

Monitoring: Prometheus et Grafana



**UP ASI
Bureau E204**

Plan du cours

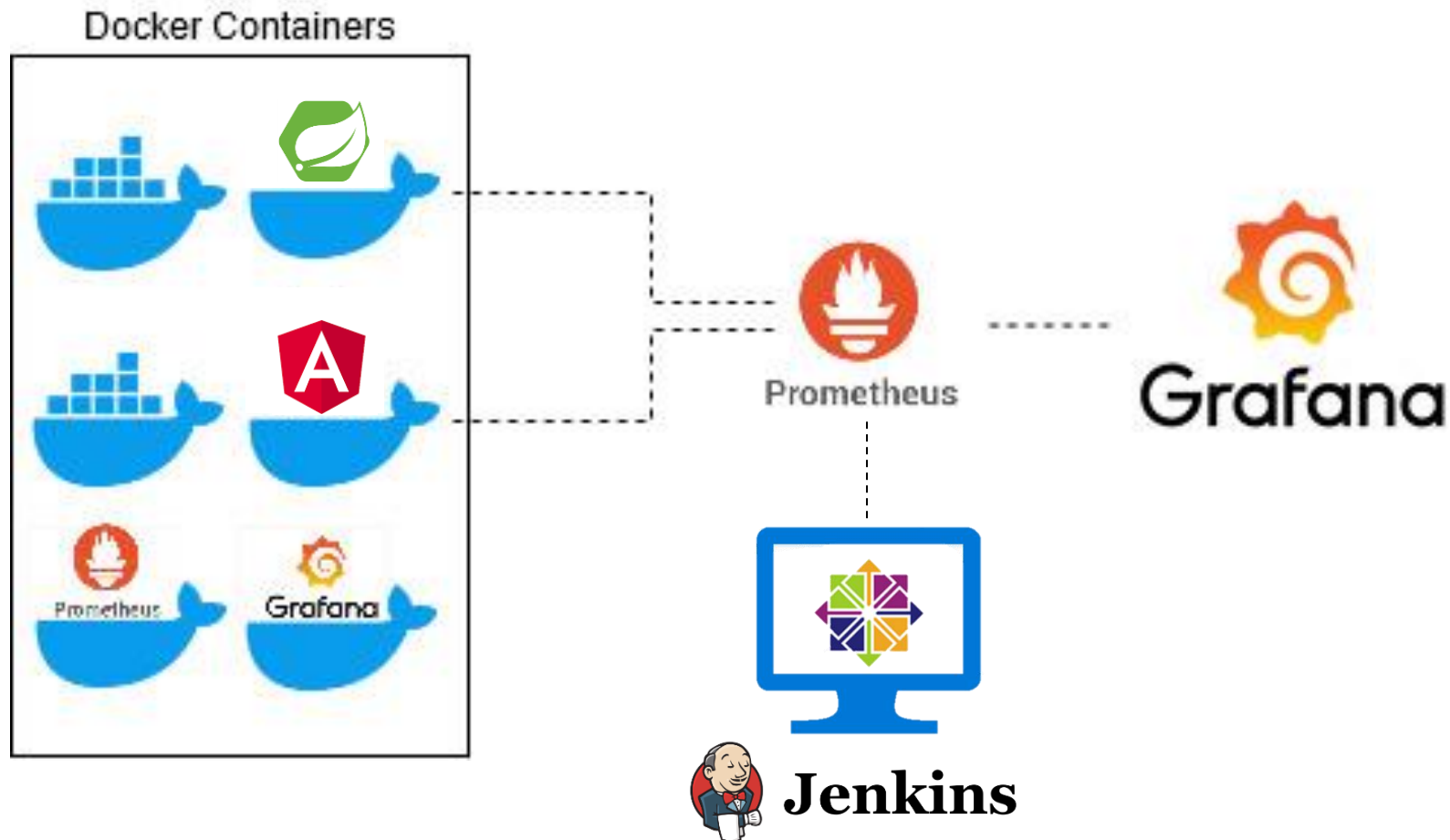
- Introduction: Superviser une application
- Prometheus
 - Définition
 - Installation
 - Configuration
- Grafana
 - Définition
 - Architecture
 - Installation
 - Configuration
- Travail à faire

Introduction

- La supervision des applications (Monitoring) est le processus de **collecte des données** de log afin d'aider l'équipe à suivre la disponibilité, les bugs, l'utilisation des ressources et les changements de performance des applications qui affectent l'expérience de l'utilisateur final (UX).
- Les outils de surveillance des applications fournissent des alertes en cas d'anomalies en direct et, grâce à un traçage distribué, permettent de suivre les événements qui forment une chaîne de causalité (impact d'un événement sur divers outils).

Introduction

→ Pour notre cas, nous allons utiliser Prometheus et Grafana pour superviser l'environnement sur lequel nous travaillons.



Prometheus - Définition

- Prometheus est un système de monitoring offrant une base de données de séries chronologiques.
- Il est basé sur le langage Go.
- Il est conçu pour surveiller des cibles (Serveurs, Bases de données, Machines virtuelles) → A peu près tout peut être surveillé avec Prometheus.
- Il travaille en **double delta**: Il calcule l'écart de la valeur par rapport à sa valeur précédente pour le suivi de performances.
- Il est composé de :
 - Une base de données (TimeSeries)
 - Un serveur web
 - Un moteur de base de données

Prometheus - Définition

- **Fonctionnement:** Il cherche sur une route donnée (@IP, URL, ...) les informations nécessaires (**Metrics**) et les stocke dans sa base de données interne (TimeSeries) avec un format standardisé (Clé, timestamp, valeur).
- Il fournit un langage de requêtage appelé **PromQL (Prometheus Query Language)** qui permet à l'utilisateur de sélectionner et d'agréger des données de séries chronologiques en temps réel. Le résultat d'une requête peut être affiché sous forme de graphique, de données tabulaires dans l'interface de Prometheus ou consommé par des systèmes externes via l'API HTTP (Grafana).

→ Très facile à mettre en place → Très performant → Scrapping continu

Prometheus - Installation

- Vous allez utiliser une image Docker.

`docker pull prom/Prometheus`

```
[root@localhost vagrant]# docker pull prom/prometheus
```

- Création d'un conteneur Docker Prometheus.

`docker run -d --name prometheus -p 9090:9090 prom/prometheus`

```
[root@localhost vagrant]# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
9c45242eec8e	prom/prometheus	"/bin/prometheus -..."	11 days ago	Up 4 seconds	0.0.0.0:9090->9090/tcp	prometheu

Prometheus - Configuration



- Accéder à l'interface web de prometheus :

<http://@IP:9090>

The screenshot shows the Prometheus web interface in a browser. The address bar displays "192.168.1.244:9090/targets?search=" with a security warning. The browser's tab bar shows several open tabs related to DevOps and Spring. The Prometheus navigation bar includes links for Alerts, Graph, Status, and Help. The main content area is titled "Targets" and features a filter bar with buttons for "All", "Unhealthy", and "Collapse All", along with a search input field. Below the filter bar, a summary indicates "prometheus (1/1 up)" with a "show less" link. A table lists the targets, showing one target for "http://localhost:9090/metrics" which is in an "UP" state. The table also displays labels, the last scrape time, and the scrape duration.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	33.383s ago	7.782ms	

Prometheus - Configuration



- Installer le plugin « Prometheus metrics » dans Jenkins:

The screenshot shows the Jenkins web interface. The browser address bar indicates the URL `192.168.1.244:8080/manage/pluginManager/installed`. The Jenkins header includes the logo, a search bar, and user information (admin). The breadcrumb trail is `Tableau de bord > Administrer Jenkins > Gestion des plugins`. The 'Plugin Manager' section has tabs for 'Mises à jour', 'Disponibles', 'Installés', and 'Avancé'. A search bar contains the text 'prome'. A table lists installed plugins, with the 'Prometheus metrics plugin' (version 2.0.11) highlighted. The table has columns for 'Nom' and 'Activé'. The 'Prometheus metrics plugin' entry shows the description 'Expose Jenkins metrics in prometheus format' and a link to 'Report an issue with this plugin'. The 'Activé' column shows a toggle switch that is turned on.

Nom ↓	Activé
Prometheus metrics plugin 2.0.11 Expose Jenkins metrics in prometheus format Report an issue with this plugin	<input checked="" type="checkbox"/>

➔ Le plugin « Prometheus metrics » ne supporte pas le chargement dynamique. Jenkins doit être redémarré pour que la mise à jour soit effective (`systemctl restart jenkins`).

Prometheus - Configuration



- Pour ajouter un serveur, vous devez accéder au fichier de configuration de l'outil « Prometheus » et ajouter un « job »

```
docker exec -it prometheus sh
```

```
tee -a /etc/prometheus/prometheus.yml <<EOF
```

```
- job_name: jenkins
```

```
  metrics_path: /prometheus
```

```
  static_configs:
```

```
    - targets: ['192.168.1.244:8080']
```

```
EOF
```

Prometheus - Configuration



- Vérifier la configuration:

```
vagrant@localhost:/home/va... X + ~
[root@localhost vagrant]# docker exec cat /etc/prometheus/prometheus.yml
^C
[root@localhost vagrant]# docker exec prometheus cat /etc/prometheus/prometheus.yml
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - targets:
            # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label 'job=<job_name>' to any timeseries scraped from this config.
  - job_name: "prometheus"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9090"]
      - job_name: jenkins
        metrics_path: /prometheus
        static_configs:
          - targets: ['192.168.1.244:8080']
[root@localhost vagrant]#
```

Prometheus - Configuration

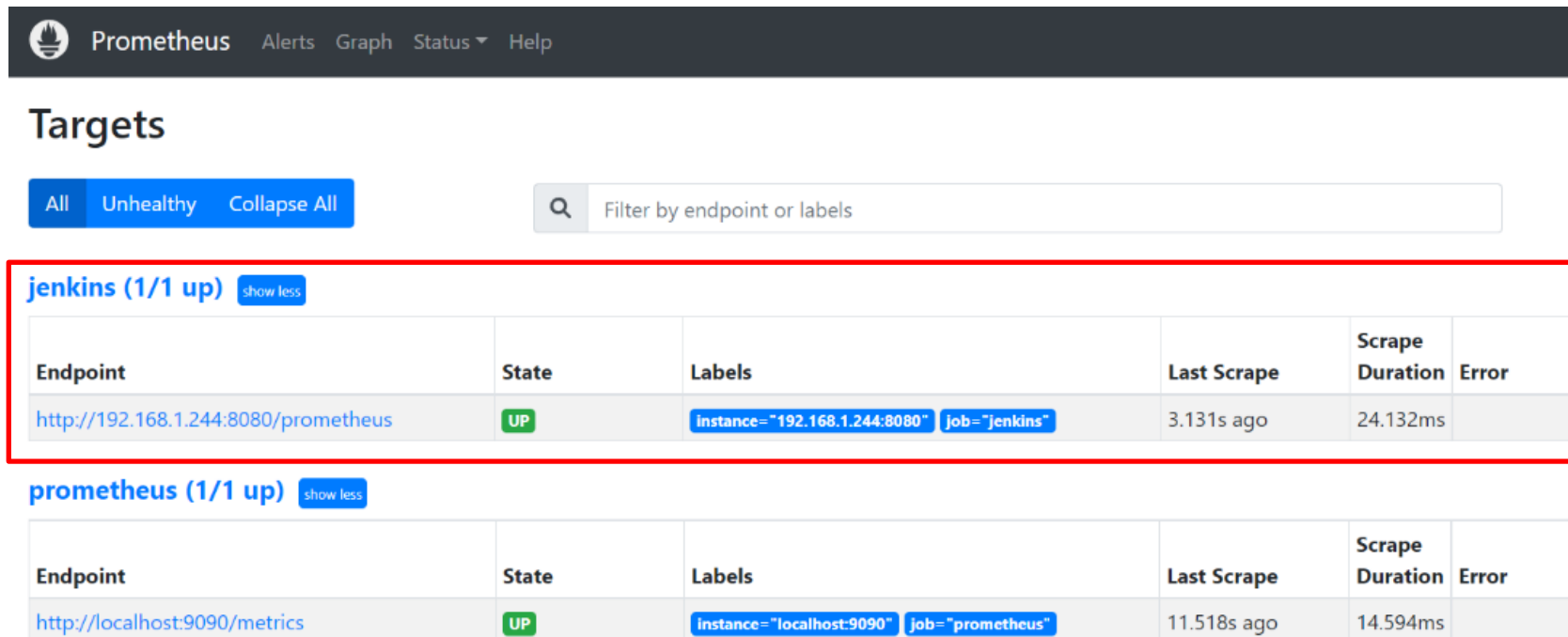


- Après la configuration, nous devons redémarrer le conteneur:

docker restart prometheus

```
[root@localhost vagrant]# docker restart prometheus  
prometheus
```

- Après le redémarrage:



The screenshot shows the Prometheus web interface with the 'Targets' tab selected. The 'jenkins' target is highlighted with a red box. The table below shows the status of the targets.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.1.244:8080/prometheus	UP	instance="192.168.1.244:8080" job="jenkins"	3.131s ago	24.132ms	
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	11.518s ago	14.594ms	

Prometheus - Configuration



- Pour visualiser les informations récupérées:

jenkins (1/1 up) [show less](#)

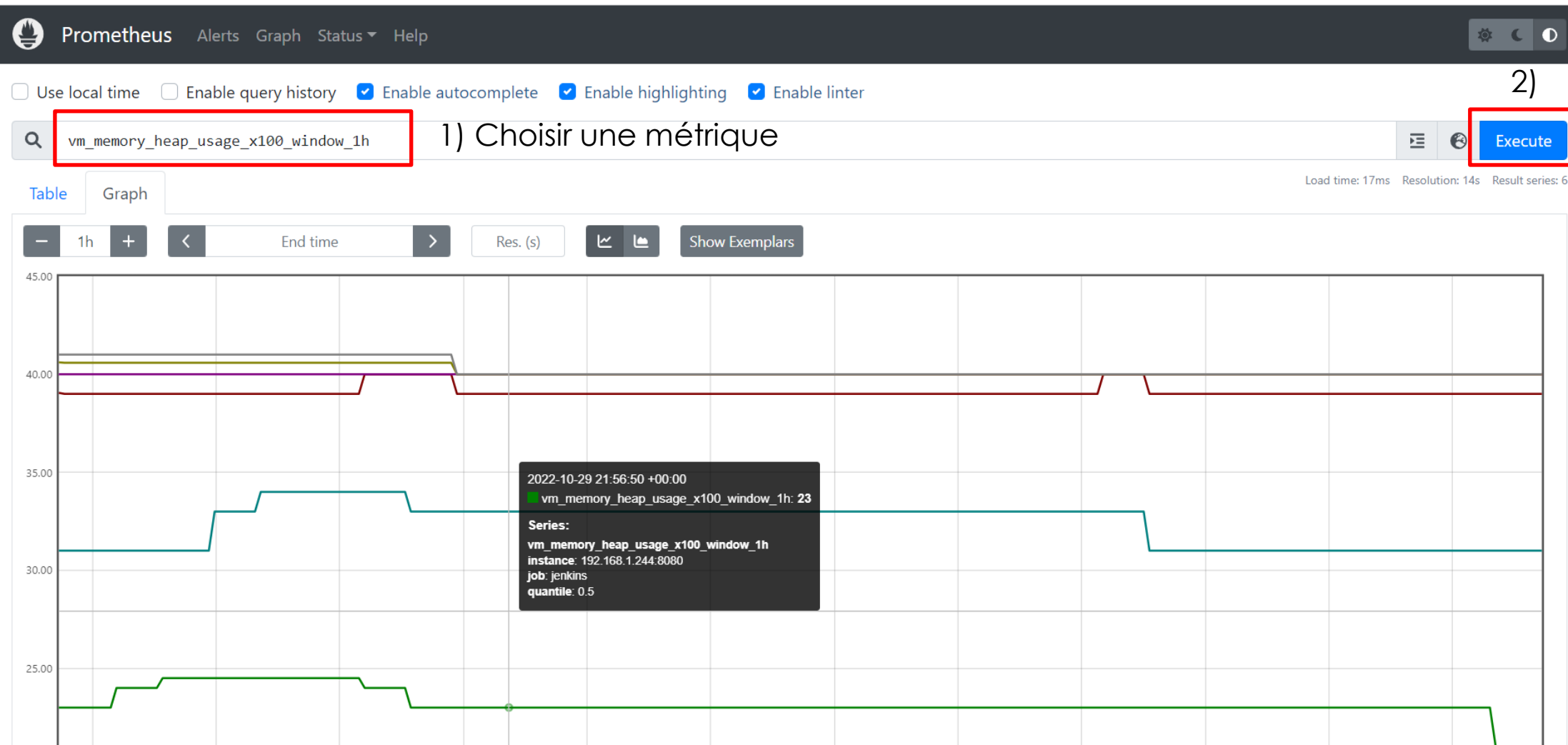
Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://192.168.1.244:8080/prometheus	UP	instance="192.168.1.244:8080" job="jenkins"	3.131s ago	24.132ms	



```
# HELP vm_memory_total_used_window_1h Generated from Dropwizard metric import (metric=vm.memory.total.used.window.1h, type=jenkins.metrics.util.AutoSamplingHistogram)
# TYPE vm_memory_total_used_window_1h summary
vm_memory_total_used_window_1h{quantile="0.5",} 7.24088384E8
vm_memory_total_used_window_1h{quantile="0.75",} 7.40475432E8
vm_memory_total_used_window_1h{quantile="0.95",} 7.569601536E8
vm_memory_total_used_window_1h{quantile="0.98",} 7.5711888E8
vm_memory_total_used_window_1h{quantile="0.99",} 7.5711888E8
vm_memory_total_used_window_1h{quantile="0.999",} 7.5711888E8
vm_memory_total_used_window_1h_count 22.0
# HELP jenkins_queue_blocked_history Generated from Dropwizard metric import (metric=jenkins.queue.blocked.history, type=jenkins.metrics.util.AutoSamplingHistogram)
# TYPE jenkins_queue_blocked_history summary
jenkins_queue_blocked_history{quantile="0.5",} 0.0
jenkins_queue_blocked_history{quantile="0.75",} 0.0
jenkins_queue_blocked_history{quantile="0.95",} 0.0
jenkins_queue_blocked_history{quantile="0.98",} 0.0
jenkins_queue_blocked_history{quantile="0.99",} 0.0
jenkins_queue_blocked_history{quantile="0.999",} 0.0
jenkins_queue_blocked_history_count 22.0
# HELP vm_memory_pools_Metaspace_used_window_1h Generated from Dropwizard metric import (metric=vm.memory.pools.Metaspace.used.window.1h, type=jenkins.metrics.util.AutoSamplingHistogram)
# TYPE vm_memory_pools_Metaspace_used_window_1h summary
```

Prometheus - Configuration

- Pour visualiser les valeurs récupérées d'une métrique:



Prometheus - Configuration



- Pour visualiser les valeurs récupérées d'une métrique:

☐ Enable query history

vm_terminated_count

Load time: 259ms
Resolution: 14s
Total time series: !

Execute

vm_terminated_count

Graph

Console

Element	Value
default_jenkins_builds_duration_milliseconds_summary_count{buildable="true",instance="192.168.40.253:8080",jenkins_job="Affichage Date (1)",job="jenkins",repo="NA"}	149
default_jenkins_builds_duration_milliseconds_summary_count{buildable="true",instance="192.168.40.253:8080",jenkins_job="Affichage date Distance (2)",job="jenkins",repo="NA"}	2
default_jenkins_builds_duration_milliseconds_summary_count{buildable="true",instance="192.168.40.253:8080",jenkins_job="part 1",job="jenkins",repo="NA"}	200
default_jenkins_builds_duration_milliseconds_summary_count{buildable="true",instance="192.168.40.253:8080",jenkins_job="part 2",job="jenkins",repo="NA"}	198
default_jenkins_builds_duration_milliseconds_summary_count{buildable="true",instance="192.168.40.253:8080",jenkins_job="part 3",job="jenkins",repo="NA"}	18661

Prometheus - Configuration

- Pour visualiser le contenu du fichier de configuration:

Status → Configuration

La configuration globale

