

Installation de Prometheus et Grafana sur Windows



I. Introduction

Afin de garantir un fonctionnement optimal des systèmes informatiques, il est essentiel de faire le suivi et la surveillance des performances de ces derniers. Prometheus est un outil open source de monitoring et d'alerting permettant la collecte des métriques et de les analyser. Grafana est une plateforme de visualisation de données qui permet d'afficher les données collectées par Prometheus de manière intuitive.

II. Objectif

L'objectif de ce TP est de se familiariser avec l'installation et la configuration de Prometheus et Grafana sur une machine Windows. À la fin de ce TP nous devons être en mesure d'installer et d'utiliser Prometheus et Grafana pour surveiller les performances de notre système et pour visualiser les données collectées de manière intuitive et efficace.

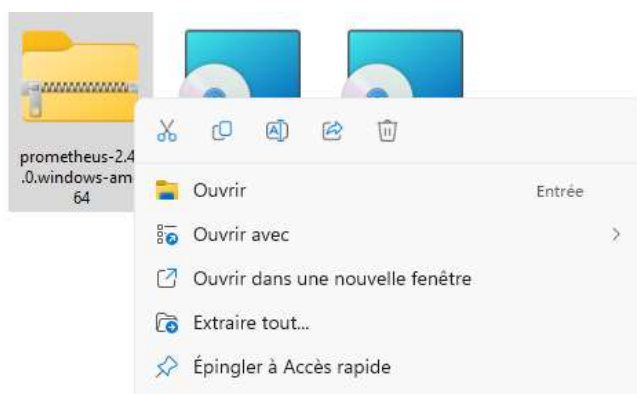
III. Matériel/logiciels

- Pc avec un environnement Windows
- Prometheus
- Grafana
- WMI Exporter

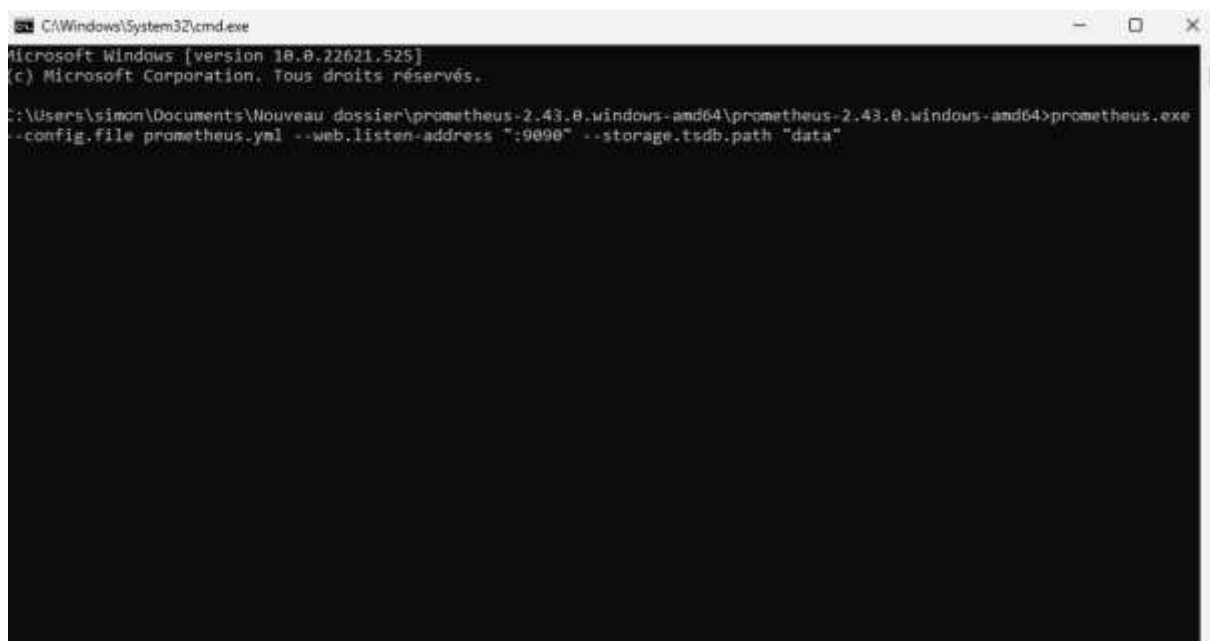
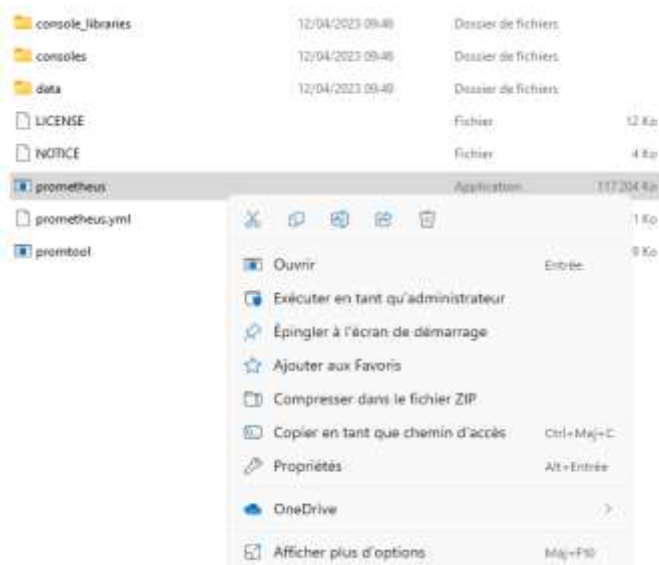
IV. Installation

