



1. Pré-requis

Sources

Lien utile :

Page officiel Prometheus : <https://prometheus.io/download/>

Page officiel Grafana : <https://grafana.com/docs/grafana/latest/setup-grafana/installation/debian/>

Dépôt Github :

Prometheus : <https://github.com/prometheus/prometheus/releases>

Node exporter : https://github.com/prometheus/node_exporter/releases

Email

Mail : plot.6.turgot@gmail.com | Mdp : Labo-113

VM

Etape 1 : Créer la VM, avec les configurations ci-dessous.

Nom	ID	OS	Stockage	RAM	CPU	Réseau	Services/Remarques
SRV-Prometheus	MV202	Ubuntu 24.04	20Go	4Go	1 processor 1 cœurs	VVMBR2	192.168.17.10/24

Configuration réseau de la machine SRV-Prometheus

Annuler Filaire Appliquer

Détails Identité IPv4 IPv6 Sécurité

Méthode IPv4

- ☐ Automatique (DHCP)
- ☒ Manuel
- ☐ Réseau local seulement
- ☐ Désactiver
- ☐ Partagée avec d'autres ordinateurs

Adresses

Adresse	Masque de réseau	Passerelle
192.168.17.10	255.255.255.0	192.168.17.254

DNS Automatique ☒

8.26.56.26

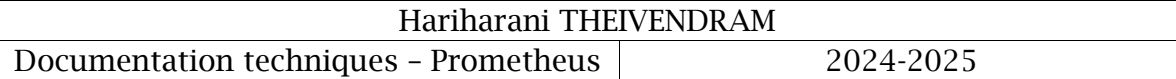
Séparer les adresses IP avec des virgules

MAJ du système

```
user@srv-Prometheus:~$ sudo apt update && sudo apt upgrade -y
```

Créer un utilisateur pour Prometheus

```
user@srv-Prometheus:~$ sudo useradd --no-create-home --shell /bin/false prometheus
```



2



Ajoutez le contenu suivant :

```
[Unit]
Description=Prometheus
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=500
StartLimitBurst=5

[Service]
User=prometheus
Group=prometheus
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/prometheus \
  --config.file=/etc/prometheus/prometheus.yml \
  --storage.tsdb.path=/var/lib/prometheus \
  --web.console.templates=/etc/prometheus/consoles \
  --web.console.libraries=/etc/prometheus/console_libraries \
  --web.listen-address=0.0.0.0:9090 \
  --web.enable-lifecycle

[Install]
WantedBy=multi-user.target
```

Recharger systemd et démarrer Prometheus

user@srv-Prometheus:~/tmp/prometheus-2.37.0.linux-amd64\$ sudo systemctl daemon-reload

user@srv-Prometheus:~/tmp/prometheus-2.37.0.linux-amd64\$ sudo systemctl enable

prometheus

user@srv-Prometheus:~/tmp/prometheus-2.37.0.linux-amd64\$ sudo systemctl start prometheus

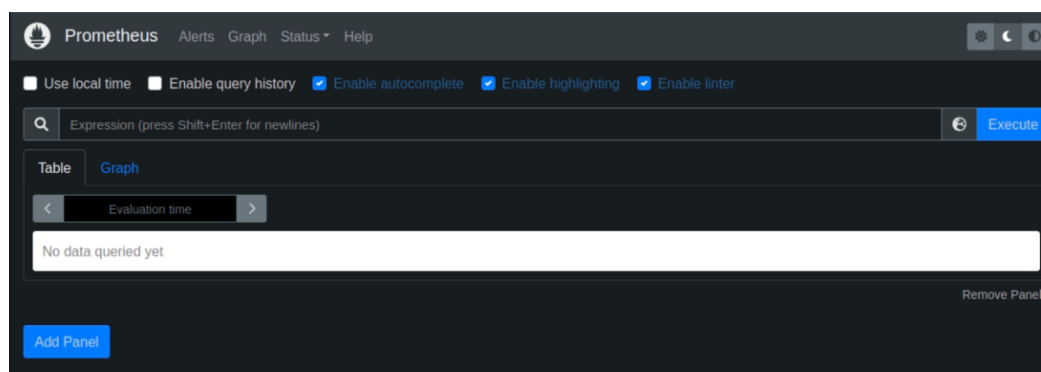
user@srv-Prometheus:~/tmp/prometheus-2.37.0.linux-amd64:~\$ sudo systemctl status

prometheus

```
harani@srv-supervision:~$ sudo systemctl enable prometheus
Synchronizing state of prometheus.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable prometheus
harani@srv-supervision:~$ sudo systemctl start prometheus
harani@srv-supervision:~$ sudo systemctl status prometheus
● prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-03-16 00:01:53 CET; 9s ago
     Main PID: 8701 (prometheus)
        Tasks: 6 (limit: 4610)
      Memory: 15.1M (peak: 15.4M)
         CPU: 37ms
    CGroup: /system.slice/prometheus.service
            └─8701 /usr/local/bin/prometheus --config.file=/etc/prometheus/prometheus.yml --storage.tsdb.path=/data...
```

Interface graphique de Prometheus

http://<Adresse_IP_server>:9090 soit ici : <http://192.168.17.10:9090>



Aller dans **Status > Targets** et on peut visualiser que notre serveur est bien active.



Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instances="localhost:9090" job="prometheus"	7.778s ago	2.332ms	

3. Node Exporter (client Linux)

Installer sur tous les machines Linux de l'infrastructure virtuel.

