COMPTE RENDU TECHNIQUE - AP 14

Objet : Mise en place d'une solution de supervision et de sauvegarde sécurisée Entreprise : MenuiMétal

Techniciens : Killian Goncalves & Cristopher Boni Fuentes Groupe : 8 Date : Février 2026

1. Introduction

Dans le cadre de la modernisation de l'infrastructure de la société MenuiMétal, nous avons été missionnés par le responsable informatique, M. Olivier Lepage, pour répondre aux problématiques de prévention contre les ransomwares.

Notre travail s'est concentré sur deux axes :

1. **La supervision métrique (Mission 1 - Prometheus/Grafana).**
2. **La sécurisation des sauvegardes (Mission 2 - WebDAV).**

Ce document détaille la mise en œuvre des prérequis (Mission 0) et la réalisation complète de la solution de sauvegarde (Mission 2).

Mission 0 : Prérequis et contraintes

1. Organisation et Planification

Conformément aux exigences de la mission, la première étape a consisté à organiser le travail en équipe.

- **Répartition des tâches :** Utilisation d'un diagramme de Gantt(lien Gantt) pour planifier les durées et la répartition entre les membres de l'équipe (Killian et Cristopher).
- **Mise à jour du schéma réseau :** Intégration des nouvelles machines virtuelles (Grafana, Prometheus, WebDAV) dans la topologie existante(lien shema).

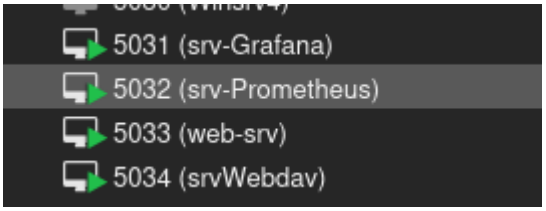
2. Infrastructure Réseau et Commutation

Le groupe **8** utilise des VLANs spécifiques basés sur le chiffre **8**.

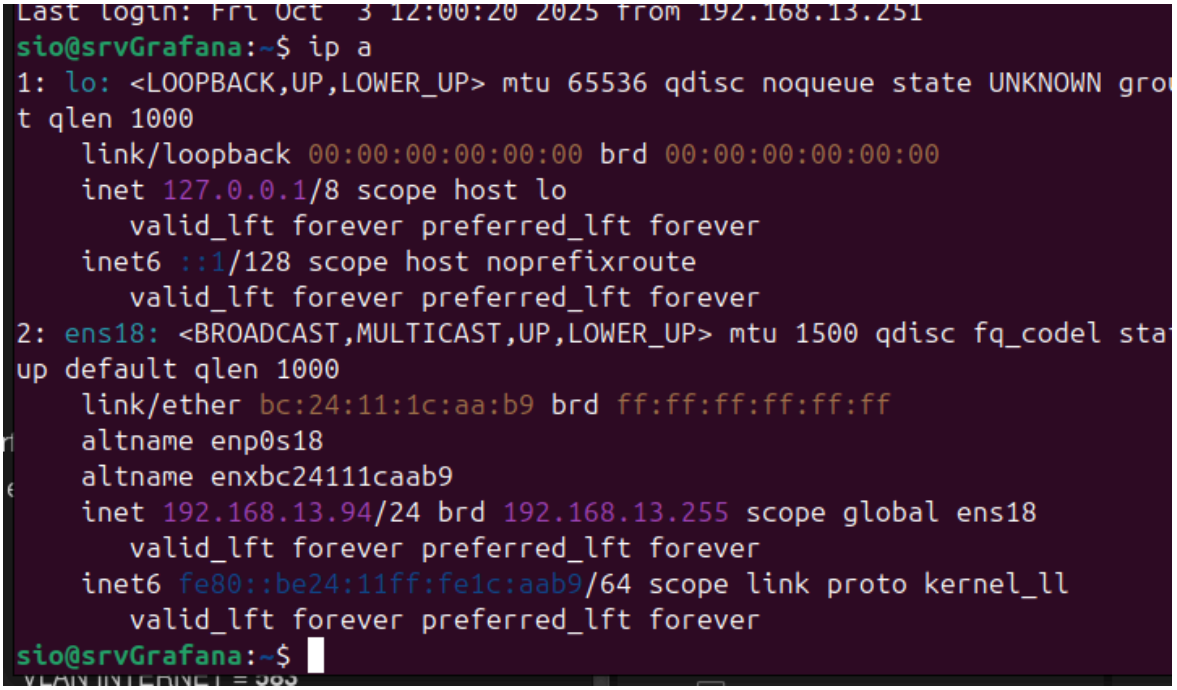
- **Configuration des switches :** Les switches Cisco Catalyst 2960G et HP ont été configurés pour supporter les VLANs suivants :
    - **VLAN 581 (LAN) :** Réseau local (192.168.11.0/24).
    - **VLAN 582 (DMZ) :** Zone démilitarisée pour le serveur WebDAV (192.168.12.0/24).
    - **VLAN 580 (GESTION) :** Réseau de supervision (192.168.13.0/24).

3. Préparation de l'environnement de virtualisation (Proxmox)

Quatre machines virtuelles ont été préparées par clonage sous Proxmox pour répondre aux besoins des missions suivantes :



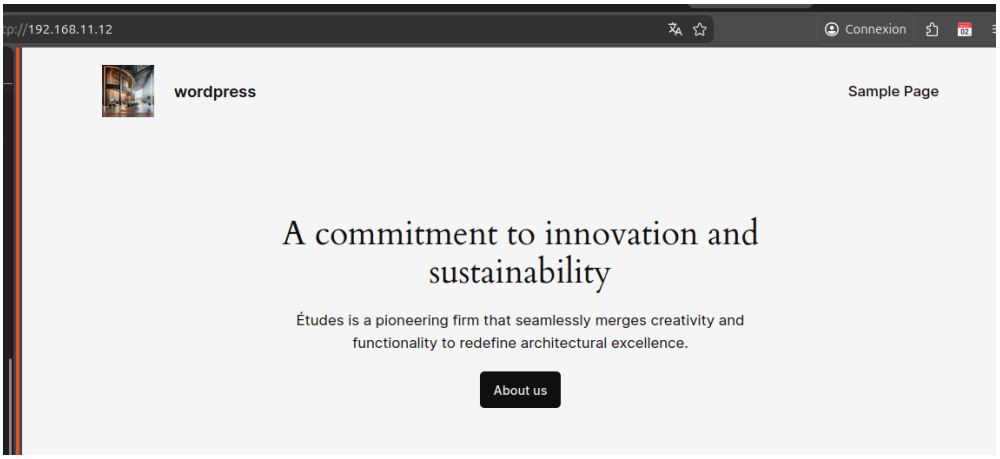
Nom de la VM	ID Proxmox	VLAN	Rôle
srv-Grafana	5031	GESTION (580)	Dashboarding et visualisation(192.168.13.94



srv-Prometheus	5032	GESTION (580)	Collecte des métriques (192.168.13.93

```
root@srvPrometheus:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
up default qlen 1000
    link/ether bc:24:11:45:c6:36 brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    altname enxbc241145c636
    inet 192.168.13.93/24 brd 192.168.13.255 scope global ens18
       valid_lft forever preferred_lft forever
    inet6 fe80::be24:11ff:fe45:c636/64 scope link proto kernel_ll
       valid_lft forever preferred_lft forever
root@srvPrometheus:~#
```

web-srv	5033	LAN (581)	Serveur WordPress (192.168.11.12)



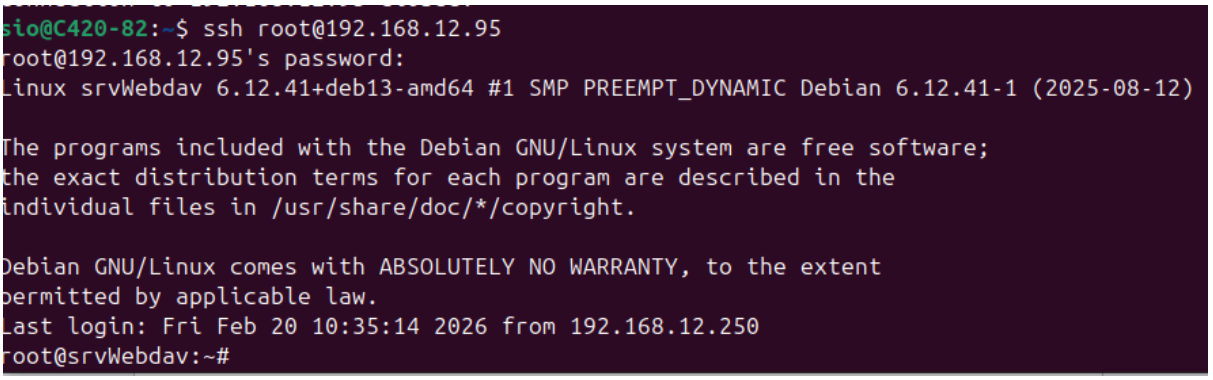
srvWebdav	5034	DMZ (582)	Sauvegarde distante (192.168.12.95)

```
sio@C420-82:~$ ssh root@192.168.12.95
root@192.168.12.95's password:
Linux srvWebdav 6.12.41+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.41-1 (2025-

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Feb 20 14:49:56 2026 from 192.168.12.251
root@srvWebdav:~#
```

