DEPLOYMENT OF A FULLY FUNCTIONAL MY-SQL CLIENT/SERVER ARCHITECTURE

Client-Server Architecture involves the connecting together of one or more computers through a network to send and receive request between one another .This type of architecture involves each machine taking up the roles of a CLIENT who continually sends a request and the other machine receives and responds to the request sent which is known as a SERVER .

In this project we would be working with the following:

A webserver that serves a client role connects and communicates back and forth with a database server whom in turn receives and respond to the webserver.

Pre-requisite for the project is the following.

1) Fundamental Knowledge of Installing and downloading software

2) Basic Understanding of Linux Commands

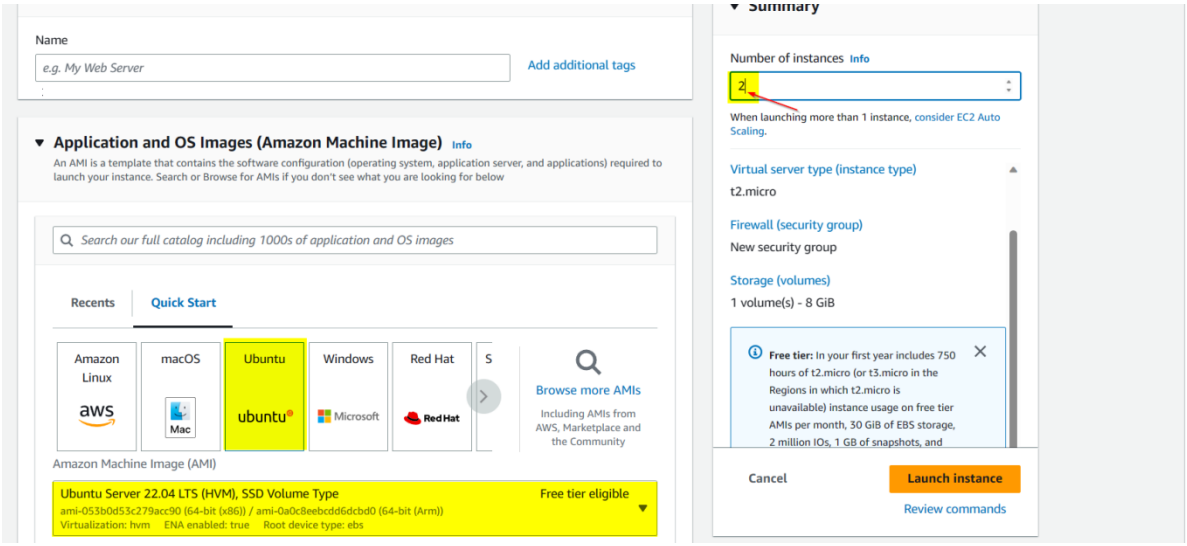
3) AWS account login with EC2 instance

4) Internet connection

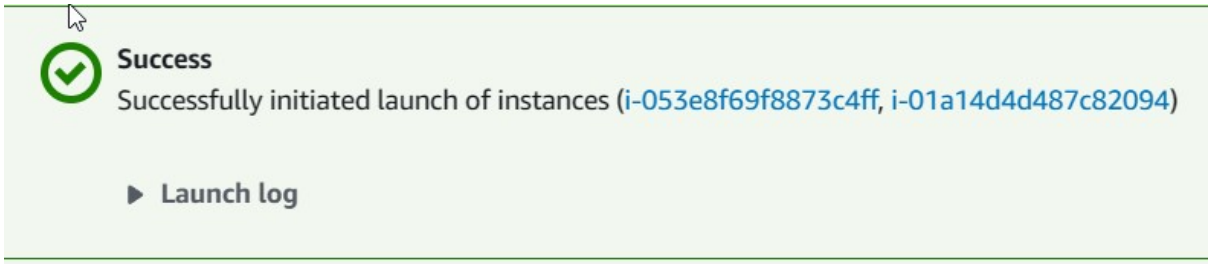
Implementation Steps:

i) Ensure you login with your details to your AWS console via the https://aws.amazon.com