1) Installation de Prometheus

Après avoir téléchargé la dernière version de Prometheus pour Windows (la version 2.40.0 pour ma part), il faut extraire le contenu du fichier



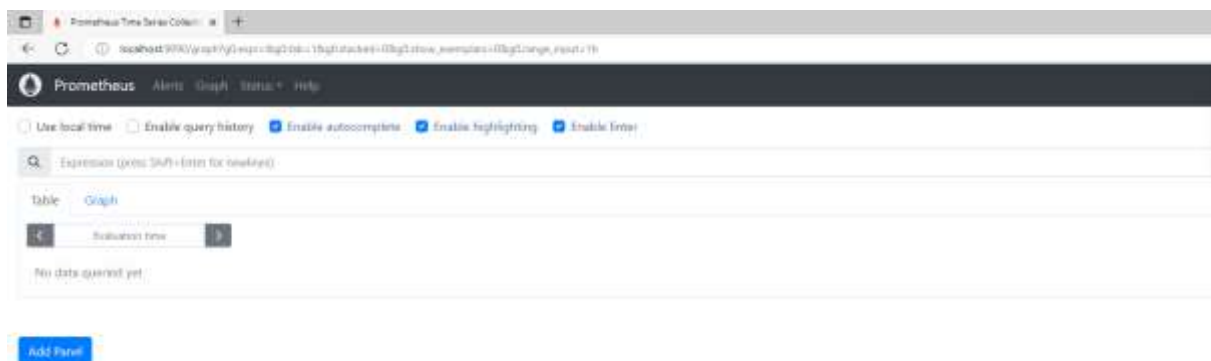
Ensuite, il faut exécuter le fichier prometheus.exe en tant qu'administrateur, ou alors, dans CMD, lancer la commande : **prometheus.exe --config.file prometheus.yml --web.listen-address ' :9090' --storage.tsdb.path 'data'**



Après le lancement de la commande, une fenêtre doit s'ouvrir comme ci-dessous.

