Sommaire

I. Contexte

Le site de Chasseneuil de TiersLieux86 accueille plusieurs structures hébergées et met à leur disposition une infrastructure informatique commune comprenant un domaine Active Directory, des services réseau et un ensemble de ressources mutualisées. Dans ce cadre, l'entreprise AuditMe doit pouvoir s'appuyer sur un outil centralisé permettant à la fois de gérer le parc informatique, de suivre les incidents utilisateurs et d'améliorer l'exploitation quotidienne des équipements. L'atelier 4 s'inscrit dans cette logique : il vise la mise en place d'une solution de gestion de parc et de support technique intégrée à l'environnement existant du site. Les attendus officiels portent sur l'inventaire des équipements, la gestion des incidents, le déploiement des agents, la sécurité des données et la haute disponibilité du service.

La problématique principale consiste à disposer d'une plateforme unique capable de recenser automatiquement les matériels et logiciels, de centraliser les demandes d'assistance, et de s'appuyer sur l'annuaire Active Directory déjà en place pour authentifier les utilisateurs et gérer leurs habilitations. Dans un environnement professionnel, cette centralisation présente plusieurs intérêts : elle facilite le suivi du parc, améliore la traçabilité des interventions, réduit les tâches d'administration manuelle et permet de structurer le support selon des profils distincts. Le besoin ne se limite donc pas à une installation applicative ; il s'agit de déployer un service exploitable, cohérent avec l'organisation du système d'information du site.

L'environnement technique retenu repose sur une infrastructure virtualisée sous VMware, conforme au cadre de réalisation choisi. La solution mise en œuvre comprend plusieurs machines virtuelles dédiées : un serveur applicatif GLPI sous Debian, un serveur de base de données MariaDB distinct, un contrôleur de domaine Windows Server assurant les services Active Directory, DNS et DHCP, ainsi que des postes clients Windows et Linux utilisés pour les tests d'intégration, d'inventaire et de support. L'architecture est complétée par un serveur HAProxy destiné à la répartition de charge et à la publication du service, ainsi que par un stockage TrueNAS utilisé pour les sauvegardes centralisées via NFS. Dans la réalisation, les rôles sont répartis entre des serveurs dédiés de type SRV-GLPI-01, SRV-GLPI-DB, SRV-GLPI-02 et SRV-HA-GLPI, ce qui traduit une architecture segmentée et plus proche d'un contexte d'exploitation réel.

Le choix de GLPI est pertinent pour une réalisation professionnelle, car cette solution combine deux fonctions majeures d'exploitation : l'inventaire automatisé du parc et la gestion des incidents. L'inventaire repose sur des agents installés sur les équipements afin de remonter les caractéristiques matérielles, logicielles et réseau vers un serveur central. L'intégration à LDAP / Active Directory permet quant à elle de réutiliser les comptes du domaine et d'associer les profils GLPI aux groupes de sécurité, ce qui renforce la cohérence de l'administration. Enfin, les exigences de continuité de service, de sécurisation des échanges HTTPS et de sauvegarde automatisée avec tests de restauration replacent cette activité dans une démarche complète d'administration des systèmes et des réseaux.

II. Objectifs

La réalisation vise à concevoir, déployer et sécuriser une solution de gestion de parc informatique et de support aux utilisateurs reposant sur GLPI, intégrée à l'infrastructure du site de Chasseneuil. L'objectif est de mettre à disposition une plateforme centralisée capable de recenser les équipements, de gérer les demandes d'assistance et de s'appuyer sur les services d'annuaire existants pour l'authentification et l'attribution des droits.

Dans ce cadre, la solution mise en œuvre doit permettre de :

- Déployer une infrastructure GLPI fonctionnelle, reposant sur un serveur applicatif dédié et un serveur de base de données distinct.
- Intégrer GLPI à LDAP / Active Directory afin de centraliser l'authentification des utilisateurs et de gérer les habilitations selon les groupes de sécurité.
- Déployer l'agent GLPI sur les postes et serveurs afin d'automatiser la remontée d'inventaire matériel et logiciel.
- Exploiter GLPI comme outil de gestion des incidents, accessible aux utilisateurs et au support informatique.
- Assurer la continuité de service grâce à une architecture en haute disponibilité s'appuyant sur HAProxy et sur un second serveur applicatif.
- Sécuriser les accès à l'application par la mise en place du HTTPS sur le point d'entrée de la plateforme.
- Garantir la protection et la pérennité des données par une solution de sauvegarde centralisée, automatisée et vérifiée par des tests de restauration.

III. Réalisations professionnelles

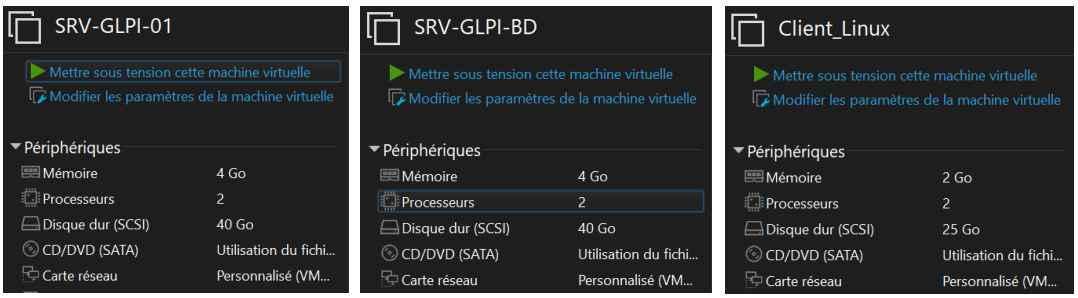
1. Déploiement de l’infrastructure GLPI

La première phase de la réalisation a consisté à mettre en place l’infrastructure technique nécessaire à l’hébergement de la solution GLPI. L’architecture retenue repose sur une séparation des rôles entre le serveur applicatif et le serveur de base de données, afin de faciliter l’administration, la maintenance et les évolutions futures de la plateforme. L’ensemble a été déployé dans un environnement virtualisé sous VMware, conformément au cadre technique retenu pour l’atelier.

* Détails en annexe 1.

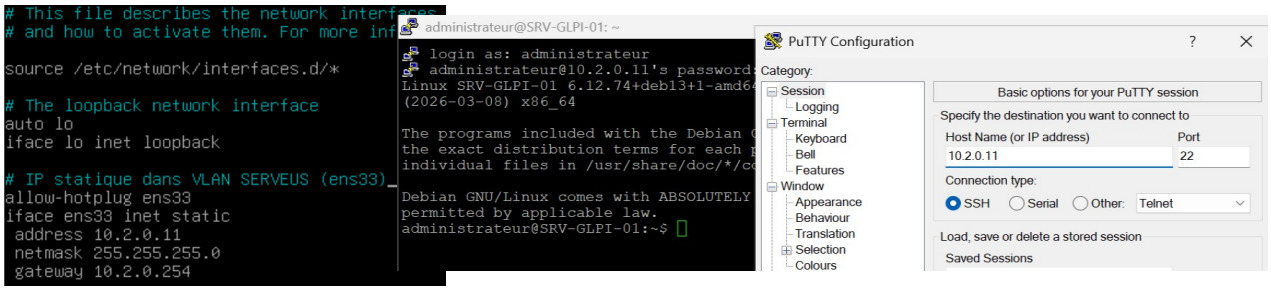
1.1 Installation des machines virtuelles VMware

Les machines virtuelles nécessaires au projet ont été créées dans VMware à partir d’images Debian. Cette étape a permis de mettre en place le serveur applicatif SRV-GLPI-01, le serveur de base de données SRV-GLPI-DB ainsi qu’un poste client Linux destiné aux tests d’intégration et d’inventaire. Les ressources matérielles ont été adaptées au rôle de chaque machine virtuelle afin de disposer d’un environnement cohérent et exploitable pour le déploiement de la solution.



1.2 Configuration réseau du serveur applicatif GLPI

Le serveur SRV-GLPI-01 a ensuite été configuré avec une adresse IP statique sur le réseau serveur afin d’héberger l’application GLPI de manière stable. La configuration réseau a été réalisée manuellement après installation du système, avec définition de l’adresse 10.2.0.11, du serveur DNS interne et du suffixe de recherche du domaine. Cette étape était nécessaire pour permettre la résolution de noms, la communication avec le contrôleur de domaine et l’accès futur à la base de données distante. Les outils d’administration de base ont également été installés afin de poursuivre la configuration en SSH.



1.3 Configuration réseau du serveur de base de données GLPI

Le serveur SRV-GLPI-DB a été déployé à partir d’un clonage du serveur applicatif, puis reconfiguré avec son propre nom d’hôte et sa propre adresse IP afin de remplir uniquement le rôle de serveur de base de données. La configuration réseau a été adaptée pour positionner ce serveur sur l’adresse 10.2.0.13, avec mise à jour des fichiers système liés à l’identité réseau. Cette organisation permet d’isoler la couche de données de la couche applicative, ce qui rend l’architecture plus claire et plus proche d’un contexte d’exploitation réel.

Fichiers modifiés : /etc/network/interfaces et /etc/hosts /etc/hostname

```
# IP statique dans VLAN SERVEUS (ens33)
allow-hotplug ens33
iface ens33 inet static
 address 10.2.0.13
 netmask 255.255.255.0
 gateway 10.2.0.254
```

```
127.0.0.1       localhost
127.0.1.1       SRV-GLPI-01

# fqdn vers glpi.chasseneuilx86.local
10.2.0.11 glpi.chasseneuilx86.local glpi
```

1.4 Installation et sécurisation de MariaDB sur GLPI-DB

Le moteur de base de données MariaDB a été installé sur SRV-GLPI-DB afin d'héberger les données de l'application GLPI. L'installation a été sécurisée avec l'outil mariadb-secure-installation, ce qui a permis de supprimer les accès anonymes, d'interdire l'accès distant au compte administrateur et de supprimer la base de test par défaut. La configuration du service a ensuite été adaptée pour autoriser l'écoute réseau, condition nécessaire à l'accès depuis le serveur applicatif. Enfin, une base glpi et un compte dédié glpiadmin ont été créés pour isoler l'accès applicatif et éviter l'utilisation du compte administrateur de la base.

