Install MySql on your Ubuntu server

This guide is based on these web resources:

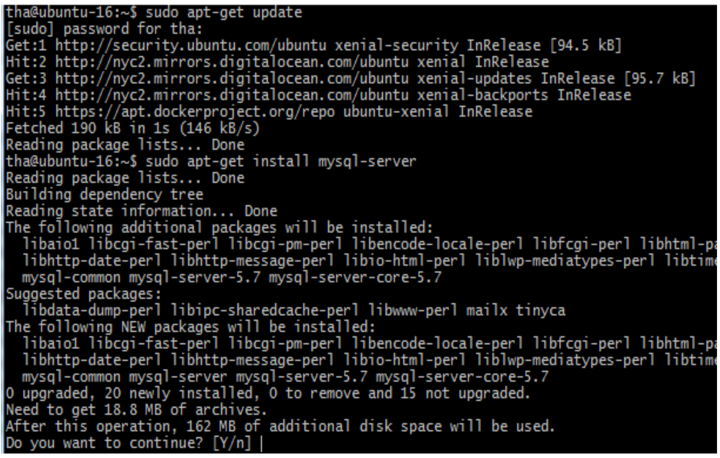
https://www.digitalocean.com/community/tutorials/how-to-install-linux-apache-mysql-php-lamp-stack-on-ubuntu-16-04 and

https://www.digitalocean.com/community/tutorials/a-basic-mysql-tutorial and

https://www.digitalocean.com/community/tutorials/how-to-create-a-new-us

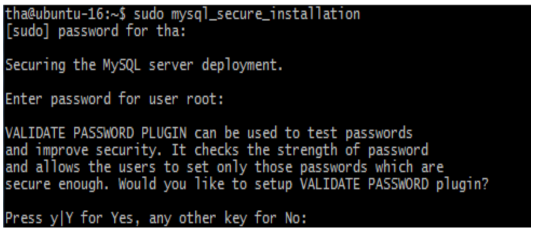
Install MySql

1. In the command line: log into your server with a user account that is not root but has sudo privileges (like you did in the tutorial: getting started with Digital Ocean). Then
2. `sudo apt-get update`
3. `sudo apt-get install mysql-server`



```
tha@ubuntu-16:~$ sudo apt-get update
[sudo] password for tha:
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [94.5 kB]
Hit:2 http://nyc2.mirrors.digitalocean.com/ubuntu xenial InRelease
Get:3 http://nyc2.mirrors.digitalocean.com/ubuntu xenial-updates InRelease [95.7 kB]
Hit:4 http://nyc2.mirrors.digitalocean.com/ubuntu xenial-backports InRelease
Hit:5 https://apt.dockerproject.org/repo ubuntu-xenial InRelease
Fetched 190 kB in 1s (146 kB/s)
Reading package lists... Done
tha@ubuntu-16:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libfcgi-perl libhtml-pa
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtime
  mysql-common mysql-server-5.7 mysql-server-core-5.7
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl libfcgi-perl libhtml-pa
  libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libtime
  mysql-common mysql-server mysql-server-5.7 mysql-server-core-5.7
0 upgraded, 20 newly installed, 0 to remove and 15 not upgraded.
Need to get 18.8 MB of archives.
After this operation, 162 MB of additional disk space will be used.
Do you want to continue? [Y/n] |
```

4. Write Y for yes
5. Set the password for the root user (remember this password for later)
6. `sudo mysql_secure_installation`
7. write no to validate password plugin:

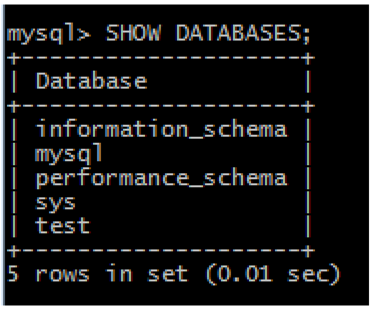


```
tha@ubuntu-16:~$ sudo mysql_secure_installation
[sudo] password for tha:

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:
```

8. For the rest of the questions write yes
9. All done!

Get into MySql console

1. `mysql -u root -p`
2. you now have the mysql> prompt
3. try writing `CREATE DATABASE test;`
4. now try: SHOW DATABASES;

5. 

```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test               |
+--------------------+
5 rows in set (0.01 sec)
```

Create a new user and grant privileges

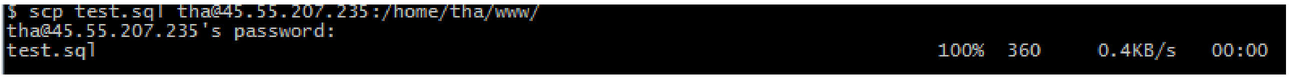
1. CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'password'; (replace the writing in red with your own username and password.

2. GRANT ALL PRIVILEGES ON * . * TO 'newuser'@'localhost'; (This will give the new user read/write privileges on all databases.

