Project 1: Deploying 2-Tier Applicaiton on AWS

Deploy a 2-tier application with the following requirements

- Create custom VPC with public and private subnets.
- Deploy the frontend UI application in the public subnet.
- Deploy the backend database (AWS RDS) in a private subnet.
- Establish a connection between the front-end and the back-end.
- Demonstrate the database CRUD operation through the front end.

This Document Covers the follwoing sections:

1. Archtitctural Diagram

2. Creating a VPC

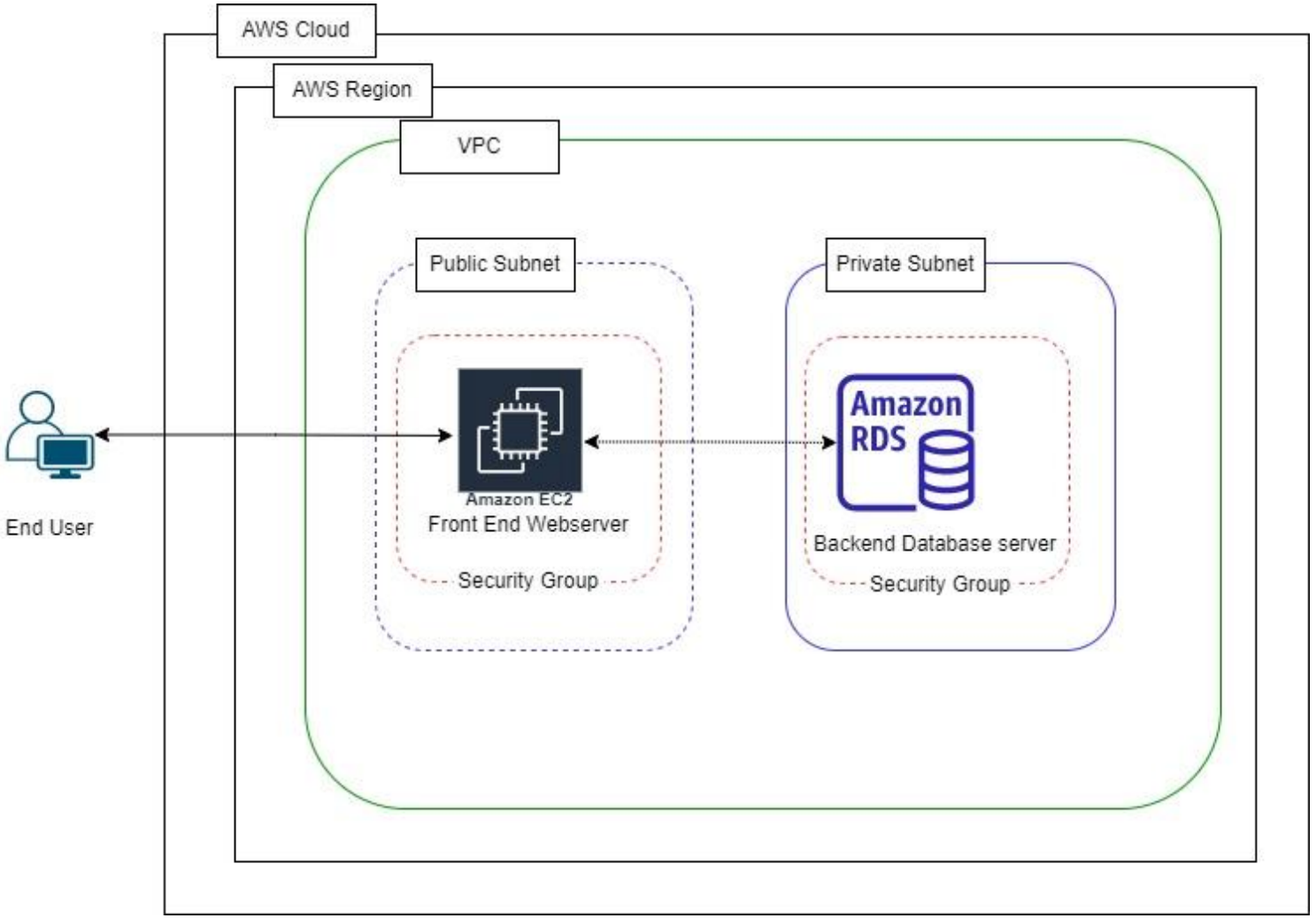
3. Creating Security Group in the new VPC

4. Creating a EC2 instance in the new VPC with user-data

5. Setting up RDS MySQL DB instance in the new VPC

6. Setting up the DB connections and testing the configuration

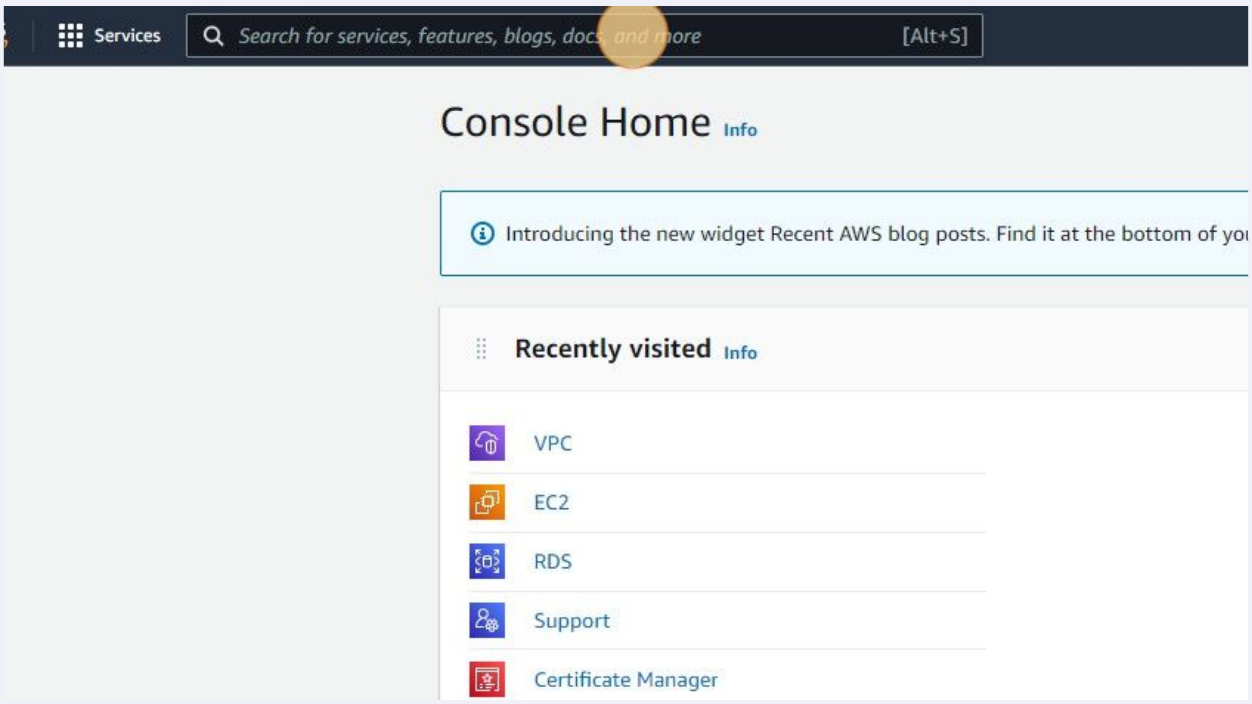
1. Architecture of 2-Tier Application in AWS



2. Creating a VPC

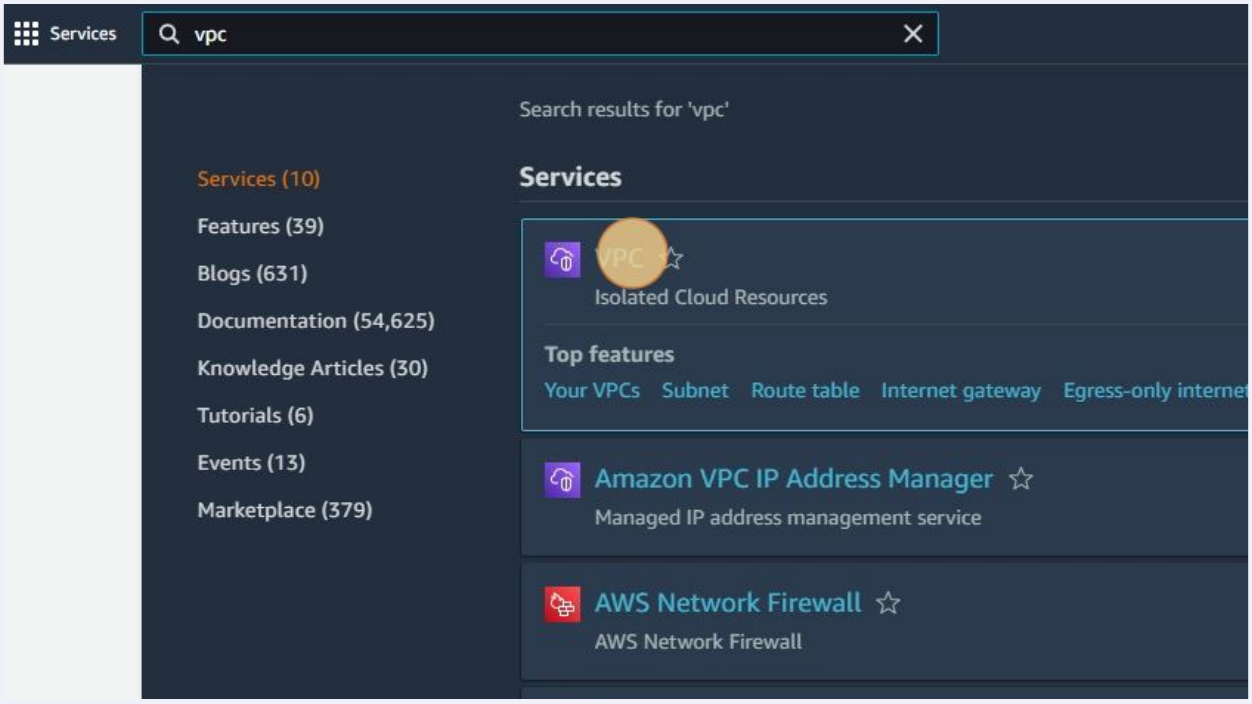
1 Navigate to https://ap-south-1.console.aws.amazon.com/console/home?region=ap-south-1

2 Click the "Search for services, features, blogs, docs, and more" field.

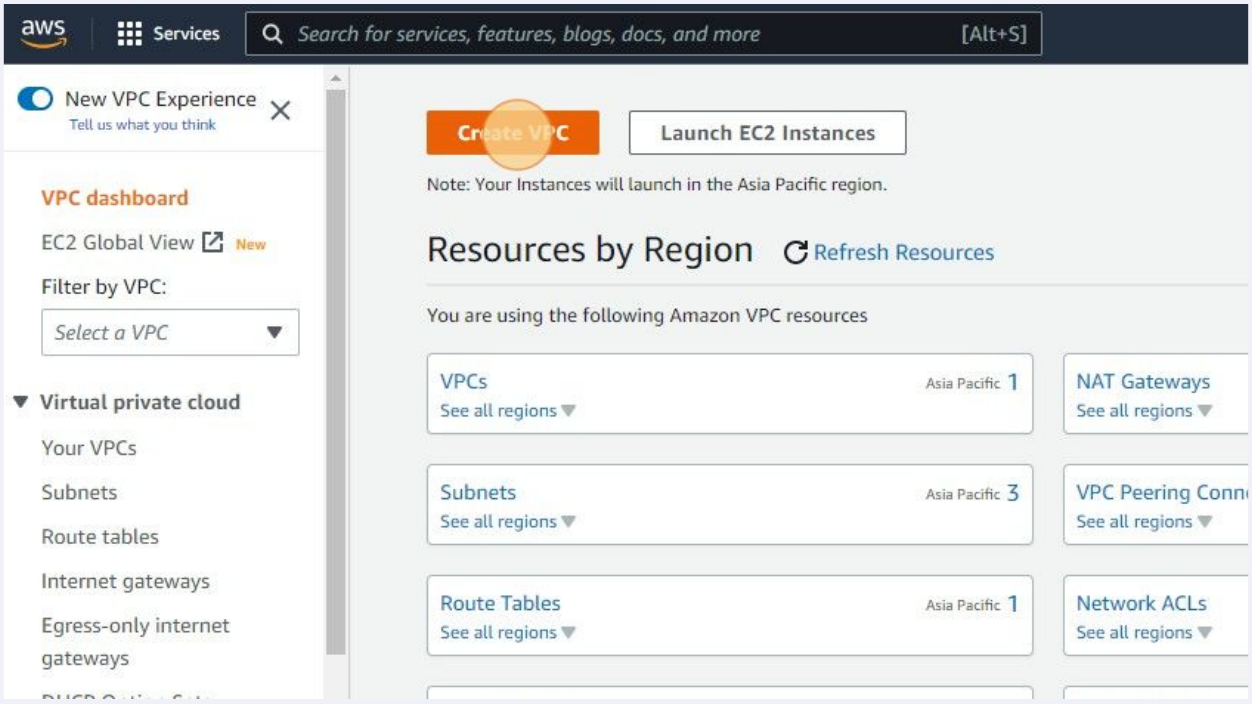


3 Type "vpc"

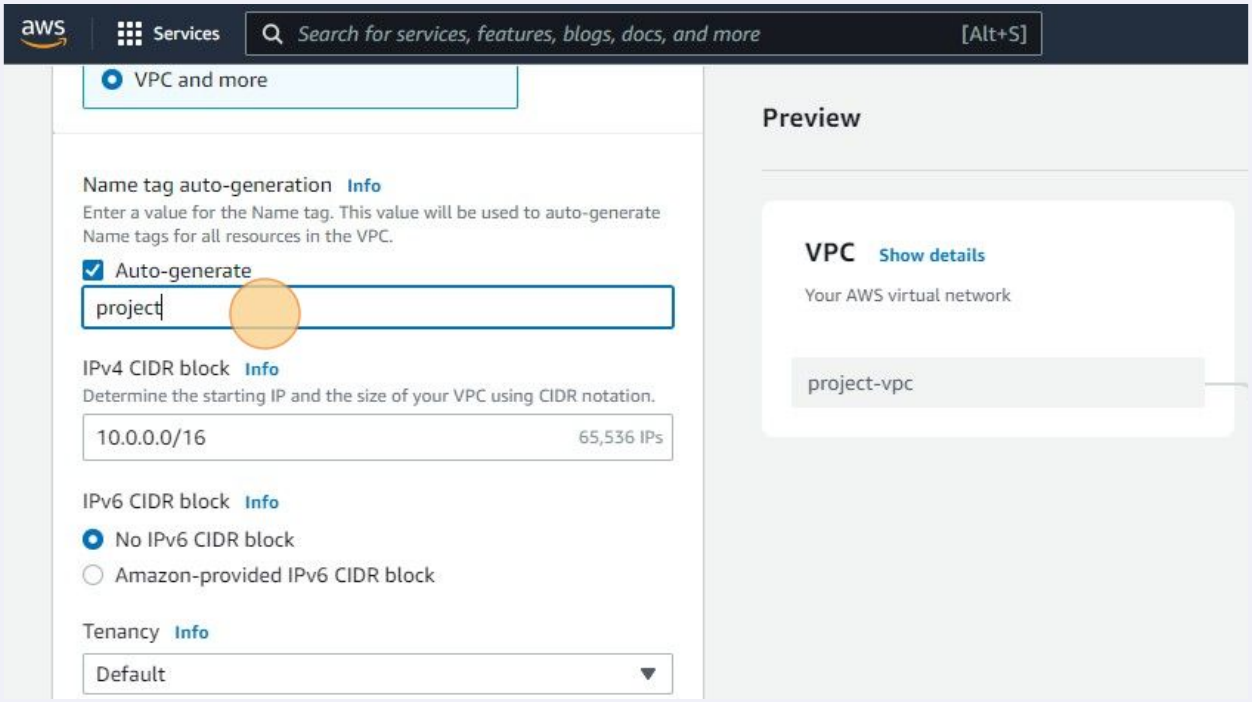
4 Click "VPC"



5 Click "Create VPC"

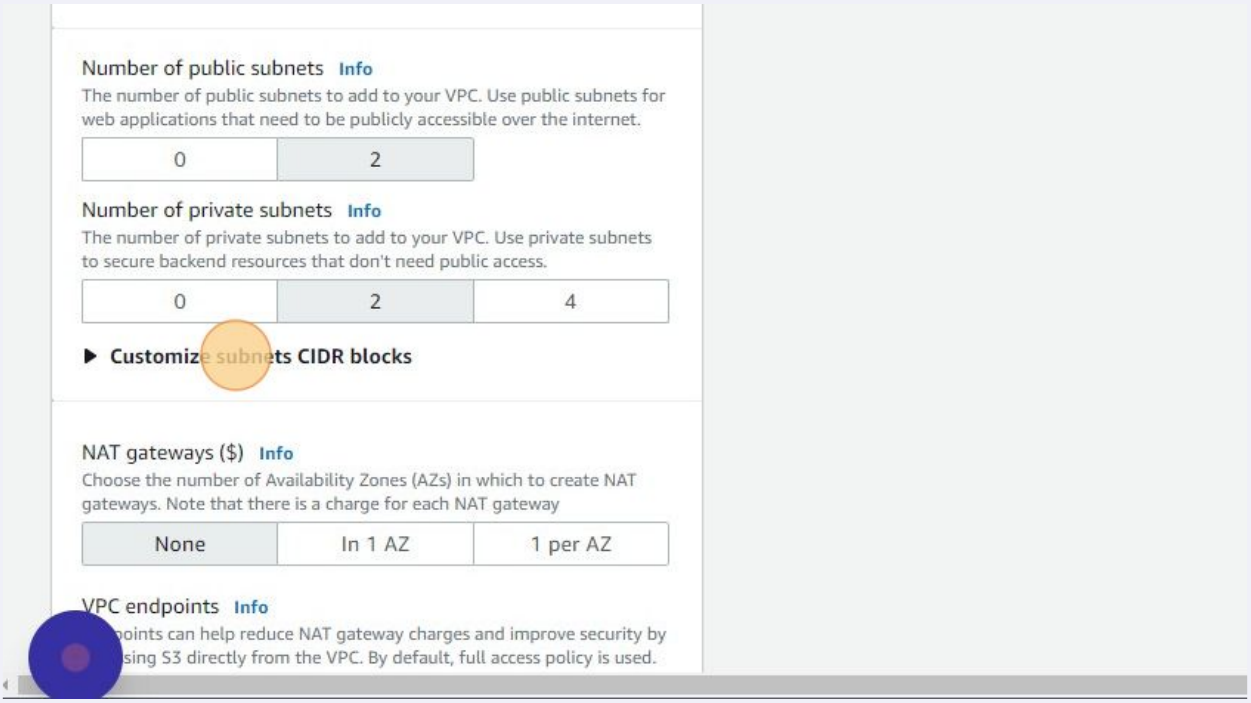


6 Double-click the "Name tag auto-generationInfo" field.

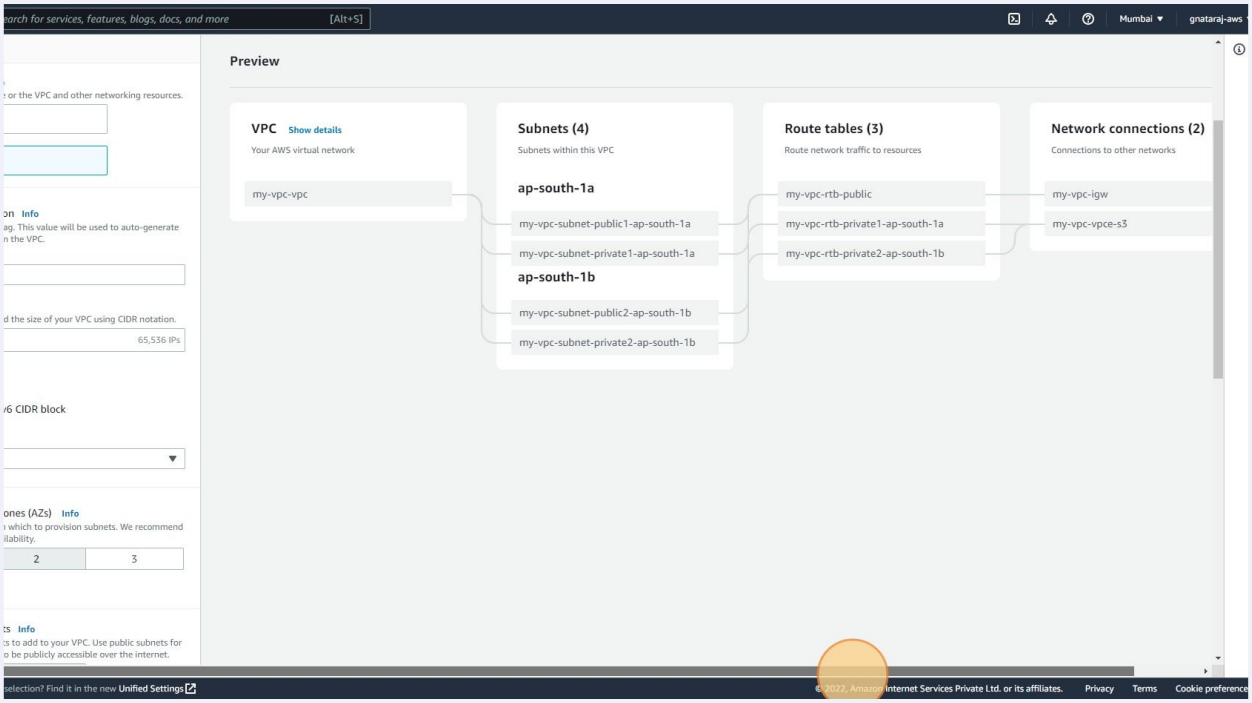


7 Type "my-vpc"

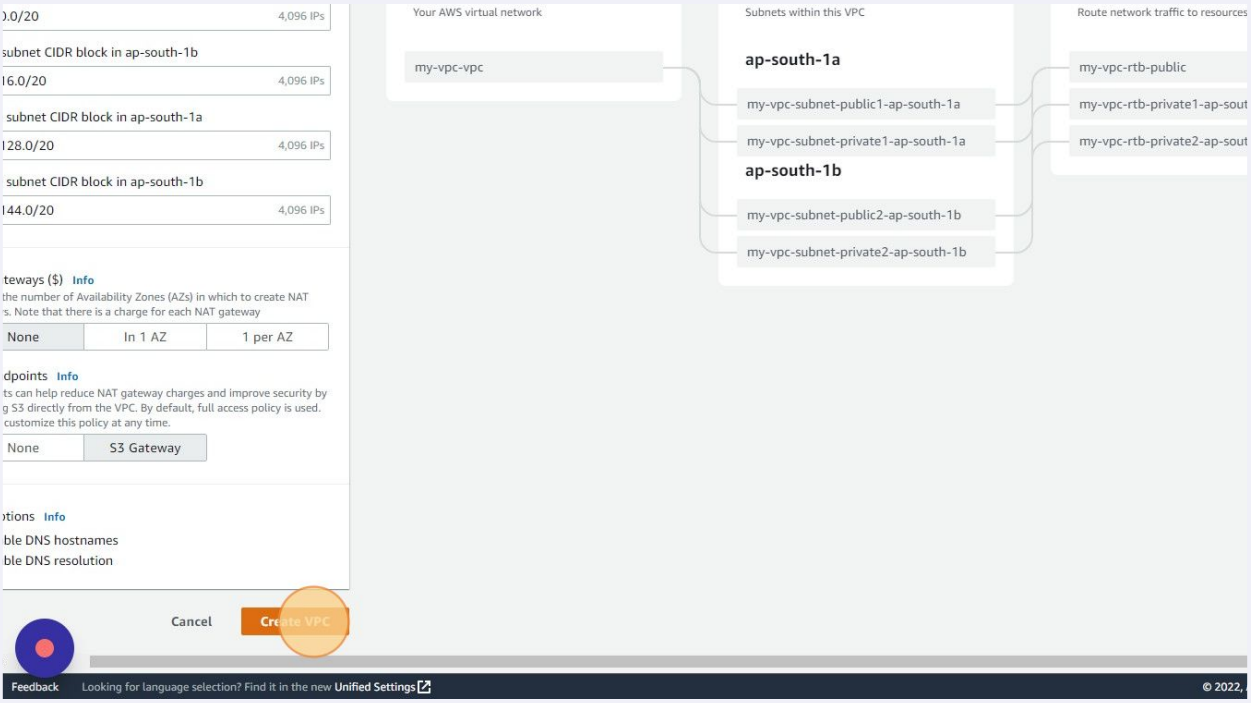
8 Click here.



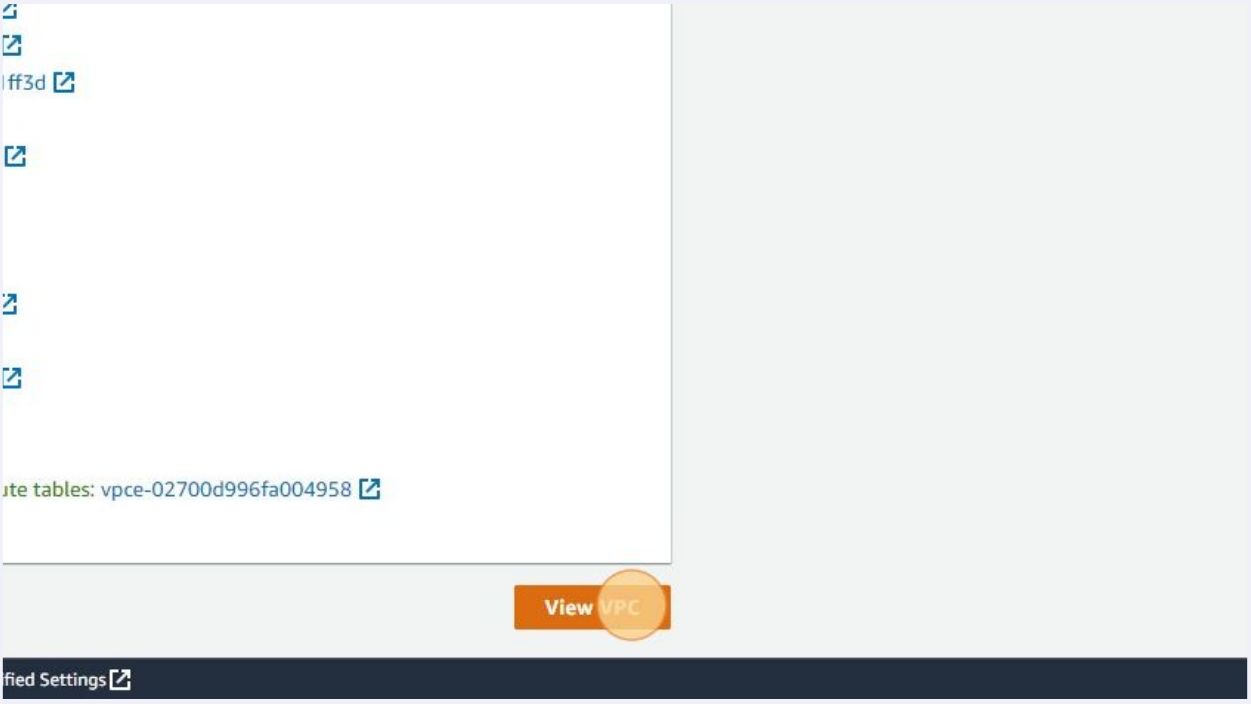
9 Verify the VPC layout



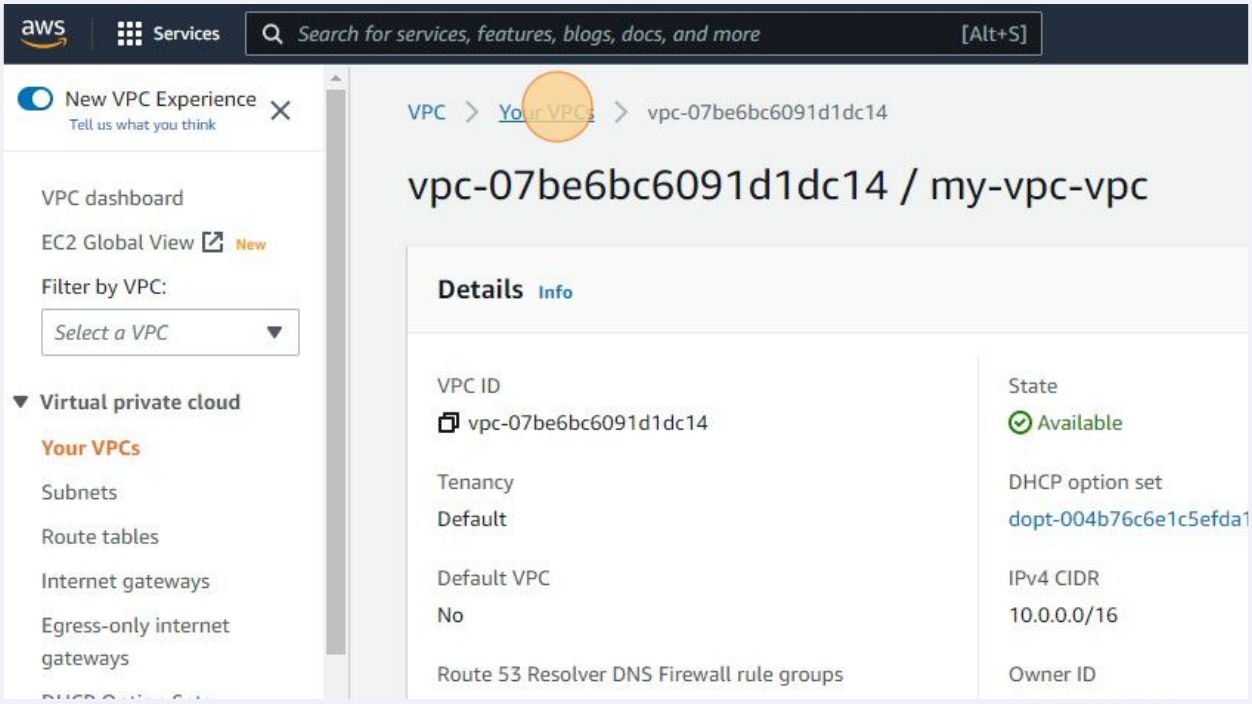
10 Click "Create VPC"



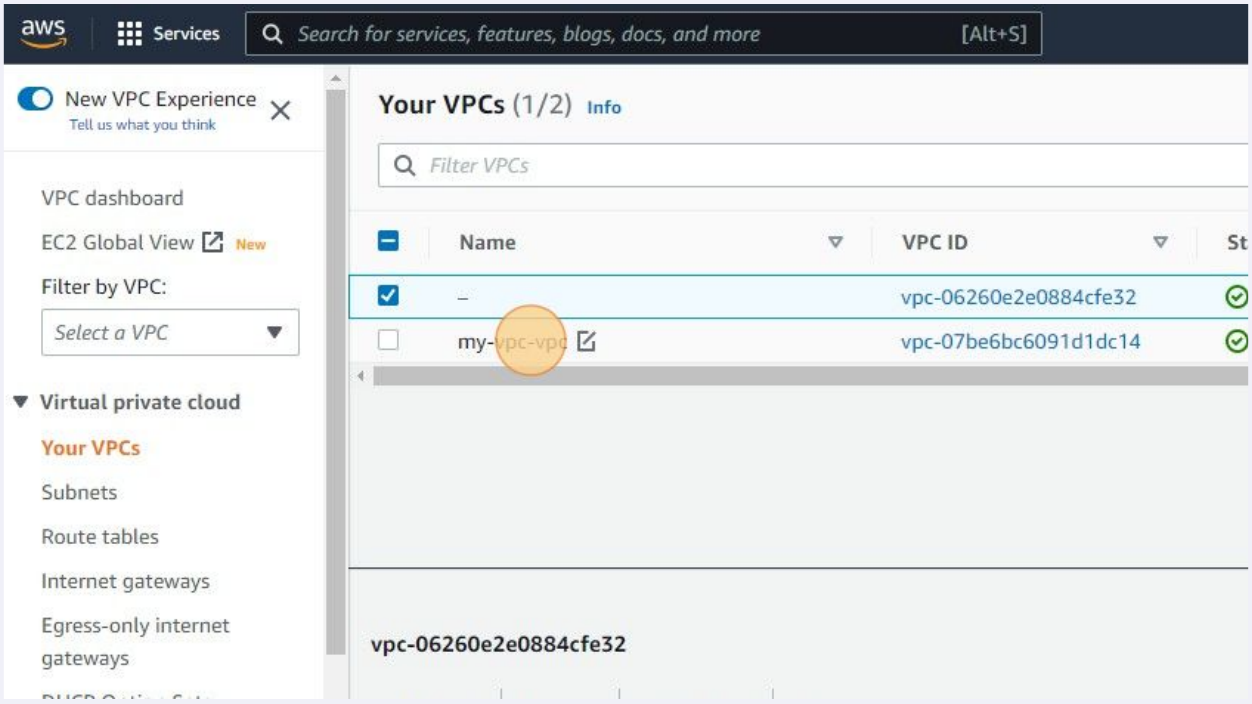
11 Click "View VPC"



12 Click "Your VPCs"



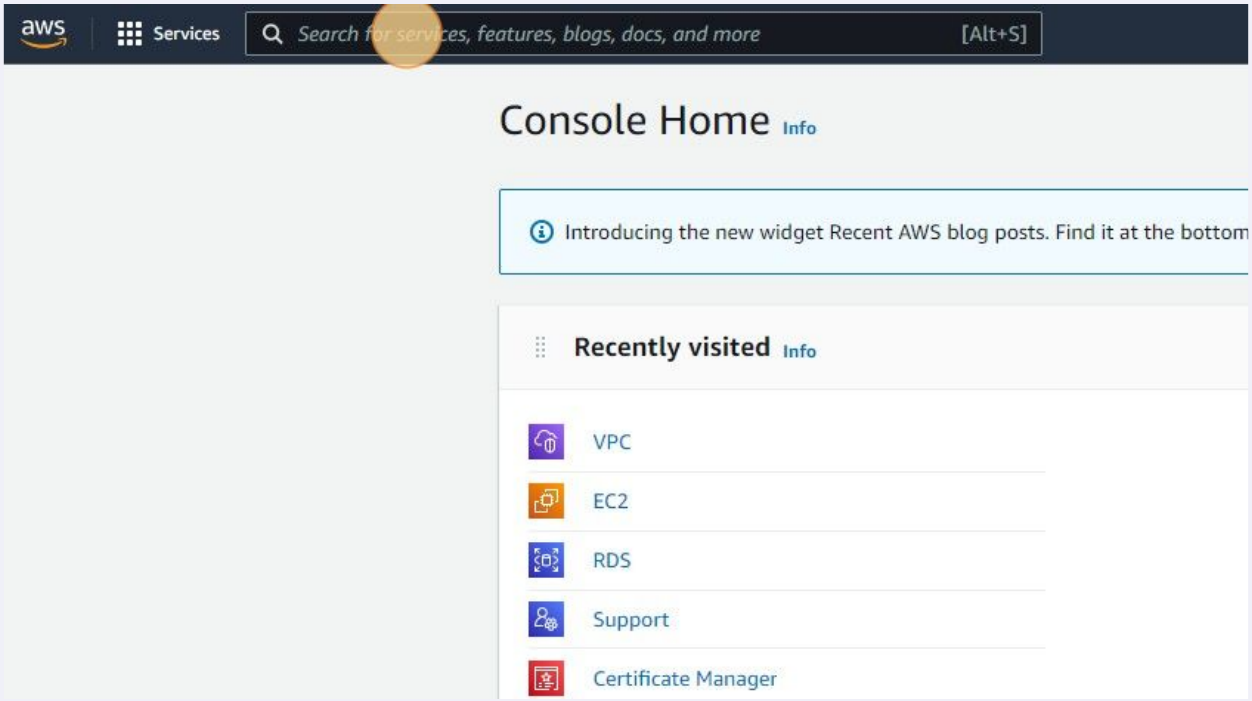
13 Click "my-vpc-vpc"



3. Creating Security Group in the new VPC

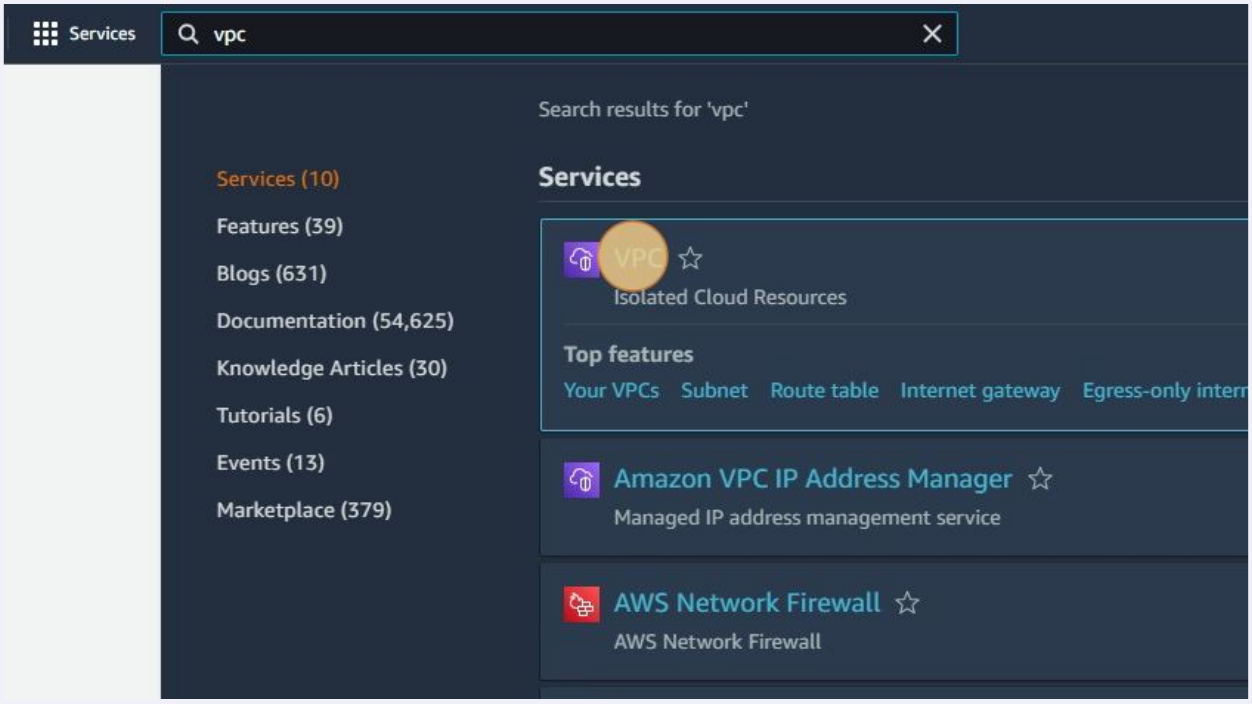
1 Navigate to https://ap-south-1.console.aws.amazon.com/console/home?region=ap-south-1

