



SOMMAIRE

1. Cahier des charges	3
1.1. Contexte et Objectif.....	3
1.2. Objectifs Spécifiques	3
1.3. Description des Outils	3
1.4. Périmètres	4
1.5. Architecture Technique.....	4
1.6. Détail des Tâches et Actions	4
1.7. Exigences Fonctionnelles	5
1.8. Exigences Techniques	5
1.9. Plan de Mise en Œuvre	5
1.10. Critères de Réussite	6
2. Architecture réseau	6
3. Documentation technique	6
3.1. Prometheus	6
3.1.1. Installation de Prometheus	6
3.1.2. Lancer Prometheus en arrière-plan au démarrage	7
3.1.3. Configuration du fichier Prometheus.yml	7
3.1.4. Interface graphique	8
3.2. Grafana	8
3.2.1. Installation de Grafana	8
3.2.2. Interface graphique	9
3.2.3. Data sources.....	9
3.2.4. Création du Dashboard.....	10
3.3. NSSM	11
3.3.1. Installation de NSSM	11
3.4. Windows Exporter	12
3.4.1. Installaiton de Windows Exporter	12
3.4.2. Lancer Windows Exporter en arrière-plan au démarrage	12
3.4.3. Configuration du fichier Prometheus.yml	12
3.4.4. Redémarrer Prometheus.....	13



3.4.5. Vérification de la collecte des métriques dans Prometheus.....	13
3.4.6. Vérification de la collecte dans le Dashboard	14
3.5. Installation de Windows Exporter par GPO	14
3.5.1. Script pour installer Windows Exporter depuis le Web.....	15
3.5.2. GPO pour déployer le script	16
3.5.3. Script pour récupérer les adresses IP + fichier de sortie	19
3.4.8. Intégration des adresses IP dans le fichier Prometheus.yml	20



1. Cahier des charges

1.1. Contexte et Objectif

Dans le cadre de l'amélioration de la gestion des infrastructures IT de l'établissement scolaire, il est nécessaire de centraliser la collecte et l'analyse des logs des machines Windows. L'objectif est d'utiliser des outils de monitoring performants et adaptés pour assurer une surveillance continue et une analyse approfondie des performances des systèmes.

L'outil retenu pour la collecte des métriques est **Prometheus**, pour leur affichage et leur analyse **Grafana** sera utilisé. En complément, **Windows Exporter** permettra de collecter des données détaillées sur les serveurs Windows. Le déploiement des scripts d'installation de **Windows Exporter** sera automatisé via **GPO** en utilisant **NSSM** (Non-Sucking Service Manager).

1.2. Objectifs Spécifiques

- **Centraliser les logs** : Collecter les métriques de toutes les machines Windows et les centraliser via Prometheus.
- **Visualisation avec Grafana** : Créer des dashboards de visualisation des logs et des métriques système sous Grafana.
- **Automatisation de l'installation** : Déployer automatiquement Windows Exporter via GPO sur toutes les machines.
- **Monitoring des machines Windows** : Suivre en temps réel la santé des machines (CPU, RAM, espace disque, processus, etc.) à l'aide des métriques remontées par Windows Exporter.

1.3. Description des Outils

- **Prometheus** : Outil de collecte de données de monitoring open-source. Il collecte les métriques via des "scrapes" HTTP et stocke les données dans une base de données temporelle.
- **Grafana** : Outil de visualisation de données qui s'intègre parfaitement avec Prometheus pour afficher des tableaux de bord (dashboards) graphiques des données collectées.
- **NSSM** : Utilisé pour gérer Windows Exporter comme service sur les machines Windows sans qu'il soit nécessaire d'ajouter de surcharge dans le processus d'installation.
- **Windows Exporter** : Un exporter Prometheus qui permet de collecter des métriques sur les serveurs Windows (CPU, RAM, disque, processus, etc.).



- **GPO (Group Policy Objects)** : Permet de déployer automatiquement des scripts et des configurations sur un ensemble de machines Windows à partir de l'Active Directory.

1.4. Périmètres

- **Machines concernées** : Toutes les machines Windows de l'établissement scolaire.
- **Infrastructure** : Serveurs, postes de travail, et autres équipements Windows.
- **Méthodes de collecte** : Utilisation de Prometheus pour la collecte et Grafana pour la visualisation des logs.

1.5. Architecture Technique

- **Prometheus** : Serveur central sur lequel les métriques des machines Windows seront collectées.
- **Grafana** : Serveur configuré pour se connecter à Prometheus et afficher les données collectées sous forme de dashboards.
- **Windows Exporter** : Déployé sur chaque machine Windows comme service via NSSM pour assurer une collecte continue des métriques.
- **Active Directory** : Utilisation des GPO pour automatiser l'installation et la configuration de Windows Exporter sur les machines cibles.

1.6. Détail des Tâches et Actions

1.6.1 Préparation de l'Environnement

- **Configurer Prometheus :**
 - Installation de Prometheus sur le serveur AD.
 - Configuration du fichier `prometheus.yml` pour la collecte des métriques depuis les machines Windows (configuration des targets de scrapes).
- **Configurer Grafana :**
 - Installation de Grafana sur le serveur AD.
 - Configuration des sources de données Prometheus dans Grafana.
 - Création de dashboards pour visualiser les métriques collectées (ex : CPU, RAM, espace disque, processus).
- **Configurer Windows Exporter :**
 - Téléchargement de Windows Exporter sur un répertoire partagé ou en local sur chaque machine Windows.
 - Préparer le script d'installation avec NSSM pour installer Windows Exporter comme service.



