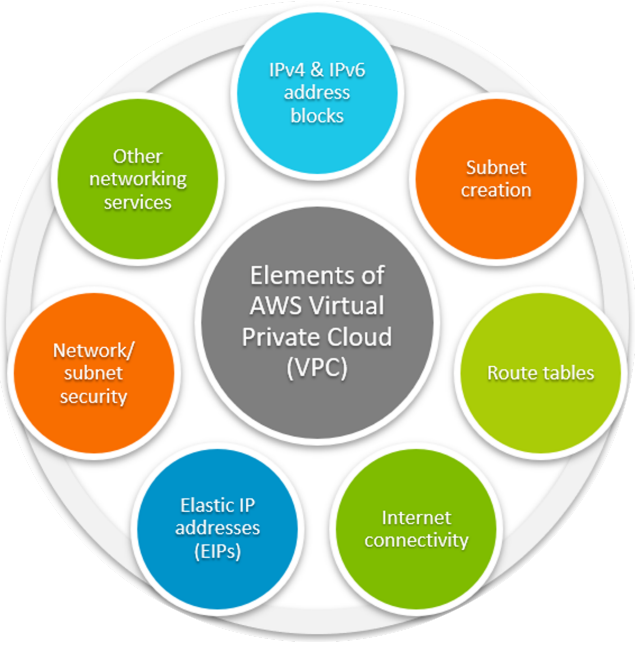
VPC Creation and setup.

👋 Welcome to my-blog!

VPC

VPC stands for **Virtual Private Cloud**. It's a virtual network dedicated to your AWS account. Just like a traditional network in a data center, you can **control** the **virtual networking environment**, including your **own IP address range**, **subnets**, **route tables**, and **network gateways.**

in Simple ways Imagine a secure, isolated section of the cloud where you can run your AWS resources. It's like having your own private piece of the internet where you decide who can access it and how they can interact with your services.

