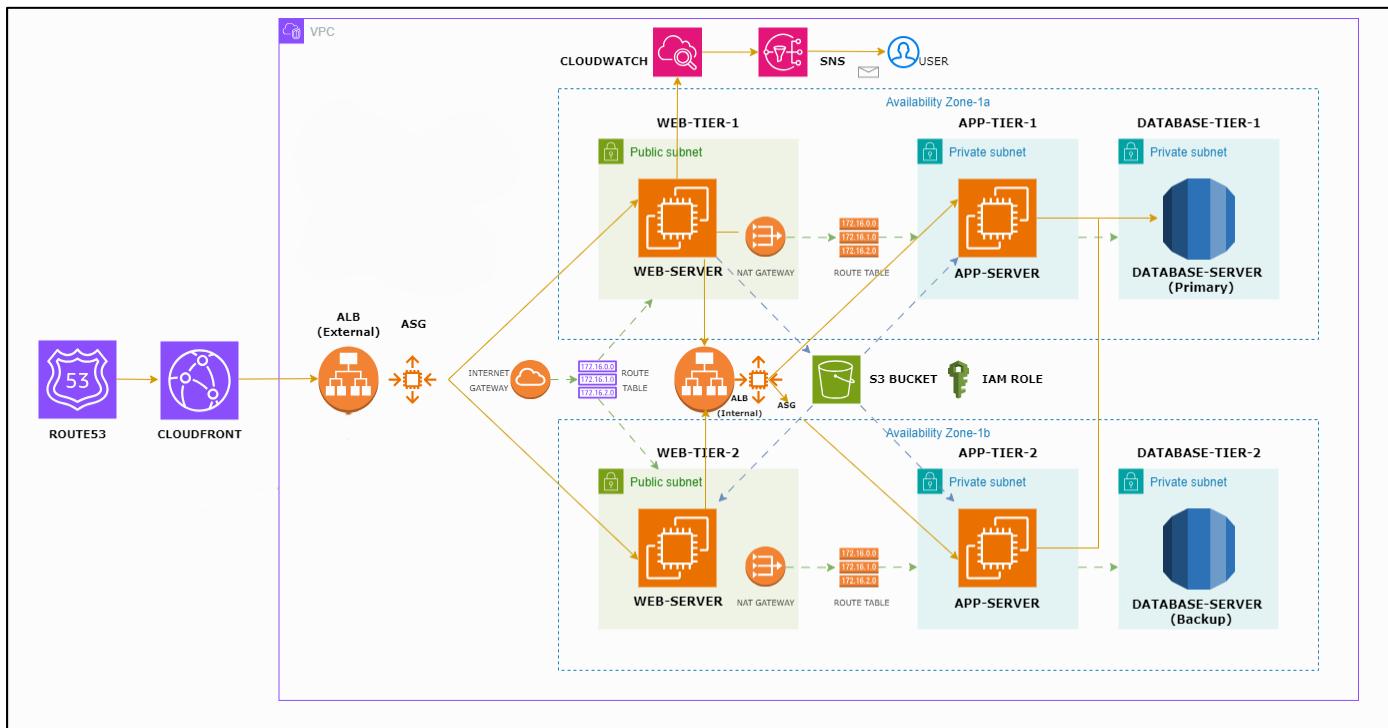


● INTRODUCTION:

Architecture Overview:



- Designed and deployed a **scalable, fault-tolerant web application** using AWS **3-Tier Architecture** across **multiple availability zones** to ensure high availability.
- Developed a **multi-layer cloud infrastructure**, separating **presentation, application, and database layers** for improved security, performance, and scalability.
- Configured **VPC, subnets, security groups, IAM (attached role to EC2 for application code access)**, and **WAF** to enforce secure access and protection. Deployed **EC2 instances with Auto Scaling Groups (ASG)** and distributed traffic using **Application Load Balancer (ALB)** for seamless scalability.
- Used **S3** to store source code and **RDS Aurora** for a highly available and scalable database solution. Set up **CloudWatch** for monitoring and integrated **SNS alerts** for proactive issue resolution.
- Optimized content delivery using **Route 53** for domain management and **CloudFront** for caching and acceleration, improving response times and user experience.
- Enhanced **availability, security, automation, and cost-efficiency**, ensuring seamless application performance across multiple AWS availability zones.

Tools & Technologies :

- VPC
- IAM
- EC2 (Installed Nginx, Node.js, and React),
- ASG, ALB,
- S3,
- Aurora DB
- CloudWatch
- SNS
- Route 53, CloudFront
- WAF

Part 1: Networking and Security

- Created an isolated network with the following components:
 - VPC
 - Subnets
 - Route Tables
 - Internet Gateway
 - NAT gateway
 - Security Groups

VPC

The screenshot shows the AWS VPC console for the 'us-east-1' region. The main page displays the details of a VPC named 'vpc-09c6fba5565ba4140'. Key information includes:

- VPC ID:** vpc-09c6fba5565ba4140
- State:** Available
- Tenancy:** default
- Main network ACL:** acl-0bc415d5ec49aafe4
- IPv6 CIDR:** -
- Default VPC:** No
- Network Address Usage metrics:** Disabled
- Block Public Access:** Off
- DNS option set:** dopt-02b371f6086bf7982
- IPv4 CIDR:** 10.0.0.0/16
- Route 53 Resolver DNS Firewall rule groups:** -
- DNS hostnames:** Disabled
- Main route table:** rtb-0160851c96f6e10c7
- IPv6 pool:** -
- Owner ID:** 253490795695

The 'Resource map' section shows the relationships between the VPC, Subnets (6), Route tables (1), and Network connections (0). The route table 'rtb-0160851c96f6e10c7' is connected to the VPC and the subnets.

Subnet

The screenshot shows the AWS VPC console for the 'us-east-1' region. The main page displays the details of a VPC named 'vpc-09c6fba5565ba4140'. Key information includes:

- VPC ID:** vpc-09c6fba5565ba4140
- State:** Available
- Tenancy:** default
- Main network ACL:** acl-0bc415d5ec49aafe4
- IPv6 CIDR:** -
- Default VPC:** No
- Network Address Usage metrics:** Disabled
- Block Public Access:** Off
- DNS option set:** dopt-02b371f6086bf7982
- IPv4 CIDR:** 10.0.0.0/16
- Route 53 Resolver DNS Firewall rule groups:** -
- DNS hostnames:** Disabled
- Main route table:** rtb-0160851c96f6e10c7
- IPv6 pool:** -
- Owner ID:** 253490795695

The 'Resource map' section shows the relationships between the VPC, Subnets (6), Route tables (1), and Network connections (0). The route table 'rtb-0160851c96f6e10c7' is connected to the VPC and the subnets. The subnets are grouped into two AZs: us-east-1a and us-east-1b.

Internet Gateway

The screenshot shows the AWS VPC Internet Gateways page. A success message at the top states: "Internet gateway igw-0f472da843dc7b835 successfully attached to vpc-09c6fba5565ba4140". The main card displays the Internet gateway ID (igw-0f472da843dc7b835), state (Attached), VPC ID (vpc-09c6fba5565ba4140 | project-vpc), and owner (253490795695). The "Tags" section shows a single tag named "Name" with the value "project-ig". The left sidebar lists various VPC components like EC2 Global View, Virtual private cloud, Internet gateways, Security, and PrivateLink and Lattice.

NAT Gateways

The screenshot shows the AWS VPC NAT Gateways page. A success message at the top states: "NAT gateway nat-02fd17d08a2387d95 | NAT-GW-AZ2 was created successfully.". The main card displays a table of four NAT gateways:

Name	NAT gateway ID	Connectivity...	State	State message	Primary public IP...	Primary private IP...	Primary network...
NAT-GW-AZ1	nat-03d1d70497080b82c	Public	Available	-	3.213.173.211	10.0.0.251	eni-0cf504f017b1
project-nat-1	nat-04a03ee6b63e3c9b5	Public	Deleted	Network vpc-09c6f...	-	10.0.0.209	eni-0709bad323c
NAT-GW-AZ2	nat-02fd17d08a2387d95	Public	Available	-	54.156.46.94	10.0.1.213	eni-04cba0318ee
project-nat-2	nat-019c3b0b869f36cca8	Public	Deleted	Network vpc-09c6f...	-	10.0.1.234	eni-06cd825f888

The left sidebar lists various VPC components like EC2 Global View, Virtual private cloud, Internet gateways, Security, and PrivateLink and Lattice.

Route Tables :

The screenshot shows the AWS VPC Route Tables page for a specific route table. A green success message at the top states: "Updated routes for rtb-07157a6dd42971240 / PublicRouteTable successfully". Below this, the title "rtb-07157a6dd42971240 / PublicRouteTable" is displayed. The "Details" tab is selected, showing the route table ID (rtb-07157a6dd42971240), VPC (vpc-09c6fba5565ba4140 | project-vpc), and owner ID (253490795695). The "Main" status is listed as "No". The "Explicit subnet associations" and "Edge associations" sections are currently empty. The "Routes" tab is active, showing two routes: one to the target igw-0f472da843dc7b835 with status Active and propagated No; another to the local subnet with status Active and propagated No. The left sidebar includes links for VPC dashboard, EC2 Global View, 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), Security (Network ACLs, Security groups), and PrivateLink and Lattice. The bottom right corner includes links for CloudShell, Feedback, Privacy, Terms, and Cookie preferences.

Edited Subnet Association and attached Route Table to both Public Subnet.