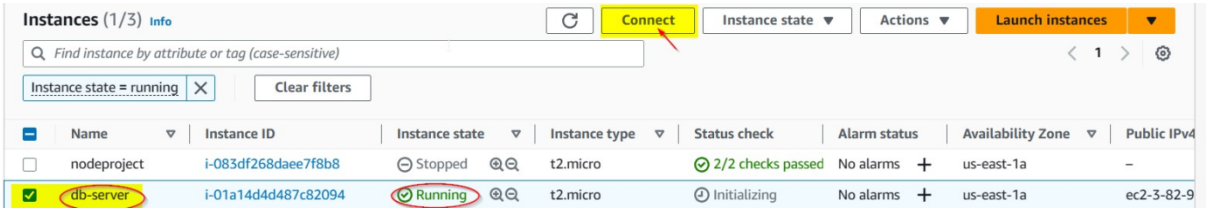
ii) Click on the EC2 link to create 2 instances namely MySQL server and MySQL client.



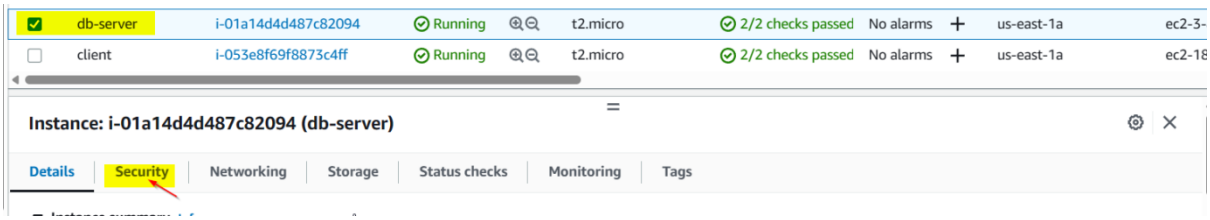
The 2 instances successfully launched and click to view Instance details with the IP address. Please note by default both instances were created in the same local virtual network, hence they would communicate on the same IP address.

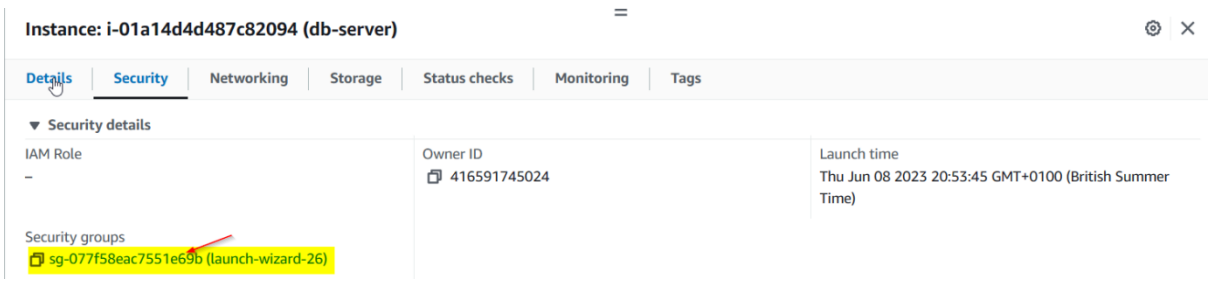


Click the “Connect” button and copy the ssh client details we would be using on the git bash console.

