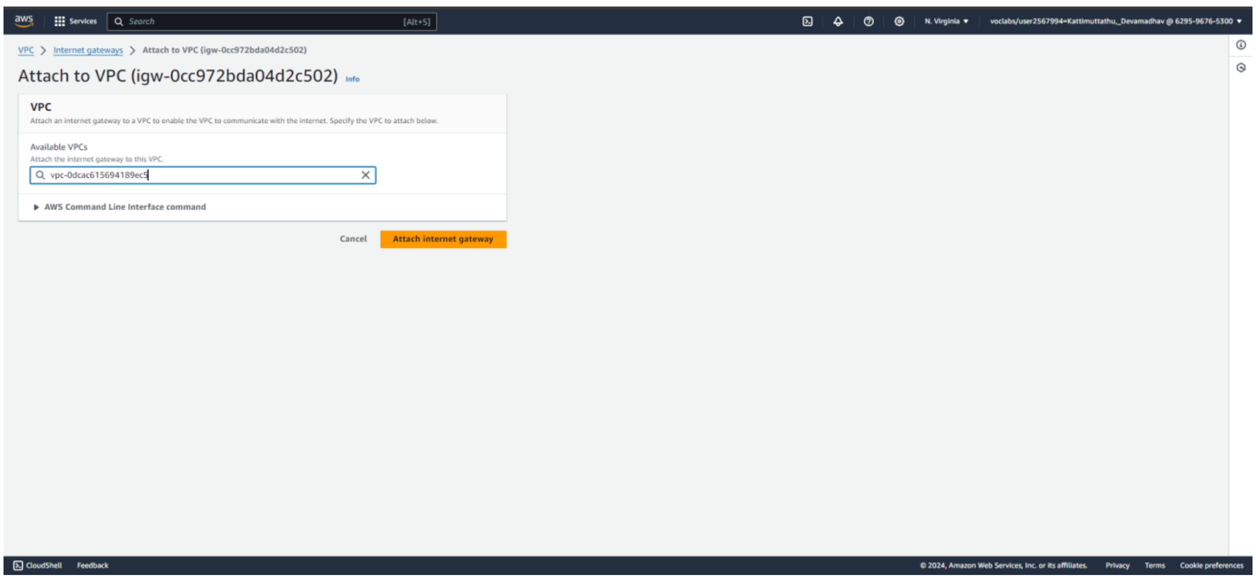

Figure 10 - Attaching the internet gateway to the VPC

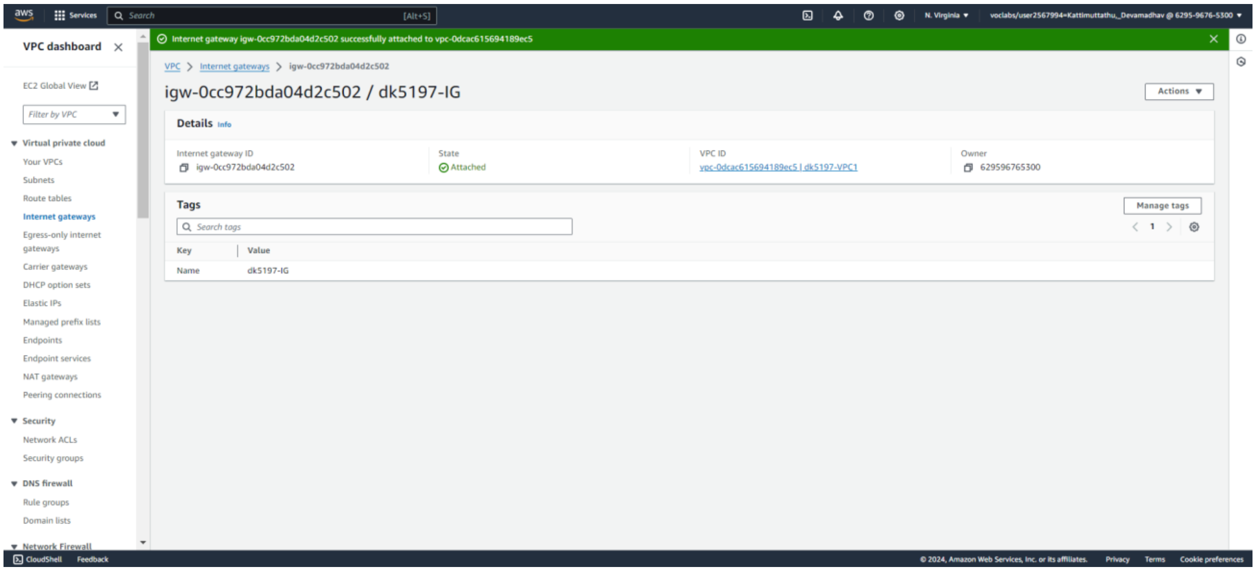

Figure 11 - Successfully attached the internet gateway to our vpc

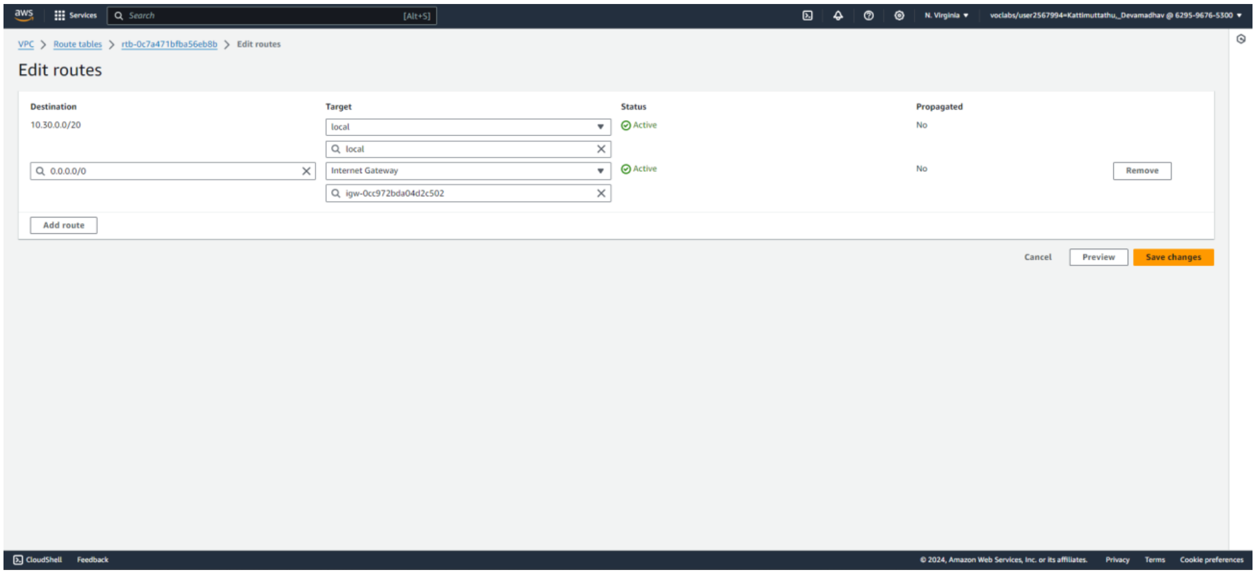

Figure 12 - Editing the route table to enable all traffic to internet through internet gateway

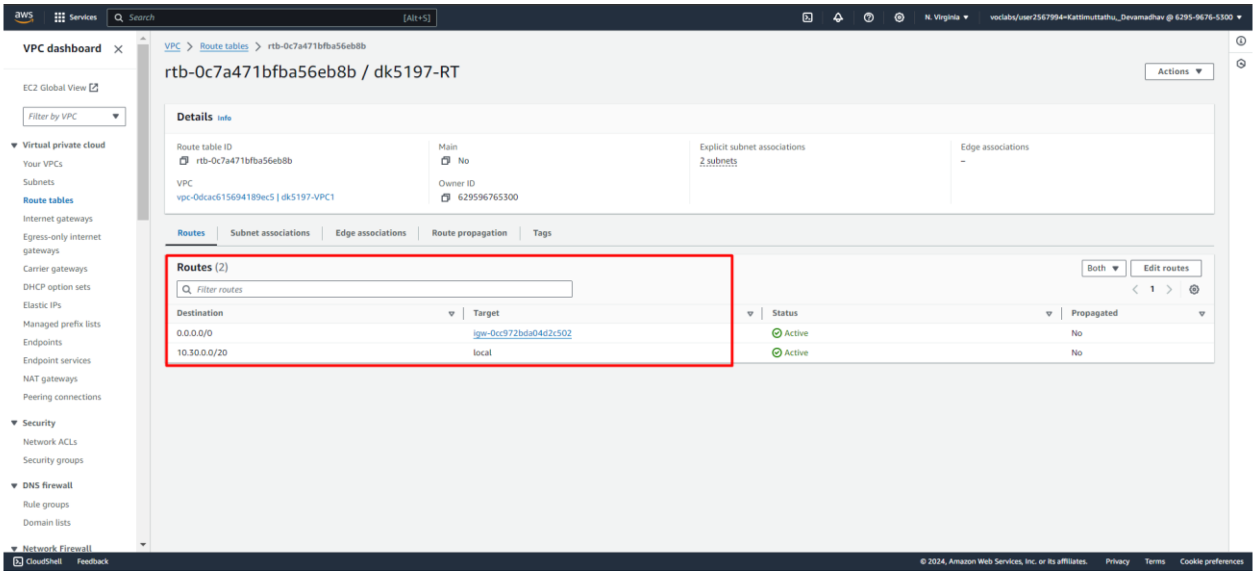

Figure 13 - Successfully created a new route

Next, we created a new web server named dk5197-server with Linux as the operating system. A new key pair was generated for the system. We configured the network by selecting the VPC and assigning subnet01 to the instance. The instance was successfully launched, and we connected to it using SSH.

Upon logging into the system, we installed the Apache web server. After the installation, we created a new webpage and copied it to the appropriate directory. To ensure the webpage was accessible over the internet, we added an additional inbound rule allowing HTTP access from any IP address (0.0.0.0/0). This configuration enabled us to successfully display our webpage using the instance's public IP address.

