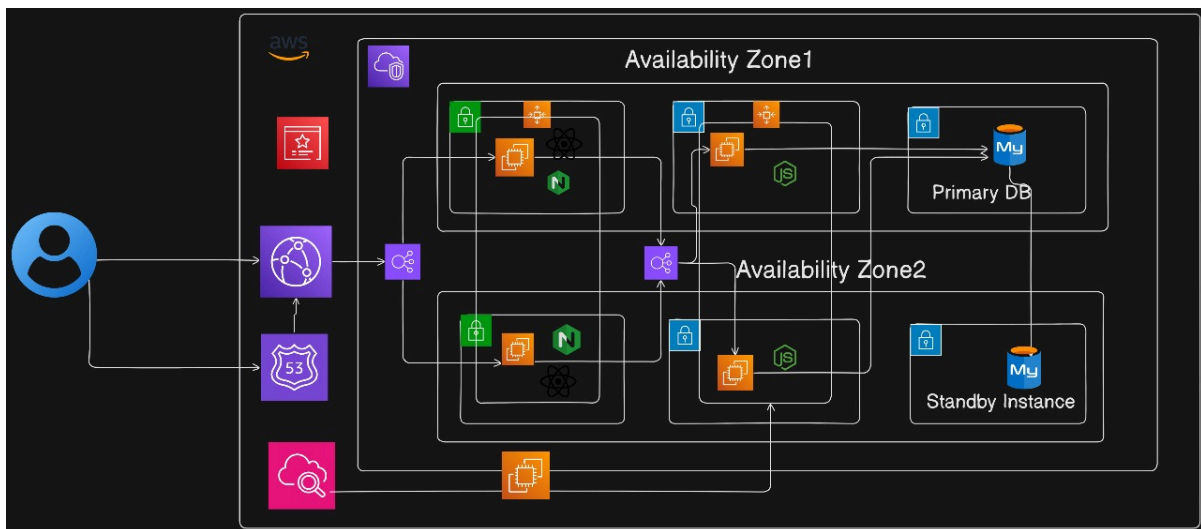


Project-1

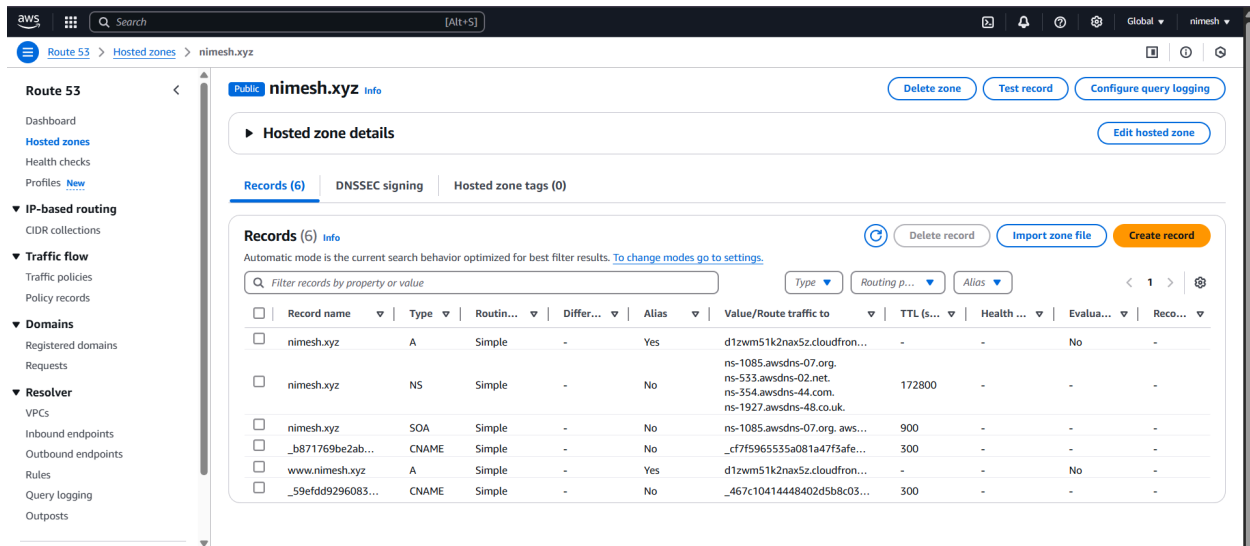
Deploying a Production Grade Highly Available & Scalable 3 Tier Architecture in AWS

Architecture of the application:



Step 1: Configuring Route53:

- Go to Route53 in AWS -
- Click on "Create Hosted Zone" .
- Give "Domain name" and Select Public Hosted Zone.
- Once Created, You will get NS records.
- Add those NS records in DNS of your Domain Provider (Our Case, GoDaddy).
- It will take more than an hour to propagate.



Step 2: Requesting a Public SSL Certificate using Amazon ACM:

- Go to Certificate Manger in AWS -
- Click on "Request a Certificate".
- Give your Domain Name, (Ex: `nimesh.xyz, www.nimesh.xyz`).
- Once Created, Click on "Create Records in Route53".
- Certificate will be issued.

e804568d-1b8c-47bd-8c79-e8378493378e Delete

Certificate status

Identifier: e804568d-1b8c-47bd-8c79-e8378493378e Status: Issued

ARN: arn:aws:acm:us-east-1:504235115994:certificate/e804568d-1b8c-47bd-8c79-e8378493378e

Type: Amazon issued

Domains (2) Create records in Route 53 Export to CSV

Domain	Status	Renewal status	Type	CNAME name	CNAME value
rimshu.yz	Success	-	CNAME	_b871769be2ab000e0614f168b817c7e.rimshu.yz.	_cf1f996555a081e47f5afe7b7552efc2d8gmvalideconu.aws.
www.rimshu.yz	Success	-	CNAME	_59e6d823608592f5aa4d7071044e059.www.rimshu.yz.	_467c10416448402d5b8c037414677fba.xdfgmvalideconu.aws.

Details

In use Yes	Serial number 0db771fa406ff61601a1b1a1cf857d015519	Requested at April 20, 2025, 14:08:25 (UTC+05:30)	Renewal eligibility Eligible
Domain name rimshu.yz	Public key info RSA 2048	Issued at April 20, 2025, 14:53:12 (UTC+05:30)	
Number of additional names 1	Signature algorithm SHA-256 with RSA	Not before April 20, 2025, 05:52:00 (UTC+05:30)	
	Can be used with CloudFront, Elastic Load Balancing, API Gateway and other integrated services.	Not after May 20, 2026, 05:29:59 (UTC+05:30)	

Step 3: Creating VPC & Subnets:

- Go to VPC Dashboard in AWS -
- Click on "Create VPC".
- Click on "VPC and More".
- Name: 3-tierproject.
- Availability Zones: 2
- Public Subnets: 2
- Private Subnets: 4
- Nat Gateway: In 1 AZ
- VPC End Points: None
- Click on Create.
- Go to Subnets, Select Public subnets.
- Click on Actions.
- Click on Edit Subnet Settings.

- Enable Public IP Automatically.

vpc-06365455b981b63a1 / 3-tierproject-vpc

Actions

Details

Info

VPC ID
vpc-06365455b981b63a1

DNS resolution
Enabled

Main network ACL
acl-059b5acca71fd54e3

IPv6 CIDR (Network border group)
-

State
Available

Tenancy
default

Default VPC
No

Network Address Usage metrics
Disabled

Block Public Access
Off

DHCP option set
dopt-0e276da483618d612

IPv4 CIDR
10.0.0.0/16

Route 53 Resolver DNS Firewall rule groups
-

DNS hostnames
Enabled

Main route table
rtb-0ab439710d6869260

IPv6 pool
-

Owner ID
904233113994

Resource map | CIDRs | Flow logs | Tags | Integrations

Resource map

Info

VPC

Show details
Your AWS virtual network

3-tierproject-vpc

Subnets (6)

Subnets within this VPC

us-east-1a

3-tierproject-subnet-public1-us-e...

3-tierproject-subnet-private3-us-e...

3-tierproject-subnet-private1-us-e...

us-east-1b

3-tierproject-subnet-public2-us-e...

3-tierproject-subnet-private2-us-e...

3-tierproject-subnet-private4-us-e...

