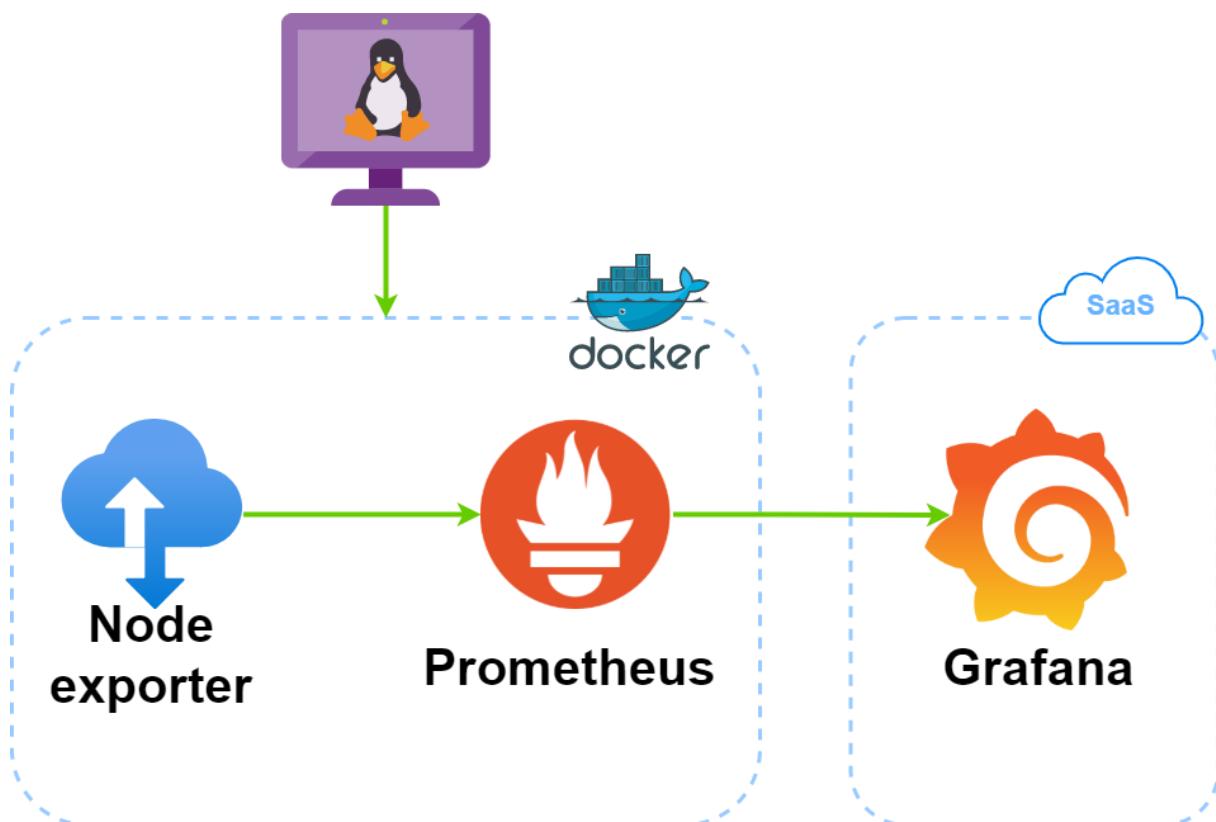


Prometheus + Node Exporter + Grafana

GUIDE COMPLET ET PROFESSIONNEL

CONTENEURS, MONITORING, LOGS, LINUX ET SECURITE



Ecrit par : BIENVENU BRANDON

Futur Administrateur Systèmes et Réseaux

Table des matières

| | |
|---|-----------|
| 1. Introduction..... | 3 |
| 2. Prérequis..... | 3 |
| 3. Création de la machine virtuelle Ubuntu..... | 5 |
| 4 — Configuration réseau et préparation de l'environnement..... | 8 |
| 4.1 — Accéder à la VM Ubuntu Server dans VirtualBox..... | 8 |
| 4.2 — Vérification du réseau de la VM | 8 |
| 4.3 — Configuration VirtualBox pour accéder à Prometheus / Node Exporter / Grafana | 8 |
| Règles que tu as configurées : | 8 |
| 4.4 — Vérification de la connectivité..... | 9 |
| 4.5 — Téléchargement des composants..... | 9 |
| Prometheus : | 9 |
| Node Exporter :..... | 10 |
| 4.6 — Lancement des services | 10 |
| Prometheus : | 11 |
| Node Exporter :..... | 11 |
| Grafana : | 11 |
| 6.7 — Accéder à Grafana | 11 |
| 7 — Grafana, Node Explorer, Prometheus | 12 |
| 7.1 — Accès à l'interface..... | 12 |
| 7.2 — Configuration de la source de données | 12 |
| 7.3 — Prometheus | 12 |
| • Exemple de requête : node_cpu_seconds_total | 12 |
| 7.4 — Node Exporter..... | 13 |
