

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 Mosquitto <https://mosquitto.org/>
Je n'ai pas configuré de mots de passe pour l'accès à notre broker ce qui fait que je n'ai pas eu à configurer le broker. A noter, tous les machines que nous allons utiliser dans ce CR seront des Debian 10.

```
apt-get install mosquitto
#utils pour publier et s'abonner aux topics MQTT
apt-get install mosquitto-clients
systemctl status mosquitto
5  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)
10  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 :
5 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 envoyés sur notre broker et comment nous allons les stocker. Il est sans doute bon de rappeler que le broker MQTT n'a pas pour rôle de stocker les données mais seulement de s'assurer de les livrer aux subscribers du topic.

En pratique, si nous voulons stocker les données émis par nos capteurs, il faut que nous ayons un "script/plugin" qui soit capable de se subscribe à tous les topics qui nous intéressent, et faire les bonnes requêtes SQL sur la bonne Base de Données pour les stocker

3.1 Choix et déploiement 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 répondant à ce besoin existe 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 Déploiement 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
5 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 supprimé la VM entre la réalisation des manipulations en cours et la rédaction du compte rendu, j'ai donc recommencé les manipulations sur une autre machine d'où l'incohérence des prompts. On vérifie 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
5 name: databases
name
----
_internal
telegraf
10 >
```

Listing 5 – accès à la base

4 Télégraf

Nous allons maintenant nous intéresser à un outil très pratique qui est Télégraf. En effet, il sert à lier notre base de données InfluxDB à différentes technologies, et dans notre cas présent, un broker MQTT, aux travers de "plugins" dont une bonne partie est déjà installée. Son rôle va donc de se subscribe aux topics de notre broker, et de stocker les données dans la base.

4.1 Déploiement 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 – Déploiement 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"
5 116 [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"]
10 5438 [ajout]     topics = [
5439 [ajout]         "telegraf/host01/cpu",
5440 [ajout]         "telegraf/+/mem",
5441 [ajout]         "sensor/#",
5442 [ajout]     ]
15 5438 [suppression] # topics = [
5439 [suppression] #     "telegraf/host01/cpu",
5440 [suppression] #     "telegraf/+/mem",
5441 [suppression] #     "sensors/#",
5442 [suppression] # ]
20 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 différents topics MQTT, et lui est précisé quelle type de valeurs il devait attendre. Si on regarde dans notre base de données, il doit y avoir les données que nous donnons au broker :