Route tables (6)

Route network traffic to resources

3-tierproject-rtb-private3-us-east-1a

rtb-0ab439710d6869260

3-tierproject-rtb-public

3-tierproject-rtb-private4-us-east-1b

3-tierproject-rtb-private2-us-east-1b

3-tierproject-rtb-private1-us-east-1a

Network connections (2)

Connections to other networks

3-tierproject-igw

3-tierproject-nat-public1-us-east-1a

Step 4: Creating Security Groups:

4.1: Creating security group for Bastion Host

- Go to Security groups in EC2 Dashboard.
- Click on Create Security Group.
- Name: Bastion-Host.
- VPC: Select 3-tierproject.
- Inbound Rule: SSH from Anywhere from Ipv4.

4.2: Create a Security Group for Presentation Tier ALB

- Click on Create Security Group.
- Name: Presentation-Tier-ALB.
- VPC: Select 3-tierproject.
- Inbound Rule: HTTP from Anywhere from Ipv4.

4.3: Creating Security Group for Presentation Tier EC2

- Click on Create Security Group
- Name: Presentation-Tier-EC2
- VPC: Select 3-tierproject.
- Inbound Rules:
 - SSH from Bastion Host
 - HTTP from Presentation-Tier-ALB

4.4: Creating Security Group for Application Tier ALB

- Click on Create Security Group
- Name: Application-Tier-ALB
- VPC: Select 3-tierproject.
- Inbound Rule: HTTP from Presentation-Tier-EC2

4.5: Creating Security Group for Application Tier EC2

- Click on Create Security Group
- Name: Application-Tier-EC2

- VPC: Select 3-tierproject.
- Inbound Rules:
 - SSH from Bastion Host
 - Custom TCP (Port 3200) from App-Tier-ALB

4.6: Creating Security Group for Data Tier

- Click on Create Security Group
- Name: Data-Tier
- VPC: Select 3-tierproject.
- Inbound Rules:
 - MySQL/Aurora from Bastion Host
 - MySQL/Aurora from Application-Tier-EC2.

Security Groups (6/39) [info](#)

Find security groups by attribute or tag

Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound rules count
launch-wizard-11	sg-059813387920247	launch-wizard-11	vpc-0f0b7f5e9b9c11	launch-wizard-11 created 2024-12-15T...	904225113994	2 Permission entries	1 Permission entry
Presentation-Tier-EC2	sg-0502210f210729a9	Presentation-Tier-EC2	vpc-0f0b7f5e9b9c11	Presentation-Tier-EC2	904225113994	2 Permission entries	1 Permission entry
launch-wizard-21	sg-05a950271876230	launch-wizard-21	vpc-0f0b7f5e9b9c11	launch-wizard-21 created 2025-04-14T...	904225113994	2 Permission entries	1 Permission entry
bastionhost	sg-05e51d99a81386a8	bastionhost	vpc-0f0b7f5e9b9c11	bastionhost	904225113994	1 Permission entry	1 Permission entry
launch-wizard-17	sg-0f0b7f5e9b9c11	launch-wizard-17	vpc-0f0b7f5e9b9c11	launch-wizard-17 created 2025-03-12T...	904225113994	1 Permission entry	1 Permission entry
launch-wizard-13	sg-0f0b7f5e9b9c11	launch-wizard-13	vpc-0f0b7f5e9b9c11	launch-wizard-13 created 2025-02-06T...	904225113994	2 Permission entries	1 Permission entry
launch-wizard-9	sg-0f0b7f5e9b9c11	launch-wizard-9	vpc-0f0b7f5e9b9c11	launch-wizard-9 created 2024-12-15T...	904225113994	1 Permission entry	1 Permission entry
launch-wizard-5	sg-0f0b7f5e9b9c11	launch-wizard-5	vpc-0f0b7f5e9b9c11	launch-wizard-5 created 2024-12-07T...	904225113994	1 Permission entry	1 Permission entry
default	sg-0f0b7f5e9b9c11	default	vpc-0f0b7f5e9b9c11	default VPC security group	904225113994	1 Permission entry	1 Permission entry
launch-wizard-29	sg-0f0b7f5e9b9c11	launch-wizard-29	vpc-0f0b7f5e9b9c11	launch-wizard-29 created 2025-01-21T...	904225113994	1 Permission entry	1 Permission entry
launch-wizard-12	sg-0f0b7f5e9b9c11	launch-wizard-12	vpc-0f0b7f5e9b9c11	launch-wizard-12 created 2025-01-09T...	904225113994	1 Permission entry	1 Permission entry
launch-wizard-25	sg-0f0b7f5e9b9c11	launch-wizard-25	vpc-0f0b7f5e9b9c11	launch-wizard-25 created 2025-04-14T...	904225113994	2 Permission entries	1 Permission entry
launch-wizard-14	sg-0f0b7f5e9b9c11	launch-wizard-14	vpc-0f0b7f5e9b9c11	launch-wizard-14 created 2025-03-11T...	904225113994	1 Permission entry	1 Permission entry
launch-wizard-4	sg-0f0b7f5e9b9c11	launch-wizard-4	vpc-0f0b7f5e9b9c11	launch-wizard-4 created 2024-12-13T...	904225113994	1 Permission entry	1 Permission entry
launch-wizard-27	sg-0f0b7f5e9b9c11	launch-wizard-27	vpc-0f0b7f5e9b9c11	launch-wizard-27 created 2025-04-14T...	904225113994	2 Permission entries	1 Permission entry
presentation-tier-alb	sg-0f0b7f5e9b9c11	presentation-tier-alb	vpc-0f0b7f5e9b9c11	presentation-tier-alb	904225113994	1 Permission entry	1 Permission entry
launch-wizard-50	sg-0f0b7f5e9b9c11	launch-wizard-50	vpc-0f0b7f5e9b9c11	launch-wizard-50 created 2025-04-21T...	904225113994	1 Permission entry	1 Permission entry
launch-wizard-16	sg-0f0b7f5e9b9c11	launch-wizard-16	vpc-0f0b7f5e9b9c11	launch-wizard-16 created 2025-03-12T...	904225113994	1 Permission entry	1 Permission entry
Application-Tier-ALB	sg-0f0b7f5e9b9c11	Application-Tier-ALB	vpc-0f0b7f5e9b9c11	Application-Tier-ALB	904225113994	1 Permission entry	1 Permission entry
launch-wizard-10	sg-0f0b7f5e9b9c11	launch-wizard-10	vpc-0f0b7f5e9b9c11	launch-wizard-10 created 2024-12-15T...	904225113994	2 Permission entries	1 Permission entry
launch-wizard-20	sg-0f0b7f5e9b9c11	launch-wizard-20	vpc-0f0b7f5e9b9c11	launch-wizard-20 created 2025-04-14T...	904225113994	2 Permission entries	1 Permission entry
launch-wizard-8	sg-0f0b7f5e9b9c11	launch-wizard-8	vpc-0f0b7f5e9b9c11	launch-wizard-8 created 2024-12-14T...	904225113994	1 Permission entry	1 Permission entry
launch-wizard-24	sg-0f0b7f5e9b9c11	launch-wizard-24	vpc-0f0b7f5e9b9c11	launch-wizard-24 created 2025-04-14T...	904225113994	2 Permission entries	1 Permission entry
default	sg-0f0b7f5e9b9c11	default	vpc-0f0b7f5e9b9c11	default VPC security group	904225113994	1 Permission entry	1 Permission entry
launch-wizard-23	sg-0f0b7f5e9b9c11	launch-wizard-23	vpc-0f0b7f5e9b9c11	launch-wizard-23 created 2025-04-14T...	904225113994	2 Permission entries	1 Permission entry
launch-wizard-22	sg-0f0b7f5e9b9c11	launch-wizard-22	vpc-0f0b7f5e9b9c11	launch-wizard-22 created 2025-04-14T...	904225113994	2 Permission entries	1 Permission entry
launch-wizard-7	sg-0f0b7f5e9b9c11	launch-wizard-7	vpc-0f0b7f5e9b9c11	launch-wizard-7 created 2024-12-14T...	904225113994	1 Permission entry	1 Permission entry
launch-wizard-19	sg-0f0b7f5e9b9c11	launch-wizard-19	vpc-0f0b7f5e9b9c11	launch-wizard-19 created 2025-04-14T...	904225113994	1 Permission entry	1 Permission entry
Data-Tier	sg-0f0b7f5e9b9c11	Data-Tier	vpc-0f0b7f5e9b9c11	Data-Tier	904225113994	2 Permission entries	1 Permission entry
launch-wizard-6	sg-0f0b7f5e9b9c11	launch-wizard-6	vpc-0f0b7f5e9b9c11	launch-wizard-6 created 2024-12-14T...	904225113994	1 Permission entry	1 Permission entry
launch-wizard-28	sg-0f0b7f5e9b9c11	launch-wizard-28	vpc-0f0b7f5e9b9c11	launch-wizard-28 created 2025-04-16T...	904225113994	2 Permission entries	1 Permission entry
launch-wizard-15	sg-0f0b7f5e9b9c11	launch-wizard-15	vpc-0f0b7f5e9b9c11	launch-wizard-15 created 2025-02-11T...	904225113994	2 Permission entries	1 Permission entry
Application-Tier-EC2	sg-0f0b7f5e9b9c11	Application-Tier-EC2	vpc-0f0b7f5e9b9c11	Application-Tier-EC2	904225113994	2 Permission entries	1 Permission entry
jenkins	sg-0f0b7f5e9b9c11	jenkins	vpc-0f0b7f5e9b9c11	jenkins	904225113994	2 Permission entries	1 Permission entry
launch-wizard-2	sg-0f0b7f5e9b9c11	launch-wizard-2	vpc-0f0b7f5e9b9c11	launch-wizard-2 created 2024-11-30T...	904225113994	1 Permission entry	1 Permission entry

