

# GLPI

Nous allons effectuer une installation **pas-à-pas de GLPI 10 sur une machine Debian 12, en mettant en place Apache2, PHP 8.2 (PHP-FPM) et MariaDB Server.**

**GLPI** est un **logiciel libre de gestion de parc informatique** permettant d'avoir une **solution de ticketing gratuite** pour le support informatique, de gérer l'inventaire des équipements, notamment les ordinateurs et les téléphones, de gérer ses contrats, ses licences, ses consommables, ses baies serveurs, etc.... Créé en 2003, **GLPI est une solution populaire** utilisée par des milliers d'entreprises.

Bien que son éditeur Teclib propose une version payante et hébergée en mode SaaS, GLPI est toujours gratuit et vous pouvez l'héberger sur votre serveur, que ce soit pour vos besoins internes ou pour vos clients, notamment pour la gestion des tickets de support.

- [Site officiel de GLPI](#)

**Si vous souhaitez mettre à jour GLPI plutôt que de réaliser une nouvelle installation, suivez plutôt ce tutoriel :**

- [Comment mettre à jour GLPI ?](#)

## II. Prérequis de GLPI

Avant d'évoquer l'installation, parlons des prérequis. GLPI a besoin d'**un serveur Web, de PHP et d'une base de données** pour fonctionner. Sous Linux, ceci correspond à un **socle LAMP**. Il supporte **plusieurs serveurs Web** : Apache2, Nginx, lighttpd et IIS.

Pour le reste, voici ce que vous devez savoir :

- **Version de PHP**
  - Minimum : **PHP 7.4 (plus supportée !)**
  - Maximum : **PHP 8.2**
- **Base de données**
  - MySQL 5.1 minimum
  - MariaDB 10.2 minimum

Il y aura également plusieurs **extensions PHP** à installer pour que GLPI puisse fonctionner.

La dernière version de GLPI à l'heure où j'écris ces lignes, à savoir **GLPI 10.0.10, ajoute le support de PHP 8.3 (future version stable) et MySQL 8.1**, en plus de corriger de nombreuses vulnérabilités critiques.

Pour cette démonstration, nous allons utiliser une machine sous **Debian 12** et nous allons installer dessus **Apache2, PHP 8.3 ainsi que MariaDB**.

Si vous avez besoin de précisions supplémentaires, vous pouvez consulter la documentation officielle :

- [Les prérequis de GLPI](#)

## III. Préparer le serveur pour installer GLPI

Commençons par l'installation par une **mise à jour des paquets sur la machine Debian 12**. Pensez également à lui attribuer une adresse IP et à effectuer la configuration du système.

```
sudo apt-get update && sudo apt-get upgrade
```

### A. Installer le socle LAMP

La première grande étape consiste à installer les paquets du socle LAMP : **Linux Apache2 MariaDB PHP**. Sous **Debian 12**, qui est la dernière version stable de Debian, **PHP 8.2** est distribué par défaut dans les dépôts officiels.

Commençons par installer ces trois paquets :

```
sudo apt-get install apache2 php mariadb-server
```

Puis, nous allons installer toutes les extensions nécessaires au bon fonctionnement de GLPI.

```
sudo apt-get install php-xml php-common php-json php-mysql php-mbstring php-curl php-gd php-intl php-zip php-bz2 php-imap php-apcu
```

**Ces commandes vont permettre de récupérer les versions de ces extensions pour PHP 8.2.**

Si vous envisagez d'associer GLPI avec un annuaire LDAP comme l'Active Directory, vous devez installer l'extension LDAP de PHP. Sinon, ce n'est pas nécessaire et vous pouvez le faire par la suite, si besoin.

```
sudo apt-get install php-ldap
```

Nous venons d'installer Apache2, MariaDB, PHP et un ensemble d'extensions.

