Projet de Monitoring Local avec Prometheus, Windows Exporter et Grafana

1. Introduction

L'objectif de ce projet est de mettre en place une solution de supervision locale pour suivre les ressources système de mon PC personnel. Grâce à l'installation de **Prometheus**, **Windows Exporter** et **Grafana**, j'ai pu collecter, visualiser et surveiller en temps réel les métriques de base telles que le **CPU**, la **mémoire vive** et l'utilisation du **disque**.

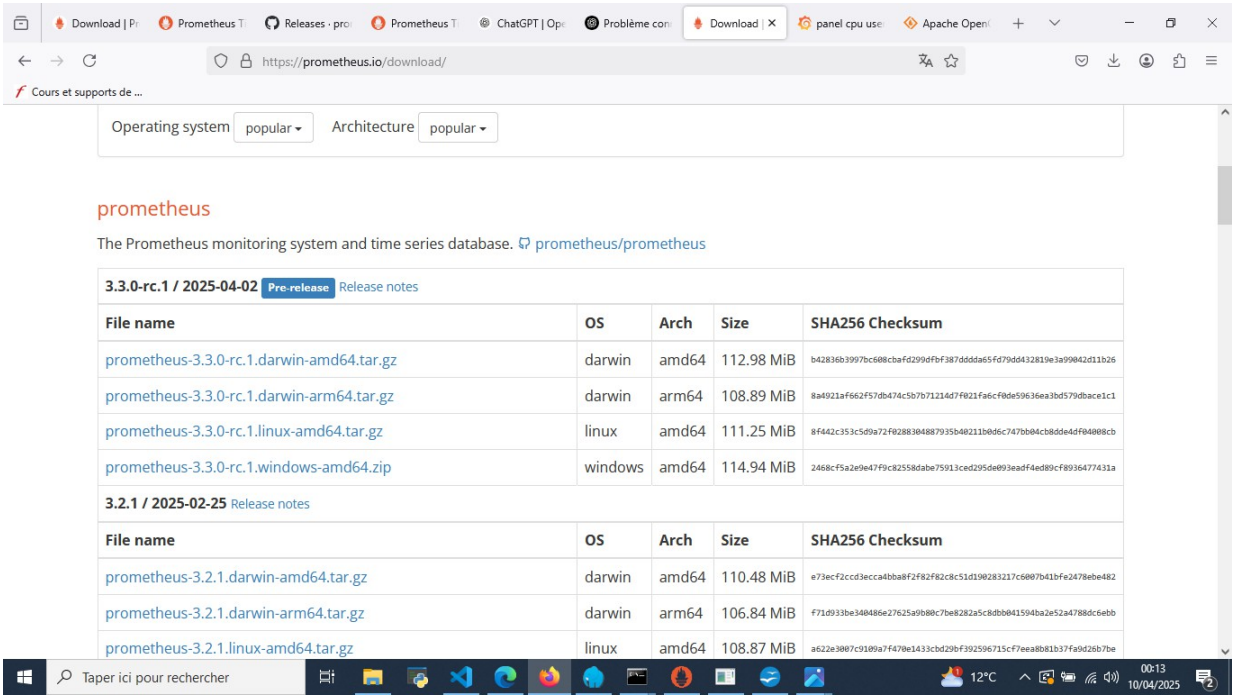
2. Environnement de travail

- **Système d'exploitation** : Windows 10 / 11
- **Outils utilisés** :
    - Prometheus
    - Windows Exporter
    - Grafana
- **Type d'installation** : locale (tout fonctionne sur ma propre machine)

