AWS DEVSECOPS-139

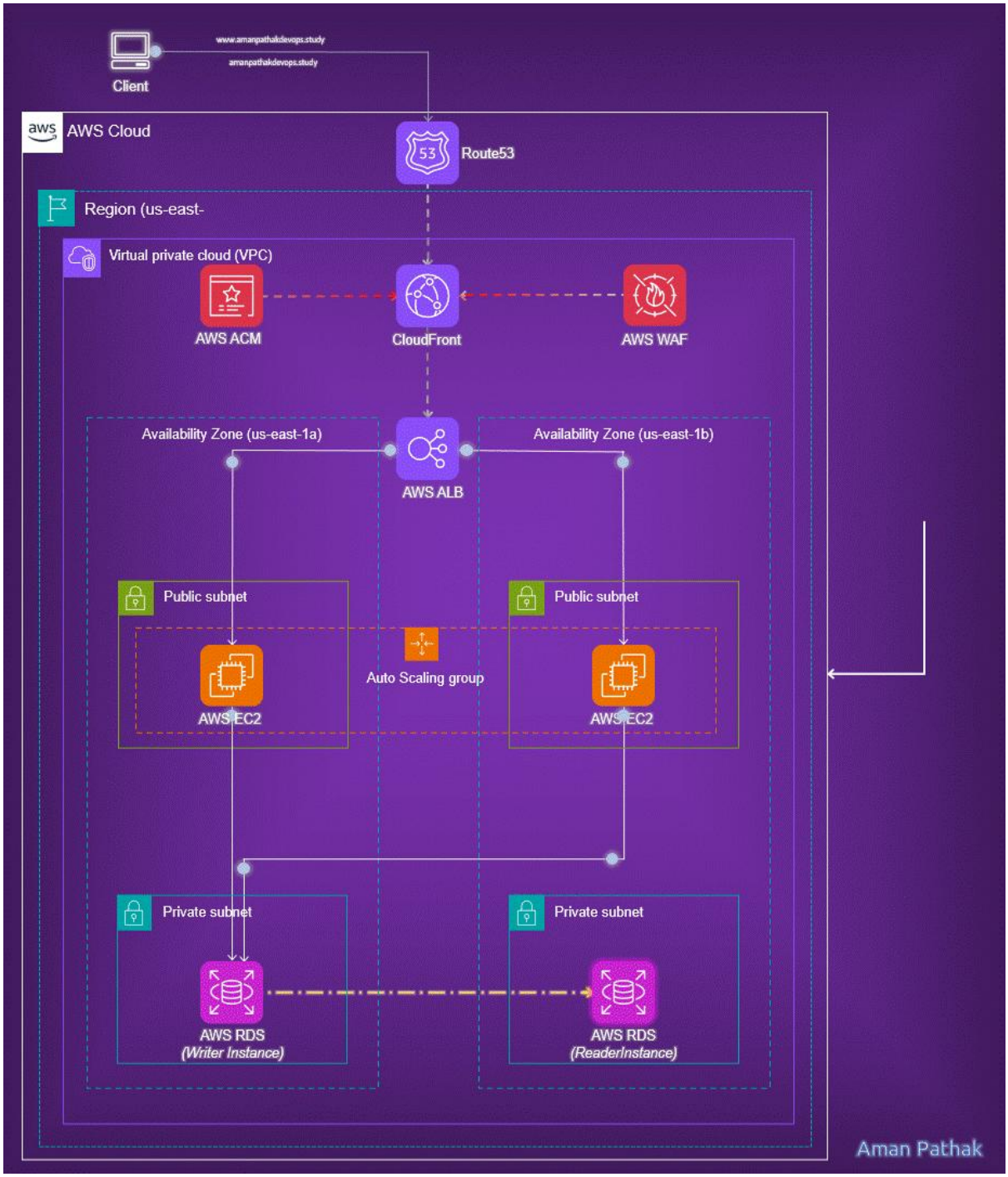
PROJECT- 02

AWS Secure, Scalable, High Available Web Hosting Architecture Using EC2, Route53, WAF, ACM, CloudFront, & RDS

BANDI ANIL KUMAR

Mail Id: bandianilkumar49@gmail.com

AWS Secure, Scalable, High Available Web Hosting Architecture Using EC2, Route53, WAF, ACM, CloudFront, & RDS



Overview

Designed and implemented a secure, scalable, and highly available cloud infrastructure for a web application using AWS services. The architecture supports global content delivery, automated scaling, and fault-tolerant database operations across multiple Availability Zones.

It utilized Amazon EC2 instances behind an Elastic Load Balancer with Auto Scaling to handle dynamic traffic loads. DNS routing was managed through Amazon Route 53, ensuring reliable global access. Security was enforced using Security Groups, and AWS WAF to protect against common web threats. The architecture also incorporated monitoring and logging via Amazon CloudWatch, enabling proactive performance management. This project demonstrates hands on expertise in building scalable, secure, and production-ready web hosting environments using core AWS services.

🌟 Key Components

- **Networking**: Configured a custom VPC with public and private subnets across two Availability Zones for high availability and isolation.
- **Security & Access**: Integrated AWS WAF for threat mitigation, ACM for SSL/TLS encryption, and IAM roles for secure service access.
- **Traffic Management**: Deployed Route 53 for DNS routing and CloudFront for global content delivery with low latency.
- **Compute Layer**: Provisioned EC2 instances within an Auto Scaling Group behind an Application Load Balancer (ALB) to ensure dynamic scaling and load distribution.
- **Database Layer**: Implemented Amazon RDS with Multi-AZ deployment, featuring a writer instance and a read replica for performance and redundancy.

🌟 Key Benefits

✅ **Scalability**: Auto Scaling and CloudFront handle traffic spikes.

✅ **High Availability**: Multi-AZ deployment ensures uptime.

✅ **Security**: WAF, ACM, private subnets, and IAM roles protect resources.

✅ **Performance**: Load balancing and read replicas optimize response times.

Outcomes

- Achieved seamless scalability and high fault tolerance across all layers.
- Ensured secure access and data protection through layered security controls.
- Optimized performance and cost-efficiency using auto scaling and read replicas.

Services:
AWS VPC • EC2 • RDS • ALB • CloudFront • Route 53 • WAF • ACM • Auto Scaling • IAM • Multi-AZ Deployment • Infrastructure Design • Cloud Security

🔹 Client Access & DNS

- **Route 53** handles domain name resolution and directs traffic to the application.
- **CloudFront** serves content globally with low latency.
- **AWS Certificate Manager (ACM)** provides SSL/TLS certificates for secure HTTPS access.
- **AWS WAF** protects against common web exploits and attacks.

🔹 Networking & Infrastructure

- A **Virtual Private Cloud (VPC)** spans two Availability Zones for redundancy.
- **Public subnets** host EC2 instances behind an **Application Load Balancer (ALB)**.
- **Private subnets** contain RDS instances for secure database access.
- **NAT Gateway** allows outbound internet access from private subnets.

🔹 Compute & Scaling

- **EC2 instances** run the application code.
- An **Auto Scaling Group** dynamically adjusts EC2 capacity based on demand.
- **ALB** distributes incoming traffic across healthy EC2 instances.

🔹 Database Layer

- **Amazon RDS** is deployed in Multi-AZ mode:
  - **Writer instance** handles write operations.
  - **Reader instance** improves read performance and availability.

Architecture Overview

This setup includes:

- **Global Access & DNS**: Route 53
- **Security & Delivery**: AWS Certificate Manager (ACM), CloudFront, AWS WAF
- **Core Infrastructure**: VPC with public/private subnets across two Availability Zones
- **Compute Layer**: EC2 instances in an Auto Scaling Group behind an Application Load Balancer (ALB)
- **Database Layer**: RDS with writer/reader instances for high availability and read scalability

Steps followed to design the Architecture:

1. Set Up Networking (VPC & Subnets)
2. Configure Security
3. Provision EC2 Instances
4. Set Up Application Load Balancer (ALB)
5. Enable Auto Scaling
6. Add AWS WAF & CloudFront
7. Configure Route 53
8. Deploy RDS Instances

1. Set Up Networking (VPC & Subnets)

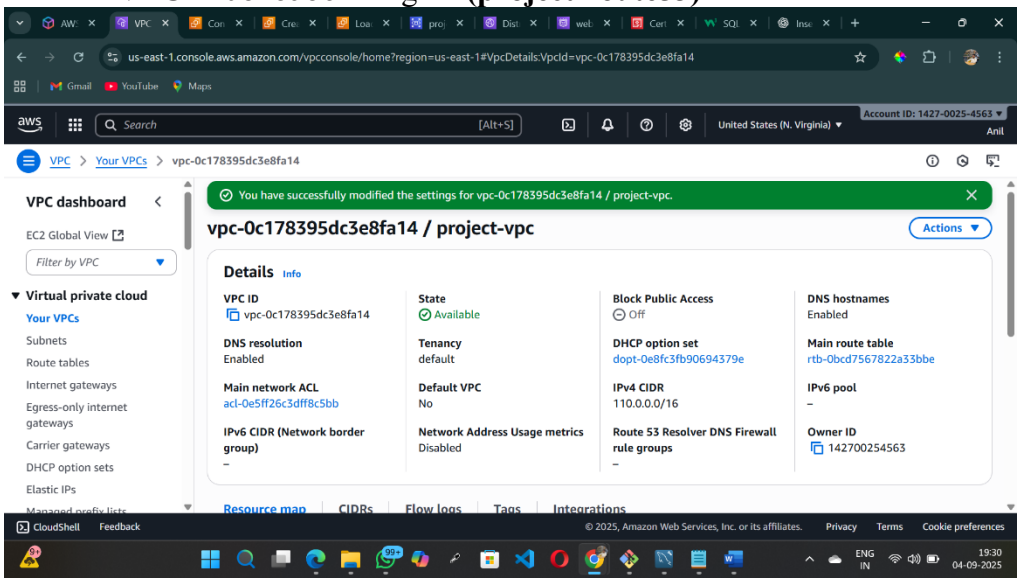
- Create a **VPC** in `us-east-1` region **(project-route53)**
- 

