Database Instruction on CentOS

Overview

MariaDB is a drop-in replacement for MySQL in CentOS. It has the same data and table definition files, protocols, APIs, connectors, package and a few other things similar with MySQL. Therefore, we will be using MariaDB for our database since it is highly supported by CentOS unlike MySQL. We want to be notified of any latest update and security patches for our database but this is simply impossible for MySQL in CentOS. In this project, we mainly use the database to store general data type in three tables: Voters, Administrators and Candidates. Hence any existing incompatibilities between MariaDB and MySQL don't really affect our database mitigation to MariaDB. Moreover, MariaDB development is open to public, has transparent security releases along with the list of CVE identifiers, and has better query optimizer that slightly increase the speed performance compared to MySQL. It should be noted that **our database is hosted on the Main Station** as discussed with our supervisor and client due to the convenience of securing one machine instead of many. There are several steps required to build our database in CentOS.

Installation on CentOS

Database Installation

- Install MariaDB in CentOS via terminal.

```
sudo yum install mariadb-server
```

- Enable the service once the installation is complete.

```
sudo systemctl enable mariadb
sudo systemctl start mariadb
```

- Strengthen our database by changing the default option in MariaDB configuration file.

```
sudo mysql_secure_installation
```

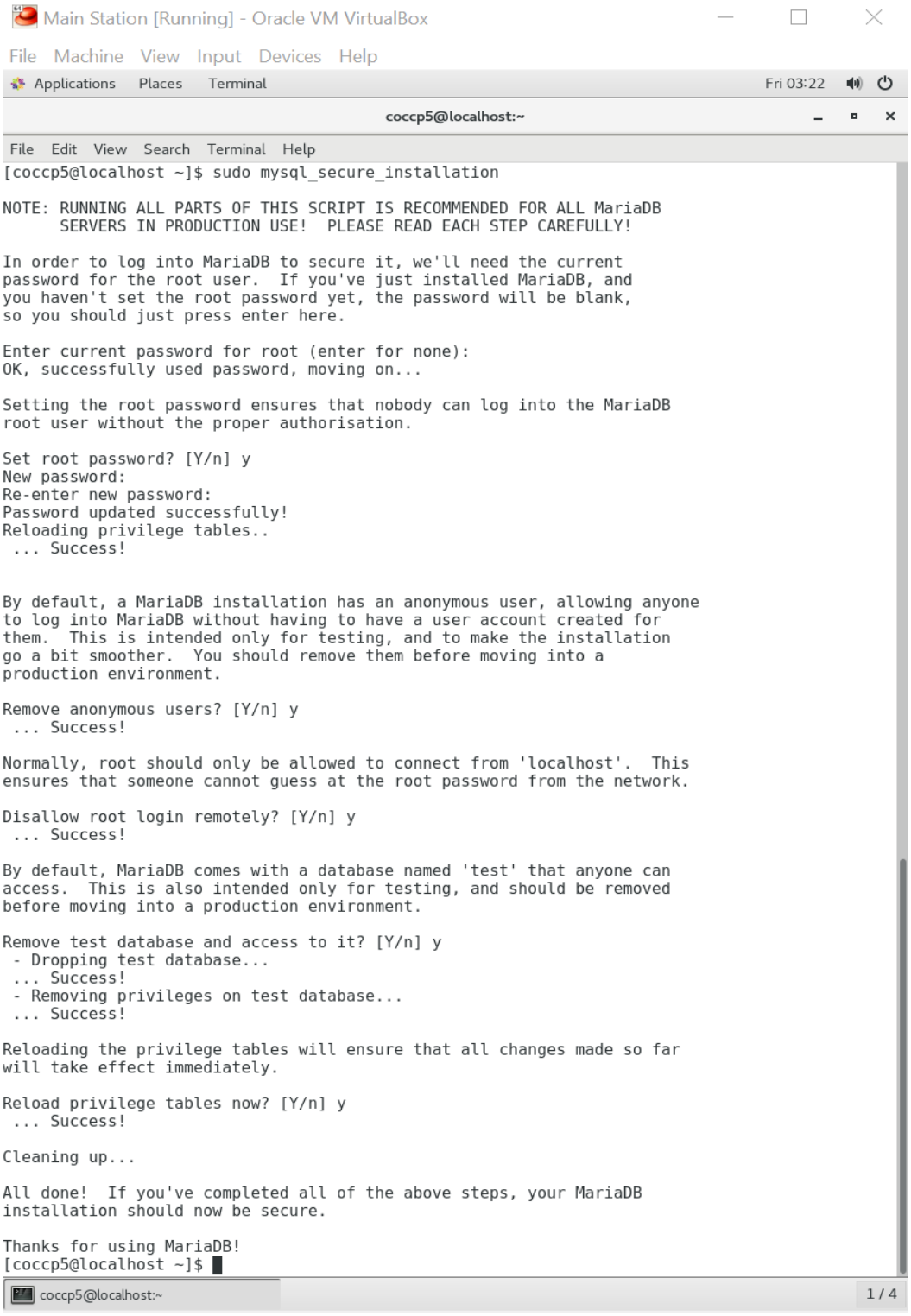

Figure 1: Steps needed for installation

We will be using “blockchain5” as the root password.