3. Installation et configuration

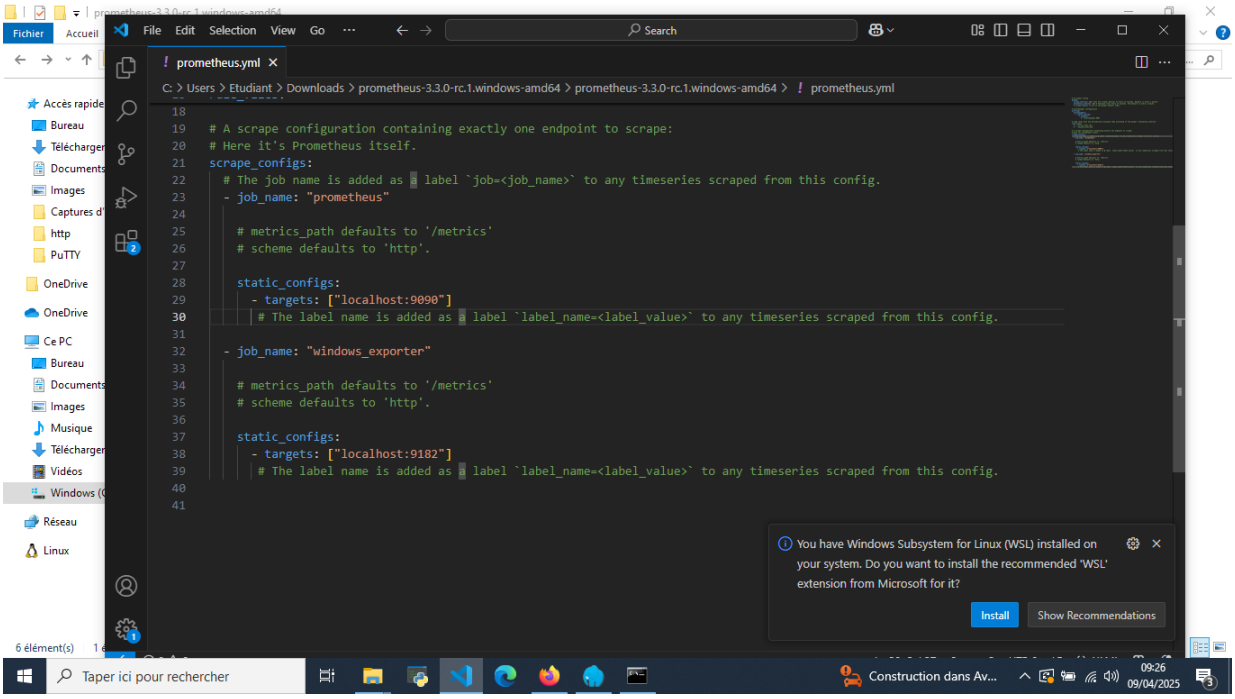
3.1 Installation de Prometheus

1. Téléchargement depuis le site officiel : https://prometheus.io



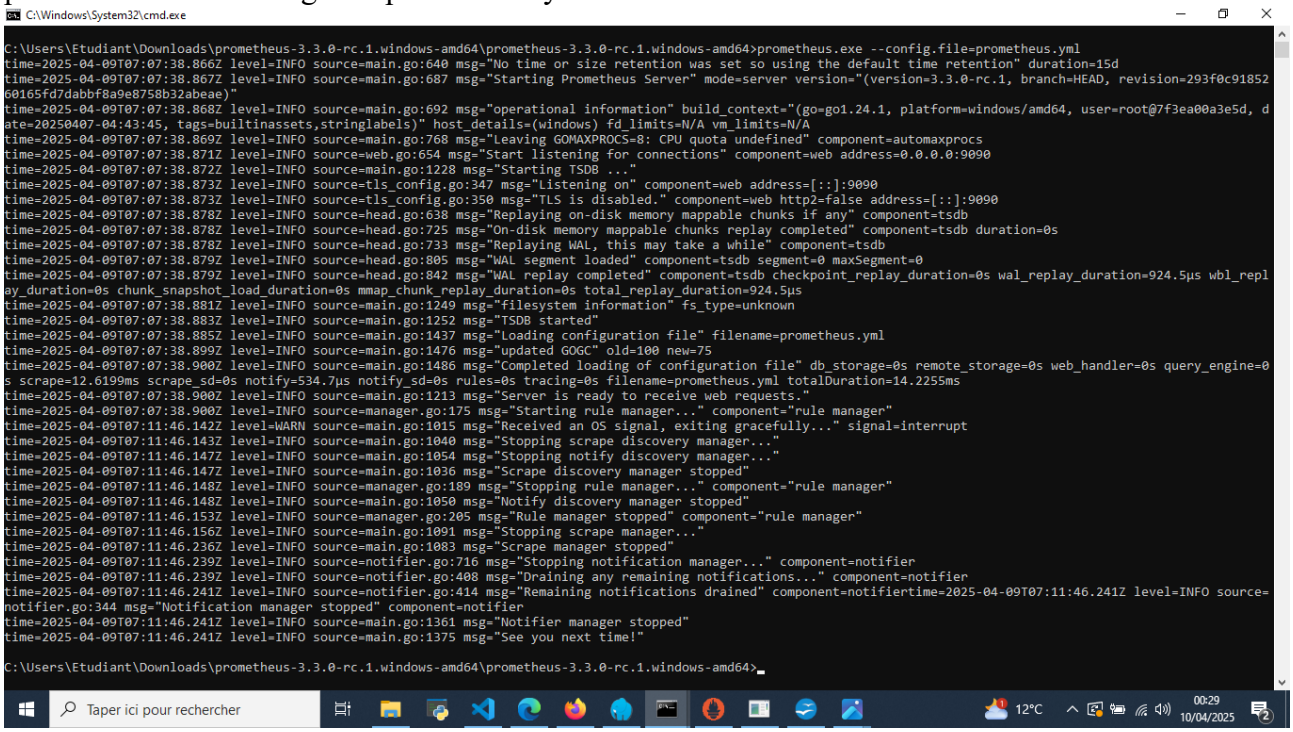
2. Extraction de l'archive dans un dossier `C:\Prometheus`

3. Configuration du fichier `prometheus.yml` :



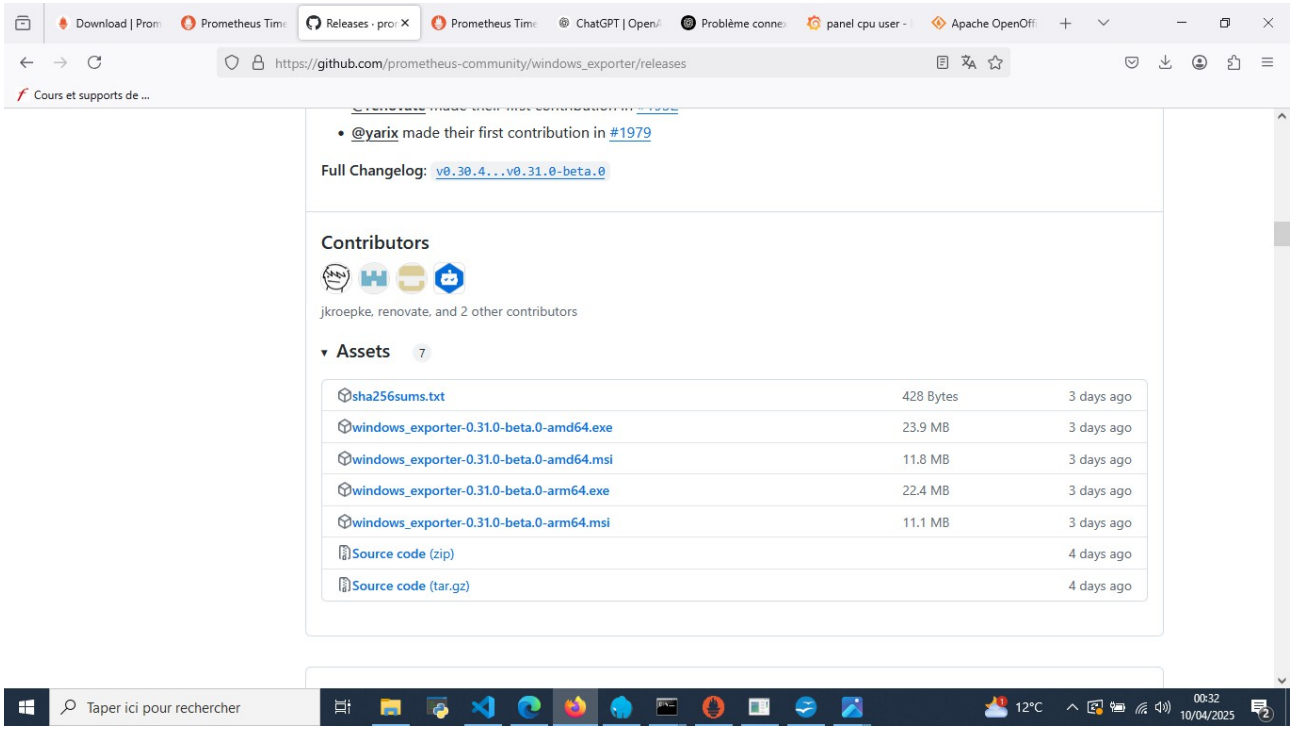
4. Lancement avec la commande :

