# Databases and Data Stores

## Overview

- [\*\*MariaDB\*\* - Open Source Relational Database](#mariadb)

- [\*\*phpMyAdmin\*\* - SQL Administration Tool for MariaDB](#phpmyadmin)

- [\*\*SQLite\*\* - Small, Fast and High reliable SQL database engine](#sqlite)

- [\*\*Redis\*\* - Open Source In-memory key–value Data Store](#redis)

- [\*\*InfluxDB\*\* - Open Source Time Series Database](#influxdb)

- [\*\*PostgreSQL\*\* - Persistent and advanced SQL database engine](#postgresql)

??? info "How do I run \*\*DietPi-Software\*\* and install \*\*optimised software\*\*?"

To install any of the \*\*DietPi optimised software\*\* listed below run from the command line:

```sh

dietpi-software

```

Choose \*\*Software Optimised\*\* and select one or more items. Finally click on `Install`. DietPi will do all the necessary steps to install and start these software items.

![DietPi-Software menu screenshot](../assets/images/dietpi-software.jpg){: width="643" height="365" loading="lazy"}

To see all the DietPi configurations options, review the [DietPi Tools](../../dietpi\_tools/) section.

[Return to the \*\*Optimised Software list\*\*](../../software/)

## MariaDB

\*\*MariaDB\*\* Server is one of the most popular open source relational databases. It’s made by the original developers of MySQL and guaranteed to stay open source[^1]. It is part of most cloud offerings and the default in most Linux distributions.

![MariaDB logo](../assets/images/dietpi-software-webstack-mariadb.png){: width="200" height="61" loading="lazy"}

Source: [MariaDB](https://mariadb.com/), [LGPL](https://commons.wikimedia.org/w/index.php?curid=55946550).

=== "Quick access"

As `root` user, run `mariadb` from command line, no separate authentication required. But note that this won’t work via sudo, but an interactive root user shell session is required.

- Username = `root`

- Password = The same as your root login password, default is `dietpi`

\*\*\*

Official documentation: <https://mariadb.org>

Getting started documentation: <https://mariadb.org/documentation/#getting-started>

### phpMyAdmin

![phpMyAdmin logo](../assets/images/dietpi-software-webstack-phpmyadmin.png){: width="160" height="120" loading="lazy"}

\*\*phpMyAdmin\*\* is a free software tool written in [PHP](#php), intended to handle the administration of MySQL / MariaDB over the Web.

Frequently used operations (\_such as: managing databases, tables, columns, relations, indexes etc.\_) can be performed via the web user interface. Using the same application you could also directly execute any SQL statement.

=== "Quick access"

- URL = `http://<your.IP>/phpmyadmin`

- Username = `phpmyadmin`

- Password = The same as your root login password, default is `dietpi`

\*\*\*

Website: <https://www.phpmyadmin.net>

Official documentation: <https://www.phpmyadmin.net/docs/>

## SQLite

![SQLite logo](../assets/images/dietpi-software-webstack-sqlite.svg){: width="200" height="95" loading="lazy"}

Source: Part of the SQLite documentation, which has been released by author D. Richard Hipp to the public domain. SVG conversion by Mike Toews. [Public Domain](https://commons.wikimedia.org/w/index.php?curid=11675072)

\*\*SQLite\*\* is an embedded relational database engine. It it a self-contained, high-reliability and full-featured SQL database engine. It is very popular and there are hundreds of millions copies worldwide in use today[^2].

=== "Quick start"

To create a database and run commands, use the [quick start documentation](https://www.sqlite.org/quickstart.html).

\*\*\*

Website: <https://www.sqlite.org/index.html>

Official documentation: <https://www.sqlite.org/docs.html>

## Redis

A non-SQL based data store.

![Redis logo](../assets/images/dietpi-software-webstack-redis.svg){: width="200" height="67" loading="lazy"}

Source: [Carlos Prioglio](https://redis.io/images/redis-logo.svg), [licence](https://commons.wikimedia.org/w/index.php?curid=95020509).

\*\*Redis\*\* is an open source (BSD licensed), in-memory data structure store, used as a database, cache and message broker.

\*\*Redis\*\* is in the family of databases called key-value stores. The essence of a key-value store is the ability to store some data, called a value, inside a key. This data can later be retrieved only if we know the exact key used to store it.

=== "Quick start"

The first thing to do in order to check Redis is working properly is sending a PING command:

```sh

redis-cli ping

```

For more commands and an introduction to Redis data types and commands, read the [quick start documentation](https://redis.io/topics/data-types-intro).

\*\*\*

Website: <https://redis.io/>

Official documentation: <https://redis.io/documentation>

Commands: <https://redis.io/commands>

## InfluxDB

\*\*InfluxDB\*\* is a \_time series\_ database and it is optimised to handle high write and query loads. For this purpose is a very good fit for saving sensor data or time series info from various logs. InfluxDB is not only a time series platform, but it provides also an Web UI and dashboard tools, background processing and monitoring agent.

The main interface to the database for management and data transferred are HTTP requests that are handled directly by the `influxdb` service (default port being used is `8086`).

