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

<u>Mini Project Report</u>

➢	**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.

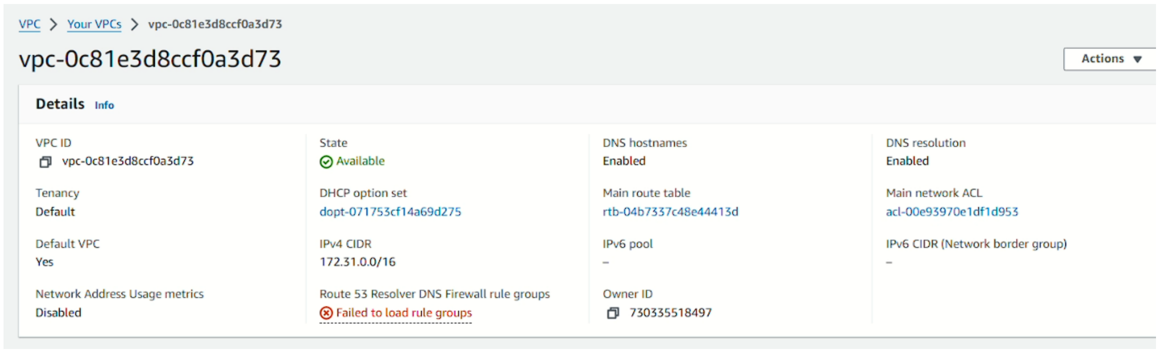
6. Implementing CRUD Operations:

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):



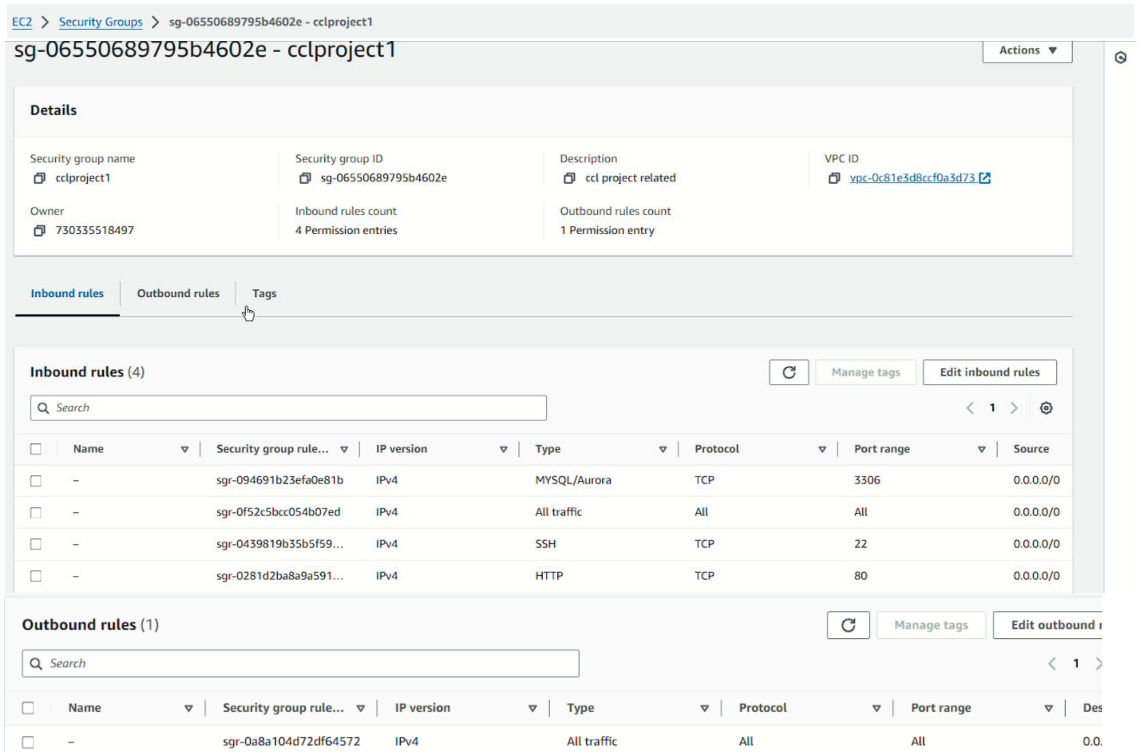
VPC > Your VPCs > vpc-0c81e3d8ccf0a3d73

vpc-0c81e3d8ccf0a3d73

Details Info

VPC ID vpc-0c81e3d8ccf0a3d73	State Available	DNS hostnames Enabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-071753cf14a69d275	Main route table rtb-04b7337c48e44413d	Main network ACL acl-00e93970e1df1d953
Default VPC Yes	IPv4 CIDR 172.31.0.0/16	IPv6 pool –	IPv6 CIDR (Network border group) –
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups Failed to load rule groups	Owner ID 730335518497	

Security Group:



EC2 > Security Groups > sg-06550689795b4602e - cclproject1

sg-06550689795b4602e - cclproject1

Details

Security group name cclproject1	Security group ID sg-06550689795b4602e	Description ccl project related	VPC ID vpc-0c81e3d8ccf0a3d73
Owner 730335518497	Inbound rules count 4 Permission entries	Outbound rules count 1 Permission entry	

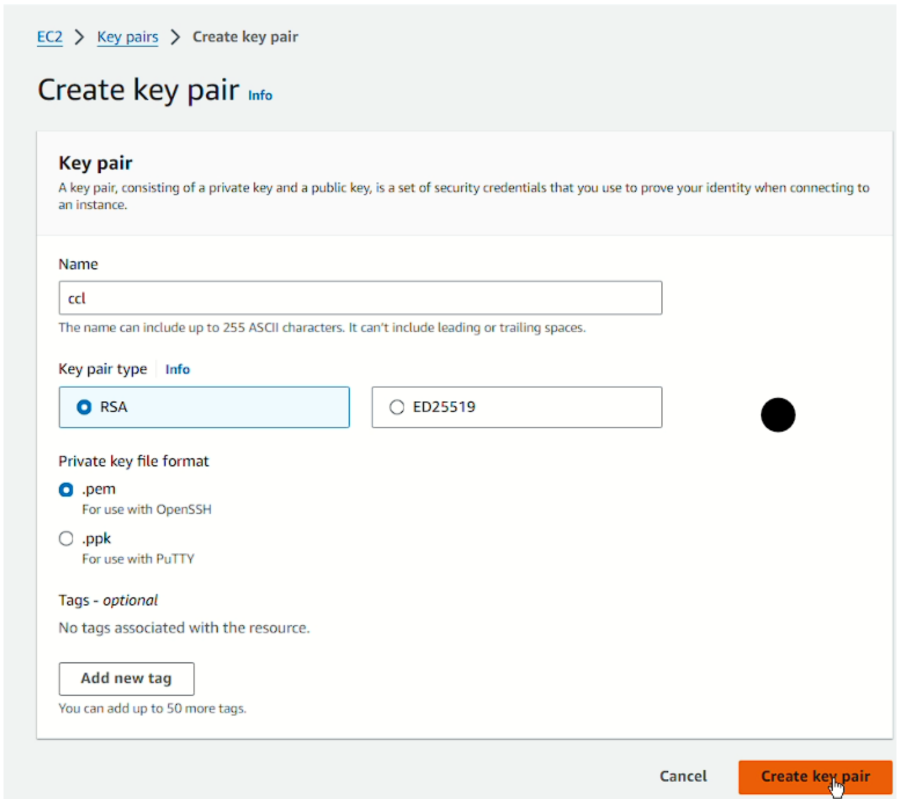
Inbound rules (4)

Name	Security group rule...	IP version	Type	Protocol	Port range	Source
–	sgr-094691b23efa0e81b	IPv4	MYSQL/Aurora	TCP	3306	0.0.0.0/0
–	sgr-0f52c5bcc054b07ed	IPv4	All traffic	All	All	0.0.0.0/0
–	sgr-0439819b35b5f59...	IPv4	SSH	TCP	22	0.0.0.0/0
–	sgr-0281d2ba8a9a591...	IPv4	HTTP	TCP	80	0.0.0.0/0

Outbound rules (1)

Name	Security group rule...	IP version	Type	Protocol	Port range	Des
–	sgr-0a8a104d72df64572	IPv4	All traffic	All	All	0.0.

