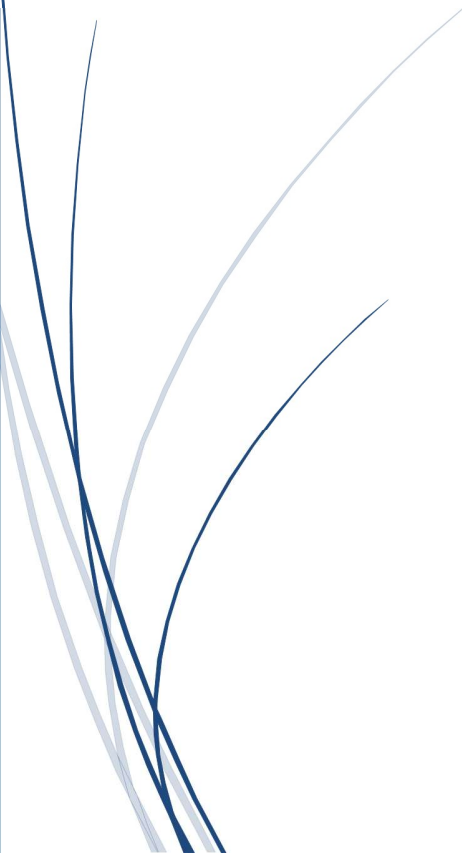




01/03/2026

# Procédure d'installation GLPI

Procédure de A à Z



GLPI Outil ticketing  
FELLAOU SOULEIMAINÉ

# Procédure d'installation

---

## 1. Informations générales

Nom de la procédure : Installation GLPI 11.0.6 sur Debian 12

Auteur : Fellaou Souleimaine

Date : 05/03/2026

Version :

Objectif : Mettre en place la solution GLPI sur un serveur Debian 13 (ou 12) afin de disposer d'un outil de gestion de parc informatique et de ticketing.

Périmètre :

- Installation du serveur web
- Installation de la base de données
- Installation de GLPI
- Configuration du service

## Attention !!

La plupart des commandes sont en root donc faire sudo si compte sans droits.

## 2. Présentation du service

Description du logiciel / service : GLPI est une solution open-source de gestion de parc informatique et d'helpdesk

Fonction principale :

- la gestion des tickets utilisateurs
- la gestion des équipements informatiques
- la gestion des licences et contrats
- la supervision du support informatique.

Architecture technique (ex : LAMP, Docker, etc.) :

- Linux (Debian)
- Apache
- MariaDB/MySQL
- PHP

### 3. Prérequis

Prérequis matériels :

- CPU: 1 coeur
- RAM: 1024Mo
- Stockage: 8Go

Prérequis logiciels :

- Debian 12
- Apache2
- MariaDB
- PHP 8.4
- Extensions PHP nécessaires

### 4. Architecture cible

Composant	Description
OS	Debian 12
Web Server	Apache2
Base de données	MariaDB
Application	GLPI
Accès	Interface Web

## 5. Procédure d'installation

Couleur rouge = Commande

Couleur verte = Texte de configuration dans un fichier

Couleur noir = Texte explicatif

### 5.1 Mise à jour du système

Commandes :

```
apt update && apt upgrade -y
```

### 5.2 Installation des dépendances

Commandes :

```
apt-get install apache2 php8.4-fpm mariadb-server
```

```
apt install php8.4-{curl,gd,intl,mysql,zip,bcmath,mbstring,xml,bz2}
```

.....  
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 install php8.4-ldap
```

.....

### 5.3 Configuration de la base de données

Commandes :

## mariadb-secure-installation

# Si vous utilisez MySQL : [mysql\\_secure\\_installation](#)

```
adm_fb@SRV-GLPI11:~$ sudo mariadb-secure-installation
NOTE: MariaDB is secure by default in Debian. Running this script is
useless at best, and misleading at worst. This script will be
removed in a future MariaDB release in Debian. Please read
mariadb-server_README.Debian for details.

Enter root user password or leave blank:
Enter current password for root (enter for none):
OK, successfully used password, moving on ...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] n ←
... skipping.

You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] y ←
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y ←
SQL executed without errors!
The operation might have been successful, or it might have not done anything.

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y ←
SQL executed without errors!
The operation might have been successful, or it might have not done anything.

By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y ←
- Dropping test database ...
SQL executed without errors!
The operation might have been successful, or it might have not done anything.
- Removing privileges on test database ...
SQL executed without errors!
The operation might have been successful, or it might have not done anything.

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y ←
... Success!

Cleaning up ...

