# Étude et implémentation de GLPI pour la gestion des services IT

**Juste Fourier ACKO**

Ingénieur en Formation en Réseaux et Cybersécurité

# Table des matières

## Introduction

Dans un contexte où la gestion efficace des ressources informatiques est devenue essentielle pour les entreprises, les solutions de gestion de parc et de services IT jouent un rôle central. GLPI (Gestionnaire Libre de Parc Informatique) est une application open-source reconnue pour sa capacité à centraliser la gestion du matériel, des logiciels, des incidents et des utilisateurs.

Ce projet a pour objectif d'installer, configurer et exploiter GLPI sur un serveur Ubuntu 22.04. L'installation sur cette distribution moderne permet de bénéficier d'un environnement stable et sécurisé, tout en utilisant les dernières versions des composants nécessaires, tels que PHP, MySQL/MariaDB et Apache.

À travers ce projet, nous allons mettre en place un système complet de gestion de parc informatique, démontrant la valeur ajoutée de GLPI dans le suivi des actifs, la maintenance et l'optimisation des ressources IT.

## I. Modules et capacités de GLPI

GLPI offre un ensemble complet de fonctionnalités destinées à faciliter la gestion des ressources informatiques et à optimiser le support aux utilisateurs au sein d'une organisation. Ses principales fonctions incluent :

### a. Gestion du parc informatique

- Inventaire automatique et manuel du matériel informatique (PC, serveurs, imprimantes, équipements réseau, etc.).
- Gestion des logiciels installés et de leurs licences.
- Suivi des configurations matérielles et logicielles.

### b. Gestion des incidents et des demandes

- Création et suivi des tickets d'incidents ou de demandes de service.
- Attribution des tickets aux techniciens selon les compétences ou la disponibilité.
- Historique complet des interventions pour analyse et reporting.

### c. Gestion des utilisateurs et des droits

- Gestion des utilisateurs, groupes et profils avec des droits d'accès personnalisés.
- Intégration possible avec LDAP/Active Directory pour centraliser les identités.

### d. Gestion des contrats et fournisseurs

- Suivi des contrats de maintenance, garanties et licences.
- Gestion des fournisseurs et des contacts pour le support ou les achats.

### e. Planification et suivi des interventions

- Planification des tâches récurrentes et préventives.
- Gestion des calendriers d'intervention et notifications automatiques.

### f. Rapports et tableaux de bord

- Génération de rapports sur le parc, les incidents, les interventions ou les coûts.
- Tableaux de bord personnalisables pour un suivi en temps réel.

### g. Extensions et plugins

- Possibilité d'étendre les fonctionnalités avec des plugins (ex. GLPI Fusion Inventory pour l'inventaire automatique).
- Intégration avec d'autres outils ITSM ou systèmes de monitoring.

## II. Mise en place de l'environnement

Avant l'installation de GLPI, il est essentiel de préparer un environnement serveur adapté, en installant les composants et dépendances nécessaires pour garantir son bon fonctionnement.

GLPI repose sur une architecture LAMP (Linux, Apache, MySQL/MariaDB, PHP) et nécessite également certains modules complémentaires.

### a. Système d'exploitation

- **Ubuntu 22.04 LTS** : version stable et récente, offrant sécurité et support à long terme.
- Mise à jour du système pour s'assurer que tous les paquets sont à jour :

```
sudo apt update && sudo apt upgrade –y
```

### b. Serveur web

- **Apache2** : serveur HTTP pour héberger l'application GLPI.

```
sudo apt install apache2 –y
```

### c. Base de données

- **MariaDB ou MySQL** : gestion des données de GLPI. MariaDB dans ce cas.
- Création d'une base dédiée et d'un utilisateur avec les droits appropriés.

```
sudo apt install mariadb-server mariadb-client –y
```

Après l'installation de MariaDB, il est nécessaire de démarrer le service et de l'activer pour qu'il se lance automatiquement à chaque démarrage du serveur.

```
sudo systemctl start mariadb    # Démarre le service MariaDB

sudo systemctl enable mariadb   # Active MariaDB au démarrage du serveur
```

### d. PHP et extensions

Pour assurer le bon fonctionnement de GLPI, il est nécessaire d'installer PHP ainsi que ses extensions indispensables, permettant la connexion à la base de données, la gestion des mails, et le traitement des données côté serveur.

```
sudo apt install php-mysql php-curl php-gd php-mbstring php-xml php-ldap
php-imap
```