1. Test de connexion SSH

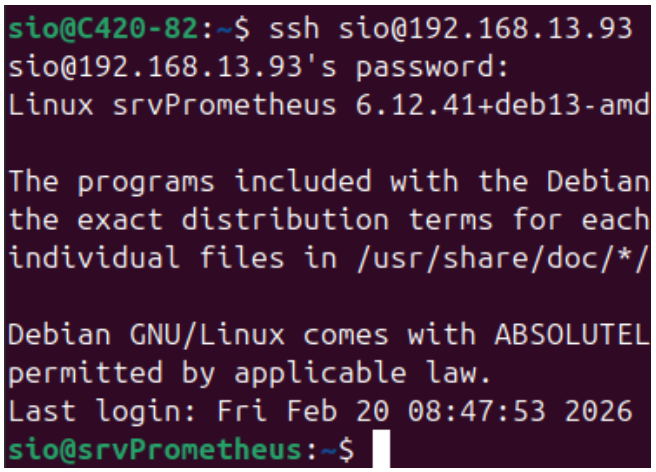
L'objectif est de prouver qu'on peut administrer le serveur WebDAV et des autres vms (**192.168.12.95**) à distance depuis une autre machine.

ssh sio@192.168.12.95

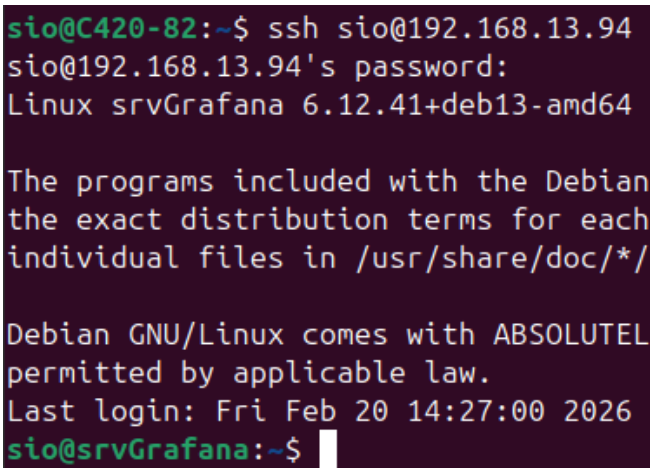


Serveur prometheus et Grafana

Prometheus



Grafana

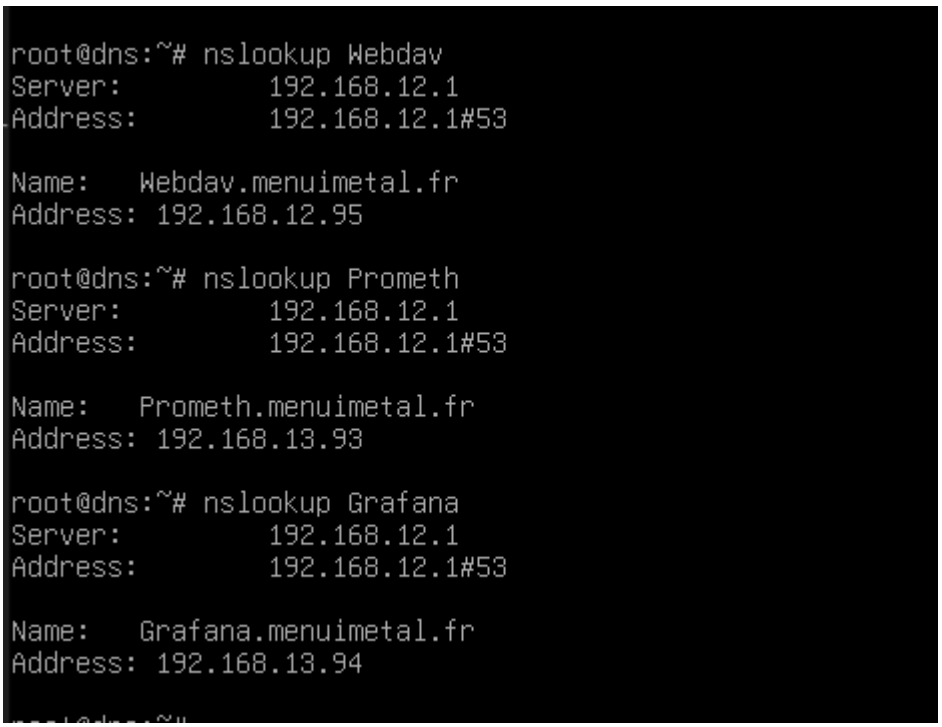


2. Test de référencement DNS

Il faut prouver que le serveur DNS (**192.168.12.1**) fait bien le lien entre le nom `nomDeLaVm.menuimetal.fr` et l'Ip du serveur.

nslookup Nomdelavm

On fait un nslookup pour tous les vms(Grafana,Prometheus et Webdav) pour vérifier le référencement des vms dans le serveur dns



Mission 1 : PROMETHEUS & GRAFANA

1. À quoi sert Prometheus ?

C'est un outil pour surveiller la santé des serveurs et du réseau, et envoyer des alertes en cas de problème.

2. Quel est son mode de fonctionnement ?

Il va chercher les données (métriques) directement sur les machines à intervalles réguliers (Pull) et les stocke dans sa base de données.

3. Métrique et Exporters

- **Métrique :** Une valeur numérique mesurée.
- **Exporter :** Un logiciel qui traduit les données d'une machine pour que Prometheus puisse les lire.

4. Différence Pull et Push

- **Pull :** Prometheus va chercher les données (mode par défaut).
- **Push :** La machine envoie les données à Prometheus (via Pushgateway).

5. Apport de Grafana

Grafana récupère les données de Prometheus pour afficher des graphiques et tableaux de bord (dashboards) faciles à lire.

6. Signification de « scrape_interval »

C'est la fréquence de collecte des données.

7. Différence 'job_name' et 'targets'

- **job_name :** Le nom du groupe de machines.
- **targets :** L'adresse IP exacte de la machine à surveiller dans ce groupe.

8. Installer et configurer Prometheus et Grafana

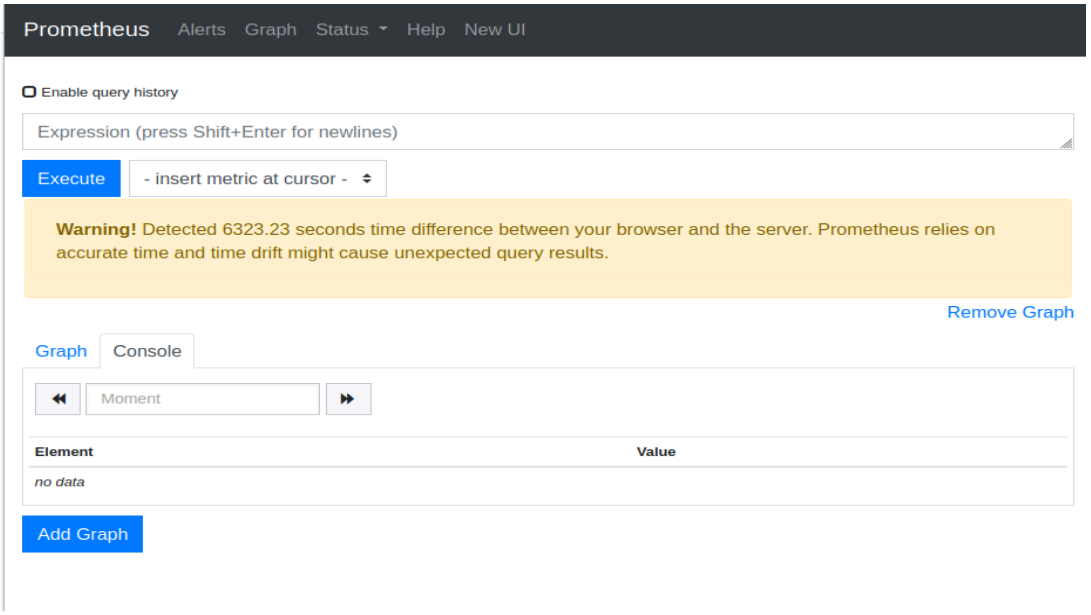
Prometheus :

```
root@srvPrometheus:~# apt install prometheus -y
```

Installation du service : J'utilise le gestionnaire de paquets `apt` pour installer Prometheus sur le serveur Linux. L'option `-y` permet d'accepter automatiquement toutes les confirmations pour une installation plus rapide.

```
system/prometheus.service'.
Traitement des actions différées (« triggers ») pour man-db (2.13.1-1) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.41-12) ...
root@srvPrometheus:~# systemctl status prometheus
● prometheus.service - Monitoring system and time series database
     Loaded: loaded (/usr/lib/systemd/system/prometheus.service; enabled; preset: enabled)
     Active: active (running) since Thu 2026-02-19 15:14:05 CET; 24s ago
 Invocation: a36556ea81164e659a5928fd04859127
       Docs: https://prometheus.io/docs/introduction/overview/
             man:prometheus(1)
   Main PID: 2777 (prometheus)
      Tasks: 6 (limit: 2303)
     Memory: 24.6M (peak: 25.9M)
        CPU: 166ms
     CGroup: /system.slice/prometheus.service
             └─2777 /usr/bin/prometheus

févr. 19 15:14:05 srvPrometheus prometheus[2777]: ts=2026-02-19T14:14:05.564Z level=info caller=tls_>
févr. 19 15:14:05 srvPrometheus prometheus[2777]: ts=2026-02-19T14:14:05.564Z caller=head.go:793 lev>
févr. 19 15:14:05 srvPrometheus prometheus[2777]: ts=2026-02-19T14:14:05.565Z caller=head.go:830 lev>
févr. 19 15:14:05 srvPrometheus prometheus[2777]: ts=2026-02-19T14:14:05.567Z caller=main.go:1159 le>
févr. 19 15:14:05 srvPrometheus prometheus[2777]: ts=2026-02-19T14:14:05.567Z caller=main.go:1162 le>
févr. 19 15:14:05 srvPrometheus prometheus[2777]: ts=2026-02-19T14:14:05.567Z caller=main.go:1344 le>
févr. 19 15:14:05 srvPrometheus prometheus[2777]: ts=2026-02-19T14:14:05.568Z caller=main.go:1381 le>
```

Vérification du fonctionnement : Après l'installation, je vérifie l'état du service avec la commande `systemctl status prometheus`. On peut voir que le service est "active (running)", ce qui confirme que Prometheus est correctement démarré et prêt à collecter des données.