The screenshot shows the AWS VPC Route Tables page for the same route table. A green success message at the top states: "You have successfully updated subnet associations for rtb-07157a6dd42971240 / PublicRouteTable". Below this, the title "rtb-07157a6dd42971240 / PublicRouteTable" is displayed. The "Details" tab is selected, showing the route table ID (rtb-07157a6dd42971240), VPC (vpc-09c6fba5565ba4140 | project-vpc), and owner ID (253490795695). The "Main" status is listed as "No". The "Explicit subnet associations" section now shows "2 subnets". The "Edge associations" section is still empty. The "Subnet associations" tab is active, showing two explicit subnet associations: Public-Web-Subnet-AZ-1 (subnet-05f41ef6206ff8ada, 10.0.0.0/24) and Public-Web-Subnet-AZ-2 (subnet-0a0572e98abb0dc8c, 10.0.1.0/24). The "Routes" tab is also visible. The left sidebar and bottom right corner are identical to the first screenshot.

Created and Attached Route Tables to Both Private Subnets

The screenshot shows the AWS VPC console with the URL us-east-1.console.aws.amazon.com/vpc/home?region=us-east-1#RouteTableDetails:RouteTableId=rtb-0e1264aac526438d1. The main content area displays the details of the route table **rtb-0e1264aac526438d1 / PrivateRouteTable-AZ1**. A green success message at the top states: "You have successfully updated subnet associations for rtb-0e1264aac526438d1 / PrivateRouteTable-AZ1". The "Routes" tab is selected, showing two routes:

Destination	Target	Status	Propagated
0.0.0.0/0	nat-03d1d70497080b82c	Active	No
10.0.0.0/16	local	Active	No

The left sidebar includes sections for VPC dashboard, EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables), Security (Network ACLs, Security groups), and PrivateLink and Lattice.

This screenshot shows the same AWS VPC console interface, but the "Subnet associations" tab is now selected. It displays the explicit subnet association for the route table:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Private-Subnet-AZ1	subnet-0fb59be9043b75f4f	10.0.2.0/24	-

Below this, a section titled "Subnets without explicit associations (3)" lists three subnets that are implicitly associated with the main route table:

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR
Private-Subnet-AZ2	subnet-0ab528404dfcc2799	10.0.3.0/24	-
Private-Subnet-AZ1	subnet-199-20-55-323299	10.0.1.0/24	-

The left sidebar remains the same as the first screenshot.

All Route Tables

The screenshot shows the AWS VPC Console with the 'Route tables' section selected. The left sidebar includes links for VPC dashboard, EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables), Security (Network ACLs, Security groups), and PrivateLink and Lattice. The main content area displays a table of route tables with columns for Name, Route table ID, Explicit subnet associations, Edge associations, Main, VPC, and Owner ID. Three route tables are listed: PrivateRouteTableAZ-2, PublicRouteTable, and PrivateRouteTable-AZ1. A message at the bottom indicates 'Route tables: rtb-04851188b38c3db08, rtb-07157a6dd42971240, rtb-0e1264aac526438d1'.

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC	Owner ID
-	rtb-0160851c96f6e10c7	-	-	Yes	vpc-09c6fba5565ba4140 proj...	253490795695
<input checked="" type="checkbox"/> PrivateRouteTableAZ-2	rtb-04851188b38c3db08	subnet-0ab328404dfcc2...	-	No	vpc-09c6fba5565ba4140 proj...	253490795695
<input checked="" type="checkbox"/> PublicRouteTable	rtb-07157a6dd42971240	2 subnets	-	No	vpc-09c6fba5565ba4140 proj...	253490795695
<input checked="" type="checkbox"/> PrivateRouteTable-AZ1	rtb-0e1264aac526438d1	subnet-0fb59be9043b75...	-	No	vpc-09c6fba5565ba4140 proj...	253490795695
-	rtb-0aca6e2bac9fc7f01	-	-	Yes	vpc-0610fd9fedfd36a33	253490795695

Security Groups

The screenshot shows the AWS VPC Console with the 'Security groups' section selected. The left sidebar includes links for VPC dashboard, EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables), Security (Network ACLs, Security groups), and PrivateLink and Lattice. The main content area shows a success message: 'Security group (sg-0f315cb52c58d62d1 | Internet-facing-sg) was created successfully'. Below it is a detailed view of the security group 'sg-0f315cb52c58d62d1 - Internet-facing-sg'. The 'Details' section shows the security group name (Internet-facing-sg), security group ID (sg-0f315cb52c58d62d1), owner (253490795695), description (External LB SG), and VPC ID (vpc-09c6fba5565ba4140). The 'Inbound rules' tab is selected, displaying two rules: one for port 80 (HTTP) and one for port 443 (HTTPS). Other tabs include Outbound rules, Sharing - new, VPC associations - new, and Tags.

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
sgr-04ae6c33120562b8b	IPv4	HTTP	TCP	80	0.0.0.0/0	
sgr-09b4c9160019b98c8	IPv4	HTTPS	TCP	443	0.0.0.0/0	

All Security Groups Created And Added Inbound Rules According to need

The screenshot shows the AWS VPC console interface. A success message at the top indicates that a security group named 'sg-0d13dd845c0e1c1cd | DB-sg' was created successfully. Below this, a table lists five security groups:

Name	Security group ID	Security group name	VPC ID	Description
sg-061186f7aab7a4ec45	Web-sg	vpc-09c6fba5565ba4140	web tier sg	
sg-0e6099b2c34421ece	default	vpc-0d10fd9fedfd36a33	default VPC security group	
sg-0f315cb52c58d62d1	Internet-facing-sg	vpc-09c6fba5565ba4140	External LB SG	
sg-06f49a811ebb397a6	App-tier-sg	vpc-09c6fba5565ba4140	App-tier-sg	
sg-02417d52c128488e2	default	vpc-09c6fba5565ba4140	default VPC security group	
sg-0d79450b84283ecd1	Internal-lb-sg	vpc-09c6fba5565ba4140	Internal facing lb sg	
sg-0d13dd845c0e1c1cd	DB-sg	vpc-09c6fba5565ba4140	DB-sg	

At the bottom of the page, a summary of the security groups is provided:

Security Groups: sg-061186f7aab7a4ec45, sg-0f315cb52c58d62d1, sg-06f49a811ebb397a6, sg-0d79450b84283ecd1, sg-0d13dd845c0e1c1cd

Part 2: Database Deployment

- Deploy Database Layer
 - Subnet Groups
 - Multi-AZ Database

Subnet Groups Created

The screenshot shows the AWS Aurora and RDS console. A success message at the top indicates that a subnet group named 'three-tier-sb-subnet-group' was successfully created. Below this, the details of the subnet group are displayed:

Subnet group details

- VPC ID: vpc-09c6fba5565ba4140
- ARN: arn:aws:rds:us-east-1:253490795695:subgrp:three-tier-sb-subnet-group
- Supported network types: IPv4
- Description: three-tier-sb-subnet-group

Subnets (2)

Availability zone	Subnet name	Subnet ID	CIDR block
us-east-1a	Private-DB-Subnet-AZ1	subnet-09c28c5fac3722904	10.0.4.0/24
us-east-1b	Private-DB-Subnet-AZ2	subnet-061b1f0258ef5a6ed	10.0.5.0/24

Tags (0)

MySQL-Compatible Amazon Aurora database Created with Reader Node in Different AZ

The screenshot shows the AWS RDS console with the 'Databases' page open. The left sidebar includes links for Dashboard, Databases, Query editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, Zero-ETL integrations, Events, Event subscriptions, and Recommendations. The main area displays a table titled 'Databases (3)' with columns for DB Identifier, Status, Role, Engine, Region, Size, and Recommendations. The table shows:

DB Identifier	Status	Role	Engine	Region	Size	Recommendations
database-1	Available	Regional c...	Aurora My...	us-east-1	2 instances	-
database-1-instance-1	Available	Writer ins...	Aurora My...	us-east-1b	db.r7g.large	28.68%
database-1-instance-1-us-east-1a	Available	Reader ins...	Aurora My...	us-east-1a	db.r7g.large	-

Part 3: App Tier Instance Deployment

Uploaded Source code of web Application to S3 so, our instances can access it and also created an AWS IAM EC2 role with AWS S3 **AmazonSSMManagedInstanceCore,AmazonS3ReadOnlyAccess** permission to connect to our instances securely and without needing to created SSH key pairs and access source code from bucket.

S3 Bucket

The screenshot shows the AWS S3 console with the 'Buckets' page open. The left sidebar includes links for General purpose buckets, Directory buckets, Table buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Block Public Access settings for this account, Storage Lens, Dashboards, Storage Lens groups, AWS Organizations settings, Feature spotlight (11), and AWS Marketplace for S3. The main area displays a table titled 'General purpose buckets (1)' with columns for Name, AWS Region, IAM Access Analyzer, and Creation date. The table shows:

Name	AWS Region	IAM Access Analyzer	Creation date
demowebapp-ud-101	US East (N. Virginia) us-east-1	View analyzer for us-east-1	May 14, 2025, 13:54:39 (UTC+05:30)

IAM Role

The screenshot shows the AWS IAM Roles details page for the role 'demo-ec2role'. The left sidebar includes sections for Identity and Access Management (IAM), Access management, and Access reports. The main content area displays the 'Summary' tab, which includes the ARN (arn:aws:iam::253490795695:role/demo-ec2role) and maximum session duration (1 hour). The 'Permissions' tab is selected, showing two managed policies: 'AmazonS3ReadOnlyAccess' and 'AmazonSSMManagedInstanceCore', both of which are AWS managed policies.

