Experiment No 08

Aim:
To design and deploy a secure Virtual Private Cloud (VPC) on Amazon Web Services (AWS) that provides isolated networking resources for hosting and managing IoT services.

Tools Required:

- AWS Management Console / AWS CLI
- AWS VPC Service (Virtual Private Cloud)
- Subnets (Public & Private)
- Route Tables, Internet Gateway (IGW), NAT Gateway
- Security Groups & Network ACLs
- AWS IoT Core
- EC2 Instance for IoT device simulation

Theory:
A Virtual Private Cloud (VPC) on AWS creates a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define. VPCs offer deep control over networking, including selection of IP address ranges, creation of subnets, and configuration of route tables and network gateways.

Importance for IoT:

IoT services handle sensitive device data and demand secure, robust infrastructure. Deploying these services inside a VPC safeguards them through network-level isolation and robust access control. Within a VPC, public and private subnets can separate network-sensitive resources (e.g., databases) from those needing public internet access. Routing and firewalls—via route tables, security groups, and network ACLs—determine how and with whom resources communicate.

- **Internet Gateway (IGW):** Attach to a public subnet for outbound and inbound internet access.
- **NAT Gateway:** Allows resources in private subnets to access the internet securely, without exposing them to inbound connections.
- **Security Groups & NACLs:** These virtual firewalls rigorously control traffic into and out of your AWS resources.

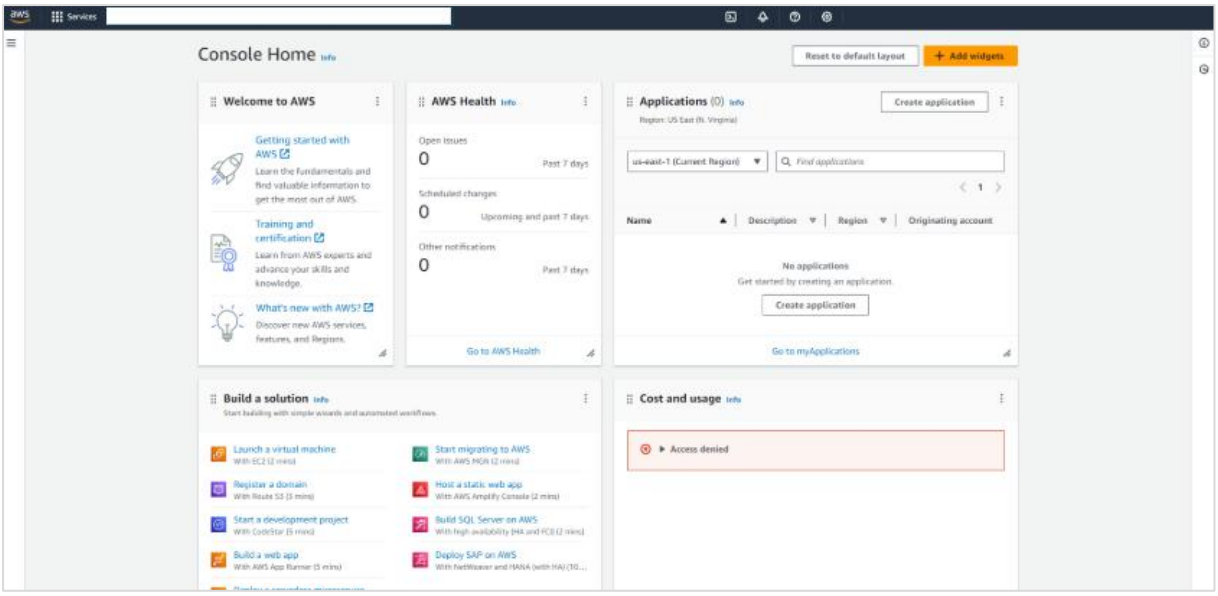
- **AWS IoT Core Integration:** VPC provides the backbone for secure, scalable IoT data collection and processing, as IoT devices and cloud endpoints communicate over regulated, encrypted paths.

