

# TP sur le stockage, l'analyse et la visualisation des données (3h)

---

## Objectif :

- Traitement sous NodeRED
  - Stockage sous Timescale
  - Analyse, visualisation des données sur Grafana
- 

<b>1. Node-RED</b>	<b>2</b>
a. Introduction	2
b. Souscription MQTT	2
c. Récupération des données	2
d. Décodage base64	3
e. Décodage des données	3
f. Création d'un dashboard Node-RED	3
<b>2. Timescale</b>	<b>4</b>
a. Introduction	4
b. Initialisation	5
c. Stockage des données reçues	5
<b>3. Grafana</b>	<b>5</b>
a. Introduction	5
b. Configuration de la source de données	5
c. Création d'un dashboard	6

---

## 1. Node-RED

### a. Introduction

Node-RED est un logiciel open-source de programmation "low-code" développé en Javascript et basé sur NodeJS . Son avantage principal est sa facilité de prise en main sans connaissance poussée en programmation. En effet, la programmation s'effectue à l'aide d'un éditeur graphique par flot composé de blocs fonctionnels appelés nœuds.

### b. Souscription MQTT

- Ouvrir l'interface web : <http://localhost:8081>
- Ajouter un nœud 
- Ouvrir le nœud
- Configurer le broker
- S'abonner à tous les topics
- Ajouter un nœud 
- Relier le nœud le **mqtt in** au nœud **debug**
- Déployer le flot avec le bouton **deploy**
- Observer les données publiées dans l'onglet debug



### c. Récupération des données

- Modifier l'abonnement MQTT pour recevoir uniquement les données de votre capteur
- Copier dans le presse-papier le JSON reçu depuis l'onglet de debug



- Ajouter un nœud 
- Ouvrir le nœud
- Coller le contenu du presse-papier dans le champ **payload**
- Identifier dans le JSON le champ contenant les données transmises

- Ajouter un nœud 
- Configurer la payload sortant pour qu'elle contienne uniquement les données transmises
- Ajouter un nœud de debug
- Relier les nœuds **inject**, **change** et **debug**
- Injecter des trames et vérifier le résultat dans l'onglet de debug

#### d. Décodage base64

Pour décoder le base64 une palette supplémentaire a été installée.

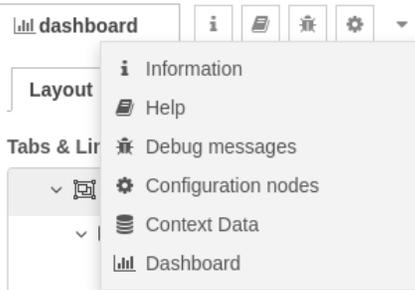
- Cliquer sur le bouton menu en haut à droite (3 lignes horizontales), puis sur **Manage palette**
- Prendre connaissance des palettes installées
  - Les nouvelles **palettes** s'installent à partir de l'onglet **Install**
- Cliquer sur **Close**
- Ajouter un nœud 
- Relier ce dernier entre le nœud **change** et **debug** afin d'afficher le base64 décodé

#### e. Décodage des données

- Ajouter un nœud 
- Ouvrir le nœud
- Ajouter le code en JavaScript réalisant le décodage des données binaires
  - Utiliser l'API buffer : <https://nodejs.org/api/buffer.html>

#### f. Création d'un dashboard Node-RED

Node-RED permet de réaliser simplement des IHM Web. Celles-ci interagissent avec les flots déployés. Une limitation importante à noter : il n'y a pas de session utilisateur. L'utilisation de ces interfaces est donc réservée à un cadre mono-utilisateur.