```
global:
  scrape_interval: 15s
  scrape_timeout: 10s
  evaluation_interval: 15s
```

Intervalle de
récupération
des données

Timeout lors du
scrapping

```
alerting:
  alertmanagers:
  - follow_redirects: true
    enable_http2: true
    scheme: http
    timeout: 10s
    api_version: v2
    static_configs:
      - targets: []
```

La configuration du scrapping pour un serveur (Exemple: Jenkins)

```
- job_name: jenkins
  honor_timestamps: true
  scrape_interval: 15s
  scrape_timeout: 10s
  metrics_path: /prometheus
  scheme: http
  follow_redirects: true
  enable_http2: true
  static_configs:
    - targets:
      - 192.168.1.244:8080
```

Nom du bloc

Route de
scrapping

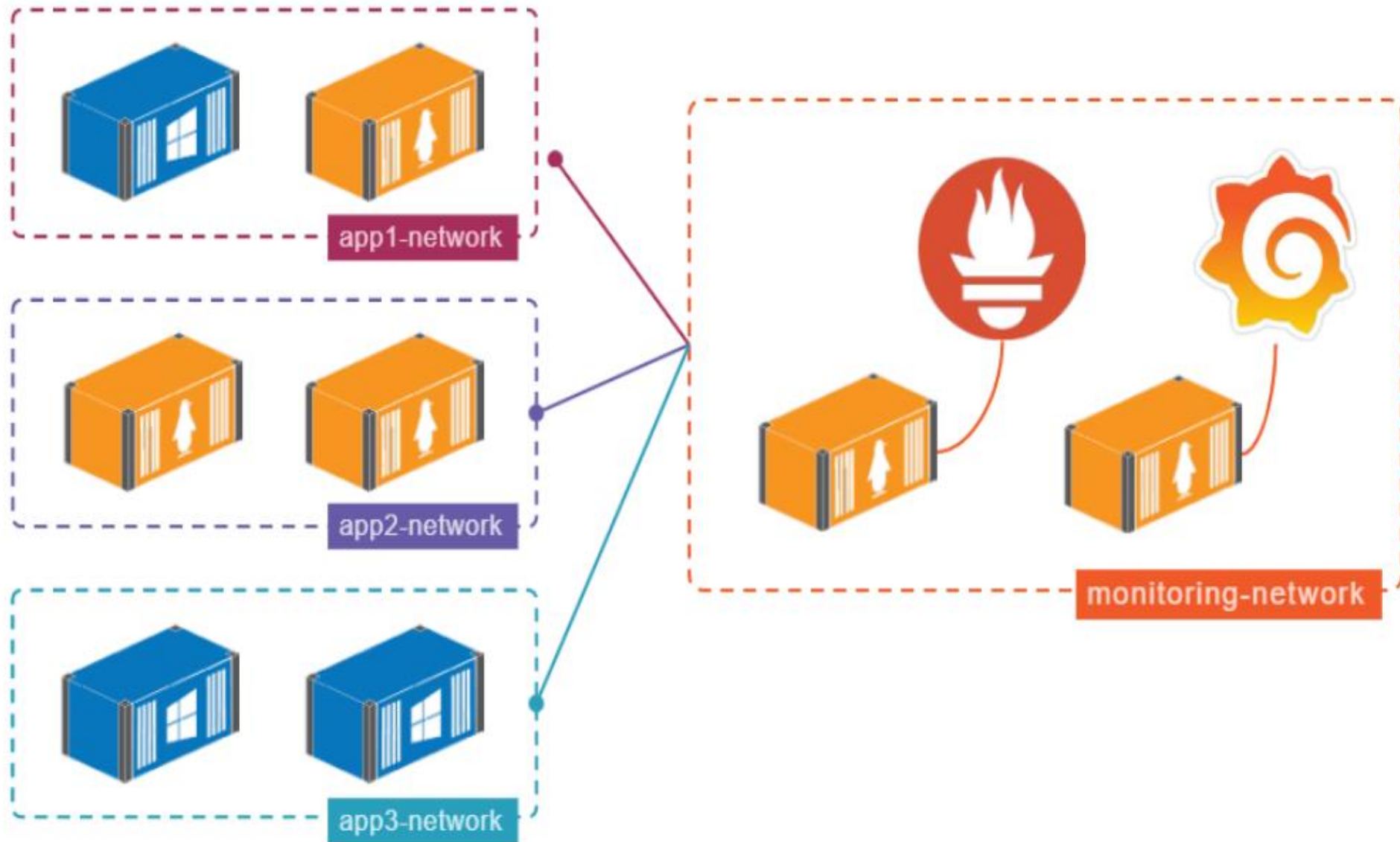
L'@ IP du serveur

Grafana - Définition



- Grafana est un logiciel **open source** de visualisation et d'analyse.
- Il permet de faire la visualisation à travers:
 - Les graphiques
 - Les tableaux
 - Les histogrammes
 - Les points
 - ...
- Il permet d'interroger, de visualiser, d'alerter et d'explorer des métriques, quel que soit l'endroit où elles sont stockées (Prometheus, influxdb, postgres, mysql, elastic search, ...)
- Il stocke ses données dans une base de données interne (SQL Lite)
- Cet outil est utilisé par des millions d'utilisateurs (Plus de 750k installation et 42k Github stars)

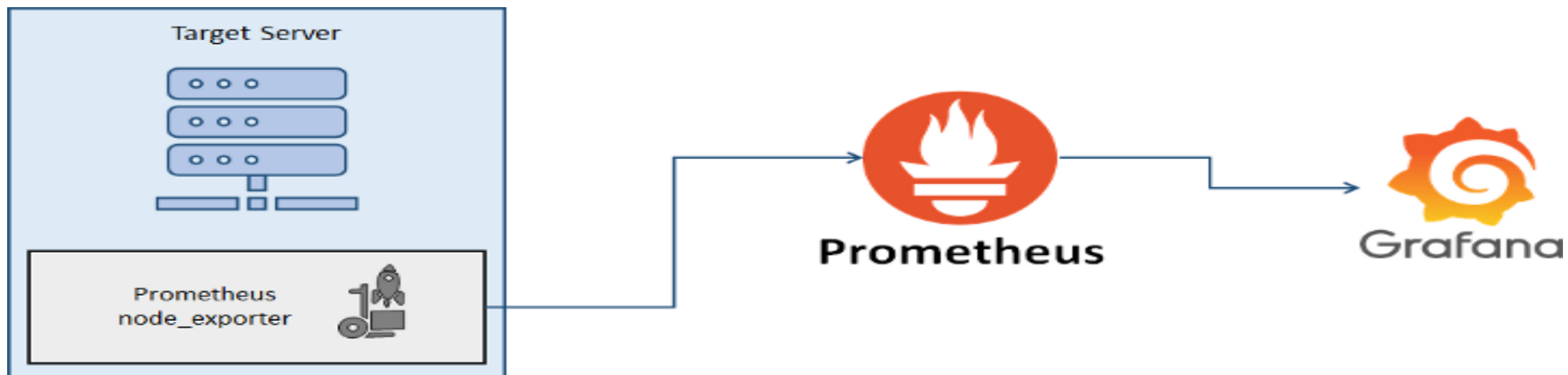
Grafana - Architecture



Architecture



- Node exporter est un exportateur Prometheus utilisé pour exposer les métriques des serveurs Linux.
- Grace au Node exporter, nous pouvons exposer diverses ressources du système telles que la RAM, l'utilisation du processeur, l'utilisation de la mémoire, l'espace disque, etc.
- Node exporter fonctionne comme un système qui rassemble les métriques de votre système et avec l'aide de Grafana, nous pouvons visualiser le métrique.



Grafana - Installation



- Vous allez utiliser une image Docker.

`docker pull grafana/grafana`

```
[root@localhost vagrant]# docker pull grafana/grafana
```

- Création d'un conteneur Docker Grafana.

`docker run -d --name grafana -p 3000:3000 grafana/grafana`