prometheus.exe --config.file=prometheus.yml

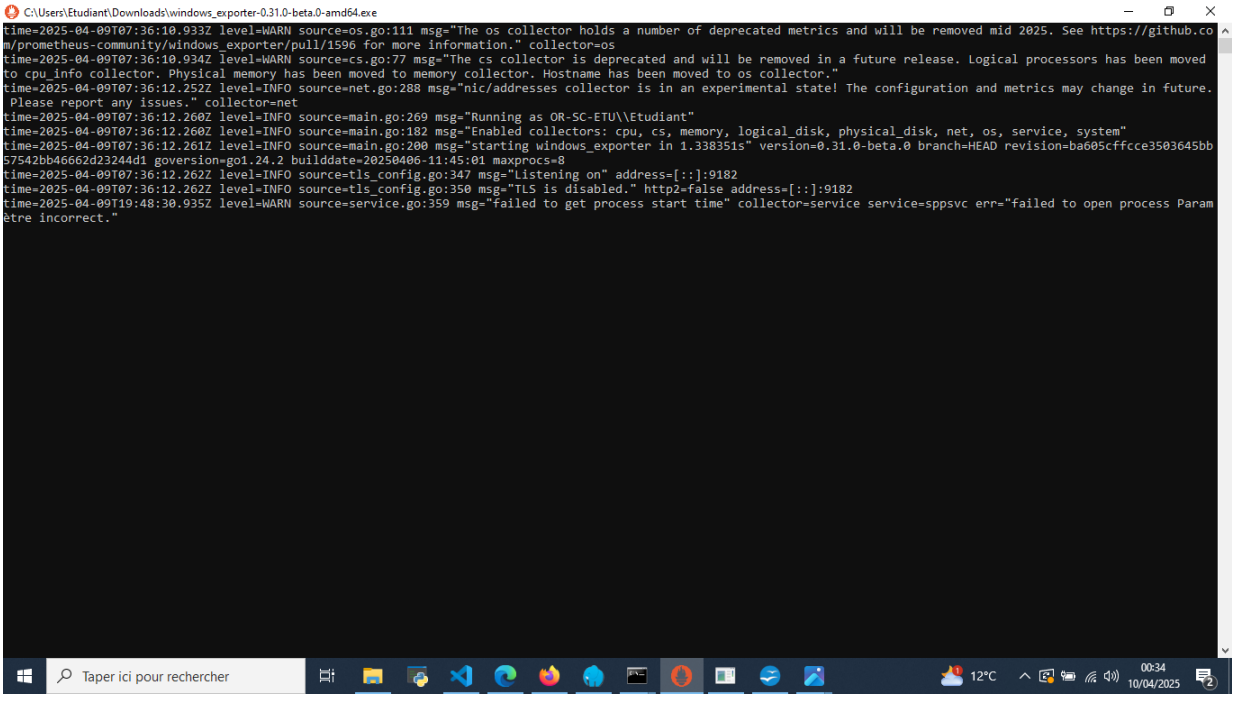


3.2 Installation de Windows Exporter

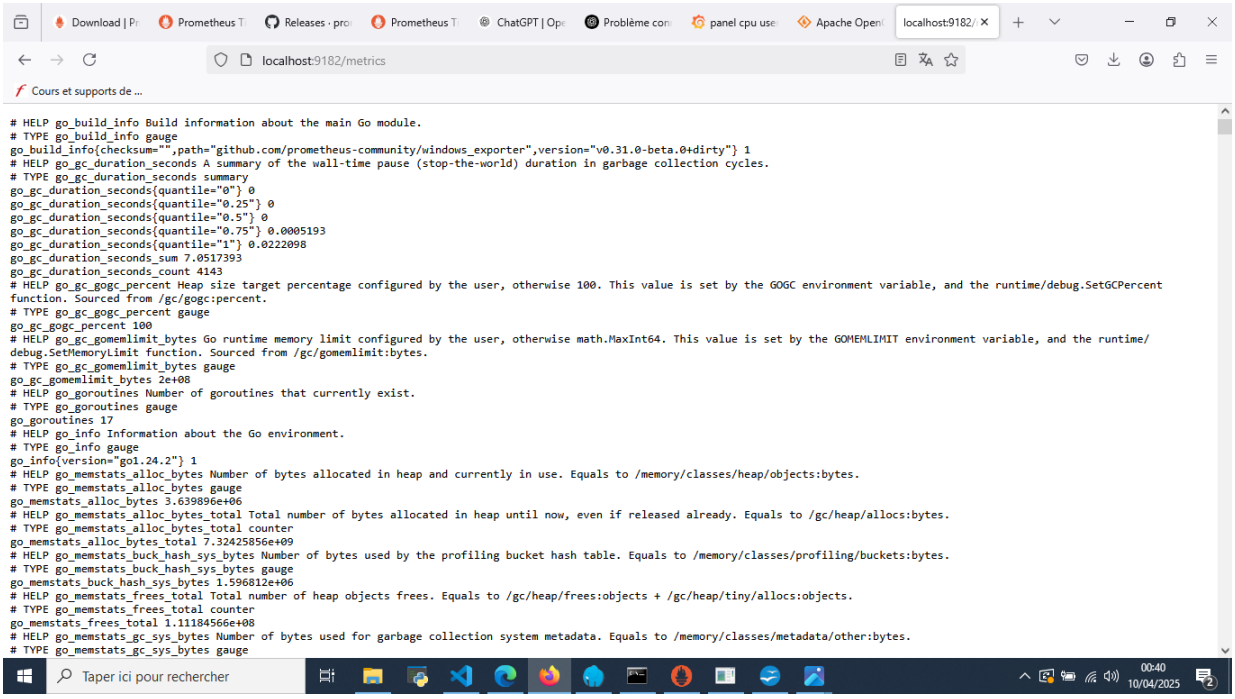
1. Téléchargement depuis GitHub :
   https://github.com/prometheus-community/windows_exporter/releases

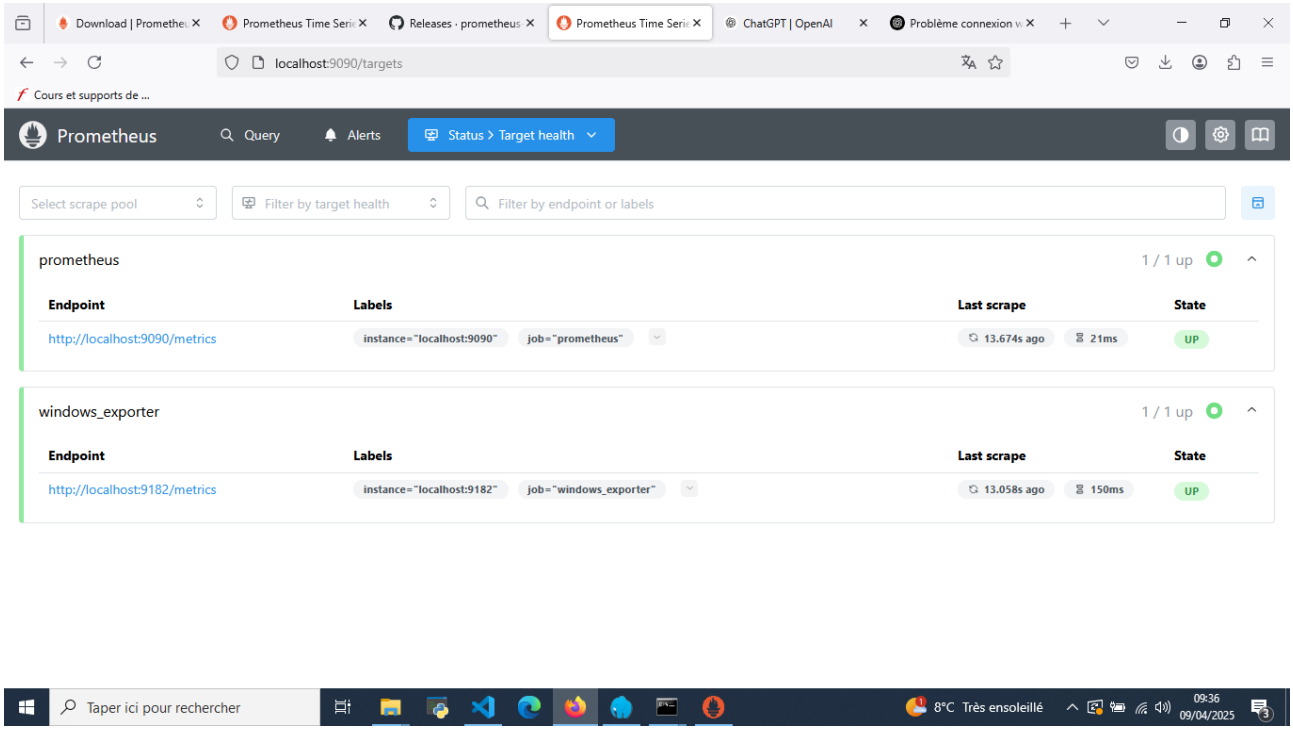


2. Lancement direct de `windows_exporter.exe`



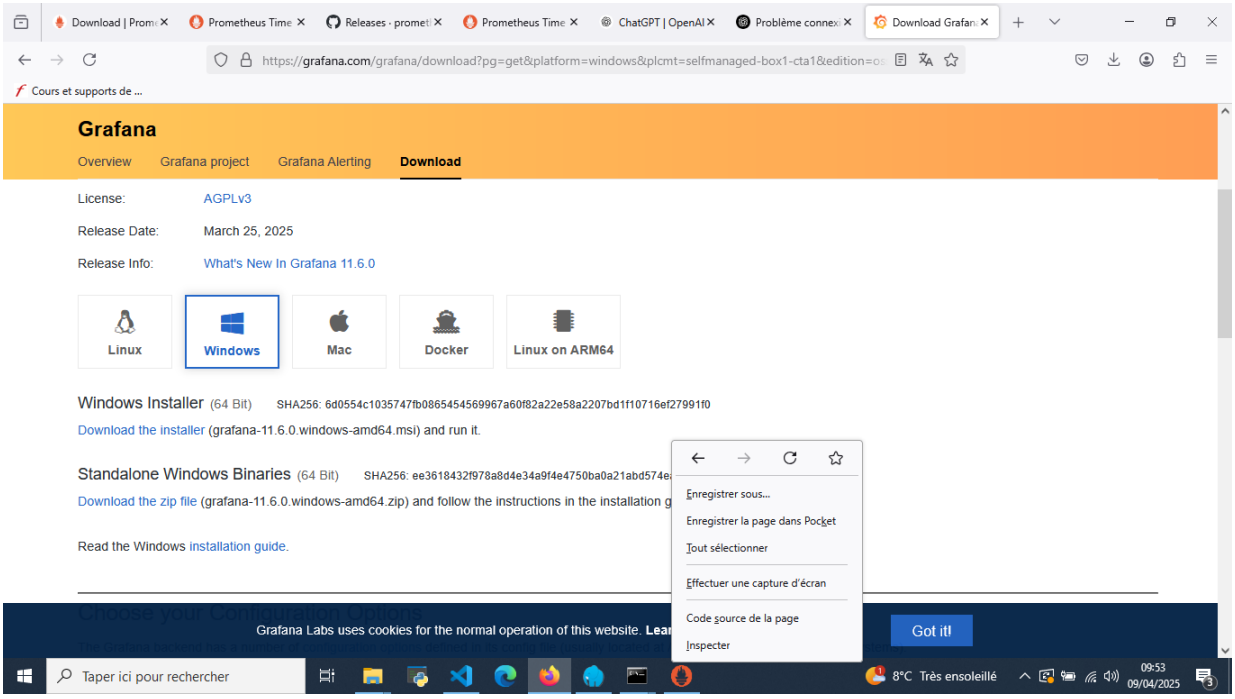
3. Vérification de l’accès à l’URL :



