Description:

Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

Problem Statement:

Company ABC wants to move their product to AWS. They have the following things set up right now:

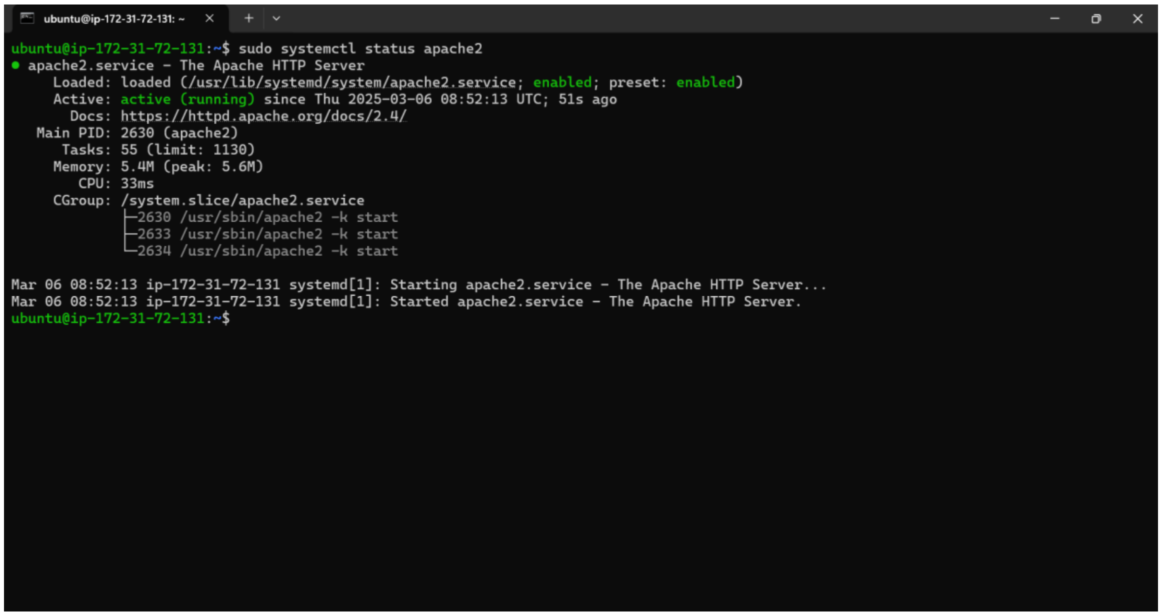
1. MySQL DB
2. Website (PHP)

The company wants high availability on this product, therefore wants Auto Scaling to be enabled on this website

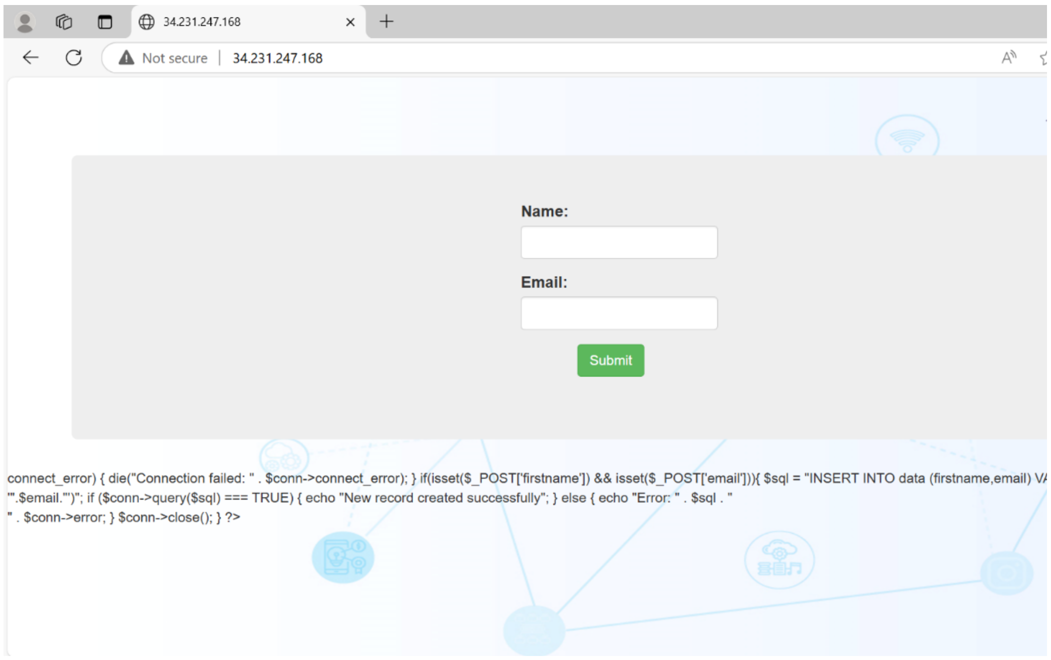
Solution:

