

Influxdb - Telegraf - Grafana

lundi 18 août 2025 12:02

Influxdb :

Influxdb est une base de donnée (de type "Objets") composée de "Bucket". Ce sont comme des boîtes dans lesquelles on va vouloir y stocker des données.

Pour pouvoir écrire dans ces Buckets, on a besoin d'un "Token", qui est comme une clé, qui permet d'obtenir l'autorisation de lire et écrire dans un Bucket (chaque Token peut être configuré avec plus ou moins de priviléges).

Telegraf :

Telegraf est un programme configurable, qui permet de lire des données, et peut également écrire ces mêmes données dans les Buckets de Influxdb.

Grafana :

Grafana lui, permet d'interpréter les données écrites dans les Buckets de Influxdb et de nous afficher ces mêmes données dans ce que l'on souhaite comme par exemple des tableaux, des graphiques, etc...

Pour travailler avec Grafana ici, on va devoir utiliser le langage FLUX avec des "Query", en lien avec telegraf et influxdb.

Partie Influxb :

/!\ On va travailler avec la version 2 /!\

Dans un serveur Linux (CentOS) :

```
curl --silent --location -O https://repos.influxdata.com/influxdata-archive.key
gpg --show-keys --with-fingerprint --with-colons ./influxdata-archive.key 2>&1 |
| grep -q '^fpr:|+24C975CBA61A024EE1B631787C3D57159FC2F927:$' |
&& cat influxdata-archive.key |
| gpg --dearmor |
| tee /etc/pki/rpm-gpg/RPM-GPG-KEY-influxdata > /dev/null
```

```
cat <<EOF | tee /etc/yum.repos.d/influxdata.repo
[influxdata]
name = InfluxData Repository - Stable
baseurl = https://repos.influxdata.com/stable/x86_64/main
enabled = 1
gpgcheck = 1
gpgkey = https://repos.influxdata.com/influxdata-archive.key
EOF
```

```
yum install influxdb2
```

```
systemctl start influxdb
```

```
systemctl status influxdb
```

Influxdb2 est installé ! ✓

Ce service tourne normalement sur le **port 8086**

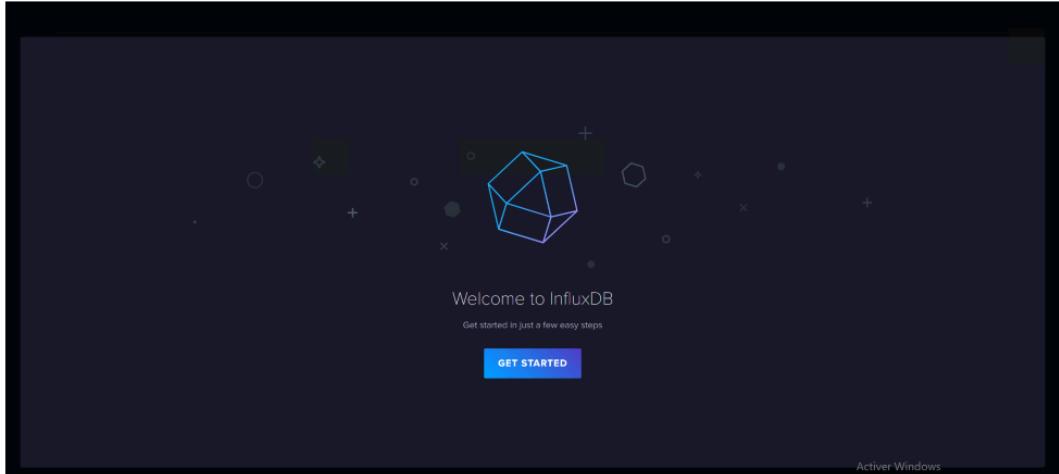
On s'y connecte via un navigateur : http://ip_de_la_machine:8086

/!\ Il se peut que influxdb nous mette une balayette et nous empêche de nous connecter car il écoute en ipv6. /!\\

Pour résoudre ce problème, il suffit de désactiver l'ipv6 sur la machine linux et c'est censé fonctionner 😊

Pour désactiver l'ipv6 : <https://www.linuxtricks.fr/wiki/desactiver-ipv6-sur-linux>

