EFFICIENT THREE-TIER APPLICATION DEPLOYMENT ON AWS WITH EC2 AND RDS

PROBLEM STATEMENT:

The current infrastructure faces challenges handling increased user traffic, impacting overall performance. Deploying a three-tier architecture on AWS is essential for optimizing scalability, reliability, and resource utilization. The objective is to enhance the user experience by leveraging EC2 for presentation, Apache server for application logic, and MySQL RDS for database storage.

USE CASE - SCENARIO:

The web application experiences fluctuations in user traffic, leading to performance degradation. The deployment of a three-tier architecture on AWS involves utilizing Amazon EC2 instances for the presentation layer, an Apache server for application logic, and MySQL RDS for efficient and secure database storage. This approach ensures scalability and effective data management tailored to the specific technology stack without incorporating S3 or Lambda.

TASKS TO BE PERFORMED:

1. Create a MySQL database on Amazon RDS:
   ➔ Database Name: php
   ➔ Database Password: phpapplication123
   ➔ Table: data
   ➔ Execute table.sql to create the necessary table structure.

2. Connect to the EC2 instance and deploy the application with Apache.
3. Verify the application output to ensure proper functionality.

****

SOLUTION:

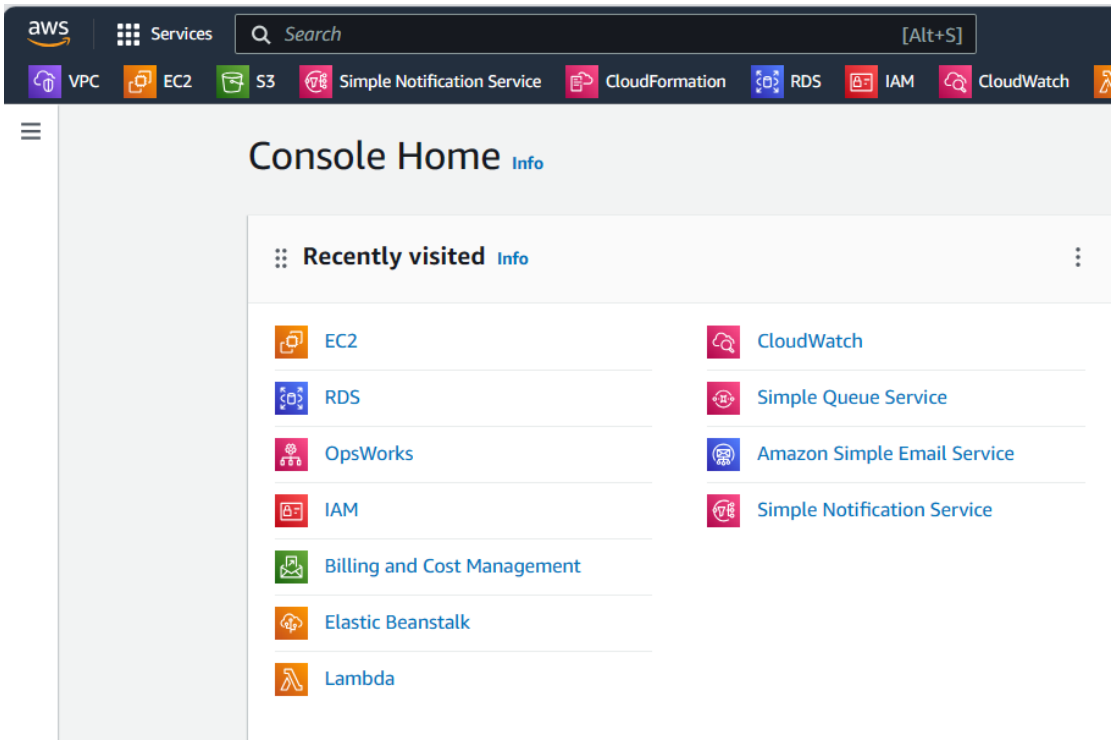
PRE-REQUIREMENTS:

- Cloud : **AWS**
- Server : **EC2**
- Database : **RDS MySQL**

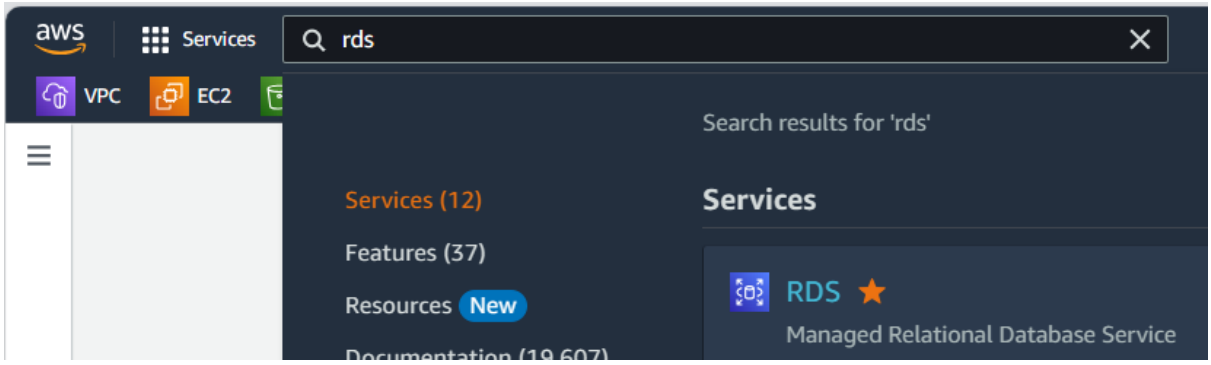
****

STEP:1 – CREATING A MYSQL RDS DATABASE:

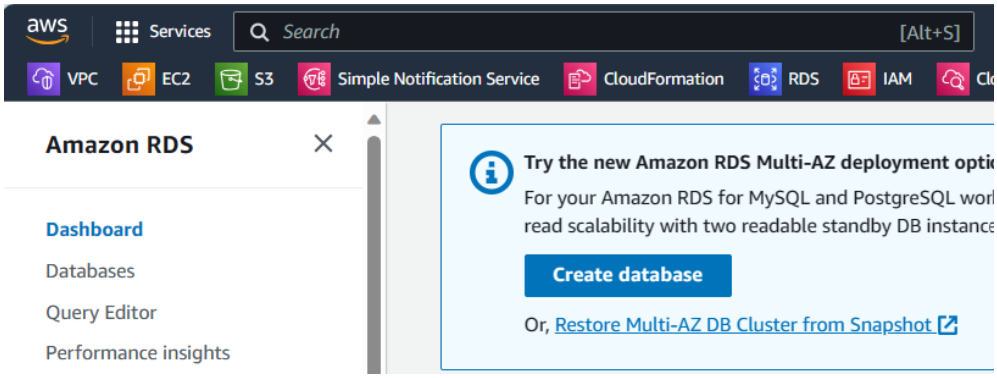
➔ First login into **AWS management console:**



