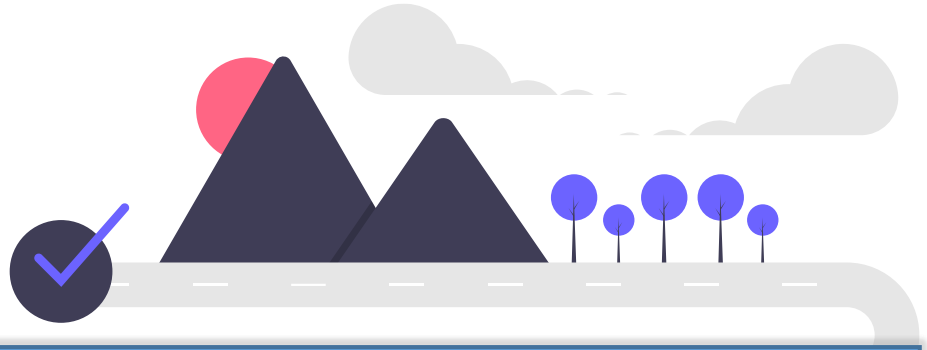
OMNISHORE

Observability & Monitoring



PROMETHEUS & GRAFANA

Marius **N'KOUBA**

Data Engineer

May 2024

Plan: Prometheus

- Définitions et notions
- Installation de Prometheus
- Jobs & Instances
- Interface Graphique
- Node , PostgresSQL Exporter, Mysql Exporter
- PromQL
- AlertManager

Plan : Grafana

- Définitions et notions
- Installation de Grafana
- Ajouter des sources
- Importer des Dashboards Depuis Grafana Labs
- Les Dashboards et les variables
- Gestion des utilisateurs

Monitoring

Prometheus

Définition

Qu'est-ce que Prometheus ? 🤔

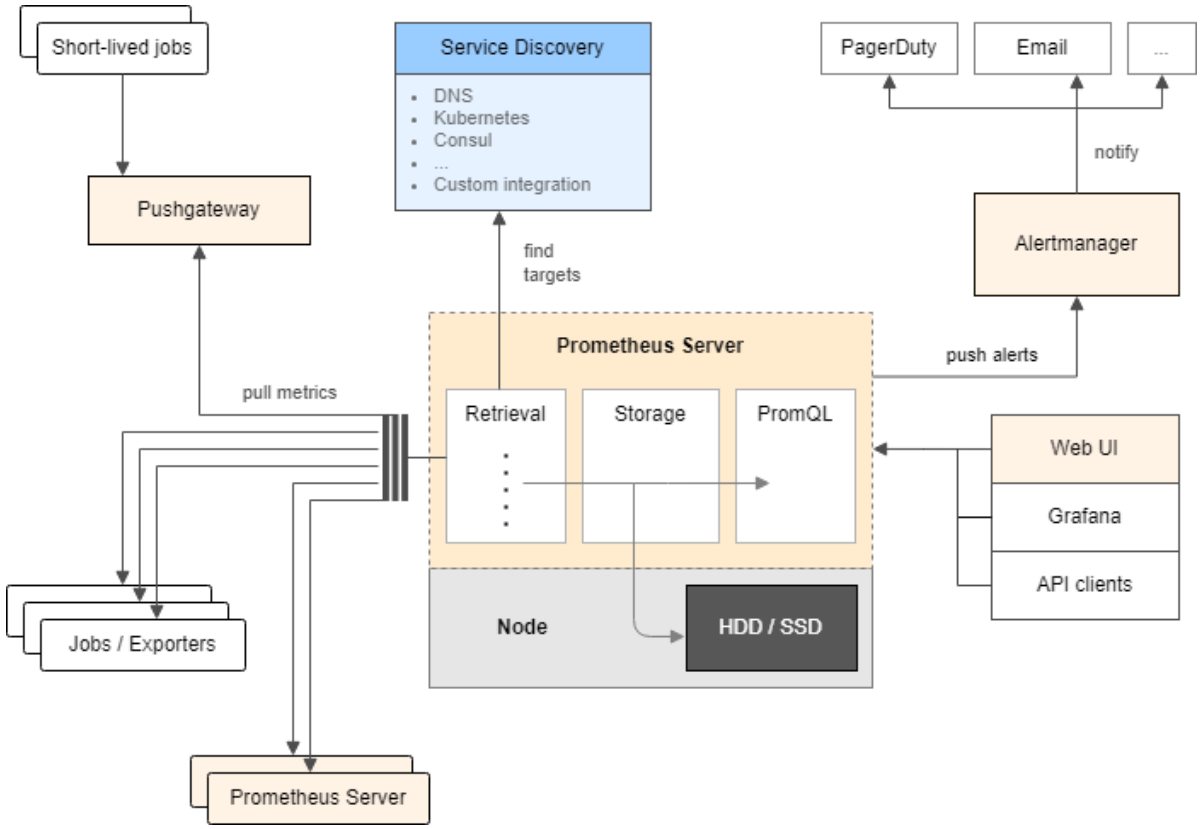
Prometheus, est un système de surveillance des systèmes et des services.

Il collecte des métriques à partir de cibles configurées à des intervalles donnés, évalue les expressions de règles, affiche les résultats et peut déclencher des alertes lorsque des conditions spécifiées sont observées.

