Note
/20

Compétences	Acquise	Non Acquise
C04 : Analyser un système informatique		
C05 : Concevoir un système informatique		

TP n°5

Stockage de Séries Temporelles : Ingestion par Telegraf et Base de Données InfluxDB v2

FICHE MISSION 2

I – PROBLEMATIQUE : Comment structurer, convertir et stocker un flux continu de métriques IIoT volatiles (JSON MQTT) dans une base de données de séries temporelles (TSDB) ?

II – OBJECTIFS :

- Comprendre le fonctionnement d'une base de données orientée séries temporelles (InfluxDB v2)
- Déployer et initialiser InfluxDB v2 sur la VM Ubuntu (**192.168.2.100**)
- Configurer le middleware Telegraf (plugin **inputs.mqtt_consumer** et parser **json_v2**)
- Valider l'ingestion des séries temporelles et exécuter des requêtes d'extraction en langage Flux

III – PRE-REQUIS :

- Avoir validé le TP4 IT (Broker **Mosquitto** opérationnel et publication **MQTT** active sur **192.168.2.100:1883**)
- Maîtrise de l'édition de fichiers de configuration sous Linux (**nano**, **vim**)
- Compréhension de la différence entre métadonnées textuelles (**Tags**) et valeurs numériques (**Fields**)

IV – MISE EN SITUATION : Lors du TP4 IT, vous avez validé la transmission des métriques de distance et de réflectivité du capteur **OGD592** sous forme de messages JSON vers le broker Mosquitto. Cependant, ces données restent volatiles : elles disparaissent dès leur affichage en console.

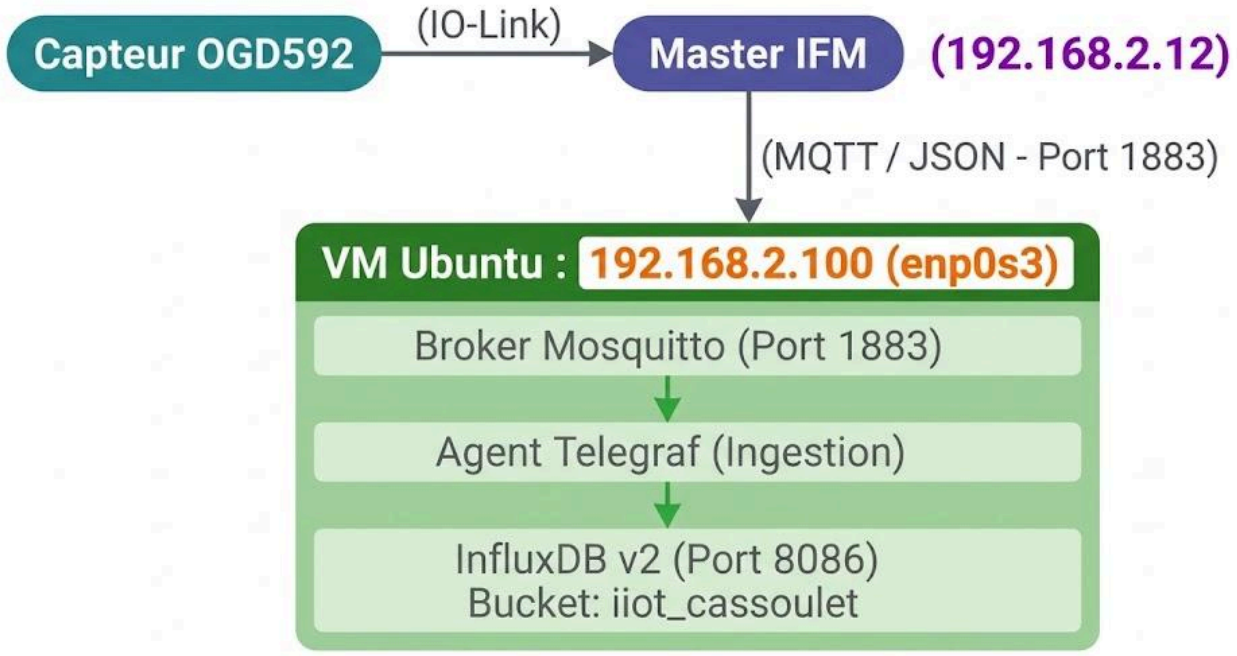
Dans cette mission, vous allez mettre en place la persistance des données sur la VM de supervision usine (**192.168.2.100**). Vous allez installer la base **InfluxDB v2** puis configurer l'agent middleware **Telegraf**. Ce dernier s'abonnera aux topics **MQTT** sur l'interface enp0s3, découpera le **JSON** IFM pour séparer les valeurs de mesure (**Fields**) des métadonnées (**Tags**) et enregistrera l'historique dans le bucket **iiot_cassoulet**.

