Le principe des Time Series DataBases

La problématique des bases de données pour les IOT

Dans les applications IOT, il est nécessaire d'utiliser des bases de données optimisées pour le stockage de métriques. Il faut par exemple pouvoir stocker un volume important de données prélevées sur l'ensemble de notre flotte d'objets connectés déployés. En effet, si vous effectuez une vingtaine de mesures par minutes sur une centaine d'objets, cela fait 2000 mesures à stocker toutes les minutes soit près de 3 millions de points/jour. **Les bases de données relationnelles traditionnelles** ne sont pas adaptées pour répondre à ce cas de figure. Il donc nécessaire d'utiliser un nouveau genre de bases de données; les **bases de données de séries chronologiques** ou temporelles (**time series database, TSDB**).

Caractéristiques des bases de données de séries chronologiques

Ce type de base de données est spécialisé dans le stockage de métriques. Elles n'ont pas vocation à devenir des bases de données généralistes comme MySQL ou autres SGBD mais se focalisent au contraire sur un certain nombre de caractéristiques qui font qu'elles deviennent incontournables dans le domaine des **IOT**.

Les caractéristiques d'une base de données de séries chronologiques sont :

- ✓ **Gros volume de données**

Le volume de données remonté par les objets connectés peut rapidement atteindre plusieurs millions de lignes. Collecter un point de données toutes les secondes pendant un an représente 31,5 millions de lignes.

- ✓ **Grande quantité de données immuables**

Les bases de données de séries chronologiques contiennent des données qui ne sont normalement pas modifiées après insertion.

- ✓ **Première clé de tri temporelle**

Les bases de données de séries chronologiques sont par concept triées chronologiquement. Il est donc facile de programmer une application qui trie les données chronologiques de façon efficace et automatique.

- ✓ **Résolution des données et statistiques**

Avec les données de séries chronologiques, vous avez souvent besoin de réduire la résolution pour pouvoir comprendre les données. Par exemple, si vous voulez voir l'évolution d'une température relevée par un objet connecté depuis un mois, vous n'avez probablement pas besoin de voir cette information à un intervalle (résolution) d'une seconde. Vous allez plutôt rechercher des maximums, minimums, moyennes, déviations par jour ou heure en fonction des données… Vous appliquez alors des fonctions statistiques.

Présentation d'InfluxDB et installation

Pour la base de données time series, nous allons utiliser InfluxDB.

Pourquoi InfluxDB ?

- ✓ La base est Open Source.
- ✓ Serveur de base de données la plus performante du marché selon DB-Engines.com pour ce qui est des Time Series Databases.
- ✓ Possibilité d'archivage de données numériques ET alphanumériques (String).
- ✓ Langage de requêtes semblable au langage SQL.
- ✓ Possibilités de scripting en Java, JavaScript, Node.js, PHP, Python….
- ✓ Réplication des données possible sur plusieurs bases. Exemple : synchroniser la base de données locale avec une base de données dans le Cloud.
- ✓ Choix de la pérennité des données. On peut choisir les données que l'on veut garder et combien de temps. C'est un des points forts d'InfluxDB.
- ✓ Librairie Node-RED existante.
- ✓ Compatible Raspberry Pi et Raspbian.

Installer InfluxDB sur Raspberry

Pour installer **InfluxDB** sur Raspberry Pi avec **Rasbian**, il y a besoin d'ajouter une nouvelle source apt supplémentaire :

```
wget -qO- https://repos.influxdata.com/influxdb.key | sudo apt-key add -
echo "deb https://repos.influxdata.com/debian buster stable" | sudo tee /etc/apt/sources.list.d/influxdb.list
```

Mettre à jour apt et installer influxDB :

```
sudo apt update && sudo apt install -y influxdb
```

Puis paramétrer le lancement automatique d'influxDB au démarrage de la Raspberry :

```
sudo systemctl unmask influxdb
sudo systemctl enable influxdb
sudo systemctl start influxdb
```

InfluxDB est maintenant installé. Pour une configuration très rapide, on va ouvrir ***/etc/influxdb/influxdb.conf*** et activer **le port HTTP** sans authentification :

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

Pour cela, il suffit de **dé-commenter** 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
```

Ensuite, on **redémarre** le service **InfluxDB** :

```
sudo systemctl restart influxdb
```

Notre système de bases de données est prêt.

Intégration InfluxDB dans Node-RED

Pour connecter une base de données InfluxDB à un flux Node-RED, il faut ajouter les librairies InfluxDB à Node-RED. Pour se faire, téléchargez le paquet "npm" qui va permettre d'ajouter la librairie InfluxDB à Node-RED.