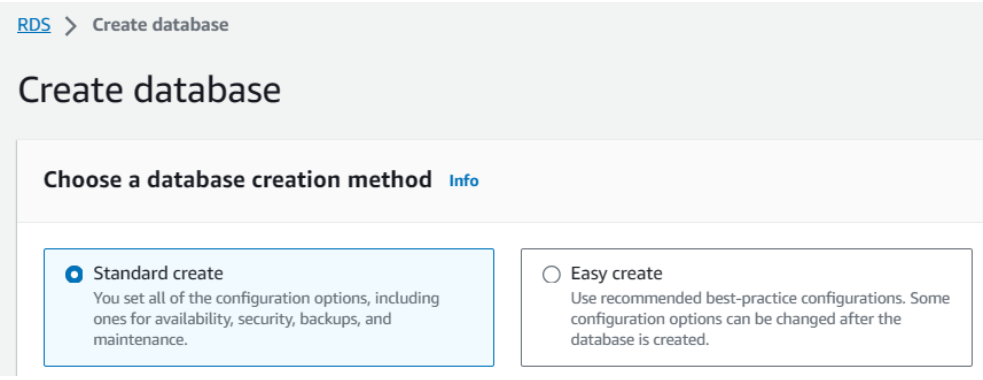
➔ Then search **RDS** on service panel and click that one:



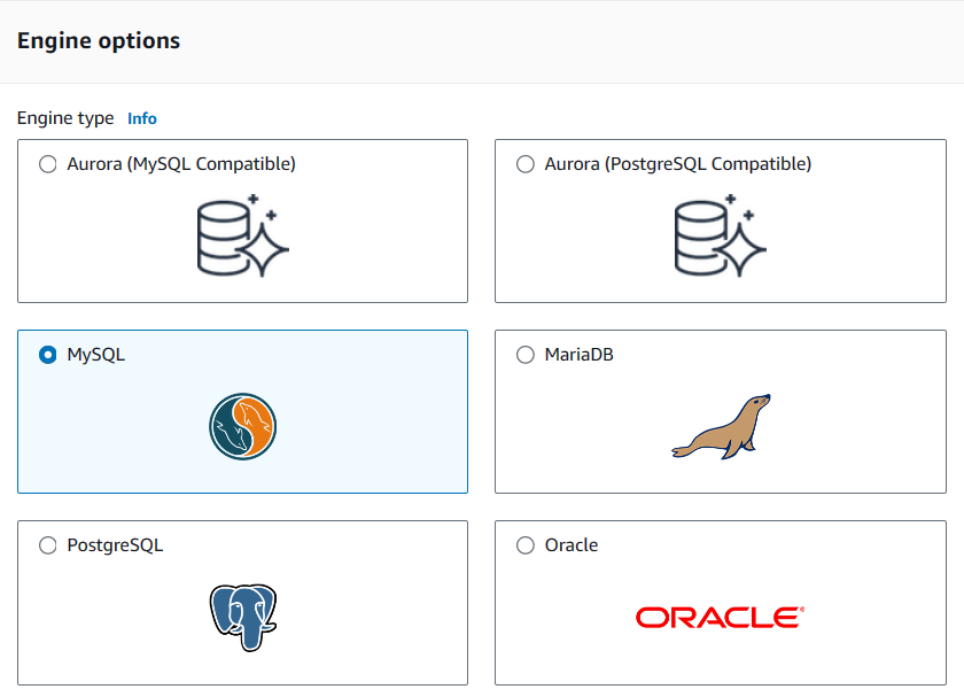
➔ Then click **create database:**



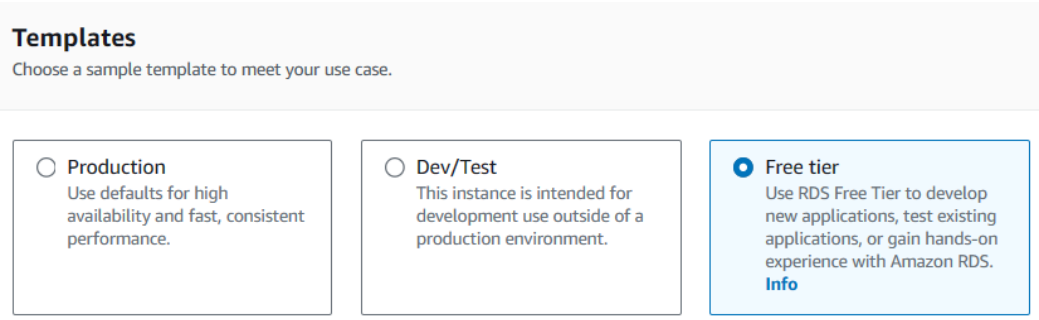
➔ Then select **standard create option:**



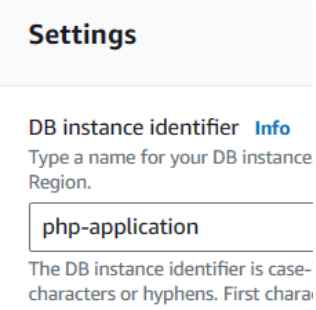
➔ Then select the **database engine as MySQL:**



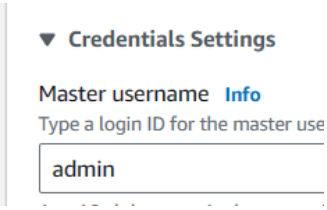
➔ Then select **free tier under templates:**



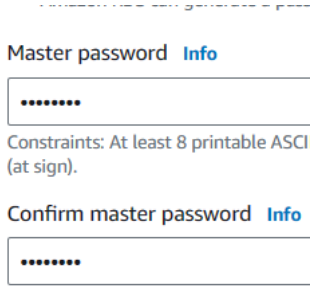
➔ Then name the **database identifier** according to your wish:



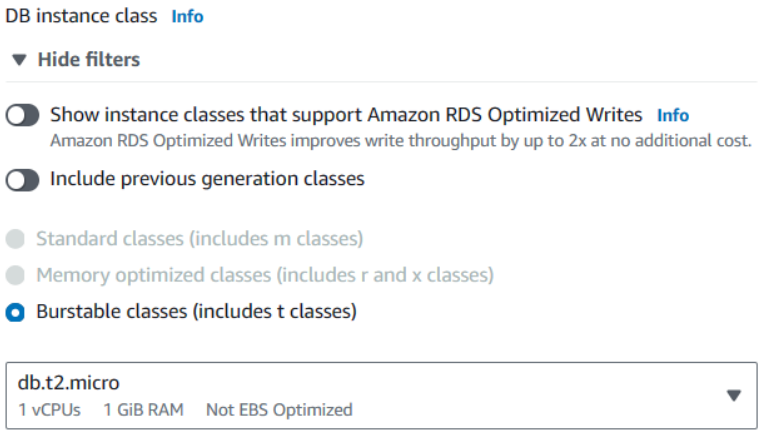
➔ Then **name the database user** according to your wish:



➔ Then set the password for the user, according to the task requirements:



➔ Then select the **instance type:**



➔ Then select the **storage value:**

Storage

Storage type Info

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage Info

20 GiB

The minimum value is 20 GiB and the maximum value is 6,144 GiB

After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes.
Learn more

➔ Then select the **connectivity** according to your wish:

Connectivity Info

Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

- Don't connect to an EC2 compute resource
  Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.
- Connect to an EC2 compute resource
  Set up a connection to an EC2 compute resource for this database.

