**Project name: Set up a multi-region disaster recovery architecture using AWS**

**GitHub repo name:** <https://github.com/Mithra1995/aws-disaster-recovery-terraform.git>

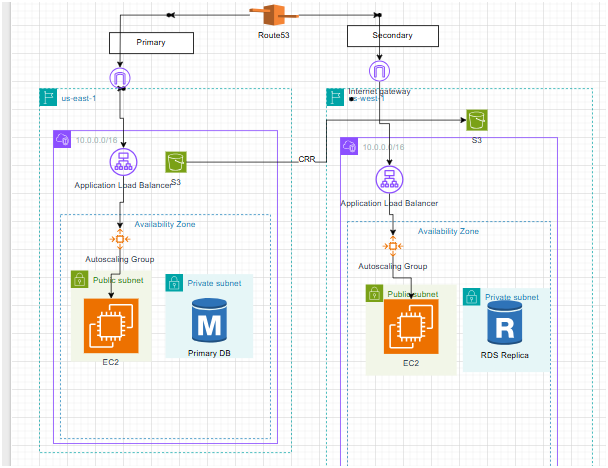
# Objective

To design and implement a multi-region disaster recovery solution using AWS services, where infrastructure is deployed in two AWS regions. The solution will replicate data across regions, ensure high availability, and implement a failover mechanism for disaster recovery.

# Architecture Overview

The architecture involves provisioning AWS resources across two regions to create a highly available and resilient infrastructure. Key components include Amazon VPC for networking, Amazon S3 for cross-region data replication, Amazon RDS for multi-region database deployment, and Route 53 for DNS failover routing. The solution ensures that both regions are synchronized and can automatically switch to the backup region if needed.

# Architecture Diagram



# Infrastructure setup

✅ VPCs in us-east-1 (Primary) and us-west-2 (Secondary)

✅ Subnets (Public/Private) in multiple Availability Zones

✅ Internet Gateways & NAT Gateways

✅ Route Tables

✅ S3 Buckets with cross-region replication

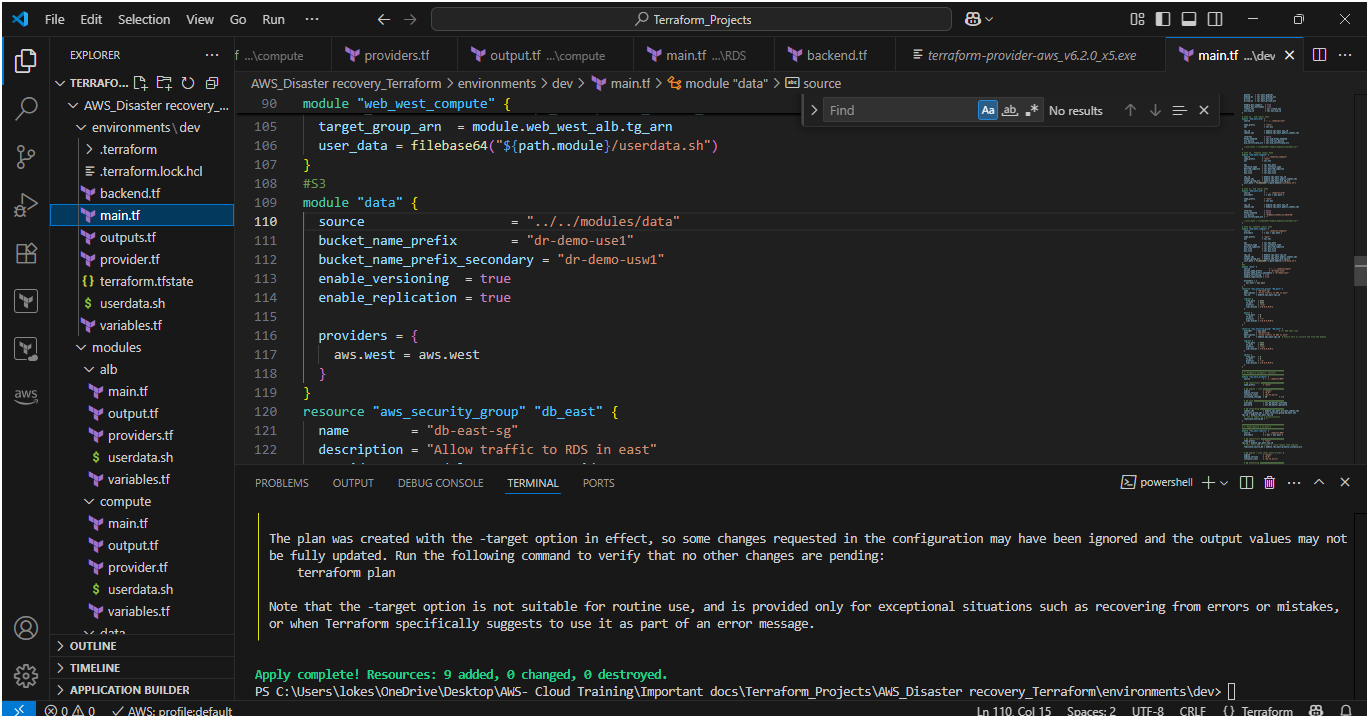
✅ RDS instances with multi-region replication

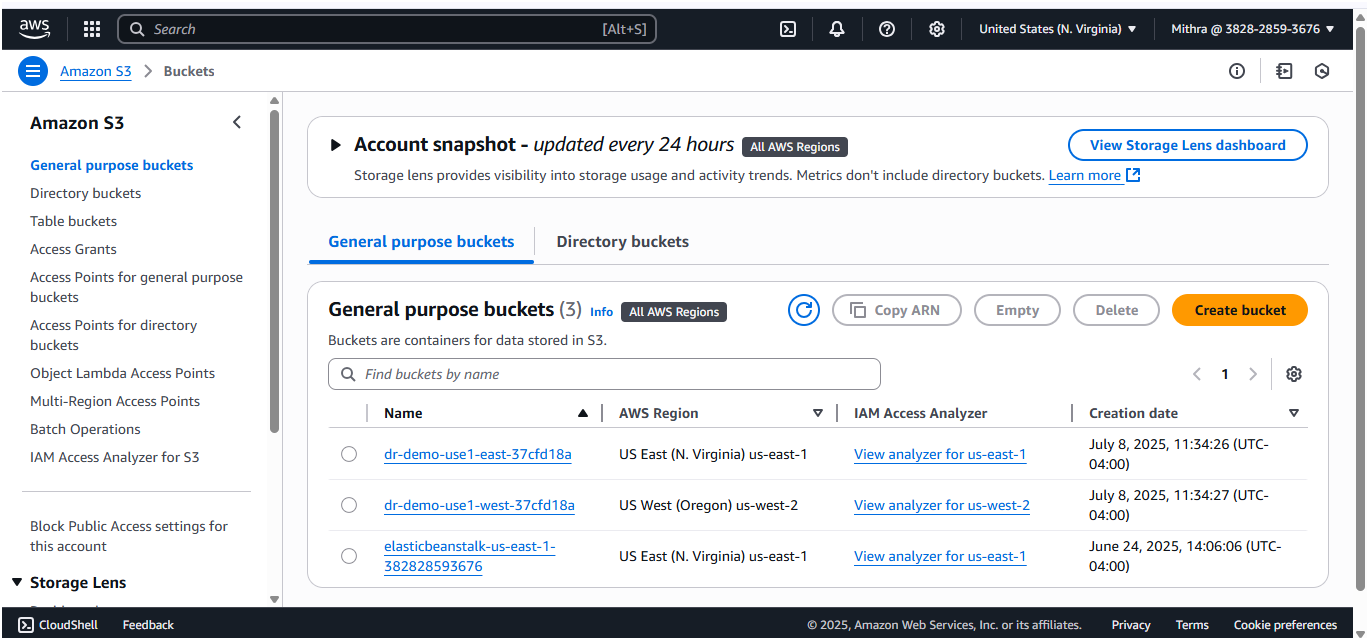
✅ Route 53 with DNS failover

✅ Auto Scaling Groups and Load Balancers

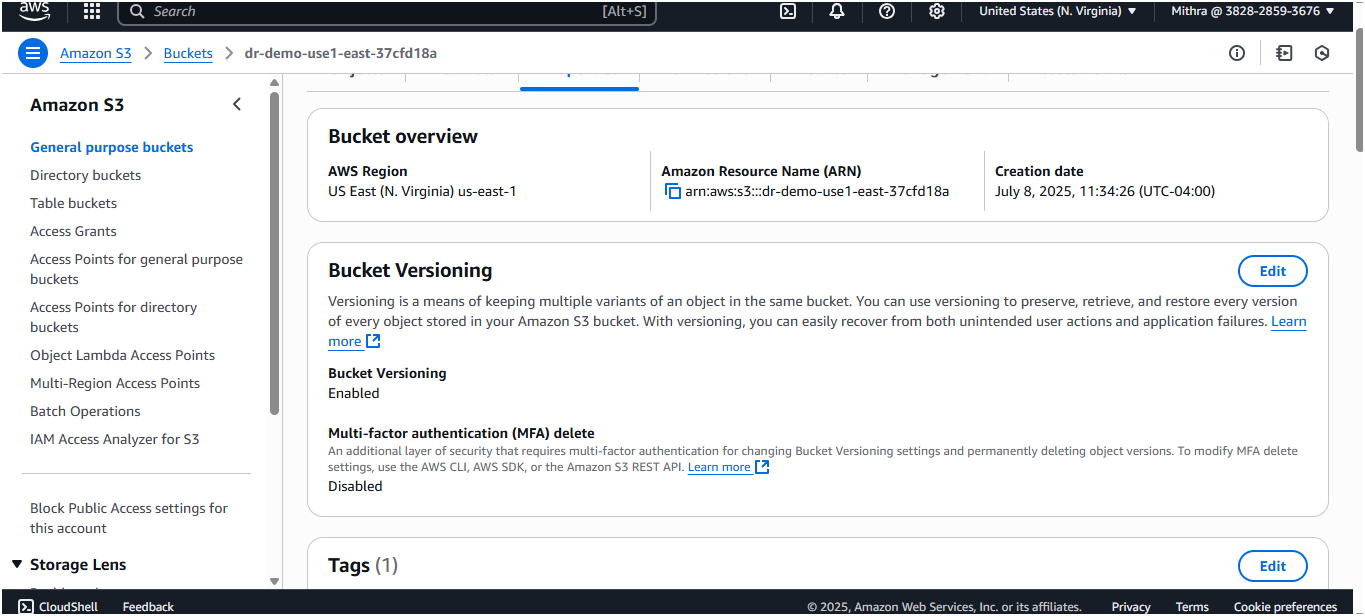
# Step by step process:

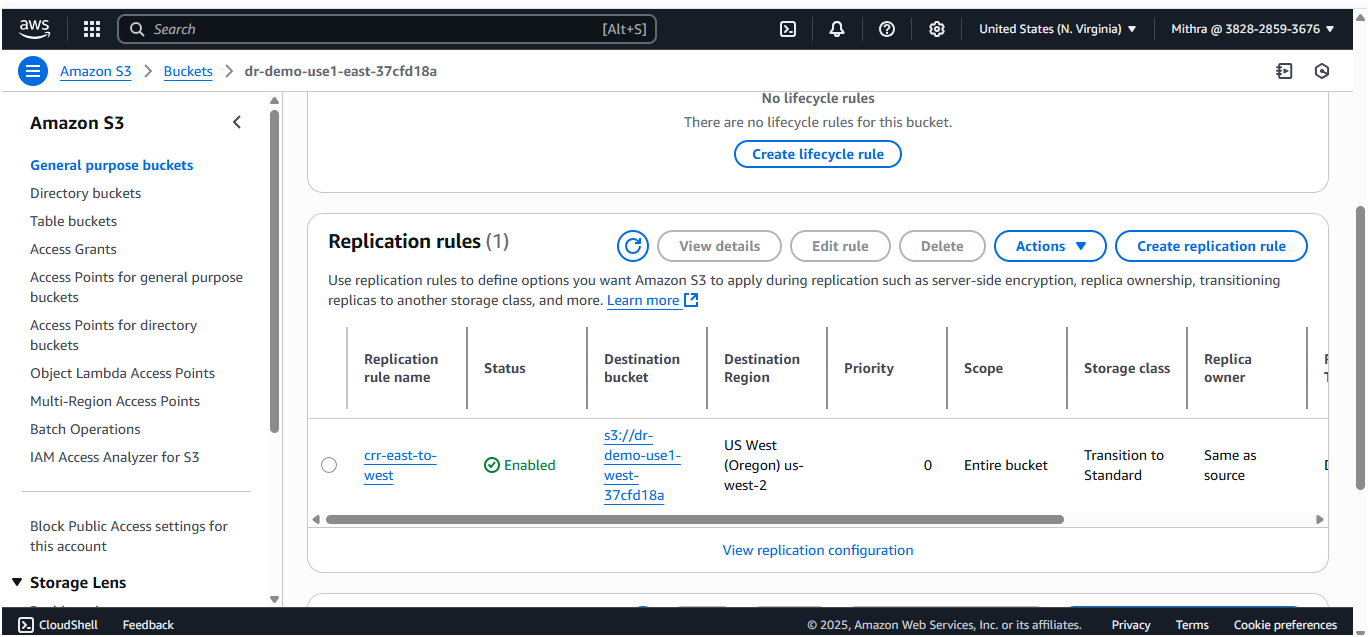
S3 bucket creation in both us-east and us-west region