Key Pair:



EC2 > Key pairs > Create key pair

Create key pair Info

Key pair

A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name: ccl

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type Info: [✓ RSA] ED25519

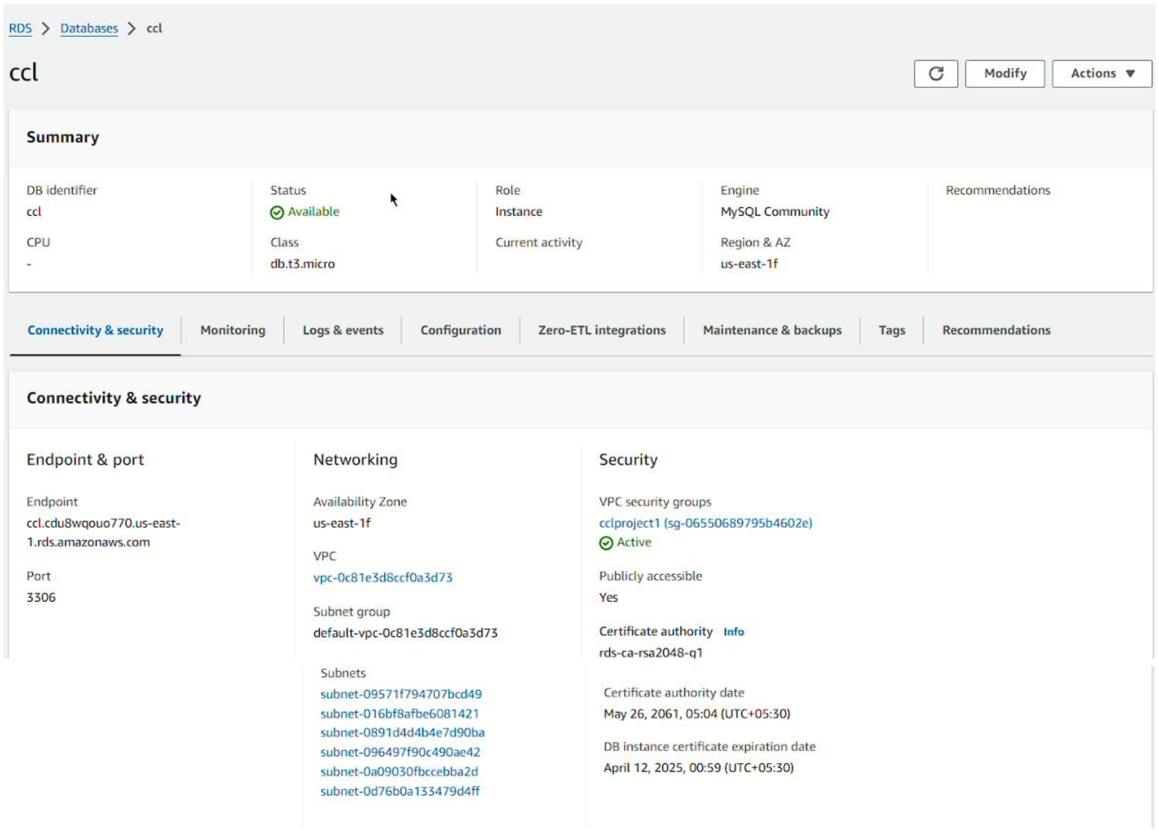
Private key file format:
- [✓ .pem] For use with OpenSSH
- .ppk For use with PuTTY

Tags - *optional*

No tags associated with the resource.

You can add up to 50 more tags.

Relational Database Service (RDS):



RDS > Databases > ccl

ccl

Summary

DB identifier	Status	Role	Engine	Recommendations
ccl	Available	Instance	MySQL Community	
CPU	Class	Current activity	Region & AZ	
-	db.t3.micro		us-east-1f	

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

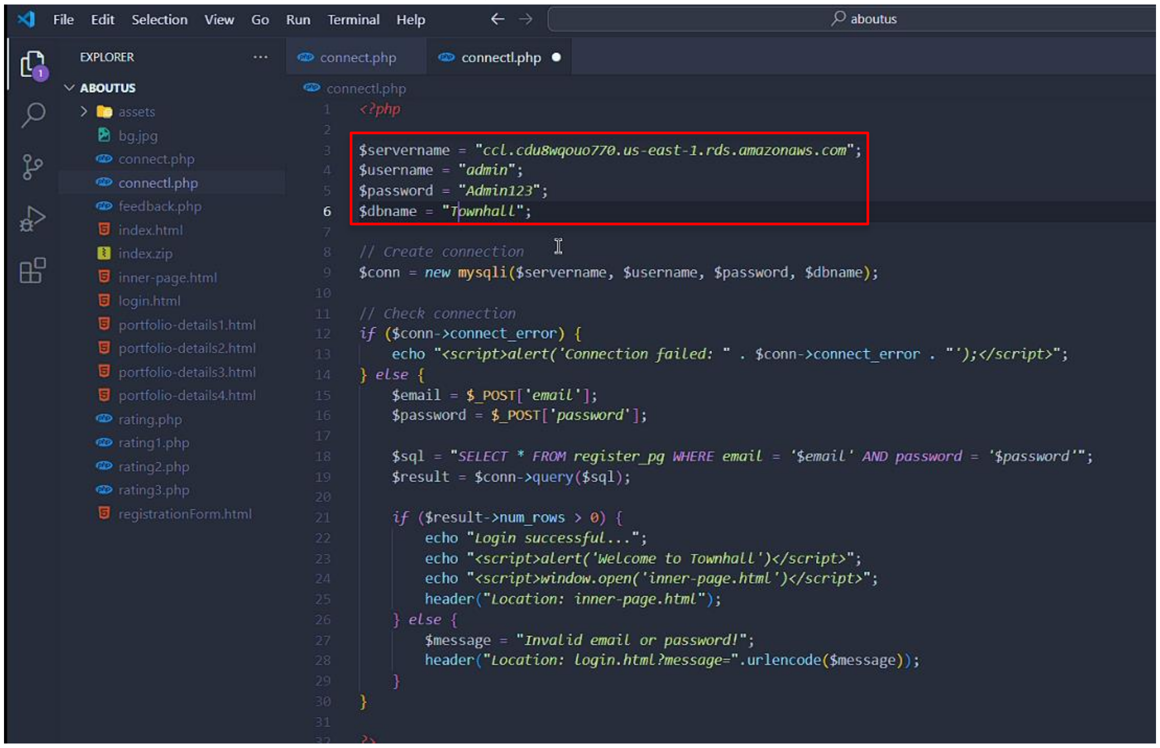
Connectivity & security

Endpoint & port
- Endpoint: ccl.cdu8wqouo770.us-east-1.rds.amazonaws.com
- Port: 3306

Networking
- Availability Zone: us-east-1f
- VPC: vpc-0c81e3d8ccf0a3d73
- Subnet group: default-vpc-0c81e3d8ccf0a3d73
- Subnets: subnet-09571f794707bcd49, subnet-016bf8afbe6081421, subnet-0891d4d4b4e7d90ba, subnet-096497f90c490ae42, subnet-0a09030fbccebba2d, subnet-0d76b0a133479d4ff

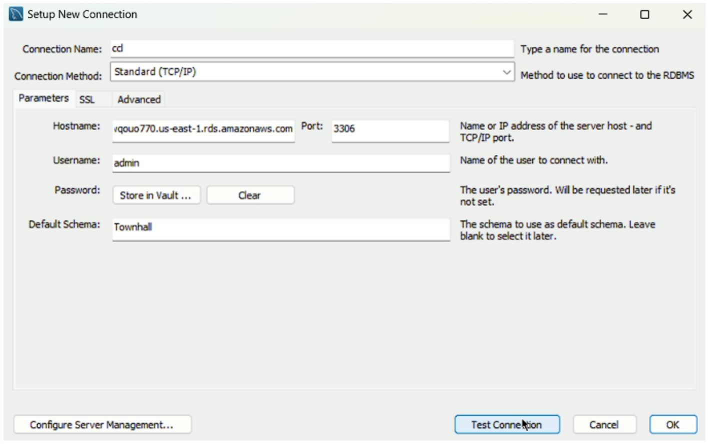
Security
- VPC security groups: cclproject1 (sg-06550689795b4602e) Active
- Publicly accessible: Yes
- Certificate authority Info: rds-ca-rsa2048-g1
- Certificate authority date: May 26, 2061, 05:04 (UTC+05:30)
- DB instance certificate expiration date: April 12, 2025, 00:59 (UTC+05:30)

<u>PHP code containing the endpoint given by RDS</u>:



```php
<?php

$servername = "ccl.cdu8wqouo770.us-east-1.rds.amazonaws.com";
$username = "admin";
$password = "Admin123";
$dbname = "Townhall";

// Create connection
$conn = new mysqli($servername, $username, $password, $dbname);

// Check connection
if ($conn->connect_error) {
    echo "<script>alert('Connection failed: " . $conn->connect_error . "');</script>";
} else {
    $email = $_POST['email'];
    $password = $_POST['password'];

    $sql = "SELECT * FROM register_pg WHERE email = '$email' AND password = '$password'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        echo "Login successful...";
        echo "<script>alert('Welcome to Townhall')</script>";
        echo "<script>window.open('inner-page.html')</script>";
        header("Location: inner-page.html");
    } else {
        $message = "Invalid email or password!";
        header("Location: login.html?message=".urlencode($message));
    }
}

?>
```

<u>Creating connection with workbench to import the database schema into RDS:</u>



Connection Name: cd — Type a name for the connection

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

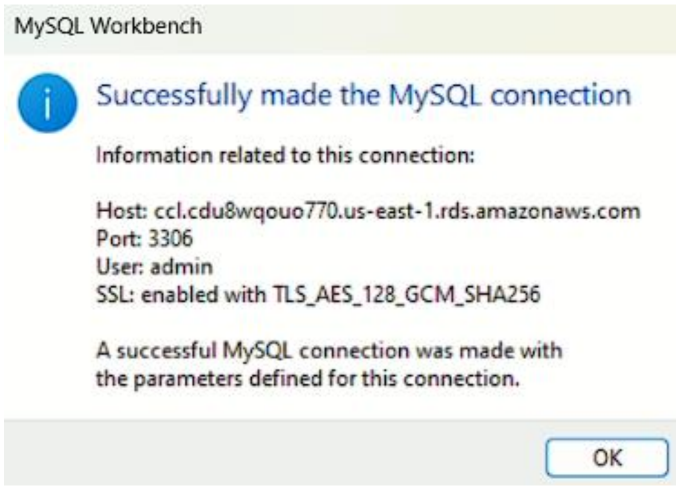
Hostname: vqouo770.us-east-1.rds.amazonaws.com Port: 3306 — Name or IP address of the server host - and TCP/IP port.

