PROJECT TITLE: “Hybrid Cloud Secure Site-to-Site VPN between AWS and GCP”

PROJECT OVERVIEW:

The objective of this project was to establish a **secure and reliable communication channel** between AWS and Google Cloud Platform (GCP) environments to enable **hybrid cloud operations**, **resource sharing**, **disaster recovery**, and **failover mechanisms** across cloud platforms. The solution was aimed at **supporting business continuity**, reducing vendor lock-in, and **enabling multi-cloud workload distribution**.

The project was initiated to establish a **secure and reliable communication link** between an organisation’s **on-premise data centre** and its newly adopted **AWS cloud environment**. The organisation needed to extend its internal services (like databases, file servers, and internal applications) hosted on-premises to the cloud without exposing them to the Internet.

A **Site-to-Site VPN** was chosen as the solution to securely route internal traffic between both networks, ensuring data confidentiality, integrity, and availability.

PURPOSE:

The purpose of this project was to **enable secure, scalable, and resilient communication between cloud environments hosted on AWS and GCP**. Organizations are increasingly adopting **multi-cloud strategies** to leverage the unique advantages of different cloud providers, reduce vendor lock-in, and optimize costs. Enable secure, encrypted communication between isolated Virtual Private Clouds (VPCs) hosted in Amazon Web Services (AWS) and Google Cloud Platform (GCP), allowing seamless interoperability, hybrid cloud deployment, and disaster recovery between the two cloud environments.

A multinational company decided to adopt a **multi-cloud strategy** to:

- Leverage GCP’s analytics capabilities (e.g., BigQuery).
- Use AWS for scalable application hosting and storage (e.g., S3, EC2).
- Maintain **data and service connectivity** between both environments.

They needed **secure communication** between these two clouds over the internet using **IPsec VPN tunnels** for a **reliable and encrypted hybrid cloud network**.

ABOUT VPN:

AWS Site-to-Site VPN is a fully-managed service that creates a secure connection between your data centre or branch office and your AWS resources using IP Security (IPSec) tunnels. When using Site-to-Site VPN, you can connect to both your Amazon Virtual Private Clouds (VPC) as well as AWS Transit Gateway, and two tunnels per connection are used for increased redundancy.

For globally distributed applications, the Accelerated Site-to-Site VPN option provides even greater performance by working with AWS Global Accelerator to intelligently route your traffic to the nearest AWS network endpoint with the best performance.

Types of VPN in Cloud Networking

1. **Site-to-Site VPN**

    - Connects **two networks** (e.g., AWS VPC and GCP VPC) over the internet.
    - Uses **IPSec (Internet Protocol Security)** for encryption and authentication.
    - Ideal for **Hybrid or Multi-Cloud Architectures**.

2. **Client-to-Site VPN**

    - Connects an individual user's device to a private cloud network.
    - Used for remote user access (e.g., employees working from home).

3. **Cloud Provider VPN**

    - Managed VPN services from cloud vendors (e.g., **AWS Site-to-Site VPN**, **GCP Cloud VPN**) that simplify setup and maintenance.

TECHNOLOGIES USED:

AWS Services:

Amazon VPC

- Virtual Private Gateway (VGW)
- Customer Gateway (CGW)
- Site-to-Site VPN
- EC2
- CloudWatch

GCP Services:

- Google Cloud VPC
- Cloud VPN Gateway
- Cloud Router
- Compute Engine

Security & Networking Tools:

- IPsec (IKEv2)
- BGP (Border Gateway Protocol)
- Cloud Firewalls (AWS Security Groups, GCP Firewall Rules)

ROLES AND RESPONSIBILITIES:

- Designed and provisioned secure network topology across AWS and GCP.
- Configured VPN gateways and routing (static/BGP).
- Automated VPN deployment using **Terraform and CLI scripts**.
- Implemented logging, monitoring, and alerting systems.
- Ensured compliance with security policies and standards (IPSec encryption, firewall rules).
- Performed connectivity testing, latency benchmarking, and performance tuning.

SCOPE OF THE PROJECT:

- Establish a **resilient Site-to-Site VPN tunnel** between AWS and GCP.
- Enable **cross-cloud communication** for critical workloads (APIs, microservices, databases).
- Provide **high availability** and **automatic failover** using dual tunnels and BGP.
- Implement **infrastructure automation** for repeatable deployments across dev, test, and prod.
- Support **hybrid multi-cloud data replication**, centralized logging, and disaster recovery.

PROBLEM IT SOLVED:

- Removed the dependency on less secure and unreliable public internet access to AWS services.
- Solved latency and packet loss issues faced with legacy third-party connectivity tools.
- Addressed compliance and data security requirements by encrypting data in transit.
- Improved access control and network visibility for IT and DevOps teams.

BUSINESS BENEFITS:

• **Secure cross-cloud data flow**: Enabled seamless and encrypted communication between AWS and GCP workloads using IPSec tunnels.

• **Reduced costs**: Leveraged cost-effective VPN and routing solutions, minimizing the need for expensive direct connectivity (e.g., Direct Connect, Interconnect).

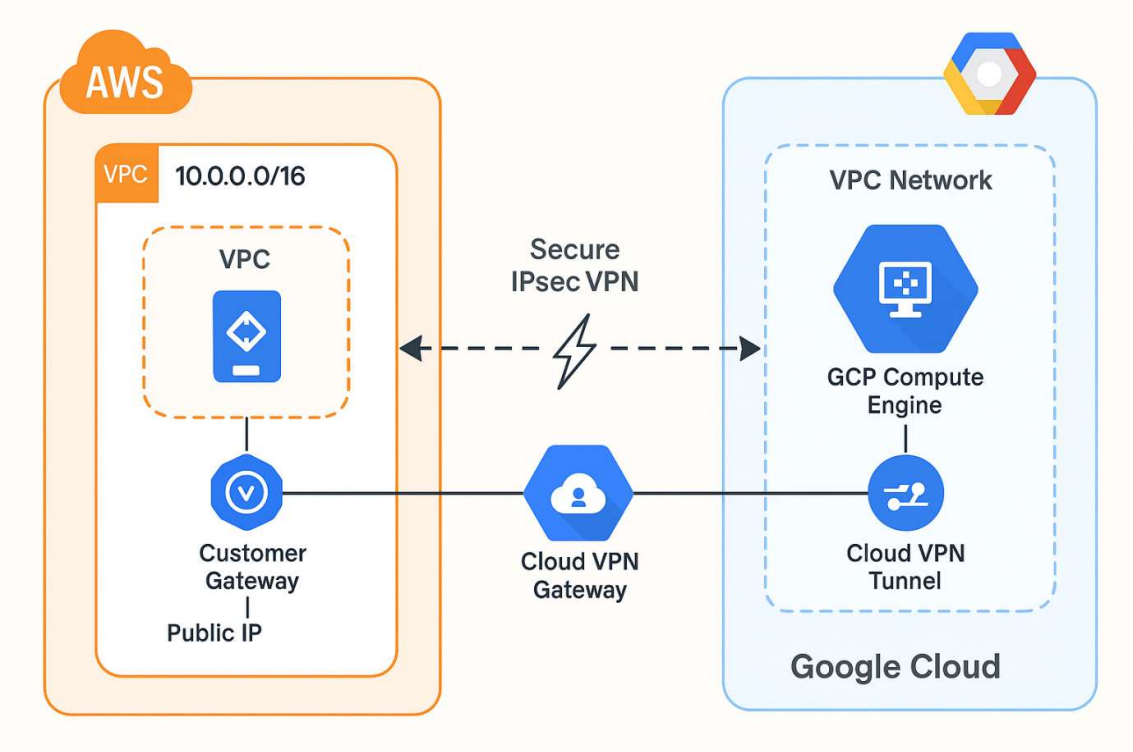
• **Improved efficiency**: Automated VPN setup with Terraform and CLI tools reduced manual configuration time by over 60%.

• **Reduced downtime**: Implemented dual IPSec tunnels with BGP for high availability and quick failover between clouds.

• **Enhanced security**: Used strong encryption protocols (IPSec/IKEv2) to ensure data confidentiality, integrity, and secure key exchange.

• **Improved scalability**: Designed a flexible, cloud-agnostic architecture that supports scaling workloads across both AWS and GCP as needed.

AWS VPN: SET UP SITE-TO-SITE VPN ARCHITECTURE



Step-by-Step Guide

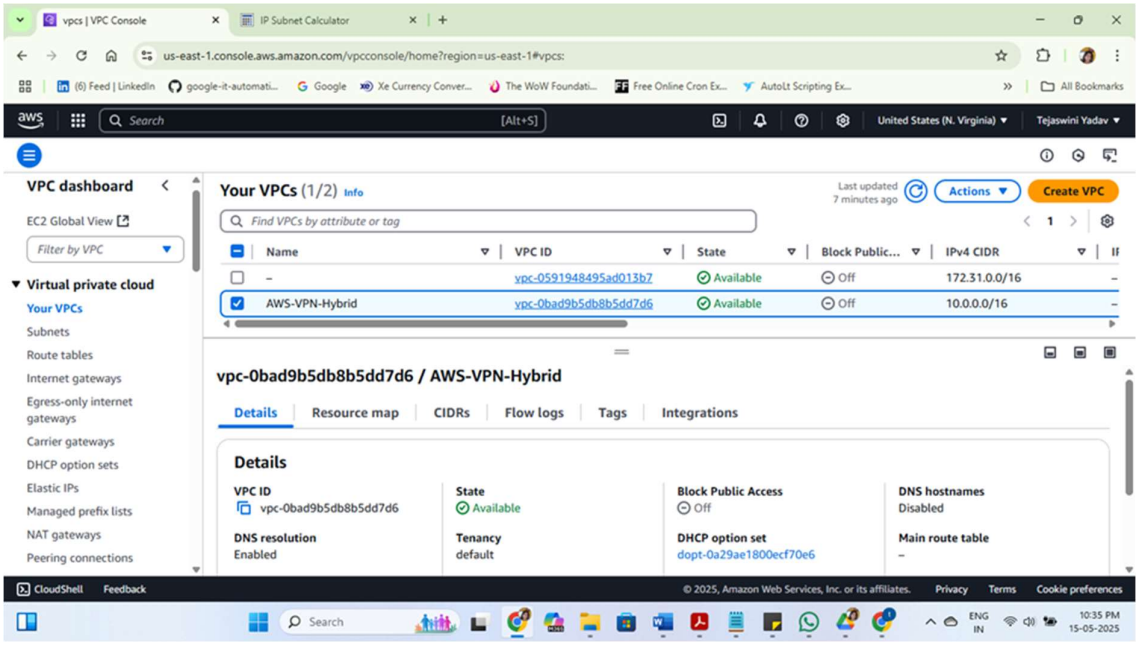
Prerequisites:

First, we need to create the foundational infrastructure required for establishing secure Site-to-Site VPN connectivity between the AWS Cloud and the on-premises network (GCP).

STEP 1: Create a VPC in AWS

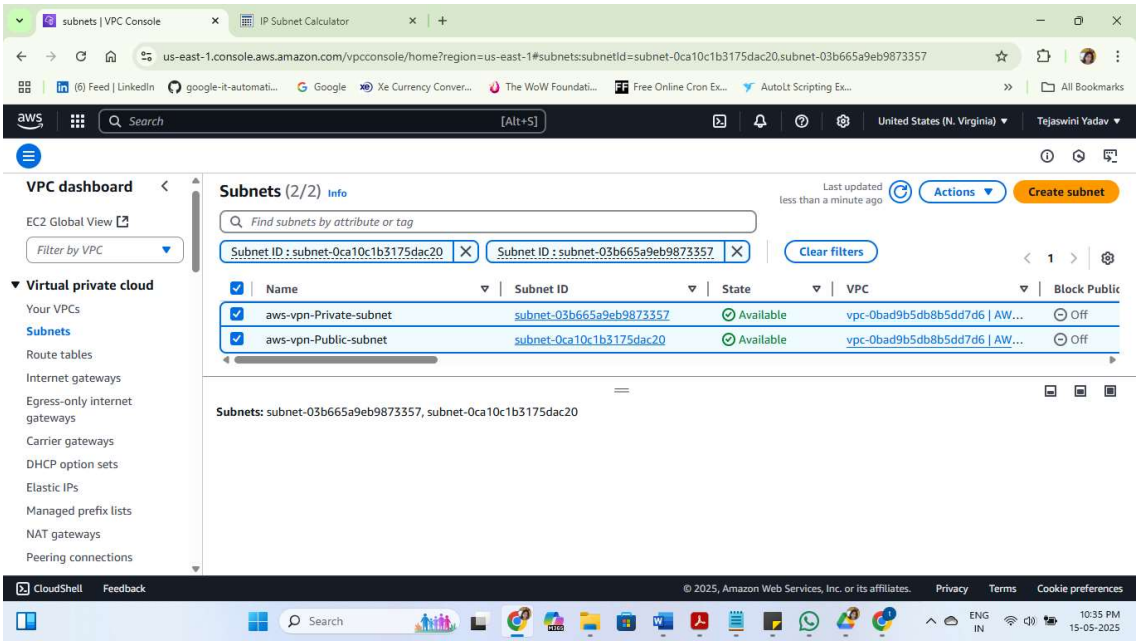
Go to the VPC Dashboard in AWS Console.

