Sensing and Actuation Networks and Systems [2022-2023]

Assignment 07 – Using Grafana for IoT data visualization

Introduction

This assignment introduces Grafana as a visualization tool to display and analyze IoT data.

Objectives

Students successfully concluding this work should be able to:

- Use Python to store data into a time series database (InfluxDB will be used)
- Visualize IoT data stored in an InfluxDB database in a Grafana dashboard

Support Material

- See Bibliography in the final section
- Python files: “grafana_data_generator.py” and “os-metric.py”

Requirements

- Internet access
- Knowledge of Python programming!
- InfluxDB Cloud free plan (https://www.influxdata.com/) – Registration required
- Grafana Cloud free plan (https://grafana.com) – Registration required
- Python3 and InfluxDB Python Client Library installed in every student machine

Note

- All the examples and exercises in this assignment can be done by students both in the University and at home.
- Warning: Grafana Cloud has some issues with Safari browser.

Background

Data visualization is an essential step in any data analysis process, especially when considering vast amounts of data as is the case of IoT systems.

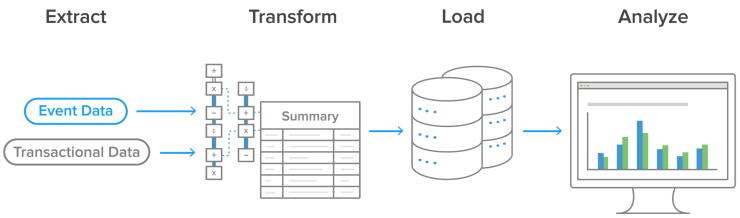


Fig. 1– ETL (Extract, Transform, Analyze) process (image source: https://www.stitchdata.com/etldatabase/etl-process/)

Grafana (https://grafana.com) is an analytic and interactive visualization tool that provides charts, graphs, queries, and alarms, supporting multiple data sources.

Fig. 2 - Grafana (from https://grafana.com/grafana/)

Note: To avoid time/date issues, the data saved in InfluxDB using the given Python script will be UTC.

Grafana tour

Before creating an account, play around with an online Grafana sandbox at https://play.grafana.org . Test the multiple graph options and discover the visualization possibilities that Grafana offers.

Note: to be able to edit a visualization panel, click on the panel menu (top right corner of the panel), or hover the mouse over the panel and press "e".

Set-up - Creating Grafana Cloud account

To avoid having to install Grafana in our own machines, a Grafana Cloud free account will be used (https://grafana.com). Fig. 3 and Fig. 4 show the restrictions and sign up screen of a Grafana Cloud free account, respectively.

