# Node-RED, InfluxDB and Grafana for IoT

Sebastian Büttrich,
IT University of Copenhagen
ICTP Trieste, 20180502
sebastian@itu.dk

# Motivation

**To get from**

- **sensor node** **(lopy or any other)**

**to**

- **managed data flows** **(Node-RED)**
- **storage** **(InfluxDB)**
- **visualization / analysis / interaction** **(Grafana)**



# Agenda for this talk

- **Introduce the three elements**
- **Show them in live demo**

**and - for those who are hungry -**

**Offer you a guide on installing these yourself.**

**Two things i will be using**

**though we will only cover them later this week:**

- **MQTT**
- **The Things Network**

# MQTT

**MQTT** is a

**lean and fast**

**publish/subscribe**

**messaging protocol**

running on top of **TCP/IP**

**(more tomorrow!)**

# The Things Network /1

- **The Things Network is a global movement, a kind of "The people's Internet of Things".**
- Open source
- Free ... to set up and run their own, in particular:

  Anyone who perpetrates a "Things Access" or a "Things Gateway" will do so **free of charge for all connecting devices and servers.**

  *This to some degree explains our current interest in TTN, in an educational context.*

Source, Details:
https://github.com/TheThingsNetwork/Manifest

# The Things Network /2

- **The Things Network is a global movement, a kind of "The people's Internet of Things".**
- It's one possible choice when doing LoRaWan.
- It's Open source, you are free ... to set up and run their own, in particular:

  Anyone who perpetrates a "Things Access" or a "Things Gateway" will do so **free of charge for all connecting devices and servers.**