```
cd ~/.node-red
npm install node-red-contrib-influxdb
```

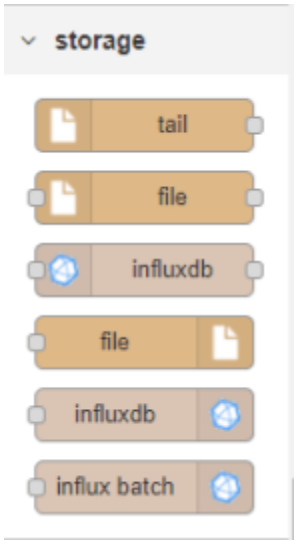
Note : npm devra toujours s'exécuter dans ***~/.node-red.***

Ensuite, redémarrez Node-RED :

```
cd ~/.node-red
node-red-stop
node-red-start
```

La librairie InfluxDB dans Node-RED est installée.

A l'ouverture de Node-Red, vous devez avoir accès aux outils InfluxDB dans la rubrique Storage



Pour d'autres exemples : https://flows.nodered.org/node/node-red-contrib-influxdb

Configuration d'InfluxDB

InfluxDB est une base de données optimisée sur les requêtes datées. La base de données InfluxDB contient des "Measurements" (comprendre Mesures) munis de plusieurs "DataPoints" (comprendre Données).

Afin d'avoir des dates compréhensibles, il faut lancer influx avec la commande -precision rfc3339 (norme d'écriture des dates au format Année-Mois-Jour H:min:sec).

```
influx -precision rfc3339
Connected to http://localhost:8086 version 1.7.10
InfluxDB shell version: 1.7.10
```

Nous allons créer une base SNIR_IOT qui contiendra toutes les mesures de température :

```
> create database SNIR_IOT
```

On vérifie:

```
> show databases
name: databases
name
----
_internal
SNIR_IOT
```

Il est possible de créer une rétention de données pour limiter la taille de la base de données. Par exemple, en entrant la commande suivante :

```
> create retention policy "un_mois" on SNIR_IOT duration 30d replication 1 default
```

Vérifier la configuration :

```
> show retention policies on SNIR_IOT
name    duration shardGroupDuration replicaN default
----    -------- ------------------ -------- -------
autogen 0s       168h0m0s           1        false
un_mois 720h0m0s 24h0m0s            1        true
```

Ensuite, simuler un capteur en insérant manuellement des mesures dans la base :

