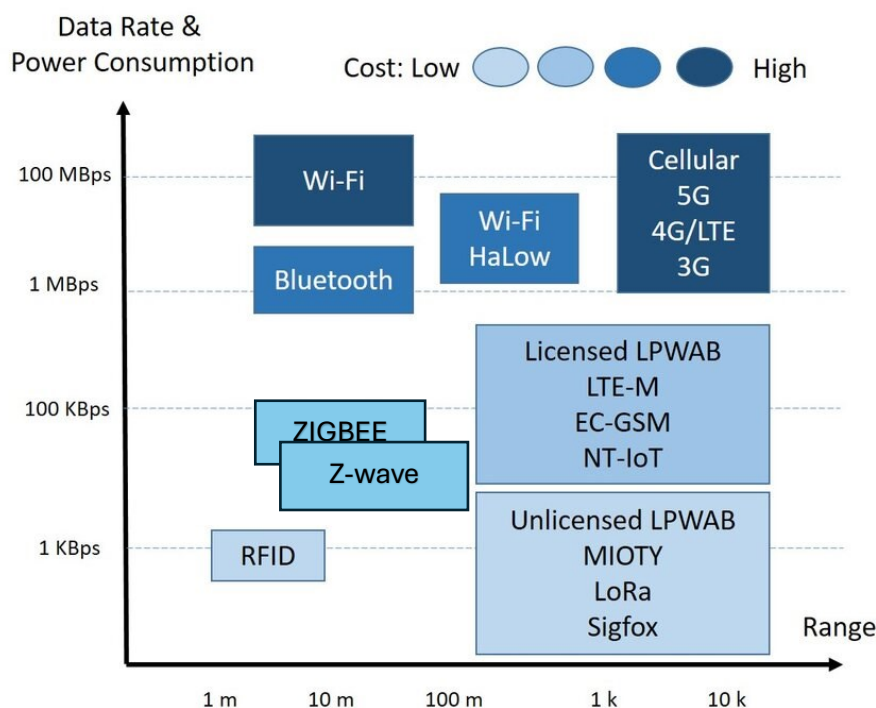


# 1 Les objets connectés HF

Il existe une multitude de protocoles disponibles pour les objets connectés HF à connecter sur une box :

- ✓ Z-wave (868MHz) faible consommation, prix entre 30 et 100 euros, réseau maillé possible pour étendre la distance , <https://www.domo-blog.fr/home-assistant-protocole-z-wave-guide/>
- ✓ Zigbee (2.4GHz), faible consommation, moins efficace en distance que Z-wave, prix entre 10 et 50 euros, nécessite une box, réseau maillé, très utilisé ( Samsung SmartThings, Amazon Echo Plus et Echo Show, Belkin WeMo, Philips Hue, ikea) avec l'arrivée de Matter (intégration via bluetooth, puis zigbee),
- ✓ Wifi (2.4GHz), consommation importante (sur secteur), prix entre 10 et 50 euros
- ✓ Lorawan (868MHz), très faible consommation, prix entre 30 et 100 euros et très longue portée
- ✓ WmBus (169MHz), très longue portée (utiliser pour le relevé d'information compteurs gaz par exemple)

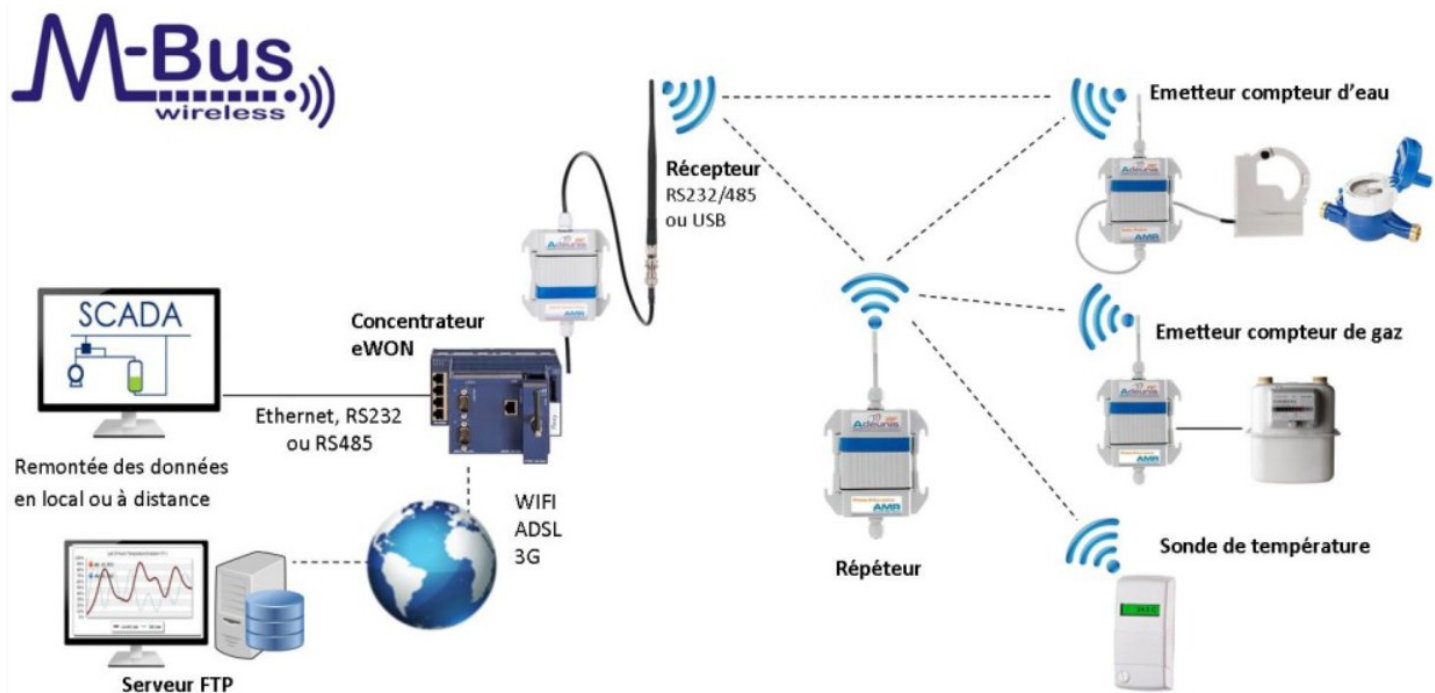
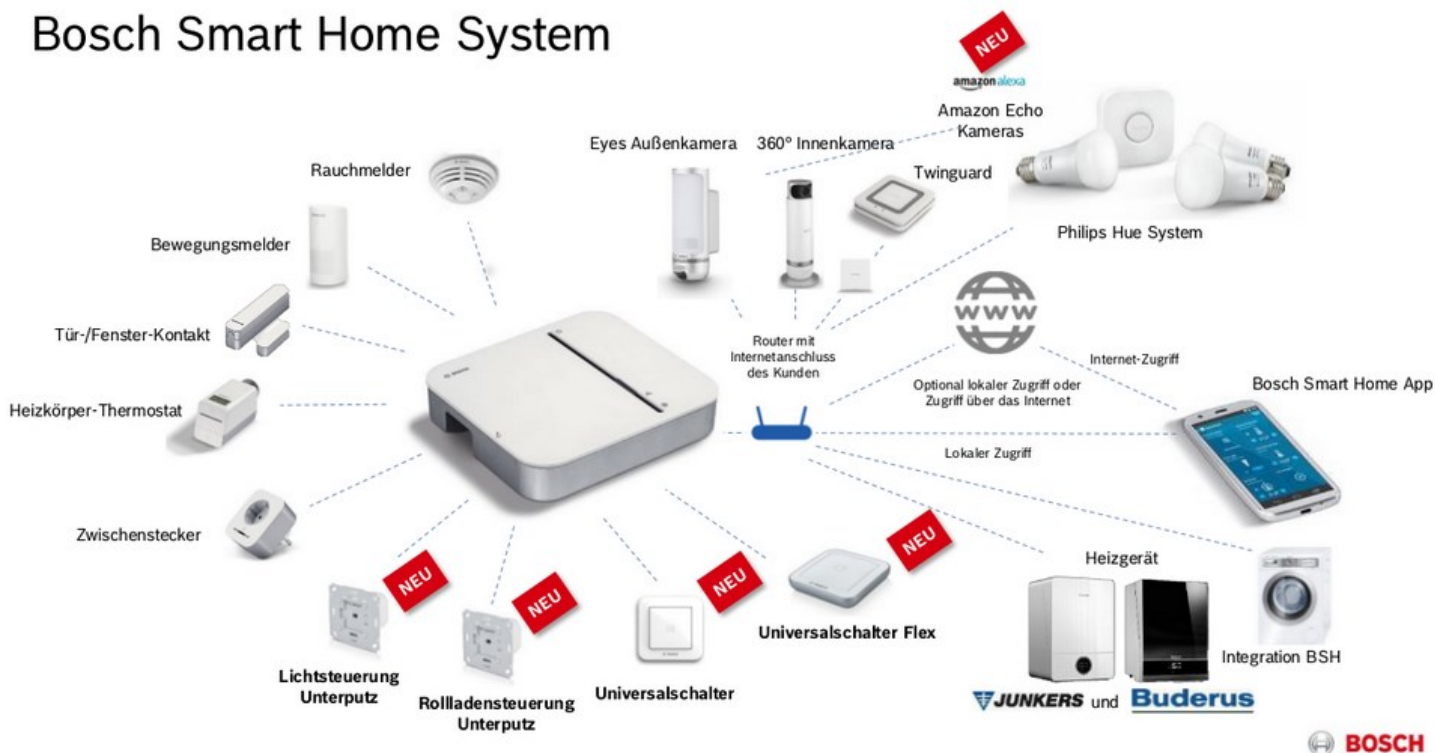
Une autre solution consiste à utiliser la 4G ou 5G (la 2G/3G est abandonné fin 2026) et une carte sim pas n'a pas besoin de box. Cette solution plus chère (nécessite un abonnement) a l'avantage de pouvoir être inséré dans les batiments et à l'extérieur des batiments.



