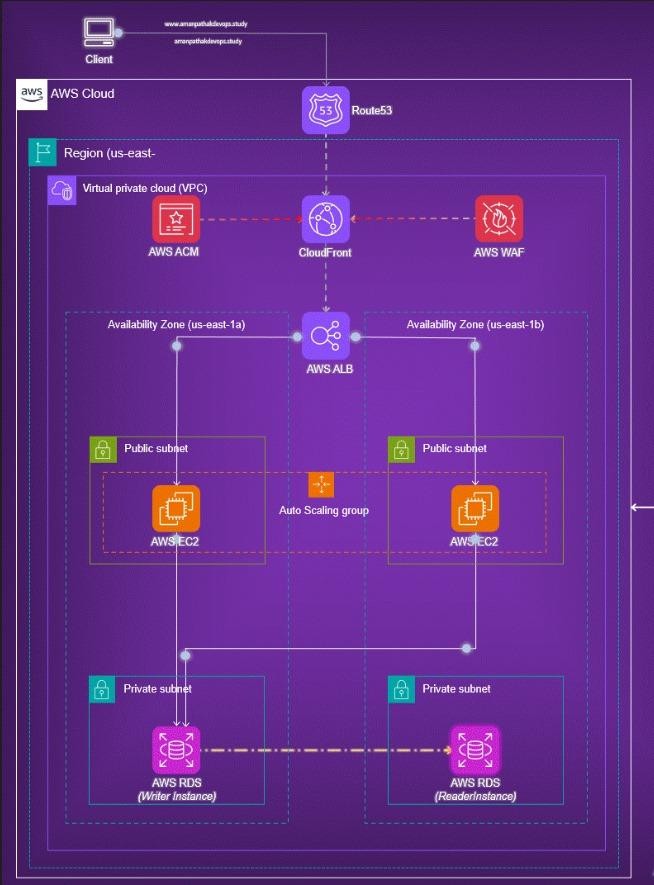
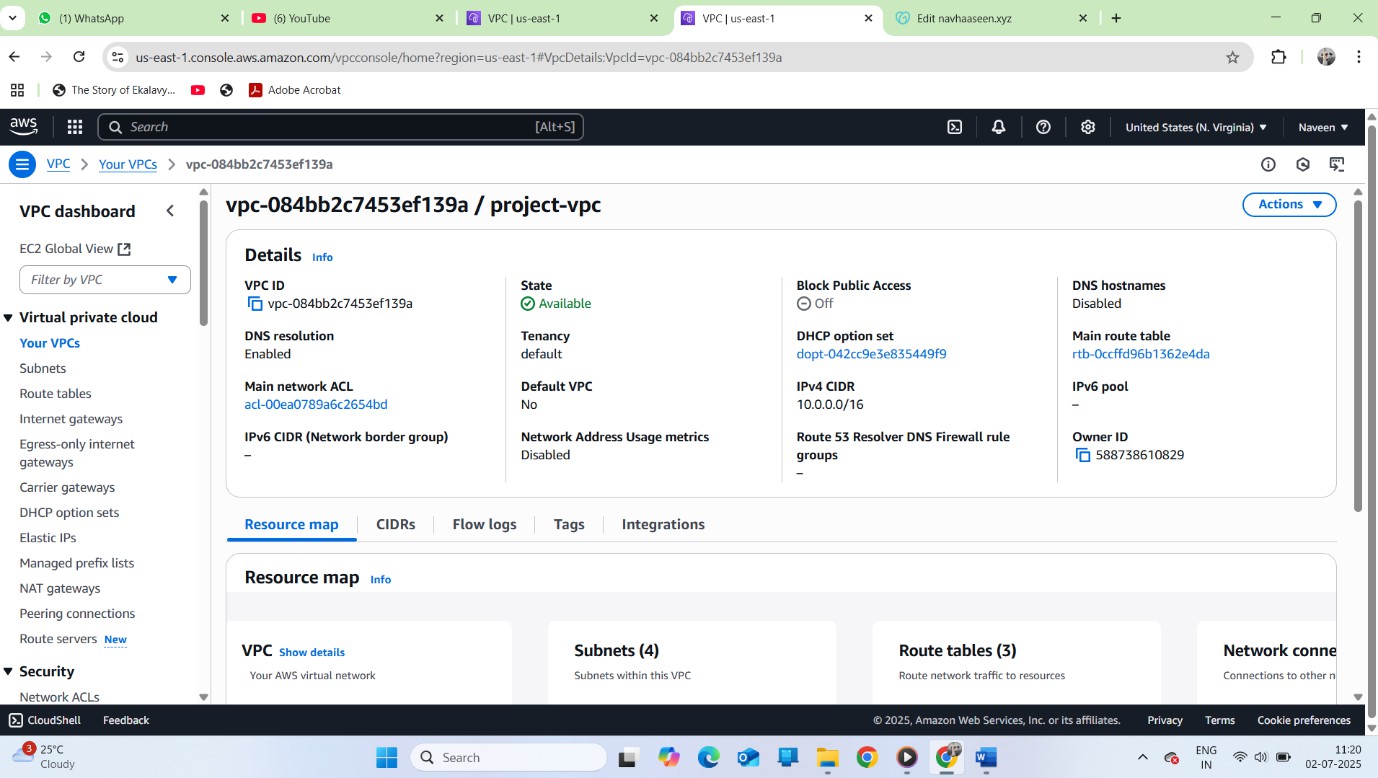
**NAME: -** Matta Veerareddy

**Mail: -** [mattaveerareddy9@gmail.com](mailto:mattaveerareddy9@gmail.com)



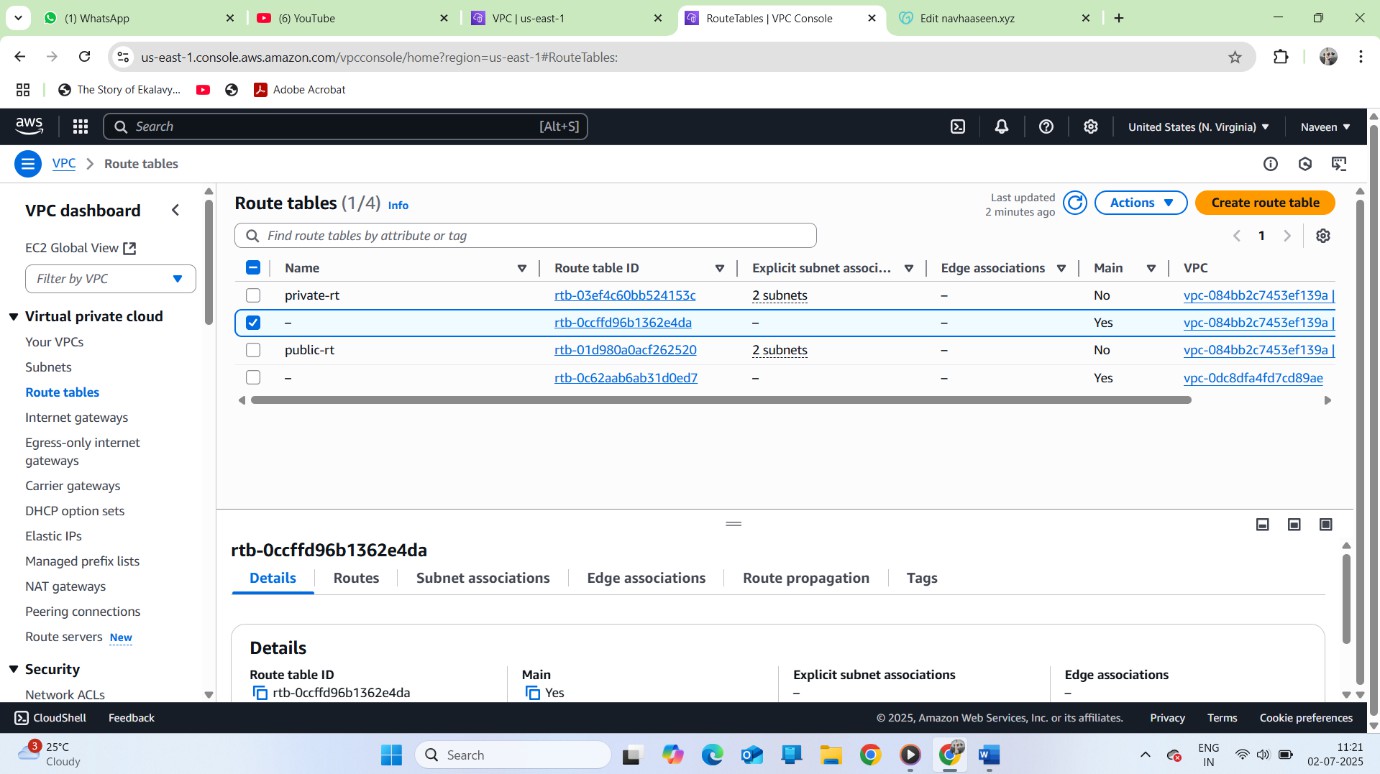
This project showcases a secure and scalable 2-tier architecture deployed on AWS. Route 53 is used for domain name resolution, directing traffic to CloudFront, which acts as a CDN for faster content delivery. AWS WAF provides protection against common web threats, while ACM ensures encrypted HTTPS communication. The infrastructure is hosted within a VPC, divided into public and private subnets across two Availability Zones for high availability.

