Le Protocole MQTT

Nicolas Vadkerti

31 décembre 2019

https://github.com/SlaynPool/CR_MQTT/

1 Le but de la seance

Comme vu dans le compte rendu de la premier journée : https://github.com/SlaynPool/CR_MQTT/blob/master/CR_DAY_ONE/MQTT.pdf nous savons comment fonctionne le Protocole MQTT. Le but aujourd'hui est donc de mettre en place une infrastructure complete utilisant ce protocole.
Ainsi, nous allons déployer un broker MQTT, une base de donnée adapté aux besoins de l'IOT et un moyen de visualiser les données de la Base de donnée.

2 Le broker MQTT

Pour cette partie, nous allons utiliser Mosquito https://mosquitto.org/
Je n'ai pas configurer de mots de passe pour l'accés à notre broker ce qui fais que je n'ai pas eu a configurer le broker. A noter, tous les machines que nous allons utiliser dans ce CR seront des Debian 10.

```
apt-get install mosquitto
#outils pour publier et s'abonner aux topics MQTT
apt-get install mosquitto-clients
systemctl status mosquitto
 mosquitto.service - Mosquitto MQTT v3.1/v3.1.1 Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor preset:
       enabled)
   Active: active (running) since Sun 2019-12-01 21:53:13 CET; 5s ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
 Main PID: 1373 (mosquitto)
    Tasks: 1 (limit: 1147)
   Memory: 1.0M
   CGroup: /system.slice/mosquitto.service
           -1373 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf
```

Listing 1 – Installation du broker

On verifie que le broker fonctionne bien :

```
# On publie sur le broker :
MonBroker$:mosquitto_pub -h localhost -t un/topic -m "HelloWorld"

# On ecoute le topic :

UnsubDuBroker $:mosquitto_sub -h 192.168.1.82 -t un/topic
HelloWorld
```

Listing 2 – Test du broker

3 La base de donnée

Cette partie sera donc consacrée à comment nous allons récupérer les infos envoyer sur notre broker et comment nous allons les stocker. Il est sans doute bon de rappeller que le broker MQTT n'a pas pour rôle de stocker les données mais seulement de s'assurer de les livrers aux subscribers du topic.
En pratique, si nous voulons stocker les données émis par nos capteurs, il faut que nous ayons un "script/programme/plugin" qui soit capable de se subcribe à tous les topics qui nous interesse, et faire les bonnes requetes SQL sur la bonne Base de Donnés pour les stocker

3.1 Choix et déploiment de la base de Donnée

Nous voulons faire de l'IOT, on peut facilement supposer que nous allons potentiellement émettre beaucoup de petites données qui auront de la cohérence en fonction du temps. Un produit repondant à ce besoin éxiste et s'appelle InfluxDB.

InfluxDB is an open-source time series database (TSDB) 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.

source: Wikipedia

3.1.1 Déploiment de InfluxDB

Pour cela, j'ai suivi la documentation disponible sur le site de InfluxDB

```
user@MonSERVEUR INFLUXDB ~:cat install.sh
#!/bin/bash
wget -qO- https://repos.influxdata.com/influxdb.key |  apt-key add -
source /etc/os-release
test $VERSION_ID = "7" && echo "deb https://repos.influxdata.com/debian wheezy stable" |
      tee /etc/apt/sources.list.d/influxdb.list
test $VERSION_ID = "8" && echo "deb https://repos.influxdata.com/debian jessie stable" |
      tee /etc/apt/sources.list.d/influxdb.list
test $VERSION_ID = "9" && echo "deb https://repos.influxdata.com/debian stretch stable"
    |  tee /etc/apt/sources.list.d/influxdb.list
apt-get update &&  apt-get install influxdb
service influxdb start
```

Listing 3 – Installation de InfluxDB

De plus, pour pouvoir interagir avec notre base :

```
apt install influxdb-client
```