Quelques exemples de cas d'usage :

- ✓ Campagne de mesures de températures dans un bâtiment, ajout de capteurs d'énergie sur radiateur, intégration de capteurs sur une aire d'autoroute . Pour ces solutions ou il n'y a pas de prises 230V à proximité du capteurs, le matériel est en location et il n'est pas prévu de travaux d'installation, les distances entre les capteurs et la box est assez importante : lorawan est la solution la plus intéressante (sur pile les matériels peuvent fonctionner plusieurs années)
- ✓ Mesures dans un local chauffage au sous-sol : dans ce cas, l'accès à une prise électrique est possible, on pourra utiliser la solution WmBus ou Lorawan qui permettent de traverser le béton pour accéder à la télérelève.
- ✓ Automatisation d'une villa : si la solution KNX n'est pas possible, on pourra utiliser les solutions zigbee-matter ou z-wave que l'on peut intégrer aux box Alexa, google home,...
- ✓ Pour les solutions de domotisation à faible cout on utilisera les E/S wifi ou zigbee.

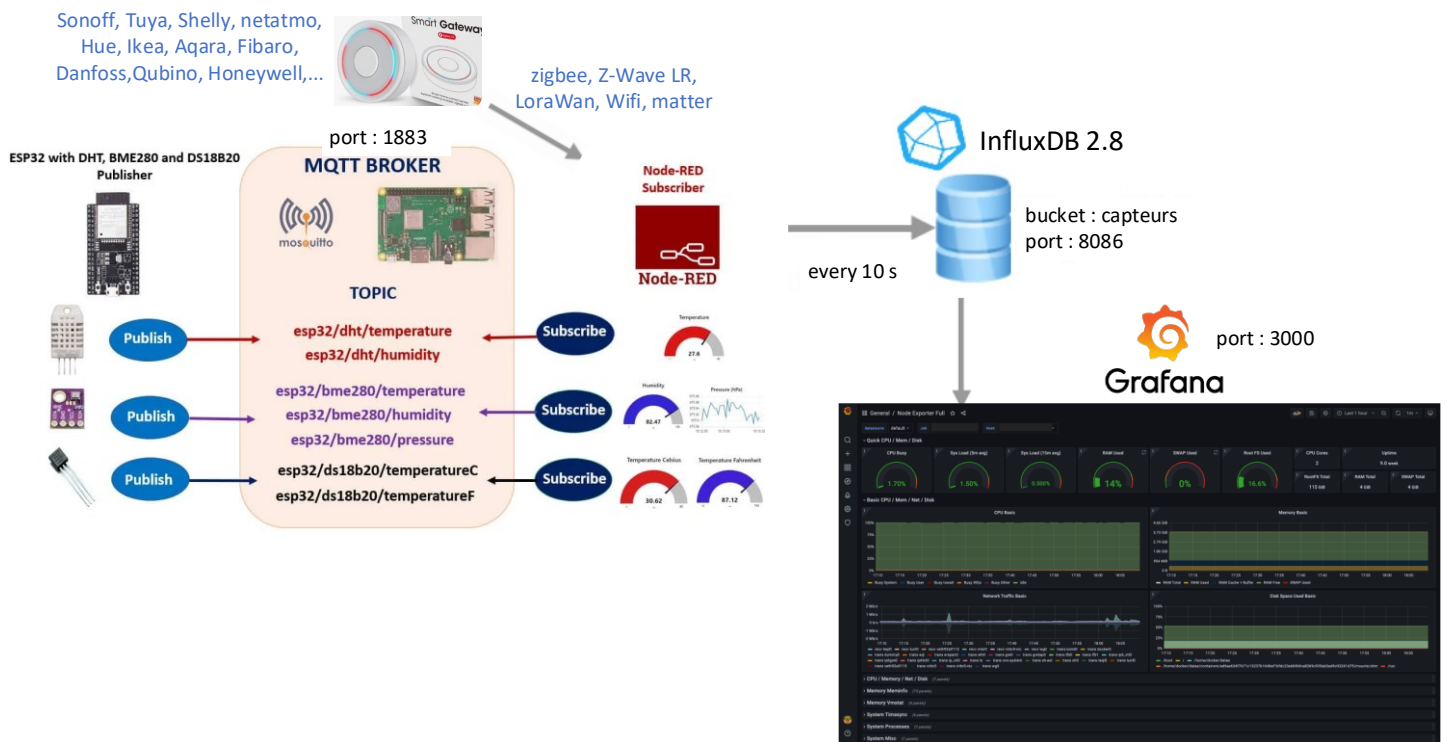
# Bosch Smart Home System



En domotique, 2 logiciels sont très utilisés :

- ✓ Home Assistant qui permet l'intégration d'un grand nombre de capteurs et actionneurs sans fil (Z-wave, Zigbee, Wifi, ...) mais aussi une intégration de KNX via l'interface IP.
- ✓ Node-red qui permet l'intégration de différents capteurs et actionneurs et de transférer ces données au travers différents protocoles vers le cloud, un affichage en local une intégration dans une base de données,...

Et pour terminer on retrouve très souvent le protocole mqtt qui permet d'envoyer ou de recevoir des données de capteurs ou actionneurs.



Afin de comprendre une architecture réseau IOT, nous allons configurer nodered, influxdb et grafana et nous terminerons par la récupération de données capteurs venant d'un client mqtt et que nous afficherons sur grafana.

## 2 Influxdb 2.8

### 2.1 Introduction

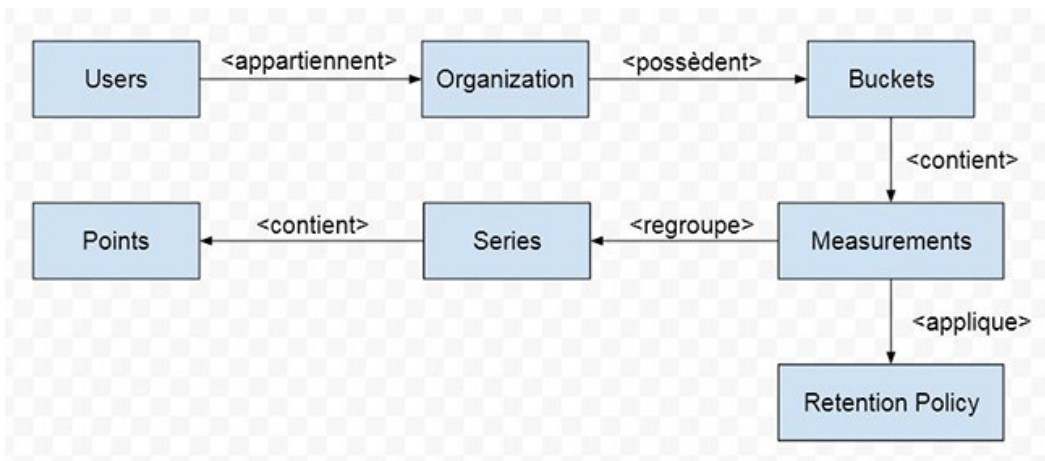
InfluxDB v2 est une base de données orientée séries temporelles (TSDB). Elle est conçue pour stocker, analyser et visualiser des données horodatées :

- ✓ capteurs
- ✓ métriques système
- ✓ IoT
- ✓ monitoring

Base de données temporelle (influxdb)	Base de données relationnelle (sql)
Bucket (avec durée de rétention configurable)	Base de données
Measurement	table
Tags	Champs indexés
Fields	Champs non indexés
Timestamp : élément indispensable	

#### Principes de fonctionnement d'influxdb :

- ✓ les utilisateurs appartiennent à des organisations qui sont donc les entités de plus haut niveau ;
- ✓ les Buckets appartiennent à des organisations, ainsi les utilisateurs ne peuvent accéder qu'aux Buckets de leur organisation. Les Buckets peuvent être assimilés aux « bases » des SGBDR ;
- ✓ les Buckets contiennent des mesures (*measurement*) qui peuvent être vues comme des tables (au sens SGBDR). Les mesures contiennent des points. La syntaxe formelle de définition d'un point est :



Pour insérer des données dans influxdb, il faut préciser (avec avoir choisi le bucket) la table (measurement puis les tags et enfin les données pour finir avec le timestamp.

```
<measurement>[,<tag_key>=<tag_value>[,<tag_key>=<tag_value>]*] <field_key>=<field_value>[,<field_key>=<field_value>]* [<timestamp>]
```

\*(il y a un espace entre la liste des tags et la liste des fields) ;

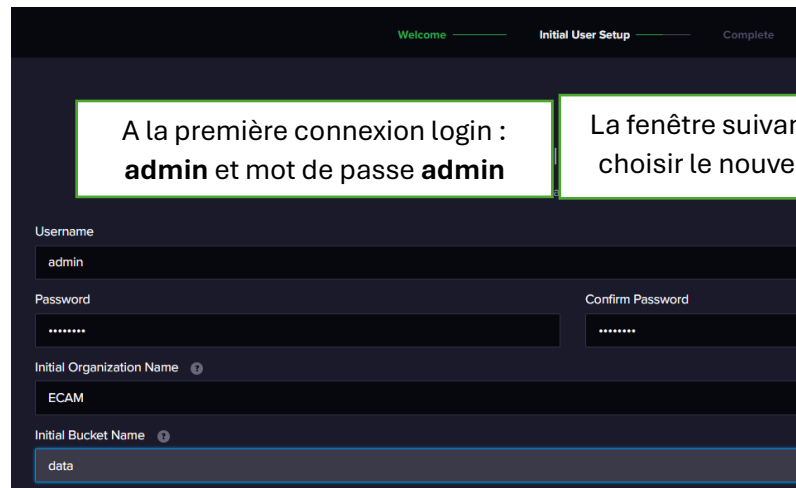
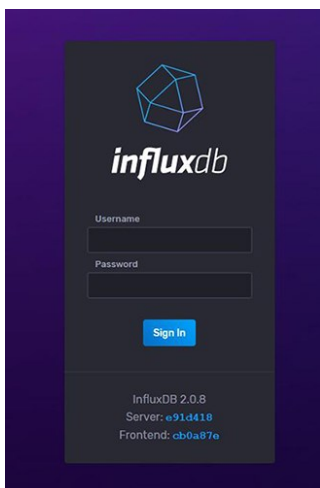
- ✓ un point est donc constitué d'un ensemble estampillé de tags optionnels et de champs valués (au moins un champ est requis). Si l'estampille est manquante lors de l'insertion du point, la date courante du serveur est utilisée ;
- ✓ pour une mesure donnée, les points qui partagent les mêmes valeurs de tags constituent une **série** ;
- ✓ les **Buckets** sont associés à une politique de rétention des données, ce qui permet de supprimer automatiquement les données obsolètes.

