Installation de Grafana et configuration

Objectif :

L'objectif de ce projet est de déployer une plateforme de visualisation de données centralisée pour analyser et surveiller l'infrastructure système en temps réel.

Mise en place d'un chiffrement HTTPS et d'une authentification centralisée (ldap)

Grafana est une plateforme open source de visualisation de données, conçue pour analyser et surveiller des systèmes en temps réel. Elle permet de créer des tableaux de bord interactifs et personnalisables, offrant une vue unifiée des données provenant de diverses sources comme Prometheus, ElasticSearch, MySQL, PostgreSQL, et bien d'autres.

Prérequis pour l'installation : 4 cœurs / 16 go de ram / 100 go de stockage (nécessaire pour la rétention des métriques sur plusieurs jours/semaine)

Sur un serveur Debian 13, avec un accès a root ou sudo

Install Grafana on Debian or Ubuntu | Grafana documentation

Compte de base grafana : Identifiant : admin

Mot de passe : grafanakillian

La solution repose sur une chaine de collecte et de visualisation :

1. Telegraf : agent de collecte (récupère les données de vCenter).
2. Prometheus : Bases de données de séries temporelles (stocke les métriques).
3. Grafana : interface de visualisation (interroge Prometheus)

[vCenter :443]

|

v

[Telegraf :9272]

|

v

[Prometheus :9090]

|

v

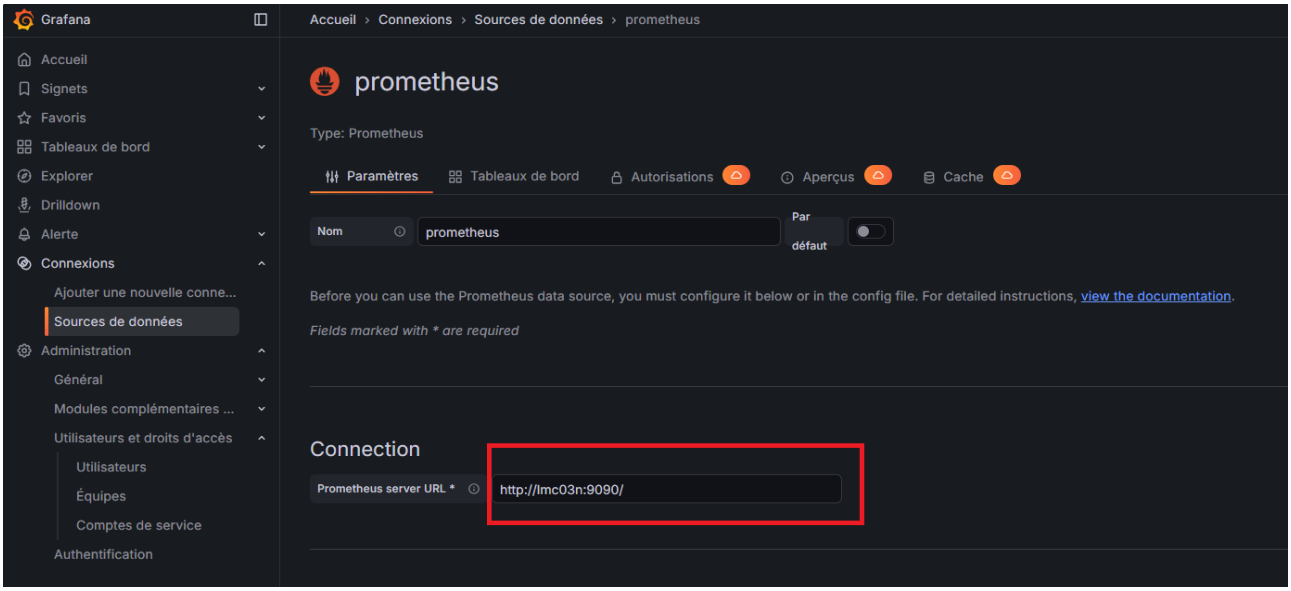
[Grafana :3000]

Installation et configuration de la collecte (Prometheus)