- Next step is to change our root name so it won't be too obvious.
    - Login to MariaDB as root

```
mysql -u root -p
```

    - In this case, for development purposes, we will be using "blockchain5" as our root username as well.

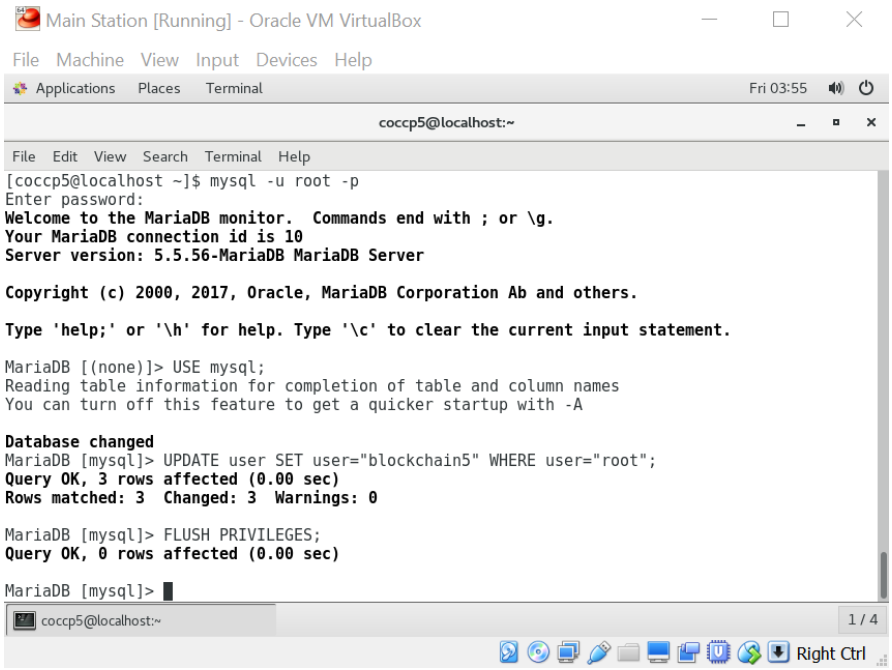

Figure 2: Steps resulting from changing root's name

Setting up database and populating it

- We shall populate the database with the same tables and fields as our previous database. The structure could be viewed in our repository or simply download Xavier's Database VM to see all of the details. The explanation regarding the blockchain address for each candidate can be viewed in Laurence's README.

    - First, create a database called `eVoting`.

```
CREATE DATABASE eVoting;
```

    - Then we shall use `eVoting` database to store our system information.

```
USE eVoting;
```

    - Create table "Administrators" and populate it.

```
CREATE TABLE Administrators(
-> id INT(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
-> name VARCHAR(32) NOT NULL,
-> username VARCHAR(32) NOT NULL,
-> password VARCHAR(32) NOT NULL,
-> loginNum INT NOT NULL DEFAULT 0);
```

Populate it using:

```
INSERT into Administrators(id, name, username, password) VALUES
-> (null, 'Xavier', 'xavier', 'xavier'),
-> (null, 'Wendy', 'wendy', 'wendy'),
-> (null, 'Laurence', 'laurence', 'laurence'),
-> (null, 'Jesse', 'jesse', 'jesse'),
-> (null, 'Mihai', 'mihai', 'mihai'),
-> (null, 'Mark', 'mark', 'mark'),
-> (null, 'Hannes', 'hannes', 'hannes'),
-> (null, 'Dave', 'dave', 'dave'),
-> (null, 'Ling', 'ling', 'ling'),
-> (null, 'Sonny', 'sonny', 'sonny');
```

- Create table `Voters` and populate it.

```
CREATE TABLE Voters(
-> id INT(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
-> name VARCHAR(32) NOT NULL,
-> username VARCHAR(32) NOT NULL,
-> password VARCHAR(32) NOT NULL),
-> votingAddress VARCHAR(64),
-> vote VARCHAR(64),
-> loginNum INT NOT NULL DEFAULT 0);
```