Step 5: Launching Bastion Host:

Go to EC2 Dashboard and click on "Launch Instance." Configure the following settings:

- Name: Bastion-Host

- AMI: Amazon Linux 2023 AMI
- Instance Type: t2.micro
- Key Pair: Select existing or create new
- VPC: 3-tierproject.
- Subnet: Public Subnet
- Security Group: Bastion-Host

Finally, click "Launch Instance."

Step 6: Setting Up Data Tier with RDS:

Go to Amazon RDS in AWS. Click "Create DB Subnet Group" and configure:

- Name: dev-db-subnet-group
- VPC: 3-tierproject
- AZs: Select both
- Subnets: Select 2 private subnets

Click "Create Subnet Group."

dev-db-subnet-group

Subnet group details

VPC ID

vpc-06365455b981b63a1 [\[?\]](#)

ARN

arn:aws:rds:us-east-1:904233113994:subgrp:dev-db-subnet-group

Supported network types

IPv4

Description

subnet-group

Then click "Create Database" and configure:

- Choose Standard creation
- Select SQL. Template: dev/test
- Enable Multi-AZ DB Instance
- Database identifier: dev-db-instance
- Set password: Create your own (use "admin123")
- VPC: 3-tierproject
- DB subnet group: Choose the one created earlier
- Security group: Data-Tier

dev-db-instance Modify Actions

Summary				
DB Identifier dev-db-instance	Status Available	Role Instance	Engine MySQL Community	Recommendations
CPU 2.63%	Class db.m7g.large	Current activity 0.00 sessions	Region & AZ us-east-1a	

Connectivity & security | Monitoring | Logs & events | Configuration | Zero-ETL integrations | Maintenance & backups | Data migrations - new | Tags | Recommendations

Connectivity & security

Endpoint & port

Endpoint
[dev-db-instance.cvqkk8ciu30e.us-east-1.rds.amazonaws.com](#)

Port
3306

Networking

Availability Zone
us-east-1a

VPC
[3-tierproject-vpc \(vpc-06365455b981b63a1\)](#)

Subnet group
dev-db-subnet-group

Subnets
[subnet-002ca28187c1fc726](#)
[subnet-018e9a56d5a0be6e4](#)

Network type
IPv4

Security

VPC security groups
[Data-Tier \(sg-0684ac5a85a16ab14\)](#)
Active

Publicly accessible
No

Certificate authority
[rds-ca-rsa2048-g1](#)

Certificate authority date
May 26, 2061, 05:04 (UTC+05:30)

DB instance certificate expiration date
April 29, 2026, 14:32 (UTC+05:30)

Click "Create Database." Note that database creation typically takes 15-20 minutes. After your database is running, open Command Prompt and locate your key pair.

- Add key to SSH Agent.

```
ssh-add your_key.pem
```

- Connect to DataBase.

```
ssh -N -L 3307:dev-db-instance.cvqkk8ciu30e.us-east-1.rds.amazonaws.com:3306 ec2-user@ec2-2-3-218-153-43.compute-1.amazonaws.com
```

```

Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\nimes>cd Downloads

C:\Users\nimes\Downloads>ssh-add devops.pem
Identity added: devops.pem (devops.pem)

C:\Users\nimes\Downloads>ssh -N -L 3307:dev-db-instance.cvqkk8ciu30e.us-east-1.rds.amazonaws.com:3306 ec2-user@ec2-3-218-153-43.compute-1.amazonaws.com
client_loop: send disconnect: Connection reset

C:\Users\nimes\Downloads>ssh -N -L 3307:dev-db-instance.cvqkk8ciu30e.us-east-1.rds.amazonaws.com:3306 ec2-user@ec2-3-218-153-43.compute-1.amazonaws.com
client_loop: send disconnect: Connection reset

C:\Users\nimes\Downloads>ssh -N -L 3307:dev-db-instance.cvqkk8ciu30e.us-east-1.rds.amazonaws.com:3306 ec2-user@ec2-3-218-153-43.compute-1.amazonaws.com
client_loop: send disconnect: Connection reset

C:\Users\nimes\Downloads>ssh -N -L 3307:dev-db-instance.cvqkk8ciu30e.us-east-1.rds.amazonaws.com:3306 ec2-user@ec2-3-218-153-43.compute-1.amazonaws.com
client_loop: send disconnect: Connection reset

```

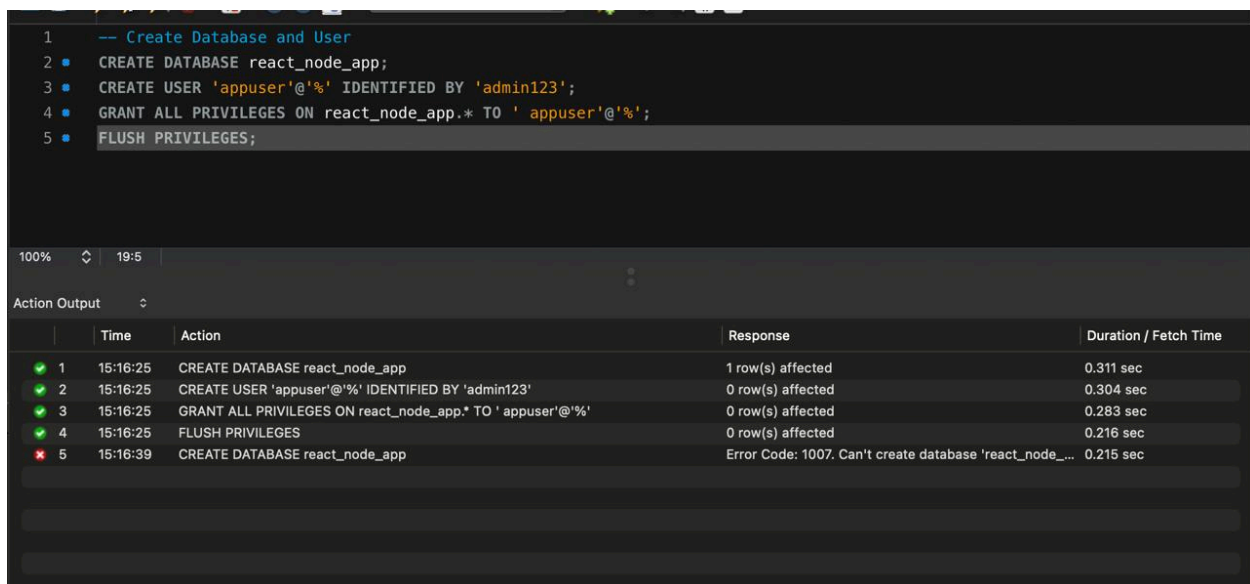

Open MySQL Workbench.