Fig 1.1: VPC

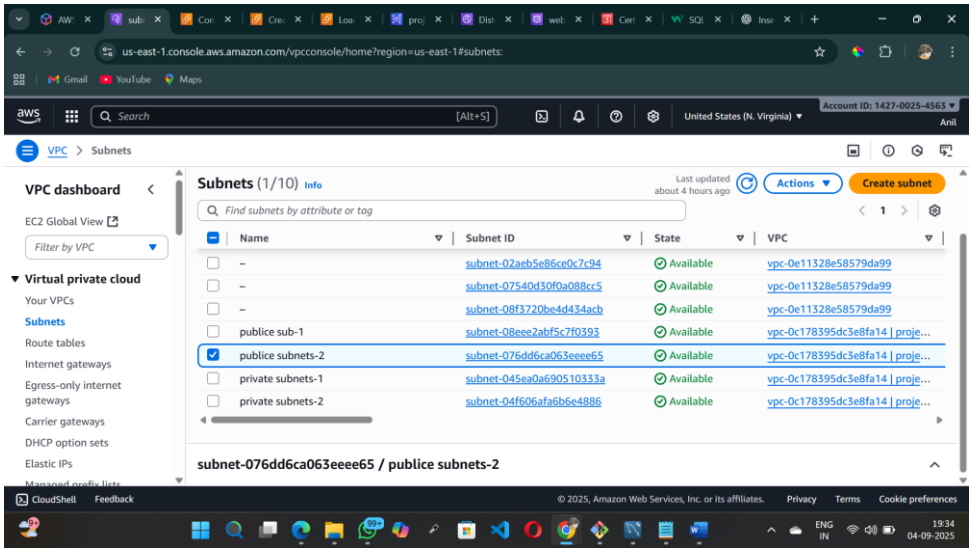
- A VPC spans two Availability Zones for redundancy **(e.g., `us-east-1a`, `us-east-1`**
- 

Fig 1.2: Availability Zones & Subnets

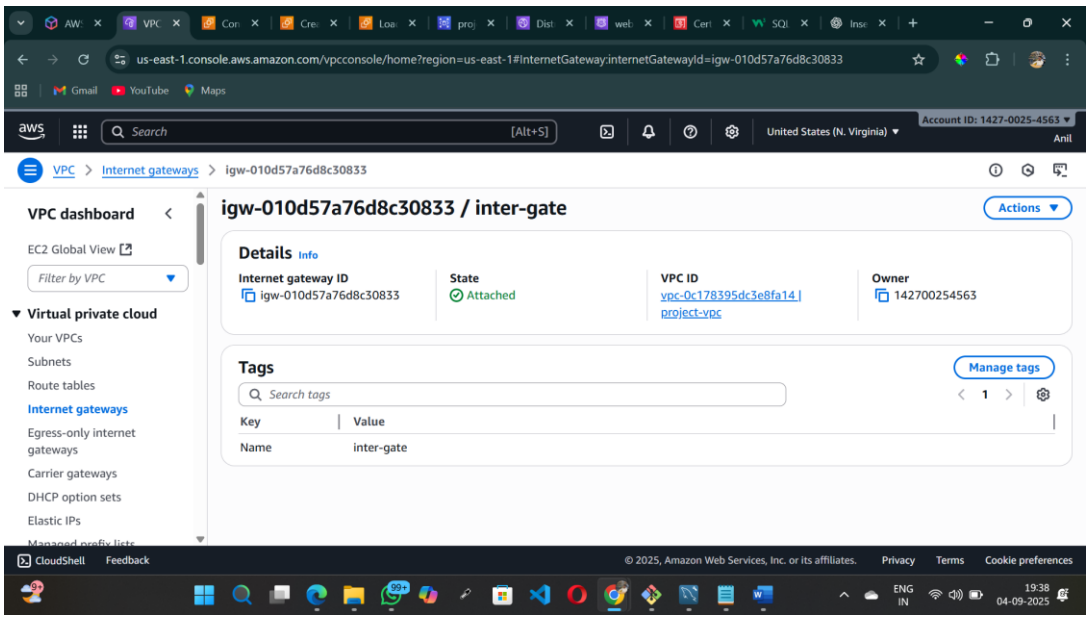
- Add **2 public subnets** & **2 private subnets** in the same AZs **(e.g., `us-east-1a`, `us-east-1b`)**
- Set up **INTERNET Gateway** in public subnet for private subnet internet access
- 

Fig 1.3: INTERNET gateway

.Configure **Internet Gateway** and route tables for public subnets

Fig 1.4: ROUTE TABLE

. There are 2 route table(1.public route, 2.private route table)

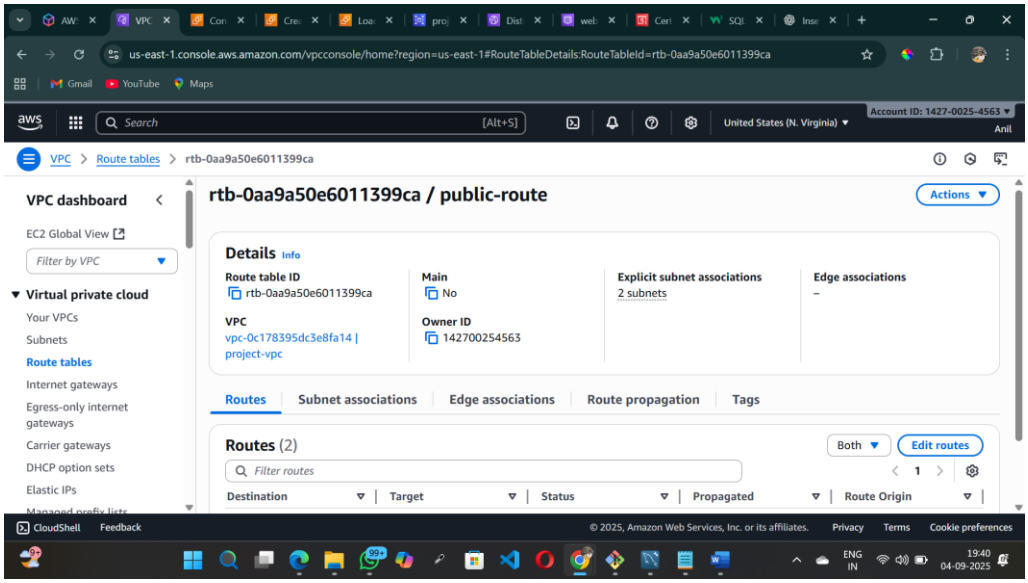
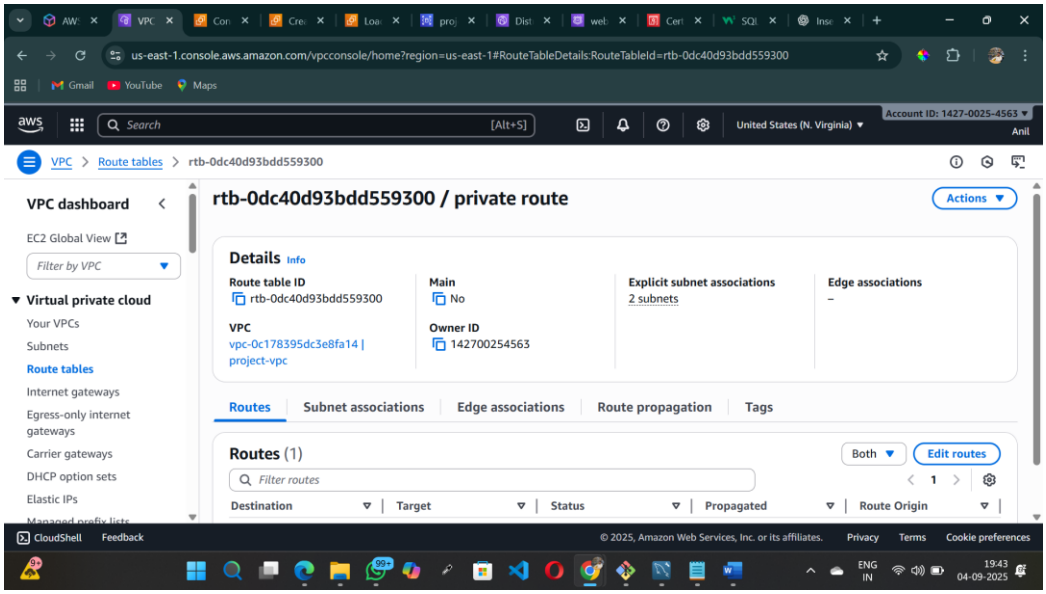
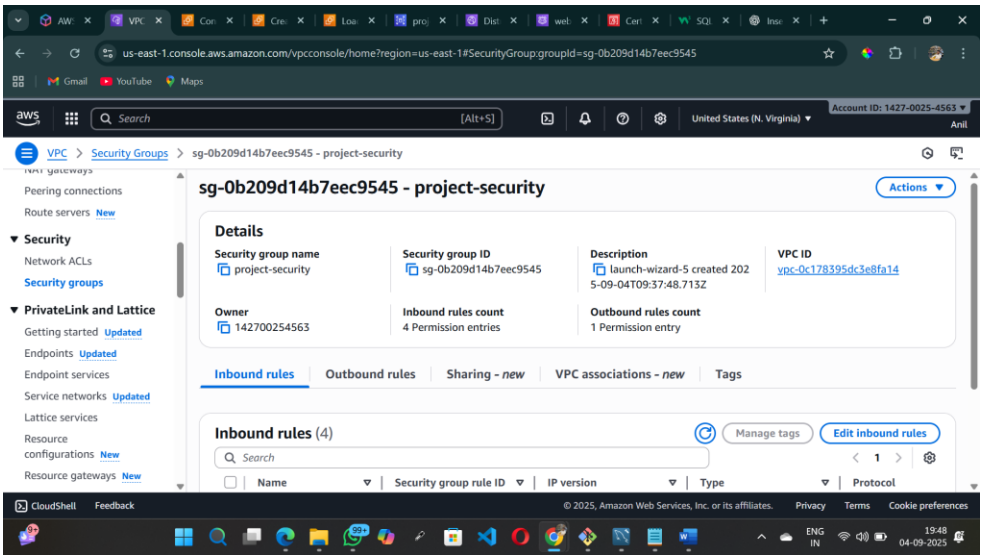


