

Real Time Data Visualization Documentation

How to Setup Ubuntu Server on Raspberry Pi

1. Download the Raspberry Pi Imager

We are going to use this to create an image of the OS onto the microSD card which will go onto our Raspberry Pi 3.

2. Choose the Raspberry Pi Device and also the Operating System

In this case we will be using Ubuntu Server 25.04 (64-bit server). However, there are many operating systems available for us to use.

3. Edit WiFi and SSH settings

Preconfigure the country, username, hostname, and SSH Password authentication settings.

4. Plug in SD card and begin to write the image onto microSD card

5. Plug in your Raspberry Pi and plug in the microSD card into the device

The device should light up red initially, give it a couple of minutes and then it should light up green when the OS has finished booting up.

After it has finished, verify that it works by plugging in an HDMI cord to a display. It should light up and give you something. You can also SSH into the device at this moment in time.

To SSH into the device you go to a terminal in MacOS or any Linux distribution would do

`<username>@<hostname.local>` or `<username>@<ipaddress>` . It then will prompt you to input your password. This is the password that you chose when creating the OS. You can input it and you should be in.

For Windows:

Download Putty which is an SSH client. This will allow you to SSH into the device and has better GUI for the device. To SSH into the device on Putty you have to input the hostname, select port 22, keep the option set the SSH, and then connect. At this point it should ask for the username and then it'll ask you for your password and you should be in.

Installing Necessary IoT Software from the MING Stack

The MING stack is a set of technologies that allow for IoT communication, collection of data from the Raspberry Pi, and also the visualization of real-time data.

M stands for Eclipse Mosquitto which is a MQTT broker. MQTT is a lightweight messaging protocol which is usually used in IoT settings. It's used over HTTP in our case because it's more optimized for resource-constrained devices. The MQTT broker facilitates communication between MQTT clients using the publish/subscribe model which allows data-exchange.

The I in MING stands for InfluxDB which is a database that is specifically built for time-series data. It is primarily used for real-time collection of data. You are also able to take the data from the bucket it is stored in using API tokens that allow communication between different applications.

The N in MING stands for Node Red which is a tool that is used for connecting together devices and determining the flow of data. It uses nodes which are draggable modules that connect together, so that you don't have to write pure JavaScript code. It wires together the communication between the MQTT client and the Raspberry Pi.

The G in MING stands for Grafana which is a real-time data visualization software that allows you to see your data and metrics in different forms. You are able to select the data that you want and use the metrics that are available.

Installing Eclipse Mosquitto and Updating OS

When you first SSH into the device you want to first run `sudo apt upgrade`. Sudo runs the command with administrative privileges, apt is the package manager for Debian OS systems, and upgrade upgrades all the packages to the latest versions that are available in the repositories.

To install Mosquitto you want to run `sudo apt install mosquitto mosquitto-clients` which installs mosquitto, mosquitto-clients, and all the necessary dependencies. After everything is installed you then want to change the config for Mosquitto. To do this, you would run `sudo nano /etc/mosquitto/mosquitto.conf`. Nano is a terminal based text editor that allows you to change the config file for Mosquitto. I used Helix in my case which is another terminal text based editor. At the end of the config file you want to add `listener 1883` and `allow_anonymous true`. Listener 1883 puts Mosquitto to listen onto port 1883 of the Raspberry Pi. Allow anonymous true makes it so that clients without a username are able to connect. To get out of nano you have to do Shift+C which write the file and then press Shift+X.

After you are out of the file you want to input `sudo systemctl restart mosquitto`. Systemctl manages systemd services which is just system services.

Installing Node-RED

To install Node-RED you need to go to the installation documentation for Linux since we are using Ubuntu Server, you can copy the code down, but it should be

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

This will then install Node-RED which will take a while to download. Once it finishes, you will scroll all the way to the bottom and press enter. It will ask you to setup user security which you will say yes to, and then set your Username and Password. You will also be able to set your theme (You have to download theme collections to your user directory to use anything other than default) and select your text editor component which I set as default. It will then ask you if you want function nodes to load external modules which you will say yes to.

