

MILLOT Elisa

RAPPORT TECHNIQUE SAE5.01 : **Concevoir, réaliser et présenter une solution technique**

Sommaire :

1. Introduction	3
2. Contexte	4
3. Objectifs	5
5. Mise en place de l'architecture Docker et explication du fichier docker-compose.yml.....	8
5.1 Structure du fichier docker-compose.yml	8
5.2 Méthodes de démarrage de l'environnement Docker	10
6. Mise en place de GLPI	12
6.1 Accès à l'interface d'installation	12
6.2 Étapes de l'installation	12
6.3 Accès à l'interface GLPI	14
7. Gestion des utilisateurs, des notifications et de l'inventaire	16
7.1 Création des utilisateurs, gestion des entités et des adresses mail	16
7.2 Mise en place des notifications par mail	19
7.3 Mise en place de l'inventaire avec GLPI Agent	23
7.4 Tentative d'automatisation de l'inventaire via script	27
8. Évolutions et perspectives du projet	30
8.1 Accessibilité et portail client	30
8.2 Automatisation du déploiement de GLPI Agent	30
8.3 Migration de la base GLPI existante vers l'architecture Docker	31
8.4 Renforcement de la sécurité et supervision avancée	31
9. Mise en lien avec les apprentissages critiques du BUT R&T	32
11. Conclusion	34

1. Introduction

Dans le cadre de ce projet technique, j'ai réalisé la mise en place d'une solution GLPI basée sur une architecture conteneurisée à l'aide de Docker. Ce projet s'inscrit dans une démarche d'analyse, de test et d'amélioration d'un outil déjà utilisé en entreprise, avec pour objectif d'explorer des alternatives techniques plus modernes et plus flexibles que l'existant.

GLPI (Gestionnaire Libre de Parc Informatique) est un outil central dans la gestion du support informatique, aussi bien pour le suivi des incidents que pour la gestion des demandes utilisateurs et l'inventaire du parc informatique. Bien que fonctionnel dans sa version actuelle en entreprise, son mode de déploiement et son utilisation présentent certaines limites, notamment en matière d'accessibilité pour les clients et de maintenance de l'infrastructure.

Ce projet a donc été réalisé dans un environnement personnel afin de pouvoir tester librement une version plus récente de GLPI, sans impacter l'outil en production. Il permet également de mieux comprendre les mécanismes internes de GLPI, les possibilités offertes par Docker, ainsi que les contraintes liées à une éventuelle mise en production future.

2. Contexte

Au sein de l'entreprise, GLPI est utilisé comme outil de gestion de tickets et de support informatique. Il sert principalement à gérer les incidents et les demandes des clients professionnels, ainsi qu'à assurer le suivi des interventions réalisées par les techniciens. L'outil est utilisé exclusivement en interne.

Actuellement, GLPI est hébergé sur une machine virtuelle fonctionnant sous Ubuntu LTS. Cette machine virtuelle est déployée sur un serveur interne de l'entreprise. L'accès à GLPI est réservé aux techniciens, ce qui signifie que les clients ne peuvent pas créer eux-mêmes leurs tickets via une interface web.

Lorsqu'un client rencontre un problème, il doit contacter l'entreprise par téléphone ou par mail. Les techniciens doivent ensuite créer manuellement les tickets dans GLPI. Cette organisation peut entraîner des interruptions fréquentes, une perte de temps et une charge supplémentaire pour les équipes techniques, notamment lorsque les sollicitations sont nombreuses.

De plus, l'utilisation d'une machine virtuelle dédiée implique une infrastructure relativement lourde à maintenir. Elle nécessite la gestion du système d'exploitation, des mises à jour de sécurité, du serveur web, de PHP et de la base de données. Toute évolution de GLPI, comme une montée de version, peut donc devenir complexe et risquée.

Dans ce contexte, l'idée de tester une solution alternative basée sur Docker est apparue pertinente. La conteneurisation permet de déployer une application de manière plus légère, plus modulable et plus simple à maintenir. Le projet a donc été réalisé hors environnement de production, dans un cadre de test.

3. Objectifs

Le premier objectif de ce projet était de mettre en place une instance GLPI fonctionnelle à l'aide de Docker, en remplaçant le modèle basé sur une machine virtuelle par une architecture conteneurisée. Cette approche devait permettre de réduire la charge liée à l'infrastructure tout en facilitant le déploiement et la maintenance de l'outil.

Un second objectif était de tester une version plus récente de GLPI que celle actuellement utilisée en entreprise. Cela permet d'anticiper une éventuelle mise à jour future et d'identifier les différences de fonctionnement ou de configuration entre les versions.

Le projet visait également à approfondir la compréhension de GLPI, notamment en ce qui concerne la gestion des utilisateurs, des profils, des entités et des droits. Une attention particulière a été portée à la distinction entre les profils techniciens et les profils clients, dans une logique d'ouverture future de l'outil aux clients professionnels.

Enfin, un objectif important était de tester des fonctionnalités essentielles dans un contexte professionnel, telles que les notifications par mail et l'inventaire automatique via GLPI Agent. Ces éléments sont indispensables pour proposer une solution complète et réellement exploitable à long terme.

4. Organisation et planification du projet (Gantt)

Afin de structurer le travail et de respecter une progression logique, le projet a été organisé à l'aide d'un diagramme de Gantt. Ce dernier permet de visualiser les différentes phases du projet ainsi que le temps consacré à chacune d'elles, exprimé en heures et réparti sur plusieurs semaines.

Deux versions du Gantt ont été réalisées :

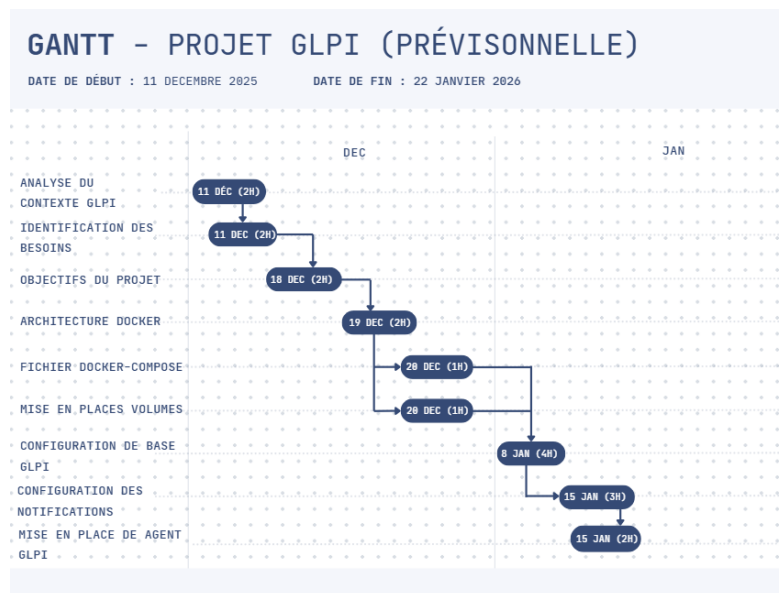


Image 1 : le diagramme de Gantt prévisionnel, établi en amont du projet pour planifier les tâches à réaliser.

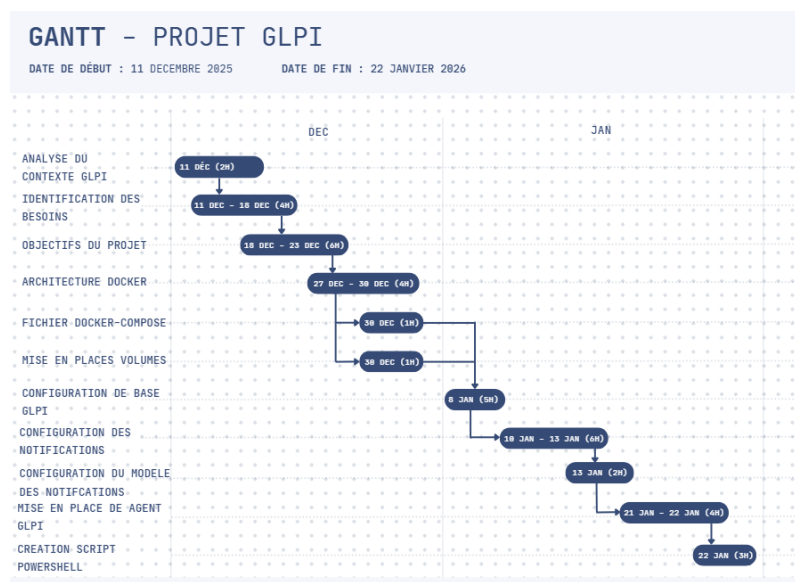


Image 2 : le diagramme de Gantt réel, construit a posteriori pour refléter le déroulement effectif du projet.