Username: admin — Name of the user to connect with.

Password: — The user's password. Will be requested later if it's not set.

Default Schema: Townhall — The schema to use as default schema. Leave blank to select it later.

<u>Connected RDS with MySQL Workbench and imported the db schema</u>:

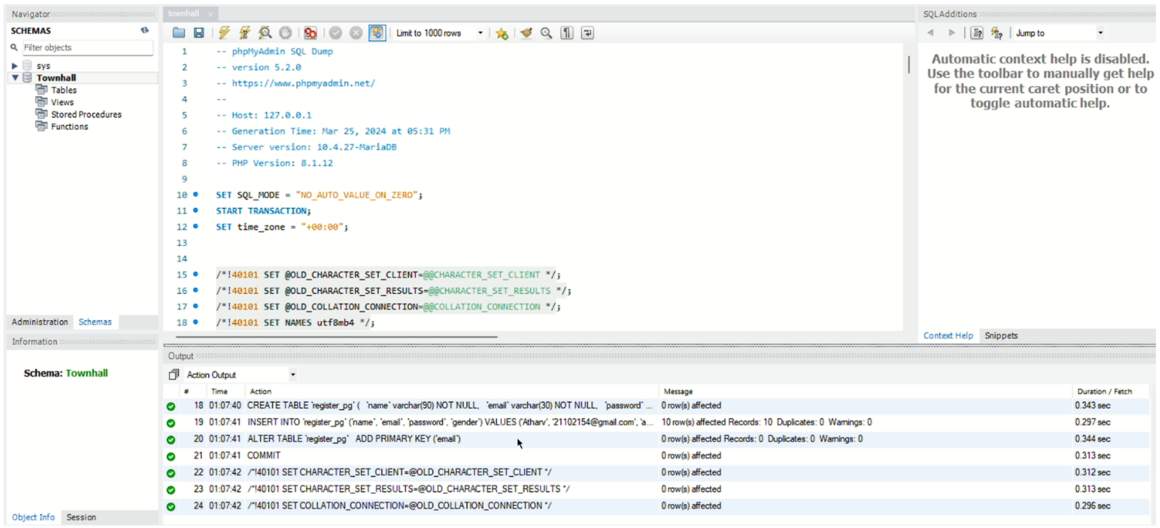


Successfully made the MySQL connection

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.



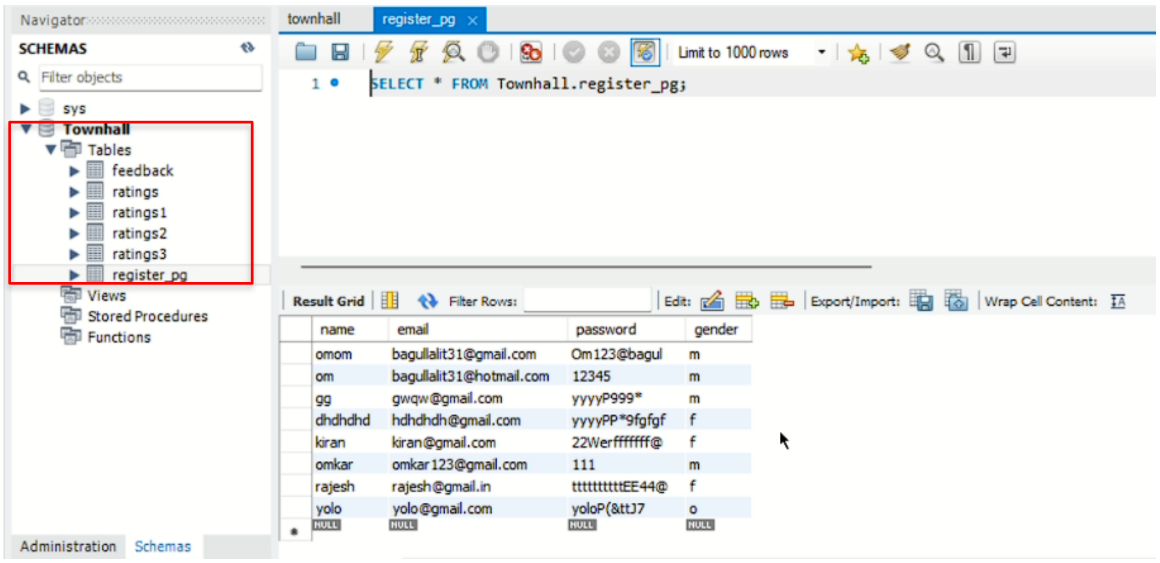
Schema: Townhall

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```sql
-- 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 */;
```

Output — Action Output

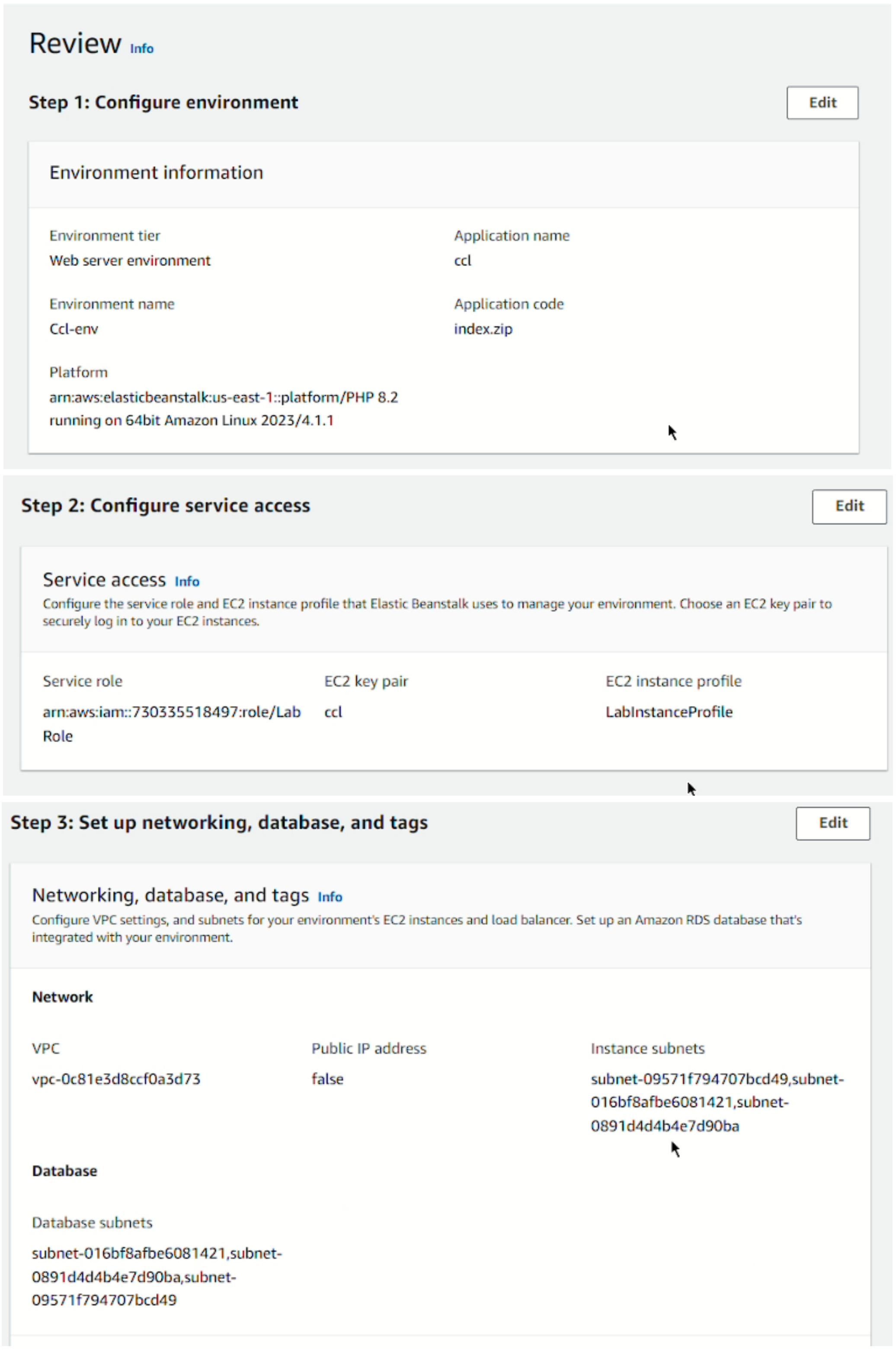
```
#   Time      Action                                                                                           Message                                                            Duration / Fetch
18  01:07:40  CREATE TABLE `register_pg` (  `name` varchar(90) NOT NULL,  `email` varchar(30) NOT NULL,  `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
```



