

REPUBLIQUE DU SENEGAL



Un peuple-Un but-Une foi

Ministère de l'Enseignement Supérieur et de la Recherche

Direction Générale de l'Enseignement Supérieur Privé

Institut Supérieur d'Informatique

ISI

Projet

Master 1 : Sécurité des systèmes d'information et monétique

Mise en place de Prometheus , Grafana et Node Exporter pour la supervision des systèmes avec alerte par email



Réalisé par :

Khadim Mbaye

Kmbamba567@gmail.com

Année académique 2024-2025

SOMMAIRE

INTRODUCTION

I. Présentation des outils

1. Prometheus

2. Node_exporter

3. Grafana

II. Architecture et fonctionnement

III. Cas d'utilisation

IV. Installation et configuration

1. Installation et configuration de Prometheus

2. Installation et configuration de Grafana

3. Installation et configuration de Node Exporter

V. Alerte par email

Conclusion

INTRODUCTION

La surveillance et l'observabilité des systèmes informatiques sont essentielles pour garantir leur bon fonctionnement et optimiser les performances. Prometheus, Grafana et Node Exporter sont des outils open-source largement utilisés pour la collecte, l'analyse et la visualisation des métriques système et applicatives

Objectif du projet : L'objectif de ce projet était de mettre en place une solution de **supervision** performante pour surveiller en temps réel les performances du système d'un serveur Ubuntu à l'adresse 192.168.1.30. En utilisant **Prometheus** pour collecter les métriques, **Node Exporter** pour exposer les données système (CPU, mémoire, disque, etc.), et **Grafana** pour les visualisés sous forme de graphiques exploitables, le but était d'obtenir une vue claire et centralisée des ressources, permettant une détection rapide des anomalies et une gestion proactive de l'infrastructure.

I. Présentation des outils

1. Prometheus

Prometheus est un outil de surveillance et d'alerte développé initialement par SoundCloud et maintenant maintenu par la Cloud Native Computing Foundation (CNCF). Il collecte des métriques sous forme de séries temporelles à l'aide de requêtes HTTP pull.

Caractéristiques principales :

- Modèle de données basé sur des séries temporelles.
- Langage de requête puissant (PromQL).
- Système d'alerte intégré (Alertmanager).
- Facilité d'intégration avec divers exportateurs et services cloud.

2. Node Exporter

Node Exporter est un agent qui collecte des métriques système (CPU, mémoire, disque, réseau, etc.) et les expose sous une forme exploitable par Prometheus.

Fonctionnalités principales :

- Surveillance des ressources matérielles et logicielles d'un serveur.
- Collecte de statistiques sur les processus, les systèmes de fichiers et l'utilisation des interfaces réseau.
- Facilité de déploiement et configuration simple.

3. Grafana

Grafana est un outil de visualisation et d'analyse de données qui permet d'afficher les métriques collectées par Prometheus sous forme de tableaux de bord interactifs.

Caractéristiques principales :

- Supporte plusieurs sources de données (Prometheus, InfluxDB, Elasticsearch, etc.).
- Interface utilisateur intuitive et personnalisable.
- Capacité à générer des alertes et à intégrer des panels dynamiques.
- Gestion des accès et des permissions pour les utilisateurs.

II. Architecture et fonctionnement

L'architecture classique d'un système de surveillance basé sur ces outils se compose des éléments suivants :

- ❖ **Node Exporter** collecte les métriques système et les expose sur un Endpoint HTTP.
- ❖ **Prometheus** interroge périodiquement Node Exporter et stocke les données sous forme de séries temporelles.
- ❖ **Grafana** se connecte à Prometheus pour récupérer et afficher les métriques sous forme de graphiques et de tableaux de bord interactifs.
- ❖ **Alertmanager** (facultatif) peut être utilisé pour gérer les alertes basées sur les règles définies dans Prometheus.

III. Cas d'utilisation

Les outils Prometheus, Grafana et Node Exporter sont utilisés dans divers scénarios tels que :

- **Surveillance des serveurs** : suivi des performances des CPU, de la mémoire, des disques et des interfaces réseau.
- **Observabilité des applications** : mesure des temps de réponse, taux d'erreur et charge des services applicatifs.
- **Gestion des alertes** : détection des anomalies et envoi de notifications en cas de dépassement de seuils critiques.
- **Optimisation des infrastructures cloud et on-premise** : analyse des tendances et prévisions de charge.

IV. Installation et configuration

1. Installation et configuration de Prometheus

Étape 1 : Mise à jour du système

Assurons-nous que notre système Ubuntu est à jour avant de démarrer :

```
root@mbaye-VirtualBox:~# apt-get update
```

Étape 2 : Installation de Prometheus

Créer un utilisateur

```
root@mbaye-VirtualBox:~# useradd --system --no-create-home --shell /bin/false prometheus
root@mbaye-VirtualBox:~#
```

Télécharger Prometheus : Rendez-vous sur la [page de téléchargement de Prometheus](#) pour obtenir la dernière version

```
root@mbaye-VirtualBox:~# wget https://github.com/prometheus/prometheus/releases/download/v3.2.1/prometheus-3.2.1.linux-amd64.tar.gz
```

Décompresser

```
root@mbaye-VirtualBox:~# tar -xvf prometheus-3.2.1.linux-amd64.tar.gz
```

On se déplace dans le dossier prometheus-3.2.1.linux-amd64.tar.gz

```
root@mbaye-VirtualBox:~# cd prometheus-3.2.1.linux-amd64
```

Créer des répertoires pour Prometheus

```
root@mbaye-VirtualBox:~/prometheus-3.2.1.linux-amd64# mkdir -p /data /etc/prometheus
```

Déplacer les fichiers

```
root@mbaye-VirtualBox:~/prometheus-3.2.1.linux-amd64# mv prometheus promtool /usr/local/bin/
```

```
root@mbaye-VirtualBox:~/prometheus-3.2.1.linux-amd64# mv prometheus.yml /etc/prometheus/prometheus.yml  
root@mbaye-VirtualBox:~/prometheus-3.2.1.linux-amd64#
```

Définir les permissions

```
root@mbaye-VirtualBox:~/prometheus-3.2.1.linux-amd64# chown -R prometheus:prometheus /etc/prometheus/ /data/
root@mbaye-VirtualBox:~/prometheus-3.2.1.linux-amd64#
```

Créer un service systemd pour Prometheus

```
root@mbaye-VirtualBox:~/prometheus-3.2.1.linux-amd64# cd /etc/systemd/system
root@mbaye-VirtualBox:/etc/systemd/system#
```

```
root@mbaye-VirtualBox:/etc/systemd/system# touch prometheus.service
root@mbaye-VirtualBox:/etc/systemd/system#
```

