# Partie Python :

## 1. Installation du Raspberry Pi :

* **Installer** l'OS sur le Raspberry Pi.

## 2. Configuration du réseau :

* **Connecter** le Raspberry Pi au réseau Wi-Fi (SSID : electroProjectWifi - mdp : B1MesureEnv).

## 3. Activation des services à distance :

* **Activer** le service **VNC** pour l'accès à distance avec interface graphique.
* **Activer** SSH pour l'accès à distance en ligne de commande.

## 4. Création d'un script Python :

* **Créer** un répertoire dans `/home/pi/` (ou tout autre chemin de votre choix).
* Dans ce répertoire, **créer** un script Python qui sera exécuté lors du démarrage du Raspberry Pi.

## 5. Création d'un service système :

* **Créer** un service Linux pour que le script Python soit automatiquement lancé au démarrage du Raspberry Pi.

## 6. Installation de MQTT :

* **Installer** un serveur MQTT (Mosquitto) sur le Raspberry Pi.
* **Configurer** un utilisateur et un mot de passe (par exemple, `pi` et `raspberry`).
* **Tester** l’accès à MQTT à partir de votre PC (Python, MQTT explorer ou équivalent)

Installation :

sudo apt update && sudo apt upgrade

*s*udo apt install -y mosquitto mosquitto-clients

sudo systemctl enable mosquitto.service

Testing :

mosquitto -v

Reference : <https://randomnerdtutorials.com/how-to-install-mosquitto-broker-on-raspberry-pi/>

## 7. Fonctionnalités du script Python :

* **Logger** : Initialiser un logger qui génère des fichiers de logs circulaires dans le répertoire du script.
* **MQTT Client** : Initialiser un client MQTT qui s'abonne à un topic spécifique.
* **Attente d'événements** : Le script doit surveiller le topic MQTT et réagir à la réception d'un message.
* **Gestion des logs** : Tous les événements doivent être loggés avec des niveaux différents (debug, info, warning, error, exception, etc.).
* **Gestion des erreurs** : Le script doit traiter toutes les erreurs et exceptions de manière robuste.
* **InfluxDb** : Le script enregistre les données reçues avec MQTTdans InfluxDb

8. Installation InfluxDb :

* **Installer et Configurer** influxdb : Voir slides.

Reference : <https://docs.influxdata.com/influxdb/v2/install/?t=Linux#manually-download-and-install-the-influxd-binary>

RASPBERRY PI 4 : ARM64

**Step 1: Download and Add the InfluxData Repository Key**

# Download the InfluxData GPG key

curl --silent --location -O https://repos.influxdata.com/influxdata-archive.key

# Verify the key's SHA256 checksum

echo "943666881a1b8d9b849b74caebf02d3465d6beb716510d86a39f6c8e8dac7515 influxdata-archive.key" | sha256sum --check -

# If the above command outputs "influxdata-archive.key: OK", proceed to import the key

sudo mkdir -p /etc/apt/trusted.gpg.d/

sudo cat influxdata-archive.key | gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/influxdata-archive.gpg > /dev/null

**Step 2: Add the InfluxData Repository**

echo 'deb [signed-by=/etc/apt/trusted.gpg.d/influxdata-archive.gpg] https://repos.influxdata.com/debian stable main' | sudo tee /etc/apt/sources.list.d/influxdata.list

**Step 3: Update and Install InfluxDB**

# Update your package lists

sudo apt-get update

# Install InfluxDB

sudo apt-get install influxdb2

**Step 4: Start and Enable InfluxDB**

sudo systemctl enable influxdb

sudo systemctl start influxdb

**Verify the Installation**

sudo systemctl status influxdb

curl <http://localhost:8086/ping>

If InfluxDB was installed as a systemd service (which is the case here), systemd manages the influxd daemon and no further action is required.

Access the UI through : <http://localhost:8086>

**Setting up INFLUXDB:**

Reference: <https://docs.influxdata.com/influxdb/v2/get-started/setup/>

**To install the CLI for influx:**

Reference : <https://docs.influxdata.com/influxdb/v2/tools/influx-cli/?t=Linux>

**Option 1: Create a Virtual Environment (Recommended)**

A virtual environment isolates your Python dependencies, preventing any issues with the system Python installation.

1. **Install the python3-venv package** (if not installed):

sudo apt install python3-venv

1. **Create a virtual environment**:

python3 -m venv myenv

1. **Activate the virtual environment**:

source myenv/bin/activate

1. **Install the influxdb-client inside the virtual environment**:

pip install influxdb-client

1. When you're done, **deactivate the virtual environment**:

Deactivate

NOTE : To run any file using influx module you need to activate virtual environment in home directory. This is where the influxdb module was installed.

To run any files from other directories, run from virtual environment from home then put full path to file to execute.