```sql
SELECT * FROM Townhall.register_pg;
```

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:



Review Info

Step 1: Configure environment

Edit

Environment information

Environment tier	Application name
Web server environment	ccl
Environment name	Application code
Ccl-env	index.zip
Platform	
arn:aws:elasticbeanstalk:us-east-1::platform/PHP 8.2 running on 64bit Amazon Linux 2023/4.1.1	

Step 2: Configure service access

Edit

Service access Info

Configure the service role and EC2 instance profile that Elastic Beanstalk uses to manage your environment. Choose an EC2 key pair to securely log in to your EC2 instances.

Service role	EC2 key pair	EC2 instance profile
arn:aws:iam::730335518497:role/LabRole	ccl	LabInstanceProfile

Step 3: Set up networking, database, and tags

Edit

Networking, database, and tags Info

Configure VPC settings, and subnets for your environment's EC2 instances and load balancer. Set up an Amazon RDS database that's integrated with your environment.

Network

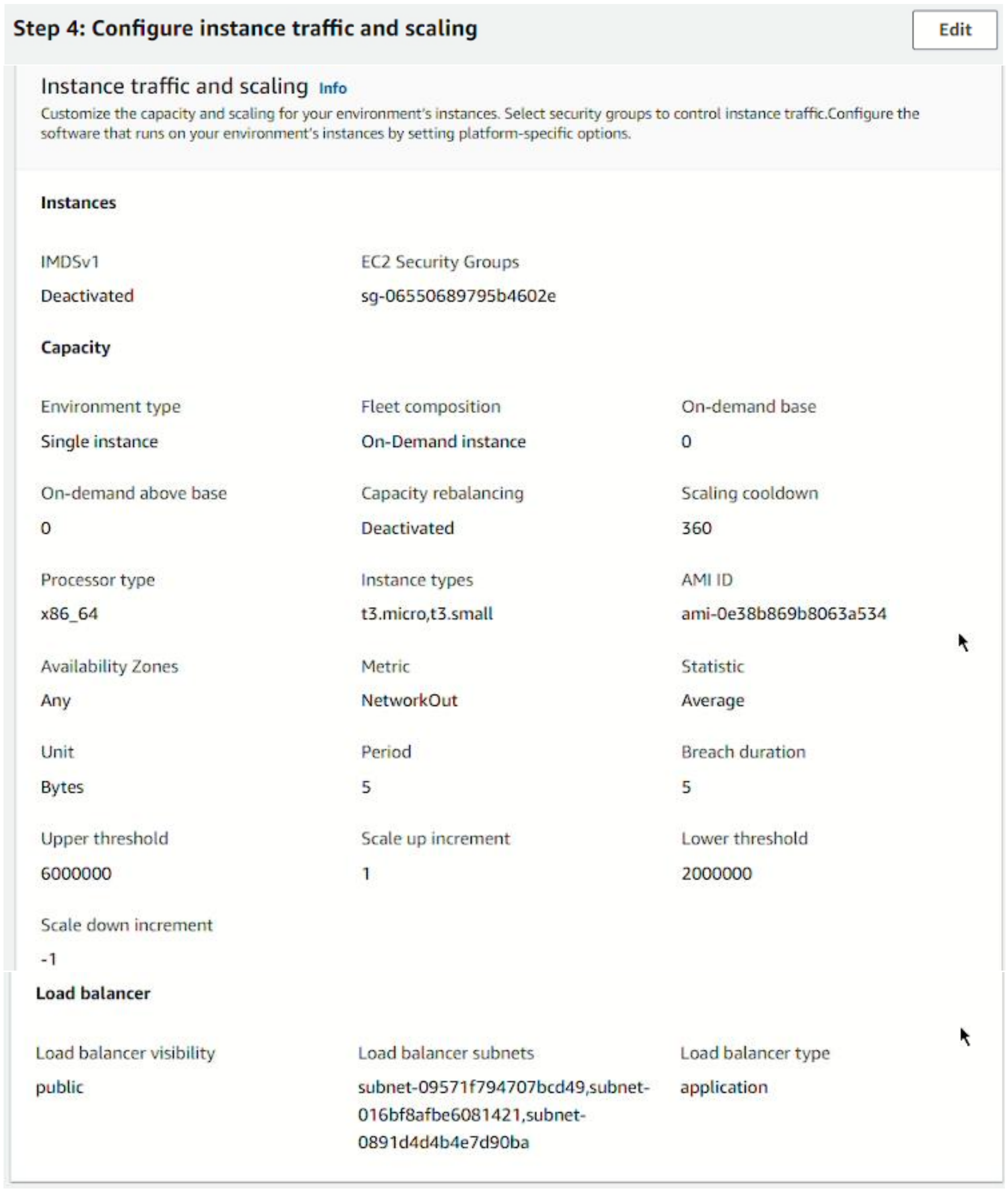
VPC	Public IP address	Instance subnets
vpc-0c81e3d8ccf0a3d73	false	subnet-09571f794707bcd49,subnet-016bf8afbe6081421,subnet-0891d4d4b4e7d90ba

Database

Database subnets

subnet-016bf8afbe6081421,subnet-0891d4d4b4e7d90ba,subnet-09571f794707bcd49

Configuration :



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

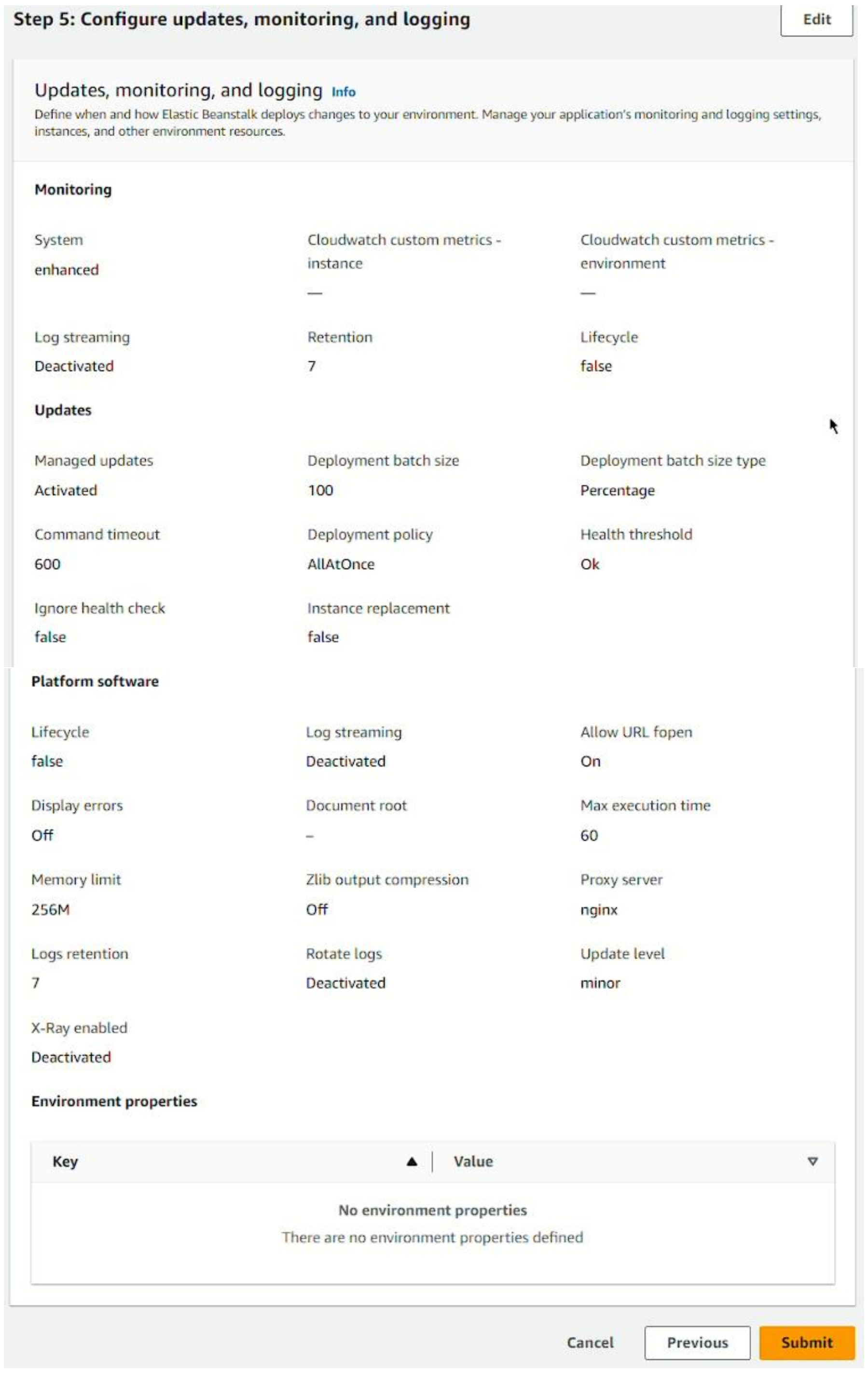
IMDSv1	EC2 Security Groups
Deactivated	sg-06550689795b4602e

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

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	Cloudwatch custom metrics - instance	Cloudwatch custom metrics - environment
