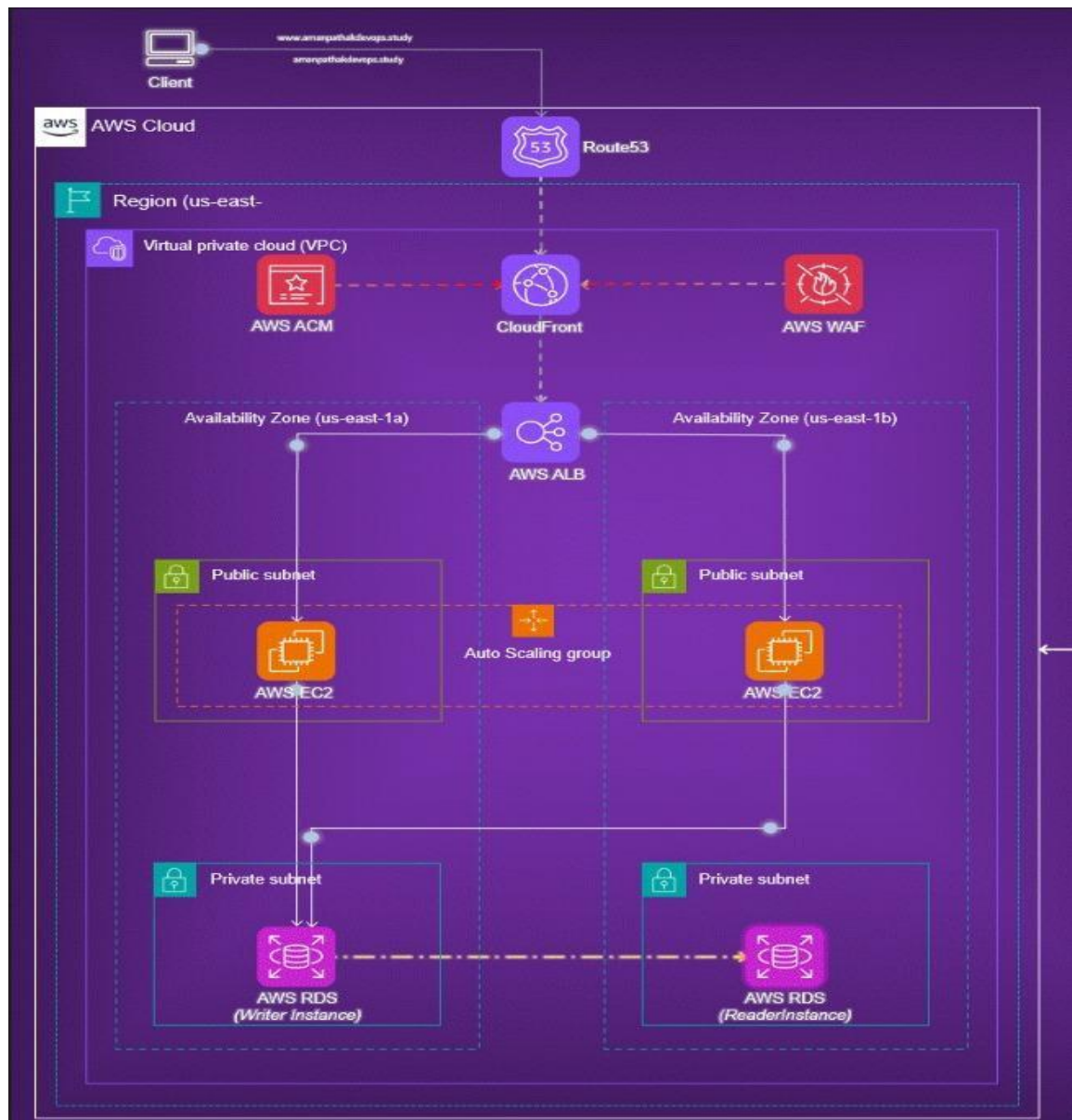


NAME: SAREDDI SUJITHA
BATCH: 138

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



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

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#VpcDetails:VpcId=vpc-011fd75cf739e233b

aws [Search] [Alt+S] United States (N. Virginia) Sujitha Reddy

VPC > Your VPCs > vpc-011fd75cf739e233b

VPC dashboard < EC2 Global View [?] Filter by VPC

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs

vpc-011fd75cf739e233b / my_vpc [Actions]

You have successfully modified the settings for vpc-011fd75cf739e233b / my_vpc.

