Setting up a PostgreSQL Database on a Raspberry Pi

by Emmet

Published Mar 26, 2021 Beginner, Servers

In this project, we will show you how to set up a PostgreSQL Database on a Raspberry Pi.

PostgreSQL is a free and open-source relational database system that implements the SQL language.

If you are familiar with MySQL or MariaDB, you will be comfortable using PostgreSQL on your Raspberry Pi.

The reason for this is that all of these servers make use of SQL to manage their databases.

There are a few reasons why you may choose to use PostgreSQL over MySQL. One of those being PostgreSQL's feature set and its compliance with implementing the SQL language.

The following sections will show you how to install PostgreSQL to your Raspberry Pi and create your very first database and table.

Setting up PostgreSQL to the Raspberry Pi

Installing PostgreSQL is a simple process thanks to it being available through the Raspberry Pi OS package repository.

The following steps will get you to install Postgres and configure the software so you can interact with the database using the pi user.

Installing the PostgreSQL Server

Our first step is to install the PostgreSQL server software to your Raspberry Pi.

This installation process is fairly straightforward and only takes two steps.

1. We need to start by updating our Raspberry Pi's operating system.

Updating ensures we have a fresh base to install the PostgreSQL software to.

Run the following two commands on your Pi to update the package list and the installed packages.

Terminal $ Copy

```
sudo apt update
sudo apt full-upgrade
```

2. Our next step is to install the PostgreSQL package to our Raspberry Pi by using the command below.

Alongside the Postgres server, this command will also install additional bits of software such as the command-line interface.

Terminal $ Copy

```
sudo apt install postgresql
```

Configuring PostgreSQL's CLI to Use the `pi` User

At the moment, if you try to use the PostgreSQL CLI using your Pi user, you will get an error saying you don't have a role setup for it.

Don't worry though, as we can change to the `postgres` user to create a new role.

1. Your first step is to change to the `postgres` user.

You can do this by running the following command on your Raspberry Pi.

Terminal $ Copy

```
sudo su postgres
```

2. Once you have changed to the `postgres` user, we can now create a new role.

Run the command below on your device to begin the process of creating a role for our `pi` user.

```
Terminal $                                                          Copy

  createuser pi -P --interactive
```

During this process, you will be prompted to answer a few details.

3. The first prompt will ask you to enter a password for this new role.

Setting a password adds an extra layer of security when connecting to your Raspberry Pi's PostgreSQL server.

```
Enter Password for new Role

  Enter password for new role:
  Enter it again:
```

4. You will now need to decide if you want this new user to be a superuser.

If you want to use the `pi` user to create databases and other stuff, answer `Y` to this question.

```
Should new Role be a Superuser

  Shall the new role be a superuser? (y/n) y
```

5. Our next step is to create a database that has the same name as our user.

When you try to use the command-line interface, it will automatically try and connect to a database with the same name as your username.

Load up the command line interface by running the following command.

```
Terminal $                                                          Copy

  psql
```

6. Now, create a database with the `pi` username by typing in the following.

This database is mainly here to make it easier to start using the CLI tool easily.

```
postgres >

  CREATE DATABASE pi;
```

7. You can quit out of the CLI tool by typing the following into PostgreSQL.

```
postgres >

  exit
```

8. With the role created for our `pi` user, we can now change back to it.

To get back to the `pi` user, all you need to do is enter the command below.

```
Terminal $                                                     Copy

  exit
```

Creating your first PostgreSQL Database on the Raspberry Pi

To test that PostgreSQL is now up and running on our Raspberry Pi, we will now create a new database on it.

These steps will show you how to use the Postgres CLI to create a database and add a table to it.

1. Let us start this process by loading into the Postgres CLI.

Run the following command to switch into the command line interface.

```
Terminal $                                                     Copy

  psql
```

2. Within the PostgreSQL command line, we can now create and manipulate databases.

As we mentioned earlier, Postgres uses the SQL programming language. If you have ever used SQL before, you will feel right out home.