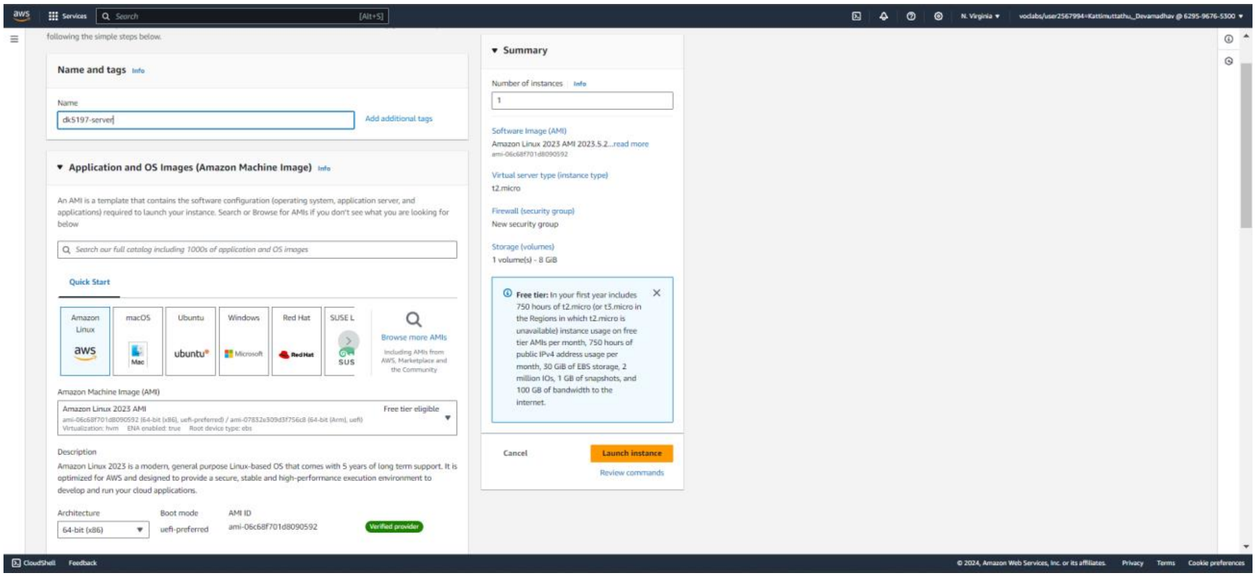

Figure 14 - Creating a new instance in AWS for our webserver

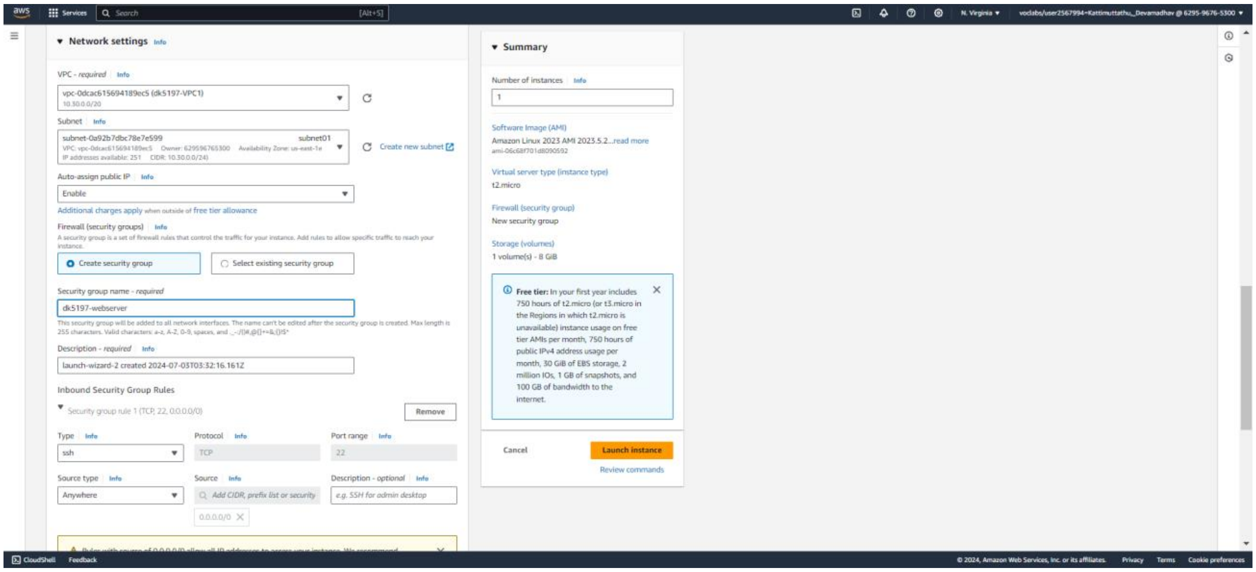

Figure 15 - Configuring the network settings in our VPC

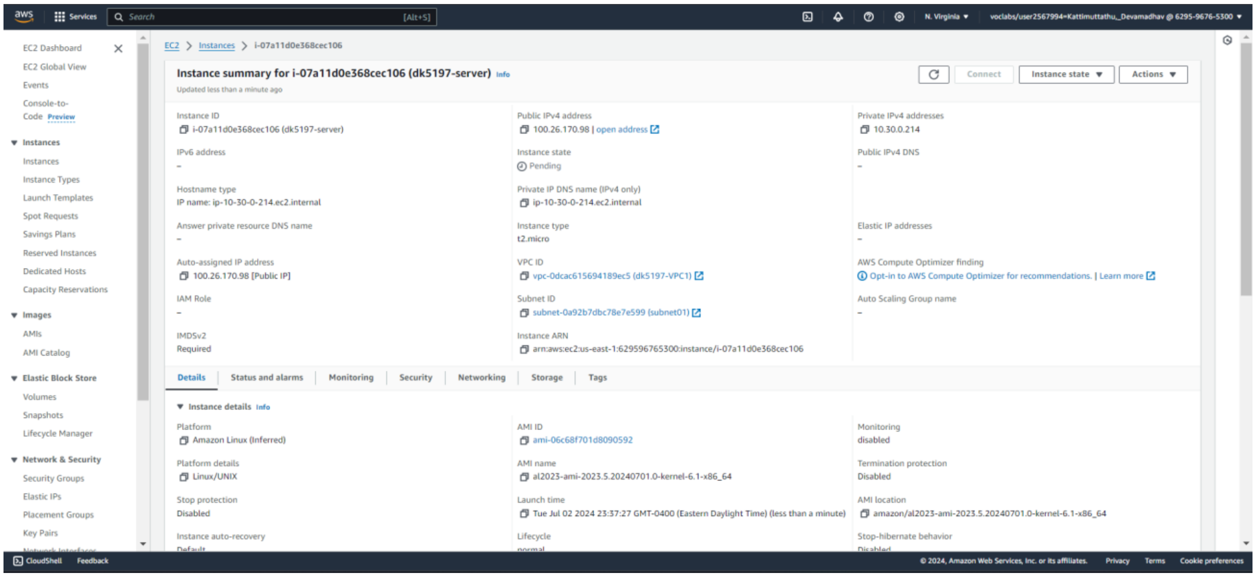

Figure 16 - Displaying the details of our instance

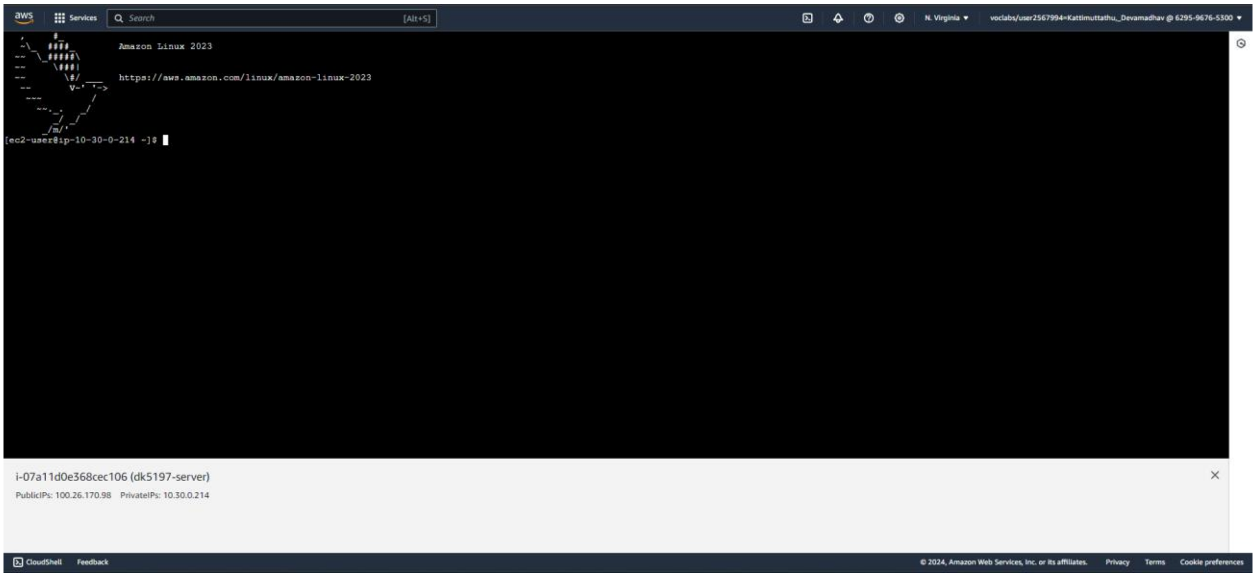

Figure 17 - Successfull SSH connection to our instance

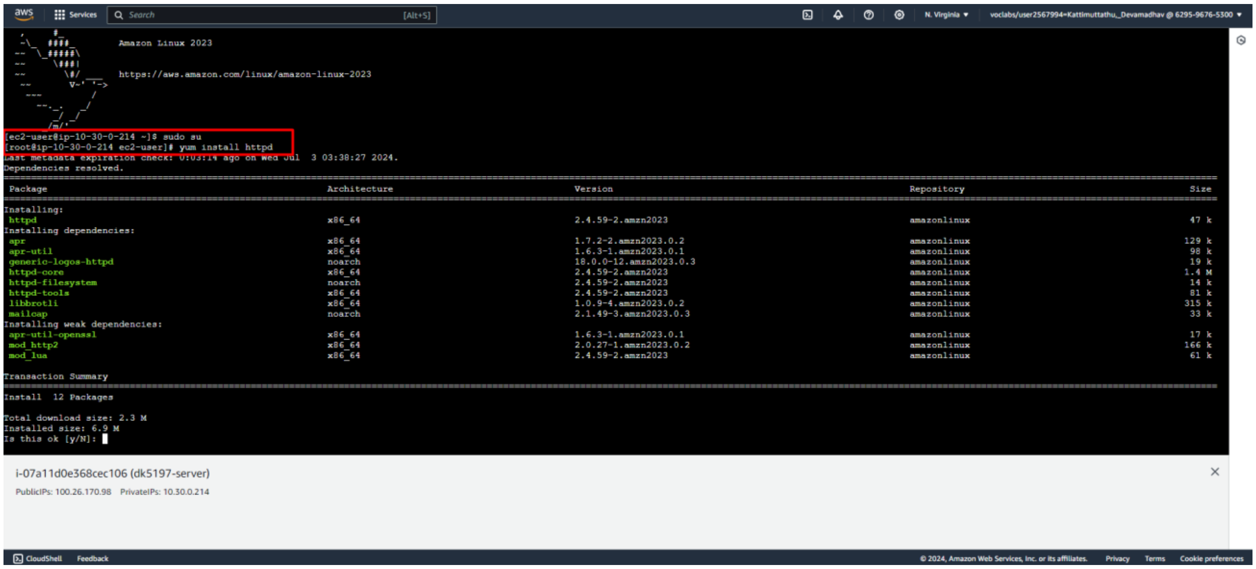

Figure 18 - Installing apache server in our instance

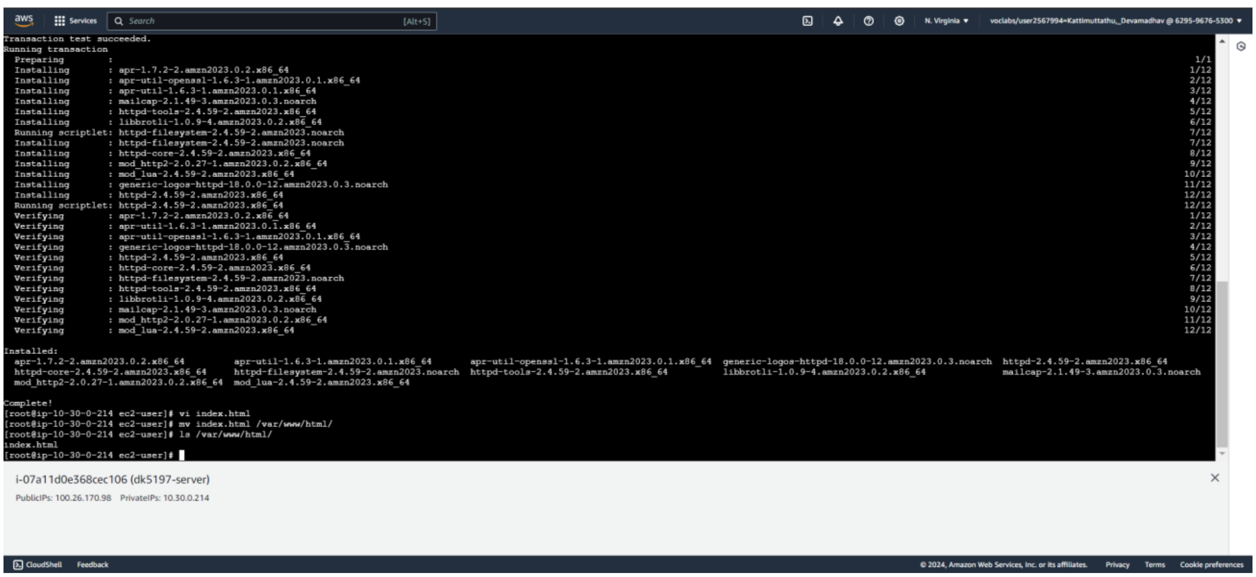

Figure 19 - Created a new webpage and copied to the corresponding folder

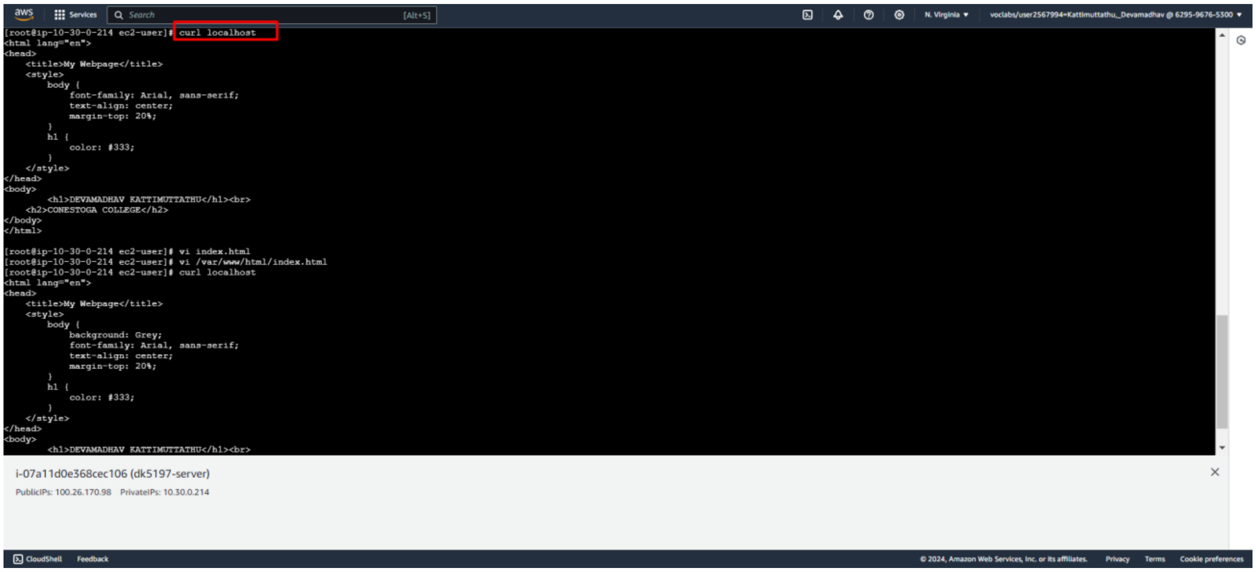

Figure 20 - Curl command to localhost

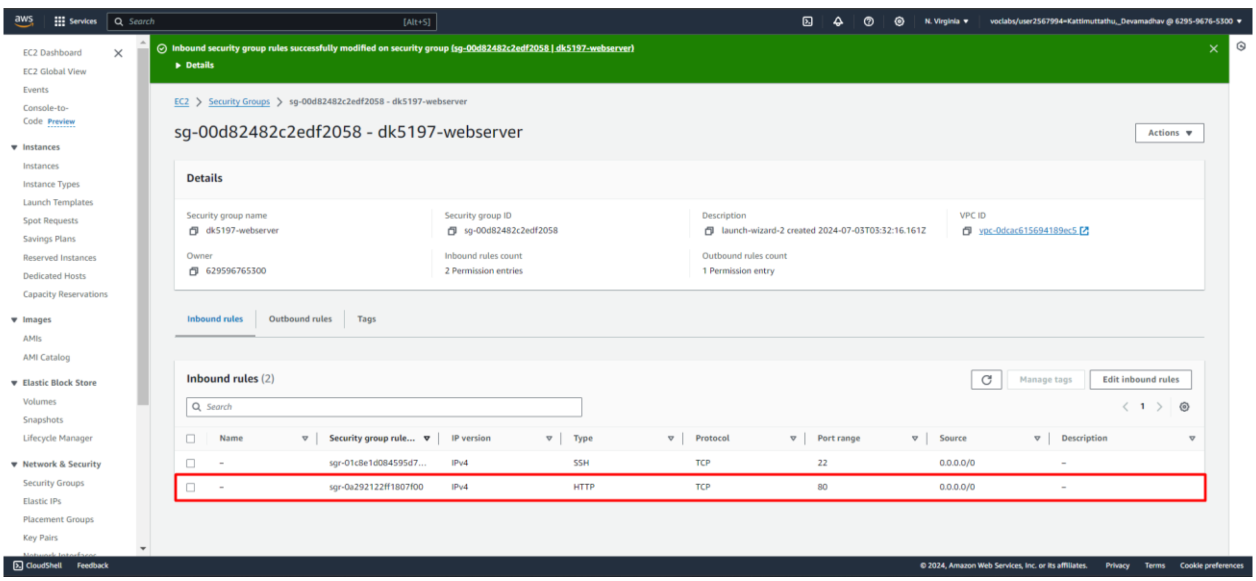

Figure 21 - Enabling HTTP traffic for our instance

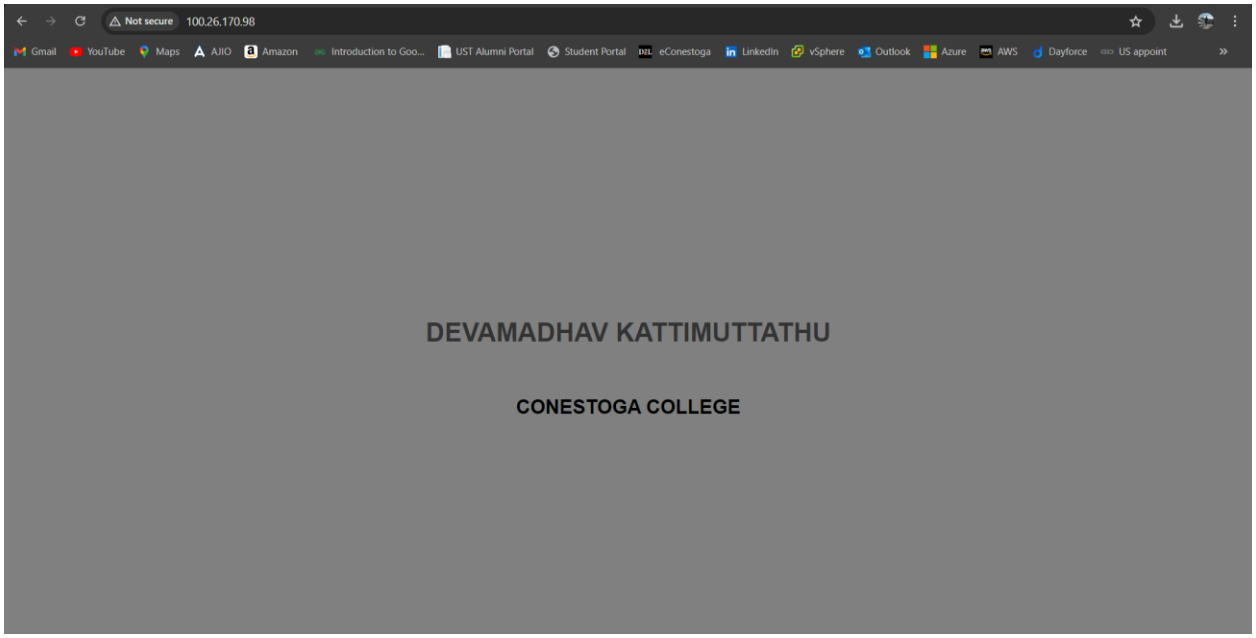

Figure 22 - Successfully access our webpage

Next, we navigated to the Google Cloud Platform (GCP) console to create a new Virtual Private Cloud (VPC). We named it dk5197-vpc2 and added two subnets. The first subnet, named sub1, was configured in the us-east1 region (matching the AWS region) with an IP range of 10.173.46.0/25. The second subnet, named sub2, was configured with an IP range of 10.173.46.128/25. After configuring the subnets, we set up firewall rules to allow custom ingress traffic for ICMP, RDP, and SSH. We also enabled global dynamic routing mode for the VPC before finalizing its creation. The VPC was successfully created.

Next, we navigated to Compute Engine and created a new instance named dk5197-gcp-server in the us-east1 region. We used the newly created VPC and sub1 for the instance, leaving the rest of the settings as default. After launching the instance, we attempted to SSH into it and ping the AWS instance using the command ping 10.30.0.214. However, there was no response because the security group in AWS had not yet been configured to allow ICMP traffic from the GCP network.

To address this, we navigated to the inbound rules in the AWS security group and added an ICMP rule for the GCP network. Although the rule was successfully added, we still couldn't get a response because AWS was not yet configured to connect to the GCP network. To complete the setup, we must establish a VPN connection for secure communication between the two cloud environments.