2 Click the "Search for services, features, blogs, docs, and more" field.

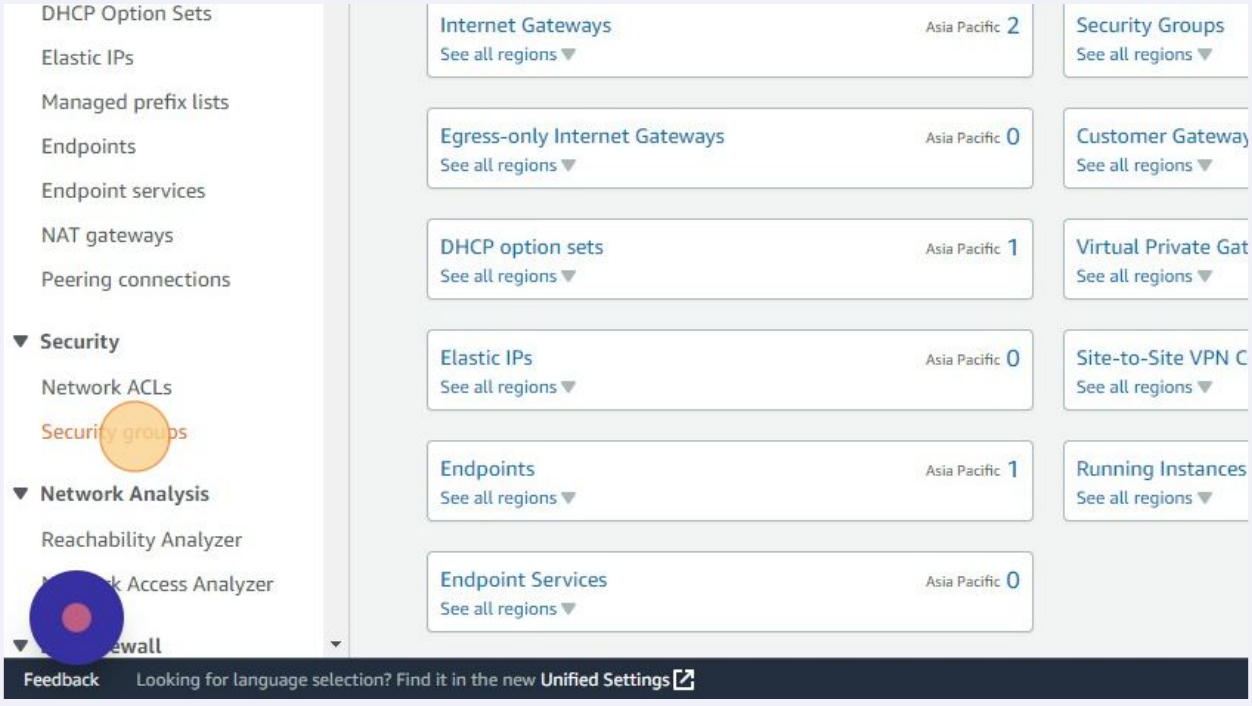


3 Type "vpc"

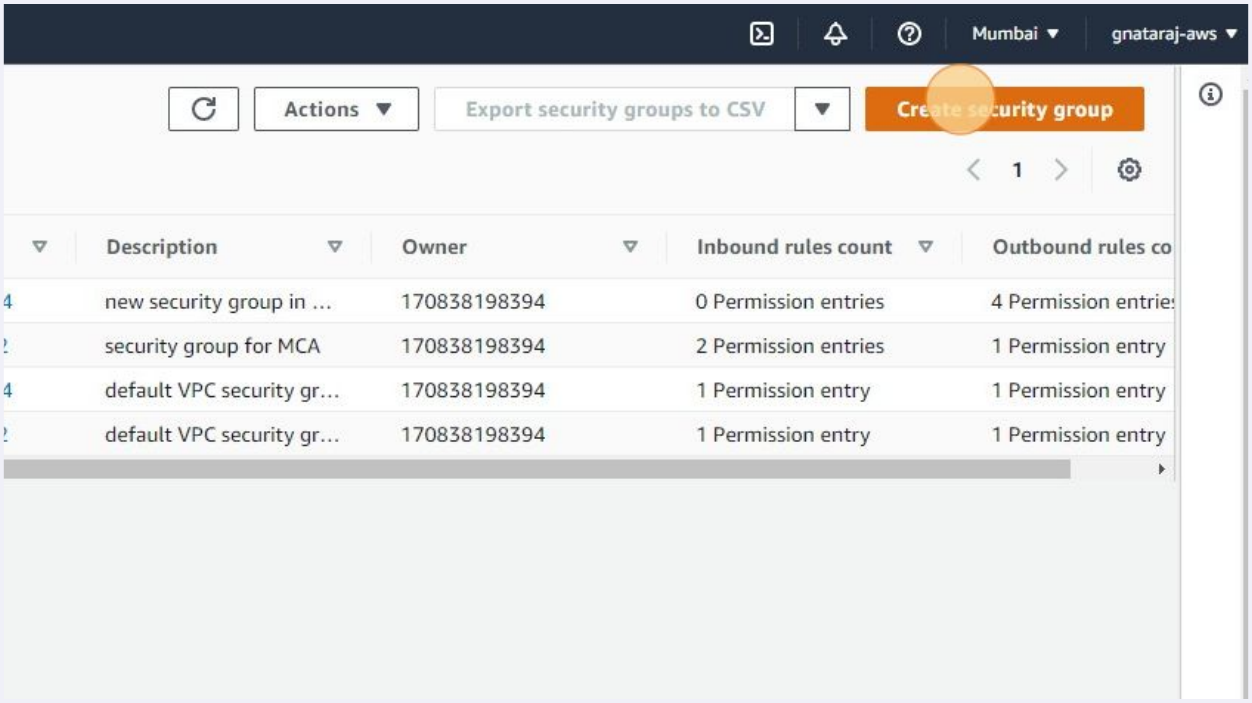
4 Click "VPC"



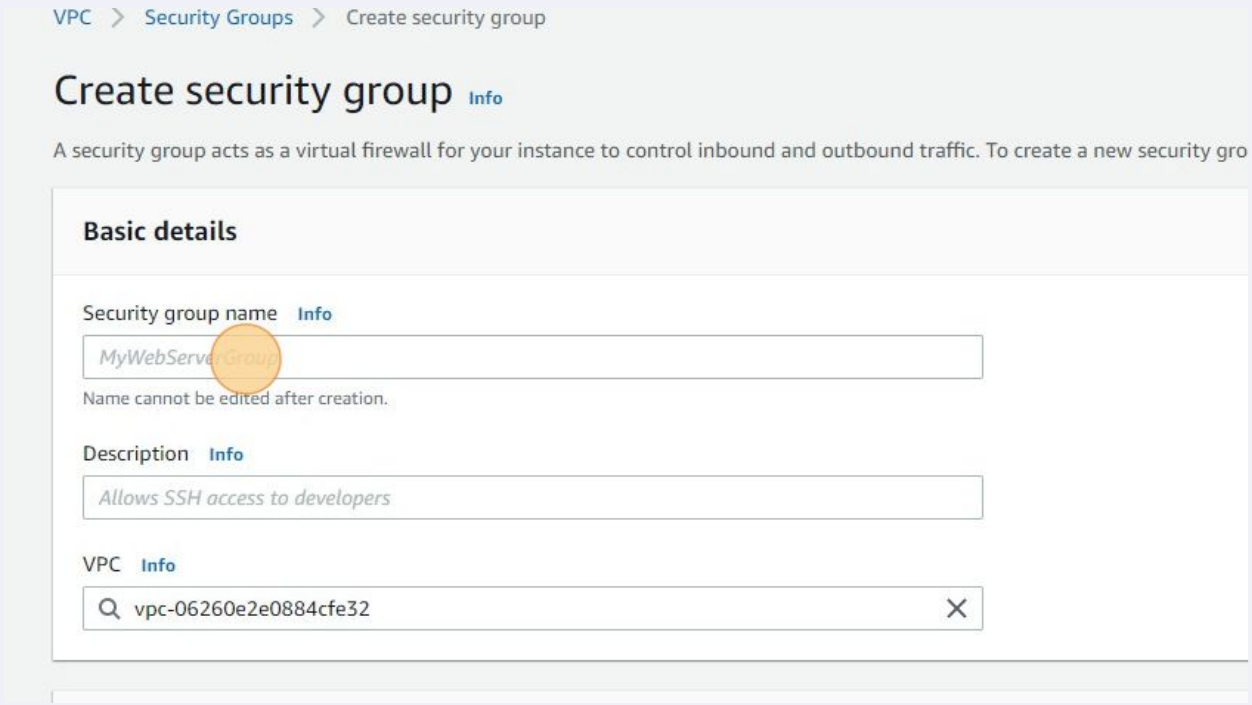
5 Click "Security groups"



6 Click "Create security group"

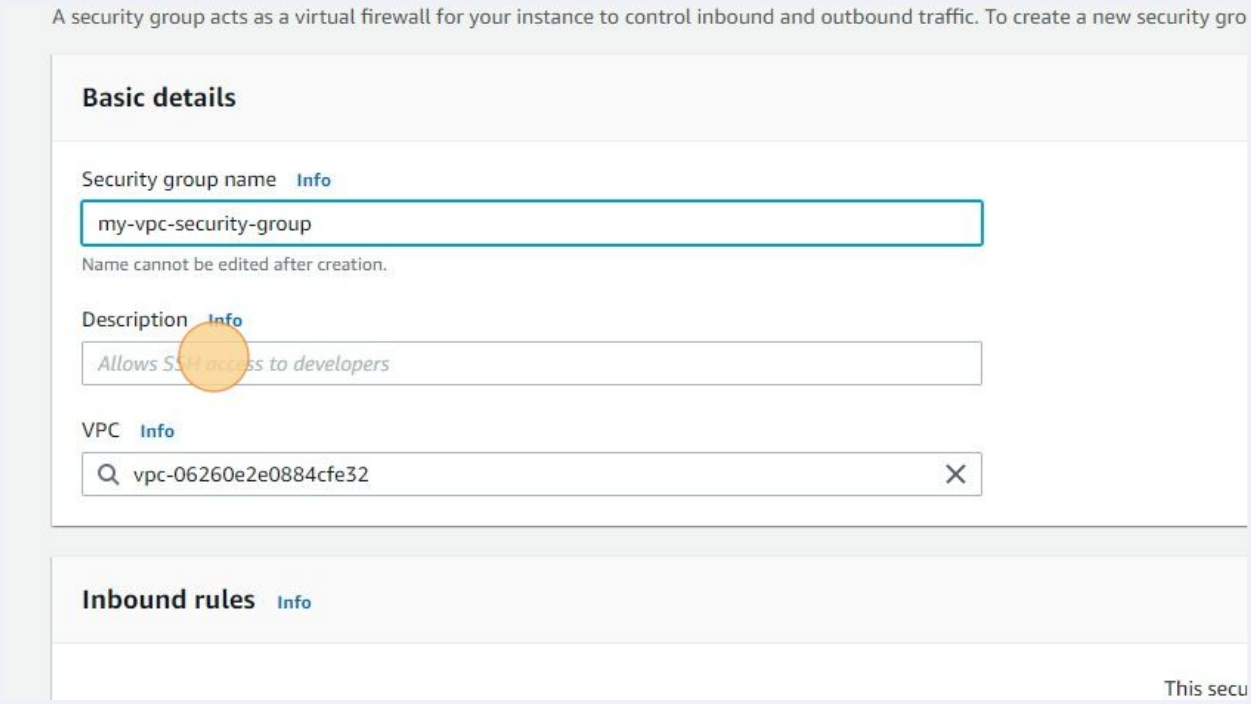


7 Click the "Security group name Info" field.



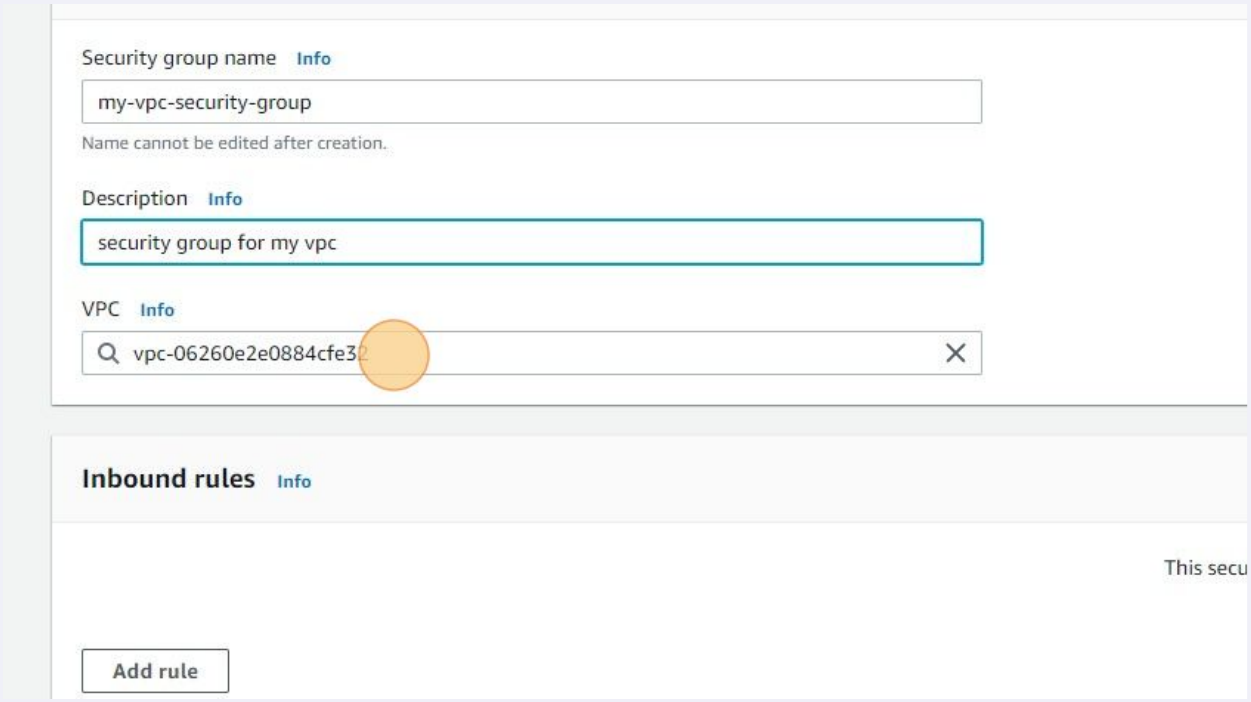
8 Type "my-vpc-security-group"

9 Click the "Description Info" field.

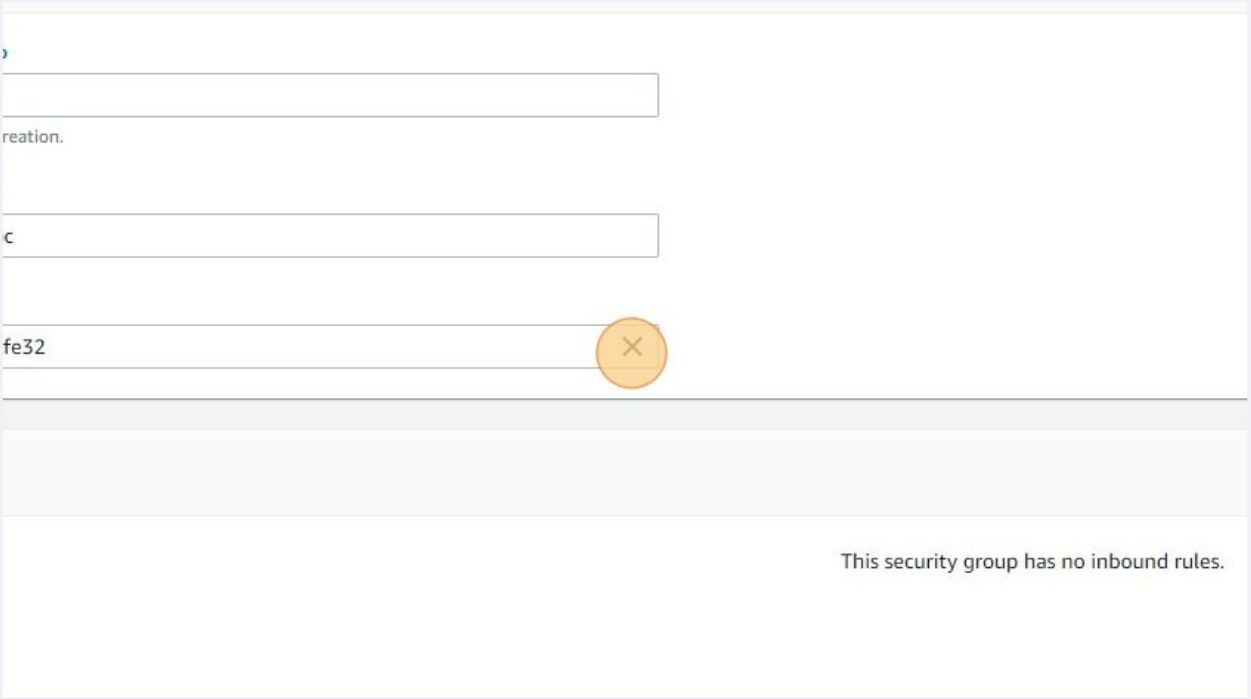


10 Type "security group for my vpc"

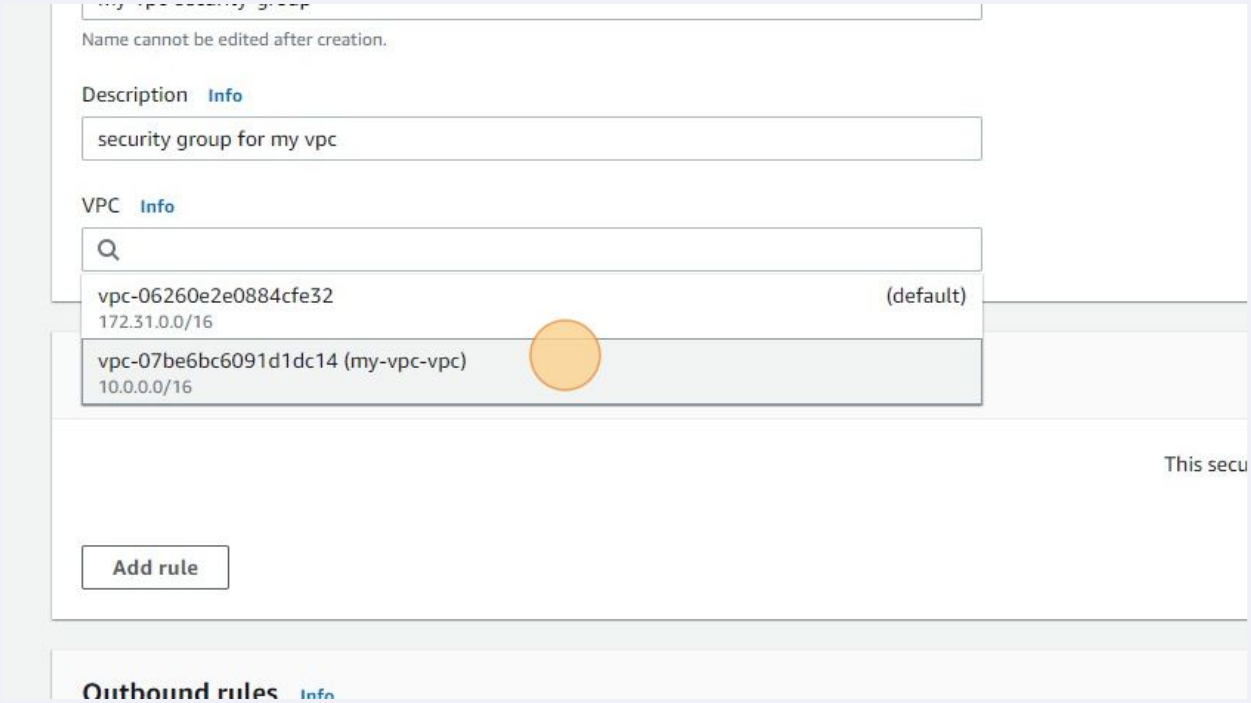
11 Click the "VPC Info" field.



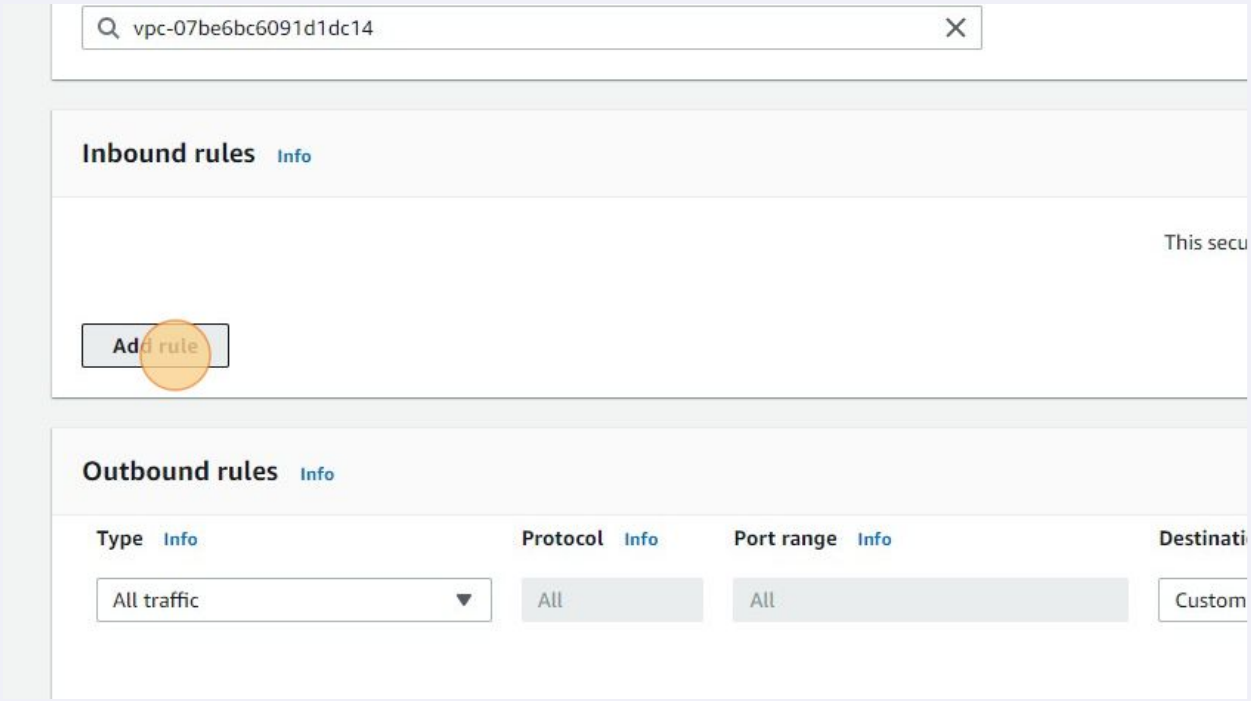
12 Click here.



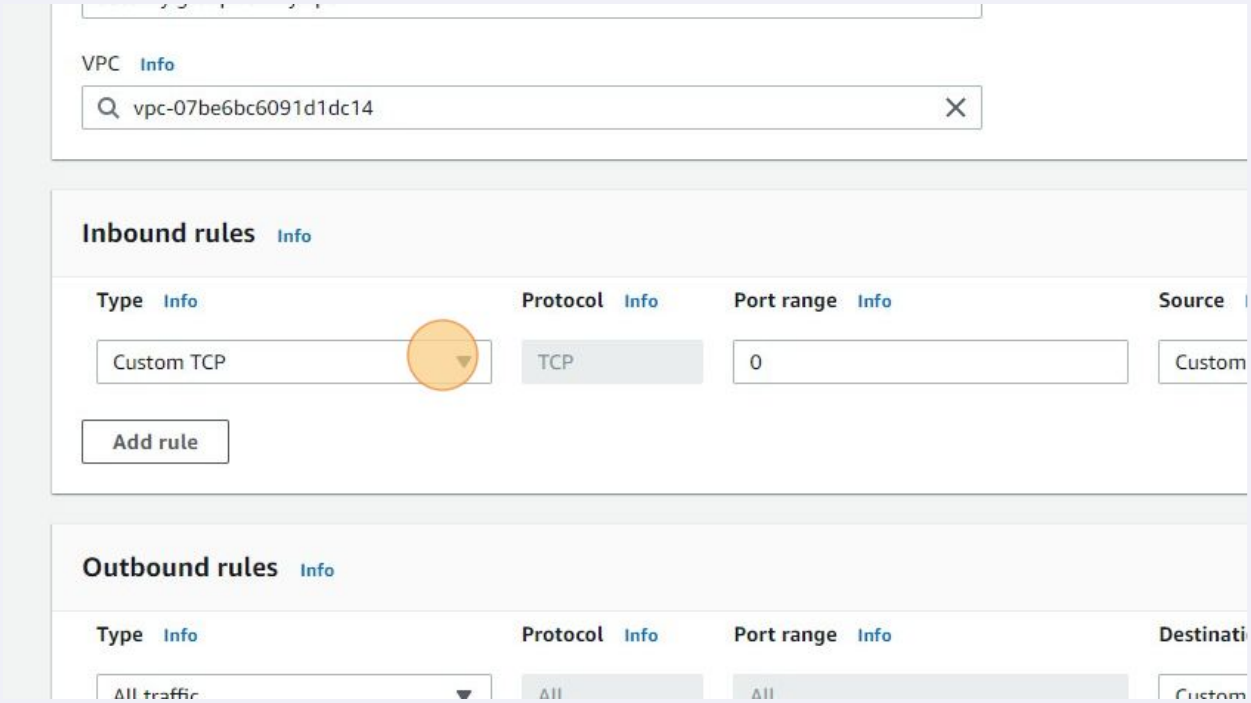
13 Click here.



14 Click "Add rule"

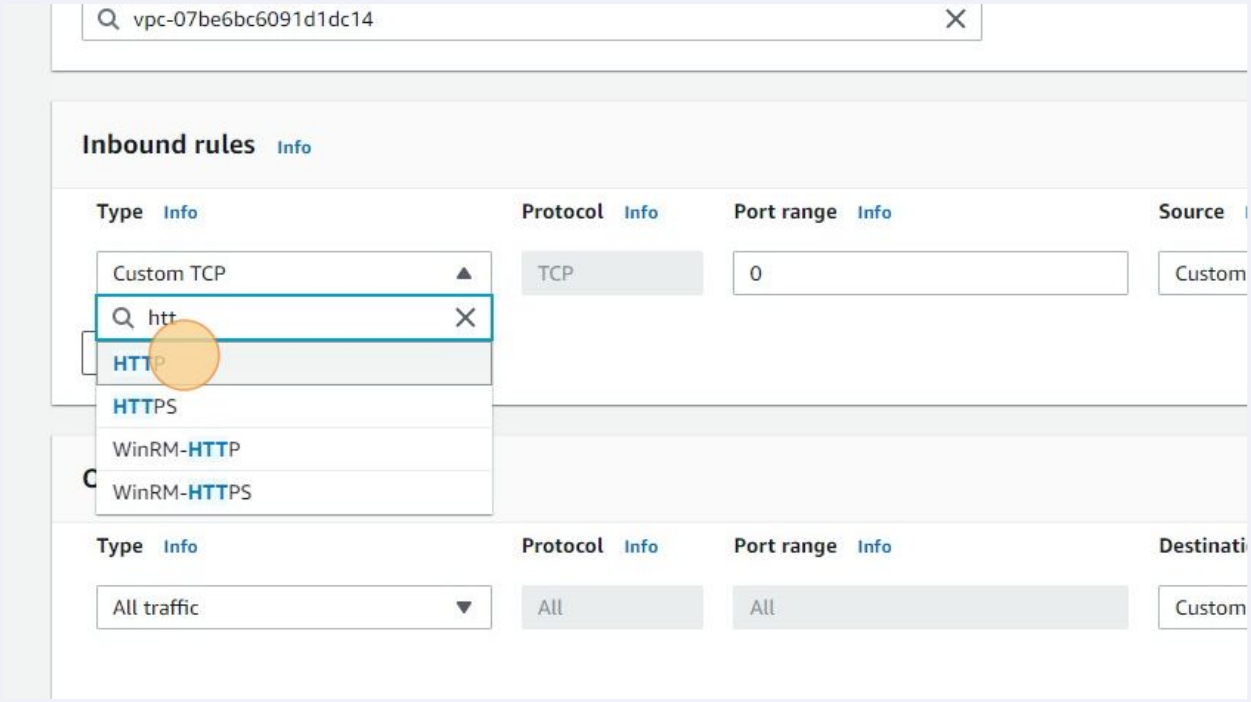


15 Click "Custom TCP"

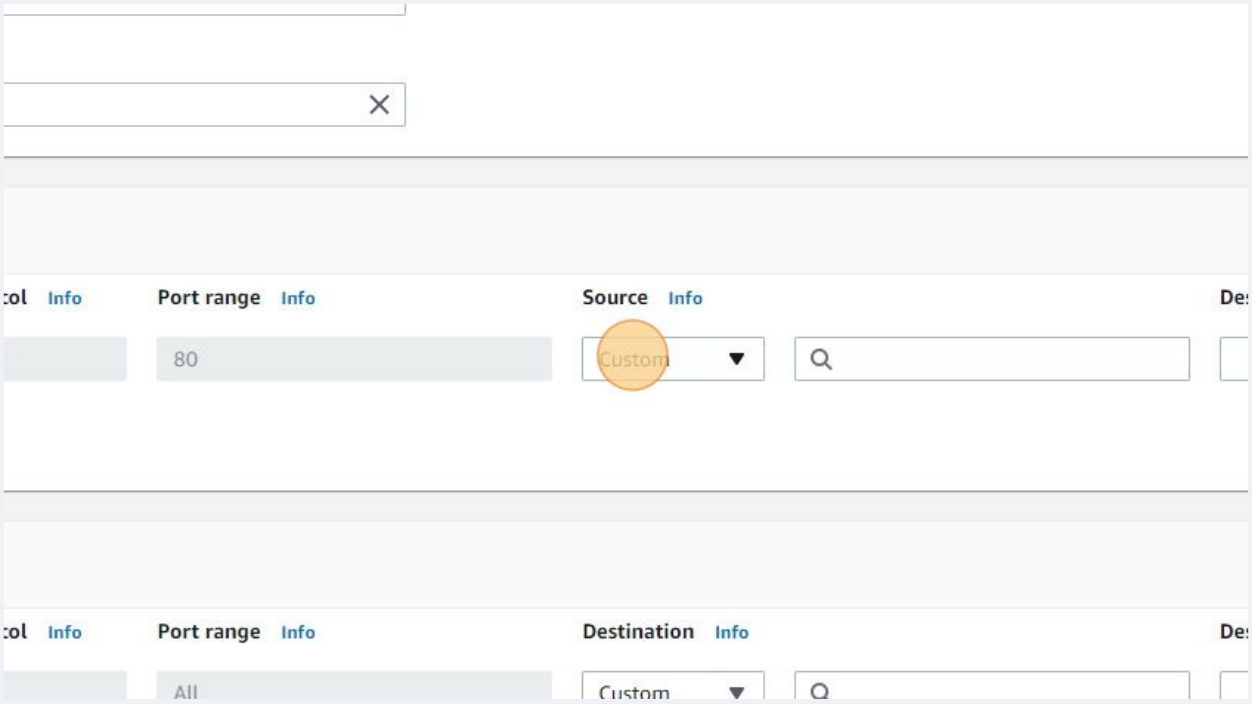


16 Type "htt"

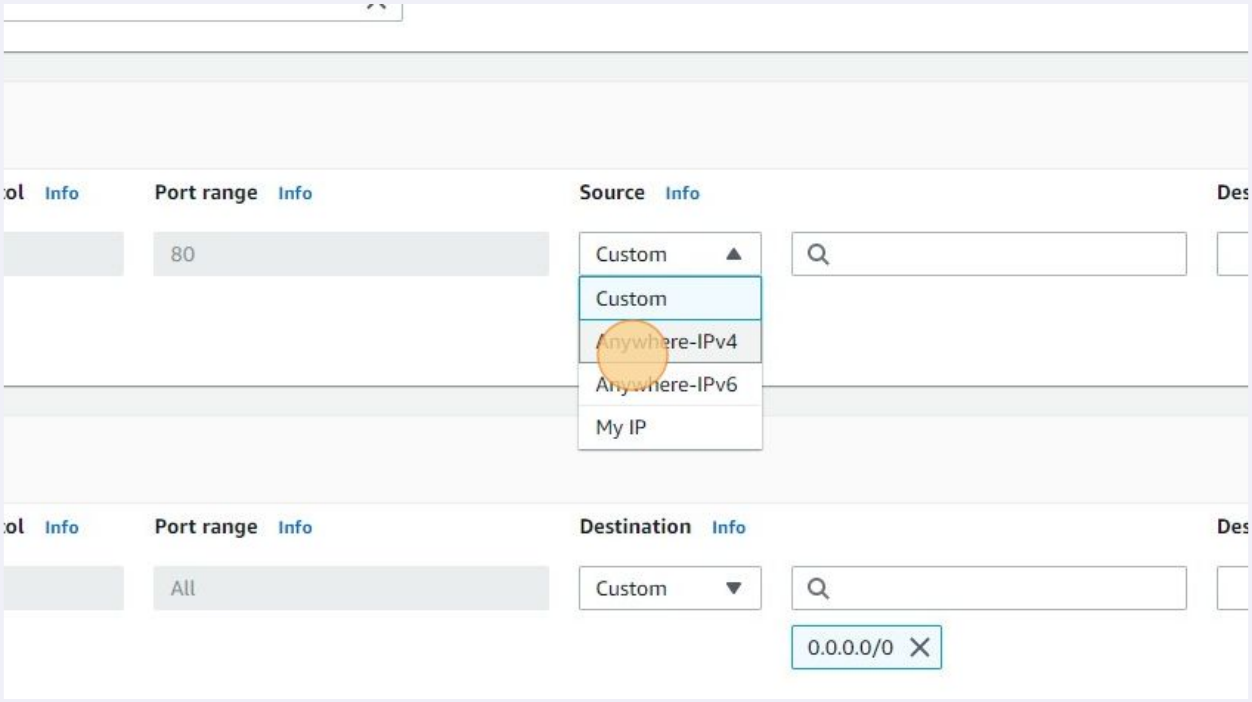
17 Click here.



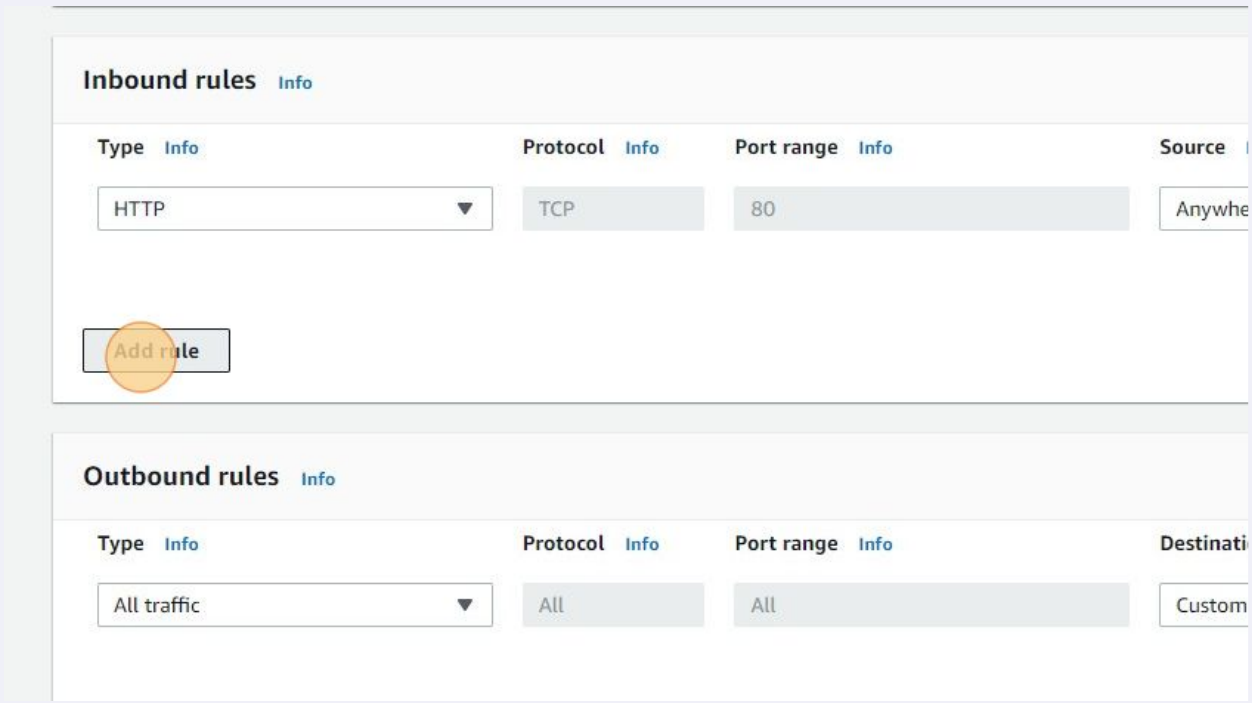
18 Click "Custom"



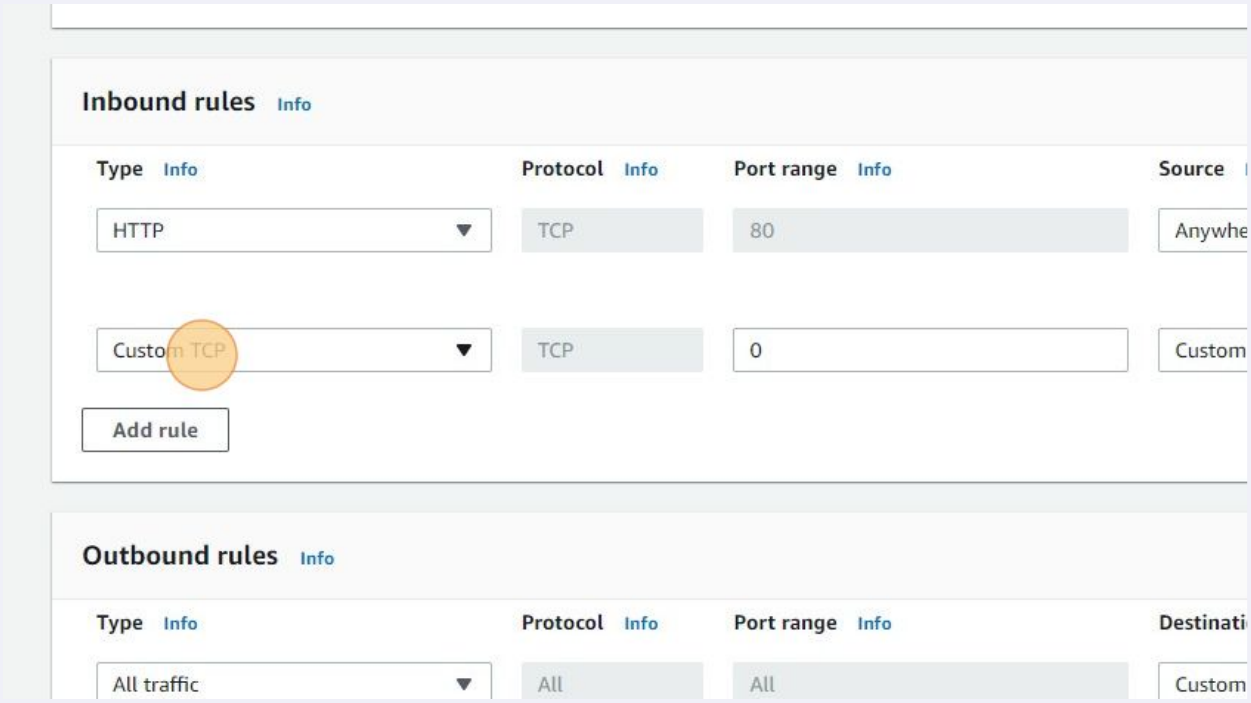
19 Click "Anywhere-IPv4"



20 Click "Add rule"

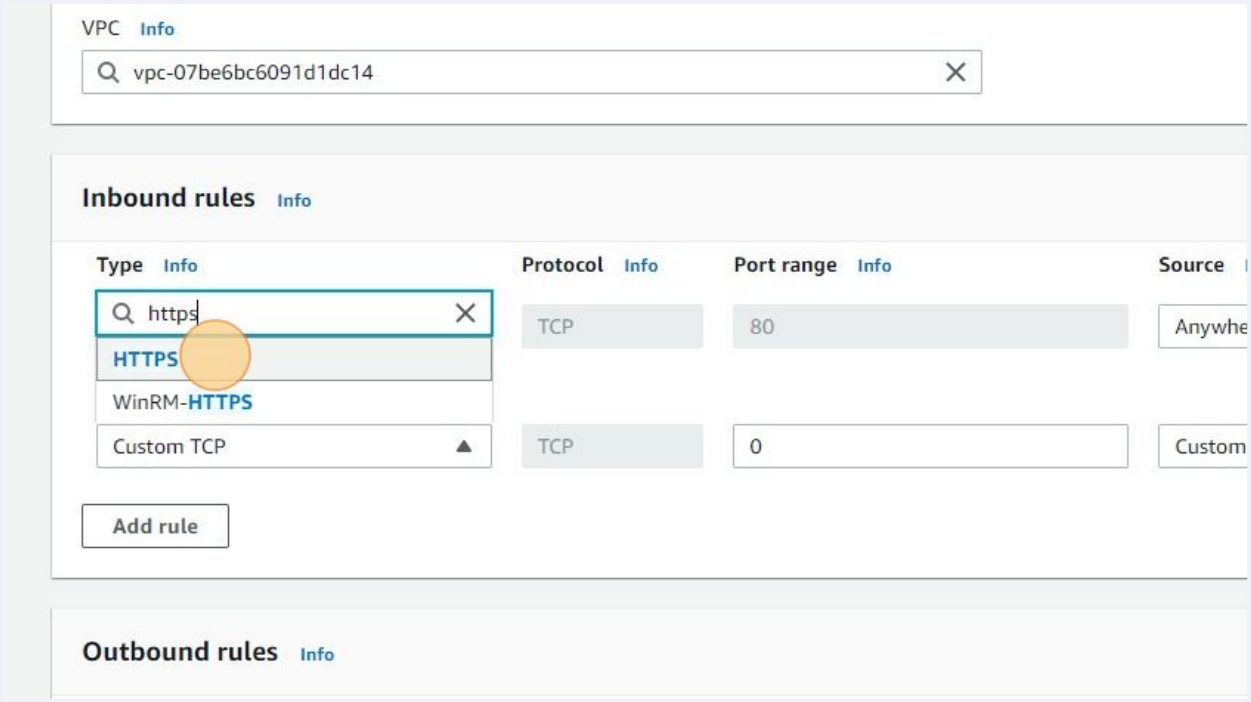


21 Click "Custom TCP"

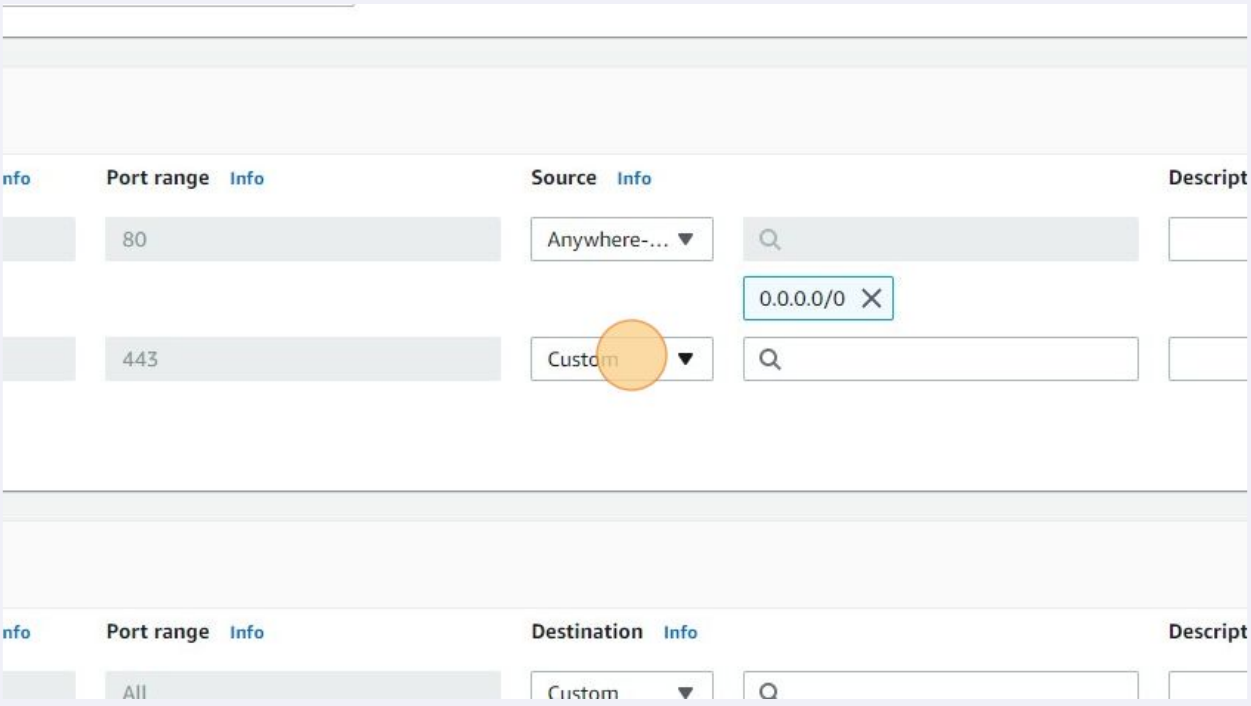


22 Type "https"

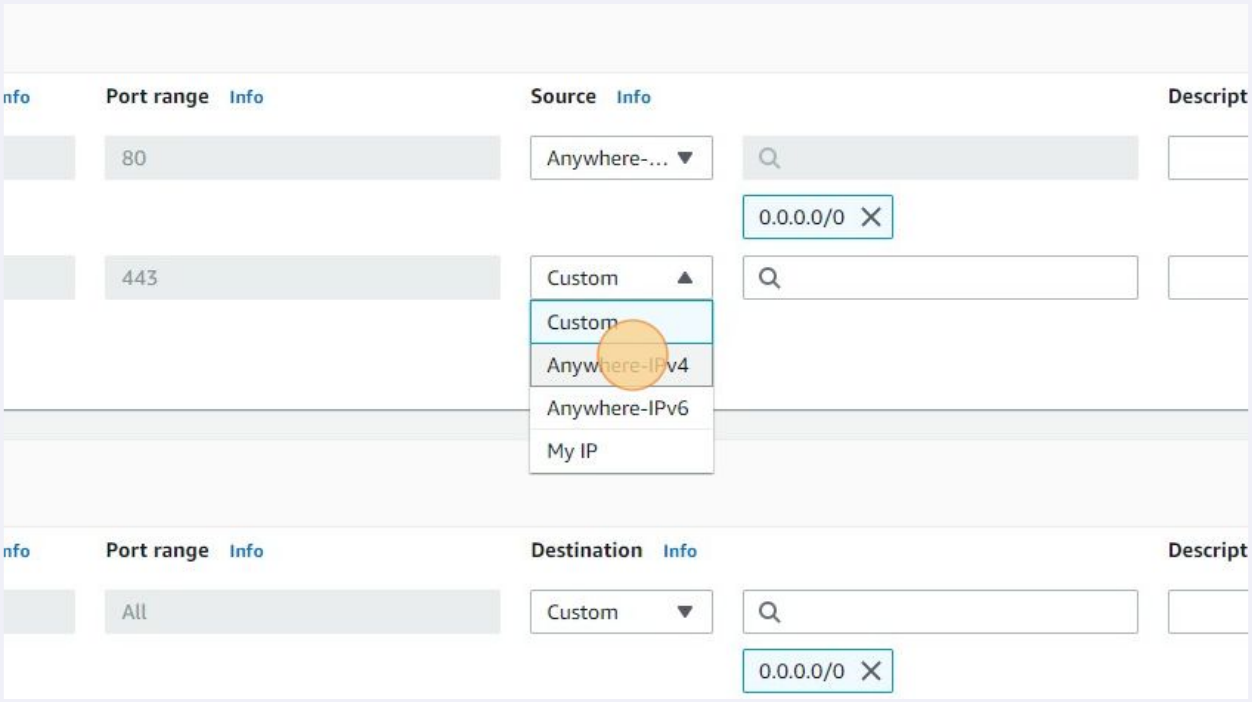
23 Click here.



