PROJECT-6

DEPLOYING A PRODUCTION GRADE HIGHLY AVAILABLE & SCALABLE 3 TIER ARCHITECTURE IN AWS

AWS:

AWS (Amazon Web Services) is a cloud computing platform provided by Amazon. It offers a wide range of cloud-based services, including computing power, storage, databases, machine learning, analytics, and more. These services are designed to help businesses and individuals build, deploy, and scale applications or services without needing to maintain physical hardware.

KEY FEATURES OF AWS:

1. Scalability: AWS allows you to scale resources up or down based on demand.
2. Pay-as-You-Go: You pay only for the services you use, which helps manage costs effectively.
3. Global Reach: AWS has data centers in various regions worldwide, allowing users to deploy applications closer to their end-users.
4. Security: AWS provides robust security features like encryption, access control, and compliance with various regulations.
5. Wide Range of Services: Services include:

   Compute: EC2 (Elastic Compute Cloud), Lambda

   Storage: S3 (Simple Storage Service), EBS (Elastic Block Store)

   Databases: RDS (Relational Database Service), DynamoDB

   Networking: VPC (Virtual Private Cloud), Route 53

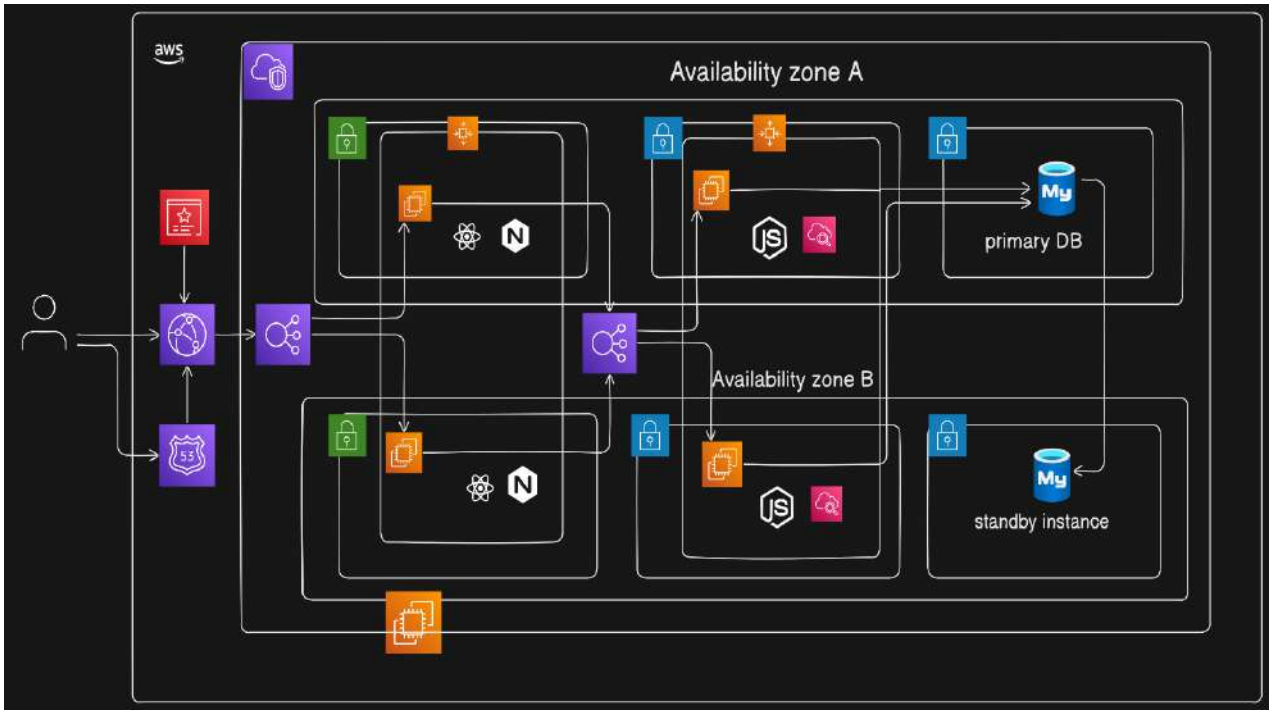
   AI/ML: SageMaker, Rekognition

   Developer Tools: CodeBuild, CodePipeline

Deploying a production-grade, highly available, and scalable 3-tier architecture in AWS involves setting up three distinct layers:

1. Presentation Layer (Front-End)
2. Application Layer (Back-End)
3. Data Layer (Database)

This architecture ensures scalability, fault tolerance, and high availability. Here's how you can design and deploy this architecture:



- This diagram represents a production-grade, highly available, and scalable 3-tier architecture in AWS.
- The pink cloud symbol labeled as Route 53.
- Purple hexagonal icon labeled as a load balancer.
- Orange square icons with a server symbol (representing EC2 instances).
- Green icons with a lock (representing private subnets).
- Blue database icons labeled MySQL (representing relational databases).
- Green private subnet icons indicating these databases are isolated in private subnets.
- The architecture is spread across multiple AZs (Availability Zones) to ensure fault tolerance and minimize downtime.
- Green lock icons indicate that resources are in private subnets for enhanced security.

ROUTE 53:

Amazon Route 53 is a highly scalable and reliable Domain Name System (DNS) web service offered by AWS. Its primary purpose is to route end-user requests to the appropriate resources based on domain names (like example.com, devopsbhargavi.xyz).

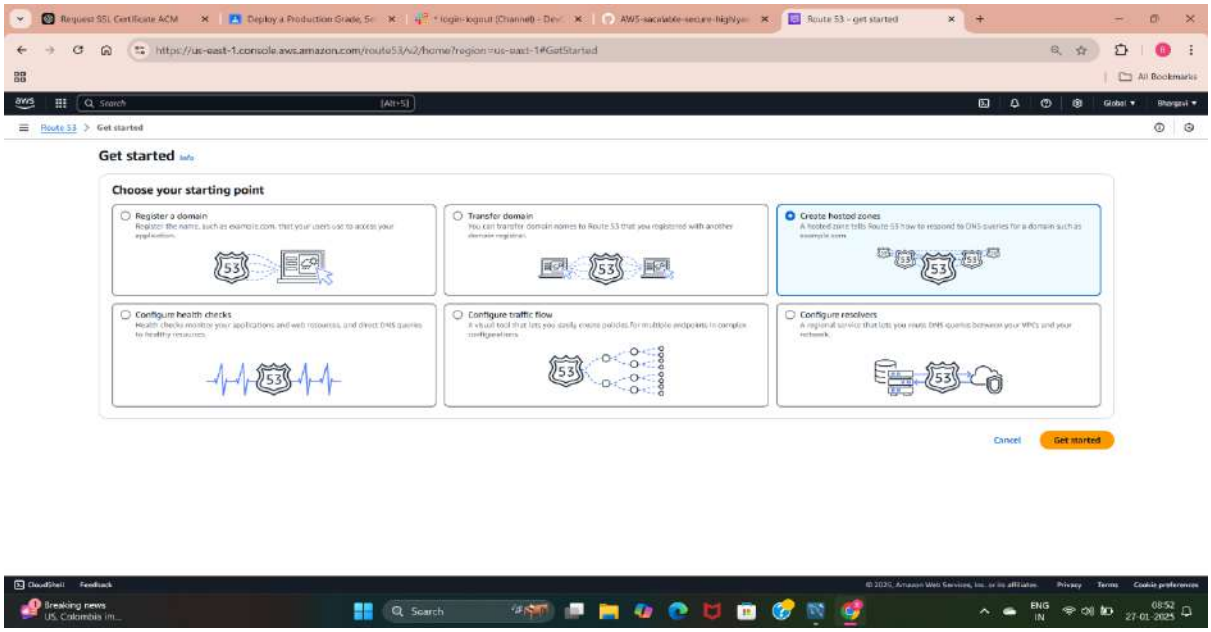
Translates human-readable domain names (e.g., www.example.com) into IP addresses that computers use to communicate (e.g., 192.168.1.1).

Routes end-user traffic to resources like web servers, load balancers, or S3 buckets using a variety of routing policies:

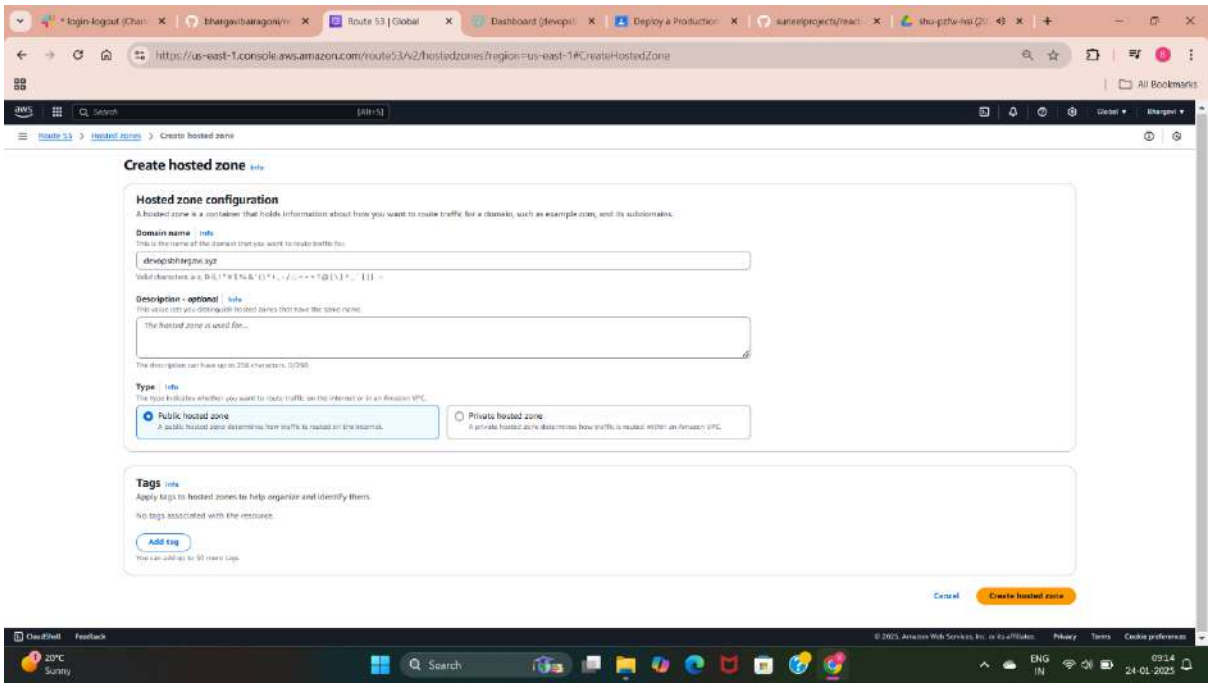
- Simple Routing: Maps one domain name to a single resource.
- Weighted Routing: Distributes traffic among multiple resources based on assigned weights.
- Latency-based Routing: Directs users to the resource with the lowest latency.
- Failover Routing: Redirects traffic to a backup resource if the primary resource fails.
- Geolocation Routing: Routes traffic based on the user's geographic location.
- Multi-Value Routing: Provides multiple healthy endpoints for DNS responses.

- Continuously monitors the health of endpoints (e.g., EC2 instances, web servers).
- Redirects traffic away from unhealthy resources automatically.
- Navigate to the AWS Management Console and open the Route 53 service.

STEP 1: CONFIGURING ROUTE53

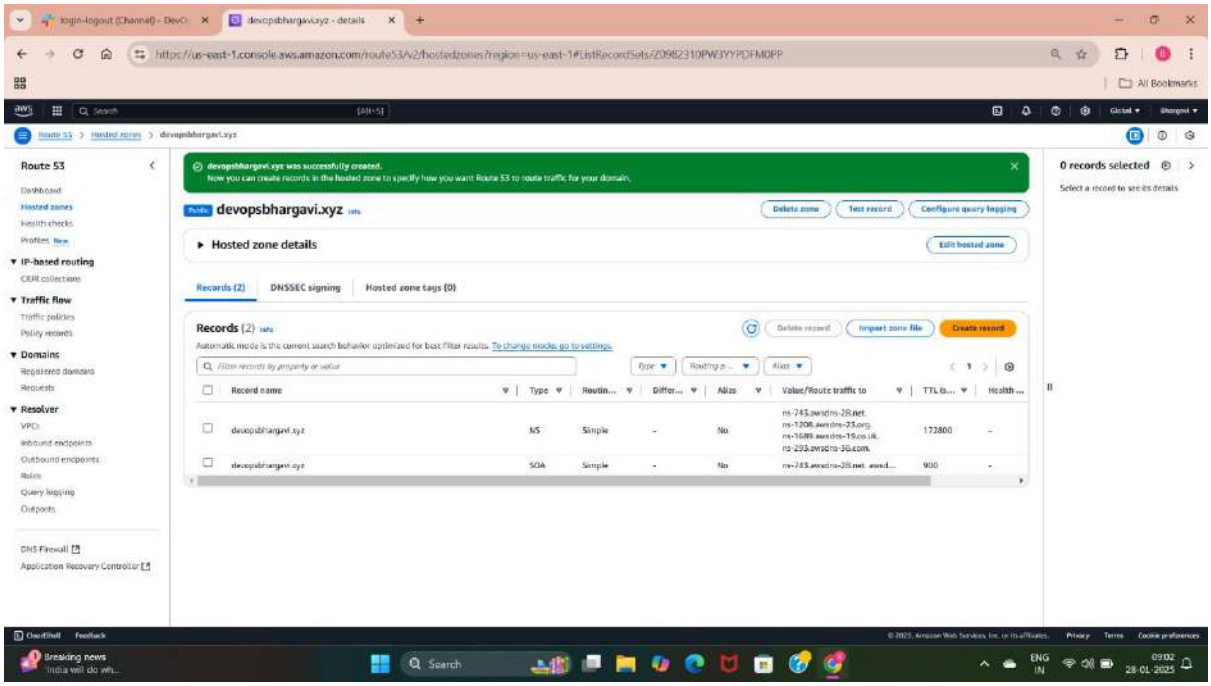


- Click on the "Create Hosted Zone" button.



- Fill in the following details:
- Enter the domain name you want to manage with Route 53 (e.g., devopsbhargavi.xyz).
- Select Public Hosted Zone (used for public-facing websites).
- A Public Hosted Zone allows Route 53 to route internet traffic to your resources.
- Click Create Hosted Zone to complete the setup.

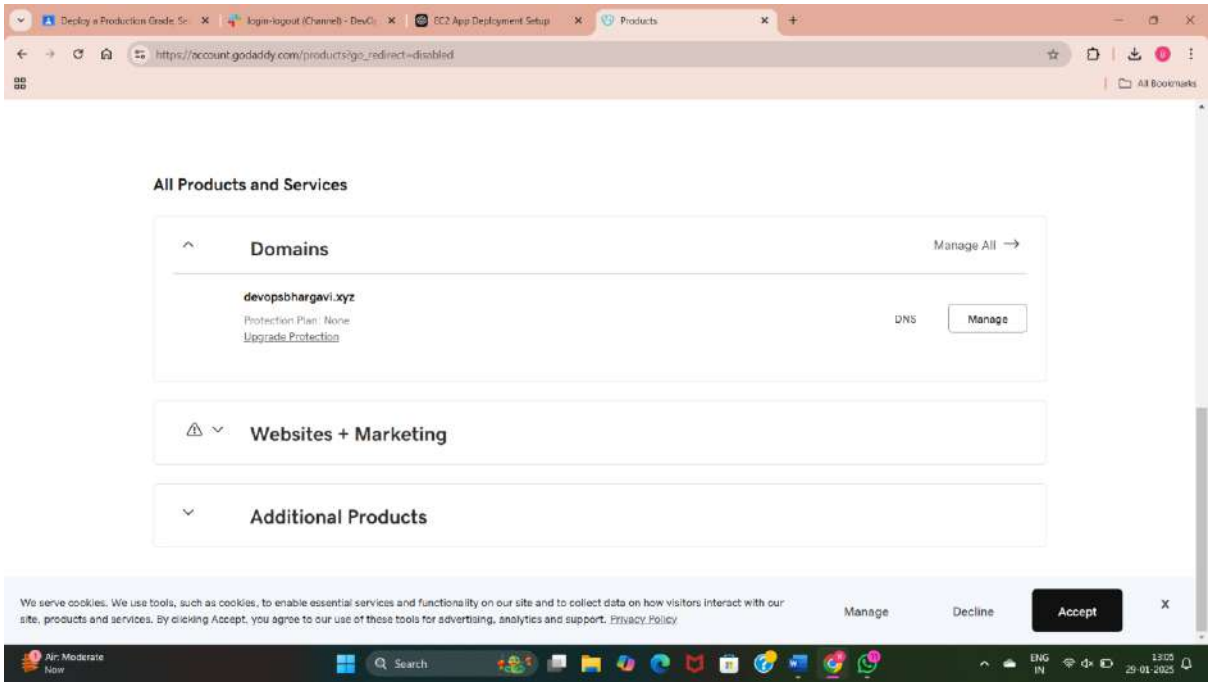
1.1 OBTAIN THE NS RECORDS



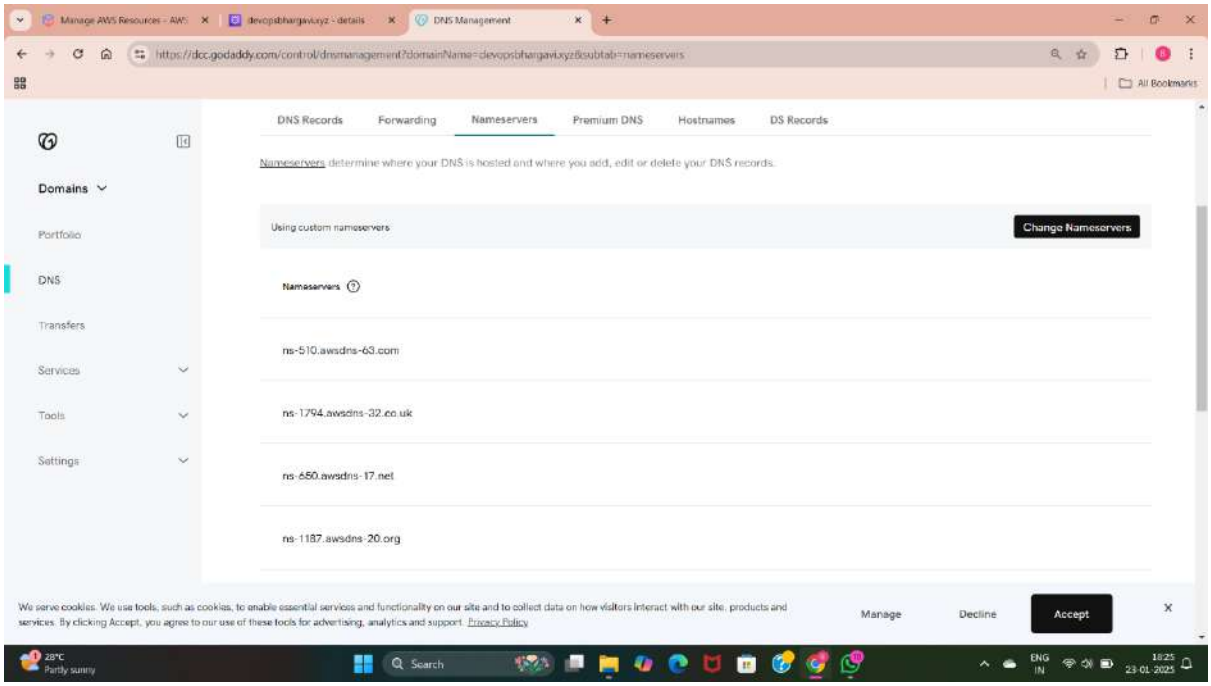
- Once the hosted zone is created, Route 53 will generate a set of Name Server (NS) records.
- These NS records are displayed in the Hosted Zone Details.

1.2 UPDATE NS RECORDS IN YOUR DOMAIN REGISTRAR (E.G., GODADDY)

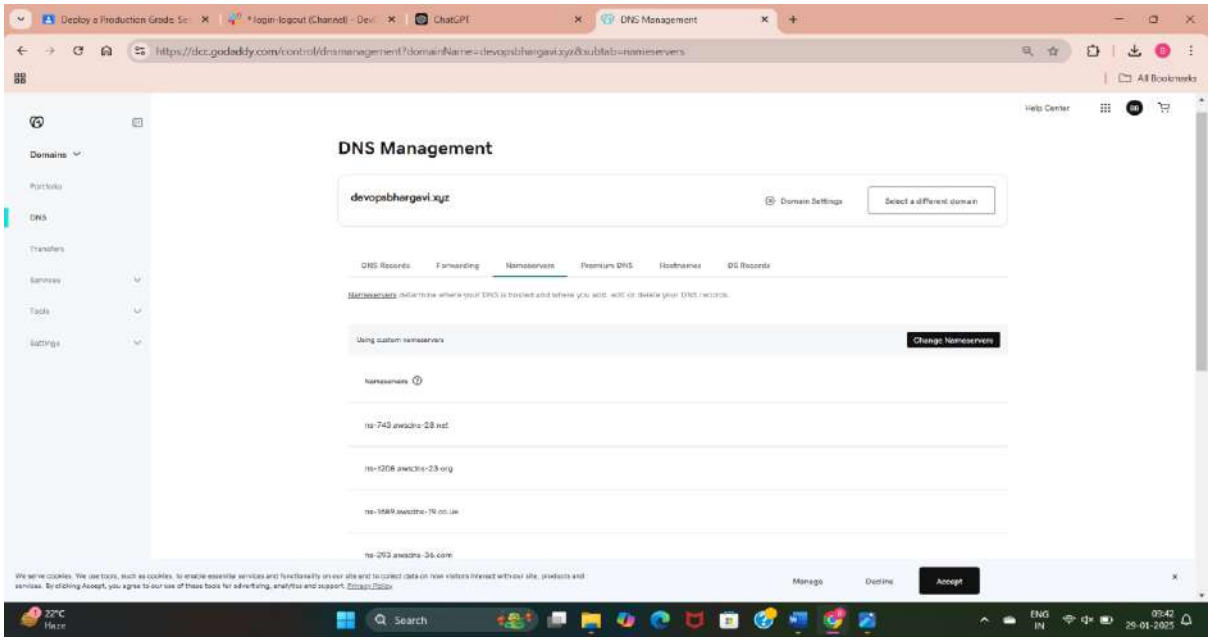
♣ Log in to your domain provider's account (e.g., GoDaddy).



♣ Locate the DNS management section for your domain.



- ♣ Replace the existing NS (Name Server) records with the Route 53 NS records.
- ♣ Delete the old NS records (if any).
- ♣ Add the four NS records provided by Route 53.
- ♣ Save the changes.



Wait for DNS Propagation

- DNS propagation can take up to 24-48 hours, but it often completes within 1-2 hours.
- During propagation, Route 53's DNS records may not yet be accessible globally.

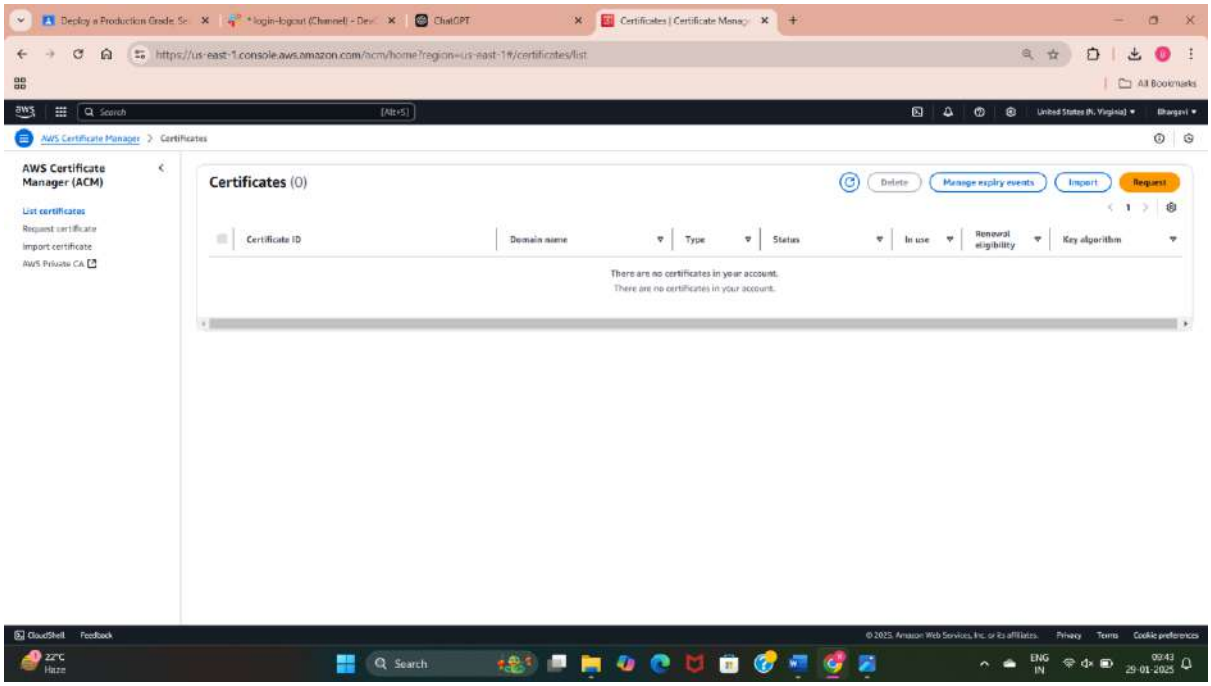
AWS CERTIFICATE MANAGER

AWS Certificate Manager is a service provided by Amazon Web Services (AWS) that allows you to easily provision, manage, and deploy SSL/TLS certificates for securing your websites, applications, and communications.

To request a public SSL certificate using Amazon ACM (AWS Certificate Manager), follow these steps:

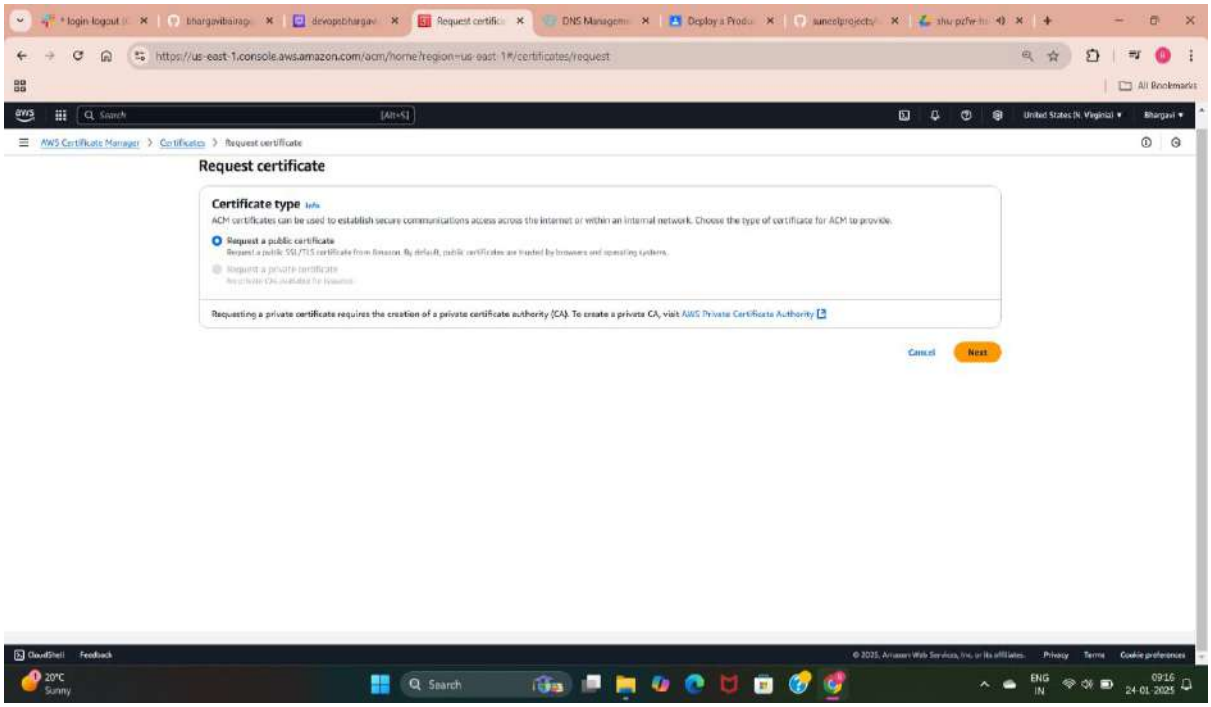
STEP 2: REQUESTING A PUBLIC SSL CERTIFICATE USING AMAZON ACM

- Log in to your AWS Management Console.
- Navigate to Services and search for Certificate Manager under the "Security, Identity, & Compliance" section.

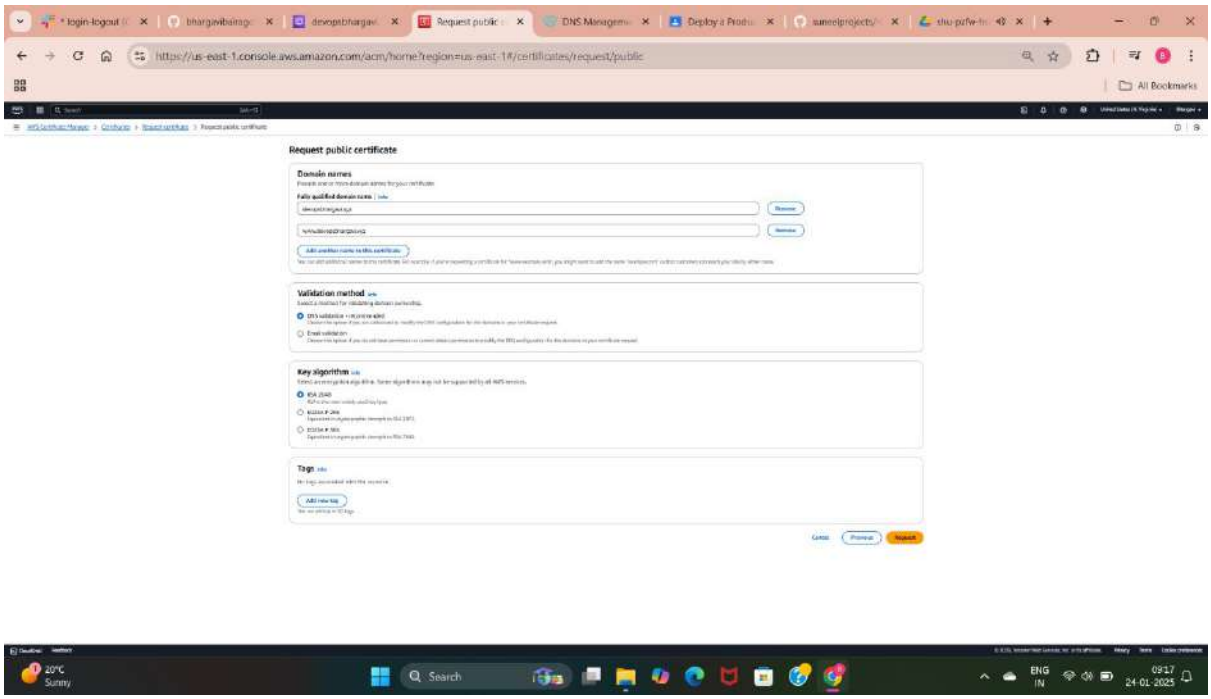


Request a Certificate:

- In the ACM console, click on Request a certificate.

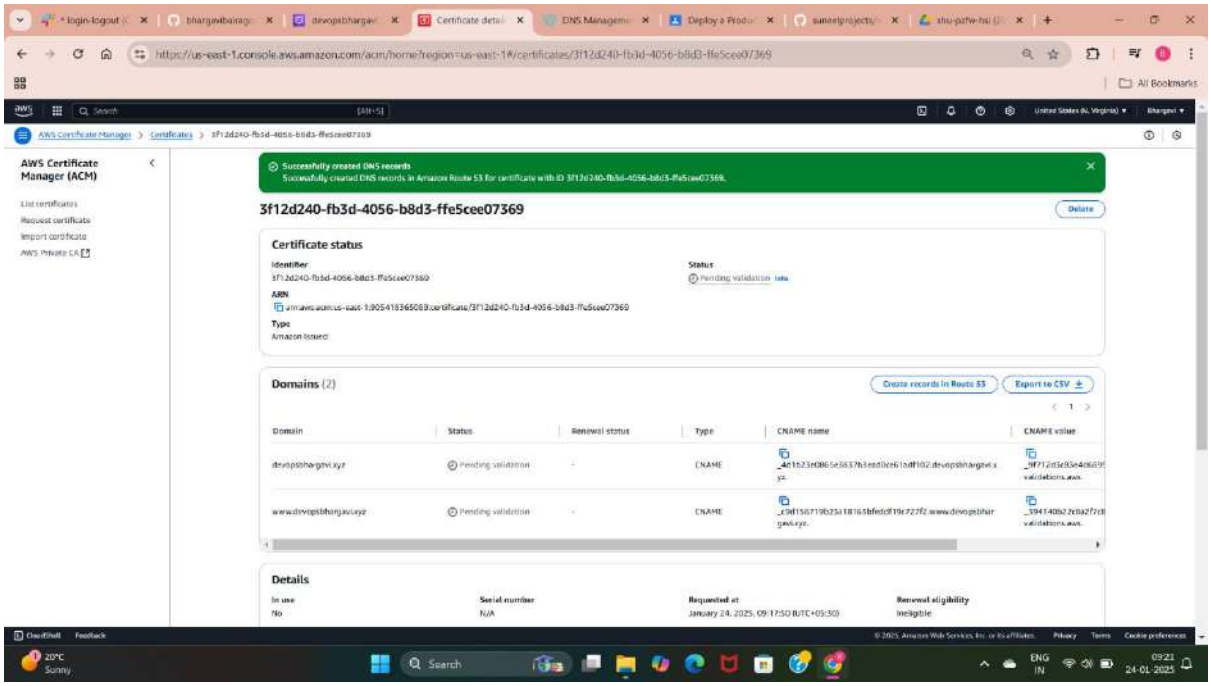


- Choose Request a public certificate.
- Click Next.



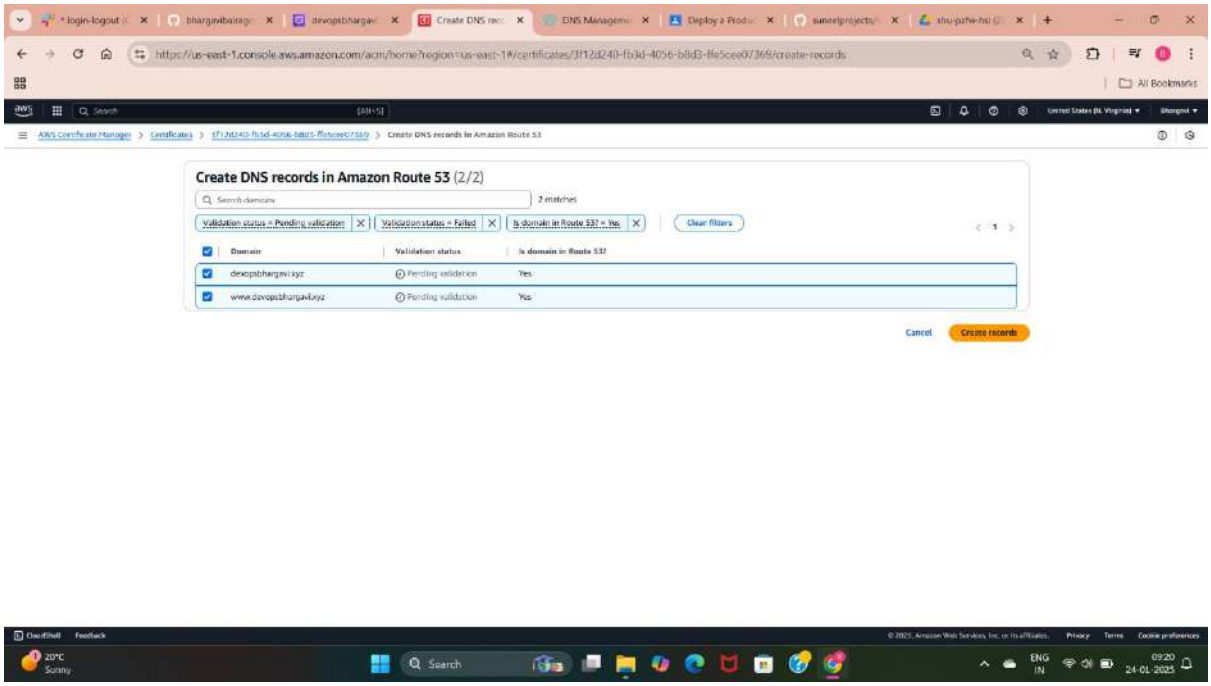
Provide Domain Names:

- Enter the domain name for which you want the SSL certificate (e.g., devopsdost.xyz or www.devopsdost.xyz).
- Select the validation method DNS validation. We need to add a DNS record to prove ownership.
- Review the details of your request and click Confirm and Request.

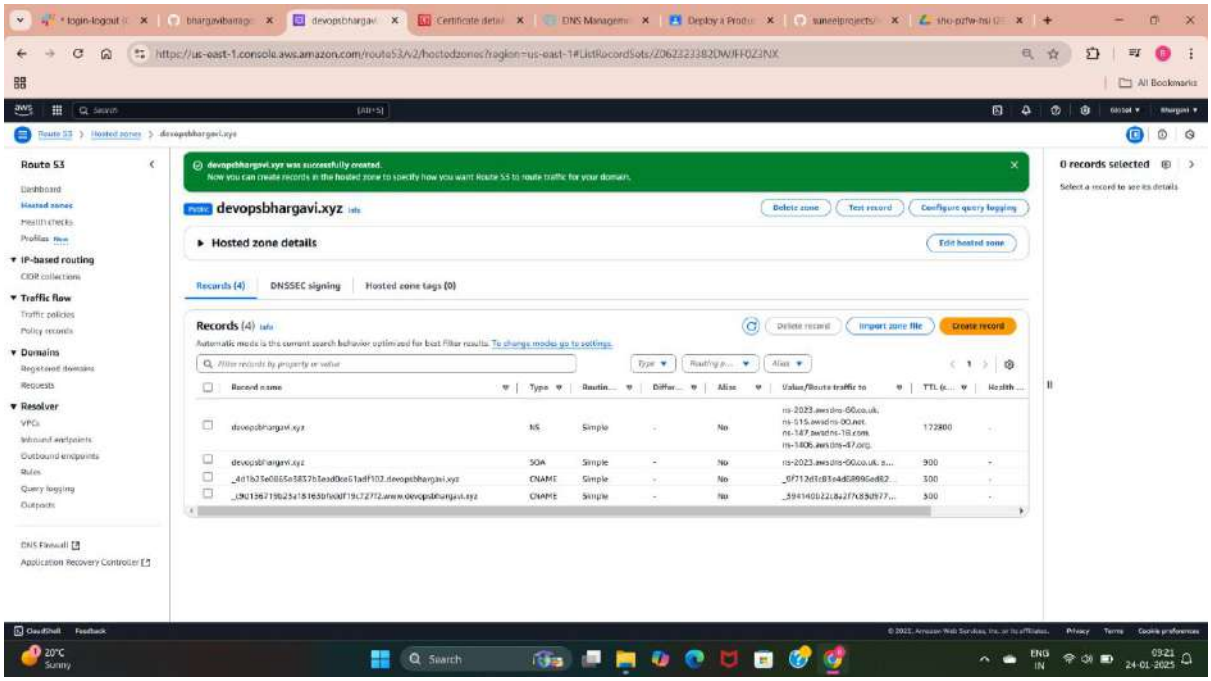


Create DNS Records:

- Once the certificate request is submitted, click on create records in Route 53 (if using Route 53) to automatically add the DNS validation record.



- After completing the validation process, the certificate will be issued and show as Issued in the ACM console.

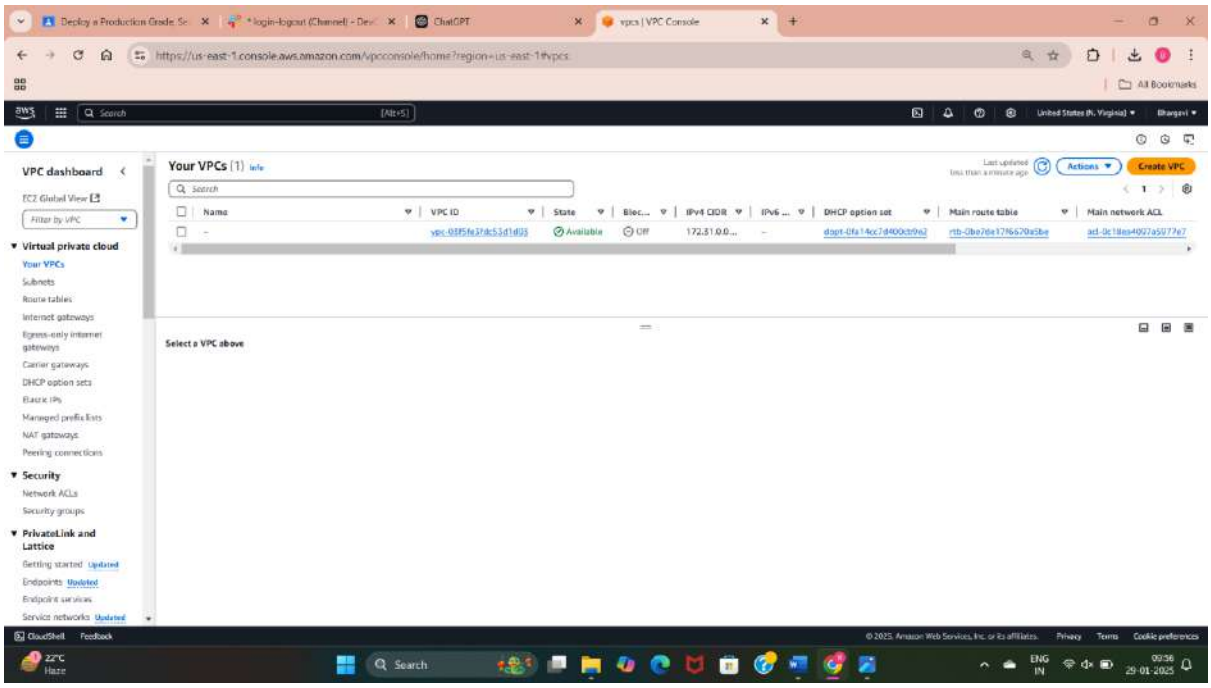


- In hosted zones two new records are added from certificate manager.

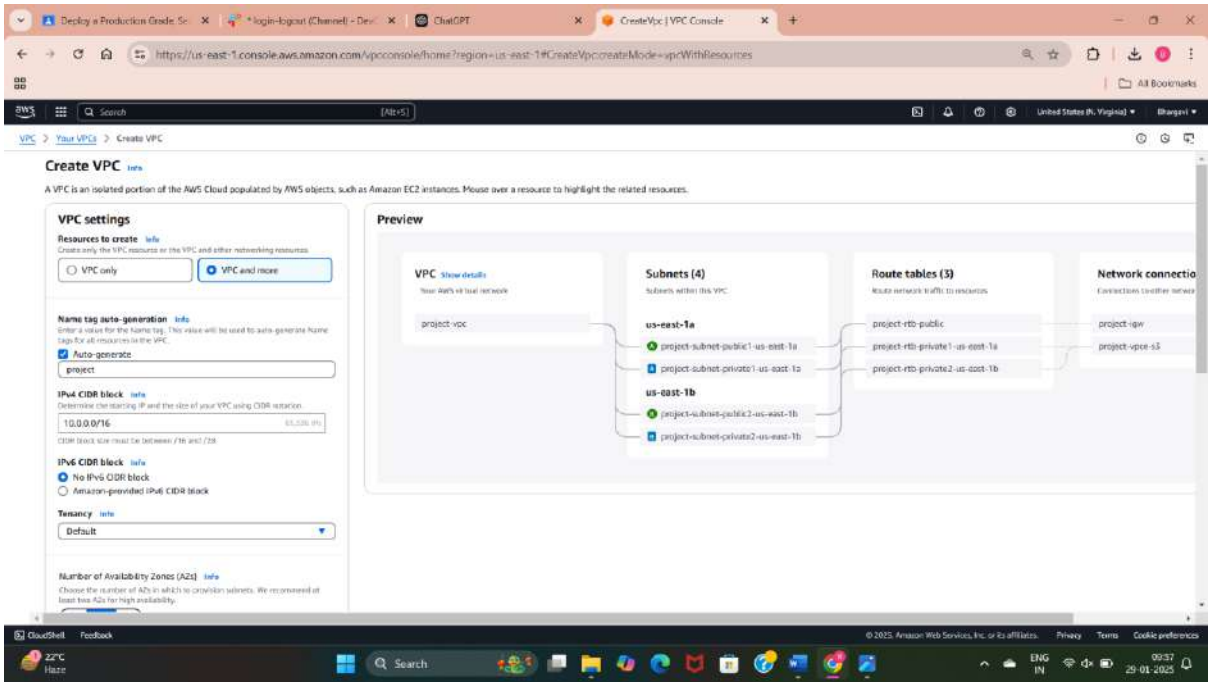
AMAZON VPC (VIRTUAL PRIVATE CLOUD)

Amazon Virtual Private Cloud (VPC) is a service provided by Amazon Web Services (AWS) that enables you to create and control a logically isolated network in the AWS Cloud. With VPC, you can launch AWS resources (like EC2 instances) in a virtual network that you define, giving you full control over the network configuration.

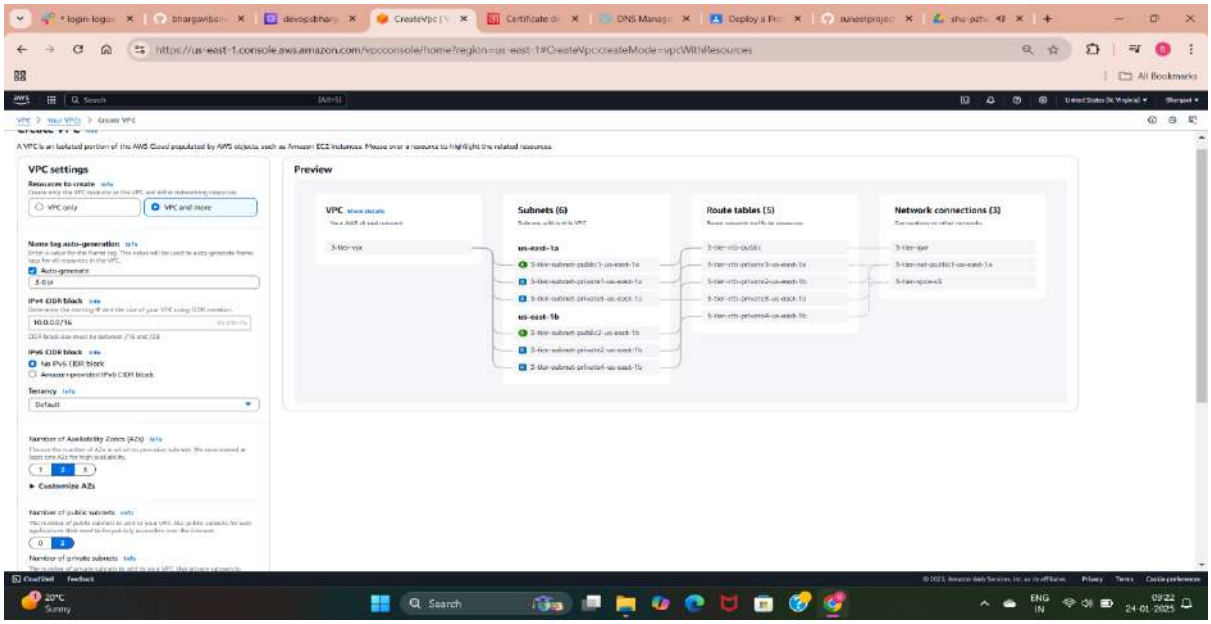
STEP 3: CREATING VPC & SUBNETS



- In the VPC Dashboard, click on Create VPC.



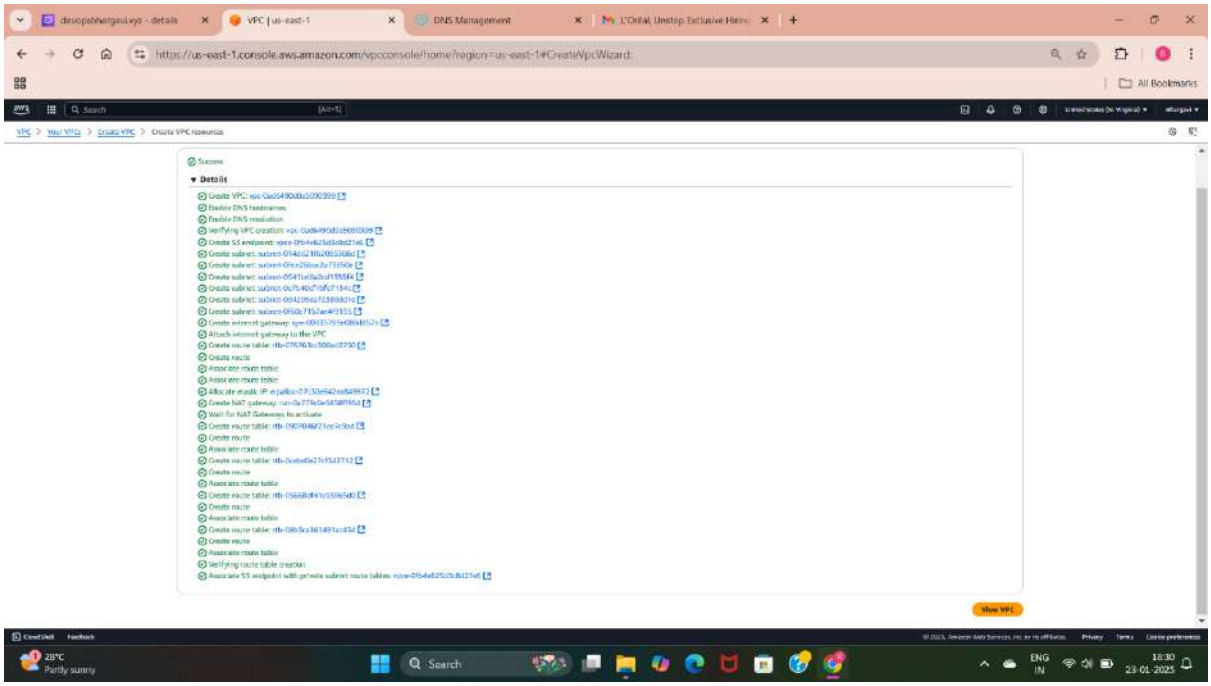
- Select VPC and More.



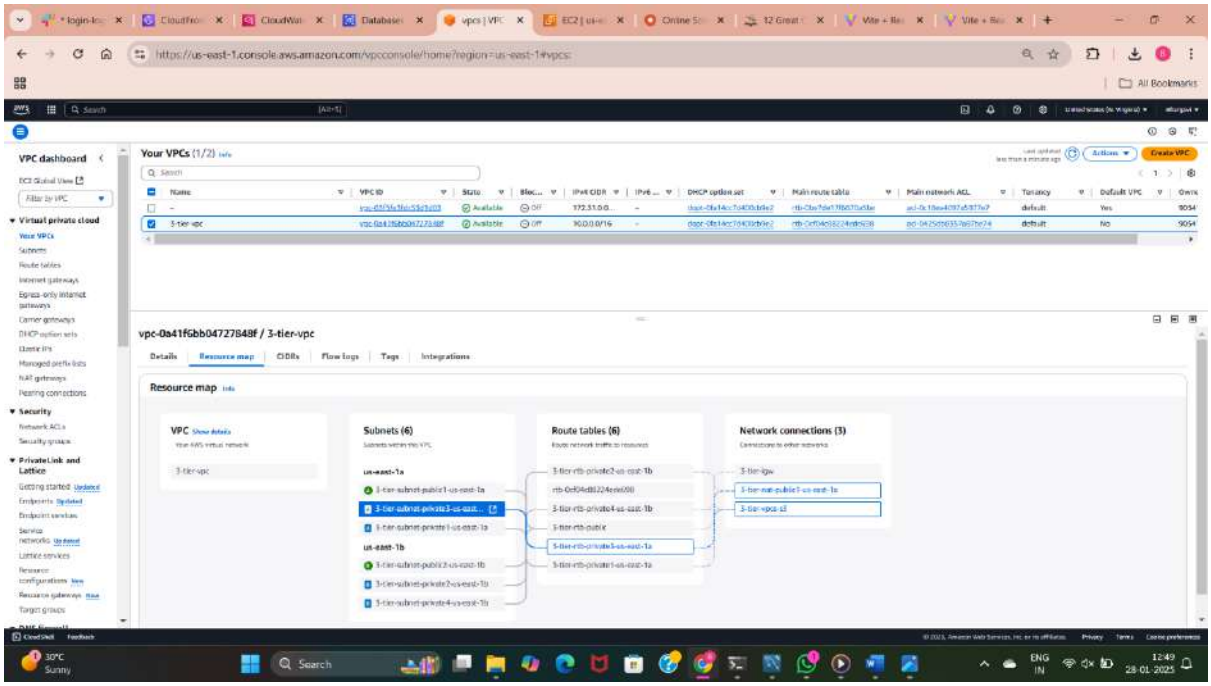
Configure the VPC:

- Name: Enter 3-tier.
- IPv4 CIDR block: Choose an appropriate CIDR block (e.g., 10.0.0.0/16).
- Availability Zones: Choose 2 availability zones for the VPC (select the appropriate zones from the dropdown).
- Public Subnets: Set to 2 subnets (each in different availability zones).
- Private Subnets: Set to 4 subnets.
- NAT Gateway: Choose 1 AZ for the NAT Gateway to provide internet access to the private subnets.

- VPC Endpoints: Set to None for now.
- Once configured, click on Create.
- The creation process will take approximately 2-3 minutes.



- The creation process will take approximately 2-3 minutes.



- The 3-tier VPC is created with subnets and route table.
- In resource map we can check all the resources are properly connected.

1. SUBNET

A subnet is a segment of a VPC's IP address range where you can place resources such as EC2 instances. Subnets allow you to divide your VPC into smaller networks for better management and security.

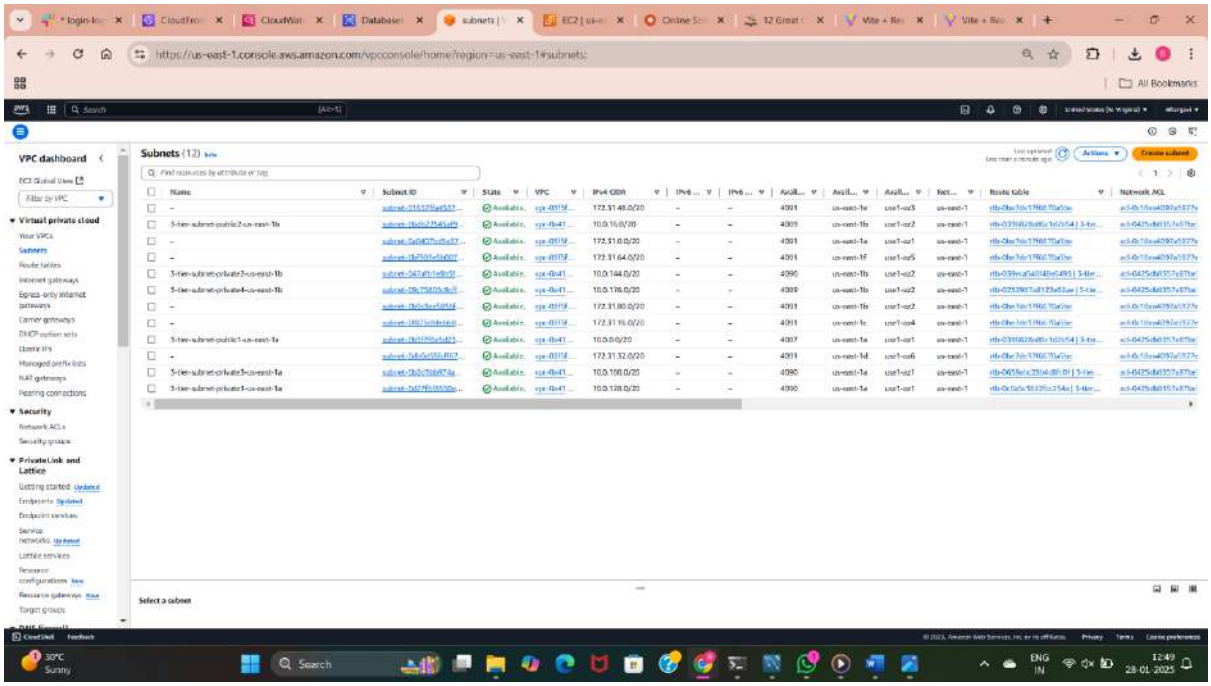
TYPES OF SUBNETS:

Public Subnet:

- Accessible from the internet.
- Associated with a route table that has a route to an Internet Gateway.
- Typically used for web servers or public-facing resources.

Private Subnet:

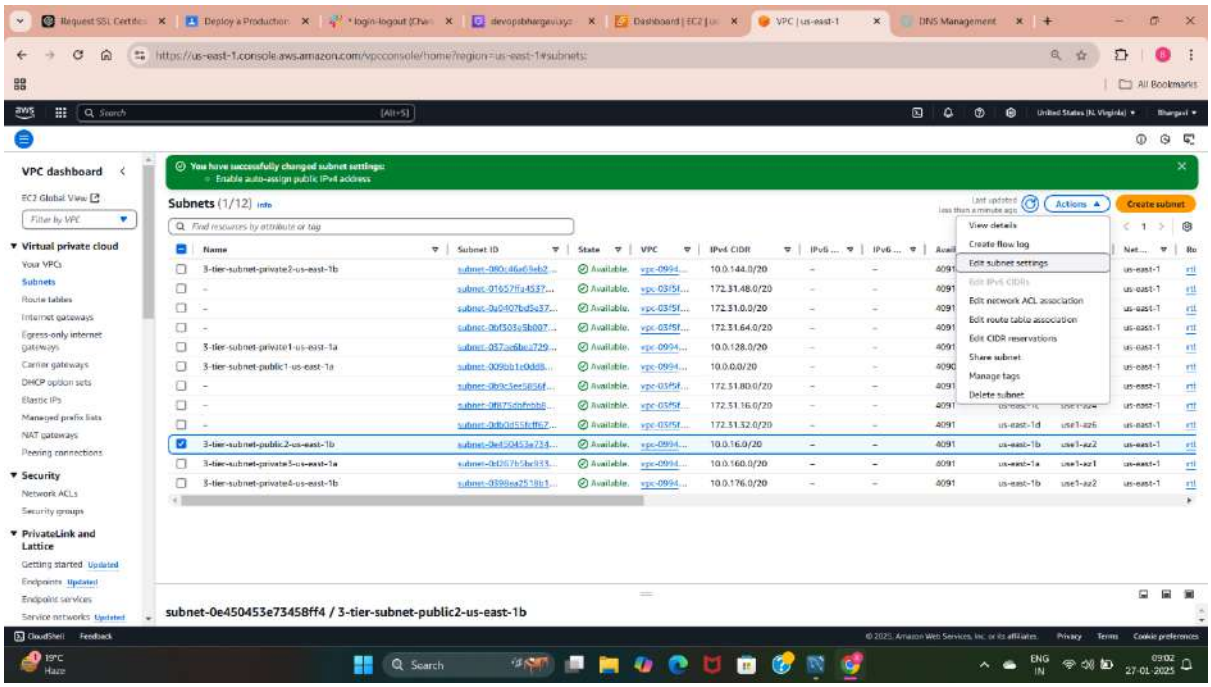
- Not directly accessible from the internet.
- Can access the internet via a NAT Gateway if required.
- Used for databases, application servers, or other internal resources.



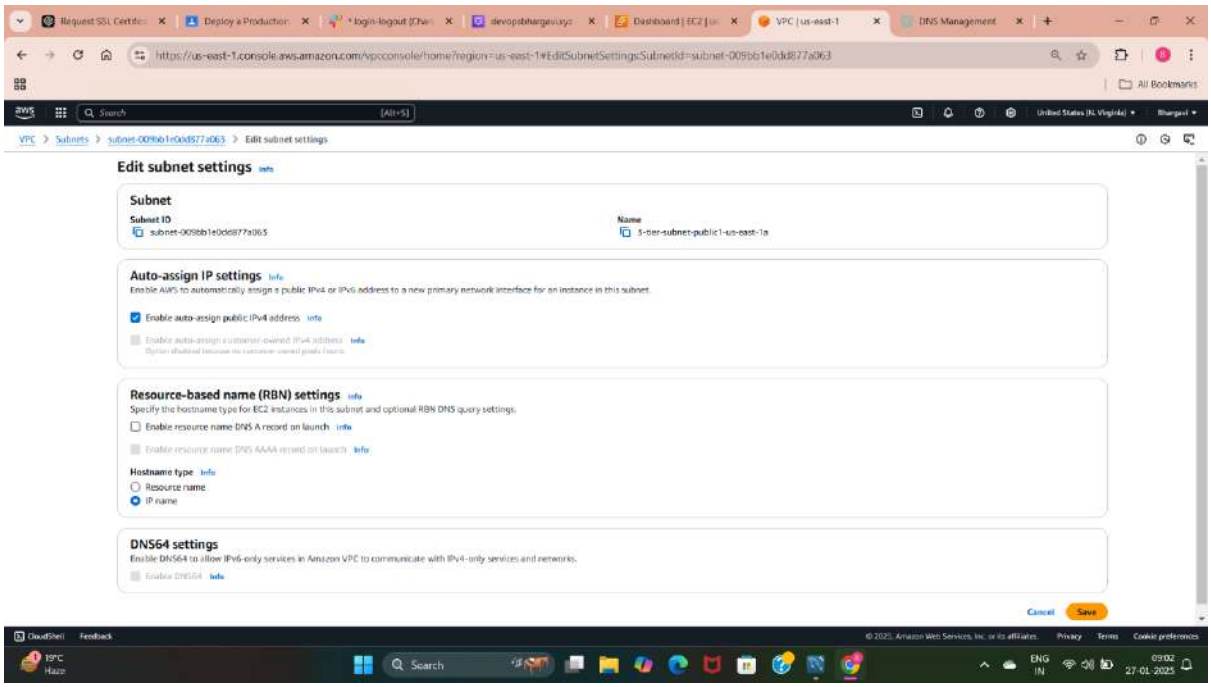
- We can check in VPC service two new public subnets and four new private subnets are created.

3.1 ENABLE PUBLIC IP FOR PUBLIC SUBNETS:

- Once the VPC is created, navigate to the Subnets section under the VPC Dashboard.
- Select the Public Subnets.



- Click on Actions in the top-right corner, and then select Edit Subnet Settings.



- In the Edit Subnet Settings dialog, enable Public IP Automatically by toggling the option.
- Click on Save.

2. ROUTE TABLE

A Route Table contains rules (routes) that define how traffic is directed within the VPC and outside of it.

- Each subnet is associated with one route table.
- Route tables consist of
  Destination: The IP range of the traffic (e.g., 0.0.0.0/0 for all traffic).
  Target: Where to send the traffic (e.g., Internet Gateway, NAT Gateway, or another subnet).

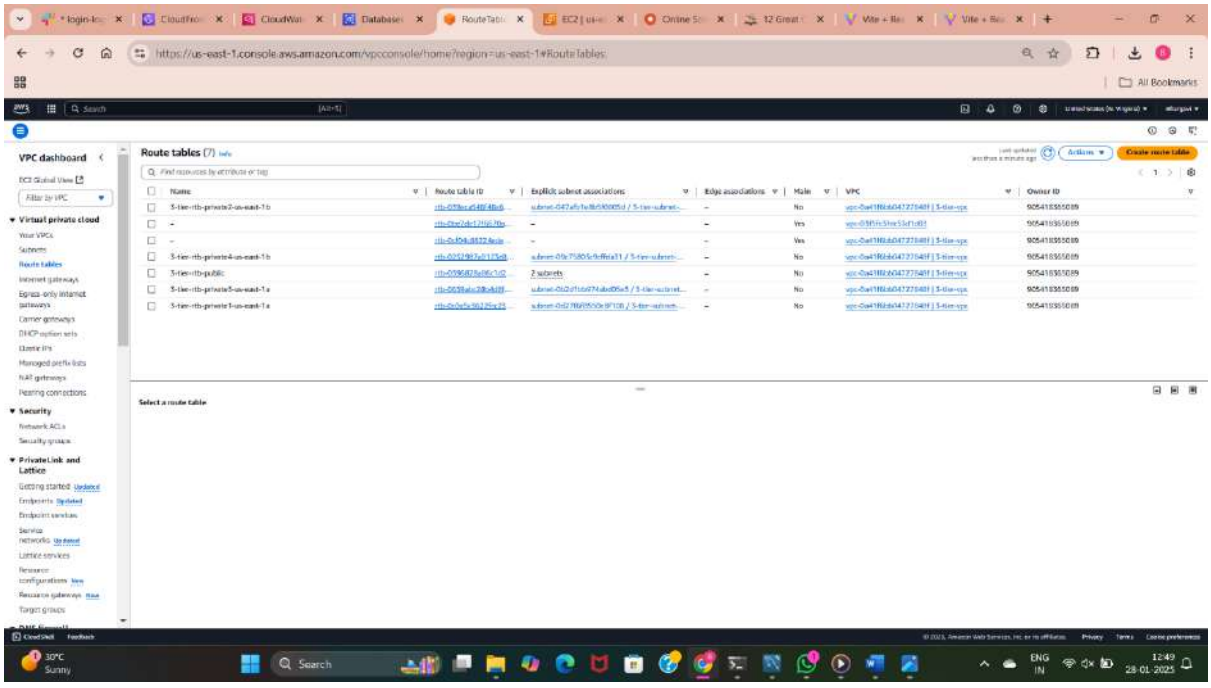
TYPES OF ROUTES:

Public Route Table:

- Contains a route to an Internet Gateway for internet traffic (0.0.0.0/0 → Internet Gateway).
- Associated with public subnets.

Private Route Table:

- Contains a route to a NAT Gateway for internet-bound traffic (0.0.0.0/0 → NAT Gateway).
- Associated with private subnets.

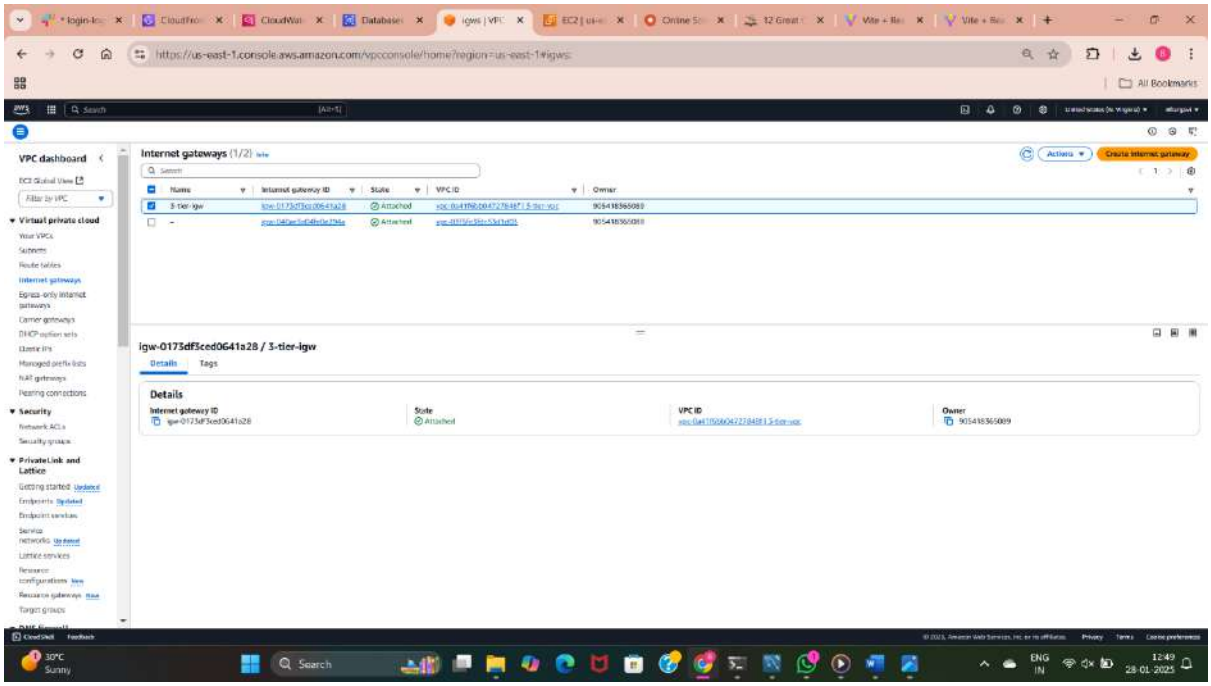


- In route tables four private route tables are created and associated with four private subnets.
- Two public route tables are created and associated with two public subnets.

