# Configuration

## InfluxDB

influx #Accéder aux commandes de InfluxDB

> CREATE DATABASE db\_grafana

> SHOW DATABASES;

> USE db\_grafana;

> CREATE USER user\_grafana WITH PASSWORD 'MotDePasse';

> GRANT ALL ON db\_grafana TO user\_grafana;

> CREATE RETENTION POLICY trente\_jours ON db\_grafana DURATION 30d REPLICATION 1 DEFAULT; #Conserver les données pendant 30 jours

> exit

sudo nano -c /etc/influxdb/influxdb.conf

[monitor]

store-interval = "300s"

## Grafana

### Changer le port (optionnel)

sudo nano /etc/grafana/grafana.ini

[server]

http\_port = 80 //Dé-commenter cette ligne en enlevant ";"

sudo service grafana-server restart

### Créer le compte admin de Grafana Server

Aller sur localhost:3000 pour se rendre sur l'interface graphique Grafana, se connecter avec **admin** / **admin** puis choisir le nouveau mot de passe pour le compte admin de Grafana.

### Ajouter sa base de données InfluxDB

Onglet Configuration à gauche (icone de l’engrenage) -> Data Sources -> + Add data sources

## C:\Users\Jean-Baptiste\Desktop\GrafanaInflux.png

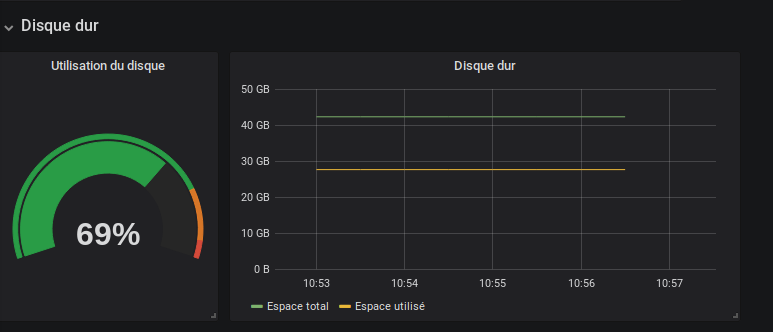
### Ajouter un dashboard

Un Dashboard (tableau de bord) regroupe plusieurs panneaux affichant les données métriques de matériels ou de services.

Exemple de création d'un Dashboard pour une machine monitorée :

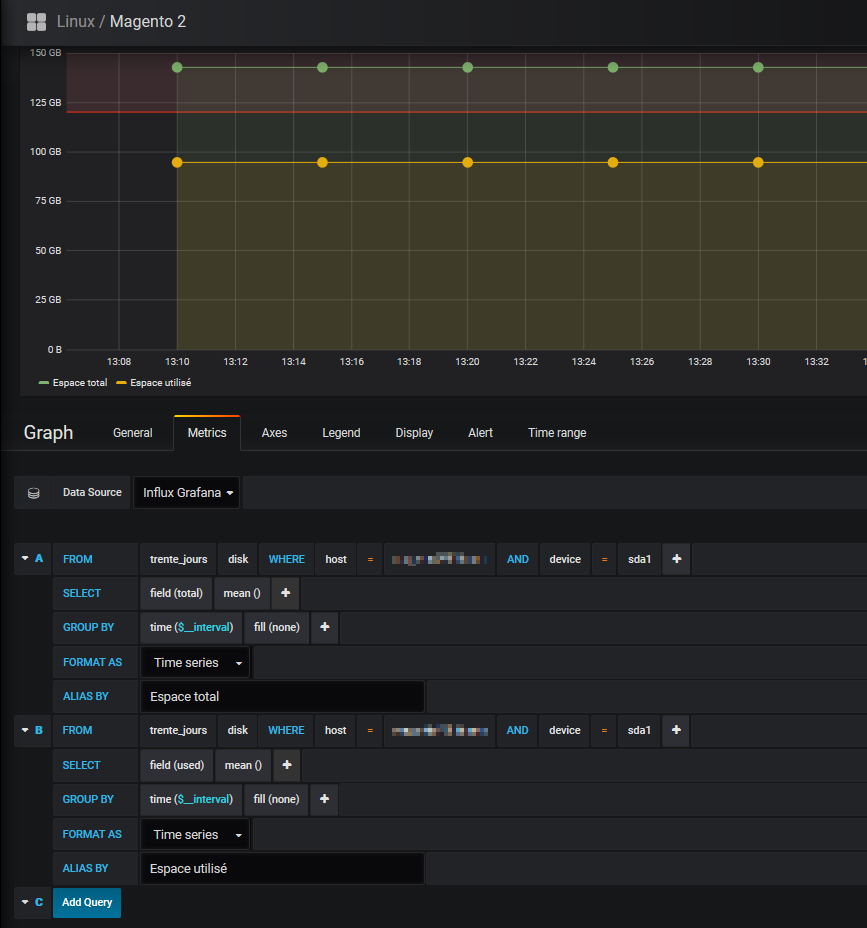
Onglet Dashboard à gauche -> Manage -> + Dashboard

Les deux panneaux les plus utilisés sont des Graph et des Singlestat. Il est impossible d'envoyer des alertes avec un Singlestat.