Fichier modifié : /etc/mysql/mariadb.conf.d/50-server.cnf avec adaptation de la directive bind-address.
Paramètres : base glpi, compte applicatif glpiadmin, accès autorisé depuis le serveur applicatif 10.2.0.11.

```
administrateur@SRV-GLPI-DB:~$ sudo systemctl status mariadb.service
● mariadb.service - MariaDB 11.8.6 database server
     Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
     Active: active (running) since Tue 2026-03-31 19:36:18 CEST; 3min 39s ago
 Invocation: 553b20e111904c81a663eeb871f6c141
       Docs: man:mariadbd(8)
             https://mariadb.com/kb/en/library/systemd/
    Process: 875 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||   VAR=`/
    Process: 959 ExecStartPost=/bin/rm -f /run/mysqld/wsrep-start-position /run/mysqld/wsrep-n
    Process: 961 ExecStartPost=/etc/mysql/debian-start (code=exited, status=0/SUCCESS)
   Main PID: 937 (mariadbd)
     Status: "Taking your SQL requests now..."
      Tasks: 10 (limit: 30426)
```

* Détails en annexe 2.

1.5 Installation et configuration du serveur applicatif GLPI

Le serveur SRV-GLPI-01 a ensuite été préparé pour héberger l'application GLPI. Cette étape a consisté à installer la pile logicielle nécessaire, à déployer les fichiers de l'application, puis à configurer le service web afin de publier GLPI sur le réseau interne à l'aide d'un nom DNS dédié. L'objectif était d'obtenir une plateforme applicative exploitable, distincte de la base de données et intégrée à l'environnement du site.

1.5.1 Installation d'Apache, PHP et des dépendances

Le serveur web Apache2 et l'environnement PHP-FPM ont été installés sur SRV-GLPI-01 avec les extensions nécessaires au fonctionnement de GLPI, notamment pour l'accès à MariaDB, la gestion des archives, le traitement XML, l'internationalisation et les opérations sur les chaînes de caractères. Cette étape permet de fournir à l'application un environnement d'exécution compatible avec les prérequis de GLPI. Le client MariaDB a également été installé afin de permettre la communication avec le serveur de base de données distant.

```
sudo systemctl status apache2
Synchronizing state of apache2.service with SysV service script with /u
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled;
     Active: active (running) since Sat 2026-03-28 21:20:06 CET; 1min 4
 Invocation: 75b0f0cd98f647599c2188e8bc591cb5
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 6810 (apache2)
      Tasks: 55 (limit: 4610)
     Memory: 5.4M (peak: 5.6M)
        CPU: 29ms
     CGroup: /system.slice/apache2.service
             ├─6810 /usr/sbin/apache2 -k start
             ├─6812 /usr/sbin/apache2 -k start
             └─6813 /usr/sbin/apache2 -k start
```

* Détails en annexe 3.

1.5.2 Téléchargement et déploiement de GLPI

Les sources de l'application GLPI ont été téléchargées depuis le dépôt officiel du projet, puis décompressées dans le répertoire d'hébergement web du serveur. La version retenue dans la réalisation est GLPI 11.0.6. Une fois les fichiers extraits, les droits d'accès ont été ajustés pour permettre au service web d'exploiter correctement l'arborescence de l'application. Cette étape met en place les fichiers applicatifs avant leur publication par Apache.

1.5.3 Configuration Apache pour GLPI

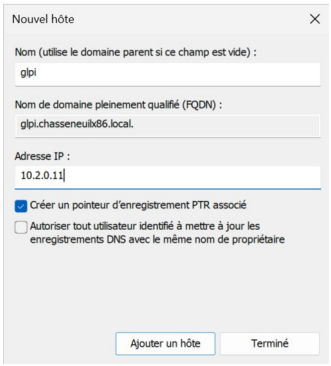
La publication de GLPI a été réalisée par la création d'un VirtualHost dédié dans Apache. La racine documentaire a été positionnée sur le répertoire public de l'application, conformément aux bonnes pratiques recommandées, afin de limiter l'exposition directe de l'arborescence complète. Les modules nécessaires ont ensuite été activés, notamment pour la réécriture d'URL et l'intégration avec PHP-FPM. Cette configuration permet de rendre l'application accessible de manière propre sur le réseau interne tout en respectant une organisation plus sécurisée du service web.

Fichier modifié : /etc/apache2/sites-available/glpi.conf.
Paramètres retenus : ServerName glpi.chasseneuilx86.local et DocumentRoot /var/www/glpi/public.

1.5.4 Création de l'enregistrement DNS

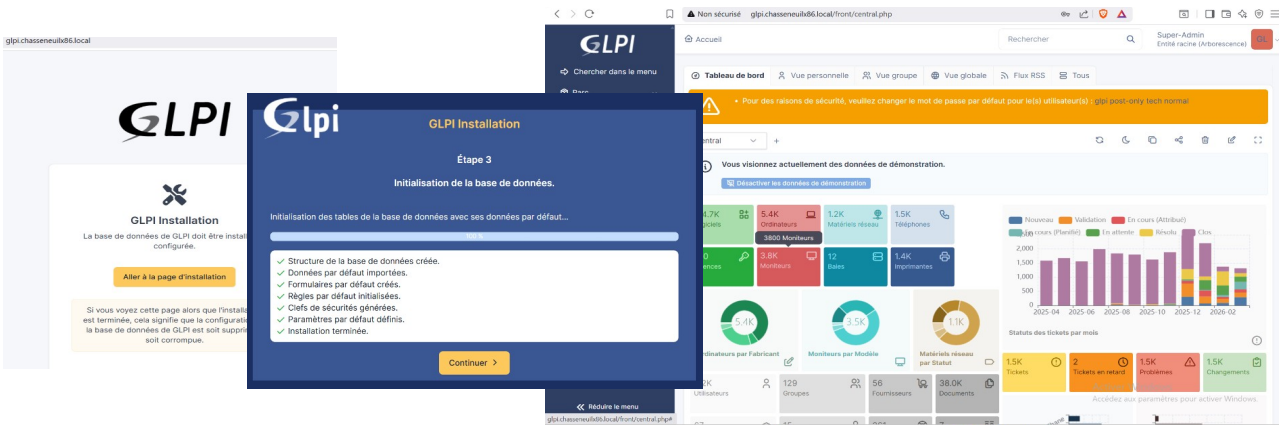
Un enregistrement DNS de type A a été créé sur le serveur de domaine afin d'associer le nom glpi.chasseneuilx86.local à l'adresse IP du serveur applicatif. Cette opération permet de publier la plateforme avec un nom cohérent et plus simple à utiliser que l'adresse IP, aussi bien pour l'administration que pour les accès utilisateurs. Elle prépare également les étapes suivantes d'intégration et de publication du service.

Nom publié : glpi.chasseneuilx86.local.
Adresse associée : 10.2.0.11.



1.5.5 Finalisation de l'installation de GLPI

La finalisation de l'installation a été effectuée depuis l'interface web de GLPI, en suivant l'assistant de configuration. Celui-ci a permis de vérifier la compatibilité de l'environnement, d'établir la connexion avec la base de données distante et d'initialiser les tables de l'application. Une fois l'installation terminée, le fichier install.php a été supprimé afin de sécuriser l'instance, puis les mots de passe des comptes par défaut ont été modifiés. Cette étape marque la mise en service initiale de la plateforme GLPI dans un état exploitable.



2. Intégration LDAP/Active Directory et gestion des habilitations

La deuxième phase de la réalisation a consisté à intégrer GLPI à l'environnement Active Directory déjà en place sur le site de Chasseneuil. L'objectif était de centraliser l'authentification des utilisateurs, de réutiliser l'organisation existante du domaine et d'automatiser l'attribution des droits dans l'application à partir des groupes de sécurité. Cette approche évite la création manuelle de comptes locaux dans GLPI et renforce la cohérence globale de l'administration.

2.1 Proposition d'arborescence pour le client AuditMe

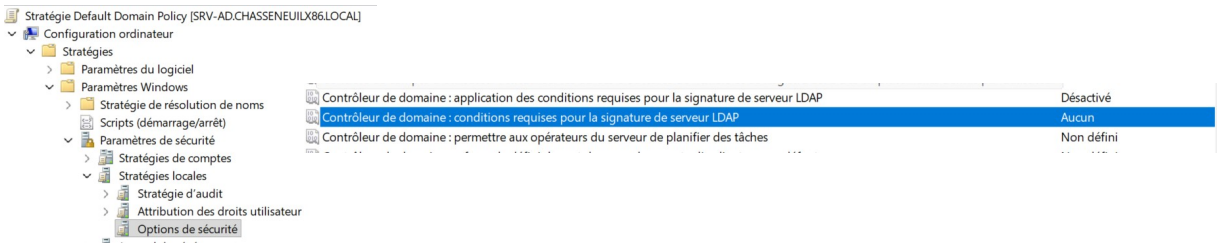
Une arborescence spécifique a été définie dans Active Directory pour intégrer l'entreprise AuditMe au sein de l'environnement mutualisé du site. Cette organisation distingue les utilisateurs, les ordinateurs, les groupes globaux et les groupes domaine locaux, tout en maintenant AuditMe dans l'ensemble plus large des clients entreprises hébergés à Chasseneuil. Cette structuration facilite l'administration, prépare l'application des droits et permet de s'appuyer sur une logique proche du modèle AGDLP, adaptée à une gestion centralisée des accès.

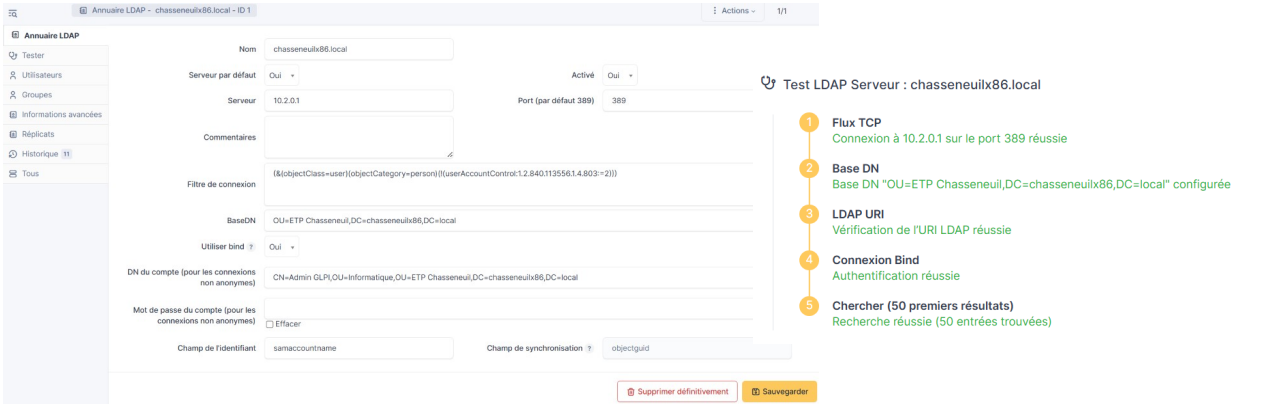
OU=ETP Chasseneuil
- OU=Administration
- OU=Informatique
- OU=Stagiaires
- OU=Adherents
- OU=Clients Entreprises
  - OU=3DPrint86
  - OU=Esporting
  - OU=ValorElec
  - OU=AuditMe
    - OU=Utilisateurs
    - OU=Ordinateurs
    - OU=Groupes globaux
    - OU=Groupes domaine locaux
- OU=Groupes
  - OU=Groupes globaux
  - OU=Groupes domaine locaux

* Détails en Annexe 4.

