

Grafana

Installing Grafana and using it for energy monitoring

B. VAN WALSTIJN, R. STAUTTENER, R. BOLDING
Amsterdam University of Applied Sciences
January 28, 2019

-Overview

Grafana is open-source software used for analytics and monitoring. It is currently the leading software for time series analytics.

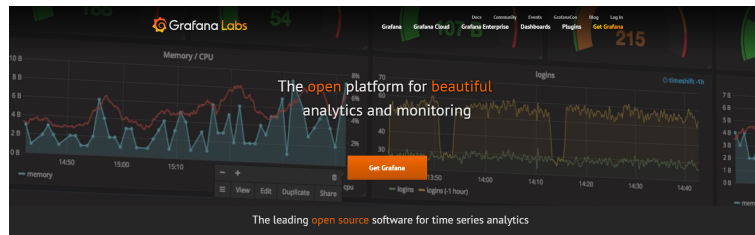


Figure 1: Grafana for monitoring computer clusters

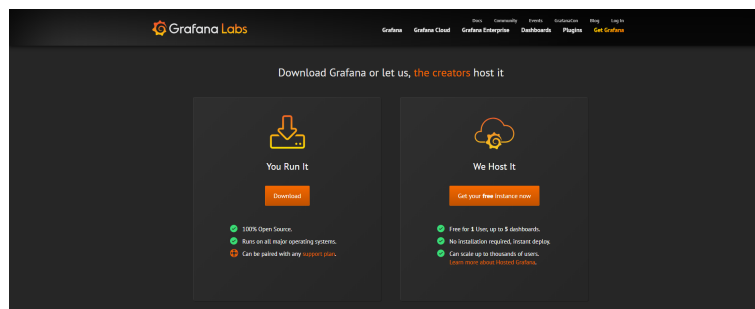
In this case Grafana is going to be used with prometheus and Monasca to monitor the raspberry pi cluster. The focus lies on current and power but metrics such as CPU usage could also be added. This document serves as a guide on installing grafana and the first time setup. /newpage

-Installing Grafana on Windows and Raspberry Pi and setting it up for monitoring.

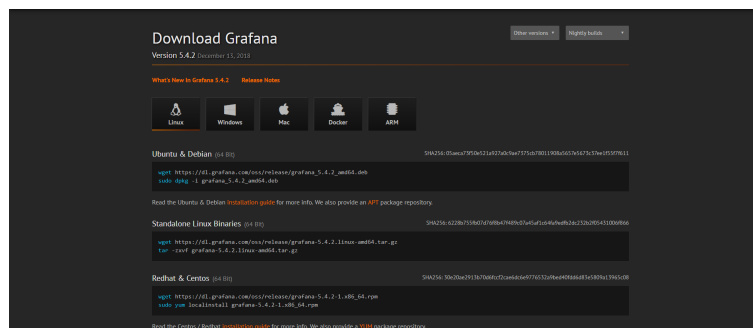
1. Navigate to the following website:
<https://grafana.com/>
2. Click on the orange 'Get Grafana' button, as seen in the screenshot below.



This leads you to the following webpage.



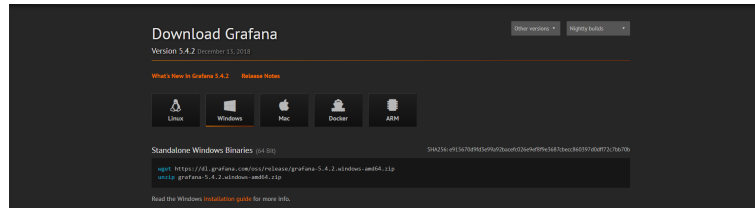
3. Select 'You Run It' and select the operating system that applies to you.



