Nom : Rhassef
Prénom : Ayoub

COMPTE RENDU D'ACTIVITÉ

AP4 – PARCINFO : Déploiement d'une ferme de serveurs GLPI redondée par un répartiteur de charge HAProxy

04/03/2026-23/03/2026

SOMMAIRE

1. Résumé du contexte

L'entreprise souhaitait mettre en place un outil de gestion de parc informatique et d'Helpdesk (GLPI). Initialement déployé sur un serveur unique, le projet a évolué pour répondre à des besoins de sécurité, de segmentation réseau (DMZ) et de haute disponibilité pour garantir un service 24h/24 aux utilisateurs.

2. Objectifs et Missions à effectuer

Le projet a été découpé en cinq phases majeures :

M0 – Infrastructure & Réseau : Configuration du routeur VyOS pour segmenter le réseau en 3 zones (Bureau, DMZ, SRV) et définition du plan d'adressage IP.

M1 – Déploiement Applicatif : Installation de la pile LAMP sur Debian 12 et mise en service de GLPI 10. Premier inventaire du parc informatique.

M2 – Gestion & LDAP : Interconnexion avec l'annuaire Active Directory pour centraliser l'authentification et configuration des profils de support (Helpdesk).

M3 – Automatisation : Déploiement massif de l'agent GLPI par GPO sur le domaine Windows pour automatiser la remontée d'inventaire des postes clients.

M4 – Sécurisation : Mise en place du protocole HTTPS (SSL) pour chiffrer les échanges et création d'un script de sauvegarde automatisée de la base de données.

M5 – Haute Disponibilité : Passage en architecture 3-tiers. Externalisation de la BDD (.13), création d'un clone Web (.11) et installation d'un Load Balancer HAProxy (.12) avec Sticky Sessions.

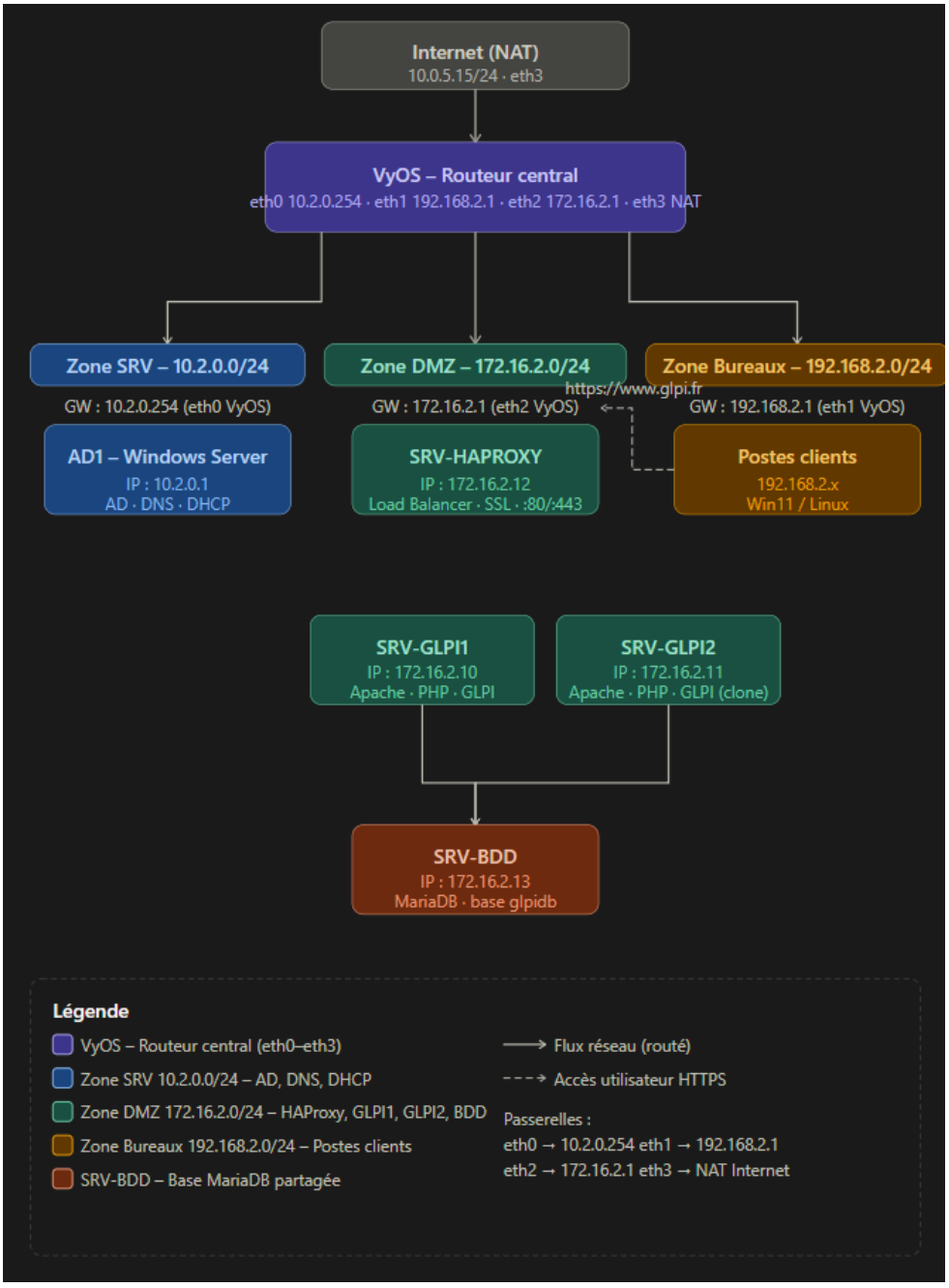
3. Présentation des Missions

3.1 M0.1 – Schéma Réseau et Plan d'adressage

Conception de l'architecture cible et définition des plages IP.
Étapes :

Analyse du cahier des charges.

Définition des 3 zones : LAN-SRV (10.2.0.0), LAN-BUREAUX (192.168.2.0), DMZ (172.16.2.0).



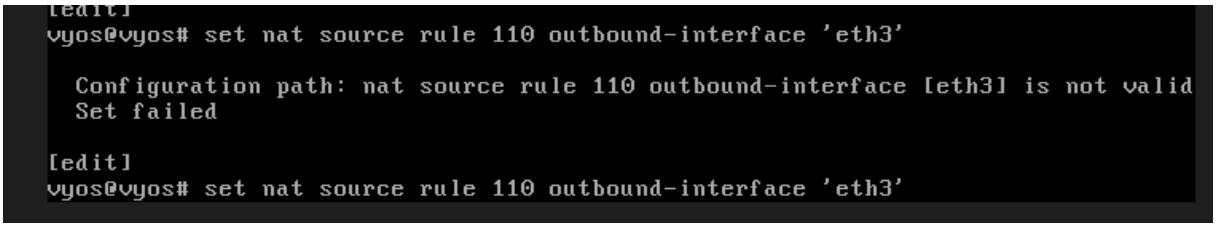
☑ Bilan de la mission — Objectif atteint. L'architecture réseau est définie avec les trois zones (DMZ, SRV, BUREAUX) et les plages IP documentées. Le schéma réseau a été produit et validé.

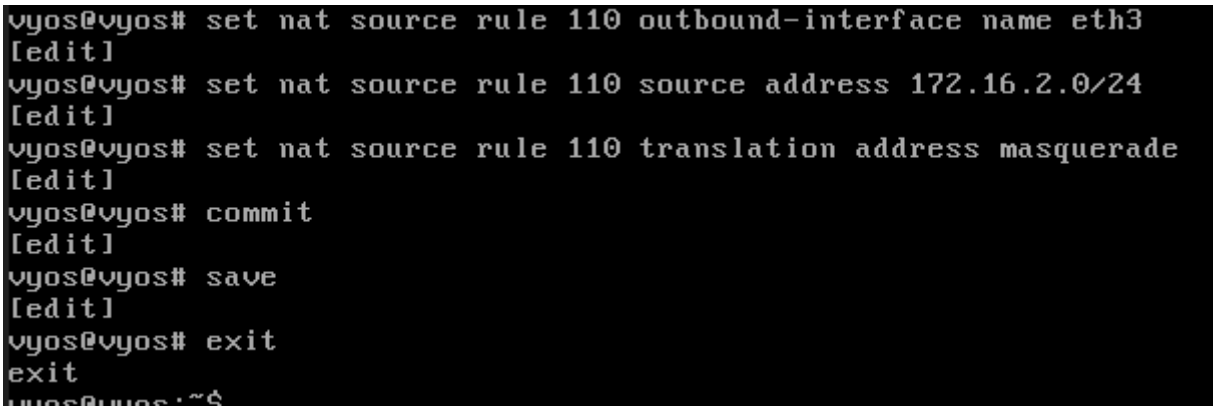
3.2 M0.2 – Configuration du Routeur VyOS (Passerelle)

Mise en place du routage inter-VLAN et de l'accès internet.

Étapes :

1. Configuration des interfaces eth0, eth1, eth2.
2. Activation du client DHCP sur eth3 (WAN).
3. Configuration du NAT Source (Masquerade) pour les réseaux internes.
4. Ajout de la route statique vers 10.0.5.2.
   Problèmes rencontrés : La syntaxe outbound-interface name eth3 est spécifique à cette version de VyOS. Une première syntaxe incorrecte a été utilisée ; la commande d'autocomplétion par tabulation a permis d'identifier la syntaxe valide.





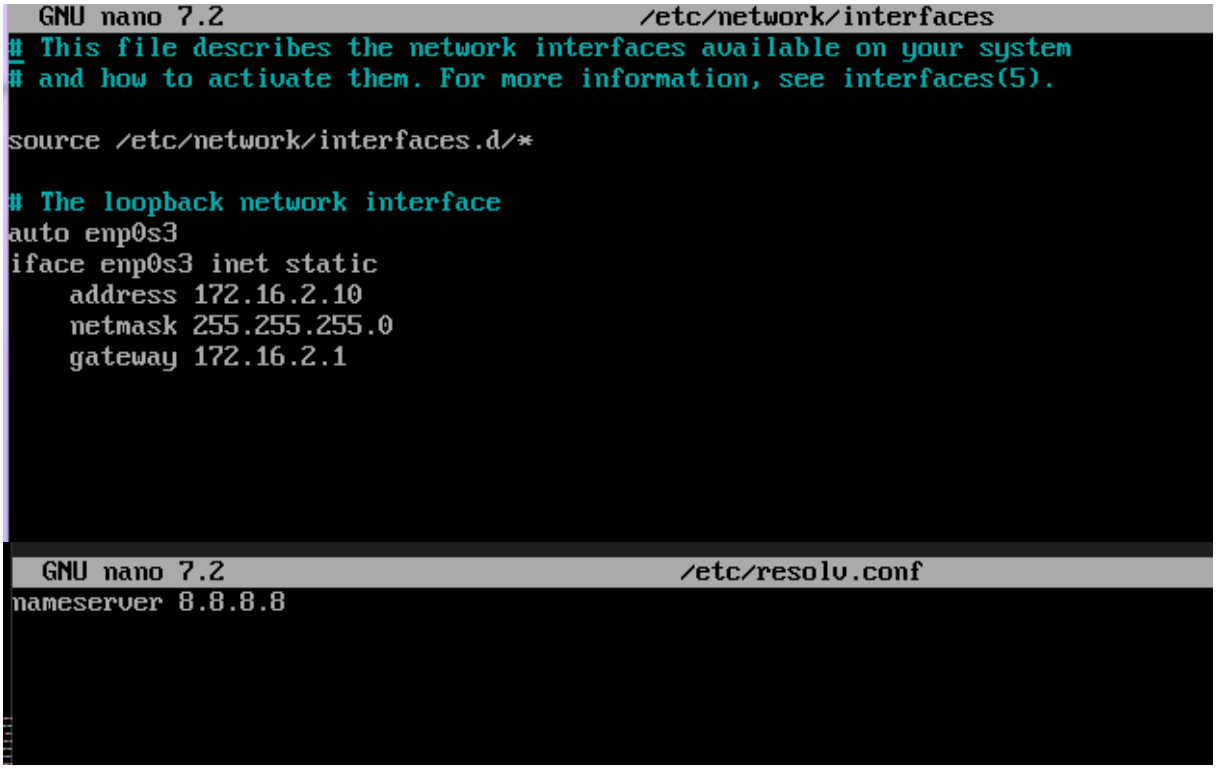
☑ **Bilan de la mission —** Objectif atteint. Le routeur VyOS assure le routage inter-zones et la sortie Internet via NAT. Les trois interfaces (eth0, eth1, eth2) sont opérationnelles et les machines des différentes zones communiquent entre elles.

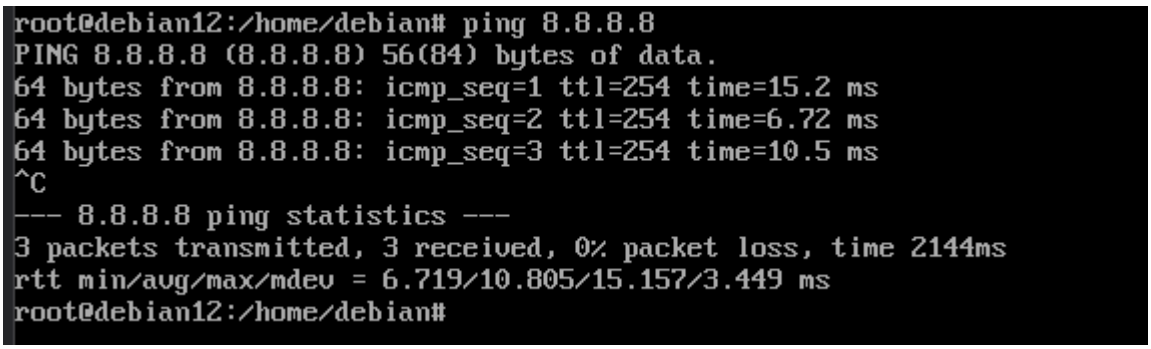
3.3 M0.3 – Préparation du serveur Debian 12

Installation de l'OS et configuration réseau de base.

Étapes :

1. Installation de Debian 12 (Netinstall).
2. Configuration de l'IP statique dans /etc/network/interfaces.
3. Configuration du DNS dans /etc/resolv.conf.
4. Test de connectivité (Ping 8.8.8.8).





3.4 M1.1 – Installation de la pile LAMP

Installation des services Web et de la base de données.
Étapes :

1. Installation d'Apache2, MariaDB-server.
2. Installation de PHP 8.2 et des extensions requises (mbstring, gd, xml, intl, etc.).
3. Vérification du statut des services (systemctl status).

☑ **Bilan de la mission —** Objectif atteint. Le serveur Debian est installé avec une IP statique, une résolution DNS fonctionnelle et une connectivité vérifiée vers Internet et les autres zones.

3.5 M1.2 – Configuration de la Base de Données et Déploiement de la solution GLPI 10

Préparer l'environnement de stockage (MariaDB) et installer les fichiers de l'application GLPI.