- Un modèle de données multidimensionnel (série chronologique définie par le nom de la métrique et un ensemble de dimensions clé/valeur)
- PromQL, un langage de requête puissant et flexible
- Aucune dépendance au stockage distribué
- Un modèle HTTP pull pour la collecte de séries chronologiques
- Les cibles sont découvertes via la découverte de services ou la configuration statique
- Plusieurs modes de prise en charge des graphiques et des tableaux de bord
- Prometheus se base sur le Langage Go
- base de données Base Time series + Serveur Web + Moteur de Base de Données

Définition

Qu'est-ce que Prometheus ? 🤔

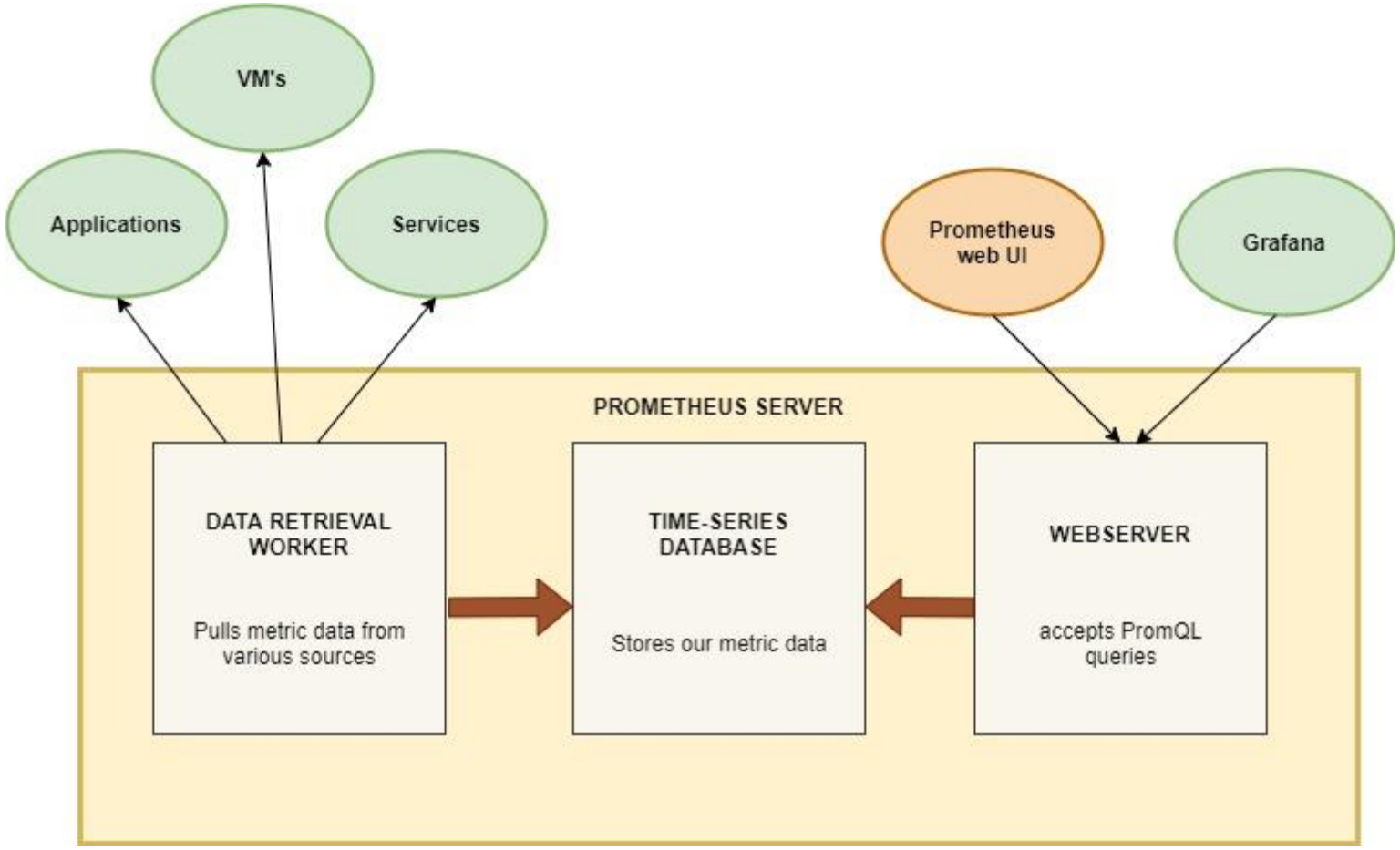


- Il fait essentiellement du Scraping de données à des fréquences régulières
- Principe de stockage : Clé / Valeur / Timestamp
- Il travaille en double Delta:

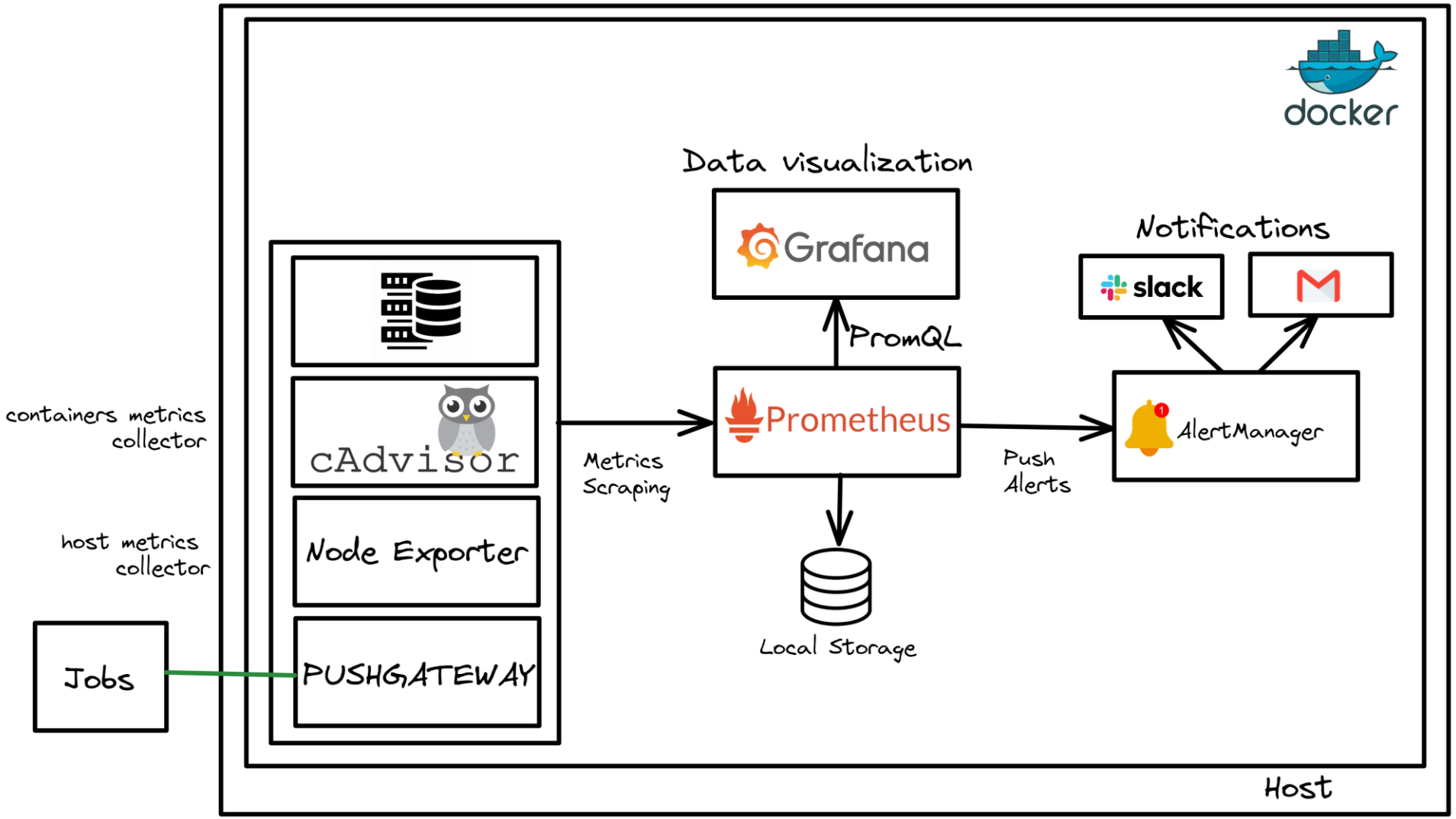
Double delta:

Calcul l'écart entre une valeur et sa valeur précédente, Si ca n'évolue pas : il ne fait pas de modifications sinon

Monitoring using prometheus



Monitoring avec Docker



Types de métriques

Counter (compteur)

Un compteur est une valeur qui augmente de manière monotone. Il est souvent utilisé pour suivre des événements qui se produisent de manière incrémentielle, tels que le nombre de requêtes HTTP reçues ou le nombre de fois qu'une certaine opération a été effectuée.

Gauge (jauge)

Une jauge est une valeur qui peut augmenter ou diminuer de manière arbitraire au fil du temps. Elle est souvent utilisée pour représenter des mesures instantanées, telles que la taille actuelle d'une file d'attente, la quantité de mémoire utilisée ou le nombre d'utilisateurs actuellement connectés.

Types de métriques

Histogram (histogramme)

Un histogramme est utilisé pour suivre la distribution des valeurs d'une variable au fil du temps. Il divise les valeurs en intervalles (appelés "buckets") et compte le nombre d'occurrences dans chaque intervalle.

Cela permet d'analyser la distribution des valeurs, par exemple, le temps passé dans différentes phases d'exécution d'une requête

Summary (résumé)

Un résumé est similaire à un histogramme, mais plutôt que de compter les occurrences dans des intervalles prédéfinis, il calcule des quantiles (par exemple, le 50e, le 90e et le 99e percentile) des valeurs observées.

Cela permet de mieux comprendre la distribution des valeurs et d'identifier les tendances de performance