- **VPC Name: AWS-VPN-Hybrid**
- **IPv4 CIDR : 10.0.0.0/16**
- **Select Remaining Default**
- **Create VPC on AWS**



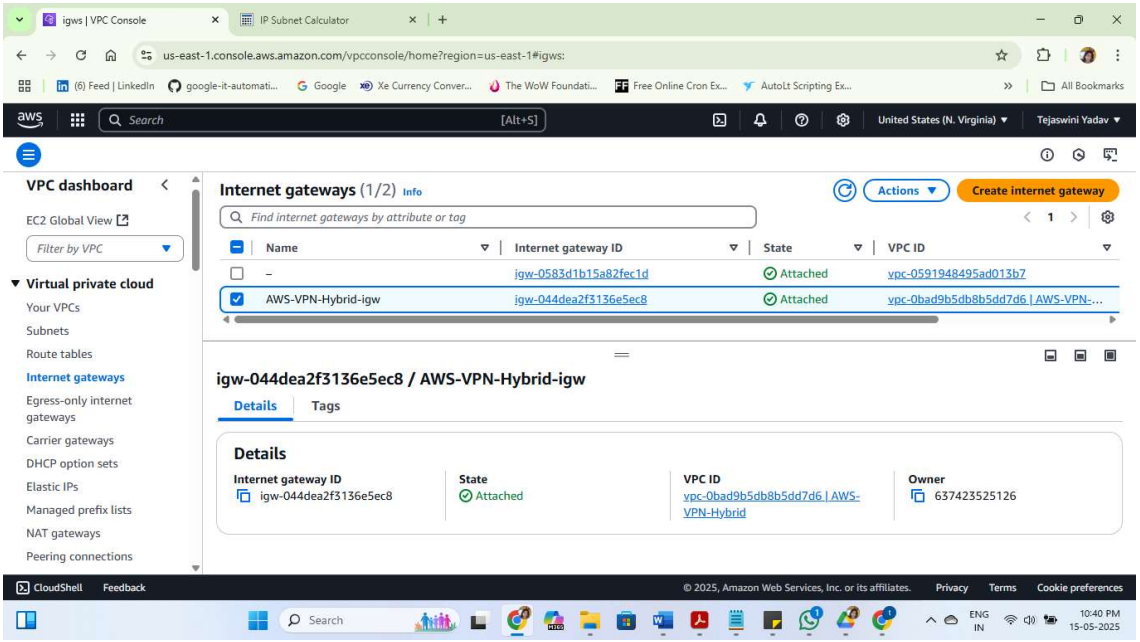
Go to the VPC Dashboard in AWS Console.

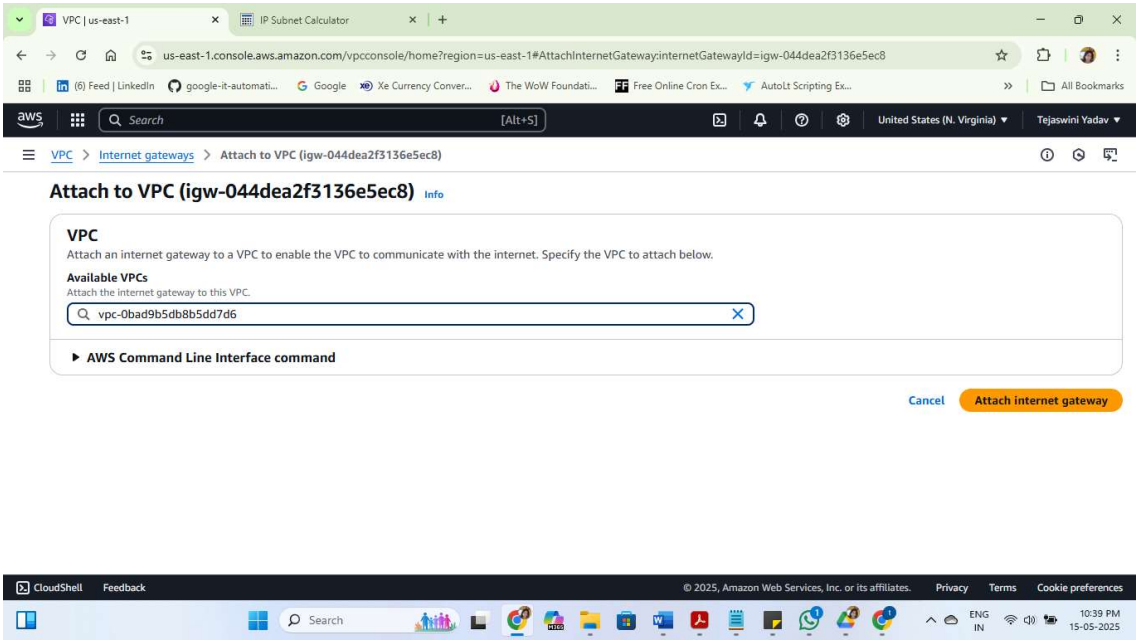
- **Go to Subnet**
- **Select VPC: AWS-VPN-Hybrid**
- **Subnet 1: aws-vpn-public-subnet**
- **AZs: us-east-1a**
- **IPv4 CIDR: 10.0.1.0/24**
- **Subnet 2 aws-vpn-public-subnet**
- **AZs: us-east-1b**
- **IPv4 CIDR: 10.0.2.0/24**
- **Select Remaining Default**
- **Create Subnet**



Go to the VPC Dashboard in AWS Console.

- **Go to Internet gateway**
- **Create Internet gateway**
- **Name: AWS-VPN-Hybrid-igw**
- **Click create**
- **Go to actions attach to VPC**
- **Select the VPC and attach Internet gateway**

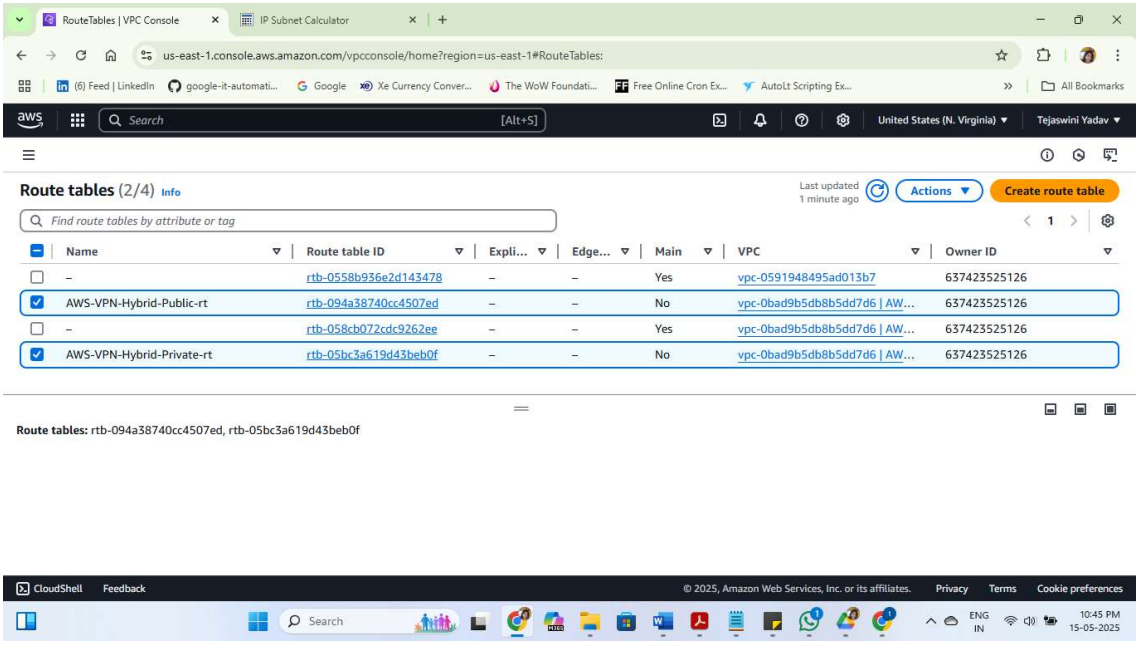




Go to the VPC Dashboard in AWS Console.

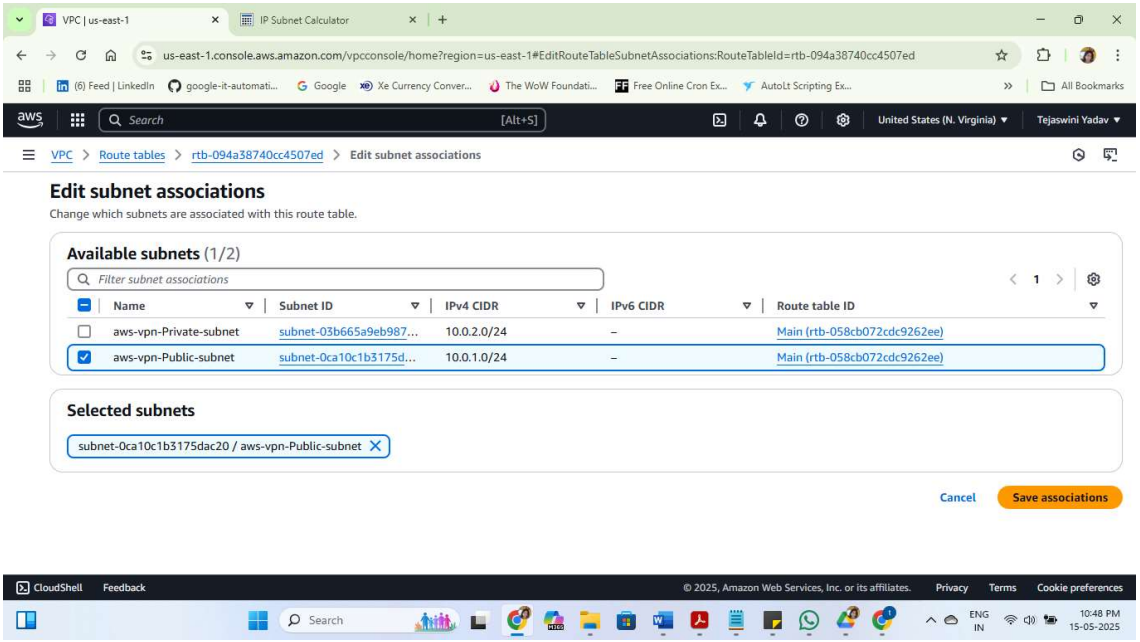
- **Go to Route Tables**
- **RT 1: AWS-VPN-Hybrid-Public-rt**
- **Select VPC**
- **Create route table**

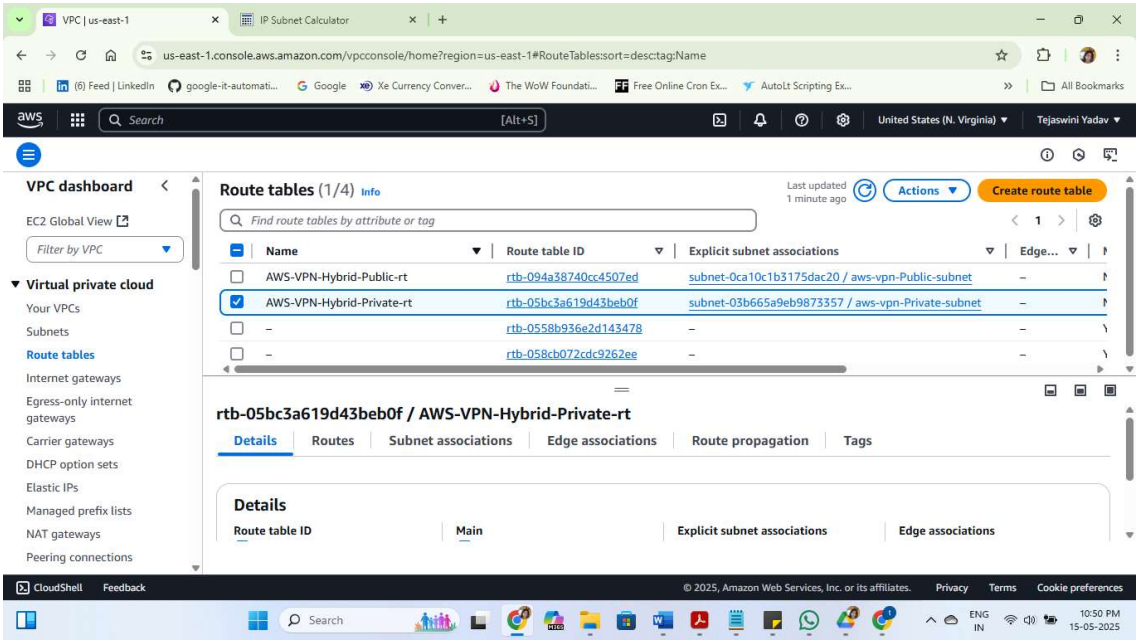
- **RT 2: AWS-VPN-Hybrid-Private-rt**
- **Select VPC**
- **Create route table**



