Referral Link:- <https://www.tecmint.com/install-mariadb-in-centos-7/>

How to Install and Secure MariaDB 10 in CentOS 7

**Step1:**  Start by adding the **MariaDB YUM** repository file **MariaDB.repo** for RHEL/CentOS and Fedora systems

#vim /etc/yum.repos.d/MariaDB.repo

[mariadb]

name = MariaDB

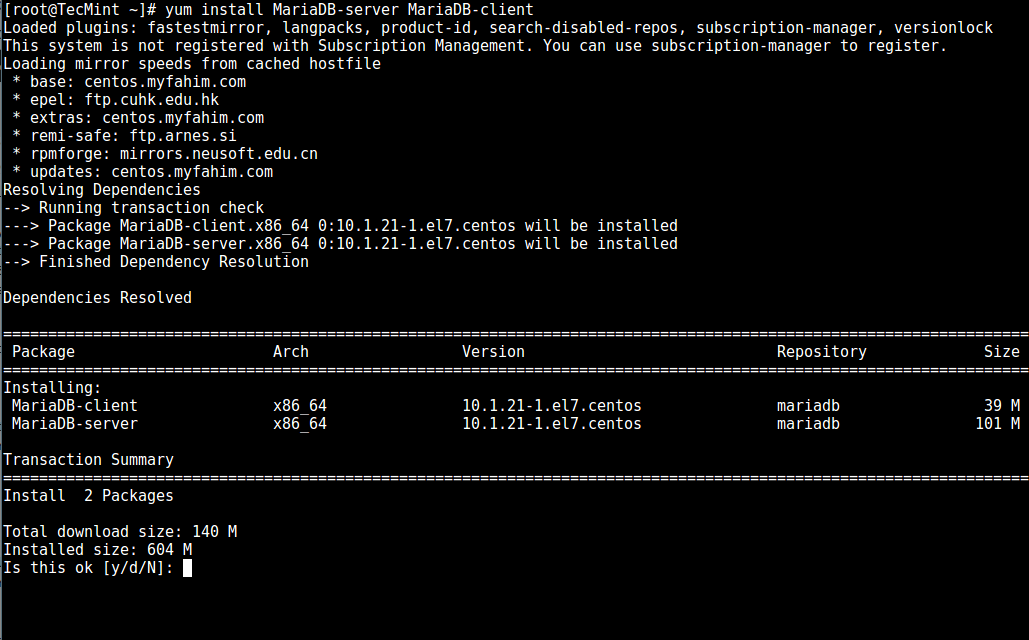
baseurl = http://yum.mariadb.org/10.1/centos7-amd64

gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB

gpgcheck=1

### Step 2: Install MariaDB in CentOS 7

#yum install MariaDB-server MariaDB-client –y



As soon as the installation of MariaDB packages completes, start the database server daemon for the time being, and also enable it to start automatically at the next boot like so:

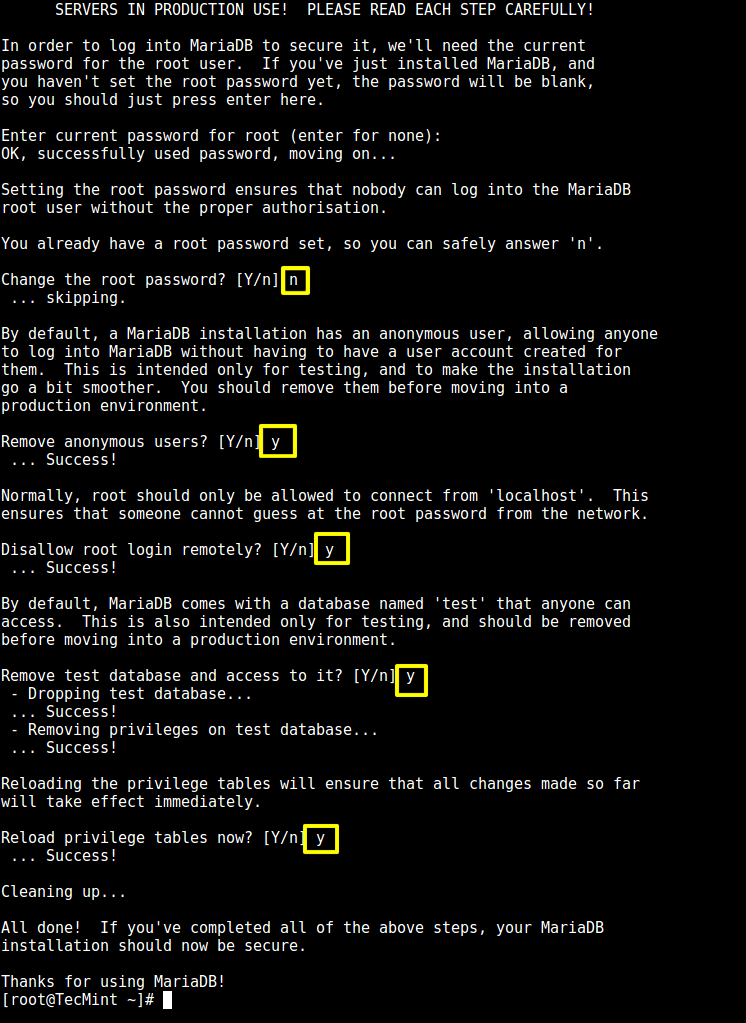
# systemctl start mariadb

# systemctl enable mariadb

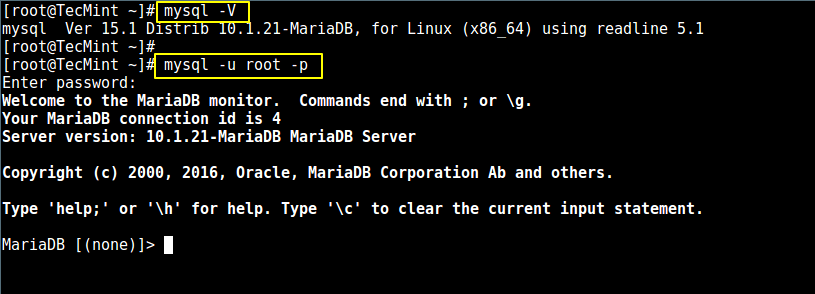
# systemctl status mariadb

### Step 3: Secure MariaDB in CentOS 7

# mysql\_secure\_installation



 After securing the database server, you may want to check certain MariaDB features such as: installed version, default program argument list, and also login to the MariaDB command shell as follows:



===================================================

How to Install and Secure MySQL8.0 in CentOS 7

1.Now download and add the following **MySQL Yum repository** to your respective Linux distribution system’s repository list to install the latest version of MySQL

# wget <https://repo.mysql.com/mysql80-community-release-el7-1.noarch.rpm>

2.After downloading the package for your Linux platform, now install the downloaded package with the following command.

# yum localinstall mysql80-community-release-el7-1.noarch.rpm

3.You can verify that the MySQL Yum repository has been added successfully by using following command.

# yum repolist enabled | grep "mysql.\*-community.\*"

4.Installing Latest MySQL Version

# yum install mysql-community-server

5. Starting the MySQL Server

#systemctl start mysqld

6.Securing the MySQL Installation

#grep 'temporary password' /var/log/mysqld.log

# mysql\_secure\_installation

* Note: Enter new Root password means your temporary password from file /var/log/mysqld.log.
* Now follow the onscreen instructions carefully, for reference see the output of the above command below.

Securing the MySQL server deployment.

Enter password for user root: **Enter New Root Password**

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: **y**

There are three levels of password validation policy:

LOW Length >= 8

MEDIUM Length >= 8, numeric, mixed case, and special characters

STRONG Length >= 8, numeric, mixed case, special characters and dictionary file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: **2**

Using existing password for root.

Estimated strength of the password: 50

Change the password for root ? ((Press y|Y for Yes, any other key for No) : **y**

New password: **Set New MySQL Password**

Re-enter new password: **Re-enter New MySQL Password**

Estimated strength of the password: 100

Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : **y**

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) : **y**

Success.