- **Déployer via GPO :**

- Créer un script de déploiement pour installer Windows Exporter sur toutes les machines Windows via GPO.
- Déployer le script via Active Directory sur toutes les machines de l'établissement.

1.6.2 Suivi des Performances

- Configurer des alertes dans Prometheus et Grafana pour notifier les administrateurs en cas de problèmes sur les machines Windows (ex : utilisation élevée du CPU, manque d'espace disque).
- Mettre en place un système de vérification régulière pour assurer la collecte des logs et la bonne marche des services.

1.7. Exigences Fonctionnelles

- **Facilité de déploiement** : Le déploiement de Windows Exporter sur chaque machine doit être automatisé et transparent pour les utilisateurs.
- **Visualisation claire et accessible** : Les dashboards Grafana doivent être lisibles et offrir des graphiques simples pour permettre aux administrateurs de suivre les performances des machines.
- **Sécurité et Accès** : Le système doit respecter les bonnes pratiques de sécurité (authentification sur Grafana, cryptage des communications entre Prometheus, Grafana et les machines).
- **Disponibilité** : Prometheus et Grafana doivent être configurés pour assurer une haute disponibilité de la collecte des métriques.

1.8. Exigences Techniques

- **Systèmes Windows Supportés** : Toutes les versions de Windows utilisées dans l'établissement (Windows 7, Windows 10, Windows Server).
- **Réseau** : Les machines doivent pouvoir communiquer avec le serveur Prometheus pour envoyer les métriques.
- **Accès à Prometheus et Grafana** : Les administrateurs doivent pouvoir accéder aux dashboards et aux alertes via un navigateur web sécurisé.

1.9. Plan de Mise en Œuvre

- **Phase 1** : Installation et configuration de Prometheus et Grafana sur les serveurs dédiés.
- **Phase 2** : Préparation du script d'installation de Windows Exporter via NSSM et création de GPO pour déploiement.

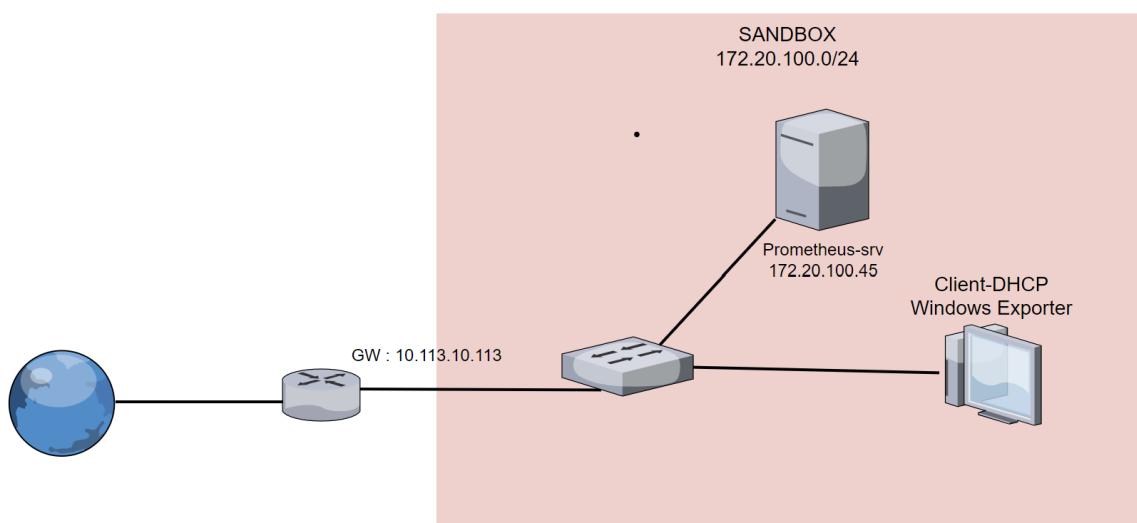


- **Phase 3** : Déploiement du script d'installation via GPO sur toutes les machines Windows.
- **Phase 4** : Configuration des dashboards dans Grafana et des alertes dans Prometheus.
- **Phase 5** : Tests de collecte des métriques et validation des alertes.
- **Phase 6** : Formation des administrateurs et utilisateurs concernés sur l'utilisation de Grafana pour visualiser les logs.

1.10. Critères de Réussite

- **Métriques collectées et affichées en temps réel** : Les logs et les métriques des machines Windows sont correctement collectés et affichés sur Grafana.
- **Automatisation complète** : Le déploiement des agents Windows Exporter via GPO est totalement transparent et automatisé.
- **Alertes et notifications fonctionnelles** : Les alertes définies sur Prometheus et Grafana sont envoyées aux administrateurs en cas de dépassement des seuils critiques.