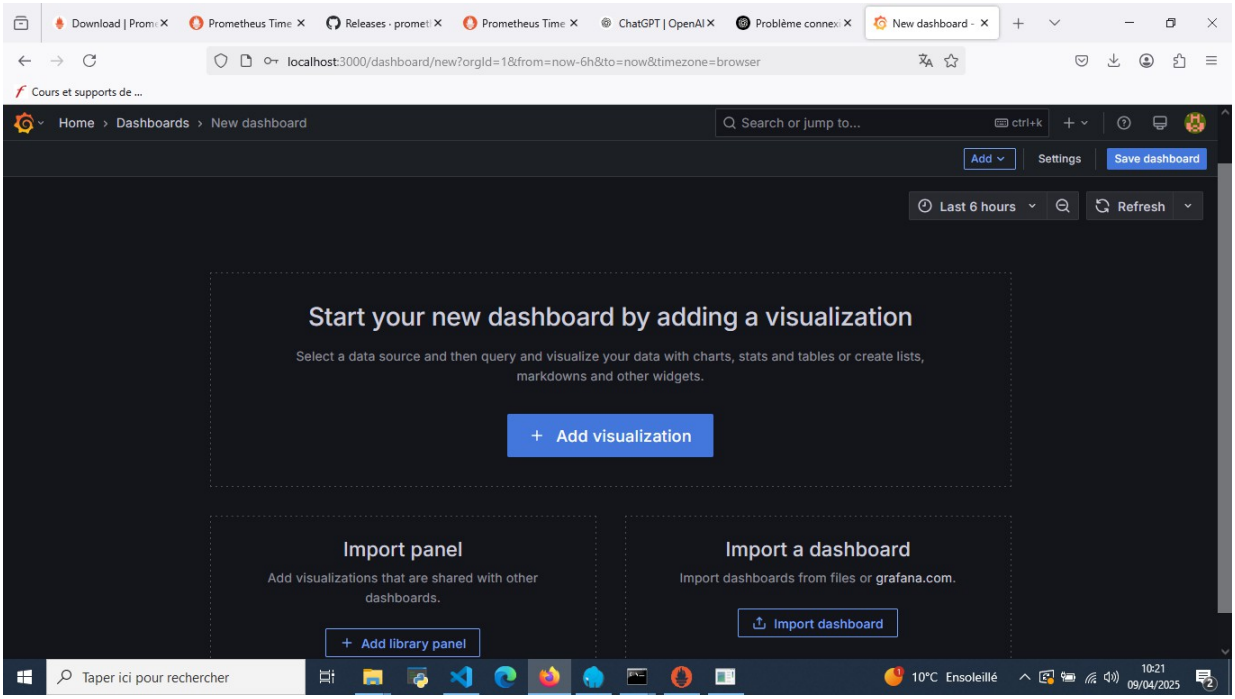


3.3 Installation de Grafana

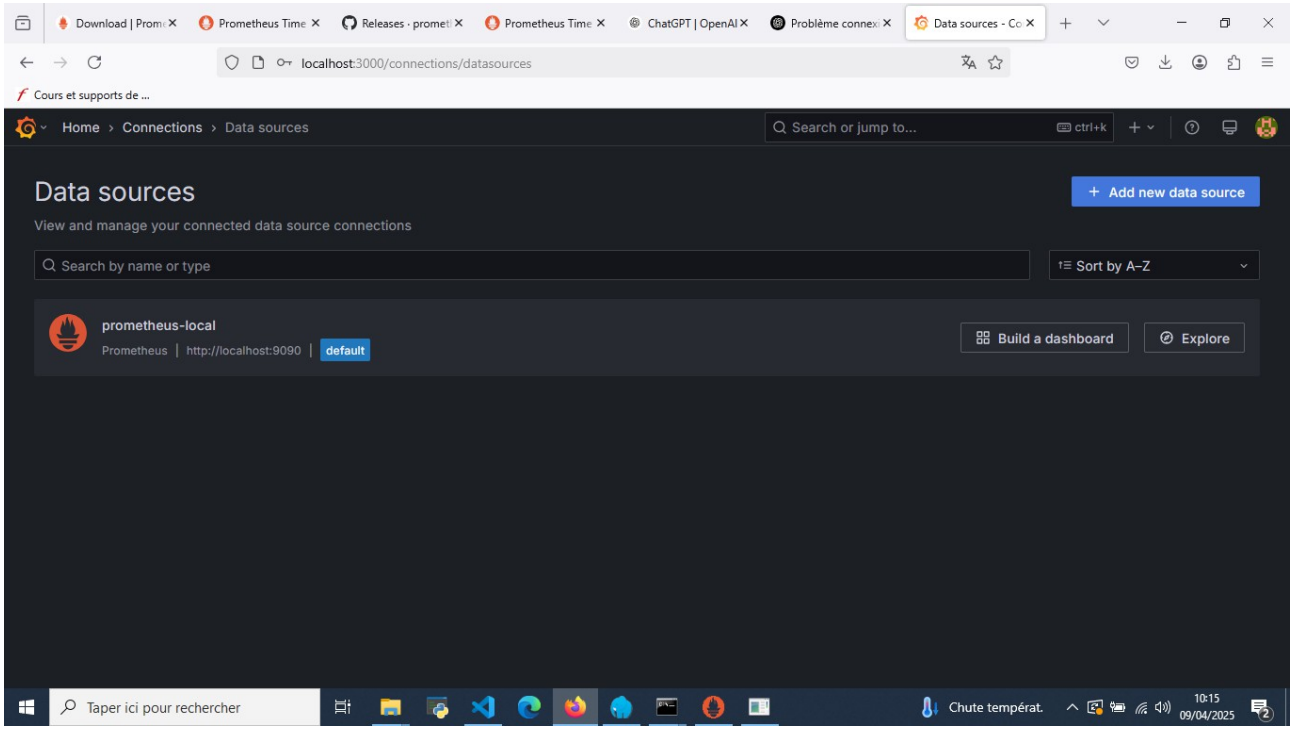
1. Téléchargement depuis https://grafana.com



