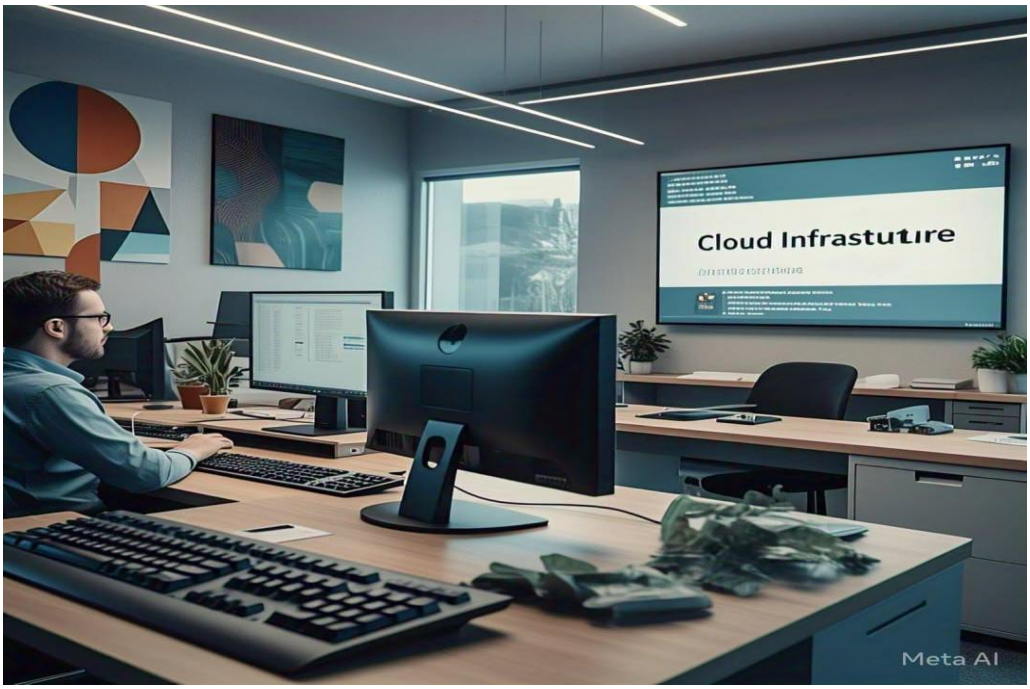
CLOUD INFRASTRUCTURE USING AWS

by:
Renuka Pulavarthi

1.Introduction

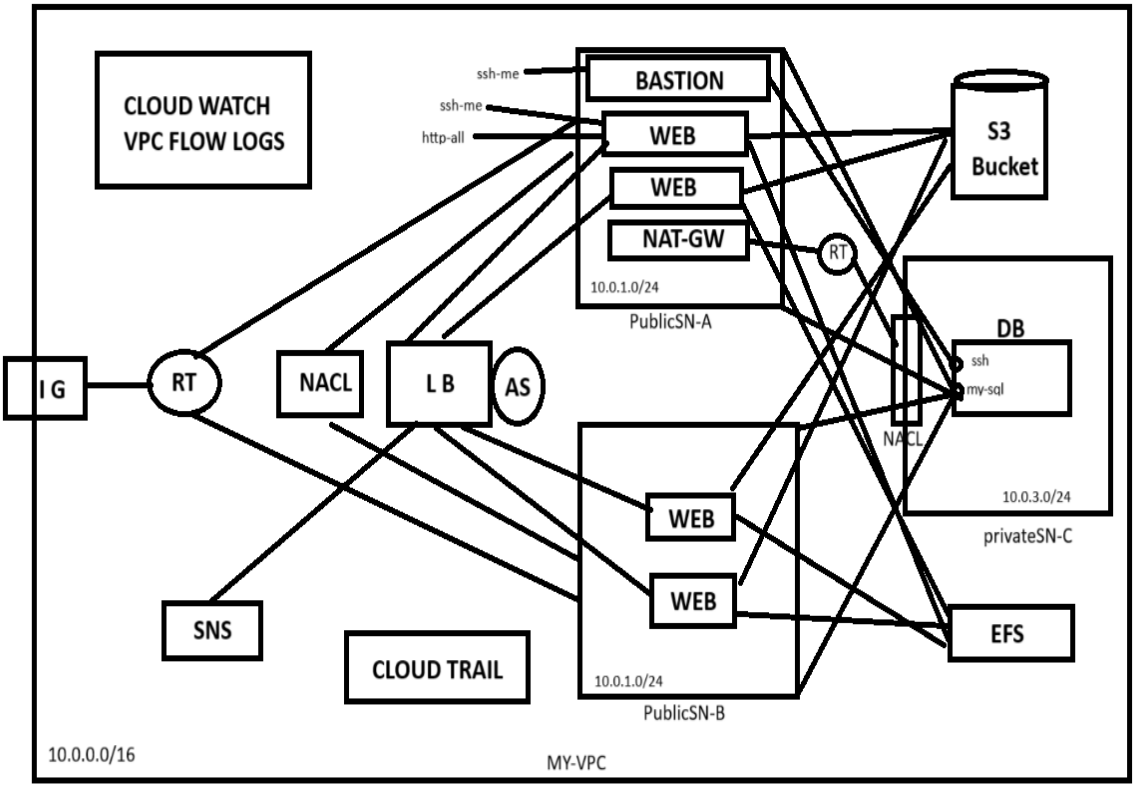
This document details the process of setting up an AWS infrastructure that includes one VPC with two public subnets and one private subnet. The architecture consists of multiple EC2 instances, a Bastion server, a NAT Gateway, a database server, an S3 bucket, EFS, SNS, CloudWatch, CloudTrail, a Load Balancer, Auto Scaling, NACL, a Route Table, and an Internet Gateway. Each component is connected and configured to ensure secure and efficient operations.



This infrastructure is designed to provide:

- **High availability and scalability** using Auto Scaling and Load Balancer.
- **Enhanced security** with Security Groups, NACLs, and Bastion Host.
- **Efficient storage and data management** through S3, EFS, and a dedicated Database Server.
- **Real-time monitoring and logging** with CloudWatch and CloudTrail.
- **Optimized network performance** using NAT Gateway, Route Tables, and Internet Gateway.

2.Overview



Structure of the cloud infrastructure

The infrastructure consists of:

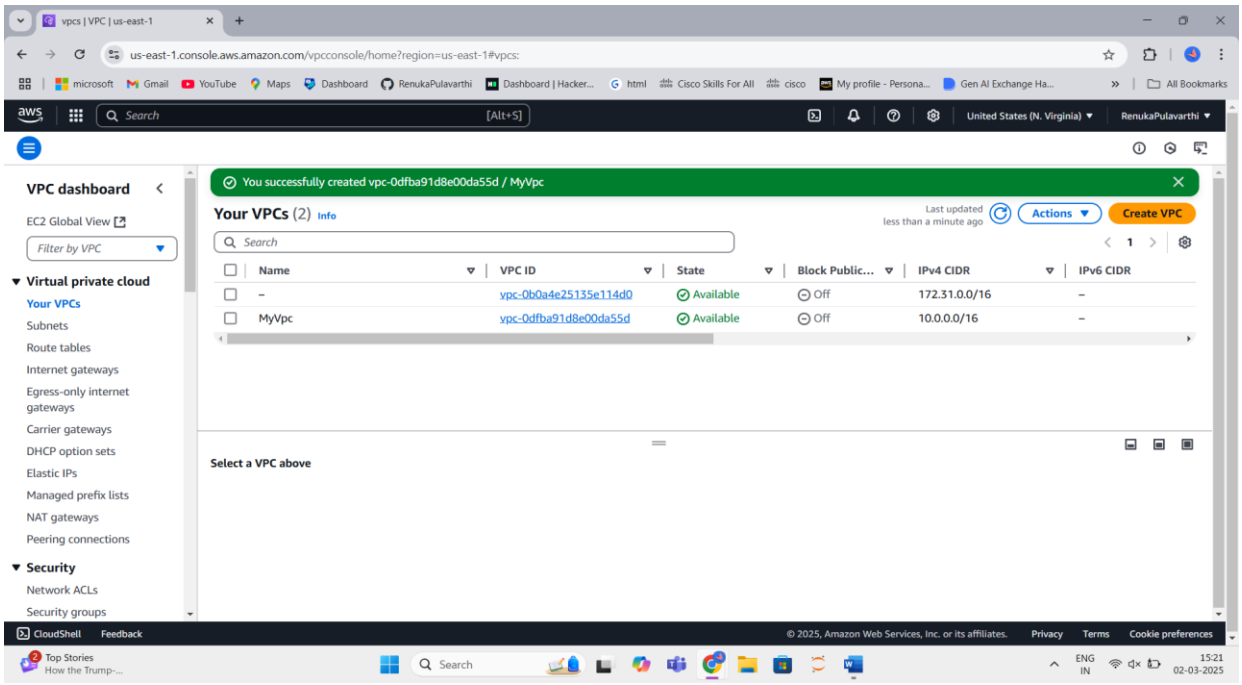
- **1 VPC**
- **2 Public Subnets**
  - **Public Subnet 1:** Contains 2 EC2 instances, a Bastion server, and a NAT Gateway**.**
  - **Public Subnet 2:** Contains 2 EC2 instances**.**
- **1 Private Subnet**
  - Hosts a Database Server**.**
- **Other AWS Resources:**
  - S3 Bucket for storage.
  - EFS for shared file storage.0
  - SNS for notifications.
  - CloudWatch for monitoring.
  - CloudTrail for tracking AWS API activity.
  - Load Balancer (LB) connected to Auto Scaling (AS).
  - NACL and Route Table for network management.
  - Internet Gateway for internet access.

3.STEPS

3.1. Creating a VPC

Purpose: A Virtual Private Cloud (VPC) provides an isolated network environment for your AWS resources.

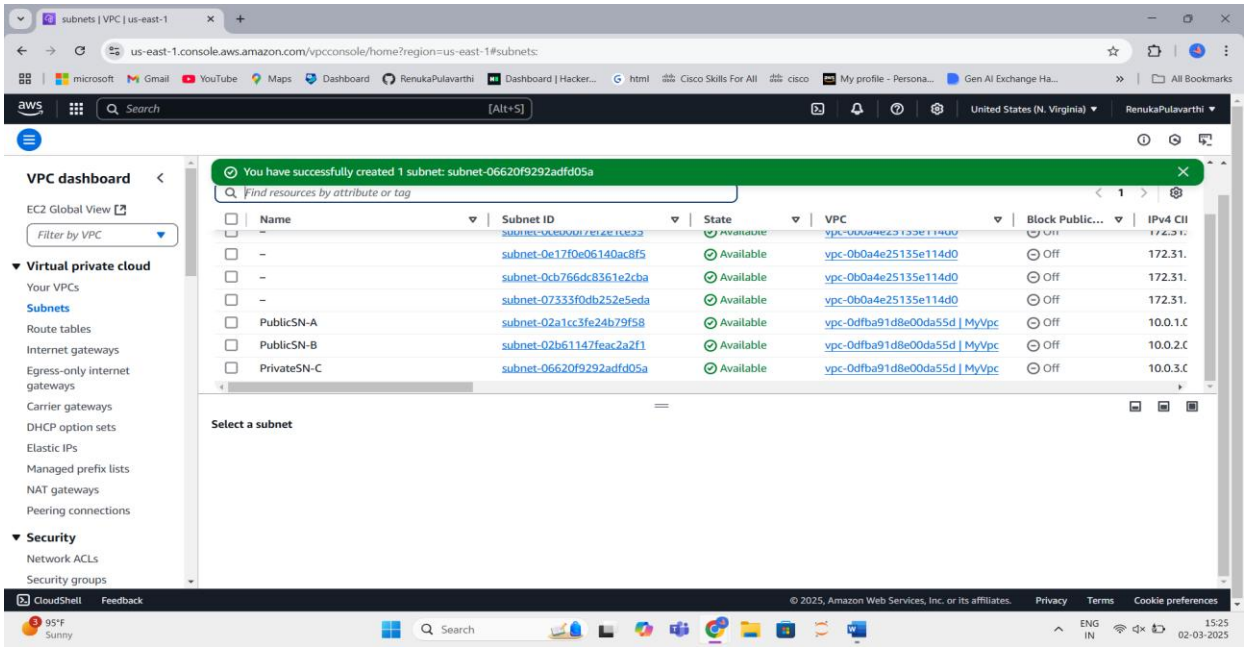
- Navigate to the **AWS VPC Console**.
- Click **Create VPC**.
- Choose **IPv4 CIDR block** (e.g., 10.0.0.0/16).
- Select **default tenancy**.
- Click **Create**.