2.2 Intégration LDAP/Active Directory

L'intégration de GLPI à l'annuaire Active Directory a été réalisée après installation du support LDAP sur le serveur applicatif. L'annuaire a ensuite été déclaré dans l'interface d'administration de GLPI afin de permettre l'authentification des comptes du domaine dans l'application. Les paramètres du serveur LDAP, du Base DN, du compte de liaison et des attributs d'identification ont été renseignés pour établir la communication avec le contrôleur de domaine. Une adaptation de la stratégie de sécurité du domaine a également été nécessaire afin d'autoriser la connexion LDAP depuis GLPI, en désactivant l'exigence de signature du serveur LDAP. Cette configuration permet aux utilisateurs de se connecter à GLPI avec leurs identifiants de domaine, tout en centralisant la gestion des comptes dans Active Directory.





* Détails en Annexe 5.

2.3 Gestion des habilitations

Les habilitations dans GLPI ont été configurées à partir de règles d'attribution dynamiques liées aux groupes Active Directory. Cette méthode permet d'associer automatiquement un profil GLPI à un utilisateur selon son appartenance à un groupe du domaine, sans intervention manuelle dans l'application. Dans la réalisation, le compte admin-glpi reçoit le profil Super-Admin, le groupe GG_Informatique reçoit le profil Admin, le groupe GG_Stagiaires reçoit le profil Technicien, et les autres utilisateurs sont affectés au profil Self-Service. Cette organisation rend la gestion des droits plus lisible, plus évolutive et plus conforme à une logique d'administration centralisée.

Chemin d'accès : Administration > Règles > Règles d'habilitation à un utilisateur.
Principe retenu : attribution automatique des profils GLPI selon les groupes de sécurité du domaine.

admin-glpi > super-admin
service informatique > admin
stagiaires > techniciens
utilisateurs > self-service

3. Déploiement de l'agent GLPI

Cette phase de la réalisation a consisté à déployer l'agent GLPI sur les équipements du parc afin d'automatiser la remontée des informations d'inventaire vers la plateforme centrale. L'objectif est de permettre l'enregistrement des postes et serveurs dans GLPI sans saisie manuelle, en s'appuyant sur une méthode adaptée à chaque environnement : installation directe sur le poste Linux de test et déploiement centralisé par GPO sur les postes Windows du domaine. Cette démarche renforce la fiabilité de l'inventaire, améliore la visibilité sur le parc et facilite l'exploitation quotidienne de la solution.

3.1 Activation de l'inventaire dans GLPI

L'inventaire natif de GLPI a d'abord été activé dans l'interface d'administration afin de permettre la réception et le traitement des remontées envoyées par les agents. Cette étape est indispensable avant tout déploiement sur les postes, car elle prépare la plateforme à enregistrer les équipements détectés dans le parc. L'activation a été réalisée directement dans les paramètres d'administration de l'application.

* Détails en Annexe 6.

3.2 Adhésion au domaine du poste client Linux

Le poste client Linux utilisé pour les tests a été intégré au domaine Active Directory afin de vérifier la compatibilité de l'environnement et de reproduire un cas d'usage réel. Les paquets nécessaires à l'intégration ont été installés, puis la découverte et l'adhésion au domaine ont été réalisées avec les outils prévus pour Debian. Une adaptation de la configuration SSSD a également été appliquée afin d'éviter les blocages liés à certaines GPO Windows non interprétables par un poste Linux. Cette étape permet d'utiliser un poste Linux intégré au domaine tout en le préparant au déploiement de l'agent GLPI.

Outils utilisés : realmd, sssd, adcli, oddjob.

```
administrateur@BAE1TLS01P02:~$ sudo realm join chasseneuilx86.local -U administrateur
Password for administrateur:
administrateur@BAE1TLS01P02:~$ sudo realm -v discover chasseneuilx86.local
 * Resolving: _ldap._tcp.chasseneuilx86.local
 * Performing LDAP DSE lookup on: 10.2.0.1
 * Successfully discovered: chasseneuilx86.local
chasseneuilx86.local
  type: kerberos
  realm-name: CHASSENEUILX86.LOCAL
  domain-name: chasseneuilx86.local
  configured: kerberos-member
  server-software: active-directory
  client-software: sssd
  required-package: sssd-tools
  required-package: sssd
  required-package: libnss-sss
  required-package: libpam-sss
  required-package: adcli
  required-package: samba-common-bin
  login-formats: %U@chasseneuilx86.local
  login-policy: allow-realm-logins
```

3.3 Installation de l'agent GLPI sur les postes Linux

L'agent GLPI a ensuite été installé sur le poste client Linux afin de permettre l'envoi des données d'inventaire vers le serveur GLPI. Dans le cas de la machine Debian utilisée pour la réalisation, l'installation a été effectuée à partir du script d'installation officiel fourni par le projet. Une fois l'agent installé, un premier envoi manuel a été exécuté afin de transmettre l'inventaire vers la plateforme. Cette méthode permet de valider rapidement le bon fonctionnement de la communication entre le poste Linux et le serveur GLPI.

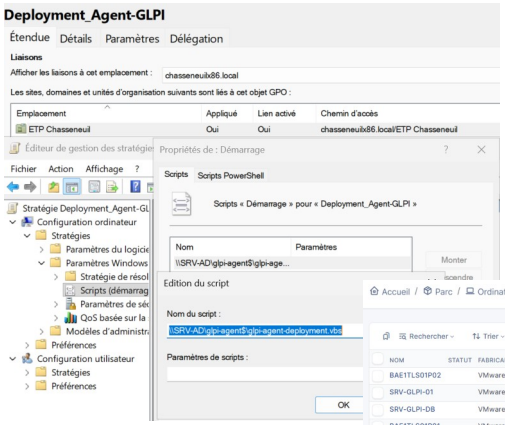
Version utilisée : agent GLPI 1.15 pour Linux.
Serveur cible : http://glpi.chasseneuilx86.local/.

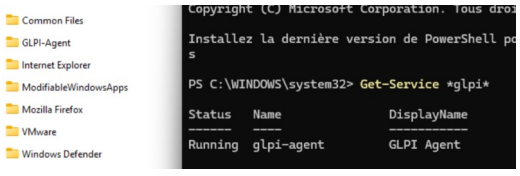
```
administrateur@BAE1TLS01P02:/tmp$ sudo glpi-agent
[info] target server0: server http://glpi.chasseneuilx86.local/
[info] sending prolog request to server0
[info] server0 answer shows it supports GLPI Agent protocol
[info] sending contact request to server0
[info] running task Inventory
[info] New inventory from BAE1TLS01P02-2026-03-30-01-37-24 for server0
```

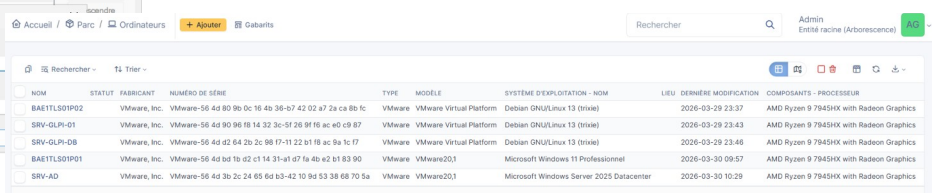
3.4 Installation de l'agent GLPI sur les postes Windows par GPO

Le déploiement de l'agent sur les postes Windows a été industrialisé au moyen d'une stratégie de groupe (GPO). Le script de déploiement fourni avec l'agent a été personnalisé pour définir la version à installer, l'emplacement du package sur le partage réseau et l'adresse du serveur GLPI chargé de recevoir les remontées d'inventaire. Le package d'installation et le script ont ensuite été placés sur un partage accessible par les postes du domaine, puis la stratégie a été liée afin d'automatiser l'installation au démarrage. Cette méthode permet d'assurer un déploiement homogène et centralisé sur les postes Windows sans intervention manuelle machine par machine.

Chemin d'accès : Gestion de stratégie de groupe > Configuration ordinateur > Paramètres Windows > Scripts (démarrage/arrêt).
Paramètres : version 1.16 de l'agent, partage réseau dédié, URL d'inventaire GLPI et tag WINDOWS.







4. Gestion des incidents

Une fois la plateforme GLPI déployée et intégrée à l'annuaire, un usage orienté support utilisateur a été mis en place à travers la gestion des incidents. L'objectif est de permettre aux utilisateurs d'AuditMe de déclarer un problème ou une demande directement dans l'outil, tout en assurant un suivi centralisé par l'équipe informatique. Cette fonctionnalité complète l'inventaire technique du parc en apportant une dimension de helpdesk et de traçabilité des interventions.

4.1 Mode opératoire pour la création d'un ticket dans GLPI

Un mode opératoire a été défini pour encadrer la remontée d'incidents par les utilisateurs depuis le portail GLPI. Après authentification avec leur compte de domaine, les utilisateurs peuvent créer un ticket en choisissant le type de demande, puis en renseignant les informations utiles au traitement, comme l'urgence, la catégorie, le lieu, le titre et la description du problème. Cette démarche permet de structurer les demandes, d'améliorer leur prise en charge par le support et d'assurer un suivi jusqu'à la validation ou l'approbation de la résolution.

* Détails en Annexe 7.