App Instance Deployment

The screenshot shows the AWS EC2 Instances page. The left sidebar lists EC2 services like Dashboard, Global View, Events, Instances, Images, Elastic Block Store, and Network & Security. The main content area shows a table of instances, with one instance named 'myWebAppServer1' highlighted. This instance is in the 'Running' state, has a Public IPv4 address of 10.0.2.56, and a Private IPv4 address of 10.0.2.56. The instance type is t2.micro, and it is located in the us-east-1a availability zone.

Configure Database

Session ID: root-p6a54bca88zpnssl24ojzvyori Instance ID: i-0a7ea90d9f08f66b9
Terminate

```
Installing      : mysql-community-client-5.7.44-1.el7.x86_64          4/4
Running scriptlet: mysql-community-client-5.7.44-1.el7.x86_64        4/4
Verifying       : ncurses-compat-libs-6.2-4.20200222.amzn2023.0.6.x86_64 1/4
Verifying       : mysql-community-client-5.7.44-1.el7.x86_64        2/4
Verifying       : mysql-community-common-5.7.44-1.el7.x86_64        3/4
Verifying       : mysql-community-libs-5.7.44-1.el7.x86_64         4/4

Installed:
mysql-community-client-5.7.44-1.el7.x86_64  mysql-community-common-5.7.44-1.el7.x86_64  mysql-community-libs-5.7.44-1.el7.x86_64  ncurses-compat-libs-6.2-4.20200222.amzn2023.0.6.x86_64

Complete!
[ec2-user@ip-10-0-2-56 ~]$ mysql
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/lib/mysql/mysql.sock' (2)
[ec2-user@ip-10-0-2-56 ~]$ mysql -h database-1-instance-1.c6js0ucg8w21.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 191
Server version: 8.0.39 8bc99e28

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create DATABASE mywebappdb;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| mywebappdb |
| performance_schema |
+-----+
```

Configure AppInstance

Uploaded AppTier Back-End Source code to S3 Bucket

Upload objects - S3 Systems Manager | Aurora and RDS | AWS Three Tier Web | uddhavgun/uddhav | (18059) Project2 - Dev | MySQL 5.7 Download
All Bookmarks

aws Search [Alt+S] Close

Amazon S3

General purpose buckets

- Directory buckets
- Table buckets
- Access Grants
- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- IAM Access Analyzer for S3

Block Public Access settings for this account

Storage Lens

- Dashboards
- Storage Lens groups
- AWS Organizations settings

Feature spotlight [11]

AWS Marketplace for S3

CloudShell Feedback

Upload: status

After you navigate away from this page, the following information is no longer available.

Summary

Destination	Succeeded	Failed
s3://demowebapp-ud-101	6 files, 47.4 KB (100.00%)	0 files, 0 B (0%)

Files and folders (6 total, 47.4 KB)

Name	Folder	Type	Size	Status
DbConfig.js	app-tier/	text/javascript	193.0 B	Succeeded
index.js	app-tier/	text/javascript	3.1 KB	Succeeded
package-lock.json	app-tier/	application/json	41.8 KB	Succeeded
package.json	app-tier/	application/json	655.0 B	Succeeded
README.md	app-tier/	-	12.0 B	Succeeded
TransactionService.js	app-tier/	text/javascript	1.7 KB	Succeeded

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

Connected to EC2 Instance to SSM and configured AppInstance and tested DB connectivity Command History :

```
Session ID: root-p6a54bca88zpnssl24ojzvyori Instance ID: i-0a7ea90d9f08f66b9
1 ping 8.8.8.8
2 clear
3 sudo wget https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
4 sudo yum install mysql57-community-release-el7-11.noarch.rpm
5 sudo yum install mysql57-community-release-el7-11.noarch.rpm
6 sudo rpm --import https://repo.mysql.com/RPM-GPG-KEY-mysql-2022
7 sudo yum install mysql57-community-release-el7-11.noarch.rpm
8 sudo yum install https://dev.mysql.com/get/mysql57-community-release-el7-11.noarch.rpm
9 mysql -h database-1-instance-1.c6js0ucg8w21.us-east-1.rds.amazonaws.com -u admin -p
10 mysql -h database-1-instance-1.c6js0ucg8w21.us-east-1.rds.amazonaws.com -u admin -p
11 mysql
12 sudo yum update -y
13 sudo yum install -y mysql
14 mysql
15 mysql -h database-1-instance-1.c6js0ucg8w21.us-east-1.rds.amazonaws.com -u admin -p
16 clear
17 curl -o https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
18 source ~/.bashrc
19 source ~/.bashrc
20 nvm install 16
21 nvm use 16
22 nvm install -g pm2
23 cd
24 ls -rlt
25 aws s3 cp s3://demowebapp-ud-101/app-tier/ app-tier --recursive
26 cd ~/app-tier
27 npm install
28 pm2 start index.js
29 pm2 list
30 pm2 logs
31 pm2 startup
32 sudo env PATH=$PATH:/home/ec2-user/.nvm/versions/node/v16.20.2/bin /home/ec2-user/.nvm/versions/node/v16.20.2/lib/node_modules/pm2/bin/pm2 startup systemd -u ec2-user --hp /home/ec2-user@ip-10-0-2-56 app-tier]$
```

Part 4: Internal Load Balancing and Auto Scaling

Created an AMI of App Tier using previously created AppTier EC2 Instance Image

EC2

Amazon Machine Images (AMIs) (1/1)

Name	AMI ID	Source	Owner	Visibility	Status
AppTierImage	ami-088705603baaca4fc	253490795695/AppTierImage	253490795695	Private	Pending

AMI ID: ami-088705603baaca4fc

Details	Permissions	Storage	Tags
AMI ID ami-088705603baaca4fc	Image type machine	Platform details Linux/UNIX	Root device type EBS
AMI name AppTierImage	Owner account ID 253490795695	Architecture x86_64	Usage operation RunInstances
Root device name /dev/xvda	Status Pending	Source 253490795695/AppTierImage	Virtualization type hvm
Boot mode uefi-preferred	State reason -	Creation date 2025-05-14T10:49:21.000Z	Kernel ID -
Description App Tier	Product codes -	RAM disk ID -	Deprecation time -
Last launched time	Block devices	Deregistration protection	Allowed image

AppTier Target Group Created

The screenshot shows the AWS EC2 Target groups console. On the left, a navigation sidebar lists various services like AMIs, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and more. The main area displays a table titled "Target groups (1/1) Info" with one entry: "AppTierTG". The table columns include Name, ARN, Port, Protocol, Target type, Load balancer, and VPC ID. Below the table, a detailed view for "Target group: AppTierTG" is shown under the "Details" tab. It provides information such as Target type (Instance), Protocol (HTTP: 4000), IP address type (IPv4), and Load balancer (None associated). It also shows metrics for Total targets (0), Healthy (0), Unhealthy (0), Unused (0), Initial (0), and Draining (0). The bottom of the page includes standard AWS footer links.

AppTier Internal Load Balancer Created

Created Application Load Balancer and added HTTP Listener to it with AppTier Target Group

The screenshot shows the AWS EC2 Load balancers console. The left sidebar lists services like Reserved Instances, Images, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main area displays a detailed view for "app-tier-internal-lb". It shows the load balancer type (Application), status (Provisioning), scheme (Internal), and hosted zone (Z355XDTRQ7XK). It also shows the VPC (vpc-09c6fba5565ba4140), availability zones (subnet-0ab328404dfcc2799, subnet-0fb59be9043b75f4f), and DNS name (internal-app-tier-internal-lb-1636082849.us-east-1.elb.amazonaws.com). Below this, the "Listeners and rules" tab is selected, showing a single listener for "Protocol:Port" (HTTP:80) which is "Forward to target group" (AppTierTG, 1 rule). The bottom of the page includes standard AWS footer links.

Launch Template created using AppTier AMI which we created previously stage

The screenshot shows the AWS EC2 Launch Templates page. On the left, a sidebar navigation includes EC2, Instances, Images, Elastic Block Store, and Network & Security. The main content area displays the details of the 'AppTier-LaunchTemplate (lt-0b4ebffe891837f51)'. It shows the launch template ID (lt-0b4ebffe891837f51), name (AppTier-LaunchTemplate), default version (1), and owner (arn:aws:iam::253490795695:root). Below this, the 'Launch template version details' section shows a single version (1 Default) with the following configuration: AMI ID (ami-088705603baaca4fc), Instance type (t2.micro), Availability Zone (-), and Security group (sg-06f49a811ebb597a6). The date created is 2025-05-14T11:39:02.000Z and it was created by arn:aws:iam::253490795695:root.

Auto Scaling Group Created.

Attached AppTier Launch Template ,also attached internal load balancer to it. Configured Group Size upto 2.

The screenshot shows the AWS Auto Scaling groups page. The top navigation bar includes EC2 and Auto Scaling groups. The main content area displays the 'Auto Scaling groups (1/1) Info' section. It lists one group named 'ApptierASG' with the following details: Launch template (AppTier-LaunchTemplate), Version (Default), Desired capacity (2), Min (2), Max (2), and Availability Zones (us-east-1a, us-east-1b). The last update was less than a minute ago. Below this, the 'Auto Scaling group: ApptierASG' section provides an overview of the group's capacity, including desired capacity (2), scaling limits (Min - Max: 2 - 2), and status (-). It also shows the date created (Wed May 14 2025 17:15:06 GMT+0530 (India Standard Time)). The bottom navigation bar includes CloudShell and Feedback.