Steps To Solve:

1. Launch an EC2 Instance

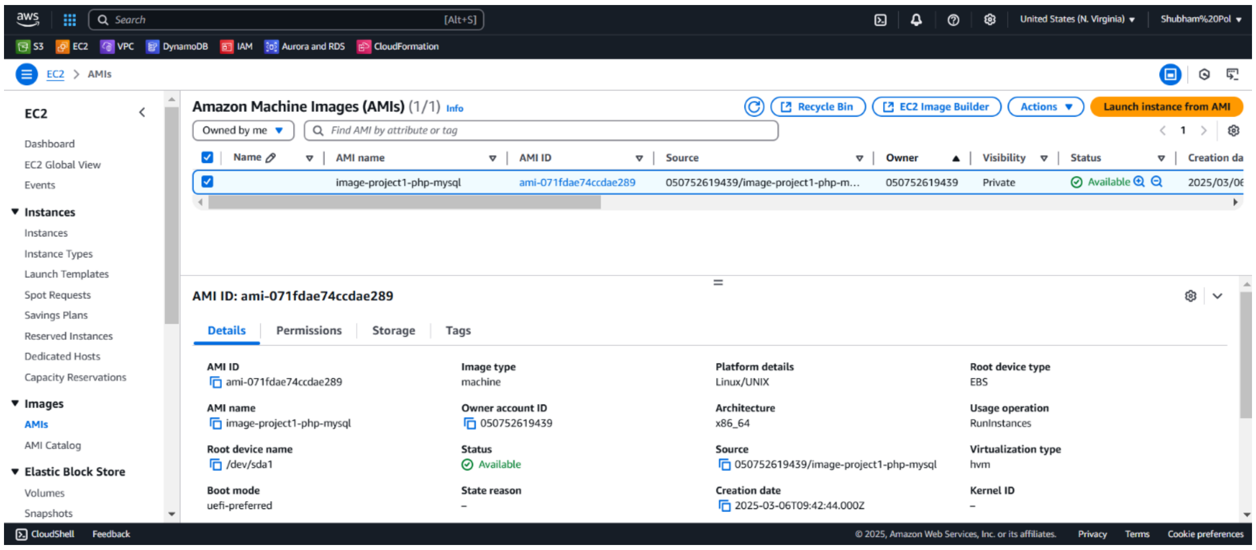


```
ubuntu@ip-172-31-72-131:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-03-06 08:52:13 UTC; 51s ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 2630 (apache2)
      Tasks: 55 (limit: 1130)
     Memory: 5.4M (peak: 5.6M)
        CPU: 33ms
     CGroup: /system.slice/apache2.service
             ├─2630 /usr/sbin/apache2 -k start
             ├─2633 /usr/sbin/apache2 -k start
             └─2634 /usr/sbin/apache2 -k start

Mar 06 08:52:13 ip-172-31-72-131 systemd[1]: Starting apache2.service - The Apache HTTP Server...
Mar 06 08:52:13 ip-172-31-72-131 systemd[1]: Started apache2.service - The Apache HTTP Server.
ubuntu@ip-172-31-72-131:~$
```



```
connect_error) { die("Connection failed: " . $conn->connect_error); } if(isset($_POST['firstname']) && isset($_POST['email'])){ $sql = "INSERT INTO data (firstname,email) VA
'".$email."')"; if ($conn->query($sql) === TRUE) { echo "New record created successfully"; } else { echo "Error: " . $sql . "
" . $conn->error; } $conn->close(); } ?>
```