3. INTERNET GATEWAY (IGW)

An Internet Gateway is a VPC component that allows resources in a public subnet to send and receive traffic from the internet.

- It’s attached to the VPC and supports bidirectional communication with the internet.
- Only instances in subnets with a route to the IGW (via the route table) can access the internet.
- Create and attach an Internet Gateway to your VPC.
- Add a route in the route table for public subnets: 0.0.0.0/0 → Internet Gateway.

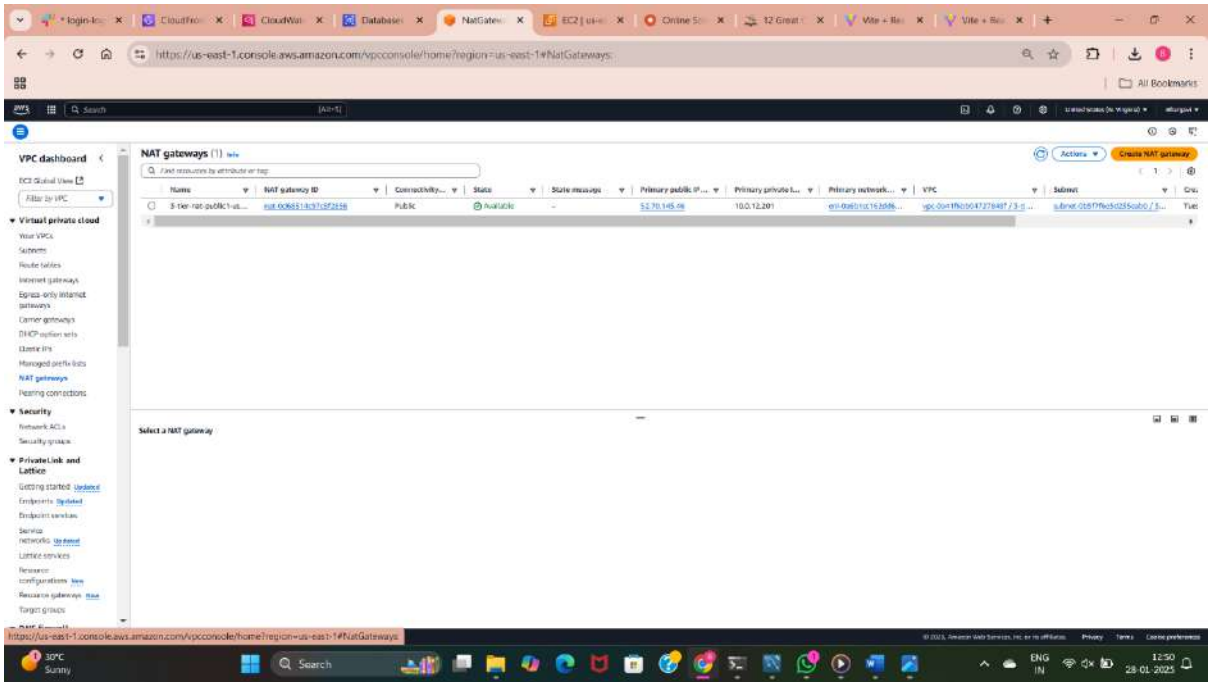


- The internet gateway is created and attached to vpc. This is used to provide internet for public subnets which are created using public subnet.

4. NAT GATEWAY

A NAT (Network Address Translation) Gateway allows instances in a private subnet to access the internet while preventing inbound traffic from the internet.

- Used for outbound internet traffic from private subnets.
- Provides security by keeping resources in private subnets inaccessible from the internet.
- A database server in a private subnet needs to download software updates or send logs to an external system. The NAT Gateway enables this without exposing the server to the internet.

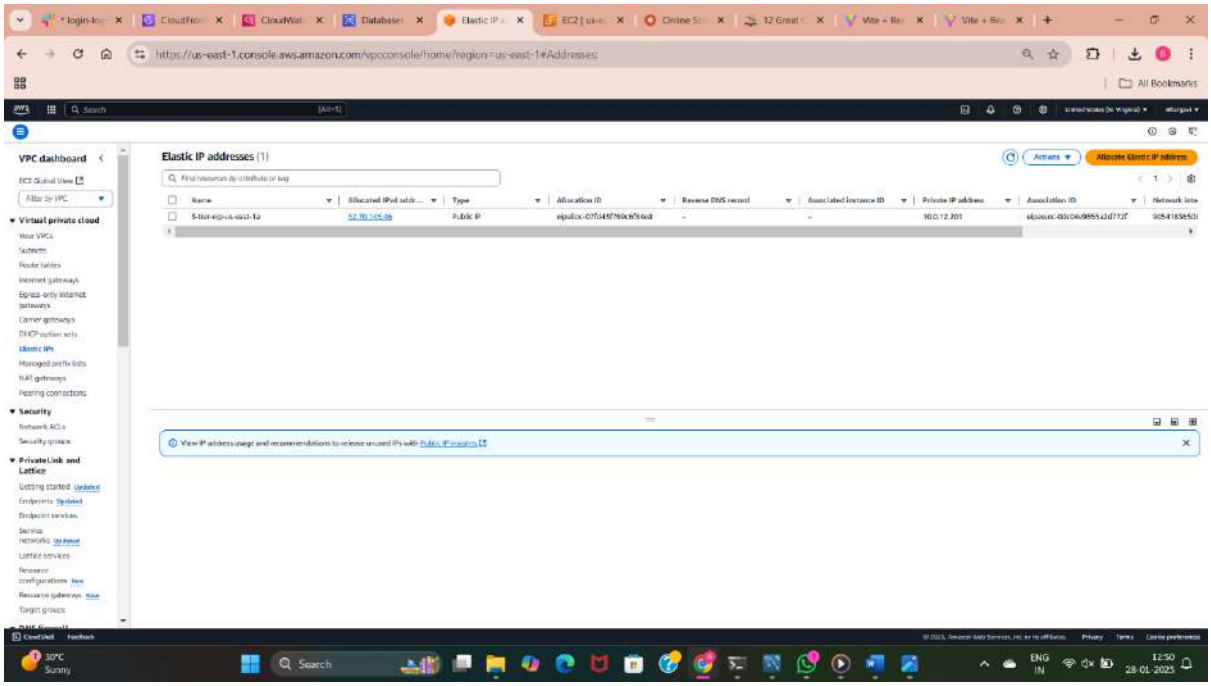


- NAT Gateway is created to provide internet for private subnets and private instances created using private subnets.

5.ELASTIC IP

An Elastic IP (EIP) in AWS is a static, public IPv4 address that you can allocate and associate with your AWS resources, such as EC2 instances, NAT Gateways, or Network Load Balancers. It is designed to provide consistent and reliable access to your resources, even if their private IP address or instance changes.

- Unlike dynamic public IPs assigned to EC2 instances, Elastic IPs remain constant even if you stop and restart an instance.



- One Elastic ip is created to attach to public instnaces.

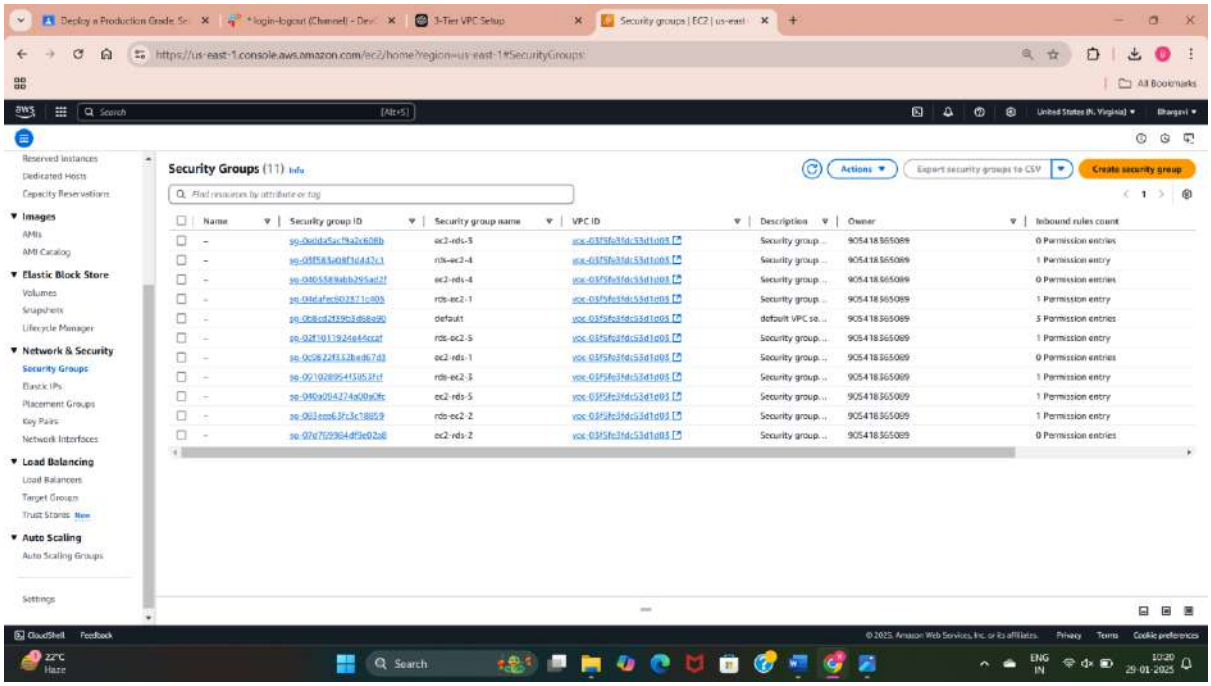
SECURITY GROUPS

A Security Group in AWS acts as a virtual firewall for your Amazon EC2 instances or other resources. It controls inbound and outbound traffic at the instance level, ensuring only authorized network traffic is allowed.

- If you allow inbound traffic, the corresponding outbound traffic is automatically allowed (and vice versa). For example, if you allow an inbound SSH connection, the outbound response is permitted without explicitly specifying it.

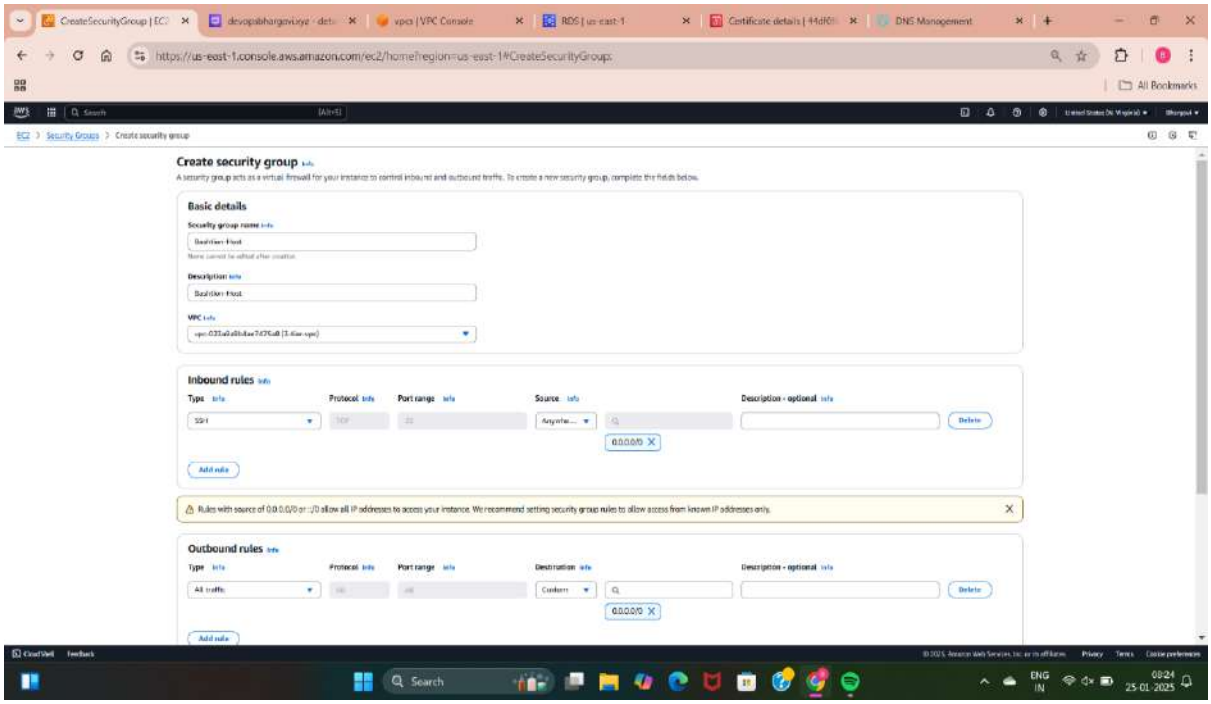
STEP 4: CREATING SECURITY GROUPS

✓ In the EC2 Dashboard, on the left-hand side, scroll down and select Security Groups under the Network & Security section.



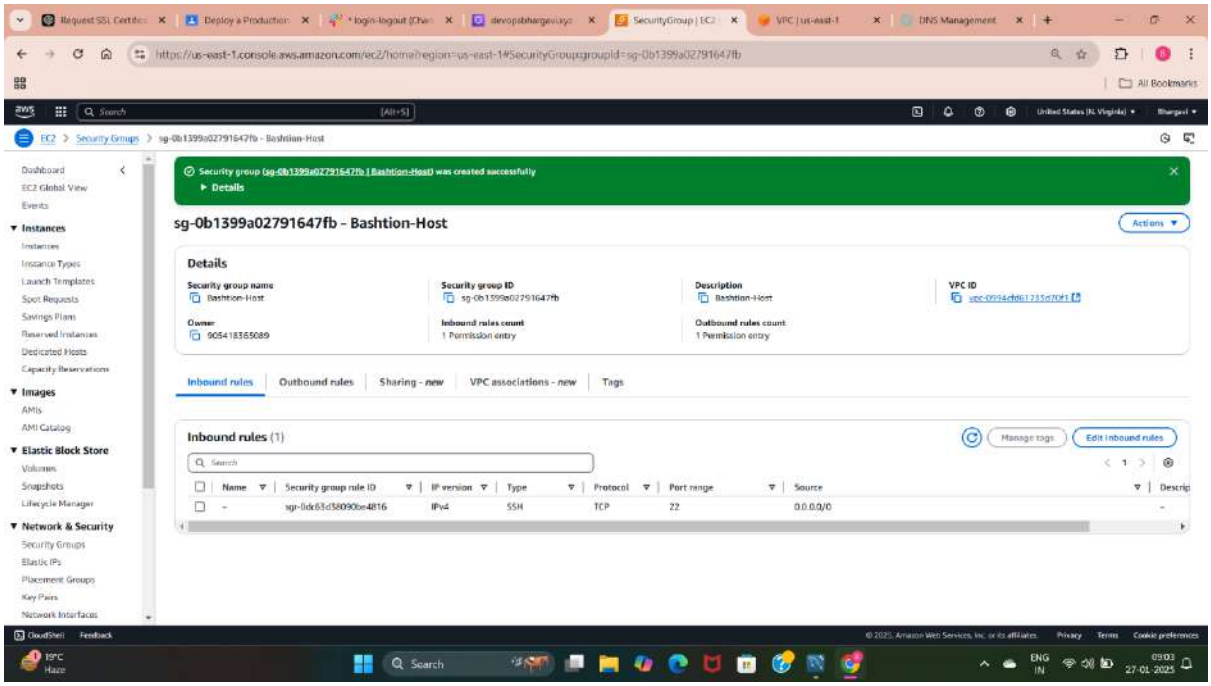
✓ Click on Create Security Group at the top of the page.

4.1: CREATING SECURITY GROUP FOR BASTION HOST



♣ Enter Bastion-Host as the name of the security group.
♣ Select the VPC you created earlier, which is 3-Tier-Architecture.
♣ Under the Inbound rules section, click on Add Rule.
♣ Select Type as SSH from the dropdown. This will automatically set the Port Range to 22.

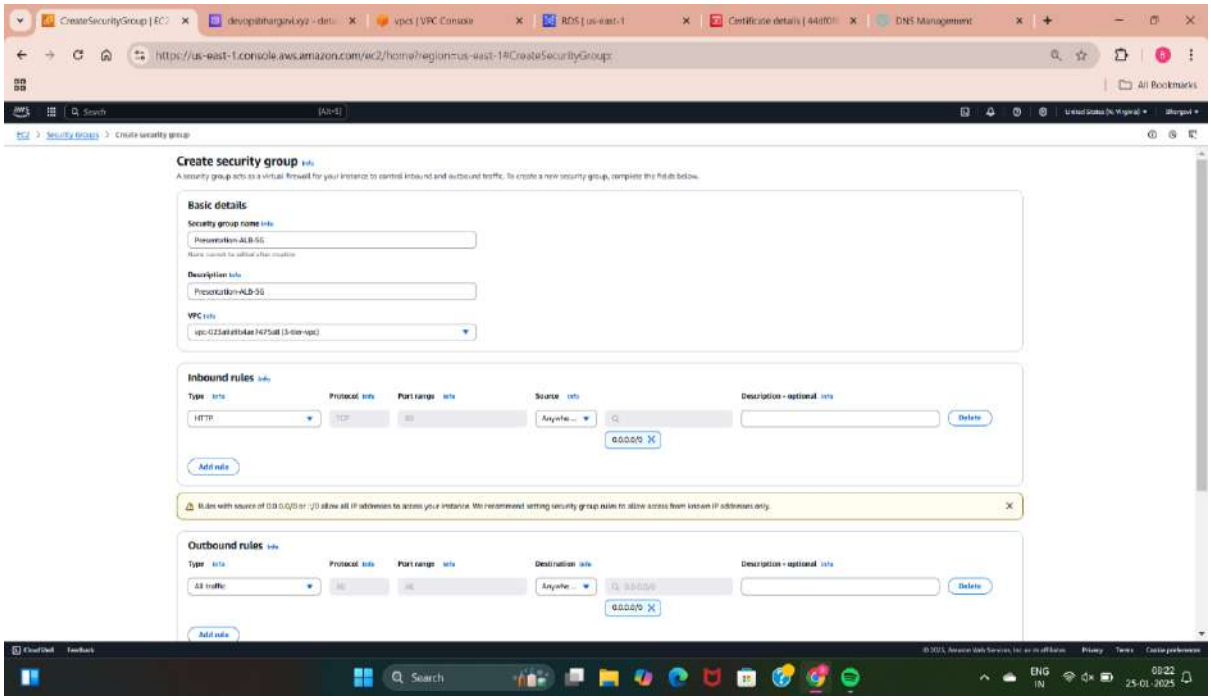
- For Source, select Anywhere-IPv4 (0.0.0.0/0) to allow SSH access from any IPv4 address. This is generally used for Bastion Hosts to provide SSH access from anywhere. You can restrict access further by specifying a particular IP range if necessary.
- Click Create Security Group.



- The security group for Bastion-Host is created.

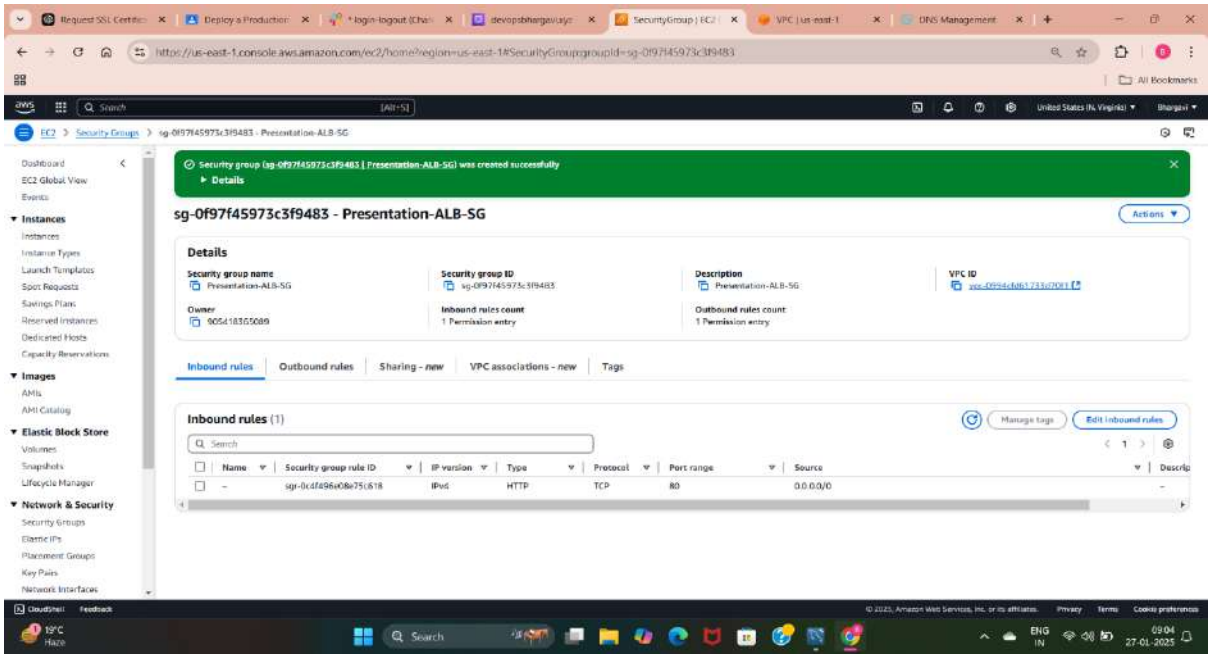
4.2: CREATE A SECURITY GROUP FOR PRESENTATION TIER ALB

- Click on Create Security Group at the top of the page.



- Enter Presentation-ALB-SG as the name of the security group.
- Select the VPC you created earlier, which is 3-tier.

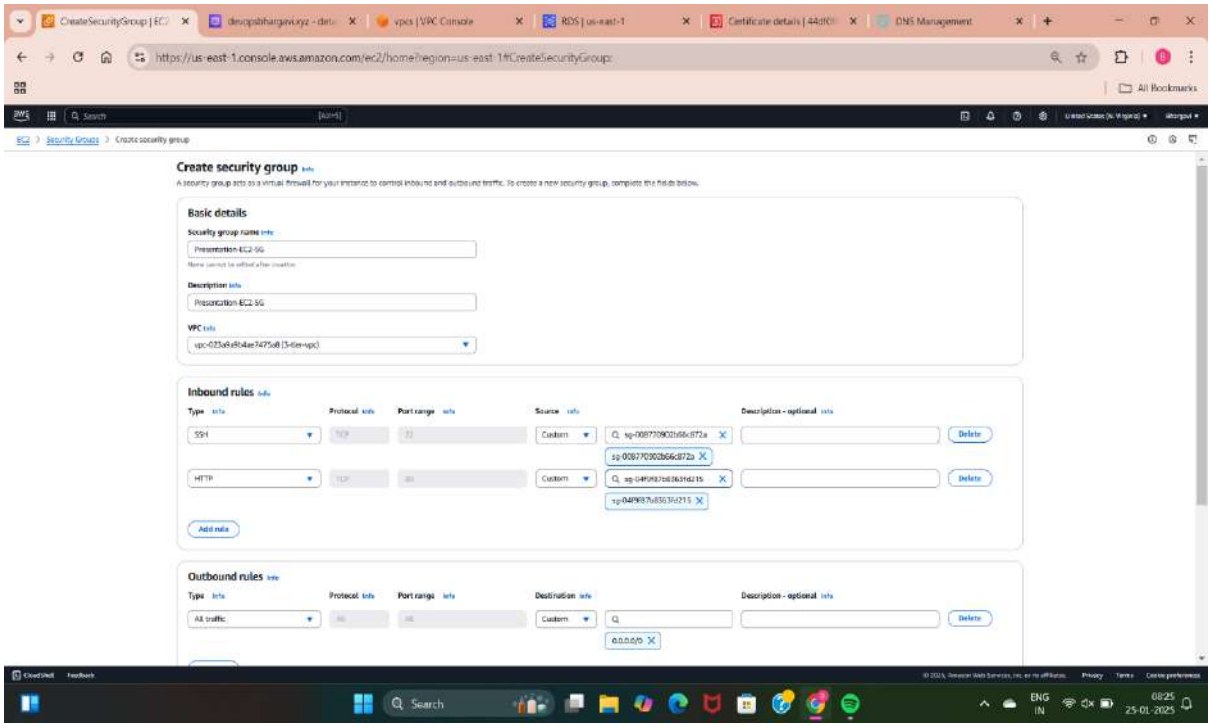
- ♣ Under the Inbound rules section, click on Add Rule, select HTTP from the dropdown. This will automatically set the Port Range to 80. Select Anywhere-IPv4 (0.0.0.0/0) to allow HTTP access from any IPv4 address. This is typical for an ALB that handles web traffic from the internet.
- ♣ Click Create Security Group.



- ♣ The Presentation-ALB-SG security group is created.

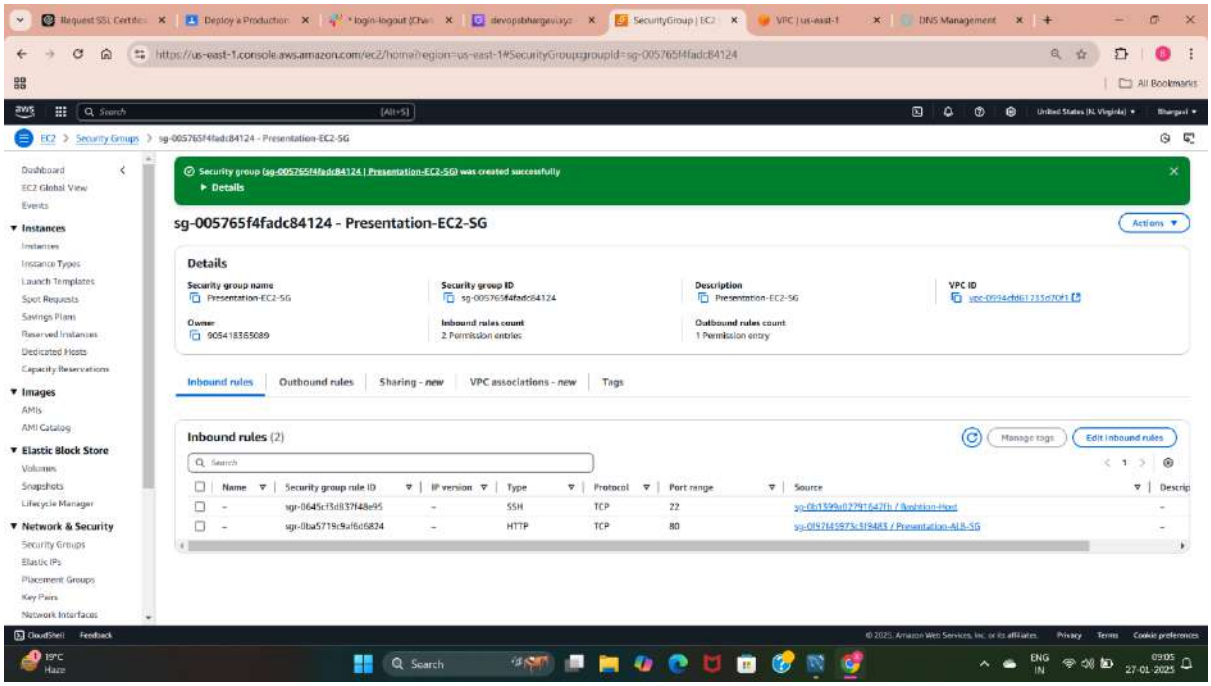
4.3: CREATING SECURITY GROUP FOR PRESENTATION TIER EC2

Click on Create Security Group at the top of the page.



- ♣ Enter Presentation-EC2-SG as the name of the security group.

- Select the VPC you created earlier, which is 3-Tier-Architecture.
- Under the Inbound rules section, click on Add Rule. Select SSH from the dropdown (this will automatically set the Port Range to 22). Select Custom and choose the Bastion-Host security group to restrict SSH access to only the Bastion Host.
- Click on Add Rule again.
- Select HTTP from the dropdown (this will automatically set the Port Range to 80). Select Custom and choose the Presentation-Tier-ALB security group to allow HTTP traffic from the ALB only.
- Once added both inbound rules, click Create Security Group.

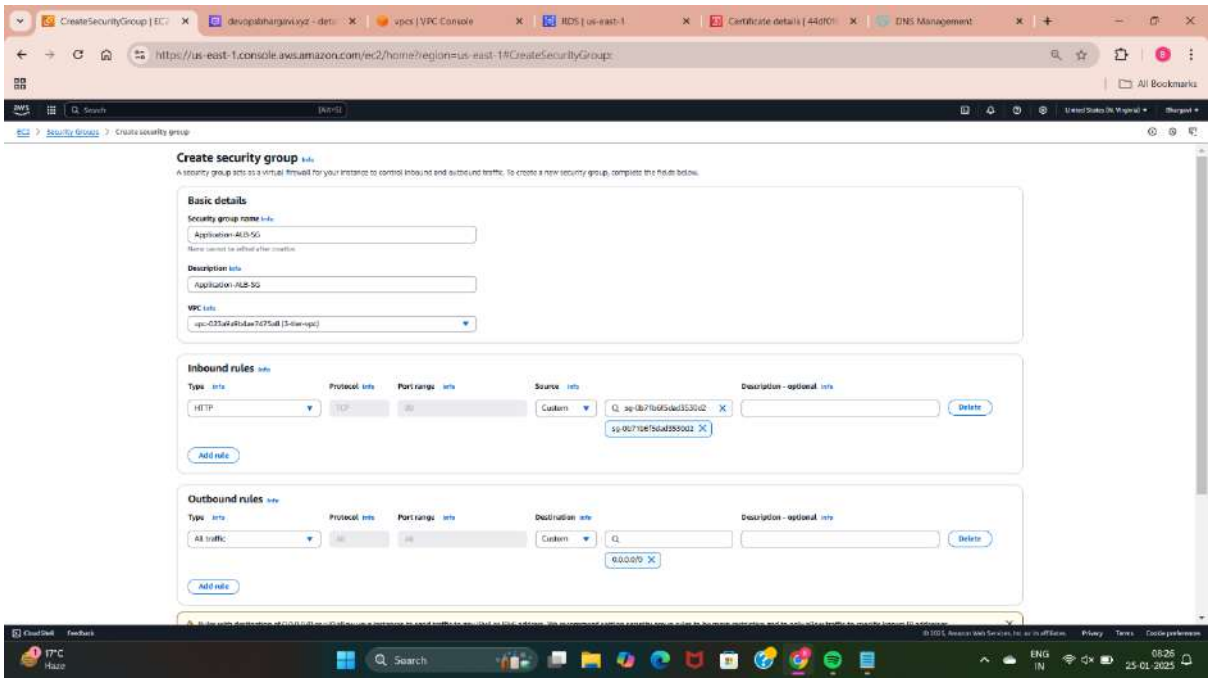


This security group will allow:

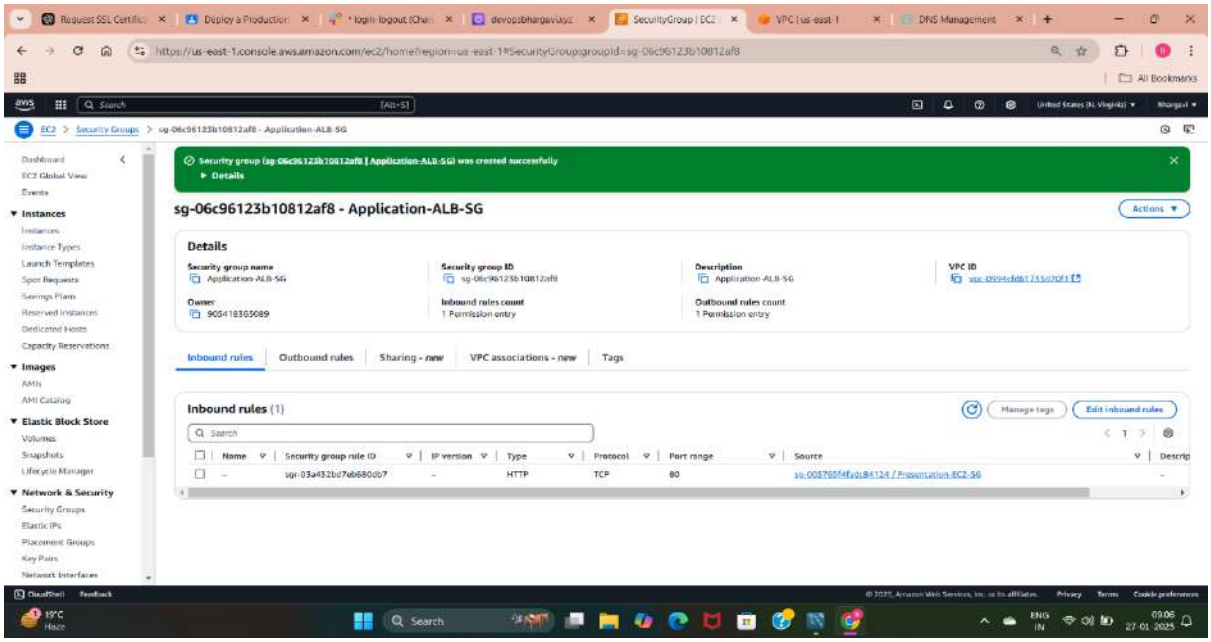
- SSH access from the Bastion Host (for management purposes).
- HTTP traffic from the Presentation-Tier-ALB (so the ALB can forward requests to the EC2 instances).

4.4: CREATING SECURITY GROUP FOR APPLICATION TIER ALB

Click on Create Security Group at the top of the page.

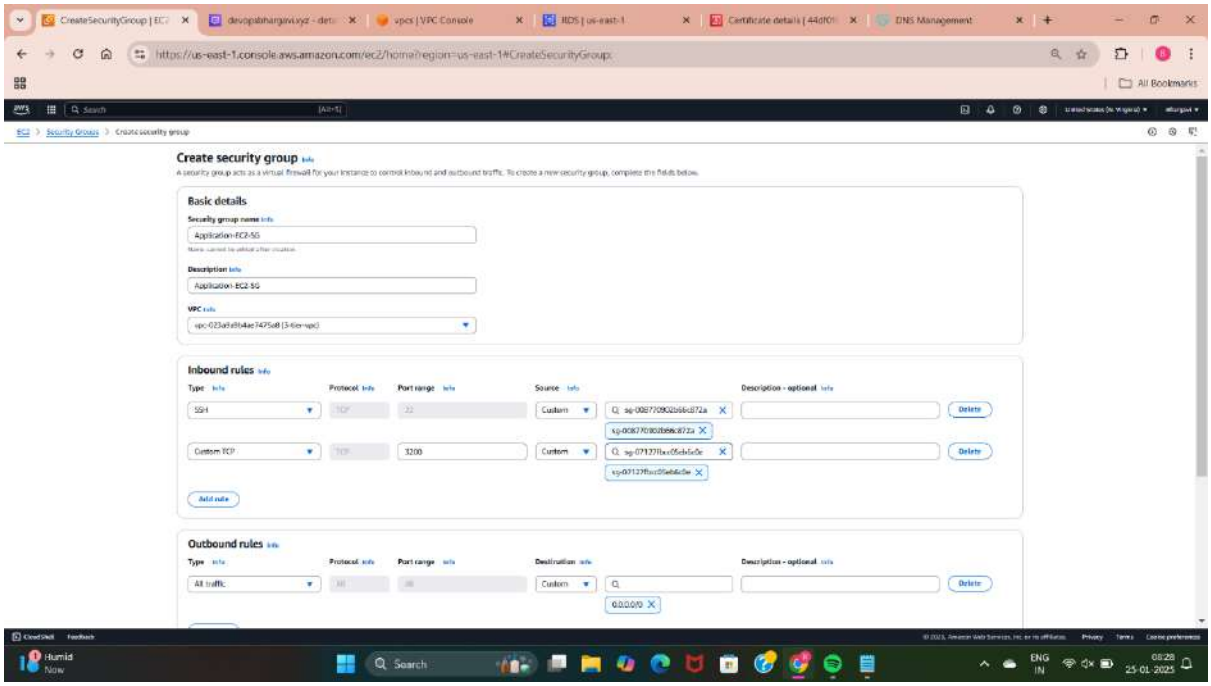


- Enter Application-ALB-SG as the name of the security group.
- Select the VPC you created earlier, which is 3-Tier-Architecture.
- Select HTTP from the dropdown (this will automatically set the Port Range to 80). Select Custom and choose the Presentation-Tier-EC2 security group to allow HTTP traffic only from the Presentation Tier EC2 instances.
- Click Create Security Group



- It is allowing HTTP traffic from the Presentation-Tier-EC2 security group (so the Application Tier ALB can communicate with the EC2 instances in the Application Tier).

4.5: CREATE SECURITY GROUP FOR APPLICATION TIER EC2



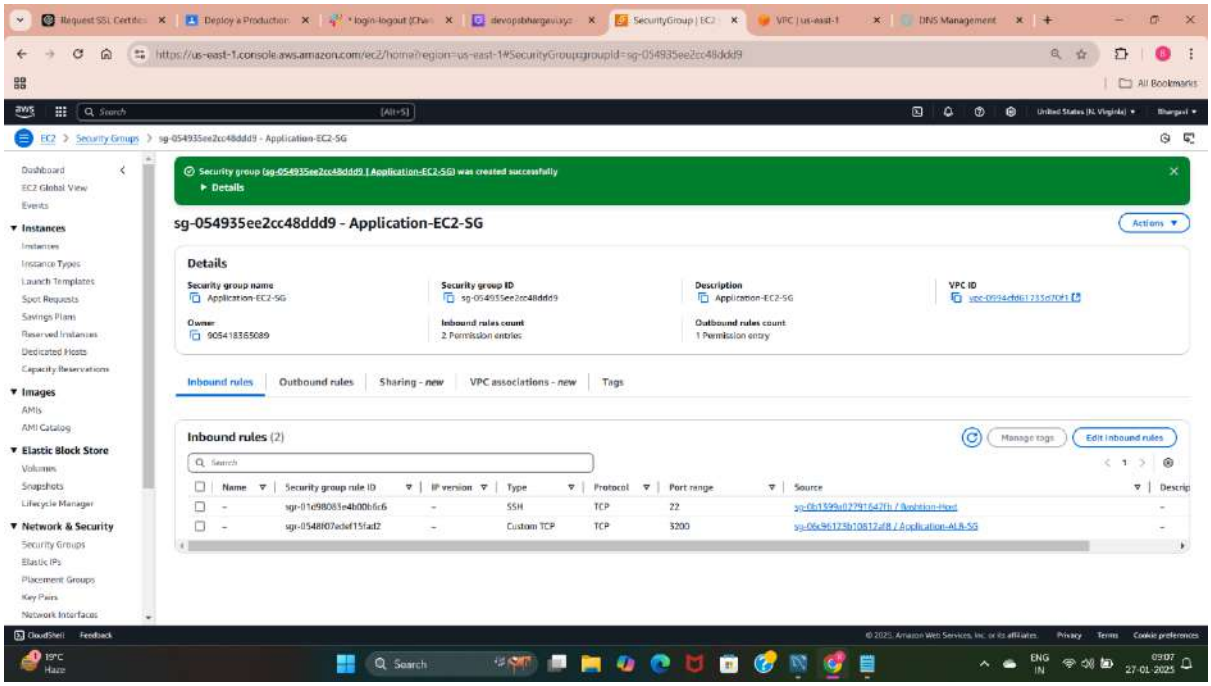
- ♣ Enter Application-EC2-SG as the name of the security group.
- ♣ Select the VPC you created earlier, which is 3-Tier-Architecture

Rule 1: SSH from Bastion Host

- ♣ Under the Inbound rules section, click on Add Rule.
- ♣ Select SSH from the dropdown (this will automatically set the Port Range to 22).
- ♣ Source:Select Custom and choose the Bastion-Host security group to restrict SSH access to only the Bastion Host.

Rule 2: Custom TCP for 3200 from App-Tier-ALB

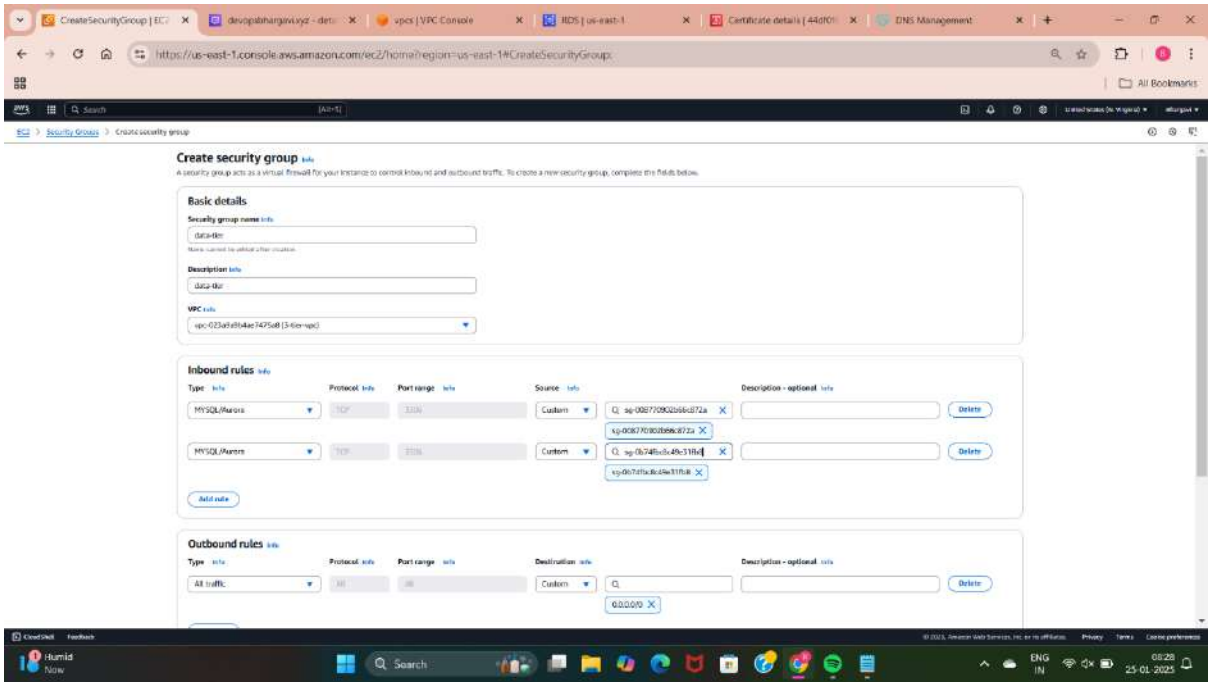
- ♣ Click on Add Rule again.
- ♣ Type: Select Custom TCP from the dropdown.
- ♣ Port Range: Enter 3200 as the port number.
- ♣ Source: Select Custom and choose the Application-Tier-ALB security group to allow traffic on port 3200 from the ALB only.
- ♣ Click Create Security Group.



This security group will allow:

- SSH access from the Bastion Host (for management purposes).
- Custom TCP traffic on port 3200 from the Application-Tier-ALB (for communication between the ALB and EC2 instances in the Application Tier).

4.6: CREATE SECURITY GROUP FOR DATA TIER



- Enter Data-Tier as the name of the security group.
- Select the VPC you created earlier, which is 3-Tier-Architecture.

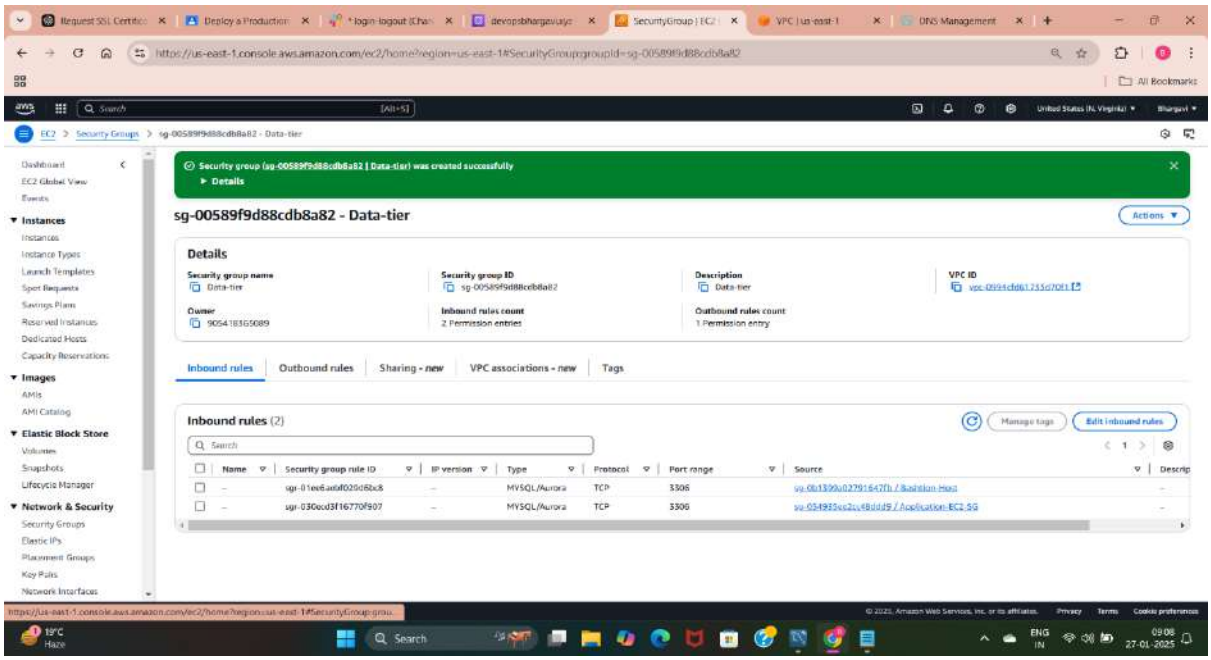
Rule 1: MySQL/Aurora from Bastion Host

- Under the Inbound rules section, click on Add Rule.

- ♣ Type: Select MySQL/Aurora from the dropdown (this will automatically set the Port Range to 3306).
- ♣ Source: Select Custom and choose the Bastion-Host security group to allow MySQL traffic from the Bastion Host.

Rule 2: MySQL/Aurora from Application-Tier-EC2

- ♣ Click on Add Rule again.
- ♣ Type: Select MySQL/Aurora from the dropdown (this will automatically set the Port Range to 3306).
- ♣ Source: Select Custom and choose the Application-Tier-EC2 security group to allow MySQL traffic from the EC2 instances in the Application Tier.
- ♣ Click Create Security Group.



This security group will allow:

- ♣ MySQL/Aurora traffic on port 3306 from the Bastion Host (for administrative access).
- ♣ MySQL/Aurora traffic on port 3306 from the Application-Tier-EC2 security group (for communication between the EC2 instances in the Application Tier and the Data Tier).

BASTION HOST:

A Bastion Host is a special-purpose EC2 instance in a public subnet that provides secure access to instances in a private subnet via SSH or RDP. It acts as a jump server to prevent exposing private instances directly to the internet.

Secure Access to Private Instances

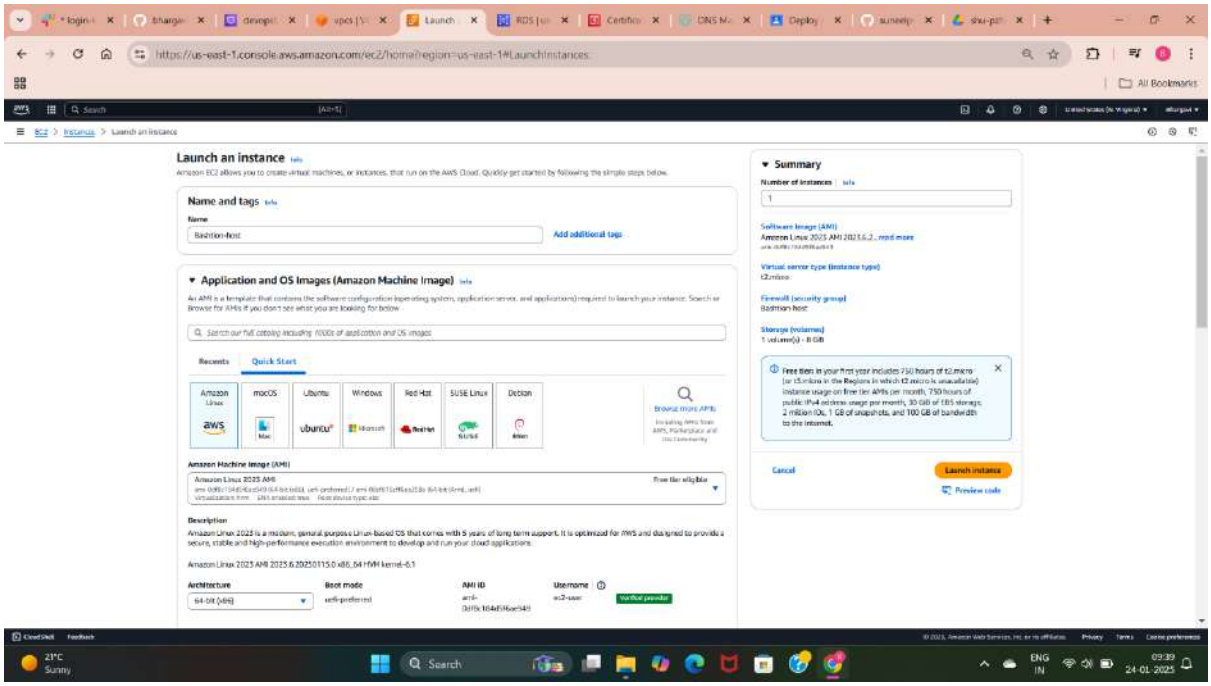
- Since private instances don't have public IPs, you can't directly SSH into them.
- The Bastion Host allows controlled access without exposing the entire network.

Minimizes Attack Surface

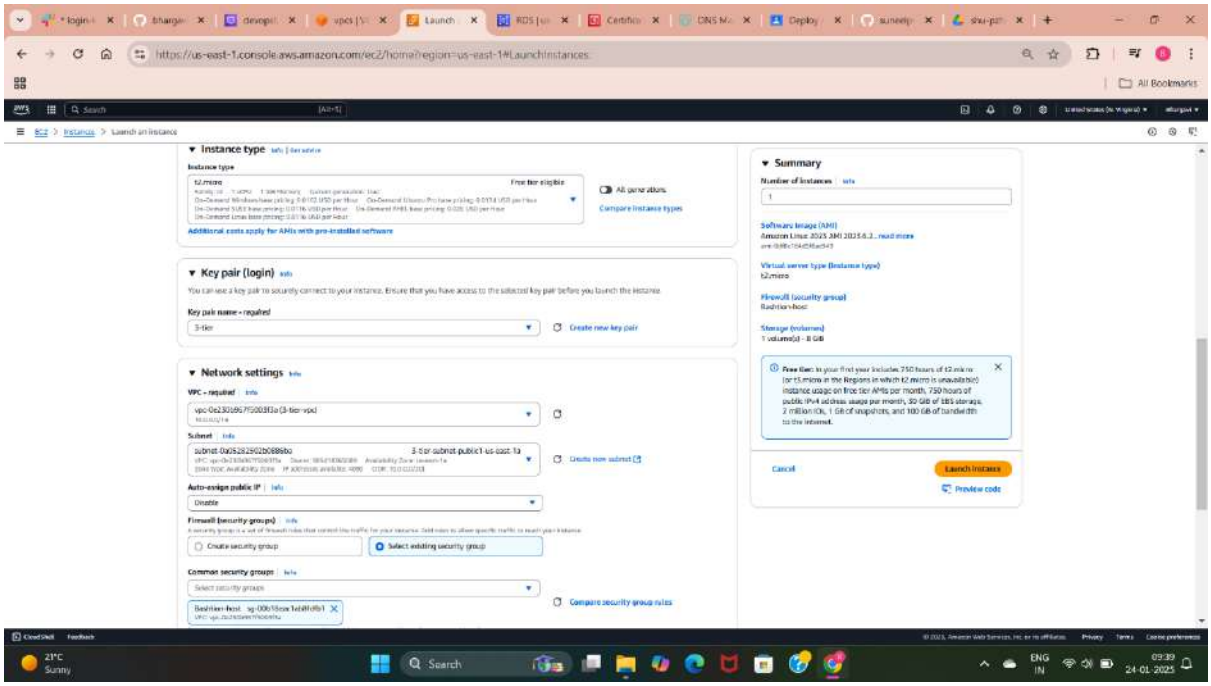
- Instead of exposing all instances to the internet, only one Bastion Host is publicly accessible.
- You can restrict access using security groups and IAM policies.

STEP 5: LAUNCHING BASTION HOST

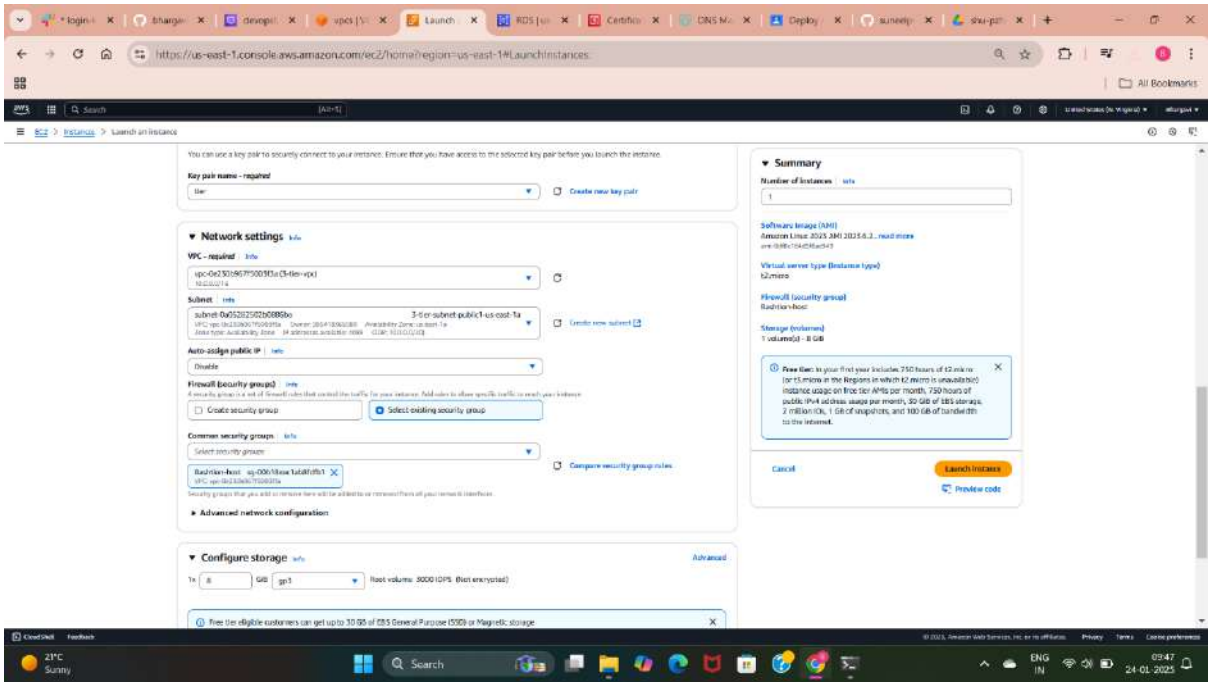
- ☑ Navigate to Services and search for EC2 under the "Compute" section.
- ☑ In the EC2 Dashboard, click on Launch Instance to create a new instance.



- ☑ Enter Bastion-Host as the name for your instance.
- ☑ AMI: Select Amazon Linux 2023 AMI (or another appropriate version for your needs).



- ☑ Instance Type: Select t2.micro (this is typically eligible for the free tier).



- ☑ Key Pair: Select an existing key pair or create a new one (tier.pem).
- ☑ Select the VPC 3-tier that you created earlier.
- ☑ Subnet: Select one of the Public Subnets you created (ensure this subnet has auto-assign Public IP enabled).
- ☑ Select the Bastion-Host security group that you created earlier. This ensures that the Bastion Host only allows SSH access from authorized sources (e.g., your IP or the VPC).
- ☑ Once everything looks good, click on Launch Instance.

RELATIONAL DATABASE SERVICE (RDS):

Amazon Relational Database Service (RDS) is a managed database service that helps you easily set up, operate, and scale a relational database in the AWS Cloud. It automates time-consuming database tasks like provisioning, backups, patching, and scaling.

Fully Managed Database

- AWS handles automatic backups, patching, scaling, and monitoring.
- Reduces manual effort compared to self-managed databases on EC2.

High Availability & Reliability

- Multi-AZ Deployment: Creates a standby copy in another Availability Zone for failover.
- Read Replicas: Improves performance by distributing read traffic.

Scalability

- Vertical Scaling: Change instance size (e.g., from db.t3.micro to db.m5.large).
- Horizontal Scaling: Add Read Replicas to distribute read queries.

Security & Compliance

- Encryption (AWS KMS) for data at rest and in transit.
- IAM-based access control and VPC security groups to restrict access.

- Automated backups for disaster recovery.

Cost-Effective

- Pay-as-you-go pricing model (only pay for what you use).
- Reserved Instances for long-term savings.

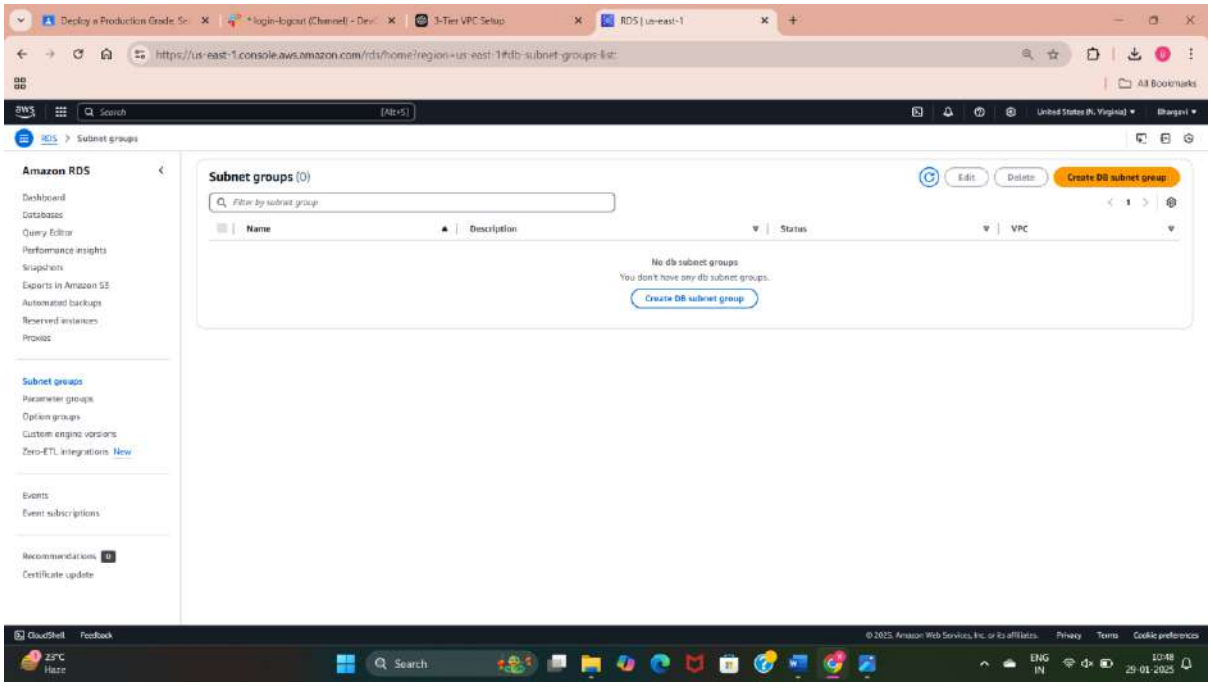
MYSQL WORKBENCH

MySQL Workbench is a GUI (Graphical User Interface) tool provided by MySQL for designing, managing, and administering MySQL databases. It allows users to interact with MySQL databases visually instead of using command-line queries.

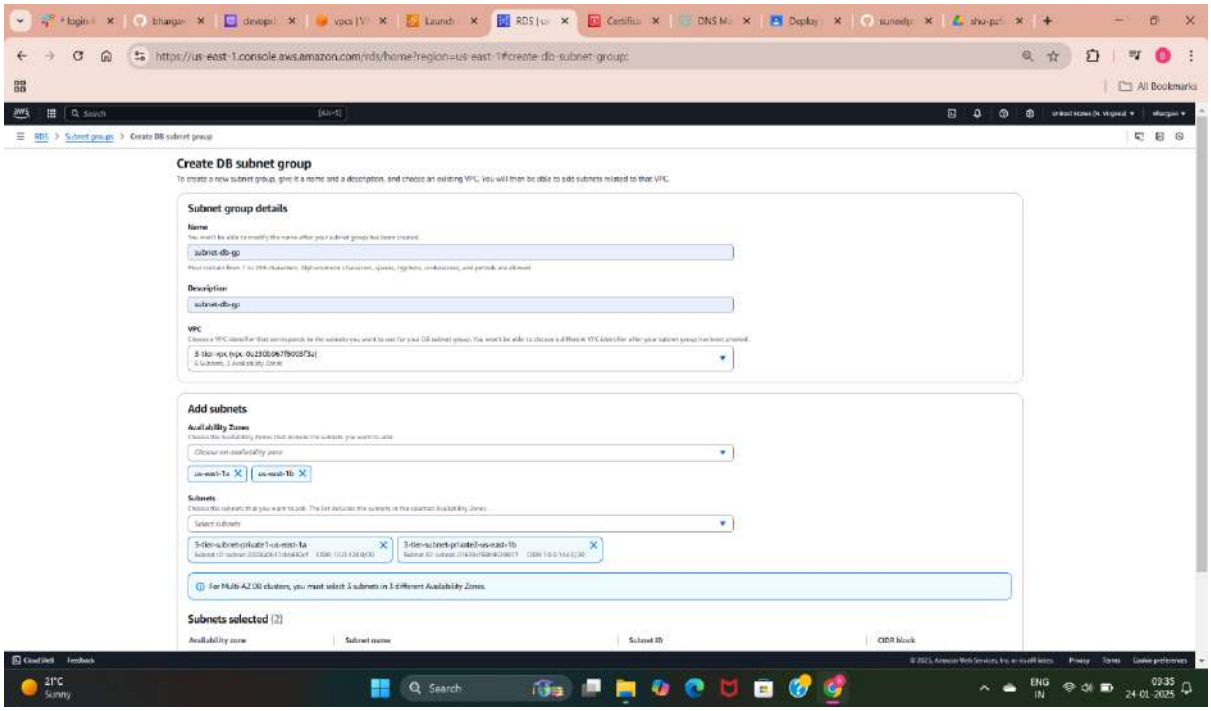
STEP 6: SETUP DATA TIER WITH RDS

6.1 CREATE DB SUBNET GROUP

➲ Navigate to Services and search for RDS under the "Database" section.
➲ In the RDS Dashboard, on the left-hand side, click on Subnet Groups under Network & Security.

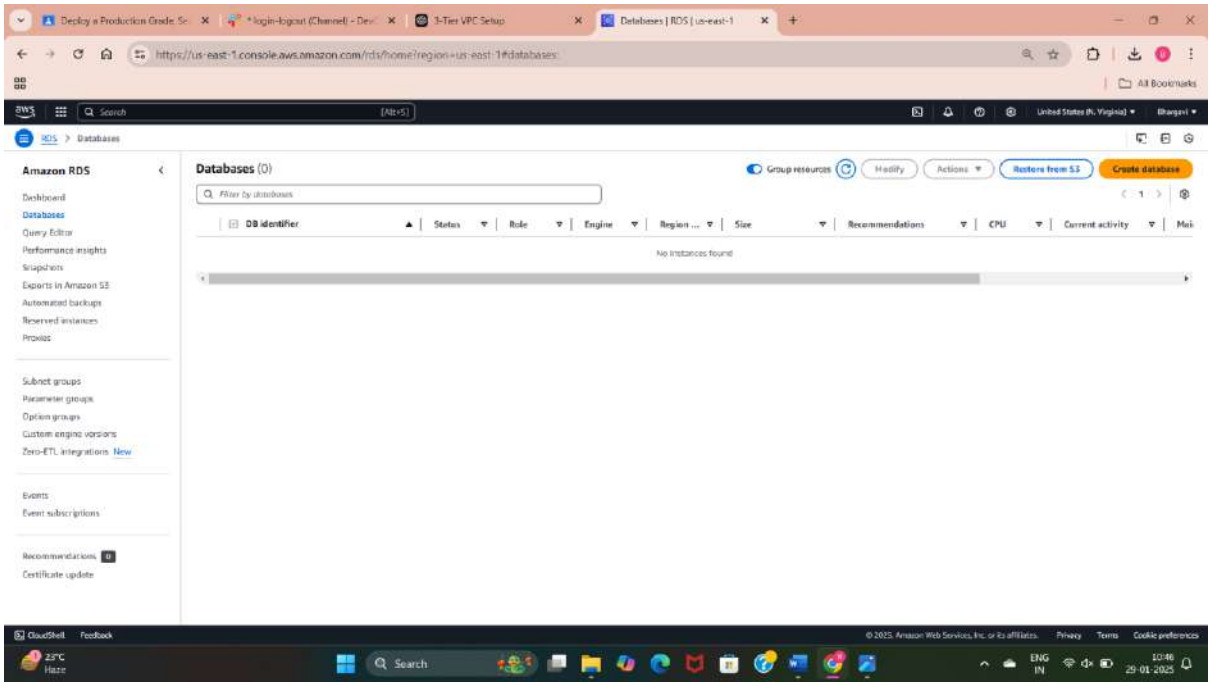


➲ Click on Create DB Subnet Group at the top of the page.

