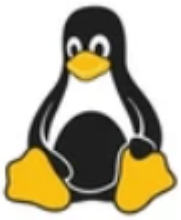
ADVANCE LINUX ADMINISTRATION MARATHON

DAY2

Sumeha Jakate



➢ What is MariaDB?

- **MariaDB** is an open-source, relational database management system (RDBMS) that is a drop-in replacement for **MySQL**.
- It was created by the original developers of MySQL after concerns arose over Oracle's acquisition of MySQL.
- MariaDB aims to stay open-source and offers enhanced features and improvements over MySQL, making it a popular choice for database management.

Key Features of MariaDB:

1. **Compatibility with MySQL**: It is designed to be fully compatible with MySQL, including APIs and commands, so existing MySQL applications can be used with MariaDB without modification.
2. **Open Source**: MariaDB remains open-source, with a strong focus on transparency and community-driven development.
3. **Improved Performance**: MariaDB offers faster performance for certain queries, improved storage engines, and optimized query execution.
4. **Security Enhancements**: Includes advanced security features like data encryption, user account management, and enhanced role-based access control.

➢ MariaDB installation on Linux Machine

Step 1: Update your system

Before installing MariaDB, ensure your system is up to date:

```
yum update
```

Step 2: Install MariaDB

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

```
— 0 bash
root@controlnode:~# yum install mariadb mariadb-server
Updating Subscription Management repositories.
Last metadata expiration check: 0:00:50 ago on Fri 27 Sep 2024 02:28:13 PM UTC.
Package mariadb-3:10.5.22-1.el9_2.x86_64 is already installed.
Package mariadb-server-3:10.5.22-1.el9_2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
root@controlnode:~#
```

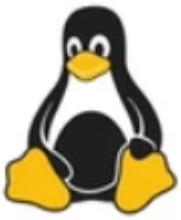
Step 3: Start and Enable the MariaDB Service

After installation, you need to start the MariaDB service and ensure it runs at startup:

```
sudo systemctl start mariadb
```

```
sudo systemctl enable mariadb
```

```
root@controlnode:~# systemctl start mariadb
root@controlnode:~# systemctl enable mariadb
Created symlink /etc/systemd/system/mysql.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/mysqld.service → /usr/lib/systemd/system/mariadb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/mariadb.service → /usr/lib/systemd/system/mariadb.service.
root@controlnode:~#
```

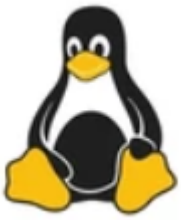


Step 4: Secure MariaDB Installation

For better security, run the MariaDB security script, which helps set the root password and removes unnecessary defaults:

```
sudo mysql_secure_installation
```

- You will be prompted to set the root password.
- You can remove anonymous users, disable remote root login, and clean up test databases.



Step 5: Verify Installation

Check if MariaDB is running correctly:

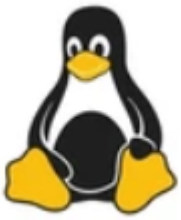
```
sudo systemctl status mariadb
```

```
root@controlnode:~# systemctl status mariadb
● mariadb.service - MariaDB 10.5 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: disabled)
     Active: active (running) since Fri 2024-09-27 14:29:24 UTC; 2min 33s ago
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 3831 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 9 (limit: 48697)
     Memory: 74.0M
        CPU: 410ms
     CGroup: /system.slice/mariadb.service
             └─3831 /usr/libexec/mariadbd --basedir=/usr

Sep 27 14:29:24 controlnode mariadb-prepare-db-dir[3787]: The second is mysql@localhost, it has no password either, but
Sep 27 14:29:24 controlnode mariadb-prepare-db-dir[3787]: you need to be the system 'mysql' user to connect.
Sep 27 14:29:24 controlnode mariadb-prepare-db-dir[3787]: After connecting you can set the password, if you would need to be
Sep 27 14:29:24 controlnode mariadb-prepare-db-dir[3787]: able to connect as any of these users with a password and without sudo
Sep 27 14:29:24 controlnode mariadb-prepare-db-dir[3787]: See the MariaDB Knowledgebase at https://mariadb.com/kb
Sep 27 14:29:24 controlnode mariadb-prepare-db-dir[3787]: Please report any problems at https://mariadb.org/jira
Sep 27 14:29:24 controlnode mariadb-prepare-db-dir[3787]: The latest information about MariaDB is available at https://mariadb.org/.
Sep 27 14:29:24 controlnode mariadb-prepare-db-dir[3787]: Consider joining MariaDB's strong and vibrant community:
Sep 27 14:29:24 controlnode mariadb-prepare-db-dir[3787]: https://mariadb.org/get-involved/
Sep 27 14:29:24 controlnode systemd[1]: Started MariaDB 10.5 database server.
root@controlnode:~#
```

