**VPC**

## **Amazon VPC**

Amazon VPC or Virtual Private Cloud is a service that allows us to create an isolated virtual network for our Amazon resources. A virtual network is a private network that is always hidden from the outside world, and you can perform certain operations that you don’t want to make public. Any user with their AWS account can host Amazon VPC. You can create, access, and manage Amazon VPC with the help of certain tools and services like the Amazon Web Service Management Console, Amazon CLI (Command Line Interface), Amazon SDK, and Query API.

* **Route Table:**

In AWS Virtual Private Cloud, route Tables are the set of rules, that are used to determine where the network traffic has to be directed. The route table specifies the destination (IP address) and target (where do want to send the traffic to that destination). The target can be an Internet gateway, NAT gateway, Virtual private gateway, VPC peering connection, etc

* **Subnet:**

 It is a portion of the network that shares a common address component. All devices whose addresses have the same prefix are in the same subnet. For example, all those devices whose IP address would start with 172.31.1 would be part of the same subnet. There are two types of subnets. **Private Subnet**where resources are not exposed to the outside world and **Public Subnet**where resources are exposed to the internet through Internet Gateway.

* **Security Groups:**

Security groups are a set of firewall rules that controls the traffic for your instance. In Amazon Firewall the only action that can be carried out is allowed. You cannot create a rule to deny. The destination is always the instance on which the service security group is running. You can have a single security group associated with multiple instances.

* **NAT Gateway:**

Network Address Translation (NAT) Gateway is used when higher bandwidth and availability with lesser administrative effort is required. NAT gateway always resides inside the public subnet of an Availability Zone. It updates the routing table of the private subnet such that it sends the traffic to the NAT gateway. Elastic IP must be attached to the NAT gateway while creating. It supports only TCP, UDP, and ICMP protocols.

* **VPC Peering:**

A VPC peering connection allows you to route traffic between two Virtual Private Cloud’s using IPv4 or IPv6 private addresses. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, or with a VPC in another AWS account. A VPC peering connection helps you to facilitate the transfer of data

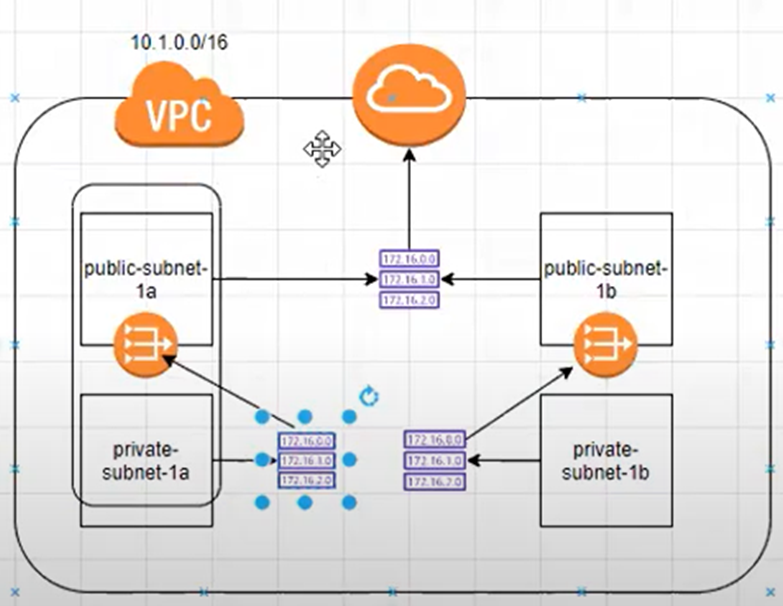
* **Network Access Control Lists (NACL):**

an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC. The default network ACL is configured to allow all traffic to flow in and out of the subnets to which it is associated.

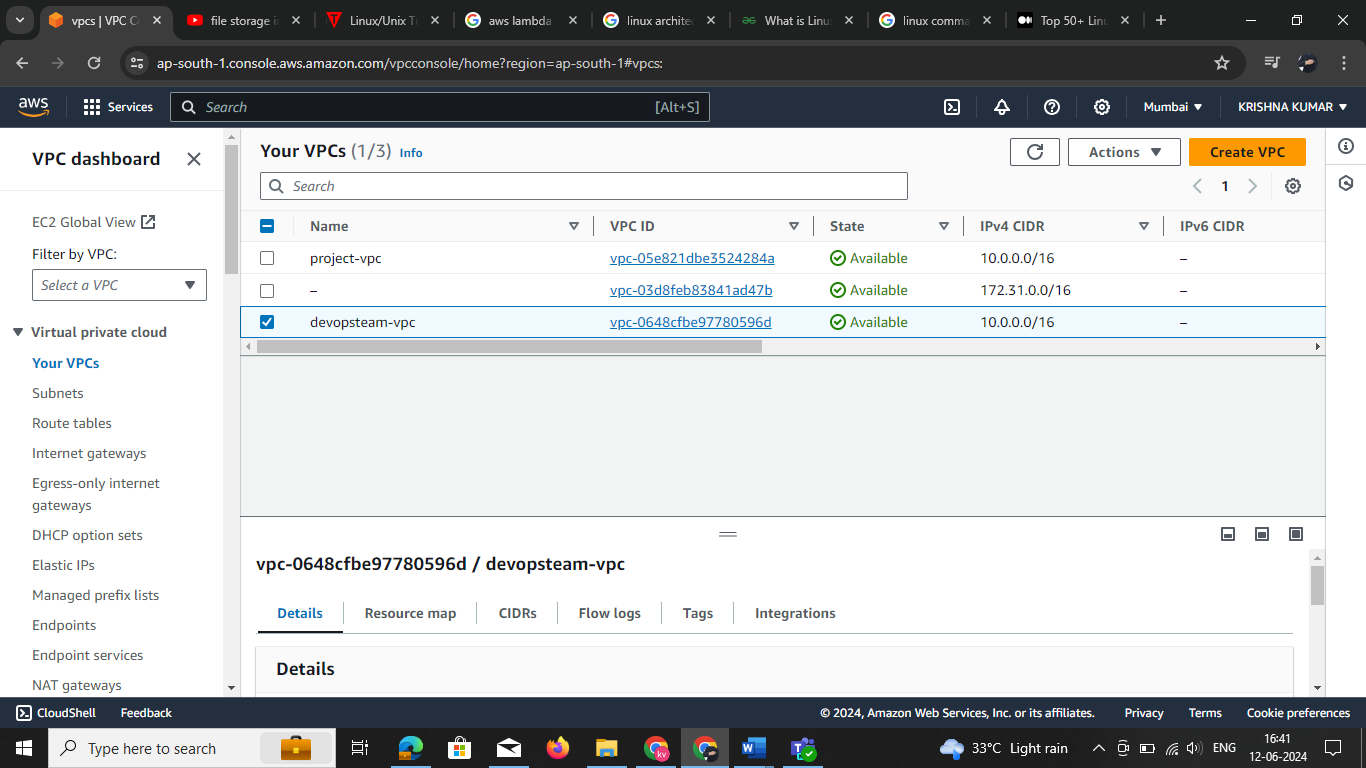
* **Virtual Private Gateway:**

A virtual private gateway is the VPN concentrator on the Amazon side of the VPN connection. You create a virtual private gateway and attach it to the VPC from which you want to create the VPN connection.

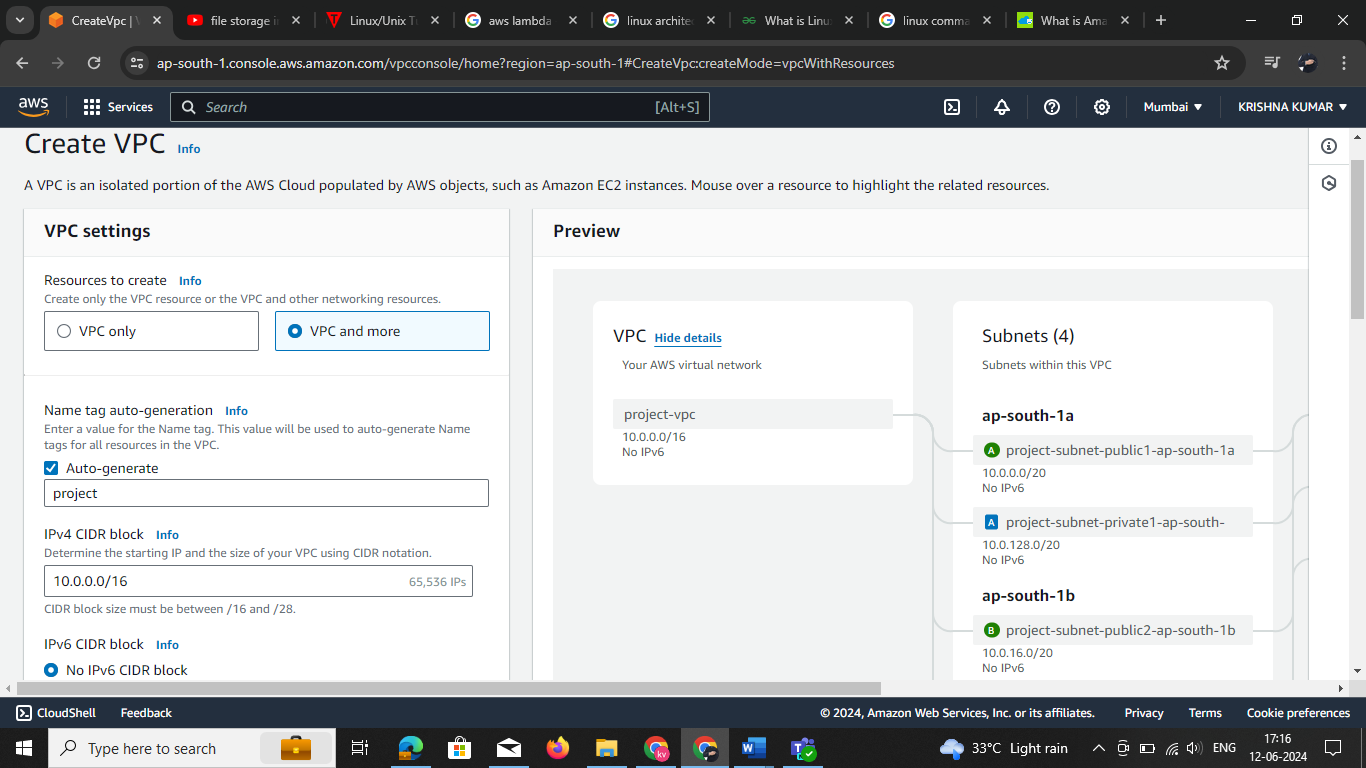
* **Customer Gateway:**  An Amazon VPC VPN connection links your data center (or network) to your Amazon VPC (virtual private cloud). A customer gateway is an anchor on your side of that connection. It can be a physical or software appliance.
* **Elastic IP:**It is a static IP address that never changes and is a reserved public IP address that can be assigned to any Instance in a particular region. An elastic IP is reserved for your AWS account and is yours until you release it.
* **Network Interface:**Network Interface is a point of connection between a public and a private network. Every instance has a default network interface, called the primary network interface. Network traffic is automatically shifted to the new instance if you move it from one instance to the other.
* **VPC Endpoints:**VPC endpoints allow private connection between your AWS VPC and other AWS services without using the internet. VPC endpoint devices are scaled, redundant, and highly available VPC components. There are two types of AWS Virtual Private Cloud endpoints **Interface endpoints**and**Gateway Endpoints.**

 **Creating VPC Architecture**

Step1 : **AWS console and click on create VPC**



Setp2 : select any one VPC are VPC more option



**Note: if you selecting🡪VPC only creating your Natgatway,subnets,routhtable,internet gateway you need crate all setup your owen subnet calalucate the CIDR values**

[**https://www.davidc.net/sites/default/subnets/subnets.html**](https://www.davidc.net/sites/default/subnets/subnets.html)

### IPv4 VPC Calculation

1. **CIDR Block**: Decide the CIDR block for your VPC. Common CIDR blocks are /16, /24, etc. This defines the range of IP addresses.
2. **Subnetting**:
   * Determine the number of subnets required.
   * Calculate the subnet masks and ranges.
3. **Host Calculation**:
   * Use the formula 2(32−subnet mask) −22^{(32 - \text{subnet mask})} - 22(32−subnet mask)−2 to calculate the number of usable IP addresses in a subnet. The -2 accounts for the network and broadcast addresses.

#### Example:

For a VPC with a CIDR block 192.168.0.0/16:

* Total IPs: 2(32−16) =655362^{(32 - 16)} = 655362(32−16)=65536
* Usable IPs: 65536−2=6553465536 - 2 = 6553465536−2=65534

If we create subnets with a /24 subnet mask within this VPC:

* Number of subnets: 2(24−16) =2562^ {(24 - 16)} = 2562(24−16) =256
* Each subnet has 2(32−24) −2=2542^{(32 - 24)} - 2 = 2542(32−24)−2=254 usable IP addresses.

### IPv6 VPC Calculation

1. **CIDR Block**: Similar to IPv4, choose the CIDR block. IPv6 addresses are much larger, typically a /56 or /64 is used for a VPC.
2. **Subnetting**:
   * Decide on the number of subnets.
   * Calculate the prefix length for subnets.
3. **Host Calculation**:
   * Use the formula 2(128−subnet mask)2^{(128 - \text{subnet mask})}2(128−subnet mask) to calculate the total number of IP addresses in a subnet.