enhanced	—	—
Log streaming	Retention	Lifecycle
Deactivated	7	false

Updates

Managed updates	Deployment batch size	Deployment batch size type
Activated	100	Percentage
Command timeout	Deployment policy	Health threshold
600	AllAtOnce	Ok
Ignore health check	Instance replacement	
false	false	

Platform software

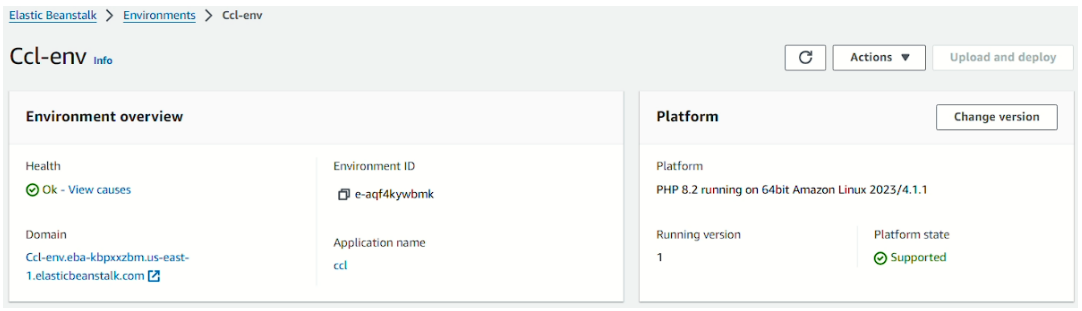
Lifecycle	Log streaming	Allow URL fopen
false	Deactivated	On
Display errors	Document root	Max execution time
Off	–	60
Memory limit	Zlib output compression	Proxy server
256M	Off	nginx
Logs retention	Rotate logs	Update level
7	Deactivated	minor
X-Ray enabled		
Deactivated		

Environment properties

Key	Value

No environment properties

There are no environment properties defined



Ccl-env Info

Environment overview

Health	Environment ID
Ok - View causes	e-aqf4kywbmk
Domain	Application name
Ccl-env.eba-kbpxxzbm.us-east-1.elasticbeanstalk.com	ccl

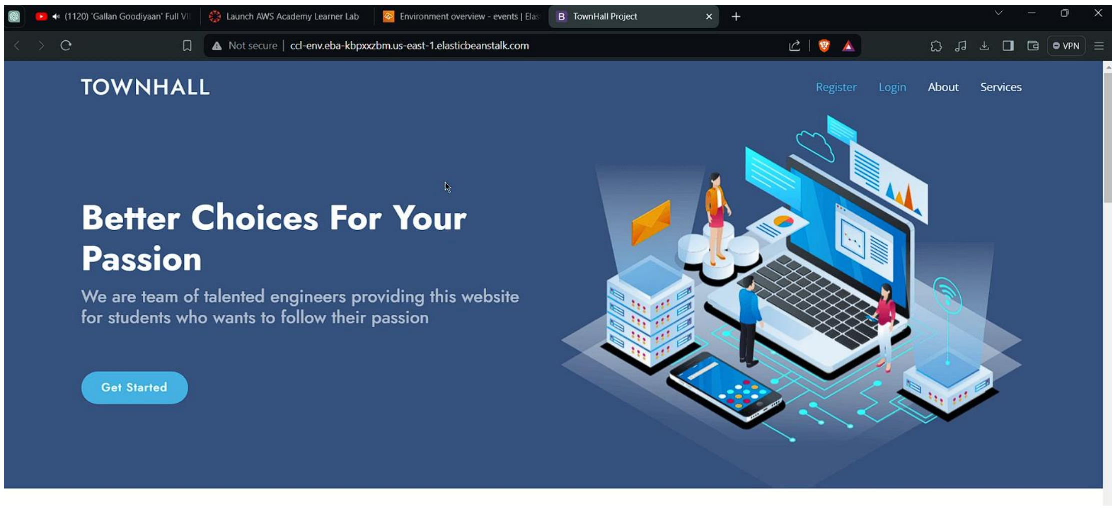
Platform

Platform

PHP 8.2 running on 64bit Amazon Linux 2023/4.1.1

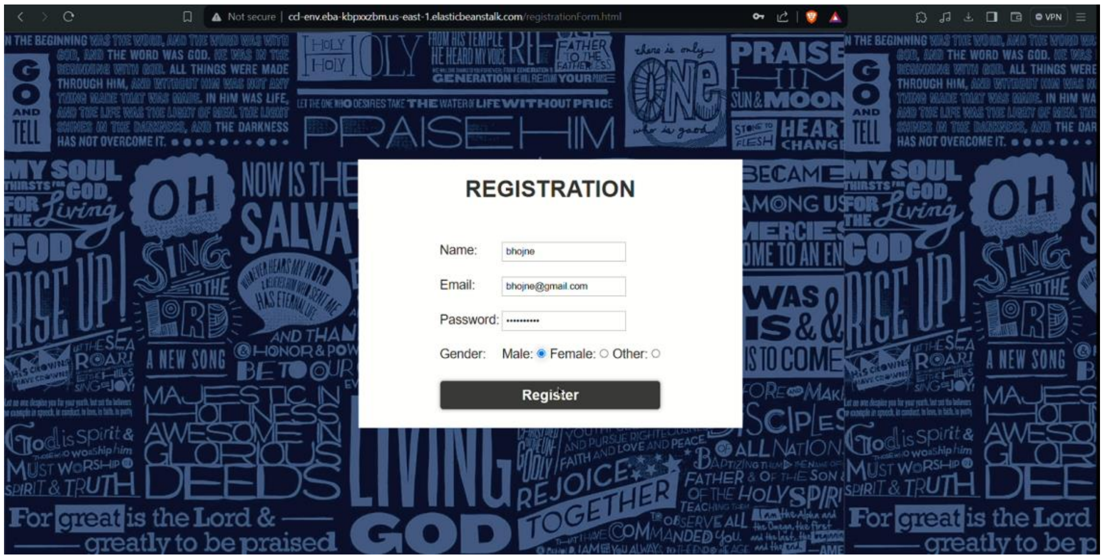
Running version	Platform state
1	Supported

<u>Deployed our project on the domain provided by elastic beanstalk</u>:



Not secure | ccl-env.eba-kbpxxzbm.us-east-1.elasticbeanstalk.com

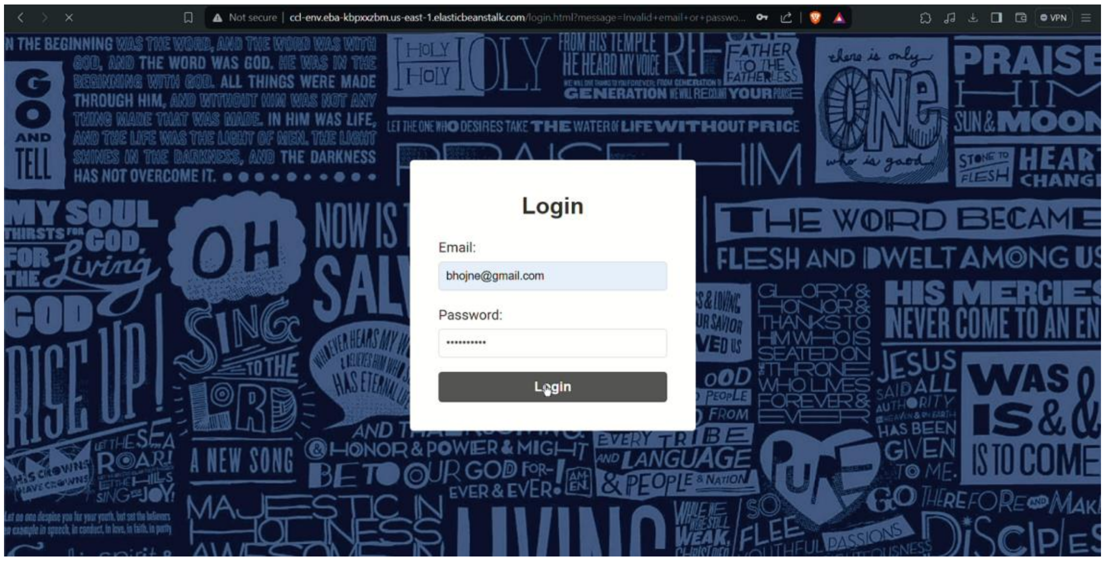
TOWNHALL
Register
Login
About
Services
Better Choices For Your Passion
We are team of talented engineers providing this website for students who wants to follow their passion
Get Started



Not secure | ccl-env.eba-kbpxxzbm.us-east-1.elasticbeanstalk.com/registrationForm.html

REGISTRATION

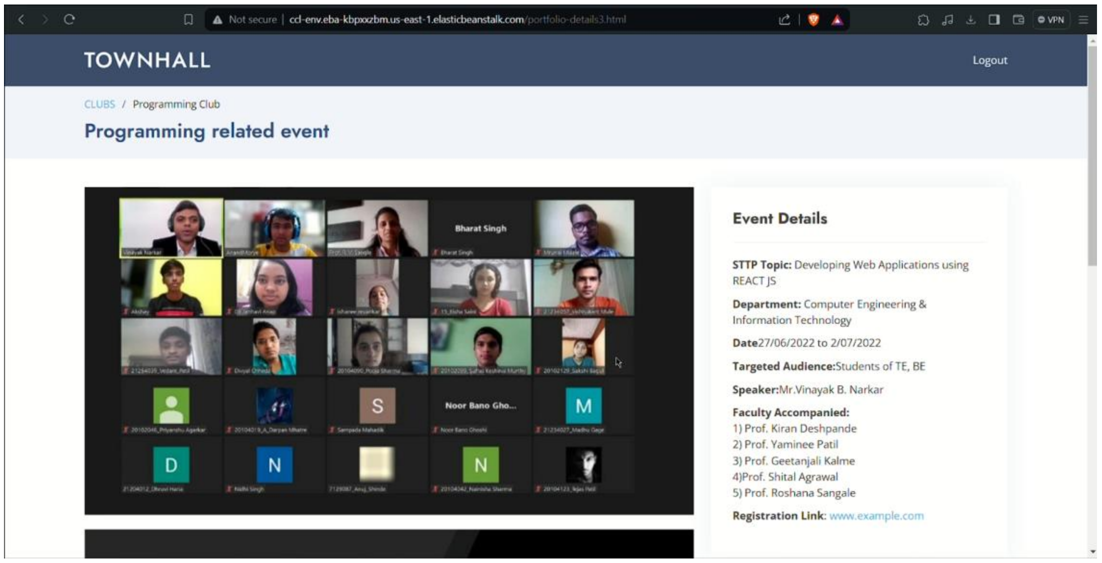
Name:	bhojne
Email:	bhojne@gmail.com