**Writing data with python**

Reference : <https://docs.influxdata.com/influxdb/v2/api-guide/client-libraries/python/>

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

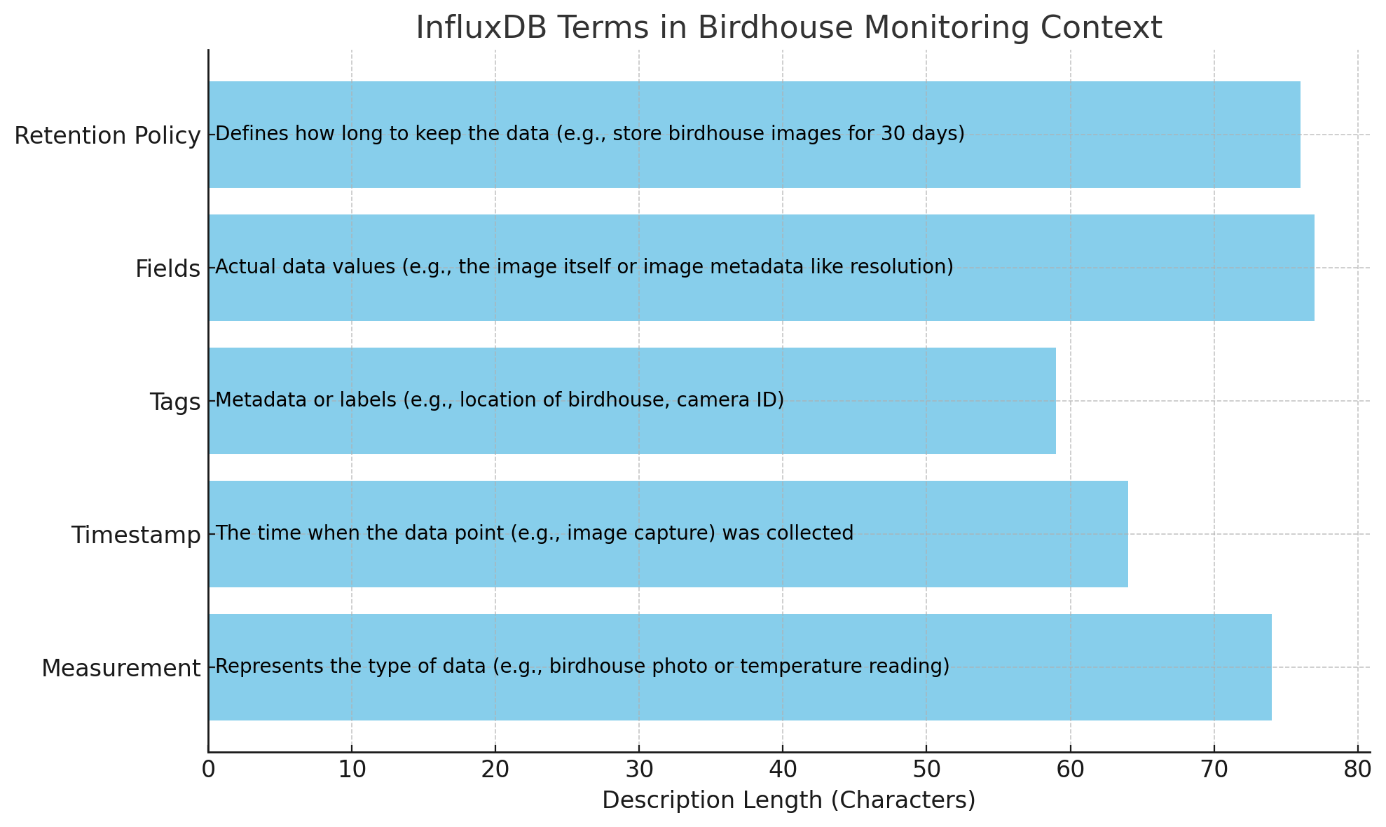
**Query**

influx config create --config-name my\_config --url http://localhost:8086 --org hepl --token KcTW\_C6srhM8pgOxOmtMKZXc72oIum7-JXYrkcxhJ3YJFbBhwaH265MwSKQ57SzjbJ-7Aym9Q9tcuIS 9tcuISV5iTQ1w==<9tcuISV5iTQ1w== --active

**Reading data with nodejs**

Reference: <https://docs.influxdata.com/influxdb/v2/api-guide/client-libraries/nodejs/>

* + User : pi
  + Password : raspberry
  + Organisation : hepl
  + Bucket : nichoir



## 9. Prochaines étapes :

* Le reste des spécifications sera défini ultérieurement.

## 10. WebPAGE

1. Install nodejs
2. Initialize node in webpage folder
3. Install dependencies (influxdb dependencies)
4. Download bootstrap css file

## 11. Other useful command

PSCP :

C:\Users\katee>pscp kate@katePi4:/home/kate/api-token.txt "C:\Users\katee\Desktop\KateGeneral\Master\_Passerelle\Projet\_SmartCities\birdHouse\_monitoring"