#### Example:

For a VPC with a CIDR block 2001:db8::/48:

* Total IPs: 2(128−48)2^{(128 - 48)}2(128−48)
* This is a vast number, so typically no host limitation concerns exist.

If we create subnets with a /64 subnet mask within this VPC:

* Number of subnets: 2(64−48)=655362^{(64 - 48)} = 655362(64−48)=65536
* Each subnet has 2(128−64)2^{(128 - 64)}2(128−64) IP addresses.

### Calculation Steps in Practice

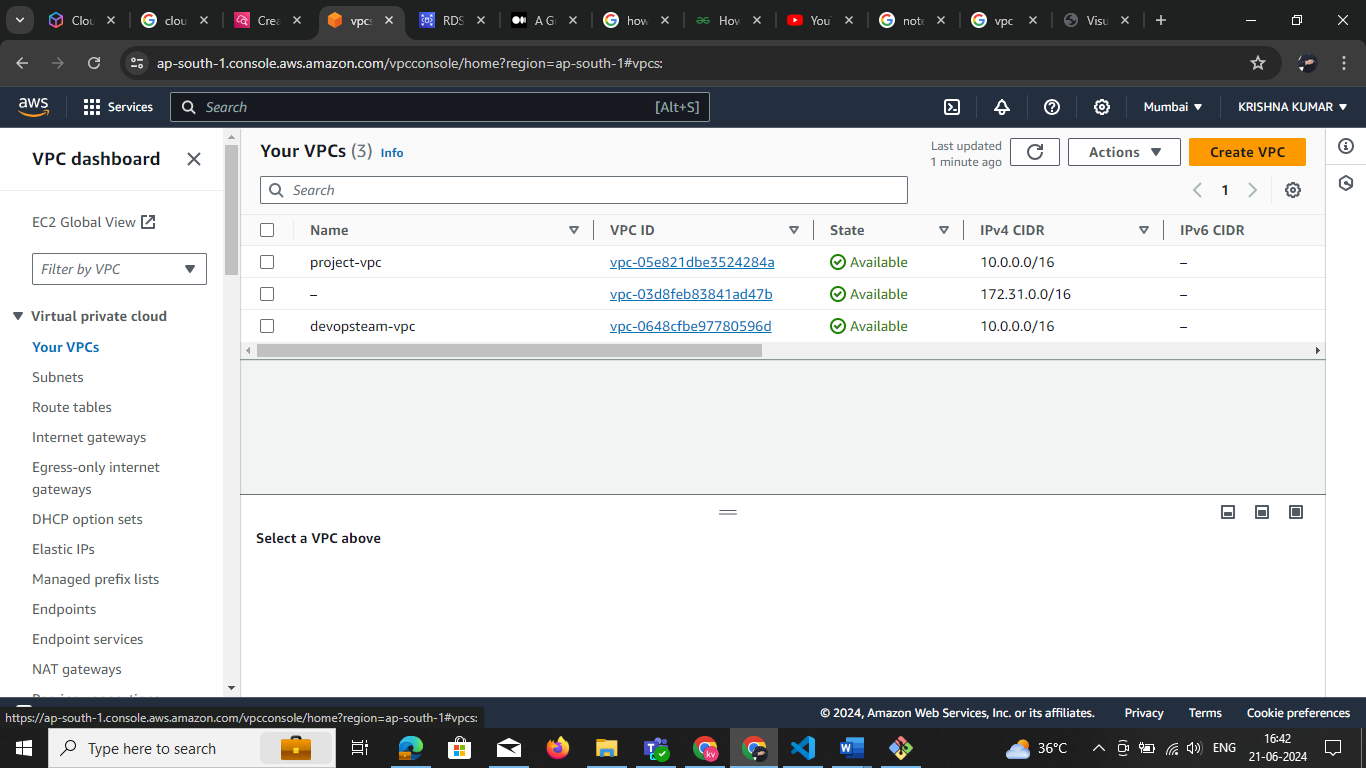
#### IPv4 Example Calculation

1. **CIDR Block**: 10.0.0.0/16
   * Total IPs: 2(32−16)=655362^{(32 - 16)} = 655362(32−16)=65536
   * Usable IPs: 65536−2=6553465536 - 2 = 6553465536−2=65534
2. **Subnetting**: If using /24 subnets:
   * Number of subnets: 2(24−16)=2562^{(24 - 16)} = 2562(24−16)=256
   * Each subnet has: 2(32−24)−2=2542^{(32 - 24)} - 2 = 2542(32−24)−2=254 usable IPs

#### IPv6 Example Calculation

1. **CIDR Block**: 2001:db8::/48
   * Total IPs: 2(128−48)≈1.2089×10242^{(128 - 48)} \approx 1.2089 \times 10^{24}2(128−48)≈1.2089×1024
2. **Subnetting**: If using /64 subnets:
   * Number of subnets: 2(64−48)=655362^{(64 - 48)} = 655362(64−48)=65536
   * Each subnet has: 2(128−64)≈1.844×10192^{(128 - 64)} \approx 1.844 \times 10^{19}2(128−64)≈1.844×1019 IPs

* **IPv4 VPC**: Define a CIDR block, calculate total and usable IPs, and divide into subnets as required.
* **IPv6 VPC**: Define a CIDR block, calculate total IPs (which are typically vast), and divide into subnets with large address spaces.
* These calculations ensure proper planning and allocation of IP addresses within a VPC, whether using IPv4 or IPv6.
* **Selecting VPC only option you need the create subnet and Routh tables , Nat gate way , by your Owen**

**Ones creating the vpc**

**VPC Resource map**

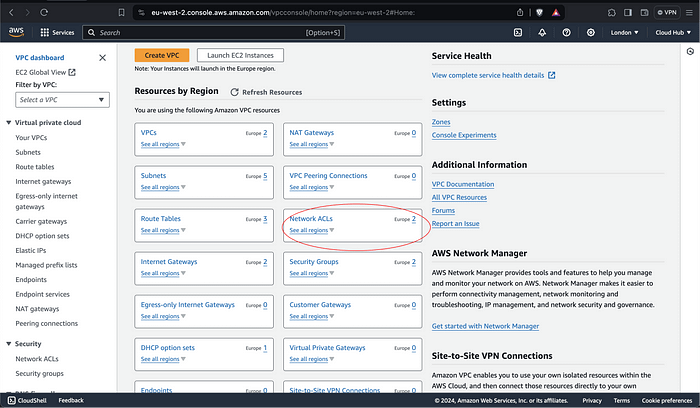


**NACL**

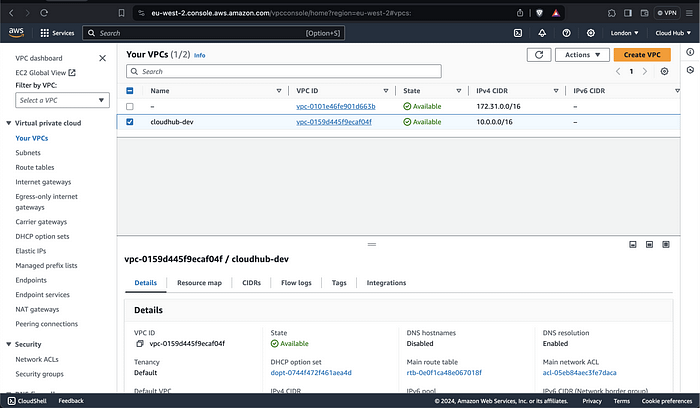
# 

# Setting Up AWS Network ACLs

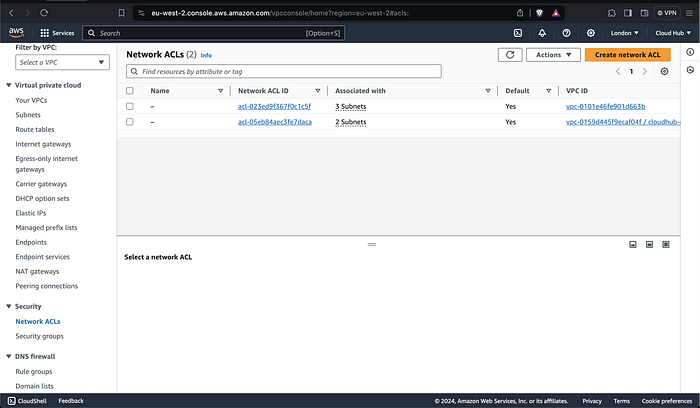
* Access the AWS Management Console and navigate to the VPC Dashboard.



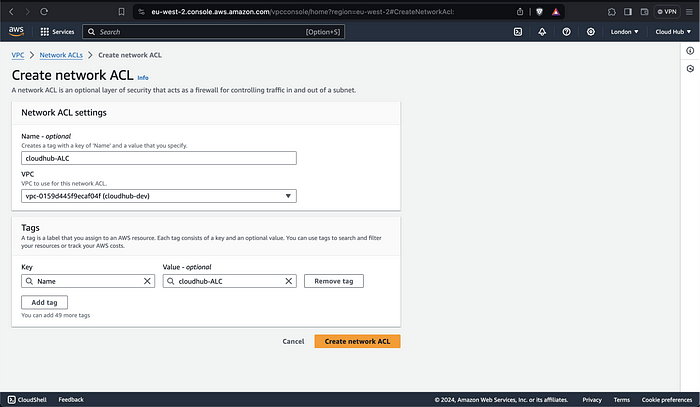
* Select the VPC for which you want to configure the Network ACL.



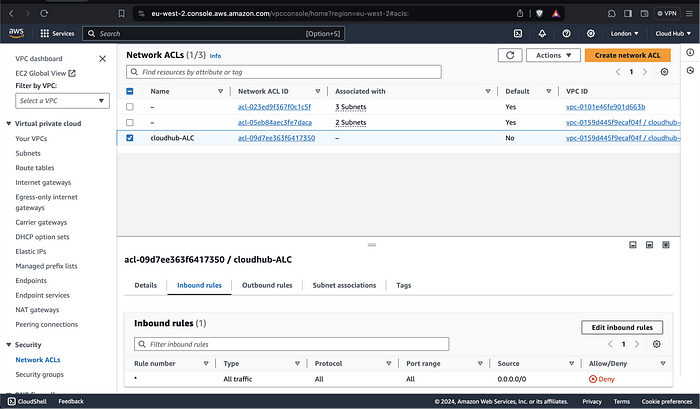
* In the navigation pane, click on “Network ACLs” and then click on “Create network ACL.”



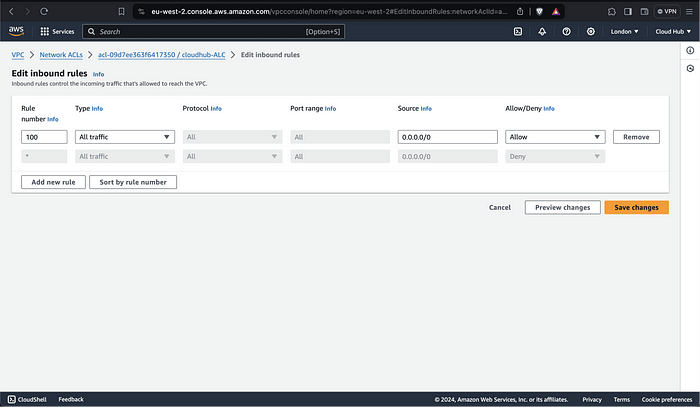
* Give your Network ACL a name and select the VPC to associate it with.



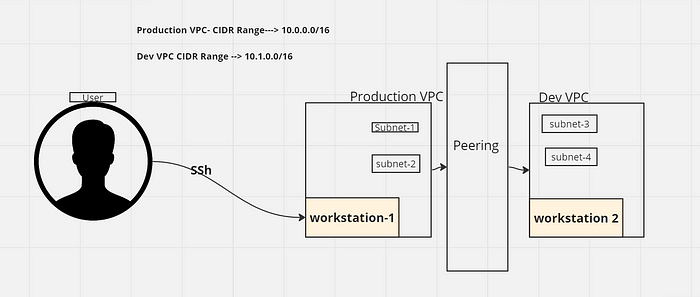
* Once the Network ACL is created, you can add inbound and outbound rules to control traffic flow.



* Click on “Edit inbound rules” or “Edit outbound rules” to add custom rules. Each rule consists of a rule number, an action (allow or deny), a protocol (e.g., TCP, UDP), a port range, and a source or destination IP range.

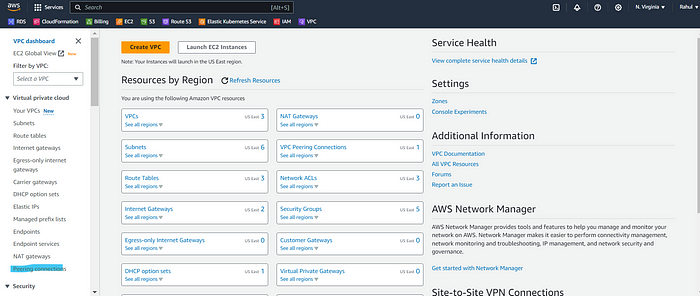


* After adding the desired rules, click on “Save” to apply the changes. Take the time to configure and maintain your Network ACLs diligently, and you’ll be well-equipped to defend your infrastructure against cyber threats in the dynamic landscape of cloud computing.
* If you are able to go through this process, you have successfully setup an ACLs for your subnets. Go ahead and select Subnet Associations and click on Edit Subnet Associations to add your Subnets.
* A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them using private IPv4 addresses or IPv6 addresses.  
  We can create VPC Peering between your VPC with the VPC in the same region or a different region or with other VPCs in a different AWS account in a different region
* In figure below if we want to connect workstation 1 to workstation2 , we need a peering connection, with out a peering connection we can not connect the workstations because both are in different subnet ranges.