Créer un utilisateur pour Node Exporter

```
user@srv-Prometheus:~$ sudo useradd --no-create-home --shell /bin/false node_exporter
user@srv-Prometheus:~$ cd /tmp
```

Télécharger, extraire et déplacer Node Exporter

```
user@srv-Prometheus:~/tmp $ wget
https://github.com/prometheus/node_exporter/releases/download/v.1.9.1/node_exporter-1.9.1.linux-amd64.tar.gz

user@srv-Prometheus:~/tmp $ tar -xvf node_exporter-1.9.0.linx-amd64.tar.gz

user@srv-Prometheus:~/tmp $ cd node_exporter-*/
```

```
user@srv-Prometheus:~/tmp/node_exporter-1.9.1.linux-amd64 $ sudo mv node_exporter
/usr/local/bin/
```

```
user@srv-Prometheus:~/tmp/node_exporter-1.9.1.linux-amd64 $ sudo chown
node_exporter:node_exporter /usr/local/bin/node_exporter
```

Création du fichier node_exporter.service

```
user@srv-Prometheus:~$ sudo nano /etc/systemd/system/node_exporter.service
```



```
GNU nano 7.2 /etc/systemd/system/node_exporter.service *
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

StartLimitIntervalSec=500
StartLimitBurst=5

[Service]
User=node_exporter
Group=node_exporter
Type=simple
Restart=on-failure
RestartSec=5s
ExecStart=/usr/local/bin/node_exporter \
  --collector.logind

[Install]
WantedBy=multi-user.target
```

Démarrer le Node Exporter

user@srv-Prometheus:~\$ sudo systemctl daemon-reload

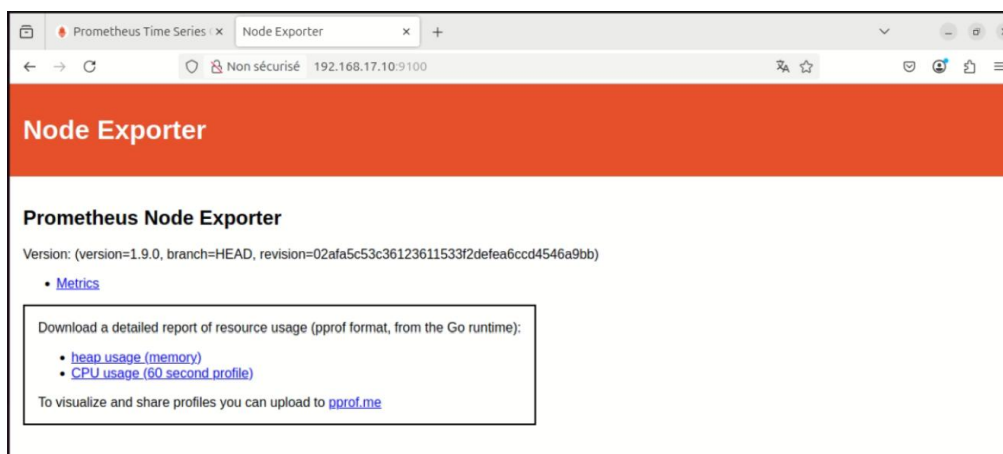
user@srv-Prometheus:~\$ sudo systemctl enable node_exporter

user@srv-Prometheus:~\$ sudo systemctl start node_exporter

user@srv-Prometheus:~\$ sudo systemctl status node_exporter

```
harani@srv-supervision:~$ sudo systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-03-16 01:03:50 CET; 9s ago
     Main PID: 11133 (node_exporter)
        Tasks: 3 (limit: 4610)
       Memory: 2.0M (peak: 2.2M)
          CPU: 5ms
      CGroup: /system.slice/node_exporter.service
              └─11133 /usr/local/bin/node_exporter --collector.logind
```

Sur le navigateur taper : 192.168.17.10:9100



Modification du fichier de configuration Prometheus.yml

user@srv-Prometheus:~\$ sudo nano /etc/prometheus/prometheus.yml

Ajouter cette section sous **scrape_configs** :



```
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: "node_exporter"
    static_configs:
      - targets: ["localhost:9100"]
```

Le localhost peut-être remplacer par l'adresse IP des hôtes aussi.

Redémarrer le service prometheus :

```
user@srv-Prometheus:~$ sudo systemctl restart Prometheus
```

Vérifier l'interface graphique de Prometheus. Vous devrez trouver comme ci-dessous dans **Status > Targets** :

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
node_exporter (1/1 up) show less					
http://localhost:9100/metrics	UP	instance="localhost:9100" job="node_exporter"	12.323s ago	23.972ms	
prometheus (1/1 up) show less					
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	9.526s ago	2.274ms	