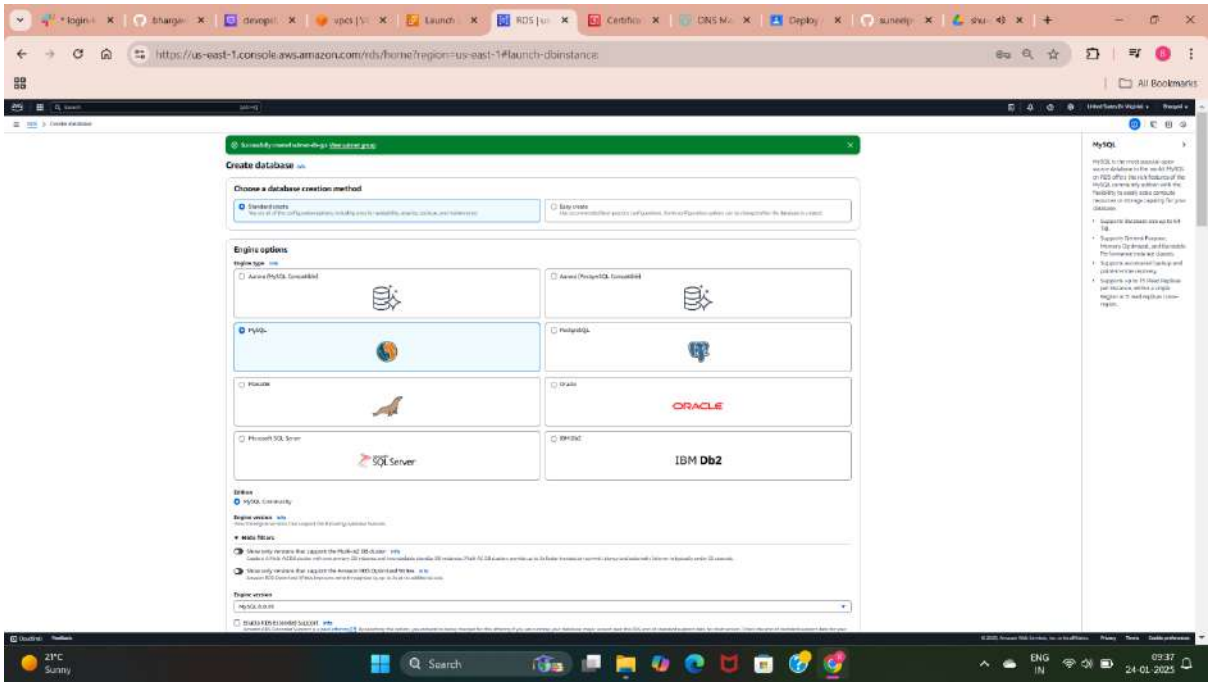


- Enter subnet-db-gp as the name of your DB Subnet Group.
- Select the VPC 3-tier you created earlier.
- Select both Availability Zones that you created subnets in (for high availability).
- Select 2 Private Subnets (to ensure your database instances are not exposed to the public internet).
- Click on Create Subnet Group after completing these configurations.
- After creating the subnet group, return to the RDS Dashboard and click on Databases.

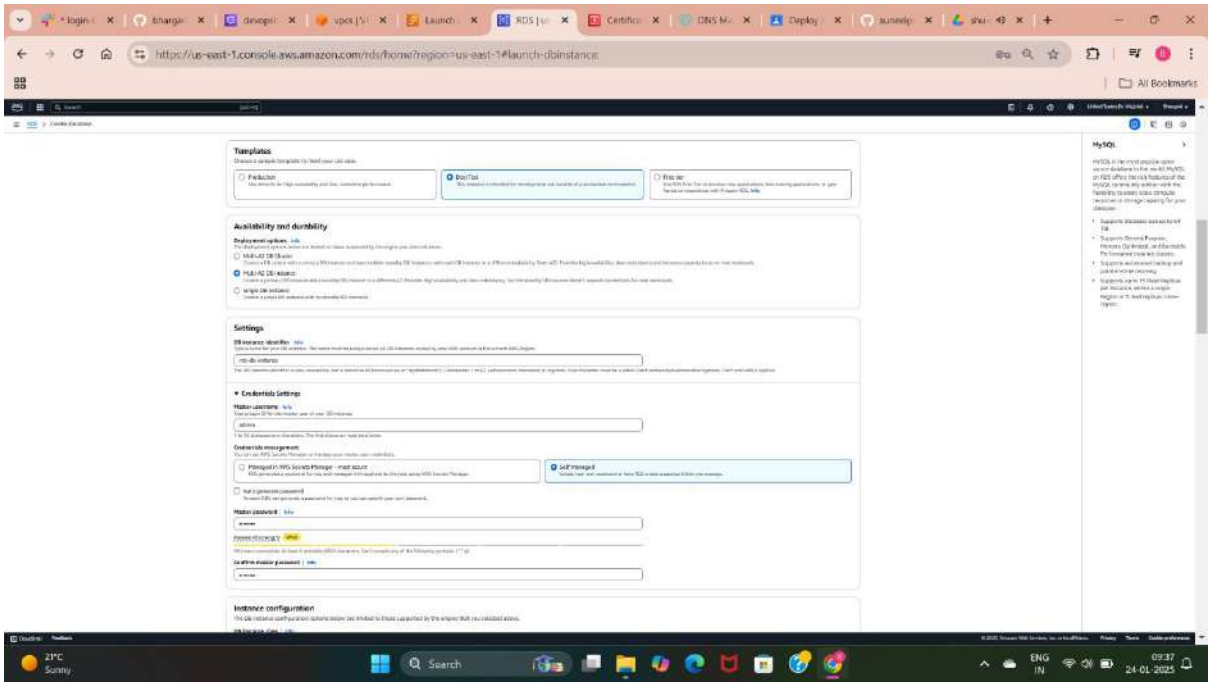
6.2 CREATE DATABASE



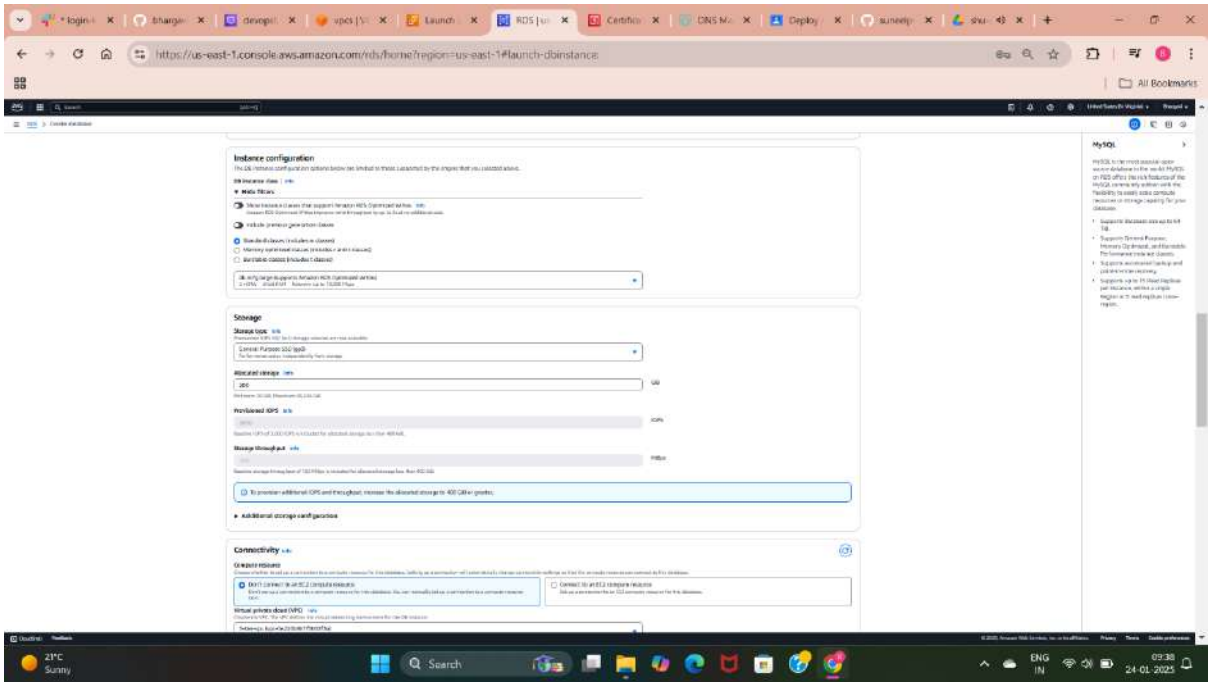
- Click on Create Database to start the database creation process.



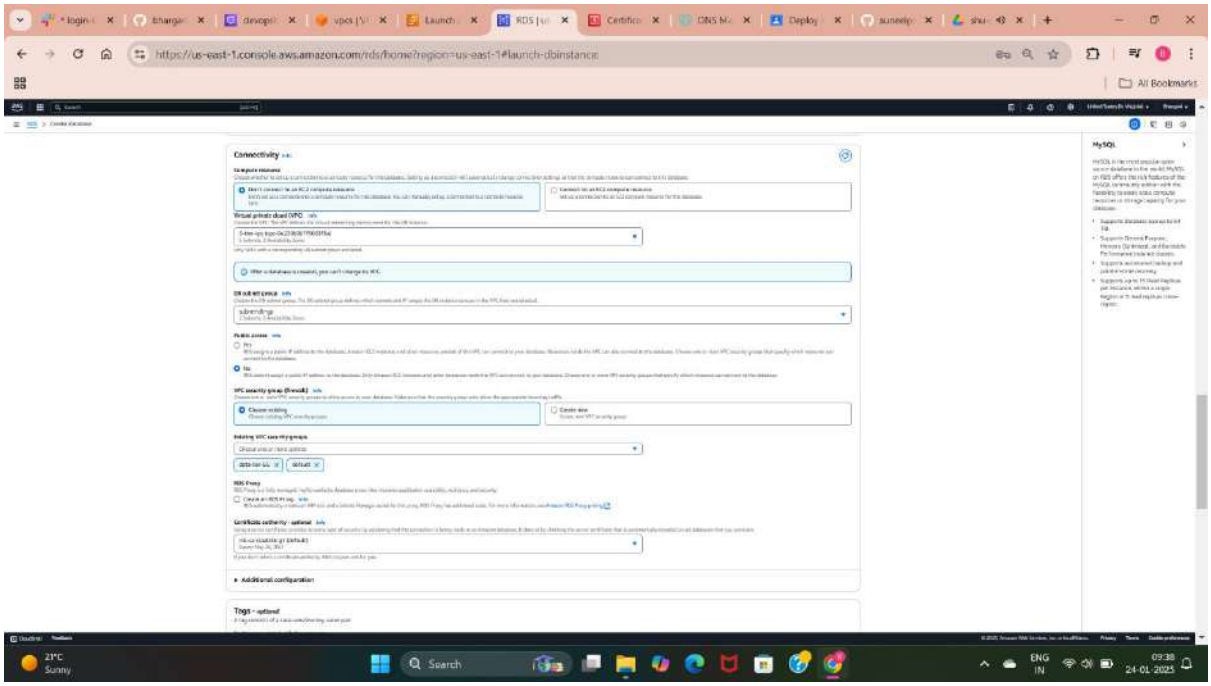
- Select Standard and choose SQL (choose the specific SQL database engine like MySQL, PostgreSQL, or others depending on your needs).



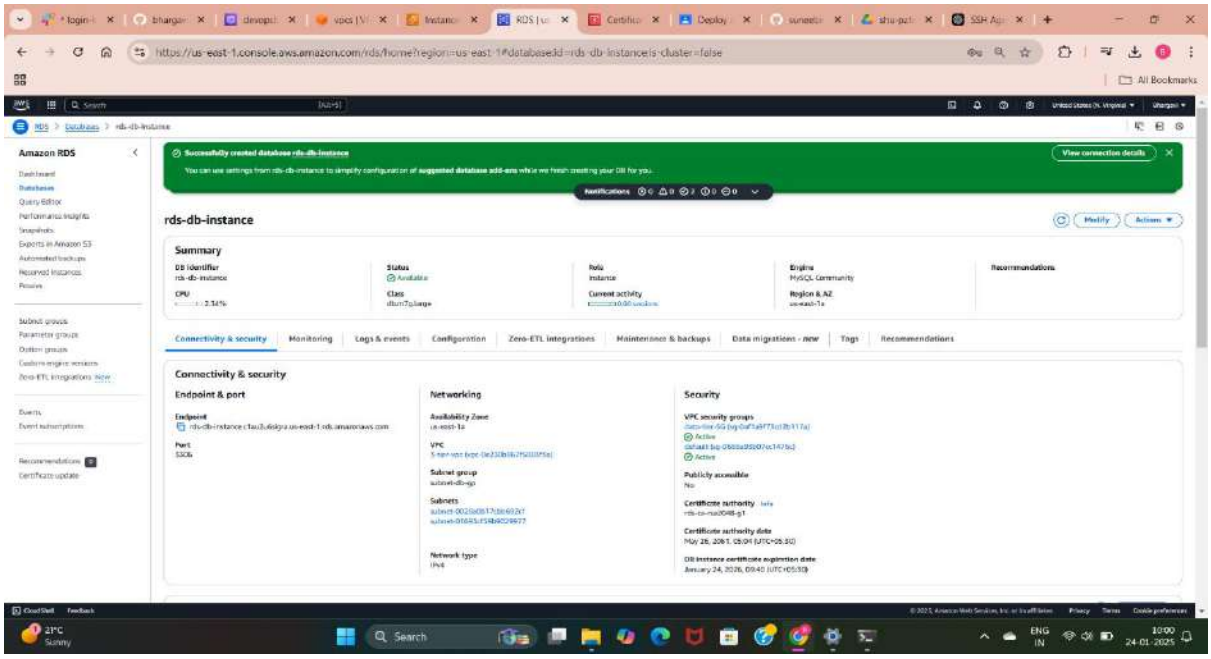
- Select the dev/test template (suitable for development or testing environments).
- Select Multi-AZ deployment to enable high availability.
- Configure other settings like DB instance identifier, username, password, storage, etc.
- Enter dev-db-instance as the identifier for your database instance.
- Select Self-managed and enter the password as admin123 (make sure to remember or store the password securely).



- Select the appropriate instance class for your requirements (e.g., db.m7g.large for testing purposes).

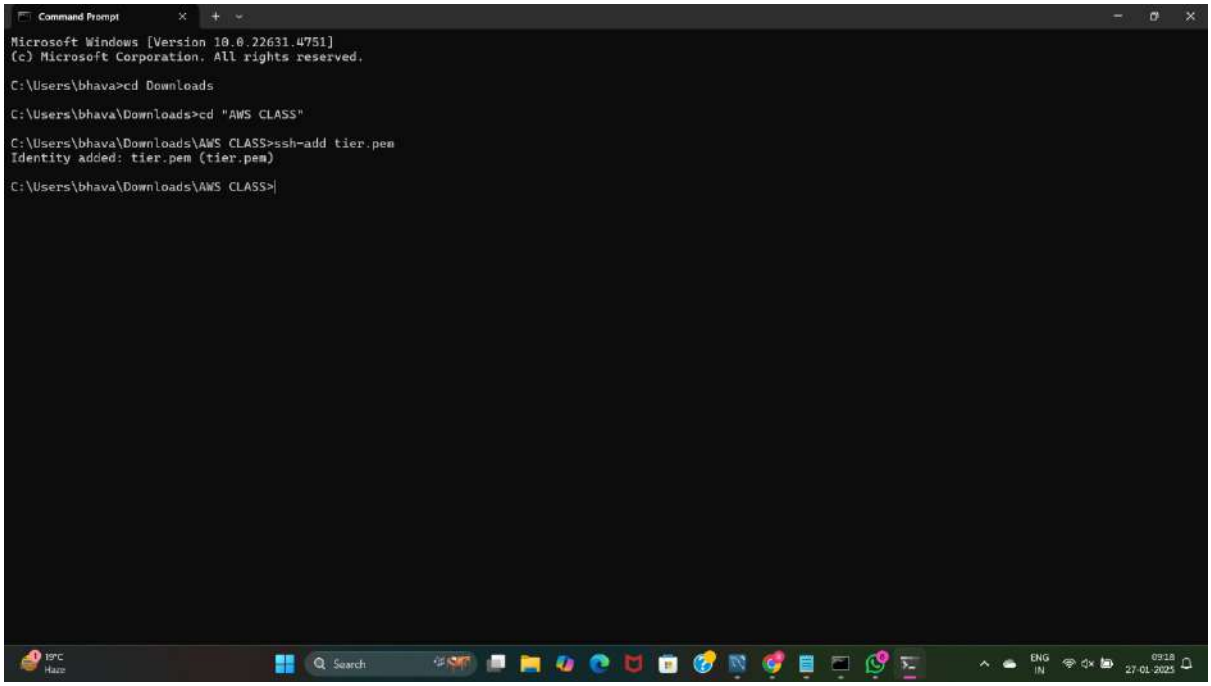


- Under the VPC section, select the 3-Tier VPC.
- For the DB Subnet Group, select the dev-db-subnet-group you created earlier.
- Configure backup retention, monitoring, and other settings according to your needs (you can leave the defaults for now).
- Review all settings and click on Create database.

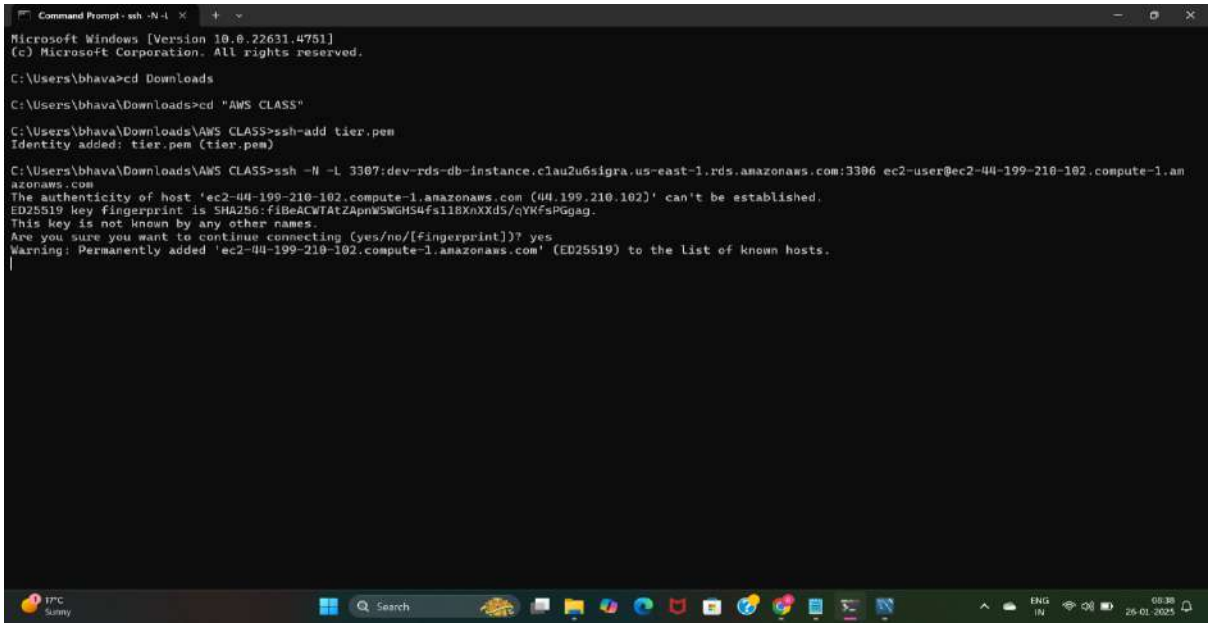


- Once the database is created, we can view it in the RDS dashboard. The instance will be deployed with Multi-AZ for high availability.

CONNECT TO YOUR RDS DATABASE



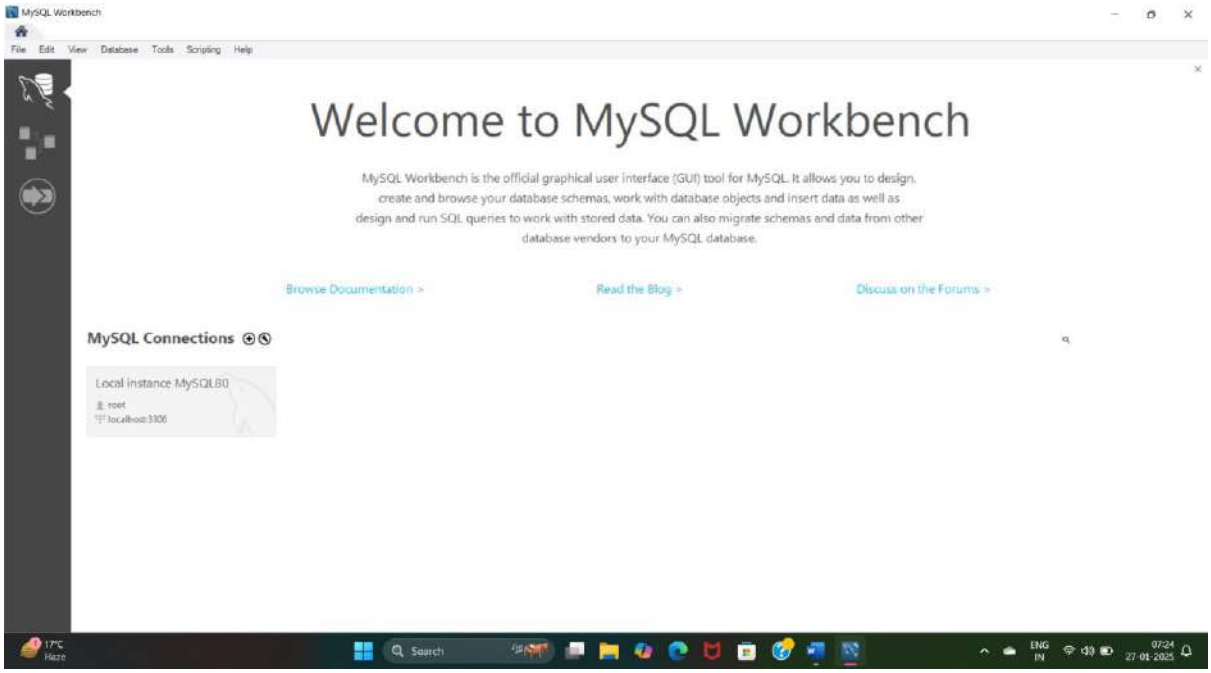
- Open Command Prompt for Windows/Terminal and locate the path of your Keypair
- To add key to SSH Agent use “ssh-add your_key.pem(tier.pem)”.



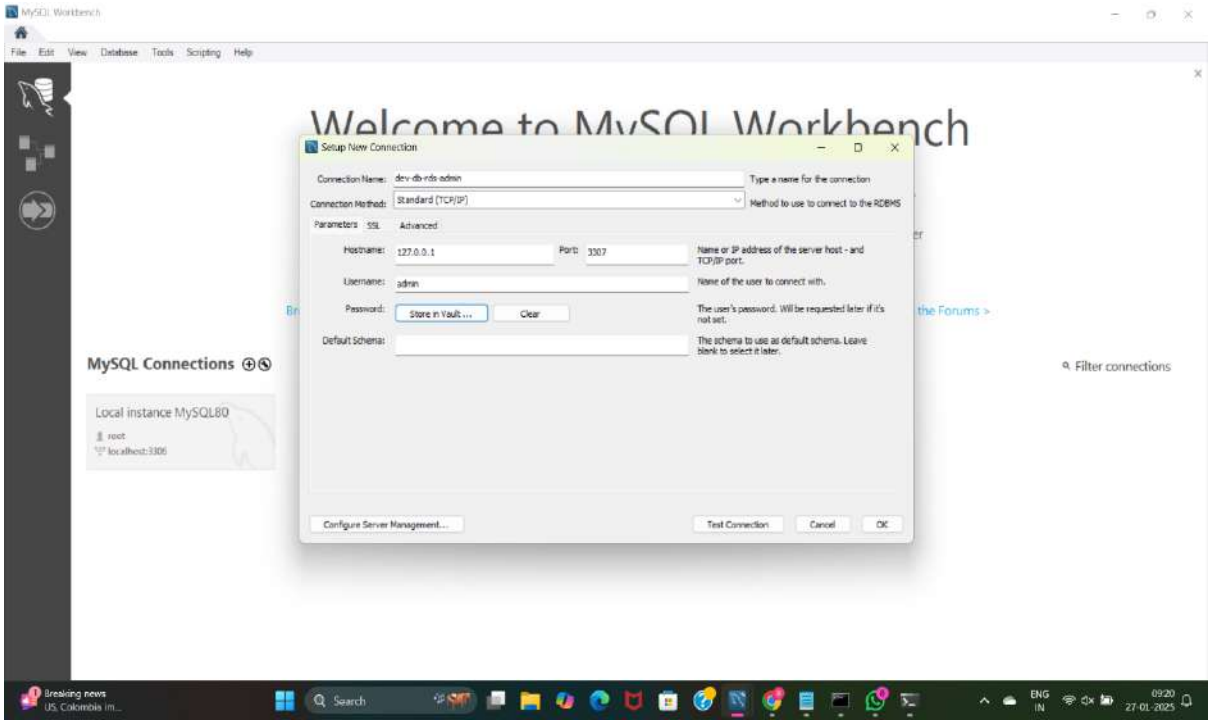
- Connect to Database using

  “ssh -N -L 3307:<Database-endpoint>:3306 ec2-user@bastion_host_public_ip”.

- Download MySQL Workbench.
- Open MySQL Workbench.

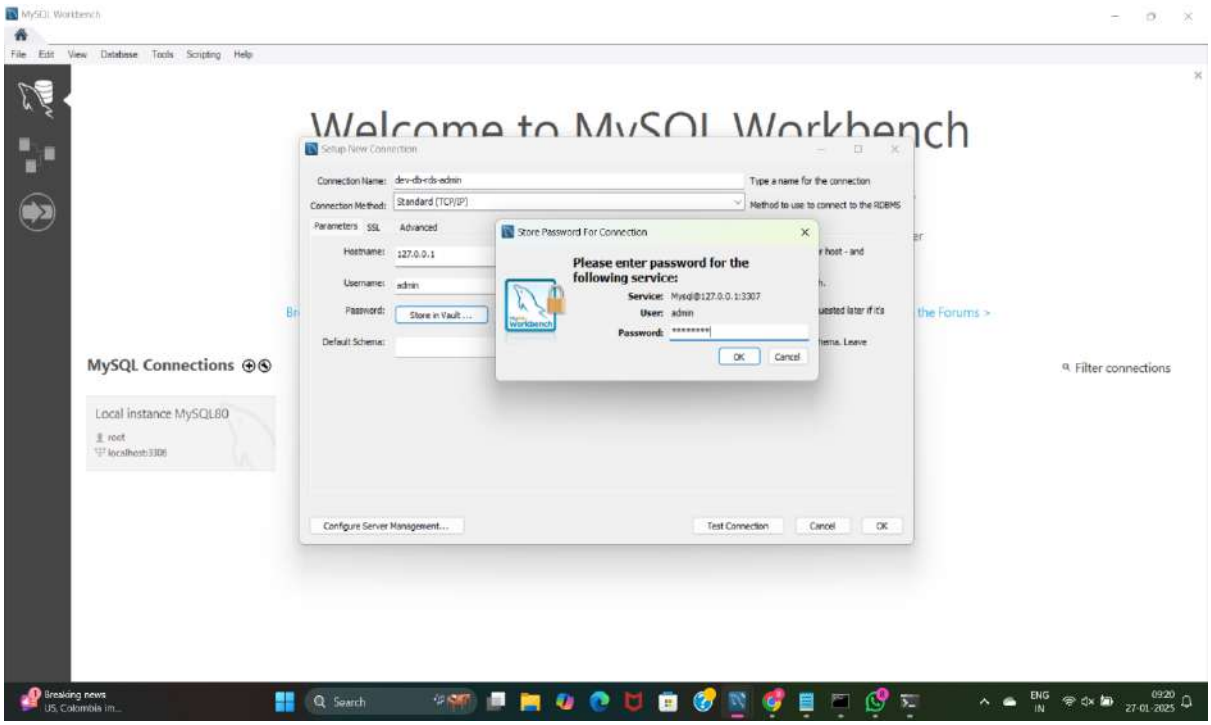


- Click the “+” symbol next to MySQL Connections to create a new connection.

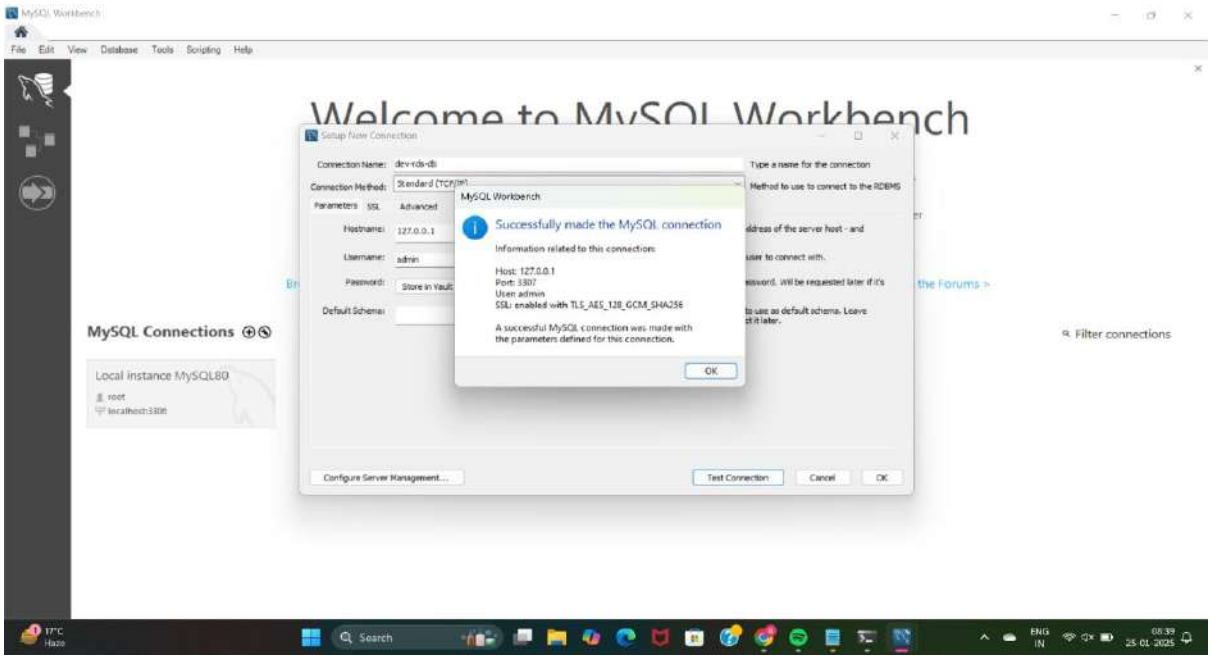


Fill in the following details:

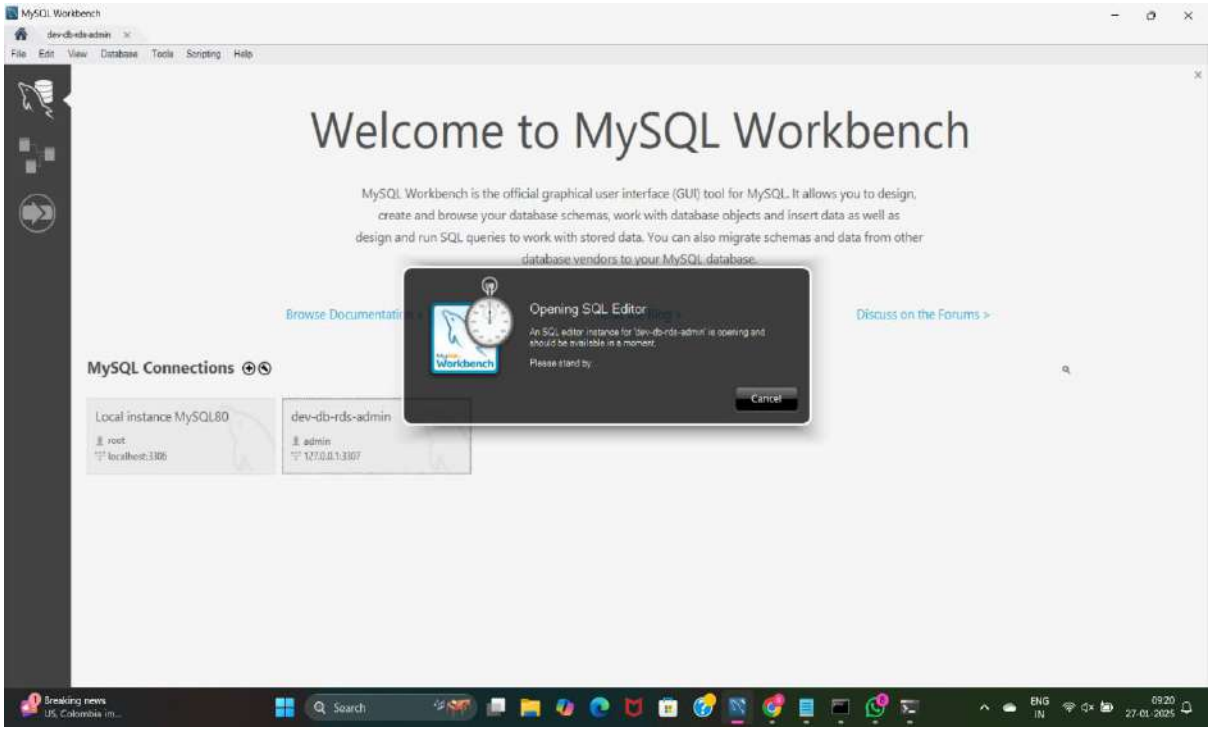
- Connection Name: dev-rds-db
- Port: 3307 (or the port you've configured; usually, MySQL uses port 3306, but change this if your setup uses 3307).
- Username: admin (the master username you set during RDS creation).



- Password: Click on Store in Vault or Store in Keychain, and enter the password admin123.

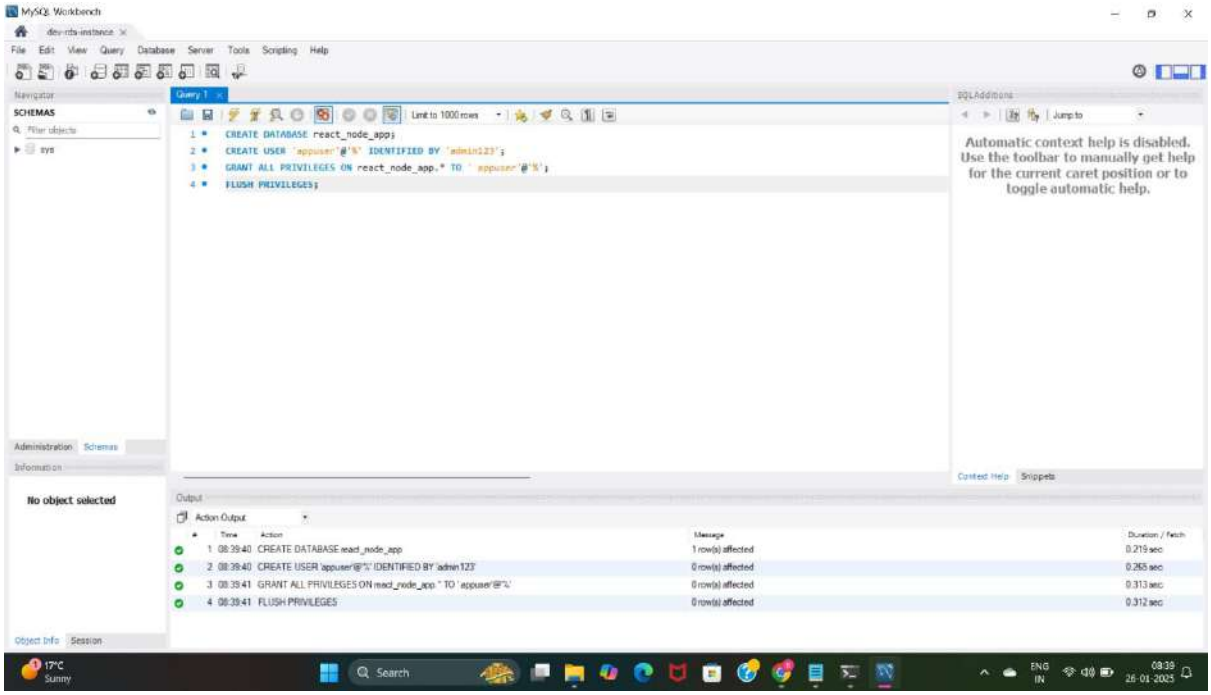


➲ Click Test Connection. If successful, it will confirm the connection. Click OK to save.

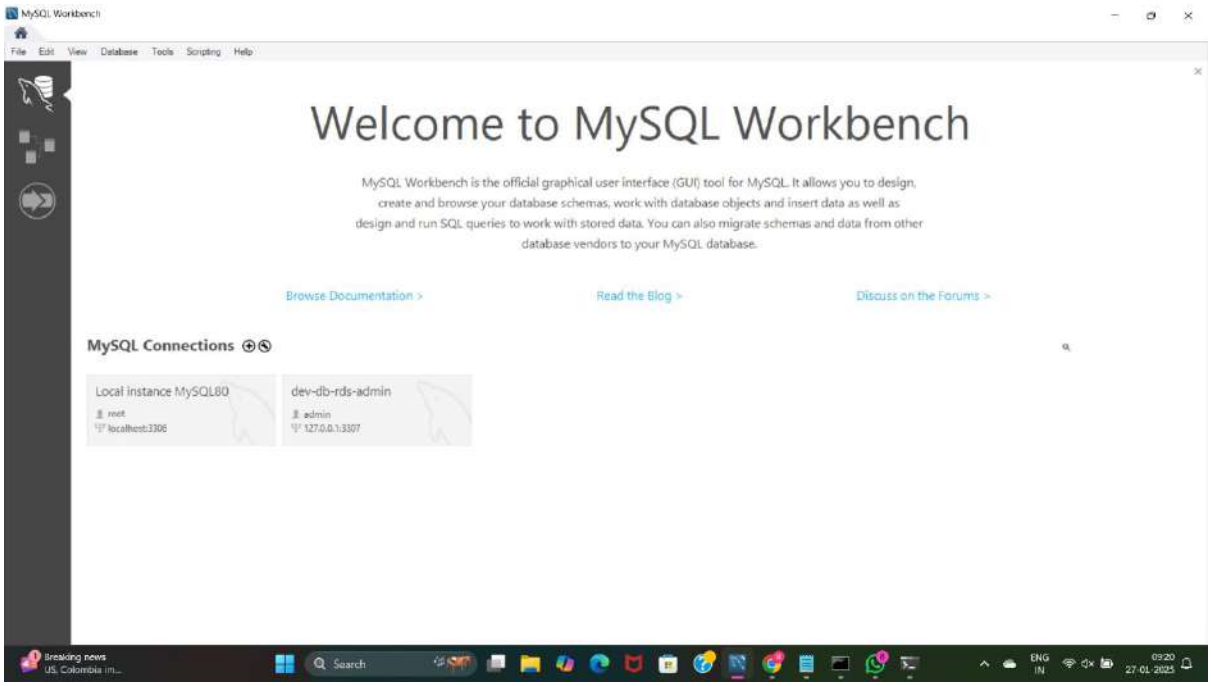


➲ Click on the newly created connection (dev-rds-db) from the MySQL Workbench homepage.

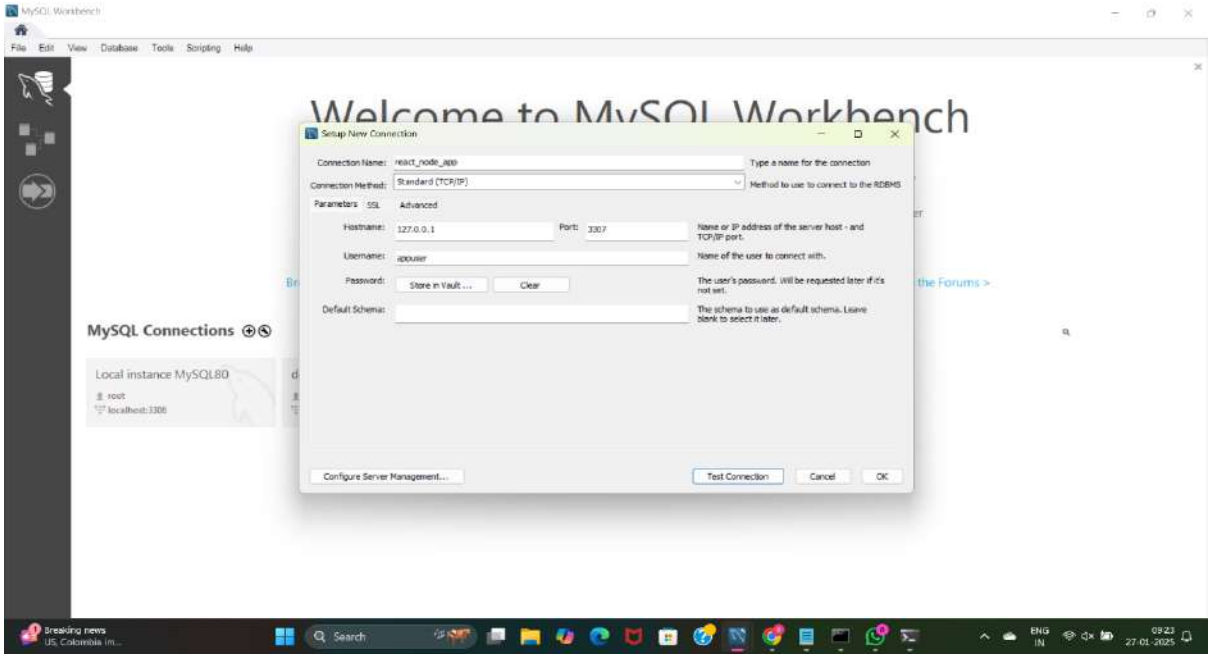
➲ This will open the SQL Editor for the connected RDS instance.



➲ Create the new database with the name “react_node_app” with username and password using above script and run the script by selecting all lines.

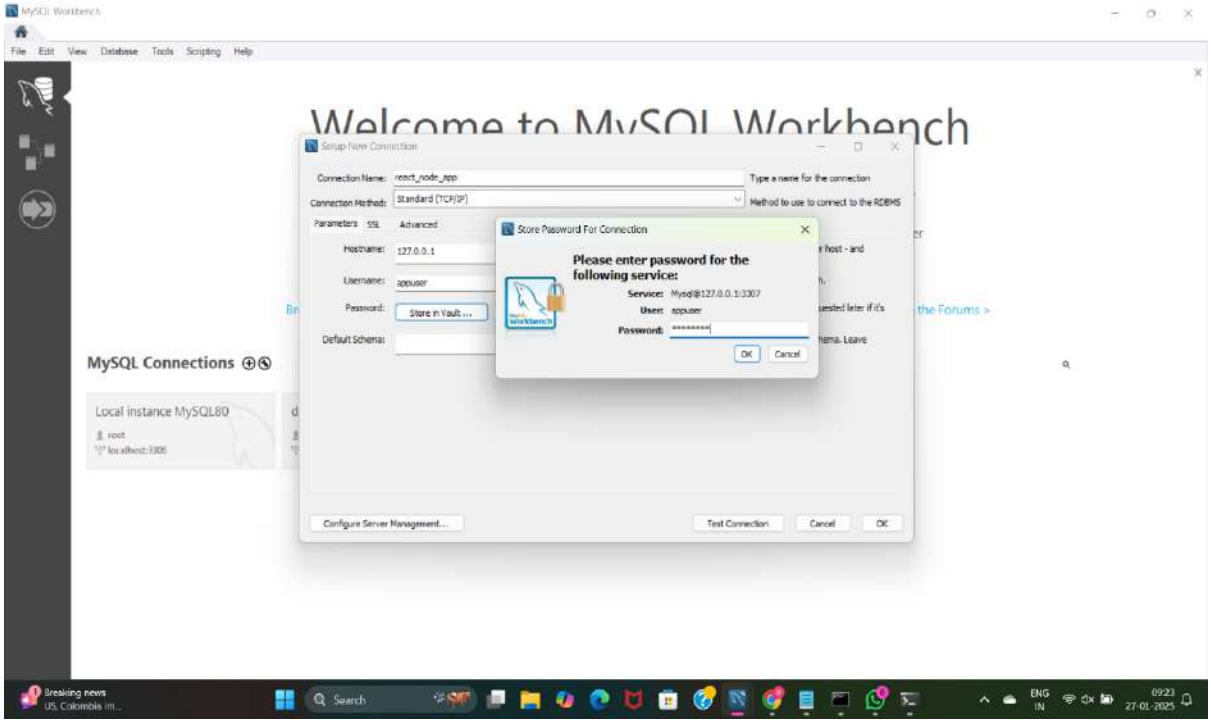


➲ Go back to the MySQL Workbench Home screen.
➲ Click “+” to create a new connection.

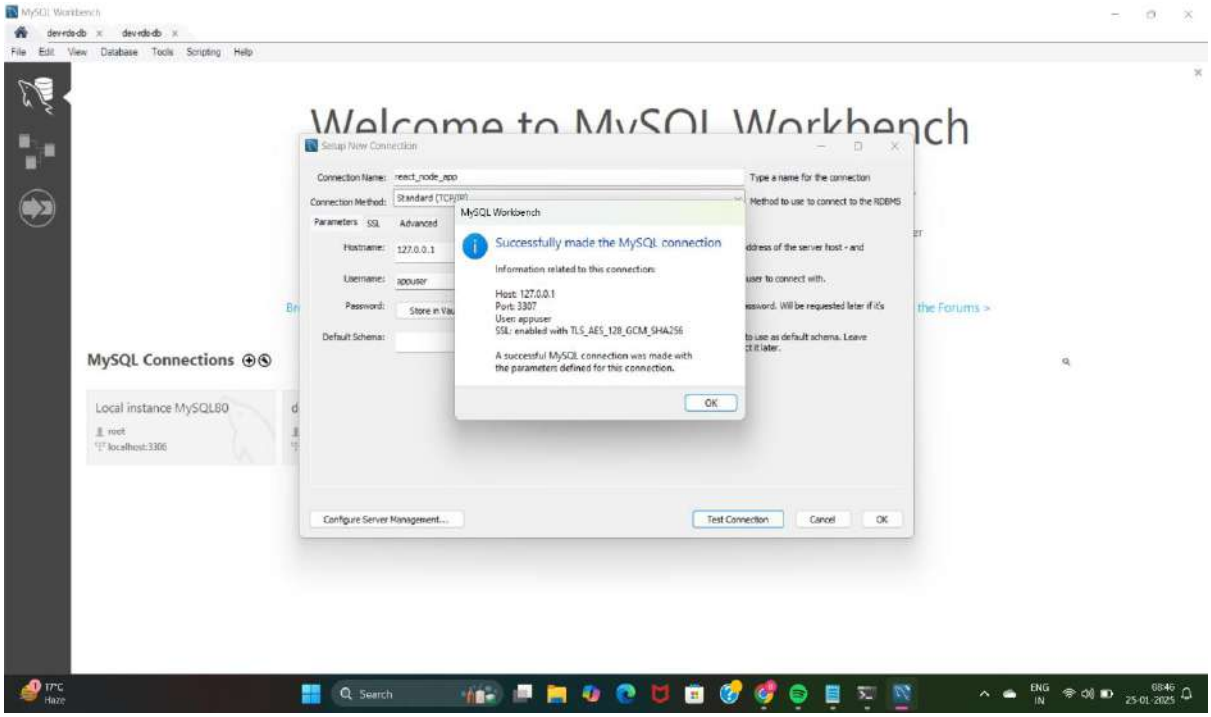


Fill the details :

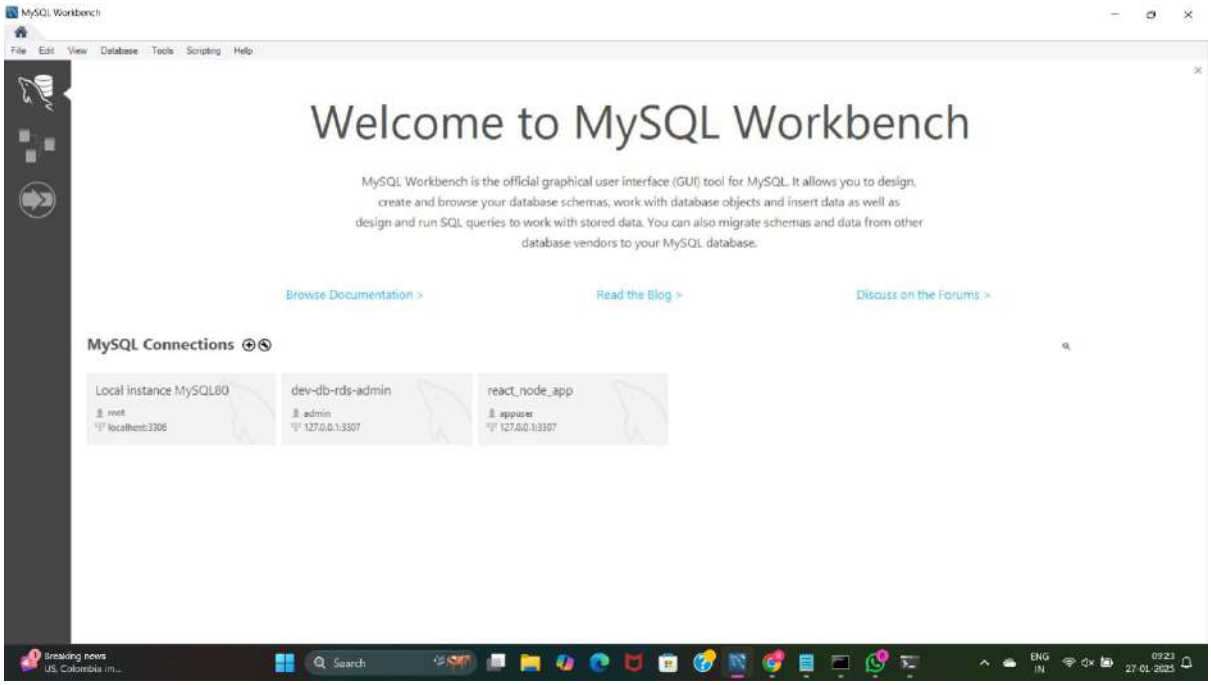
- Name: rds-dev-db-appuser
- Hostname: Your RDS endpoint
- Port: 3307
- Username: appuser



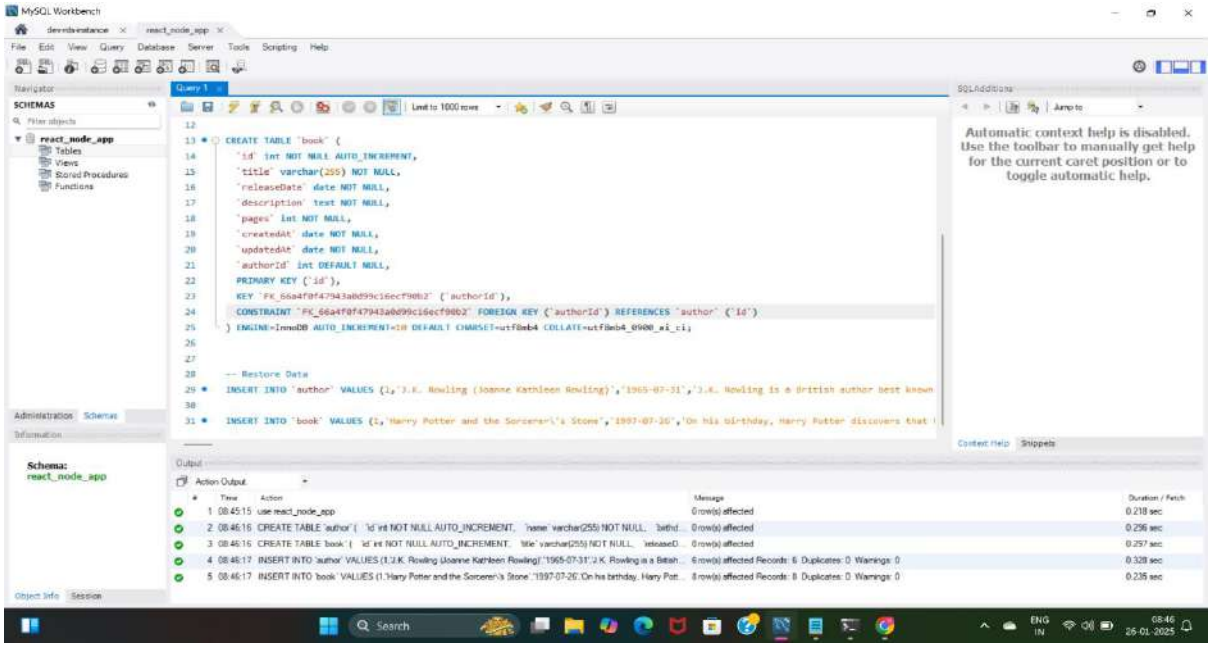
- Password: admin123 (store in vault/keychain)



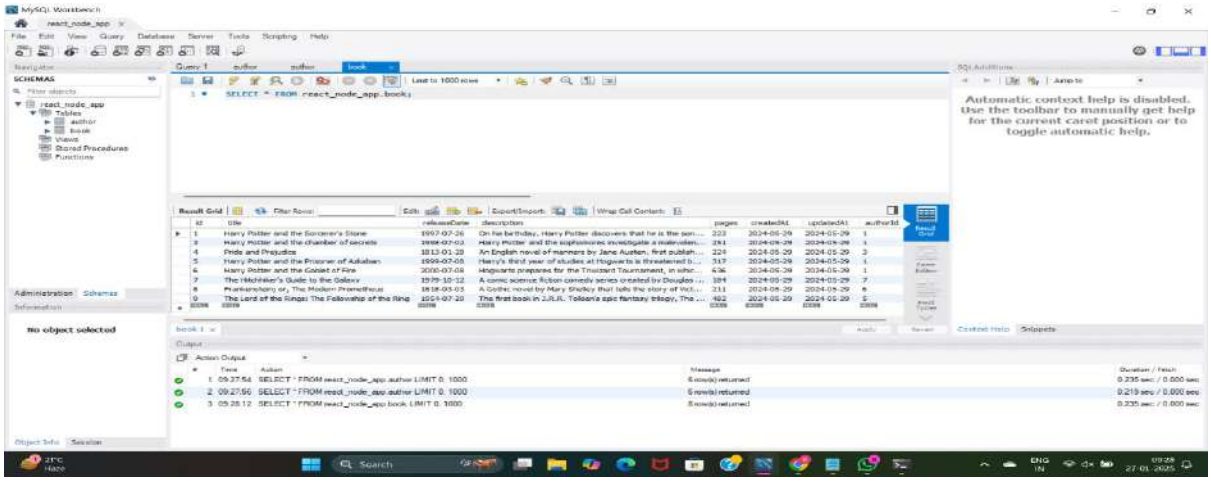
➲ Test the connection and click OK if successful.

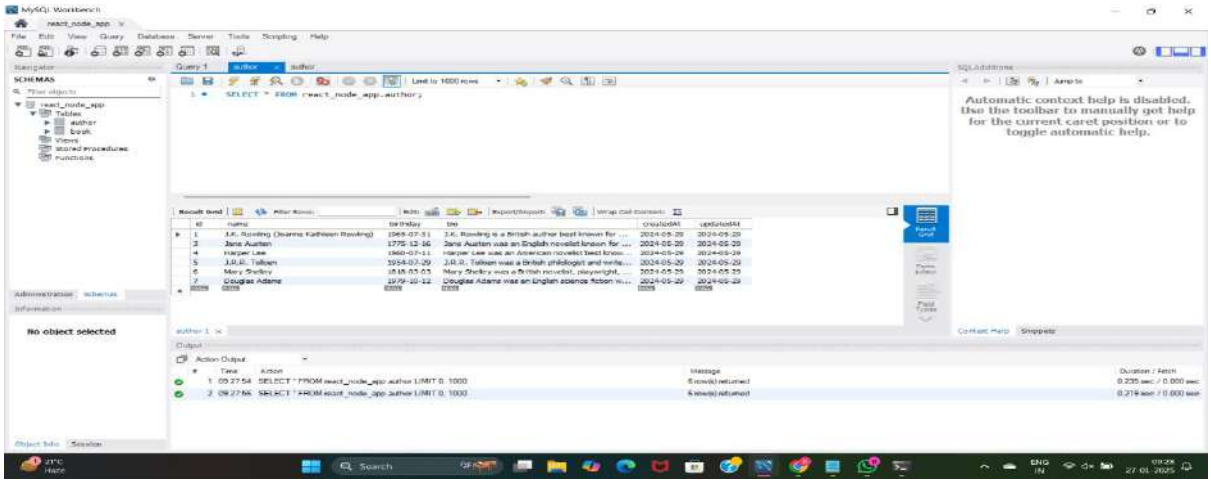


➲ Open newly created database and run some queries.



- Reconnect using rds-dev-db-appuser and open the SQL Editor. Now you can execute queries on the dev_database schema.





- We can also check the tables related to books and authors.

PRESENTATION TIER

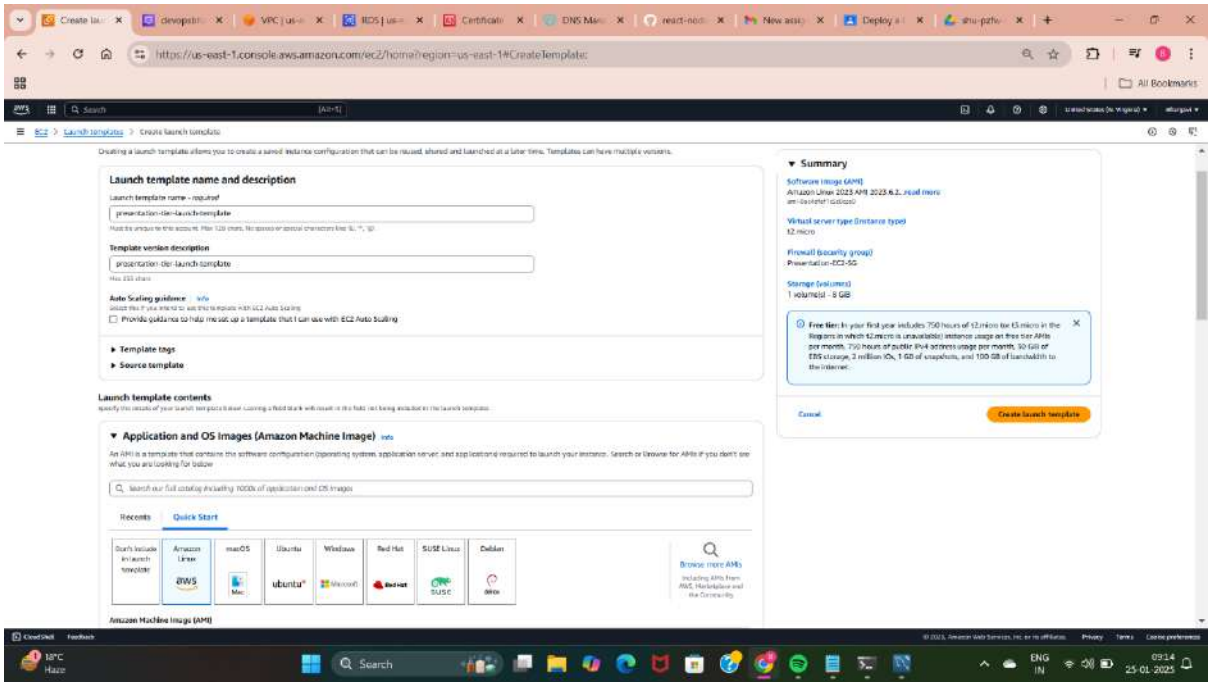
The Presentation Tier (also called the Frontend Tier) is the top layer in a 3-tier architecture. It is responsible for:

- ☑ Interacting with users(UI/UX).
- ☑ Displaying data fetched from the backend.
- ☑ Sending user input to the business logic tier.

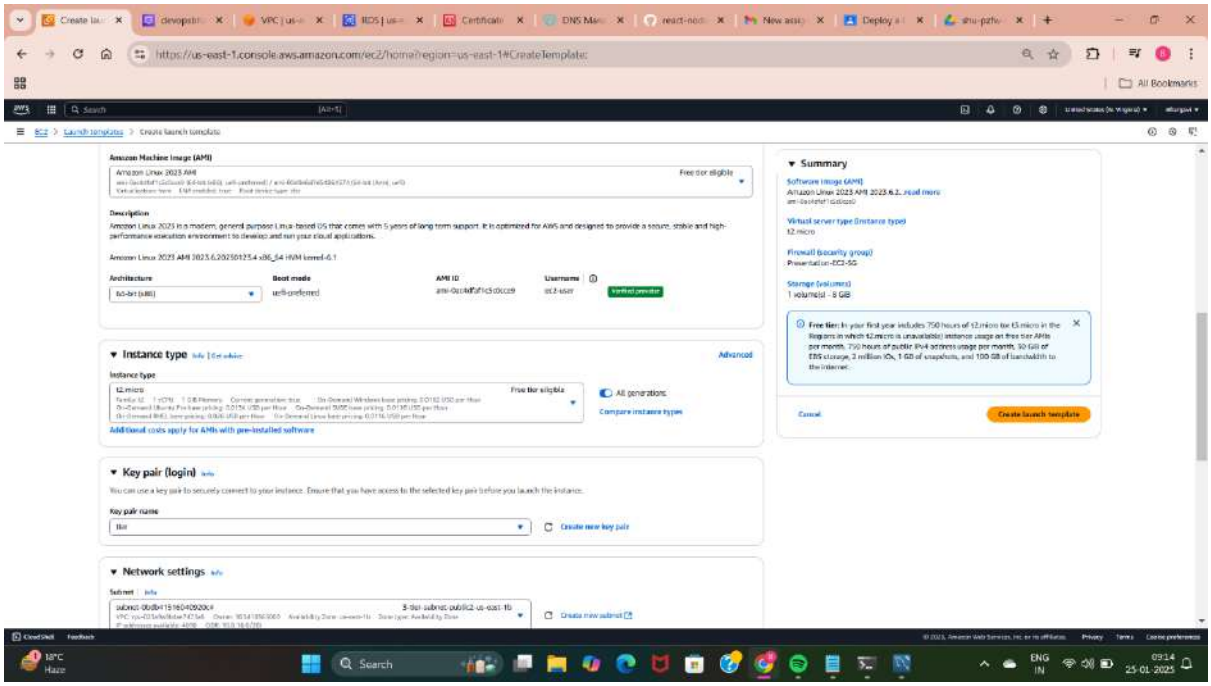
STEP 7: SETTING UP THE PRESENTATION TIER

7.1: CREATE A LAUNCH TEMPLATE FOR PRESENTATION TIER

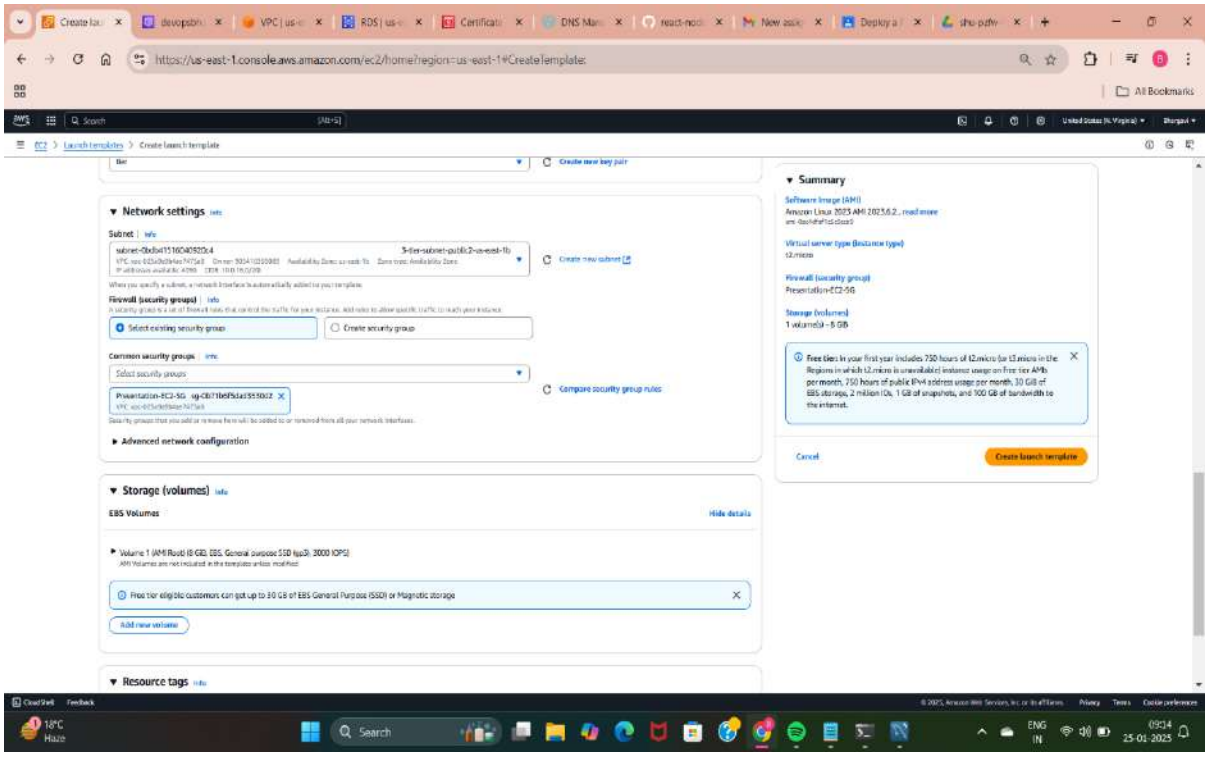
- ➢ Go to the EC2 Dashboard in the AWS Management Console.
- ➢ Click on "Launch Templates" under the Instances section.
- ➢ Click "Create Launch Template" and fill in the required details.



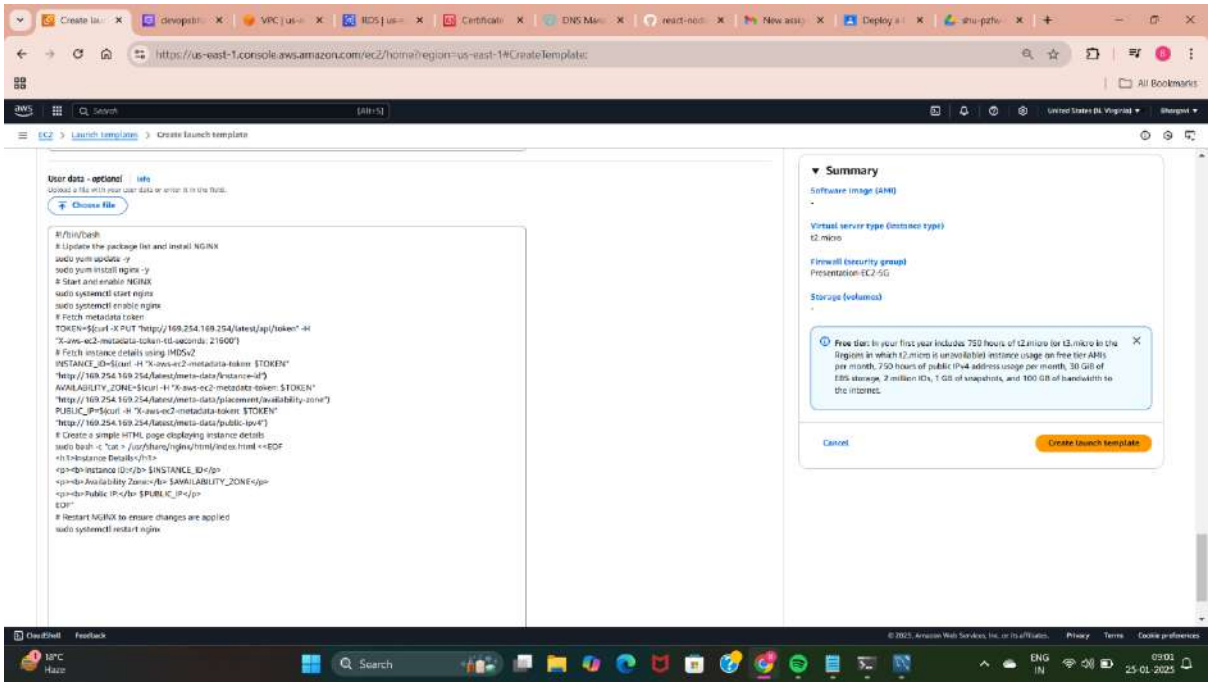
- ➢ Launch Template Name: Presentation-Tier-LaunchTemplate.
- ➢ Version Description: Presentation-Tier-LaunchTemplate.
- ➢ AMI: Use Amazon Linux 2023 or a specific AMI based on your application's requirements.