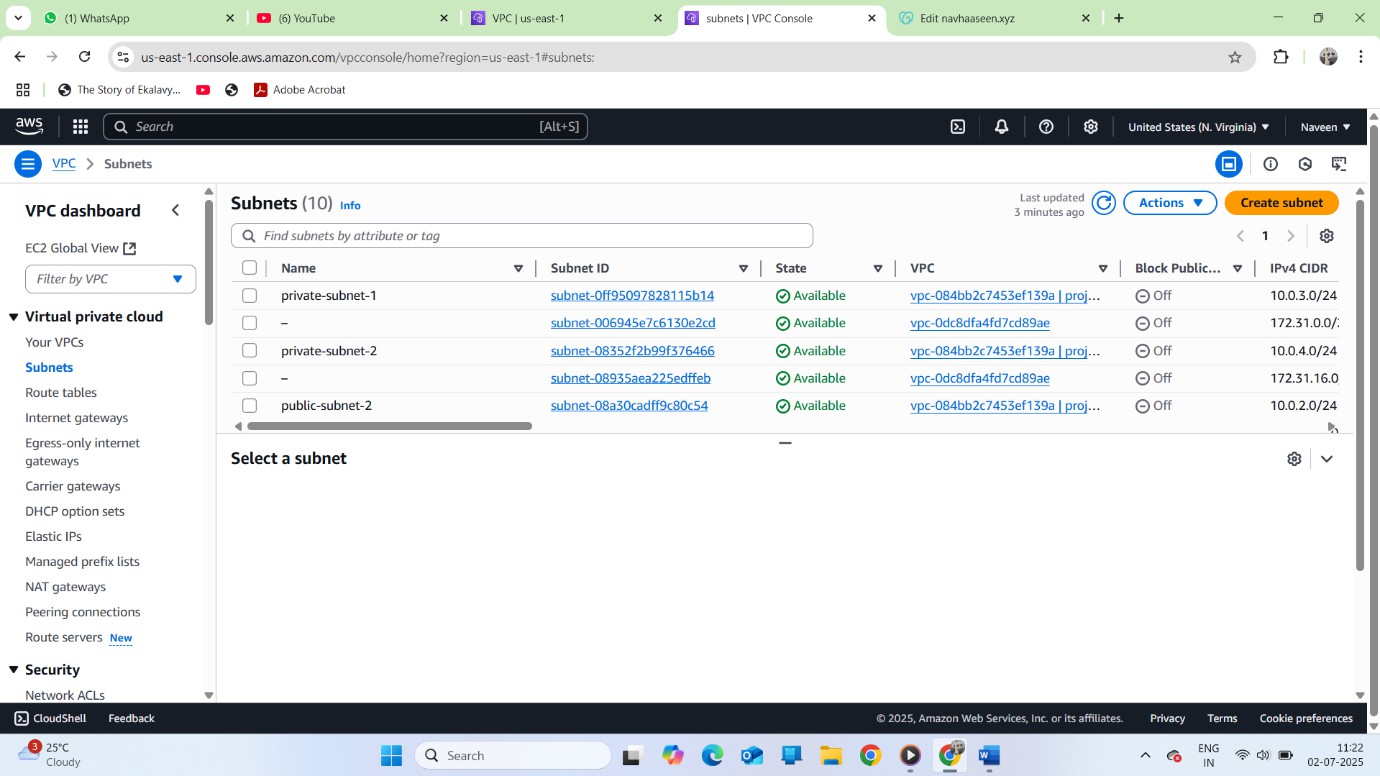
Public subnets host EC2 instances managed by an Auto Scaling Group behind an Application Load Balancer. Private subnets securely house AWS RDS instances, with a Writer for handling transactions and a Reader for read operations. This setup ensures both horizontal scalability and data security. It supports dynamic scaling, high fault tolerance, and better performance. Ideal for web applications requiring robust backend services and minimal downtime.

