Installing InfluxDB to the Raspberry Pi

In this Raspberry Pi InfluxDB tutorial, we will be showing you how to set up and install InfluxDB on the Raspberry Pi.

We will also be showing you how to enable authentication on your InfluxDB server to improve its security, as well as showing you how to interact with the database through the command line.

For those who are wondering what InfluxDB is, it is a time series based database system. This means that each data point in the database will contain a timestamp.

Being time series based makes InfluxDB one of the best databases for monitoring metrics and events. You can easily use InfluxDB to store information like the temperature in a room or the CPU usage of a system.

InfluxDB is the perfect database software to go alongside the popular visualization tool Grafana. Grafana has inbuilt support for displaying data from an InfluxDB database.

Equipment

Below is all the equipment that you will need to set up InfluxDB on your Raspberry Pi.

Recommended Equipment

- Raspberry Pi
- Micro SD Card
- Power Supply
- Ethernet Cable or Wi-Fi

Optional Equipment

- USB Keyboard
- USB Mouse
- HDMI Cable

This tutorial was last tested on a Raspberry Pi 400 running Raspberry Pi OS Bookworkm.

Installing InfluxDB to the Raspberry Pi

Over the next couple of sections, we will walk you through the process of setting up your Raspberry Pi to install and run InfluxDB.

Preparing your Raspberry Pi for InfluxDB

1. The first thing we should do before installing InfluxDB to the Raspberry Pi is making sure that all the currently installed packages are up to date.

We can upgrade all installed packages by running the following two commands.

```
sudo apt update
sudo apt upgrade
```

Adding the InfluxDB Repository

2. With everything now up to date, we can now proceed with installing InfluxDB to the Raspberry Pi.

Our next step is to add the InfluxDB repository key to our Raspberry Pi.

Adding the key will allow the package manager on Raspberry Pi OS to search the repository and verify the packages its installing.

We can add the InfluxDB key by running the following command.

```
curl https://repos.influxdata.com/influxdata-archive.key | gpg --dearmor | sudo tee
/usr/share/keyrings/influxdb-archive-keyring.gpg >/dev/null
```

This command will download the key using curl and pass it directly into the "**gpg**" program by using a pipe "`|`". Once the keyring has been de-armored it gets saved into the "`/usr/share/keyrings/`" directory.

3. Now that we have the InfluxDB repository key installed to our Raspberry Pi, we will need to go ahead and add its repository to the sources list.

Now enter the following command to add the InfluxDB repository to the sources list. This command will work for anyone running a Debian based operating system such as Ubuntu or Raspberry Pi OS.

```
echo "deb [signed-by=/usr/share/keyrings/influxdb-archive-keyring.gpg]
https://repos.influxdata.com/debian stable main" | sudo tee
/etc/apt/sources.list.d/influxdb.list
```

4. With the repository added, we now need to go ahead and update the package list again.

We need to do this so that the apt package manager searches the repository that we just added for packages. The operating system does not automatically do this.

Run the following command on your Raspberry Pi to update the package list.

```
sudo apt update
```

Installing InfluxDB to your Raspberry Pi

5. Now that we have set up the repository, we can now move on to installing the InfluxDB software. Before you do this though, you must decide whether you want to run InfluxDB V2 or V1 on your Raspberry Pi.

These versions have some significant differences, so double check what version is required for your use case. Some users prefer to stick with V1 when using Grafana.

Installing InfluxDB V2 to the Raspberry Pi

6a. Installing InfluxDB V2 is as straightforward as using the following command within your terminal.

The package manager will install both the database as well as a CLI tool.

```
sudo apt install influxdb2
```

Installing InfluxDB V1

6b. To instal InfluxDB V1 on your Raspberry Pi, then you will want to use the following command.

```
sudo apt install influxdb
```

Starting the InfluxDB Server

7. With InfluxDB now installed to our Raspberry Pi, let's now get it to start at boot.

We can do this by making use of the systemctl service manager to enable our InfluxDB service file.

Run the following two commands to enable InfluxDB to start at boot on your Raspberry Pi.

```
sudo systemctl unmask influxdb
sudo systemctl enable influxdb
```

The first command we use unmasks the influxdb service file. Unmasking the service ensures that we can enable and start the service as a masked service is unable to be started.