Details Info

VPC ID vpc-011fd75cf739e233b	State Available	Block Public Access Off	DNS hostnames Enabled
DNS resolution Enabled	Tenancy default	DHCP option set dopt-005a61e4e757329b9	Main route table rtb-0c6588ba819fbd1f2
Main network ACL acl-0f5c2422c6e480daf	Default VPC No	IPv4 CIDR 10.0.0.0/16	IPv6 pool -
IPv6 CIDR (Network border group) -	Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 252177172204

Step 1: Create VPC and Networking Components

- Created a custom VPC in AWS with a CIDR block (e.g., 10.0.0.0/16).

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#subnets:

aws [Search] [Alt+S] United States (N. Virginia) Sujitha Reddy

VPC > Subnets

VPC dashboard < EC2 Global View [?] Filter by VPC

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists

You have successfully created 4 subnets: subnet-011161a4fd91f868b, subnet-08d03992aba33ba1b, subnet-04c510e0f15221226, subnet-05673e7f5ceec2eac. Last updated less than a minute ago. [Actions] [Create subnet]

Subnets (10) Info

Find subnets by attribute or tag

<input type="checkbox"/>	Name	Subnet ID	State	VPC
<input type="checkbox"/>	-	subnet-0cd9645da0777940a	Available	vpc-089b5746b75e32c13
<input type="checkbox"/>	-	subnet-02ebf9ac1cf638f22	Available	vpc-089b5746b75e32c13
<input type="checkbox"/>	public1	subnet-011161a4fd91f868b	Available	vpc-011fd75cf739e233b my_vpc
<input type="checkbox"/>	private2	subnet-05673e7f5ceec2eac	Available	vpc-011fd75cf739e233b my_vpc
<input type="checkbox"/>	public2	subnet-08d03992aba33ba1b	Available	vpc-011fd75cf739e233b my_vpc
<input type="checkbox"/>	private1	subnet-04c510e0f15221226	Available	vpc-011fd75cf739e233b my_vpc

Select a subnet

- Created **two public subnets** (one in each Availability Zone).
- Created **two private subnets** (also across different Availability Zones).
- Set up an **Internet Gateway** and attached it to the VPC.

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#InternetGateway:internetGatewayId=igw-04f44449635b659f4

Virtual private cloud

Internet gateways

igw-04f44449635b659f4

Internet gateway igw-04f44449635b659f4 successfully attached to vpc-011fd75cf739e233b

igw-04f44449635b659f4 / proj_igw

Details Info

Internet gateway ID igw-04f44449635b659f4	State Attached	VPC ID vpc-011fd75cf739e233b my_vpc	Owner 252177172204
--	-------------------	--	-----------------------

Tags

Search tags

Key	Value
Name	proj_igw

Manage tags

- Created a **Route Table** for the public subnets and associated it.

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-0009b5976e0e21d19

Virtual private cloud

Route tables

rtb-0009b5976e0e21d19

Updated routes for rtb-0009b5976e0e21d19 / pub_rt successfully

rtb-0009b5976e0e21d19 / pub_rt

Details Info

Route table ID rtb-0009b5976e0e21d19	Main No	Explicit subnet associations 2 subnets	Edge associations -
VPC vpc-011fd75cf739e233b my_vpc	Owner ID 252177172204		

Routes Subnet associations Edge associations Route propagation Tags

Routes (2)

Filter routes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-04f44449635b659f4	Active	No

- Enabled auto-assign public IPs for public subnets.
- Configured **NAT Gateway** in a public subnet to allow internet access for private subnets.

us-east-1.console.aws.amazon.com/vpconsole/home?region=us-east-1#NatGatewayDetails:natGatewayId=nat-06c98346b53b9bb4d

aws Search [Alt+S] United States (N. Virginia) Sujitha Reddy

VPC > NAT gateways > nat-06c98346b53b9bb4d

Filter by VPC

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways**
- Peering connections
- Route servers [New](#)

NAT gateway nat-06c98346b53b9bb4d | proj_natgw [Actions](#)

Details

NAT gateway ID nat-06c98346b53b9bb4d	Connectivity type Public	State Pending	State message Info -
NAT gateway ARN arn:aws:ec2:us-east-1:252177172204:natgateway/nat-06c98346b53b9bb4d	Primary public IPv4 address -	Primary private IPv4 address -	Primary network interface ID -
VPC vpc-011fd75cf739e233b / my_vpc	Subnet subnet-04c510e0f15221226 / private1	Created Friday, August 1, 2025 at 17:26:19 GMT+5:30	Deleted -

[Secondary IPv4 addresses](#) | [Monitoring](#) | [Tags](#)

- Created another **Route Table** for private subnets and set routing via the NAT Gateway.