Key Benefits:

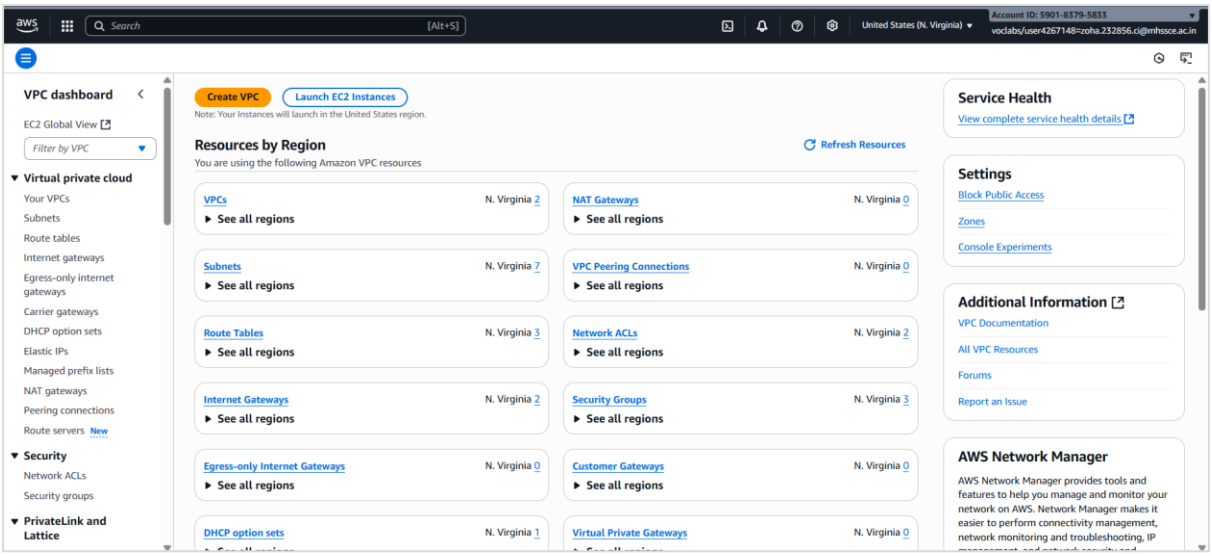
- Network segmentation and isolation for security
- Flexible routing and controlled access
- Compliance with security standards for sensitive workloads
- Easy scaling and reliable connectivity for diverse IoT scenarios

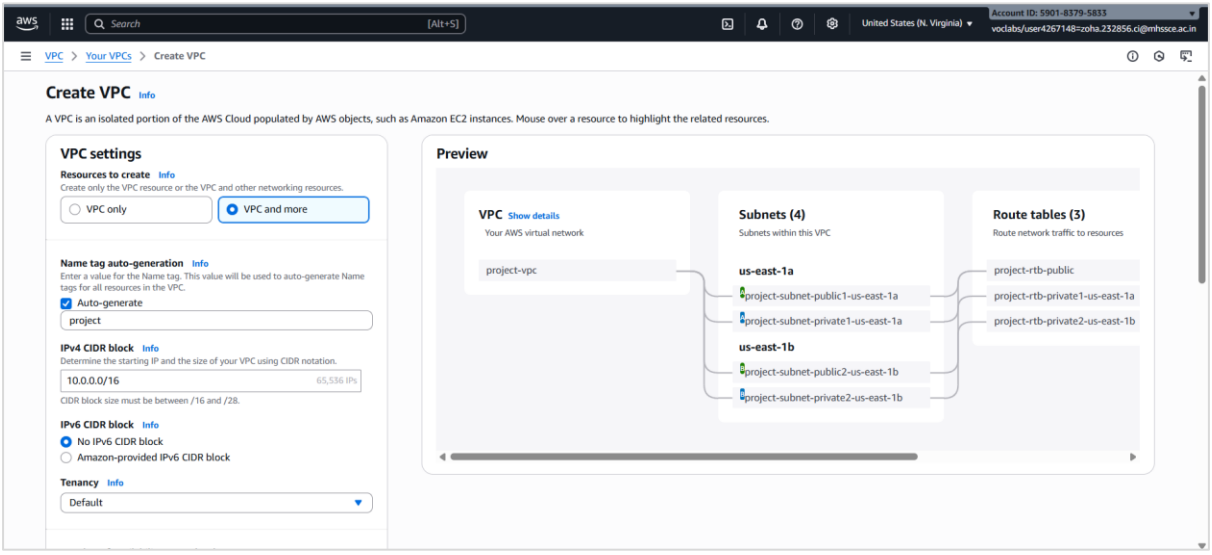
Procedure:

1. **Login to AWS Console:**
   - Sign in to your AWS account and access the AWS Management Console.