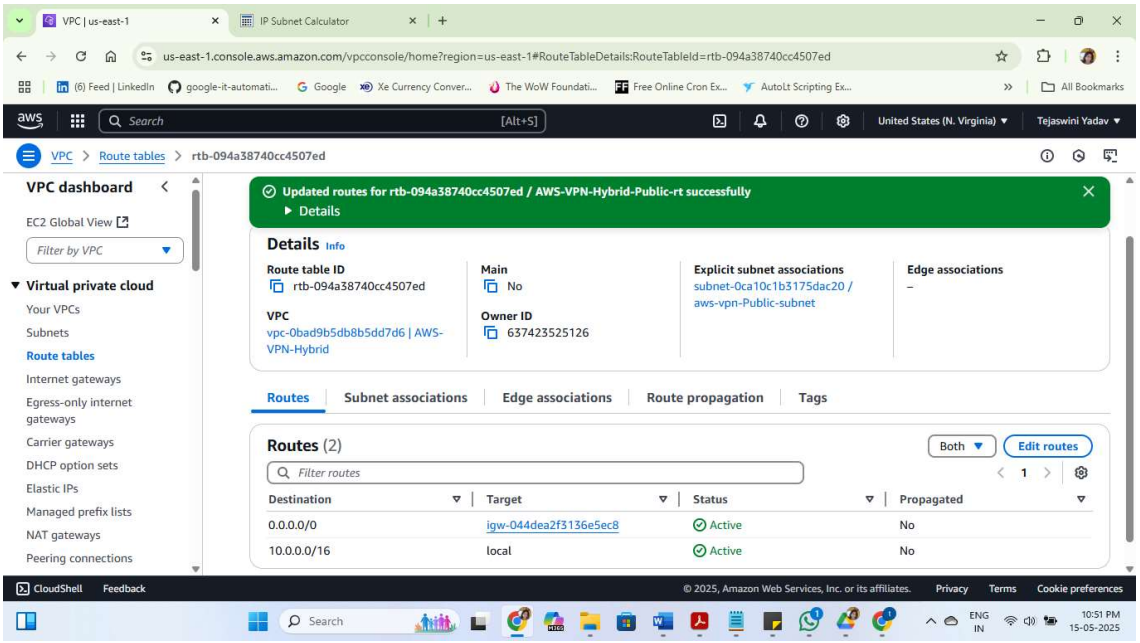
Go to public route table associate only public subnet

- **Click on Subnet Associations**
- **Edit Subnet associations**
- **Select Public Subnet**
- **Save associations**
- **Go to private Route Table**
- **Click on Subnet Associations**
- **Edit Subnet associations**
- **Select Private Subnet**
- **Save associations**


VPC | us-east-1
IP Subnet Calculator
us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#EditRouteTableSubnetAssociations:RouteTableId=rtb-094a38740cc4507ed
Search
[Alt+S]
United States (N. Virginia)
Tejaswini Yadav
VPC > Route tables > rtb-094a38740cc4507ed > Edit subnet associations
Edit subnet associations
Change which subnets are associated with this route table.
Available subnets (1/2)
Filter subnet associations
Name
Subnet ID
IPv4 CIDR
IPv6 CIDR
Route table ID
aws-vpn-Private-subnet
subnet-03b665a9eb987...
10.0.2.0/24
Main (rtb-058cb072cdc9262ee)
aws-vpn-Public-subnet
subnet-0ca10c1b3175d...
10.0.1.0/24
Main (rtb-058cb072cdc9262ee)
Selected subnets
subnet-0ca10c1b3175dac20 / aws-vpn-Public-subnet
Cancel
Save associations
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


VPC | us-east-1
IP Subnet Calculator
us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTables:sort=desc:tag:Name
Search
[Alt+S]
United States (N. Virginia)
Tejaswini Yadav
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Route tables (1/4) Info
Last updated 1 minute ago
Actions
Create route table
Find route tables by attribute or tag
Name
Route table ID
Explicit subnet associations
Edge...
AWS-VPN-Hybrid-Public-rt
rtb-094a38740cc4507ed
subnet-0ca10c1b3175dac20 / aws-vpn-Public-subnet
AWS-VPN-Hybrid-Private-rt
rtb-05bc3a619d43beb0f
subnet-03b665a9eb9873357 / aws-vpn-Private-subnet
rtb-0558b936e2d143478
rtb-058cb072cdc9262ee
rtb-05bc3a619d43beb0f / AWS-VPN-Hybrid-Private-rt
Details
Routes
Subnet associations
Edge associations
Route propagation
Tags
Details
Route table ID
Main
Explicit subnet associations
Edge associations
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

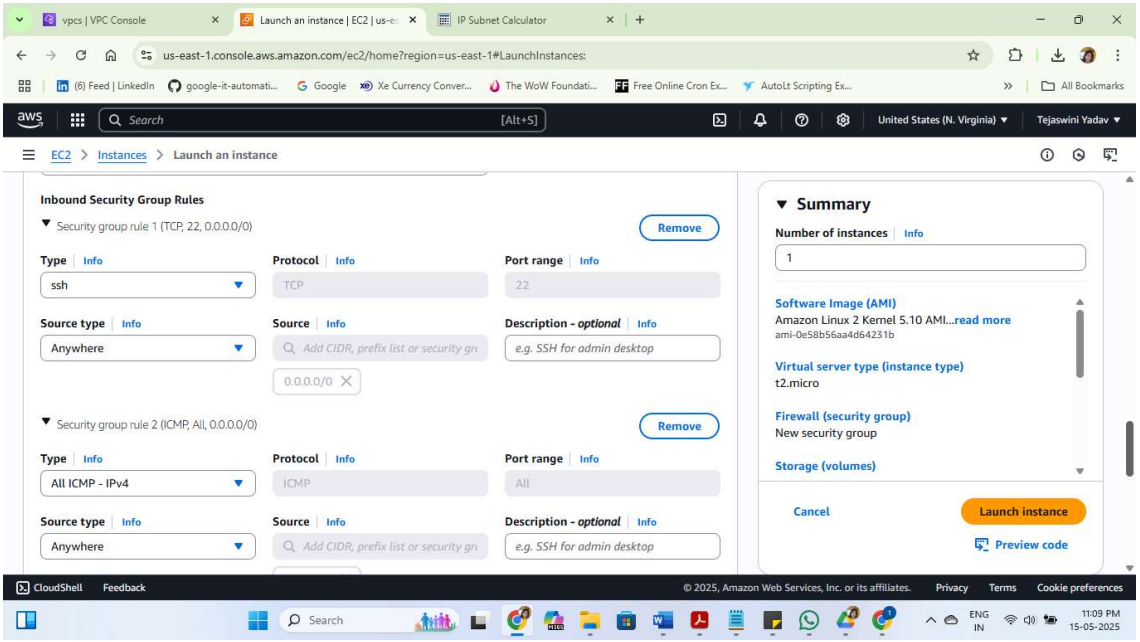
- **Go to Route tables**
- **Go to Public Route Table**
- **Click on edit routes**
- **Add routes 0.0.0.0.0/0**
- **Select the Internet gateway**
- **Click Save changes**



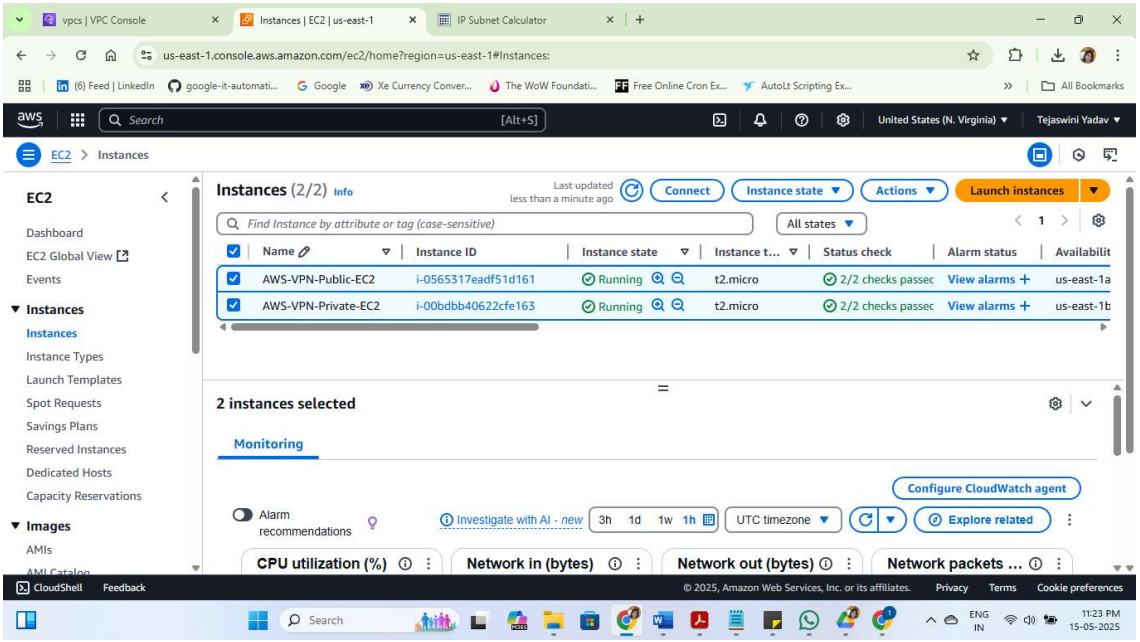
Create Two EC2 Instance:

- **EC2 1 Name: AWS-VPN-Private-EC2**
- **Amazon Linux Machine**
- **Instatance type: t2 micro**
- **Key pair: aws-key.pem**
- **Select VPC**
- **Select Subnet Private**
- **Create security group: All ICMP – IPv4 , Anywhere**
- **Launch Instance**

- **EC2 1 Name: AWS-VPN-Public-EC2**
- **Amazon Linux Machine**
- **Instatance type: t2 micro**
- **Key pair: aws-key.pem**
- **Select VPC**
- **Select Subnet Public**
- **Enable Public IP**
- **Create security group: All ICMP – IPv4 , Anywhere**
- **Launch Instance**



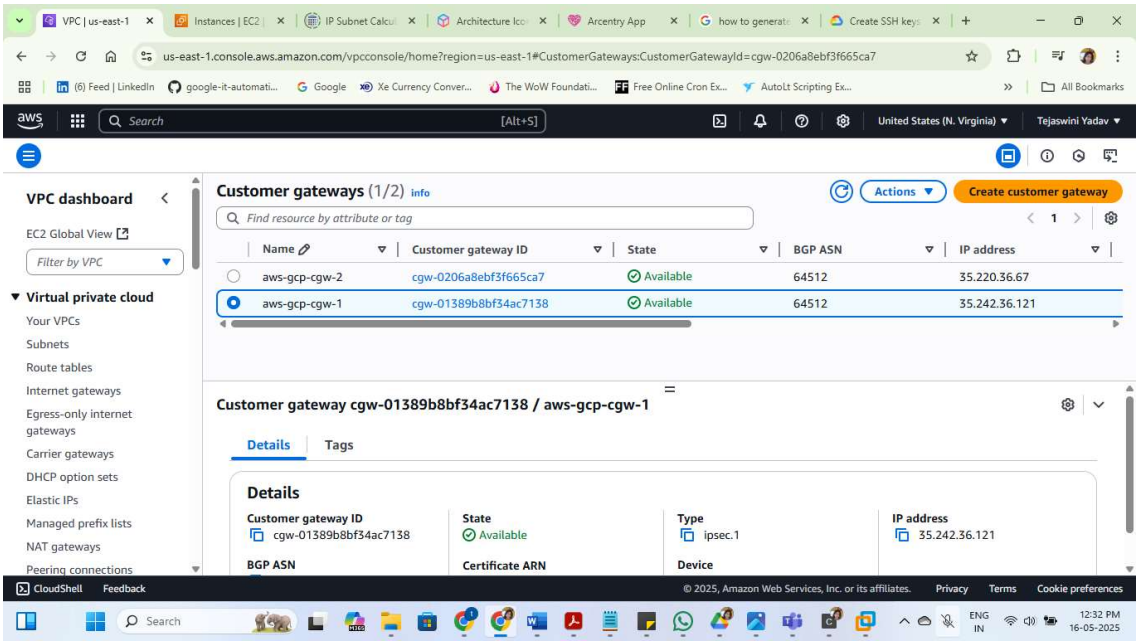
Both Private and Public EC2 Instance Created:



customer gateway

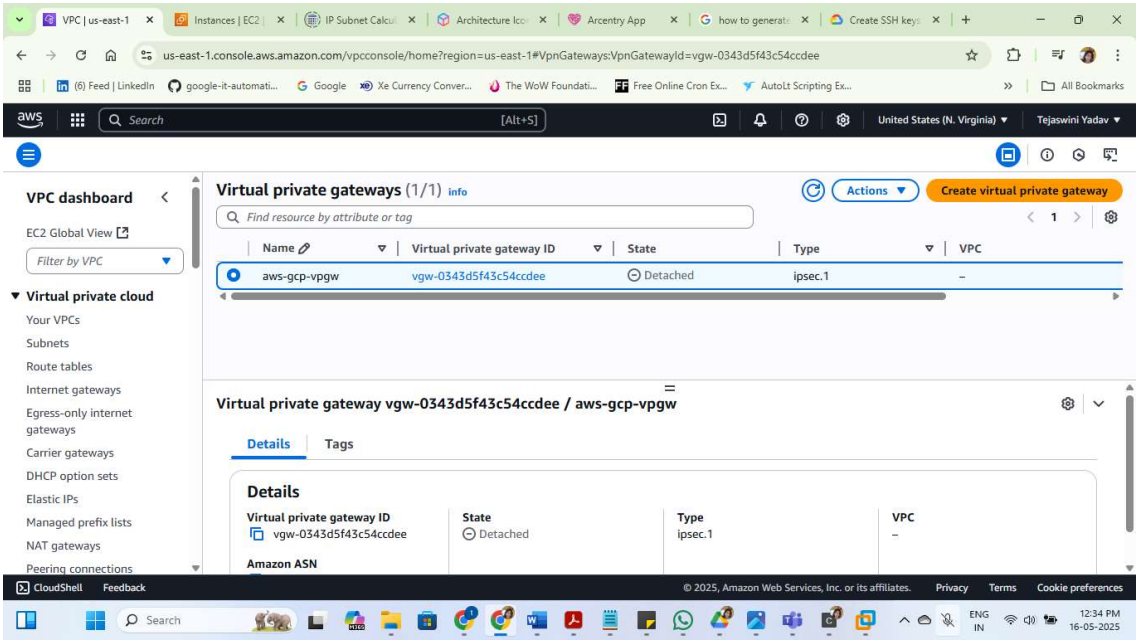
- **Name: aws-gcp-cgw-1**
- **BGP ASN: 64512**
- **IP address: 35.242.36.121**
- **Select default**
- **Create customer gateway**

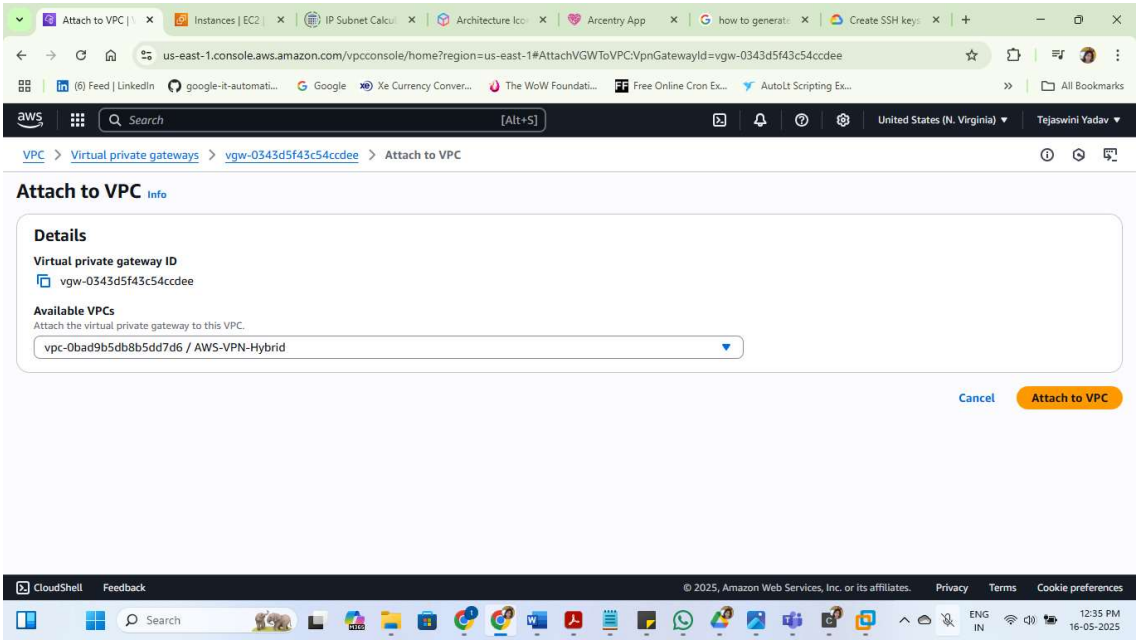
- **Name: aws-gcp-cgw-2**
- **BGP ASN: 64512**
- **IP address: 35.220.36.67**
- **Select default**
- **Create customer gateway**

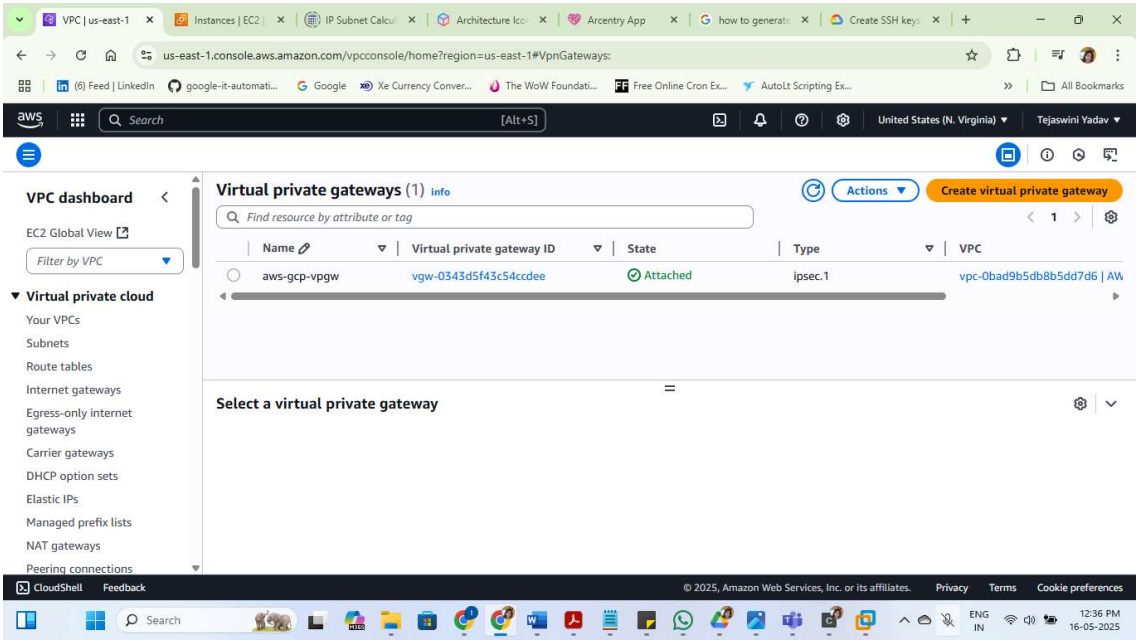


CREATE VIRTUAL PRIVATE GATEWAY

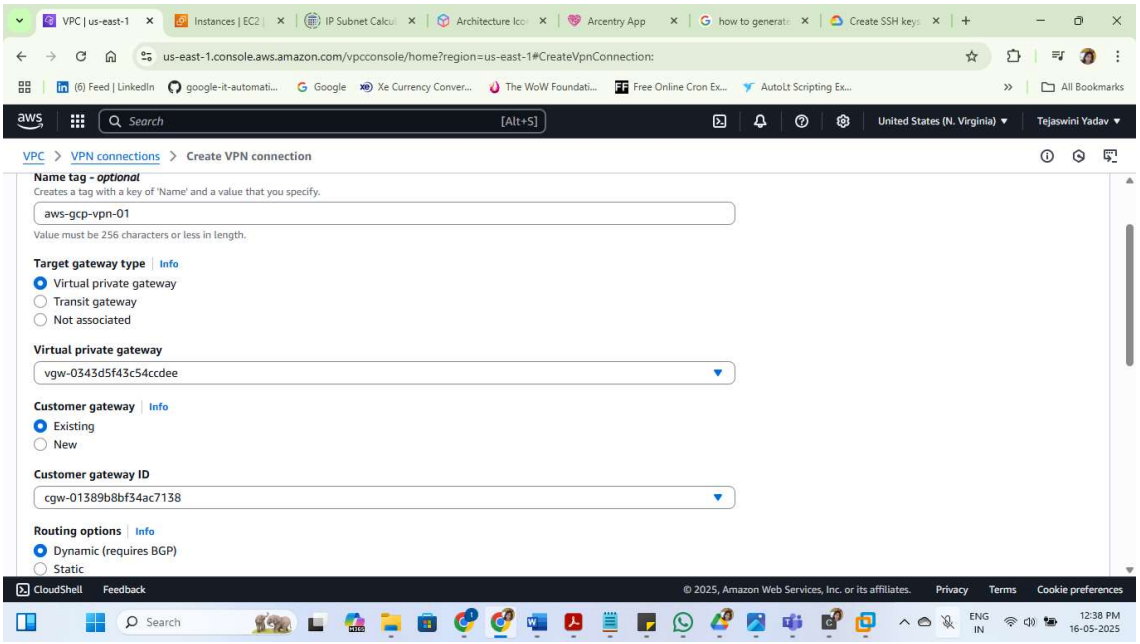
- **Name: aws-gcp-vpgw**
- **Amazon ASN : 64513**
- **Create Virtual Private gateway**
- **Attach to VPC**

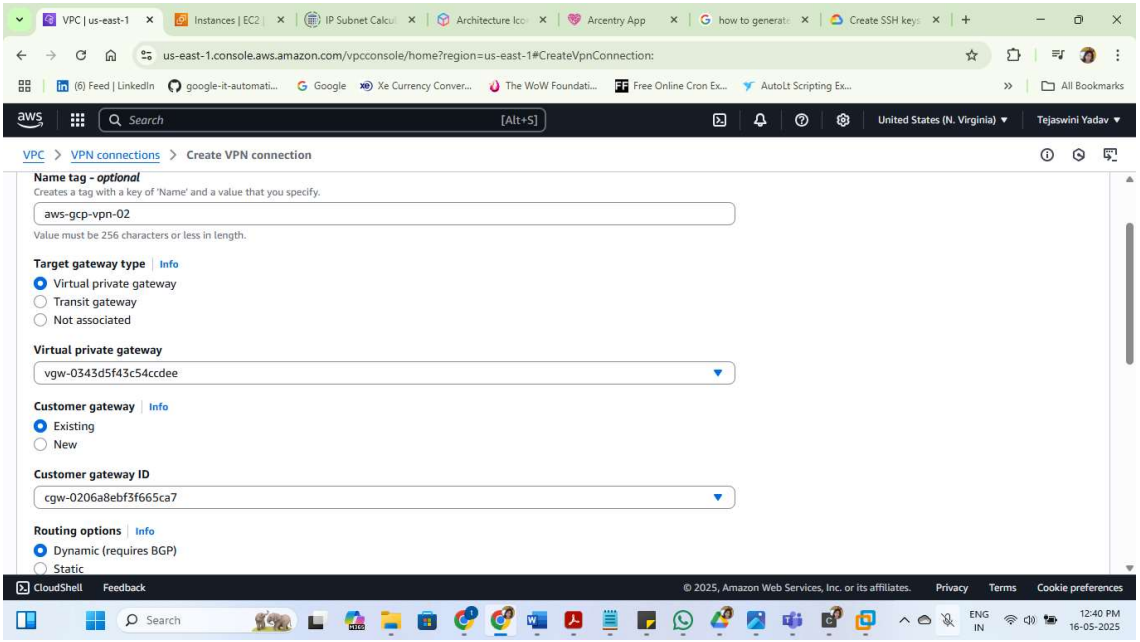

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#VpnGateways:VpnGatewayId=vgw-0343d5f43c54ccdee
VPC dashboard
EC2 Global View
Filter by VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
NAT gateways
Peering connections
Virtual private gateways (1/1) info
Actions
Create virtual private gateway
Find resource by attribute or tag
Name
Virtual private gateway ID
State
Type
VPC
aws-gcp-vpgw
vgw-0343d5f43c54ccdee
Detached
ipsec.1
Virtual private gateway vgw-0343d5f43c54ccdee / aws-gcp-vpgw
Details
Tags
Virtual private gateway ID
vgw-0343d5f43c54ccdee
State
Detached
Type
ipsec.1
VPC
Amazon ASN
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#AttachVGWToVPC:VpnGatewayId=vgw-0343d5f43c54ccdee
VPC > Virtual private gateways > vgw-0343d5f43c54ccdee > Attach to VPC
Attach to VPC Info
Details
Virtual private gateway ID
vgw-0343d5f43c54ccdee
Available VPCs
Attach the virtual private gateway to this VPC.
vpc-0bad9b5db8b5dd7d6 / AWS-VPN-Hybrid
Cancel
Attach to VPC
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

