

## 1 – Présentation de InfluxDB

InfluxDB est une base de données qui enregistre les informations en fonction du temps (*timestamp*). Elle permet notamment de définir une période d'existence des données après laquelle elles sont automatiquement effacées, ce qui n'est pas le cas pour une base de données comme MySQL, plus connue et plus adaptée aux bases de données relationnelles, comme des fiches clients par exemple.

InfluxDB est très utilisée dans le domaine des objets connectés.

Influxdb utilise 2 langages : InfluxQL et Flux. On se limitera ici à InfluxQL (Query Langage).

## 2 - Installation sur Raspberry

Dans le cadre des TP, la base de données sera installée sur une Raspberry. Dans le cadre du projet, la base de données sera installée sur un serveur externe (notre serveur par exemple).

**Lancer les commandes suivantes :**

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

**Editer le fichier influxdb.conf**

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

**Décommenter 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
```

**Démarrer le service InfluxDB**

```
sudo service influxdb restart
```

### 3 – Commandes en CLI (ligne de commande)

Pour obtenir l'invite de commande de InfluxDB, taper **influx** :

```
pi@raspberrypi:~ $ influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
>
```

#### Créer une base de données :

```
> CREATE DATABASE capteur_maison
```

#### Afficher les bases de données créées

```
> SHOW DATABASES
```

```
> SHOW DATABASES
name: databases
name
----
_internal
capteur_maison
```

#### Utiliser une base de données

```
> USE capteur_maison
```

#### Insérer des données dans une nouvelle table

```
> INSERT capteur_salon,ident=1234 temp=25,humidite=78,co2=450
```

```
> INSERT capteur_salon,ident=1234 temp=24,humidite=68,co2=350
```

Le format est le suivant :

```
INSERT <measurement>[,<tag-key>=<tag-value>...] <field-key>=<field-value>[,<field2-key>=<field2-value>...] [unix-nano-timestamp]
```

measurement correspond au nom de la nouvelle table (ici capteur\_salon)

tag correspond à une donnée non indexée (exemple une référence, un lieu, ...) (ici ident)

field correspond à une donnée indexée (ici temp, humidite et co2)

Un espace sépare les différents champs, une virgule pour les différents tags ou fields.

#### Afficher les différentes valeurs d'une table

```
> SELECT * FROM capteur_salon
```

```
> SELECT temp, humidite FROM capteur_salon
```

#### Afficher la date au format rfc3339 : YYYY-MM-DDTHH:MM:SS.nnnnnnnnnnZ

Quitter InfluxDB

```
> exit
```

Lancer influxDB avec l'option -precision rfc3339

\*

Visualiser le nouveau format du temps.

### Définir une politique de rétention

```
> CREATE RETENTION POLICY "un jour" ON "capteur_maison" DURATION 1d REPLICATION 1
```

Les données de la base de données capteur\_maison sont ici conservées 1 jour (1 day) et dupliquées 1 seule fois.

Le format est le suivant:

```
CREATE RETENTION POLICY <retention_policy_name> ON <database_name> DURATION <duration> REPLICATION <n>
[SHARD DURATION <duration>] [DEFAULT]
```

REPLICATION : duplication sur un autre support (mettre 1 si pas de duplication).

SHARD DURATION : partition des données. Par défaut les données sont mémorisées par groupe de :

RP Duration	Shard Group Duration
< 2 days	1 hour
>= 2 days and <= 6 months	1 day
> 6 months	7 days

DEFAULT : Définit la nouvelle politique de rétention comme politique de rétention par défaut pour la base de données. Facultatif.

### Afficher la politique de rétention

```
> SHOW RETENTION POLICIES ON capteur_maison
```

```
> SHOW RETENTION POLICIES ON capteur_maison
name      duration shardGroupDuration replicaN default
-----
autogen 0s      168h0m0s          1      true
un_jour 24h0m0s   1h0m0s           1      false
```

Autres commandes :

### Modifier une politique de rétention

```
ALTER RETENTION POLICY <retention_policy_name> ON <database_name> DURATION <duration> REPLICATION
<n> SHARD DURATION <duration> DEFAULT
```

### Supprimer une politique de rétention

```
DROP RETENTION POLICY <retention_policy_name> ON <database_name>
```

### Autres commandes

#### Effacer une base de données