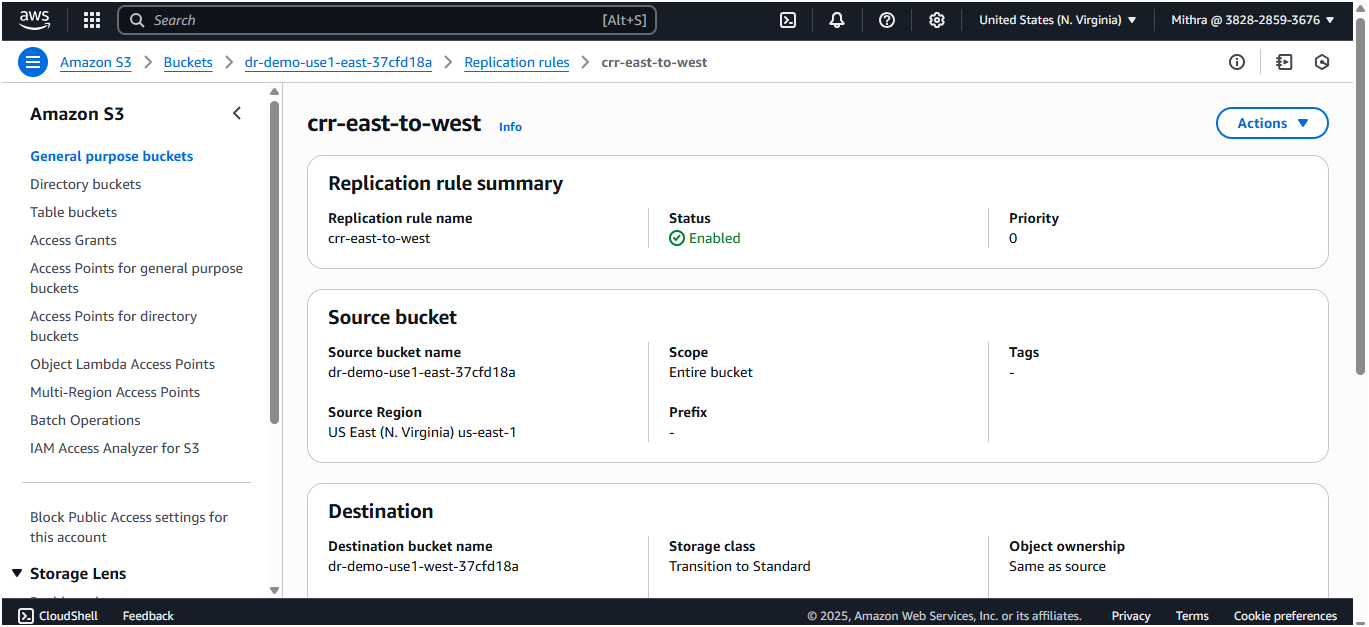




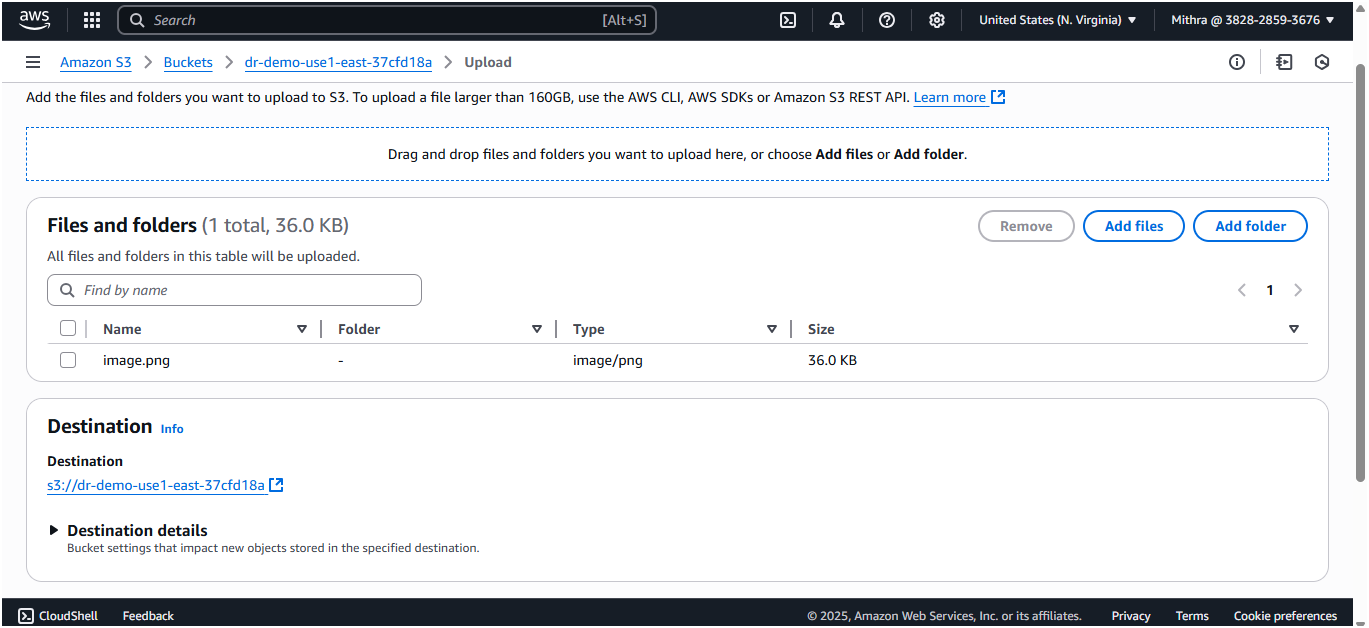
Now I will upload the image in us-east region and incorporate that content to launch template