**Exemple** : nous imitons des données météorologiques qui seraient fournies par des capteurs géographiquement répartis sur le territoire français. Nous pouvons créer des points de mesure ainsi :

```
meteo,station="Toulouse" t=273.5,pmer=100830i 1556813561098000000
meteo,station="Lyon" t=275.2,pmer=100570 1556813561218000000
meteo,station="Toulouse" t=274.8,pmer=100640i 1556814842127000000
```

La mesure se nomme ici **meteo**. Il y a 2 séries, car le tag **station** a 2 valeurs différentes : Toulouse et Lyon. Chaque point contient 2 champs : la température **t** et pression au niveau de la mer **pmer**. Les points sont estampillés (au format Unix timestamp et exprimés en nanosecondes par défaut).

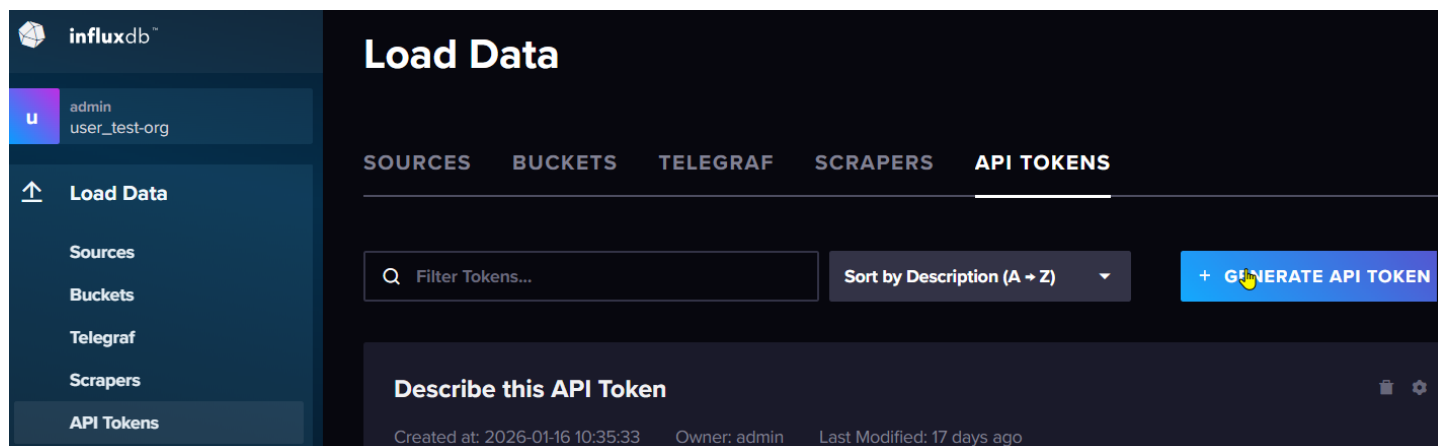
A l'installation d'influxdb 2.8, à la première connexion sur <http://217.154.16.99:8086/> on configure le mot de passe de l'utilisateur principal ( qui a tous les droits), qui va nous permettre de gérer l'administration d'influxdb



A la première connexion login : **admin** et mot de passe **admin**

La fenêtre suivante : obligation de choisir le nouveau mot de passe

**Remarque :** Toutes les communications vers influxdb passent soit par un login et mot de passe, soit pas un token, qu'il est possible de générer et qu'il faut noter une fois généré (impossible ensuite de le relire, il faudra en créer un autre si celui-ci n'a pas été noté.



C'est ce qui a été fait à la première connexion par l'utilisateur admin (super utilisateur d'influxdb)

## 2.2 Organisation influxdb

Chaque utilisateur est associé à une organisation et est propriétaire de cette organisation : par exemple barrat2025 est propriétaire de l'organisation barrat2025-org, ce qui permet de compartimenter chaque base (bucket) afin que chaque étudiant ait son espace de développement propre.

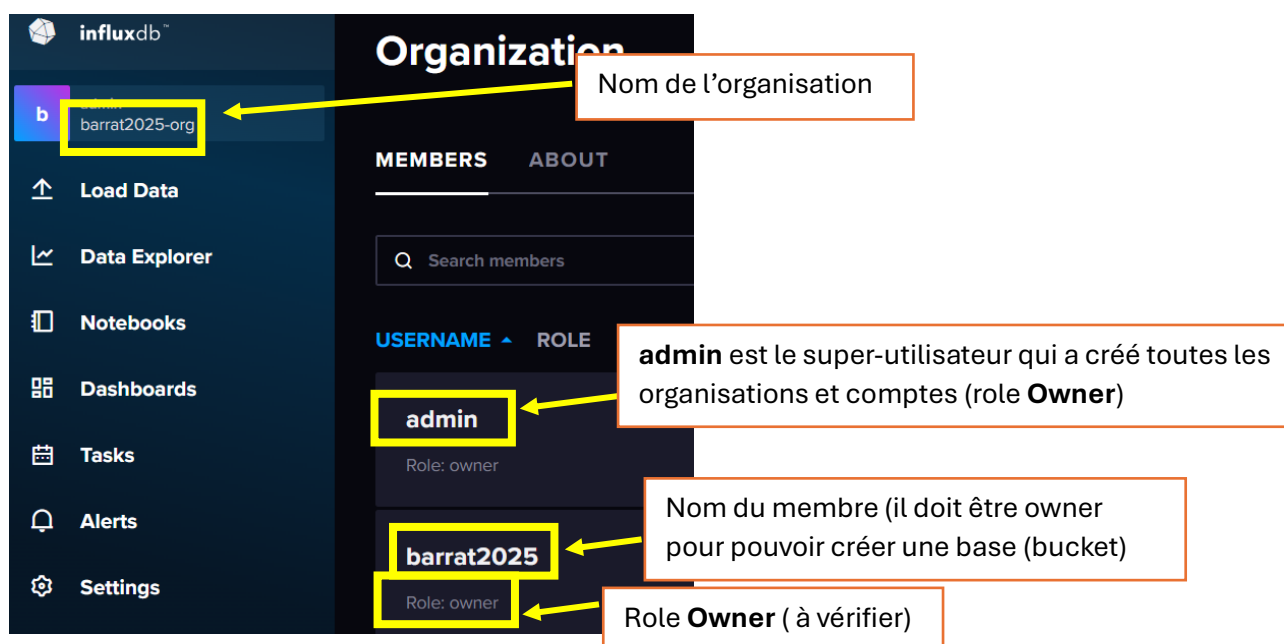
C'est l'administrateur (admin) qui a créé d'abord l'organisation pour chaque étudiant, puis l'étudiant avec son mot de passe et enfin l'insertion de l'utilisateur dans l'organisation name-example.org (en tant que Owner).

Pour information, voici les lignes de commandes qui ont permis la création de chaque utilisateur.

```
influx org create --name example-org --token xxxxxxxx(token de l'admin)
influx user create --name name-example --password "123456789" --token xxxxxxxx(token de l'admin)
influx org members add --name example-org --member id_name-example--owner --token xxxxxxxx(token de l'admin)
```

Prenons l'exemple de barrat2025, celui-ci va se connecter sur <http://217.154.16.99:8086/> et se connecter avec son login : barrat2025 et son mot de passe 123456789.

Une fois connectée il pourra vérifier (clic sur b) dans Organisation qu'il fait parti de l'organisation barrat2025-org en tant que owner (comme l'utilisateur admin qui est le créateur de l'organisation).



## 2.3 Création de la base de données (bucket)

Vous allez vous connecter via le navigateur sur le serveur web géré par influxdb.

