VPC PEERING

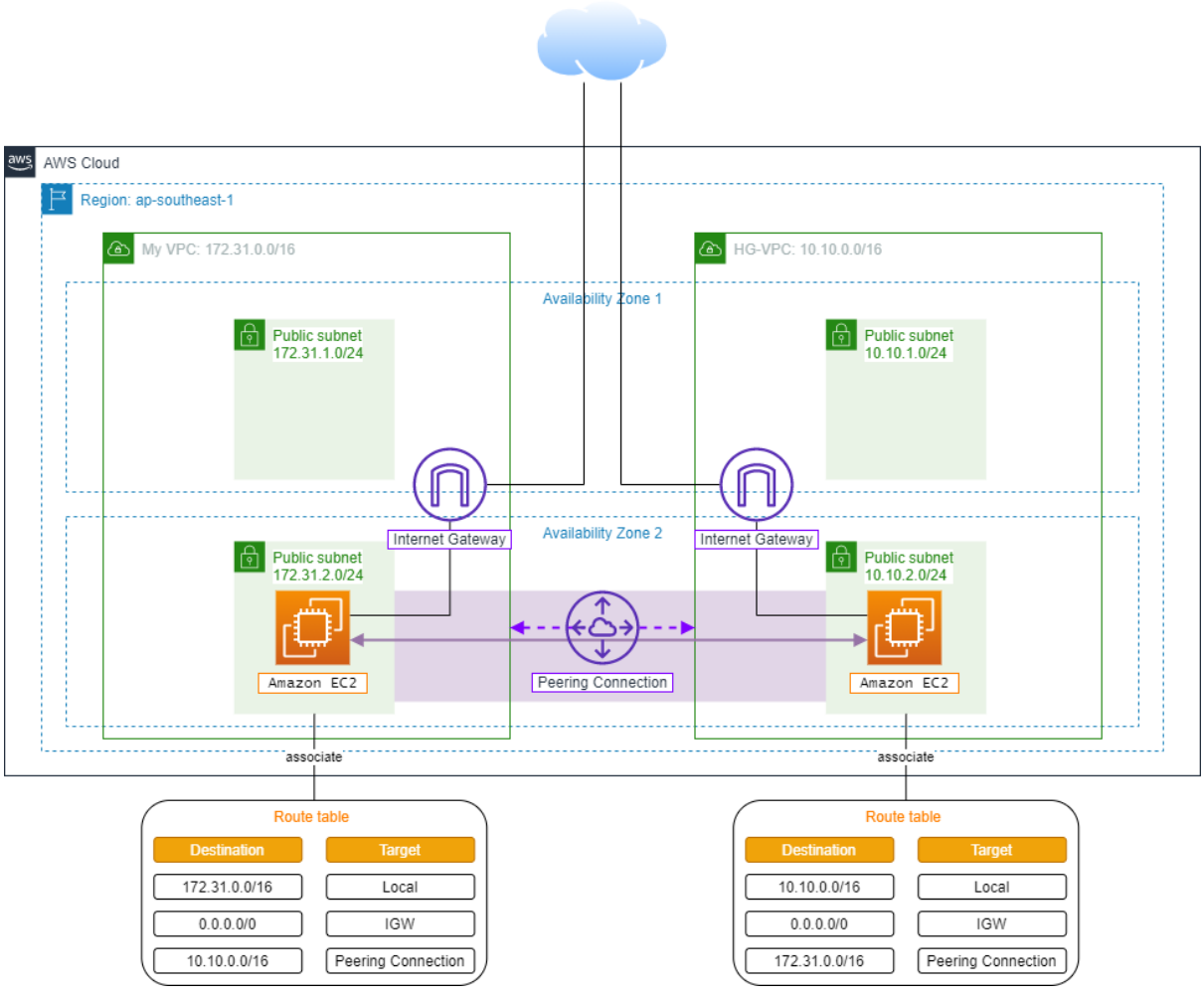
VPC Peering is a networking feature in Amazon Web Services (AWS) that enables you to connect two Virtual Private Clouds (VPCs) together, allowing them to communicate as though they were in the same network.

🌐 It allows for private connectivity between the VPCs without the need for gateways, VPN, or separate physical infrastructure.

🌐 This can be useful in many scenarios, such as sharing resources across accounts, regions or deploying multi-tier applications across multiple VPCs.

🌐 The VPC Peering connection is established between two VPCs, where the peering connection is configured, by creating a route in both VPC route tables to allow traffic to flow between the two VPCs.

🌐 It should be noted that VPC Peering is only possible within the same region and between VPCs that have non-overlapping IP address ranges. To connect VPCs that have overlapping IP address ranges, you will need to use a different method, such as VPN or Direct Connect.



Step 1: Create two VPCs

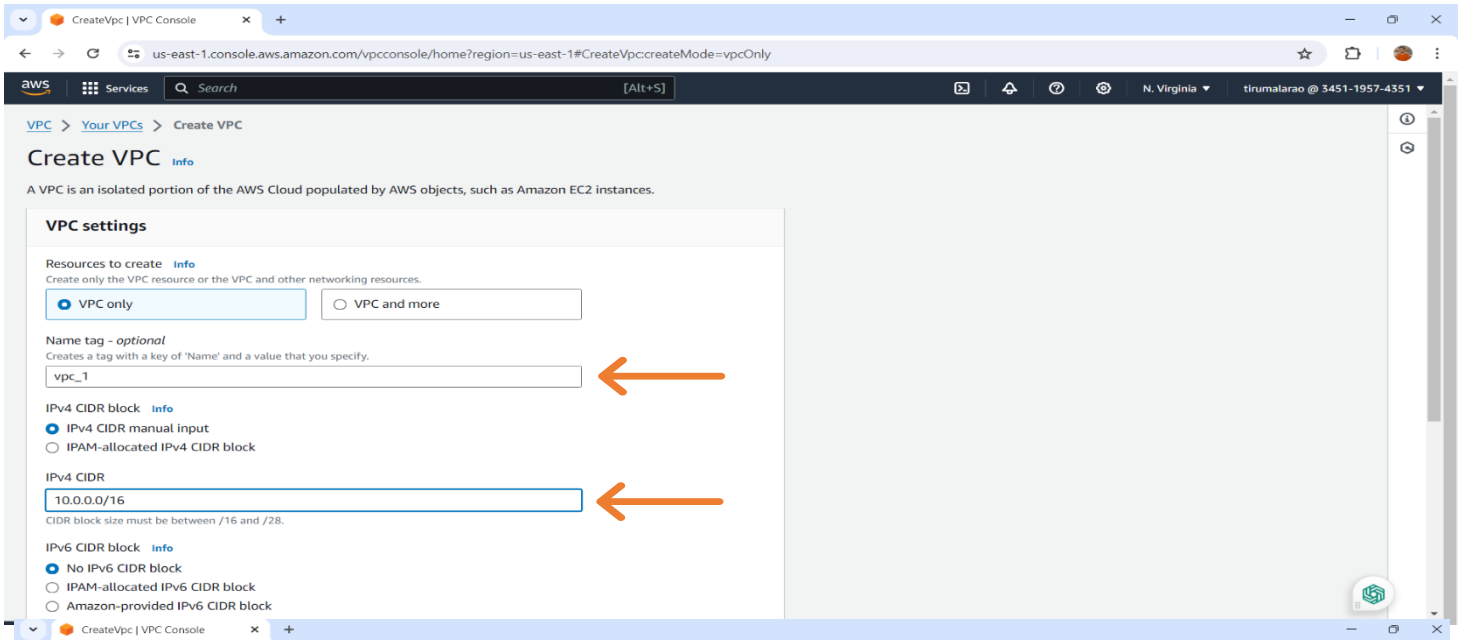
🌐 The first step of VPC peering is to create at least two VPCs. These VPCs should be in the same region but can be in different availability zones. You can create these VPS using the AWS console or CLI.

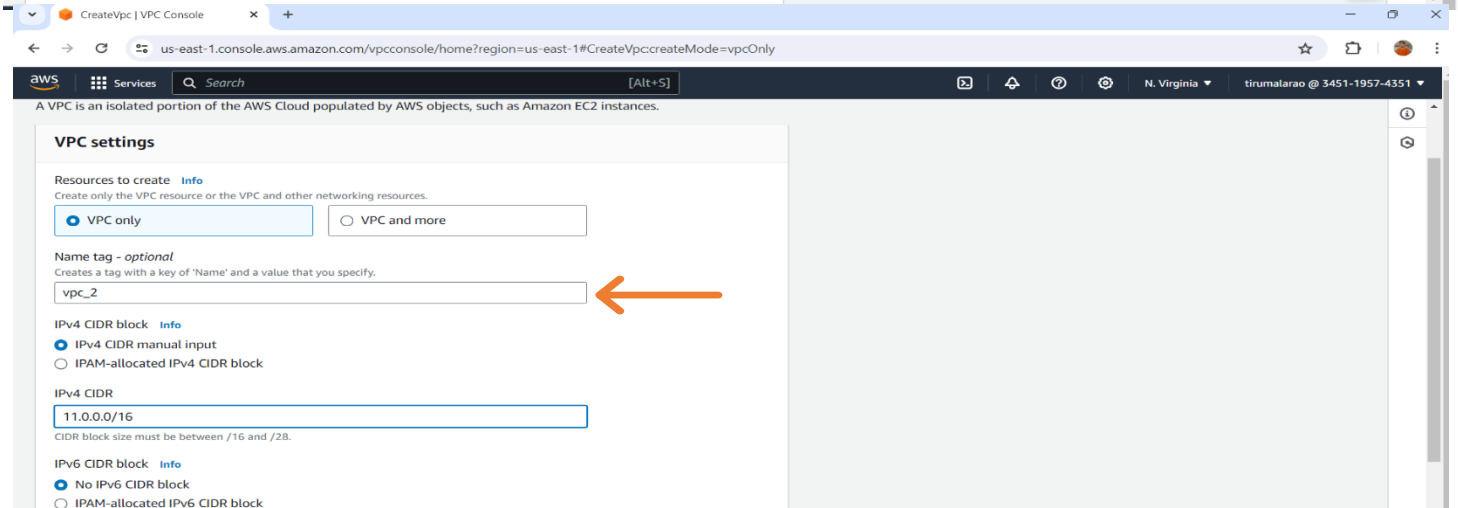
🌐 CIDR (Classless Inter-Domain Routing) is a method used to allocate IP addresses and route traffic in a network. In the context of a VPC (Virtual Private Cloud) in AWS (Amazon Web Services), CIDR is used to define the IP address range that the VPC will use.

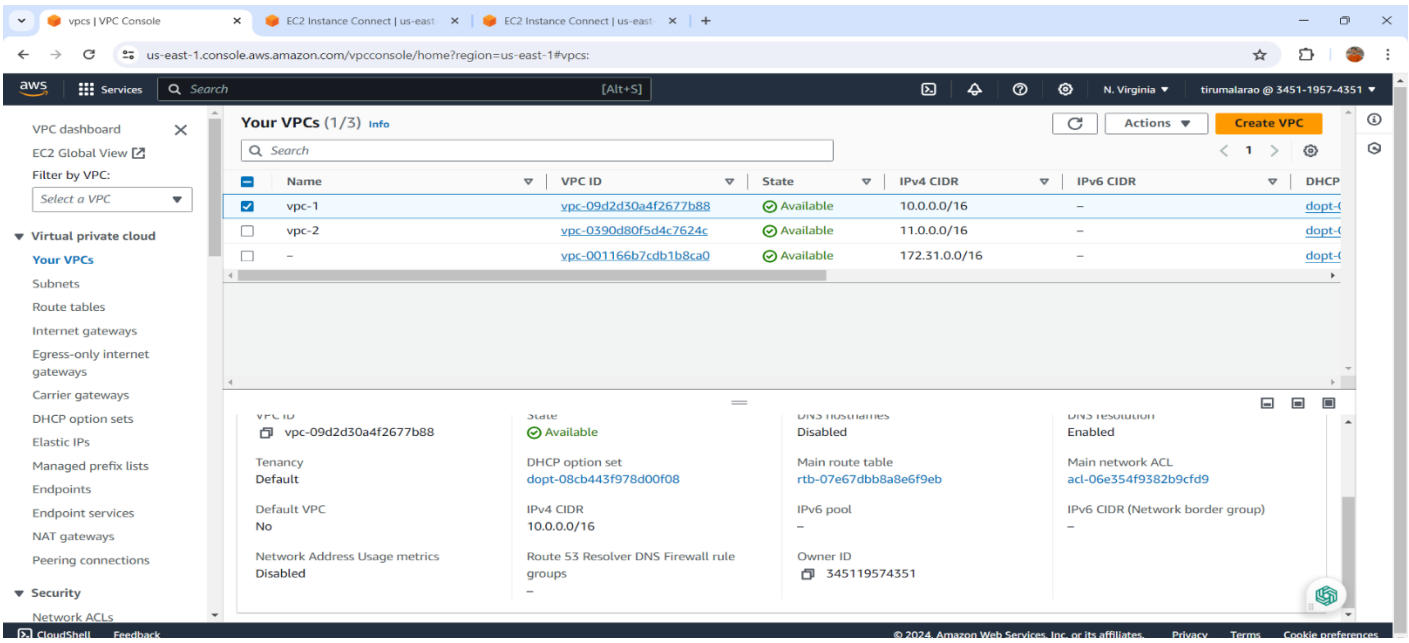
🌐 When creating a VPC in AWS, the user needs to specify a CIDR block that will be associated with the VPC. The CIDR block is a range of IP addresses expressed in CIDR notation, for example, 10.0.0.0/16. This means that the VPC will have a range of IP addresses from 10.0.0.0 to 10.0.255.255.

