

📄 Installation and Configuration of MariaDB in an LXC Container

Introduction

MariaDB is a powerful, open-source **Relational Database Management System (RDBMS)** that was developed as a free fork of **MySQL**. It originated after Oracle acquired MySQL, aiming to ensure its openness and community-driven development.

MariaDB is used anywhere structured data needs to be stored, processed, and queried—from small web applications to large enterprise systems. Typical use cases include:

- **Web Applications:** Storing user data, content, and configuration data.
- **Business Applications:** Managing orders, customer data, or inventories.
- **Data Analysis & Reporting:** Serving as the foundation for Business Intelligence tools.
- **Cloud and Container Environments:** As a scalable, highly available database solution.

Due to its high compatibility with MySQL, good performance, replication, and clustering features, MariaDB is one of the most popular databases in the open-source world.

i Note on Terminology: Although MariaDB is an independent database management system, terms like `mysql` are often still used in the command line and tools—for example, with commands like `mysql_secure_installation` or the database client `mysql`. The reason for this is the complete compatibility with MySQL: MariaDB intentionally uses the same command names and interfaces so that existing applications, scripts, and documentation can be reused without changes. The appearance of the term "mysql" therefore does not mean that MySQL is installed—it is merely retained naming to ensure compatibility.

📋 Prerequisites

- LXC container with Ubuntu 20.04/22.04/24.04 (tested with Ubuntu 24.04)
- Network access to the container
- Root or `sudo` permissions
- User `pdal` set up with sudo rights

💻 1. Update Package List

```
sudo apt update
```

📦 2. Install MariaDB

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

```
pdal@mariadb120:~$ sudo apt install mariadb-server -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
galera-4 gawk libcgi-fast-perl libcgi-pm-perl libclone-perl libcom32c0
libfcgi-perl libfcgi0t64 libgdbm-compat4t64 libhtml-parser-perl libio-socket-ssl-perl
libio-html-perl liblwp-mediatypes-perl liblzo2-2 libmariadb3 libmemcached4.0
libtimedate-perl liburi-perl liburing2 mariadb-client mariadb-client-core
mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo mariadb-socat
Suggested packages:
gawk-doc libmldb-perl libnet-daemon-perl libsql-statement-perl libxml-libxml-perl
libbusiness-isbn-perl libregexp-ipv6-perl libwww-perl mailx mariadb-client-doc
libtap-harness-archive-perl doc-base
```

Check the installation:

```
mariadb --version
```

```
pdal@mariadb120:~$ mariadb --version
mariadb Ver 15.1 Distrib 10.11.13-MariaDB, for debian-linux-gnu (x86_64) using EditLine wrapper
pdal@mariadb120:~$
```

🔧 3. Start and Enable MariaDB Service

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

```
pdal@mariadb120:~$ sudo systemctl start mariadb
sudo systemctl enable mariadb
Synchronizing state of mariadb.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable mariadb
pdal@mariadb120:~$
```

Check status:

```
sudo systemctl status mariadb
```

```
pdal@mariadb120:~$ systemctl status mariadb
● mariadb.service - Mariadb 10.11.15 database server
    Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
    Active: active (running) since Fri 2025-06-27 09:37:44 UTC; 6min ago
      Docs: man:mariadb(8)
             https://mariadb.com/kb/en/library/systemd/
   Main PID: 7305 (mariadb)
     Status: "Taking your SQL requests now..."
        Tasks: 9 (limit: 28969)
       Memory: 37.9M (peak: 84.5M swap: 43.3M swap peak: 43.3M)
          CPU: 692ms
        CGroup: /system.slice/mariadb.service
                  `--7305 /usr/sbin/mariadb
pdal@mariadb120:~$
```

After starting and enabling the MariaDB service, the database server is now running in the background and is accessible via localhost by default. This means that local applications or commands like `mysql -u root -p` can immediately establish a connection to the database.

In the next steps, the database will first be secured to prevent unauthorized access. The integrated security script is executed for this purpose: `sudo mysql_secure_installation`

This script helps set a root password, remove anonymous users, disable remote root login, and delete the test database. Only after this will the necessary adjustments be made to allow secure **remote access** to the database.

🔒 4. Secure MariaDB

```
sudo mysql_secure_installation
```

Recommended Answers:

switch to unix socket authentication? **N**

change root password? **Y**

Remove anonymous users? **Y**

Disallow root login remotely? **Y** (for production systems)

Remove test database and access to it? **Y**

Reload privilege tables? **Y**

```
pdal@mariadb120:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n
... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB 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? [Y/n] 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? [Y/n] y
... Success!

By default, MariaDB 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? [Y/n] 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.
```

🌐 5. Allow Remote Access (Optional)

Allow external connections, e.g., from a web server container. The listen address needs to be changed for this.