Source, Details:
https://github.com/TheThingsNetwork/Manifest

# The Things Network /3

# Overview /1


THE THINGS NETWORK

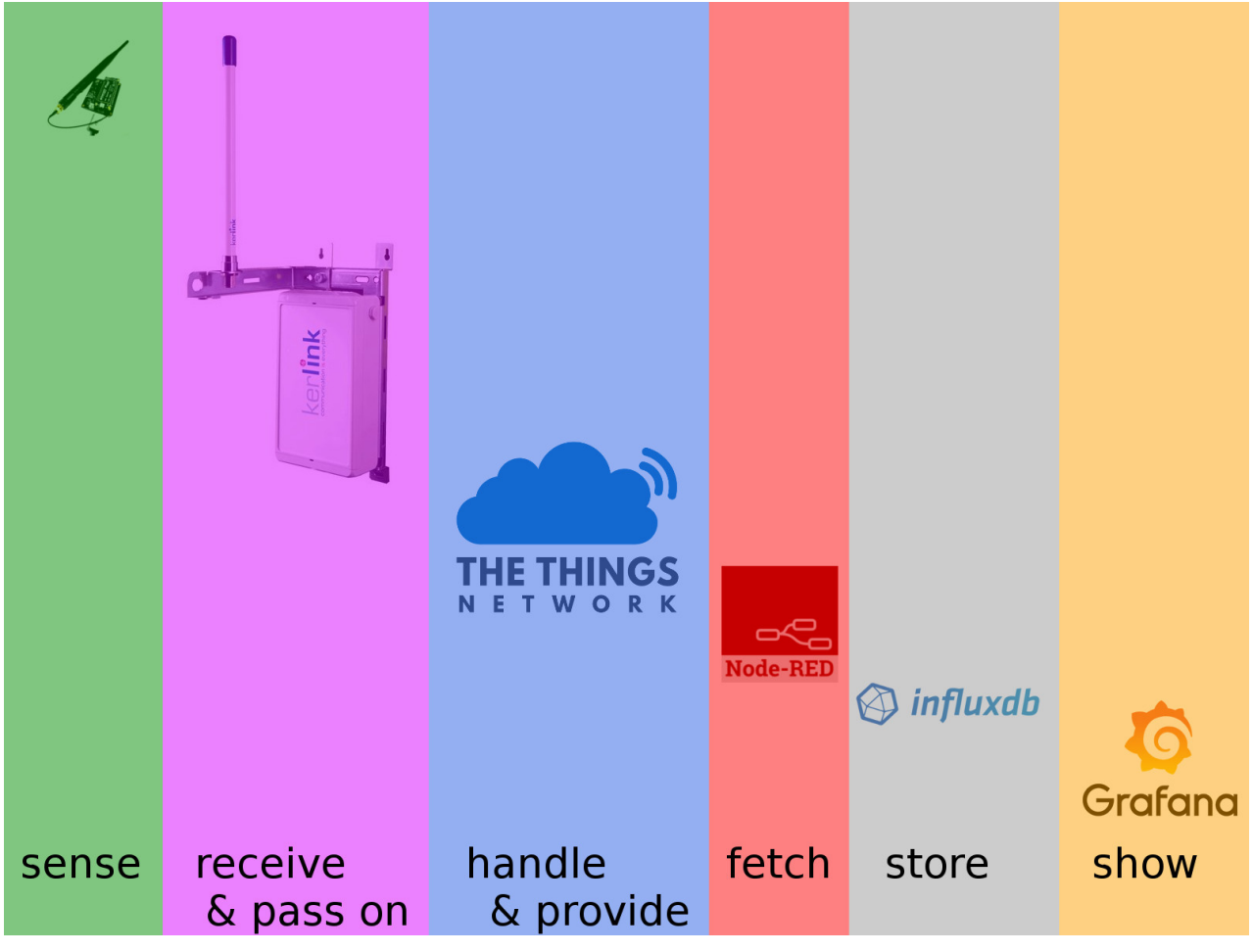
Node-RED

influxdb

Grafana

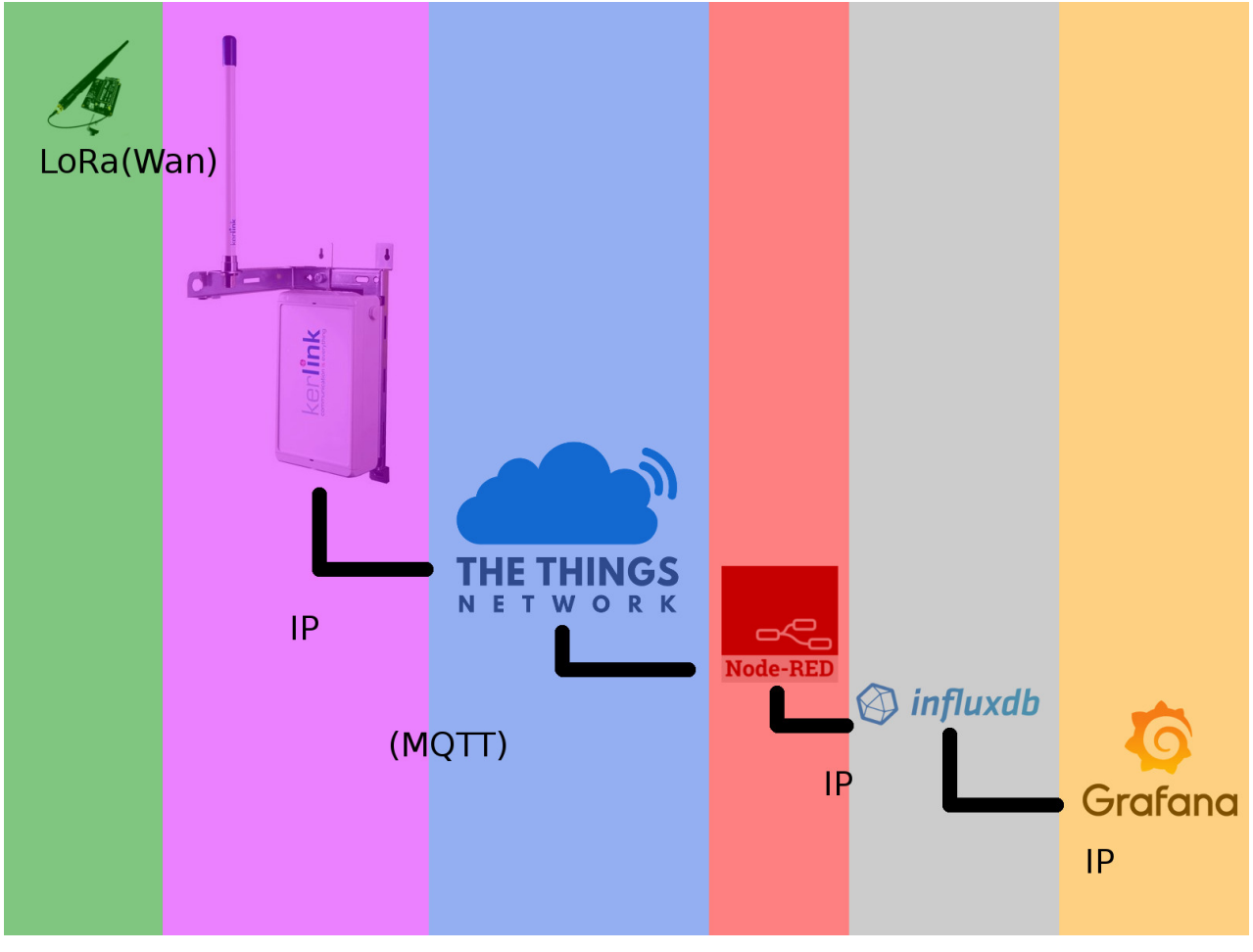
# Overview /2

## Roles



# Overview /3

## Network



# Overview /4

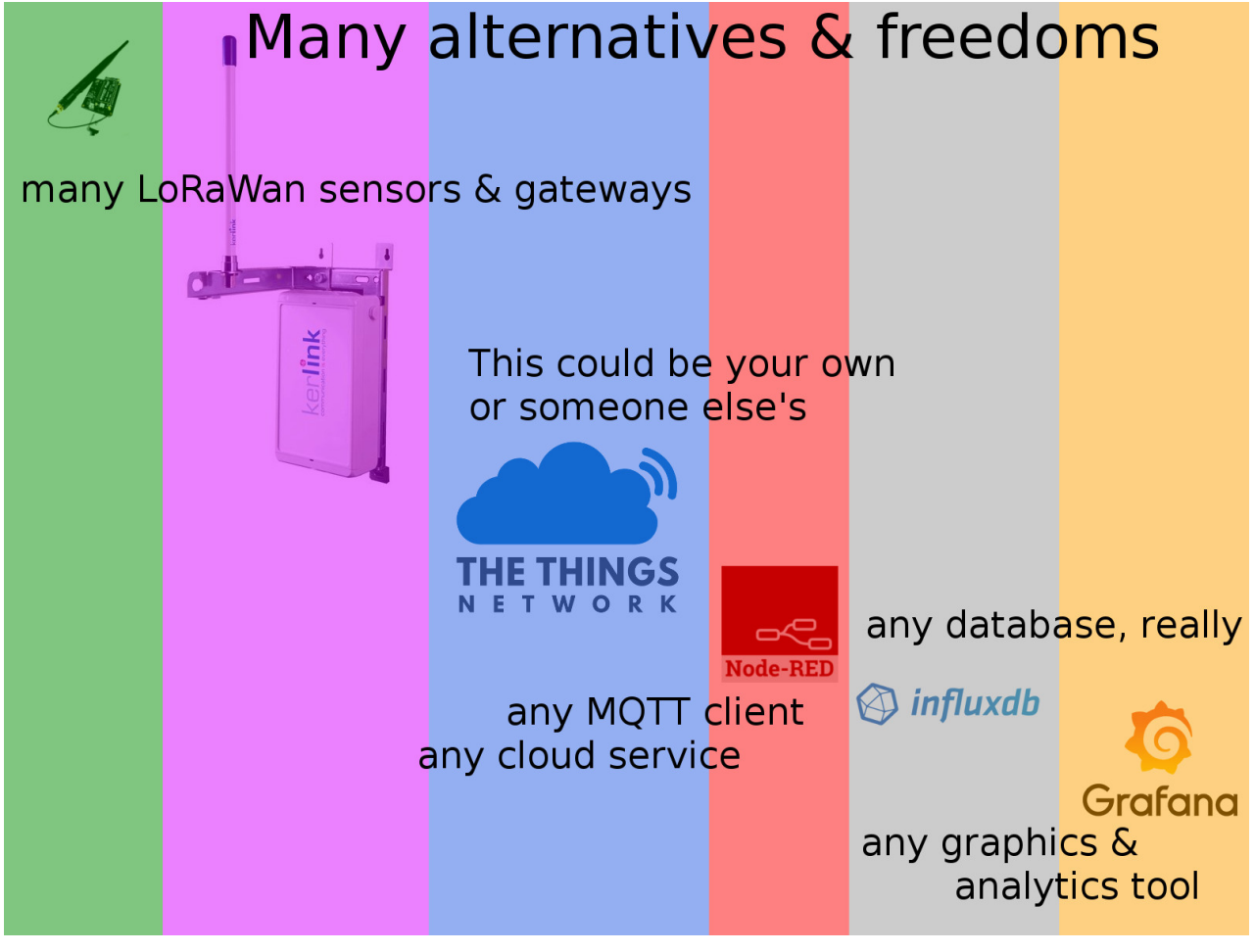
## Locations

# Overview /5

**Freedom**

**Of**

**Choice**



# Node-RED /1

Node-RED is a

**graphical tool for wiring together**

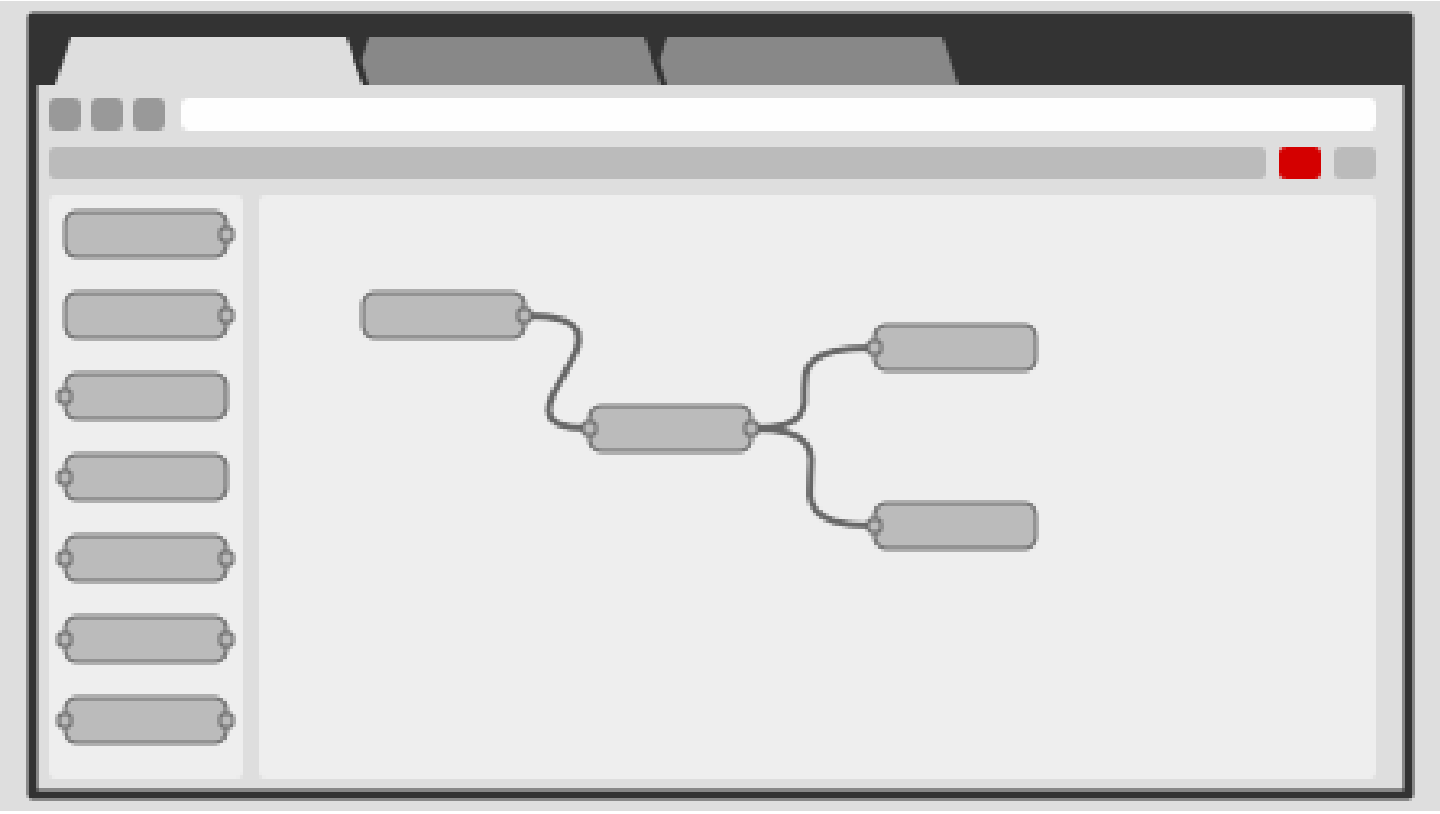
**hardware devices, APIs and online services**

in new and interesting ways.


It provides a **browser-based editor** that makes it easy to wire together **flows** using the wide range of **nodes** in the **palette** that can be deployed to its runtime in a single-click.

Built on **Node.js**

# Node-RED /2 principle



**nodes** ..... **flows**

# Node-RED /3

Examples of existing nodes:

**Input/Output:** tcp, udp, http, mqtt, ttn, ...

debug, status, inject, link, trigger

**Functions:** logic, analytics

**Storage:** e.g. databases

**Social:** e.g. tweet, mail

You can write your own nodes!