Part 5: Web Tier Instance Deployment

Updated NGINX Configuration File. open up the **application-code/nginx.conf** file from the source code. Scroll down to **line 58** and replaced [INTERNAL-LOADBALANCER-DNS] with our internal load balancer's DNS entry.

DNS name Copied!

The screenshot shows the AWS CloudWatch Metrics console with the URL <https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancers>. The left sidebar navigation includes AMIs, AMI Catalog, Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), Network & Security (Security Groups, Elastic IPs, Placement Groups, Key Pairs, Network Interfaces), Load Balancing (Load Balancers, Target Groups, Trust Stores), and Auto Scaling (Auto Scaling Groups). The main content area displays a table titled "Load balancers (1)". The table has columns: Name, State, VPC ID, Availability Zones, Type, and Date created. One row is listed: "app-tier-internal-lb..." (State: Active, VPC ID: vpc-09c6fba5565ba4140, Availability Zones: 2, Type: application, Date created: May 14, 2025, 16:56 (UTC+05:30)). A tooltip "DNS name copied" is shown over the "Name" column for this row. The bottom of the page includes CloudShell, Feedback, and standard footer links.

nginx.conf file

The screenshot shows a code editor displaying the **nginx.conf** file. The file path is C:\Users\Lenovo\Downloads>aws-three-tier-web-architecture-workshop-main>aws-three-tier-web-architecture-workshop-main>application-code>nginx.conf. The code editor interface includes tabs for DbConfig.js and index.js, and a search bar. The code itself is a standard NGINX configuration file. A red box highlights the line at line 58: `proxy_pass http://[INTERNAL-LOADBALANCER-DNS]:80/;`. The code editor also shows line numbers and various icons for file operations.

```
17 http {
38     server {
40         listen      [::]:80;
41         server_name _;

43         #health check
44         location /health {
45             default_type text/html;
46             return 200 "<!DOCTYPE html><p>Web Tier Health Check</p>\n";
47         }

49         #react app and front end files
50         location / {
51             root    /home/ec2-user/web-tier/build;
52             index index.html;
53             try_files $uri /index.html;
54         }

56         #proxy for internal lb
57         location /api/ {
58             proxy_pass http://[INTERNAL-LOADBALANCER-DNS]:80/;
59         }
60     }
61 }
62 }

64     # Settings for a TLS enabled server.
65 #
66     #     server {
67     #         listen      443 ssl http2;
68     #         listen      [::]:443 ssl http2;
69     #         server_name _;
70     #         root       /usr/share/nginx/html;
71     #
72     #         ssl_certificate "/etc/pki/nginx/server.crt";
73     #         ssl_protocols "TLSv1 TLSv1.1 TLSv1.2 TLSv1.3";
74     #         ssl_ciphers "ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-SHA256:ECDHE-RSA-AES128-SHA256";
75     #         ssl_prefer_server_ciphers on;
76     #     }
77 }
```

Uploaded WebTier code in S3 bucket also nginx.conf

The screenshot shows the AWS S3 console with the bucket 'demowebapp-ud-101' selected. The 'Objects' tab is active, displaying three items: 'app-tier/' (Folder), 'nginx.conf' (conf), and 'web-tier/' (Folder). The 'nginx.conf' file is highlighted with a red box.

Web Instance Deployment

The screenshot shows the AWS EC2 instance details page for instance 'i-08c2f029af3f40f84'. The instance is listed as 'Running' with a public IPv4 address of 54.91.74.98. Other details include the VPC ID (vpc-09c6fbba556ba4140), Subnet ID (subnet-05f41ef6206ff8ada), and Instance ARN (arn:aws:ec2:us-east-1:253490795695:instance/i-08c2f029af3f40f84).

Connected to WebTier EC2 Instance and configure it using below commands:

```
# Switch to ec2-user with root privileges
sudo -su ec2-user
ping 8.8.8.8

# Install NVM (Node Version Manager)
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.38.0/install.sh | bash
source ~/bashrc
nvm install 16
nvm use 16
```

```

# Change to home directory
cd ~/
# Download web-tier files from S3 bucket
aws s3 cp s3://BUCKET_NAME/web-tier/ web-tier --recursive

#install and build Node.js Application
cd ~/web-tier
npm install
npm run build

# Install Nginx from Amazon Linux Extras
sudo amazon-linux-extras install nginx1 -y

cd /etc/nginx
ls

# Remove default Nginx config file and download custom Nginx config file
sudo rm nginx.conf
sudo aws s3 cp s3://BUCKET_NAME/nginx.conf .

# Restart Nginx to apply new configuration
sudo service nginx restart

# Set proper permissions for ec2-user home directory
chmod -R 755 /home/ec2-user

# Enable Nginx to start on boot
sudo chkconfig nginx on

```

Part 6: External Load Balancer and Auto Scaling

Created WebTier Target Group

The screenshot shows the AWS CloudWatch Metrics Insights search interface. The search bar at the top contains the query "CloudWatch Metrics Insights". Below the search bar, there are two tabs: "Metrics" and "Logs". Under the "Metrics" tab, there is a table with columns: Metric Name, Namespace, and Value. One row shows "AWS/CloudWatchMetricsInsights/CloudWatchMetricsInsights/CloudWatchMetricsInsights" with a value of "1". At the bottom of the page, there is a navigation bar with links for "CloudShell", "Feedback", and copyright information.

Internet Facing Load Balancer created and added HTTP Listener with WebTier Target Group

The screenshot shows the AWS Cloud Console interface for managing load balancers. On the left, a navigation sidebar lists various services like AMIs, Elastic Block Store, Network & Security, Load Balancing, Auto Scaling, and more. The main content area displays a table of existing load balancers:

Name	DNS name	Status	VPC ID	Availability Zones	Type	Date created
app-tier-internal-lb	internal-app-tier-internal-lb...	Active	vpc-09c6fba5565ba4140	2 Availability Zones	application	May 14, 2025, 16:56 (UTC+05:30)
web-tier-external-lb	web-tier-external-lb-25964...	Active	vpc-09c6fba5565ba4140	2 Availability Zones	application	May 15, 2025, 14:40 (UTC+05:30)

Below the table, a detailed view of the selected load balancer "web-tier-external-lb" is shown. The "Listeners and rules" tab is active, displaying the following configuration:

- Details:** Load balancer type: Application, Status: Active, Scheme: Internet-facing, Hosted zone: Z355XDTRQ7X7K.
- Listeners and rules:** One listener is listed: port 80 (HTTP) to target group "WebServerASG".
- Network mapping:** Shows the VPC connection.
- Security:** Basic security settings.
- Monitoring:** Metrics and CloudWatch Metrics settings.
- Integrations:** No integrations defined.
- Attributes:** No attributes defined.
- Capacity:** Desired capacity: 2, Min: 2, Max: 2.
- Tags:** No tags defined.

Afterthat Launch Template created for ASG in WebTier using AMI of WebTier Instance.

Auto Scaling Group for Internet Facing Load Balancer of WebTier

The screenshot shows the AWS Cloud Console interface for managing Auto Scaling groups. The left sidebar includes options for AMIs, Elastic Block Store, Network & Security, Load Balancing, and Auto Scaling. The main area shows the successful creation of an Auto Scaling group named "WebServerASG":

Auto Scaling groups (1/2) Info

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability...
WebServerASG	WebServerLaunchTemplate Version Default	0	Updating capacity...	2	2	2	us-east-1a...
AnotherASG	AnotherLaunchTemplate Version Default	-	-	2	2	2	us-east-1a...

Auto Scaling group: WebServerASG

Details tab details:

- WebServerASG Capacity overview:** Desired capacity: 2, Scaling limits (Min - Max): 2 - 2, Desired capacity type: Units (number of instances), Status: Updating capacity.
- Date created:** Thu May 15 2025 14:50:04 GMT+0530 (India Standard Time).

Launch template tab details:

- Launch template:** WebServerLaunchTemplate | Version Default.
- Launch configuration:** WebServerLaunchTemplate | Version Default.

Then tested Application Connectivity using Internet facing Load Balancer DNS into Browser.

ID	AMOUNT	DESC
1	300	food
2	50	milk
3	400	milk
4	250	cold drink

Part 7: Website Hosting using Route53 and Cloudfront

Create Hosted Zone in Route 53 for your domain (saurabhcloud.fun).

saurabhcloud.fun was successfully created.
Now you can create records in the hosted zone to specify how you want Route 53 to route traffic for your domain.

Record name	Type	Value/Route traffic to	TTL (s...)
saurabhcloud.f...	NS	Simple ns-923.awsdns-51.net. ns-1248.awsdns-28.org. ns-367.awsdns-45.com. ns-1985.awsdns-56.co.uk.	172800
saurabhcloud.f...	SOA	Simple ns-923.awsdns-51.net. awsd...	900

Updated Name Servers in your domain provider(Hostinger) with NS records from the hosted zone.

