PARSHVANATH CHARITABLE TRUST'S

A. P. Shah Institute of Technology

Thane, 400615

Academic Year: 2023-24

Department of Computer Engineering

CSL605 SKILL BASED LAB COURSE: CLOUD COMPUTING

Mini Project Report

- **Title of Project** : Townhall
- **Year and Semester** : T.E , Sem VI
- **Group Members Name and Roll No.** : Atharv Vinay Agharkar – 21102154
  Niraj Anant Bade – 21102188
  Lalit Yoginath Bagul - 21102048

Table of Contents

Problem Definition

The project aims to deploy the "Townhall" website onto the Amazon Web Services (AWS) platform using Elastic Beanstalk and establish a connection with an Amazon Relational Database Service (Amazon RDS) instance. The website contains functionalities such as user registration, login, feedback submission, and ratings. The primary objectives include:

1) Deploying the "Townhall" website on Elastic Beanstalk to ensure scalability and reliability.
2) Setting up an Amazon RDS instance to store user data, feedback, and ratings securely.
3) Configuring security groups to control inbound and outbound traffic between the Elastic Beanstalk environment and the Amazon RDS instance.
4) Establishing a seamless connection between the website deployed on Elastic Beanstalk and the Amazon RDS database to enable CRUD (Create, Read, Update, Delete) operations.
5) Implementing CRUD functionalities within the website to facilitate user interactions and data management effectively.
6) The project seeks to leverage AWS cloud services to build a robust and scalable web application infrastructure while ensuring data integrity, security, and seamless user experience.
7) By deploying the "Townhall" website on AWS Elastic Beanstalk and integrating it with Amazon RDS, the project aims to demonstrate best practices in cloud-based application development and database management.

Introduction

In today's digital age, the demand for scalable, reliable, and secure web applications is ever-growing. As businesses and organizations strive to provide seamless online experiences to their users, leveraging cloud computing platforms has become essential. Amazon Web Services (AWS) stands at the forefront of cloud services, offering a wide array of tools and services to build, deploy, and manage applications with ease. In this context, the project focuses on deploying the "Townhall" website onto AWS using Elastic Beanstalk and integrating it with Amazon RDS, thereby harnessing the power of cloud computing to create a robust and efficient web application ecosystem.

The "Townhall" website serves as a platform for users to engage in various community-driven activities such as exploring various active clubs of the college, finding details of such clubs , participating in interested ones by registering, submitting feedbacks, and rating various clubs as per user’s experience. With features like user registration, login authentication, and data storage, the website aims to foster a collaborative online environment where users can interact, share information as well as ideas, and provide valuable feedback. Elastic Beanstalk, a Platform as a Service (PaaS) offering from AWS, provides an ideal solution for deploying and managing web applications without the complexity of infrastructure management. By abstracting away the underlying infrastructure details, Elastic Beanstalk allows developers to focus on writing code and building features, while AWS handles the deployment, scaling, and monitoring aspects seamlessly. This project leverages Elastic Beanstalk to deploy the "Townhall" website, ensuring scalability, fault tolerance, and ease of management.

In conjunction with Elastic Beanstalk, the project utilizes Amazon RDS to set up a relational database for storing user data, feedback submissions, and ratings. Amazon RDS offers a fully managed database service, eliminating the need for manual database administration tasks such as provisioning, patching, and backups. By leveraging Amazon RDS, the project ensures data integrity, security, and high availability, enabling smooth operation of the "Townhall" website. The integration between Elastic Beanstalk and Amazon RDS enables seamless communication between the web application and the database, facilitating efficient data retrieval, storage, and manipulation. With the ability to perform CRUD operations, the "Townhall" website can effectively manage user interactions, store user-generated content, and provide personalized experiences to its users. Overall, the project demonstrates the power of cloud computing in building scalable, reliable, and feature-rich web applications. By harnessing the capabilities of AWS Elastic Beanstalk and Amazon RDS, the "Townhall" website is poised to deliver a seamless and engaging user experience while leveraging the benefits of cloud-based infrastructure.

Description

The deployment of the "Townhall" website onto AWS using Elastic Beanstalk and integration with Amazon RDS involves several intricate steps aimed at creating a robust, scalable, and secure web application ecosystem. Below is a detailed description of each phase of the project:

1. Setting up Elastic Beanstalk Environment:

The first step involves creating an Elastic Beanstalk environment to host the "Townhall" website. Using the AWS Management Console, developers select the appropriate region, platform, and environment type (e.g., web server environment). They configure environment settings such as instance type, auto-scaling options, and environment variables. Additionally, developers can customize environment options, including load balancer settings, monitoring options, and logging configurations.

2. Deploying the "Townhall" Website:

Once the Elastic Beanstalk environment is set up, developers deploy the "Townhall" website onto the environment. They can choose to upload the application code directly to Elastic Beanstalk or deploy from a version-controlled repository such as GitHub. Elastic Beanstalk handles the deployment process, including provisioning EC2 instances, configuring load balancers, and setting up auto-scaling policies. Developers monitor the deployment progress through the Elastic Beanstalk dashboard, ensuring successful deployment and availability of the website.

3. Configuring Amazon RDS:

Simultaneously, developers set up an Amazon RDS instance to serve as the relational database for the "Townhall" website. They navigate to the Amazon RDS console and select the desired database engine (e.g., MySQL, PostgreSQL). Developers specify configuration details such as database instance size, storage type, and backup options. Security measures, including IAM roles, encryption, and parameter groups, are configured to enhance data protection. Once the RDS instance is provisioned, developers note down the database endpoint, username, and password for later use.

4. Establishing Security Groups:

Security groups are configured to control inbound and outbound traffic between the Elastic Beanstalk environment and the Amazon RDS instance. Developers create separate security groups for the Elastic Beanstalk environment and the RDS instance, applying appropriate firewall rules to restrict access to specific ports and IP addresses. They ensure that only necessary ports (e.g., HTTP, HTTPS, MySQL) are open to minimize security risks and prevent unauthorized access to the infrastructure.

5. Connecting Elastic Beanstalk to Amazon RDS:

To enable communication between the "Townhall" website deployed on Elastic Beanstalk and the Amazon RDS database, developers modify the website's configuration files to include the database connection details. They update the database connection settings in the application code to use the RDS endpoint, username, password, and database name. This ensures seamless connectivity between the web server and the database server, allowing the website to perform CRUD operations and retrieve/store data securely.

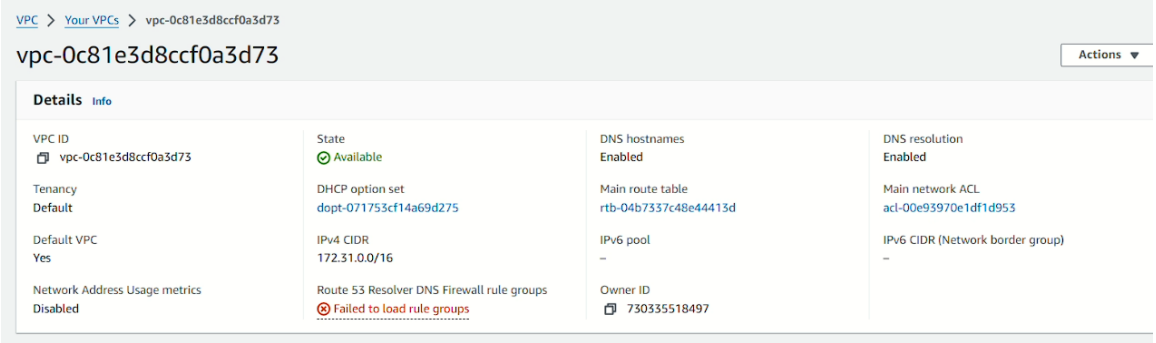
6. Implementing CRUD Operations:

With the infrastructure set up and the connectivity established, developers proceed to implement CRUD (Create, Read, Update, Delete) operations within the "Townhall" website. Backend APIs or server-side scripts are developed to handle user registration, authentication, feedback submission, and rating functionalities. Frontend interfaces are integrated with the backend APIs to facilitate user interactions and data management effectively. Developers thoroughly test the CRUD functionalities to ensure proper data storage, retrieval, update, and deletion, thereby delivering a seamless user experience.