Le Gantt prévisionnel présente une organisation linéaire et optimisée, avec des durées estimées réduites pour certaines tâches afin de tenir dans un volume horaire hebdomadaire réaliste (environ 5 à 6 heures par semaine). Il s'appuie sur une logique progressive : analyse du contexte GLPI, identification des besoins, définition des objectifs, puis conception de l'architecture Docker et mise en place technique.

Le Gantt réel, quant à lui, illustre les ajustements nécessaires au fil du projet. Certaines tâches ont été plus longues que prévu, notamment la configuration des notifications et les tests liés à GLPI-Agent. D'autres ont été décalées ou regroupées en fonction des contraintes techniques rencontrées. La création du script PowerShell, bien qu'entamée, n'a pas pu être finalisée dans le temps imparti.

La comparaison entre les deux diagrammes met en évidence les écarts entre planification théorique et réalité terrain, tout en soulignant la capacité d'adaptation et de réorganisation face aux imprévus techniques.

5. Mise en place de l'architecture Docker et explication du fichier docker-compose.yml

La solution technique repose sur l'utilisation de **Docker Compose** pour déployer et gérer l'ensemble de l'infrastructure GLPI. Ce choix permet de décrire l'architecture dans un **fichier unique** (docker-compose.yml), facilitant la compréhension, la maintenance et la reproductibilité de l'environnement.

L'architecture est composée de **deux services principaux** :

- **Le service de base de données** (MariaDB)
- **Le service applicatif GLPI** (interface web)

Cette séparation respecte les bonnes pratiques en matière de conteneurisation, en assurant une **isolation fonctionnelle** entre les composants.

5.1 Structure du fichier docker-compose.yml

Le fichier est rédigé en YAML et contient les définitions suivantes :

- **Service db :**
 - Utilise l'image mariadb:10.11
 - Définit les variables d'environnement nécessaires à la création automatique de la base (MYSQL_DATABASE, MYSQL_USER, MYSQL_PASSWORD, MYSQL_ROOT_PASSWORD)
 - Active le plugin mysql_native_password pour assurer la compatibilité avec GLPI
 - Monte un volume persistant db_data pour conserver les données


```
docker-compose.yml X
SAES01 > docker-compose.yml
1 # Version du format docker-compose. Peut évoluer selon les besoins ou les nouveautés Docker.
2 # Pour vérifier la compatibilité, consulter la documentation officielle :
3 # https://docs.docker.com/compose/compose-file/compose-versioning/
4 version: "3.8"
5
6
7 Run All Services
8 services:
9   # Service de base de données MariaDB
10  db:
11    # Image utilisée. La version (10.11) peut être mise à jour si besoin de correctifs ou nouvelles fonctionnalités.
12    image: mariadb:10.11
13    container_name: glpi_db
14    restart: always # Redémarre automatiquement le conteneur en cas de crash
15    environment:
16      # ATTENTION : Les mots de passe sont en clair ici. À sécuriser pour la production !
17      MYSQL_ROOT_PASSWORD: rootpass # Mot de passe root de la base (à changer en prod)
18      MYSQL_DATABASE: glpi # Nom de la base créée au démarrage
19      MYSQL_USER: glpi # Utilisateur par défaut
20      MYSQL_PASSWORD: glpipass # Mot de passe utilisateur (à changer en prod)
21      # Option nécessaire pour compatibilité avec certains clients (GLPI, etc.)
22      command: --default-authentication-plugin=mysql_native_password
23    volumes:
24      # Stockage persistant des données de la base
25      - db_data:/var/lib/mysql
```

Image 3 : Extrait du fichier docker-compose.yml — service MariaDB

- **Service glpi :**
 - Utilise l'image communautaire diouxx/glpi
 - Expose le port 80 du conteneur sur le port 8080 de la machine hôte (8080:80)
 - Configure le fuseau horaire (TZ: Europe/Paris)
 - Monte un volume glpi_data pour les fichiers applicatifs
 - Dépend du service db (via depends_on) pour garantir l'ordre de démarrage
- **Volumes persistants :**
 - db_data pour MariaDB
 - glpi_data pour GLPI

```
25
26 # Service applicatif GLPI
27 >Run Service
28 glpi:
29   # Image officielle communautaire. Peut évoluer, vérifier les mises à jour sur Docker Hub.
30   image: diouxx/glpi
31   container_name: glpi_app
32   restart: always
33   ports:
34     - "8080:80" # Expose le port 80 du conteneur sur le port 8080 de la machine hôte
35   environment:
36     TZ: Europe/Paris # Fuseau horaire (adapter si besoin)
37   volumes:
38     # Stockage persistant des fichiers GLPI (plugins, documents, etc.)
39     - glpi_data:/var/www/html
40   depends_on:
41     - db # Démarre le service db avant glpi
42
43 # Définition des volumes persistants (données conservées même si les conteneurs sont supprimés)
44 volumes:
45   db_data:
46   glpi_data:
47
```

Image 4 : Extrait du fichier docker-compose.yml — service GLPI

Cette configuration garantit que les données sont conservées même en cas de redémarrage ou suppression des conteneurs.

5.2 Méthodes de démarrage de l'environnement Docker

Une fois le fichier docker-compose.yml rédigé et placé dans le répertoire du projet, deux méthodes permettent de lancer l'environnement :

Méthode 1 — Via le terminal

1. Ouvrir un terminal dans le dossier contenant le fichier docker-compose.yml
2. Exécuter la commande suivante :

docker compose up -d

- L'option -d permet de lancer les conteneurs en arrière-plan
- Les services glpi et db sont automatiquement créés, configurés et démarrés

Méthode 2 — Via Docker Desktop

1. Ouvrir Docker Desktop
2. Naviguer dans l'onglet **Containers & Apps**

3. Repérer le projet (nom du dossier contenant le docker-compose.yml)
4. Cliquer sur la **flèche de démarrage** à droite du projet

☐	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Start ns
☐	sae501	-	-	-	N/A	18 seconds ago	  
☐	glpi_app	5ae00f08c6b8	dioux/glpi	8080:80	N/A	18 seconds ago	  
☐	glpi_db	18818982b538	mariadb:10.11		N/A	18 seconds ago	  

Image 5 : Vue Docker Desktop — conteneurs glpi.app et glpi_db en cours d'exécution

Cette méthode visuelle permet de suivre l'état des conteneurs, consulter les logs, et redémarrer les services si nécessaire.

6. Mise en place de GLPI

Une fois l'environnement Docker opérationnel et les conteneurs lancés, l'installation de GLPI peut commencer via l'interface web accessible à l'adresse `http://localhost:8080`. Cette interface propose un processus d'installation guidé, étape par étape, permettant de configurer l'application et de la connecter à la base de données MariaDB définie dans le fichier `docker-compose.yml`.

6.1 Accès à l'interface d'installation

Dès que le conteneur GLPI est actif, une première page s'affiche dans le navigateur. Celle-ci indique que la base de données n'est pas encore configurée et propose un bouton intitulé "Aller à la page d'installation". Cette étape confirme que le service GLPI est bien lancé et que l'environnement est prêt à recevoir la configuration initiale.

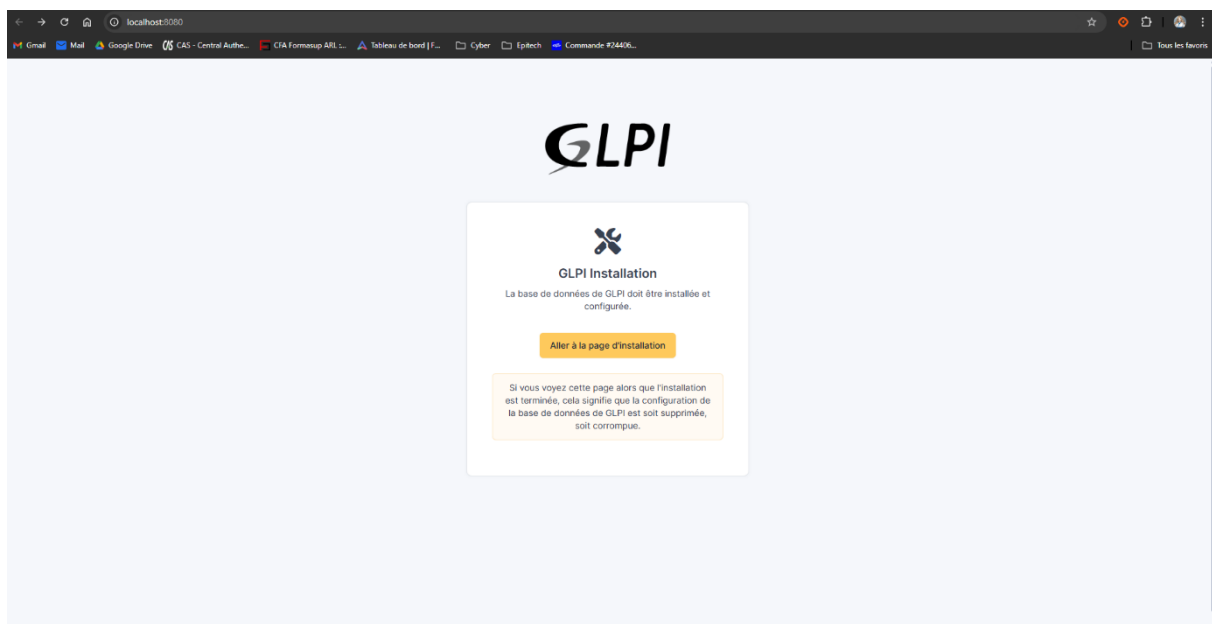


Image 6 : Page d'accueil de l'installation GLPI

6.2 Étapes de l'installation

L'installation se déroule en plusieurs phases successives :