*It is a bit like a DJs patchbay.*

# Node-RED /4 node config

Nodes are configured by double-clicking and editing the necessary info,

e.g.

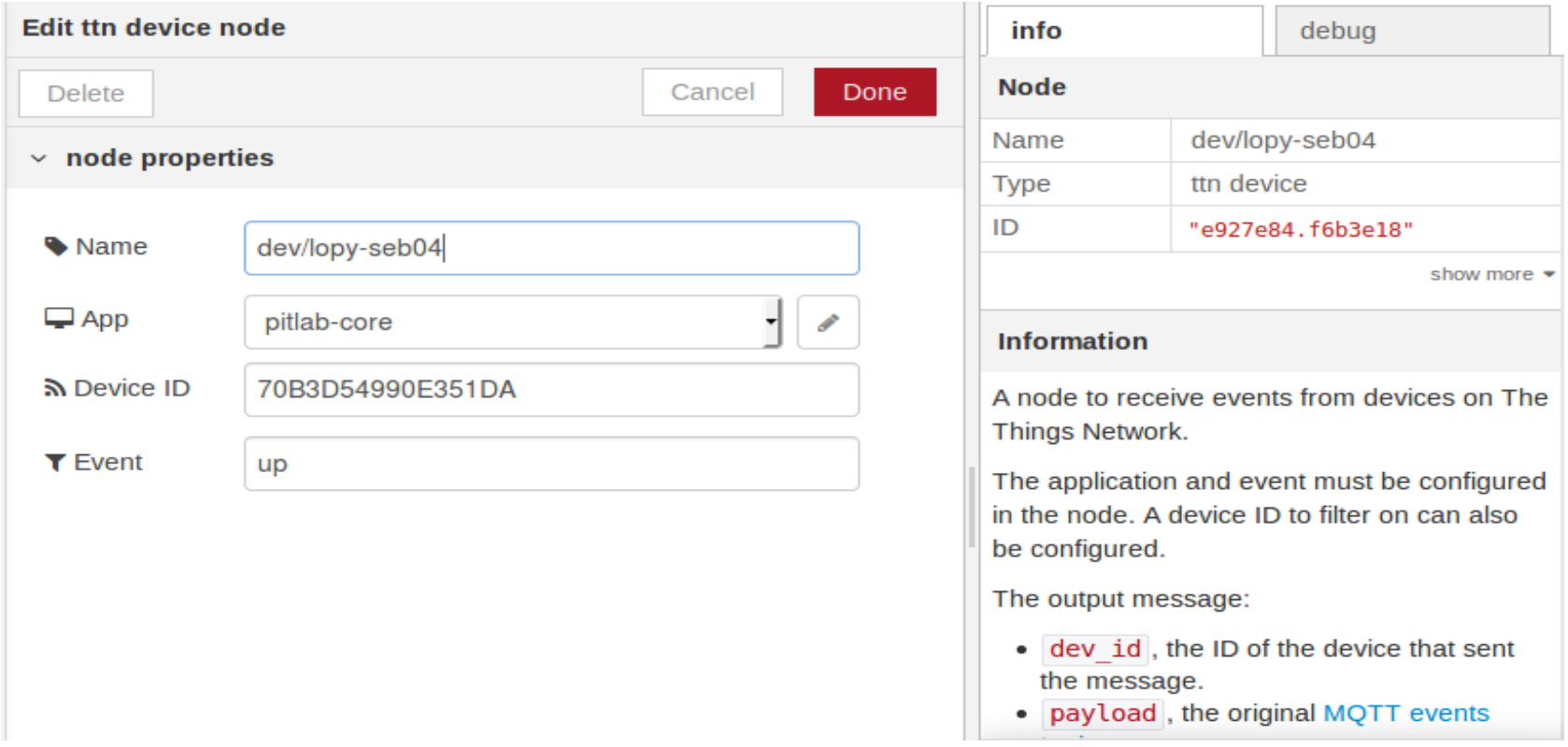
**MQTT** topics,

**http** URLs,

**TheThingsNetwork** applications and security keys

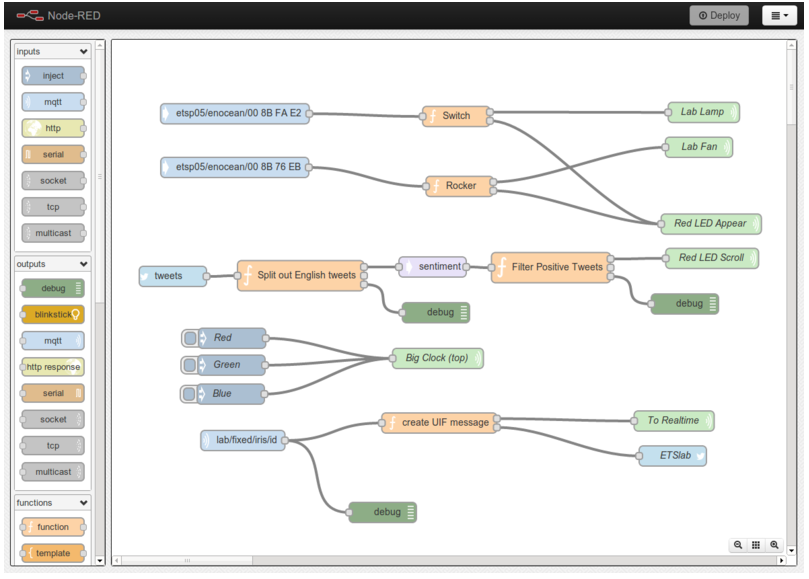
# Node-RED /5 node config TTN

Example of a TTN node:



# Node-RED /6 flows

**Flows are combinations of inputs, outputs, connections, Actions, etc which you can share and reuse.**




**e.g. receive sensor messages, do calculations on the values, keep averages or deltas, put them in a database, trigger an actuator, and inform the owner via messages.**