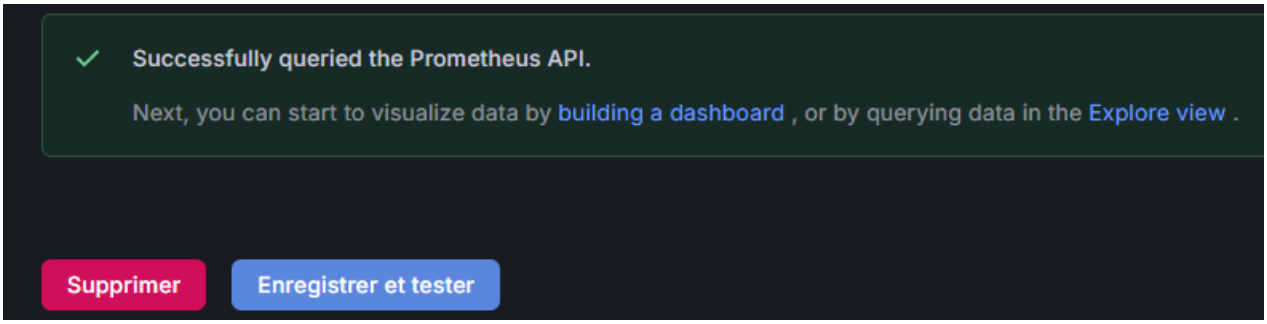
Dans mon cas j'ai utilisé Prometheus pour collecter, stocker et analyser des métriques de mon serveur netbox, et de l'esxi,

Voici le fichier de configuration /etc/prometheus/prometheus.yml

```
# Sample config for Prometheus.
global:
  scrape_interval: 30s
  evaluation_interval: 30s

  # Attach these labels to any time series or alerts when communicating with
  # external systems (federation, remote storage, Alertmanager).
  external_labels:
      monitor: 'example'

# Alertmanager configuration
alerting:
  alertmanagers:
  - static_configs:
    - targets: ['localhost: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: 'prometheus'

    # Override the global default and scrape targets from this job every 5 seconds.
    scrape_interval: 5s
    scrape_timeout: 5s

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

    static_configs:
      - targets: ['localhost:9090']

  - job_name: node
    # If prometheus-node-exporter is installed, grab stats about the local
    # machine by default.

  - job_name: 'node_exporter_netbox_server'
    static_configs:
      - targets: ['192.168.159.19:9100']

  - job_name: 'telegraf_vsphere'
    static_configs:
      - targets: ['localhost:9272']  # Telegraf expose ici les metrics
```



Il faut changer l'url du serveur avec le port Prometheus qui est 9090 et Faire Enregistrer et tester, un message de validation et envoyer :