Our second command enables the influxdb service. This command will tell the service manager to keep an eye on the "**influxdb.service**" file and setup the service based on its contents.

8. Now that everything has been set up, we can now proceed to start up InfluxDB on our Raspberry Pi.

To start up the InfluxDB server, we will need to run the following command. The service manager will then start up the service and begin monitoring it.

```
sudo systemctl start influxdb
```

How to Use InfluxDB V2 on the Raspberry Pi

One of th keye ways that InfluxDB V1 and V2 differ is how you access them. While V1 expects you to use the terminal for most things, V2 relies heavily on a GUI and access tokens.

This section will cover the initial setup experience of InfluxDB V2 on your Raspberry Pi. If you chose to use V1 instead, then skip this section.

Accessing the InfluxDB V2 Web Interface

1. First, you will need to know the local IP address of your Raspberry Pi. If you don't know it, you can always use the hostname command as shown below.

```
hostname -I
```

2. Your next step is to go to the following URL in your favourite web browser. This will take you to InfluxDB V2's web interface.

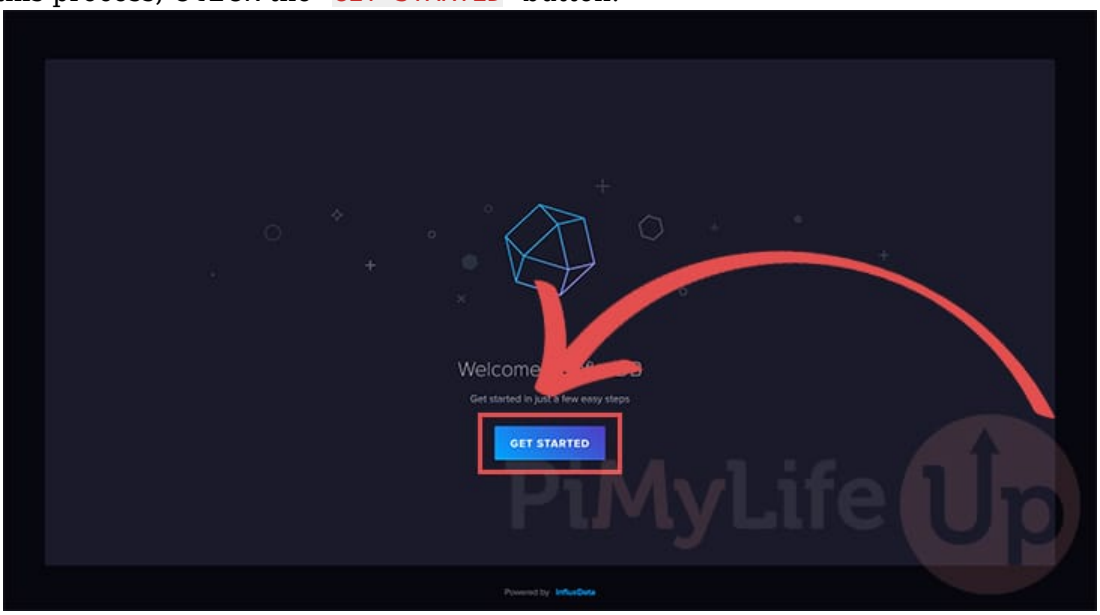
Ensure you replace "`<IPADDRESS>`" with your IP. If you are opening this page on the Pi itself, you can also use "`localhost`" or "`127.0.0.1`".

```
http://<IPADDRESS>:8086/
```