2. Enable Auto Scaling on these instances (minimum 2)



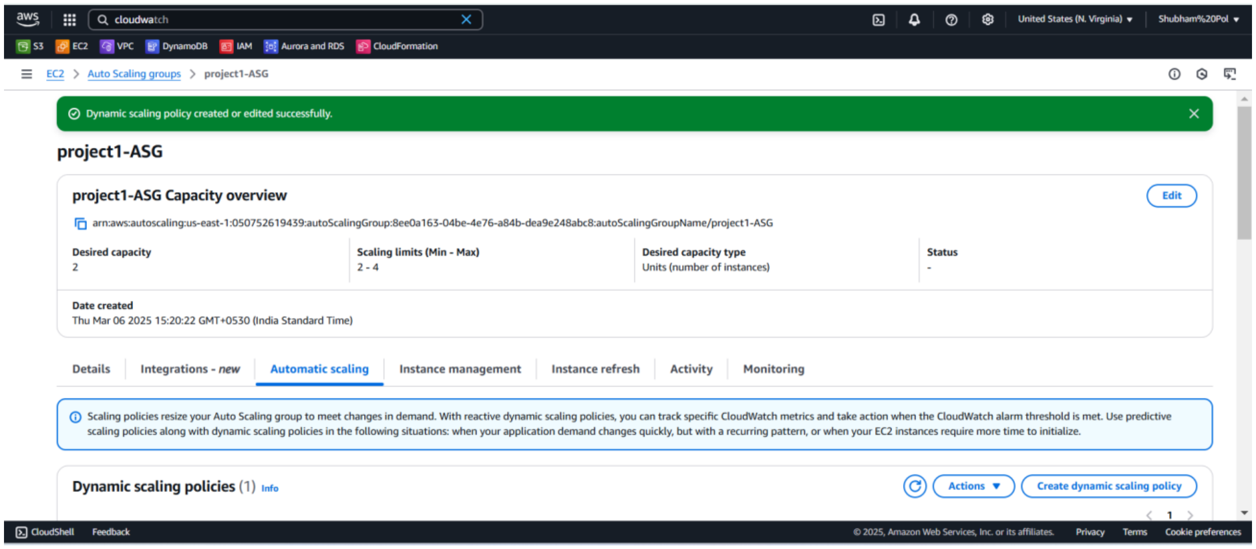
Amazon Machine Images (AMIs) (1/1) Info

Name	AMI name	AMI ID	Source	Owner	Visibility	Status	Creation da
	image-project1-php-mysql	ami-071fdae74ccdae289	050752619439/image-project1-php-m...	050752619439	Private	Available	2025/03/06

AMI ID: ami-071fdae74ccdae289

Details | Permissions | Storage | Tags

AMI ID ami-071fdae74ccdae289	Image type machine	Platform details Linux/UNIX	Root device type EBS
AMI name image-project1-php-mysql	Owner account ID 050752619439	Architecture x86_64	Usage operation RunInstances
Root device name /dev/sda1	Status Available	Source 050752619439/image-project1-php-mysql	Virtualization type hvm
Boot mode uefi-preferred	State reason –	Creation date 2025-03-06T09:42:44.000Z	Kernel ID –



Dynamic scaling policy created or edited successfully.

project1-ASG

project1-ASG Capacity overview

arn:aws:autoscaling:us-east-1:050752619439:autoScalingGroup:8ee0a163-04be-4e76-a84b-dea9e248abc8:autoScalingGroupName/project1-ASG

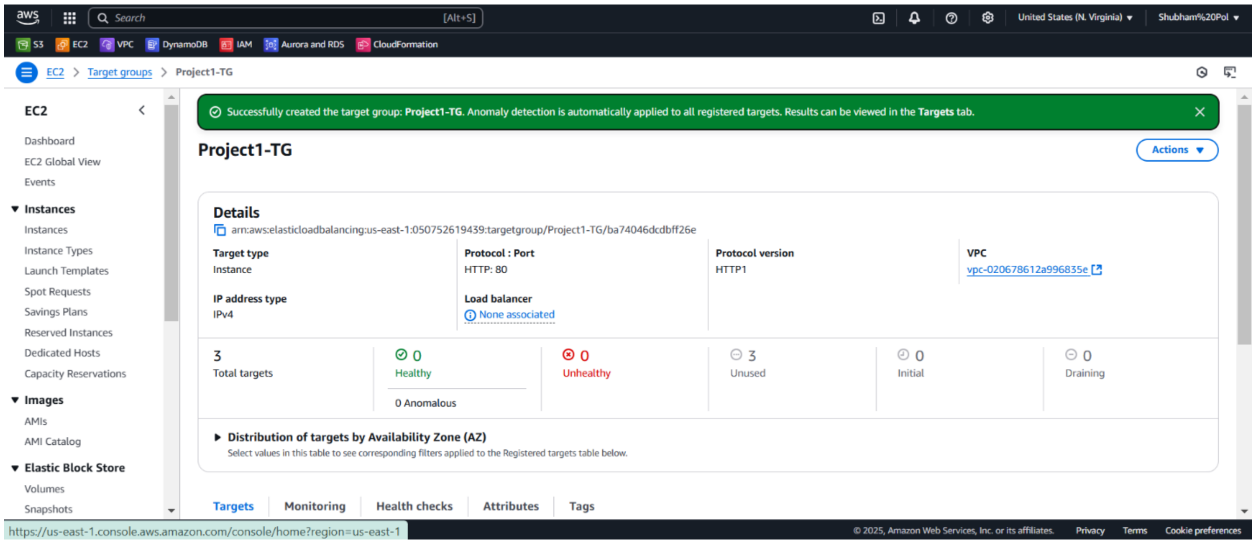
Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
2	2 - 4	Units (number of instances)	-

