

1 - Installation sur Raspberry

Dans le cadre du TP on utilisera une raspberry comme serveur. Dans le cadre d'un projet on utilisera grafana installé sur un serveur externe.

Vérifier que grafana est installé sur la raspberry en tapant dans la barre d'adresse de l'explorateur Internet sur PC : ip_raspberry:3000

Le login et le mot de passe sont par défaut : admin et admin
(skip pour ne pas modifier le mot de passe)

Si grafana n'est pas installé, la procédure à suivre est la suivante :

• Installer Grafana sur raspberry

```
wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -  
echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list  
sudo apt-get update  
sudo apt-get install -y grafana
```

• Lancer le logiciel grafana

```
sudo /bin/systemctl enable grafana-server  
sudo /bin/systemctl start grafana-server
```

• Accéder au menu grafana

Dans la barre d'adresse de l'explorateur Internet sur PC : ip_raspberry:3000

Le login et le mot de passe sont par défaut : admin et admin
(skip pour ne pas modifier le mot de passe)

2 – Créer une base de données

Pour tester grafana, nous allons utiliser la même base de données que celle créée lors du TP influxDB.

Base de données : capteur_maison

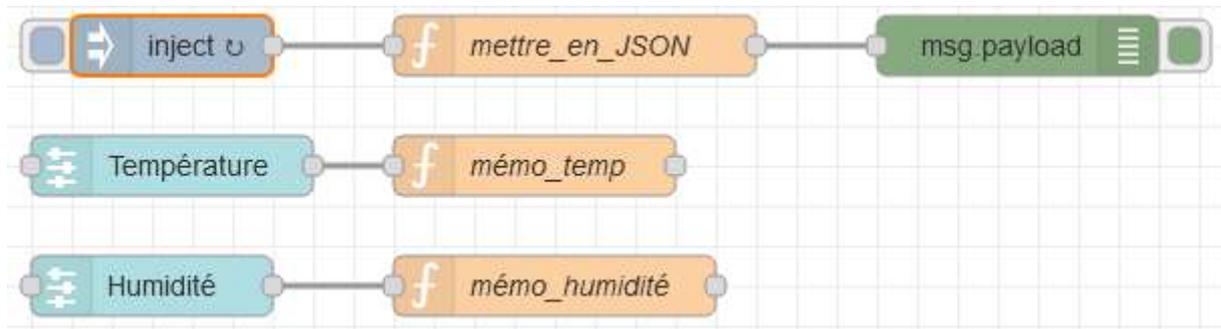
Measurement : capteur_salon

⇒ Effacer les données enregistrées au préalable avec la commande : DELETE FROM "capteur_salon"
(il faut lancer influxdb et utiliser la base de données capteur_maison)

3 – Simuler l'enregistrement de donnée provenant d'un capteur

Pour simuler l'enregistrement de donnée provenant d'un capteur , on utilise Node-RED.

⇒ Sous node-RED, créer le flux suivant (2 slider – 3 fonction – 1 inject – 1 debug)



Edit slider node

Delete Cancel Do

Properties

Group [test] Simulation des grandeurs

Size auto

Label Température

Tooltip optional tooltip

Range min 0 max 40 step 0.1

Output continuously while sliding

If msg arrives on input, pass through to output:

When changed, send:

Payload Current value

Topic msg.topic

Class Optional CSS class name(s) for widget

Edit slider node

Delete Cancel Do

Properties

Group [test] Simulation des grandeurs

Size auto

Label Humidité

Tooltip optional tooltip

Range min 0 max 100 step 1

Output continuously while sliding

If msg arrives on input, pass through to output:

When changed, send:

Payload Current value

Topic msg.topic

Class Optional CSS class name(s) for widget

Edit function node

Delete

Properties

Name mémo_temp

Setup On Start On Me

```

1 flow.set("temp",msg.payload);
2 return msg;

```

Edit function node

Delete

Properties

Name mémo_humidité

Setup On Start On Me

```

1 flow.set("humidite",msg.payload);
2 return msg;

```

Remarque : flow est une variable reconnue sur une même feuille. Alors que global est une variable reconnue sur toutes les feuilles.

la fonction flow.set va permettre d'enregistrer une valeur dans une variable.

la fonction flow.get va permettre de lire la valeur d'une variable.