What we want to do next is have Node-RED not take up so much space on the Raspberry Pi. To do this we will run `node-red-pi --max-old-space-size=256`. It will then start Node-RED automatically. To end the process and keep it in the background you can press Ctrl+C. to stop Node-RED you can input `node-red-stop` and to start it back up you enter `node-red-start`

Installing InfluxDB

To install InfluxDB you want to run these two commands

```
curl --silent --location -O \https://repos.influxdata.com/influxdata-archive.key
```

```
echo "943666881a1b8d9b849b74caebf02d3465d6beb716510d86a39f6c8e8dac7515
influxdata-archive.key" \ | sha256sum --check - && cat influxdata-archive.key \
| gpg --dearmor \ | sudo tee /etc/apt/trusted.gpg.d/influxdata-archive.gpg >
/dev/null \ && echo 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdata-
archive.gpg] https://repos.influxdata.com/debian stable main' \ | sudo tee
/etc/apt/sources.list.d/influxdata.list
```

```
sudo apt-get update && sudo apt-get install influxdb2
```

You then want to enter `sudo systemctl enable influxdb` which will enable Influx DB as a systemd program which will run in the background.

Then you want to run it using `sudo systemctl start influxdb`

Installing Grafana

To install Grafana you want to enter `sudo apt install grafana`

And you then want to run `sudo systemctl enable grafana-server` which enables Grafana as a systemd program which will run in the background.

Getting Raspberry Pi WLAN IP

If your Raspberry Pi doesn't come with net tools installed you want to run `sudo apt install net-tools` which will download the net-tools package using apt.

From this you can run `ifconfig` which is a command that allows you to get your network interface information.

The IP that you want is under wlan0 under inet. It should be a number in this format:
111.111.1.111

You want to copy that because we are using it later to access our services.

Logging Into Our MING Stack Services

To log into our services you have to enter the IP that you just grabbed + the port number that it's on.

Node-RED

The port that Node-RED is on is 1880. To log in you'll enter (filling in your system information) `ip:port-number` in a web browser. Entering that address will bring you to a website page where you can log in with your username and password that you configured earlier. We'll come back to configuring everything in the next sections when we log into all of our services.

InfluxDB

The port that InfluxDB is on is 8086. To log in you'll enter (filling in your system information) `ip:port-number` in a web browser. Entering that address will bring you to a website page where you can press the get started button. You'll want to setup the initial user by creating a username and a password that you will use to log in. You can also set your initial organization name and initial bucket name (keep these names in mind).

Grafana

The port that Grafana is on is 3000. To log in you'll enter (filling in your system information) `ip:port-number` in a web browser. Entering that address will bring you to a website page where you can log in with your username and password from your Raspberry Pi. We'll come back to configuring everything in the next sections.

Creating Python Script and Installing Dependencies in a Virtual Environment

My python script was configured in a virtual environment that is provided by the Python3 package.

Python3 may already be installed on your system, but to download these packages you want to enter `sudo apt install python3 python3-pip python3-venv`

`python3-pip`: Pip is a Python-specific package installer/manager. This will allow you to install packages and libraries that you need for your Python script to work, while also installing the necessary dependencies

`python3-venv`: This is a Python3 Virtual Environment. It acts like a container that contains all of the packages and dependencies for your project. This allows you to download only necessary packages and dependencies without compromising other code since it only exists within the Virtual Environment.

You first want to create a virtual environment in a directory. This is done through entering `mkdir ~/whateverName` where you can replace the directory name with whatever name you want. You then want to get into the directory which is done by entering `cd ~/whateverName` which brings you into the directory. Once you're inside the directory you can create your virtual environment by entering `python3 -m venv whateverName` where you can choose the name for your virtual environment. You then want to start the virtual environment by entering `source whateverName/bin/activate` which will activate your virtual environment.

Once you're inside the virtual environment you want to start downloading the packages that are necessary for your project. In our case since we are saving to CSV we want Pandas.

Here is a list of libraries that need to be installed using pip:

Pandas: `pip install pandas`

Paho-MQTT: `pip install paho-mqtt`

Python3 MPU6050: `pip install mpu6050-raspberrypi`

Python3-smbus: `sudo apt install python3-smbus`

After you're done with installing everything, you can start working on your Python script. This will be done in the directory that you created earlier in the virtual environment. To create a file you can use nano and configure it yourself. I personally used Helix to create a file and edit it. I will include my Python script that you can use as a reference to get your accelerometer to read.

After you have created your python script you then run it using `python3 yourFile.py` which will run your file. If you have configured it correctly you should start reading raw accelerometer values.

Node-RED Configuration

We are going to configure Node-RED to configure the dataflow between our MQTT client and the database.