Date created
Thu Mar 06 2025 15:20:22 GMT+0530 (India Standard Time)

Scaling policies resize your Auto Scaling group to meet changes in demand. With reactive dynamic scaling policies, you can track specific CloudWatch metrics and take action when the CloudWatch alarm threshold is met. Use predictive scaling policies along with dynamic scaling policies in the following situations: when your application demand changes quickly, but with a recurring pattern, or when your EC2 instances require more time to initialize.

Dynamic scaling policies (1) Info

Actions ▾
Create dynamic scaling policy
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



Successfully created the target group: **Project1-TG**. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the **Targets** tab.

Project1-TG

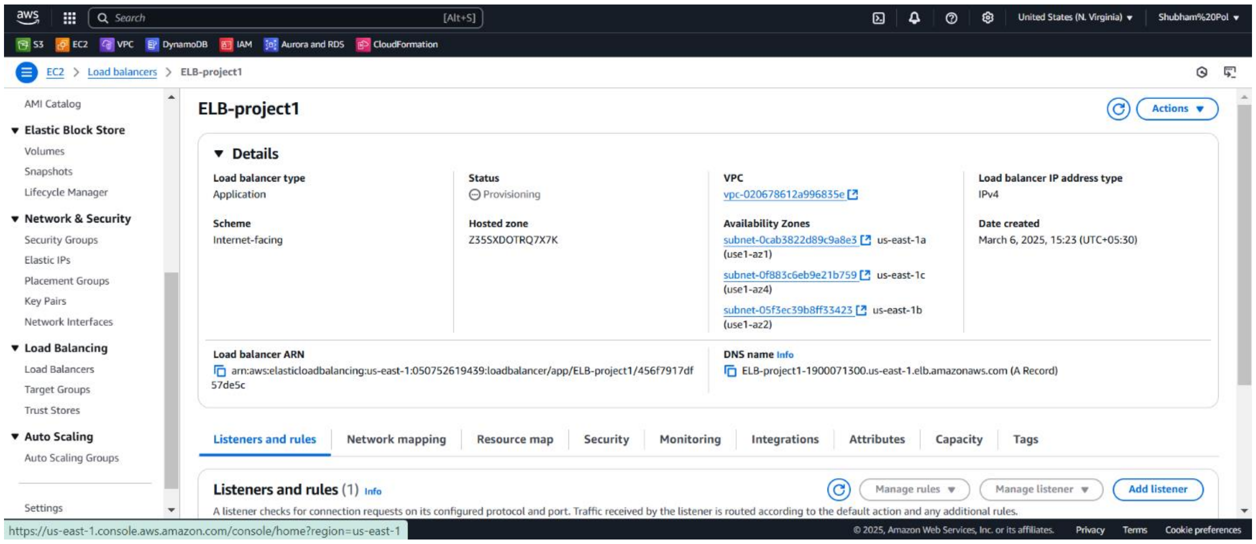
Details
arn:aws:elasticloadbalancing:us-east-1:050752619439:targetgroup/Project1-TG/ba74046dcdbff26e

Target type	Protocol : Port	Protocol version	VPC
Instance	HTTP: 80	HTTP1	vpc-020678612a996835e
IP address type	**Load balancer**		
IPv4	None associated		