Click the "+" button and configure the connection:

- Name: rds-dev-db-admin
- Port: 3307
- Username: admin
- Store in Key: admin123

Test the connection. If successful, click OK. The connection will open in the SQL Editor.

run the below queries



The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following queries:

```
1 -- Create Database and User
2 CREATE DATABASE react_node_app;
3 CREATE USER 'appuser'@'%' IDENTIFIED BY 'admin123';
4 GRANT ALL PRIVILEGES ON react_node_app.* TO 'appuser'@'%';
5 FLUSH PRIVILEGES;
```

The Action Output pane shows the execution results:

	Time	Action	Response	Duration / Fetch Time
✓ 1	15:16:25	CREATE DATABASE react_node_app	1 row(s) affected	0.311 sec
✓ 2	15:16:25	CREATE USER 'appuser'@'%' IDENTIFIED BY 'admin123'	0 row(s) affected	0.304 sec
✓ 3	15:16:25	GRANT ALL PRIVILEGES ON react_node_app.* TO 'appuser'@'%'	0 row(s) affected	0.283 sec
✓ 4	15:16:25	FLUSH PRIVILEGES	0 row(s) affected	0.216 sec
✗ 5	15:16:39	CREATE DATABASE react_node_app	Error Code: 1007. Can't create database 'react_node_....	0.215 sec

Open MySQL Workbench again and click the "+" button. Configure the connection with these settings:

- Name: rds-dev-db-appuser
- Port: 3307
- Username: appuser
- Store in Key: admin123

Test the connection. If successful, click OK.

open SQL Editor and run the following Queries:

```
1 use react_node_app;
2 CREATE TABLE `author` (
3   `id` int NOT NULL AUTO_INCREMENT,
4   `name` varchar(255) NOT NULL,
5   `birthday` date NOT NULL,
6   `bio` text NOT NULL,
7   `createdAt` date NOT NULL,
8   `updatedAt` date NOT NULL,
9   PRIMARY KEY (`id`)
10 ) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
11
12 CREATE TABLE `book` (
13   `id` int NOT NULL AUTO_INCREMENT,
14   `title` varchar(255) NOT NULL,
15   `releaseDate` date NOT NULL,
16   `description` text NOT NULL,
17   `pages` int NOT NULL,
18   `createdAt` date NOT NULL,
19   `updatedAt` date NOT NULL,
20   `authorId` int DEFAULT NULL,
21   PRIMARY KEY (`id`),
22   KEY `FK_66a4f0f47943a0d99c16ecf90b2` (`authorId`),
23   CONSTRAINT `FK_66a4f0f47943a0d99c16ecf90b2` FOREIGN KEY (`authorId`) REFERENCES `author` (`id`)
24 ) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
25
26
27 -- Restore Data
28 INSERT INTO `author` VALUES (1,'J.K. Rowling (Joanne Kathleen Rowling)',1965-07-31,'J.K. Rowling is a British author best
```

100% 37:13

Action Output

	Time	Action	Response	Duration / Fetch Time
✖	15:18:47	CREATE TABLE `author` (`id` int NOT NULL AUTO_INCREMENT, `name` varchar...	Error Code: 1046. No database selected Select the de...	0.235 sec
✔	15:19:21	use react_node_app	0 row(s) affected	0.312 sec
✔	15:19:22	CREATE TABLE `author` (`id` int NOT NULL AUTO_INCREMENT, `name` varchar...	0 row(s) affected	0.385 sec
✔	15:19:22	CREATE TABLE `book` (`id` int NOT NULL AUTO_INCREMENT, `title` varchar(2...	0 row(s) affected	0.361 sec
✔	15:19:22	INSERT INTO `author` VALUES (1,'J.K. Rowling (Joanne Kathleen Rowling)',1965-07...	6 row(s) affected Records: 6 Duplicates: 0 Warnings...	1.005 sec
✔	15:19:23	INSERT INTO `book` VALUES (1,'Harry Potter and the Sorcerer's Stone',1997-07-2...	8 row(s) affected Records: 8 Duplicates: 0 Warnings...	0.424 sec

You'll find this code in repository
backend folder/db.sql

<https://github.com/n-nimesh/react-node-mysql-app>

Step 7: Setting Up Presentation Tier:

7.1 Creating Launch Template

Go to Launch Templates in EC2 dashboard:

- Name: presentation-tier-lt.
- Version: 01
- Enable Auto Scaling Guidance
- AMI: Amazon Linux (Quick Start)
- Instance Type: t2.micro
- Key Pair: Select the same one used for bastion host
- Security Group: Presentation-Tier-EC2
- Advanced Settings → User Data: Paste the following script

For Auto Scaling Group setup.

```
#!/bin/bash
# Update the package list and install NGINX
sudo yum update -y
sudo yum install nginx -y

# Start and enable NGINX
sudo systemctl start nginx
sudo systemctl enable nginx

# Fetch metadata token
TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

# Fetch instance details using IMDSv2
INSTANCE_ID=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" "http://169.254.169.254/latest/meta-data/instance-id")
AVAILABILITY_ZONE=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" "http://169.254.169.254/latest/meta-data/placement/availability-zone")
PUBLIC_IP=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" "http://169.254.169.254/latest/meta-data/public-ipv4")

# Create a simple HTML page displaying instance details
sudo bash -c "cat > /usr/share/nginx/html/index.html <<EOF
<h1>Instance Details</h1>
<p><b>Instance ID:</b> $INSTANCE_ID</p>
<p><b>Availability Zone:</b> $AVAILABILITY_ZONE</p>
<p><b>Public IP:</b> $PUBLIC_IP</p>
EOF"

# Restart NGINX to ensure changes are applied
sudo systemctl restart nginx
```

presentation-tier-lt (lt-0004ade5c1a89bca4)

Actions Delete template

Launch template details

Launch template ID lt-0004ade5c1a89bca4	Launch template name presentation-tier-lt	Default version 1	Owner arn:aws:iam::904233113994:root
--	--	----------------------	---

Details Versions Template tags

Launch template version details

Version: 1 (Default)

Description: version 1

Date created: 2025-04-29T09:09:43.000Z

Created by: arn:aws:iam::904233113994:root

Instance details Storage Resource tags Network interfaces Advanced details

AMI ID ami-0e449927258d45bc4	Instance type t2.micro	Availability Zone -	Key pair name devops
Security groups -	Security group IDs sg-00d2376f31fd794a0		

7.2 Creating Presentation Tier Target Group

- Navigate to Target Groups in EC2 Dashboard
- Select Instances
- Name: Presentation-Tier-TG
- VPC: Select 3-tierproject
- Health Check Path: /health
- Click Next, then Create Target Group

presentation-tier-tg

Actions

Details

arn:aws:elasticloadbalancing:us-east-1:904233113994:targetgroup/presentation-tier-tg/43e1c70b3410ed16

Target type Instance	Protocol : Port HTTP: 80	Protocol version HTTP1	VPC vpc-06365455b981b63a1
IP address type IPv4	Load balancer presentation-tier-alb		

2 Total targets	2 Healthy 0 Anomalous	0 Unhealthy	0 Unused	0 Initial	0 Draining
--------------------	-----------------------------	----------------	-------------	--------------	---------------

Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets Monitoring Health checks Attributes Tags

Registered targets (2)

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

Filter targets

	Instance ID	Name	Port	Zone	Health status	Health status details	Administrativ...	Override details	Launch time	Anomaly d
<input type="checkbox"/>	i-06ca54947ae3ba0d9	presentation	80	us-east-1a (us...	Healthy	-	No override	No override is cu...	April 29, 2025, 1...	Normal
<input type="checkbox"/>	i-03f9bdb82fbf25611	presentation	80	us-east-1b (us...	Healthy	-	No override	No override is cu...	April 29, 2025, 1...	Normal

7.3 Creating Presentation Tier Load Balancer

- Go to Load Balancer in EC2 Dashboard
- Click Create Load Balancer
- Name: Presentation-Tier-ALB
- Type: Internet Facing
- VPC: Select 3-tierproject
- Subnets: Select Public Subnets
- Security Group: Presentation-Tier-ALB
- Select the previously created Target Group
- Click Create Load Balancer.

The screenshot displays the AWS Management Console interface for a load balancer named "presentation-tier-alb". The top navigation bar includes a "Details" tab and an "Actions" dropdown. The main content area is divided into several sections:

- Details:** This section contains key information about the load balancer:
 - Load balancer type:** Application
 - Scheme:** Internet-facing
 - Status:** Active (indicated by a green checkmark)
 - Hosted zone:** Z35XDOTRQ7X7K
 - VPC:** vpc-06365455b981b63a1
 - Availability Zones:** subnet-077e96e6090d4d452 (us-east-1a (use1-az1)) and subnet-0f65993de926e8a67 (us-east-1b (use1-az2))
 - Load balancer IP address type:** IPv4
 - Date created:** April 29, 2025, 14:42 (UTC+05:30)
 - Load balancer ARN:** arn:aws:elasticloadbalancing:us-east-1:904233113994:loadbalancer/app/presentation-tier-alb/fa5fbabb8224504b
 - DNS name:** presentation-tier-alb-1983944464.us-east-1.elb.amazonaws.com (A Record)
- Listeners and rules:** This section shows the configuration for the load balancer's listeners and rules. It includes a search bar, a table of listeners, and a table of rules. The table of rules shows a single rule for HTTP-80, which forwards traffic to the target group "presentation-tier-tg" (100%) with a target group stickiness of Off.