You can also log in to MariaDB to verify the installation:

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

```
root@controlnode:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 10
Server version: 10.5.22-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```



Step 6: Create a database

Create a new database into our MariaDB server

```
create database <name>
```

```
MariaDB [(none)]> create database unnati;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| unnati             |
+--------------------+
4 rows in set (0.000 sec)

MariaDB [(none)]>
```

Step 7: Create table

Use the database we just created

```
use <name>
```

Create a table into our database

```
create table students ( id INT, name varchar(50),email varchar(30),phone varchar(20));
```

```
MariaDB [(none)]> use unnati
Database changed
MariaDB [unnati]> create table customers (id INT, name varchar(50), email varchar(40), number varchar(20));
Query OK, 0 rows affected (0.007 sec)
```

Step 8: Insert values

Insert values into the table you just created.

```
MariaDB [unnati]> insert into customers (id,name,email,number) values (1,"Sumeha", "sumehasjakate@gmail.com", "9999999999");
Query OK, 1 row affected (0.002 sec)

MariaDB [unnati]> select * from customers ;
+------+--------+------------------------+------------+
| id   | name   | email                  | number     |
+------+--------+------------------------+------------+
|    1 | Sumeha | sumehasjakate@gmail.com | 9999999999 |
+------+--------+------------------------+------------+
1 row in set (0.001 sec)

MariaDB [unnati]>
```

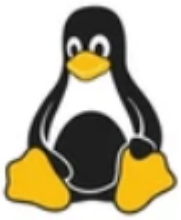
Step 9: Hash password

Get password in the hashed format.

```
MariaDB [(none)]> select password ('mypassword');
+-------------------------------------------+
| password ('mypassword')                   |
+-------------------------------------------+
| *FABE5482D5AADF36D028AC443D117BE1180B9725 |
+-------------------------------------------+
1 row in set (0.000 sec)

MariaDB [(none)]>
```

Step 10: Create a user

```
MariaDB [(none)]> create user 'sumeha'@localhost identified by '*FABE5482D5AADF36D028AC443D117BE1180B9725';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> SELECT User FROM mysql.user;
+------------------+
| User             |
+------------------+
| sumeha@localhost |
| mariadb.sys      |
| mysql            |
| root             |
| sumeha           |
+------------------+
5 rows in set (0.001 sec)

MariaDB [(none)]>
```



➢ MySQL dump

mysqldump is a command-line utility in **MySQL** and **MariaDB** used to create backups of databases. It generates a text file that contains SQL statements to recreate databases, tables, and data. You can use **mysqldump** to back up a single database, multiple databases, or all databases on a MySQL server.

Basic mysqldump Syntax:

```
mysqldump [options] database_name > backup_file.dump
```

Common Use Cases for mysqldump:

1. **Backing Up a Single Database**

   To back up a single database, use the following command:

   ```
   mysqldump -u [username] -p [database_name] > [backup_file.dump]
   ```

- -u [username]: Specifies the MySQL username.
- -p: Prompts for the MySQL password.
- [database_name]: The name of the database to back up.
- [backup_file.dump]: Redirects the output to a .dump file.

2. **Restoring from a MySQL Dump**

   To restore a database from a dump file, use the mysql command:

   ```
   mysql -u [username] -p [database_name] < [backup_file.sql]
   ```

   mysqldump is a versatile tool for database backups and migrations, allowing you to secure your data and easily restore it when needed.