### B. Préparer une base de données pour GLPI

Nous allons préparer MariaDB pour qu'il puisse héberger la base de données de GLPI. La première action à effectuer, c'est d'exécuter la commande ci-dessous pour **effectuer le minimum syndical en matière de sécurisation de MariaDB**.

```
sudo mysql_secure_installation
```

Vous serez invité à changer le mot de passe root, mais aussi à supprimer les utilisateurs anonymes, désactiver l'accès root à distance, etc... Tout est bien expliqué. Voici un exemple sur mon serveur pour vous guider :

Ensuite, nous allons créer **une base de données dédiée pour GLPI** et celle-ci sera accessible par **un utilisateur dédié**. Hors de question d'utiliser le compte root de MariaDB : une base de données = un utilisateur.

Connectez-vous à votre instance MariaDB :

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

Saisissez le mot de passe root de MariaDB, que vous venez de définir à l'étape précédente.

Puis, nous allons exécuter les **requêtes SQL** ci-dessous pour **créer la base de données "db23\_glpi"** ainsi que l'utilisateur **"glpi\_adm"** avec le **mot de passe "MotDePasseRobuste"** (que vous changez, bien sûr). Cet utilisateur aura tous les droits sur cette base de données (et uniquement sur celle-ci).

```
CREATE DATABASE db23_glpi;  
GRANT ALL PRIVILEGES ON db23_glpi.* TO glpi_adm@localhost IDENTIFIED BY  
"MotDePasseRobuste";  
FLUSH PRIVILEGES;  
EXIT
```

Voilà, la base de données prête.

## C. Télécharger GLPI et préparer son installation

La prochaine étape consiste à **télécharger l'archive ".tgz"** qui contient les sources d'installation de GLPI. A partir du **GitHub de GLPI**, récupérez le lien vers la dernière version. Ici, c'est la version **GLPI 10.0.10** qui est installée.

- [GitHub de GLPI](#)

L'archive sera téléchargée dans le répertoire **"/tmp"** :

```
cd /tmp
wget https://github.com/glpi-project/glpi/releases/download/10.0.10/glpi-10.0.10.tgz
```

Puis, nous allons exécuter la commande ci-dessous pour **décompresser l'archive .tgz dans le répertoire "/var/www/"**, ce qui donnera le chemin d'accès **"/var/www/glpi"** pour GLPI.

```
sudo tar -xzf glpi-10.0.10.tgz -C /var/www/
```

Nous allons définir l'utilisateur **"www-data"** correspondant à **Apache2**, en tant que **propriétaire** sur les fichiers GLPI.

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

Ensuite, nous allons devoir **créer plusieurs dossiers** et sortir des données de la racine Web (/var/www/glpi) de manière à les stocker dans les nouveaux dossiers que nous allons créer. Ceci va permettre de faire une **installation sécurisée de GLPI, qui suit les recommandations de l'éditeur**.

- **Le répertoire /etc/glpi**

Commencez par **créer le répertoire "/etc/glpi"** qui va recevoir les fichiers de configuration de GLPI. Nous donnons des autorisations à www-data sur ce répertoire car il a besoin de pouvoir y accéder.

```
sudo mkdir /etc/glpi
sudo chown www-data /etc/glpi/
```

Nous n'avons rien à déplacer dans ce répertoire.

- **Créer les fichiers de configuration**

Nous devons configurer GLPI pour qu'il sache où aller chercher les données. Autrement dit, nous allons déclarer les nouveaux répertoires fraîchement créés.

Nous allons créer ce premier fichier :

```
sudo nano /var/www/glpi/inc/downstream.php
```

Afin d'ajouter le contenu ci-dessous qui indique le chemin vers le **répertoire de configuration** :

```
<?php
define('GLPI_CONFIG_DIR', '/etc/glpi/');
if (file_exists(GLPI_CONFIG_DIR . '/local_define.php')) {
    require_once GLPI_CONFIG_DIR . '/local_define.php';
}
```

Ensuite, nous allons créer ce second fichier :

```
sudo nano /etc/glpi/local_define.php
```

Afin d'ajouter le contenu ci-dessous permettant de **déclarer deux variables** permettant de préciser les chemins vers **les répertoires "files" et "log"** que l'on a préparé précédemment.

```
<?php
define('GLPI_VAR_DIR', '/var/lib/glpi/files');
define('GLPI_LOG_DIR', '/var/log/glpi');
```

Voilà, cette étape est terminée.

