COMPLETED PROJECT 5 REVIEWED

Task: Implement a Client Server Architecture using MySQL Database Management System (DBMS).

Goal/Aim: To deploy a fully functional mysql client-server set up.

Step1: Create and configure two Linux-based virtual servers (EC2 instances in AWS).

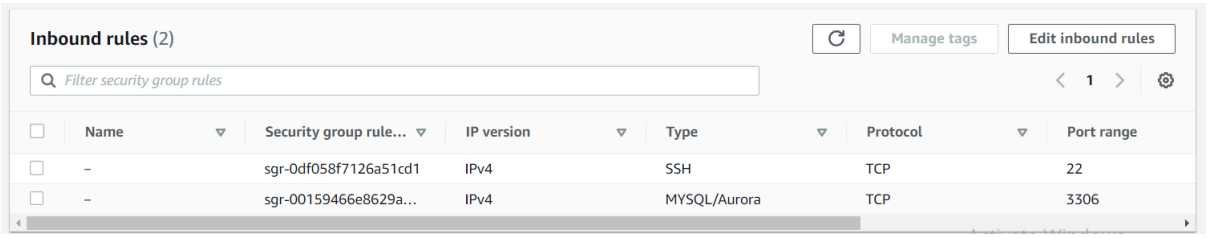
- Installing mysql server:

Sudo apt update and sudo apt install mysql-server

- Installing mysql client:

Sudo apt update and sudo apt install mysql-client

Step2: Open port 3306 on mysql server:



Step3: Configure MySQL server to allow connections from remote hosts.

- Open the mysql server daemon configuration file and replace 127.0.0.1 to 0.0.0.0

Sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf

```
#
# The MySQL database server configuration file.
#
# One can use all long options that the program supports.
# Run program with --help to get a list of available options and with
# --print-defaults to see which it would actually understand and use.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

# Here is entries for some specific programs
# The following values assume you have at least 32M ram

[mysqld]
#
# * Basic Settings
#
user            = mysql
# pid-file      = /var/run/mysqld/mysqld.pid
# socket        = /var/run/mysqld/mysqld.sock
# port          = 3306
# datadir       = /var/lib/mysql


# If MySQL is running as a replication slave, this should be
# changed. Ref https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_tmpdir
# tmpdir                = /tmp
#
# 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
#
key_buffer_size         = 16M
# max_allowed_packet    = 64M
# thread_stack          = 256K

# thread_cache_size       = -1
```


CONFIGURING MYSQL SERVER

Step1:

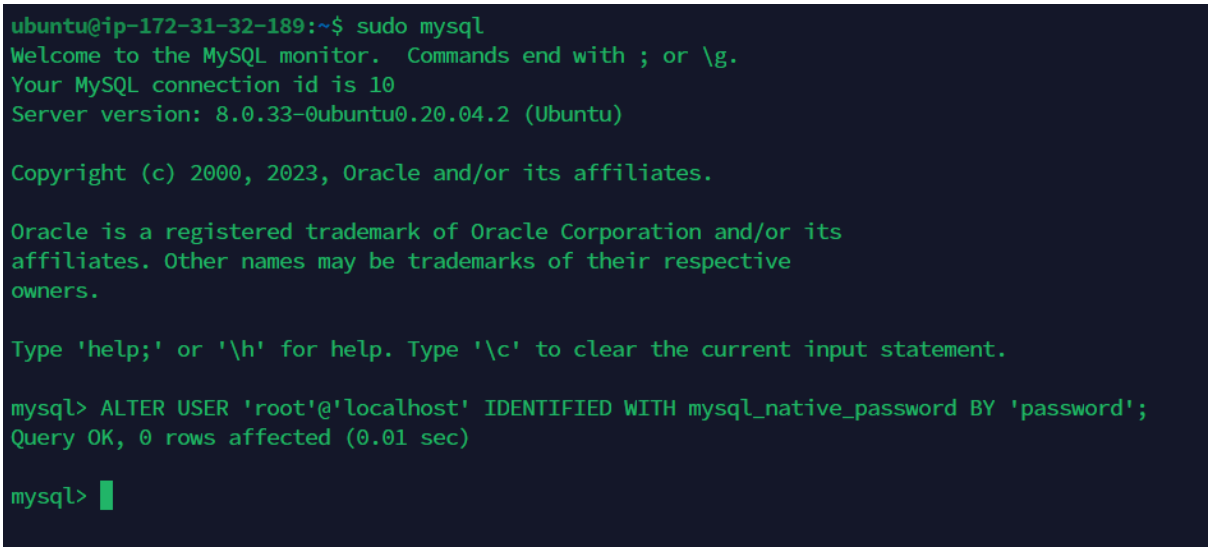
- **Login to mysql server**

sudo mysql

```
ubuntu@ip-172-31-32-189:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 10
Server version: 8.0.33-0ubuntu0.20.04.2 (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 run the following `ALTER USER` command to change the root user's authentication method to one that uses a password. The following example changes the authentication method to `mysql_native_password`:

ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY 'password';



- After making this change exit the mysql prompt:
  exit

Now you can run secure_installation_mysql without issue
sudo mysql_secure_installation

This will take you through a series of prompts where you can make some changes to your MySQL installation's security options. The first prompt will ask whether you'd like to set up the Validate Password Plugin, which can be used to test the password strength of new MySQL users before deeming them valid.

If you elect to set up the Validate Password Plugin, any MySQL user you create that authenticates with a password will be required to have a password that satisfies the policy you select:

```
ubuntu@ip-172-31-32-189:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:
Error: Access denied for user 'root'@'localhost' (using password: YES)
ubuntu@ip-172-31-32-189:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:

VALIDATE PASSWORD COMPONENT can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD component?

Press y|Y for Yes, any other key for No: y

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary
```

```
Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
Using existing password for root.

Estimated strength of the password: 50
Change the password for root ? ((Press y|Y for Yes, any other key for No) : y

New password:

Re-enter new password:

Estimated strength of the password: 100
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No)
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : y
Success.


Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.

By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
 - Dropping test database...
Success.
```

```
 - Removing privileges on test database...
Success.

Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.

All done!
ubuntu@ip-172-31-32-189:~$
```

Once the script completes, your MySQL installation will be secured. You can now move on to creating a dedicated database user with the MySQL client.

CREATE A DEDICATED MYSQL USER AND GRANT PRIVILEGES

Step1:

- **Login to Mysql Server**

  mysql -u root -p

```
ubuntu@ip-172-31-32-189:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.33-0ubuntu0.20.04.2 (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>
```

Step2:

Create a new user:

CREATE USER 'jude'@'%' IDENTIFIED WITH mysql_native_password BY 'password';

```
mysql> CREATE USER 'jude'@'%' IDENTIFIED WITH mysql_native_password BY 'password';
Query OK, 0 rows affected (0.01 sec)
```

After creating your new user (jude), you can grant jude the appropriate privileges. The general syntax for granting user privileges is as follows:

GRANT ALL PRIVILEGES ON *.* TO 'jude'@'%' WITH GRANT OPTION;

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'jude'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.04 sec)
```

Following this, it's good practice to run the `FLUSH PRIVILEGES` command. This will free up any memory that the server cached as a result of the preceding `CREATE USER` and `GRANT` statements:

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

mysql>
```

Now Create database:

```
mysql> CREATE DATABASE test_db;
Query OK, 1 row affected (0.01 sec)
```

exit

Now go to the mysql client server and run this command:

sudo mysql -u jude -h 172.31.32.189 -p

The IP address there is the IP address of the database server (mysqlserve)

And then type the user jude password:

```
ubuntu@ip-172-31-46-95:~$ sudo mysql -u jude -h 172.31.32.189 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.33-0ubuntu0.20.04.2 (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>
```

To confirm check to see the database test_db that was created before:

```
mysql> show database;
show database;
^C
mysql> Show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test_db            |
+--------------------+
5 rows in set (0.01 sec)

mysql>
```