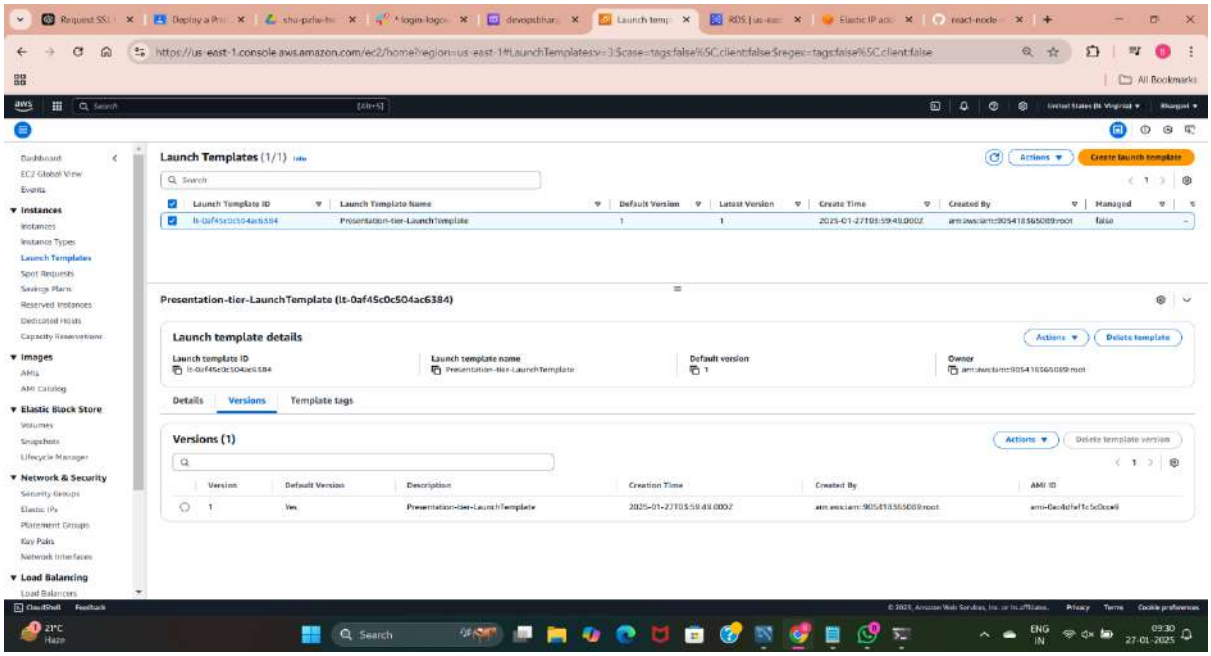
- Instance Type: t2.micro (or another type depending on your workload).
- Key Pair: Select an existing key pair or create a new one.



- Select the public subnet.
- Security Groups: Select the Presentation-Tier-EC2 security group created earlier.
- IAM Role: Attach a role with permissions for the instance to communicate with other AWS services if needed.



- Add the script to the Advanced Details > User Data section to set up the Presentation Tier EC2 instance.
- Click "Create Launch Template".



- Presentation launch template is created with default version as 1.

TARGET GROUP

A Target Group in AWS is a collection of instances (EC2), IP addresses, or Lambda functions that serve as the backend for Elastic Load Balancers (ELB)**.** It defines how AWS routes traffic to your application.

- It routes incoming traffic to the registered targets evenly across multiple EC2 instances.
- Supports different target types (EC2, IP, Lambda).
- Works with Application Load Balancer (ALB), Network Load Balancer (NLB), Gateway Load Balancer (GWLB).
- Allows health checks to monitor target availability.
- Used in Auto Scaling for distributing requests efficiently.

TYPES OF TARGET GROUPS

Instance Target Group

- Targets: EC2 instances
- Best for applications running on EC2 instances behind a Load Balancer.

IP Target Group

- Targets: IP addresses (IPv4 only)
- Best for on-premises servers, hybrid cloud setups, or containerized applications.

Lambda Target Group

Targets: AWS Lambda functions

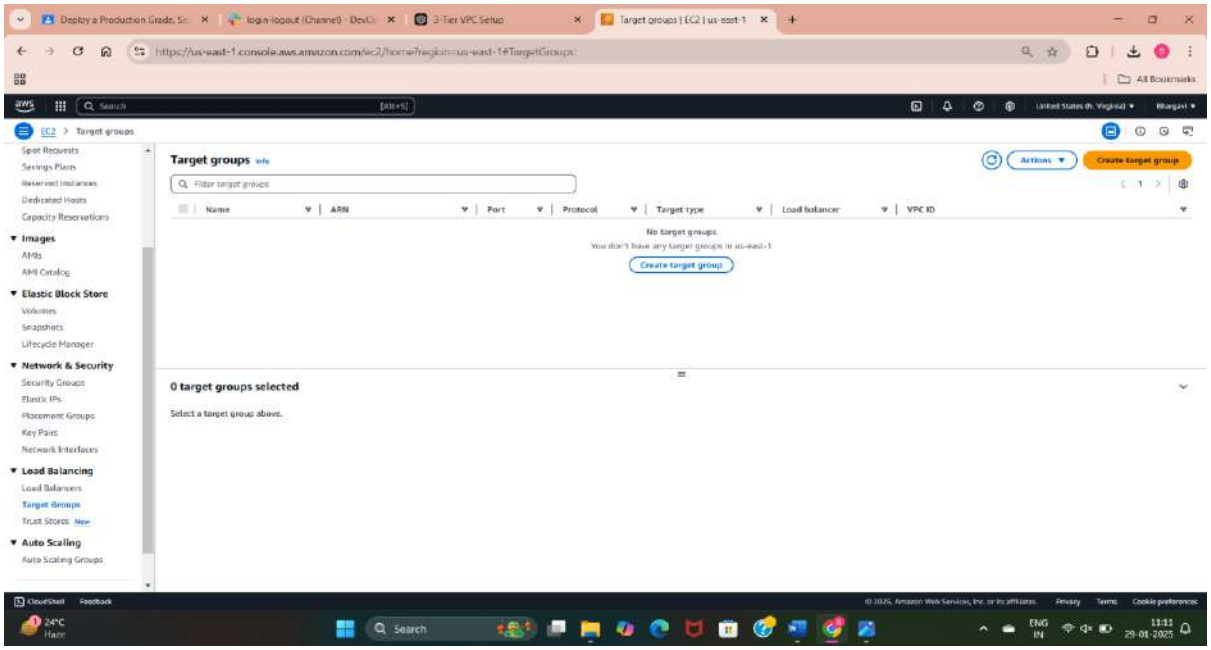
- Best for serverless applications where a Load Balancer triggers Lambda functions instead of EC2 instances.

Application Load Balancer (ALB) Target Group

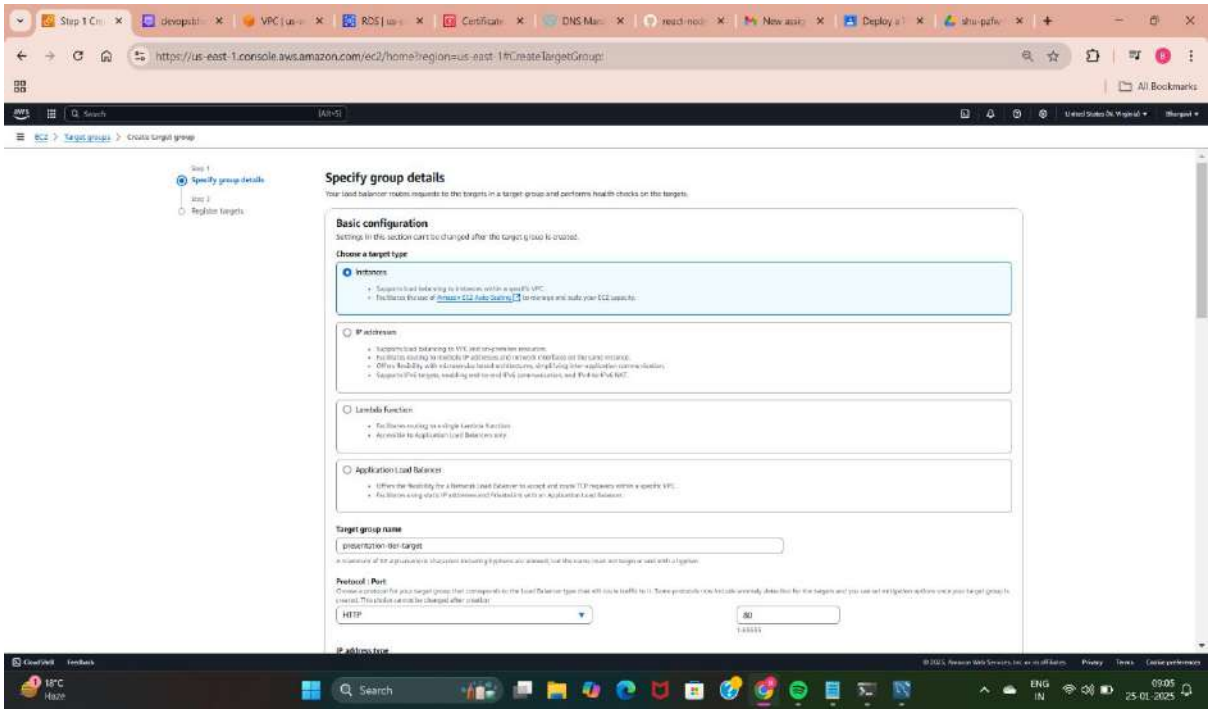
- Targets: Another ALB
- Best for layered architectures where one Load Balancer routes traffic to another ALB.

7.2 CREATING PRESENTATION TIER TARGET GROUP:

- Go to the EC2 Dashboard in the AWS Management Console.
- Under the Load Balancing section, click on Target Groups.

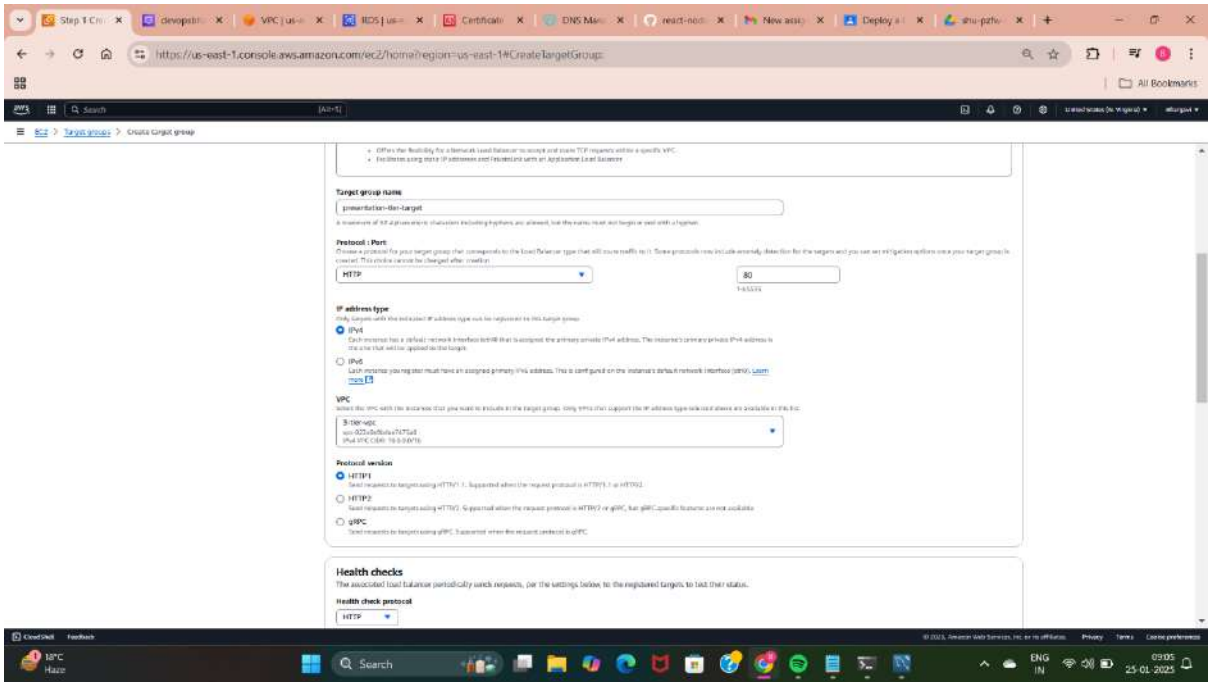


- Click on "Create target group".



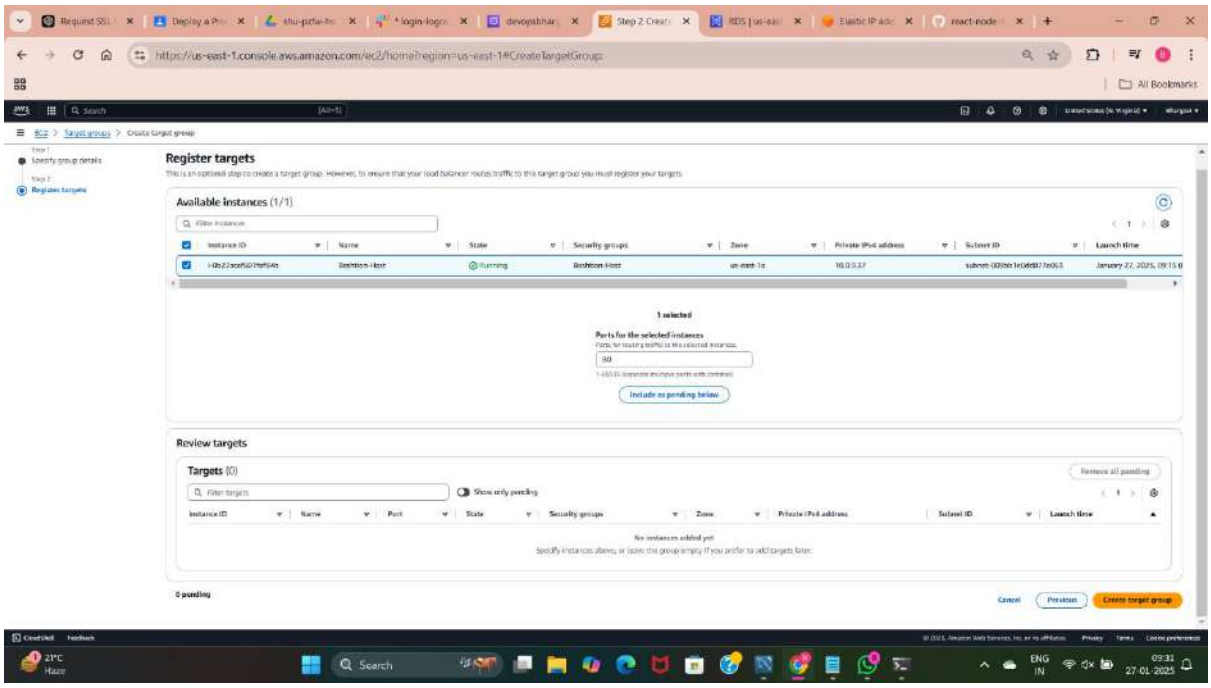
Choose a Target Type:

- Select Instances as the target type.

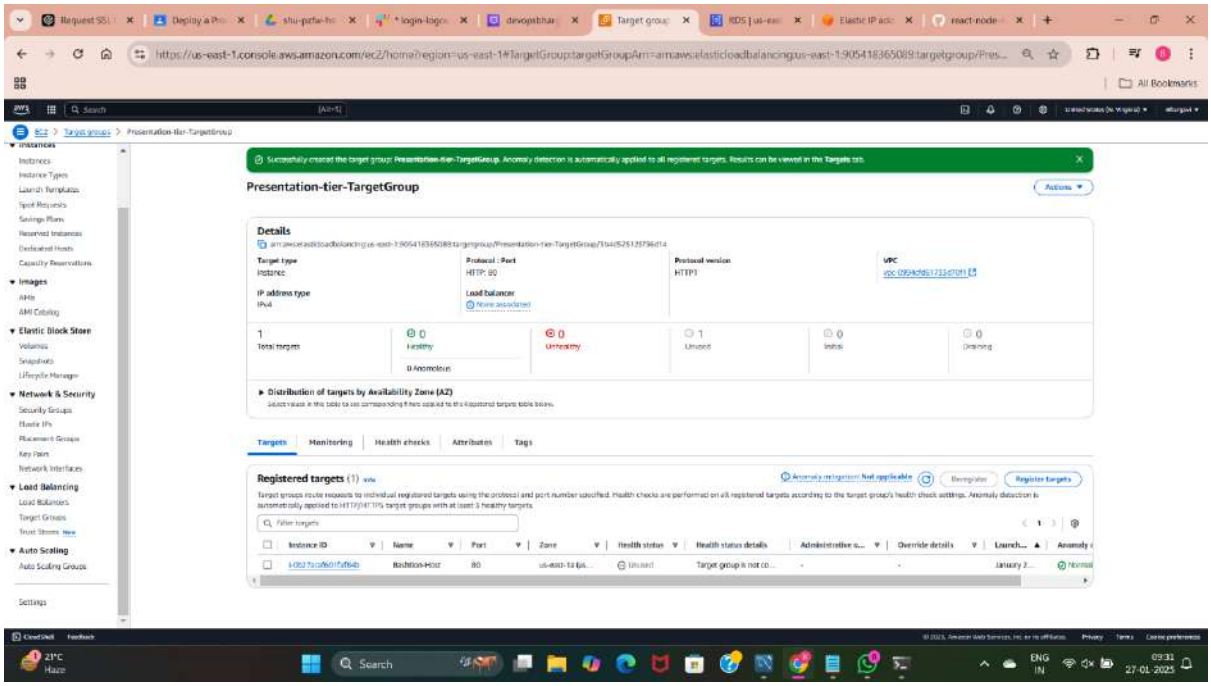


Provide the Target Group Details as mentioned below:

- Name: Presentation-Tier-target.
- Protocol: HTTP.
- Port: 80.
- VPC: Select the 3-tier VPC.



- In the Register Targets section, select the instances that will serve the Presentation Tier from the list of available instances (these are typically instances from the Presentation-Tier-EC2 Auto Scaling Group).
- Click Include as Pending.
- Click Create target group.



- ✓ Presentation tier target group is created.

LOAD BALANCER

A Load Balancer is a network appliance or software tool that distributes incoming network traffic across multiple backend servers (also known as targets) to ensure high availability and reliability of applications. The goal of load balancing is to:

- ❖ Distribute traffic evenly across servers to avoid overloading any single server.
- ❖ Increase scalability by automatically adding or removing servers based on demand.
- ❖ Improve availability by rerouting traffic to healthy servers if one goes down.

Types of Load Balancers in AWS

AWS offers multiple types of Load Balancers:

- ❖ Application Load Balancer (ALB)
- ❖ Network Load Balancer (NLB)
- ❖ Classic Load Balancer (CLB)
- ❖ Gateway Load Balancer (GWLB)

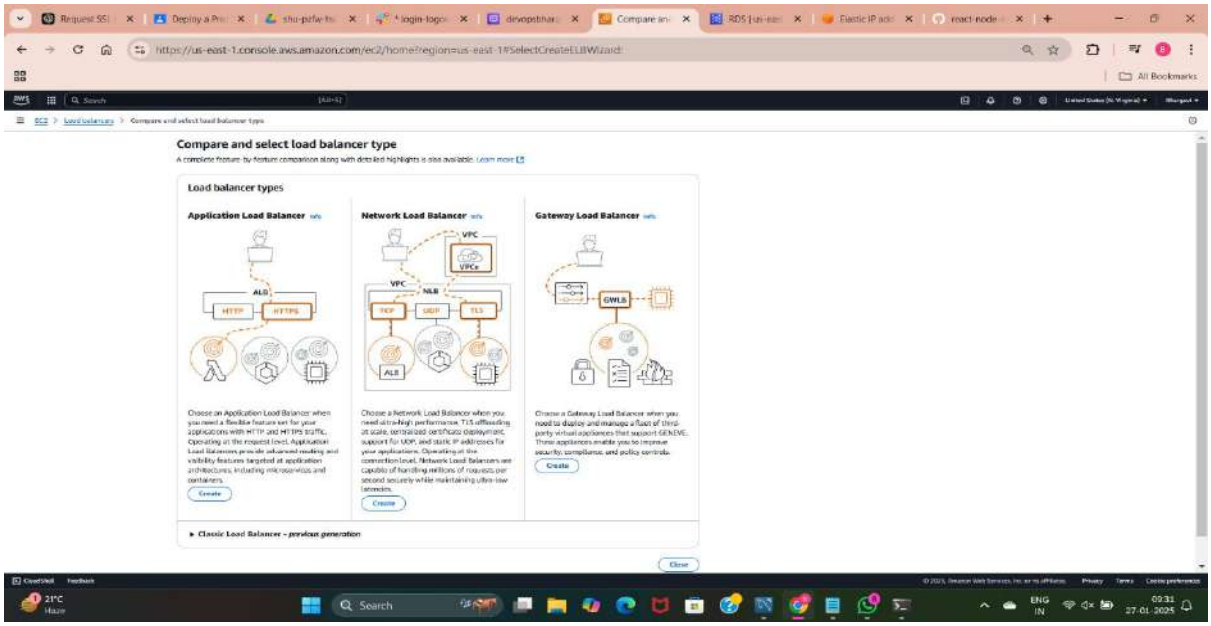
APPLICATION LOADBALANCER:

The Application Load Balancer (ALB) is a Layer 7 (Application Layer) load balancer that routes HTTP/HTTPS traffic based on content and advanced routing logic. It operates at the application layer of the OSI model and supports routing based on:

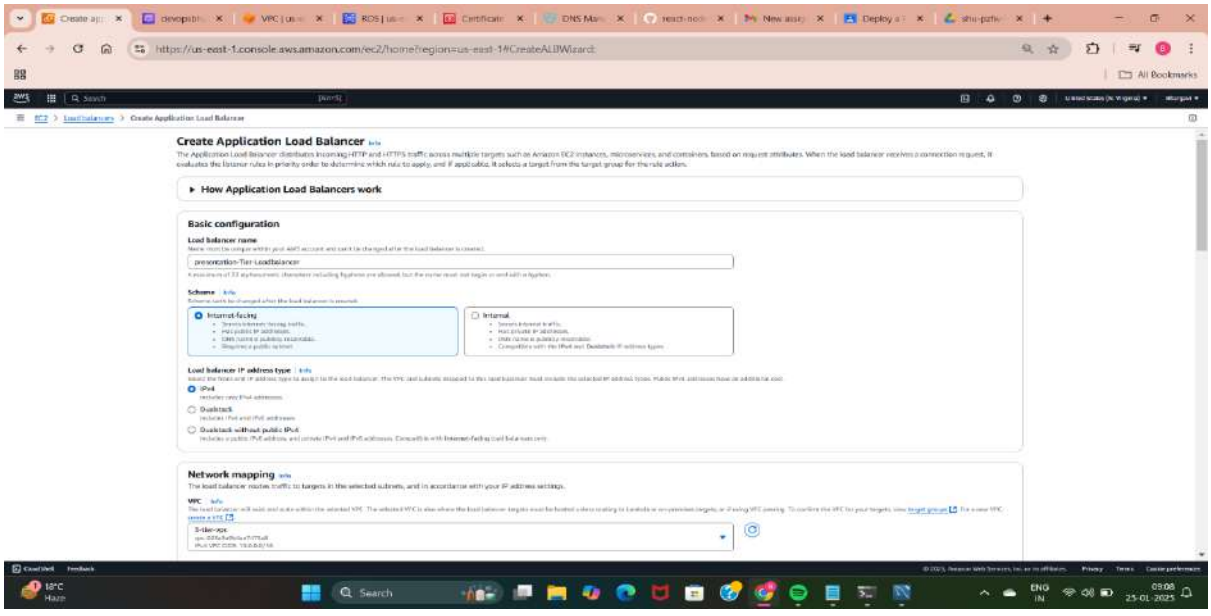
- URL path (e.g., /api/* routes to an API server)
- Host headers (e.g., devopsbhargavi.xyz or api.example.com)
- Query strings or HTTP headers

7.3 CREATING PRESENTATION TIER LOAD BALANCER:

- ❖ Go to the EC2 Dashboard in the AWS Management Console.
- ❖ Under Load Balancing, click on Load Balancers.
- ❖ Click Create Load Balancer.

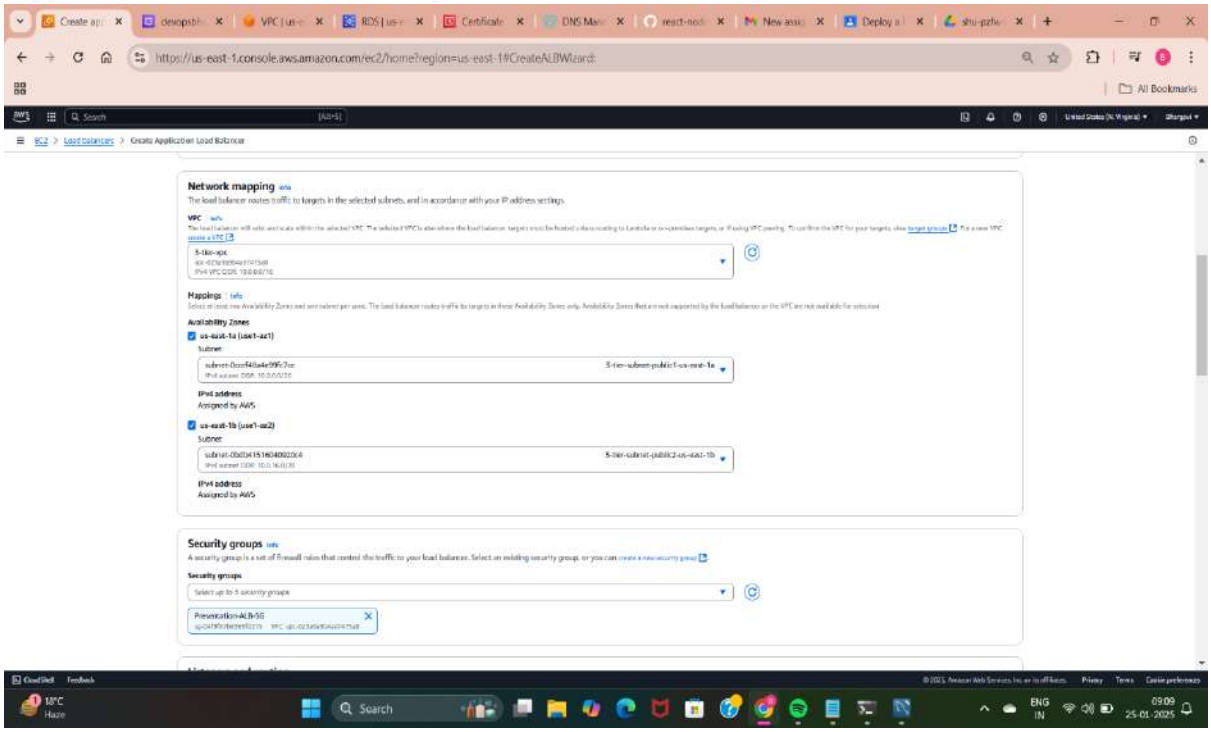


- ❖ Choose the loadbalancer type as Application Load Balancer (ALB).
- ❖ Click Create.

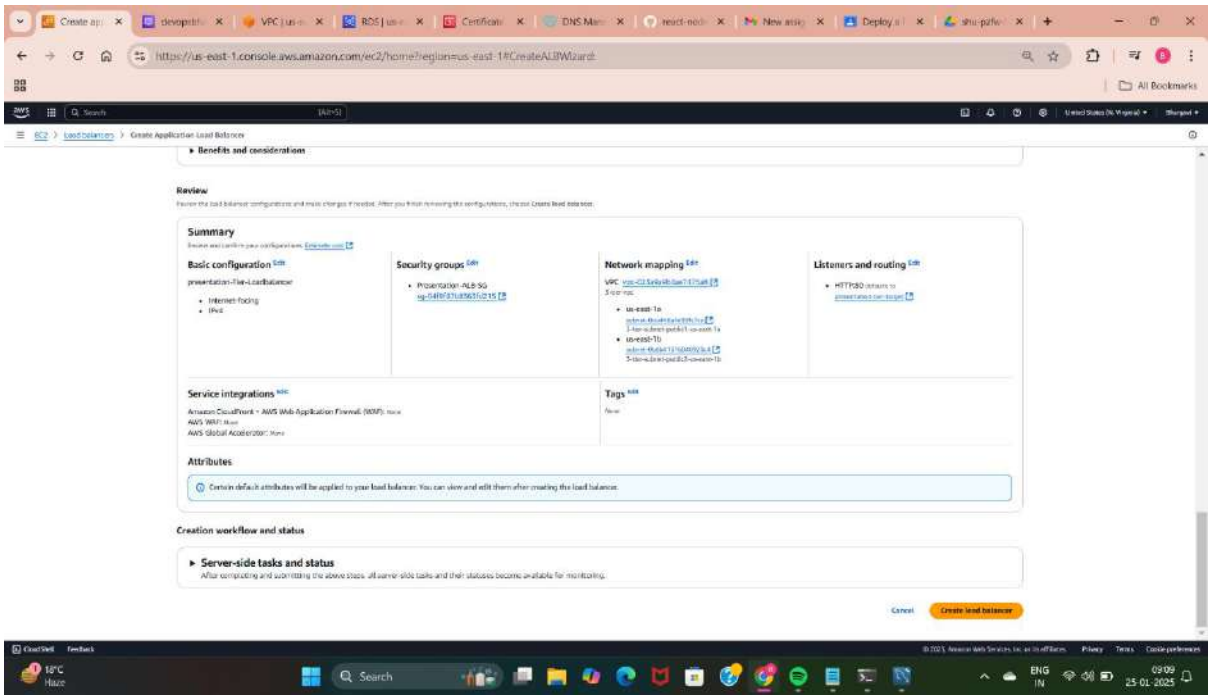


Configure Load Balancer Basics

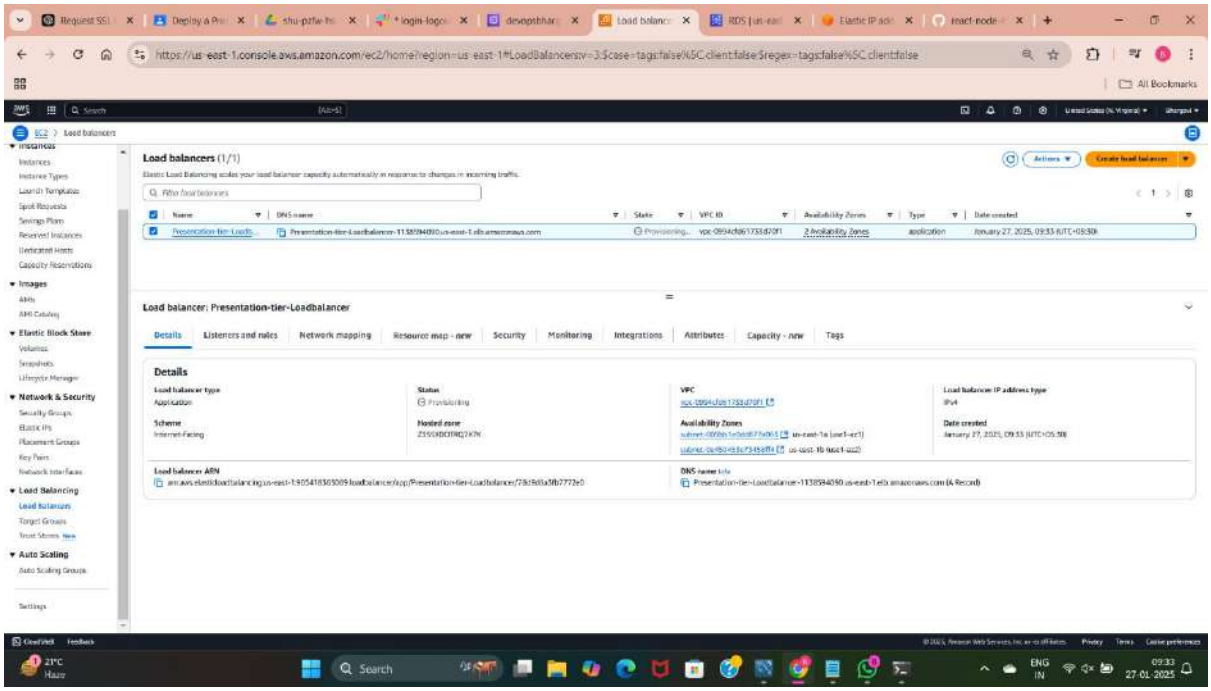
1. Name: Presentation-Tier-Loadbalancer.
2. Scheme: Select Internet-facing.
3. IP Address Type: IPv4.
4. Listeners:
    - Protocol: HTTP.
    - Port: 80.



- Select the 3-tier VPC.
- Select Public Subnets (ensure you select the same subnets configured earlier).
- Select the Presentation-Tier-ALB security group (created in Step 4.2).
- Under Default Routing, choose Existing Target Group.
- Select the Presentation-Tier-target group (created in Step 7.2).



- Review the configuration.
- Click Create Load Balancer.



- Presentation tier load balancer is created.

AUTO SCALING GROUP (ASG)

An Auto Scaling Group (ASG) is a service in AWS that automatically adjusts the number of Amazon EC2 instances in your application's fleet based on demand. This helps to ensure that your application has the appropriate amount of compute resources to handle varying levels of traffic, while also controlling costs by scaling down when demand is low.

The Auto Scaling Group works in tandem with the Auto Scaling Policies to manage the lifecycle of EC2 instances, ensuring that you have the right number of instances running based on metrics like CPU utilization, memory usage, or network traffic.

Automatic Scaling:

Adds new EC2 instances when demand increases (e.g., when traffic spikes). Removes EC2 instances when demand decreases (e.g., during periods of low traffic).

Elasticity:

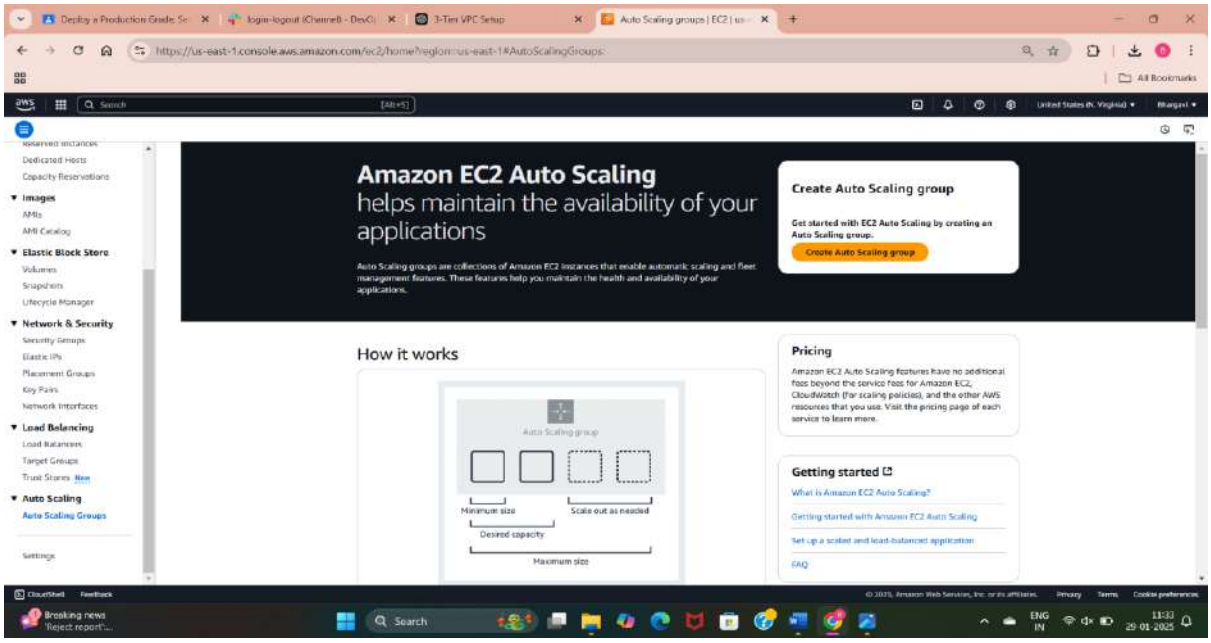
The number of instances in the group can increase or decrease automatically without manual intervention. You define the desired capacity (the ideal number of instances) and the minimum and maximum limits for scaling.

Launch Configurations/Templates:

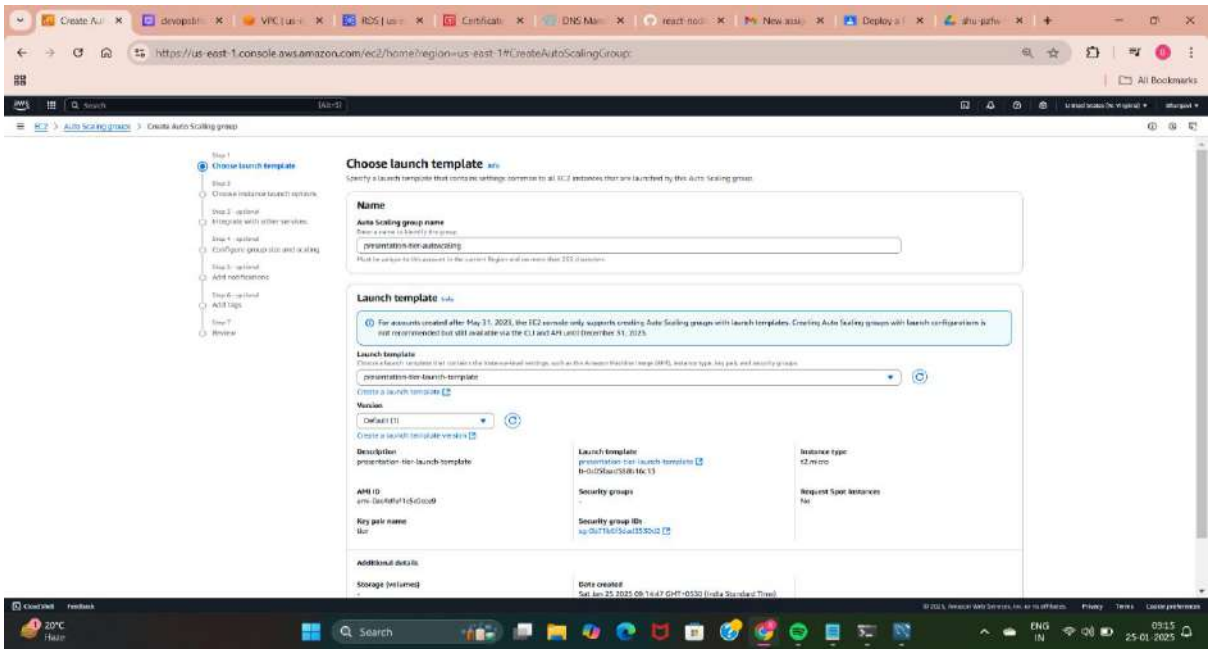
You define a Launch Configuration or Launch Template which specifies the EC2 instance type, AMI, key pair, security groups, and other settings to be used for launching new instances in the group.

7.4 CREATING PRESENTATION TIER AUTO SCALING GROUP WITH DESIRED CAPACITY=3, MIN=2, MAX=4

- ✓ Open the EC2 Dashboard.
- ✓ Scroll down to the Auto Scaling section.
- ✓ Click on Auto Scaling Groups.



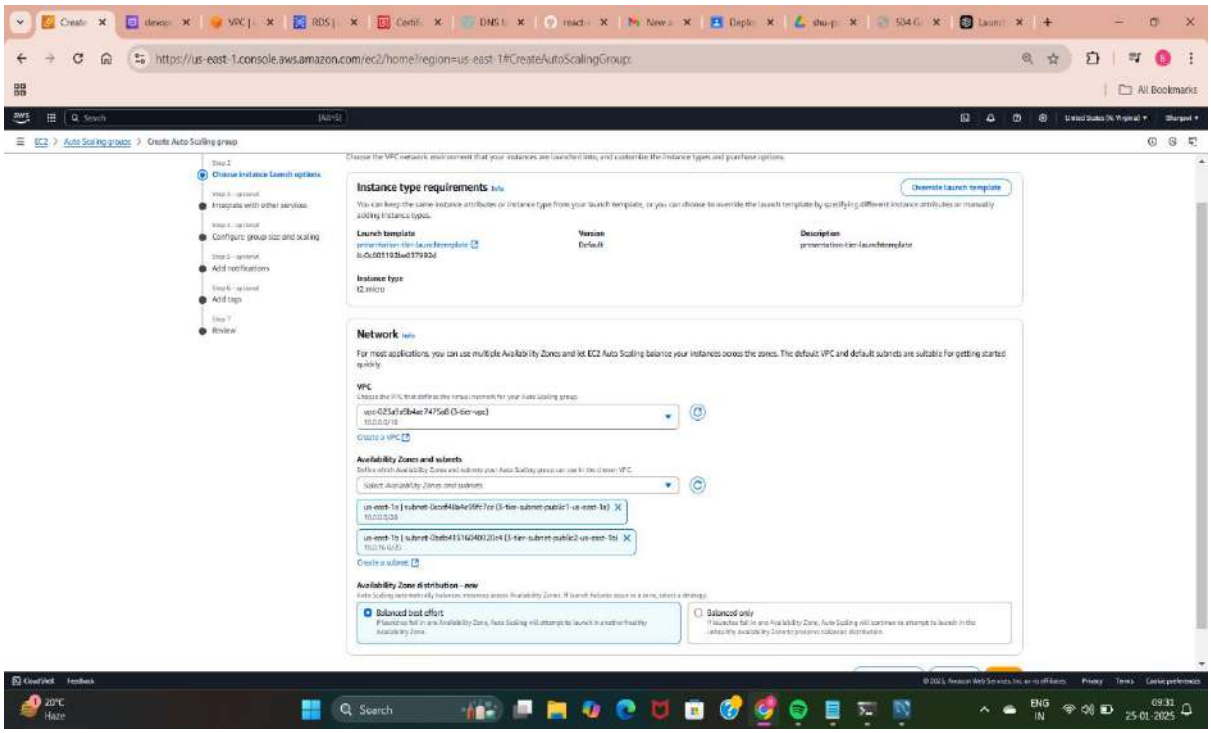
- ✓ Click Create Auto Scaling Group.



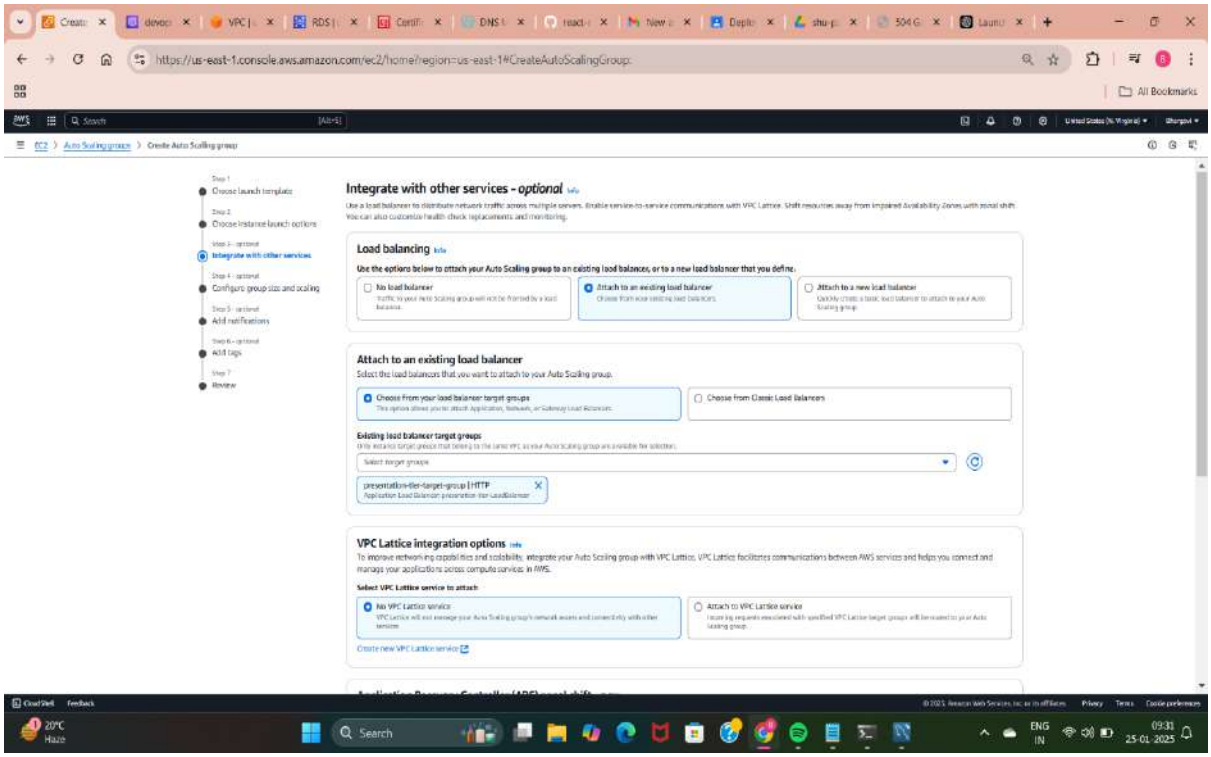
Configure Basic Details

- ✓ Auto Scaling Group Name: Presentation-Tier-autoscaling.
- ✓ Launch Template: Select the Launch Template created earlier for the Presentation Tier (Step 7.1).

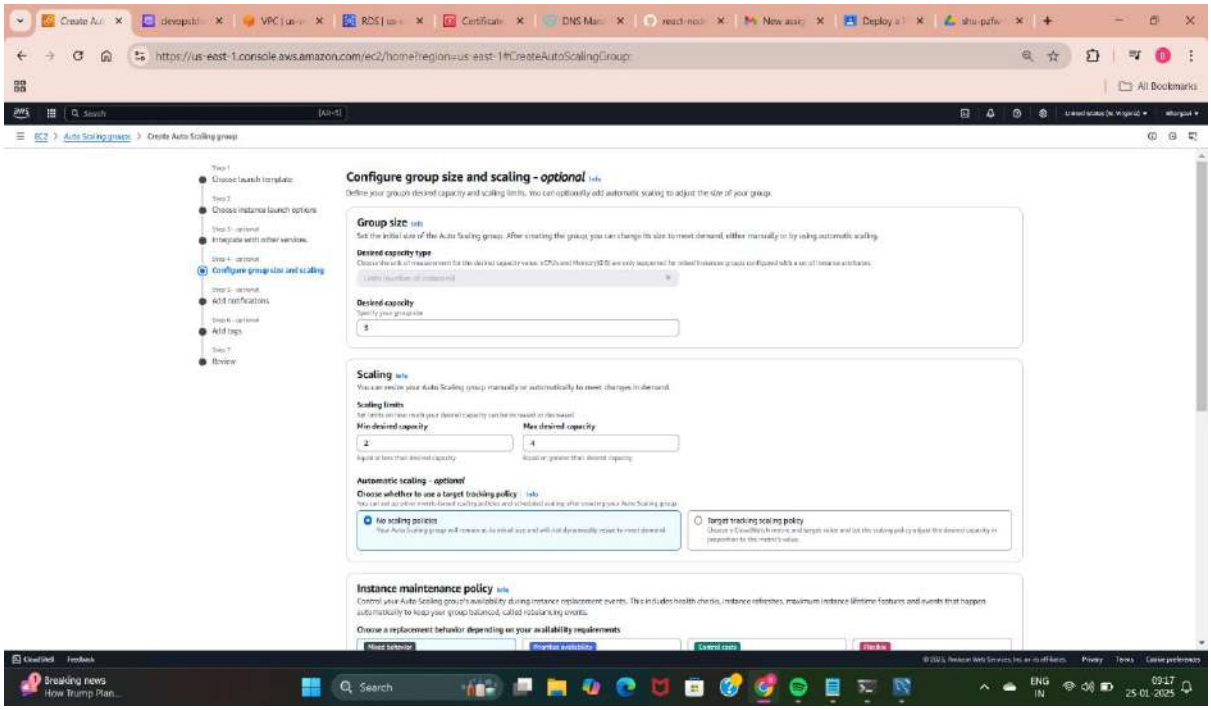
- ✓ Select the version of launch template.
- ✓ Click Next.



- ✓ Select the 3-Tier-Architecture VPC and select the Public Subnets from your VPC.

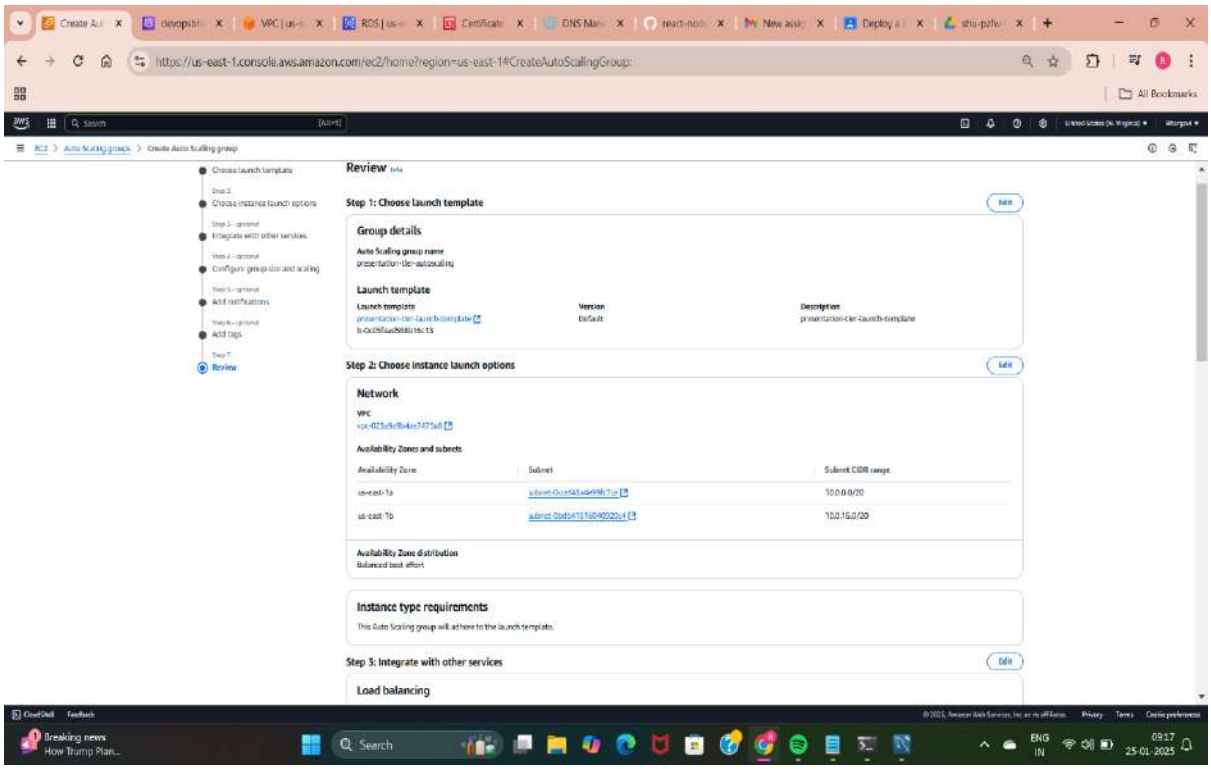


- ✓ In the Load Balancing section select Application Load Balancer.
- ✓ Choose the Presentation-Tier-TG Target Group created in Step 7.2.

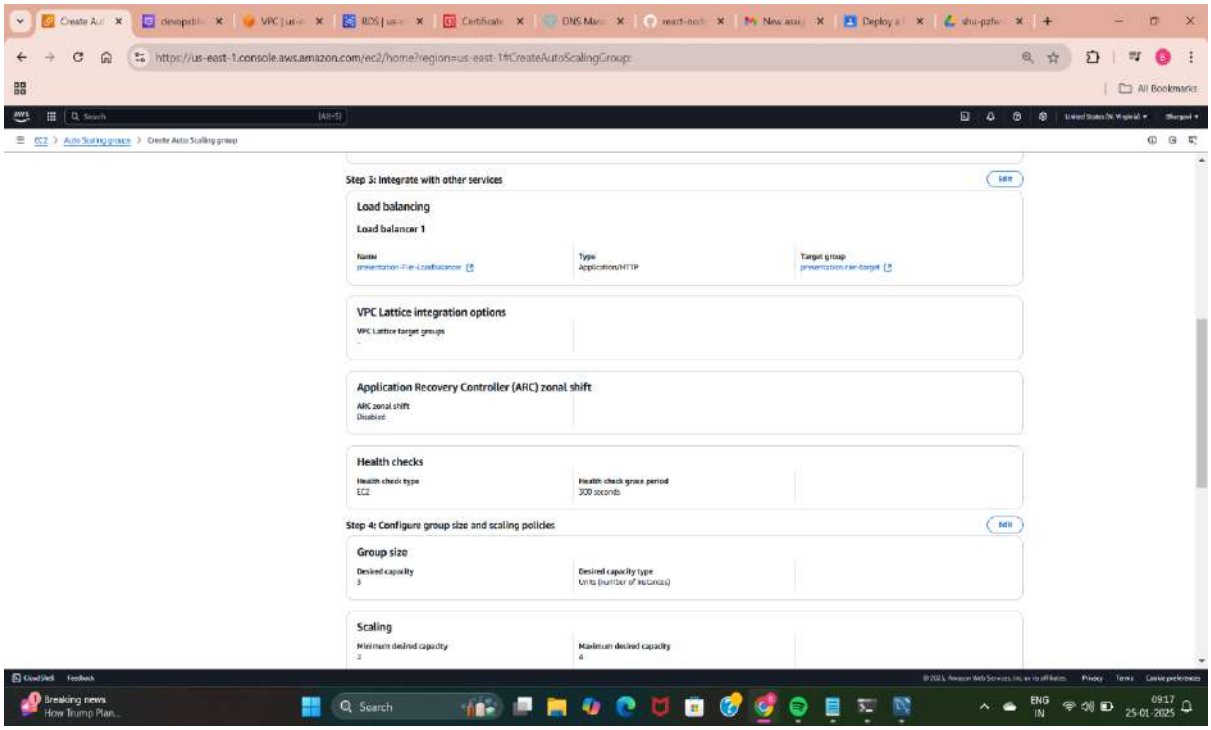


Configure Scaling Policies

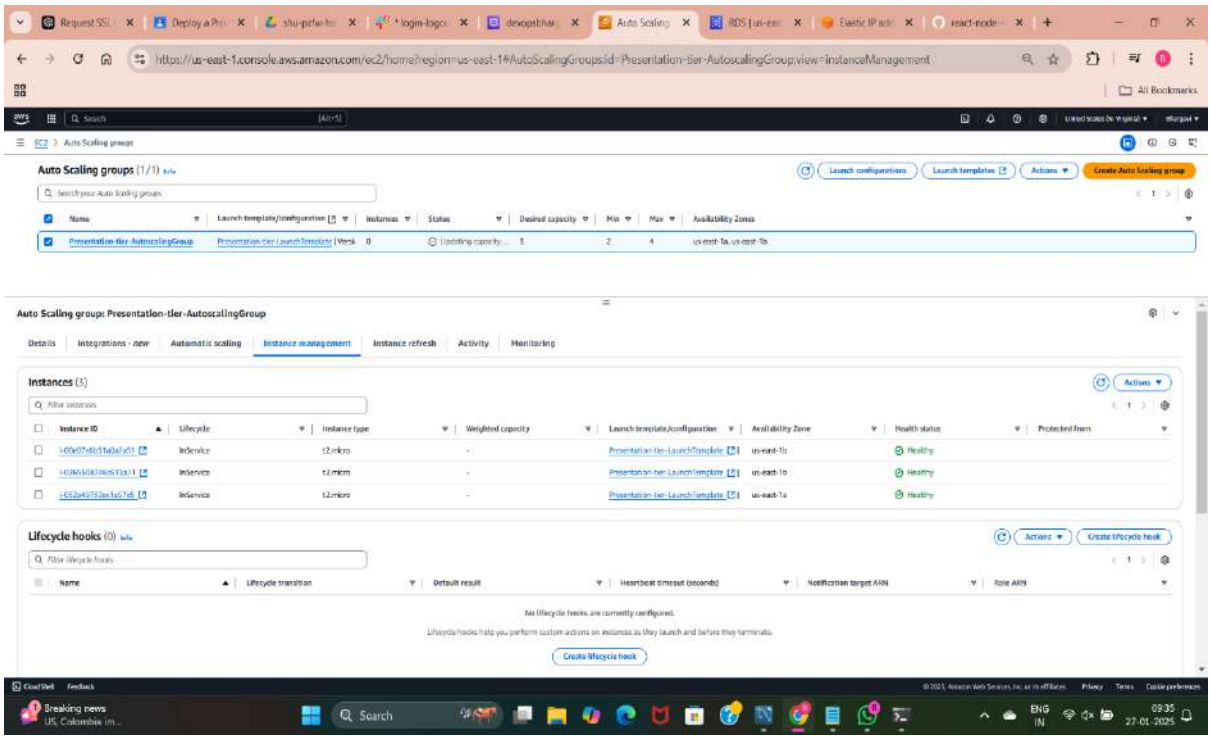
1. Desired Capacity: 3
2. Minimum Capacity: 2
3. Maximum Capacity: 4



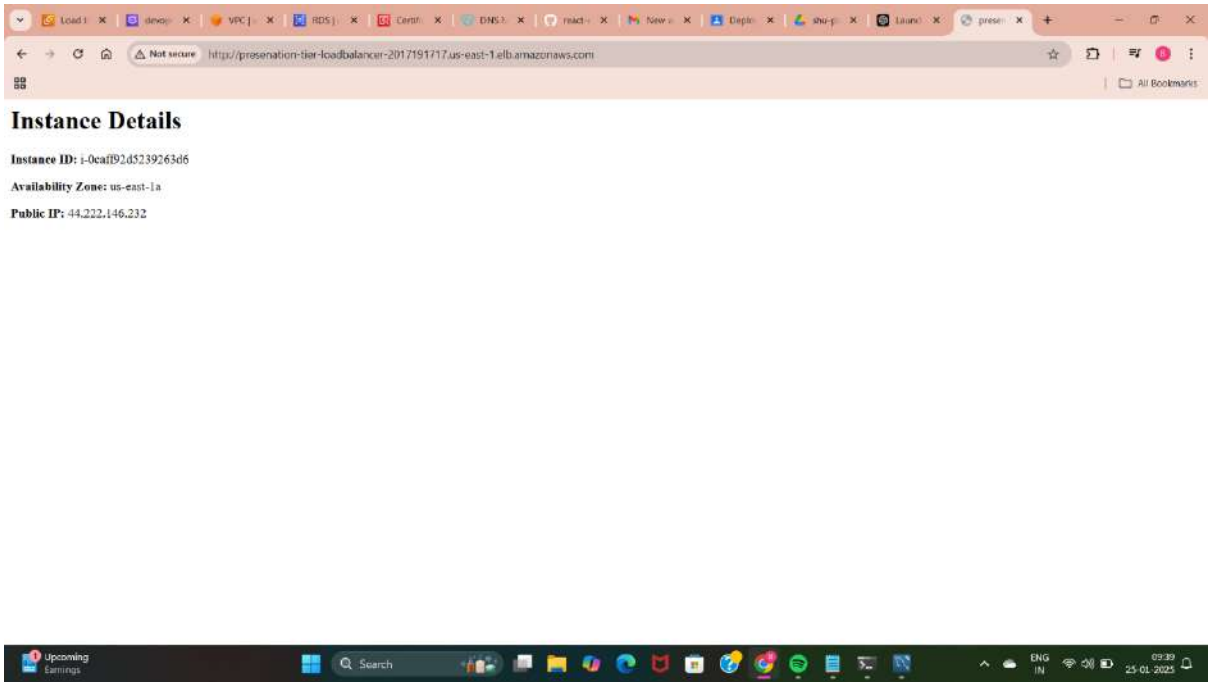
✓ Review the configuration.



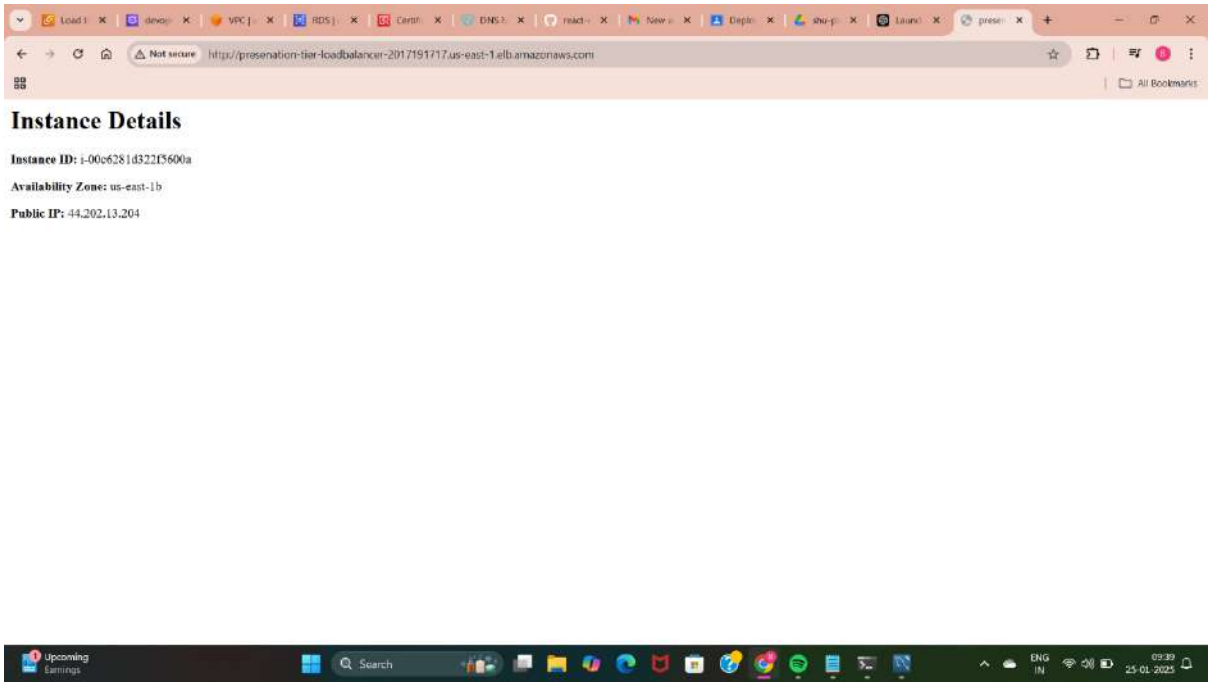
✓ Click Create Auto Scaling Group.



✓ Autoscaling group is created for presentation tier.



- ✓ Access the DNS name of the load balancer in your browser to confirm it works as expected.



- ✓ If the Instance Metadata Details are visible when accessing the DNS name, everything is set up perfectly.

APPLICATION TIER

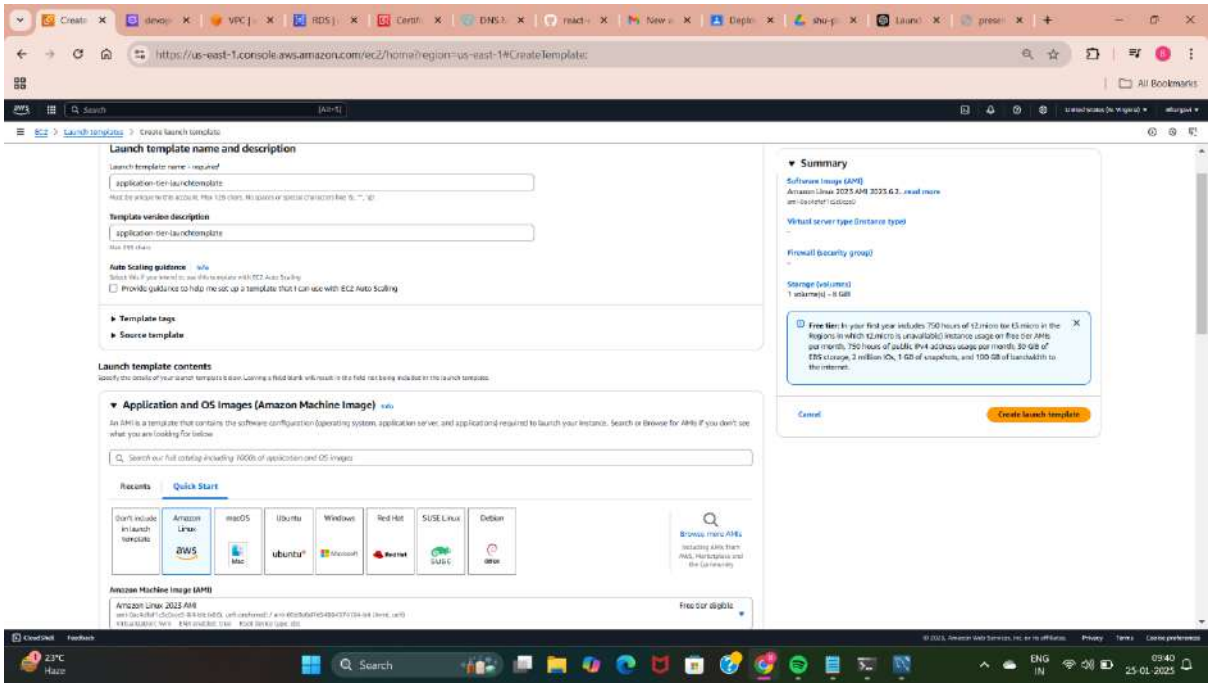
The Application Tier is one of the layers in a multi-tier application architecture, commonly used in the design of complex, scalable, and maintainable systems. Data Tier (Database)

The Application Tier is also known as the Middle Tier or Business Logic Layer. It contains the core functionality of the application and is responsible for processing the business logic, handling requests, and making decisions based on user input.