2. Architecture réseau



3. Documentation technique

3.1. Prometheus

3.1.1. Installation de Prometheus

Aller sur le site officiel : <https://prometheus.io/download/>



Choisir comme ci-dessous :

Operating system windows ▾ Architecture amd64 ▾

Télécharger le .ZIP de la version récente

[prometheus](#)

The Prometheus monitoring system and time series database. [prometheus/prometheus](#)

3.1.0 / 2025-01-02 Release notes				
File name	OS	Arch	Size	SHA256 Checksum
prometheus-3.1.0.windows-amd64.zip	windows	amd64	111.40 MiB	fc9bd1397d8793c2c16fc3032065c41cd79082bdcfb3ba44ebe1adb3859eb63c
2.53.3 / 2024-11-04 LTS Release notes				
File name	OS	Arch	Size	SHA256 Checksum
prometheus-2.53.3.windows-amd64.zip	windows	amd64	101.63 MiB	91081d06538800454c01cd21269d2afaf6f2a07c4a559d00162cb9a7cce0f64b1

Décompresser avec 7-zip ou autre et placer dans le répertoire souhaité.

Nom	Modifié le	Type	Taille
data	17/01/2025 10:16	Dossier de fichiers	
LICENSE	19/12/2024 07:58	Fichier	12 Ko
NOTICE	19/12/2024 07:58	Fichier	4 Ko
prometheus.exe	19/12/2024 07:58	Application	145 205 Ko
prometheus.yml	15/01/2025 08:30	Fichier YML	2 Ko
promtool.exe	19/12/2024 07:58	Application	138 462 Ko

Lancer le fichier .EXE

3.1.2. Lancer Prometheus 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.

3.1.3. Configuration du fichier Prometheus.yml

Ouvrir ce fichier avec un éditeur comme NotePad++, et configurer comme ceci



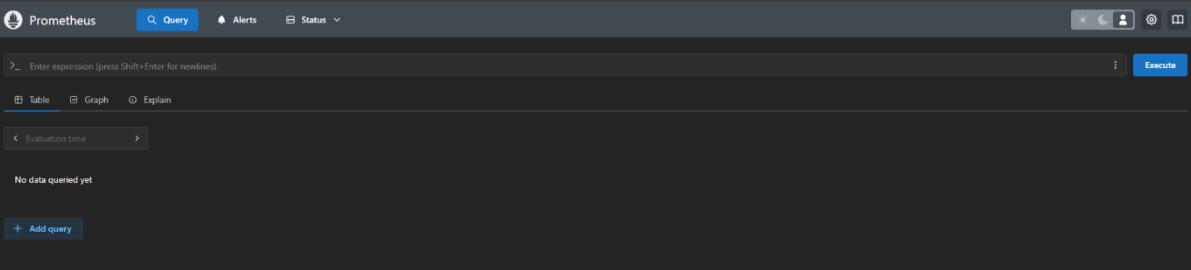
```

1  # my global config
2  global:
3      scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
4      evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
5      # scrape_timeout is set to the global default (10s).
6
7  # Alertmanager configuration
8  alerting:
9      alertmanagers:
10         - static_configs:
11             - targets:
12                 - alertmanager:9093
13
14 # Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
15 rule_files:
16     - "first_rules.yml"
17     - "second_rules.yml"
18
19 # A scrape configuration containing exactly one endpoint to scrape:
20 # Here it's Prometheus itself.
21 scrape_configs:
22     # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
23     - job_name: "Node"
24
25         # metrics_path defaults to '/metrics'
26         # scheme defaults to 'http'.
27
28         static_configs:
29             - targets: ["localhost:9090"]
30

```

3.1.4. Interface graphique

Pour accéder à l'interface graphique voici son URL : <http://localhost:9090/>



The screenshot shows the Prometheus web interface. The top navigation bar includes 'Prometheus', 'Query' (which is selected), 'Alerts', and 'Status'. Below the navigation is a search bar with the placeholder 'Enter expression (press Shift+Enter for newlines)'. Under the search bar are three tabs: 'Table' (selected), 'Graph', and 'Explain'. Below these tabs is a 'Evaluation time' selector with arrows for navigating between time ranges. A message 'No data queried yet' is displayed. At the bottom is a 'Add query' button.

3.2. Grafana

3.2.1. Installation de Grafana

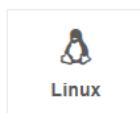
Aller sur le site officiel : <https://grafana.com/grafana/download/>

Choisir l'OS (ici Windows) :

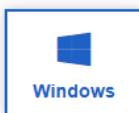


Release Info:

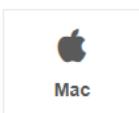
[What's New In Grafana 11.4.0](#)



Linux



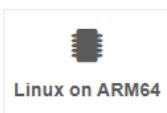
Windows



Mac



Docker



Linux on ARM64

Windows Installer (64 Bit) SHA256: 01bbbfb81addc1658ce6aad8d2550c761b55d18a5db9f41910e8bfc6c757ffb83

[Download the installer](#) (grafana-enterprise-11.4.0.windows-amd64.msi) and run it.

Standalone Windows Binaries (64 Bit) SHA256: 0a5d65dc7650ec42e87caf68c8f45b021a71d1c0e79f68cb3d2a66a14a045fa0

[Download the zip file](#) (grafana-enterprise-11.4.0.windows-amd64.zip) and follow the instructions in the installation guide below.

Read the Windows [installation guide](#).

Télécharger le dossier .ZIP

Décompresser dans le dossier souhaité et commencer l'installation en cliquant sur suivant.

