

Prometheus -

Node_Exporter - Grafana

vendredi 29 août 2025 10:34

Prometheus : C'est un multi-outil de monitoring et d'alerting qui peut servir également de base de données type TSDB (Time Series DataBase).

Node Exporter : Viens collecter des métriques système (variante de Telegraf en gros)

Grafana : voir [Grafana](#)

Partie Prometheus :

Tout d'abord on va venir créer un dossier "**prometheus**" dans le répertoire **/opt** de notre Linux.

Ensuite on vient télécharger la **dernière version** de prometheus avec **wget** qu'on peut retrouver ici :
<https://github.com/prometheus/prometheus/releases/>

J'ai la version "**prometheus-3.5.0.linux-amd64**", donc il faudra adapter les fichiers de configuration à la version qu'on a.

On dézippe le fichier avec : **tar -xzvf -C /opt/prometheus**

On aimerais que **Prometheus** tourne en tant que **service**.

Donc on ajoute le user prometheus : **useradd -s /sbin/false prometheus**

Et on donne les droits à prometheus : **[root@almalinux opt]# chmod 644 prometheus**
[root@almalinux opt]# chown -Rf prometheus:prometheus /opt

On ajoute prometheus en tant que service dans systemd :
nano /etc/systemd/system/prometheus.service

On y ajoute dedans cette **configuration** (à adapter selon sa version de prometheus) :

[Unit]

*Description=Serveur Prometheus
Wants=network-online.target
After=network-online.target*

[Service]

*User=prometheus
Group=prometheus
Type=simple
ExecStart=/opt/prometheus/prometheus-3.5.0.linux-amd64/prometheus |
--config.file=/opt/prometheus/prometheus-3.5.0.linux-amd64/prometheus.yml |
--storage.tsdb.path=/opt/prometheus/prometheus-3.5.0.linux-amd64/data |
--web.console.templates=/opt/prometheus/prometheus-3.5.0.linux-amd64/consoles |
--web.console.libraries=/opt/prometheus/prometheus-3.5.0.linux-amd64/console_libraries*

[Install]

WantedBy=multi-user.target

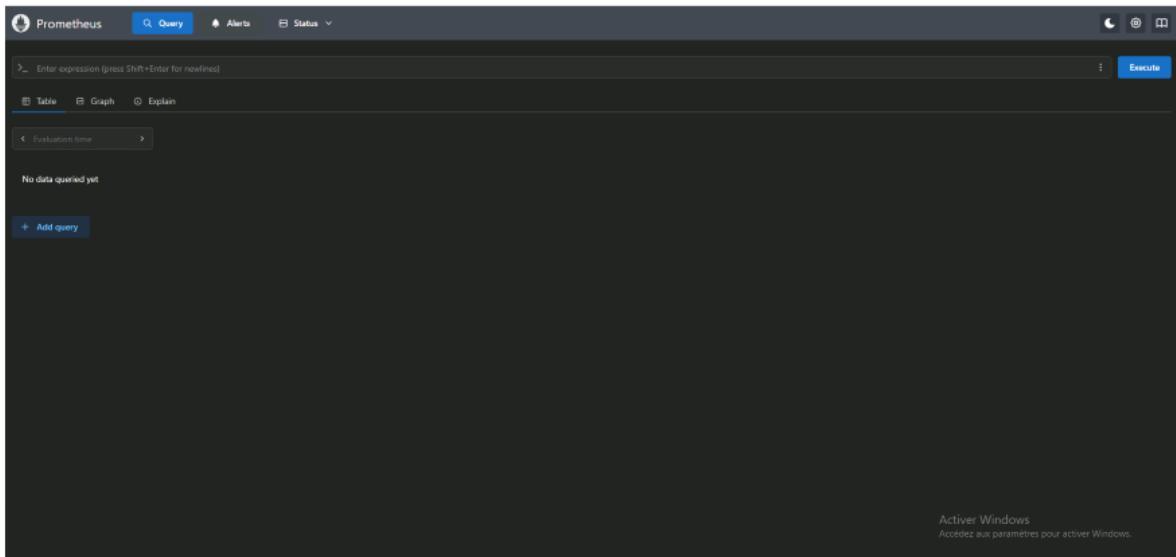
Prometheus est installé en tant que service ! 