The screenshot shows the 'DNS / Nameservers' section of the Hostinger control panel. On the left sidebar, 'DNS / Nameservers' is selected. In the main content area, under 'Nameservers', four new nameservers are listed: ns-1248.awsdns-28.org, ns-1985.awsdns-56.co.uk, ns-367.awsdns-45.com, and ns-923.awsdns-51.net. A red box highlights these four entries. Below them is a 'Change Nameservers' button. Another section titled 'Manage DNS records' contains a message: 'Your domain is not pointing to Hostinger. To manage DNS records, change your nameservers to Hostinger's'. A red box highlights the 'United States (N. Virginia)' dropdown menu at the top right of the page.

Requested an ACM Certificate (in us-east-1 for CloudFront) with your domain name ([saurabhcloud.fun](#)).

The screenshot shows the AWS Certificate Manager console. It displays a certificate named '46fee873-79a4-464b-a865-c01b6ce34ace'. The 'Certificate status' section shows it is 'Issued'. The 'Domains' section lists 'saurabhcloud.fun' with a CNAME record pointing to '_98a4764c852ceb187b610dea7100e120.saurabhcloud.fun'. The 'Details' section provides certificate metadata: Serial number 07:96:c7:c6:d2:9f:74:a2:a4:a3:35:4e:9f:7d:21:d8, Requested at May 16, 2025, 14:17:41 (UTC+05:30), and Renewal eligibility Ineligible. A red box highlights the 'United States (N. Virginia)' dropdown menu at the top right of the page.

Validated the certificate by adding a CNAME record in Route 53 as per ACM instructions.

The screenshot shows the AWS Route 53 console with the hosted zone for `saurabhcloud.fun`. The left sidebar shows various AWS services like CloudFront, CloudWatch, and CloudWatch Metrics. The main pane shows the hosted zone details with three records listed:

Record name	Type	Value	TTL
saurabhcloud.fun	NS	ns-923.awsdns-51.net. ns-1248.awsdns-28.org. ns-367.awsdns-45.com. ns-1985.awsdns-56.co.uk.	172800
saurabhcloud.fun	SOA	ns-923.awsdns-51.net.awsd...	900
_98a4764c852ce...	CNAME	_c9a889a1dbd8589fb9f93b...	300

Created a CloudFront distribution with the ALB DNS as the origin and custom domain name set (www.saurabhcloud.fun).

The screenshot shows the AWS CloudFront console with the distribution `E2TKU1HES7IPKT`. The left sidebar shows various AWS services like CloudFront, CloudWatch Metrics, and CloudWatch Metrics Insights. The main pane shows the distribution details:

General

- Distribution domain name:** `d2k5x0860v8w60.cloudfront.net`
- ARN:** `arn:aws:cloudfront::253490795695:distribution/E2TKU1HES7IPKT`
- Last modified:** May 16, 2025 at 9:40:58 AM UTC

Settings

- Description:** -
- Alternate domain names:** `saurabhcloud.fun`
- Custom SSL certificate:** `saurabhcloud.fun`
- Standard logging:** Off
- Cookie logging:** Off
- Default root object:** -

Continuous deployment: [Create staging distribution](#)

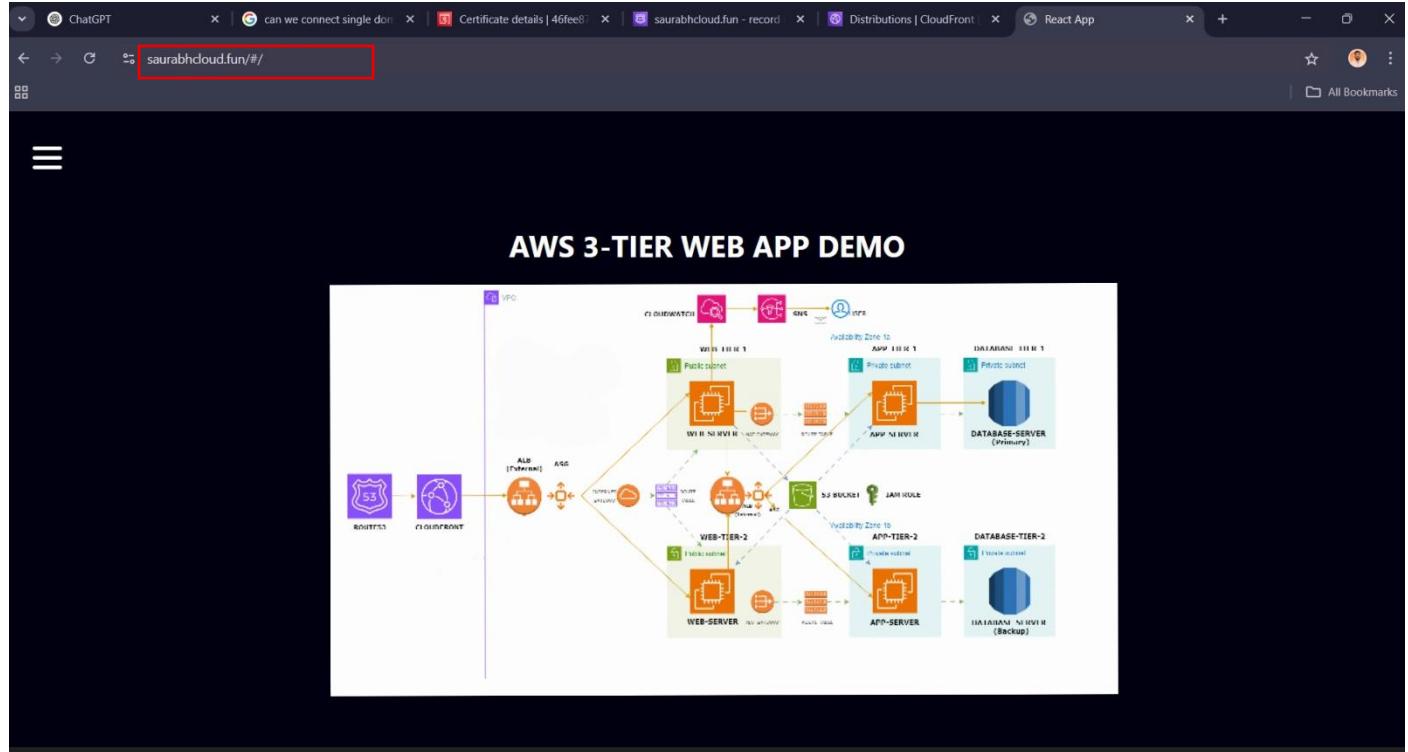
Also Attached the WAF Web ACL to your CloudFront distribution via its settings.

The screenshot shows the AWS CloudFront Distributions settings page. Under the 'Web Application Firewall (WAF)' section, the 'Enable security protections' option is selected, with a note explaining it keeps the application secure from common web threats. Below this, the 'Use monitor mode' option is selected, indicating how many requests would be blocked. The 'Included security protections' section lists three items: protecting against common vulnerabilities, malicious actors, and blocking IP addresses. The 'Additional protections for dynamic applications and APIs' section has 'SQL protections' and 'Rate limiting' checked. 'Rate limiting' is set to 300 requests per IP address per 5-minute period, with a note that it's a starting point and will be placed in monitor mode. At the bottom, there are links for 'Price estimate', 'CloudShell', and 'Feedback'.

Created an A (Alias) Record in Route 53 pointing www.saurabhcloud.fun to the CloudFront distribution.

The screenshot shows the AWS Route 53 Hosted Zones settings page. Under the 'Create record' section, a new 'Record 1' is being created. The 'Record name' is 'saurabhcloud.fun' and the 'Record type' is 'A - Routes traffic to an IPv4 address and some AWS resources'. The 'Route traffic to' field is set to 'Alias to CloudFront distribution' with 'US East (N. Virginia)' selected. The 'Routing policy' is 'Simple routing'. The 'Evaluate target health' option is turned off. At the bottom, there are buttons for 'Add another record', 'Cancel', and 'Create records'.