🌐 The CIDR block defines the maximum number of IP addresses that can be assigned to resources within the VPC. Once the CIDR block is defined, the user can create subnets within the VPC by dividing the CIDR block into smaller ranges. Each subnet can be associated with a different availability zone and can be used to launch instances or other resources.

🌐 CIDR notation also allows for efficient routing of traffic within the VPC and between the VPC and external networks. The VPC route table contains information on how to route traffic to and from different network addresses based on CIDR notation ranges. This allows for efficient use of network resources and optimal performance.







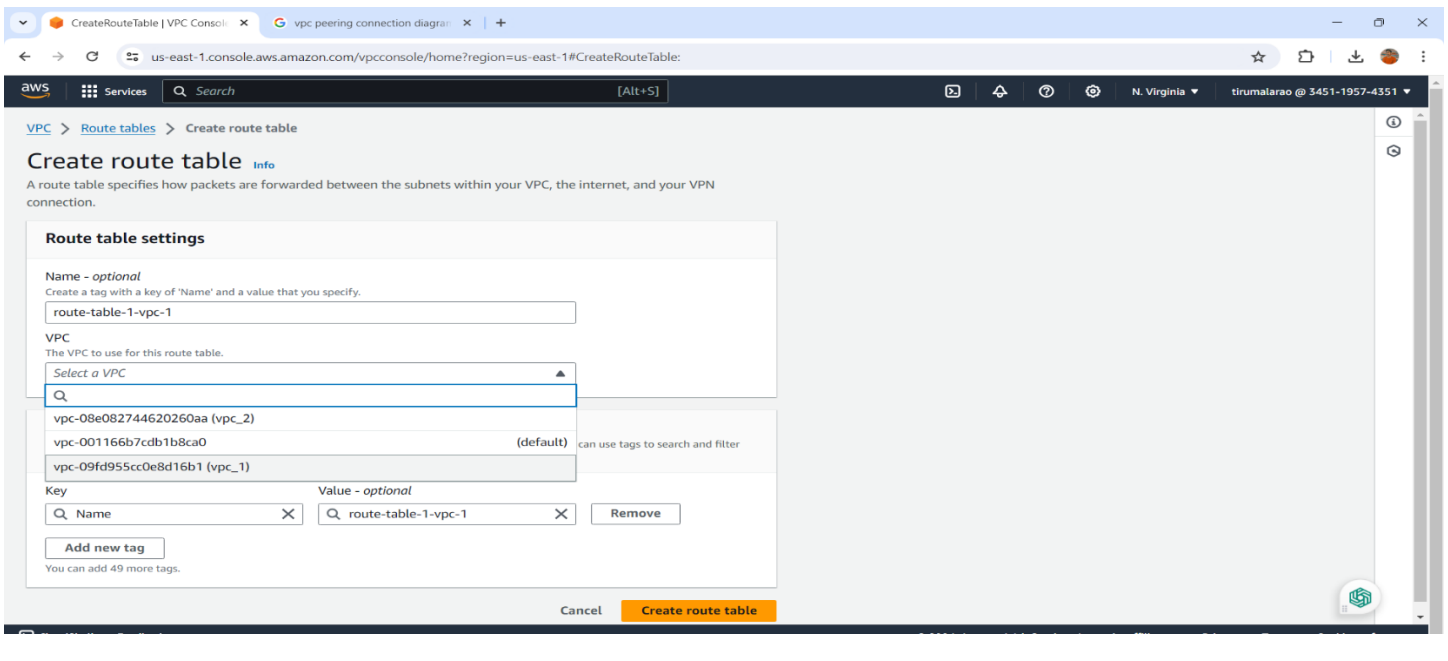
Step 2: Route table creation

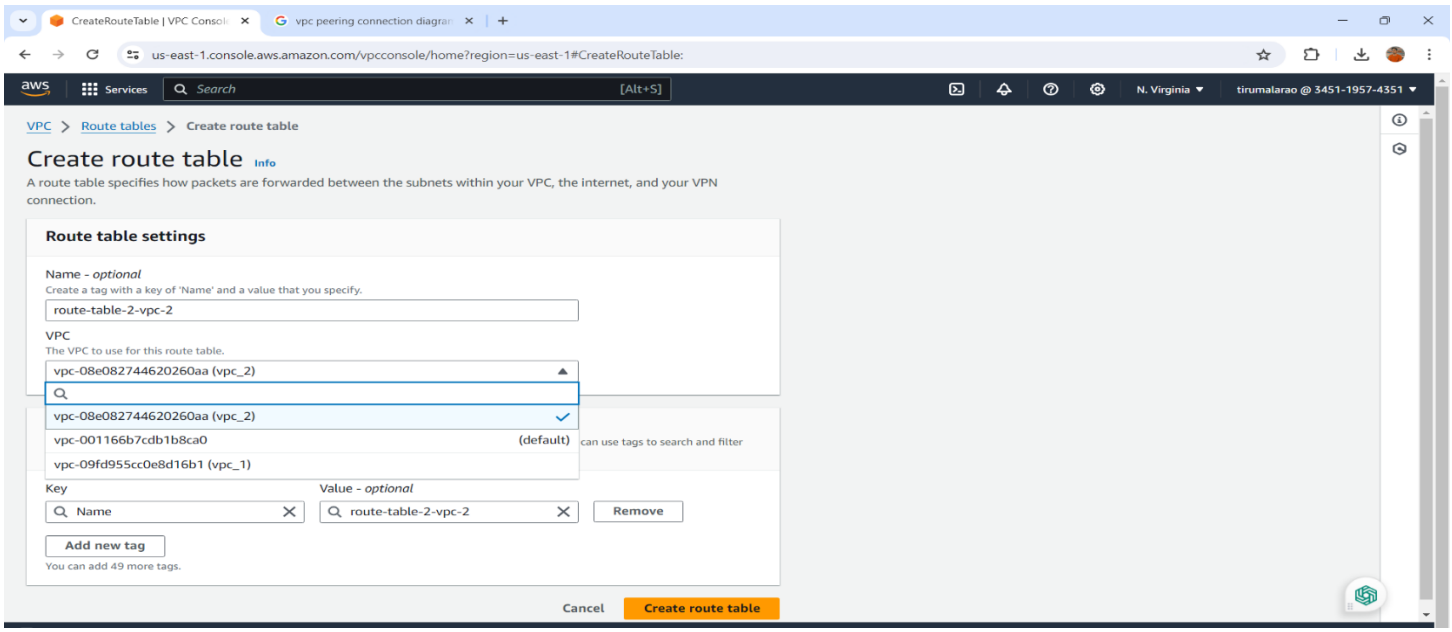
🌐 A route table in Amazon Web Services (AWS) is a key component of the networking functionality. A route table contains a set of rules called routes, which are used to determine where network traffic is directed.

🌐 When an instance in an AWS virtual private cloud (VPC) needs to send traffic to a destination, the VPC uses the route table associated with the subnet it's in to determine where to send the traffic.

🌐 Each route in a route table specifies a destination CIDR block and the target where traffic should be sent. Targets can be instances, gateways, or virtual private gateways.

🌐 Route tables can be associated with subnets, and a subnet can be associated with only one route table at a time. By default, each subnet in a VPC is associated with the main route table for the VPC, but you can create additional custom route tables and associate them with subnets as needed.