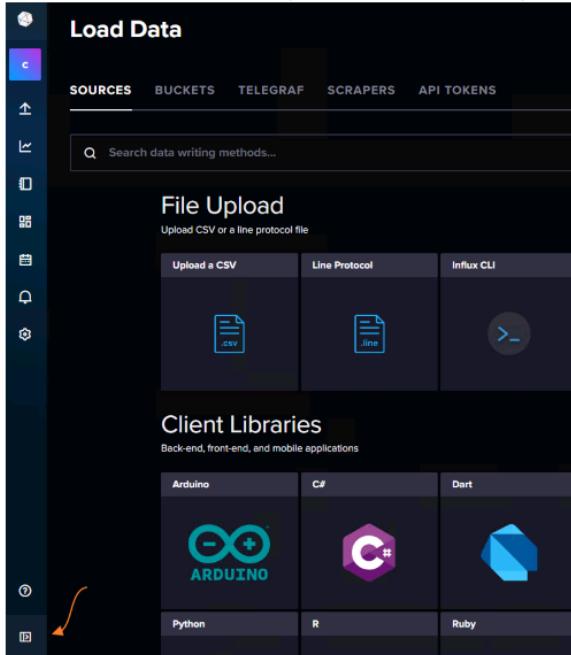
Après, on arrive normalement là-dessus :



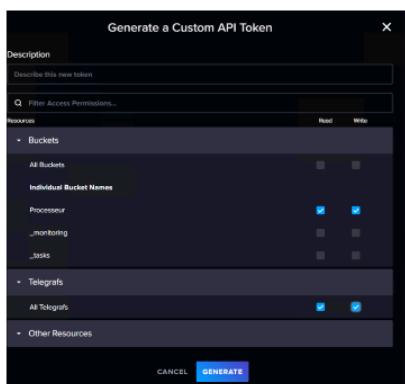
Il suffit de suivre les étapes tout en notant bien les informations (Mdp, Token, etc...) dans un KeePass par exemple.

Pour créer des Buckets, voici les étapes à suivre :

- Dans le menu de droite (à faire dérouler en cliquant sur ce bouton)



- Aller dans : "Load Data" → "Buckets"
- Faites "+ CREATE BUCKET" → Nommer le Bucket → Delete Data (older than 1 year)
- Se déplacer ensuite dans "API TOKENS" → "+ GENERATE API TOKEN" → "Custom API TOKEN"
Cocher les droits souhaités pour le Bucket qu'on vient de créer, Telegraf cocher toutes les cases et "GENERATE"
/!\ Bien noter le Token généré car il ne sera pas possible de le consulté après /!



Ne pas hésiter à renommer le Token tout fraîchement créé.

On a fini ! ✓

Partie Telegraf :

```
yum install telegraf -y
```

Arborescence pour Telegraf :

/etc/telegraf → Répertoire principal de Telegraf

/etc/telegraf/telegraf.conf → Ceci est le **fichier de configuration principal** de Telegraf (Globalement ne pas y toucher)

/etc/telegraf/telegraf.d → Ceci est le **dossier dans lequel on va vouloir y créer nos fichiers de configuration** (à savoir : 1 fichier de conf pour 1 bucket)

Depuis ce répertoire, on peut y faire nos configurations comme on le souhaite.

Exemple de fichier de configuration Telegraf pour un Bucket (ici des métriques CPU que j'ai appelé "processeur.conf") :