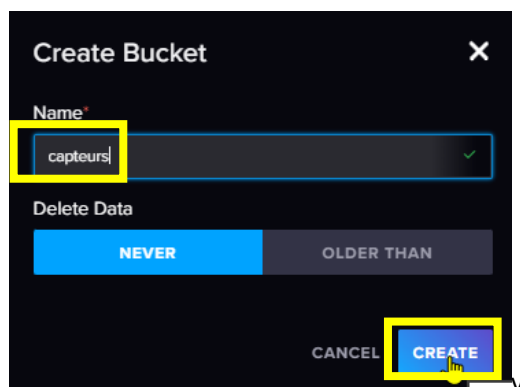
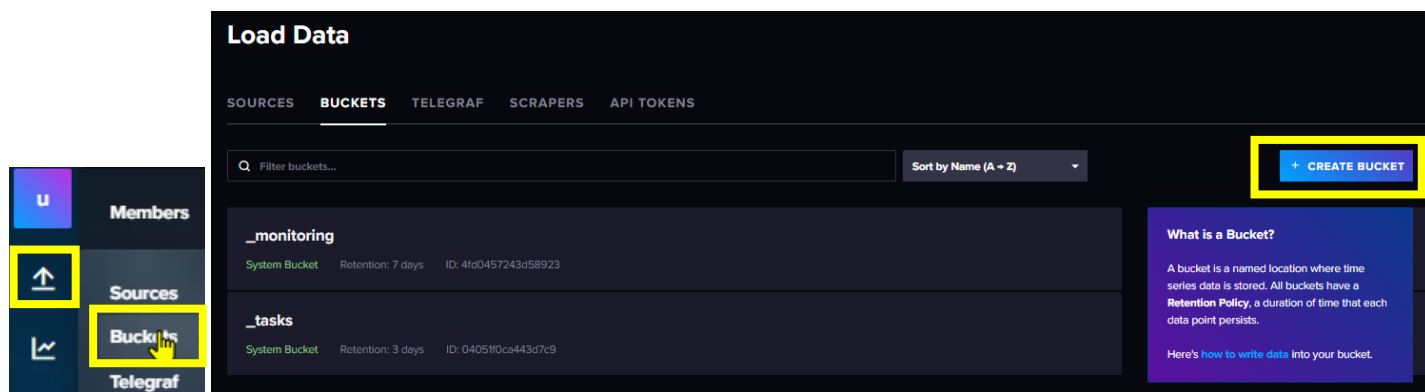
1. Connectez-vous sur <http://217.154.16.99:8086/> et vérifiez que vous êtes le propriétaire de votre organisation (nécessaire pour créer un bucket (Cf page précédente))

Placer votre login (sauf verrier2025 qui est verrier) et mot de passe : 123456789

**Remarque :** si le serveur ne répond pas, vérifiez que le service a bien été lancé (utiliser putty)

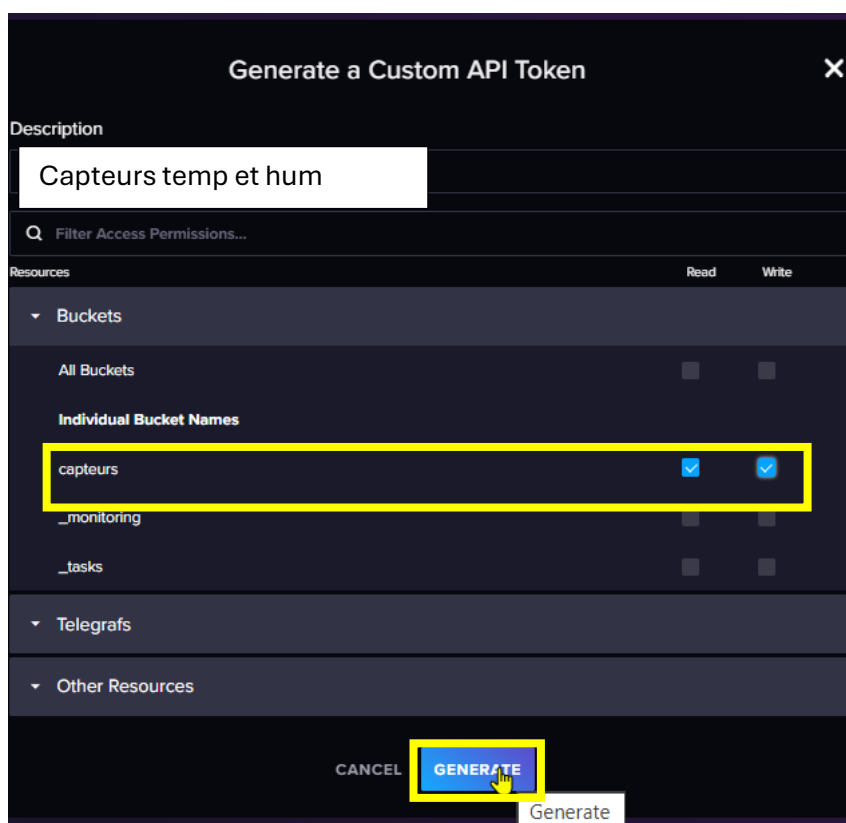
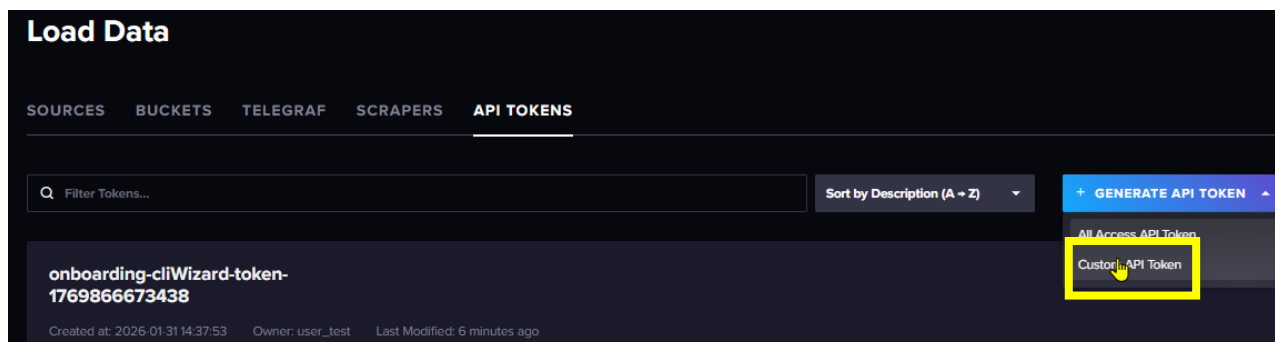
```
user_test@my-vps:~$ systemctl status influxdb
● influxdb.service - InfluxDB is an open-source, distributed, time series data
   Loaded: loaded (/lib/systemd/system/influxdb.service; enabled; preset: en
   Active: active (running) since Sat 2026-01-24 20:48:37 UTC; 6 days ago
     Docs: https://docs.influxdata.com/influxdb/
   Process: 426 ExecStart=/usr/lib/influxdb/scripts/influxd-systemd-start.sh
   Main PID: 433 (influxd)
     Tasks: 8 (limit: 2193)
    Memory: 72.5M
         CPU: 2min 33.182s
    CGroup: /system.slice/influxdb.service
           └─433 /usr/bin/influxd
```

2. Vous allez maintenant créer le bucket (la base) qui va accueillir les données capteurs

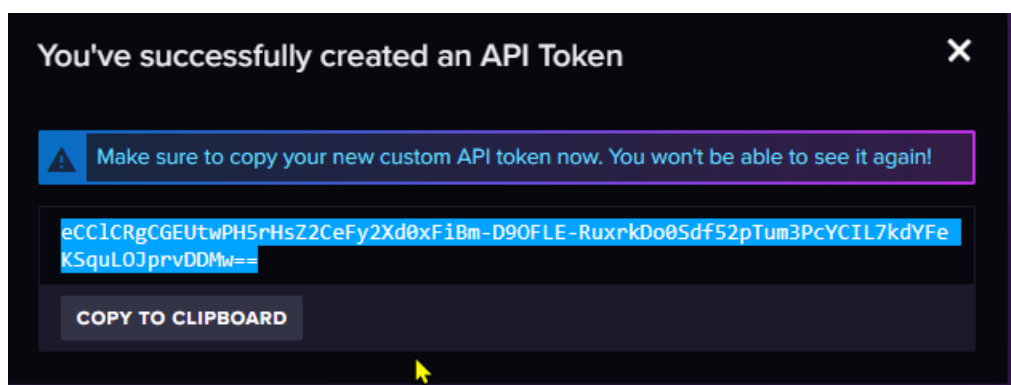


Pour se connecter à votre bucket, il faudra soit utiliser votre login et mot de passe (peu utilisé) soit utiliser votre token. C'est cette deuxième solution que nous allons privilégier.

3. Créer le token associé à votre bucket et copier ce token. Si vous le perdez, il faudra recréer un nouveau token, celui que vous allez créer n'est visible qu'à la création, il est impossible ensuite de le récupérer autrement qu'en l'effaçant et en recréant un nouveau. (PS : nous aurions pu créer un **all access Token** qui permet de gérer tous les buckets créé, mais nous allons utiliser un **custom token** qui ne sera associé qu'au bucket capteurs, c'est ce que l'on fait en général pour des raisons de sécurité)



4. Copier cette clé dans le bloc note pour pouvoir l'utiliser ensuite dans node-red



C'est terminé pour la configuration de votre base de données. Vous pouvez maintenant utiliser cette base pour écrire des données (Cf documentation <https://docs.influxdata.com/influxdb/v2/write-data/>)

## 2.4 Vous allez maintenant tester la communication avec node-red

5. Lancer node-red (dans putty) et connectez-vous sur `http://217.154.16.99:votre-port-node-red/` puis installer le plugin influxdb.



Code de la fonction	Type de variables												
<pre>msg.payload = [{   temperature: +(18 + Math.random() * 6).toFixed(2),   humidite: +(40 + Math.random() * 20).toFixed(2) }], {   site: "bureau",   capteur: "DHT22" }]; return msg;</pre>	<table border="1"><thead><tr><th>Élément</th><th>Valeur</th></tr></thead><tbody><tr><td>bucket</td><td>capteurs</td></tr><tr><td>measurement</td><td>meteo</td></tr><tr><td>tags</td><td>site, capteur</td></tr><tr><td>fields</td><td>temperature, humidite</td></tr><tr><td>time</td><td>automatique</td></tr></tbody></table>	Élément	Valeur	bucket	capteurs	measurement	meteo	tags	site, capteur	fields	temperature, humidite	time	automatique
Élément	Valeur												
bucket	capteurs												
measurement	meteo												
tags	site, capteur												
fields	temperature, humidite												
time	automatique												

### 6. Configurer influxdb V2

The image shows two screenshots of the Node-RED configuration interface for the InfluxDB v2 node. The left screenshot shows the 'Propriétés' (Properties) section with the following values: Nom: InfluxDB v2, Server: [v2.0] influx, Organization: user\_test-org, Bucket: capteurs, Measurement: meteo, and Time Precision: Milliseconds (ms). The right screenshot shows the 'Propriétés' section with the following values: Nom: influx, Version: 2.0, URL: http://localhost:8086, Token: [redacted], and Connection timeout (seconds): 10. Red arrows point from the 'user\_test-org' field to a red box labeled 'Votre organisation', from the 'Token' field to a red box labeled 'Votre token', and from the '10' field to a red box labeled 'Envoie des données toutes les 10s'.