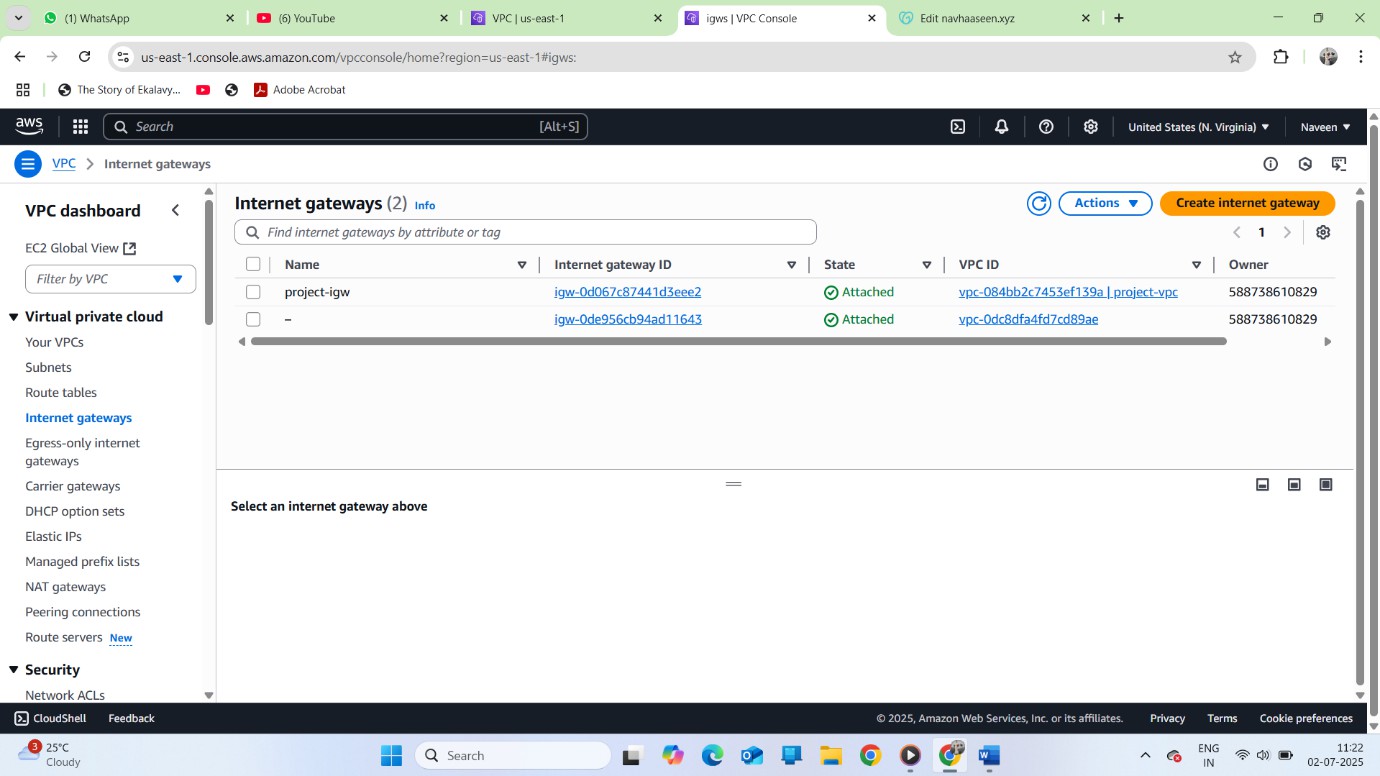


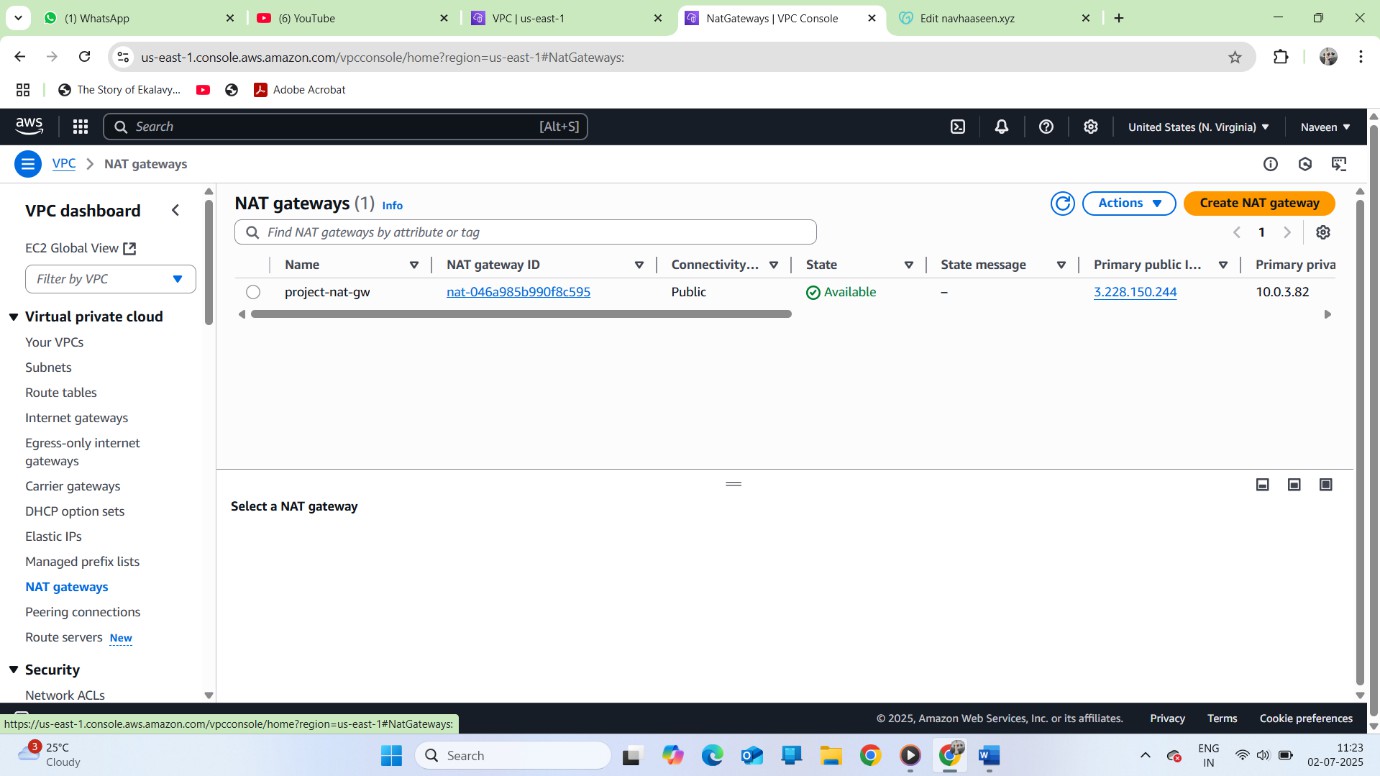
# Step 0: Create VPC and Networking Components

* Created a custom **VPC** in AWS with a CIDR block (e.g., 10.0.0.0/16).
* Created **two public subnets** (one in each Availability Zone).
* Created **two private subnets** (also across different Availability Zones).
* Set up an **Internet Gateway** and attached it to the VPC.
* Created a **Route Table** for the public subnets and associated it.
* Enabled auto-assign public IPs for public subnets.
* Configured **NAT Gateway** in a public subnet to allow internet access for private subnets.
* Created another **Route Table** for private subnets and set routing via the NAT Gateway.
* Verified VPC setup by successfully launching instances in public and private subnets.