- **Choix du mode d'installation** : L'utilisateur est invité à choisir entre une nouvelle installation ou une mise à jour d'une version existante. Dans le cadre de ce projet, une nouvelle installation est sélectionnée. Il est toutefois important de noter que GLPI permet également de **recupérer les données d'une autre instance GLPI**. Cette

fonctionnalité est utile en cas de migration : il suffit d'importer l'ancienne base de données et les fichiers associés pour retrouver l'intégralité des tickets, utilisateurs, configurations et historiques.



Image 7 : Choix entre installation et mise à jour

- **Connexion à la base de données** : GLPI demande les paramètres de connexion à la base MariaDB. Ces informations sont directement liées aux variables définies dans le fichier docker-compose.yml. Les champs à renseigner sont :
 - Serveur SQL : db (nom du service Docker)
 - Utilisateur SQL : glpi
 - Mot de passe SQL : glpipass Une fois ces données validées, GLPI initialise automatiquement les tables nécessaires au fonctionnement de l'application.



The screenshot shows the GLPI installation interface. At the top, the GLPI logo is on the left and 'GLPI Installation' is on the right. Below this, it says 'Étape 1' (Step 1) and 'Configuration de la connexion à la base de données' (Configuration of the database connection). There are three input fields: 'Serveur SQL (MariaDB ou MySQL)' with the value 'db', 'Utilisateur SQL' with the value 'glpi', and 'Mot de passe SQL' with masked characters. A yellow 'Continuer >' button is at the bottom.

Image 8 : Configuration de la base de données

- **Création des comptes par défaut** : À la fin du processus, GLPI affiche les identifiants standards pour les différents profils :



The screenshot shows the final step of the GLPI installation. It says 'Étape 6' (Step 6) and 'L'installation est terminée' (Installation is completed). Below this, it lists the default usernames and passwords: 'glpi/glpi' for administrator, 'tech/tech' for technician, 'normal/normal' for normal user, and 'post-only/postonly' for postonly user. A yellow button with a thumbs up icon and the text 'Utiliser GLPI' is at the bottom.

Image 9 : Confirmation de l'installation et identifiants par défaut

6.3 Accès à l'interface GLPI

Une fois l'installation terminée, l'utilisateur est redirigé vers le tableau de bord principal de GLPI. Cette interface centralise toutes les fonctionnalités de l'outil : gestion des tickets, des utilisateurs, des entités, des problèmes, des changements, du planning, et des statistiques. À ce stade, aucune donnée n'est encore enregistrée, ce qui est normal pour une installation fraîche.

L'interface est claire, structurée, et permet une personnalisation fine selon les rôles et les besoins. Elle constitue le point de départ pour la configuration fonctionnelle de l'outil (profils, droits, entités, notifications, etc.).

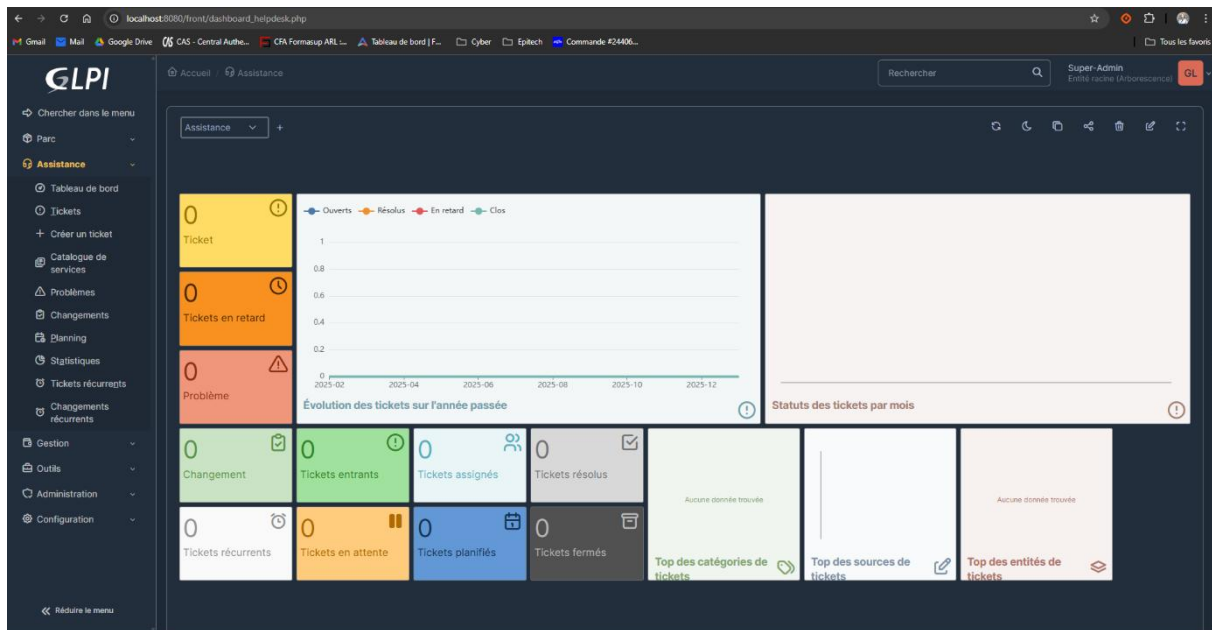


Image 10 : Tableau de bord GLPI après installation

7. Gestion des utilisateurs, des notifications et de l'inventaire

Une fois l'instance GLPI installée et fonctionnelle, une phase de configuration approfondie a été réalisée afin de reproduire un environnement proche de celui d'une utilisation professionnelle réelle. Cette étape est essentielle, car GLPI ne se limite pas à un simple outil de ticketing : sa pertinence repose en grande partie sur la gestion des utilisateurs, des droits, des notifications et de l'inventaire du parc informatique.

L'objectif de cette partie était de tester la capacité de GLPI à être utilisé non seulement par les techniciens, mais également par les clients, tout en assurant une communication automatique et un suivi précis des équipements.

7.1 Création des utilisateurs, gestion des entités et des adresses mail

La première action a consisté à créer différents comptes utilisateurs. Deux types de profils ont été mis en place :

- **Les comptes techniciens**, disposant de droits étendus leur permettant d'accéder à l'ensemble des fonctionnalités de GLPI. Ces utilisateurs peuvent créer, modifier, attribuer et clôturer des tickets, gérer les utilisateurs, consulter l'inventaire, configurer les entités et accéder aux paramètres avancés.
- **Les comptes clients**, nommés de Client1 à Client5, volontairement restreints en termes de droits. Chaque client dispose d'un accès limité à son propre espace, lui permettant de créer des tickets (demande ou incident), de suivre leur évolution, et de consulter uniquement les tickets qui lui sont associés. Cette restriction garantit la confidentialité des données et évite toute manipulation involontaire.

Chaque compte client est associé à une **adresse mail valide**, indispensable pour le bon fonctionnement du système de notifications. Ces adresses permettent à GLPI d'envoyer automatiquement des messages lors d'événements liés aux tickets (création, mise à jour, clôture, etc.).