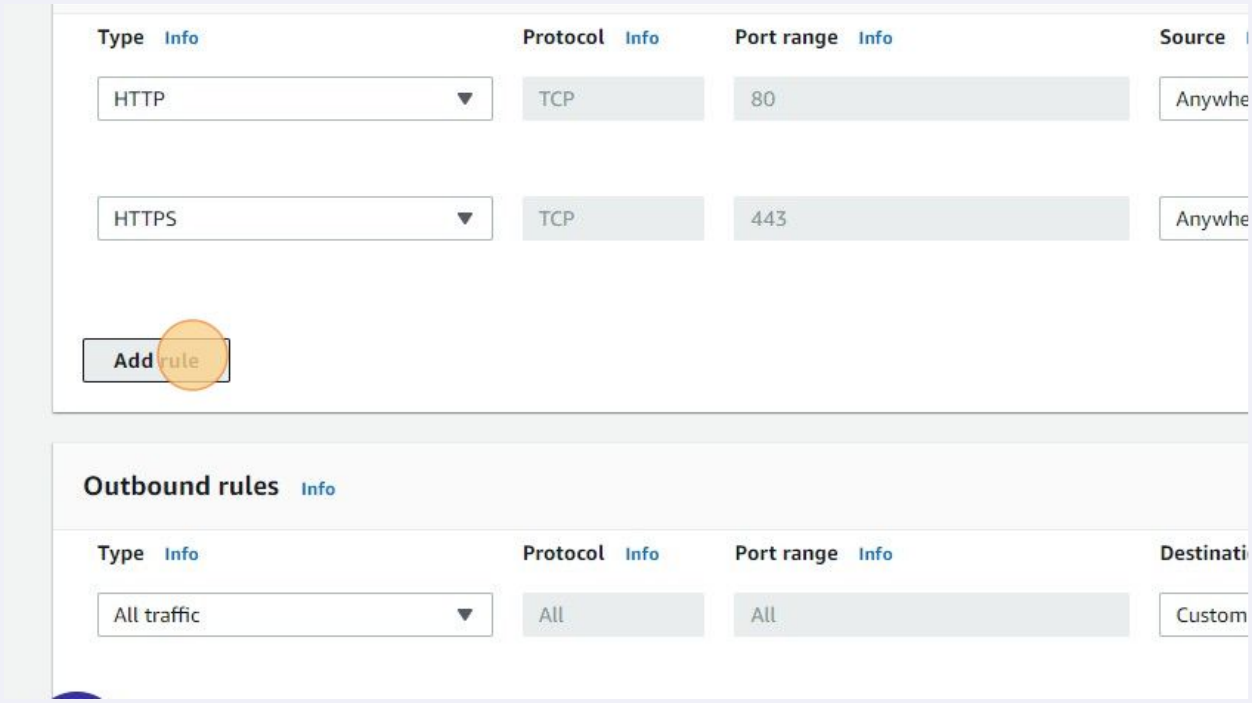
24 Click "Custom"



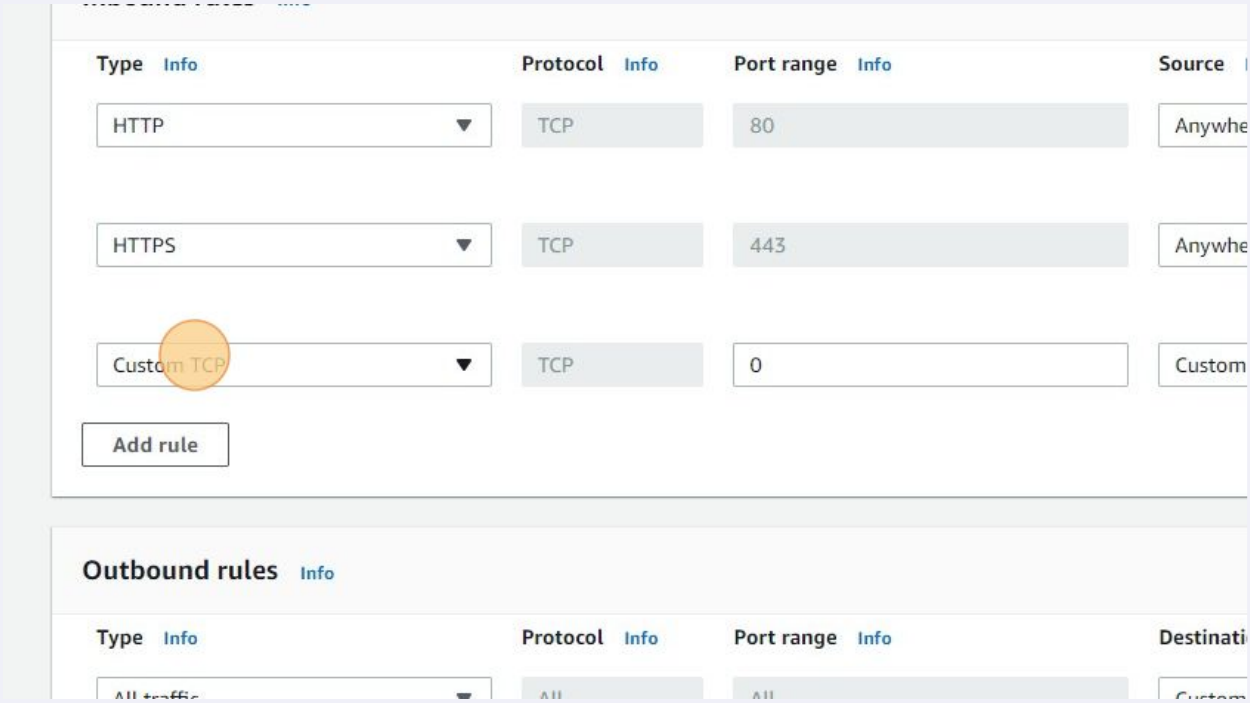
25 Click "Anywhere-IPv4"



26 Click "Add rule"

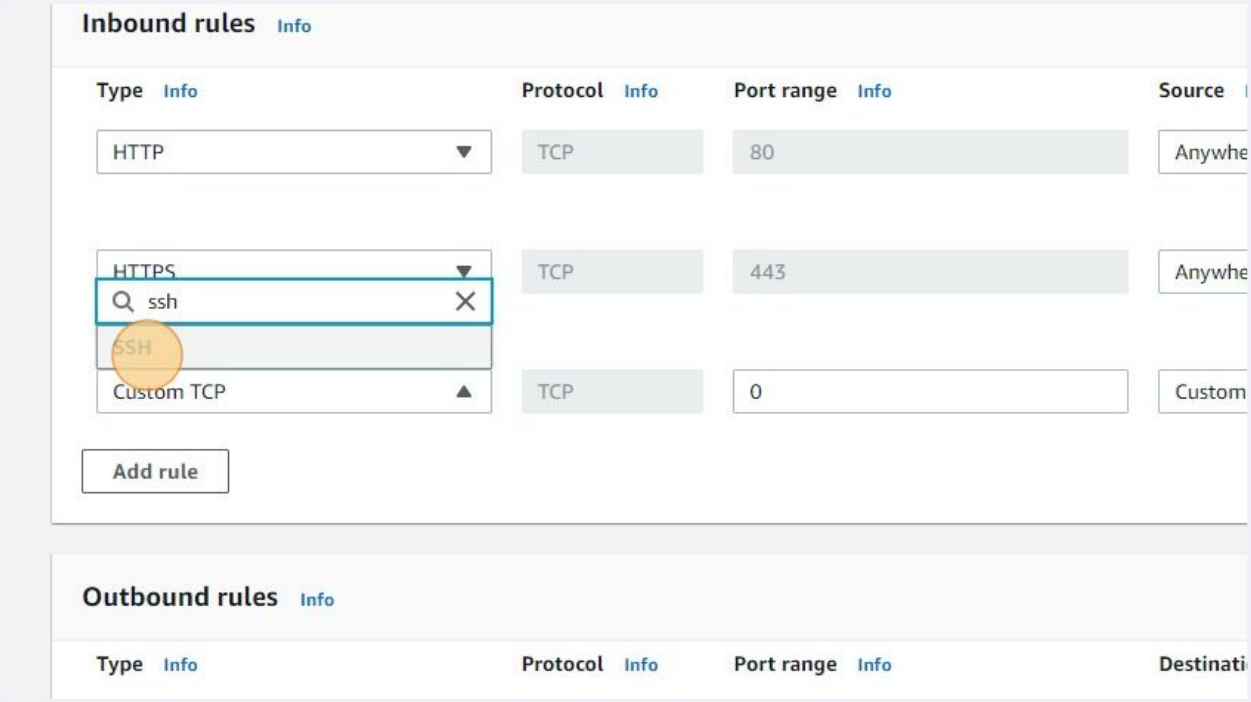


27 Click "Custom TCP"

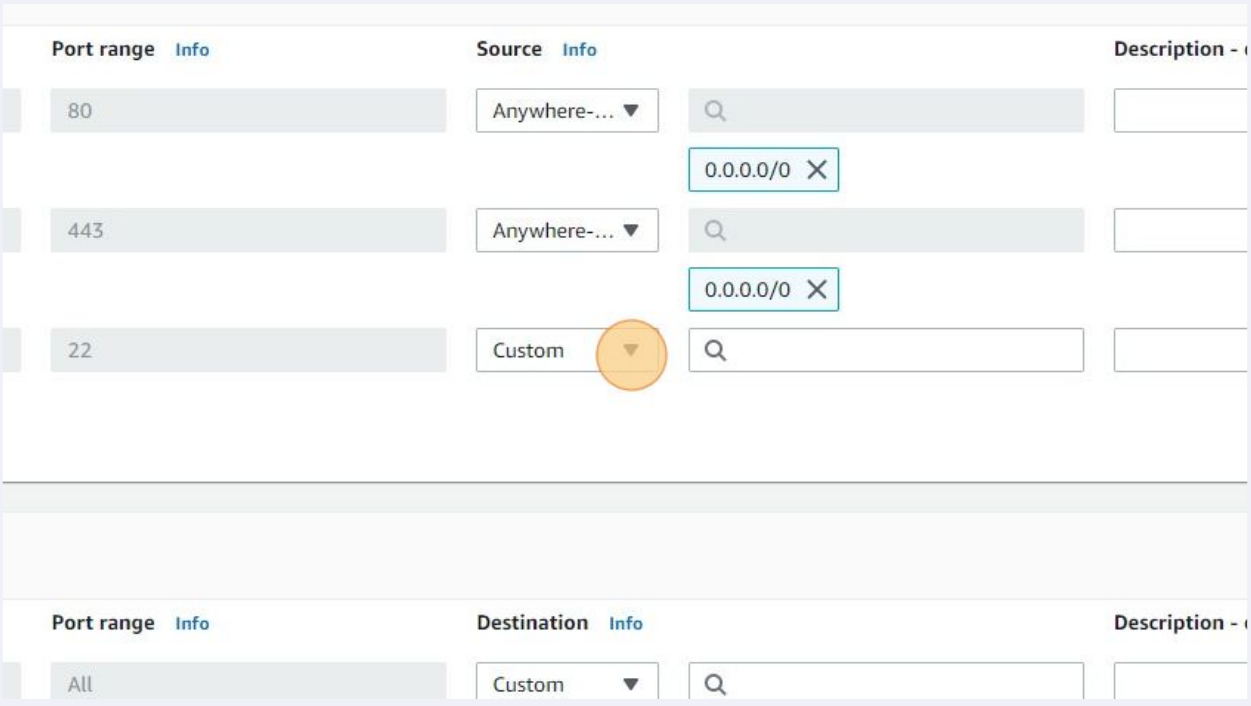


28 Type "ssh"

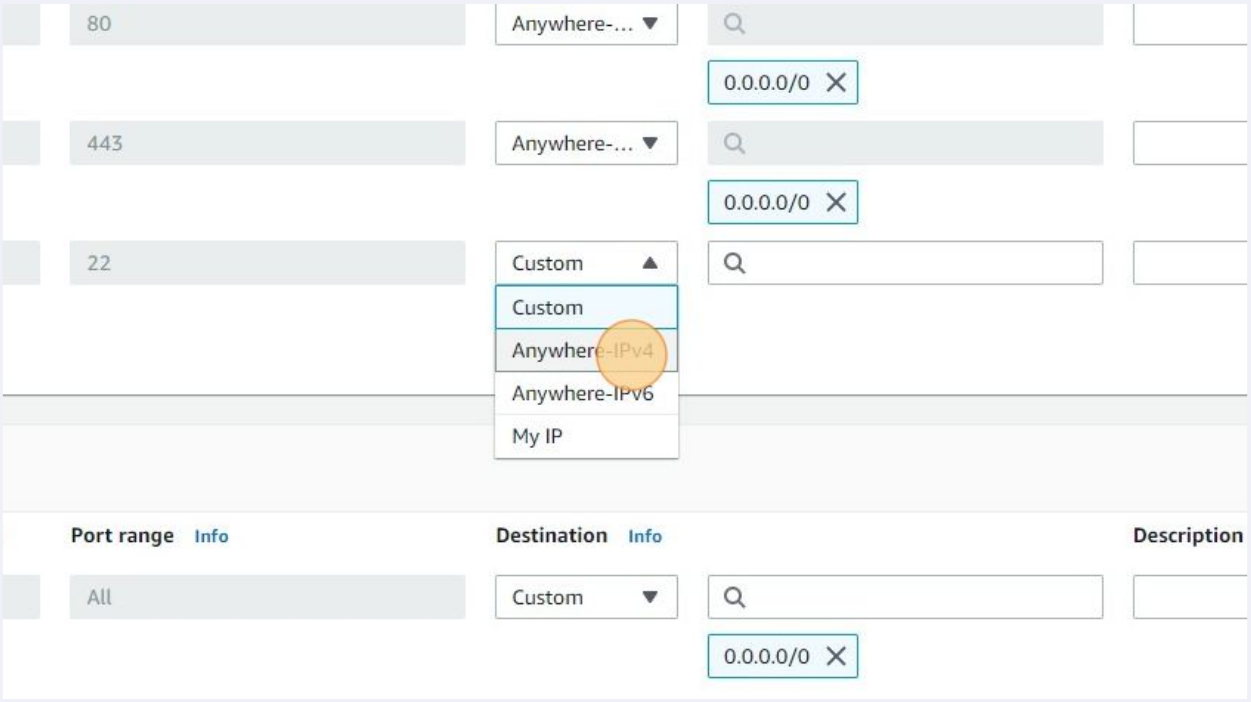
29 Click "SSH"



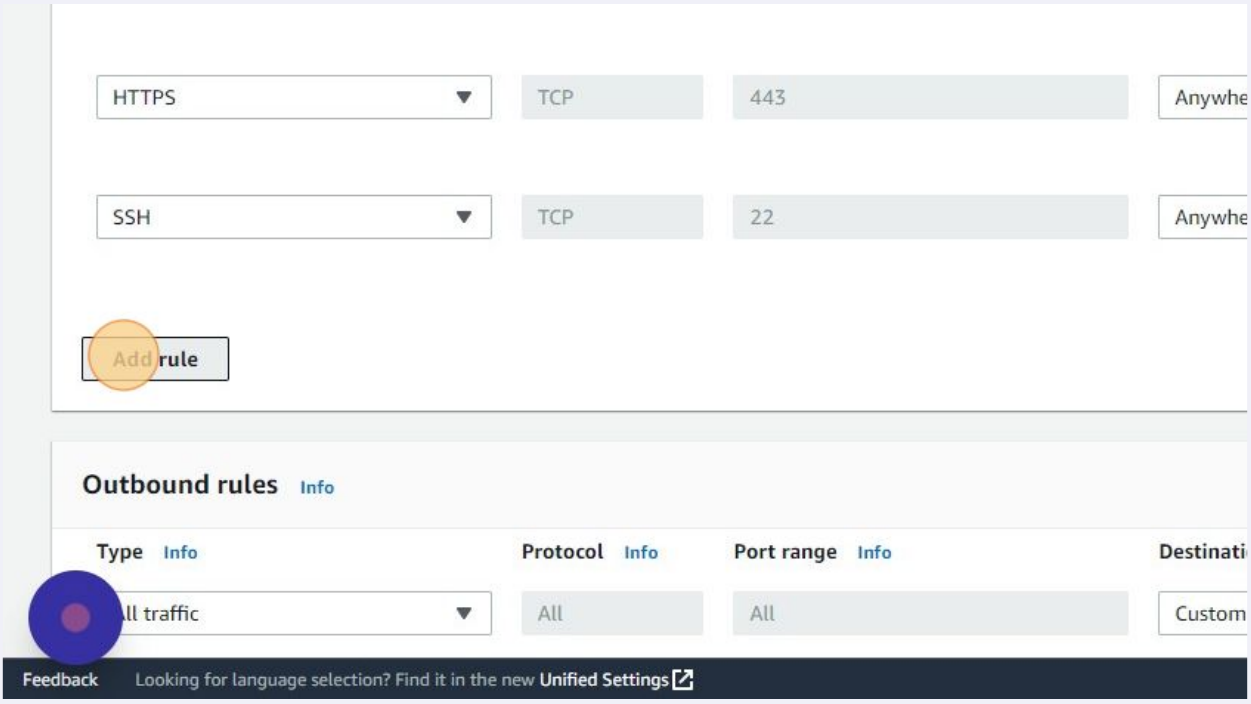
30 Click here.



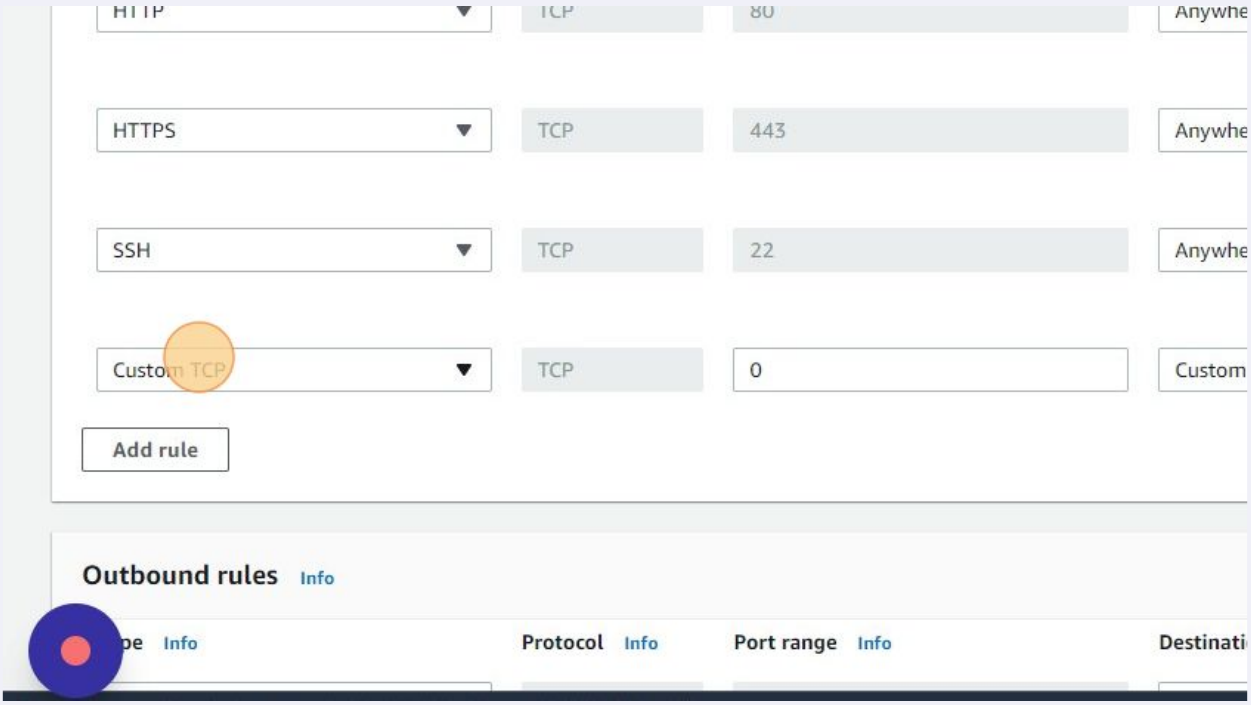
31 Click "Anywhere-IPv4"



32 Click "Add rule"

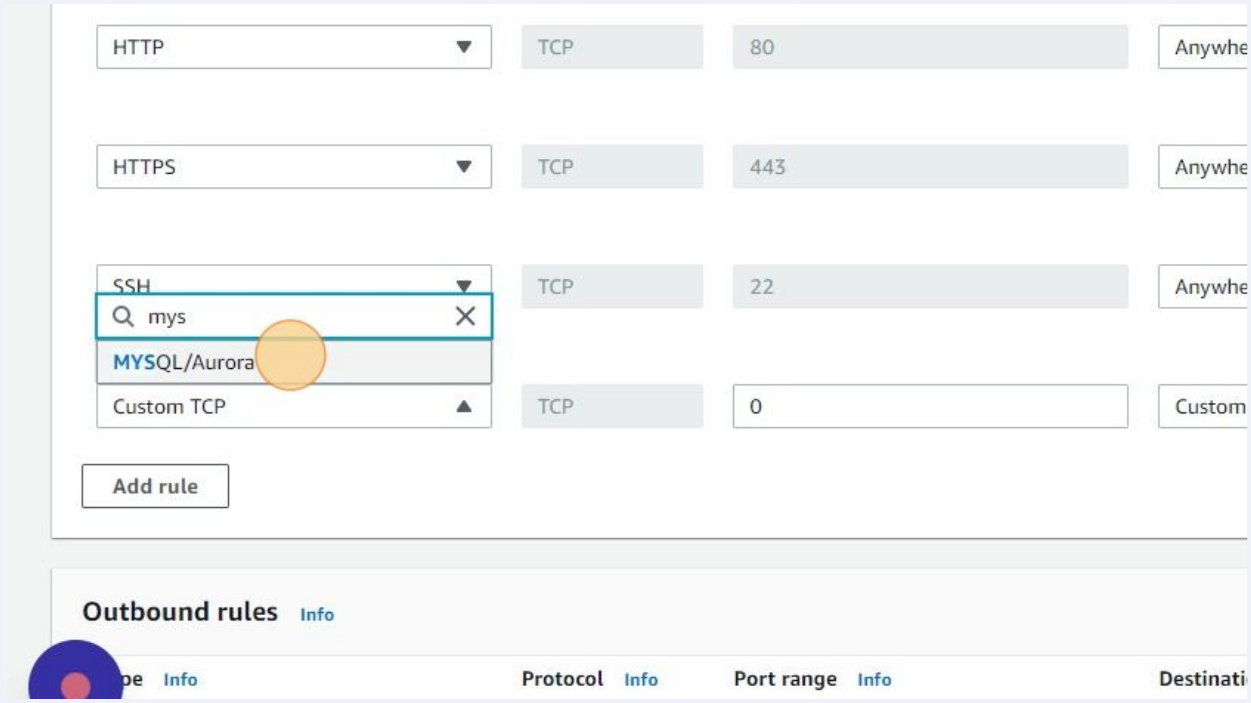


33 Click "Custom TCP"

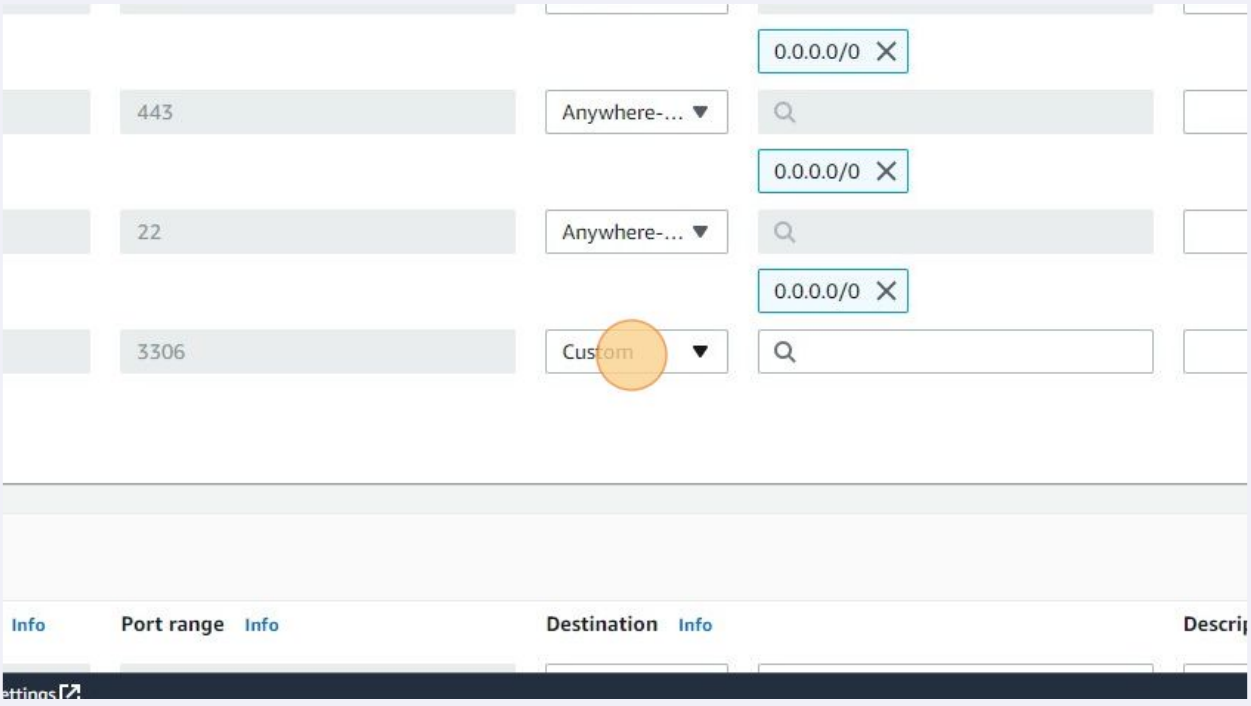


34 Type "mys"

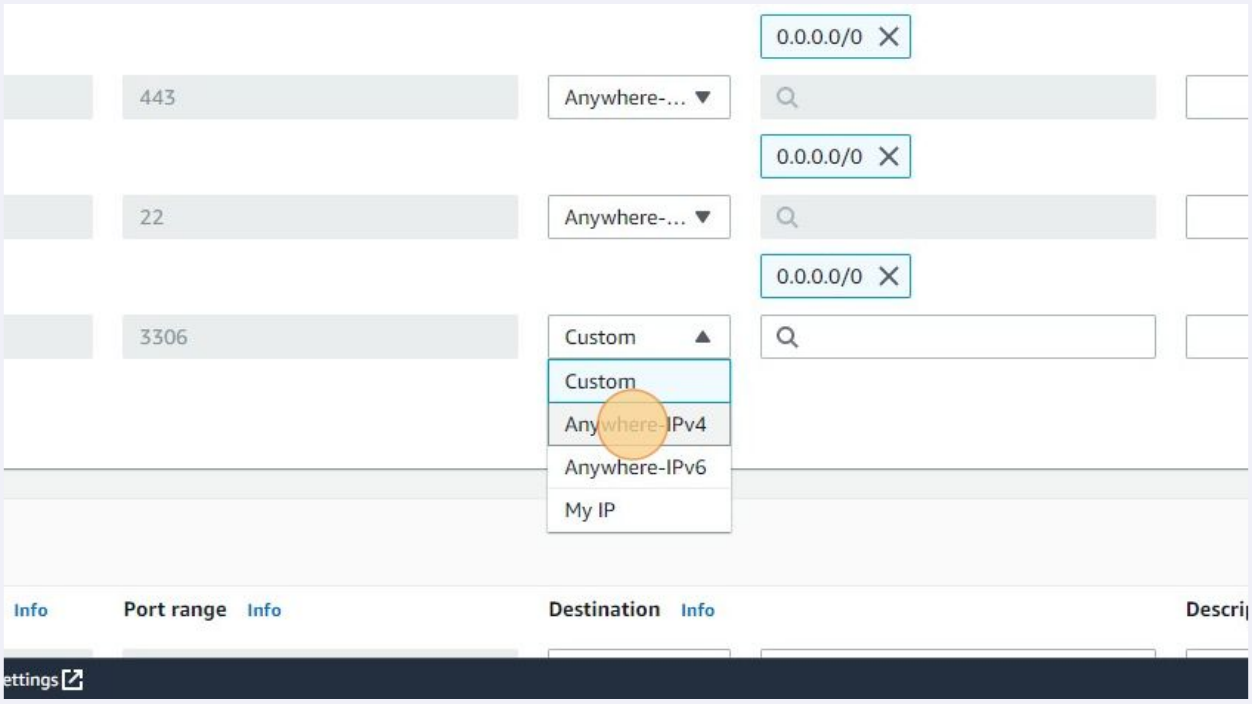
35 Click here.



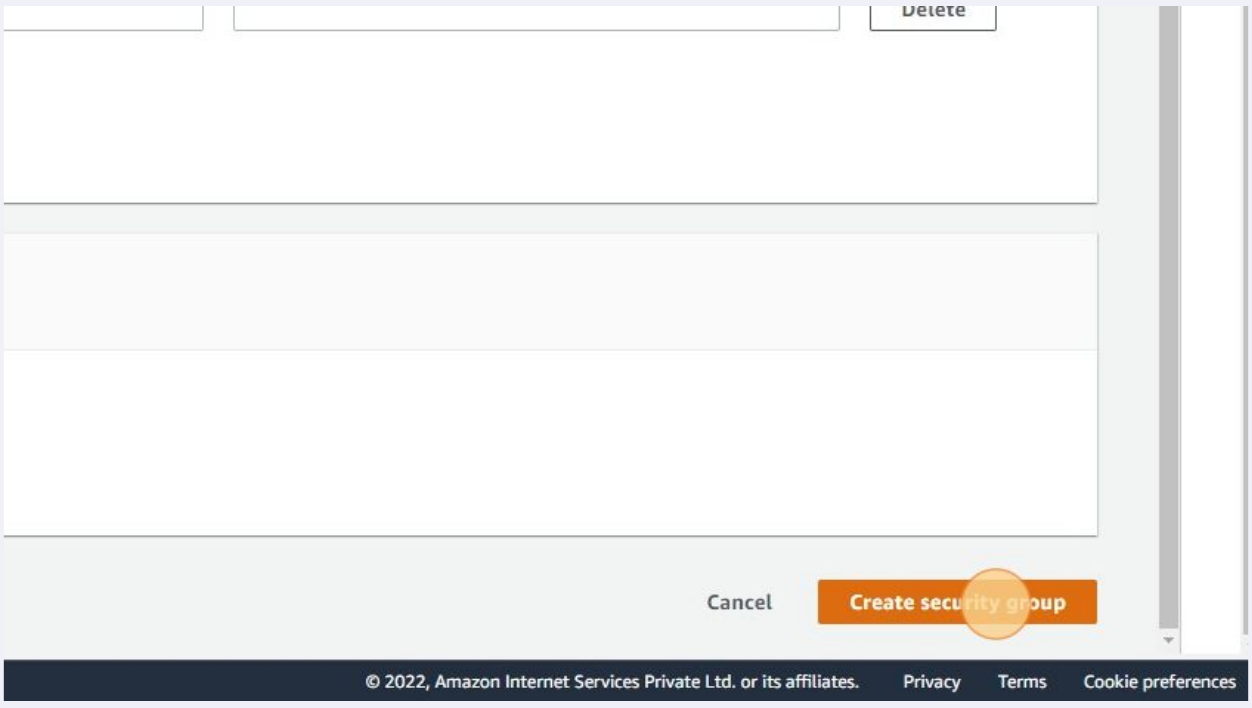
36 Click "Custom"



37 Click "Anywhere-IPv4"



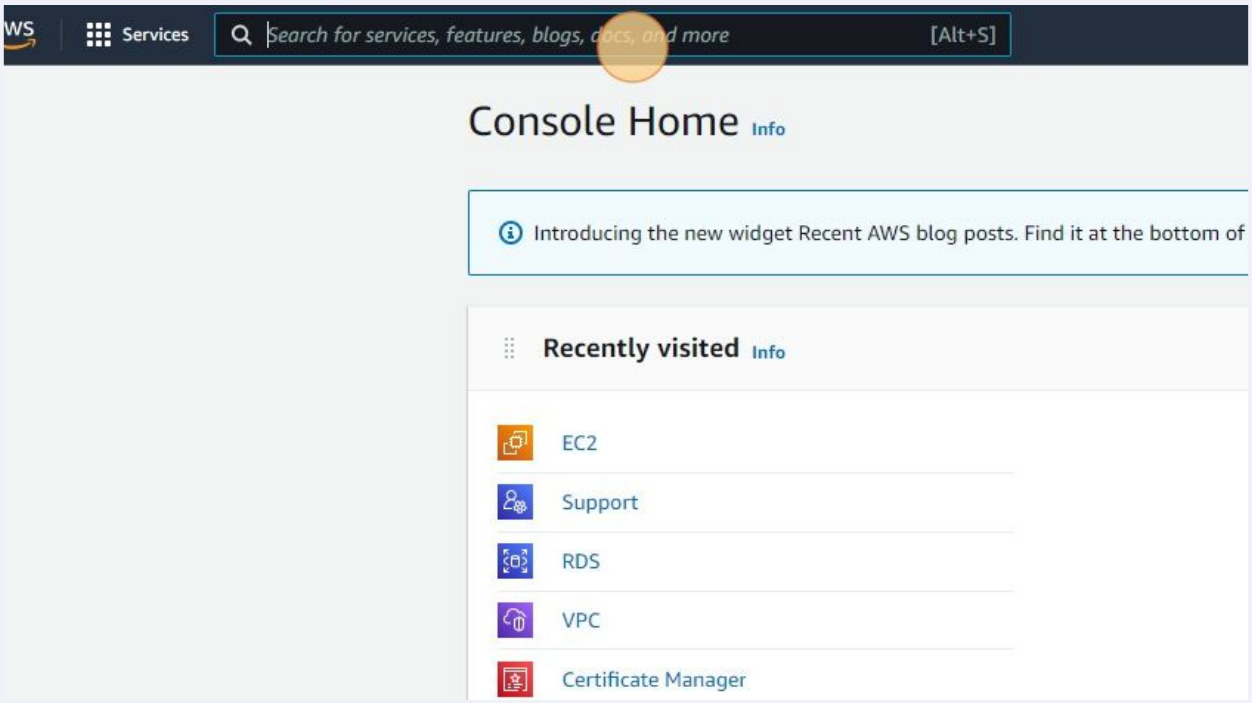
38 Click "Create security group"



4. Creating a EC2 instance in the new VPC with user-data

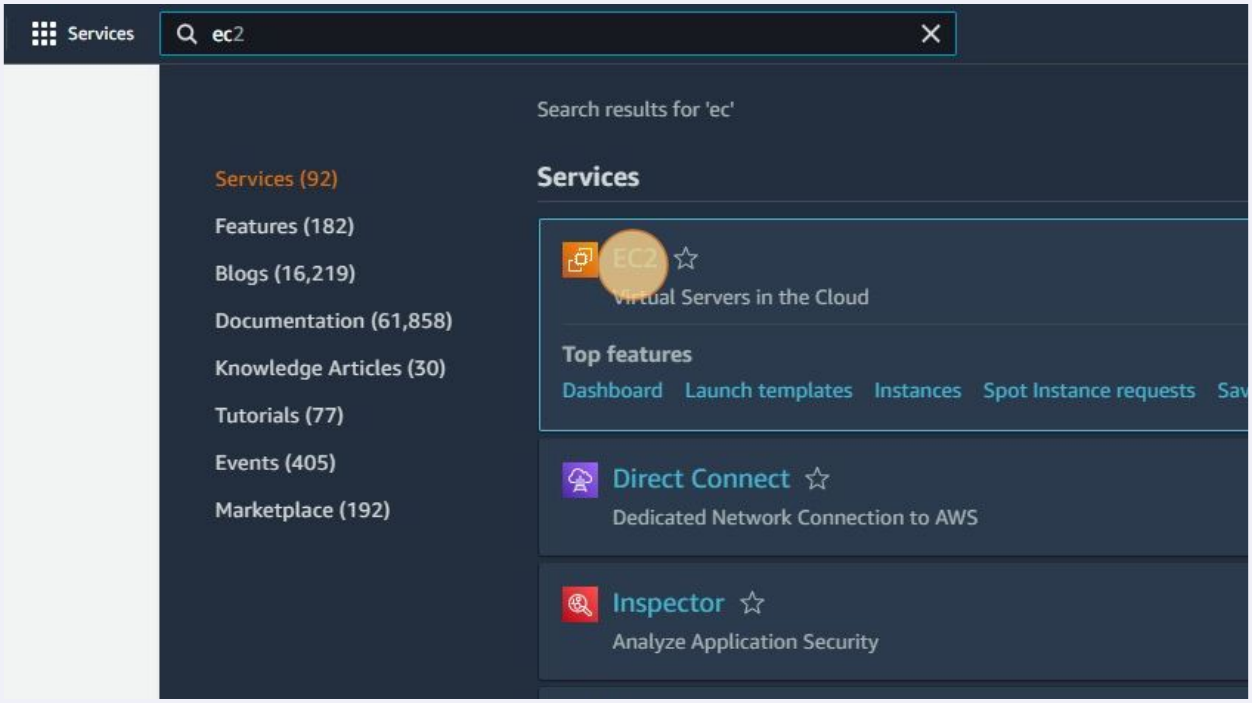
1 Navigate to https://ap-south-1.console.aws.amazon.com/console/home?region=ap-south-1