Et ajoutez ce contenu

```
GNU nano 6.2 prometheus.service *
[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=/data \
  --web.console.templates=/etc/prometheus/consoles \
  --web.console.libraries=/etc/prometheus/console_libraries \
  --web.listen-address=0.0.0.0:9090 \

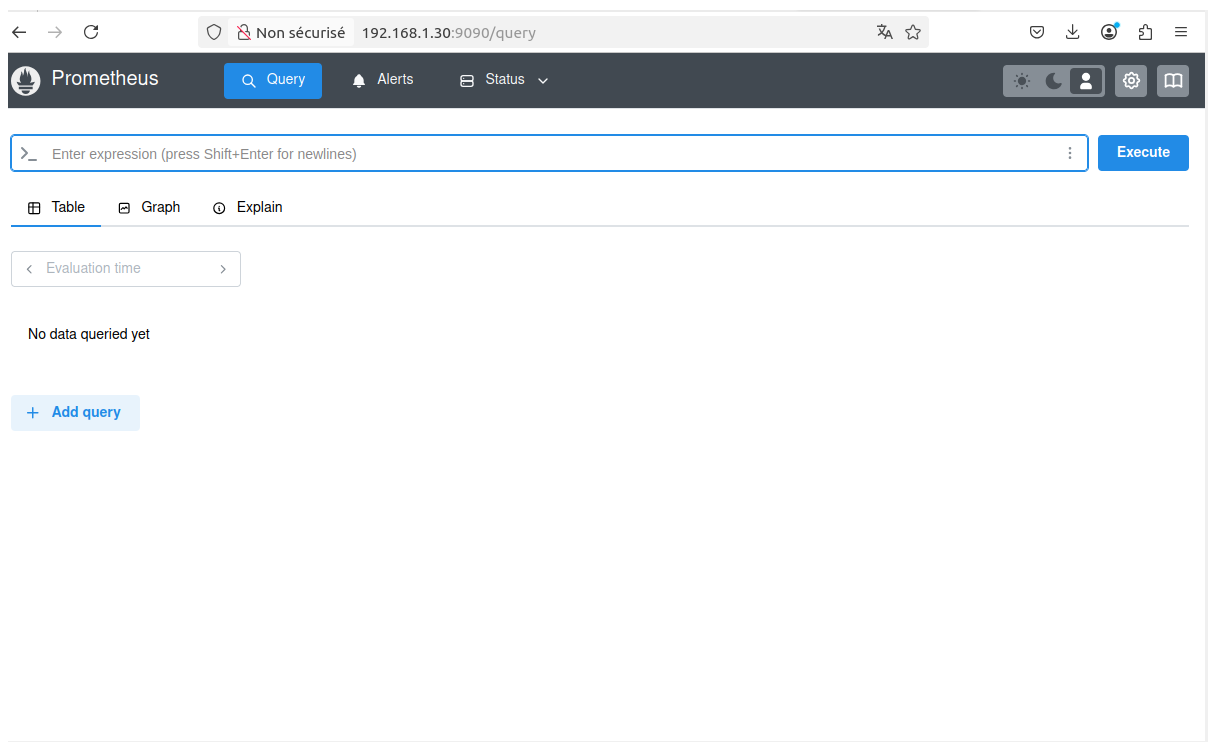
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller   ^J Justifier ^/ Aller ligne
```

Sauvegardez (Ctrl+O, Enter, Ctrl+X) et activez le service

```
etc/systemd/system/prometheus.service.
ot@mbaye-VirtualBox:/etc/systemd/system# systemctl start prometheus
ot@mbaye-VirtualBox:/etc/systemd/system# systemctl status prometheus
prometheus.service - Prometheus
   Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor pr
   Active: active (running) since Thu 2025-03-20 16:32:28 GMT; 9s ago
 Main PID: 84073 (prometheus)
   Tasks: 6 (limit: 2269)
  Memory: 79.9M
     CPU: 324ms
   CGroup: /system.slice/prometheus.service
           └─84073 /usr/local/bin/prometheus --config.file=/etc/prometheus/pr

r 20 16:32:30 mbaye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.91>
r 20 16:32:30 mbaye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.91>
r 20 16:32:30 mbaye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.91>
r 20 16:32:30 mbaye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.93>
r 20 16:32:30 mbaye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.93>
r 20 16:32:30 mbaye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.93>
r 20 16:32:30 mbaye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.94>
r 20 16:32:30 mbaye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.94>
r 20 16:32:30 mbaye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.94>
r 20 16:32:30 mbaye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.94>
nes 1-20/20 (END)
```

Vérifier que Prometheus fonctionne : Accédez à <http://<IP-de-votre-serveur>:9090> dans un navigateur. Vous devriez voir l'interface web de Prometheus.



2. Installation et configuration de Grafana

Ajouter le dépôt Grafana

```
root@mbaye-VirtualBox:~# apt-get install -y apt-transport-https software-properties-common
```

```
root@mbaye-VirtualBox:~# wget -q -O - https://packages.grafana.com/gpg.key | sudo apt-key add -  
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).  
OK  
root@mbaye-VirtualBox:~# echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/sources.list.d/grafana.list  
deb https://packages.grafana.com/oss/deb stable main  
root@mbaye-VirtualBox:~#
```

Installateur Grafana

```
root@mbaye-VirtualBox:~# apt-get update
```

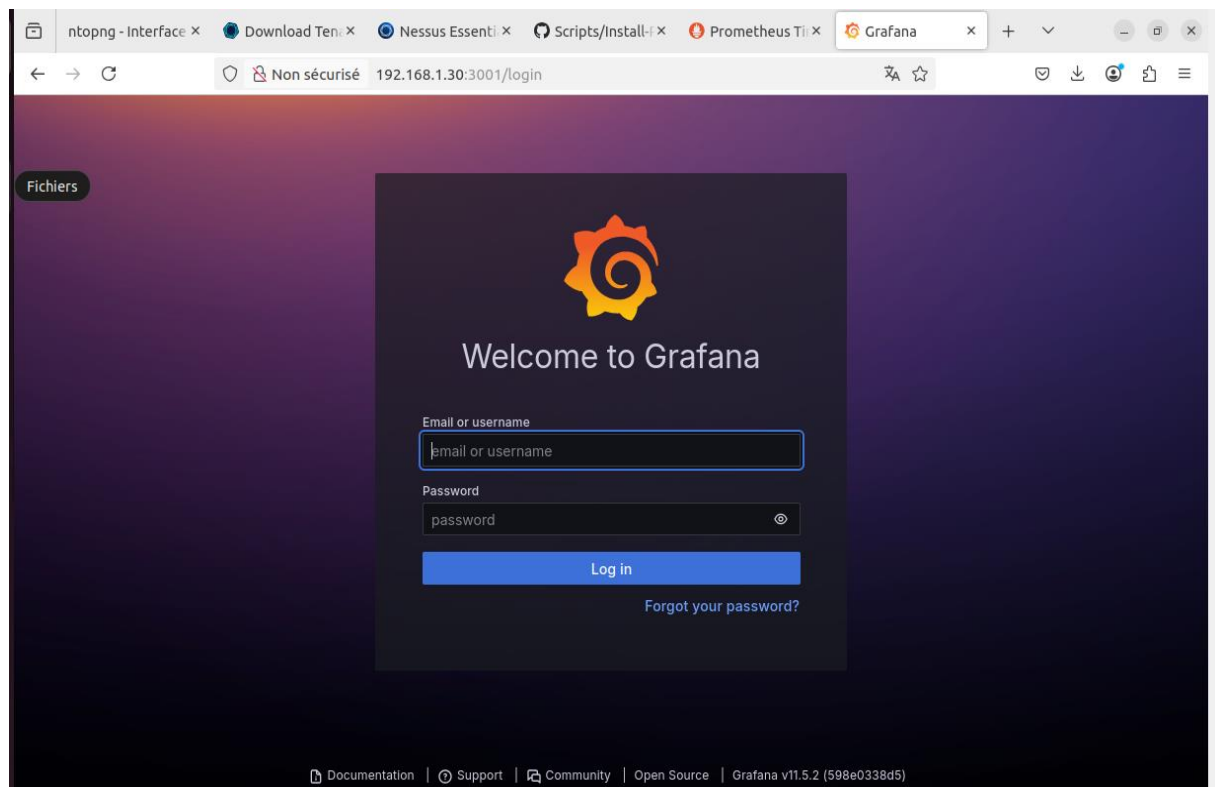
```
root@mbaye-VirtualBox:~# apt-get -y install grafana
```