us-east-1.console.aws.amazon.com/vpconsole/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-0cb4659fa468b74c4

aws Search [Alt+S] United States (N. Virginia) Sujitha Reddy

VPC > Route tables > rtb-0cb4659fa468b74c4

Filter by VPC

Virtual private cloud

- Your VPCs
- Subnets
- Route tables**
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections
- Route servers [New](#)

Updated routes for rtb-0cb4659fa468b74c4 / pvt_rt successfully

Details

Route table ID rtb-0cb4659fa468b74c4	Main No	Explicit subnet associations 2 subnets	Edge associations -
VPC vpc-011fd75cf739e233b / my_vpc	Owner ID 252177172204		

[Routes](#) | [Subnet associations](#) | [Edge associations](#) | [Route propagation](#) | [Tags](#)

Routes (2) [Both](#) [Edit routes](#)

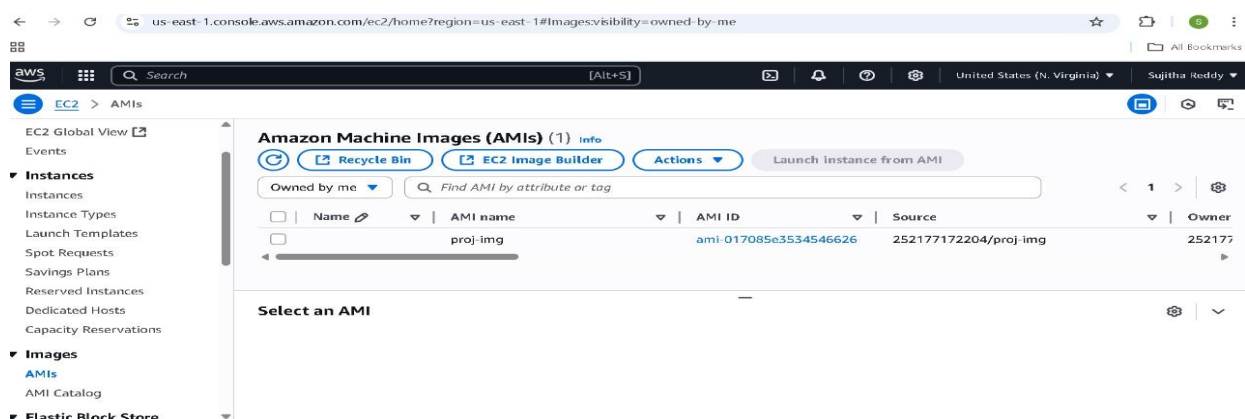
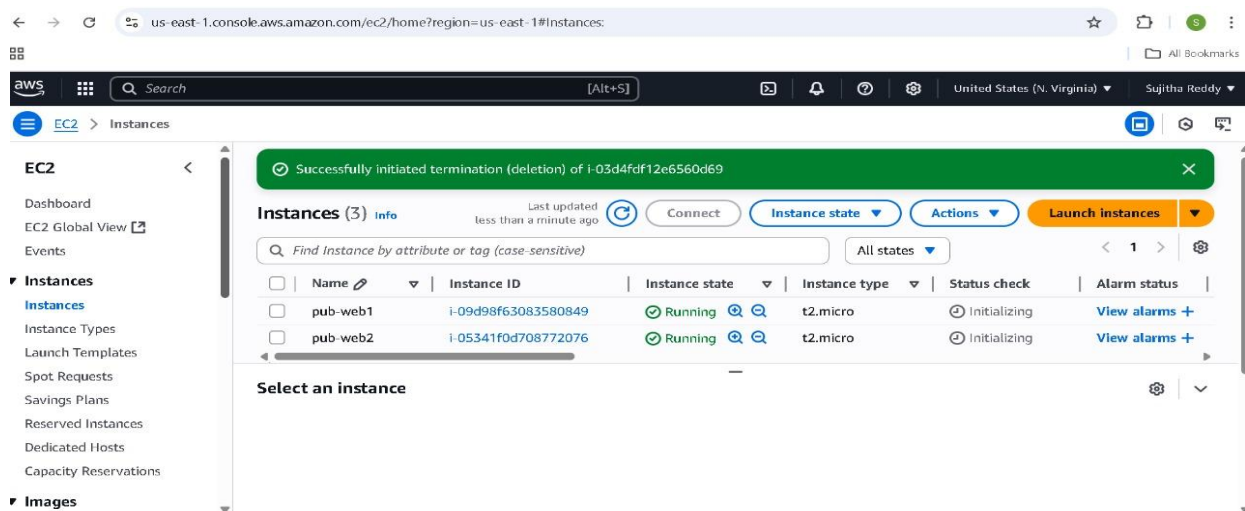
Destination	Target	Status	Propagated
0.0.0.0/0	nat-06c98346b53b9bb4d	Active	No
10.0.0.0/16	local	Active	No

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Verified VPC setup by successfully launching instances in public and private subnets.