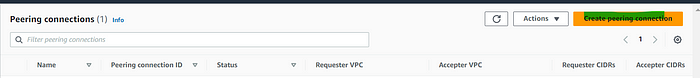
**VPC Peering** 

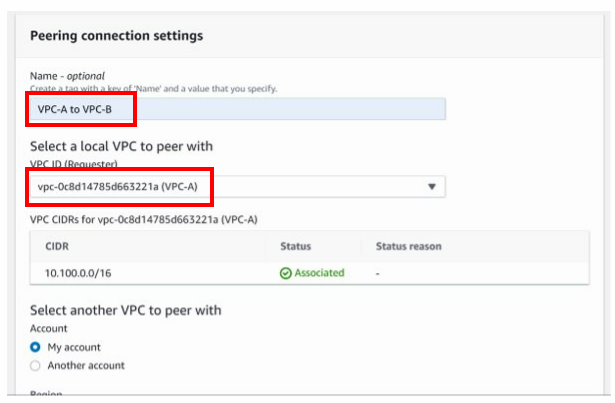
**Setup: VPC Peering Connection:-**

1. Create two VPCs, for example, Production VPC(CIDR: 10.0.0.0/16) and Dev VPC(CIDR: 10.1.0.0/16). And create subnets for VPCs and also you can edit the routing table if required.
2. Go to the “Peering Connections” in the AWS VPC Console

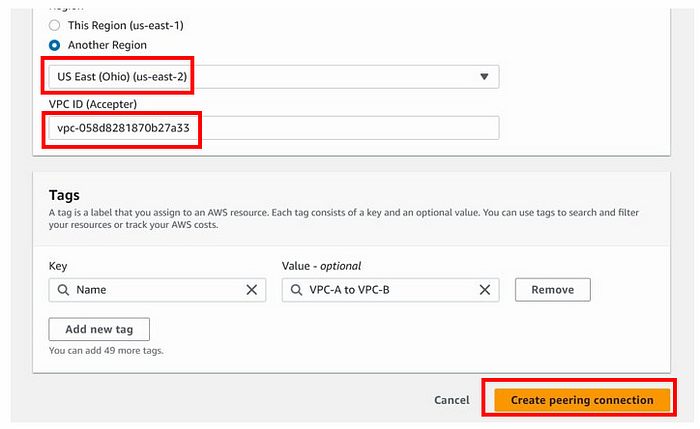


3. Click on “Create Peering Connection” and configure the peering as per the requirements.

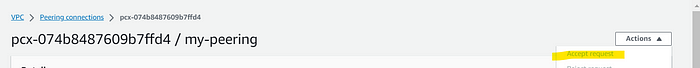




**NOTE: creating the VPC peer in my account the check CIDR block values**

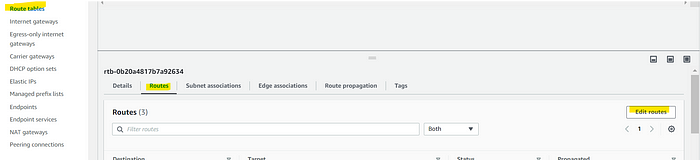


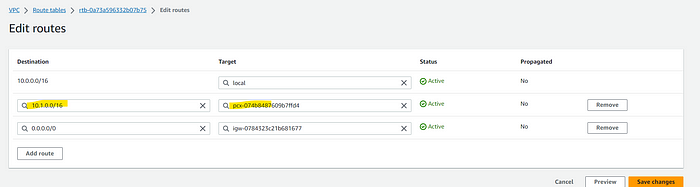
4. After creating the peering connection, you can see “Accept request” in the Action menu.



5. The next setup is to make an entry to the VPC route table. In route table select “**Production** ” route table, then go to the Routes tab and Edit it.

Click on “Add another route”. In the ‘destination field’ enters the IP of “ Dev VPC” like 10.1.0.0/16 and select the target as “VPC Peering Connection ID” of Production VPC to Dev VPC.





Repeat the same setups for all other Spoke VPC also.

**Advantages of Peering:**

1. **Improve security.** VPC peering comes with the major advantage of improving security by enabling private connectivity between two or more VPC networks, isolating traffic from the public Internet. Because your traffic never leaves the cloud provider’s network, you reduce a whole class of risks for your stack.
2. **Save money on network costs.** With VPC peering, you save on network transit costs and benefit from improved network latency. Because peering traffic does not leave your cloud provider’s network, that reduces public IP latency. And since peered networks use internal IPs to communicate, transferring data over the cloud provider’s network is cheaper than over the public Internet.

You cannot have more than one VPC peering connection between two VPCs at the same time.

You cannot create a VPC peering connection between VPCs that have matching or overlapping IPv4 or IPv6 CIDR blocks.

If you have multiple IPv4 CIDR blocks, you can’t create a VPC peering connection if any of the CIDR blocks overlap, even if you intend to use only the non-overlapping CIDR blocks or only IPv6 CIDR blocks.

# HA IPSec VPN tunnel between GCP and AWS.

In contemporary business operations, the internet plays an indispensable role, yet this dependence also exposes organizations to the risk of cyberattacks.

As instances of cybercrime continue to rise, businesses grapple with the challenge of safeguarding data without impeding their regular functions.

For numerous enterprises, the Highly Available IPSec tunnel emerges as a solution, facilitating the secure and reliable exchange and retrieval of data.

***What Is An IPSec Tunnel?***

An Internet Protocol Security (IPSec) tunnel is a set of standards and protocols originally developed by the[**Internet Engineering Task Force (IETF)**](https://www.ietf.org/) to support secure communication as packets of information are transported from an IP address across network boundaries and vice versa.

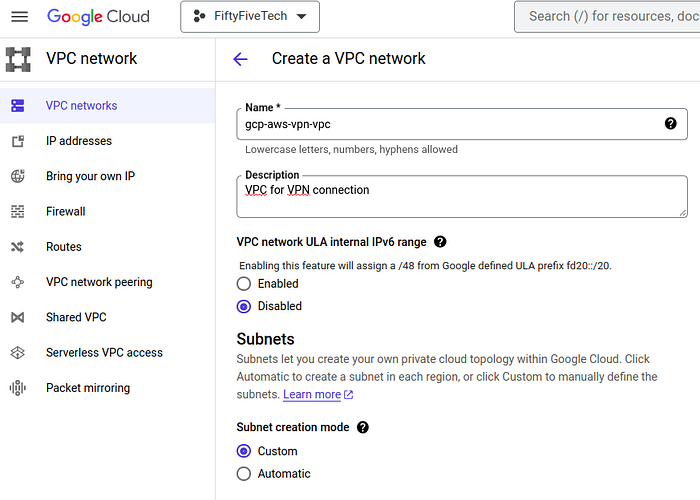
An IPSec tunnel allows for the implementation of a virtual private network (VPN) which an enterprise may use to securely extend its reach beyond its own network to customers, partners, and suppliers. IPSec provides the most robust cryptographic security.

***Objectives***

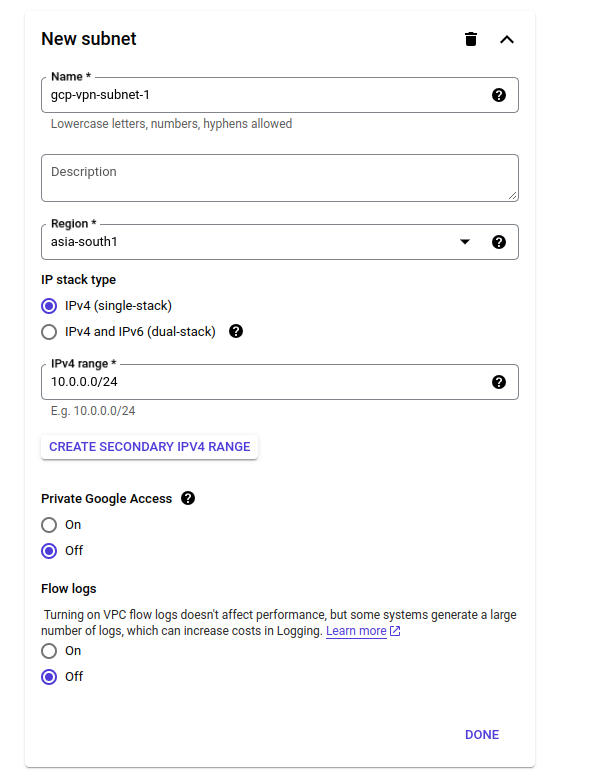
* Create a VPC network on Google Cloud.
* Create a VPC network on AWS.
* Create an HA VPN gateway and Cloud Router on Google Cloud.
* Create Customer gateways and Transit gateway on AWS.
* Create a VPN connection with dynamic routing on AWS.
* Create an external VPN gateway and VPN tunnels on Google Cloud.
* Verify and test the VPN connection between VPC networks on Google Cloud and AWS.

***1. Create a VPC and Subnet in GCP***

Create a VPC named **gcp-aws-vpn-vpc** and add a subnet to it.

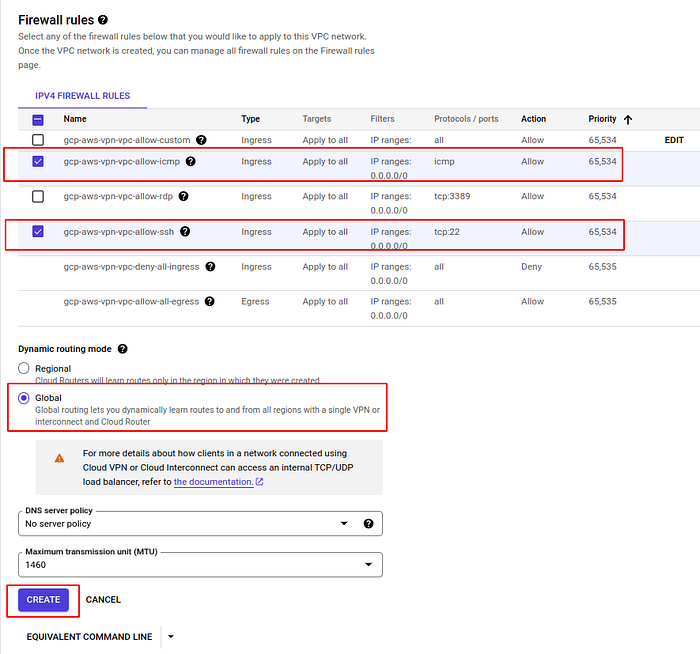


* Now, add a subnet and specify any region (**asia-south-1**) where you want your compute engine to launch. Add an IP address range as per your convenience (**10.0.0.0/24**).

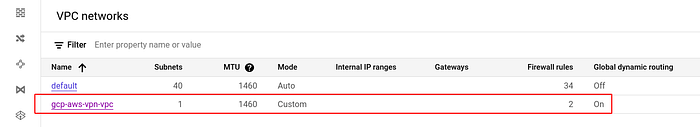


Select the firewall rules to allow **ssh** and **icmp** in the VPC.

Select Dynamic Routing mode as “Global” which helps to dynamically detect routes from all regions if you are using VPN or Cloud Router and then create VPC .

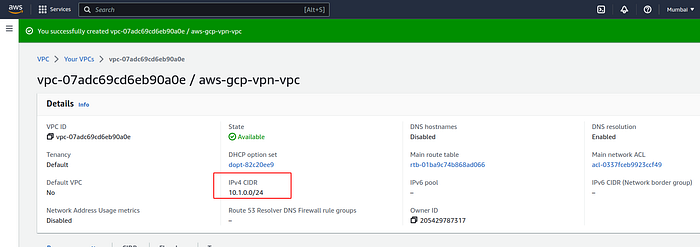


See, now a VPC is created in GCP Cloud.