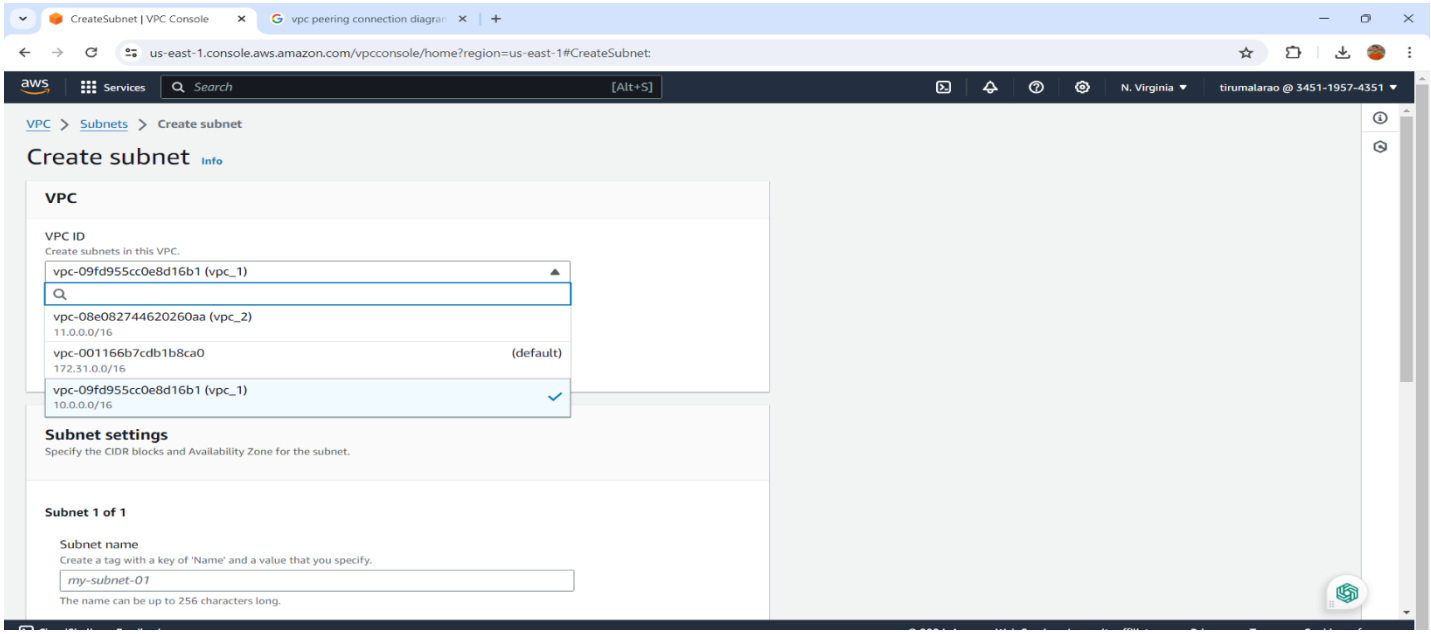


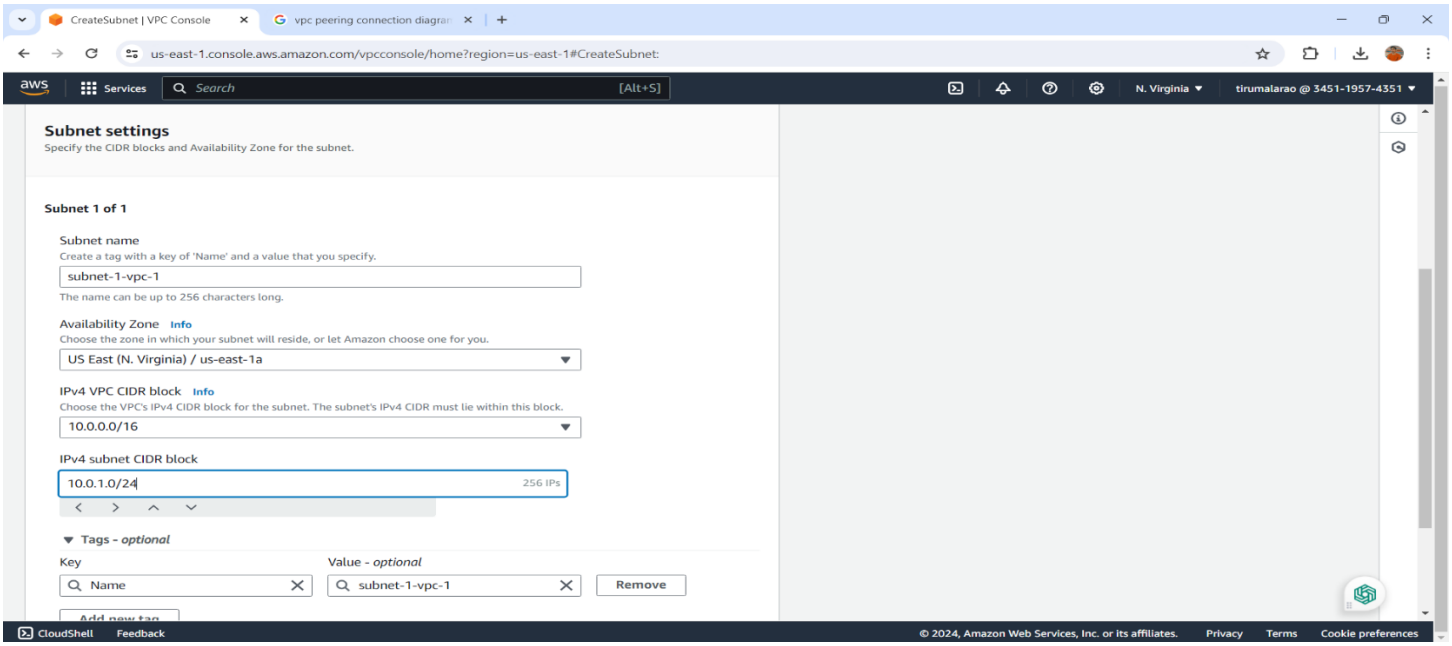
Step 3: Subnets

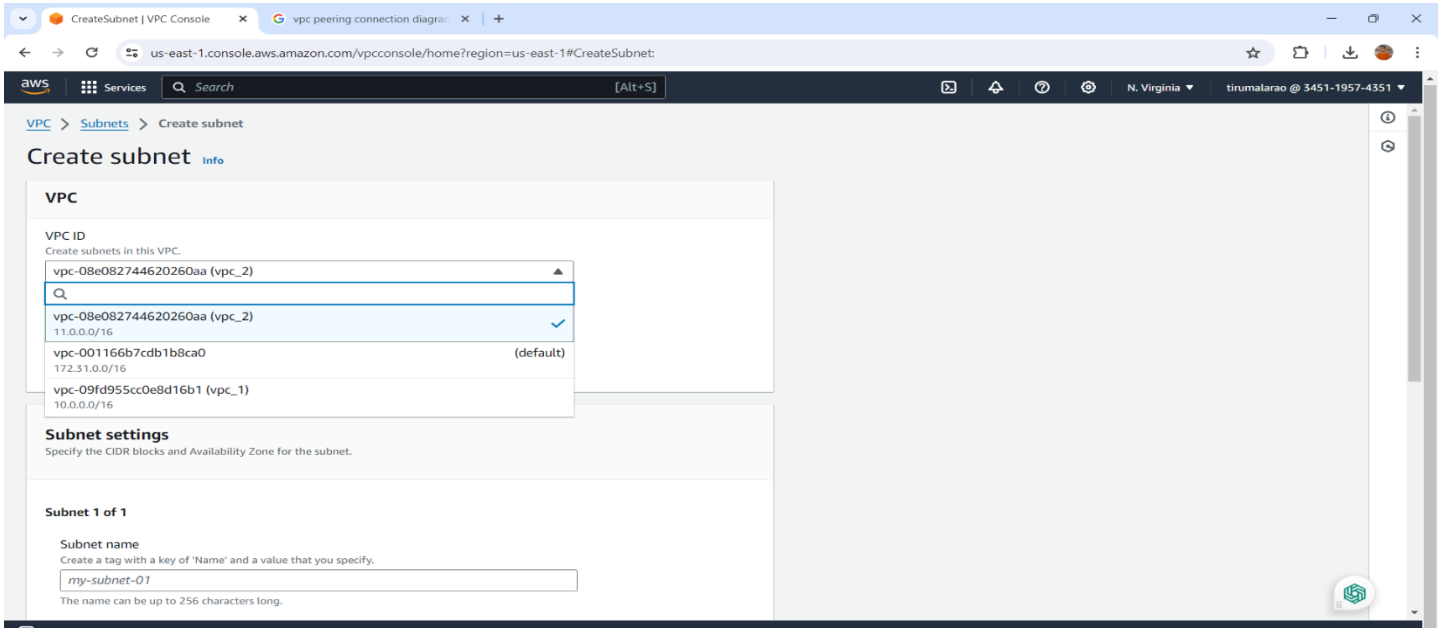
🌐 Subnet association in AWS refers to the process of associating a subnet with a particular resource or service. A subnet is a network partition that enables the segregation of resources within a virtual private cloud (VPC).

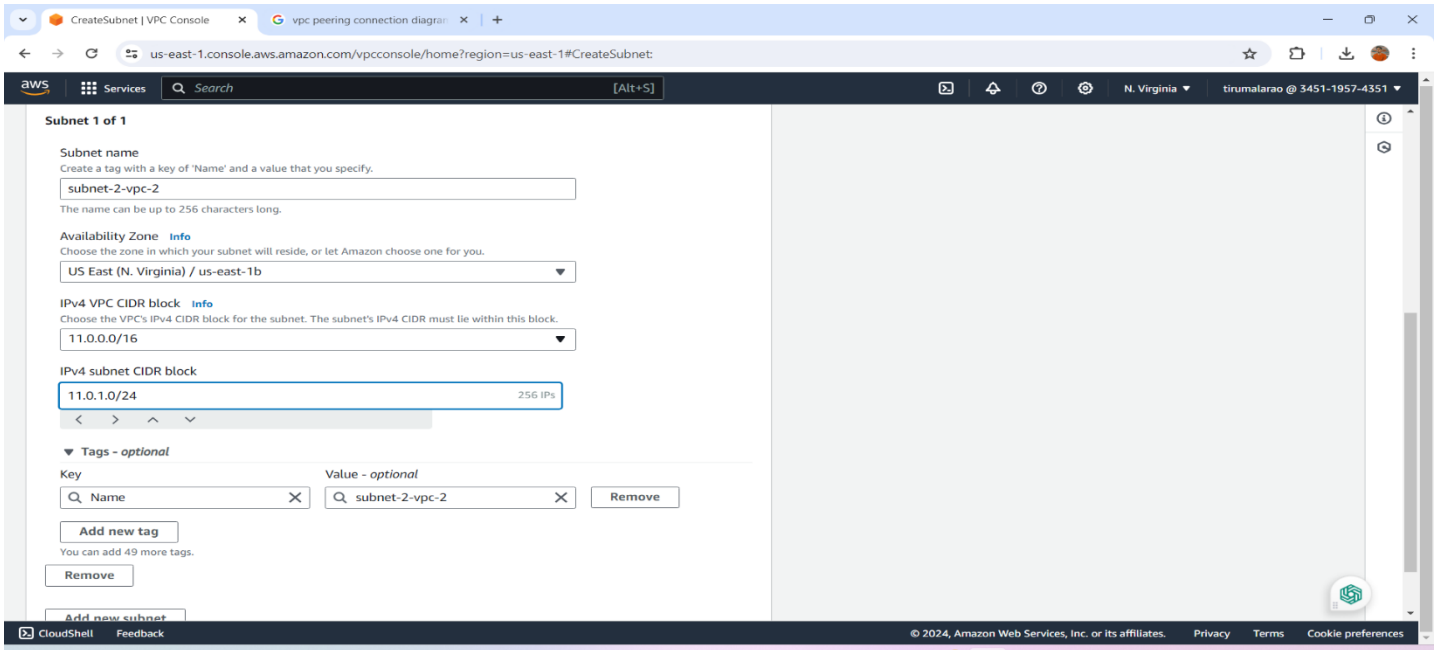
🌐 When creating resources within AWS, such as EC2 instances, a subnet can be associated with them to determine which IP addresses are reachable and to ensure that they operate within the defined network boundaries.

🌐 By associating a subnet with a resource, AWS ensures that it receives the correct IP configuration, and is assigned to the right availability zone, which enables easy access to other resources within the same subnet. Additionally, an AWS subnet can be linked to internet gateways (IGWs), virtual private gateways (VGWs), and other networking components to create and maintain a secure, scalable, and flexible network infrastructure.




CreateSubnet | VPC Console
vpc peering connection diagram
us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet:
Services
Search
[Alt+S]
N. Virginia
tirumalarao @ 3451-1957-4351
Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.
Subnet 1 of 1
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
subnet-1-vpc-1
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16
IPv4 subnet CIDR block
10.0.1.0/24
256 IPs
Tags - optional
Key
Value - optional
Name
subnet-1-vpc-1
Remove
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


CreateSubnet | VPC Console
vpc peering connection diagram
us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet:
Services
Search
[Alt+S]
N. Virginia
tirumalarao @ 3451-1957-4351
VPC > Subnets > Create subnet
Create subnet Info
VPC
VPC ID
Create subnets in this VPC.
vpc-08e082744620260aa (vpc_2)
vpc-08e082744620260aa (vpc_2)
11.0.0.0/16
vpc-001166b7cdb1b8ca0 (default)
172.31.0.0/16
vpc-09fd955cc0e8d16b1 (vpc_1)
10.0.0.0/16
Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.
Subnet 1 of 1
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
my-subnet-01
The name can be up to 256 characters long.


CreateSubnet | VPC Console
vpc peering connection diagram
us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateSubnet:
Services
Search
[Alt+S]
N. Virginia
tirumalarao @ 3451-1957-4351
Subnet 1 of 1
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
subnet-2-vpc-2
The name can be up to 256 characters long.
Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1b
IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
11.0.0.0/16
IPv4 subnet CIDR block
11.0.1.0/24
256 IPs
Tags - optional
Key
Value - optional
Name
subnet-2-vpc-2
Remove
Add new tag
You can add 49 more tags.
Remove
Add new subnet
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

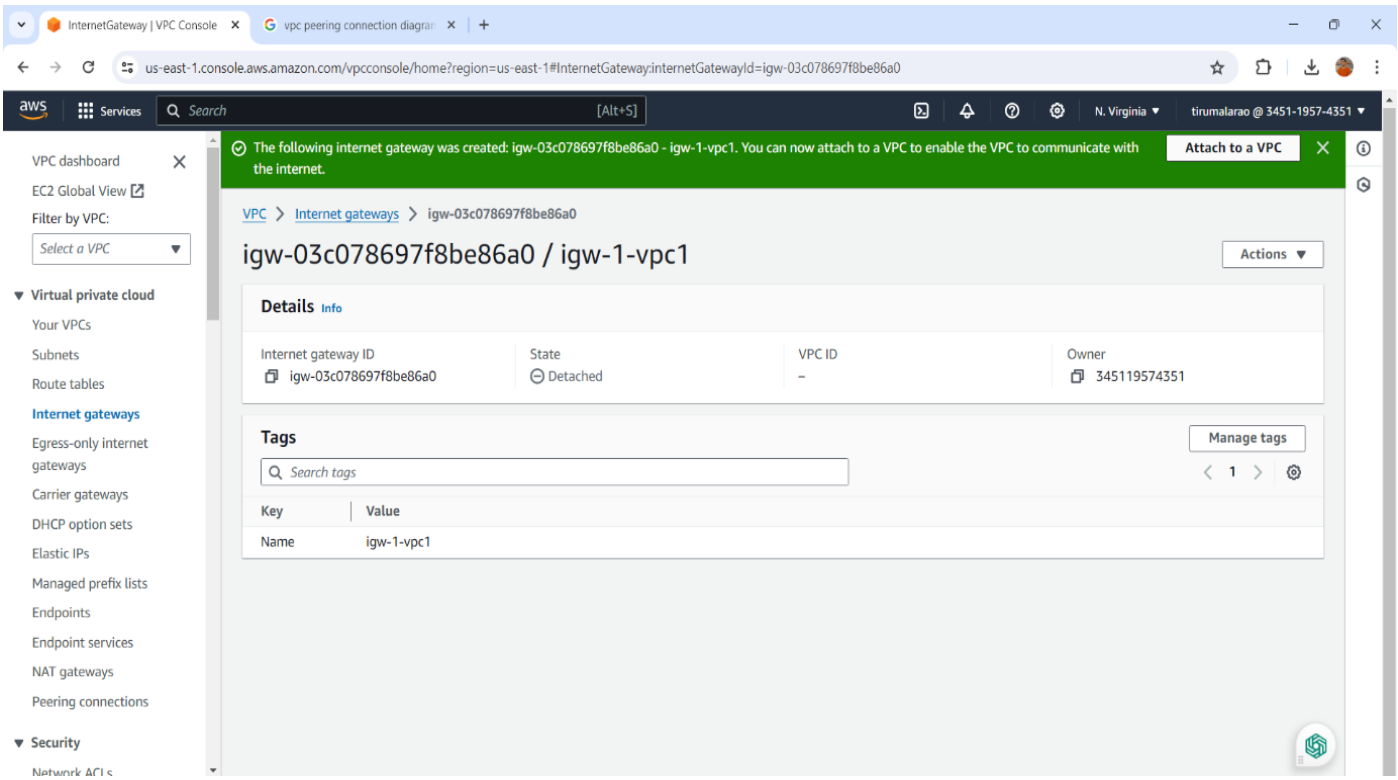
Step 4 : internet gateway

🌐 An internet gateway (IGW) in AWS is a horizontally scalable, redundant, and highly available virtual network component that serves as an essential entry and exit point between a Virtual Private Cloud (VPC) and the internet. It allows users to access the internet and provide a route for communication between the VPC and other resources or services outside the VPC, such as users, systems, applications, and third-party services.