Étape 1 : Création de la Base de Données (SQL)

Se connecter à MariaDB : sudo mariadb -u root

- **Créer la DB :** CREATE DATABASE glpi;
- **Créer l'utilisateur :** CREATE USER 'adminglpi'@'localhost' IDENTIFIED BY 'TonMotDePasse';
- **Attribuer les droits :** GRANT ALL PRIVILEGES ON glpi.* TO 'adminglpi'@'localhost';
- **Valider :** FLUSH PRIVILEGES; EXIT;

```
root@debian12:/home/debian# sudo mariadb -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.14-MariaDB-0+deb12u2 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> creat database glpi;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
MariaDB server version for the right syntax to use near 'creat database glpi' at line 1
MariaDB [(none)]>
MariaDB [(none)]> create database glpi;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> create user 'adminglpi'@'localhost' identified by 'Sio123*';
Query OK, 0 rows affected (0.012 sec)

MariaDB [(none)]> grant all privileges on glpi 'adminglpi'@'localhost';
.ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your
 MariaDB server version for the right syntax to use near ''adminglpi'@'localhost'' at line 1
MariaDB [(none)]> grant all privileges on glpi.* to  'adminglpi'@'localhost';
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> exit;
Bye
root@debian12:/home/debian#
```

Étape 2 : Téléchargement et Installation des Sources

Se placer dans le dossier temporaire : cd /tmp

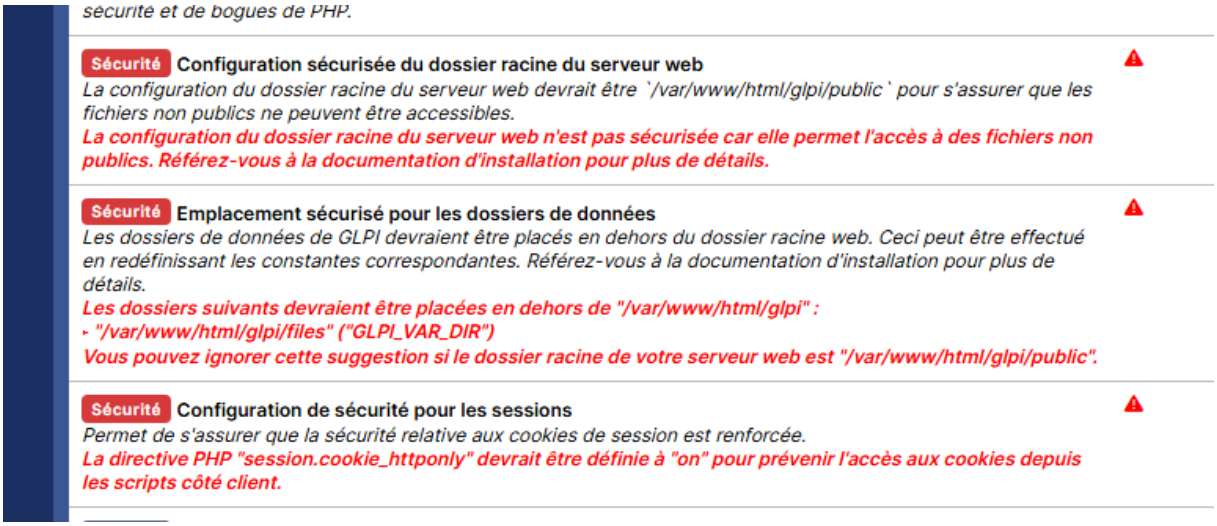
- **Télécharger GLPI :** wget https://github.com/glpi-project/glpi/releases/download/10.0.12/glpi-10.0.12.tgz
- **Extraire vers le répertoire Web :** sudo tar -xvzf glpi-10.0.12.tgz -C /var/www/html/

Étape 3 : Configuration des Permissions (Sécurité)

Il est impératif qu'Apache soit propriétaire des fichiers pour l'installation via navigateur.

- **Changer le propriétaire :** sudo chown -R www-data:www-data /var/www/html/glpi
- **Ajuster les droits :** sudo chmod -R 755 /var/www/html/glpi

```
root@debian12:/tmp# sudo chown -R www-data:www-data /var/www/html/glpi
root@debian12:/tmp# sudo chmod -R 755 /var/www/html/glpi
root@debian12:/tmp# ls
glpi-10.0.12.tgz
systemd-private-fc394b128df740839d5fd2e7fa0af209-apache2.service-uuOSDH
systemd-private-fc394b128df740839d5fd2e7fa0af209-systemd-logind.service-XA7M2K
systemd-private-fc394b128df740839d5fd2e7fa0af209-systemd-timesyncd.service-8AgmiK
root@debian12:/tmp# _
```



Sécurisation du dossier racine (DocumentRoot)

GLPI 10 exige désormais que le serveur web pointe sur le dossier /public.

1. Modification du fichier de configuration d'Apache :

   nano /etc/apache2/sites-available/000-default.conf

2. Changement de la ligne DocumentRoot et ajout du bloc Directory :

   DocumentRoot /var/www/html/glpi

   <Directory /var/www/html/glpi/public>

   AllowOverride All

   Require all granted

   </Directory>

3. Activation du module de réécriture et redémarrage d'Apache :

   a2enmod rewrite

   systemctl restart apache2

2. Sécurisation des sessions PHP

L'alerte indique que les cookies de session pourraient être interceptés par des scripts.

1. Édite le fichier php.ini (vérifie ta version de PHP, ici 8.2 par défaut sur Debian 12) :

   nano /etc/php/8.2/apache2/php.ini
2. Utilise Ctrl+W pour chercher session.cookie_httponly.
3. Change la valeur de Off à On :

   session.cookie_httponly = on
4. Sauvegarde et redémarre Apache :

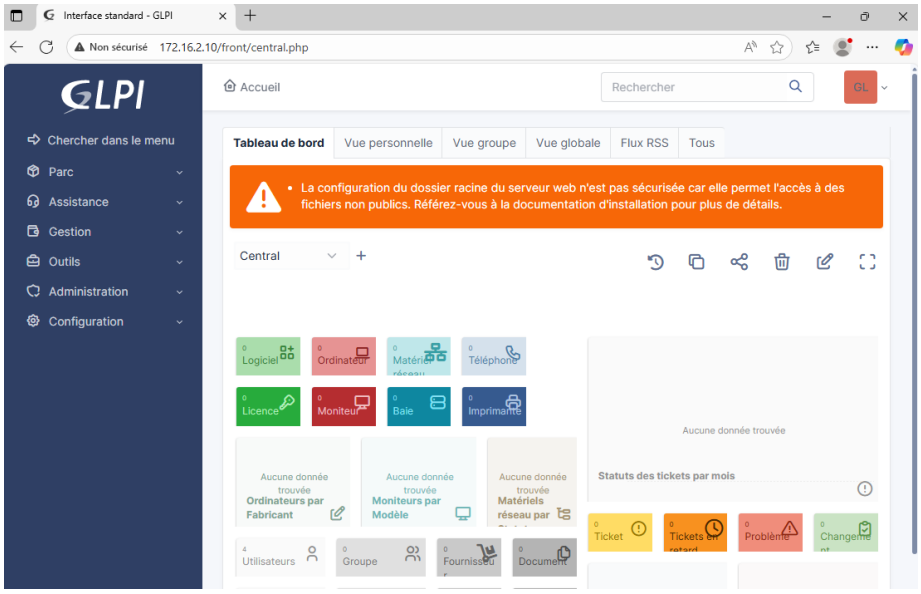
   systemctl restart apache2

3. Finalisation de l'installation via l'interface web (depuis le poste client Windows) :

1. **Bouton "Installer" : On clique sur le bouton jaune**
2. **Licence : On accepte les termes et clique sur "Continuer".**
3. **Vérification des tests : On clique sur "Continuer".**
4. **Connexion à la base de données :**
   - **Serveur SQL : localhost**
   - **Utilisateur SQL : glpi**

- **Mot de passe SQL : Sio123* (ou celui défini lors de la création)**

5. **Choix de la base : Sélectionne glpi dans la liste qui apparait**



Bilan de la mission — Objectifs atteints. -GLPI 10 est installé et accessible depuis les postes clients. La base de données est configurée, les permissions Apache sont correctes et l'interface web répond sur l'IP du serveur.

- La pile LAMP est installée et opérationnelle. Apache, MariaDB et PHP 8.2 fonctionnent, les services démarrent automatiquement au boot.

3.6 M1.3 – Recensement et Inventaire (Export CSV)

Objectif :

Mettre en place un inventaire automatisé et centralisé de l'infrastructure réseau via GLPI.

I. Installation et Configuration des Agents (Debian 12)

Sur la **Debian12cliente** et le **Serveur-GLPI**, les commandes suivantes ont été exécutées :

1. **Installation de l'agent :**

   sudo apt update

   sudo apt install glpi-agent -y

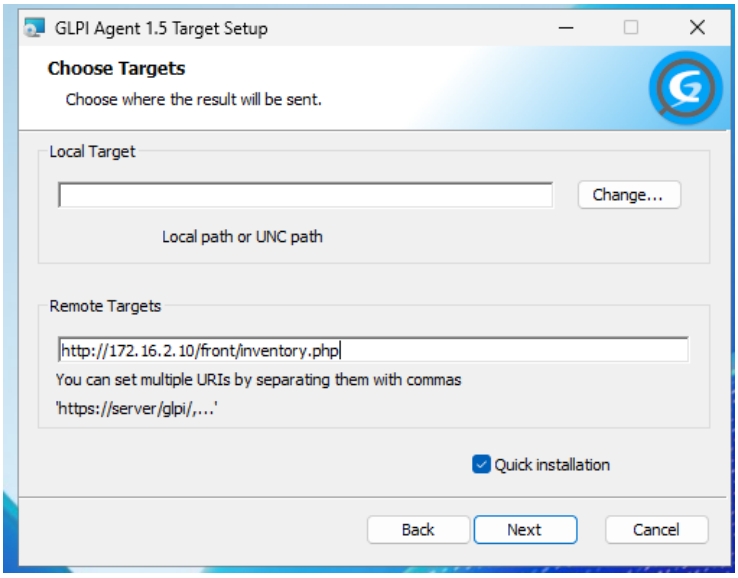
2. **Configuration du serveur cible :** L'agent a été configuré pour pointer vers l'adresse IP du serveur GLPI : server = http://172.16.2.10/front/inventory.php
3. **Forçage de l'inventaire (Debian) :** Pour faire remonter la machine immédiatement dans GLPI :

   sudo glpi-agent --server=http://172.16.2.10/front/inventory.php –force

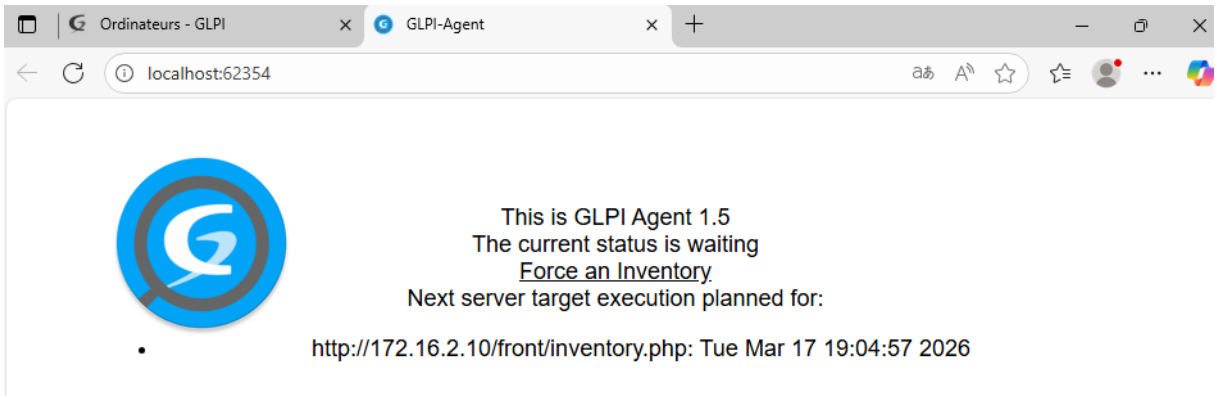
```
root@debian12:~# glpi-agent --version
GLPI Agent (1.5-1)
Built by Debian
Source time: 2023-06-21 13:59
root@debian12:~# udo glpi-agent --server=http://172.16.2.10/front/inventory.php --force
bash: udo: command not found
root@debian12:~# sudo glpi-agent --server=http://172.16.2.10/front/inventory.php --force
[info] target server0: server http://172.16.2.10/front/inventory.php
[info] sending prolog request to server0
[info] server0 answer shows it supports GLPI Agent protocol
[info] running task Inventory
[info] New inventory from debian12-2026-03-17-12-06-10 for server0
[error] usb.ids not found
root@debian12:~#
```

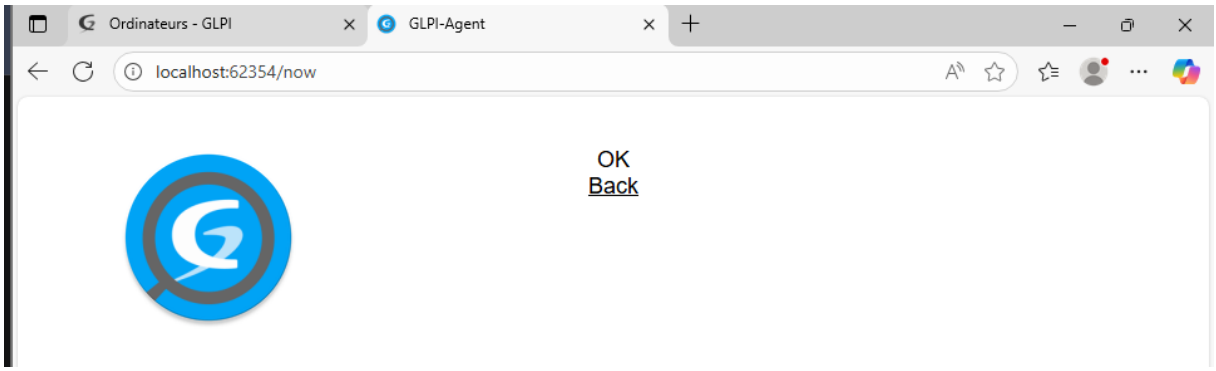
II. Configuration et Forçage (Windows)

Sur la machine hôte Windows, après l'installation du package .msi :

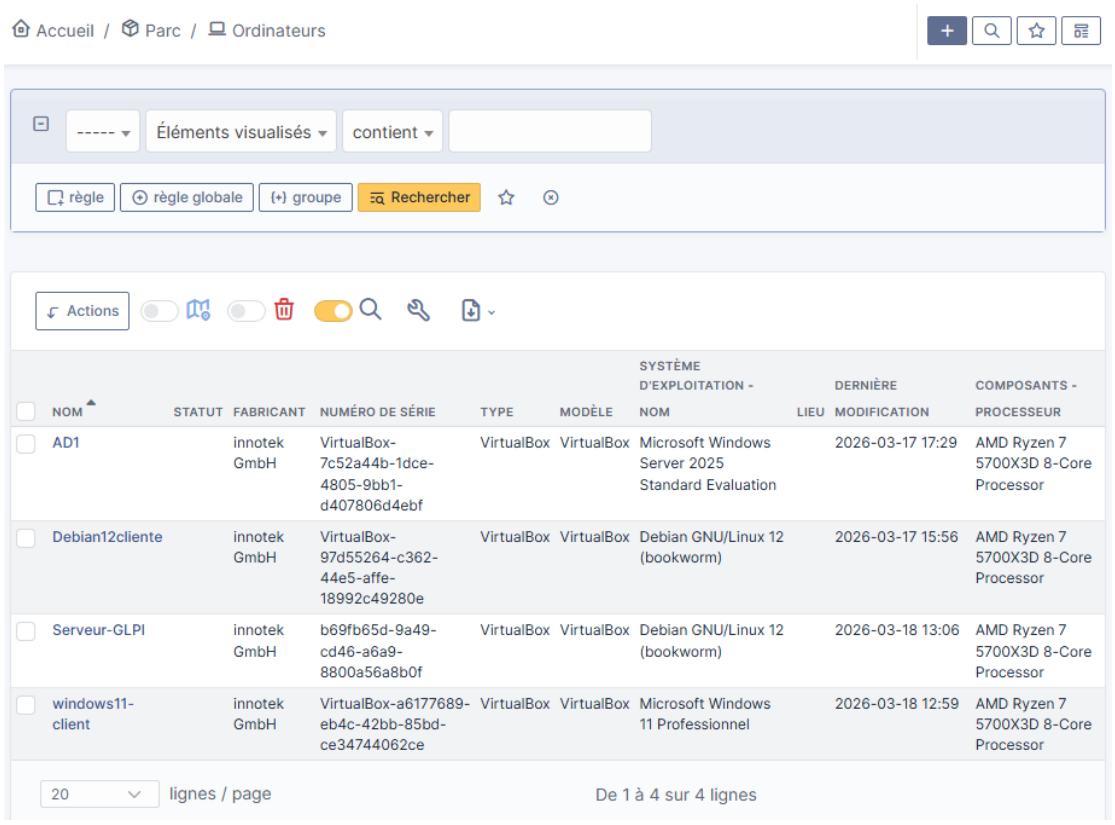


1. **Correction de l'URL cible (Remote Target) :** Utilisation de l'URL validée : http://172.16.2.10/front/inventory.php
2. **Forçage de l'inventaire via l'interface locale :** Ouverture du navigateur à l'adresse : http://localhost:62354 Action : Clic sur le bouton **"Force Inventory"**.