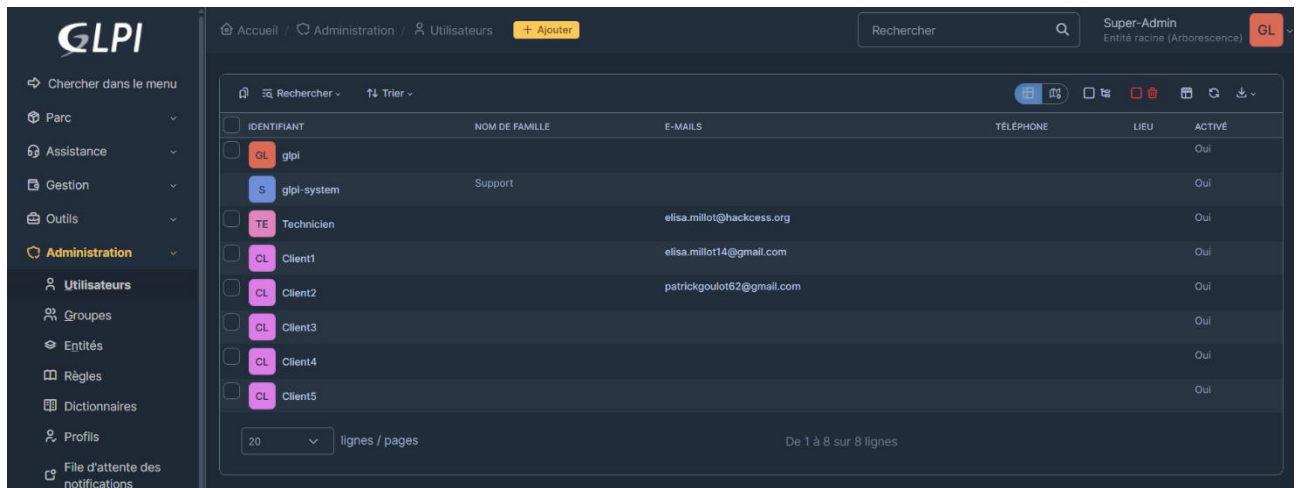


Image 11 : Création des différents utilisateurs

La configuration des **entités** a également été mise en place pour structurer l'environnement GLPI. Les entités permettent de segmenter les utilisateurs, les tickets et les ressources selon des périmètres définis. Dans un contexte multi-client, cette organisation est cruciale pour cloisonner les informations et limiter la visibilité des tickets à leur propriétaire ou à leur entité d'appartenance. Dans les paramètres de chaque entiers (ou sur l'entité racine) ne pas oubliée de configurée les notifications sur Oui et un délai de 0 minutes pour la suite.

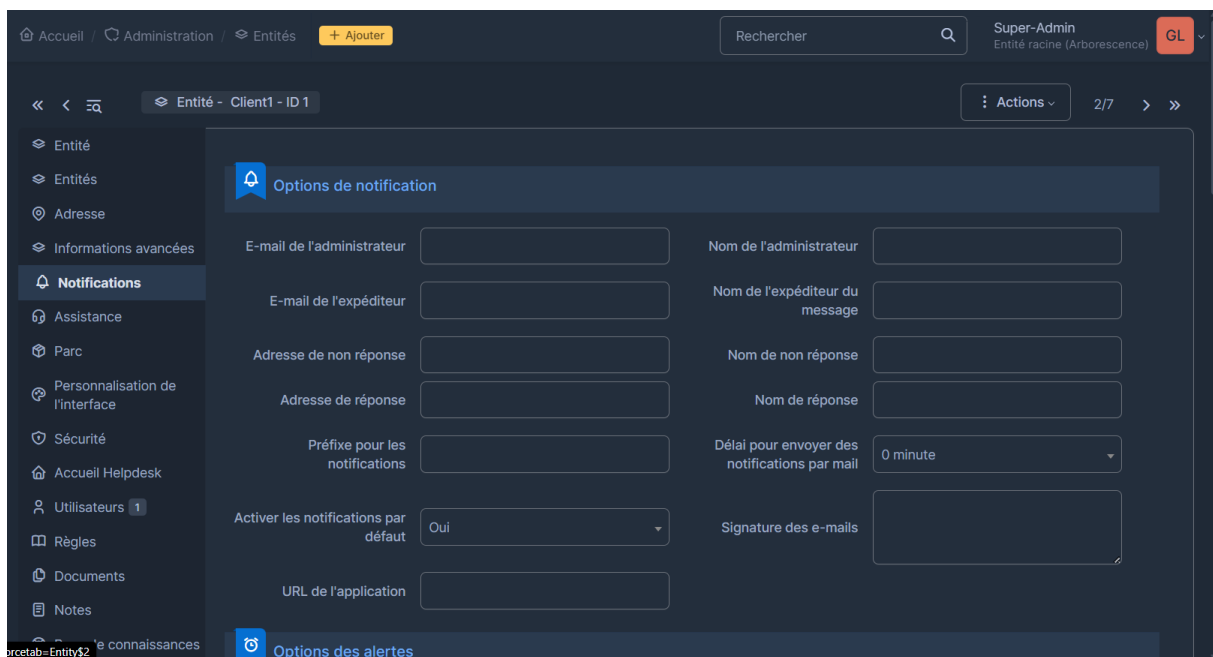


Image 12 : Création des différents entités clients avec la configuration

Une attention particulière a été portée à l'interface utilisateur afin d'adapter l'expérience selon le rôle de chaque profil. Les clients disposent d'un **affichage simplifié**, centré sur la création et le suivi de leurs propres tickets. Cette interface est activée directement dans la configuration du **profil utilisateur**, en sélectionnant le mode "Interface simplifiée". Ce paramètre permet de masquer les menus techniques et d'alléger l'environnement visuel pour les utilisateurs non-administrateurs.

Une fois le profil créé, les **droits d'accès** sont ensuite ajustés dans les paramètres du profil. Il est possible de définir précisément ce que chaque utilisateur peut voir ou faire : création de ticket, consultation, suivi, accès aux documents, visibilité sur les entités, etc. Cette granularité permet de garantir que les clients ne peuvent consulter que leurs propres demandes, tout en empêchant toute modification non autorisée.

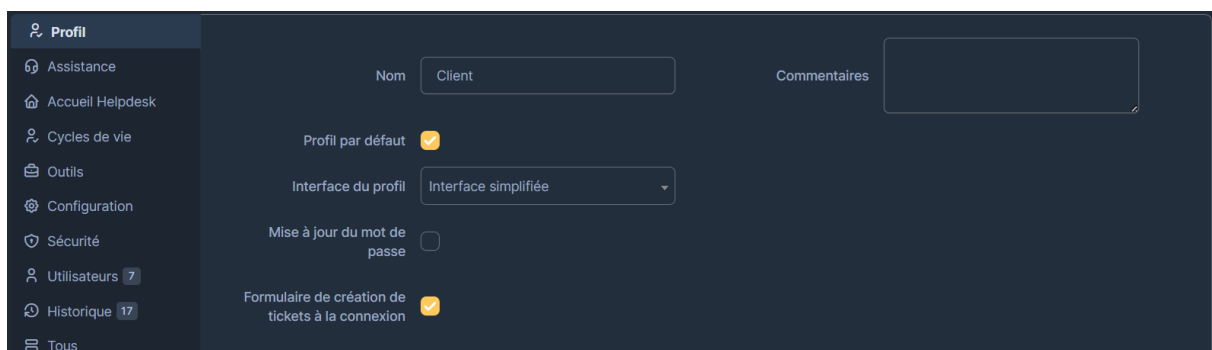


Image 13 : Création du profil pour les utilisateurs & configuration

Les techniciens, quant à eux, sont associés à un **profil complet**, avec une interface étendue leur donnant accès à l'ensemble des fonctionnalités de GLPI. Ils peuvent ainsi intervenir efficacement sur les tickets, gérer les utilisateurs, consulter l'inventaire, configurer les entités et piloter l'environnement global.

Cette configuration initiale permet de valider que GLPI peut être utilisé dans un cadre multi-utilisateur, avec des rôles différenciés, une gestion fine des droits, et une organisation structurée par entité. Elle constitue la base fonctionnelle sur laquelle les autres modules (notifications, inventaire, automatisation) viendront s'appuyer.

7.2 Mise en place des notifications par mail

La mise en place des notifications par mail constitue une étape essentielle dans la configuration de GLPI, notamment dans un contexte de support technique multi-utilisateur. Cette fonctionnalité permet d'automatiser la communication entre les clients et les techniciens, en assurant un suivi en temps réel de l'évolution des tickets.

Configuration générale des notifications

La première étape consiste à activer le système de notifications dans GLPI. Cette activation se fait depuis le menu **Configuration > Notifications**, où plusieurs options sont disponibles :

- Activer les notifications globales
- Activer les notifications par mail
- Activer les notifications navigateur (optionnelle)

Une fois ces options activées, GLPI est en mesure de générer des messages à chaque événement lié aux tickets (création, mise à jour, clôture, etc.).