Initial Setup

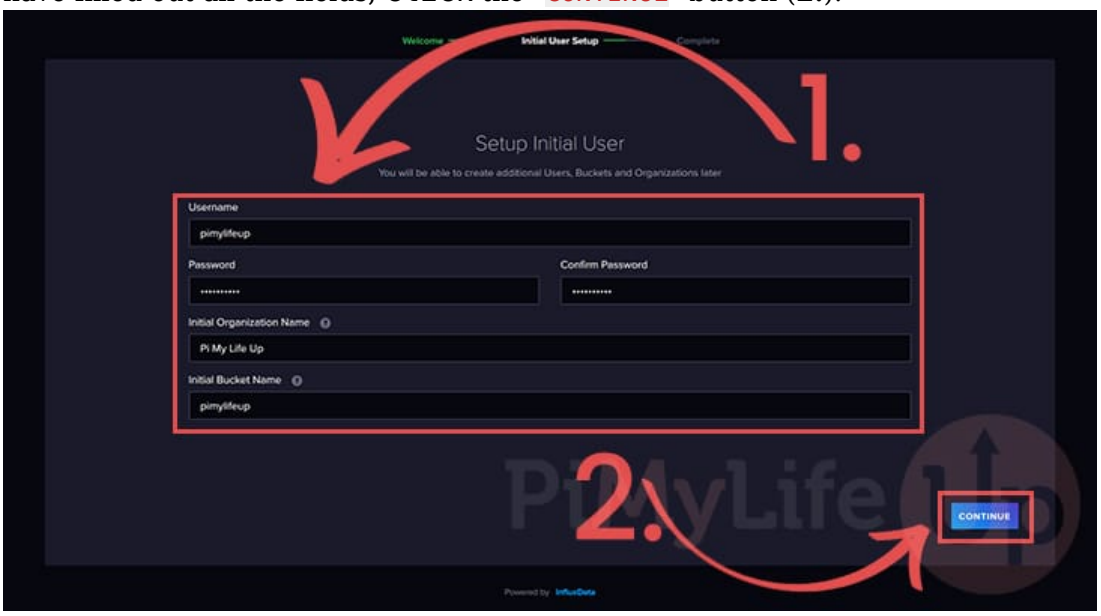
3. When you first access the InfluxDB 2 interface, you will need to create an initial user.

To begin this process, `click` the "`GET STARTED`" button.



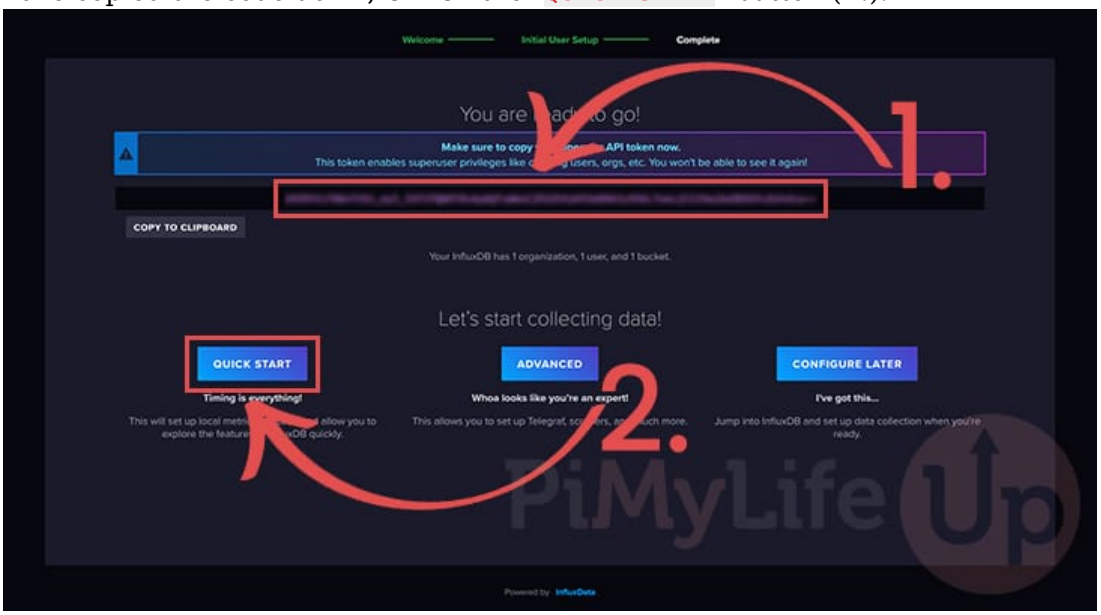
4. You will now need to fill out details for your user (**1.**). Ensure you set a strong password, as this is what you will use to access this web interface in the future.

Once you have filled out all the fields, `click` the "`CONTINUE`" button (**2.**).



5. This next step is extremely important. InfluxDB2 will now have generated an API token on your Raspberry Pi. Copy down the API token as shown on your screen (**1.**). This token is what gives you superuser privileges and will not be shown once you leave this page.