Password:	••••••••••
Gender:	Male: [✓ ●] Female: ○ Other: ○



Not secure | ccl-env.eba-kbpxxzbm.us-east-1.elasticbeanstalk.com/login.html?message=Invalid+email+or+passwo...

Login

Email:	bhojne@gmail.com
Password:	••••••••••



Not secure | ccl-env.eba-kbpxxzbm.us-east-1.elasticbeanstalk.com/portfolio-details3.html

TOWNHALL

Logout

CLUBS / Programming Club

Programming related event

Event Details

STTP Topic: Developing Web Applications using REACT JS

Department: Computer Engineering & Information Technology

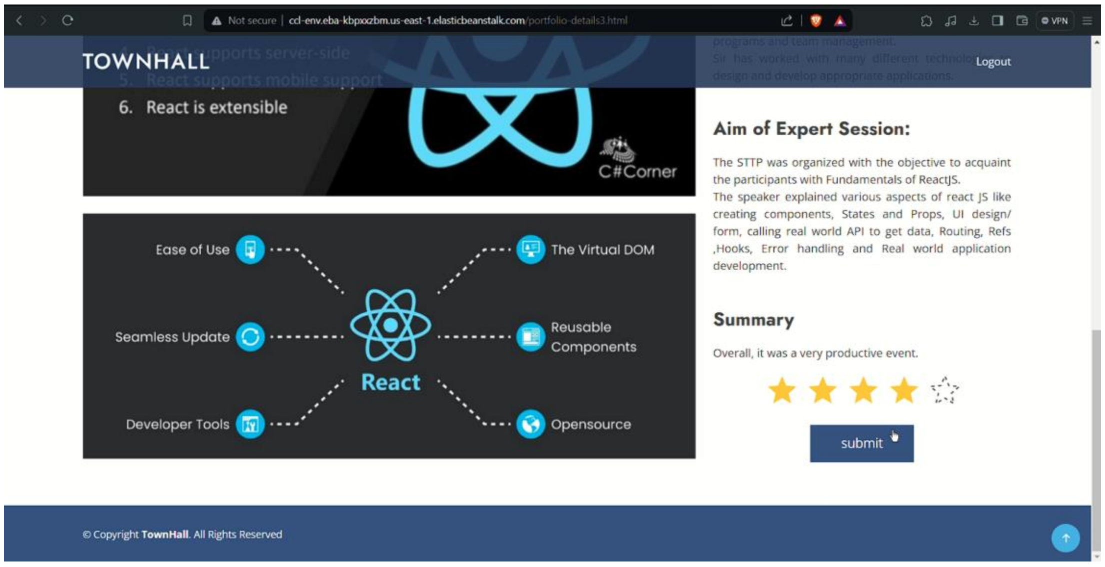
Date27/06/2022 to 2/07/2022

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-kbpxxzbm.us-east-1.elasticbeanstalk.com/portfolio-details3.html

TOWNHALL

Logout

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.

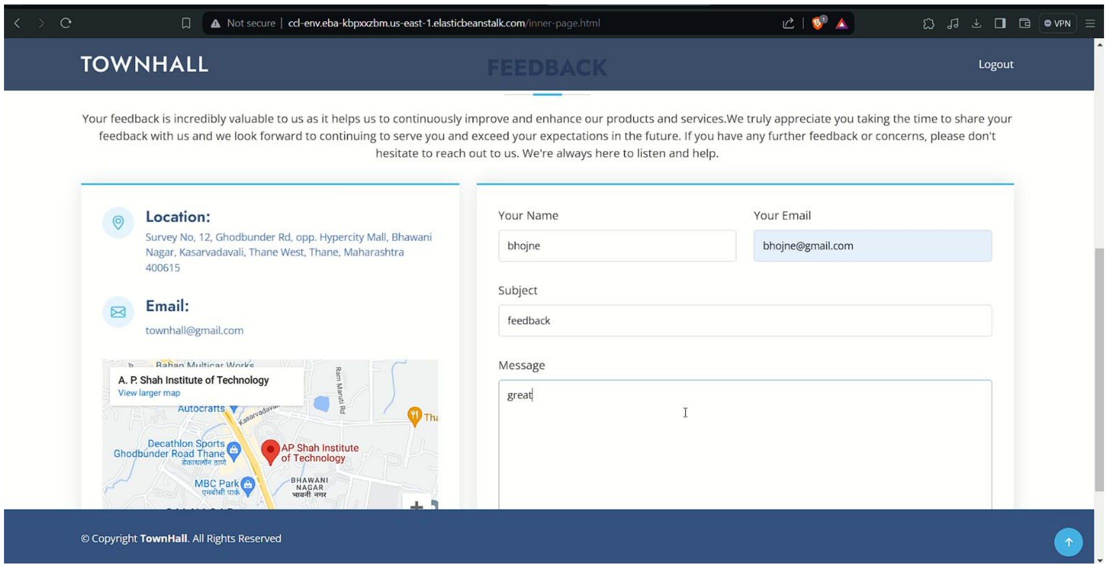
Summary

Overall, it was a very productive event.

★★★★☆

submit

© Copyright **TownHall**. All Rights Reserved



Not secure | ccl-env.eba-kbpxxzbm.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

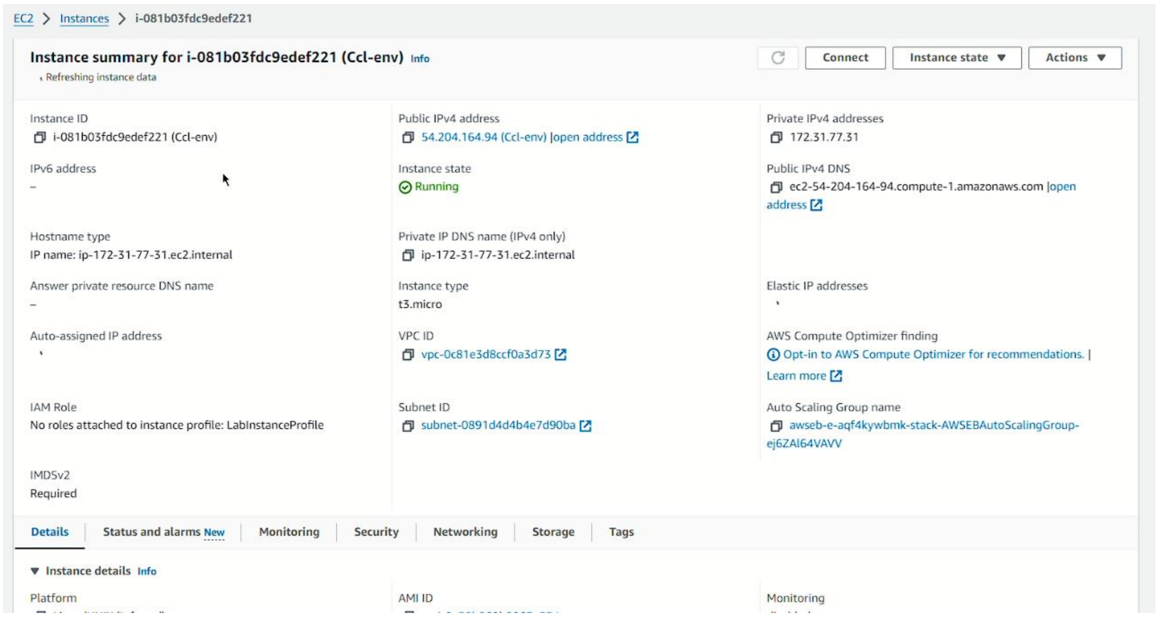
Email:
townhall@gmail.com

A. P. Shah Institute of Technology
View larger map

Your Name	bhojne
Your Email	bhojne@gmail.com
Subject	feedback
Message	great

© Copyright **TownHall**. All Rights Reserved

Elastic Compute Cloud (EC2):



EC2 > Instances > i-081b03fdc9edef221

Instance summary for i-081b03fdc9edef221 (Ccl-env) Info

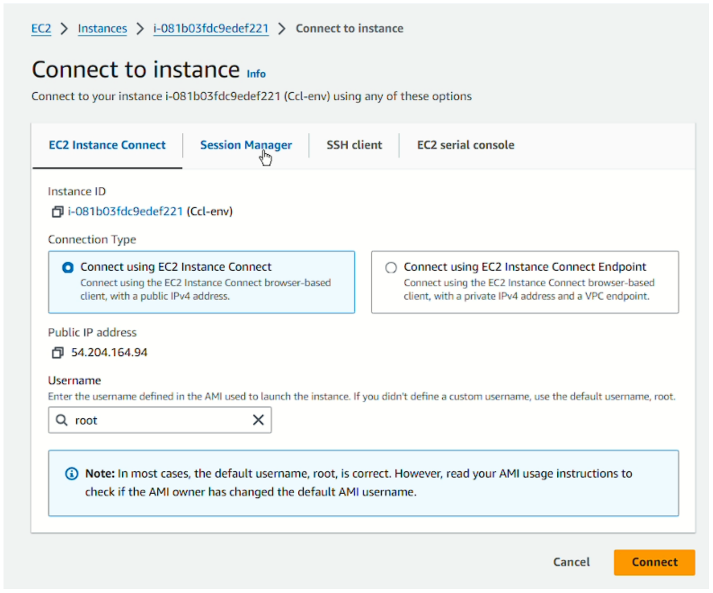
Refreshing instance data

Instance ID i-081b03fdc9edef221 (Ccl-env)	Public IPv4 address 54.204.164.94 (Ccl-env) |open address	Private IPv4 addresses 172.31.77.31