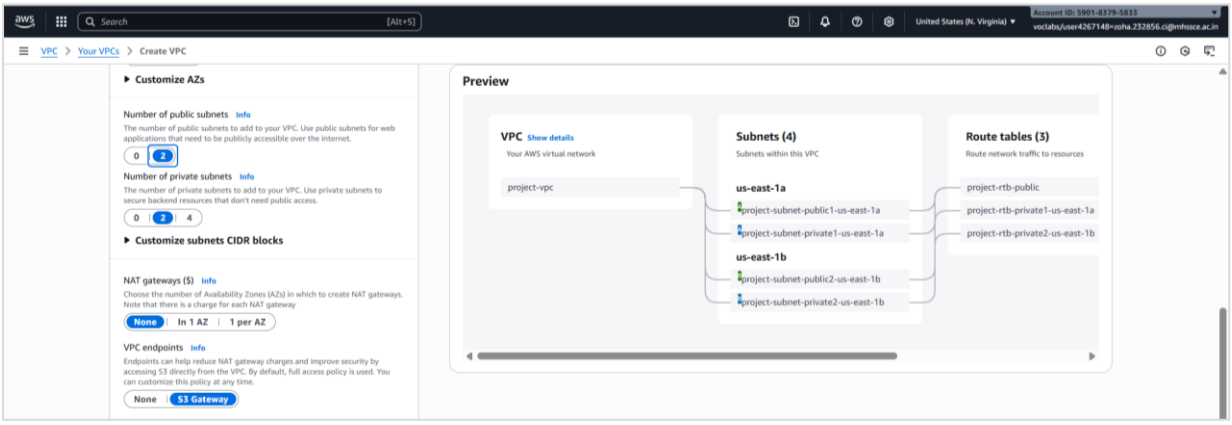
2. **Create a VPC:**
   - Go to the VPC Dashboard, click "Create VPC", and select the preset to include one public and one private subnet.
   - Assign an appropriate IPv4 CIDR block (e.g., 10.0.0.0/16).