7.4 Creating Presentation Tier Auto Scaling Group

- Navigate to Auto Scaling Group in EC2 Dashboard
- Click Create ASG
- Name: Presentation-Tier-ASG
- Select Launch Template from step 7.1
- VPC: 3-tierproject
- Subnets: Select public subnets
- Click Next
- Attach to existing Load Balancer and select "Presentation-Tier-TG"

- Enable Load Balancer health checks
- Enable CloudWatch monitoring
- Click Next
- Set capacity:
 - Desired: 3
 - Minimum: 2
 - Maximum: 4
 - Select Target Tracking Scaling Policy
 - Set Average CPU Utilization to 90%
 - Create ASG.

presentation-tier-asg

presentation-tier-asg Capacity overview [Edit](#)

[arn:aws:autoscaling:us-east-1:904233113994:autoScalingGroup:76f41796-195b-42a6-a946-b4c01749bce:autoScalingGroupName/presentation-tier-asg](#)

Desired capacity 2	Scaling limits (Min - Max) 2 - 2	Desired capacity type Units (number of instances)	Status -
------------------------------	--	---	--------------------

Date created
Tue Apr 29 2025 14:46:46 GMT+0530 (India Standard Time)

[Details](#) | [Integrations - new](#) | [Automatic scaling](#) | [Instance management](#) | [Instance refresh](#) | [Activity](#) | [Monitoring](#)

Launch template [Edit](#)

Launch template lt-0004ade5c1a89bca4 presentation-tier-lt	AMI ID ami-0e449927258d45bc4	Instance type t2.micro	Owner arn:aws:iam::904233113994:root
Version 2	Security groups -	Security group IDs sg-00d2376f31fd794a0	Create time Tue Apr 29 2025 17:02:47 GMT+0530 (India Standard Time)
Description version 2 View details in the launch template console	Storage (volumes) -	Key pair name devops	Request Spot Instances No

Network [Edit](#)

Availability Zones us-east-1a, us-east-1b	Subnet ID subnet-077e96e6090d4d452, subnet-0f65993de926e8a67	Availability Zone distribution Balanced best effort
---	--	---

Instance type requirements [Edit](#)

Your Auto Scaling group adheres to the launch template for purchase option and instance type.

[Load balancing and VPC Lattice options have moved to the new integrations tab.](#) [View integrations tab](#)

Health checks [Edit](#)

Health check type EC2, ELB	Health check grace period 300
--------------------------------------	---

Instance maintenance policy [Edit](#)

Once configured correctly, three instances will appear in the EC2 Dashboard. Access the Load Balancer DNS name to verify the Instance Metadata Details.

Step 8: Setup Application Tier:

8.1 Creating Launch Template:

- Name: Application-Tier-It
- Version: 1
- Enable Auto Scaling Guidance
- AMI: Amazon Linux
- Instance Type: T2.micro
- Key Pair: Select existing key pair
- Security Group: Application-Tier-EC2
- Advanced Settings → User Data: Copy and paste the script below with these modifications:
 - RDS Endpoint
 - Database User: "appuser"
 - Database Password: "admin123"
 - Database Name: "react_node_app"

```

#!/bin/bash
# Update package list and install required packages
sudo yum update -y
sudo yum install -y git

# Install Node.js (use NodeSource for the latest version)
curl -fsSL https://rpm.nodesource.com/setup_18.x | sudo bash -
sudo yum install -y nodejs

# Install PM2 globally
sudo npm install -g pm2

# Define variables
REPO_URL="https://github.com/suneelprojects/react-node-mysql-app.git"
BRANCH_NAME="feature/add-logging"
REPO_DIR="/home/ec2-user/react-node-mysql-app/backend"
ENV_FILE="$REPO_DIR/.env"

# Clone the repository
cd /home/ec2-user
sudo -u ec2-user git clone $REPO_URL
cd react-node-mysql-app

# Checkout to the specific branch
sudo -u ec2-user git checkout $BRANCH_NAME
cd backend

# Define the log directory and ensure it exists
LOG_DIR="/home/ec2-user/react-node-mysql-app/backend/logs"
mkdir -p $LOG_DIR
sudo chown -R ec2-user:ec2-user $LOG_DIR

# Append environment variables to the .env file
echo "LOG_DIR=$LOG_DIR" >> "$ENV_FILE"
echo "DB_HOST=\"<rds-instance.end.point.region.rds.amazonaws.com>\"" >> "$ENV_FILE"
echo "DB_PORT=\"3306\"" >> "$ENV_FILE"
echo "DB_USER=\"<db-user>\"" >> "$ENV_FILE"
echo "DB_PASSWORD=\"<db-user-password>\"" >> "$ENV_FILE" # Replace with actual password
echo "DB_NAME=\"<db-name>\"" >> "$ENV_FILE"

# Install Node.js dependencies as ec2-user
sudo -u ec2-user npm install

# Start the application using PM2 as ec2-user
sudo -u ec2-user npm run serve

# Ensure PM2 restarts on reboot as ec2-user
sudo -u ec2-user pm2 startup systemd
sudo -u ec2-user pm2 save

```


application-tier-lt (lt-03fb9c047dad0d92)

Actions Delete template

Launch template details

Launch template ID
lt-03fb9c047dad0d92

Launch template name
application-tier-lt

Default version
1

Owner
arn:aws:iam::904233113994:root

Details Versions Template tags

Launch template version details

Actions Delete template version

Version

1 (Default)

Description

version 1

Date created

2025-04-29T11:12:04.000Z

Created by

arn:aws:iam::904233113994:root

Instance details Storage Resource tags Network interfaces Advanced details

AMI ID

ami-0e449927258d45bc4

Instance type

t2.micro

Availability Zone

-

Key pair name

devops

Security groups

-

Security group IDs

sg-00abfbdabf4ea2b3e

8.2 Creating Target Group

- Click on Create Target Group → Instances
- Name: Application-Tier-TG
- Port: 3200
- VPC: 3-Tier
- Health Check Path: /health
- Click Create Target Group

application-tier-tg

Actions

Details

arn:aws:elasticloadbalancing:us-east-1:904233113994:targetgroup/application-tier-tg/3e8562afb144ab8a

Target type

Instance

Protocol : Port

HTTP: 3200

Protocol version

HTTP1

VPC

vpc-06365455b981b63a1

IP address type

IPv4

Load balancer

application-tier-alb

2

Total targets

2

Healthy

0

Unhealthy

0

Unused

0

Initial

0

Draining

0 Anomalous

► Distribution of targets by Availability Zone (AZ)

Select values in this table to see corresponding filters applied to the Registered targets table below.

Targets Monitoring Health checks Attributes Tags

Registered targets (2) info

Anomaly mitigation: Not applicable

Deregister

Register targets

Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.

<div>Filter targets</div>										<div>< 1 > ⚙</div>	
<input type="checkbox"/>	Instance ID	Name	Port	Zone	Health status	Health status details	Administrative override	Override details	Launch...		
<input type="checkbox"/>	i-0e62aa372124b18c1	application	3200	us-east-1a (us...	Healthy	-	<input type="radio"/> No override	No override is currently act...	April 29, 2...		
<input type="checkbox"/>	i-096abe4f0d1e2858c	application	3200	us-east-1b (us...	Healthy	-	<input type="radio"/> No override	No override is currently act...	April 29, 2...		