| • Service systemd actif → métriques accessibles sur Prometheus | 13 |
| • Exemple : surveillance mémoire, CPU, disque | 13 |
| 7.5 — Grafana..... | 13 |
| • Dashboard Node Exporter configuré → visualisation en temps réel..... | 13 |
| • Exemple d'utilisation : détecter pics CPU/mémoire, vérifier ressources | 13 |
| 7.6 — Ce qui n'a pas été fait | 13 |
| • Pas d'alerting Prometheus configuré | 13 |

- Pas de métriques Grafana exportées vers Prometheus13
- Pas de multi-node13
- Pas de création d'utilisateurs spécifiques pour les services13

Documentation : Mise en place d'une stack de monitoring

Prometheus + Node Exporter + Grafana

1. Introduction

Cette documentation décrit la mise en place complète d'une stack de monitoring utilisant **Prometheus**, **Node Exporter** et **Grafana** dans un environnement virtualisé avec Ubuntu Server. L'objectif est de démontrer :

- L'installation d'un serveur Linux dans VirtualBox
- La configuration de Prometheus pour scrapper des métriques système
- L'installation de Node Exporter pour exposer les métriques du système
- La connexion de Grafana à Prometheus pour visualiser les métriques via des dashboards

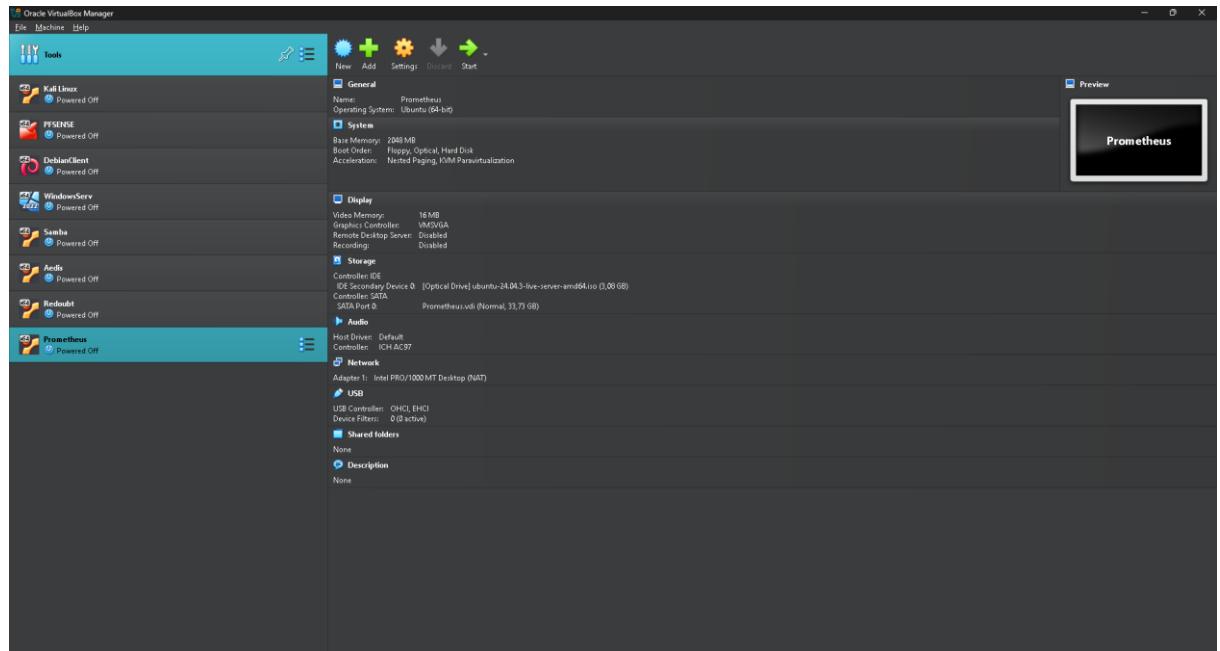
Cette documentation est pensée pour un projet LinkedIn, avec toutes les étapes reproduites et expliquées.

