

Prerequisite:

- The Linux server should be connected to the internet to be able to download and install required packages.
- The installation of packages is done with the **\$sudo** (**\$sudo**) or **root** (**#**) privilege (In thus guide, I used the **\$sudo** command to install and configure.
- Though the Linux Ubuntu was used to produce this guide, the same procedure is applied to other type of Linux Operating Systems.
- MySQL is almost similar to Mariadb server, the latter is a forked of MySQL.

Case 1: Mariadb - Server

1. Check if mariadb package exist and its status:

\$sudo systemctl status mariadb

2. Download the latest package information from the configured repositories and update the package index files:

\$sudo apt update

3. Install mariadb-server by running:

\$sudo apt install mariadb-server

Mariadb-server should now be installed and running. To check if mysql is now up and running, use the command:

4. Check mariadb status after intallation:

\$sudo systemctl status mariadb

If the system is up and running, you will see a screen similar to the one below:

5. Secure mariadb (create the password for mariadb root user) by using the following command and instructions below:

\$sudo mysql secure installation

- This will ask you for the Mariadb **root** password that you set during installation. Type it in and press ENTER. Now you'll answer a series of **yes** or **no** prompts. Let's go through *them*:

- Next you'll be asked if you want to change the **root** password. Since we just created
 the password when we installed Mariadb, we can safely skip this. Hit **ENTER** to
 continue without updating the password.
- The rest of the prompts can be answered **yes**. You will be asked about removing the **anonymous** Mariadb user, disallowing remote **root** login, removing the **test** database, and reloading privilege tables to ensure the previous changes take effect properly. These are all a good idea. Type **Y** and hit **ENTER** for each.

6. Open mariadb connection:

```
$sudo mysql -u root -p
```

7. Run the climsoft database script in mariadb environment to create both **Climsoft operation** and **test database** as follow:

mariadb> source /path (e.g.: source /home/wmo/Dowloads/mariadb climsoft db v4 all)

8. Grant Global privilege to the user root

Grant the global privileges (%) to the user to allow this user to be able to login remotely from any Client PC (Windows or Linux OS) and to read and write in the database as follow; In MySQL environment, type the 2 commands in bold:

```
Mariadb> GRANT ALL PRIVILEGES ON *. * TO 'root'@'%' IDENTIFIED BY 'new-password';
Mariadb> GRANT ALL PRIVILEGES ON *. * TO 'root'@'%' WITH GRANT OPTION;
Mariadb> FLUSH PRIVILEGES.
```

9. Restart mariadb service to apply the change

\$sudo systemctl restart mariadb

10. Allow Remote Access Login into the Linux Server

To enable Remote access users from clients PC (running windows and having Climsoft installed), follow the step below to edit the Linux server network configuration file (see below example for the case of Ubuntu). Change the network configuration file as follow: Login with root or \$sudo privileges:

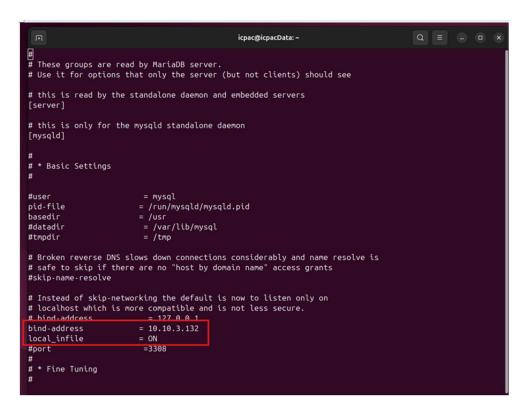
\$sudo Vi /etc/mysql/mariadb.conf.d/50-server.cnf (and edit the file as follow: Comment the default IP address and add the static IP address of the server).

E.g.: \$sudo bind-address = 127.0.0.1 and add a line of bind-address to the server IP address. i.e.: bind-address = 10.10.3.132

Local_infile = ON

Port=3306

Below is an example of the configuration file screenshot:



- Save and quit editing mode using the vi Editor command to save and quit: wq! and restart mariadb service using the command: \$sudo service mariadb restart

11. Test Connection via any of the database admin tool

- Connect from the Client PC using any Database Administration tool (in my case, I have used HeidiSQL) to see if you can access remotely the server.
- Login in Climsoft on the Client side by setting database connection to the server IP address, database and port number to test if the connection is well established.