Paramétrage du serveur SMTP

Pour permettre l'envoi des mails, un serveur SMTP externe a été configuré. Dans ce projet, le service **Titan Email** a été utilisé. Les paramètres renseignés dans GLPI sont les suivants :

- Hôte SMTP : smtp.titan.email
- Port : 465 (port sécurisé avec SSL)
- Adresse d'expédition : elisa.millot@hackcess.org
- Mode d'envoi : SMTP
- Vérification du certificat : activée
- Nombre de tentatives : 5
- Intervalle entre les tentatives : 5 minutes

The screenshot shows the GLPI interface for configuring notifications. The left sidebar contains a menu with options like 'Parc', 'Assistance', 'Gestion', 'Outils', 'Administration', 'Configuration', 'Actifs personnalisés', 'Intitulés', 'Composants', 'Notifications', 'Webhooks', 'Niveaux de services', 'Générale', 'Unicité des champs', and 'Actions automatiques'. The main content area is titled 'Configuration' and includes a 'Historique' tab. The 'Configuration' section has fields for 'E-mail de l'administrateur', 'E-mail de l'expéditeur', 'Adresse de réponse', 'Adresse de non réponse', 'Ajouter des documents dans les notifications de ticket', 'Signature des e-mails', 'Mode d'envoi des e-mails', and 'Tentatives d'envoi max.'. The 'SERVEUR MAILS' section includes 'Vérifier le certificat', 'Hôte SMTP', 'Port', 'Identifiant SMTP (optionnel)', 'Mot de passe SMTP (optionnel)', and 'Expéditeur du message'. A 'Sauvegarder' button is at the bottom right.

Image 14 : Configuration des notifications

Ces paramètres sont accessibles dans **Configuration > Notifications > Serveur de mails**. Une fois la configuration enregistrée, un **mail de test** peut être envoyé à l'administrateur pour valider la connexion au serveur.

[GLPI] Mail de test

The screenshot shows a test email received in a GLPI interface. The email is from 'me <elisa.millot@hackcess.org>' and contains the text 'Ceci est le mail de test.' followed by a signature line. The interface shows standard email actions like 'Répondre', 'Répondre à tous', 'Transférer', and 'Répondre avec l'IA'.

Image 15 : Mail de test à l'administrateur

Déclencheurs et modèles de notification

GLPI permet de définir précisément les événements qui déclenchent l'envoi d'un mail. Dans ce projet, les notifications ont été activées pour les événements suivants :

- Nouveau ticket
- Mise à jour d'un ticket
- Clôture d'un ticket
- Ajout d'un suivi
- Ajout d'une tâche
- Résolution du ticket

Actions						
☐	NOM	ENTITÉ	TYPE	ACTIVÉ [▲]	ÉVÉNEMENTS	MODÈLES DE NOTIFICATIONS
☒	New Ticket	Entité racine	Ticket	Oui	Nouveau ticket	Tickets
☒	Update Ticket	Entité racine	Ticket	Oui	Mise à jour d'un ticket	Tickets
☒	Close Ticket	Entité racine	Ticket	Oui	Clôture du ticket	Tickets
☒	Add Followup	Entité racine	Ticket	Oui	Nouveau suivi	Tickets
☒	Add Task	Entité racine	Ticket	Oui	Nouvelle tâche	Tickets
☒	Resolve ticket	Entité racine	Ticket	Oui	Ticket résolu	Tickets
☐	Alert Tickets not closed	Entité racine	Ticket	Non	Tickets non résolus	Alert Tickets not closed

Image 16 : Activations des notifications que nous voulons envoyer

Chaque événement est associé à un **modèle de notification** personnalisable. Ces modèles définissent le contenu du mail, les variables dynamiques (nom du ticket, auteur, date, etc.), et les destinataires. Les modèles ont été adaptés pour garantir une **lisibilité optimale** des messages reçus par les clients.

Accueil | Configuration | Notifications | Modèles de not... [+ Ajouter](#)

Rechercher

Super-Admin
Entité racine (Arborescence) GL

Traduction de modèle - Traduction par défaut - ID 4

Traduction de modèle

Prévisualisation

Historique 22

Tous

Modèle de notifications Tickets

Voir la liste des balises disponibles

Langue Traduction par défaut

Sujet ##ticket.action## ##ticket.title##

Corps texte du mail
(laisser le champ vide pour qu'il soit généré depuis le champ HTML)

##ticket.action## concernant votre ticket

ID : ##ticket.id##
Titre : ##ticket.title##
Catégorie : ##ticket.category##
Demandeur : ##ticket.authors##
Date d'ouverture : ##ticket.creationdate##
Statut : ##ticket.status##
Urgence : ##ticket.urgency##
Impact : ##ticket.impact##
Priorité : ##ticket.priority##

--- Contenu du ticket ---
##ticket.content##

Image 17 : Configuration du modèle pour les tickets

Résolution d'un blocage dans la file d'attente

Lors des premiers tests, un dysfonctionnement a été observé : les mails restaient bloqués dans la **file d'attente** et n'étaient pas transmis immédiatement. Après analyse, il est apparu que le problème ne venait ni du serveur SMTP ni d'un défaut de configuration réseau, mais d'un paramètre interne de GLPI : les **actions automatiques**.

La tâche "Notification" n'était pas exécutée régulièrement. Pour corriger cela, une modification a été apportée dans **Configuration > Actions automatiques**, en réglant l'exécution de la tâche "Notification" toutes les minutes. Cette correction a permis de débloquent la file d'attente et de rendre l'envoi des mails pleinement fonctionnel.

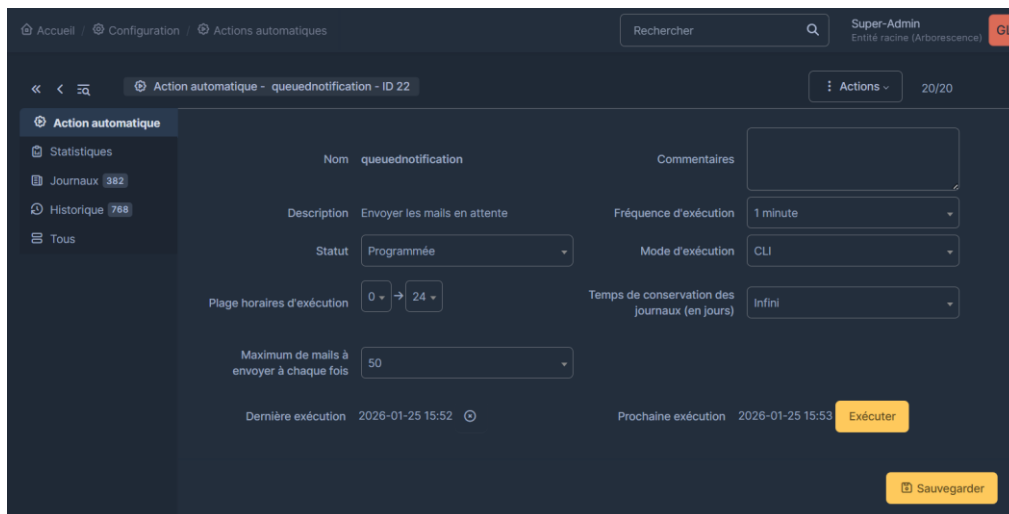


Image 18 : Configuration du Actions Automatiques afin que les mails s'envoient

Validation du système

Une série de tests a été réalisée pour valider le bon fonctionnement du système de notification. Des tickets ont été créés, mis à jour, suivis et clôturés, avec des comptes clients et techniciens. À chaque étape, un mail a été envoyé automatiquement à l'utilisateur concerné, contenant un lien direct vers le ticket dans l'interface GLPI.

☐	☆	▶	elisa.millot 2	[GLPI #0000009] Nouveau suivi Test mail 8 - Client 1 - URL : http://localhost:8080/index.php?redirect=ticket_9 Ticket : Des...	10:02
☐	☆	▶	elisa.millot	[GLPI #0000009] Clôture du ticket Test mail 8 - Client 1 - URL : http://localhost:8080/index.php?redirect=ticket_9 Ticket : ...	10:02
☐	☆	▶	elisa.millot	[GLPI #0000009] Ticket résolu Test mail 8 - Client 1 - URL : http://localhost:8080/index.php?redirect=ticket_9 Ticket%241 ...	10:02
☐	☆	▶	elisa.millot	[GLPI #0000009] Nouvelle tâche Test mail 8 - Client 1 - URL : http://localhost:8080/index.php?redirect=ticket_9 Ticket : De...	10:00
☐	☆	▶	elisa.millot	[GLPI #0000009] Nouveau ticket Test mail 8 - Client 1 - URL : http://localhost:8080/index.php?redirect=ticket_9 Ticket : De...	09:58

Image 19 : Mails reçu à la suite de la modifications tickets pour le client 1 (elisa.millot14@gmail.com)

Cette configuration confirme que GLPI est capable de gérer les notifications de manière fiable, à condition que les paramètres SMTP, les modèles de notification et les actions automatiques soient correctement définis.

7.3 Mise en place de l'inventaire avec GLPI Agent

L'inventaire automatique constitue l'un des modules les plus importants de GLPI, car il permet de centraliser l'ensemble des informations matérielles et logicielles des postes clients. Cette fonctionnalité est indispensable dans un environnement professionnel, où la gestion du parc informatique repose sur une vision précise et actualisée des équipements.