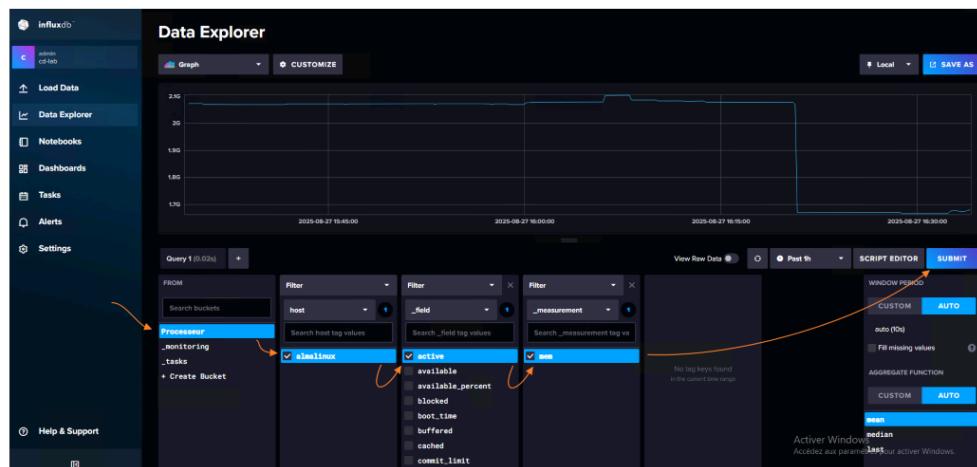
```
GNU nano 3.6.1
# Lis les métriques d'usage du cpu
[[inputs.cpu]]
## Faut-il signaler ou non les statistiques par processeur ?
percpu = true
## Faut-il signaler les statistiques totales du processeur du système ou non ?
totalcpu = true
## Si vrai, collectez les mesures de temps CPU brutes
collect_cpu_time = false
## Si vrai, calcule et signale la somme de tous les états non inactifs du processeur
## NOTE: Le champ « time_active » résultant INCLUT « iowait » !
report_active = false
## Si c'est vrai et que les informations sont disponibles, ajoutez les balises core_id et physical_id
core_tags = false

# Envoyer les données vers (grosses mode)
[[outputs.influxdb_v2]]
## Indique vers où envoyer les données
urls = ["http://127.0.0.1:8086"]
## Token du Bucket visé
token = "Y62_CQFOXLK7KBlHf6RlfVRu0P228C446CwXQiR8HQ3tx6gYY6ok04A89r_Yep1rwZ04d8LwaSmx1VeuGrA=="
## Nom de l'organisation renseignée
organization = "cd-lab"
## Nom du Bucket visé
bucket = "Processeur"
```

Vérifier que ce fichier a bien les droits suivants : -rw-r--r--

Puis dans l'interface graphique de Influxdb, aller dans : "**Data Explorer**"

Ici sélectionner son bucket, puis toujours filtrer le premier bloc de filtre à partir de "host", puis filtrer comme on le souhaite.



Si on obtient quelque chose, alors c'est dans la poche ! ✓

Voici 2 liens très utiles pour cette partie (possibilité de se débrouiller uniquement avec ces liens) :

Documentation Grafana : <https://grafana.com/docs/grafana/latest/setup-grafana/installation/>

Documentation Query Flux : <https://docs.influxdata.com/influxdb/v2/query-data/>

Sinon, je vais vous aider :)

Installation de Grafana :

- 1 Import the GPG key:

```
bash Copy
wget -q -O gpg.key https://rpm.grafana.com/gpg.key
sudo rpm --import gpg.key
```

- 2 Create `/etc/yum.repos.d/grafana.repo` with the following content:

```
bash Copy
[grafana]
name=grafana
baseurl=https://rpm.grafana.com
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://rpm.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt
```

- 3 To install Grafana OSS, run the following command:

```
bash Copy
sudo dnf install grafana
```

Installation done ✓

- 1 To start the service, run the following commands:

```
bash Copy
sudo systemctl daemon-reload
sudo systemctl start grafana-server
```

- 2 To verify that the service is running, run the following command:

```
bash Copy
sudo systemctl status grafana-server
```

Configure the Grafana server to start at boot using systemd

To configure the Grafana server to start at boot, run the following command:

```
bash Copy
sudo systemctl enable grafana-server.service
```

Démarrage de Grafana done ✓

Démarrage de Grafana done ✓

A noter que Grafana tourne par défaut sur le **port 3000**

Si on souhaite donc se connecter sur Grafana, il suffit d'aller sur un navigateur et taper : <http://localhost:3000>

Si vous n'y arrivez pas, vérifier le **Firewalld** de Linux qui peut bloquer.