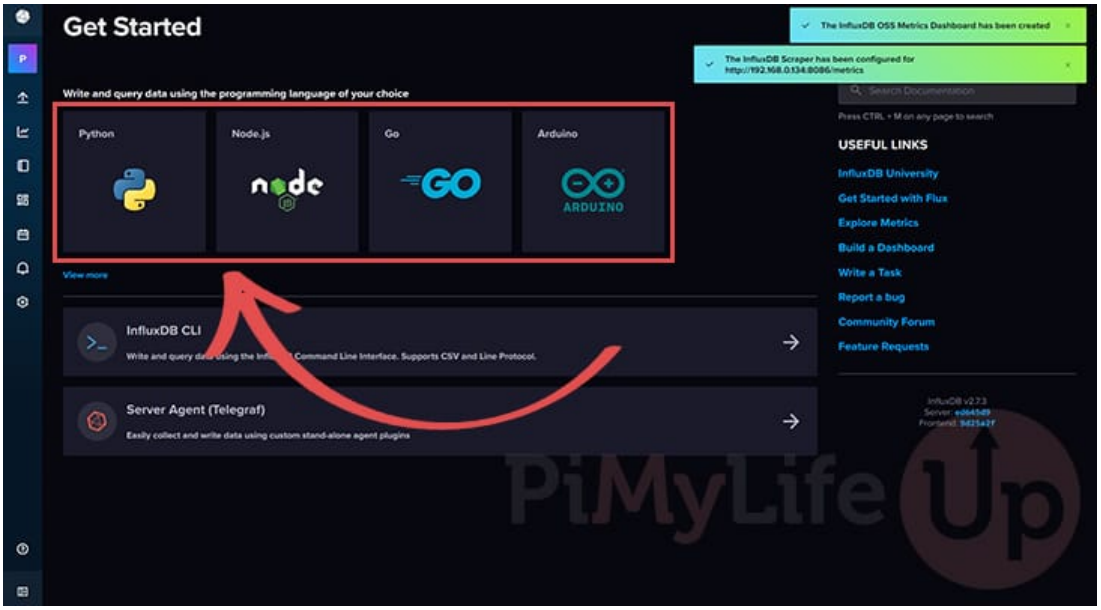
Once you have copied the code down, `click` the "`QUICK START`" button (**2.**).



Getting Started

6. InfluxDB v2 provides various guides on using this database on your Raspberry Pi.

Select the box that represents the programming language you want to use and follow the steps.



Using InfluxDB V1 on your Raspberry Pi

In this section we will be exploring how you can use InfluxDB V1 on the Raspberry Pi. Msot of your interaction with this verison of InfluxDB will be using its command line interface.

Testing out InfluxDB V1

1. As we have now installed InfluxDB V1, we can now start talking with the database.

To do this, we will need to launch up Influx's command-line tool by running the command below.

You don't have to worry about specifying an address to connect to as the tool will automatically detect the local installation of InfluxDB.

By default, InfluxDB has no users setup. In our next section, we will explore creating an admin user to lock down access to your InfluxDB. For now, however, we will quickly explore InfluxDB.

```
influx
```

2. InfluxDB comes with no databases by default, so our first task will be to create one.

Creating a database is simple in InfluxDB and can be done by using "`CREATE DATABASE <DBNAME>`".

For our example, we will be creating a database called "`pimylifeuptemperature`".

```
CREATE DATABASE pimylifeuptemperature
```

3. Before we can start modifying our new database, we must tell the CLI to "`use`" it.

Using a database is as simple as running the following command.

```
USE pimylifeuptemperature
```

4. Our next step is to write some data to our newly created InfluxDB database.

To do this, we must first get a basic understanding of InfluxDB's datastore.

Data in InfluxDB are sorted by "**time series**". These "time series" can contain as many or as little data points as you need. Each of these data points represents a single sample of that metric.

A data point consists of the **time**, a **measurement** name such as "temperature", and at least one **field**. You can also use **tags** which are indexed pieces of data that are a string only. Tags are essential for optimizing database lookups.

If you are familiar with the general layout of an SQL table, you can consider "**time**" to be the primary index, **measurement** as the table name, and the **tags** and **fields** as the column names.

You do not need to specify the timestamp unless you want to specify a specific time and date for the data point.

Below we have included the basic format of an InfluxDB data point.

```
<measurement>[,<tag-key>=<tag-value>...] <field-key>=<field-value>[,<field2-
key>=<field2-value>...] [unix-nano-timestamp]
```

If you would like to learn more about the InfluxDB line syntax, then you can check out the InfluxDB official documentation.