### e. Configuration d'Apache et dossier d'hébergement de GLPI

Après avoir installé GLPI et ses dépendances, il est nécessaire de configurer le serveur web Apache pour qu'il puisse héberger l'application et la rendre accessible via un navigateur. Sur Ubuntu, la configuration se fait généralement via les fichiers situés dans ***/etc/apache2/sites-available/***.

```
sudo nano /etc/apache2/sites-available/glpi.conf
```

```
<VirtualHost *:80>

    ServerName glpi.lan

    DocumentRoot /var/www/html/glpi

    <Directory /var/www/html/glpi>

        AllowOverride All

        Require all granted

    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/glpi_error.log

    CustomLog ${APACHE_LOG_DIR}/glpi_access.log combined

</VirtualHost>
```

### f. Activation du site et modules Apache

- Activer le site GLPI :

```
sudo a2ensite glpi.conf
```

- Activer le module rewrite (nécessaire pour GLPI) :

```
sudo a2enmod rewrite
```

- Vérifier la configuration et redémarrer Apache :

```
sudo systemctl restart apache2
```

## III. Connexion à MariaDB et création de la base de données pour GLPI

Après avoir installé et démarré MariaDB, il faut créer la base de données et l'utilisateur dédiés à GLPI, afin de sécuriser l'accès et de gérer les privilèges correctement.

### a. Connexion à MariaDB

Se connecter au serveur MariaDB en tant qu'utilisateur root :

```
sudo mysql -u root -p
```

- Il vous sera demandé le mot de passe root défini lors de la sécurisation de MariaDB.

### b. Création de la base de données

Créer une base de données pour GLPI :

```
CREATE DATABASE glpidb;
```

### c. Création de l'utilisateur GLPI

Créer un utilisateur dédié, par exemple glpiuser, avec un mot de passe sécurisé :

```
CREATE USER 'glpiuser'@'localhost' IDENTIFIED BY 'motdepasse';

//modifier motdepasse par votre motdepasse si besoin
```

### d. Accorder les privilèges

Donner à cet utilisateur tous les privilèges sur la base GLPI :

```
GRANT ALL PRIVILEGES ON glpidb.* TO 'glpiuser'@'localhost';
FLUSH PRIVILEGES;
```

FLUSH PRIVILEGES permet d'appliquer immédiatement les modifications.

## IV. Téléchargement de GLPI depuis GitHub

La première étape consiste à récupérer la dernière version stable de GLPI depuis son dépôt officiel GitHub. Cela permet de disposer des fichiers d'installation les plus récents et de bénéficier des corrections de bugs et des nouvelles fonctionnalités.

### a. Se placer dans le dossier d'hébergement

Si le dossier /var/www/html/glpi a été créé précédemment :

```
cd /var/www/html
```

### b. Téléchargement depuis GitHub

On peut cloner le dépôt officiel de GLPI avec Git :

```
sudo apt install git -y           # Installer Git si ce n'est pas déjà fait

sudo wget https://github.com/glpi-
project/glpi/releases/download/10.0.19/glpi-10.0.19.tgz
```

Le dossier glpi contiendra tous les fichiers nécessaires pour l'installation.

### c. Extraire l'archive

```
Tar xvzf glpi-10.0.19.tgz
```

### d. Vérification des droits

Après le téléchargement, il est important de s'assurer que le serveur web Apache a les droits corrects sur le dossier :

```
sudo chown -R www-data:www-data /var/www/html/glpi
```

```
sudo chmod -R 755 /var/www/html/glpi
```

## V. Accès à l'interface de configuration de GLPI

Une fois les fichiers de GLPI téléchargés et le serveur Apache correctement configuré, l'installation se poursuit via l'interface web de configuration. Cette interface permet de finaliser l'installation en reliant GLPI à la base de données et en configurant les paramètres initiaux.