Ensuite, on atterri après avoir configuré ses logins ici :

The screenshot shows the Grafana home page. On the left, there's a sidebar with links like Home, Bookmarks, Starred, Dashboards, Explore, Drilldown, Alerting, Connections, and Administration. The main area has three cards: 'DATA SOURCE AND DASHBOARDS' (Incomplete), 'Grafana fundamentals' (Incomplete), and 'Create your first dashboard' (Incomplete). Below these are sections for 'Dashboards' (Starred dashboards, Recently viewed dashboards) and 'Latest from the blog' (two blog posts by Victoria Nulda).

Intégration de Influxdb :

Il faudra se rendre dans :

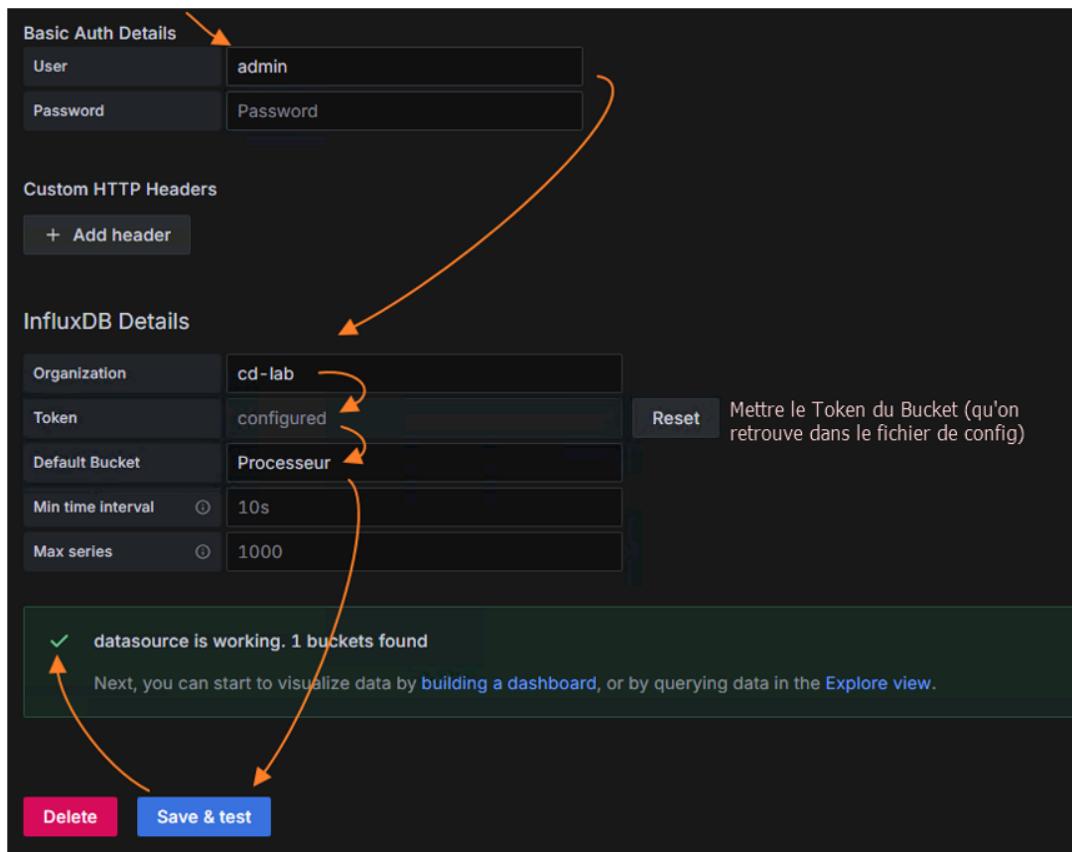
The screenshot shows the 'Add new connection' page in Grafana. The left sidebar has 'Connections' selected, with 'Add new connection' highlighted. The main area shows a search bar with 'Influx' typed in, and a list of data sources. 'InfluxDB' is listed under 'Data sources' with a status of 'Installed'. A red arrow points from the text above to the 'Add new connection' link in the sidebar, and another red arrow points from the text below to the 'InfluxDB' entry in the list.

Cliquez sur "[Add new data source](#)" en haut à gauche de l'écran.