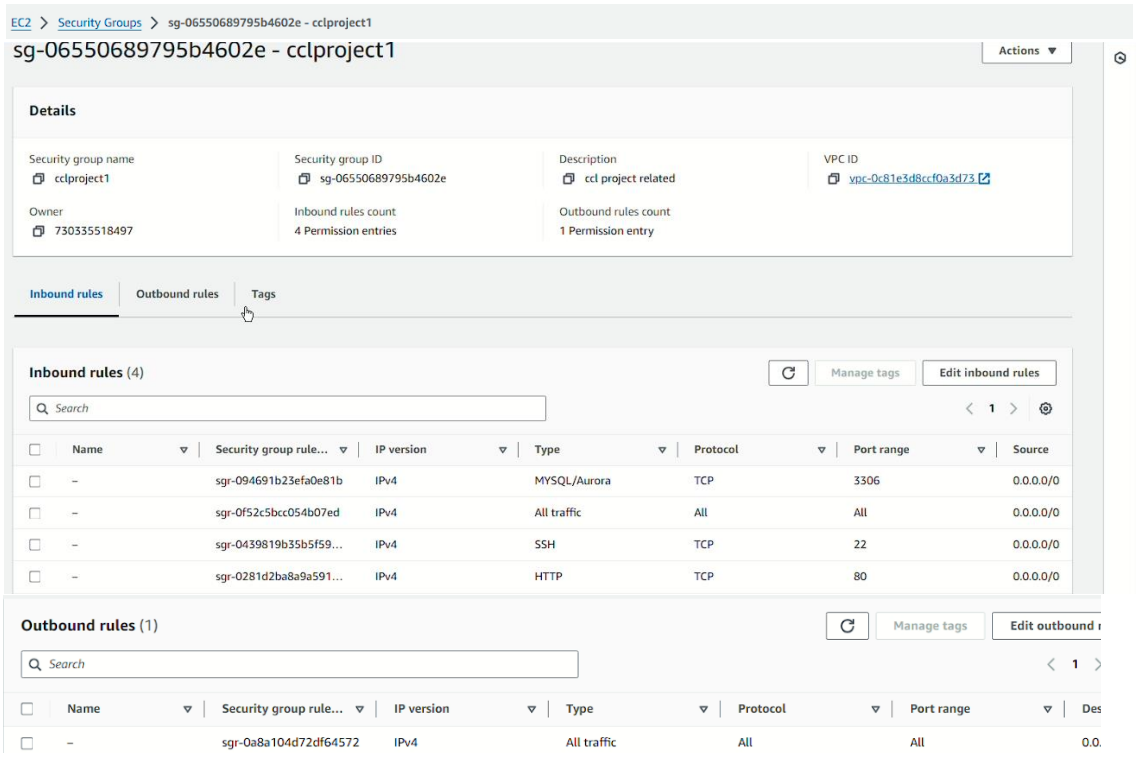
Through meticulous planning, configuration, and implementation, the "Townhall" website is successfully deployed onto AWS using Elastic Beanstalk and seamlessly integrated with Amazon RDS. The resulting web application ecosystem is scalable, reliable, and secure, capable of handling user interactions, storing data securely, and delivering personalized experiences to its users.

Implementation Details

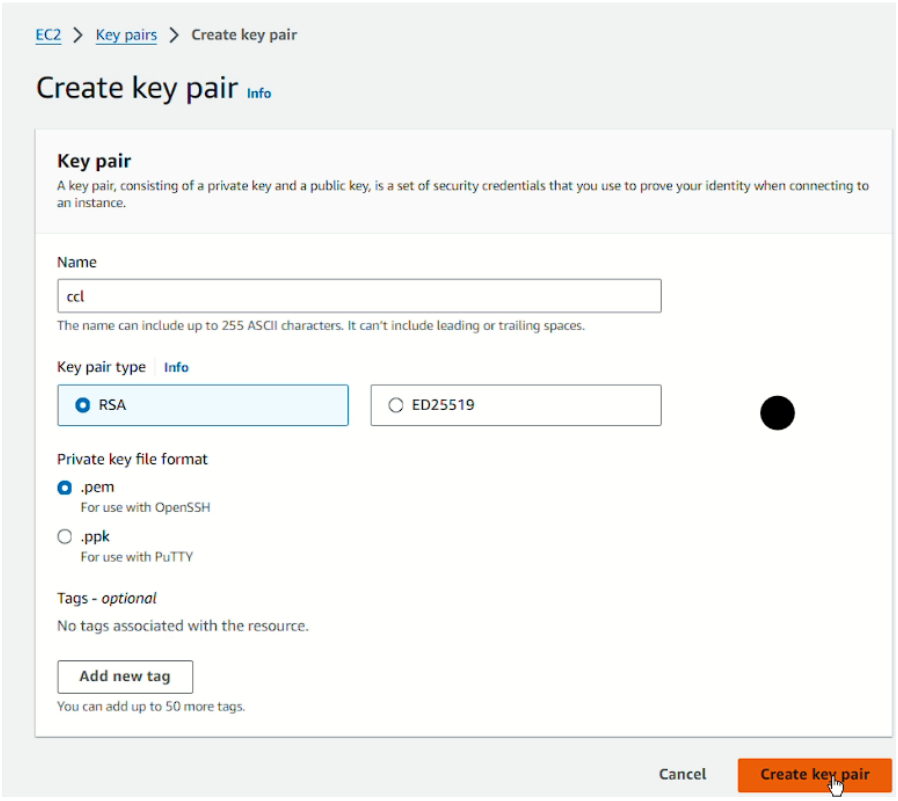
Virtual Private Cloud (VPC):



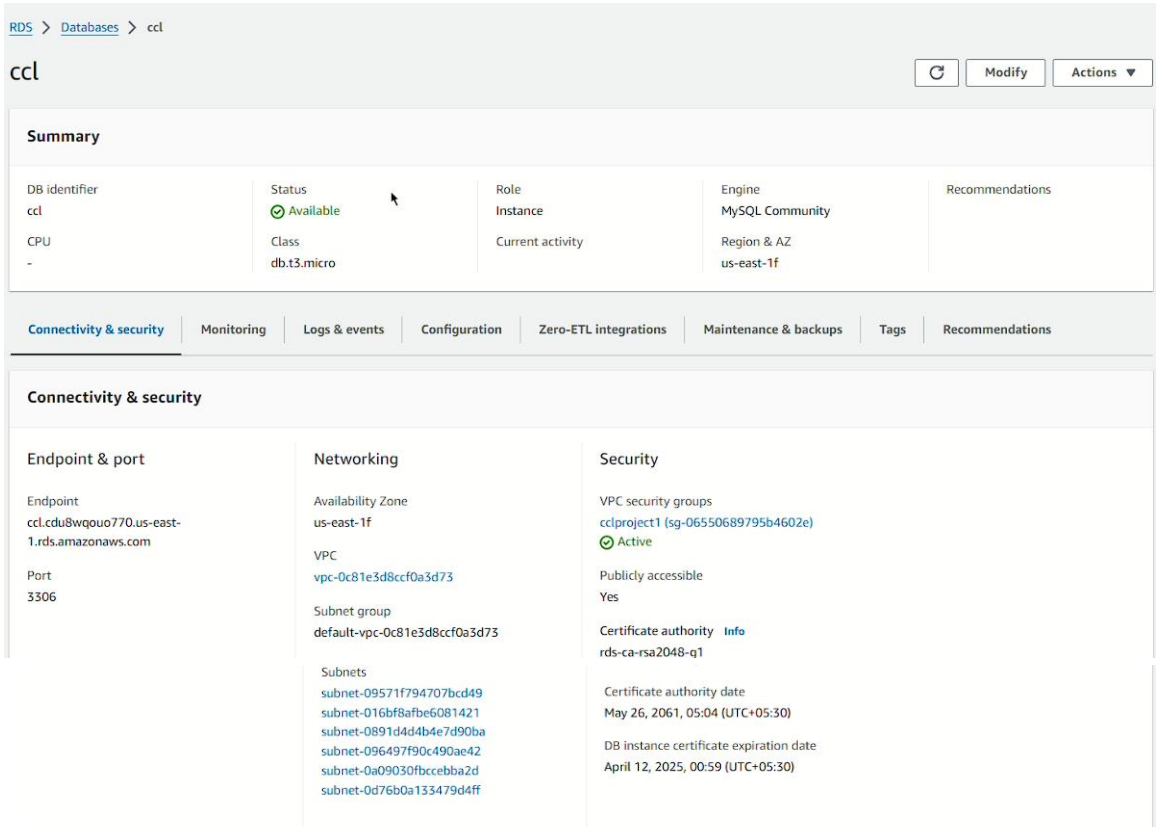
Security Group:



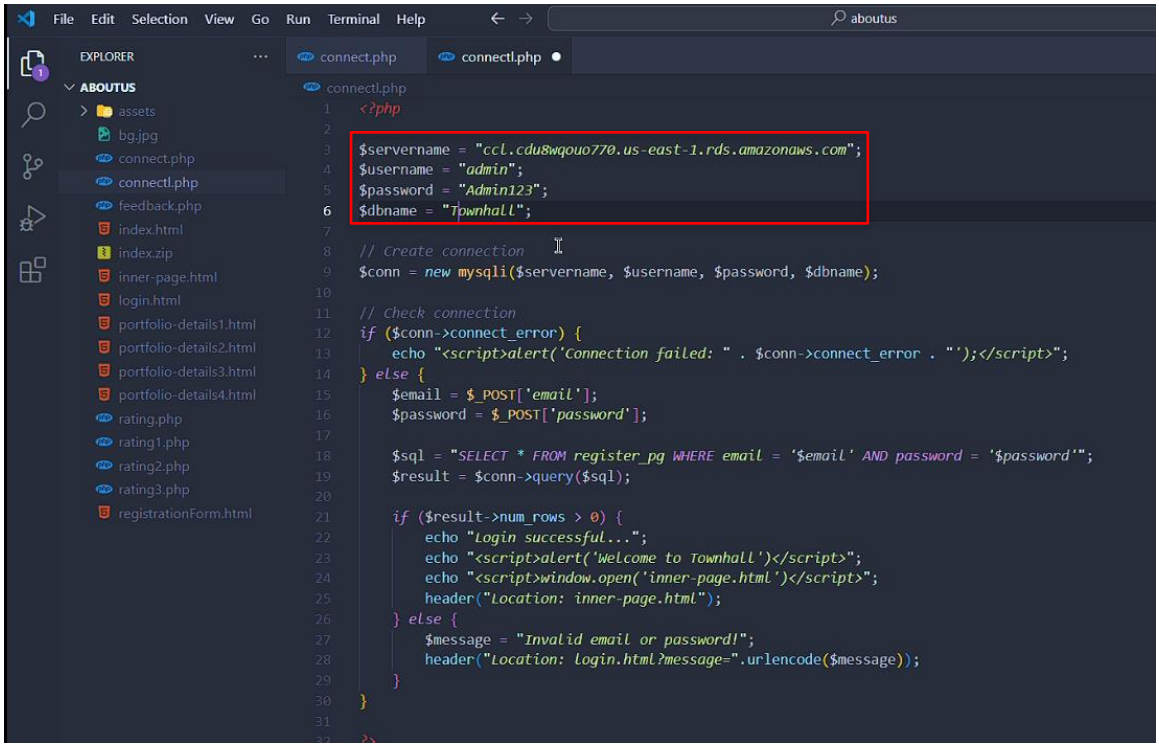
Key Pair:



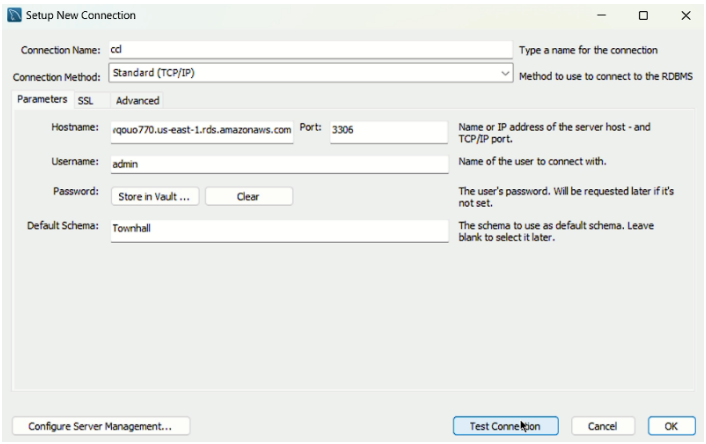
Relational Database Service (RDS):

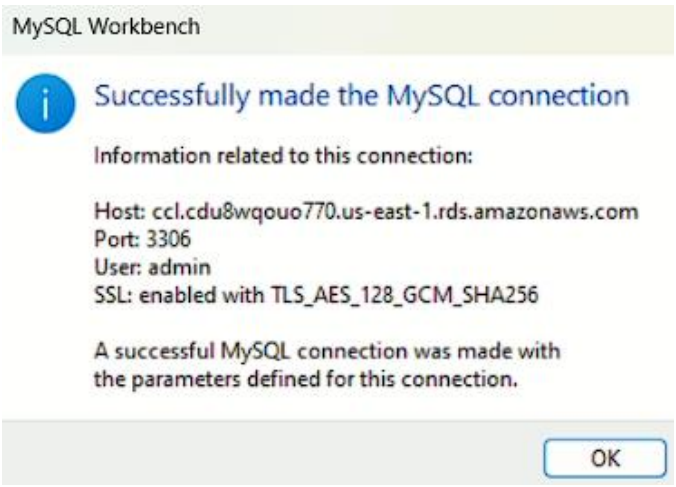


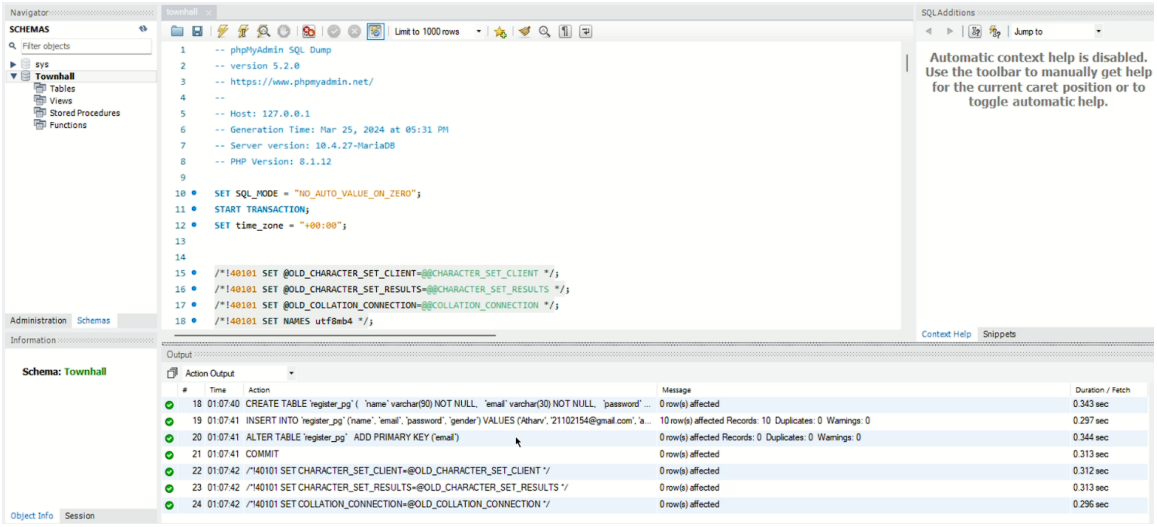
PHP code containing the endpoint given by RDS**:**