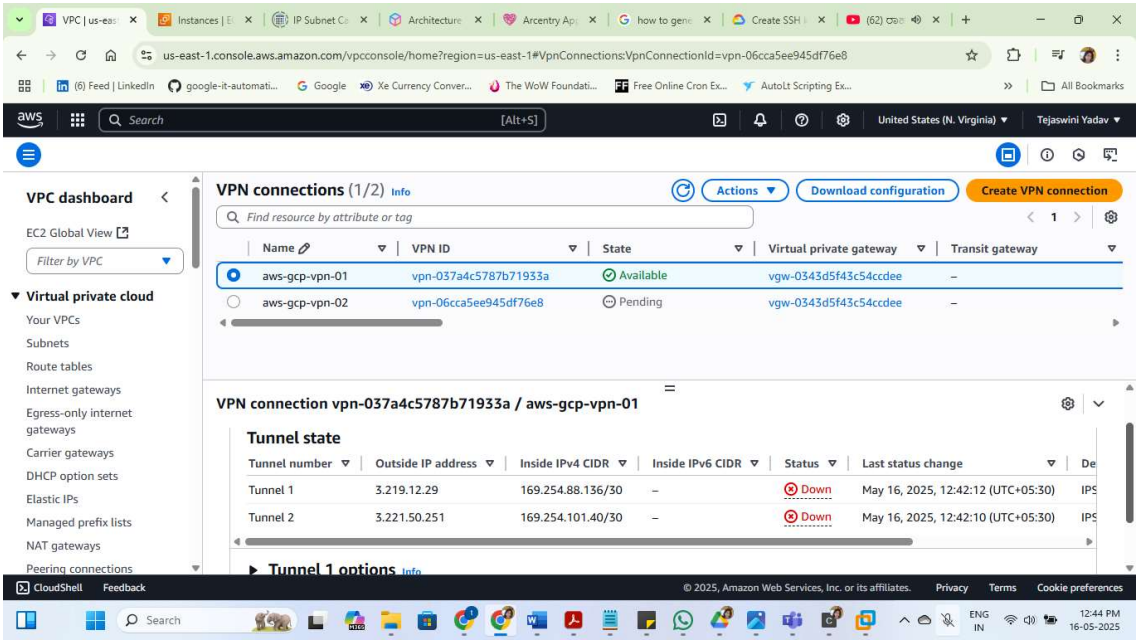


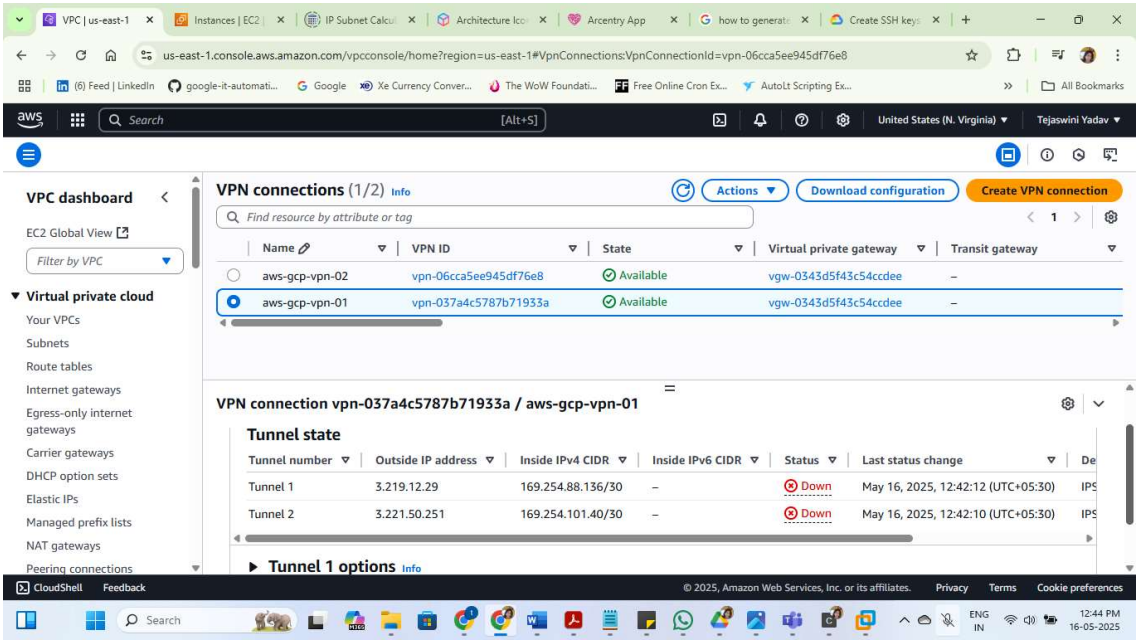
- **Create VPN Connection**
- **Select site -to-site-connections**
- **Create two VPN Connections and meanwhile attached to VPN gateway**



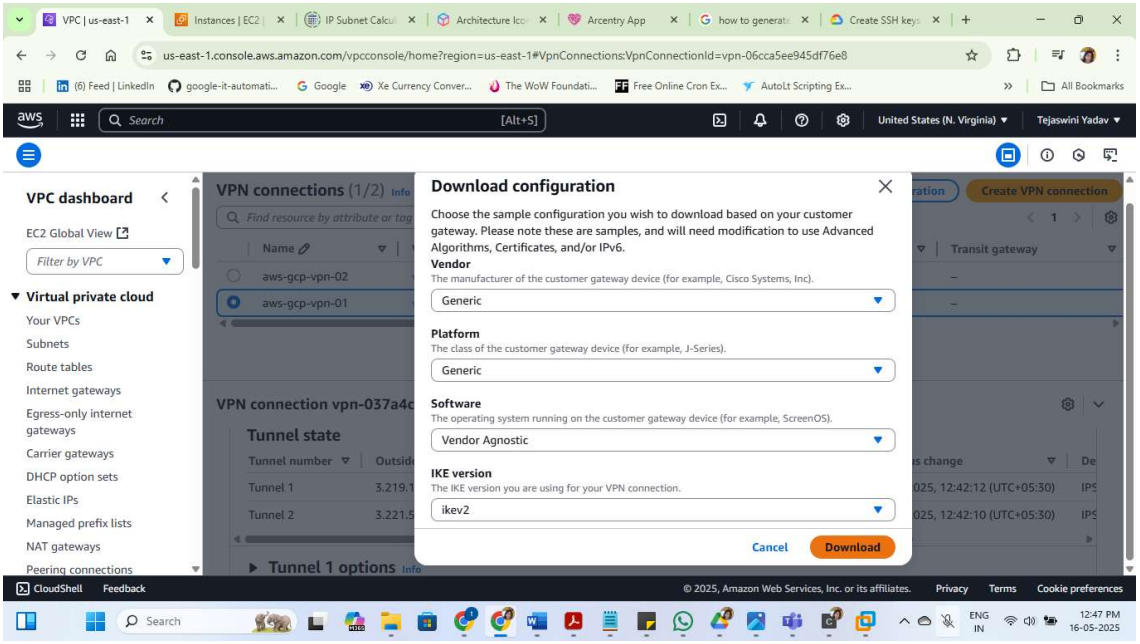


Here two tunnels are Down in Status need to set up tunnel connectivity:

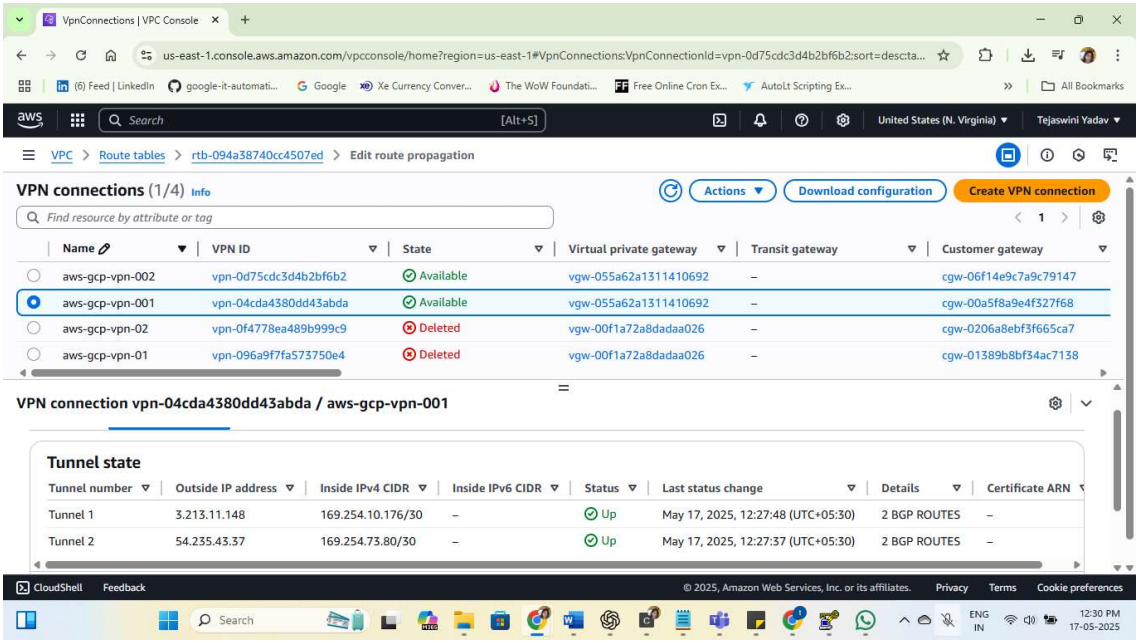




Select the VPN Connection click on Downloads configuration make the changes as below and download from two VPN connections



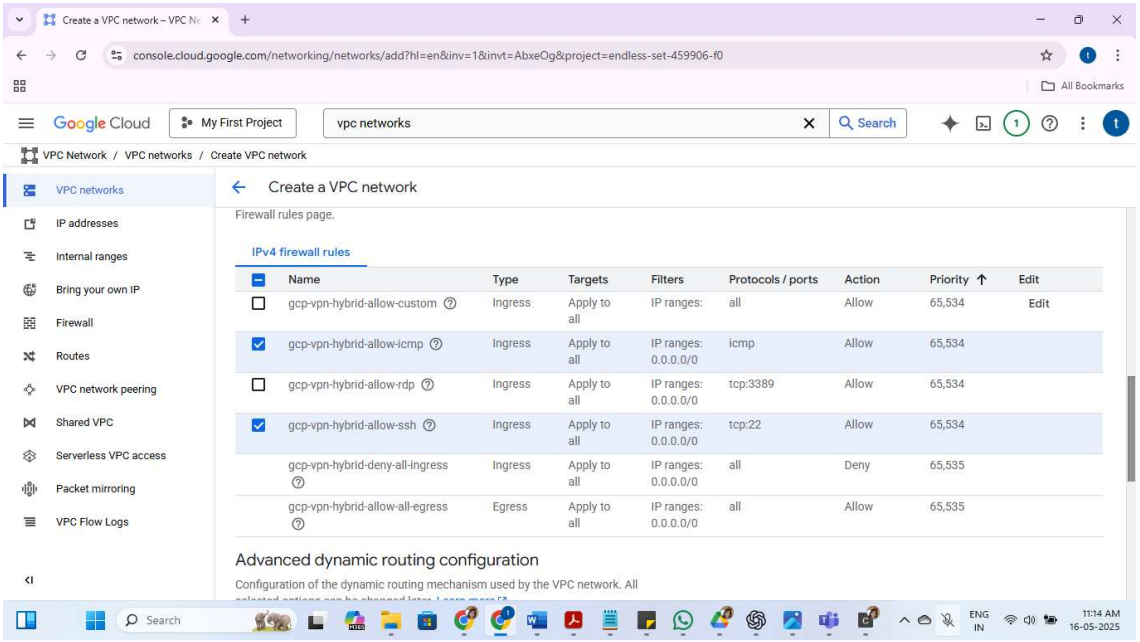
Output Result:

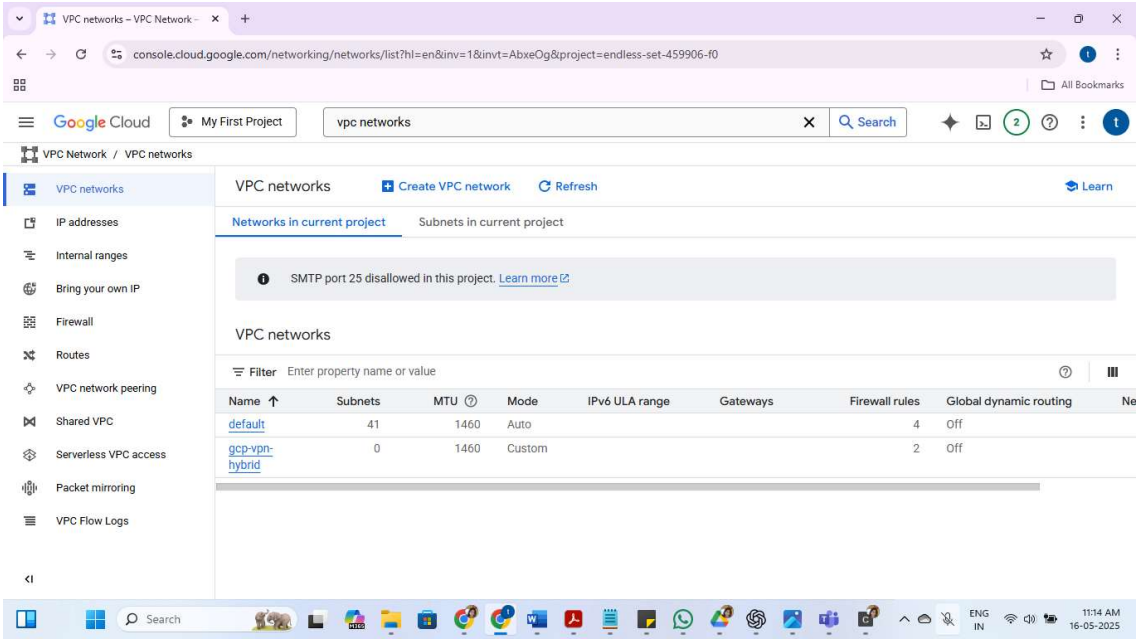


STEP 1: Create a VPC in GCP

Go to the VPC Dashboard in GCP Console.