Populate using:

```
INSERT into Voters (id, name, username, password) VALUES
-> (null, 'Xavier', 'xavier', 'blockchain5'),
-> (null, 'Oliver', 'liver', 'oliver'),
-> (null, 'Tiffany', 'tiffany', 'tiffany'),
-> (null, 'Liam', 'liam', 'liam'),
-> (null, 'Glenn', 'glenn', 'glenn'),
-> (null, 'Mckinley', 'mckinley', 'mckinley'),
-> (null, 'Shalon', 'shalon', 'shalon'),
-> (null, 'Dane', 'dane', 'dane'),
-> (null, 'Jolynn', 'jolynn', 'jolynn'),
-> (null, 'Twyla', 'twyla', 'twyla'),
-> (null, 'Meggan', 'meggan', 'meggan'),
-> (null, 'Winfred', 'winfred', 'winfred'),
-> (null, 'Synthia', 'synthia', 'synthia'),
-> (null, 'Landon', 'landon', 'landon'),
-> (null, 'Corinne', 'corinne', 'corinne'),
-> (null, 'Azucena', 'azucena', 'azucena'),
-> (null, 'Letisha', 'letisha', 'letisha'),
-> (null, 'Mardell', 'mardell', 'mardell'),
-> (null, 'Ling', 'ling', 'ling'),
-> (null, 'Domenic', 'domenic', 'domenic'),
-> (null, 'Christia', 'christia', 'christia'),
-> (null, 'Kala', 'kala', 'kala'),
-> (null, 'Johnetta', 'johnetta', 'johnetta'),
-> (null, 'Star', 'star', 'star'),
-> (null, 'Sandee', 'sandee', 'sandee'),
-> (null, 'Mack', 'mack', 'mack'),
-> (null, 'Grazyna', 'grazyna', 'grazyna'),
-> (null, 'Colette', 'colette', 'colette'),
-> (null, 'Liz', 'liz', 'liz'),
```

```
-> (null, 'Alissa', 'alissa', 'alissa'),
-> (null, 'Lachelle', 'lachelle', 'lachelle'),
-> (null, 'Franklin', 'franklin', 'franklin'),
-> (null, 'Darcie', 'darcie', 'darcie'),
-> (null, 'Regan', 'regan', 'regan'),
-> (null, 'Mafalda', 'mafalda', 'mafalda'),
-> (null, 'Mittie', 'mittie', 'mittie'),
-> (null, 'Crysta', 'crysta', 'crysta'),
-> (null, 'Blanche', 'blanche', 'blanche'),
-> (null, 'Herman', 'herman', 'herman'),
-> (null, 'Ruth', 'ruth', 'ruth'),
-> (null, 'Imelda', 'imelda', 'imelda'),
-> (null, 'Lorenza', 'lorenza', 'lorenza'),
-> (null, 'Temika', 'temika', 'temika'),
-> (null, 'Dora', 'dora', 'dora'),
-> (null, 'Angla', 'angla', 'angla'),
-> (null, 'Ahmad', 'ahmad', 'ahmad'),
-> (null, 'Elfreda', 'elfreda', 'elfreda'),
-> (null, 'Melba', 'melba', 'melba'),
-> (null, 'Melia', 'melia', 'melia');
```

- Create table `Candidates` and populate it.

```
CREATE TABLE Candidates(
-> id INT(11) NOT NULL PRIMARY KEY AUTO_INCREMENT,
-> name VARCHAR(32) NOT NULL,
-> blockchainAddress VARHCAR(100));
```

In this case, we will be using the previous blockchain address for consistency purpose in our development and hence all of these details are manually inputted with query.

```
INSERT into Candidates (id, name, blockchainAddress) VALUES
-> (null, 'James', '1UZp45NCVktrJriyKC1WhtH87krEveaTNF33qE'),
-> (null, 'Rick', '1MR4DymXcMp6Cl7Bn7W5Q1E9Aaz8gk4Ks7sjZc'),
-> (null, 'Adam', '1LXUkrowBdWRzvw114NdfYeN4ijPuQMAHtvpkd');
```

- Create a list of procedures for our programs:

```
delimiter $$

create procedure getVoter(IN inName VARCHAR(32)) begin select * from voters
where username=inName; end$$

create procedure getAllCandidates() begin select * from candidates; end$$

create procedure checkAdmin(IN inName VARCHAR(32), IN inPass VARCHAR(32))
begin select * from administrators where username=inName and
password=inPass ; end$$

create procedure checkVoter(IN inName VARCHAR(32), IN inPass VARCHAR(32))
begin select * from voters where username=inName and password=inPass ;
end$$
```

```
create procedure saveVoteStatus(IN inStatus VARCHAR(64), IN inName
VARCHAR(32)) begin update voters set vote=inStatus where username=inName;
end$$

create procedure saveVotingAddress(IN inAddress VARCHAR(64), IN inName
VARCHAR(32)) begin update voters set votingAddress=inAddress where
username=inName; end$$

create procedure loginVoter(IN inLogin CHAR(1), IN inName VARCHAR(32))
begin update voters set login=inLogin where username=inName; end$$

create procedure loginAdmin(IN inLogin CHAR(1), IN inName VARCHAR(32))
begin update administrators set login=inLogin where username=inName; end$$

create procedure incrementVoterLogin(IN inName VARCHAR(32), IN increment
INT) begin update voters set loginNum=increment where username=inName;
end$$

create procedure incrementAdminLogin(IN inName VARCHAR(32), IN increment
INT) begin update administrators set loginNum=increment where
username=inName; end$$

create procedure getAdmin(IN inName VARCHAR(32)) begin select * from
administrators where username=inName; end$$

delimiter ;
```