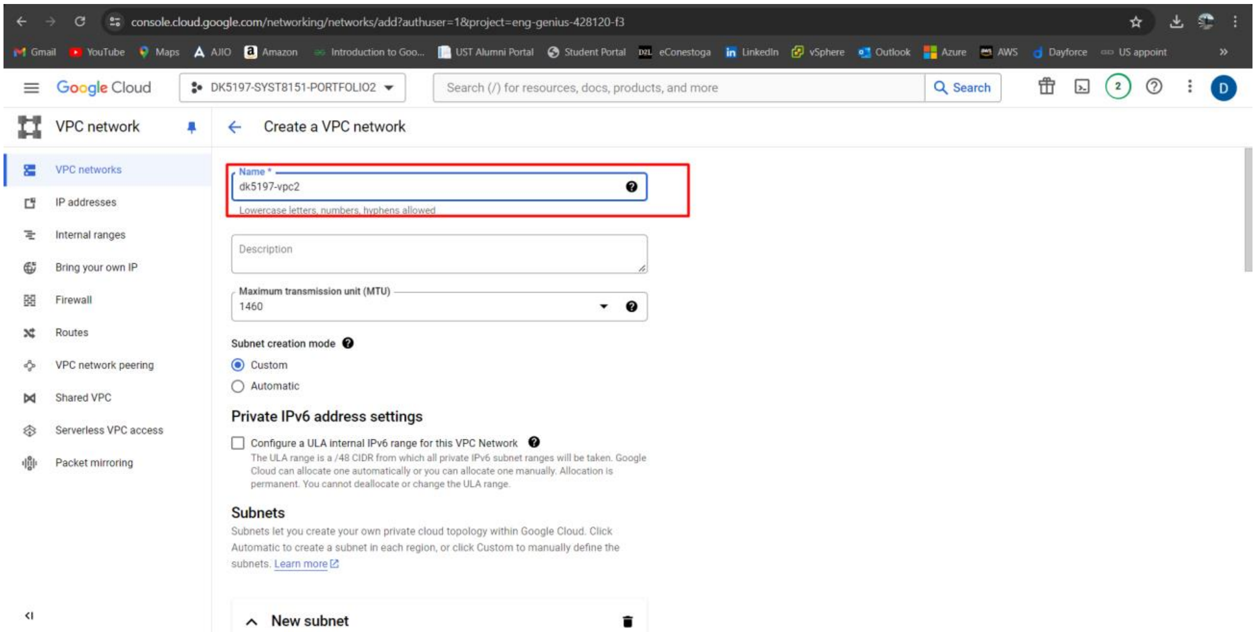

Figure 23 - Creating a new instance in gcp

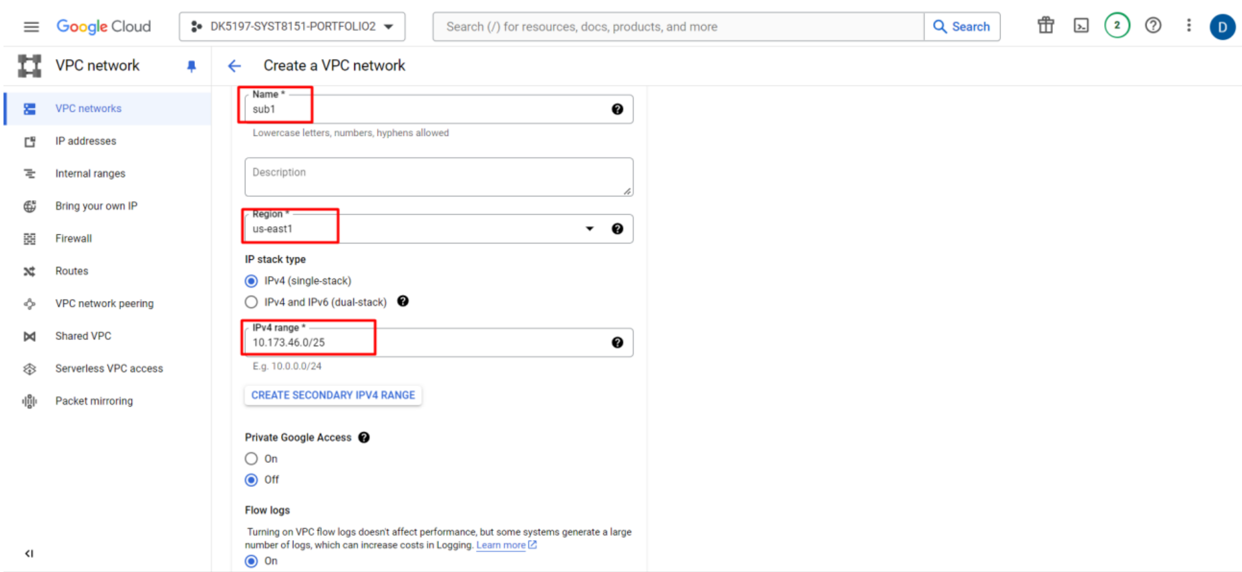

Figure 24 - Creating first subnet in our vpc

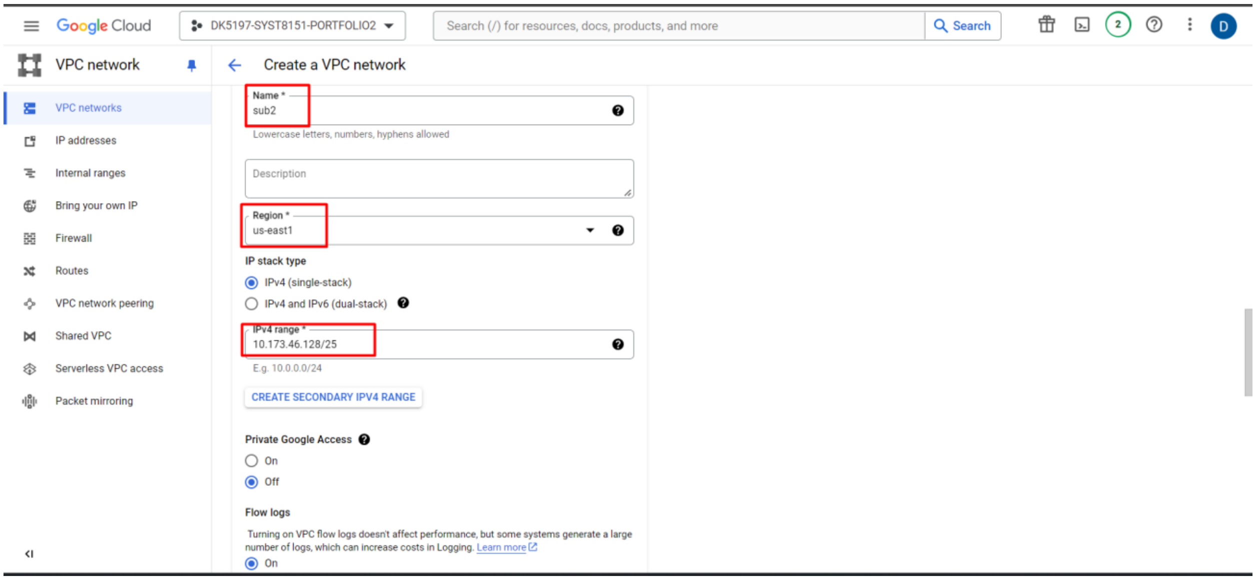

Figure 25 - Adding second subnet for our VPC

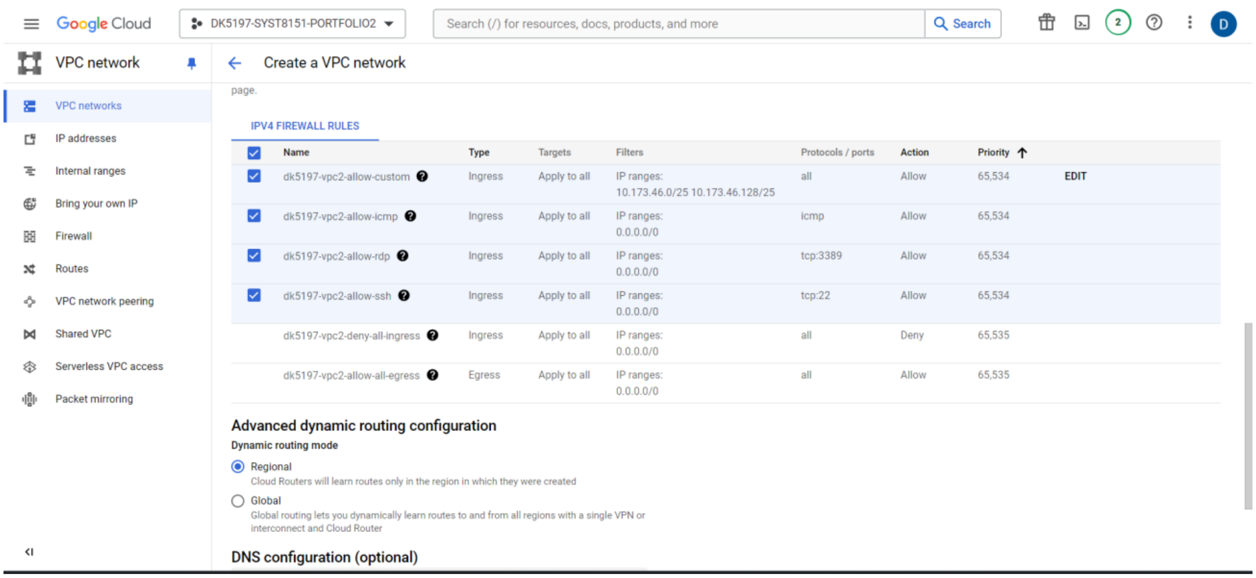

Figure 26 - Configuring the firewall for our VPC

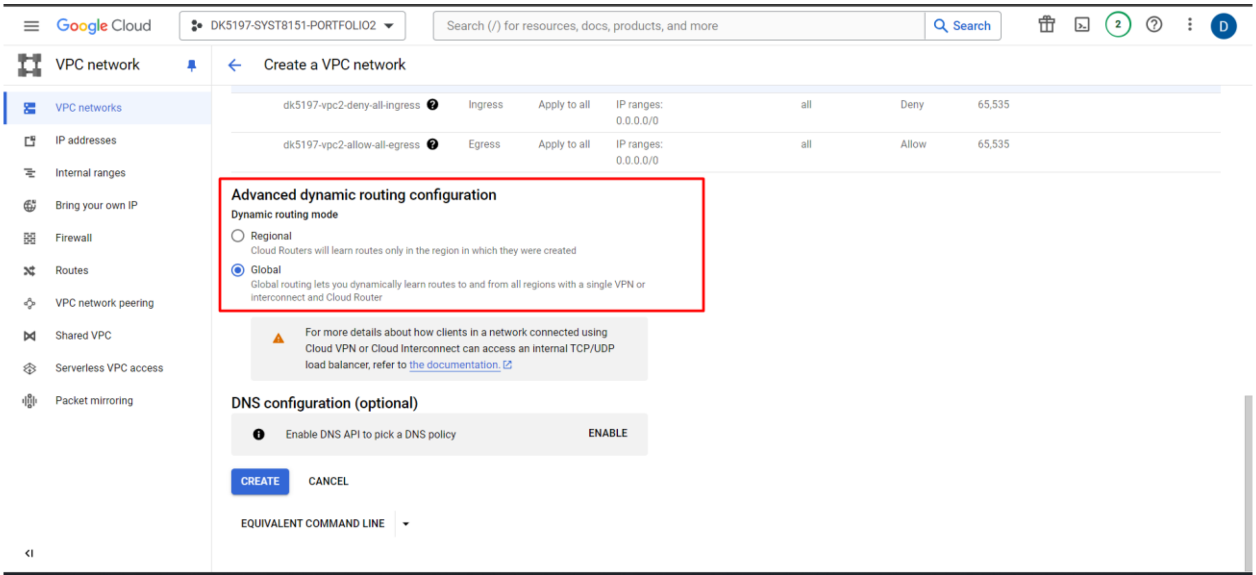

Figure 27 - Enabling global dynamic routing

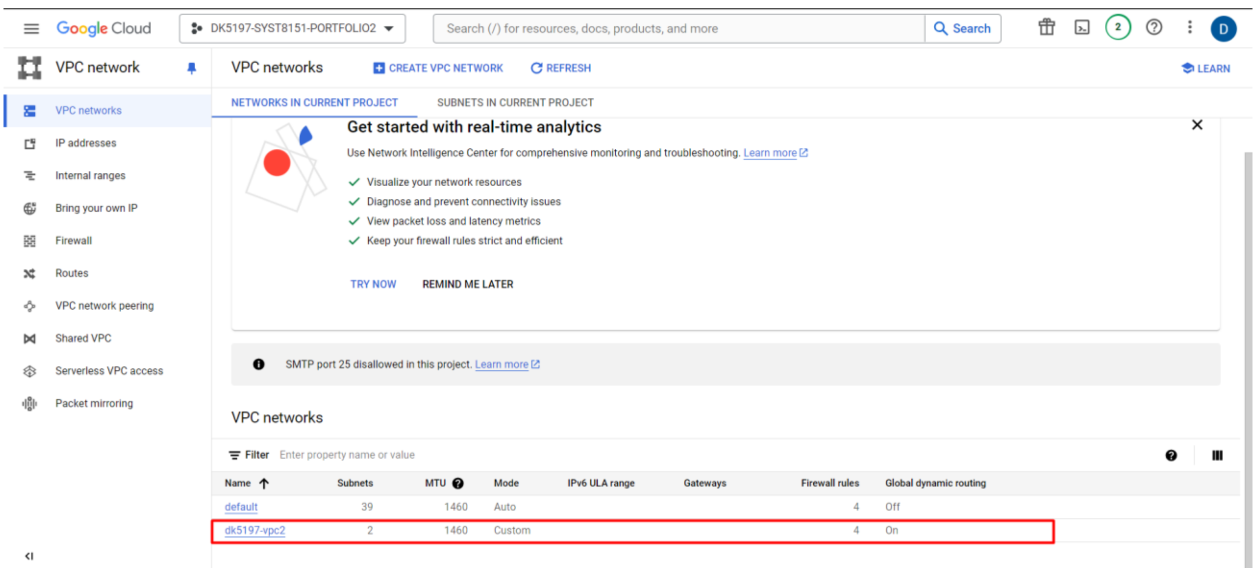

Figure 28 - Successfully created a new VPC in gcp

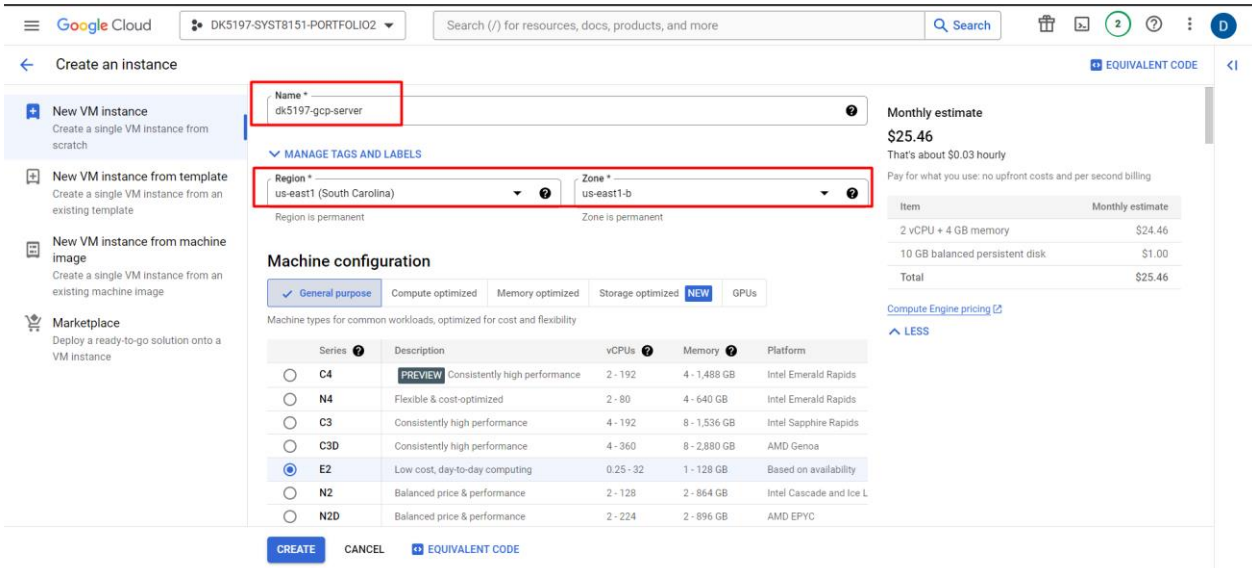

Figure 29 - Creating a new instance in GCP

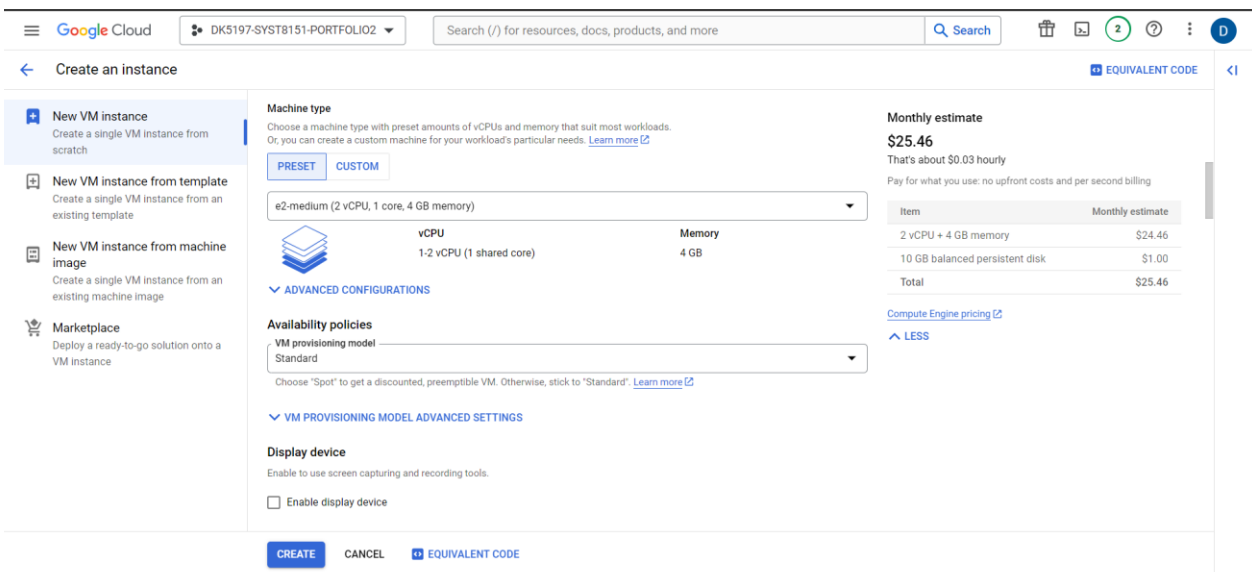

Figure 30 - Kept rest of the settings as defualt for the creaton of instance

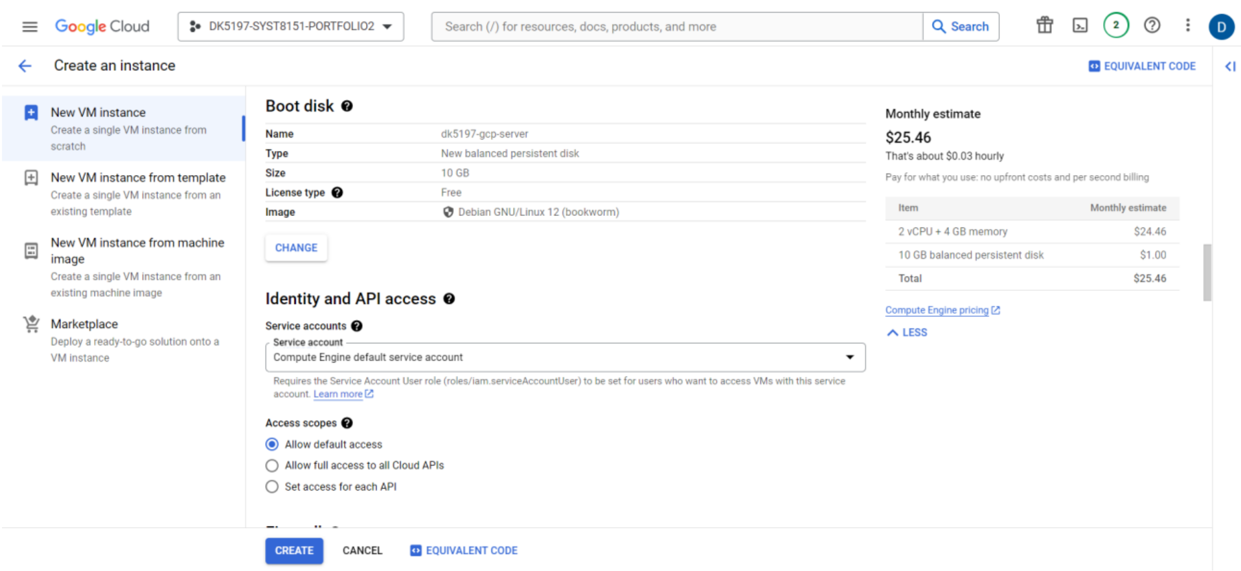

Figure 31 - leaving all other options as default for the creation of instance

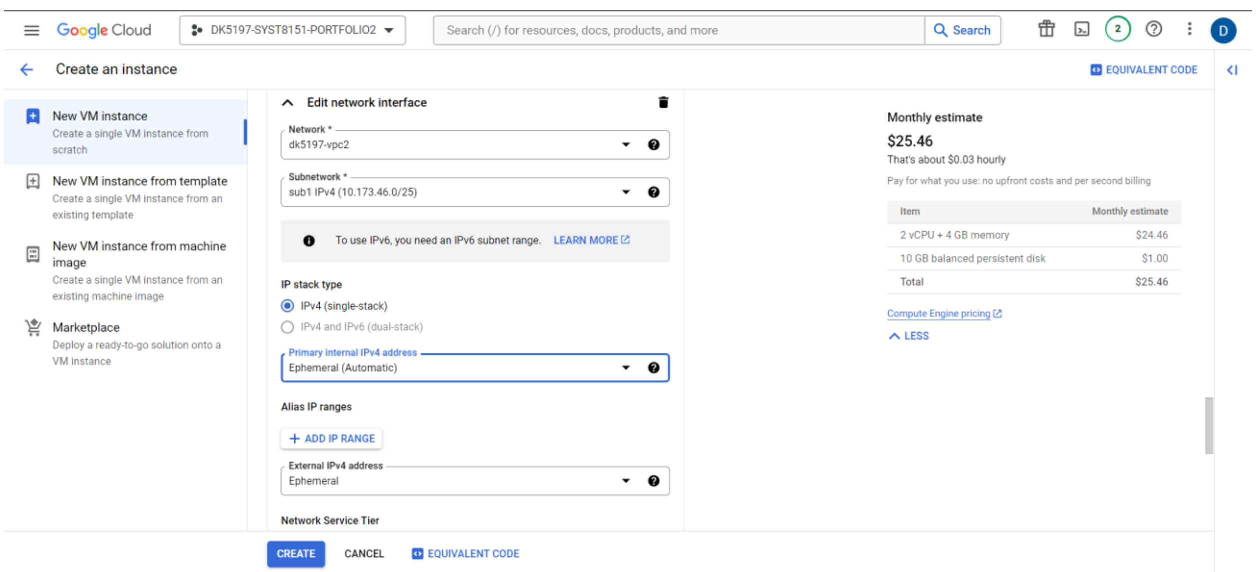

Figure 32 - Selecting our vpc and subnet1 to use as the insance network'