Puis :

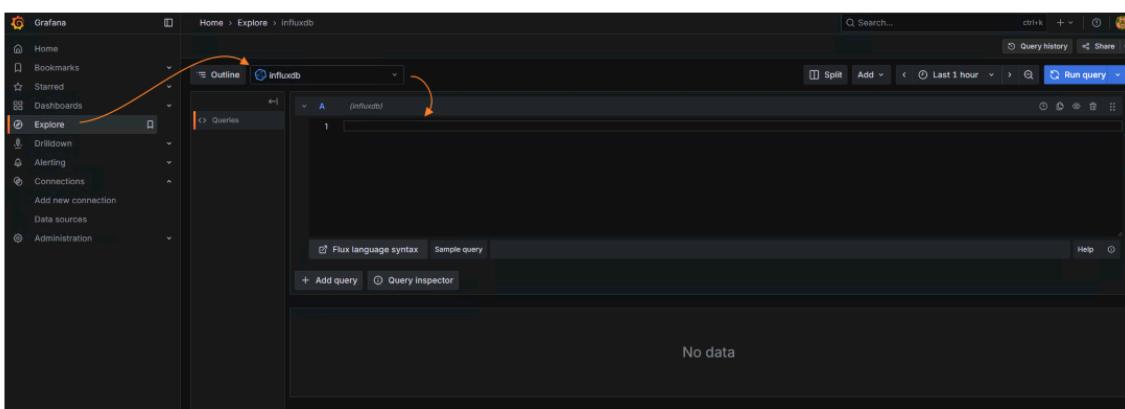
The screenshot shows the Grafana interface with the sidebar navigation open. The 'Data sources' option is selected, highlighted with an orange arrow. The main content area displays the configuration for a data source named 'influxdb'. The 'Query language' dropdown is set to 'Flux', indicated by another orange arrow. A note below states: 'Support for Flux in Grafana is currently in beta. Please report any issues to: https://github.com/grafana/grafana/issues'. The 'HTTP' section includes a 'URL' field containing 'http://192.188.1.50:8086', also highlighted with an orange arrow. Under the 'Auth' section, the 'Basic auth' toggle switch is turned on, while 'TLS Client Auth', 'Skip TLS Verify', and 'Forward OAuth Identity' are turned off.

Puis :



Désormais, on va mettre en forme tout ça !!

Se rendre ici :

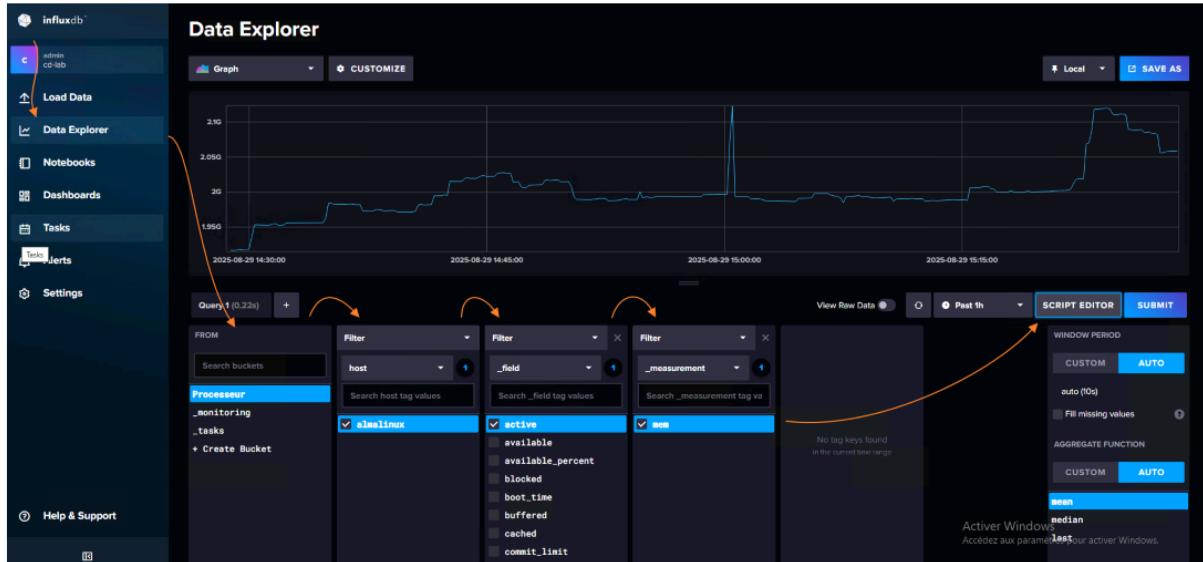


Pour éviter de s'embêter à apprendre les requêtes de FLUX
(ce qui reste possible via ce lien : <https://docs.influxdata.com/influxdb/v2/query-data/>)