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-54-204-164-94.compute-1.amazonaws.com |open address
Hostname type IP name: ip-172-31-77-31.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-77-31.ec2.internal	
Answer private resource DNS name –	Instance type t3.micro	Elastic IP addresses
Auto-assigned IP address	VPC ID vpc-0c81e3d8ccf0a3d73	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role No roles attached to instance profile: LabInstanceProfile	Subnet ID subnet-0891d4d4b4e7d90ba	Auto Scaling Group name awseb-e-aqf4kywbmk-stack-AWSEBAutoScalingGroup-ej6ZAl64VAVV
IMDSv2 Required		

Details | Status and alarms New | Monitoring | Security | Networking | Storage | Tags

Instance details Info

Platform | AMI ID | Monitoring



EC2 > Instances > i-081b03fdc9edef221 > Connect to instance

Connect to instance Info

Connect to your instance i-081b03fdc9edef221 (Ccl-env) using any of these options

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID
i-081b03fdc9edef221 (Ccl-env)

Connection Type

[✓ Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.]

Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
54.204.164.94

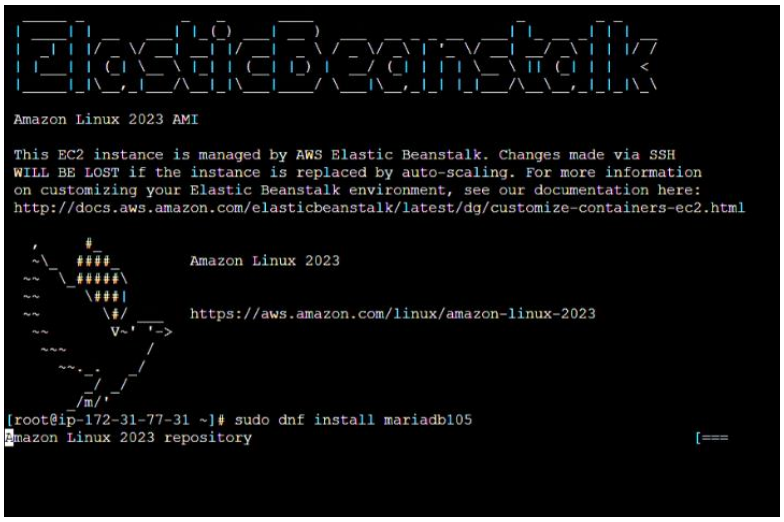
Username
Enter the username defined in the AMI used to launch the instance. If you didn't define a custom username, use the default username, root.

root

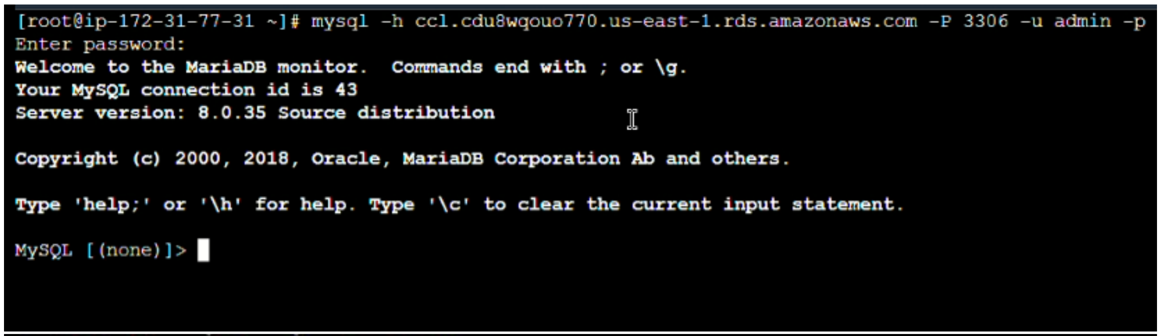
Note: In most cases, the default username, root, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Cancel | Connect

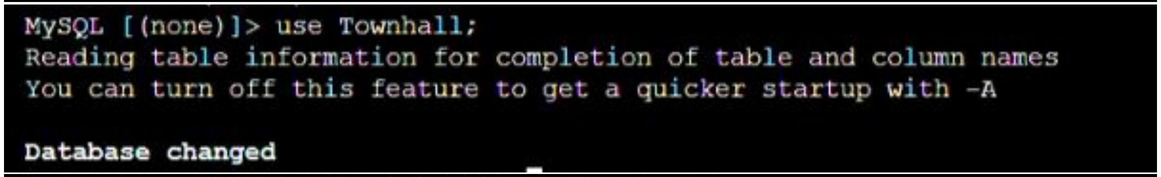
<u>Installing MariaDB for performing mysql CRUD operations</u>:



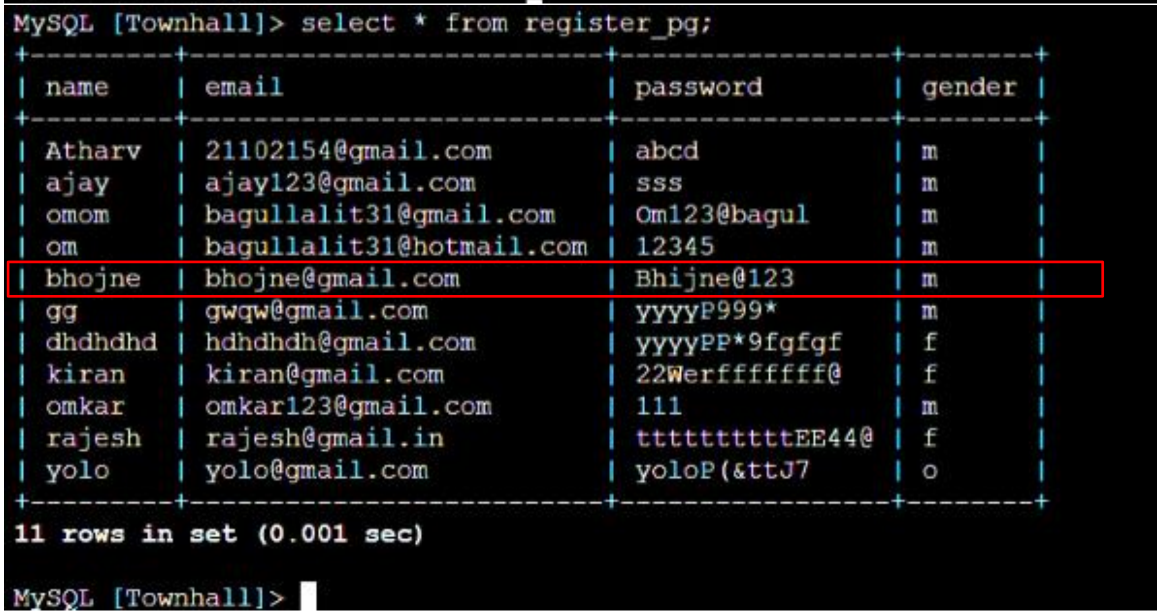
```
Amazon Linux 2023 AMI