Démarrer Grafana

```
root@mbye-VirtualBox:~# systemctl start grafana-server
root@mbye-VirtualBox:~# systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/lib/systemd/system/grafana-server.service; enabled; vendor
   Active: active (running) since Fri 2025-03-21 00:08:13 GMT; 10s ago
     Docs: http://docs.grafana.org
   Main PID: 91630 (grafana)
    Tasks: 5 (limit: 2269)
   Memory: 164.0M
     CPU: 2.478s
   CGroup: /system.slice/grafana-server.service
           └─91630 /usr/share/grafana/bin/grafana server --config=/etc/grafan
mar 21 00:08:23 mbye-VirtualBox grafana[91630]: logger=migrator t=2025-03-21T0
mar 21 00:08:23 mbye-VirtualBox grafana[91630]: logger=migrator t=2025-03-21T0
mar 21 00:08:23 mbye-VirtualBox grafana[91630]: logger=migrator t=2025-03-21T0
mar 21 00:08:23 mbye-VirtualBox grafana[91630]: logger=migrator t=2025-03-21T0
mar 21 00:08:23 mbye-VirtualBox grafana[91630]: logger=migrator t=2025-03-21T0
mar 21 00:08:23 mbye-VirtualBox grafana[91630]: logger=migrator t=2025-03-21T0
mar 21 00:08:23 mbye-VirtualBox grafana[91630]: logger=migrator t=2025-03-21T0
mar 21 00:08:23 mbye-VirtualBox grafana[91630]: logger=migrator t=2025-03-21T0
mar 21 00:08:23 mbye-VirtualBox grafana[91630]: logger=migrator t=2025-03-21T0
mar 21 00:08:23 mbye-VirtualBox grafana[91630]: logger=migrator t=2025-03-21T0
mar 21 00:08:23 mbye-VirtualBox grafana[91630]: logger=migrator t=2025-03-21T0
```

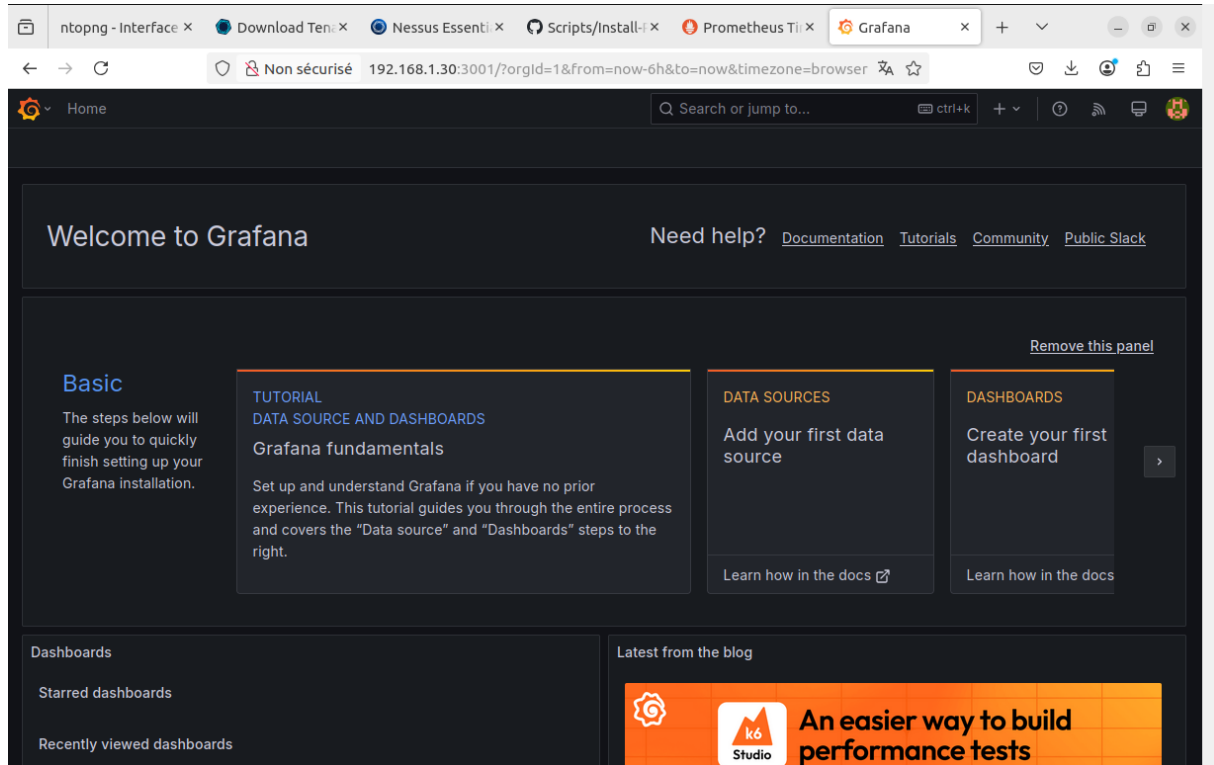
Accéder à Grafana : Ouvrez `http://<IP-de-votre-serveur> :3000` dans un navigateur.

Connectons-nous avec les identifiants par défaut : `admin / admin` (on devra changer le mot de passe ensuite).