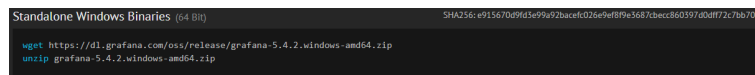
If you want to install Grafana for Windows go to section 3A. If you are installing Grafana on Raspberry pi go to section 3B

-3A Installing Grafana on Windows

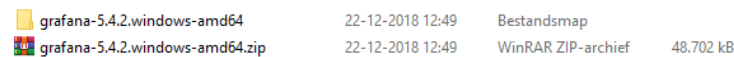
Select Windows. You should now see a page like the following one:



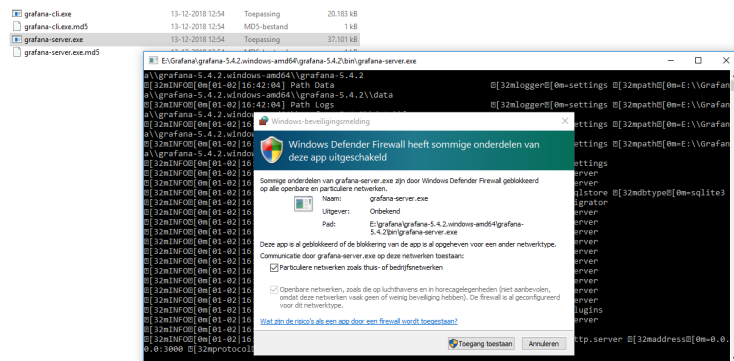
Now click on the link after WGET.



Grafana is now being downloaded as a ZIP file. Unzip this file once the download is complete. Its corresponding folder should look like this:

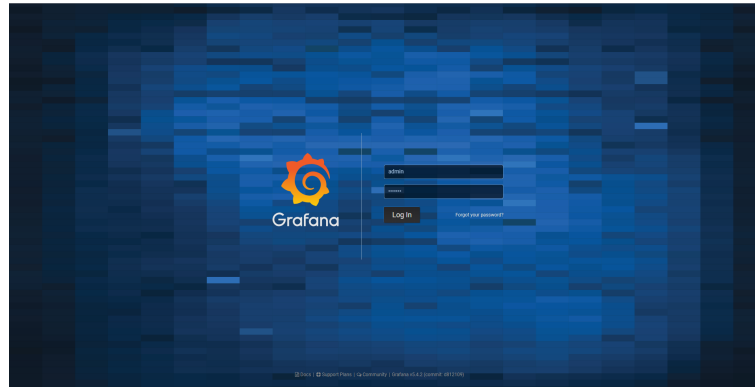


Open the folder you just extracted and navigate to the bin folder. Click on 'Grafana-server.exe'. It may show the following warning:



Click on 'Allow Access' if the warning appears. Please keep in mind that this requires administrator privileges.

Once Grafana-server.exe is running you can navigate to localhost:3000. Doing this should show you the screen as seen below.

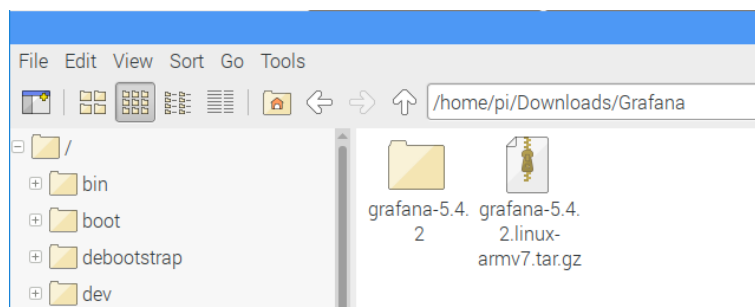


3B. Installing Grafana on ARM (Raspberry Pi)

The raspberry Pi 3B+ has an ARMv6 Broadcom Processor. Grafanas web-site only lists download links for ARM64 and ARMv7. ARMv7 is backwards-compatible with ARMv6. For these reasons, choose the ARMv7 standalone Linux Binary.

```
Standalone Linux Binaries (ARMv7) SHA256: 26db4a494a3775879e385babb76484b911476352d83be494d148ee8cc896cb
wget https://dl.grafana.com/oss/release/grafana-5.4.2.linux-armv7.tar.gz
tar -zxvf grafana-5.4.2.linux-armv7.tar.gz
```

Click on the link after WGET to start downloading Grafana. Unzip the file after downloading.