2. Prérequis

Avant de commencer, vous aurez besoin de :

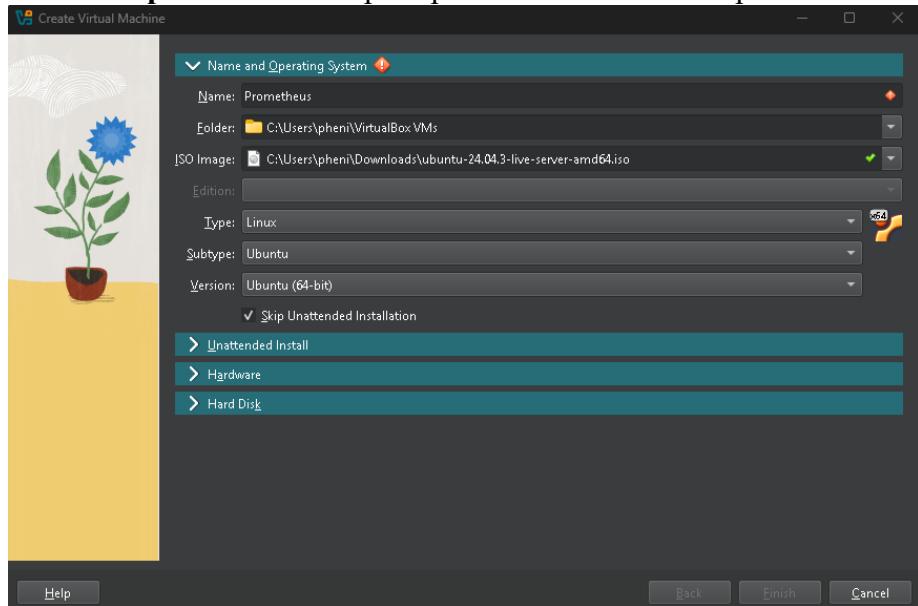
- VirtualBox installé sur votre machine hôte
- Ubuntu Server 22.04 LTS ou supérieur (VM)
- Accès à internet pour télécharger **Prometheus**, **Node Exporter** et **Grafana**
- Connaissances de base en terminal Linux