Adjust the configuration file:

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

Find the following line:

```
bind-address = 127.0.0.1

GNU nano 7.2                               /etc/mysql

# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]

#
# * Basic Settings
#


#user          = mysql
pid-file      = /run/mysqld/mysqld.pid
basedir        = /usr
#datadir       = /var/lib/mysql
#tmpdir        = /tmp

# Broken reverse DNS slows down connections considerably and name res
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address      = 127.0.0.1

#
# * Fine Tuning
```

Change it to:

bind-address = 192.168.137.120 (IP address of your own LXC container)

```
# Broken reverse DNS slows down connections considerably and n
# safe to skip if there are no "host by domain name" access gr
#skip-name-resolve

# Instead of skip-networking the default is now to listen only
# localhost which is more compatible and is not less secure.
bind-address      = 192.168.137.120

#
```

Explanation: With this change, the MariaDB server no longer listens only on localhost but also on the specified container IP address (192.168.137.120). This makes the database accessible from the local network, e.g., from another LXC container or an external server. A client can now connect using the following command: `mysql -h 192.168.137.120 -u USERNAME -p` ⚠ Important: For connections

from external clients, the firewall (if active) must also allow access to port 3306/tcp. Furthermore, the database user must have corresponding privileges for % (all hosts) or the specific IP of the client.

Alternatively, you can use the IP address **0.0.0.0**. This tells MariaDB to listen on all ports. This option is recommended if you use a DHCP/DNS server and cannot ensure that the container/server always receives the same IP address.

6. Define Default Character Set and Collation

i Introduction: In a database, the **default character set** (`character set`) and **collation** (`collation`) define how text data is stored, compared, and sorted.

- The **character set** defines which characters can be stored at all (e.g., letters, numbers, special characters, emojis).
- **Collation** determines how these characters are compared, e.g., whether case is considered or which sorting order is used.

A consistent setting for character set and collation is important to avoid problems with queries, comparisons, and data migrations, and to ensure that all applications interpret the data correctly.

For safety, add or adjust the character set and collation parameters in the same configuration file (section `[mysqld]`):

Before:

```
character-set-server = utf8mb4
collation-server = utf8mb4_general_ci
```

Change to:

```
character-set-server = utf8mb4
collation-server = utf8mb4_general_ci
```

```
#  
  
# MySQL/MariaDB default is Latin1, but in Debian we rather default  
# utf8 4-byte character set. See also client.cnf  
character-set-server = utf8mb4  
collation-server = utf8mb4_general_ci  
  
#  
# * InnoDB  
#  
  
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mys  
# Read the manual for more InnoDB related options. There are many!
```

 **Explanation:** `character-set-server = utf8mb4` ensures that the database uses the modern UTF-8 character set by default, which supports all Unicode characters (e.g., emojis or special

characters). For collation, there are two variants:

- `utf8mb4_general_ci` → **case-sensitive**, case is considered during comparison (e.g., "Test" ≠ "test").
- `utf8mb4_general_ci` → **case-insensitive**, case is not considered (e.g., "Test" = "test"). The `utf8mb4_general_ci` setting is commonly used because it is more practical for most use cases like usernames, logins, or search functions.

```
# * Character sets
#
# MySQL/MariaDB default is Latin1, but in Debian we rather default to the full
# utf8 4-byte character set. See also client.cnf
character-set-server = utf8mb4
collation-server     = utf8mb4_general_ci
#
# * InnoDB
#
# InnoDB is enabled by default with a 10MB datafile in /var/lib/mysql/.
# Read the manual for more InnoDB related options. There are many!
# Most important is to give InnoDB 80 % of the system RAM for buffer use:
# https://mariadb.com/kb/en/innodb-system-variables/#innodb_buffer_pool_size
```

Then restart MariaDB:

```
sudo systemctl restart mariadb
```

7. Create MariaDB User for External Access

i Introduction: In this step, a database user is created who can access MariaDB from an external computer or container. This is a **pure database user** who does not necessarily have to exist as a system user (`pdal`) on the server. The name and password can be chosen freely. With the privileges set (`GRANT ALL PRIVILEGES`), the user gains access to all databases and tables and can grant privileges to other users.