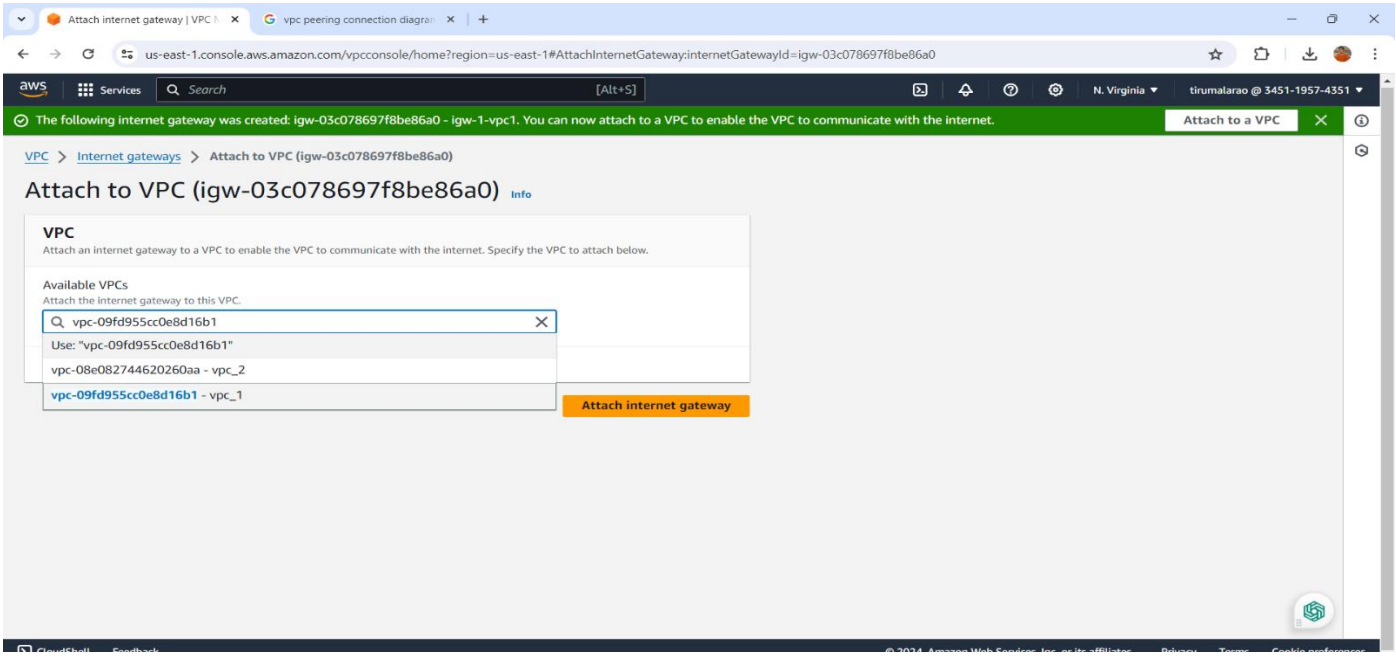
🌐 An internet gateway plays a vital role in routing traffic between the VPC and the internet by providing a public IP address for the instances running in the VPC, allowing them to communicate with the internet. The IGW also performs network address translation (NAT), which enables instances in the VPC to access the internet by translating their private IP addresses to public IP addresses.

🌐 Additionally, an internet gateway is integrated with Amazon Route 53, which enables users to set up DNS servers for their domain name system (DNS) resolution.

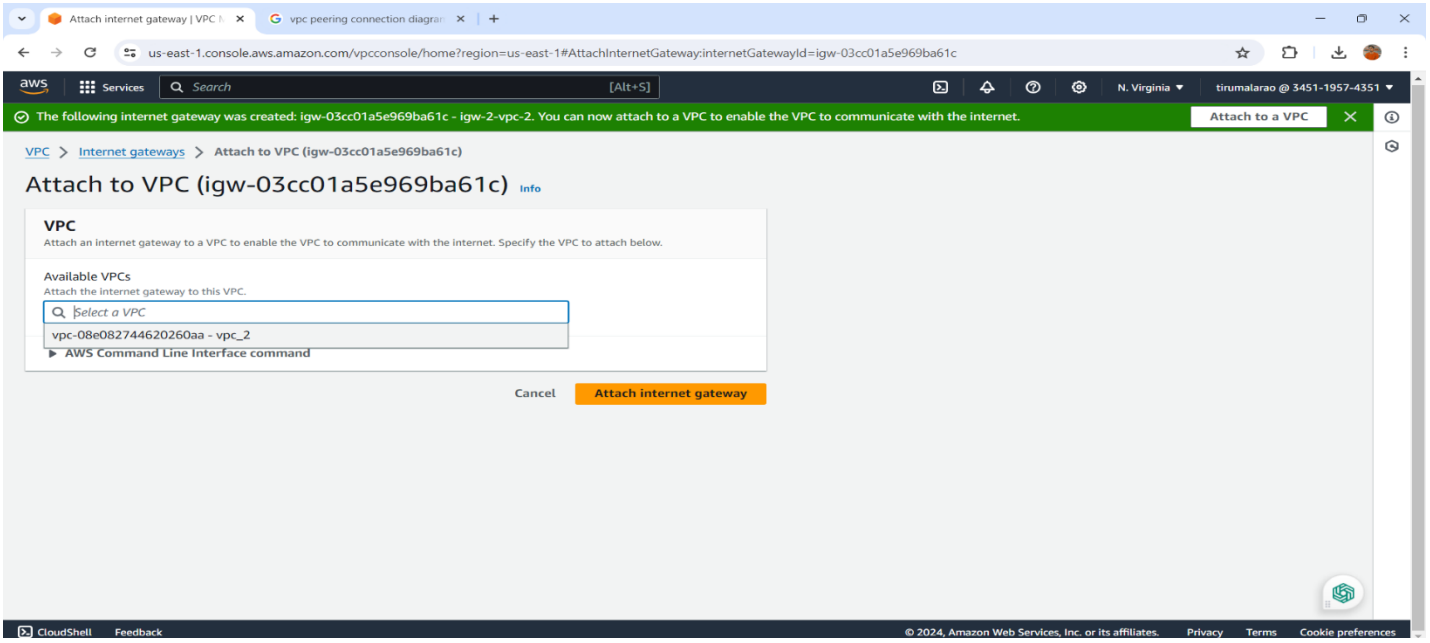
🌐 Overall, an internet gateway is an essential component in creating a secure, scalable, and highly available environment in AWS, allowing users to access the internet and communicate with other resources or services outside their VPC.

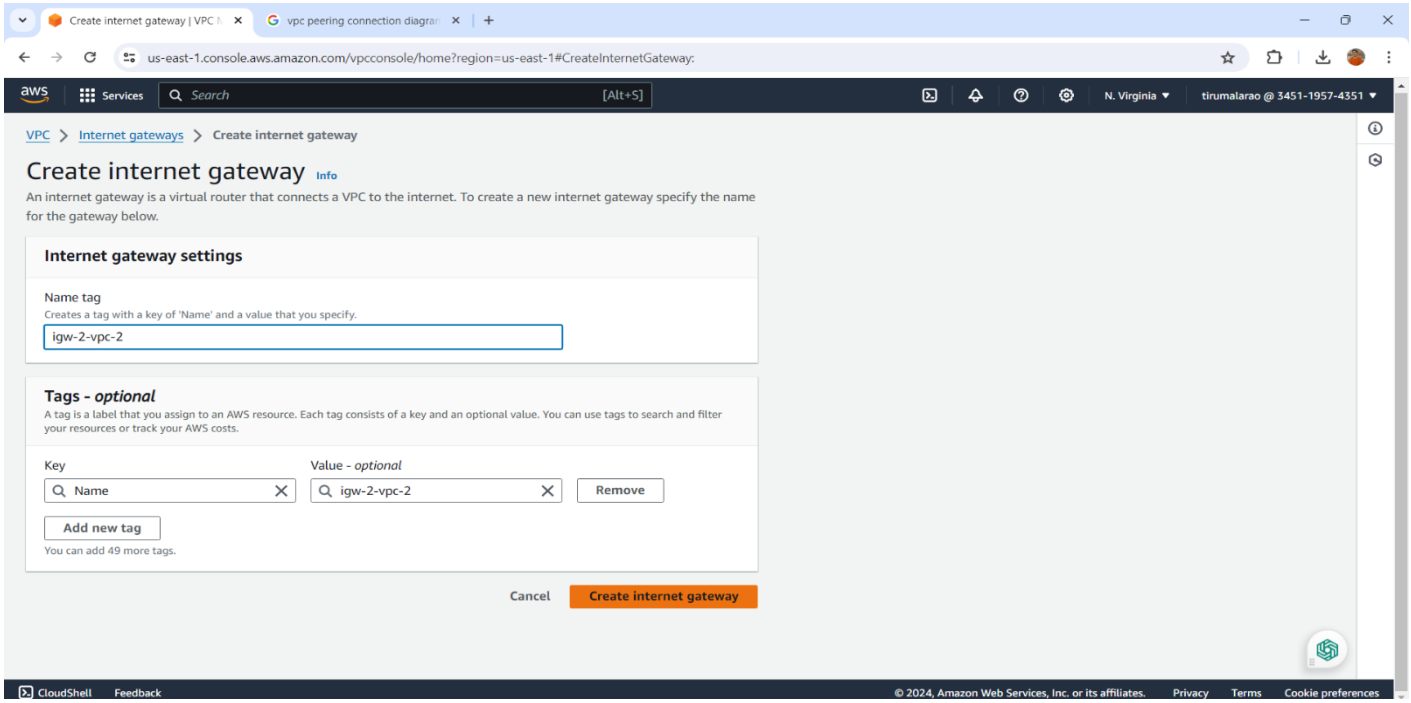


🌐 When an **Internet Gateway is attached** to a VPC, it provides a target for internet traffic destined for the instances in the VPC. The IGW acts as a gateway to the public internet, enabling instances to connect to services outside of the VPC and allowing inbound internet traffic to reach the instances inside the VPC. That's why it is essential to attach the Internet Gateway to a VPC in AWS.