3. Création de la machine virtuelle Ubuntu

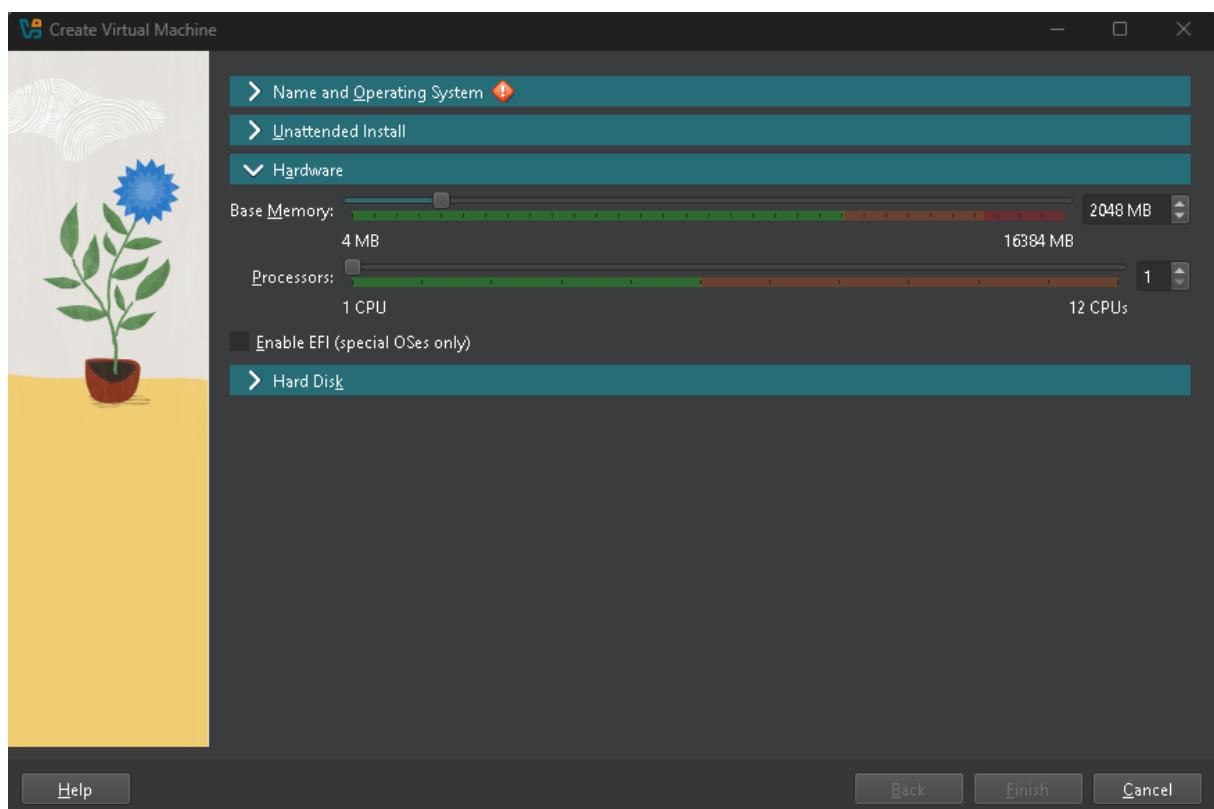
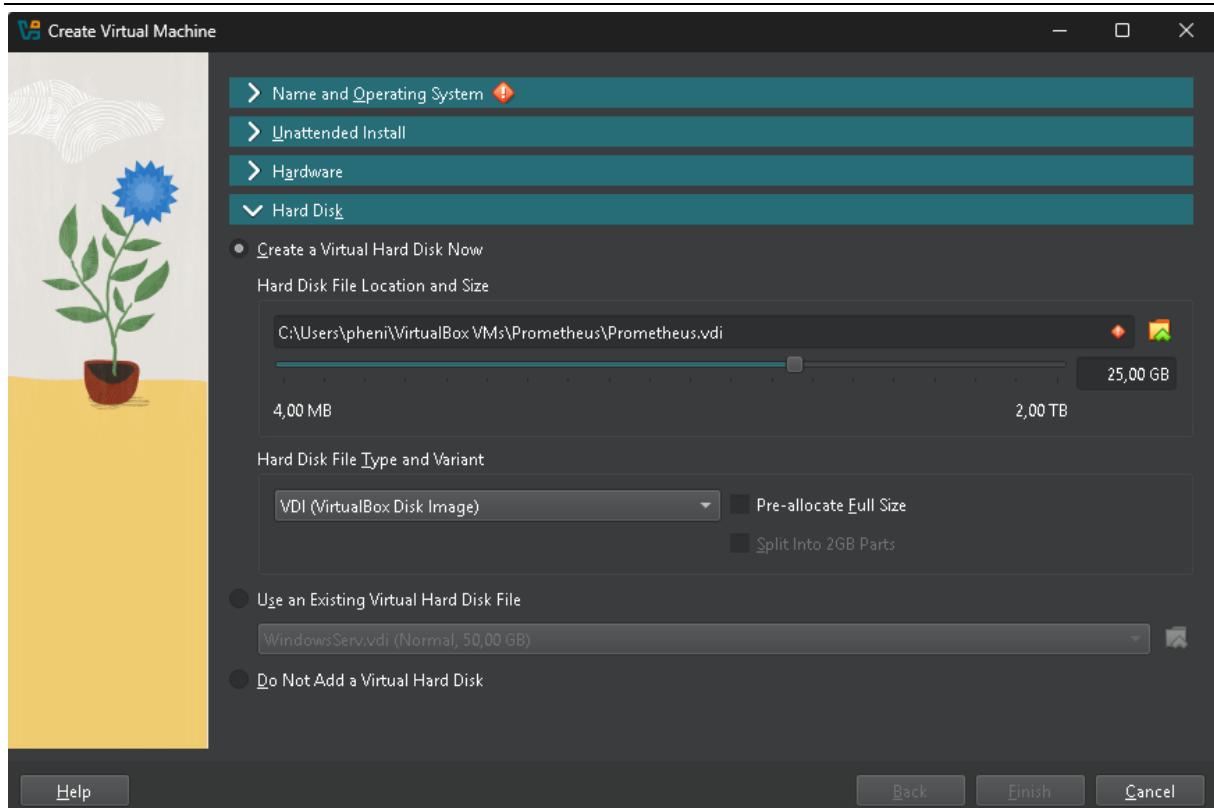