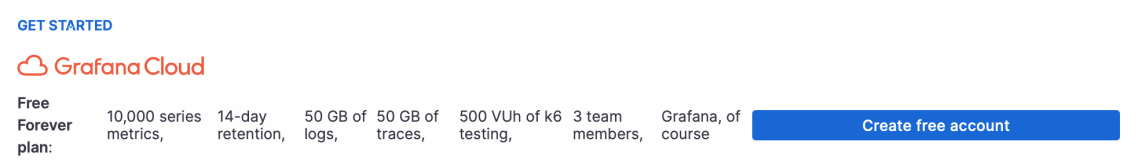


Fig. 3 – Grafana Cloud free account

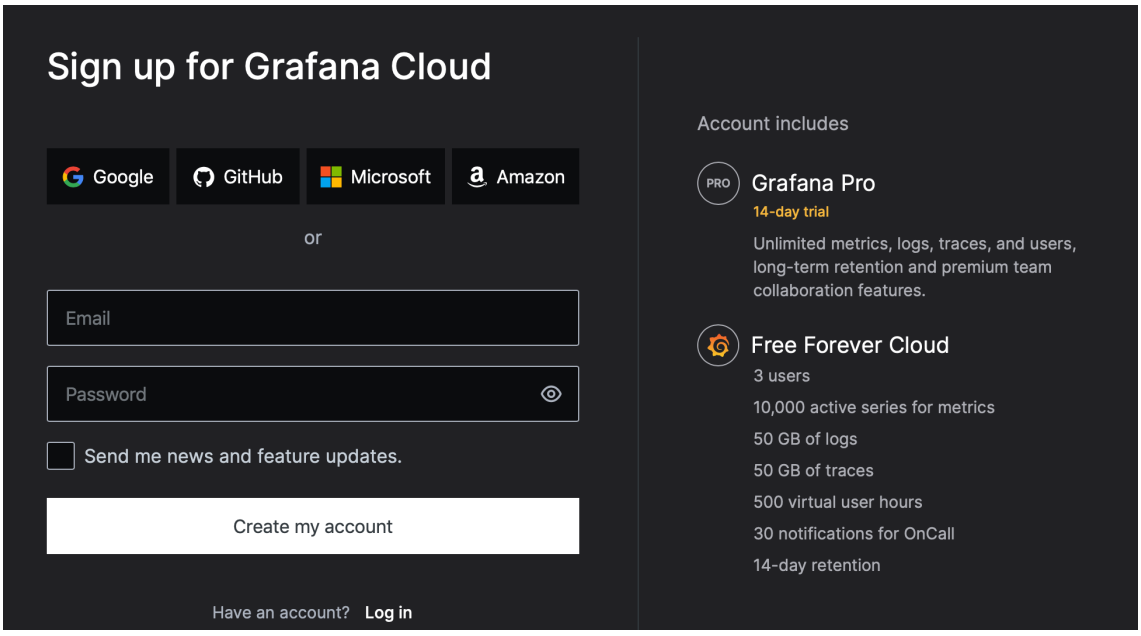


Fig. 4 – Grafana Cloud sign up screen

You are now ready to start.

A first Grafana panel

This exercise has two parts: the generation of dummy data to be visualized, and the creation of charts and graphs in Grafana.

Dummy temperature data generator

Before starting the data generation, an InfluxDB Cloud account is necessary. The instructions to create and use an InfluxDB Cloud database were addressed in Assignment 6. Create a bucket named "SRSA" and also a new measurement named "Dummy_temp_sensor".

Analyse the program "grafana_data_generator.py" given with this Assignment. The file will generate dummy temperature data every 2 seconds and send it to a InfluxDB Cloud database. Change the code to use the credentials and token of your InfluxDB Cloud account and database, and test it in a terminal window (Fig. 5).

```
SRSA-Server$./grafana_data_generator.py
Value inserted: device1 temperature= 19.587811341520226
Value inserted: device1 temperature= 19.91807444555269
Value inserted: device1 temperature= 19.473478297073616
Value inserted: device1 temperature= 19.333916769399565
```

Fig. 5 – Generation of dummy temperature data

To verify that the data is reaching the database, use your InfluxDB Cloud account, access the left menu and choose "Data Explorer". Check if the values are being inserted using the following command:

```
SELECT * FROM '' Dummy_temp_sensor''
```