# Node-RED /7 text

**While this is a graphical tool,**

**behind the scenes it s all text files**


```
[{"id":"e7fb9fb.21ef46","type":"tab","label":"Seb's data flows","disabled":false,"info":""},{"id":"fe383a2e.538d78","type":"ttn app","z":"","appId":"pitlab-
ds18b20","region":"eu","accessKey":"ttn-account-v2.D7-uy5QHbEvbjOv65hV_DI4chfWPAC0tWTGNQoiGYno"},{"id":"7714594.09e8ca8","type":"ttn app","z":"","appId":"pitlab-
ds18b20","region":"eu","accessKey":"ttn-account-v2.D7-uy5QHbEvbjOv65hV_DI4chfWPAC0tWTGNQoiGYno"},
{"id":"2e1a5f7.ecc9aa","type":"influxdb","z":"","hostname":"127.0.0.1","port":"8086","protocol":"http","database":"pit001","name":"","usetls":false,"tls":""},
{"id":"132c2fb3.7d3c","type":"ttn app","z":"e7fb9fb.21ef46","appId":"","region":"","accessKey":""},{"id":"4e72e8d2.31e58","type":"ttn app","z":"","appId":"pitlab-
core","region":"eu","accessKey":"ttn-account-v2.XLXDZg4cTdT0X-4obr4F3A5StKGlt49VeqBMQl-QIRo"},{"id":"c433206c.cb88b","type":"mqtt-
broker","z":"","broker":"influx.itu.dk","port":"1883","clientid":"","usetls":false,"compatmode":true,"keepalive":"60","cleansession":true,"willTopic":"","willQos":"0","willPa
{"id":"7ddf9c9c.690d7c","type":"ttn app","z":"","appId":"pitlab-test-seb-201804","region":"eu","accessKey":"ttn-account-v2.uNQlc1iO2r441nJVyb9fgkVPEgNllj8OwN8tzY8Lrmg"},
{"id":"bbe92aff.b8c38","type":"ttn app","z":"","appId":"pitlab-seb-temperature-humidity","region":"eu","accessKey":"ttn-account-
v2.d1HGrFbctUyngVrcXcO_D8V2Qw4meJ_M7VC4B9qk5Kg"},{"id":"ebf65f2f.3c1748","type":"ttn app","z":"","appId":"dk-cph-itu-pitlab-01","region":"eu","accessKey":"ttn-account-
v2.2ohV653kNiCAOPsmOf9u9KIBmAjJP3Wq62Ywv6fYzdM"},{"id":"e927e84.f6b3e18","type":"ttn device","z":"e7fb9fb.21ef46","name":"dev/lopy-
seb04","app":"4e72e8d2.31e58","dev_id":"70B3D54990E351DA","event":"up","x":96.53570556640625,"y":715.321533203125,"wires":[["71231f80.fdba38"]]},
{"id":"6e482785.6e1468","type":"influxdb
out","z":"e7fb9fb.21ef46","influxdb":"2e1a5f7.ecc9aa","name":"","measurement":"tindiestick","precision":"","retentionPolicy":"","x":1180,"y":40,"wires":[]},
{"id":"d6ac24de.c86878","type":"ttn message","z":"e7fb9fb.21ef46","name":"msg/app:dk-cph-itu-
pitlab-01","app":"ebf65f2f.3c1748","dev_id":"","field":"","x":129,"y":31,"wires":[["5ff3781c.a6152","2d37e02c.830c28","1db6ebe3.327dec"]]},
{"id":"2d37e02c.830c28","type":"function","z":"e7fb9fb.21ef46","name":"function 3 - 3 sticks and waterTemp","func":"node.log (\"function 3 called\")\nvar msgAll =
{ payload: msg.payload};\nvar msg0 = { payload :msg.payload[0] };\nvar msg1 = { payload :msg.payload[1] };\nvar msg2 = { payload: msg.payload[2] };\nvar msg3 = { payload:
msg.payload[3] };\nvar msg4 = { payload: msg.payload[4] };\nvar msg5 = { payload: msg.payload[5] };\nvar msg6 = { payload: msg.payload[6] };\nvar msg7 = { payload:
msg.payload[7] };\nvar msg8 = { payload: msg.payload[8] };\nvar msg9 = { payload: msg.payload[9] };\nvar msg10 = { payload: msg.payload[10] };\nvar msg11 = { payload:
msg.payload[11] };\nvar msg12 = { payload: msg.payload[12] };\nreturn [ msg0, msg1, msg2, msg3, msg4, msg5, msg6, msg7,msg8, msg9, msg10,msg11, msg12 ];
\n","outputs":"13","noerr":0,"x":462.857177734375,"y":145.71426391601562,"wires":[["109102f9.a35295","f464746b.2c73c"],["f6162812.22b968"],["2ce784ac.12558c"],
["195b976d.dd29e9"],["6f89d82d.af84b8"],["12d1fc81.2209a3"],["62791b96.df9c84"],["863f73e.2f8189"],["10afd128.2aabcf"],["a23ae5fd.b2d2c8"],["5e7d504e.3905e8"],
["8c5135df.f6a148"],["f4314019.a81c58"]]},{"id":"687145e1.4c21bc","type":"influxdb
out","z":"e7fb9fb.21ef46","influxdb":"2e1a5f7.ecc9aa","name":"","measurement":"lopySeb02","precision":"","retentionPolicy":"","x":1180,"y":80,"wires":[]},
{"id":"5ff3781c.a6152","type":"function","z":"e7fb9fb.21ef46","name":"function 04","func":"node.log (\"function 04 called\")\nvar msg1 = {payload: msg.payload[2]};\nreturn
[ msg1];\n","outputs":"1","noerr":0,"x":390.0000305175781,"y":24.28571319580078,"wires":[[]]},{"id":"f6162812.22b968","type":"influxdb
out","z":"e7fb9fb.21ef46","influxdb":"2e1a5f7.ecc9aa","name":"","measurement":"003-01-moist","precision":"","retentionPolicy":"","x":1272,"y":182,"wires":[]},
{"id":"109102f9.a35295","type":"influxdb out","z":"e7fb9fb.21ef46","influxdb":"2e1a5f7.ecc9aa","name":"","measurement":"003-01-
temp","precision":"","retentionPolicy":"","x":1173,"y":142,"wires":[]},{"id":"2ce784ac.12558c","type":"influxdb
out","z":"e7fb9fb.21ef46","influxdb":"2e1a5f7.ecc9aa","name":"","measurement":"003-01-light","precision":"","retentionPolicy":"","x":1257,"y":224,"wires":[]},
{"id":"f4314019.a81c58","type":"influxdb out","z":"e7fb9fb.21ef46","influxdb":"2e1a5f7.ecc9aa","name":"","measurement":"003-05-
temp","precision":"","retentionPolicy":"","x":1264,"y":622,"wires":[]},{"id":"12d1fc81.2209a3","type":"influxdb
out","z":"e7fb9fb.21ef46","influxdb":"2e1a5f7.ecc9aa","name":"","measurement":"003-02-light","precision":"","retentionPolicy":"","x":1257,"y":344,"wires":[]},
{"id":"6f89d82d.af84b8","type":"influxdb out","z":"e7fb9fb.21ef46","influxdb":"2e1a5f7.ecc9aa","name":"","measurement":"003-02-
moist","precision":"","retentionPolicy":"","x":1264,"y":302,"wires":[]},{"id":"195b976d.dd29e9","type":"influxdb
out","z":"e7fb9fb.21ef46","influxdb":"2e1a5f7.ecc9aa","name":"","measurement":"003-02-temp","precision":"","retentionPolicy":"","x":1267,"y":264,"wires":[]},
{"id":"10afd128.2aabcf","type":"influxdb out","z":"e7fb9fb.21ef46","influxdb":"2e1a5f7.ecc9aa","name":"","measurement":"003-03-
light","precision":"","retentionPolicy":"","x":1257,"y":464,"wires":[]},{"id":"863f73e.2f8189","type":"influxdb
```