Pour activer cette fonctionnalité, l'outil **GLPI Agent** a été utilisé. Il s'agit du nouvel agent officiel du projet GLPI, remplaçant FusionInventory et OCS Inventory. Il est capable de remonter automatiquement les informations des postes vers le serveur GLPI, sans intervention manuelle.

Activation de l'inventaire dans GLPI

Avant d'installer l'agent sur un poste, il est nécessaire d'activer la fonctionnalité d'inventaire dans GLPI. Dans le menu **Administration > Inventaire**, l'option *Activer l'inventaire* a été cochée, ce qui permet à GLPI d'accepter les données envoyées par les agents.

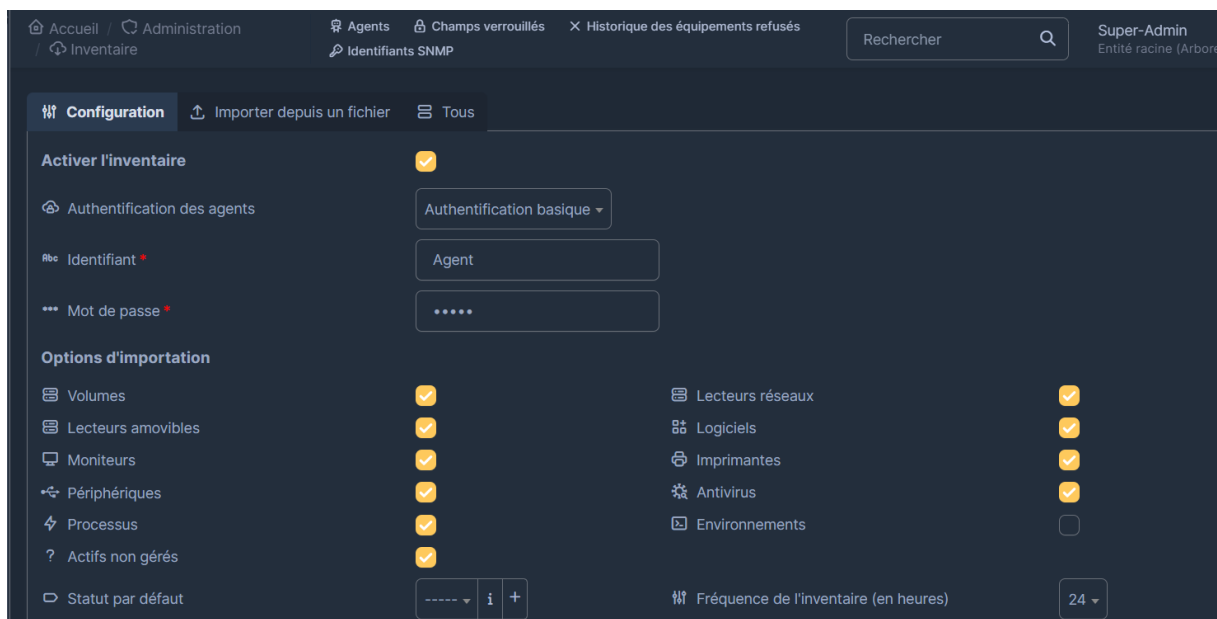


Image 20 : Activation de l'inventaire

Une fois cette option activée, GLPI affiche un message confirmant que la configuration a été mise à jour. Cette étape est indispensable : sans activation préalable, les agents ne peuvent pas envoyer leurs données au serveur.

Installation de GLPI Agent sur un poste client

L'agent a été installé sur un poste Windows à l'aide de l'installateur officiel (GLPI Agent 1.15).

L'assistant d'installation propose plusieurs étapes successives :

- choix du type d'installation (typique, personnalisée ou complète)

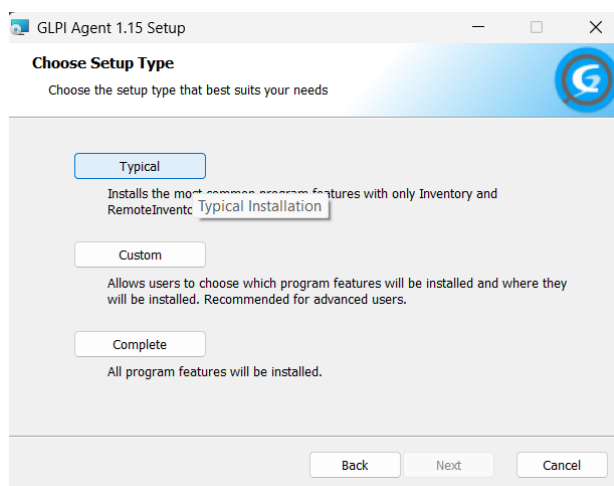


Image 21 : Etape 1 installation Agent-GLPI

- configuration de la destination locale

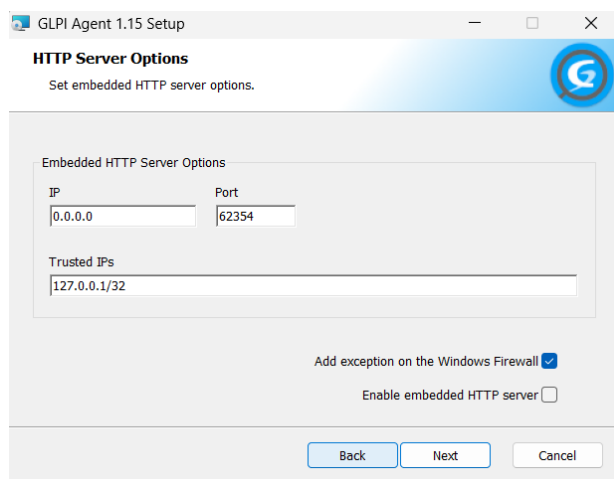


Image 22 : Etape 2 installation Agent-GLPI

- définition de la ou des cibles distantes (URL du serveur GLPI)

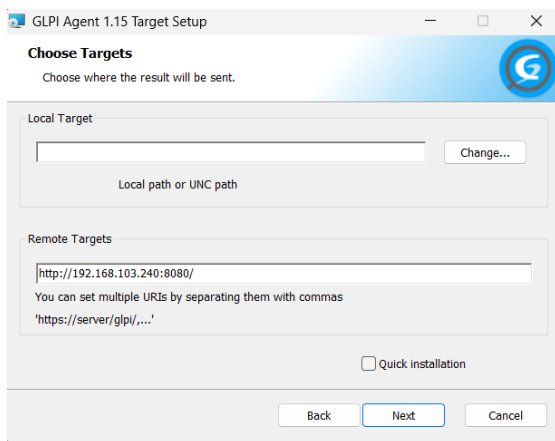


Image 23 : Etape 3 installation Agent-GLPI

- configuration éventuelle d'un proxy ou d'une authentification

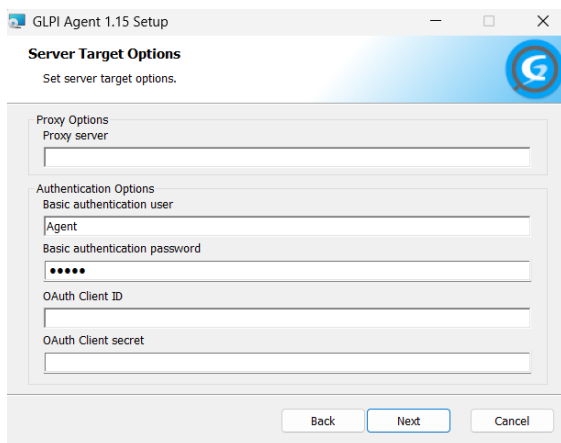


Image 24 : Etape 4 installation Agent-GLPI

- choix du mode d'exécution (service Windows, tâche planifiée ou manuel)