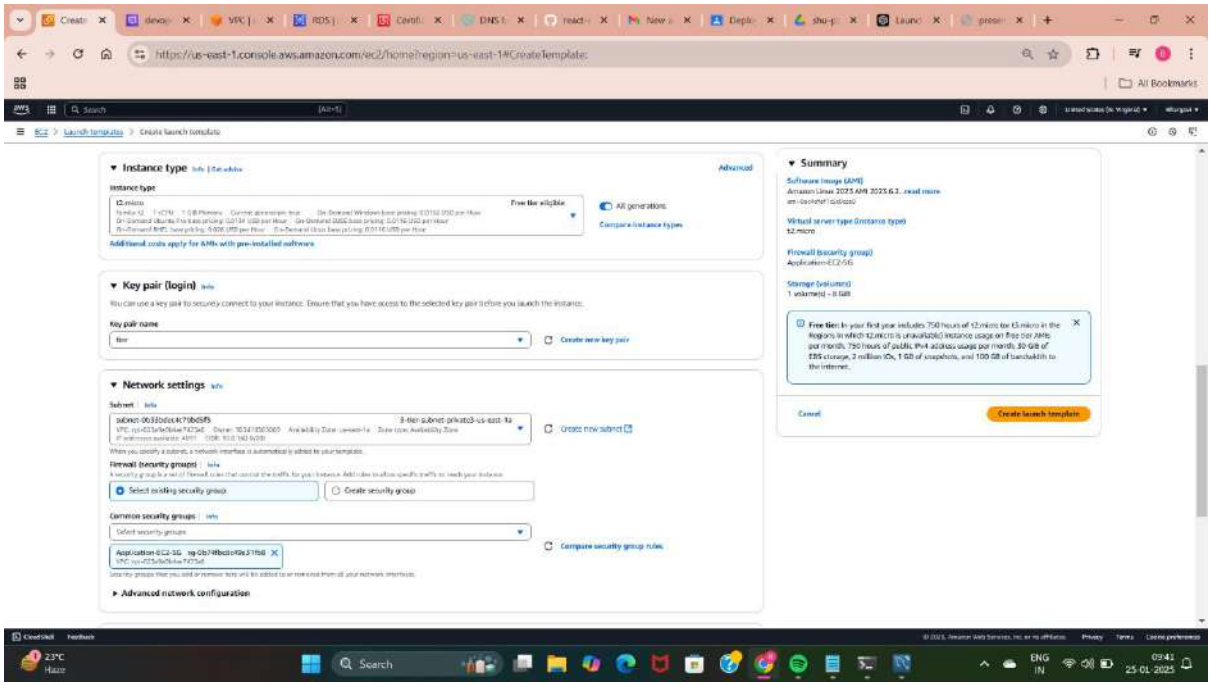
STEP 8: SETUP APPLICATION TIER

8.1 CREATE LAUNCH TEMPLATE WITH THE USER DATA AS FOLLOWS

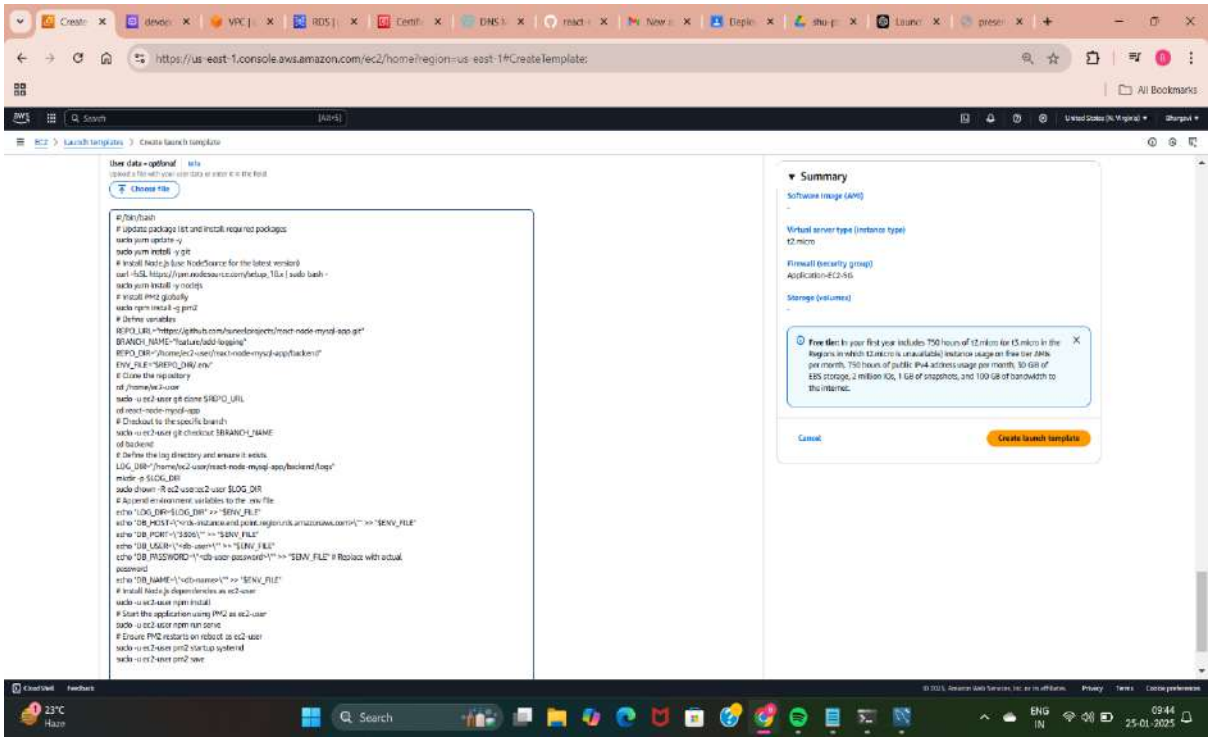
☑ Go to EC2 Dashboard → Launch Templates → Create Launch Template.



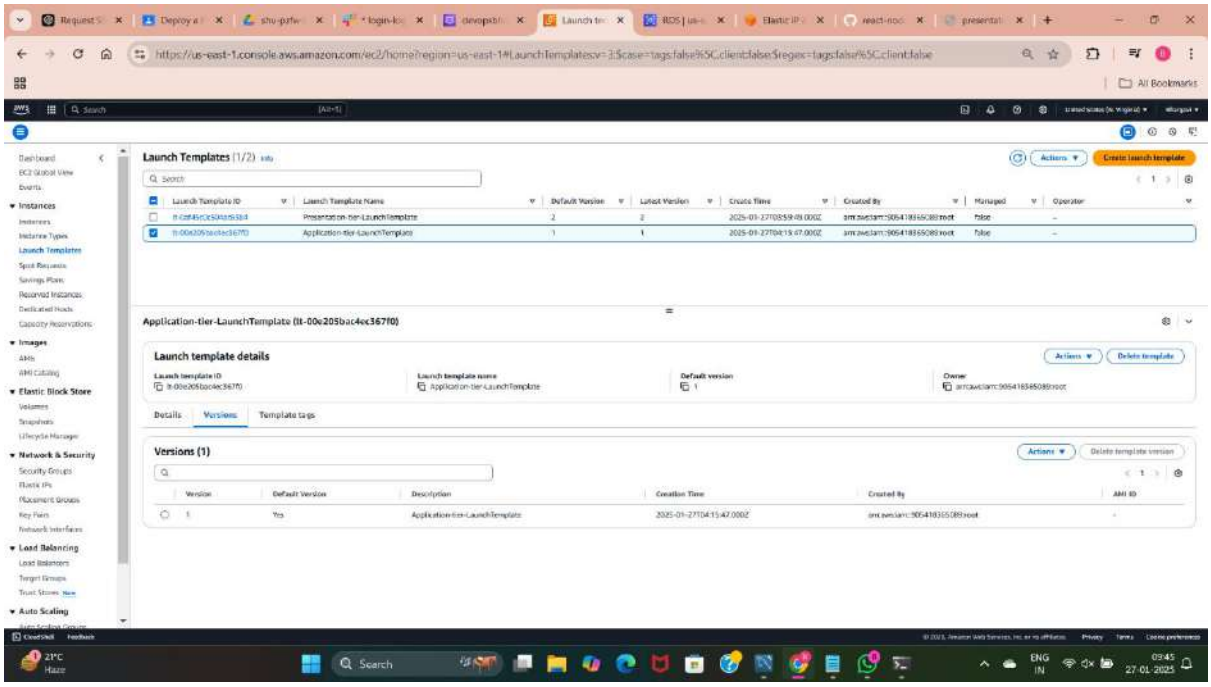
☑ Template Name: application-tier-launchtemplate.
☑ AMI: Select Amazon Linux 2023 AMI or the appropriate AMI for your application.



☑ Instance Type: Use t2.micro or an instance type suitable for your application.
☑ Key Pair: Select your key pair (tier.pem).
☑ Select the 3-Tier-Architecture VPC.
☑ Ensure instances will be launched in private subnets.
☑ Assign the Application-Tier-EC2 Security Group created in Step 4.5.



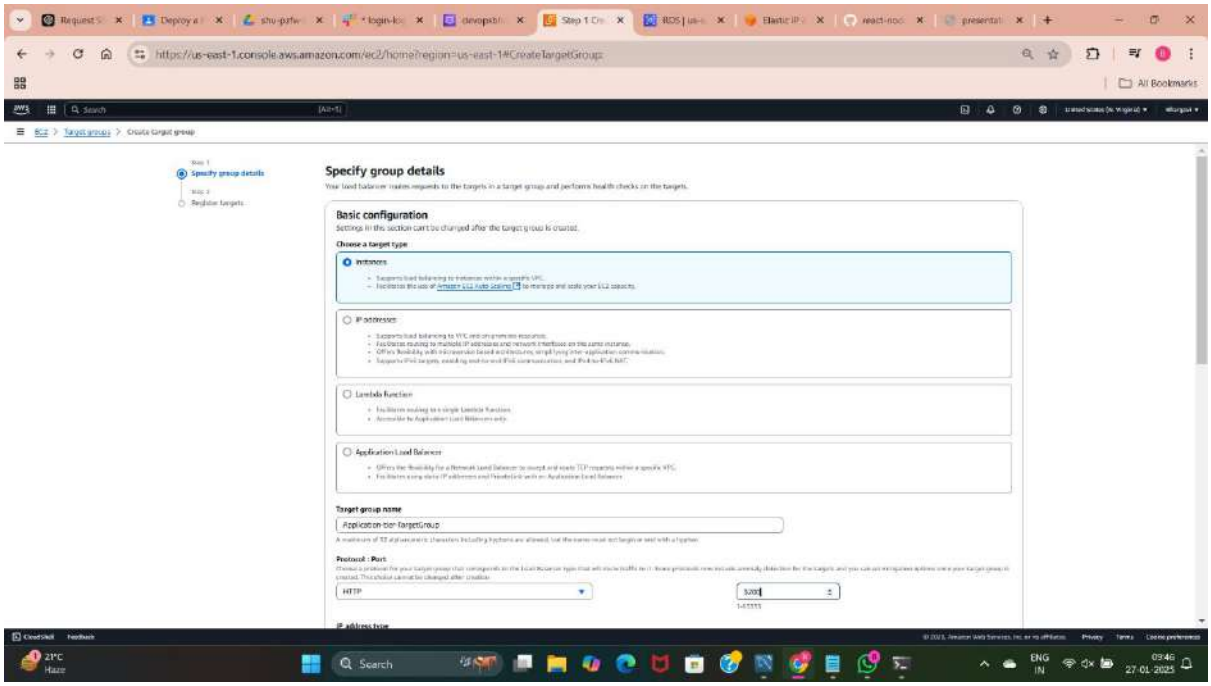
☑ Add user data to configure the application on launch (e.g., installing dependencies, deploying the app).



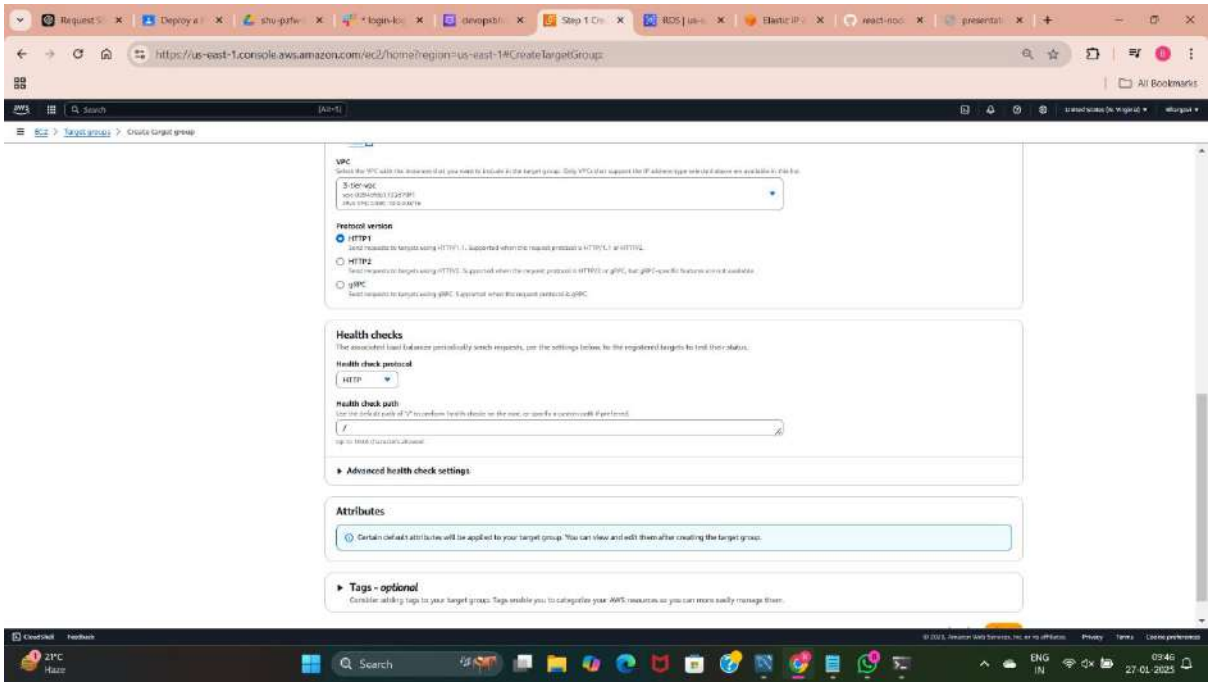
☑ Launch template for application tier is created.

8.2 CREATE TARGET GROUP

- ☑ In the AWS Management Console, navigate to the EC2 Dashboard.
- ☑ In the left sidebar, under Load Balancing, click on Target Groups.
- ☑ Click on Create target group.

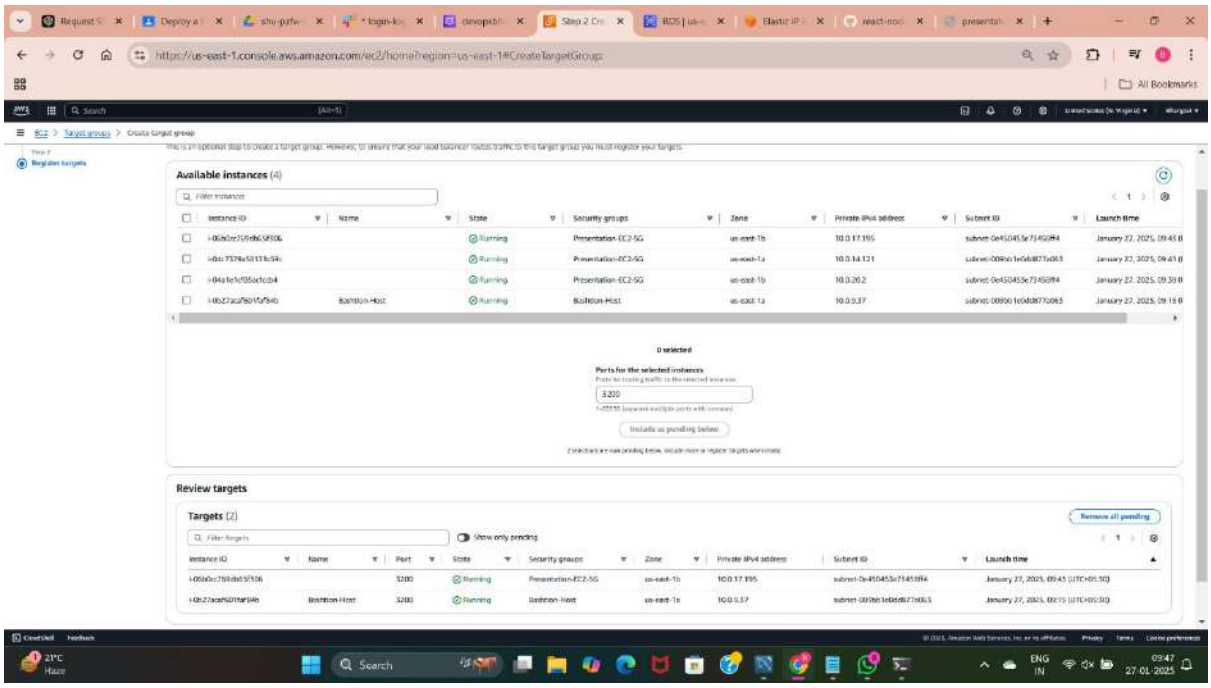


- ☑ Target Type: Choose Instances (since you'll be registering EC2 instances in this target group).

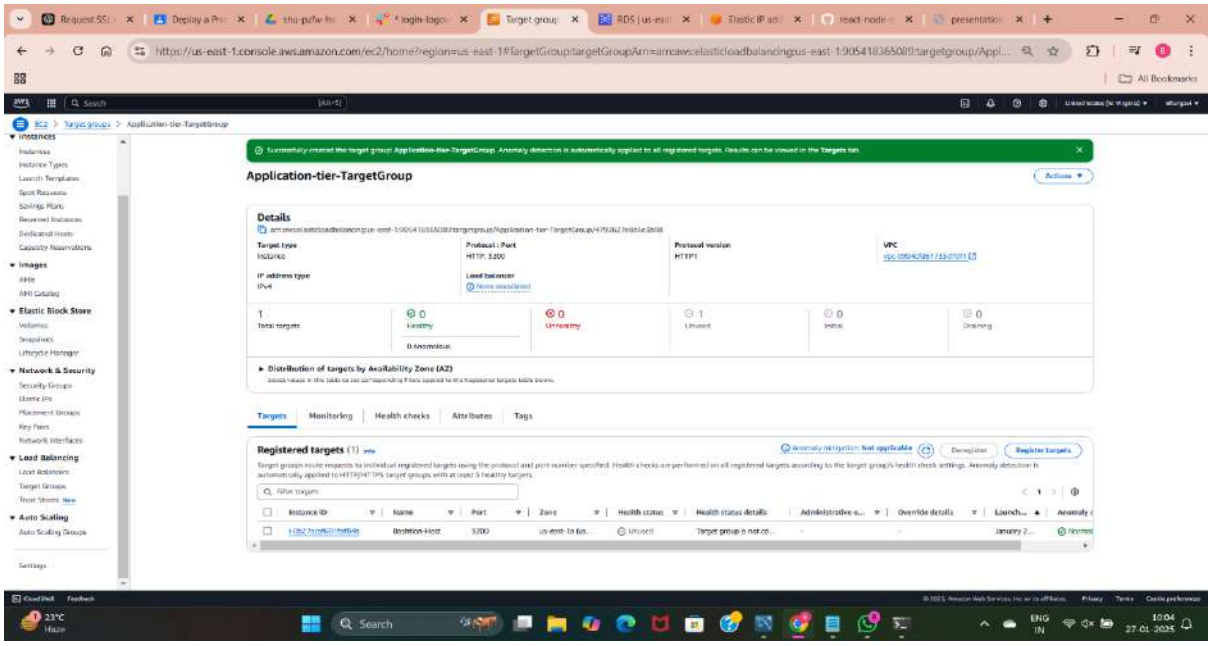


- ☑ Name: Enter a name, such as Application-Tier-TG\argetGroup.
- ☑ Protocol: Select HTTP.
- ☑ Set the port to 3200 (the port where the Application Tier will listen).
- ☑ Select 3-Tier-Architecture VPC.

☑ Health checks:
Protocol: HTTP.
Path: / or choose a specific health check endpoint if applicable.
Port: Set it to traffic port.
☑ Click Next.



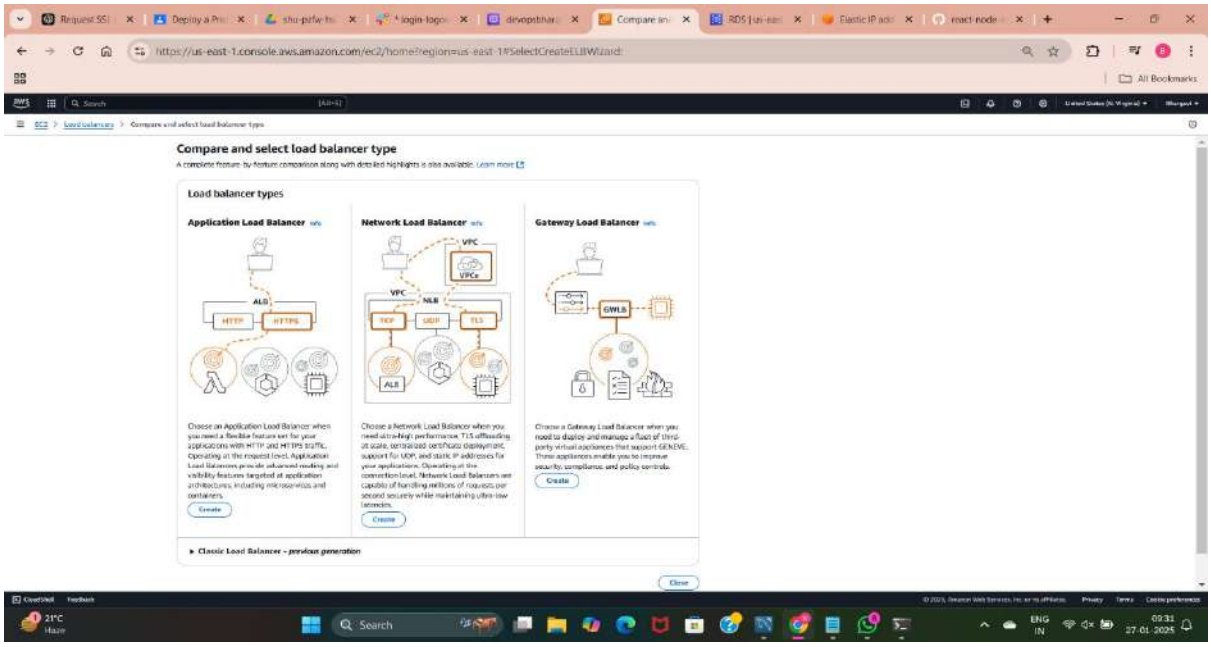
☑ After creating the target group, you can register your Application Tier EC2 instances.
☑ Select Add to registered targets.
☑ Click Create to finish the process.



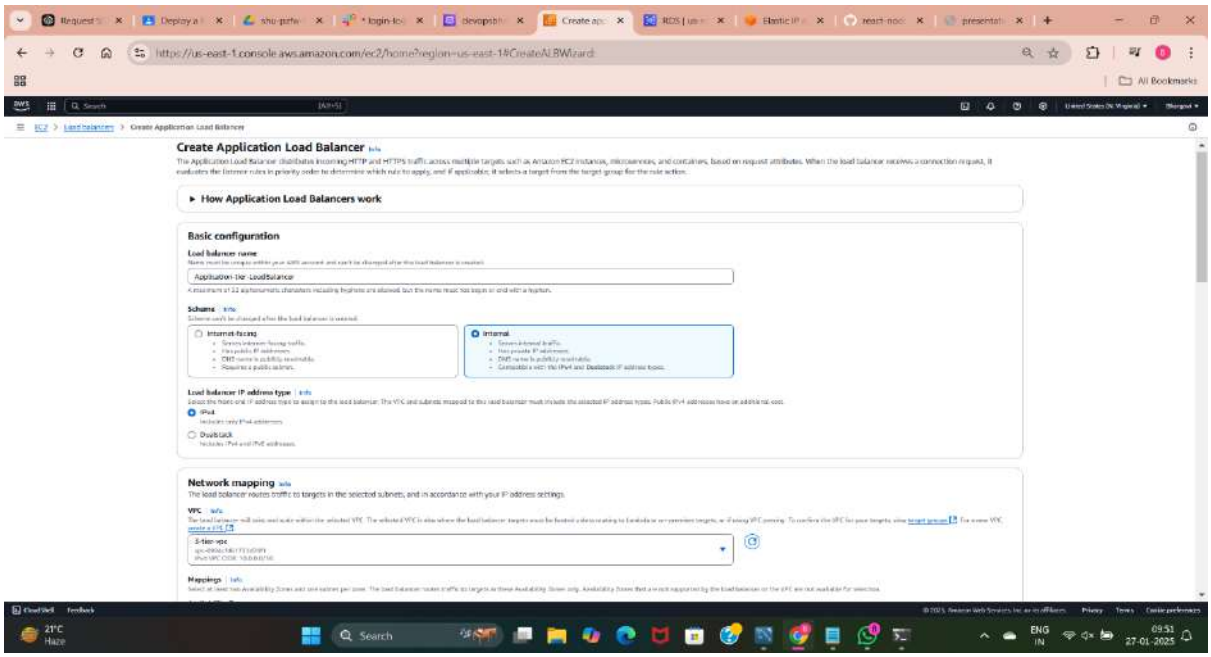
☑ Application tier target group is created. This will create the target group, and we can now attach it to our load balancer and configure scaling policies as needed.

8.3 CREATING APPLICATION TIER LOAD BALANCER

- ☑ In the AWS Management Console, navigate to the EC2 Dashboard.
- ☑ In the left sidebar, under Load Balancing, click on Load balancers.
- ☑ Click on Create Load Balancer.

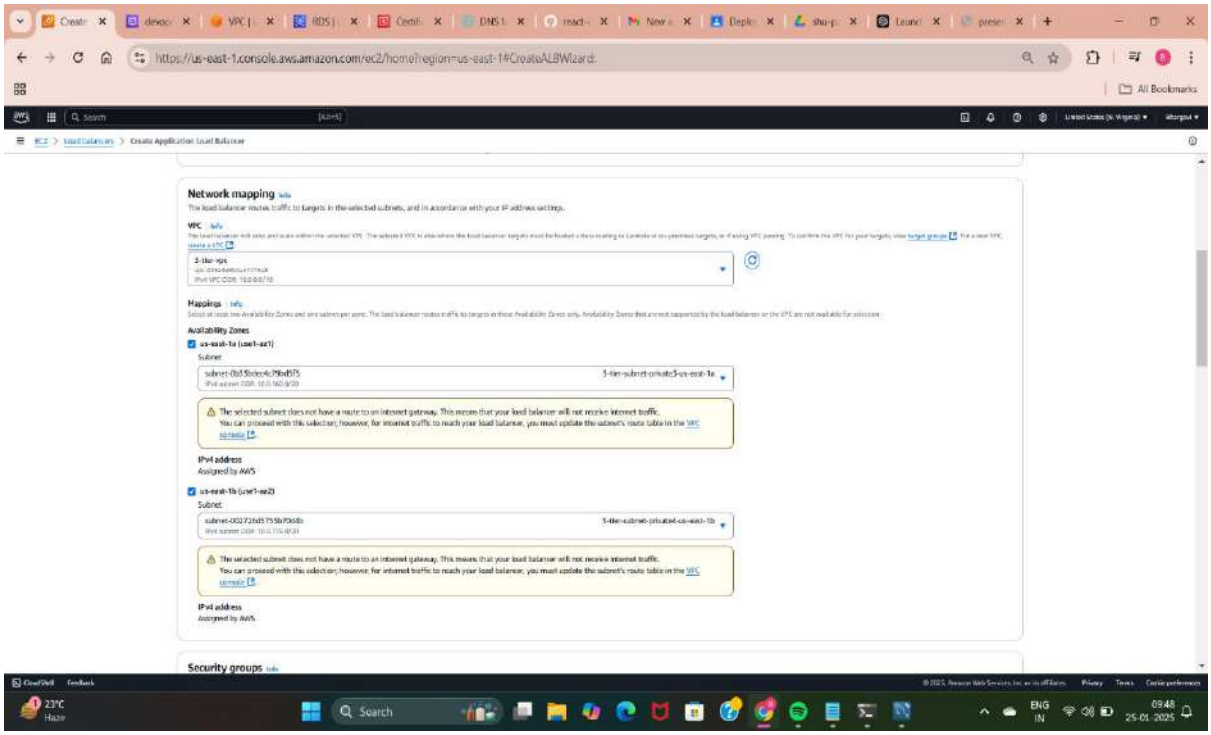


- ☑ Select Application Load Balancer (ALB).

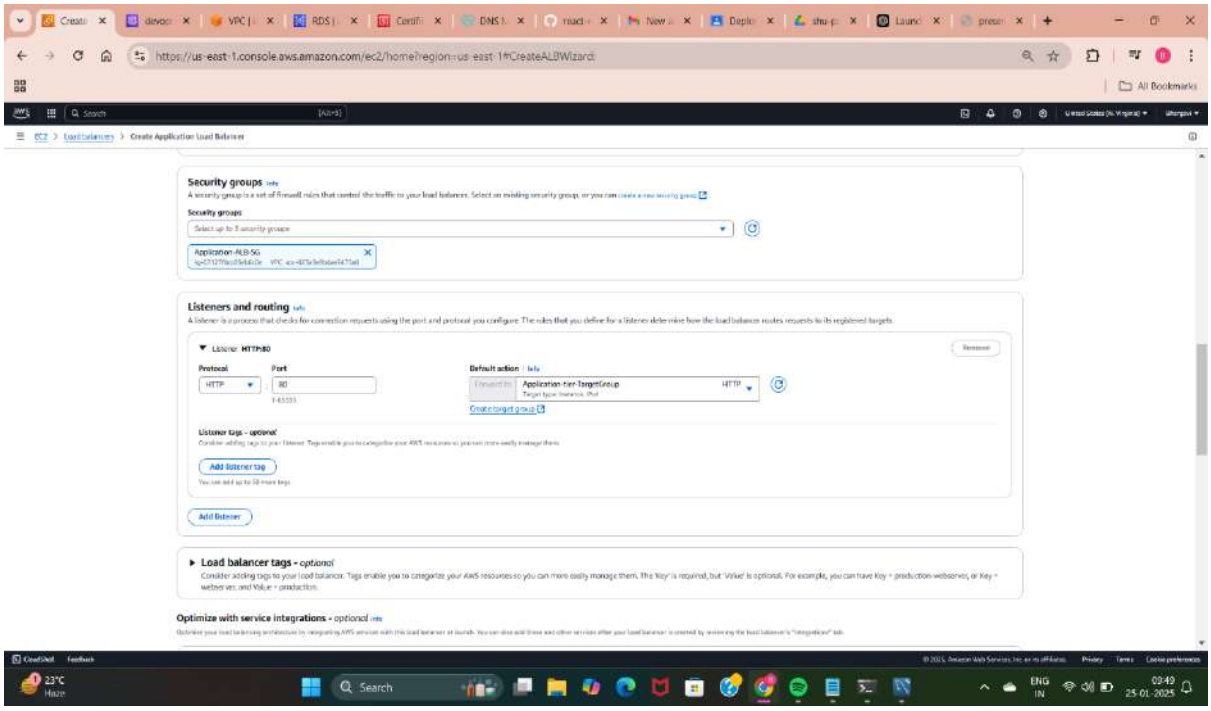


Configure Load Balancer Details:

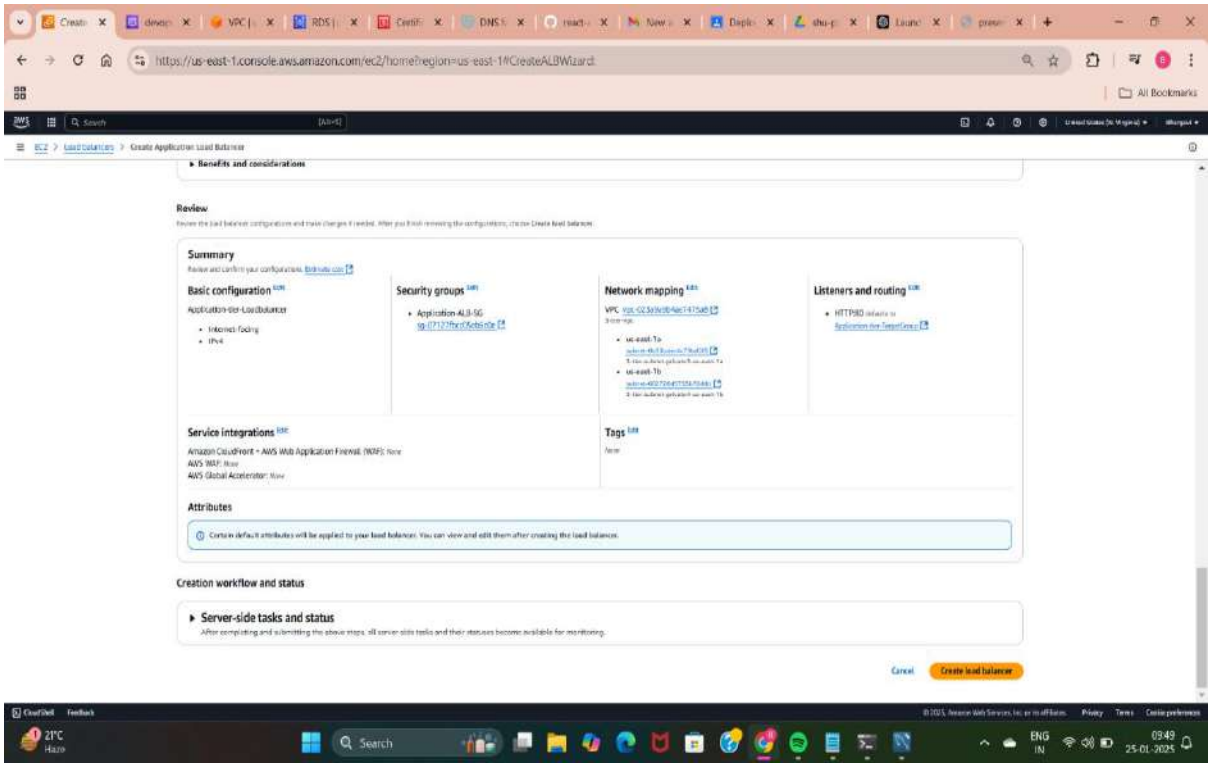
- ☑ Name: Enter a name, such as Application-Tier-LoadBalancer.
- ☑ Scheme: Choose internal based on your requirements.
- ☑ IP Address Type: Choose ipv4.



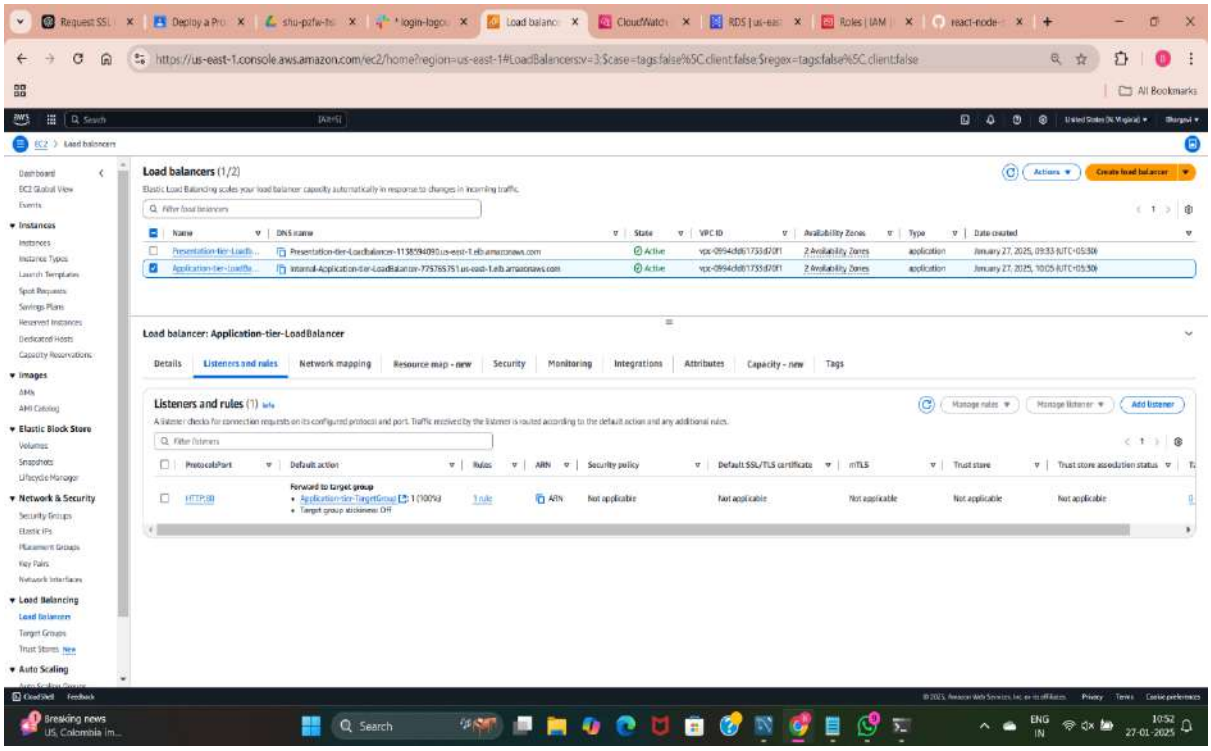
- ☑ Select 3-tier VPC.
- ☑ Select at least 2 availability zones from the region.
- ☑ Select two private subnets.



- ☑ Select the Application-Tier-ALB security group that you previously created.
- ☑ Listeners: By default, it should have an HTTP listener on port 80. You can leave this as it is or configure custom ports.



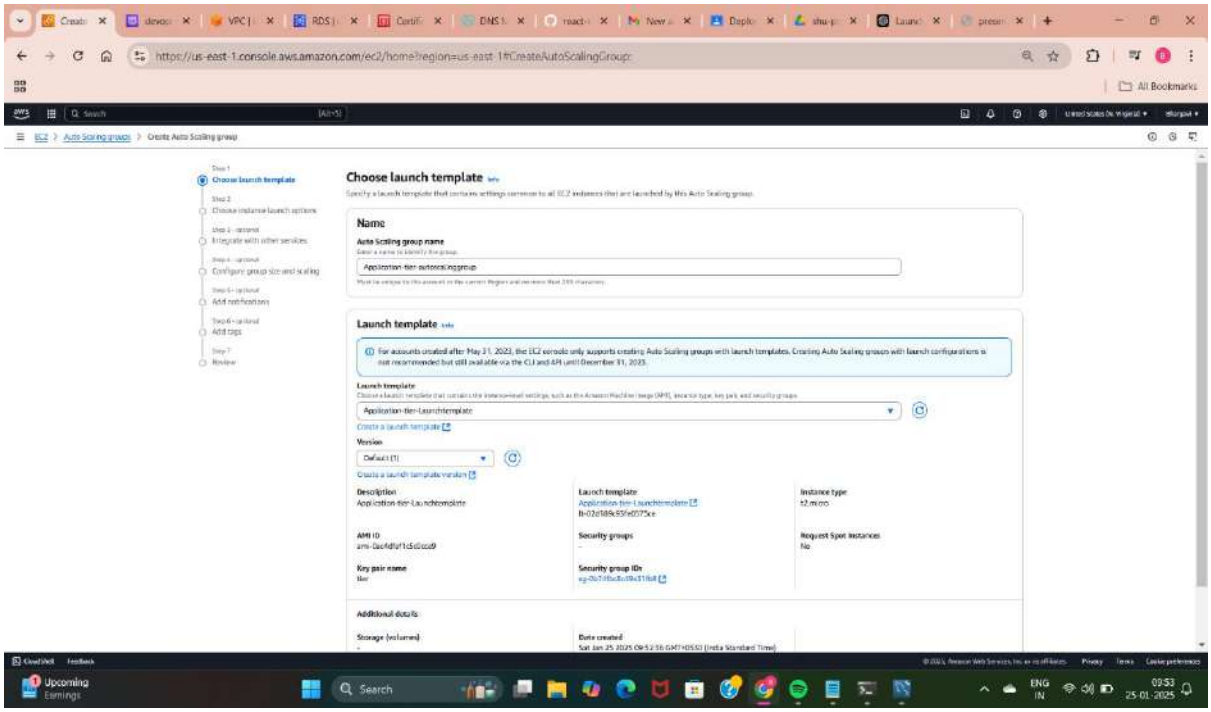
☑ Review all the configurations. Click Create to launch the Application Load Balancer.



☑ The Application Load Balancer is created for our Application Tier, and it will start routing traffic to the EC2 instances registered in the target group.

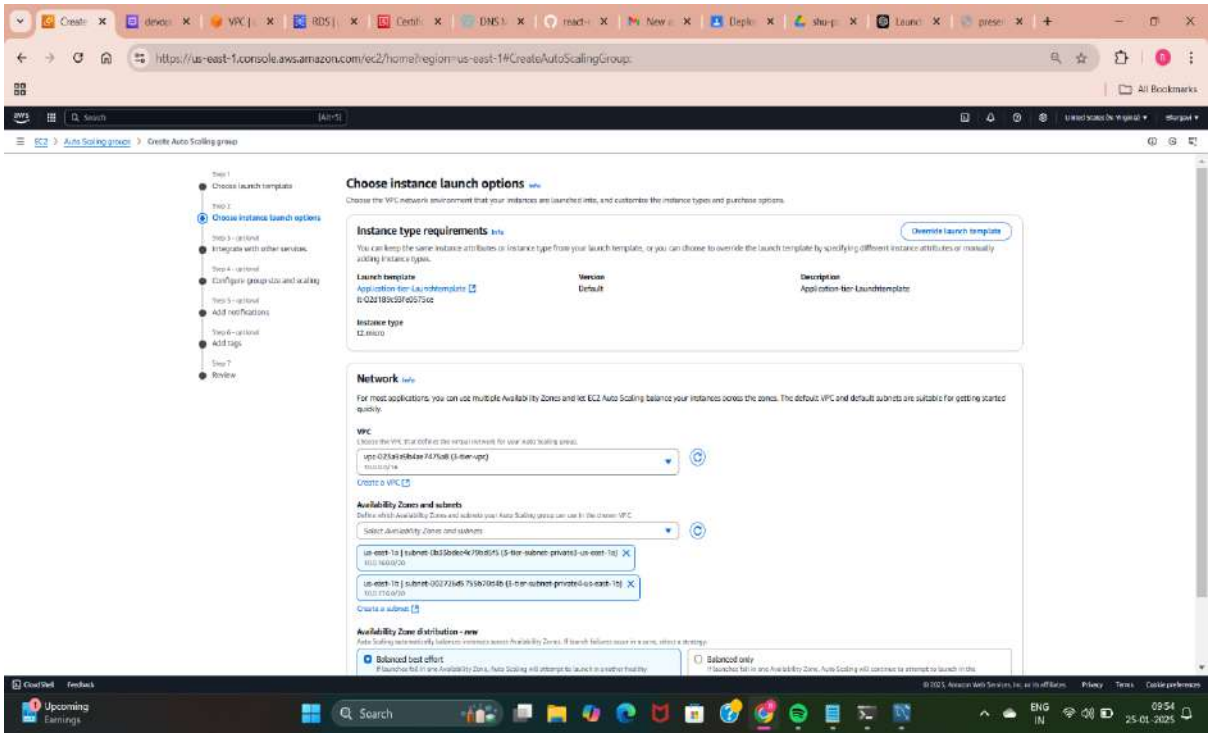
8.4 CREATING AUTO SCALING GROUP FOR APPLICATION TIER WITH DESIRED CAPACITY=3, MINIMUM=2, MAXIMUM=4

- ☑ In the AWS Management Console, navigate to EC2 Dashboard.
- ☑ In the left-hand sidebar, under Auto Scaling, click on Auto Scaling Groups.
- ☑ Click on the Create Auto Scaling group button.

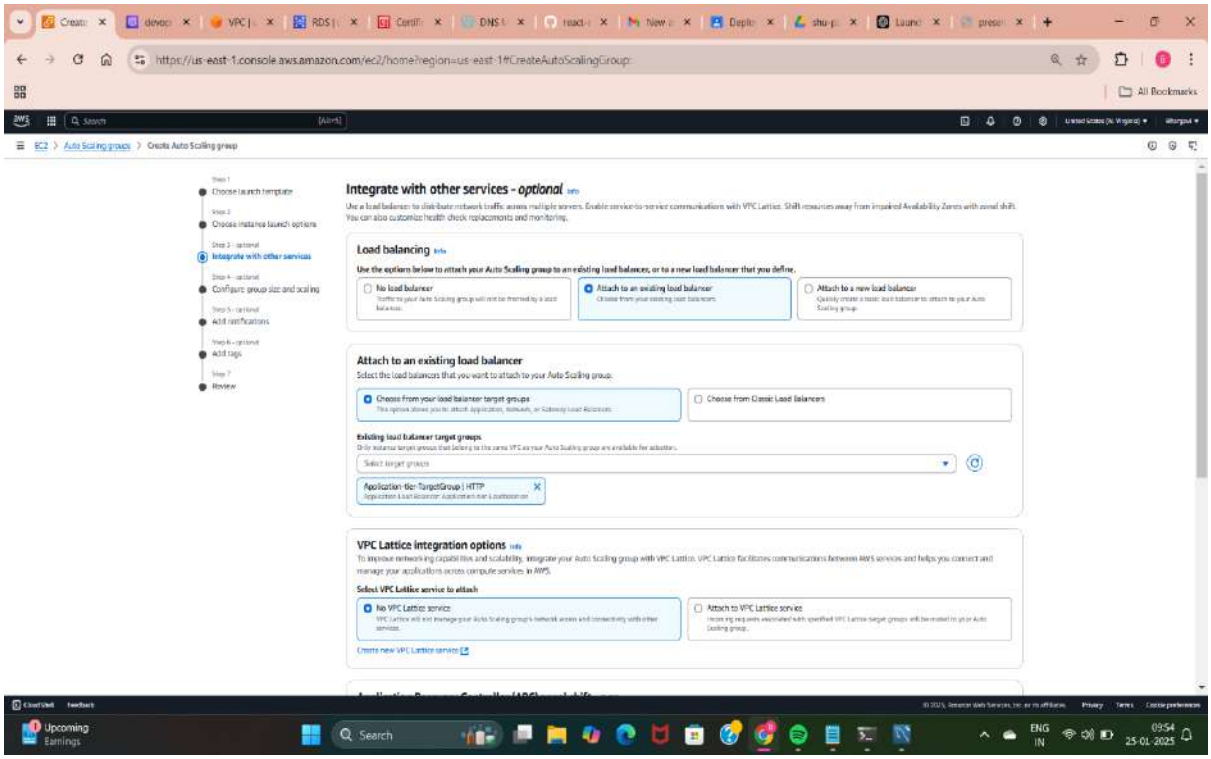


Basic Configuration:

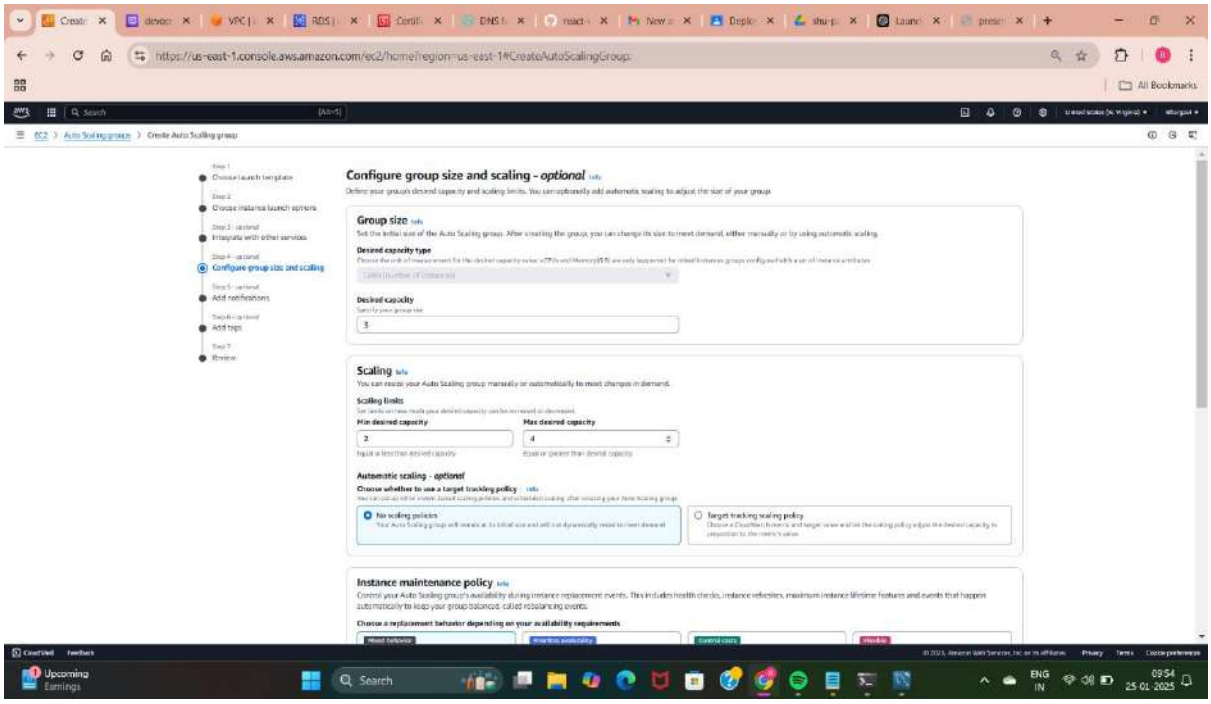
- ☑ Name: Enter a name, such as Application-tier-autoscalinggroup.
- ☑ Launch Template:
  - Select the Launch Template you created earlier for the Application Tier.
  - Ensure the correct version of the template is selected.



☑ Select 3-Tier VPC. Select the private subnets that correspond to the Application Tier.

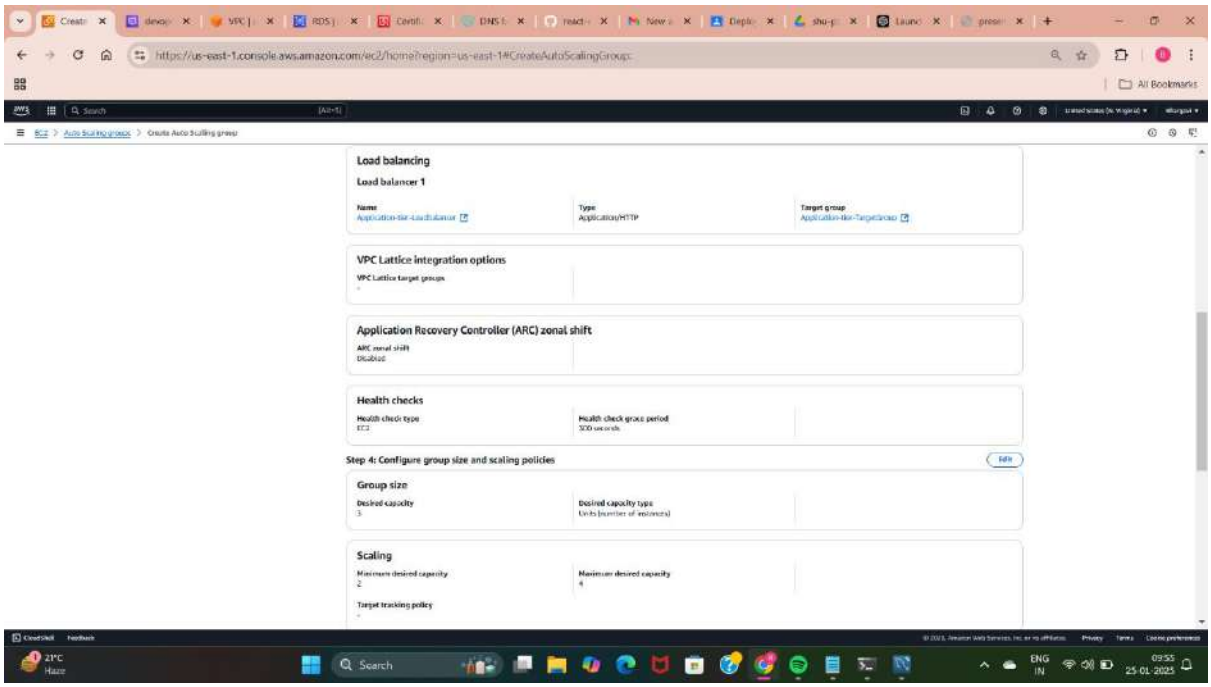


☑ Enable Application Load Balancer or Network Load Balancer.
☑ Select the Application-Tier-ALB Target Group you created earlier.

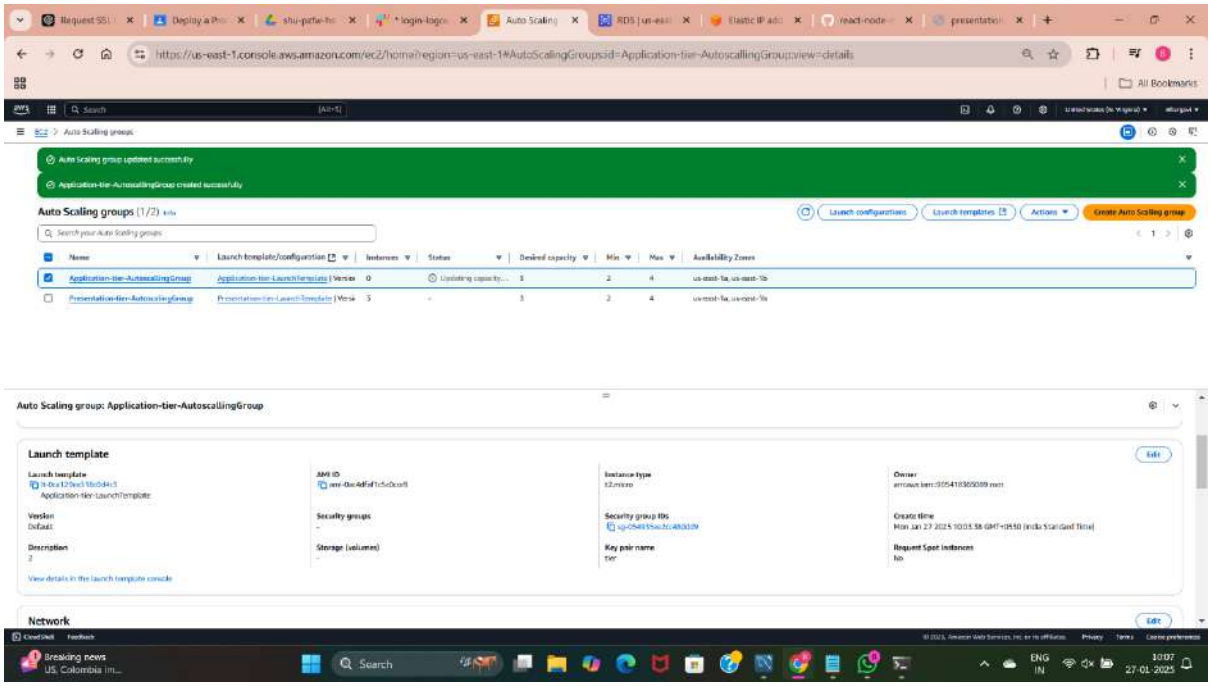


Configure Group Size as mentioned below:

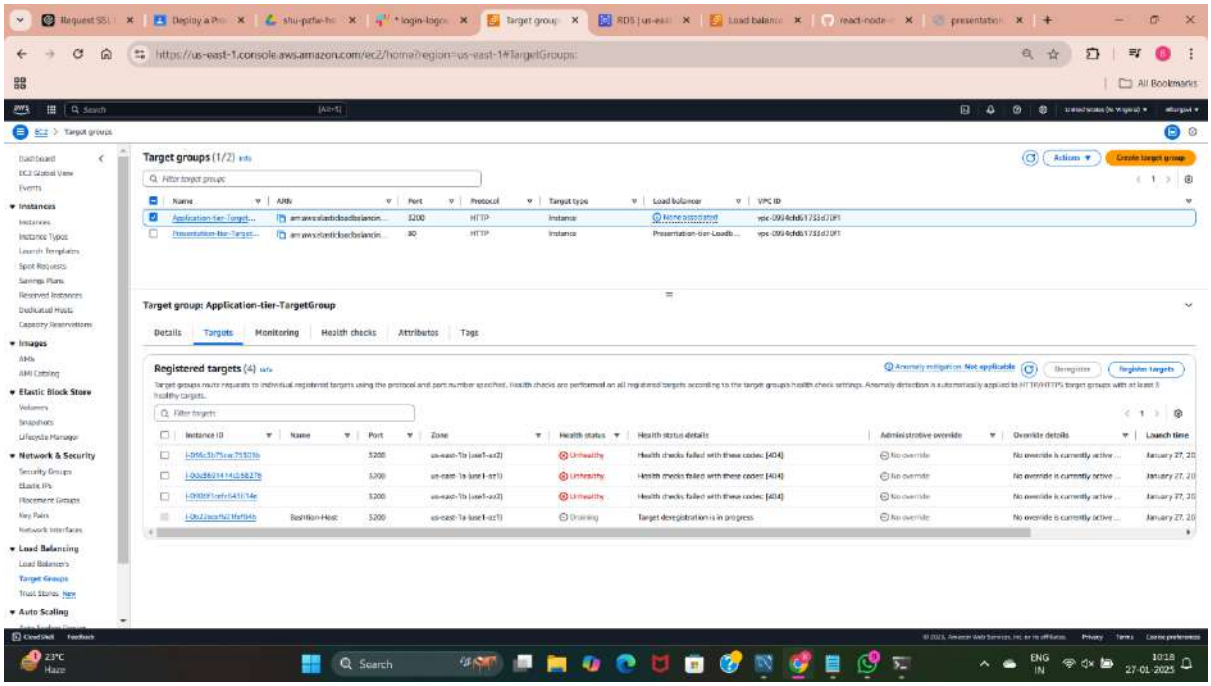
- ☑ Desired Capacity: 3
- ☑ Minimum Capacity: 2
- ☑ Maximum Capacity: 4



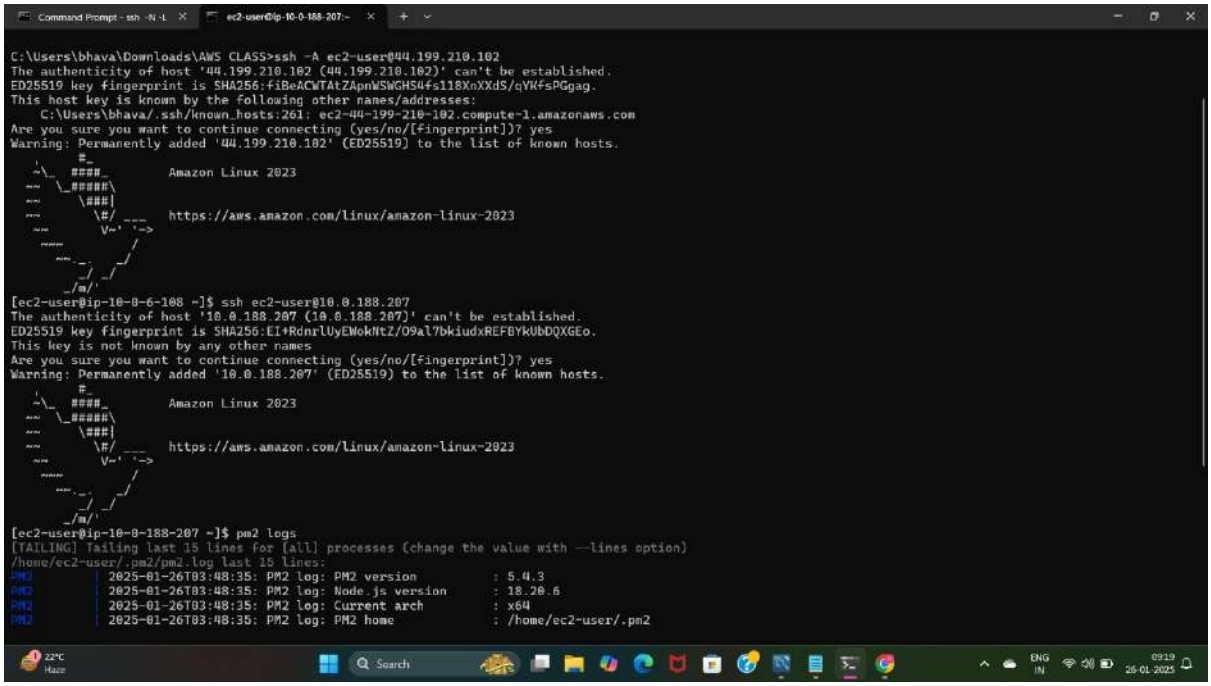
- ☑ Review all your configurations. Click Create Auto Scaling Group.



☑ If your configuration is correct, three instances will be created under EC2 Dashboard in the private subnets as part of the Application Tier Auto Scaling Group



☑ The created application tier ec2 instances are added to application tier target group.

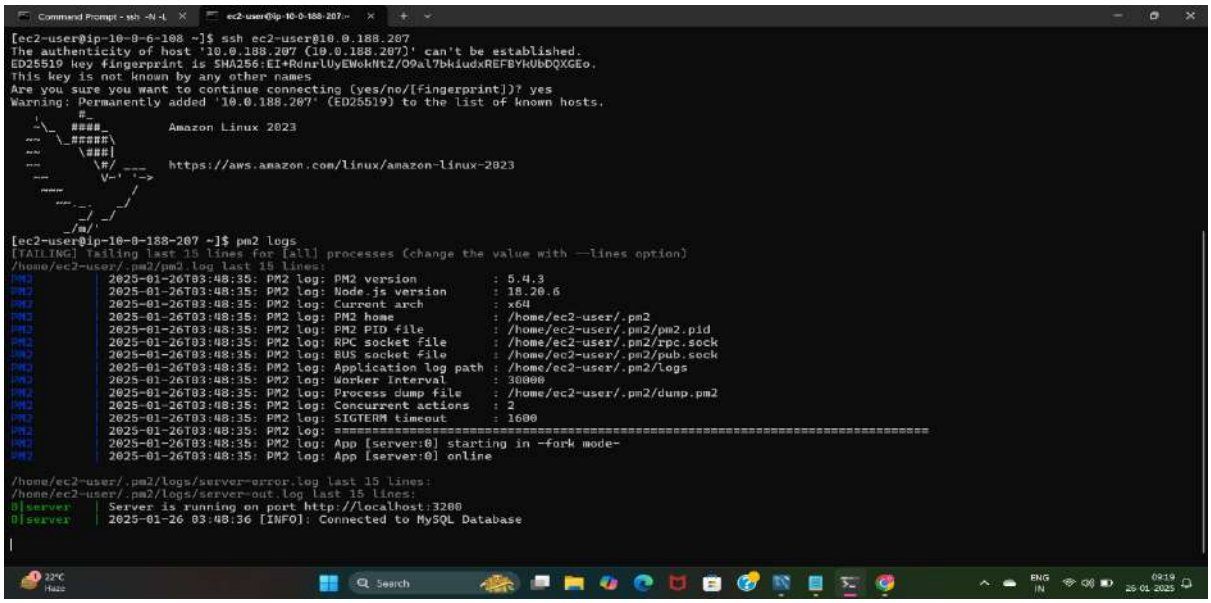


- ☑ SSH into the Bastion Host. Replace bastion_host_public_ip with the public IP of your Bastion Host.

  "ssh -A ec2-user@bastion_host_public_ip"
- ☑ Go to the EC2 Dashboard and check the private IP addresses of the instances created under the Application Tier.
- ☑ From the Bastion Host, connect to a private instance using its private IP address:

  "ssh ec2-user@private-ip"
- ☑ Run the following command to view logs from the backend application

  "pm2 logs"

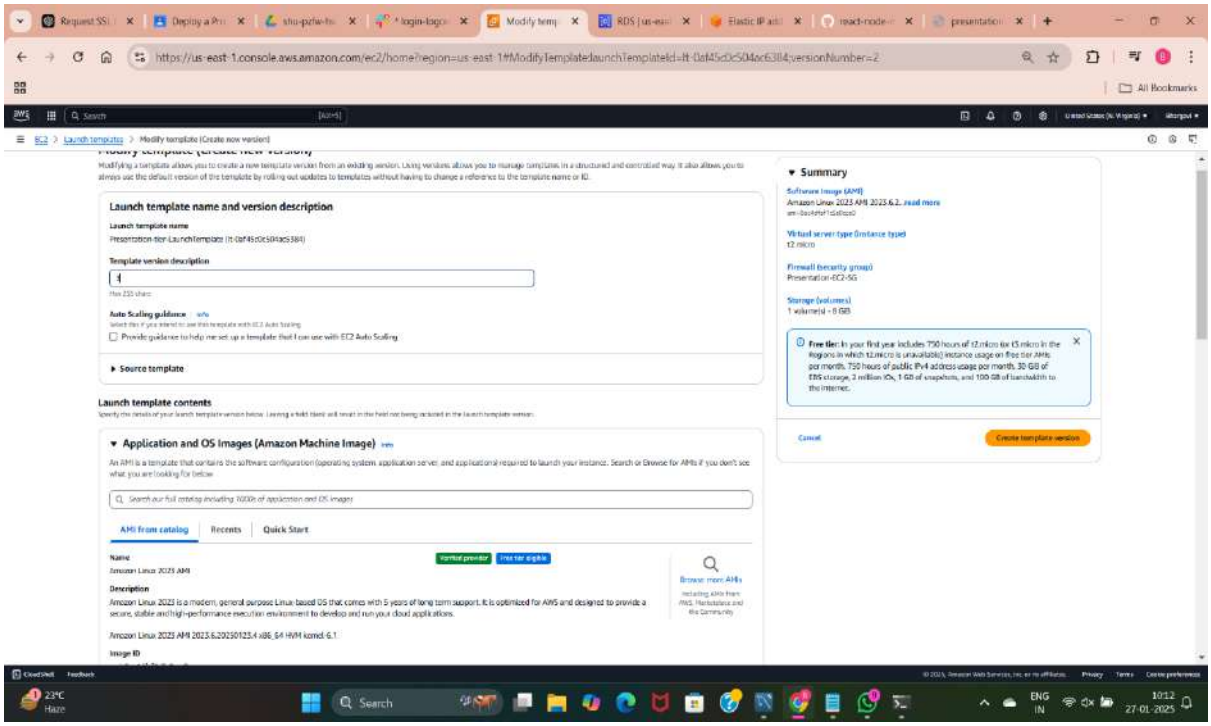


- ☑ The logs should display activity from our backend application running in the private instance. Look for entries like successful requests or database interactions.
- ☑ Your Application Tier Backend is now set up correctly, and the instances are processing requests as expected.

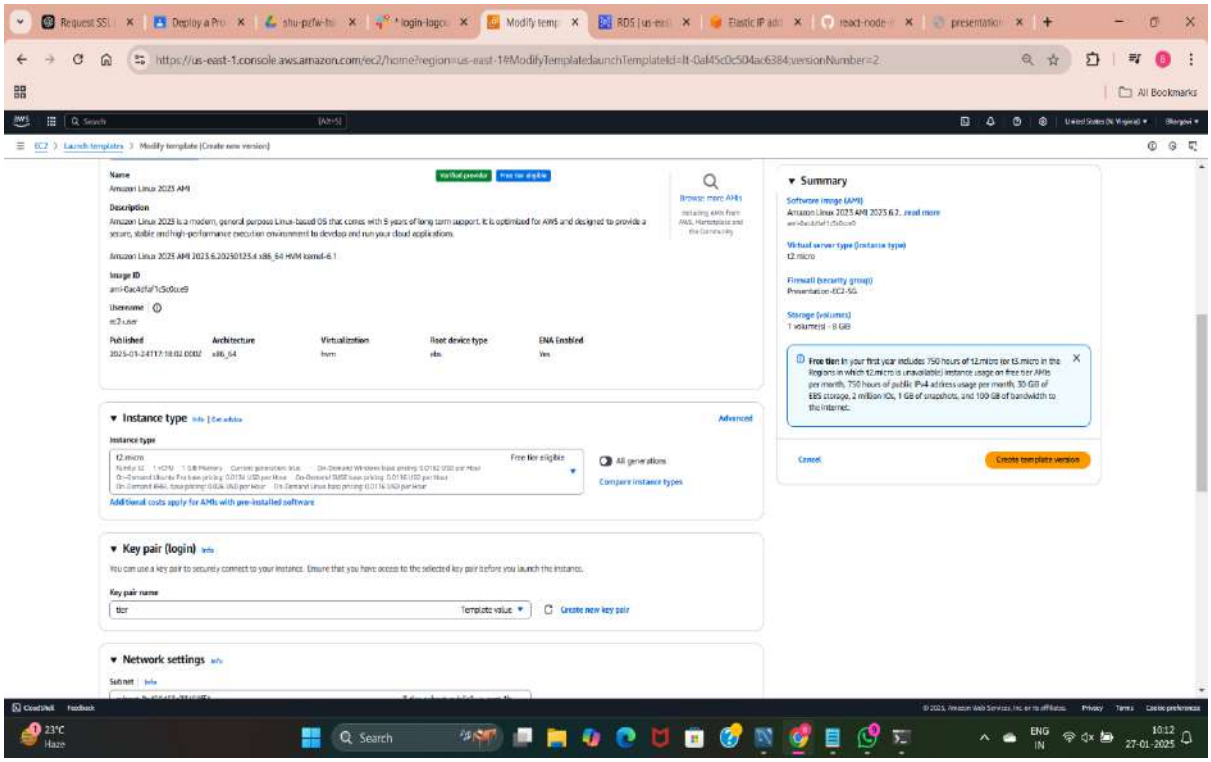
STEP 9: MODIFYING PRESENTATION TIER LAUNCH TEMPLATE

9.1 MODIFYING LAUNCH TEMPLATE WITH VERSION 2 USINGBELOW USER DATA

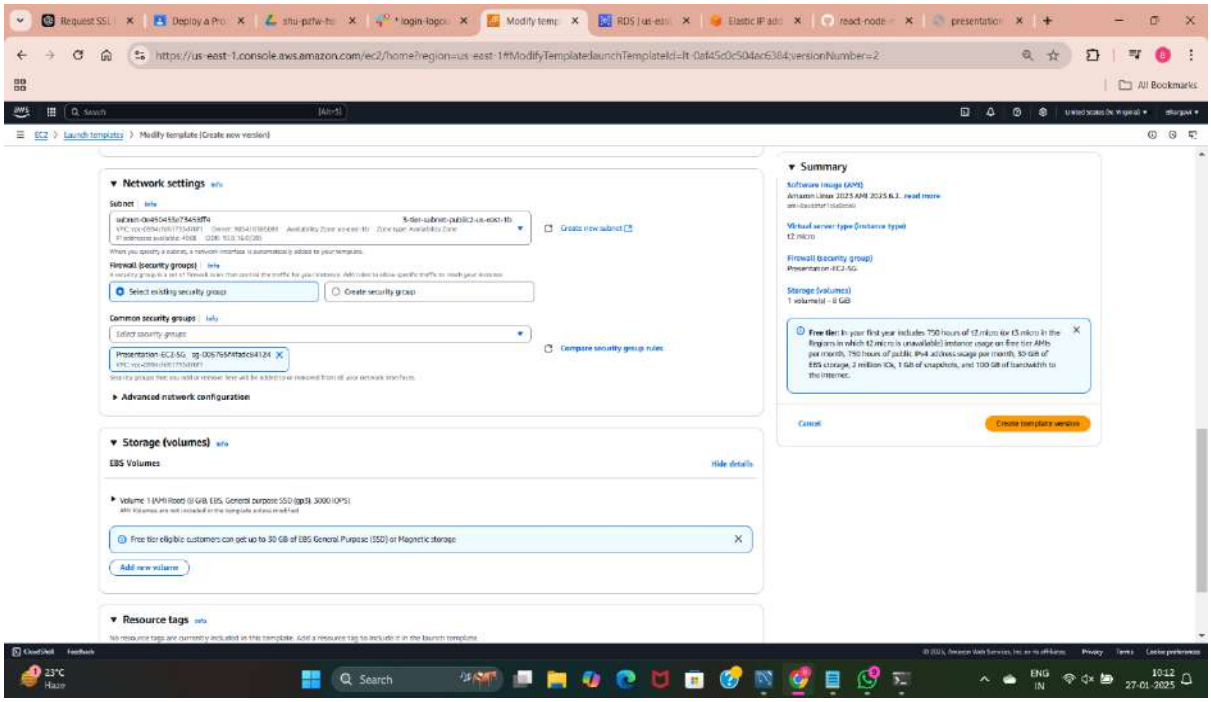
- To modify the existing Launch Template for the Presentation Tier and create a new version with updated User Data.
- Navigate to the EC2 Dashboard.
- On the left-hand menu, click Launch Templates.
- Select the Launch Template created for the Presentation Tier.