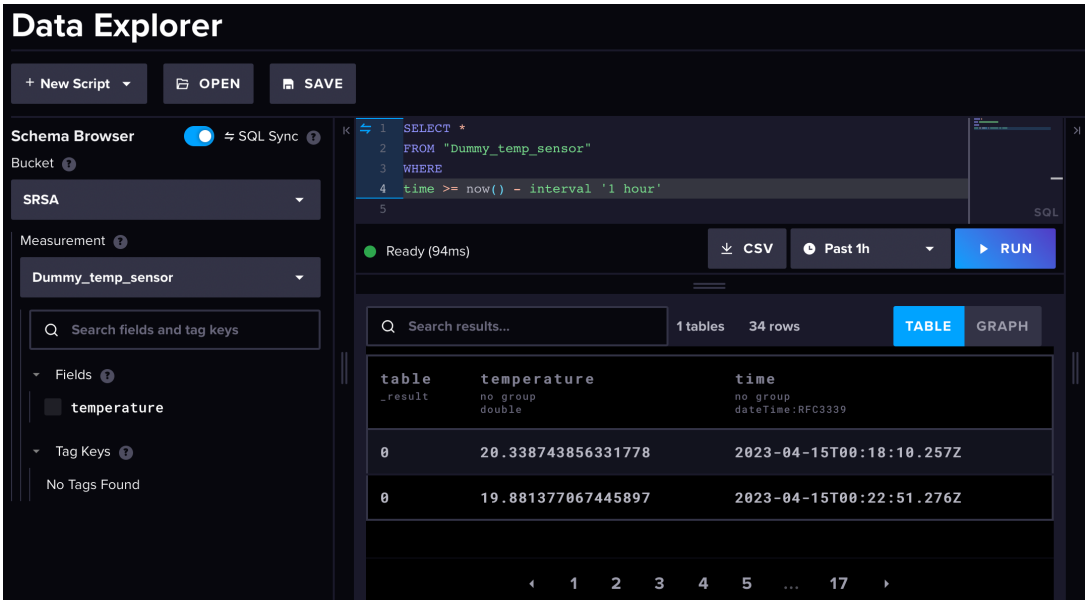


Fig. 6 - Data in InfluxDB Cloud database

After confirming that the data is reaching the database as planned, stop the process by pressing Ctrl + C in the terminal windows where it is running.

You are now ready to go to Grafana.

A first Grafana visualization

Launch Grafana from Grafana Cloud Portal using your previously created account (Fig. 7).

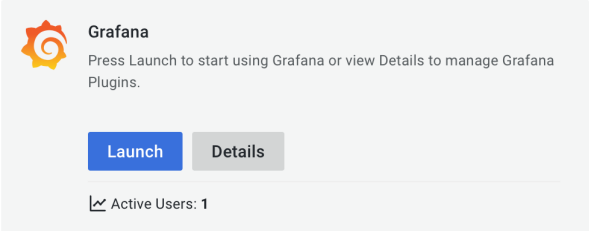


Fig. 7 – Launch Grafana from Grafana Cloud Portal

Create a new datasource

To use the previously created InfluxDB as the data source for Grafana, click on Grafana menu and select "Administration-> Data Sources".

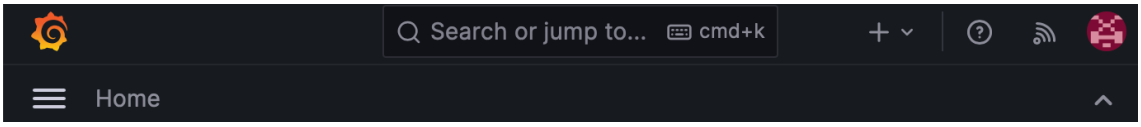


Fig. 8 – Grafana top bar (menu button on the left)

Click on "Add new data source" button and then select "InfluxDB" as the new data source type (Fig. 9).

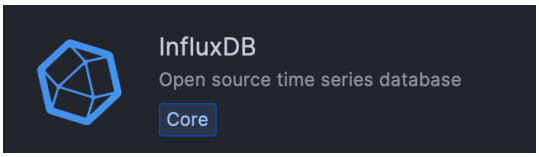


Fig. 9 – InfluxDB data source type in Grafana

A configuration panel will open where the specifics of your InfluxDB Cloud account and database must be specified.

- Change the "Name" field to "InfluxDB-SRSA".
- Select "Flux" as the Query Language.
- Change the URL to the one specified in your InfluxDB Cloud account.
- Change the "Basic Auth Details" with your InfluxDB Cloud account login info.
- Insert all the necessary information on the "InfluxDB Details": "Organization", "Token", "Default Bucket". All this information is available from your InfluxDB Cloud account.
- Change the "Min time interval" to "2s" to match the writing speed of the previously used Python script ("grafana_data_generator.py").