1. Ouvrir VirtualBox → Nouvelle VM
2. Paramètres recommandés :
 - o Nom : Prometheus
 - o Type : Linux
 - o Version : Ubuntu (64-bit)
 - o RAM : ≥ 2GB
 - o Disque dur : ≥ 20GB, type VDI dynamique
3. Réseau :
 - o Adaptateur 1 → NAT (accès internet)
 - o Adaptateur 2 → Réseau interne/Host-Only (pour accéder aux services depuis la machine hôte)
4. Démarrer la VM et installer **Ubuntu Server** avec les paramètres par défaut.

5. Installer **OpenSSH Server** pour pouvoir se connecter depuis l'hôte si besoin.



Brandon Bienvenu



4 — Configuration réseau et préparation de l'environnement

4.1 — Accéder à la VM Ubuntu Server dans VirtualBox

Tu as installé **Ubuntu Server** dans VirtualBox et tu t'es connecté avec l'utilisateur créé pendant l'installation (par l'installateur Ubuntu).

Pas d'étape supplémentaire ici.

4.2 — Vérification du réseau de la VM

Tu as vérifié que la VM avait une connexion réseau en utilisant :

```
ip a
```

Tu as identifié l'interface (enp0s3 / enp0s8 selon ton setup) et noté l'adresse IP locale.

4.3 — Configuration VirtualBox pour accéder à Prometheus / Node Exporter / Grafana

Pour accéder aux services depuis ta machine hôte, tu as utilisé **NAT + Port Forwarding** dans VirtualBox.

Règles que tu as configurées :

| Service | Port VM | Port Hôte |
|---------------|---------|-----------|
| Prometheus | 9090 | 9090 |
| Node Exporter | 9100 | 9100 |
| Grafana | 3000 | 3000 |

Tu les as ajoutées dans :

Paramètres → Réseau → Redirection de ports

| Nom | Protocole | IP hôte | Port hôte | IP invité | Port invité | |
|--------|-----------|---------|-----------|-----------|-------------|--|
| Rule 1 | TCP | | 9090 | | 9090 | |
| Rule 2 | TCP | | 9100 | | 9100 | |
| Rule 3 | TCP | | 3000 | | 3000 | |

OK Annuler

4.4 — Vérification de la connectivité

✓ La VM peut accéder à Internet

```
ping -c 3 google.com
```

✓ La VM a une adresse IP fonctionnelle

```
ip a
```

4.5 — Téléchargement des composants

Tu as téléchargé et extrait uniquement :