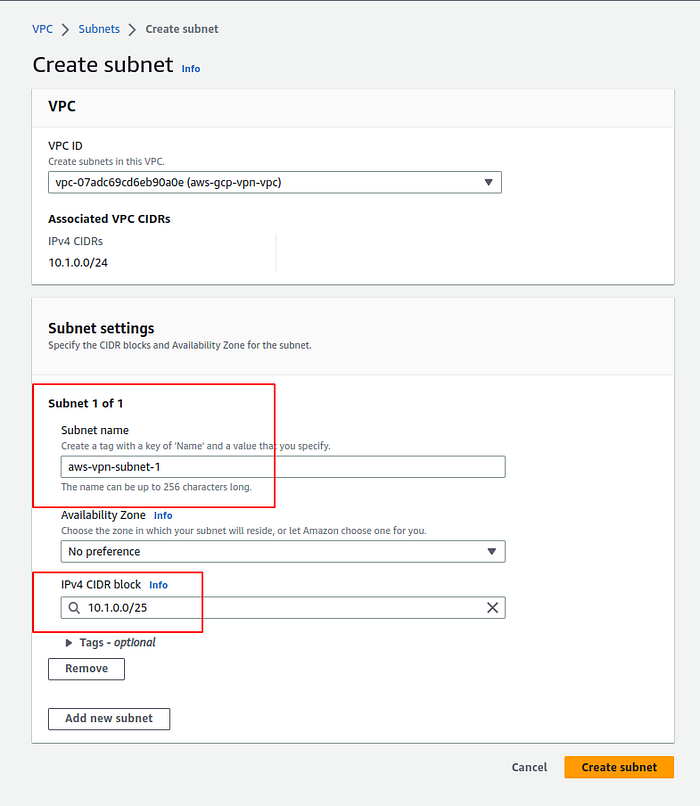


***2. Create a VPC and Subnet in AWS***

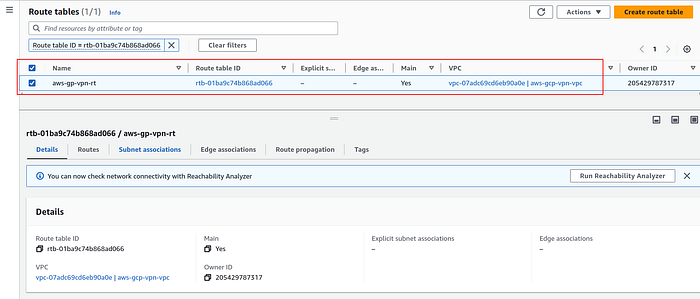
Create a VPC named as **aws-gcp- vpn-vpc** in AWS Cloud in us-east-1 region and add the cidr block as per your convenience and click create.



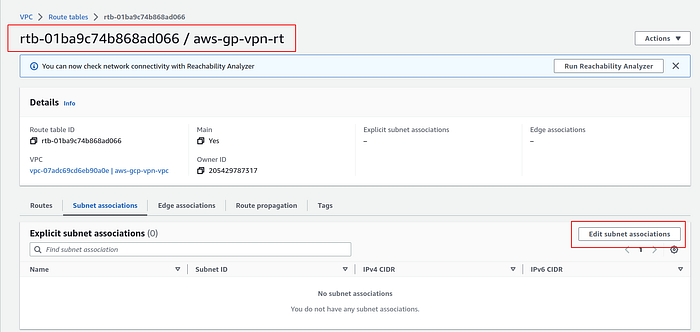
Now, create a subnet

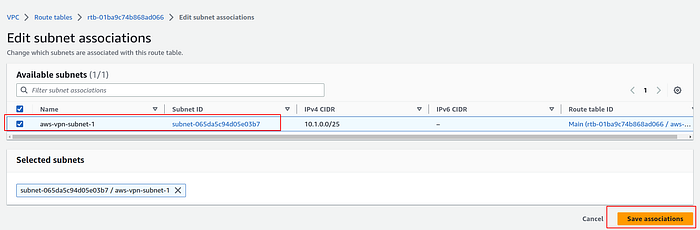


You will see that a default route table automatically gets created . Now you need to modify it.

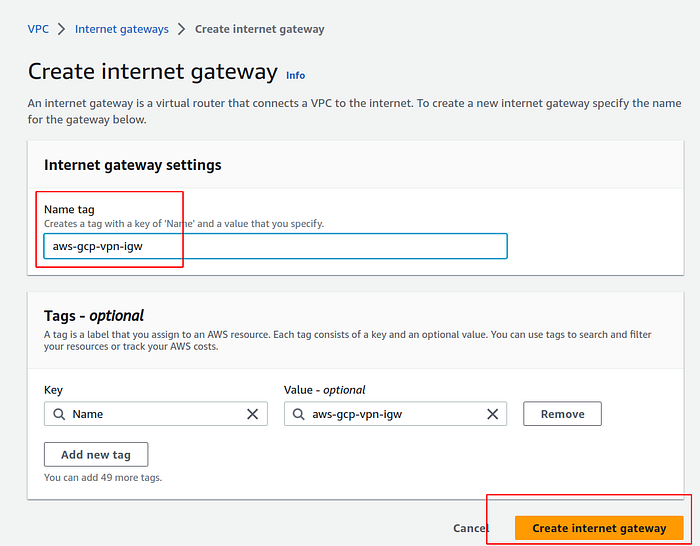


Attach your AWS subnet (aws-vpn-subnet-1) to route table in E**dit subnet association** field.

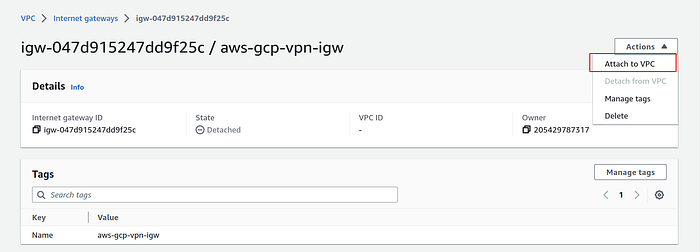


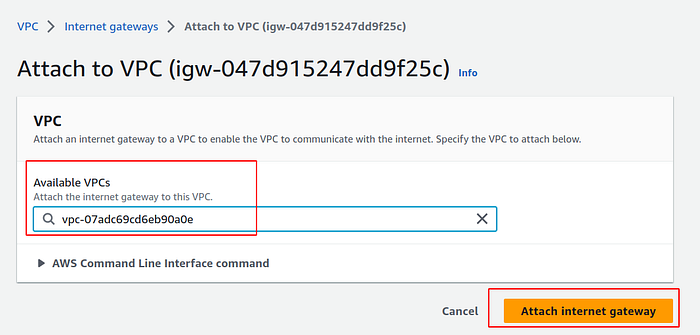


***3. Create Internet Gateway and update your default VPC route table in AWS***

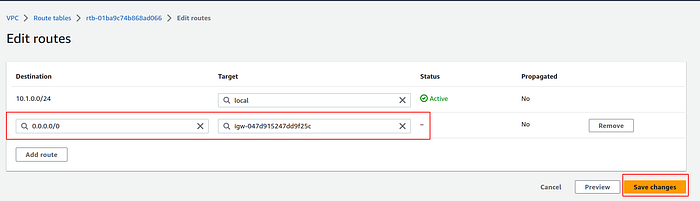


* Create an Internet gateway named as **aws-gcp-vpn-igw** and attach it to the new VPC which is created . It will allows communication between your VPC and the internet.





Then you need to edit your route table routes . Add another route which tells if any traffic comes from any ip apart from **10.1.0.0/25** will go to Internet Gateway.

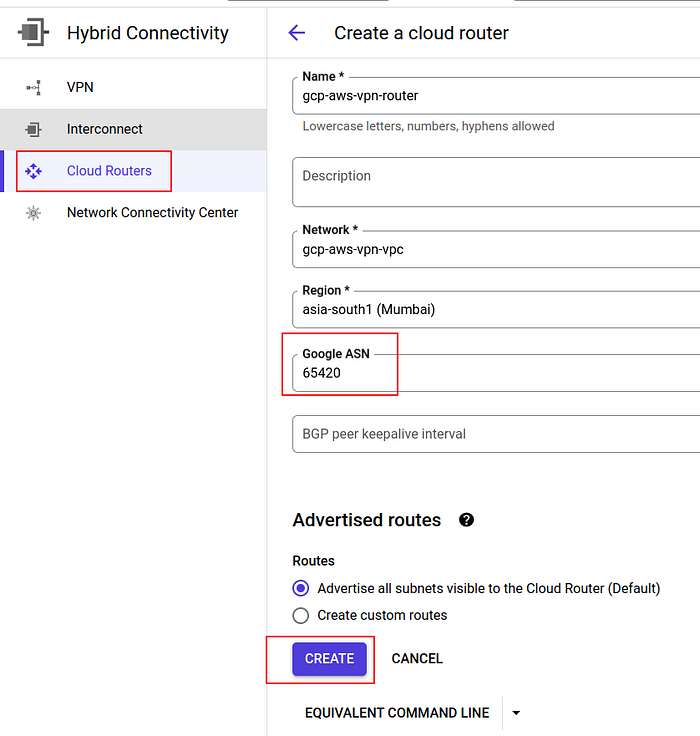


***4. Create a Cloud Router in GCP.***

Create a cloud router named as **gcp-aws-vpn-router,**attach it to the newly created VPC and assign a Google [ASN](https://www.thousandeyes.com/learning/glossary/as-autonomous-system#:~:text=The%20AS%20is%20assigned%20a,for%20private%20and%20reserved%20purposes.) number.

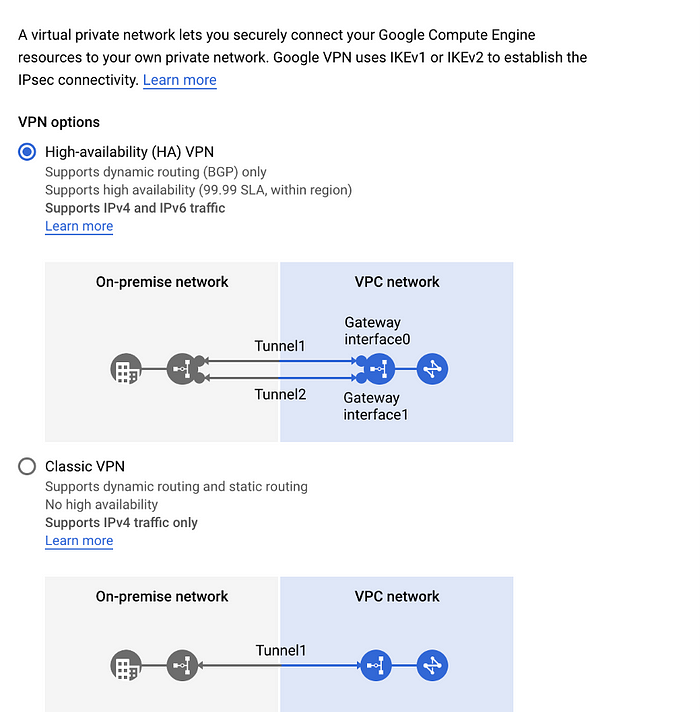
It can be any private ASN in the range**64512–65534** or **4200000000–4294967294** that you aren’t already using as a peer ASN in the same region and network.

Remember this Google ASN as same ASN number will be added in customer gateway in AWS Cloud too. This cloud router will provide dynamic routing using BGP for your VPC networks.

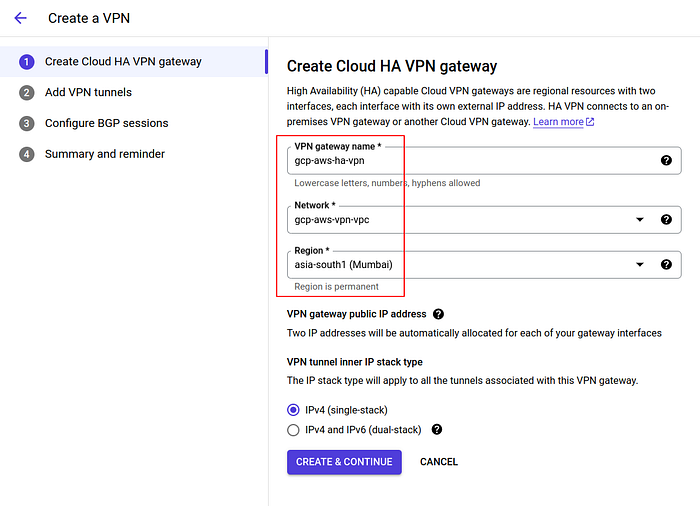


*5.****Create a HA VPN in GCP Cloud.***

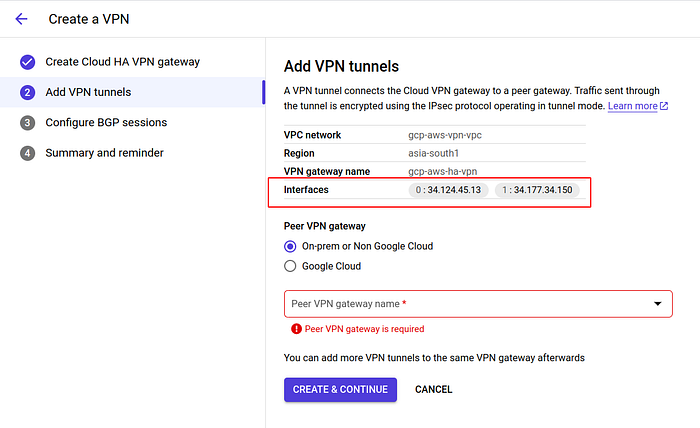
Choose High-availability (HA) VPN which supports dynamic routing and high availability and supports IPv4 and IPv6 traffic.