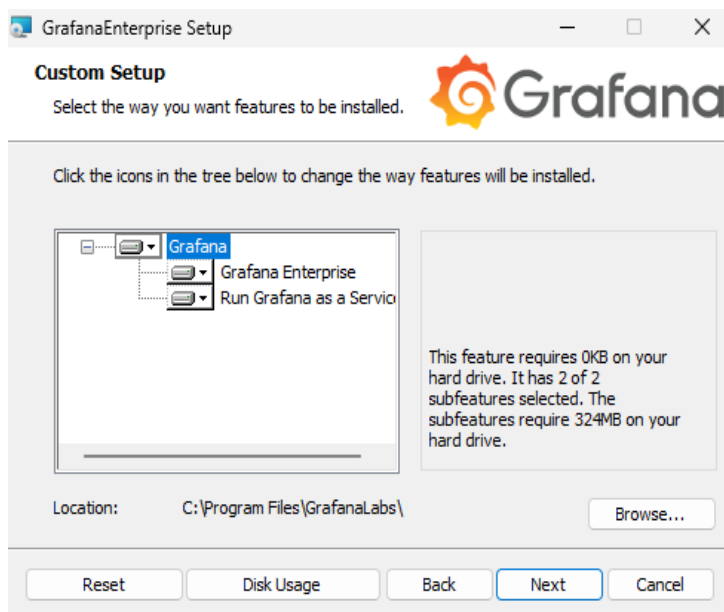
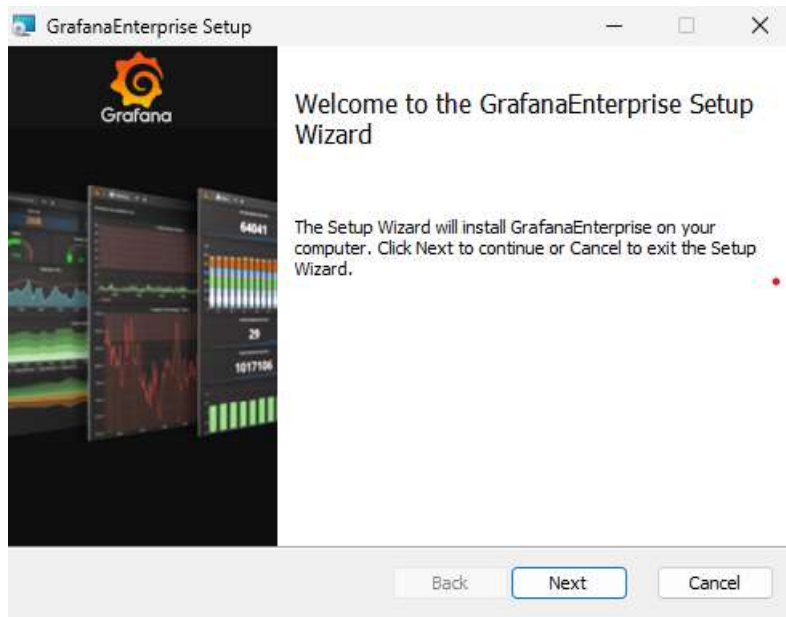
```
C:\Users\monr\Documents\Nouveau dossier\prometheus-2.43.0\windows-amd64\prometheus-2.43.0\windows-amd64\prometheus.exe
ts=2023-04-12T07:49:26.869Z caller=web.go:561 level=info component=web msg="Start listening for connections" address=0.0.0.0:9090
ts=2023-04-12T07:49:26.872Z caller=main.go:1985 level=info msg="Starting TSDB ..."
ts=2023-04-12T07:49:26.873Z caller=tls_config.go:232 level=info component=web msg="Listening on" address=[::]:9090
ts=2023-04-12T07:49:26.873Z caller=tls_config.go:235 level=info component=web msg="TLS is disabled." http2=false address=[::]:9090
ts=2023-04-12T07:49:26.899Z caller=head.go:587 level=info component=tsdb msg="Replaying on-disk memory mappable chunks if any"
ts=2023-04-12T07:49:26.100Z caller=head.go:658 level=info component=tsdb msg="On-disk memory mappable chunks replay completed" duration=0s
ts=2023-04-12T07:49:26.100Z caller=head.go:664 level=info component=tsdb msg="Replaying WAL, this may take a while"
ts=2023-04-12T07:49:26.103Z caller=head.go:735 level=info component=tsdb msg="WAL segment loaded" segment=0 maxSegment=0
ts=2023-04-12T07:49:26.103Z caller=head.go:772 level=info component=tsdb msg="WAL replay completed" checkpoint_replay_duration=0s wal_replay_duration=1.3791ms wal_replay_duration=0s total_replay_duration=1.2444ms
ts=2023-04-12T07:49:26.104Z caller=main.go:1828 level=info fs_type=unknown
ts=2023-04-12T07:49:26.105Z caller=main.go:1829 level=info msg="TSDB started"
ts=2023-04-12T07:49:26.105Z caller=main.go:1289 level=info msg="Loading configuration file" filename=prometheus.yml
ts=2023-04-12T07:49:26.121Z caller=main.go:1246 level=info msg="Completed loading of configuration file" filename=prometheus.yml totalDuration=16.2868ms
ts=2023-04-12T07:49:26.126Z caller=main.go:990 level=info msg="Server is ready to receive web requests."
ts=2023-04-12T07:49:26.126Z caller=manager.go:974 level=info component="rule manager" msg="Starting rule manager..."
```

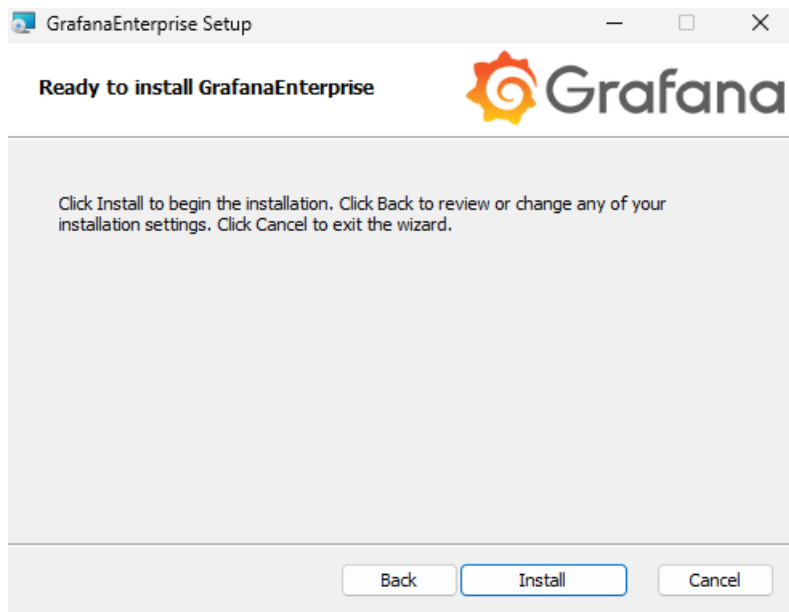
En tapant **localhost:9090** dans la barre de recherche d'un navigateur web, on arrive sur le tableau de bord de prometheus, que l'on vient d'installer.