```
[root@localhost vagrant]# docker ps -a
```

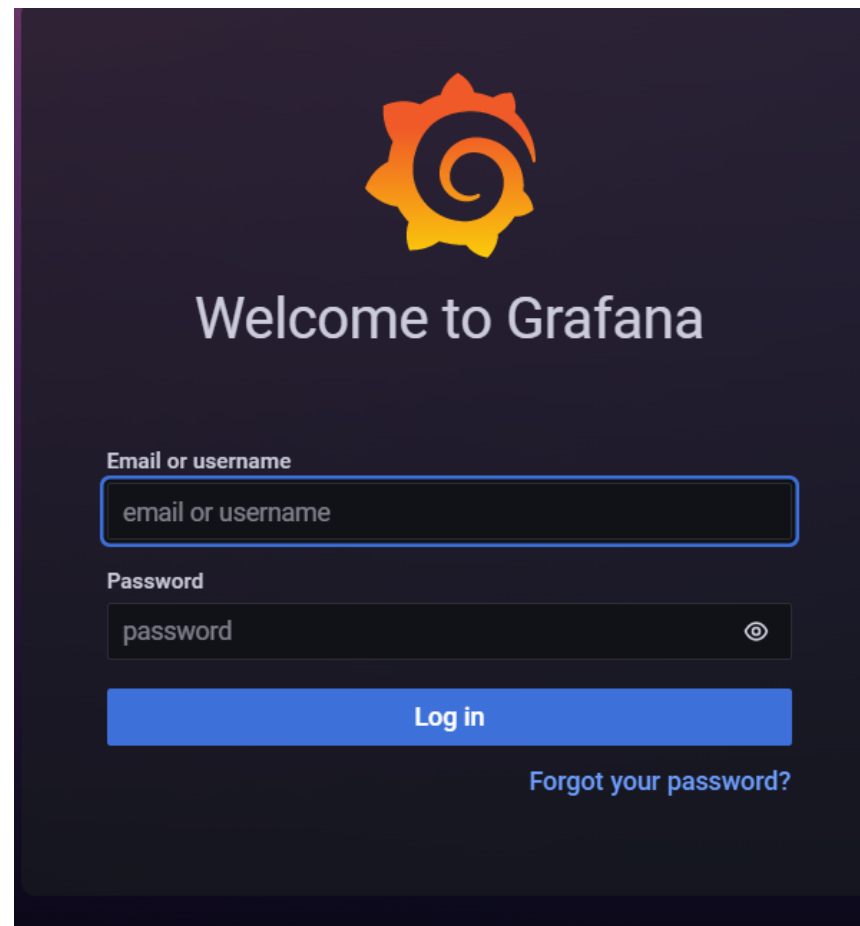
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
e72dcb971cec	grafana/grafana	"/run.sh"	15 seconds ago	Up 10 seconds	0.0.0.0:3000->3000/tcp	grafana

Grafana - Configuration




- Accéder à l'interface web de grafana : <http://@IP:3000>

Les paramètres d'accès par défaut: admin/admin



- Changer le mot de passe:



Welcome to Grafana

New password

Confirm new password

Submit

[Skip](#)

Grafana - Configuration



← → ↺ Non sécurisé | 192.168.1.244:3000/?orgId=1

Firewall Authenticat... f @ in y M 28 DevOps Tools SSD DevOps 2022-2023... Spring 2022-2023 Spring DevOps et CI Dash prof »

General / Home

Welcome to Grafana


Need help? [Documentation](#) [Tutorials](#) [Community](#) [Public Slack](#)

Basic


The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.




DATA SOURCES
Add your first data source



Learn how in the docs [↗](#)

DASHBOARDS
Create your first dashboard



Learn how in the docs [↗](#)

[Remove this panel](#)

Dashboards

Starred dashboards


Recently viewed dashboards

Latest from the blog

oct. 27

Watch this: An inside look at GrafanaLive