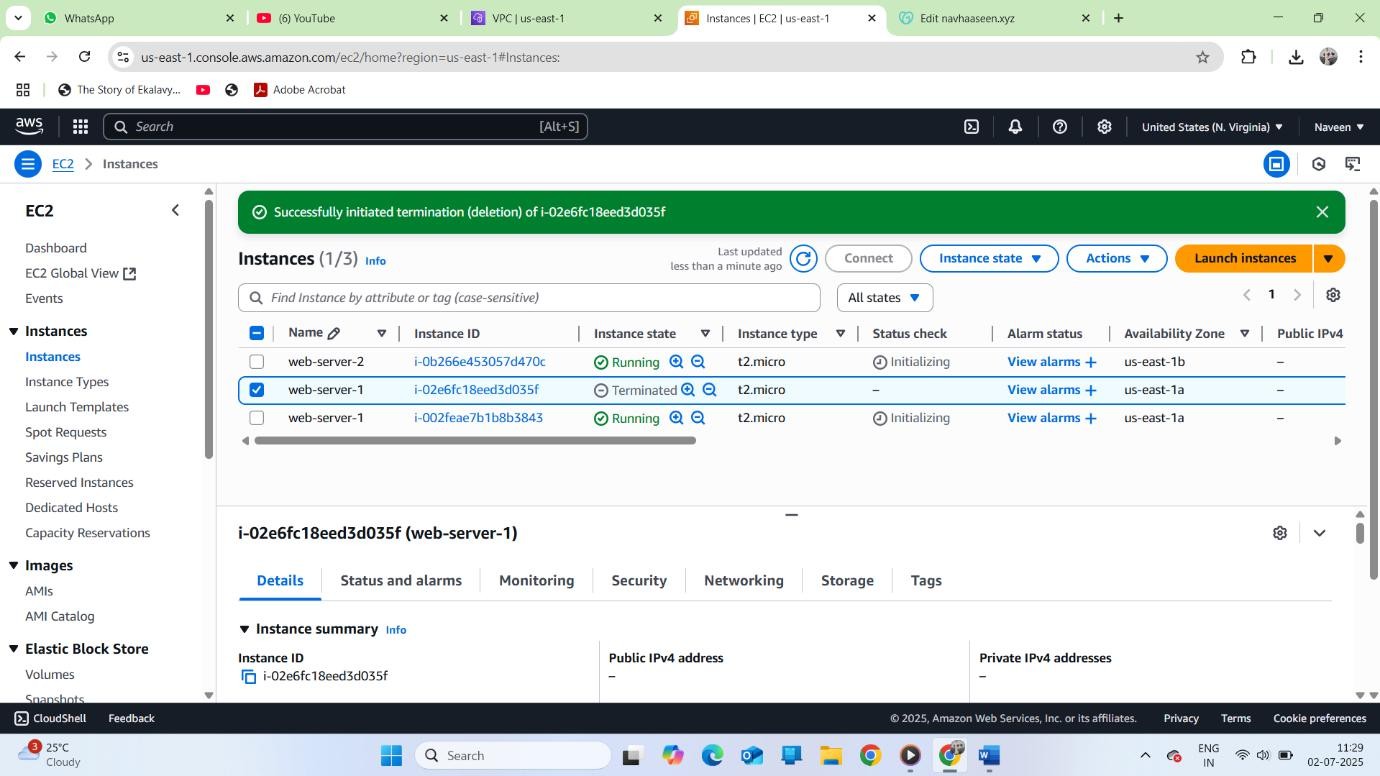


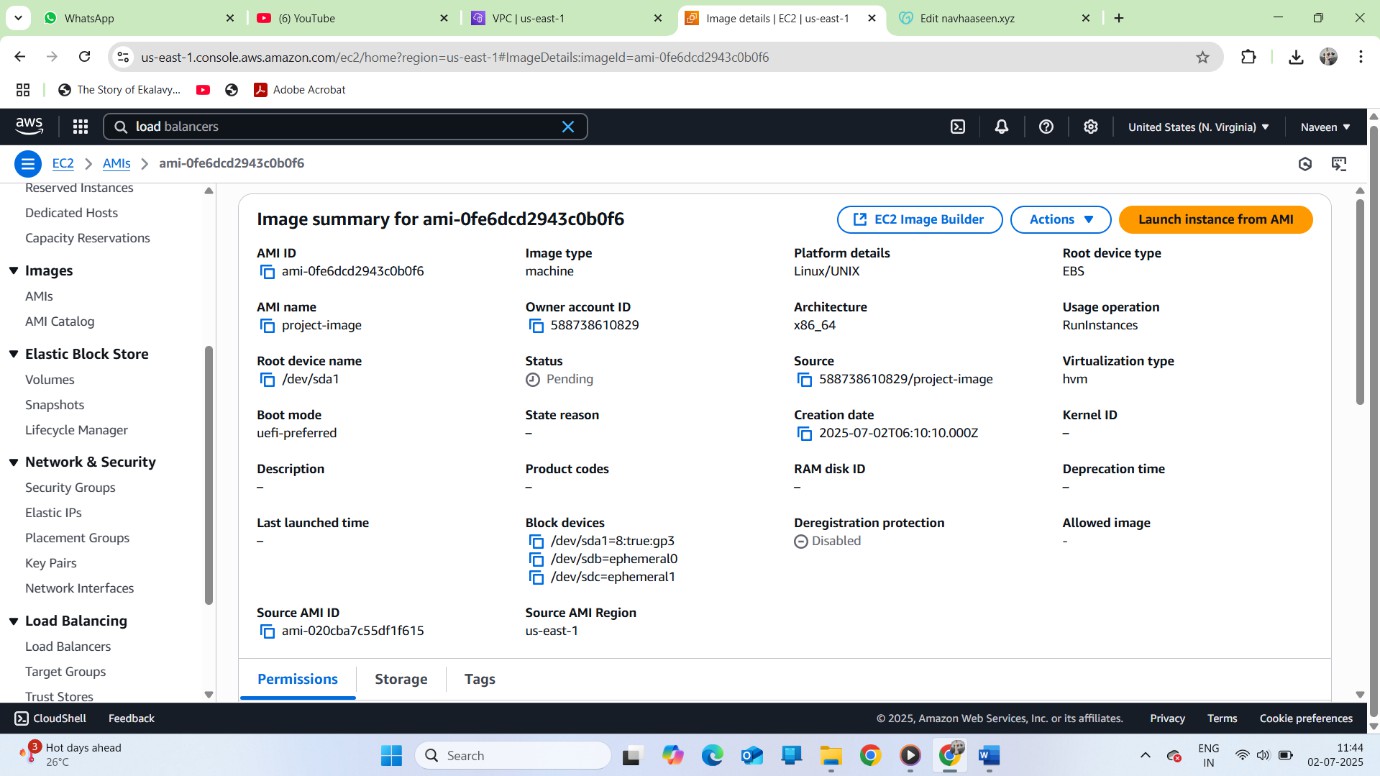
Step 1: Launch EC2 Instances

* Created two EC2 instances in AWS.
* Named them webserver-1 and webserver-2.
* Connected to each instance via SSH using the key pair.
* Installed the Apache web server with:

sudo apt update && sudo apt install apache2 -y

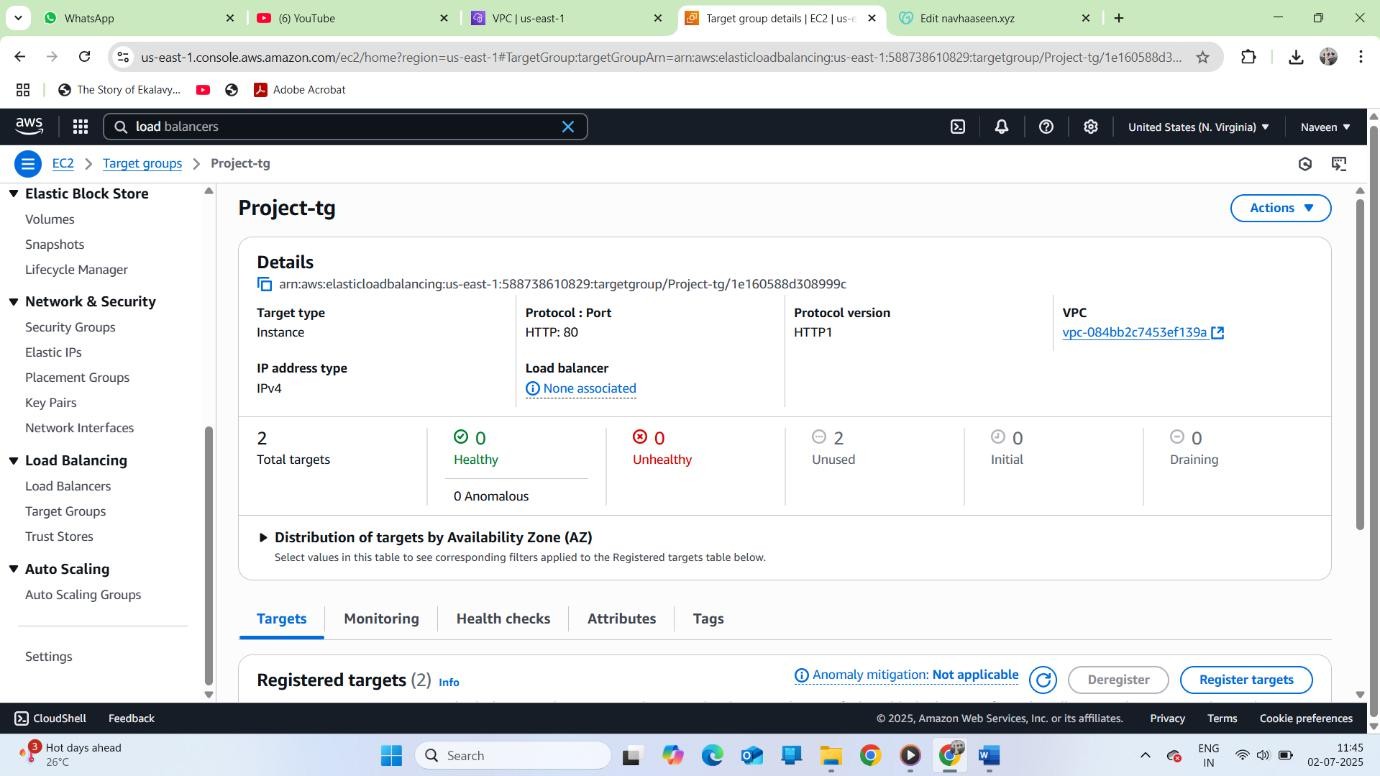
* Created a sample index.html file inside /var/www/html/.
* Verified setup by accessing the public IP in a browser.