Fig 1.5: ROUTE TABLE



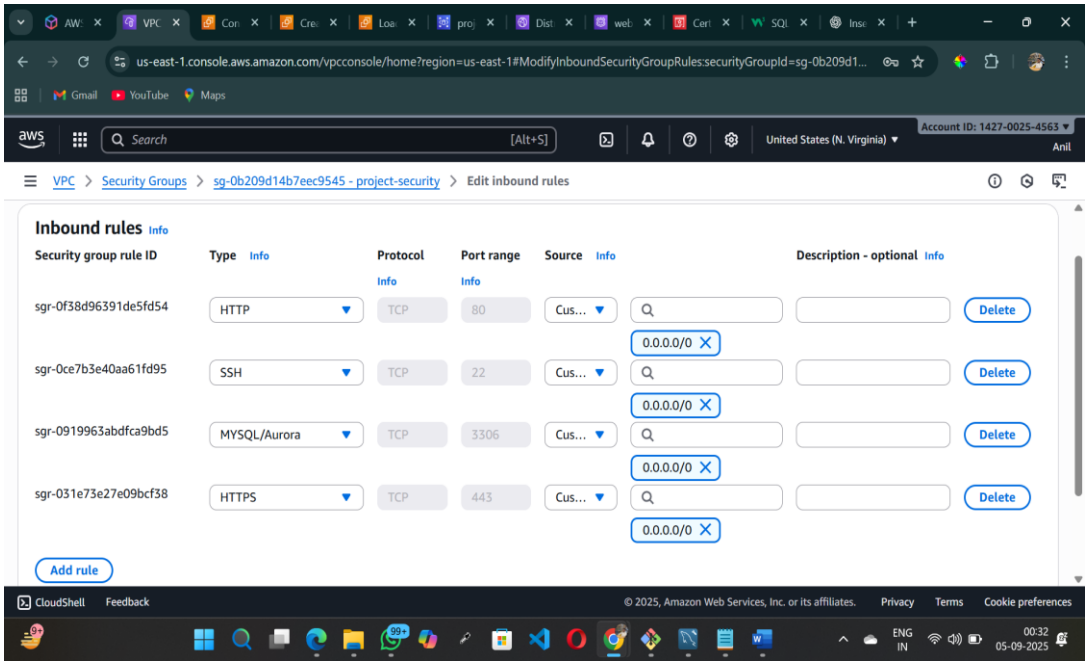
- Successfully configured VPC with all components and routes .
- Publice connect with internet gateway with subnet association (PUBLIC)
- Private route connect with NAT GATEWAY with subnets association(PRIVATE)

2. Configure Security

- Create **Security Groups** for:
  - 

    EC2 (allow HTTP, HTTPS &SSH traffic from anywhere-**default**).

Fig 2.1: Security Group for EC2



RDS (allow MySQL/Aurora &SSH traffic from anywhere-**rds-sg**)

Fig 3.1: EC2 INSTANCES

3. Provision EC2 Instances

- Launch two EC2 instances in public subnets
    - ec2-project web-1
    - ec2-project web-2

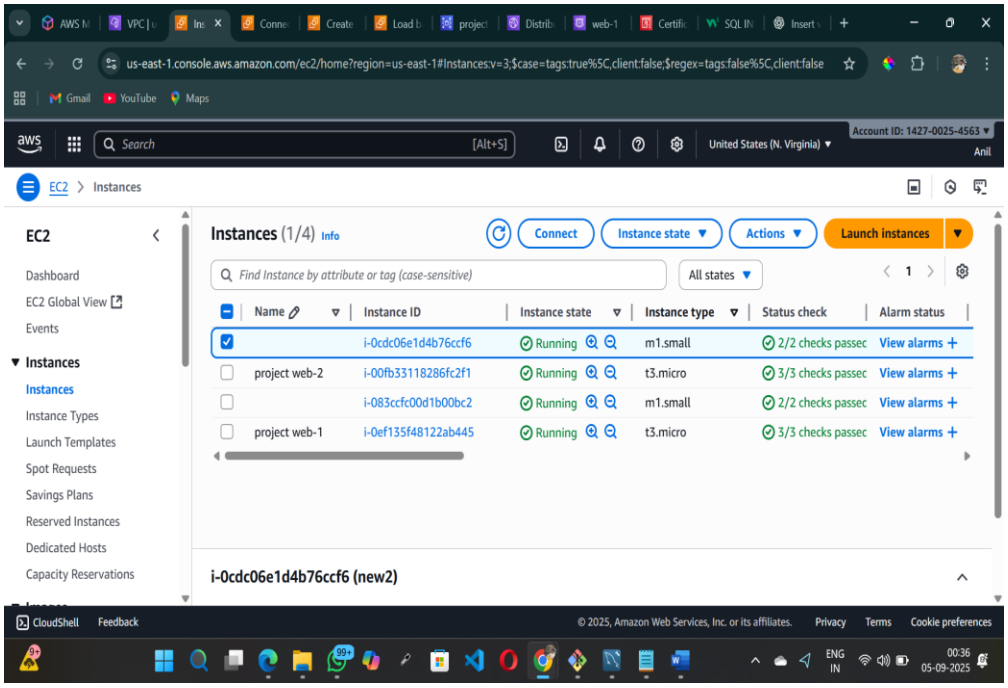


Fig 3.1: Ec2-1

- **Name**: Give your instance a meaningful name (project web-1 & project web-2)
- **AMI**: Choose an Amazon Machine Image (ubuntu)
- **Instance Type**: t3.medium for better performance
- **Key Pair**: Create or select an existing key pair for SSH access

.ec2-poject web-1

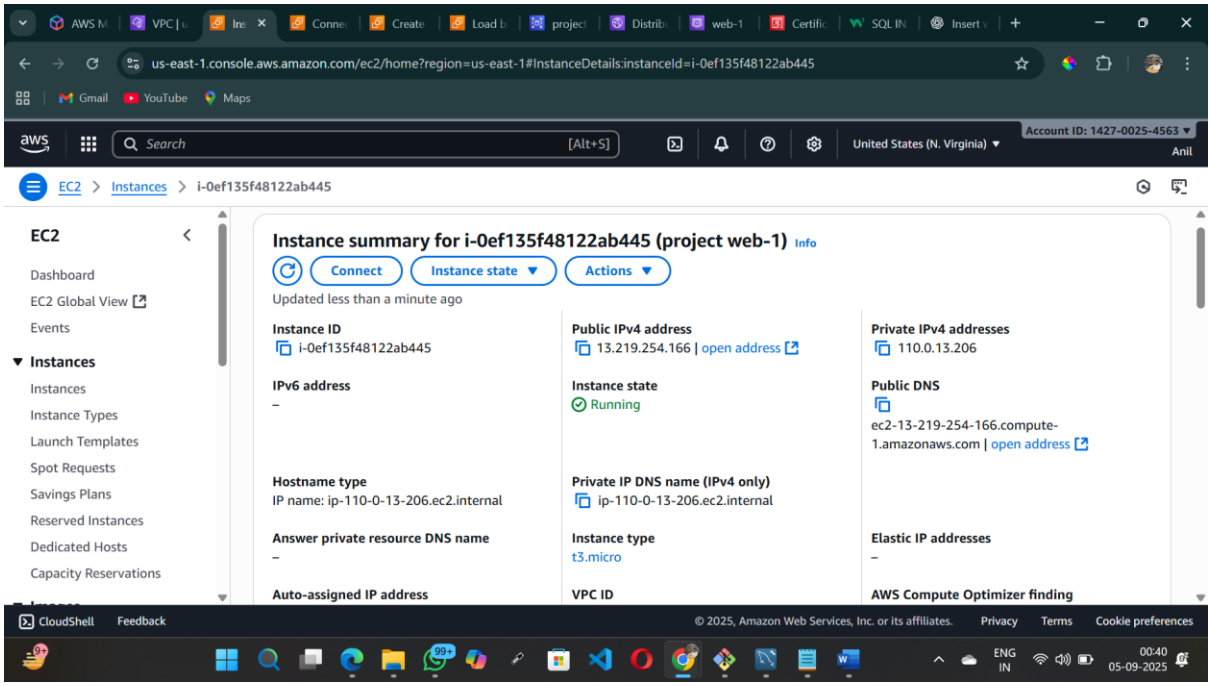
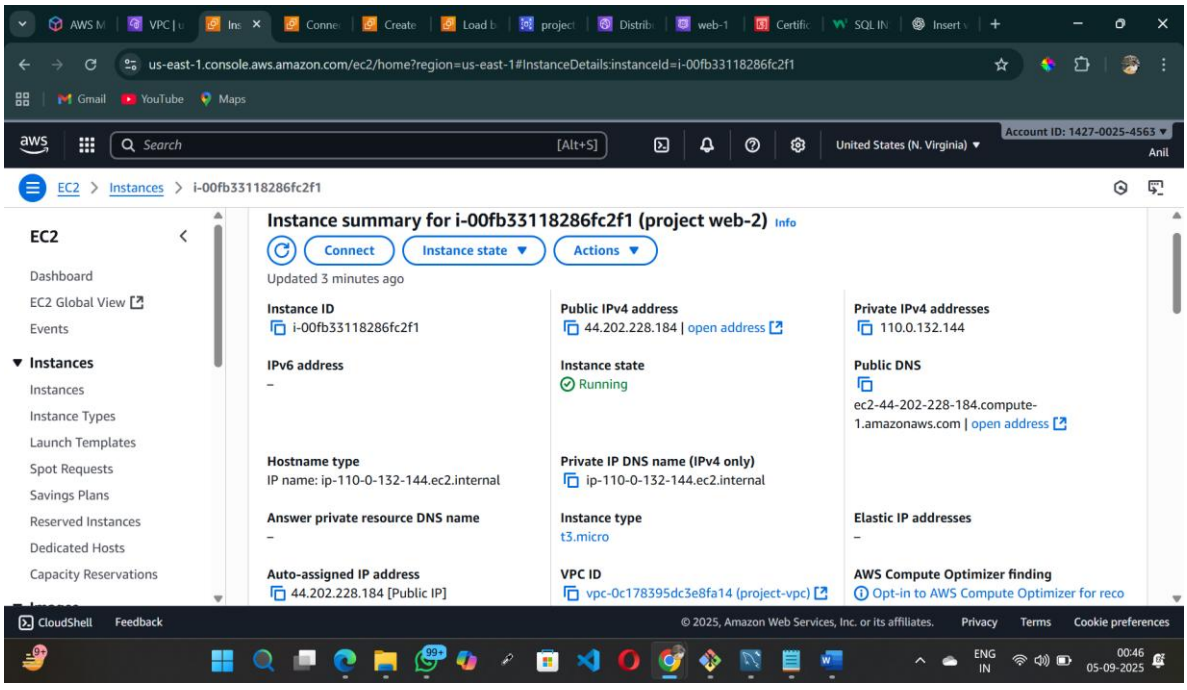