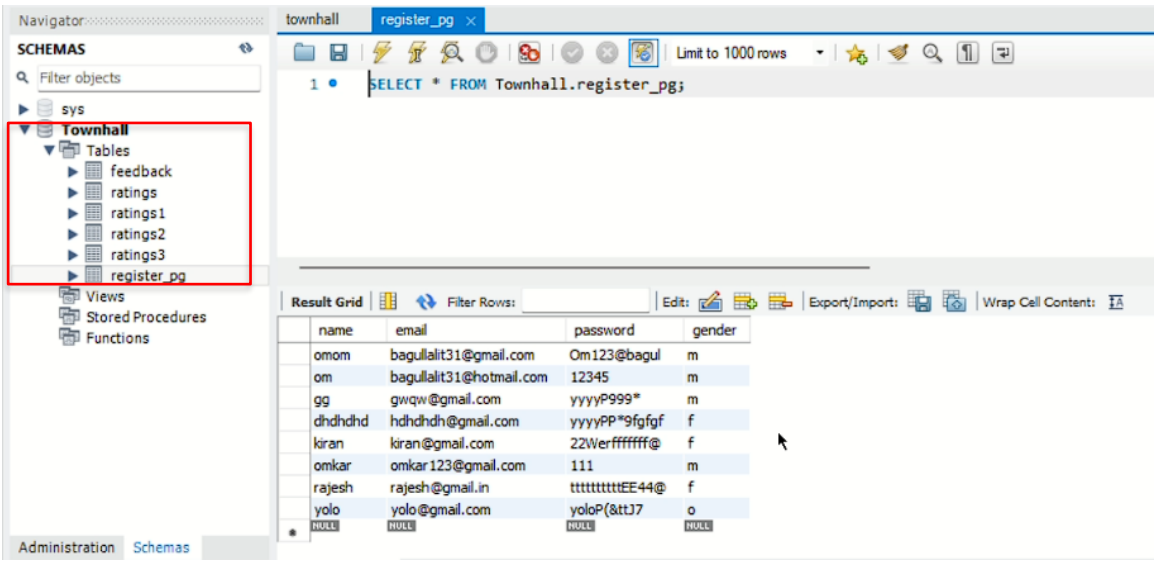


Creating connection with workbench to import the database schema into RDS:

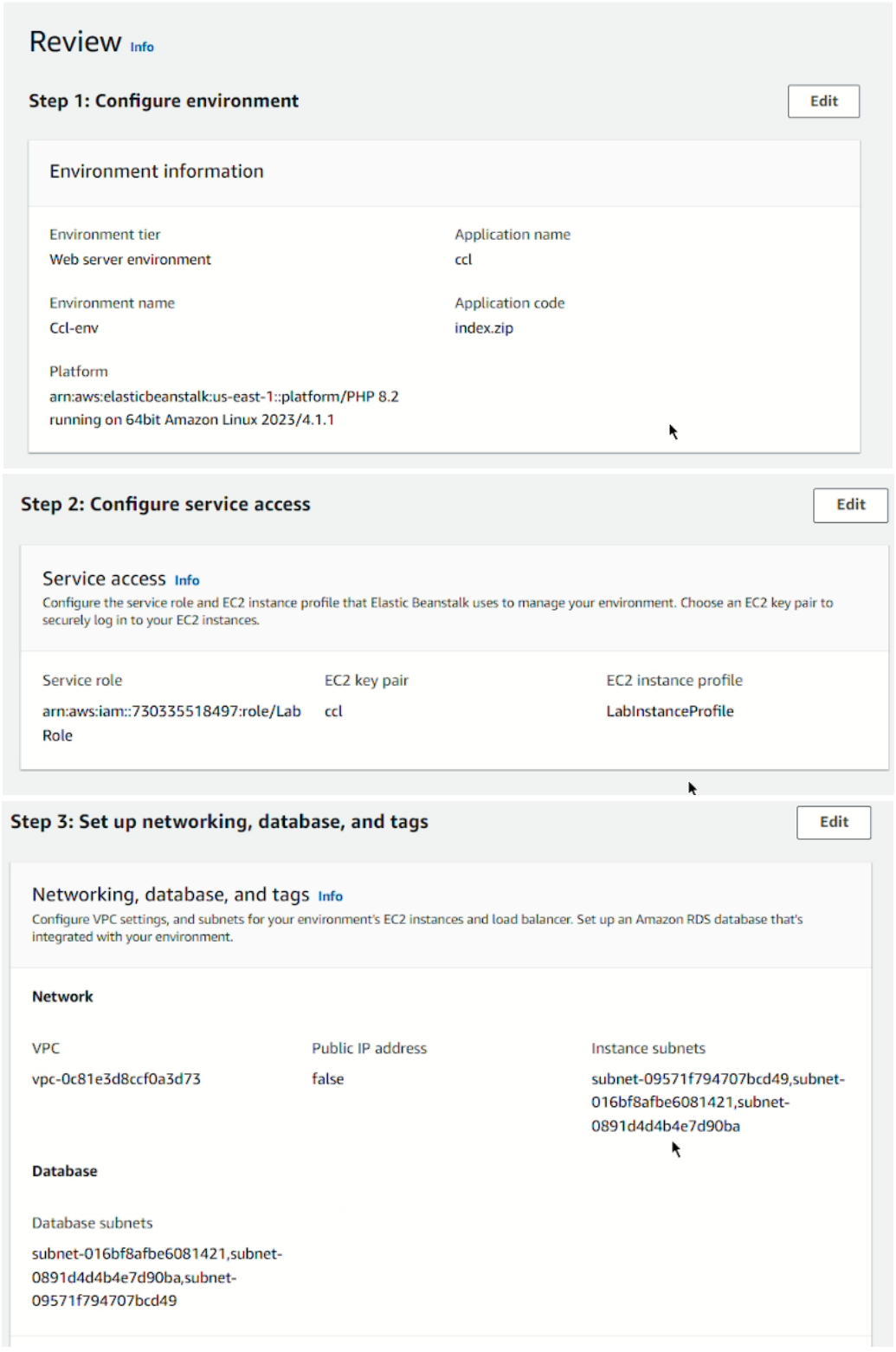



MySQL Workbench
Successfully made the MySQL connection
Information related to this connection:
Host: ccl.cdu8wqouo770.us-east-1.rds.amazonaws.com
Port: 3306
User: admin
SSL: enabled with TLS_AES_128_GCM_SHA256
A successful MySQL connection was made with the parameters defined for this connection.
OK


Navigator
SCHEMAS
Filter objects
sys
Townhall
Tables
Views
Stored Procedures
Functions
townhall
Limit to 1000 rows
-- phpMyAdmin SQL Dump
-- version 5.2.0
-- https://www.phpmyadmin.net/
-- Host: 127.0.0.1
-- Generation Time: Mar 25, 2024 at 05:31 PM
-- Server version: 10.4.27-MariaDB
-- PHP Version: 8.1.12
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";
/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help
Snippets
Administration
Schemas
Information
Schema: Townhall
Object Info
Session
Output
Action Output
Time Action Message Duration / Fetch
18 01:07:40 CREATE TABLE `register_pg` (`name` varchar(90) NOT NULL, `email` varchar(30) NOT NULL, `password` ... 0 row(s) affected 0.343 sec
19 01:07:41 INSERT INTO `register_pg` (`name`, `email`, `password`, `gender`) VALUES ('Atharv', '21102154@gmail.com', 'a... 10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0 0.297 sec
20 01:07:41 ALTER TABLE `register_pg` ADD PRIMARY KEY (`email`) 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.344 sec
21 01:07:41 COMMIT 0 row(s) affected 0.313 sec
22 01:07:42 /*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */ 0 row(s) affected 0.312 sec
23 01:07:42 /*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */ 0 row(s) affected 0.313 sec
24 01:07:42 /*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */ 0 row(s) affected 0.296 sec


Navigator
SCHEMAS
Filter objects
sys
Townhall
Tables
feedback
ratings
ratings1
ratings2
ratings3
register_pg
Views
Stored Procedures
Functions
Administration
Schemas
townhall
register_pg
Limit to 1000 rows
1 SELECT * FROM Townhall.register_pg;
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
name email password gender
omom bagullalit31@gmail.com Om123@bagul m
om bagullalit31@hotmail.com 12345 m
gg gwqw@gmail.com yyyyP999* m
dhdhdhd hdhdhdh@gmail.com yyyyPP*9fgfgf f
kiran kiran@gmail.com 22Werfffffff@ f
omkar omkar123@gmail.com 111 m
rajesh rajesh@gmail.in ttttttttttEE44@ f
yolo yolo@gmail.com yoloP(&ttJ7 o
NULL NULL NULL NULL

Elastic Beanstalk:



Step 4: Configure instance traffic and scaling

Edit

Instance traffic and scaling Info

Customize the capacity and scaling for your environment's instances. Select security groups to control instance traffic.Configure the software that runs on your environment's instances by setting platform-specific options.

