

## CLIENT/SERVER ARCHITECTURE USING A MYSQL RELATIONAL DATABASE MANAGEMENT SYSTEM

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

Server A name - `mysql server`

Server B name - `mysql client`

<input checked="" type="checkbox"/>	Server A Name _mysql server	i-09c03f7a622f87293
<input checked="" type="checkbox"/>	Server B Name _mysql client	i-0726d0a05e581ceda

On the server side/Client side, we would run the following command below:

```
Update ubuntu  
sudo apt update
```

```
Upgrade ubuntu  
sudo apt upgrade -y
```

Then on the server side we proceed to Install mysql using the command below  
`sudo apt install mysql-server -y`

2. On mysql server Linux Server install MySQL Server software.

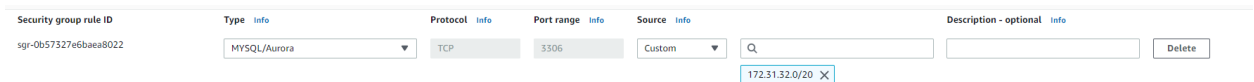
Make sure to enable the MySQL.service after installing  
`sudo systemctl enable mysql`

```
ubuntu@ip-172-31-23-192:~$ sudo apt install mysql-server -y  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
mysql-server is already the newest version (8.0.34-0ubuntu0.22.04.1).  
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.  
ubuntu@ip-172-31-23-192:~$ 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-172-31-23-192:~$
```

- On mysql client Linux Server install MySQL Client software.  
`sudo apt install mysql-client -y`

```
ubuntu@ip-172-31-23-192:~$ sudo apt install mysql-client -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
mysql-client is already the newest version (8.0.34-0ubuntu0.22.04.1).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-23-192:~$
```

- Use mysql server's local IP address to connect from mysql client. MySQL server uses TCP port 3306 by default, so you will have to open it by creating a new entry in 'Inbound rules' in 'mysql server' Security Groups. For extra security, do not allow all IP addresses to reach your 'mysql server' – allow access only to the specific local IP address of your 'mysql client'.



Security group rule ID	Type	Protocol	Port range	Source	Description - optional	Actions
sgr-0b57327e6baea8022	MySQL/Aurora	TCP	3306	Custom	172.31.32.0/20	Delete

- For MySQL secure installation use the following

```
ubuntu@ip-172-31-29-77:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.
```

- After the installation you might need to create a password for root user
- On MySQL server create a user and a database

```
mysql> CREATE USER 'remote_user'@'%' IDENTIFIED WITH mysql_native_password BY 'password';
Query OK, 0 rows affected (0.02 sec)
```

- Create a test.db and grant privileges

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

mysql> GRANT ALL ON test_db.* TO 'remote_user'@'%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'PRIVILEGES' at line 1
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)
```

9. You might need to configure MySQL server to allow connections from remote hosts.

sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf and Replace '127.0.0.1' to '0.0.0.0' like this:

```
# 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
"/etc/mysql/mysql.conf.d/mysqld.cnf" 78L, 2218C 31,1
```

10. From mysql client Linux Server connect remotely to mysql server Database Engine without using SSH. You must use the mysql utility to perform this action.

Check that you have successfully connected to a remote MySQL server and can perform SQL queries:

```
ubuntu@ip-172-31-40-218:~$ sudo mysql -u remote_user -h 172.31.32.53 -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.33-0ubuntu0.20.04.2 (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.
```

```
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> show databases;
+-----+
| Database |
+-----+
| information_schema |
| performance_schema |
| test_db |
+-----+
3 rows in set (0.00 sec)

mysql> |
```