3.2.2. Interface graphique

Voici le lien pour avoir accès à l'interface graphique : <http://localhost:3000/>

Se connecter avec id : admin | mdp : admin

The screenshot shows the Grafana welcome screen. On the left, a sidebar menu includes Home, Bookmarks, Starred, Dashboards, Playlists, Snapshots, Library panels, Public dashboards, Explore (Metrics, Alerting, Connections), Administration (General, Plugins and data, Users and access, Teams, Service accounts, Authentication), and Authentication. The main content area has a "Welcome to Grafana" header. It features a "Basic" section with a "TUTORIAL" card (DATA SOURCE AND DASHBOARDS, Grafana fundamentals) and a "COMPLETE" section with a "DATA SOURCE" card (Add your first data source) and a "DASHBOARD" card (Create your first dashboard). Below this, there's a "Dashboards" section with "Starred dashboards" and "Recently viewed dashboards" (Prometheus - Windows Exporter 2024). On the right, there's a "Latest from the blog" section with three posts: "Grafana Tempo 2.7 release: new TraceQL metrics functions, operational improvements, and more!" (janv. 16), "Grafana Play updates: recent growth, new privacy policies, and more" (janv. 15), and "Grafana SLO: Easily predict the likelihood that you'll hit your target" (janv. 14).

3.2.3. Data sources

Aller dans Data Sources et par défaut, il vous le montrera.

Sinon cliquer dans Add > Prometheus > taper l'URL <http://localhost:9090> > valider.



3.2.4. Création du Dashboard

Aller sur le site officiel : <https://grafana.com/grafana/dashboards/>

Rechercher Windows Exporter et choisir celui qui vous convient le mieux. Dans cette situation, j'ai choisi Windows exporter 2024 Dashboard.

Aller dans Dashboard > New > Import

Soit rechercher l'ID du Dashboard ou télécharger le JSON model.

Cliquer sur valider et aller dans el Dashboard qui vient d'être créé. C'est normal qu'il soit vide.

A screenshot of the Grafana 'Import dashboard' dialog. The title is 'Import dashboard'. It says 'Import dashboard from file or Grafana.com'. The 'Options' section has 'Name' set to 'Pormetheus Windows Exporter Dashboard' and 'Folder' set to 'Dashboards'. Under 'Unique identifier (UID)', it shows 'IV0hu1m7z' with a 'Change uid' button. In the 'Prometheus' section, there is a search bar with 'prometheus' and a dropdown arrow. At the bottom are 'Import' and 'Cancel' buttons.



Import dashboard

Import dashboard from file or Grafana.com



Upload dashboard JSON file

Drag and drop here or click to browse
Accepted file types: .json, .txt

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

Grafana.com dashboard URL or ID

Load

Import via dashboard JSON model

```
  "2d",
  "7d",
  "30d"
],
},
"timezone": "browser",
"title": "Windows Exporter Dashboard",
"uid": "IV0hu1m7z",
"version": 5
}
```

Load

Cancel

Nommer le Dashboard et choisir le Data source Prometheus.

Cliquer sur Valider, puis consulter le Dashboard. Il n'y aura pas de données. C'est normal.

3.3. NSSM

3.3.1. Installation de NSSM

Aller sur le site officiel : <https://nssm.cc/download>

Télécharger la version récente.

Latest release

nssm 2.24 (2014-08-31)
[be7b3577c6e3a280e5106a9e9db5b3775931cefc]



3.4. Windows Exporter

3.4.1. Installation de Windows Exporter

Aller sur le dépôt Github : https://github.com/prometheus-community/windows_exporter/releases

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

File	Size	Last Updated
sha256sums.txt	396 Bytes	Jul 23, 2022
windows_exporter-0.19.0-386.exe	18.3 MB	Jul 23, 2022
windows_exporter-0.19.0-386.msi	9.5 MB	Jul 23, 2022
windows_exporter-0.19.0-amd64.exe	18.3 MB	Jul 23, 2022
windows_exporter-0.19.0-amd64.msi	9.39 MB	Jul 23, 2022
Source code (zip)		Jul 23, 2022
Source code (tar.gz)		Jul 23, 2022

8 likes 3 comments 11 people reacted

3.4.2. 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

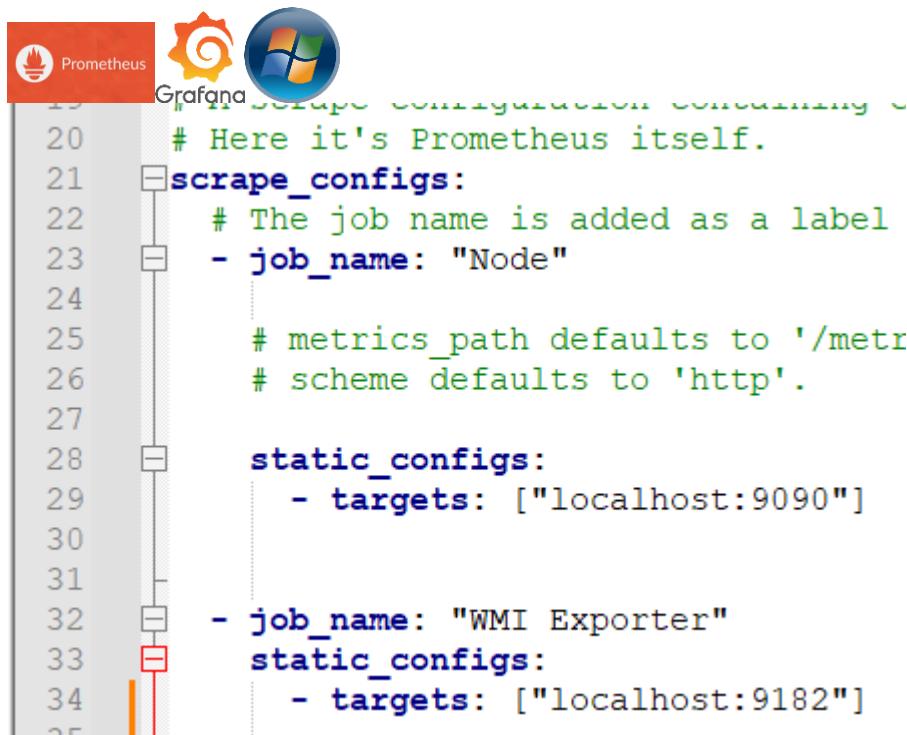
Win + R > shell:startup déplacer le raccourci dedans.