2 Click the "Search for services, features, blogs, docs, and more" field.

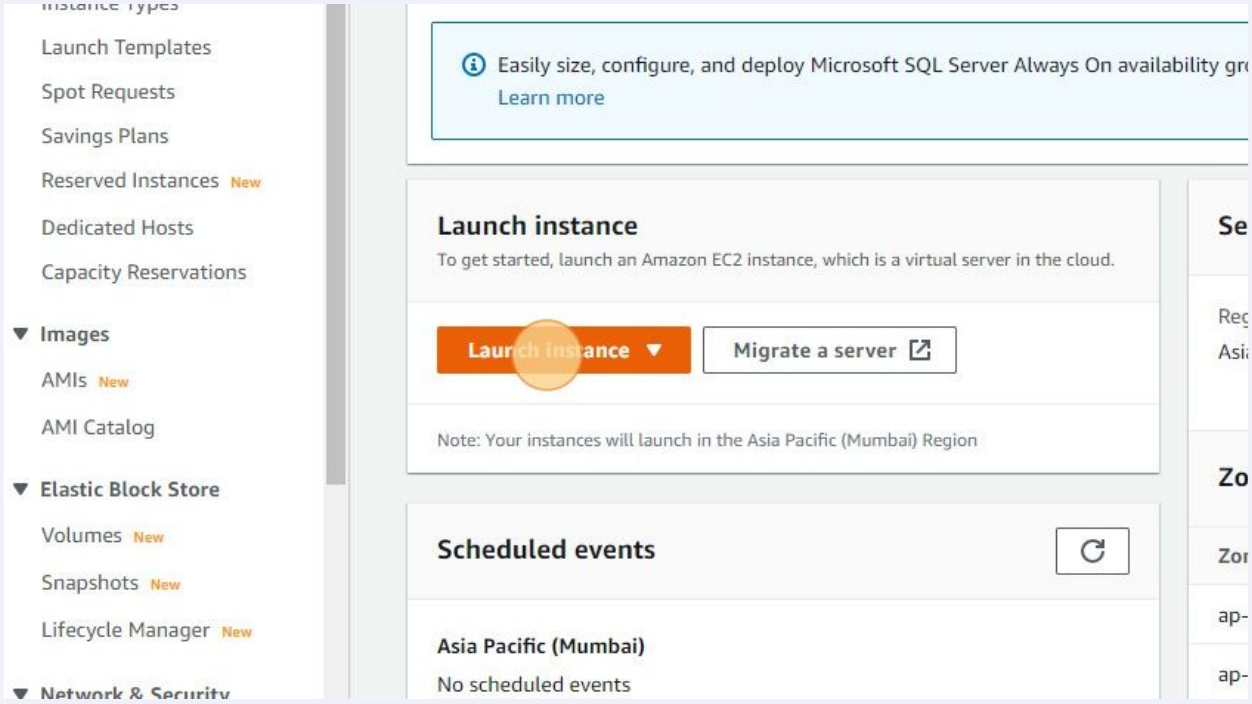


3 Search for"ece"

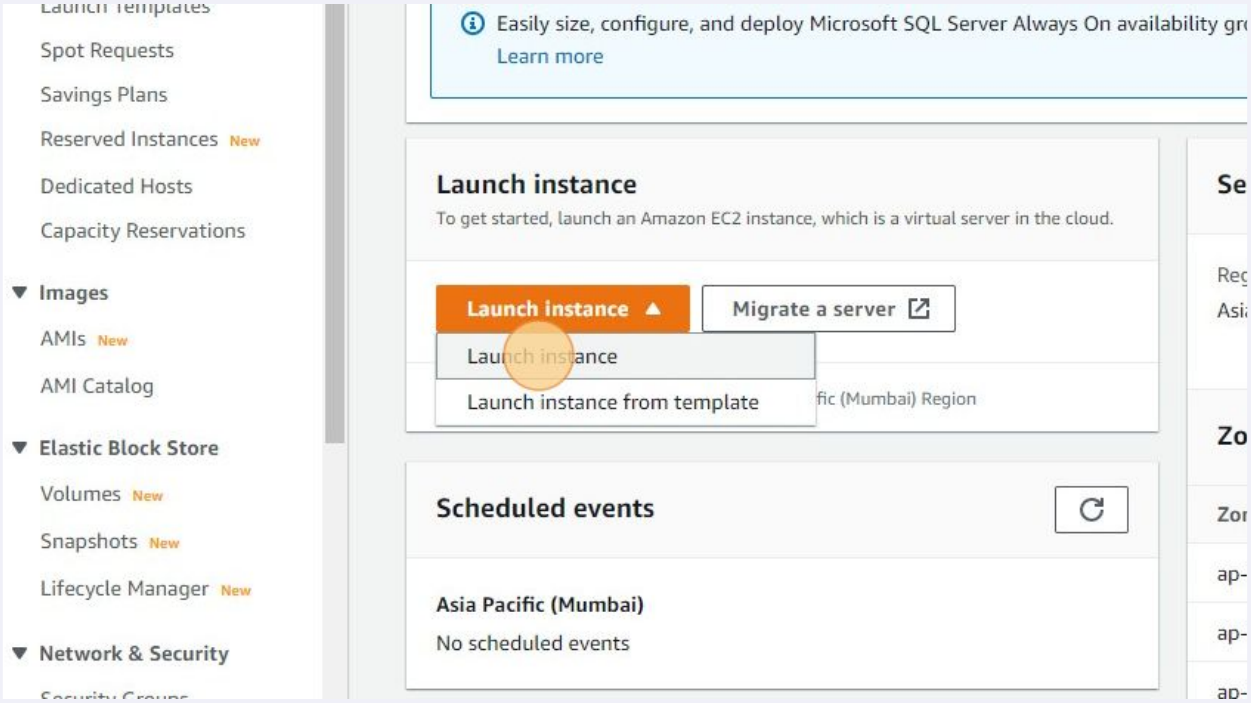
4 Click "EC2"



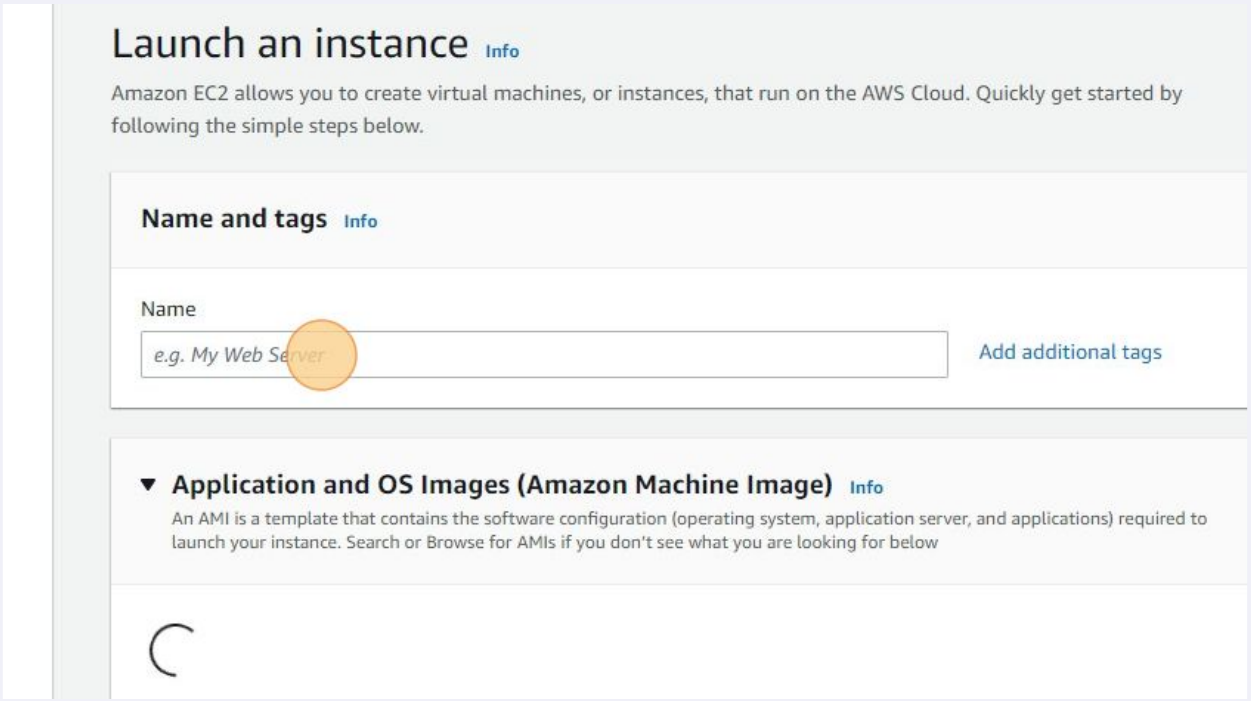
5 Click "Launch instance"



6 Click "Launch instance"

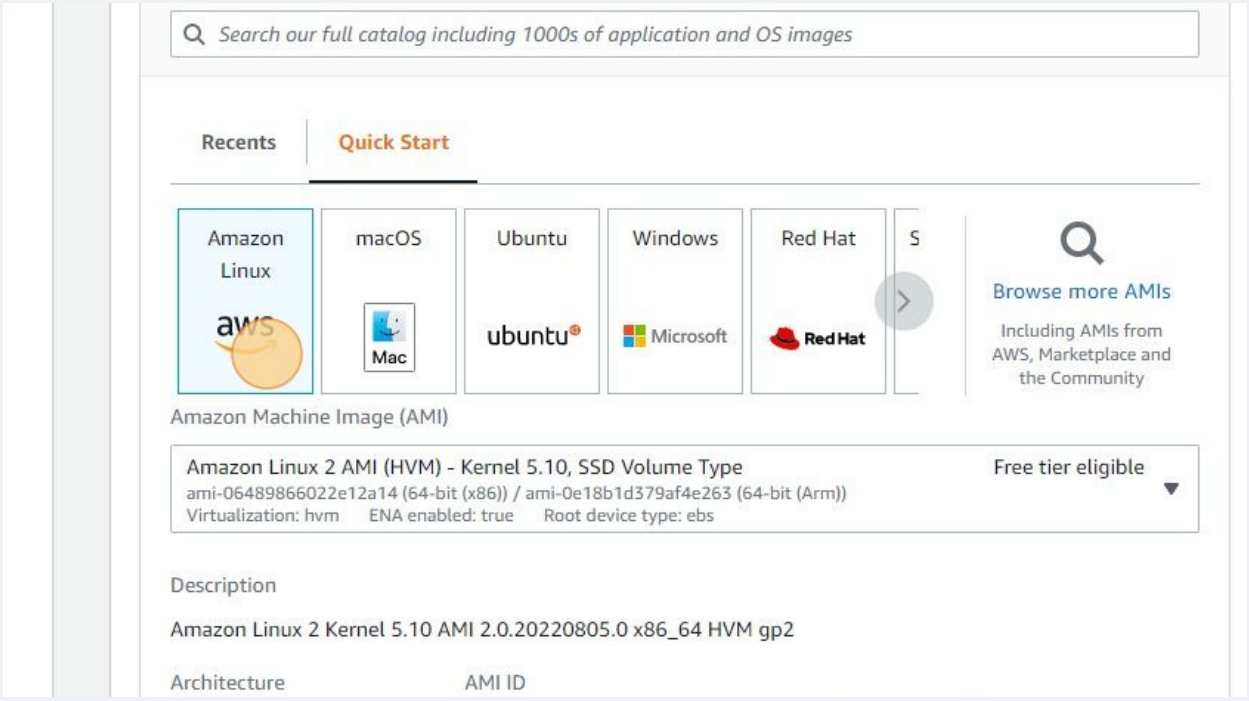


7 Click the "Name" field.

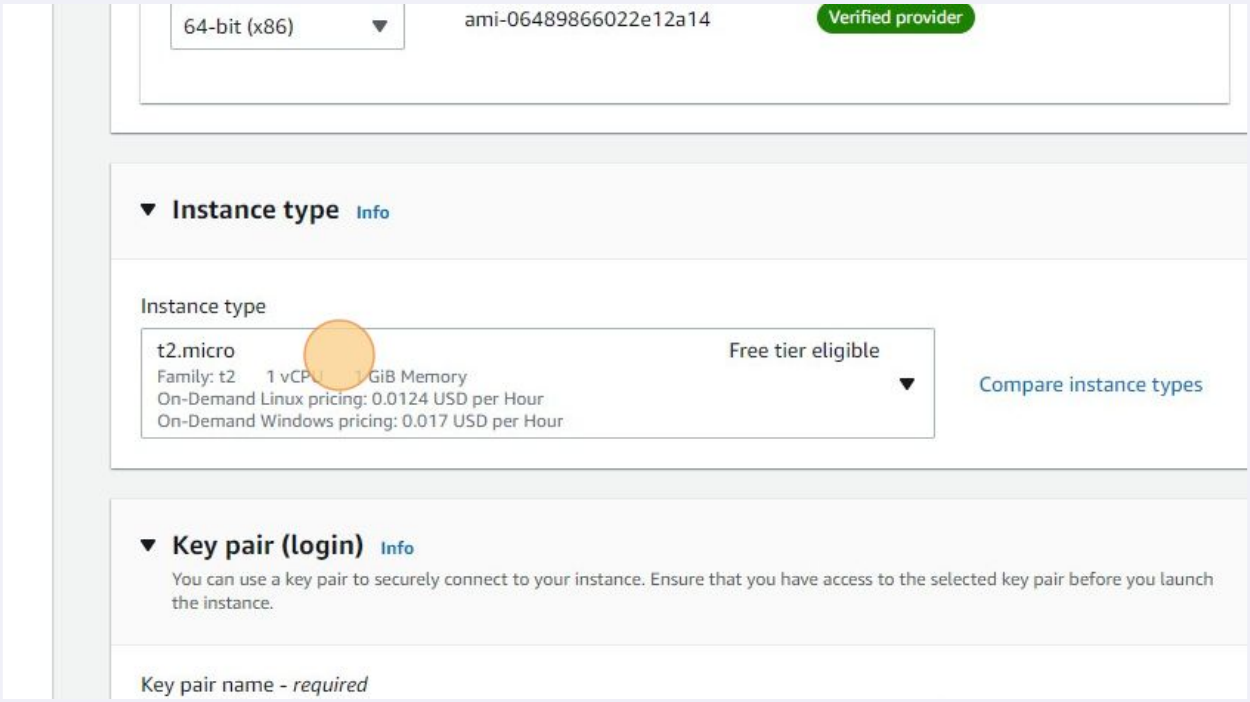


8 Type "my-server"

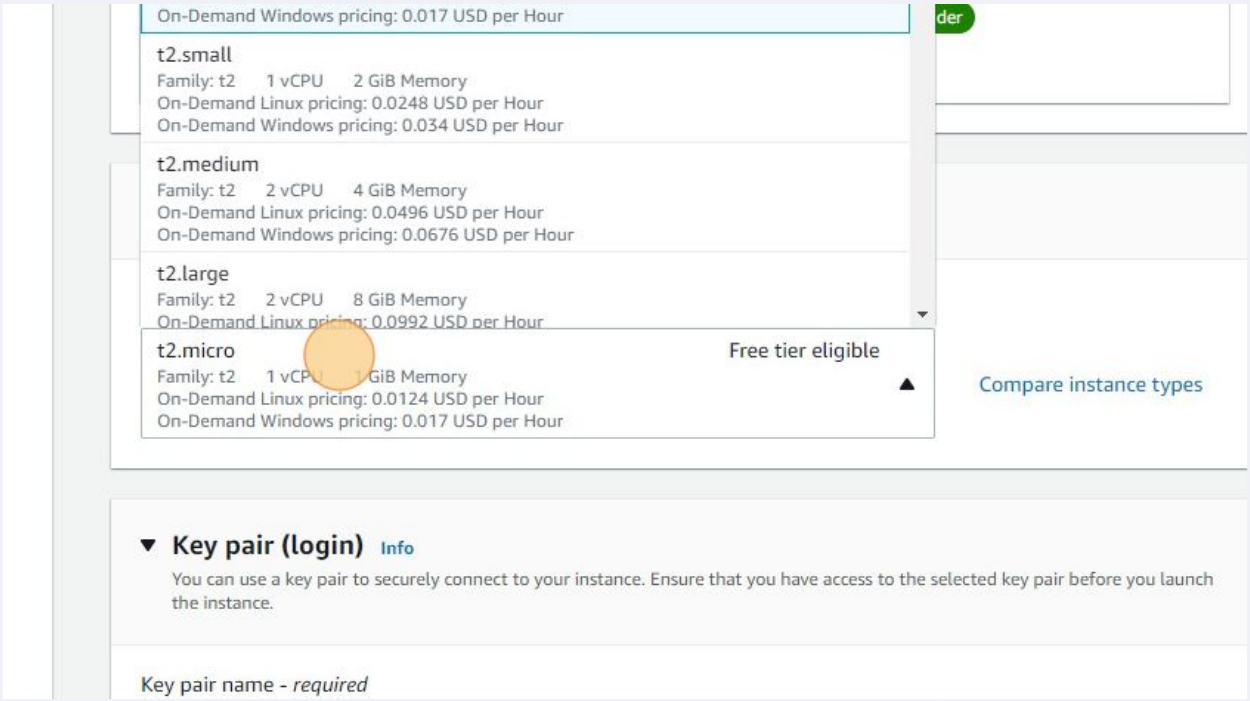
9 Click this icon.



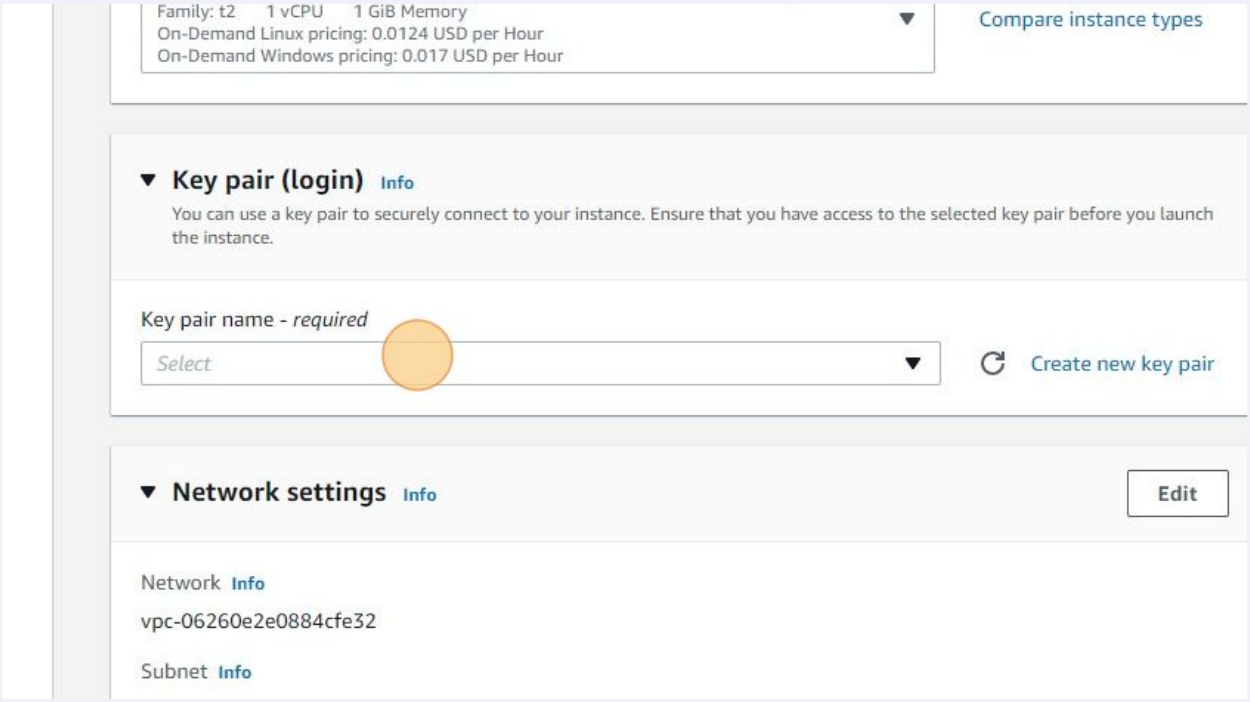
10 Click "Free tier eligible"



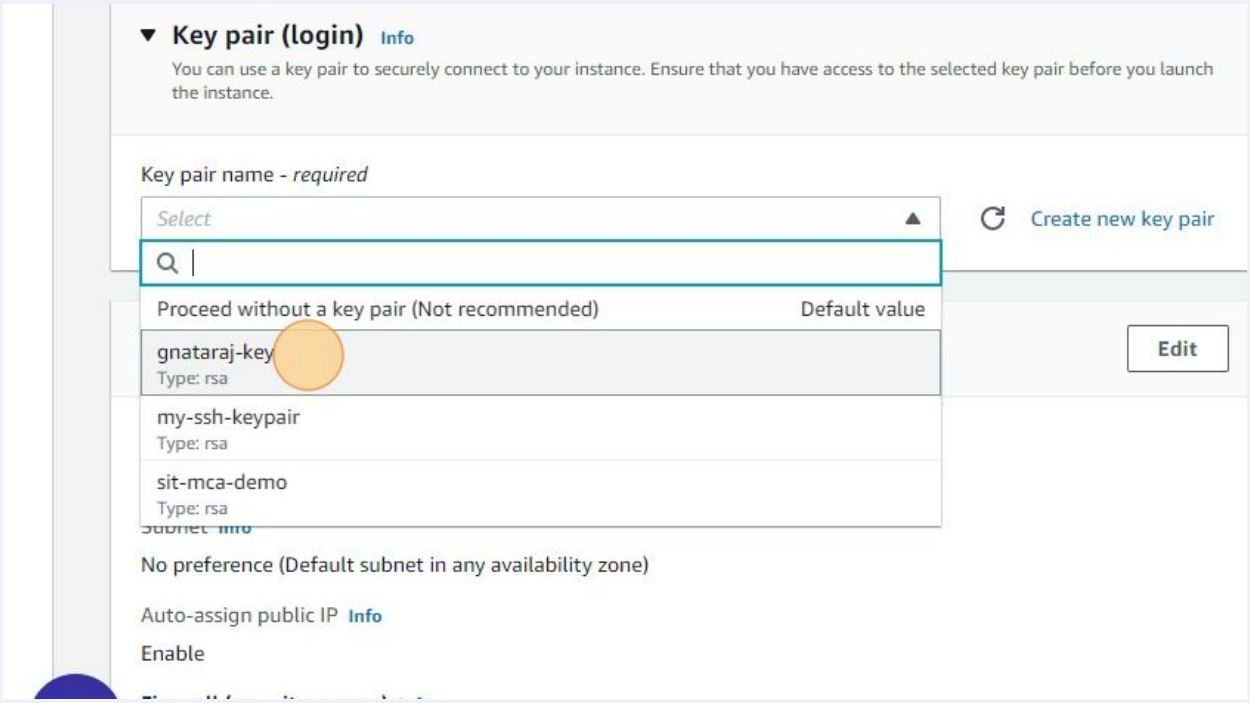
11 Click "Free tier eligible"



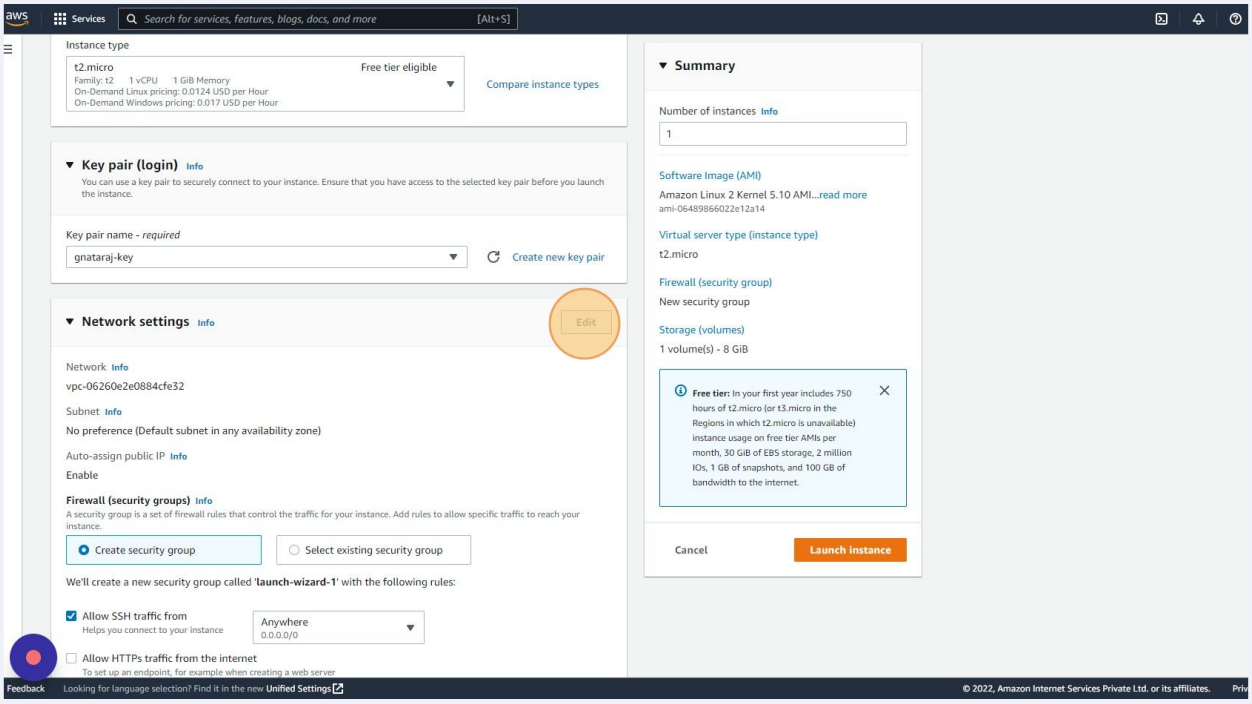
12 Click "Select"



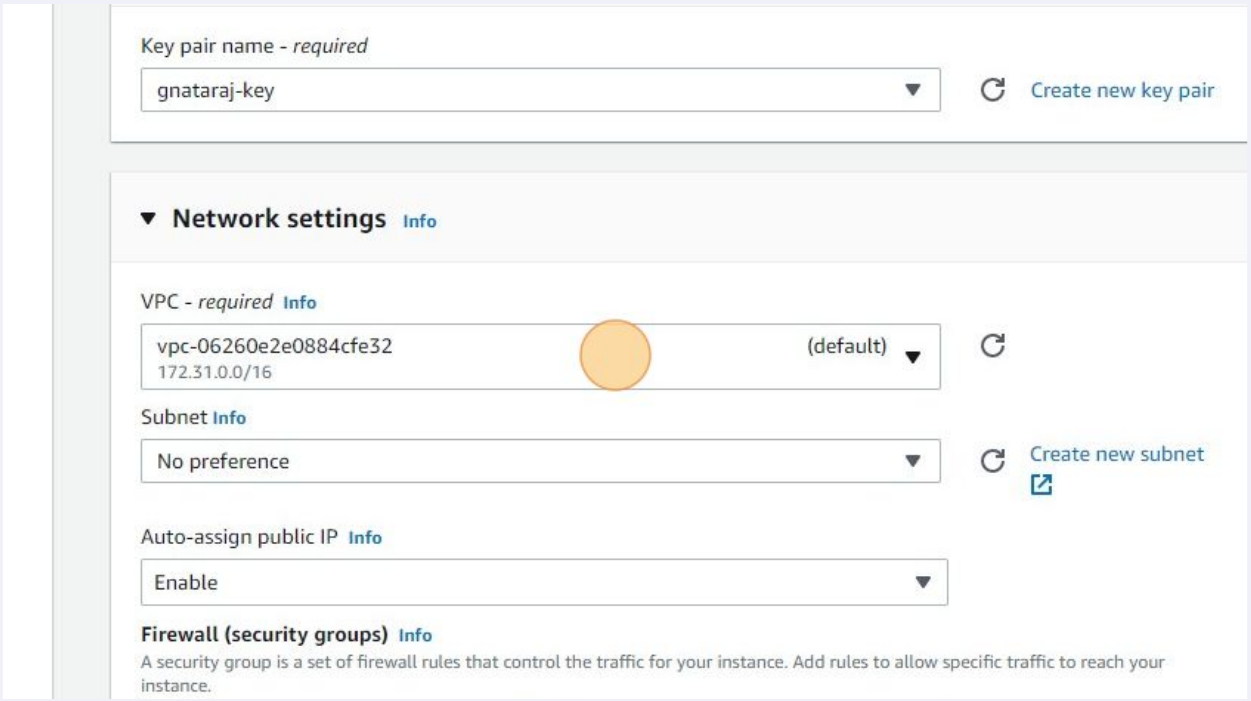
13 Select the key-pair you created earlier



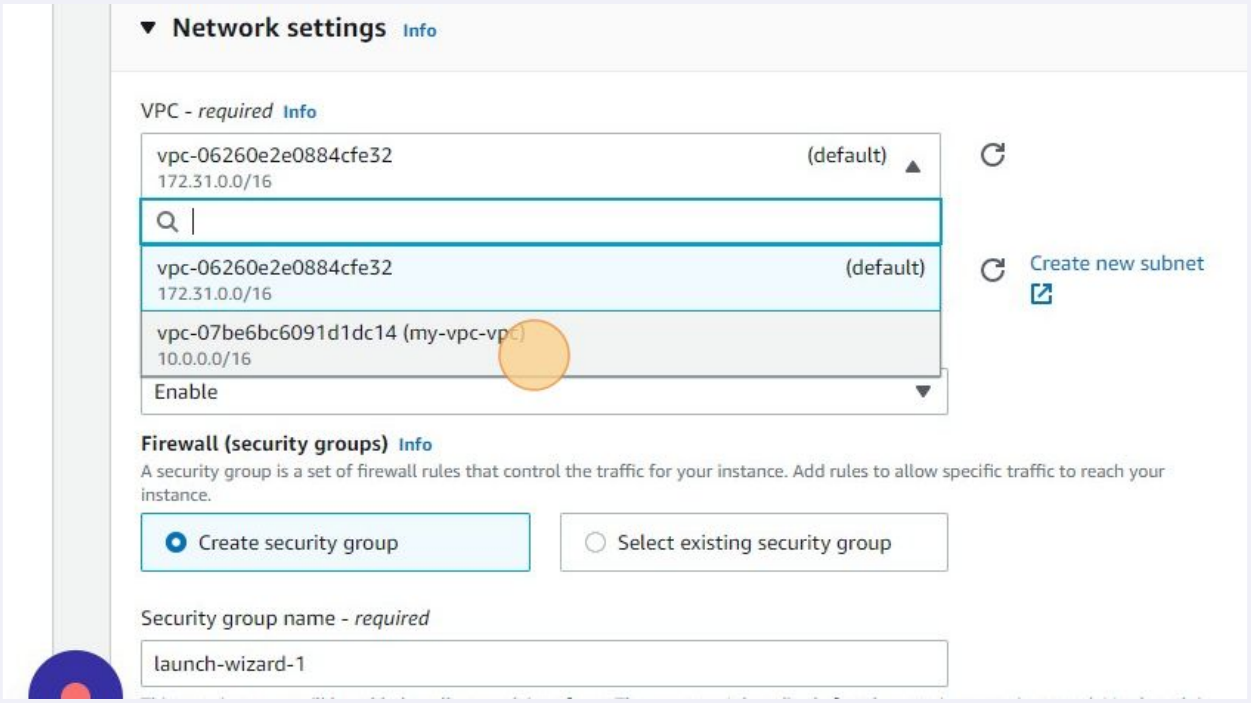
14 In the "Network Setting" Click "Edit"



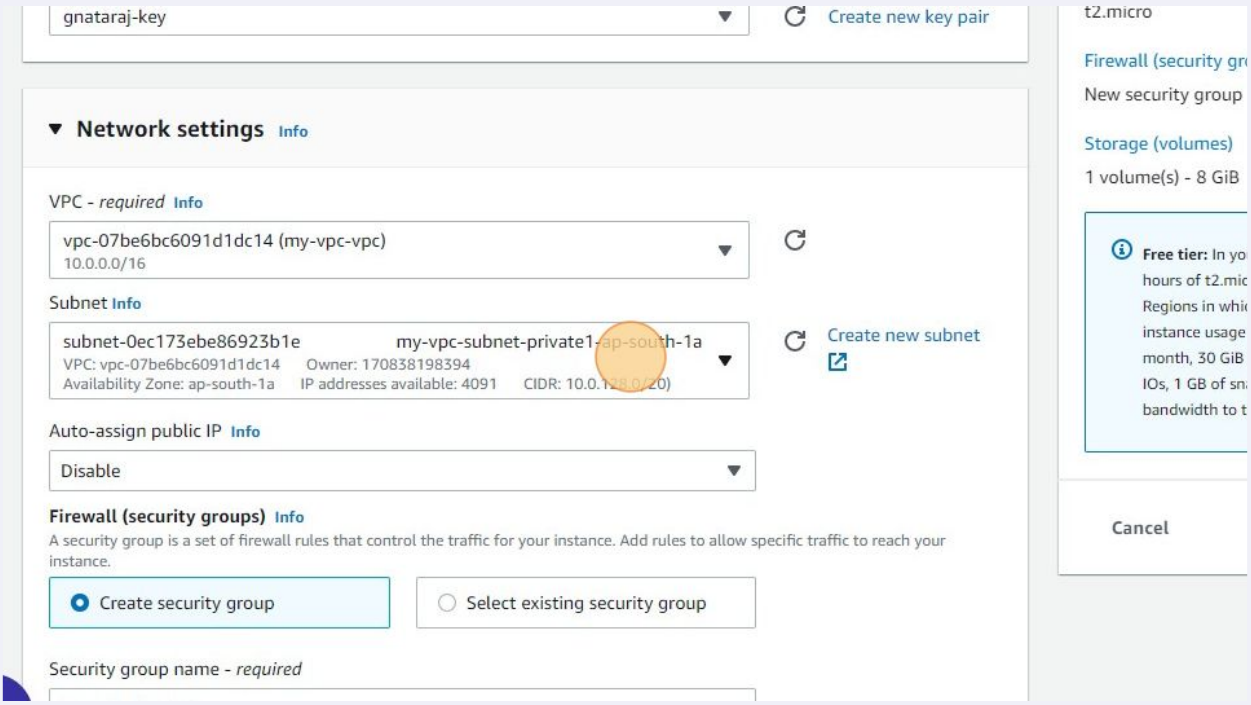
15 Click on the drop-down menu



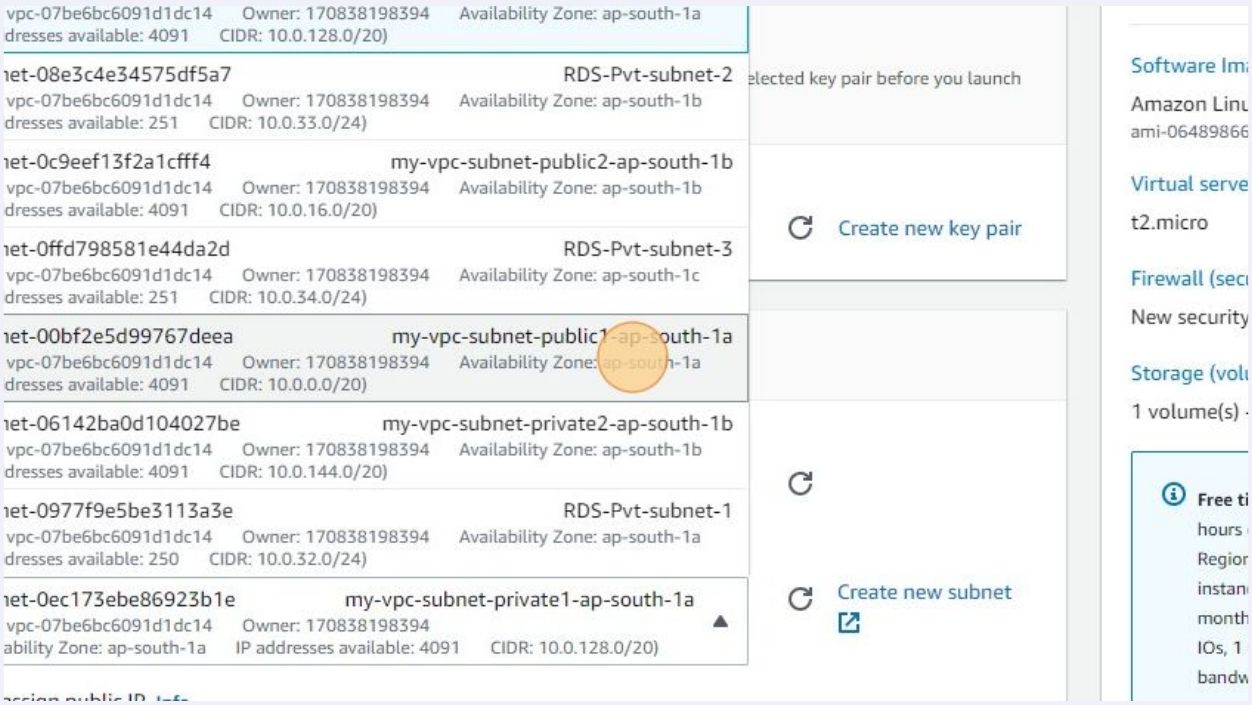
16 Select the previously created VPC



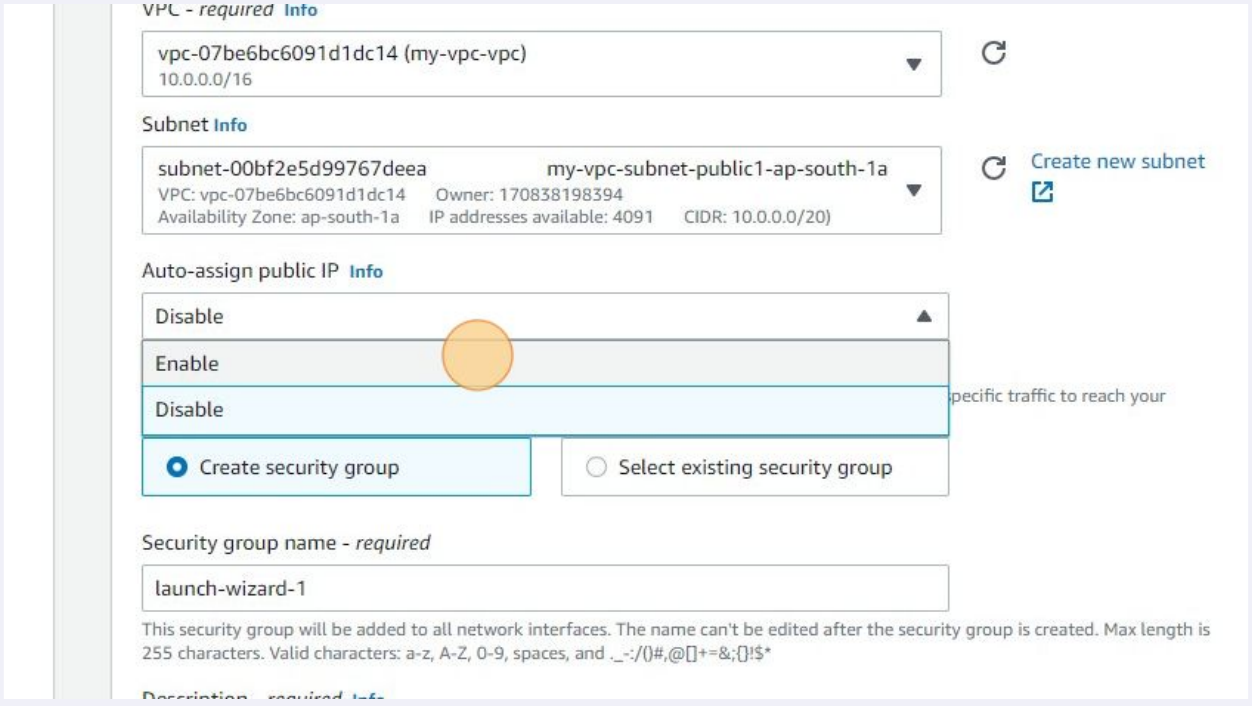
17 In the Subent dropdown,



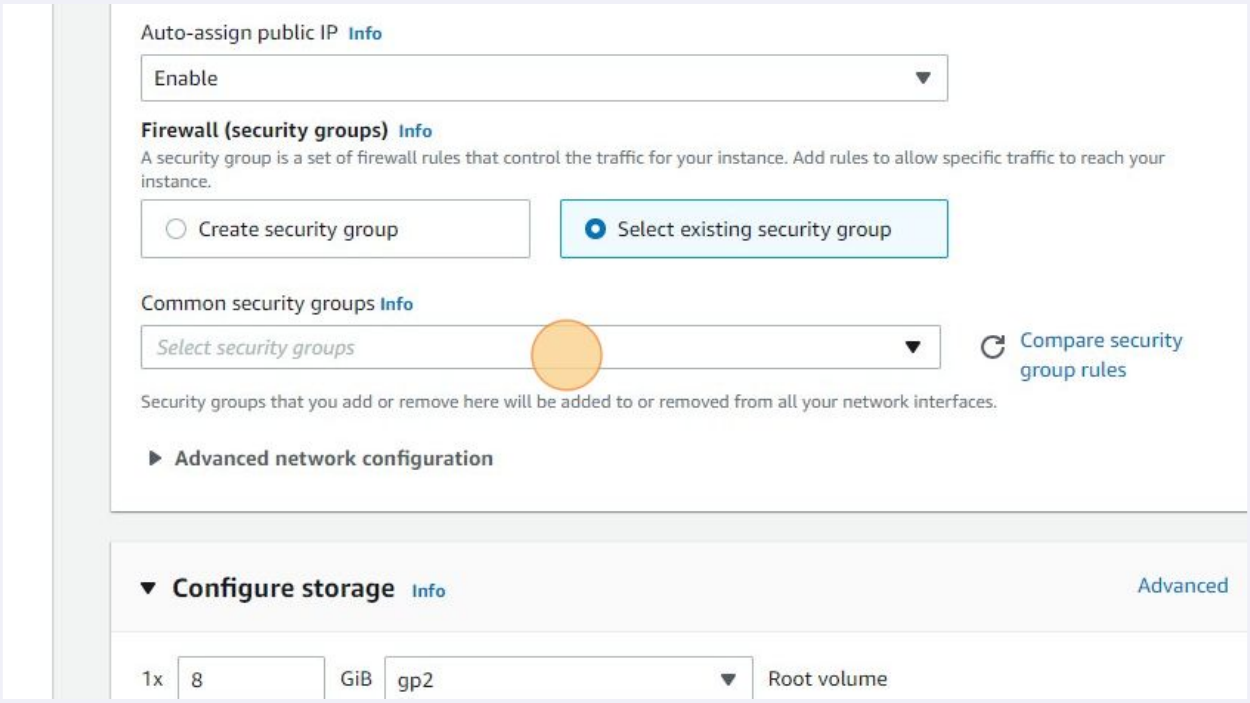
18 Select any of the Public Subnets



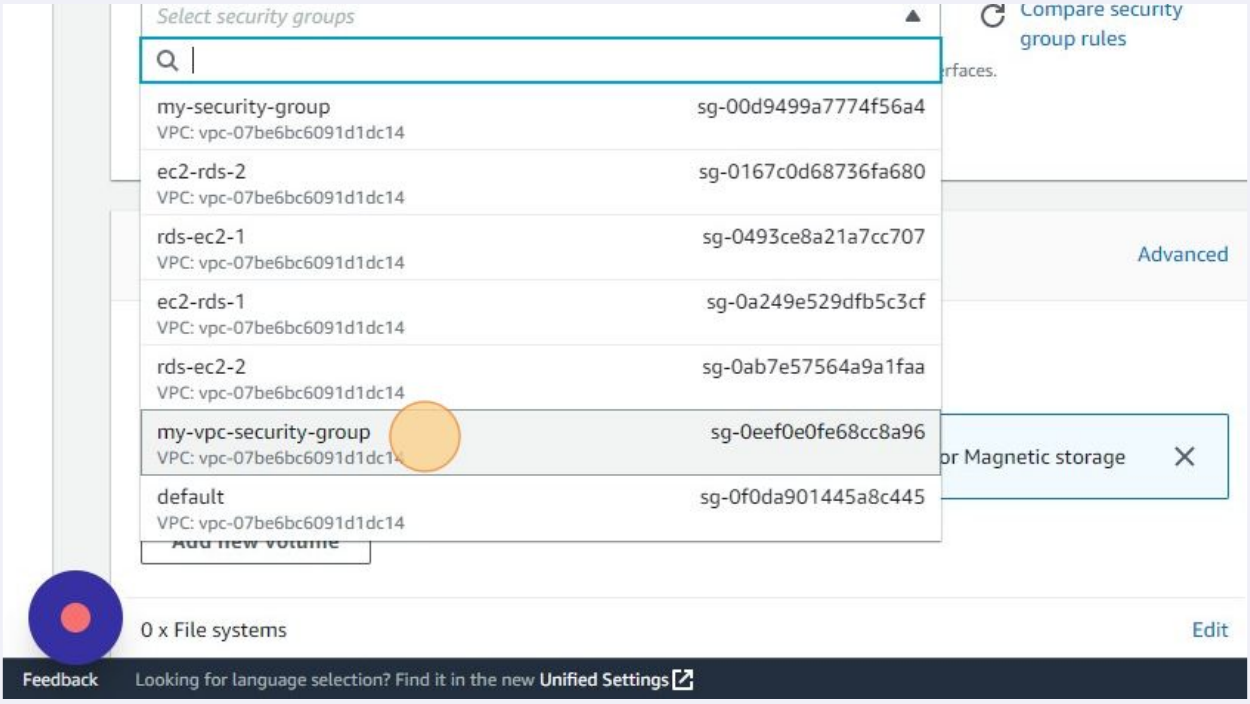
19 Select "Enable" for the Auto-Assign public IP