Now create a Cloud HA VPN Gateway with inputs as shown below



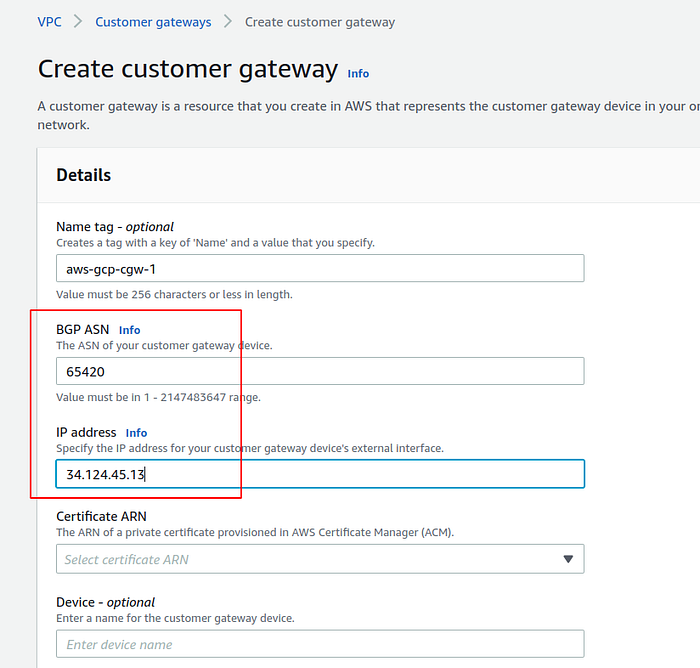
Now you can see that 2 public ip’s are automatically allocated for each of our gateway interfaces. We will need them for customer gateway creation in AWS.

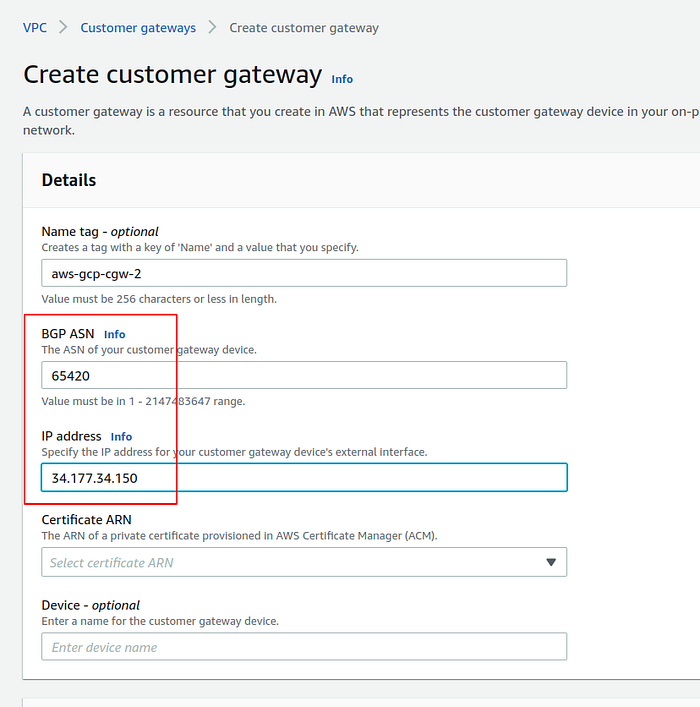


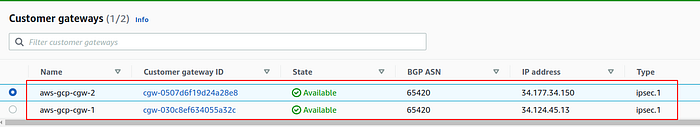
Now, here we need to take a halt and create some resources in AWS.

***6. Create Customer Gateway and Transit Gateway in AWS Cloud***

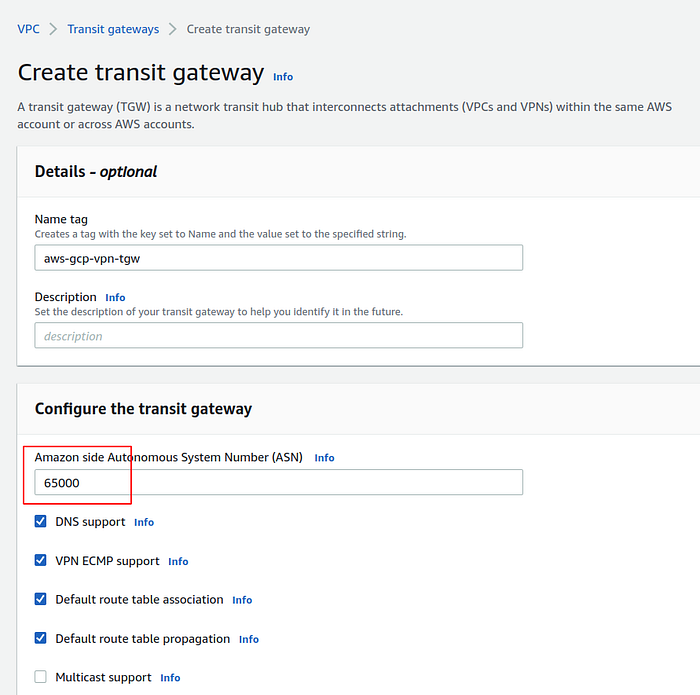
* Now, we need to create two customer gateways and add the same ASN number “65420” which you have assigned during VPN gateway creation in GCP.
* Provide same IP address which you got in GCP Console. The customer gateway that you created in AWS represents the customer gateway device in your GCP network.



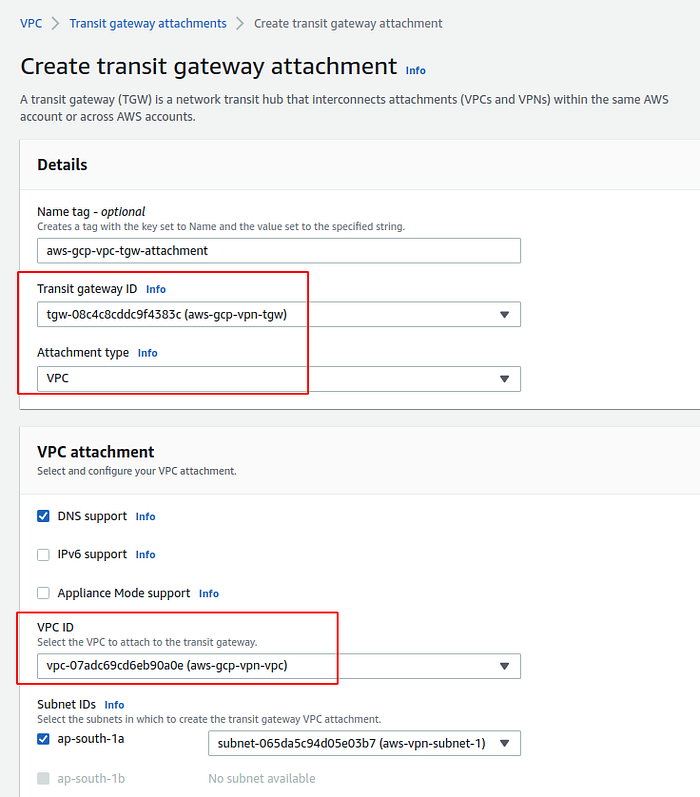


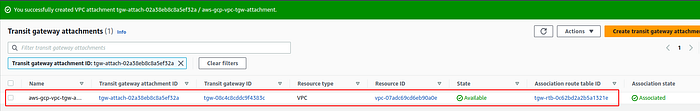


* Then create a Transit gateway**aws-gcp-vpn-tgw** and add a custom ASN no **(65000)** in the range 64512–65534 or 4200000000–4294967294.



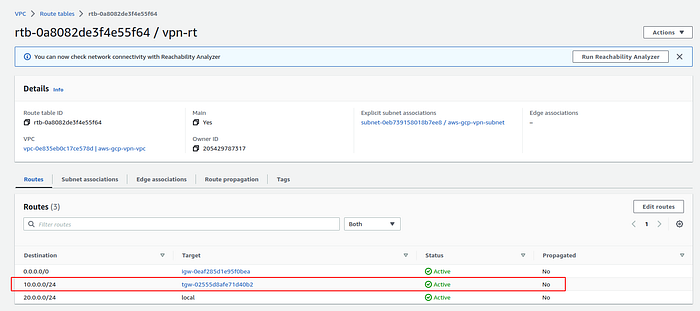
* Now, we need to create a **transit gateway attachment**to attach the transit gateway to the **VPC**.
* Select the transit gateway and VPC to be attached to it as shown below.





Once, the attachment is created we need to add a route in the default route table of the VPC as shown below.

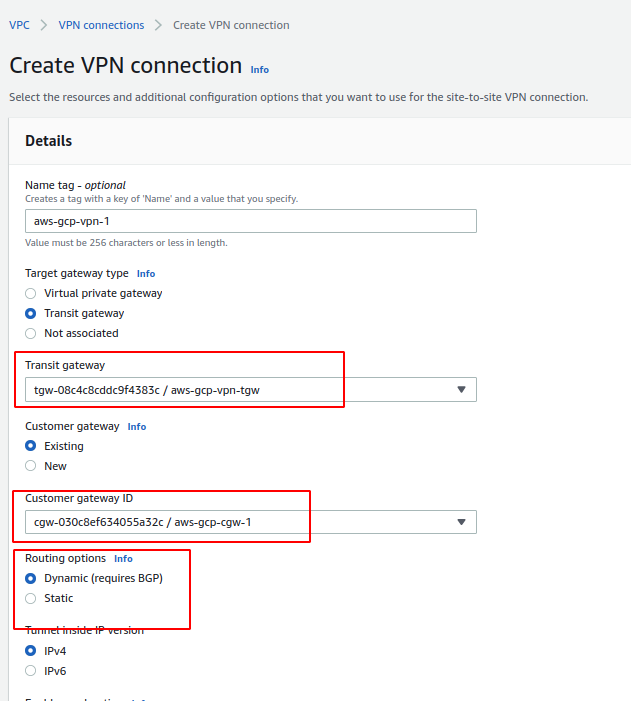
This route contains the VPC cidr of the GCP and target set as the **transit gateway.**

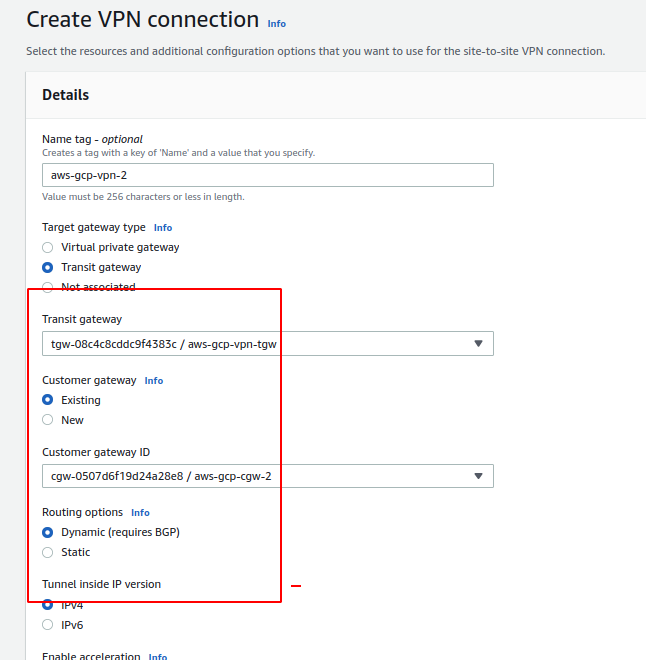


***6. Create Site to Site VPN connections in AWS***

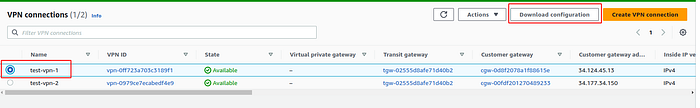
Here, we need to create two VPN connections since it’s a HA VPN. Each VPN connection includes two VPN tunnels which you can simultaneously use for high availability.

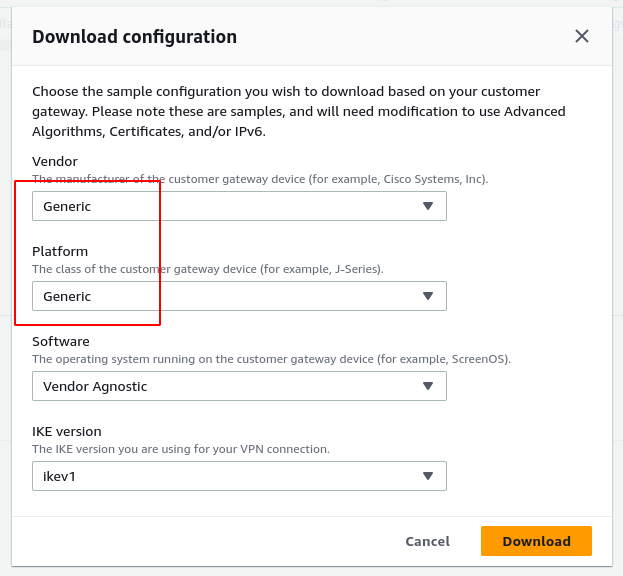
You need to select transit gateway and customer gateway and routing needs to be dynamic.





Now, download the configuration for each of the VPN Connections you created.