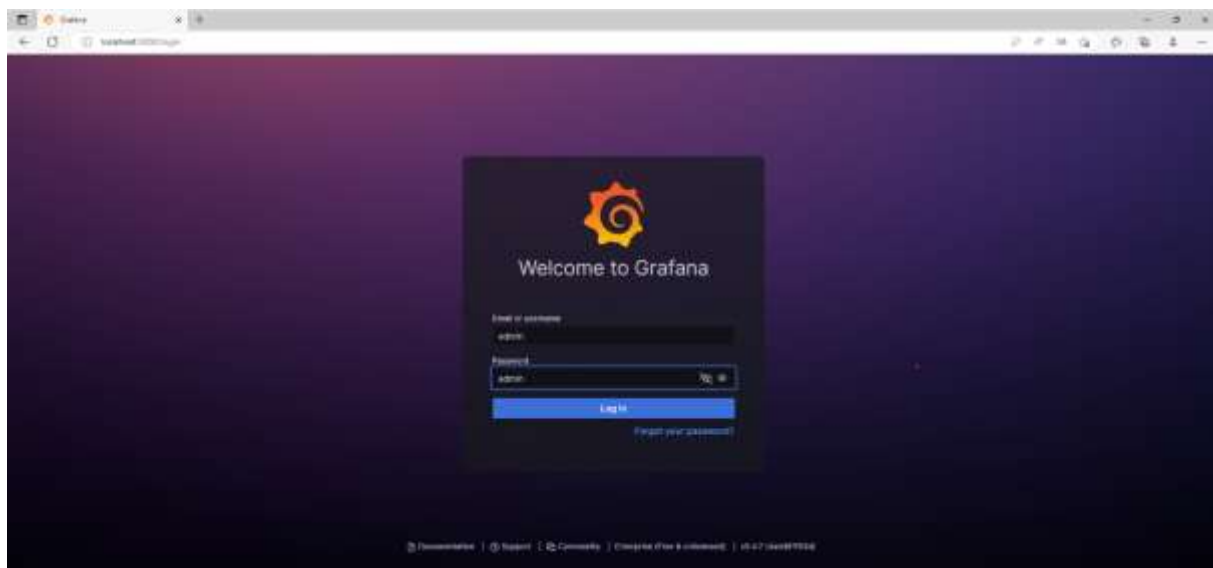
2) Installation de Grafana

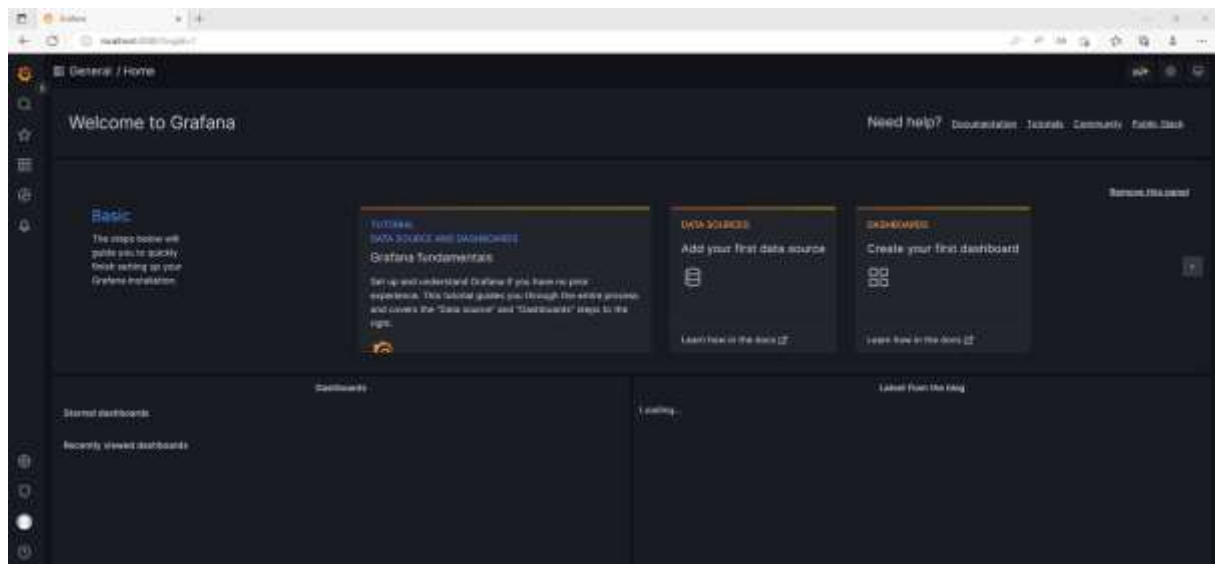
Après avoir téléchargé la dernière version de Grafana pour windows, il faut lancer l'installation.