Click on “Save & test” on the bottom of the page. If everything is correct, a green check mark will appear (Fig. 10).

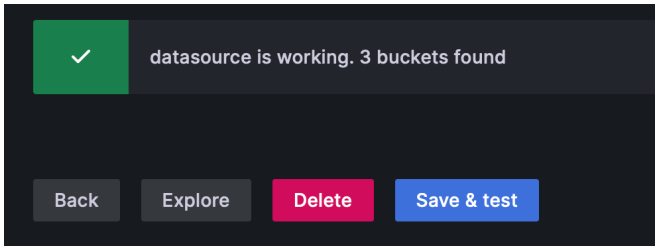


Fig. 10 – A green check mark indicates a successful connection to the InfluxDB account

To explore the data from the new data source created, click on the “Explore” button (optional step).

Create a new dashboard

The dashboard is a page that presents a set of selected information in a graphical way. Each dashboard may have different panes and rows. A pane contains the visualization of data from specific queries, and each row groups visualizations into expandable sections.

Fig. 11 – Example of multiple visualizations and rows in Grafana (from https://rudimartinsen.com)

Create Panel 1

Click on Grafana menu and select “Dashboards”. Next, click on the “New” button and select “New Dashboard”. To add a new visualization, click on “+ Add visualization”.

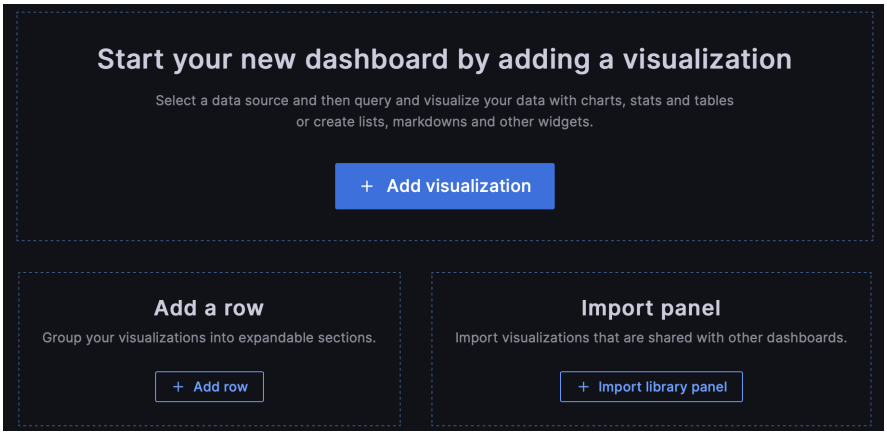


Fig. 12 – Creation of a Grafana visualization

You can now specify the data to be used in this specific visualization panel, and the type and properties of the visualization object (Fig. 13).