Listing 4 – Installation du client

note : J'ai supprimer la VM entre la realisation des manipulations en cours et la redaction du compte rendu, j'ai donc recommemencé les manipulations sur une autre machines d'où l'incohérence des promptes On verifie que l'on peut bien accéder à notre base :

```
root@debian:~# influx
Connected to http://localhost:8086 version 1.6.4
InfluxDB shell version: 1.6.4
> SHOW DATABASES
name: databases
name
----
_internal
telegraf
>
```

Listing 5 – accées à la base

4 Télégraf

Nous allons maintenant nous intérésser à un outils très pratiques qui est Télégraf. En effet, il sert à lier notre base de données InfluxDB à differentes téchnologie, et dans notre cas present, un broker MQTT, aux travers de "plugins" dont une bonne partie est près installé . Son rôle va donc de se subcribe aux topics de notre broker, et de stocker les données dans la base.

4.1 Déploiment de Télégraf

source : `https://portal.influxdata.com/downloads/`

```
wget https://dl.influxdata.com/telegraf/releases/telegraf_1.13.0-1_amd64.deb
sudo dpkg -i telegraf_1.13.0-1_amd64.deb
```

Listing 6 – Déploiment de télégraf

Et pour ce qui est de la configuration : La commande diff n'est pas de moi mais trouvé sur internet.

```
root@debian:/etc/telegraf# diff telegraf.conf telegraf.conf.old -i -b -B --ignore-file-
    name-case --ignore-case -a --new-line-format='%dn [suppression] %L' --old-line-format
    ='%dn [ajout] %L' --unchanged-line-format=''
112 [ajout]    urls = ["http://127.0.0.1:8086"]
112 [suppression]   # urls = ["http://127.0.0.1:8086"]
116 [ajout]   database = "mesure"
116 [suppression]   # database = "telegraf"
5432 [ajout]  [[inputs.mqtt_consumer]]
5432 [suppression] # [[inputs.mqtt_consumer]]
5435 [ajout]    servers = ["tcp://127.0.0.1:1883"]
5435 [suppression] #   servers = ["tcp://127.0.0.1:1883"]
5438 [ajout]    topics = [
5439 [ajout]      "telegraf/host01/cpu",
5440 [ajout]      "telegraf/+/mem",
5441 [ajout]      "sensor/#",
5442 [ajout]    ]
5438 [suppression] #   topics = [
5439 [suppression] #     "telegraf/host01/cpu",
5440 [suppression] #     "telegraf/+/mem",
5441 [suppression] #     "sensors/#",
5442 [suppression] #   ]
5495 [ajout]    data_format = "value"
5496 [ajout]    data_type = "float"
5495 [suppression] #   data_format = "influx"
```

Listing 7 – Configuration de télégraf

J'ai donc configuré télégraf pour qu'il se subscribe à differents topics MQTT, et lui est préciser quelle type de valeurs il devait attendre. Si on regarde dans notre base de donnés, il doit y avoir les données que nous donnons au broker, ici un Raspberry connécter à un accelerometre qui envoie les valeurs de celui-ci :

```
root@debian:/etc/telegraf# influx
Connected to http://localhost:8086 version 1.6.4
InfluxDB shell version: 1.6.4
> SHOW DATABASES
name: databases
name
----
_internal
telegraf
mesure
> USE mesure
Using database mesure
> SHOW MEASUREMENTS
name: measurements
name
----
cpu
disk
diskio
kernel
mem
mqtt_consumer
processes
swap
system
> SELECT * FROM mqtt_consumer
name: mqtt_consumer
time                host   topic                 value
----                ----   -----                 -----
1576507420189291721 debian sensor/accelero/roll  341.7
1576507420189807876 debian sensor/accelero/pitch 359.2
1576507420189873525 debian sensor/accelero/yaw   360
1576507421214200158 debian sensor/accelero/roll  341.7
1576507421214733488 debian sensor/accelero/pitch 359.2
1576507421214830379 debian sensor/accelero/yaw   360
1576507422267303102 debian sensor/accelero/roll  341.7
```

Listing 8 – Preuve que Télégraf fonctionne

5 Visualiser les données

Nous sommes capables d'émettre nos données, nous somme capable de les stocker, il nous reste donc plus qu'a les visualiser via une interface Web par exemple. Une solution très efficase et bien documenté est Grapfana. Grapfana est capable de se connécter à une base Influxdb et de générer des graphiques à partir de celui-ci.