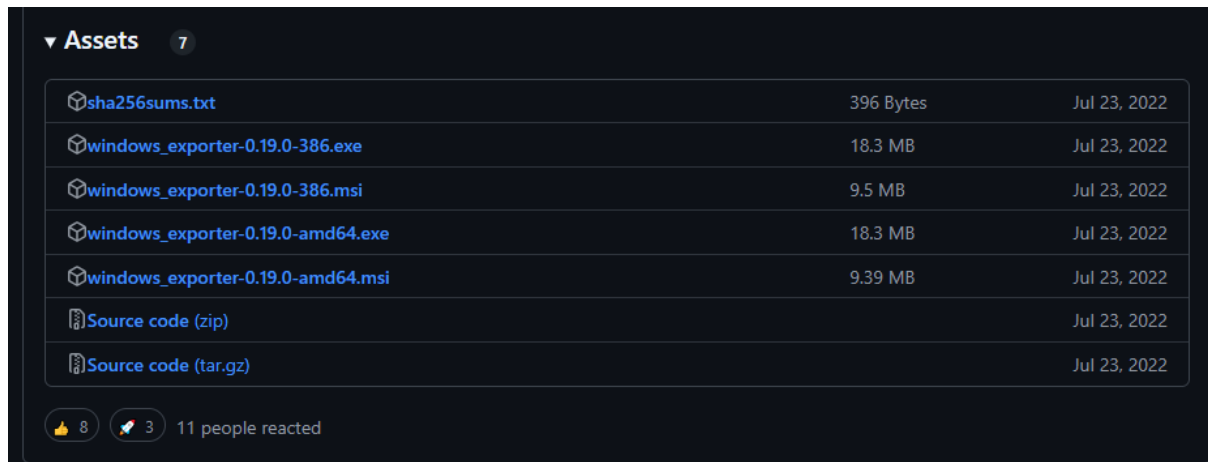


4. Windows Exporter (client Windows)

Installation de Windows Exporter

Dépôt Github : https://github.com/prometheus-community/windows_exporter/releases

Ici, je vais choisir la version 0.19.0 et télécharger le .exe



Lancer le fichier, une fois télécharger.

Lancer Windows Exporter en arrière-plan au démarrage

Pour éviter de le lancer à chaque fois on va le lancer en tant que service en arrière-plan au démarrage :

Clic droit > créer un raccourci

Win + R > shell:startup déplacer le raccourci dedans.

Ou

C:\Users\user\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup

Et copier le fichier .exe



5. Grafana

Installation de Grafana

```
user@srv-Prometheus:~$ sudo apt-get install -y apt-transport-https software-properties-common wget
```

```
user@srv-Prometheus:~$ sudo mkdir -p /etc/apt/keyrings/
```

```
user@srv-Prometheus:~$ wget -q -O - https://apt.grafana.com/gpg.key | gpg --dearmor | sudo tee /etc/apt/keyrings/grafana.gpg > /dev/null
```

```
user@srv-Prometheus:~$ echo "deb [signed-by=/etc/apt/keyrings/grafana.gpg] https://apt.grafana.com stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list
```

```
user@srv-Prometheus:~$ sudo apt-get update
```

```
user@srv-Prometheus:~$ sudo apt-get install grafana
```

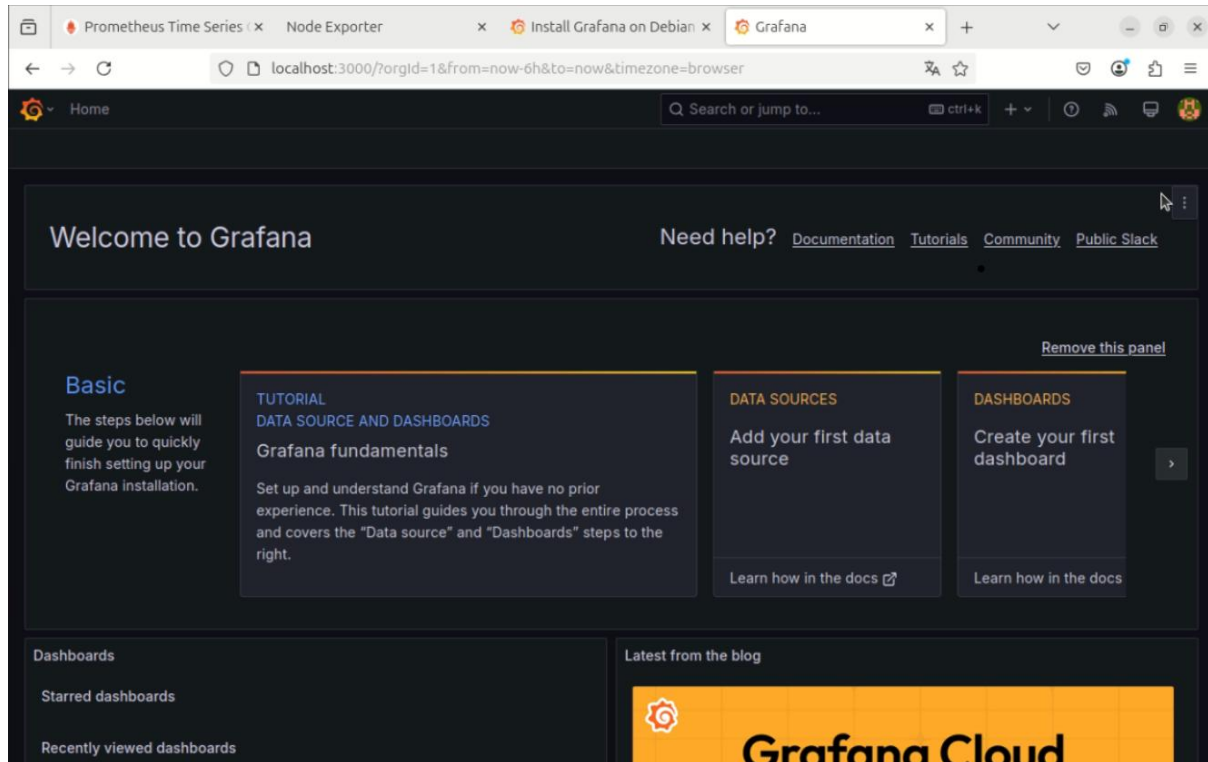
```
user@srv-Prometheus:~$ sudo /bin/systemctl start grafana-server
```

```
user@srv-Prometheus:~$ sudo /bin/systemctl status grafana-server
```

```
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; disabled; preset: enabled)
   Active: active (running) since Sun 2025-03-16 21:35:49 CET; 1min 25s ago
     Docs: http://docs.grafana.org
    Main PID: 4841 (grafana)
      Tasks: 7 (limit: 4610)
    Memory: 106.5M (peak: 110.9M)
       CPU: 2.576s
    CGroup: /system.slice/grafana-server.service
            └─4841 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pidfile=/run/grafana/grafana.pid
```