5. Haute disponibilité

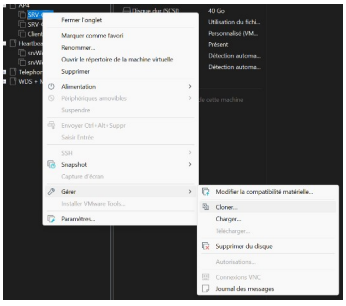
Afin d'améliorer la continuité de service de la plateforme GLPI, une architecture de haute disponibilité applicative a été mise en place. Le principe retenu repose sur l'ajout d'un second serveur applicatif et sur l'utilisation d'un serveur HAProxy chargé de distribuer les requêtes entre les deux nœuds. Cette organisation permet de maintenir l'accès à l'application en cas d'indisponibilité de l'un des serveurs web, tout en conservant une base de données centralisée.

5.1 Création des serveurs supplémentaires dans Vmware

La mise en place de la haute disponibilité a nécessité le déploiement de nouvelles machines virtuelles dans VMware. Un second serveur applicatif, SRV-GLPI-02, a été créé à partir du serveur principal afin de disposer d'un nœud web supplémentaire. Un serveur distinct, SRV-HA-GLPI, a également été préparé pour héberger HAProxy et jouer le rôle de point d'entrée unique de la plateforme. Cette séparation des rôles permet d'isoler la fonction de répartition de charge de la couche applicative.

5.1.1 Configuration du serveur SRV-GLPI-02

Le serveur SRV-GLPI-02 a été configuré comme second nœud applicatif de la plateforme. Après clonage, son identité réseau a été adaptée avec un nouveau nom d'hôte et une adresse IP propre, afin de le différencier du serveur principal. Ce second serveur a ensuite été préparé pour exécuter le même service web que SRV-GLPI-01, de façon à pouvoir recevoir les requêtes distribuées par HAProxy. L'objectif de cette duplication est d'introduire une redondance au niveau applicatif.



```
administrateur@SRV-GLPI-02:~$ hostname
SRV-GLPI-02
administrateur@SRV-GLPI-02:~$ hostname -I
10.2.0.12
administrateur@SRV-GLPI-02:~$ ip route
default via 10.2.0.254 dev ens33 onlink
10.2.0.0/24 dev ens33 proto kernel scope link src 10.2.0.12
administrateur@SRV-GLPI-02:~$
```

```
administrateur@SRV-HA-GLPI:~$ hostname
SRV-HA-GLPI
administrateur@SRV-HA-GLPI:~$ hostname -I
10.2.0.10
administrateur@SRV-HA-GLPI:~$ ip route
default via 10.2.0.254 dev ens33 onlink
10.2.0.0/24 dev ens33 proto kernel scope link src 10.2.0.10
administrateur@SRV-HA-GLPI:~$
```

5.1.2 Configuration de serveur SRV-HA-GLPI

Le serveur SRV-HA-GLPI a été dédié à la fonction de répartition de charge. Il a été préparé pour ne conserver que les éléments nécessaires à l'exécution de HAProxy, sans héberger directement l'application GLPI. Ce choix permet d'utiliser une machine intermédiaire dédiée au frontal applicatif, chargée de recevoir les connexions des utilisateurs et de les rediriger vers l'un des deux serveurs web disponibles. Cette organisation améliore la lisibilité de l'architecture et facilite l'administration du point d'entrée du service.

5.2 Installation et configuration de HAProxy

Le logiciel HAProxy a été installé sur SRV-HA-GLPI afin d'assurer la répartition de charge entre les deux serveurs applicatifs. La configuration mise en place définit un frontend écoutant sur l'adresse de publication du service et un backend composé de SRV-GLPI-01 et SRV-GLPI-02, avec une méthode de distribution de type round robin. Des contrôles d'état ont également été activés afin que le répartiteur puisse détecter l'indisponibilité éventuelle d'un nœud et cesser de lui transmettre du trafic. Cette configuration constitue le cœur de la continuité de service au niveau applicatif.

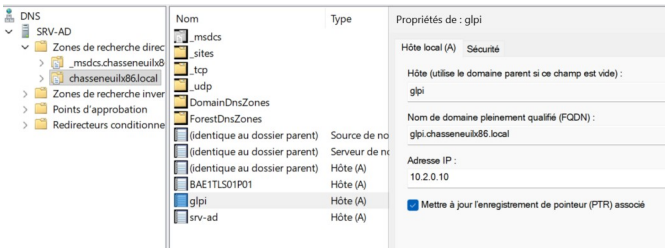
Fichier modifié : /etc/haproxy/haproxy.cfg.
Paramètres retenus : frontal sur 10.2.0.10:80, répartition round robin, backend composé de 10.2.0.11:80 et 10.2.0.12:80.

```
Synchronizing state of haproxy.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable haproxy
● haproxy.service - HAProxy Load Balancer
     Loaded: loaded (/usr/lib/systemd/system/haproxy.service; enabled; preset: enabled)
     Active: active (running) since Mon 2026-03-30 17:59:12 CEST; 714ms ago
 Invocation: 0292da7276fe4e2ea9f52092aea7621a
       Docs: man:haproxy(1)
             file:/usr/share/doc/haproxy/configuration.txt.gz
   Main PID: 10295 (haproxy)
     Status: "Ready."
      Tasks: 3 (limit: 4610)
     Memory: 65.5M (peak: 65.7M)
        CPU: 31ms
     CGroup: /system.slice/haproxy.service
             ├─10295 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/haproxy-master.sock
             └─10298 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/haproxy-master.sock
```

* Détails en Annexe 8.

5.3 Mise à jour de l'enregistrement DNS

L'enregistrement DNS du service GLPI a été modifié afin de faire pointer le nom glpi.chasseneuilx86.local non plus vers le serveur applicatif principal, mais vers le serveur HAProxy. Cette modification permet de publier un point d'accès unique pour les utilisateurs, tout en laissant à HAProxy la responsabilité de distribuer les requêtes vers le nœud disponible. Le service reste ainsi accessible via le même nom, indépendamment du serveur applicatif réellement utilisé en arrière-plan.



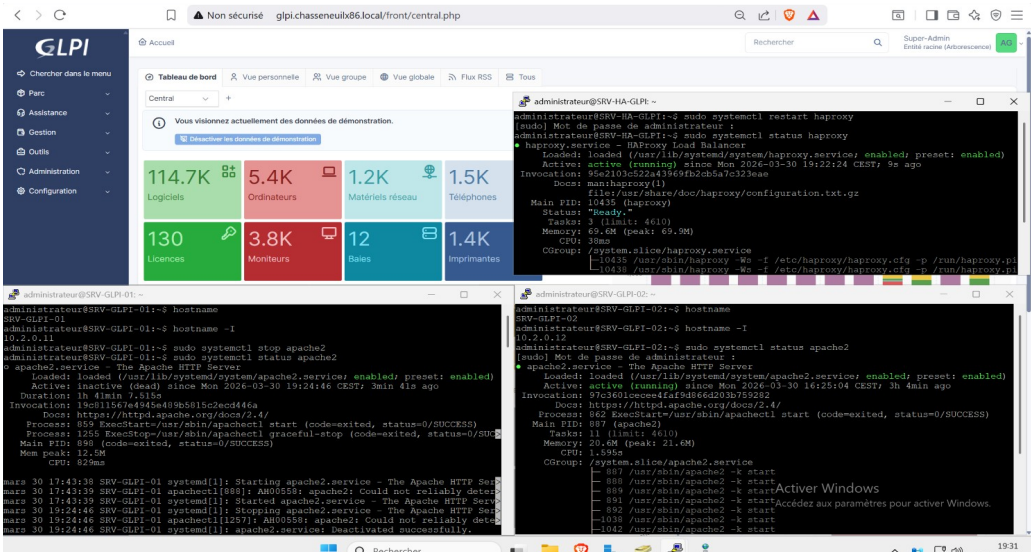
5.4 Configuration de l'accès à la base de données pour SRV-GLPI-02

Afin de permettre au second serveur applicatif de fonctionner dans les mêmes conditions que le premier, un accès dédié à la base de données GLPI a été configuré sur MariaDB. Un compte applicatif autorisé depuis l'adresse IP de SRV-GLPI-02 a été créé, avec les droits nécessaires sur la base glpi. Cette étape permet au second nœud de se connecter à la même base de données centralisée, garantissant ainsi la cohérence fonctionnelle entre les deux serveurs applicatifs.

* Détails en annexe 9.

5.5 Test de vérification de HAProxy

Le fonctionnement de la répartition de charge a été vérifié en simulant l'indisponibilité du serveur applicatif principal. Le service web a été arrêté sur **SRV-GLPI-01**, puis un accès a été réalisé via le nom **glpi.chasseneuilx86.local**. Le maintien de l'accès à la plateforme a confirmé que **HAProxy** redirige correctement les requêtes vers le second nœud applicatif lorsque le premier n'est plus disponible. Ce test valide le principe de continuité de service mis en place au niveau du frontal web.



6. Sécurité des échanges

Une fois la continuité de service mise en place, la sécurisation des accès à la plateforme a été réalisée afin de protéger les échanges entre les utilisateurs et GLPI. L'objectif est de ne plus exposer l'application en HTTP simple, mais de publier le service en HTTPS depuis le serveur HAProxy, qui constitue désormais le point d'entrée unique de la solution. Cette étape permet de renforcer la confidentialité des échanges et d'améliorer le niveau global de sécurité de la plateforme.

6.1 Installation SSL sur HAProxy

La sécurisation des échanges a été mise en œuvre sur SRV-HA-GLPI par l'installation des composants nécessaires à l'utilisation du chiffrement SSL/TLS. Un certificat auto-signé a été généré pour le nom glpi.chasseneuilx86.local, puis la clé privée et le certificat ont été fusionnés dans un fichier PEM exploitable par HAProxy. Ce choix permet de terminer le chiffrement directement sur le frontal applicatif, sans modifier l'architecture interne des serveurs GLPI.

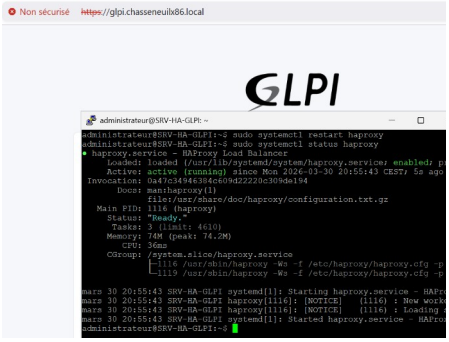
Fichiers créés : /etc/ssl/glpi/glpi.key, /etc/ssl/glpi/glpi.crt, /etc/ssl/glpi/glpi.pem.
Nom couvert par le certificat : glpi.chasseneuilx86.local.