Instances

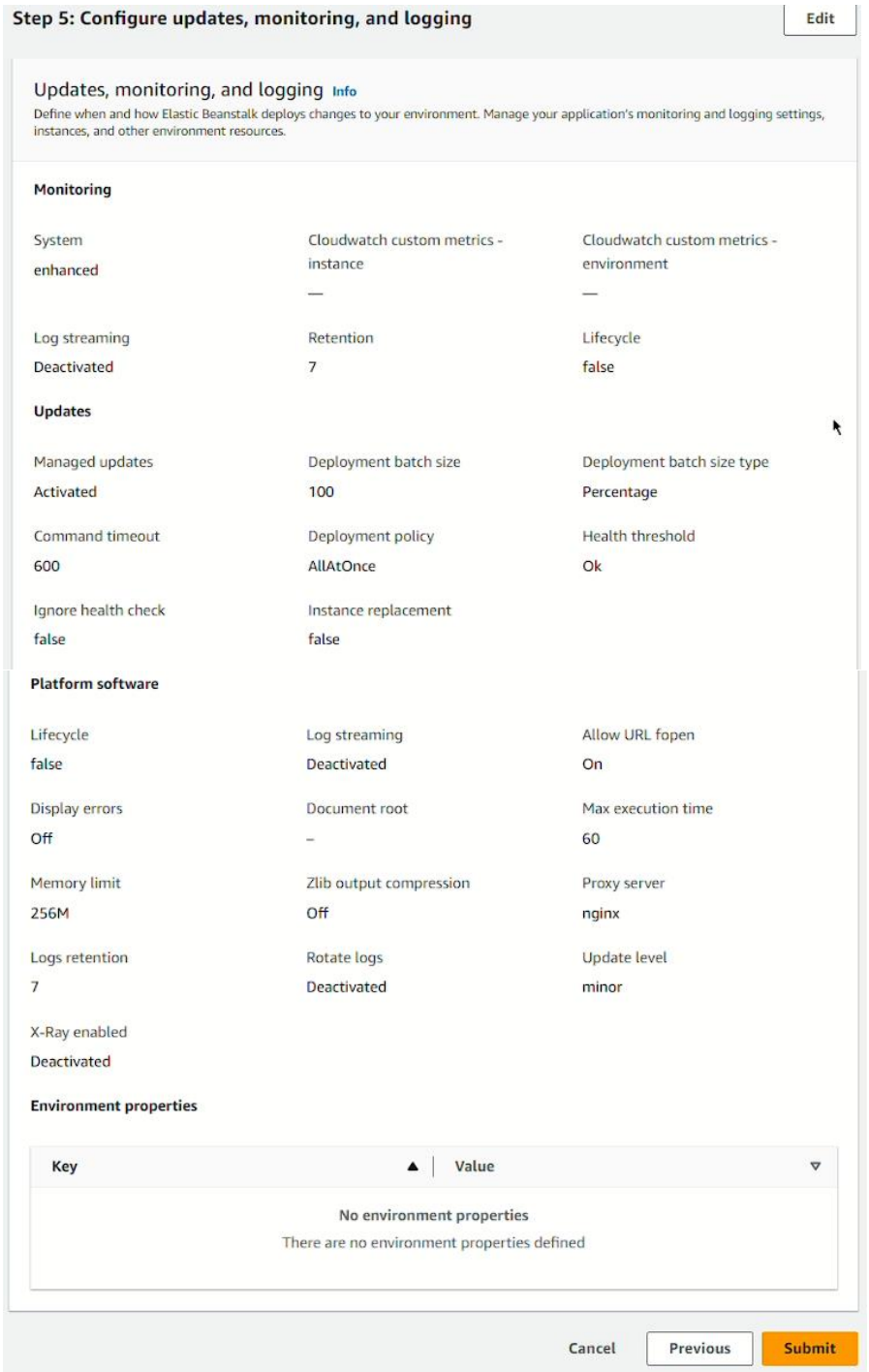
IMDSv1	EC2 Security Groups
Deactivated	sg-06550689795b4602e

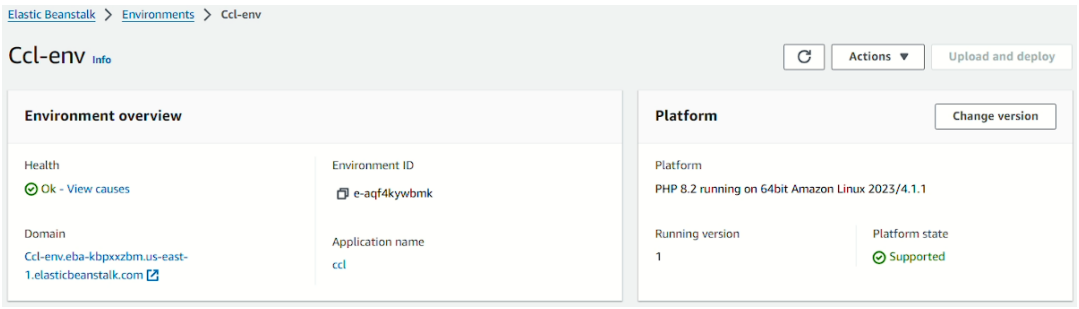
Capacity

Environment type	Fleet composition	On-demand base
Single instance	On-Demand instance	0
On-demand above base	Capacity rebalancing	Scaling cooldown
0	Deactivated	360
Processor type	Instance types	AMI ID
x86_64	t3.micro,t3.small	ami-0e38b869b8063a534
Availability Zones	Metric	Statistic
Any	NetworkOut	Average
Unit	Period	Breach duration
Bytes	5	5
Upper threshold	Scale up increment	Lower threshold
6000000	1	2000000
Scale down increment		
-1		