5.1 Déploiment de grafana

source : `https://grafana.com/docs/grafana/latest/installation/debian/`

```
#!/bin/bash
apt-get install -y apt-transport-https
apt-get install -y software-properties-common wget
add-apt-repository "deb https://packages.grafana.com/oss/deb stable main"
add-apt-repository "deb https://packages.grafana.com/oss/deb beta main"
wget -q -O - https://packages.grafana.com/gpg.key |  apt-key add -
apt-get update
apt-get install grafana
service grafana start
```

Listing 9 – installation de grafana

On peut donc verifier que grafana fonctionne bien :

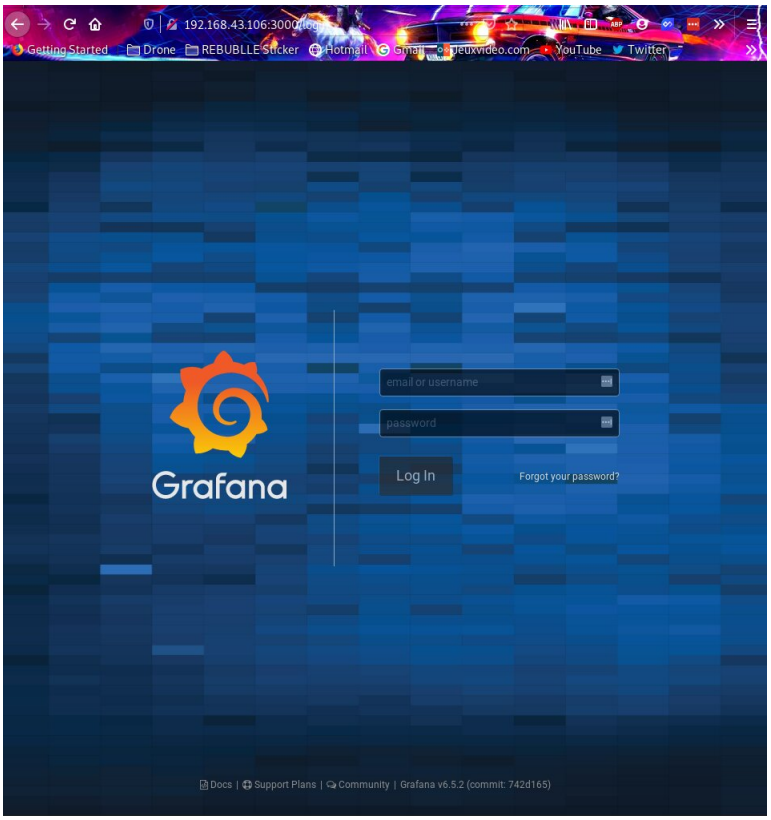

FIGURE 1 – Interface

On peut donc ce connecter avec les mots de passes par default : " admin" "admin" et configurer les data sources comme ceci dans notre cas :
On sauvegrade celle-ci. Quand nous sauvegardons, il grafana essaye desuite de ce connecter à la base de donné donc nous savons directement si elle est correctement configuré ou non.