* Détails en Annexe 10

6.2 Configuration HTTPS de HAProxy

La configuration de HAProxy a ensuite été adaptée afin d'ajouter une écoute en 443/TCP avec chargement du certificat SSL, tout en redirigeant automatiquement les requêtes HTTP vers HTTPS. Cette organisation permet d'imposer un accès sécurisé à la plateforme depuis le point d'entrée unique, sans exposition directe des serveurs applicatifs. Le chiffrement est ainsi pris en charge au niveau du frontal, tandis que les serveurs web internes continuent de fonctionner derrière le répartiteur de charge.

Fichier modifié : /etc/haproxy/haproxy.cfg.
Paramètres : frontend HTTPS sur 10.2.0.10:443, certificat PEM chargé par HAProxy et redirection automatique du trafic HTTP vers HTTPS.



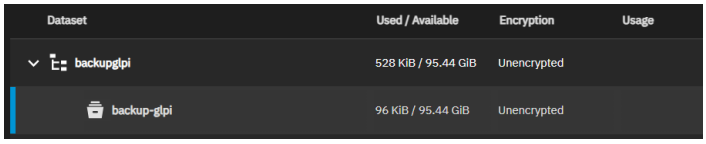
7. Sauvegarde des données

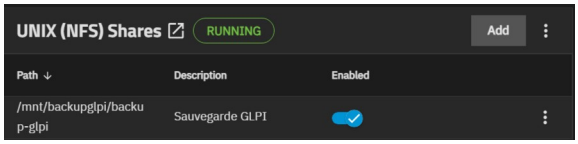
La dernière phase de la réalisation a consisté à mettre en place une stratégie de sauvegarde centralisée pour les données critiques de la plateforme GLPI. L'objectif est de protéger à la fois la base de données et les fichiers applicatifs, afin de pouvoir restaurer le service en cas d'incident logiciel, d'erreur de manipulation ou de panne d'un des serveurs. La solution retenue repose sur un stockage externe TrueNAS accessible en NFS, ce qui permet de dissocier l'emplacement des sauvegardes de l'infrastructure applicative elle-même.

7.1 Création du dataset sur TrueNAS

Un espace de stockage dédié aux sauvegardes a été créé sur le serveur TrueNAS afin d'héberger les données exportées depuis les serveurs GLPI. Un dataset spécifique a été ajouté au pool de stockage pour isoler les sauvegardes de la plateforme et en faciliter l'administration. Cette organisation permet de disposer d'un emplacement centralisé, distinct des serveurs applicatifs, pour conserver les exports de la base et les archives de fichiers.

Chemin : Storage > Pools > Add Dataset.
Dataset créé : backup-glpi.





* Détails en Annexe 11.

7.2 Montage du partage sur les serveurs Linux

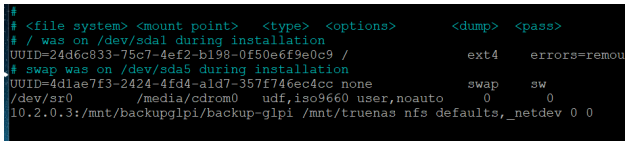
Le dataset de sauvegarde a ensuite été publié en NFS depuis TrueNAS, puis monté sur les serveurs SRV-GLPI-01 et SRV-GLPI-DB. Cette configuration permet à chaque serveur d'écrire directement ses sauvegardes dans un espace réseau commun, accessible depuis l'infrastructure. Des répertoires distincts ont été créés pour séparer les dumps de base de données des archives applicatives, ce qui améliore la lisibilité de l'organisation des sauvegardes.

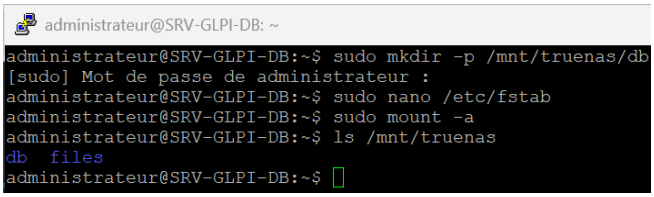
Emplacement monté : /mnt/truenas.
Sous-répertoires créés : /mnt/truenas/db et /mnt/truenas/files.

7.3 Pérennisation du montage NFS

Afin de garantir la disponibilité permanente du partage de sauvegarde après redémarrage des serveurs, le montage NFS a été rendu persistant. Les deux serveurs Linux ont été configurés pour monter automatiquement le partage réseau au démarrage à l'aide du fichier /etc/fstab. Cette étape évite les montages manuels après redémarrage et assure la continuité du mécanisme de sauvegarde automatisée.

Fichier modifié : /etc/fstab.





7.4 Sauvegarde de la base GLPI sur SRV-GLPI-DB

Un script Bash a été créé sur SRV-GLPI-DB pour automatiser l'export de la base glpi à l'aide de mysqldump. Le script génère un fichier horodaté dans le répertoire dédié du partage NFS, puis supprime automatiquement les sauvegardes anciennes selon une politique de rétention simple. Cette méthode permet de disposer d'une sauvegarde quotidienne exploitable pour restaurer les données applicatives en cas d'incident.

Fichier : /opt/scripts/backup_glpi_db.sh.

```
  GNU nano 8.4                    /opt/scripts/backup_glpi_db.sh *
#!/bin/bash

DATE=$(date +%F)
DEST="/mnt/truenas/db"
DB_NAME="glpi"
DB_USER="glpiadmin"
DB_PASS="Sio1234*2A"

mkdir -p $DEST
mysqldump -u $DB_USER -p$DB_PASS $DB_NAME > $DEST/glpi_$DATE.sql
find $DEST -type f -name "*.sql" -mtime +7 -delete
```

* Détails en Annexe 12

7.5 Sauvegarde des fichiers GLPI sur SRV-GLPI-01

Un second script Bash a été mis en place sur SRV-GLPI-01 afin de sauvegarder les fichiers de l'application GLPI. Le contenu du répertoire applicatif est archivé au format compressé, puis stocké sur le partage NFS dans un répertoire distinct de celui de la base de données. Cette sauvegarde complète les exports SQL en couvrant les éléments nécessaires à une restauration globale du service, notamment les fichiers applicatifs et leur arborescence.

Fichier : /opt/scripts/backup_glpi_files.sh.

```
  GNU nano 8.4                  /opt/scripts/backup_glpi_files.sh *
#!/bin/bash

DATE=$(date +%F)
DEST="/mnt/truenas/files"

mkdir -p $DEST

tar -czf $DEST/glpi01_$DATE.tar.gz /var/www/html/glpi

find $DEST -type f -name "glpi01_*.tar.gz" -mtime +7 -delete
```

7.6 Automatisation des sauvegardes

Les deux scripts de sauvegarde ont été planifiés à l'aide de cron afin d'automatiser leur exécution quotidienne. Le script de sauvegarde de la base de données est lancé en premier, puis celui des fichiers applicatifs est exécuté quelques minutes plus tard. Cette planification garantit une exécution régulière sans intervention manuelle et permet de disposer d'un historique de sauvegardes renouvelé automatiquement.

7.7 Vérification de la présence des sauvegardes

La bonne exécution du dispositif a été contrôlée en vérifiant la présence des fichiers générés sur le partage **TrueNAS**. Cette vérification permet de confirmer que les exports SQL et les archives compressées sont bien créés et stockés dans les emplacements prévus. La séparation entre sauvegarde de la base et sauvegarde des fichiers facilite également la lecture du contenu du dépôt de sauvegarde et la préparation d'éventuelles restaurations.

Depuis l'un des serveurs, contrôler :
ls -l /mnt/truenas/db
ls -l /mnt/truenas/files

On doit retrouver :
un dump SQL dans db
une archive GLPI dans files

```
administrateur@SRV-GLPI-01: ~
administrateur@SRV-GLPI-01:~$ ls -ld /opt/scripts
drwxr-xr-x 2 root root 4096 31 mars  01:13 /opt/scripts
administrateur@SRV-GLPI-01:~$ sudo nano /opt/scripts/backup_glpi_files.sh
administrateur@SRV-GLPI-01:~$ sudo chmod +x /opt/scripts/backup_glpi_files.sh
administrateur@SRV-GLPI-01:~$ sudo /opt/scripts/backup_glpi_files.sh
ls -l /mnt/truenas/files
tar: Suppression de « / » au début des noms des membres
tar: /var/www/html/glpi : stat impossible: Aucun fichier ou dossier de ce nom
tar: Arrêt avec code d'échec à cause des erreurs précédentes
total 1
-rw-r--r-- 1 root root 45 31 mars  01:28 glpi01_2026-03-31.tar.gz
administrateur@SRV-GLPI-01:~$
```

7.8 Tests de restauration

Des tests de restauration ont enfin été réalisés afin de valider l'exploitabilité des sauvegardes produites. La restauration de la base de données a été effectuée depuis le dump SQL sur SRV-GLPI-DB, tandis que les fichiers applicatifs ont été restaurés à partir de l'archive compressée sur SRV-GLPI-01. Cette étape est essentielle, car elle permet de confirmer que les sauvegardes ne se limitent pas à de simples exports de fichiers, mais qu'elles peuvent réellement être utilisées pour remettre le service en état de fonctionnement.

* Détails en Annexe 13

IV. Scénarios de tests

Les tests suivants permettent de valider le bon fonctionnement de la plateforme GLPI, son intégration à Active Directory, la haute disponibilité, la sécurisation des échanges et la stratégie de sauvegarde.

Test 1. Accès à la plateforme GLPI
Objectif : vérifier l'accessibilité de GLPI par son nom DNS.
Procédure : ouvrir un navigateur et accéder à glpi.chasseneuilx86.local.
Résultat attendu : la page d'authentification GLPI s'affiche correctement.

Test 2. Authentification LDAP / Active Directory
Objectif : vérifier que les comptes du domaine peuvent se connecter à GLPI.
Procédure : se connecter avec un compte utilisateur puis avec un compte administrateur.
Résultat attendu : l'authentification réussit et le profil attribué correspond au groupe Active Directory.

Test 3. Remontée d'inventaire depuis le poste Linux
Objectif : vérifier le bon fonctionnement de l'agent GLPI sur Linux.
Procédure : lancer un envoi d'inventaire depuis le poste Linux puis contrôler son apparition dans GLPI.
Résultat attendu : le poste Linux apparaît dans l'inventaire avec ses caractéristiques.