```
use SNIR_IOT
Using database SNIR_IOT
> insert temperature,location=g133 value=20
> insert temperature,location=g133 value=21
> insert temperature,location=g133 value=22
> insert temperature,location=g133 value=20
> insert temperature,location=g133 value=21
> insert temperature,location=g133 value=22
> insert temperature,location=g131 value=20
> insert temperature,location=g131 value=25
> insert temperature,location=g131 value=24
```

Ensuite, visualiser les données enregistrées dans la base :

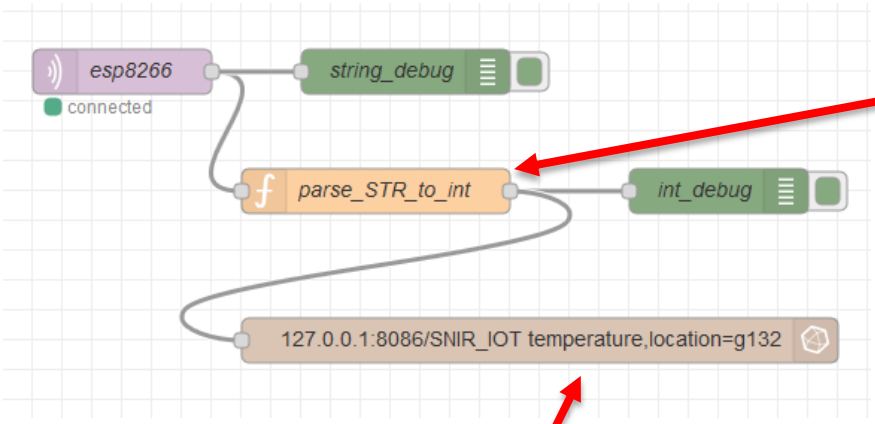
```
> select * from temperature
name: temperature
time                           location value
----                           -------- -----
2020-03-01T21:24:53.741546673Z g133     20
2020-03-01T21:25:00.417211725Z g133     21
2020-03-01T21:25:05.536757495Z g133     22
2020-03-01T21:25:16.220578776Z g133     20
2020-03-01T21:25:20.076926939Z g133     21
2020-03-01T21:25:25.483885732Z g133     22
2020-03-01T21:26:55.751372636Z g131     20
2020-03-01T21:27:00.157683684Z g131     25
2020-03-01T21:27:05.541633289Z g131     24
```

Modifier la requête pour visualiser uniquement les données de la salle G131 :

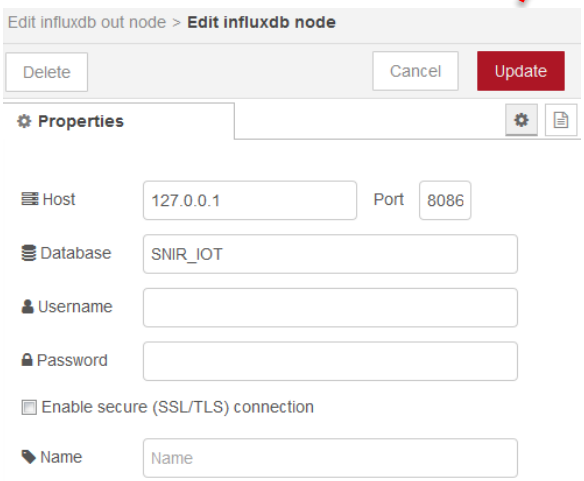
```
> select * from temperature where location='g131'
name: temperature
time                           location value
----                           -------- -----
2020-03-01T21:26:55.751372636Z g131     20
2020-03-01T21:27:00.157683684Z g131     25
2020-03-01T21:27:05.541633289Z g131     24
```

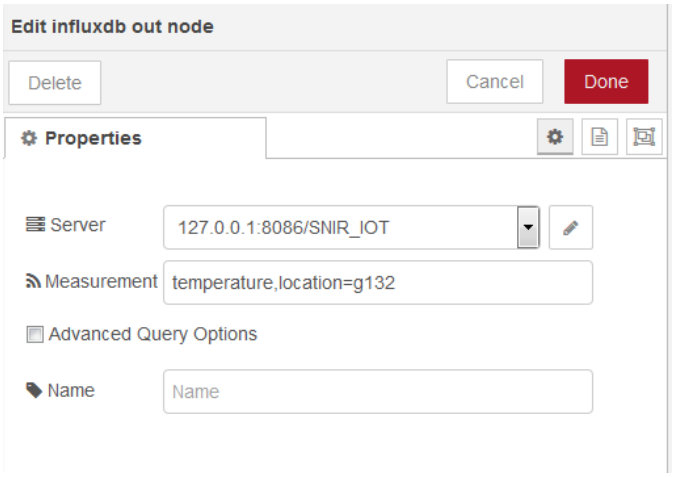
Mise en œuvre avec Node-Red

Les mesures que nous allons enregistrer proviennent d'un module esp8266. Il faut donc reprendre le flow précédent et y intégrer des enregistrements dans InfluxDB. Il faut aussi effectuer une conversion « string to int », car les données issues du broker mqtt est de type String or dans influxDB, il faut insérer des données de type INT.



Configuration du nœud InfluxDB :





Déployer le flow et vérifier que l'acquisition dans Node-Red et les enregistrements dans influxDB fonctionnent :

```
> select * from temperature
name: temperature
time                           location value
----                           -------- -----
2020-03-01T21:24:53.741546673Z g133     20
2020-03-01T21:25:00.417211725Z g133     21
2020-03-01T21:25:05.536757495Z g133     22
2020-03-01T21:25:16.220578776Z g133     20
2020-03-01T21:25:20.076926939Z g133     21
2020-03-01T21:25:25.483885732Z g133     22
2020-03-01T21:26:55.751372636Z g131     20
2020-03-01T21:27:00.157683684Z g131     25
2020-03-01T21:27:05.541633289Z g131     24
2020-03-01T21:52:14.991765908Z g132     21.6
2020-03-01T21:52:16.995132498Z g132     21.5
2020-03-01T21:52:18.99589614Z  g132     21.4
2020-03-01T21:52:20.998245352Z g132     21.3
2020-03-01T21:52:22.998062814Z g132     21.2
2020-03-01T21:52:24.998949079Z g132     21.1
2020-03-01T21:52:26.999584049Z g132     21
2020-03-01T21:52:29.000664283Z g132     21
2020-03-01T21:52:31.001459734Z g132     21.1
2020-03-01T21:52:33.002161337Z g132     21.2
```

Données insérées depuis node-red