3.4.3. Configuration du fichier Prometheus.yml

Créer un nouveau job_name nommé : WMI Exporter

- `job_name: "WMI Exporter"`

Et taper l'adresse IP de votre machine. Puisqu'on est sur le serveur, on va inscrire ["localhost:9182"]



```

15
16
17
18
19
20     # Here it's Prometheus itself.
21     scrape_configs:
22         # The job name is added as a label
23         - job_name: "Node"
24
25             # metrics_path defaults to '/metrics'
26             # scheme defaults to 'http'.
27
28         static_configs:
29             - targets: ["localhost:9090"]
30
31
32         - job_name: "WMI Exporter"
33             static_configs:
34                 - targets: ["localhost:9182"]

```

3.4.4. Redémarrer Prometheus

Relancer Prometheus depuis le Gestionnaire des tâches > Services > Prometheus:

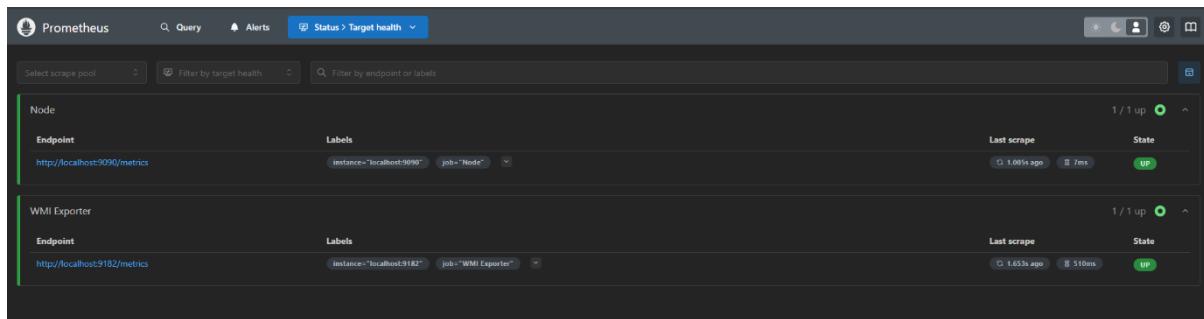
Clic droit et redémarrer le service.



3.4.5. Vérification de la collecte des métriques dans Prometheus

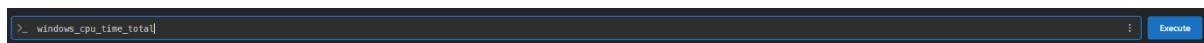
Lancer un navigateur et taper : <http://localhost:9090/targets>

Résultat à obtenir :



Aller dans <http://localhost:9090/query>

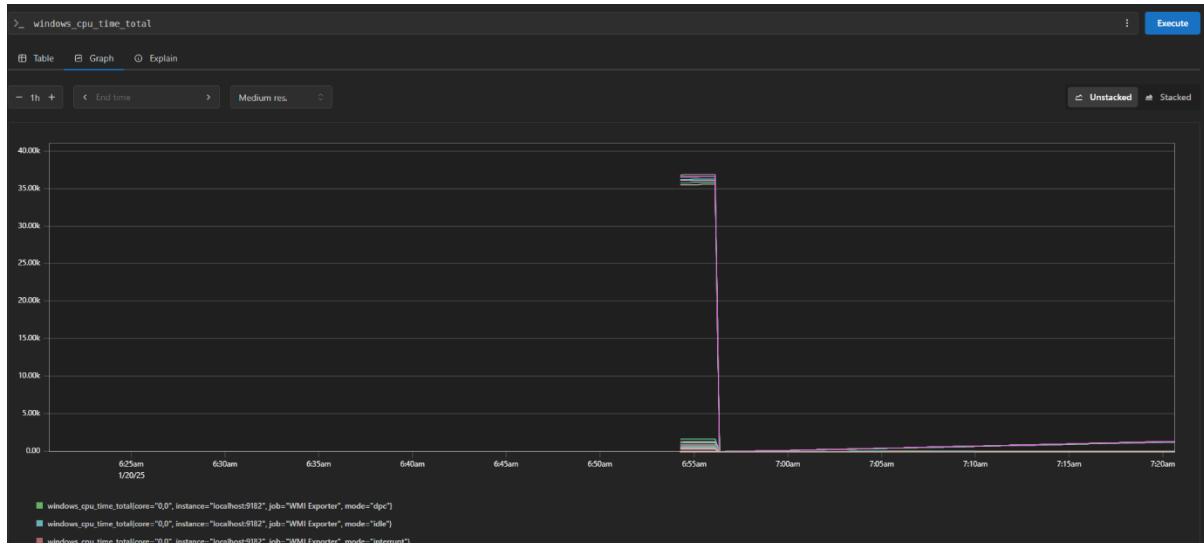
Dans la barre d'expression, taper :



Cliquer sur Execute et se mettre en mode Graph.