Case 2: mysql - Server

1. Check if mysql exist and its status:

\$sudo systemctl status mysql

2. Download the latest package information from the configured repositories and update the package index files:

\$sudo apt update

3. Install MySQL by running:

sudoapt install mysql-server

MySQL should now be installed and running. To check if mysql is now up and running, use the command:

4. Check mysql status after intallation:

\$sudo systemctl status mysql

If the system is up and running, you will see a screen similar to the one below:

5. Secure mysql (create the password for mysql root user) by using the following command and instructions below. Secure your MySQL installation to protect your data and ensure the database is not vulnerable to unauthorized access. \$sudo mysql secure installation

This will ask you for the MySQL **root** password that you set during installation. Type it in and press ENTER. Now you'll answer a series of **yes** or **no** prompts. Let's go through *them:*

- Next, you'll be asked if you want to change the **root** password. Since we just created
 the password when we installed MySQL, we can safely skip this. Hit **ENTER** to
 continue without updating the password.
- The rest of the prompts can be answered **yes**. You will be asked about removing the **anonymous** MySQL user, disallowing remote **root** login, removing the **test** database, and reloading privilege tables to ensure the previous changes take effect properly. These are all a good idea. Type **Y** and hit **ENTER** for each.

6. Open mysql connection:

\$sudo mysql -u root -p

7. Run the climsoft database script in mysql environement to create both Climsoft operation and test database as follow:

mysql> source /path (e.g.: source /home/wmo/Dowloads/mariadb_climsoft_db_v4_all)

8. Grant the Global privilege to the user with grant option

Grant the global privileges (%) to the user to allow this user to be able to login remotely from any Client PC (Windows or Linux OS) and have full access to the database (read and write) as follow; In MySQL environment, type the 2 commands in bold:

```
Mysql> GRANT ALL PRIVILEGES ON *. * TO 'root'@'%' IDENTIFIED BY 'new-password';
Mysql> GRANT ALL PRIVILEGES ON *. * TO 'root'@'%' WITH GRANT OPTION;
Mysql> FLUSH PRIVILEGES.
```

9. Restart mysql service to apply the change

\$sudo systemctl restart mysql

10. Allow Remote Access Login into the Linux Server

To enable Remote access users from clients PC (running windows and having Climsoft installed), follow the step below to edit the Linux server network configuration file (see below example for the case of Ubuntu). The Network configuration file (msqld) path to edit with \$sudo or root privilege.

\$sudo Vi /etc/mysql/mysql.conf.d/mysqld.cnf

11. Comment the default bind-address = **127.0.0.1** in the configuration file shown in the path below (i.e. \$sudo bind-address = **127.0.0.1** and add a line of bind-address to the server IP address.by default the port number is 3306

For my case.: **bind-address = 10.10.3.132.**

local_infile = ON (ensure you add this line in the configuration file)

Below is an example of the configuration file screenshot:

Save and quit editing mode using the vi Editor command to save and quit: wq! and restart mysql service using the command: \$sudo service mysql restart

12. Test Connection via any of the database administration tool

- Connect from the Client PC using any Database Administration tool (in my case, I have used HeidiSQL) to see if you can access remotely the server.
- Login in Climsoft on the Client side by setting database connection to the **server IP address**, **database** and **port number** to test if the connection is well established.

Congratulations!

For any question or further clarifications, contact the Climsoft Helpdesk at: support@climsoft.org

ANNEX: Important Mariadb and Linux Commands

How to check users and permissions locally(localhost) and remote:

mariadb> SELECT User, Host FROM mysql.user;

How to check port number IN MARIADB/MYSQL:

mariadb> SHOW GLOBAL VARIABLES LIKE 'PORT';

How to check listening port in ubuntu:

\$sudo ss -tlnp |grep 3308

Always restart the service after making any changes (e,g port change) on mariadab by running the command below:

\$sudo service mysql restart or

\$sudo systemctl restart mysql

How to remove MySQL or MariaDB package from your Ubuntu system using the terminal

A) Case of MySQL

Stop the MySQL service using:

\$sudo systemctl stop mysql

• Purge MySQL packages using:

\$sudo apt-get purge mysql-server mysql-client mysql-common mysql-server-core-* mysql-client-core-*

Remove MySQL configuration and data directories:

\$sudo rm -rf /etc/mysql /var/lib/mysql

Remove unnecessary packages:

\$sudo apt-get autoremove

Clean up the apt cache:

\$sudo apt-get autoclean

Delete the MySQL user and group:

\$sudo deluser --remove-home mysql \$sudo delgroup mysql

• Check for Remaining Files (Optional):

dpkg -l | grep mysql

· check the status

\$sudo systemctl status mysql

B) Case of MariaDB

• Stop the MySQL service using:

\$sudo systemctl stop mariadb

Uninstall MariaDB Packages

\$sudo apt-get remove --purge mariadb-server mariadb-client mariadb-common

• Remove Configuration Files

\$sudo apt-get purge mariadb-server-*

• Delete Data Directories (Optional)

\$sudo rm -r /var/lib/mysql \$sudo rm -r /etc/mysql

Clean Up Residual Files

\$sudo apt-get autoremove \$sudo apt-get autoclean

Verify Removal Check for any remaining MariaDB-related files:

dpkg -l | grep mariadb

· check the status

\$sudo systemctl status mariadb

How to create a directory with mkdir in Linux Ubuntu:

\$sudo mkdir /var/DirectoryName

How to give all permissions to the folder with chmod -R 777:

\$sudo chmod -R 777 /var/DirectoryName