Log in to the database:

```
sudo mariadb
```

```
pdal@mariadb120:~$ sudo mariadb
[sudo] password for pdal:
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 52
Server version: 10.11.13-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

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)]>
```

Create user and set privileges:

```
CREATE USER 'pdal'@'192.168.137.%' IDENTIFIED BY 'JadeHS20';
GRANT ALL PRIVILEGES ON *.* TO 'pdal'@'192.168.137.%' WITH GRANT OPTION;
FLUSH PRIVILEGES;
EXIT;
```

💡 Explanation of Commands:

- **CREATE USER 'pdal'@'192.168.137.%' IDENTIFIED BY 'JadeHS20';** Creates a new database user named **pdal** who can connect from all IP addresses in the network range **192.168.137.***, with the password **JadeHS20**.
- **GRANT ALL PRIVILEGES ON *.* TO 'pdal'@'192.168.137.%' WITH GRANT OPTION;** Grants all rights on all databases and tables to this user. With **WITH GRANT OPTION**, the user can also grant privileges to other users.
- **FLUSH PRIVILEGES;** Updates the privileges so that the changes take effect immediately.
- **EXIT;** Disconnects from the MariaDB database and returns to the shell.

```
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.13-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

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)]> CREATE USER 'pdal'@'192.168.137.%'
    -> IDENTIFIED BY 'JadeHS20';
Query OK, 0 rows affected (0.039 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON *.* TO 'pdal'@'192.168.137.%' WITH GRANT OPTION;
Query OK, 0 rows affected (0.039 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> EXIT;
Bye
pdal@mariadb120:~$
```

Note: If access from outside the local network is required (e.g., from the client PC), 'pdal'@'%' would have to be used. This allows any IP address.

🔍 8. Check Port Release (Optional)

The standard port for MariaDB is **3306**. Test with:

```
ss -tulpen | grep 3306
```

The output of the command **ss -tulpen | grep 3306** shows the network connections (or sockets) that are active on port 3306 (standard port for MySQL/MariaDB). You used the options **-tulpen**:

-t: TCP **-u**: UDP **-l**: only listening sockets **-p**: show process information **-e**: extended information (e.g., inode, sk) **-n**: show IPs/Ports instead of DNS names

Explanation of your line (broken down):

```
tcp    LISTEN  0      80      192.168.137.120:3306      0.0.0.0:*      uid:108
ino:27953 sk:2005 cgroup:/system.slice/mariadb.service <->
```

Column / Field	Meaning
tcp	Protocol: TCP
LISTEN	Socket state: listening for incoming connections
0	Recv-Q: Receive Queue (0 = nothing waiting for processing)
80	Send-Q: Send Queue (e.g., waiting packets; almost always >0 for LISTEN)
192.168.137.120:3306	IP address and port (MariaDB is listening here on this interface/port)
0.0.0.0:*	Remote Address/Port: accepts connections from all IPs (all clients)
uid:108	UID of the process (User, e.g., mysql or mariadb)
ino:27953	Inode number of the socket (for internal management in the kernel)
sk:2005	Socket ID (reference to kernel socket, useful for debugging)
cgroup:/system.slice/mariadb.service	CGroup (Systemd service mariadb.service)
<->	Placeholder (irrelevant for LISTEN as no counterpart exists)

Summary:

What this specifically means:

- MariaDB has started successfully
- is listening on Port **3306**
- on all network interfaces, i.e.:
 - 127.0.0.1 (localhost)
 - 192.168.137.120 (your container IP)
 - possibly other local IPs
- Connections from the network (e.g., 192.168.137.0/24) are possible – provided that:
 - the user is correctly set up (in our case, pdal@192.168.137.%)
 - there is no firewall rule blocking access

 Tip: Test Access

From another host in the same network:

```
mysql -u pdal -p -h 192.168.137.120
```

```
apache101 login: pdal
Password:
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.12-11-pve x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro
pdal@apache101:~$ mysql -u pdal -p -h 192.168.137.120
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 53
Server version: 5.5.5-10.11.13-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

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> ■
```

A minimalist Mariadb client is required for this. You can use the [mariadb-client-core](#). To do this, simply enter the following in another container:

```
sudo apt install mariadb-client-core
```

and confirm with [y](#).

If this works: Network access is functional.

If not: Check if a firewall (e.g., ufw or iptables) is blocking the port on the client:

```
sudo ufw status
```

Our applications can now automatically reach MariaDB directly (e.g., direct access from our programs). For manual access, we will set up phpMyAdmin to administer MariaDB in the next document.

Sources

- "MariaDB Documentation | MariaDB Documentation". Accessed: September 25, 2025. [Online]. Available at: [MariaDB Documentation](#)

- "Installing MariaDB Server Guide | MariaDB Documentation". Accessed: September 25, 2025. [Online]. Available at: [MariaDB Server Installation](#)
 - "Connecting to MariaDB Server | MariaDB Documentation". Accessed: September 25, 2025. [Online]. Available at: [Connecting MariaDB Server](#)
 - "Essential Queries Guide | MariaDB Documentation". Accessed: September 25, 2025. [Online]. Available at: [Essential Querries Guide](#)
 - "Server Management | MariaDB Documentation". Accessed: September 25, 2025. [Online]. Available at: [Server Management](#)
 - "Security | MariaDB Documentation". Accessed: September 25, 2025. [Online]. Available at: [Security](#)
-

License

This work is licensed under the **Creative Commons Attribution - ShareAlike 4.0 International License**.

[To the license text on the Creative Commons website](#)