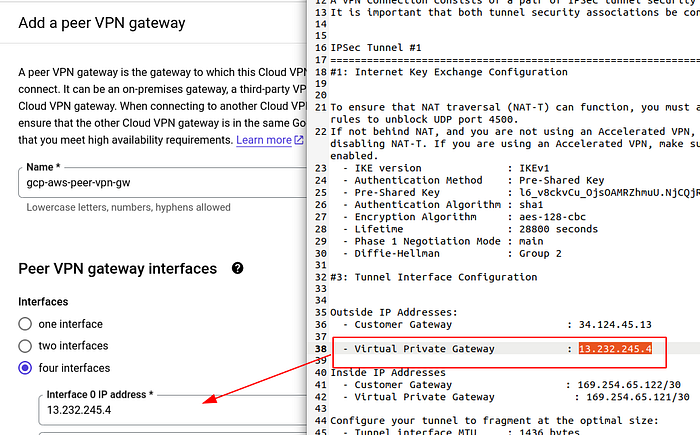


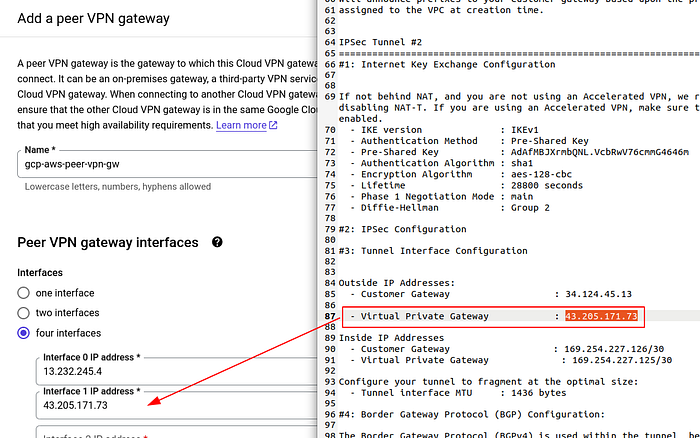


***7.******Add VPN Tunnels in your Google Cloud***

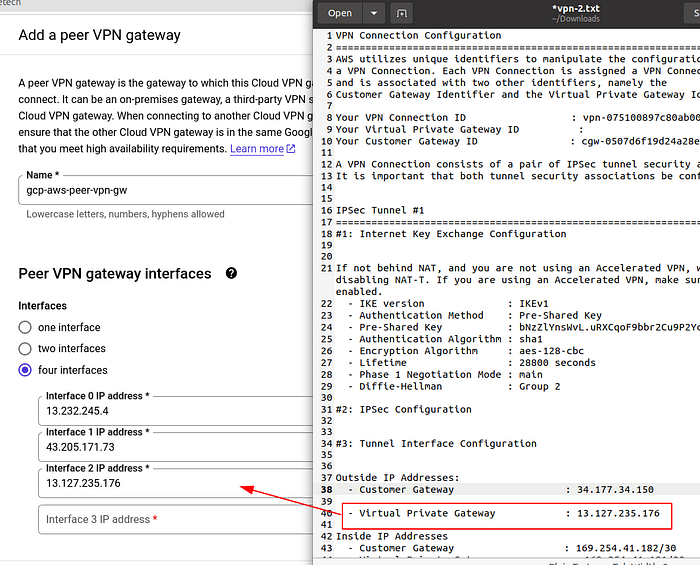
Now, we need to carry on on from where we left in GCP. i.e creating P**eer VPN gateway**.

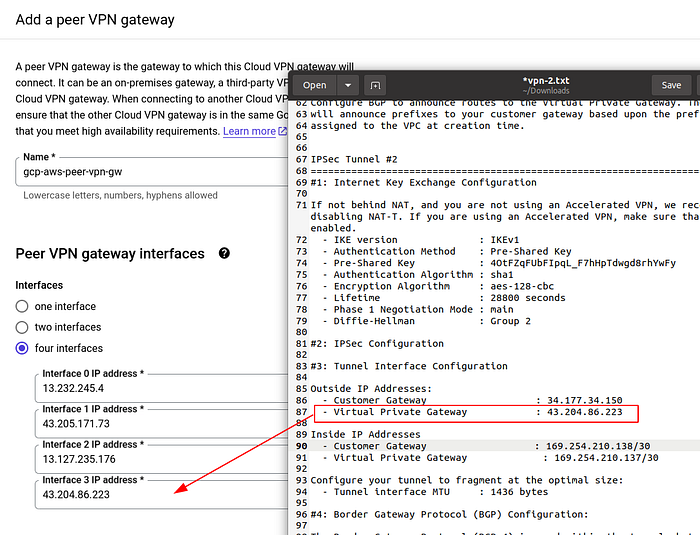
**VPN 1 configuration file:-**

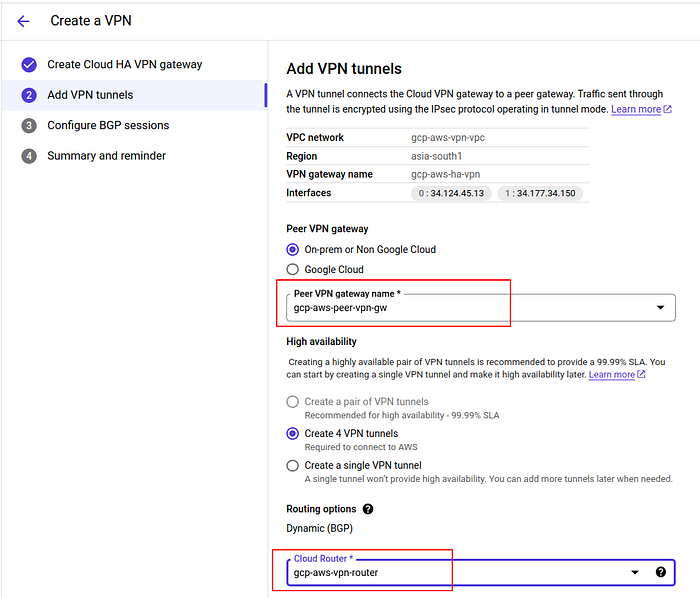




**VPN 2 configuration file:-**



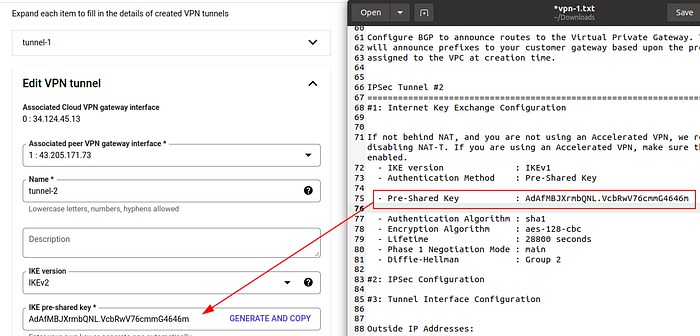




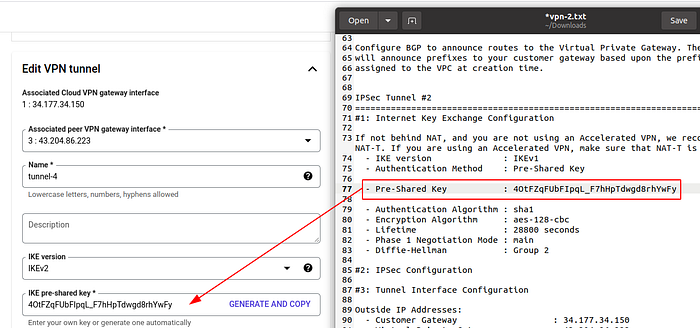
We need to create 4 vpn tunnels to connect to AWS, select **On-prem or Non Google Cloud option**and also the **cloud router** we created.

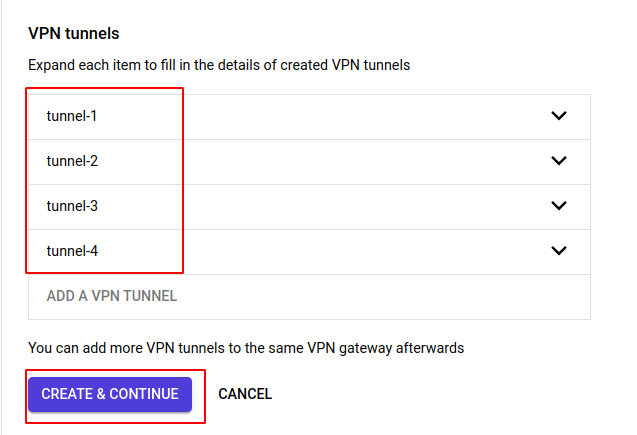
Now, we need to configure the VPN tunnels with **pre-shared key** as shown below.





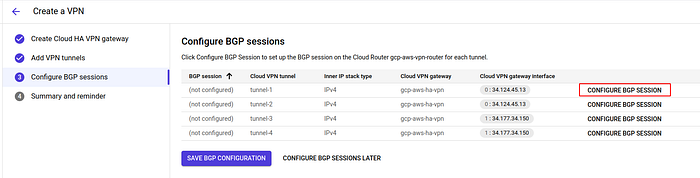






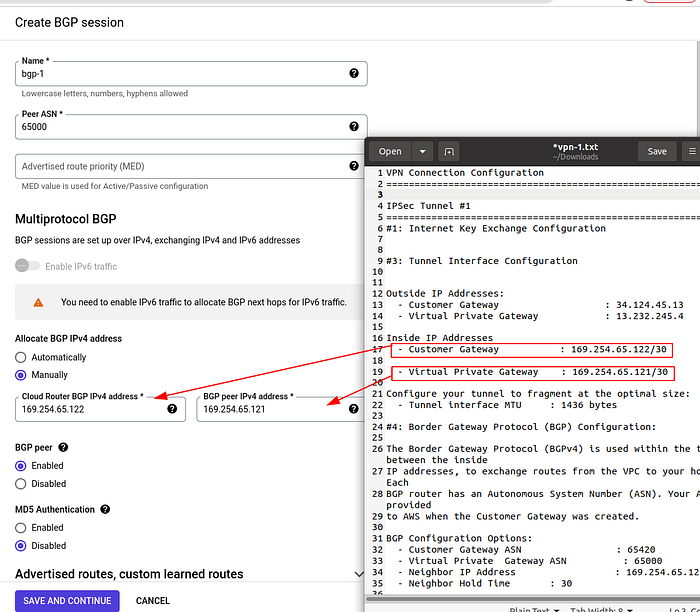
After you add the key for each of the 4 tunnels and click on continue.

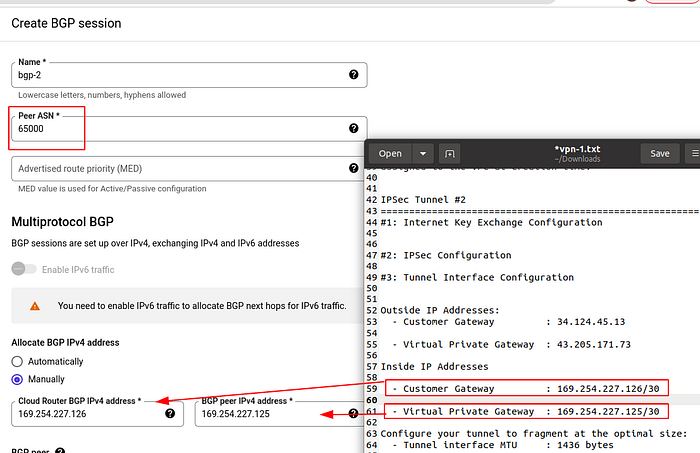
**7. Configure BGP sessions for each tunnel**



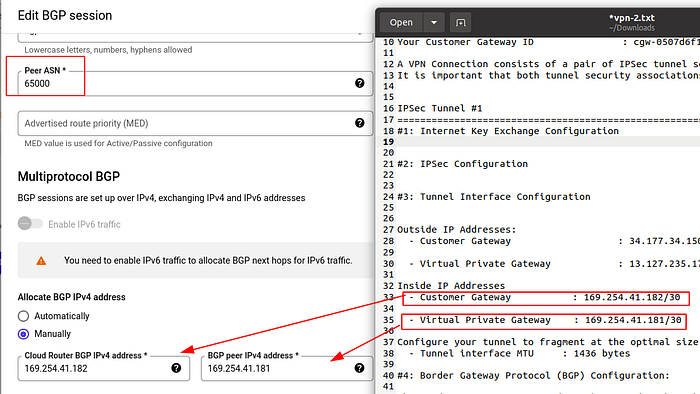
Enter the inside IP addresses and enter the Peer ASN (AWS Transit gateway ASN) **65000**as shown below.