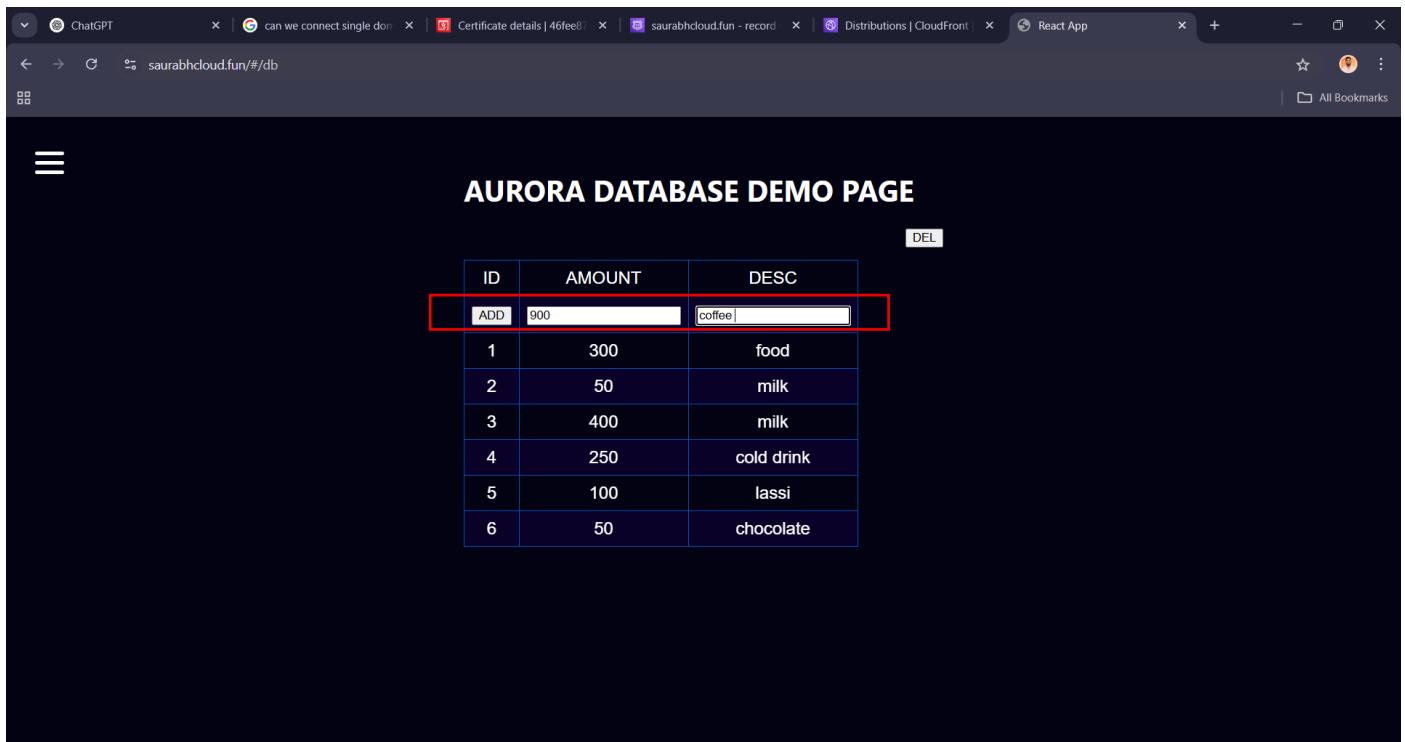
Afterthat Accessed My web application with Domain Name saurabhcloud.fun



The screenshot shows the Aurora Database Demo Page. The left sidebar has links for "HOME" and "DB DEMO". The main content area displays a table of data from the database:

ID	AMOUNT	DESC
1	300	food
2	50	milk
3	400	milk
4	250	cold drink
5	100	lassi
6	50	chocolate

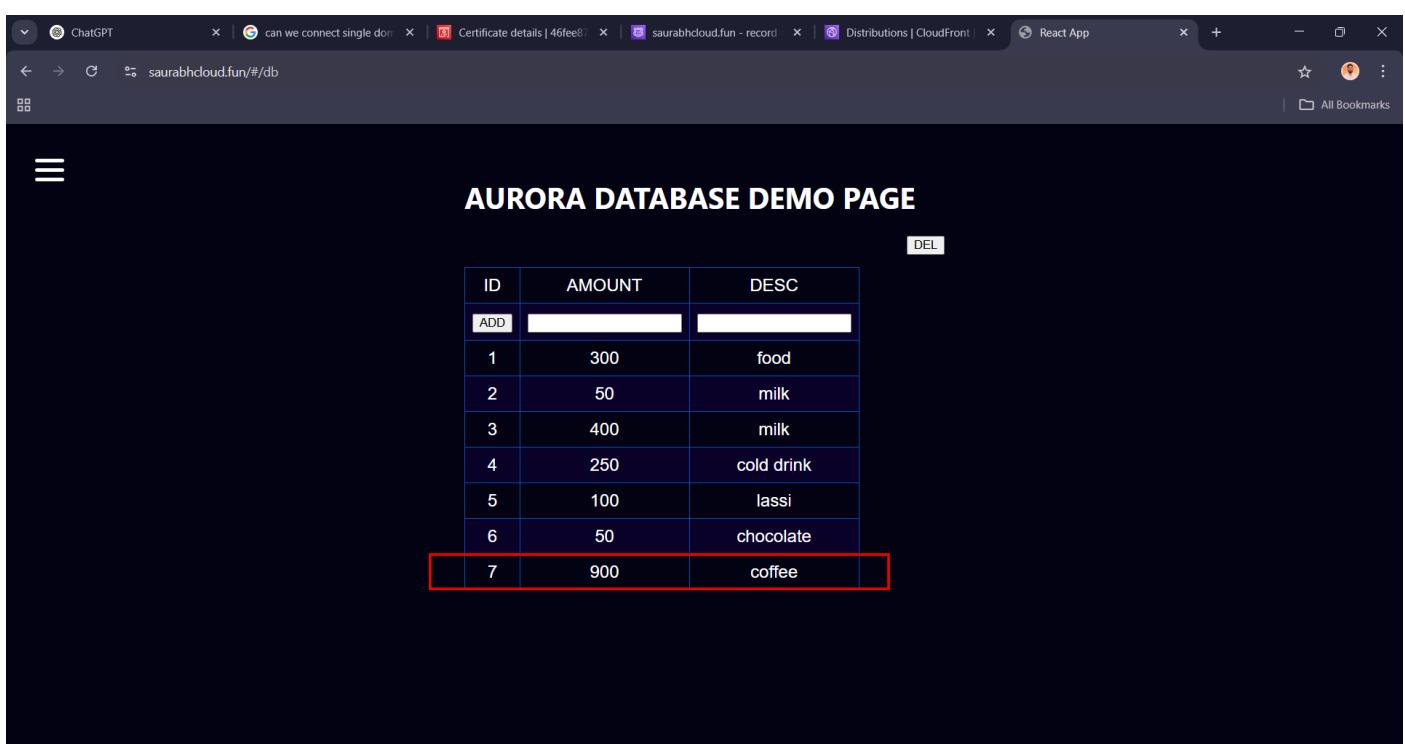
Buttons for "ADD" and "DEL" are visible above the table.



AURORA DATABASE DEMO PAGE

[DEL]

ID	AMOUNT	DESC
	ADD <input type="text" value="900"/>	<input type="text" value="coffee"/>
1	300	food
2	50	milk
3	400	milk
4	250	cold drink
5	100	lassi
6	50	chocolate



AURORA DATABASE DEMO PAGE

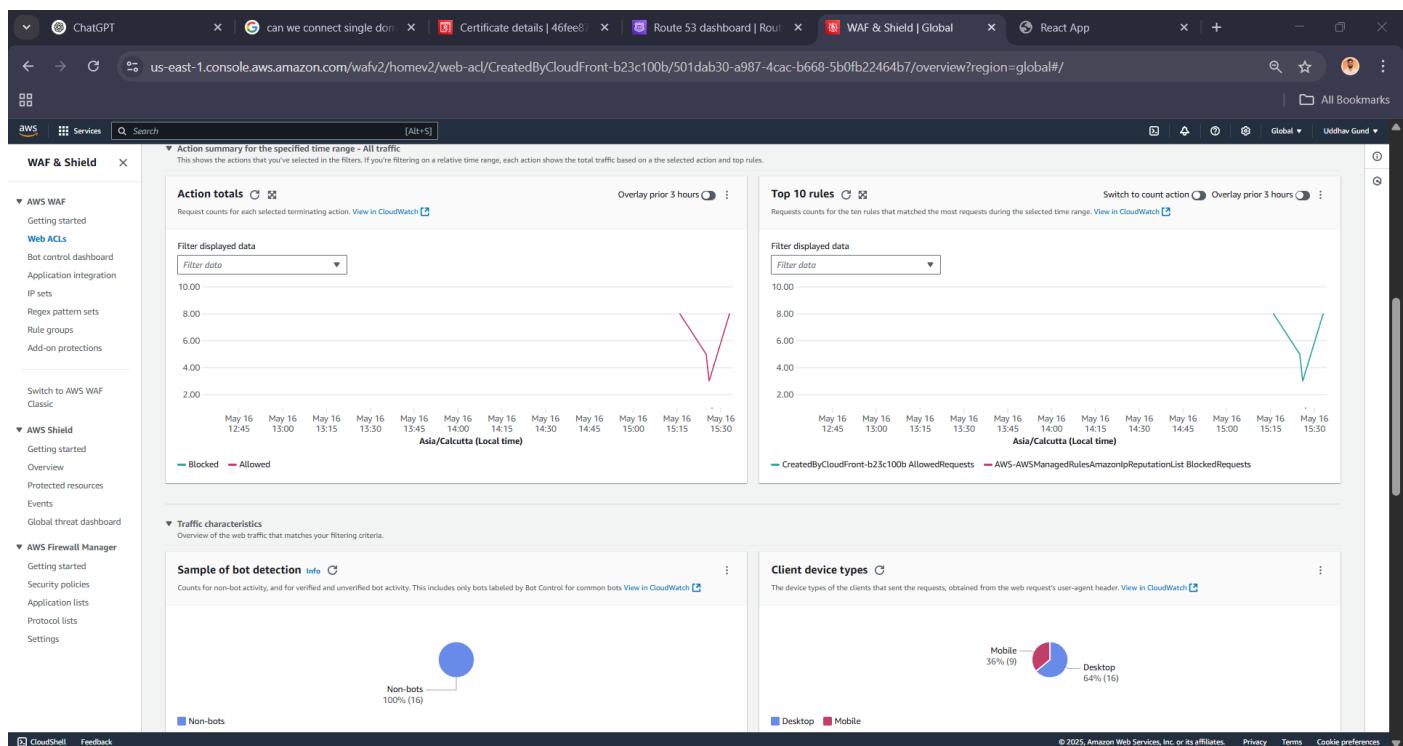
[DEL]