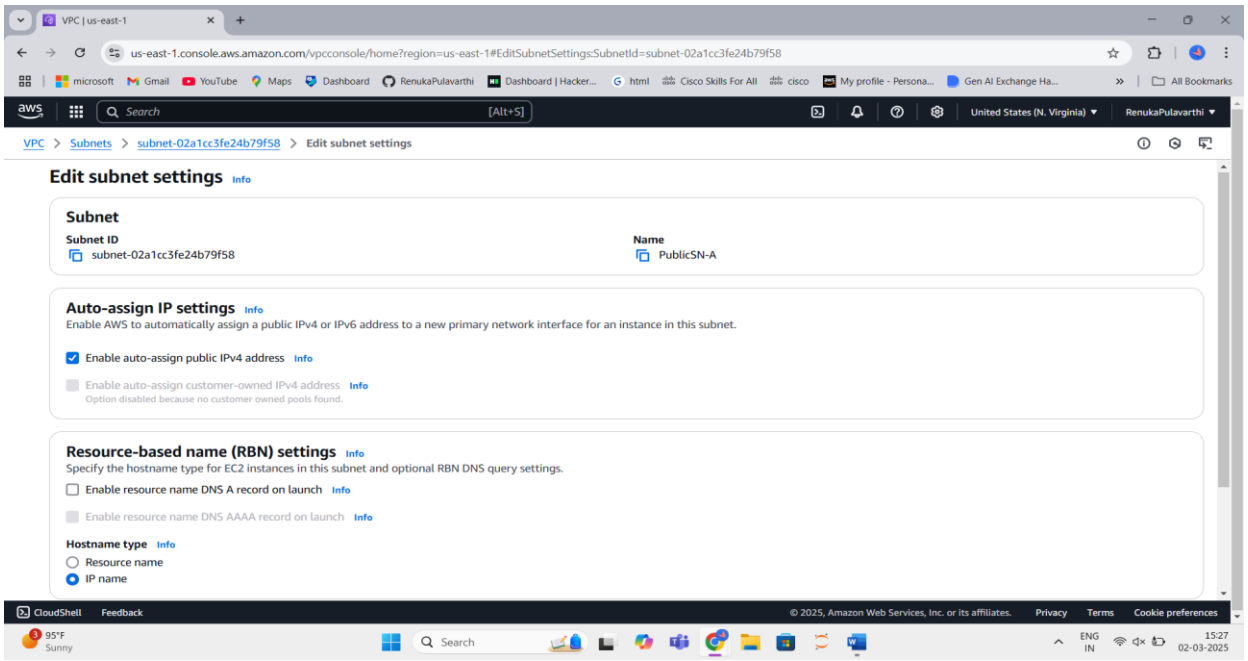


3.2. Creating Public and Private Subnets

Purpose: Subnets divide the VPC into smaller network segments, enabling better organization and security.

- In the VPC dashboard, select **Subnets** > **Create Subnet**.
- Choose the created **VPC**.
- Define CIDR blocks:
    - Public Subnet A: 10.0.1.0/24
    - Public Subnet B: 10.0.2.0/24
    - Private Subnet C: 10.0.3.0/24
- Ensure public subnets have **auto-assign public IP enabled**.

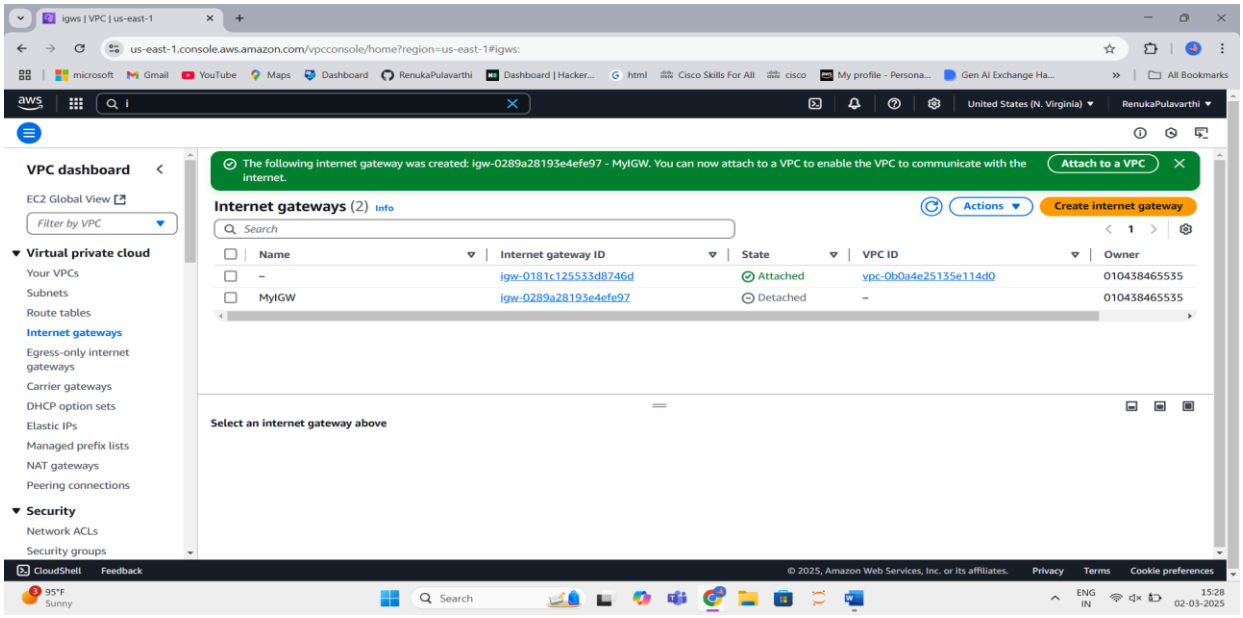


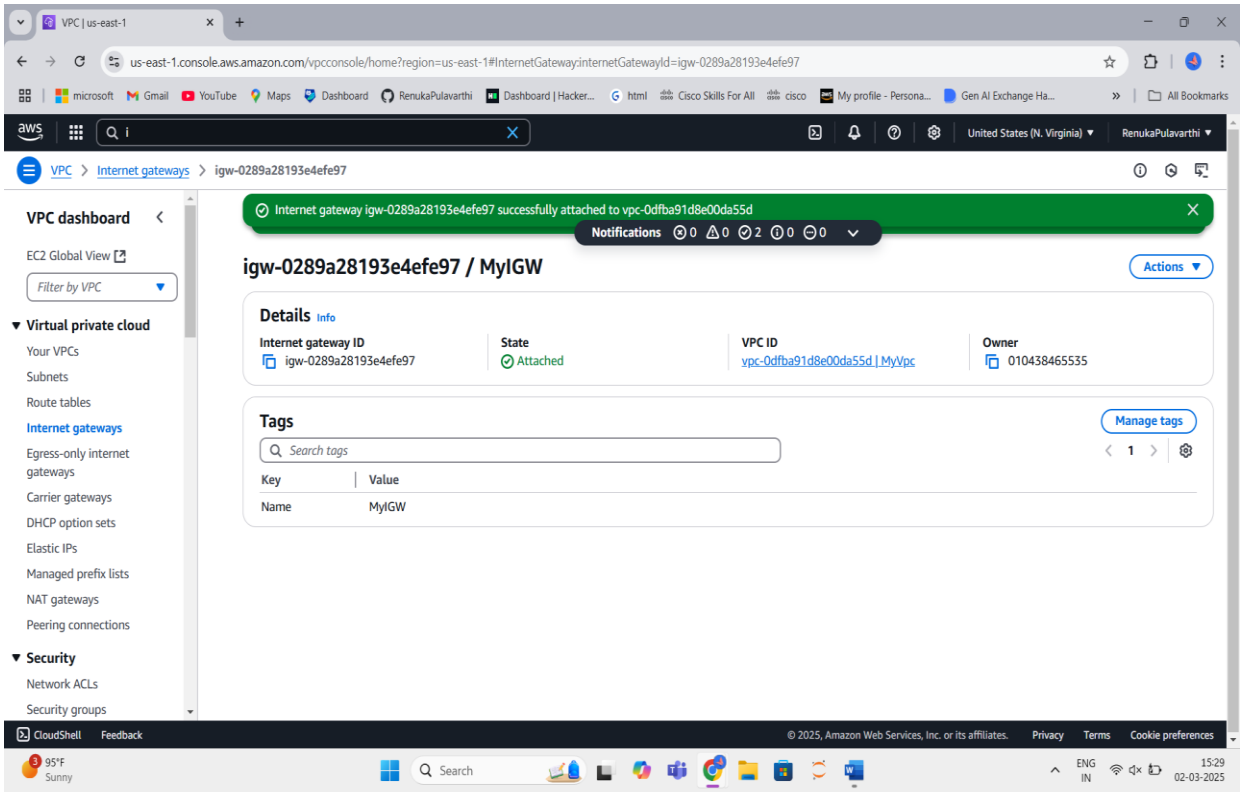


3.3. Connecting an Internet Gateway to the VPC

Purpose: Allows public subnets to access the internet.

- Navigate to **Internet Gateways**.
- Click **Create Internet Gateway**.
- Attach it to the **VPC**.