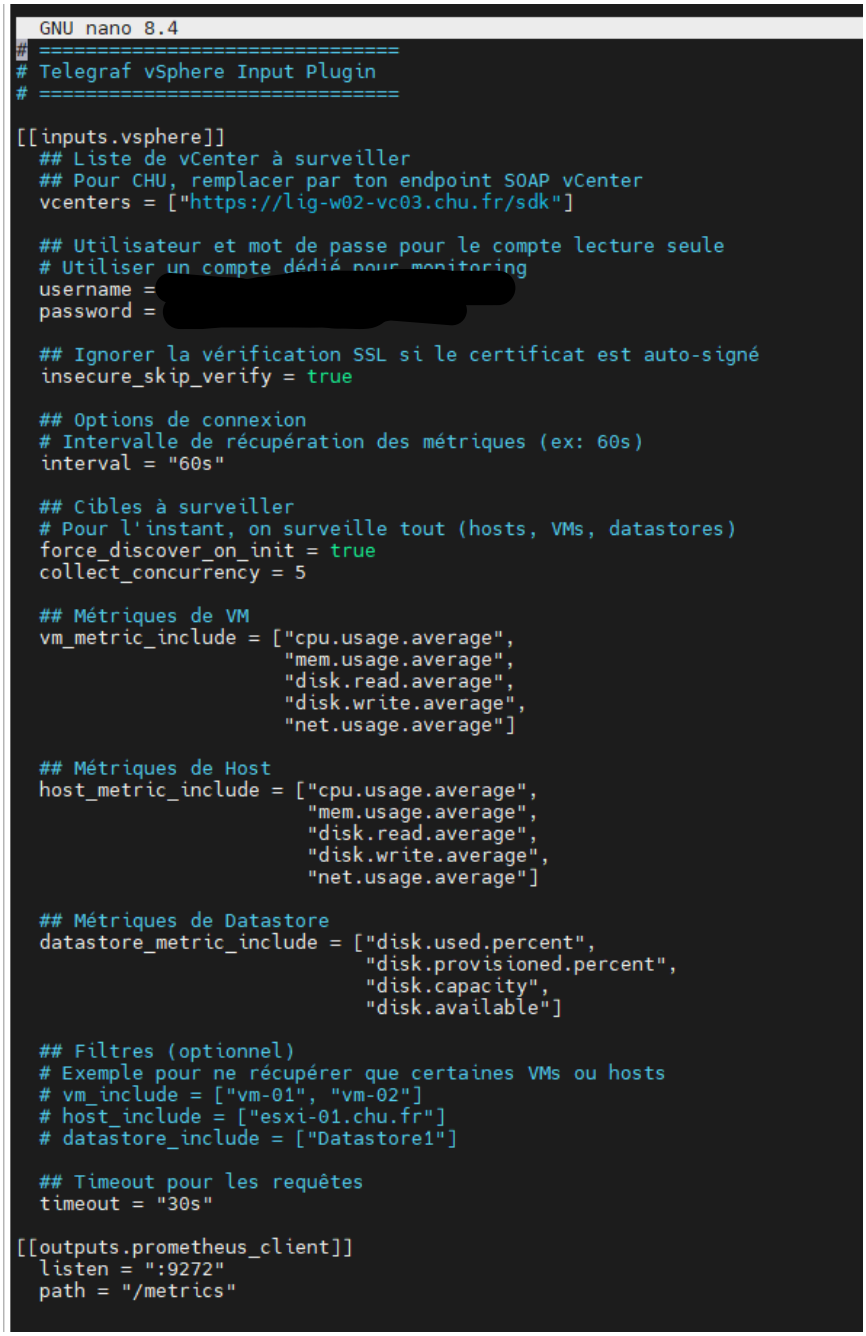


Collecte des métriques VMware (Telegraf)

Telegraf est un agent utiliser ici pour se connecter à l'API de vCenter, récuperer les performances des hyperviseurs (ESXI) et des VMs, puis les exposer dans un format lisible par Prometheus.

Fichier de config pour vsphere : /etc/telegraf/telegraf.d/vsphere.conf

Note de sécurité : L'utilisateur configuré doit être en lecture seule sur le vCenter pour limiter les risques.



```
  GNU nano 8.4
# ==============================
# Telegraf vSphere Input Plugin
# ==============================

[[inputs.vsphere]]
  ## Liste de vCenter à surveiller
  ## Pour CHU, remplacer par ton endpoint SOAP vCenter
  vcenters = ["https://lig-w02-vc03.chu.fr/sdk"]

  ## Utilisateur et mot de passe pour le compte lecture seule
  # Utiliser un compte dédié pour monitoring
  username = [redacted]
  password = [redacted]

  ## Ignorer la vérification SSL si le certificat est auto-signé
  insecure_skip_verify = true

  ## Options de connexion
  # Intervalle de récupération des métriques (ex: 60s)
  interval = "60s"

  ## Cibles à surveiller
  # Pour l'instant, on surveille tout (hosts, VMs, datastores)
  force_discover_on_init = true
  collect_concurrency = 5

  ## Métriques de VM
  vm_metric_include = ["cpu.usage.average",
                       "mem.usage.average",
                       "disk.read.average",
                       "disk.write.average",
                       "net.usage.average"]

  ## Métriques de Host
  host_metric_include = ["cpu.usage.average",
                         "mem.usage.average",
                         "disk.read.average",
                         "disk.write.average",
                         "net.usage.average"]

  ## Métriques de Datastore
  datastore_metric_include = ["disk.used.percent",
                              "disk.provisioned.percent",
                              "disk.capacity",
                              "disk.available"]

  ## Filtres (optionnel)
  # Exemple pour ne récupérer que certaines VMs ou hosts
  # vm_include = ["vm-01", "vm-02"]
  # host_include = ["esxi-01.chu.fr"]
  # datastore_include = ["Datastore1"]

  ## Timeout pour les requêtes
  timeout = "30s"

[[outputs.prometheus_client]]
  listen = ":9272"
  path = "/metrics"
```

Sécurisation : Authentification LDAP

L'authentification LDAP permet à Grafana d'utiliser un annuaire centralisé (Active Directory ou LDAP) pour gérer les utilisateurs. Cela évite la création de comptes locaux et permet une gestion centralisée des droits**.** Dans notre cas on utilise ldaps sur le port 636