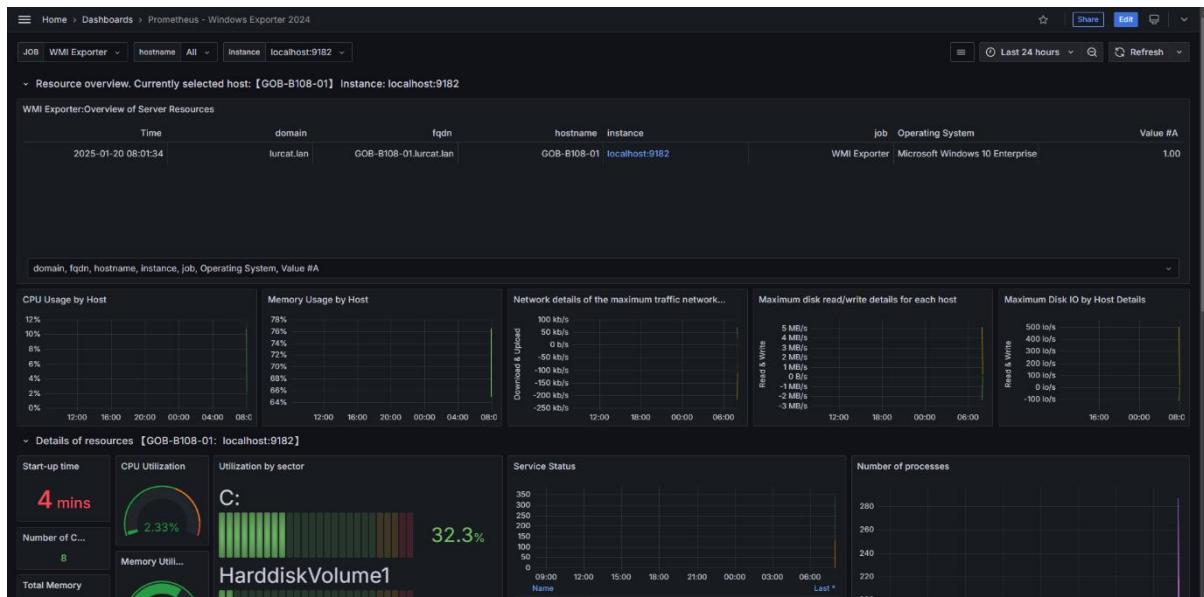
Résultat :



Prometheus reçoit bien la collecte des données.

3.4.6. Vérification de la collecte dans le Dashboard

Aller dans le Dashboard et voici le résultat que vous devez obtenir d'après la collecte de votre machine sur lequel Windows Exporter est installé.



3.5. Installation de Windows Exporter par GPO

<https://www.it-connect.fr/windows-executer-script-powershell-tache-planifiee-immediate-gpo/>



3.5.1. Script pour installer Windows Exporter depuis le Web

Install-WMI-depuis-web.ps1

```
# Définir l'URL de téléchargement et le chemin de destination
```

```
$downloadUrl = "https://github.com/prometheus-  
community/windows_exporter/releases/download/v0.19.0/windows_exporter-0.19.0-  
386.exe"
```

```
$destinationFolder = "C:\temp"
```

```
$destinationFile = "$destinationFolder\windows_exporter-0.19.0-386.exe"
```

```
# Vérifier si l'exécutable existe déjà
```

```
if (Test-Path -Path $destinationFile) {
```

```
    Write-Host "Windows Exporter est déjà téléchargé dans : $destinationFile"
```

```
} else {
```

```
# Créer le dossier de destination s'il n'existe pas
```

```
if (-not (Test-Path -Path $destinationFolder)) {
```

```
    Write-Host "Création du dossier $destinationFolder..."
```

```
    New-Item -ItemType Directory -Force -Path $destinationFolder
```

```
}
```

```
# Télécharger l'exécutable depuis GitHub
```

```
Write-Host "Téléchargement de Windows Exporter..."
```

```
try {
```

```
    Invoke-WebRequest -Uri $downloadUrl -OutFile $destinationFile
```

```
    Write-Host "Le fichier a été téléchargé avec succès à : $destinationFile"
```

```
} catch {
```

```
    Write-Host "Erreur lors du téléchargement : $_"
```

```
    exit
```

```
}
```

```
}
```



Lancer l'exécutable en arrière-plan (exécution continue)

```
Write-Host "Lancement de Windows Exporter en arrière-plan..."
```

```
Start-Process -FilePath $destinationFile -ArgumentList "--collectors.enabled=cpu,mem,net,system --no-collector.textfile" -WindowStyle Hidden
```

```
Write-Host "Windows Exporter est maintenant en cours d'exécution en arrière-plan."
```

Tester le script manuellement sur une machine hôte. Si cela fonctionne, enregistrer le script et on peut passer à l'étape suivante.