Après l'installation de Grafana nous pouvons y accéder via un navigateur de recherche en tapant localhost:3000 dans la barre de recherche. On tombe sur une page de connexion, les identifiants par défaut sont : admin admin.





Dans C:\Program Files\GrafanaLabs\grafana\conf\defaults, il faut chercher la partie smtp et remplacer la valeur false par true dans enabled

```

*defaults - Bloc-notes

Fichier  Modifier  Affichage

# Reset basic roles permissions on boot
# Warning left to true, basic roles permissions will be reset
reset_basic_roles = false

##### SMTP / Emailing #####
[smtp]
enabled = true
host = localhost:25
user =
# If the password contains # or ; you have to wrap it with triple quotes. Ex ""#password;""
password =
cert_file =
key_file =
skip_verify = false
from_address = admin@grafana.localhost
from_name = Grafana
ehlo_identity =
startTLS_policy =

[emails]
welcome_email_on_sign_up = false
templates_pattern = emails/*.html, emails/*.txt
content_types = text/html

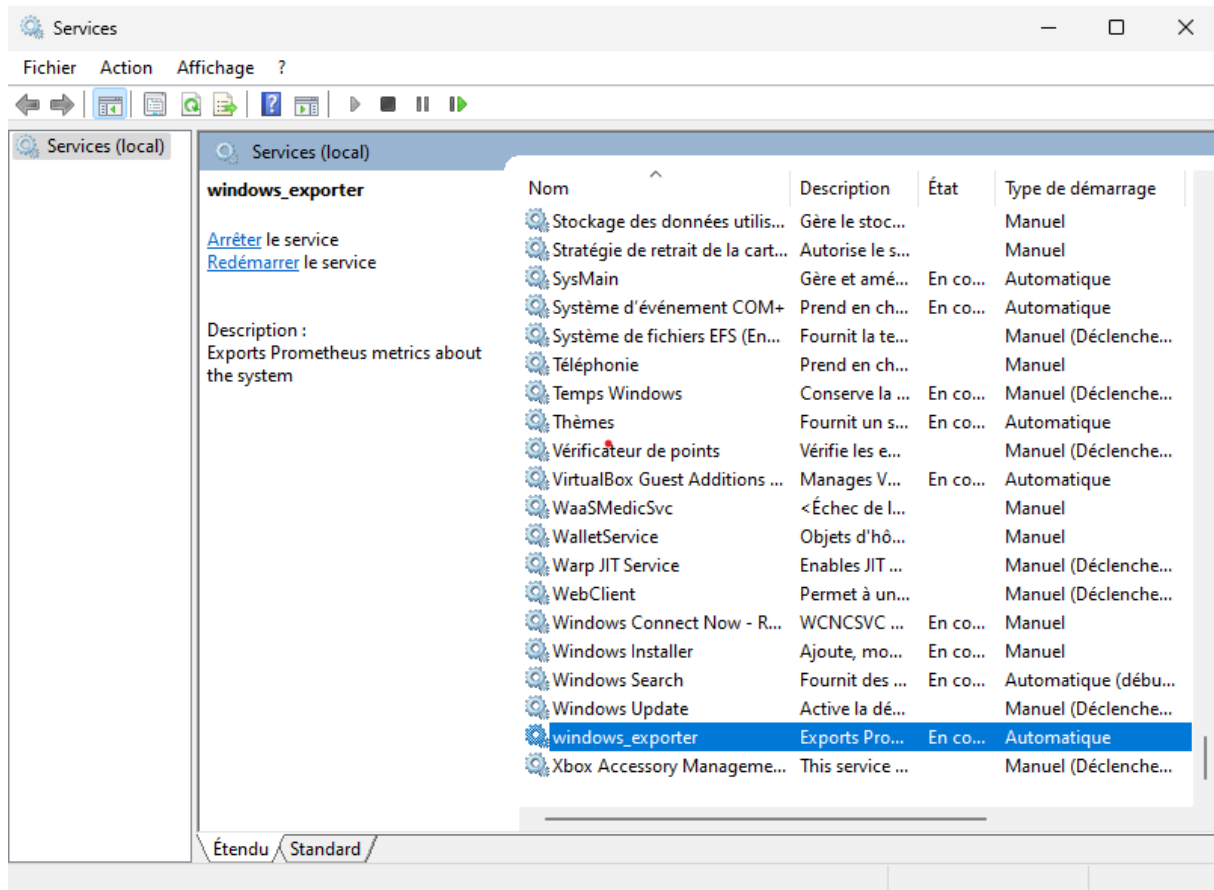
##### Logging #####
[log]
# Either "console", "file", "syslog". Default is console and file
# Use console to generate multiple modes: a "console file"

```

3) Installation de WMI Exporter

Après avoir téléchargé windows_exporter-0.22.0-amd64 sur <https://github.com/prometheus-community/windows-exporter/releases>, il faut lancer directement l'installation.