3	✓ 0	✗ 0	3	0	0
Total targets	Healthy	Unhealthy	Unused	Initial	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
https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



ELB-project1

▾ **Details**

Load balancer type	Status	VPC	Load balancer IP address type
Application	Provisioning	vpc-020678612a996835e	IPv4
Scheme	**Hosted zone**	**Availability Zones**	**Date created**
Internet-facing	Z35SXDOTRQ7X7K	subnet-0cab3822d89c9a8e3 us-east-1a (use1-az1) subnet-0f883c6eb9e21b759 us-east-1c (use1-az4) subnet-05f3ec39b8ff33423 us-east-1b (use1-az2)	March 6, 2025, 15:23 (UTC+05:30)

Load balancer ARN
arn:aws:elasticloadbalancing:us-east-1:050752619439:loadbalancer/app/ELB-project1/456f7917df57de5c

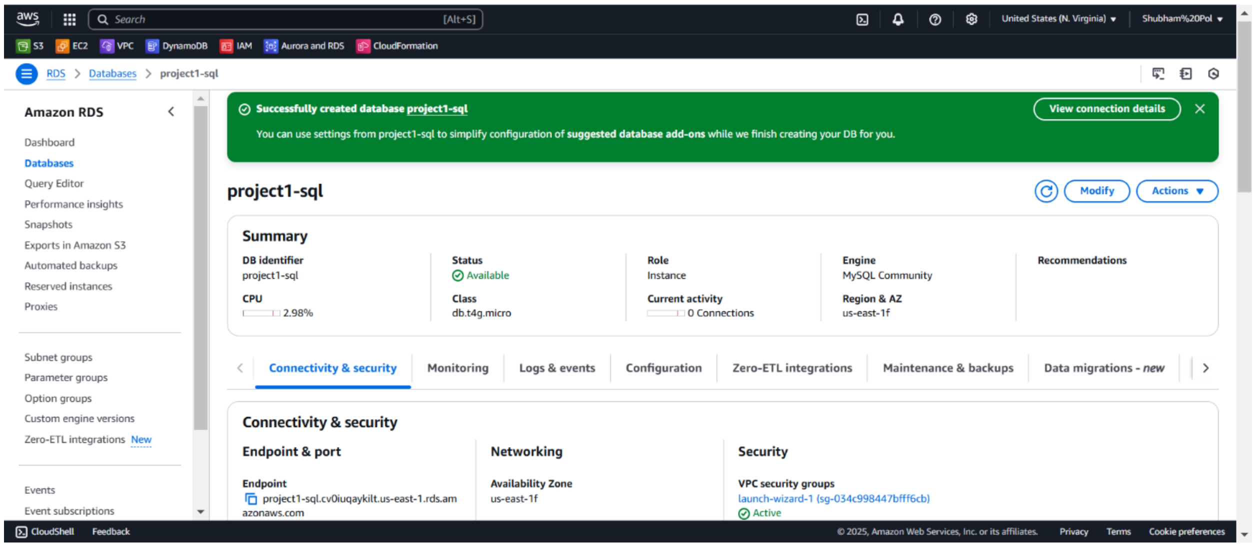
DNS name Info
ELB-project1-1900071300.us-east-1.elb.amazonaws.com (A Record)

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.

https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

3. Create an RDS Instance



Successfully created database project1-sql

You can use settings from project1-sql to simplify configuration of **suggested database add-ons** while we finish creating your DB for you.

project1-sql

Summary

DB identifier	Status	Role	Engine	Recommendations
project1-sql	Available	Instance	MySQL Community	
CPU	**Class**	**Current activity**	**Region & AZ**	
2.98%	db.t4g.micro	0 Connections	us-east-1f	