The image shows two side-by-side configuration windows in Node-RED. The left window is titled 'Edit inject node' and contains a 'Delete' button, a 'Cancel' button, and a 'Properties' section. Under 'Properties', there is a 'Name' field with the value 'Name'. Below that, there are two payload configuration rows: 'msg.payload1' set to 'flow.temp' and 'msg.payload2' set to 'flow.humidite'. At the bottom, there is a '+ add' button, a checkbox for 'Inject once after 0.1 seconds, then', and a 'Repeat' section with a dropdown set to 'interval' and 'every 10 seconds'.

The right window is titled 'Edit function node' and contains a 'Delete' button, a 'Properties' section, and a 'Name' field with the value 'mettre_en_JSON'. Below the 'Name' field, there are three tabs: 'Setup', 'On Start', and 'On Message', with 'On Message' selected. The 'On Message' tab shows a JavaScript code block with the following code:

```
1 var vtemp=msg.payload1;
2 var vhumidite=msg.payload2;
3 msg.payload={"temp":vtemp,"humidite":vhumidite};
4 return msg;
```

L'instruction var (en javascript) permet de déclarer une variable.

La fonction inject, va permettre d'envoyer des données toutes les 10 secondes (avec cette programmation).

⇒ Tester en observant les informations dans la fenêtre DEBUG et en agissant sur les curseurs (ip_rasp:1880/ui) comme ci-dessous.

debug [i] [📄] [🔍] [⚙️] [🗑️]

all nodes [🗑️]

```

10/01/2022, 17:20:34 node: 9d098ea2.145c58
msg.payload : Object
  ▶ { temp: 9.3, humidite: 19 }

10/01/2022, 17:20:44 node: 9d098ea2.145c58
msg.payload : Object
  ▶ { temp: 9.3, humidite: 19 }

10/01/2022, 17:20:54 node: 9d098ea2.145c58
msg.payload : Object
  ▶ { temp: 9.3, humidite: 19 }

10/01/2022, 17:21:04 node: 9d098ea2.145c58
msg.payload : Object
  ▶ { temp: 15.4, humidite: 42 }

10/01/2022, 17:21:14 node: 9d098ea2.145c58
msg.payload : Object
  ▶ { temp: 21.1, humidite: 42 }

10/01/2022, 17:21:24 node: 9d098ea2.145c58
msg.payload : Object
  ▶ { temp: 21.1, humidite: 42 }

10/01/2022, 17:21:34 node: 9d098ea2.145c58
msg.payload : Object
  ▶ { temp: 21.1, humidite: 66 }

10/01/2022, 17:21:44 node: 9d098ea2.145c58
msg.payload : Object
  ▶ { temp: 21.1, humidite: 66 }

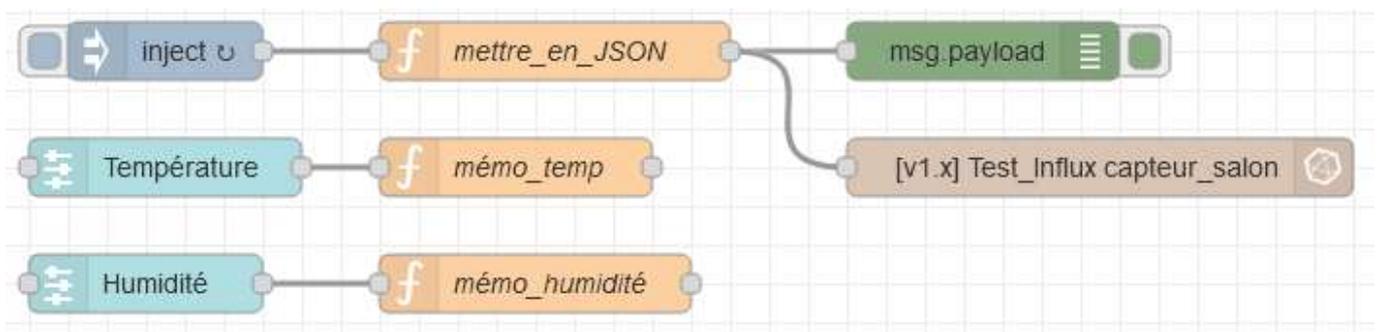
```