Prometheus :

```
wget https://github.com/prometheus/prometheus/releases/download/v3.7.3/prometheus-3.7.3.linux-amd64.tar.gz
```

```
tar -xvzf prometheus-3.7.3.linux-amd64.tar.gz
```

Node Exporter :

```
https://github.com/prometheus/node\_exporter/releases/download/v1.6.1/node\_exporter-1.6.1.linux-amd64.tar.gz\*
```

```
tar -xvzf node_exporter-1.6.1.linux-amd64.tar.gz
```

Grafana :

```
sudo apt-get install -y adduser libfontconfig1 musl

wget https://dl.grafana.com/grafana-enterprise/release/12.2.1/grafana-enterprise_12.2.1_18655849634_linux_amd64.deb

sudo dpkg -i grafana-enterprise_12.2.1_18655849634_linux_amd64.deb

sudo apt update

sudo apt install grafana -y
```

Aller dans le dossier où Prometheus a été extrait :

```
cd ~/prometheus-3.7.3.linux-amd64
```

2. Vérifier le fichier de configuration prometheus.yml

```
• global:
  •   scrape_interval: 15s
  •   evaluation_interval: 15s
  •
  • scrape_configs:
    • - job_name: "prometheus"
      static_configs:
        • - targets: ["localhost:9090"]
    •
    • - job_name: "node_exporter"
      static_configs:
        • - targets: ["localhost:9100"]
    •
    • - job_name: "grafana"
      static_configs:
        • - targets: ["localhost:3000"]
```

4.6 — Lancement des services

Tu as lancé les services à la main :

Prometheus :

```
./prometheus --config.file=prometheus.yml
```

Node Exporter :

```
./node_exporter
```

Grafana :

```
sudo systemctl enable grafana-server
```

```
sudo systemctl start grafana-server
```

6.7 — Accéder à Grafana.

Dans ton navigateur (machine hôte) :