## 2.5 Questions de compréhension

Toutes les questions de compréhension sont en gras (répondez à la suite)

**✍️ Pour l'url de connexion à influxdb, on pourrait utiliser localhost, 127.0.0.1 ou l'adresse 217.154.16.99, expliquer la différence entre les 3 solutions et pourquoi l'utilisation de localhost ou 127.0.0.1 est préférable.**

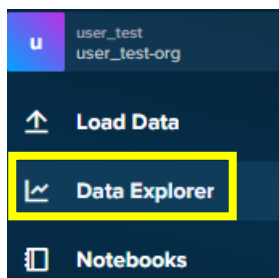
**✍️ Chercher sur internet comment se créer en javascript un tableau et un objet. Donner un exemple de création d'un tableau de 2 éléments qui pourrait être dans msg.payload.**

**✍️ Quelle est le type de variable (tableau et/ou objet) qui doit être envoyé à influxdb pour insérer un élément dans sa base ?**

**✍️ On désire envoyer 1 élément dans la base courant et 1 élément dans la base tension, quel serait le message à envoyer dans msg.payload.**

## 2.6 Tester votre base sur influxdb

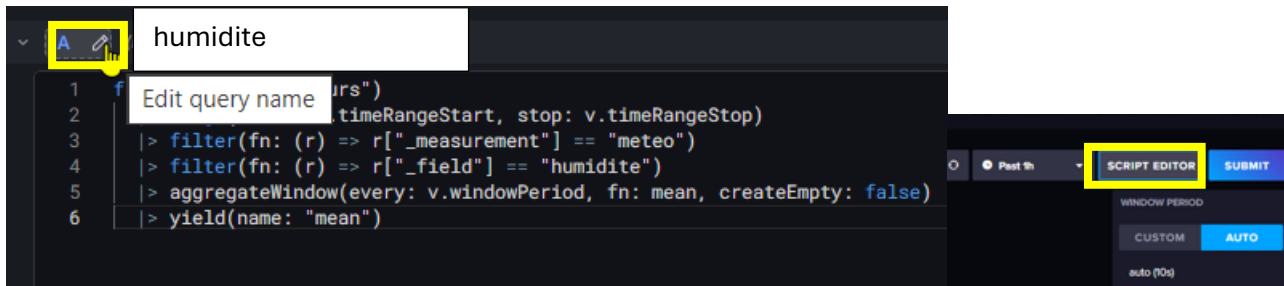
Si node-red est fonctionnel, les données humidite et temperature sont directement envoyées à influxdb. Si la connexion est valide (pas d'erreur lors du déploiement sur node-red), vous pouvez aller visualiser les données qui sont envoyés toutes les 10s (Data Explorer dans Influxdb ui).



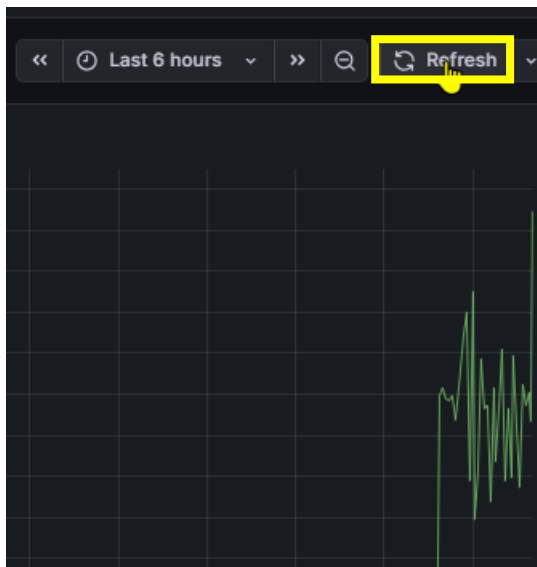
Aller dans **Data Explorer** et aller chercher les données d'humidité



Vous pouvez cliquer sur Script Editor pour visualiser la requête flux qui a été générée, et changer le nom de la requête (exemple pour l'humidité)




Vous pouvez aussi modifier le temps de visualisation (par exemple toutes les 15mn)



Il est possible (solution proposée dans la documentation <https://docs.influxdata.com/influxdb/v2/write-data/>) d'insérer des données dans la base à partir du client influx (qui se trouve déjà installé sur la machine 217.154.16.99). Lancer la commande dans putty.

```
influx write \  
--bucket capteurs \  
--org nom_org \  
--token ton_koken \  
'test_measurement,capteur=salon temperature=21.5  
test_measurement,capteur=cuisine temperature=22.1'
```

 **vérifier que la donnée température dans la table test\_measurement existe. Donner ici le code générer de lecture de cette température**

 **Arrêter node-red et ajouter via la commande influx 1 valeurs intégrant une température de 100°C et une humidité de 100%. Relancer node\_red et visualiser sur le site influx que ca a fonctionné. Montrer à l'enseignant.**

## 3 grafana :

### 3.1 Introduction

**Grafana** est un **outil open-source de visualisation, d'analyse et d'alerting** de données, principalement orienté **données temporelles**.

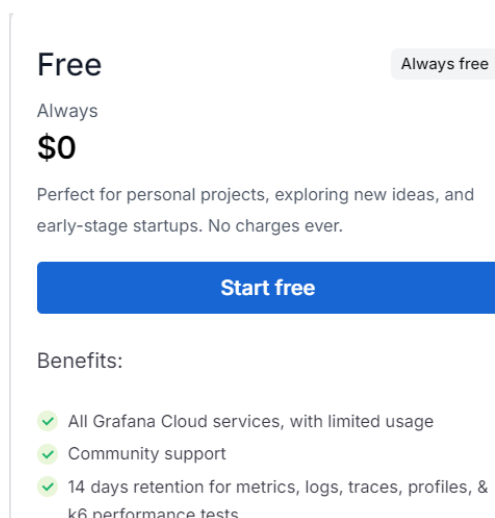
👉 Son rôle principal est de :

- transformer des données brutes en **tableaux de bord visuels**
- faciliter la **compréhension de l'évolution dans le temps**
- permettre la **surveillance** et la **détection d'anomalies**

Grafana est aujourd'hui un standard dans les domaines :

- du **monitoring informatique**
- de l'**IoT**
- de l'**industrie**
- de la **data temps réel**

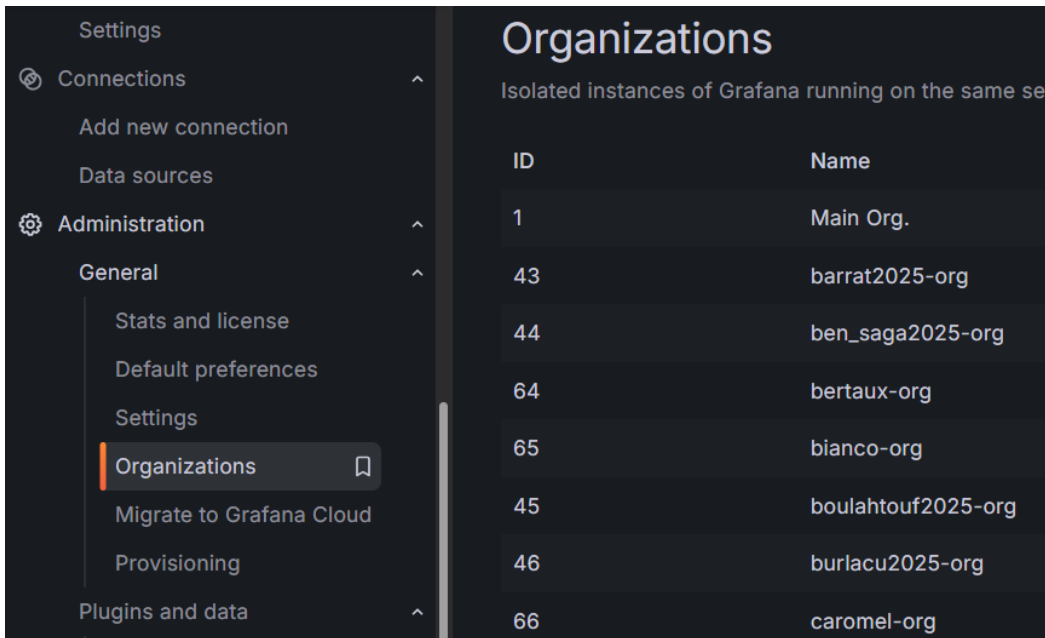
Il est possible d'utiliser un compte de test sur Grafana Cloud <https://grafana.com/products/cloud/>