Accès à l'interface de gestion : Je teste l'accès à l'interface graphique via un navigateur web. L'affichage de la console Prometheus confirme que le serveur est accessible sur le réseau. *Note : Petit problème de décalage horaire*

Grafana :

```
sio@srvGrafana:~$ su -
Mot de passe :
root@srvGrafana:~# apt-get install -y adduser libfontconfig1 musl
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
adduser est déjà la version la plus récente (3.152).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :
  libcbor0.8 libfuse2 libicu72 libldap-2.5-0 libnsl2 libperl5.36 libpython3.11-minimal
  libpython3.11-stdlib perl-modules-5.36 python3-autocommand python3-httplib2 python3-inflect
  python3-jaraco.context python3-jaraco.functools python3-more-itertools python3-pkg-resources
  python3-pycurl python3-pyparsing python3-pysimplesoap python3-six python3-typeguard
  python3-typing-extensions python3.11 python3.11-minimal
Veuillez utiliser « apt autoremove » pour les supprimer.
Les paquets supplémentaires suivants seront installés :
  fontconfig-config fonts-dejavu-core fonts-dejavu-mono
Les NOUVEAUX paquets suivants seront installés :
  fontconfig-config fonts-dejavu-core fonts-dejavu-mono libfontconfig1 musl
0 mis à jour, 5 nouvellement installés, 0 à enlever et 58 non mis à jour.
Il est nécessaire de prendre 2 467 kB dans les archives.
Après cette opération, 5 632 ko d'espace disque supplémentaires seront utilisés.
Réception de : 1 http://deb.debian.org/debian trixie/main amd64 fonts-dejavu-mono all 2.37-8 [489 kB]
Réception de : 2 http://deb.debian.org/debian trixie/main amd64 fonts-dejavu-core all 2.37-8 [840 kB]
Réception de : 3 http://deb.debian.org/debian trixie/main amd64 fontconfig-config amd64 2.15.0-2.3 [3
18 kB]
Réception de : 4 http://deb.debian.org/debian trixie/main amd64 libfontconfig1 amd64 2.15.0-2.3 [392
kB]
Réception de : 5 http://deb.debian.org/debian trixie/main amd64 musl amd64 1.2.5-3 [427 kB]
2 467 ko réceptionnés en 0s (9 294 ko/s)
```

Préparation de l'environnement : Avant d'installer Grafana, j'installe les dépendances nécessaires (paquets comme `libfontconfig1` et `musl`) à l'aide de la commande `apt-get install`. Ces composants sont indispensables au bon fonctionnement de l'interface graphique de Grafana sur Debian.

```
root@srvGrafana:~# wget https://dl.grafana.com/grafana-enterprise/release/12.3.3/grafana-enterprise_1
2.3.3_21957728731_linux_amd64.deb
```

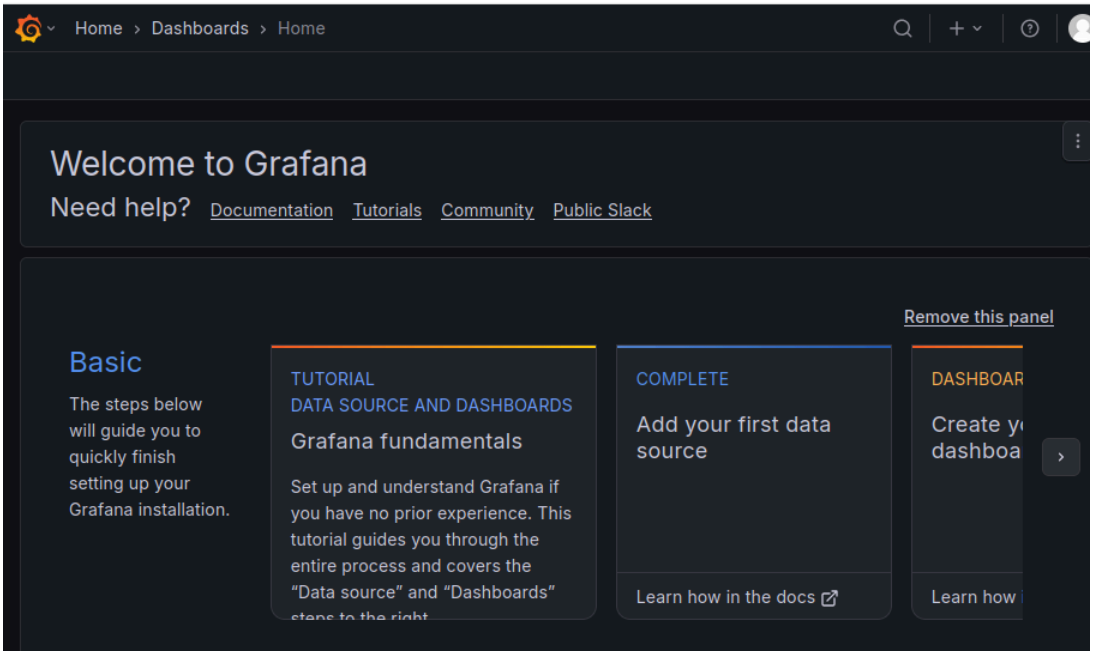
Récupération de l'installateur : J'utilise la commande `wget` pour télécharger directement le paquet d'installation officiel de Grafana Enterprise (version 11.2.3) depuis les serveurs de l'éditeur. Ce fichier au format `.deb` permettra une installation propre et locale sur le serveur.

installation de Grafana:

```
root@srvGrafana:~#  dpkg -i grafana-enterprise_12.3.3_21957728731_linux_amd64.deb
Sélection du paquet grafana-enterprise précédemment désélectionné.
(Lecture de la base de données... 45371 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de grafana-enterprise_12.3.3_21957728731_linux_amd64.deb ...
Dépaquetage de grafana-enterprise (12.3.3) ...
Paramétrage de grafana-enterprise (12.3.3) ...
### NOT starting on installation, please execute the following statements to configure grafana to sta
rt automatically using systemd
 sudo /bin/systemctl daemon-reload
 sudo /bin/systemctl enable grafana-server
### You can start grafana-server by executing
 sudo /bin/systemctl start grafana-server
root@srvGrafana:~#
```

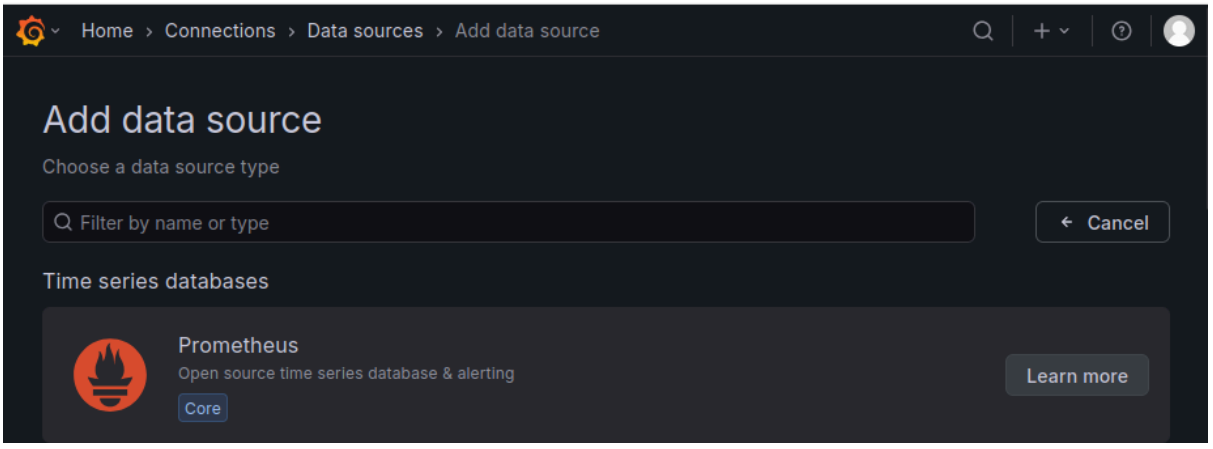
Installation du logiciel : J'utilise la commande `dpkg -i` pour installer le paquet Debian précédemment téléchargé. Cette commande décompresse et installe les fichiers de Grafana

sur le système. Le terminal m'indique ensuite les commandes `systemctl` à exécuter pour relancer et activer le service au démarrage du serveur.