At Grafana Labs, we love open source, which is another way to say we love being part of a community. That's why we were so excited and grateful to finally get back together face-to-face this year with GrafanaLive. There's supposed to be a video here, but for some reason there isn't. Either we entered the id wrong



SF Bay area Washington, D.C. London Chicago

Grafana - Configuration



- **Première chose à faire:** Ajouter une source de données

← → ↻ Non sécurisé | 192.168.1.244:3000/?orgId=1

Firewall Authentica... f i y M 28 DevOps Tools SSD DevOps 2022-2023... Spring 2022-2023 Spring DevOps et CI Dash prof

General / Home

Welcome to Grafana

Need help? [Documentation](#) [Tutorials](#) [Community](#) [Public Slack](#)


Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL DATA SOURCE AND DASHBOARDS


Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.



DATA SOURCES


Add your first data source



[Learn how in the docs](#)

DASHBOARDS

Create your first dashboard



[Learn how in the docs](#)

[Remove this panel](#)

Dashboards

Starred dashboards


Recently viewed dashboards

Latest from the blog

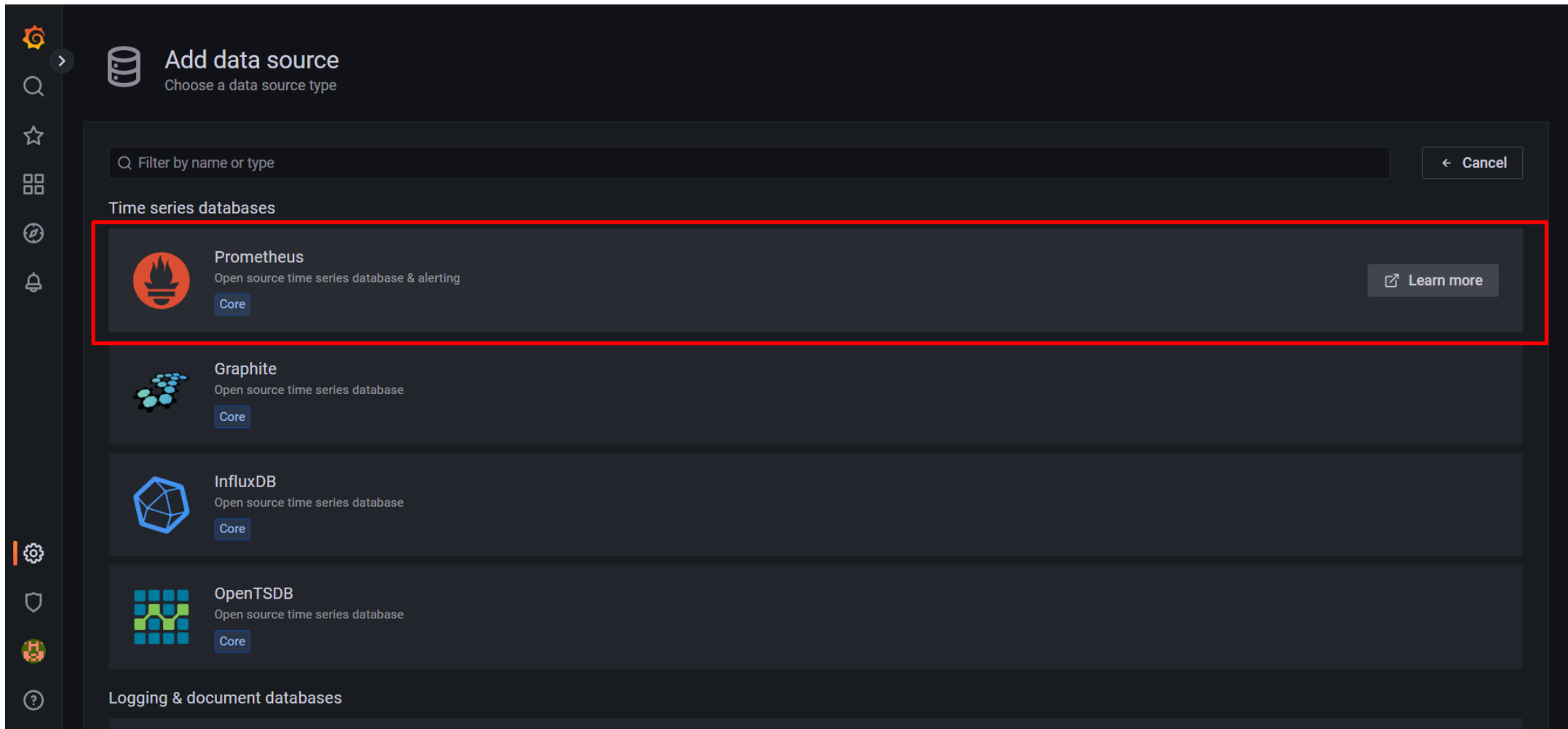
oct. 27

Watch this: An inside look at GrafanaLive

At Grafana Labs, we love open source, which is another way to say we love being part of a community. That's why we were so excited and grateful to finally get back together face-to-face this year with GrafanaLive. There's supposed to be a video here, but for some reason there isn't. Either we entered the id wrong