V – <u>RESSOURCES MATERIELLES</u> : Vous réaliserez cette mission en travaillant avec un **PC** connecté à **Internet** et un accès à un environnement de travail de type **Workspace** avec les outils bureautiques :

- Traitement de texte.
- Tableur.
- Logiciel de diaporama.
- PC Étudiant (**192.168.2.20**) connecté au réseau IT
- Serveur sur une Machine Virtuelle **Ubuntu 26.04 LTS** (Réseau Interne **192.168.2.100** et Accès par Pont Carte Réseau Sans-Fil)
- Master **IO-Link** IFM (AL1304 / AL1306) avec double interface : IT (**192.168.2.12**) et OT (**192.168.1.12**)
- Capteur photoélectrique **IO-Link** OGD592
- Automate PLC Siemens S7-1200 : OT (**192.168.1.10**)
- Logiciel Moneo Configure
- Clients **MQTT** (mosquitto-clients, MQTT Explorer)
- Logiciels installés sur la VM : Broker **Mosquitto**, **InfluxDB v2**, **Agent Telegraf**

VI – <u>RESSOURCES PEDAGOGIQUES</u> : Vous utiliserez le dossier technique fourni ainsi que la documentation technique IFM du Master **IO-Link** et du capteur OGD592.

VII – <u>SCHEMA DU RESEAU</u> :



VIII – ETUDE PREALABLE DE LA PERSISTANCE DES DONNEES :

Dans cette nouvelle mission, nous allons nous intéresser à la persistance des données.

Quelle est la différence majeure entre une base de données relationnelle (**SQL**) et une base de données Time-Series (**InfluxDB**) ? Pourquoi InfluxDB est-il privilégié en **IIoT** ?

……
……

Quel est le rôle de l'agent middleware **Telegraf** ? Pourquoi est-il préférable d'utiliser cet agent intermédiaire plutôt que de faire écrire directement le Master **IO-Link** dans **InfluxDB** ?

……
……

Expliquer la notion de Parsing **JSON** et la distinction entre **Fields** (champs) et **Tags** (étiquettes) au sein d'**InfluxDB**.

……
……

IX – INSTALLATION D'INFLUXDB V2 :

Nous allons commencer par installer **InfluxDB v2** sur notre machine virtuelle, taper les commandes suivantes pour l'installer :

```
wget https://dl.influxdata.com/influxdb/releases/influxdb2_2.7.5-1_amd64.deb
sudo dpkg -i influxdb2_2.7.5-1_amd64.deb
```

Activer **InfluxDB v2** et vérifier que le service est correctement lancé en tapant les commandes suivantes :

```
sudo systemctl daemon-reload
sudo systemctl enable --now influxdb
sudo systemctl status influxdb
```

Donner la capture d'écran du bon fonctionnement d'**InfluxDB v2**.

……
……

Vous pouvez accéder maintenant à l'interface de configuration d'**InfluxDB v2** en utilisant un navigateur Internet et en tapant l'**URL** suivante : **http://192.168.2.100:8086**

Remplir les informations suivantes et relever le Token généré :

- Organization : **Lycee_Jules_Fil**
- Bucket : **iiot_cassoulet**

Donner les captures d'écran de la configuration d'**InfluxDB v2** et relever le **Token** généré.

.....
.....

X – INSTALLATION DE TELEGRAF :

Nous allons maintenant installer **Telegraf**, utiliser la procédure suivante pour l'installer :

Procédure d'installation de Telegraf sur Ubuntu Server 26.04

1. Importer la clé GPG InfluxData dans le trousseau système :

Importer la clé publique officielle **DA61C26A0585BD3B** directement dans le trousseau système d'**APT** :

```
sudo gpg --keyserver keyserver.ubuntu.com --recv-keys DA61C26A0585BD3B
sudo gpg --export DA61C26A0585BD3B | sudo tee /etc/apt/trusted.gpg.d/influxdata.gpg >
/dev/null
```

2. Déclarer le dépôt APT InfluxData :

Configurer l'adresse du dépôt officiel dans le répertoire des sources d'**APT** :

```
echo 'deb [https://repos.influxdata.com/debian](https://repos.influxdata.com/debian) stable
main' | sudo tee /etc/apt/sources.list.d/influxdata.list
```

3. Mettre à jour la liste des paquets et installer Telegraf :

Recharger les dépôts et lancer l'installation :

```
sudo apt update && sudo apt install telegraf -y
```

Note : Si l'écran bleu de configuration d'APT apparaît, sélectionner l'option **« Garder la version actuellement installée »** et valider avec **<Ok>**.

4. Démarrer et vérifier le service Telegraf :

Activer le démarrage automatique au boot de la VM et lance le service immédiatement :

```
sudo systemctl enable --now telegraf
```

Pour vérifier le bon fonctionnement du service :

```
sudo systemctl status telegraf
```

(Le statut doit indiquer **active (running)**).

