BDD - InfluxDB, une base TSDB

[Mise à jour le 10/8/2021]

- **Sources**
  - Site influxdata
  - Tutoriel Installer InfluxDB sur Raspberry Pi, Windows, macOS.
  - Tutoriel Enregistrer les mesures d'un objet connecté MySensors vers InfluxDB à l'aide de Node-RED
- **Lectures connexes**
  - Wiki Raspberry Pi - Installer un broker MQTT Mosquitto sur un Raspberry Pi
  - Wiki Arduino - Mettre en œuvre un client MQTT sur un EP8266 (ESP32) Feather Huzzah, MKR1010 ou Arduino Uno Wifi 2
  - Wiki Réseau - Tester un broker Mosquitto avec MQTTlens
  - Wiki Raspberry Pi sous Linux - Créer un flux de données et une interface utilisateur avec Node-RED
  - Wiki Web - Créer un client MQTT (Websockets) avec Eclipse Paho

---

1. Généralités

InfluxDB est un système de gestion de base de données orientée **TSDB**[1)] ou **séries chronologiques** hautes performances, écrit avec le langage de programmation Go et distribué sous licence MIT.

Les bases de données traditionnelles (MariaDB, MySQL etc.) sont conçues pour stoker et mettre en relation les données . Elles sont bien adaptées pour gérer les comptes client et enregistrer les commandes d'un site marchand. Rien n'interdit l'enregistrement de données qui dépendent du temps mais dans ce cas il faut effacer manuellement les données périmées. D'autres architectures de bases de données ont été imaginées.

InfluxDB répond à **trois objectifs** :

- **Simplifier** la configuration et la gestion
- Moteur d'enregistrement et d'accès aux données qui dépendent du temps (**horodatage**)
- Créer des règles de rétention et de suppression **automatique** des données périmées

2.Installer InfluxDB

2.1 Sur Raspbian Stretch ou supérieur pour Raspberry Pi

- **Source** sur Github

Mettre à jour le système.

*.bash

```
sudo apt-get update && sudo apt-get upgrade
```

Avant de pouvoir installer InfluxDB sur Raspbian, il est nécessaire d'indiquer au système ou se trouvent les clés du dépôts du projet InfluxDB.

*.bash

```
curl -sL https://repos.influxdata.com/influxdb.key | sudo apt-key add
```

Il existe deux versions d'InfluxDB. La première est adaptée à Debian 8.0 ou supérieur (Raspbian Jessie, Stretch ou Buster), l'autre pour Debian 7.0 (Raspbian Wheezy)

Pour connaître la version de votre système, exécutez la commande **lsb_release -a**

*.bash

```
pi@RPi3bp2:~ $ lsb_release -a
No LSB modules are available.
Distributor ID: Raspbian
Description:    Raspbian GNU/Linux 10 (buster)
Release:    10
Codename:   buster
```

Exécutez ensuite les commandes ci-dessous pour ajouter le paquet apt-transport-https et le lien vers la source d'InfluxDB puis mettre à jour le système et enfin installer InfluxDB.

*.bash

```
sudo apt install apt-transport-https
echo "deb https://repos.influxdata.com/debian jessie stable" | sudo tee
/etc/apt/sources.list.d/influxdb.list
sudo apt-get update
sudo apt-get install influxdb
```

InfluxDB est maintenant installé déjà pré-configuré mais pour enregistrer des données depuis un objet connecté à l'aide d'une requête HTTP ou depuis Node-RED on doit autoriser les connexions depuis le port HTTP. Pour cela, il suffit de modifier le fichier de configuration qui se trouve dans le répertoire **/etc/influxdb/influxdb.conf** avec la commande ci-dessous.

