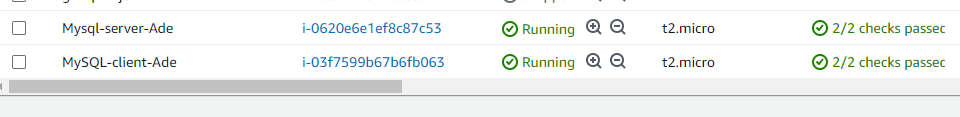
MySQL is an open-source database management system, commonly installed as part of the popular [LAMP](https://www.digitalocean.com/community/tutorials/how-to-install-lamp-stack-on-ubuntu) (Linux, Apache, MySQL, PHP/Python/Perl) stack. It implements the relational model and uses Structured Query Language (better known as SQL) to manage its data

This project demonstrates a basic client-server using MySQL RDBMS, follow the below instructions

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

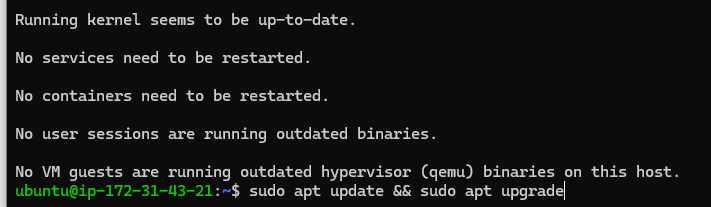
Server A name - `mysql server`

Server B name - `mysql client`



1. On mysql server Linux Server install MySQL **Server** software.

## Sudo apt Update && upgrade to update your operating system and install the updates.



## sudo apt install mysql-server

Start the service, to start you might be prompted to restart your server.

## sudo systemctl start mysql.service

These commands will install and start MySQL, but will not prompt you to set a password or make any other configuration changes. this leaves your installation of MySQL insecure.

## 3. You can skip this for the test and move to step 4.

## [**Configuring MySQL**](https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04#step-2-configuring-mysql)

For fresh installations of MySQL, you’ll want to run the DBMS’s included security script. This script changes some of the less secure default options for things like remote root logins and sample users.

As of July 2022, an error will occur when you run the mysql\_secure\_installation script without some further configuration. The reason is that this script will attempt to set a password for the installation’s root MySQL account but, by default on Ubuntu installations, this account is not configured to connect using a password.

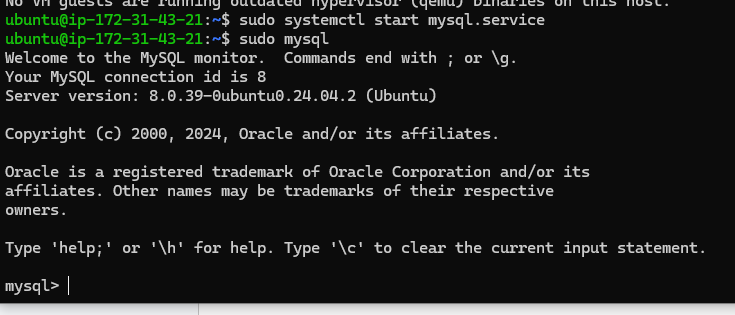
Before July 2022, this script would silently fail after attempting to set the root account password and continue with the rest of the prompts. However, as of this writing the script will return an error.

This will lead the script into a recursive loop which you can only get out of by closing your terminal window.

Because the mysql\_secure\_installation script performs several other actions that are useful for keeping your MySQL installation secure, it’s still recommended that you run it before you begin using MySQL to manage your data. To avoid entering this recursive loop, though, you’ll need to first adjust how your root MySQL user authenticates.

First, open up the MySQL prompt:

## sudo 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:

##Mysql exit

Following that, you can run the mysql\_secure\_installation script without issue.

Once the security script completes, you can then reopen MySQL and change the root user’s authentication method back to the default.

