

Laboratory work 6

What I did:

Part 1: “Do some research, choose and locally install open source database (postgresql, mariadb, mysql...)”

Idea is to get some open source database management system installed on the VM OS. Reason for using one or the other option is same as ever – first that popped up with a decent installation tutorial. This time it is MySQL – another “Digital Ocean” tutorial. MySQL is part of Oracle corporation. If distributing your software as open source then it can fall under GPL license, but for commercial use the license from Oracle must be bought.

To get the MySQL server it is again a simple install command: `sudo apt-get install mysql-server`. First time ever I’ve seen that a special input is required while installing a package – it asked to set a password for a root user and of course I used “admin”. Next I run the suggested automatic configuration script: `sudo mysql_secure_installation` (Fig. 1). I left mostly default/testing values as this is by no means a production thing. Using `systemctl status mysql.service` I test if MySQL is running and it is (Fig. 2).

Part 2: “Configure the database user (create initial user and set password)”

To start off I connect to MySQL DBMS using root account (Fig. 3). And there is two things to do to make a user: one creates the user and another gives him privileges (Fig. 4). In my case I basically made the main OS user have near same rights as the root user when connecting from localhost with the second best password ever: “password”.

Part 3: “Create first database with the name “mydb” and connect to it”

Single SQL command to achieve this (Fig. 5).

Part 4: Working with the database

First a table is required for storing stuff in it. Will make the example one (Fig. 6). This creates an empty table (Fig. 7). An then there is the simple insert statement specifying into which column what data to insert (Fig. 8). And that concludes basic functionality of working with a database.

```
et@FailBox:~$ sudo mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:

VALIDATE PASSWORD PLUGIN 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 plugin?

Press y|Y for Yes, any other key for No: n
Using existing password for root.
Change the password for root ? ((Press y|Y for Yes, any other key for No) : n

... skipping.
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) : n

... skipping.

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)
: n

... skipping.
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!
```

Fig. 1.

```
et@FailBox:~$ systemctl status mysql.service
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: en
   Active: active (running) since Sun 2017-10-15 02:34:26 PDT; 6min ago
     Main PID: 26058 (mysqld)
       CGroup: /system.slice/mysql.service
               └─26058 /usr/sbin/mysqld

Oct 15 02:34:24 FailBox systemd[1]: Starting MySQL Community Server...
Oct 15 02:34:26 FailBox systemd[1]: Started MySQL Community Server.
lines 1-9/9 (END)
```

Fig. 2.

```
et@FailBox:~$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 13
Server version: 5.7.11-0ubuntu6 (Ubuntu)

Copyright (c) 2000, 2016, 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> █
```

Fig. 3.

```
mysql> CREATE USER 'et'@'localhost' IDENTIFIED BY 'password';
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON *.* TO 'et'@'localhost' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)
```

Fig. 4.

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)

mysql> CREATE DATABASE mydb;
Query OK, 1 row affected (0.00 sec)

mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mydb |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)
```

Fig. 5

```
mysql> SHOW TABLES
-> ;
Empty set (0.00 sec)

mysql> CREATE TABLE TEST(ID INT PRIMARY KEY NOT NULL, NAME TEXT NOT NULL, PHONE
TEXT NOT NULL);
Query OK, 0 rows affected (0.04 sec)

mysql> SHOW TABLES;
+-----+
| Tables_in_mydb |
+-----+
| TEST |
+-----+
```

Fig .6.

```
mysql> SELECT * FROM TEST;
Empty set (0.18 sec)
```

Fig. 7.

```
mysql> INSERT INTO TEST (ID, NAME, PHONE) VALUES (1, 'Vardenis', '+37062819293')
;
Query OK, 1 row affected (0.01 sec)

mysql> SELECT * FROM TEST;
+-----+-----+-----+
| ID | NAME      | PHONE      |
+-----+-----+-----+
| 1 | Vardenis | +37062819293 |
+-----+-----+-----+
```

Fig. 8