Nous on va tricher un peu ;)
Se rendre sur l'interface graphique de Influxdb et aller ici :

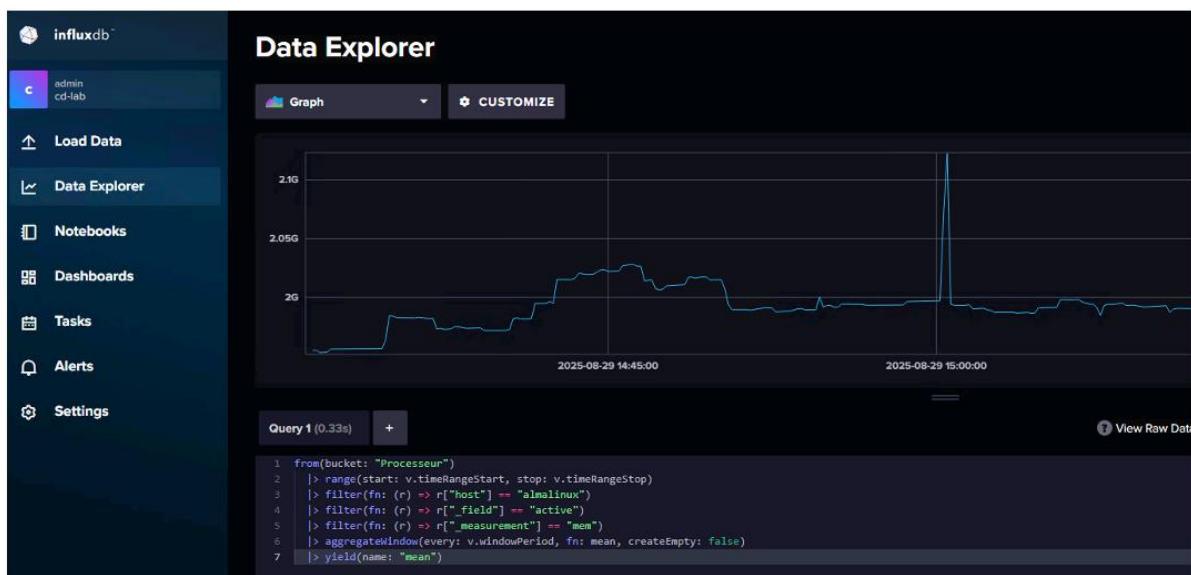
Nous on va tricher un peu ;)

Se rendre sur l'interface graphique de Influxdb et aller ici :