2. Installation standard sous Windows
3. Accès à Grafana via http://localhost:3000



4. Ajout de Prometheus comme **Data Source** via `Settings > Data Sources`

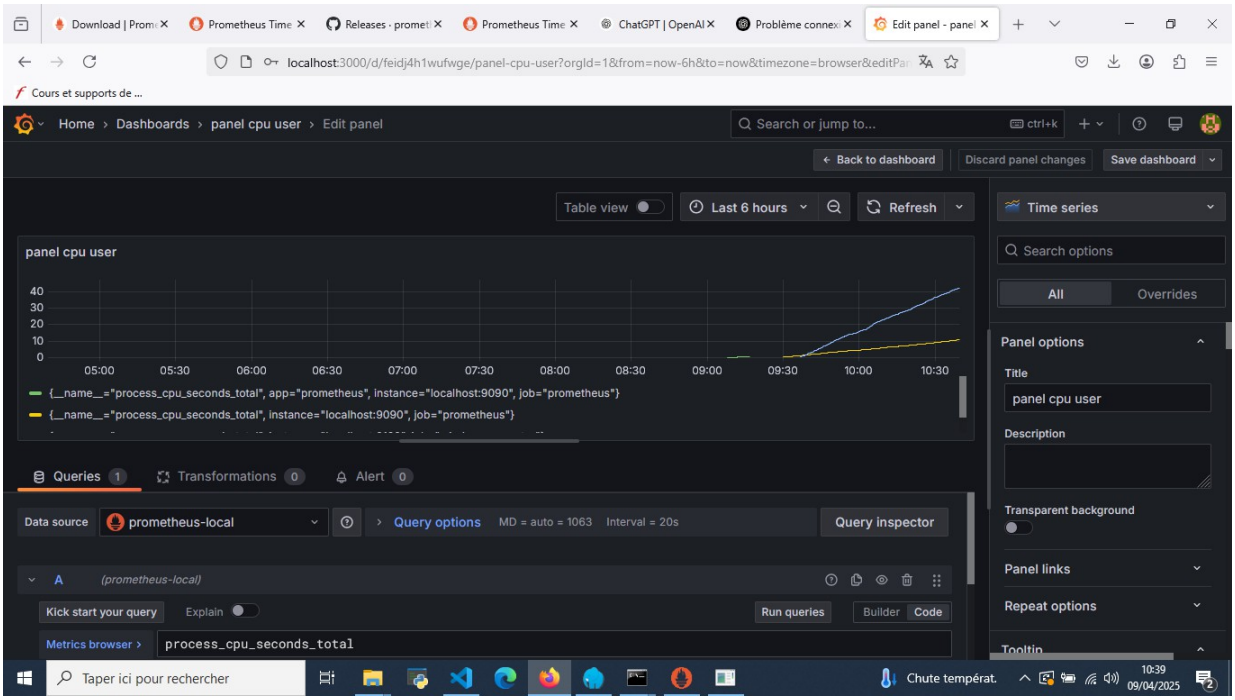


4. Création du dashboard Grafana

4.1 Panels intégrés

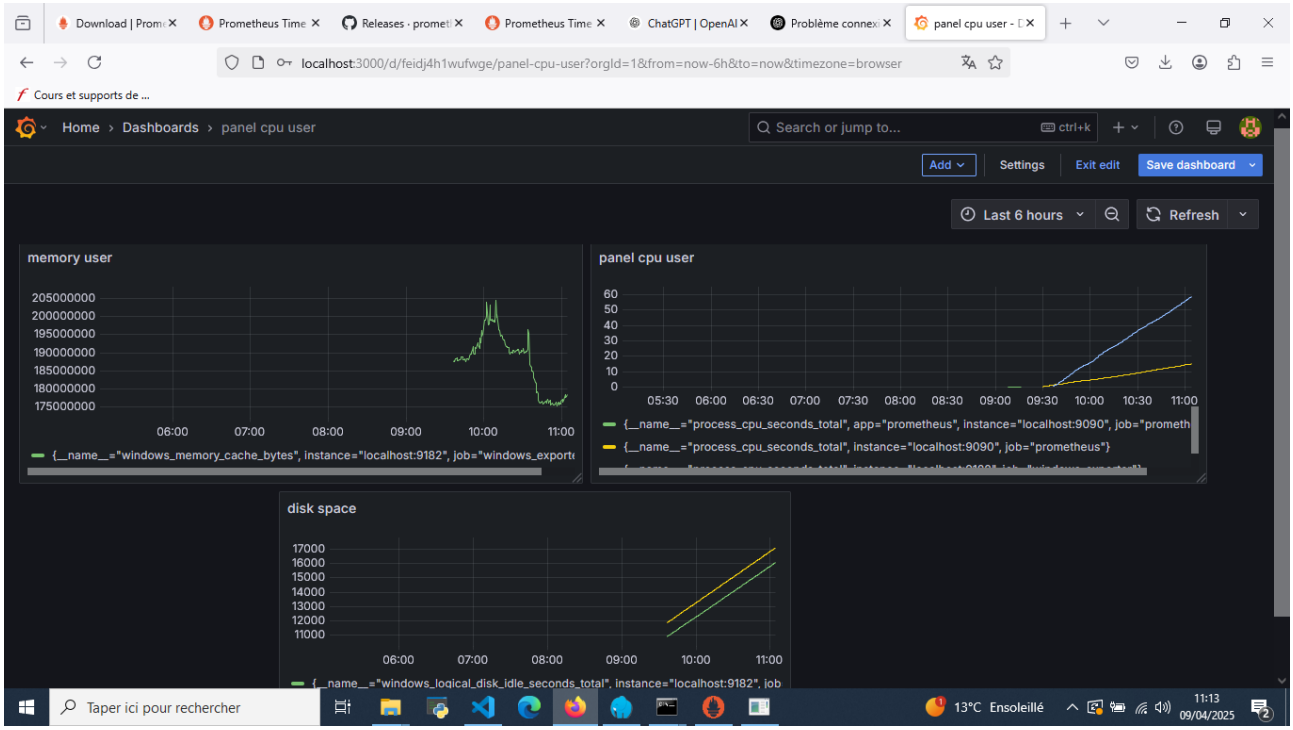
J'ai créé un dashboard personnalisé intitulé **"Monitoring Local PC"** avec les éléments suivants :

- **Utilisation CPU** (Graphique + Jauge)



- **Mémoire RAM utilisée**

- **Espace disque disponible**
- **Taux de charge CPU par cœur**



4.2 Design et lisibilité

Description du dashboard Grafana

Ce tableau de bord Grafana a été conçu pour visualiser en temps réel l'état des ressources système de mon PC local grâce aux métriques collectées par **Prometheus** via le **Windows Exporter**. Il regroupe trois panels principaux, chacun dédié à une ressource clé :

1. Panel "memory user"

- **Métrique utilisée** : `windows_memory_cache_bytes`
- **Source** : Windows Exporter via Prometheus (`localhost:9182`)
- **Affichage** : Courbe montrant l'évolution de la mémoire cache utilisée.
- **Utilité** : Permet de suivre l'utilisation de la mémoire cache et détecter les pics ou fuites potentielles de mémoire.

2. Panel "panel cpu user"

- **Métriques utilisées** : `process_cpu_seconds_total` (plusieurs instances)
- **Sources** : Prometheus (`localhost:9090`) et autres process locaux
- **Affichage** : Courbes représentant l'utilisation CPU par différents processus.
- **Utilité** : Utile pour surveiller la charge CPU générée par Prometheus ou d'autres services, et identifier ceux qui consomment le plus de ressources.

3. Panel "disk space"

- **Métrique utilisée** : `windows_logical_disk_idle_seconds_total`
- **Source** : Windows Exporter (`localhost:9182`)
- **Affichage** : Courbe représentant le temps d'inactivité des disques.
- **Utilité** : Fournit une vue indirecte de l'activité disque — moins il y a d'inactivité, plus le disque est sollicité.