Connectivity & security

Endpoint & port

Endpoint: project1-sql.cv0iuqaykilt.us-east-1.rds.amazonaws.com

Networking

Availability Zone: us-east-1f

Security

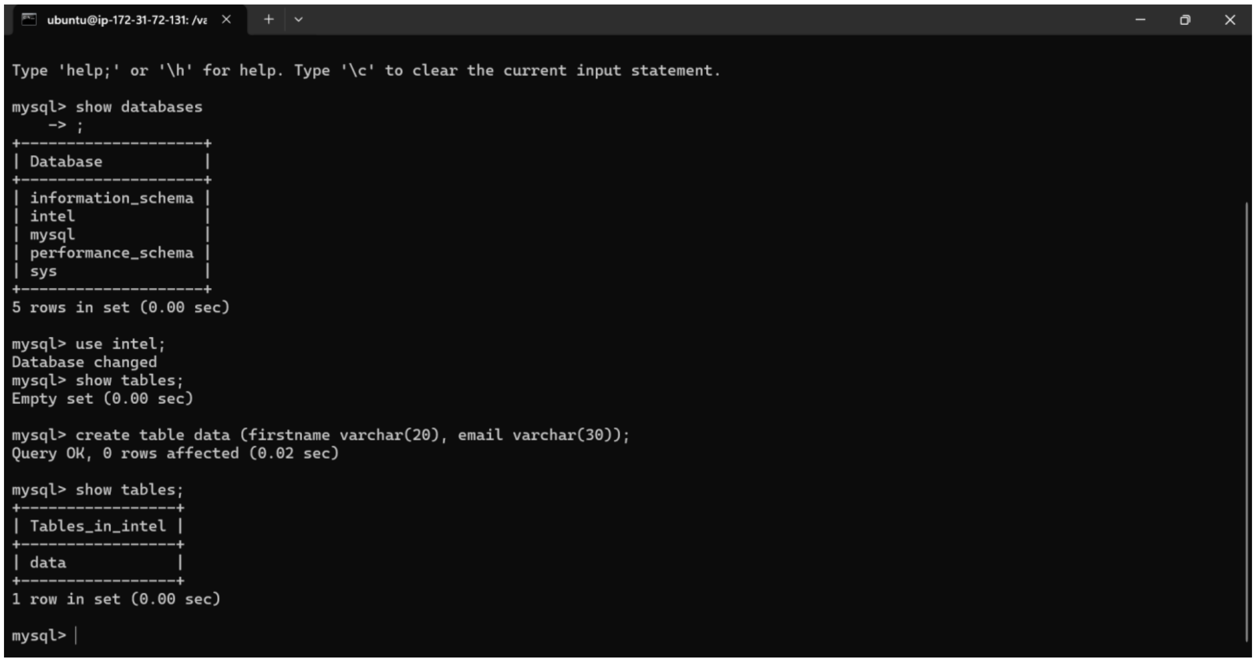
VPC security groups: launch-wizard-1 (sg-034c998447bfff6cb) Active