Après l'installation de WMI Exporter, il faut vérifier si l'installation s'est bien déroulée, il faut donc aller dans Services et rechercher windows exporter, et vérifier s'il est en cours d'exécution.



```

# HELP go_gc_duration_seconds A summary of the pause duration of garbage collection cycles.
# TYPE go_gc_duration_seconds summary
go_gc_duration_seconds{quantile="0"} 0
go_gc_duration_seconds{quantile="0.25"} 0
go_gc_duration_seconds{quantile="0.5"} 0
go_gc_duration_seconds{quantile="0.75"} 0
go_gc_duration_seconds{quantile="1"} 0.0059848
go_gc_duration_seconds_sum 0.0067714
go_gc_duration_seconds_count 10
# HELP go_goroutines Number of goroutines that currently exist.
# TYPE go_goroutines gauge
go_goroutines 11
# HELP go_info Information about the Go environment.
# TYPE go_info gauge
go_info{version="go1.20.1"} 1
# HELP go_memstats_alloc_bytes Number of bytes allocated and still in use.
# TYPE go_memstats_alloc_bytes gauge
go_memstats_alloc_bytes 3.460272e+06
# HELP go_memstats_alloc_bytes_total Total number of bytes allocated, even if freed.
# TYPE go_memstats_alloc_bytes_total counter
go_memstats_alloc_bytes_total 5.438471e+07
# HELP go_memstats_buck_hash_sys_bytes Number of bytes used by the profiling bucket hash table.
# TYPE go_memstats_buck_hash_sys_bytes gauge
go_memstats_buck_hash_sys_bytes 1.042343e+06
# HELP go_memstats_frees_total Total number of frees.
# TYPE go_memstats_frees_total counter
go_memstats_frees_total 84410
# HELP go_memstats_gc_sys_bytes Number of bytes used for garbage collection system metadata.
# TYPE go_memstats_gc_sys_bytes gauge
go_memstats_gc_sys_bytes 7.427664e+06
# HELP go_memstats_heap_alloc_bytes Number of heap bytes allocated and still in use.
# TYPE go_memstats_heap_alloc_bytes gauge
go_memstats_heap_alloc_bytes 3.460272e+06
# HELP go_memstats_heap_idle_bytes Number of heap bytes waiting to be used.
# TYPE go_memstats_heap_idle_bytes gauge
go_memstats_heap_idle_bytes 5.950431e+06
# HELP go_memstats_heap_inuse_bytes Number of heap bytes that are in use.
# TYPE go_memstats_heap_inuse_bytes gauge
go_memstats_heap_inuse_bytes 5.160335e+06
# HELP go_memstats_heap_objects Number of allocated objects.
# TYPE go_memstats_heap_objects gauge
go_memstats_heap_objects 18058
# HELP go_memstats_heap_released_bytes Number of heap bytes released to OS.
# TYPE go_memstats_heap_released_bytes gauge
go_memstats_heap_released_bytes 9.890048e+06
# HELP go_memstats_heap_sys_bytes Number of heap bytes obtained from system.
# TYPE go_memstats_heap_sys_bytes gauge
go_memstats_heap_sys_bytes 1.132076e+07
# HELP go_memstats_last_gc_time_seconds Number of seconds since 1970 of last garbage collection.
# TYPE go_memstats_last_gc_time_seconds gauge
go_memstats_last_gc_time_seconds 1.6812921771282330e+09
# HELP go_memstats_lookups_total Total number of pointer lookups.
# TYPE go_memstats_lookups_total counter
go_memstats_lookups_total 0
```

Il faut également configurer le fichier Prometheus.yml dans prometheus-2.43.0.windows-amd64/prometheus-2.43.0.windows-amd64 de cette façon :

- job_name : 'WMI Exporter'

metrics_path defaults to '/metrics'

#scheme defaults to 'http'

Static_configs :

-targets : ['Adresse_ip :9182']