Dans le fichier de configuration grafana : /etc/grafana/grafana.ini il faut rajouter des lignes pour activé ldap :

[auth.ldap]

enabled = true

config_file = /etc/grafana/ldap.toml

allow_sign_up = true

Une fois activé il faut créer le fichier de configuration /etc/grafana/ldap.toml

Voici la configuration utilisé pour ldap :

```
# =========================
# LDAP Server Configuration
# =========================

[[servers]]
# AD / LDAP host
host = "ldaps.chu.fr"
port = 636
use_ssl = true
start_tls = false
ssl_skip_verify = false
timeout = 10

# Bind user (service account) pour faire les recherches
bind_dn = "CN=svc netbox,OU=Comptes de service,DC=chu,DC=fr"
bind_password = [redacted]

# =========================
# Filtre de recherche utilisateur
# =========================
# On accepte soit sAMAccountName (ex: X13144) ou userPrincipalName (ex: X13144@chu.fr)
search_filter = "(|(sAMAccountName=%s)(userPrincipalName=%s))"

# Base DN pour la recherche des utilisateurs
search_base_dns = ["DC=chu,DC=fr"]

# =========================
# Mapping des attributs utilisateur
# =========================
[servers.attributes]
name = "givenName"
surname = "sn"
username = "sAMAccountName"
email = "mail"

# =========================
# Groupes et roles Grafana
# =========================

# Mapping des groupes AD vers roles Grafana
[[servers.group_mappings]]
group_dn = "CN=Utilisateurs_Netbox,OU=Groupes Ressources,DC=chu,DC=fr"
org_role = "Editor"
grafana_admin = false

# Pour tous les autres utilisateurs, rôle Viewer
[[servers.group_mappings]]
group_dn = "*"
org_role = "Viewer"

# =========================
# Debug LDAP (optionnel)
# =========================
# Pour activer le debug, ajoute dans /etc/grafana/grafana.ini
# [log]
# filters = ldap:debug
```

Pour que la configuration soit correcte il faudrait créer dans l'active directory un compte de service dédié à grafana avec une gestion des utilisateurs personnalisé.

Sécurisation : Passage en HTTPS via Reverse Proxy (nginx)

Pourquoi un reverse proxy

Par defaut, grafana communique en http (port 3000). Pour chiffrer les échanges (HTTPS) sans modifier le cœur de Grafana, nous utilisons Nginx comme Reverse Proxy. Il gere la terminaison TLS (certificats SSL) et redirige le trafic sécurisé vers Grafana en local.

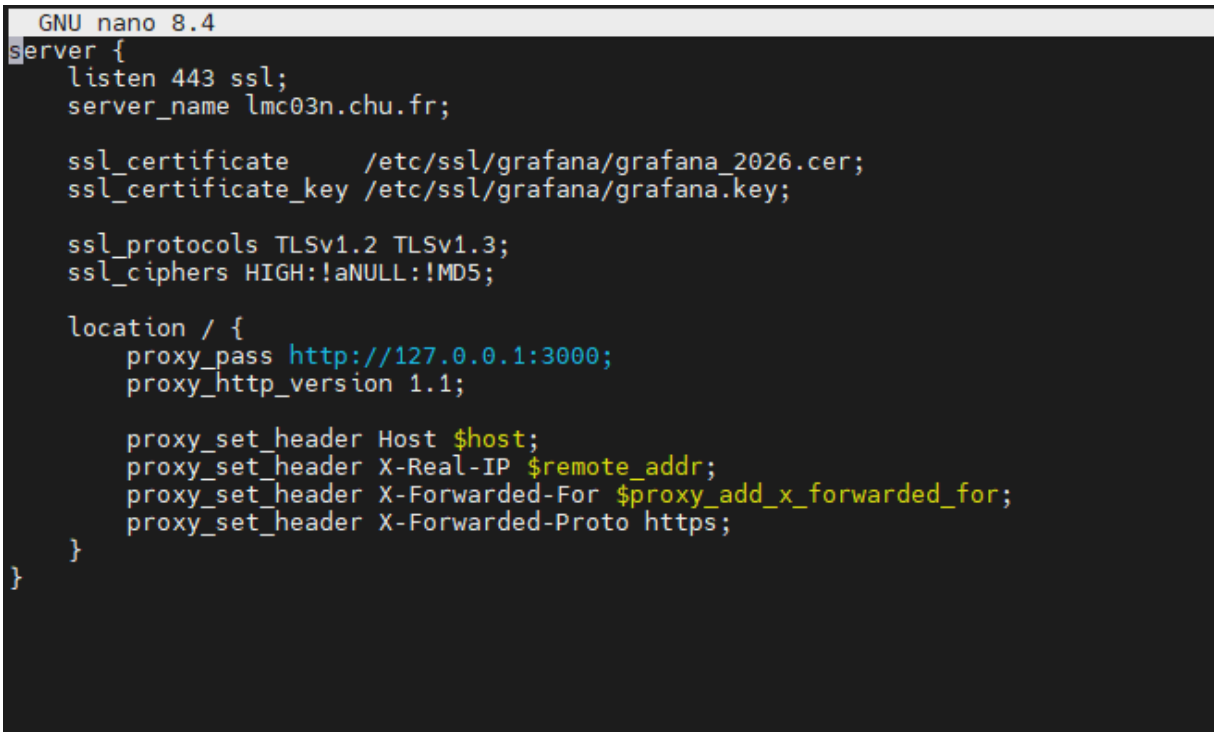
Installation et configuration de Nginx :