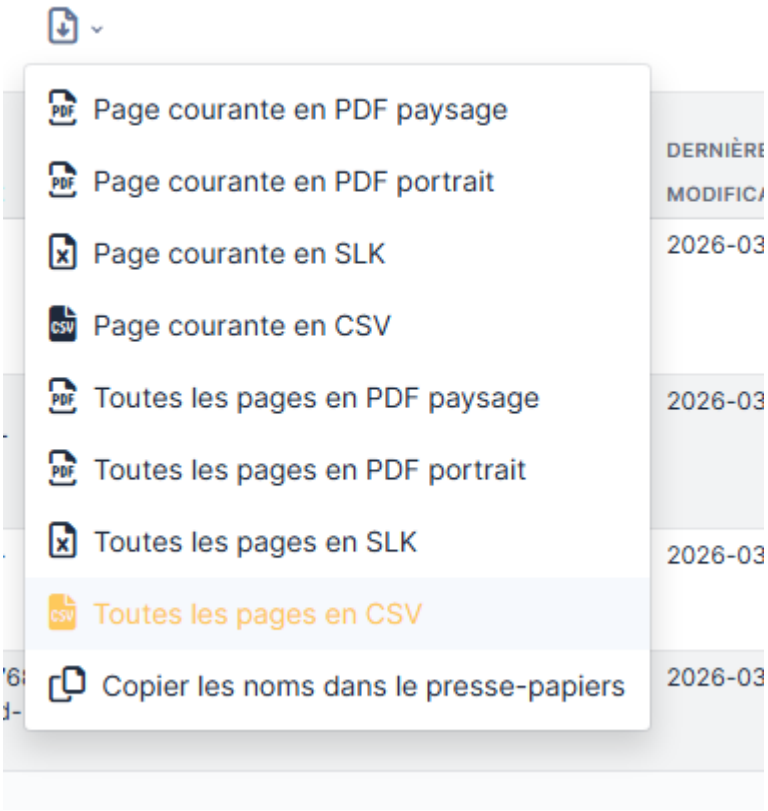




III. Vérification de la remontée dans l'interface GLPI



IV. Export des données au format CSV.



Bilan de la mission — Objectif atteint. Les agents GLPI sont installés sur les machines Debian et Windows. L'inventaire remonte automatiquement dans GLPI et l'export CSV est fonctionnel.

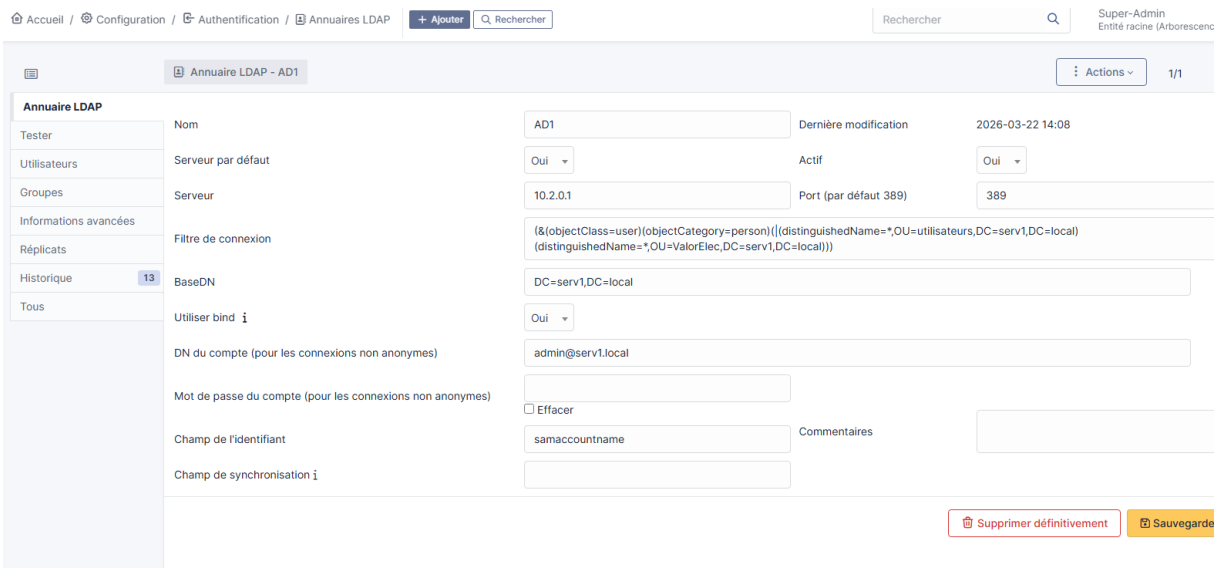
3.7 M2.1 – Synchronisation Active Directory (LDAP)

Importation automatique des utilisateurs du domaine depuis le serveur Windows AD vers GLPI.

Étapes :

1. Configuration du lien LDAP dans GLPI (Serveur 10.2.0.1).
2. Configuration du Filtre dans GLPI
3. Dans **Configuration > Authentification > Annuaires LDAP**.
4. Clique sur ton serveur serv1.local.
5. Dans l'onglet **Annuaire LDAP**, le champ **Filtre de connexion**.
6. Ce filtre LDAP utilise l'opérateur ET (&) pour ne sélectionner que les comptes utilisateurs actifs :

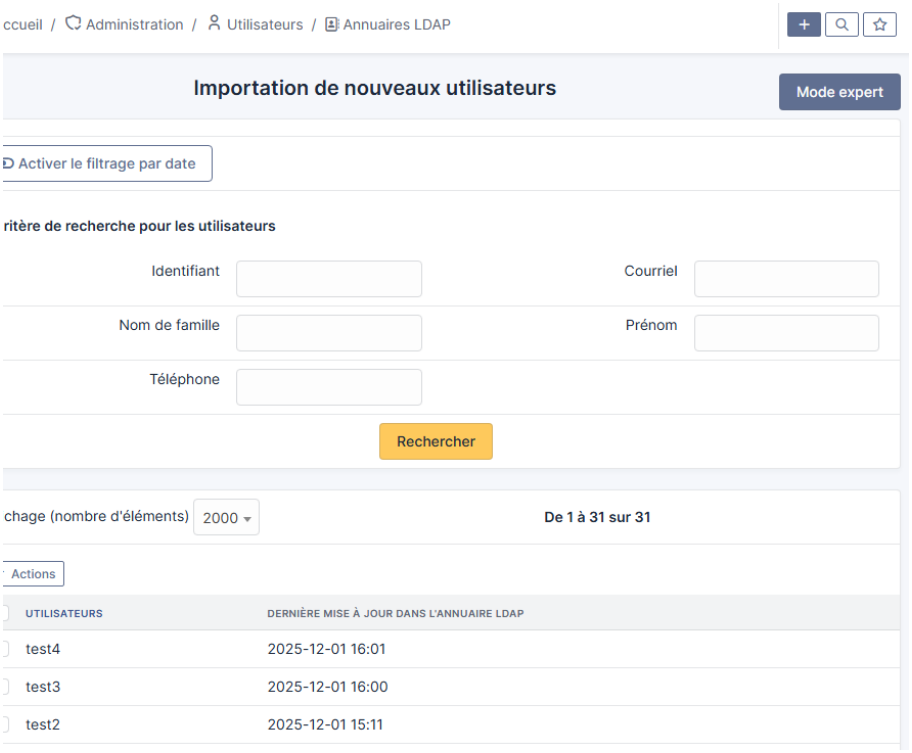
(&(objectClass=user)(objectCategory=person)(!(userAccountControl:1.2.840.113556.1.4.803:=2)))

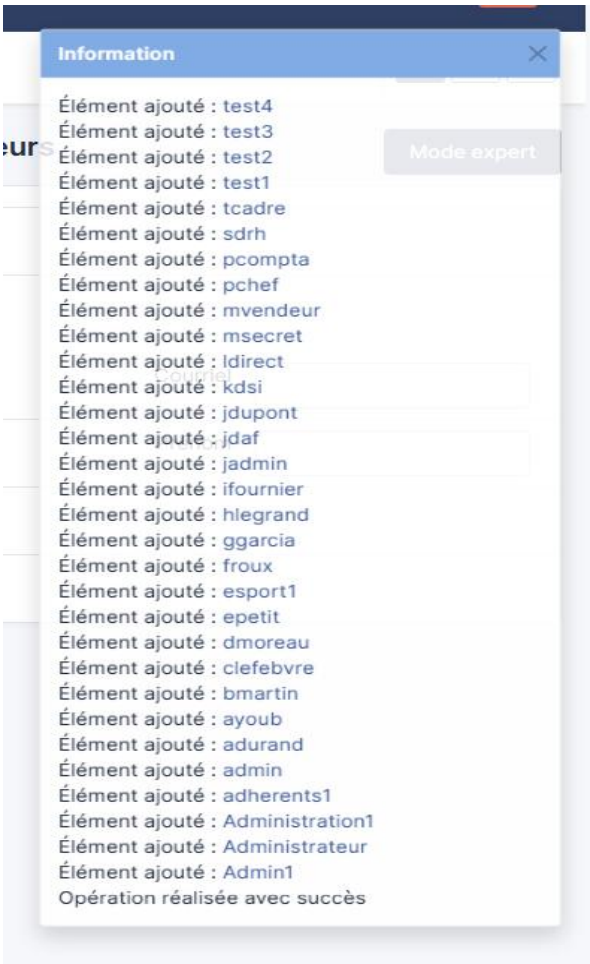


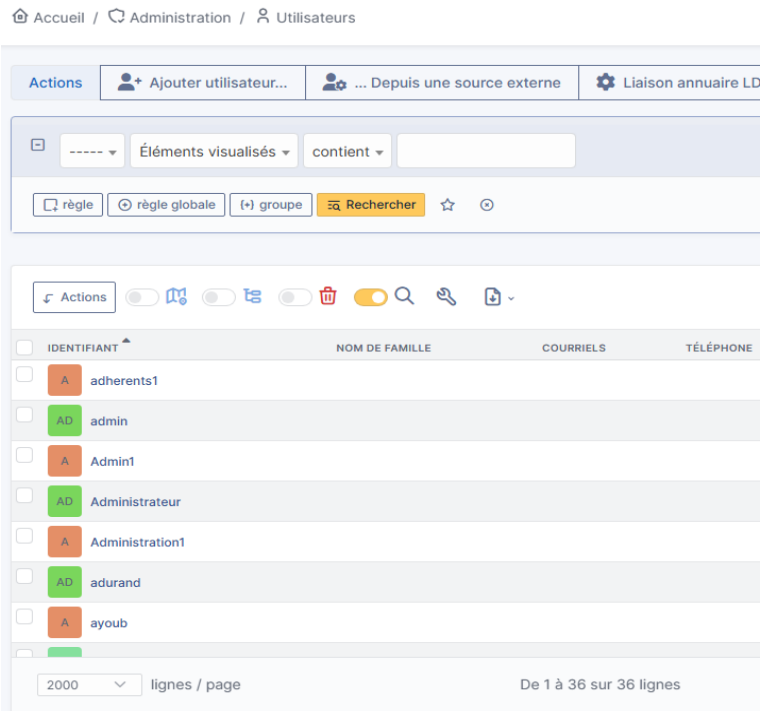
Explication rapide du filtre :

- &(objectClass=user)(objectCategory=person) : On ne veut que des vrais utilisateurs (pas les groupes ou les ordinateurs).
- L'attribut userAccountControl:1.2.840.113556.1.4.803:=2 est un identifiant LDAP Active Directory signifiant : **"Cherche dans cette OU ET dans tous ses enfants (sous-UO)"**.