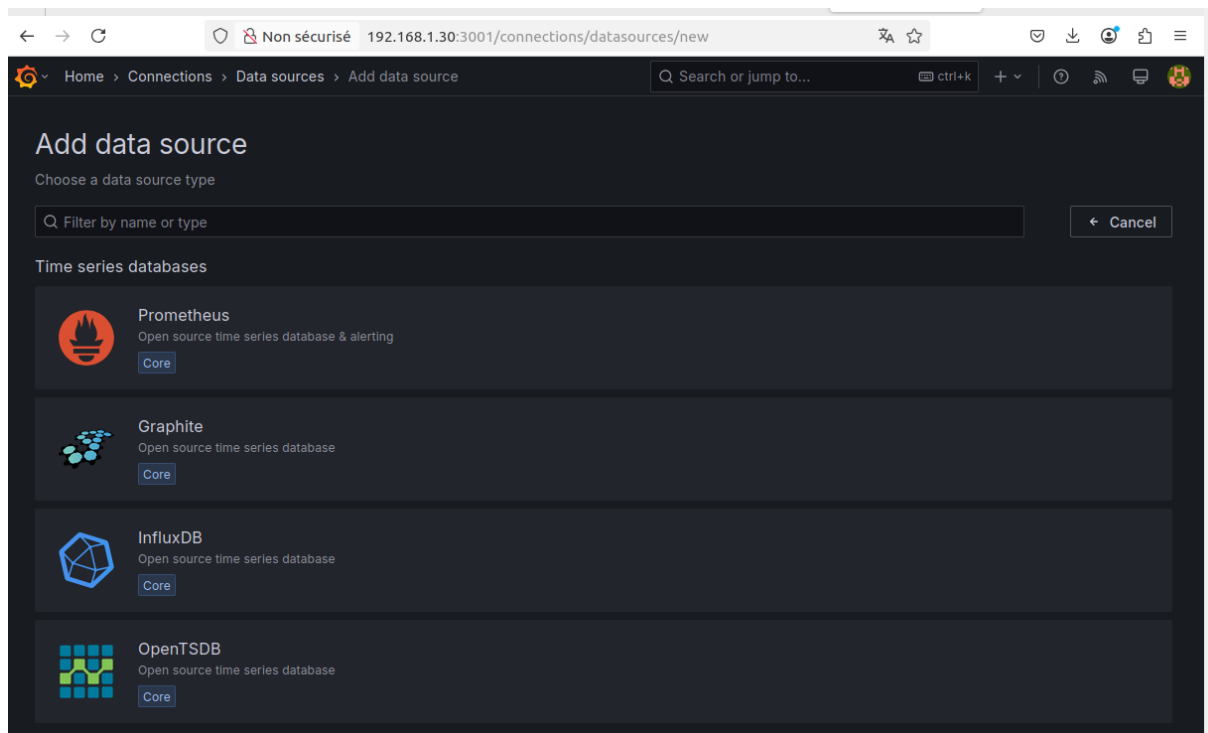


Connecter Grafana à Prometheus

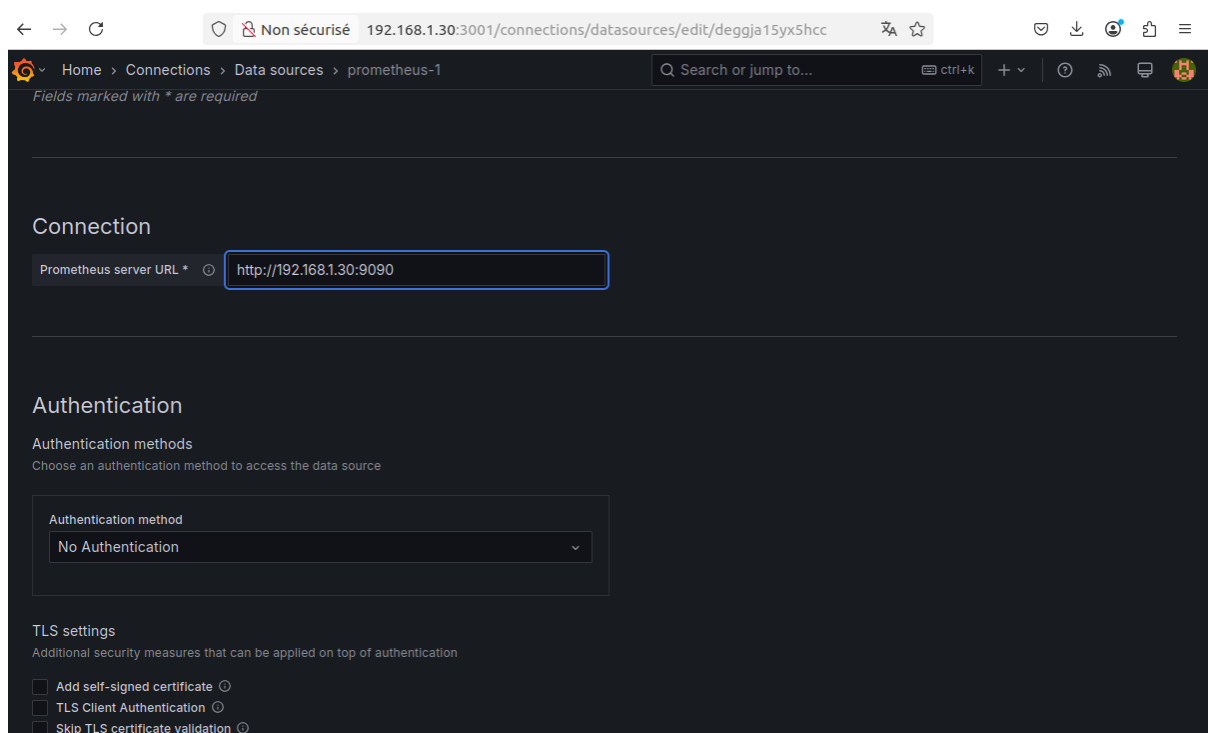
- Dans Grafana, allons dans Connexions > Sources de données



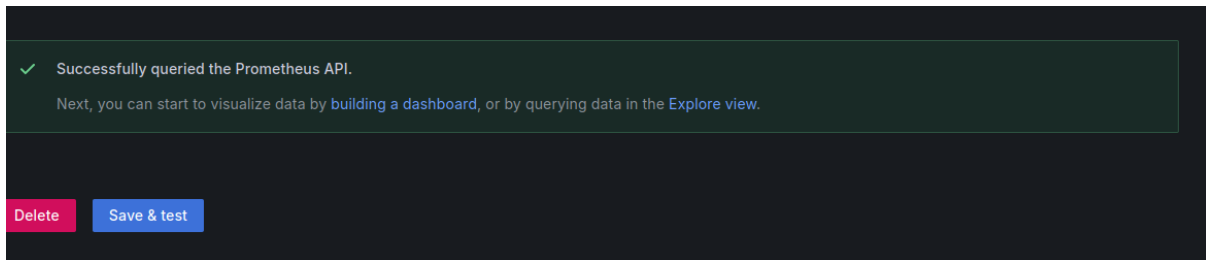
- Cliquons sur Ajouter une source de données et sélectionnons Prometheus



- **On Entre l'URL de Prometheus : <http://192.168.1.30:9090>**



- **Cliquons sur Enregistrer et tester. Si tout est correct, nous verrons un message de confirmation.**



3. Installation et configuration de Node Exporter

Télécharger Node Exporter

```
root@mbaye-VirtualBox:~# wget https://github.com/prometheus/node_exporter/releases/download/v1.9.0/node_exporter-1.9.0.linux-amd64.tar.gz
```

Décompresser

```
root@mbaye-VirtualBox:~# tar xvfz node_exporter-1.9.0.linux-amd64.tar.gz
node_exporter-1.9.0.linux-amd64/
node_exporter-1.9.0.linux-amd64/LICENSE
node_exporter-1.9.0.linux-amd64/NOTICE
node_exporter-1.9.0.linux-amd64/node_exporter
root@mbaye-VirtualBox:~#
```

On se déplace dans le dossier `node_exporter-1.9.0.linux-amd64`

```
root@mbaye-VirtualBox:~# cd node_exporter-1.9.0.linux-amd64
root@mbaye-VirtualBox:~/node_exporter-1.9.0.linux-amd64#
```

Créer un service systemd pour Node Exporter et ajoutez ce contenu

```
GNU nano 6.2 /etc/systemd/system/node_exporter.service *
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=prometheus
Group=prometheus
Type=simple
ExecStart=/usr/local/bin/node_exporter

[Install]
WantedBy=multi-user.target

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller   ^J Justifier ^/ Aller ligne
```

Sauvegardez et activez

```
root@mbaye-VirtualBox:~/node_exporter-1.9.0.linux-amd64# systemctl start node_exporter
root@mbaye-VirtualBox:~/node_exporter-1.9.0.linux-amd64# systemctl status node_exporter
● node_exporter.service - Node Exporter
   Loaded: loaded (/etc/systemd/system/node_exporter.service; enabled; vendor)
   Active: active (running) since Fri 2025-03-21 17:50:59 GMT; 9s ago
     Main PID: 96393 (node_exporter)
        Tasks: 4 (limit: 2269)
       Memory: 15.2M
          CPU: 40ms
      CGroup: /system.slice/node_exporter.service
              └─96393 /usr/local/bin/node_exporter

mar 21 17:50:59 mbye-VirtualBox node_exporter[96393]: time=2025-03-21T17:50:59>
mar 21 17:50:59 mbye-VirtualBox node_exporter[96393]: time=2025-03-21T17:50:59>
mar 21 17:50:59 mbye-VirtualBox node_exporter[96393]: time=2025-03-21T17:50:59>
mar 21 17:50:59 mbye-VirtualBox node_exporter[96393]: time=2025-03-21T17:50:59>
mar 21 17:50:59 mbye-VirtualBox node_exporter[96393]: time=2025-03-21T17:50:59>
mar 21 17:50:59 mbye-VirtualBox node_exporter[96393]: time=2025-03-21T17:50:59>
mar 21 17:50:59 mbye-VirtualBox node_exporter[96393]: time=2025-03-21T17:50:59>
mar 21 17:50:59 mbye-VirtualBox node_exporter[96393]: time=2025-03-21T17:50:59>
```