This EC2 instance is managed by AWS Elastic Beanstalk. Changes made via SSH
WILL BE LOST if the instance is replaced by auto-scaling. For more information
on customizing your Elastic Beanstalk environment, see our documentation here:
http://docs.aws.amazon.com/elasticbeanstalk/latest/dg/customize-containers-ec2.html

   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[root@ip-172-31-77-31 ~]# sudo dnf install mariadb105
Amazon Linux 2023 repository                                          [===
```



```
[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)]>
```

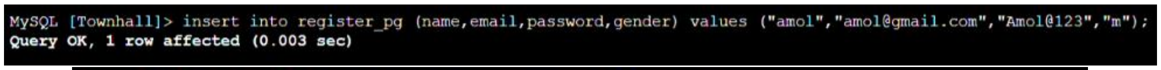


```
MySQL [(none)]> use Townhall;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
```

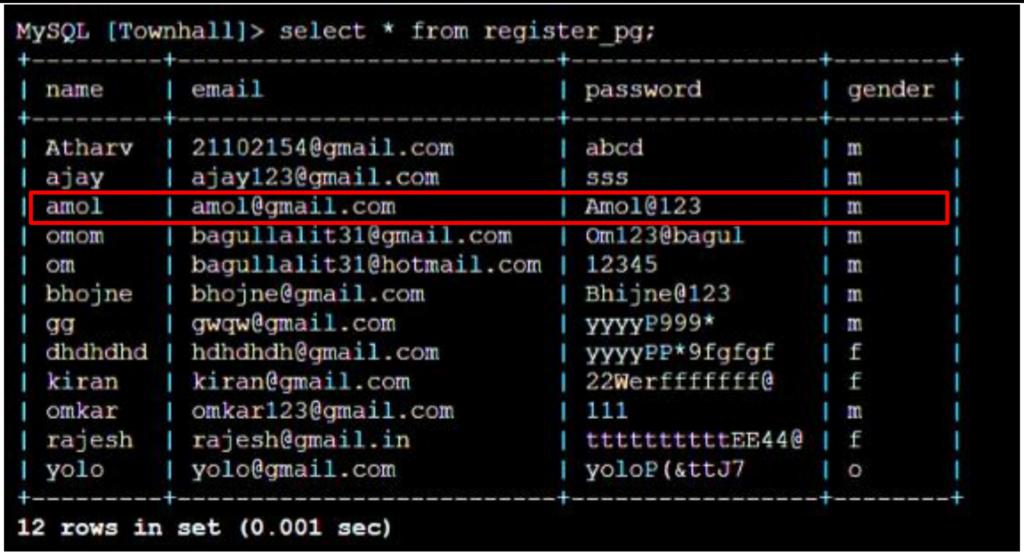


```
MySQL [Townhall]> select * from register_pg;
+---------+--------------------------+------------------+--------+
| name    | email                    | password         | gender |
+---------+--------------------------+------------------+--------+
| Atharv  | 21102154@gmail.com       | abcd             | 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          | 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)

MySQL [Townhall]>
```

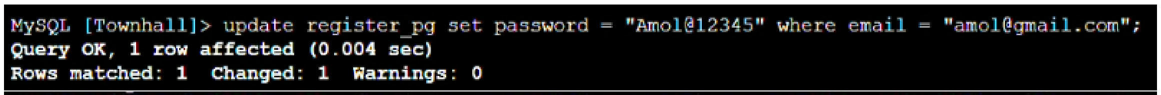
<u>Create , Read , Update & Delete Operations</u>:



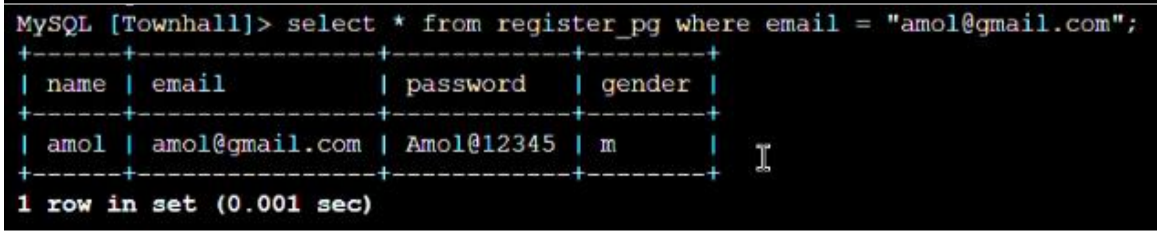
```
MySQL [Townhall]> insert into register_pg (name,email,password,gender) values ("amol","amol@gmail.com","Amol@123","m");
Query OK, 1 row affected (0.003 sec)
```



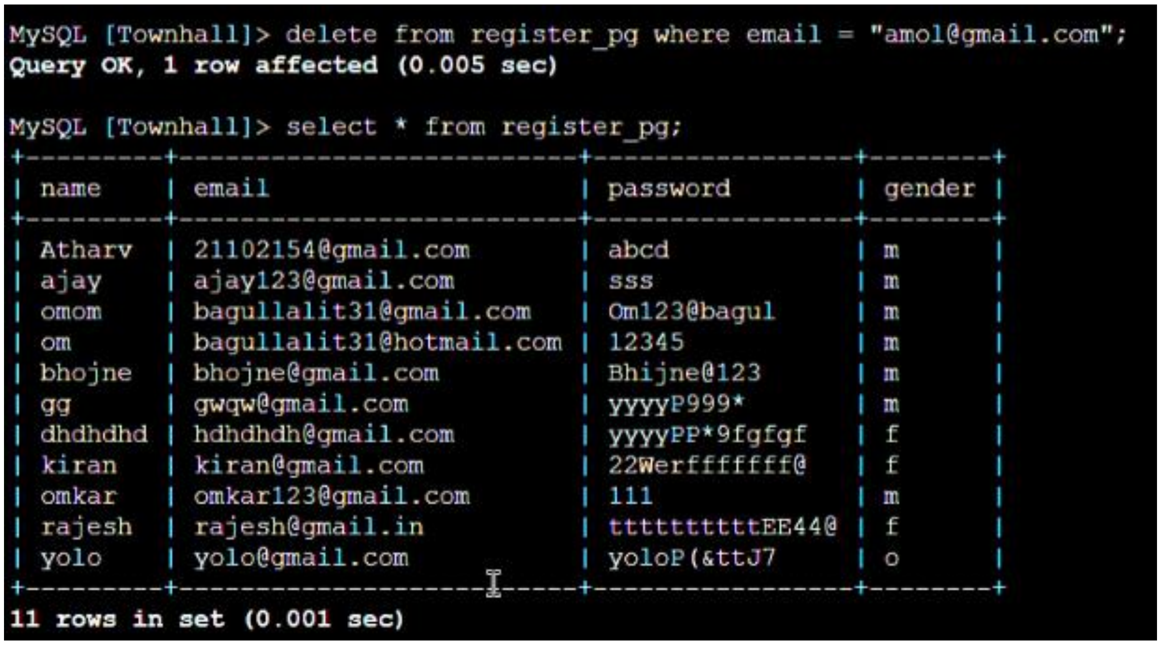
```
MySQL [Townhall]> select * from register_pg;
+---------+--------------------------+-----------------+--------+
| name    | email                    | password        | gender |
+---------+--------------------------+-----------------+--------+
| Atharv  | 21102154@gmail.com       | abcd            | 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.