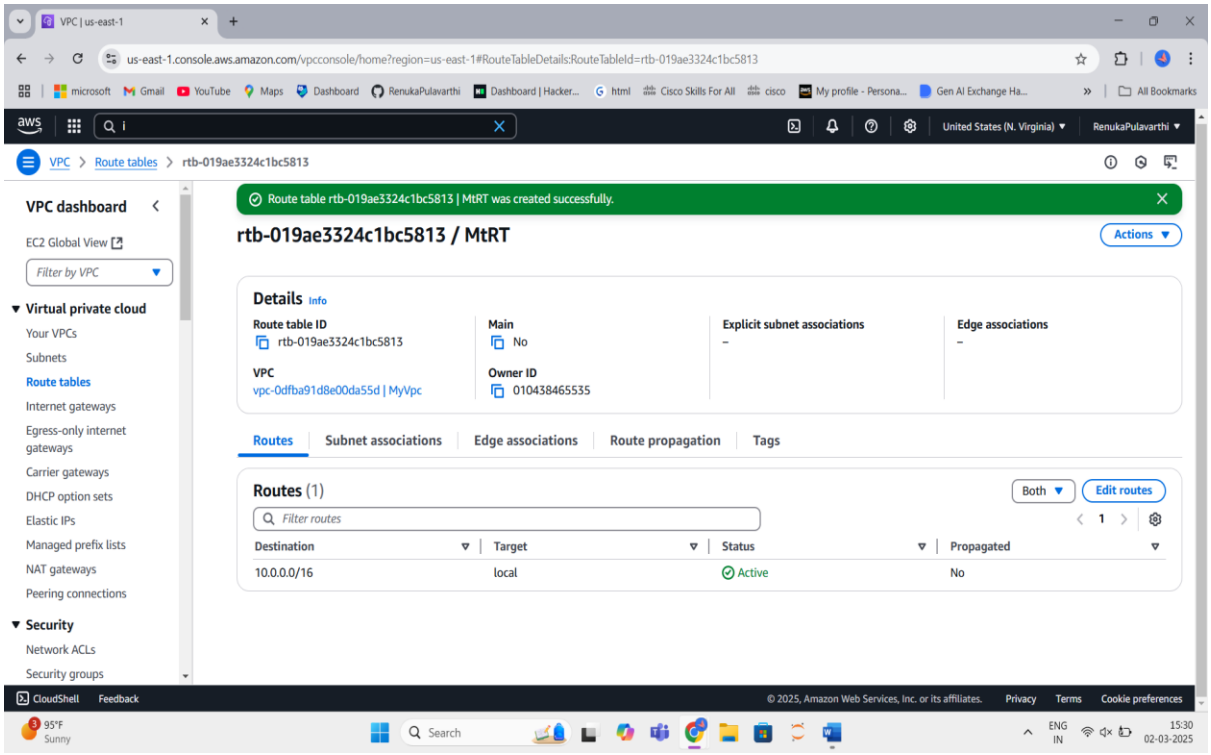


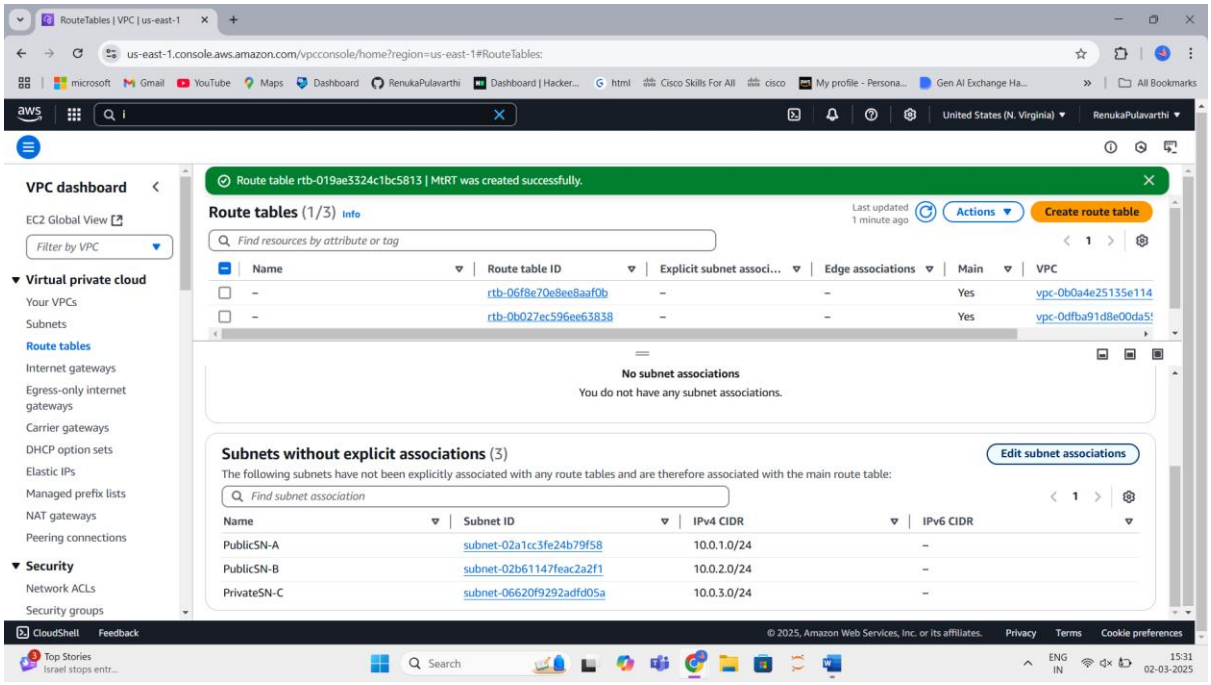


3.4. Creating a Route Table

Purpose: A route table controls the traffic routing between subnets and external resources**.**

- Go to **Route Tables** in the VPC dashboard.
- Click **Create Route Table** and associate it with the **VPC**.





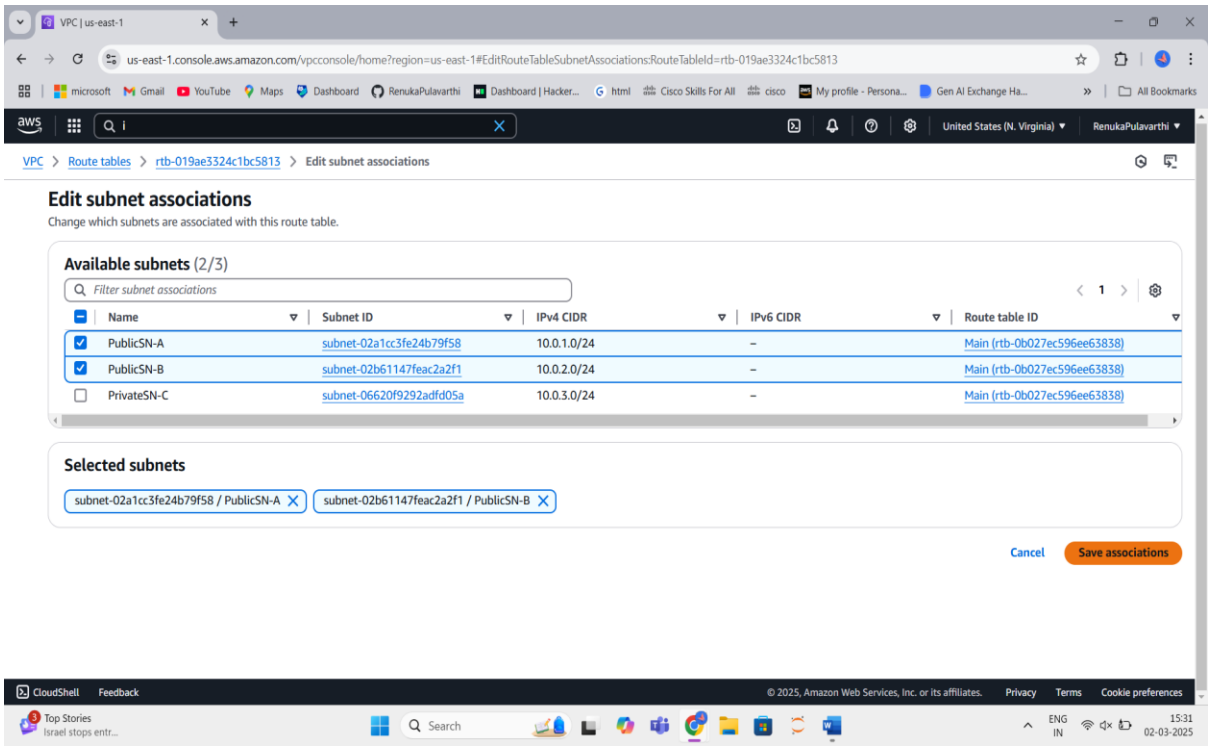
3.5. Connecting the Route Table to the Internet Gateway

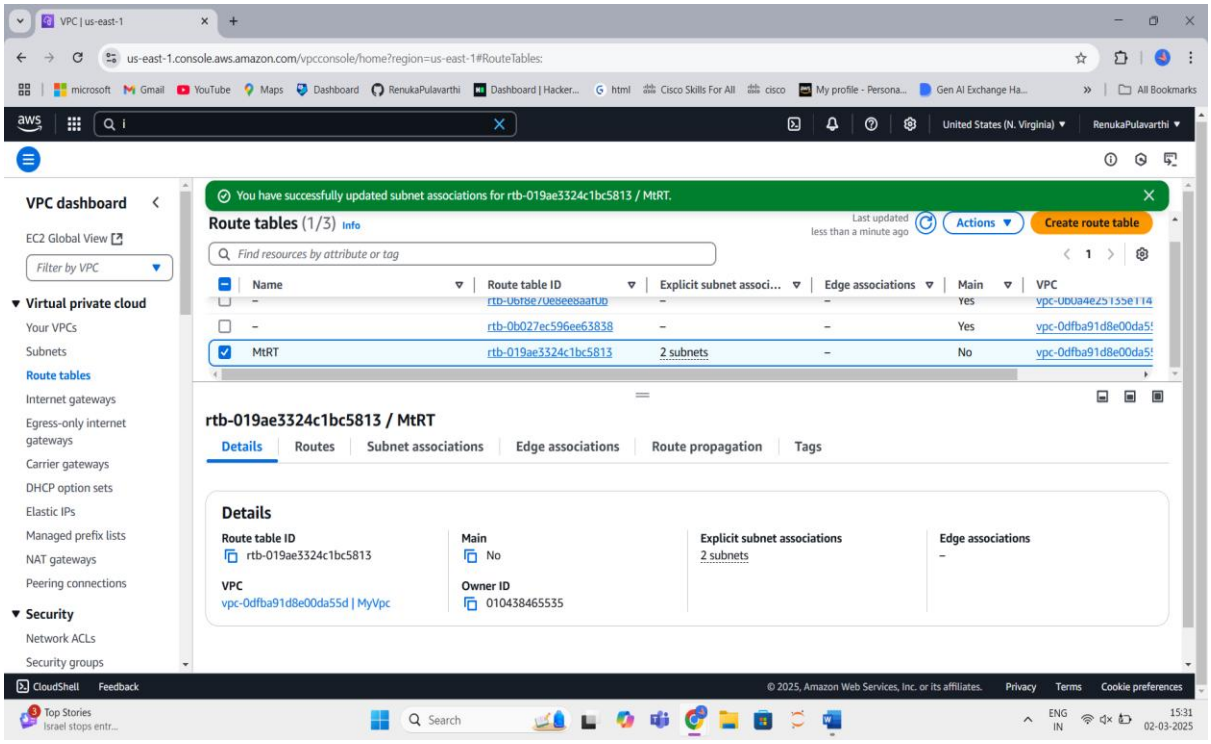
Purpose: Enables internet access for resources in public subnets.

- Edit the **Route Table**.
- Add a new route:
    - Destination: **0.0.0.0/0**

- Target: **Internet Gateway**

- Associate the route table with public subnets.