Configurer Prometheus pour scraper Node Exporter : Modifiez

/etc/prometheus/prometheus.yml

```
GNU nano 6.2 /etc/prometheus/prometheus.yml *
- 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: "node"

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

static_configs:
- targets: ["192.168.1.30:9100"]

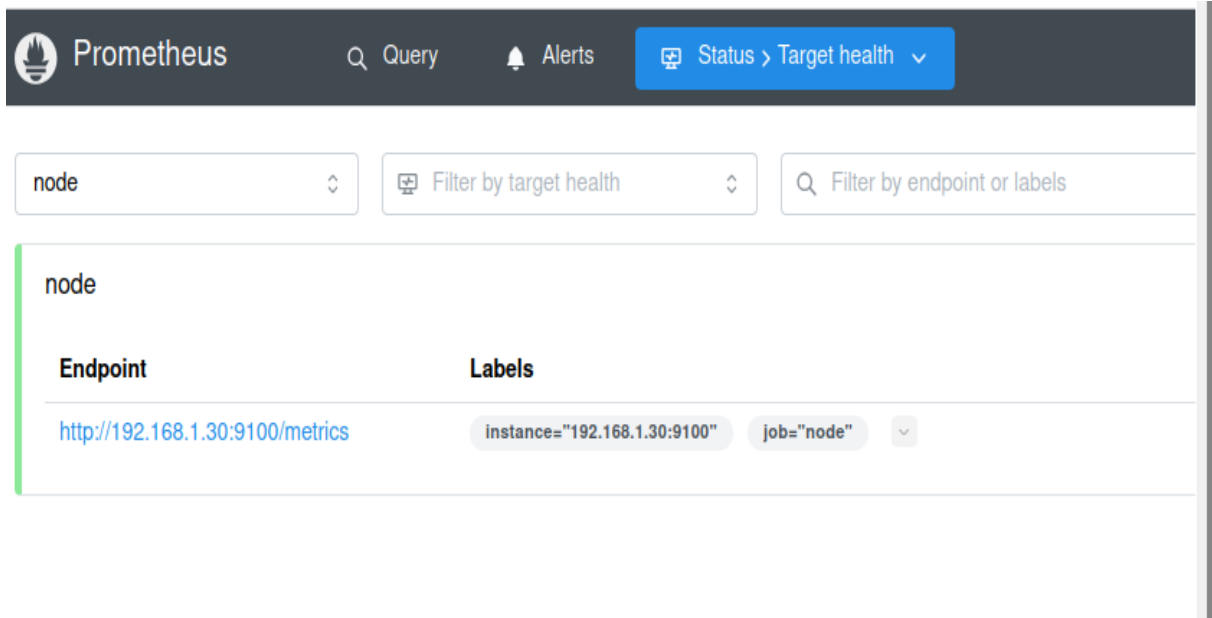
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich ^_ Remplacer ^U Coller   ^J Justifier ^/ Aller ligne
```

Redémarrez Prometheus

```
etc/systemd/system/prometheus.service.
ot@mbaye-VirtualBox:/etc/systemd/system# systemctl start prometheus
ot@mbaye-VirtualBox:/etc/systemd/system# systemctl status prometheus
prometheus.service - Prometheus
  Loaded: loaded (/etc/systemd/system/prometheus.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2025-03-20 16:32:28 GMT; 9s ago
  Main PID: 84073 (prometheus)
  Tasks: 6 (limit: 2269)
  Memory: 79.9M
  CPU: 324ms
  CGroup: /system.slice/prometheus.service
          └─84073 /usr/local/bin/prometheus --config.file=/etc/prometheus/pr

r 20 16:32:30 mbye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.91>
r 20 16:32:30 mbye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.91>
r 20 16:32:30 mbye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.91>
r 20 16:32:30 mbye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.93>
r 20 16:32:30 mbye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.93>
r 20 16:32:30 mbye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.93>
r 20 16:32:30 mbye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.94>
r 20 16:32:30 mbye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.94>
r 20 16:32:30 mbye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.94>
r 20 16:32:30 mbye-VirtualBox prometheus[84073]: time=2025-03-20T16:32:30.94>
lines 1-20/20 (END)
```

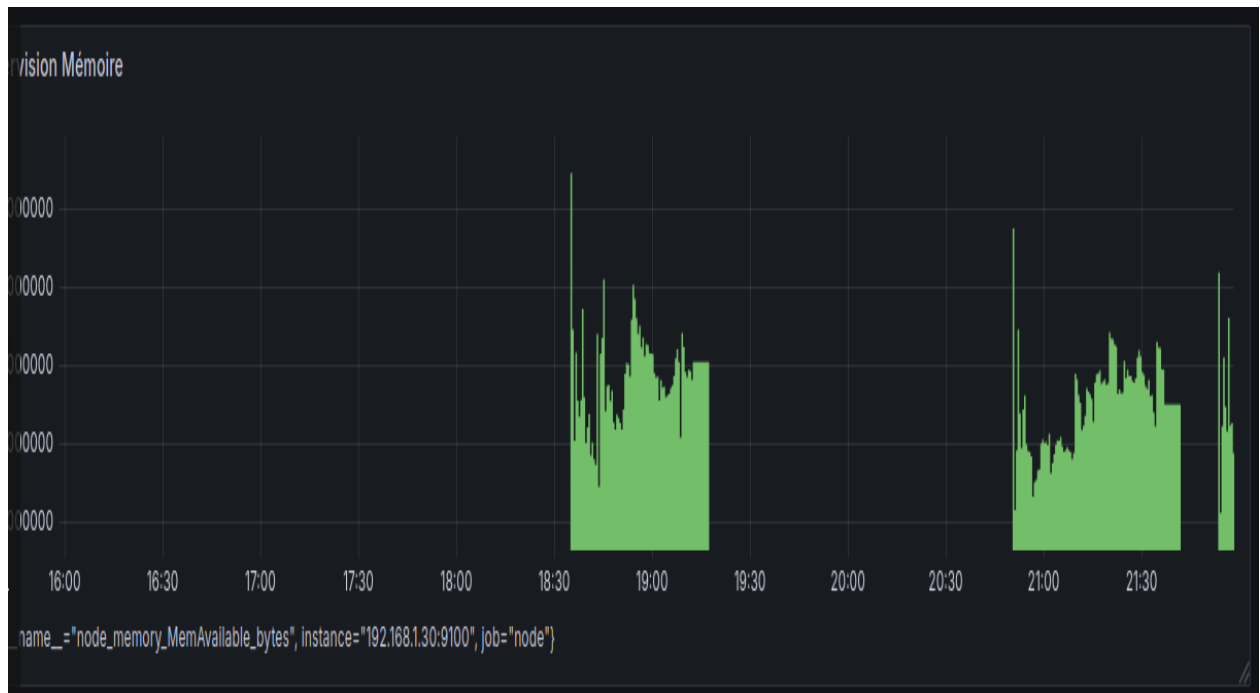
Prometheus supervisant avec succès les métriques système via Node Exporter sur 192.168.1.30:9100. Le job node est opérationnel, confirmant une configuration de surveillance efficace !