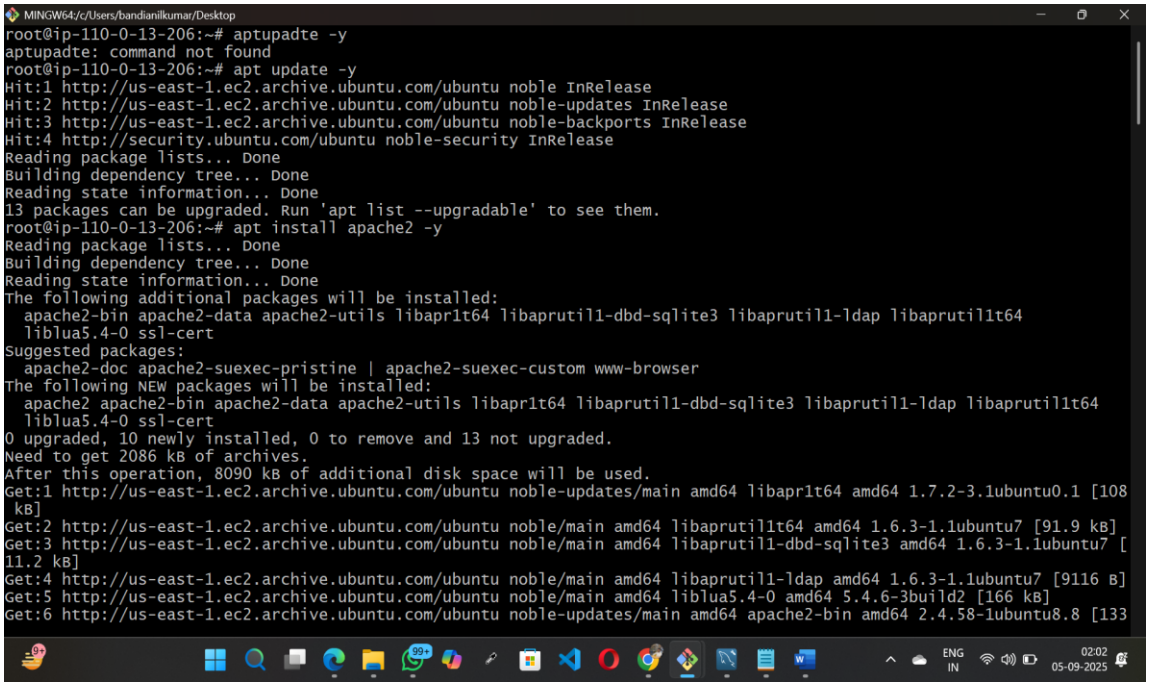
Fig 3.2: Ec2-1



Connect the both ec2 instances to gitbash with ssh link and Install your web application in them(apache2)

- Use following commands

- sudo -i
- yum update -y
- yum install apache2 -y
- cd /var/www/html/index.html (check your index.html file)
- systemctl status apache2 (if it shows inactive use below command to activate)
- Systemctl start/restart apache2



AMIS

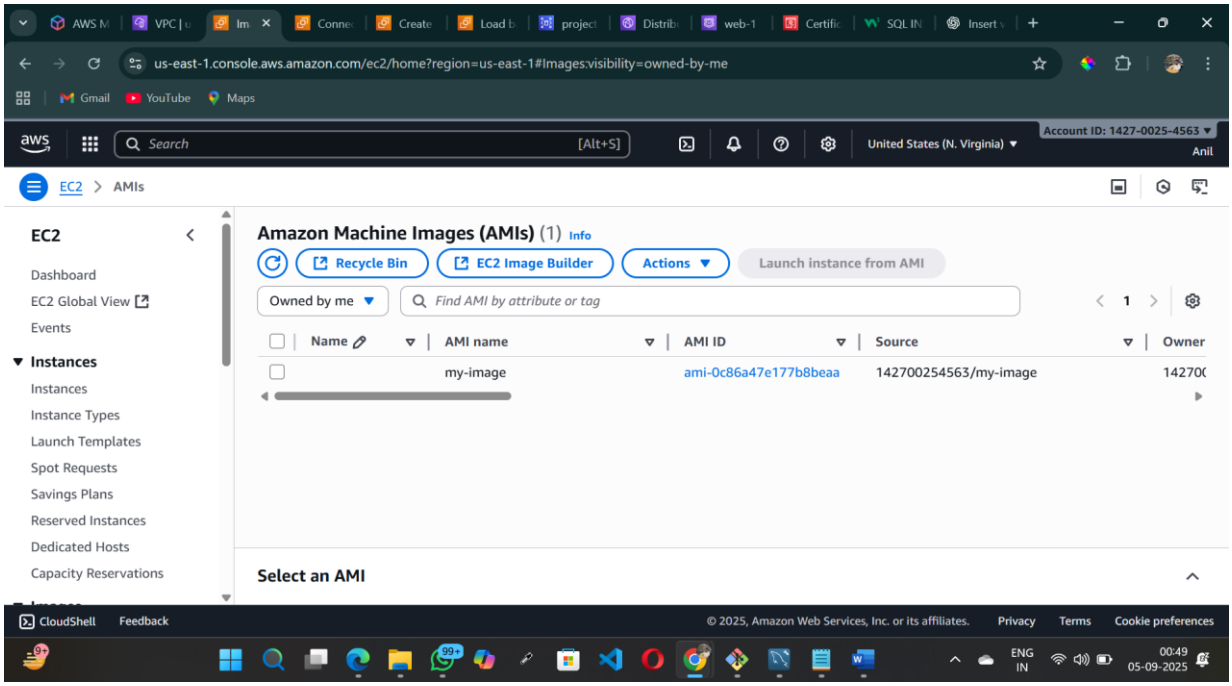
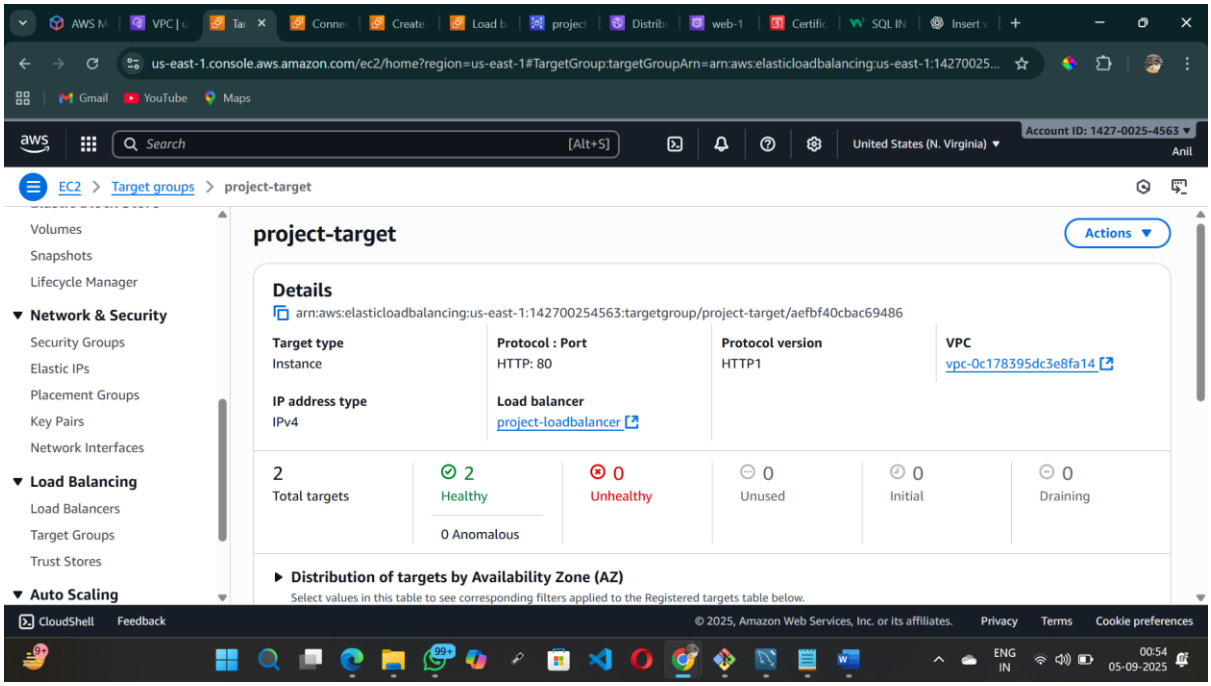
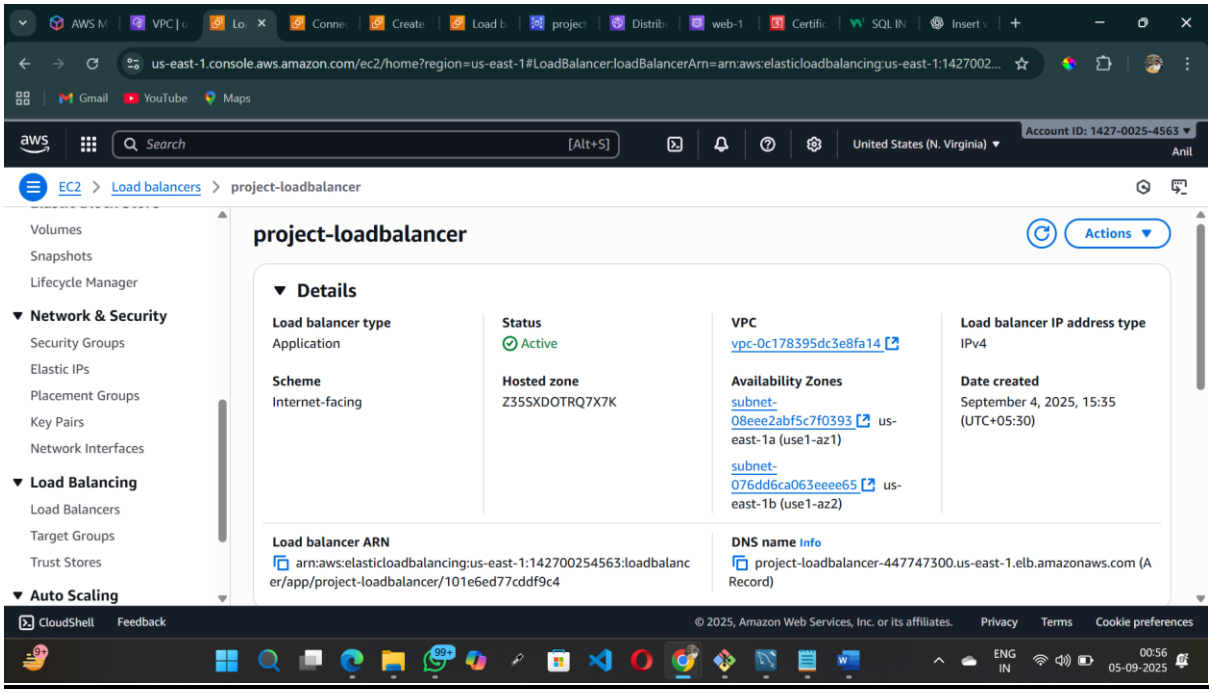