-

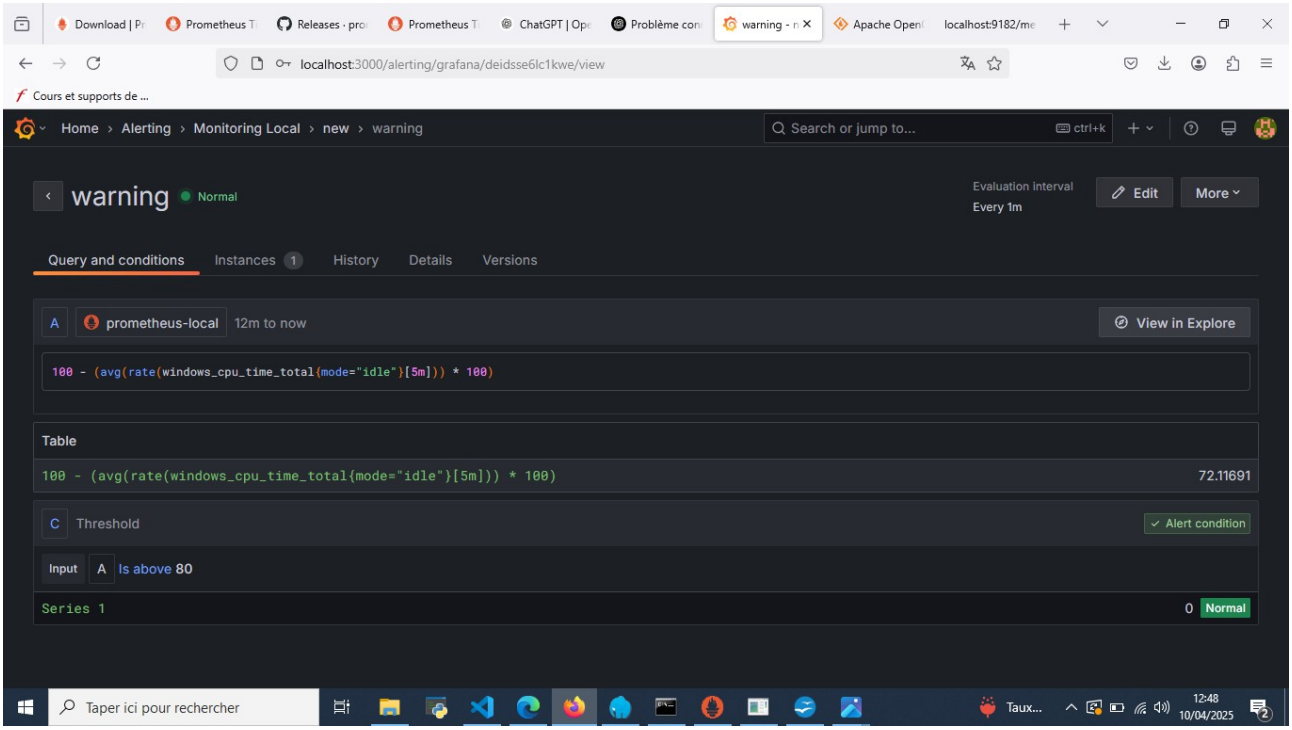
5. Alertes et notifications

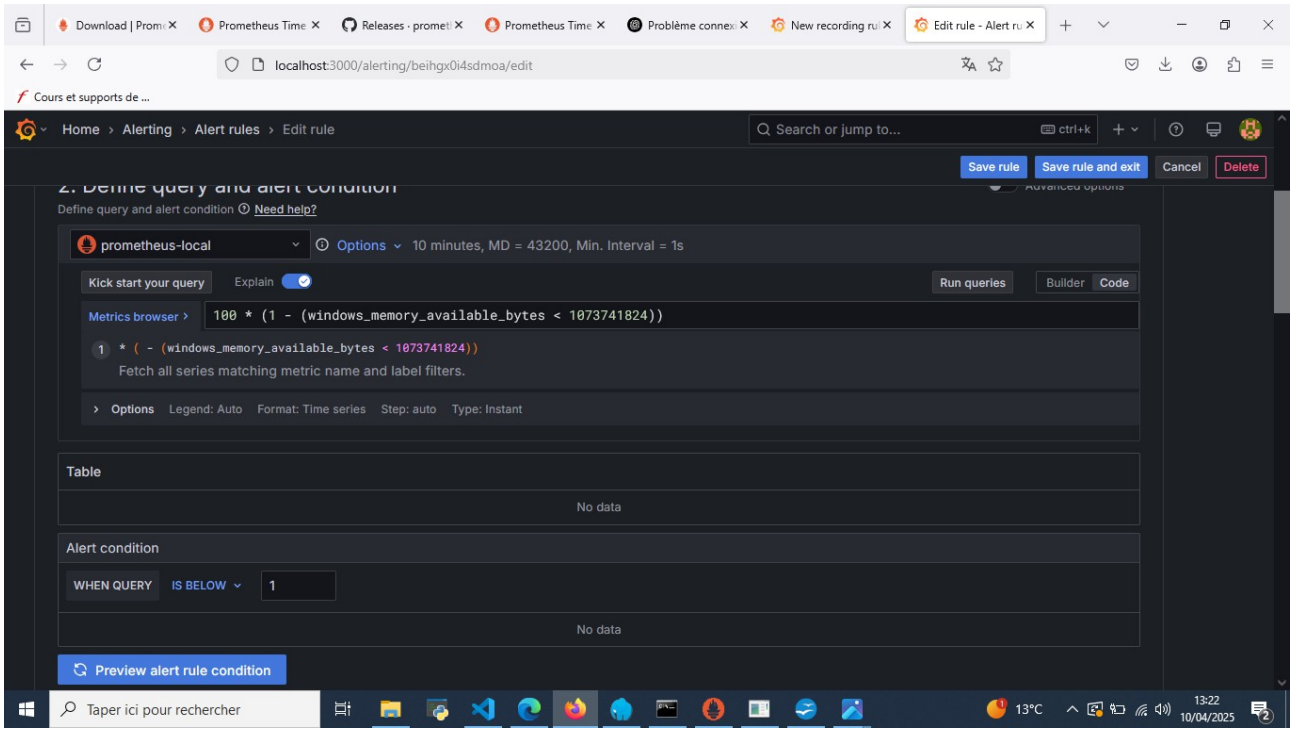
5.1 Seuils d'alerte définis

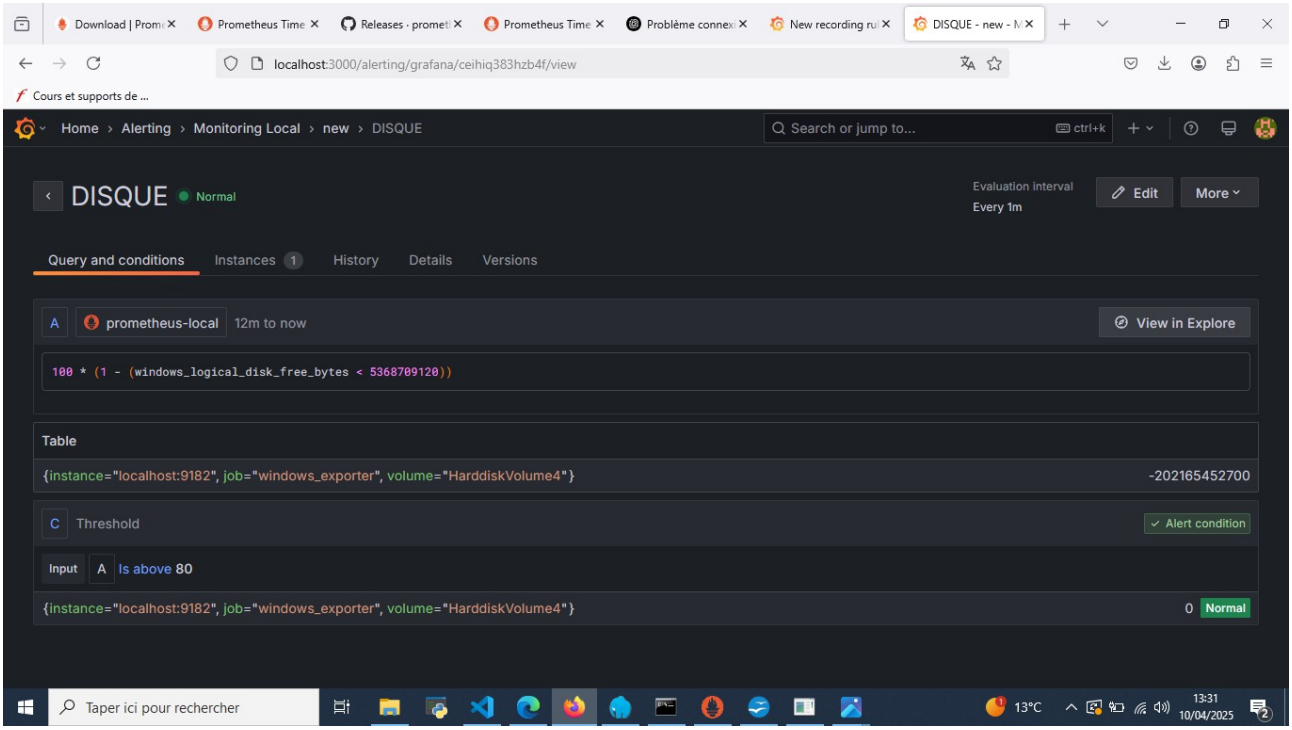
Ressource	Condition	Seuil	Justification
CPU	`avg(rate(process_cpu_seconds_total[1m])) > 0.8`	> 80%	Charge élevée à surveiller
RAM	`windows_os_physical_memory_free_bytes` < 1Go	< 1 Go	Risque de saturation mémoire
Disque	`windows_logical_disk_free_bytes` < 5Go	< 5 Go	Manque d'espace sur le disque