**which you can access via shell, edit, export, import, ...**

# InfluxDB /1

InfluxDB is an **open-source time series database** developed by InfluxData.

It is written in **Go** and optimized for fast, high-availability storage and retrieval of time series data in fields such as operations monitoring, application metrics, **Internet of Things sensor data**, and real-time analytics.

**Open Source:** Each component (except for some ...) of the InfluxData platform or TICK Stack is on github and available via a simple download.

Company Influxdata is venture funded.

# InfluxDB /2

**SQL-like** language with builtin **time-centric** functions for querying a data structure composed of measurements, series, and points.

Each point consists of several **key-value pairs** called the fieldset and a **timestamp.** When grouped together by a set of key-value pairs called the tagset, these define a series. Finally, series are grouped together by a string identifier to form a **measurement.**

```
measurement(,tag_key=tag_val)* field_key=field_val(,field_key_n=field_value_n)*
(nanoseconds-timestamp)
```

Values can be 64-bit integers, 64-bit floating points, strings, and booleans.

# InfluxDB /3

Main differences to a classical SQL database or NoSQL database:

It s for **time series** - nothing else!

A "table" (called measurement here) has 2 “columns”, not more:

a **timestamp** and the **value** for that point in time.

(You can add optional “tags” to create some structure).

You would not keep an address database or such in InfluxDB.

# InfluxDB /4 look inside: the shell

```
InfluxDB shell 0.10.0
> show databases
name: databases
---------------
name
_internal
pit001
pit002
pit003
```

# InfluxDB /5 look inside: the shell

```
> use pit001
Using database pit001
> show measurements
name: measurements
------------------
name
003-01-light
003-01-moist
003-01-temp
003-02-light
```

# InfluxDB /6 look inside: the shell

```
> select * from "004-01-temp"


1525096049108783758 50
1525096124247923458 49
1525096199476486703 49
1525096274435089220 50
1525096349502370874 50
1525096424982100010 50
1525096499794455963 49
```

# InfluxDB /7 access

Typically, you would access data via network rather than direct -

InfluxDB accepts data via **HTTP, TCP, and UDP**.

For example you could connect from ....