FIG 4.1: TAGRET GROUP

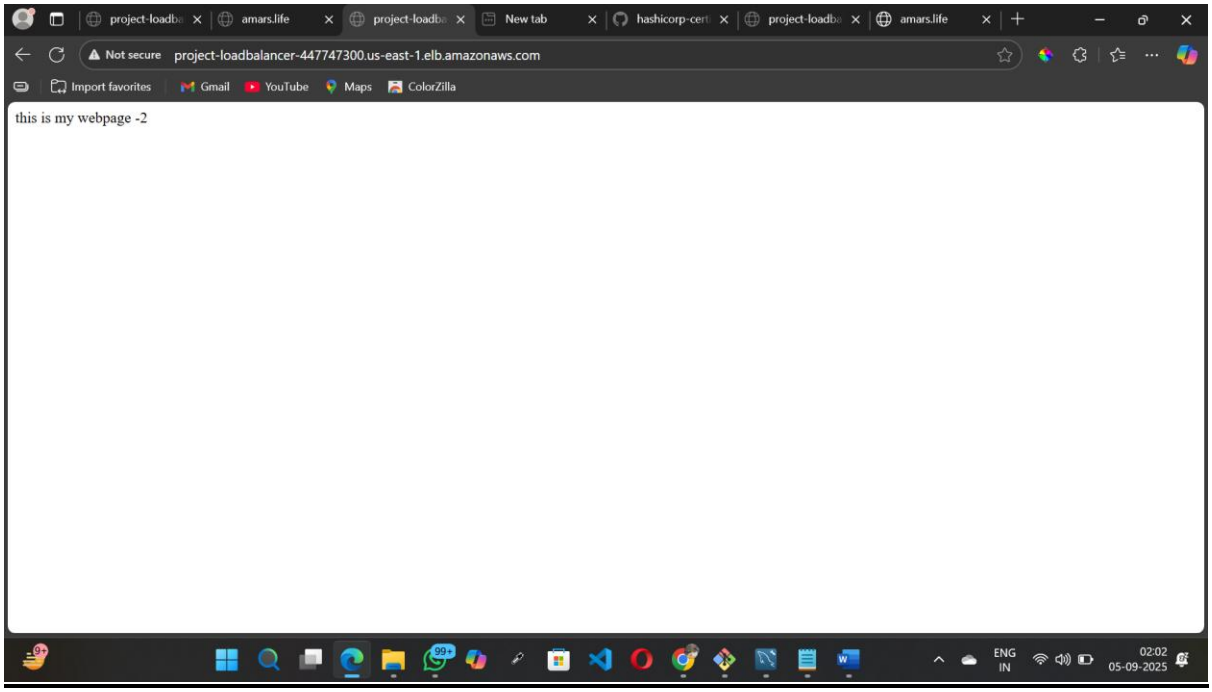


LOAD BALANCER

FIG 4.2:LOAD BALANCER IMAGE

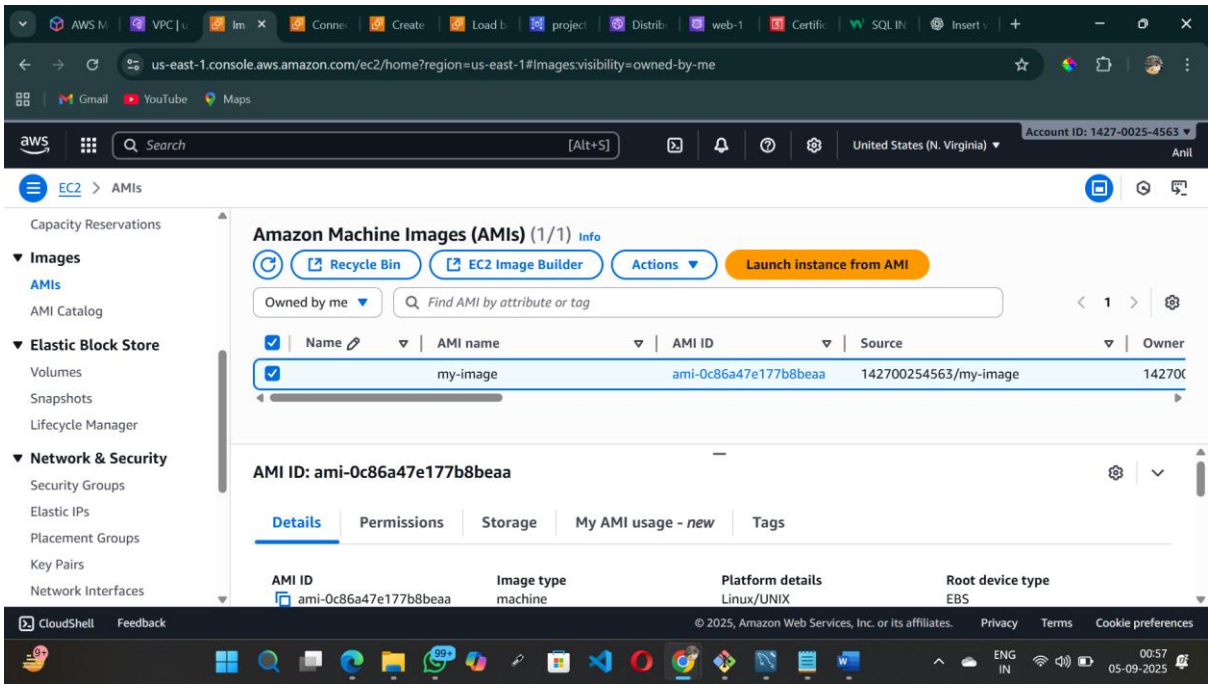


→past DNS check ur server



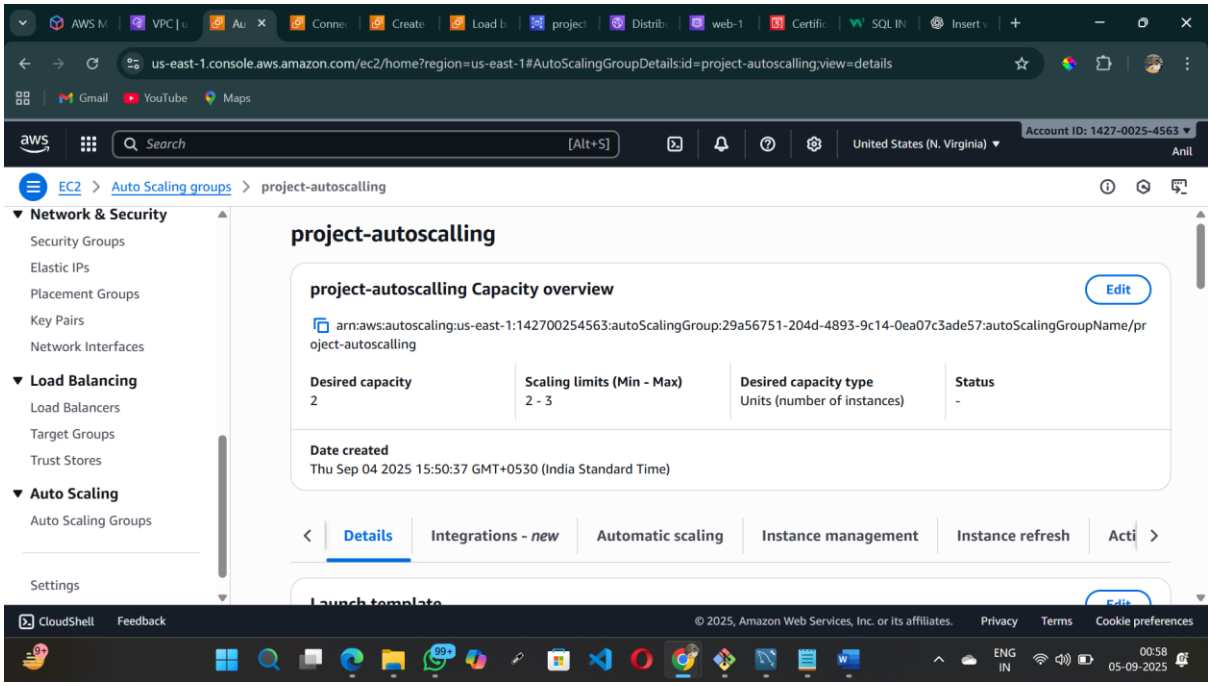
Create **image** from the existing ec2 server (img-1).