### a. Accès via un navigateur

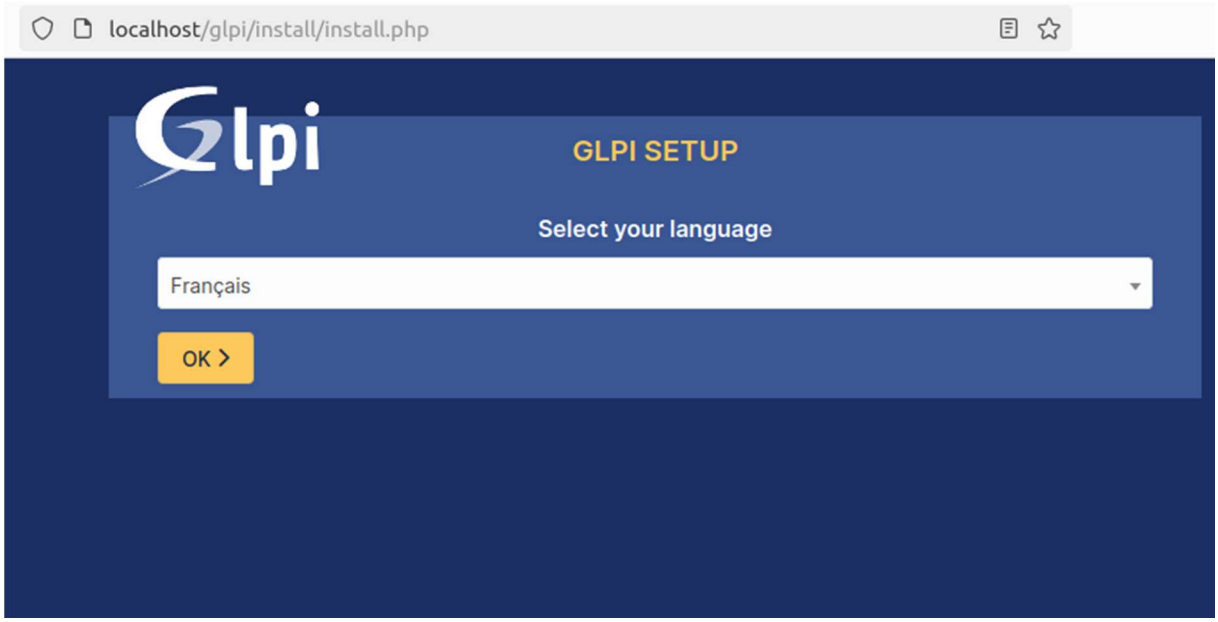
Depuis un navigateur web, accéder à l'adresse suivante :

http://localhost/glpi

### b. Lancement de l'assistant d'installation

Lors du premier accès, GLPI affiche l'assistant d'installation :

- Choix de la langue
- Acceptation de la licence
- Vérification des prérequis (PHP, extensions, permissions



### c. Configuration de la base de données

L'assistant demande ensuite les informations de connexion à MariaDB :

- Nom de la base de données : glpi
- Utilisateur : glpiuser
- Mot de passe : celui défini lors de la création de l'utilisateur


Si la connexion est valide, GLPI crée automatiquement les tables nécessaires.

### d. Finalisation de l'installation

Une fois l'installation terminée :

- Les comptes par défaut sont créés (administrateur, technicien, utilisateur)
- GLPI demande de supprimer ou renommer le dossier install pour des raisons de sécurit

```
sudo rm -rf /var/www/html/glpi/install
```

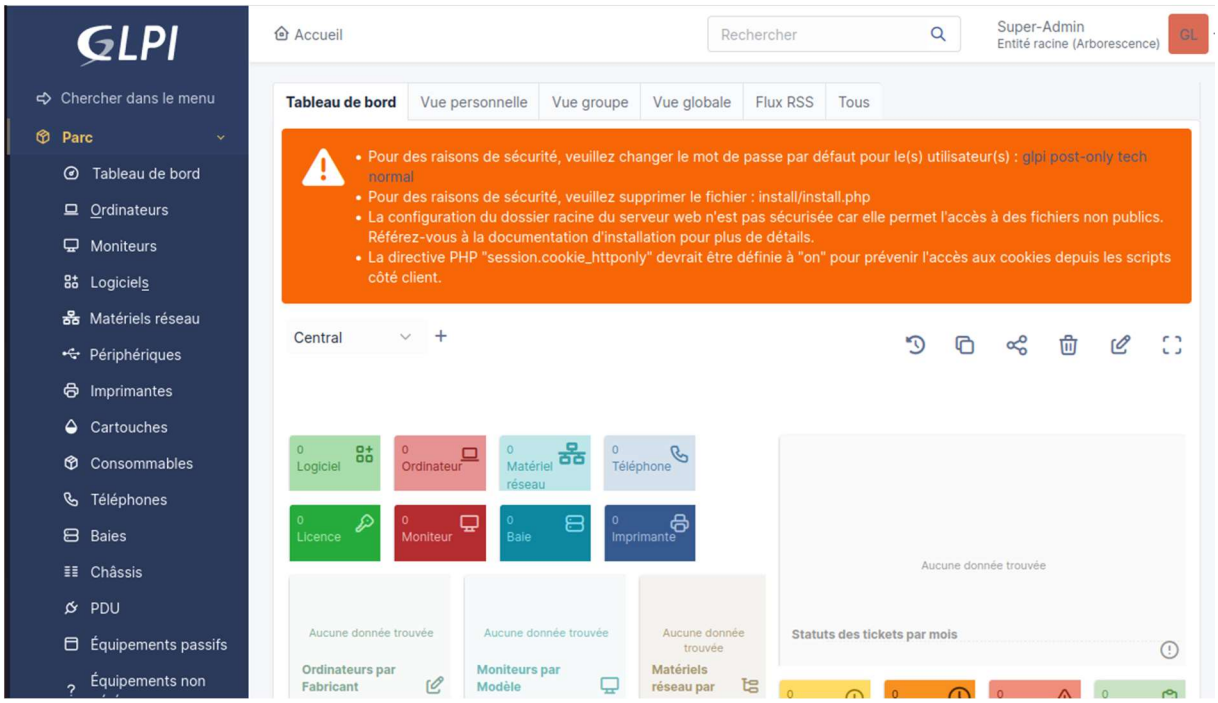
### e. Accès à l'interface GLPI

Après la finalisation, l'interface principale de GLPI est accessible et prête à être utilisée pour la gestion du parc informatique et des tickets.