# Grafana /1

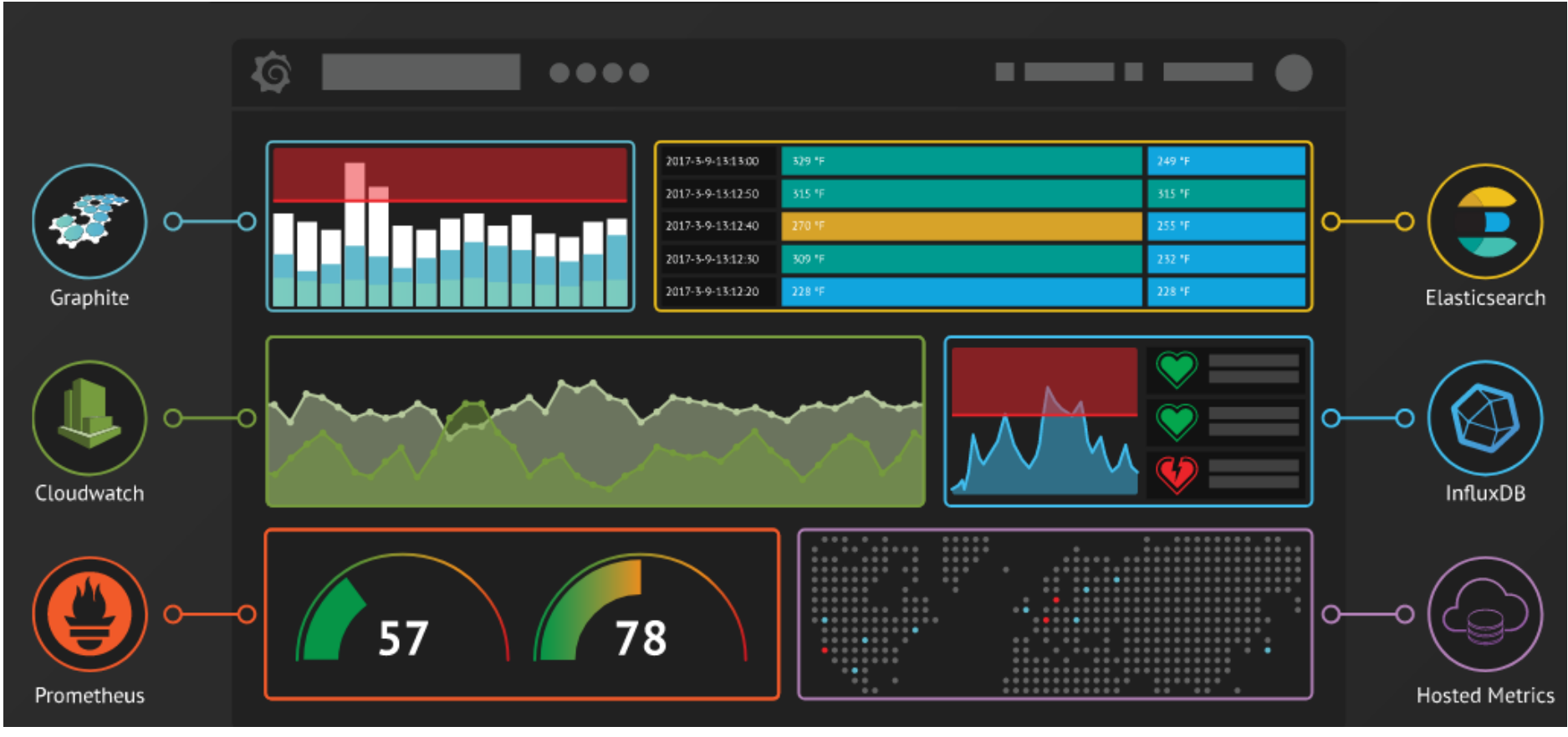
**Visualizing, Monitoring, Analyzing, Querying, Alerting**


As of right now, there are

**41 data sources, 30 panels, 17 apps and 857 dashboards available.**

**Data sources** include: influxDB, MySql, Postgres, Azure, ...

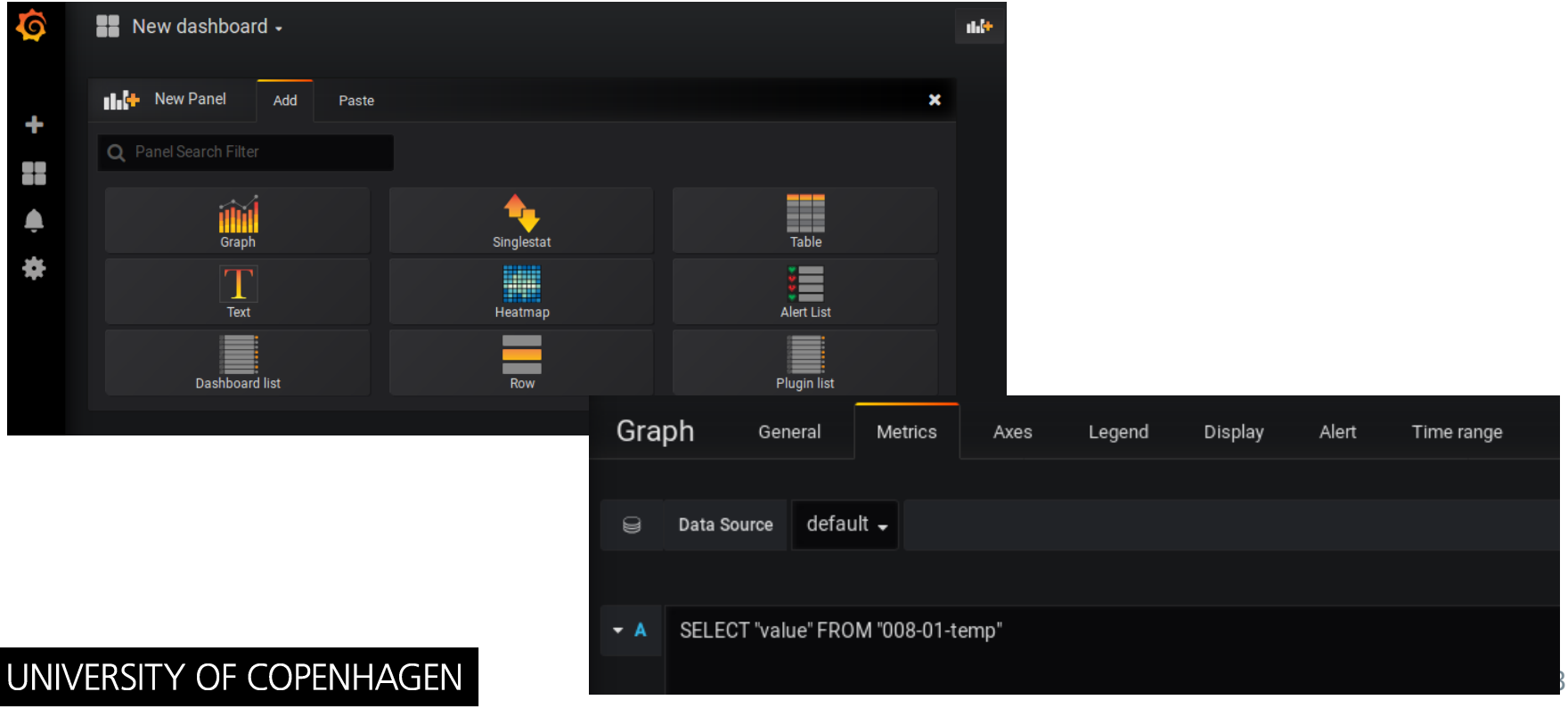
# Grafana /2


Graphite
Cloudwatch
Prometheus
Elasticsearch
InfluxDB
Hosted Metrics
2017-3-9-13:13:00
329 °F
249 °F
2017-3-9-13:12:50
315 °F
315 °F
2017-3-9-13:12:40
270 °F
255 °F
2017-3-9-13:12:30
309 °F
232 °F
2017-3-9-13:12:20
228 °F
228 °F
57
78