Virtual private cloud (VPC) Info
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.

Default VPC (vpc-0a095731c8715a9df)
3 Subnets, 3 Availability Zones

Only VPCs with a corresponding DB subnet group are listed.

➔ Then **restrict the public access** for the database:

Public access Info

- Yes
  RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.
- No
  RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

➔ Then create the **security group** for this database:

VPC security group (firewall) Info

Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.

- ○ Choose existing
  Choose existing VPC security groups
- ◉ Create new
  Create new VPC security group

New VPC security group name

php-db

Availability Zone Info

No preference

➔ The select **the database authentication** method as password authentication

Database authentication

Database authentication options Info

- ◉ Password authentication
  Authenticates using database passwords.
- ○ Password and IAM database authentication
  Authenticates using the database password and user credentials through AWS IAM users and roles.
- ○ Password and Kerberos authentication
  Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

➔ Then **name the database: according to the task**

Database options

Initial database name Info

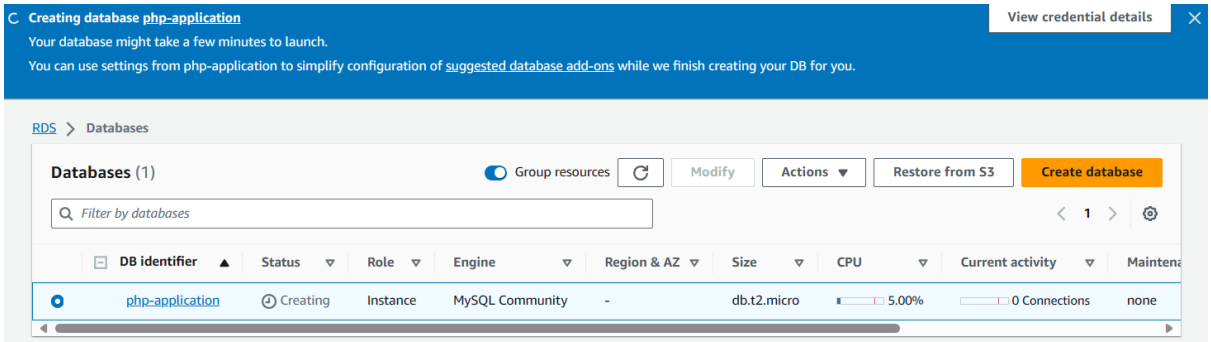
php

If you do not specify a database r

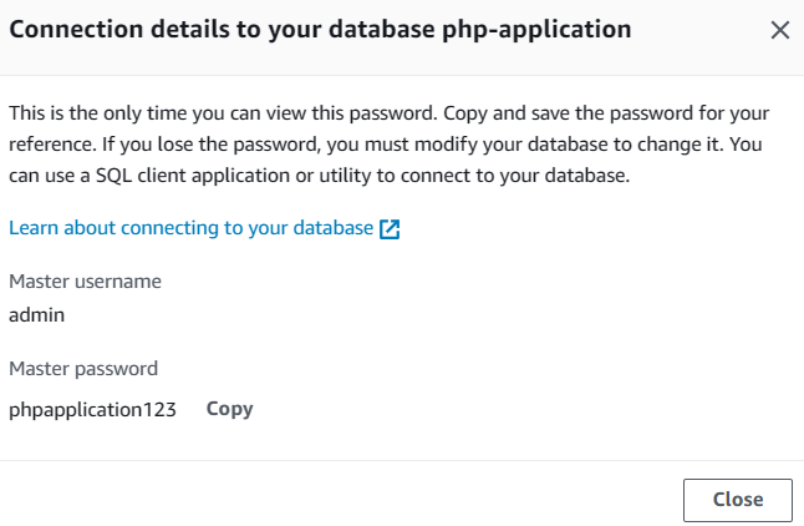
➔ Then click create database:

Cancel | Create database

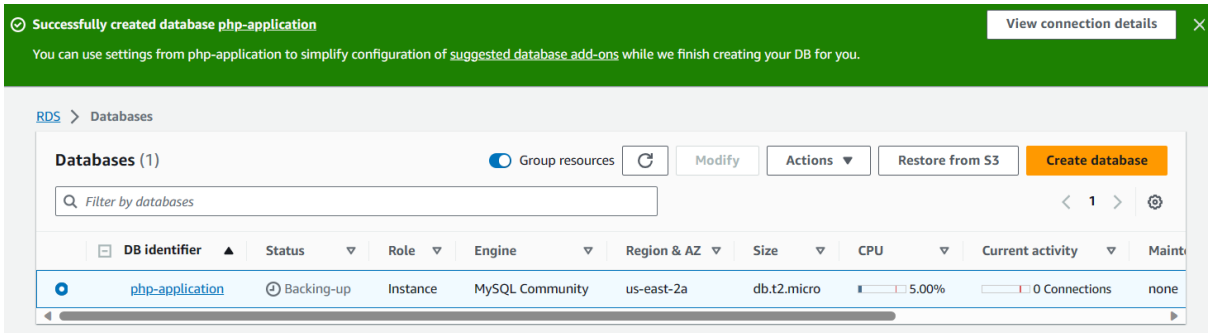
➔ The database creation has been initiated successfully:



➔ The database credentials:



➔ The database has been created successfully:



➔ Let see the database configurations:

DB identifier	Status	Role	Engine	Recommendations
php-application	Backing-up	Instance	MySQL Community	
CPU	Class	Current activity	Region & AZ	
40.33%	db.t2.micro	0 Connections	us-east-2a	

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

Connectivity & security

Endpoint & port

Endpoint
php-application.chttjdyzo3c7.us-east-2.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-2a

VPC
vpc-0a095731c8715a9df

Subnet group
default

Subnets
subnet-0eb1dcebaaca4e0c7
subnet-0cf8bb1ef73391f12
subnet-00ff1d289f674670e

Security

VPC security groups
php-db (sg-01048f1d4ecc5701c)
Active

Publicly accessible
No

Certificate authority Info
rds-ca-2019

Certificate authority date
August 22, 2024, 22:38 (UTC+05:30)

DB instance certificate expiration date

Instance

Configuration

DB instance ID
php-application

Engine version
8.0.35

DB name
php

License model
General Public License

Option groups
default:mysql-8-0 In sync

Amazon Resource Name (ARN)
arn:aws:rds:us-east-2:619355063360:db:php-application

Resource ID
db-OJVPDO4RVPEJMBN7MNESXR34VM

Created time
December 29, 2023, 12:49 (UTC+05:30)

DB instance parameter group

Instance class

Instance class
db.t2.micro

vCPU
1

RAM
1 GB

Availability

Master username
admin

Master password
****

IAM DB authentication
Not enabled

Multi-AZ
No

Secondary Zone
-

Storage

Encryption
Not enabled

Storage type
General Purpose SSD (gp2)

