

Lab 01: Install MySQL and Create User Account

Faculty: Dr. Odelu Vanga

Date: 10-08-2020

Topics to be covered:

- Install MySQL
- Start MySQL Server
- Start MySQL shell
- Adjusting User Authentication and Privileges
- Create User Account
- Connect to the MySQL

Install MySQL

MySQL is an Open-source database management system. It uses a relational database and SQL (Structured Query Language) to manage its data.

To install MySQL server, type the below commands in terminal (assume that the OS is Ubuntu):

- `$ sudo apt update`
- `$ sudo apt install mysql-server`
- `$ sudo mysql_secure_installation utility`

This utility prompts you to define the *mysql* root password and other security-related options, including removing remote access to the root user and setting the root password.

Remark 1 *For default settings, enter yes for all permissions.*

Start the MySQL service

After the installation is complete, you can start the database service by running the following command. If the service is already started, a message informs you that the service is already running:

- `$ sudo systemctl start mysql`

Start the mysql shell

There is more than one way to work with a MySQL server, but this article focuses on the most basic and compatible approach, the mysql shell.

At the command prompt, run the following command to launch the mysql shell and enter it as the root user:

- `$ mysql -u root -p`

Remark 2 *Note that even though you have set a password for the root MySQL user, this user is not configured to authenticate with a password when connecting to the MySQL shell. If you would like, you can adjust this setting.*

Remark 3 *In Ubuntu systems running MySQL 5.7 (and later versions), the root MySQL user is set to authenticate using the `auth_socket` plugin by default rather than with a password. This allows for some greater security and usability in many cases, but it can also complicate things when you need to allow an external program*

(Optional) Adjusting User Authentication and Privileges

In order to use a password to connect to MySQL as root, you will need to switch its authentication method from `auth_socket` to `mysql_native_password`. To do this, open up the MySQL prompt from your terminal:

- `$ sudo mysql`

Enter your system password in the prompt. Next execute the following commands:

```
mysql> SELECT user, authentication_string, plugin, host
FROM mysql.user;
```

will display the following table:

user	authentication_string	plugin	host
root		auth_socket	localhost
mysql.session	*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE	mysql_native_password	localhost
mysql.sys	*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE	mysql_native_password	localhost
debian-sys-maint	*DF057A49A4501735F04F2D5135A2ECEDBEE7C7AB	mysql_native_password	localhost
4 rows in set (0.00 sec)			

We can see that the root user does in fact authenticate using the `auth_socket` plugin. To configure the root account to authenticate with a password, run the following `ALTER USER` command. Be sure to change password to a **strong password (at least one capital, one small, one digit, and one symbol)** of your choosing, and note that this command will change the root password you set above.

```
mysql> ALTER USER 'root'@'localhost'
IDENTIFIED WITH mysql_native_password
BY 'Odelu@2020';
```

Then, run **FLUSH PRIVILEGES** which tells the server to reload the grant tables and put your new changes into effect:

```
mysql> FLUSH PRIVILEGES;
```

Check the authentication methods employed by each of your users again to confirm that root no longer authenticates using the *auth_socket* plugin:

```
mysql> SELECT user, authentication_string, plugin, host
FROM mysql.user;
```

Display the following table:

user	authentication_string	plugin	host
root	*8B192D87421646C89CAC45CE5EE149CF05986941	mysql_native_password	localhost
mysql.session	*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE	mysql_native_password	localhost
mysql.sys	*THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE	mysql_native_password	localhost
debian-sys-maint	*DF057A49A4501735F04F2D5135A2ECEDBEE7C7AB	mysql_native_password	localhost

4 rows in set (0.00 sec)

You can see in the above output that the root MySQL user now authenticates using a password. Once you confirm this on your own server, you can exit the MySQL shell:

```
mysql> exit
```

Create User Account

The following will run your MySQL client with regular user privileges, and you will gain administrator privileges within the database by authenticating, and create a new user (*dbms*) and give it a password (here *CSEdbms@2020*):

```
% mysql -u root -p
Enter password: *****
mysql> CREATE USER 'dbms'@'localhost' IDENTIFIED BY 'CSEdbms@2020';
mysql> GRANT ALL ON root.* TO 'dbms'@'localhost';
Query OK, 0 rows affected (0.09 sec)
mysql> quit
Bye
```

Note: Given no explicit hostname, MySQL assumes that the server is running on the local host.

Connect to the MySQL

After creating the *dbms* account, verify that you can use it to connect to the MySQL server. From the host that was named in the CREATE USER statement, run the following command to do this (the host named after *-h* should be the host where the MySQL server is running):

```
% mysql -h localhost -u dbms -p
Enter password: ***** the above password
```

```
drodelu@dr-odelu:~$ mysql -u dbms -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.7.31-0ubuntu0.18.04.1 (Ubuntu)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

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> █
```

References

- *Database System Concepts*, Silberschatz, H. Korth and S. Sudarshan, McGraw-Hill Education, 6th edition, 2010.
- <https://dev.mysql.com/doc/mysql-installation-excerpt/5.7/en/>