Vérifier en accédant l'interface Web avec : <http://localhost:3000>

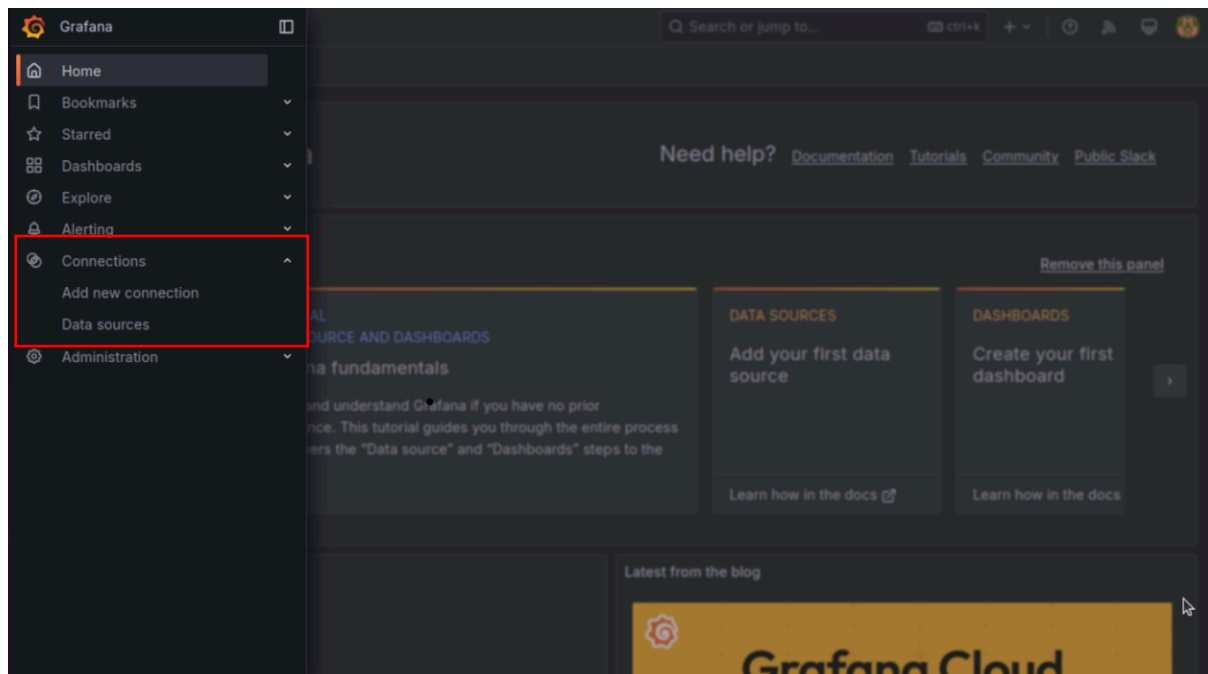
Changer le mot de passe par défaut et vous tomberez sur cette page :