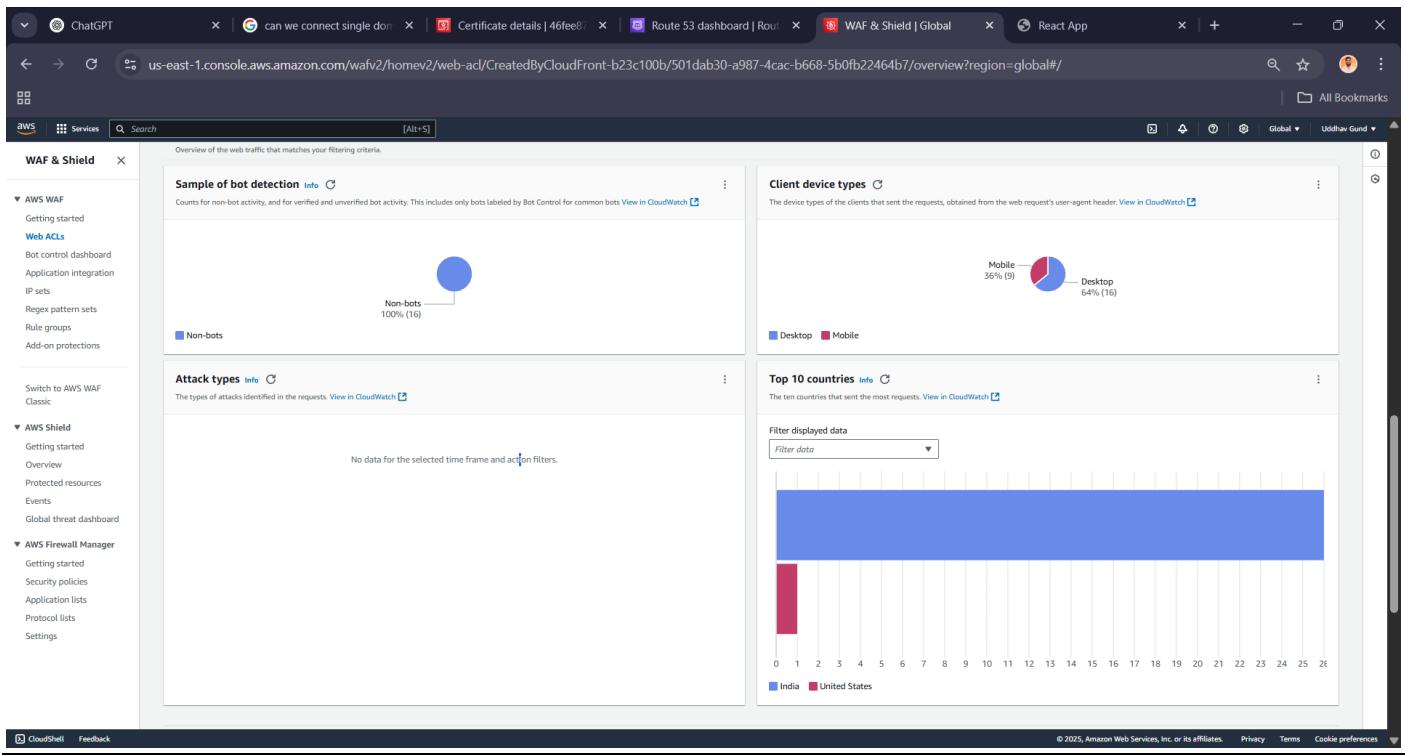
ID	AMOUNT	DESC
	ADD <input type="text"/>	<input type="text"/>
1	300	food
2	50	milk
3	400	milk
4	250	cold drink
5	100	lassi
6	50	chocolate
7	900	coffee

All WAF ACL's Report:

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

WAF & Shield	Metric name	Source IP	URI	Rule inside rule group	Action	Time
▼ AWS WAF	CreatedByCloudFront-b23c100b	103.81.242.250 (IN)	/api/transaction	-	ALLOW	Fri May 16 2025 15:16:47 GMT+0530 (India Standard Time)
	CreatedByCloudFront-b23c100b	103.81.242.250 (IN)	/static/js/main.31b1a151.js	-	ALLOW	Fri May 16 2025 15:25:35 GMT+0530 (India Standard Time)
	CreatedByCloudFront-b23c100b	103.81.242.250 (IN)	/static/js/main.31b1a151.js	-	ALLOW	Fri May 16 2025 15:16:10 GMT+0530 (India Standard Time)
	CreatedByCloudFront-b23c100b	103.81.242.250 (IN)	/api/transaction	-	ALLOW	Fri May 16 2025 15:16:47 GMT+0530 (India Standard Time)
	CreatedByCloudFront-b23c100b	103.81.242.250 (IN)	/	-	ALLOW	Fri May 16 2025 15:16:10 GMT+0530 (India Standard Time)
	CreatedByCloudFront-b23c100b	103.81.242.250 (IN)	/api/transaction	-	ALLOW	Fri May 16 2025 15:26:06 GMT+0530 (India Standard Time)
	CreatedByCloudFront-b23c100b	103.81.242.250 (IN)	/static/media/3TierArch.0486e7150e53d305d1c2.png	-	ALLOW	Fri May 16 2025 15:16:11 GMT+0530 (India Standard Time)
	CreatedByCloudFront-b23c100b	103.81.242.250 (IN)	/	-	ALLOW	Fri May 16 2025 15:25:34 GMT+0530 (India Standard Time)
	CreatedByCloudFront-b23c100b	103.81.242.250 (IN)	/favicon.ico	-	ALLOW	Fri May 16 2025 15:16:12 GMT+0530 (India Standard Time)

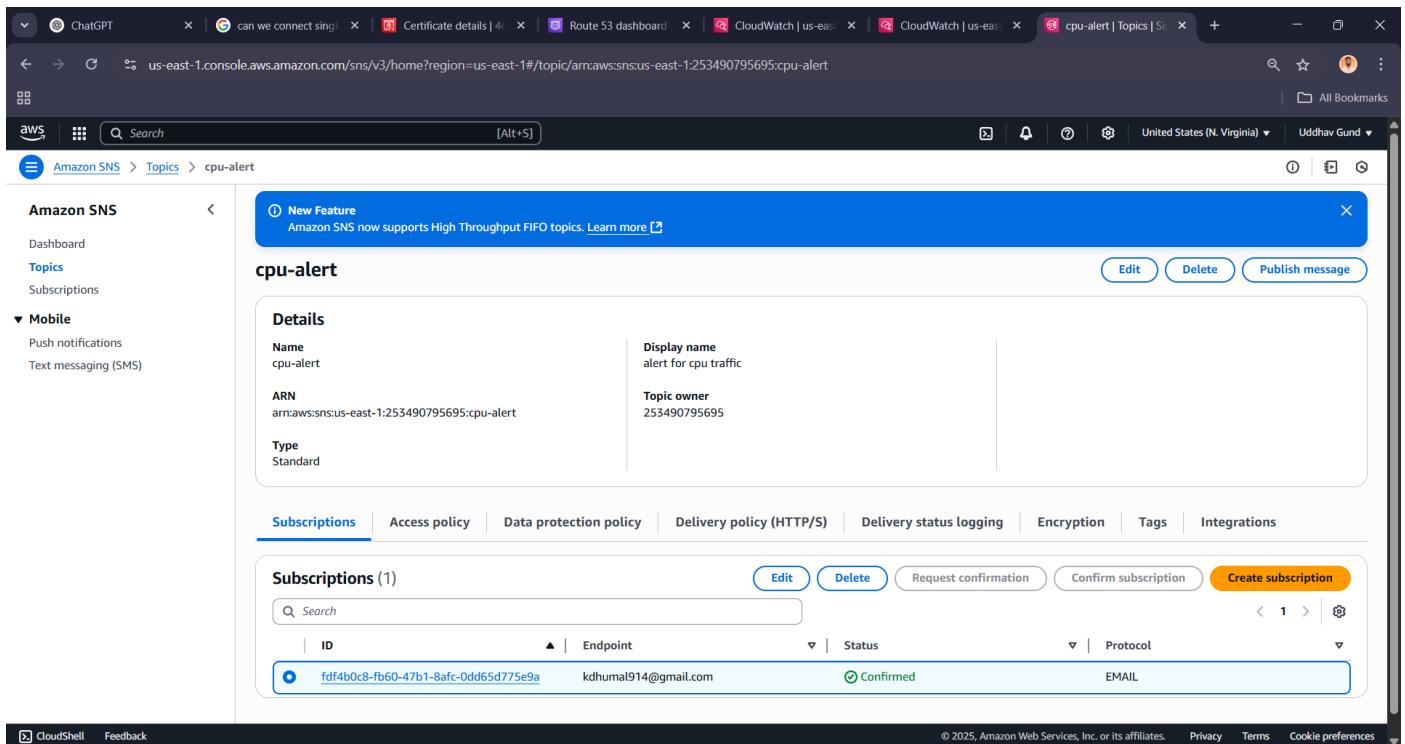




Part 8: CloudWatch and SNS integration

Created an **SNS topic (cpu-alert)** in Amazon SNS.

Subscribed my **email address to the SNS topic** and confirmed the subscription via email.