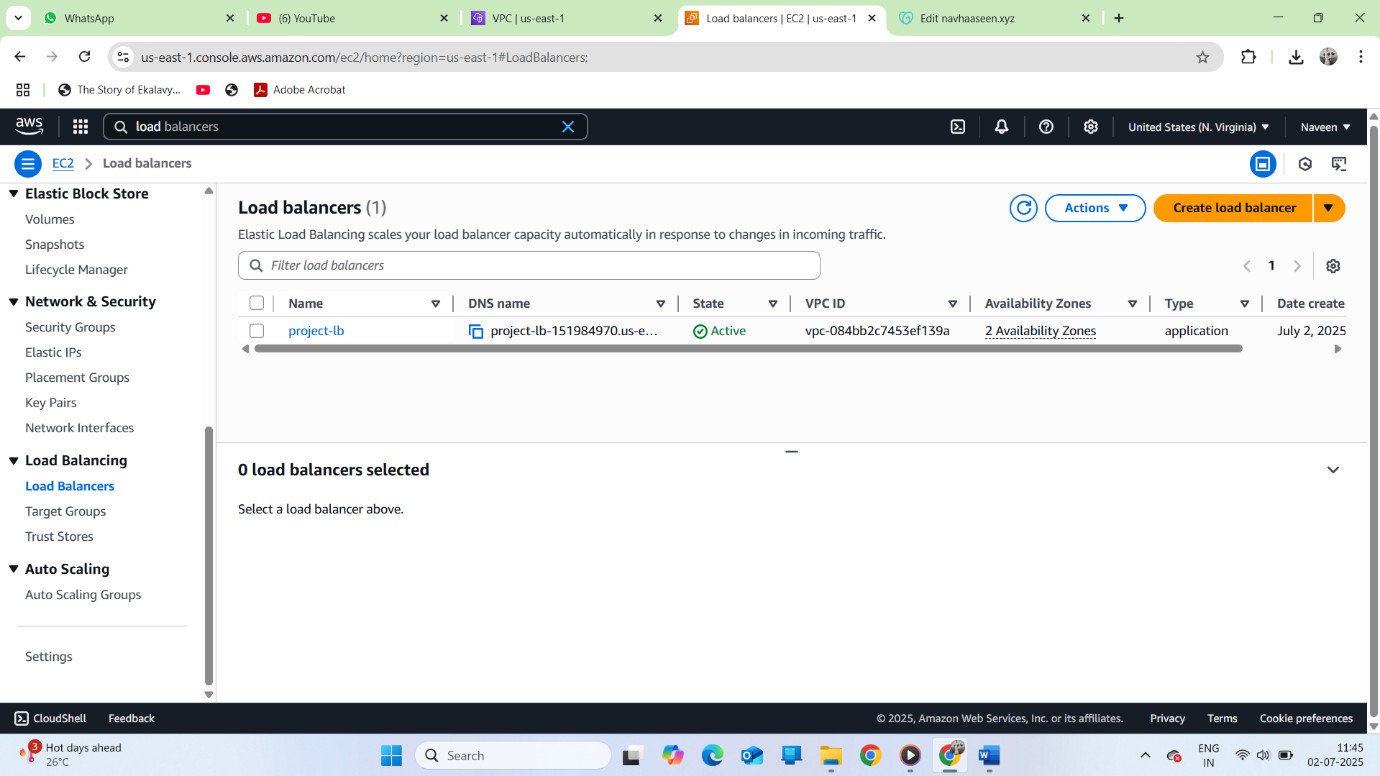


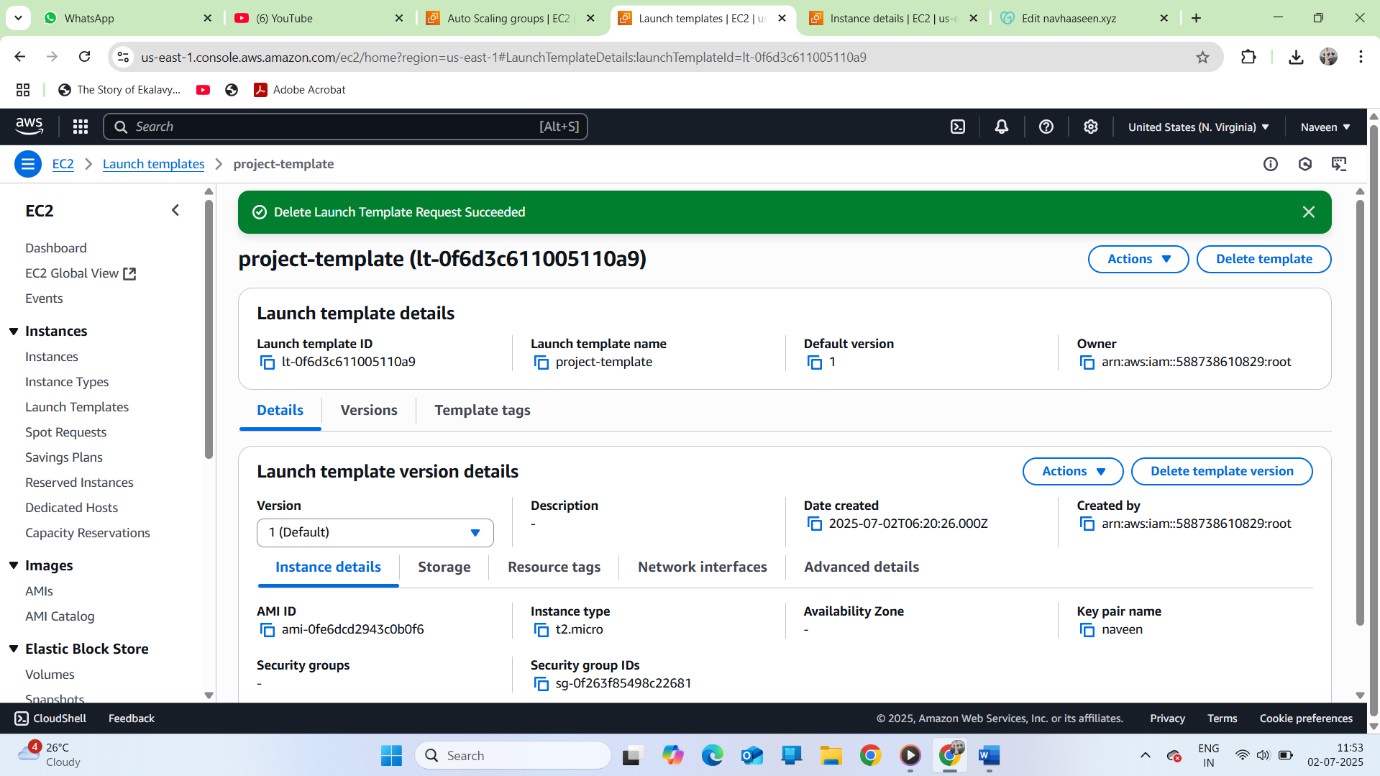


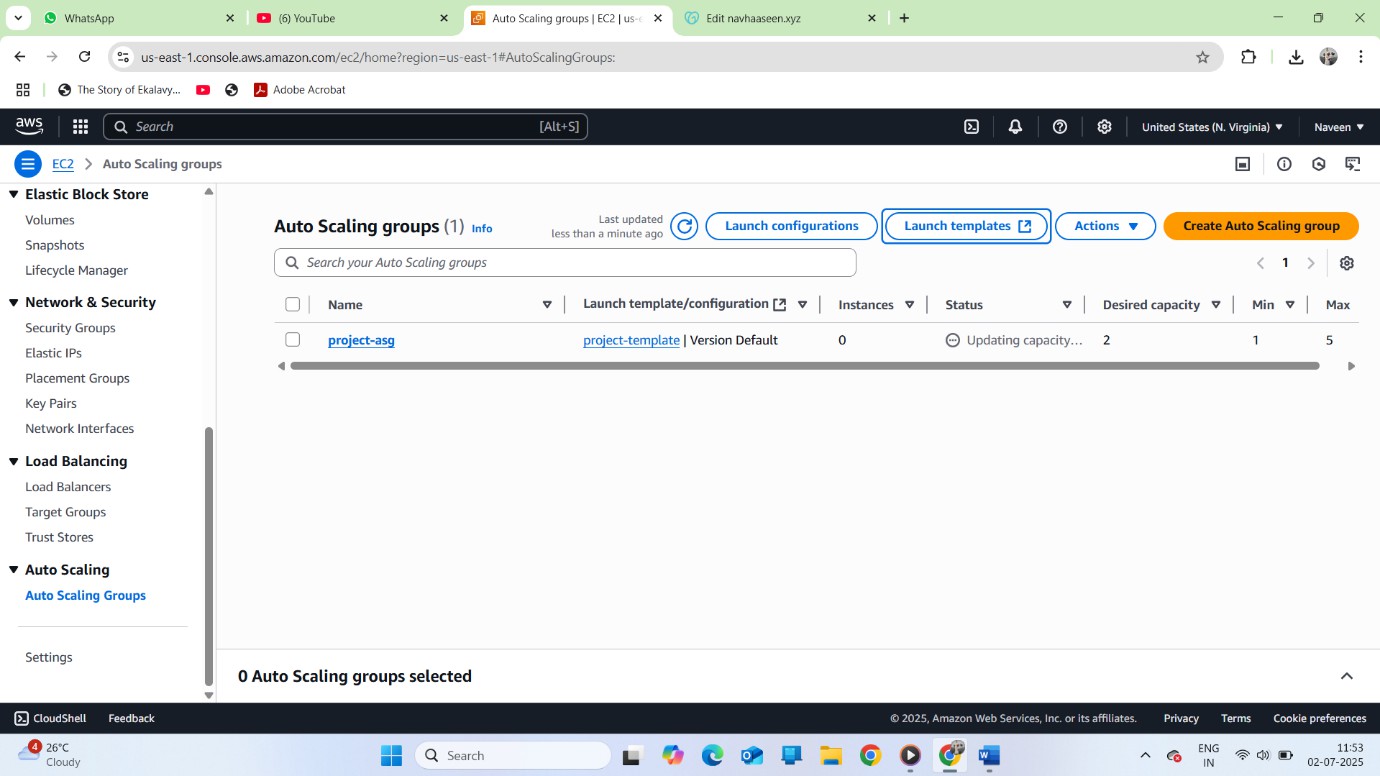
# Step 2: Create Target Group and Load Balancer

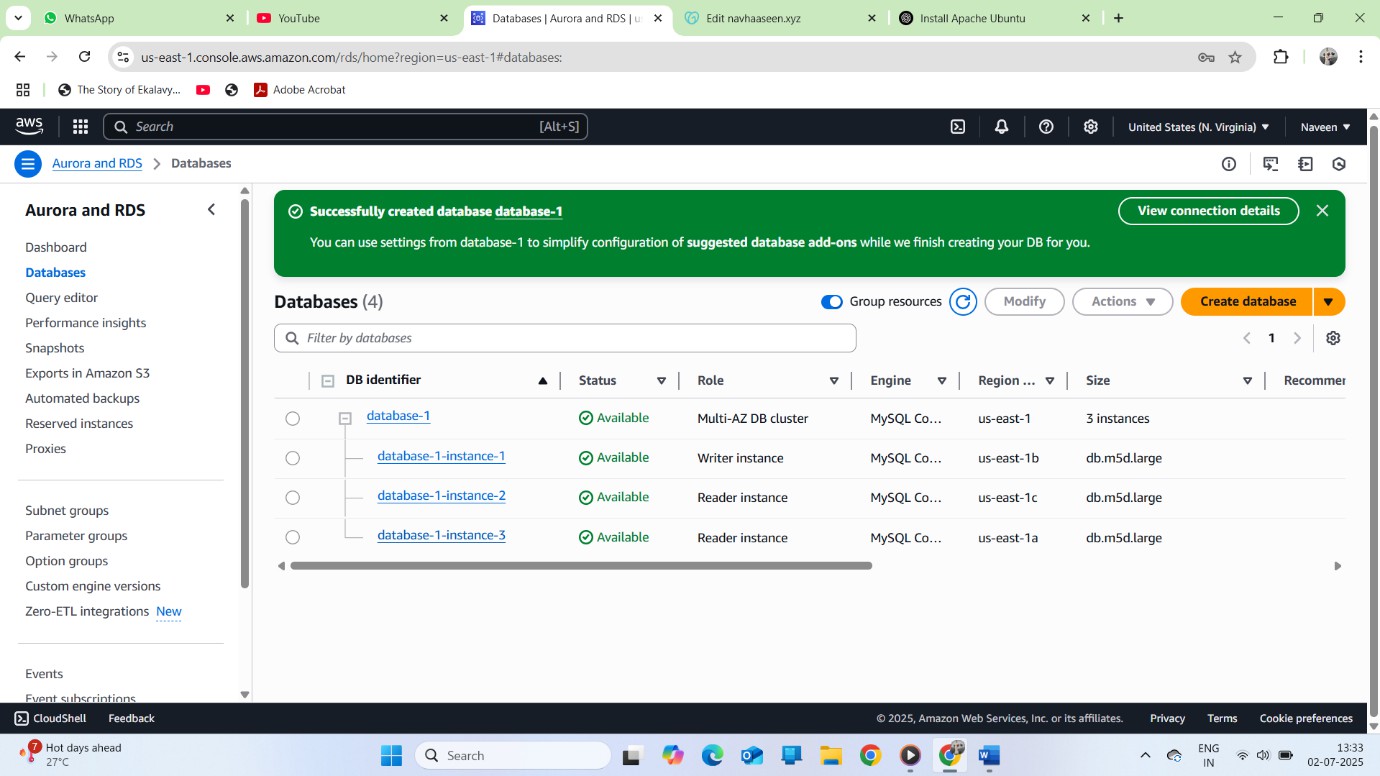
* Created a **Target Group** and registered both EC2 instances to it.
* Set up an **Application Load Balancer**.
* Linked the Load Balancer to the Target Group.
* Verified Load Balancer by accessing it and confirming traffic was balanced between instances.