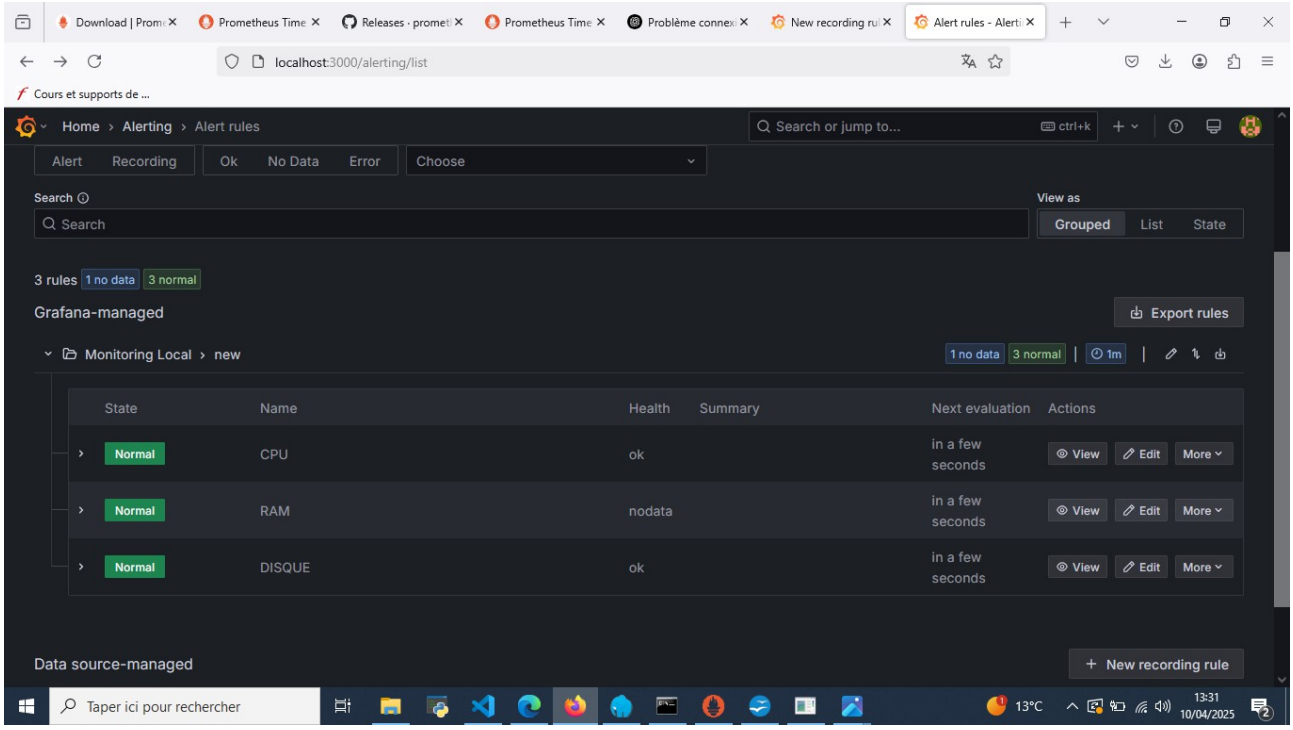
5.2 Configuration des alertes

Création dans Prometheus ou Grafana Alerts




Home › Alerting › Alert rules › Edit rule
Search or jump to...
Save rule
Save rule and exit
Cancel
Delete
Define query and alert condition
Need help?
prometheus-local
Options
10 minutes, MD = 43200, Min. Interval = 1s
Kick start your query
Explain
Run queries
Builder
Code
Metrics browser
100 * (1 - (windows_memory_available_bytes < 1073741824))
Fetch all series matching metric name and label filters.
Options
Legend: Auto
Format: Time series
Step: auto
Type: Instant
Table
No data
Alert condition
WHEN QUERY
IS BELOW
1
No data
Preview alert rule condition
Taper ici pour rechercher
13°C
13:22
10/04/2025

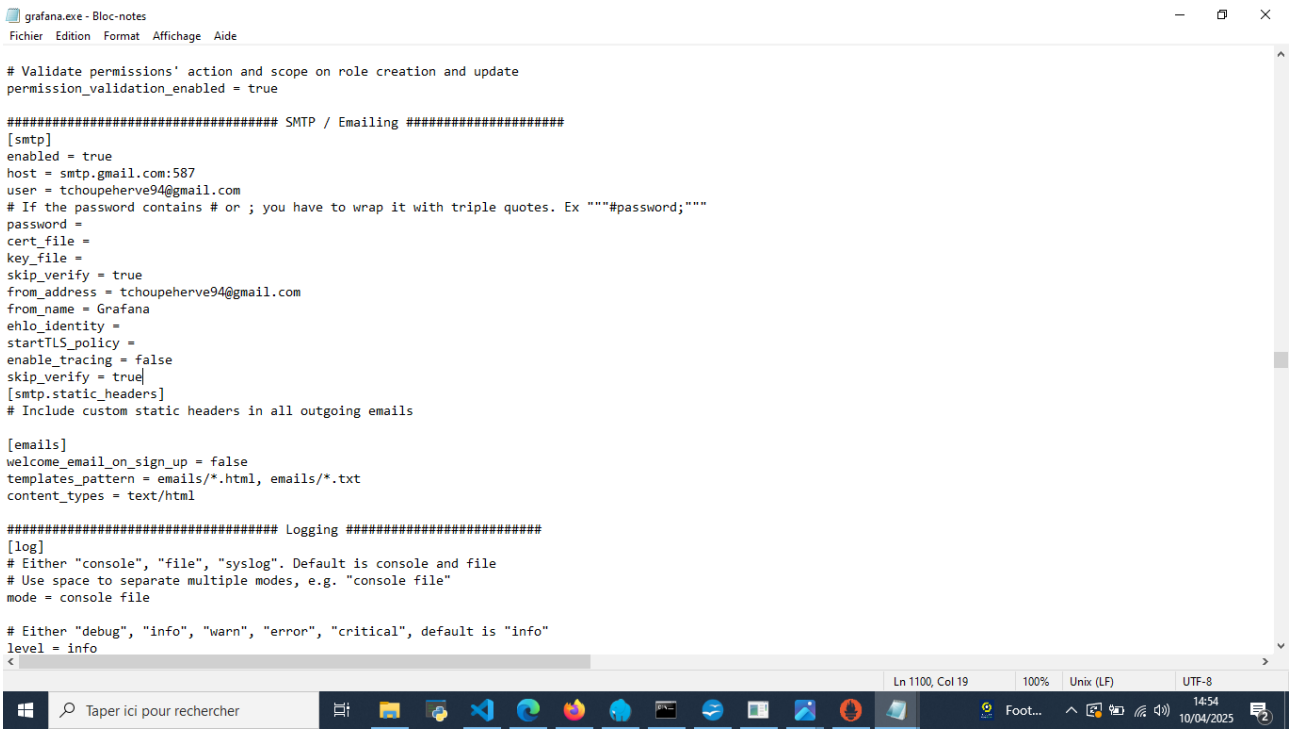

Home › Alerting › Monitoring Local › new › DISQUE
Search or jump to...
DISQUE
Normal
Evaluation interval
Every 1m
Edit
More
Query and conditions
Instances 1
History
Details
Versions
A
prometheus-local
12m to now
View in Explore
100 * (1 - (windows_logical_disk_free_bytes < 5368709120))
Table
{instance="localhost:9182", job="windows_exporter", volume="HarddiskVolume4"}
-202165452700
C
Threshold
Alert condition
Input
A
Is above 80
{instance="localhost:9182", job="windows_exporter", volume="HarddiskVolume4"}
0
Normal
Taper ici pour rechercher
13°C
13:31
10/04/2025