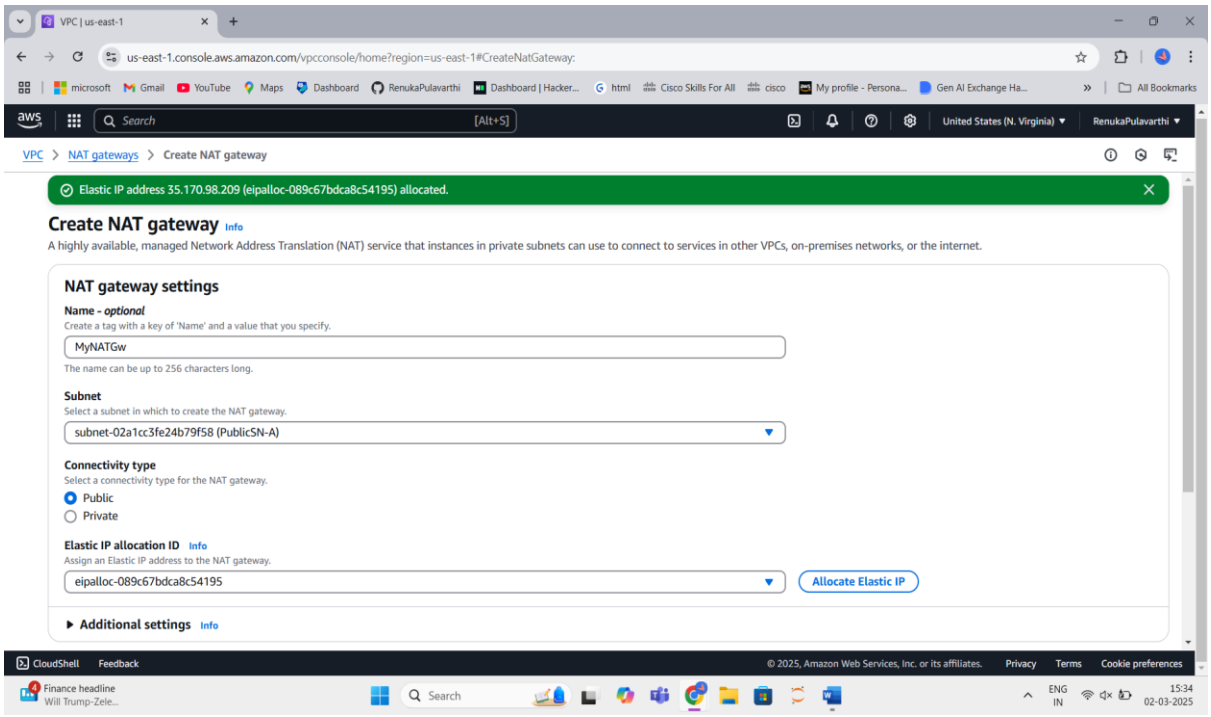


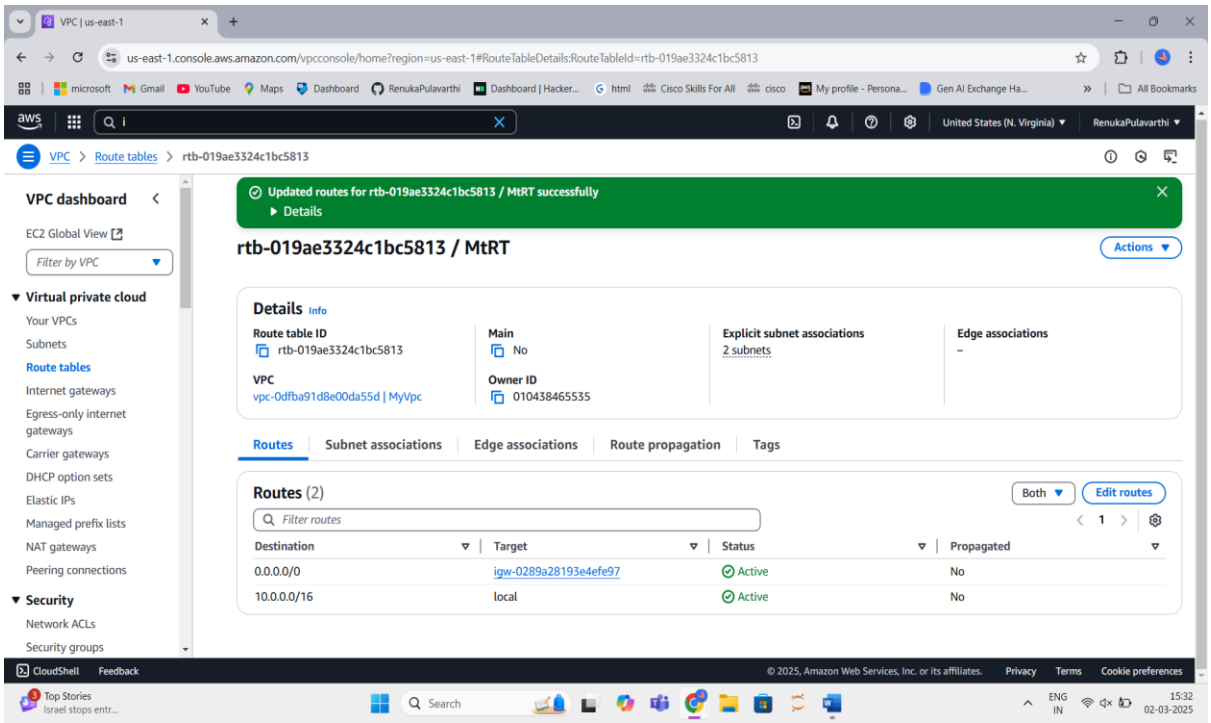


3.6. Creating a NAT Gateway

Purpose: Allows private subnet instances to access the internet securely.

- Go to **NAT Gateway** in the VPC console.
- Select a public subnet.
- Assign an **Elastic IP**.
- Create the NAT Gateway.

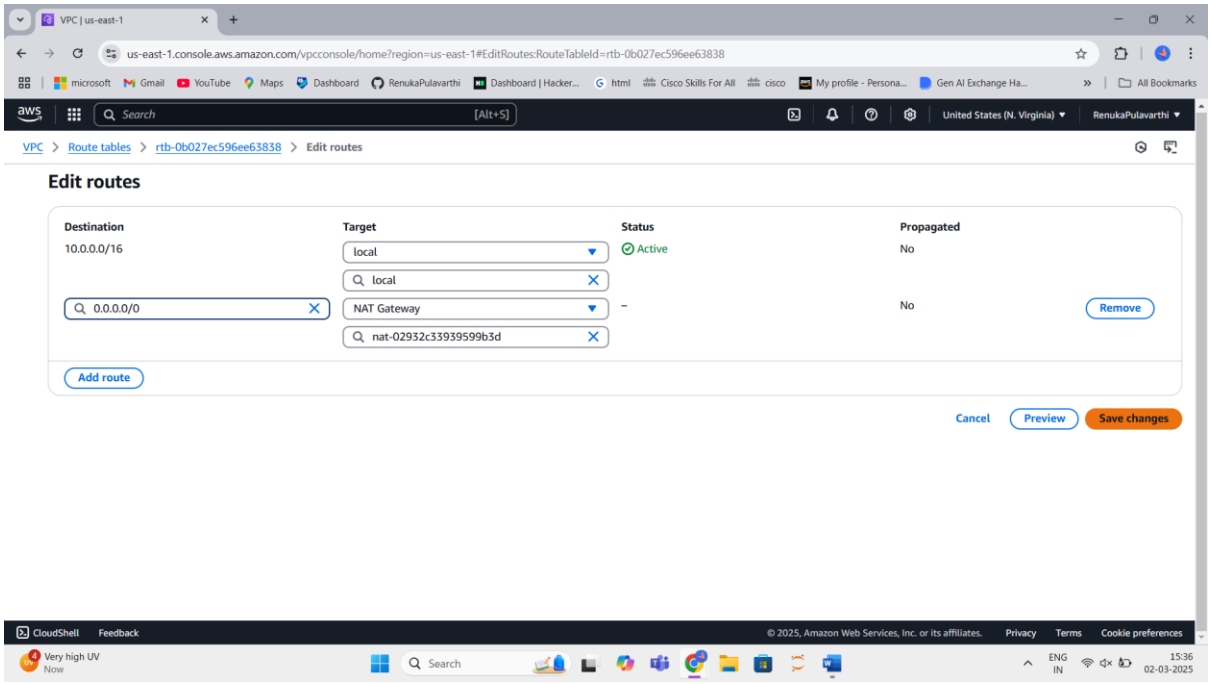


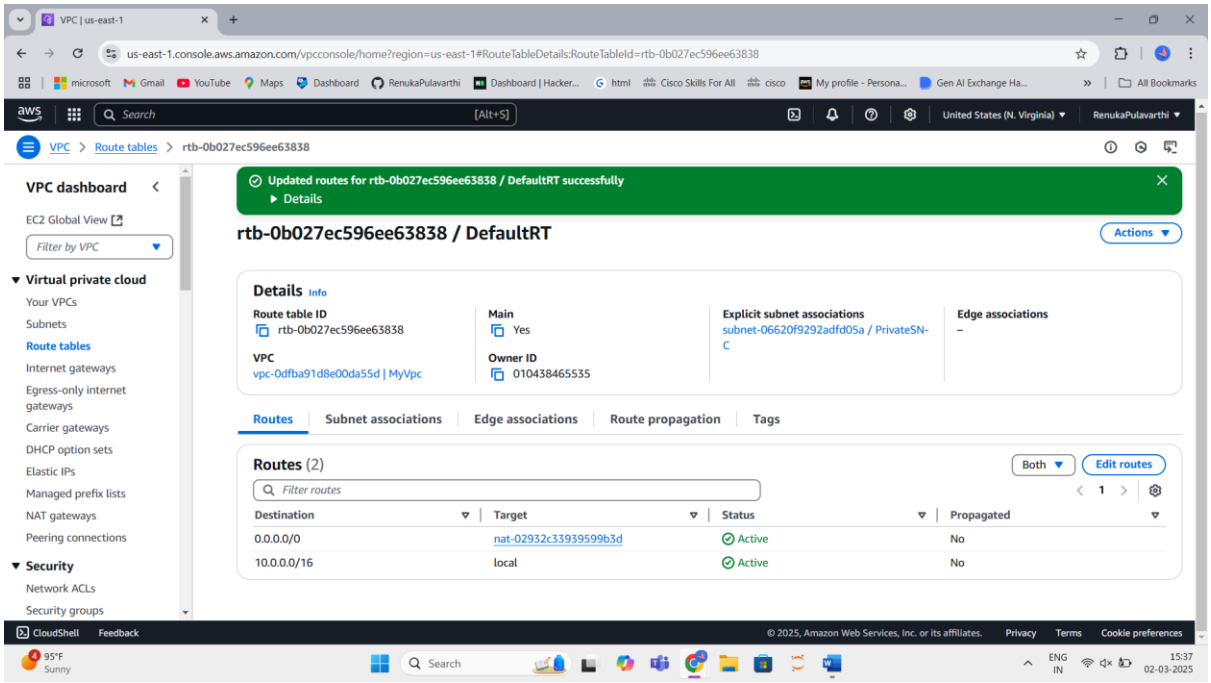


3.7. Connecting the Default Route Table to the Private Subnet and NAT Gateway

Purpose: Ensures instances in the private subnet can reach the internet via the NAT Gateway.

- Modify the **private route table**.
- Add a new route:
    - Destination: **0.0.0.0/0**
    - Target: **NAT Gateway**

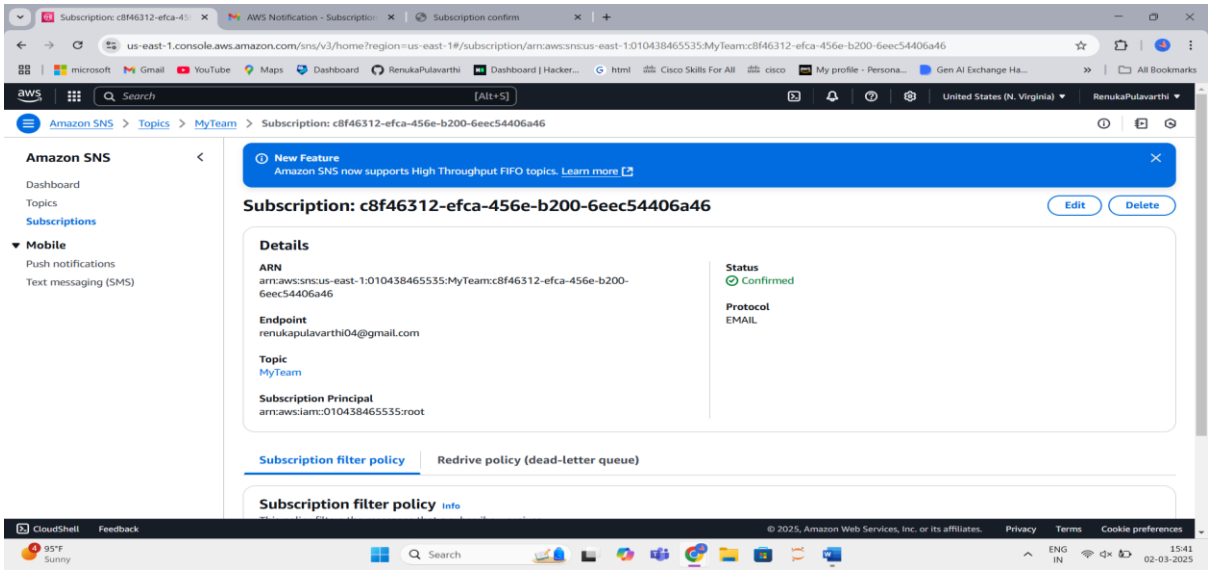




3.8. Creating Simple Notification Service (SNS)

Purpose: Enables automated messaging and alerts for AWS services.