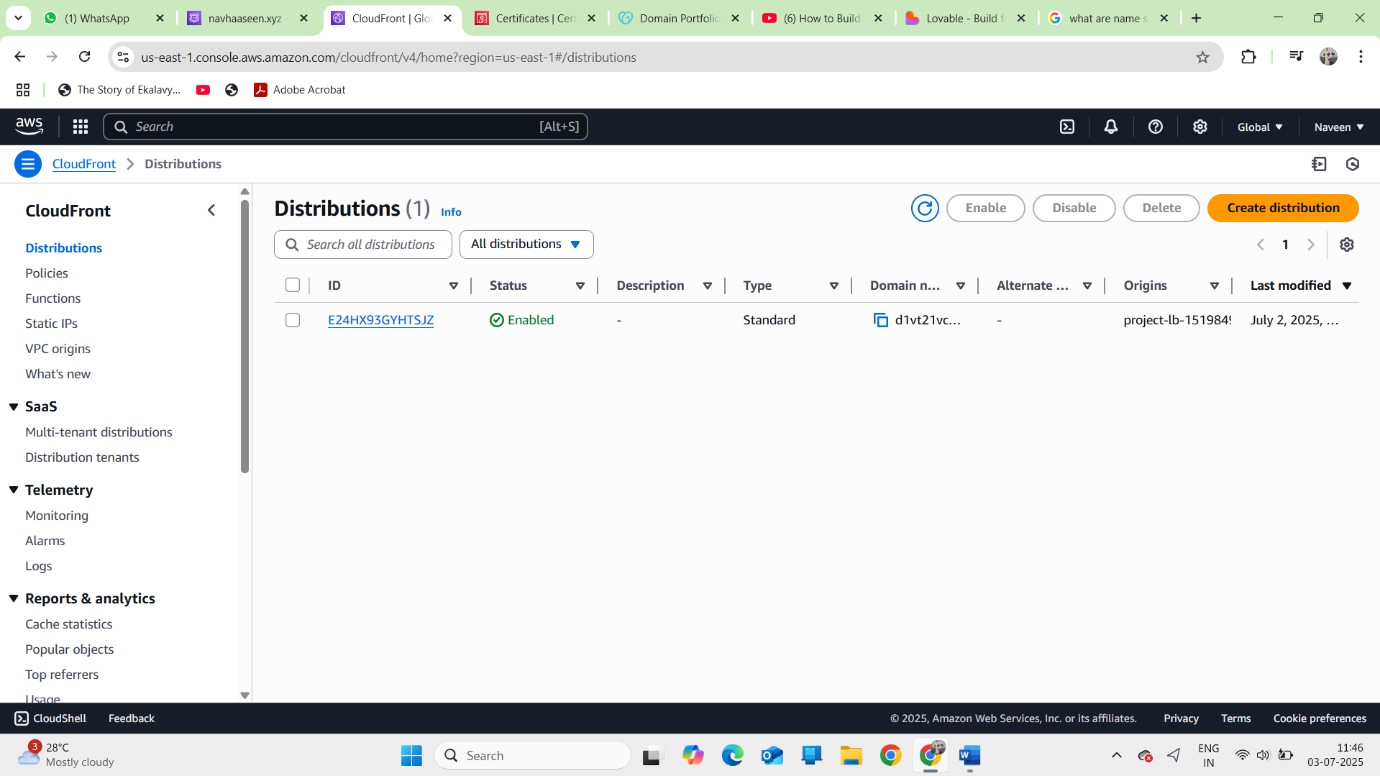






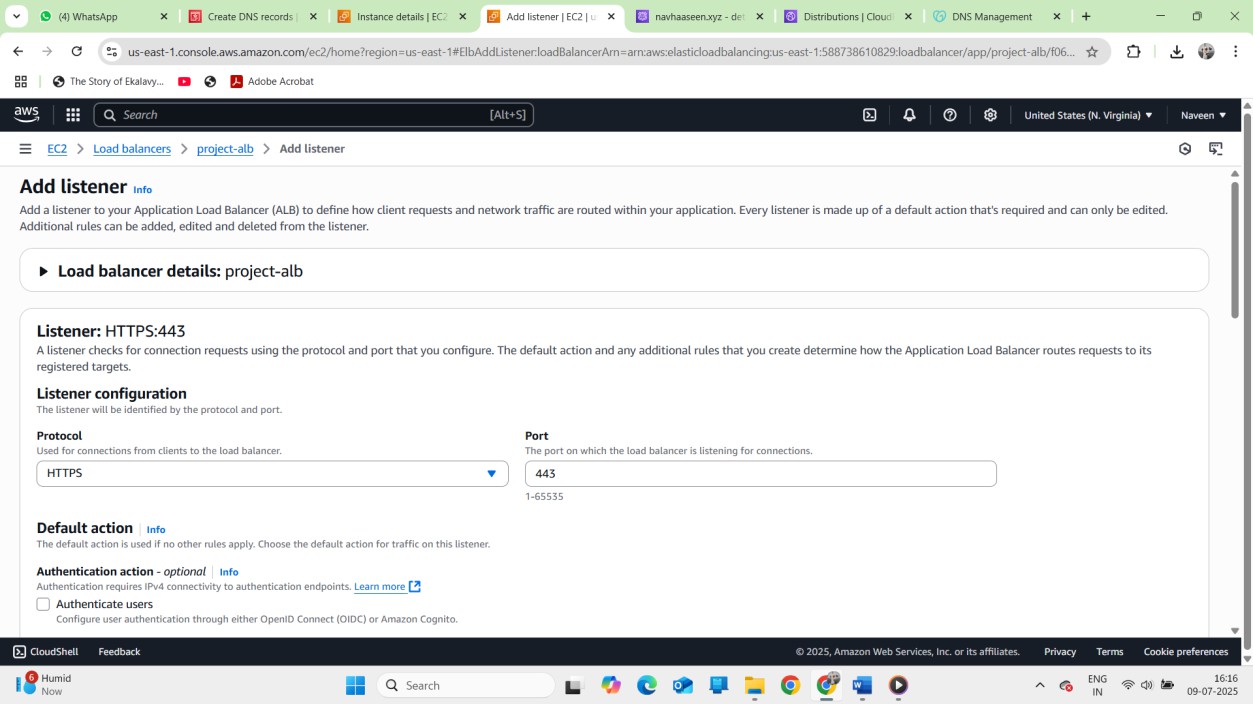
# Step 3: Configure Route 53 and Domain Name

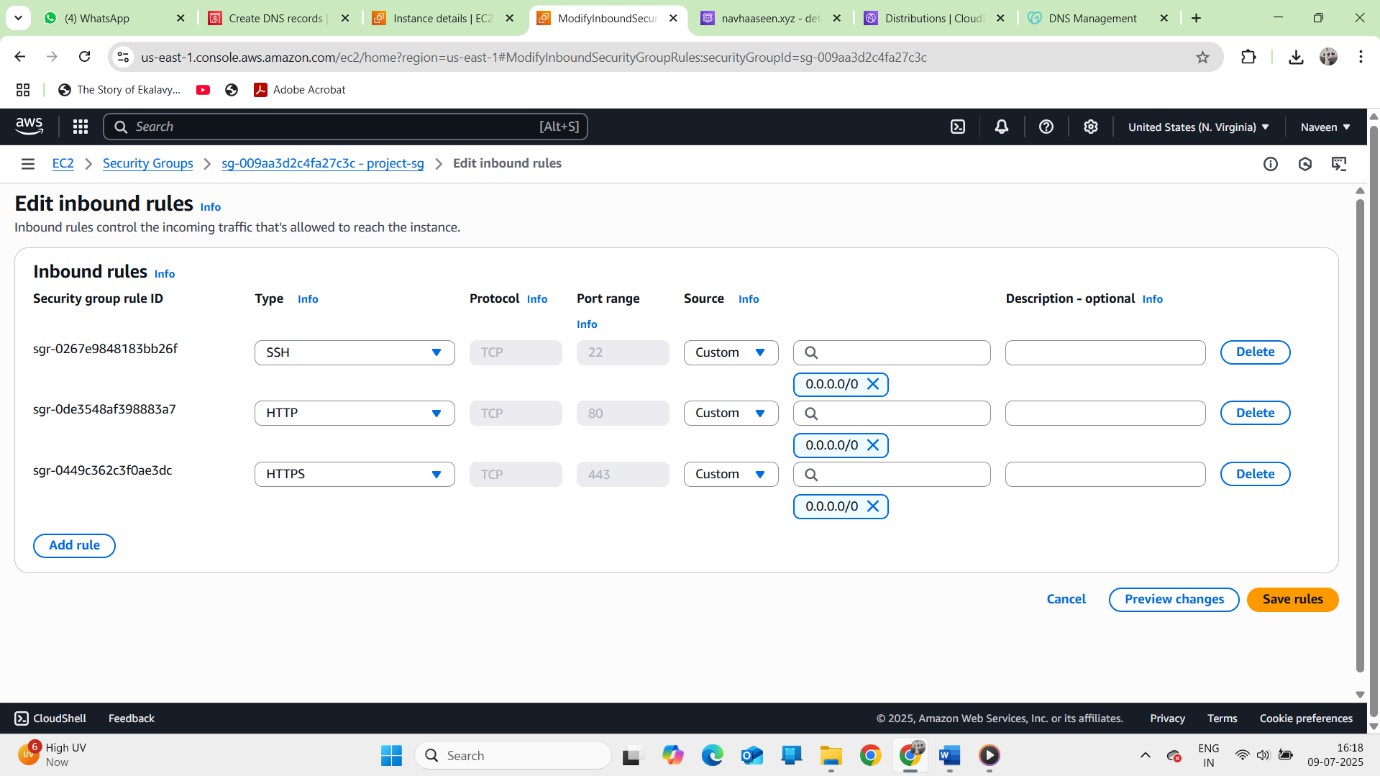
* Created a **Hosted Zone** in Route 53 using a custom domain name.
* Created an **Alias Record** pointing to the Load Balancer.
* Added an additional Alias Record if required.
* Mapped the **Name Servers (NS)** provided by Route 53 to the domain name in a domain registrar (e.g., GoDaddy).
* Verified that the domain name was resolving to the Load Balancer successfully.