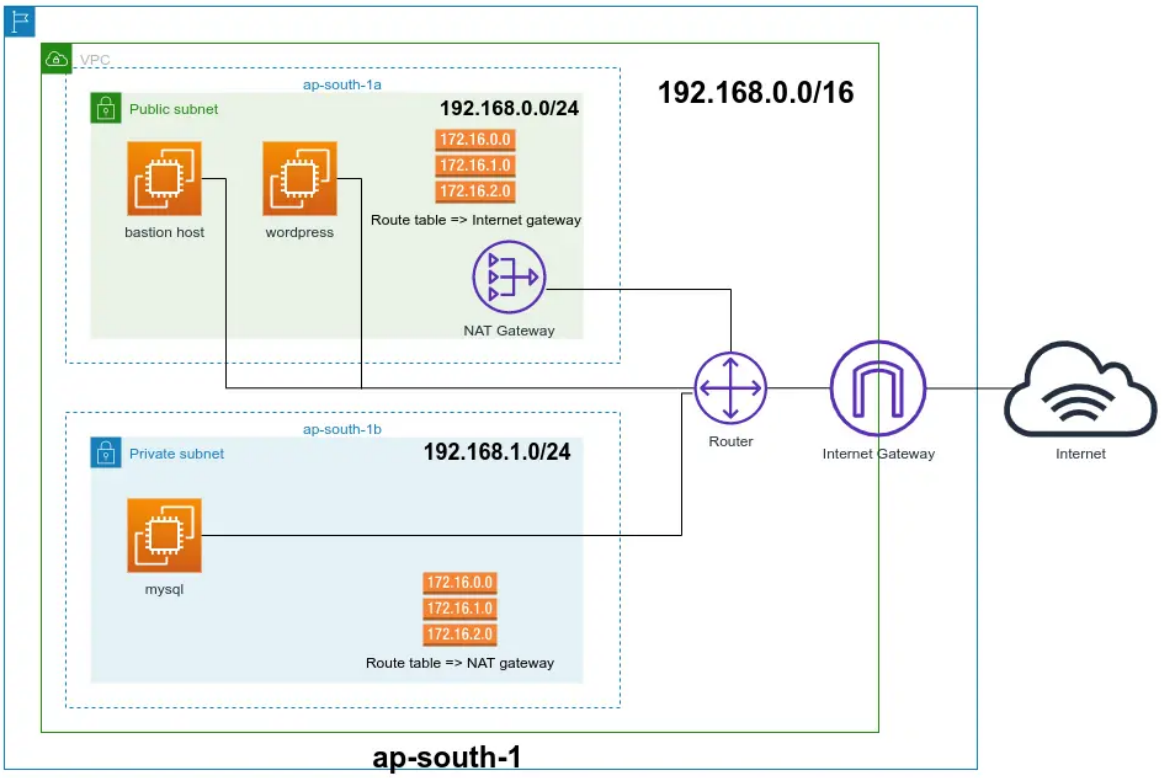


Use Case’s

Imagine you're setting up a VPC for your company's cloud infrastructure. Here's how it might look:

TIP : **(this descriptions are just for imagination).**

1. **VPC Layout:** Your VPC is like a digital office building. It has different floors (subnets) for different departments, such as HR, Finance, and Development.
2. **Subnets:** Each subnet is like a room on a floor. You might have one subnet for HR, one for Finance, and one for Development. Each subnet has its own IP address range.
3. **Security Groups:** These act like security guards at the entrance of each room. You decide who can enter (access rules) and what they can do inside (inbound/outbound traffic rules).
4. **Internet Gateway:** This is like the main entrance to your office building. It allows resources in your VPC to communicate with the internet.
5. **Route Tables:** These are like maps that tell data packets where to go. You create routes to direct traffic within your VPC and to external destinations.
6. **NAT Gateway:** If your resources need to access the internet but you want to keep them hidden from inbound traffic, you can use a NAT gateway. It's like a proxy server that acts on behalf of your resources.

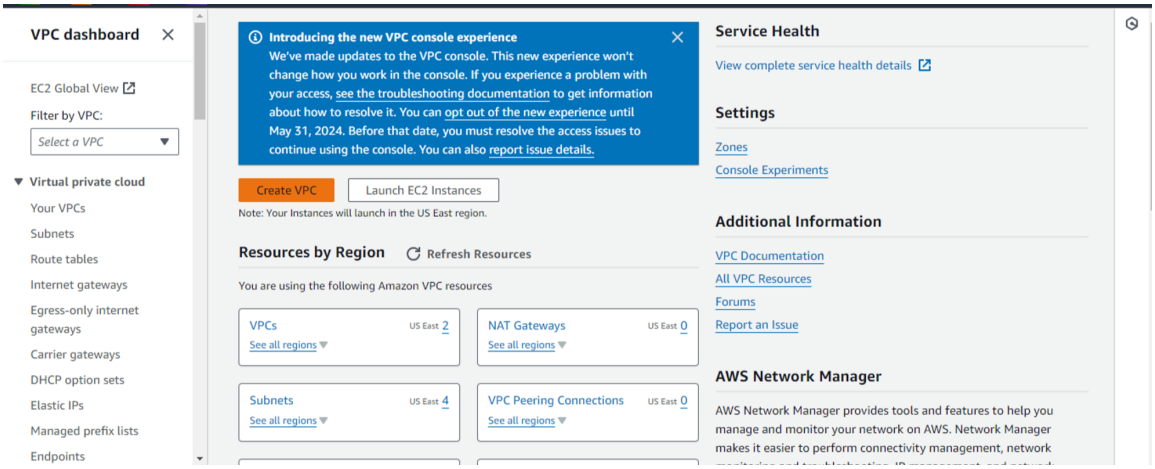


STEPS TO CREATE OUR OWN VPC NETWORK.

VPC network is basically our isolated networks created by AWS for increasing security with respect to the client.