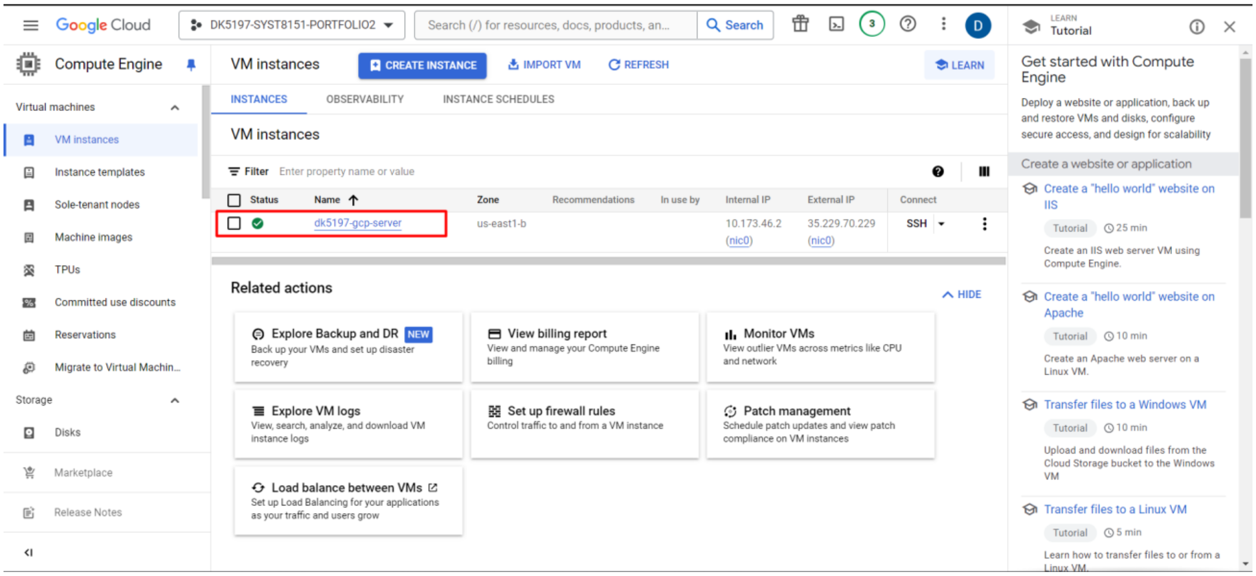

Figure 33 - Successfully created the new instance in gcp

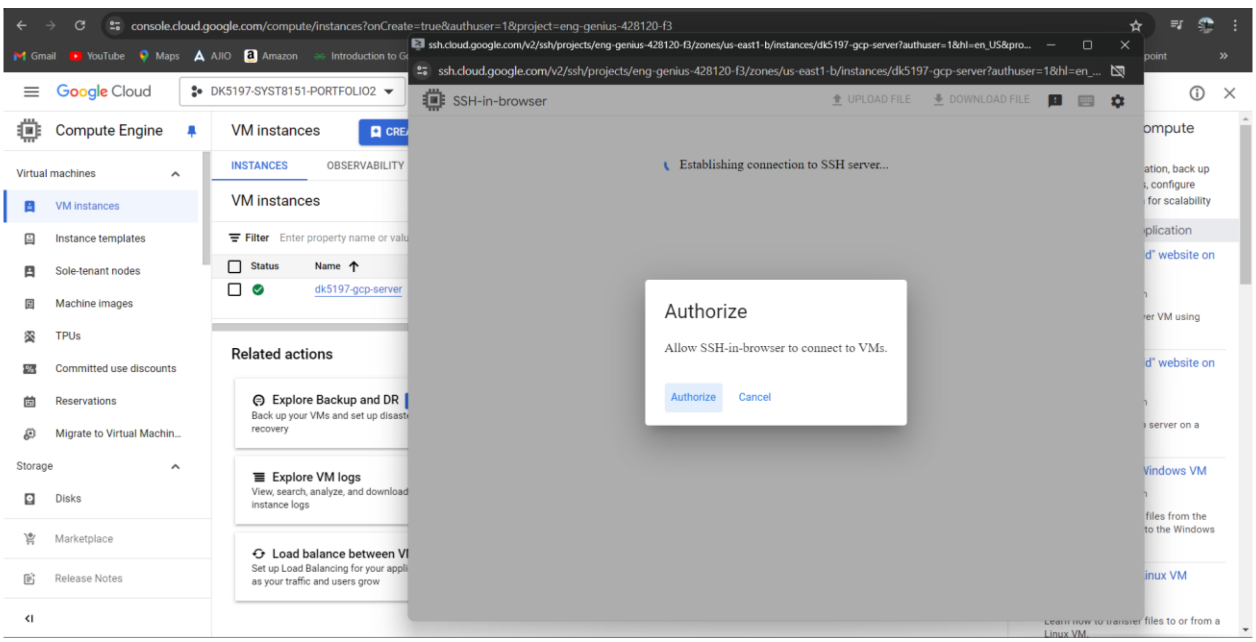

Figure 34 - SSH connection to GCP instance

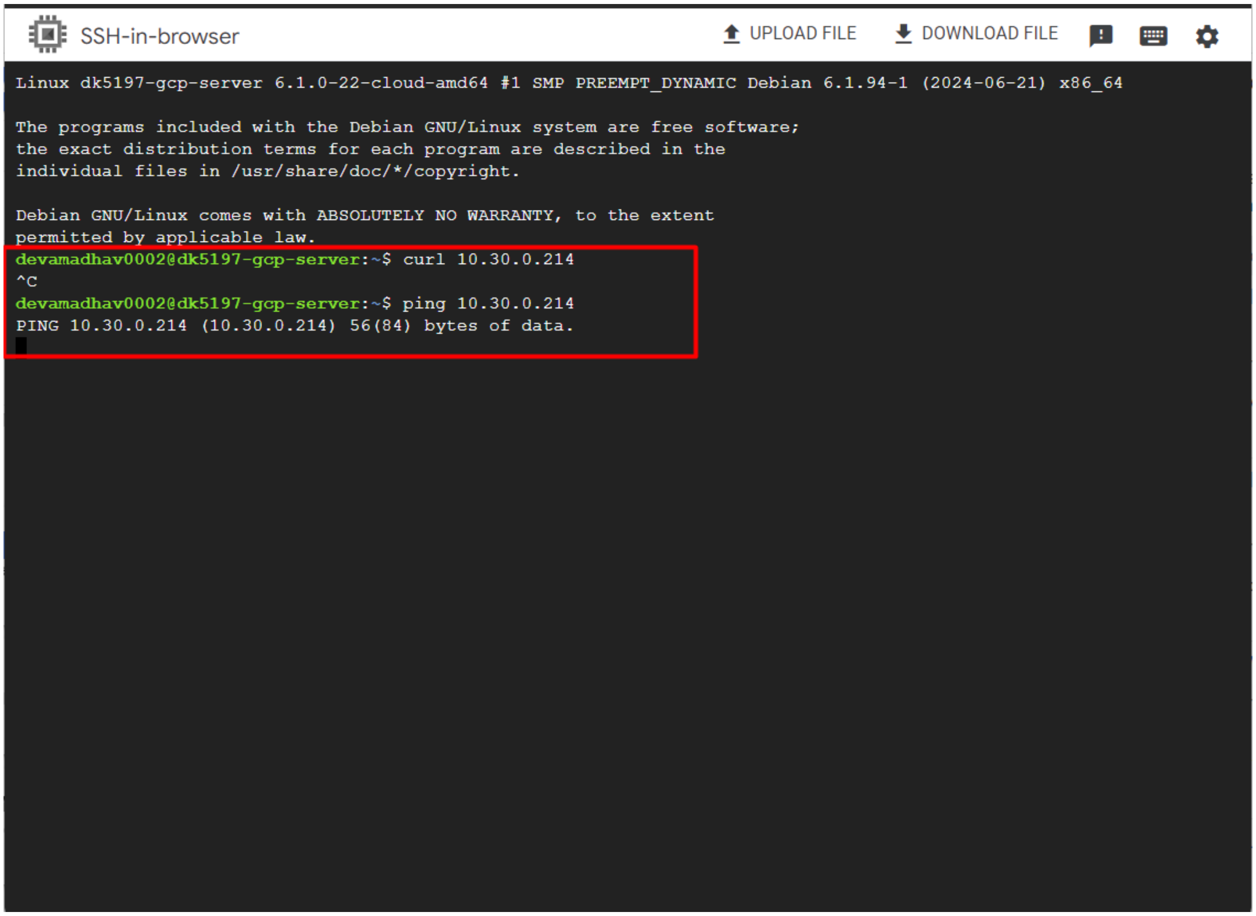

Figure 35 - Tried to connect to the aws security group, but haven't received any response

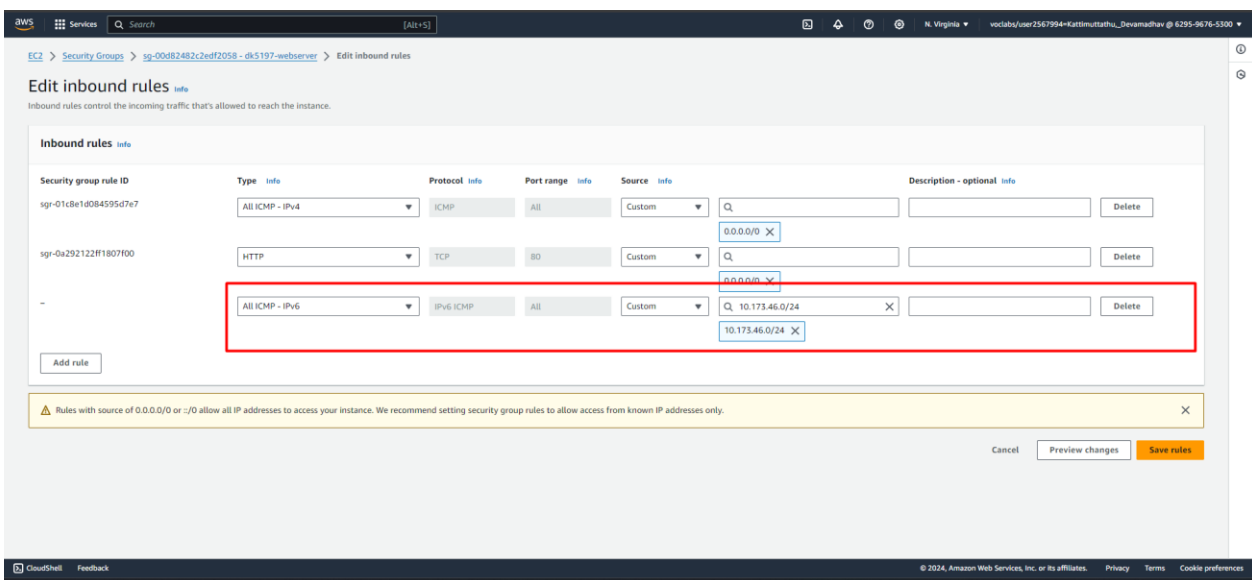

Figure 36 - Enabling the ICMP traffic in inbound rules for AWS

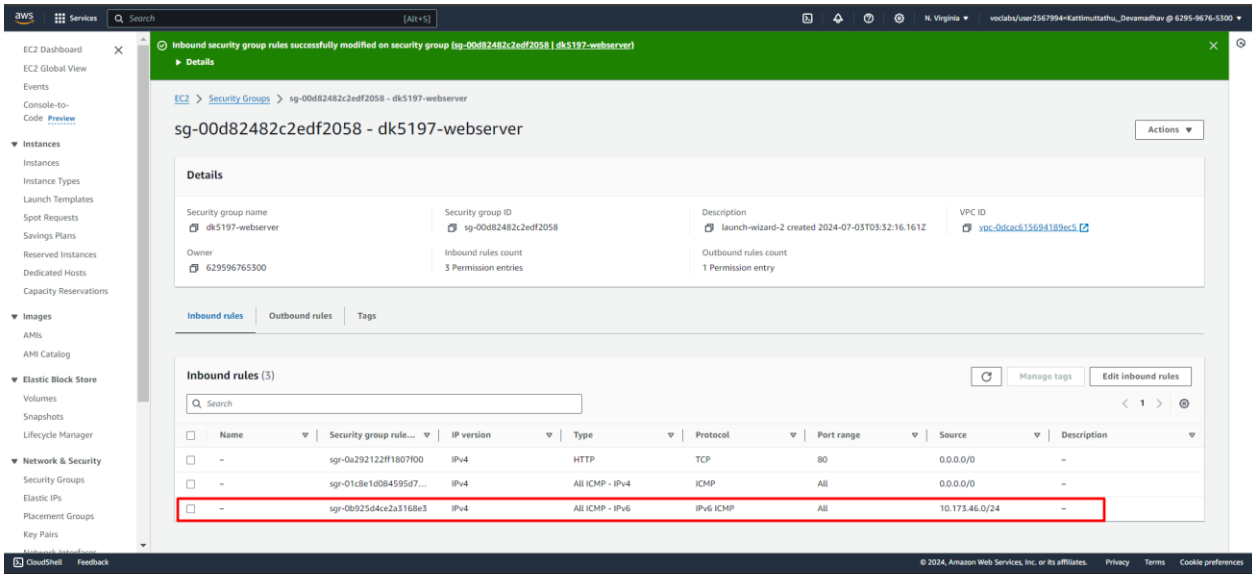

Figure 37 - Successfully added a new inbound rule in aws to allow ping

To establish the VPN connection, we first navigated to the GCP console and selected the Cloud Router option to create a new router for the VPN connection. We named the router gcp-router, selected our VPC network, and set the ASN (Autonomous System Number) value to 65000. The ASN is crucial for creating BGP (Border Gateway Protocol) sessions, which facilitate communication between different networks across providers. The router was successfully created.

Next, we navigated to the VPN section and chose the High Availability (HA) VPN option to create the tunnel. We created a new VPN gateway, associating it with our VPC and selecting the US-east1 region. Once the gateway was set up, our next task was to configure the VPN tunnel. GCP provides two public IP addresses for the VPN interfaces.

To complete the VPN setup, we also needed to define a non-Google Cloud peer VPN gateway, which meant configuring a similar VPN gateway on the AWS side. To do this, we navigated to the Virtual Private Gateway option in the AWS VPC console to begin setting up the corresponding gateway.