Test 4. Déploiement de l'agent GLPI sur un poste Windows
Objectif : vérifier le déploiement automatique de l'agent par GPO.
Procédure : démarrer un poste Windows ciblé, puis contrôler le service GLPI Agent et la remontée dans GLPI.
Résultat attendu : l'agent est installé, actif et le poste est inventorié.

Test 5. Création d'un ticket utilisateur
Objectif : vérifier le fonctionnement de la gestion des incidents.
Procédure : se connecter avec un compte utilisateur, créer un ticket, puis consulter son suivi.
Résultat attendu : le ticket est enregistré et visible dans GLPI.

Test 6. Bascule de service avec HAProxy
Objectif : vérifier la continuité de service en cas d'indisponibilité d'un serveur applicatif.
Procédure : arrêter le service web sur SRV-GLPI-01, puis accéder à GLPI via glpi.chasseneuilx86.local.
Résultat attendu : la plateforme reste accessible via SRV-GLPI-02.

Test 7. Accès sécurisé en HTTPS
Objectif : vérifier la publication sécurisée de la plateforme.
Procédure : accéder à https://glpi.chasseneuilx86.local.
Résultat attendu : GLPI s'ouvre en HTTPS et le trafic HTTP est redirigé vers HTTPS.

Test 8. Vérification de la présence des sauvegardes
Objectif : vérifier la génération des sauvegardes de la base et des fichiers applicatifs.
Procédure : contrôler les répertoires /mnt/truenas/db et /mnt/truenas/files.
Résultat attendu : un dump SQL et une archive GLPI sont présents dans les emplacements prévus.

Test 9. Test de restauration
Objectif : vérifier que les sauvegardes permettent de remettre la plateforme en service.
Procédure : restaurer la base GLPI, restaurer les fichiers applicatifs, puis relancer le service web.
Résultat attendu : la plateforme redevient opérationnelle avec des données exploitables.

V. Bilan technique

La mise en œuvre de cette solution a permis de déployer une plateforme GLPI complète, intégrée à l'infrastructure du site de Chasseneuil et adaptée aux besoins de l'entreprise AuditMe. L'architecture retenue repose sur une séparation claire des rôles entre le serveur applicatif, le serveur de base de données, le répartiteur de charge HAProxy et l'espace de sauvegarde centralisé sur TrueNAS. Cette organisation a permis d'obtenir une solution plus lisible, plus maintenable et plus proche d'un environnement d'exploitation réel.

Sur le plan fonctionnel, la plateforme répond aux objectifs fixés. L'installation de GLPI a permis de centraliser l'inventaire du parc et la gestion des incidents au sein d'un même outil. L'intégration à LDAP / Active Directory a rendu possible l'authentification des utilisateurs à partir des comptes du domaine, ainsi que l'attribution dynamique des profils selon les groupes de sécurité. Le déploiement des agents sur les postes Linux et Windows a permis d'automatiser la remontée des informations matérielles et logicielles, ce qui améliore la fiabilité de l'inventaire et limite les opérations manuelles d'administration.

La solution mise en place présente également un intérêt fort en matière d'exploitation. L'utilisation de HAProxy et d'un second serveur applicatif a permis de mettre en œuvre une première logique de haute disponibilité, validée par des tests de bascule. La publication du service en HTTPS renforce la sécurité des échanges entre les utilisateurs et la plateforme. Enfin, la mise en place d'une sauvegarde centralisée de la base de données et des fichiers applicatifs, avec automatisation par scripts et tests de restauration, contribue à la protection des données et à la continuité du service en cas d'incident.

Cette réalisation reste néanmoins perfectible. Certains points peuvent encore être améliorés, notamment l'usage d'un certificat signé par une autorité de confiance plutôt qu'un certificat auto-signé, le renforcement de la supervision des services, ainsi qu'une meilleure industrialisation de certains paramétrages. De plus, la haute disponibilité mise en place améliore la continuité du service applicatif, mais elle ne constitue pas à elle seule une architecture totalement tolérante aux pannes sur l'ensemble de la chaîne, notamment au niveau de la base de données.

Dans l'ensemble, cette réalisation aboutit à une infrastructure fonctionnelle, centralisée, sécurisée et administrable, capable de répondre aux besoins d'inventaire, d'assistance et d'exploitation du parc informatique d'AuditMe. Elle met en évidence une démarche complète d'administration système et réseau, allant du déploiement initial de la solution jusqu'à la sécurisation, à la continuité de service et à la sauvegarde des données.

VII. Annexes

Annexe 1. Configuration initiale réseau et outils d’administration sur SRV-GLPI-01 et SRV-GLPI-BD

1. Configurer la résolution DNS :
nano /etc/resolv.conf
Renseigner :
nameserver 10.2.0.1
search chasseneuil86.local

Configurer le nom /etc/hosts et /etc/hostname

Renseigner :
SRV-GLPI-01 dans le deux fichier.

Puis redémarrer le service réseau :
systemctl restart networking

2. Mettre à jour les sources APT :
nano /etc/apt/sources.list

Renseigner :
deb http://deb.debian.org/debian/ trixie main non-free contrib non-free-firmware
deb http://security.debian.org/debian-security trixie-security main non-free contrib non-free-firmware
deb http://deb.debian.org/debian/ trixie-updates main non-free contrib non-free-firmware

3. Installer les outils de base :
apt update && apt upgrade -y
apt install ssh -y
apt install sudo -y
usermod -aG sudo administrateur
reboot

4. Installer les outils d’administration complémentaires après connexion SSH :
sudo apt install -y vim curl wget gnupg2 ca-certificates lsb-release apt-transport-https

5. Cloner le serveur applicatif pour créer le serveurs de base de données SRV-GLPI-DB

Configurer les fichiers :
sudo nano /etc/network/interfaces
sudo nano /etc/hosts
sudo nano /etc/hostname
Reinseger :
Nom d’hôte : SRV-GLPI-DB
Adresse IP : 10.2.0.13

Annexe 2. Installation et sécurisation du serveur de base de donnée sur SRV-GLPI-DB

1. Installer mariadb et sécuriser l'installation.
sudo apt install -y mariadb-server
sudo mariadb-secure-installation

Renseigner :
Switch to unix_socket authentication [Y/n] n
Change the root password? [Y/n] y
Remove anonymous users? [Y/n] y
Disallow root login remotely? [Y/n] y
Remove test database and access to it? [Y/n] y
Reload privilege tables now? [Y/n] y

2. Configurer le service mariadb
sudo nano /etc/mysql/mariadb.conf.d/50-server.cnf

Modifier la ligne :
bind-address = 0.0.0.0

sudo systemctl restart mariadb
sudo systemctl enable mariadb
sudo ss -tulpn | grep 3306

3. Créer la base GLPI et compte applicatif
sudo mariadb -u root -p

Reinseigner :
CREATE DATABASE glpi;
CREATE USER 'glpiadmin'@'10.2.0.11' IDENTIFIED BY 'Sio1234*';
GRANT ALL PRIVILEGES ON glpi.* TO 'glpiadmin'@'10.2.0.11';
FLUSH PRIVILEGES;
EXIT;

Annexe 3. Installation du serveur applicatif GLPI sur SRV-GLPI-01

1. Installer les dépendances apacher et php
sudo apt-get install apache2 php8.4-fpm mariadb-client unzip bzip2 -y
sudo apt install php8.4-{curl,gd,intl,mysql,zip,bcmath,mbstring,xml,bz2} -y

sudo systemctl enable apache2
sudo systemctl start apache2
sudo systemctl status apache2

2. Télécharger et déployer GLPI
cd /tmp
wget https://github.com/glpi-project/glpi/releases/download/11.0.6/glpi-11.0.6.tgz
sudo tar -xzvf glpi-11.0.6.tgz -C /var/www/
sudo chown www-data /var/www/glpi/ -R

3. Configurer Apache
sudo nano /etc/apache2/sites-available/glpi.conf

Reinseigner :
```
<VirtualHost *:80>
    ServerName glpi.chasseneuilx86.local
    DocumentRoot /var/www/glpi/public
    <Directory /var/www/glpi/public>
        Require all granted
        RewriteEngine On
        RewriteCond %{HTTP:Authorization} ^(.+)$
        RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ index.php [QSA,L]
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/glpi_error.log
    CustomLog ${APACHE_LOG_DIR}/glpi_access.log combined
</VirtualHost>
```

sudo a2enmod rewrite proxy_fcgi setenvif
sudo a2enconf php8.4-fpm
sudo a2dismod php8.4
sudo systemctl restart php8.4-fpm
sudo systemctl restart apache2

4. Créer un enregistrement DNS de type A sur le serveur du domaine :
- Nom : glpi.chasseneuilx86.local
- Adresse IP : 10.2.0.11

5. Finaliser ensuite l’installation depuis l’interface web de GLPI en renseignant la base distante glpi et le compte applicatif glpiadmin.