5. Now that we have a basic understanding of data in InfluxDB, we can now proceed to add our very first data point to our database.

For this example database, we are going to be storing measurements of the "**temperature**" of various **locations** around a house.

So for this, we will be inserting data points with a measurement name of "**temperature**" and a tag key of "**location**", and a field key of "**value**".

For our first sample point, we will be saying the location is the "**living_room**", and the value is "**20**" .

```
INSERT temperature,location=living_room value=20
```

6. To make the data more interesting for showing off "selecting" data in InfluxDB, let's go ahead and add some more random data.

Enter the following few commands to enter some extra data into our database. These are just variations of the above "**INSERT**" command but with the value and location adjusted.

```
INSERT temperature,location=living_room value=10
INSERT temperature,location=bedroom value=34
INSERT temperature,location=bedroom value=23
```

7. Now that we have some sample data, we can now show you how to query this data using "**SELECT**".

To start with, you can retrieve all data from a measurement by using a command like below. This command will grab all fields and keys from the specified measurement.

```
SELECT * FROM temperature
```

Using that command with our sample data you should get a result like we have below.

```
name: temperature
time                location    value
----                --------    -----
1574055049844513350 living_room 20
1574055196564029842 living_room 10
1574055196576516557 bedroom     34
1574055197188117724 bedroom     23
```

8. Let's say that you now only wanted to retrieve the temperature of the bedroom. You can do that by making use of the "**WHERE**" statement alongside a "**SELECT**" statement.

We also specify the name of the tags/fields that we want to retrieve the values from.

When querying tag fields, you need to remember that all tags are considered to be strings. This means that we must wrap the value we are searching for in single quotes.

```
SELECT value FROM temperature WHERE location='bedroom'
```

With that command, you should receive the following data set, showing only the temperature value in the bedroom.

Which in our example data's case, this should be **34** and **23**.

```
name: temperature
time                location    value
----                --------    -----
1574055049844513350 living_room 20
1574055196564029842 living_room 10
1574055196576516557 bedroom     34
1574055197188117724 bedroom     23
```

9. At this point, you should now have a basic understanding of InfluxDB and how its data works.

Adding Authentication to InfluxDB V1

1. The next step is to add extra authentication to our InfluxDB installation on the Raspberry Pi. Without authentication, anyone could interact with your database.

To get started, we need to first create a user to act as our admin.

To create this user, we must first load up the InfluxDB CLI tool by running the following command.

```
influx
```

2. Within this interface, we can create a user that will have full access to the database. This user will act as our admin account.

To create this admin user, run the following command within InfluxDB's CLI tool.

Make sure that you replace **<password>** with a secure password of your choice.

```
CREATE USER admin WITH PASSWORD '<password>' WITH ALL PRIVILEGES
```

This command will create a new user called "**admin**" with your chosen password and grant it all privileges.

3. With that done, you can now exit out of InfluxDB by typing in "**exit**" and pressing ENTER.

4. Our next job is to modify the InfluxDB config file to enable authentication.

We can begin editing the file by using the command below.

```
sudo nano /etc/influxdb/influxdb.conf
```

5. Within this file, use CTRL + W to find the **[HTTP]** section and add the following options underneath that section.

Find

```
[HTTP]
```

Add Below

```
auth-enabled = true
pprof-enabled = true
pprof-auth-enabled = true
ping-auth-enabled = true
```

6. Once added, save the file by pressing CTRL + X, then Y, followed by ENTER.

7. Now, as we have made changes to InfluxDB's configuration, we will need to go ahead and restart the service by using the following command.

Restarting the service will ensure that our configuration changes are read in.

```
sudo systemctl restart influxdb
```

8. As we have now turned InfluxDB's authentication on, we will need to enter our username and password before using the InfluxDB CLI tool.

You can use the "**auth**" command within the tool or pass the username and password in through the command line, as we have below.

```
influx -username admin -password <password>
```

Conclusion

Hopefully, at this point, you will now have successfully set up InfluxDB on your Raspberry Pi. You should now have a basic understanding of InfluxDB as well as have authentication mode enabled.

You can easily see how using this database model will make storing data from sensors and other sources very useful.