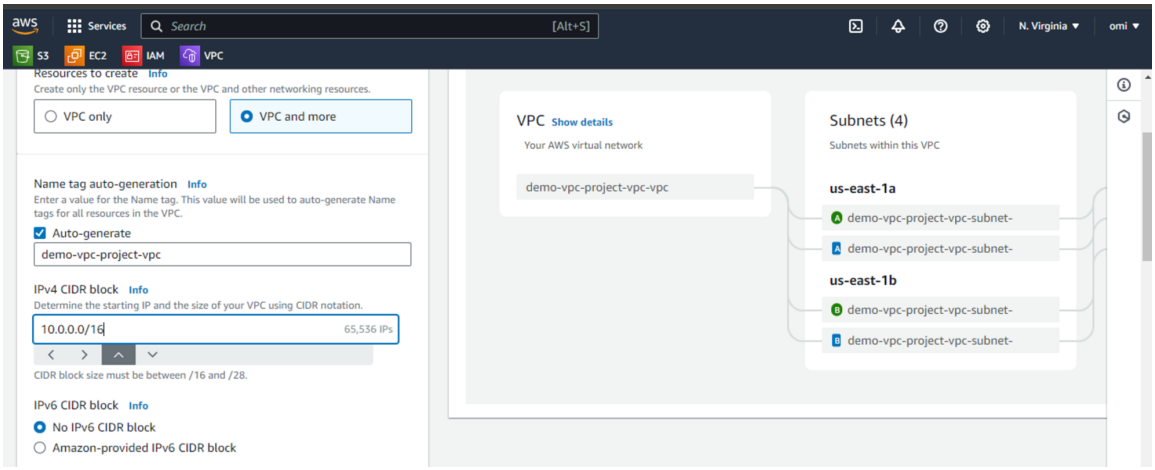
to create VPC network in AWS we follows the process as below's

1. **Sign in to the AWS Management Console:**
    - Log in to your AWS account using your credentials.
2. **Navigate to the VPC Dashboard:**
    - In the AWS Management Console, go to the "Services" menu and select "VPC" under the "Networking & Content Delivery" section.



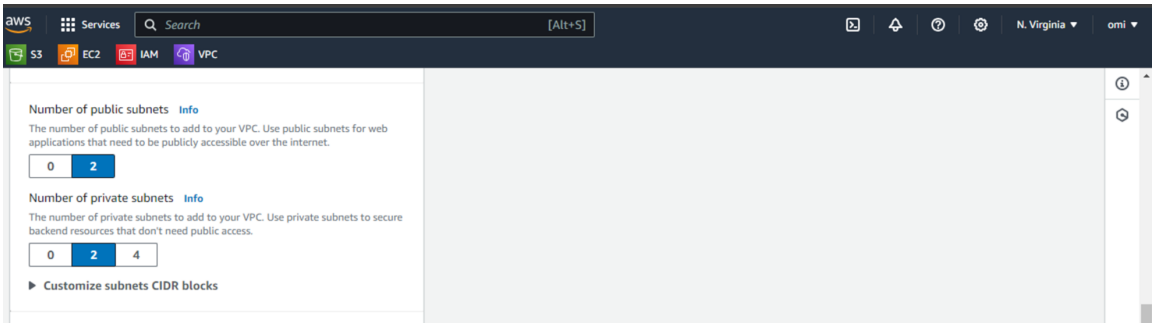
3. **Create a VPC:**

   - start creating a VPC. Specify the name tag to your VPC. the IPv4 CIDR block for your VPC. This is the primary IP address range for your VPC. For example, `10.0.0.0/16`.