Fig 5.1: Amazon Machine Image (AMI)



- Create **Template** for Auto scaling the servers

Fig 5.2 AUTOSCALLING



→AFTER CREATING AUTOSCALLING GROUPS

→GO AND CHECK EC2 INSTANCE NEW 2 SERVER ARE CREATED

→WITH NAME WE MENTIONED NEW-1 & NEW-2

Fig 5.3: Registering Targets

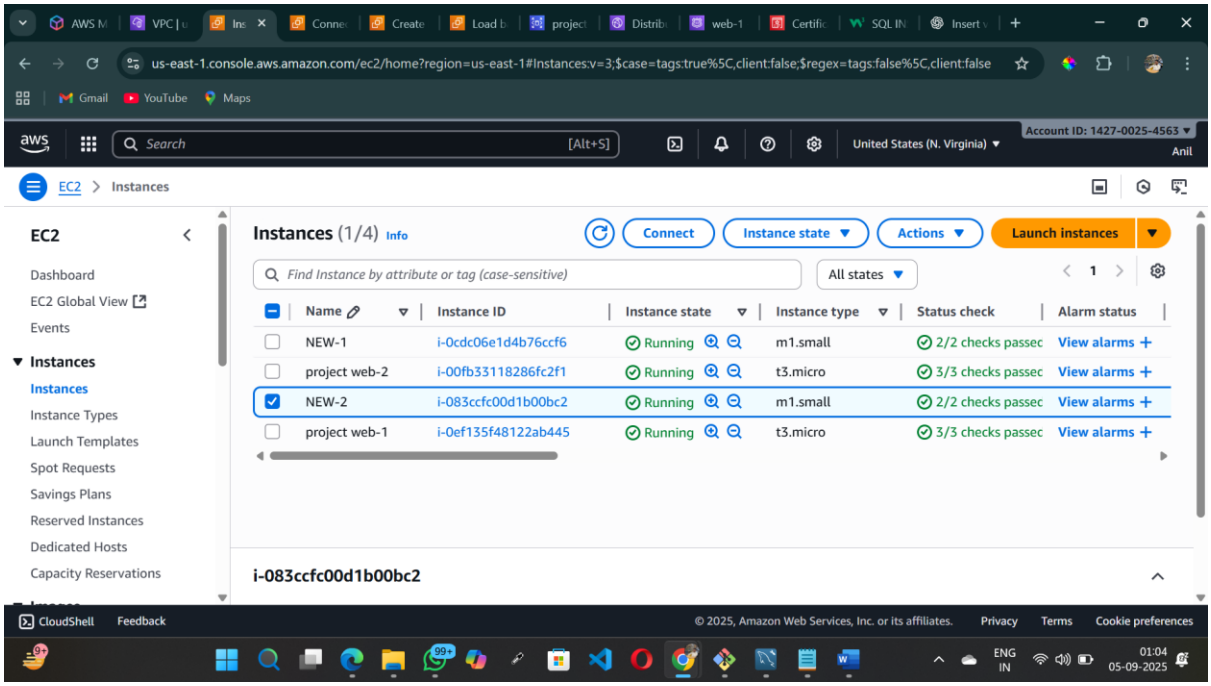
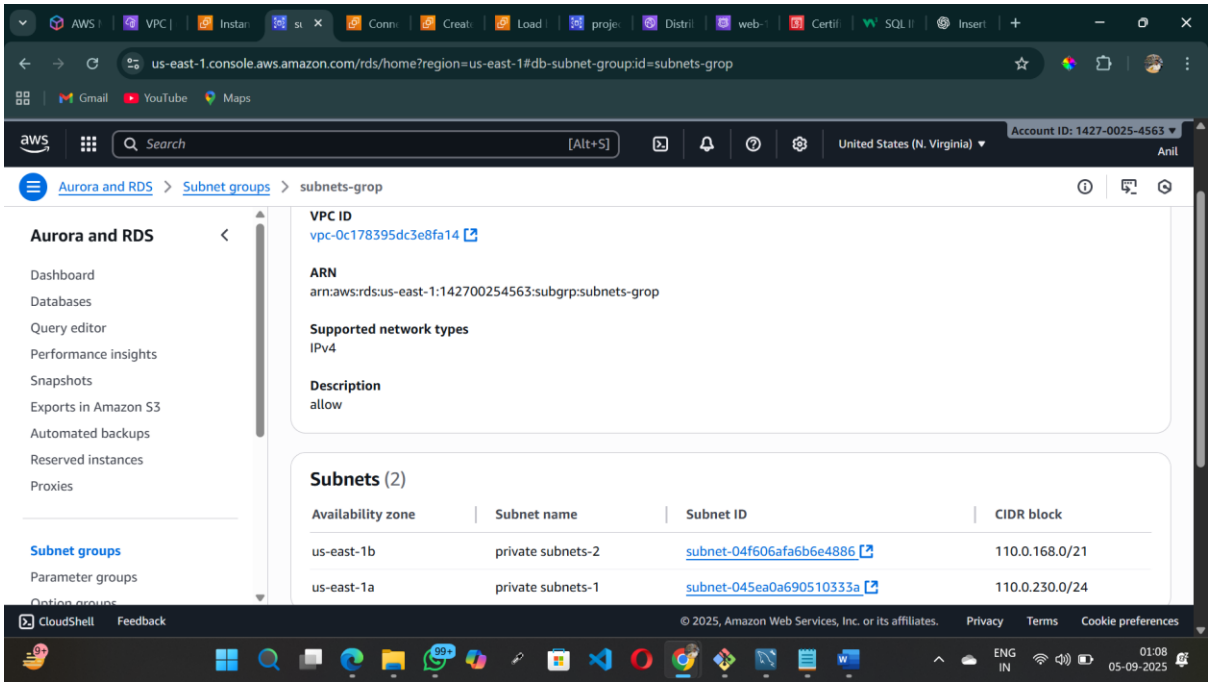
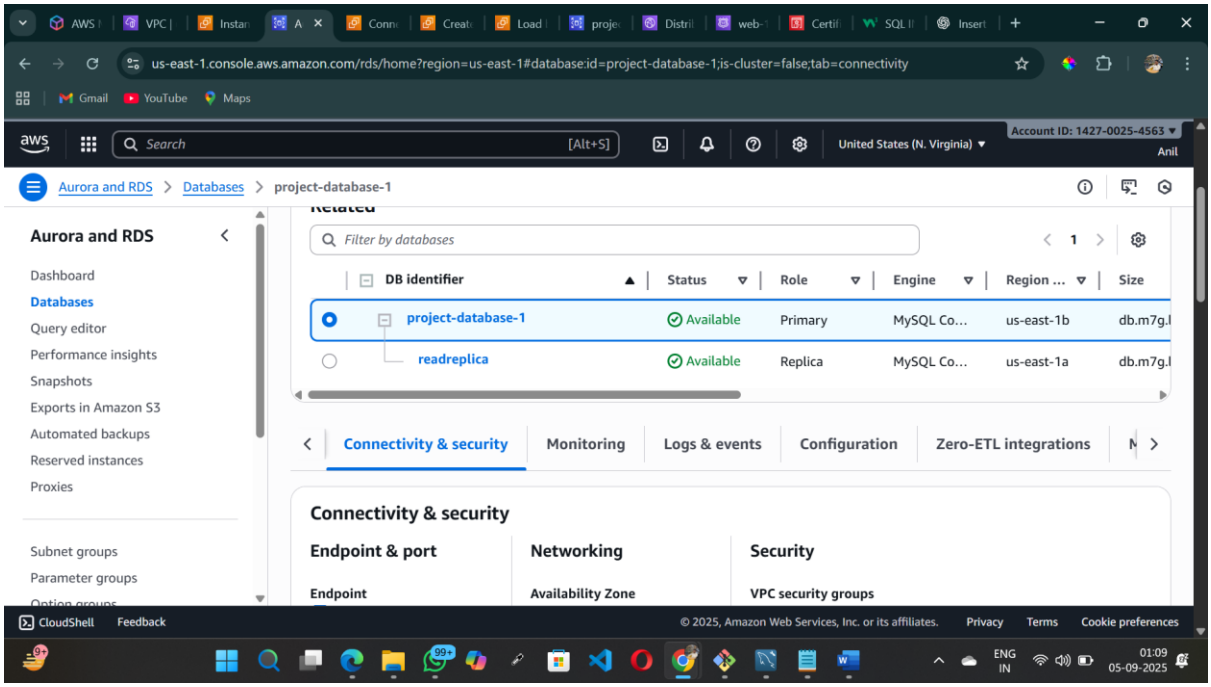


Fig 6.1: RDS

→CREATE A SUBNETSGROUPS



AND CREATE A DATABASE



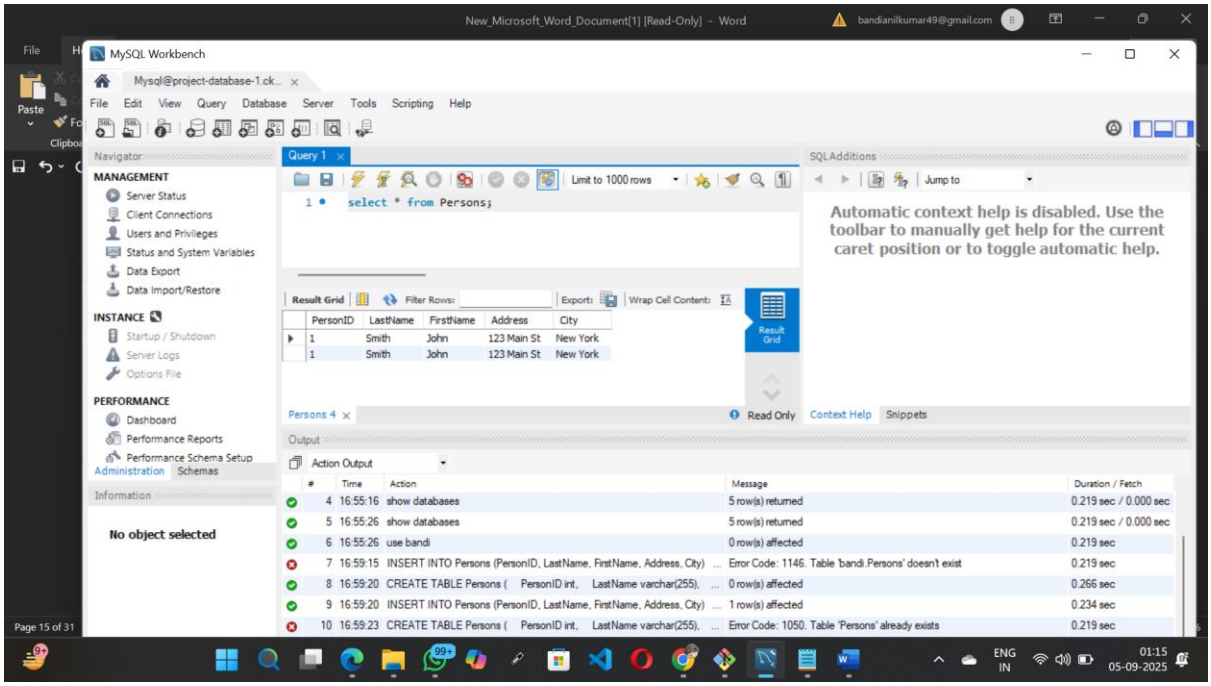
→GO SECURITY GROUPS (ADD INBOUNDRULES (MYSQL&AURORA))