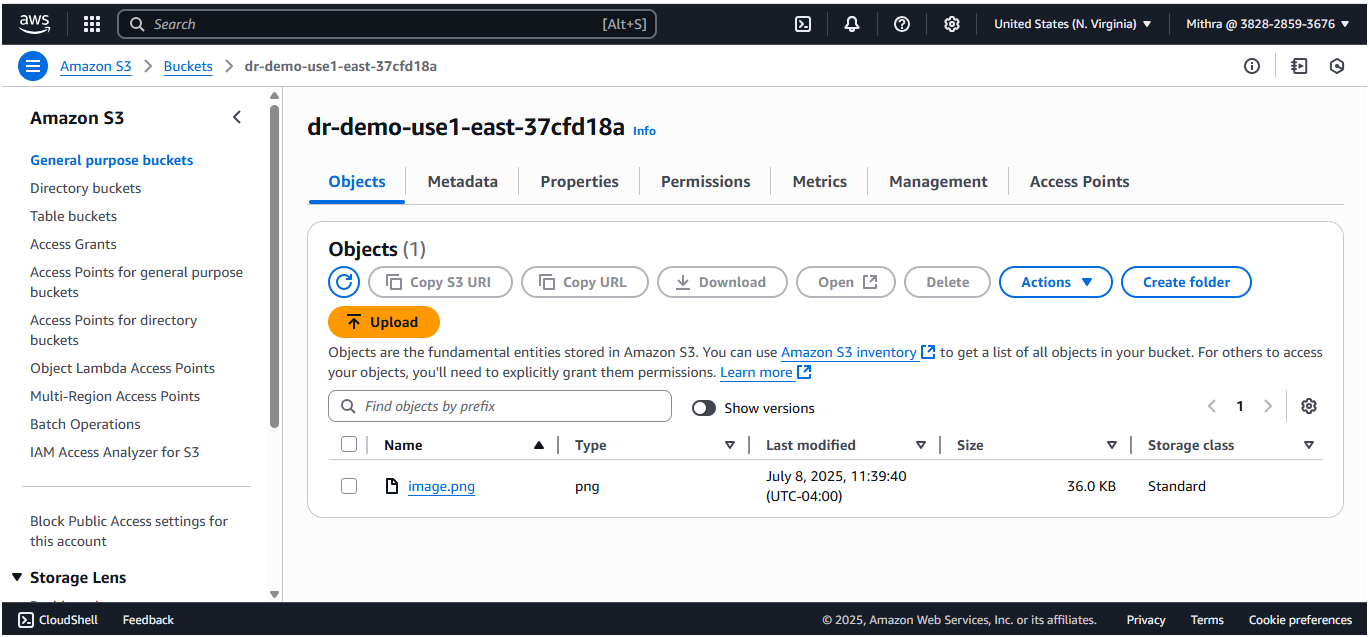




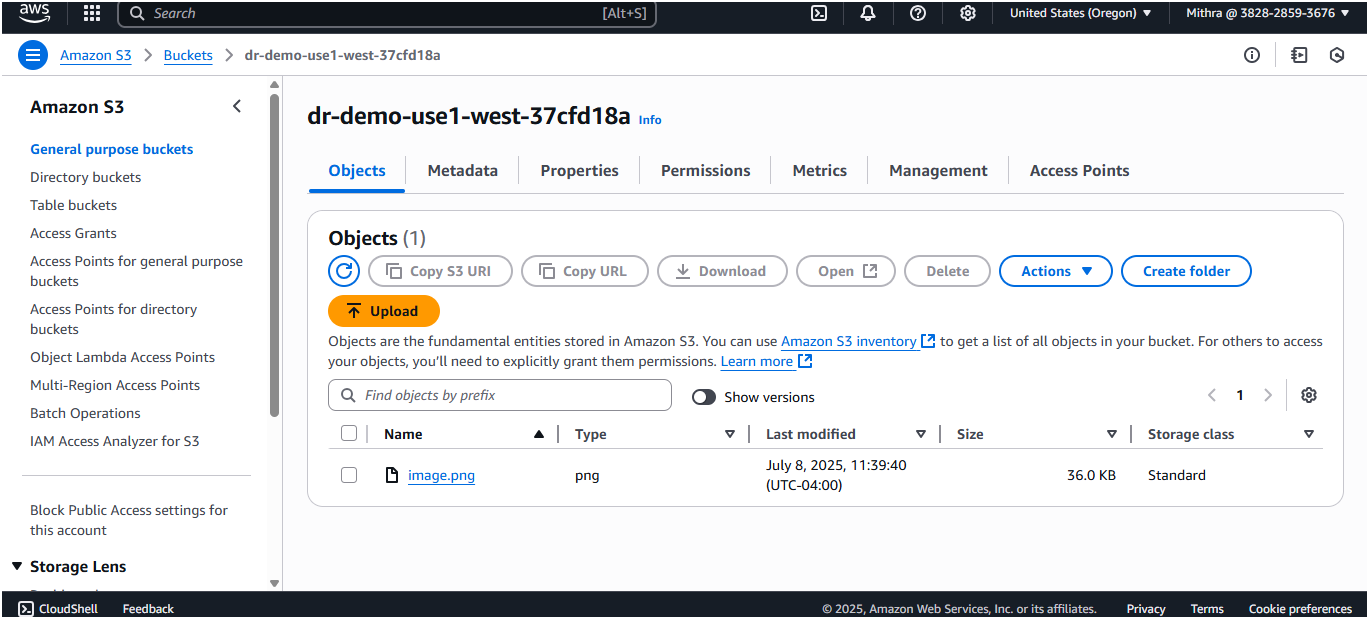


Upload image in east region

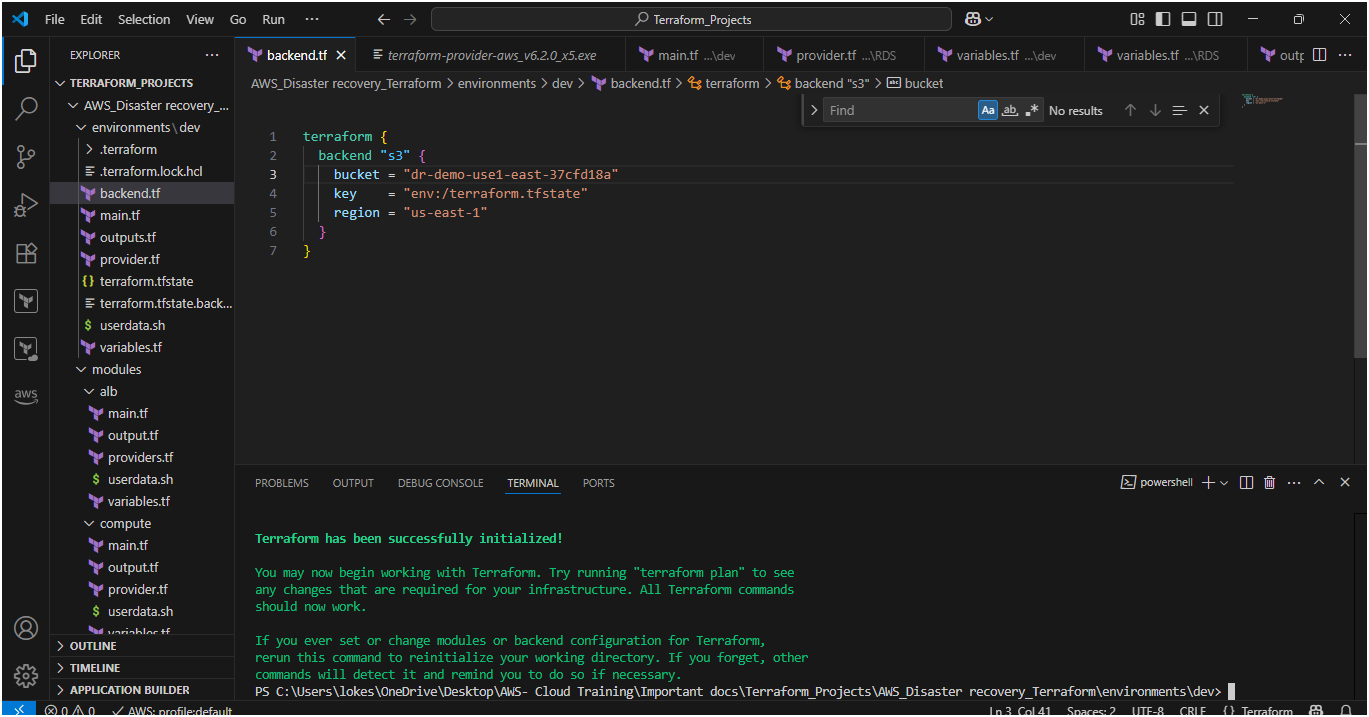




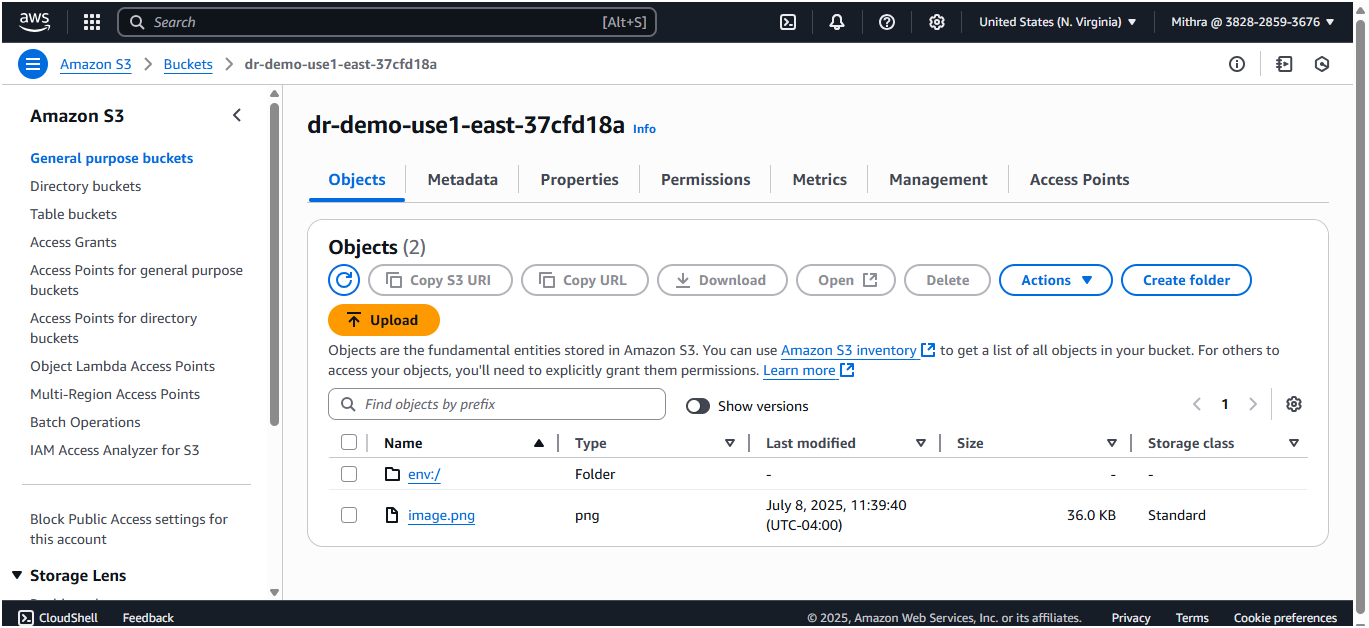
The image got replicated in west region as well

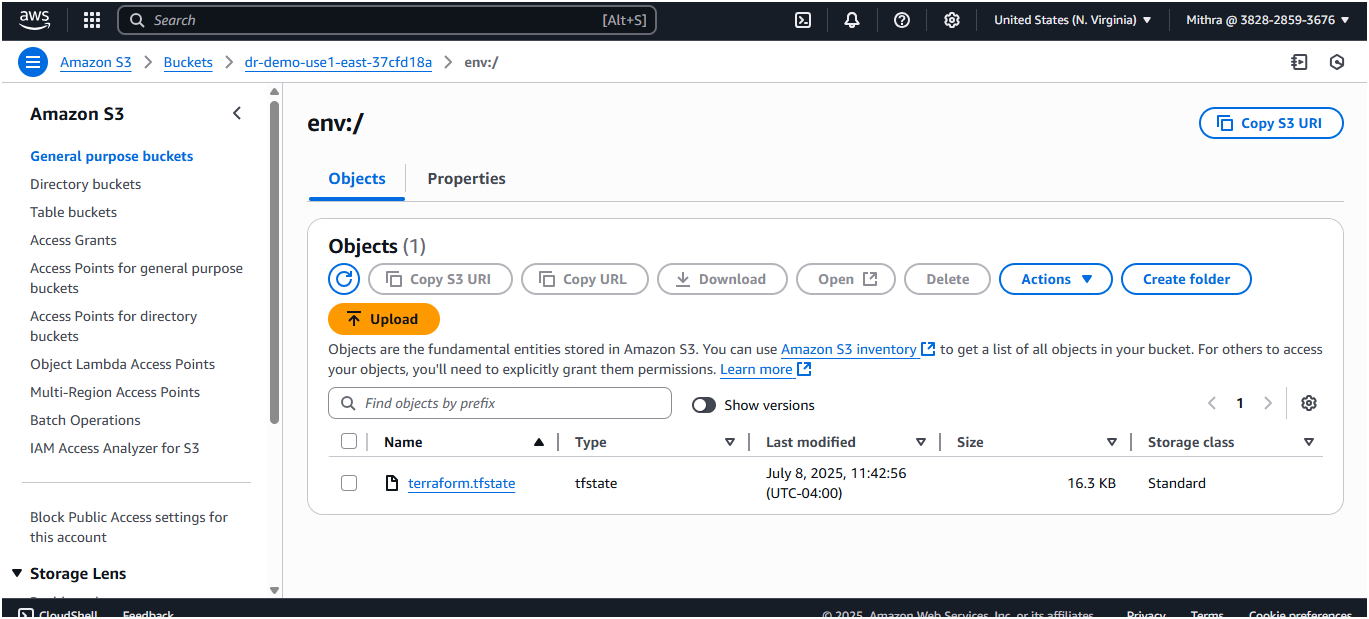


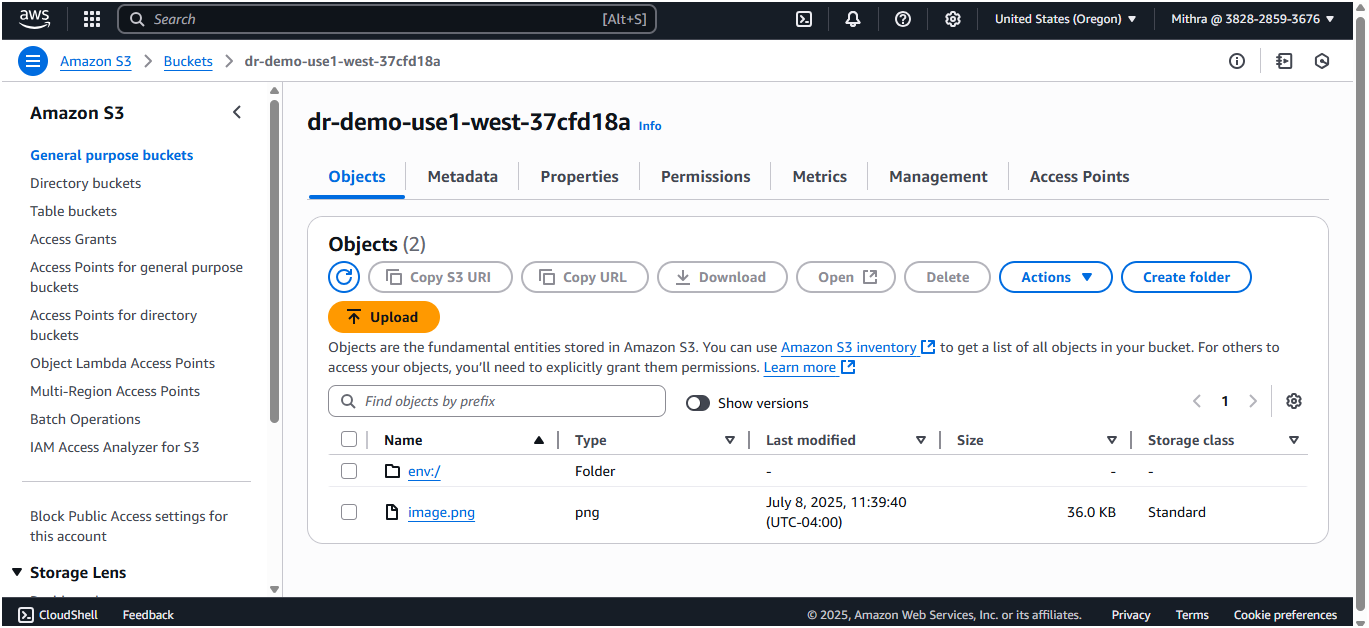
Now I am enabling the s3 backend to store my state file



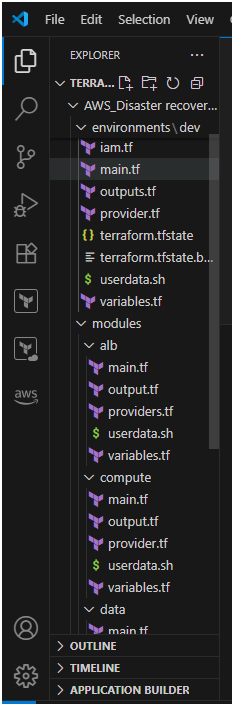
And my terraform state file got stored in both the regions