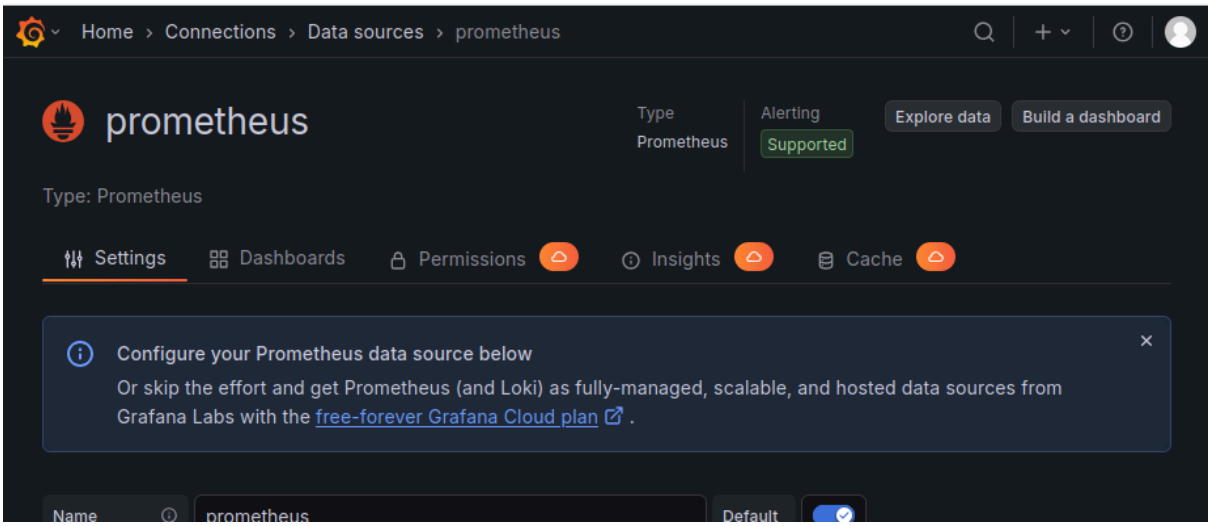


Finalisation et accès Web : Une fois le service démarré, j'accède à l'interface de Grafana via un navigateur (port 3000). La page "Welcome to Grafana" confirme que l'installation est réussie. À partir d'ici, je vais pouvoir configurer Prometheus comme source de données (Data Source) pour créer des tableaux de bord personnalisés.

9. Visualiser les données de Prometheus sur un tableau de bord via Grafana

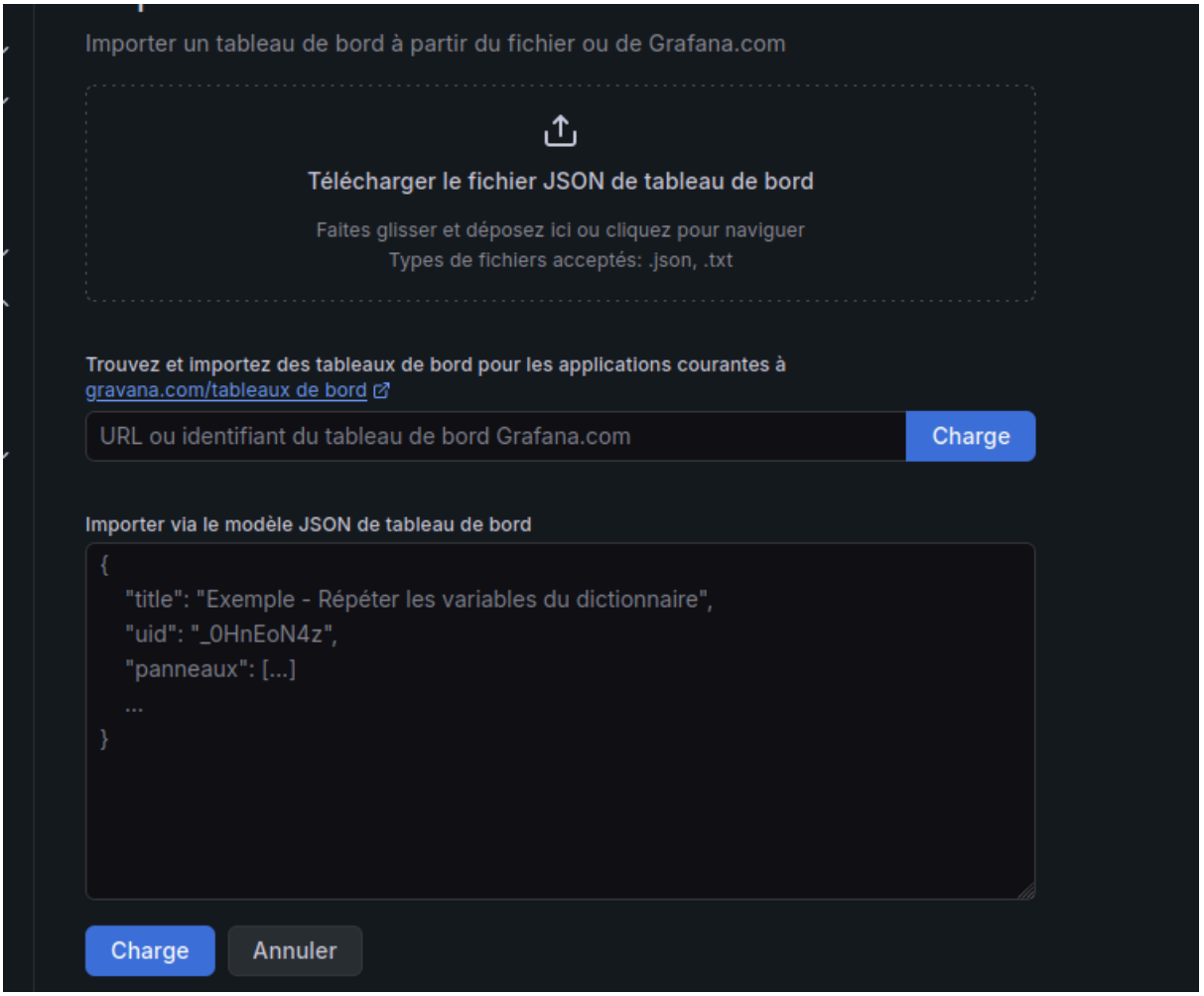


Ajout de la source de données (Data Source) : Dans les paramètres de Grafana, je me rends dans la section "Data Sources" pour connecter mon serveur de métriques. Je sélectionne Prometheus dans la liste des bases de données de séries temporelles (Time series databases) pour indiquer à Grafana d'où proviennent les informations à afficher.



Paramétrage de la connexion : Je configure ensuite la source de données en renseignant l'adresse URL du serveur Prometheus (généralement `http://192.168.13.93:9090`). Cette étape est indispensable pour que Grafana puisse interroger Prometheus en temps réel et transformer les données brutes en graphiques lisibles sur un tableau de bord.

Création du tableau de bord : Une fois la source de données connectée, je fais importer un nouveau tableau de bord



Utilisation de modèles (JSON) : Pour gagner du temps et bénéficier de visuels professionnels, Grafana permet d'importer des tableaux de bord via des fichiers JSON

Utilisation de modèles (JSON) : Pour gagner du temps et bénéficier de visuels professionnels, Grafana permet d'importer des tableaux de bord via des fichiers JSON

Importer un tableau de bord

Importer un tableau de bord à partir du fichier ou de Grafana.com

Options

Nom

Node Exporter Full

Dossier

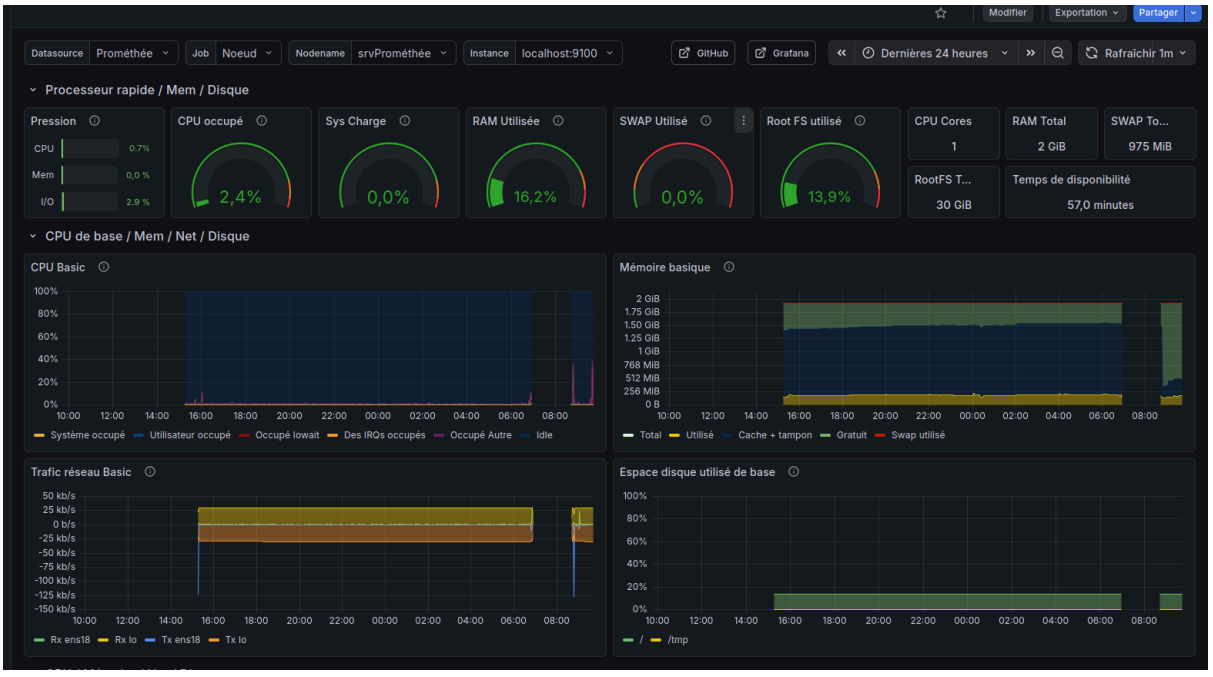
Tableaux de bord

Identificateur unique (UID)