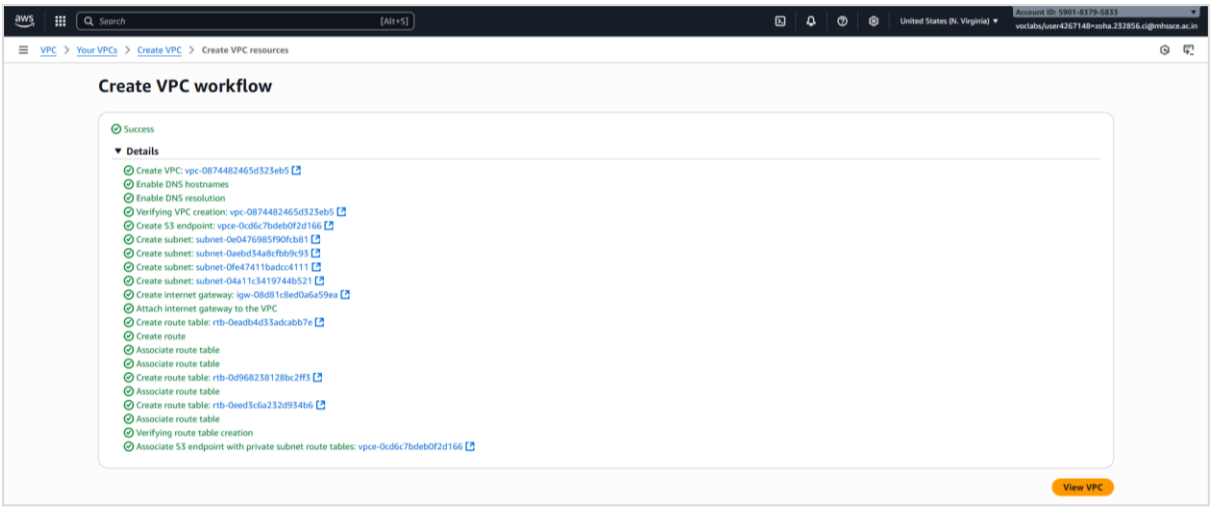


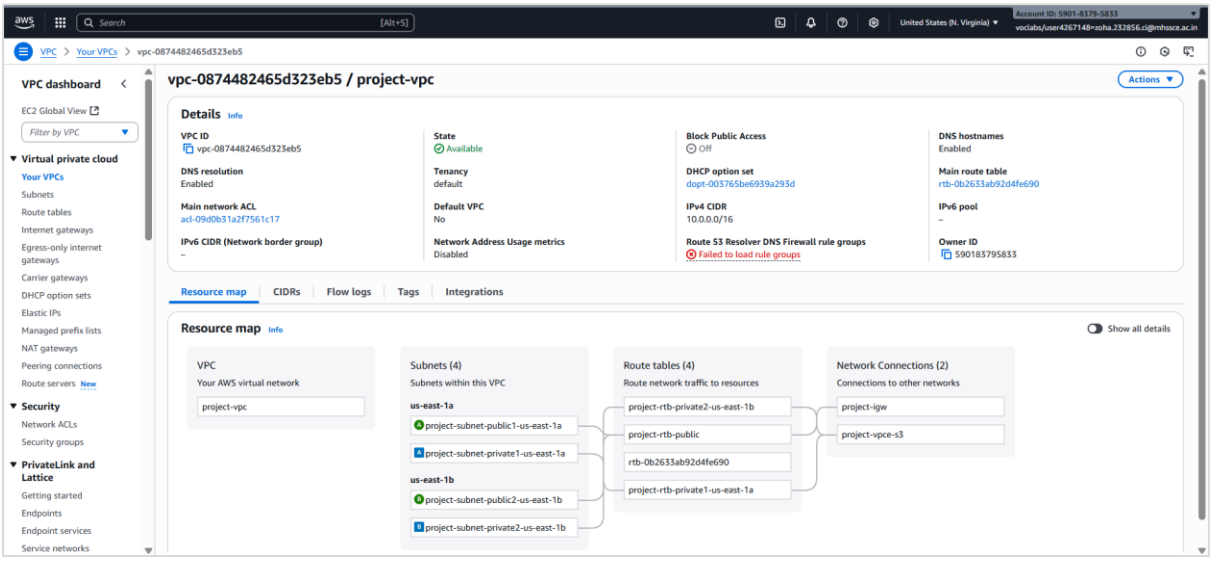


3. Configure Subnets:

- Create a public subnet (e.g., 10.0.1.0/24) for exposing the IoT gateway or EC2 instance.
- Create a private subnet (e.g., 10.0.2.0/24) for backend processing services.