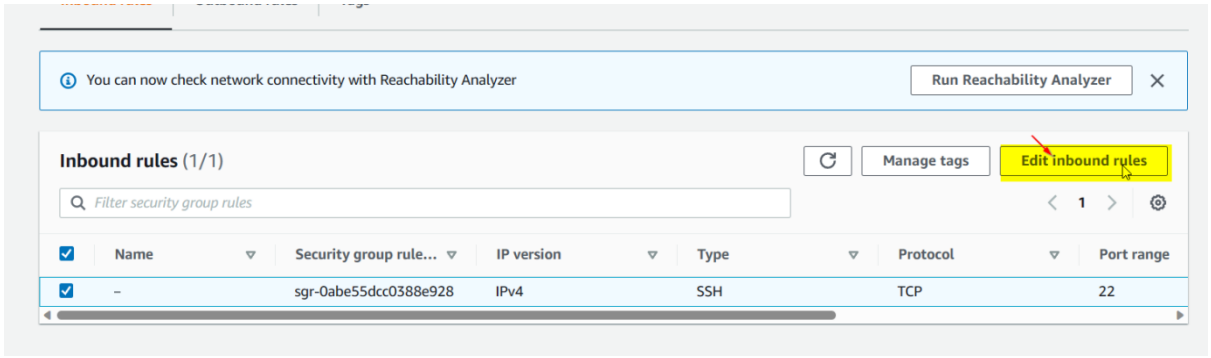


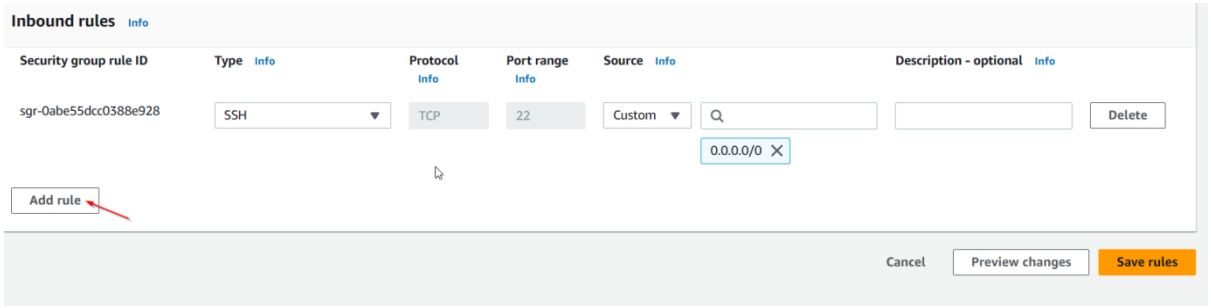
Click on security button and then proceed to click the security group link.

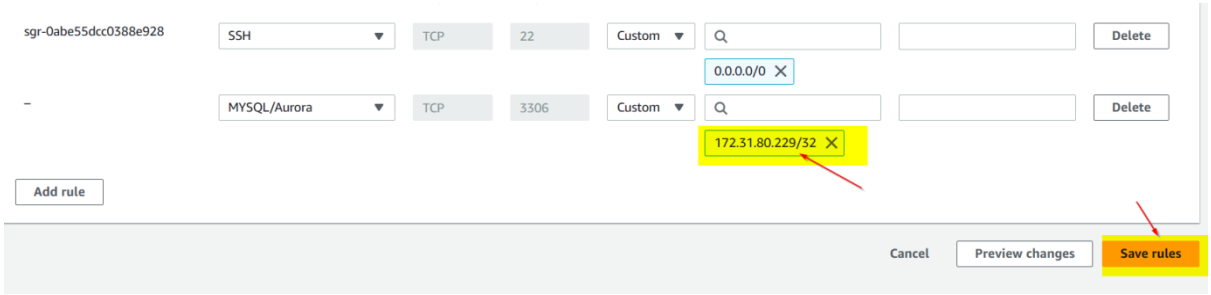




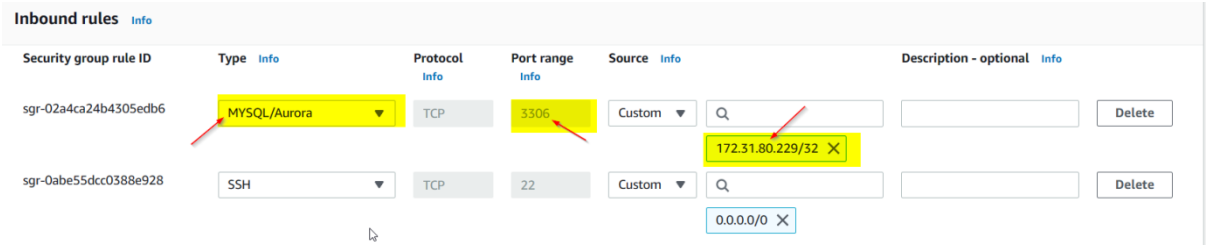
Proceed to editing the inbound rules to add and save rules. Ensure you allocate port 3306 on MYSQL server with the ip address shown below. Always make sure the Ip address is only accessible by the MySQL client server







Rules added.