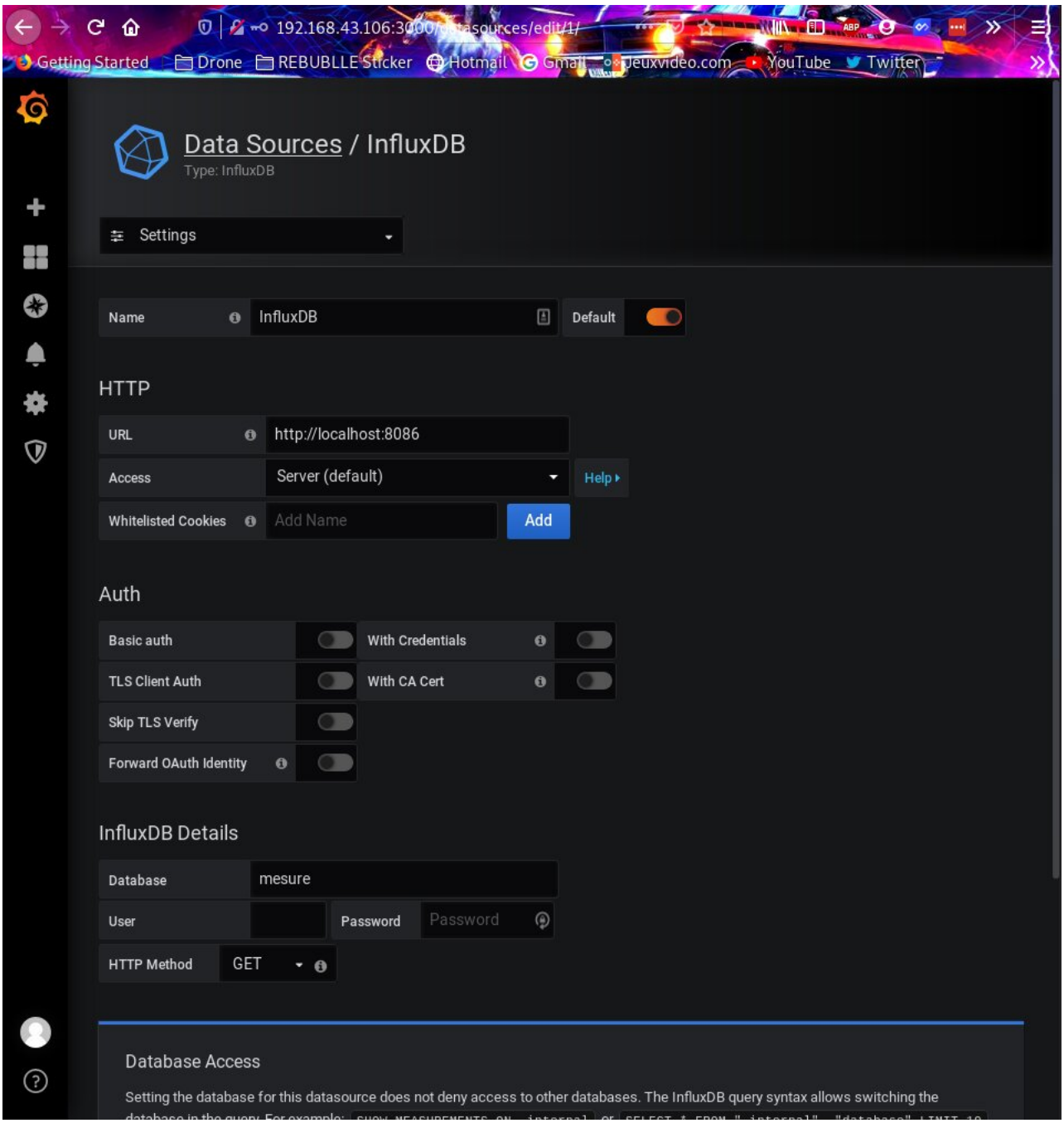

FIGURE 2 – Network Schema

Des lors, nous pouvons configurer nos nouveaux dashboards comme ci dessous :
Il suffit de lui donner les bonnes requetes (ce qui via l'interface est très intuitif par ailleur) et il génére le grafique qui va bien.