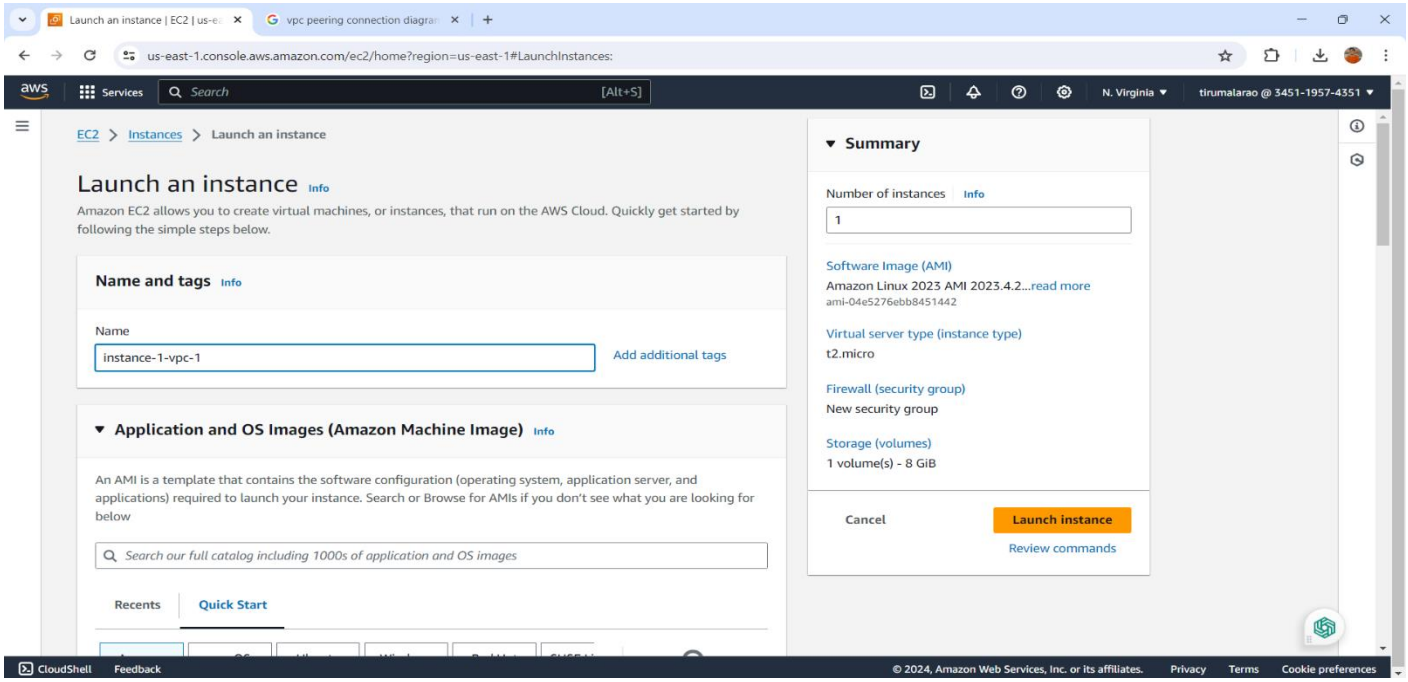
🌐 Here, we will create another Internet Gateway (IGW) and attach it to the second VPC."





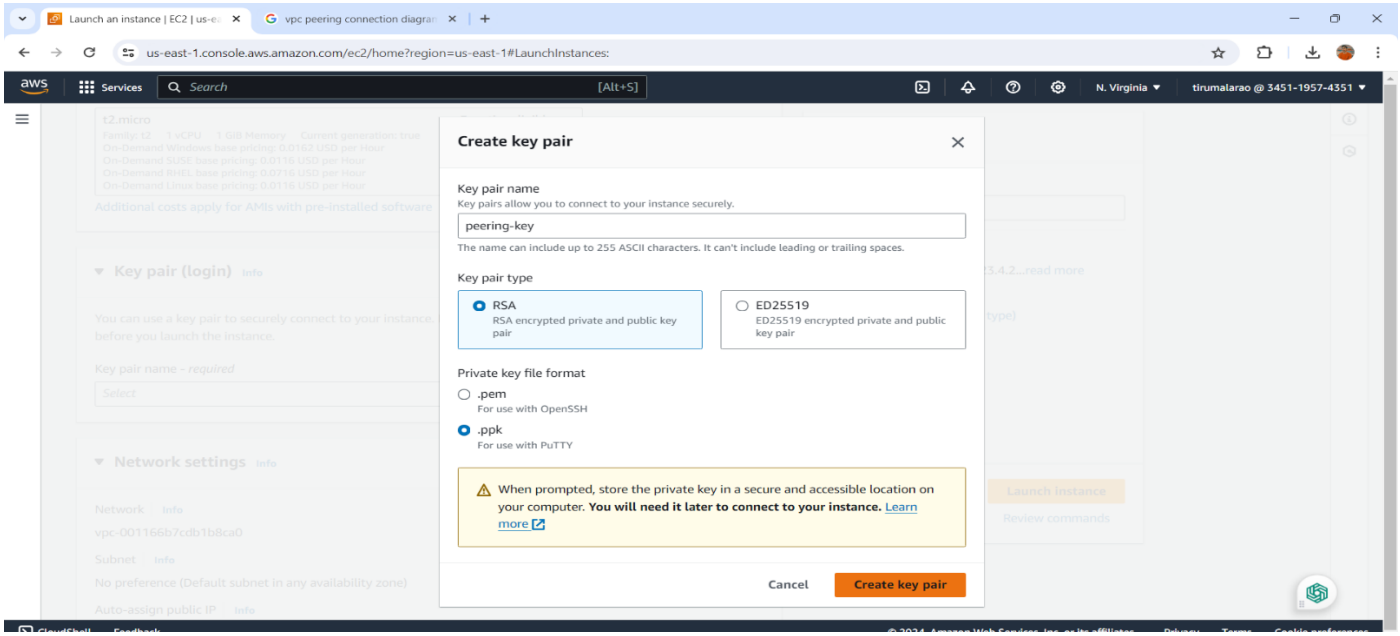
🌐 An instance in AWS refers to a virtual machine that can be used to run applications and services within the Amazon Web Services cloud computing platform. Instances can be launched on demand and can be configured with various configurations, including different operating systems, storage options, network settings, and security controls.

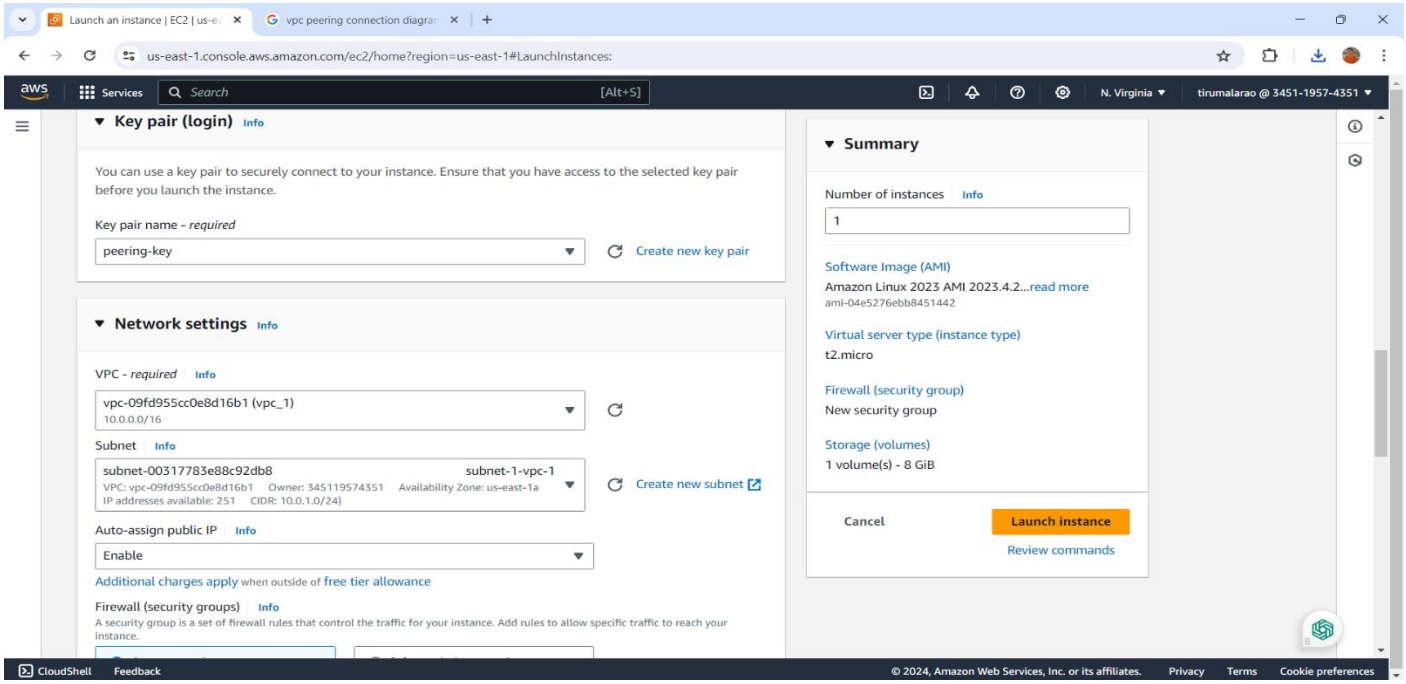
🌐 AWS provides a variety of instance types to suit different use cases, such as compute-intensive workloads, memory-intensive applications, and storage-intensive tasks. Customers can also customize instances with their own software and configurations to create a tailored environment that meets their specific needs.



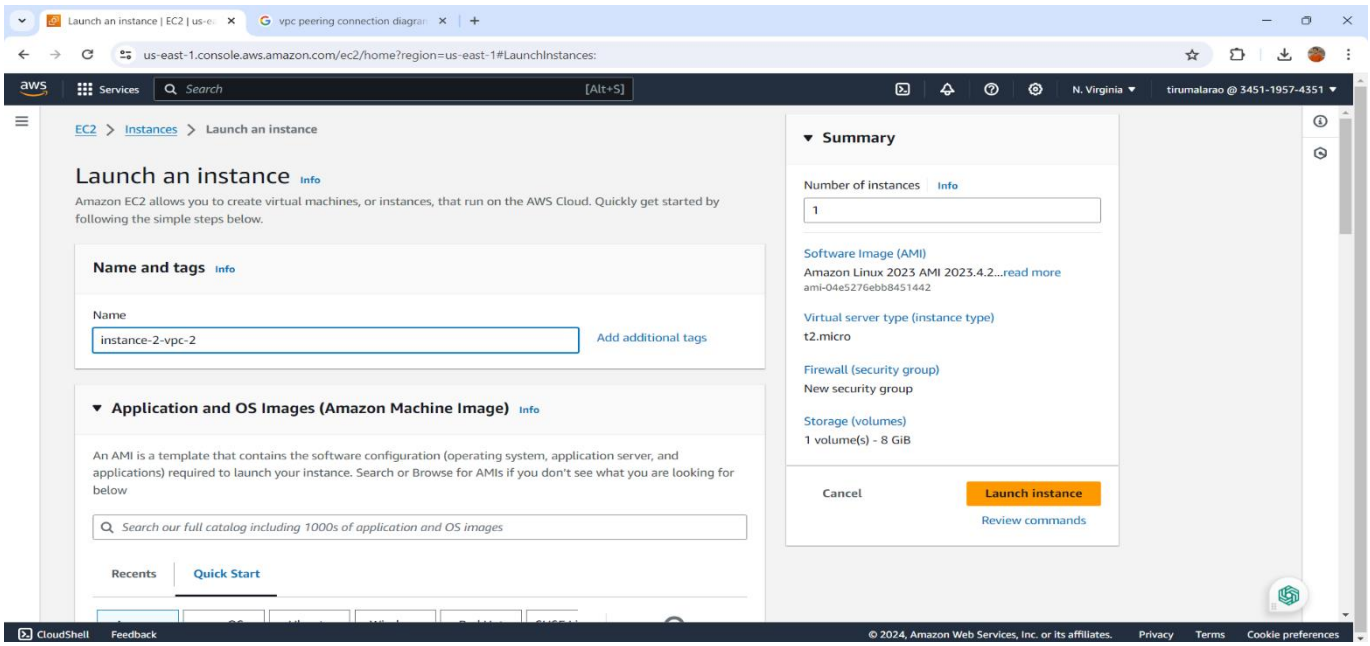
🌐 In AWS, a **key pair** is a set of two security credentials: a public key and a private key. The public key is used to encrypt data, while the private key is used to decrypt it. Key pairs are commonly used for secure authentication.