- **VPC Name: gcp-vpc-hybrid**
- **Enable two rules: Allow icmp and allow ssh**
- **Dynamic Routing Mode: Regional**
- **Best Path selection mode: Legacy**
- **Select Remaining Default**
- **Create VPC on GCP**

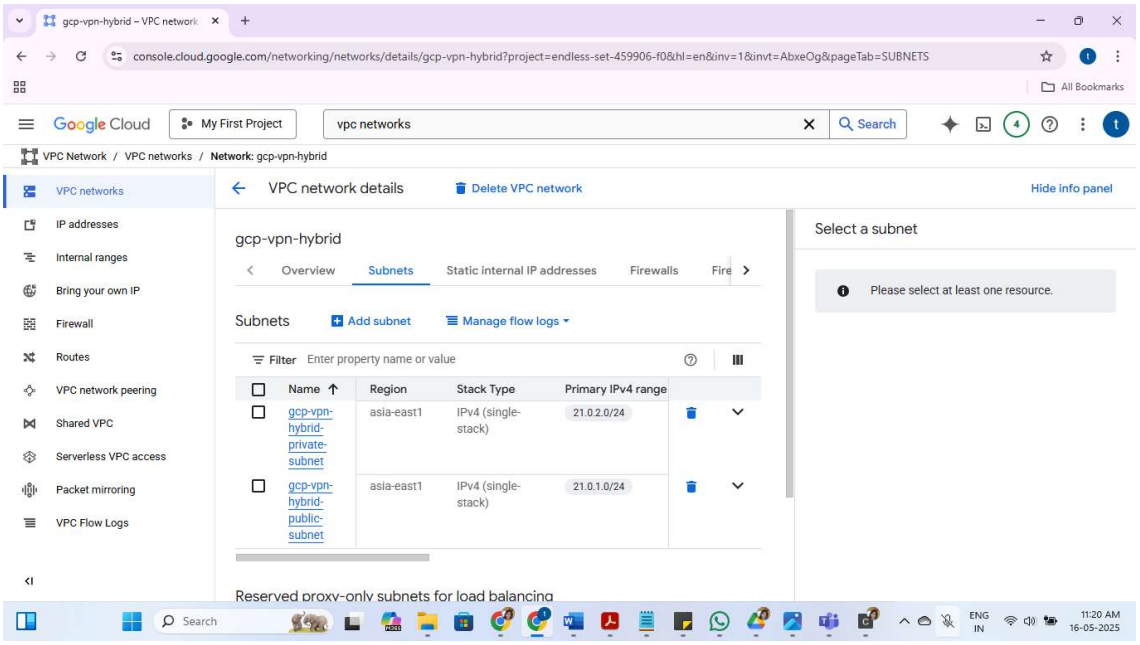




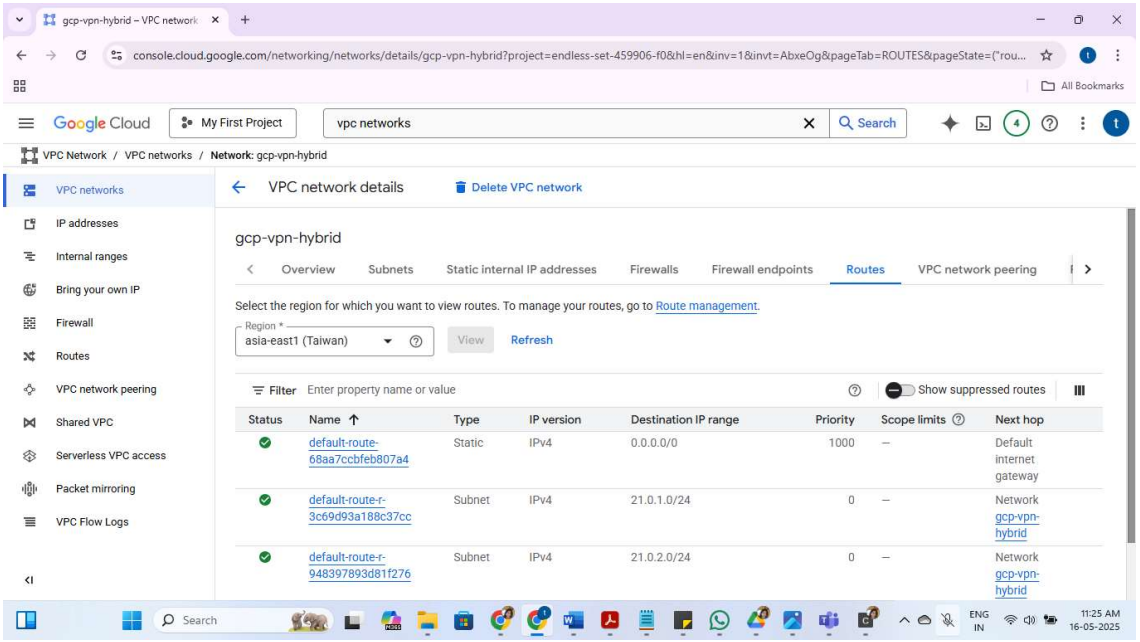
Go to the VPC Dashboard in GCP Console.

- **Go to Subnet**
- **Select VPC: gcp-vpn-hybrid**
- **Select option Subnet click on add Subnet**
- **Subnet 1: gcp-vpn-hybrid-public-subnet**
- **Region : asia-east1**
- **IPv4 CIDR: 21.0.1.0/24**
- **Click add Subnet**

- **Subnet 2: gcp-vpn-hybrid-private-subnet**
- **Region: asia-east1**
- **IPv4 CIDR: 21.0.2.0/24**
- **Click add Subnet**

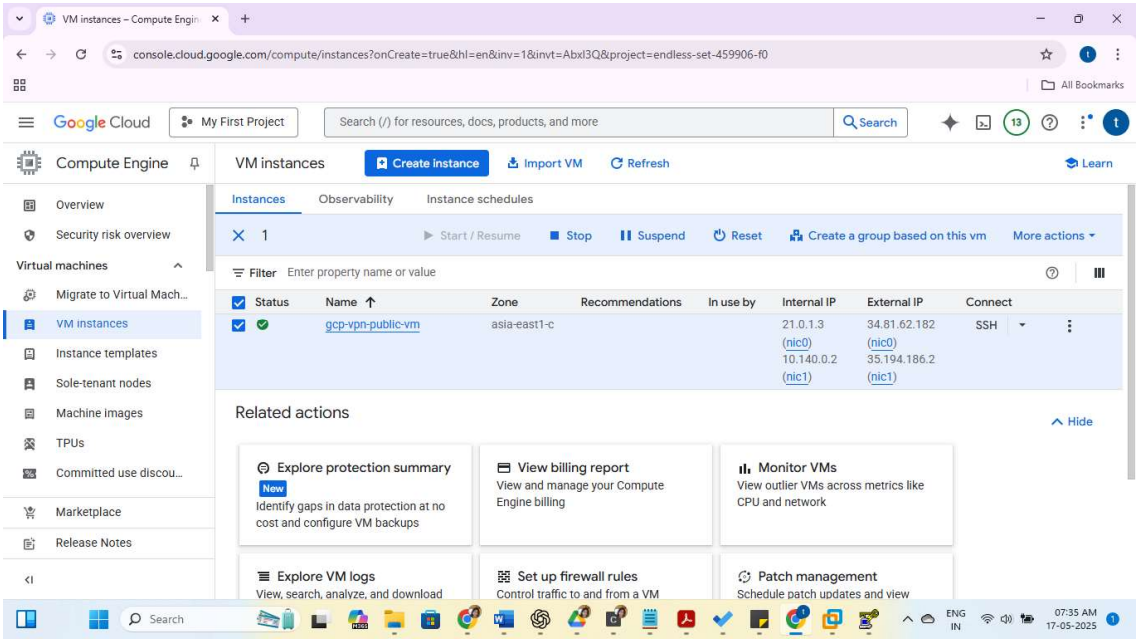


- **Click on Routes**
- **All the default routes available in GCP**



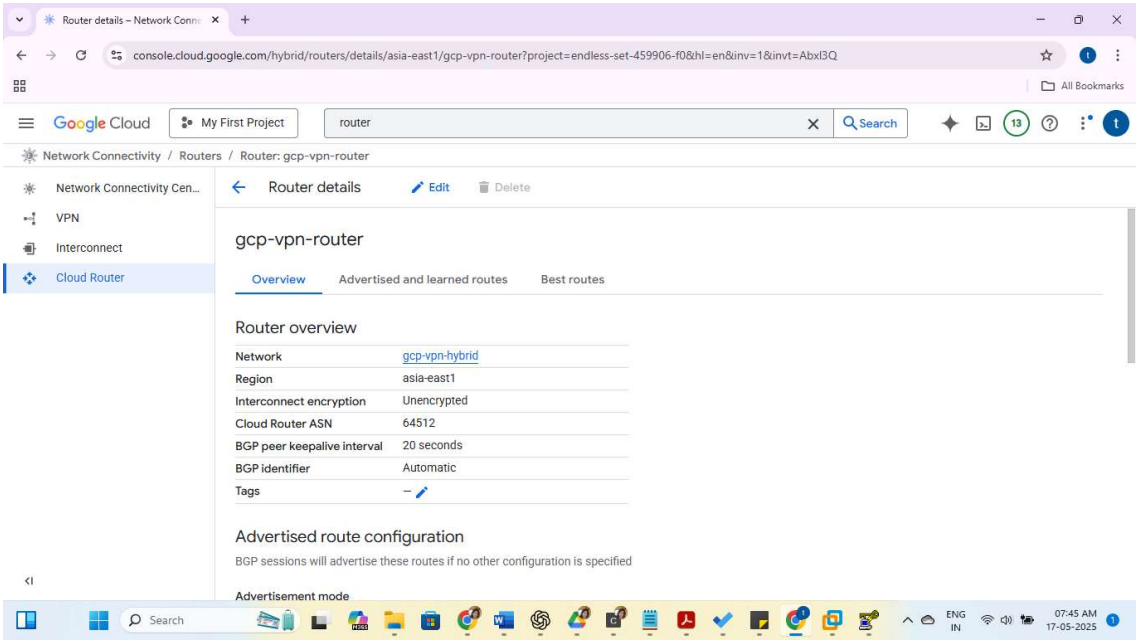
Create VM Instance in GCP Console

- **Name : gcp-vpn-public-vm**
- **Region : asia-east1 (Taiwan)**
- **Machine : E2**
- **Network interfaces : gcp-vpn-hybrid**
- **Subnetwork : gcp-vpn-hybrid-public-subnet IPv4 (21.0.1.0/24)**
- **Security: Add Public SSH key**