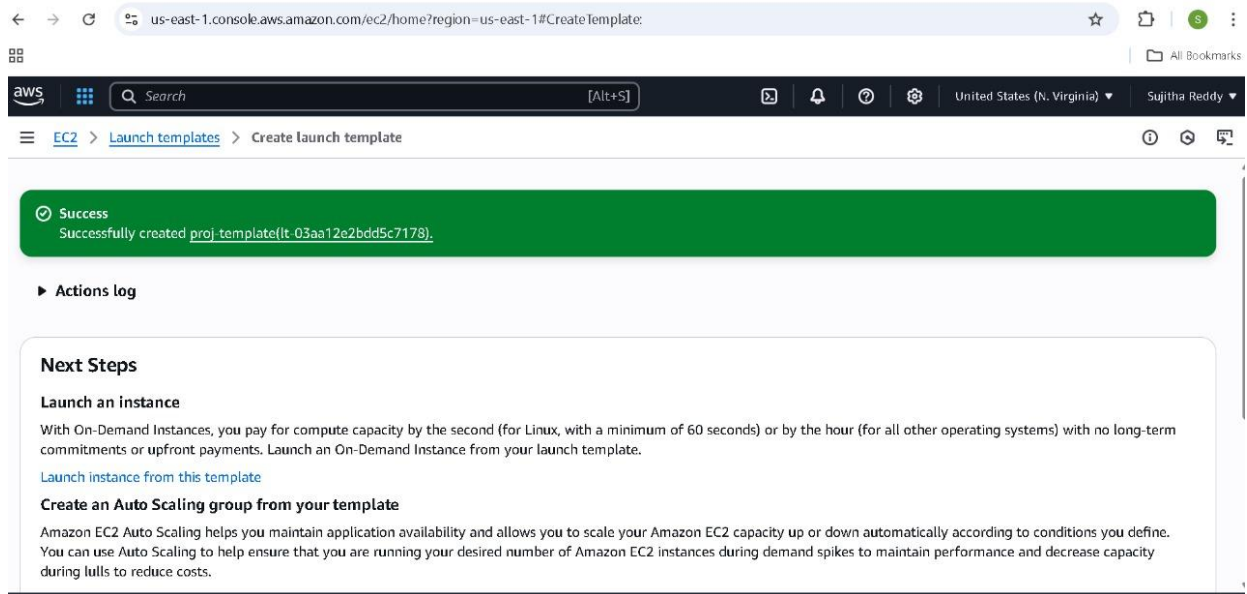
Step 2: Launch EC2 Instances

- Created two EC2 instances in AWS.
- Named them webserver-1 and webserver-2.
- Connected to each instance via SSH using the key pair.
- Installed the Apache web server with:
sudo -i, apt update -y and apt install apache2 commands
- Created a sample index.html file inside /var/www/html/.
- Verified setup by accessing the public IP in a browser.