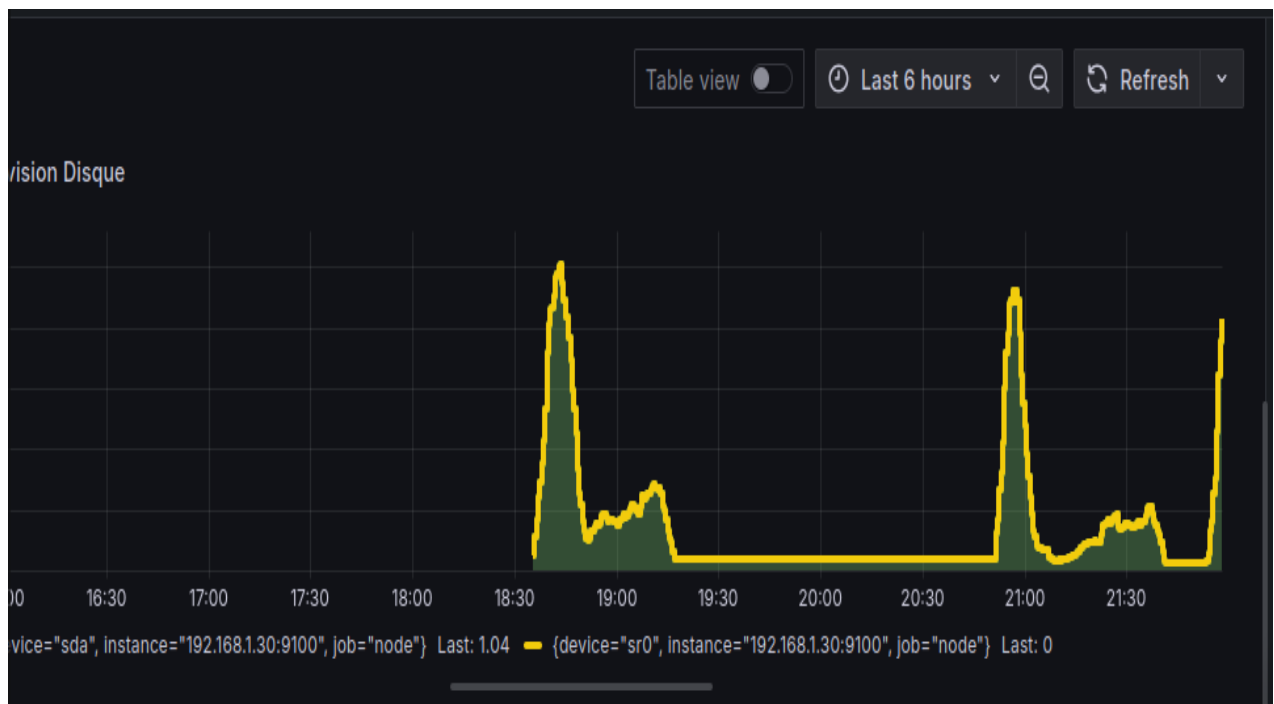
Grafana montre un graphique de l'utilisation CPU ($\text{rate}(\text{node_cpu_seconds_total}[5\text{m}])$) sur 192.168.1.30:9100 . On observe une forte hausse vers 18h30, avec des pics dans plusieurs modes (idle, iowait, user), indiquant une activité intense sur le serveur.



Grafana montre un Graphique de l'utilisation de la mémoire disponible sur 6 heures avec des pics à 18h00 et 20h30



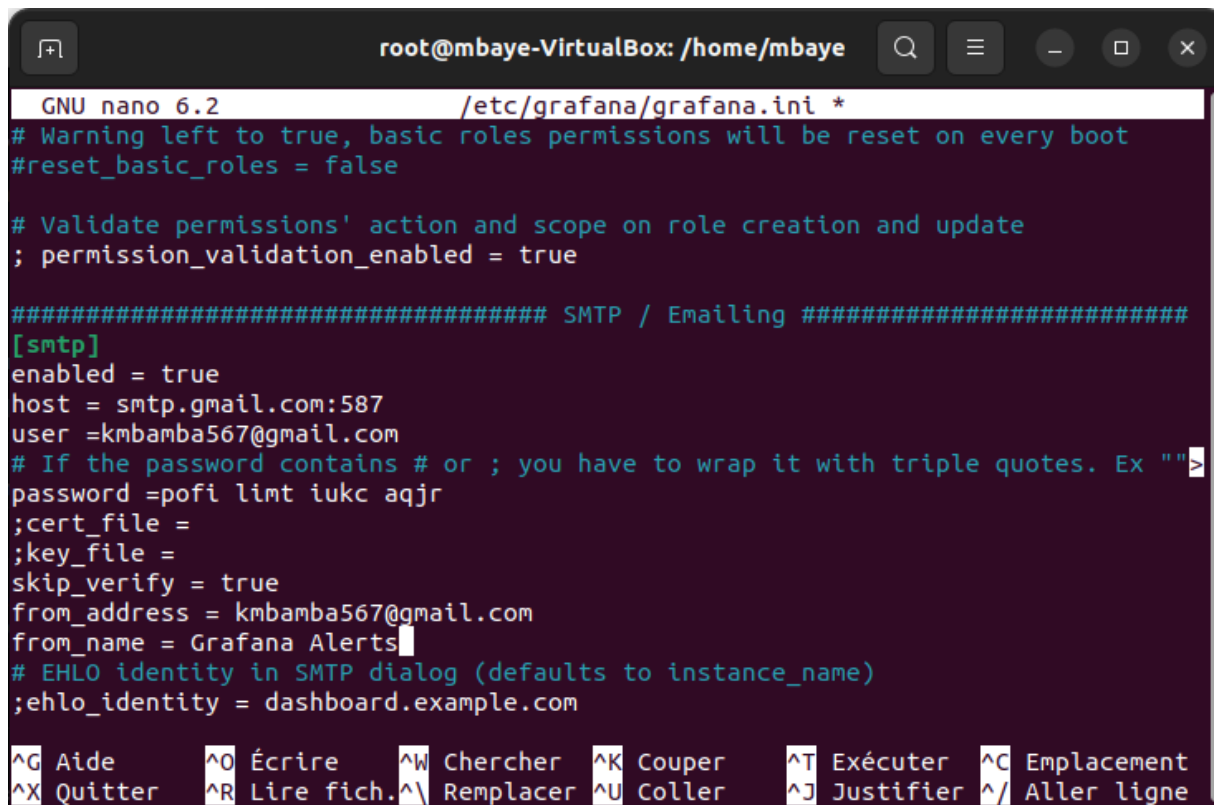
Grafana montre un Graphique de l'I/O disque avec des pics sur sda à 18h00 et 21h00.



V. Alerte par email

Pour permettre l'envoi de notifications en cas de dépassement des seuils critiques, un canal de notification par email a été configuré dans Grafana. Les paramètres SMTP ont été définis dans le fichier de configuration `/etc/grafana/grafana.ini`, en utilisant un compte Gmail avec un mot de passe d'application généré pour des raisons de sécurité. La figure 1 montre les paramètres SMTP configurés, incluant l'hôte (`smtp.gmail.com:587`), l'utilisateur, et l'adresse d'expédition.

Paramètres SMTP configurés pour l'envoi de notifications par email via Gmail.



```
root@mbaye-VirtualBox: /home/mbaye
GNU nano 6.2 /etc/grafana/grafana.ini *
# Warning left to true, basic roles permissions will be reset on every boot
#reset_basic_roles = false

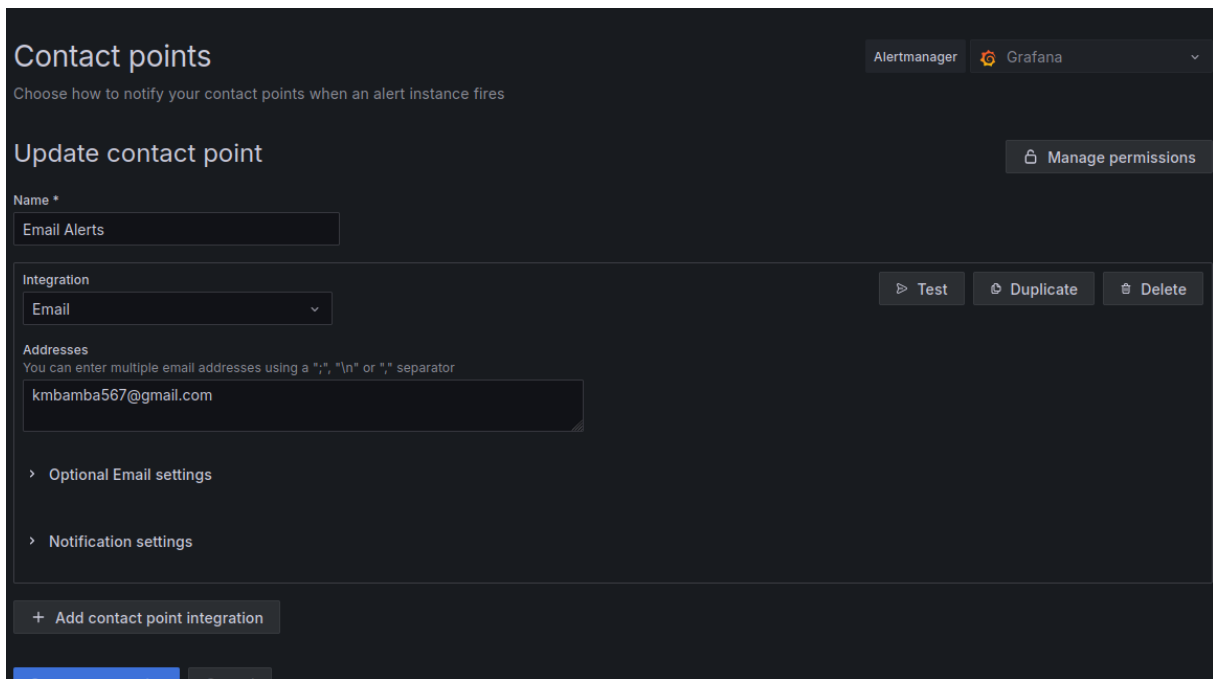
# Validate permissions' action and scope on role creation and update
; permission_validation_enabled = true

##### SMTP / Emailing #####
[smtplib]
enabled = true
host = smtp.gmail.com:587
user =kmbamba567@gmail.com
# If the password contains # or ; you have to wrap it with triple quotes. Ex "">
password =pofi limt iukc aqjr
;cert_file =
;key_file =
skip_verify = true
from_address = kmbamba567@gmail.com
from_name = Grafana Alerts
# EHLO identity in SMTP dialog (defaults to instance_name)
;ehlo_identity = dashboard.example.com

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller   ^J Justifier ^/ Aller ligne
```