Open git bash on visual studio code or whichever console is convenient to use. We are using git bash here with Visual Studio Code. Type YES to connect.

```
oshor@Oshority MINGW64 ~/Downloads (master)
$ ec2-3-82-97-183.compute-1.amazonaws.com
bash: ec2-3-82-97-183.compute-1.amazonaws.com: command not found

oshor@Oshority MINGW64 ~/Downloads (master)
$ ssh -i "mysqlserverclient.pem" ubuntu@ec2-3-82-97-183.compute-1.amazona
ws.com
The authenticity of host 'ec2-3-82-97-183.compute-1.amazonaws.com (3.82.9
7.183)' can't be established.
ED25519 key fingerprint is SHA256:FijwFrXcscxakXsoZcz3B5BZJ9FG38mTGkr+dnk
YcIc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

Proceed to updating the lists of packages in the package manager.

```
ubuntu@ip-172-31-81-225:~$ sudo apt update -y
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRele
ase [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRe
lease [108 kB]
```

Then we proceed to install MySQL server

```
ubuntu@ip-172-31-81-225:~$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgi-fast-perl libcgi-pm-perl libclone-perl libencode-locale-perl
```

We enable the MySQL server and then proceed to secure the server using a blank password .

```
ubuntu@ip-172-31-81-225:~$ sudo systemctl enable mysql
Synchronizing state of mysql.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mysql
ubuntu@ip-172-31-81-225:~$ sudo my_sql_secure_installation
sudo: my_sql_secure_installation: command not found
ubuntu@ip-172-31-81-225:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Connecting to MySQL using a blank password.
```

After this we connect back to our MySQL database server to create a data base ,grant all permission for the user created and finally flush privileges and quit the MySQL

```
Last login: Thu Jun  8 20:48:28 2023 from 81.152.237.15
ubuntu@ip-172-31-81-225:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 16
Server version: 8.0.33-0ubuntu0.22.04.2 (Ubuntu)

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 state
ment.

mysql>  ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_passwo
rd BY 'PassWord.1';
Query OK, 0 rows affected (0.00 sec)

mysql> xit
    -> exit
    ->  exit
    ->
    ->
    ->
    -> exit
    -> ^C
mysql> ^C
mysql> ^C
mysql> ^C
mysql> ^C
mysql> CREATE USER 'remote_user'@'%' IDENTIFIED WITH mysql_native_passwor
d BY 'password';
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE DATABASE test_db;
Query OK, 1 row affected (0.02 sec)

mysql> GRANT ALL ON test_db.* TO 'remote_user'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.01 sec)