- Ouvrir l'onglet **dashboard** 

- Ajouter un onglet **M5LoRaKit**

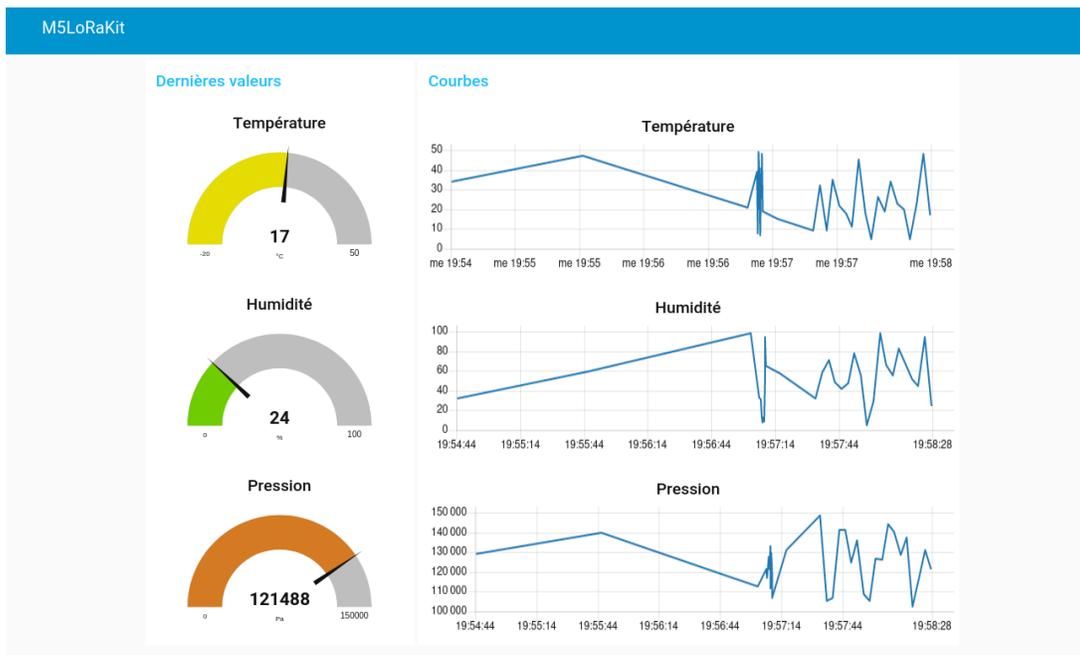
Tabs & Links



- Ajouter un groupe **Dernières valeurs**



- Ajouter un nœud  gauge
- Déployer
- Consulter votre UI contenant la jauge : <http://localhost:8081/ui>
- Afficher la température dans cette jauge
- Compléter votre flot afin d'obtenir le tableau de bord suivant



## 2. Timescale

### a. Introduction

TimescaleDB est une base de données relationnelle open-source pour les séries chronologiques et les analyses. En réalité Timescale est une extension de PostgreSQL pour gérer les séries temporelles.

---

## b. Initialisation

- Importer le flot Node-RED **timescale\_basis\_flow.json** fourni
- Prendre connaissance des requêtes SQL dans les nœuds 
- Créer la table
  - Identifier les différentes colonnes constituant la table
- Insérer des données aléatoires
- Consulter les données stockées
- Supprimer la table
- Recréer la table

## c. Stockage des données reçues

- Insérer en base les données reçues de votre objet
  - S'appuyer sur le nœud **Insert value** donné précédemment
- Consulter les données stockées

## 3. Grafana

### a. Introduction

Grafana est une plateforme open source de visualisation de séries temporelles. Celle-ci gère la création de tableaux de bord plus ou moins complexes. Elle permet de les partager aisément. La plateforme propose aussi de l'alerting.