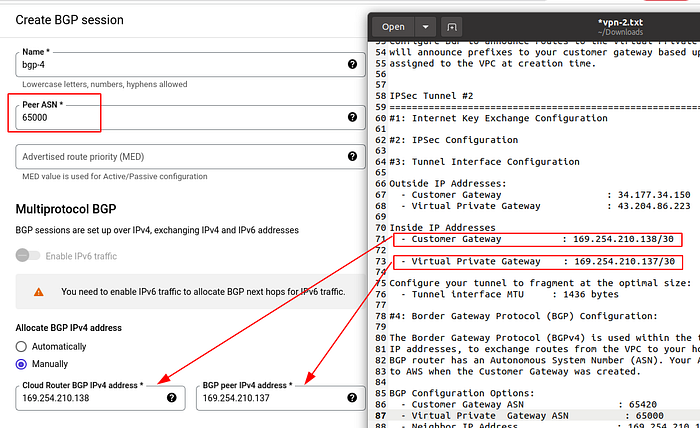
**VPN 1 configuration file:-**

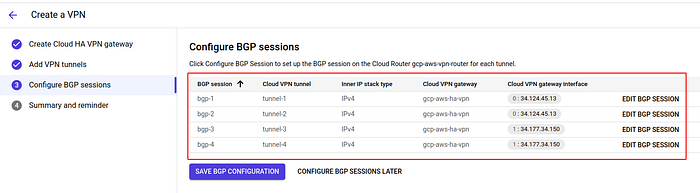




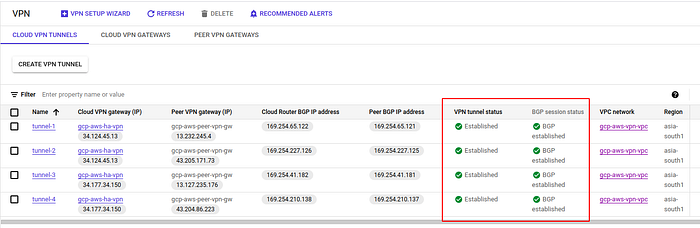
**VPN 2 configuration file:-**



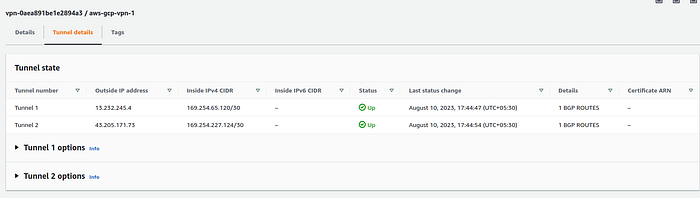


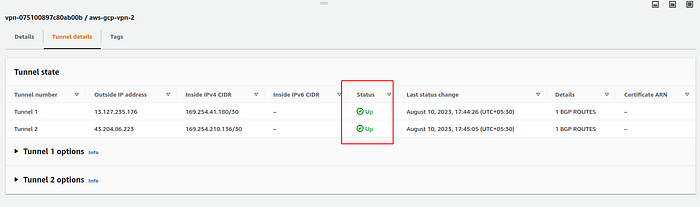


* Now, click on **save BGP configuration** and It’s done..!
* Make sure that the VPN tunnel status and BGP session status is green as shown below. This means that VPN has been successfully configured.

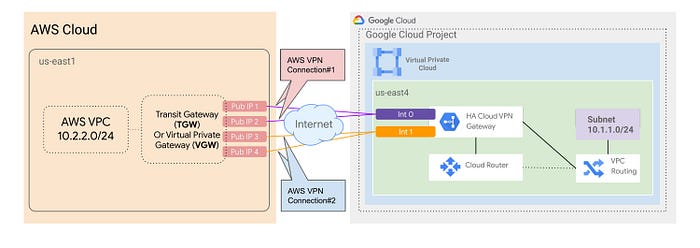


* If we go into AWS Console and click on Site-to-Site VPN and click on any vpn connections. You can see the tunnel is UP.





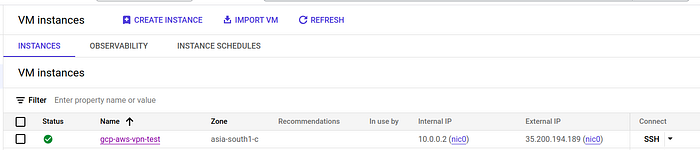
This is how the VPN architecture looks :-



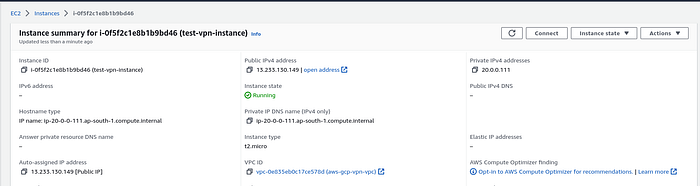
***9. Instances in GCP and AWS and check connectivity***

As shown below, we have created instances in AWS and GCP . Make sure you launch the instances in the newly created VPC’s.

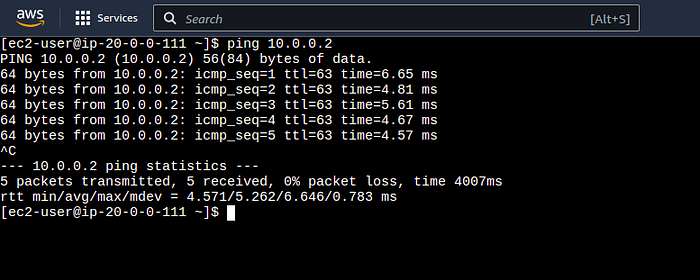
**GCP:-**



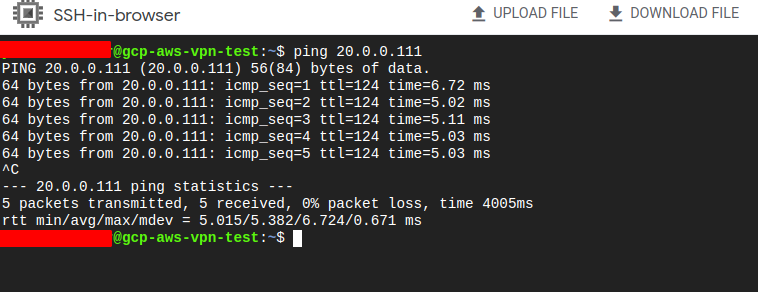
**AWS:-**



**AWS to GCP:-**



**GCP to AWS:-**

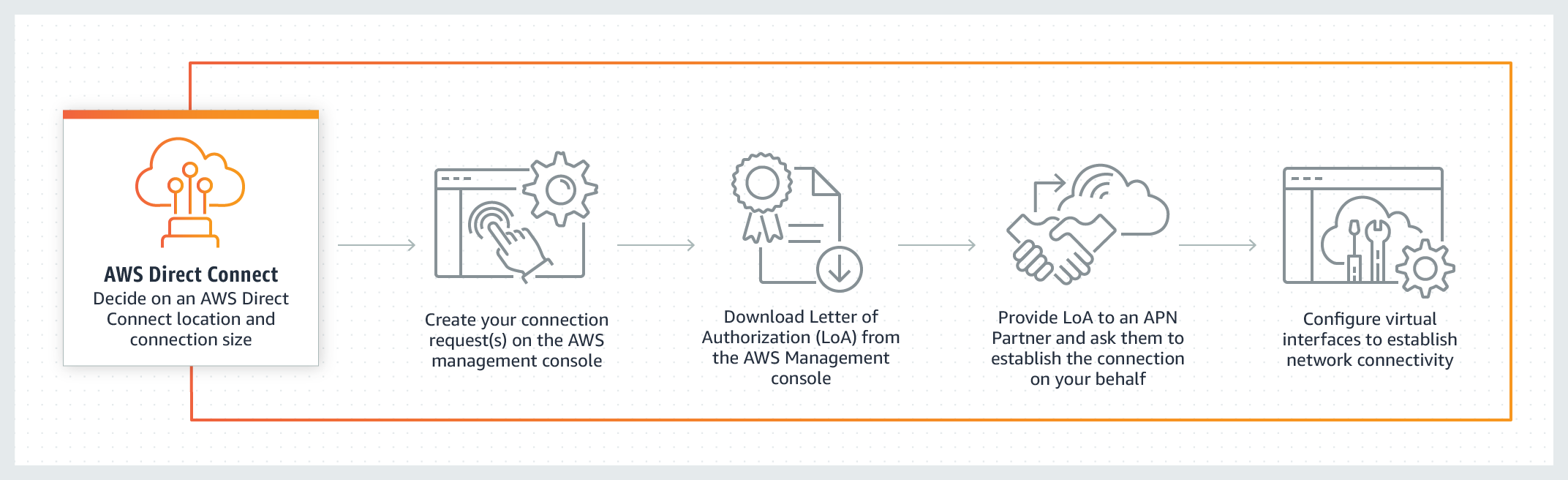


As seen above, we can now ping the instances with their private IP and see the packets are getting transferred without and data loss.

**Direct Connect Overview**

**AWS Direct Connect** connects your on-premises network to AWS via a dedicated network link. With this connection established, you can bypass your internet service provider and [develop virtual interfaces](https://k21academy.com/amazon-web-services/aws-vpc-virtual-private-cloud/) that link directly to the AWS Cloud. This can offer a more reliable network experience.

## **What is AWS Direct Connect**

* It utilizes a regular Ethernet fiber-optic cable to connect your internal network to an AWS Direct Connect site. The cable is attached to your router on one end and to an AWS Direct Connect router on the other. By passing internet service providers in your network path, you can construct virtual interfaces to public AWS services using this connection (for instance, to [Amazon S3](https://k21academy.com/amazon-web-services/amazon-s3-bucket-and-storage-classes/) or [Amazon VPC](https://k21academy.com/amazon-web-services/aws-vpc-virtual-private-cloud/)).
* AWS Direct Connect locations give users access to AWS in the region they are connected to. To access public AWS services in all other public Regions, you can use a single connection in a public Region or AWS GovCloud (US).

## AWS Direct Connect

## **How does it work?**

The quickest route to your AWS resources is through the AWS Direct Connect cloud service. Your network traffic never enters the open internet while it is in transit because it stays on the [AWS global network](https://k21academy.com/amazon-web-services/aws-solutions-architect/aws-global-infrastructure/). This lowers the likelihood of encountering bottlenecks or unanticipated latency increases.

You can deploy a dedicated connection from AWS at more than 100 AWS Direct Connect locations worldwide, or you can select a hosted connection offered by an AWS Direct Connect Delivery Partner. You can establish private network connections between the offices and data centers in your global network by sending data between AWS Direct Connect sites using AWS Direct Connect SiteLink.

## Working of Direct Connect **Components**

* **Connections** – To build a network connection from your premises to an AWS Region, create a connection at an AWS Direct Connect location. You can connect to all of the AZs in the area using Direct Connect.
* **Virtual interfaces** – To enable access to AWS services, create a virtual interface. Access to public services like S3 is made possible by a public virtual interface. Access to your VPC is made possible using a private virtual interface.

1. You must set up a public virtual interface and create a **Border Gateway Protocol** session in order to access public resources in a distant Region.

2. You can set up a **Direct Connect gateway** in any public Region. Use it to connect VPCs in your account that are situated in various Regions to your Direct Connect connection using a private virtual interface.

3. Request and set up two dedicated connections to AWS to offer failover. One or two routers in your network may be the final destination of these connections. Various setup options are available:

* **Active/Active (BGP multipath)** – This is the default configuration, where both connections are active. If one connection becomes unavailable, all traffic is routed through the other connection.
* **Active/Passive (failover)** – One connection is handling traffic, and the other is on standby. If the active connection becomes unavailable, all traffic is routed through the passive connection.

5. **Autonomous System numbers (ASN)** are employed to identify networks that provide the Internet with a fully defined external routing policy.

## **Features**

* **Reduces bandwidth costs −**It sends the data straight to and from AWS, which lowers the cost in both directions.
* **Compatible with all AWS services −** It supports all of the AWS online services, including Amazon S3,[Amazon EC2](https://k21academy.com/amazon-web-services/aws-solutions-architect/aws-ec2-instance/), Amazon VPC, etc.
* **Private connectivity to Amazon VPC** − Additionally, it is utilized to create a high-bandwidth private virtual link from your home network to Amazon VPC.
* **Elastic −** It offers connections ranging from 1 Gbps to 10 Gbps and allows you to create as many connections as you need.
* **Easy and simple** − It is easy to sign up using the [AWS Management Console](https://k21academy.com/amazon-web-services/aws-management-console-walkthrough/).

## **Use Cases**

1. **Build hybrid networks:**Connect your on-premises and AWS networks to create cross-environment apps without sacrificing performance.
2. **Extend your existing network:** You can use SiteLink to transport data between your locations once you’ve connected your network to Direct  Connect. Data moves over the quickest route possible while utilizing SiteLink.
3. **Manage large datasets:** For real-time analysis, quick data backup, or broadcast media processing, make sure seamless and reliable data transfers are performed on a large scale.

## use cases

## **Benefits**

1. **Optimal performance:**AWS compute nodes in the area are closer geographically, resulting in fewer network hops and up to 44% less latency.
2. **Reduced costs:** Reduced network expenses and cost reductions of between 60% and 70% on data egress rates.
3. **Increased security:**From your on-premises network to your Amazon VPC, a private and secure interface is provided.
4. **Reliability:**Reduced unpredictability of up to 60% when connecting at the edge of the AWS network backbone.

## **Difference Between AWS Direct Connect & Site to Site VPN**

## difference between direct connect and Site to site VPN**Monitoring**

* Your Direct Connect resources can potentially have tags applied to them in order to be managed or categorized. You define both the key and the optional values that make up a tag.
* All AWS Direct Connect API calls are recorded by [CloudTrail](https://k21academy.com/amazon-web-services/amazon-cloudwatch-and-cloudtrail/) as events.
* Create [CloudWatch](https://k21academy.com/amazon-web-services/deploy-aws-codepipeline/) alarms to keep an eye on metrics.

## Monitoring of AWS direct Connect

## **Pricing**

* Port hours and outgoing data transfer are the two components of billing for it. Capacity and connection type impact how much a port hour costs (dedicated connection or hosted connection).
* Data Exchange The [AWS account](https://k21academy.com/amazon-web-services/create-aws-free-tier-account/) is responsible for the Data Transfer will be charged for private interfaces and transit virtual interfaces. The use of an AWS Direct Connect gateway with multiple accounts is free.
* For publicly addressable AWS resources (such as Amazon S3 buckets, Classic EC2 instances, or EC2 traffic that passes through an internet gateway), the Data Transfer Out (DTO) usage is metered toward the resource owner at the AWS Direct Connect data transfer rate if the outbound traffic is headed for public prefixes owned by the same AWS payer account and actively advertised to AWS through an AWS Direct Connect public virtual Interface.