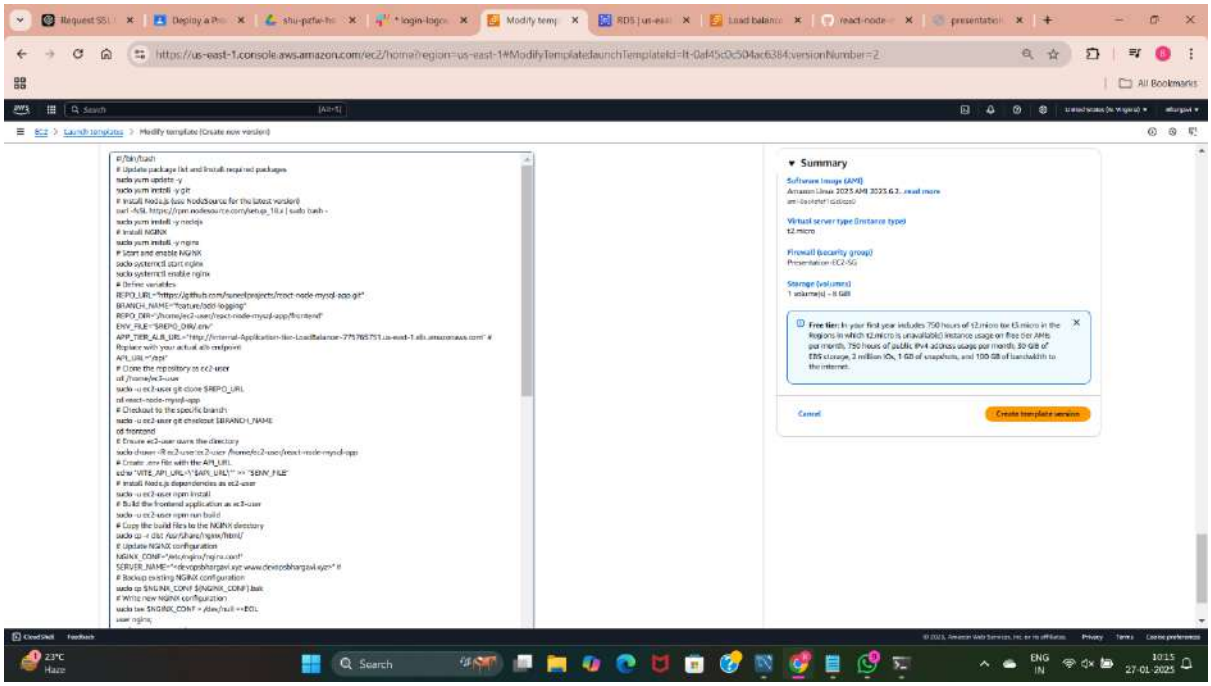
- Select Actions and click the dropdown menu and choose Create new version.



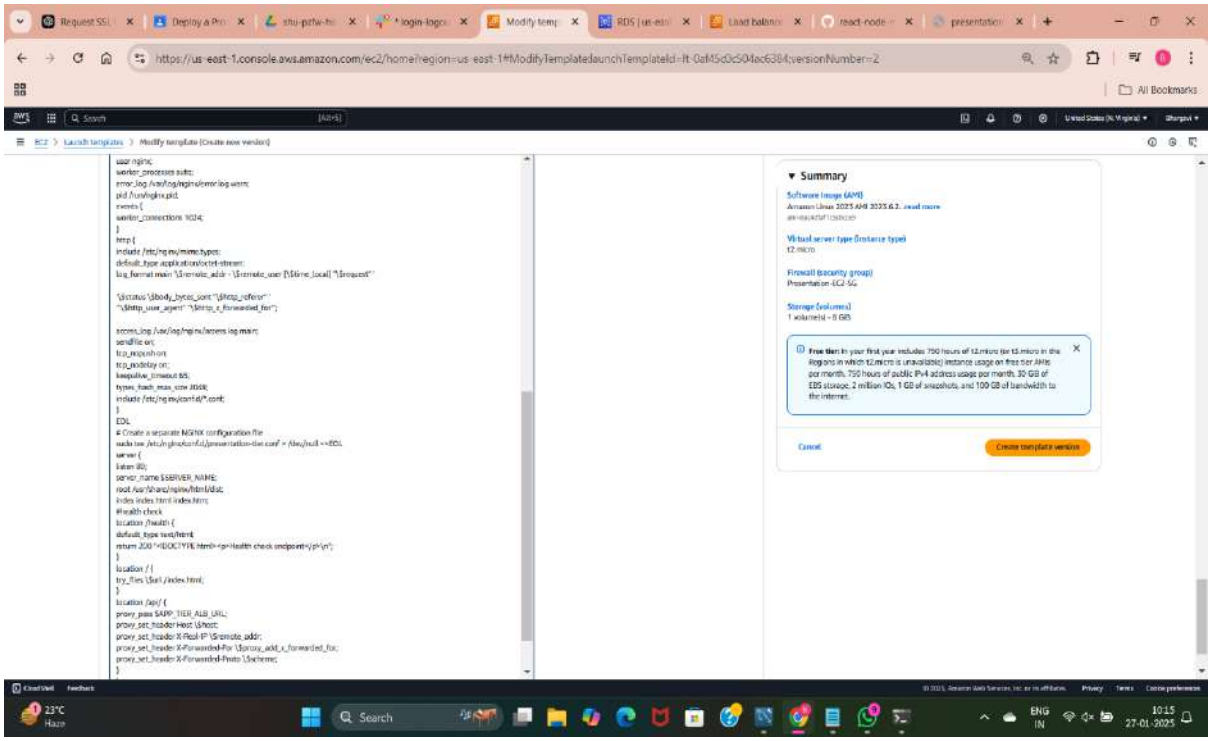
- Select key pair and check AMI details.

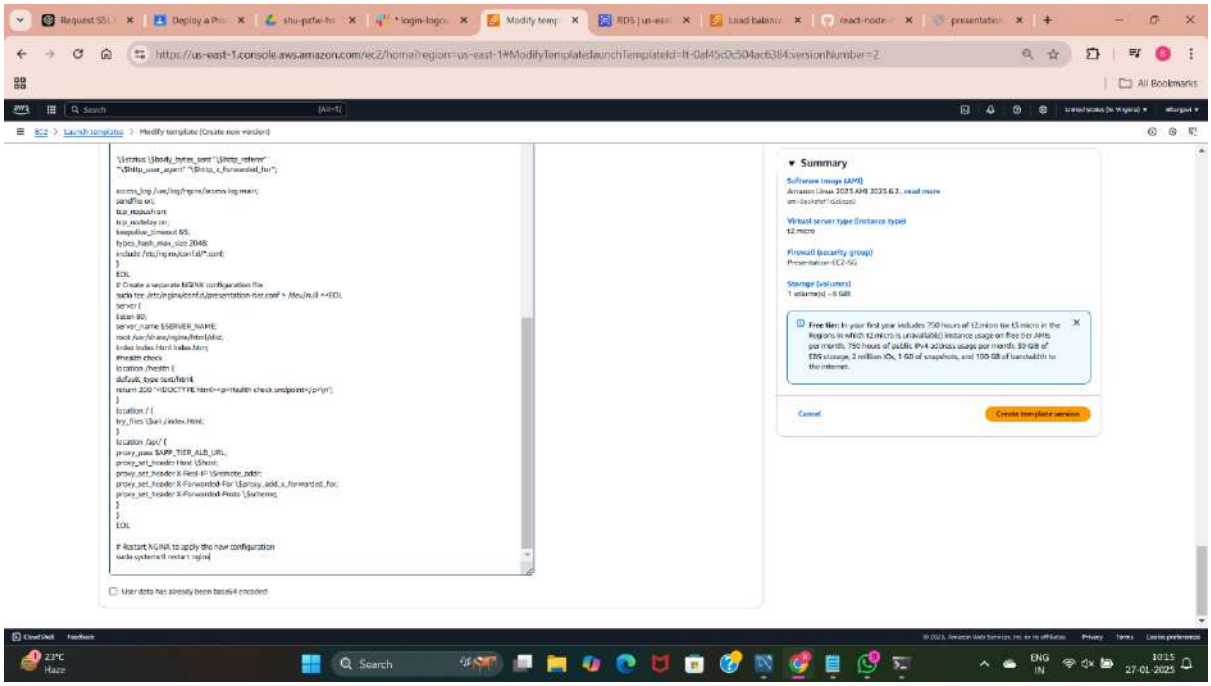


- Select the public subnet, security group for presentation tier launch template.

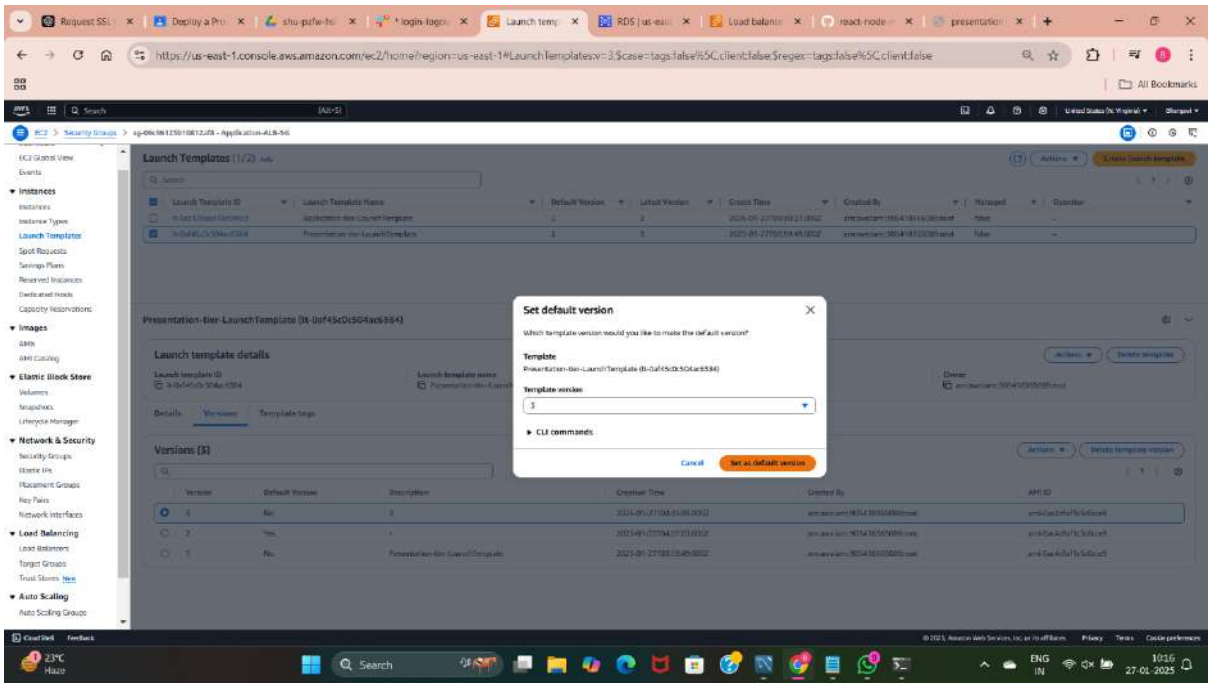


- Replace the existing User Data with the new script.





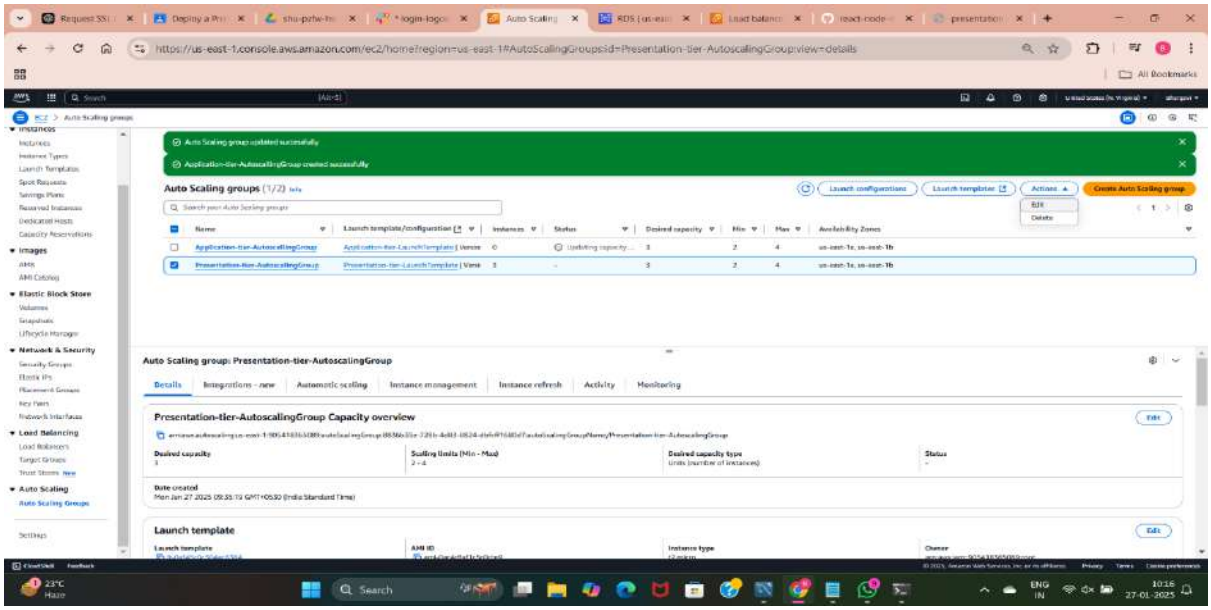
- Click Create Launch Template Version to finalize the new version.



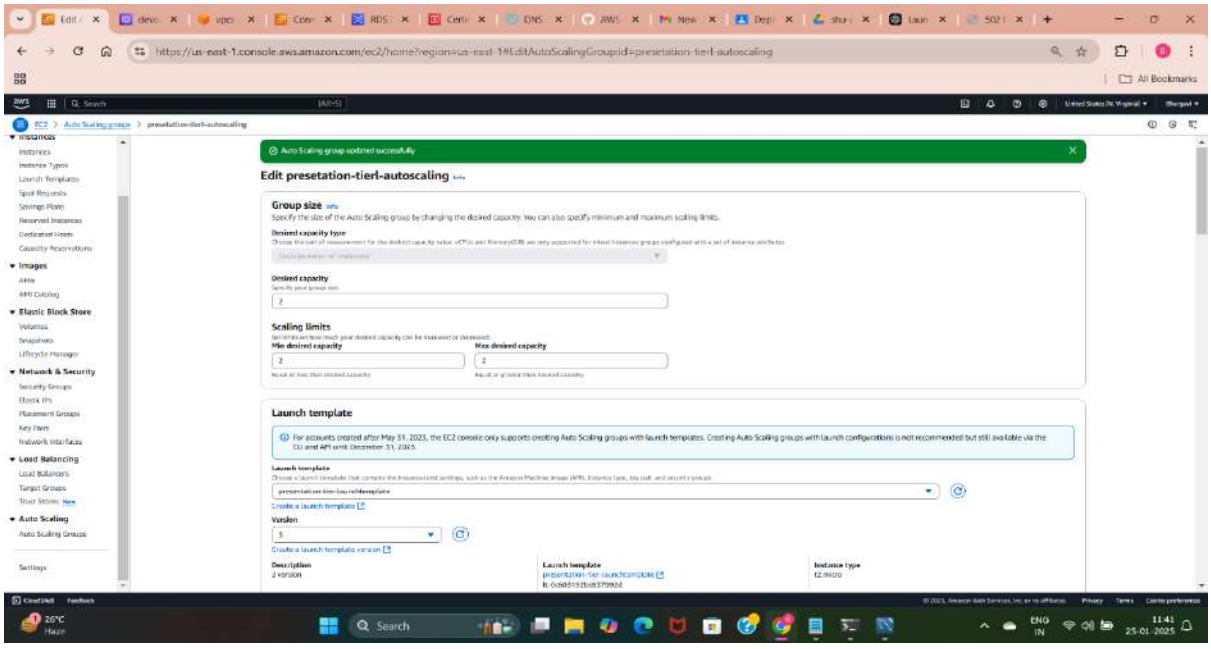
- Once the new version is created, select Actions → Set Default Version.

9.2 MODIFYING AUTO SCALING GROUP OF PRESENTATION TIER BY SELECTING THE LAUNCH TEMPLATE VERSION 2, GIVE DESIRED CAPACITY, MIN & MAX AS 2

- To update the Presentation Tier Auto Scaling Group with the newly created Launch Template Version 2
- Go to the EC2 Dashboard in the AWS Console.
- On the left-hand menu, select Auto Scaling Groups.
- Find and select the Presentation Tier Auto Scaling Group.



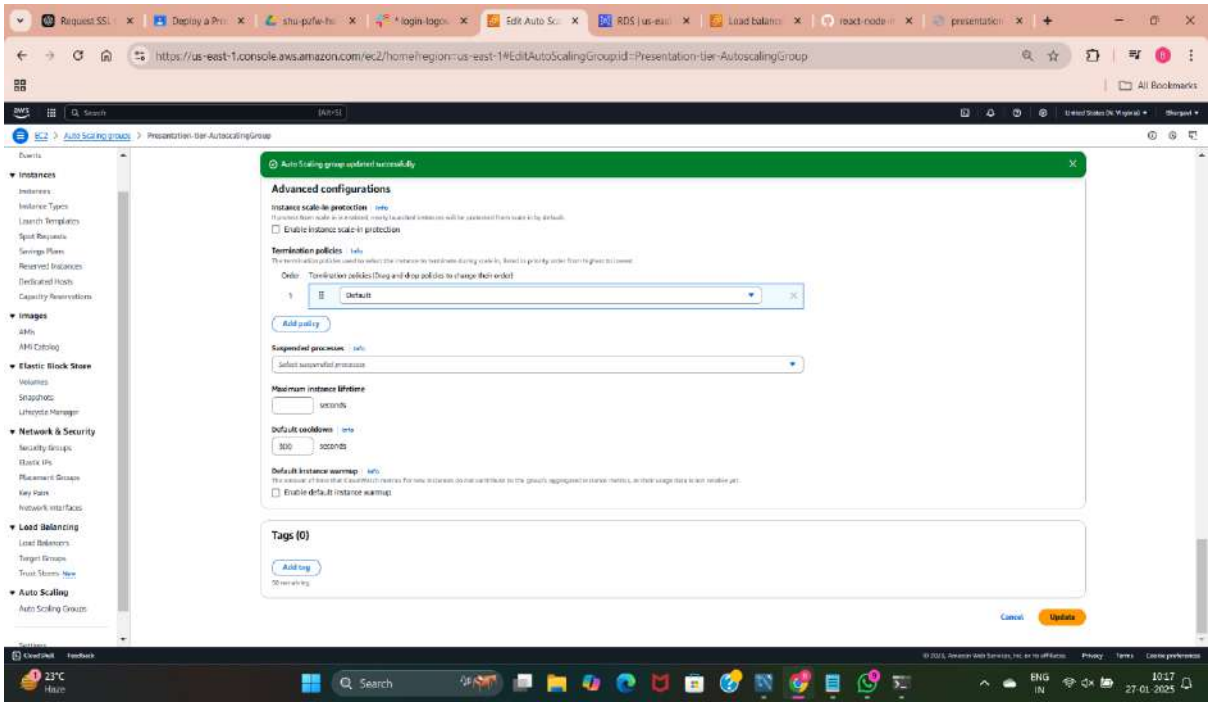
- Click Edit in the details section of the selected Auto Scaling Group.



Adjust Capacity Settings:

- Update the Desired Capacity: Set it to 2.

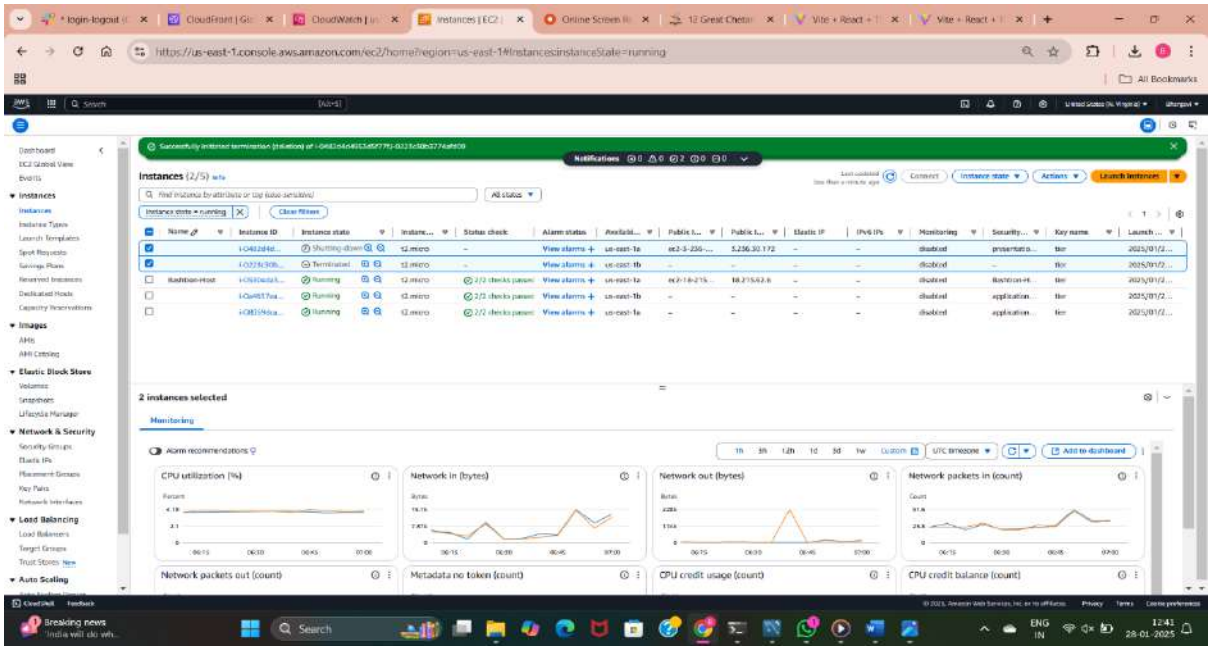
- Update the Minimum Capacity: Set it to 2.
- Update the Maximum Capacity: Set it to 2 (or adjust as needed for scaling flexibility).



- Click Update to apply the new settings.

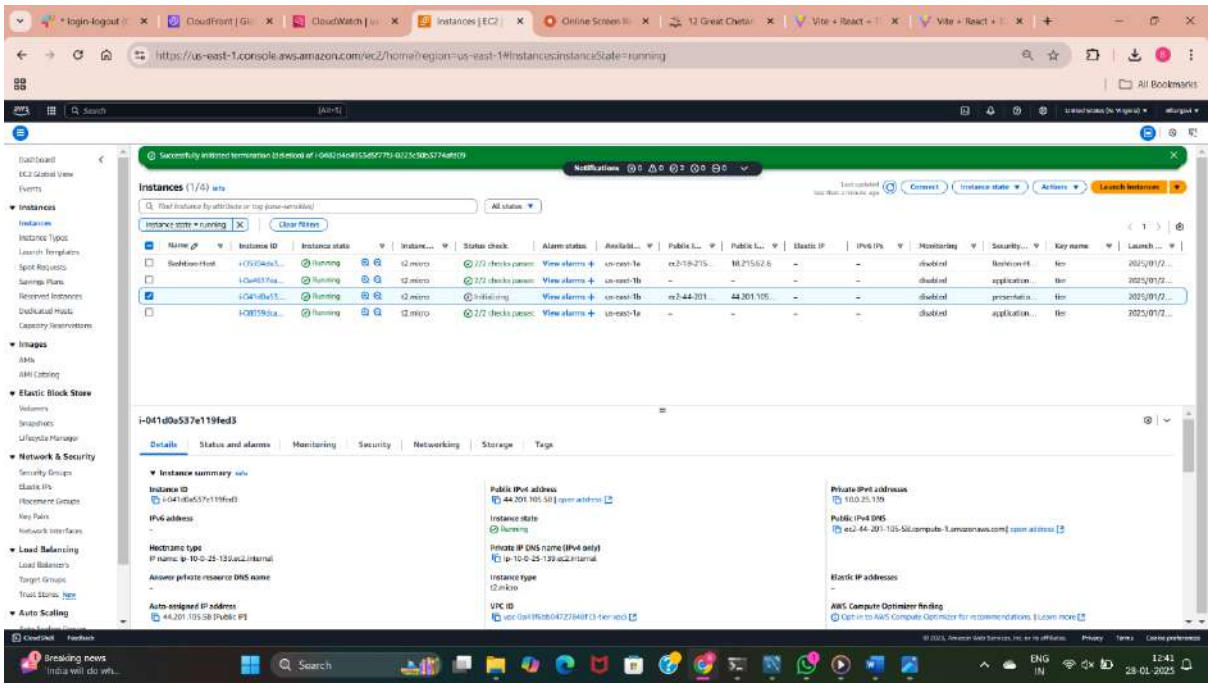
Under Launch Template, ensure the following:

- Select the Launch Template Name for the Presentation Tier.
- Set the Launch Template Version to 3.
- New Autoscaling group is created

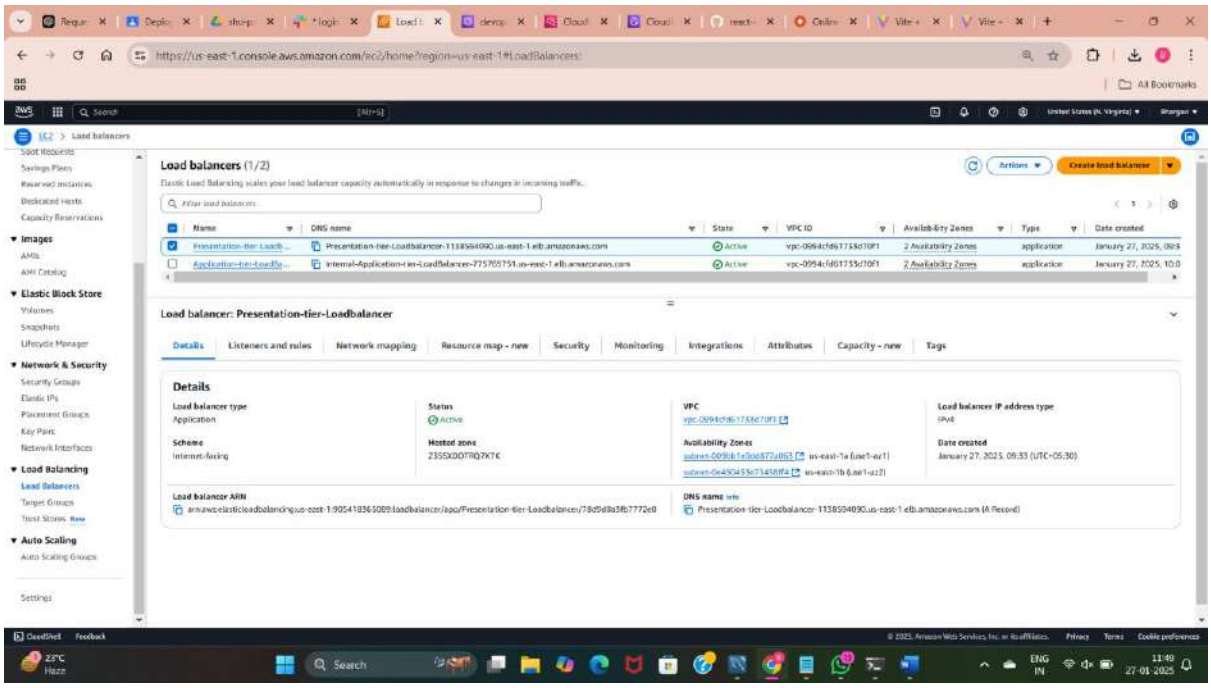


- Go to the EC2 Instances section in the AWS Console.

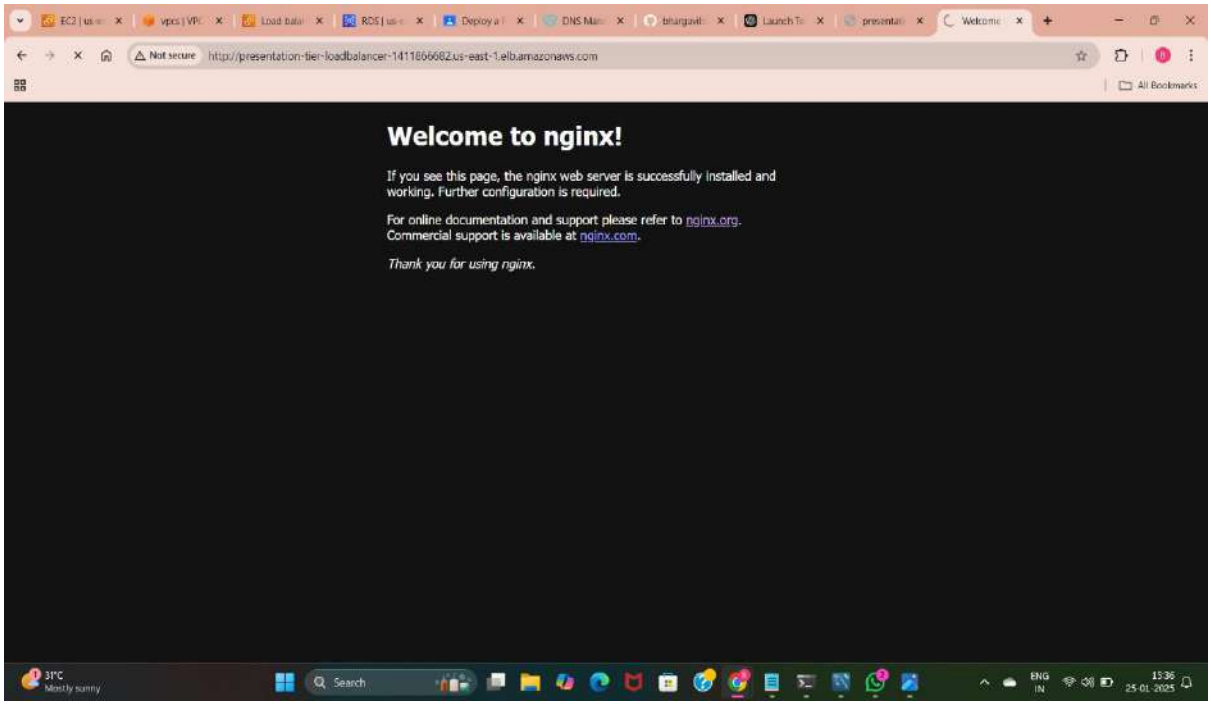
- Filter for instances associated with the Presentation Tier Auto Scaling Group.
- Terminate all running instances that are using the old configuration:
- Select the instances and click Instance State → Terminate.



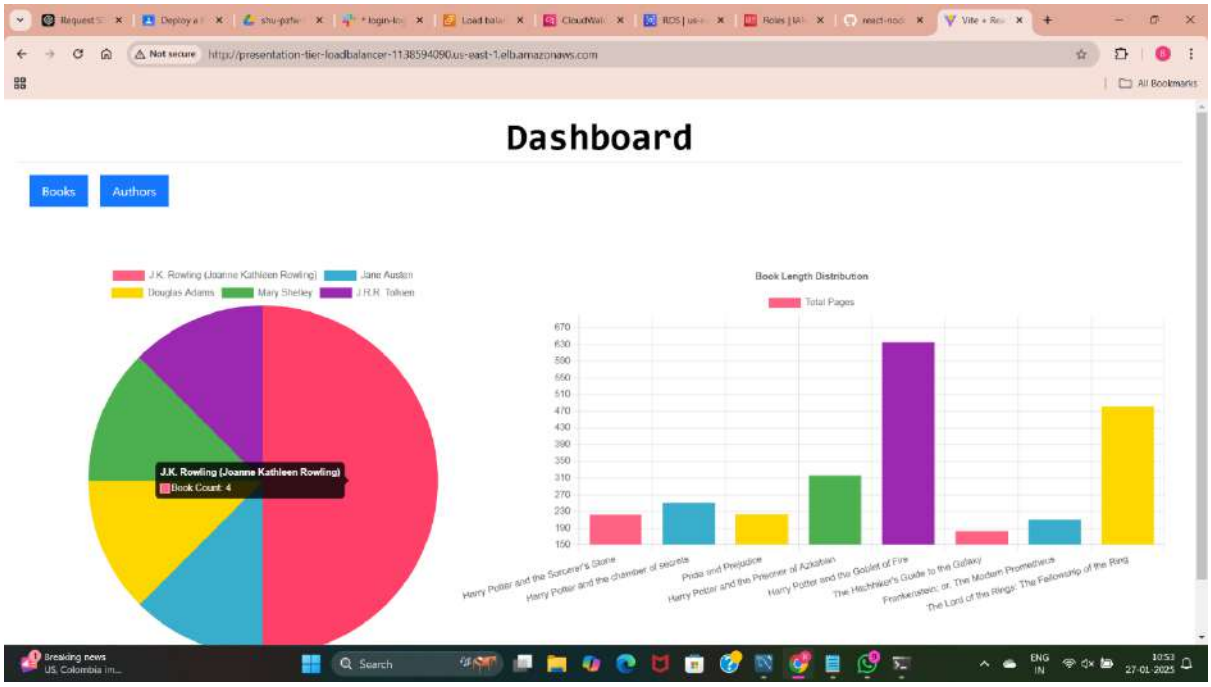
- The Auto Scaling Group will automatically launch new instances with the update configuration.



- Once the new instances are launched, go to the Load Balancer DNS Name for the Presentation Tier.
- The DNS Name is located in the Load Balancer section of the EC2 Dashboard under Description.



- Open the DNS Name in a browser.



- Verify that the updated configuration is reflected (e.g., custom web page message or application features).

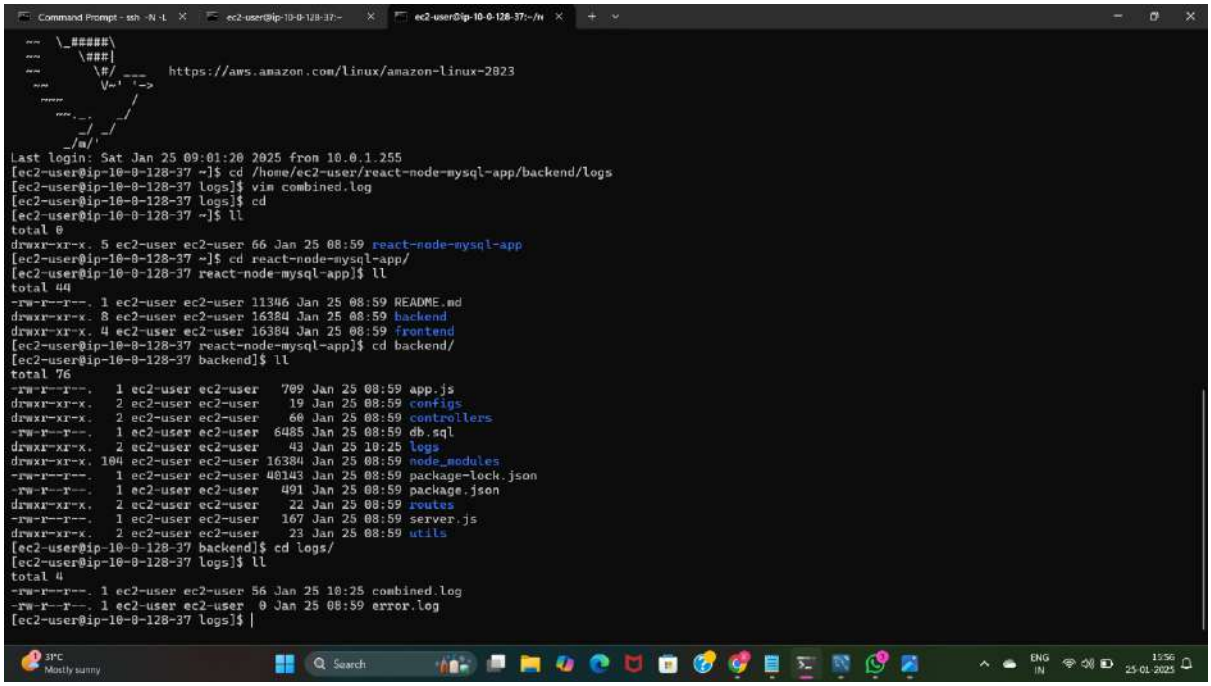
AMAZON CLOUDWATCH

Amazon CloudWatch is a monitoring and observability service provided by AWS that gives you insight into the performance and health of your AWS resources and applications. It allows you to collect, monitor, and analyze various metrics, logs, and events to help ensure that your applications are running smoothly and meet your performance and operational goals.

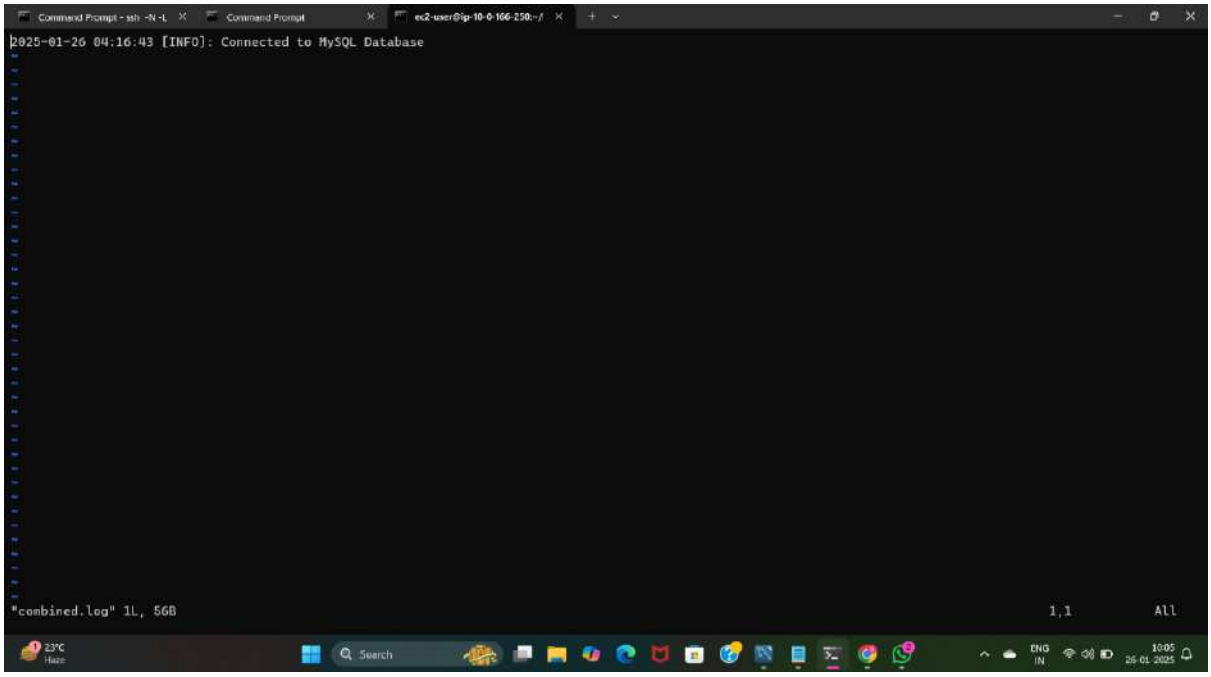
CloudWatch is used for a wide range of use cases, such as tracking the performance of EC2 instances, Lambda functions, databases, and other AWS services. It can also be used to create alarms, automate responses, and visualize data through dashboards.

STEP 10: INTEGRATING APPLICATION LOGS WITH CLOUDWATCH

- To integrate the application logs from the backend of the Application Tier with Amazon CloudWatch, Follow these steps.
- From your terminal, SSH into the Application Tier EC2 instance through the Bastion Host:
  "ssh -A ec2-user@<bastion_host_public_ip>"
  "ssh ec2-user@<application_tier_private_ip>"



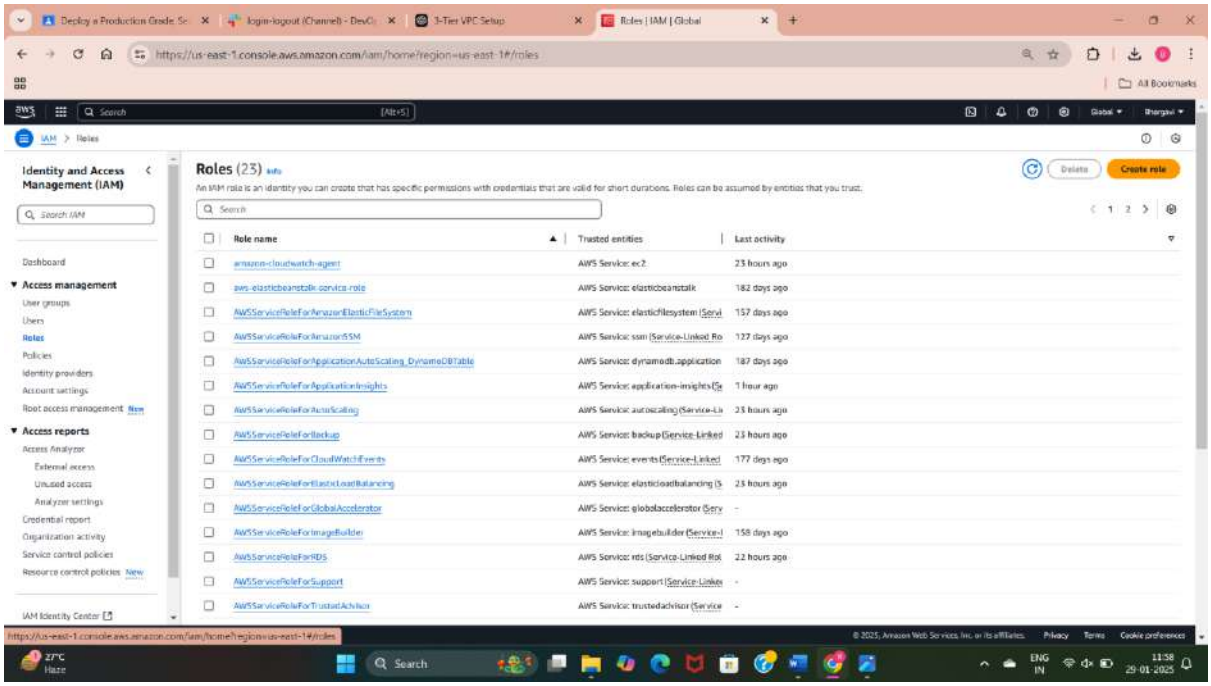
- Once connected, navigate to the logs directory
  "cd /home/ec2-user/react-node-mysql-app/backend/logs"
  "ll"
- Open the log file to confirm it contains application logs
  "vim combined.log"



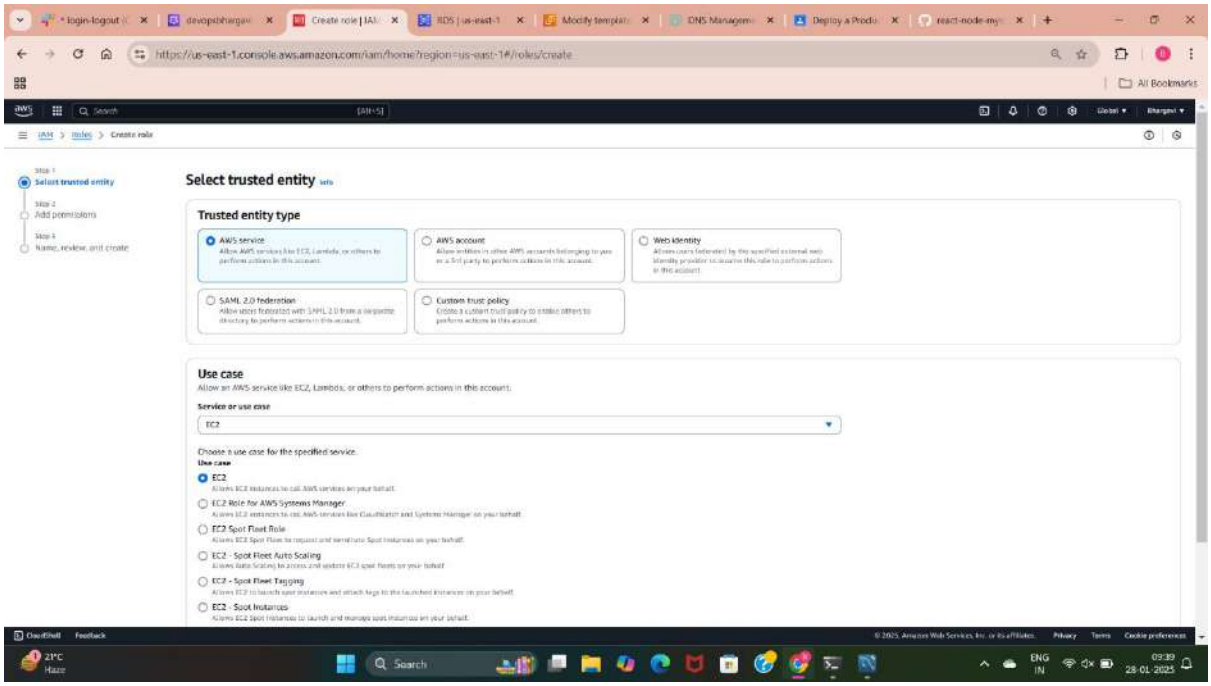
- As expected we can see the mysql database is connected.

10.1 CREATING IAM ROLE USING BELOW POLICIES

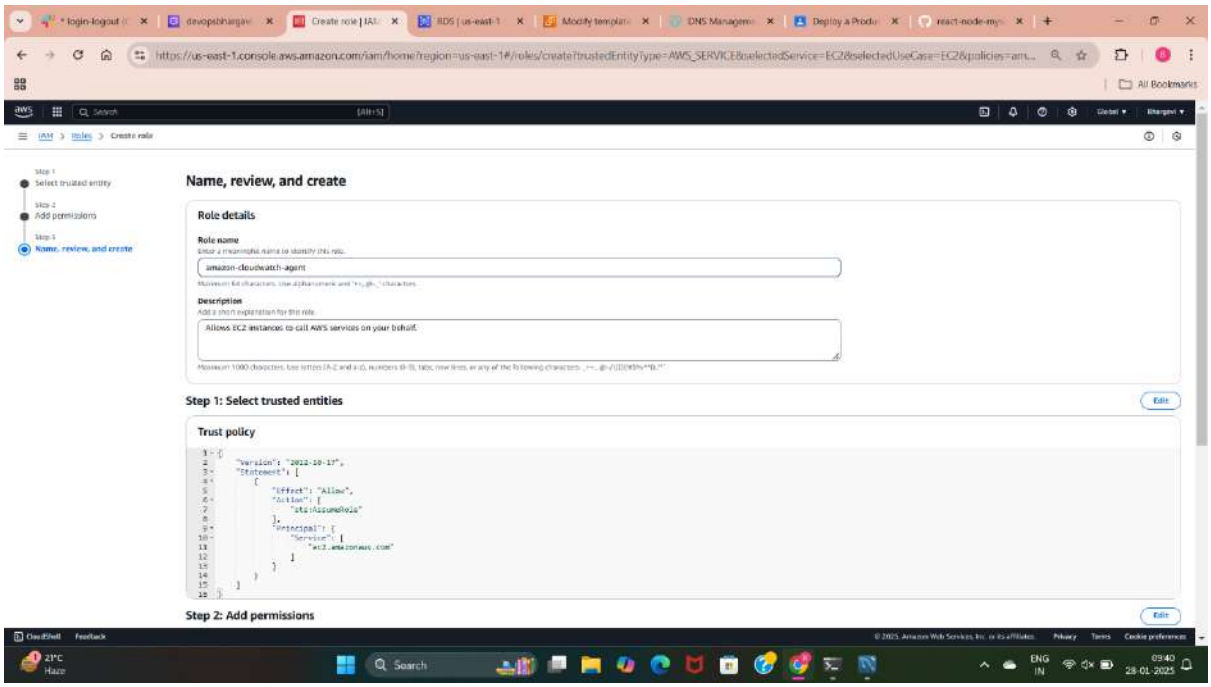
- To enable the CloudWatch Agent on your EC2 instances, you need to create an IAM Role with the necessary policies.
- Open the aws IAM console.
- In the left sidebar, click Roles.



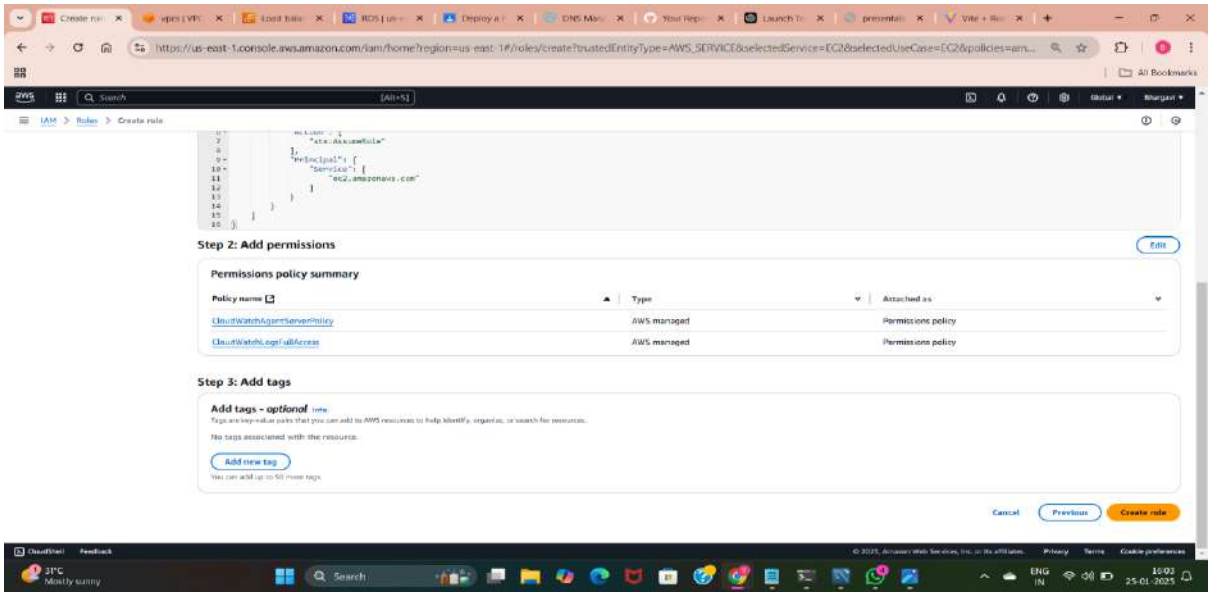
- Click the create role button



- Under Trusted entity type, select AWS Service.
- For Use case, choose EC2.
- Click Next.



- Give role name as amazon-cloudwatch-agent and select the roles.

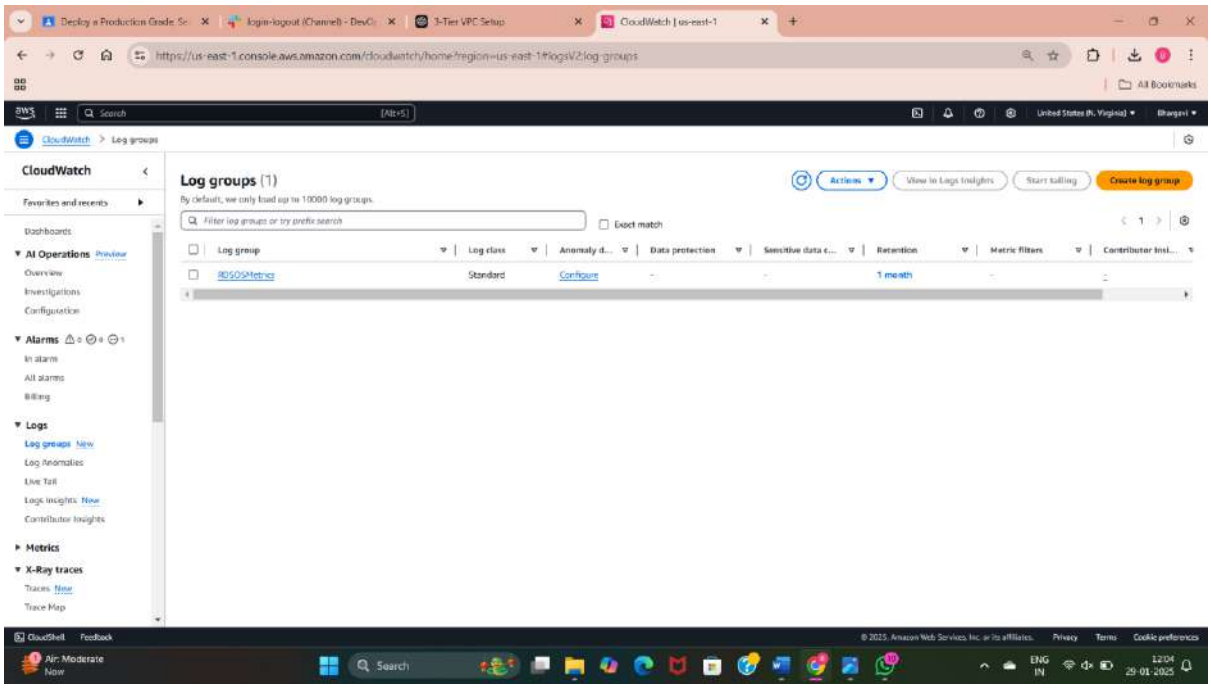


Search for and attach the following policies:

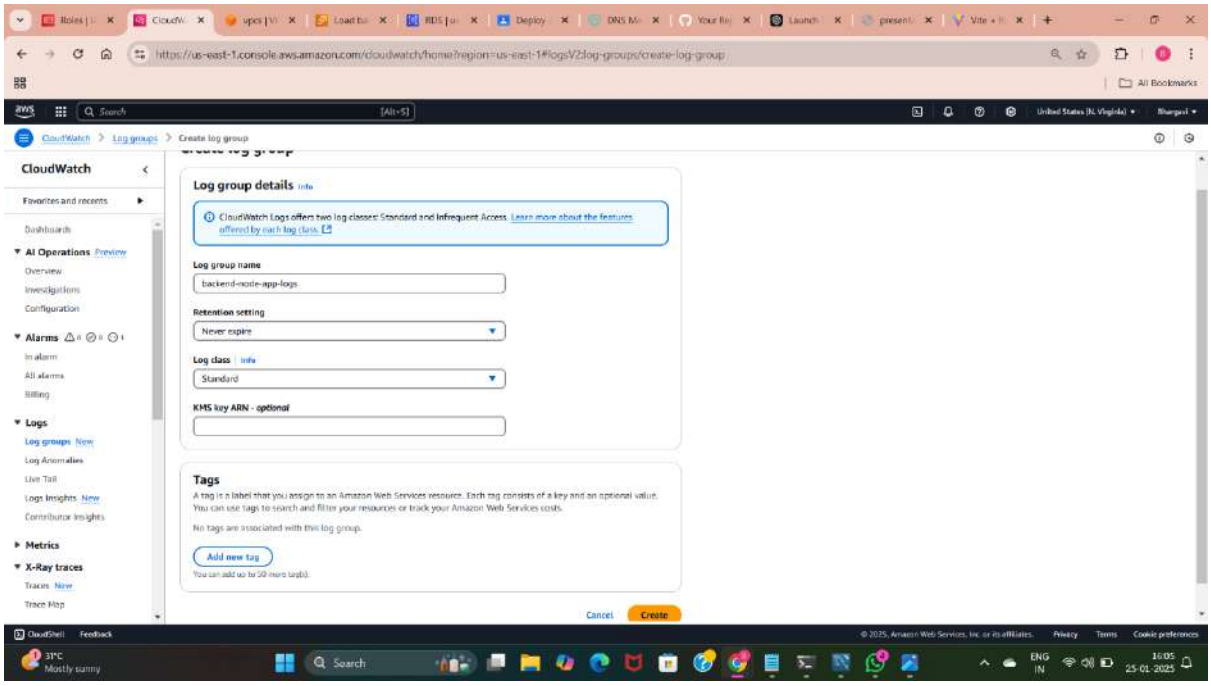
- CloudWatchLogsFullAccess: Allows full access to manage CloudWatch Logs.
- CloudWatchAgentServerPolicy: Provides permissions to the CloudWatch Agent to write metrics and logs to CloudWatch.
- Click on create role.

10.2 CREATING CLOUD WATCH LOG GROUP WITH NAME

- Open the AWS Management Console.
- Navigate to the CloudWatch service.
- In the CloudWatch dashboard, click on Logs from the left-hand menu.
- Select Log Groups.



- Click on the Create log group button.

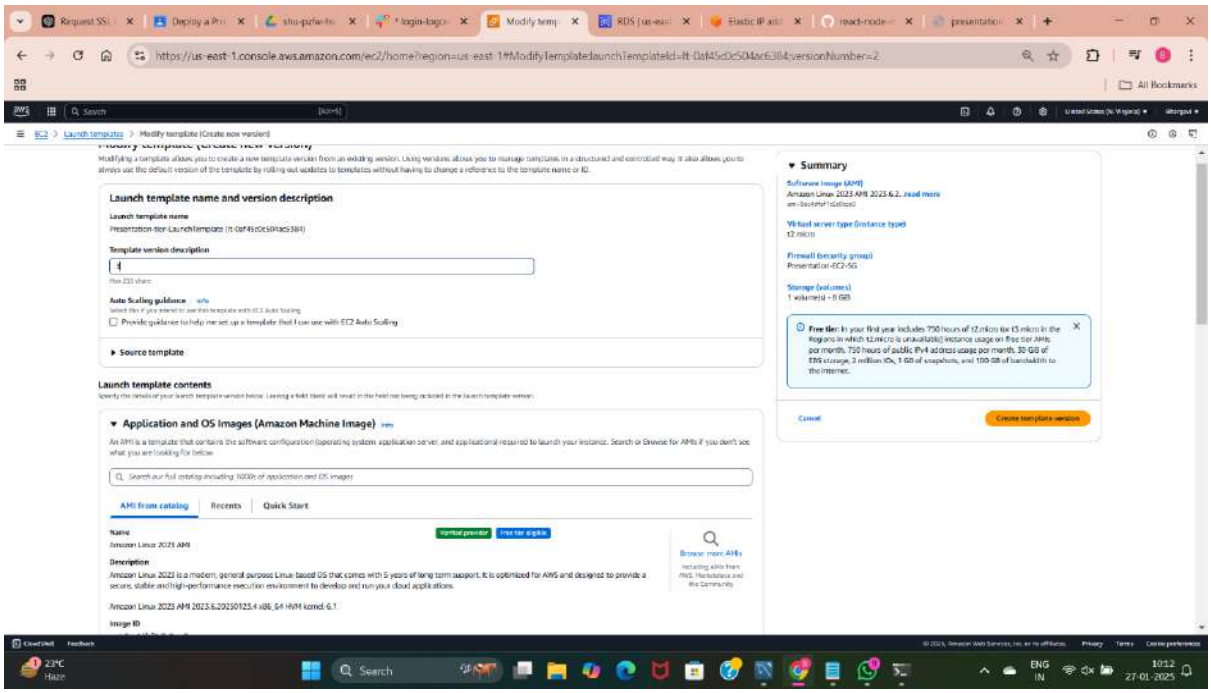


- Enter your desired log group name as backend-node-app-logs.

- Set a retention period for your logs (e.g., 7 days, 30 days, etc.).
- This ensures that logs are automatically deleted after the specified period.
- Click Create to finalize the log group.

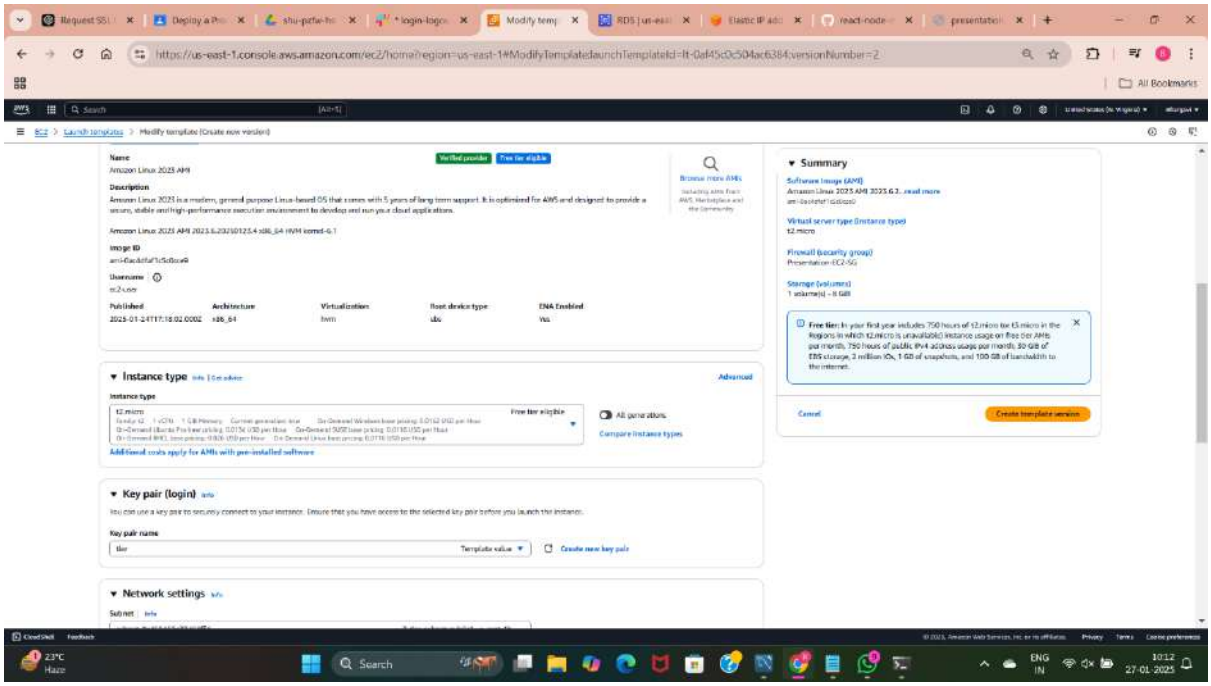
STEP 11: MODIFYING APPLICATION TIER LAUNCH TEMPLATE AS VERSION 2

- To modify the Application Tier Launch Template and create a new version (Version 3), follow the steps below to ensure all the changes are properly implemented:
- Go to the AWS Management Console.
- Navigate to EC2 Dashboard > Launch Templates.
- Select the launch template for the Application Tier.

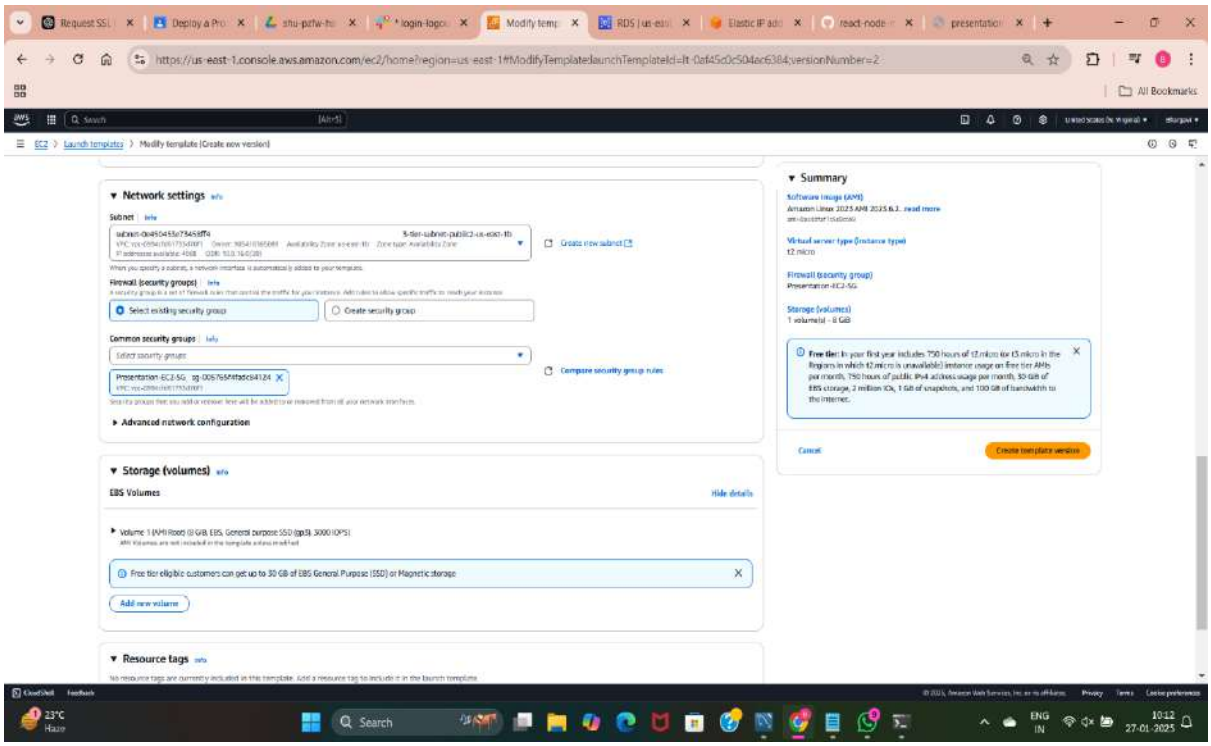


- Click on the Actions dropdown and select Modify Template (Create New Version).