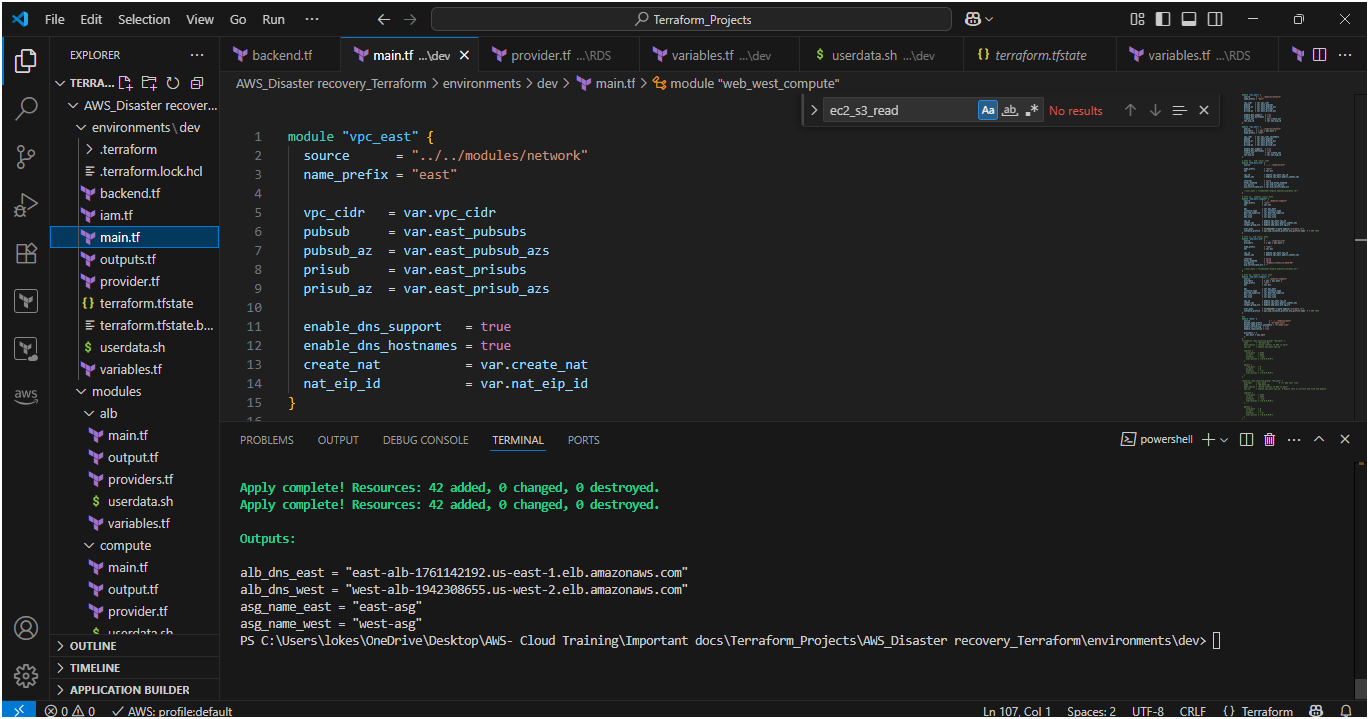


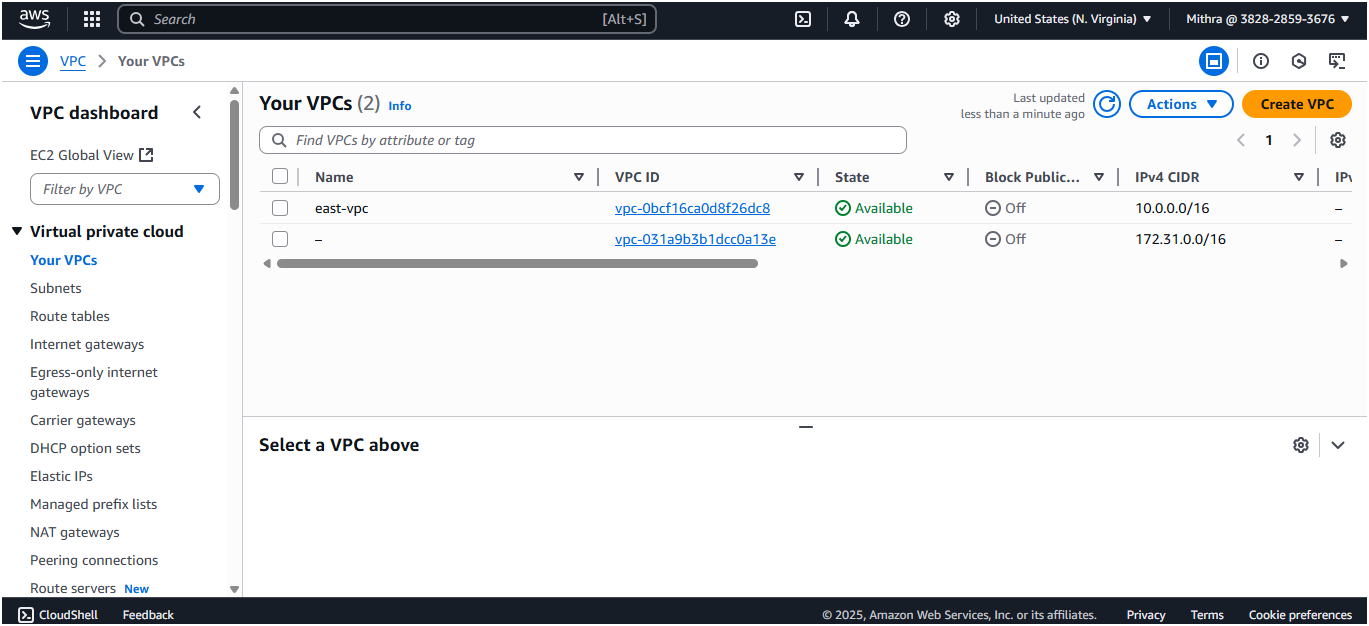


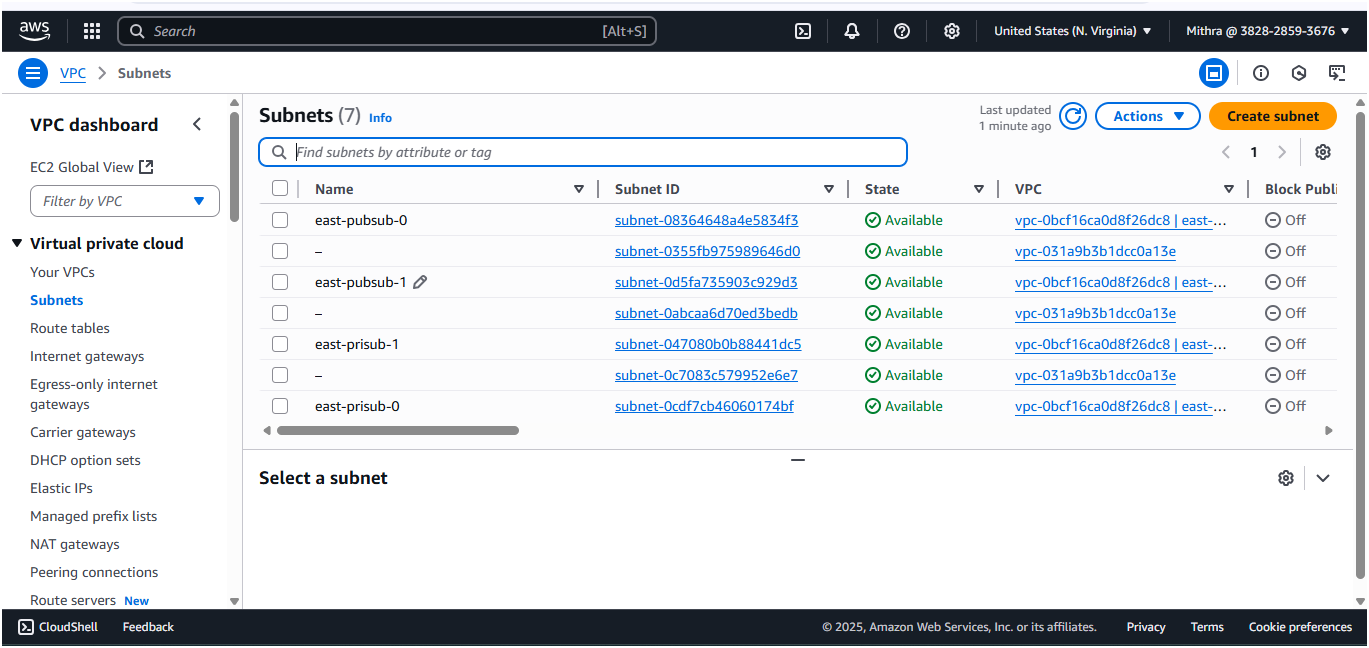


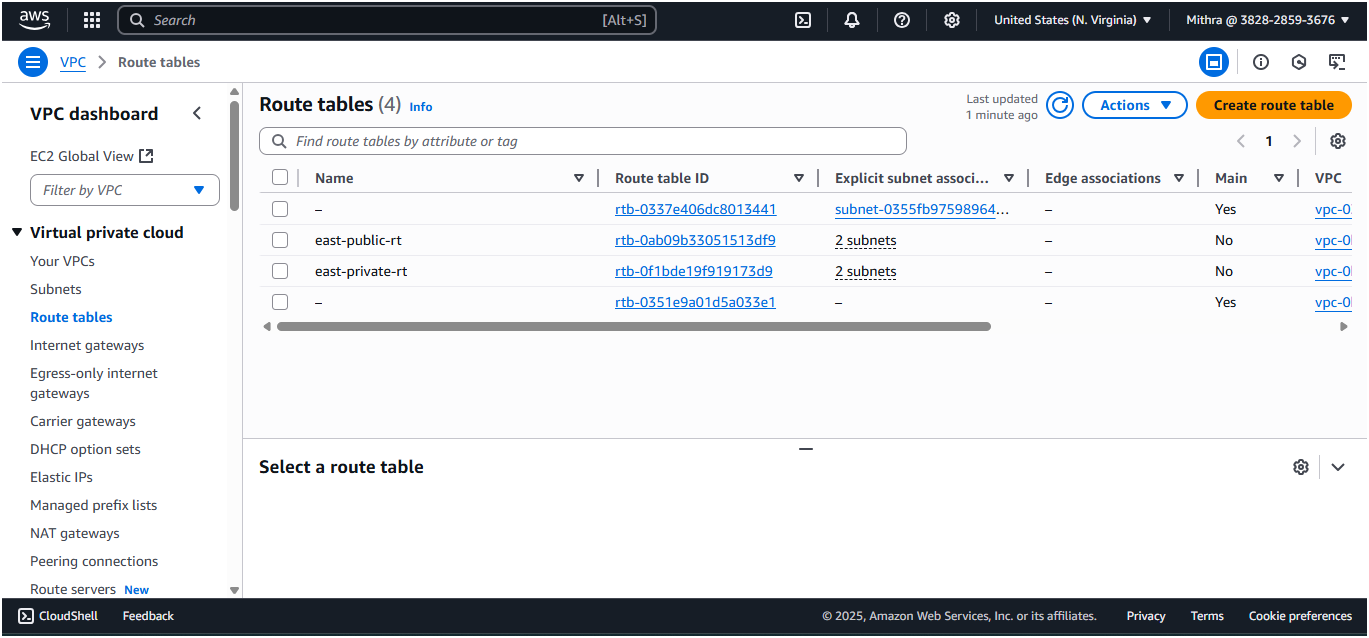
Provisioned the resources like EC2, VPC, Subnets, route tables, internet gateway, NAT gateway, S3, Launch templates and load balancer,RDS