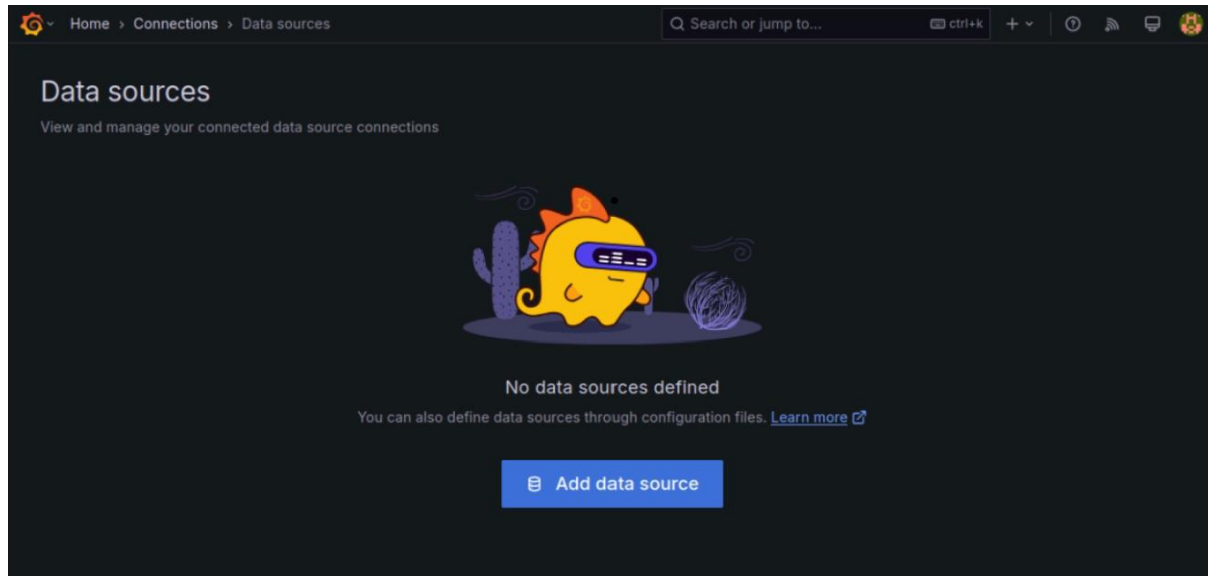
Par défaut : id : admin | mdp : admin

Liées les sources de données

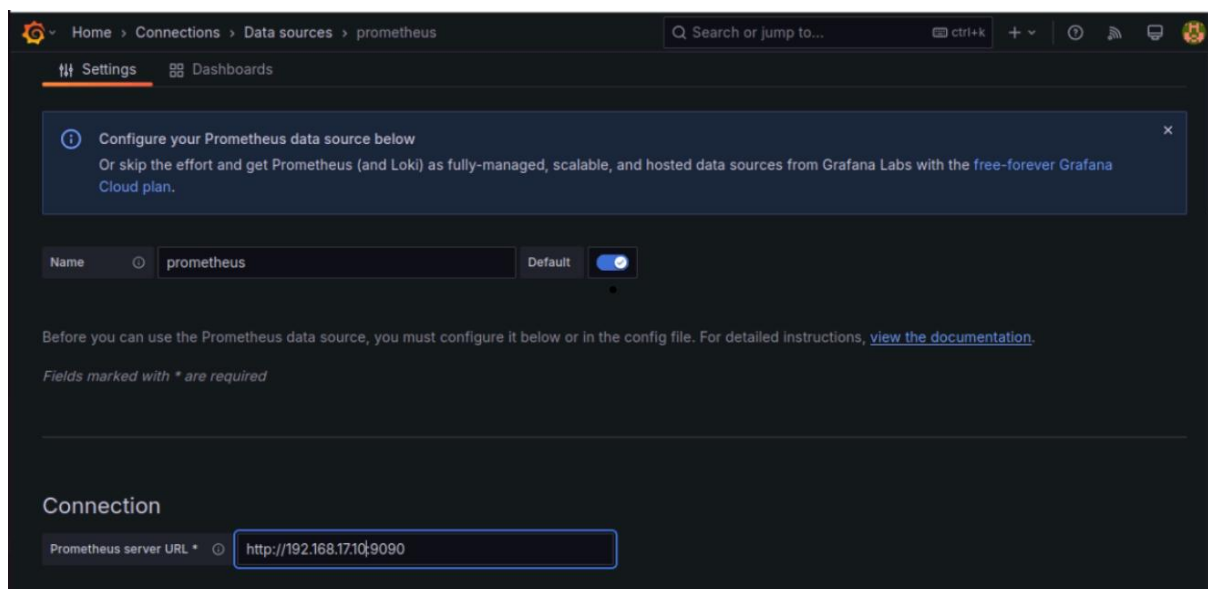
Aller dans **Connections > Data sources**



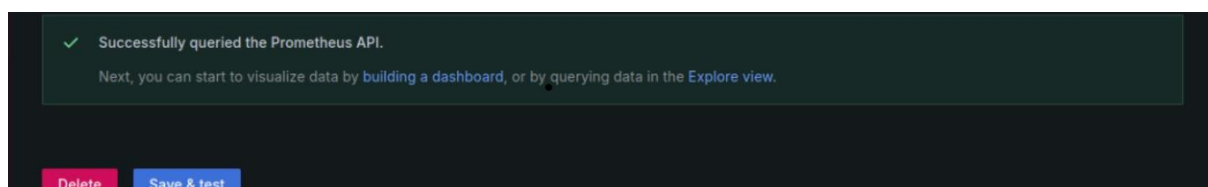
Cliquer sur « **Add data source** » :



Sélectionner Prometheus et renseigner l'URL : <http://localhost:9090> oui bien l'adresse du serveur :



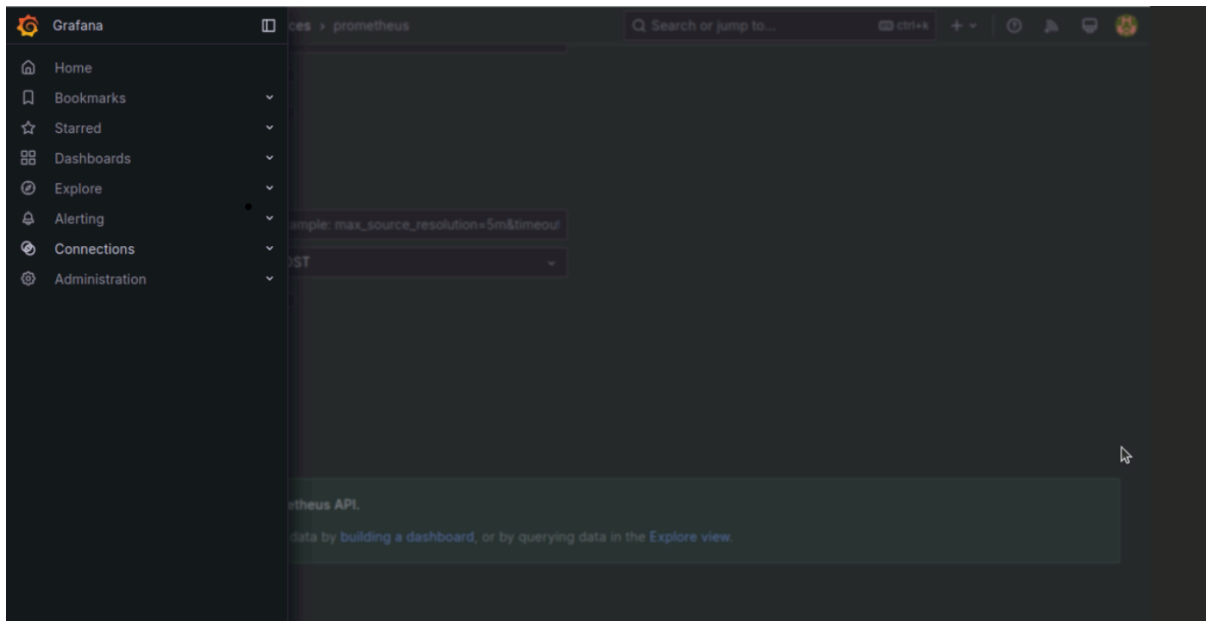
Aller tout en bas et cliquer sur Valider & Test :