Grafana - Configuration



Grafana - Configuration



Data Sources / Prometheus
Type: Prometheus

Settings Dashboards

Alerting supported

Name Default ☒

HTTP

URL

Allowed cookies

Timeout

Auth

Basic auth ☐ With Credentials ☐

TLS Client Auth ☐ With CA Cert ☐

Skip TLS Verify ☐

Forward OAuth Identity ☐

Custom HTTP Headers

Custom HTTP Headers

+ Add header

Alerting

Manage alerts via Alerting UI ☒

Alertmanager data source

Scrape interval

Query timeout

HTTP Method

Misc

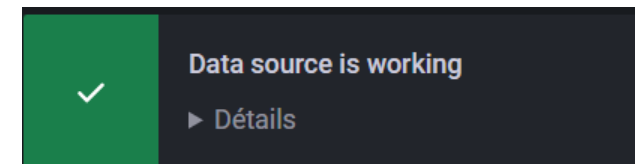
Disable metrics lookup ☐

Custom query parameters

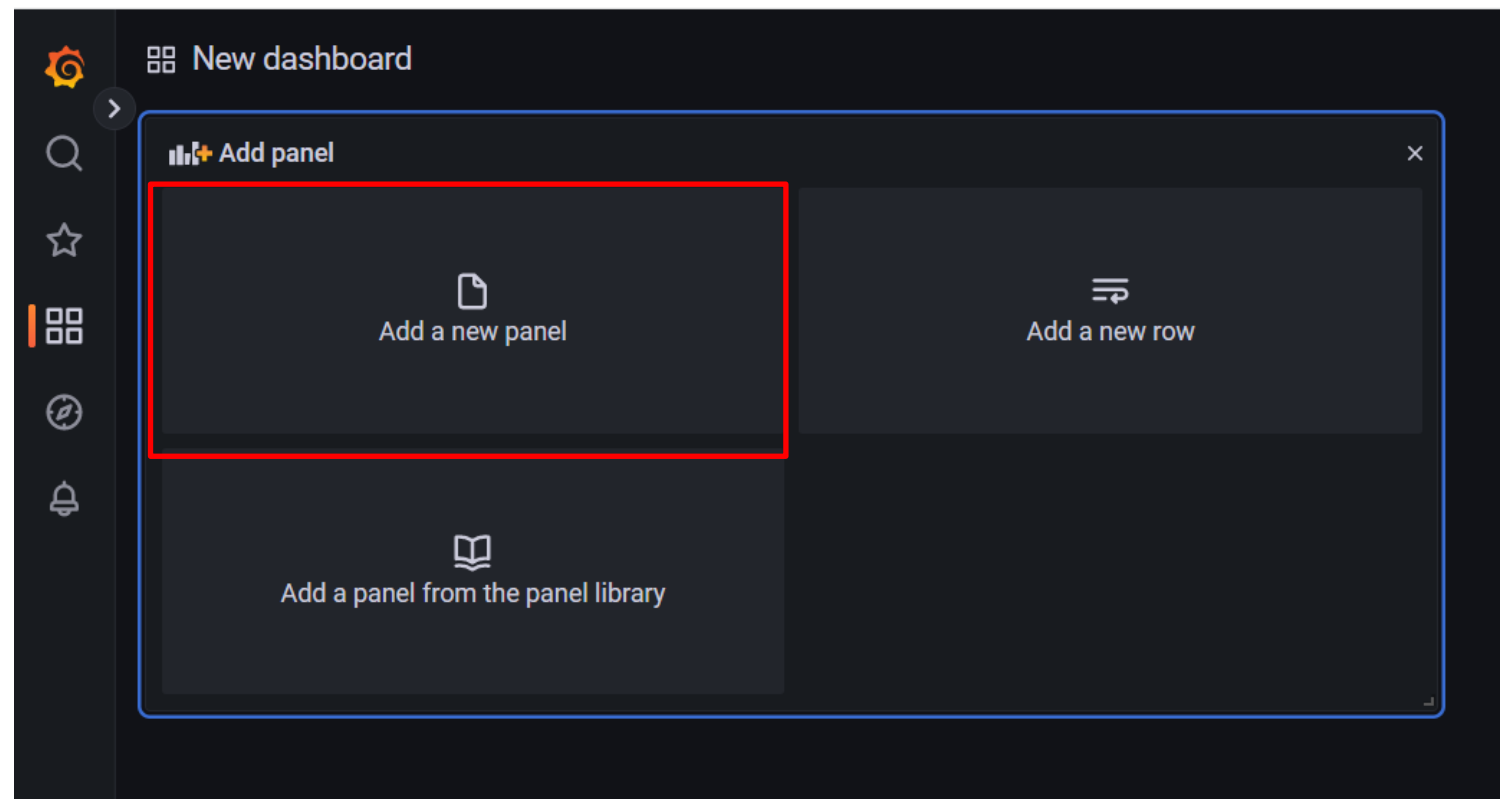
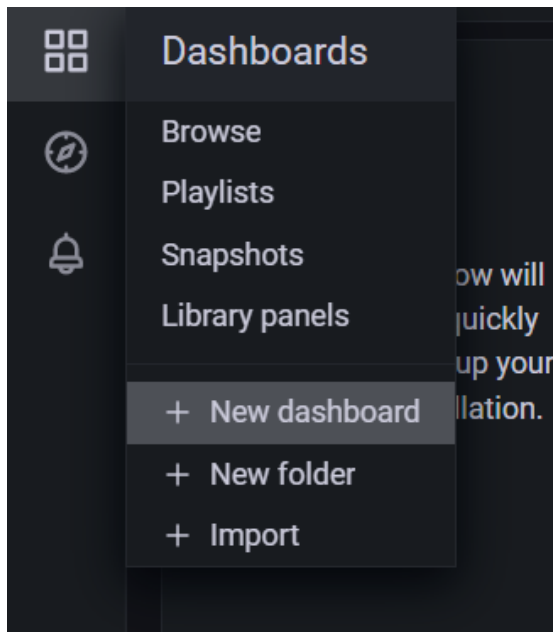
Exemplars

+ Add

Back Explore Delete **Save & test**



- **Deuxième chose à faire:** Représenter les données
 - ✓ 1ère manière: Créer un dashboard



Grafana - Configuration



✓ 1^{ère} manière: Créer un dashboard

4- Sauvgarder

New dashboard / Edit Panel

1- Choisir le type du panel

Time series

Panel Title

500
400
300
200
100
0

19:00 19:15 19:30 19:45 20:00 20:15 20:30 20:45 21:00 21:15 21:30 21:45 22:00 22:15 22:30 22:45 23:00 23:15 23:30 23:45 00:00 00:15 00:30 00:45

Query 1 Transform 0 Alert 0