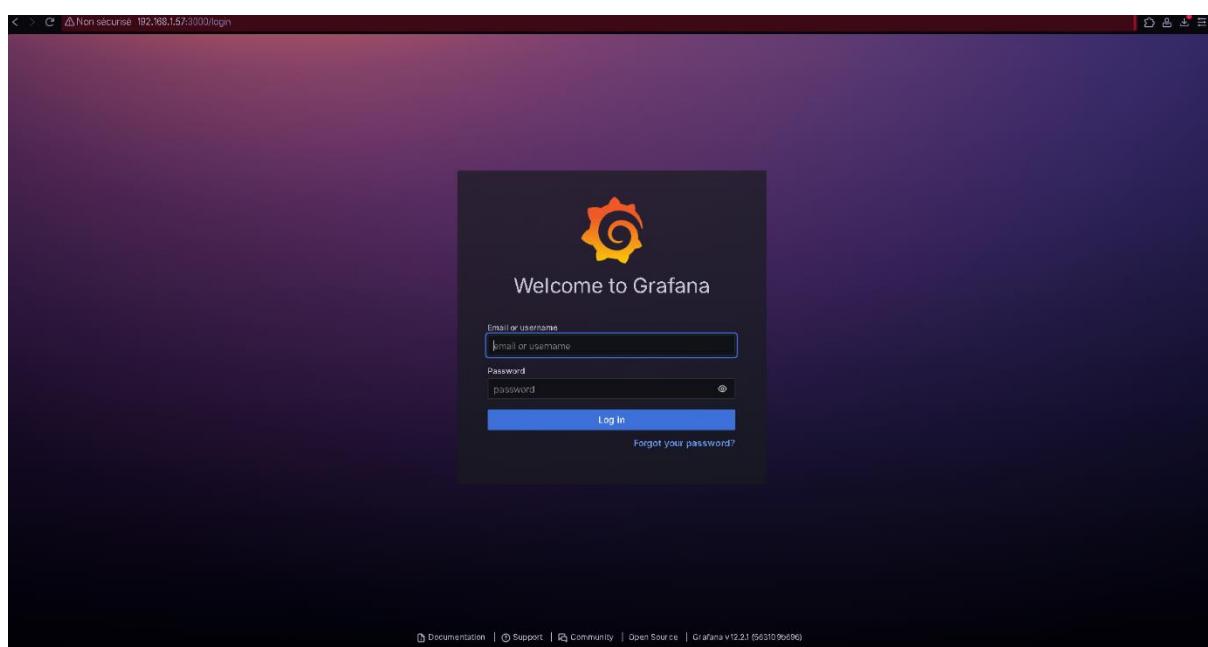
<http://localhost:3000>

ou

<http://<ADRESSEIP>:3000>

Identifiants par défaut :

- **admin**
- **admin**



7 — Grafana, Node Explorer, Prometheus

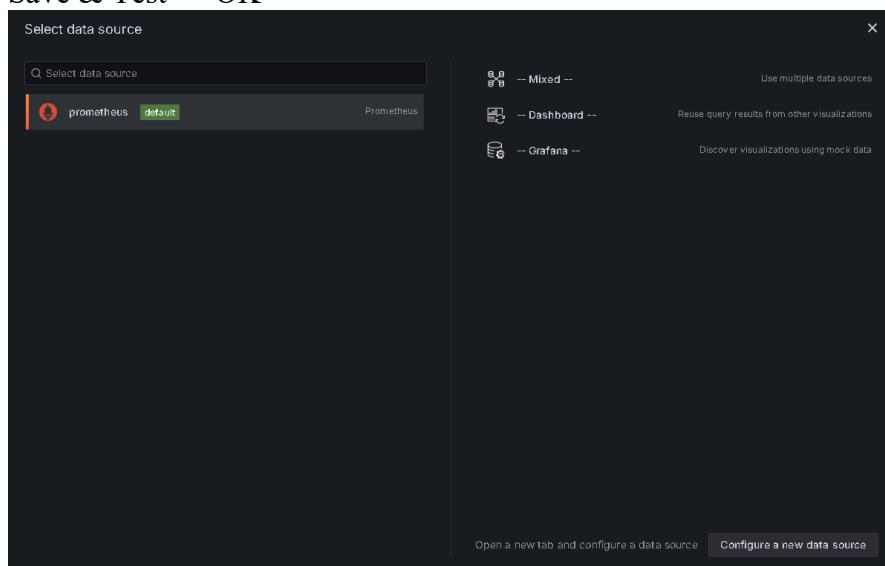
7.1 — Accès à l'interface

Grafana était déjà installé et accessible :

- URL : `http://localhost:3000`
- Login : admin / admin (mot de passe changé ensuite)

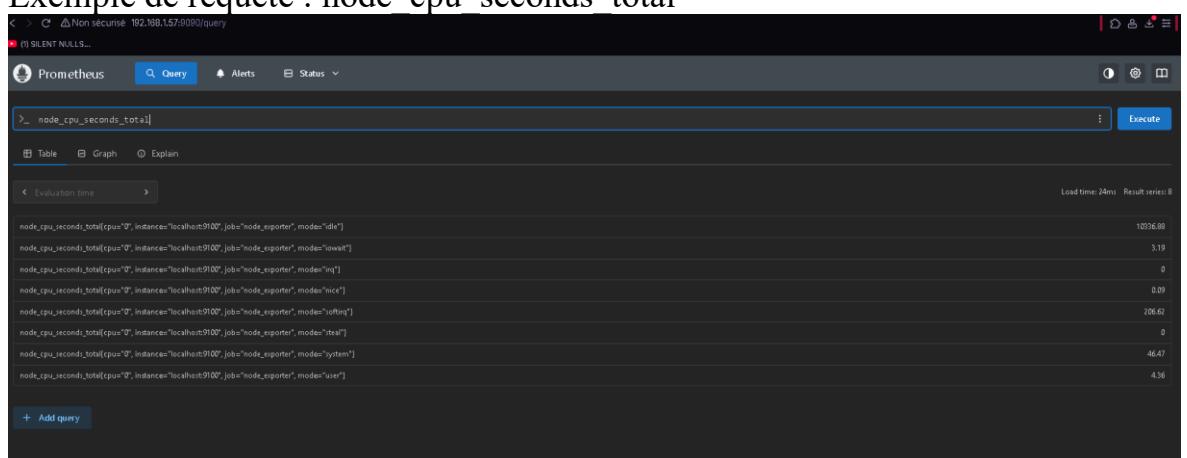
7.2 — Configuration de la source de données