Installation de Prometheus

OMNISHORE

Installation paquet Debian

sudo apt update
sudo apt-get install prometheus

Configuration de prometheus.yml

sudo nano /etc/prometheus/prometheus.yml

Prometheus

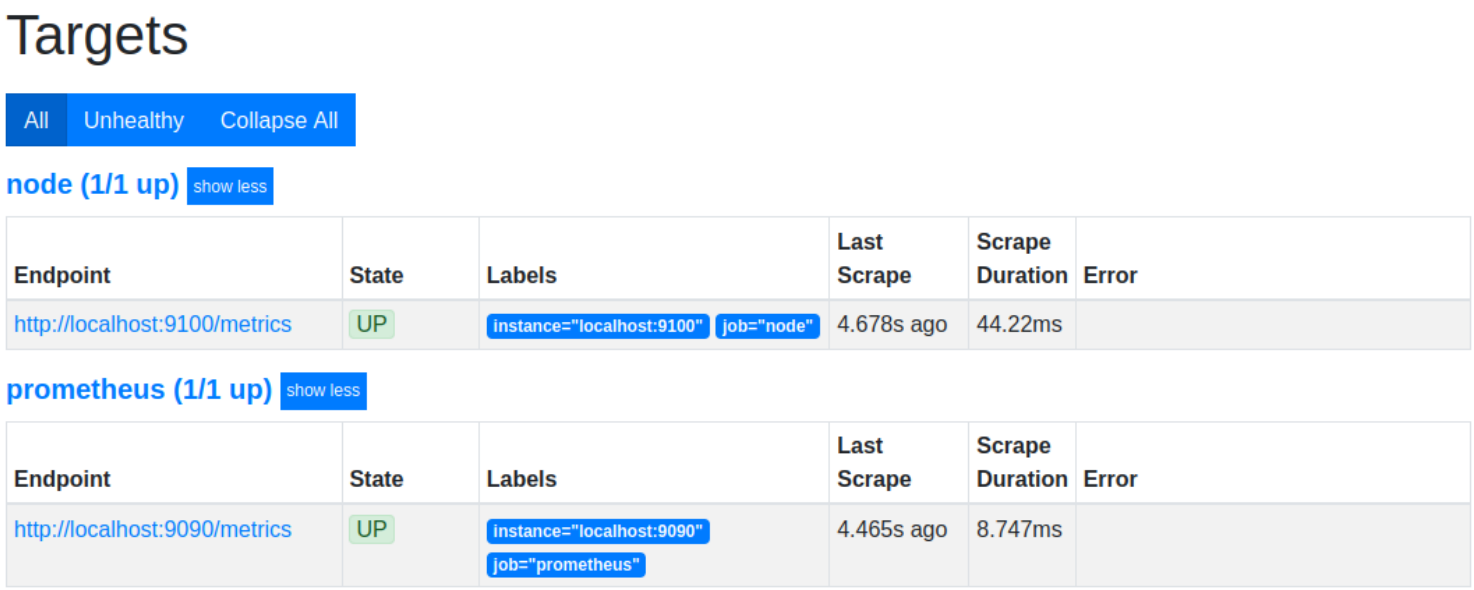
http://localhost:9090

Prometheus Metrics

http://localhost:9090/metrics

Node Metrics

http://localhost:9100/metrics



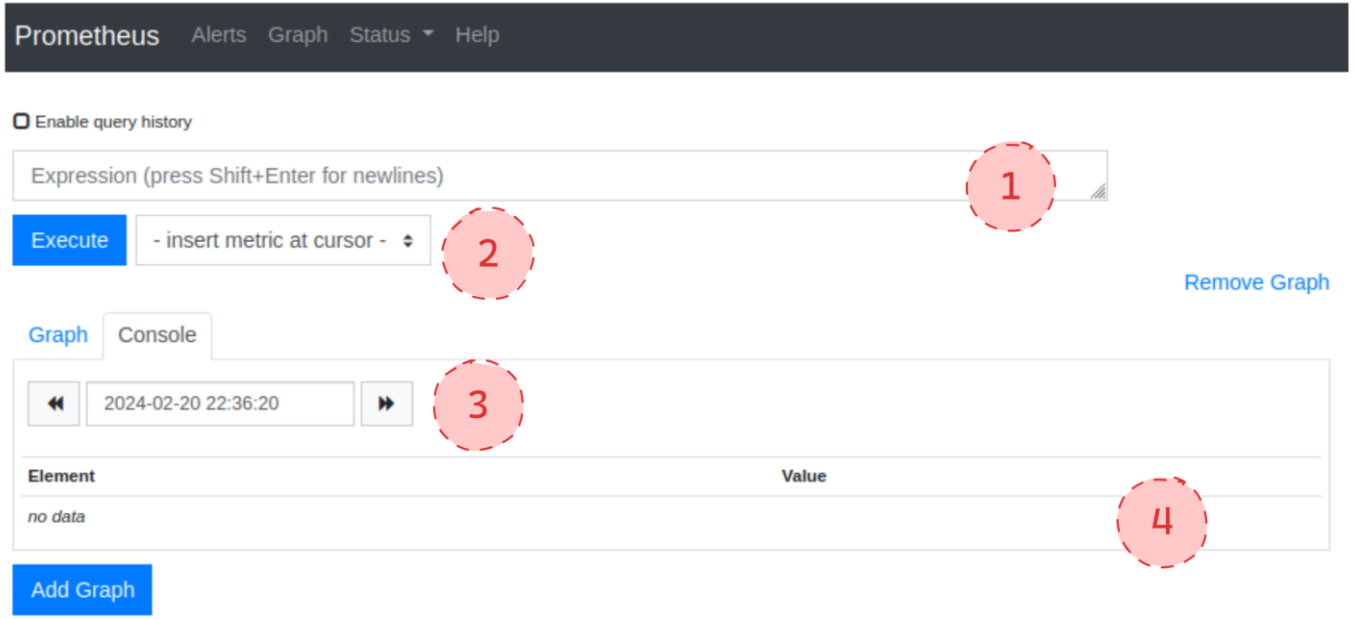
Configuration prometheus.yml

- Configuration
  - Global : Configurations qui s'appliquent en général
    - scrape_interval (ou job): Intervale de récupération des données
    - evelaution_interval : réévaluation des rules (alertes)
    - scrape_timeout: timeout lors du scraping

  - rule_files: configuration des alertes

  - scrape_configs : configuration particulière du scraping
    - job_name : nom du bloc (http://localhost:9090/classic/service-discovery)
    - metric_path: route de scraping (/metrics)
    - static_config :
      - labels: Definition de labels (important pour grafana/ Standardiser)
      - targets : url/ip:port

Interface graphique




```
go_threads
go_threads{instance="localhost:9090",job="prometheus"}
```

Installation de quelques Exporters

Node exporter

Agent qui permet de collecter des métriques systèmes :

- disques
- mémoire
- cpu
- load average
- nfs
- time
- uname
- vmstat
- Stats

Nombres de métriques : 500

Exemple de métriques retournés:
https://grafana.com/oss/prometheus/exporters/node-exporter/assets/node_exporter_sample_scrape.txt

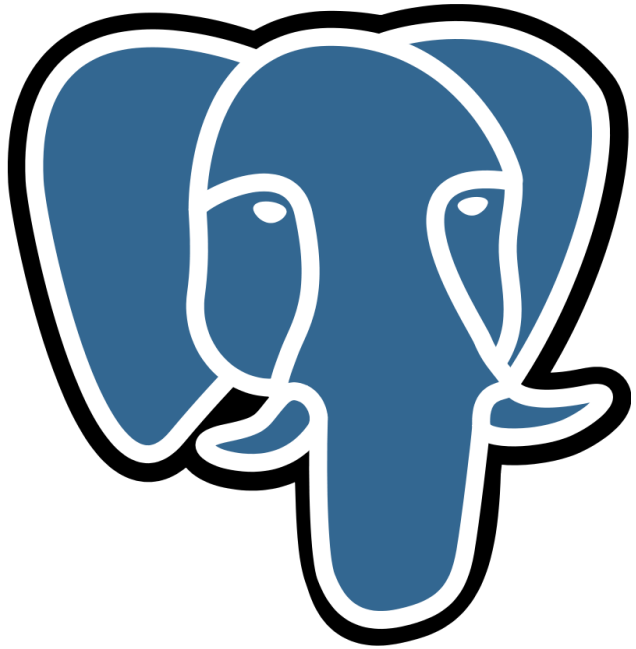
https://grafana.com/oss/prometheus/exporters/node-exporter/
https://github.com/prometheus/node_exporter

Postgres exporter

Agent qui permet de collecter des métriques d'une base de données Postgres

Nombres de métriques : 450

Exemple de métriques retournés:
https://grafana.com/oss/prometheus/exporters/postgres-exporter/assets/postgres_metrics_scrape.txt



https://grafana.com/oss/prometheus/exporters/postgres-exporter/
https://github.com/prometheus-community/postgres_exporter