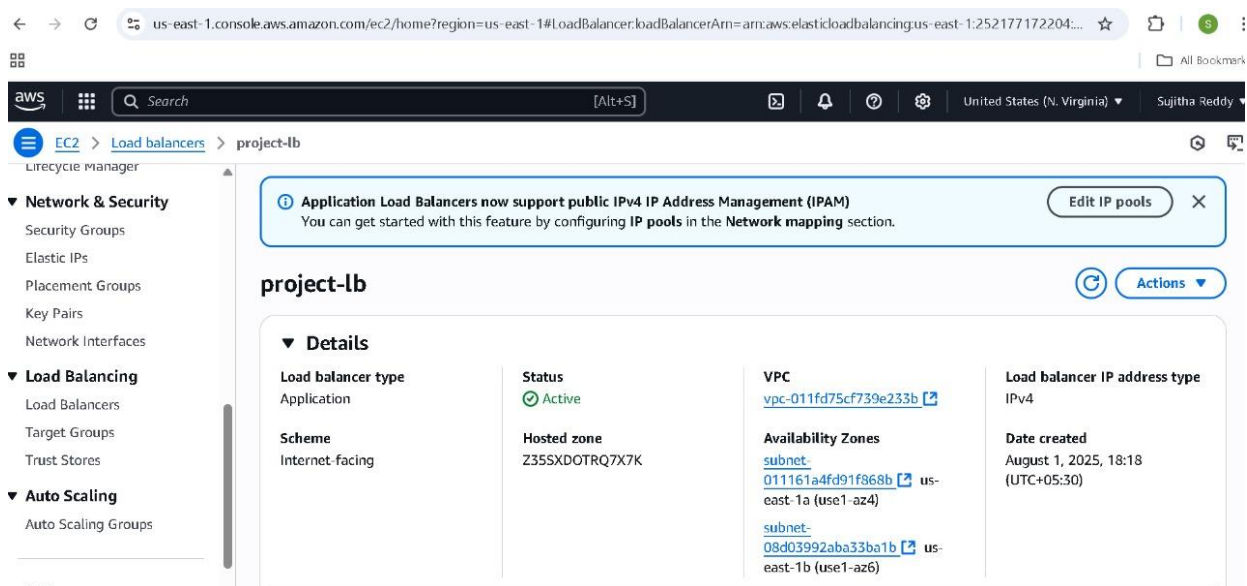


Step 3: Create Target Group and Load Balancer

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



Create AMI, and the Launch Template



Create DB Subnet Group

Click “Create DB Subnet Group”

Add Subnets

These must be private subnets (for security).

Create RDS database

Choose a database engine (e.g., **MySQL**)

DB Instance Settings:

DB Instance Identifier (e.g., mydb)

Master username (e.g., admin)

Master password and confirm password

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups:

aws Search [Alt+S] United States (N. Virginia) Sujitha Reddy

EC2 > Auto Scaling groups

Auto Scaling groups (1) Info Last updated less than a minute ago

Launch configurations Launch templates Actions Create Auto Scaling group

Search your Auto Scaling groups

<input type="checkbox"/>	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max
<input type="checkbox"/>	proj-asg	proj-template Version Default	2	-	2	1	5

us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#db-subnet-groups-list:

aws Search [Alt+S] United States (N. Virginia) Sujitha Reddy

Aurora and RDS > Subnet groups

Aurora and RDS

- Dashboard
- Databases
- Query editor
- Performance insights
- Snapshots
- Exports in Amazon S3
- Automated backups
- Reserved instances
- Proxies