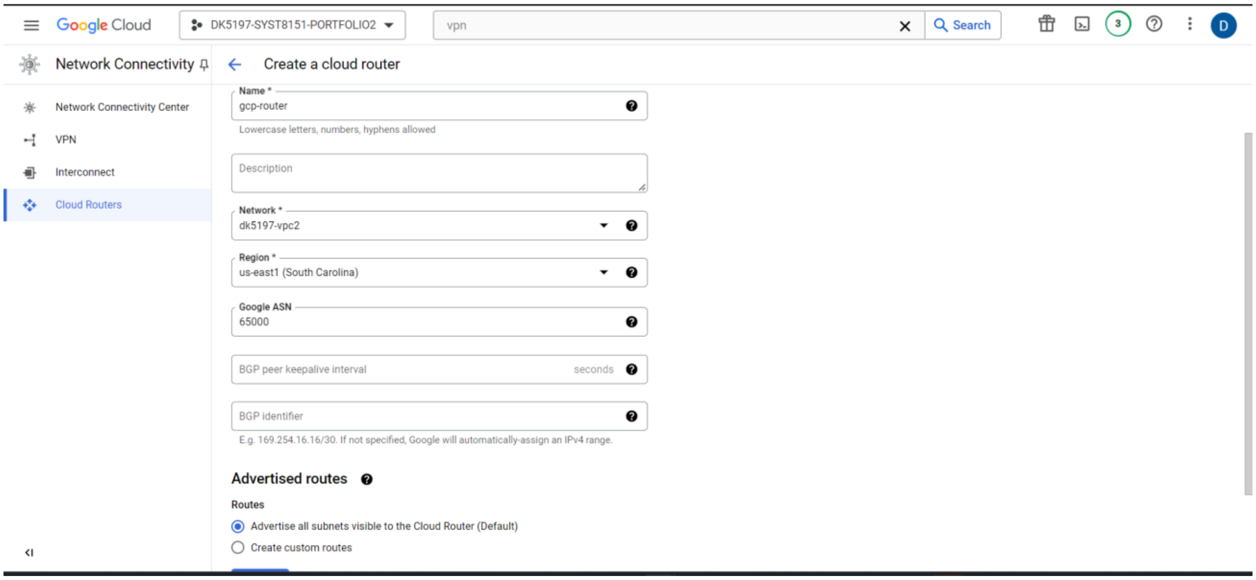

Figure 38 - Creating a router in gcp for VPN connection

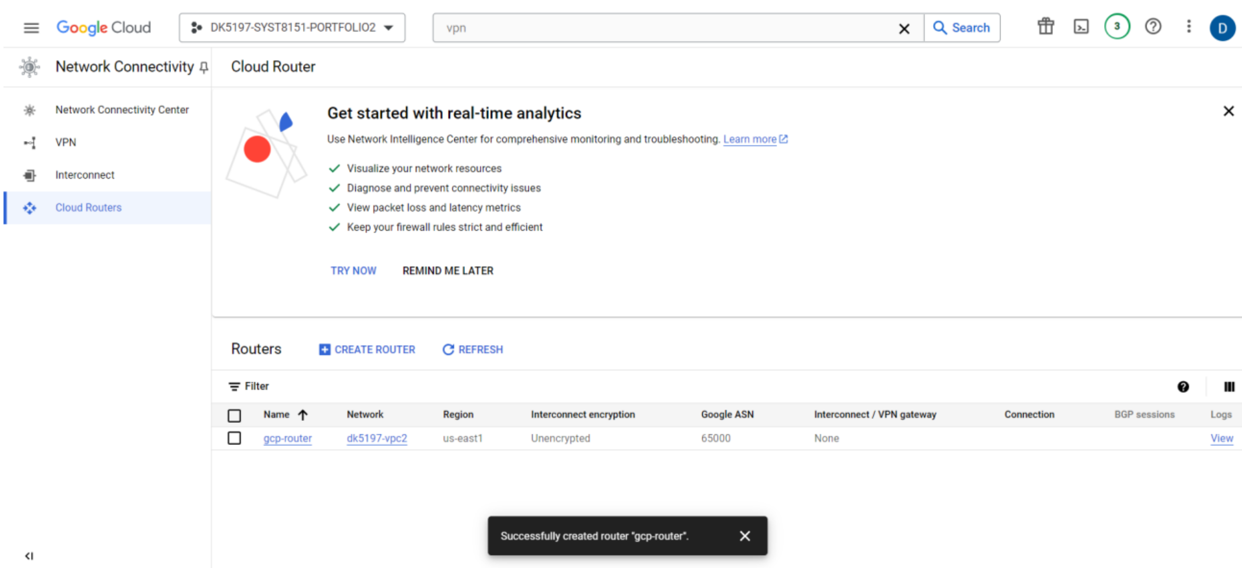

Figure 39 - Successfully created a new router

Figure 40 - Selecting HA VPN for the tunnelling in VPN

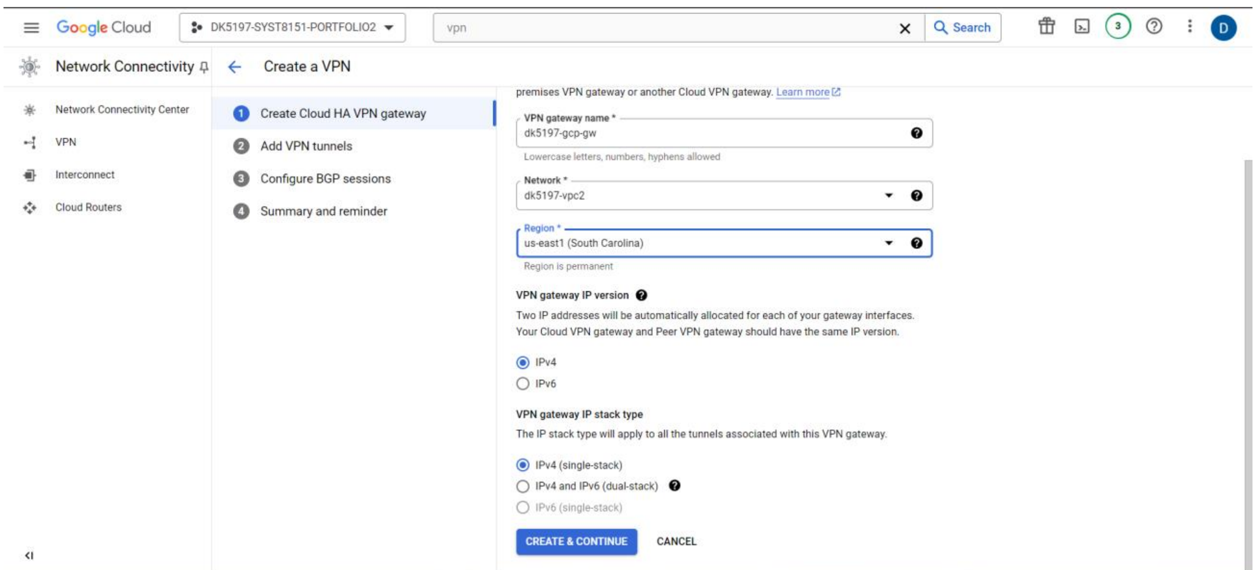

Figure 41 - Creating a new HA VPN gateway

In AWS, we created a new Virtual Private Gateway and named it aws-vpn, then attached it to our VPC, dk5197-VPC1. After setting up the Virtual Private Gateway, we needed to create Customer Gateways to represent the external network we wanted to connect to via the VPN. We named the first Customer Gateway dk5197-gcp-gw1, used 65000 as the ASN (matching Google Cloud's ASN), and set the IP address to the first interface IP of the Google Cloud router we had previously noted. Similarly, we created a second Customer Gateway named dk5197-gcp-gw2, using the second interface IP address.

With these configurations in place, we proceeded to create a new VPN connection in AWS. We named it aws-gcp-vpn, selected the transit gateway type as a Virtual Private Gateway, and linked it to the Customer Gateways. For both tunnel 1 and tunnel 2, we used the pre-shared key “hellocloud” for authentication between the Virtual Private Gateway and the Customer Gateways. The VPN connection was successfully created on the AWS side. Finally, we downloaded the VPN connection configuration, which included the Virtual Private Gateway addresses for each tunnel.

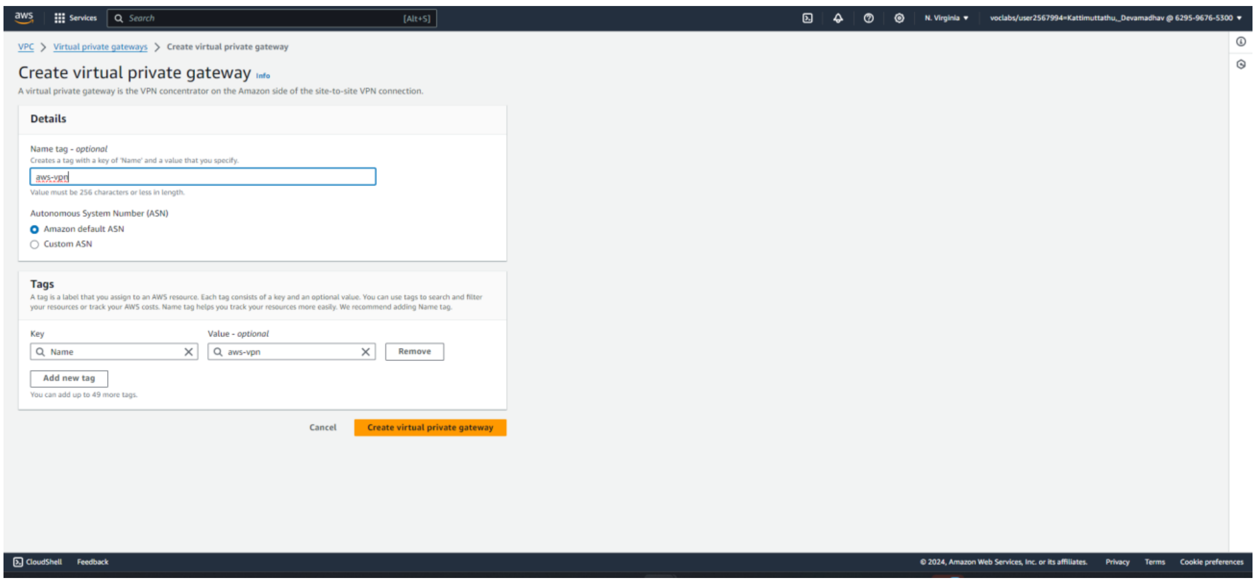

Figure 42 - Creating the virtual private gateway in AWS

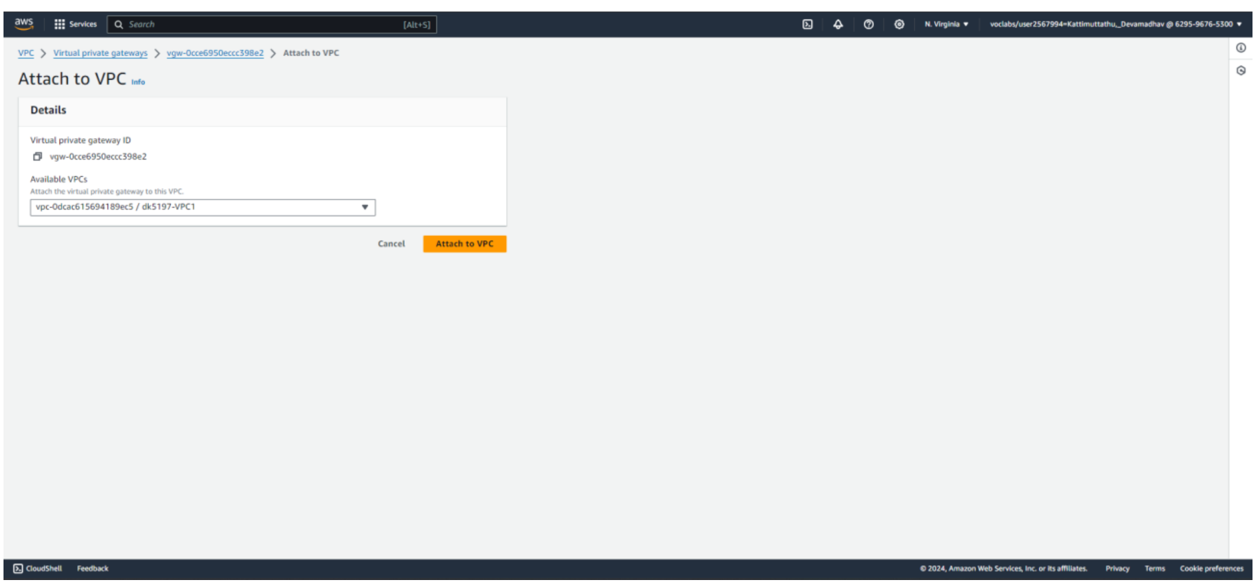

Figure 43 - Attaching the virtual private gateway to vpc

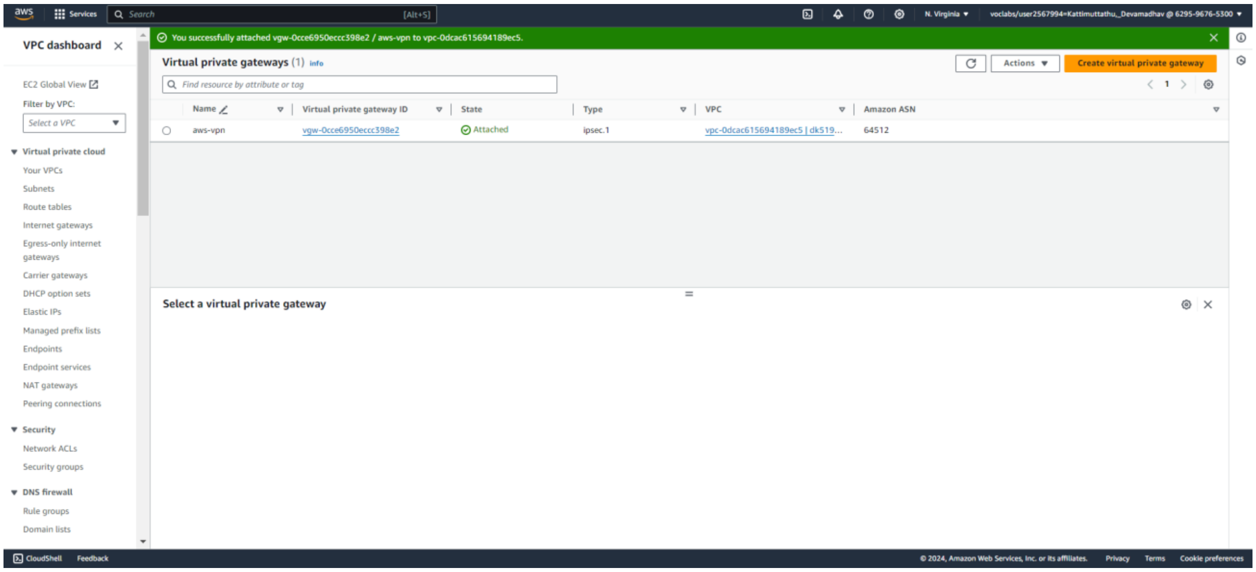

Figure 44 - Successfully attached the virtual private gateway to th the VPC

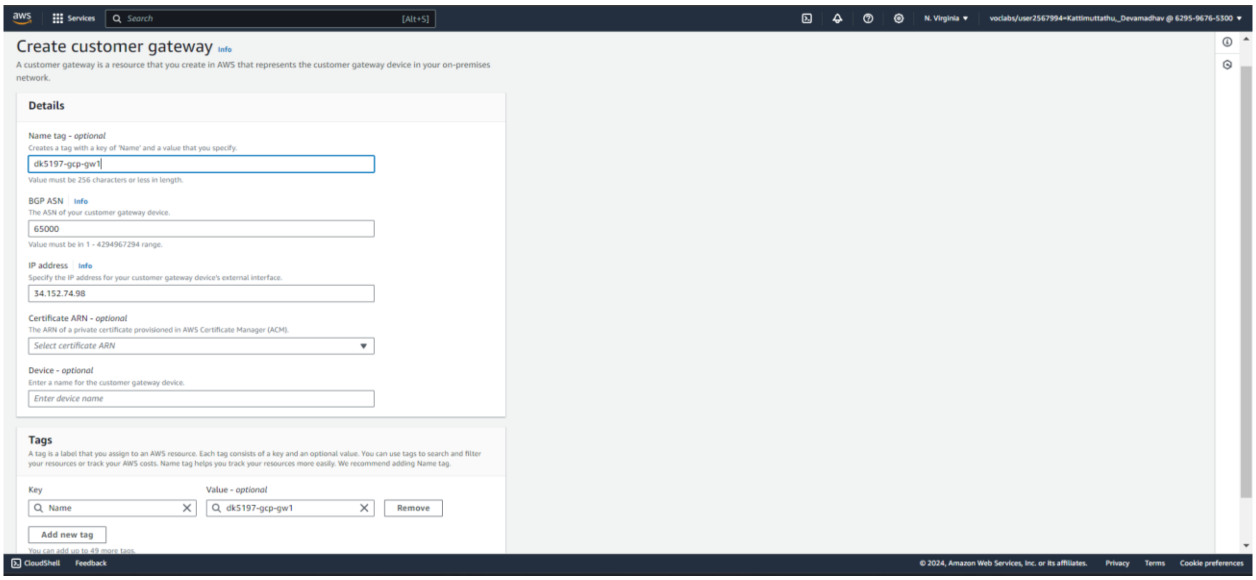

Figure 45 - Creating a new customer gateway with the interface0 of gcp as IP

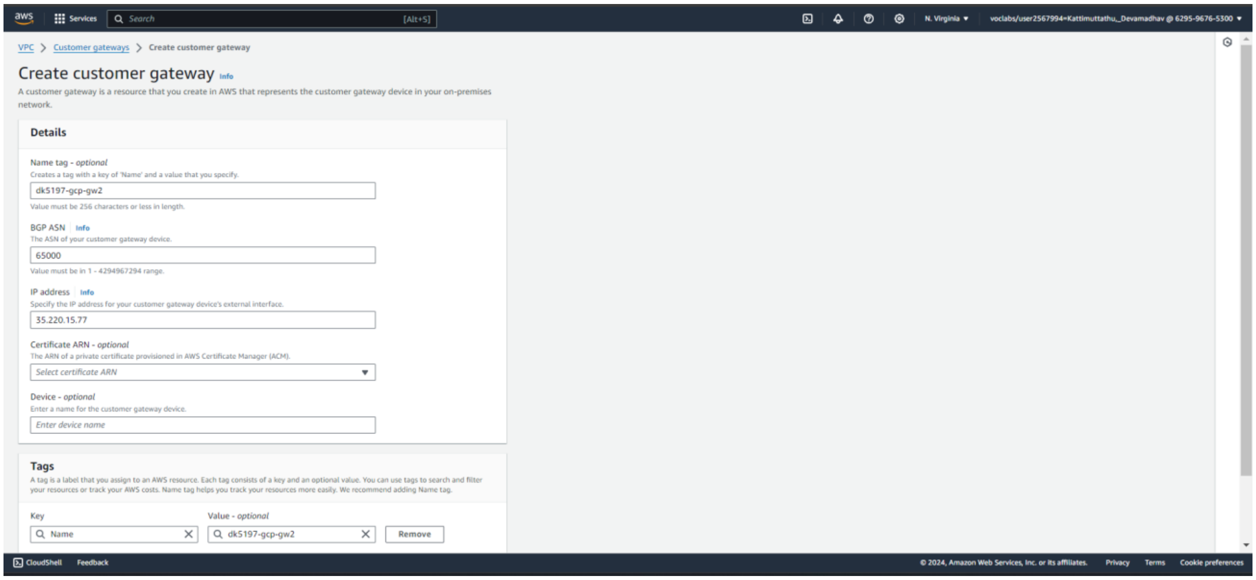

Figure 46 - Creating the second customer gateway with the interface1 of gcp as IP

