CAUBEL AKSEL

PRUVOST ARNAUD

DOMERGUE MATHYS

GROUPE IOT-51

TP-01

MISE EN PLACE DE L'INFRASTRUCTURE RÉSEAU

INSTALLATION INFLUXDB

INSTALLATION DU PACKAGE INFLUXDB

Pour installer la base de données InfluxDB :

Installer d'abord le package InfluxDB

```
wget -q https://repos.influxdata.com/influxdb.key
echo'23a1c8836f0afc5ed24e0486339d7cc8f6790b83886c4c96995b88a061c5bb5d influxdb.key' | sha256sum -c && cat influxdb.key | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/influxdb.gpg > /dev/null

echo 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdb.gpg] https://repos.influxdata.com/debian stable main' | sudo tee /etc/apt/sources.list.d/influxdata.list

sudo apt update

sudo apt install influxdb2

sudo systemctl start influxdb
```

Il faut ensuite créer les bases de données et les utilisateurs.

```
root@scw-nervous-keller:~# influx setup
> Welcome to InfluxDB 2.0!
? Please type your primary username admin
? Please type your password ********
? Please type your password again ********
? Please type your primary organization name iutbeziers
? Please type your primary bucket name telegraf
? Please type your retention period in hours, or 0 for infinite 0
? Setup with these parameters?
  Username:          admin
  Organization:      iutbeziers
  Bucket:            telegraf
  Retention Period:  infinite
 Yes
User        Organization    Bucket
admin       iutbeziers telegraf
```

On peut ensuite se connecter à version graphique depuis un navigateur en recherchant : "http://51.158.110.29:8086/"

Il suffit ensuite de se connecter en utilisant les identifiants créés précédemment.

INSTALLATION TELEGRAPH

```
sudo apt install wget curl gnupg2 lsb-release ca-certificates apt-transport-https software-properties-common -y
wget -qO- https://repos.influxdata.com/influxdb.key | sudo apt-key add -

echo "deb https://repos.influxdata.com/debian $(lsb_release -cs) stable"|sudo tee /etc/apt/sources.list.d/influxdb.list

sudo apt update

sudo apt install telegraf
```

Telegraph est la partie récupération de donnée pour la mettre sur InfluxDB. Nous avons essayé de la mettre en place, mais cela n'a pas abouti. Après réflexion, la solution du problème a été trouvée, mais aillant déjà mit un autre système en place, Telegraph n'a pas était modifié. Le problème avec la configuration donnée que nous essayons de nous connecter sur un mosquitto (fait par nous) pour récupérer une information déjà transformée, mais la partie du transfert de données ne marchait pas. Au vu des configurations vu, j'ai pu en déduire que le problème venait du dataType.
Voici tout de même un extrait de la configuration faite :

```
[[inputs.mqtt_consumer]]
  ## Broker URLs for the MQTT server or cluster.  To connect to multiple
  ## clusters or standalone servers, use a separate plugin instance.
  ##   example: servers = ["tcp://localhost:1883"]
  ##            servers = ["ssl://localhost:1883"]
  ##            servers = ["ws://localhost:1883"]
  servers = ["tcp://51.158.110.29:1883"]
  ## Topics that will be subscribed to.
  topics =["/sensors/#",]

  [[inputs.mqtt_consumer.topic_parsing]]
    topic = "/sensors/+/+"
    measurement = "/measurement/_/_"
    tags = "/_/devEUI/dataType"
    #fields = "/_/_/value"
    data_format = "value"
    data_type = "float"

  username = "admin"
  password = "IOT51AMA"
```

INSTALLATION GRAPHANA

INSTALLATION DU PACKAGE GRAPHANA

```
sudo apt update && sudo apt -y full-upgrade
[ -f /var/run/reboot-required ] && sudo reboot -f
sudo apt install -y gnupg2 curl software-properties-common
curl -fsSL https://packages.grafana.com/gpg.key | sudo gpg -dearmor -o /etc/apt/trusted.gpg.d/grafana.gpg
sudo add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"

sudo apt update

sudo apt -y install grafana
```

ACTIVATION DU SERVEUR

Pour démarrer le serveur

```
sudo systemctl start grafana-server
```

Les ports du serveur sont déjà ouvert auprès du DSI.