mysql> exit
Bye
ubuntu@ip-172-31-81-225:~$
```

We would need to configure MySQL server to allow connection to remote host as shown below by editing the .cnf file and change the bind address to 0.0.0.0 as seen below

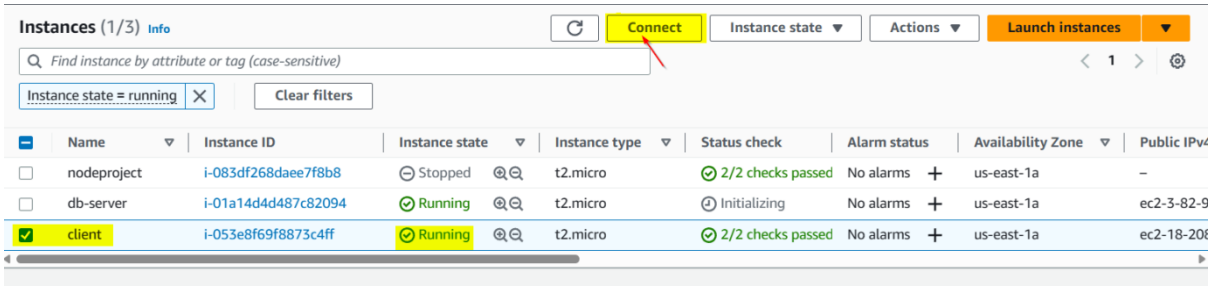
```
Bye
ubuntu@ip-172-31-81-225:~$ sudo vi /etc/mysql/mysql.conf.d/mysql.cnf
ubuntu@ip-172-31-81-225:~$ sudo vi /etc/mysql/mysql.conf.d/mysql.cnf
ubuntu@ip-172-31-81-225:~$ sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
ubuntu@ip-172-31-81-225:~$ sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
mysqlx-bind-address     = 127.0.0.1
#
# * Fine Tuning
#
```

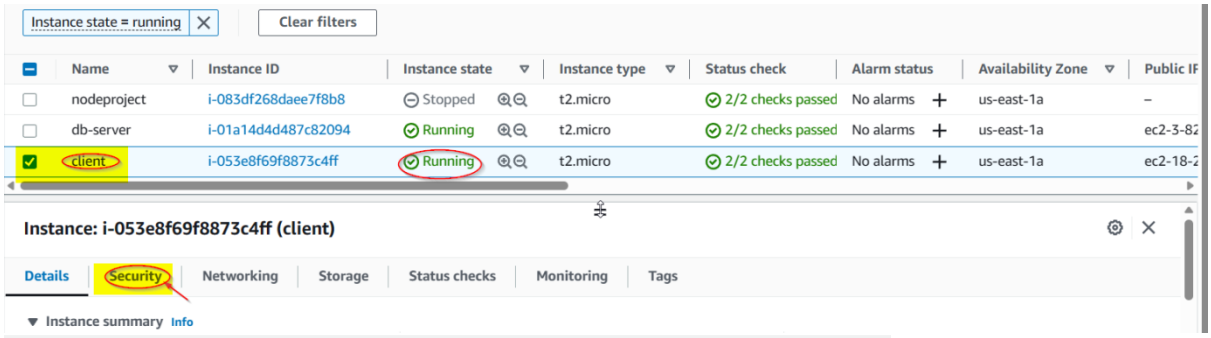
CONFIGURATION OF MYSQL – CLIENT

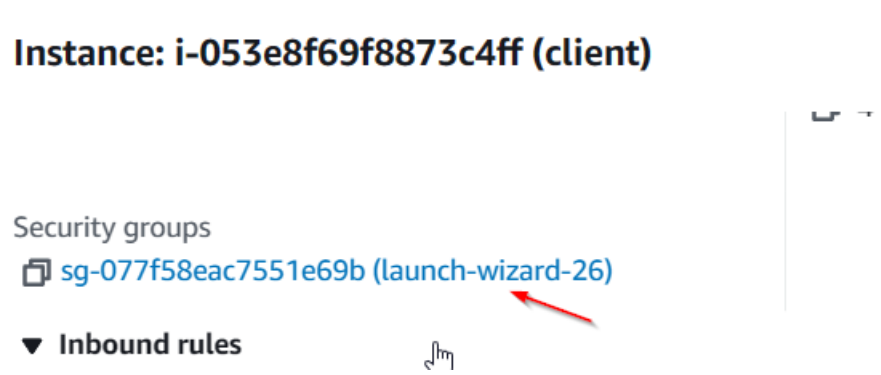
Then we go back to AWS to connect the MYSQL -client server

Click the “Connect” button and copy the ssh client details we would be using on the git bash console.

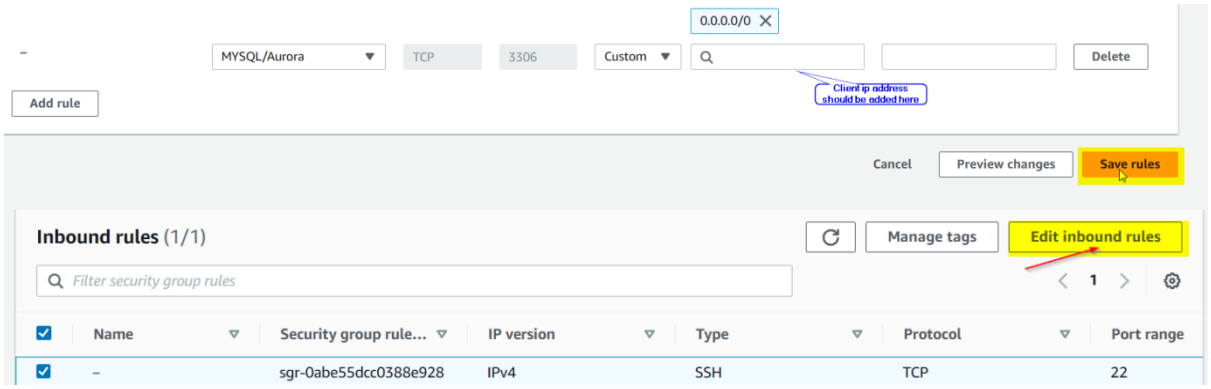


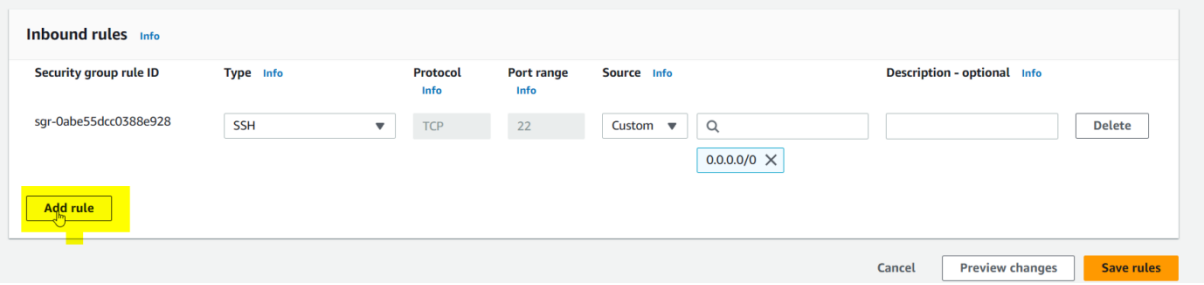