8.3 Creating Application Tier Load Balancer

- Click on Create Load Balancer
- Name: Application-Tier-ALB
- Scheme/Type: Internal
- VPC: 3-Tier-Architecture
- Subnets: Select private subnets
- Security Group: Application-Tier-ALB
- Target Group: Select Application-Tier-TG
- Click Create Load Balancer

The screenshot displays the AWS Management Console interface for an Application Load Balancer (ALB) named 'application-tier-alb'. The console shows the following details:

- Details:**
 - Load balancer type:** Application
 - Status:** Active
 - VPC:** vpc-06365455b981b63a1
 - Load balancer IP address type:** IPv4
 - Scheme:** Internal
 - Hosted zone:** Z35SXDOTRQ7X7K
 - Availability Zones:** subnet-002ca28187c1fc726 (us-east-1b (use1-az2)), subnet-018e9a56d5a6be6e4 (us-east-1a (use1-az1))
 - Date created:** April 29, 2025, 16:45 (UTC+05:30)
 - Load balancer ARN:** arn:aws:elasticloadbalancing:us-east-1:904233113994:loadbalancer/app/application-tier-alb/dd5f80af02ae022c
 - DNS name:** internal-application-tier-alb-137902212.us-east-1.elb.amazonaws.com (A Record)
- Listeners and rules:**
 - Listeners and rules (1) Info:** A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.
 - Filter listeners:** HTTP:80
 - Forward to target group:** application-tier-tg (1 (100%)), Target group stickiness: Off
 - Rules:** 1 rule
 - ARN:** Not applicable
 - Security policy:** Not applicable
 - Default SSL/TLS certificate:** Not applicable
 - mTLS:** Not applicable
 - Trust store:** Not applicable

8.4 Creating Auto Scaling Group for Application Tier

- Click on Create Auto Scaling Group
- Select Launch Template: Application-Tier-It
- VPC: 3-Tier-Architecture
- Subnets: Private
- Load Balancer: Select existing "Application-Tier-TG"
- Enable Load Balancer health checks
- Enable CloudWatch Monitoring
- Configure capacity:
 - Desired: 3
 - Minimum: 4

- Maximum: 4
 - Set Target Tracking Scaling Policy with 90% average CPU utilization
 - Click Create ASG

application-tier-asg

application-tier-asg Capacity overview

arn:aws:autoscaling:us-east-1:904233113994:autoScalingGroup:2e82b0b4-d734-4b97-8b4c-824b72d86f95:autoScalingGroupName/application-tier-asg

Desired capacity

2

Scaling limits (Min - Max)

2 - 4

Desired capacity type

Units (number of instances)

Status

-

Date created

Tue Apr 29 2025 16:47:35 GMT+0530 (India Standard Time)

Details

Integrations - new

Automatic scaling

Instance management

Instance refresh

Activity

Monitoring

Launch template

Launch template

lt-03fbf9c047dad0d92

application-tier-lt

Version

2

Description

version 2

AMI ID

ami-0e449927258d45bc4

Security groups

-

Storage (volumes)

-

Instance type

t2.micro

Security group IDs

sg-00abfbcdabf4ea2b3e

Key pair name

devops

Owner

arn:aws:iam::904233113994:root

Create time

Tue Apr 29 2025 17:51:21 GMT+0530 (India Standard Time)

Request Spot Instances

No

View details in the launch template console

Network

Availability Zones

us-east-1a, us-east-1b

Subnet ID

subnet-002ca28187c1fc726, subnet-018e9a56d5a6be6e4

Availability Zone distribution

Balanced best effort

Instance type requirements

Your Auto Scaling group adheres to the launch template for purchase option and instance type.

Load balancing and VPC Lattice options have moved to the new integrations tab.

View integrations tab

Health checks

Health check type

EC2, ELB

Health check grace period

300

Instance maintenance policy

Replacement behavior

Min healthv oercentage

Max healthv oercentage

Once configured correctly, three new instances will appear in EC2. To verify the backend is working, connect to the Bastion Host from Terminal:

```
ssh -A ec2-user@ BastionHost IP
```

Login to Private Instance of Application Tier

```
ssh ec2-user@private-ip
```

Once Connected, Run the following command

```
pm2 logs
```


- Go to Advanced → User Data → Copy and paste the below script with necessary changes

Changes to do: -Add Application Tier DNS Name -

Add Domain names `nimesh.xyz,www.nimesh.xyz`

```
#!/bin/bash
# Update package list and install required packages
sudo yum update -y
sudo yum install -y git

# Install Node.js (use NodeSource for the latest version)
curl -fsSL https://rpm.nodesource.com/setup_18.x | sudo bash -
sudo yum install -y nodejs

# Install NGINX
sudo yum install -y nginx

# Start and enable NGINX
sudo systemctl start nginx
sudo systemctl enable nginx

# Define variables
REPO_URL="https://github.com/suneelprojects/react-node-mysql-app.git"
BRANCH_NAME="feature/add-logging"
REPO_DIR="/home/ec2-user/react-node-mysql-app/frontend"
ENV_FILE="$REPO_DIR/.env"
APP_TIER_ALB_URL="http://<internal-application-tier-alb-end-point.region.elb.amazonaws.com>" # Replace with your actual alb end
API_URL="/api"

# Clone the repository as ec2-user
cd /home/ec2-user
sudo -u ec2-user git clone $REPO_URL
cd react-node-mysql-app

# Checkout to the specific branch
sudo -u ec2-user git checkout $BRANCH_NAME
cd frontend

# Ensure ec2-user owns the directory
sudo chown -R ec2-user:ec2-user /home/ec2-user/react-node-mysql-app

# Create .env file with the API_URL
echo "VITE_API_URL=\"$API_URL\" >> \"$ENV_FILE"

# Install Node.js dependencies as ec2-user
sudo -u ec2-user npm install

# Build the frontend application as ec2-user
sudo -u ec2-user npm run build

# Copy the build files to the NGINX directory
sudo cp -r dist /usr/share/nginx/html/
```

```

# Update NGINX configuration
NGINX_CONF="/etc/nginx/nginx.conf"
SERVER_NAME="<domain subdomain>" # Replace with your actual domain name

# Backup existing NGINX configuration
sudo cp $NGINX_CONF ${NGINX_CONF}.bak

# Write new NGINX configuration
sudo tee $NGINX_CONF > /dev/null <<EOL
user nginx;
worker_processes auto;

error_log /var/log/nginx/error.log warn;
pid /run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '
        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

    access_log /var/log/nginx/access.log main;

    sendfile on;
    tcp_nopush on;
    tcp_nodelay on;
    keepalive_timeout 65;
    types_hash_max_size 2048;

    include /etc/nginx/conf.d/*.conf;
}
EOL

```

```

# Create a separate NGINX configuration file
sudo tee /etc/nginx/conf.d/presentation-tier.conf > /dev/null <<EOL
server {
    listen 80;
    server_name $SERVER_NAME;
    root /usr/share/nginx/html/dist;
    index index.html index.htm;

    #health check
    location /health {
        default_type text/html;
        return 200 "<!DOCTYPE html><p>Health check endpoint</p>\n";
    }

    location / {
        try_files $uri /index.html;
    }

    location /api/ {
        proxy_pass $APP_TIER_ALB_URL;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header X-Forwarded-Proto $scheme;
    }
}
EOL

# Restart NGINX to apply the new configuration
sudo systemctl restart nginx

```

- Click on Update

9.2 Modifying Auto Scaling Group of Presentation Tier

- Click on Edit
- Desired: 2
- Min: 2
- Max: 2
- Select Launch Template Version: 2
- click on update.

presentation-tier-asg

presentation-tier-asg Capacity overview [Edit](#)

[arn:aws:autoscaling:us-east-1:904233113994:autoScalingGroup:76f41796-195b-42a6-a946-b4c01749bce0:autoScalingGroupName/presentation-tier-asg](#)

Desired capacity 2	Scaling limits (Min - Max) 2 - 2	Desired capacity type Units (number of instances)	Status -
------------------------------	--	---	--------------------

Date created
Tue Apr 29 2025 14:46:46 GMT+0530 (India Standard Time)

[Details](#) | [Integrations - new](#) | [Automatic scaling](#) | [Instance management](#) | [Instance refresh](#) | [Activity](#) | [Monitoring](#)

Launch template [Edit](#)

Launch template
[lt-0004ade5c1a89bca4](#)
presentation-tier-lt

Version
2

Description
version 2

[View details in the launch template console](#)

AMI ID
[ami-0e449927258d45bc4](#)

Security groups
-

Storage (volumes)
-

Instance type
t2.micro

Security group IDs
[sg-00d2376f31fd794a0](#)

Key pair name
devops

Owner
arn:aws:iam::904233113994:root

Create time
Tue Apr 29 2025 17:02:47 GMT+0530 (India Standard Time)

Request Spot Instances
No

Network [Edit](#)

Availability Zones
us-east-1a, us-east-1b

Subnet ID
subnet-077e96e6090d4d452, subnet-0f65993de926e8a67

Availability Zone distribution
Balanced best effort

Instance type requirements [Edit](#)

Your Auto Scaling group adheres to the launch template for purchase option and instance type.