3.5.2. GPO pour déployer le script

Créer une nouvelle GPO dans l'OU concerner ou depuis la racine selon le besoin.

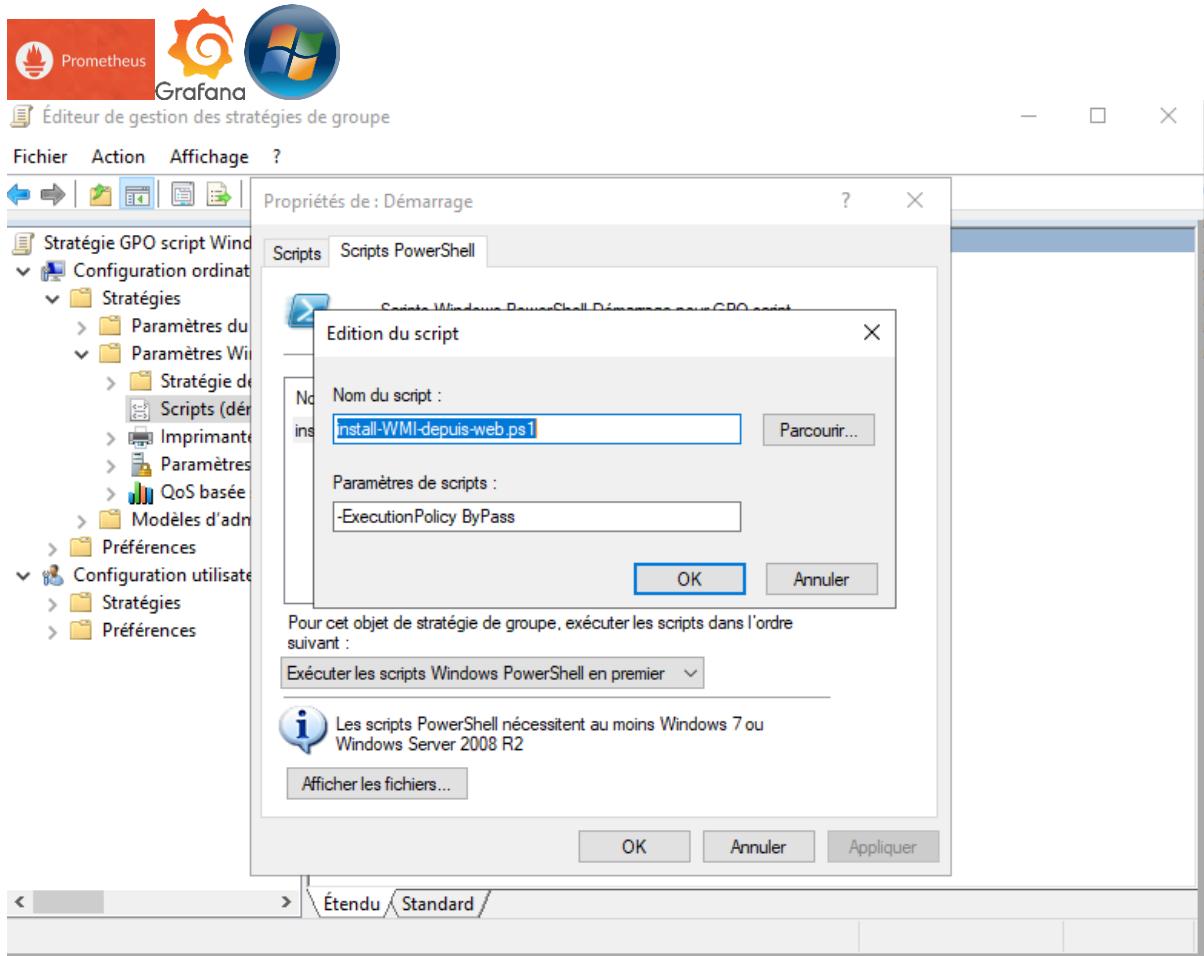
Créer un objet GPO, nommé : GPO script Windows Exporter deploy.

Clic droit > modifier.

Étape 1 : Intégrer le script

Configuration ordinateur > Stratégies > Paramètres Windows > Démarrage > Script PowerShell > Ajouter

Copier, coller le fichier install-WMI-depuis-web.ps1, dans le répertoire qui vient de s'ouvrir et dans paramètre de scripts : -ExecutionPolicy ByPass



Cliquer sur appliquer et fermer la fenêtre.

Étape 2 : Activer l'exécution des scripts

Configuration ordinateur > Stratégies > Modèles d'administrations > Composants Windows > Windows PowerShell

Cliquer sur Activer l'exécution des scripts > coché “activé” et stratégie d'exécution : Autoriser les scripts locaux et les scripts signés distants.



Activer l'exécution des scripts

Activer l'exécution des scripts

Paramètre précédent Paramètre suivant

Non configuré Commentaire :

Activé

Désactivé Pris en charge sur :

Au moins Microsoft Windows 7 ou famille Windows Server 2008

Options : Aide :

Stratégie d'exécution

Autoriser les scripts locaux et les scripts signés distants

Ce paramètre de stratégie vous permet de configurer la stratégie d'exécution de scripts en déterminant quels scripts peuvent être exécutés.