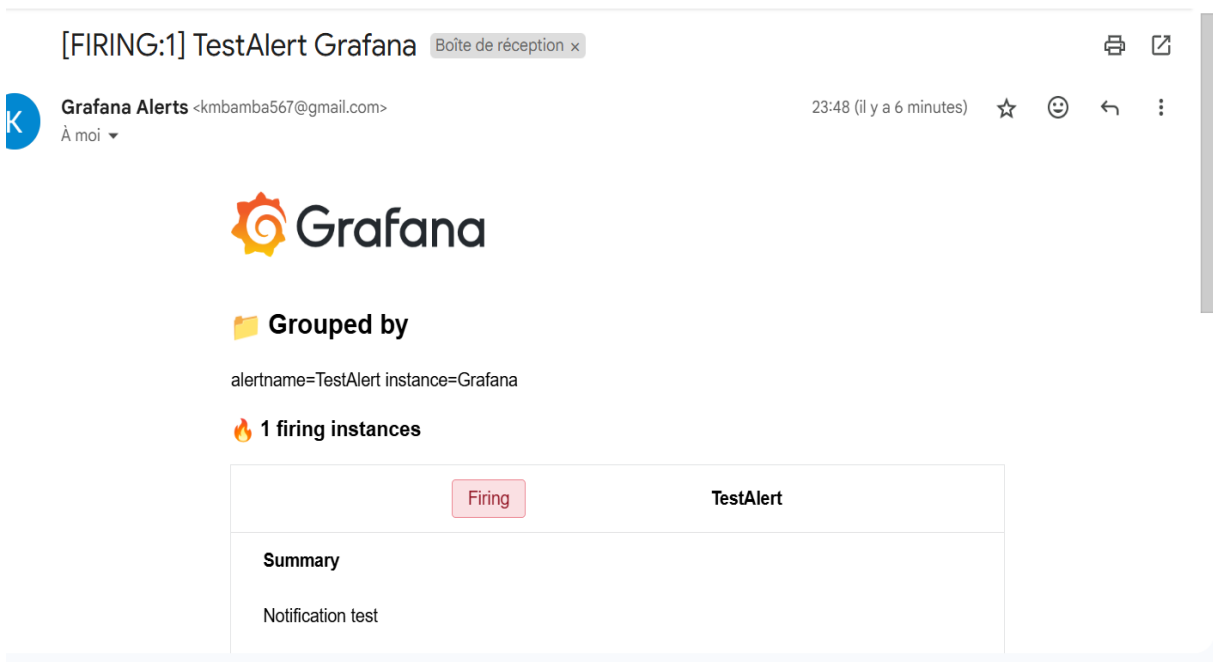
Un contact point nommé "Email Alerts" a été créé dans Grafana pour lier les alertes à ce canal de notification. La figure 2 montre la configuration de ce contact point, avec l'adresse email de destination (`kmbamba567@gmail.com`). Un test de notification a été effectué pour valider la configuration.

Contact point "Email Alerts" configuré pour envoyer des notifications à l'adresse spécifiée



La figure 3 montre l'email de test reçu, confirmant que la configuration SMTP est correcte et que Grafana peut envoyer des notifications par email. L'email indique "[FIRING:1] TestAlert Grafana" avec un résumé "Notification test"

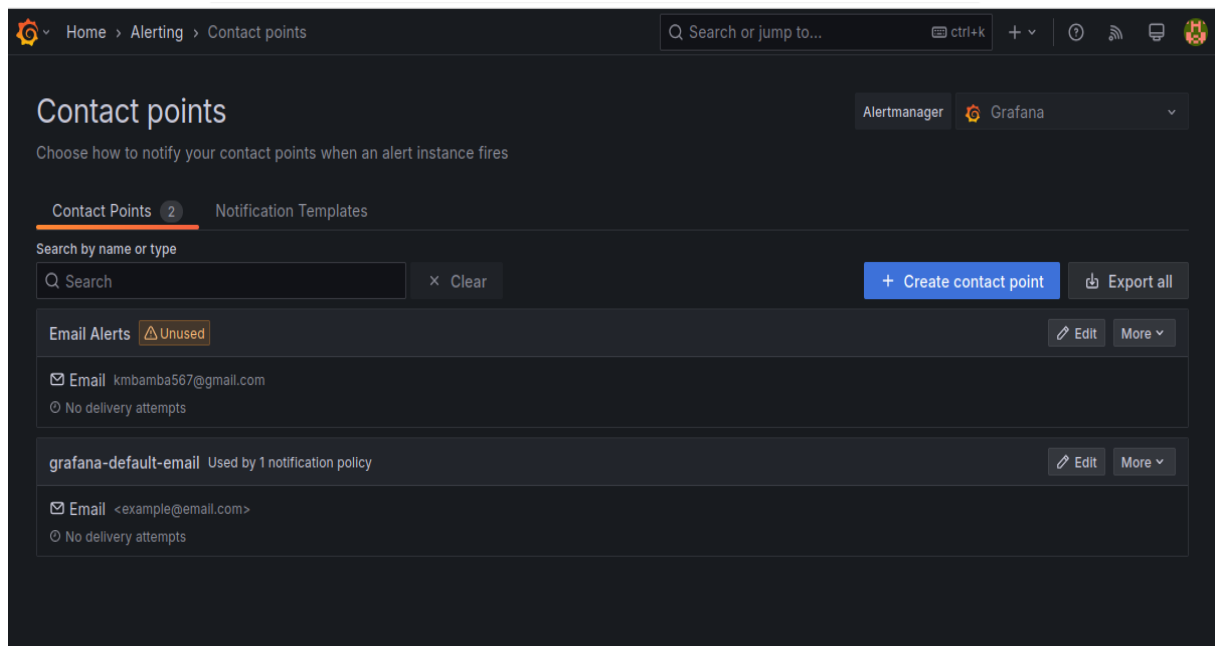
Email de test confirmant le bon fonctionnement de la configuration SMTP dans Grafana.