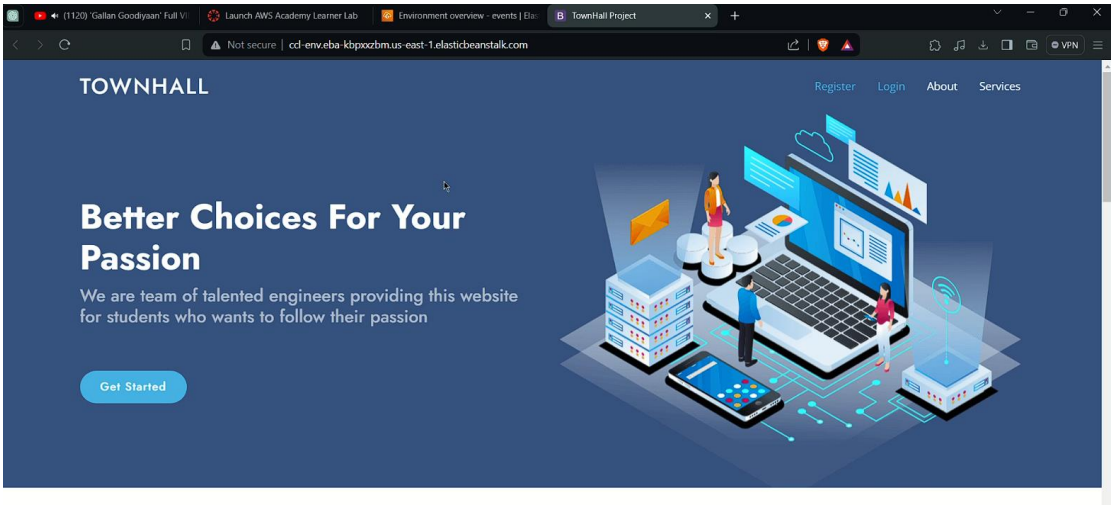
Load balancer

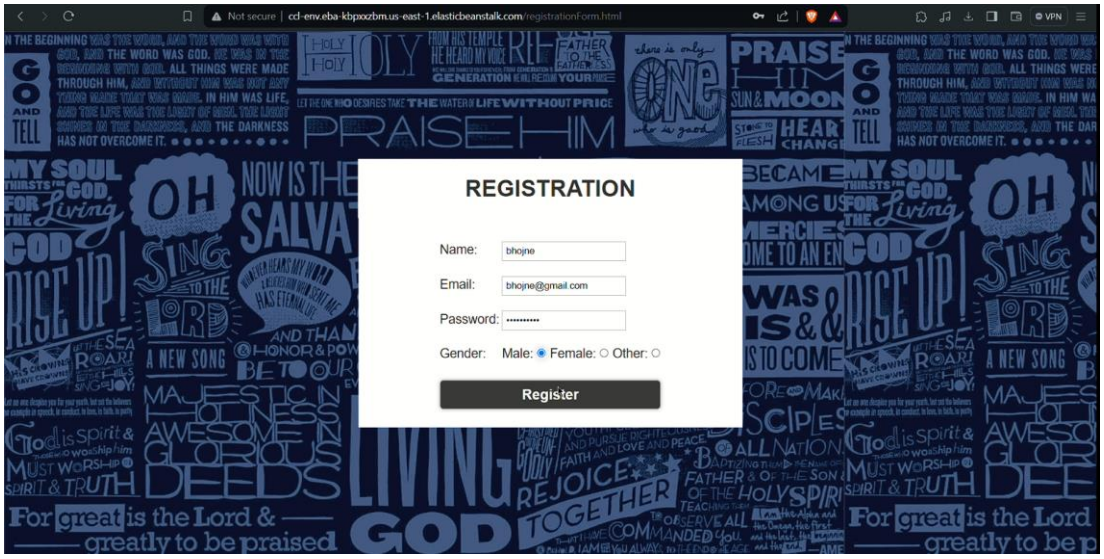
Load balancer visibility	Load balancer subnets	Load balancer type
public	subnet-09571f794707bcd49,subnet-016bf8afbe6081421,subnet-0891d4d4b4e7d90ba	application

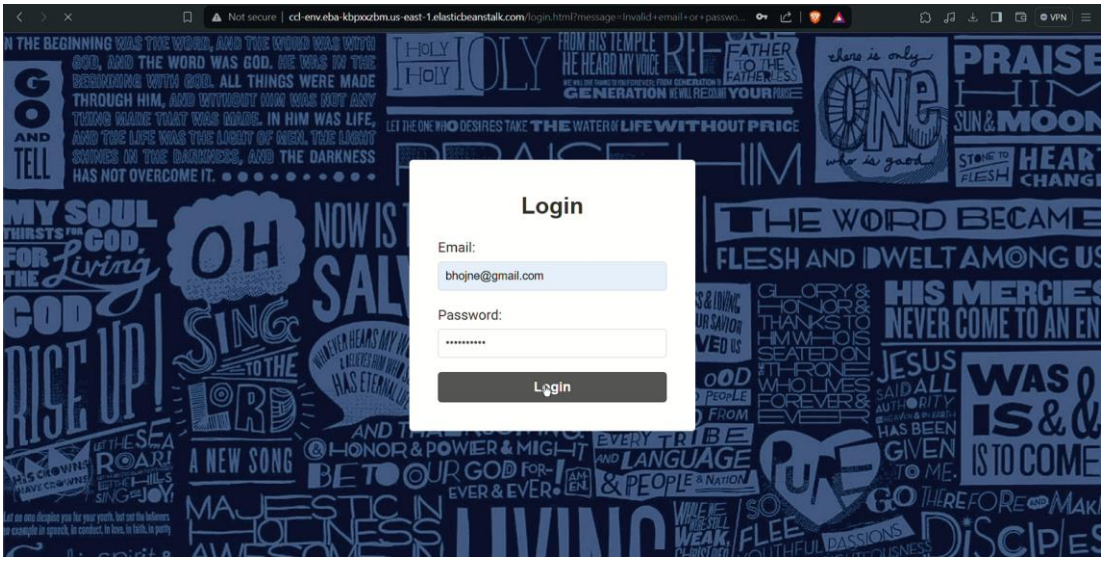

Step 5: Configure updates, monitoring, and logging
Edit
Updates, monitoring, and logging Info
Define when and how Elastic Beanstalk deploys changes to your environment. Manage your application's monitoring and logging settings, instances, and other environment resources.
Monitoring
System
enhanced
Cloudwatch custom metrics - instance
—
Cloudwatch custom metrics - environment
—
Log streaming
Deactivated
Retention
7
Lifecycle
false
Updates
Managed updates
Activated
Deployment batch size
100
Deployment batch size type
Percentage
Command timeout
600
Deployment policy
AllAtOnce
Health threshold
Ok
Ignore health check
false
Instance replacement
false
Platform software
Lifecycle
false
Log streaming
Deactivated
Allow URL fopen
On
Display errors
Off
Document root
–
Max execution time
60
Memory limit
256M
Zlib output compression
Off
Proxy server
nginx
Logs retention
7
Rotate logs
Deactivated
Update level
minor
X-Ray enabled
Deactivated
Environment properties
Key
Value
No environment properties
There are no environment properties defined
Cancel
Previous
Submit

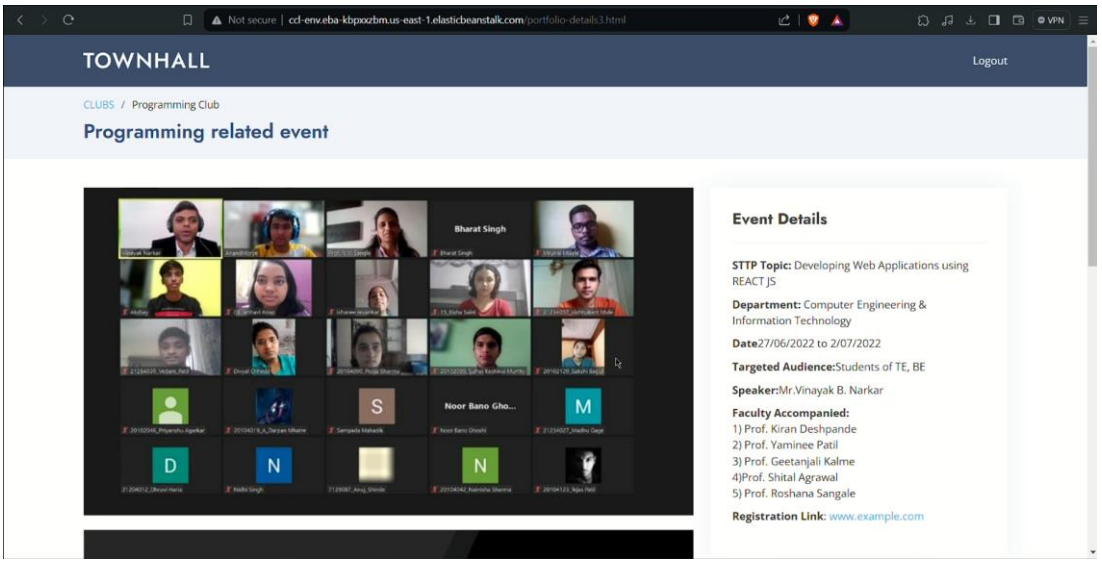

Elastic Beanstalk > Environments > Ccl-env
Ccl-env Info
Actions
Upload and deploy
Environment overview
Health
Ok - View causes
Environment ID
e-aqf4kywbmk
Domain
Ccl-env.eba-kbpxxzbm.us-east-1.elasticbeanstalk.com
Application name
ccl
Platform
Change version
Platform
PHP 8.2 running on 64bit Amazon Linux 2023/4.1.1
Running version
1
Platform state
Supported