*.bash

```
sudo nano /etc/influxdb/influxdb.conf
```

On dé-commente les lignes suivantes

```
[http]
  # Determines whether HTTP endpoint is enabled.
  enabled = true

  # The bind address used by the HTTP service.
  bind-address = ":8086"

  # Determines whether user authentication is enabled over HTTP/HTTPS.
  auth-enabled = false
```

2.2 Sur Windows 64-bits

Vous pouvez l'installer en le téléchargeant depuis cette page ou en installant wget pour Windows. Dans ce cas, exécutez la commande suivante :

*.bash

```
wget
https://dl.influxdata.com/influxdb/releases/influxdb-1.1.0_linux_amd64.
tar.gz
tar xvfz influxdb-1.1.0_linux_amd64.tar.gz
```

3. Tester le fonctionnement

Une fois l'installation terminée, vérifions que tout fonctionne correctement. InfluxDB expose une API avec laquelle on peut facilement vérifier son fonctionnement. La commande curl ci-dessous devrait donner :

*.bash

```
$ curl -sl -I localhost:8086/ping
HTTP/1.1 204 No Content
Content-Type: application/json
Request-Id: 165b0fa4-ad88-11e6-8a6f-000000000000
X-Influxdb-Version: v1.1.0
```

```
Date: Fri, 18 Nov 2016 12:11:08 GMT
```

4. Terminologie, différences avec une base SQL classique

Comme toutes les base de données non relationnelles, InfluxDB dispose de son propre vocabulaire qu'il convient de connaître avant de se lancer.

- **Measurement** est équivalent à une **table** SQL
- **Tags** est similaire à une **colonne indexée** (column) dans une base SQL
- **Fields** est similaire aux **colonnes non indexées** dans une base SQL
- **Points** est équivalent à une **ligne** (row) dans une base SQL

5. Gestion des bases de données

Il existe 6 méthodes pour la gestion des bases de données :

- CREATE DATABASE,
- DROP DATABASE,
- DROP SERIES,
- DELETE,
- DROP MEASUREMENT,
- DROP SHARD).

Voici comment fonctionnent les 2 premières, pour les autres, tout se trouve ici

5.1 Créer une base de données (create database)

- **Syntaxe**

*.sql

```
CREATE DATABASE "database_name" [WITH [DURATION "duration"]
[REPLICATION] [SHARD DURATION "shard_duration"] [NAME "retention-
policy-name"]]
```

- **Paramètres**
  - *database_name* : nom de la base de données
  - *duration* : on peut indiquer une durée de rétention (conservation) des données mais il est plus judicieux de mettre en place une politique de rétention (voir paragraphe suivant)
- **Exemple**

*.sql

```
CREATE DATABASE "temperatures"
```

5.2 Effacer une base de données (drop database)

- **Syntaxe**

*.sql

```
DROP DATABASE "database_name"
```

On indique juste le nom de la base (database_name) à supprimer.

6. Mettre en place une politique de rétention

influxDB intègre un système de suppression automatique de données, c'est la Politique de Rétention (Retention Policy). Passé un certain temps, les données sont automatiquement supprimées de la base. Ce mécanisme est très pratique car on a une bonne maitrise de la taille de la base et on n'a pas à se soucier de cette opération de maintenance. Libre à chacun de définir la durée de vie de telle ou telle mesure. Si on veut suivre la température de son microprocesseur, une durée d'un mois peut être suffisante, au contraire la politique de rétention devra être plus souple si on veut pouvoir comparer la température dans une maison d'une année sur l'autre. Il faudra aussi penser au système en amont, inutile de stocker une mesure de température d'ambiance toute les 10 secondes ! Une mesure par minute, voir 15 minutes, c'est largement suffisant. Au contraire dans un process industriel, on voudra certainement aller chercher la seconde...ou beaucoup moins.

On dispose donc de 3 méthodes

- **Create** : pour créer un règle de conservation des données
- **Alter** : pour modifier une règle
- **Drop** : pour supprimer une règle

6.1 Créer un politique de rétention (create retention policy)

- **Syntaxe**

*.sql

```
CREATE RETENTION POLICY "nom_politique_retention" ON "nom_database"
DURATION "duree"REPLICATION [SHARD DURATION "duree_shard"] [DEFAULT]
```

- **Paramètres**
  - **nom_politique_retention** : un nom court sans espaces de préférence
  - **nom_database** : la base de données sur laquelle s'applique la politique de rétention
  - **duree** : durée de conservation des données. C'est une expression littérale qui doit respecter la convention InfluxDB (ici). Par exemple si on veut conserver les données 30 jours, on indiquera 30d. Si on veut garder sans limitation de temps, on indique INF (infini).