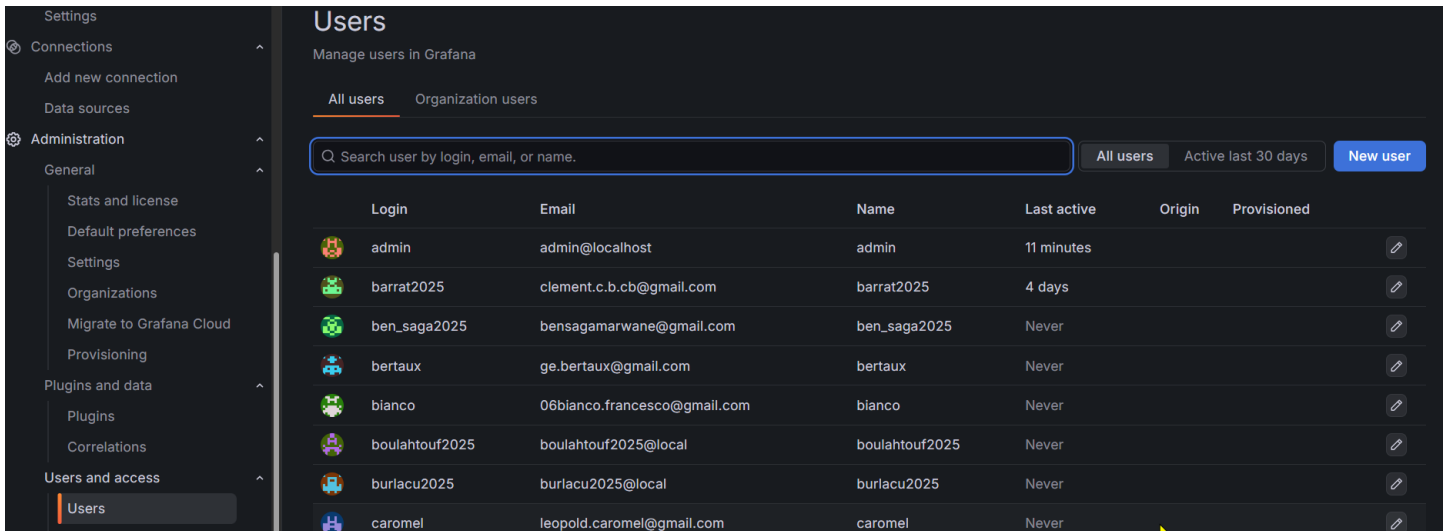
The screenshot shows the 'Free' tier of Grafana Cloud. It features a 'Free' label with a 'Always free' badge. Below this, it says 'Always \$0' and describes the tier as 'Perfect for personal projects, exploring new ideas, and early-stage startups. No charges ever.' A prominent blue 'Start free' button is visible. Underneath, a 'Benefits:' section lists three items with green checkmarks: 'All Grafana Cloud services, with limited usage', 'Community support', and '14 days retention for metrics, logs, traces, profiles, & k6 performance tests'.

Ce n'est pas la solution que nous allons tester ici (mais elle reste tout à fait possible), nous allons utiliser **Grafana** sur <http://217.154.16.99:3000/>. Comme InfluxDB, tout se passe lors de la première connexion, par défaut, les identifiants de connexion sont **admin** pour le nom d'utilisateur et **admin** pour le mot de passe. À la première connexion, on vous demandera de changer le mot de passe par défaut (cette partie a déjà été faite).

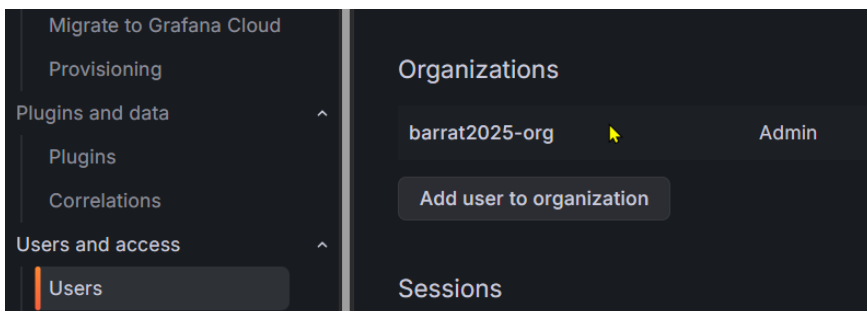
Comme pour Influxdb, Grafana travaille avec des organisations et des users. Il est possible en tant qu'administrateur de créer les organisations sur le site :



Chaque étudiant a ensuite été créé



Puis ensuite il a été associé à son organisation en tant qu'admin (chaque étudiant a tous les droits sur son organisation).



### 3.2 Première connexion :

1. Connectez-vous sur Grafana <http://217.154.16.99:3000/>

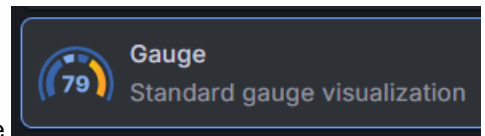
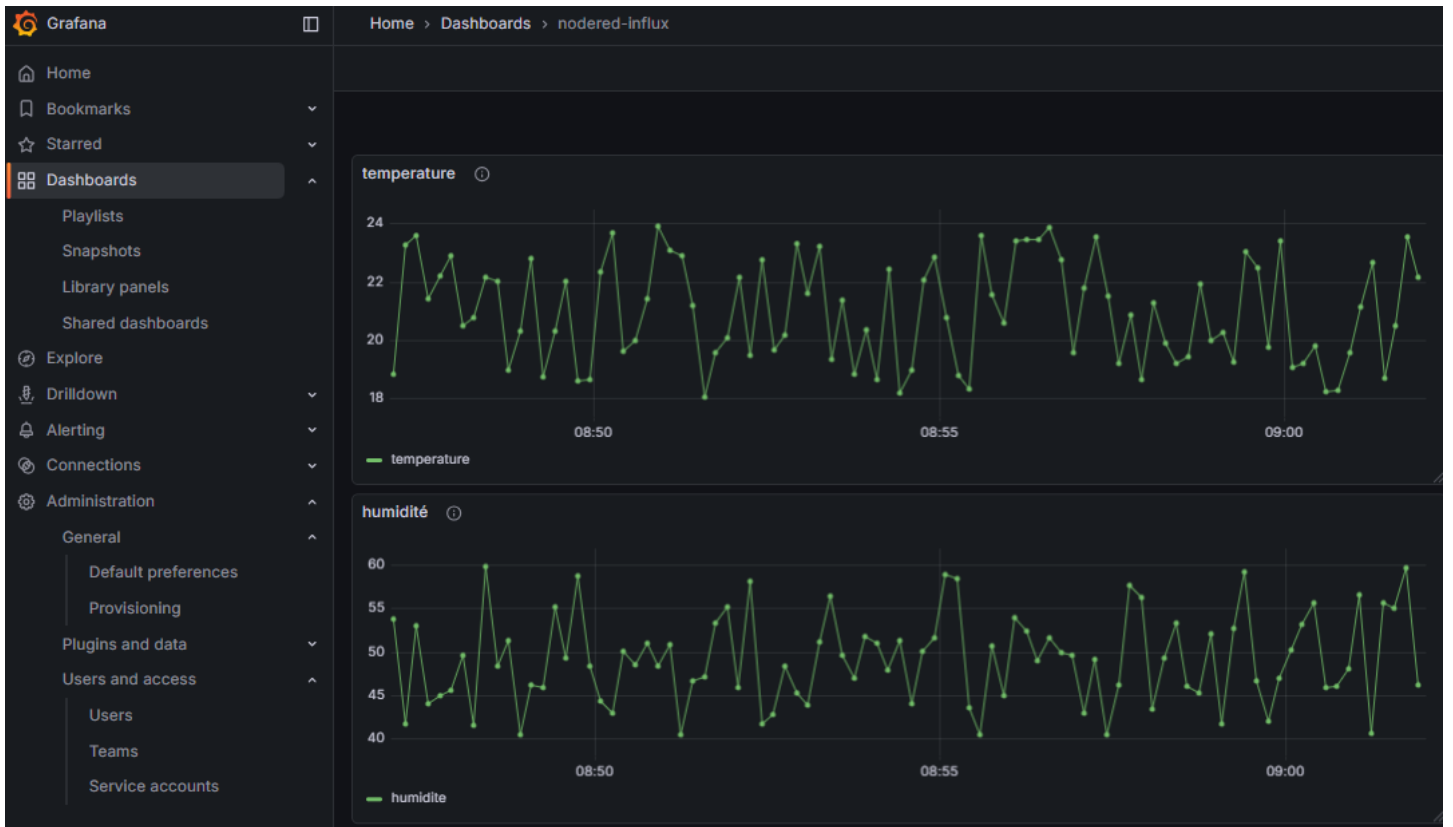
login : verrier2025

mot de passe : 123456789

Vous allez ajouter une connexion vers la base de données influxDB. Vous avez 2 solutions de connexion vers influxdb , solution 1 via votre login et mot de passe influxdb ou solution 2 : via le token influxdb (ici les 2 solutions ont été saisies mais une seule est suffisante).

The image shows two side-by-side screenshots of the Grafana web interface. The left screenshot displays the 'Add new connection' screen where the 'InfluxDB' data source is selected. The right screenshot shows the configuration form for the InfluxDB data source, with fields for 'User', 'Password', 'Organization', 'Token', and 'Default Bucket'. A red box highlights the 'User' and 'Password' fields, with a red arrow pointing to the text 'Votre login et mot de passe influxdb (solution 1)'. Another red box highlights the 'Organization' field, with a red arrow pointing to the text 'Votre organisation finluxdb'. A third red box highlights the 'Token' field, with a red arrow pointing to the text 'Votre token influxdb (solution 2)'. A fourth red box highlights the 'Default Bucket' field, with a red arrow pointing to the text 'La table (bucket)'. At the bottom of the right screenshot, there are 'Delete' and 'Save & test' buttons. Below the screenshots, a green notification box displays the message: '✓ datasource is working. 1 buckets found. Next, you can start to visualize data by building a dashboard , or by querying data in the Explore view .'

