MariaDB installation to Ubuntu Server

Installation and documentation completed by Arttu Pesonen, Geoffrey Thielman and Simon Barry

First, open the Virtual Machine you are using to store the database.

Before installing MariaDB on the machine we need to update the ubuntu system. To do this, use the following command.

```
sudo apt update
```

Once ubuntu is updated, we then install the MariaDB server. Use the following command to install MariaDB.

```
sudo apt install mariadb-server
```

We now need to ensure the installation is secure. To do this we run a built in security script to revoke anonymous accounts and set general security settings. Use the following command to access the security script.

```
sudo mysql_secure_installation
```

Once the script is running you will be prompted with the following questions. We have provided the recommended answers below. In this documentation 'Y' denotes yes and 'n' denotes no.

Set root password: Y

Remove anonymous users: Y

Disallow root login remotely: Y

Remove test database: Y

Reload privilege tables: Y

Next, we need to open the MariaDB prompt. We do this using the following command

```
sudo mariadb
```

We now have to create the admin user with password authentication so the root user does not need to be used. The admin user should be granted all rights. We can do this with the following command.

```
GRANT ALL ON *.* TO 'admin'@'localhost' IDENTIFIED BY 'password' WITH GRANT OPTION;
```

Note: The keywords ‘admin’ and ‘password’ denote the username and password you are giving the user. Replace the keyword ‘password’ with your specified password that you would like to give to the admin user.

To ensure the user rights are granted we must now flush the cache. use the following command to do this.

```
FLUSH PRIVILEGES;
```

We must now check that the database is running. First we must exit the MariaDB prompt and then check the status of the database. Use the following commands.

```
exit;  //Type exit or quit and press enter to leave the prompt
```

```
sudo service mysql status
```

```
sudo systemctl status mariadb
```

Note: The above two statements do the same thing, so choose one depending on preference.

Once we know that the database is up and running, we can check if the admin user we have created is working. To log in with these details use the following command.

```
mysqladmin -u admin -p version
```

Note: ‘-u’ denotes the username and ‘-p’ denotes the password. You do not need to add the password in the command line as you will be prompted it when you commit the command. Use the password you have provided the admin account, when prompted to do so, to log in.

Now we are logged in as the admin, we can now create the lowest level user. To do this, use the following command.

```
CREATE USER 'testuser'@'localhost' IDENTIFIED BY 'pass123';
```

Note: By default, this user does not have permission to create databases, use database (permission to use database must be granted), and cannot see users or databases available.

Next, we can create the database for use. The following command creates the database that will be used on the project.

```
CREATE TABLE outerSuite(id INT NOT NULL AUTO_INCREMENT, startTime TIMESTAMP NOT NULL, endTime TIMESTAMP NOT NULL, testtrigger VARCHAR(50) NOT NULL, testType VARCHAR(50) NOT NULL, componentName VARCHAR(50) NOT NULL, totalFail INT NOT NULL, totalPass INT NOT NULL, documentation VARCHAR(50), hasPassed BOOLEAN NOT NULL, PRIMARY KEY(id));
```

Note: We initially wanted to use the keyword ‘trigger’ as a column in the database. However, the keyword ‘trigger’ is a used keyword in MariaDB and therefore cannot be used in the database.

We can now grant permissions to the user. We can do this with the following command.

```
GRANT SELECT ON outerSuite to 'testuser'@'localhost';
```

Note: Granting select on the outerSuite table allows the user to see the database, use it. However, they cannot see other tables, create/drop tables or insert.

If we want the user to have write permissions on the table, we can use the following command.

```
GRANT INSERT ON outerSuite to 'testuser'@'localhost';
```

Note: User can now insert on the outerSuite table.

Finally, We must now allow users to connect remotely. To do this we need to access the configuration file and change the bind-address. First, exit the prompt as before. Then, access the file use the following command.

```
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf
```

Find the bind-address in the file and change the address from **127.0.0.1** and change the address to **0.0.0.0**. This will now allow remote access to the database.