Subnet groups

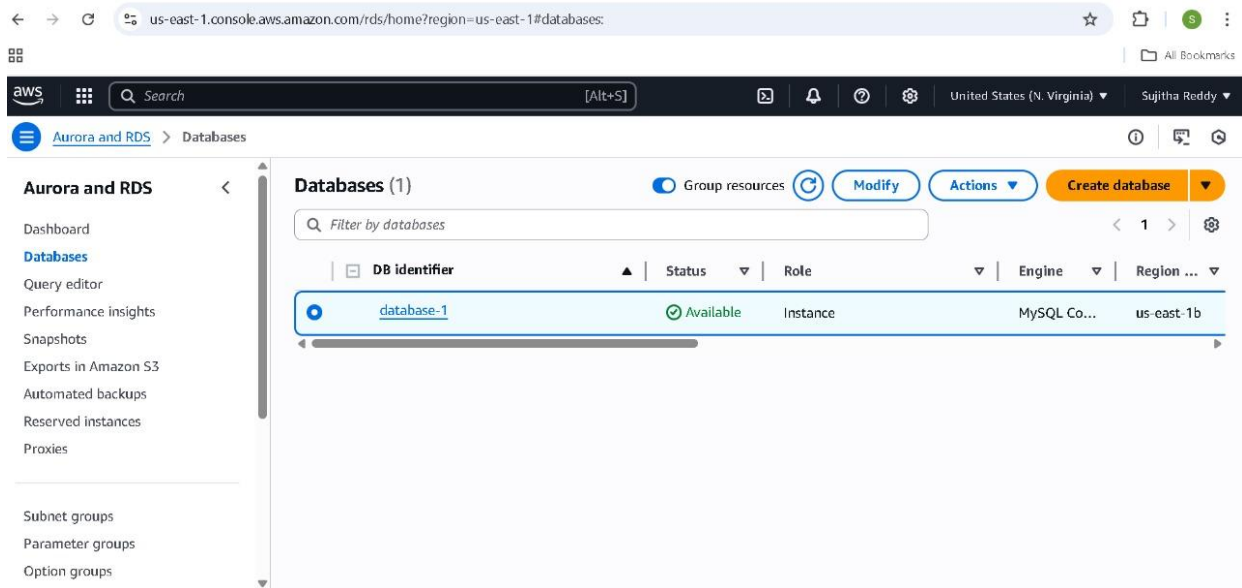
Parameter groups

Successfully created rds-subnetgrp. View subnet group

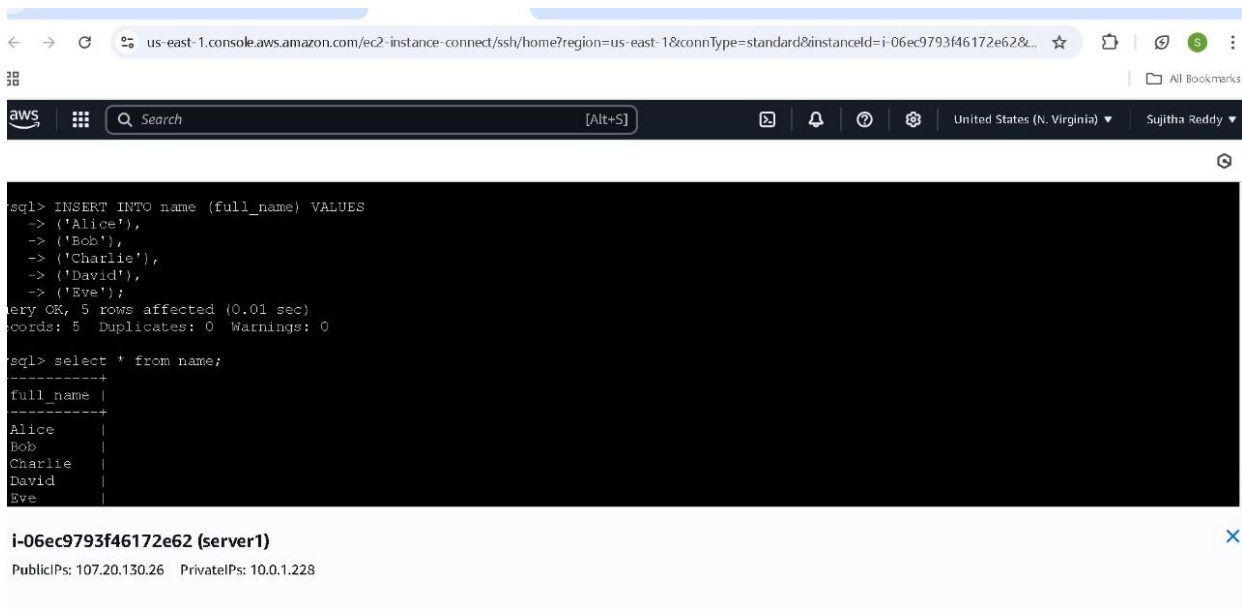
Subnet groups (1) Edit Delete Create DB subnet group

Filter by subnet group

<input type="checkbox"/>	Name	Description	Status	VPC
<input type="checkbox"/>	rds-subnetgrp	allow	Complete	vpc-011fd75cf739e233b