[Load balancing and VPC Lattice options have moved to the new integrations tab.](#) [View integrations tab](#)

Health checks [Edit](#)

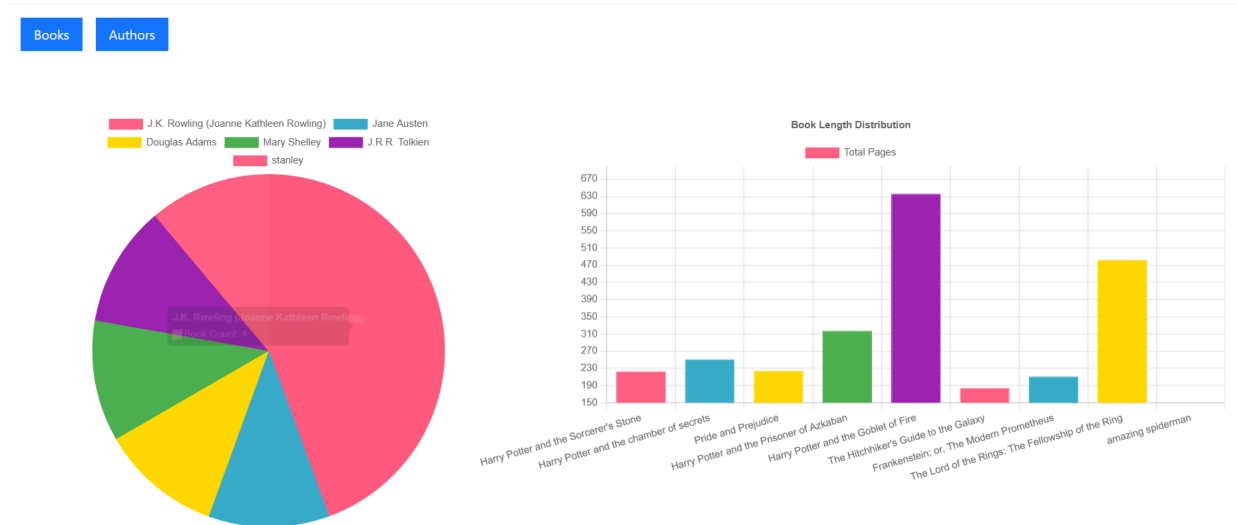
Health check type
EC2, ELB

Health check grace period
300

Instance maintenance policy [Edit](#)

- Terminate Old Instances to get new instances with new configuration.

Now You can Access the Application using DNS Name of Presentation Tier Load Balancer.



Step 10: Integrating Application Logs with CloudWatch

Connect one of our App Tier Ec2 instances

```
ls
```

Go to Repo Directory/bakcend/logs

```
vim combined.log
```



```
Command Prompt - ssh -N -L x ec2-user@ip-10-0-139-166:~ x ec2-user@ip-10-0-139-166:~/I

[ec2-user@ip-10-0-139-166 ~]$ ll
total 0
drwxr-xr-x. 5 ec2-user ec2-user 66 Apr 29 11:19 react-node-mysql-app
[ec2-user@ip-10-0-139-166 ~]$ cd react-node-mysql-app
[ec2-user@ip-10-0-139-166 react-node-mysql-app]$ ll
total 44
-rw-r--r--. 1 ec2-user ec2-user 11346 Apr 29 11:19 README.md
drwxr-xr-x. 8 ec2-user ec2-user 16384 Apr 29 11:19 backend
drwxr-xr-x. 4 ec2-user ec2-user 16384 Apr 29 11:19 frontend
[ec2-user@ip-10-0-139-166 react-node-mysql-app]$ cd backend
[ec2-user@ip-10-0-139-166 backend]$ ll
total 76
-rw-r--r--. 1 ec2-user ec2-user 709 Apr 29 11:19 app.js
drwxr-xr-x. 2 ec2-user ec2-user 19 Apr 29 11:19 configs
drwxr-xr-x. 2 ec2-user ec2-user 60 Apr 29 11:19 controllers
-rw-r--r--. 1 ec2-user ec2-user 6485 Apr 29 11:19 db.sql
drwxr-xr-x. 2 ec2-user ec2-user 43 Apr 29 11:19 logs
drwxr-xr-x. 104 ec2-user ec2-user 16384 Apr 29 11:19 node_modules
-rw-r--r--. 1 ec2-user ec2-user 40143 Apr 29 11:19 package-lock.json
-rw-r--r--. 1 ec2-user ec2-user 491 Apr 29 11:19 package.json
drwxr-xr-x. 2 ec2-user ec2-user 22 Apr 29 11:19 routes
-rw-r--r--. 1 ec2-user ec2-user 167 Apr 29 11:19 server.js
drwxr-xr-x. 2 ec2-user ec2-user 23 Apr 29 11:19 utils
[ec2-user@ip-10-0-139-166 backend]$ cd logs
[ec2-user@ip-10-0-139-166 logs]$ ll
total 8
-rw-r--r--. 1 ec2-user ec2-user 1259 Apr 29 12:03 combined.log
-rw-r--r--. 1 ec2-user ec2-user 188 Apr 29 11:44 error.log
[ec2-user@ip-10-0-139-166 logs]$ vim combined.log
[ec2-user@ip-10-0-139-166 logs]$ |
```

You can See logs

10.1 Creating IAM Role

- Creating IAM Role for EC2 CloudWatch Logs
- Navigate to **IAM Dashboard** → **Roles**.
- Click **Create Role** → Select **EC2**.

- Click **Create**.

ec2instance-rolefor-cloudwatchlogs info

Allows EC2 instances to call AWS services on your behalf.

Summary

- Creation date:** April 29, 2025, 17:44 (UTC+05:30)
- ARN:** arn:aws:iam::904233113994:role/ec2instance-rolefor-cloudwatchlogs
- Instance profile ARN:** arn:aws:iam::904233113994:instance-profile/ec2instance-rolefor-cloudwatchlogs
- Last activity:** 10 minutes ago
- Maximum session duration:** 1 hour

Permissions | Trust relationships | Tags | Last Accessed | Revoke sessions

Permissions policies (2) info

You can attach up to 10 managed policies.

Policy name	Type	Attached entities
CloudWatchAgentServerPolicy	AWS managed	1
CloudWatchLogsFullAccess	AWS managed	1

► **Permissions boundary** (not set)

10.2 Creating CloudWatch Log Group

- Navigate to **CloudWatch** → **Log Groups**.
- Click **Create Log Group**.
- Enter **Name:** `backend-node-app-logs`.
- Click **Create**.

CloudWatch > Log groups > backend-node-app-logs

backend-node-app-logs

Log group details

- Log class:** Standard
- ARN:** arn:aws:logsus-east-1:904233113994:log-group:backend-node-app-logs*
- Creation time:** 45 minutes ago
- Retention:** Never expire
- Stored bytes:** -
- Metric filters:** 0
- Subscription filters:** 0
- Contributor Insights rules:** -
- KMS key ID:** -
- Anomaly detection:** Configure
- Data protection:** -
- Sensitive data count:** -
- Field indexes:** Configure
- Transformer:** Configure

Log streams (2)

Log stream	Last event time
i-096abe4f0d1e2858c	2025-04-29 18:07:09 (UTC+05:30)
i-0e62aa372124b18c1	2025-04-29 18:05:22 (UTC+05:30)

Step11:Modifying Application Tier Launch Template

- Select Application-Tier-It
- Select Version 2
- Go to Advanced Details
- Select IAM Instance Profile
- Attach Role We created
- Enable Detailed CloudWatch Monitoring
- User Data → Add the Below Script

```

#!/bin/bash
# Update package list and install required packages
sudo yum update -y
sudo yum install -y git

# Install Node.js (use NodeSource for the latest version)
curl -fsSL https://rpm.nodesource.com/setup_18.x | sudo bash -
sudo yum install -y nodejs

# Install PM2 globally
sudo npm install -g pm2

# Define variables
REPO_URL="https://github.com/suneelprojects/react-node-mysql-app.git"
BRANCH_NAME="feature/add-logging"
REPO_DIR="/home/ec2-user/react-node-mysql-app/backend"
ENV_FILE="$REPO_DIR/.env"

# Clone the repository
cd /home/ec2-user
sudo -u ec2-user git clone $REPO_URL
cd react-node-mysql-app

# Checkout to the specific branch
sudo -u ec2-user git checkout $BRANCH_NAME
cd backend

# Define the log directory and ensure it exists
LOG_DIR="/home/ec2-user/react-node-mysql-app/backend/logs"
mkdir -p $LOG_DIR
sudo chown -R ec2-user:ec2-user $LOG_DIR

# Append environment variables to the .env file
echo "LOG_DIR=$LOG_DIR" >> "$ENV_FILE"
echo "DB_HOST=\"<rds-instance.end.point.region.rds.amazonaws.com>\"" >> "$ENV_FILE"
echo "DB_PORT=\"3306\"" >> "$ENV_FILE"
echo "DB_USER=\"<db-user>\"" >> "$ENV_FILE"
echo "DB_PASSWORD=\"<db-user-password>\"" >> "$ENV_FILE" # Replace with actual password
echo "DB_NAME=\"<db-name>\"" >> "$ENV_FILE"

# Install Node.js dependencies as ec2-user
sudo -u ec2-user npm install

# Start the application using PM2 as ec2-user
sudo -u ec2-user npm run serve

# Ensure PM2 restarts on reboot as ec2-user
sudo -u ec2-user pm2 startup systemd
sudo -u ec2-user pm2 save