To authenticate as the root MySQL user using a password, run this command:

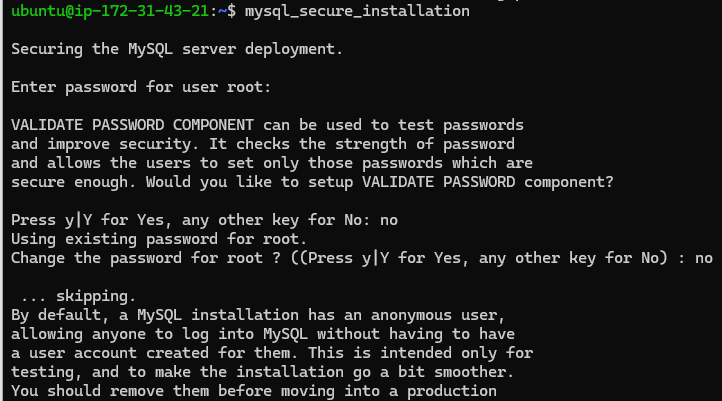
## mysql -u root -p

Then go back to using the default authentication method using this command:

## ALTER USER 'root'@'localhost' IDENTIFIED WITH auth\_socket;

Run the security script with sudo:

## 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. The strongest policy level — which you can select by entering 2 — will require passwords to be at least eight characters long and include a mix of uppercase, lowercase, numeric, and special characters:

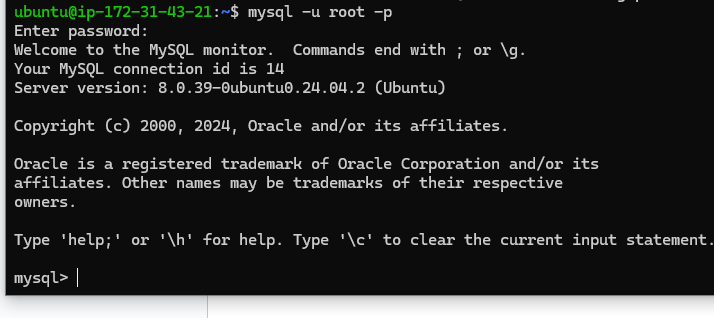
From there, you can press Y and then ENTER to accept the defaults for all the subsequent questions. This will remove some anonymous users and the test database, disable remote root logins, and load these new rules so that MySQL immediately respects the changes you have made.

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.

## [**Step 4 — Creating a Dedicated MySQL User and Granting Privileges**](https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04#step-3-creating-a-dedicated-mysql-user-and-granting-privileges)

Upon installation, MySQL creates a root user account which you can use to manage your database. This user has full privileges over the MySQL server, meaning it has complete control over every database, table, user, and so on. Because of this, it’s best to avoid using this account outside of administrative functions. This step outlines how to use the root MySQL user to create a new user account and grant it privileges.

## mysql -u root -p or sudo mysql

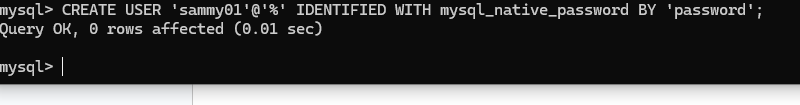


Once you have access to the MySQL prompt, you can create a new user with a CREATE USER statement. These follow this general syntax:

## CREATE USER 'sammy'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'password';

After CREATE USER, you specify a username. This is immediately followed by an @ sign and then the hostname from which this user will connect. If you only plan to access this user locally from your Ubuntu server, you can specify localhost. Wrapping both the username and host in single quotes isn’t always necessary, but doing so can help to prevent errors.

Or use. ## CREATE USER 'sammy01'@'%' IDENTIFIED WITH mysql\_native\_password BY 'password'; this grant access from any host.



1. On mysql client Linux Server install MySQL **Client** software.

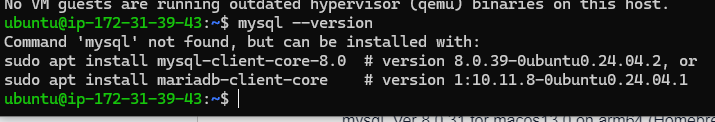
## Sudo apt Update && upgrade to update your operating system and install the updates.



Before you start, you should confirm that you don't have MySQL client installed. If you have installed MySQL server before, likely you will have MySQL client installed as well. If you see something like below, then MySQL client is already installed and you can stop reading now.

$ mysql --version

You will see the error below if MYSQL is not installed.