5. Renommer le fichier de configuration en sauvegarde :

```
sudo mv /etc/telegraf/telegraf.conf /etc/telegraf/telegraf.conf.bak
```

Nous allons configurer **Telegraf** avec la structure réelle du JSON IFM dans le fichier.

➔ Créer le fichier et complétez-le en remplaçant le Token par celui généré lors de l'installation d'**InfluxDB v2** : **/etc/telegraf/telegraf.conf**

```
[[outputs.influxdb_v2]]
 urls = ["http://127.0.0.1:8086"]
 token =
"mW7olkeyOEW0mpxnb9vLgxUGbu_t6MhXAthscw4j2lMFT_6pAsviGwk3pALcyM8pM2MxsONerRVmdLg6Z0ceIQ=="
 organization = "Lycee_Jules_Fil"
 bucket = "iiot_cassoulet"

[[inputs.mqtt_consumer]]
 servers = ["tcp://192.168.2.100:1883"]
 topics = [
   "usine/castelnaudary/ligne1/#"
 ]
 client_id = "telegraf_iiot_client"
 data_format = "json_v2"

 [[inputs.mqtt_consumer.json_v2]]
   [[inputs.mqtt_consumer.json_v2.field]]
     path = "data.payload.*pdin.data"
     rename = "pdin_hex"
     type = "string"

   [[inputs.mqtt_consumer.json_v2.field]]
     path = "data.payload.*counter.data"
     rename = "compteur_event"
     type = "int"
```

Quelle est la commande pour relancer le service **Telegraf** ? Donner la capture d'écran pour vérifier qu'il a bien redémarré et qu'aucune erreur de configuration n'apparaît.

……
……

XI – <u>VERIFICATION ET TRAITEMENT DES DONNEES AVEC FLUX</u> :

Nous allons maintenant vérifier que **Telegraf** récupère bien les données et les envoie à **InfluxDB**.

1. <u>Vérification des Logs système</u> :

Pour vérifier que **Telegraf** fonctionne convenablement, taper la commande suivante :

```
sudo journalctl -u telegraf -f -n 20
```

Donner une capture d'écran des Logs de **Telegraf** et analyser ces Logs.

……
……

2. <u>Visualisation directe des données dans l'interface Web d'InfluxDB v2</u> :

Il est maintenant possible de visualiser les données enregistrées dans **InfluxDB v2**. Pour cela, vous allez devoir choisir votre Bucket puis la sélection de la mesure. Le mieux est d'utiliser une requête de type script, taper cette requête dans Script Editor :

```
from(bucket: "iiot_cassoulet")
 |> range(start: -15m)
 |> filter(fn: (r) => r["_measurement"] == "mqtt_consumer")
 |> filter(fn: (r) => r["_field"] == "pdin_hex")
```

Expliquer la procédure permettant de contrôler et de visualiser les données ingérées directement depuis l'interface Web d'**InfluxDB v2** sans passer par un outil tiers. Donner une capture d'écran des données enregistrées dans **InfluxDB v2** sous forme de **Table**.

……
……

Expliquer pourquoi les données peuvent s'afficher dans une **Table** mais pas dans un **Graph**.

……
……

3. Simulation d'encrassement de la lentille et maintenance prédictive :

Nous avons simulé un encrassement de l'optique du capteur IFM OGD592 par la pose d'un ruban adhésif translucide (scotch) sur la fenêtre optique. Ce dispositif atténue le faisceau laser émis et réduit l'intensité du signal lumineux retourné au récepteur, sans couper intégralement la mesure.

Nous relevons ces données dans la table **InfluxDB** :

2026-09-20 17:23:54...	2026-09-20 17:38:54...	2026-09-20 17:38:35...	0021FD00004A0001	pdin_hex	mqtt_consumer	ubuntu	usine/castelnaudary/l...
2026-09-20 17:23:54...	2026-09-20 17:38:54...	2026-09-20 17:38:36...	0066FD0000050001	pdin_hex	mqtt_consumer	ubuntu	usine/castelnaudary/l...

Expliquer comment ces données mettent en évidence l'encrassement du capteur et conclure sur l'intérêt d'avoir une maintenance prédictive avec ce type de données.

……
……

Grille de Notation			Validation Compétences
Etude préalable de la persistance des données		**/4**	
Installation et Configuration d'**InfluxDB v2**		**/4**	
Installation et Déploiement de **Telegraf**		**/3**	**Valide C05**
Ingestion **MQTT**, Parsing **JSON** et Requêtes **Flux**		**/3**	
Visualisation **Table**/**Graph** et Maintenance Prédictive		**/6**	**Valide C04**
Barème de notation total		**/20**	