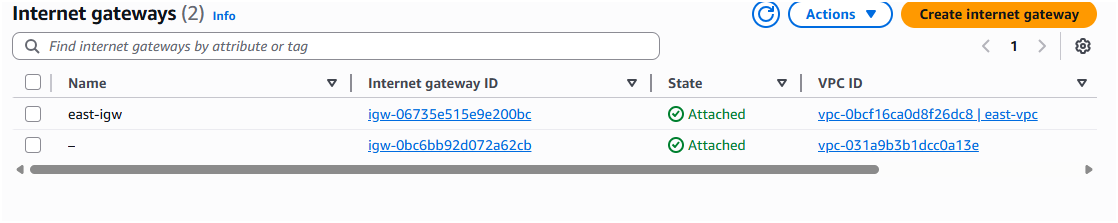


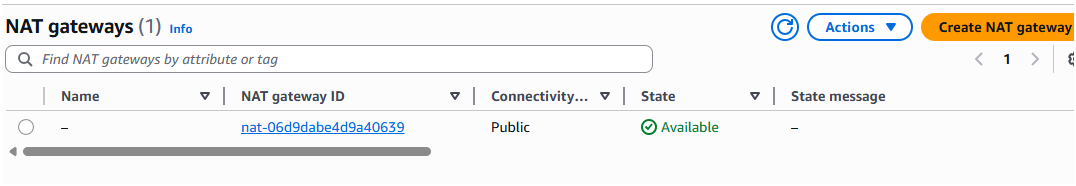




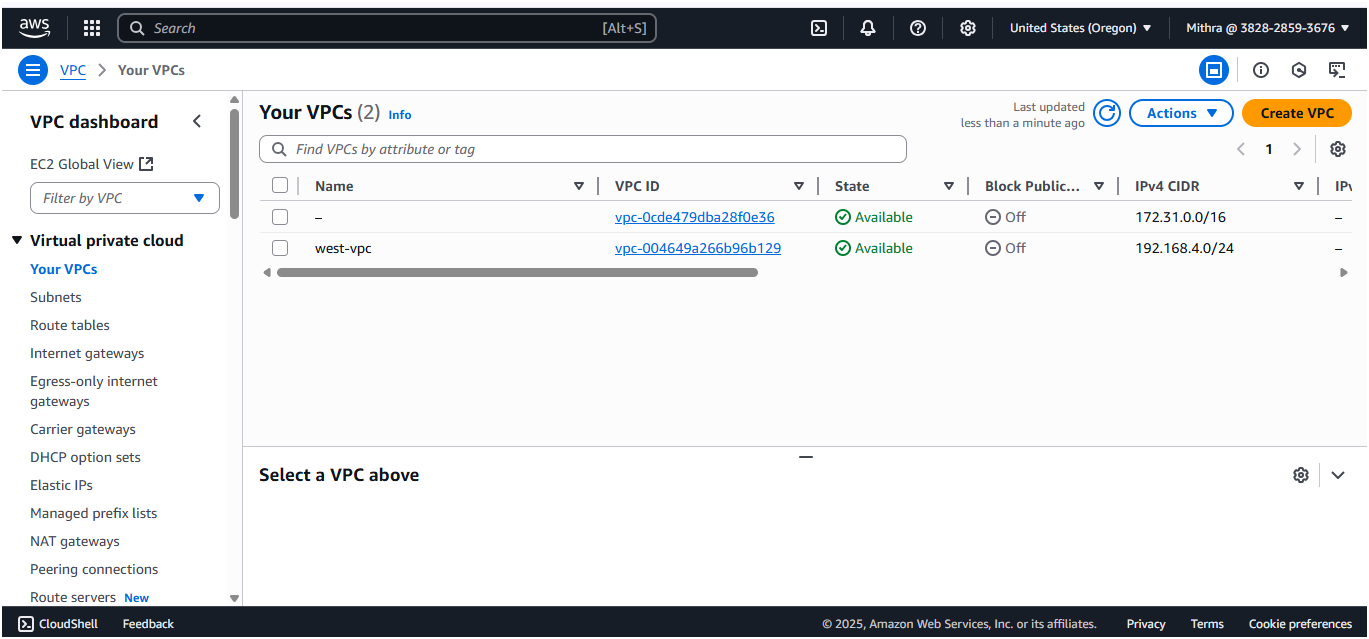


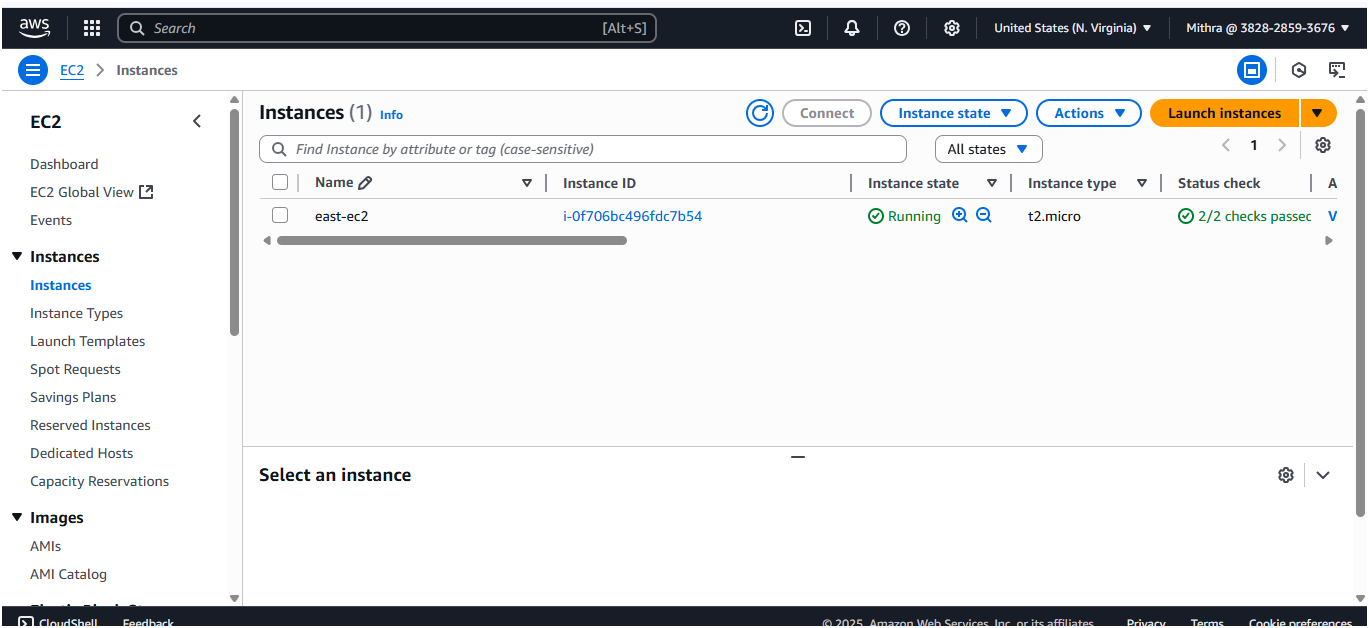


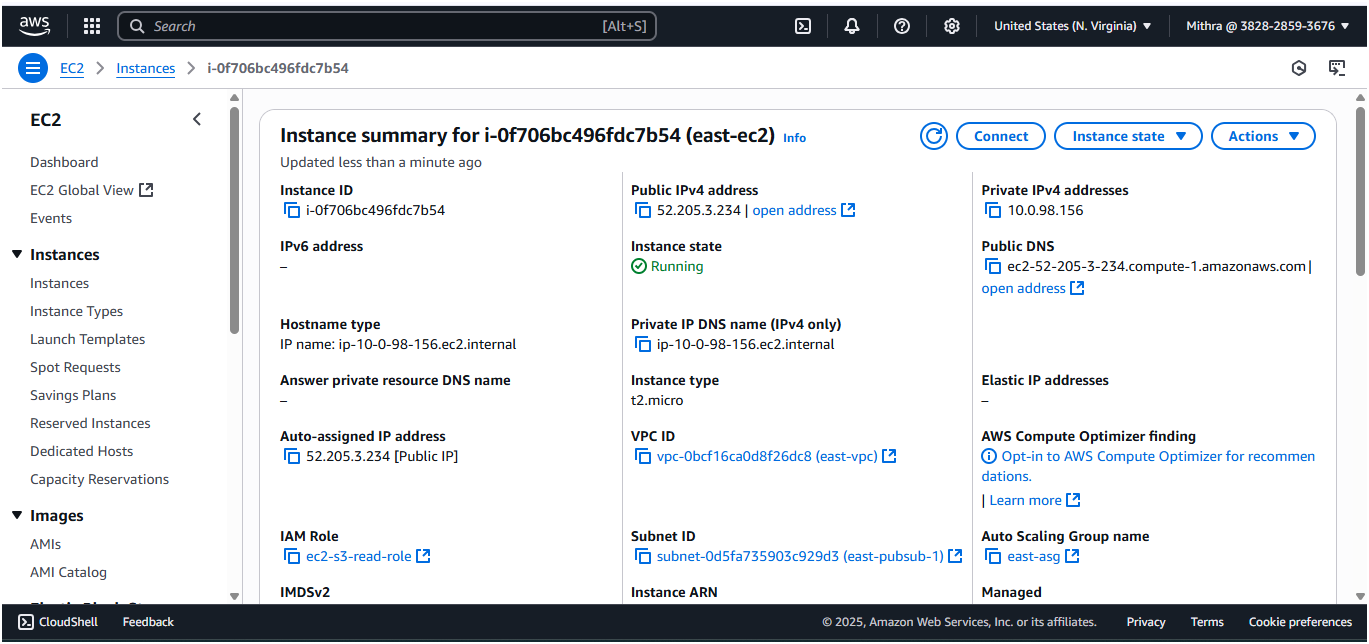




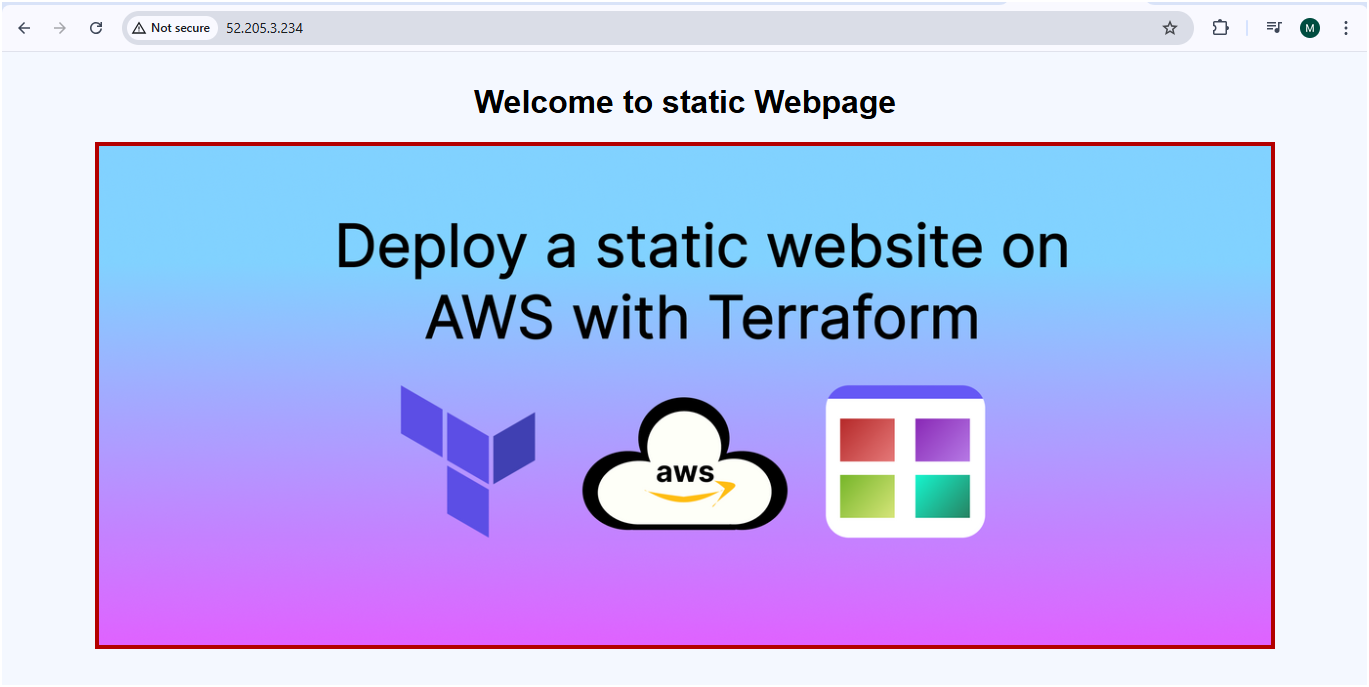
West region – us-west-2

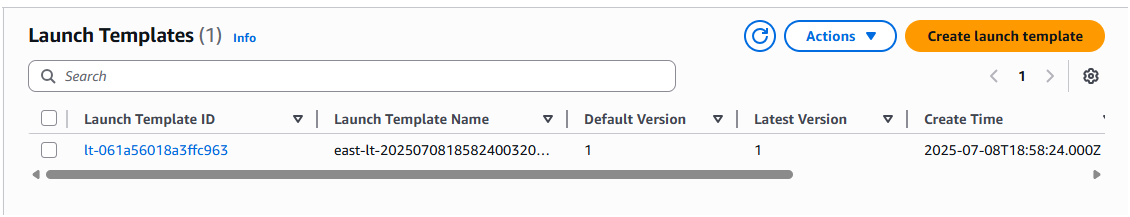






Displaying static webpage from EC2-public ip

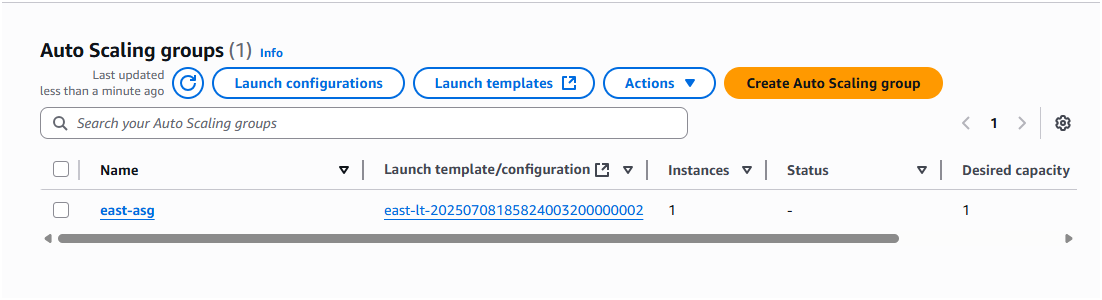




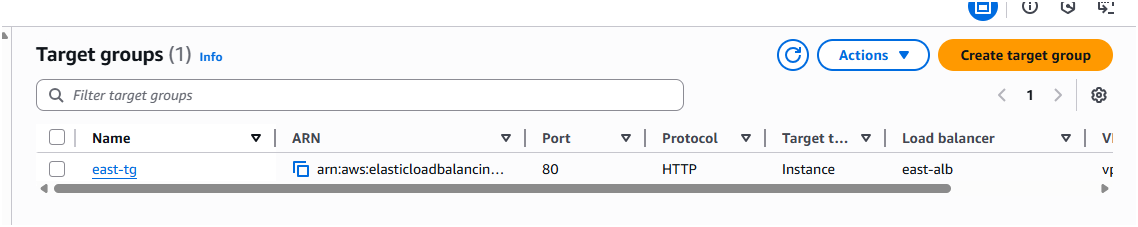
Security group EC2 and ALB



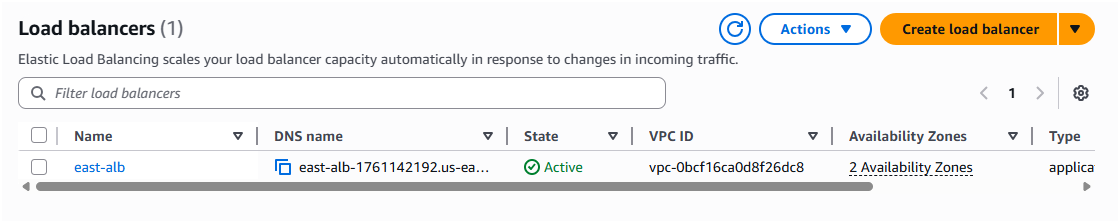
ASG – min,max and desired capacity = 1