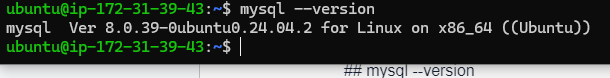
Install the MySQL client by running the following command:

## sudo apt-get install mysql-client -y

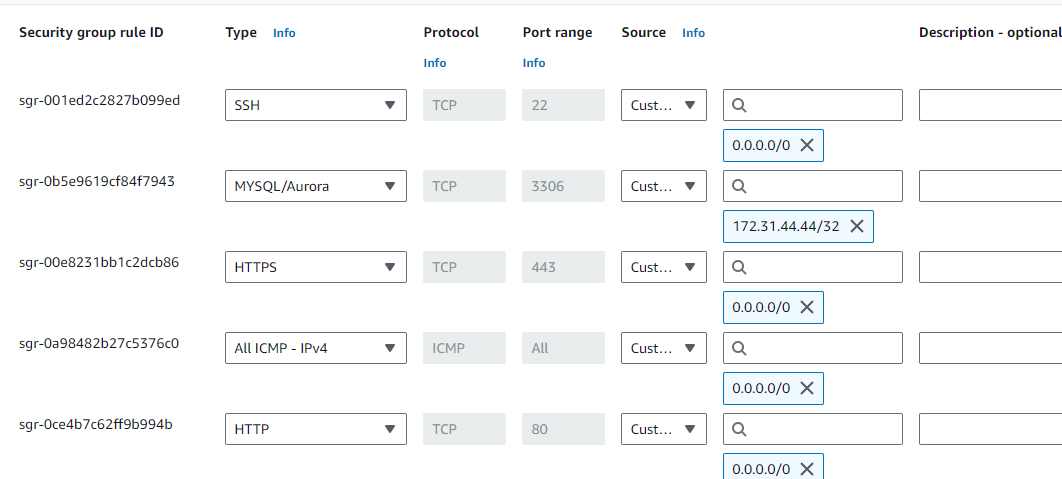


After the installation is complete, you can verify that the MySQL client is installed by running the following command:

## mysql --version



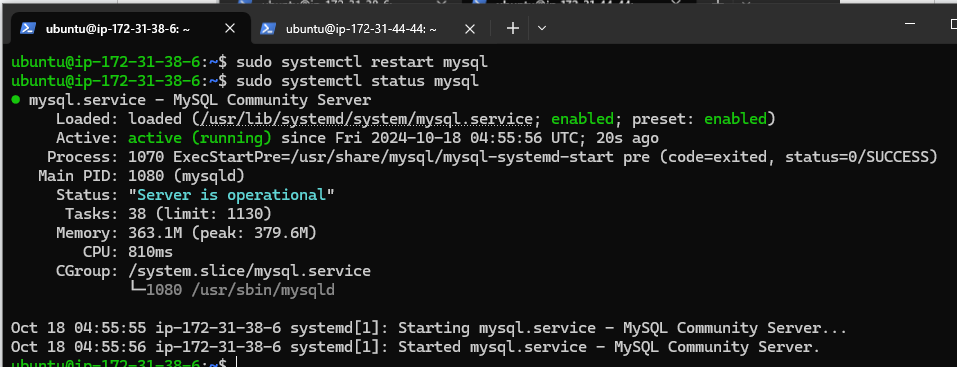
1. By default, both of your EC2 virtual servers are located in the same local virtual network, so they can communicate using **local IP addresses**. Use mysql server's local IP address to connect from MySQL client. MySQL server uses TCP port 3306 by default, so you will have to open it by creating a 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'.



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

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

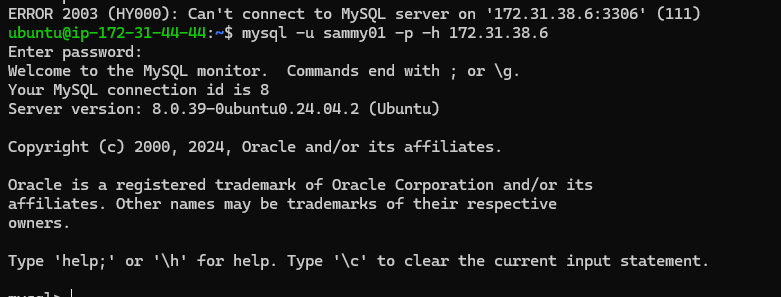
Replace '127.0.0.1' to '0.0.0.0' like this: restart your service after this configuration change.



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

## mysql -u sammy01 -p -h your\_hostname or private IP, for example, mysql -u sammy -p -h 172.31.38.6

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



Show databases;

If you see an output similar to the below image, then you have successfully completed this project you have deployed a fully functional MySQL Client-Server setup. Well Done! You are getting there gradually. You can further play around with this setup and practice creating/dropping databases & tables and inserting/selecting records to and from them.