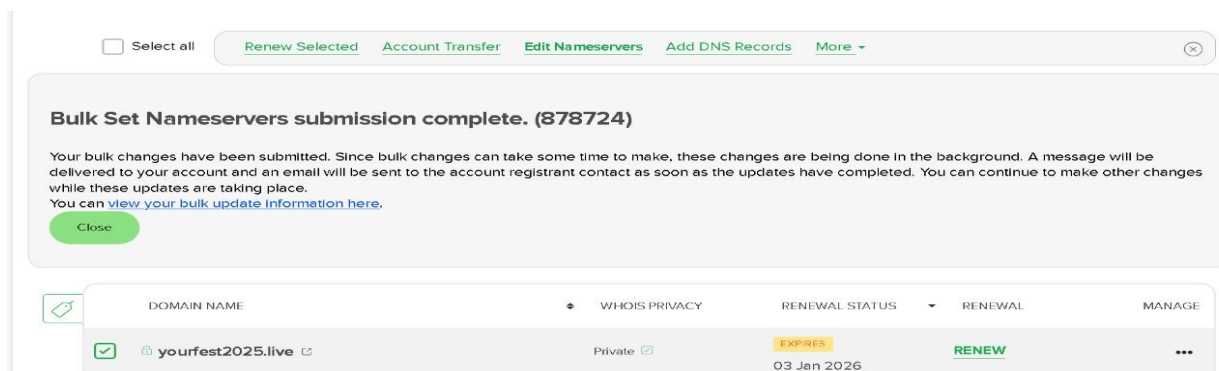
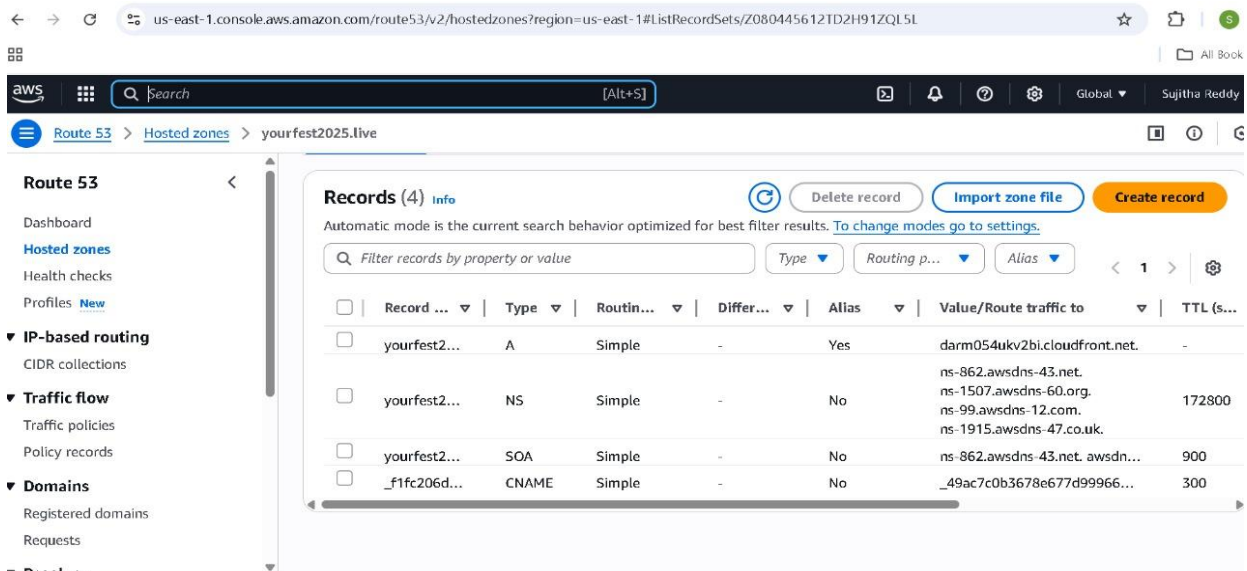
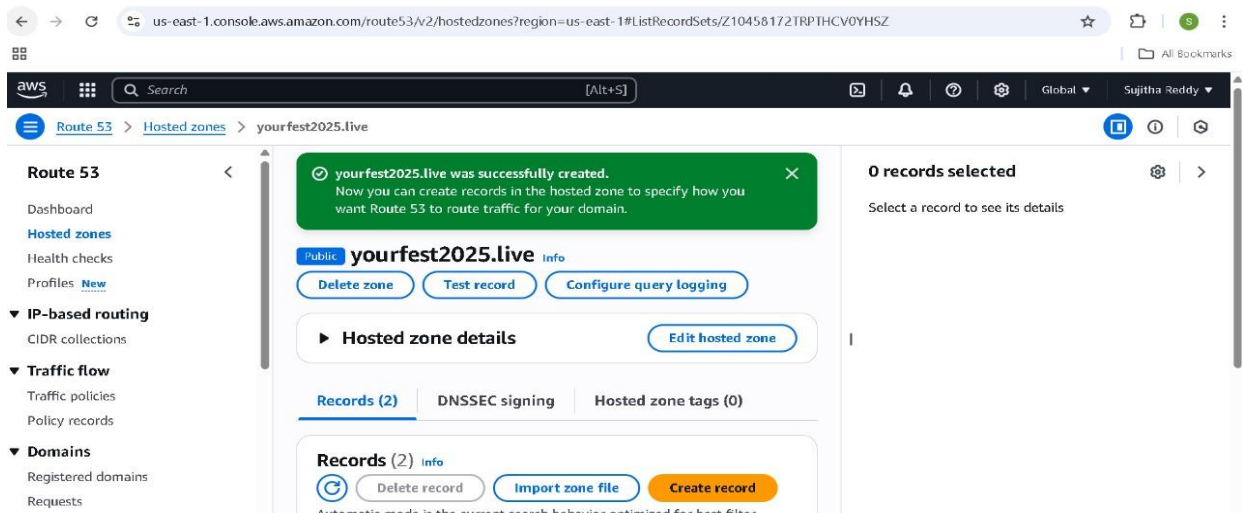


Install mysql-server in gitbash ---> systemctl start mysql.service
 mysql -h <endpoint> -u admin -p -----> Enter password
 Create database;
 Show databases;
 Use name;
 Create table; ---> Insert data;
 select * from name;



Step 4: Configure Route 53 and Domain Name

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



Step 5: Secure the Domain

- Initially, the domain used HTTP and was not secure.
- To make it secure:

Enable the **WAF (Web Application Firewall)**.