**Got 🡪 to directed connecter -🡪 create connect -🡪**



**Creating a Direct Connect (DX) Connection**

The DX connection can be created in two options (connection ordering types) — See Figure 03.

1. Classic — Can create connections one at a time
2. Connection Wizard / Resiliency Toolkit — Can create connections using resiliency recommendations. (Recommended)

Out of these two, it is recommended to follow the connection wizard / resiliency toolkit, since it gives you three options for resiliency.

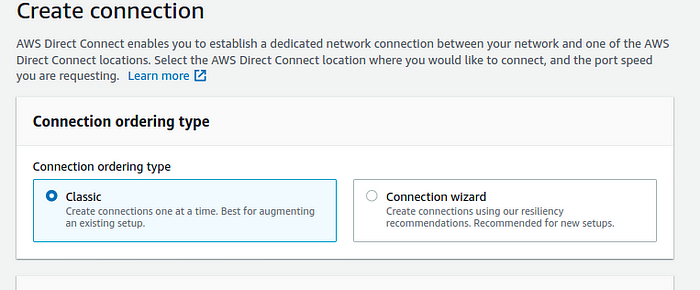
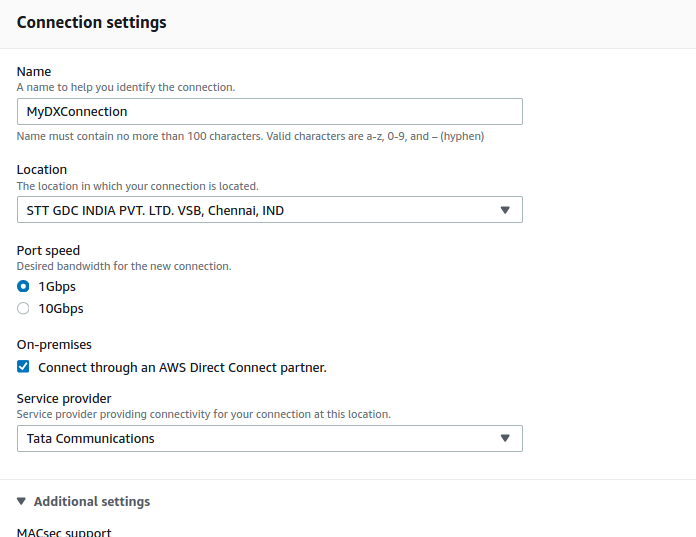


Figure 03 — Connection ordering types

Once the connection ordering type is selected, you can proceed with the DX connection creation.

**The Classic Approach**

Lets assume we selected the “classic” approach here (See Figure 04).



* While creating the connection, you are required to select a “closer” location as the Direct Connect Location. In the above figure, I have selected “STT, GDC INDIA PVT, LTD in Chennai. You may adjust your location depending on your customer data center location.
* Once the “location” is selected, you will be able to connect to a port within 72 hours. If you have the connectivity for the port defined, you can extend that connection to your customer data center without any issue. If the selected location is not able to find a connection to your customer data center, AWS partners can help you on this process.

Selecting this partners can be done while creating the connection itself (See Figure 04). You can see a check box checked by default. You may select the “Service Provider” drop down to choose your preferred partner. In this example, I have chosen “Tata Communication” as the partner . It is your decision to select a partner of your choice.

**The Resiliency Toolkit Approach**

Launched in 2019. With this approach, you can have multiple resiliency approaches. Depending on your resiliency requirement, you may chose the best available option (See Figure 05).

1. Development and Test
2. High Resiliency
3. Maximum Resiliency

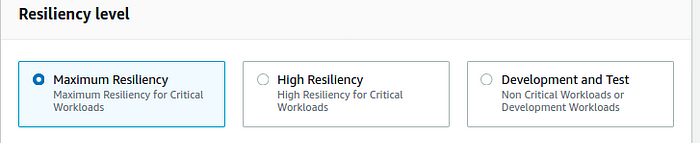


Figure 05 — Resiliency Levels

**Development and Test Resiliency**

This is used for non-critical workloads or development workloads. **Provides resiliency by having separate devices in a single location** (See Figure 06).

This provides resiliency against device failures but does not provide resiliency against location failures.

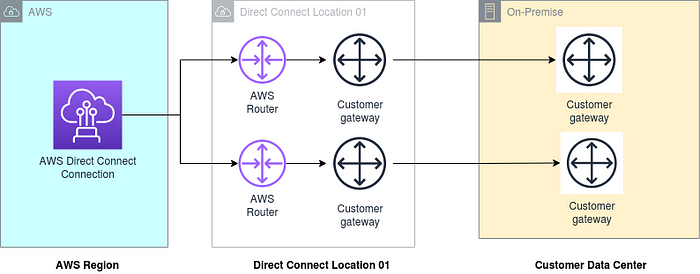


Figure 06 — Device level resiliency

**High Resiliency**

This is used for critical workloads. **Provides resiliency by having two separate connections in multiple locations** (See Figure 07).

Therefore, it provides resiliency at the device and the location levels.

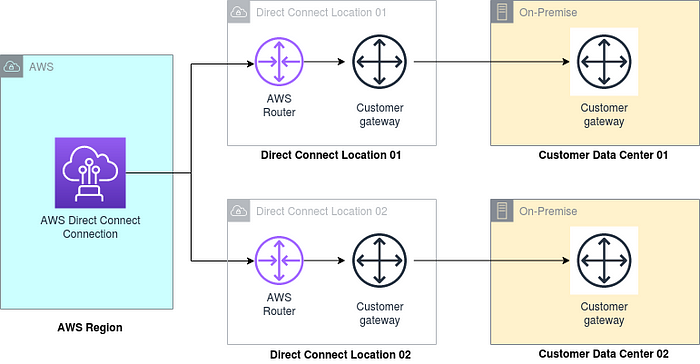


Figure 07 — Device and Location level resiliency

**Maximum Resiliency**

This is used for critical workloads. **Provides resiliency by having two separate connections in multiple locations**, **where each location have multiple devices**providing maximum level of resiliency (See Figure 08).

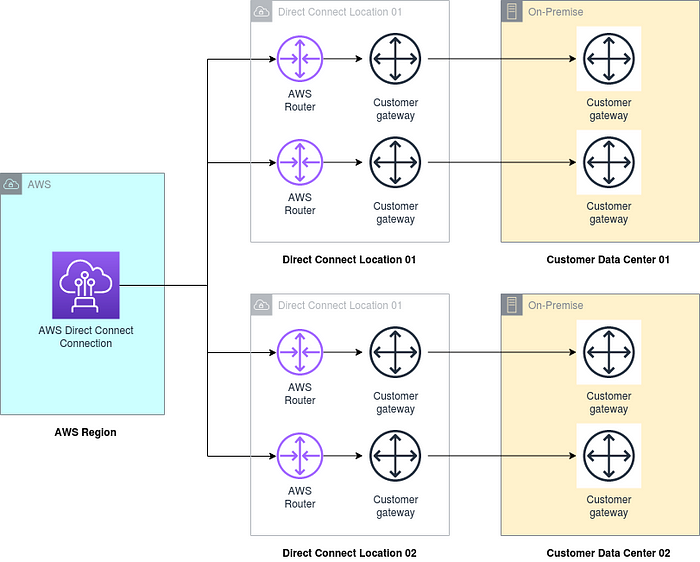


Figure 08 — Device level and location level maximum resiliency

**AWS Direct Connect (DX) — Logical Connection**

Once the physical connectivity is established, we are now ready to implement the logical connectivity.

The logical connectivity primarily implemented through Virtual Interfaces (VIFs).

**Virtual Interfaces (VIFs)**

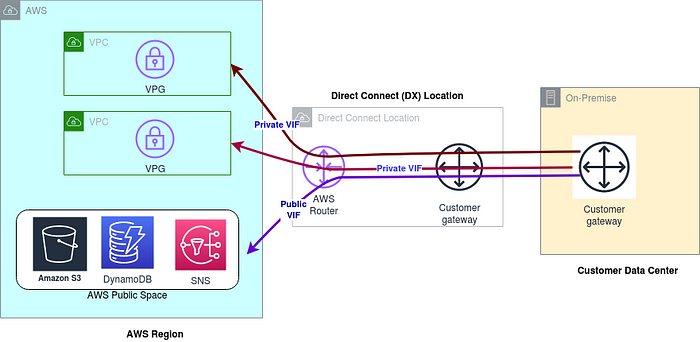


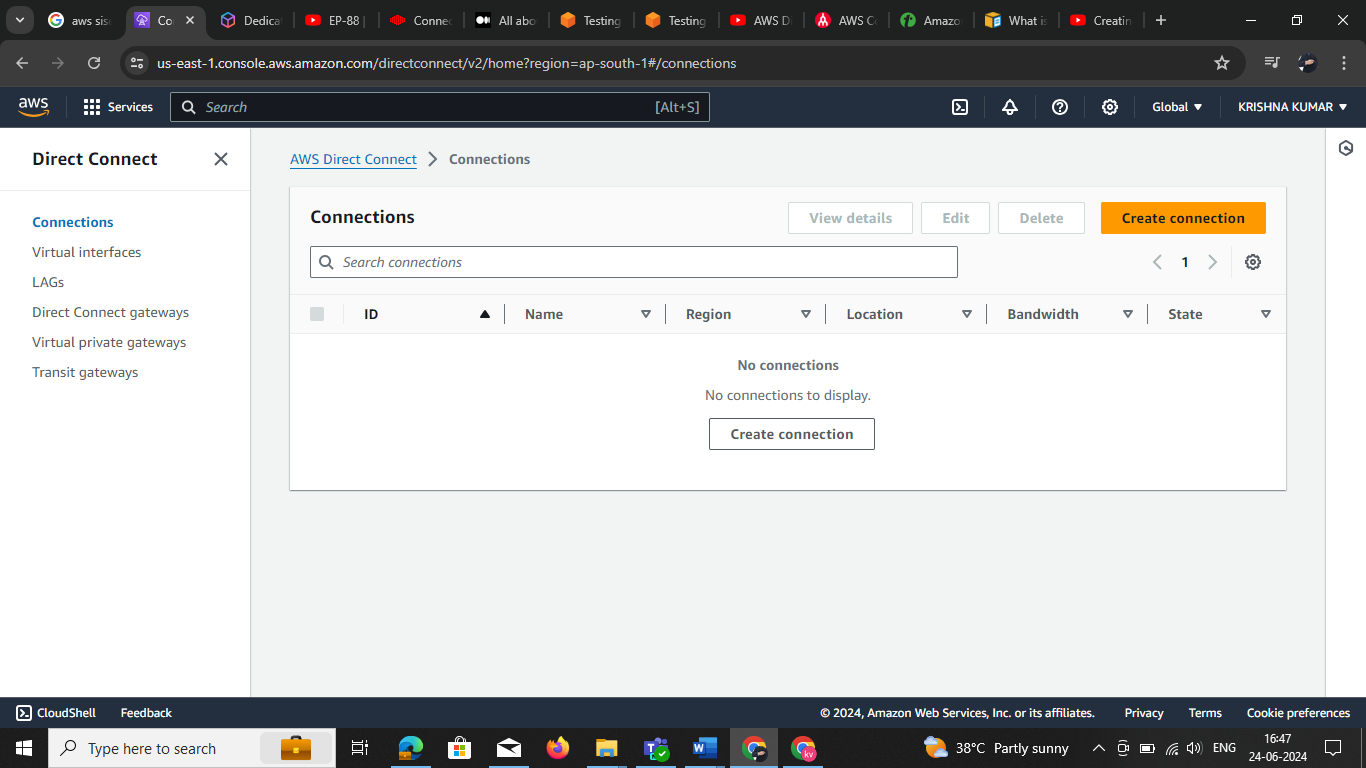
Figure 09 — DX Virtual Interfaces (VIFs)

There are two types of VIFs (See Figure 09).

1. **Public VIF** — This lets you connect to AWS services, which are in the public space such as S3, DynamoDB, etc
2. **Private VIF** — This lets you connect to AWS VPCs. Each private VIF connects to a separate VPC in AWS. For example, if there are two VPCs, there will be two private VIFs.

There needs to be some routing configurations should happen between AWS Router and the Customer Gateway within the DX location for every VIF that you are planning to configure. Primarily it will be a VLAN connection with BGP routing.

* **ANOTHER METHOD TO DO IT refer the below doc LINKS**



**Virtual interfaces doc**

<https://aws.amazon.com/blogs/networking-and-content-delivery/testing-aws-direct-connect-resiliency-with-resiliency-toolkit-failover-testing/>

**Transit gateway doc**

<https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/aws-direct-connect-aws-transit-gateway.html>

**Virtual private gateways**

<https://docs.aws.amazon.com/directconnect/latest/UserGuide/virtualgateways.html>

**Direct Connect gateways**

<https://docs.aws.amazon.com/directconnect/latest/UserGuide/direct-connect-gateways-intro.html>

**LAGs**

<https://docs.aws.amazon.com/directconnect/latest/UserGuide/associate-connection-with-lag.html>