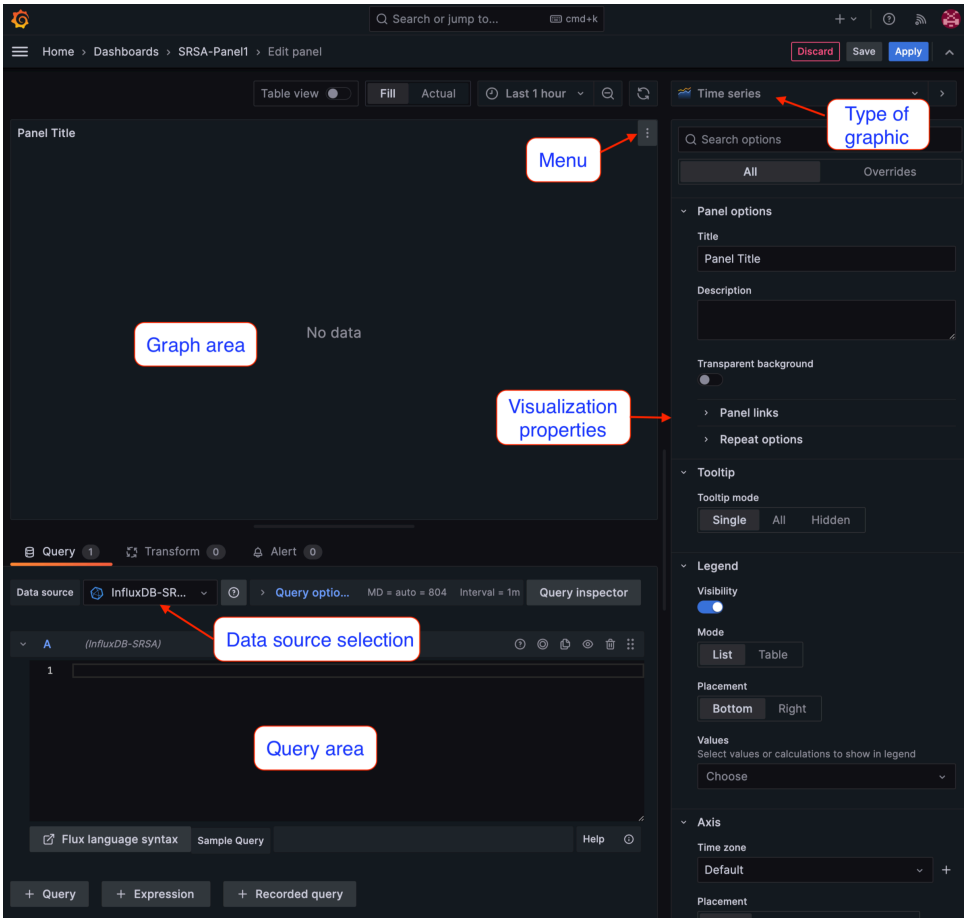


Fig. 13 – Visualization screen

The column on the right allows for the configuration of how the data will be displayed (e.g., titles, legends, colors). At the top of the column, the type of graphic can be selected. In the lower panel, the information to be used can be selected, using a query that gets information from a specific Data source.

For this example, you will use the data previously generated by file "grafana_data_generator.py", whose values were saved in your InfluxDB Cloud account.

Choose the "Data source" previously configured on Grafana and, on the Query area, select the data to be used using the following query:

```
from(bucket: "SRSA")
  |> range(start: -1h)
  |> filter(fn: (r) => (r._measurement == "Dummy_temp_sensor") and (r._field ==
"temperature"))
  |> yield(name: "_results")
```

This code, written in Flux, InfluxData's functional data scripting language, will select data saved on bucket "SRSA" on the last 1h, where the measurement has the name `"Dummy_temp_sensor"`, and only for the field called "temperature". The query results will have the name "_results".

As the python script saved data in InfluxDB uses UTC time, and Grafana is using Portuguese time by default, in "Query options" set "Time shift" to 0h.

Fig. 14 – Selecting the data for the visualization panel

Click on the graph area to display a graphic with the available data. If only a few data points exist, run the initial "grafana_data_generator.py" script for additional values. On the top of the graph area select the time range to be displayed.

Use the right column to change the type and details of the graphic displayed:

- Change "Title" to "Temperature";
- In the "Legend" display the last, max and min value;
- In the "Axis" "Label" use "ºC";
- In the "Standard options" change the unit to Celsius degrees;
- Set a "Value mappings" named "Base" at 20ºC.
- Create 3 new "Thresholds", at 18 (blue), 20 (green) and 22 (red), and show them "As lines (dashed)".

Next, a moving average will be added to the temperature chart.

The moving average will be calculated by a new "Query B". Start by duplicating "Query A" using the icon in Fig. 15 and change it to match the code in Fig. 16. This query will produce a moving average of 30 periods.