```
DROP DATABASE "database_name"
```

#### Ajouter un utilisateur

```
CREATE USER name_user WITH PASSWORD 'motdepasse'
```

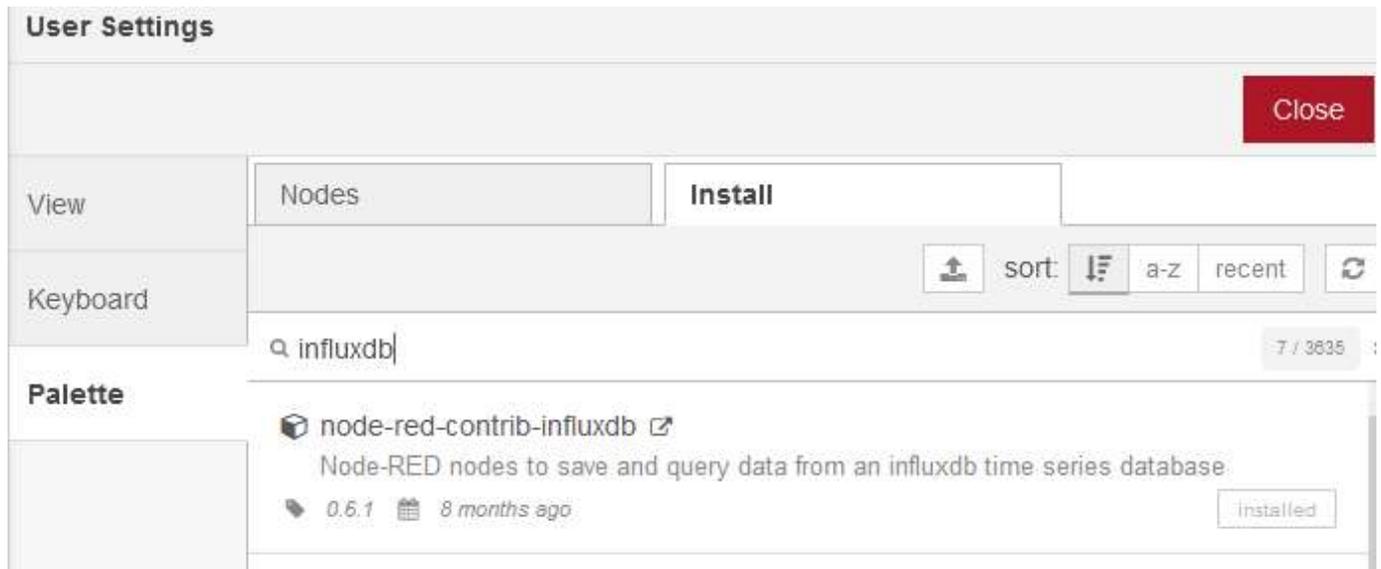
## Donner tous les droits à un utilisateur