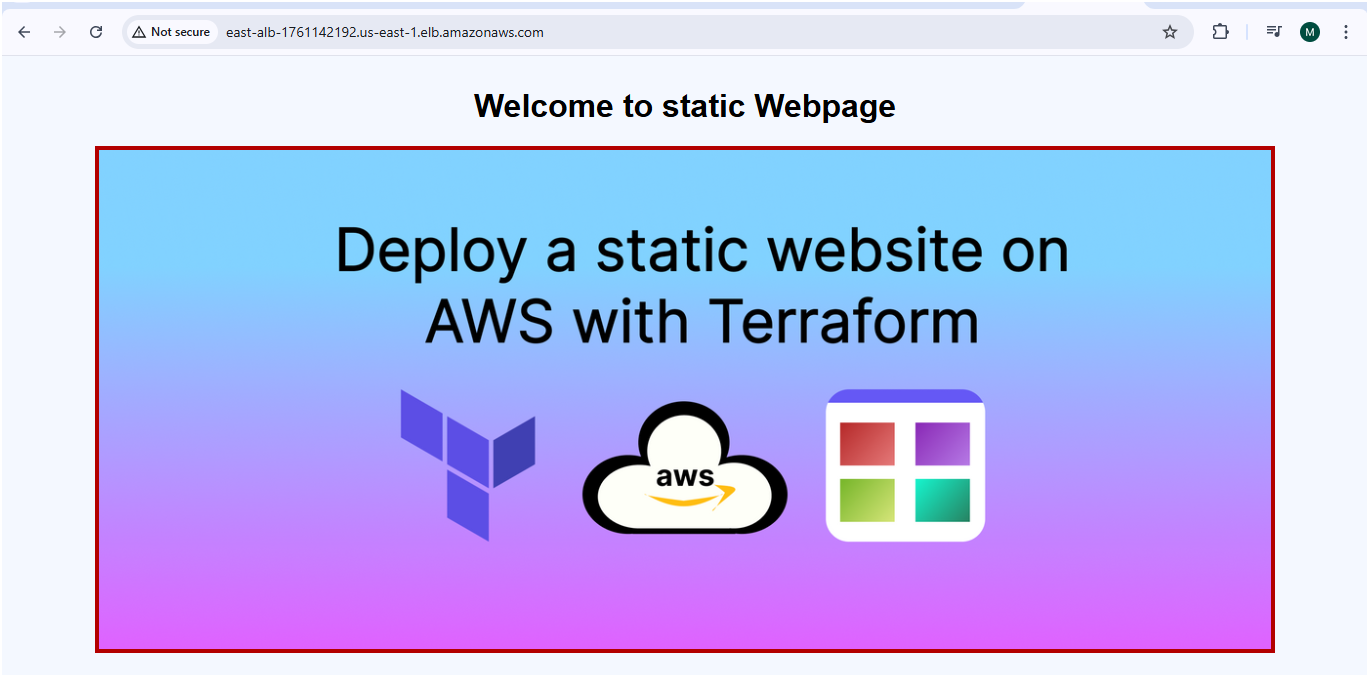
Target group



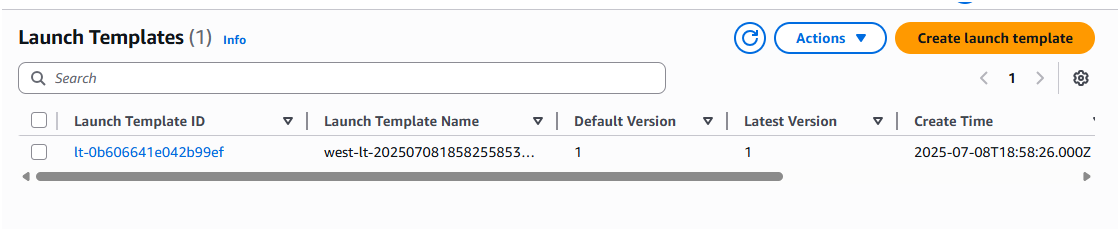
Load Balancer



Static webpage from ALB – east region

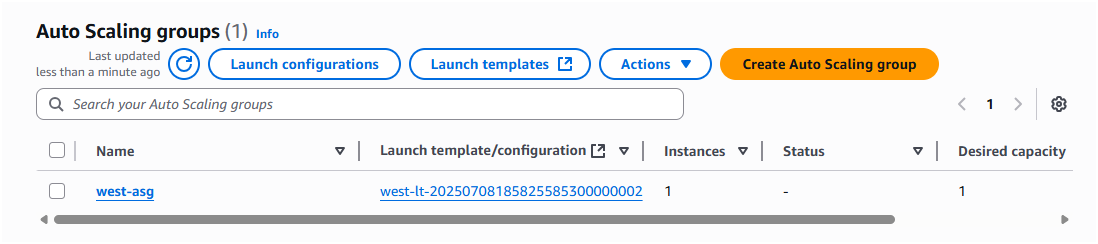


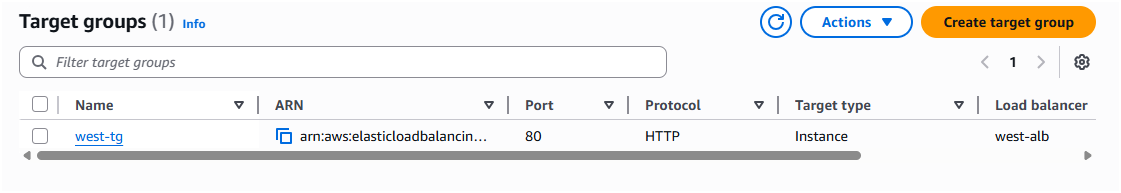
Now check the west region ALB



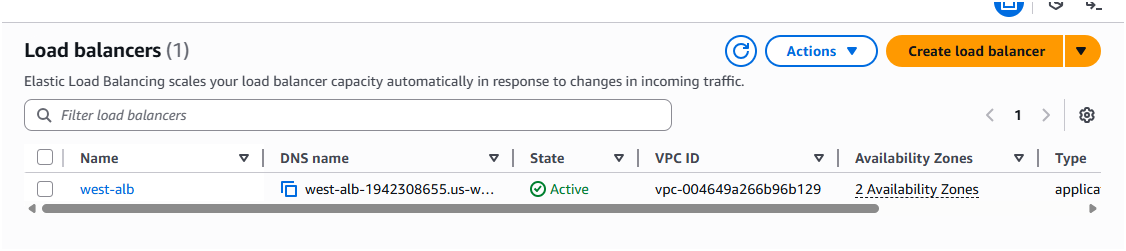
SG for EC2 and ALB

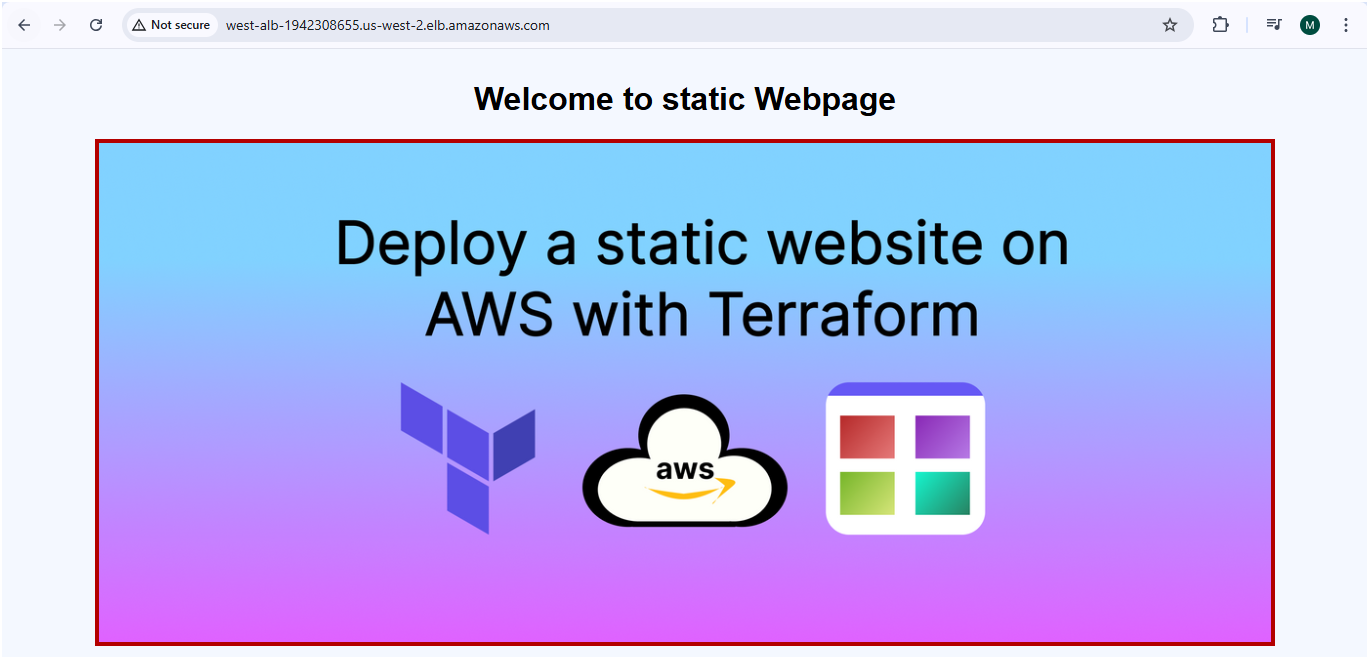






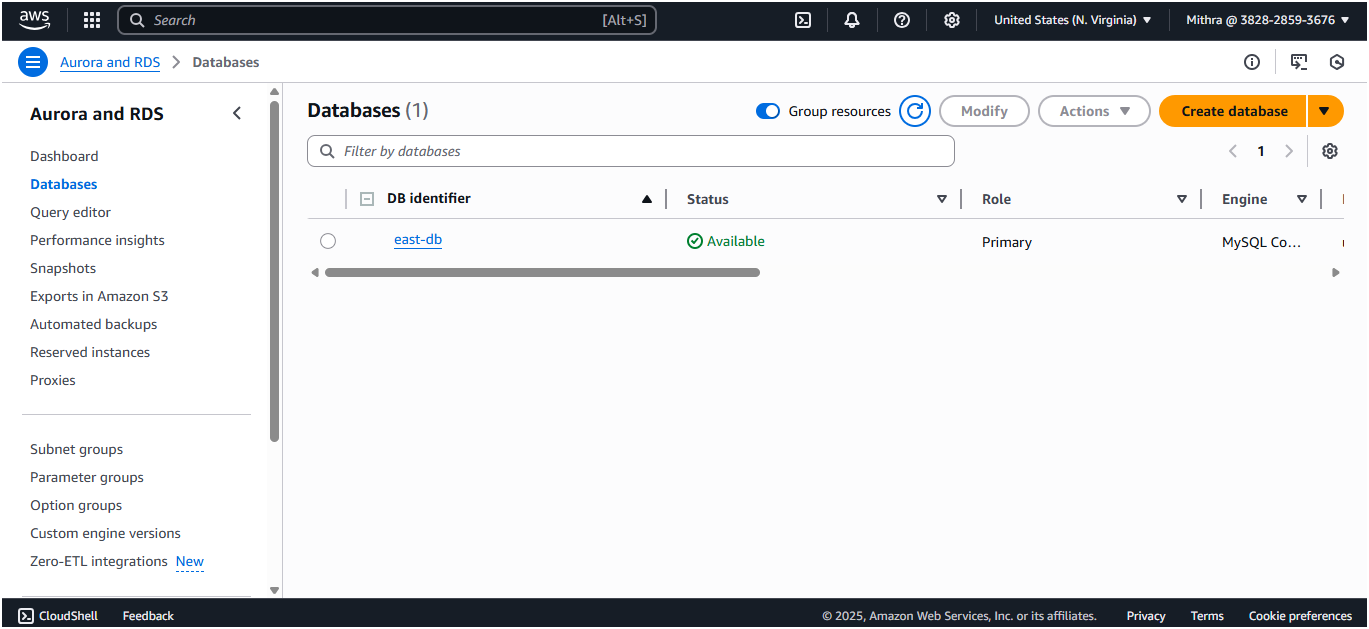
Load Balancer

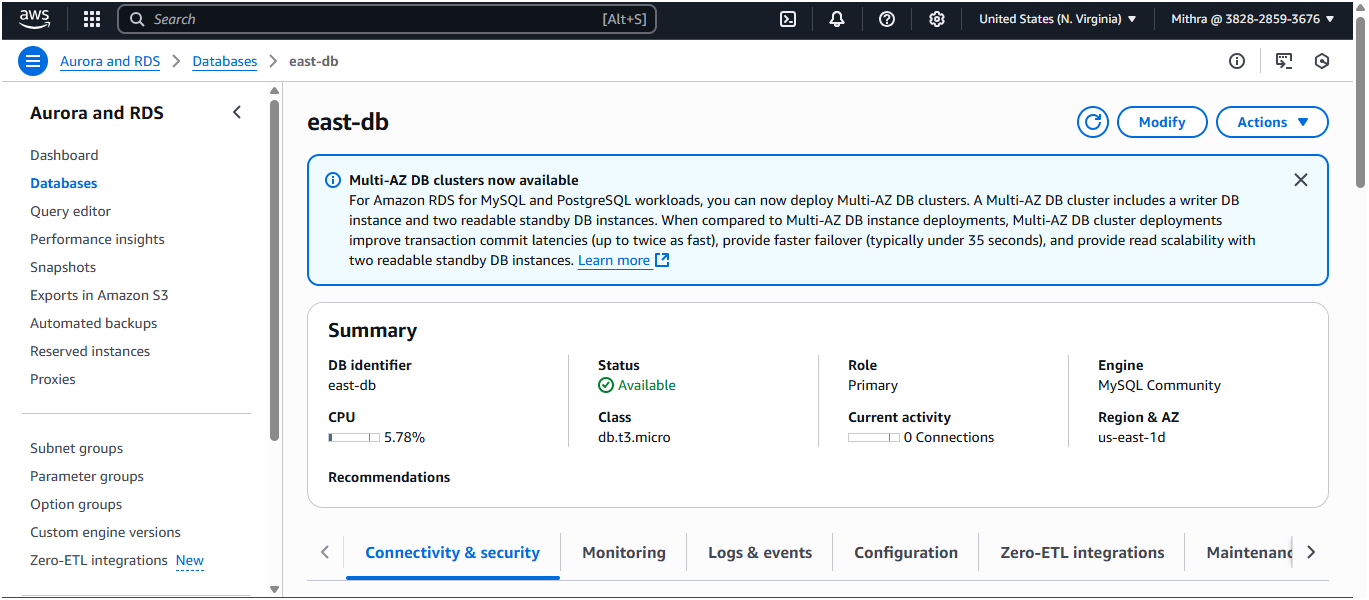


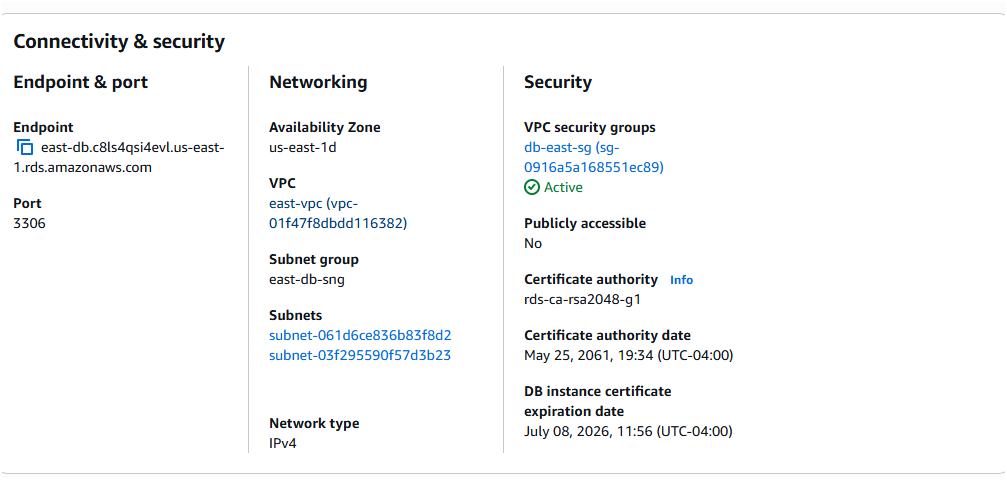


Now check the RDS

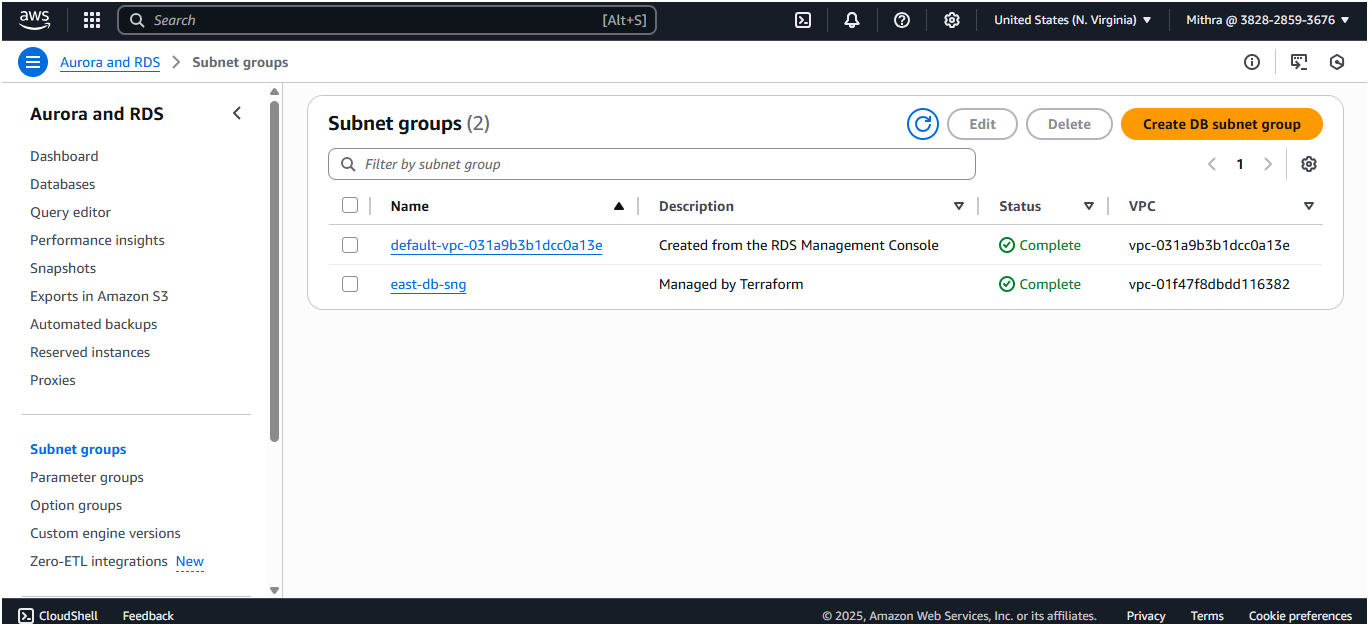