1. Dans le menu latéral gauche, cliquez sur l'icône "+" et sélectionnez "Dashboard".
2. Choisissez entre "Add new panel" (ajouter un nouveau panneau)
3. Si vous choisissez "Add new panel", sélectionnez la source de données que vous avez ajoutée précédemment et choisissez un type de visualisation (par exemple, graphique, jauge, tableau, etc.).
4. Configurez le panneau en fonction de vos besoins, en sélectionnant les métriques, les filtres et les options de visualisation appropriés.
5. Cliquez sur le bouton "Apply" pour appliquer les modifications et visualiser le panneau.
6. Répétez les étapes 3 à 5 pour ajouter d'autres panneaux au tableau de bord.
7. Enregistrez le tableau de bord en cliquant sur l'icône "Save dashboard" (disquette) dans le coin supérieur droit.



8. Utiliser une gauge pour afficher la température et ajouter un titre

**Thresholds**

+ Add threshold

- 30 🗑️
- 15 🗑️
- Base

**Standard options**

Unit: Celsius (°C) ✕

Min: 0  
Leave empty to calculate based on all values

Max: 50  
Leave empty to calculate based on all values

9. Ajouter une stat

10. ajouter le treshold

**Visualization**

12.4 Stat ▼

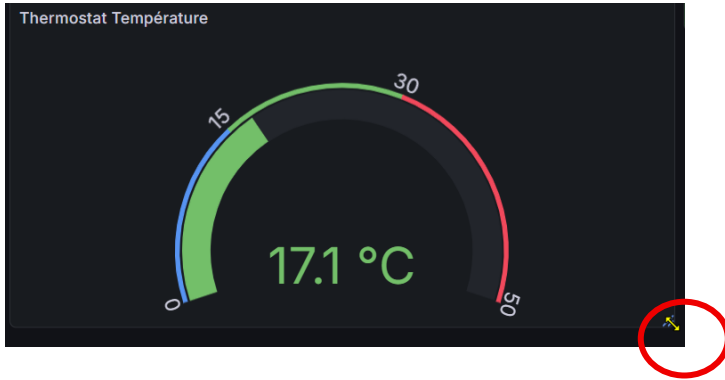
**Graph mode**  
Stat panel graph / sparkline mode

None Area

+ Add threshold

- 30 🗑️
- 25 🗑️
- 20 🗑️
- 0 🗑️
- Base

## 11. Modifier la taille du panneau



## 12. Ajouter un autre panel de type stat et modifier le value mappings

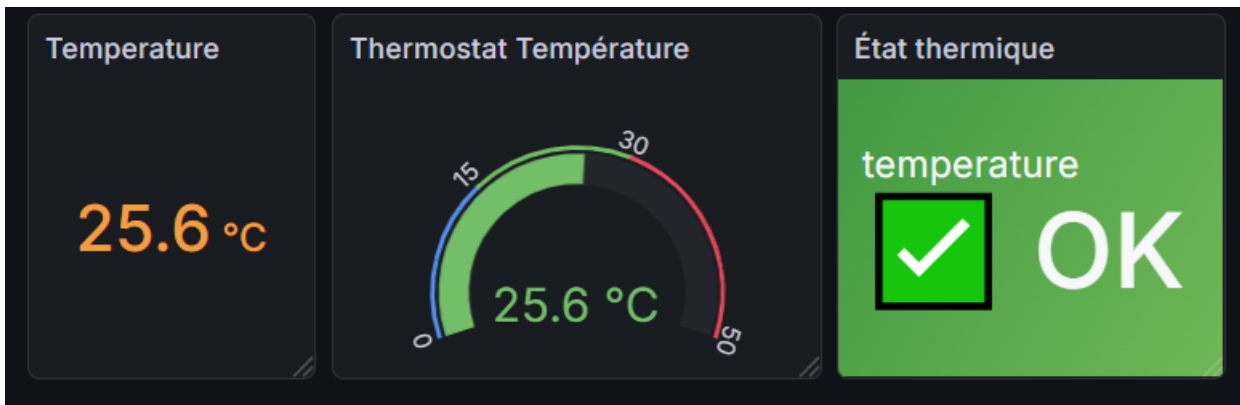
Value mappings

Condition	Display text	Color
Range 30,01 100	Trop chaud	Red
Range 15 30	OK	Green
Range -100 14,99	Trop froid	Blue

+ Add a new mapping

Cancel Update

Vous devriez obtenir :



 **Montrer à l'enseignant**

<https://www.youtube.com/watch?v=7t62iu3W7Y4>

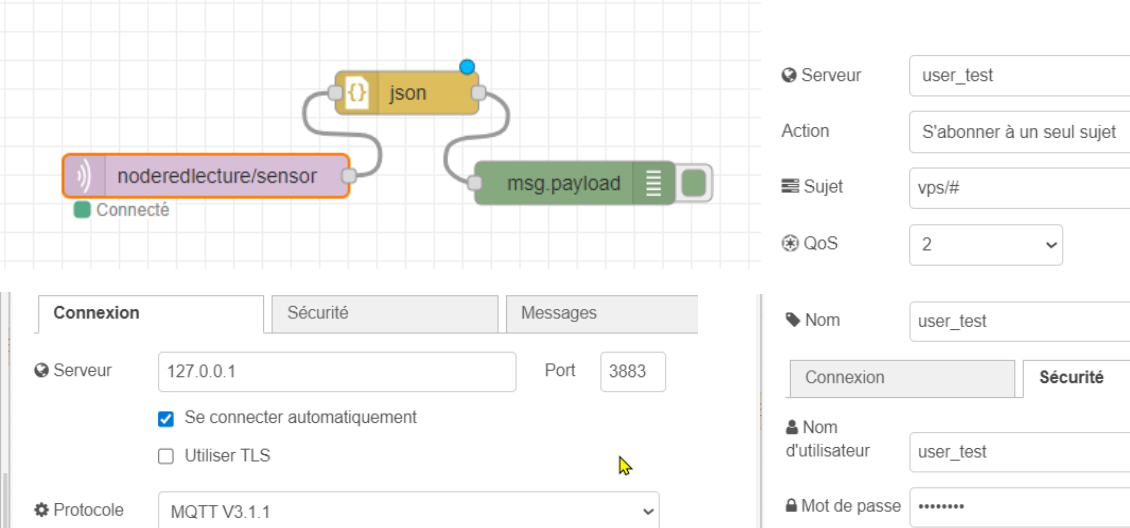
<https://www.youtube.com/playlist?list=PLDGkOdUX1Ujo27m6qiTPPCpFHVfyKq9jT>

<https://www.youtube.com/watch?v=aUq85rp7yQU&list=PLVCgi5HZ0-Yt5UGLObhh6rHHtye0xKKFp&index=7>

### 3.3 Intégration d'une base de données mysql-mqtt-grafana

Nous allons maintenant insérer les données reçues d'un capteurs et venant de mqtt dans la base de données mysql puis afficher ces données sur Grafana.

 **Reprendre et tester la communication de nodered du TD3 (partie 3.4) et vérifier que les données mqtt sont bien reçues.**



The screenshot shows the Node-RED interface. On the left, a flow is visible with three nodes: 'noderedlecture/sensor' (purple), 'json' (yellow), and 'msg.payload' (green). The 'noderedlecture/sensor' node is connected to the 'json' node, which is connected to the 'msg.payload' node. The 'noderedlecture/sensor' node is also connected to the MQTT connection settings panel. The MQTT connection settings panel is open, showing the following configuration:

- Server: 127.0.0.1
- Port: 3883
- Se connecter automatiquement:
- Utiliser TLS:
- Protocole: MQTT V3.1.1
- QoS: 2
- Nom d'utilisateur: user\_test
- Mot de passe: 1234

Pour se connecter au serveur mqtt il faudra utiliser le login `user_test` et le mot de passe `1234` (un seul user pour mqtt, il ne faut pas mettre votre login et mot de passe).

 **Connectez-vous à votre base de données et créer la table `mqtt_data`**

 Attention l'exemple ci-dessous est donné pour l'utilisateur `user_test01`. Vous devez vous connecter sur votre base de données qui se nomme `db_votre_login`.

Connexion pour l'utilisateur `user_test01`

```
user_test@my-vps:~$ mysql -u user_test01 -p
```

(votre mot de passe 1234)

Voir toutes les bases mysql disponibles sur le serveur mariadb

```
MariaDB [(none)]> show databases ;
```

Choisir sa base de données

```
MariaDB [(none)]> use db_user_test01;
```

Voir toutes les tables de la base

```
MariaDB [db_user_test]> show tables;
```

Visualiser tous les éléments de la table Animal par exemple

```
MariaDB [db_user_test]> select * from Animal;
```

 **Créer la table mqtt\_data qui va accueillir les données mqtt qui vont être insérer dans nodered.**

```
CREATE TABLE mqtt_data (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  time TIMESTAMP NOT NULL,  
  device VARCHAR(50),  
  CPU FLOAT,  
  MEM FLOAT,  
  RAMP FLOAT,  
  TEMP FLOAT,  
  INDEX idx_device_time (device, time)  
);
```



Tester l'insestion

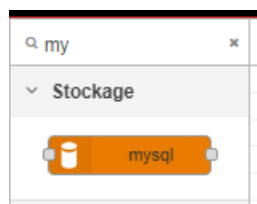
```
INSERT INTO mqtt_data (time,device, CPU, MEM, RAMP, TEMP)  
VALUES (NOW(),"vps", 2, 3, 4, 5);
```