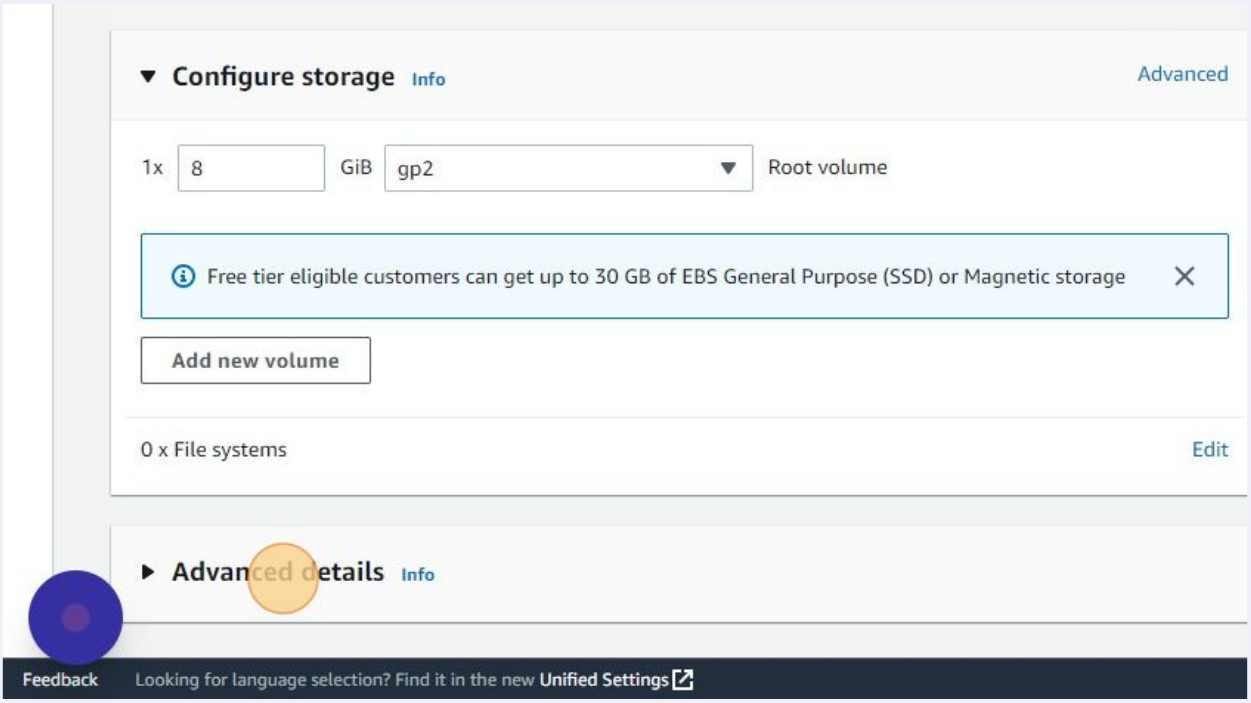
20 Click "Select security groups" In the Firewall selection



21 Click the previously created security group.



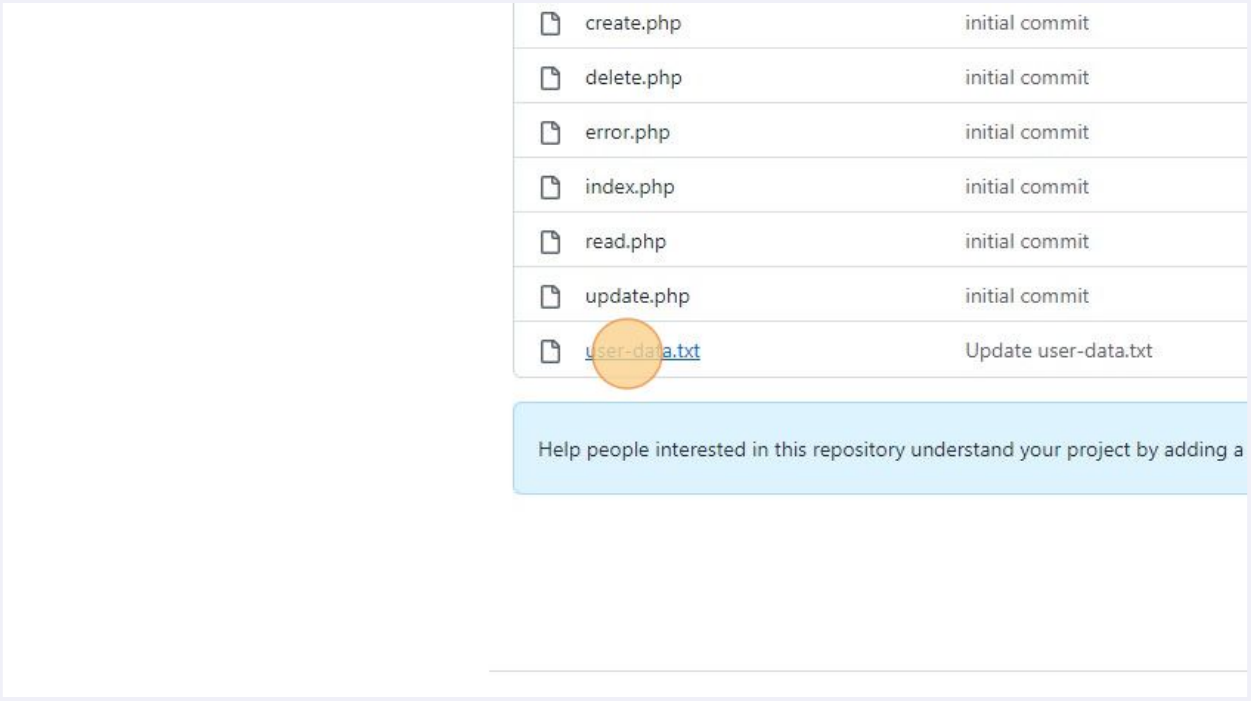
22 Click "Advanced details"



23 Click this field.

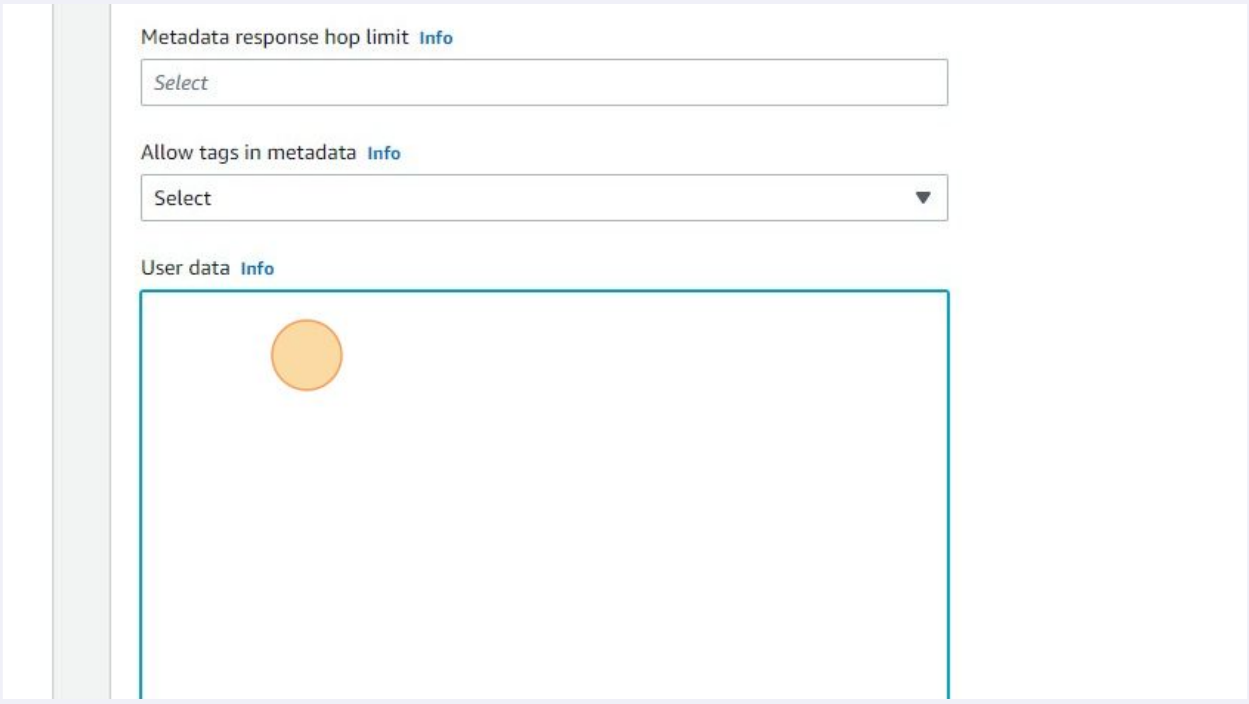
24 On a different tab, access the https://github.com/gnataraj/php-mysql in a

25 Click "user-data.txt"

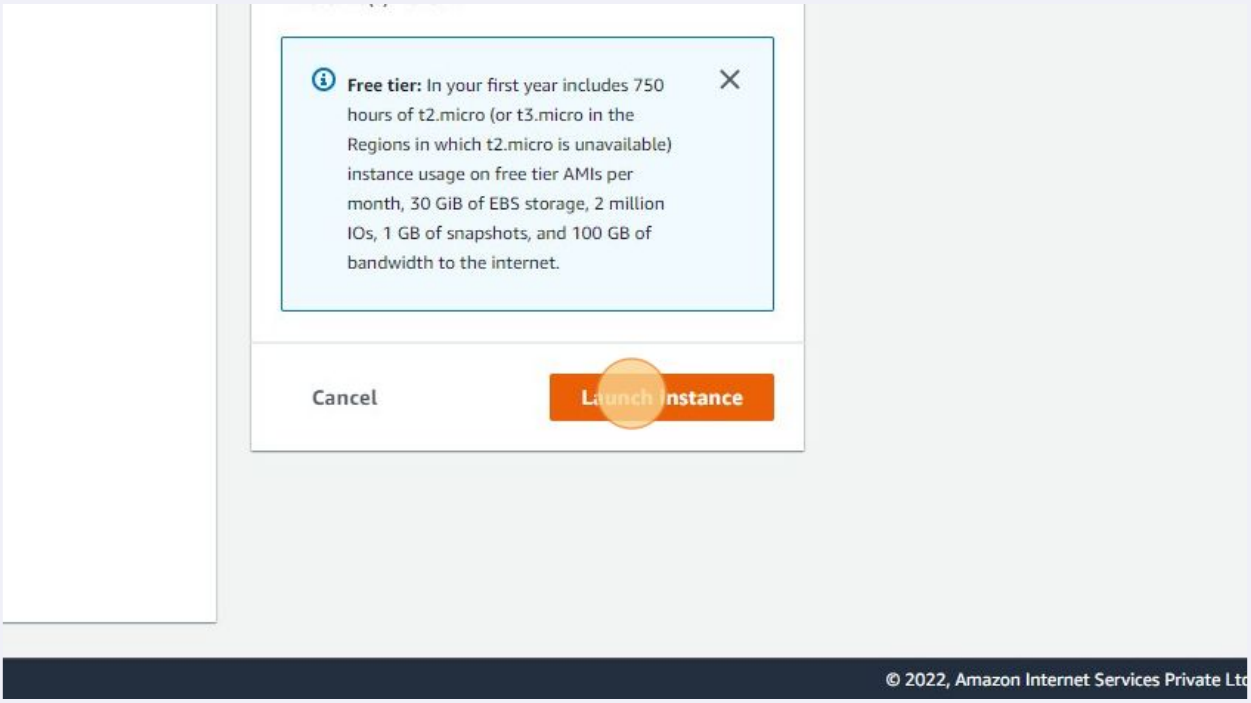


26 Copy the complete content

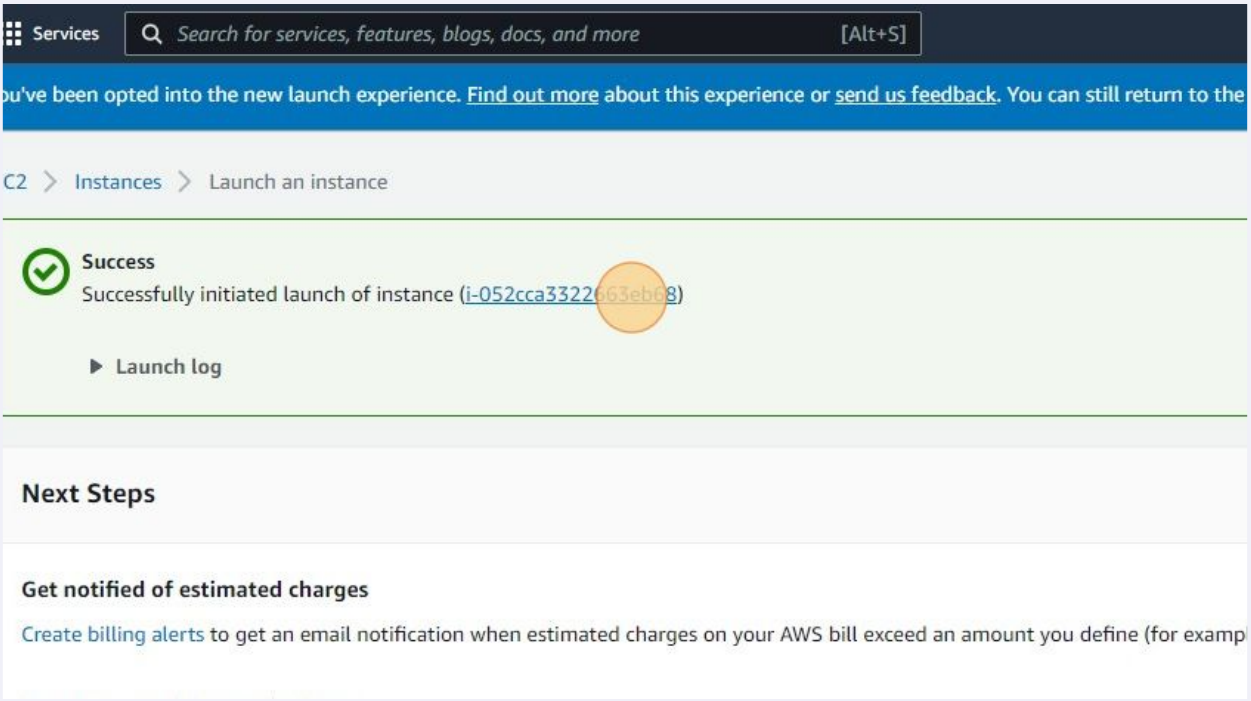
27 Paste the content in the user data



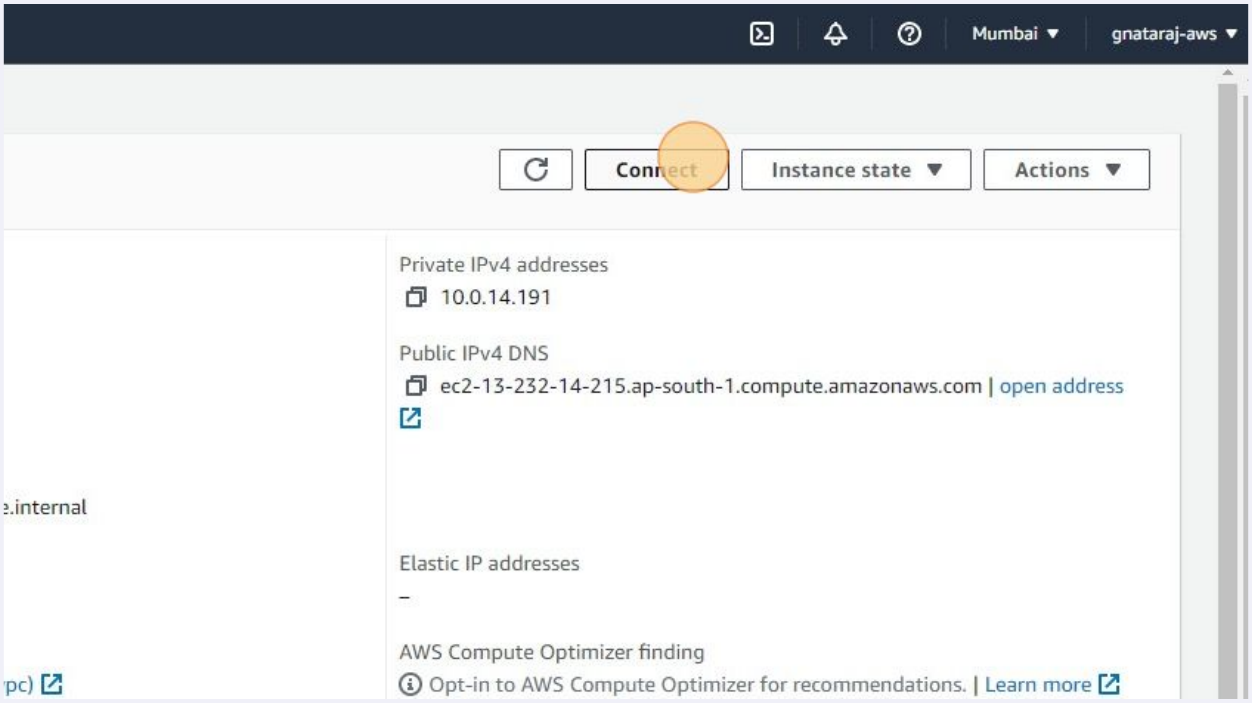
28 Click "Launch instance"



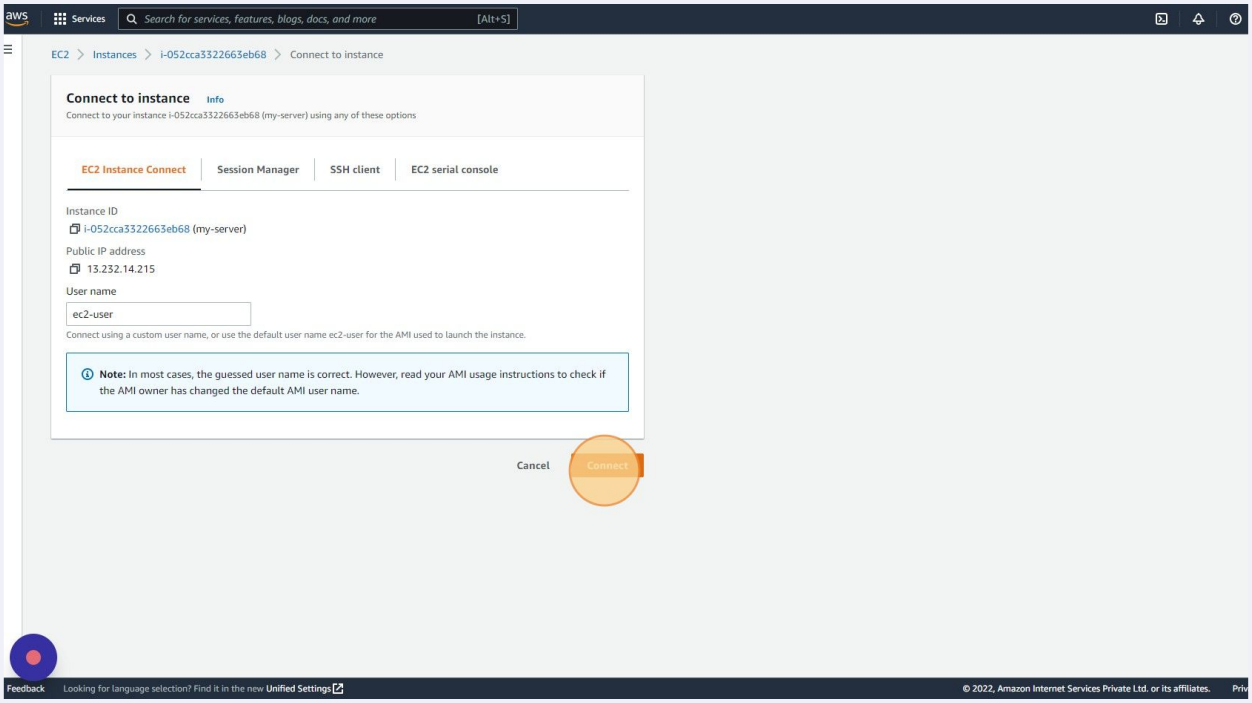
29 Click on the instance



30 Click "Connect"



31 Select the EC2 Instance Connect and click Connect

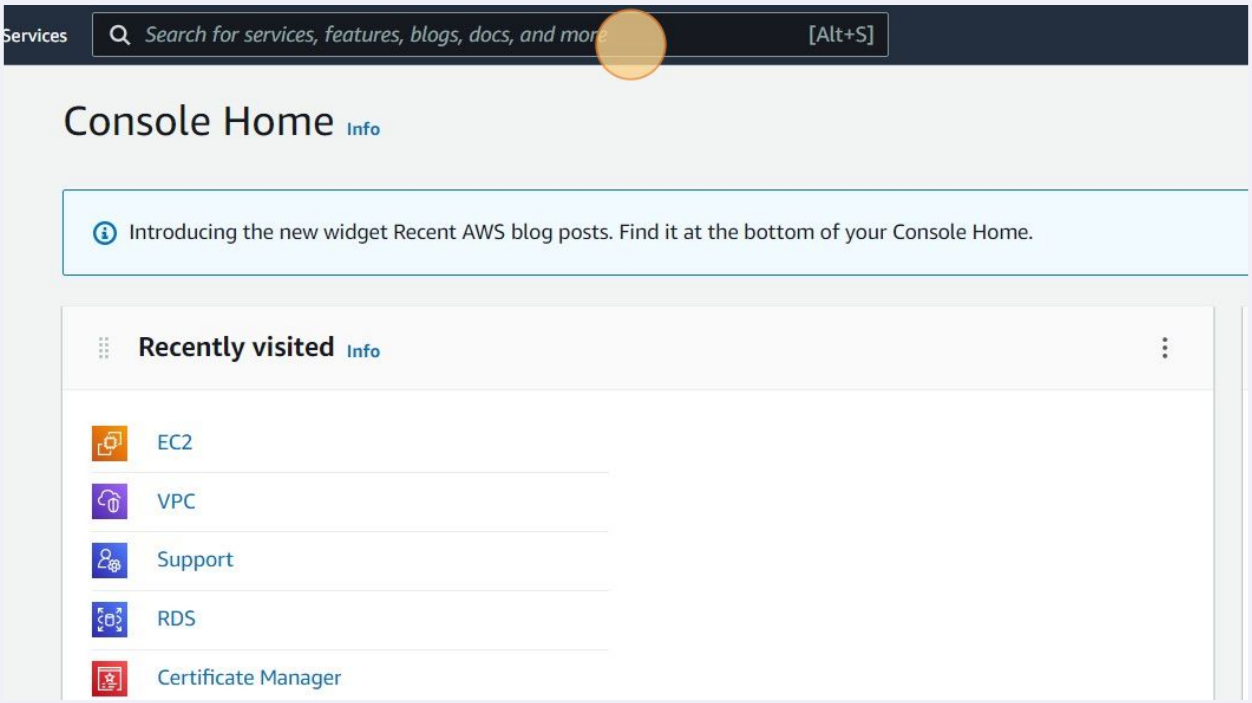


32 You will be provided with the EC2 instance Shell

5. Setting up RDS MySQL DB instance in the new VPC

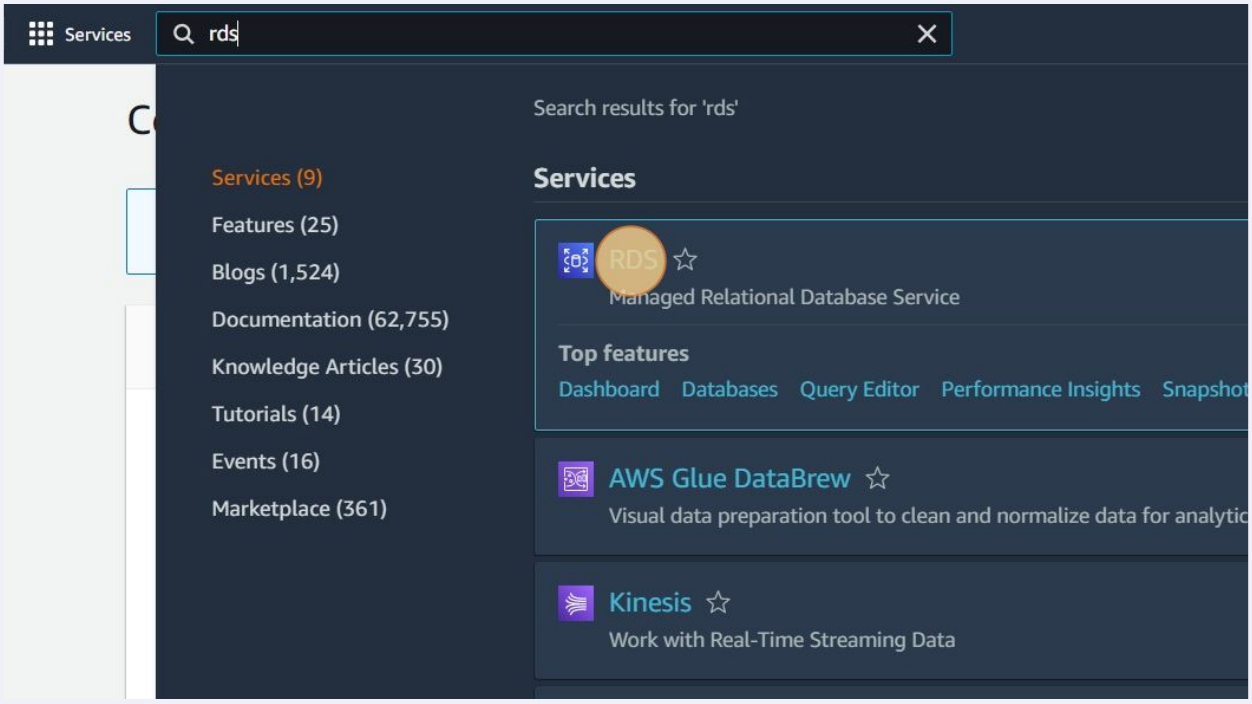
1 Navigate to https://ap-south-1.console.aws.amazon.com/console/home?region=ap-south-1

2 Click the "Search for services, features, blogs, docs, and more" field.

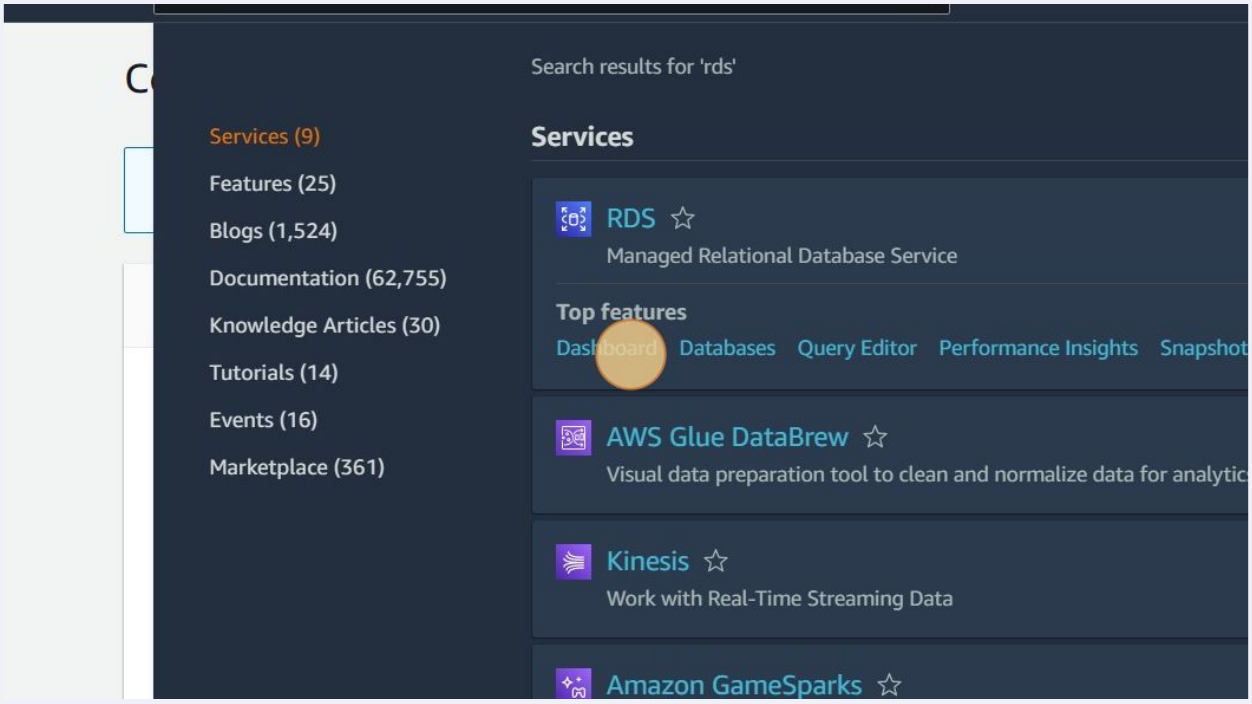


3 Type "rds"

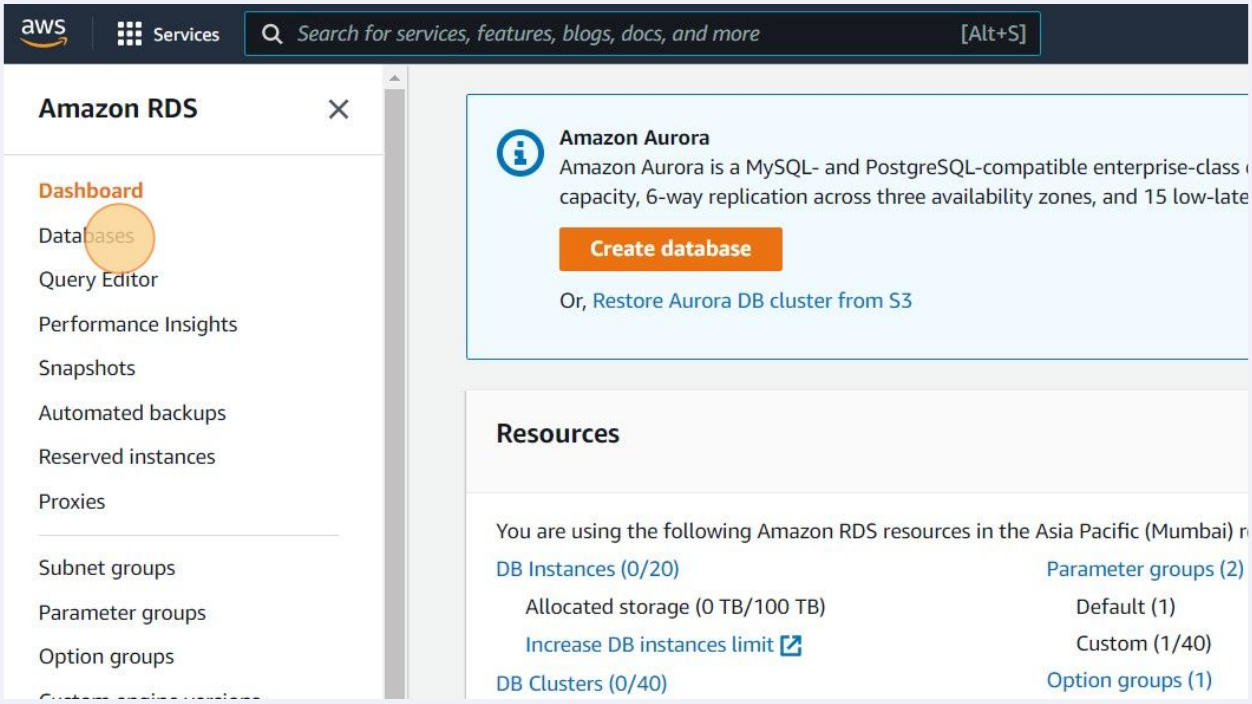
4 Click "RDS"



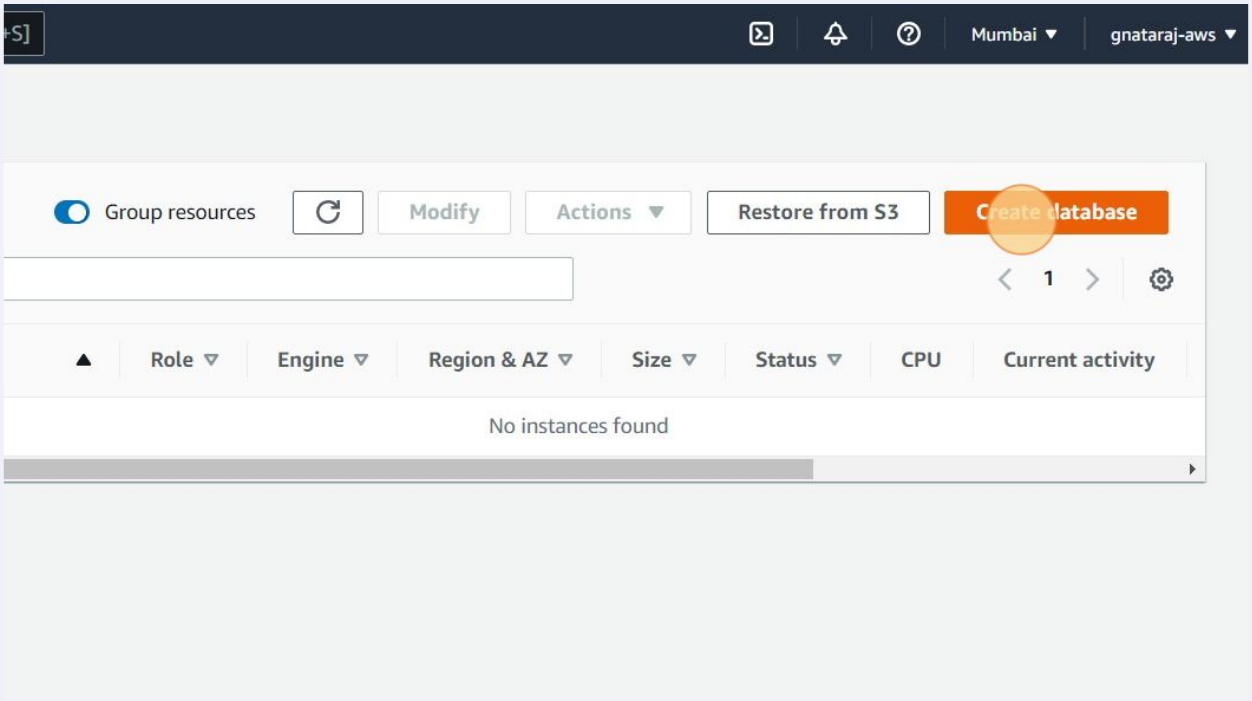
5 Click "Dashboard"



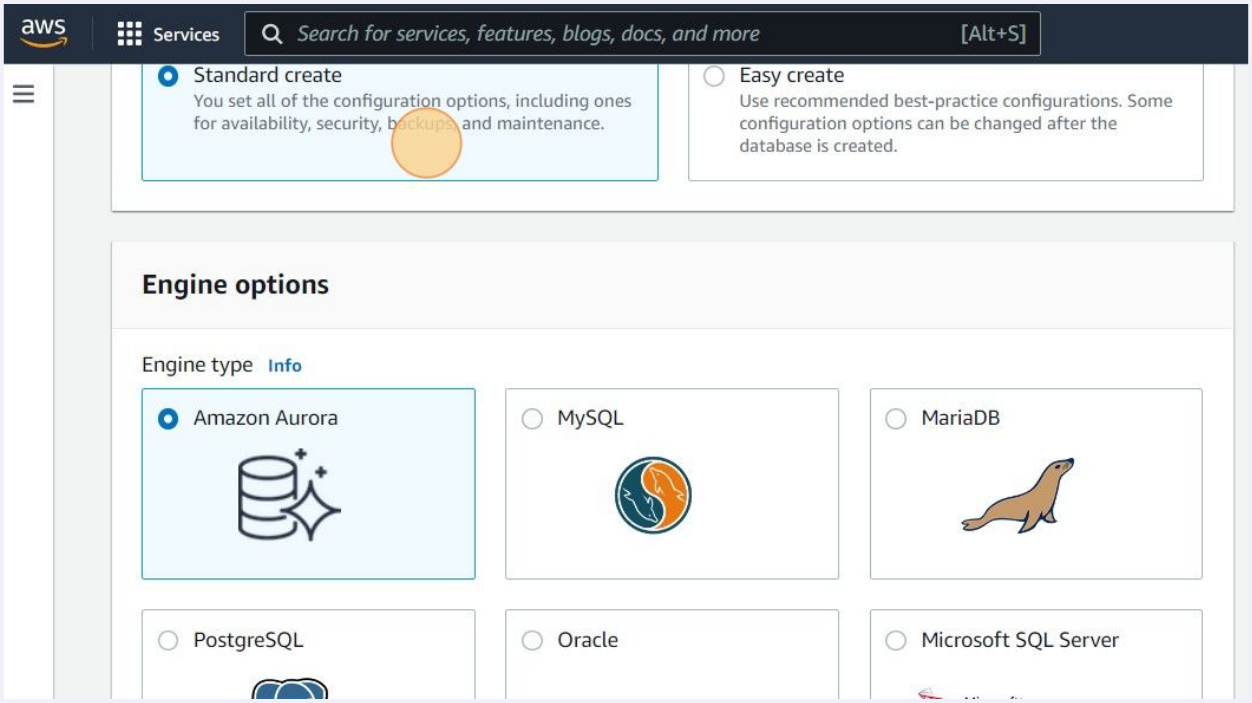
6 Click "Databases"