6. Supprimer le script d’installation et modifier les mots de passe par défaut.
sudo rm /var/www/glpi/install/install.php

Annexes 4. Nouvelle arborescence pour le client AuditMe

```
# Creation des OU
Import-Module ActiveDirectory

# Variables
$base = "OU=ETP Chasseneuil,DC=chasseneuilx86,DC=local"

# Création des nouvelles OU dans la structure existante
New-ADOrganizationalUnit -Name "Informatique" -Path $base
New-ADOrganizationalUnit -Name "Stagiaires" -Path $base

# Entreprises
$entreprises = "AuditMe"

foreach ($ent in $entreprises) {
    $pathEnt = "OU=Clients Entreprises,$base"
    New-ADOrganizationalUnit -Name $ent -Path $pathEnt
    $entOU = "OU=$ent,$pathEnt"
    New-ADOrganizationalUnit -Name "Groupes globaux" -Path $entOU
    New-ADOrganizationalUnit -Name "Groupes domaine locaux" -Path $entOU
    New-ADOrganizationalUnit -Name "Utilisateurs" -Path $entOU
    New-ADOrganizationalUnit -Name "Ordinateurs" -Path $entOU
}

# Création des utilisateurs du service informatique
$itOU = "OU=Informatique,$base"
$Password = ConvertTo-SecureString "Sio1234*" -AsPlainText -Force

$it = @(
    @{Nom="Lefevre"; Prenom="Nina"},
    @{Nom="Garcia"; Prenom="Mateo"},
    @{Nom="Kowalski"; Prenom="Jakub"},
    @{Nom="Patel"; Prenom="Anaya"},
    @{Nom="Morel"; Prenom="Hugo"}
)

foreach ($user in $it) {
    $login = ($user.Prenom.Substring(0,1) + $user.Nom).ToLower()

    New-ADUser `
        -Name "$($user.Prenom) $($user.Nom)" `
        -GivenName $user.Prenom `
        -Surname $user.Nom `
        -SamAccountName $login `
        -UserPrincipalName "$login@chasseneuilx86.local" `
        -Path $itOU `
        -AccountPassword $Password `
        -Enabled $true `
        -PasswordNeverExpires $true
}

# Création des utilisateurs stagiaires
$stagiairesOU = "OU=Stagiaires,$base"

$stagiaires = @(
    @{Nom="Roussel"; Prenom="Clara"},
    @{Nom="Fernandez"; Prenom="Lucia"},
    @{Nom="Novak"; Prenom="Elena"},
    @{Nom="Reddy"; Prenom="Arjun"},
    @{Nom="Bernier"; Prenom="Louis"}
)

foreach ($user in $stagiaires) {
    $login = ($user.Prenom.Substring(0,1) + $user.Nom).ToLower()

    New-ADUser `
        -Name "$($user.Prenom) $($user.Nom)" `
        -GivenName $user.Prenom `
        -Surname $user.Nom `
        -SamAccountName $login `
        -UserPrincipalName "$login@chasseneuilx86.local" `
        -Path $stagiairesOU `
        -AccountPassword $Password `
        -Enabled $true `
```

```
        -PasswordNeverExpires $true
}

# Création des utilisateurs de AuditMe
$auditOU = "OU=Utilisateurs,OU=AuditMe,OU=Clients Entreprises,$base"

$audit = @(
    @{Nom="Dubois"; Prenom="Emma"},
    @{Nom="Torres"; Prenom="Diego"},
    @{Nom="Muller"; Prenom="Sofia"},
    @{Nom="Sharma"; Prenom="Ishan"},
    @{Nom="Fontaine"; Prenom="Jules"}
)

foreach ($user in $audit) {
    $login = ($user.Prenom.Substring(0,1) + $user.Nom).ToLower()

    New-ADUser `
        -Name "$($user.Prenom) $($user.Nom)" `
        -GivenName $user.Prenom `
        -Surname $user.Nom `
        -SamAccountName $login `
        -UserPrincipalName "$login@chasseneuilx86.local" `
        -Path $auditOU `
        -AccountPassword $Password `
        -Enabled $true `
        -PasswordNeverExpires $true
}

# Variables pour les groupes
$OU_Global = "OU=Groupes globaux,OU=Groupes,OU=ETP Chasseneuil,DC=chasseneuilx86,DC=local"
$ENT_Audit_GG = "OU=Groupes globaux,OU=AuditMe,OU=Clients Entreprises,$base"

# Variables pour les utilisateurs
$OU_it = "OU=Informatique,$base"
$OU_Stagiaires = "OU=Stagiaires,$base"
$OU_Ent_Audit = "OU=Utilisateurs,OU=AuditMe,OU=Clients Entreprises,$base"

# Création des groupes globaux
New-ADGroup -Name "GG_Informatique" -GroupScope Global -GroupCategory Security -Path $OU_Global
New-ADGroup -Name "GG_Stagiaires" -GroupScope Global -GroupCategory Security -Path $OU_Global
New-ADGroup -Name "GG_AuditMe" -GroupScope Global -GroupCategory Security -Path $ENT_Audit_GG

# Peuplement de groupes globaux
# Informatique
Get-ADUser -Filter * -SearchBase $OU_it | ForEach-Object {
    Add-ADGroupMember -Identity "GG_Informatique" -Members $_
}

# Stagiaires - Correction du commentaire
Get-ADUser -Filter * -SearchBase $OU_Stagiaires | ForEach-Object {
    Add-ADGroupMember -Identity "GG_Stagiaires" -Members $_
}

# AuditMe - Correction du commentaire (anciennement "Esporting")
Get-ADUser -Filter * -SearchBase $OU_Ent_Audit | ForEach-Object {
    Add-ADGroupMember -Identity "GG_AuditMe" -Members $_
}
```

Annexe 5. Configuration LDAP et règles d’habilitation GLPI

1. Intégration de GLPI à l’annuaire LDAP / Active Directory

Installer le support LDAP sur le serveur applicatif :
sudo apt-get update
sudo apt-get install php-ldap
Déclarer l’annuaire dans GLPI :
Chemin d’accès : Configuration > Authentification > Annuaires LDAP > Ajouter
Renseigner les paramètres de l’annuaire :

- serveur LDAP
- port
- Base DN
- compte de liaison
- attribut d’identification
- paramètres de synchronisation

Activer l’authentification LDAP dans GLPI.
Tester la connexion à l’annuaire.

2. Principe d’attribution des profils
Les profils GLPI sont attribués automatiquement en fonction de l’appartenance des utilisateurs à des groupes Active Directory.

Correspondances retenues :
admin-glpi → Super-Admin
GG_Informatique → Admin
GG_Stagiaires → Technicien
autres utilisateurs du domaine → Self-Service

3. Création d’une règle pour le compte d’administration GLPI

Chemin d’accès : Administration > Règles > Règles d’habilitation à un utilisateur

Créer une règle avec les paramètres suivants :
Nom : admin-glpi_superadmin
Description : Affecte le profil Super-Admin au compte LDAP admin-glpi
Opérateur logique : ET
Actif : Oui
Critère :
Champ : Login
Condition : est
Valeur : admin-glpi
Action :
Attribuer le profil : Super-Admin

4. Création d’une règle pour le groupe GG_Informatique
Créer une règle avec les paramètres suivants :
Nom : GG_Informatique_admin
Critère :
Champ : LDAP directory : memberOf
Condition : contient
Valeur : CN=GG_Informatique,OU=Groupes globaux,OU=Groupes,OU=ETP Chasseneuil,DC=chasseneuilx86,DC=local
Actions :
Attribuer le profil : Admin
Attribuer l’entité : Entité racine
Récursif : Oui

5. Création d’une règle pour le groupe GG_Stagiaires
Créer une règle avec les paramètres suivants :
Nom : GG_Stagiaires_technicien
Critère :
Champ : LDAP directory : memberOf
Condition : contient
Valeur : CN=GG_Stagiaires,OU=Groupes globaux,OU=Groupes,OU=ETP Chasseneuil,DC=chasseneuilx86,DC=local
Actions :
Attribuer le profil : Technicien
Attribuer l’entité : Entité racine
Récursif : Oui

6. Résultat attendu
Après authentification via l’annuaire :
le compte admin-glpi obtient le profil Super-Admin ;
les utilisateurs du groupe GG_Informatique obtiennent le profil Admin ;
les utilisateurs du groupe GG_Stagiaires obtiennent le profil Technicien ;
les autres comptes du domaine disposent du profil Self-Service.

Annexe 6. Déploiement de l’agent GLPI sur les postes Windows par GPO

1. Récupérer le script de déploiement de l’agent GLPI.
Utiliser le fichier glpi-agent-deployment.vbs.

2. Adapter le script avec les paramètres du site.
Modifier les lignes suivantes :
SetupVersion = "1.16"
SetupLocation = "\\SRV-AD\\glpi-agent$"
SetupOptions = "/quiet RUNNOW=1 ADD_FIREWALL_EXCEPTION=1 SERVER='http://glpi.chasseneuilx86.local/front/inventory.php'
TAG=WINDOWS"

3. Copier les fichiers sur un partage réseau.
Placer dans le partage :
le script glpi-agent-deployment.vbs
le package d’installation .msi de l’agent

4. Créer et lier la GPO de déploiement.
Chemin d’accès : Gestion de stratégie de groupe > Configuration ordinateur > Paramètres Windows > Scripts (démarrage/arrêt)
Associer le script de déploiement à la stratégie, puis lier la GPO à l’unité d’organisation contenant les postes concernés.

5. Mettre à jour les stratégies sur le poste client.
Redémarrer le poste, ou forcer l’actualisation des stratégies de groupe.

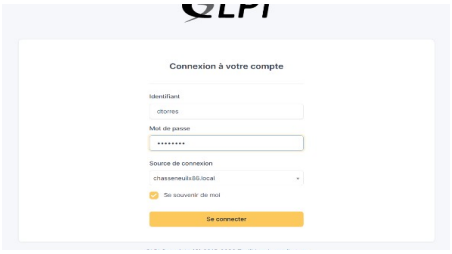
6. Vérifier l’installation de l’agent.
Contrôler la présence du service GLPI Agent sur le poste Windows.

7. Compléter la configuration si nécessaire.
Si le serveur d’inventaire n’est pas remonté automatiquement, ajouter la ligne suivante dans le fichier :
C:\Program Files\GLPI-Agent\etc\agent.cfg

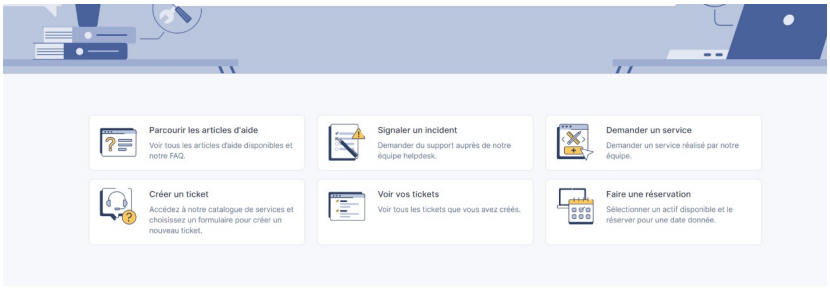
Contenu à ajouter :
server = http://glpi.chasseneuilx86.local/front/inventory.php

Annexe 7. Création d'un ticket dans GLPI

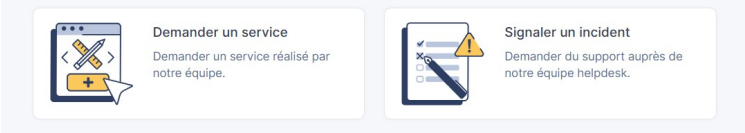
1. Ouvrir le portail GLPI sur : http://glpi.chasseneuilx86.local/
2. Se connecter avec son compte de domaine.