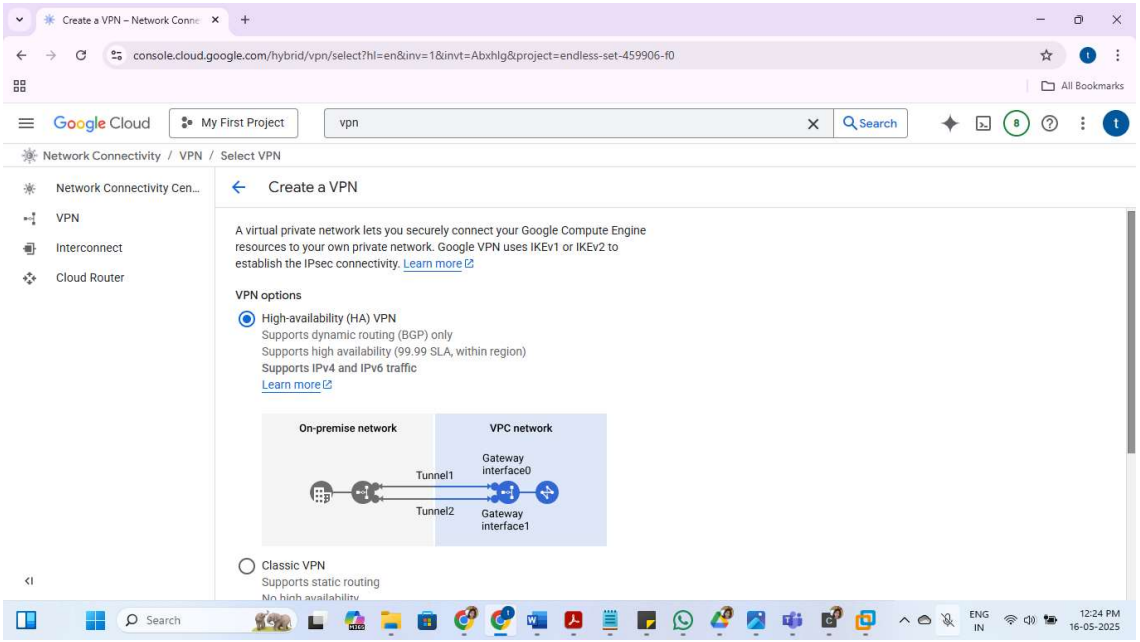


- **Create Router**

- **Name : gcp-vpn-router**
- **ASN Number: 64512**

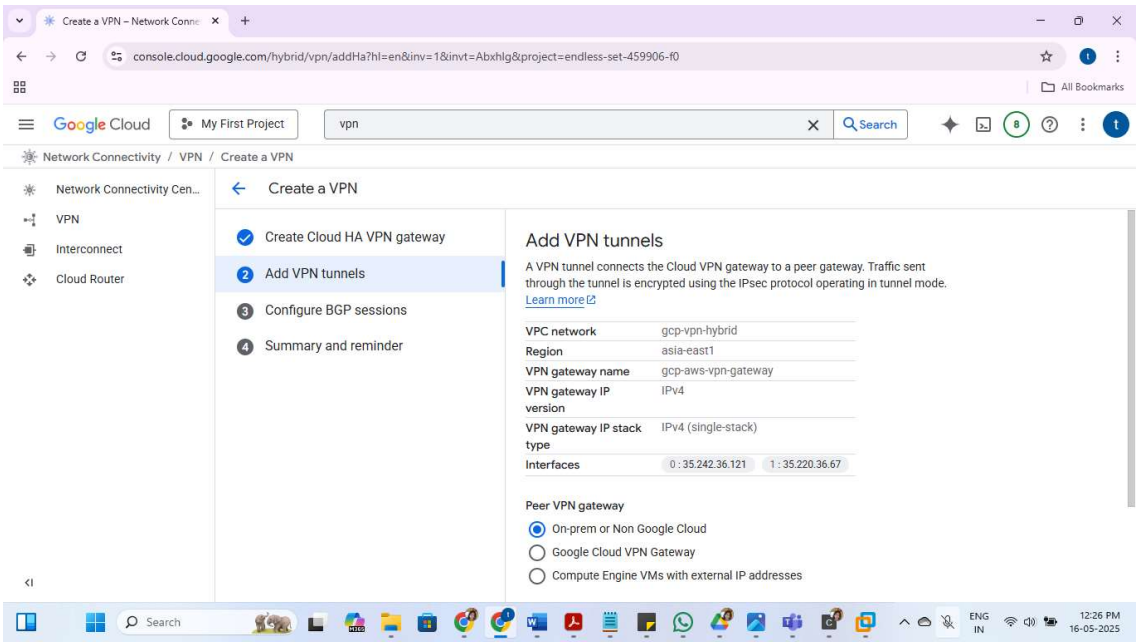


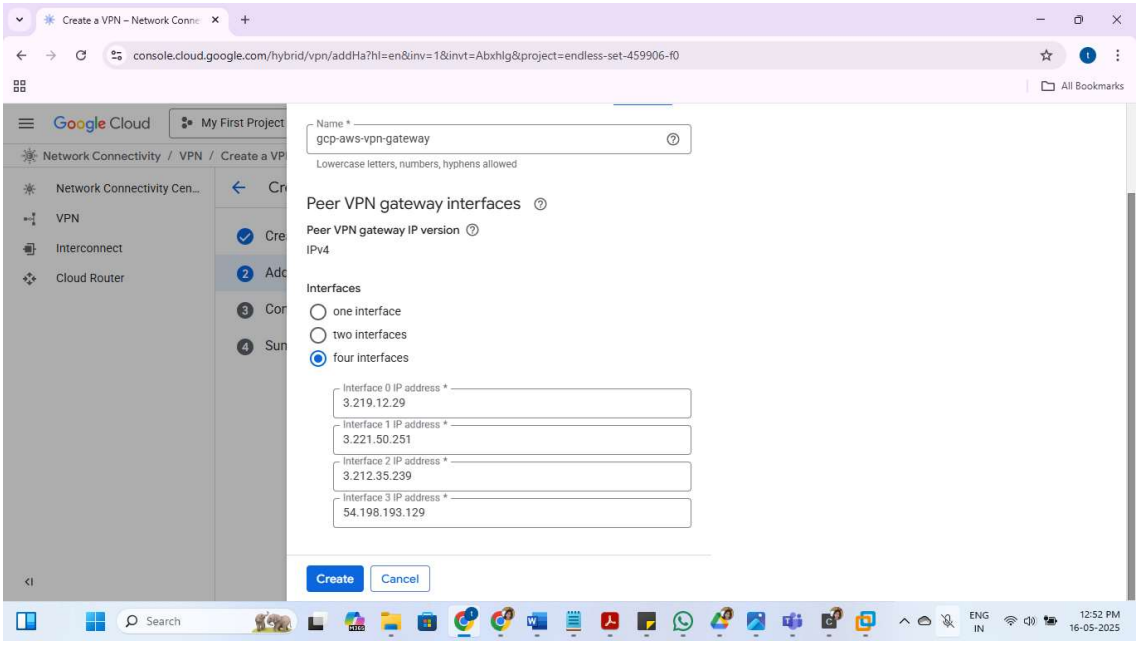
- **Create VPN**
- **Select High Availability VPN**
- **Click continue**
- **VPN Gateway: gcp-aws-vpn-gateway**
- **Select VPC network**