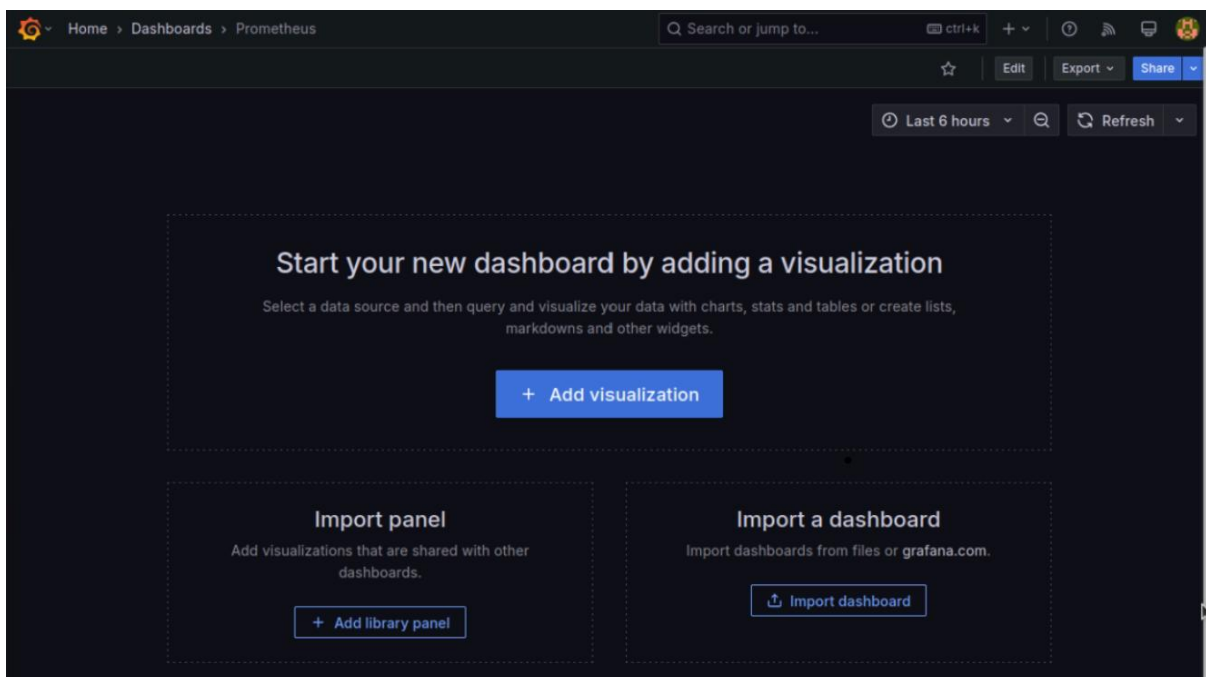


Création du Dashboard

Cliquer dans Dashboard



Cliquer à droite dans New > New Dashboard > Import Dashboard et nommé le Dashboard Prometheus.



Importation d'un modèle de Dashboard :

Grafana propose des dashboards prêts à l'emploi sur **Grafana.com**. dans notre contexte :

1. Chercher **Node Exporter Full** (ID 1860), cliquer sur **"Import"**, puis entre l'ID du dashboard et cliquer sur **load**



Find and import dashboards for common applications at grafana.com/dashboards

1860 Load

Sélectionner Prometheus pour les données de sources et importer le :

Home > Dashboards > Import dashboard

Import dashboard from file or Grafana.com

Importing dashboard from [Grafana.com](https://grafana.com)