## D. Préparer la configuration Apache2

Passons à la configuration du serveur web Apache2. Nous allons créer un nouveau fichier de configuration qui va permettre de configurer le VirtualHost dédié à GLPI. Dans mon cas, le fichier s'appelle "support.it-connect.tech.conf" en référence au nom de domaine choisi pour accéder à GLPI : **support.it-connect.tech**. L'idéal étant d'avoir un nom de domaine (même interne) pour accéder à GLPI afin de pouvoir positionner un certificat SSL par la suite.

```
sudo nano /etc/apache2/sites-available/support.it-connect.tech.conf
```

Ce qui donne la configuration suivante (selon le modèle officiel de la documentation) :

```
<VirtualHost *:80>
    ServerName support.it-connect.tech

    DocumentRoot /var/www/glpi/public

    # If you want to place GLPI in a subfolder of your site (e.g. your
    virtual host is serving multiple applications),
    # you can use an Alias directive. If you do this, the DocumentRoot
    directive MUST NOT target the GLPI directory itself.
```

```
# Alias "/glpi" "/var/www/glpi/public"

<Directory /var/www/glpi/public>
    Require all granted

    RewriteEngine On

    # Redirect all requests to GLPI router, unless file exists.
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteRule ^(.*)$ index.php [QSA,L]
</Directory>
</VirtualHost>
```

Quand la configuration est prête, enregistrez le fichier.

Puis, nous allons **activer ce nouveau site dans Apache2** :

```
sudo a2ensite support.it-connect.tech.conf
```

Nous en profitons également pour **désactiver le site par défaut** car il est inutile :

```
sudo a2dissite 000-default.conf
```

Nous allons aussi **activer le module "rewrite"** (pour les règles de réécriture) car on l'a utilisé dans le fichier de configuration du VirtualHost (*RewriteCond* / *RewriteRule*).

```
sudo a2enmod rewrite
```

Il ne reste plus qu'à **redémarrer le service Apache2** :

```
sudo systemctl restart apache2
```

## E. Utilisation de PHP8.2-FPM avec Apache2

Pour utiliser PHP en tant que moteur de scripts avec Apache2, il y a deux possibilités : **utiliser le module PHP pour Apache2 (libapache2-mod-php8.2) ou utiliser PHP-FPM.**

Il est **recommandé d'utiliser PHP-FPM** car il est plus performant et se présente comme un service indépendant. Dans l'autre mode, chaque processus Apache2 exécute son propre moteur de scripts PHP.

Si vous souhaitez utiliser PHP-FPM, suivez les étapes ci-dessous. Sinon, passez à la suite mais veuillez à **configurer l'option "session.cookie\_httponly"** évoquée ci-dessous.

Nous allons commencer par **installer PHP8.2-FPM** avec la commande suivante :

```
sudo apt-get install php8.2-fpm
```

Puis, nous allons activer deux modules dans Apache et la configuration de PHP-FPM, avant de recharger Apache2 :

```
sudo a2enmod proxy_fcgi setenvif  
sudo a2enconf php8.2-fpm  
sudo systemctl reload apache2
```

Pour **configurer PHP-FPM pour Apache2**, nous n'allons pas éditer le fichier **"/etc/php/8.2/apache2/php.ini"** mais **plutôt** ce fichier :

```
sudo nano /etc/php/8.2/fpm/php.ini
```

Dans ce fichier, recherchez l'option **"session.cookie\_httponly"** et indiquez la valeur "on" pour l'activer, afin de protéger les cookies de GLPI.

```
; Whether or not to add the httpOnly flag to the cookie, which makes it;  
inaccessible to browser scripting languages such as JavaScript.;  
https://php.net/session.cookie-httponly  
session.cookie_httponly = on
```

Enregistrez le fichier quand c'est fait. Par la suite, vous pourriez être amené à effectuer d'autres modifications, notamment pour augmenter la taille des uploads sur GLPI, etc.