```
GRANT ALL ON base_donnees TO name_user
```

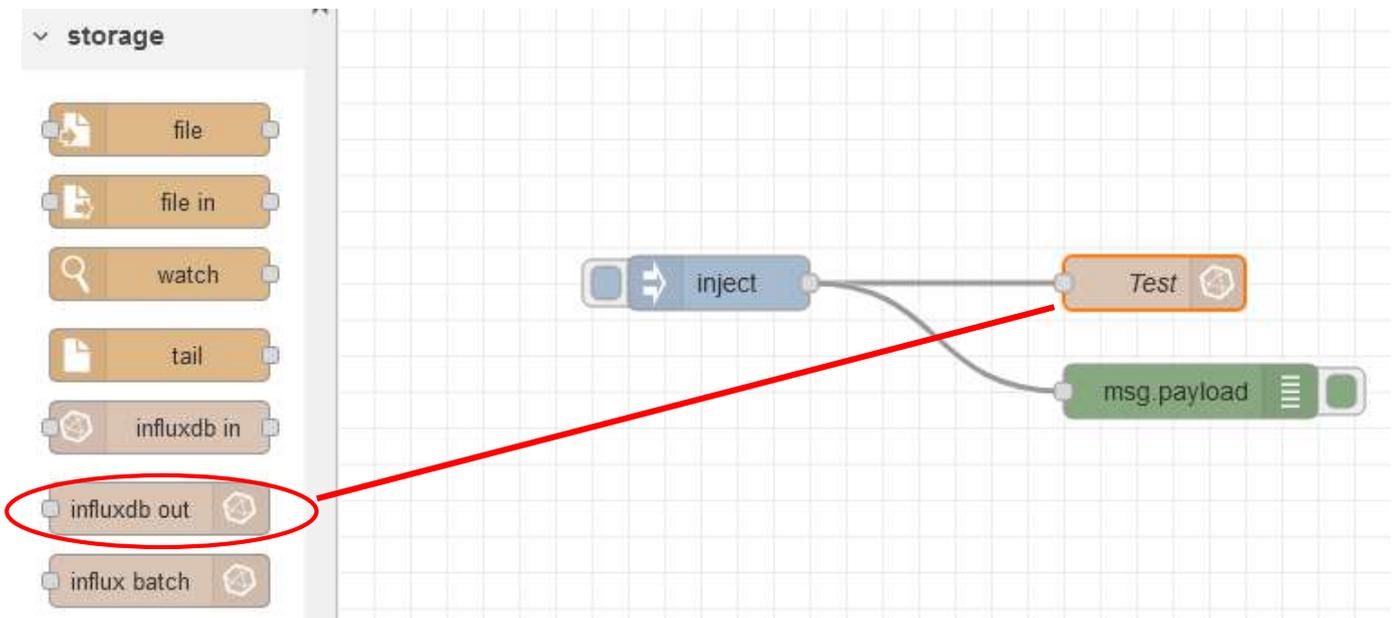
Autres commandes à voir sur Internet... [https://docs.influxdata.com/influxdb/v1.8/query\\_language/](https://docs.influxdata.com/influxdb/v1.8/query_language/)

## 4 – Enregistrer des données avec Node-RED

⇒ Sous Node-RED, dans le menu Manage palette, ajouter node-red-contrib-influxdb



⇒ Réaliser le flow suivant :



### Edit inject node

Delete Cancel

**Properties**

Name: Name

msg. payload = {} {"temp":35,"co2":510,"humidite":65}

### Edit inject node > JSON editor

Edit JSON

```

1 {
2   "temp": 35,
3   "co2": 510,
4   "humidite": 65
5 }
```

### Edit influxdb out node

Delete Cancel

**Properties**

Name: Test

Server: [v1.x] Test\_Influx

Measurement: capteur\_salon

Advanced Query Options

**Tip:** If no retention policy is specified, **autogen** will be assumed.

### Edit influxdb out node > Edit influxdb node

Delete Cancel

**Properties**

Name: Test\_Influx

Version: 1.x

Host: 127.0.0.1 Port: 8086

Database: capteur\_maison

Username:

Password:

Enable secure (SSL/TLS) connection

⇒ Injecter des données et observer la fenêtre DEBUG. Vérifier les données enregistrées dans la base de données, avec la commande SELECT dans une fenêtre CLI.

### debug

all nodes

04/01/2022, 21:06:29 node: 391f6e3d.9d7fe2

msg.payload: Object

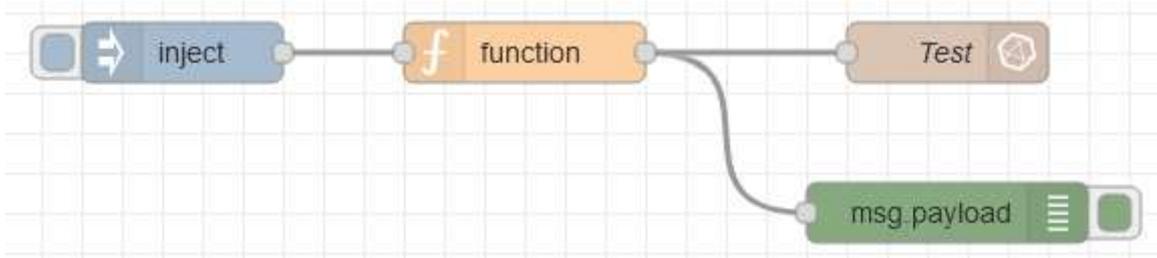
```