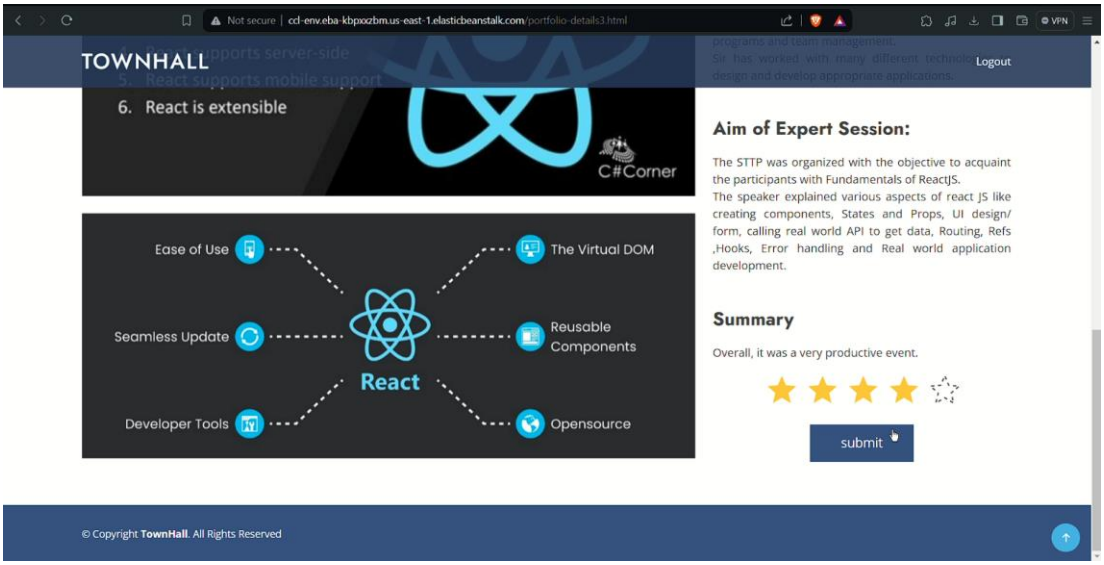
Deployed our project on the domain provided by elastic beanstalk:

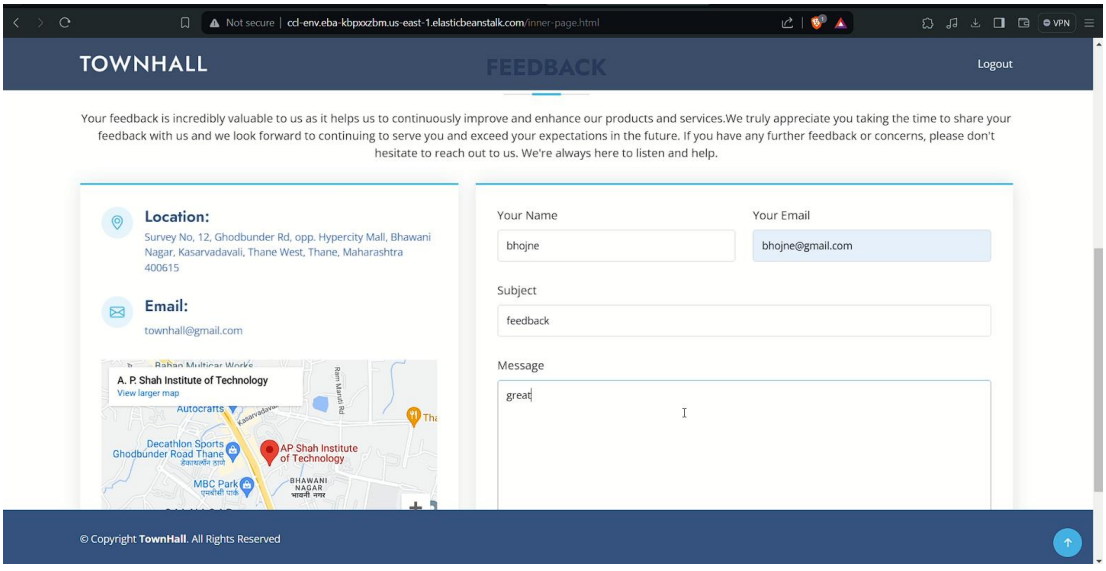




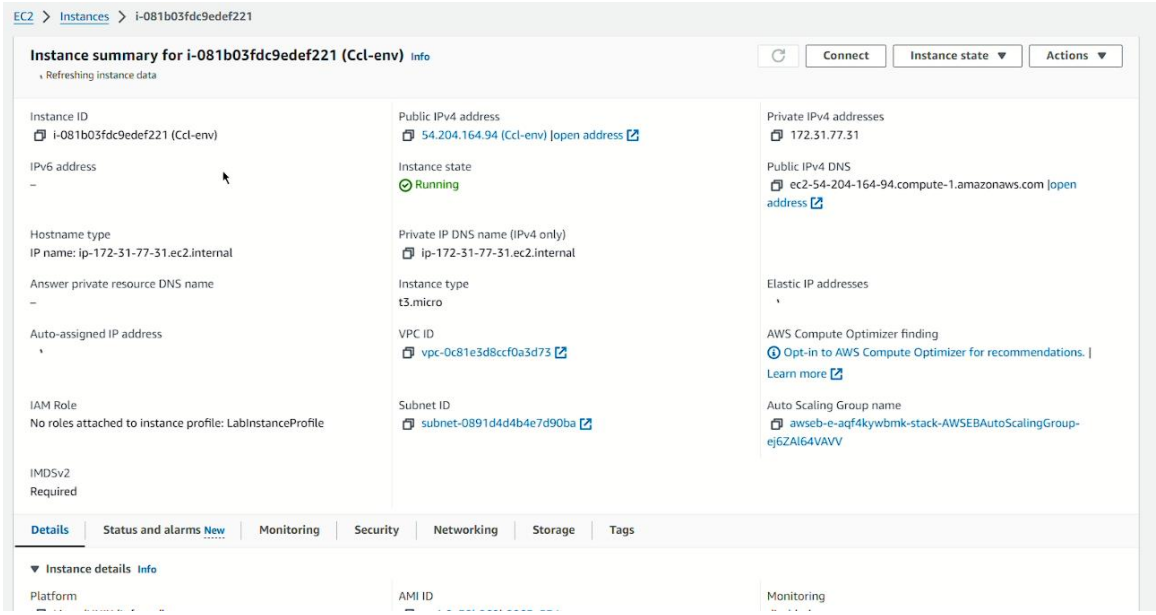


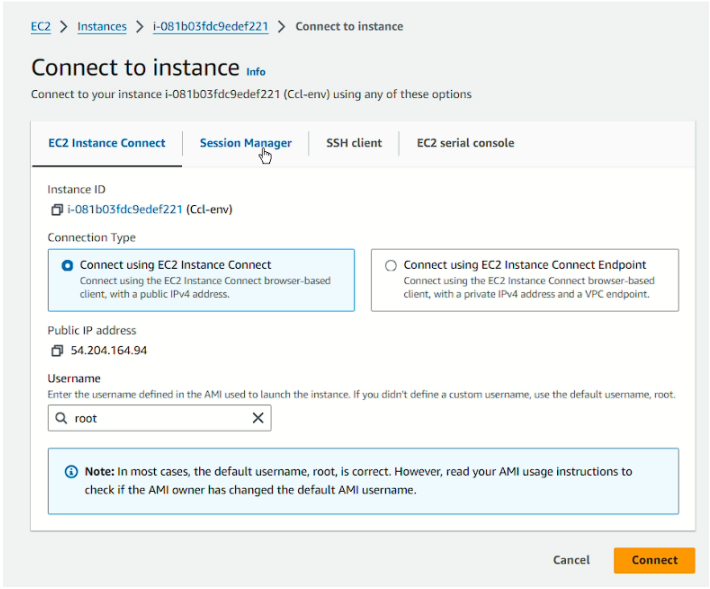

Not secure | ccl-env.eba-kbpxzbm.us-east-1.elasticbeanstalk.com/portfolio-details3.html
TOWNHALL
Logout
CLUBS / Programming Club
Programming related event
Bharat Singh
Noor Bano Gho...
Event Details
STTP Topic: Developing Web Applications using REACT JS
Department: Computer Engineering & Information Technology
Date27/06/2022 to 2/07/2022
Targeted Audience:Students of TE, BE
Speaker:Mr.Vinayak B. Narkar
Faculty Accompanied:
1) Prof. Kiran Deshpande
2) Prof. Yaminee Patil
3) Prof. Geetanjali Kalme
4)Prof. Shital Agrawal
5) Prof. Roshana Sangale
Registration Link: www.example.com