1. Test de l'importation d'un utilisateur de test.


ccueil / Administration / Utilisateurs / Annuaires LDAP
Importation de nouveaux utilisateurs
Mode expert
Activer le filtrage par date
ritère de recherche pour les utilisateurs
Identifiant
Courriel
Nom de famille
Prénom
Téléphone
Rechercher
chage (nombre d'éléments) 2000
De 1 à 31 sur 31
Actions
UTILISATEURS
DERNIÈRE MISE À JOUR DANS L'ANNUAIRE LDAP
test4 2025-12-01 16:01
test3 2025-12-01 16:00
test2 2025-12-01 15:11


Information
Élément ajouté : test4
Élément ajouté : test3
Élément ajouté : test2
Élément ajouté : test1
Élément ajouté : tcadre
Élément ajouté : sdrh
Élément ajouté : pcompta
Élément ajouté : pchef
Élément ajouté : mvendeur
Élément ajouté : msecret
Élément ajouté : ldirect
Élément ajouté : kdsi
Élément ajouté : jdupont
Élément ajouté : jdaf
Élément ajouté : jadmin
Élément ajouté : ifournier
Élément ajouté : hlegrand
Élément ajouté : ggarcia
Élément ajouté : froux
Élément ajouté : esport1
Élément ajouté : epetit
Élément ajouté : dmoreau
Élément ajouté : clefebvre
Élément ajouté : bmartin
Élément ajouté : ayoub
Élément ajouté : adurand
Élément ajouté : admin
Élément ajouté : adherents1
Élément ajouté : Administration1
Élément ajouté : Administrateur
Élément ajouté : Admin1
Opération réalisée avec succès
Mode expert



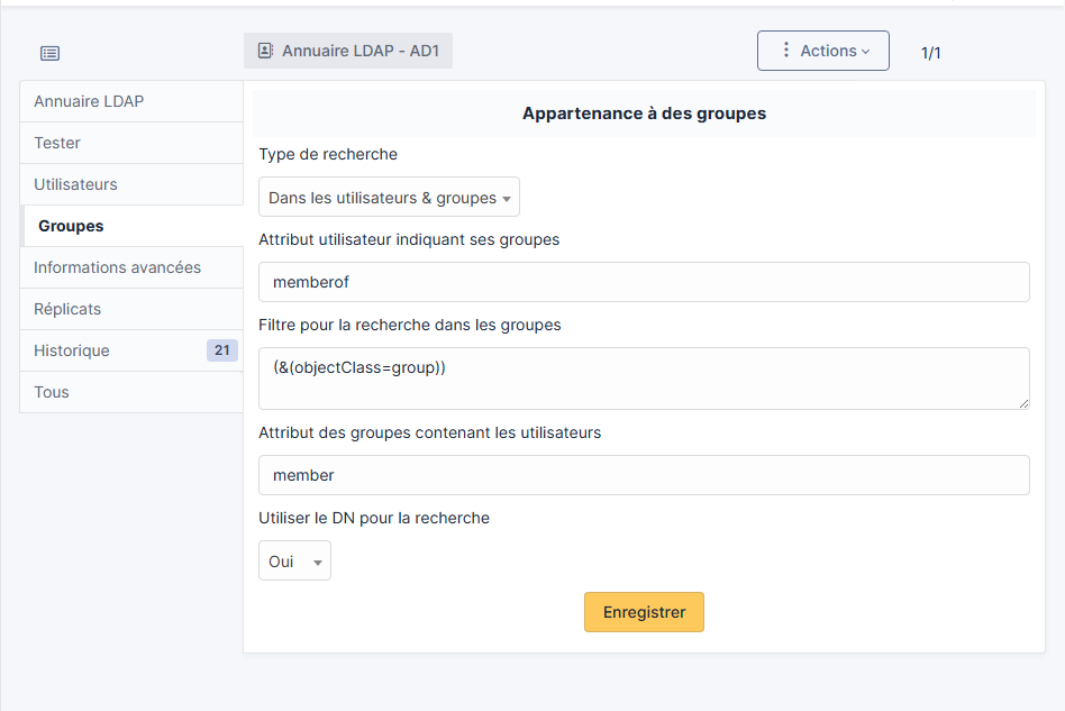
Bilan de la mission — Objectif atteint. La synchronisation LDAP est opérationnelle. Les utilisateurs du domaine Active Directory sont importés automatiquement dans GLPI avec filtrage des comptes désactivés.

3.8 M2.2 – Workflow et Profils Utilisateurs

1. Configuration du Volet "Groupes" dans l'Annuaire LDAP

Pour que GLPI puisse trier les utilisateurs, il doit savoir lire leur appartenance aux groupes AD.

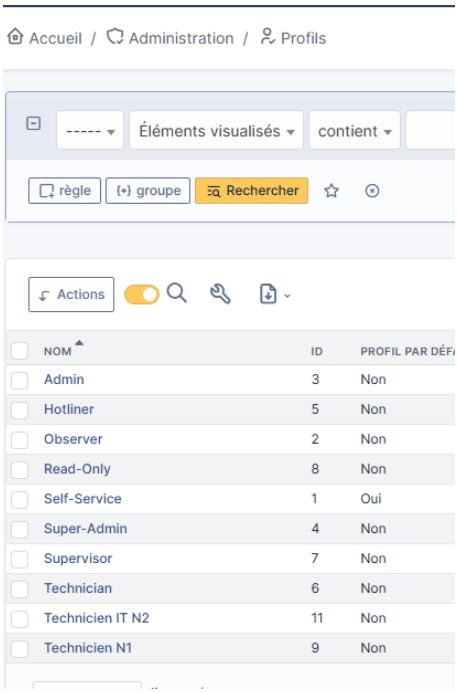
- **Configuration :** Nous avons configuré l'onglet **Groupes** de notre serveur LDAP dans GLPI.
- **Attribut crucial :** Utilisation de l'attribut de recherche memberOf pour récupérer les groupes GLPI_STAGIAIRES, GLPI_IT et GLPI_EMPLOYES.



2. Création et Personnalisation des Profils

Nous avons défini les rôles métiers en créant des profils adaptés aux besoins de ValorElec.

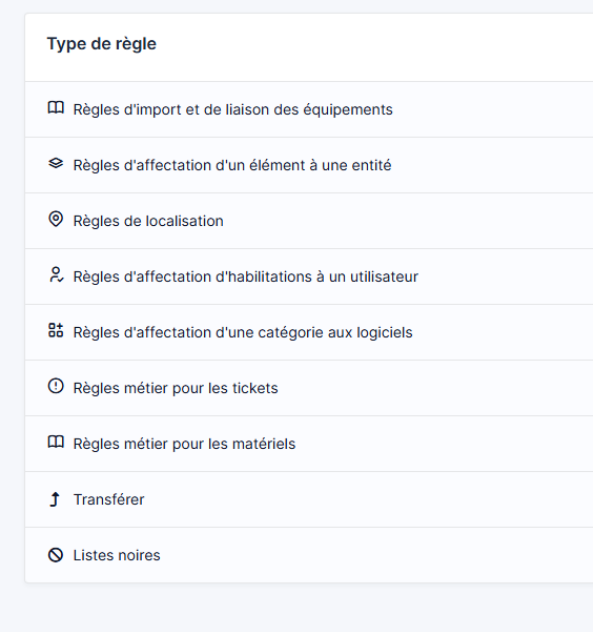
- **Technicien N1** : Interface standard, droits de support de proximité.
- **Technicien IT N2** : Interface standard, droits d'administration complets.
- **Self-Service** : Interface simplifiée pour les employés d'AuditME.



3.Mise en place des Habilitations (Critères et Actions)

C’est ici que nous avons créé l'intelligence du système pour l'assignation automatique.

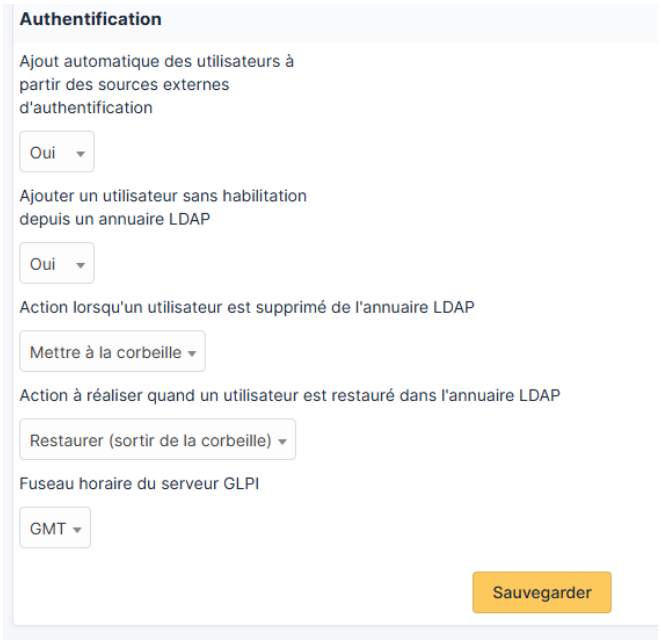
- **Création des règles :** Trois règles ont été créées
- **Configuration :** Pour chaque règle, nous avons défini un **Critère** (ex: Groupe est GLPI_STAGIAIRES) et une **Action** (ex: Assigner Profil Technicien N1 + EntitéTiersLieux)



4.Paramétrage des Actions de Suppression et Restauration

Pour garantir la cohérence entre l'AD et GLPI, nous avons réglé le comportement automatique lors de la modification des comptes.

- **Action :** Configuration dans les paramètres globaux d'authentification.
- **Réglage :** Si supprimé dans l'AD ➡️ **Mettre à la corbeille**. Si restauré ➡️ **Restaurer**.



Automatisation Système sur le serveur Debian

Dernière étape pour rendre le système 100% autonome sans intervention humaine.

- **Action :** Mise en place d'une tâche **Cron** sur la Debian pour forcer la synchronisation toutes les 5 minutes.
- **Commande validée :** php bin/console ldap:synchronize_users --ldap-server-id=3 -n.

```
  GNU nano 7.2                          /tmp/crontab.bp0hJw/crontab
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
*/5 * * * * php /var/www/html/glpi/bin/console ldap:synchronize_users --ldap-server-id=3 -n
```

```
root@Serveur-GLPI:~# php /var/www/html/glpi/bin/console ldap:synchronize_users --ldap-server-id=3
+---------------+---------+
| Serveurs LDAP | AD1 (3) |
| Filtre LDAP   |         |
| Date de début |         |
| Date de fin   |         |
+---------------+---------+
Voulez-vous continuer ? [Yes/no]yes
Serveur LDAP "3" en cours de traitement ...
Importation des utilisateurs du serveur "3" ...
 1/1 [============================] 100%
Synchronisation des utilisateurs avec le serveur "3" ...
 32/32 [============================] 100%
+--------------+---------+-------------+-------------------------+-------------------------------
-+
| Serveur LDAP | Importé | Synchronisé | Supprimé du serveur LDAP | Restauré depuis un annuaire LDAP
 |
+--------------+---------+-------------+-------------------------+-------------------------------
-+
| 3            | 0       | 32          | 0                       | 0
 |
+--------------+---------+-------------+-------------------------+-------------------------------
-+
root@Serveur-GLPI:~#
```

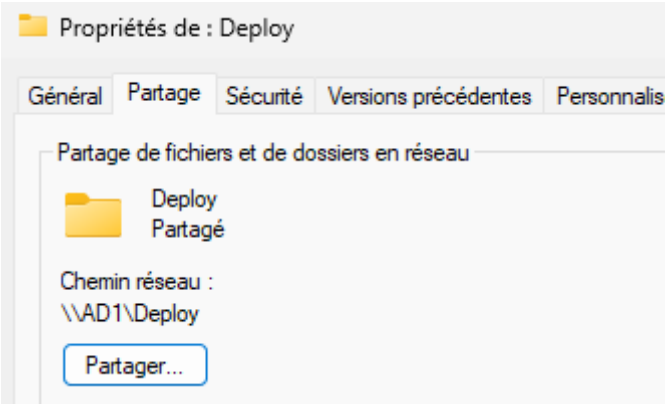
☑ **Bilan de la mission —** Objectif atteint. Les profils Technicien N1, Technicien IT N2 et Self-Service sont créés et associés aux groupes AD (GLPI_STAGIAIRES, GLPI_IT, GLPI_EMPLOYES) via des règles d'habilitation. La synchronisation cron toutes les 5 minutes garantit la cohérence avec l'Active Directory.

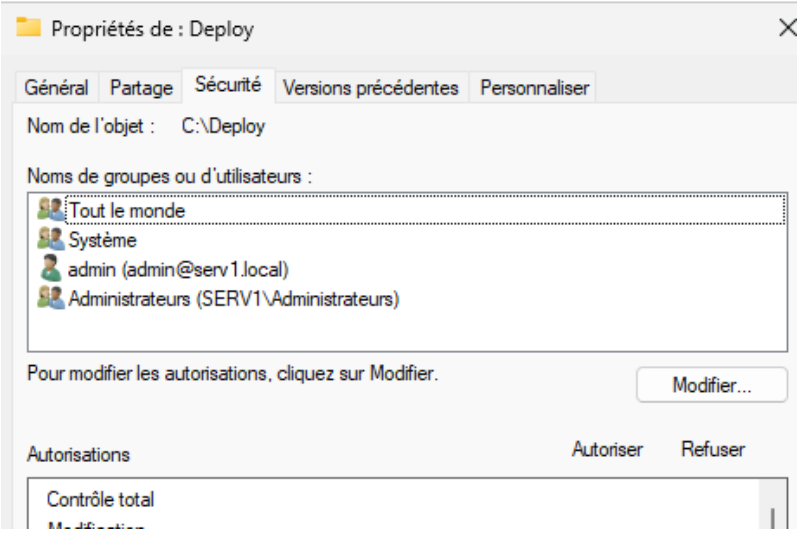
3.9 M3.1 – Déploiement de l'agent par GPO

Automatisation de l'installation sur le parc Windows via le serveur AD.

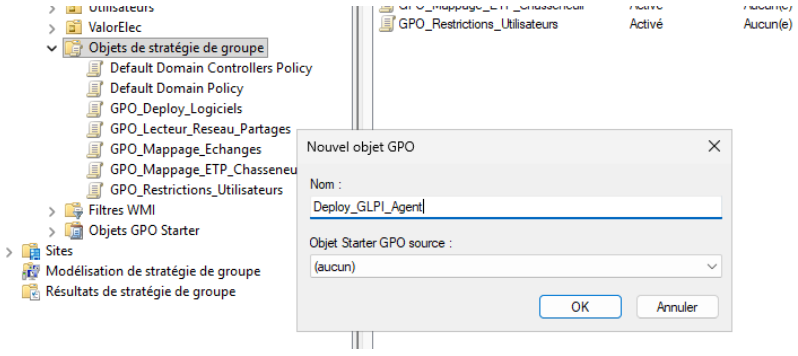
Étapes :

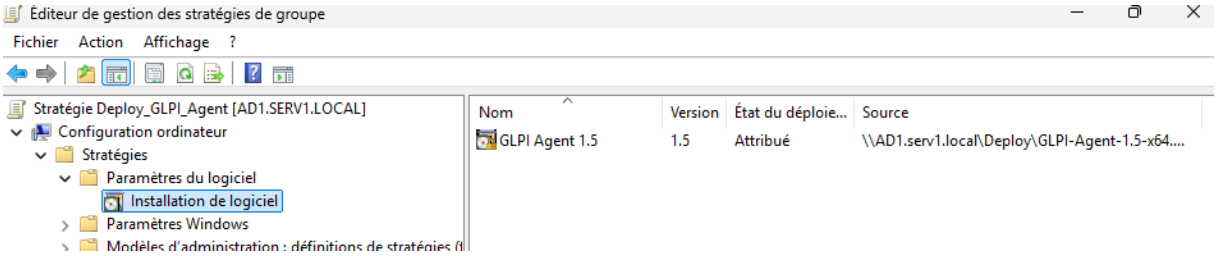
Création d'un partage réseau pour le .msi de l'agent.