Primary RDS

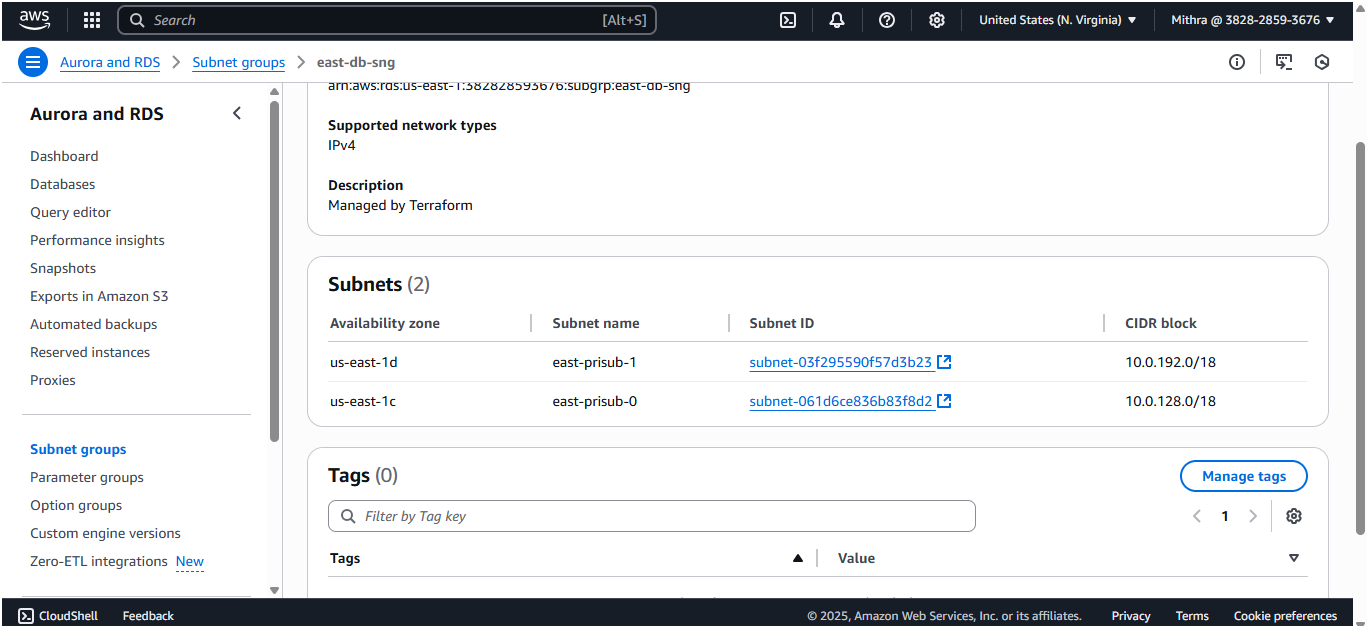






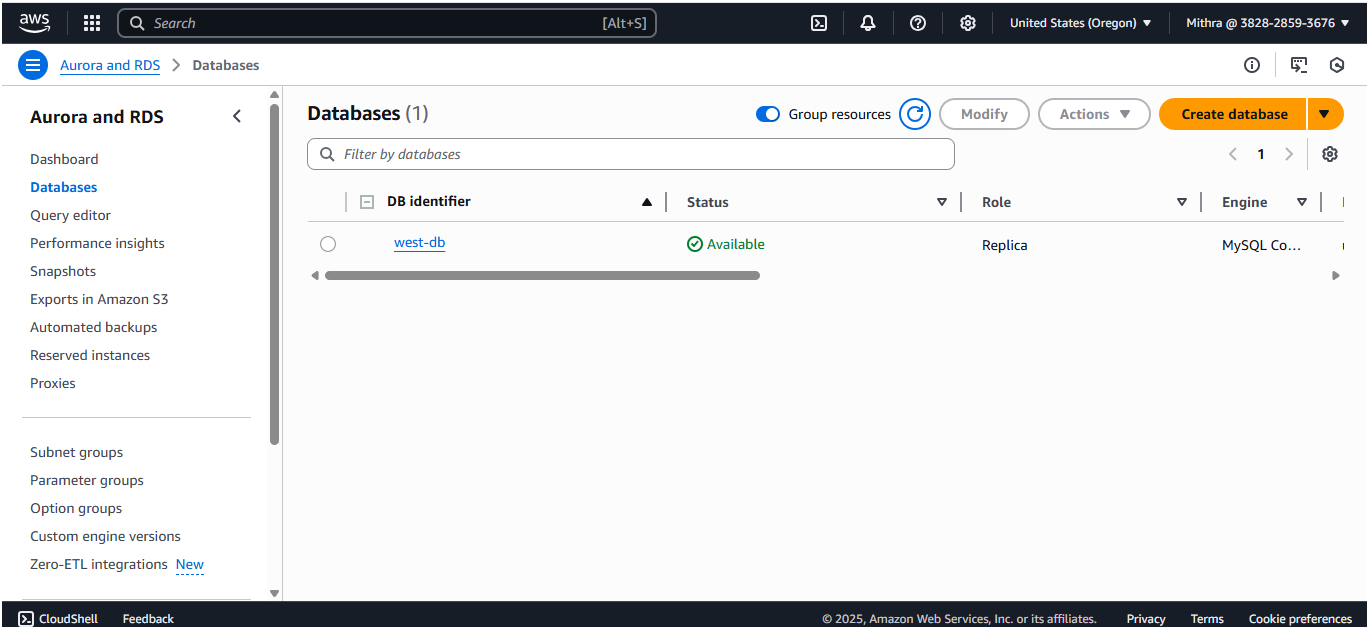
Subnet group

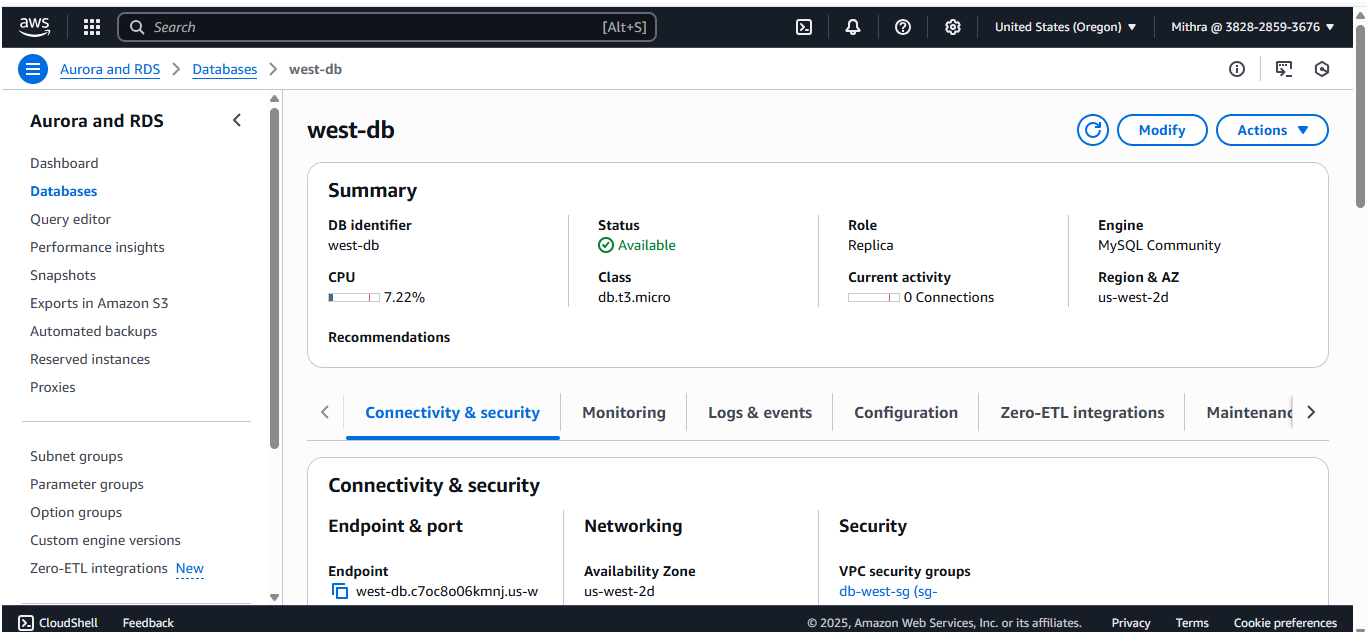


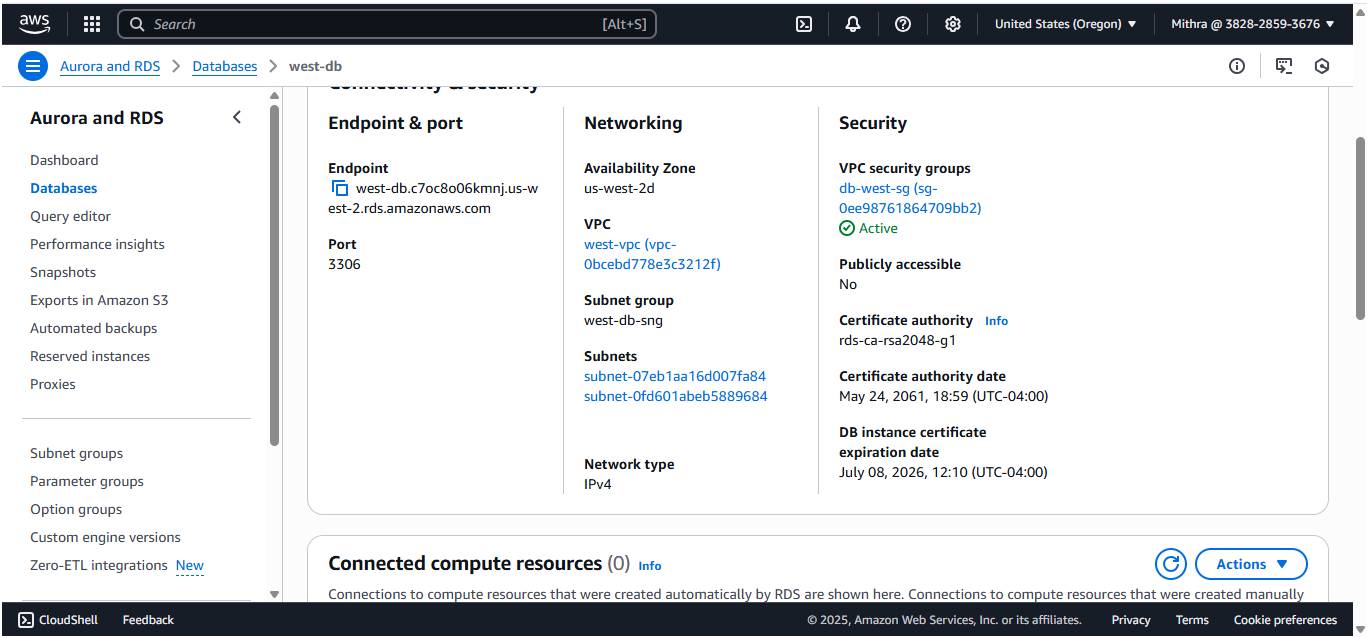


West region – RDS Replica



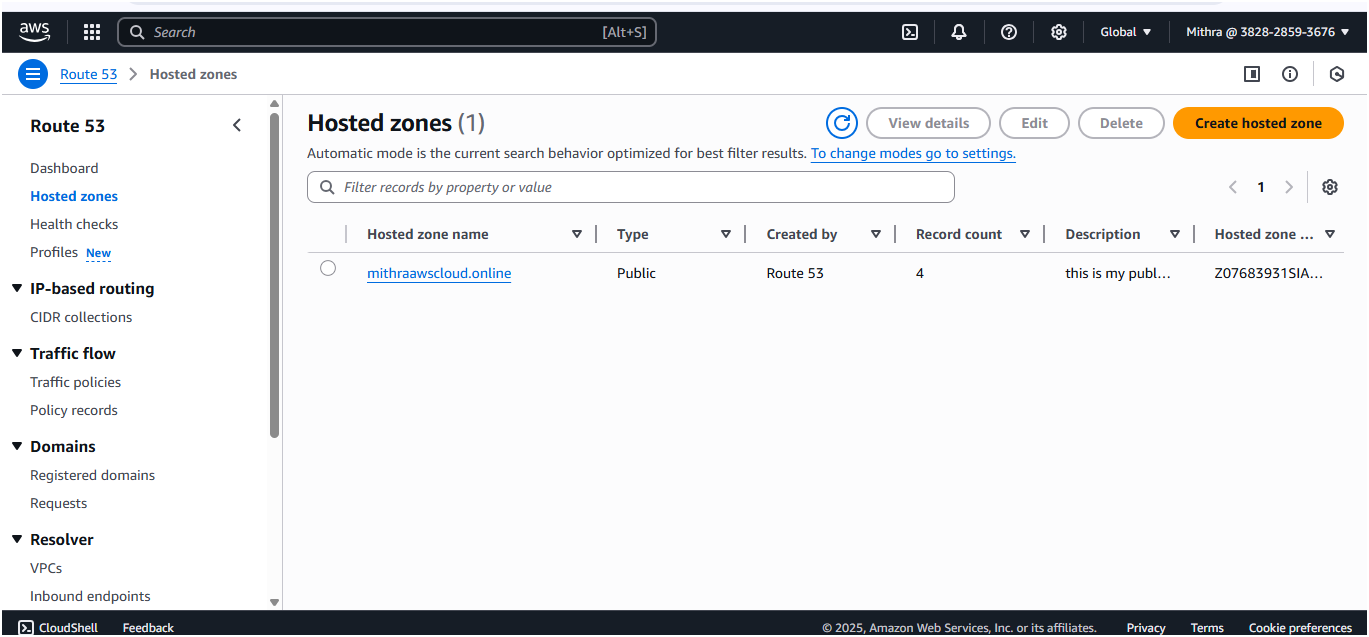






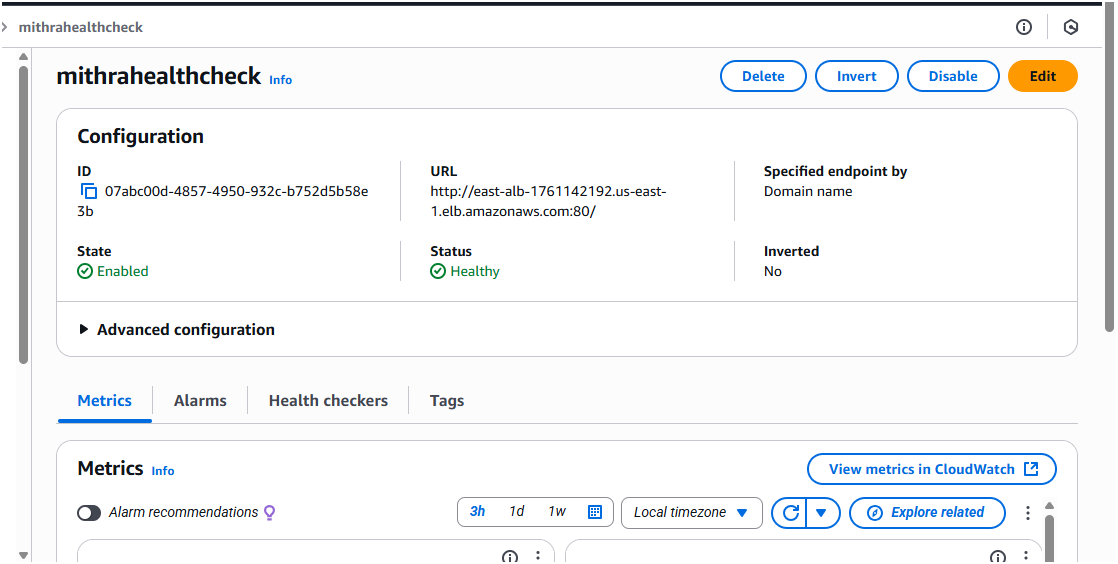
Create Route53

**Domain name - www.mithraawscloud.online**

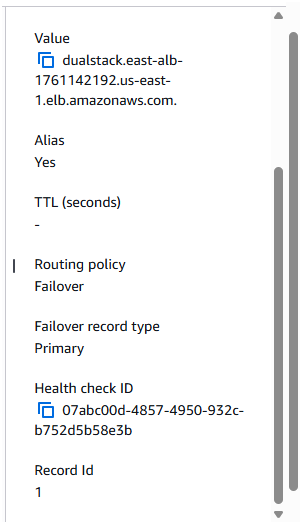




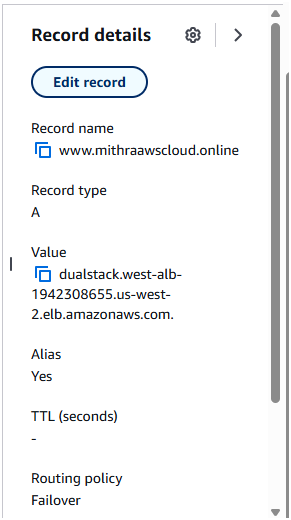
Create health check for primary



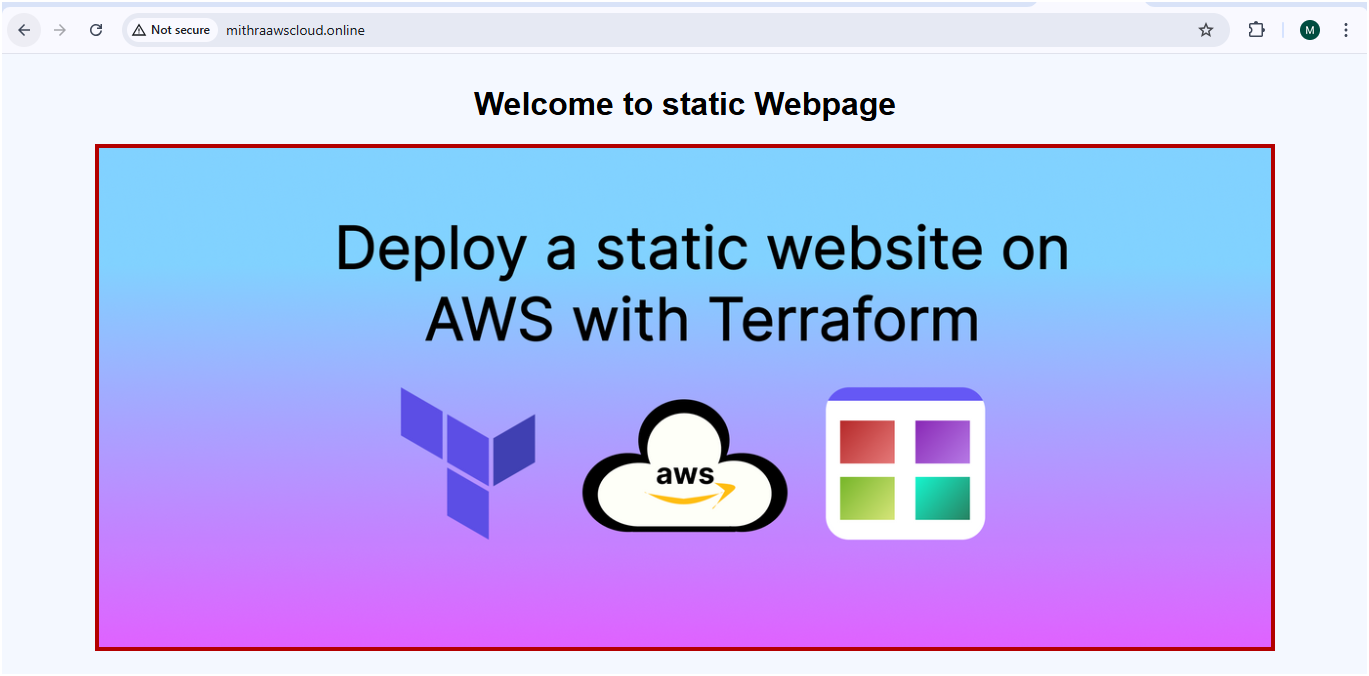
Primary routing policy linked with east region load balancer