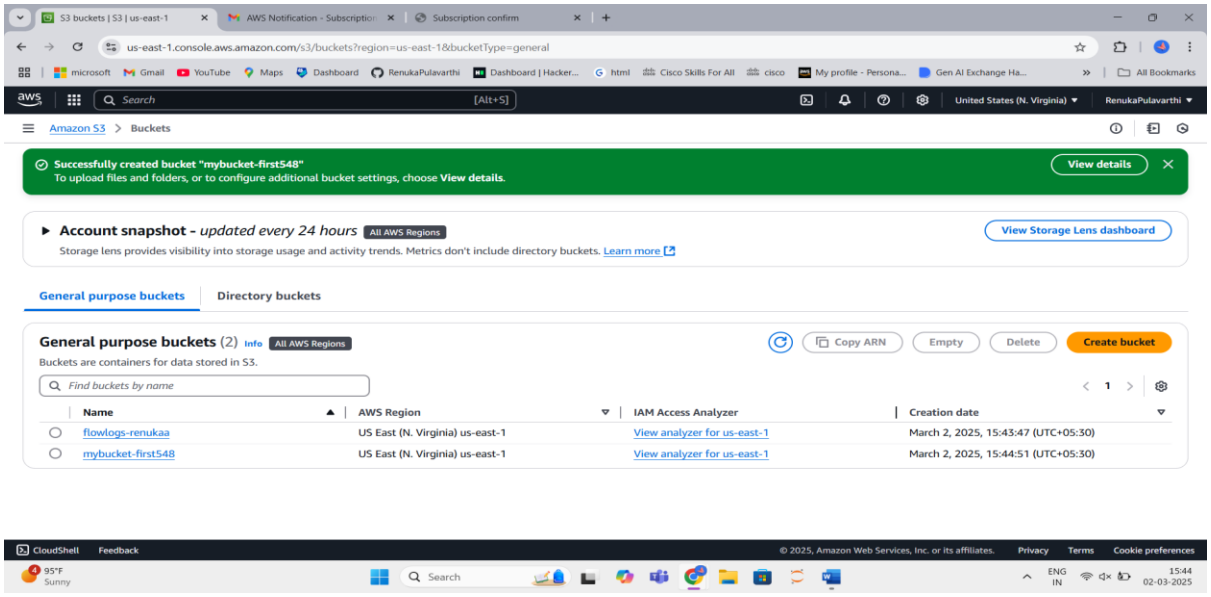
- Go to **SNS** in AWS Console.
- Click **Create Topic**.
- Define a name and create the topic.
- Create subscriptions as needed.



3.9. Creating an S3 Bucket

Purpose: Provides scalable storage for files and backups.

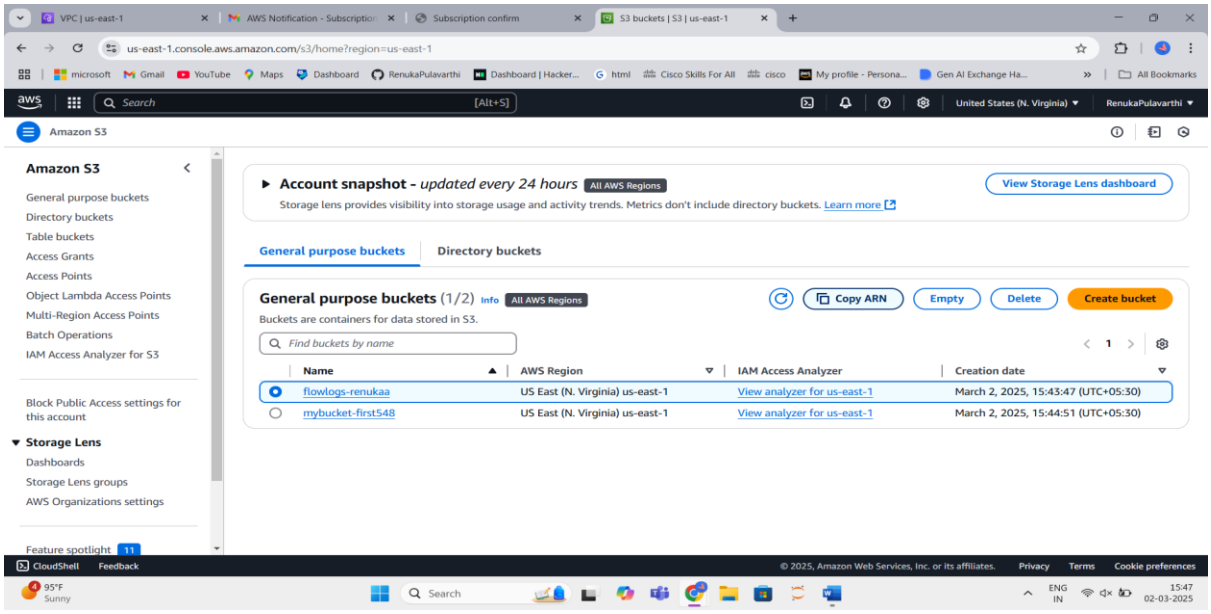
- Navigate to **S3** in AWS.
- Click **Create Bucket**.
- Define the name and region.
- Set appropriate permissions.



3.10. Creating Flow Logs for the VPC

Purpose: Captures network traffic logs for security and monitoring.

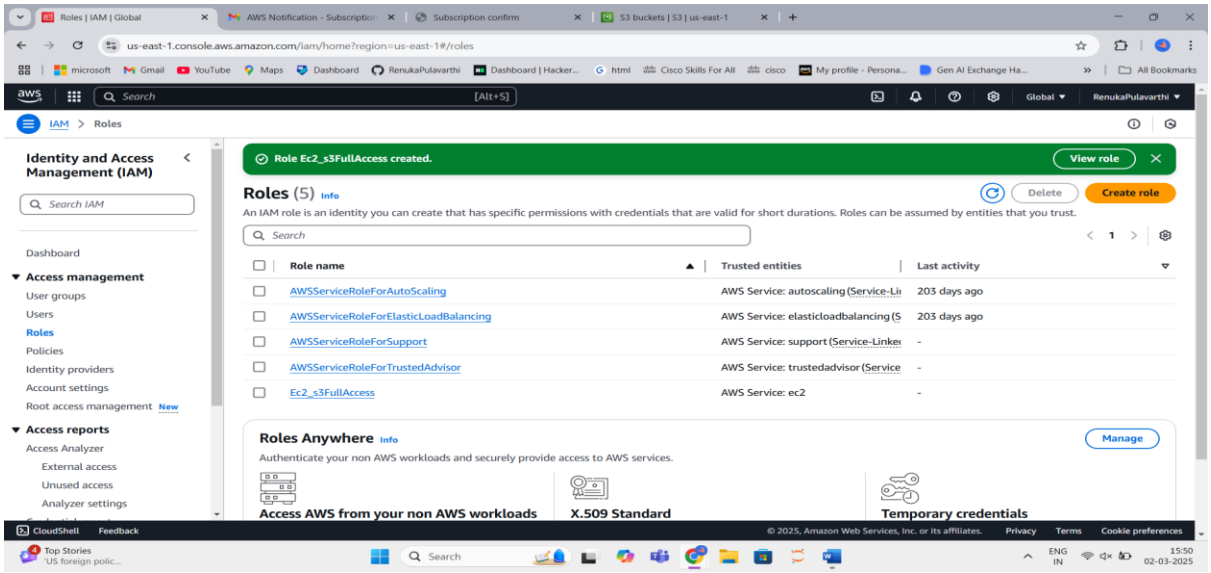
- In the **VPC dashboard**, go to **Flow Logs**.
- Click **Create Flow Log**.
- Choose **VPC** and destination (S3 or CloudWatch).



3.11. Creating IAM Role (EC2-S3 Full Access)

Purpose: Grants EC2 instances permissions to interact with S3.

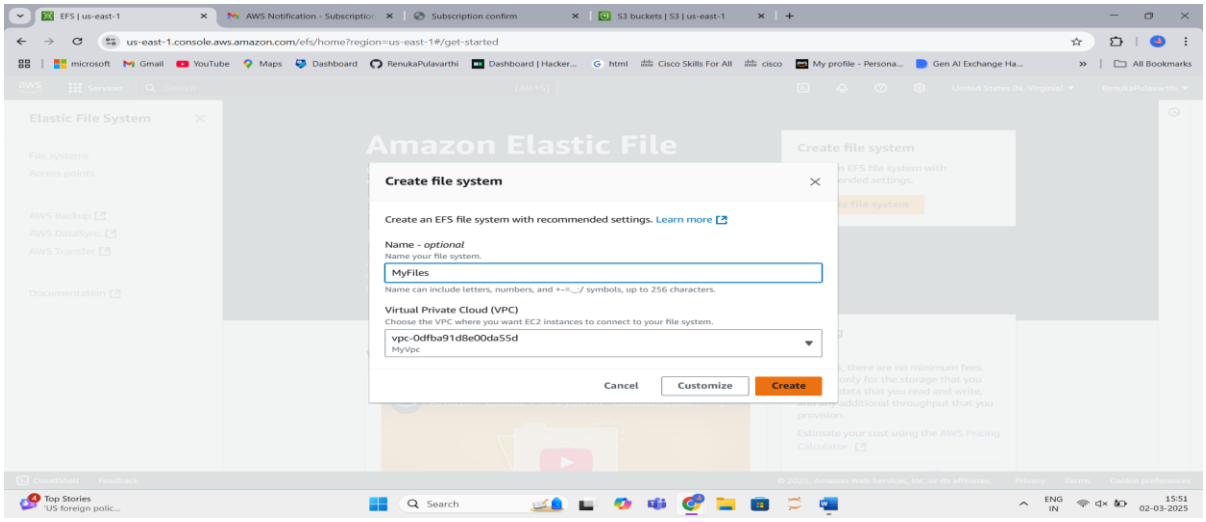
- Go to **IAM** > **Roles**.
- Click **Create Role**.
- Select **AWS Service** > **EC2**.
- Attach **AmazonS3FullAccess** policy.



3.12. Creating EFS

Purpose: Provides scalable and shared file storage for multiple EC2 instances.