sudo apt update

sudo apt install nginx -y

systemctl status nginx

Configuration nginx pour utilisation de https dans le fichier nano /etc/nginx/sites-available/grafana



Ensuite il faut l'active avec cette commande :

sudo ln -s /etc/nginx/sites-available/grafana /etc/nginx/sites-enabled/

Et tester et redémarrer :

sudo nginx –t

sudo systemctl restart nginx

Configuration de Grafana :

Dans le fichier de configuration il faut laisser ça c'est nginx qui s'occupe de tout :

```
#################################### Server ####################################
[server]
protocol = http
http_port = 3000
http_addr = 127.0.0.1
# Protocol to use (http, https, socket)

# Port to listen on
```

Redémarrer le service :

sudo systemctl restart grafana-server

Gestion des certificats SSL

Pour obtenir un certificat valide signé par l'autorité de certification (PKI) de l'organisation, une demande de signature de certificat (CSR) est générée via OpenSSL.

J'ai créé un fichier dédier au ssl de grafana dans le dossier /etc/ssl/grafana ou je vais stocker mes clés.

Creation de clé privée :

openssl genrsa -out grafana.key 2048

Créer le certificat à partir d'un fichier texte paramètre :

[req]

default_bits = 2048

prompt = no

default_md = sha256

req_extensions = req_ext

distinguished_name = dn

[dn]

C=FR

ST=France

L=Limoges

O=CHU de Limoges

OU=DSI

emailAddress= dsi.systemes@chu-limoges.fr

CN = lmc03n.chu.fr

[req_ext]

subjectAltName = @alt_names

[alt_names]

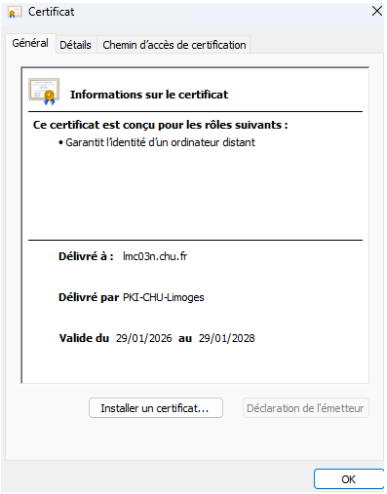
DNS.1 = grafana.chu.fr

Commande :

openssl req -new -newkey rsa:2048 -nodes -keyout grafana.key -out grafana.csr -config /tmp/certificat/param_grafana.txt

Ce fichier .csr permettra à créer le certificat a partie du PKI du chu, ce certificat durera 2 ans à partir de la création.

On obtient un fichier grafana_2026.cer



Une fois le fichier grafana.key et grafana_2026.cer dans le fichier /etc/ssl/grafana et que la configuration pointe les deux fichiers, on utilise deux commande pour tester c'est les clés sont les mêmes

openssl rsa -noout -modulus -in grafana.key | openssl md5

openssl x509 -noout -modulus -in grafana_2026.cer | openssl md5

On peut tester sur le navigateur web :

https://grafana.chu.fr