Simulation des grandeurs

Température

Humidité

⇒ Ajouter l'enregistrement dans la base de données capteur_maison, measurement capteur_salon, avec le node influxdb out comme ci-dessous :

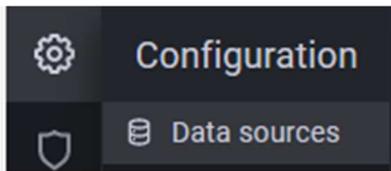


⇒ Vérifier l'enregistrement des données dans la base de données comme ci-dessous (commande CLI).

```
pi@raspberrypi:~ $ influx -precision rfc3339
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> USE capteur_maison
Using database capteur_maison
> SELECT * FROM capteur_salon
name: capteur_salon
time                humidite temp
-----
2022-01-10T16:37:09.862236595Z 66      21.1
2022-01-10T16:37:19.864059407Z 66      21.1
2022-01-10T16:37:29.867022853Z 66      21.1
2022-01-10T16:37:39.871111412Z 66      21.1
2022-01-10T16:37:49.878317102Z 66      21.1
2022-01-10T16:37:59.881677789Z 66      21.1
2022-01-10T16:38:09.883470072Z 66      21.1
2022-01-10T16:38:19.891151235Z 66      21.1
```

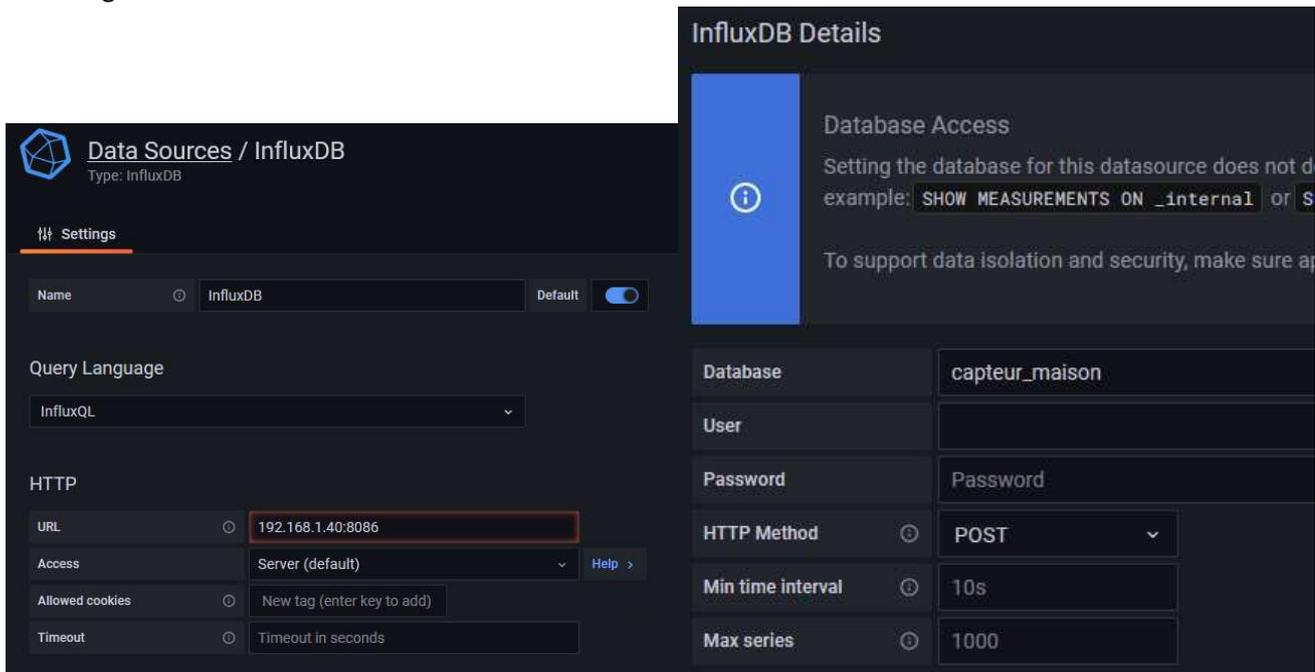
4 – Affichage avec grafana

⇒ Configurer la source (base de donnée)



⇒ Cliquer sur Add data source et Choisir InfluxDB

⇒ Configurer la source



The screenshot shows the Grafana 'Data Sources / InfluxDB' configuration page. The 'Settings' section includes:

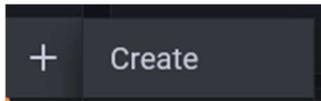
- Name: influxDB (Default:)
- Query Language: InfluxQL
- HTTP URL: 192.168.1.40:8086
- Access: Server (default)
- Allowed cookies: New tag (enter key to add)
- Timeout: Timeout in seconds

The 'InfluxDB Details' section on the right includes:

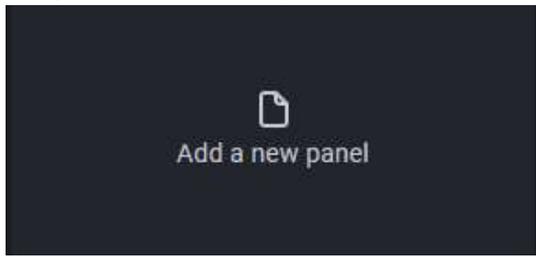
- Database: capteur_maison
- User: (empty)
- Password: Password
- HTTP Method: POST
- Min time interval: 10s
- Max series: 1000

⇒ Cliquer sur Save & test (influxDB doit être lancée au préalable).

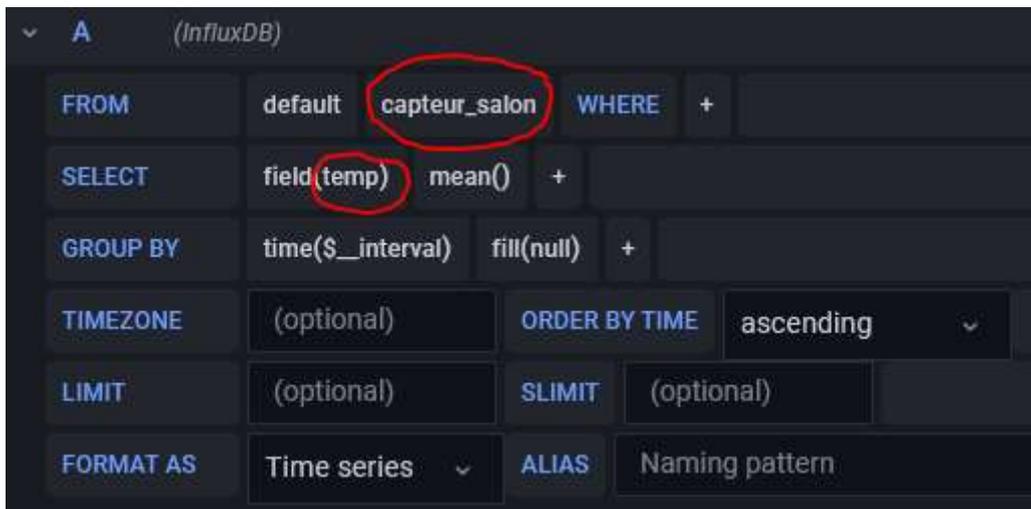
⇒ Créer un nouveau tableau de bord (New Dashboard).