3. That´s it: write: `exit` to get back to ubuntu prompt.

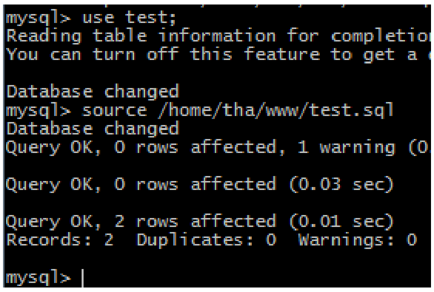
Run script file on the MySql Server

1. Create a script file (E.g. one that can add tables to the test database) and test it locally
2. Use scp to upload the file to Ubuntu (cd into the local folder holding the script/sql file first)



```
$ scp test.sql tha@45.55.207.235:/home/tha/www/
tha@45.55.207.235's password:
test.sql                                                               100%  360     0.4KB/s   00:00
```

3. Log into Ubuntu and run the mysql prompt: `mysql -u root -p`
4. Run: `use test;`
5. Run: source <path to sql file>
6. Check to see if script has run by running a select statement.



```
mysql> use test;
Reading table information for completio
You can turn off this feature to get a 

Database changed
mysql> source /home/tha/www/test.sql
Database changed
Query OK, 0 rows affected, 1 warning (0

Query OK, 0 rows affected (0.03 sec)

Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> |
```

How to connect to your database using MySql Workbench

By default MySQL does not allow remote connections.
To change this you will need to modify the configuration file. This file is to be found in one of two places
"`/etc/mysql/my.cnf`" (if this file doesn't exist or has no content – instead go to)
"`/etc/mysql/mysql.conf.d/mysqld.cnf`"

Find section: [mysqld] and the line that says "bind-address"

```
bind-address   = localhost
```

Comment out that line entirely (use the hash-tag). - Then it will listen on all IPs and ports which you need because you will be connecting remotely to it over public IPv4.

After that add a user to access your database like this:

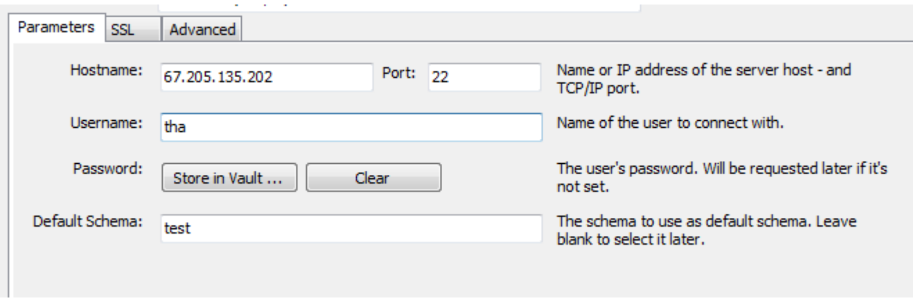
```
mysql> mysql -u root –p
mysql> CREATE USER 'testuser'@'localhost' IDENTIFIED BY 'password123';
mysql> CREATE USER 'testuser'@'%' IDENTIFIED BY 'password123';
mysql> GRANT ALL ON test.* TO 'testuser'@'localhost';
mysql> GRANT ALL ON test.* TO 'testuser'@'%';
```

Replace % with your local IP address of your laptop/desktop or if it is dynamic (like it would be if you are sitting on the school ip) you can add them either by: '192.168.0.%' as a dynamic C-class or just keep the '%' if you want to be able to connect from anywhere (this is less secure).
Also replace the writing in red with your own username and password.

Now restart MySQL:
`exit` (To get back to the Ubuntu prompt)

```
service mysql restart
```

Now on your computer open up MySql Workbench and "Connect to database"



Hostname: 67.205.135.202 Port: 22 — Name or IP address of the server host - and TCP/IP port.

Username: tha — Name of the user to connect with.

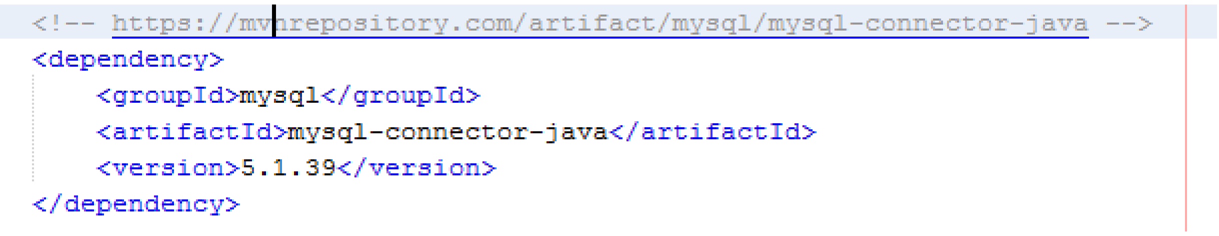
Password: — The user's password. Will be requested later if it's not set.

Default Schema: test — The schema to use as default schema. Leave blank to select it later.

Put your credentials in here and log in.

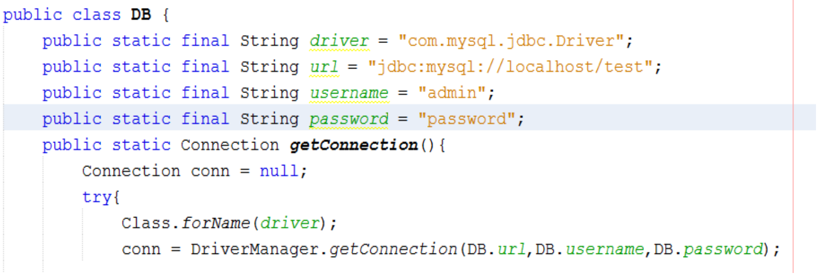
How to connect to your database from your java web application

1. Create a table in you mysql database and put in some entries (data)
2. Create a Maven Web project in Netbeans
3. In the Pom.xml add this depency: (Get it here)



```
<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>5.1.39</version>
</dependency>
```

4. Create a Servlet and read from the test database and print the content to the response output stream.
5. Check the result in a browser.



```
public class DB {
    public static final String driver = "com.mysql.jdbc.Driver";
    public static final String url = "jdbc:mysql://localhost/test";
    public static final String username = "admin";
    public static final String password = "password";
    public static Connection getConnection(){
        Connection conn = null;
        try{
            Class.forName(driver);
            conn = DriverManager.getConnection(DB.url,DB.username,DB.password);
```

Connect with your JPA application