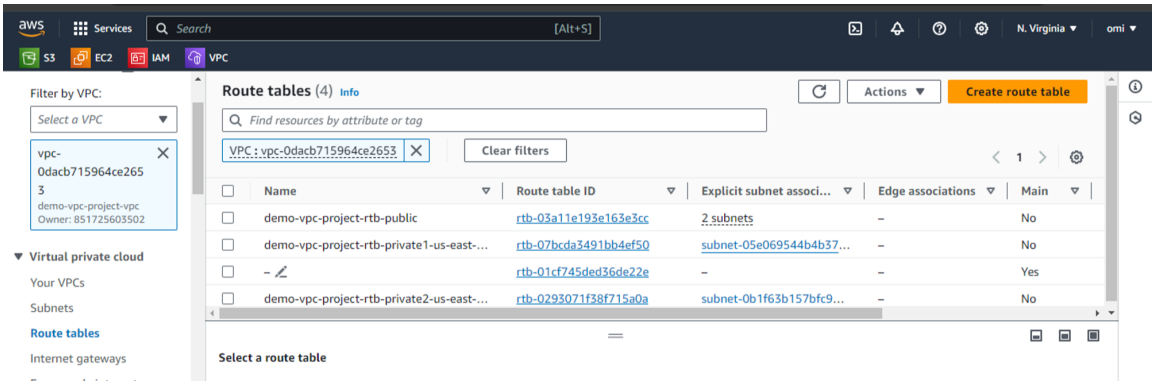
4. **Configure Subnets:**

   - Within your VPC, create subnets. Subnets are segments of your VPC where you deploy your resources. Define each subnet's IPv4 CIDR block and the Availability Zone it belongs to. For example, `10.0.1.0/24` in `us-east-1a`. here we are keeping it default.



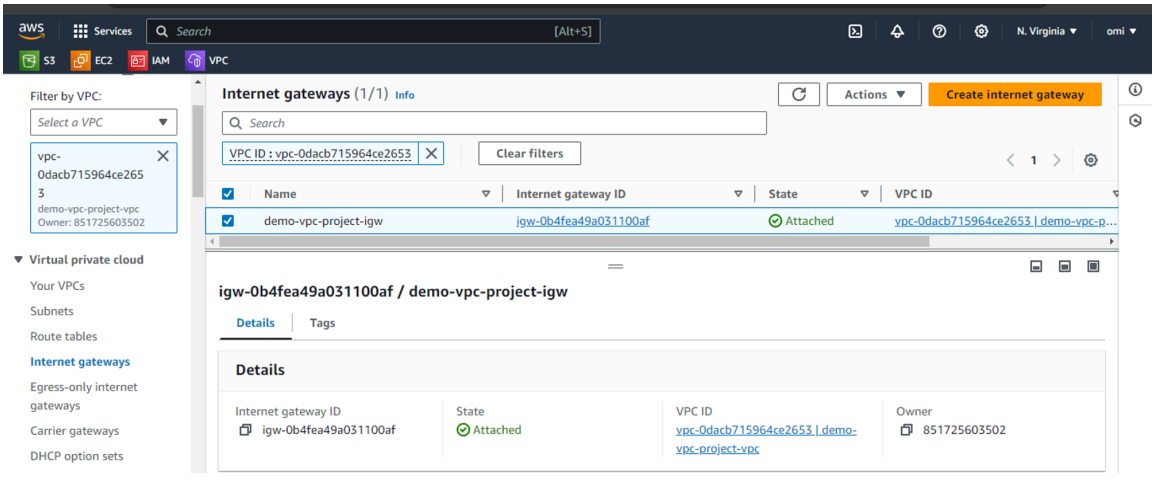
5. **Create Route Tables:**

   - Route tables determine where network traffic is directed. By default, a main route table is created with your VPC. You may need to create additional route tables for specific purposes, such as public and private subnets.