Not secure | ccl-env.eba-kbpxzbm.us-east-1.elasticbeanstalk.com/portfolio-details3.html
TOWNHALL
Logout
6. React is extensible
C#Corner
Ease of Use
The Virtual DOM
Seamless Update
Reusable Components
React
Developer Tools
Opensource
Aim of Expert Session:
The STTP was organized with the objective to acquaint the participants with Fundamentals of ReactJS.
The speaker explained various aspects of react JS like creating components, States and Props, UI design/ form, calling real world API to get data, Routing, Refs ,Hooks, Error handling and Real world application development.
Summary
Overall, it was a very productive event.
submit
© Copyright TownHall. All Rights Reserved


Not secure | ccl-env.eba-kbpxzbm.us-east-1.elasticbeanstalk.com/inner-page.html
TOWNHALL
FEEDBACK
Logout
Your feedback is incredibly valuable to us as it helps us to continuously improve and enhance our products and services.We truly appreciate you taking the time to share your feedback with us and we look forward to continuing to serve you and exceed your expectations in the future. If you have any further feedback or concerns, please don't hesitate to reach out to us. We're always here to listen and help.
Location:
Survey No, 12, Ghodbunder Rd, opp. Hypercity Mall, Bhawani Nagar, Kasarvadavali, Thane West, Thane, Maharashtra 400615
Email:
townhall@gmail.com
A. P. Shah Institute of Technology
View larger map
Autocratts
Decathlon Sports Ghodbunder Road Thane
AP Shah Institute of Technology
MBC Park
BHAWANI NAGAR
Your Name
bhojne
Your Email
bhojne@gmail.com
Subject
feedback
Message
great
© Copyright TownHall. All Rights Reserved

Elastic Compute Cloud (EC2):





Installing MariaDB for performing mysql operations:


```
[root@ip-172-31-77-31 ~]# mysql -h ccl.cdu8wqouo770.us-east-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 43
Server version: 8.0.35 Source distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MySQL [(none)]>
```

```
MySQL [(none)]> use Townhall;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

```
MySQL [Townhall]> select * from register_pg;
+---------+--------------------------+-----------------+--------+
| name    | email                    | password        | gender |
+---------+--------------------------+-----------------+--------+
| Atharv  | 21102154@gmail.com       | abcd            | m      |
| ajay    | ajay123@gmail.com        | sss             | m      |
| omom    | bagullalit31@gmail.com   | Om123@bagul     | m      |
| om      | bagullalit31@hotmail.com | 12345           | m      |
| bhojne  | bhojne@gmail.com         | Bhijne@123      | m      |
| gg      | gwqw@gmail.com           | yyyyP999*       | m      |
| dhdhdhd | hdhdhdh@gmail.com        | yyyyPP*9fgfgf   | f      |
| kiran   | kiran@gmail.com          | 22Werffffffff@  | f      |
| omkar   | omkar123@gmail.com       | 111             | m      |
| rajesh  | rajesh@gmail.in          | ttttttttttEE44@ | f      |
| yolo    | yolo@gmail.com           | yoloP(&ttJ7     | o      |
+---------+--------------------------+-----------------+--------+
11 rows in set (0.001 sec)

MySQL [Townhall]>
```

```
MySQL [Townhall]> insert into register_pg (name,email,password,gender) values ("amol","amol@gmail.com","Amol@123","m");
Query OK, 1 row affected (0.003 sec)
```

```
MySQL [Townhall]> select * from register_pg;
+---------+--------------------------+-----------------+--------+
| name    | email                    | password        | gender |
+---------+--------------------------+-----------------+--------+
| Atharv  | 21102154@gmail.com       | abcd            | m      |
| ajay    | ajay123@gmail.com        | sss             | m      |
| amol    | amol@gmail.com           | Amol@123        | m      |
| omom    | bagullalit31@gmail.com   | Om123@bagul     | m      |
| om      | bagullalit31@hotmail.com | 12345           | m      |
| bhojne  | bhojne@gmail.com         | Bhijne@123      | m      |
| gg      | gwqw@gmail.com           | yyyyP999*       | m      |
| dhdhdhd | hdhdhdh@gmail.com        | yyyyPP*9fgfgf   | f      |
| kiran   | kiran@gmail.com          | 22Werfffffff@   | f      |
| omkar   | omkar123@gmail.com       | 111             | m      |
| rajesh  | rajesh@gmail.in          | ttttttttttEE44@ | f      |
| yolo    | yolo@gmail.com           | yoloP(&ttJ7     | o      |
+---------+--------------------------+-----------------+--------+
12 rows in set (0.001 sec)
```

```
MySQL [Townhall]> update register_pg set password = "Amol@12345" where email = "amol@gmail.com";
Query OK, 1 row affected (0.004 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

```
MySQL [Townhall]> select * from register_pg where email = "amol@gmail.com";
+------+----------------+------------+--------+
| name | email          | password   | gender |
+------+----------------+------------+--------+
| amol | amol@gmail.com | Amol@12345 | m      |
+------+----------------+------------+--------+
1 row in set (0.001 sec)
```

```
MySQL [Townhall]> delete from register_pg where email = "amol@gmail.com";
Query OK, 1 row affected (0.005 sec)

MySQL [Townhall]> select * from register_pg;
+---------+--------------------------+-----------------+--------+
| name    | email                    | password        | gender |
+---------+--------------------------+-----------------+--------+
| Atharv  | 21102154@gmail.com       | abcd            | m      |
| ajay    | ajay123@gmail.com        | sss             | m      |
| omom    | bagullalit31@gmail.com   | Om123@bagul     | m      |
| om      | bagullalit31@hotmail.com | 12345           | m      |
| bhojne  | bhojne@gmail.com         | Bhijne@123      | m      |
| gg      | gwqw@gmail.com           | yyyyP999*       | m      |
| dhdhdhd | hdhdhdh@gmail.com        | yyyyPP*9fgfgf   | f      |
| kiran   | kiran@gmail.com          | 22Werfffffff@   | f      |
| omkar   | omkar123@gmail.com       | 111             | m      |
| rajesh  | rajesh@gmail.in          | ttttttttttEE44@ | f      |
| yolo    | yolo@gmail.com           | yoloP(&ttJ7     | o      |
+---------+--------------------------+-----------------+--------+
11 rows in set (0.001 sec)
```

Learning Outcome

The deployment of the "Townhall" website onto AWS using Elastic Beanstalk and integration with Amazon RDS offers valuable learning outcomes in AWS services, deployment best practices, database management, security implementation, application development, and problem-solving. We gain proficiency in AWS services, including Elastic Beanstalk and Amazon RDS, and learn to navigate the AWS Management Console, configure environment settings, deploy applications, provision database instances, and manage security groups effectively. We also develop database management skills by setting up and configuring Amazon RDS instances, understanding database engines, provisioning resources, and implementing security measures. Through the project, we enhance our application development skills by implementing CRUD operations within the "Townhall" website, including developing backend APIs, integrating frontend interfaces, and implementing user authentication and data management functionalities. We also gain insights into deployment best practices for web applications on AWS, security implementation, problem-solving, and troubleshooting complex cloud infrastructure issues. Overall, the project provides practical skills and knowledge essential for building, deploying, and managing cloud-based applications in today's digital landscape.