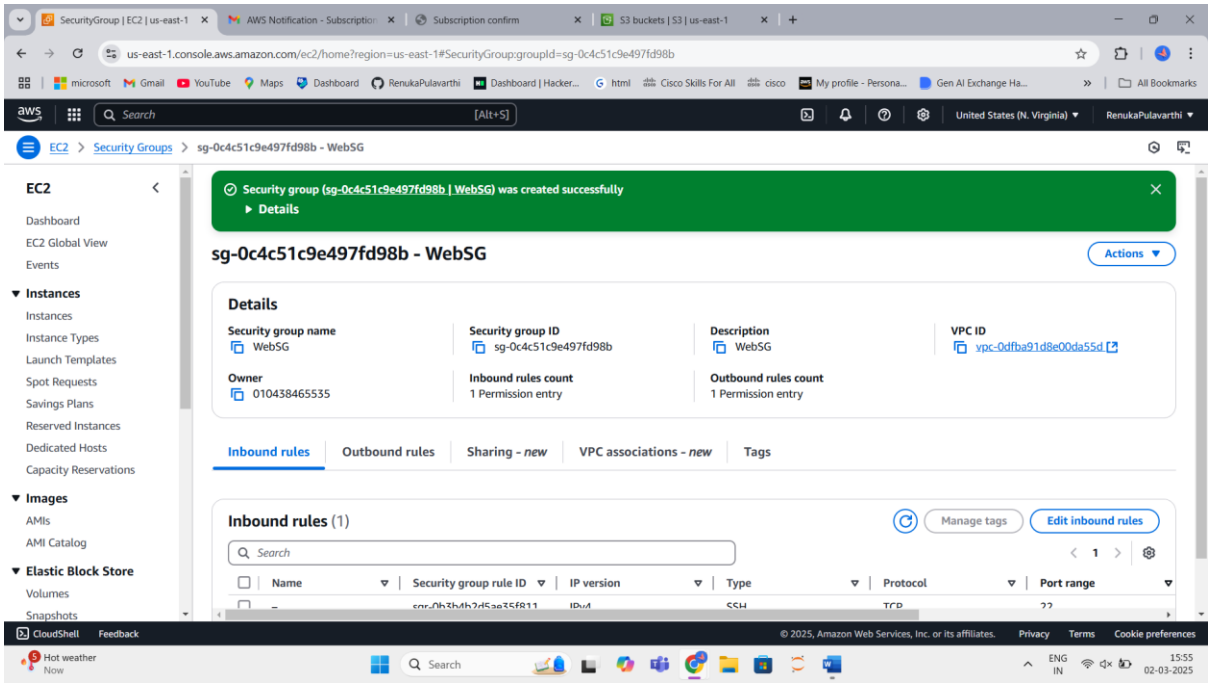
- Navigate to **EFS** in AWS.
- Click **Create File System**.
- Associate with **VPC** and define permissions.



3.13. Creating Security Groups

Purpose: Controls access to AWS resources based on IP and port rules.

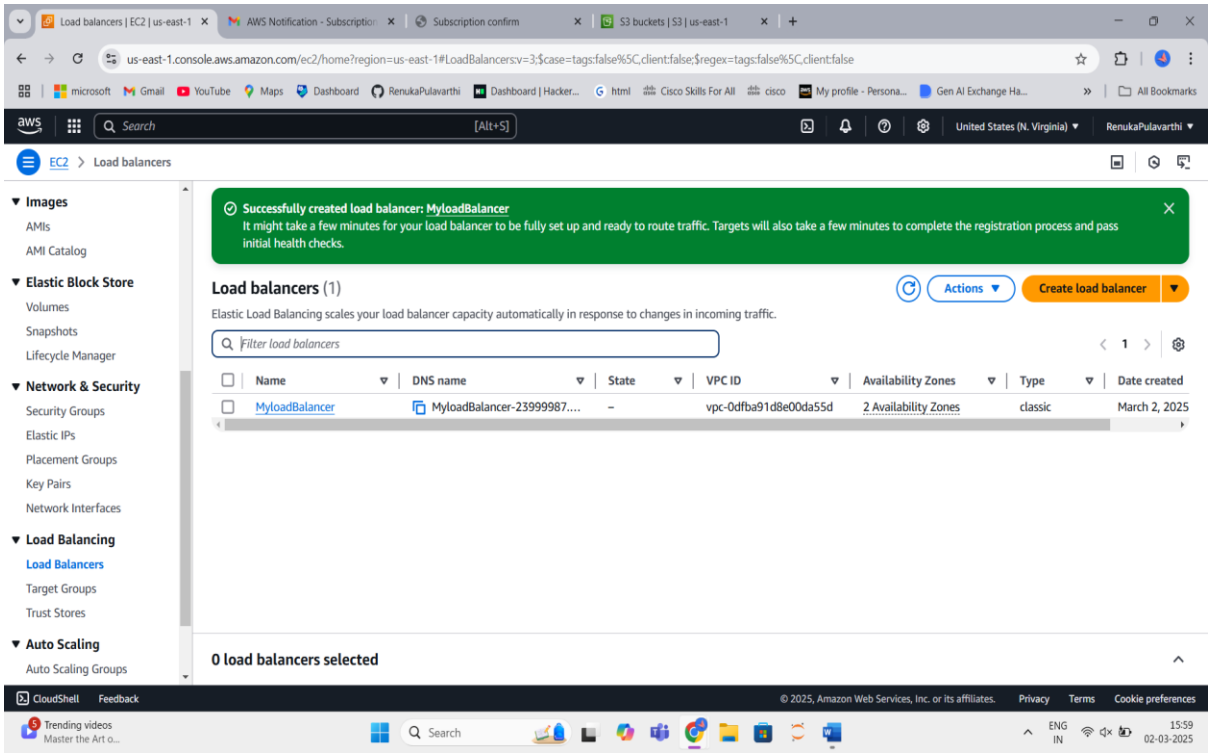
- Go to **Security Groups**.
- Create separate **security groups** for EC2, database, and load balancer with appropriate inbound/outbound rules.



3.14. Creating a Load Balancer

Purpose: Distributes traffic evenly across multiple EC2 instances.

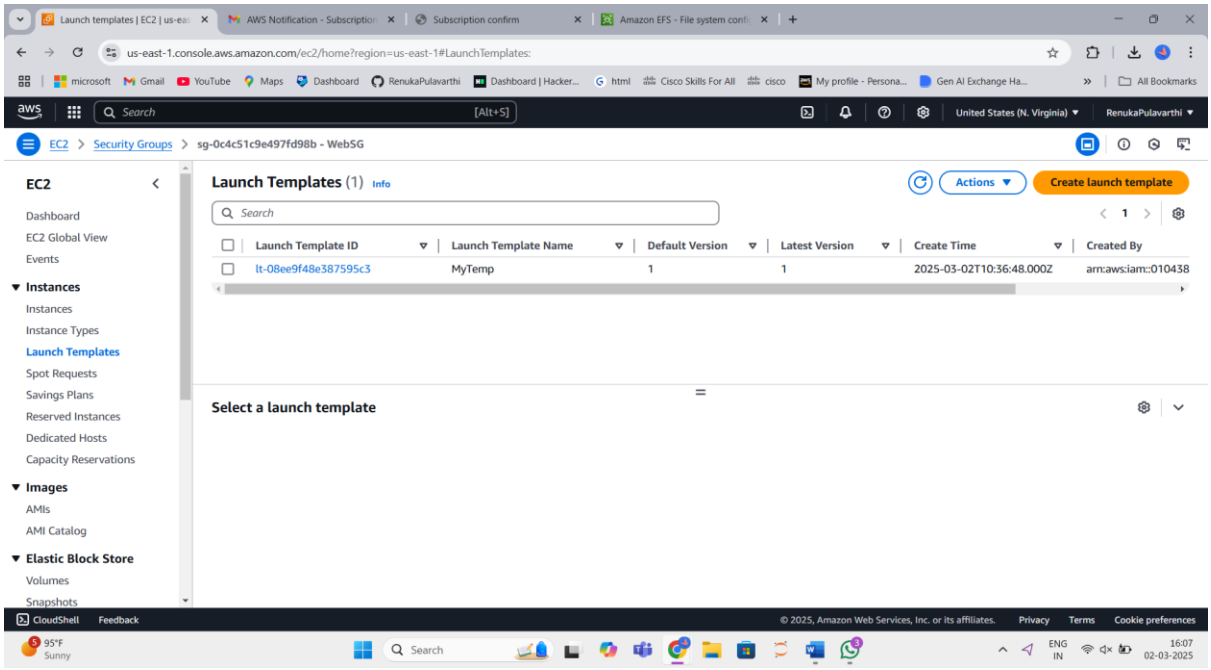
- Navigate to **EC2** > **Load Balancers**.
- Click **Create Load Balancer**.
- Select type (**Application/Network**).
- Assign **subnets, security groups**, and **target groups**.



3.15. Launching EC2 Instances (Using Launch Template)

Purpose: Provides reusable configurations for instance deployment.

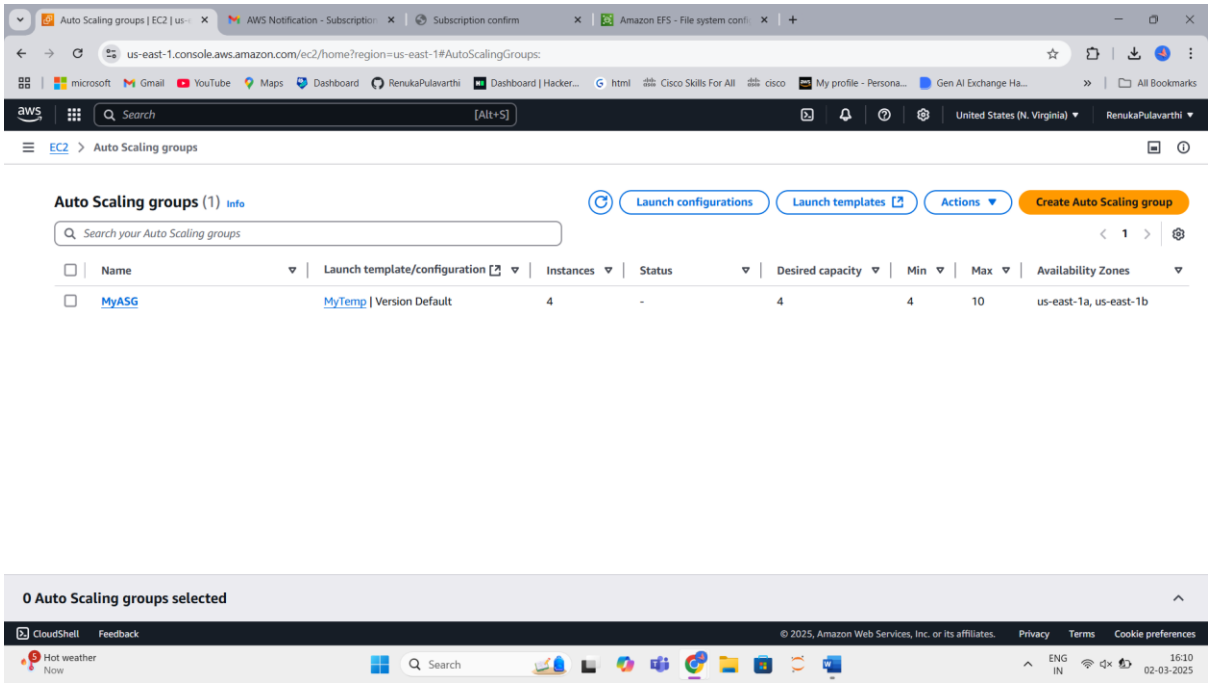
- Go to **EC2** > **Launch Templates**.
- Click **Create Template**.
- Define **AMI, instance type, security group, and IAM role**.