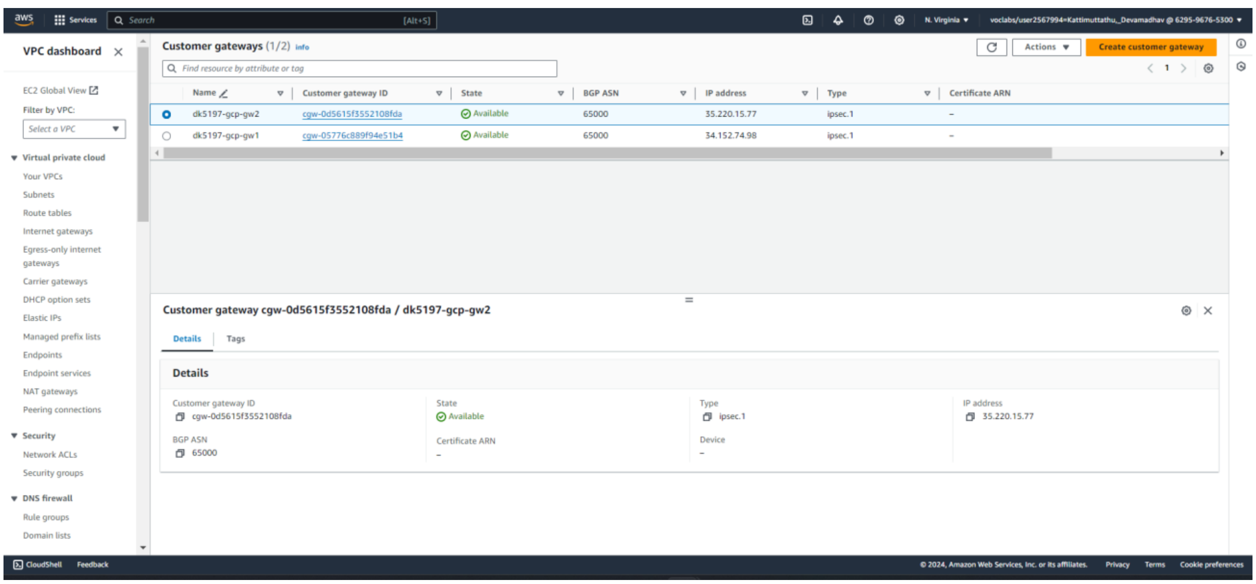

Figure 47 - Successfully created two customer gateways

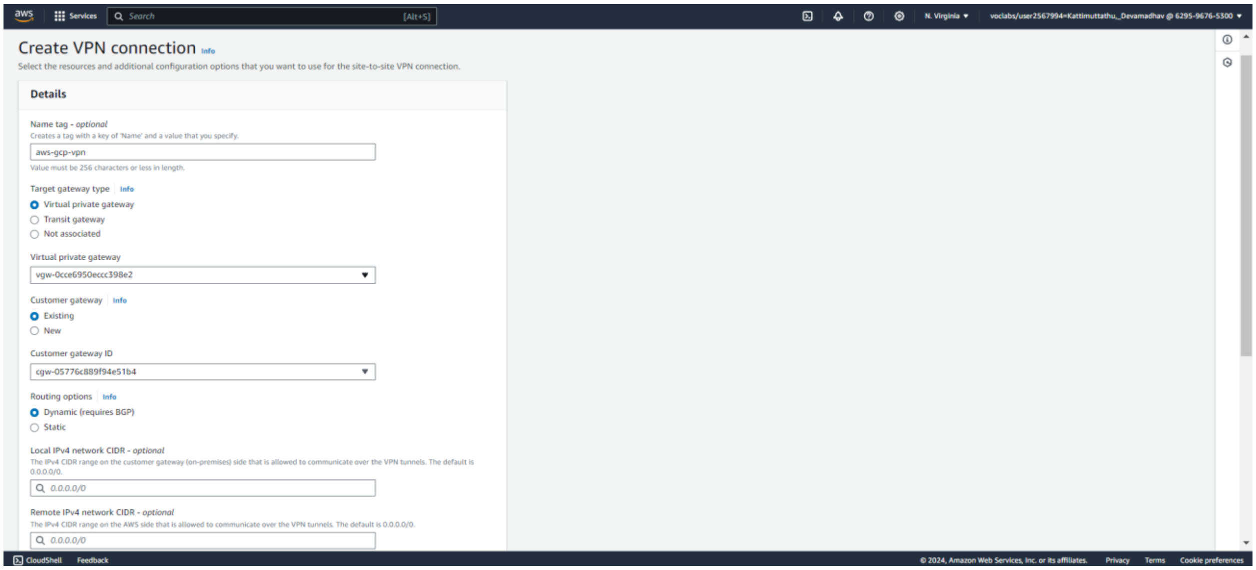

Figure 48 - Creating a new vpn connection in aws

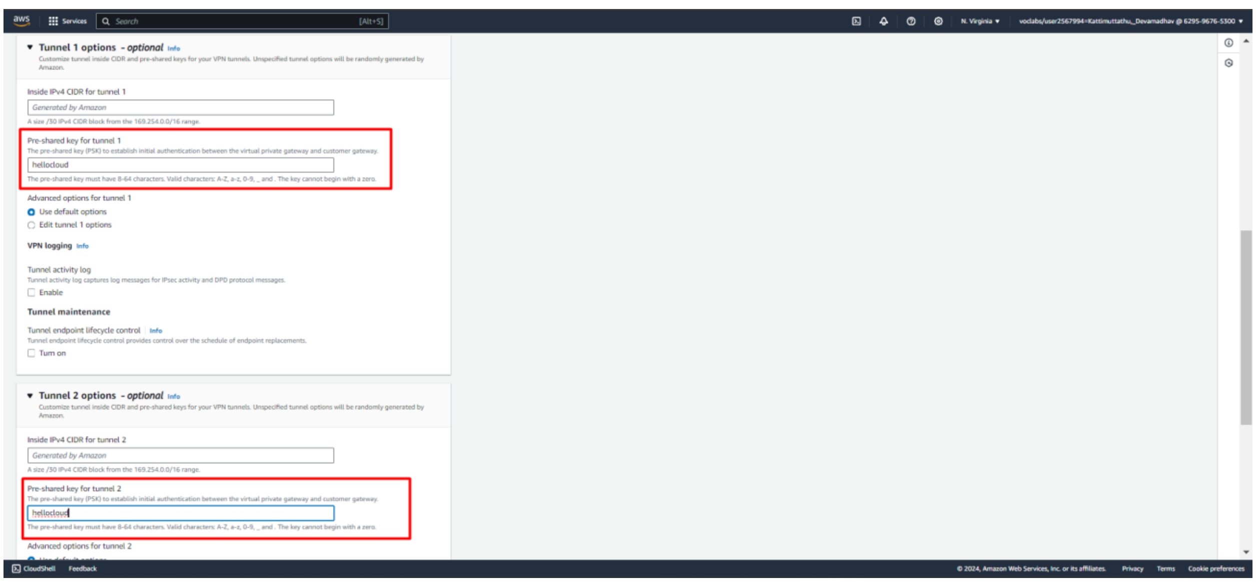

Figure 49 - Configuring tunneling in VPN connection and set hellocloud as preshared key for authentication

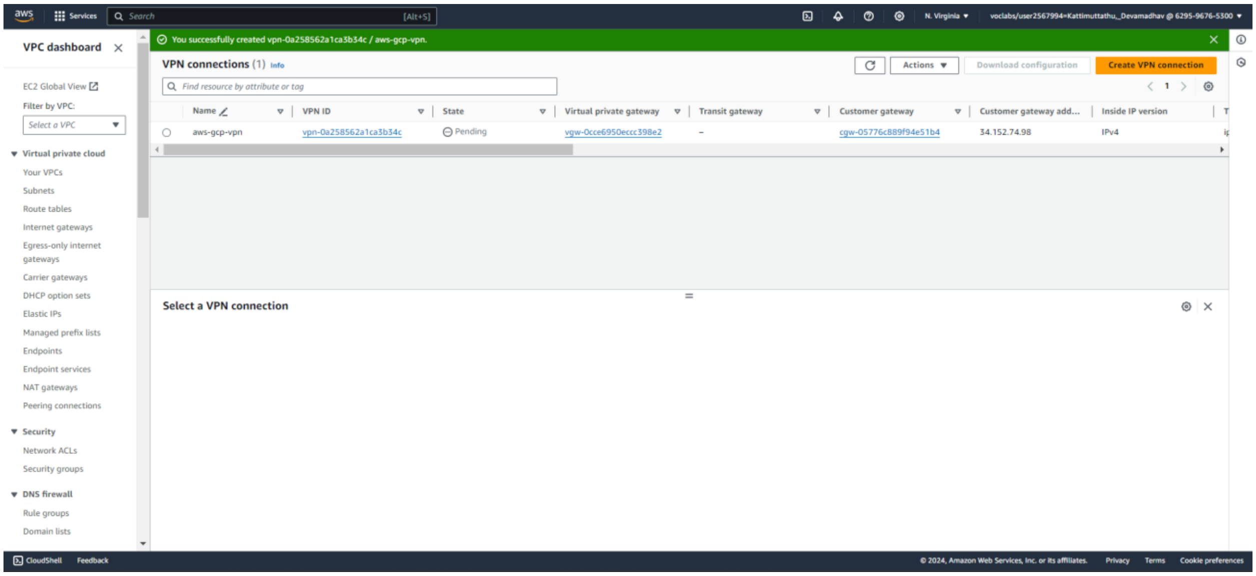

Figure 50 - Successfully created a new vpn connection

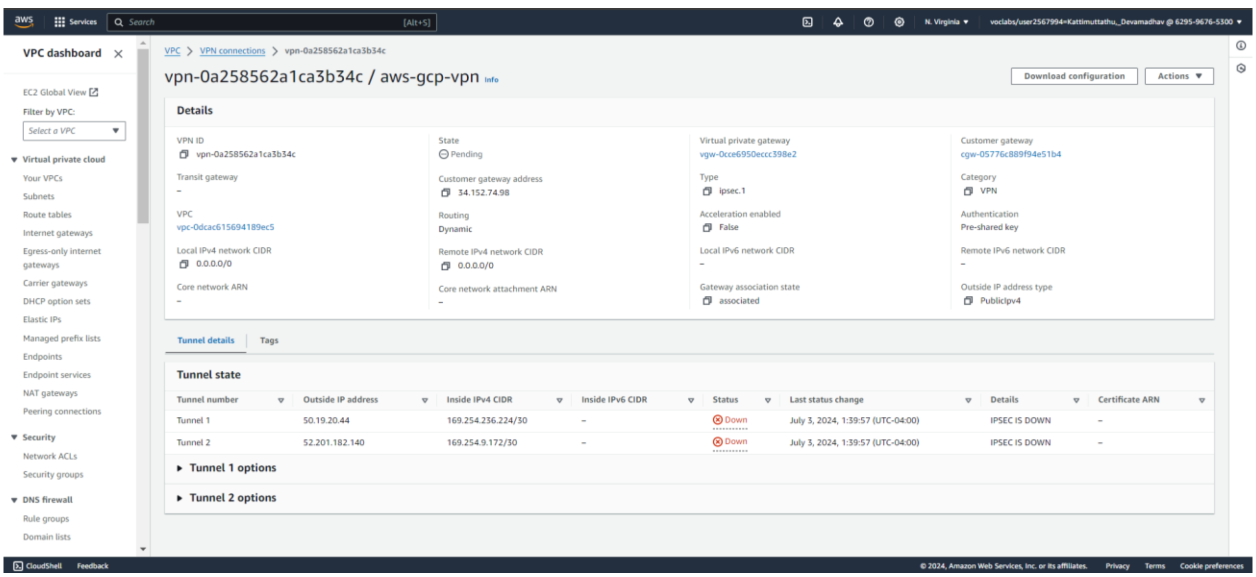

Figure 51 - Tunnels are in down state

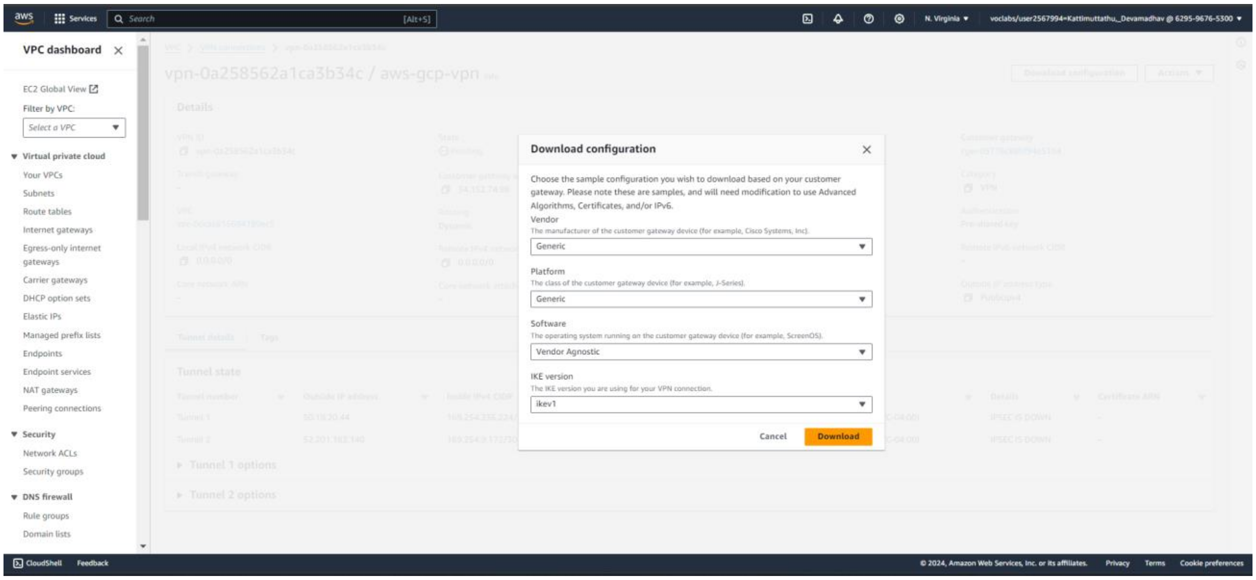

Figure 52 - Downloaded the vpn configuration file

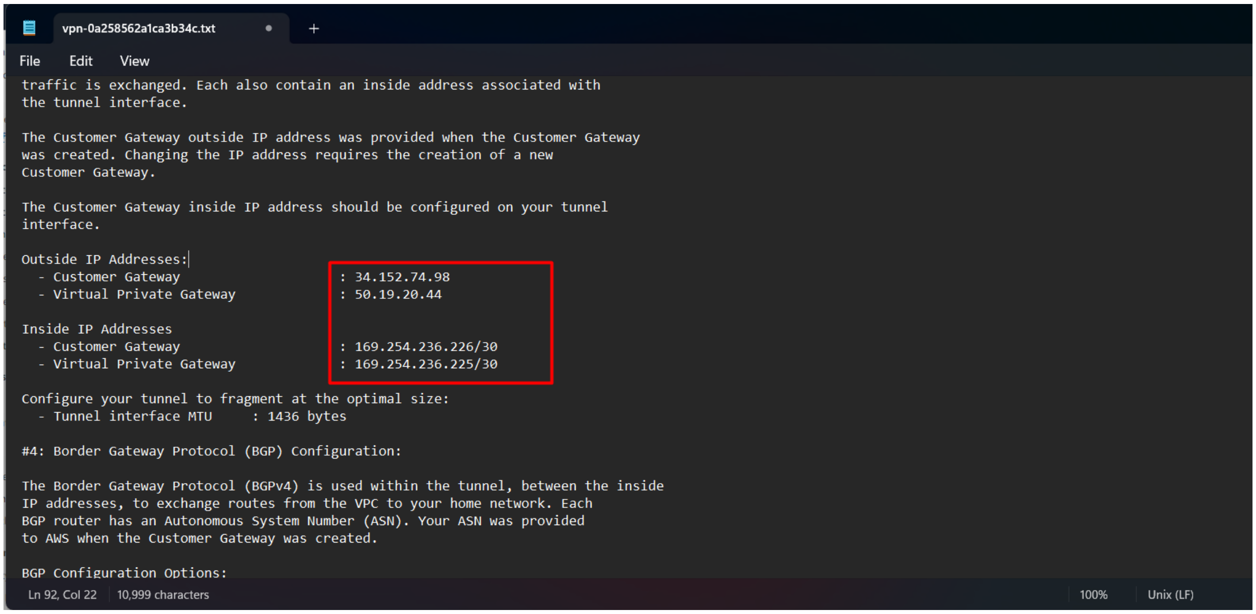

Figure 53 - Virtual private gateway details for tunnel1'

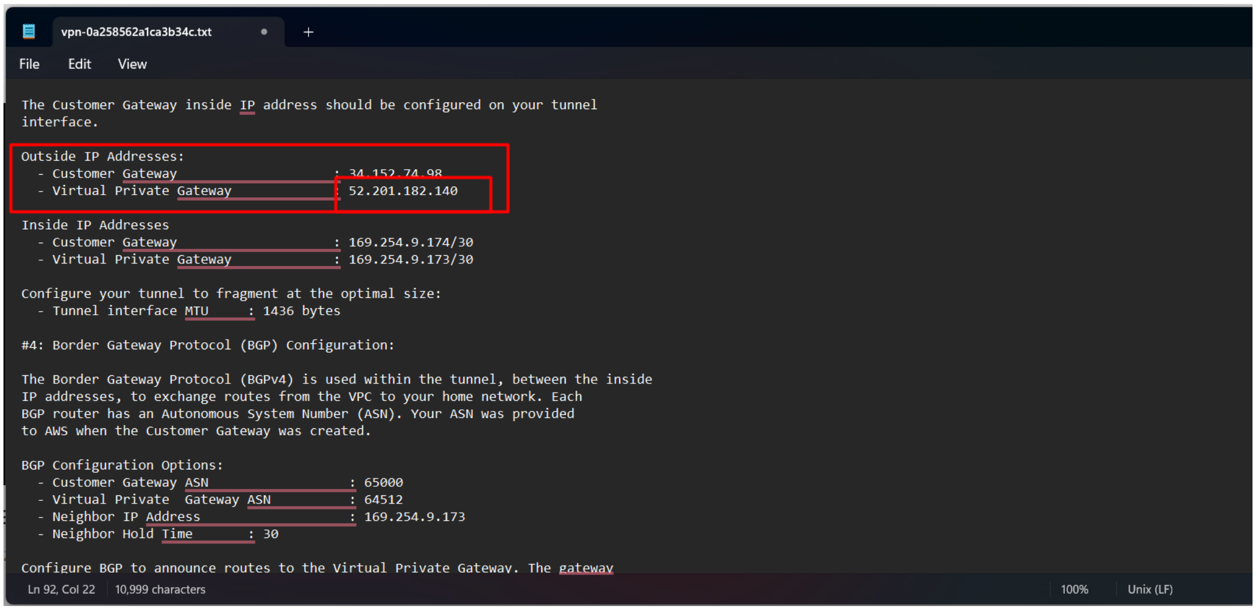

Figure 54 - Virtual private gateway detail for tunnel2

After configuring the VPN tunnel in AWS, we returned to the Google Cloud console to set up the corresponding VPN gateway. We created a new VPN gateway in GCP using the interface address of the AWS Virtual Private Gateway we had just established. We selected our newly created router for the VPN and configured the VPN gateway interfaces, naming them dk5197-gcp-tunnel and dk5197-gcp-tunnel2. We used the same pre-shared key, “hellocloud,” for authentication between the AWS and GCP gateways. The VPN gateway was successfully created.

Next, we configured the BGP session for tunnel 1, naming it dk5197-bgp, with AWS’s ASN set to 64512. We used the inside IP address from the AWS VPN tunnel configuration. After setting up the BGP session, the tunnel was successfully established. We confirmed this by checking the AWS console, where the tunnel status had changed to "UP."

Additionally, we updated the AWS route table to include a new route directing traffic to the Google Cloud network (10.173.46.0/24) through the Virtual Private Gateway. Afterward, we successfully pinged the AWS instance from the Google Cloud instance, confirming the connection. We also tested the connection by pinging google.com from the instance.

With these steps, we successfully established communication between the two cloud networks across different platforms.