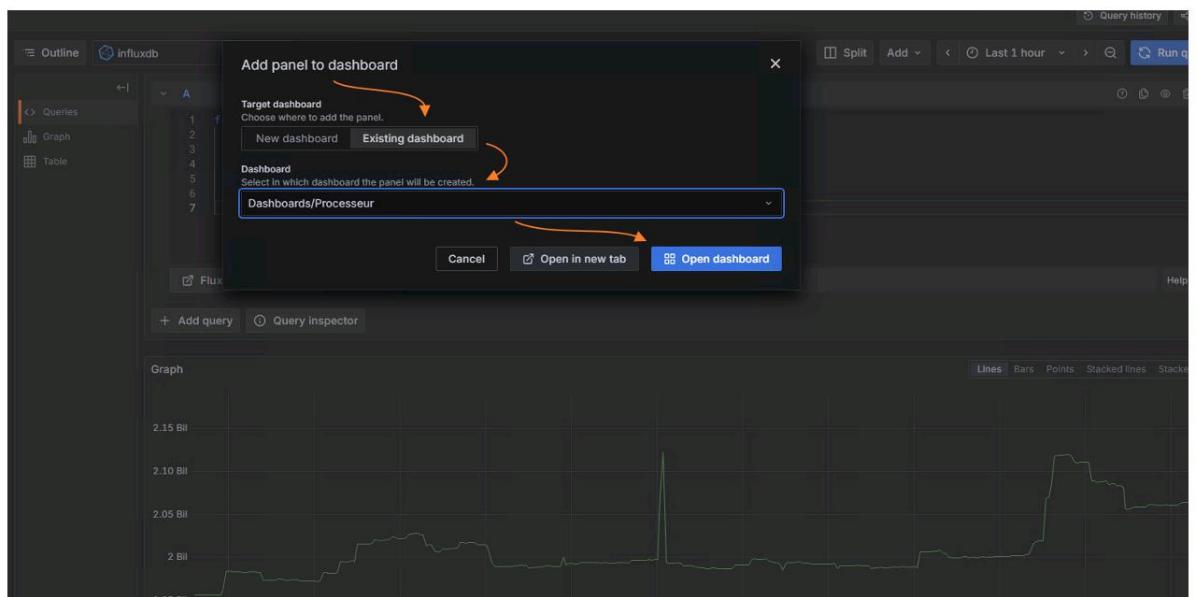
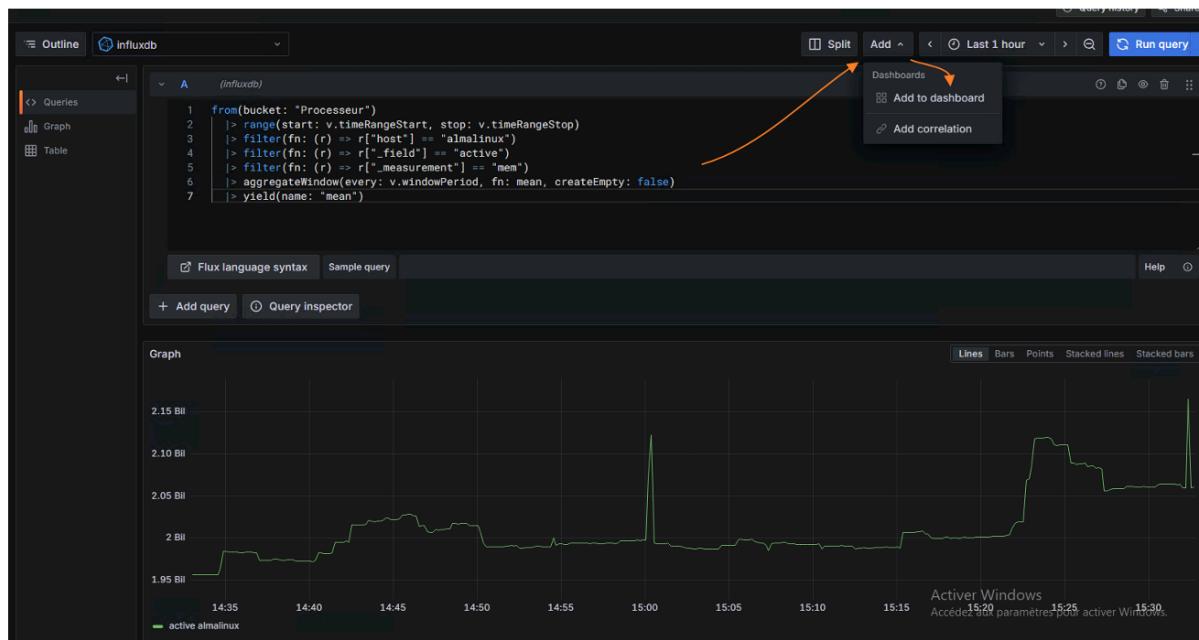


Une fois nos filtres configurés comme on le souhaite. En cliquant sur "SCRIPT EDITOR", on tombe sur une requête toute faite !

Comme ici :



Reste plus qu'à la copier-coller dans Grafana et à ajouter ce graph dans notre Dashboard :



✓ !!!!!!! ET ON A FINIT !!!!!!! ✓