Published by rfm02

Updated on 2024-05-22 18:07:35

Options

Name Node Exporter Full

Folder prometheus default prometheus

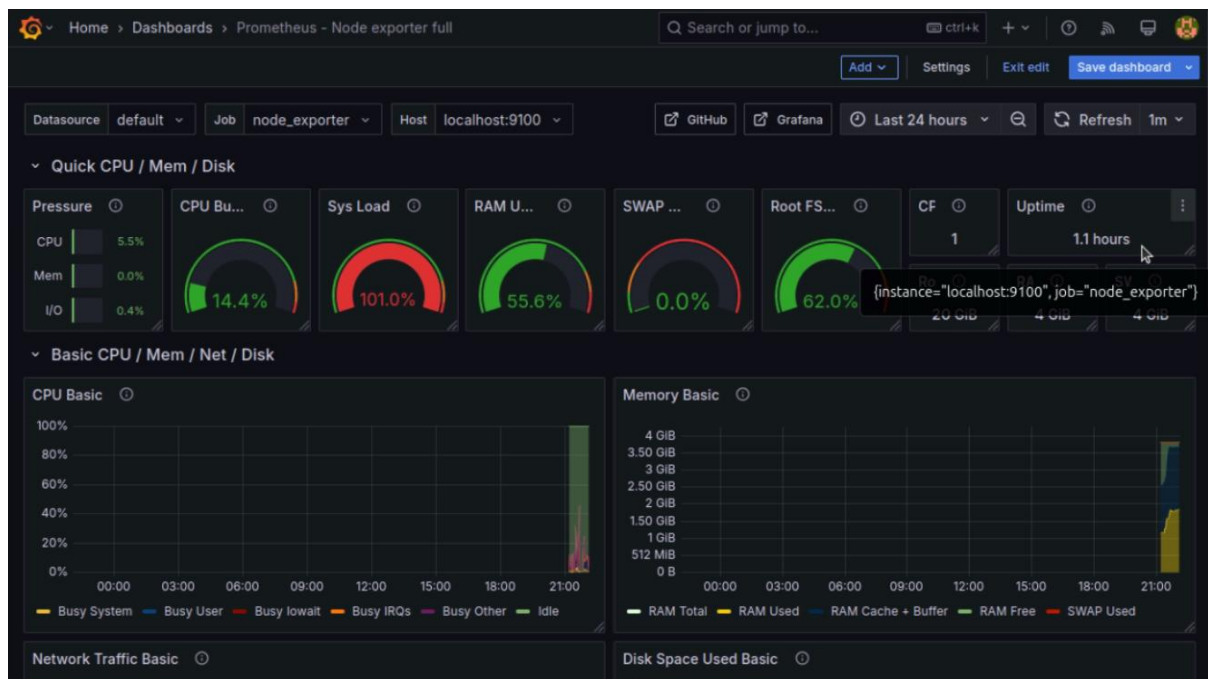
used for uniquely identify a dashboard. A UID allows having consistent IDs of a dashboard will not break

Open advanced data source picker + Change uid

Select a Prometheus data source

Import Cancel

Résultat obtenu du Dashboard :





6. Alertmanager

Créer les dossiers nécessaires

```
user@srv-Prometheus:~$ sudo useradd --no-create-home --shell /bin/false alertmanager
```

```
user@srv-Prometheus:~$ sudo mkdir -p /etc/alertmanager /var/lib/alertmanager
```

```
user@srv-Prometheus:~$ sudo chown -R alertmanager:alertmanager /etc/alertmanager  
/var/lib/alertmanager
```

Télécharger Alertmanager

```
user@srv-Prometheus:~$ cd /tmp
```

```
user@srv-Prometheus:~/tmp$ wget
```

```
https://github.com/prometheus/alertmanager/releases/download/v0.28.0/alertmanager-  
0.28.0.linux-amd64.tar.gz
```

```
user@srv-Prometheus:~/tmp$ tar xvf alertmanager-*
```

```
user@srv-Prometheus:~/tmp$ cd alertmanager-*/
```

```
user@srv-Prometheus:~/tmp/alertmanager-0.28.0.linux-amd64$ sudo cp alertmanager amtool  
/usr/local/bin/
```

```
user@srv-Prometheus:~/tmp/alertmanager-0.28.0.linux-amd64$ sudo chown  
alertmanager:alertmanager /usr/local/bin/alertmanager /usr/local/bin/amtool
```

Créer un mot de passe applicative pour alertmanager

On ne peut pas utiliser le mot de passe Gmail classique ici. Il faut créer un mot de passe d'application.

Étapes :

1. Aller sur : <https://myaccount.google.com/security>
2. Activer la **validation en deux étapes**
3. Ensuite, tu verras une option “**Mots de passe des applications**”
4. Générer un mot de passe pour l'application (ex : “Alertmanager”)
5. Noter ce mot de passe

Créer le fichier de configuration

```
user@srv-Prometheus:~/tmp/alertmanager-0.28.0.linux-amd64$ sudo nano  
/etc/alertmanager/alertmanager.yml
```



```
global:
  smtp_smarthost: 'smtp.gmail.com:587'
  smtp_from: 'plot.6.turgot@gmail.com'
  smtp_auth_username: 'plot.6.turgot@gmail.com'
  smtp_auth_password: 'gmgeoijtbdxwahxb'
  smtp_require_tls: true
  resolve_timeout: 5m

route:
  receiver: 'default'

receivers:
  - name: 'default'
    email_configs:
      - to: 'plot.6.turgot@gmail.com'
```

Dans smtp_auth_password, entrer le mot de passe généré par Google sans espace.