Mysql exporter

Agent qui permet de collecter des métriques d'une base de données Mysql

Nombres de métriques : 1000

Exemple de métriques retournés:
https://grafana.com/oss/prometheus/exporters/mysql-exporter/assets/mysql_metrics_scrape.txt

https://github.com/prometheus-community/postgres_exporter

PromQL

PromQL

Filtres sur les Labels

- Equivalent à une clause WHERE
- Utilise les regex GO
- https://prometheus.io/docs/practices/naming/ (Documentation)

Renommer un Label

```
relabel_configs:
      - source_labels: [__meta_consul_service]
        target_label: job
```

WHERE

```
up{job="node"}
up{job="node_exporter"}
node_network_receive_bytes_total{device="ens33"}
```

DIFFERENT DE

```
node_network_receive_bytes_total{device!="ens33"}
```

REGEX

```
node_network_receive_bytes_total{device=~"ens.*"}
node_network_receive_bytes_total{device=~"ens33|lo"}
```

MULTIPLE FILTRES

```
node_network_receive_bytes_total{instance=~"localhost:.+",
device=~"ens33|lo"}
```

PromQL

Opération d'aggrégation

- count(node_cpu_seconds_total)

- count(node_cpu_seconds_total) by(instance)
- count(node_cpu_seconds_total) by(instance, cpu)

- count(count(node_cpu_seconds_total) by(instance, cpu)) by (instance)

Intervalles de temps

Range Vector : []
- cherche tous les timeseries sur un interval de temps
- par defaut par rapport à maintenant

```
node_load1[2m]
```

Offset : offset 3m
- change la date à laquelle on recherche la timeseries

```
node_load1[1m] offset 3m
```

Conversion

```
date -d "@1711004584.27"
```

Unités de temps

```
s : seconds
m : minutes
h : hours
d : days
w : weeks
y : years
```

PromQL

Opération d'aggrégation

- sum_over_time(node_cpu_seconds_total[3m])
- avg_over_time(node_cpu_seconds_total[3m])
- sum_over_time(node_load1[3m])

Offset

Idéal pour comparer deux périodes

node_load1 - node_load1 offset 30m
node_load1 - sum_over_time(node_load1[5m] offset 5m)

PromQL

Manipulation de Labels

élément pour filtrer et typer des métriques

Skipper les métriques de nodes commençant par 192.168(.*)

```
scrape_configs:
  - job_name: prometheus
    static_configs:
      - targets: ['localhost:9090']
  - job_name: node_exporter
    static_configs:
      - targets: ['localhost:9100']
    relabel_configs:
      - source_labels: [__address__]
        regex: '192.168(.*)'
        action: drop
```

Garder les métriques de nodes commençant par 192.168(.*)

```
scrape_configs:
  - job_name: prometheus
    static_configs:
      - targets: ['localhost:9090']
  - job_name: node_exporter
    static_configs:
      - targets: ['localhost:9100']
    relabel_configs:
      - source_labels: [__address__]
        regex: '192.168(.*)'
        action: keep
```