On **recharge** les conf de systemctl :
systemctl reload-daemon

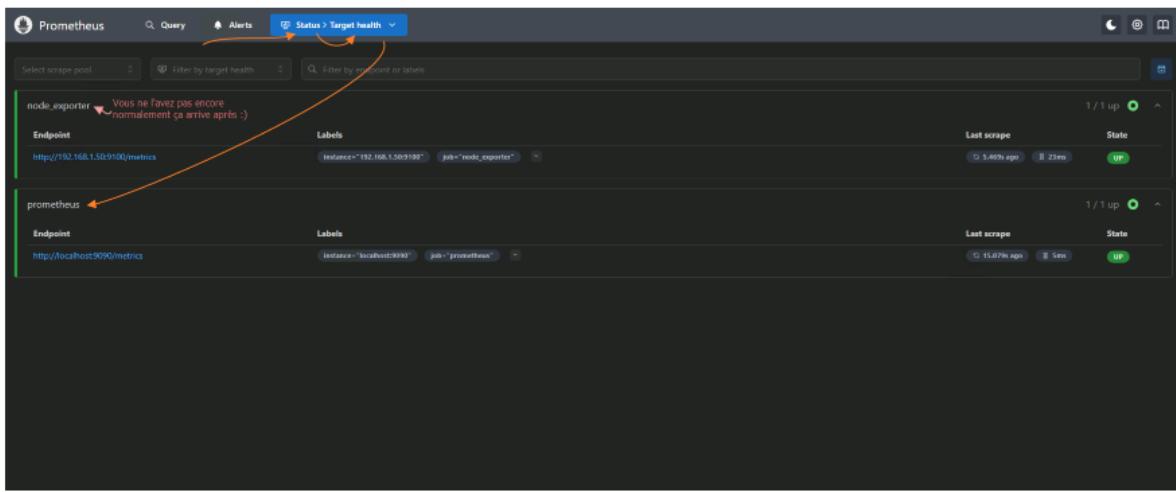
On **démarre** prometheus et on vérifie qu'il tourne bien :
systemctl start prometheus
systemctl status prometheus

Prometheus tourne par défaut sur le **port 9090**.
On s'y connecte via un **navigateur** : <http://localhost:9090>

Et de mémoire, on configure les logins pour ensuite atterrir ici :



On peut vérifier que tout fonctionne correctement en allant dans "Status → Target health" :



Prometheus done ✓

Partie Node_Exporter :

On recommence en commençant par créer un dossier "**node_exporter**" dans **/opt/** de notre Linux.

Puis on fait un **wget** pour télécharger la **dernière version** de Node_Exporter qu'on retrouve ici :
https://github.com/prometheus/node_exporter/releases

On dézippe le fichier avec : **tar -xzvf -C /opt/node_exporter**

Et pareil, on aimerais que Node_Eporter tourne en tant que **service**.

Donc on ajoute prometheus en tant que service dans systemd :
nano /etc/systemd/system/node_exporter.service

On y ajoute dedans cette **configuration** (à adapter selon sa version de Node_Exporter) :

[Unit]

```
Description=node_exporter
Wants=network-online.target
After=network-online.target
```

[Service]

```
User=prometheus
Group=prometheus
Type=simple
ExecStart=/opt/node_exporter/node_exporter-1.9.1.linux-amd64/node_exporter
```

[Install]

```
WantedBy=multi-user.target
```

Node_Exporter est installé en tant que service ! 

On recharge les conf de systemctl : `systemctl reload-daemon`

On démarre prometheus et on vérifie qu'il tourne bien :

`systemctl start node_exporter`

`systemctl status node_exporter`

Node_Exporter tourne par défaut sur le port 9100

Node_Exporter done ✓

Il faut maintenant ajouter Node_Exporter à Prometheus.

Pour cela, il faut modifier le fichier "`prometheus.yml`" et y ajouter un "job".

`nano /opt/prometheus/prometheus-3.5.0.linux-amd64/prometheus.yml`

On vient y ajouter le job (à adapter à sa machine) node_exporter sous le job "prometheus" qui se trouvent sous "scrape_configs" :

```
GNU nano 5.6.1          /opt/prometheus/prometheus-3.5.0.linux-amd64/prometheus.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: "node_exporter"
    static_configs:
      - targets: ['192.168.1.50:9100']
```

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

On redémarre le service Prometheus :

`systemctl restart prometheus`

Puis on vient vérifier les cibles dans Prometheus :

The screenshot shows the Prometheus web interface with two targets listed under the 'Targets' section:

- node_exporter**:
 - Endpoint: `http://192.168.1.50:9100/metrics`
 - Labels:
 - instance: "192.168.1.50:9100"
 - job: "node_exporter"
 - Last scrape: 3.183s ago
 - State: UP
- prometheus**:
 - Endpoint: `http://localhost:9090/metrics`
 - Labels:
 - instance: "localhost:9090"
 - job: "prometheus"
 - Last scrape: 12.294s ago
 - State: UP

An orange arrow points from the text "- targets: ['localhost:9100']" in the code block above to the "node_exporter" target in the UI. Another orange arrow points from the text "Il est bien là :)" in the UI to the "node_exporter" target.