Créer le service systemd

```
user@srv-Prometheus:~/tmp/alertmanager-0.28.0.linux-amd64 $ sudo nano
/etc/systemd/system/alertmanager.service
```

```
[Unit]
Description=Alertmanager
Wants=network-online.target
After=network-online.target

[Service]
User=alertmanager
Group=alertmanager
Type=simple
ExecStart=/usr/local/bin/alertmanager \
  --config.file=/etc/alertmanager/alertmanager.yml \
  --storage.path=/var/lib/alertmanager
Restart=always

[Install]
WantedBy=multi-user.target
```

```
user@srv-Prometheus:~/tmp/alertmanager-0.28.0.linux-amd64 $ sudo systemctl daemon-
reexec
```

```
user@srv-Prometheus:~/tmp/alertmanager-0.28.0.linux-amd64 $ sudo systemctl daemon-
reload
```

```
user@srv-Prometheus:~/tmp/alertmanager-0.28.0.linux-amd64 $ sudo systemctl start
alertmanager
```

```
user@srv-Prometheus:~/tmp/alertmanager-0.28.0.linux-amd64 $ sudo systemctl enable
alertmanager
```

```
user@srv-Prometheus:~/tmp/alertmanager-0.28.0.linux-amd64 $ sudo systemctl status
alertmanager
```




```
user@srv-prometheus:/tmp/alertmanager-0.28.0.linux-amd64$ sudo systemctl status alertmanager
● alertmanager.service - Alertmanager
   Loaded: loaded (/etc/systemd/system/alertmanager.service; enabled; preset: enabled)
   Active: active (running) since Sun 2025-04-06 19:56:47 CEST; 7s ago
     Main PID: 46833 (alertmanager)
        Tasks: 6 (limit: 4610)
      Memory: 13.7M (peak: 13.9M)
         CPU: 45ms
    CGroup: /system.slice/alertmanager.service
            └─46833 /usr/local/bin/alertmanager --config.file=/etc/alertmanager/alertmanager.yml --storage.path=/var/lib/alertmanager

avril 06 19:56:47 srv-prometheus systemd[1]: Started alertmanager.service - Alertmanager.
avril 06 19:56:47 srv-prometheus alertmanager[46833]: time=2025-04-06T17:56:47.514Z level=INFO source=main.go:191 msg=">
avril 06 19:56:47 srv-prometheus alertmanager[46833]: time=2025-04-06T17:56:47.515Z level=INFO source=main.go:192 msg=">
avril 06 19:56:47 srv-prometheus alertmanager[46833]: time=2025-04-06T17:56:47.515Z level=INFO source=cluster.go:185 ms>
avril 06 19:56:47 srv-prometheus alertmanager[46833]: time=2025-04-06T17:56:47.532Z level=INFO source=cluster.go:674 ms>
avril 06 19:56:47 srv-prometheus alertmanager[46833]: time=2025-04-06T17:56:47.553Z level=INFO source=coordinator.go:11>
avril 06 19:56:47 srv-prometheus alertmanager[46833]: time=2025-04-06T17:56:47.554Z level=INFO source=coordinator.go:12>
avril 06 19:56:47 srv-prometheus alertmanager[46833]: time=2025-04-06T17:56:47.555Z level=INFO source=tls_config.go:347>
avril 06 19:56:47 srv-prometheus alertmanager[46833]: time=2025-04-06T17:56:47.555Z level=INFO source=tls_config.go:350>
avril 06 19:56:49 srv-prometheus alertmanager[46833]: time=2025-04-06T17:56:49.532Z level=INFO source=cluster.go:699 ms>
lines 1-20/20 (END)
```

Configurer les règles d'alerte dans Prometheus

user@srv-Prometheus:~\$ sudo nano /etc/prometheus/alert_rules.yml

Configurer une règle d'alerte dans Prometheus pour envoyer une alerte lorsque l'espace disque est inférieure à 20%.

```
groups:
- name: alert_rules
  rules:
  - alert: LowDiskSpace
    expr: (node_filesystem_avail_bytes{fstype=~"ext4|xfs"} / node_filesystem_size_bytes{fstype=~"ext4|xfs"}) < 0.2
    for: 5m
    labels:
      severity: critical
    annotations:
      summary: "Espace disque faible sur {{ $labels.instance }}"
      description: "L'espace disque est inférieur à 20% sur {{ $labels.instance }}"
```

Explication :

La capture montre la configuration d'une règle d'alerte dans un fichier YAML utilisé pour un système de surveillance. Voici l'explication des lignes principales :

- **Groups** : Crée un groupe de règles, ici nommé `alert_rules`.
- **Rules** : Contient une liste de règles d'alerte spécifiques.
- **Alert**: `LowDiskSpace` : Nom de l'alerte, qui est déclenchée en cas de faible espace disque.
- **expr** :
(node_filesystem_avail_bytes{fstype=~"ext4|xfs"} / node_filesystem_size_bytes{fstype=~"ext4|xfs"}) < 0.2
 - Vérifie si l'espace disponible (en octets) est inférieur à 20 % de la taille totale sur des systèmes de fichiers de type ext4 ou xfs.
- **for**: `5m` : La condition doit durer 5 minutes avant que l'alerte soit déclenchée.
- **labels**: `severity: critical` : Indique que l'alerte a un niveau de gravité critical.
- **Annotations** :
 - **summary** : Résume l'alerte (ici, espace disque faible).
 - **Description** : Fournit un message détaillé en français, précisant l'instance et le point de montage concernés.



Configurer Prometheus pour envoyer des alertes

```
user@srv-Prometheus:~/alertmanager-0.28.0.linux-amd64$ sudo nano  
/etc/prometheus/prometheus.yml
```

Modifier les parties comme ci-dessous :

```
# Alertmanager configuration  
alerting:  
  alertmanagers:  
    - static_configs:  
      - targets:  
        - 'localhost:9093'
```

```
# Load rules once and periodic  
rule_files:  
  - "alert_rules.yml"
```

Redémarrer tout les services

```
user@srv-Prometheus:~$ sudo systemctl restart prometheus  
user@srv-Prometheus:~$ sudo systemctl restart alertmanager
```

Vérification

Aller sur l'interface web de Prometheus :

http://localhost:9090

- Menu Alerts
- Tu devrais voir "LowDiskSpace"
- Déclenche-la manuellement en réduisant ton espace disque (ou en modifiant la règle pour tester : < 0.95 par exemple).