▶ { temp: 35, co2: 510, humidite: 65 }
```

```

> SELECT co2,humidite,temp FROM capteur_salon
name: capteur_salon
time                co2 humidite temp
-----
2022-01-03T16:40:36.661547477Z 450 78      25
2022-01-03T16:44:33.129241314Z 350 68      24
2022-01-03T17:43:05.687883119Z 350 68      24
2022-01-03T20:42:42.332448669Z          28
2022-01-03T20:46:41.203Z          35
2022-01-03T20:48:38.887Z           510 35
2022-01-03T20:59:39.69Z            510 35
2022-01-04T09:57:46.01Z            510 35
2022-01-04T18:11:49.651Z            510 35
2022-01-04T18:18:23.778Z            510 35
2022-01-04T18:18:56.751Z            510 35
2022-01-04T19:33:10.948Z            510 35
2022-01-04T19:33:42.69Z             510 35
2022-01-04T19:55:29.707Z            510 65 35
2022-01-04T20:02:37.822Z            510 65 35
2022-01-04T20:06:29.126Z            510 65 35
```

⇒ Ajouter une fonction pour enregistrer uniquement la température, comme ci-dessous.



**Edit function node**

Delete

**Properties**

Name

Setup On Start **On Message**

```
1 msg.payload={"temp":msg.payload.temp};
2 return msg;
```

⇒ Vérifier l'inscription dans la base de données.

⇒ Modifier la fonction pour enregistrer la température et l'humidité uniquement. Vérifier.

## 5 – Lire des données dans influxdb avec Node-RED

A la page précédente, ajouter le flow suivant ( fonction : influxdb in)

The image shows a Node-RED flow with three nodes connected in a line: a blue 'timestamp' node, a brown '[v1.x] Test\_Influx' node, and a green 'msg.payload' node. Below the flow are three edit panels:

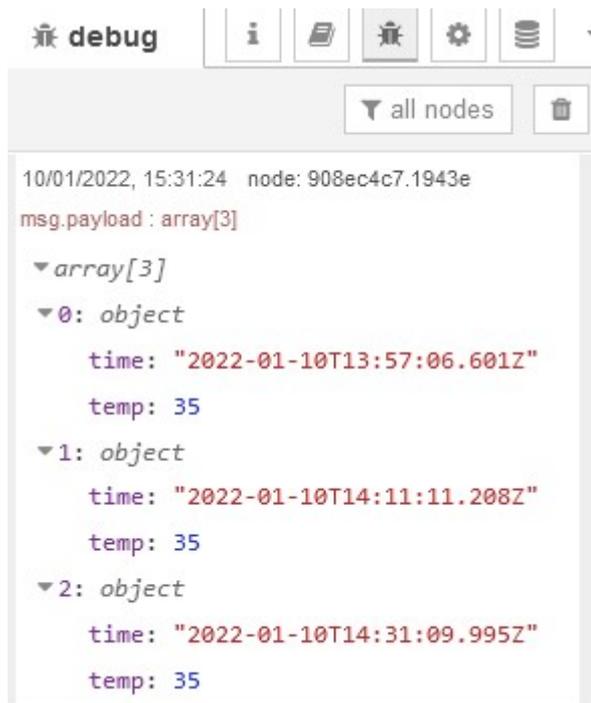
- Edit inject node:** Shows a 'Delete' button and a 'Properties' section with a 'Name' field and two mappings: 'msg.payload' mapped to 'timestamp' and 'msg.topic' mapped to 'a\_z'.
- Edit debug node:** Shows a 'Delete' button and a 'Properties' section with an 'Output' dropdown set to 'msg.payload', a 'To' section with 'debug window' checked, and a 'Name' field.
- Edit influxdb in node:** Shows a 'Delete' button, 'Cancel', and 'Done' buttons. The 'Properties' section includes a 'Name' field, a 'Server' dropdown set to '[v1.x] Test\_Influx', and checkboxes for 'Raw Output' and 'Advanced Query Options'. The 'Query' section contains the SQL query: 

```
1 SELECT temp FROM capteur_salon WHERE time > '2022-01-10T13:57:00Z'
```

Dans l'édition de Query ci-dessus, seules les mesures enregistrées après le 10/01/2022 à 13h57mn00s (heure UTC) seront sélectionner.

⇒ Adapter la date avec la date courante dans l'édition de Query

⇒ Observer les valeurs lues dans la fenêtre DEBUG comme ci-dessous :



⇒ Modifier Query pour afficher l'humidité également.

## 6 – Liens utiles

<https://bentek.fr/influxdb-grafana-raspberry-pi/>

<https://projetsdiy.fr/tutoriel-influxdb-installation-configuration-macos/>

[https://archive.docs.influxdata.com/influxdb/v1.1/introduction/getting\\_started/](https://archive.docs.influxdata.com/influxdb/v1.1/introduction/getting_started/)