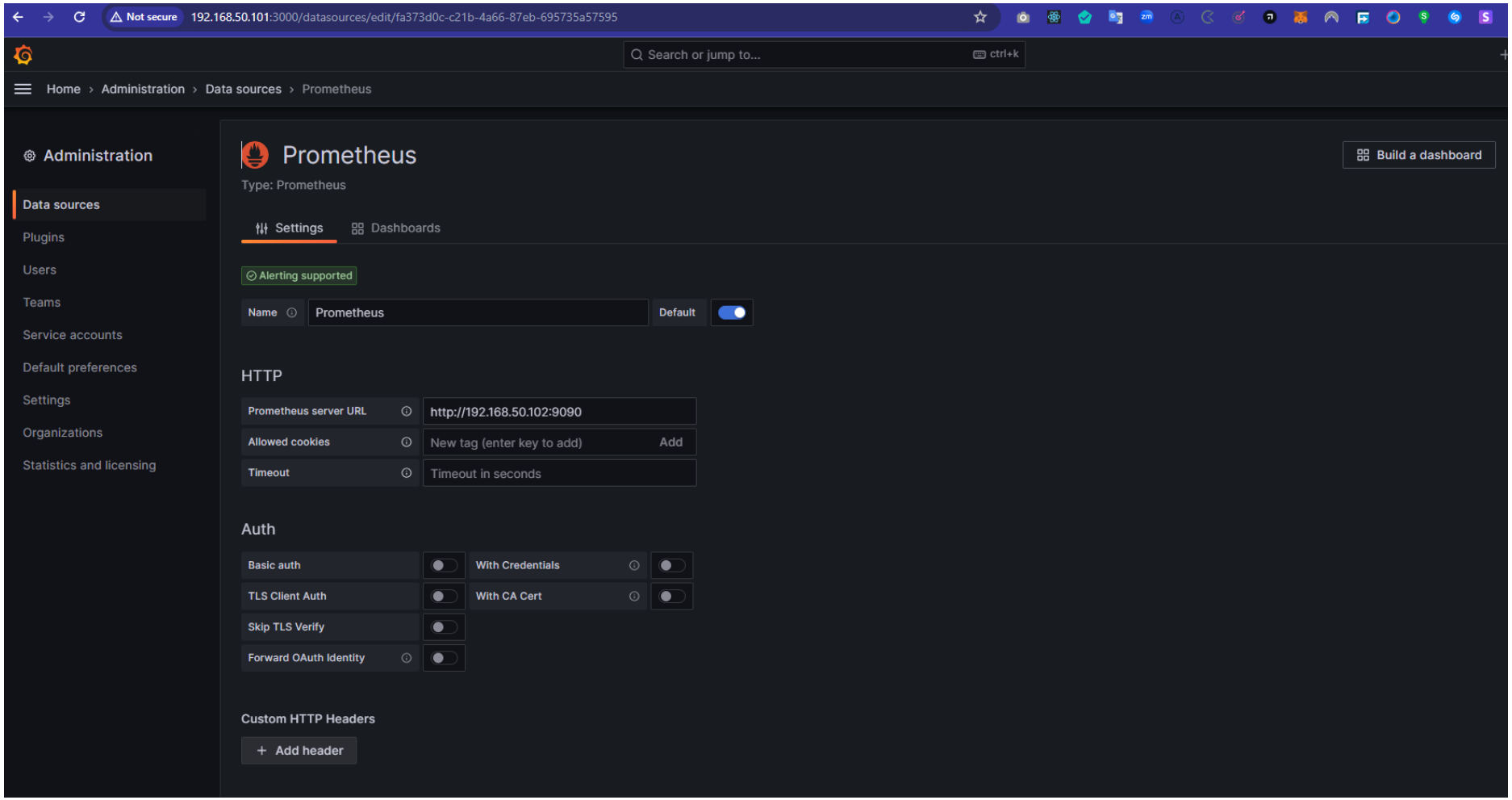
Grafana

Grafana

Grafana est une plateforme de visualisation de données interactive Open Source développée par Grafana Labs, qui permet aux utilisateurs de consulter leurs données via des graphiques unifiés dans un ou plusieurs tableaux de bord afin de mieux les interpréter et les comprendre.

C’est un outil pour afficher données chronologiques. A partir d’une série de données collectées, nous obtiendrons un panorama graphique de la situation d’une entreprise ou organisation

Ajouter le DataSource Prometheus

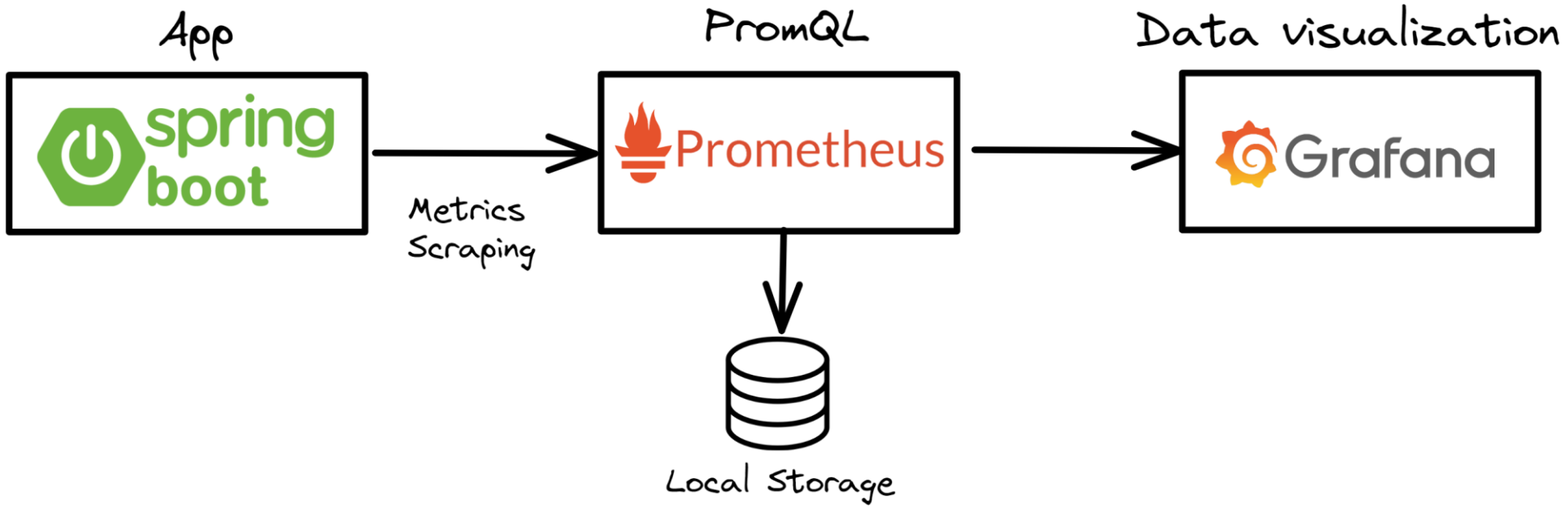


Exporters Dashboard ID

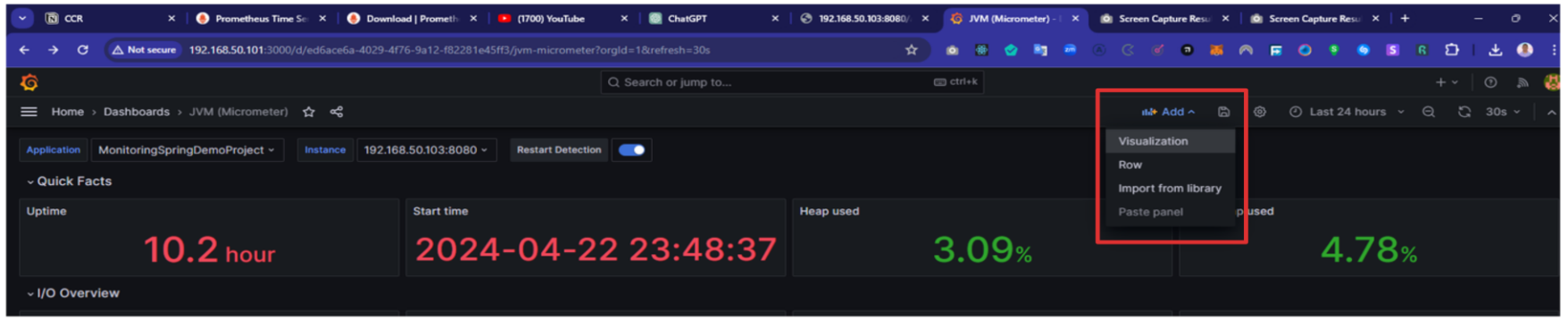
Node Exporter : **1860**
Postgres Exporter : **9628**
Mysql Exporter : **14057**