### b. Configuration de la source de données

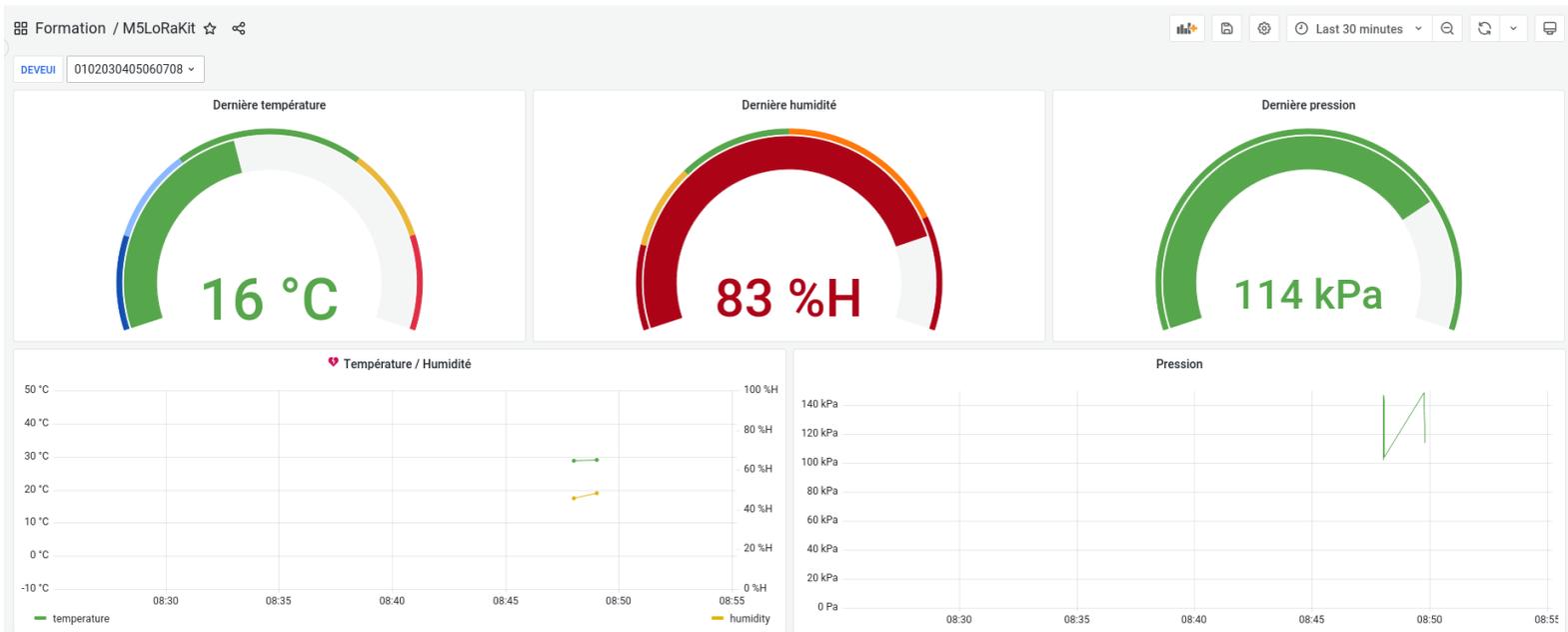
- Ouvrir l'interface web : <http://localhost:8082>
- S'Identifier avec l'utilisateur **admin** et le mot de passe **admin**
- Ouvrir la page de configuration des **Data sources** à partir de l'icône de configuration en bas de la barre de menu latérale gauche
- Cliquer sur **Add Data source**
- Sélectionner **PostgreSQL**
- Configurer la connexion
  - Hôte : timescaledb
  - Base : postgres

- 
- Utilisateur : timescale
  - Mot de passe : password
  - Sans TLS
  - Version Postgres : 14
  - Valider la configuration avec le bouton **Save & test**

### c. Création d'un dashboard

- Dans l'onglet **Dashboards**, créer un dossier **Formation**
- Dans ce dossier, créer un nouveau **dashboard M5LoRaKit**
- Ajouter un premier **panel Température** affichant une courbe de la température.  
Configurer :
  - L'unité
  - Le min et le max
- Ajouter un **panel Dernière Température** de type **Gauge** affichant la dernière température de la plage temporelle sélectionnée en configurant
  - L'unité
  - Le min et le max
- Sur le **panel Température** ajouter une courbe de l'humidité. Le renommer **Température / Humidité**. Configurer :
  - L'unité
  - Le min et le max
  - L'axe à droite
- Configurer une agrégation temporelle de type moyenne
- Prendre connaissance de la requête SQL générée par le **Query builder**
- Ouvrir les paramètres du **dashboard**
- Dans l'onglet **Variables** créer une nouvelle variable DEVEUI
  - Écrire une requête SQL listant l'ensemble des DEVEUI de la table
- Utiliser cette variable dans vos panneaux pour filtrer les données
- Sur le **panel Température / Humidité** ajouter une alerte si la température d'un objet dépasse les 35°C

- Modifier votre **dashboard** pour obtenir l'affichage suivant



- Partager un **snapshot** du **dashboard** sur <https://snapshots.raintank.io>