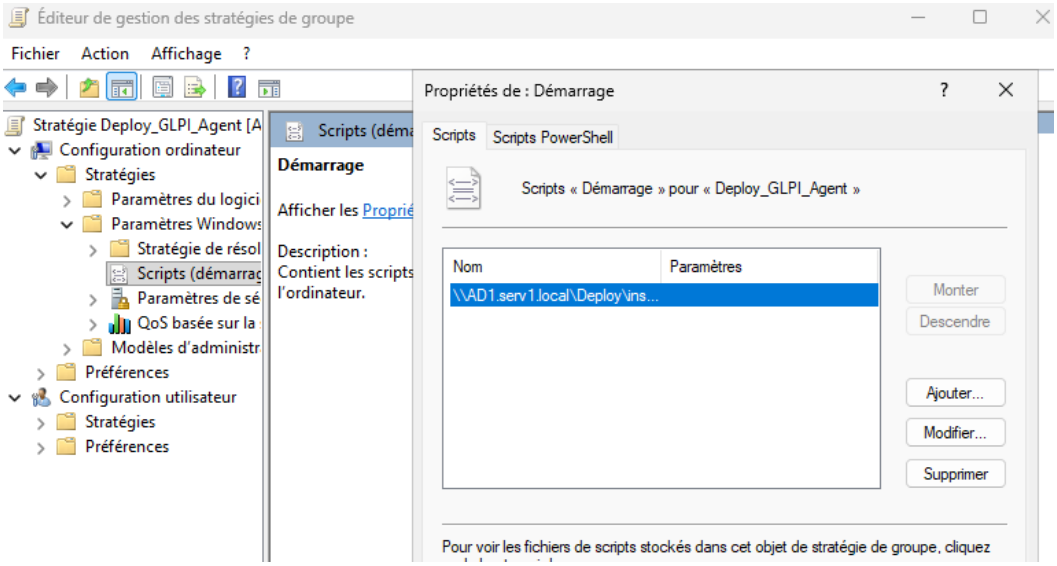




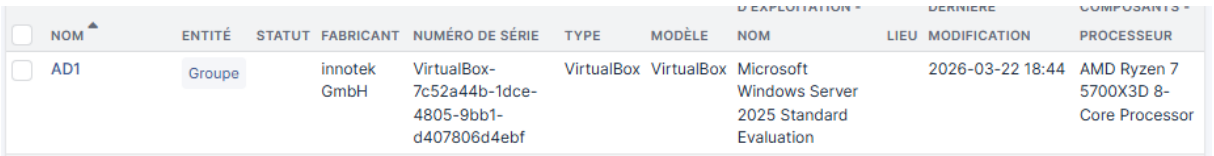
Création de la GPO d'installation de logiciel sur l'AD.







- Création d'un nouveau fichier texte nommé install_agent.bat contenant le script suivant :
- msiexec /i "\\serv1.local\Deploy\GLPI-Agent-1.5-x64.msi" /quiet RUNNOW=1 SERVER='http://192.168.x.x/glpi/front/inventory.php'
- On enregistre et on ajoute ce script dans la liste des scripts de démarrage.



NOM	ENTITÉ	STATUT	FABRICANT	NUMÉRO DE SÉRIE	TYPE	MODÈLE	NOM	LIEU	MODIFICATION	PROCESSEUR
AD1	Groupe		innotek GmbH	VirtualBox-7c52a44b-1dce-4805-9bb1-d407806d4ebf	VirtualBox	VirtualBox	Microsoft Windows Server 2025 Standard Evaluation		2026-03-22 18:44	AMD Ryzen 7 5700X3D 8-Core Processor

☑ **Bilan de la mission —** Objectif atteint. L'agent GLPI est déployé automatiquement sur les postes Windows du domaine via GPO au démarrage de session. L'inventaire remonte sans intervention manuelle.

3.10 M4.1 – Sécurisation SSL (HTTPS)

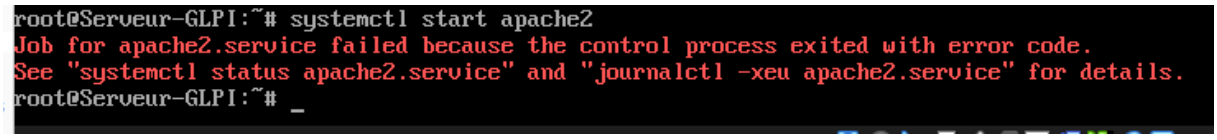
Objectif : Mise en place du protocole HTTPS sur le serveur GLPI (Debian) pour chiffrer les échanges de données et sécuriser l'authentification des utilisateurs.

Actions réalisées

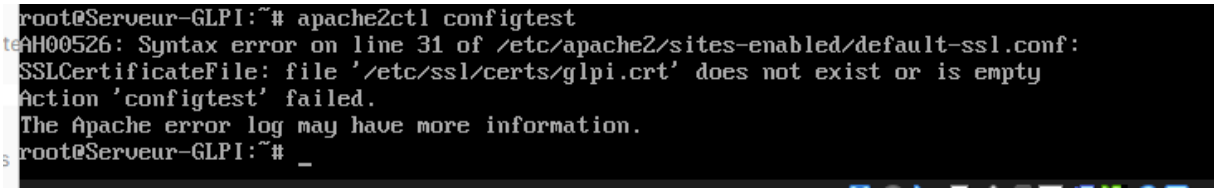
- **Génération du certificat** : Création d'un certificat auto-signé et d'une clé privée avec **OpenSSL** (RSA 2048 bits).
- **Configuration Apache** :
    - # Activation du module SSL d'Apache sudo a2enmod ssl
    - # Génération de la clé privée et du certificat auto-signé (valable 365 jours) sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/glpi.key -out /etc/ssl/certs/glpi.crt
    - Configuration du VirtualHost SSL
- **Forçage du HTTPS** : (Optionnel si tu l'as fait) Mise en place d'une règle de réécriture pour rediriger le flux du port 80 vers le port 443.

Incidents & Résolution (Points clés)

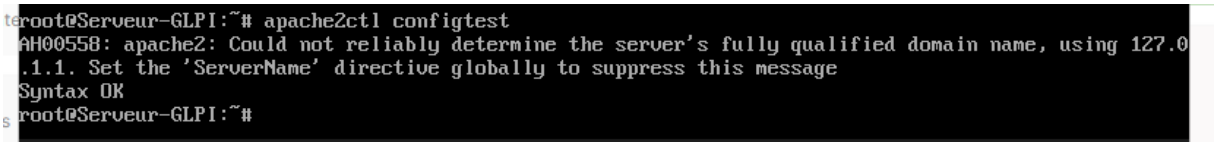
- **Erreur au démarrage d'Apache** : Le service ne se lançait pas suite à l'activation du SSL



- **Diagnostic** : Utilisation de la commande apache2ctl configtest qui a révélé l'absence du fichier de certificat



- **Correction** : Génération correcte des fichiers et validation de la syntaxe



```
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
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:172.16.2.10
Email Address []:
root@Serveur-GLPI:~#
root@Serveur-GLPI:~#
root@Serveur-GLPI:~# apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0
.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
root@Serveur-GLPI:~# systemctl restart apache2
root@Serveur-GLPI:~#
root@Serveur-GLPI:~#
root@Serveur-GLPI:~#
```

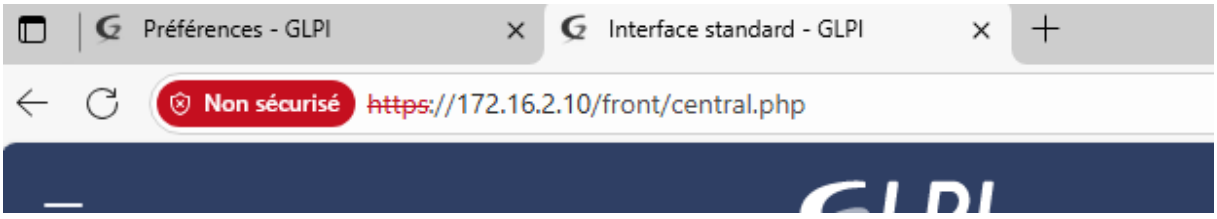
- Activation de session.cookie_secure dans /etc/php/8.2/apache2/php.ini :

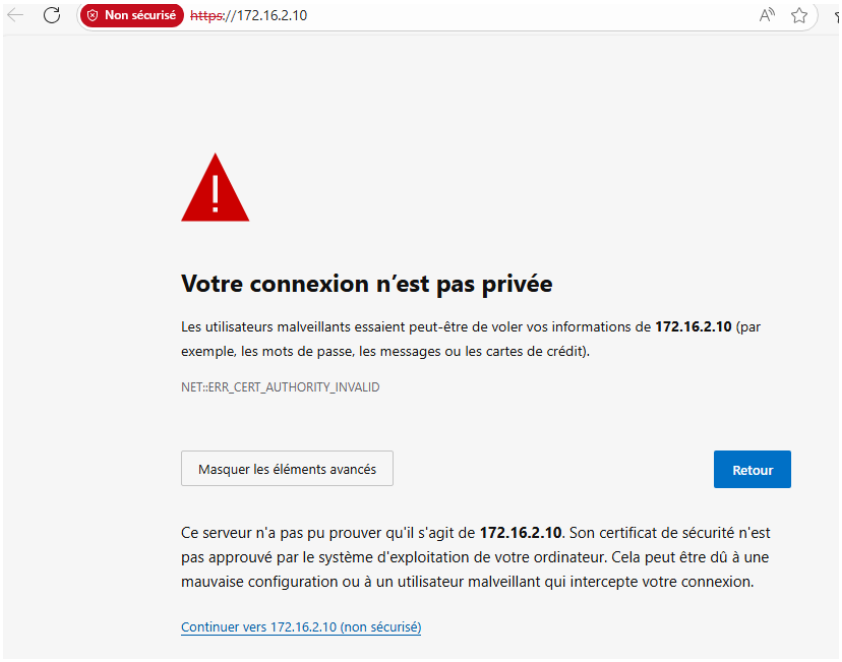
```
 GNU nano 7.2                        /etc/php/8.2/apache2/php.ini *
; https://php.net/session.save-path
;session.save_path = "/var/lib/php/sessions"

; Whether to use strict session mode.
; Strict session mode does not accept an uninitialized session ID, and
; regenerates the session ID if the browser sends an uninitialized session ID.
; Strict mode protects applications from session fixation via a session adoptio
; vulnerability. It is disabled by default for maximum compatibility, but
; enabling it is encouraged.
; https://wiki.php.net/rfc/strict_sessions
session.use_strict_mode = 0

; Whether to use cookies.
; https://php.net/session.use-cookies
session.use_cookies = 1

; https://php.net/session.cookie-secure
session.cookie_secure = on
```

Résultat

Accès sécurisé fonctionnel sur l'adresse https://172.16.2.10.

Note : Le marquage rouge du navigateur est normal car le certificat est auto-signé (non validé par une autorité de certification externe).





☑ **Bilan de la mission —** Objectif atteint. Le protocole HTTPS est actif sur le serveur GLPI. Les échanges sont chiffrés (RSA 2048 bits), la redirection HTTP → HTTPS est en place et le cookie de session est sécurisé (session.cookie_secure = On).

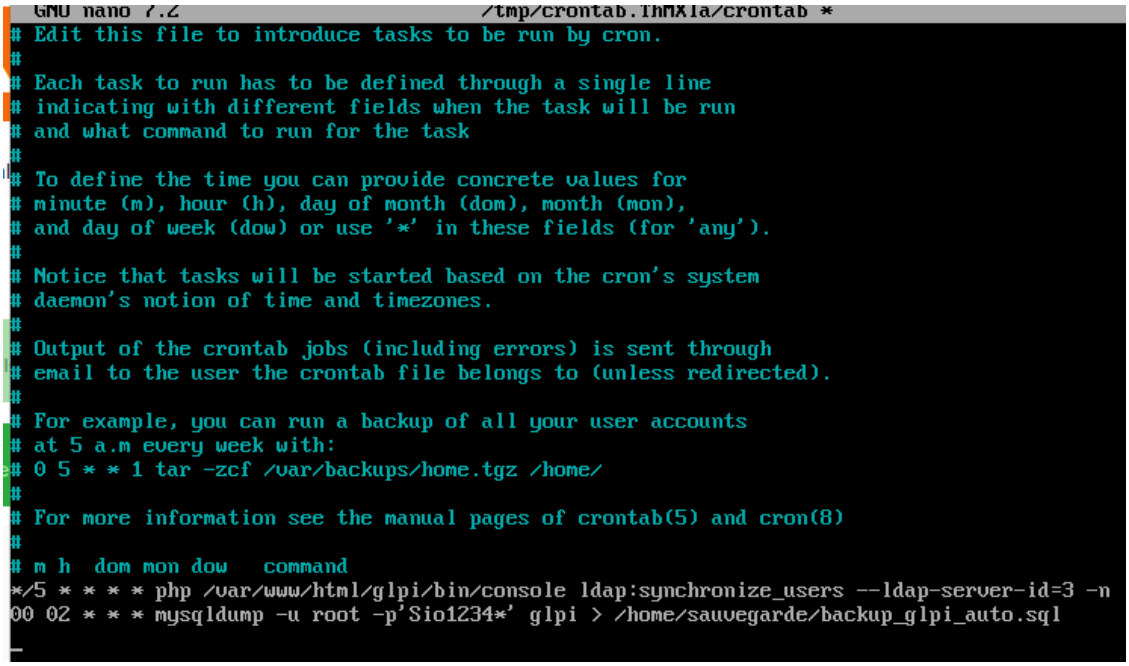
3.11 M4.2 – Sauvegarde de la base de données

Mise en place d'une stratégie de sauvegarde régulière pour prévenir la perte de données (inventaire, tickets, profils) en cas d'incident sur le serveur Debian.

🛠️ Réalisation technique

1. **Création du répertoire** : Un dossier dédié a été créé à la racine (/home/sauvegarde).
2. **Extraction de la base** : Utilisation de l'utilitaire mysqldump pour exporter les données de MariaDB vers un fichier SQL.
3. **Commande utilisée** :
   mysqldump -u root -p glpi > /home/sauvegarde/backup_glpi_$(date +%F).sql

```
root@Serveur-GLPI:~# mkdir /home/sauvegarde
root@Serveur-GLPI:~# chmod 700 /home/sauvegarde
root@Serveur-GLPI:~# mysqldump -u root -p glpi > /home/sauvegarde/backup_glpi_$(date +%F).sql
Enter password:
root@Serveur-GLPI:~#
```



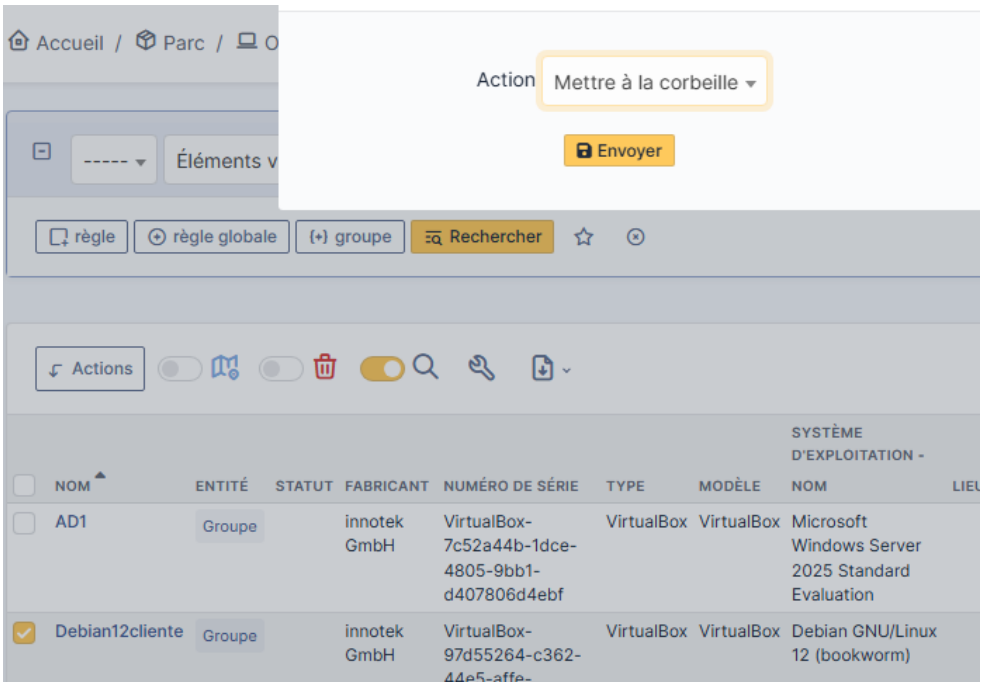
Preuve de réussite

Procédure de Sauvegarde (Backup)

- **Commande utilisée** : mysqldump -u root -p glpi > /home/sauvegarde/backup_glpi_$(date +%F).sql

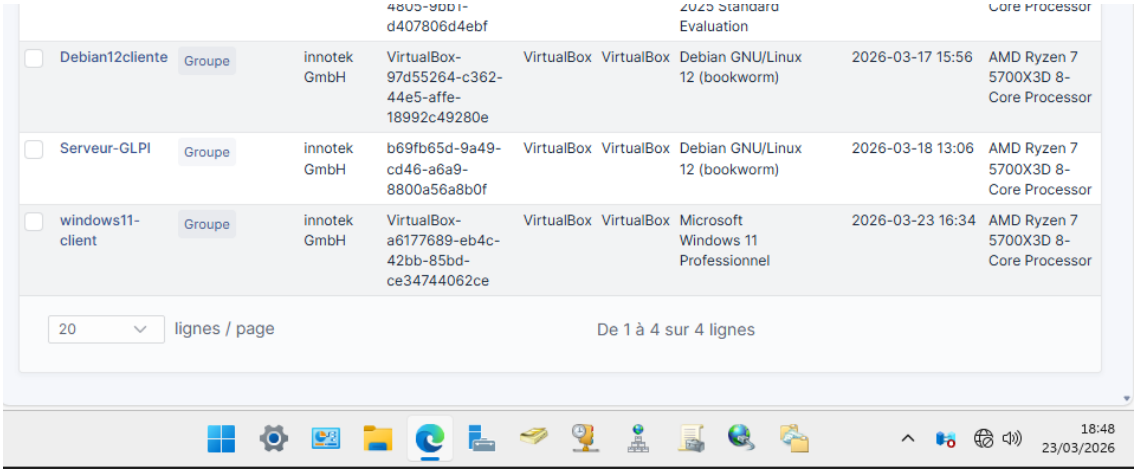
Test de Restauration (Restore)

- **Scénario de panne** : Simulation d'une erreur humaine par la suppression définitive d'un ordinateur du parc informatique dans l'interface GLPI.