4. Create Database & Table in RDS instance:

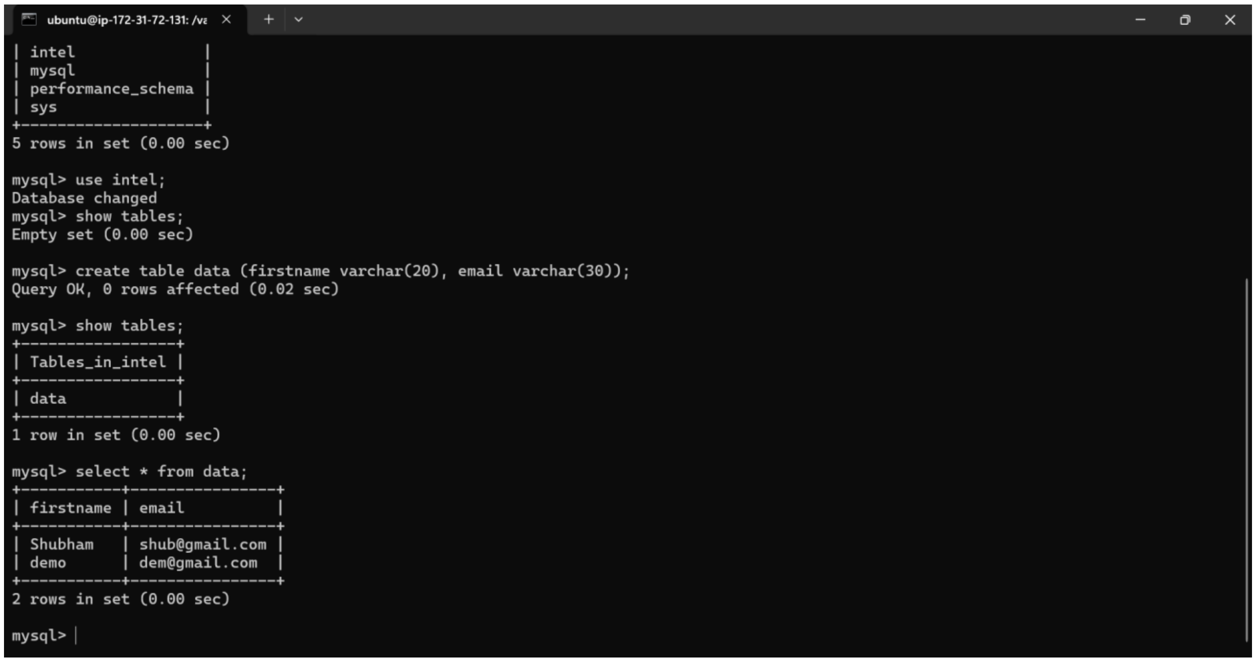
a. Database name: intel

b. Table name: data

c. Database password: intel123

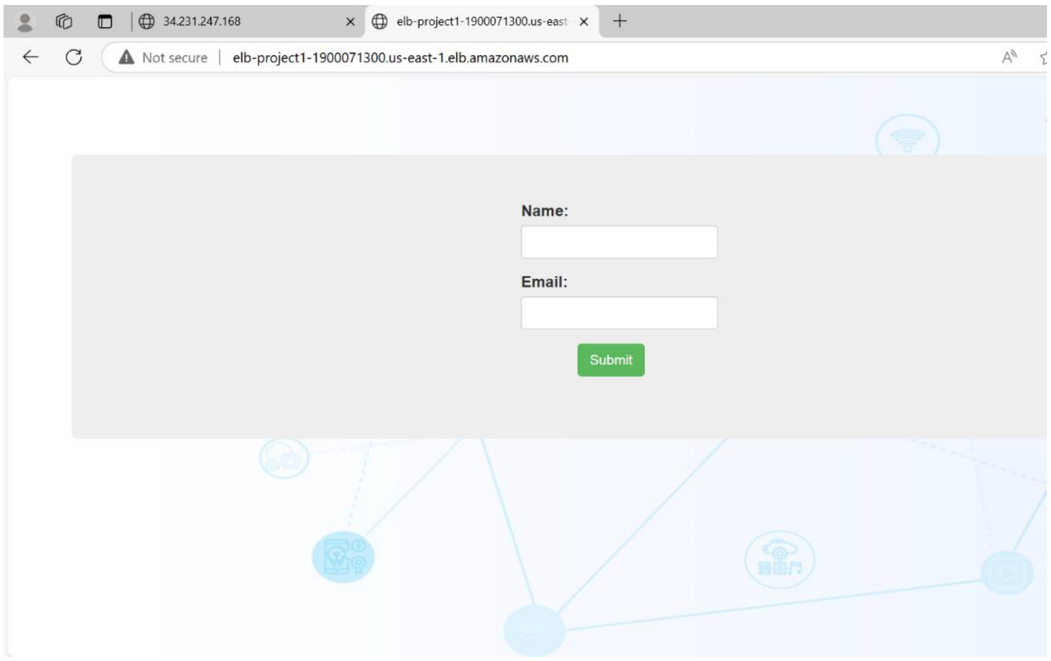


```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases
    -> ;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| intel              |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use intel;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> create table data (firstname varchar(20), email varchar(30));
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----------------+
| Tables_in_intel |
+-----------------+
| data            |
+-----------------+
1 row in set (0.00 sec)

mysql> 
```