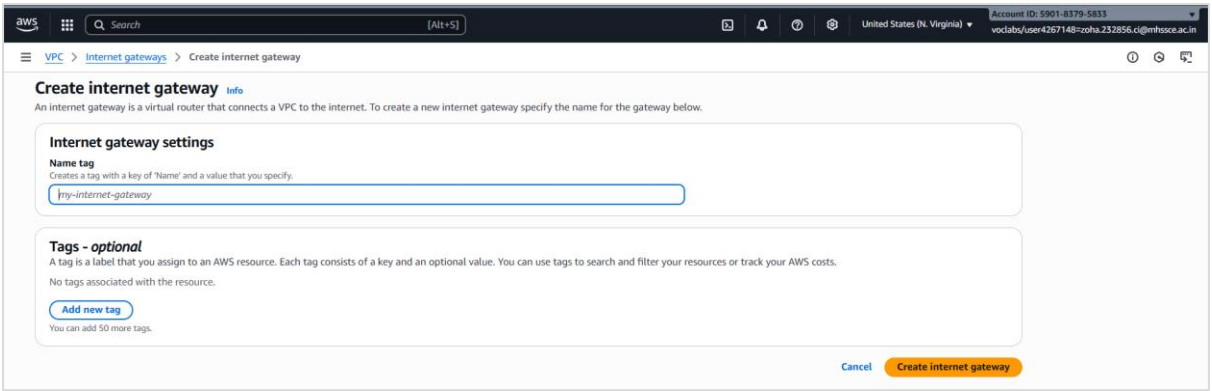






4. Attach Internet Gateway:

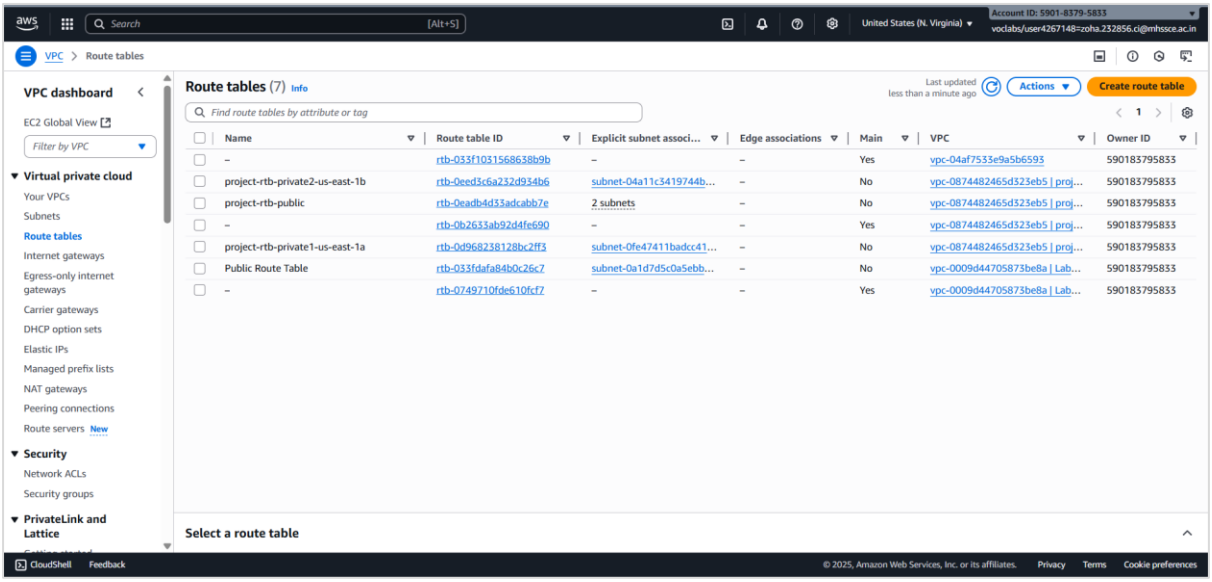
- Create and attach an IGW to the VPC.
- Attach the IGW to the public subnet for internet connectivity.



5. Set Up Route Tables:

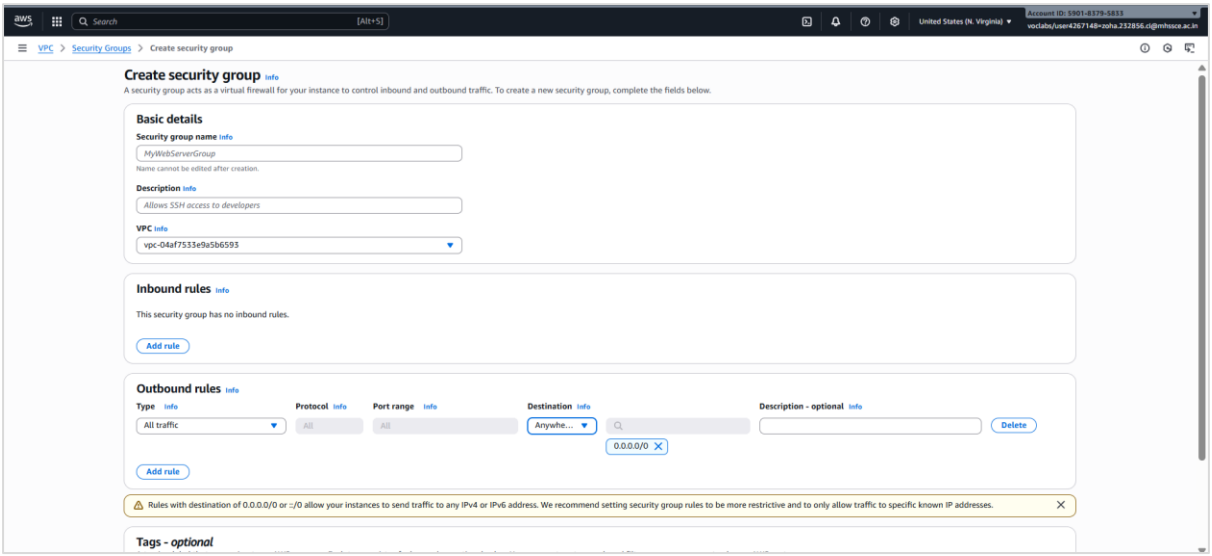
- Configure the route table for the public subnet to allow traffic through the IGW.