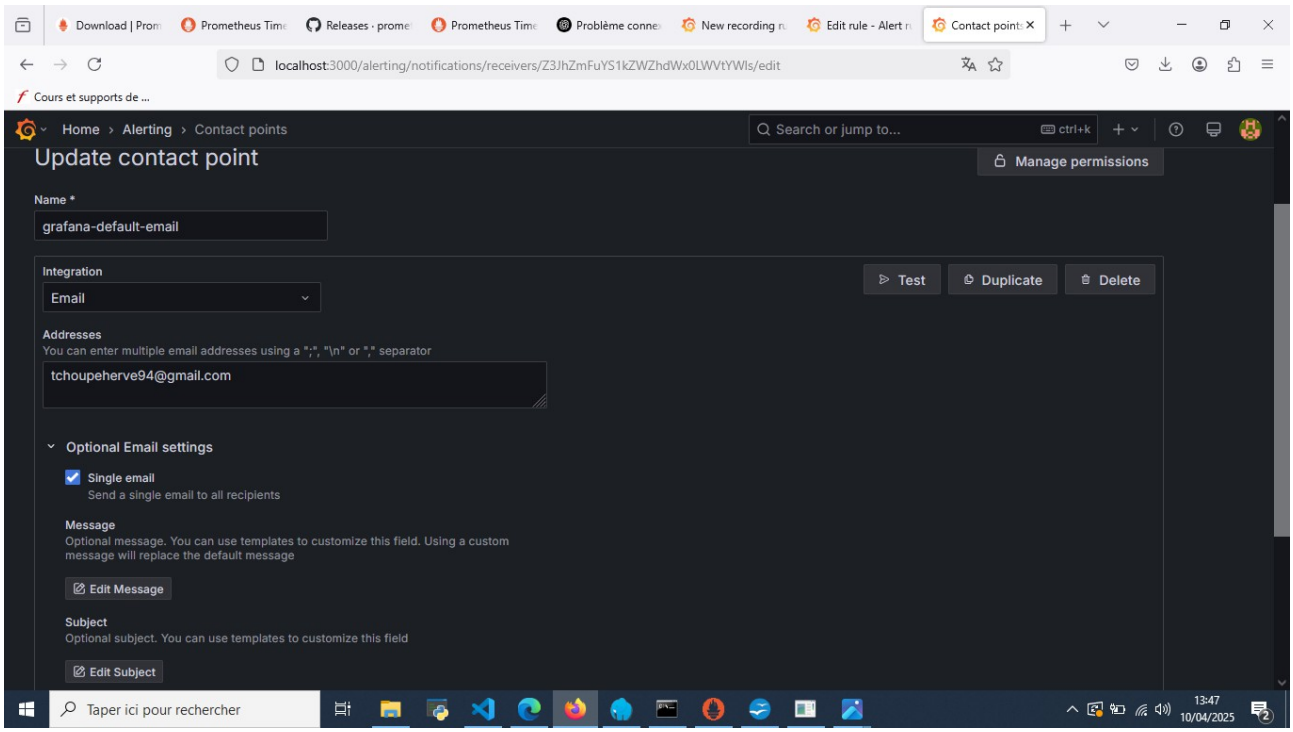


Notre travail sera accompagne d'un fichier yaml ou se trouve les configuration des alerte

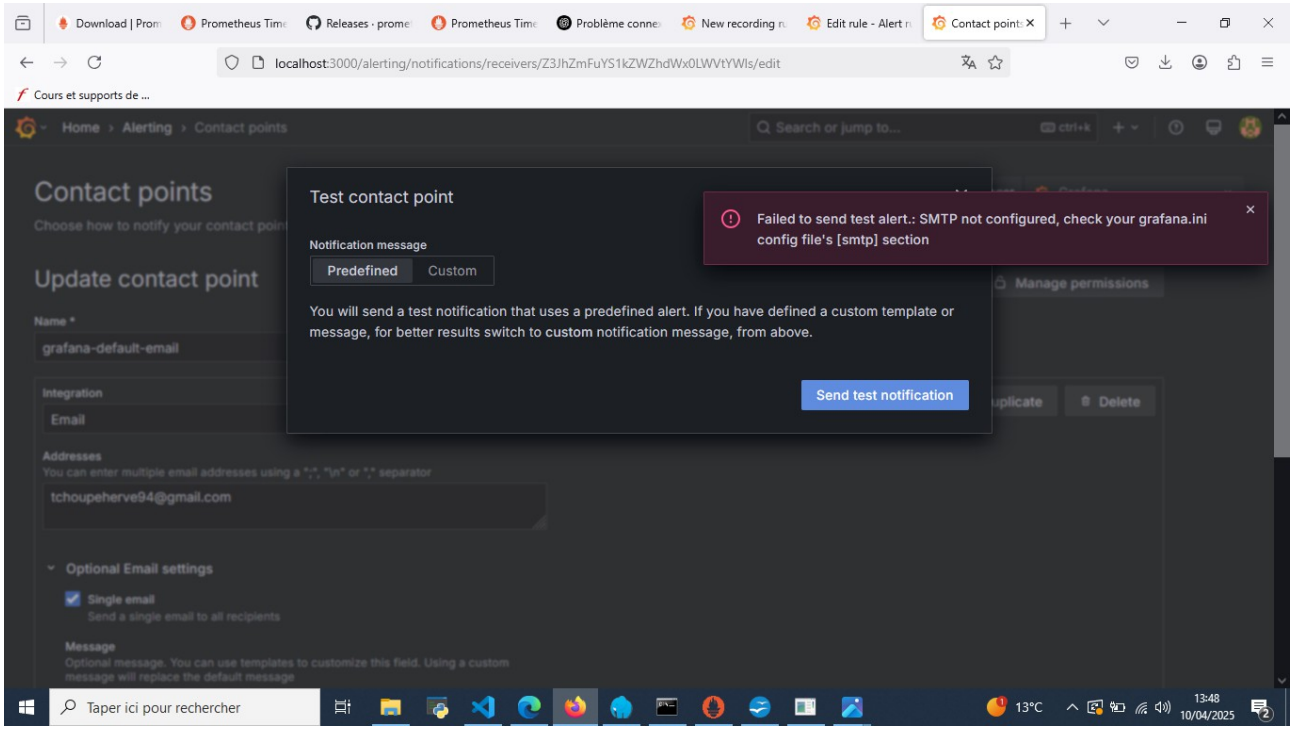
5.3 Notification par email (ou Slack)

- Configuration d’un canal de notification email (SMTP)





- Test de déclenchement d'alerte (en simulant une charge CPU élevée)



voilà la difficulté que je rencontre ce qui m'en peche de conclure et menée a bien mon projet

6. Conclusion

Grâce à ce projet, j’ai mis en place une **solution complète de monitoring local**. Cette infrastructure me permet de surveiller l’état de santé de mon PC en temps réel, d’anticiper les problèmes de ressources, et d’améliorer la visibilité sur l’utilisation du système