Cloud Data Processing with NIG

1. NIG Stack



The NIG stack consists of three software packages:

- NodeRed
- InfluxDB
- Grafana

In the IoT world, this is a commonly used combination of software packages. All three are also (mostly) open-source. You can run this stack locally or in the cloud. Because it would be too much for this workshop series, we will not show you how to set this up. But if you want to get this stack up and running yourself, here are some steps that might help you. We encourage you to do some research on this yourself.

1. If you are on Linux, you can forward to step 2. If you are on Windows we recommend installing the Windows-Subsystem-for-Linux (WSL) with Ubuntu as the distribution.
2. Now that we are in our Linux (testing done in Ubuntu) environment, we can install docker. You can follow the steps on the official website: Install Docker Ubuntu. If you are on WSL, every time you start the system you need to restart docker with: `sudo service docker start`
3. Download NIG docker files here:

   Download local NIG files
4. Unpack the files into an empty folder
5. Open your WSL instance/Terminal and navigate to that folder. If you are using WSL you can find all the windows files under “/mnt/c/”.
6. `docker compose up`
7. Open the individual application locally:
   1. NodeRed
   2. Grafana
   3. Influx

2. NodeRed

Node-RED is a UI programming tool for writing code with just graphical blocks. With this, you can easily wire together hardware devices with APIs and other online services. Node-RED is all about flows. A flow is really what it says it is. The data flows from an entry point (like MQTT or HTTP) to an endpoint. On the way, you can modify, compare it with other data, or do whatever you want to do with it. The endpoint could also be an outgoing MQTT-Message or HTTP-Request. In our case, you will

probably want to save it permanently in the given influx database and later display it in some way (that is what grafana is for).

- Documentation (with a getting started and some tutorials)

1. Log in with the username `nodered` and password `nodered`. You will now see an empty flow.
2. We have to install some special nodes (nodes are the blocks you use for programming) for interacting with the influx database. Click on the burger icon in the top right corner. You will find an option named “Manage palette”, click it. In the palette, you can manage all the plugins you currently have installed. You will now see all currently installed nodes/plugins.
3. Click on “Install” in the top bar and search for “influxdb”. You will find one named “node-red-contrib-influxdb”. Click on the “Install”-button to install it.
4. You can close the palette now. On the left side of your screen, you can find all nodes that are available to you. Scroll down until you find a node called “influxdb out”. You can also search for it in the top left corner. When you find it just drag it somewhere into your flow.
5. Now double-click on the node you just dropped. Because you never used it you will have to connect to a new server. Just click on the little pen on the right side of “Server”. Here you will have to type in all needed information. These are the same, never mind the group you are in.
    - Version: 2.0
    - URL: `http://influxdb:8086`
    - Token: `influx` (!This is just for demonstration purposes, in production, you should generate a custom token in the InfluxDB UI!)
    - Uncheck “Verify Server Certificate”
    - Organisation: `unicaes`
    - Bucket: `workshop`
    - Measurement: Makeup something nice - this is where you will find your data
6. Click on save. The database is now saved in node-red and you can use it on every influxdb-node you want to. If you want to, you can delete the node we just created.
7. Drag in a “mqtt in”-node and connect it to our just created influx node. Double-click to open its settings.
8. Create a new mqtt server connection by clicking the little pen icon again.
9. Here you just need to fill in Server with `broker.hivemq.com` and click Save in the top right corner.
10. In the topic field you need to fill in the `outTopic` you defined earlier in the Arduino sketch. Also, set the Output to `Parsed JSON-Object`. Then you can Save the current settings.
11. All your sensor data should now be saved in the influx database 😀

3. InfluxDB

Influxdb is a time series database that is really good designed for IoT applications. It can handle very high write and read loads. It's very easy to use because the used InfluxQL query language is somewhat close to SQL. But you can learn more about it here:

- InfluxDB
- Some Guides - Documentation

1. Open up your Influx instance. You can log in with the username `influx` and the password `influxdb`
2. On the left-hand side you will find a sidebar. Click on the little graph symbol which is the Data

Explorer
3. The Data Explorer can be used to review your data and help with developing new database queries. For this tutorial it is very easy to use:
    1. In the FROM column you need to select the bucket which you want to read out. In this case `workshop`
    2. Then in the second column you can select the measurement that you want to read out
    3. As we are saving only one simple number in the database, you must select `value` in the last column
    4. Click SUBMIT to have a look at your data in a nice graph
4. For use in Grafana we will now need to copy the actual query that we just built with the help of the handy query builder. To get this query you need to click `SCRIPT EDITOR`. Copy the complete query that is now shown. We will need it later.

4. Grafana

Grafana describes itself as the

> The analytics platform for all your metrics

With grafana, you can query the data from the database and show it off in graphs. You can also create alerts. So when you want to get a notifications when let's say a value gets above a certain threshold, grafana can do that for you. You can find more information here:

- What is Grafana?
- Getting started

1. Open up your Grafana instance. You can log in with the username `grafana` and the password `grafana`
2. Then before we can get started with creating our nice graphs, we need to create a data source. Of course, our data source will be our influx database. Open the burger menu on the left-hand side. As one of the last options you will find the `Connections`. Click that.
3. It will now ask you to create a new connection. So search for `Influx` and click the upcoming `InfluxDB` option. In the top right corner, you can then select `Add new data source`.
4. As a query language we want to select `Flux`
5. Now we need to fill in some of the information we used before in NodeRed as well. Just for documentation here they are again:
    1. HTTP - URL: `http://influxdb:8086`
    2. InfluxDB details:
        1. Organisation: `unicaes`
        2. Token: `influx` (!This is just for demonstration purposes, in production, you should generate a custom token in the InfluxDB UI!)
        3. Default bucket: `workshop`
6. Click `Save & test`. There should be no error and we can continue with creating our first graph. Grafana should give you a little success message. In this message, you can click `building a dashboard`.
7. Before we start, lets make sure to save our Dashboard first. For that just click the Save icon in the top bar. Give it a nice name and hit Save.
8. After doing that, we can now finally click the big blue button `+ Add visualization`. In the upcoming pop-up, we need to choose the just-created data source.
9. Now we are in the UI where we can create our graph. At the bottom of the screen, you can see

a text box. In this text box, we need to paste in the query we created earlier. If you click out of the text box this should already show us our graph!!

10. The next steps are pretty self-explanatory. First, you can for example change the name of the graph. All of the options for modifying the

Recording

Additional Links

The NIG stack was previously mentioned here:

- Introduction to IoT - Sensors and Data processing
- First steps with your NIG

Unfortunately, these guides are **not** up to date anymore, as they use an older version of the InfluxDB. Since Version 2 of that database, some details in the process have changed a bit and there is slighty more work to it now.

From:
https://wiki.eolab.de/ - **HSRW EOLab Wiki**

Permanent link:
https://wiki.eolab.de/doku.php?id=latinet:unicaes:workshops:nig&rev=1693593072

Last update: **2023/09/01 20:31**