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) : **y**

- Dropping test database...

Success.

- Removing privileges on test database...

Success.

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

### Connecting to MySQL Server

**10.** Connecting to newly installed MySQL server by providing username and password.

# mysql -u root -p

#### Sample Output:

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 19

Server version: **8.0.1** MySQL Community Server (GPL)

Copyright (c) 2000, 2015, 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>

======================================================================================================

**MySQL CREATE DATABASE:-**

**Syntax:-** create database DATABASE\_NAME;

**Example:-**

MariaDB [(none)]>create database Sai;

# MySQL SELECT DATABASE:-

**Syntax:-** use DATABASE\_NAME;

**Example:-**

MariaDB [(none)]>use Sai;

# MySQL CREATE TABLE:-

# Syntax:- CREATE TABLE table\_name (column\_name column\_type...);

**Example:-**

MariaDB [Sai]>CREATE TABLE Mahi(ID int, Name varchar(20), Class varchar(20));

# MySQL INSERT Statement:-

# MySQL INSERT statement is used to insert data in MySQL table within the database. We can insert single or multiple records using a single query in MySQL.

**Syntax:-** INSERT INTO table\_name ( field1, field2,...fieldN ) VALUES ( value1, value2,...valueN );

**Example:-**

MariaDB [Sai]>INSERT INTO Mahi(ID, Name, Class) VALUES (101, ‘Prabha’,’XClass’);

# MySQL SELECT Statement:-

# Syntax for specified fields:

# Syntax:- SELECT expressions FROM tables [WHERE conditions];

# Example:-

# MariaDB [Sai]>select ID from Mahi;

# +------+

# | ID |

# +------+

# | 101 |

Syntax for all fields:

**Syntax:-** SELECT \* FROM table\_name;

**Example:-**

MariaDB [Sai]>select \* from Mahi;

+------+--------+--------+

| ID | Name | Class |

+------+--------+--------+

| 101 | Prabha | XClass |

# MySQL ALTER TABLE:-

# MySQL ALTER statement is used when you want to change the name of your table or any table field. It is also used to add or delete an existing column in a table.

## ADD a column in the table:-

**Syntax:-** ALTER TABLE table\_name ADD new\_column\_name column\_definition;

**Example:-**

MariaDB [Sai]>ALTER TABLE Mahi ADD Address varchar(20);

## ADD a Multiple column in the table:-

# Syntax:- ALTER TABLE table\_name ADD new\_column\_name1 column\_definition, new\_column\_name1 column\_definition;

**Example:-**

MariaDB [Sai]>ALTER TABLE Mahi ADD (Gmail varchar(20), Marks int);

Syntax for all fields:

**Syntax:-** SELECT \* FROM table\_name;

Example:-

MariaDB [Sai]>SELECT \* FROM Mahi;

+------+--------+--------+---------+-------+-------+

| ID | Name | Class | Address | Gmail | Marks |

+------+--------+--------+---------+-------+-------+

| 101 | Prabha | XClass | NULL | NULL | NULL |

## MODIFY column in the table

**Syntax:-** ALTER TABLE table\_name MODIFY column\_name column\_definition

**Example:-**

MariaDB [Sai]>ALTER TABLE Mahi MODIFY Name varchar(10);

## DROP column in the table

# Syntax:- ALTER TABLE table\_name DROP column column\_name;

# Example:- ALTER TABLE Mahi DROP column Gmail;

# Syntax for all fields:

## MariaDB [Sai]> select \* from Mahi;

## +------+--------+--------+---------+-------+

## | ID | Name | Class | Address | Marks |

## +------+--------+--------+---------+-------+

## | 101 | Prabha | XClass | NULL | NULL |

## RENAME column in the table

**Syntax:-** ALTER TABLE table\_name CHANGE COLUMN old\_name new\_name column\_definition;

**Example:-**

MariaDB [Sai]> ALTER TABLE Mahi CHANGE column ID REGID int;

Syntax for all fields:

## MariaDB [Sai]> select \* from Mahi;

+-------+--------+--------+---------+-------+

| REGID | Name | Class | Address | Marks |

+-------+--------+--------+---------+-------+

| 101 | Prabha | XClass | NULL | NULL |

RENAME TABLE:-

**Syntax:-** ALTER TABLE table\_name RENAME TO new\_table\_name;

**Example:-**

MariaDB [Sai]> ALTER TABLE Mahi RENAME TO Prabha;

# MySQL UPDATE Statement:-

MySQL UPDATE statement is used to update data of the MySQL table within the database. It is used when you need to modify the table.

**Syntax:-** UPDATE table\_name SET field1=new-value1, field2=new-value2 [WHERE Clause]

**Example:-** UPDATE Prabha SET Gmail=’abhi@gmail.com’,Marks=450;

========================================================================================================================

**How to Recover DB root user Password**

Refreal Link :- <https://stackoverflow.com/questions/33374314/can-not-login-to-mysql-5-7-9-after-change-password>

1. Stop mysql:

systemctl stop mysqld

2. Set the mySQL environment option

systemctl set-environment MYSQLD\_OPTS="--skip-grant-tables"

3. Start mysql usig the options you just set

systemctl start mysqld

4. Login as root

mysql -u root

5. Update the root user password with these mysql commands

mysql> UPDATE mysql.user SET authentication\_string = PASSWORD('MyNewPassword')

-> WHERE User = 'root' AND Host = 'localhost';

mysql> FLUSH PRIVILEGES;

mysql> quit

6. Stop mysql

systemctl stop mysqld

7. Unset the mySQL envitroment option so it starts normally next time

systemctl unset-environment MYSQLD\_OPTS

8. Start mysql normally:

systemctl start mysqld

Try to login using your new password:

7. mysql -u root –p

========================================================================================================================

**MySQL Master-Slaver replication With Single Database**

**Master :-**

# vi /etc/mysql/my.cnf

server-id = 1

binlog-do-db=**tecmint**

log-bin = /var/lib/mysql/mysql-bin

#systemctl restart mysqld

# mysql -u root -p

mysql> GRANT REPLICATION SLAVE ON \*.\* TO '**slave\_user**'@'%' IDENTIFIED BY '**Prabha@123**';

mysql> FLUSH PRIVILEGES;

mysql> FLUSH TABLES WITH READ LOCK;

mysql> SHOW MASTER STATUS;

#  mysqldump -u root -p --all-databases --master-data > /root/dbdump.db

# mysql -u root -p

mysql> UNLOCK TABLES;

#scp /root/dbdump.db root@**(SlaveIP)**:/root/

**Slave :-**

#vim /etc/mysql/my.cnf

server-id = 2

log\_bin=/var/lib/mysql/mysql-bin

replicate-do-db=tecmint

#systemctl restart mysqld

# mysql -u root -p < /root/dbdump.db

# mysql -u root -p

mysql> slave stop;

mysql> CHANGE MASTER TO MASTER\_HOST='**192.168.1.1(MasterIP)**', MASTER\_USER='**slave\_user**', MASTER\_PASSWORD='**Prabha@123**', MASTER\_LOG\_FILE='**mysql-bin.000003**', MASTER\_LOG\_POS=’**11128001’**;

mysql> slave start;

mysql> show slave status\G;

**Master :-**

mysql> create database tecmint;

mysql> use tecmint;

mysql> CREATE TABLE employee (c int);

mysql> INSERT INTO employee (c) VALUES (1);

mysql> SELECT \* FROM employee;

**Slaver :-**

mysql> use tecmint;

mysql> SELECT \* FROM employee;

========================================================================================================================

**Master \_ Slave Replication With Multiple databases**

**Master Configuration**  
  
#systemctl stop mysqld  
  
#vim /etc/my.cnf  
[mysqld]  
log-bin=mysql-bin  
server-id=1  
innodb\_flush\_log\_at\_trx\_commit=1  
sync\_binlog=1  
  