- Associate the private subnet with a NAT Gateway (optional) for secure outbound internet access.



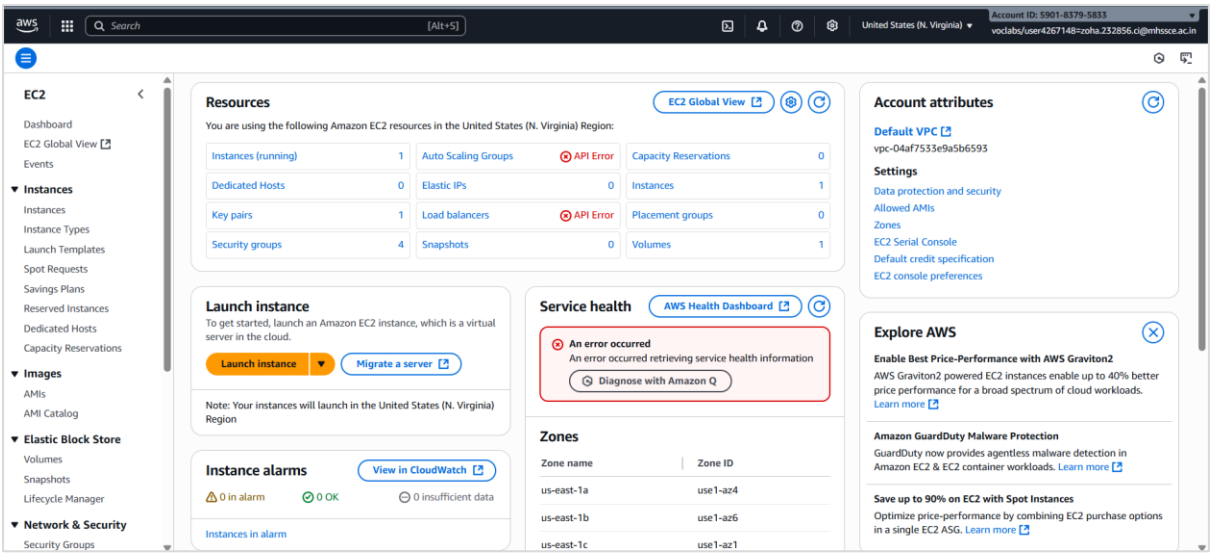
6. Configure Security Groups:

- Define security group rules to allow necessary traffic (e.g., MQTT ports 1883/8883 for IoT, HTTPS port 443 for cloud APIs).



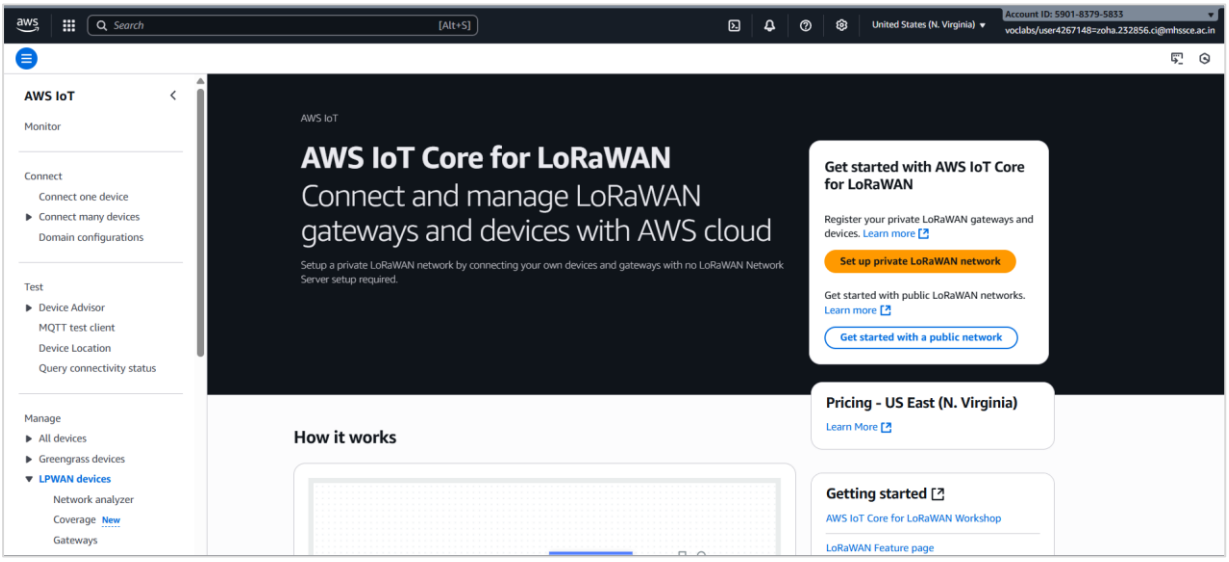
7. Launch EC2 Instance:

- Deploy an EC2 instance within the public subnet for IoT gateway simulation.
- Configure the instance to communicate securely with AWS IoT Core.



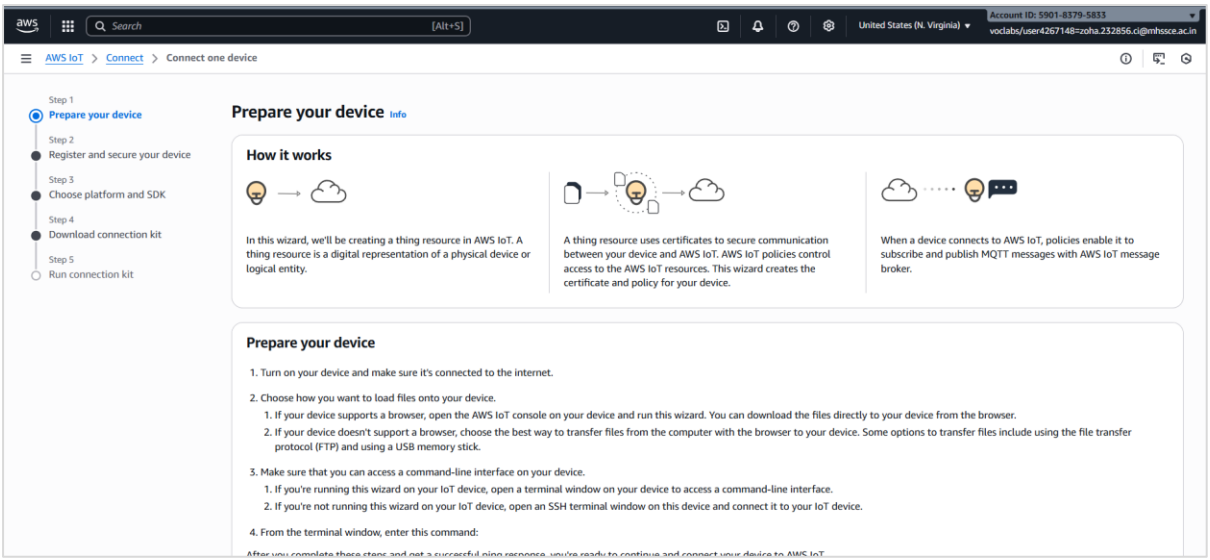
8. Integrate with AWS IoT Core:

- Register a virtual IoT device in AWS IoT Core.
- Generate security certificates and attach the correct policies.
- Configure the EC2 gateway or a simulated device to publish and subscribe to IoT messages as per requirements.



9. Test & Validate Communication:

- Use the AWS IoT MQTT test client to verify secure communication between your EC2 instance (or IoT simulator) and AWS IoT Core endpoints via your VPC.



Result:

A secure VPC was successfully deployed and configured with public and private subnets, IGW, route tables, and robust security groups. The implemented architecture ensured that IoT devices and backend services could communicate securely and efficiently, with strict control over network exposure.

Conclusion:

This experiment demonstrated the deployment of an AWS VPC tailored for IoT scenarios, emphasizing security, isolation, and flexibility. Utilizing subnet segmentation, routing, and security configurations, the experiment established a scalable cloud network that met the needs of modern IoT services while mitigating risks associated with public network exposure. A properly designed VPC infrastructure is fundamental to establishing secure, reliable, and manageable cloud environments for real-world IoT applications.