L'identifiant unique (UID) d'un tableau de bord peut être utilisé pour identifier de manière unique un tableau de bord entre plusieurs installations de Grafana. L'UID permet d'avoir des URL cohérentes pour accéder aux tableaux de bord, de sorte que la modification du titre d'un tableau de bord ne cassera aucun lien marqué vers ce tableau de bord.

192.168.13.93

Import Annuler



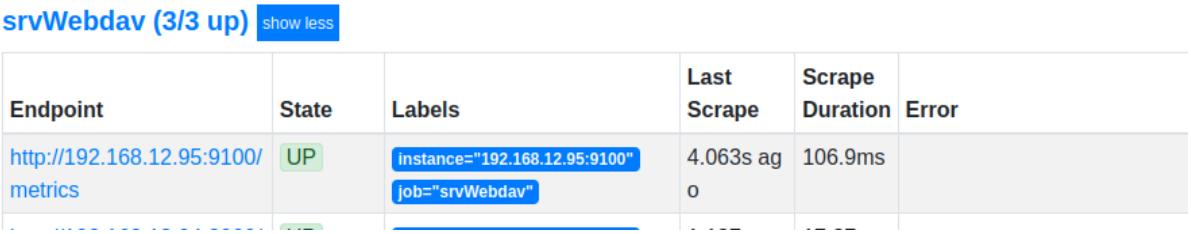
10. A l’aide de node_exporter faire en sorte de visualiser les métriques de vos serveurs Web sur le tableau de bord de Grafana

```
Linux srvWebdav 6.12.41+deb13-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.12.41-1 (2025-08-12) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Feb 20 09:54:27 2026 from 192.168.12.251
root@srvWebdav:~# apt update && sudo apt install prometheus-node-exporter -y
Atteint : 1 http://deb.debian.org/debian trixie InRelease
Réception de : 2 http://deb.debian.org/debian trixie-updates InRelease [47,3 kB]
Réception de : 3 http://security.debian.org/debian-security trixie-security InRelease [43,4 kB]
Réception de : 4 http://security.debian.org/debian-security trixie-security/main Sources [136 kB]
Réception de : 5 http://security.debian.org/debian-security trixie-security/main amd64 Packages [11
kB]
Réception de : 6 http://security.debian.org/debian-security trixie-security/main Translation-en [72
 kB]
411 ko réceptionnés en 0s (1 425 ko/s)
53 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
Installation de :
  prometheus-node-exporter
```

Installation de l'agent de métriques : Sur le serveur Web (`srvWebdav`), j'installe `prometheus-node-exporter`. C'est un petit programme qui va récolter les statistiques matérielles du serveur (utilisation du processeur, RAM, disque) pour les mettre à disposition de Prometheus.

```
# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
crape_configs:
# The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
- job_name: 'prometheus'
- job_name: 'grafana'
- job_name: 'srvWebdav'
  # 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']
    - targets: ['192.168.13.94:3000']
    - targets: ['192.168.12.95:9100']
- job_name: node
  # If prometheus-node-exporter is installed, grab stats about the local
  # machine by default.
  static_configs:
    - targets: ['localhost:9100', '192.168.12.95:9100', '192.168.13.94:3000']
```

Déclaration de la nouvelle cible (Target) : Je modifie le fichier de configuration de Prometheus pour lui indiquer d'aller "scanner" (scraper) le serveur Web. J'ajoute l'adresse IP du serveur suivi du port 9100 (le port par défaut de Node Exporter). Cela permet de centraliser toutes les données sur mon serveur de monitoring.



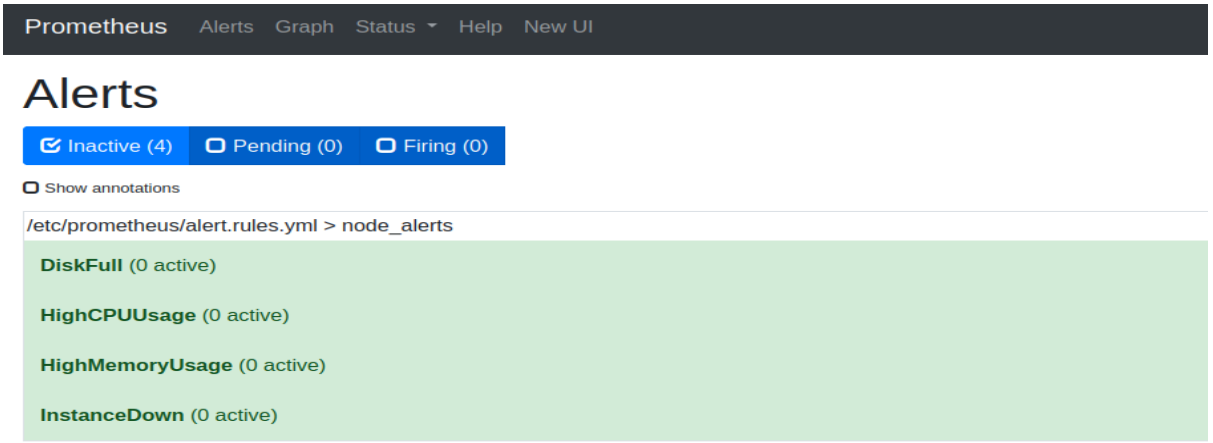
Validation de la remontée de données : Dans l'interface de Prometheus, je vérifie que le "Endpoint" de mon serveur Web est bien marqué comme "UP". Cela confirme que la communication réseau est établie et que Prometheus reçoit correctement les métriques envoyées par `srvWebdav`.

11. Faire en sorte que Prometheus remonte des alertes sur des services arrêtés ou des métriques anormales.

```
 GNU nano 8.4                    /etc/prometheus/alert.rules.yml *
roups:
 - name: node_alerts
   rules:

     # Alerte si serveur down
     - alert: InstanceDown
       expr: up == 0
       for: 1m
       labels:
         severity: critical
       annotations:
         summary: "Serveur DOWN"
         description: "Le serveur {{ $labels.instance }} est inaccessible."

     # Alerte si CPU > 80%
     - alert: HighCPUUsage
       expr: 100 - (avg by(instance) (rate(node_cpu_seconds_total{mode="idle"}[5m])) * 100) > 80
       for: 2m
       labels:
         severity: warning
       annotations:
         summary: "CPU élevé"
         description: "CPU > 80% sur {{ $labels.instance }}"

     # Alerte si RAM > 90%
     - alert: HighMemoryUsage
       expr: (1 - (node_memory_MemAvailable_bytes / node_memory_MemTotal_bytes)) * 100 > 90
       for: 2m
       labels:
         severity: warning
       annotations:
         summary: "Mémoire élevée"
         description: "RAM > 90% sur {{ $labels.instance }}"

     # Alerte si disque > 90%
     - alert: DiskFull
       expr: (node_filesystem_avail_bytes / node_filesystem_size_bytes) * 100 < 10
       for: 2m
       labels:
         severity: critical
       annotations:
         summary: "Disque presque plein"
         description: "Moins de 10% libre sur {{ $labels.instance }}"
```

Configuration des seuils d'alerte : Je crée un fichier de règles (`alert.rules.yml`) pour définir les événements critiques qui doivent déclencher une alerte. J'utilise le langage de requête PromQL pour surveiller des indicateurs précis :

- InstanceDown : si un serveur ne répond plus.
- HighCPUUsage : si la charge processeur dépasse 80%.
- HighMemoryUsage : si la RAM utilisée dépasse 90%.
- DiskFull : s'il reste moins de 10% d'espace disque.