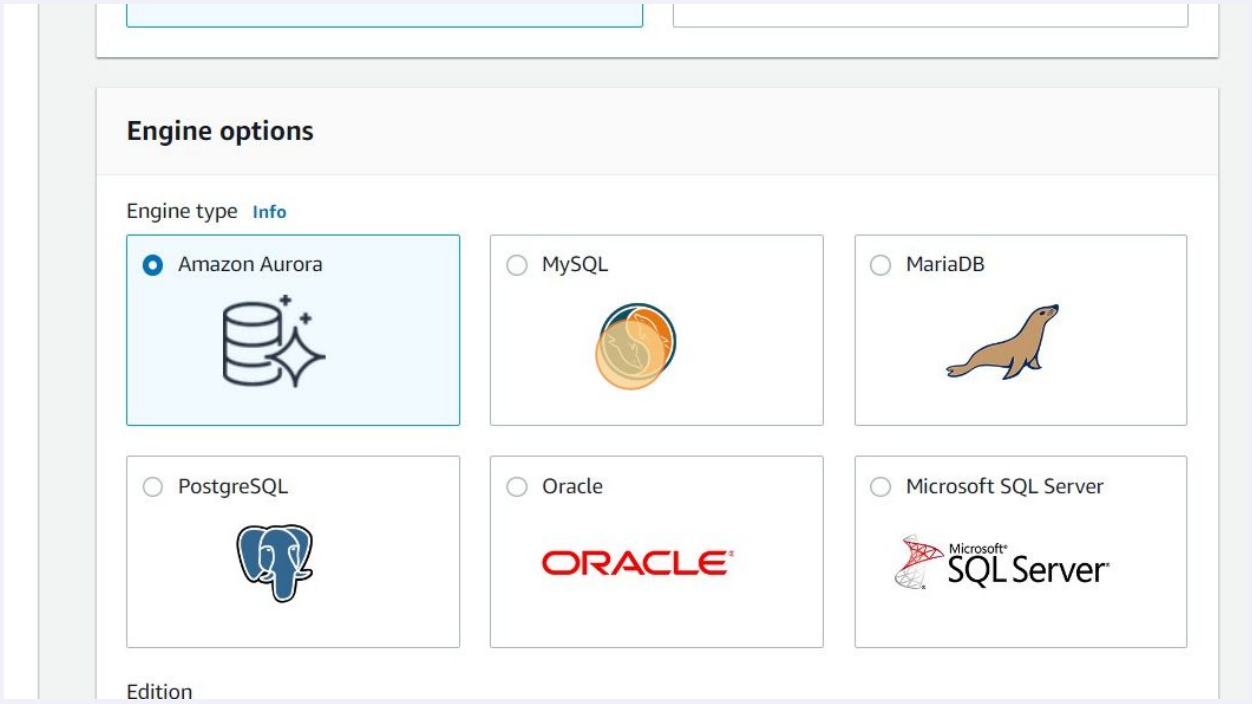
7 Click "Create database"



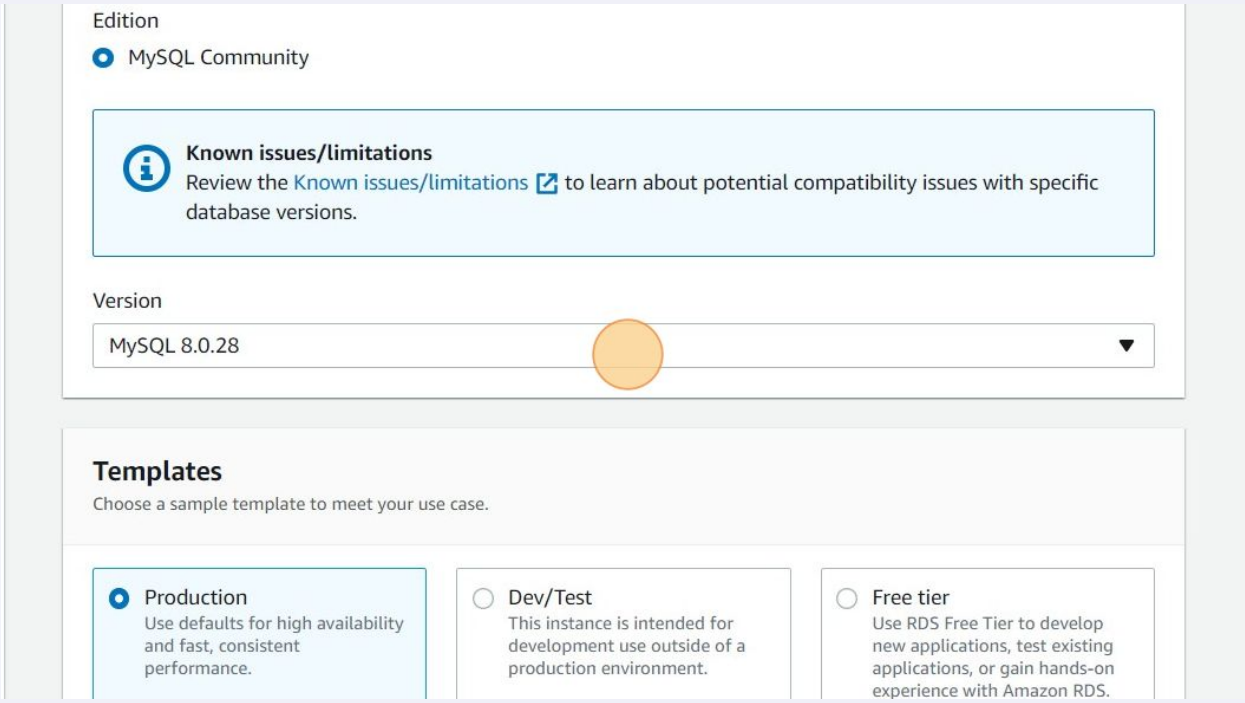
8 Click Standard Create - "You set all of the configuration options, including ones for availability, security, backups, and maintenance."



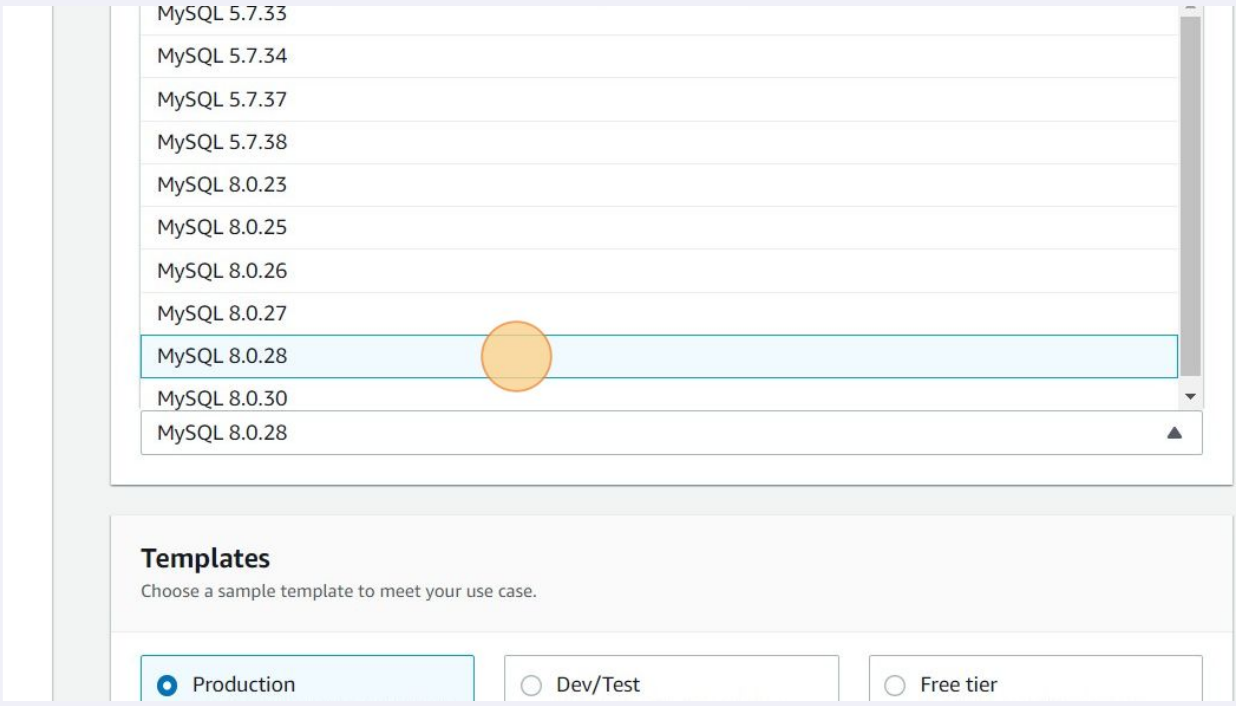
9 Select "MySQL"



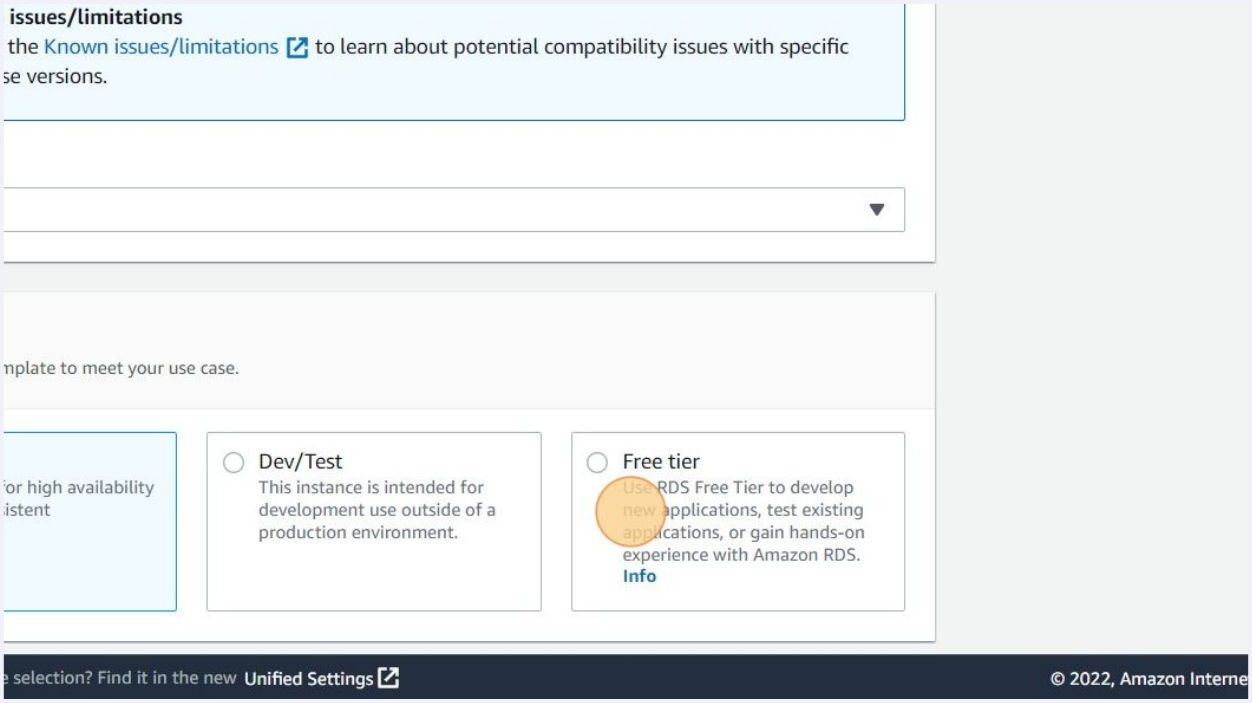
10 Click "MySQL 8.0.28"



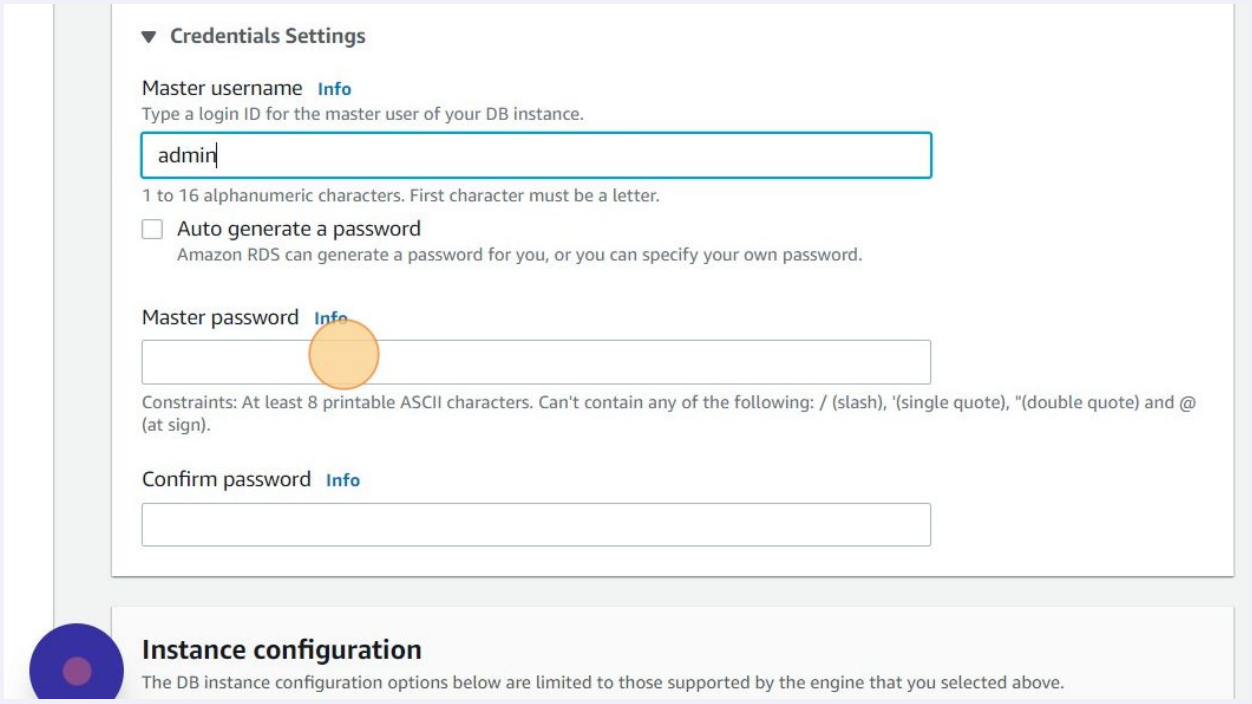
11 Click here.



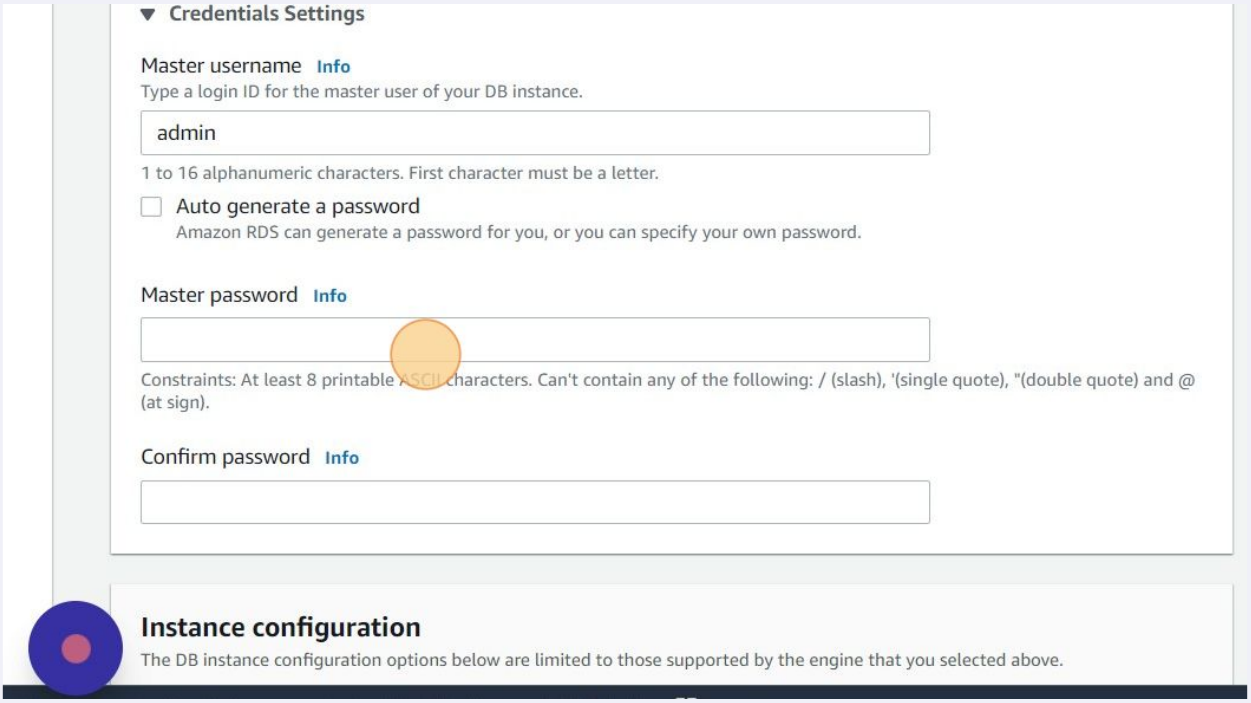
12 Select - Free Tier



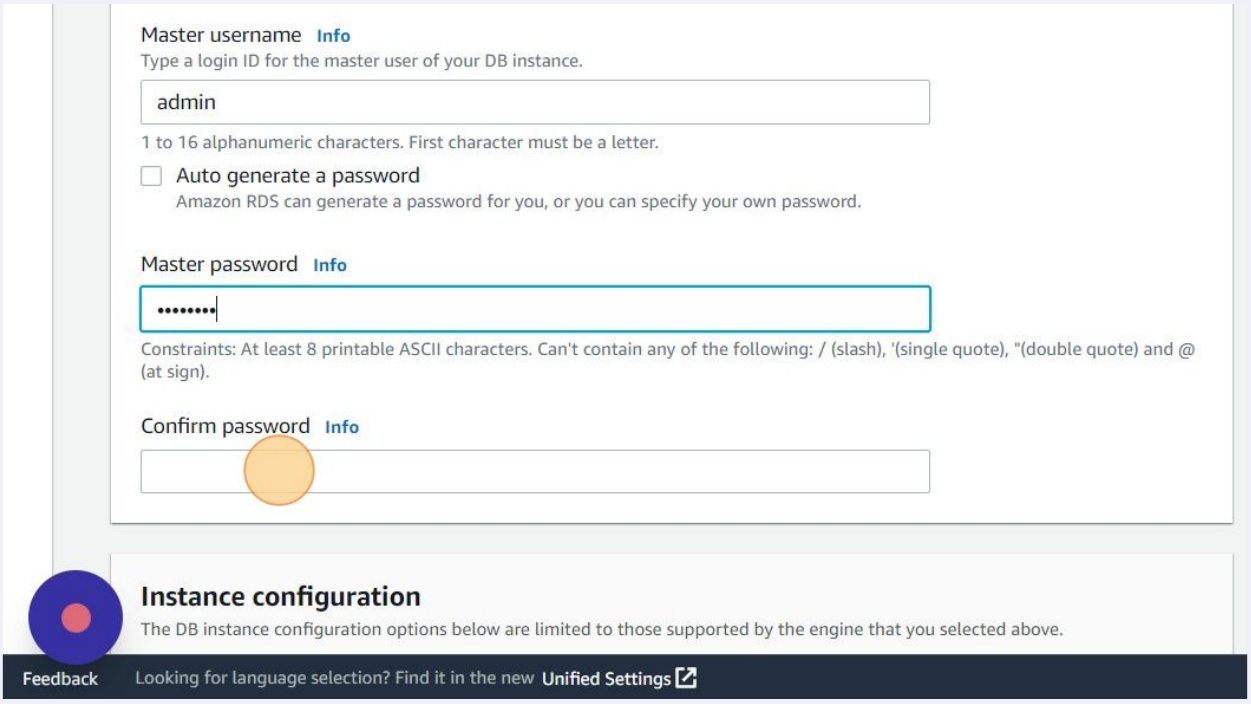
13 Set "admin" as the Master username



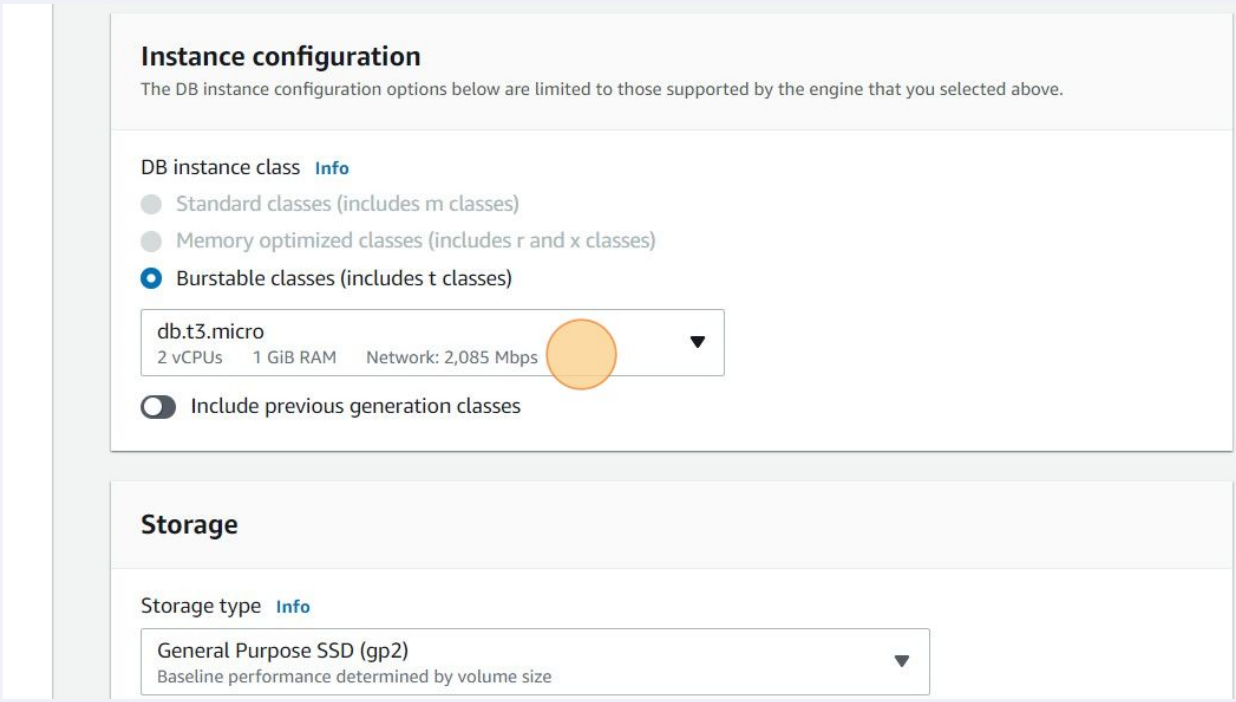
14 Click the "Master password" field.



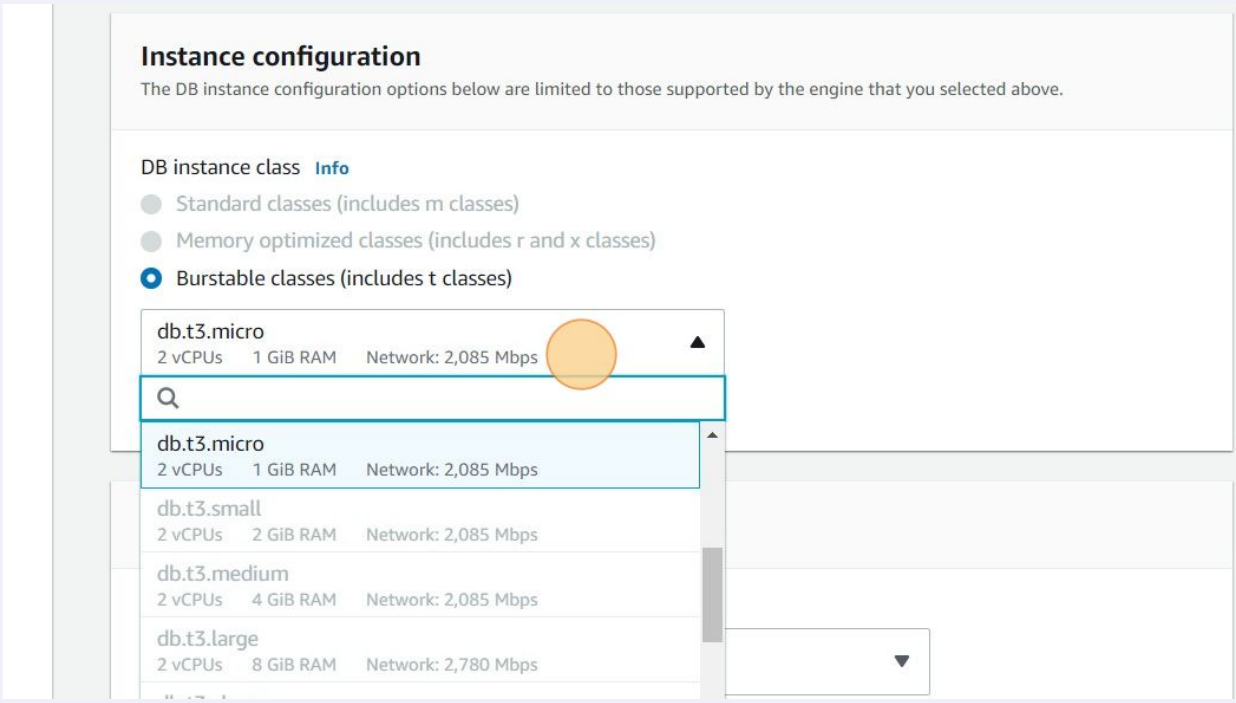
15 Type some password for Master Password and retype the password



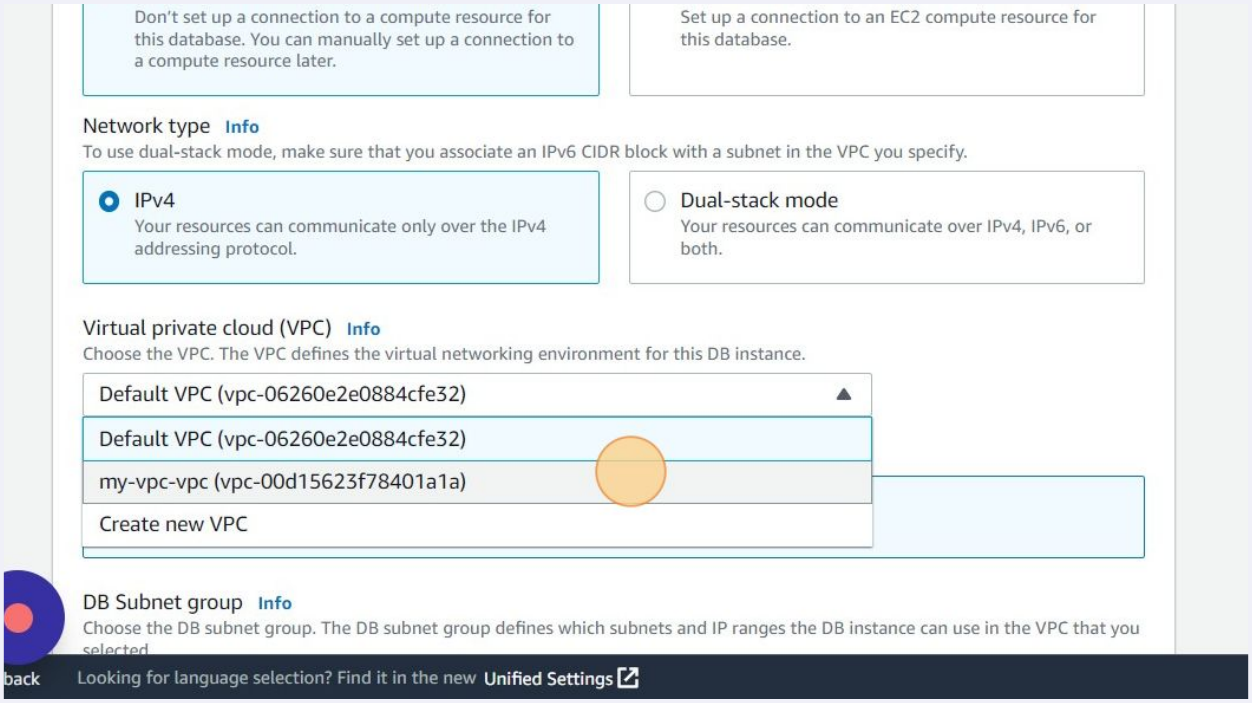
16 Select the DB Instance Class



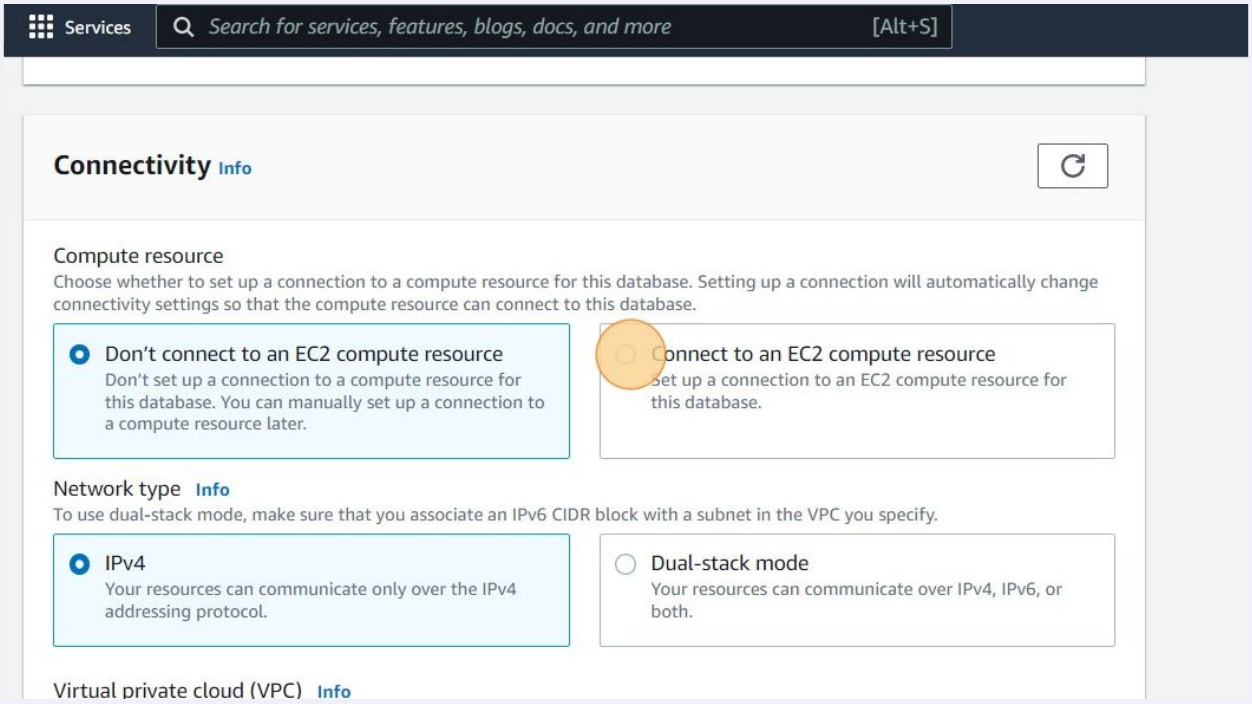
17 Click here.



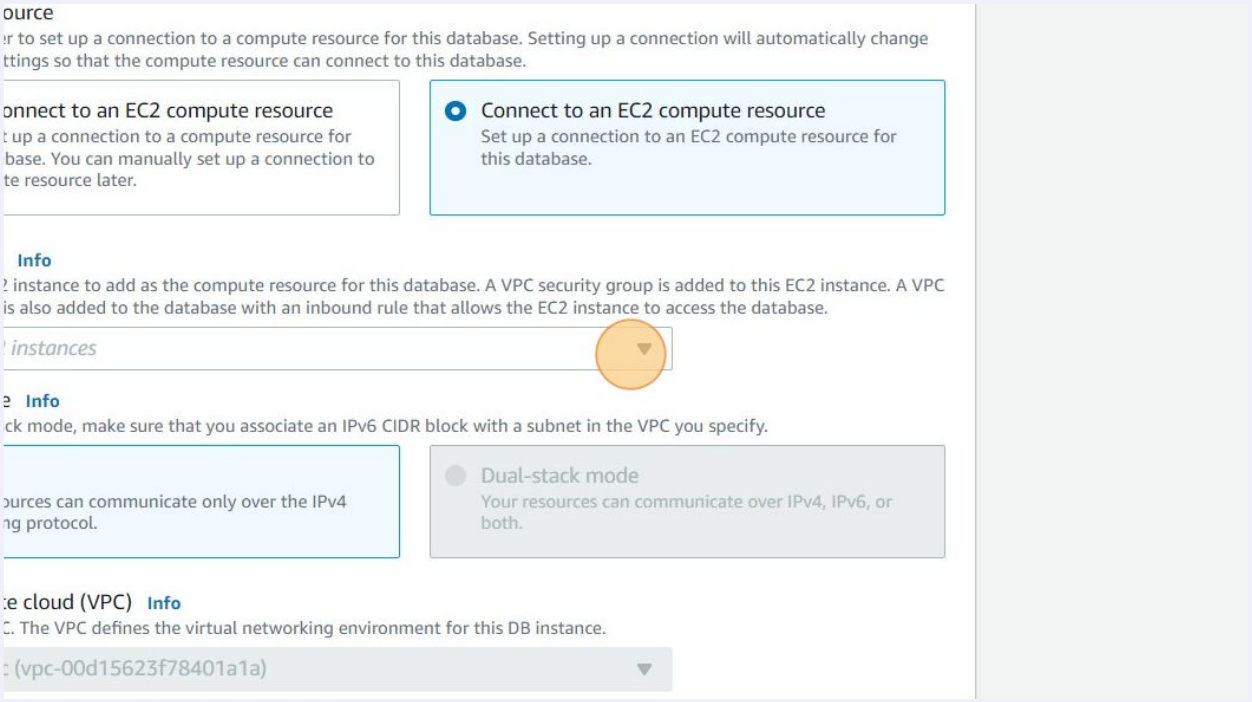
18 Choose the VPC created.



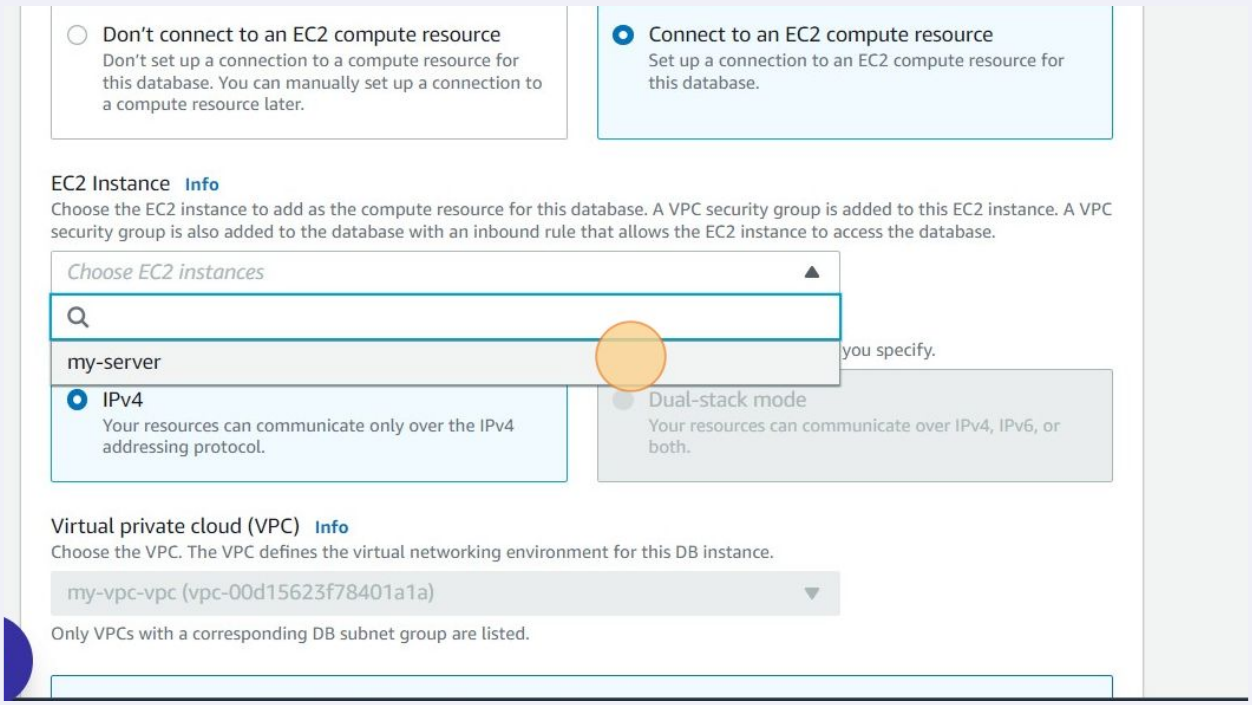
19 Click the "Connect to an EC2 compute resourceSet up a connection to an EC2 compute resource for this database." field.



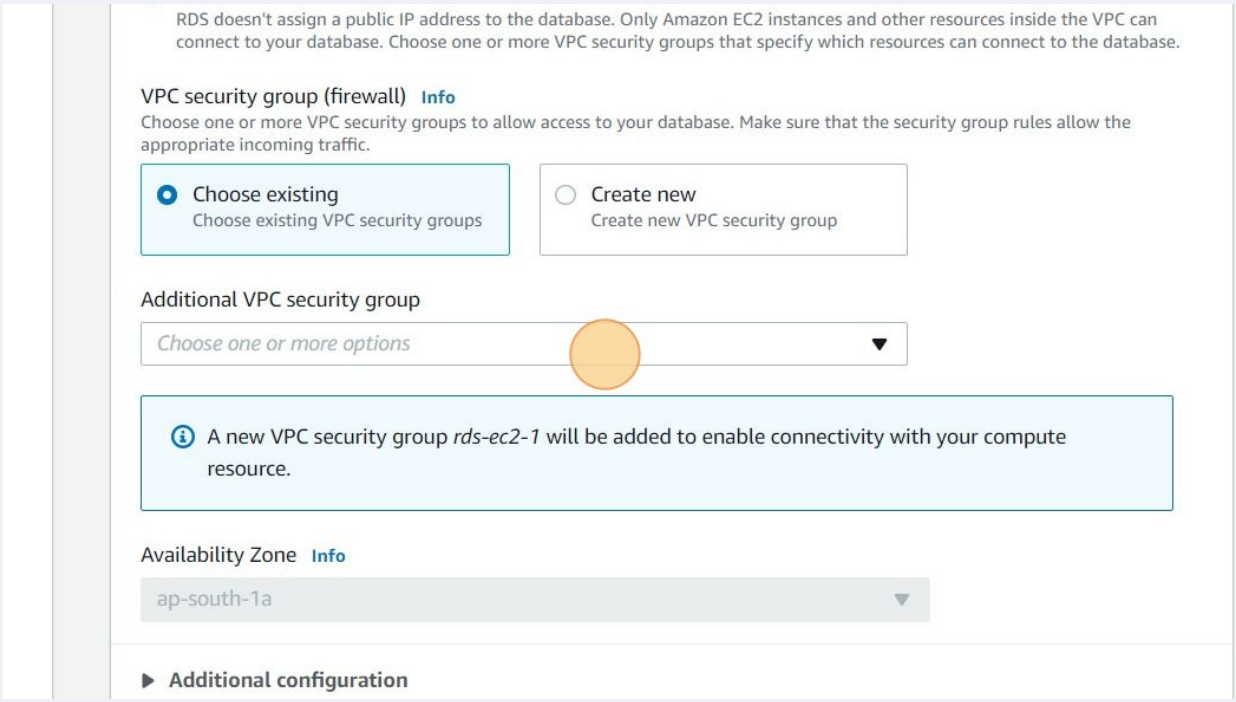
20 Click the "Connect to an EC2 compute resource " field.