⇒ Ajouter un nouveau panneau

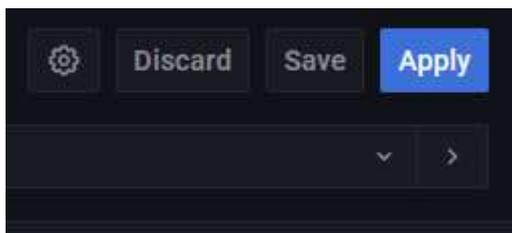


⇒ Configurer la mesure à afficher (capteur_salon et temp)



⇒ Modifier le titre (entrer Température)

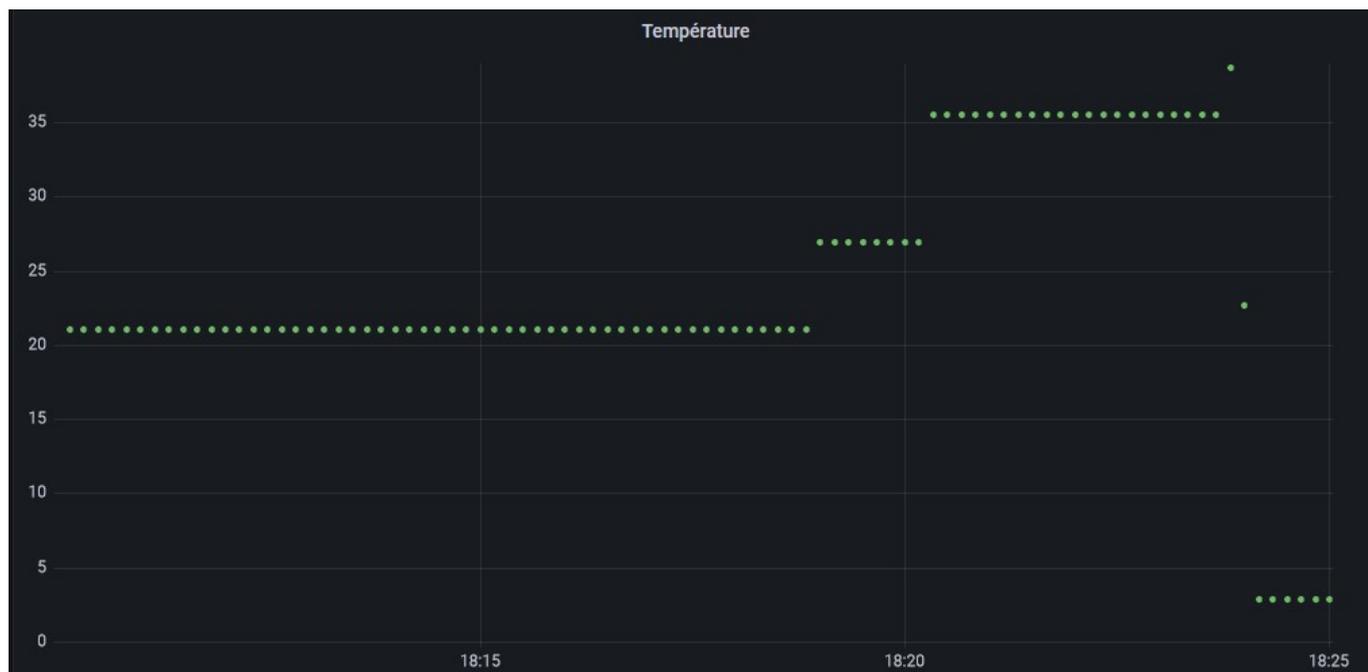
⇒ Cliquer sur Apply



⇒ Afficher les 15 dernières minutes et un rafraîchissement de 10s



⇒ Agir sur le curseur de température et visualiser les variations sur le tracé. (possibilité de définir les valeurs max. et min. du graphe)



⇒ De la même manière, ajouter un panneau pour l'humidité avec un affichage sous forme d'une jauge avec une échelle de 0 à 100%.

⇒ Tester en agissant sur le curseur humidité