First, from the network tab on the left, you want to drag in a node called MQTT in. You'll then edit this node by double clicking on it. You can edit the server by pressing the edit icon which will put you into "Edit mqtt-broker mode". In this mode you can set the server as localhost and the port to 1883 then you update it on the top right using the update button. The topic will be the JSON file path that you configured earlier in your Python script since that's going to be saved to InfluxDB and you can also set QoS to 1 from the dropdown menu.

You then want to drag a debug node from the common tab on the left and place it to the right of the MQTT in node. You'll then want to link the two together by clicking on the grey area on the left node and dragging it over to the debug section. Once you connect the two, you can click on the debug icon on the panel on the right and you will be able to see the values from your accelerometer. If you aren't seeing any readings you will want to press deploy at the top.

The last thing we need to do is get an InfluxDB node which will save our data into InfluxDB. The InfluxDB nodes might not be in our dashboard by default so to get them into there we want to click on the icon on the top right with the 3 lines. Press manage palette, press install, search for influx, and you will install "node-red-contrib-influxdb".

We can then go back to our regular dashboard and we should have a node called influxdb out which we will drag and then double click to edit. You can set host as localhost. We are going to click on the edit icon next to the server section. This will then allow you to edit the URL and the Token. The url should be `http://localhost:8086` which is the localhost on port 8086. The token you are going to get by logging into InfluxDB, hovering over the icons on the left until you see

API tokens and then click on it. From there you want to generate an all access API token by clicking on the left. You can set the description of the token as "nodered". You then copy the token that it gives you and paste it into the Token section in nodered. You then want to update this by clicking the update icon on the top right. Then on the main section you want to update the information for the Organization, Bucket, and measurement (can input any name for it). This will be from information that you set previously in InfluxDB. If you forgot, you can go back to InfluxDB and check for the names.

We are done with our Node-RED configuration.

Grafana Configuration and Visualization on the Dashboard

To configure Grafana you want to generate another API token on InfluxDB. You will copy this API token and then log into Grafana. On Grafana you can press "Add Data Source" in the main panel to add your first data source. You can then search for InfluxDB and click on it. In the dropdown for the Query language you can then pick Flux, and for the URL you can set it to `http://localhost:8086`. You will then set the organization name as the organization name you chose in InfluxDB and then paste the token that you copied earlier. You will also set the default bucket as the bucket that you created earlier in InfluxDB and then you will click save and test.

Then we will go and click on Dashboards on the left and create a dashboard in Grafana. You will then click on Add a visualization and then it will create a visualization panel for you. The data source will automatically be set to InfluxDB, but to get the wanted data we will navigate back to InfluxDB to get a sample query.

To do this you will navigate to buckets in InfluxDB, then at the bottom you will see two sections that are labeled "from" and "filter". Inside of the "from" section you will select the bucket that you created earlier. Inside of the filter section you will select "measurement" from the dropdown and search for the measurement name that you created earlier in Node-RED. Once you click on the measurement name it should pull up another section and in this section you will select the type of data that you want to visualize (X, Y, and Z values). Once you select the values that you want, you can click on the script editor button on the right, and you should get a script. You will want to copy this script and then paste it in the Grafana sample query section. You can also delete `yield(name: "mean")` from the query that you copied if you want to look at raw values only.

Then you should be able to see your visualization on the dashboard. You will then save this dashboard by clicking save dashboard at the top right.

We are then done with getting our real-time accelerometer data to be visualized on Grafana. You can configure your dashboards in whichever way that you'd prefer. You can also have multiple panels inside of your dashboard.

Resources:

Putty Client:

<https://www.putty.org/>

Paho MQTT Library:

<https://github.com/eclipse-paho/paho.mqtt.python>

Python3Smbus Library

<https://github.com/Gadgetoid/py-smbus/tree/master>

Python3 MPU6050 Library:

<https://github.com/m-rtijn/mpu6050>

Installing Mosquitto on Ubuntu Server:

<https://docs.vultr.com/install-mosquitto-mqtt-broker-on-ubuntu-20-04-server>

Node-Red Install:

<https://nodered.org/docs/getting-started/raspberrypi>

InfluxDB Install:

<https://docs.influxdata.com/influxdb/v2/install/?t=Raspberry+Pi#download-and-install-influxdb-v2>

Grafana Install:

<https://grafana.com/tutorials/install-grafana-on-raspberry-pi/>

Grafana Tutorials:

https://grafana.com/tutorials/grafana-fundamentals/?utm_source=grafana_gettingstarted

Mosquitto Config Page:

<https://mosquitto.org/man/mosquitto-conf-5.html>