```

```
# Restart NGINX to apply the new configuration
sudo systemctl restart nginx

# Install CloudWatch agent
sudo yum install -y amazon-cloudwatch-agent

# Create CloudWatch agent configuration
sudo tee /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json >
/dev/null <<EOL
{
  "logs": {
    "logs_collected": {
      "files": {
        "collect_list": [
          {
            "file_path": "/home/ec2-user/react-node-mysql-app/backend/logs/*.log",
            "log_group_name": "backend-node-app-logs",
            "log_stream_name": "{instance_id}",
            "timestamp_format": "%Y-%m-%d %H:%M:%S"
          }
        ]
      }
    }
  }
}
EOL
```

```
# Start CloudWatch agent
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a
fetch-config -m ec2 -c
file:/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json -s
```

11.1 Modifying ASG

- Click on Application-Tier-ASG
- Edit - Version 2 - Update
To see changes, Terminate the Old Instances, New instances will get created

Access the Application → Add Book → Go to Cloud
Watch Log Group → Check Logs

CloudWatch

Log groups

backend-node-app-logs

i-062aa372124b18c1

CloudWatch

Log groups

backend-node-app-logs

i-062aa372124b18c1

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Clear

1m

30m

1h

12h

Custom

Local timezone

Display

Timestamp

Message

2025-04-29T18:06:22.000+05:30

2025-04-29 12:36:22 [INFO]: AuthorController [GET]

2025-04-29T18:06:22.000+05:30

2025-04-29 12:36:22 [INFO]: Authors count: 8

2025-04-29T18:07:07.000+05:30

2025-04-29 12:37:07 [INFO]: BooksController [GET]

2025-04-29T18:07:07.000+05:30

2025-04-29 12:37:07 [INFO]: Books count: 9

2025-04-29T18:07:09.000+05:30

2025-04-29 12:37:09 [INFO]: AuthorController [GET]

2025-04-29T18:07:09.000+05:30

2025-04-29 12:37:09 [INFO]: Authors count: 8

2025-04-29T18:15:27.000+05:30

2025-04-29 12:45:27 [INFO]: BooksController [GET]

2025-04-29T18:15:27.000+05:30

2025-04-29 12:45:27 [INFO]: Books count: 9

2025-04-29T18:15:33.000+05:30

2025-04-29 12:45:33 [INFO]: AuthorController [GET]

2025-04-29T18:15:33.000+05:30

2025-04-29 12:45:33 [INFO]: Authors count: 8

2025-04-29T18:27:26.000+05:30

2025-04-29 12:57:26 [INFO]: BooksController [GET]

2025-04-29T18:27:26.000+05:30

2025-04-29 12:57:26 [INFO]: Books count: 9

2025-04-29T18:27:30.000+05:30

2025-04-29 12:57:30 [INFO]: AuthorController [GET]

2025-04-29T18:27:30.000+05:30

2025-04-29 12:57:30 [INFO]: Authors count: 8

2025-04-29T18:27:57.000+05:30

2025-04-29 12:57:57 [INFO]: BooksController [CREATE] - title: amazing spiderman-2, description: spiderman, releaseDate: 1999-10-10, pages: 180, authorId: 8

2025-04-29T18:27:57.000+05:30

2025-04-29 12:57:57 [INFO]: Book created successfully. books count: 10

2025-04-29T18:28:05.000+05:30

2025-04-29 12:58:05 [INFO]: BooksController [GET]

2025-04-29T18:28:05.000+05:30

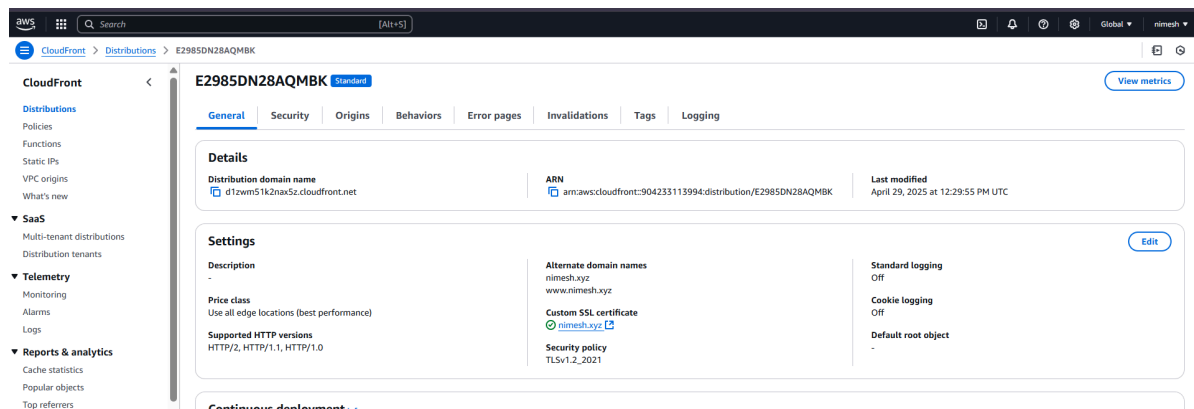
2025-04-29 12:58:05 [INFO]: Books count: 10

No more events at this moment. Auto refresh enabled. [Refresh](#)

Step 12: Creating CloudFront Distribution

Go to CloudFront

- Create → Origin Domain Presentation-Tier-ALB → HTTP Redirect HTTP to HTTPS
- Do not Enable WAF → Select needed regions Add Alternate Domain name → "nimesh.xyz" "www.nimesh.xyz"
- Attach SSL Certificate
Create Distribution



Step 13: Create DNS Records from CloudFront in Route53

Go to Route53

- Open Our Hosted Zone → Click on Add Record → Alias → To Cloud Front Distribution
Select your distribution. Do the Same for www.devopsdost.com also.

Book Details	
ID	10
TITLE	amazing spiderman
RELEASE DATE	1952-01-12T00:00:00.000Z
BOOK DESCRIPTION	marvel
TOTAL PAGES	115
CREATED DATE	2025-04-29T00:00:00.000Z
UPDATED DATE	2025-04-29T00:00:00.000Z
	8
	stanley
BIRTHDAY	1921-01-01T00:00:00.000Z
AUTHOR DESCRIPTION	spiderman,ironman,captain america,thor,hulk,deadpool,wolverine
AUTHOR	stanley

Dashboard		MANAGE AUTHORS			
Add Author		Add Author			
		Name			
		[Name] 1			
		Birthday			
		[Birthday]			
		Description			
		[Description]			
		Cancel Add			
ID	Author	Birthday	Description	Updated Date	Actions
1	J.K. Rowling	1963-07-01	J.K. Rowling is a Scottish author and screenwriter. She is best known for her series of books about the wizarding world, Harry Potter.	2024-05-29T00:00:00.000Z	Edit Delete
2	Jane Austen	1775-12-16	Jane Austen was an English novelist, writer, and letter-writer. She is best known for her novels, Pride and Prejudice and Emma.	2024-05-29T00:00:00.000Z	Edit Delete
3	Jane Austen	1775-12-16	Jane Austen was an English novelist, writer, and letter-writer. She is best known for her novels, Pride and Prejudice and Emma.	2024-05-29T00:00:00.000Z	Edit Delete
4	Hogwarts	1990-07-11	Hogwarts is a fictional school of magic in the Harry Potter series. It is located in Scotland and is the largest and most famous of the wizarding schools.	2024-05-29T00:00:00.000Z	Edit Delete
5	J.R.R. Tolkien	1894-01-03	J.R.R. Tolkien was an English writer, philologist, and professor. He is best known for his epic high fantasy novel, The Lord of the Rings.	2024-05-29T00:00:00.000Z	Edit Delete
6	Mary Shelley	1797-01-30	Mary Shelley was a British novelist, playwright, and short story writer. She is best known for her novel, Frankenstein; or, the Modern Prometheus.	2024-05-29T00:00:00.000Z	Edit Delete
			Stephen Adams was an English screenwriter, producer, director, and actor. He is best known for his work on the television series, Doctor Who.	2024-05-29T00:00:00.000Z	Edit Delete