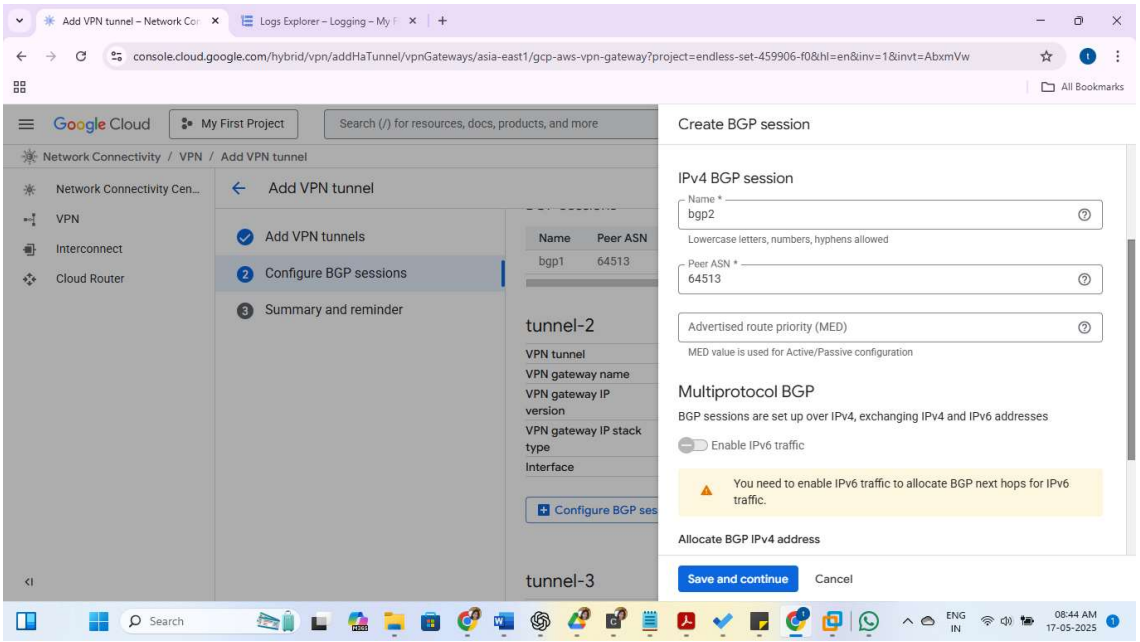


****Note down the IP address generate in further steps we need:**

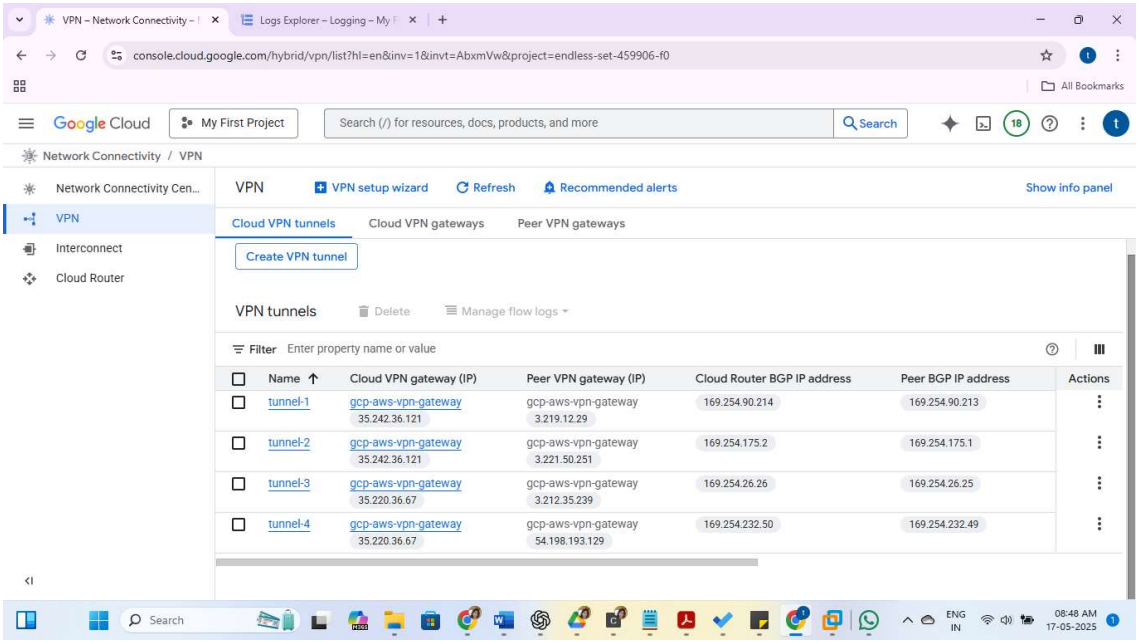
IPs: 0 : 35.242.36.121 1 : 35.220.36.67

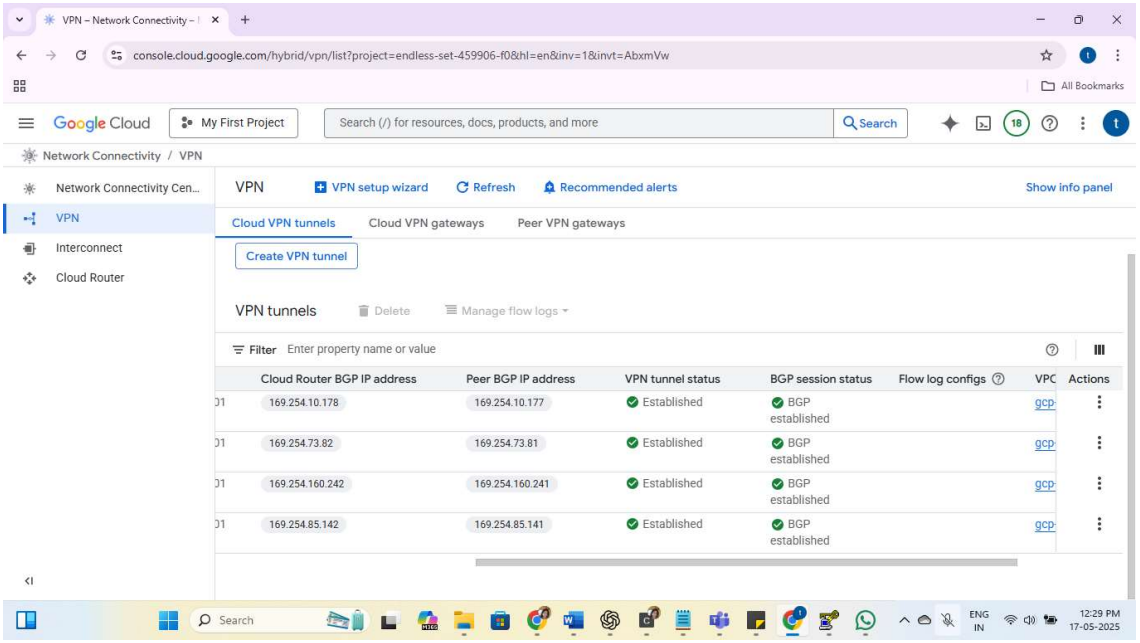


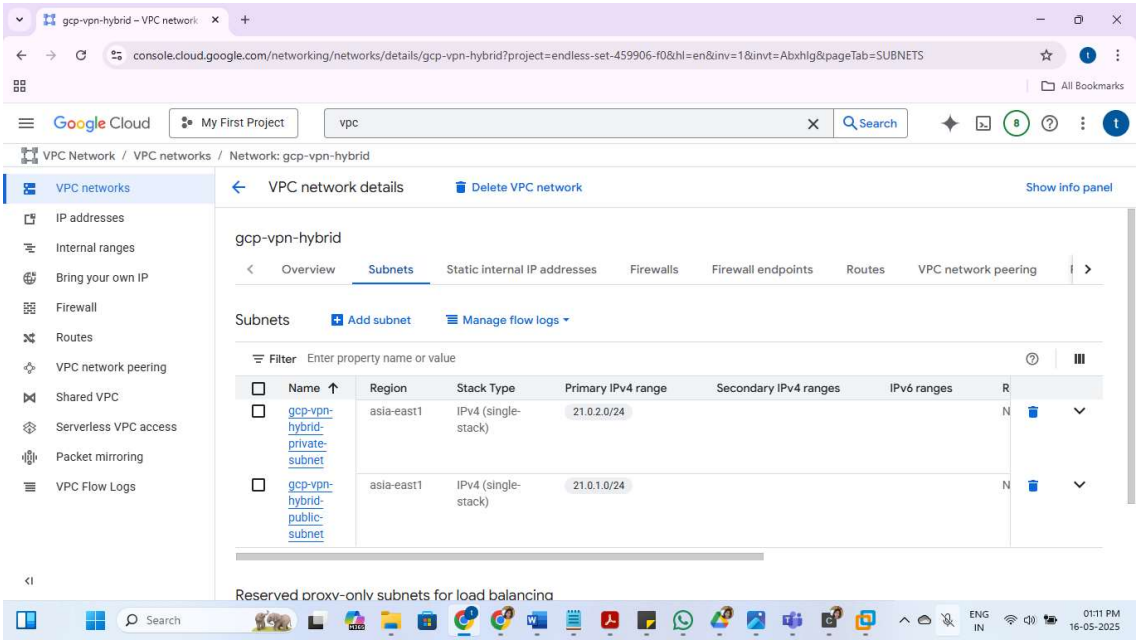

Create a VPN – Network Conne
console.cloud.google.com/hybrid/vpn/addHa?hl=en&inv=1&invt=Abxhlg&project=endless-set-459906-f0
Google Cloud
My First Project
Network Connectivity / VPN / Create a VP
Network Connectivity Cen...
VPN
Interconnect
Cloud Router
Name *
gcp-aws-vpn-gateway
Lowercase letters, numbers, hyphens allowed
Peer VPN gateway interfaces
Peer VPN gateway IP version
IPv4
Interfaces
one interface
two interfaces
four interfaces
Interface 0 IP address *
3.219.12.29
Interface 1 IP address *
3.221.50.251
Interface 2 IP address *
3.212.35.239
Interface 3 IP address *
54.198.193.129
Create
Cancel
Search
ENG
IN
12:52 PM
16-05-2025

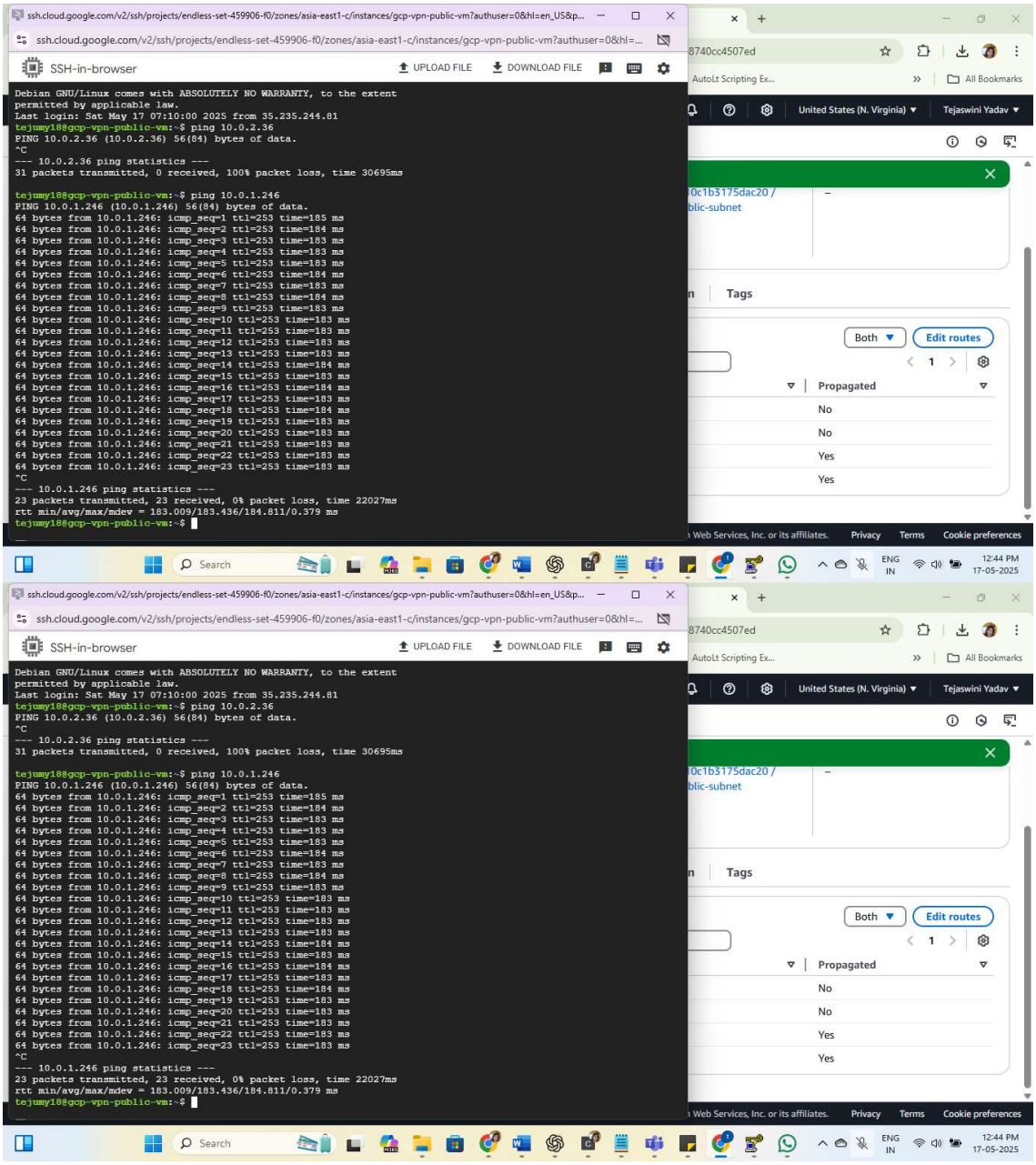

Add VPN tunnel – Network Cor
Logs Explorer – Logging – My F
console.cloud.google.com/hybrid/vpn/addHaTunnel/vpnGateways/asia-east1/gcp-aws-vpn-gateway?project=endless-set-459906-f0&hl=en&inv=1&invt=AbxmVw
Google Cloud
My First Project
Search (/) for resources, docs, products, and more
Network Connectivity / VPN / Add VPN tunnel
Network Connectivity Cen...
VPN
Interconnect
Cloud Router
Add VPN tunnel
Add VPN tunnels
Configure BGP sessions
Summary and reminder
Name
Peer ASN
bgp1
64513
tunnel-2
VPN tunnel
VPN gateway name
VPN gateway IP version
VPN gateway IP stack type
Interface
Configure BGP ses
tunnel-3
Create BGP session
IPv4 BGP session
Name *
bgp2
Lowercase letters, numbers, hyphens allowed
Peer ASN *
64513
Advertised route priority (MED)
MED value is used for Active/Passive configuration
Multiprotocol BGP
BGP sessions are set up over IPv4, exchanging IPv4 and IPv6 addresses
Enable IPv6 traffic
You need to enable IPv6 traffic to allocate BGP next hops for IPv6 traffic.
Allocate BGP IPv4 address
Save and continue
Cancel
Search
ENG
IN
08:44 AM
17-05-2025

After few the all-tunnel start connected AWS cloud






gcp-vpn-hybrid – VPC network
console.cloud.google.com/networking/networks/details/gcp-vpn-hybrid?project=endless-set-459906-f0&hl=en&inv=1&invt=AbxhIg&pageTab=SUBNETS
Google Cloud
My First Project
vpc
Search
VPC Network / VPC networks / Network: gcp-vpn-hybrid
VPC networks
IP addresses
Internal ranges
Bring your own IP
Firewall
Routes
VPC network peering
Shared VPC
Serverless VPC access
Packet mirroring
VPC Flow Logs
VPC network details
Delete VPC network
Show info panel
gcp-vpn-hybrid
Overview
Subnets
Static internal IP addresses
Firewalls
Firewall endpoints
Routes
VPC network peering
Subnets
Add subnet
Manage flow logs
Filter
Enter property name or value
Name
Region
Stack Type
Primary IPv4 range
Secondary IPv4 ranges
IPv6 ranges
gcp-vpn-hybrid-private-subnet
asia-east1
IPv4 (single-stack)
21.0.2.0/24
gcp-vpn-hybrid-public-subnet
asia-east1
IPv4 (single-stack)
21.0.1.0/24
Reserved proxy-only subnets for load balancing
Search
ENG
IN
01:11 PM
16-05-2025


ssh.cloud.google.com/v2/ssh/projects/endless-set-459906-f0/zones/asia-east1-c/instances/gcp-vpn-public-vm?authuser=0&hl=en_US&p...
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat May 17 07:10:00 2025 from 35.235.244.81
tejumy18@gcp-vpn-public-vm:~$ ping 10.0.2.36
PING 10.0.2.36 (10.0.2.36) 56(84) bytes of data.
^C
--- 10.0.2.36 ping statistics ---
31 packets transmitted, 0 received, 100% packet loss, time 30695ms
tejumy18@gcp-vpn-public-vm:~$ ping 10.0.1.246
PING 10.0.1.246 (10.0.1.246) 56(84) bytes of data.
64 bytes from 10.0.1.246: icmp_seq=1 ttl=253 time=185 ms
64 bytes from 10.0.1.246: icmp_seq=2 ttl=253 time=184 ms
64 bytes from 10.0.1.246: icmp_seq=3 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=4 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=5 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=6 ttl=253 time=184 ms
64 bytes from 10.0.1.246: icmp_seq=7 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=8 ttl=253 time=184 ms
64 bytes from 10.0.1.246: icmp_seq=9 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=10 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=11 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=12 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=13 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=14 ttl=253 time=184 ms
64 bytes from 10.0.1.246: icmp_seq=15 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=16 ttl=253 time=184 ms
64 bytes from 10.0.1.246: icmp_seq=17 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=18 ttl=253 time=184 ms
64 bytes from 10.0.1.246: icmp_seq=19 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=20 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=21 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=22 ttl=253 time=183 ms
64 bytes from 10.0.1.246: icmp_seq=23 ttl=253 time=183 ms
^C
--- 10.0.1.246 ping statistics ---
23 packets transmitted, 23 received, 0% packet loss, time 22027ms
rtt min/avg/max/mdev = 183.009/183.436/184.811/0.379 ms
tejumy18@gcp-vpn-public-vm:~$
United States (N. Virginia)
Tejaswini Yadav
Tags
Both
Edit routes
Propagated
No
No
Yes
Yes
Privacy
Terms
Cookie preferences
Search
12:44 PM
17-05-2025

HAPPY LEARNING

Regards,
Tejaswini J

https://github.com/tejaswiniyadav/Site-to-Site-VPN-

THANK YOU