Enregistrer et quitter.



```
# my global config
global:
  scrape_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.
  evaluation_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.
  # scrape_timeout is set to the global default (10s).

# Alertmanager configuration
alerting:
  alertmanagers:
    - static_configs:
        - 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: 'WMI Exporter'

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

    static_configs:
      - targets: ['10.100.0.11:9090']
```

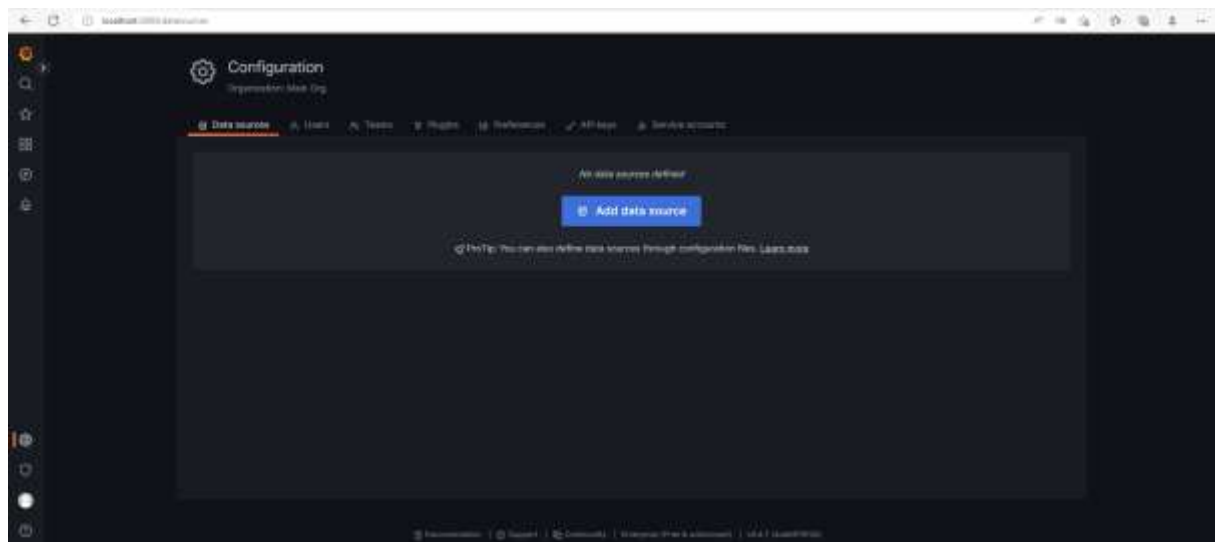
Après avoir enregistré et quitté le fichier prometheus.yml, il faut relancer la commande :
prometheus.exe --config.file prometheus.yml --web.listen-address ' :9090' --storage.tsdb.path 'dada' dans le cmd.

En rafraichissant le tableau de bord prometheus, WMI Exporter doit apparaître.