- Grafana → Configuration → Data Sources → Add → Prometheus
- URL : `http://localhost:9090`
- Save & Test → OK



7.3 — Prometheus

- Lancer Prometheus → `./prometheus --config.file=prometheus.yml`
- Vérifier que les targets Prometheus et Node Exporter sont UP
- Exemple de requête : `node cpu seconds total`



7.4 — Node Exporter

- Service systemd actif → métriques accessibles sur Prometheus
- Exemple : surveillance mémoire, CPU, disque

The screenshot shows the Prometheus web interface. At the top, there are tabs for 'Prometheus', 'Query', 'Alerts', and 'Status > Target health'. Below this, there's a search bar and filters for 'Select scrape pool', 'Filter by target health', and 'Filter by endpoint or labels'. Two targets are listed:

- node_exporter**: Endpoint `http://localhost:9100/metrics`, Labels: `instance="localhost:9100"`, `job="node_exporter"`. Last scrape: 7.82s ago, State: UP.
- prometheus**: Endpoint `http://localhost:9090/metrics`, Labels: `app="prometheus"`, `instance="localhost:9090"`, `job="prometheus"`. Last scrape: 8.638s ago, State: UP.

7.5 — Grafana

- Dashboard Node Exporter configuré → visualisation en temps réel
- Exemple d'utilisation : détecter pics CPU/mémoire, vérifier ressources

The screenshot shows the Grafana configuration interface. On the left, there's a sidebar with navigation links like Home, Dashboards, Shared, Recent, Plugins, Editors, Metrics, Metrics, and Admin. The main area has sections for 'Basic' (with a 'Welcome to Grafana' message), 'TUTORIAL' (linking to 'DATA SOURCES AND DASHBOARDS' and 'Grafana fundamentals'), 'DATA SOURCES' (with a 'Add your first data source' button), and 'DASHBOARDS' (with a 'Create your first dashboard' button). Below these are sections for 'Dashboards', 'Shared dashboards', and 'Recently viewed dashboards'. A sidebar on the right lists 'Recent dashboards' and 'Recent queries'. At the bottom, there are news items from the blog.

7.6 — Ce qui n'a pas été fait

- Pas d'alerting Prometheus configuré
- Pas de métriques Grafana exportées vers Prometheus
- Pas de multi-node
- Pas de création d'utilisateurs spécifiques pour les services
- (Ce chapitre montre que **tout le setup est opérationnel**, prêt à être présenté. Il n'est pas nécessaire d'ajouter des configurations supplémentaires ou des fonctionnalités non implémentées mais il est probablement recommandé de le faire.)

8 — Conclusion / Résumé du projet

Ce projet montre une **stack de monitoring complète** :

- Ubuntu Server sous VirtualBox
- Prometheus pour la collecte des métriques
- Node Exporter pour surveiller le serveur
- Grafana pour visualiser les données via dashboards

Grâce à ce projet, toutes les étapes de **l'installation, de la configuration et de la vérification des services** ont été réalisées avec succès. La stack est pleinement fonctionnelle et prête à être utilisée pour la surveillance d'un serveur, tout en constituant un projet concret et démontrable pour un portfolio.

📌 Points clés à retenir :

- Les services Prometheus, Node Exporter et Grafana sont opérationnels.
- Les métriques sont collectées et visualisées en temps réel.
- Le projet est complet, fonctionnel et documenté, prêt à être partagé.