→AND CREATE DATABASES IN MYSQL

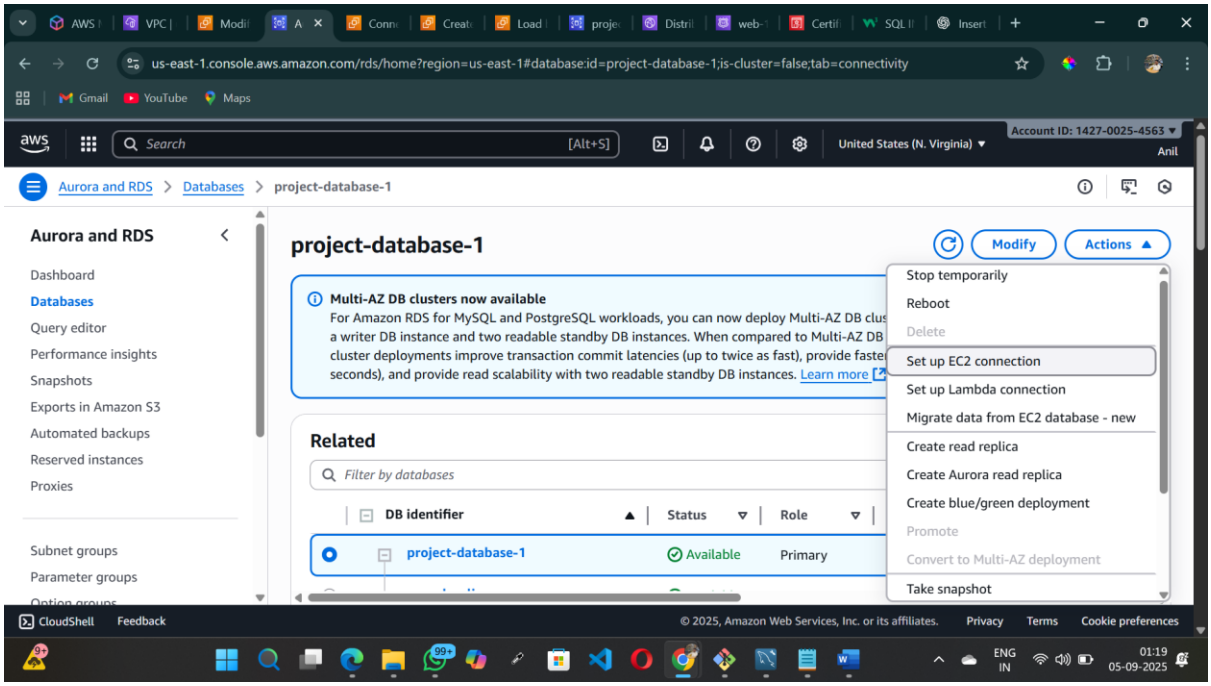
→CREATE TABLETS

→SELECT * FROM TABLES

FIG:6.2—MYSQL



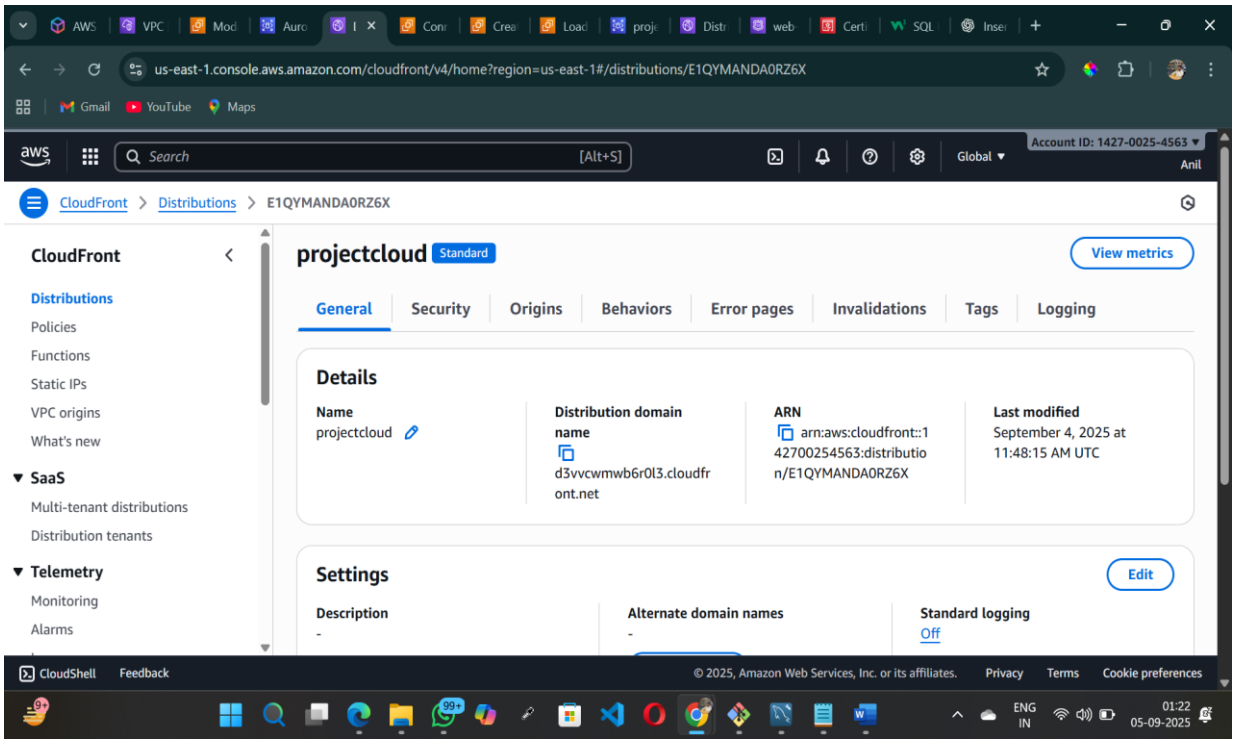
→AFTER DATABASES SET UP EC2 CONNECTION



CLOUD FRONT

➔ CRAETE A CLOUDFRONT AND ATTACH LOADBALANCER
➔ AND ENABLE SECURITY WAF

FIG:7.2 CLOUDFRONT



ROUTE53

→CREATE ROUTE53

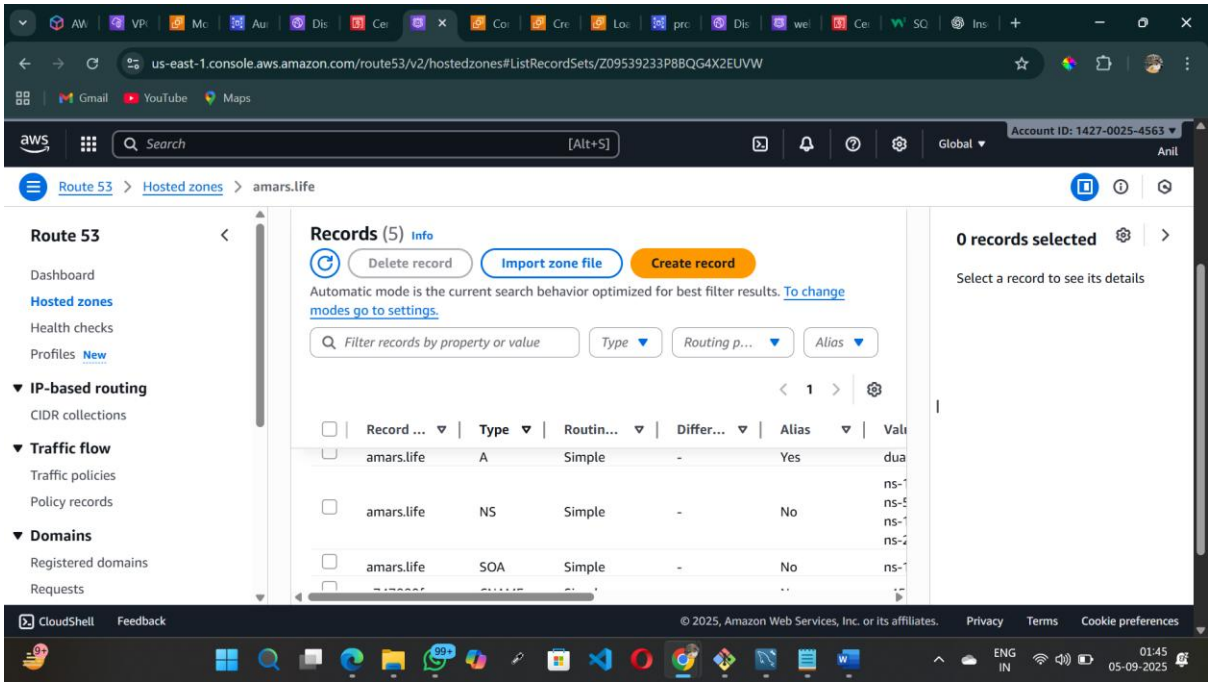
→CREATE HOSTED ZONE

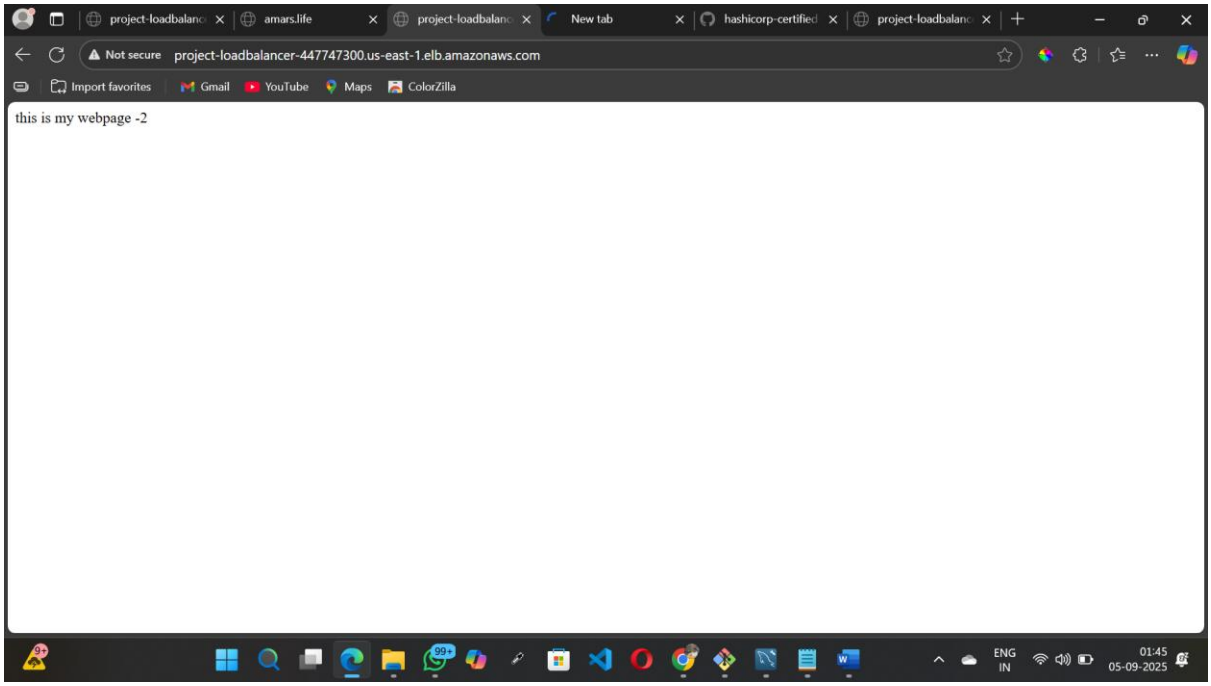
→FULL DOMAIN NAME

→CREATE A 2 CREATE RECORD

→NS DOMAIN 4 SERVER CHANGE

→CNAME ROUTE53





➔ TO SECURE SERVER USING (ACM)

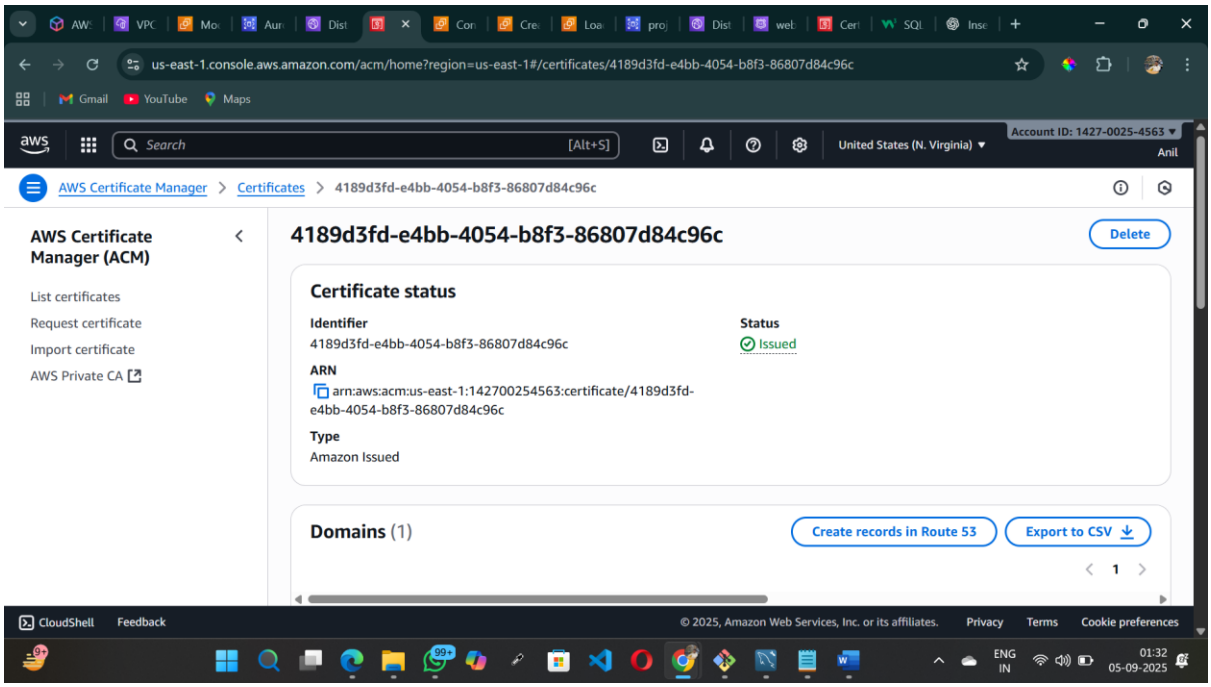
ACM(AMAZON CERTIFICATE MANAGER)

→CREATE A REQUEST CERTIFICATE

→FULL DOMAIN NAME (----------)

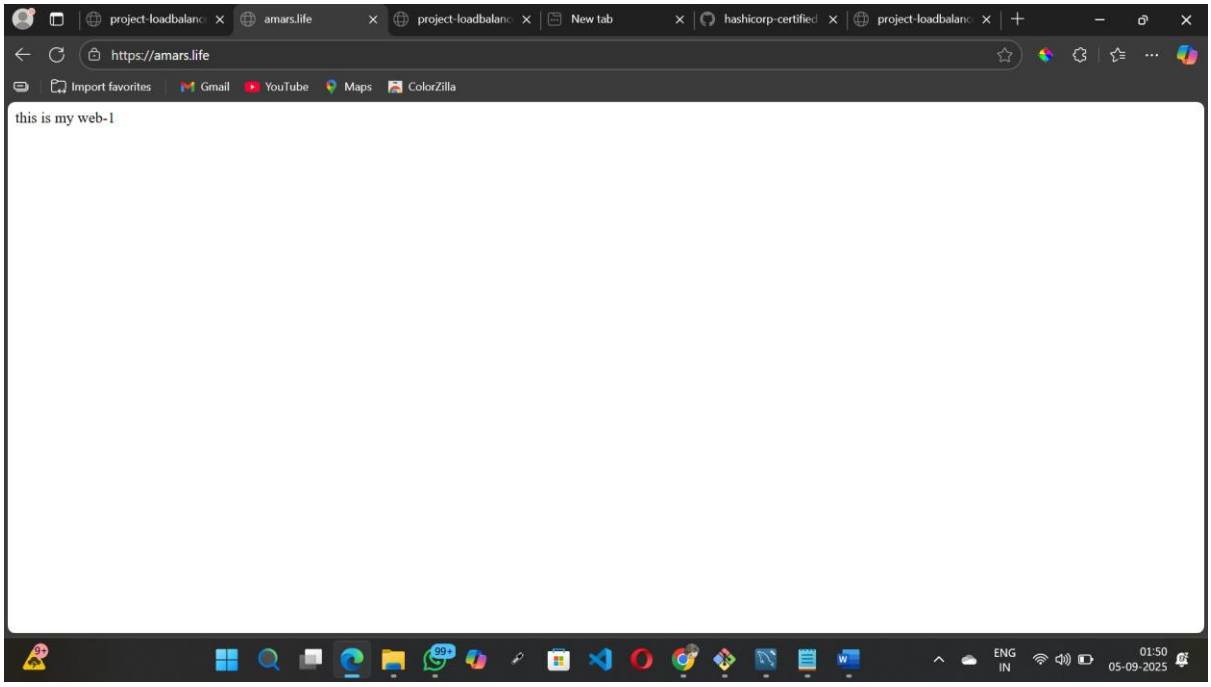
→CREATE RECORDS ROUTE53

→TO SECURE SERVER (HTTPS)



→**GO TO SECURITY GROUPS (HTTPS)**

→**THEN GO TO LOADBALANCER AND ADD LISTENER(HTTPS)**



→USE UR DOMAIN NAME PAST IN BROWSE & WE CAN SEE THE SERVER WAS SECURE

CONCLUSION

The architecture represents a robust, scalable, and secure deployment of a web application on AWS, designed to ensure high availability and performance. It begins with Route 53, which handles DNS resolution and directs user traffic to CloudFront, a global content delivery network that accelerates access while integrating with AWS Certificate Manager (ACM) for secure HTTPS communication. AWS WAF adds a layer of protection against common web threats. Traffic is then routed to an Application Load Balancer (ALB), which distributes requests across EC2 instances hosted in public subnets across multiple Availability Zones, ensuring fault tolerance and scalability. These EC2 instances are managed by an Auto Scaling Group, which dynamically adjusts capacity based on demand. The backend database layer is securely placed in private subnets using Amazon RDS, configured with a writer instance and a reader instance across different zones to support read-heavy workloads and disaster recovery. The entire setup resides within a Virtual Private Cloud (VPC), providing network isolation and control. Overall, this architecture exemplifies AWS best practices by combining performance optimization, security hardening, and operational efficiency—making it an ideal blueprint for hosting modern, production-grade web applications.