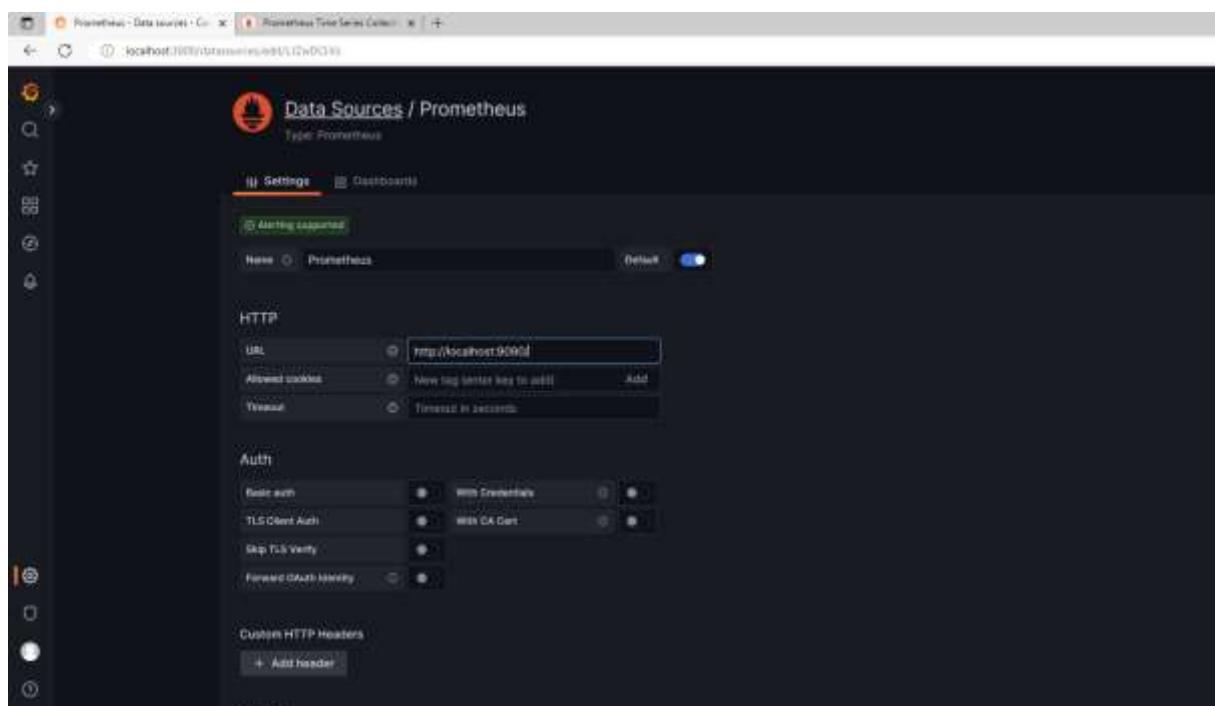


3) Configuration de Grafana comme DataSource de Prometheus

Dans Grafana – Configuration, il faut ajouter une data source.

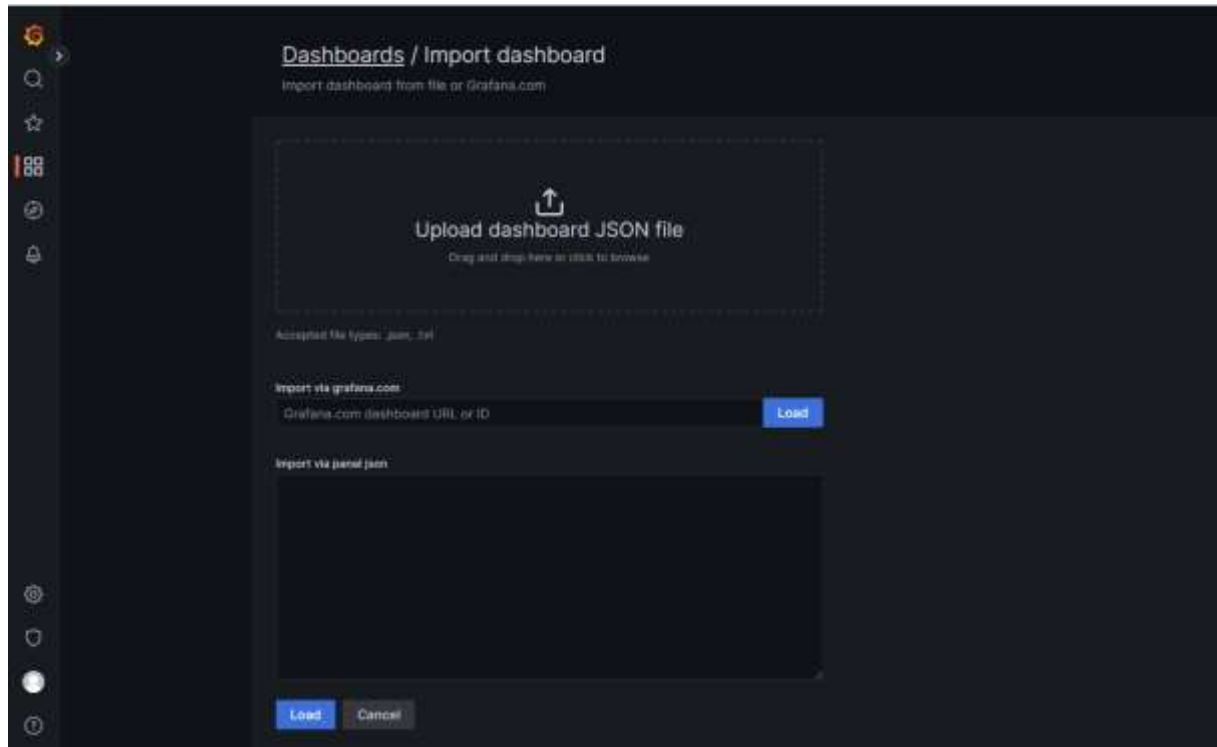


Dans Data Sources / Prometheus, il faut préciser l'URL de Prometheus, qui est <http://localhost:9090/>. Il faut ensuite cliquer sur save & test.



4) Création d'un tableau de bord grafana

On peut télécharger un tableau de bord via un fichier json, ou par ID.



V. Conclusion

En conclusion à ce TP, l'installation de prometheus et grafana sur un environnement windows s'est déroulé avec succès. Malgré que l'installation de ces outils de supervisions puisse paraître plus complexe, surtout pour des personnes débutantes, il est tout à fait possible d'installer ces deux outils de surveillance de manière efficace si on suit les instructions étape par étape. En utilisant Prometheus et Grafana, nous avons la capacité de surveiller en temps réel les performances d'un système et d'obtenir une vue d'ensemble complète des performances de nos applications et services.