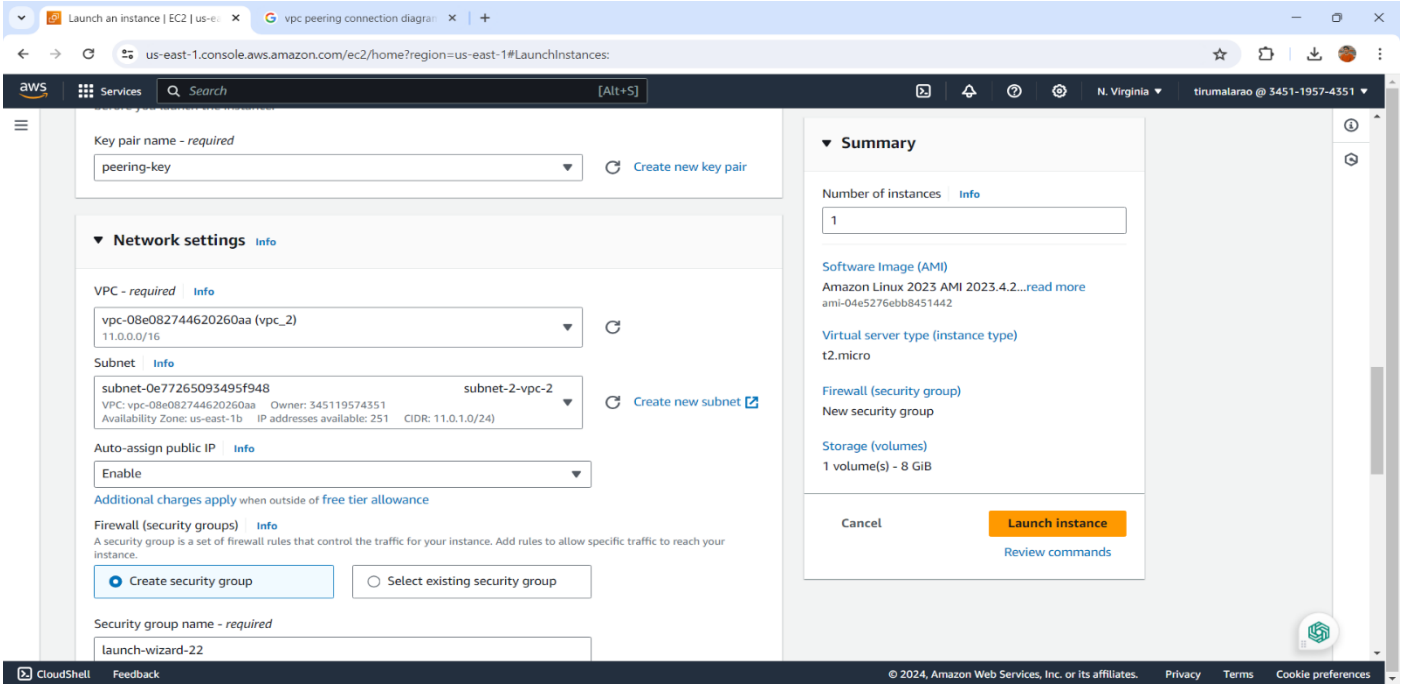
🌐 For example, to login to an EC2 instance using SSH. In AWS, key pairs can be created and managed through the EC2 Management Console or through the AWS CLI. It is important to keep the private key secure and not share it with anyone.





🌐 Here we will create a second instance for a second VPC



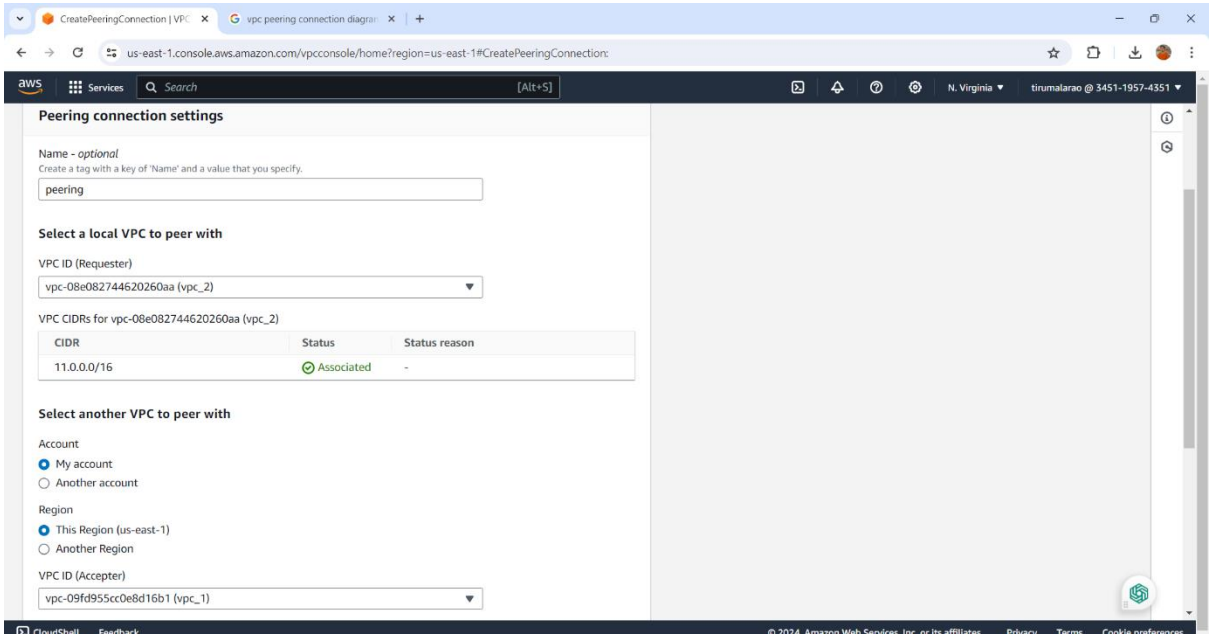


Step 5: Peering connection

In Amazon Web Services (AWS), a peering connection is a networking connection that allows two virtual private clouds (VPCs) to communicate with each other privately over the AWS network.

🌐 Peering connections can be established between two VPCs in the same AWS region or in different regions, as long as both VPCs are owned by the same AWS account.

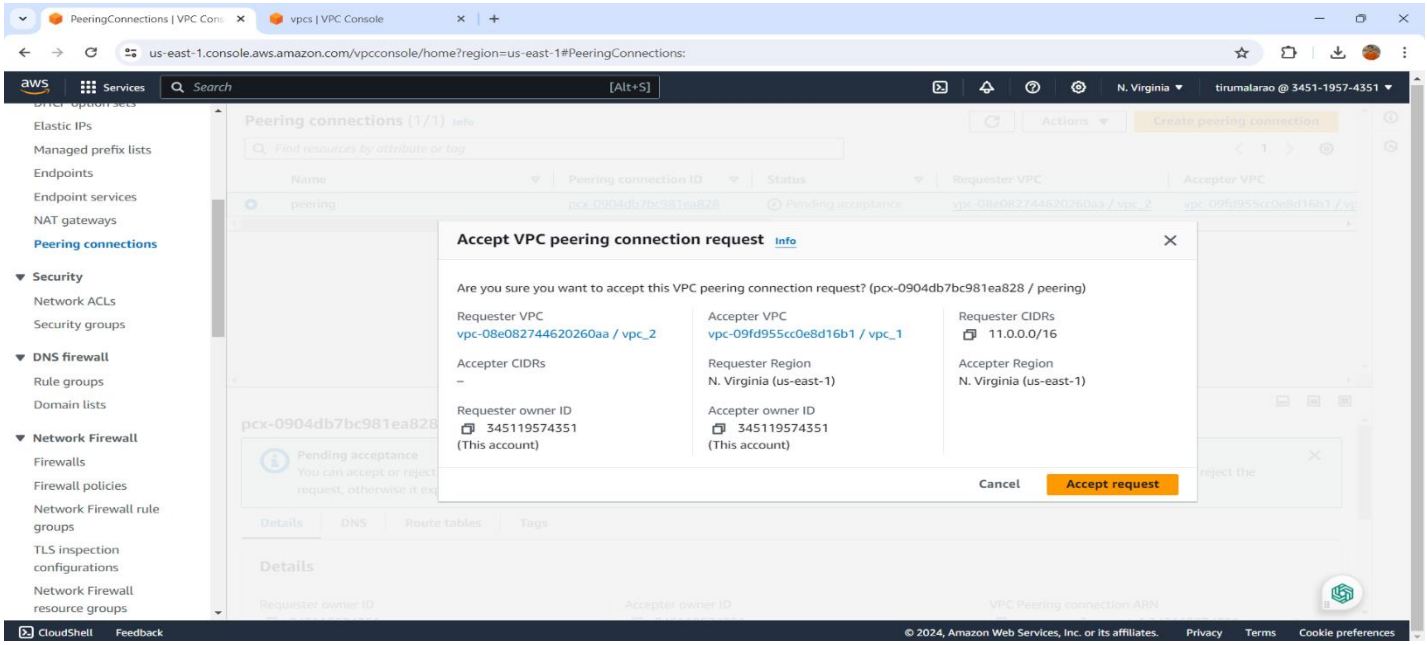
🌐 Peering connections can improve the performance, security, and availability of applications that are deployed across different VPCs. They allow for faster data transfer and reduce the risk of data breaches since communication between VPCs is done over a private connection rather than the internet.



A peering connection accepter in AWS is an entity that approves or accepts a peering connection request. A peering connection is a direct network connection between two VPCs (Virtual Private Clouds) or between a VPC and an external network. When a peering connection request is initiated, it must be accepted by the owner of the requested VPC to establish the connection.

🌐 The accepter can be the owner of the requested VPC or a delegated AWS account with the appropriate permissions. Once the accepter approves the request, the peering connection is established, and resources can communicate with each other across the connection using private IP addresses.

🌐 In AWS, the peering connection accepter can approve the request using the AWS Management Console, the AWS CLI (Command Line Interface), or the AWS SDKs (Software Development Kits). The accepter must ensure that the VPC routing tables are properly configured to enable appropriate traffic flow across the peering connection.