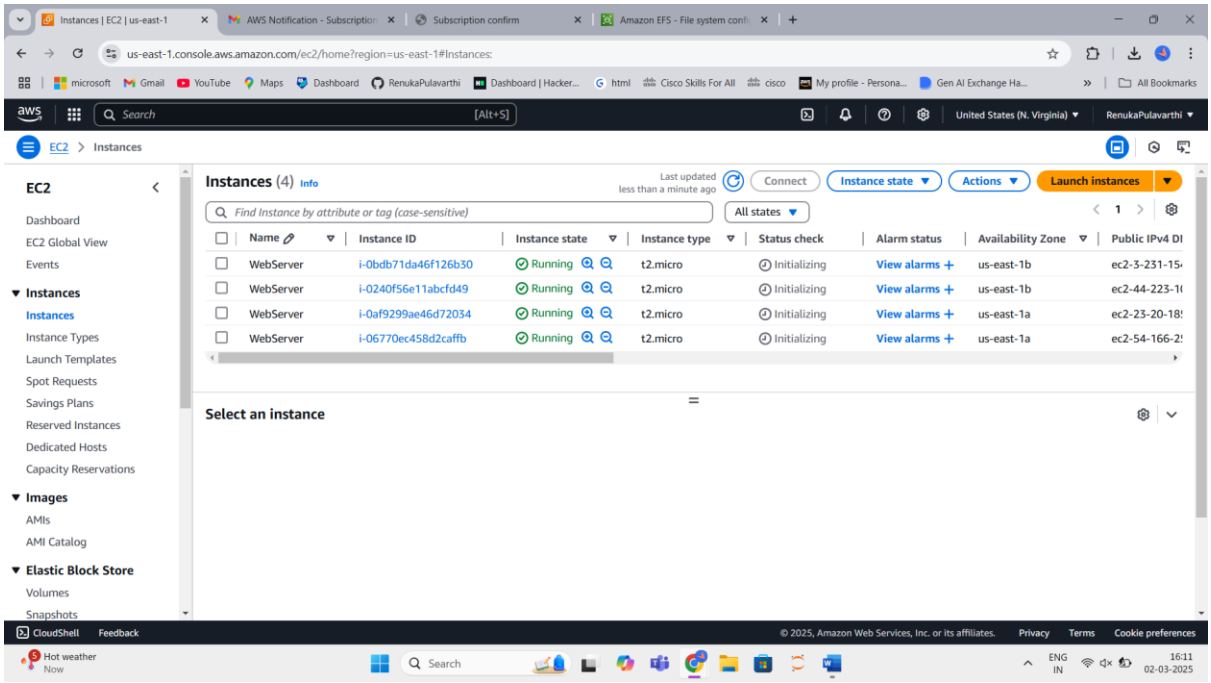


3.16. Auto Scaling Configuration

Purpose: Automatically adjusts the number of running instances based on demand.

- Navigate to **EC2** > **Auto Scaling Groups**.
- Click **Create Auto Scaling Group**.
- Choose the launch template.
- Set desired instance count and scaling policies.





3.17. Creating a Bastion Server

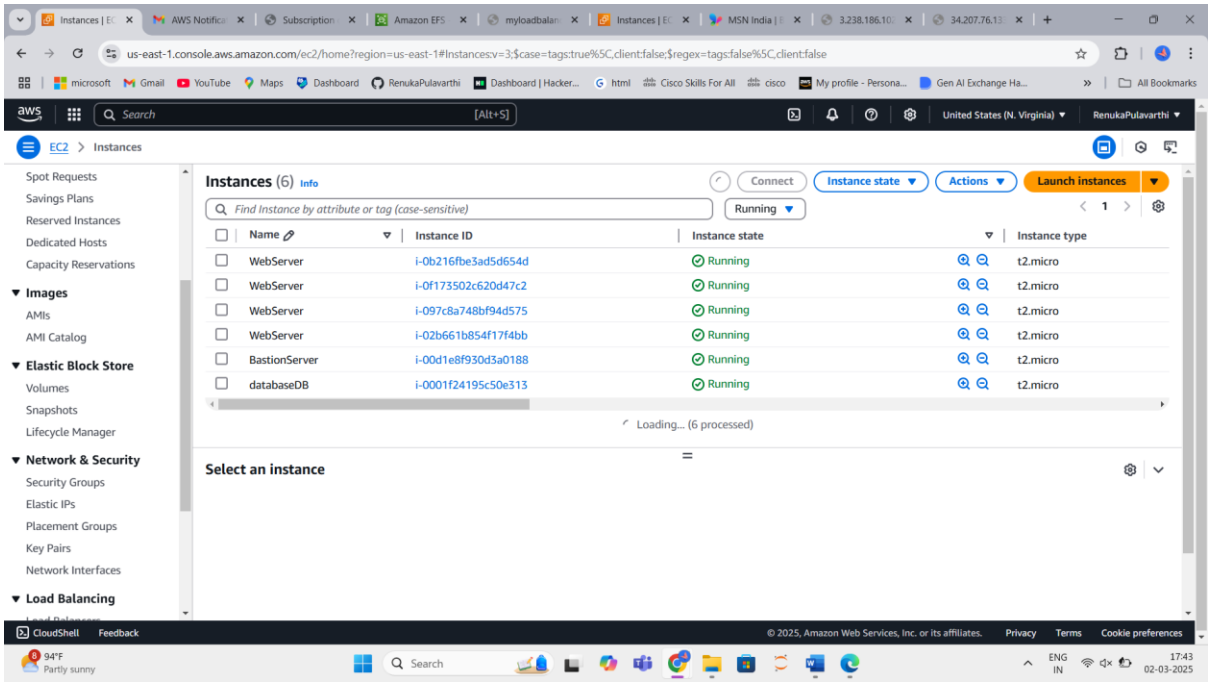
Purpose: Securely access instances in the private subnet.

- Launch an **EC2 instance** in a **public subnet**.
- Allow **SSH access** from trusted IPs.
- This acts as a secure gateway to the private network.

3.18. Creating a Database Server

Purpose: Provides backend data storage for applications.

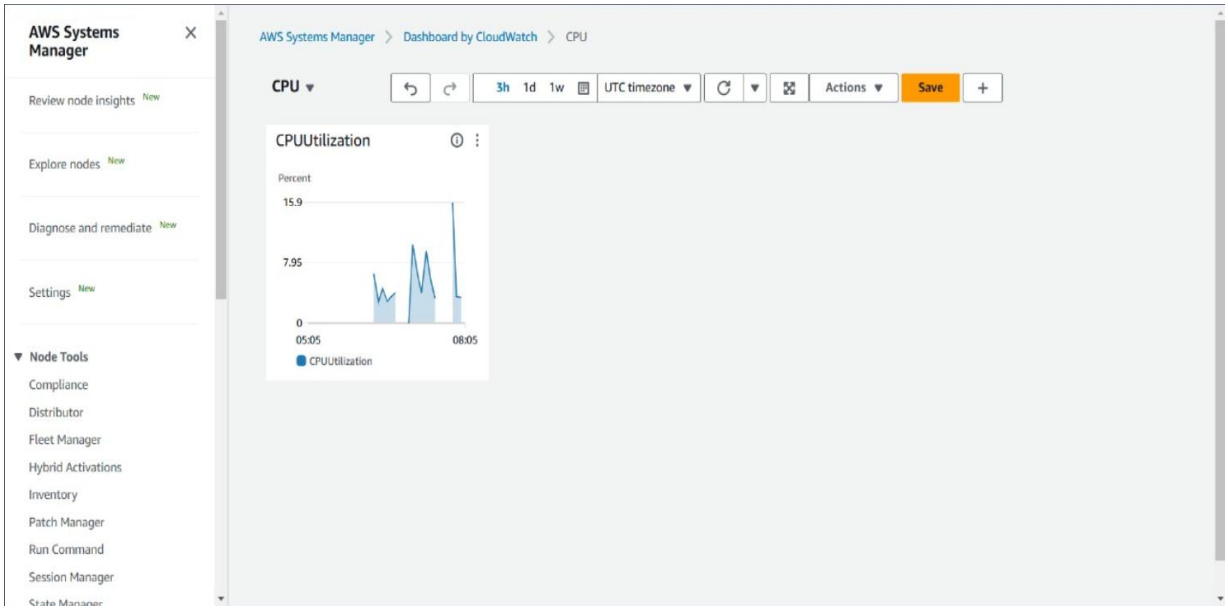
- Launch an **RDS instance** or an **EC2 database server** in the **private subnet**.
- Configure security group for restricted access.



3.19. Configuring CloudWatch

Purpose: Enables monitoring and alerting for AWS resources.

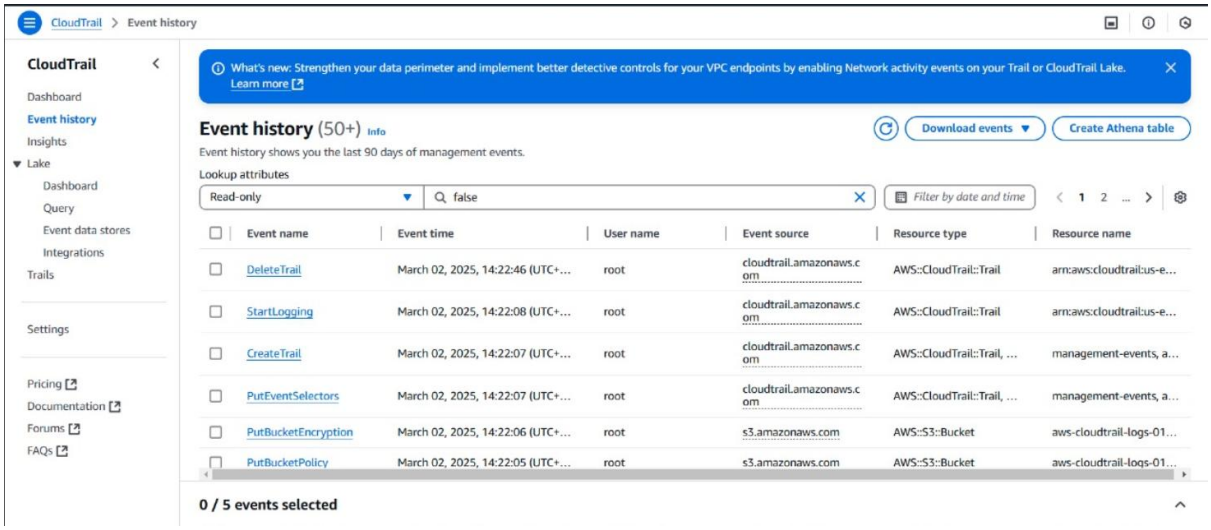
- Navigate to **CloudWatch**.
- Set up **logs, metrics, and alarms** for resources.



3.20. Configuring CloudTrail

Purpose: Records AWS API calls for security and compliance.

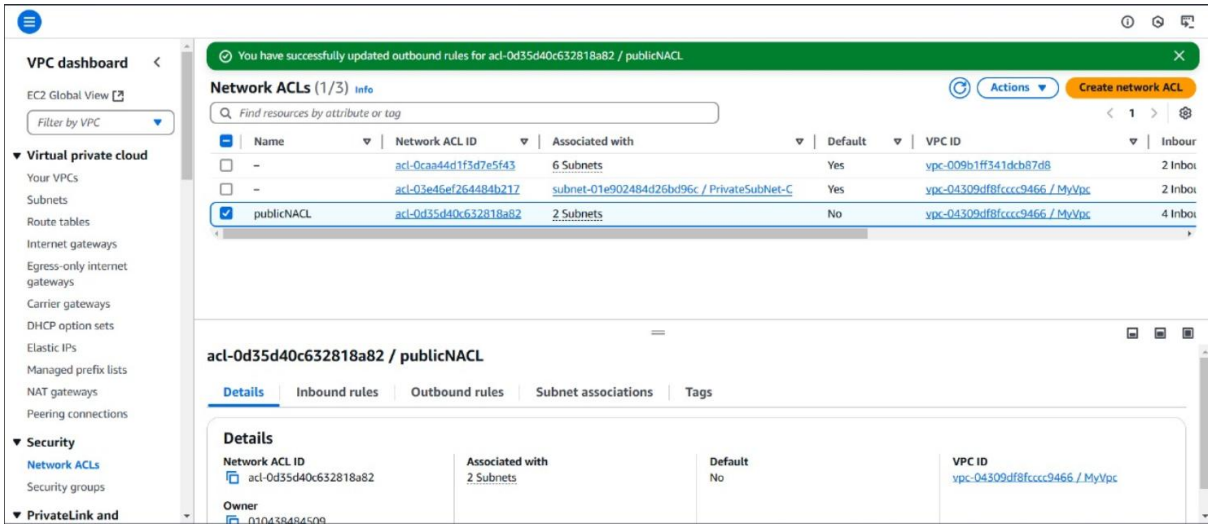
- Go to **CloudTrail**.
- Click **Create Trail**.
- Configure logging for **AWS API events**.