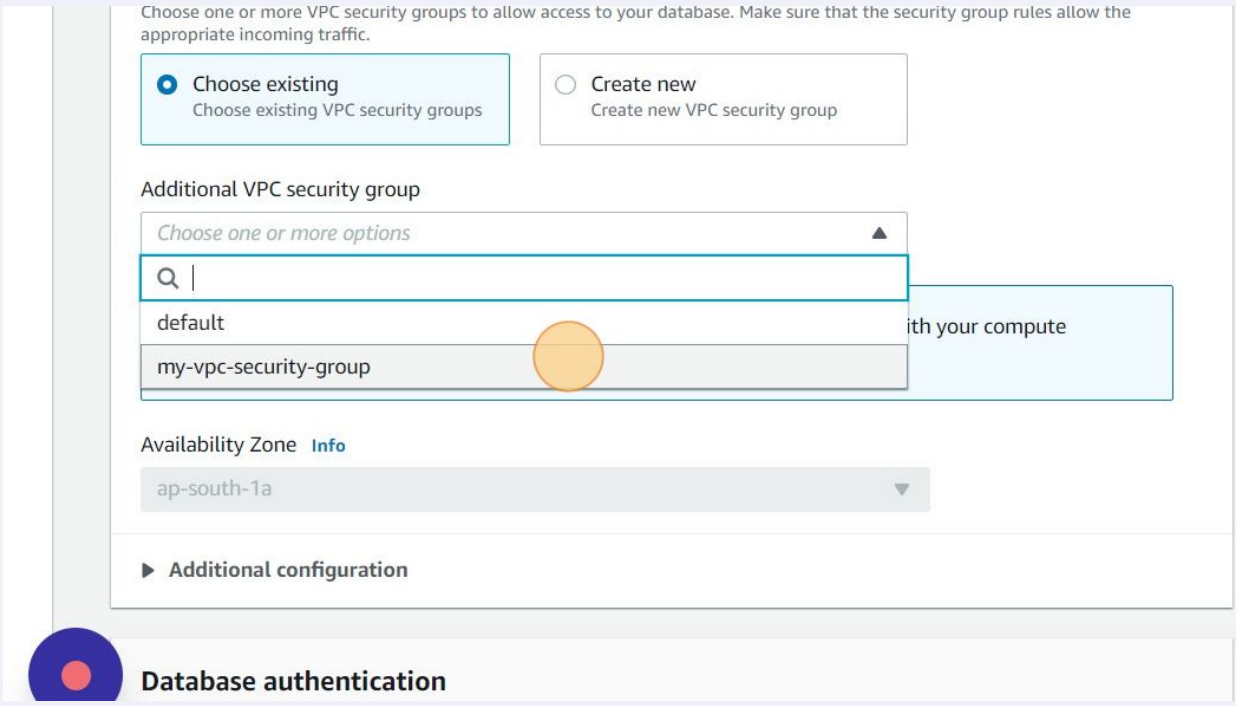
21 In the EC2 Instance drop down, select the server you created before.



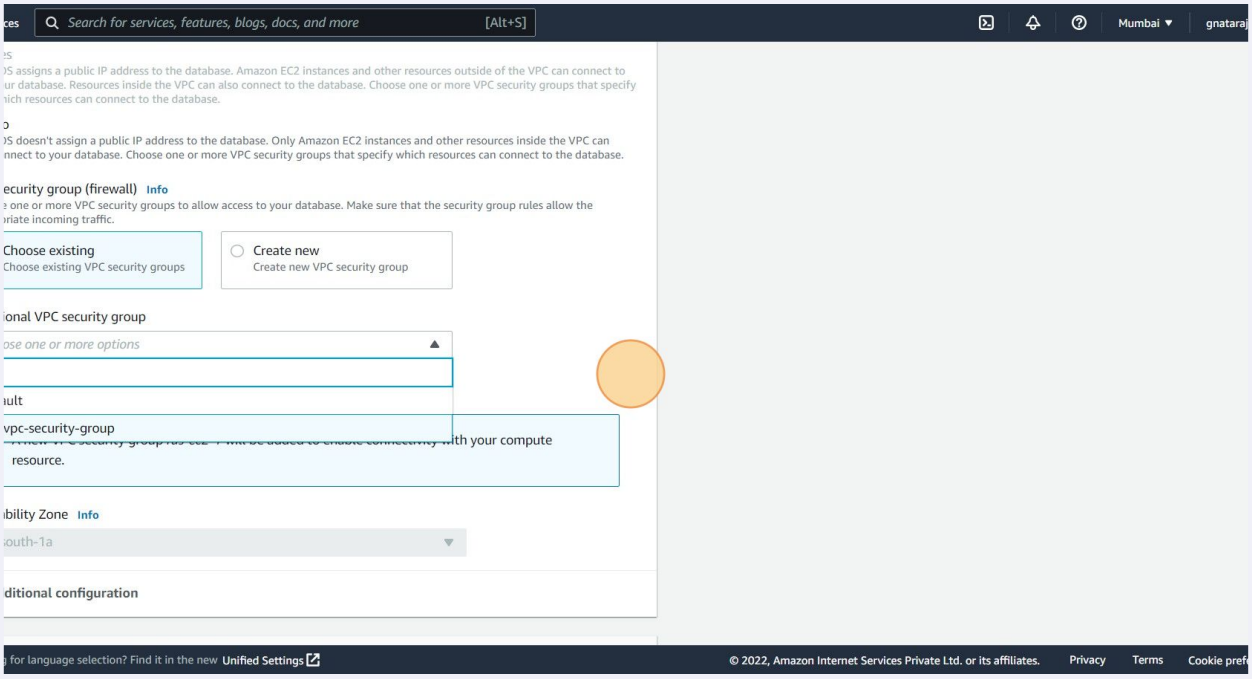
22 In the VPC Security group, select the "Choose Existing"



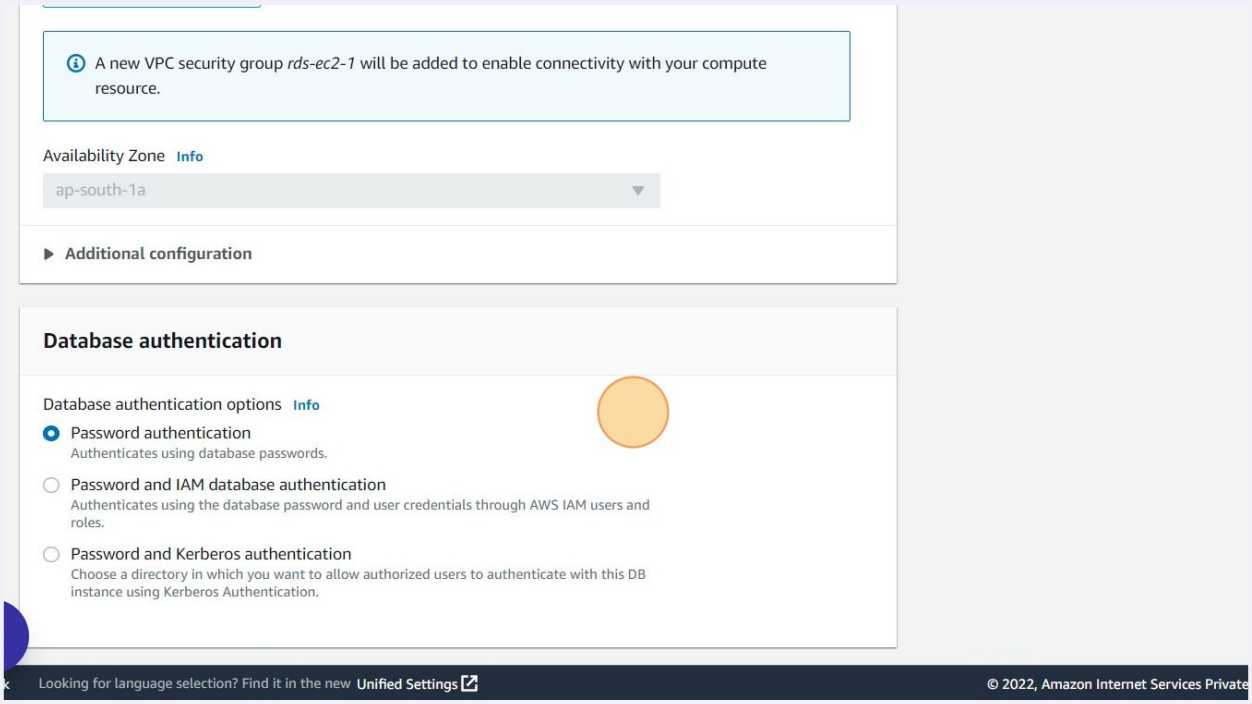
23 Select the previously created VPC Security Group



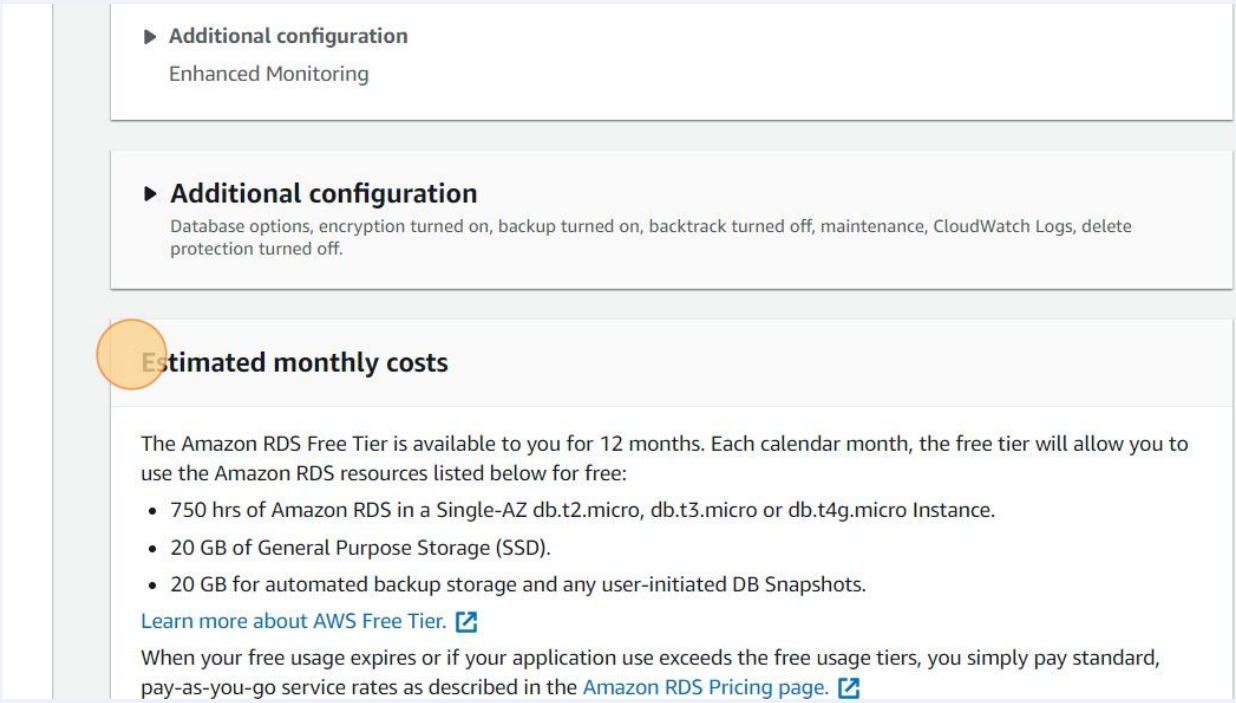
24 Click here.



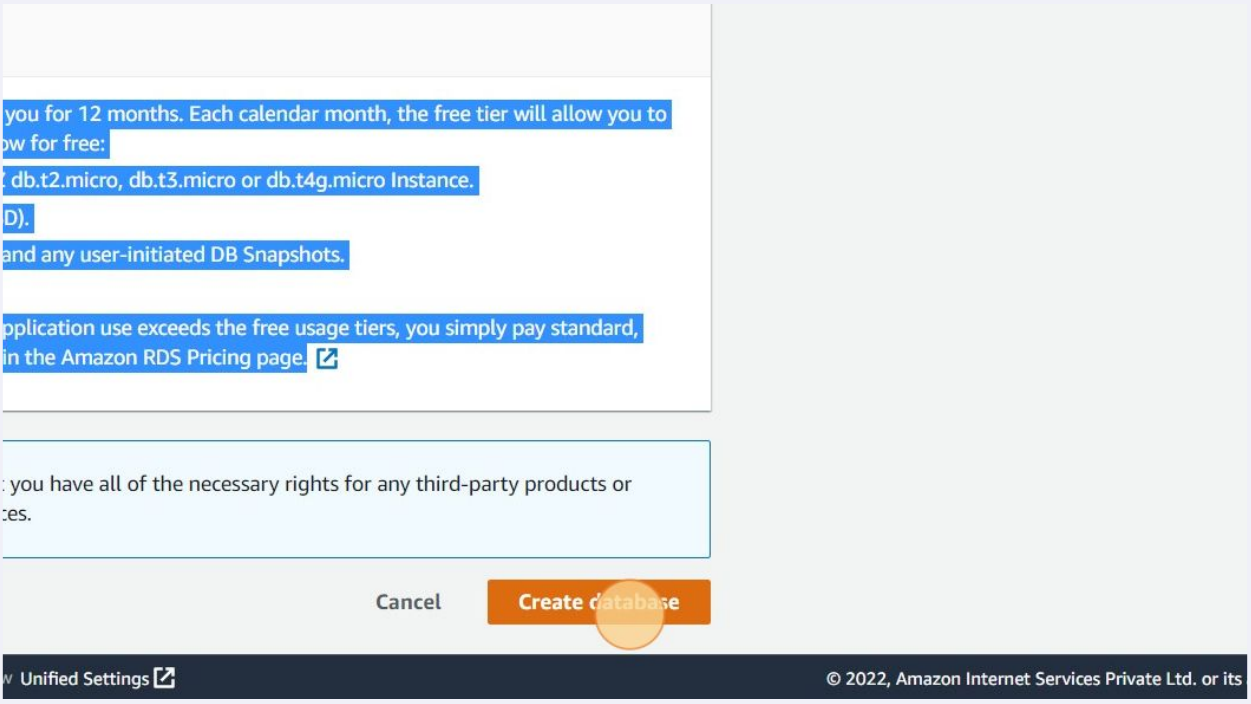
25 For Database Authentication, choose Password authentication.



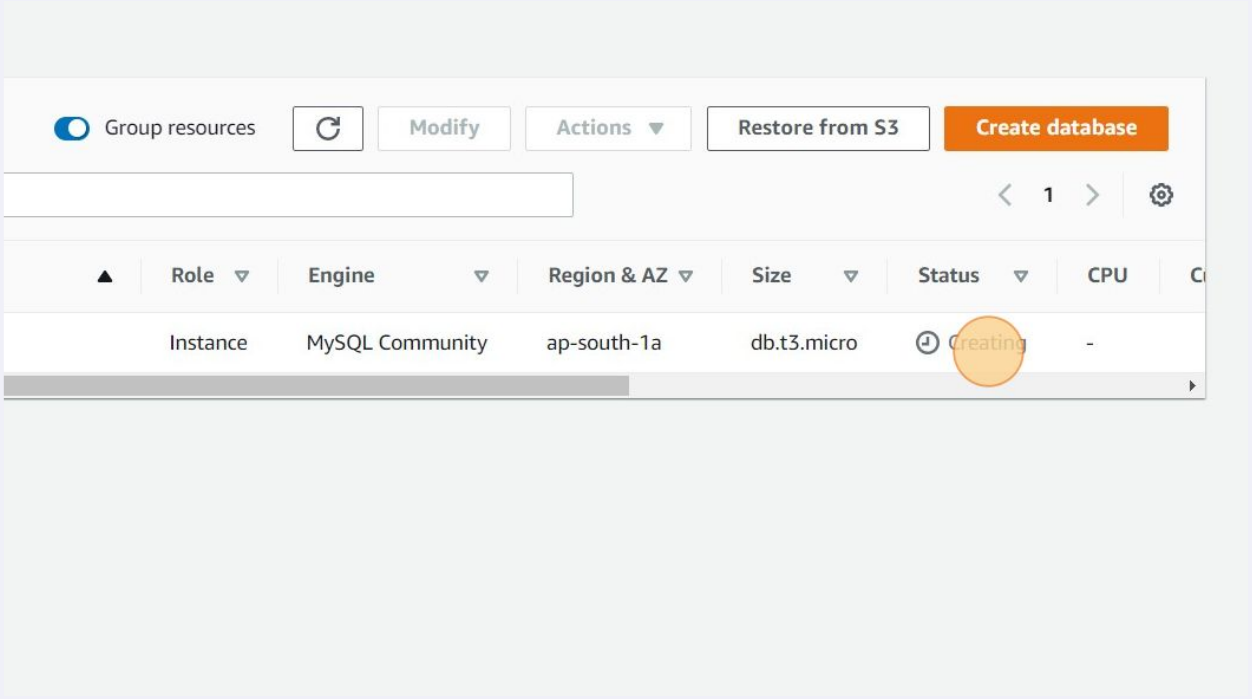
26 Review the Estimated monthly costs of the DB Instance



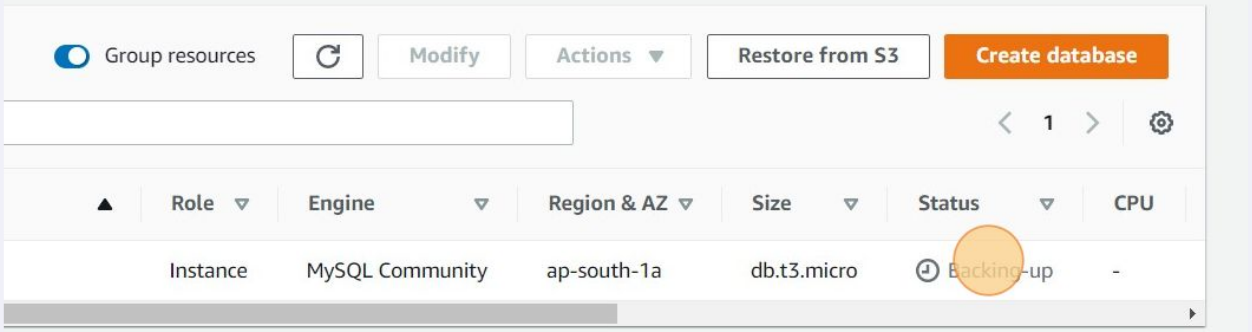
27 Click "Create database"



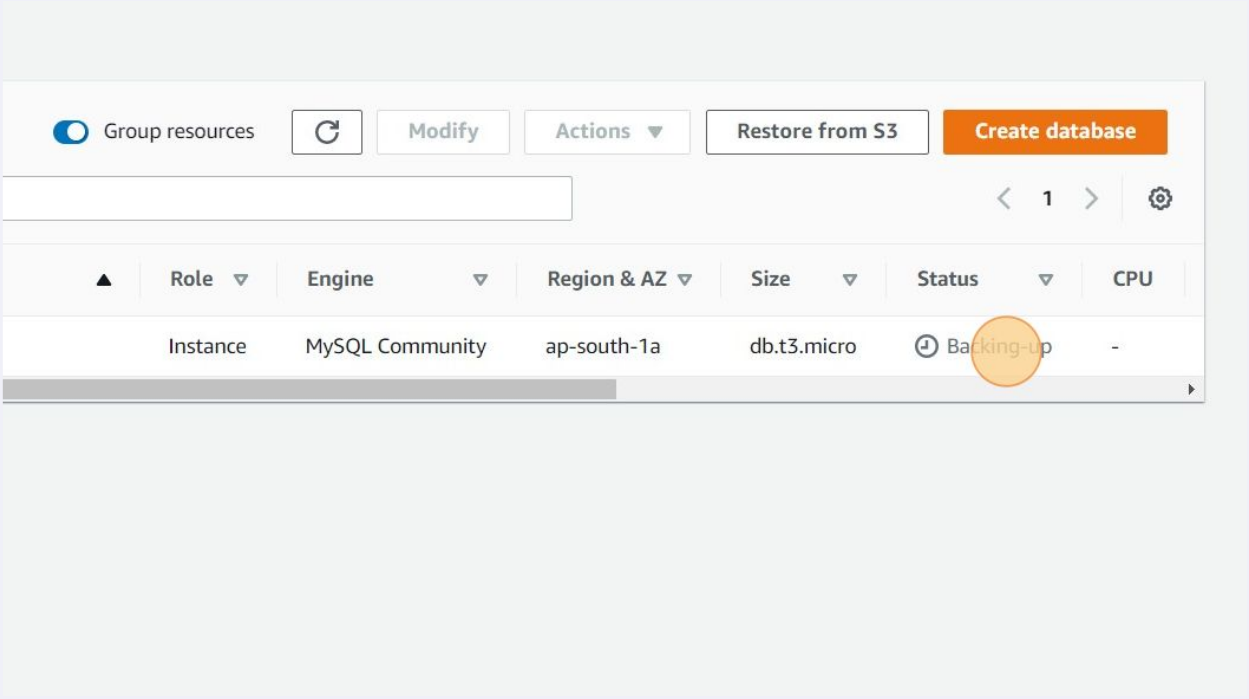
28 Double-click "Creating"



29 Click "Backing-up"

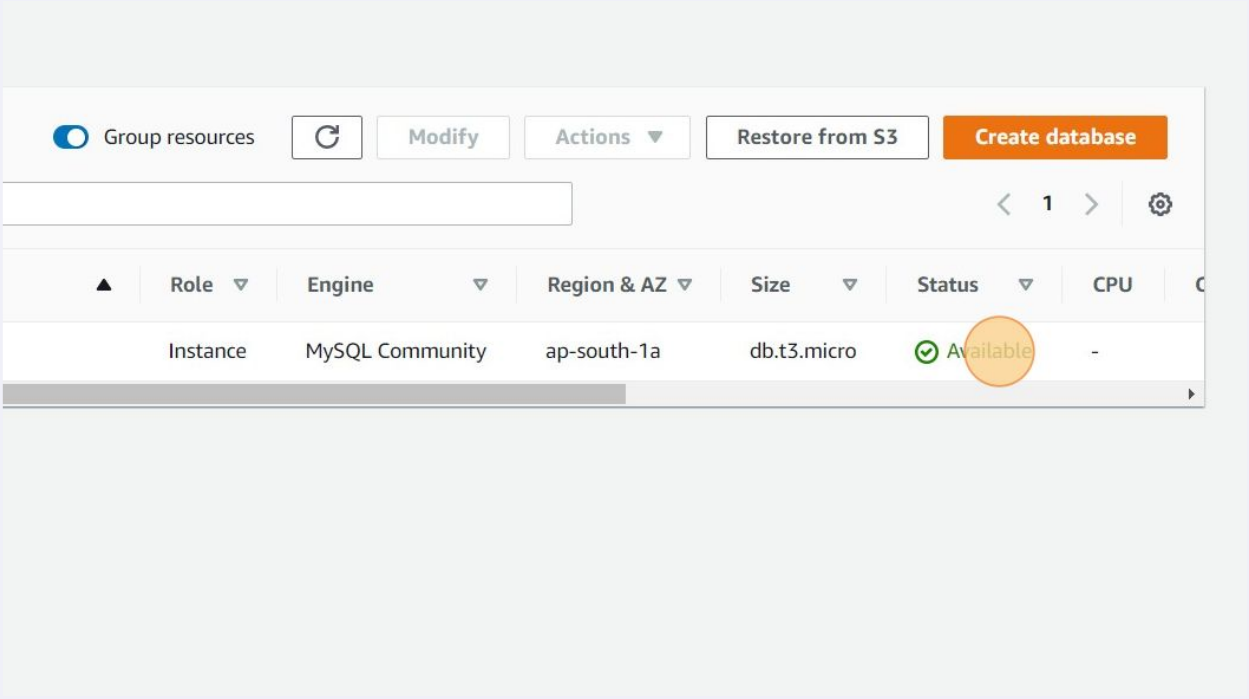


30 Double-click "Backing-up"

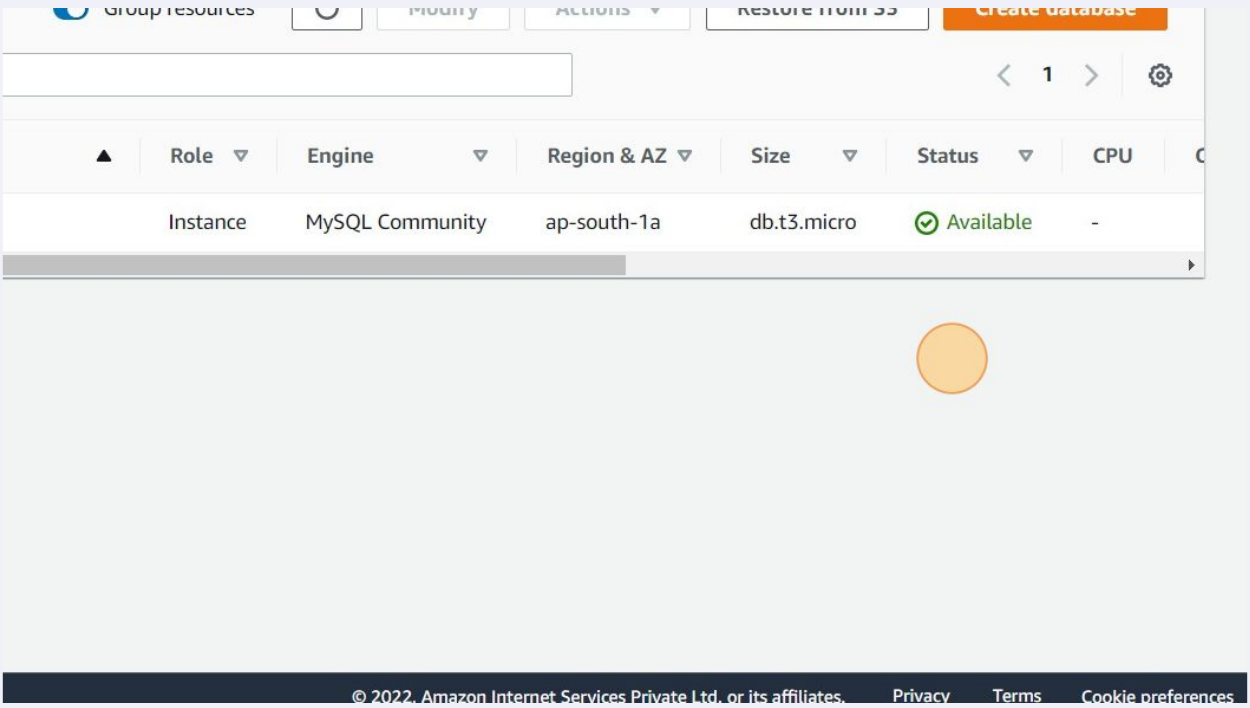


31 Click "Backing-up"

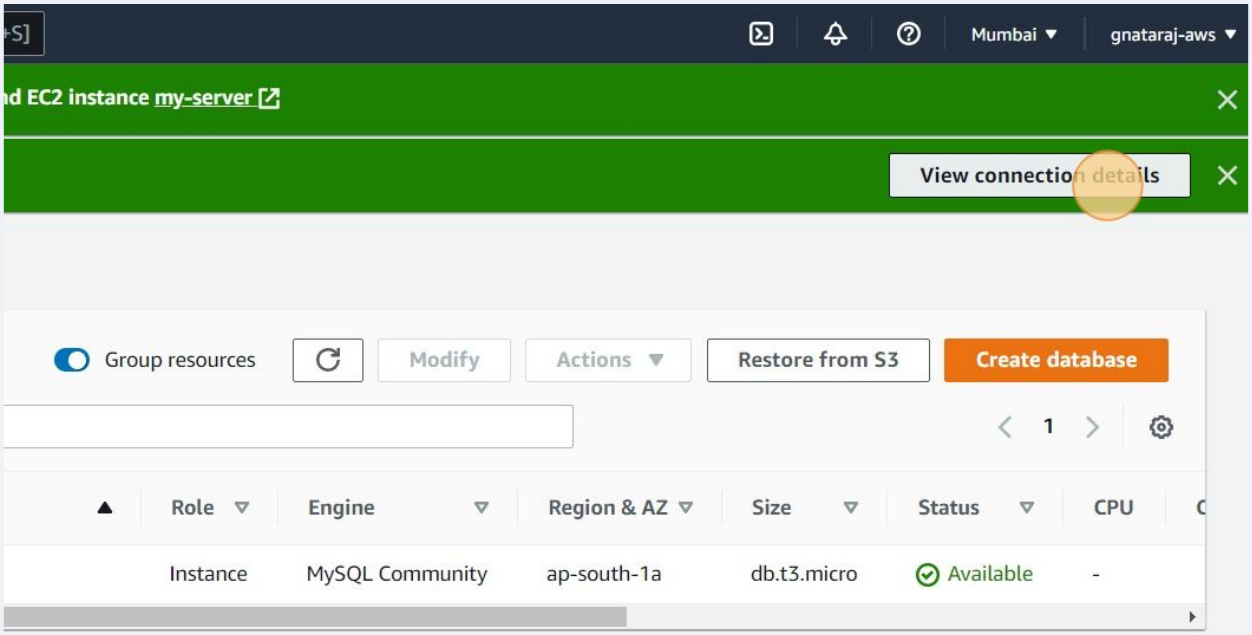
32 Double-click "Available"



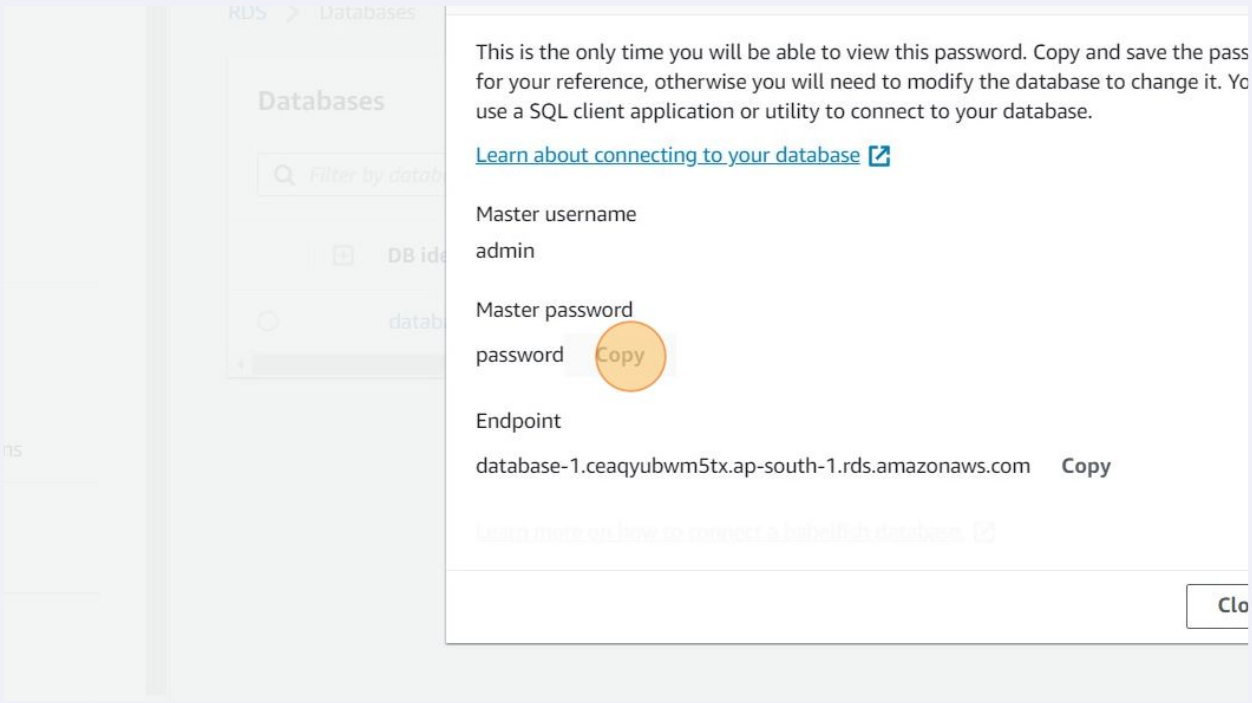
33 Click here.



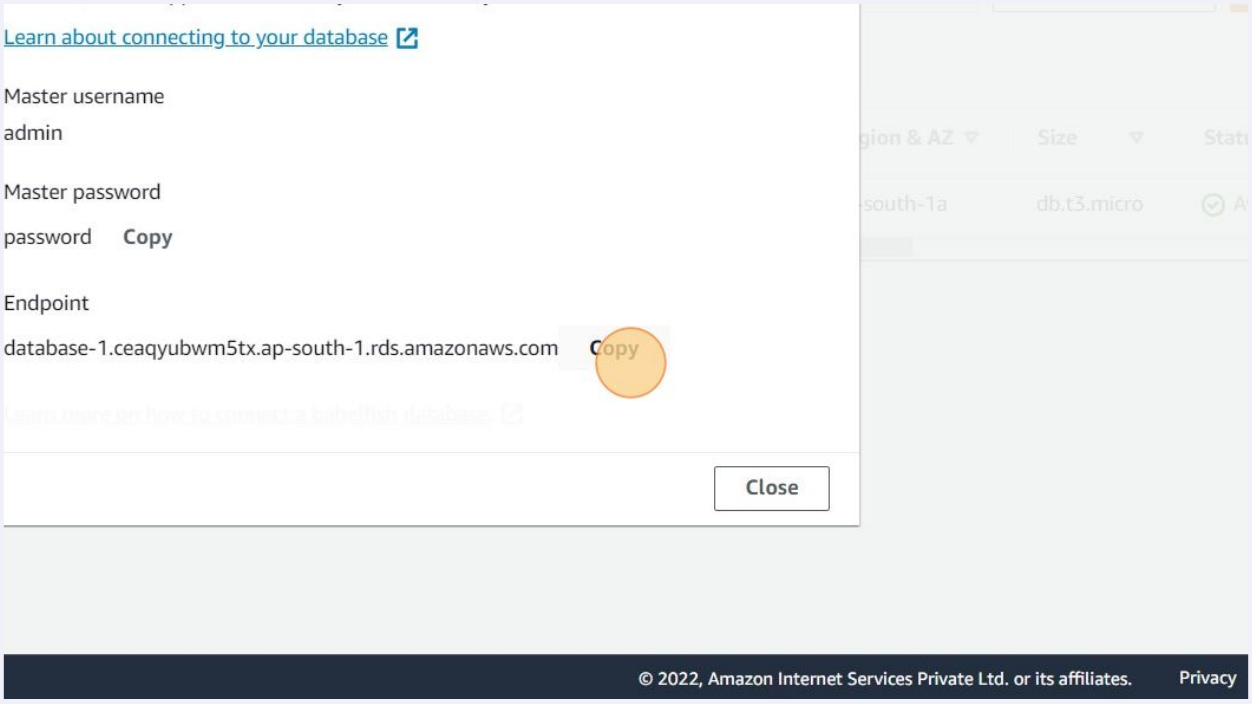
34 Click "View connection details"



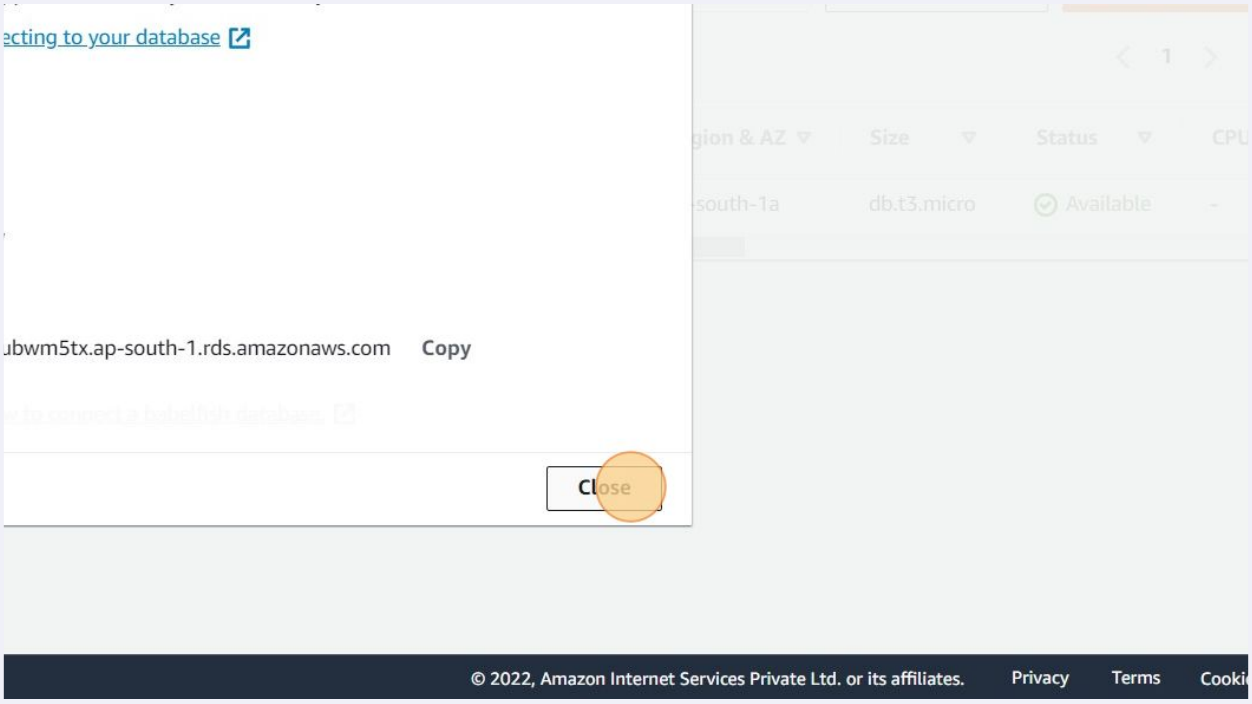
35 Click "Copy"



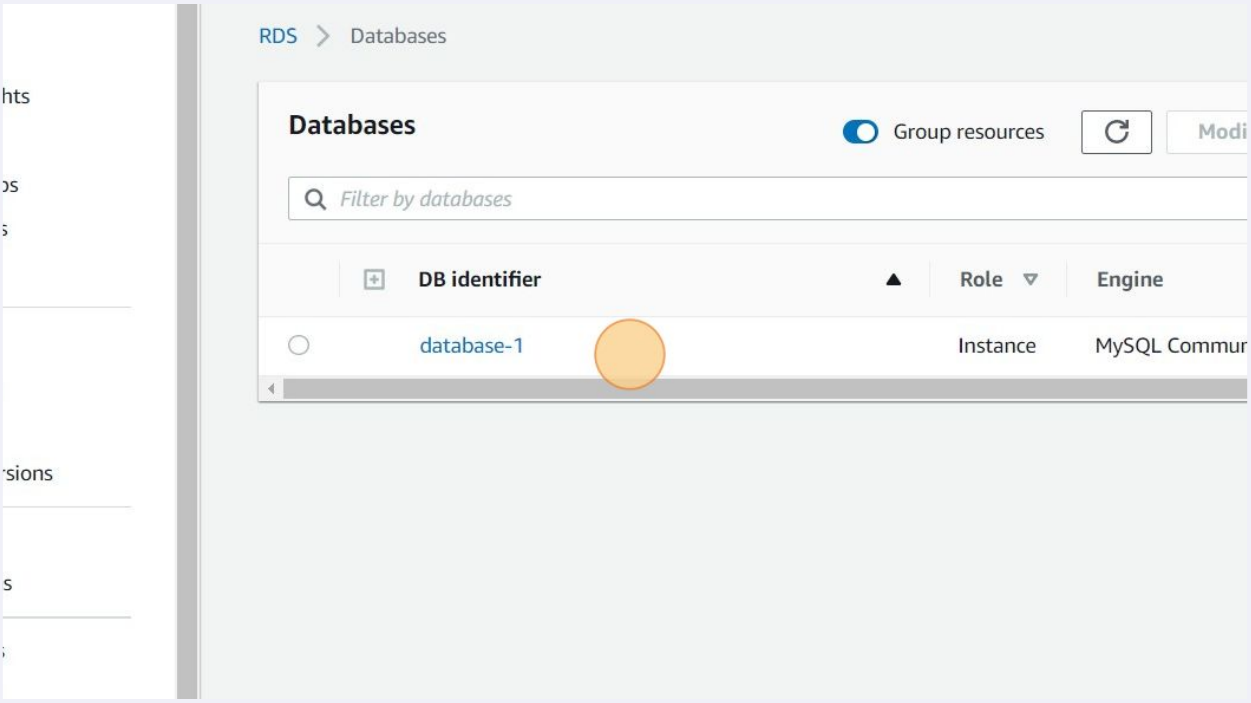
36 Click "Copy"



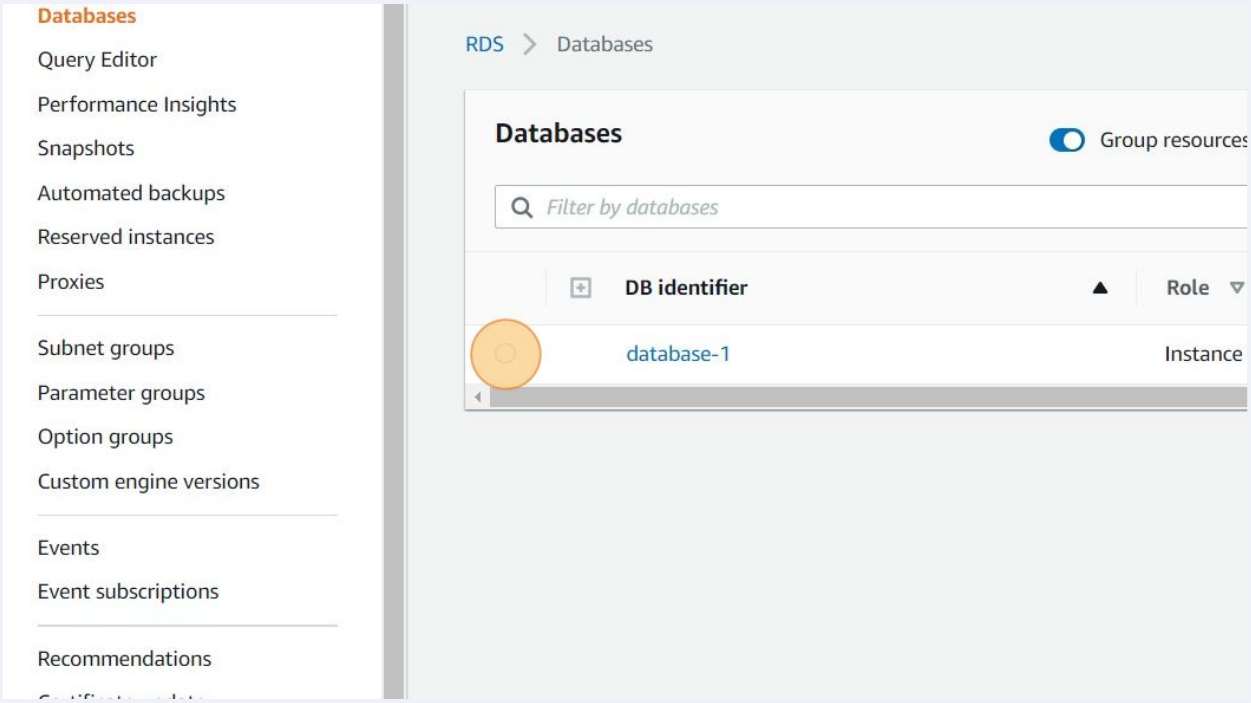
37 Click "Close"



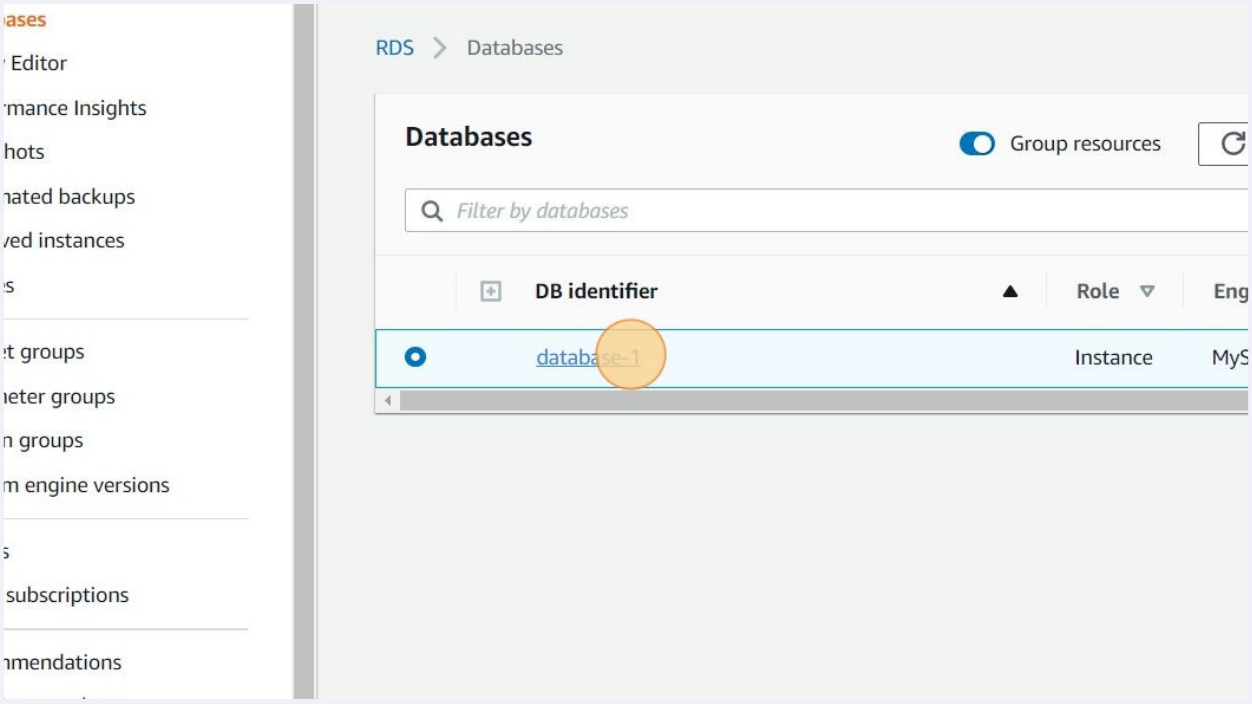
38 Click here.