#systemctl restart mysqld  
  
#mysql –u root -p  
  
#GRANT REPLICATION SLAVE ON \*.\* to ‘slave\_user’@‘%’ identified by ‘Root@123’;  
  
#FLUSH PRIVILEGES;  
  
#FLUSH TABLES WITH READ LOCK;  
  
  
  
mysql> SHOW MASTER STATUS;  
+------------------+----------+--------------+------------------+-------------------+  
| File             | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_Set |  
+------------------+----------+--------------+------------------+-------------------+  
| mysql-bin.000001 |      996 |              |                  |                   |  
+------------------+----------+--------------+------------------+-------------------+  
1 row in set (0.00 sec)  
  
  
# mysqldump -u root -p --all-databases --master-data > /root/dbdump.db  
  
#mysql –u root -p  
  
UNLOCK TABLES;  
  
quit;  
  
**Slave configuration**

#vim /etc/my.cnf  
  
[mysqld]  
server-id=2  
  
#systemctl restart mysqld

#mysql –u root -p  
  
mysql>CHANGE MASTER TO MASTER\_HOST=‘192.168.2.210’, MASTER\_USER=‘slave\_user’, MASTER\_PASSWORD=‘Root@123’, MASTER\_LOG\_FILE='mysql-bin.000001’, MASTER\_LOG\_POS=996;  
  
  
# mysql -u root -p < /root/dbdump.db

# mysql -u root -p

mysql> start slave;  
  
mysql> show slave status\G;

========================================================================================================================

***REPLICATION MASTER\_SLAVE-1\_SLAVE-2 :-***

***MASTER CONFIGURATION*:-**

#vim /etc/my.cnf

[mysqld]  
log-bin=mysql-bin  
server-id=1  
innodb\_flush\_log\_at\_trx\_commit=1  
sync\_binlog=1

#systemctl restart mysqld

#mysql –u root –p

Mysql>GRANT REPLICATION SLAVE ON \*.\* to ‘slave\_user’@‘%’ identified by ‘Root@123’;  
  
Mysql> FLUSH PRIVILEGES;

Mysql> FLUSH TABLES WITH READ LOCK;

Mysql> SHOW MASTER STATUS;

+------------------+----------+--------------+------------------+---------------                                             ----+

| File             | Position | Binlog\_Do\_DB | Binlog\_Ignore\_DB | Executed\_Gtid\_                                             Set |

+------------------+----------+--------------+------------------+---------------                                             ----+

| mysql-bin.000010 |     1055 | |              |                          |

Mysql>exit

# mysqldump -u root -p itblabs> /root/dbdump.db

#mysql -u root -p

Mysql>unlock tables;

Mysql>exit

#scp -r /root/dbdump.db root@<slaveip>:/root/

***SLAVE-1 CONFIGURATION:-***

#vim /etc/my.cnf

[mysqld]

Server-id=2

#systemctl restart mysqld

#mysql –u root –p

Mysql>stop slave;

Mysql> CHANGE MASTER TO MASTER\_HOST='192.168.2.206', MASTER\_USER='slave\_user', MASTER\_PASSWORD='Root@123', MASTER\_LOG\_FILE='mysql-bin.000010', MASTER\_LOG\_POS=1055;

Mysql>exit

#mysql -u root -p < /root/dbdump.db

#mysql -u root -p

Mysql>start slave;

Mysql>show slave status\G;

***SLAVE-2 CONFIGURATION:-***

#vim /etc/my.cnf

[mysqld]  
server-id=3

replicate-do-db=itblabs

#systemctl restart mysqld

#mysql -u root –p

mysql>stop slave

mysql>CHANGE MASTER TO MASTER\_HOST=‘192.168.2.206’, MASTER\_USER=‘slave\_user’, MASTER\_PASSWORD=‘Root@123’, MASTER\_LOG\_FILE='mysql-bin.000010’, MASTER\_LOG\_POS=1055;

#mysql -u root -p < /root/dbdump.db

#mysql -u root -p

mysql>start slave;

mysql> show slave status\G;

========================================================================================================================