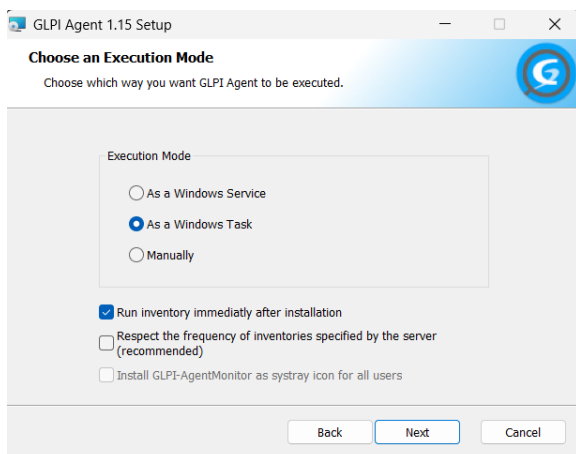


Image 25 : Etape 5 installation Agent-GLPI

- configuration de la fréquence d'inventaire

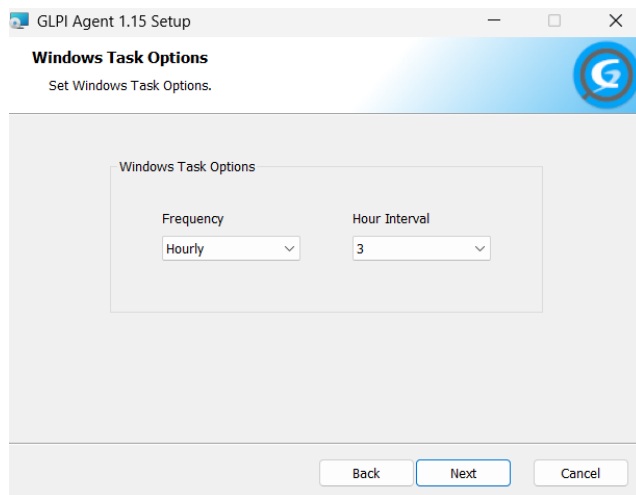


Image 26 : Etape 6 installation Agent-GLPI

- options avancées (serveur HTTP embarqué, logs, niveau de debug)

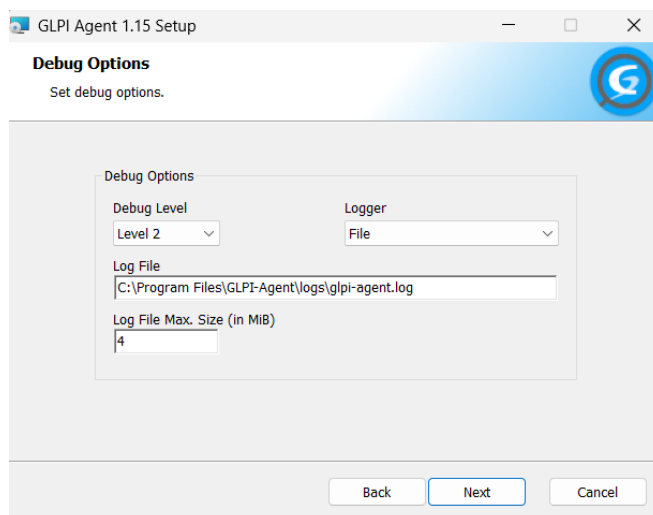


Image 27 : Etape 7 installation Agent-GLPI

Lors de la configuration, l'adresse du serveur GLPI a été renseignée dans le champ *Remote Targets*, ce qui permet à l'agent de savoir où envoyer les inventaires. L'option *Run inventory immediately after installation* a été activée afin de forcer un premier envoi dès la fin de l'installation.

Remontée des informations dans GLPI

Une fois l'installation terminée, l'agent a automatiquement envoyé un inventaire complet au serveur GLPI. Les informations suivantes ont été correctement remontées :

- modèle et fabricant du poste
- numéro de série
- processeur, mémoire, stockage
- système d'exploitation
- logiciels installés
- interfaces réseau
- informations diverses (antivirus, périphériques, etc.)

Le poste apparaît alors dans **Parc > Ordinateurs**, avec l'ensemble des données collectées.

Cette étape confirme le bon fonctionnement de l'inventaire automatique et la capacité de GLPI à centraliser les informations du parc informatique.

Cette fonctionnalité est essentielle dans une optique de gestion à grande échelle, car elle permet d'obtenir une vision complète et actualisée des équipements sans intervention manuelle.

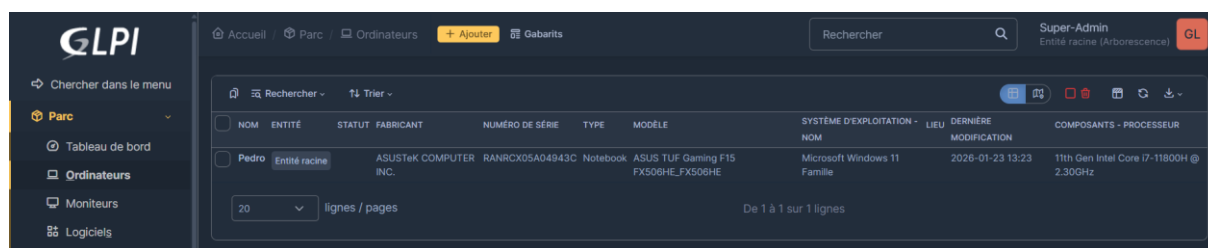


Image 28 : Remonter de donnée de l'Agent-GLPI

7.4 Tentative d'automatisation de l'inventaire via script

Dans une démarche d'optimisation et de déploiement massif, une tentative d'automatisation de l'installation de GLPI Agent a été réalisée à l'aide d'un script PowerShell. L'objectif était de permettre l'installation de l'agent sur plusieurs postes sans intervention humaine, en automatisant :

- le téléchargement de l'installateur
- l'exécution silencieuse
- la configuration de l'URL du serveur GLPI
- la création du mode d'exécution
- le lancement immédiat d'un inventaire

```
C: > Users > elisa > OneDrive > Bureau > Install-GLPIAgent.ps1
1  # Variables
2  $AgentVersion = "1.5"
3  $MsiFile = "GLPI-Agent-$AgentVersion-x64.msi"
4  $DownloadUrl = "https://github.com/glpi-project/glpi-agent/releases/download/$AgentVersion/$MsiFile"
5  $TempPath = "$env:TEMP\$MsiFile"
6  $ConfigPath = "C:\ProgramData\GLPI-Agent"
7  $ConfigFile = "$ConfigPath\glpi-agent.cfg"
8  $ServerURL = "http://192.168.103.240:8080/"
9  $User = "Agent"
10 $Password = "Agent"
11
12 # Téléchargement du MSI
13 Write-Host "Téléchargement de GLPI-Agent $AgentVersion..."
14 Invoke-WebRequest -Uri $DownloadUrl -OutFile $TempPath -UseBasicParsing
15
16 # Installation silencieuse
17 Write-Host "Installation silencieuse de GLPI-Agent..."
18 Start-Process msixexec.exe -ArgumentList "/i `"$TempPath`" /qn" -Wait
19
20 # Création du dossier de configuration
21 if (!(Test-Path $ConfigPath)) {
22     New-Item -ItemType Directory -Path $ConfigPath -Force
23 }
24
25 # Création du fichier glpi-agent.cfg
26 @"
27 server = $ServerURL
28 user = $User
29 password = $Password
30 "@ | Out-File -FilePath $ConfigFile -Encoding UTF8 -Force
31
32 Write-Host "Configuration créée dans $ConfigFile"
33
34 # Chemin réel de l'exécutable GLPI-Agent 1.5
35 $AgentExe = "C:\Program Files\GLPI-Agent\perl\bin\glpi-agent.exe"
36
37 # Lancement manuel de l'inventaire
38 if (Test-Path $AgentExe) {
39     Write-Host "Lancement de l'inventaire..."
40     & $AgentExe --server "http://192.168.103.240:8080/" --debug
41 } else {
42     Write-Host "Erreur : l'exécutable GLPI-Agent n'a pas été trouvé."
43 }
44
```

Image 29 : Script .ps1 à mettre dans un powershell Administrateur

Fonctionnement du script

Le script développé permet :

- de télécharger automatiquement l'installateur depuis une URL
- de lancer l'installation en mode silencieux
- de configurer l'agent avec les paramètres par défaut
- de définir la cible distante (URL GLPI)
- de lancer un premier inventaire

Sur le plan technique, le script fonctionne correctement : l'agent est bien installé sur les postes, et les services associés sont créés.

Problème rencontré

Malgré une installation réussie, les inventaires ne remontent pas automatiquement dans GLPI.

Plusieurs pistes ont été explorées :

- vérification de l'URL cible
- vérification des droits réseau
- analyse des logs de l'agent
- comparaison entre installation manuelle et installation automatisée

Le problème semble lié à un paramètre interne de l'agent lors d'une installation silencieuse, probablement au niveau de la configuration initiale ou de l'enregistrement du service.

Faute de temps et la faite que j'ai du tout refaire de A à Z, ce dysfonctionnement n'a pas pu être entièrement résolu.

8. Évolutions et perspectives du projet

L'instance GLPI mise en place dans le cadre de ce projet constitue une base solide, stable et parfaitement exploitable pour une utilisation professionnelle. Plusieurs pistes d'évolution peuvent être envisagées afin d'étendre ses fonctionnalités, d'améliorer son intégration dans un environnement réel et d'optimiser son exploitation au quotidien.