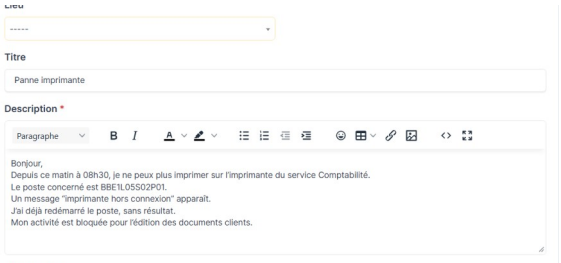
3. Cliquer sur Créer un ticket.



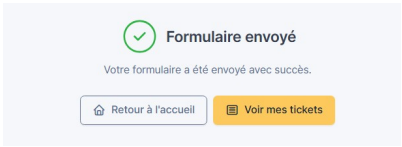
4. Choisir entre Demander un service et Signaler un incident.



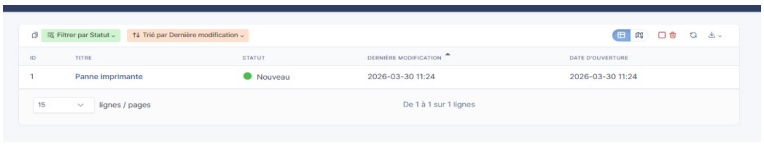
5. Renseigner les informations suivants :
   - Urgence
   - Catégorie
   - Lieu
   - Titre
   - Description



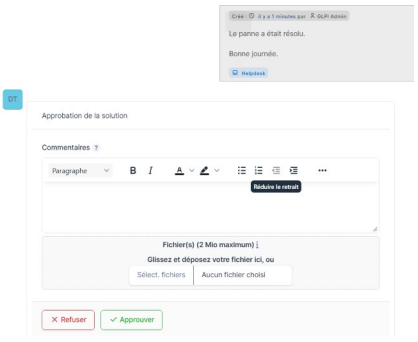
6. Valider la création du ticket.

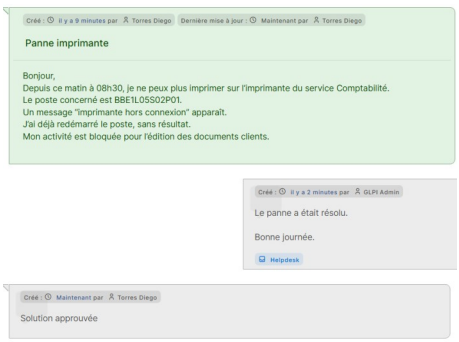


7. Suivre ensuite le ticket dans Mes tickets.



8. Approuver la résolution du ticket





Annexe 8. Installation et configuration de HAProxy

1. Installer HAProxy :
sudo apt update
sudo apt install -y haproxy
sudo haproxy -v

2. Sauvegarder la configuration d'origine :
sudo cp /etc/haproxy/haproxy.cfg /etc/haproxy/haproxy.cfg.bak

3. Modifier le fichier de configuration :
sudo nano /etc/haproxy/haproxy.cfg

Ajouter la configuration suivante :

```
frontend glpi_front
    bind *:80
    option forwardfor
    default_backend glpi_back

backend glpi_back
    balance roundrobin
    cookie SRV insert indirect nocache
    server glpi01 10.2.0.11:80 check cookie glpi01
    server glpi02 10.2.0.12:80 check cookie glpi02

listen stats
    bind 10.2.0.10:8080
    stats enable
    stats uri /stats
    stats auth admin:Admin123!
    stats refresh 30s
```

4. Vérifier la configuration :
sudo /usr/sbin/haproxy -c -V -f /etc/haproxy/haproxy.cfg

5. Redémarrer et activer le service :
sudo systemctl restart haproxy
sudo systemctl enable haproxy
sudo systemctl status haproxy

Annexe 9. Configuration de l'accès à la base de données pour SRV-GLPI-02

1. Se connecter à MariaDB sur SRV-GLPI-DB :
sudo mariadb -u root -p

2. Autoriser l'accès du second serveur applicatif à la base GLPI :

```
CREATE USER 'glpiadmin'@'10.2.0.12' IDENTIFIED BY 'Sio1234*';
GRANT ALL PRIVILEGES ON glpi.* TO 'glpiadmin'@'10.2.0.12';
FLUSH PRIVILEGES;
```

3. Autoriser l'accès en local à la base GLPI :

```
CREATE USER 'glpiadmin'@'localhost' IDENTIFIED BY 'Sio1234*';
GRANT ALL PRIVILEGES ON glpi.* TO 'glpiadmin'@'localhost';
FLUSH PRIVILEGES;
```

4. Vérifier la connexion depuis SRV-GLPI-02 :
mariadb -h 10.2.0.13 -u glpiadmin -p

5. Redémarrer HAProxy après mise à jour de l'architecture :
sudo systemctl restart haproxy

6. Éliminer les cookies du navigateur.

Annexe 10. Mise en place du HTTPS sur HAProxy

1. Installer les paquets nécessaires :
apt update
apt install haproxy openssl -y

2. Créer le répertoire des certificats :
mkdir /etc/ssl/glpi

3. Générer le certificat SSL :
openssl req -x509 -nodes -days 365 \
-newkey rsa:2048 \
-keyout /etc/ssl/glpi/glpi.key \
-out /etc/ssl/glpi/glpi.crt

Renseigner les valeurs suivantes :

- Country : FR
- State : Rhone-Alpes
- City : Lyon
- Organization : TiersLieux86
- Unit : Chasseneuilx86
- Common Name : glpi.chasseneuilx86.local
- Email : [vide]

4. Fusionner la clé privée et le certificat au format PEM :
sudo cat /etc/ssl/glpi/glpi.key /etc/ssl/glpi/glpi.crt | sudo tee /etc/ssl/glpi/glpi.pem > /dev/null

5. Modifier la configuration de HAProxy :
sudo nano /etc/haproxy/haproxy.cfg

Ajouter le frontend HTTPS :
```
frontend glpi_https
    bind *:443 ssl crt /etc/ssl/glpi/glpi.pem
    option forwardfor
    http-request set-header X-Forwarded-Proto https
    http-request set-header X-Forwarded-Port 443
    default_backend glpi_back
```

Modifier le frontend HTTP :
```
frontend glpi_http
   bind *:80
   redirect scheme https code 301
```

6. Redémarrer HAProxy :
sudo systemctl restart haproxy

Annexe 11. Création du dataset TrueNAS et montage du partage NFS

1. Créer le dataset sur TrueNAS.
Chemin d'accès : Storage > Pools > backupglpi > Add Dataset

Renseigner :
- Nom : backup-glpi

2. Créer le partage NFS.
Chemin d'accès : Shares > Unix Shares (NFS) > Add

Renseigner :
- Chemin : /mnt/backupglpi/backup-glpi
- Description : Sauvegarde GLPI
- Maproot User : root
- Maproot Group : root

Puis activer le partage.

3. Installer le support NFS sur SRV-GLPI-01 et SRV-GLPI-DB :
sudo apt update
sudo apt install nfs-common -y

4. Créer le point de montage et monter le partage :
sudo mkdir -p /mnt/truenas
sudo mount -t nfs 10.2.0.3:/mnt/backupglpi/backup-glpi /mnt/truenas

5. Vérifier le montage :
df -h | grep truenas
ls -l /mnt/truenas

6. Créer les répertoires de sauvegarde :
Sur SRV-GLPI-DB :
sudo mkdir -p /mnt/truenas/db

Sur SRV-GLPI-01 :
sudo mkdir -p /mnt/truenas/files

7. Rendre le montage persistant sur les deux serveurs.
Ajouter dans /etc/fstab :
10.2.0.3:/mnt/backupglpi/backup-glpi /mnt/truenas nfs defaults,_netdev 0 0

8. Tester le montage automatique :
sudo mount -a

Annexe 12. Scripts de sauvegarde GLPI et automatisation

1. Créer le répertoire des scripts :
sudo mkdir -p /opt/scripts

2. Créer le script de sauvegarde de la base de données sur SRV-GLPI-DB :
sudo nano /opt/scripts/backup_glpi_db.sh

Renseigner :
#!/bin/bash

DATE=$(date +%F)
DEST="/mnt/truenas/db"
DB_NAME="glpi"
DB_USER="glpiadmin"
DB_PASS="Sio1234*2A"

mkdir -p $DEST
mysqldump -u $DB_USER -p$DB_PASS $DB_NAME > $DEST/glpi_$DATE.sql
find $DEST -type f -name "*.sql" -mtime +7 -delete

Rendre le script exécutable :
sudo chmod +x /opt/scripts/backup_glpi_db.sh

Tester manuellement :
sudo /opt/scripts/backup_glpi_db.sh
ls -l /mnt/truenas/db

Résultat attendu : présence d'un fichier de type glpi_2026-03-30.sql

3. Créer le script de sauvegarde des fichiers GLPI sur SRV-GLPI-01 :
sudo nano /opt/scripts/backup_glpi_files.sh

Renseigner :
#!/bin/bash

DATE=$(date +%F)
DEST="/mnt/truenas/files"
mkdir -p $DEST
tar -czf $DEST/glpi01_$DATE.tar.gz /var/www/html/glpi
find $DEST -type f -name "glpi01_*.tar.gz" -mtime +7 -delete

Rendre le script exécutable :
sudo chmod +x /opt/scripts/backup_glpi_files.sh

Tester manuellement :
sudo /opt/scripts/backup_glpi_files.sh
ls -l /mnt/truenas/files

Résultat attendu : présence d'un fichier de type glpi01_2026-03-30.tar.gz

4. Planifier l'exécution automatique des sauvegardes.
Sur SRV-GLPI-DB :
sudo crontab -e
Ajouter :
0 2 * * * /opt/scripts/backup_glpi_db.sh

Sur SRV-GLPI-01 :
sudo crontab -e
Ajouter :
10 2 * * * /opt/scripts/backup_glpi_files.sh

Annexe 13. Procédure de restauration de GLPI

1. Restaurer la base de données sur SRV-GLPI-DB :
mysql -u glpiadmin -p glpi < /mnt/truenas/db/glpi_2026-03-30.sql

Vérifier ensuite que GLPI peut se reconnecter à la base.

2. Restaurer les fichiers applicatifs sur SRV-GLPI-01 :
sudo tar -xzf /mnt/truenas/files/glpi01_2026-03-30.tar.gz -C /

3. Redémarrer Apache :
sudo systemctl restart apache2