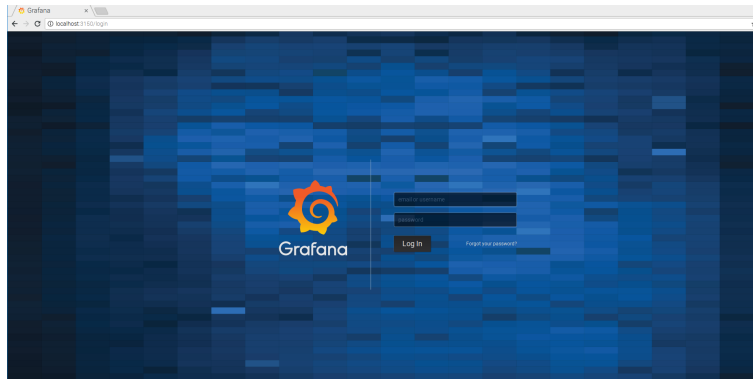
The standard port for Grafana on the Raspberry pi is 8080. Since Prometheus is already posting data to this port it is required that it is changed before using Grafana. This guide now shows how to change the port to 3150.

!!! If you navigate to the folder using the filesystem, you will get a message that you do not have the required permissions to edit the file !!!
The following screen should appear:

Navigate to `http_port` and remove the semicolon. Change the port to 3150. Press `ctrl + x` and answer with `y`. press enter afterwards and reboot the raspberry pi. Navigate to the bin folder in the Grafana folder.



Navigate to Localhost:3150

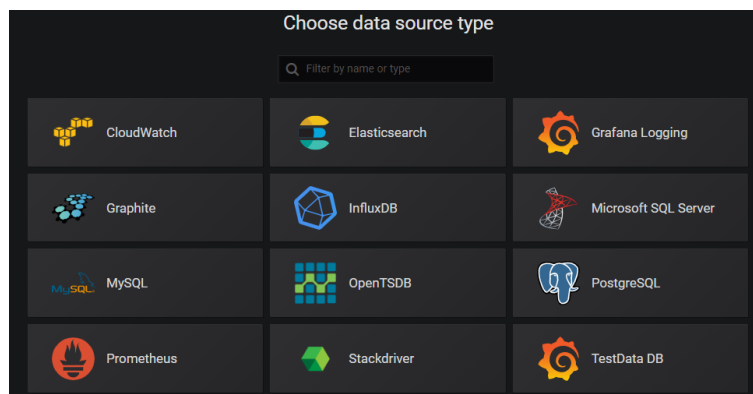


4. First time Login.

Enter the username and password (both admin) and click on log in. When logging in for the first time Grafana will ask you to change the password for the admin account. I choose to set the password to sfs2018

5. Adding a datasource

Grafana's installation is now complete and can be used to monitor the raspberry pi. Click on "Create your first Datasource" and select Prometheus.



You should now see the following screen:

Data Sources / Prometheus-2
Type: Prometheus

Settings

Name: Prometheus-2 Default ☐

HTTP

URL: http://localhost:9090

Access: Server (Default) Help

Whitelisted Cookies: Add Name

Auth

Basic Auth ☐ With Credentials ☐

TLS Client Auth ☐ With CA Cert ☐

Skip TLS Verify ☐

Scrape interval: 15s

Query timeout: 60s

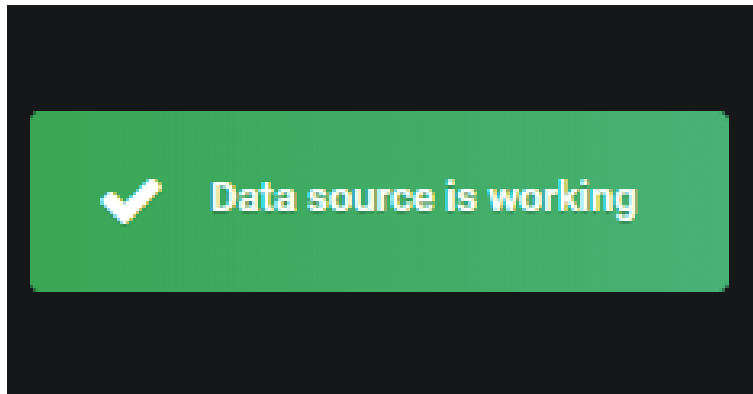
HTTP Method: GET

Save & Test Delete Back

First, under settings change the name. The name isn't particularly important but improves readability further in the process.

Under HTTP, enter the URL to Prometheus with port 9090, in my case: `http://192.168.178.25:9090`.

All fields under Auth can be left blank. Ensure that 'HTTP Method' is set to GET. Once this is all entered, click save and test. The following message should appear if everything is correct.



Now click on the squares on the left side of the screen. This should bring you back to the screen where you started. Click 'Create your first Dashboard'.