8.1 Accessibilité et portail client

À moyen terme, l'une des évolutions les plus pertinentes consisterait à rendre GLPI accessible directement aux clients professionnels via une interface web dédiée. Cette ouverture permettrait aux utilisateurs de créer, consulter et suivre leurs tickets en toute autonomie, sans dépendre systématiquement d'un appel téléphonique ou d'un échange de mails. Une telle organisation contribuerait à réduire les interruptions pour les techniciens, à fluidifier le traitement des demandes et à améliorer la traçabilité des échanges.

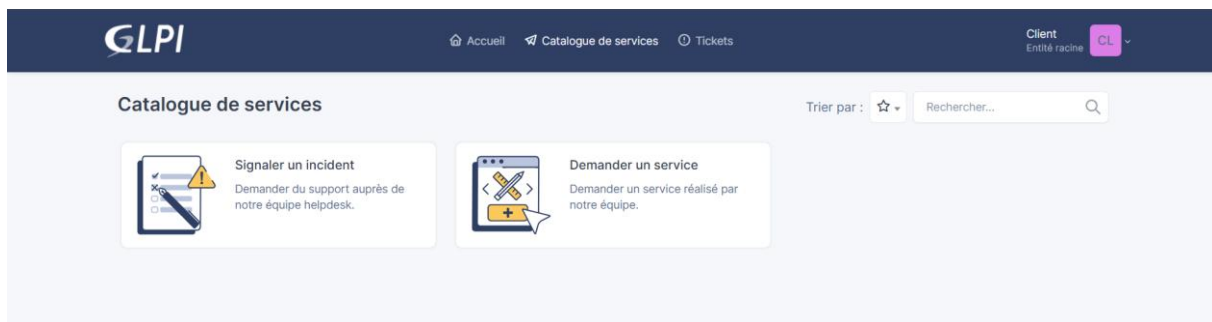


Image 30 : Affichage pour les clients

8.2 Automatisation du déploiement de GLPI Agent

L'automatisation du déploiement de GLPI Agent représente une autre évolution majeure. Le script PowerShell développé dans le cadre du projet constitue une première base fonctionnelle, mais pourrait être finalisé et renforcé afin de permettre une installation totalement automatisée de l'agent sur l'ensemble des postes clients. Une automatisation faciliterait considérablement la mise en place de l'inventaire, garantirait une homogénéité des configurations et réduirait le temps nécessaire au déploiement sur un parc informatique de grande taille.

8.3 Migration de la base GLPI existante vers l'architecture Docker

À plus long terme, il serait envisageable de migrer la base GLPI actuellement utilisée en entreprise vers cette nouvelle architecture Docker. Une migration offrirait plusieurs avantages : meilleure portabilité, simplification des mises à jour, isolation des services et possibilité de déployer l'environnement sur différents serveurs ou infrastructures cloud. Cette opération nécessiterait toutefois une phase de tests afin de garantir la compatibilité des données, la stabilité de l'environnement et la continuité du service pour les utilisateurs.

8.4 Renforcement de la sécurité et supervision avancée

Enfin, plusieurs améliorations pourraient être apportées en matière de sécurité et de supervision :

- mise en place d'un accès sécurisé via HTTPS
- durcissement de la gestion des droits utilisateurs
- supervision avancée des conteneurs Docker (logs, alertes, monitoring)
- sauvegardes automatisées de la base de données et des volumes persistants

Ces évolutions permettraient d'atteindre un niveau de fiabilité et de sécurité conforme aux standards professionnels, tout en assurant une meilleure résilience de l'infrastructure.

9. Mise en lien avec les apprentissages critiques du BUT

R&T

La réalisation de ce projet autour de la mise en place d'une instance GLPI conteneurisée m'a permis de mobiliser et de valider plusieurs apprentissages critiques du BUT Réseaux & Télécommunications, parcours Cybersécurité. L'ensemble des tâches effectuées comme la conception, déploiement, configuration, automatisation, documentation, correspond directement aux compétences attendues en fin de formation.

Tout d'abord, la conception de l'architecture Docker et l'organisation des services GLPI et MariaDB m'ont permis d'appliquer les principes de conception d'un système d'information fiable et structuré. Le choix de séparer les services, d'utiliser des volumes persistants, de configurer un serveur SMTP sécurisé et de réfléchir à l'ouverture future de l'outil aux clients s'inscrit pleinement dans les compétences liées à la **conception d'un projet réseau intégrant sécurité, supervision et disponibilité** (AC31.01).

La rédaction du rapport technique, la structuration des étapes, l'explication des choix et la justification des configurations répondent directement aux exigences de **documentation technique d'un projet** (AC31.02). Chaque partie du projet a été documentée : architecture Docker, installation, configuration fonctionnelle, gestion des utilisateurs, notifications, inventaire, automatisation. Cette production constitue une documentation complète et exploitable.

La mise en place d'une instance GLPI fonctionnelle, accessible via navigateur, avec des utilisateurs, des tickets, des notifications et un inventaire automatique, constitue une **maquette de démonstration opérationnelle** (AC31.03). Elle permet de visualiser concrètement le fonctionnement d'un outil de support informatique moderne et de montrer la pertinence d'une architecture conteneurisée.

Tout au long du projet, j'ai dû analyser des problèmes techniques, comparer différentes solutions, justifier mes choix et proposer des améliorations futures (HTTPS, portail client, migration de la base, automatisation du déploiement de GLPI Agent). Cette démarche correspond à la compétence **défendre et argumenter un projet** (AC31.04), en montrant une capacité à prendre du recul et à proposer des évolutions réalistes.

La gestion du projet a également été un point central. L'élaboration d'un Gantt prévisionnel, puis d'un Gantt réel, m'a permis d'organiser les tâches, d'anticiper les difficultés, d'adapter la

planification et de gérer les imprévus techniques (problèmes SMTP, actions automatiques, script PowerShell). Cette démarche valide la compétence **gérer un projet et ses étapes en respectant les délais** (AC31.06).

Le projet a également permis de travailler sur la mise en place d'un système de communication interne via les notifications GLPI. La configuration du serveur SMTP, la personnalisation des modèles de notification et la validation des envois automatiques correspondent à la compétence **déployer un système de communication pour l'entreprise** (AC32.01).

La gestion des profils, des droits, des entités et la segmentation des accès entre techniciens et clients répondent à la compétence **permettre aux collaborateurs de se connecter de manière sécurisée au système d'information** (AC32.04). Le cloisonnement des tickets, l'interface simplifiée pour les clients et les droits avancés pour les techniciens illustrent une gestion fine et sécurisée des accès.

Enfin, la mise en place de GLPI Agent, la configuration de l'inventaire automatique, l'analyse des données remontées et la tentative d'automatisation via PowerShell relèvent de la compétence **déployer et maintenir une solution informatique** (AC33.04). Ce travail a permis de comprendre les mécanismes d'inventaire, les interactions entre agents et serveur, et les limites d'un déploiement automatisé.

11. Conclusion

La mise en place d'un GLPI conteneurisée à l'aide de Docker a permis de mener un projet complet, allant de la conception de l'architecture jusqu'à la configuration avancée des fonctionnalités de l'outil. Ce travail a démontré qu'il est possible de déployer une solution moderne, flexible et plus simple à maintenir que l'architecture actuellement utilisée en entreprise, tout en conservant l'ensemble des fonctionnalités nécessaires à un environnement professionnel.

L'installation de GLPI via Docker, la configuration de la base de données, la gestion des utilisateurs, la mise en place des entités, des profils et des droits ont permis de reproduire un environnement réaliste, adapté aussi bien aux techniciens qu'aux clients. La mise en place des notifications par mail, associée à la configuration du serveur SMTP et à l'ajustement des actions automatiques, a permis d'obtenir un système de communication fiable et automatisé, indispensable dans un contexte de support informatique.

L'intégration de GLPI Agent et la mise en place de l'inventaire automatique m'ont confirmé la capacité de GLPI à centraliser efficacement les informations matérielles et logicielles des postes clients. Même si l'automatisation complète du déploiement de l'agent via script PowerShell n'a pas pu être finalisée, cette tentative constitue une base solide pour de futurs travaux et met en évidence l'automatisation dans un environnement hétérogène.

Ce projet m'a également permis de prendre du recul sur l'outil actuellement utilisé en entreprise et d'identifier des pistes d'amélioration concrètes : ouverture d'un portail client, migration vers une architecture conteneurisée, renforcement de la sécurité, automatisation du déploiement des agents, supervision avancée. Ces perspectives montrent que le travail réalisé ne se limite pas à un exercice académique, mais peut servir de fondation pour une évolution réelle de l'infrastructure existante.

En définitive, ce projet a été particulièrement formateur. Il m'a permis de développer des compétences techniques avancées, d'acquérir une méthodologie rigoureuse de résolution des problèmes et de produire une documentation complète. Il constitue une expérience significative, en cohérence avec les attentes du BUT 3 Réseaux & Télécommunications parcours Cybersécurité.