  - **duree_shard** : la durée de stockage des données dans la zone tampon. Voir ici pour plus d'infos.
  - **n** : nombre de réplication (obligatoire), mettre 1 si aucun cluster installé

- **Exemple**

*.sql

```
CREATE RETENTION POLICY "one_day_only" ON "NOAA_water_database"
DURATION 1d REPLICATION 1
```

Cette politique permet de conserver les enregistrements durant un jour (one_day_only) sur la base NOAA_water_database. Les données sont stockées dans l'espace shard 1 jour. Replication 1 indique qu'une copie de chaque élément est recopié dans le cluster (il faut qu'il cluster existe).

6.2 Modifier une politique de rétention (alter retention policy)

Cette méthode permet de modifier une politique de rétention existante.

- **Syntaxe**

*.sqm

```
ALTER RETENTION POLICY "nom_politique_retention" ON "nom_database"
DURATION "duree" REPLICATION SHARD DURATION "duree_shard" DEFAULT
```

- **Paramètres**
  - **nom_politique_retention** : le nom de la politique à modifier
  - **nom_database** : base de donnée sur laquelle elle s'applique
  - **duree, duree_shard** : voir ci-dessus
  - **n** : nombre de réplication (obligatoire), mettre 1 si aucun cluster installé

- **Exemple**

*.sql

```
ALTER RETENTION POLICY "what_is_time" ON "NOAA_water_database" DURATION
3w SHARD DURATION 30m DEFAULT
```

La méthode retourne un résultat vide.

6.3 Détruite une politique de rétention (drop retention policy)

- **Syntaxe**

*.sql

```
DROP RETENTION POLICY "nom_polique_retention" ON "nom_database"
```

- **Paramètres**
  - **nom_politique_retention** : le nom de la politique à modifier
  - **nom_database** : base de donnée sur laquelle elle s'applique

7. Le shell InfluxDB

InfluxDB dispose d'un shell qui permet de réaliser toutes les opérations sur les bases de données. Pour le lancer, il suffit de saisir **influx** dans le Terminal.

*.bash

```
pi@RPi3bp2:~ $ influx
Connected to http://localhost:8086 version 1.7.7
InfluxDB shell version: 1.7.7
```

La commande **exit** permet de sortir du shell.

Toutes les commande décrites dans ce paragraphe peuvent également être réalisées avec **curl**.

8. Une première mise en oeuvre

8.1 Création d'une base

*.sql

```
> CREATE DATABASE basetest
> SHOW DATABASES
name: DATABASES
name
----
_internal
basetest
```

On ajoute un utilisateur

*.sql

```
> CREATE USER utilisateur WITH PASSWORD 'motdepasse'
```

On lui donne tous les droits sur la base basetest.

*.sql

```
> GRANT ALL ON basetest TO projetsdiy
```

Et on ajoute une politique de rétention de 30 jours.

*.sql

```
> CREATE RETENTION POLICY "trente_jours" ON basetest DURATION 30d
REPLICATION 1 DEFAULT
```

On vérifie la politique de rétention.

*.sql

```
> SHOW RETENTION POLICIES ON basetest

name    duration shardGroupDuration replicaN DEFAULT
----    -------- ------------------ -------- -------
autogen 0s       168h0m0s           1        TRUE
```

Destruction de la base.

*.sql

```
> DROP DATABASE "basetest"
```

9. Enregistrer des mesures dans InfluxDB avec Node-RED

A faire

1)
Time Series DataBase

From:
https://webge.fr/dokuwiki/ - **WEBGE Wikis**

Permanent link:
https://webge.fr/dokuwiki/doku.php?id=info:bdd:influxdb_nr&rev=1628676877

Last update: **2021/08/11 12:14**