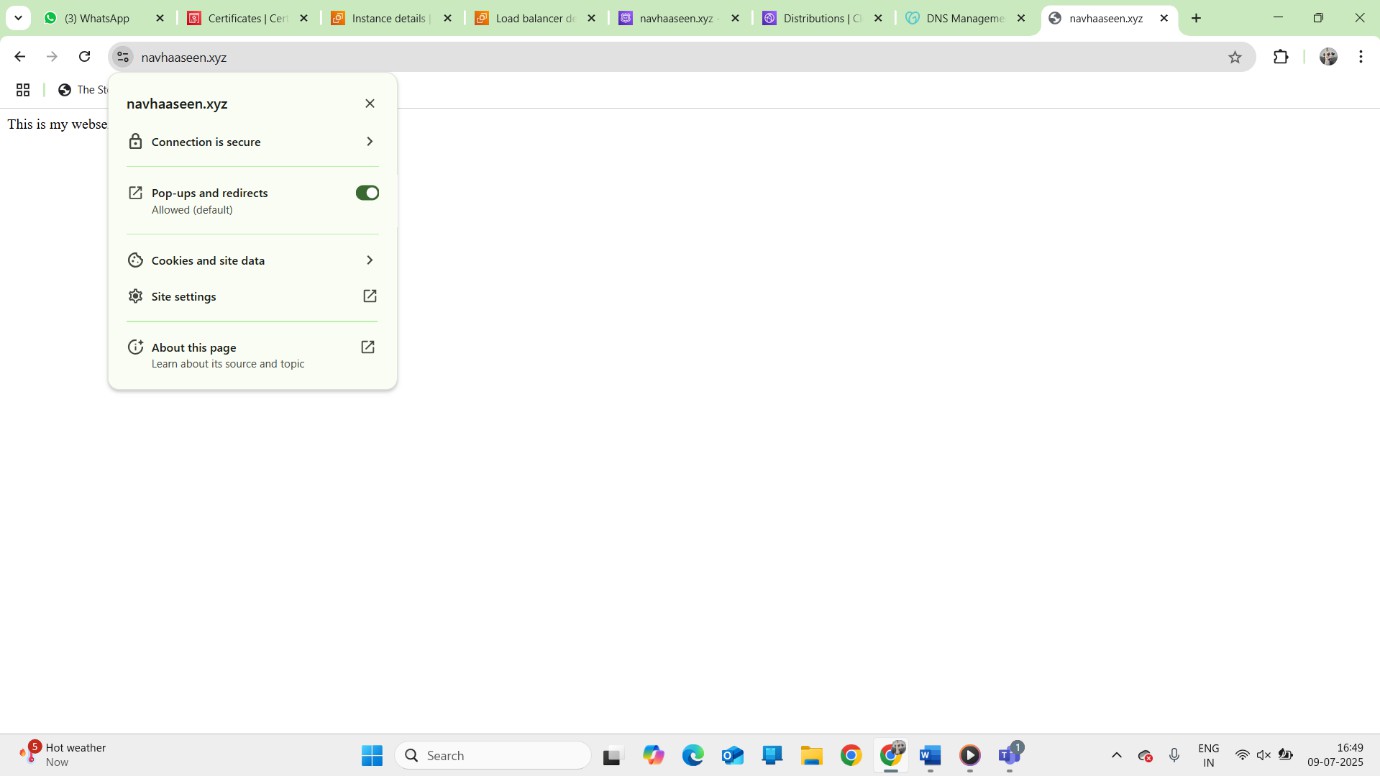


# Step 4: Secure the Domain

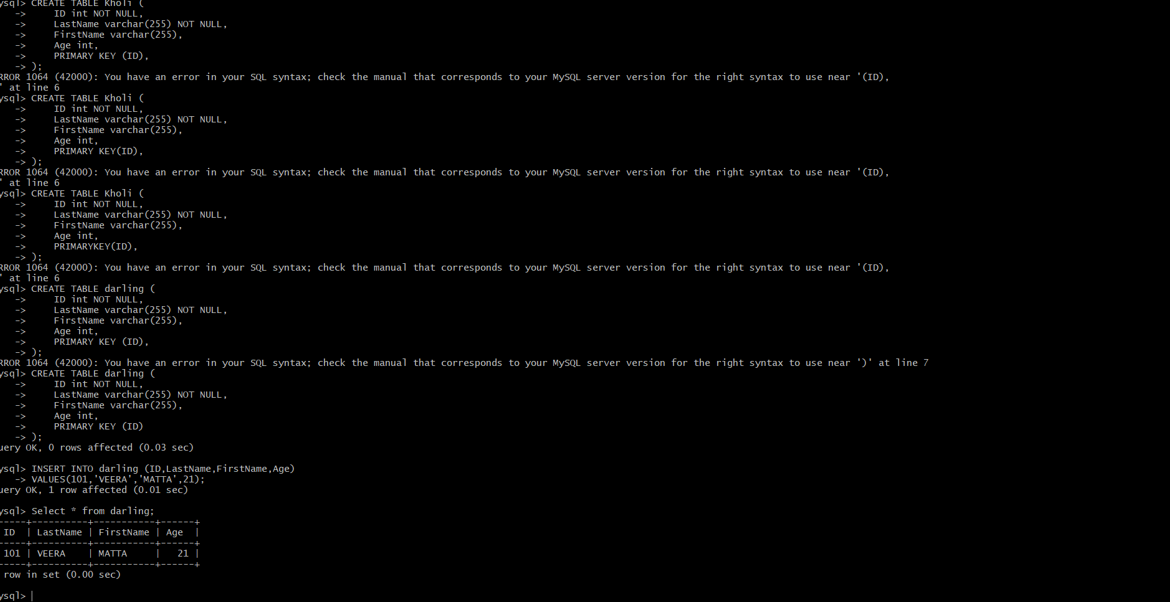
* Initially, the domain used HTTP and was not secure.
* To make it secure:
  + Created a **WAF (Web Application Firewall)**.
  + Requested an SSL/TLS certificate in **AWS Certificate Manager (ACM)**.
  + Set up a **CloudFront Distribution** using the certificate.
  + Added an **HTTPS listener** to the Load Balancer.
* Verified the domain was accessible securely (HTTPS) using the CloudFront domain.











# Final Conclusion:

* Launched and configured two EC2 instances with Apache.
* Set up load balancing using Target Group and Application Load Balancer.
* Registered a domain and configured Route 53 records.
* Mapped domain name servers to Route 53 for DNS resolution.
* Made the domain secure using ACM, WAF, and CloudFront.
* The website was successfully deployed and is accessible securely through the domain.