- **Action corrective** : Utilisation du fichier de sauvegarde pour restaurer l'état précédent de la base de données.
- **Commande de restauration** : mysql -u root -p glpi < /home/sauvegarde/backup_glpi_2026-03-23.sql

```
root@Serveur-GLPI:~# mysql -u root -p glpi < /home/sauvegarde/backup_glpi_2026-03-23.sql
Enter password:
root@Serveur-GLPI:~# _
```



Bilan de la mission — Objectif atteint. La sauvegarde automatique de la base GLPI est en place via mysqldump et planifiée par cron. Le test de restauration a validé l'intégrité du fichier SQL en restaurant avec succès un ordinateur supprimé.

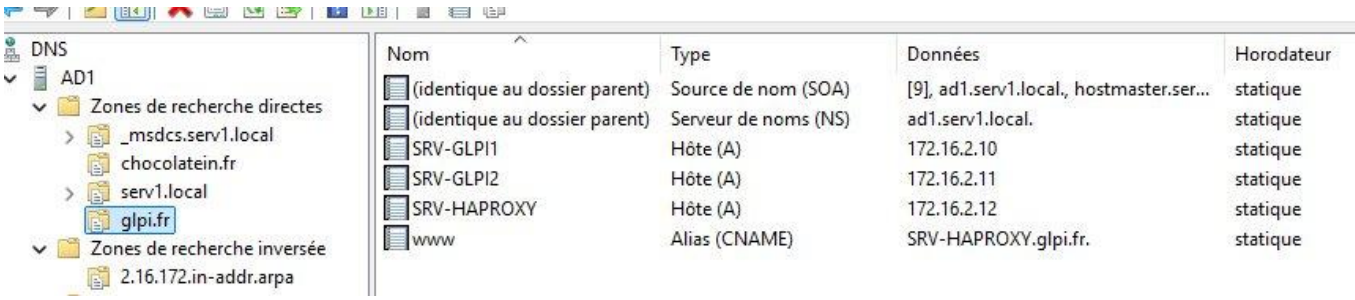
3.12 M5.1 – Cluster Web et HAProxy

Trois objectifs principaux :

- Externaliser la base de données sur un serveur MariaDB dédié (SRV-BDD – 172.16.2.13)
- Mettre en place un second serveur web (SRV-GLPI2 – 172.16.2.11) en redondance
- Déployer HAProxy (172.16.2.12) comme point d'entrée unique avec répartition de charge et HTTPS

I.Configuration DNS (AD Windows)

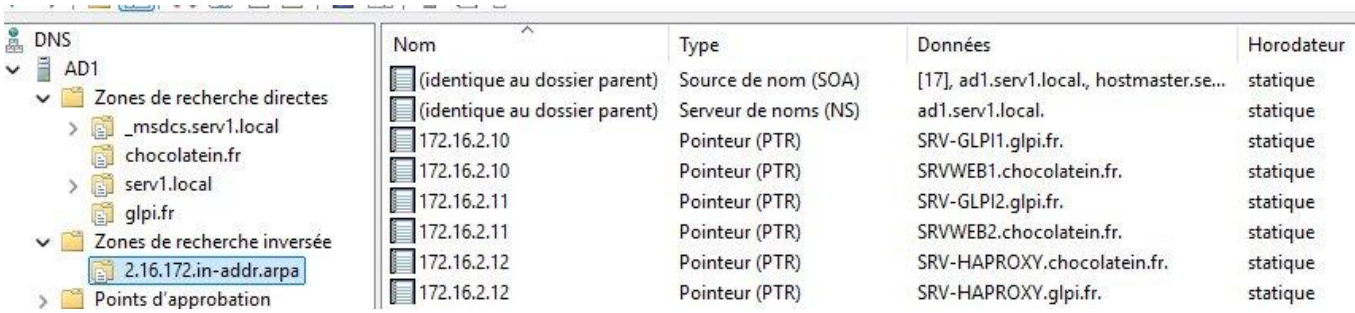
Sur le serveur AD Windows (10.2.0.1), une zone de recherche directe glpi.fr a été créée avec les enregistrements suivants :



Enregistrements créés :

- SRV-GLPI1 (A) → 172.16.2.10
- SRV-GLPI2 (A) → 172.16.2.11
- SRV-HAPROXY (A) → 172.16.2.12
- www (CNAME) → SRV-HAPROXY.glpi.fr

La zone de recherche inverse a également été mise à jour avec les enregistrements PTR correspondants :



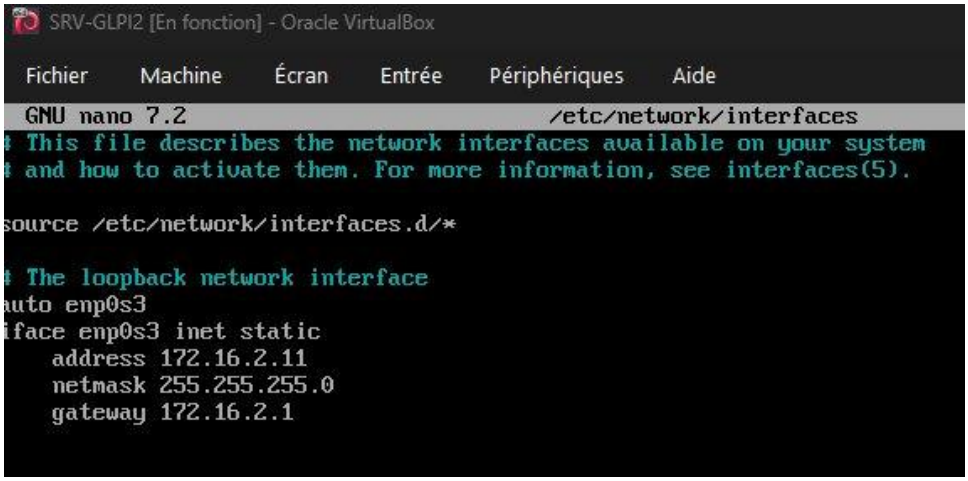
II. Préparation du second serveur web (SRV-GLPI2)

1. Clonage de SRV-GLPI1

La VM SRV-GLPI1 (172.16.2.10) a été clonée dans VirtualBox avec l'option « Générer de nouvelles adresses MAC pour toutes les cartes réseau » (Clone intégral).

2. Modification de l'identité du clone

Le fichier /etc/network/interfaces a été modifié pour attribuer la nouvelle IP :



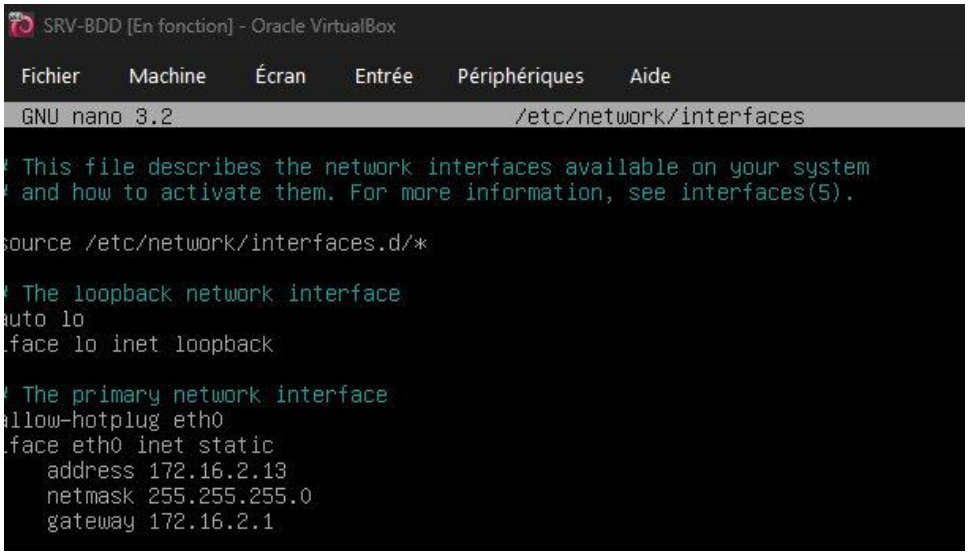
Le hostname a été mis à jour dans /etc/hostname (SRV-GLPI2) puis reboot appliqué.

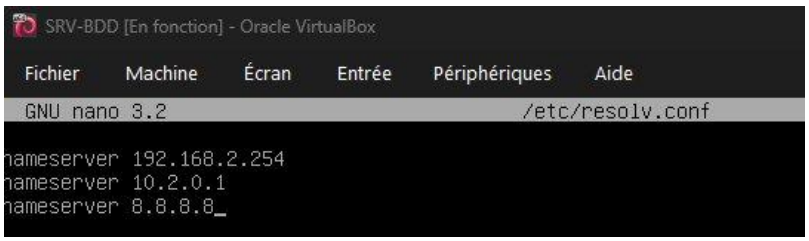
3. Vérification du load balancing

En accédant à www.glpi.fr, la réponse alterne bien entre les deux serveurs – preuve que HAProxy répartit correctement la charge :

III. Déploiement du serveur de base de données (SRV-BDD)

1. Configuration réseau





2. Correction des dépôts Debian Buster

Les dépôts standards de Debian Buster étant archivés, apt update échouait avec des erreurs 404. Le fichier sources.list a été mis à jour :

```
root@SRV-BDD: # apt update
Ign :1 http://security.debian.org/debian-security buster/updates InRelease
Err :2 http://security.debian.org/debian-security buster/updates Release
  404  Not Found [IP : 151.101.194.132 80]
Ign :3 http://ftp.fr.debian.org/debian buster InRelease
Ign :4 http://ftp.fr.debian.org/debian buster-updates InRelease
Err :5 http://ftp.fr.debian.org/debian buster Release
  404  Not Found [IP : 212.27.32.66 80]
Err :6 http://ftp.fr.debian.org/debian buster-updates Release
  404  Not Found [IP : 212.27.32.66 80]
Lecture des listes de paquets... Fait
E: Le dépôt http://security.debian.org/debian-security buster/updates Release ne contient plus de fi
chier Release.
N: Les mises à jour depuis un tel dépôt ne peuvent s'effectuer de manière sécurisée, et sont donc dé
sactivées par défaut.
N: Voir les pages de manuel d'apt-secure(8) pour la création des dépôts et les détails de configurat
ion d'un utilisateur.
E: Le dépôt http://ftp.fr.debian.org/debian buster Release ne contient plus de fichier Release.
N: Les mises à jour depuis un tel dépôt ne peuvent s'effectuer de manière sécurisée, et sont donc dé
sactivées par défaut.
N: Voir les pages de manuel d'apt-secure(8) pour la création des dépôts et les détails de configurat
ion d'un utilisateur.
E: Le dépôt http://ftp.fr.debian.org/debian buster-updates Release ne contient plus de fichier Relea
se.
N: Les mises à jour depuis un tel dépôt ne peuvent s'effectuer de manière sécurisée, et sont donc dé
sactivées par défaut.
N: Voir les pages de manuel d'apt-secure(8) pour la création des dépôts et les détails de configurat
ion d'un utilisateur.
root@SRV-BDD:~#
```

```
  GNU nano 3.2                          /etc/apt/sources.list

#

# deb cdrom:[Debian GNU/Linux 10.10.0 _Buster_ - Official amd64 NETINST 20210619-16:11]

#deb cdrom:[Debian GNU/Linux 10.10.0 _Buster_ - Official amd64 NETINST 20210619-16:11]/

deb http://archive.debian.org/debian/ buster main
deb-src http://archive.debian.org/debian/ buster main

deb http://archive.debian.org/debian-security/ buster/updates main
deb-src http://archive.debian.org/debian-security/ buster/updates main

# buster-updates, previously known as 'volatile'
deb http://archive.debian.org/debian/ buster-updates main
deb-src http://archive.debian.org/debian/ buster-updates main

# This system was installed using small removable media
# (e.g. netinst, live or single CD). The matching "deb cdrom"
# entries were disabled at the end of the installation process.
# For information about how to configure apt package sources,
# see the sources.list(5) manual.
```

```
apt update -o Acquire::Check-Valid-Until=false
apt install mariadb-server -y
```

3. Ouverture des connexions distantes

La directive bind-address a été modifiée dans /etc/mysql/mariadb.conf.d/50-server.cnf :

```
  GNU nano 3.2                  /etc/mysql/mariadb.conf.d/50-server.cnf

#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
[mysqld]

#
# * Basic Settings
#
user                    = mysql
pid-file                = /run/mysqld/mysqld.pid
socket                  = /run/mysqld/mysqld.sock
#port                   = 3306
basedir                 = /usr
datadir                 = /var/lib/mysql
tmpdir                  = /tmp
lc-messages-dir         = /usr/share/mysql
#skip-external-locking

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0

#
```

4. Création de la base de données et de l'utilisateur

```
root@SRV-BDD:~# mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.3.39-MariaDB-0+deb10u2 Debian 10

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE glpidb;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> CREATE USER 'glpi'@'%' IDENTIFIED BY 'Sio1234*';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON glpidb.* TO 'glpi'@'%';
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> EXIT_
```