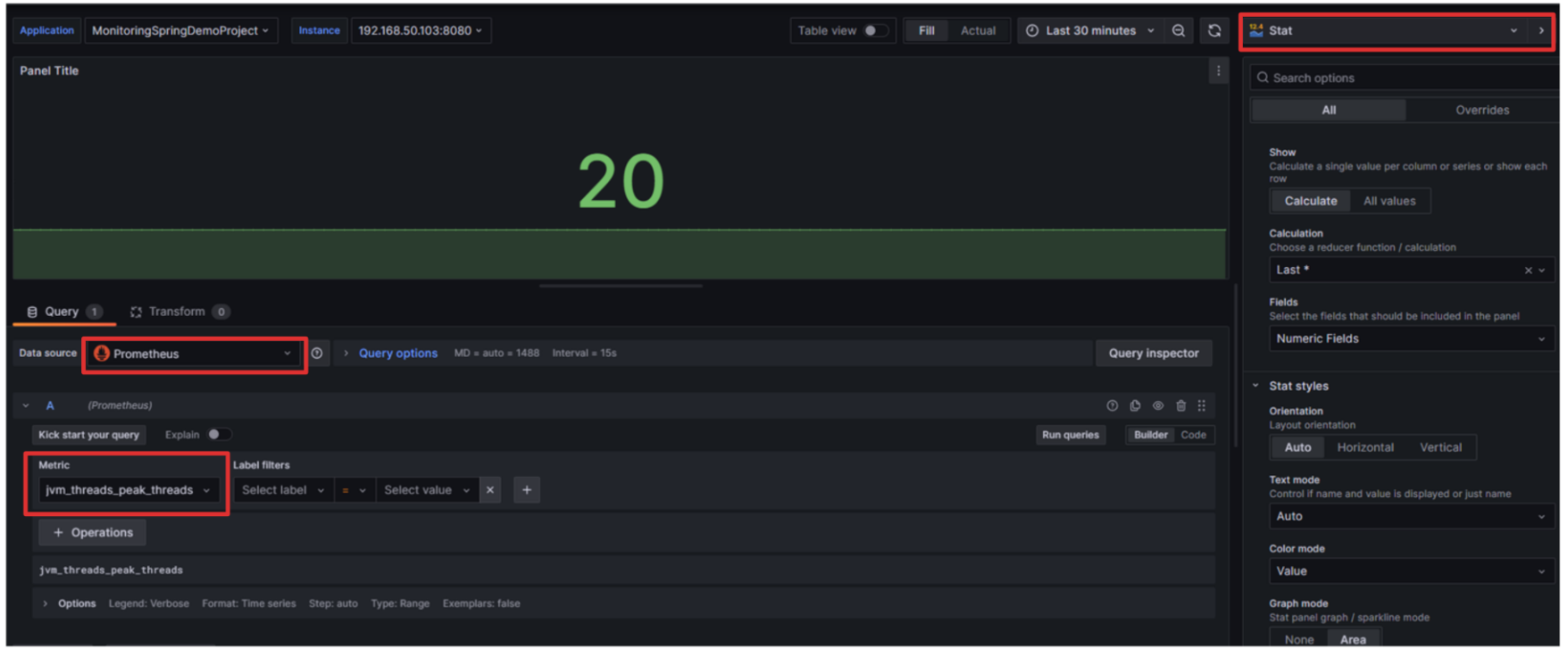
Monitoring of SpringBoot App

OMNISHORE



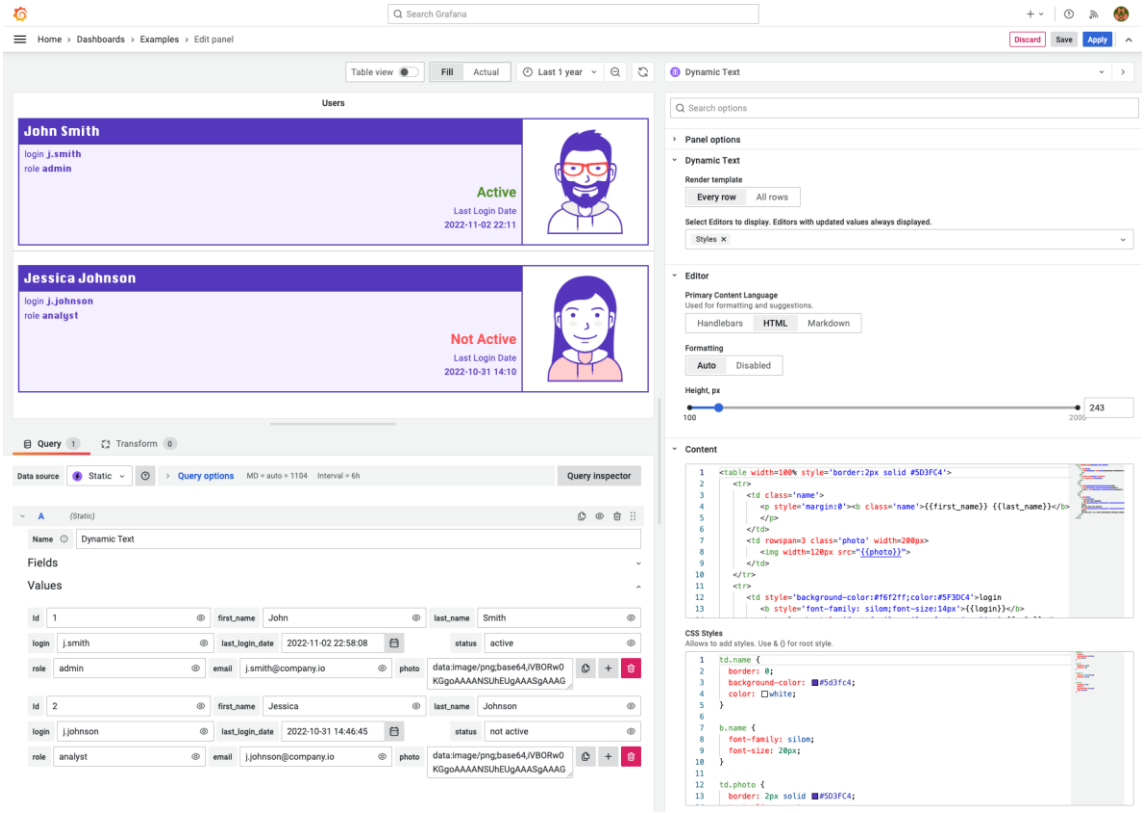
AJOUT D'UN PANEL DE MÉTRIQUE PERSONNALISÉ