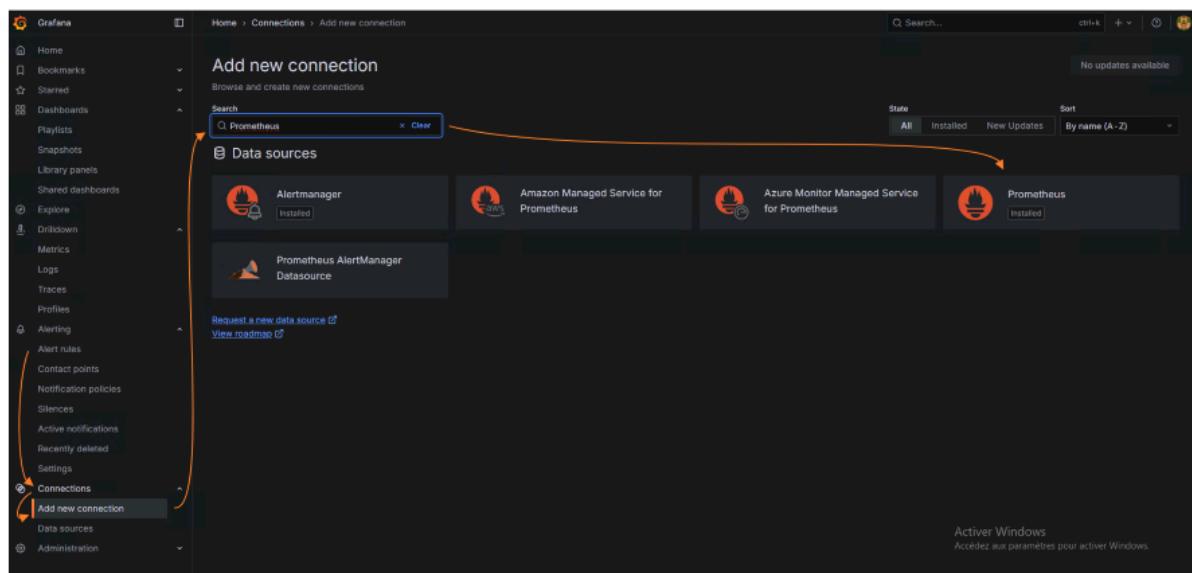
Node_Exporter done ✓

Partie Grafana :

Voir installation de Grafana : [Installation de Grafana](#)

On va venir désormais intégrer Prometheus dans Grafana.

Aller sur **Grafana** et suivre cet exemple :



Cliquer sur "Add new data source" en haut à gauche de l'écran.

Puis :

The screenshot shows the 'Connections > Data sources > prometheus' page in Grafana. The 'Name' field is set to 'prometheus'. The 'Prometheus server URL' is 'http://192.168.1.50:9090'. Under 'Authentication', 'No Authentication' is selected. In the 'TLS settings' section, the 'Skip TLS certificate validation' checkbox is checked. A yellow arrow points from the text 'Ensuite, cochez la case "Ignorer la vérification du certificat TLS"' to the 'Skip TLS certificate validation' checkbox.

Enfin, en bas de page, cliquer sur "**Save & test**" et ce message est censé apparaître :

The screenshot shows the same 'Connections > Data sources > prometheus' page after saving and testing. A success message at the bottom states 'Successfully queried the Prometheus API.' with a green checkmark. Below the message, there are two buttons: 'Delete' and 'Save & test'. A yellow arrow points from the text 'Cliquez sur le bouton "Enregistrer et tester"' to the 'Save & test' button. To the right of the message, there is a small 'Activate Windows' watermark.

Maintenant que Prometheus est intégré.
On souhaite voir les données récoltées.

Pour cela on suit ces étapes :

The screenshot shows the Grafana interface with the sidebar open. The 'Dashboards' menu item is selected. In the main area, there is a table titled 'Dashboards' with two entries: 'Node Exporter Full' and 'Processeur'. At the top right of the main area, there is a 'New' button with a dropdown menu containing 'New dashboard', 'New folder', and 'Import'. A blue arrow points from the 'Import' option in the dropdown to the 'Import' section of the dialog.

Et vous pouvez importer un Dashboard déjà tout fait pour tester, j'ai pris le "1860" :

The screenshot shows the 'Import dashboard' dialog in Grafana. The sidebar on the left is visible, showing the 'Dashboards' menu item is selected. The main area has a title 'Import dashboard' and a sub-instruction 'Import dashboard from file or Grafana.com'. Below this is a dashed box with an upward arrow icon labeled 'Upload dashboard JSON file' and the text 'Drag and drop here or click to browse Accepted file types: .json, .txt'. To the right of this is a search bar containing '1860' with a blue arrow pointing to the 'Load' button next to it. Below the search bar is a section titled 'Import via dashboard JSON model' containing a JSON snippet. At the bottom of the dialog are 'Load' and 'Cancel' buttons, with a blue arrow pointing to the 'Load' button.

Et vous atterrissez normalement sur quelque chose comme ça :



Grafana done ✓