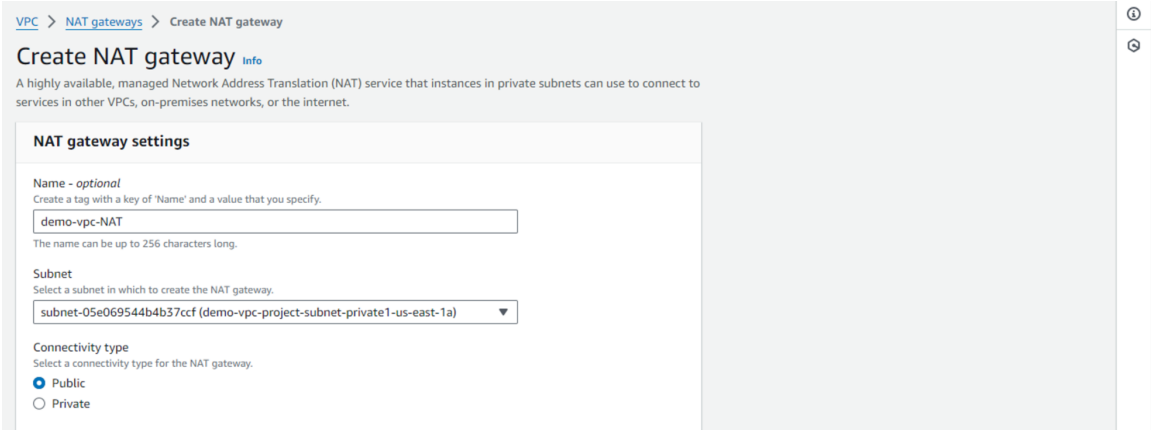


6. **Configure Internet Gateway (IGW) (for Public Subnets):**

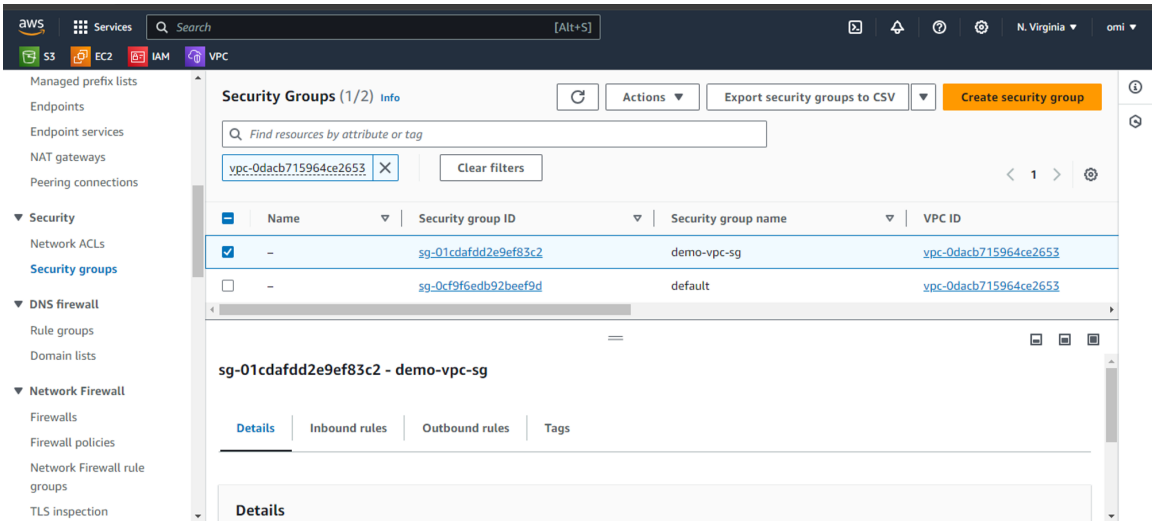
   - If your VPC needs to communicate with the internet, create an internet gateway here I given it name as **demo-vpc-project-igw** and then attach it to your VPC. Associate the internet gateway with the route table of your public subnets.



7. **Configure NAT Gateway or NAT Instance (for Private Subnets):**
    - If resources in your private subnets need internet access, you can set up a NAT gateway or NAT instance. This allows private subnet resources to initiate outbound connections to the internet while remaining inaccessible from the outside.