Data source Jenkins Query options MD = auto = 1616 Interval = 15s

3- Visualiser

Query inspector

2- Définir la métrique

Metric Label filters

default_jenkins_builds_success_build_count_total Select label = Select value x +

+ Operations hint: add rate()

Raw query

default_jenkins_builds_success_build_count_total

Options Legend: Auto Format: Time series Step: auto Type: Range Exemplars: false

Panel options

Title

Panel Title

Description

Transparent background

Panel links

Repeat options

Tooltip

Tooltip mode

Single All Hidden

Legend

Visibility

Mode

List Table

Placement

Bottom Right

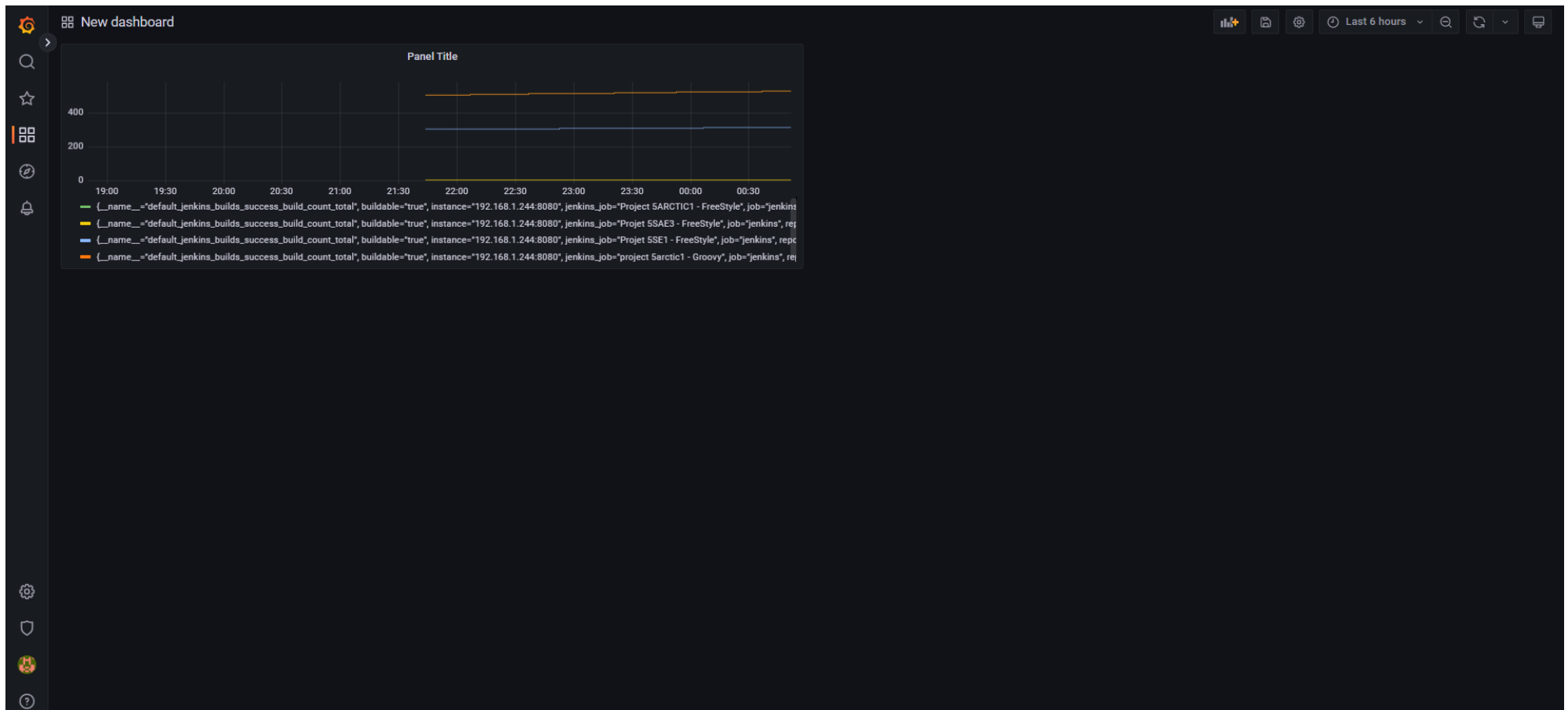
Values

Select values or calculations to show in legend

Grafana - Configuration



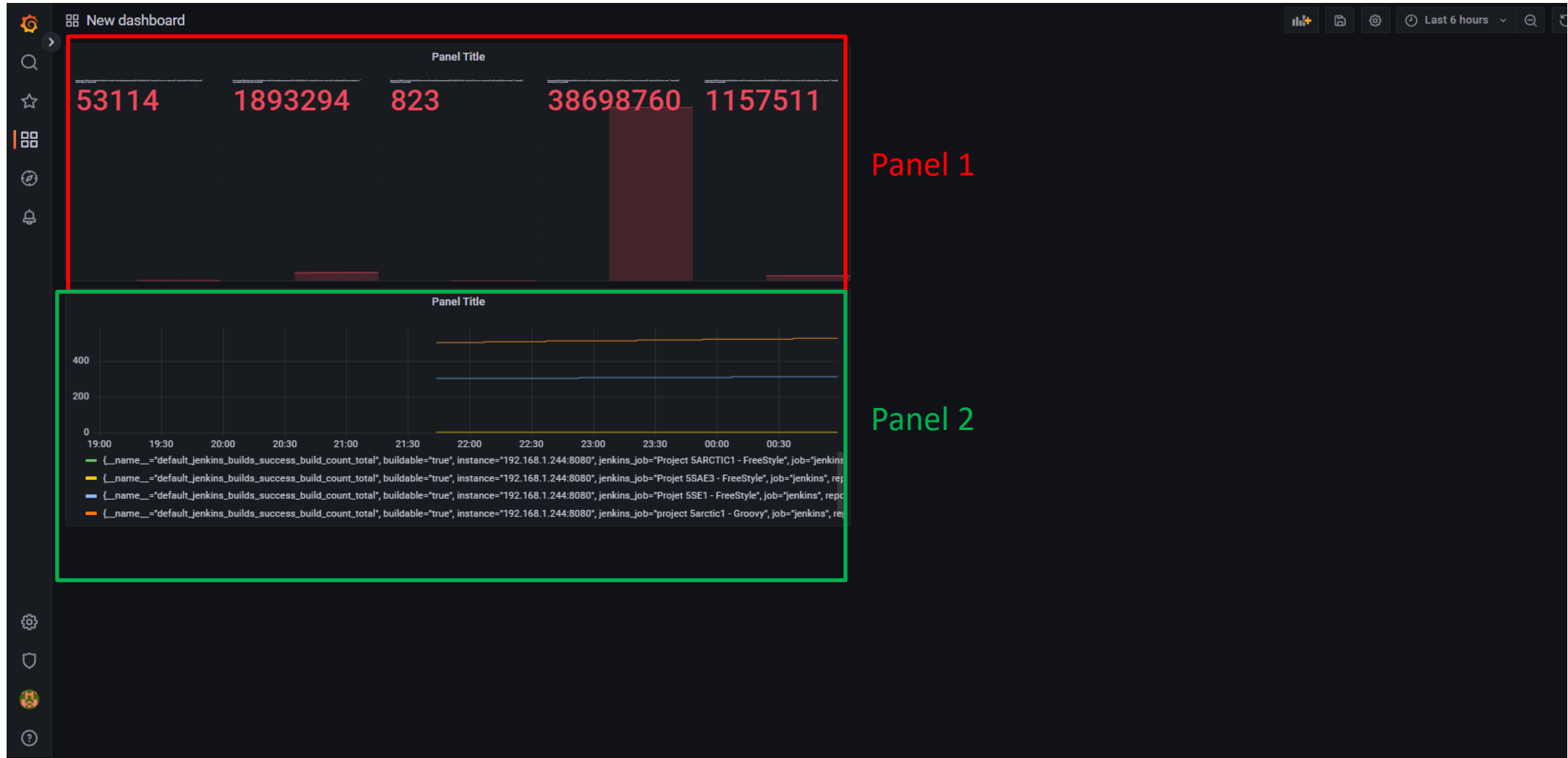
✓ 1ère manière: Créer un dashboard



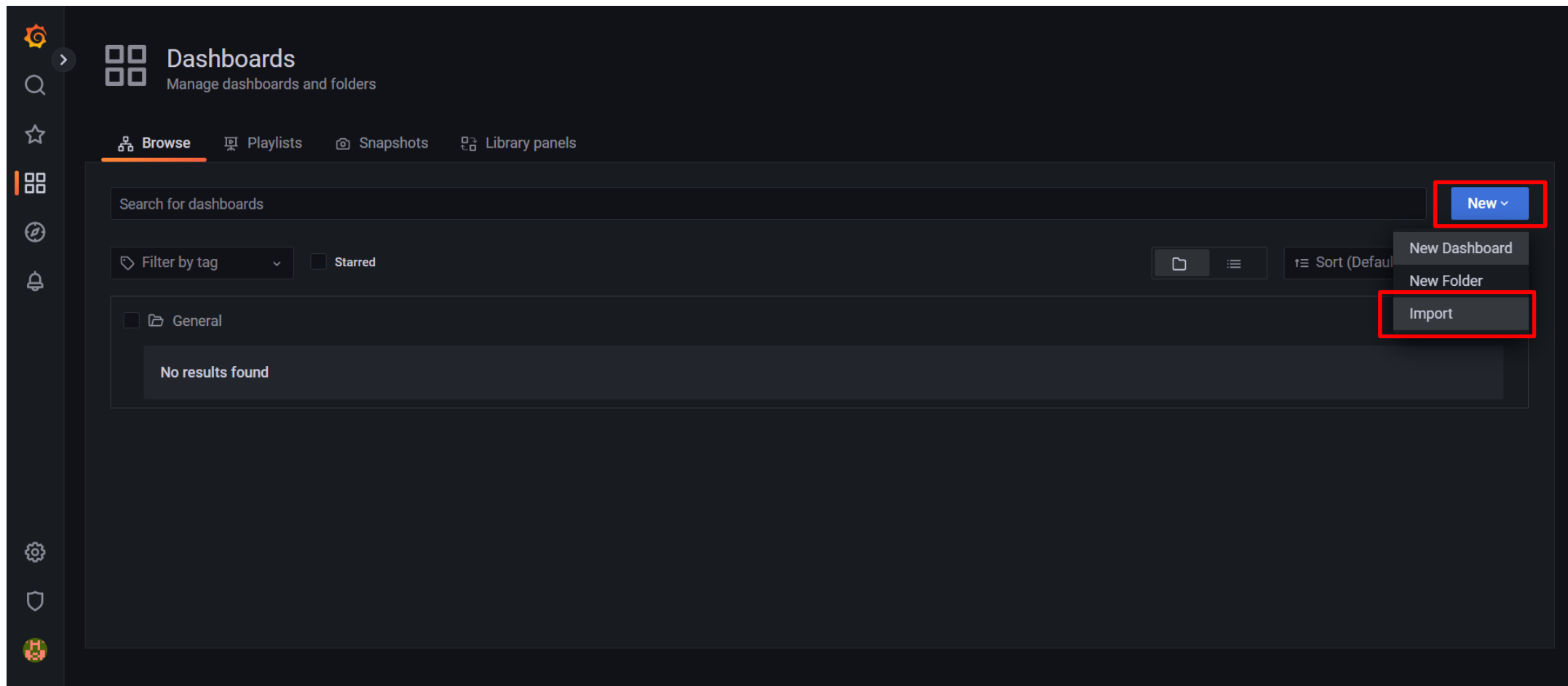
Grafana - Configuration



✓ 1ère manière: Créer un dashboard

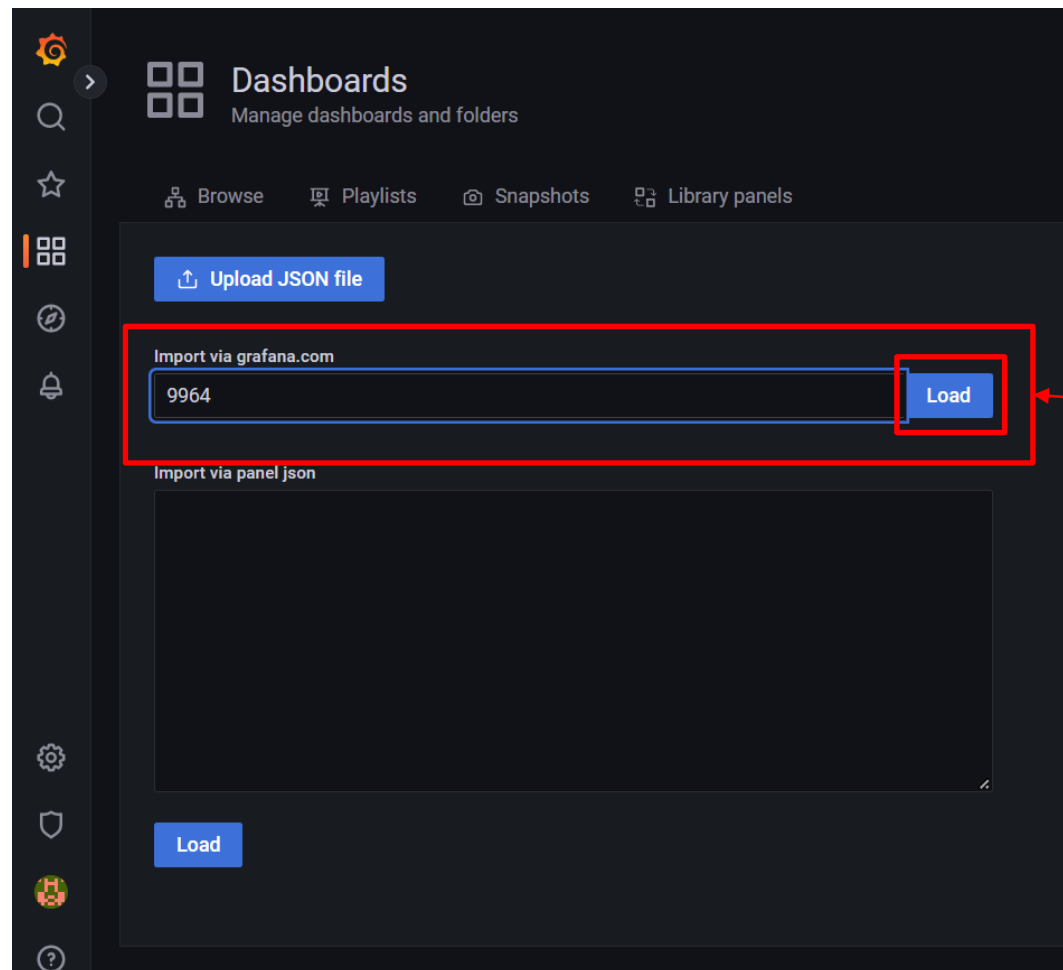


- **Deuxième chose à faire:** Représenter les données
 - ✓ 2ème manière: Importer un dashboard



✓ 2ème manière: Importer un dashboard

Choisir un template: <https://grafana.com/grafana/dashboards/>



Saisir l'identifiant du template

Grafana - Configuration



✓ 2ème manière: Importer un dashboard