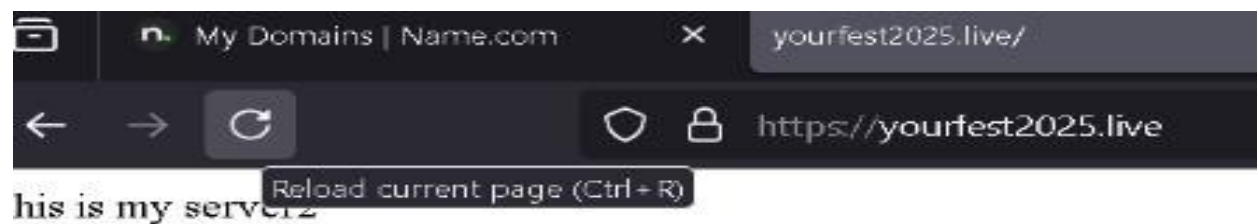
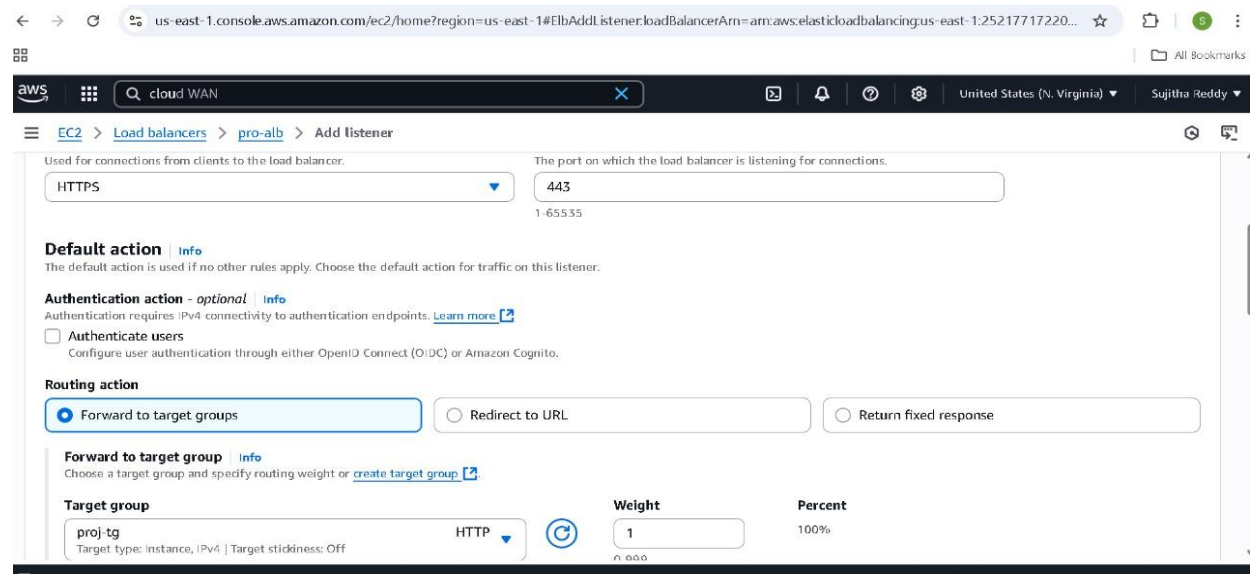
- Requested an SSL/TLS certificate in **AWS Certificate Manager (ACM)**.
- Set up a **CloudFront Distribution** using the certificate.

- Added an **HTTPS listener** to the Load Balancer.
- Verified the domain was accessible securely (HTTPS) using the CloudFront domain.

The screenshot shows the AWS CloudFront console interface. At the top, a green notification bar states "Successfully created new distribution." The breadcrumb navigation shows "CloudFront > Distributions > E3WW3RQ8CB632". The distribution name is "projectcf" with a "Standard" cache profile. A "View metrics" button is in the top right. Below the tabs (General, Security, Origins, Behaviors, Error pages, Invalidations, Tags, Logging), the "Details" section displays:

Details			
Name projectcf ✎	Distribution domain name dchse9dvvedoc.cloudfront.net	ARN arn:aws:cloudfront::2521771722:04:distribution/E3WW3RQ8CB632	Last modified ⌚ Deploying

The "Settings" section is partially visible below, with an "Edit" button in the top right. It includes fields for "Description", "Alternate domain names", and "Standard logging" (currently set to "Off").



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