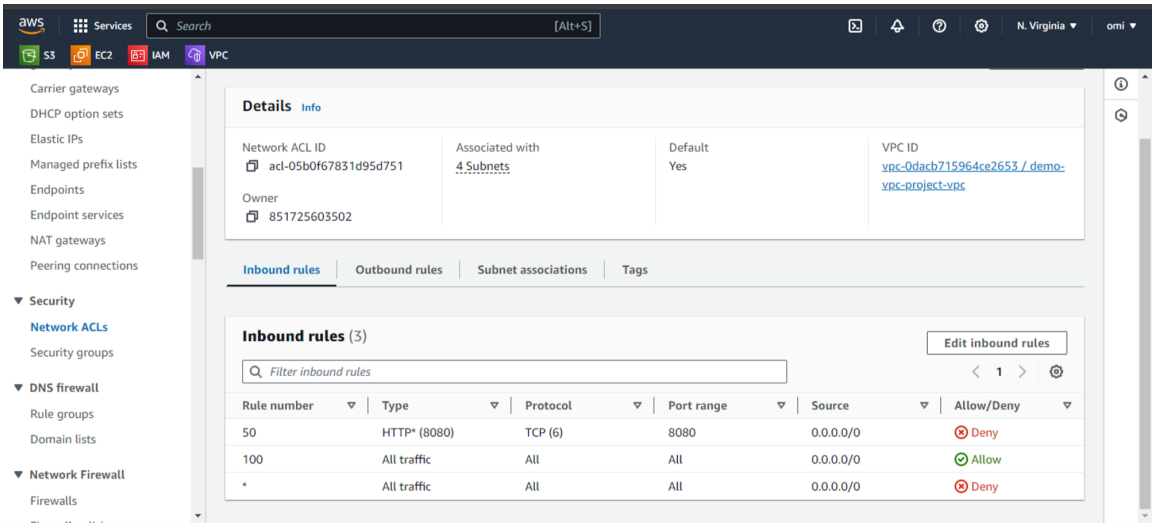


8. **Set Up Security Groups:**
    - Security groups act as virtual firewalls for your EC2 instances and other resources. Define inbound and outbound traffic rules to control access to your resources. Associate security groups with your instances.



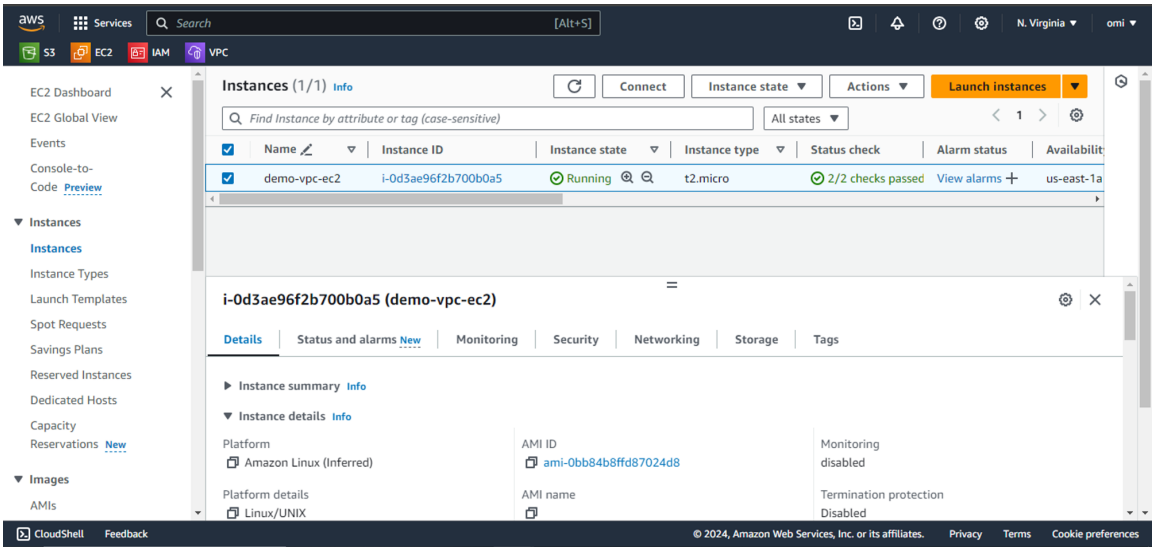
9. **(Optional) Configure Network ACLs:**
    - **Network Access Control Lists** (ACLs) are stateless firewalls that control traffic at the subnet level. You can define rules to allow or deny traffic

based on IP addresses, ports, and protocols.