Storage
20 GiB

Provisioned IOPS
-

Storage throughput
-

Storage autoscaling
Disabled

Storage file system configuration
Current

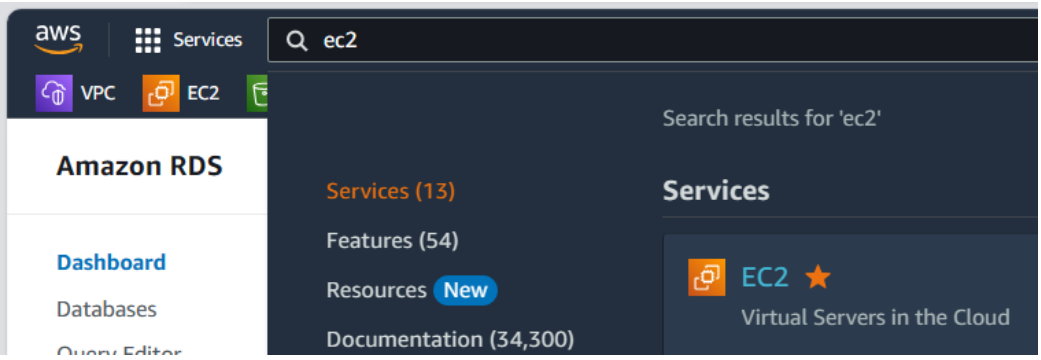
Performance Insights

Performance Insights enabled
Turned off

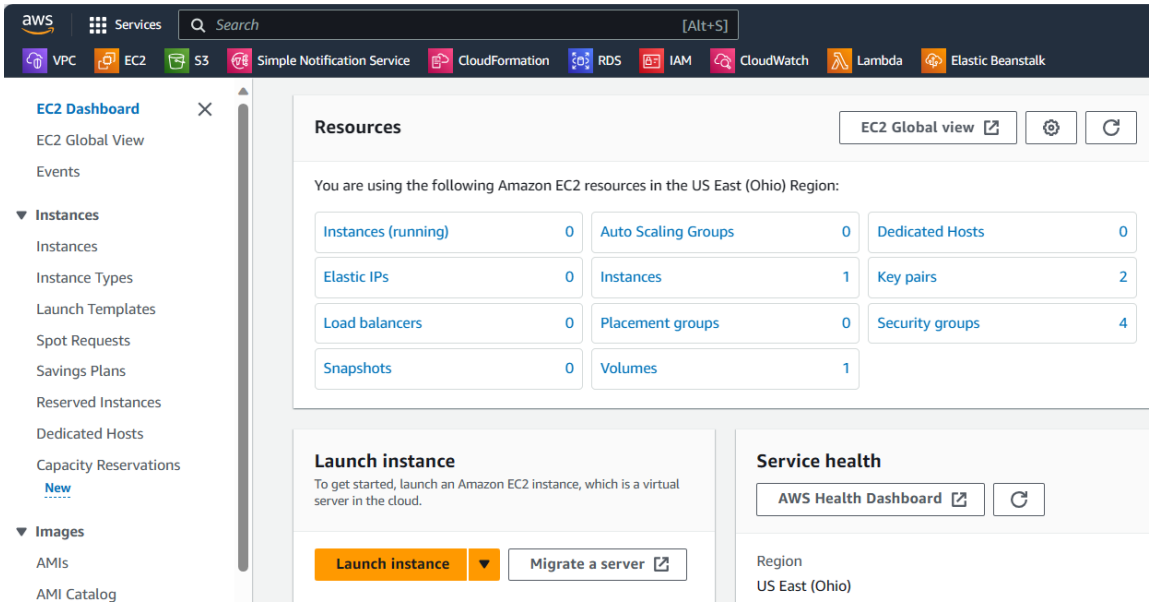
****

STEP:2 – CREATING AN EC2 INSTANCE:

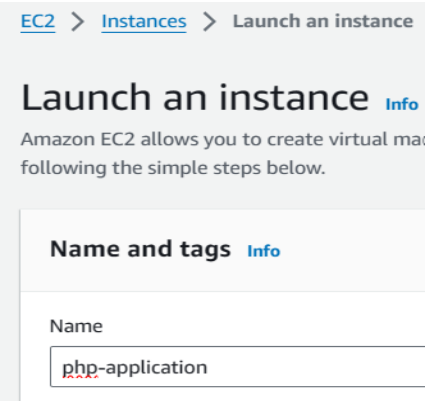
➔ On the service panel search **EC2**, and click that one:



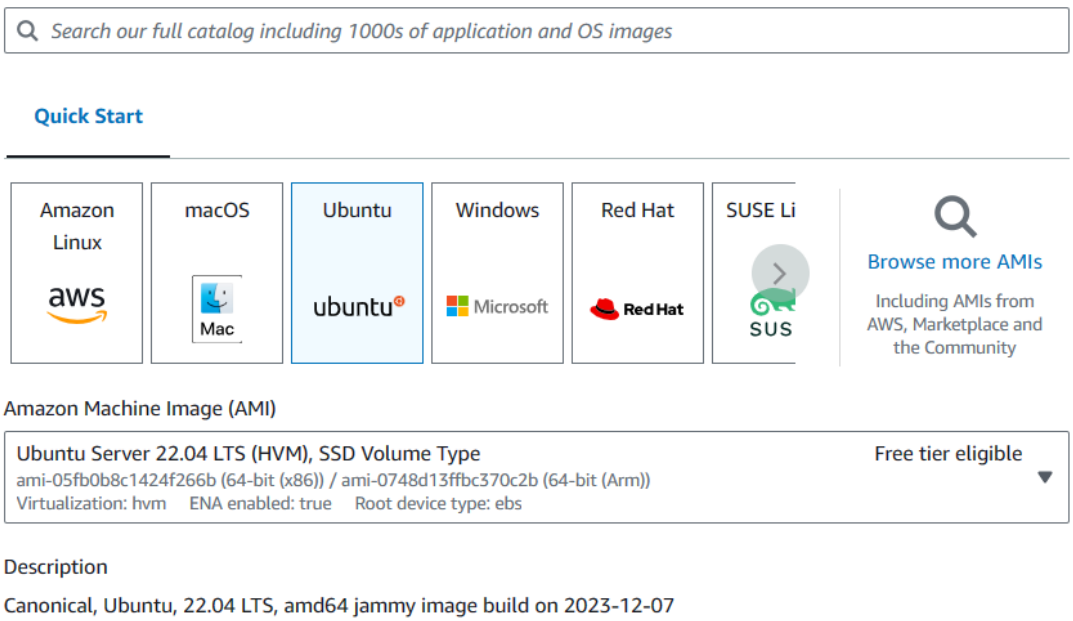
➔ Then click **launch instance** option:



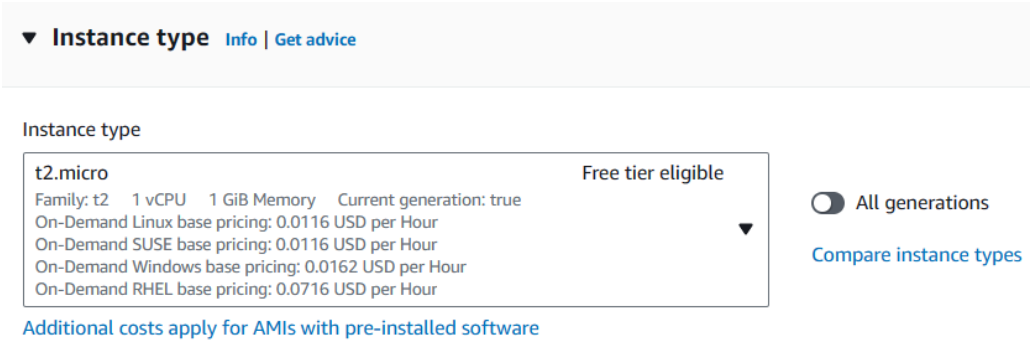
➔ Then name the instance according to your wish:



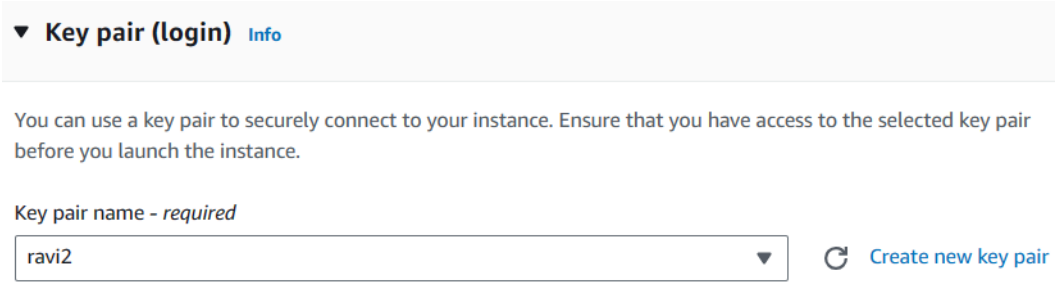
➔ Then selecting **os** according to your preferences, here I am using **ubuntu os:**



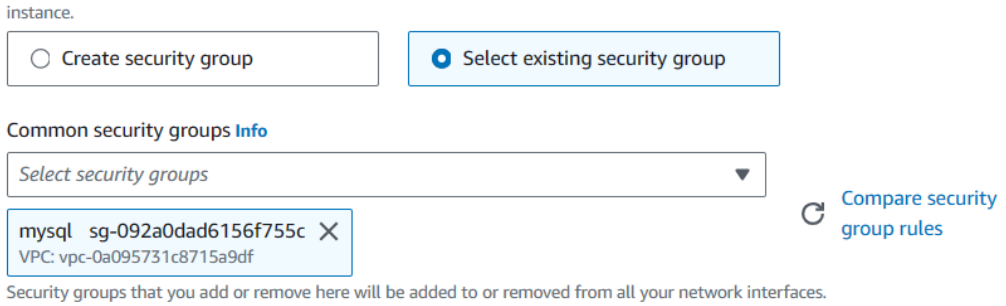
➔ Then select the **instance type:**



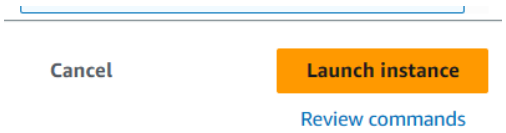
➔ Then select the **keypair** according to your wish:



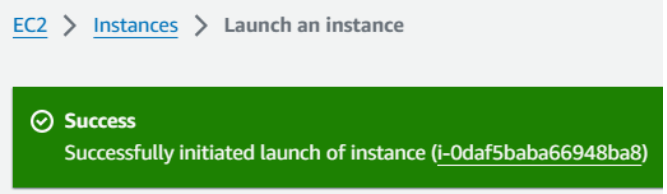
➔ Then select the security group or create a new security group:



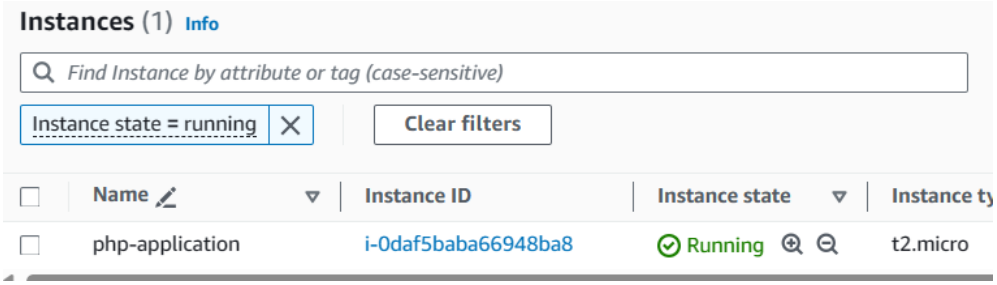
➔ Then click launch instance option:



➔ The instance launching has been initiated successfully:

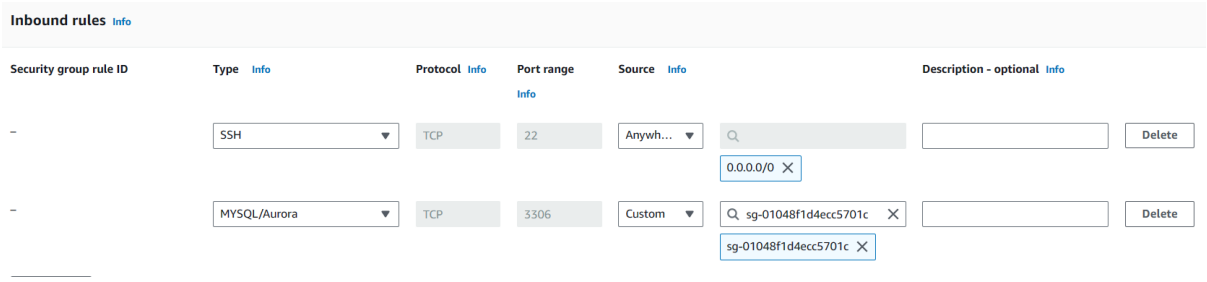


➔ The instance had created successfully:



➔ Then alerting the security group of both instance and db instance:

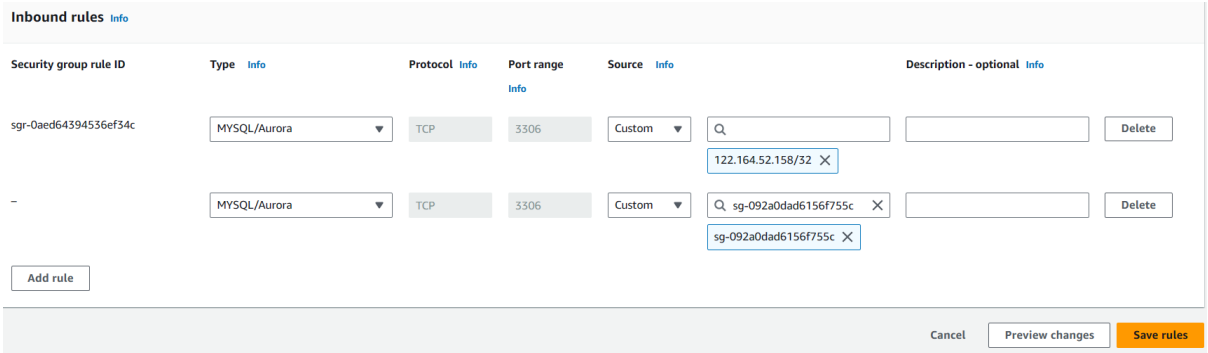
Instance security group: php-Db security group as database traffic



Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
-	SSH	TCP	22	Anywh...	0.0.0.0/0		Delete
-	MYSQL/Aurora	TCP	3306	Custom	sg-01048f1d4ecc5701c sg-01048f1d4ecc5701c		Delete

Database security group: MySQL security group as server traffic to connect this database:



Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-0aed64394536ef34c	MYSQL/Aurora	TCP	3306	Custom	122.164.52.158/32		Delete
-	MYSQL/Aurora	TCP	3306	Custom	sg-092a0dad6156f755c sg-092a0dad6156f755c		Delete

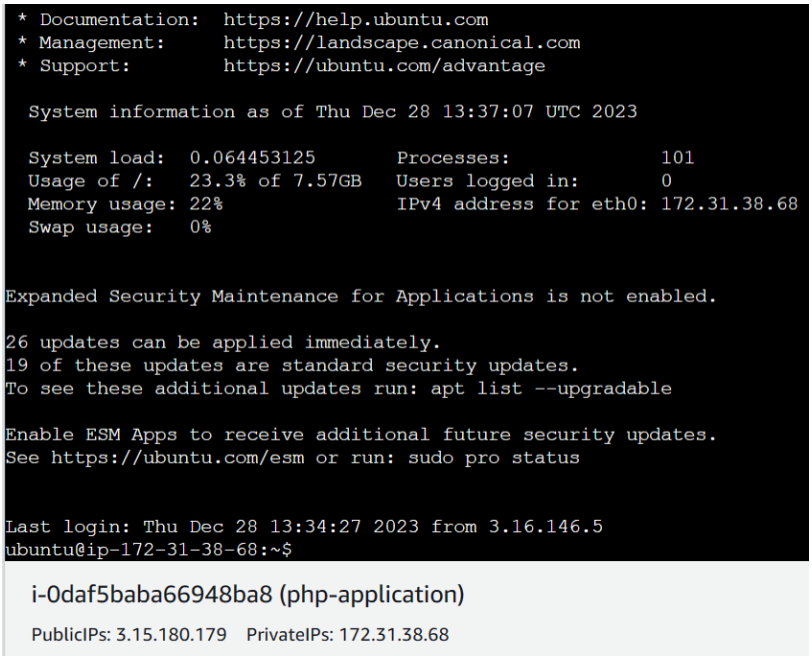
Add rule

Cancel | Preview changes | Save rules

****

STEP:3 – INSTALLING REQUIRED SOFTWARE OR DEPENDENCIES:

➔ Connecting the instance with **instance connect:**



```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Thu Dec 28 13:37:07 UTC 2023

  System load:  0.064453125       Processes:             101
  Usage of /:   23.3% of 7.57GB   Users logged in:       0
  Memory usage: 22%               IPv4 address for eth0: 172.31.38.68
  Swap usage:   0%


Expanded Security Maintenance for Applications is not enabled.

26 updates can be applied immediately.
19 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


Last login: Thu Dec 28 13:34:27 2023 from 3.16.146.5
ubuntu@ip-172-31-38-68:~$
```

i-0daf5baba66948ba8 (php-application)

PublicIPs: 3.15.180.179 PrivateIPs: 172.31.38.68

➔ Creating a script for installing dependencies required for this task:

Script contains:

```
#!/bin/bash
```

```
#updating the os:
apt-get update
#installing apache2:
apt-get install -y apache2
#installing php:
apt-get install php -y
#installing mysqlclient:
apt install mysql-client-core-8.0 -y
```

➔ After entering the script**, changing the file permission, and executing it with the command:**

```
vi package.sh
chmod +x package.sh
./package.sh
```

```
ubuntu@ip-172-31-38-68:~$ sudo su
root@ip-172-31-38-68:/home/ubuntu# vi package.sh
root@ip-172-31-38-68:/home/ubuntu# chmod +x package.sh
root@ip-172-31-38-68:/home/ubuntu# ./package.sh
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
```

➔ Checking whether the packages are installed or not:

```
root@ip-172-31-38-68:/home/ubuntu# apache2 --version
[Thu Dec 28 13:47:35.010692 2023] [core:warn] [pid 9636] AH00111: Config variable
apache2: Syntax error on line 80 of /etc/apache2/apache2.conf: DefaultRuntimeDir m
root@ip-172-31-38-68:/home/ubuntu# mysql --version
mysql  Ver 8.0.35-0ubuntu0.22.04.1 for Linux on x86_64 ((Ubuntu))
root@ip-172-31-38-68:/home/ubuntu# php --version
PHP 8.1.2-1ubuntu2.14 (cli) (built: Aug 18 2023 11:41:11) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.14, Copyright (c), by Zend Technologies
root@ip-172-31-38-68:/home/ubuntu#
```

All the packages have been installed successfully.

****

STEP:4 - CONNECTING THE DATABASE AND CREATING TABLE:

➔ Connecting the database with the command:

```
mysql -u admin -p php -h php-application.chttjdyzo3c7.us-east-
2.rds.amazonaws.com
```

➔ After entering the command, **enter the database password:**

```
root@ip-172-31-38-68:/home/ubuntu# mysql -u admin -p php -h php-application.chttjdyzo3c7.us-east-2.rds.amazonaws.com
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 24
Server version: 8.0.35 Source distribution

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

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

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

mysql>
```

➔ Then we need to create table, for this application by using SQL query:

```sql
CREATE DATABASE php;
USE php;
CREATE TABLE data (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) NOT NULL,
    password VARCHAR(255) NOT NULL
);
```

➔ Executing the SQL query and checking the output:

```
mysql> CREATE DATABASE php;
ERROR 1007 (HY000): Can't create database 'php'; database exists
mysql>
mysql> USE php;
Database changed
mysql> CREATE TABLE data (
    ->     id INT AUTO_INCREMENT PRIMARY KEY,
    ->     username VARCHAR(255) NOT NULL,
    ->     password VARCHAR(255) NOT NULL
    -> );
Query OK, 0 rows affected (0.02 sec)

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

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

mysql>
```

Table have created successfully:

****

STEP:5 – DEPLOYING THE APPLICATION ON APACHE SERVER:

➔ Cloning the application, the **on /var/www/html/** location:

```
root@ip-172-31-38-68:/home/ubuntu# cd /var/www/
root@ip-172-31-38-68:/var/www# git clone https://github.com/Ravivarman16/php-application.git
Cloning into 'php-application'...
Username for 'https://github.com': Ravivarman16
Password for 'https://Ravivarman16@github.com':
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 26 (delta 14), reused 10 (delta 7), pack-reused 0
Receiving objects: 100% (26/26), 58.40 KiB | 3.89 MiB/s, done.
Resolving deltas: 100% (14/14), done.
root@ip-172-31-38-68:/var/www# ls
html  php-application
root@ip-172-31-38-68:/var/www# cd php-application/
root@ip-172-31-38-68:/var/www/php-application# ls
index.php  loginform.php  loginprocess.php  php.jpg  register.php  registration_success.php  success.php  table.sql
root@ip-172-31-38-68:/var/www/php-application# mv * /var/www/html/
root@ip-172-31-38-68:/var/www/php-application# ls
root@ip-172-31-38-68:/var/www/php-application# cd ..
root@ip-172-31-38-68:/var/www# cd html/
root@ip-172-31-38-68:/var/www/html# ls
index.html  index.php  loginform.php  loginprocess.php  php.jpg  register.php  registration_success.php  success.php  table.sql
root@ip-172-31-38-68:/var/www/html#
```

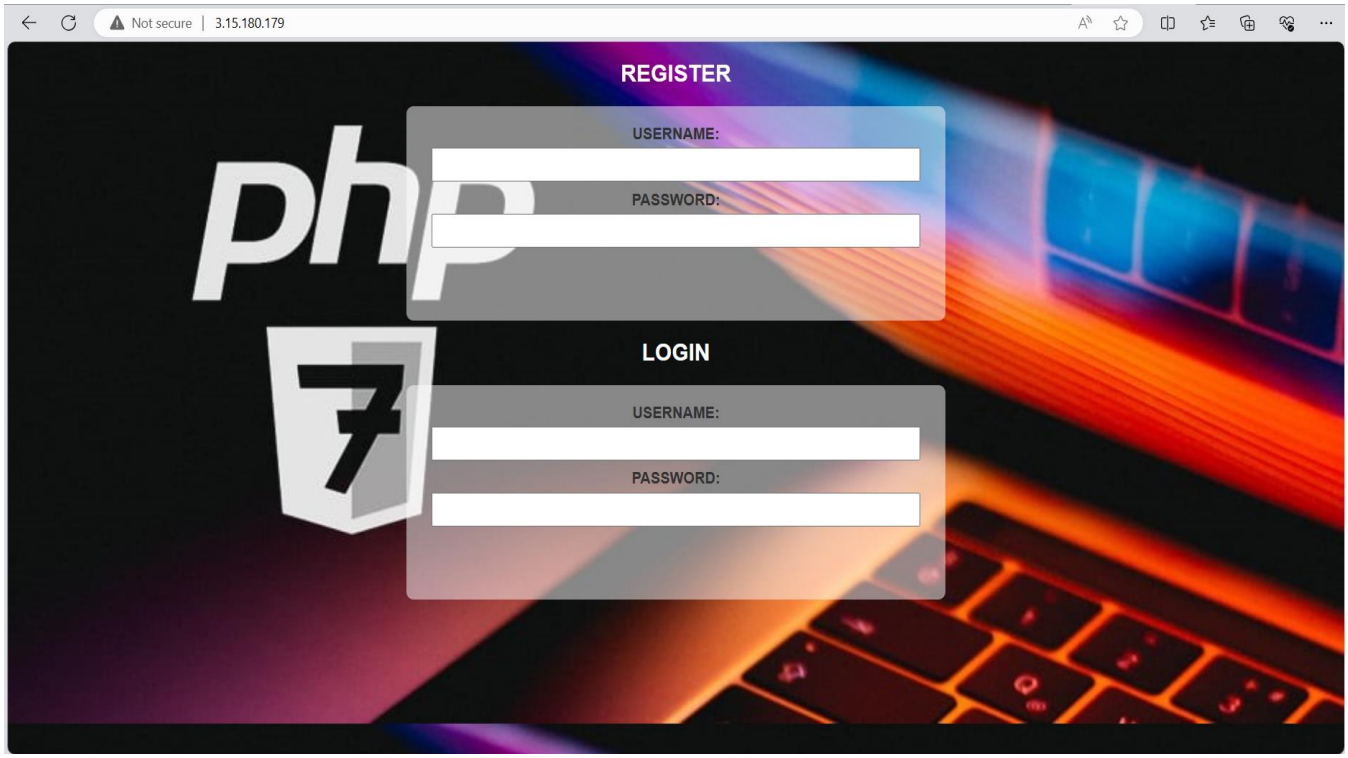
➔ Adding the database configurations on **loginprocess.php & register.php files:**

```
$host = 'php-application.chttjdyzo3c7.us-east-2.rds.amazonaws.com';
$username = 'admin';
$password = 'phpapplication123';
$database = 'php';
```

➔ Just replace the above lines in both files and save it.

```
root@ip-172-31-38-68:/var/www/html# vi register.php
root@ip-172-31-38-68:/var/www/html# vi loginprocess.php
root@ip-172-31-38-68:/var/www/html#
```

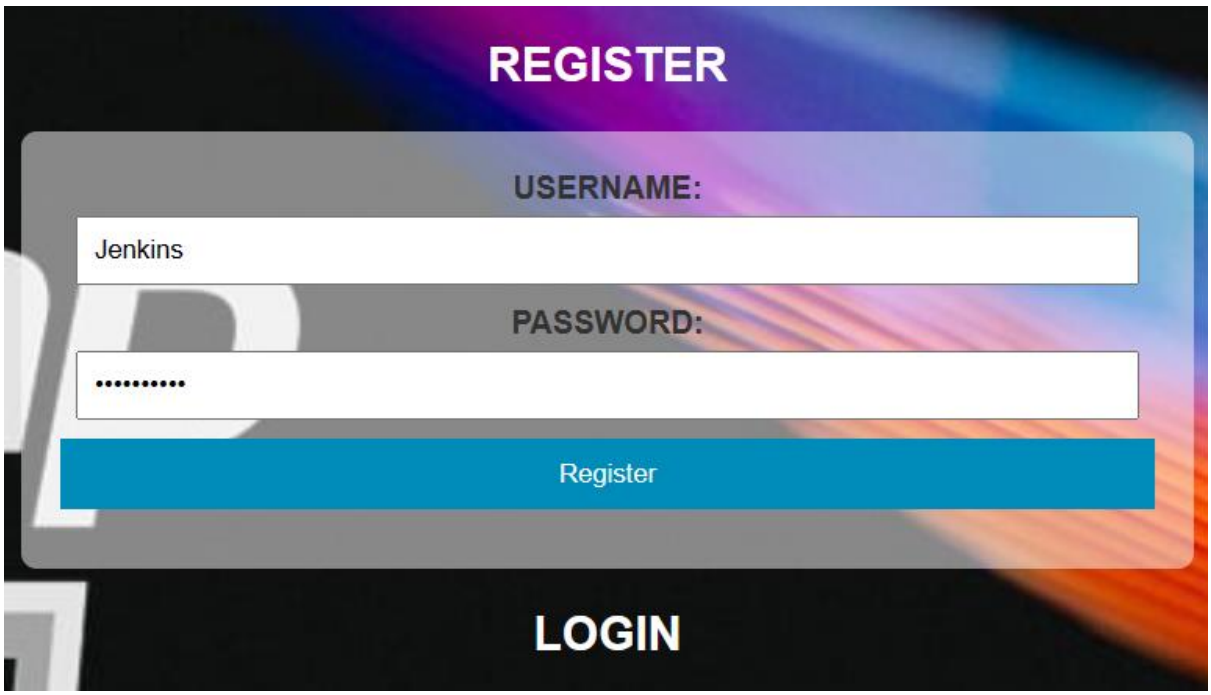
➔ Checking the output on the browser:



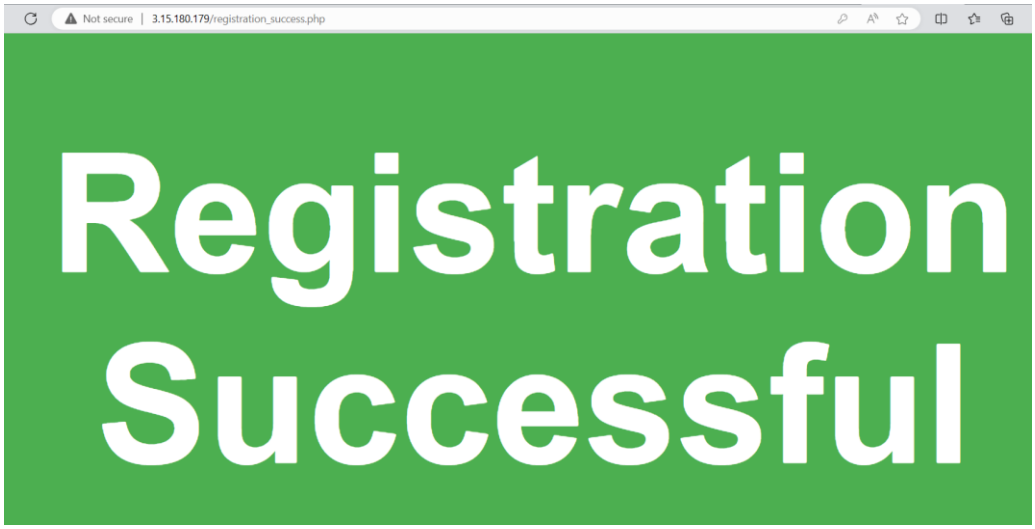
➔ Then checking the application by registering.

Username: Jenkins

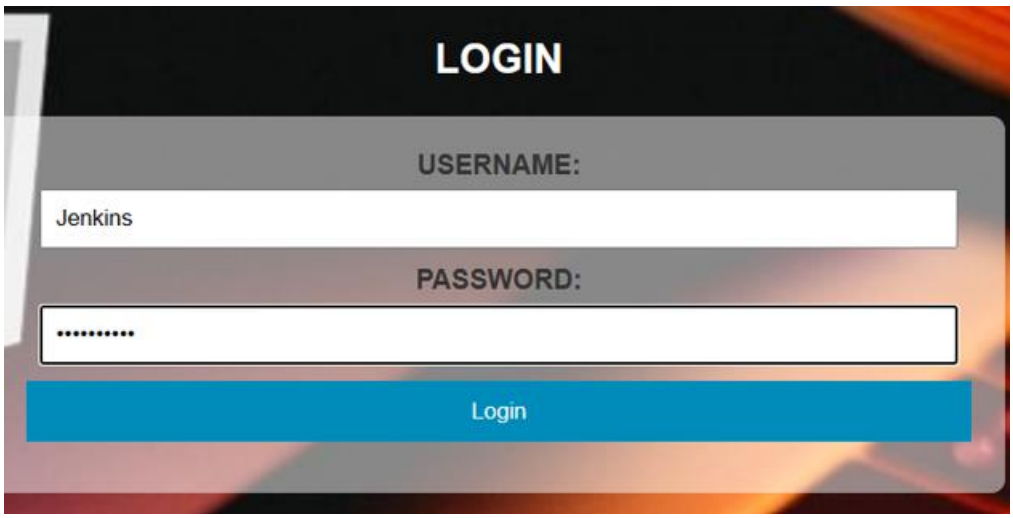
Password: jenkins123



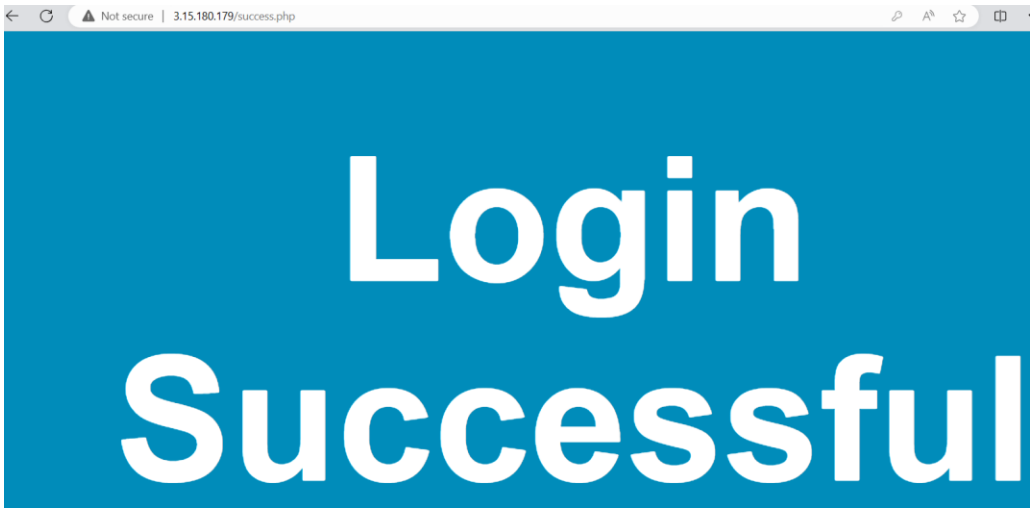
OUTPUT OF REGISTRATION:



➔ Now trying to login with those credentials:



➔ We can able to login with those credentials:



****

BENEFITS OF DOING ABOVE TASK:

- ➔ **Scalability with EC2:** Horizontally scale the presentation layer using Amazon EC2 instances to handle varying user loads.

- ➔ **Application Logic with Apache Server:** Leverage the Apache server for application logic, ensuring efficient processing of user requests.

- ➔ **Reliable Database Storage with MySQL RDS:** Ensure high availability and data integrity using MySQL RDS for efficient and secure database storage.

- ➔ **Resource Optimization:** Efficiently manage resources with a dedicated architecture for presentation, application, and database layers.

- ➔ **Cost-Effective Solution:** Pay only for the utilized resources, optimizing costs based on actual demand.

****

GitHub repository: **https://github.com/Ravivarman16/php-application.git**

All the required files and codes has been uploaded in above GitHub repository as a reference.

********** **THE – END** **********