# Grafana /1

**Dashboards, Graphs, Queries:**

**Setting up a new one is mainly click, drag and drop.**



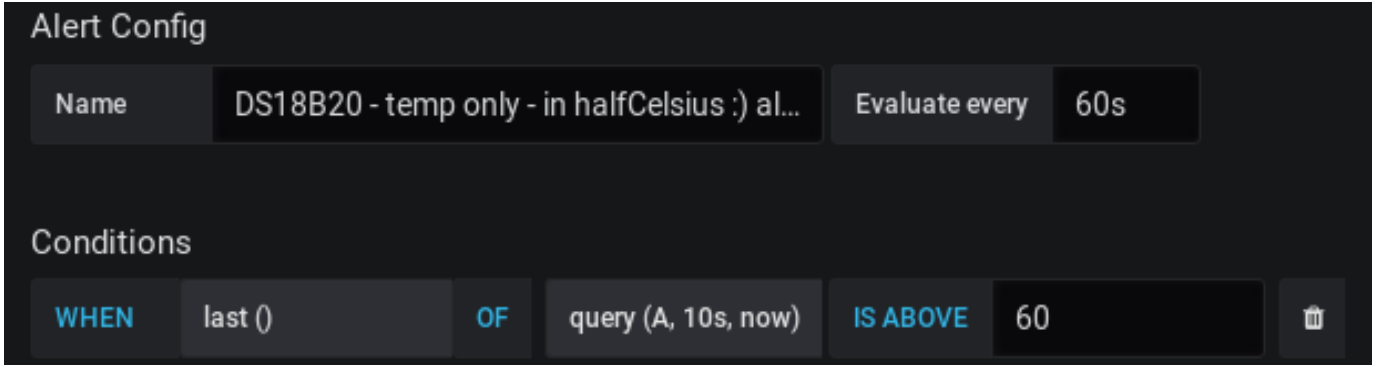
# Grafana /4

**Alerts**

**let you define flexible alert conditions**

**and**

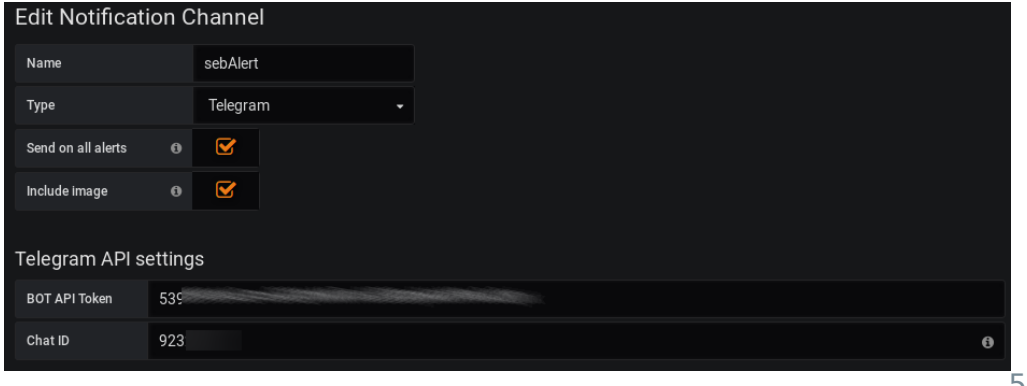
**may trigger**




**emails,**

**telegram messages,**

**slack messages, ...**



# Node-RED, Grafana & Influx demo

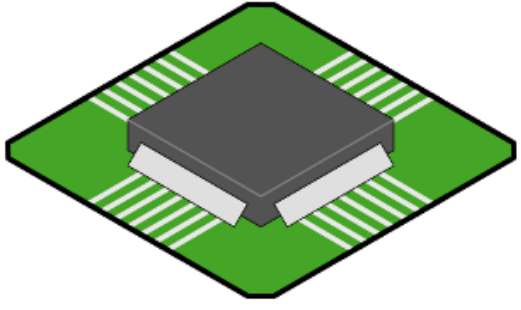
# Installing your own

## Get Started

Node-RED is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.

### Run locally

- Getting started
- Docker



### On a device

- Raspberry Pi
- BeagleBone Black
- Interacting with Arduino
- Android

### In the cloud

- IBM Bluemix
- SenseTecnic FRED
- Amazon Web Services
- Microsoft Azure

# Installing your own

- While a server is the preferred place for these, you can do a demo on your laptop.
- Our guide is at

  https://github.com/ITU-PITLab/public/blob/master/TheThingsNetwork+node-red+influxdb+grafana.md

  It mainly links to guides on the websites of TTN, Node-RED, influxdb and Grafana, and adds a few tips and hints.
- It is targeted at Debian/Ubuntu – users on other operating systems might find this more useful:

  https://www.thethingsnetwork.org/labs/story/store-and-visualize-data-using-influxdb-and-grafana

# Installing your own

## For the impatient, here s the short version:

```
# Installation / commands summary (Debian/Ubuntu)

// Node-RED

# apt-get install nodejs     (<<< or nodejs-legacy)

# nodejs -v                  ( or: node -v)

# apt install npm

# npm install -g --unsafe-perm node-red

(start with)

# node-red

// Influx

# apt-get install influxdb influxdb-client

(test with)

# influx

// Grafana

# wget https://s3-us-west-2.amazonaws.com/grafana-releases/release/grafana_5.1.0_amd64.deb

# sudo dpkg -i grafana_5.1.0_amd64.deb

# service grafana-server start
```

# Thank you!

- **Contact me via**

  **sebastian@itu.dk**