Click on security button and then proceed to click the security group link.

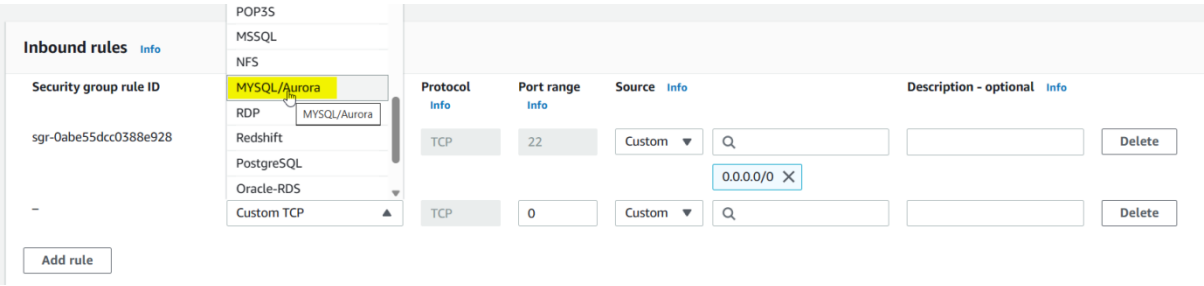




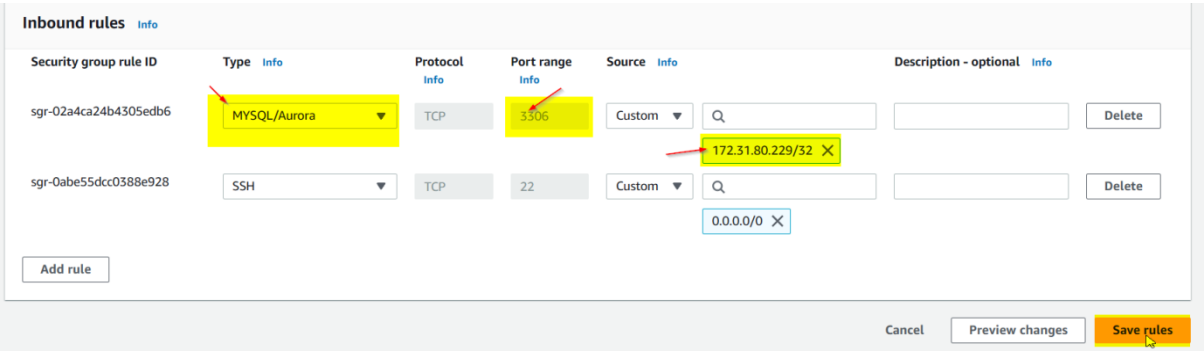
Proceed to editing the inbound rules to add and save rules. Ensure you allocate port 3306 on MYSQL server with the ip address shown below .Always make sure the Ip address is only accessible by the MySQL client server



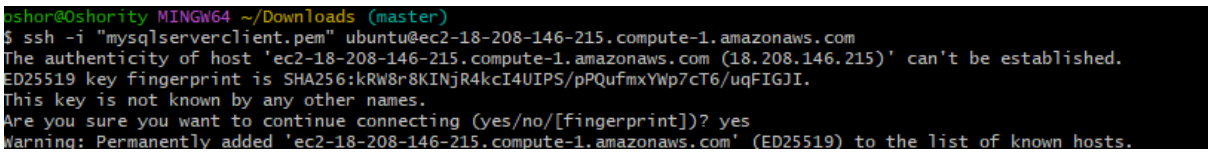




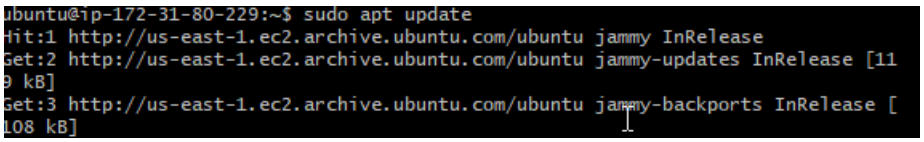
Rules added



Open git bash on visual studio code or whichever console is convenient to use. We are using git bash here with Visual Studio Code. Type YES to connect.



Proceed to updating the lists of packages in the package manager.



Then we proceed to install MySQL client and confirm the initial Ip address on the server

```
ubuntu@ip-172-31-80-229:~$ sudo apt install mysql-client -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
```

```
ubuntu@ip-172-31-80-229:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9001 qdisc fq_codel state UP group default qlen 1000
    link/ether 12:1e:6c:47:59:b1 brd ff:ff:ff:ff:ff:ff
    inet 172.31.80.229/20 metric 100 brd 172.31.95.255 scope global dynamic eth0
       valid_lft 3512sec preferred_lft 3512sec
    inet6 fe80::101e:6cff:fe47:59b1/64 scope link
       valid_lft forever preferred_lft forever
```

We then connect with the command below to the remote database server.

```
ubuntu@ip-172-31-80-229:~$ sudo mysql -u remote_user -h 172.31.81.225 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.33-0ubuntu0.22.04.2 (Ubuntu)

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 confirm by showing the databases as shown below.

```
mysql> Show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| performance_schema |
| test_db            |
+--------------------+
3 rows in set (0.01 sec)

mysql>
```

This shows that we have successfully deployed a fully functional My-SQL Client-Server set up.