Importing dashboard from [Grafana.com](#)

Published by haryan

Updated on 2019-03-24 17:58:20

Options

Name
Jenkins: Performance and Health Overview

Folder
General

Unique identifier (UID)
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URLs for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.
haryan-jenkins [Change uid](#)

Prometheus
Prometheus Data Source
Jenkins (default)

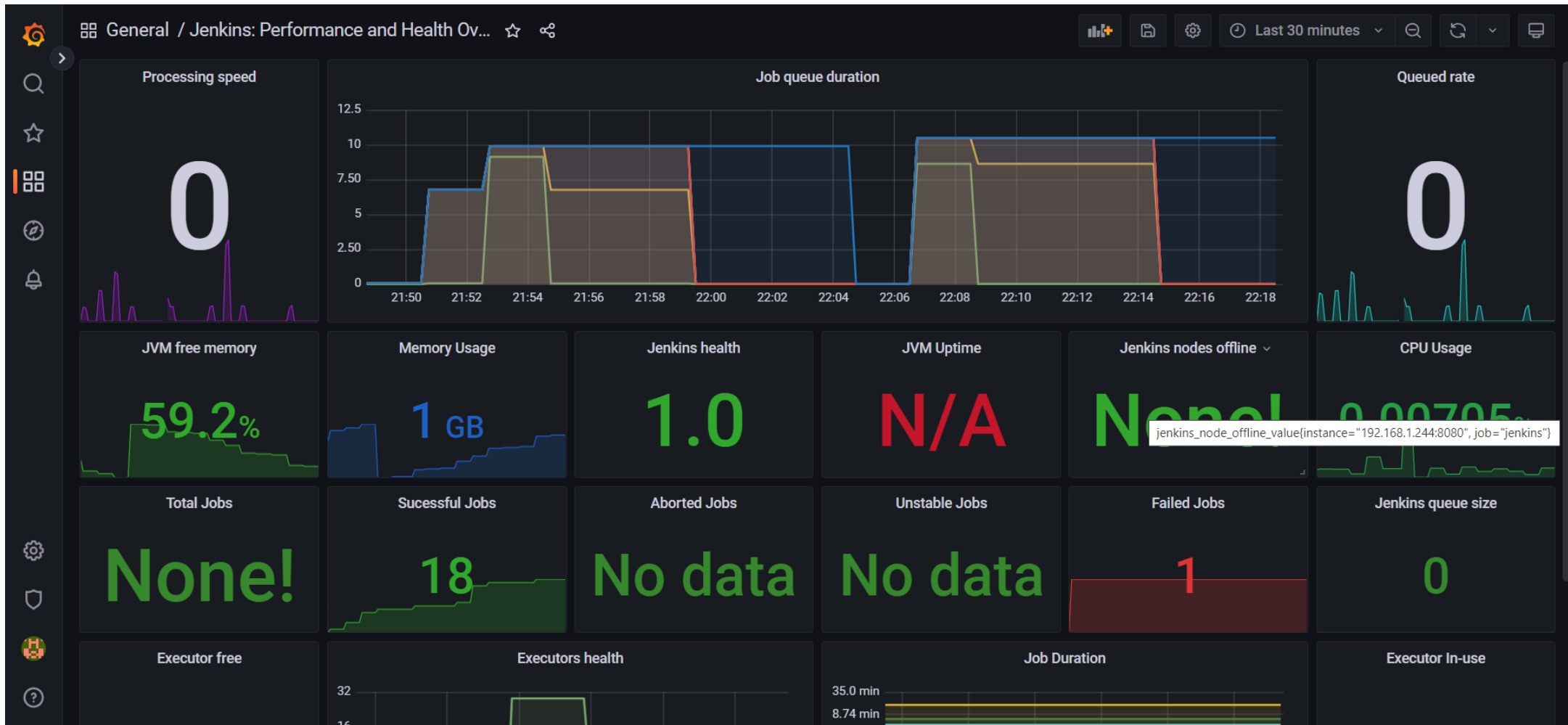
[Import](#) [Cancel](#)

Choisir la source des données

Grafana - Configuration



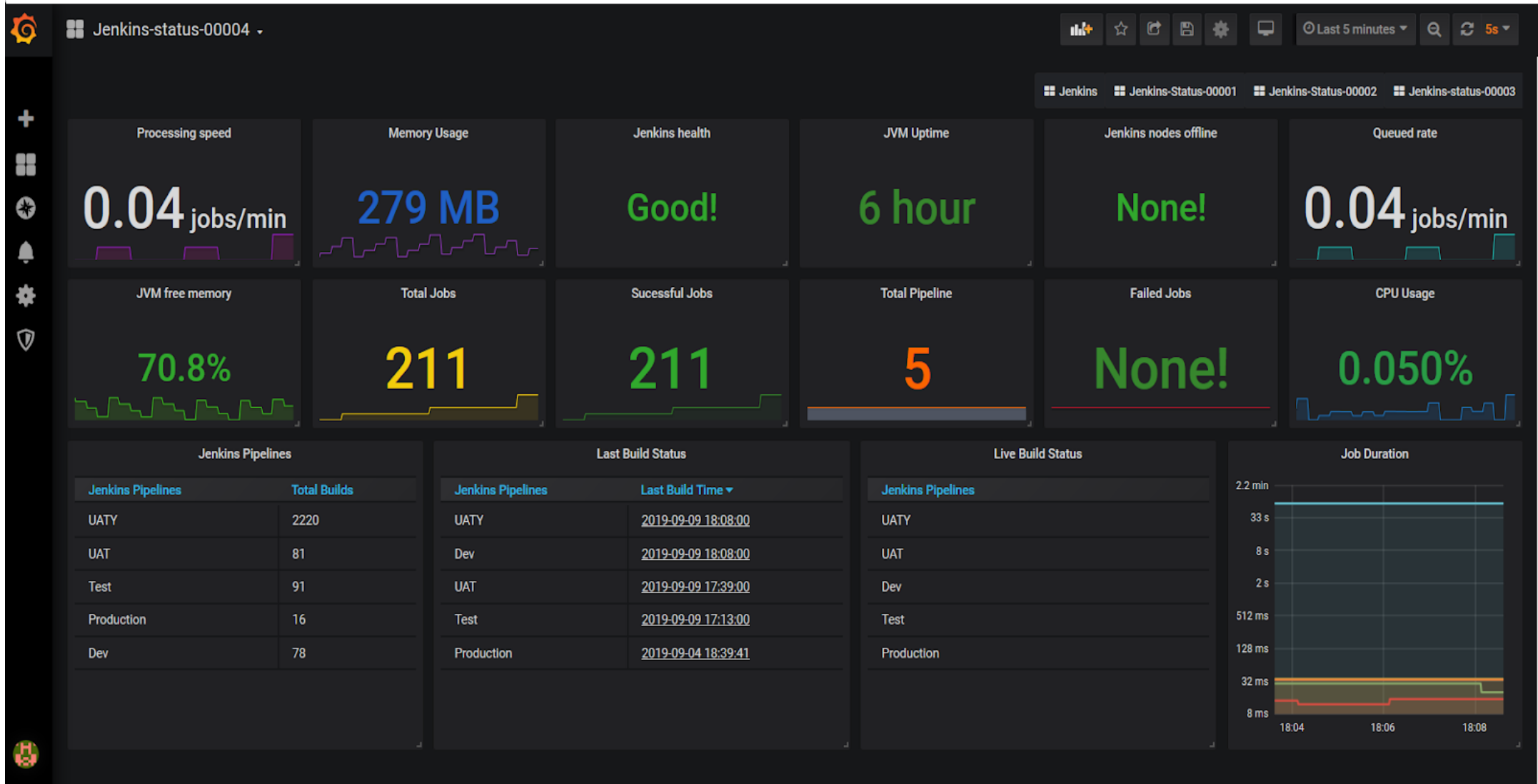
✓ 2ème manière: Importer un dashboard



Grafana - Configuration



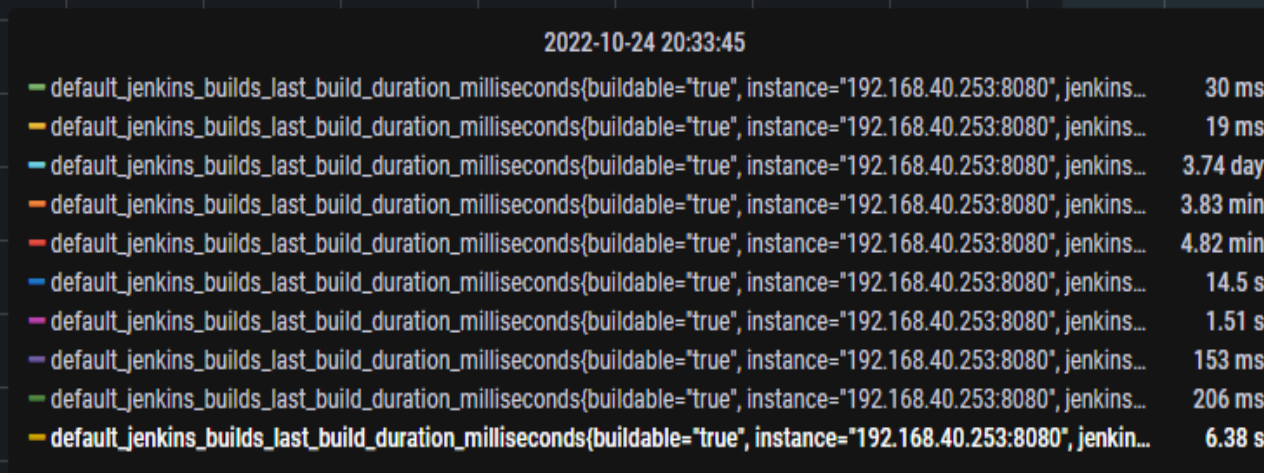
✓ 2^{ème} manière: Chaque dashboard affiche des métriques bien précises



Grafana - Configuration



Job Duration ▾



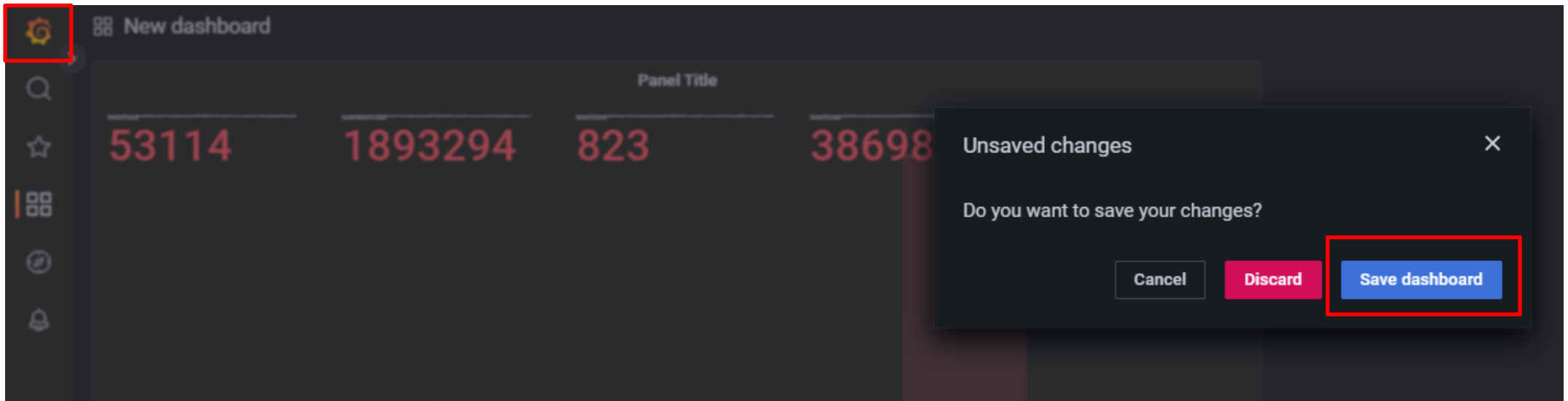
Active Windows

Accédez aux paramètres pour activer Windows

Grafana - Configuration

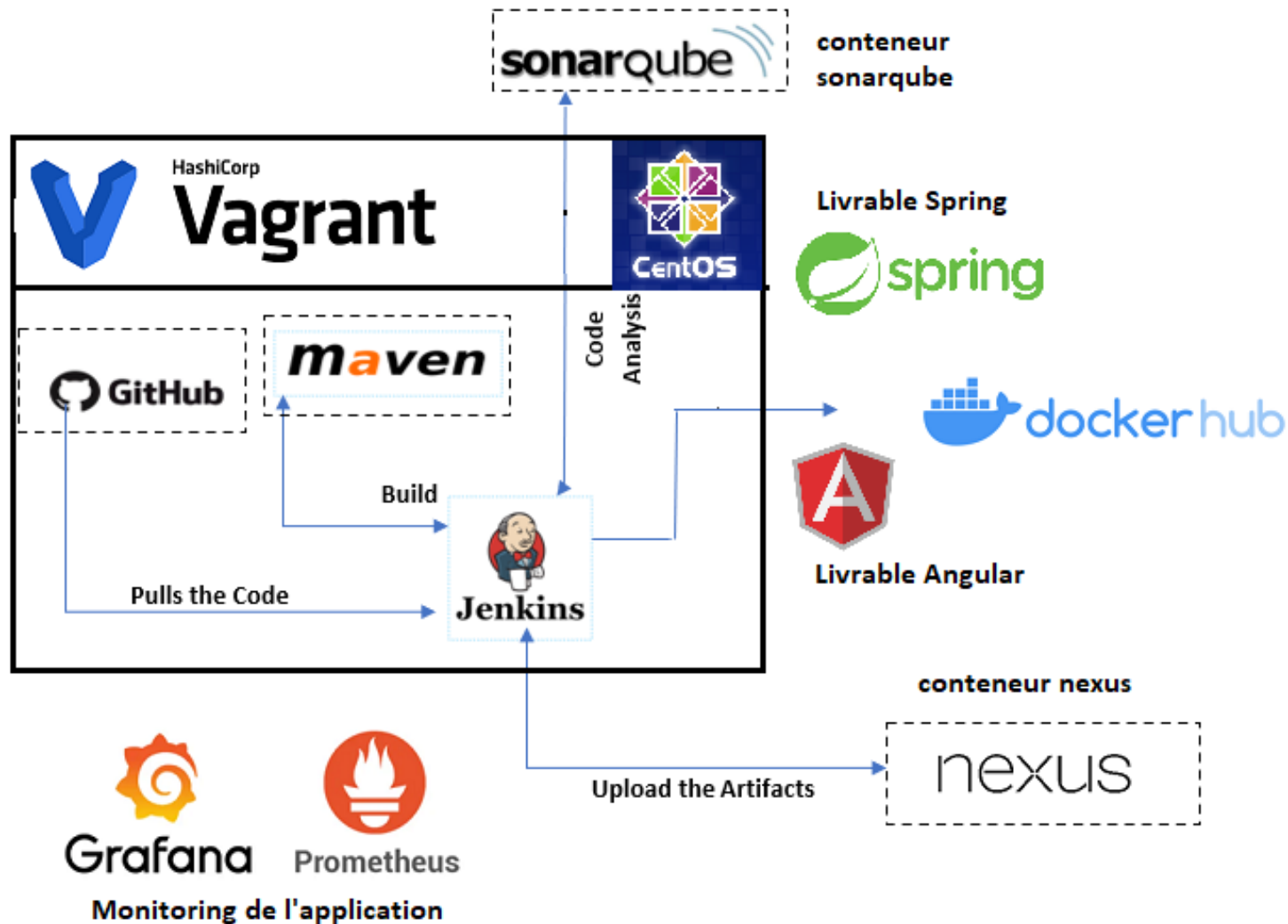


Pour sauvgarder le dashboard



Travail à faire

- Créer les dashboards pour superviser les différents serveurs.



Monitoring: Prometheus et Grafana

Si vous avez des questions, n'hésitez pas à nous contacter :

Département Informatique
UP Architectures des Systèmes d'Information

Bureau E204

Monitoring: Prometheus et Grafana



**UP ASI
Bureau E204**