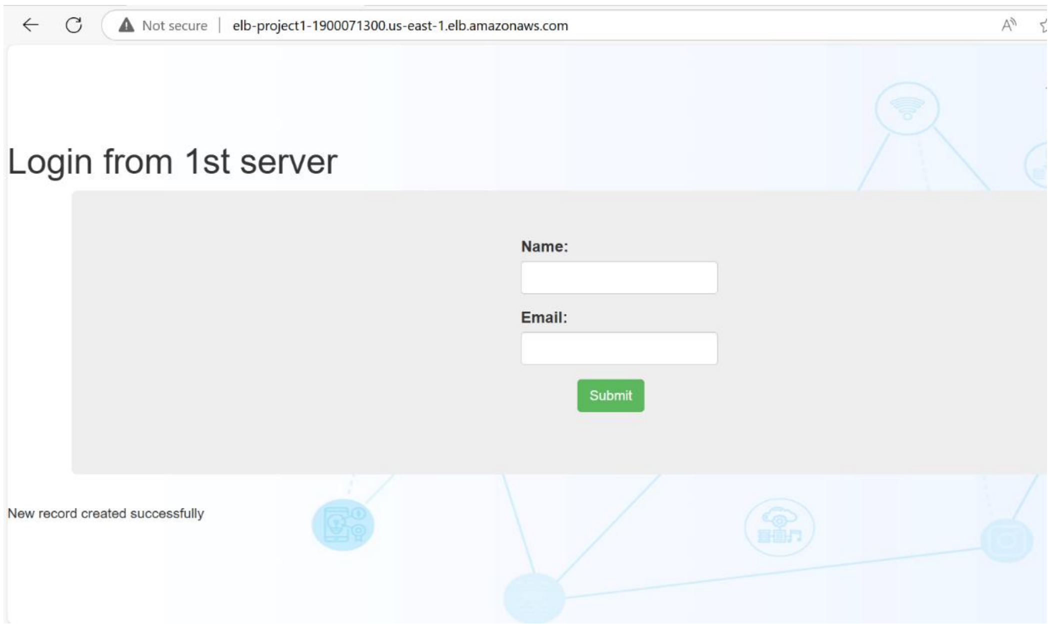


```
| intel              |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)

mysql> use intel;
Database changed
mysql> show tables;
Empty set (0.00 sec)

mysql> create table data (firstname varchar(20), email varchar(30));
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----------------+
| Tables_in_intel |
+-----------------+
| data            |
+-----------------+
1 row in set (0.00 sec)

mysql> select * from data;
+-----------+----------------+
| firstname | email          |
+-----------+----------------+
| Shubham   | shub@gmail.com |
| demo      | dem@gmail.com  |
+-----------+----------------+
2 rows in set (0.00 sec)

mysql> 
```

5. Using ELB hosted the webpage


34.231.247.168
elb-project1-1900071300.us-east
Not secure
elb-project1-1900071300.us-east-1.elb.amazonaws.com
Name:
Email:
Submit

6. Modified the source code to see traffic is served by multiple server

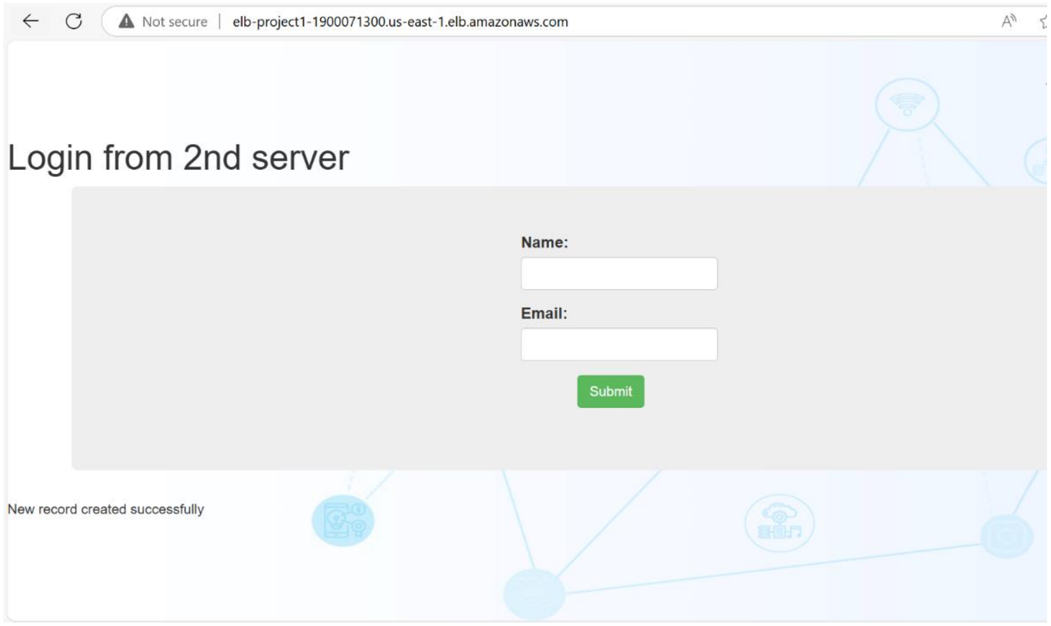


Login from 1st server

Name:

Email:

New record created successfully



Login from 2nd server

Name:

Email:

New record created successfully