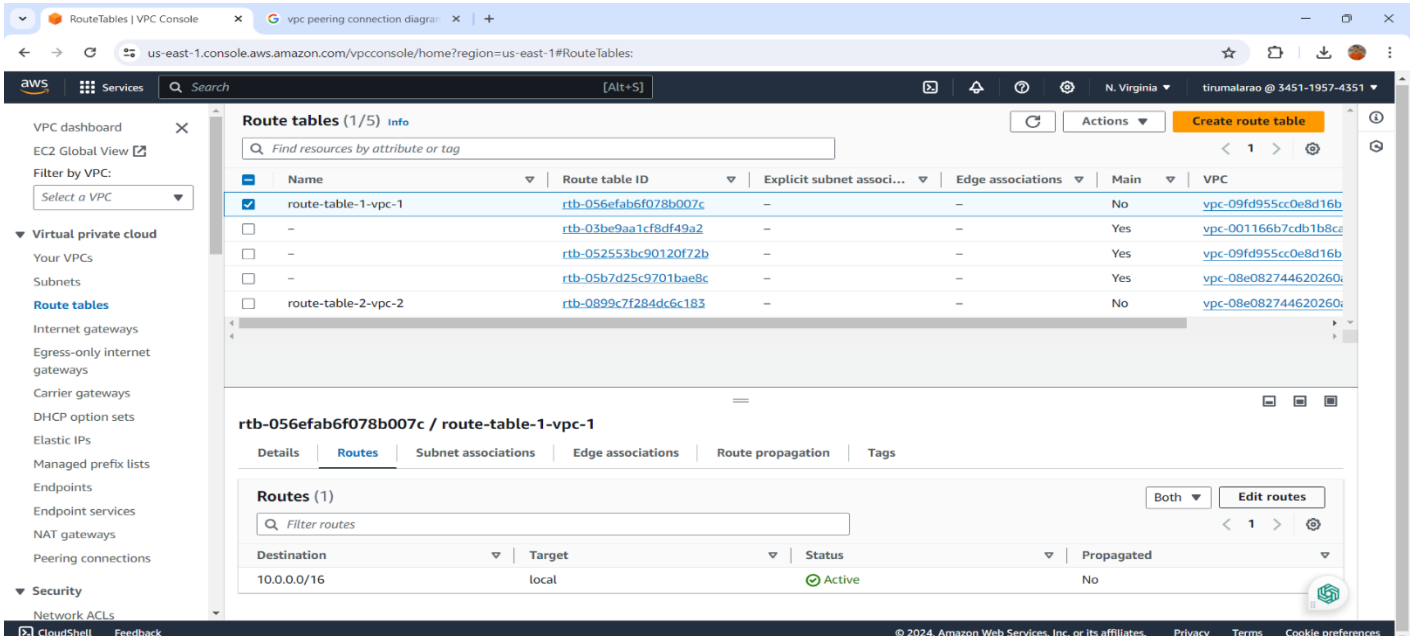


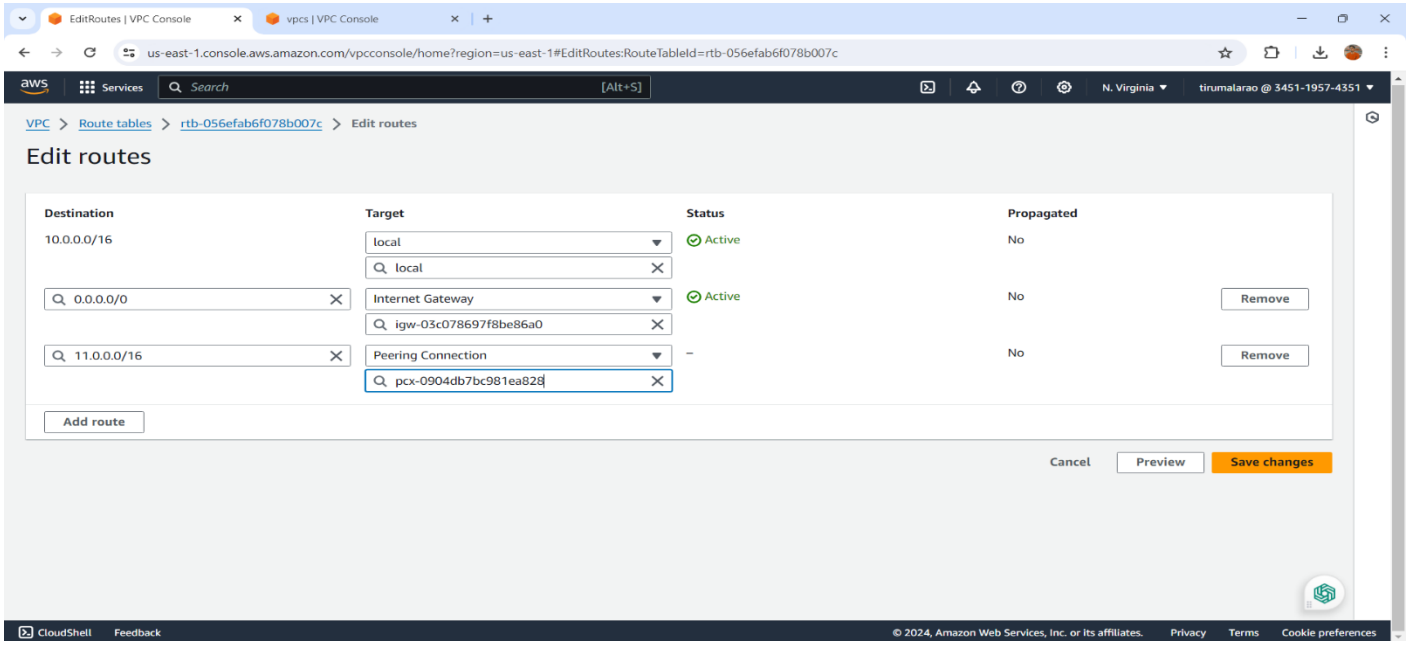
Editing routes is essential for managing and maintaining AWS network configurations, ensuring that traffic flows efficiently and securely between different components.

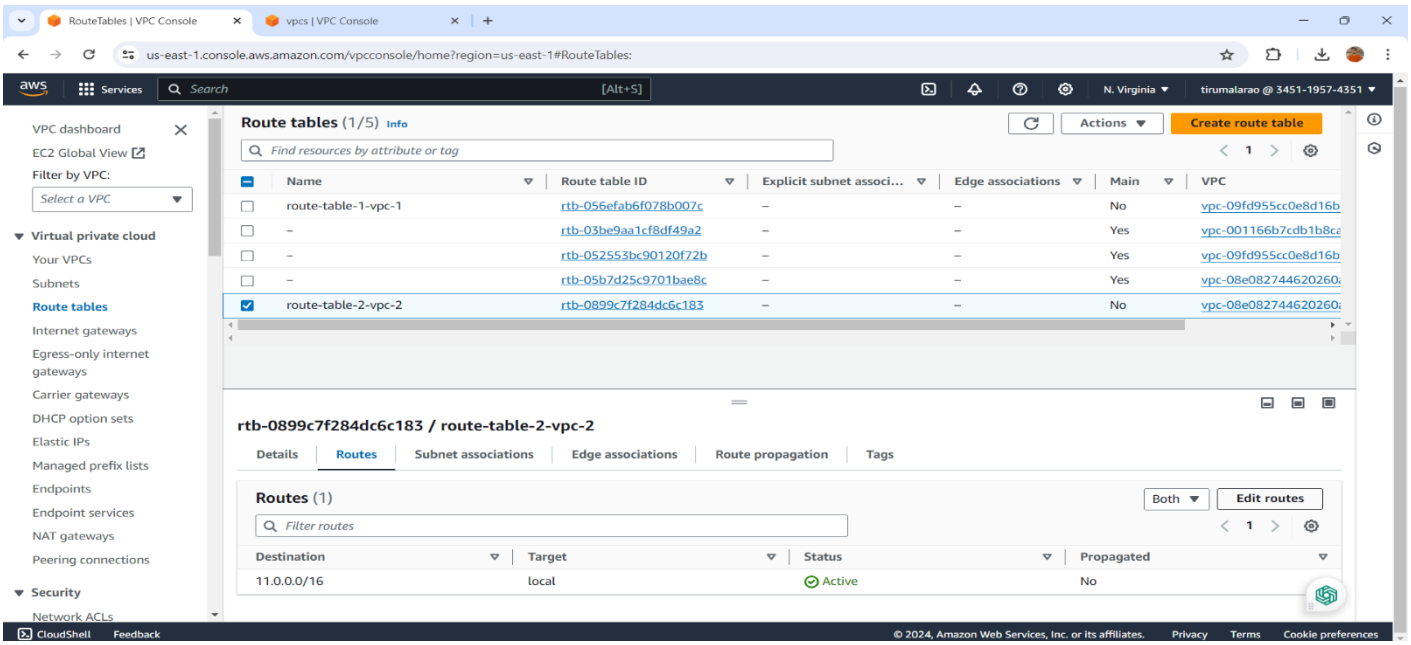
🌐 Enable routing to different destinations including specific IP addresses or IP address ranges, as well as routing through virtual private gateways, VPN connections, and Direct Connect gateways.

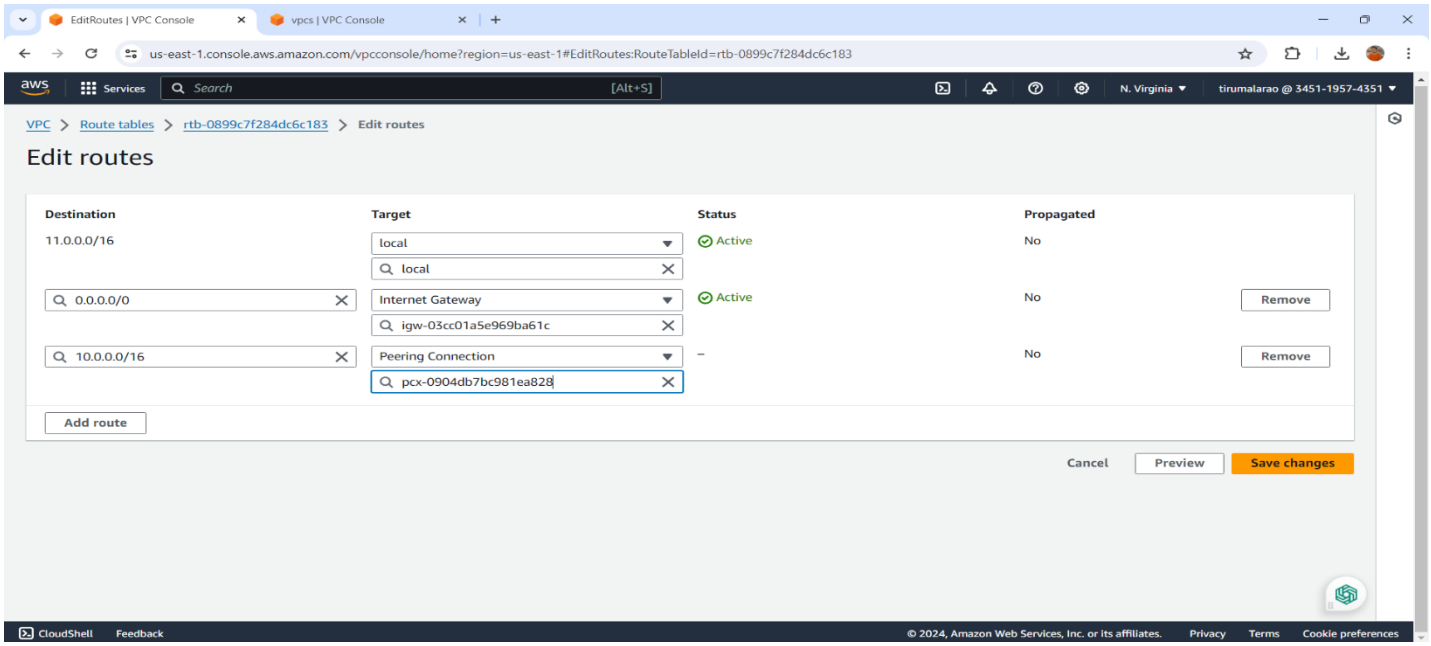
🌐 Edit routes to redirect traffic in the event of an outage or to optimize network performance.

Establish custom routes for specific scenarios, such as sending traffic to specific internet gateways or peering connections.




EditRoutes | VPC Console
vpcs | VPC Console
us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#EditRoutes:RouteTableId=rtb-056efab6f078b007c
Services
Search
[Alt+S]
N. Virginia
tirumalarao @ 3451-1957-4351
VPC > Route tables > rtb-056efab6f078b007c > Edit routes
Edit routes
Destination
Target
Status
Propagated
10.0.0.0/16
local
Active
No
local
0.0.0.0/0
Internet Gateway
Active
No
Remove
igw-03c078697f8be86a0
11.0.0.0/16
Peering Connection
–
No
Remove
pcx-0904db7bc981ea828
Add route
Cancel
Preview
Save changes
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


RouteTables | VPC Console
vpcs | VPC Console
us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTables:
Services
Search
[Alt+S]
N. Virginia
tirumalarao @ 3451-1957-4351
VPC dashboard
EC2 Global View
Filter by VPC:
Select a 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
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Network ACLs
Route tables (1/5) Info
Actions
Create route table
Find resources by attribute or tag
Name
Route table ID
Explicit subnet associ...
Edge associations
Main
VPC
route-table-1-vpc-1
rtb-056efab6f078b007c
No
vpc-09fd955cc0e8d16b
rtb-03be9aa1cf8df49a2
Yes
vpc-001166b7cdb1b8ca
rtb-052553bc90120f72b
vpc-09fd955cc0e8d16b
rtb-05b7d25c9701bae8c
vpc-08e082744620260a
route-table-2-vpc-2
rtb-0899c7f284dc6c183
vpc-08e082744620260a
rtb-0899c7f284dc6c183 / route-table-2-vpc-2
Details
Routes
Subnet associations
Edge associations
Route propagation
Tags
Routes (1)
Both
Edit routes
Filter routes
Destination
Target
Status
Propagated
11.0.0.0/16
local
Active
No
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


EditRoutes | VPC Console
vpcs | VPC Console
us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#EditRoutes:RouteTableId=rtb-0899c7f284dc6c183
Services
Search
[Alt+S]
N. Virginia
tirumalarao @ 3451-1957-4351
VPC > Route tables > rtb-0899c7f284dc6c183 > Edit routes
Edit routes
Destination
Target
Status
Propagated
11.0.0.0/16
local
Active
No
local
0.0.0.0/0
Internet Gateway
Active
No
Remove
igw-03cc01a5e969ba61c
10.0.0.0/16
Peering Connection
–
No
Remove
pcx-0904db7bc981ea828
Add route
Cancel
Preview
Save changes
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