Pour appliquer les modifications, nous devons redémarrer PHP-FPM :

```
sudo systemctl restart php8.2-fpm.service
```

Pour finir, nous devons **modifier notre VirtualHost** pour préciser à Apache2 que PHP-FPM doit être utilisé pour les fichiers PHP :

```
<FilesMatch \.php$>  
    SetHandler "proxy:unix:/run/php/php8.2-fpm.sock|fcgi://localhost/"  
</FilesMatch>
```

Quand c'est fait, relancer Apache2 :

```
sudo systemctl restart apache2
```

**Voilà, tout est prêt ! Il ne reste plus qu'à installer GLPI !**

## IV. Installation de GLPI

Pour effectuer l'installation de GLPI, nous devons utiliser un navigateur Web afin d'accéder à l'adresse du GLPI. Il s'agit de l'adresse déclarée dans le fichier de configuration Apache2 (*ServerName*).

Si vous avez suivi toutes les étapes correctement, vous devriez arriver sur cette page. Nous allons commencer par choisir la langue.

Puisqu'il s'agit d'une nouvelle installation, nous cliquons sur **"Installer"**.

Etape importante : **GLPI vérifie la configuration de notre serveur** pour déterminer si tous les prérequis sont respectés. Tout est bon, donc nous pouvons continuer.

A l'étape suivante, nous devons renseigner les informations pour se connecter à la base de données. Nous indiquons **"localhost"** en tant que serveur SQL puisque MariaDB est installé en local, sur le même serveur que GLPI. Puis, nous indiquons notre utilisateur "glpi\_adm" et le mot de passe associé.

Après avoir cliqué sur **"Continuer"**, nous devons choisir la base de données **"db23\_glpi"** créée précédemment.

Poursuivez...

Suivez les dernières étapes qui n'ont pas de réel impact. Le plus dur est fait !

**Félicitations, vous venez d'installer GLPI !** Comme le précise la dernière étape, le compte **administrateur** par défaut est **"glpi/glpi"** !

Nous allons donc nous connecter avec le compte "glpi" et le mot de passe "glpi".

**Bienvenue sur votre nouveau serveur GLPI !**

Même si l'installation est terminée, nous avons encore quelques actions à réaliser pour la finaliser :

- Changer le mot de passe de tous les comptes par défaut (cliquez sur les liens situés dans l'encadré orange)
- Supprimer le fichier "install.php" puisqu'il n'est plus nécessaire et représente un risque (relancer l'installation)

```
sudo rm /var/www/glpi/install/install.php
```

Voilà, c'est fait. Désormais, votre GLPI est prêt à être utilisé et configuré (création d'utilisateurs, de catégories, de tickets, etc...).

## V. Conclusion



**En suivant ce tutoriel pas à pas, vous devez être en mesure d'installer GLPI sur un serveur Debian 12 !** À quelques détails près, cette procédure peut s'appliquer à d'autres systèmes et versions. N'hésitez pas à poster un commentaire ou à vous connecter sur notre serveur Discord si vous avez une question.

**Souhaitez-vous plus de tutoriels sur GLPI ?** En voici déjà quelques-uns disponibles dès à présent :

- [Comment mettre à jour GLPI ?](#)
- [Comment installer un certificat HTTPS sur un serveur GLPI ?](#)
- [Comment déployer l'agent GLPI sur Windows par GPO ?](#)
- [Comment synchroniser GLPI avec l'Active Directory ?](#)