Identifiants de connexion

Identifiant : glpi

Password : glpi




## VI. Passage en mode sécurisé HTTPS avec un certificat OpenSSL

Dans un environnement de test ou interne, il est possible de sécuriser l'accès à GLPI en utilisant un certificat SSL/TLS auto-signé généré avec OpenSSL. Cette solution permet de chiffrer les échanges entre le client et le serveur sans dépendre d'une autorité de certification externe.

### a. Activation du module SSL d'Apache

Avant de configurer HTTPS, il est nécessaire d'activer le module SSL d'Apache :

```
sudo a2enmod ssl
sudo systemctl restart apache2
```

### b. Création du certificat auto-signé

Créer un répertoire pour stocker le certificat et la clé privée :

```
sudo mkdir /etc/apache2/ssl
```

Générer le certificat et la clé avec OpenSSL :

```
sudo openssl req -x509 -nodes -days 365 \
-newkey rsa:2048 \
-keyout /etc/apache2/ssl/glpi.key \
-out /etc/apache2/ssl/glpi.crt
```

- Le certificat est valide pendant 365 jours
- La clé privée est générée sans mot de passe (-nodes)
- Les informations demandées (CN) doivent correspondre au nom de domaine ou à l'adresse IP du serveur

## VII. Configuration du VirtualHost HTTPS

Créer ou modifier le fichier de configuration SSL :

```
sudo nano /etc/apache2/sites-available/glpi-ssl.conf
```

Exemple de configuration :

```
<VirtualHost *:443>
    ServerName glpi.local
    DocumentRoot /var/www/html/glpi

    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/glpi.crt
    SSLCertificateKeyFile /etc/apache2/ssl/glpi.key

    <Directory /var/www/html/glpi>
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/glpi_ssl_error.log
    CustomLog ${APACHE_LOG_DIR}/glpi_ssl_access.log combined
</VirtualHost>
```

### a. Activation du site SSL

Activer le site HTTPS et redémarrer Apache :

```
sudo a2ensite glpi-ssl.conf
sudo systemctl restart apache2
```

### b. Accès sécurisé à GLPI

GLPI est désormais accessible via :

https://localhost:443

Un avertissement de sécurité apparaît dans le navigateur, car le certificat est auto-signé. Il peut être accepté manuellement dans un contexte de test.

Grâce à la mise en place d'un certificat SSL auto-signé avec OpenSSL et à la redirection forcée vers HTTPS, l'accès à GLPI est désormais sécurisé. Les échanges entre le serveur et les utilisateurs sont chiffrés, garantissant la confidentialité des informations, même dans un environnement de test ou de laboratoire.

## Conclusion

Ce projet a permis de mettre en place une solution complète de gestion de parc informatique et de support aux utilisateurs à travers l'installation et la configuration de GLPI sur un serveur Ubuntu 22.04. L'ensemble des étapes a été réalisé de manière progressive et structurée, depuis la préparation de l'environnement jusqu'à la sécurisation de l'accès à l'application.

La mise en œuvre des composants essentiels, notamment Apache, MariaDB et PHP, a permis de garantir un environnement stable et fonctionnel. La création d'une base de données dédiée et d'un utilisateur spécifique a renforcé la sécurité et la bonne gestion des accès. L'installation de GLPI via son interface web a ensuite facilité la configuration initiale et la mise en service de l'application.

Enfin, le passage en mode sécurisé HTTPS à l'aide d'un certificat SSL auto-signé généré avec OpenSSL a permis de chiffrer les échanges et d'assurer la confidentialité des données, même dans un environnement de test. Cette étape illustre l'importance de la sécurité dans le déploiement d'applications web, en particulier pour des outils de gestion IT.

Ainsi, ce projet démontre que GLPI constitue une solution efficace et robuste pour la gestion des ressources informatiques et des services IT. Il peut être enrichi par la suite par des configurations avancées telles que l'intégration LDAP, la gestion des sauvegardes ou l'ajout de plugins, afin de répondre aux besoins spécifiques d'une organisation.