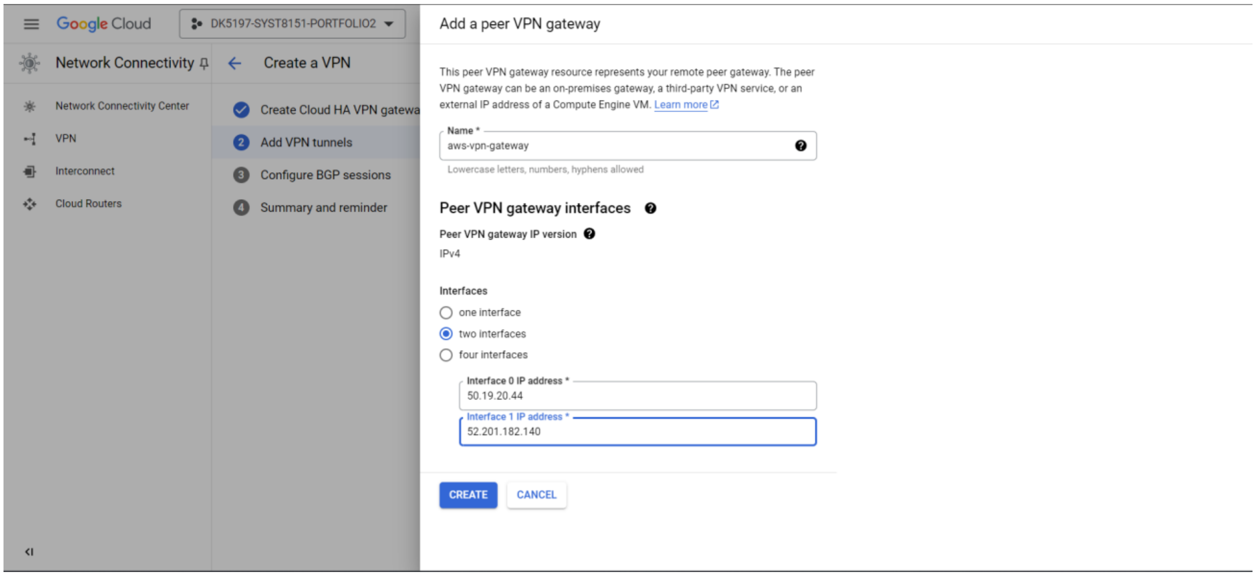

Figure 55 - Configuring the peer VPN gateway in gcp using the virtual private gateway from aws