Gauche : Singlestat                                     Droite : Graph 

Une fois le panneau Graph ajouté, l'éditer en cliquant sur son nom (par défaut Panel Title)

**Exemple avec l’espace total et utilisé d’un disque dur :**



### Importer un dashboard

Il est possible d'importer des Dashboard déjà configurés en copiant le JSON d’un dashboard déjà exsistant. Il y existe une liste de dashboard créés par d'autres utilisateurs sur <https://grafana.com/dashboards>. Les Dashboard importés ne sont que des graphiques, il faut toujours configurer soi-même les metrics et le collecteur telegraf si besoin.

Une fois l’ID ou le JSON copié : Onglet Dashboard à gauche -> Manage -> + Import

## Telegraf

Il faut relancer le service Telegraf après chaque modification pour les prendre en compte.

Copier le fichier de config d'origine pour en faire une sauvegarde puis modifier le fichier de config actuel

sudo cp /etc/telegraf/telegraf.conf /etc/telegraf/telegraf.conf.orig

sudo nano /etc/telegraf/telegrag.conf

### Configuration du collecteur

[agent]

#Définit tous les combien de temps Telegraf va collecter les données, par défaut à 10s

interval ="300s"

flush\_interval = "300s"

**flush\_interval** ne doit pas être inférieur à **interval**

### Configuration avec la base de données

#Permet de configurer Telegraf avec la base de données InfluxDB du serveur de monitoring

[[outputs.influxdb]]

#Adresse du serveur qui contient InfluxDB

urls = ["http://192.168.0.10:8086"]

database = "db\_grafana"

#Indiquer que l'on veut utiliser la police de rétention créée pour lui dire de stocker les données pendant 30 jours

retention\_policy = "trente\_jours"

timeout = "15s"

username = "db\_user"

password = "MotDePasse"

### Configuration des services

sudo service --status-all #Pour lister tous les services sur Ubuntu

[[inputs.procstat]]

pattern = "tomcat7"

[[inputs.procstat]]

pattern = "postgresql"

[[inputs.procstat]]

pattern = "apache2"

### Test de la connexion avec InfluxDB

Pour tester l'envoi des données vers la base de données, entrer dans l'URL [**http://192.168.0.10:8086**](http://192.168.0.10:8086). Si une page « 404 page not found » s'affiche, le serveur monitoré peut envoyer ses données, sinon il y a timeout ou connexion refusée. Dans ce cas, régler les pares-feu.

## Telegraf pour Windows

Le fichier se trouve dans C:\Program Files (x86)\Telegraf\telegraf.conf

#L20 - Définit tous les combien de temps Telegraf va collecter les données, par défaut à 10s

interval = "300s

#L39 - Ne doit pas être inférieur à interval

flush\_interval = "300s"

#L51 - Ajouter le (x86) (s’il y est dans Program Files)

logfile = "/Program Files (x86)/Telegraf/telegraf.log"

#L62 - Même principe de configuration que pour la version Linux

[[ouputs.influxdb]]

urls = ["http://192.168.0.10:8086"]

database = "db\_grafana"

retention\_policy = "trente\_jours"

precision = "s"

timeout = "15s"

username = "user\_grafana"

password = "MotDePasse"

En plus de laisser les inputs.win\_perf, on peut utiliser les inputs classiques comme inputs.mem, inputs.disk ... à partir de la ligne 225 qui utilisent plus de ressources système mais qui apportent plus de données.

Pour le moniroting des composants, inputs.cpu n'a pas besoin d'être activé car le inputs.win\_perf\_counter du Processor est déjà suffisant.

inputs.disk et inputs.mem sont activés (ligne 239 et 260) car ils apportent le total et le used par exemple.

### Monitoring des services

Ajouter dans le fichier de conf ligne 223 :

[[inputs.win\_services]]

    service\_names = [

    "SQLSERVERAGENT",

    "AtlassianBitbucket",

    "GMC License Manager",

    "redmineApache",

    "W3SVC", //Correspond à IIS

    "MSSQLSERVER",

    "MySQL57"

    ]

Ne mettre aucun service les affichera tous

.

### Commandes Powershell pour utiliser Telegraf

Si Telegraf fonctionne, il devrait retourner après la commande de test des valeurs pour différents composants de Windows.

C:\"Program Files (x86)"\Telegraf\telegraf.exe --config C:\"Program Files (x86)"\Telegraf\telegraf.conf --test

Dans les services Windows, le service est **Telegraf Data Collector Service**.

### Désinstaller Telegraf

C:\"Program Files (x86)"\Telegraf\telegraf.exe --service uninstall

+ supprimer le dossier Telegraf