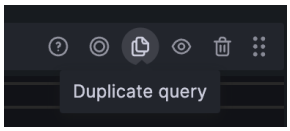


Fig. 15 – Duplication of a query

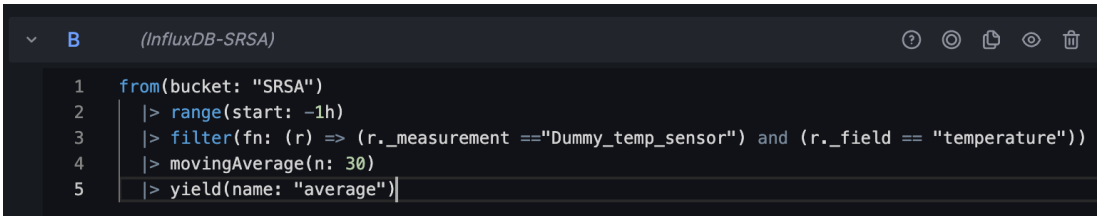


Fig. 16 – Query B code

To make the moving average line more visible in the chart, and easily distinguishable from the underlying temperature data, its color must be changed. To make it possible the color property of the moving average line must be override from its default color.

In the right panel, choose the “Overrides” tab and change it according to Fig. 17.

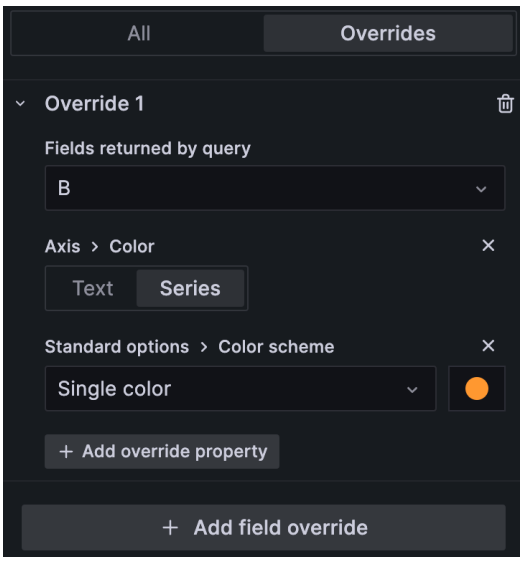
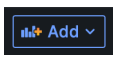


Fig. 17 – Overriding a specific query result

Save the dashboard as “SRSA-Dash1”

Create Panel 2

Click “Add”, “Visualization” to create another visualization panel. 

This time, using the same data source, create a graphic of type “Stat” where the last temperature value is shown.

Change the following settings:

- In “Value options” set the “Calculation” field to “Last”;
- In “Stat styles” set “Graph mode” to “None”;

- In the “Standard options” change the unit to Celsius degrees and the “Color Scheme” to “From thresholds (by value)”;
- Set 3 thresholds: blue for values lower than 18 (“Base”), green for values above or equal to 18, and red for values above or equal 22.

Final settings and notes:

- In the top bar of the Dashboard, change the refresh interval to 5s;
- You can resize or move any panel;
- To edit a panel just put the mouse over it and press key “e” or use the panel menu;
- Test your queries before using them in visualizations; go to Grafana Menu and select “Explore”.

Fig. 18 – Final dashboard

Exercise 1

Using the previous example as reference, create a new data source that sends temperature, humidity and pressure values every 5 seconds. Call this measurement "Dummy_sensor1". Design a new dashboard in Grafana to be able to visualize the values. The result should look like Fig. 19.

Fig. 19 – Final exercise dashboard

Exercise 2

Using the file "os_metrics.py" as base, create a program that sends both your system available RAM and the average CPU usage in the last 3 seconds, to your Influx database. Visualize the data collected in Grafana.

Bibliography

- Grafana Tutorials
  - https://grafana.com/tutorials/
- Grafana Cloud documentation
  - https://grafana.com/docs/grafana-cloud/
- Flux
  - https://docs.influxdata.com/influxdb/v2.7/query-data/get-started/