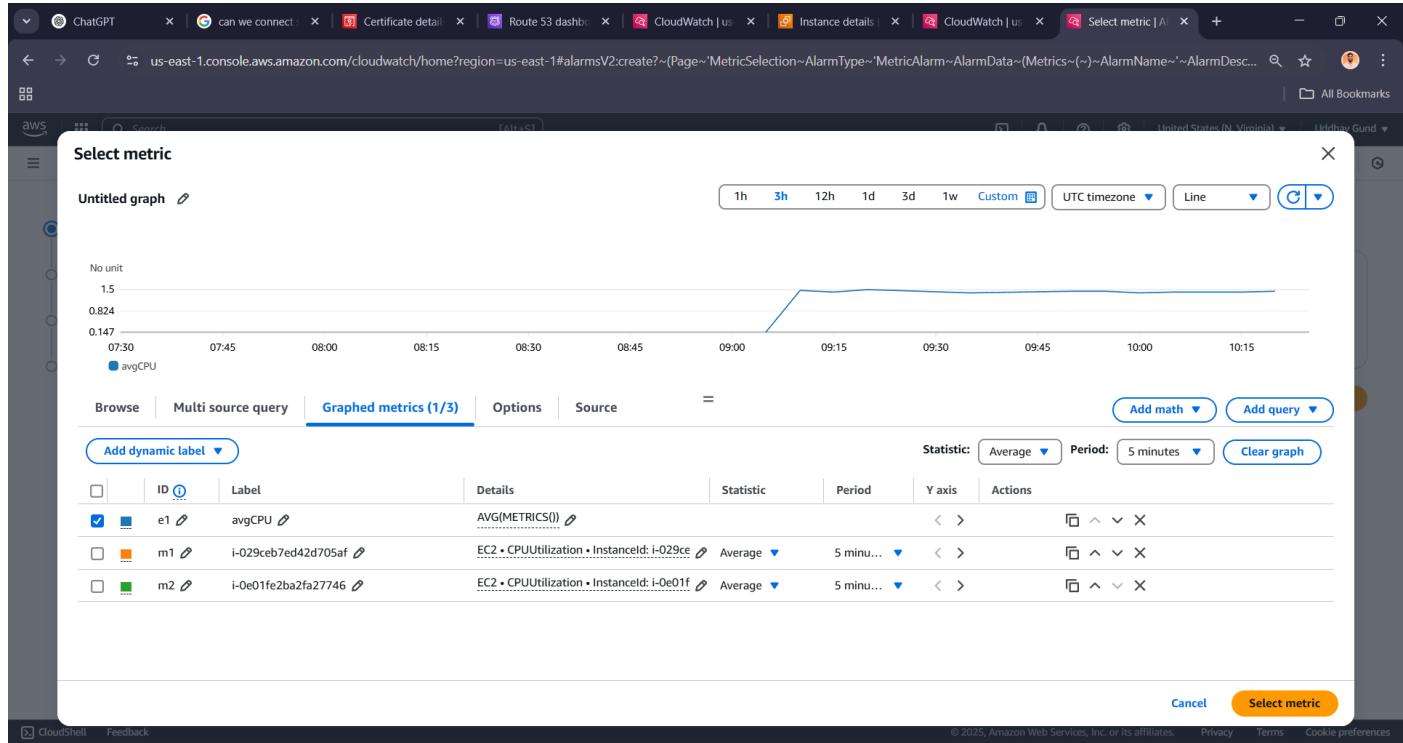


Navigated to CloudWatch → Alarms and clicked "Create Alarm".

Selected EC2 → Per-Instance Metrics → CPUUtilization for the desired instance(s).

Added a math expression (if needed) like AVG([m1, m2]) to calculate average CPU.

Chose the math expression or single instance metric as the alarm target.



Chose the math expression or single instance metric as the alarm target.

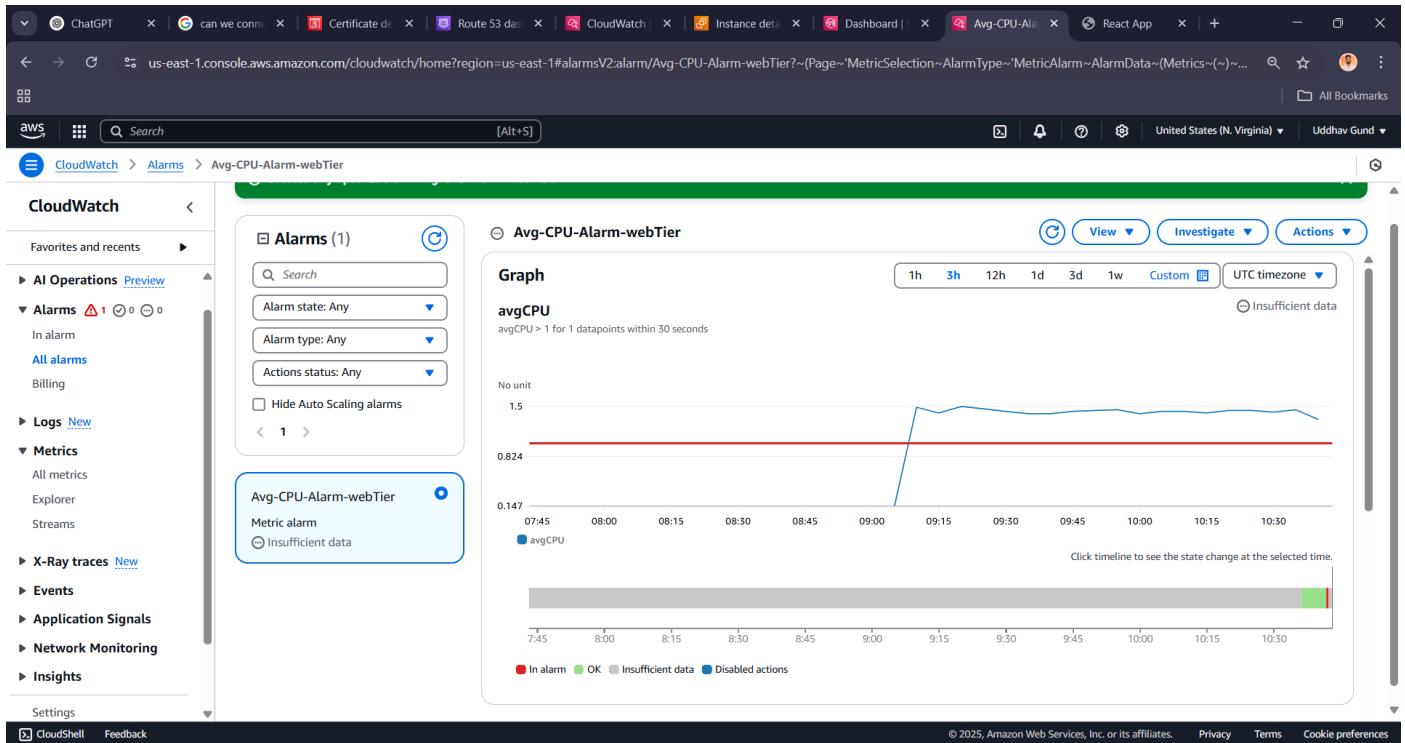
Configured the alarm condition (e.g., CPUUtilization > 1% for 30 seconds).

Attached the previously created SNS topic as the notification action.

Named the alarm and created it.

The screenshot shows the 'Create alarm' interface in the AWS CloudWatch console, specifically the 'Configure actions' step. On the left, a sidebar shows steps: Step 1 (Specify metric and conditions), Step 2 (Configure actions - currently selected), Step 3 (Add name and description), and Step 4 (Preview and create). The main area is titled 'Configure actions'. It starts with a 'Notification' section. Under 'Alarm state trigger', there are three options: 'In alarm' (selected), 'OK', and 'Insufficient data'. Below this, there's a section 'Send a notification to...' with a search bar containing 'cpu-alert'. It says 'Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.' Below the search bar are three options: 'Select an existing SNS topic' (selected), 'Create new topic', and 'Use topic ARN to notify other accounts'. At the bottom of this section is a button 'Add notification'. Below this is a 'Lambda action' section, which is currently empty.

Triggered high CPU usage to test the alarm.



Received an **email notification** from SNS when the alarm state changed to ALARM.

ALARM: "Avg-CPU-Alarm-webTier" in US East (N. Virginia)

A alert for cpu traffic<no-reply@sns.amazonaws.com>
To: You
Fri 2025-05-16 16:11

You are receiving this email because your Amazon CloudWatch Alarm "Avg-CPU-Alarm-webTier" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [1.3416890281504692 (16/05/25 10:40:00)] was greater than the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Friday 16 May, 2025 10:41:49 UTC".

View this alarm in the AWS Management Console:
<https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/Avg-CPU-Alarm-webTier>

Alarm Details:

- Name: Avg-CPU-Alarm-webTier
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [1.3416890281504692 (16/05/25 10:40:00)] was greater than the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Friday 16 May, 2025 10:41:49 UTC
- AWS Account: 253490795695
- Alarm Arn: arn:aws:cloudwatch:us-east-1:253490795695:alarm:Avg-CPU-Alarm-webTier

Threshold:

- The alarm is in the ALARM state when the metric is GreaterThanThreshold 1.0 for at least 1 of the last 1 period(s) of 30 seconds.

Monitored Metrics:

- MetricExpression: AVG(METRICS())
- MetricLabel: avgCPU

State Change Actions:

- OK:
- ALARM: [arn:aws:sns:us-east-1:253490795695:cpu-alert]
- INSUFFICIENT_DATA:

--
If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
<https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:253490795695:cpu-alert:fdf4b0c8-fb60-47b1-8afc-0dd65d775e9a&Endpoint=kdhuma914@gmail.com>

Successfully! Implemented AWS 3-Tier Web Architecture!