Si vous activez ce paramètre de stratégie, l'exécution des scripts sélectionnés dans la liste déroulante est autorisée.

Le paramètre de stratégie « Autoriser uniquement les scripts signés » permet uniquement l'exécution des scripts signés par un éditeur approuvé.

Le paramètre de stratégie « Autoriser les scripts locaux et les scripts signés distants » permet l'exécution de tous les scripts locaux. Les scripts provenant d'Internet doivent être signés par un éditeur approuvé.

Le paramètre de stratégie « Autoriser tous les scripts » permet l'exécution de tous les scripts.

Si vous désactivez ce paramètre de stratégie, aucun script ne peut être exécuté.

OK Annuler Appliquer

Étape 3 : Appliquer la GPO

Clic droit et coché sur “Appliqué” et “Lien activé”.

6	GPO script Windows Exporter Dep...	Qui	Qui	Activé	Aucun(e)	
5	test AgentGLPI				Aucun(e)	
3	Test Déploiement Pronote				Aucun(e)	

Modifiez le menu contextuel pour afficher les options “Appliqué” et “Lien activé”.

Se positionner sur la racine et actualiser la gestion de stratégie de groupe.

Étape 4 : Forcer la mise à jour des GPO sur la machine de test

Aller dans CMD et taper : gpupdate /force



Il faudra alors redémarrer la machine de test et vérifier avec la commande gpreresult /r si la GPO script Windows Exporter Deploy s'est bien intégré.

3.5.3. Script pour récupérer les adresses IP + fichier de sortie

```
# Importer le module Active Directory
```

```
Import-Module ActiveDirectory
```

```
# Chemin du fichier de sortie
```

```
$outputFilePath = "C:\Documents\recup-ip.txt"
```

```
# Récupérer tous les ordinateurs du domaine avec leurs adresses IP
```

```
$computers = Get-ADComputer -Filter * -Property IPv4Address
```

```
# Initialiser un tableau pour stocker les résultats formatés
```

```
$results = @()
```

```
# Filtrer et formater les adresses IP des machines
```

```
$computers | Where-Object { $_.IPv4Address -ne $null } | ForEach-Object {
```

```
    # Ajouter l'adresse IP formatée dans le tableau
```

```
    $results += "-targets [\"$($_.IPv4Address):9182\"]"
```

```
}
```

```
# Exporter les résultats dans un fichier texte
```

```
$results | Out-File -FilePath $outputFilePath -Encoding UTF8
```

```
Write-Host "Les résultats ont été exportés vers $outputFilePath"
```



Dans ce fichier-là "C:\Documents\recup-ip.txt", nous avons les adresses IP récupérer à partir du serveur AD.

3.4.8. Intégration des adresses IP dans le fichier Prometheus.yml

Copier toutes les adresses IP qui se trouvent dans le fichier et coller dans la partie [WMI Exporter] :

```
20  # Here it is Prometheus itself.
21  scrape_configs:
22      # The job name is added as a label `job=<job_name>` to any timeserie
23      - job_name: "Node"
24
25          # metrics_path defaults to '/metrics'
26          # scheme defaults to 'http'.
27
28      static_configs:
29          - targets: ["localhost:9090"]
30
31
32      - job_name: "WMI Exporter"
33          static_configs:
34              - targets: ["localhost:9182"]
35              - targets: ["172.20.14.240:9182"]
36
37
```



4. Cahier de recettes

4.1 Vérification de l'installation

ID	Vérification	Statut
R1	Prometheus est installé et accessible sur le port 9090	✓
R2	Grafana est installé et accessible sur le port 3000	✓
R3	Windows Exporter est installé et accessible sur le port 9182	✓

4.2 Vérification des configurations

ID	Vérification	Statut
R4	Prometheus collecte bien les métriques de Windows Exporter	✓
R5	Grafana affiche bien les métriques collectées par Prometheus	✓
R6	Les dashboards sont bien configurés et affichent les données attendues	✓

4.3 Tests finaux

ID	Vérification	Statut
R7	Les services redémarrent automatiquement après un reboot du serveur	✓
R8	Aucune erreur critique ou crash observé après 24h de fonctionnement	✓
R9	Documentation fournie pour la prise en main par les équipes IT	✓



5. Cahier de test

3.1 Installation des composants

ID	Description du test	Résultat attendu
T1	Télécharger et installer Prometheus	Installation réussie, service en écoute sur le port 9090
T2	Télécharger et installer Grafana	Installation réussie, accès via http://localhost:3000
T3	Télécharger et installer Windows Service Exporter	Service en écoute et exportation des métriques sur le port 9182

3.2 Vérification des services

ID	Description du test	Résultat attendu
T4	Vérifier que le service Prometheus est en cours d'exécution	Le service tourne sans erreur
T5	Vérifier que le service Grafana est en cours d'exécution	Accès au tableau de bord via le navigateur
T6	Vérifier que Windows Exporter expose bien les métriques	Accès aux métriques via http://localhost:9182/metrics

3.3 Configuration et supervision

ID	Description du test	Résultat attendu
T7	Configurer Prometheus pour collecter les métriques de Windows Exporter	Les métriques sont visibles via l'interface de Prometheus
T8	Ajouter Prometheus comme source de données dans Grafana	La connexion est réussie et testée
T9	Créer un dashboard basique affichant des métriques de Windows Exporter	Les données sont affichées correctement