Supervision active : La console Prometheus affiche la liste des alertes configurées. On peut voir ici que toutes les règles (CPU, RAM, Disque) sont actives et prêtes à prévenir l'administrateur en cas de problème sur le réseau.

```
reated symlink '/etc/systemd/system/multi-user.target.wants/prometheus-alertmanager.service' → '/us
/lib/systemd/system/prometheus-alertmanager.service'.
raitement des actions différées (« triggers ») pour man-db (2.13.1-1) ...
oot@srvPrometheus:~# systemctl enable prometheus-alertmanager
ynchronizing state of prometheus-alertmanager.service with SysV service script with /usr/lib/system
/systemd-sysv-install.
xecuting: /usr/lib/systemd/systemd-sysv-install enable prometheus-alertmanager
oot@srvPrometheus:~# systemctl start prometheus-alertmanager
oot@srvPrometheus:~# systemctl status prometheus-alertmanager
 prometheus-alertmanager.service - Alertmanager for prometheus
     Loaded: loaded (/usr/lib/systemd/system/prometheus-alertmanager.service; enabled; preset: enab>
     Active: active (running) since Fri 2026-02-20 16:21:42 CET; 2min 17s ago
Invocation: bd5214e36c85436f8a1f9e8b6607dd36
       Docs: https://prometheus.io/docs/alerting/alertmanager/
   Main PID: 10106 (prometheus-aler)
      Tasks: 6 (limit: 2303)
     Memory: 15M (peak: 15.4M)
        CPU: 202ms
     CGroup: /system.slice/prometheus-alertmanager.service
             └─10106 /usr/bin/prometheus-alertmanager

évr. 20 16:21:42 srvPrometheus prometheus-alertmanager[10106]: time=2026-02-20T15:21:42.962Z level>
évr. 20 16:21:42 srvPrometheus prometheus-alertmanager[10106]: time=2026-02-20T15:21:42.964Z level>
évr. 20 16:21:42 srvPrometheus prometheus-alertmanager[10106]: time=2026-02-20T15:21:42.969Z level>
évr. 20 16:21:42 srvPrometheus prometheus-alertmanager[10106]: time=2026-02-20T15:21:42.974Z level>
évr. 20 16:21:43 srvPrometheus prometheus-alertmanager[10106]: time=2026-02-20T15:21:43.065Z level>
évr. 20 16:21:43 srvPrometheus prometheus-alertmanager[10106]: time=2026-02-20T15:21:43.066Z level>
évr. 20 16:21:43 srvPrometheus prometheus-alertmanager[10106]: time=2026-02-20T15:21:43.085Z level>
évr. 20 16:21:43 srvPrometheus prometheus-alertmanager[10106]: time=2026-02-20T15:21:43.086Z level>
évr. 20 16:21:44 srvPrometheus prometheus-alertmanager[10106]: time=2026-02-20T15:21:44.975Z level>
évr. 20 16:21:52 srvPrometheus prometheus-alertmanager[10106]: time=2026-02-20T15:21:52.978Z level>

oot@srvPrometheus:~#
```

Installation de l'agent distant : Je me connecte sur le serveur cible `srvWebdav` pour y installer le paquet `prometheus-node-exporter`. Cet agent est indispensable car il permet d'extraire les données matérielles du serveur web pour les mettre à disposition du serveur de monitoring.

```yaml
# Alertmanager configuration
alerting:
  alertmanagers:
  - static_configs:
    - targets: ['localhost:9093']
```

Déclaration des cibles dans Prometheus : Je modifie le fichier de configuration principal de Prometheus pour y ajouter les différentes machines à surveiller. On peut voir ici les "jobs" créés pour Prometheus.

Supervision des alertes via l'interface Grafana : Cette image montre le récapitulatif des règles d'alertes directement dans Grafana. On peut voir que les quatre alertes configurées (InstanceDown, HighCPUUsage, HighMemoryUsage et DiskFull) sont à l'état "Normal", ce qui signifie que tous les serveurs surveillés respectent les seuils

de sécurité définis. Cette vue centralisée permet à l'administrateur de vérifier d'un seul coup d'œil la santé globale de l'infrastructure.

Mission 2 : Étude et mise en place d'une solution

1. Étude et Justification de la solution WebDAV

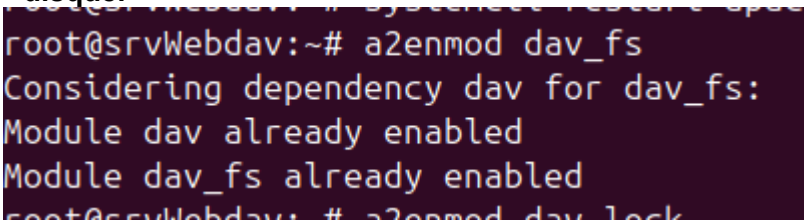
Conformément aux attentes de Monsieur Lepage, la mise en place du protocole WebDAV pour sauvegarder les bases de données de GLPI et les articles du CMS est une solution parfaitement adaptée à l'infrastructure de MenuiMétal :

- Gratuité : Le service repose sur Apache2, une solution Open Source totalement gratuite.
- Sécurité avancée : Les échanges ont été chiffrés grâce à la génération de certificats SSL (HTTPS). De plus, nous avons couplé une restriction par adresse IP à une authentification forte par mot de passe.
- Authentification : L'accès est protégé par un fichier d'utilisateurs (`.htpasswd`), mais l'architecture permet d'évoluer vers une authentification basée sur une base de données MySQL/MariaDB si besoin.
- Performance : L'utilisation du système de fichiers natif de Debian 12 permet des transferts fluides et fiables.

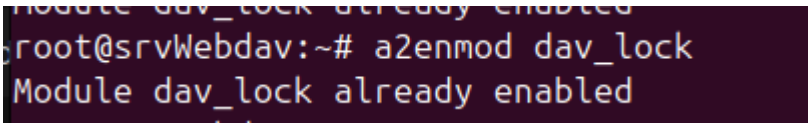
2. Rôle des modules WebDAV

Pour faire fonctionner le service sur notre serveur Apache, trois modules ont été activés:

- `dav` : C'est le module principal qui active les méthodes HTTP spécifiques au protocole WebDAV.
- `dav_fs` : Il sert de "backend" (moteur) pour gérer le système de fichiers, permettant au serveur d'écrire et de lire les fichiers de sauvegarde sur le disque.



- `dav_lock` : Il gère le système de verrouillage des fichiers. Cela évite que deux processus modifient le même fichier de sauvegarde simultanément, prévenant ainsi la corruption des données.

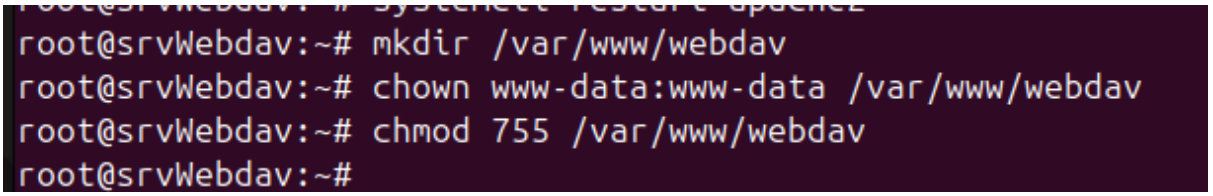


3. Mise en œuvre technique

Le serveur WebDAV a été déployé dans la DMZ (VLAN 582) avec l'adresse IP 192.168.12.95.

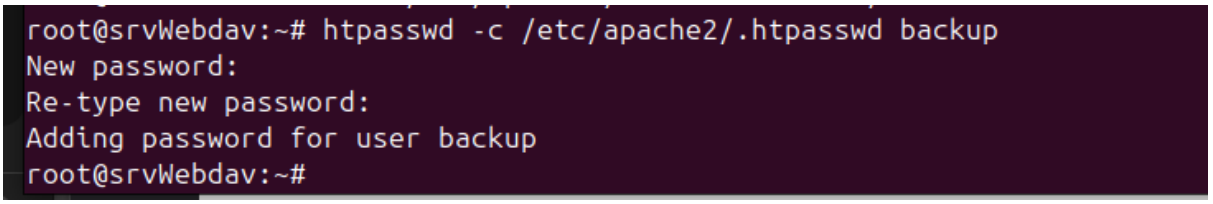
A. Configuration du service

Nous avons créé l'arborescence `/var/www/webdav` avec les bons droits (`chown www-data:www-data`), activé les modules nécessaires et créé l'utilisateur `backup` pour l'authentification.