10. **Launch Instances and Resources:**

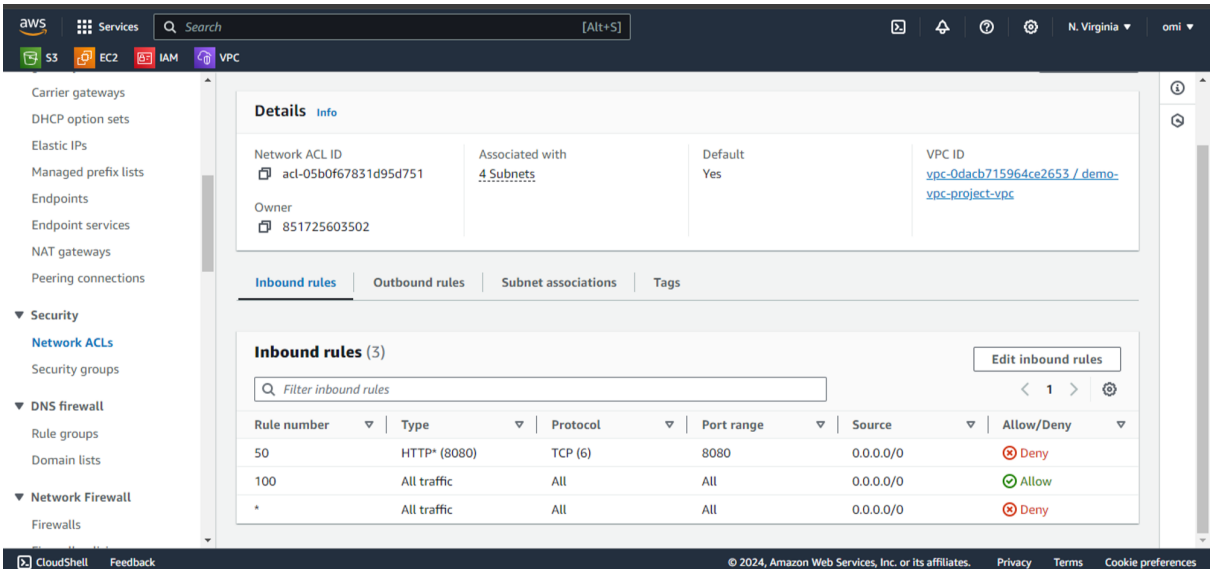
- Now that your VPC is set up, you can launch EC2 instances, ELB load balancers, and other AWS resources as per your requirements within your VPC.



11. **Test Connectivity:**

- After launching resources, verify connectivity both within your VPC and to external networks to ensure that **routing and security configurations** are **working** as expected.

Here I **checked** Network ACLs by **customizing default configurations** and **changing the priorities** by changing the **Rule numbers** of **Inbound rules.**


aws
Services
Search
[Alt+S]
N. Virginia
omi
S3
EC2
IAM
VPC
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Network ACLs
Security groups
DNS firewall
Rule groups
Domain lists
Network Firewall
Firewalls
Details Info
Network ACL ID
acl-05b0f67831d95d751
Owner
851725603502
Associated with
4 Subnets
Default
Yes
VPC ID
vpc-0dacb715964ce2653 / demo-vpc-project-vpc
Inbound rules
Outbound rules
Subnet associations
Tags
Inbound rules (3)
Edit inbound rules
Filter inbound rules
Rule number
Type
Protocol
Port range
Source
Allow/Deny
50
HTTP* (8080)
TCP (6)
8080
0.0.0.0/0
Deny
100
All traffic
All
All
0.0.0.0/0
Allow
*
All traffic
All
All
0.0.0.0/0
Deny
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

THANK YOU !