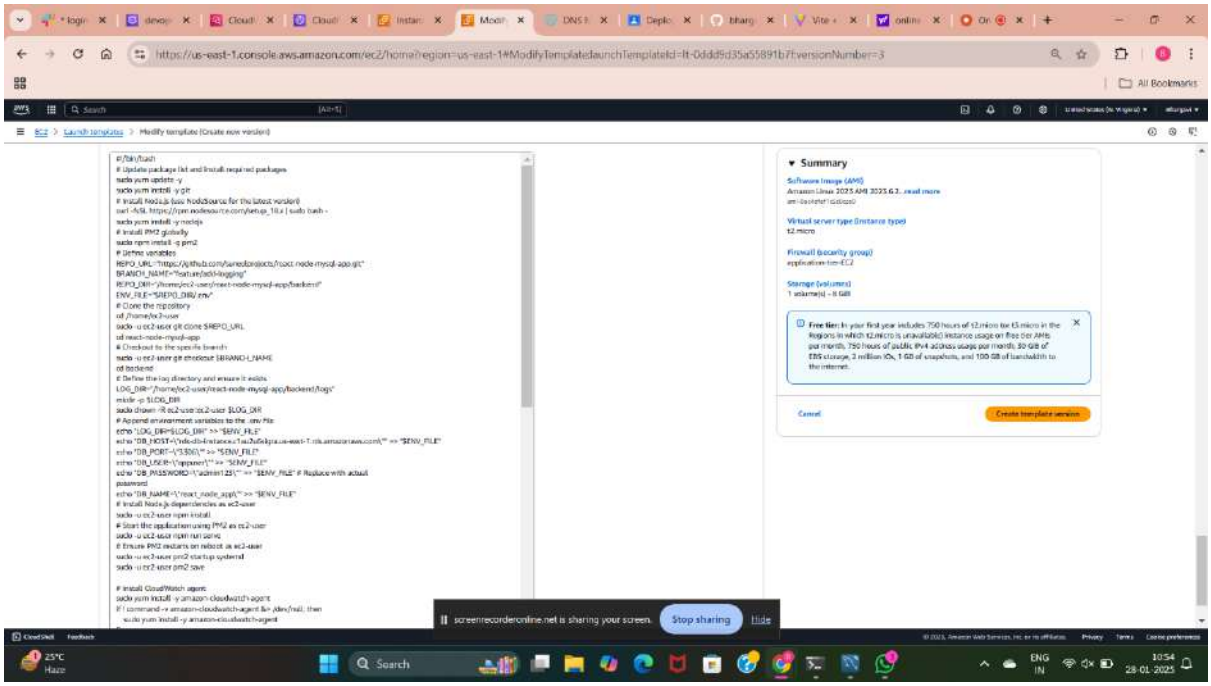
Update the following fields as per the new requirements



- If necessary, change the instance type to handle increased workload. Here I have selected t2.micro with key pair.

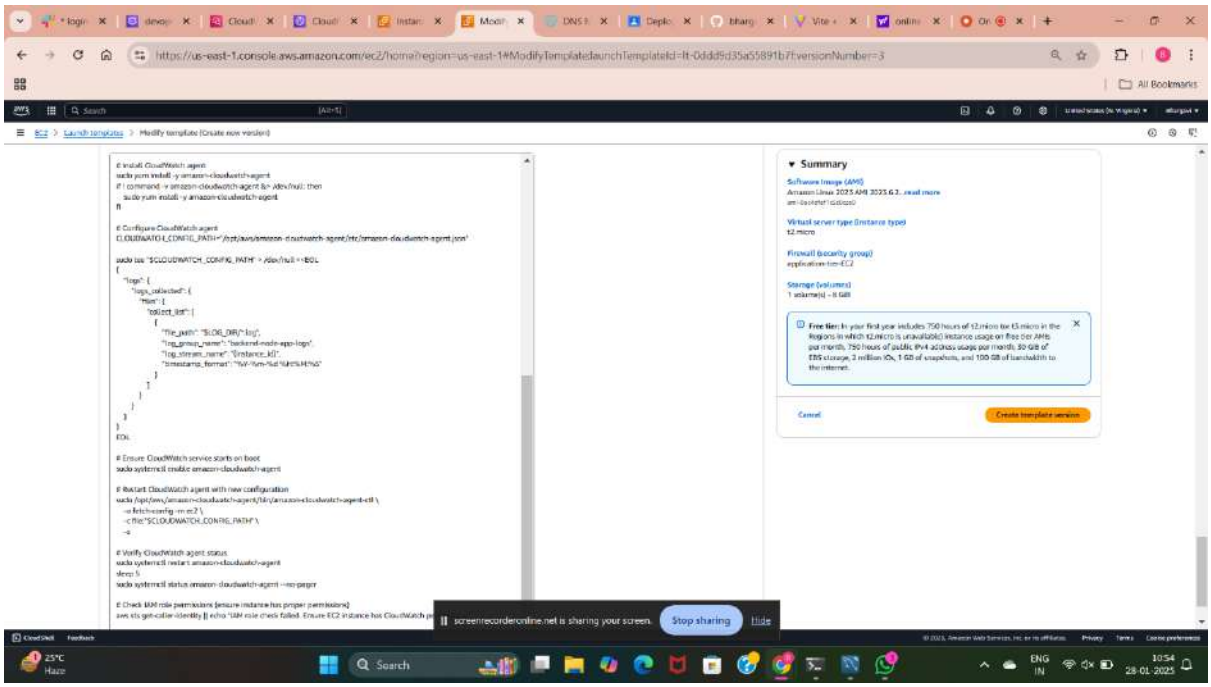


- Ensure the appropriate security group for the Application Tier is selected (e.g., allows necessary ports like 3200, 3306, and others).

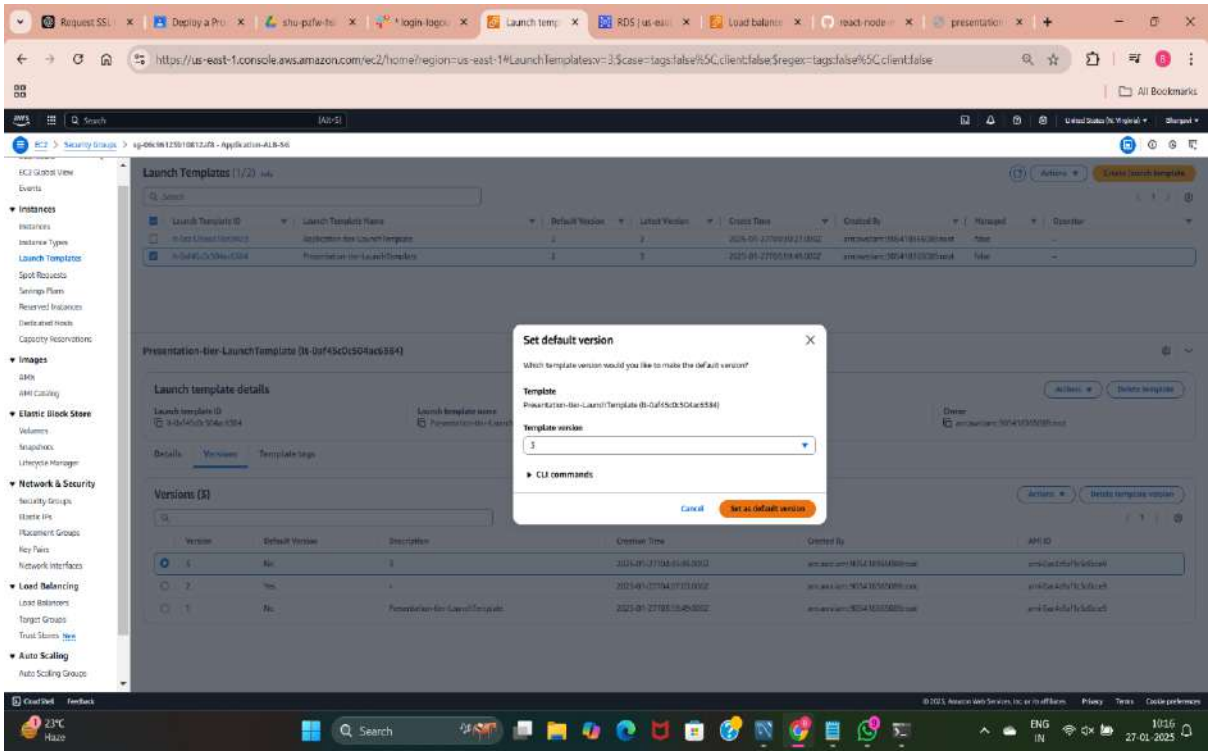


Update the script to include any new configurations.

- Ensure the DB_HOST, DB_USER, DB_PASSWORD, and other relevant environment variables are correctly added or updated.
- Modify the DB_USER to appuser and add the database name react_node_app.
- Ensure the RDS Endpoint matches the new database setup.



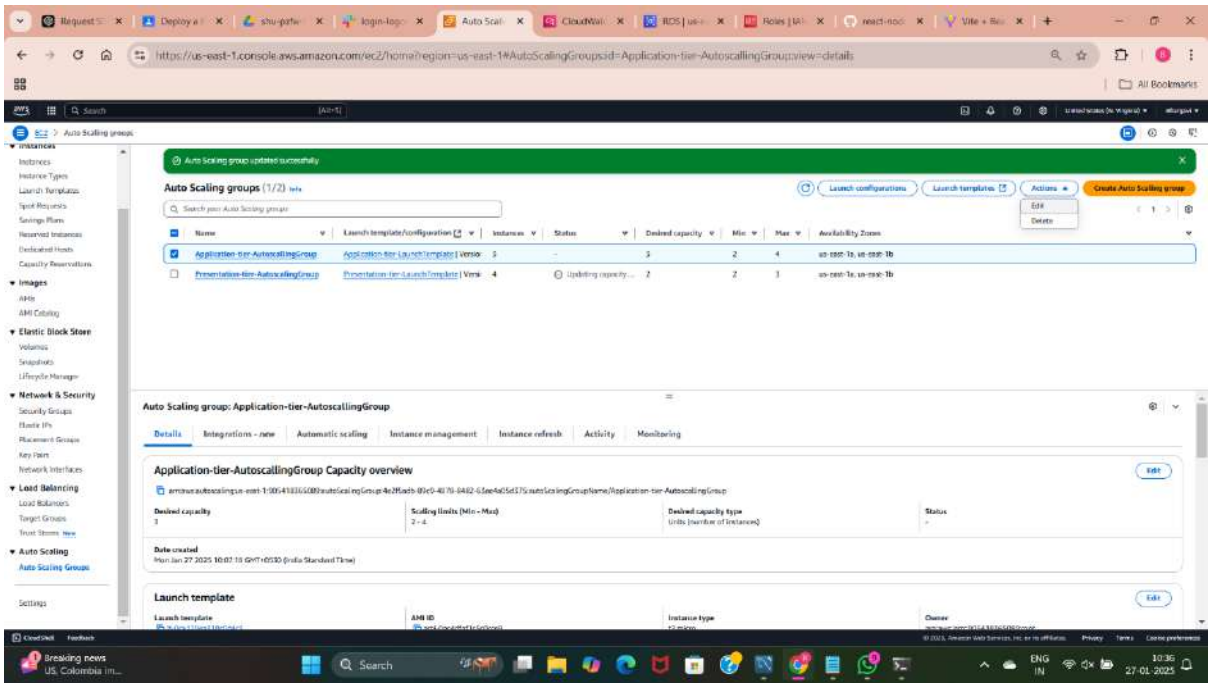
- Save the changes and click Create Template Version.



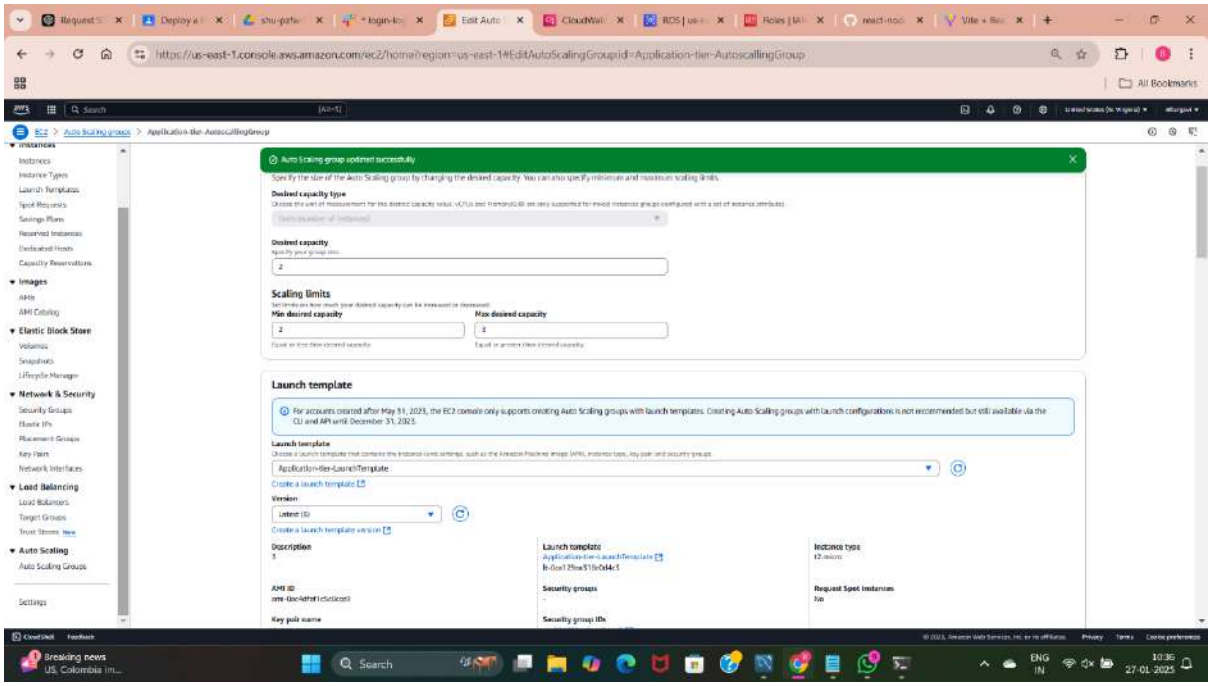
- Am changing the newly created version as default version of launch template.

11.1 MODIFYING ASG WITH VERSION 2, AND INSTANCES AS 3

- Go to the Auto Scaling Groups section in the EC2 Dashboard.
- Select the Auto Scaling Group for the Application Tier.

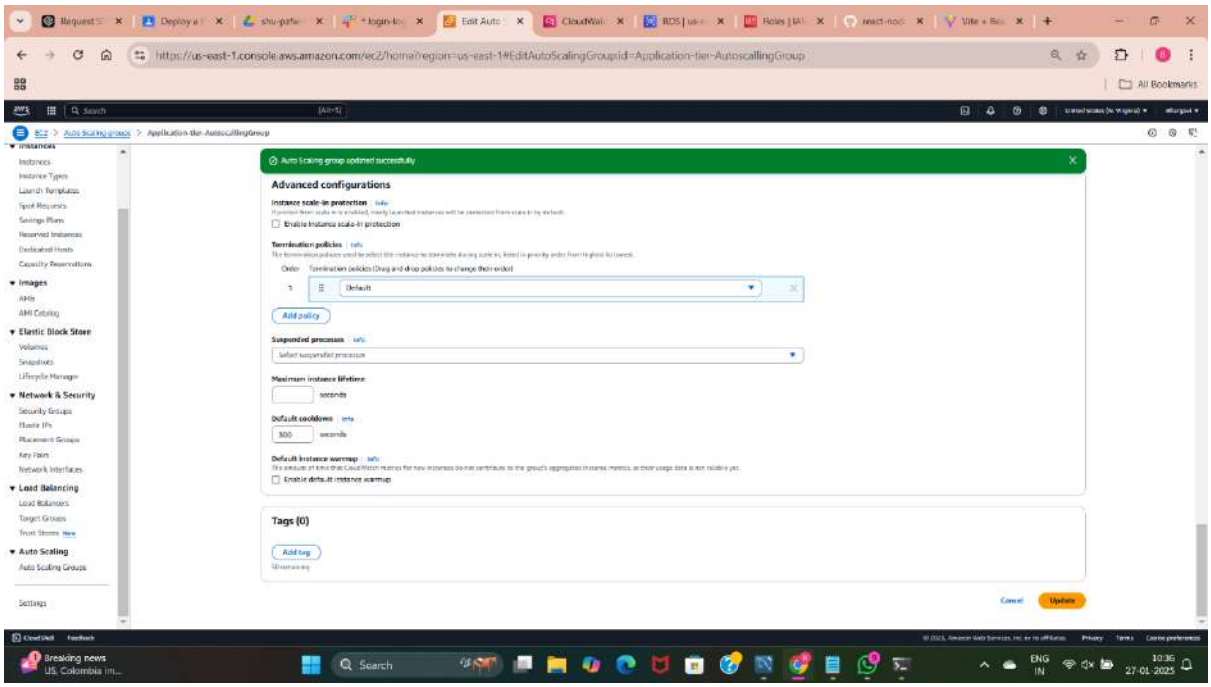


- Click Edit and update the Launch Template to use Version 3

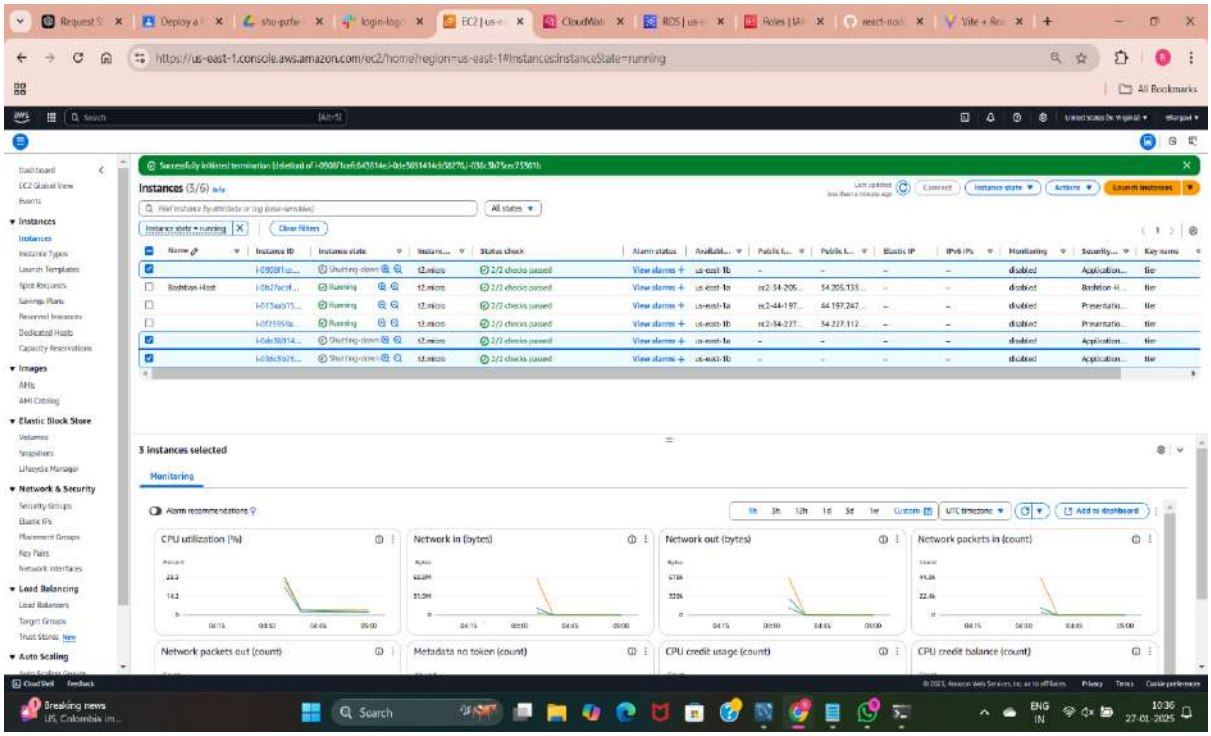


Update the version, Desired Capacity, Minimum, and Maximum Instances:

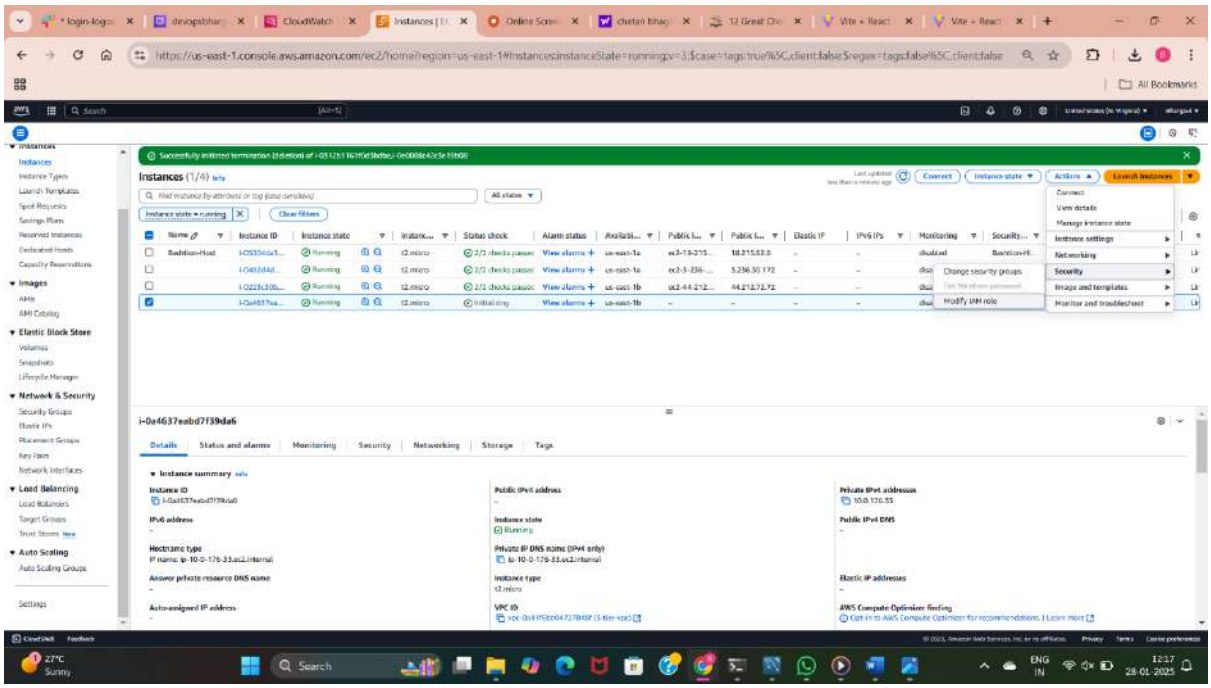
- Version: 3
- Desired Capacity: 2
- Minimum: 2
- Maximum: 4



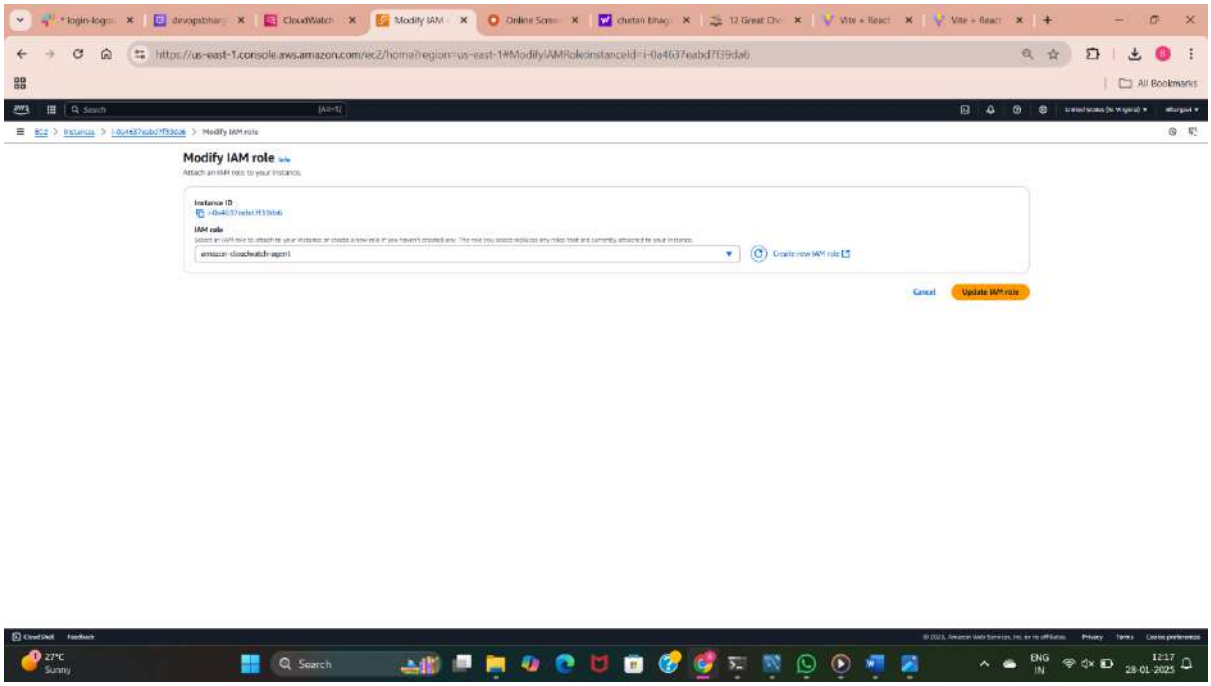
- Update the changes.



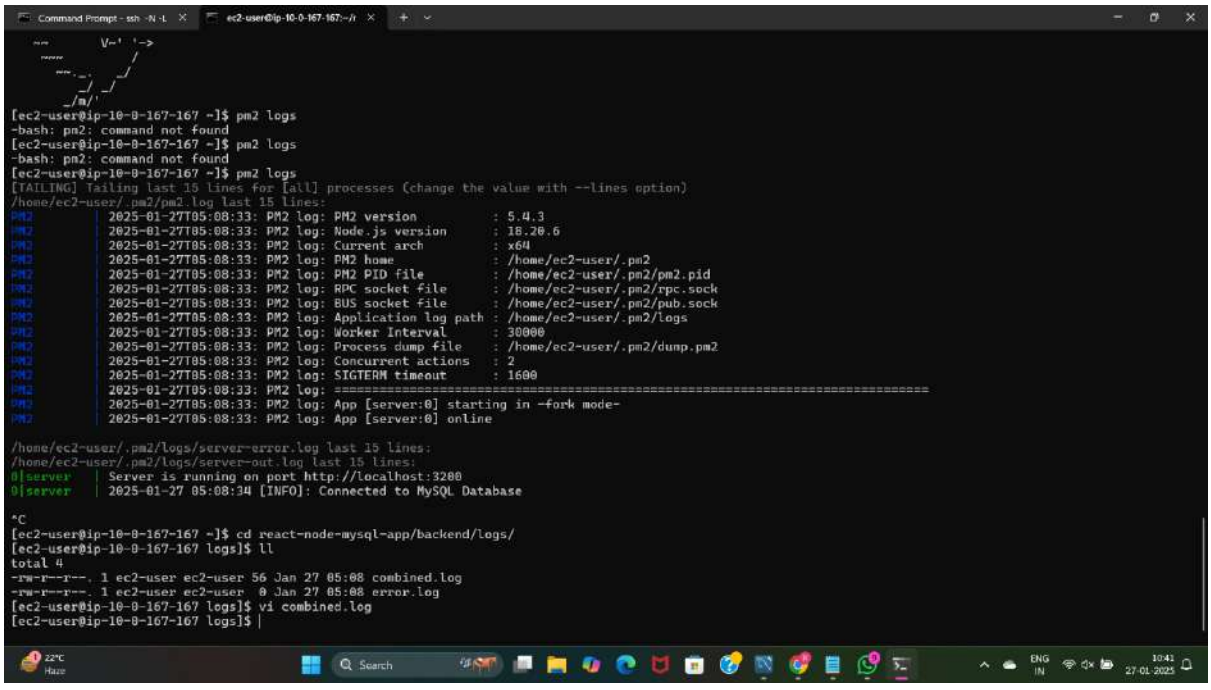
- Go to the Instances section in the EC2 Dashboard.
- Locate the old instances launched using Launch Template Version 1.
- Manually terminate these instances.



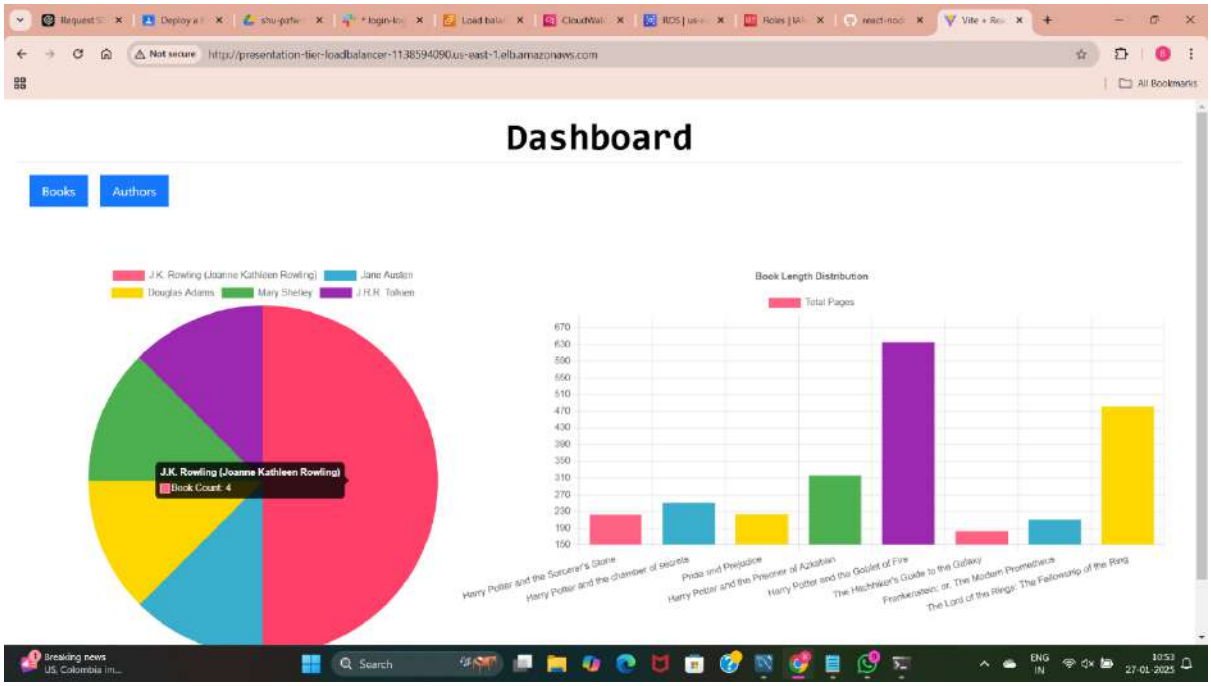
- The ASG will automatically replace them with new instances using Launch Template Version 3. Now attach the created IAM role by selecting actions>security> modify IAM role.



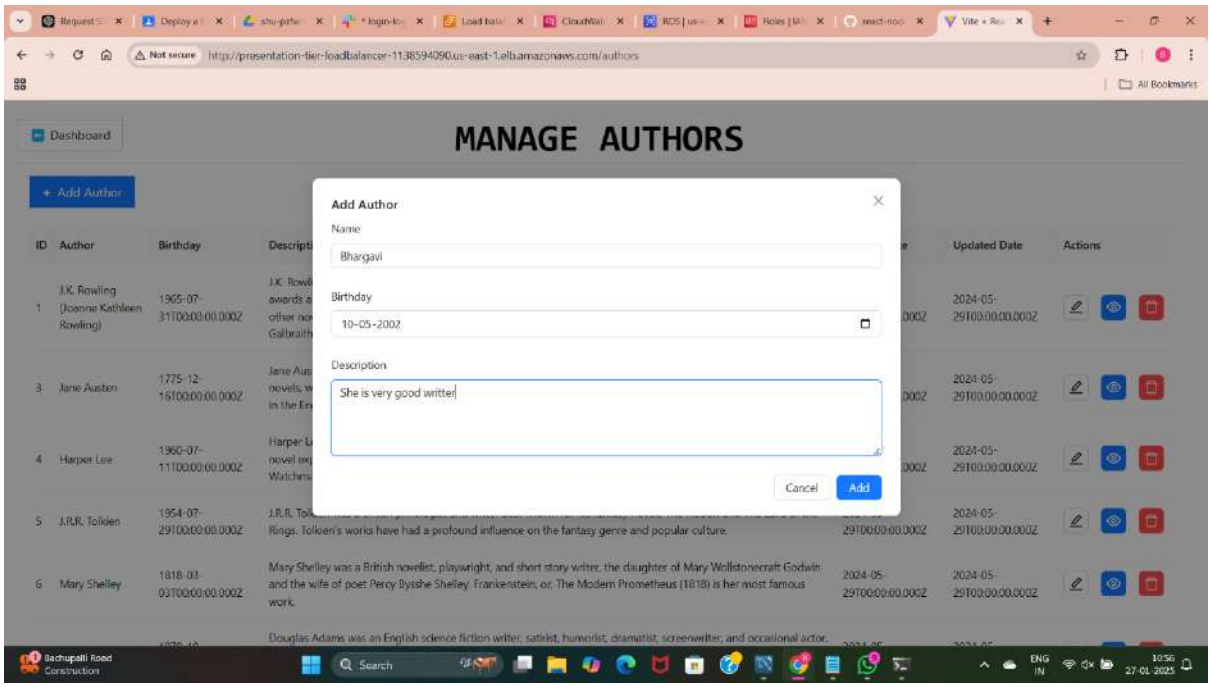
- Provide the name of IAM role and update IAM role. Similarly update the IAM role for other application tier ec2 instance.



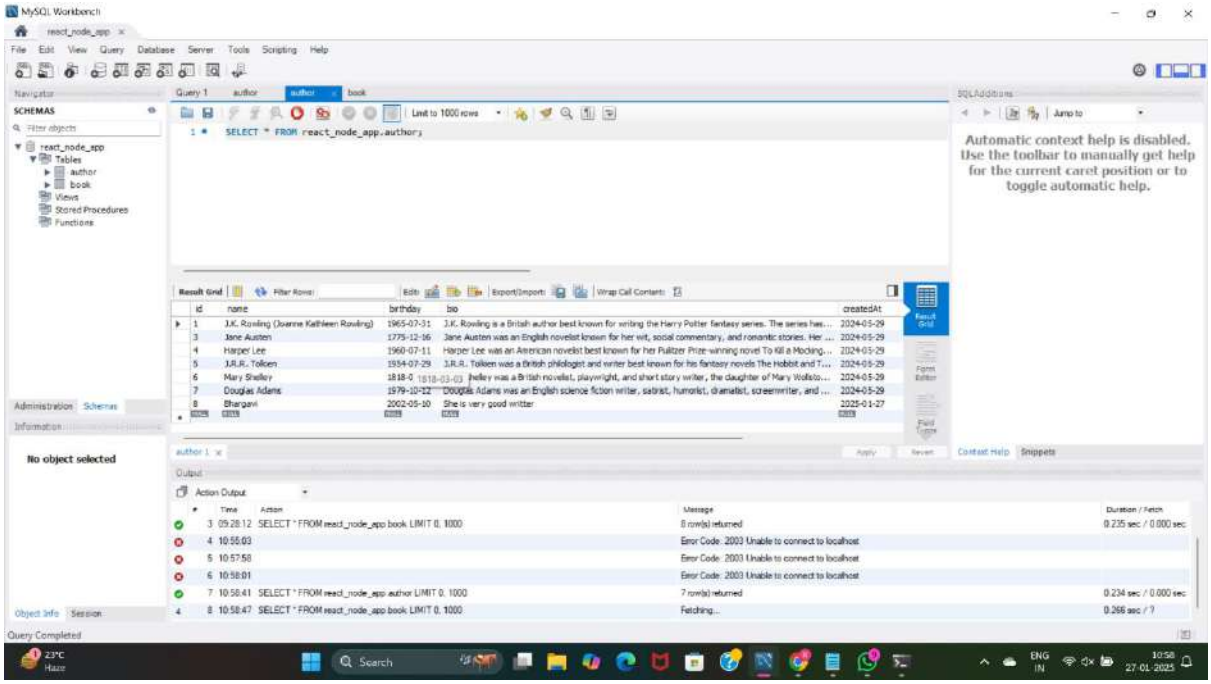
- Once the new instances are created, connect the application tier instance and check the database connection.



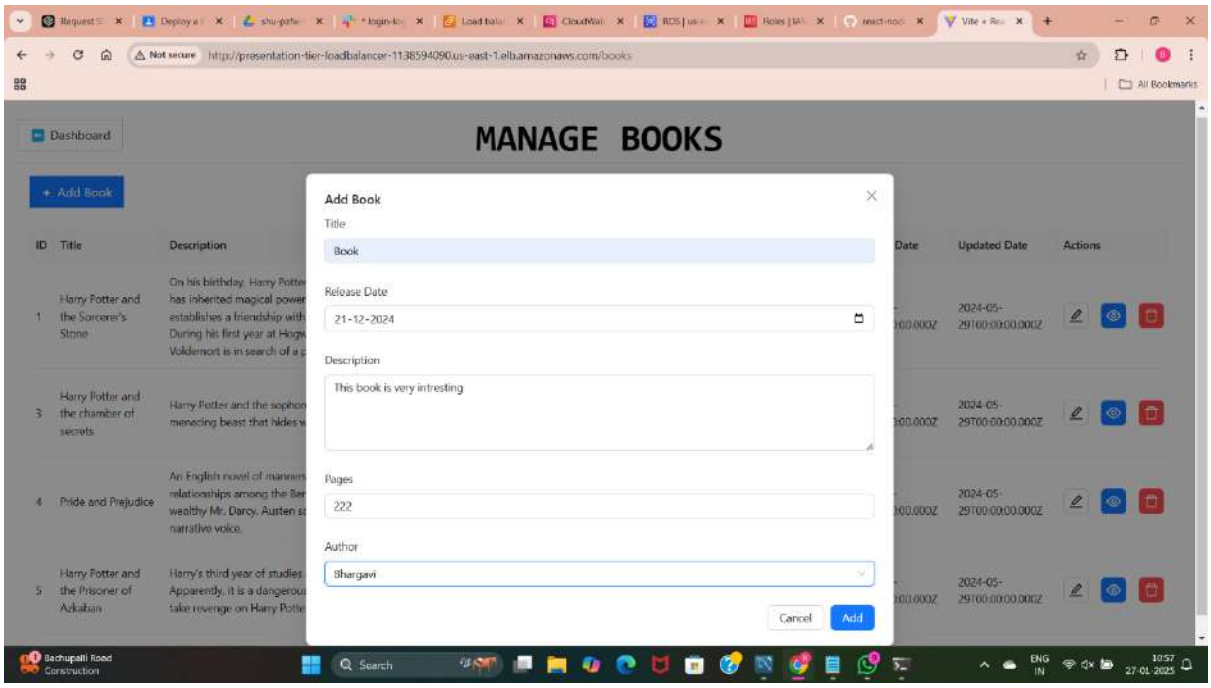
- Use the DNS Name of the Presentation Load Balancer (PLB) associated with the Application Tier.



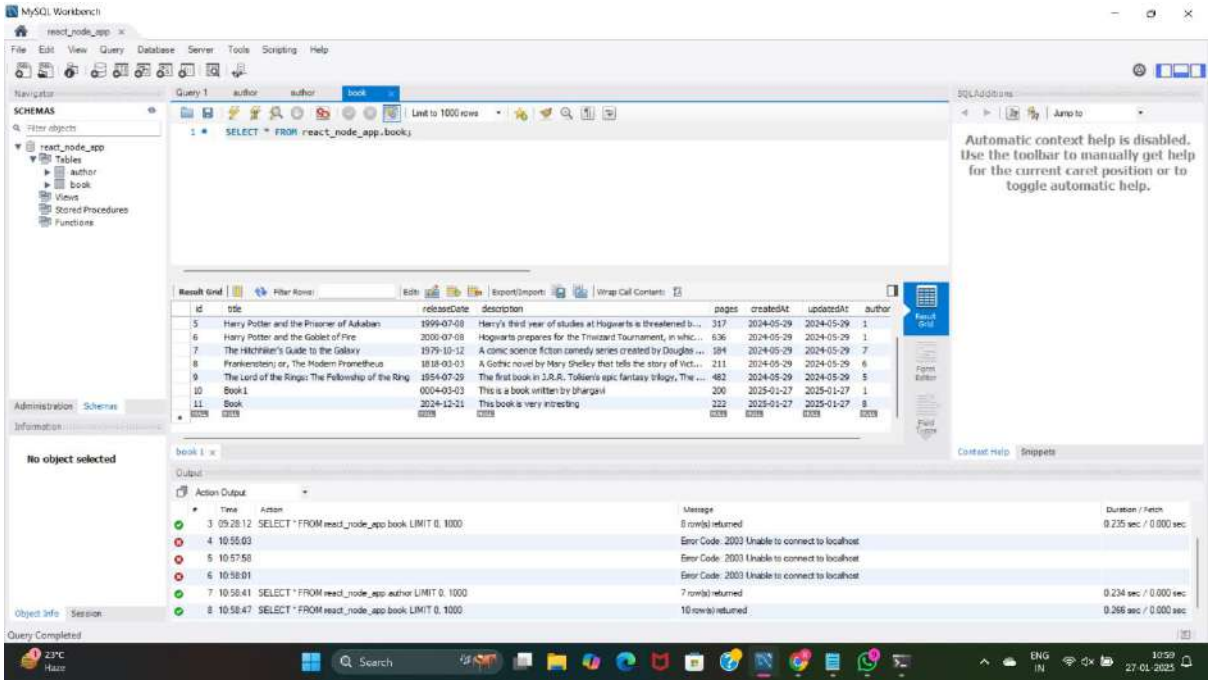
- Perform an operation like Add Authors to test the functionality of the backend application.



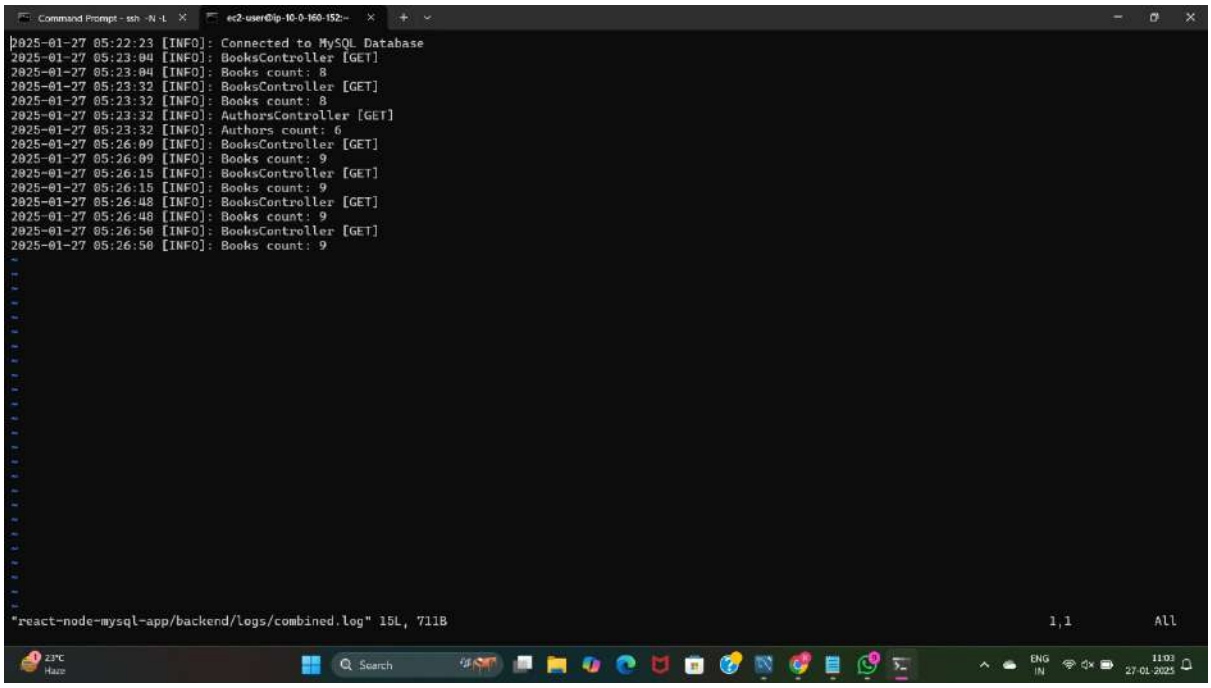
- Now go to mysql workbench and check the author name whether newly added author is present or not.



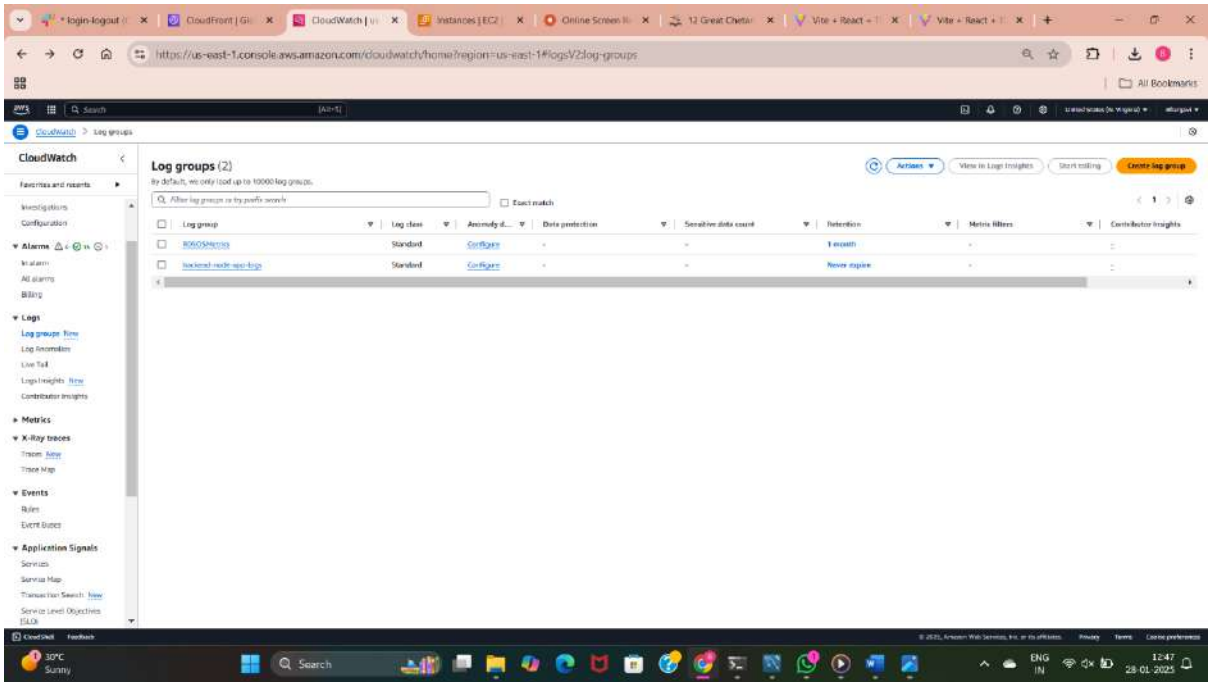
- Perform an operation like Add Book to test the functionality of the backend application.



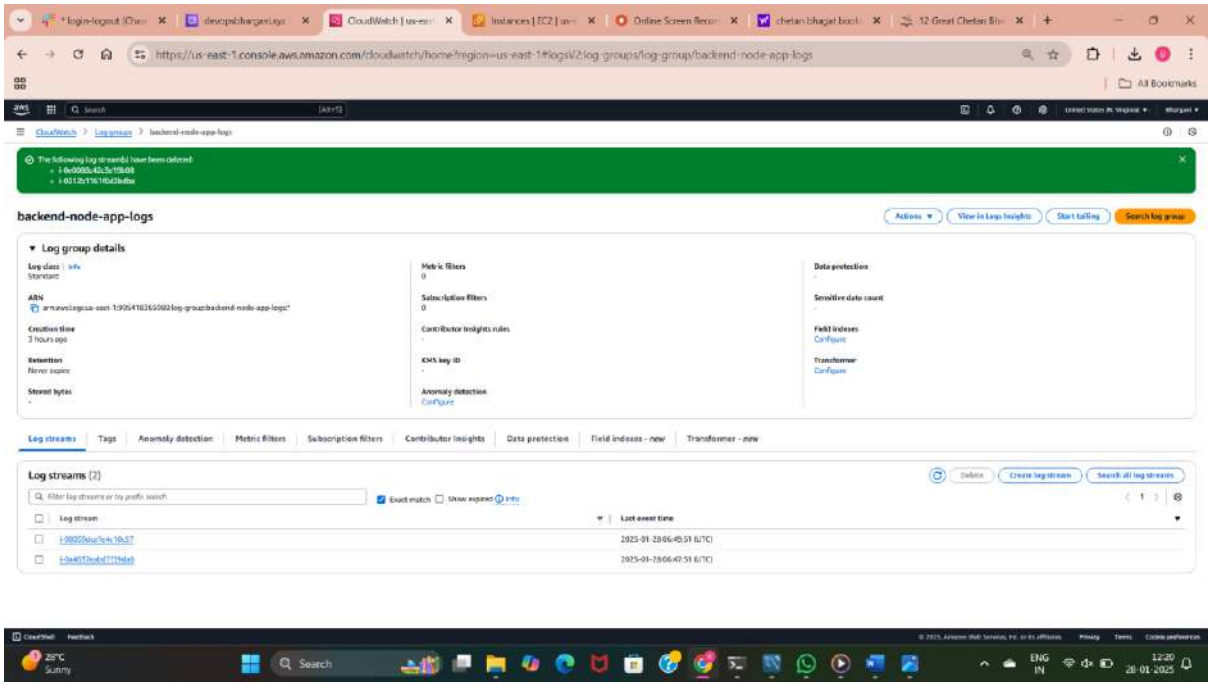
- Now go to mysql workbench and check the book names whether newly added books is present or not.



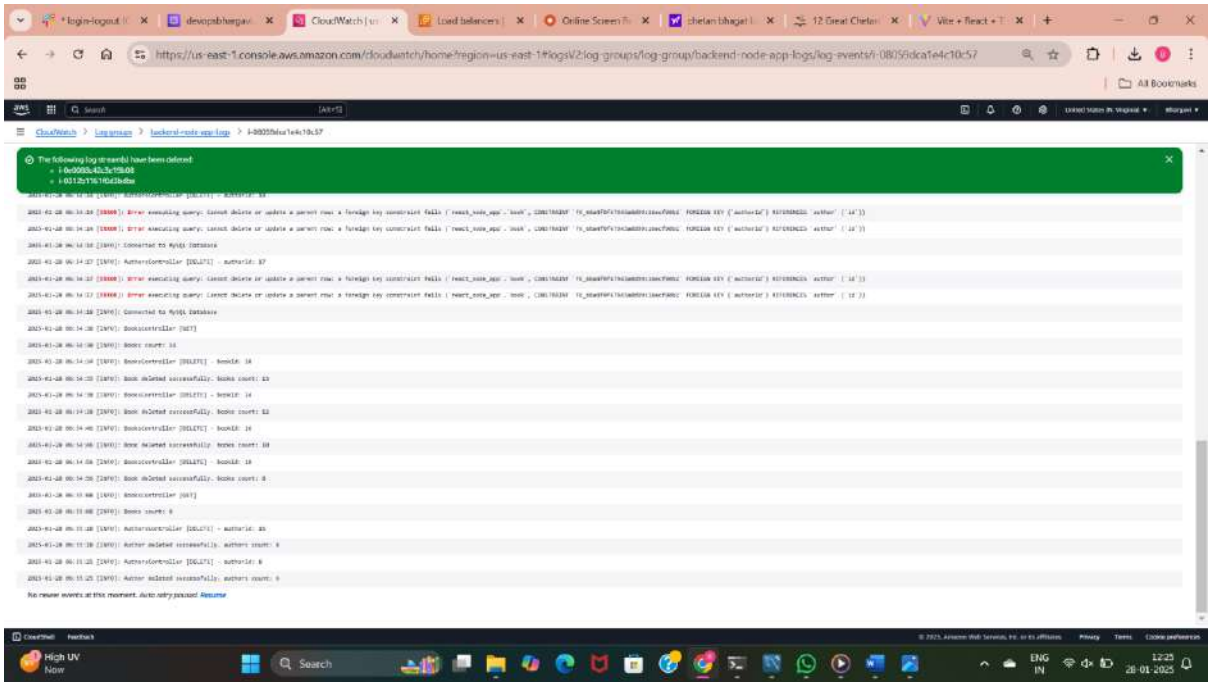
- Look for the log group you created for the backend application, e.g., backend-node-app-logs.
- Verify that logs are being recorded for operations performed on the application (e.g., adding a book).



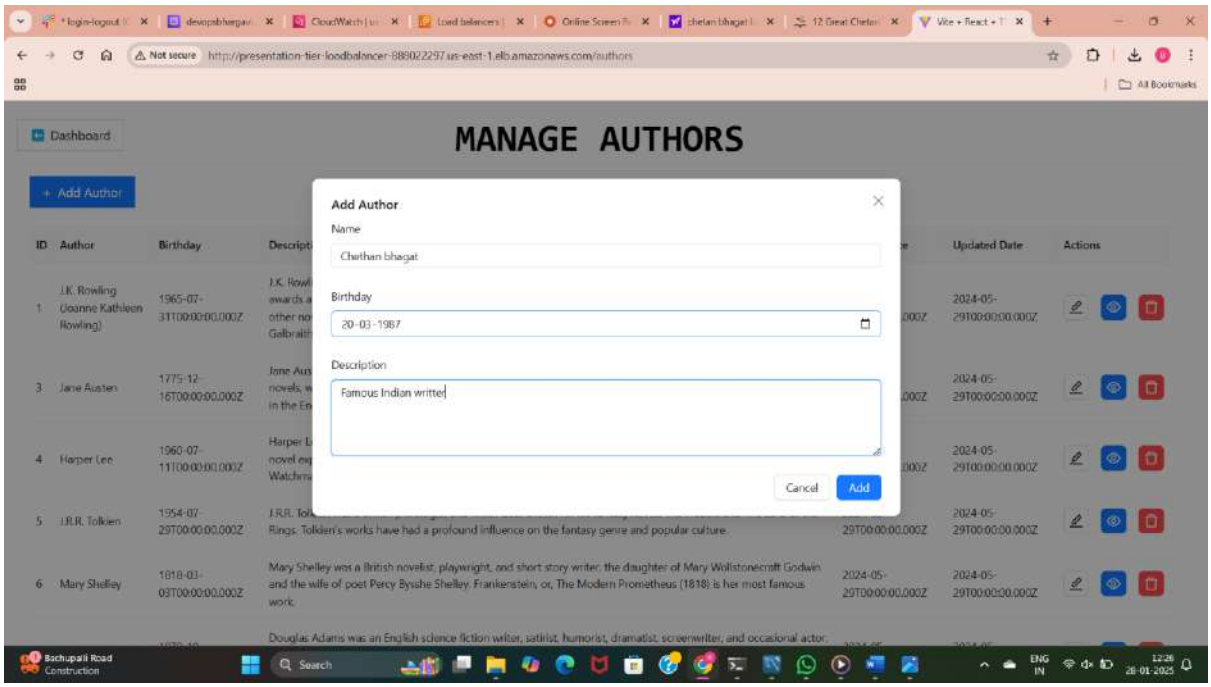
- To see the logs for the new instances. Go to CloudWatch in the AWS Management Console.
- Under Logs, navigate to the log group that you've configured for your application (e.g., backend-node-app-logs).



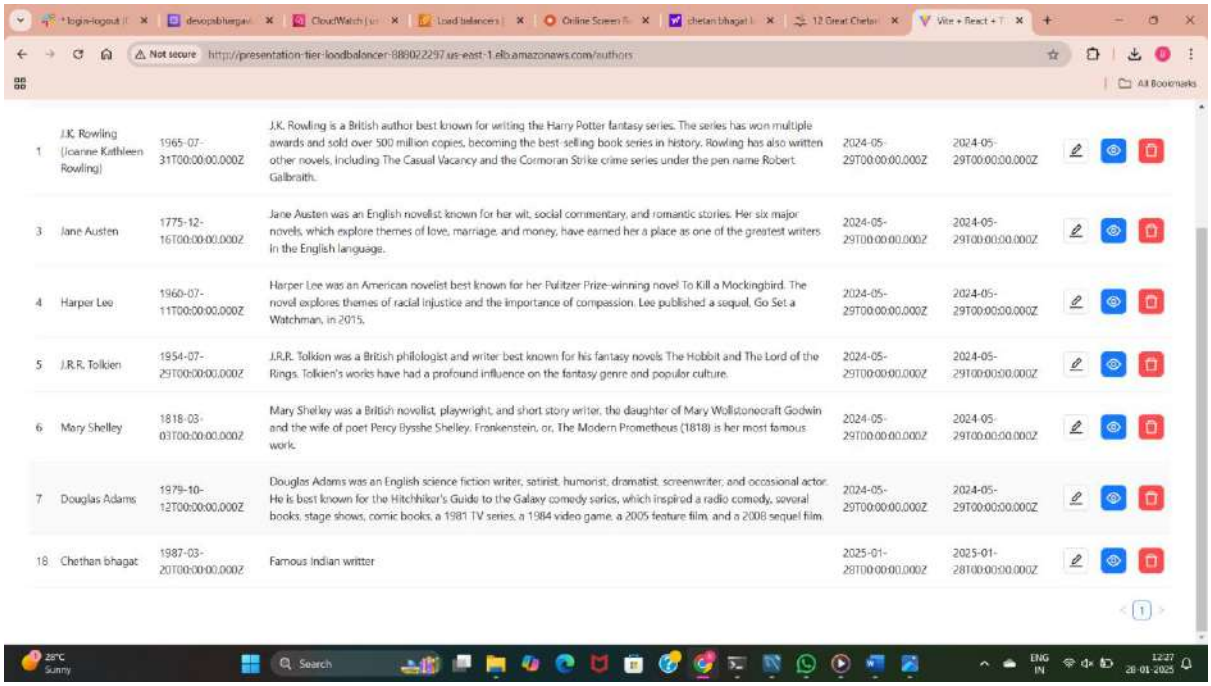
- Look for the two new instances created by the Auto Scaling Group. These should be associated with the same ASG.
- We can see two instance IDs.



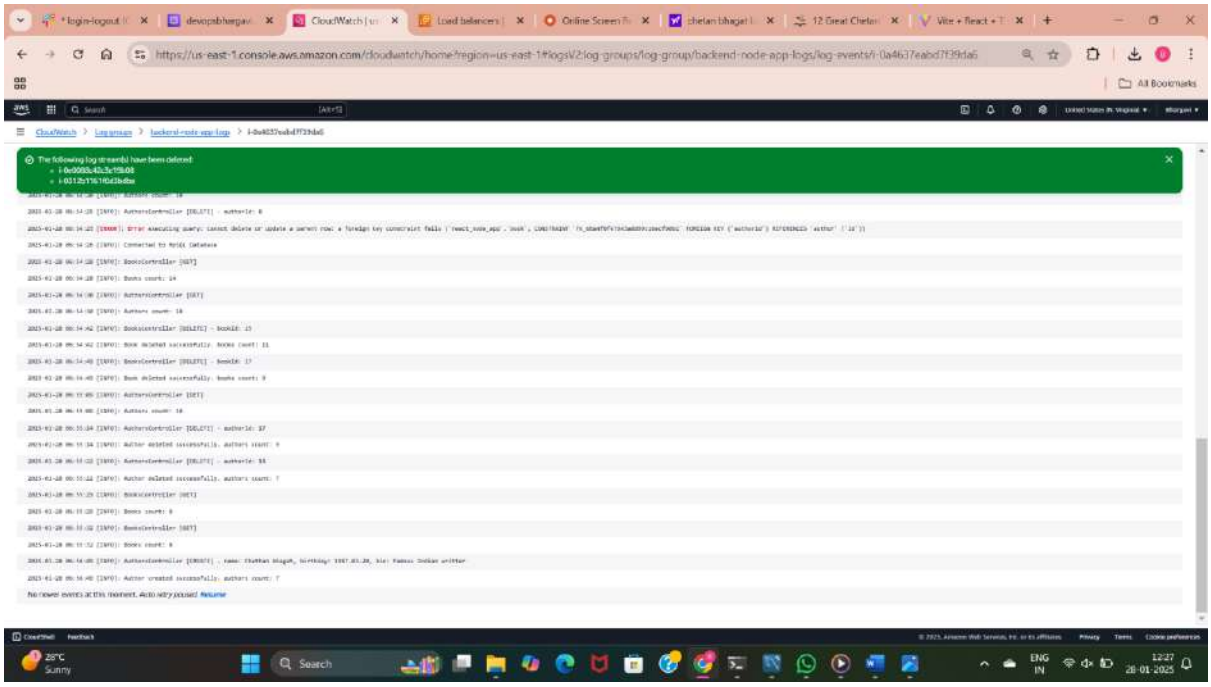
- Click on an instance to go to the instance details page. Under the Monitoring tab, look for CloudWatch Logs or any log streaming service configured on the instance.



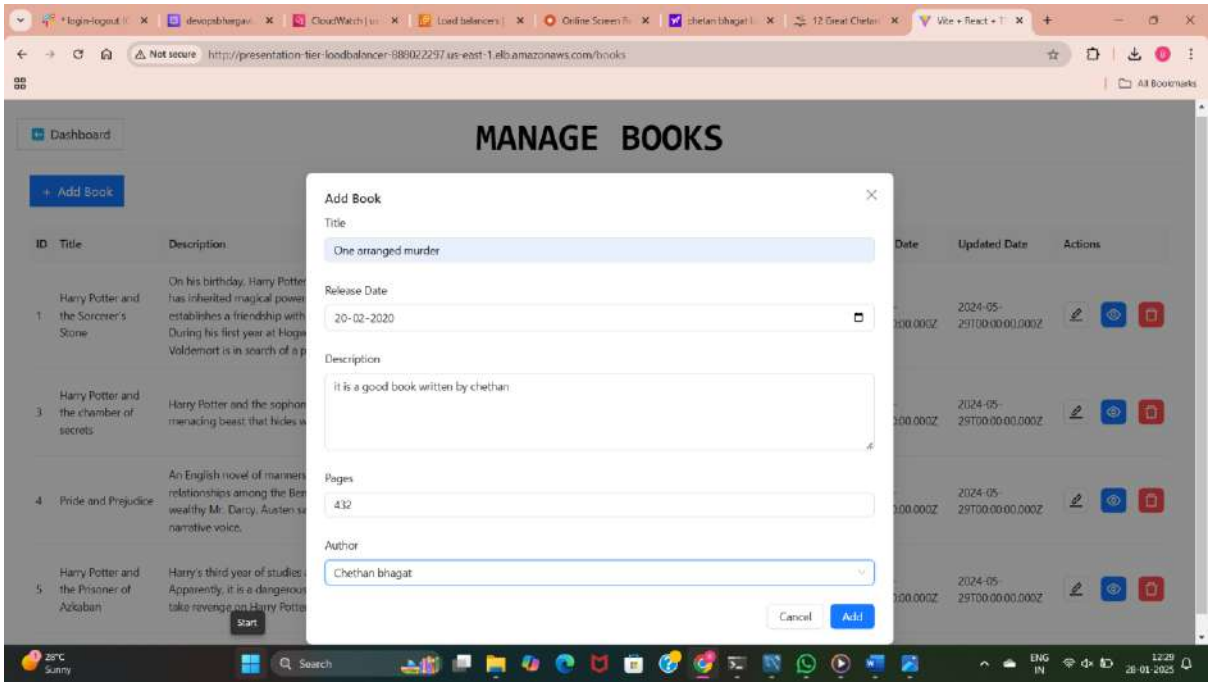
- Similarly am adding other author for checking the application and database.



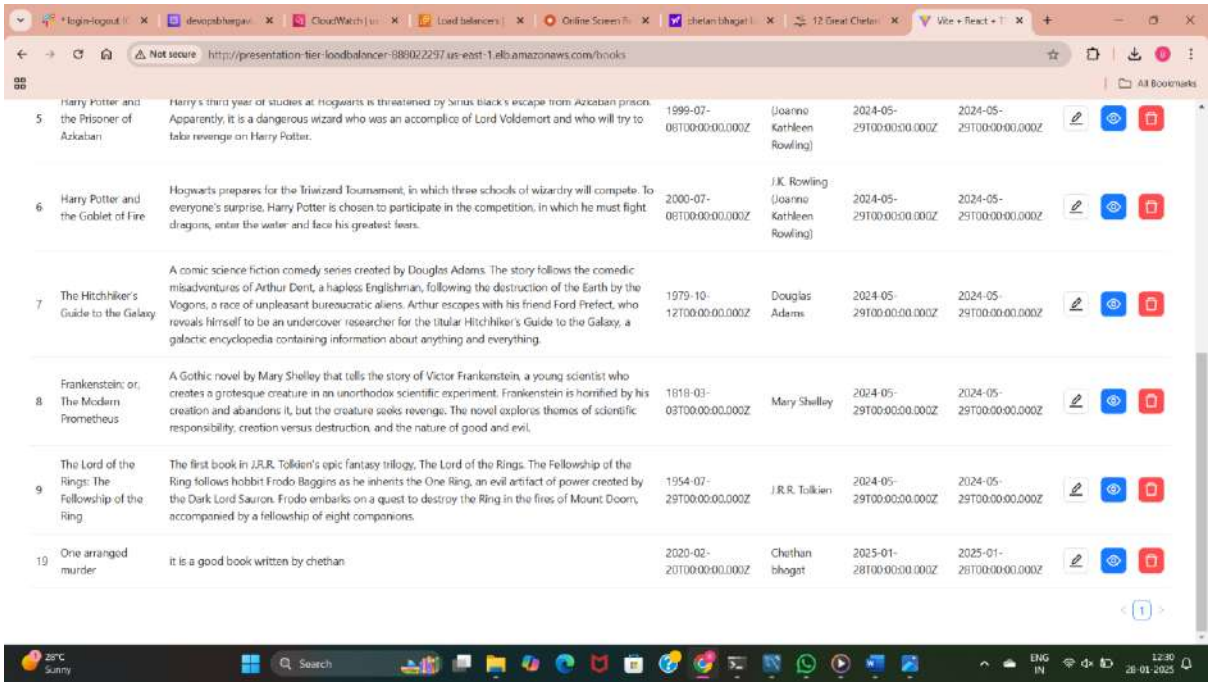
- Here new author is added.