The data can be nicely viewed with [\*\*Grafana\*\*](../hardware\_projects/#grafana). This installation and documentation was possible, thanks to [@marcobrianza](https://github.com/MichaIng/DietPi/issues/1784#issuecomment-390778313).

=== "Quick start"

After the installation, the data transfers are made via the HTTP requests, and are handled directly by the InfluxDB service running on `http://<your.IP>:8086`.

Create a database using `influxdb` via command line tool. This tool also uses HTTP, so it can manage a database on a remote machine setting the `-host` option:

```sh

influx -execute 'create database myfirstdb'

```

Create a database using a HTTP request and `curl` tool:

```sh

curl -i -X POST http://<your.IP>:8086/query --data-urlencode 'q=CREATE DATABASE myfirstdb'

```

Post data:

```sh

curl -i -X POST 'http://<your.IP>:8086/write?db=myfirstdb' --data-binary 'temperature value=20.12'

```

Retrieve and display data from the database:

```sh

influx -database myfirstdb -execute 'SELECT \* FROM temperature'

```

Retrieve data using a HTTP request and `curl` tool:

```sh

curl -i -XPOST http://<your.IP>:8086/query?db=mydb --data-urlencode "q=SELECT \* FROM temperature"

```

=== "Users and security"

\*\*Create users and authorizations\*\* using `influx` CLI

To start the InfluxDB database management interface enter:

```sh

influx -username admin -password admin01

```

Then create the database entries:

```sh

CREATE USER admin WITH PASSWORD 'admin01' WITH ALL PRIVILEGES

CREATE USER test\_user WITH PASSWORD 'test\_user01'

GRANT ALL ON mydb TO test\_user

exit

```

=== "Enable secured access (HTTPS)"

By default the \_HTTP\_ authentication is disabled. To enable it, follow next two steps:

1. Change next setting in the configuration file `/etc/influxdb/influxdb.conf`:

```

auth-enabled = true

```

2. Restart the service:

```sh

systemctl restart influxdb

```

=== "Install information"

The data location for InfluxDB is stored respectively linked with symbolic links to the DietPi userdata directory: `/mnt/dietpi\_userdata/influxdb`

\*\*\*

Website: <https://www.influxdata.com/products/influxdb/>

Official documentation: <https://docs.influxdata.com/influxdb/v1.8/>

Getting started: <https://docs.influxdata.com/influxdb/v1.8/introduction/get-started/>

## PostgreSQL

PostgreSQL is a persistent advanced object-relational database server, used in similar scenarios as MariaDB.

![PostgreSQL logo](../assets/images/dietpi-software-postgresql.png){: width="200" height="206" loading="lazy"}

=== "Implementation details"

While the Debian package by ships PostgreSQL with an active TCP/IP listener, though on localhost only, when installed via DietPi-Software this is disabled by default. We recommend using the UNIX domain socket in `/run/postgresql` to connect to the database, which has performance benefits. When TCP/IP connections are required, best practice is to create an override config like `/etc/postgresql/\*/main/conf.d/99local.conf` and setting the listening address:

```

listen\_addresses = 'localhost'

```

Replace `localhost` with an actual IP address to allow remote access or with `\*` to all access via all LAN and public IP addresses and domain names.

When installed via DietPi-Software, the actual database files are stored in `/mnt/dietpi\_userdata/postgresql`, so that it can easily moved to an external drive, together with other DietPi userdata. For backwards-compatibility, a symlink is created at `/var/lib/postgresql`.

=== "Configuration"

- Config directory:

`/etc/postgresql/\*/main`

with the asterisk being the PostgreSQL version number, e.g. `11` or `13`

- Main config file:

`/etc/postgresql/\*/main/postgresql.conf`

- DietPi config override:

`/etc/postgresql/\*/main/conf.d/00dietpi.conf`

To add or change settings, best practice is to create a new override configuration, e.g.:

```

/etc/postgresql/\*/main/conf.d/99local.conf

```

For changes to take effect, the service needs to be reloaded:

```sh

systemctl reload postgresql

```

=== "Service handling"

The systemd service `postgresql.service` is used to start and control the PostgreSQL server. The following commands can be used:

- Start: `systemctl start postgresql`

- Stop: `systemctl stop postgresql`

- Restart: `systemctl restart postgresql`

- Reload config: `systemctl reload postgresql`

- Print status: `systemctl start postgresql`

=== "View logs"

Service logs are done to the system journal an can be viewed via:

```sh

journalctl -u postgresql

```

The server itself by default logs to a file:

```sh

cat /var/log/postgresql/postgresql-\*-main.log

```

=== "Update to latest version"

Since PostgreSQL is installed via APT, it can be updated via:

```sh

apt install postgresql

```

\*\*\*

Official website: <https://www.postgresql.org/>

Official documentation: <https://www.postgresql.org/docs/>

Source code: <https://git.postgresql.org/gitweb/?p=postgresql.git>

License: [PostgreSQL Licence](https://www.postgresql.org/about/licence/)

[Return to the \*\*Optimised Software list\*\*](../../software/)

[^1]: [About MariaDB Server and MariaDB Foundation](https://mariadb.org/about/)

[^2]: [About SQLite](https://www.sqlite.com/about.html)