5. Migration de la base de données

La base existante sur SRV-GLPI1 a été exportée avec mysqldump :

```
root@SRV-GLPI1:~# mysql -u root -p -e "SHOW DATABASES;"
Enter password:
+--------------------+
| Database           |
+--------------------+
| glpi               |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
root@SRV-GLPI1:~# mysqldump -u root -p glpi > backup_glpi.sql
Enter password:
root@SRV-GLPI1:~# _
```

```
root@SRV-GLPI1:~# scp backup_glpi.sql sio@172.16.2.13:/tmp/
sio@172.16.2.13's password:
backup_glpi.sql                                   100% 1382KB  30.7MB/s   00:00
root@SRV-GLPI1:~# _
```

Vérification que les tables GLPI sont bien présentes sur SRV-BDD :

```
root@SRV-BDD:~# ls -lh /tmp/backup_glpi.sql
-rw-r--r-- 1 sio sio 1,4M mars  24 18:53 /tmp/backup_glpi.sql
root@SRV-BDD:~# _
```

IV. Configuration HAProxy (SRV-HAPROXY)

1. Génération du certificat SSL

Un certificat auto-signé a été généré directement sur SRV-HAPROXY :

```
root@SRV-HAPROXY:~# cat /etc/ssl/certs/glpi.crt /etc/ssl/private/glpi.key > /etc/ssl/glpi.pem
```

Fusion de la clé et du certificat en fichier.pem (format requis par HAProxy) :

```
root@SRV-HAPROXY:~# openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/glp
i.key -out /etc/ssl/certs/glpi.crt
Generating a RSA private key
```

2. Configuration initiale de haproxy.cfg

Évolution de la configuration : la version initiale était en HTTP simple (port 80), puis le HTTPS a été ajouté :

Config de base :

Fronted glpi-frontend

Bind * :80

Default_backend fermeweb

backend fermweb

balance roundrobin

server web1 192.168.1.10:80 check

server web2 192.168.1.11:80 check

listen stats

bind * :8080

stats enable

stats url /statsHaproxy

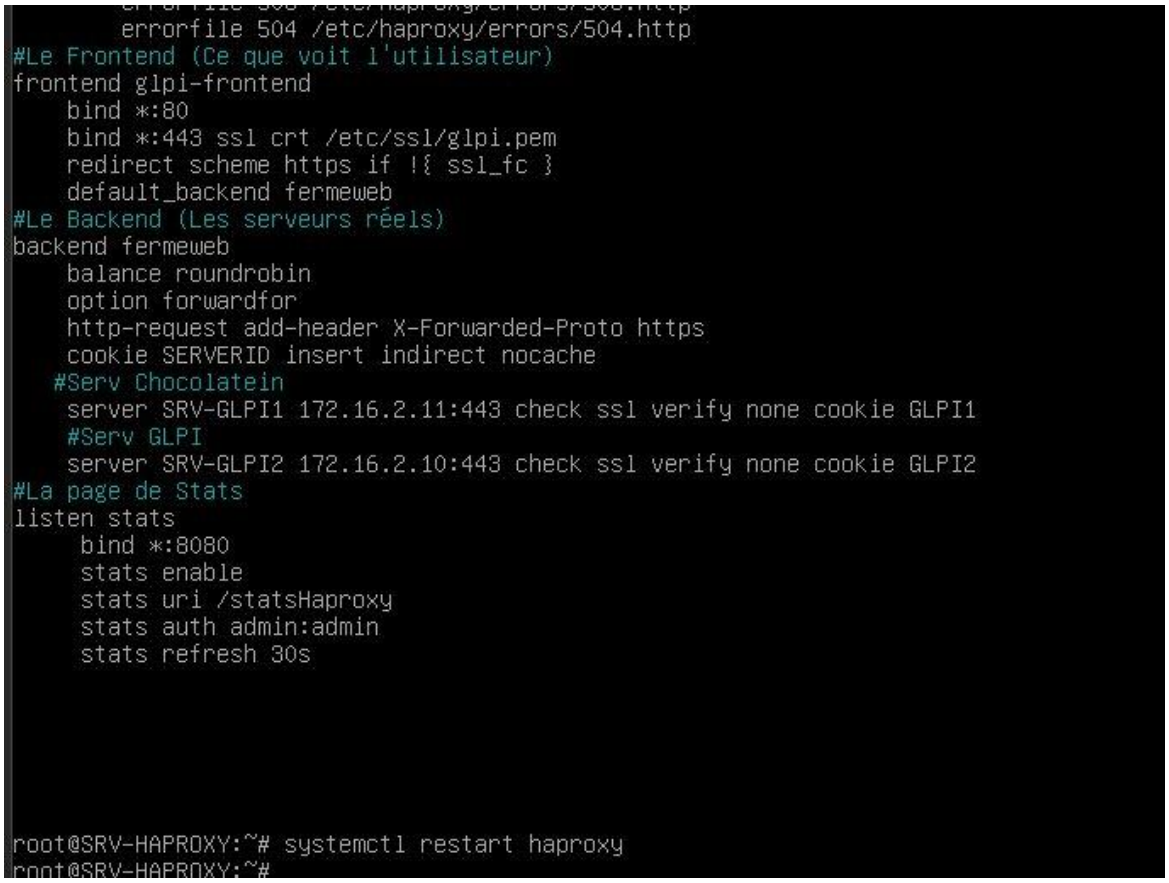
stats auth admin :admin

stats refresh 30s

```
        errorfile 408 /etc/haproxy/errors/408.http
        errorfile 500 /etc/haproxy/errors/500.http
        errorfile 502 /etc/haproxy/errors/502.http
        errorfile 503 /etc/haproxy/errors/503.http
        errorfile 504 /etc/haproxy/errors/504.http
#Le Frontend (Ce que voit l'utilisateur)
frontend glpi-frontend
    bind *:80
    bind *:443 ssl crt /etc/ssl/glpi.pem
    redirect scheme https if !{ ssl_fc }
    default_backend fermeweb
#Le Backend (Les serveurs réels)
backend fermeweb
    balance roundrobin
    option forwardfor
    http-request add-header X-Forwareded-Proto https
    #Serv Chocolatein
    server SRV-GLPI1 172.16.2.11:443 check ssl verify none
    #Serv GLPI
    server SRV-GLPI2 172.16.2.10:443 check ssl verify none
#La page de Stats
listen stats
     bind *:8080
     stats enable
     stats uri /statsHaproxy
     stats auth admin:admin
     stats refresh 30s
```

3. Configuration finale avec Sticky Sessions

La solution finale qui a résolu le problème de session CSRF : ajout de la persistance de session par cookie SERVERID :



Contenu de la configuration finale :

```
frontend glpi-frontend
    bind *:80
    bind *:443 ssl crt /etc/ssl/glpi.pem
    redirect scheme https if !{ ssl_fc }
    default_backend fermeweb

backend fermeweb
    balance roundrobin
    option forwardfor
    http-request add-header X-Forwarded-Proto https
    cookie SERVERID insert indirect nocache
    server SRV-GLPI1 172.16.2.11:443 check ssl verify none cookie GLPI1
    server SRV-GLPI2 172.16.2.10:443 check ssl verify none cookie GLPI2

listen stats
    bind *:8080
    stats enable
    stats uri /statsHaproxy
    stats auth admin:admin
    stats refresh 30s
```

V. Configuration des serveurs web

1. Redirection vers la base de données distante

Sur les deux serveurs (SRV-GLPI1 et SRV-GLPI2), le fichier config_db.php a été modifié :

Version avec avant configuration du proxy :

```
 GNU nano 7.2                    /var/www/html/glpi/config/config_db.php *
<?php
class DB extends DBmysql {
   public $dbhost = '172.16.2.10';
   public $dbuser = 'glpi';
   public $dbpassword = 'Sio1234*';
   public $dbdefault = 'glpidb';
   public $use_utf8mb4 = true;
   public $allow_myisam = false;
   public $allow_datetime = false;
   public $allow_signed_keys = false;
   }
```

Version finale avec configuration du proxy :

```
 GNU nano 7.2                    /var/www/html/glpi/config/config_db.php
<?php
class DB extends DBmysql {
   public $dbhost = '172.16.2.13';
   public $dbuser = 'glpi';
   public $dbpassword = 'Sio1234*';
   public $dbdefault = 'glpidb';
   public $use_utf8mb4 = true;
   public $allow_myisam = false;
   public $allow_datetime = false;
   public $allow_signed_keys = false;
   public $proxy_name = 'www.glpi.fr';
   public $application_url = 'https://www.glpi.fr';
}
```

2. Sécurisation des sessions PHP

Activation de session.cookie_secure dans /etc/php/8.2/apache2/php.ini :

```
; https://php.net/session.cookie-secure
session.cookie_secure = on
```

3. Compatibilité HTTPS derrière proxy

Ajout dans /var/www/html/glpi/inc/define.php pour forcer la reconnaissance du HTTPS :

```
 GNU nano 7.2                    /var/www/html/glpi/inc/define.php *
<?php
$_SERVER['HTTPS'] = 'on';
$_SERVER['SERVER_PORT'] = 443;
/**
```

VI. Ajout des nouvelles VM dans l'inventaire.

Installation du paquet .deb manuellement

1. Télécharge le paquet :
   wget https://github.com/glpi-project/glpi-agent/releases/download/1.10/glpi-agent_1.10-1_all.deb

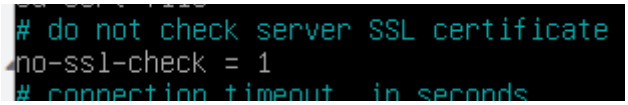
2. Ensuite on l'installe :
   apt install ./glpi-agent_1.10-1_all.deb

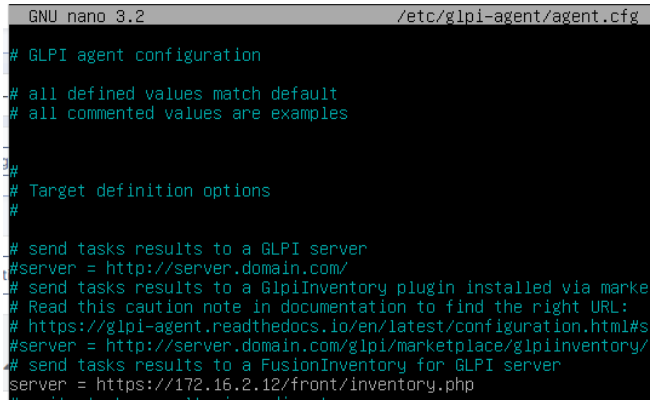
Une fois installé, on n'oublie pas :

on va configurer le fichier agent.cfg

1. Édite le fichier : nano /etc/glpi-agent/agent.cfg
2. ajoutez ces deux lignes :