Plugins dans Grafana

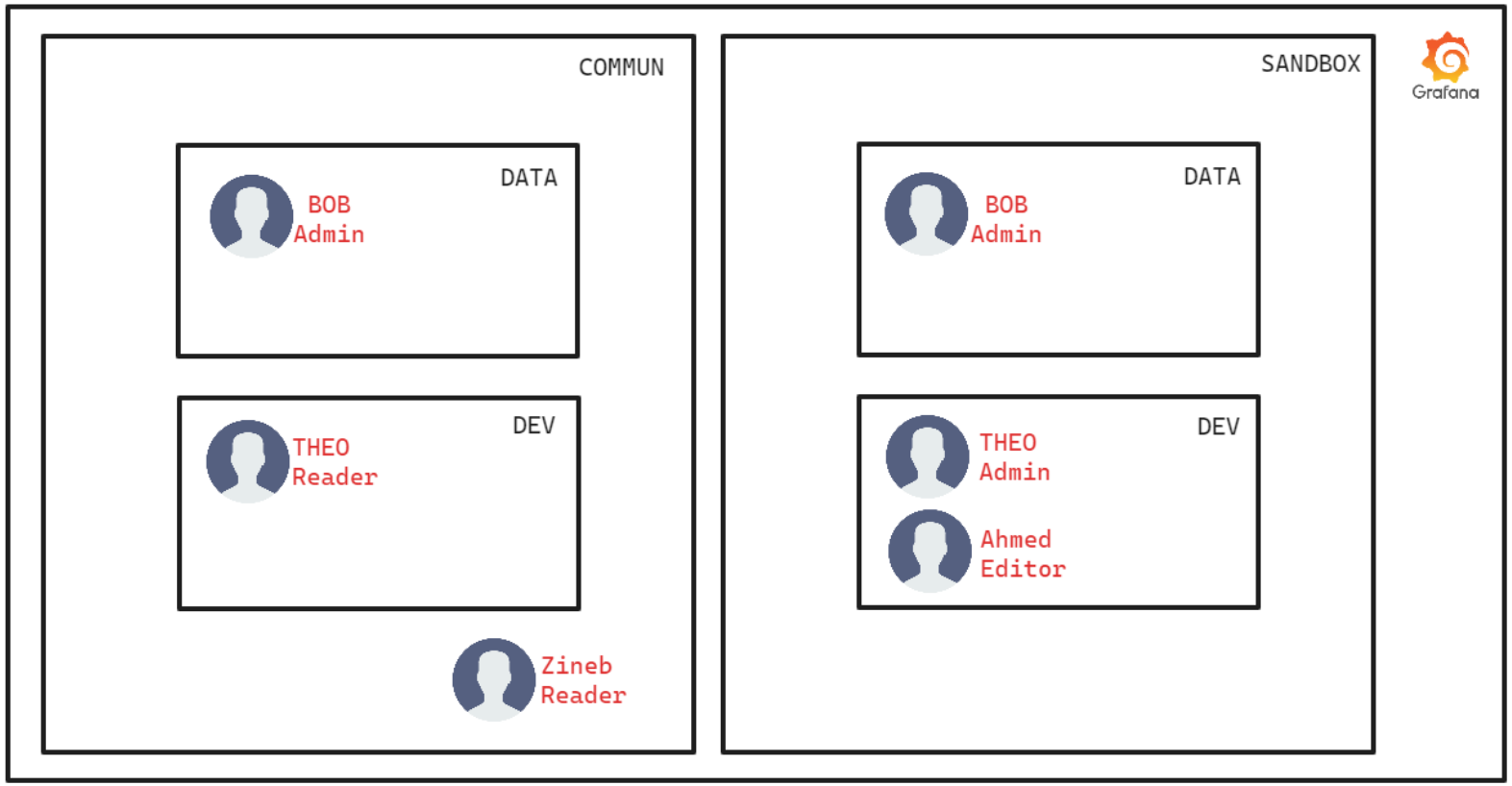
Dynamic Text Panel for Grafana



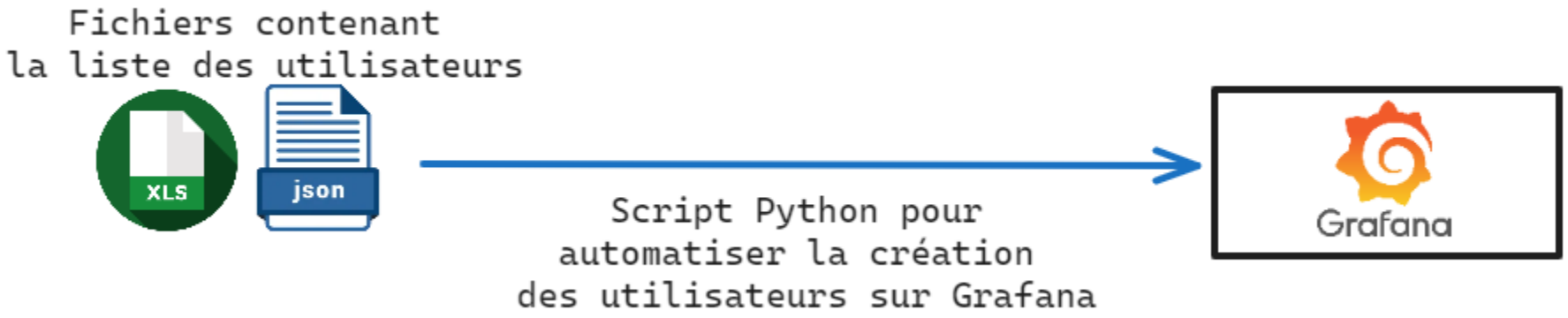
https://grafana.com/grafana/plugins/all-plugins/

TP: Variables dans Grafana

TP: Gérer les utilisateurs d'une organisation



TP: Gérer les utilisateurs d'une organisation



Questions