3.21. Creating Network ACL (NACL)

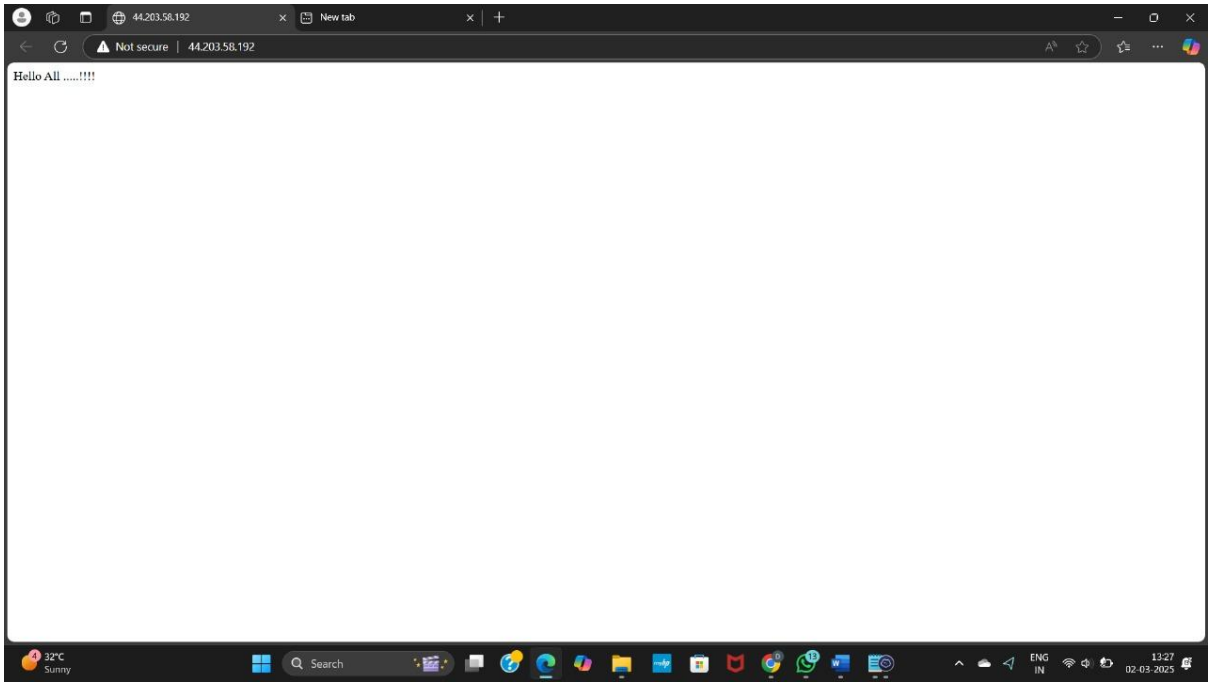
Purpose: Controls traffic at the subnet level.

- Navigate to **VPC** > **Network ACLs**.
- Create and associate it with subnets.
- Define **rules** for **inbound and outbound traffic**.

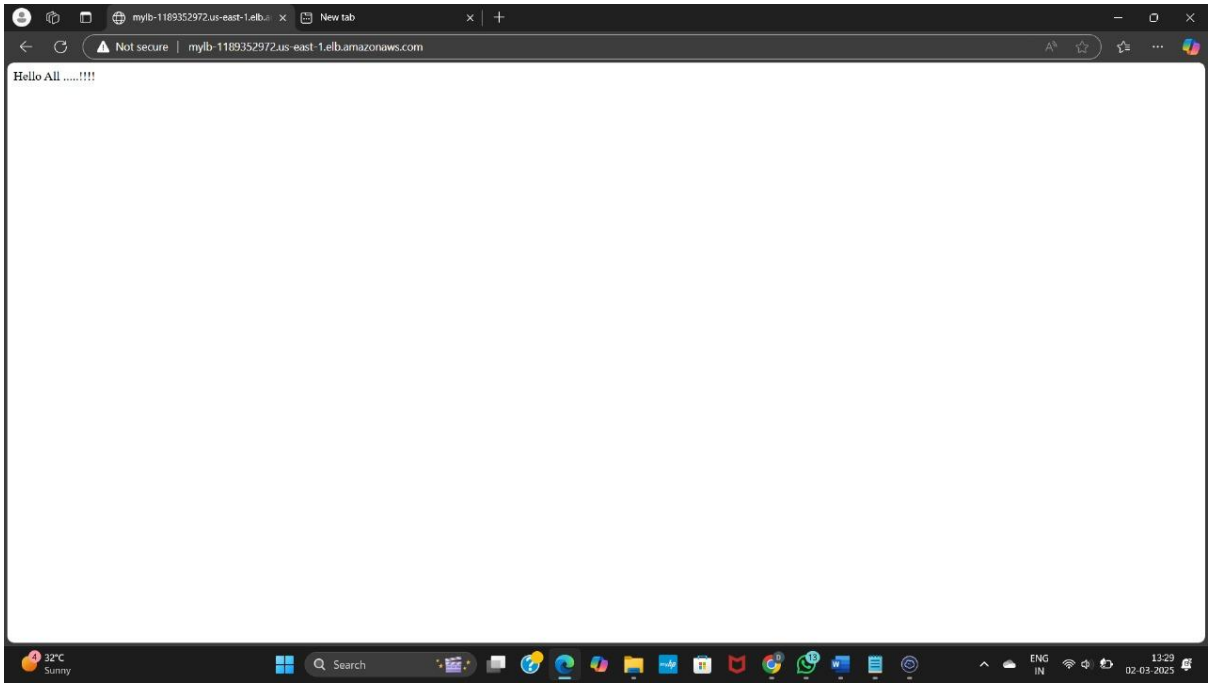


4.Testing

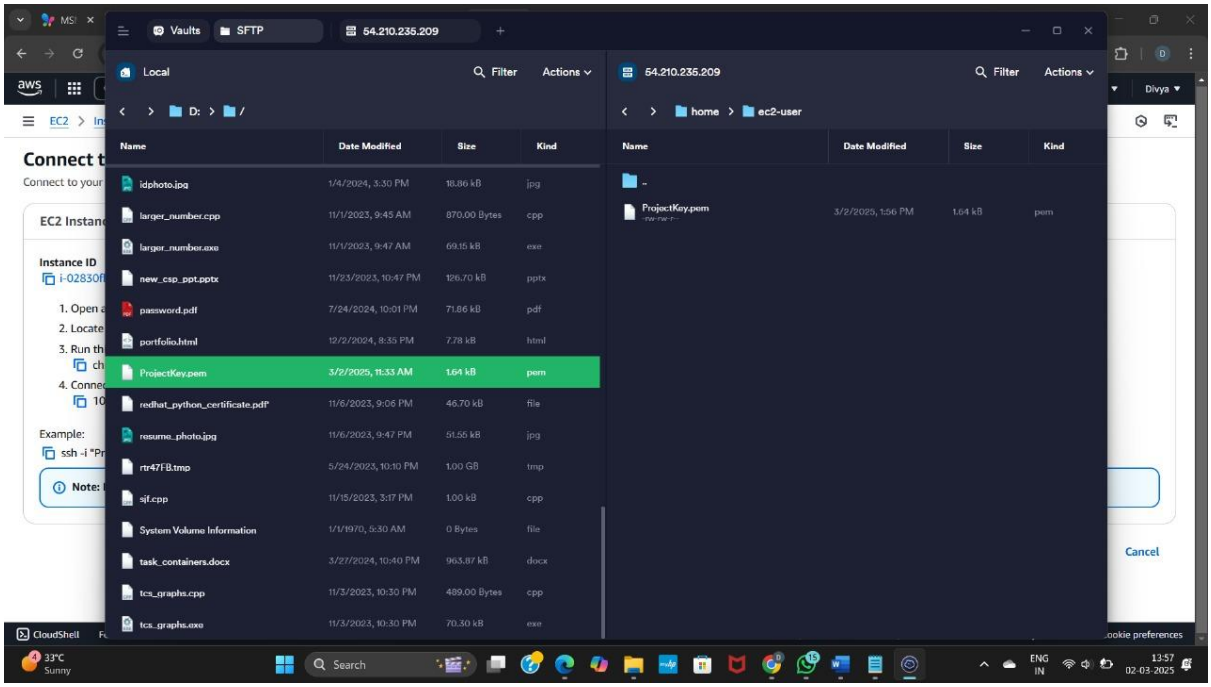
4.1. Testing Web Server

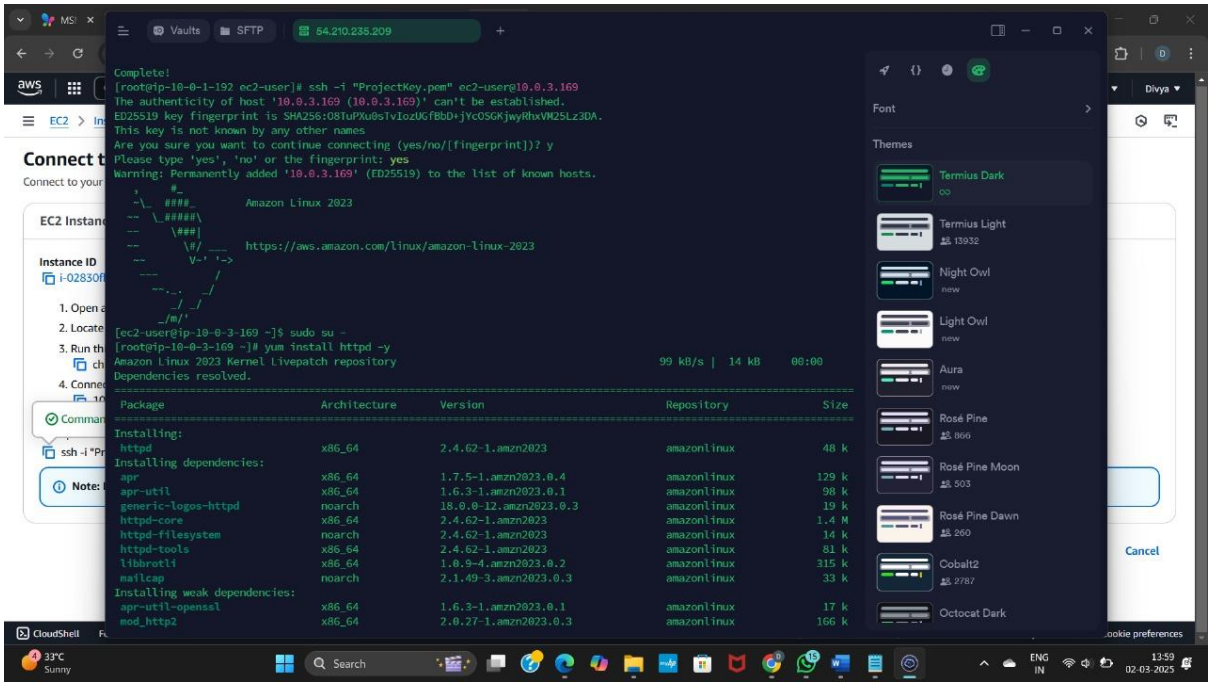


4.2. Testing Load Balancers



4.3. Testing Bastion Servers





5.Conclusion

Building a well-architected AWS infrastructure requires careful planning and execution to ensure security, scalability, and high availability. This guide provides a structured approach to creating a robust cloud environment with key components such as VPC, subnets, EC2 instances, load balancing, auto-scaling, and monitoring solutions.

By implementing these steps, organizations can achieve a highly efficient and secure cloud setup, enabling seamless operations while optimizing costs. The combination of CloudWatch, CloudTrail, and IAM policies enhances visibility and control over resources, while services like S3, EFS, and SNS improve data management and communication.

With this AWS infrastructure in place, businesses can focus on innovation and growth while leveraging the power of cloud computing to drive efficiency and reliability. Whether for hosting applications, databases, or enterprise solutions, this setup provides a solid foundation for cloud-based success.