39 Click this radio button.

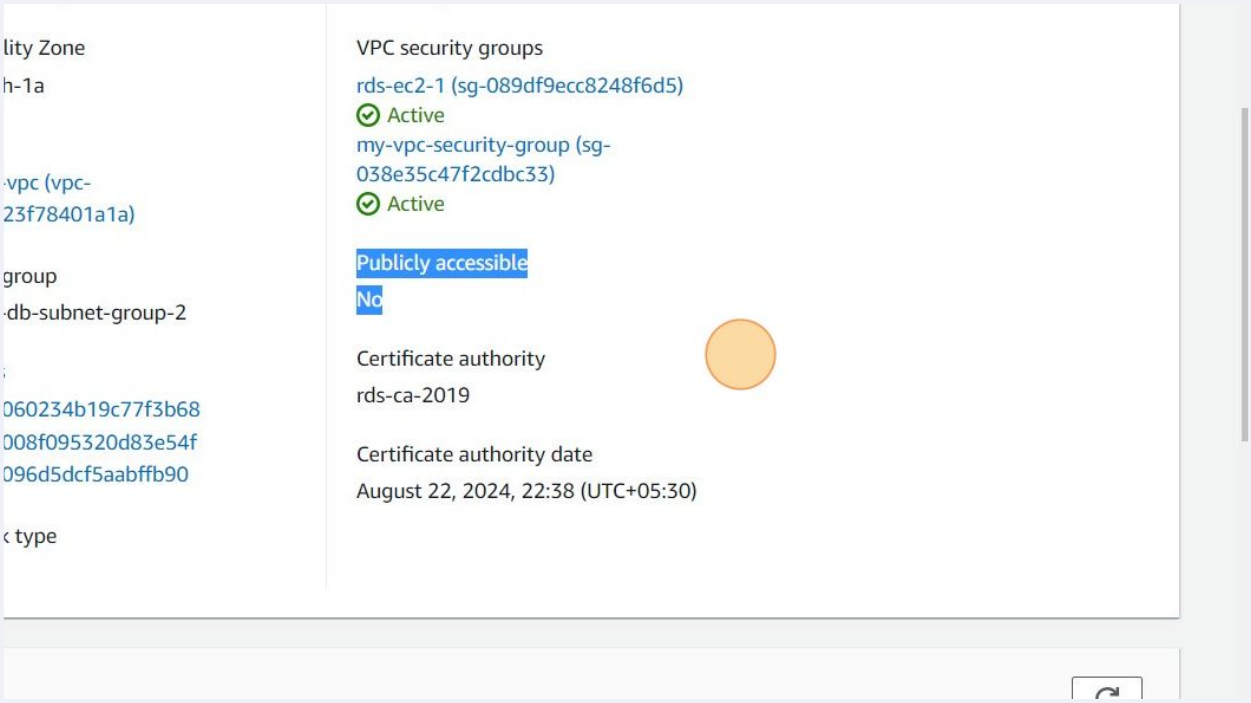


40 Click "database-1"



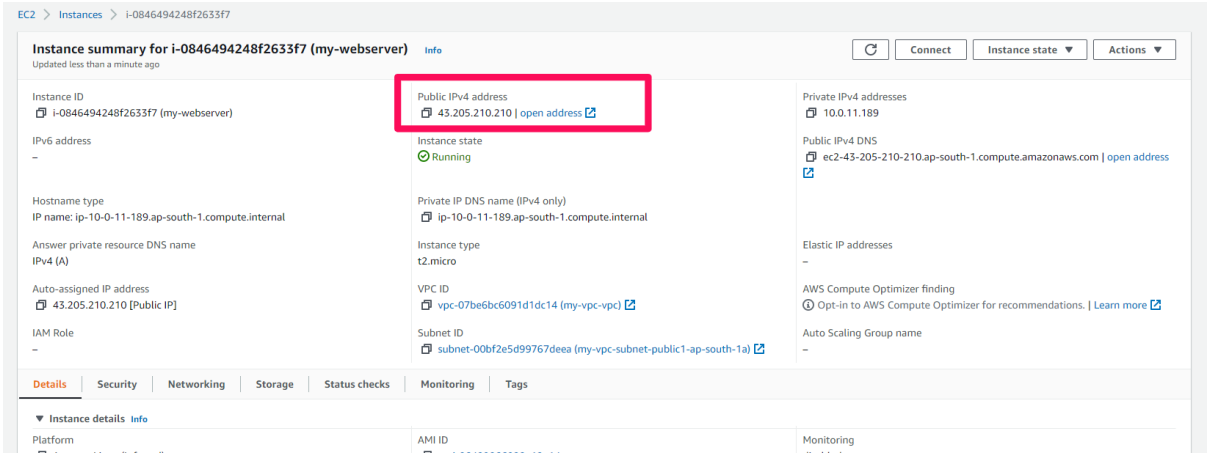
41 Click "No"

42 Click here.

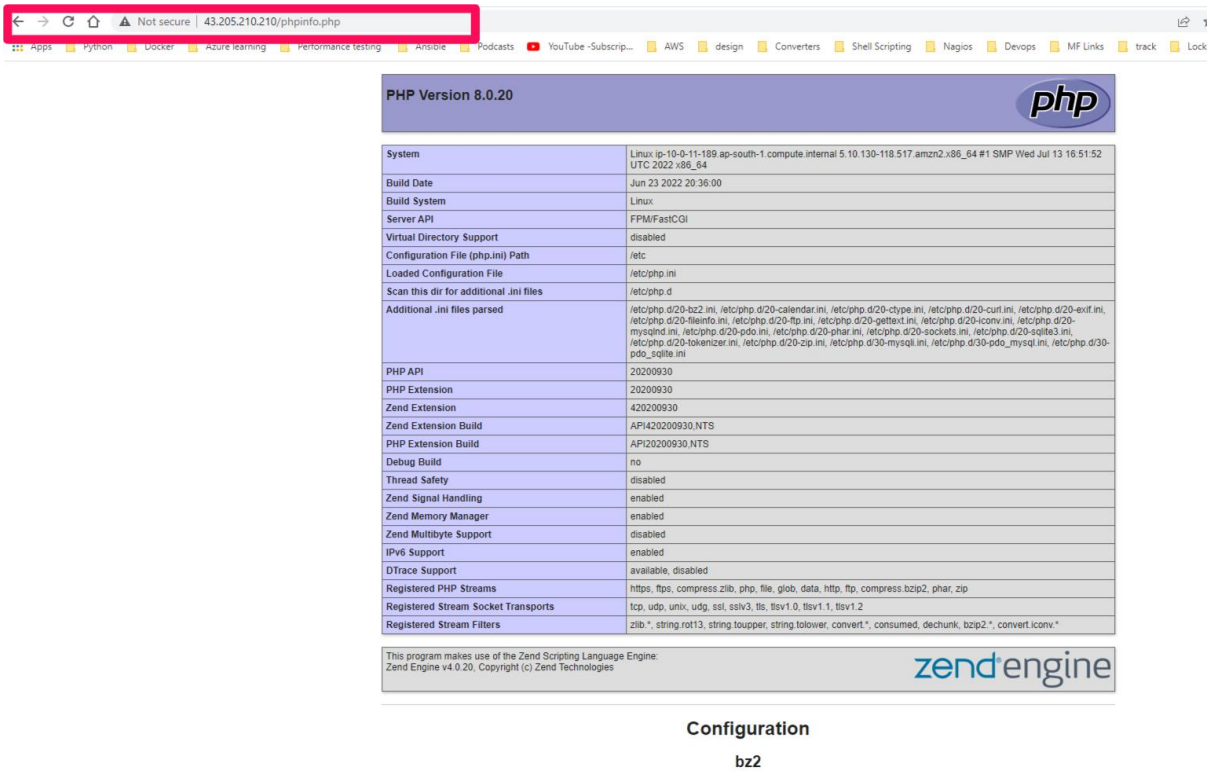


6. Setting up the DB connections and testing the configuration

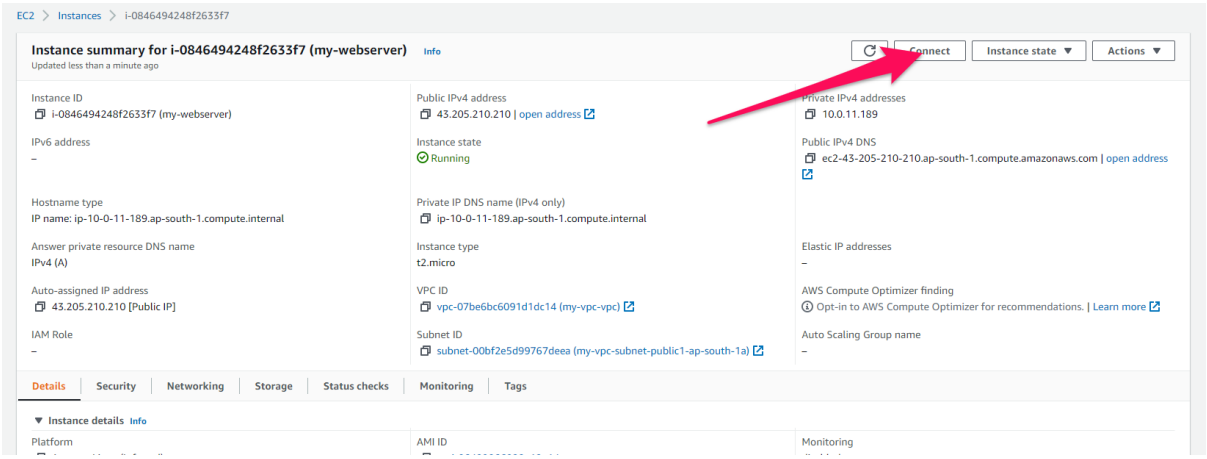
1. Access the EC2 Instance you created and copy the Public IPv4 Address



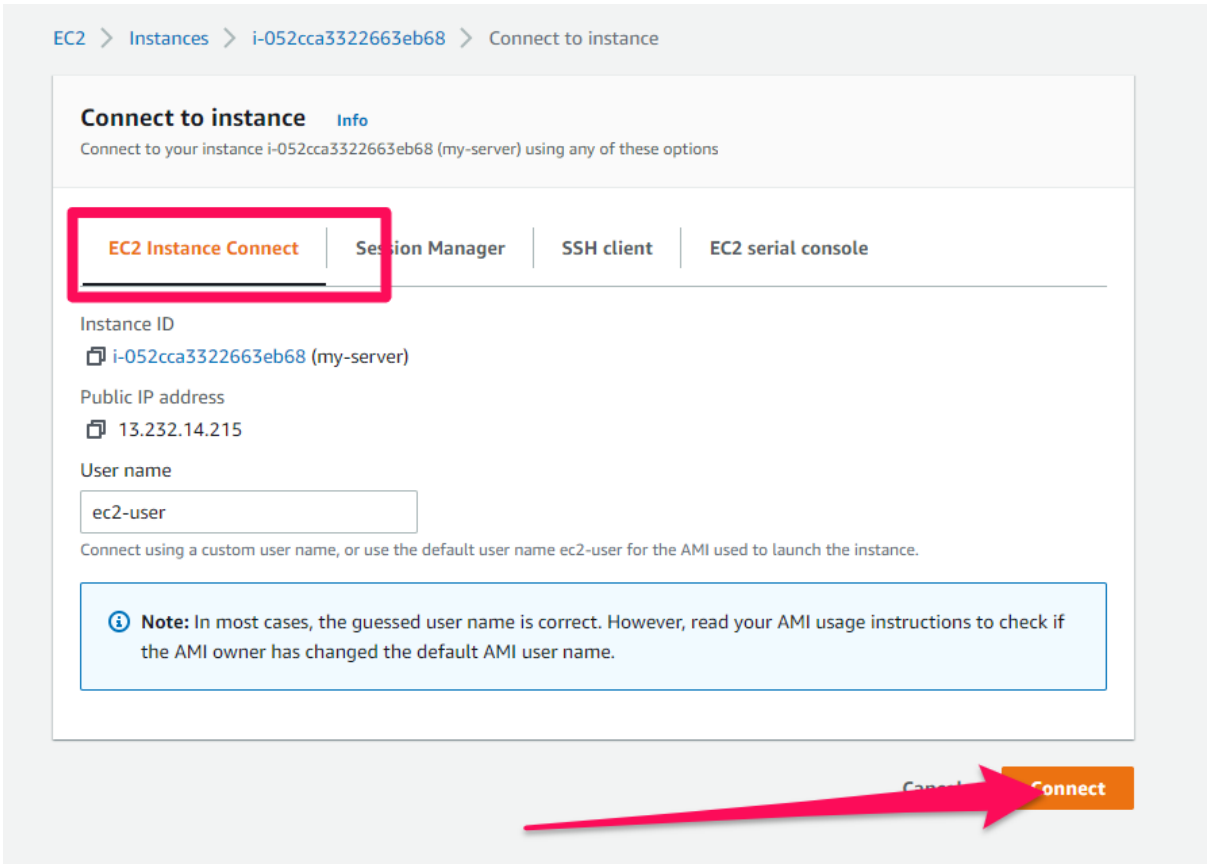
2. Access the phpinfo.php page from the browser with the Public IPv4 address



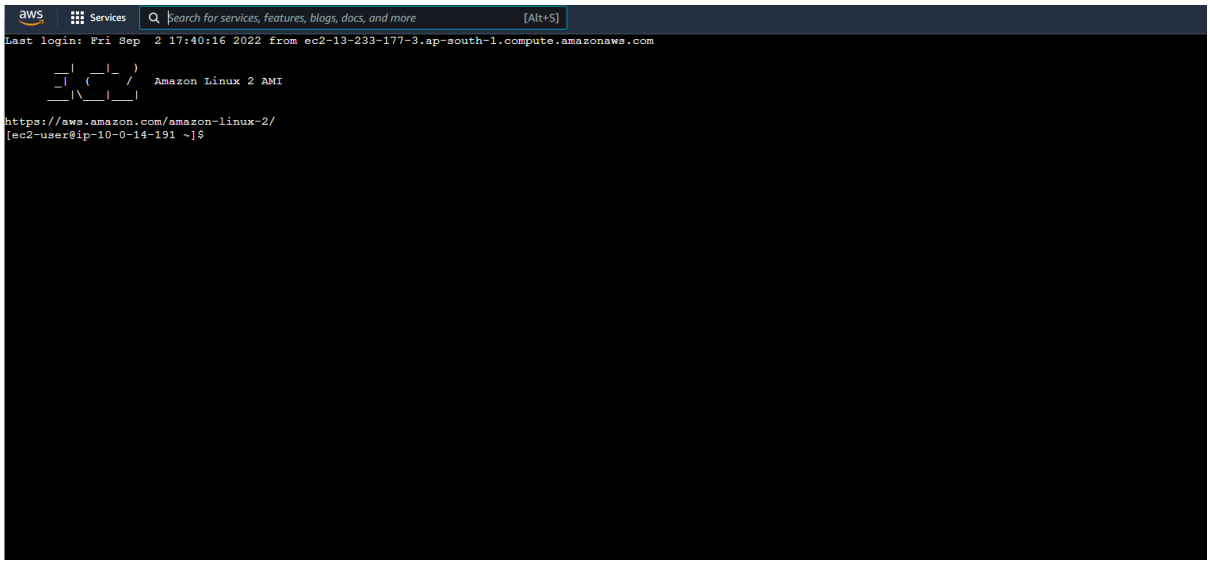
3. Connect to the EC2 instance:



4.



5.



6. Connect to the SQL server and create the database and table:

   [ec2-user@ip-10-0-14-191 ~]$ **mysql -u admin -p'password' -h database-1.ceaqyubwm5tx.ap-south-1.rds.amazonaws.com**

   Welcome to the MariaDB monitor. Commands end with ; or \g.
   Your MySQL connection id is 50
   Server version: 8.0.28 Source distribution

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

   Type 'help;' or '\h' for help. Type '\c'

7. Create the database
   MySQL [(none)]> **create database employees_db;**
   Query OK, 1 row affected (0.005 sec)

8. Check the database
   MySQL [(none)]> **show databases;**
   +--------------------+
   | *Database* |
   +--------------------+
   | *employees_db* |
   | *information_schema* |
   | *mysql* |
   | *performance_schema* |

| *sys* |
+--------------------+
5 rows in set (0.001 sec)

9. Change to the database:
   MySQL [(none)]> **use employees_db;**
   Database changed
   MySQL [employees_db]>

10. Create table:
    MySQL [employees_db]> **CREATE TABLE employees (**
    -> id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    -> name VARCHAR(100) NOT NULL,
    -> address VARCHAR(255) NOT NULL,
    -> salary INT(10) NOT NULL
    ->);
    Query OK, 0 rows affected, 1 warning (0.033 sec)

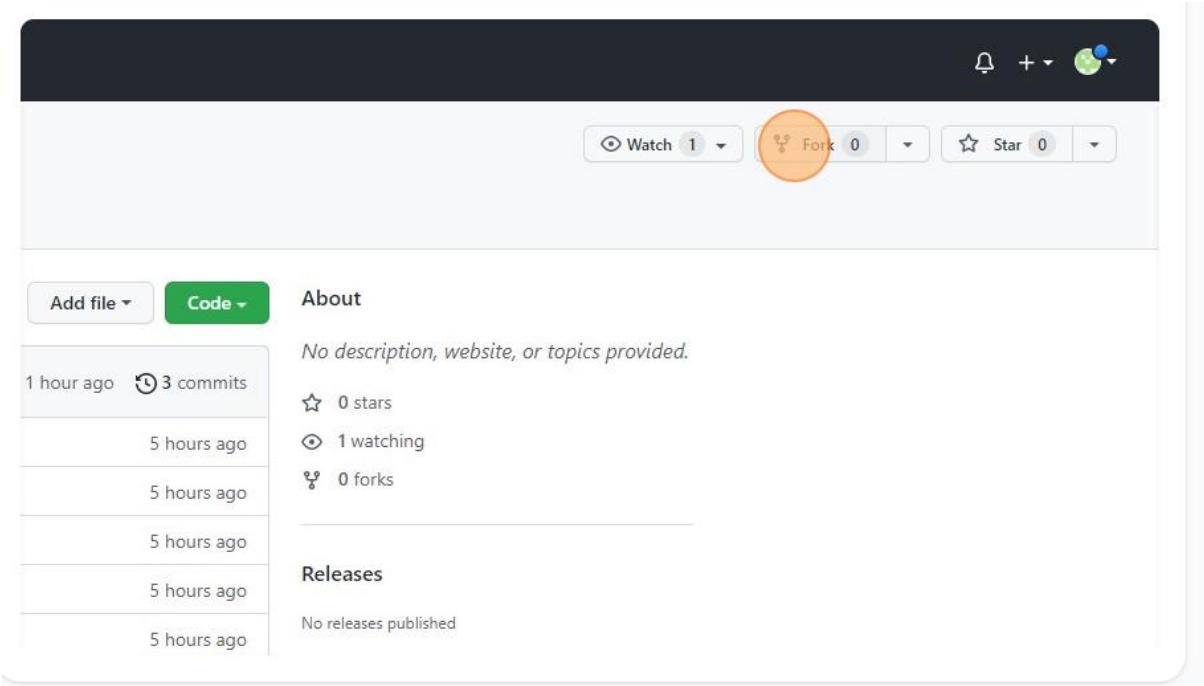
    For copy and paste, use the below:

    CREATE TABLE employees (
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100) NOT NULL,
    address VARCHAR(255) NOT NULL,
    salary INT(10) NOT NULL
    );

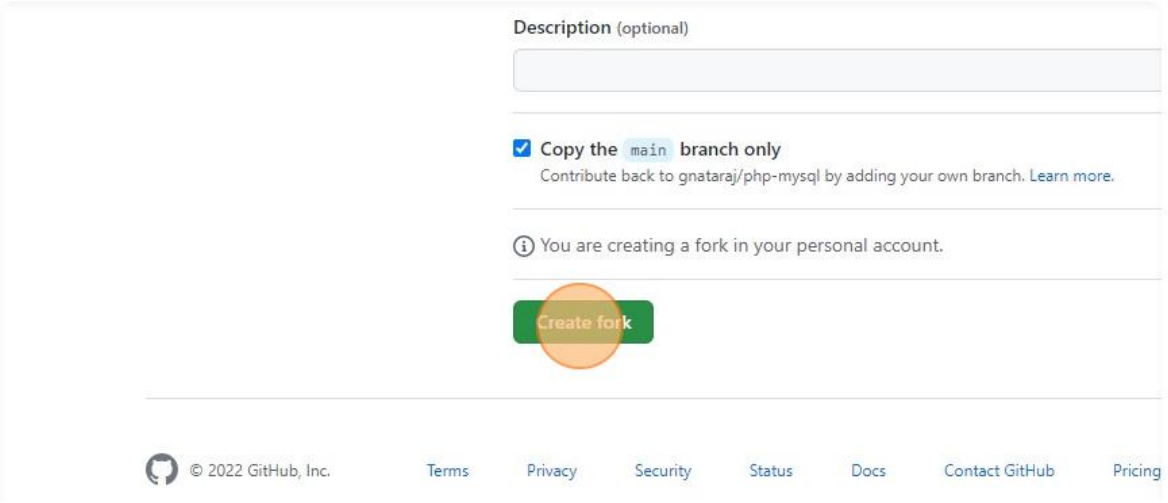
11. Check the tables:
    MySQL [employees_db]> **show tables;**
    +----------------------+
    | *Tables_in_employees_db* |
    +----------------------+
    | *employees* |
    +----------------------+
    1 row in set (0.001 sec)

11.a -> Exit the mysql prompt by typing exit command.

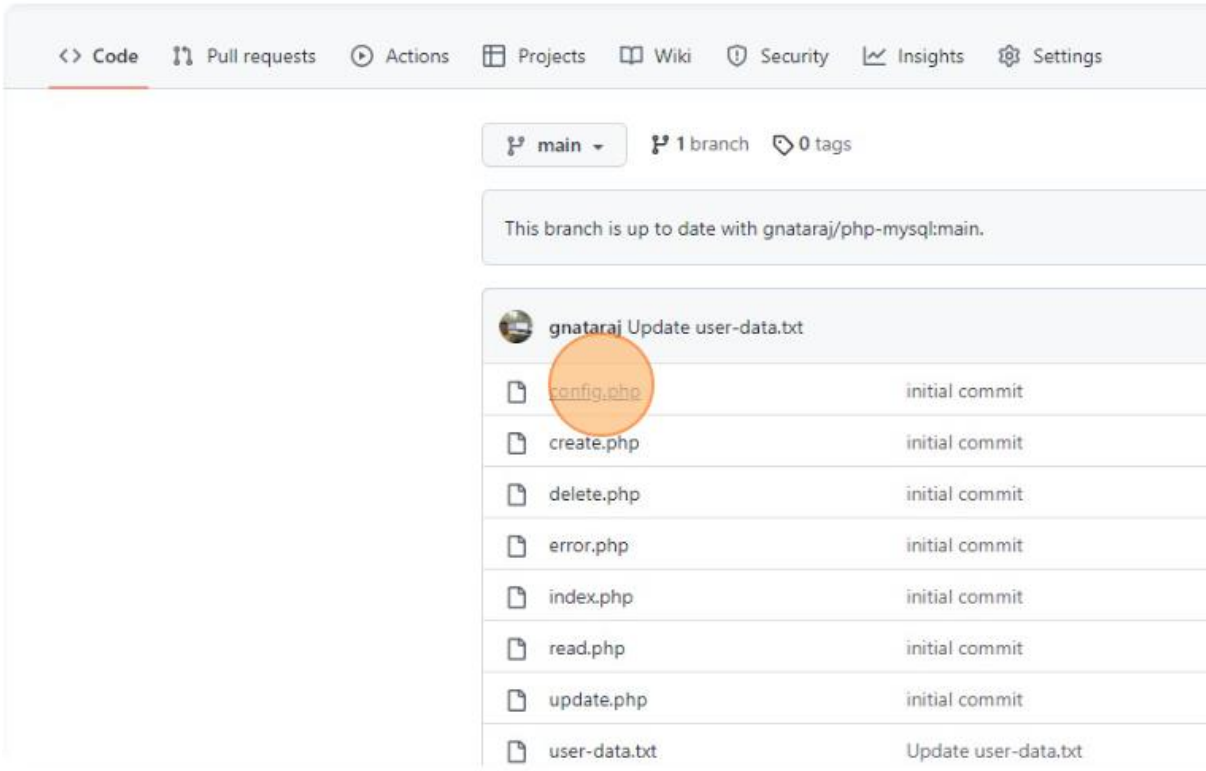
12. Fork the Github repo to your Github account and update the dB detail:
    a. Navigate to **github.com/gnataraj/php-mysql**
    b. Click on Fork



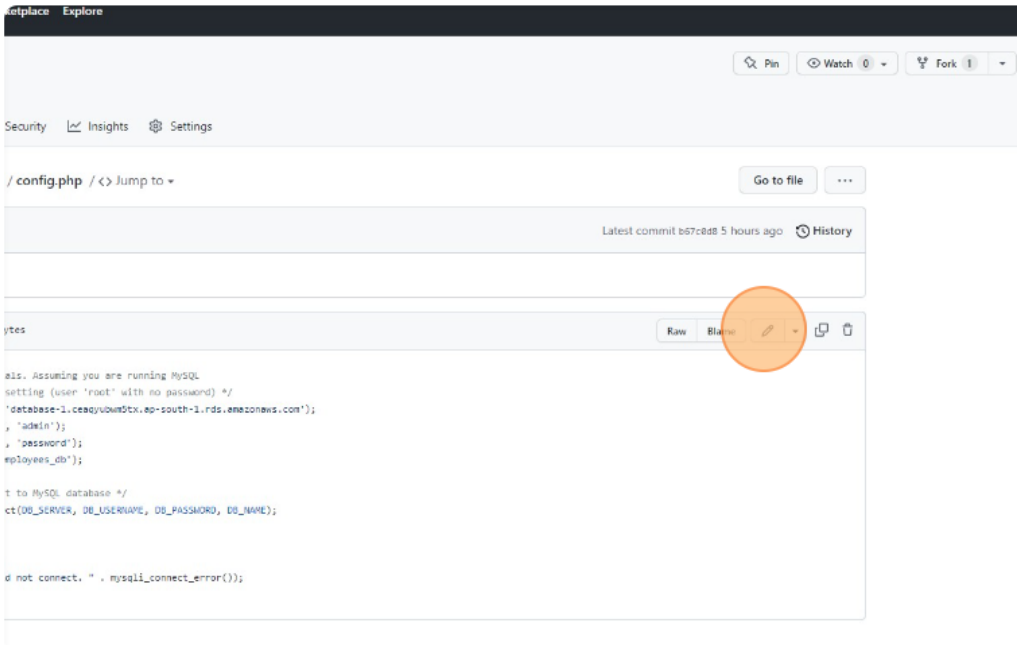
c. Click on Create fork



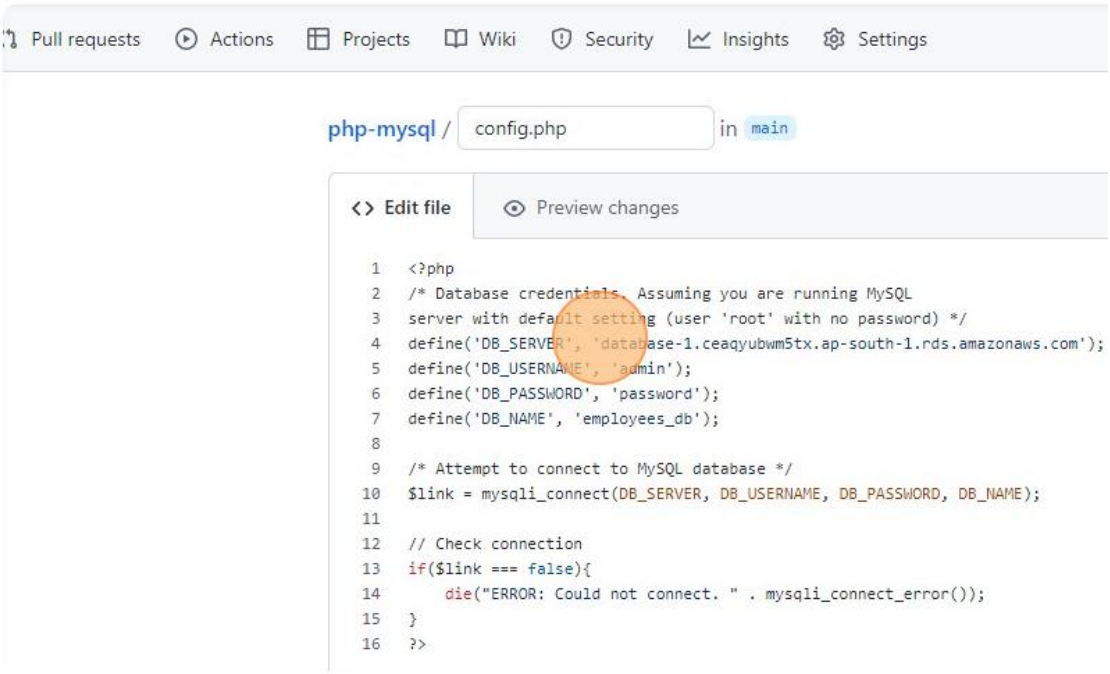
d. This will create the fork of this project in your account (Next steps have to be performed in your account)
e. Open the “config.php” file



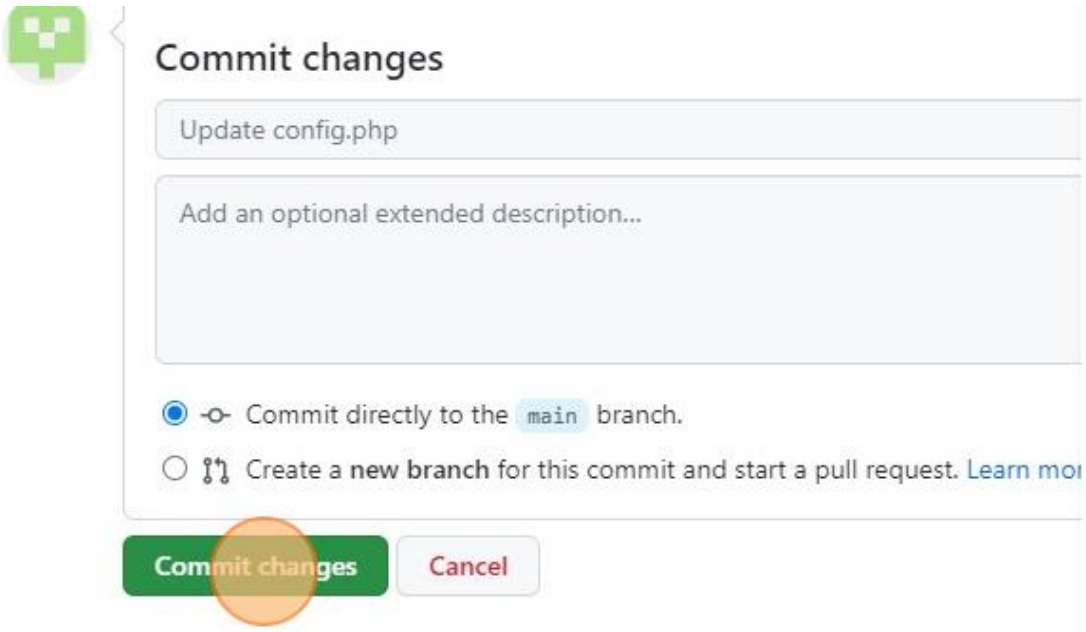
f. Edit the file by clicking on Edit icon



g. Update the value ‘DB_SERVER’ of with your db_server



h. Commit the changes in to github in your project.



13. Updating the EC2 instance with the PHP project (Below steps have to be performed in EC2 instance.

a. Remove the phpinfo.php

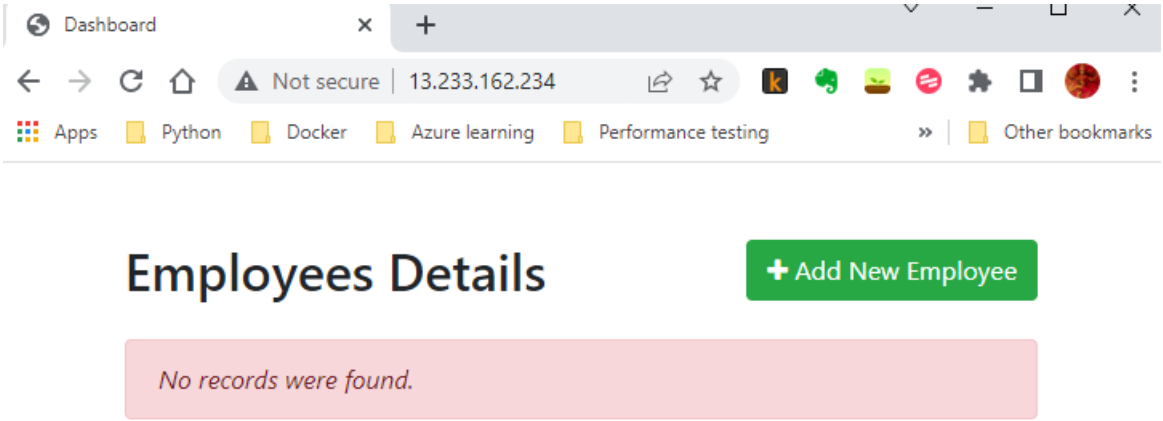
ec2-user@ip-10-0-14-191 ~]$ **rm /var/www/html/phpinfo.php**
rm: remove write-protected regular file '/var/www/html/phpinfo.php'? ***Y***

b. Clone your git project to /var/www/html folder (replace gnataraj-mf with your github account(step number 12)

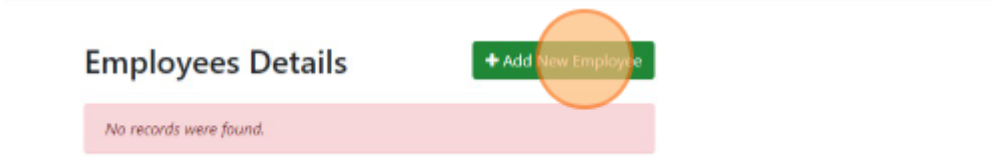
[ec2-user@ip-10-0-14-191 ~]$ **git clone https://github.com/gnataraj-mf/php-mysql.git /var/www/html/**
Cloning into '/var/www/html'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (21/21), done.
remote: Compressing objects: 100% (21/21), done.
remote: Total 21 (delta 9), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (21/21), 9.00 KiB | 3.00 MiB/s, done.
Resolving deltas: 100% (9/9), done.

14. Testing the configuration:

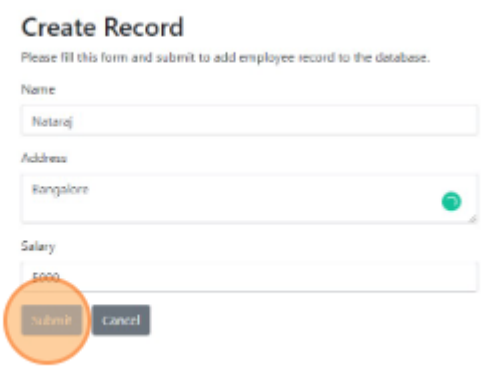
a. Access the public IPv4 IP address of the EC2 instance.



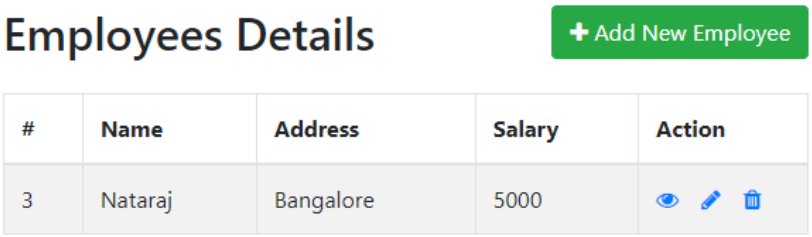
b. Click on Add New Employee



c. Create few employees with details



d.



e. Verify the db updated using mysql command:

*MySQL [employees_db]> select * from employees;*
+----+---------+-----------+--------+
| id | name | address | salary |
+----+---------+-----------+--------+
| 3 | Nataraj | Bangalore | 5000 |
+----+---------+-----------+--------+
1 row in set (0.001 sec)

Repeat the above steps a to e to Create/Read/Update and Delete (CRUD) operations.