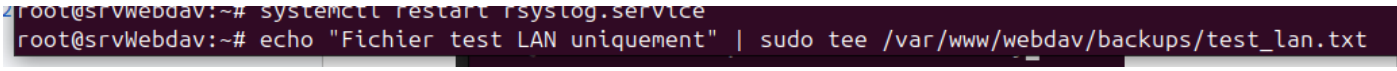
Création de l'utilisateur 'adminbackup'

sudo htpasswd -c /etc/apache2/.htpasswd admin_backup

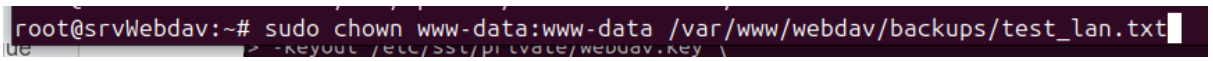


Création du fichier téléchargeable

Pour valider que la sauvegarde est opérationnelle, nous avons généré un fichier de test :



En lui donne des droits au fichier



B. Sécurisation avec un certificat

Nous avons également généré un certificat SSL (`webdav.key` et `webdav.crt`) pour chiffrer les communications.

```
root@srvWebdav:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
> -keyout /etc/ssl/private/webdav.key \
> -out /etc/ssl/certs/webdav.crt
.......+++++++++++++++++++++++++++++++++++++++*.+.......................+......+
...+++++++++++++++++++++++++++++++++++++++*....+.........+..........+...+.+...
..+...........+...+.+.....+..............++++++
....+.....+.+......+........+.....+.......+..+......+.+.............+....+...+
........+....+..+.+..+.+..+....+...+.....................+..+.........+...+++++
++++++++++++++++++++++++++++++++*...+...+..+.+..+.......+.....+......+.+...+..
.+.....+........+...+...........+.+..+...+........+......+...+....+..+......+
.......+..+.+...........+.....++++++++++++++++++++++++++++++++++++*....+....
..+....+..+.......+..+....+........+...+..+.......+......++++++
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
```

c. Sécurisation des accès (Apache et Fail2ban)

La configuration du `srvWebdav` impose que la connexion provienne obligatoirement du réseau LAN (`Require ip 192.168.11.0/24`) et nécessite un utilisateur valide (`Require valid-user`). dans le fichier webdav-ssl.conf

```
root@srvWebdav:~# nano /etc/apache2/sites-available/webdav-ssl.conf
```

```
  GNU nano 8.4                  /etc/apache2/sites-available/webdav-ssl.conf *
<VirtualHost 192.168.12.95:443>
    ServerName webdav.menuimetal.fr

    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/webdav.crt
    SSLCertificateKeyFile /etc/ssl/private/webdav.key

    Alias /sauvegardes /var/www/webdav/backups

    <Directory /var/www/webdav/backups>
        Dav On
        AuthType Basic
        AuthName "Acces Sauvegardes Menuimetal"
        AuthUserFile /etc/apache2/.htpasswd

        Require ip 192.168.11.0/24
        Require valid-user
    </Directory>
</VirtualHost>
```

Activaction de la page web webdav-ssl

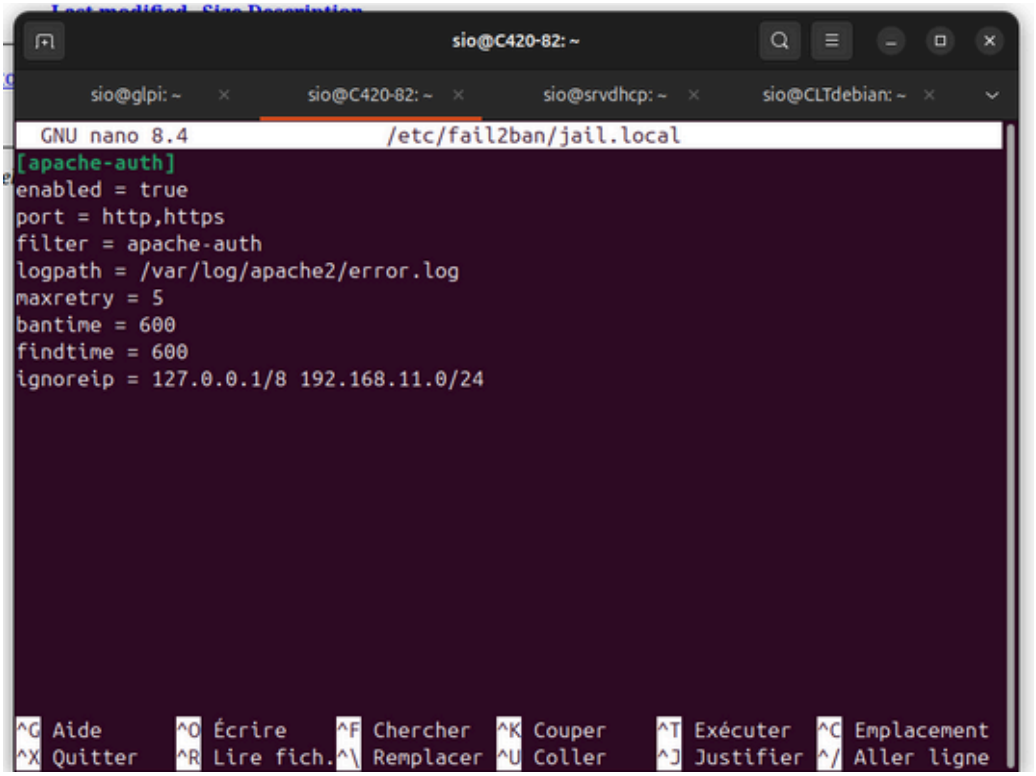
```
root@srvWebdav:~# a2ensite webdav-ssl
systemctl reload apache2
Site webdav-ssl already enabled
root@srvWebdav:~# systemctl reload apache2
```

INSTALLATION de FAIL2BAN

Avec un apt install fail2ban

```
root@srvWebdav:~# apt install fail2ban -y
```

Pour contrer les attaques de type force brute, Fail2ban a été installé. Une prison (`jail`) a été configurée pour bannir les tentatives d'authentification échouées, tout en ignorant spécifiquement notre réseau LAN (192.168.11.0/24) via la directive `ignoreip`.



4. Supervision, Logs et Inventaire

- **Remontée des Logs : Le service Rsyslog a été configuré pour envoyer les journaux du serveur WebDAV vers notre serveur de logs centralisé.**

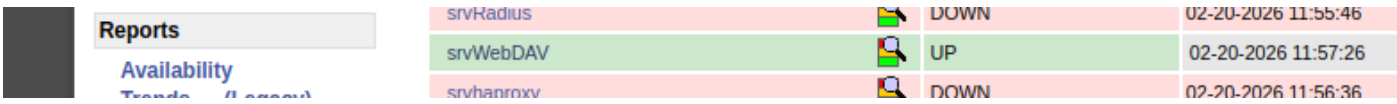
```
dpkg.log.1              srvhaproxy2/
root@SrvRsyslog:~# ls /var/log/
192.168.13.29/          dpkg.log.2.gz          SrvMail/
192.168.14.70/          dpkg.log.3.gz          srvNAGIOS/
alternatives.log        dpkg.log.4.gz          srvPostfix/
alternatives.log.1      faillog                SrvRsyslog/
alternatives.log.2.gz   installer/             srvVpn/
alternatives.log.3.gz   journal/               srvWebdav/
```

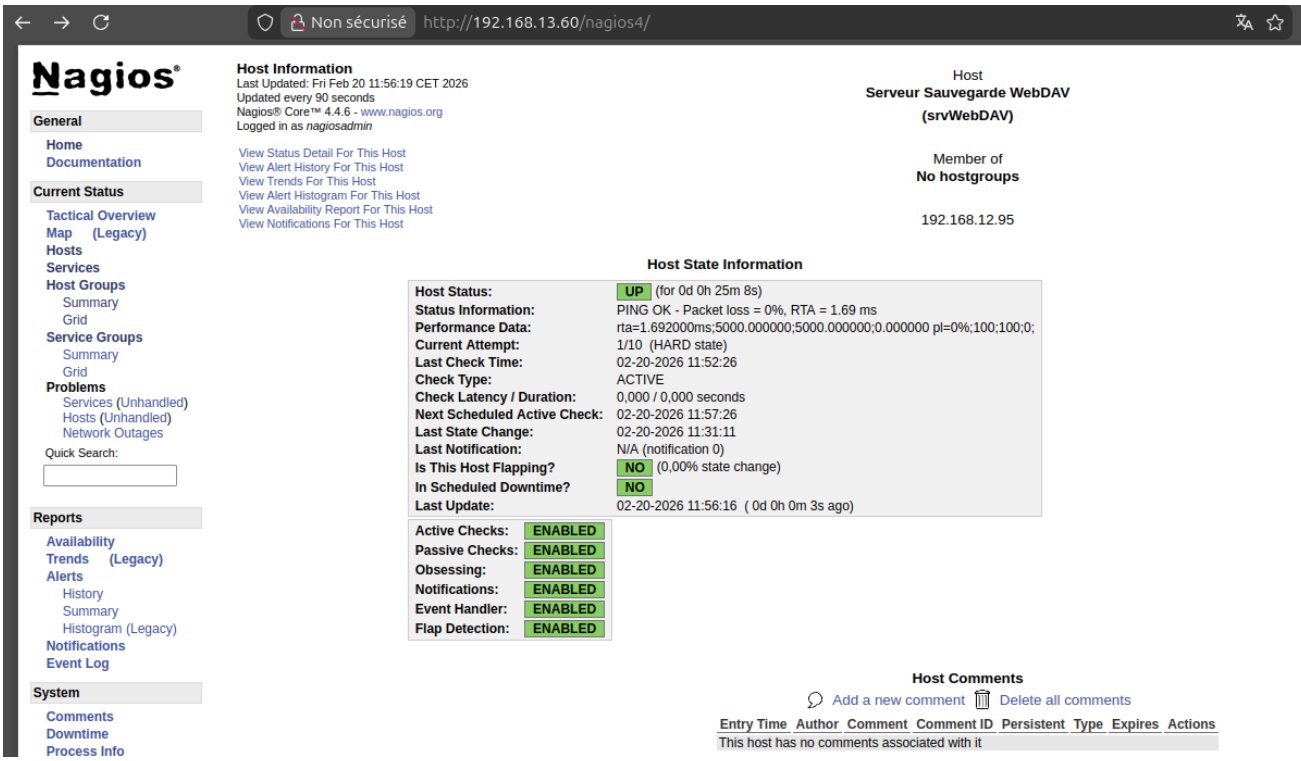
```
qemu-ga.log   root.log      rsyslogd.log  sudo.log      systemd.log
root@SrvRsyslog:~# tail -f /var/log/srvWebdav/systemd.log
analyse_ssh.sh    .lesshst          .profile
.bash_history     .local/           .ssh/
.bashrc           .mariadb_history  .viminfo
root@SrvRsyslog:~# tail -f /var/log/srvWebdav/systemd.log
2026-02-20T11:14:27+01:00 srvWebdav systemd[1]: Stopping rsyslog.service - System Logging Service...
2026-02-20T11:14:27+01:00 srvWebdav systemd[1]: rsyslog.service: Deactivated successfully.
2026-02-20T11:14:27+01:00 srvWebdav systemd[1]: Stopped rsyslog.service - System Logging Service.
2026-02-20T11:14:27+01:00 srvWebdav systemd[1]: Starting rsyslog.service - System Logging Service...
2026-02-20T11:14:27+01:00 srvWebdav systemd[1]: Started rsyslog.service - System Logging Service.
```