For our first task, let us create a new database on our Raspberry Pi. For this example, we will call this database " `exampledb` ".

```
postgres >

  CREATE DATABASE exampledb;
```

3. Next, you will need to change to this new database.

By default, when you launch the Postgres CLI, you will interact with the database with the same name as your user.

To change to our new DB, we need to use the " `\connect` " command followed by our DB's name.

```
postgres >

  \connect exampledb;
```

Once the CLI has connected to your new database, you should get a message as we have below.

```
Successful Database Connection

  You are now connected to database "exampledb" as user "pi".
```

4. Our next step to show PostgreSQL is running on the Raspberry Pi is to create a table within our database.

We are going to make this a simple table called " `authors` " that has two columns. One column will be called " `name` ", and the other will be called " `website` ".

We will set the data type for both of these columns to " `text` ".

```
postgres >

  CREATE TABLE authors (name text, website text);
```

4. Let us now add a row of data to this brand new table.

Having support for SQL makes adding this data a simple process. For our row, we will add an author with the name " `Emmet` " and the website " `pimylifeup.com` "

```
postgres >

  INSERT INTO authors VALUES ('Emmet', 'pimylifeup.com');
```

5. Finally, our last task is to get PostgreSQL to output the data stored in this table.

To do this, we can use the following simple select statement.

```
postgres >

  SELECT * FROM authors
```

From this command, you should see all of the data that is stored within the specified table.

As we have only added a single row to this table, that is all you should see outputted into the command line.

```
Postgres Select Output

   name  |    website
  -------+----------------
   Emmet | pimylifeup.com
  (1 row)
```

Conclusion

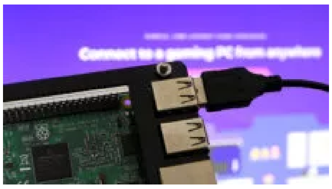
You should now have a good understanding of how you can install and set up a PostgreSQL database on your Raspberry Pi.

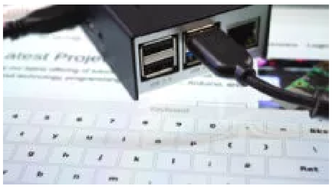
PostgreSQL is a solid alternative to MySQL and MariaDB. It has a plethora of features and has one of the most complete implementations of SQL.

If you have had any issues with getting the Postgres server running on your device, please leave a comment below.

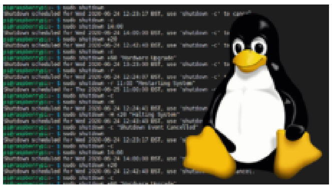
Be sure to check out some of our other Raspberry Pi projects to see what else the device is capable of.

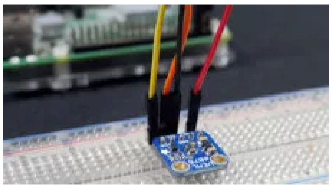
Recommended


Setting up Parsec on the Raspberry Pi


Setting up an On-Screen Keyboard on the Raspberry Pi

Wake Word Detection on the Raspberry Pi with Porcupine


Using the shutdown Command in Linux


Raspberry Pi UV Sensor using the VEML6075

Compiling FFmpeg on the Raspberry Pi

One Comment

Vincent on September 20, 2021 at 9:43 am

Great guide, love it.

If anyone else is having an issue with the SQL statement, don’t forget the semicolon at the head like I did, spent a lot of time trying to realize when it was that all along.

Reply

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

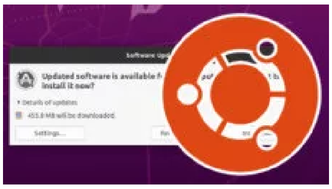
Email *

□ Notify me of follow-up comments by email.

Post Comment

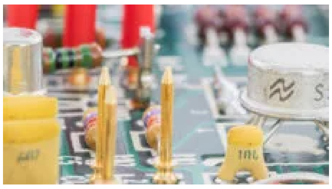
Trending

How to setup a Raspberry Pi TOR Access Point



How to Update Software on Ubuntu

How to Setup Surfshark on the Raspberry Pi



Understanding Ohm's Law

Back to Top

Disclaimer & Privacy Policy | About us | Contact