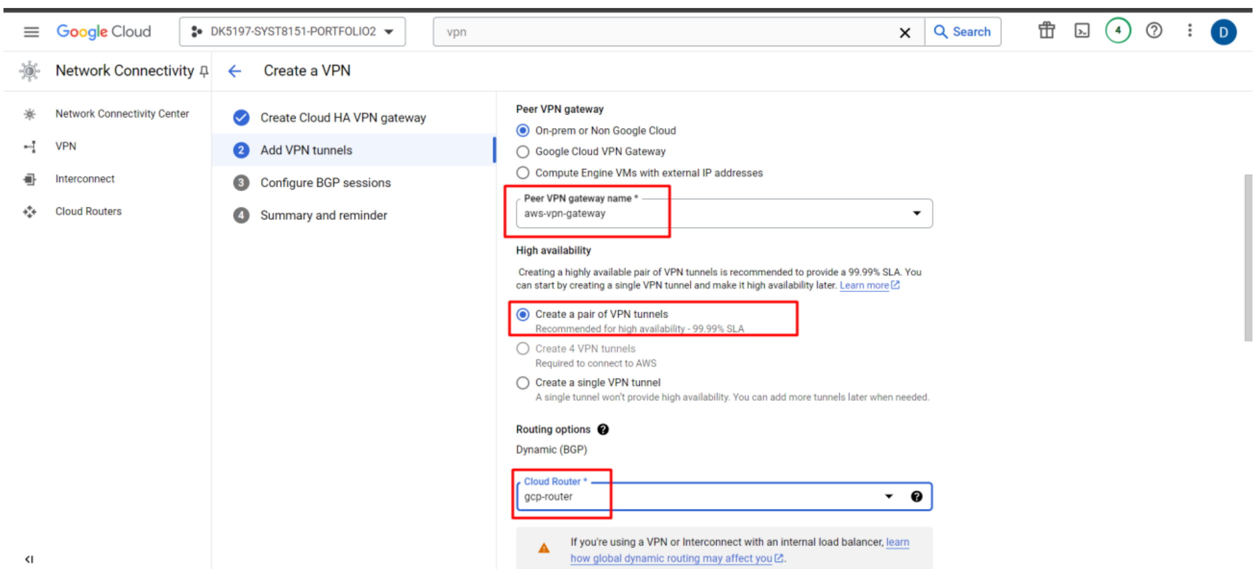

Figure 56 - Creating a vpn tunne in gcp

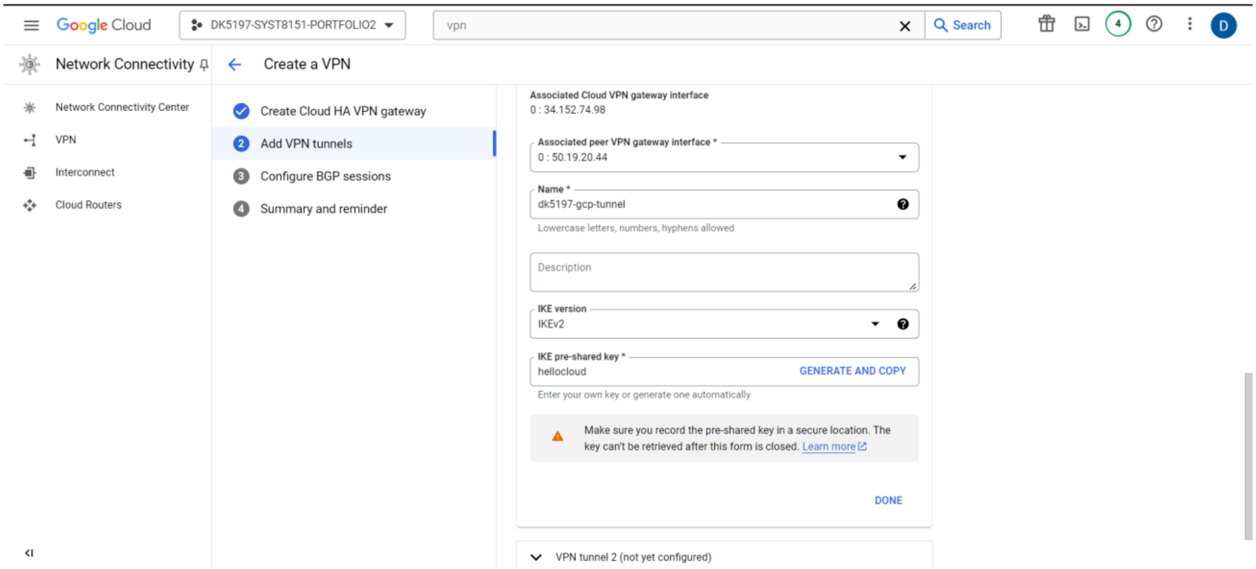

Figure 57 - Conffiguring tunnel1 details for vpn

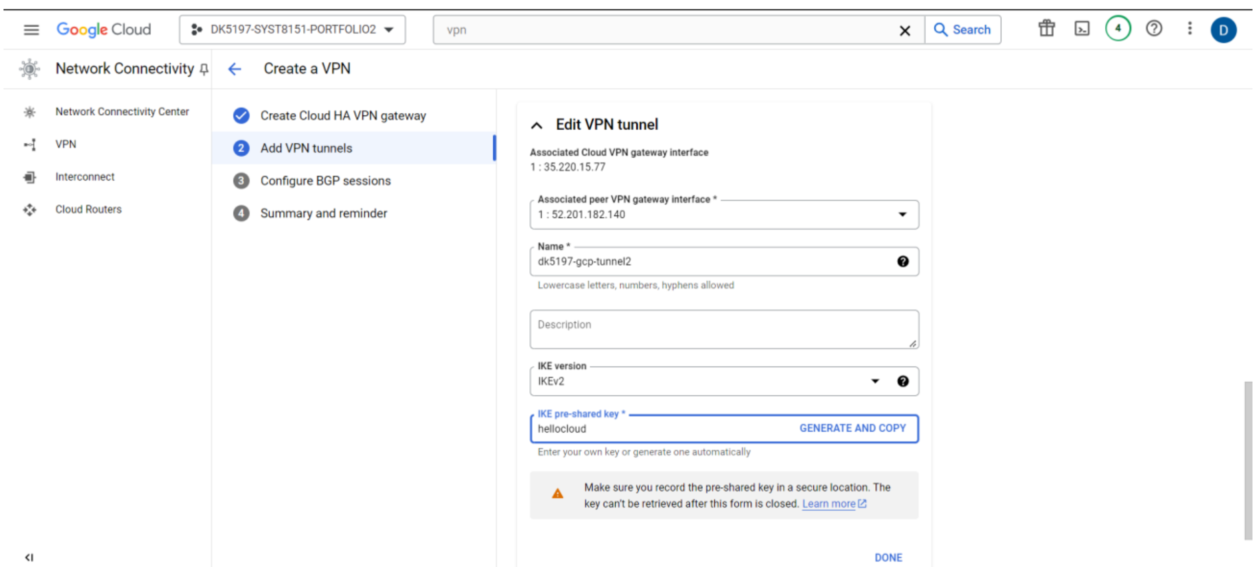

Figure 58 - Configuring tunnel2 details in vpn

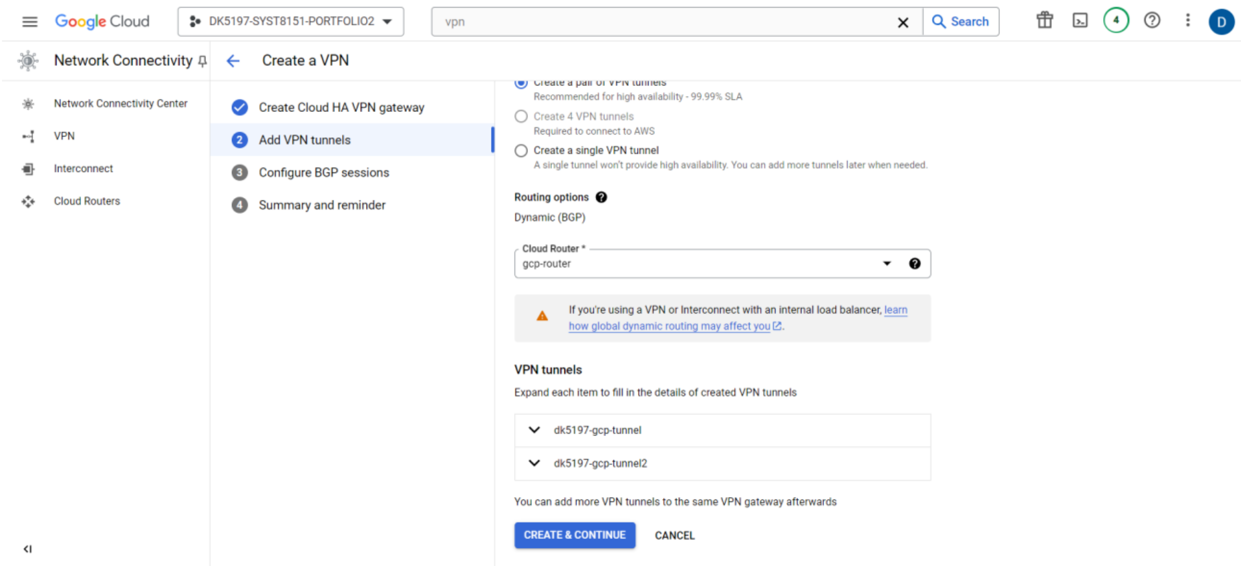

Figure 59 - Successfully configured all details for vpn tunnel

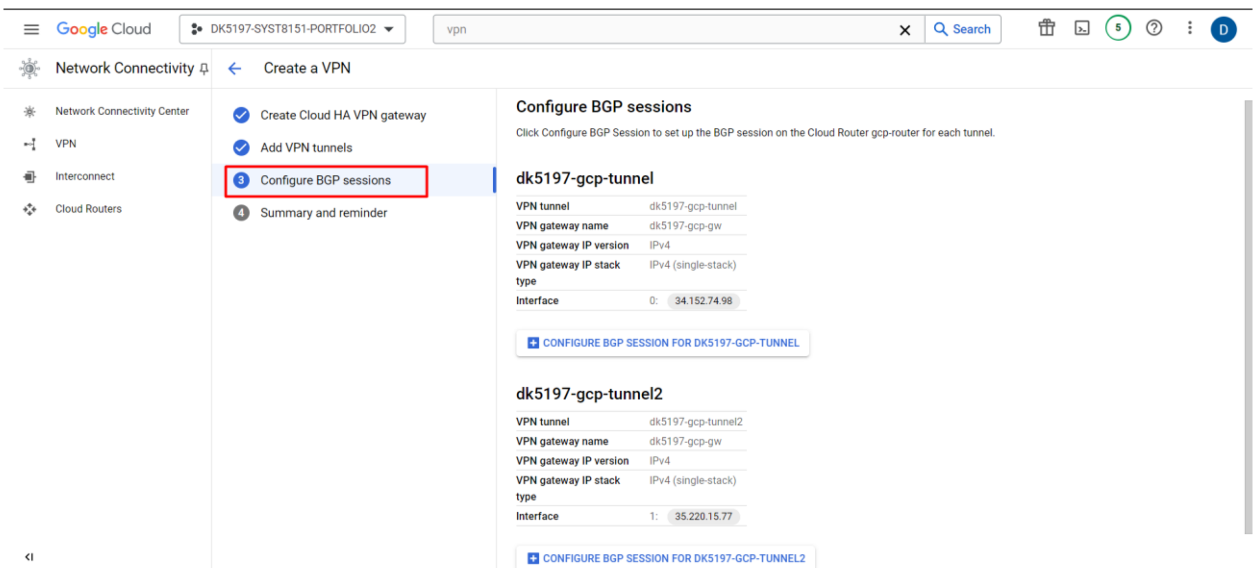

Figure 60 - Configuring the BGP sessions for the VPN

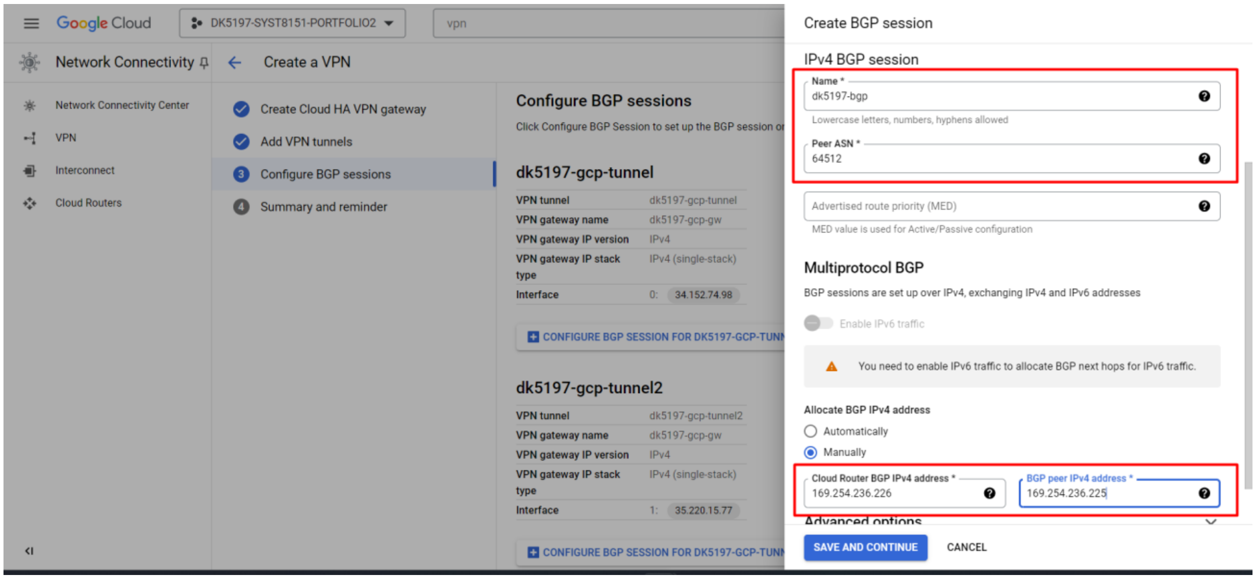

Figure 61 - Creating a new BGP connection for tunnel1

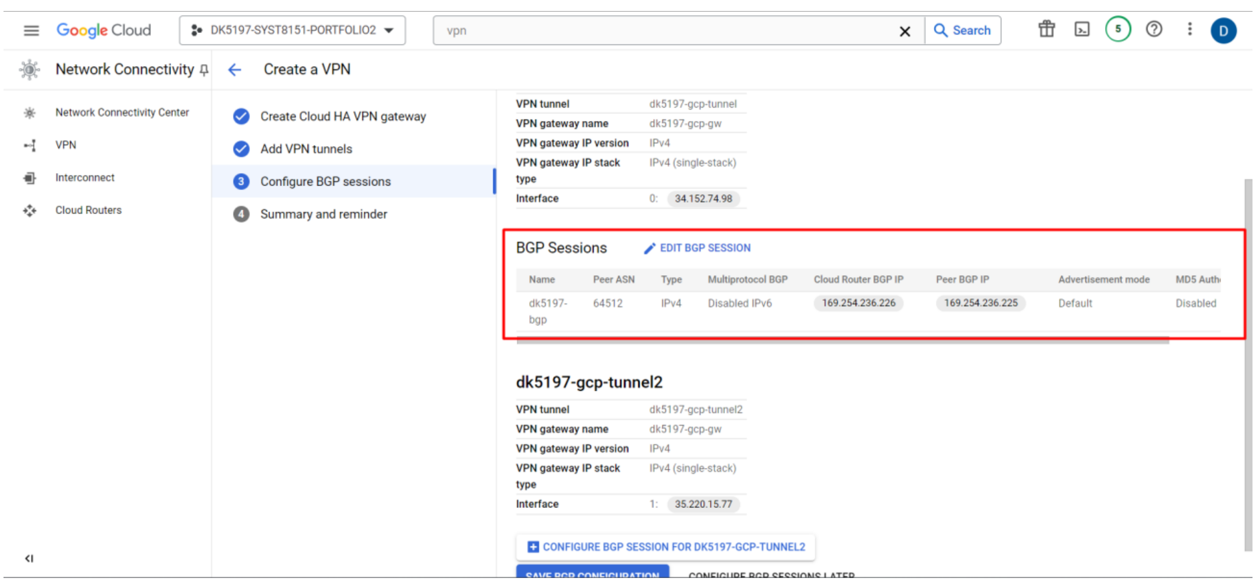

Figure 62 - Successfully created a new bgp session

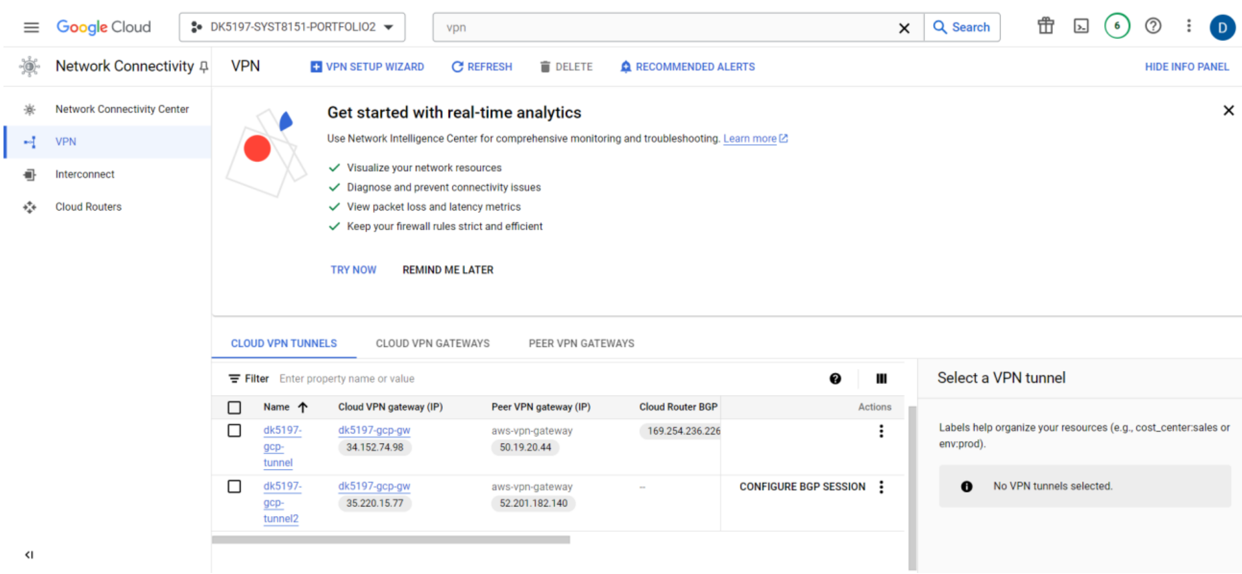

Figure 63 - Successfully created two vpn tunnels

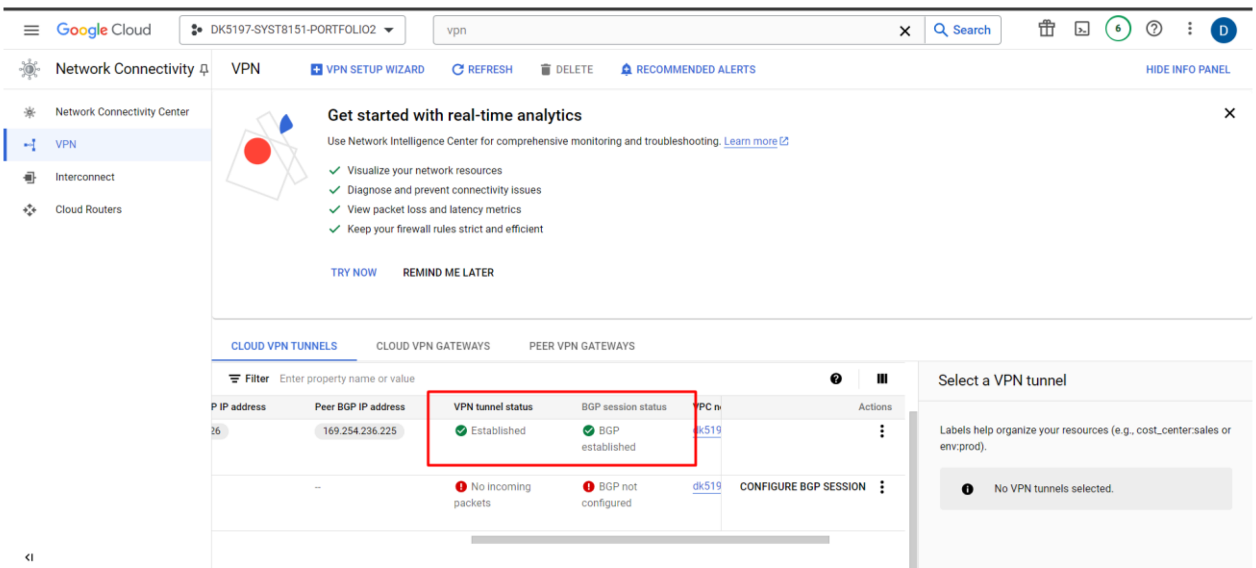

Figure 64 - Tunnel1 status is in established state

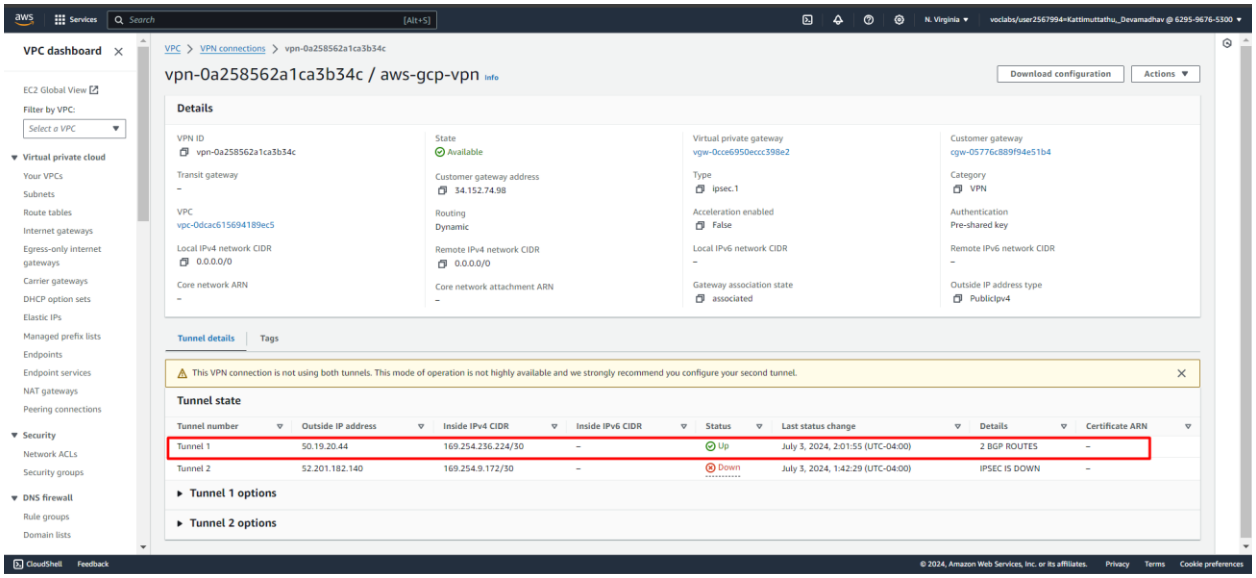

Figure 65 - Tunnel status in aws has changed to UP .Thereby enabling communication

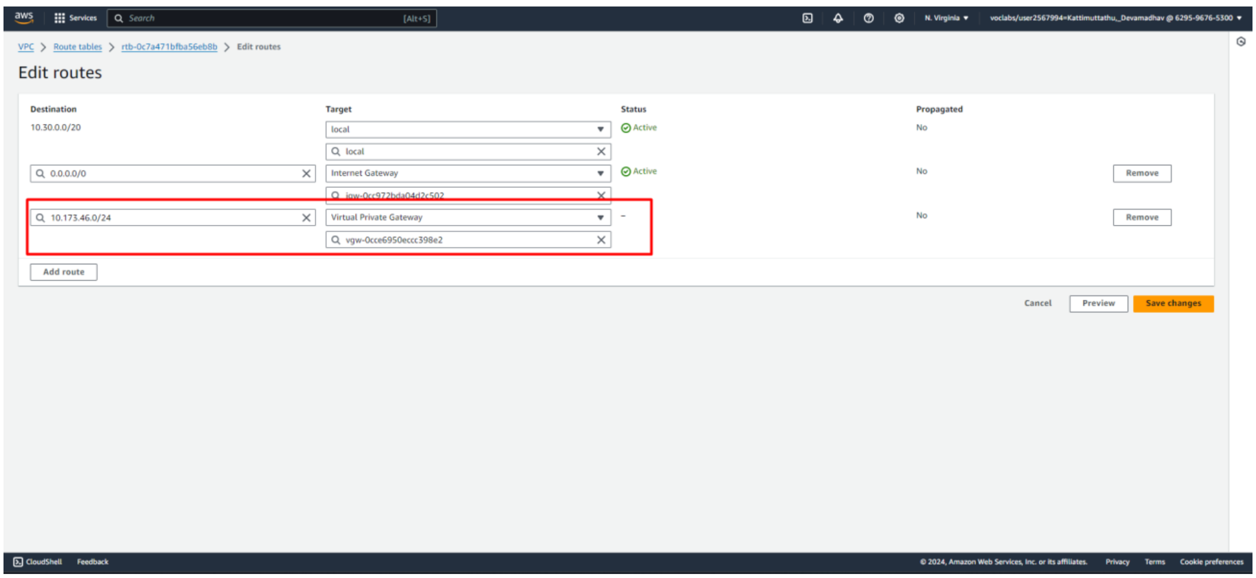

Figure 66 - Creating a new route to the gcp through virtual private gateway

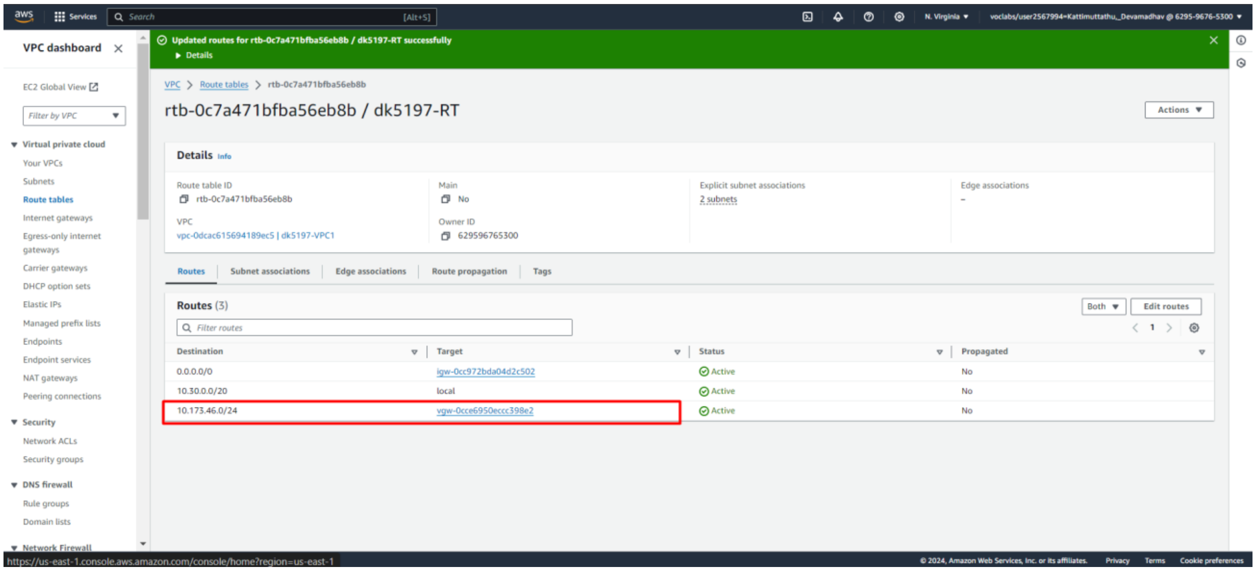

Figure 67 - Successfully created a new route

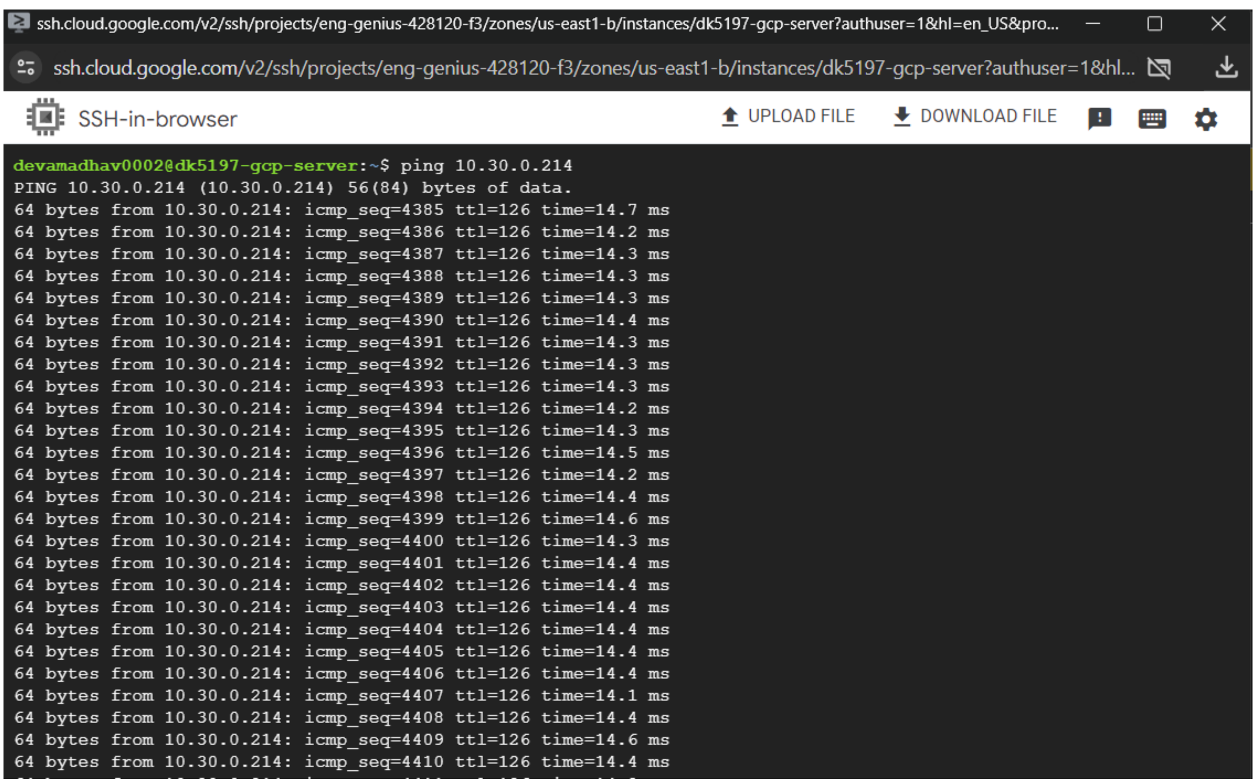

Figure 68 - Successful communication with aws instance from gcp instance

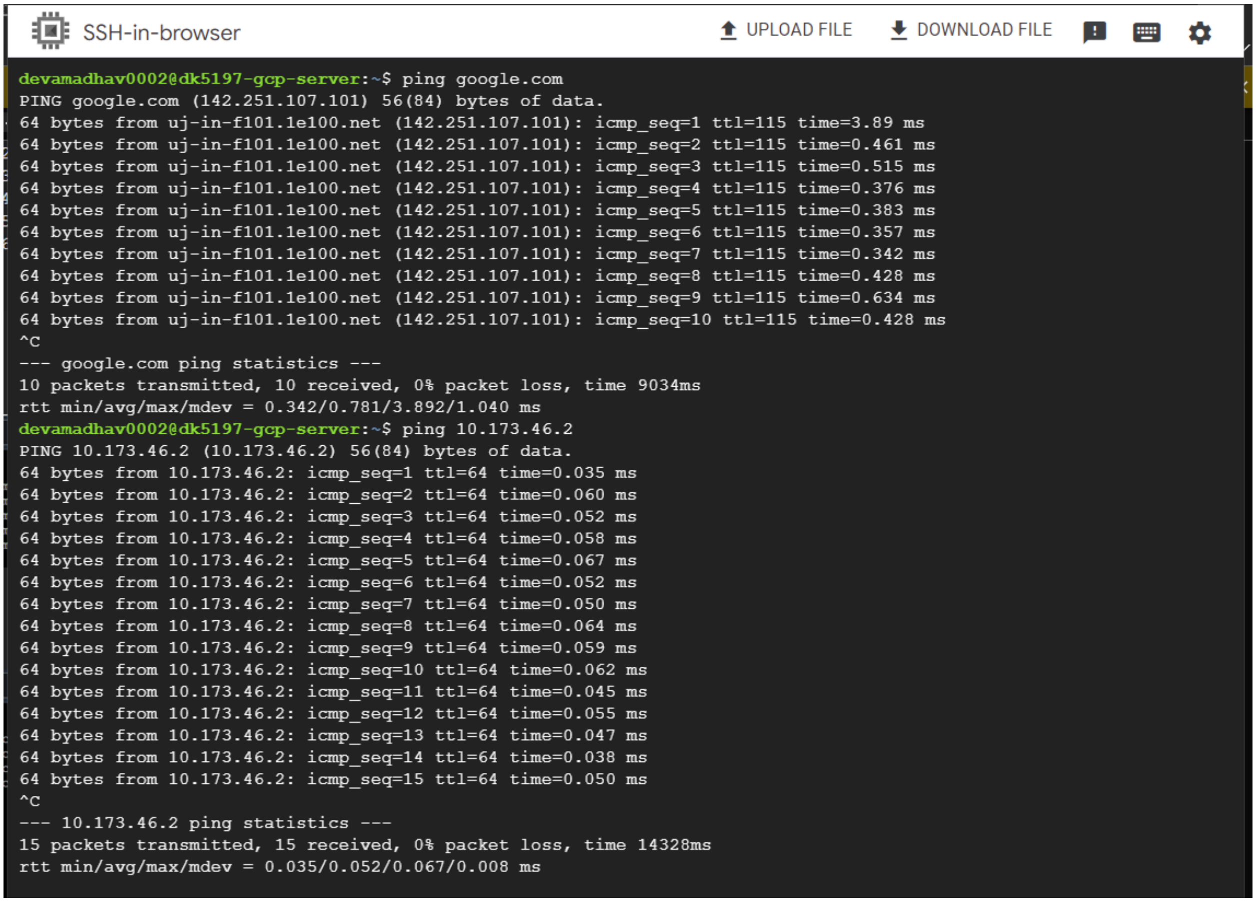

Figure 69 - successfully verified the connection