Tester la lecture

```
Select * from mqtt_data ;
```

 **Vous allez maintenant vous connecter sur votre base de données au travers de node-red**

 node-red-node-mysql   
3.0.0  
2 noeuds



Ou se trouve le serveur mariadb

Le port par défaut du serveur

Modifier le noeud mysql > Modifier le noeud MySQLdatabase

Supprimer Annuler Sauver

Propriétés

Nom db\_user\_test

Host 127.0.0.1

Port 3306

User Votre login mqtt

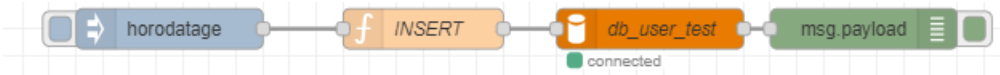
Password Votre mot de passe mqtt

Database Votre base mqtt

Timezone ±hh:mm

Charset UTF8

## Puis ajouter la fonction d'insertion



```
const time = new Date(); // Date JS

msg.topic = `
INSERT INTO mqtt_data (time, device, CPU, MEM, RAMP, TEMP)
VALUES (?, ?, ?, ?, ?, ?)
`;

msg.payload = [
  time,
  'vps',
  10,
  11,
  12,
  13
];

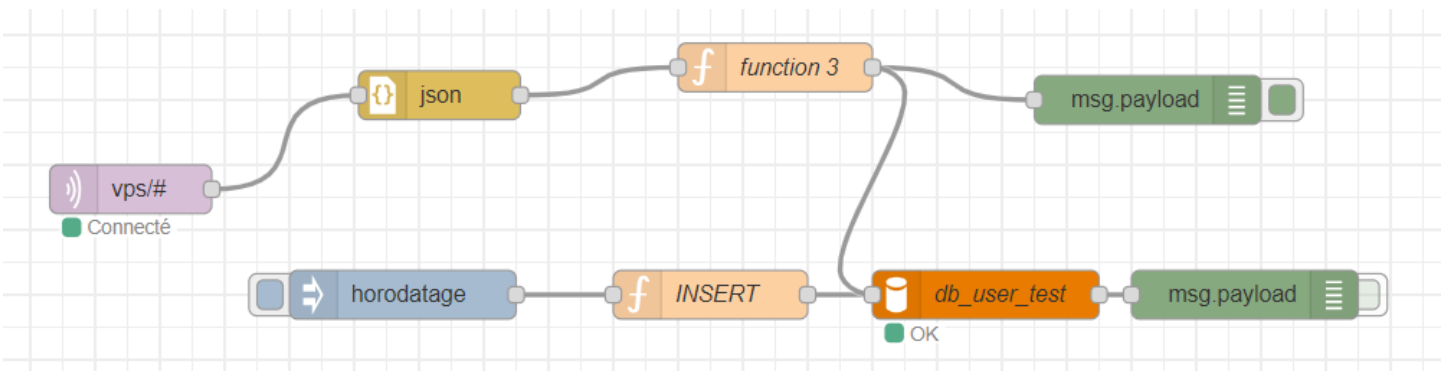
return msg;
```

## Aller dans putty et vérifier que la table mqtt\_data a bien une ligne supplémentaire après avoir cliquer sur inject dans node-red

Nous allons maintenant assembler les 2 fonctions :

- ✓ Lecture via mqtt des données capteurs
- ✓ Intégration de ces valeurs dans votre base de données dans la table mqtt\_data

## Insérer fonction3 et tester son fonctionnement



**Remarque :** 4 données sont envoyées en mqtt, le topic intègre l'information publiée et le payload la valeur publiée. Il y a 4 messages qui sont publiées et nous voulons intégrer ces 4 messages dans une requête mqtt. Il faut donc utiliser une variable globale que nous allons appeler mqtt\_data.

msg.topic	" vps/cpu"	" vps/mem"	" vps/ramp"	" vps/temp"
Msg.payload	Valeur_cpu	Valeur_mem	Valeur_ramp	Valeur_temp

Code de la fonction 3 ;

```
// msg.topic = "vps/cpu" ou "vps/mem" etc.
// msg.payload = string "0", "40.1", etc.

let data = flow.get('mqtt_data') || {};

// On récupère la valeur en float
let value = parseFloat(msg.payload);

if (msg.topic.endsWith("/cpu")) {
  data.CPU = value;
} else if (msg.topic.endsWith("/mem")) {
  data.MEM = value;
} else if (msg.topic.endsWith("/ramp")) {
  data.RAMP = value;
} else if (msg.topic.endsWith("/temp")) {
  data.TEMP = value;
}

// On met à jour l'objet stocké dans le flow
flow.set('mqtt_data', data);

// Vérifier si toutes les valeurs sont présentes
if (data.CPU !== undefined && data.MEM !== undefined && data.RAMP !== undefined && data.TEMP !== undefined)
{
  const time = new Date();
  msg.topic = `
  INSERT INTO mqtt_data (time, device, CPU, MEM, RAMP, TEMP)
  VALUES (?, ?, ?, ?, ?, ?)
  `;
  msg.payload = [time, 'vps', data.CPU, data.MEM, data.RAMP, data.TEMP];

  // Reset les valeurs pour le prochain cycle
  flow.set('mqtt_data', {});

  return msg; // envoi vers MySQL
} else {
  // pas encore toutes les valeurs
  return null;
}
```



**Aller dans putty et vérifier que la table mqtt\_data est mis à jour par les données mqtt**

### 3.4 Insertion dans grafana

Nous allons maintenant ajouter la visualisation de ces données dans grafana. Il faut tout d'abord ajouter une nouvelle source de données de type mysql.

## Créer une nouvelle connexion vers votre base de données mysql

### Add new connection

Browse and create new connections

Search

 × Clear

### Connection

Host URL \*

Database name


### Authentication


Username \*

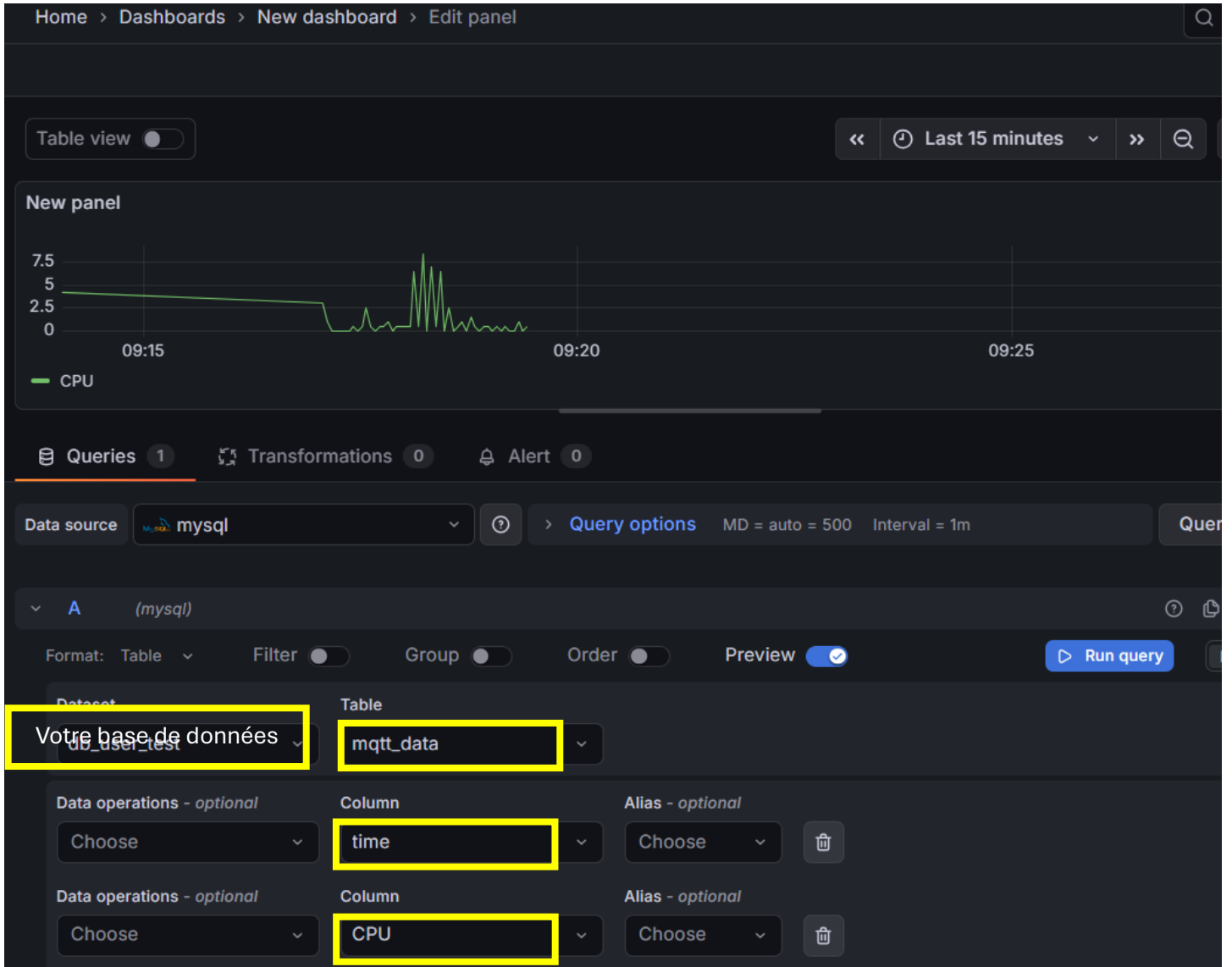
Password

✓ Database Connection OK

Next, you can start to visualize data by [building a dashboard](#) , or by querying data in the [Explore](#)

Delete Save & 

 Puis tester l'insertion de la données CPU par exemple



Home > Dashboards > New dashboard > Edit panel

Table view

Last 15 minutes

New panel

7.5  
5  
2.5  
0

09:15 09:20 09:25

CPU

Queries 1 Transformations 0 Alert 0

Data source mysql

Query options MD = auto = 500 Interval = 1m

(mysql)

Format: Table Filter Group Order Preview  Run query

Dataset: Votre base de données

Table: mqtt\_data

Data operations - optional: Choose

Column: time

Alias - optional: Choose

Data operations - optional: Choose

Column: CPU

Alias - optional: Choose

 Insérer les 3 autres données