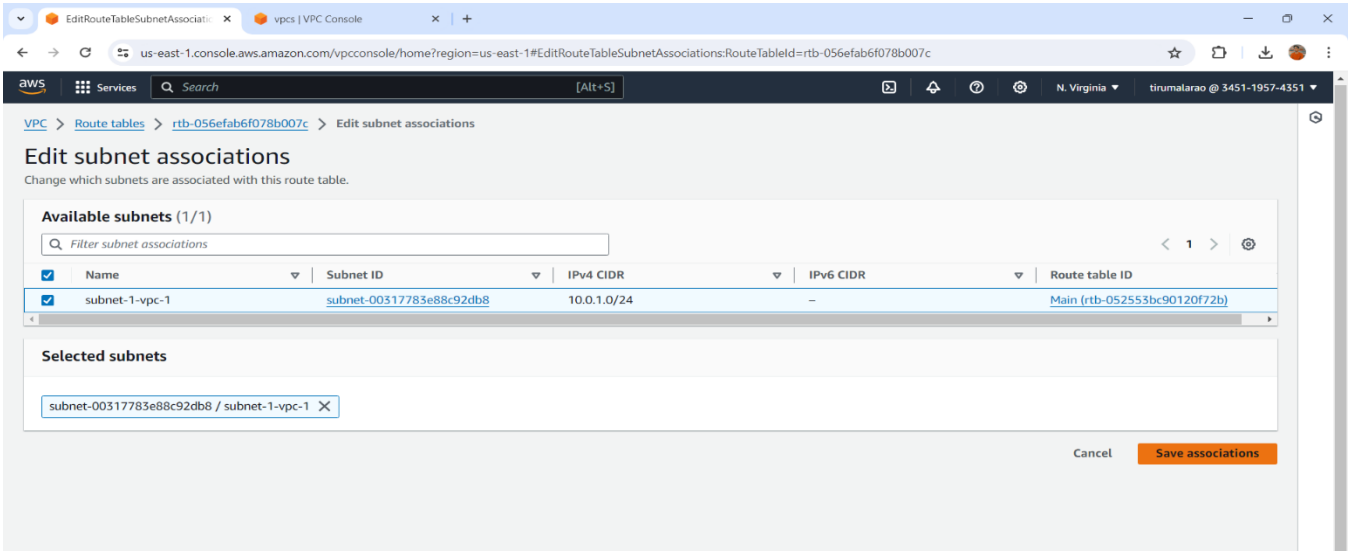
Subnet association In AWS, subnet association refers to the process of assigning a subnet to a specific Amazon VPC (Virtual Private Cloud) or network interface. This association allows instances in the VPC to communicate with resources outside the VPC or to connect with resources inside the VPC using a private IP address.

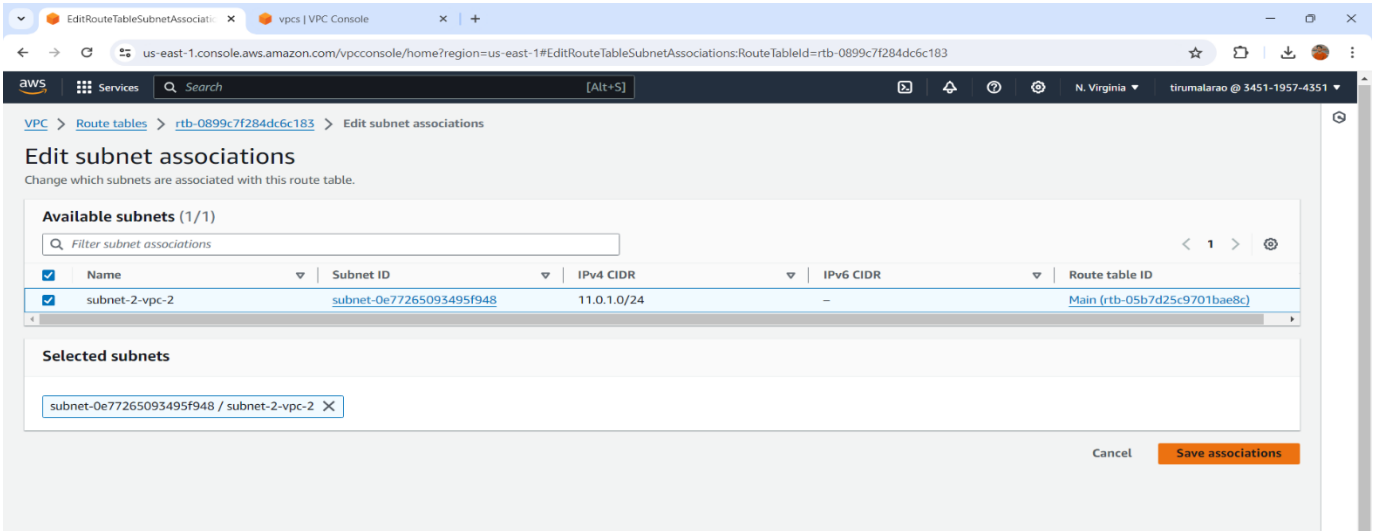
🌐 To associate a subnet with a VPC, you can use the Amazon VPC console or the AWS Command Line Interface (CLI). You can also associate subnets to network interfaces of instances using the console, CLI, or SDK.

🌐 When creating a VPC, you can create multiple subnets in different Availability Zones (AZs) to achieve high availability and fault tolerance. You can associate a subnet with a specific AZ to ensure that your instances are launched in the same AZ as the subnet.

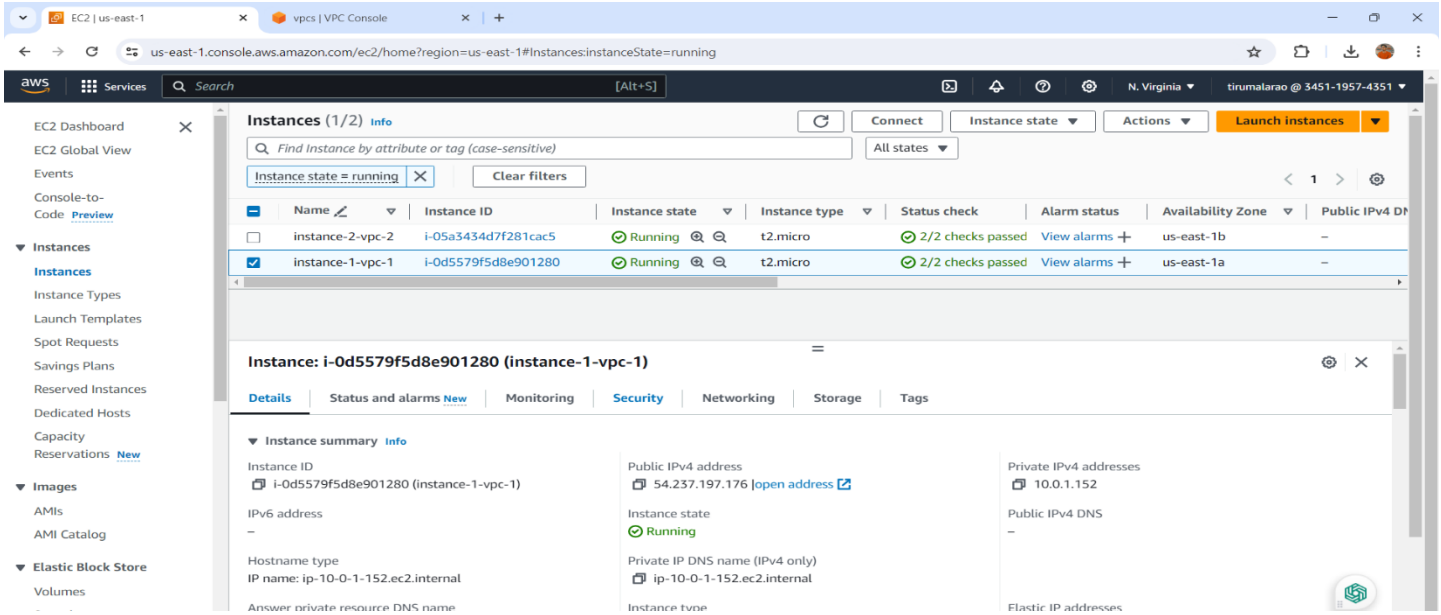
🌐 In addition, you can use subnets to control traffic in and out of your VPC using Network Access Control Lists (NACLs) and Security Groups (SGs). NACLs provide a stateless, rule-based filter to control traffic at the subnet level, while SGs provide a stateful, instance-level firewall at the network interface level.

🌐 Overall, subnet association is a critical aspect of designing and deploying secure and scalable VPC architectures in AWS.



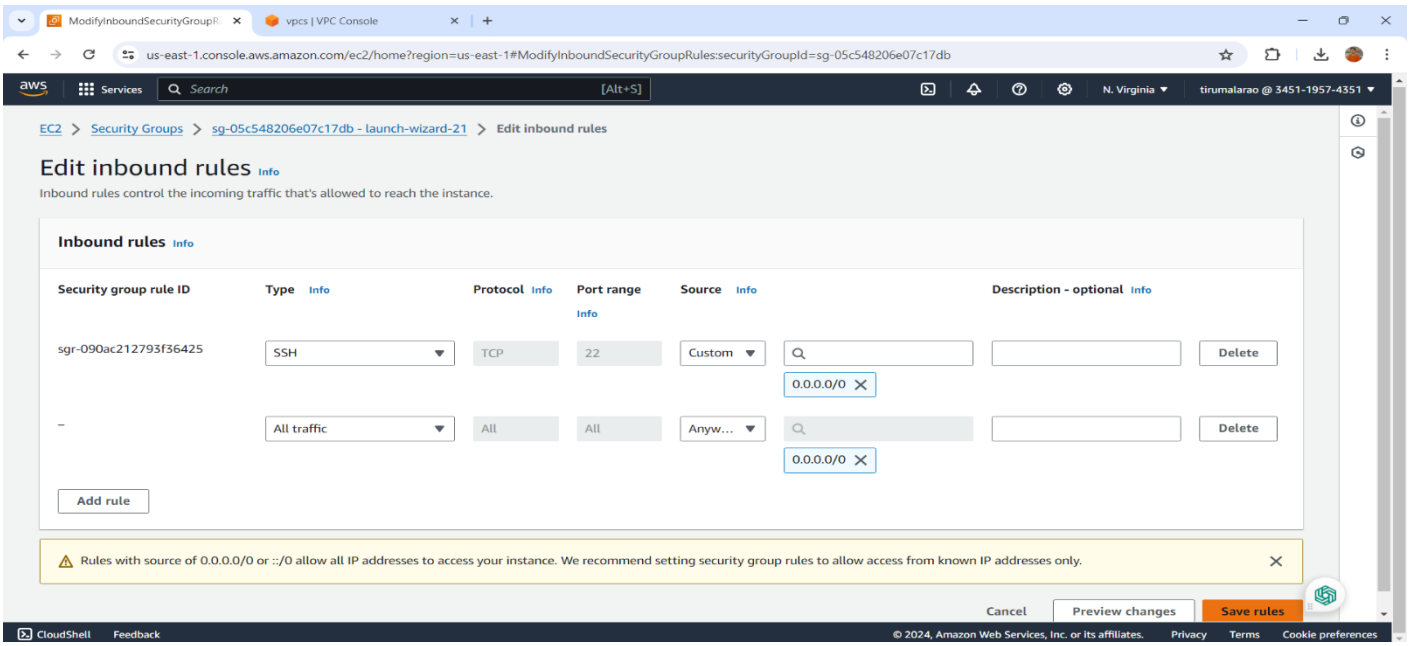


Security group in AWS is a virtual firewall that controls incoming and outgoing traffic for one or more instances. It acts as a piece of virtual networking infrastructure that can be used to filter and restrict traffic to and from an EC2 instance. It allows you to specify inbound and outbound traffic rules that allow network traffic from specific IP addresses, ports, or protocols. Security groups can be associated with one or more instances, and changes to a security group apply immediately to all instances associated with that group.



Editing inbound rules is necessary because it allows you to configure access to your instance according to your requirements. For example, you can add a new rule to allow traffic from a specific IP address or range, to restrict malicious traffic to your EC2 instance, or to allow users to access your application.

Note : We need to edit the inbound rules on both of the two servers.



Step 6 : Peering connection testing

Command :-

Ping <Second server IP address> ->in first server

Ping <First server IP address> ->in second server