répertoire `/sauvegardes` avec les identifiants de test.

```
  GNU nano 8.4                          /etc/nagios4/servers/WebDAV.cfg *
# Définition du serveur WebDAV
define host {
    use                     linux-server
    host_name               srvWebDAV
    alias                   Serveur Sauvegarde WebDAV
    address                 192.168.12.95
}

# Surveillance du service HTTP WebDAV
define service {
    use                     generic-service
    host_name               srvWebDAV
    service_description     Check WebDAV Backups
    # -u : chemin configuré dans ton Apache (/sauvegardes)
    # -a : identifiants créés avec htpasswd (utilisateur:motdepasse)
    check_command           check_http!-u /sauvegardes -a admin_backup:sio
}
```

- **INTERFACE**





- **Inventaire et DNS : La VM remonte correctement dans l'inventaire GLPI via son agent, et la résolution de nom `Webdav.menuimetal.fr` est opérationnelle.**

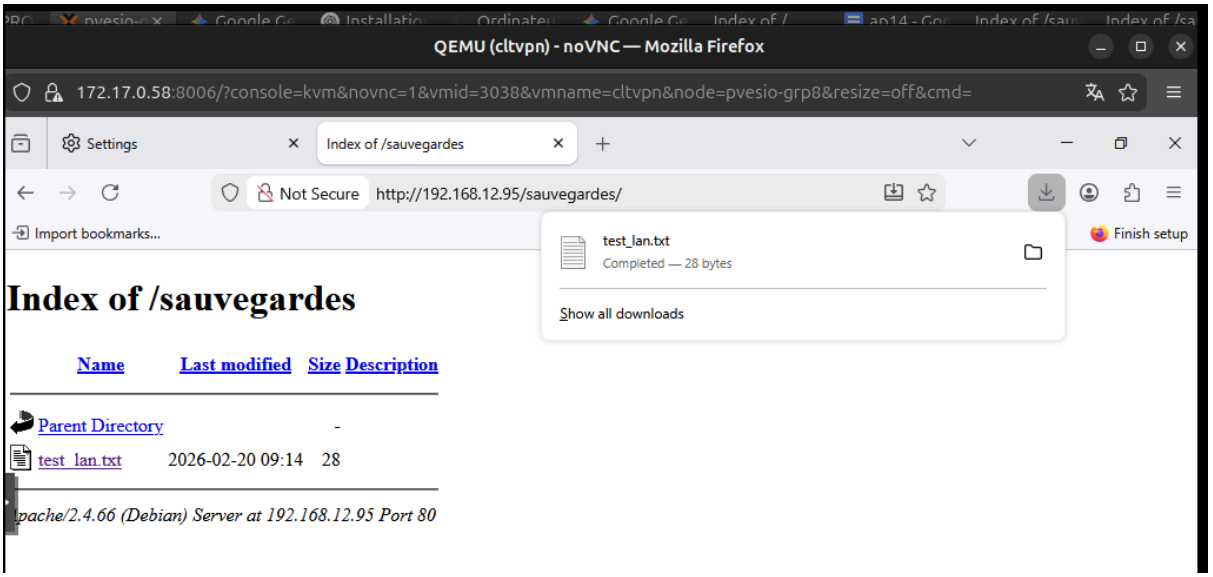
DNS:

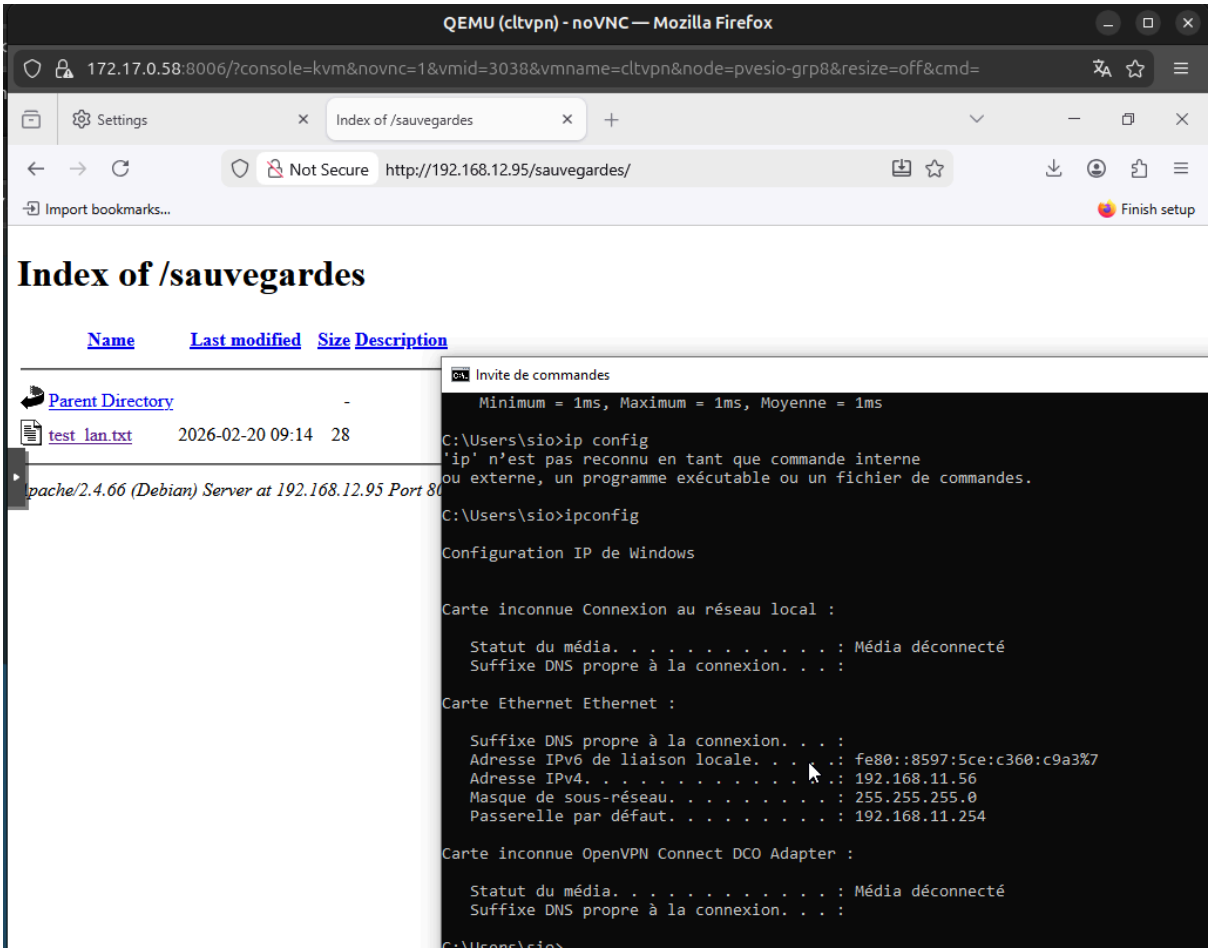
GLPI

5. Test de fonctionnement

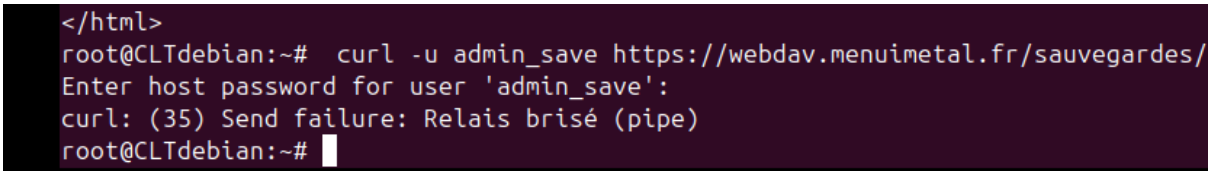
Les tests ont été concluants. L'accès depuis un navigateur web situé sur le réseau autorisé affiche bien l'index du répertoire de sauvegardes après authentification.

test depuis un client Windows en VLAN LAN dans une interface





Depuis une vm cliente dans la Vlan Gestion



</html>
root@CLTdebian:~# curl -u admin_save https://webdav.menuimetal.fr/sauvegardes/
Enter host password for user 'admin_save':
curl: (35) Send failure: Relais brisé (pipe)
root@CLTdebian:~#