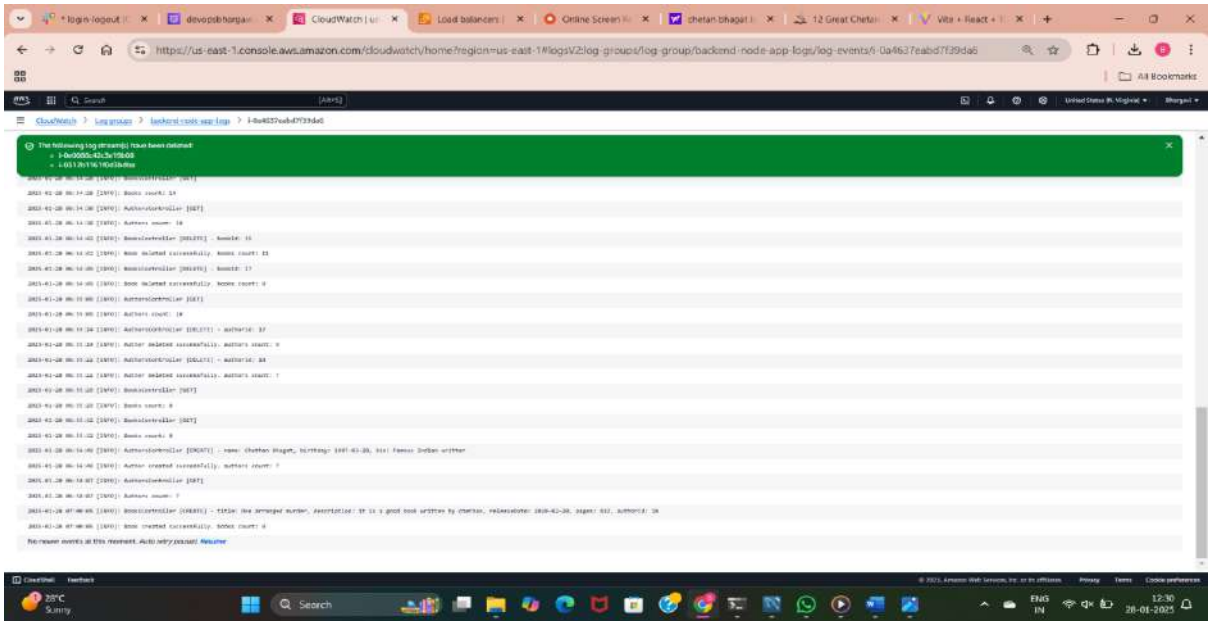
- We can also check in cloud watch whether the updates are monitoring or not. Here cloud watch is monitoring everything related to database updates and deletions.



- Add other book.



- Book is updated here.



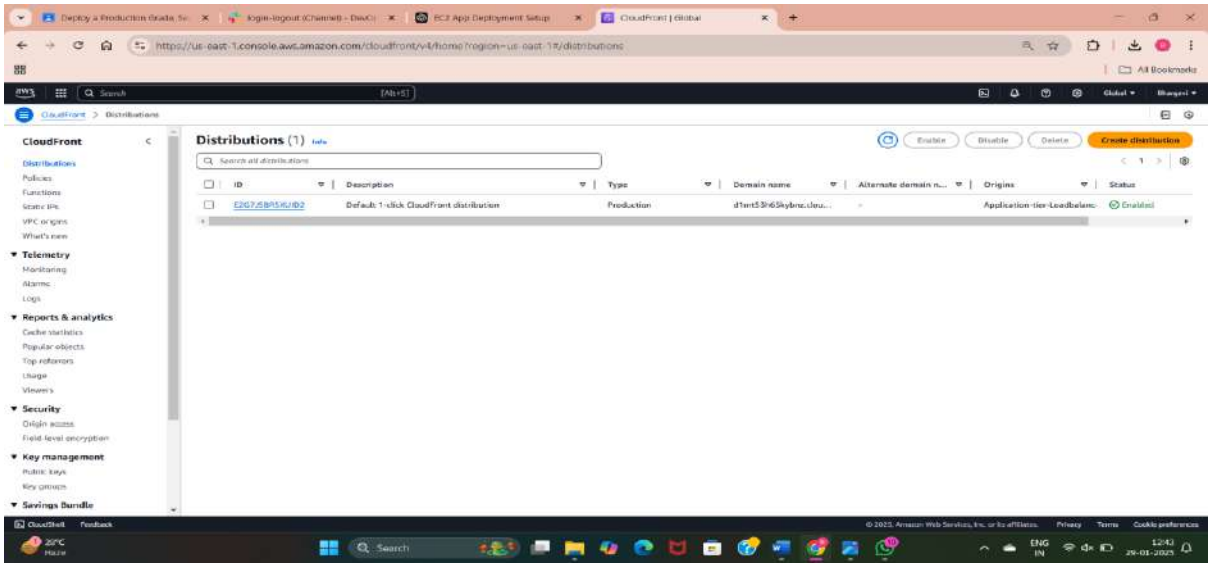
- ❖ Similarly in cloud watch we can check the updates of new book.

CLOUDFRONT DISTRIBUTION

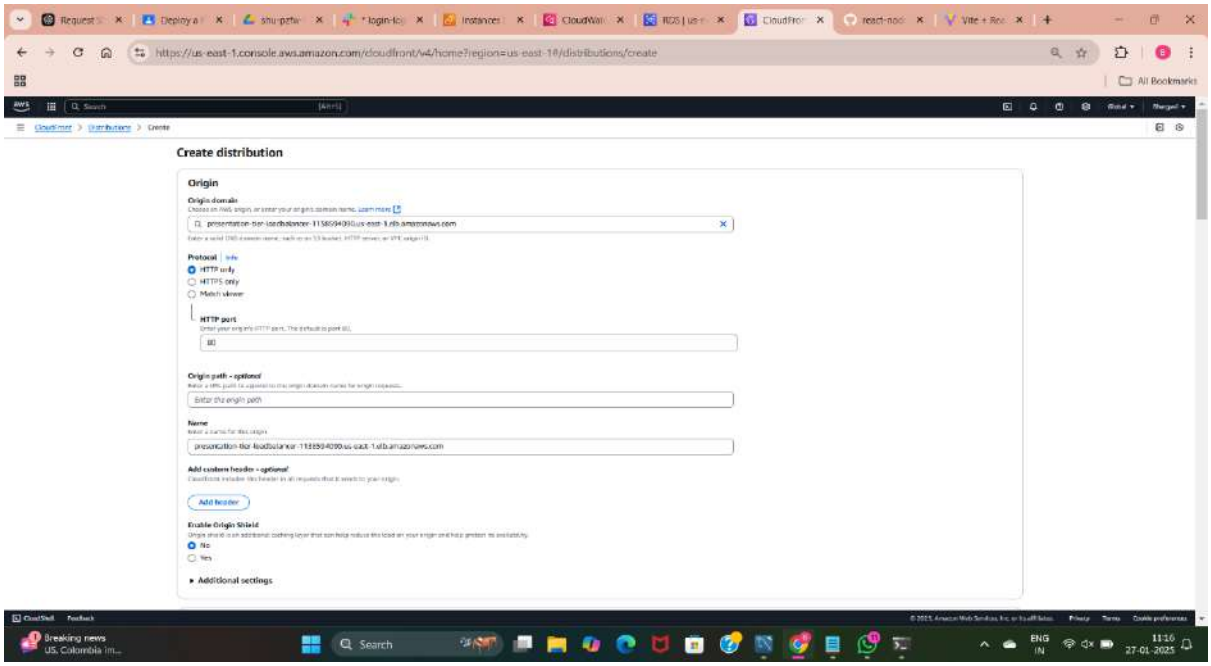
CloudFront Distribution is a feature of Amazon Web Services (AWS) that enables the delivery of content to users globally with low latency and high transfer speeds. It is part of the AWS Content Delivery Network (CDN) service. A CloudFront Distribution is essentially a configuration that tells CloudFront how to distribute your content, such as static files (images, JavaScript, CSS) or dynamic content (like API calls or HTML pages), to users via AWS's global network of edge locations.

STEP 12: CREATING CLOUDFRONT DISTRIBUTION

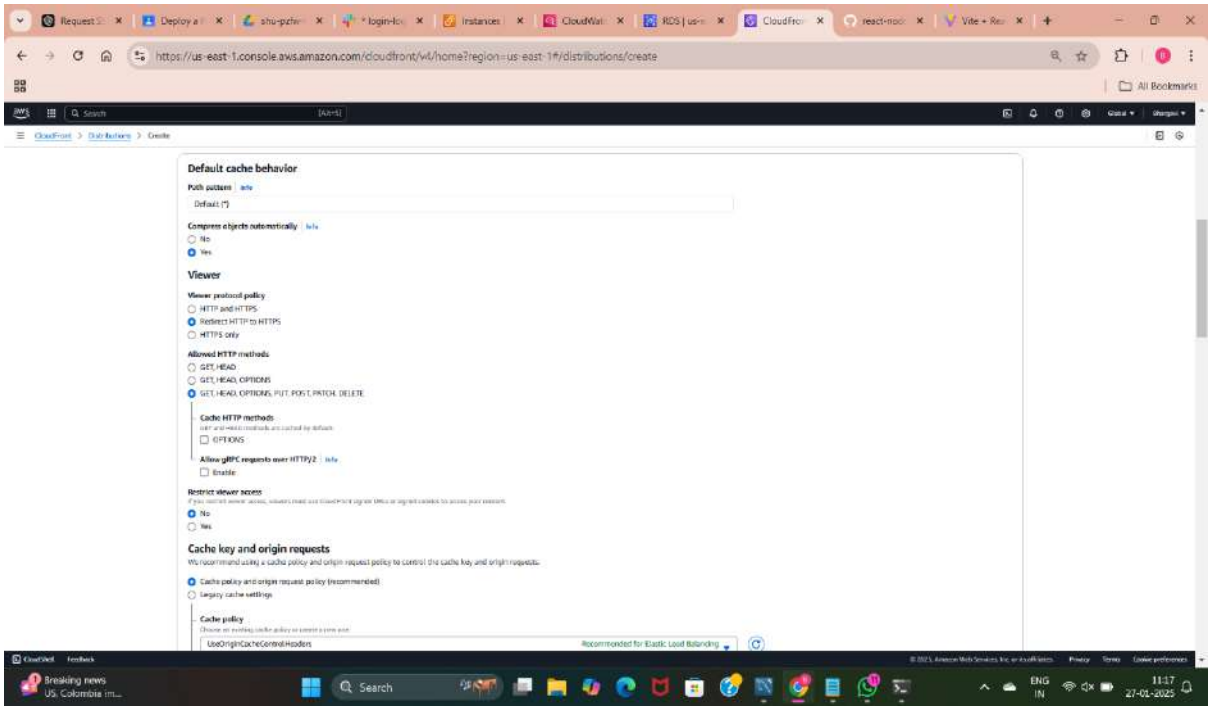
- ☑ Open the AWS Management Console.
- ☑ Go to CloudFront under the Networking and Content Delivery section.



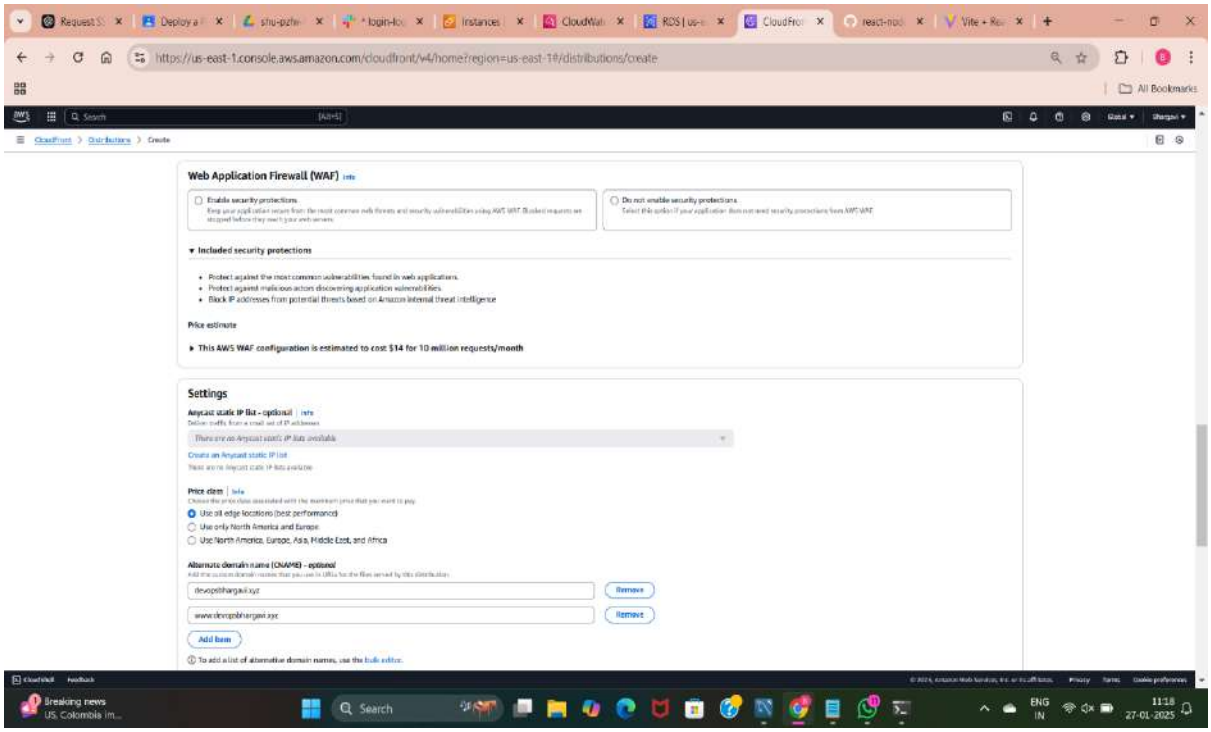
- ☑ Click on Create Distribution.



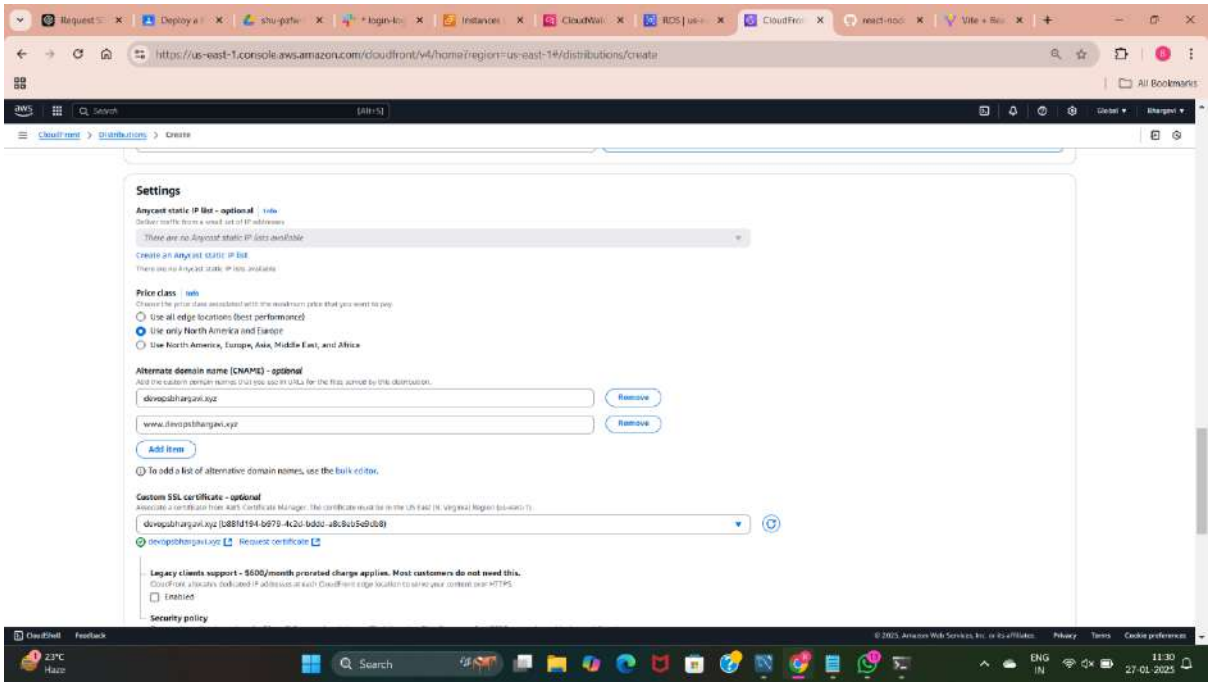
- ☑ Origin Domain: Select the DNS Name of the Presentation-Tier ALB from the dropdown or enter it manually.
- ☑ Protocol: HTTP



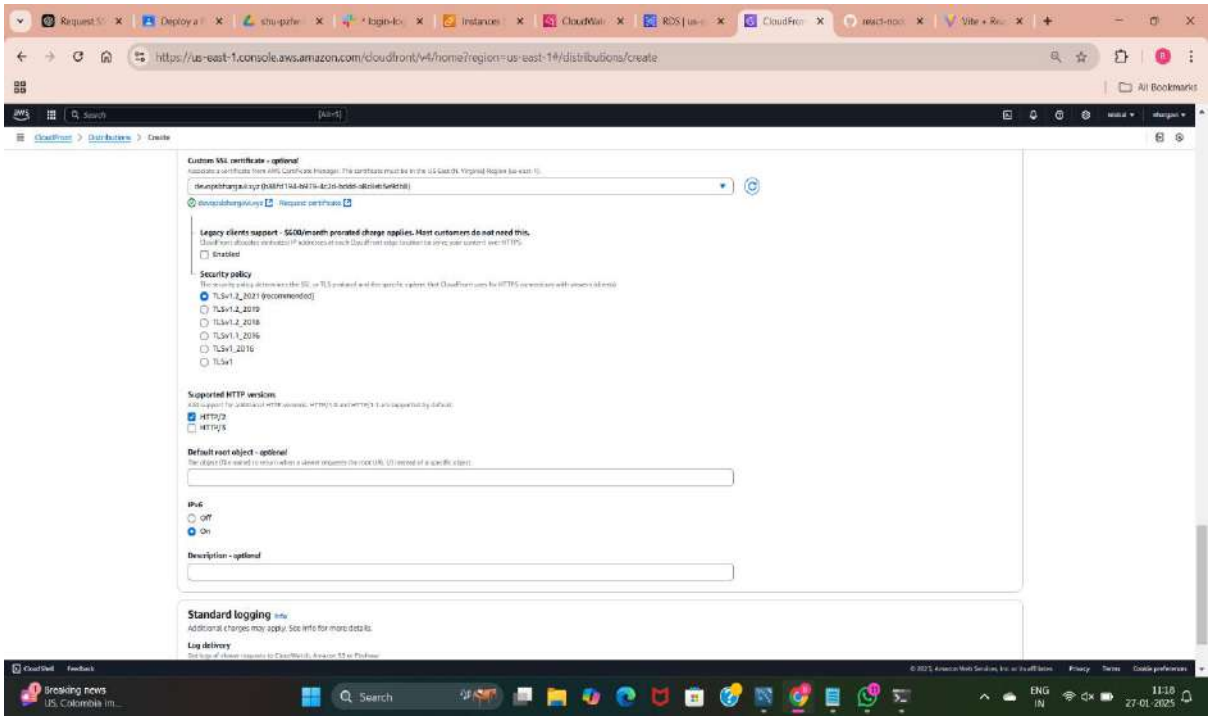
- ☑ Protocol Policy: Select Redirect HTTP to HTTPS to ensure secure traffic only.



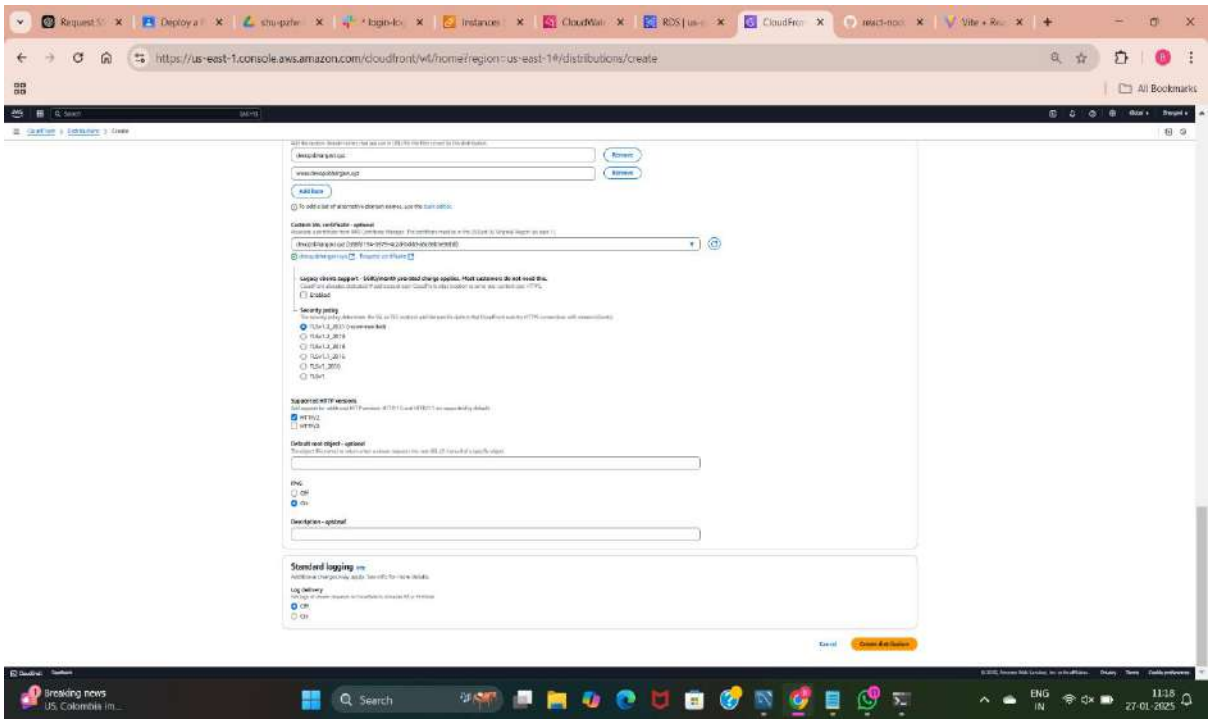
- ☑ Select Do Not Enable WAF unless you require specific security configurations.
- ☑ Choose the required AWS regions where your content will be distributed.



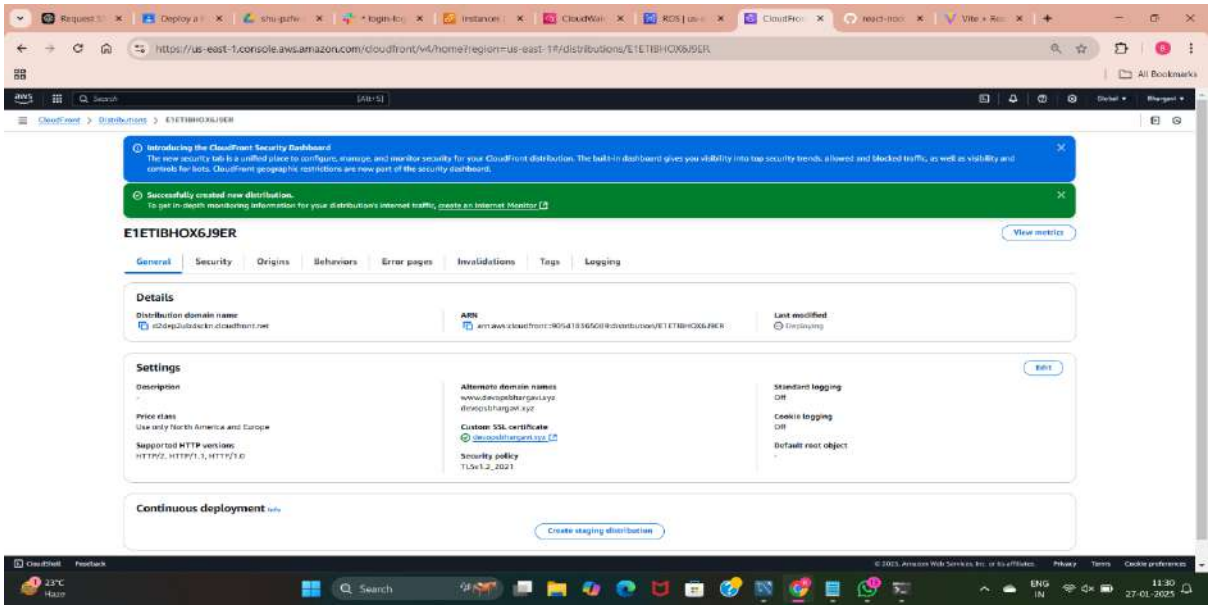
- ☑ Under the Settings section, provide alternate domain names(CNAME)
- ☑ Add devopsdost.xyz and www.devopsbhargavi.xyz.



☑ Choose a certificate from ACM (AWS Certificate Manager) that matches your domain names (devopsdost.xyz and www.devopsbhargavi).



☑ Click Create Distribution to finalize the setup.

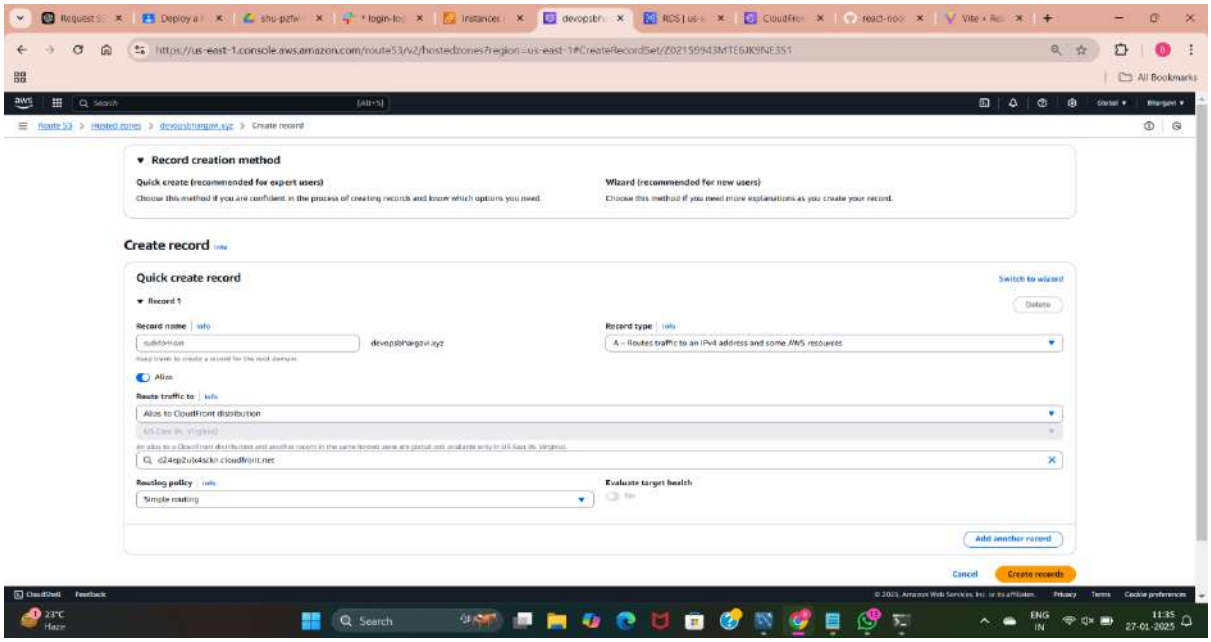


☑ Wait for the distribution status to change to Deployed (this may take several minutes).

STEP 13: CREATE DNS RECORDS FOR CLOUDFRONT IN ROUTE53

- ♣ Open the AWS Management Console.
- ♣ Go to Route53 under the Networking and Content Delivery section.
- ♣ Open your Hosted Zone for the domain devopsdost.xyz.
- ♣ Click on Create Record

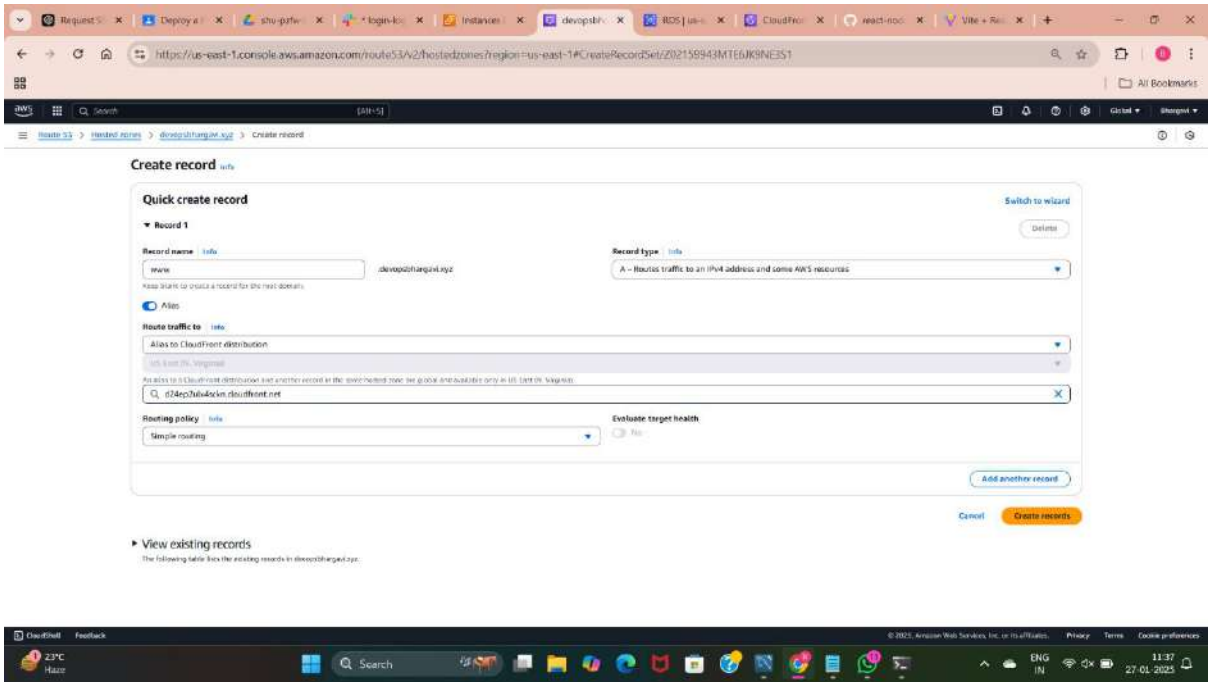
ADD AN ALIAS RECORD FOR THE ROOT DOMAIN



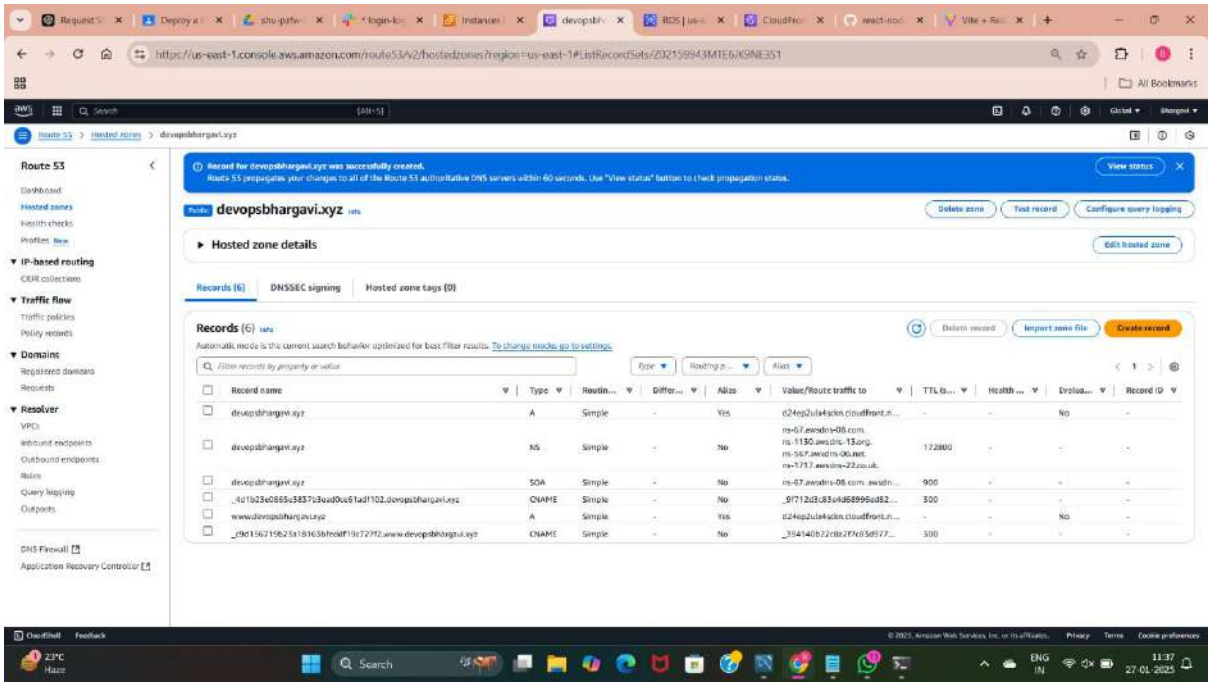
- ♣ Record Type: Select A - IPv4 Address.
- ♣ Alias: Select Yes.
- ♣ Choose route traffics to AWS to CloudFront distributions.
- ♣ Select your CloudFront distribution.
- ♣ Save the record.

ADD AN ALIAS RECORD FOR THE WWW SUBDOMAIN

- ♣ Click on Add Record.



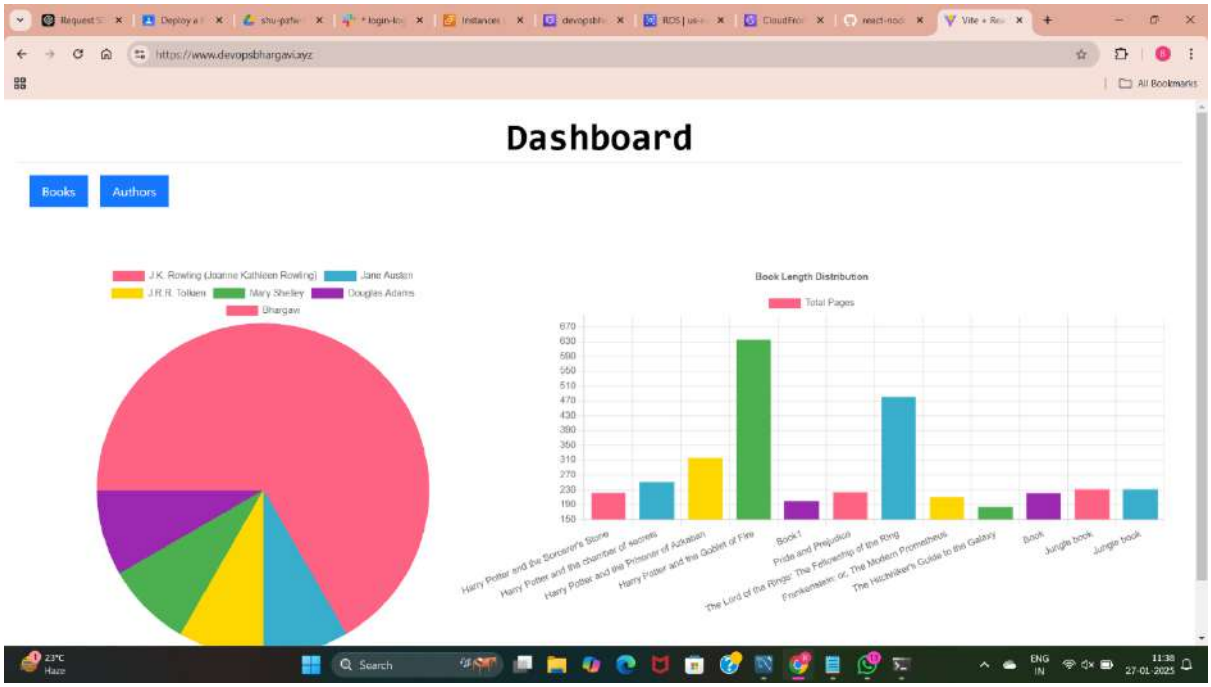
- ♣ Name: Enter www.
- ♣ Record Type: Select A - IPv4 Address.
- ♣ Alias: Select Yes.
- ♣ Choose route traffics to AWS to CloudFront distributions.
- ♣ Select your CloudFront distribution.
- ♣ Save the record.



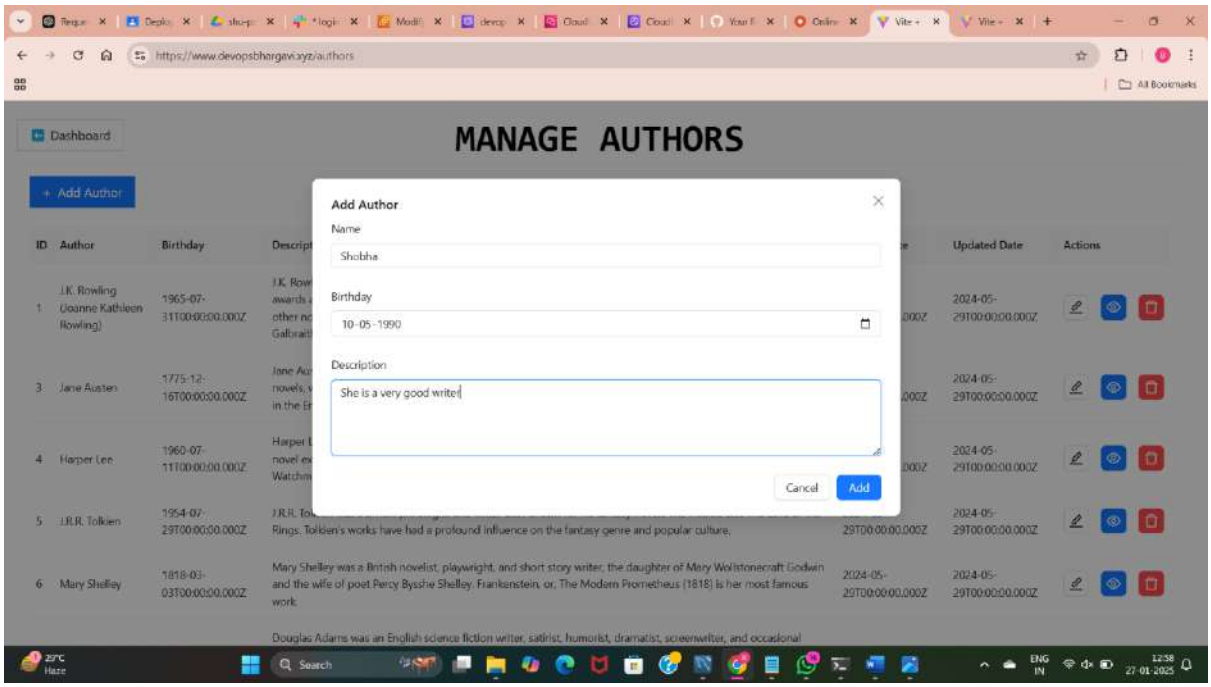
- ♣ Here we can see new records are added to our hosted zones.

STEP 14: TEST THE APPLICATION

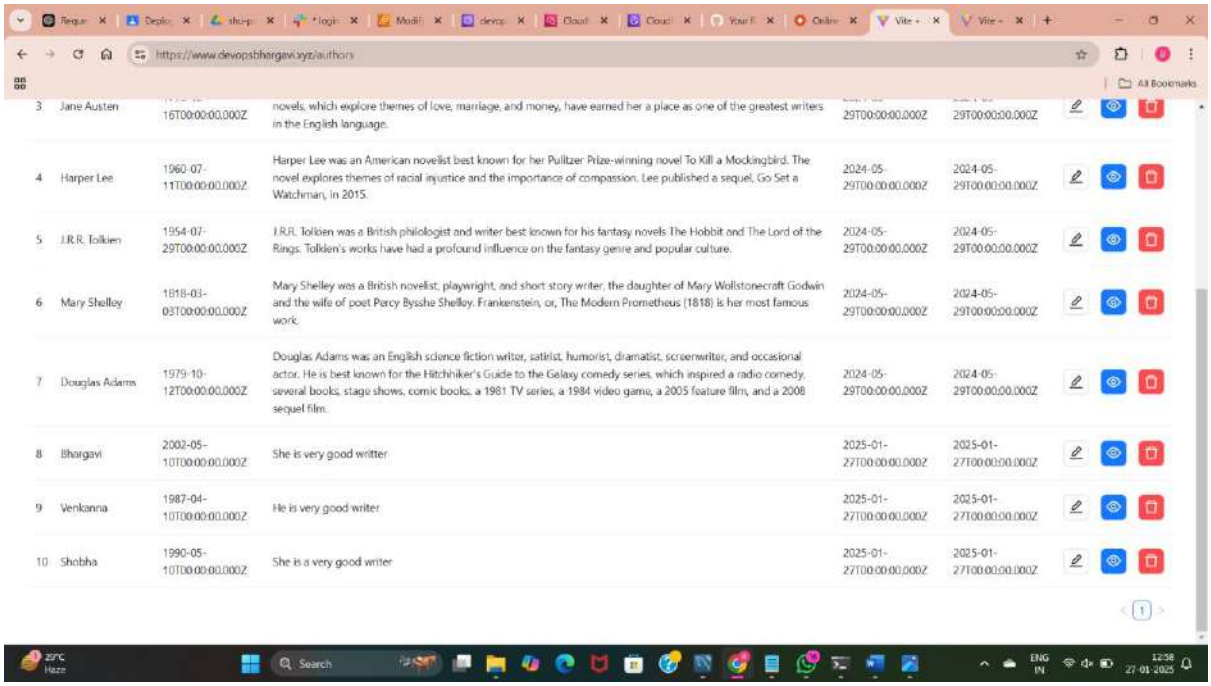
- Open a web browser.



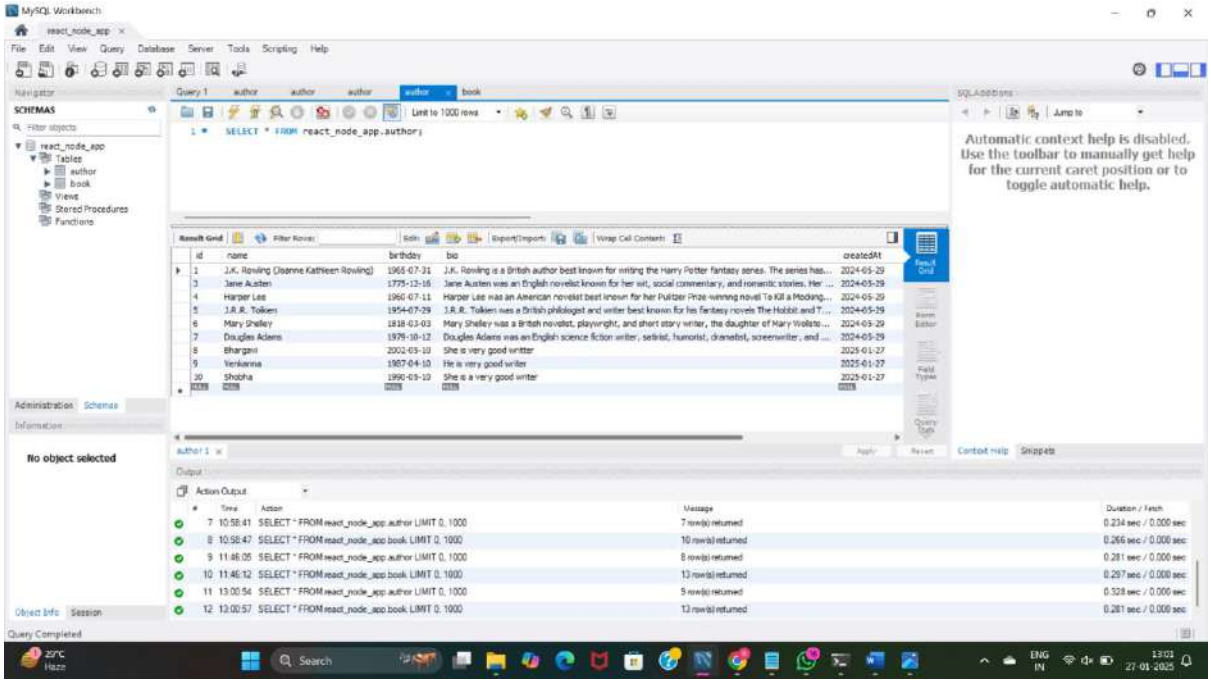
- Navigate to your domain www.devopsbhargavi.xyz and devopsbhargavi.xyz.
- Verify the homepage loads as expected.
- Ensure the site is loading over HTTPS with a secure padlock icon in the browser.



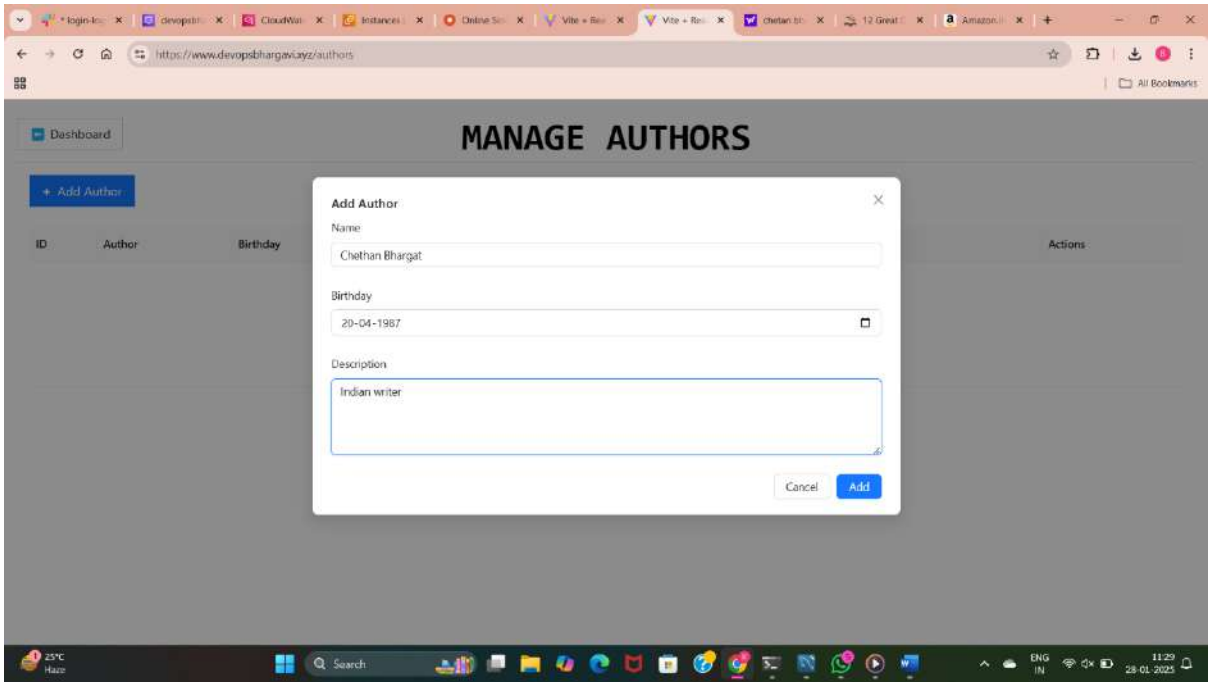
- Navigate through different pages of the application.
- Add a author (or any operation) in the application.



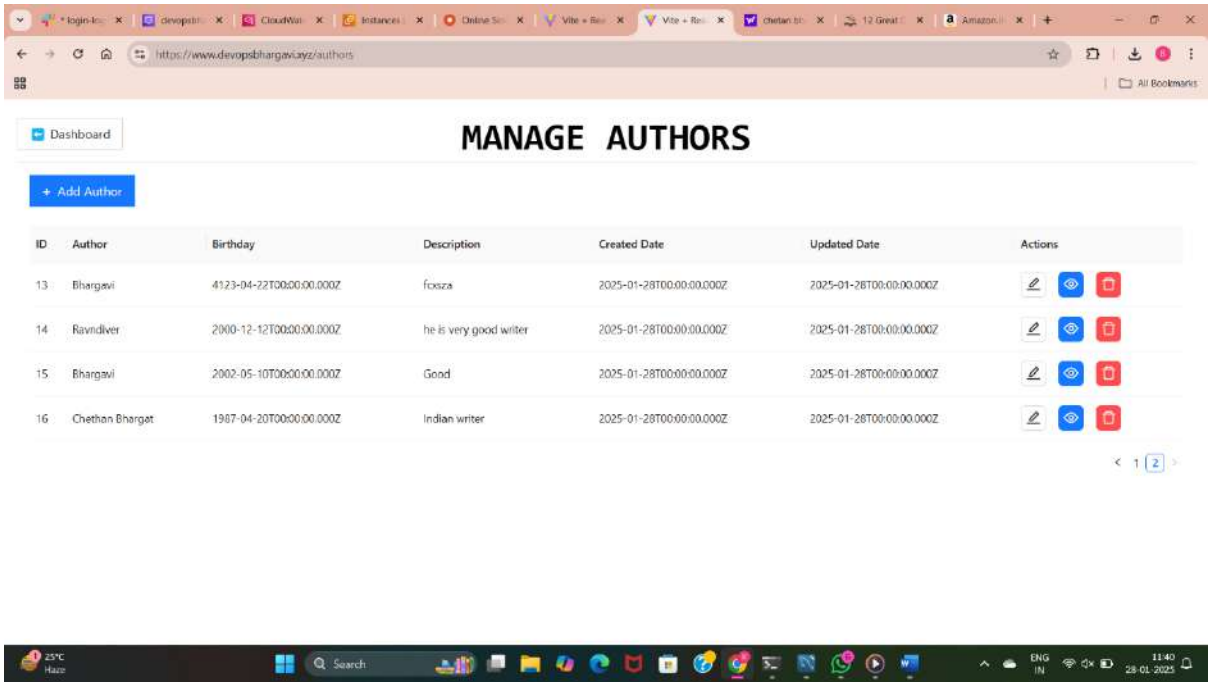
- Author is added successfully.



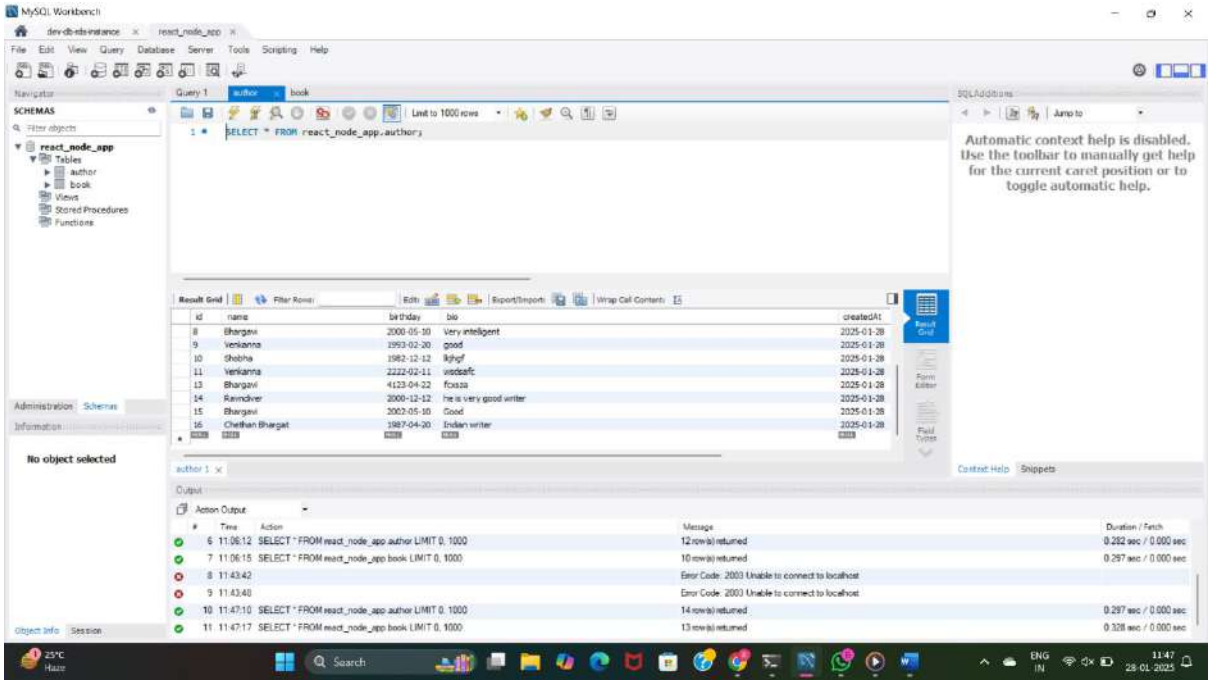
- Check in the mysql workbench by selecting the author.



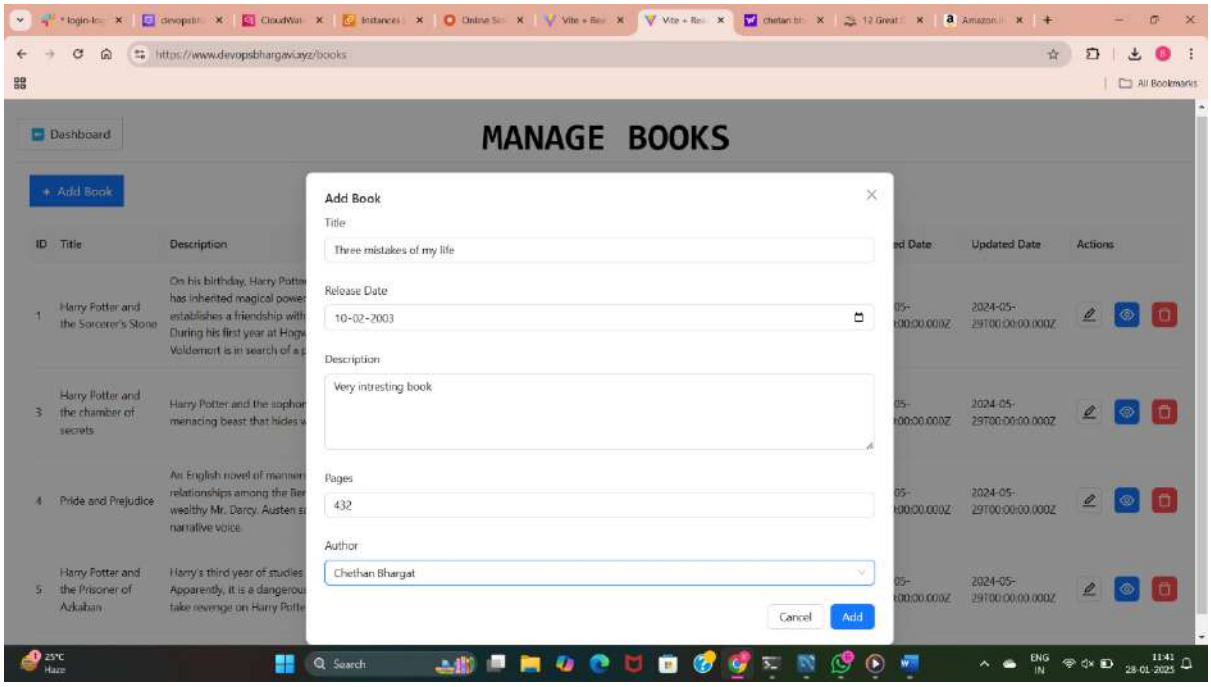
❖ Add a book (or any operation) in the application.



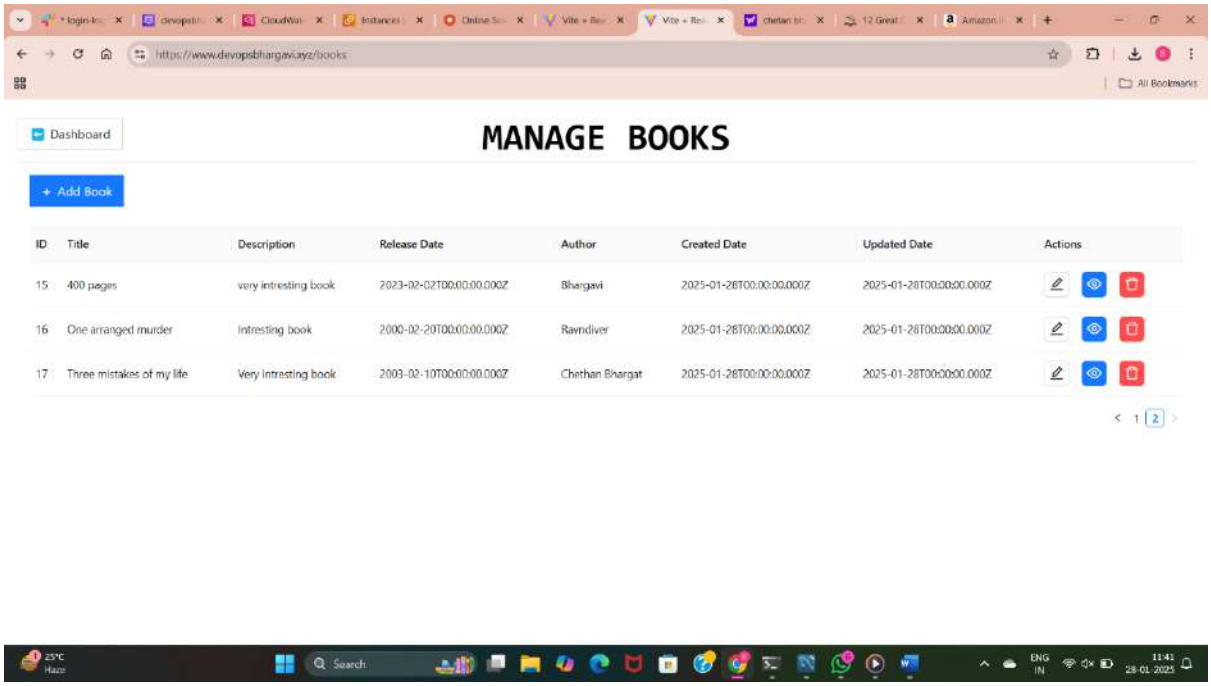
❖ Here the new book is added.



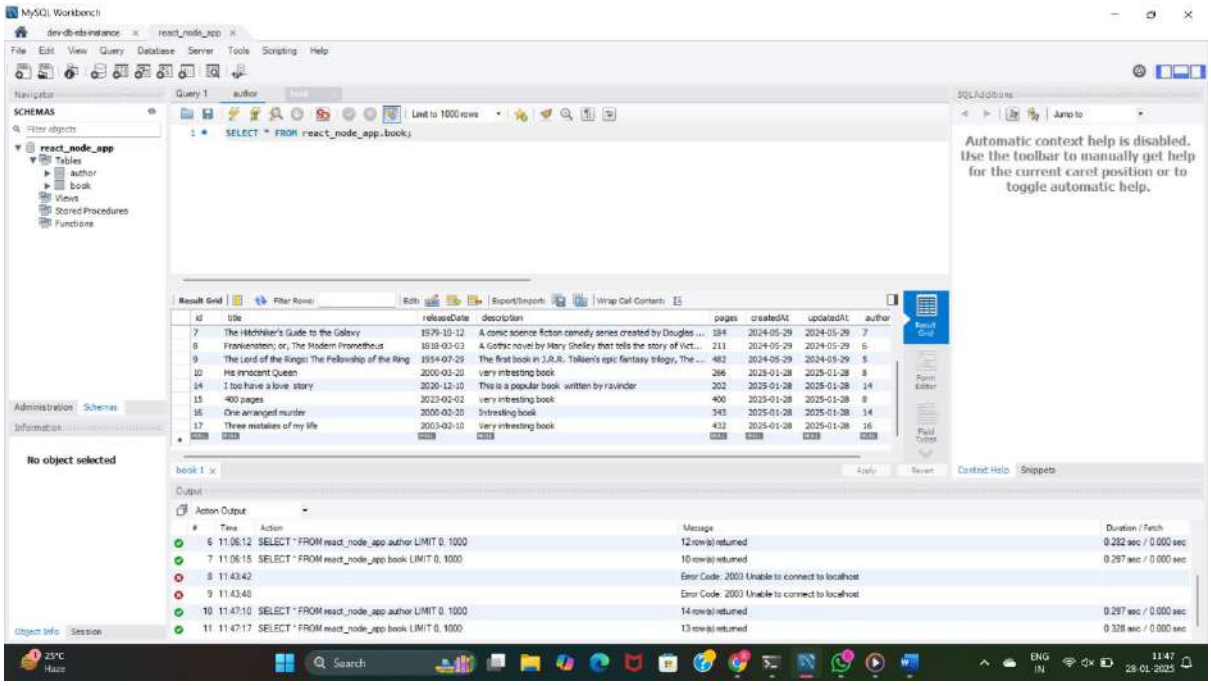
- The author name and his details are updated in mysql database.



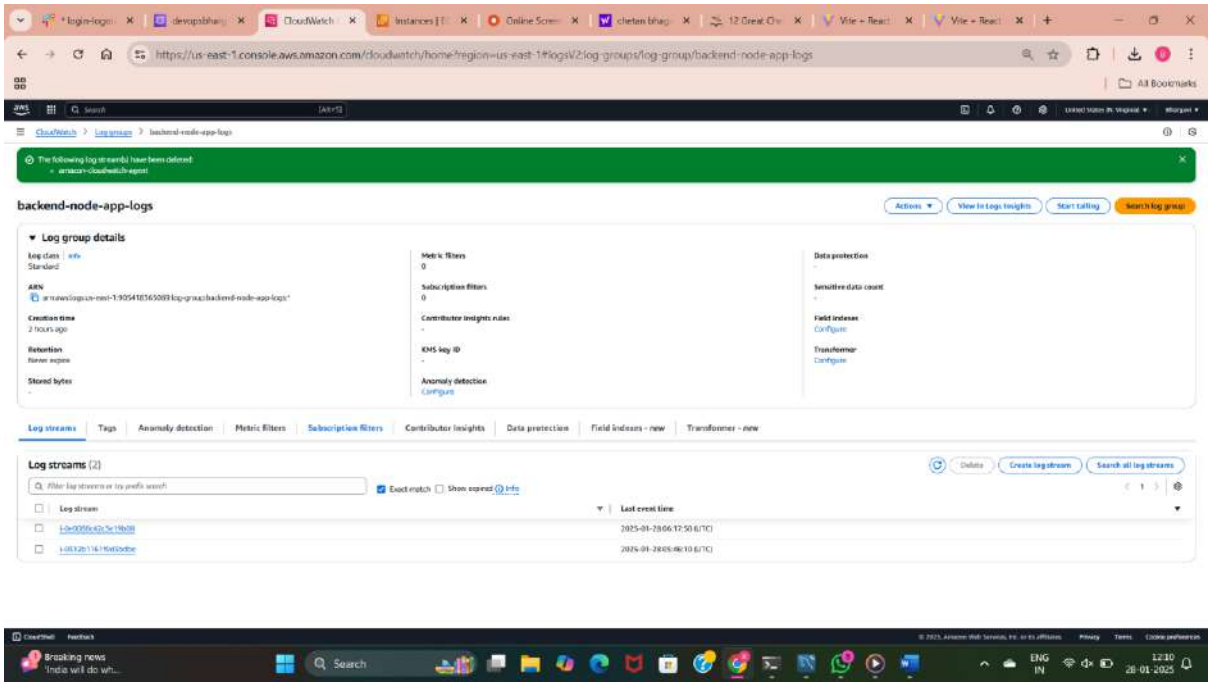
- We can also check the new book data in workbench of react_node_app database.



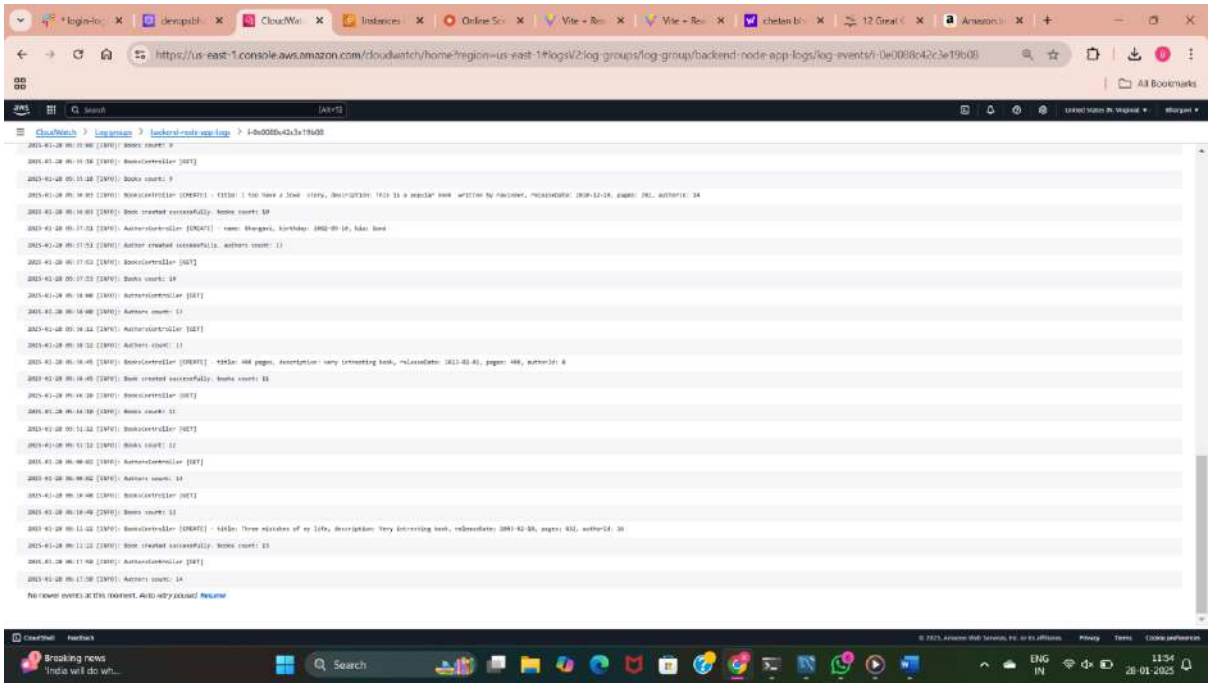
- New book is added in the website.



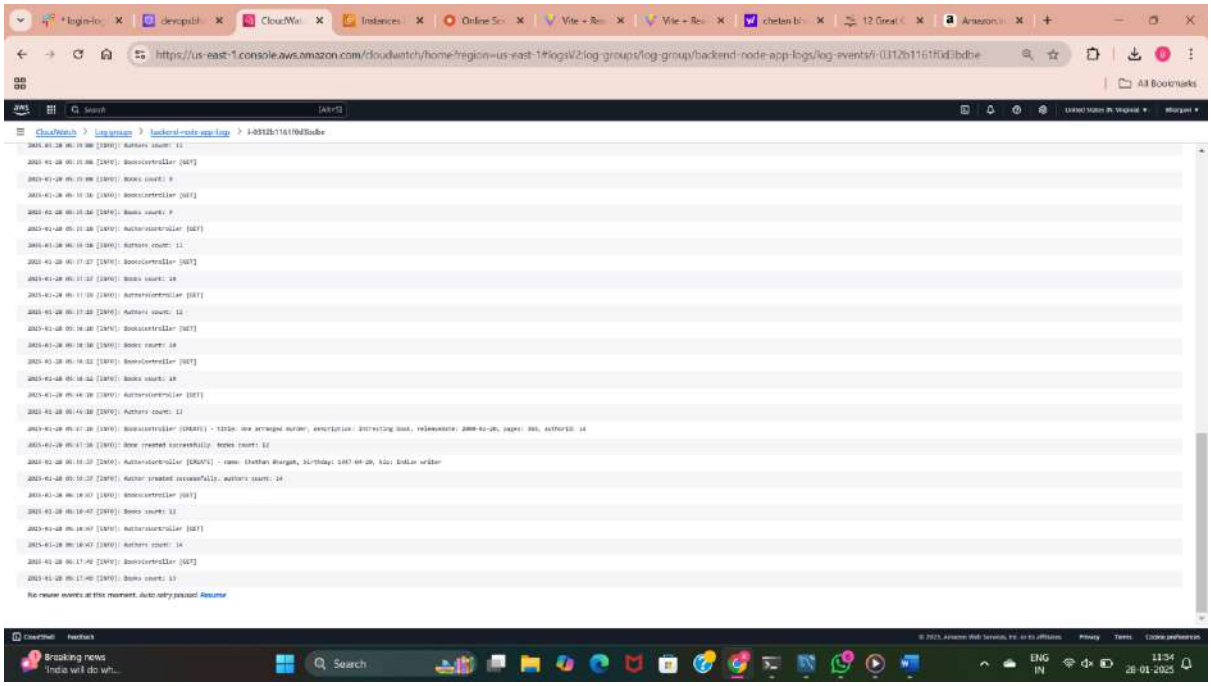
- New book details are updated in mysql database.



- Now go to cloudwatch logs and check for the updates. Here two instances are monitoring by cloudwatch.
- Select one instance and check the monitoring details.



- I am checking the database updating details here.
- Here the book creating details are monitored using cloudwatch.



- I have used other instance. Here the author details are monitored by cloudwatch.