6. Creating a Dashboard.

Before creating a dashboard, please ensure that Prometheus and metric_INA219 are running on the Raspberry Pi. These can be ran as follows:

0.1 prometheus.

```
Prometheus: cd Documents/ AstronPowerMonitoring / AstronPowermonotoring/ Prometheus/ prometheus-2.5.0.linux-armv7/
./Prometheus -config.file=prometheus.yml
```

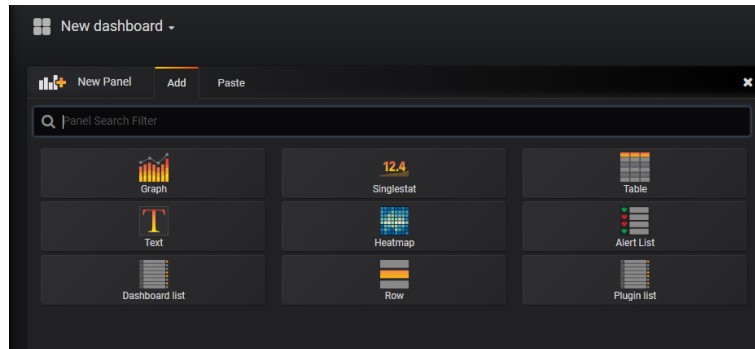
0.2 Metric_ina219:

```
cd Documents/ AstronPowermonotoring/CurrentMeasurement/ python3
./metric_INA219.py
```

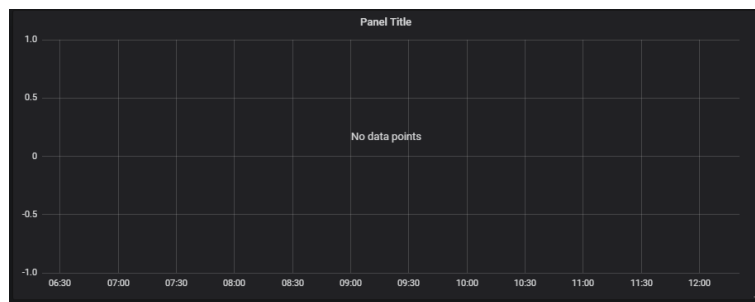
Another thing to note: The time on the raspberry Pi and the machine that runs Grafana-server have to be at the same time. If the raspberry pi is ahead in time of the machine that runs server, the server is going to query for data that the raspberry pi perceives to be as the future. If the machine is ahead, this will be the other way around.

This can usually be corrected by restarting the Raspberry pi which lets NTPDate connect to the internet.

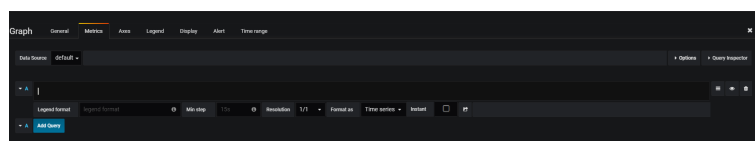
Now click 'Create your first Dashboard' The following screen should appear:



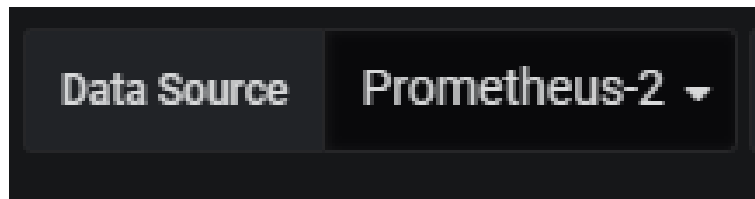
Our team uses the graph to plot the data. Click on graph and the following screen should appear:



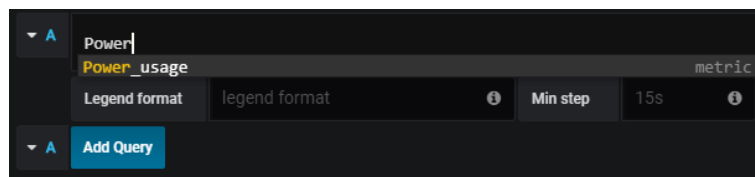
Hover over 'Panel Title' and a little arrow should appear on the right side. Click on this arrow and click 'Edit'. A new page is opened and several categories are displayed. It should open on the 'Metrics' tab as seen in the screenshot below:



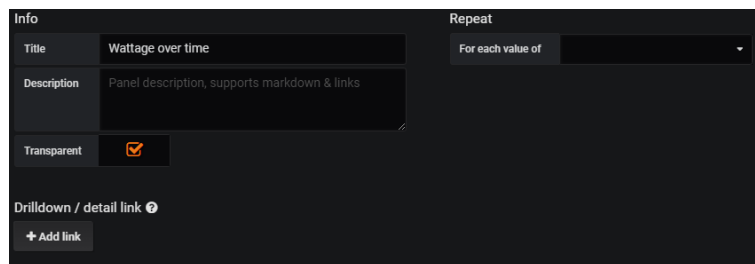
Most of the fields above are left blank. We are interested in plotting the power usage. This is stored in a variable called Power_usage. First set the datasource to the one we just created (Prometheus-2).



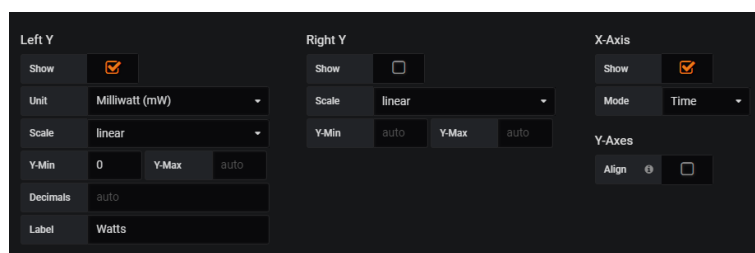
Enter Power_usage in the field below. It should autocomplete itself.



Now, under legend format, enter 'Wattage'.
Now change the tab to 'General' and change the title to Wattage over time.
Enter a description if desired.



Under axes, change the unit to Milliwatt(mW) and set the label to 'Watts'



Under legend, check the 'show' box and in the values field, enable average(avg) and current(Current).
Under alert, click 'create Alert'.
Change the name and evaluate every as desired. We choose the starting

Options	Values	Hide series
Show <input checked="" type="checkbox"/>	Min <input type="checkbox"/> Max <input type="checkbox"/>	With only nulls <input type="checkbox"/>
As Table <input type="checkbox"/>	Avg <input checked="" type="checkbox"/> Current <input checked="" type="checkbox"/>	With only zeros <input type="checkbox"/>
To the right <input type="checkbox"/>	Total <input type="checkbox"/> Decimals auto	

parameters. Under conditions, only enter a value in the is above field. Based on the stress testing done by our team, we chose a value of 3000.

Alert Config

Alert Config

Name

High Wattage Alert

Evaluate every

1m

For

5m

State history

Delete

Conditions

WHEN

avg ()

OF

query (B, 5m, now)

IS ABOVE

3000

+

If no data or all values are null

SET STATE TO

No Data

If execution error or timeout

SET STATE TO

Alerting

Test Rule

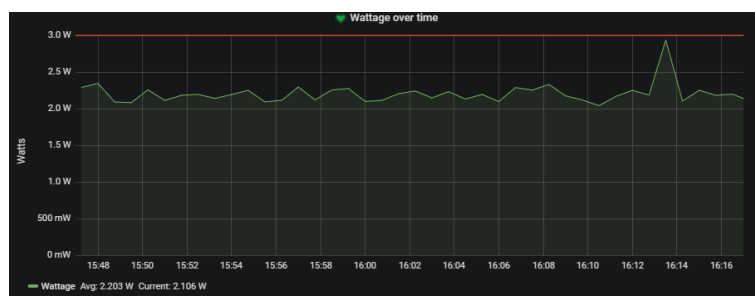
Now go to time range. Most of these fields can be left blank. The Time range can be changed in the top-right corner of the screen.

<input type="checkbox"/>	Override relative time	Last	1h
<input type="checkbox"/>	Add time shift	Amount	1h
<input type="checkbox"/>	Hide time override info	<input type="checkbox"/>	

The following button can now be used to change the time range:



The results in the graph seen below:



Repeat these steps for Voltage. The metric is called Voltage. In alert, set is above to 650(mV). The red line resembles the border that triggers an alert.



The result.

Please note that Grafana DOES NOT store data. It only visualizes incoming data from datasources. Grafana does have a temporary log, but this log stores data about Grafana.