Après avoir installé et lancé le serveur, on se connecte depuis un navigateur web avec l'adresse suivante:

adresse_de_la_machine :3000

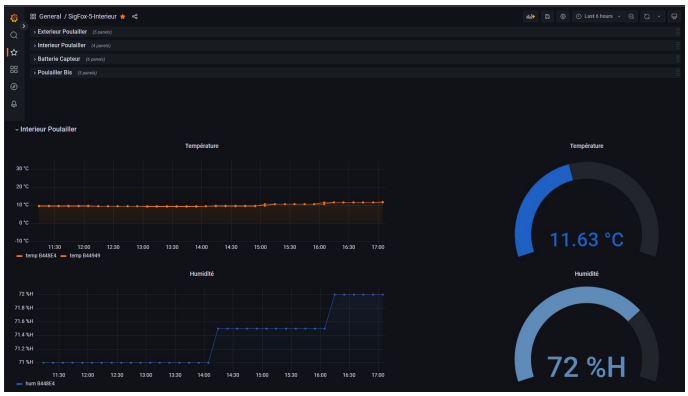
Puis vous vous connectez avec l'utilisateur admin et le mot de passe admin. Vous changez alors le mot de passe de l'utilisateur admin.

Vous devez alors relier votre InfluxDB avec votre Grafana en modifiant une des data sources. Pour cela, on va utiliser le langage Flux, car InfluxQL est obsolète depuis la version 2. Ensuite, on informe l'url de l'InfluxDB qui est :

adresse_de_la_machine: 8086

Puis on demande une authentification basic avec justificatif. Pour finir dans la partie liaison de notre InfluxDB et Grafana, on modifie "InfluxDB Détails", on ajoute l'organisation, le token généré sur InfluxDB, et le bucket. Pour finir la configuration de Grafana, vous allez créer une dashboard. Une dashbord est l'interface où vous pourrez consulter vos informations de votre base de données.

Pour cela, vous choisissez la partie dashboard, puis cliquez sur le bouton "New", puis choisissez "Add a new panel", c'est alors que vous allez commencer les filtres avec le langage Flux pour récupérer les données qui nous interressent.



Par exemple:

```
from(bucket: "IUT_Devices")
  |> range(start: v.timeRangeStart, stop: v.timeRangeStop)
  |> filter(fn: (r) => r["_measurement"] == "sensors")
  |> filter(fn: (r) => r["device"] == "B44037")
  |> filter(fn: (r) => r["_field"] == "temp")
  |> aggregateWindow(every: v.windowPeriod, fn: mean, createEmpty: false)
  |> yield(name: "mean")
```

Comme vous pouvez le voir, les données sont rangées de manière ordonnée pour avoir une racine qui leur correspond "sensors"(capteurs) puis le type de donnée que l'on souhaite regarder pour finir sur quel capteur voulons-nous regarder. Les capteurs aillant la nécessiter de devoir être unique, ils sont enregistrés en fonction de leur devEUI qui est un identifiant unique.

SCRIPT PYTHON POUR SIGFOX

```
import requests
import json
from influxdb_client import InfluxDBClient, Point
from influxdb_client.client.write_api import SYNCHRONOUS
from time import sleep

devices = ["B448E4","B44037","B44949"] # Liste des capteurs Sigfox
login = "6334446bbdcbd20e4daf92b4"
password = "3806e9d6cead4f45803df27eee00994a"
authentication = (login, password) # creds pour Sigfox


while True: # toute les 10mins

    for device in devices:


        response = requests.get(f"https://backend.sigfox.com/api/v2/devices/{device}/messages",
        auth=authentication) # questionne l'API REST Sigfox

        jsonData = json.loads(response.text) # transforme le Str en JSON
        lastData = jsonData["data"][0] #prend la dernière valeurs


        my_hexdata = lastData['data'] # extrait le payload ( Hexa )


        scale = 16 ## nb caractère

        num_of_bits = len(my_hexdata)*4

        binData = bin(int(my_hexdata, scale))[2:].zfill(num_of_bits) # hexa -> bin


        """ Parcing payload avec information Sigfox """

        batterie = (int(binData[:5],2)* 0.05) + 2.7

        reserved = binData[5:8]
        dataType = binData[8:13]

        flag = binData[13:14]

        dataTmp = (int(binData[14:24],2)-200)/8

        dataHum = int(binData[24:32],2)/2


        bucket = "IUT_Devices" # Bucket influxDB
        token = "NZuKx9nuCgkreX6PJ7jd2IdEGNVAosqmxcJUnJi944WTODCL-
XJ1WXcAEHcMwmpEbML20G4P60QEr99YiU-xcg==" # Token Bucket
```

```
    client = InfluxDBClient(url="http://51.158.110.29:8086", token=token, org="iutbeziers") # Login to Influx
    write_api = client.write_api(write_options=SYNCHRONOUS) # Write mode + synchronise

    t = Point("sensors").tag("device",device).field("temp",dataTmp) # create the influx point
    h = Point("sensors").tag("device",device).field("hum",dataHum) # create the influx point
    v = Point("sensors").tag("device",device).field("vdd",batterie)

    write_api.write(bucket=bucket, record=t) # write
    write_api.write(bucket=bucket, record=h) # write
    write_api.write(bucket=bucket, record=v) # write

    print(f"Data push : tmp = {dataTmp} | hum = {dataHum}")
    client.close() # ferme la connexion client



sleep(60*10) # dors 10mins
```