La figure 4 montre la liste des contact points configurés dans Grafana. Le contact point "Email Alerts" est visible, mais marqué comme "Unused" car aucune règle d'alerte n'y était encore

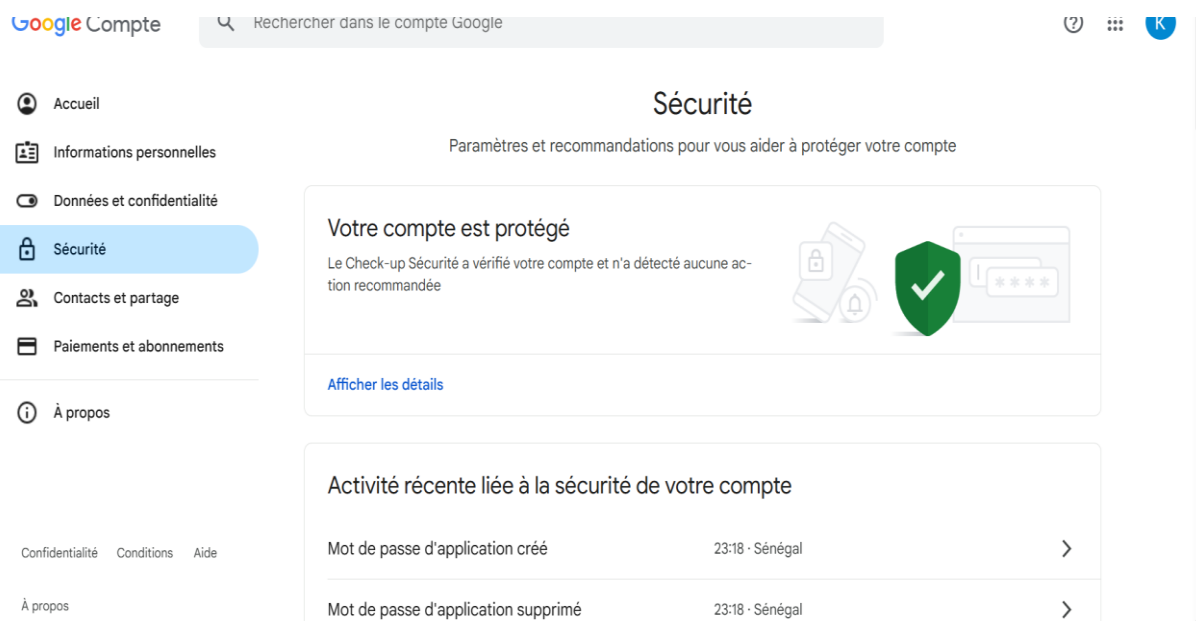
associée à ce moment. Un contact point par défaut (grafana-default-email) est également présent, utilisé par la politique de notification par défaut.

Vue d'ensemble des contact points configurés, incluant "Email Alerts" et le contact par défaut.



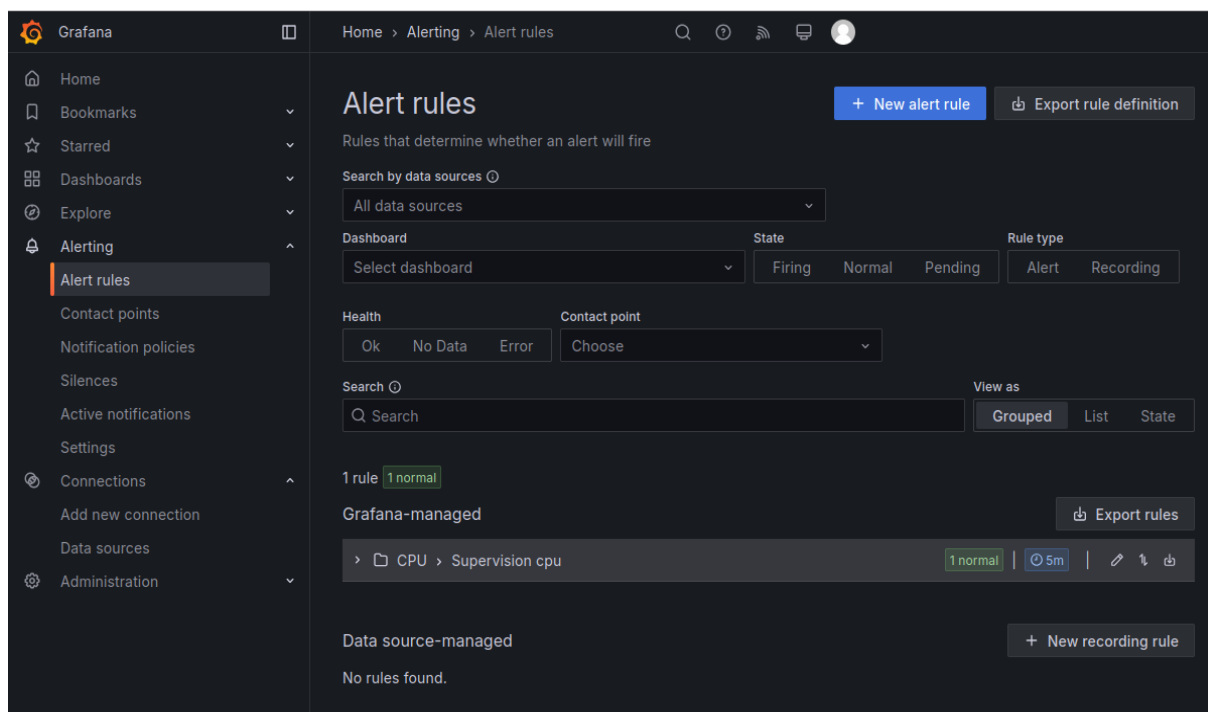
Pour utiliser Gmail comme serveur SMTP, un mot de passe d'application a été généré via les paramètres de sécurité de Google, car l'authentification à deux facteurs était activée. La figure 5 montre la page de sécurité de Google, où un mot de passe d'application nommé "grafana" a été créé.

Page de sécurité de Google montrant la création d'un mot de passe d'application pour Grafana



La figure 6 montre la section Alert Rules dans Grafana, où la règle "Supervision CPU " est visible dans le dossier "CPU" du dashboard "Supervision CPU". À ce moment, l'état est "Normal", car aucune surcharge n'a été détectée.

Règle "Supervision CPU " configurée dans le dossier "CPU", avec un état "Normal



Après un certain temps, l'utilisation CPU de la machine (192.168.1.56:9100) a dépassé 80%, déclenchant l'alerte

La figure 7 montre l'email reçu, indiquant "[FIRING:4] cpu CPU (0 192.168.1.56:9100 node)" avec un résumé "supervision cpu". Cela confirme que l'alerte a été correctement déclenchée et que la notification par email fonctionne comme prévu.

Email reçu confirmant une utilisation CPU élevée sur la machine surveillée.



The screenshot shows an email interface with a header bar containing navigation icons and the text "1 sur 1228". The email subject is "[FIRING:4] cpu CPU (0 192.168.1.56:9100 node)" with a "Boîte de réception x" label. The sender is "Grafana Alerts <kmbamba567@gmail.com>" with a "À moi" dropdown. The time is "22:34 (il y a 8 minutes)". The email body features the Grafana logo, a breadcrumb "CPU > cpu", and "4 firing instances" with a flame icon. Below this is a table with one row:

Firing	cpu	View alert
Summary supervision cpu		
Description		

Conclusion

La mise en place de **Prometheus**, **Grafana** et **Node Exporter** sur notre serveur Ubuntu (192.168.1.30) a permis de créer une solution de supervision robuste et efficace. Prometheus collecte les métriques système via Node Exporter (port 9100), comme validé par l'interface "Targets" affichant le job node actif. Grafana, configuré sur le port 3001, transforme ces données en visualisations claires, comme le graphique d'utilisation CPU qui a révélé des pics d'activité vers 17h30. Ensemble, ces outils offrent une vue complète et en temps réel des performances, facilitant une gestion proactive de l'infrastructure. Des alertes ont été configurées pour surveiller l'utilisation CPU, avec des notifications par email en cas de dépassement des seuils critiques. En termes de perspectives on peut ajouter des alertes pour la mémoire et le disque, ou intégrer Alertmanager pour une gestion plus avancée des notifications.