   - server = https://172.16.2.12/front/inventory.php
   - no-ssl-check = 1

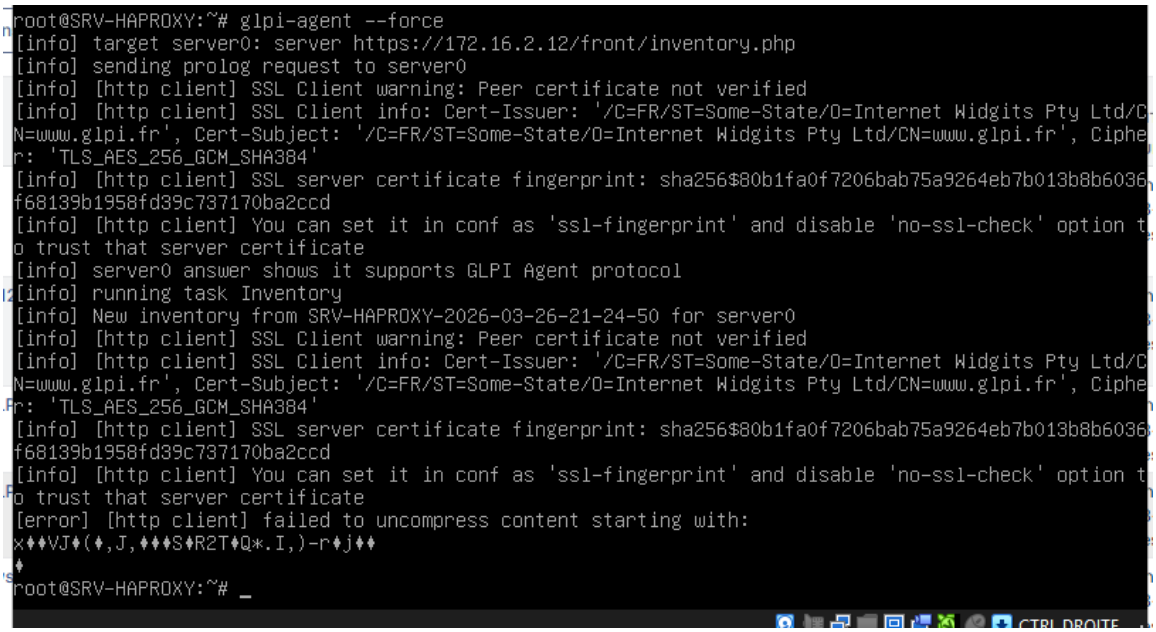


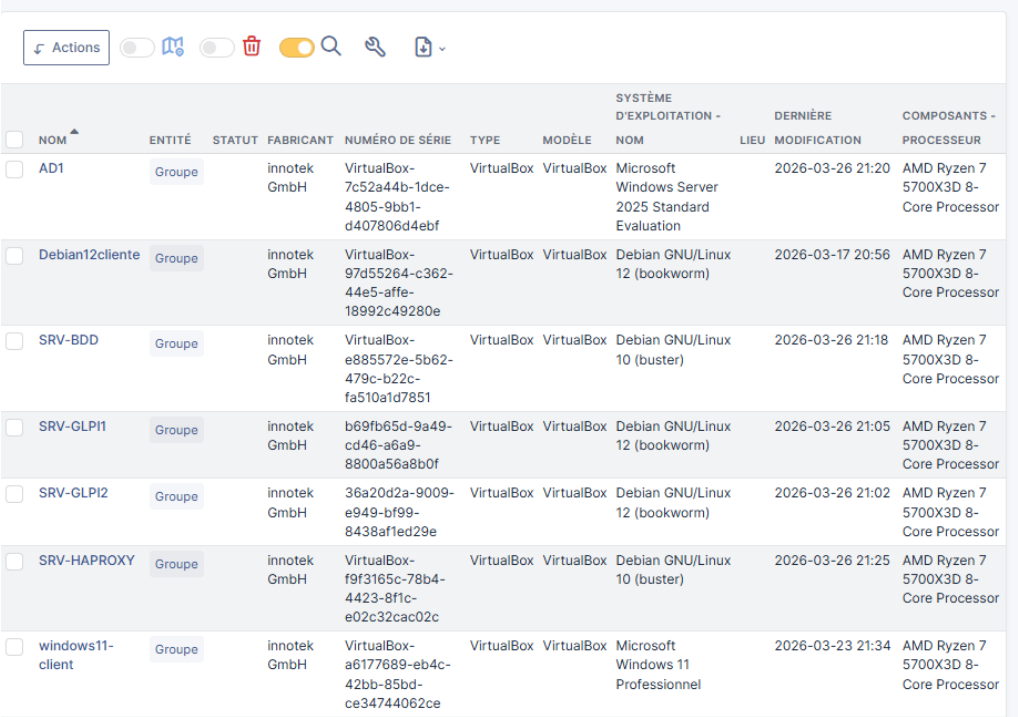


3. Relance l'agent

   systemctl restart glpi-agent

   glpi-agent –force



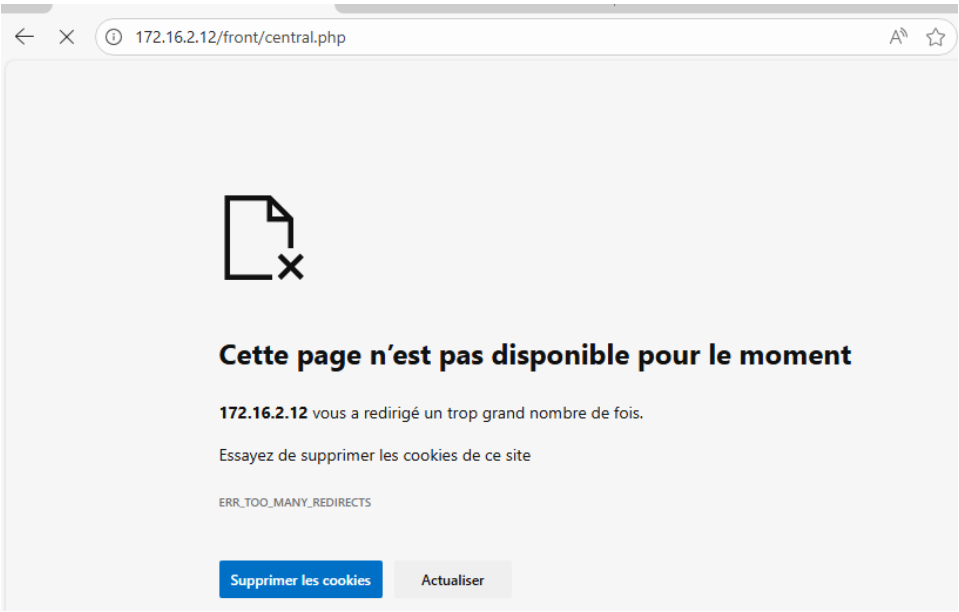


NOM	ENTITÉ	STATUT	FABRICANT	NUMÉRO DE SÉRIE	TYPE	MODÈLE	SYSTÈME D'EXPLOITATION - NOM	LIEU	DERNIÈRE MODIFICATION	COMPOSANTS - PROCESSEUR
AD1	Groupe		innotek GmbH	VirtualBox-7c52a44b-1dce-4805-9bb1-d407806d4ebf	VirtualBox	VirtualBox	Microsoft Windows Server 2025 Standard Evaluation		2026-03-26 21:20	AMD Ryzen 7 5700X3D 8-Core Processor
Debian12cliente	Groupe		innotek GmbH	VirtualBox-97d55264-c362-44e5-affe-18992c49280e	VirtualBox	VirtualBox	Debian GNU/Linux 12 (bookworm)		2026-03-17 20:56	AMD Ryzen 7 5700X3D 8-Core Processor
SRV-BDD	Groupe		innotek GmbH	VirtualBox-e885572e-5b62-479c-b22c-fa510a1d7851	VirtualBox	VirtualBox	Debian GNU/Linux 10 (buster)		2026-03-26 21:18	AMD Ryzen 7 5700X3D 8-Core Processor
SRV-GLPI1	Groupe		innotek GmbH	b69fb65d-9a49-cd46-a6a9-8800a56a8b0f	VirtualBox	VirtualBox	Debian GNU/Linux 12 (bookworm)		2026-03-26 21:05	AMD Ryzen 7 5700X3D 8-Core Processor
SRV-GLPI2	Groupe		innotek GmbH	36a20d2a-9009-e949-bf99-8438af1ed29e	VirtualBox	VirtualBox	Debian GNU/Linux 12 (bookworm)		2026-03-26 21:02	AMD Ryzen 7 5700X3D 8-Core Processor
SRV-HAPROXY	Groupe		innotek GmbH	VirtualBox-f9f3165c-78b4-4423-8f1c-e02c32cac02c	VirtualBox	VirtualBox	Debian GNU/Linux 10 (buster)		2026-03-26 21:25	AMD Ryzen 7 5700X3D 8-Core Processor
windows11-client	Groupe		innotek GmbH	VirtualBox-a6177689-eb4c-42bb-85bd-ce34744062ce	VirtualBox	VirtualBox	Microsoft Windows 11 Professionnel		2026-03-23 21:34	AMD Ryzen 7 5700X3D 8-Core Processor

VII. Problèmes rencontrés et solutions

Problème	Cause	Solution
Erreur « Action non autorisée » via HAProxy	Sans persistance de session, HAProxy changeait de serveur à chaque requête, invalidant le jeton CSRF	Ajout de cookie SERVERID insert indirect nocache dans le backend (Sticky Sessions)
Boucle de redirection ERR_TOO_MANY_REDIRECTS	GLPI forcé en HTTPS mais ne recevait pas le header X-Forwarded-Proto	Ajout du header dans HAProxy + variables $_SERVER dans define.php
Erreur import SQL « Unknown command \- »	Redirection shell incompatible avec le fichier SQL	Utilisation de la commande SOURCE à l'intérieur du prompt MariaDB
apt update échouait en 404	Dépôts Debian Buster obsolètes (archivés)	Migration vers archive.debian.org dans /etc/apt/sources.list
Alerte session.cookie_secure	PHP non configuré pour les cookies HTTPS	session.cookie_secure = On dans php.ini + restart Apache

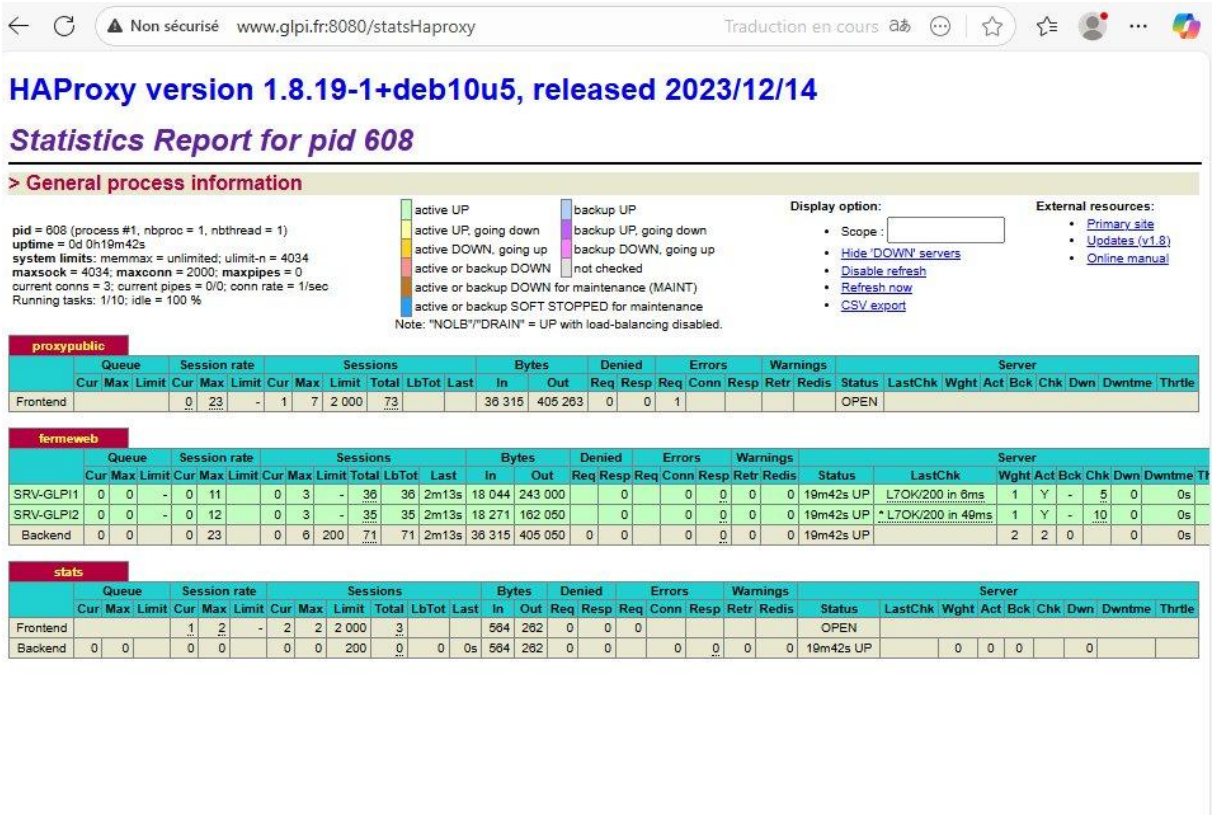
Illustrations des erreurs rencontrées :



VIII. Validation et résultats

1. Page de statistiques HAProxy

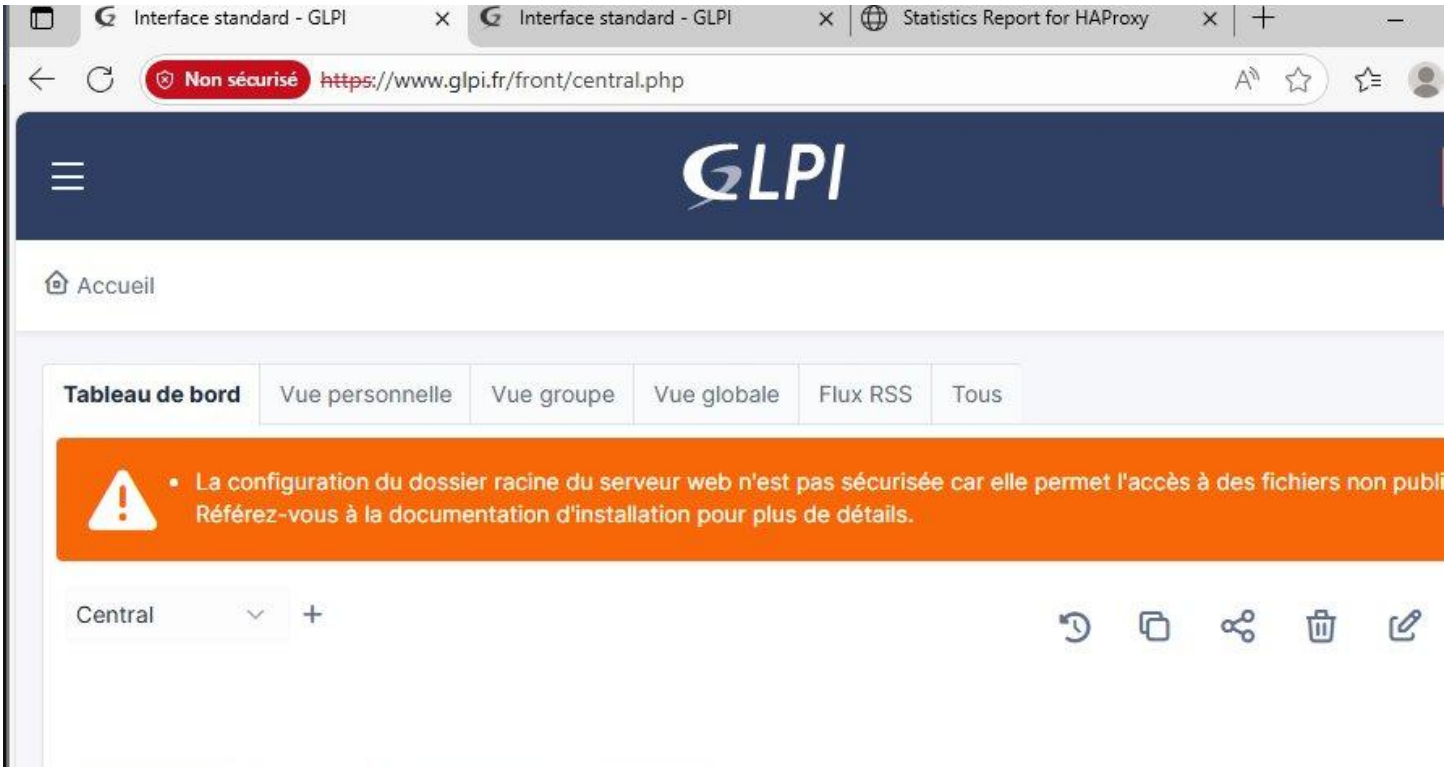
La page de statistiques (http://www.glpi.fr:8080/statsHaproxy) confirme que les deux serveurs web sont actifs (statut UP) et que le trafic est bien réparti en roundrobin :



2. Accès GLPI via www.glpi.fr – Résultat final

Après mise en place des sticky sessions (cookie SERVERID), l'accès à https://www.glpi.fr est pleinement fonctionnel. On constate sur la capture ci-dessous :

- L'URL est bien https://www.glpi.fr/front/central.php (passage par le HAProxy)
- Deux onglets GLPI ouverts simultanément + l'onglet HAProxy Stats – les trois répondent
- Le tableau de bord affiche les données réelles migrées depuis SRV-BDD (1,9K logiciels, 4 ordinateurs…)



Bilan de la mission — Objectif atteint. L'architecture 3-tiers est opérationnelle : HAProxy répartit la charge entre SRV-GLPI1 et SRV-GLPI2, la base de données est externalisée sur SRV-BDD, et l'accès HTTPS via www.glpi.fr fonctionne depuis les postes clients. Les sticky sessions résolvent le problème de jeton CSRF.

5. Scénario de démonstration

Action	Résultat
Accès Utilisateur	Le client Windows accède à https://www.glpi.fr (DNS géré par AD1).
Authentification	Utilisation d'un compte de l'Active Directory.
Test de Résilience	Coupure brutale du serveur GLPI principal.
Continuité	Le ticket en cours est validé sans erreur car la session est maintenue par le HAProxy et les données sont sur la BDD isolée.

6. Bilan Final

Cet atelier professionnel m'a permis de mettre en œuvre une infrastructure informatique complète, en partant d'une installation de base pour aboutir à une architecture de niveau production. J'ai acquis des compétences concrètes en administration système Linux (Debian), en gestion de services réseau (VyOS, DNS, LDAP) et en déploiement applicatif (GLPI, Apache, MariaDB).

La partie la plus complexe a été la mission M5.1, qui m'a confronté à des problèmes réels rencontrés en entreprise : gestion des sessions dans un environnement load-balancé (erreur CSRF), configuration SSL sur un proxy inverse, et migration de base de données. Ces obstacles m'ont appris à diagnostiquer méthodiquement, à utiliser les logs système et à adapter ma configuration étape par étape.

Ce projet m'a également permis de comprendre les enjeux de la haute disponibilité : garantir la continuité de service même en cas de panne d'un composant est une exigence fondamentale en entreprise. L'architecture mise en place (HAProxy + deux serveurs web + BDD déportée) répond directement à cette exigence et constitue un modèle applicable dans un contexte professionnel réel.

7. Liste des compétences couvertes (Référentiel SIO)

- **B1 : Gérer le patrimoine informatique (Inventaire, maintenance des serveurs Debian).**
- **B2 : Répondre aux incidents (Mise en place du Helpdesk et de la redondance).**
- **B3 : Développer la présence en ligne (Configuration HAProxy, certificats SSL, noms de domaine).**
- **B4 : Travailler en mode projet (Découpage en missions M0 à M5, suivi par carte Trello, documentation progressive du compte rendu).**
- **B5 : Mettre à disposition des utilisateurs un service informatique (Déploiement de GLPI accessible depuis les postes clients Windows via https://www.glpi.fr, authentification LDAP et gestion des tickets Helpdesk).**