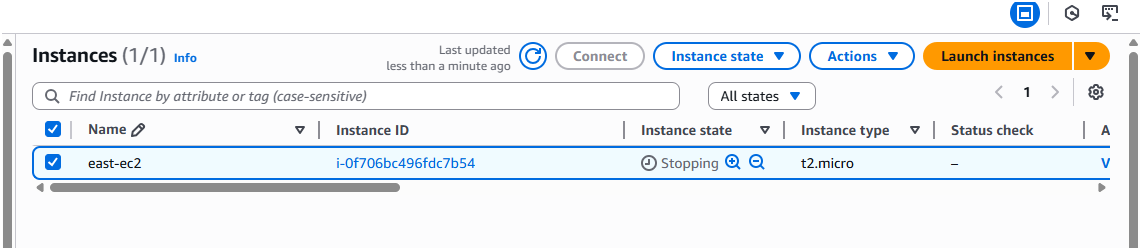
Secondary routing policy linked with west region load balancer



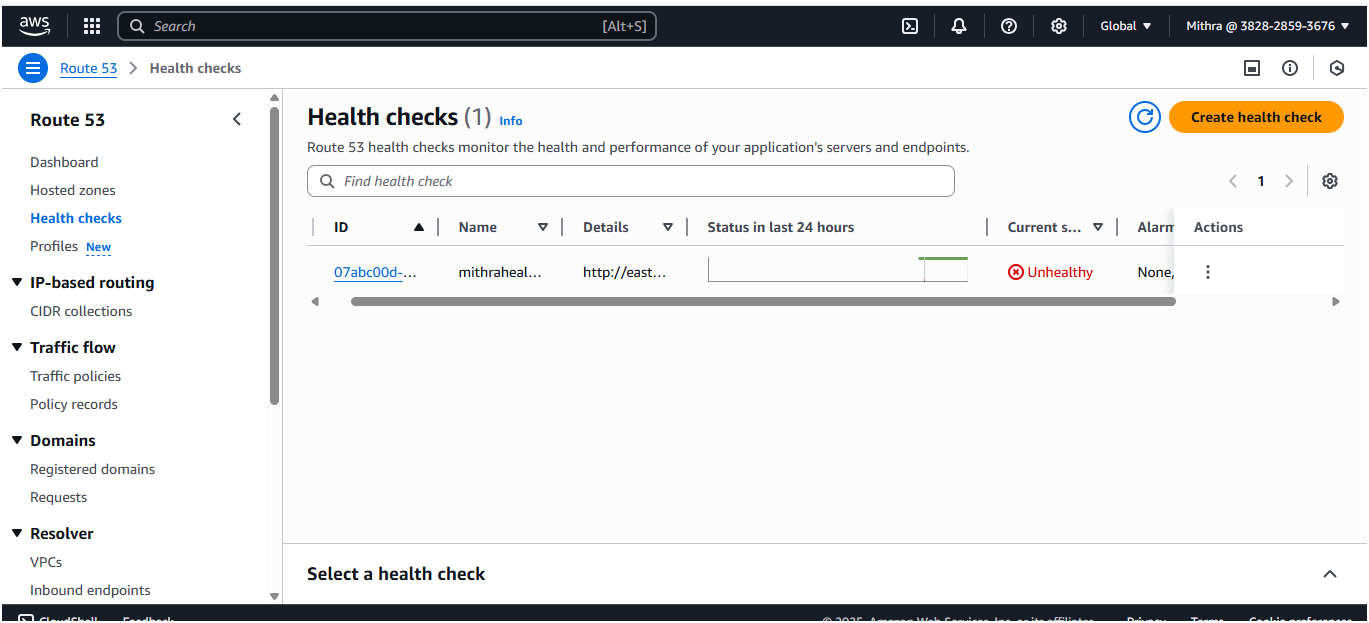
Now check the domain name is working fine



Now manually test that by stopping instance in east region ec2 then the west region should able to work



Now the health check should be unhealthy and it should automatically redirect to secondary



# Final Verifications:

Test it with:  <www.mithraawscloud.online>

It Displays static webpage with secondary region after manually stopped the primary region instance