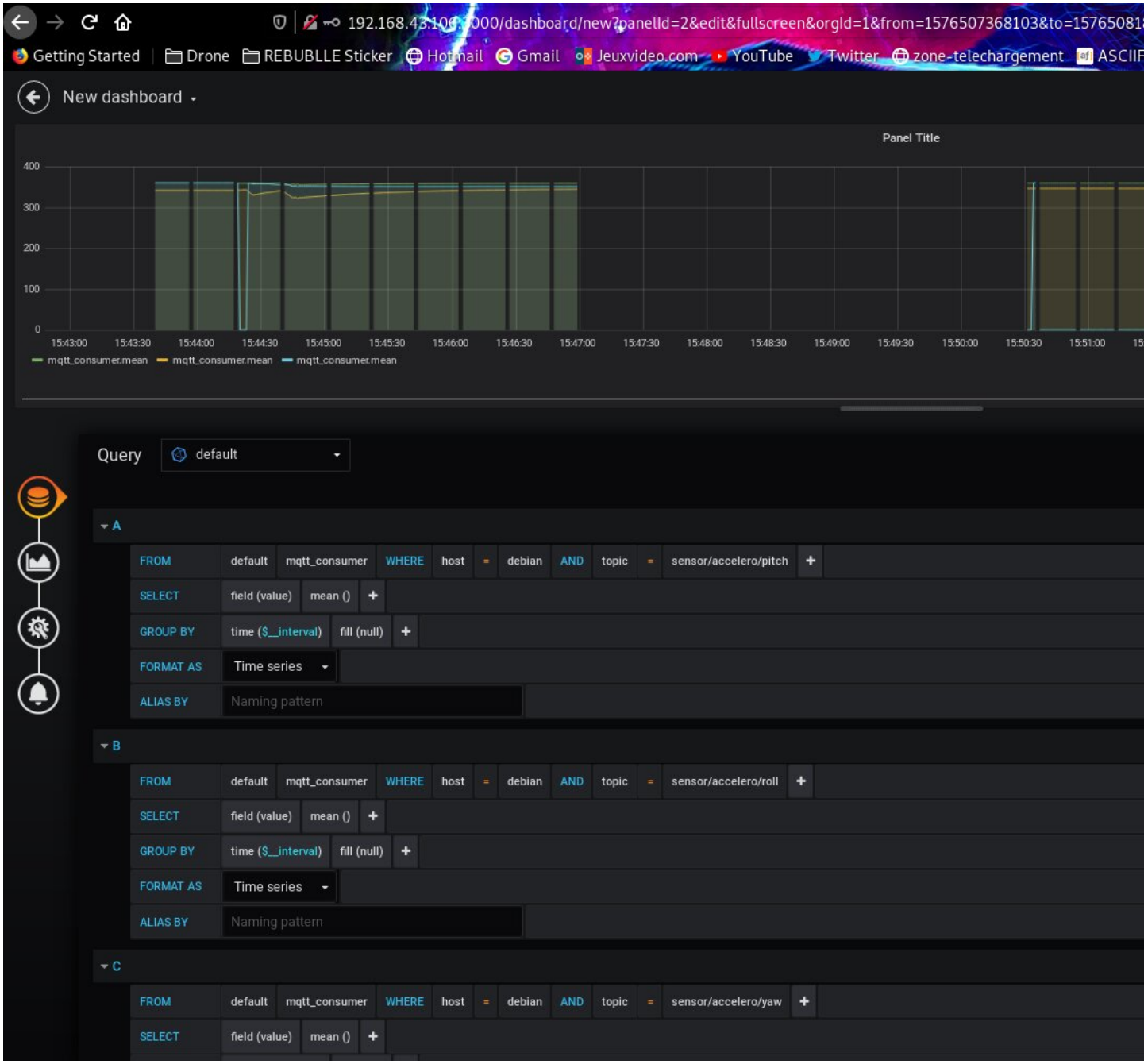

FIGURE 3 – Network Schema