All done! If you've completed all of the above steps, your MariaDB
```

`mysql -u root -p`

`CREATE DATABASE db25_glpi;`

`GRANT ALL PRIVILEGES ON db25_glpi.* TO glpi_adm@localhost IDENTIFIED BY`

`"MotDePasseRobuste";`

`FLUSH PRIVILEGES;`

`EXIT`

## 5.4 Installation de l'application

Commandes :

`cd /tmp`

`wget https://github.com/glpi-project/glpi/releases/download/11.0.6/glpi-11.0.6.tgz`

`tar -xzvf glpi-11.0.6.tgz -C /var/www/`

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

## Etape intermédiaire (Bonne pratique) # non obligatoire

```
mkdir /etc/glpi
chown www-data /etc/glpi/
mv /var/www/glpi/config /etc/glpi
mkdir /var/lib/glpi
chown www-data /var/lib/glpi/
mv /var/www/glpi/files /var/lib/glpi
mkdir /var/log/glpi
chown www-data /var/log/glpi
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 créer le deuxième fichier

```
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**. Pour rappel, ces deux répertoires ont été préparés précédemment.

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

## 5.5 Configuration du serveur web

Commandes :

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

```
<VirtualHost *:80> ServerName GLPIServer 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 # Ensure
authorization headers are passed to PHP. # Some Apache configurations may filter them
and break usage of API, CalDAV, ... RewriteCond %{HTTP:Authorization} ^(.+)$ RewriteRule
.* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}] # Redirect all requests to GLPI
router, unless file exists. RewriteCond %{REQUEST_FILENAME} !-f RewriteRule ^{.*}$
```

```
index.php [QSA,L] </Directory> </VirtualHost>
```

Qui nous donne ça :

```
<VirtualHost *:80>
    ServerName GLPIServer

    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

        # Ensure authorization headers are passed to PHP.
        # Some Apache configurations may filter them and break usage of API, CalDAV, ...
        RewriteCond %{HTTP:Authorization} ^(.*)$
        RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]

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

    </Directory>
<FilesMatch \.php$>
    SetHandler "proxy:unix:/run/php/php8.4-fpm.sock|fcgi://localhost/"
</FilesMatch>
</VirtualHost>
```

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

```
a2ensite glpi.conf
```

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

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

Nous allons aussi **activer le module rewrite** (nécessaire pour activer la prise en charge des règles de réécriture) car nous l'avons utilisé dans le fichier de configuration du VirtualHost (via les directives **RewriteCond / RewriteRule**).

```
a2enmod rewrite
```

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

```
systemctl restart apache2
```

```
a2enmod proxy_fcgi setenvif
```

```
a2enconf php8.4-fpm
```

```
systemctl reload apache2
```

Pour **configurer PHP-FPM pour Apache2**, nous n'allons pas éditer le fichier `/etc/php/8.4/apache2/php.ini`. À la place, le fichier que vous devez éditer est celui-ci :

```
nano /etc/php/8.4/fpm/php.ini
```

Dans ce fichier, recherchez l'option **session.cookie\_httponly** (CTRL+W avec nano) 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
```

Toujours pour renforcer la sécurité, configurez la directive **session.cookie\_samesite** avec la valeur **Lax** conseillée par la documentation de GLPI. Elle contrôle la façon dont le navigateur envoie le cookie de session, ce qui bloque notamment certaines **attaques CSRF** (Cross-Site Request Forgery).

```
; Add SameSite attribute to cookie to help mitigate Cross-Site Request Forgery (CSRF/XSRF)  
; Current valid values are "Strict", "Lax" or "None". When using "None",  
; make sure to include the quotes, as `none` is interpreted like `false` in ini files.  
; https://tools.ietf.org/html/draft-west-first-party-cookies-07  
session.cookie_samesite = Lax
```

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 (**upload\_max\_filesize** limité à 2 Mo par défaut), etc. Je pense aussi à la directive **session.cookie\_secure** que nous passerons sur **on** une fois GLPI accessible en HTTPS.

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

```
systemctl restart php8.4-fpm.service
```

Pour finir, nous devons **modifier notre VirtualHost** pour préciser à Apache2 que PHP-FPM doit être utilisé pour les fichiers PHP. Cette configuration vise à pointer vers le socket de PHP-FPM pour traiter les fichiers avec l'extension **.php**. Editez le fichier **glpi.conf** pour ajouter ceci :

```
<FilesMatch \.php$>  
    SetHandler "proxy:unix:/run/php/php8.4-fpm.sock|fcgi://localhost/"  
</FilesMatch>
```

Voici un exemple :

```

<VirtualHost *:80>
    ServerName GLPIServer

    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

        # Ensure authorization headers are passed to PHP.
        # Some Apache configurations may filter them and break usage of API, CalDAV, ...
        RewriteCond %{HTTP:Authorization} ^(.*)$
        RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}]

        # Redirect all requests to GLPI router, unless file exists.
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ index.php [QSA,L]
    </Directory>
<FilesMatch \.php$>
    SetHandler "proxy:unix:/run/php/php8.4-fpm.sock|fcgi://localhost/"
</FilesMatch>
</VirtualHost>

```

à placer en dessous de Directory.

Pour finir Relancer Apache2 :

`systemctl restart apache2`

## Faut-il séparer les répertoires "files", "config" et "logs" de GLPI ?

Oui, cela est recommandé en production. Stocker les répertoires files/, config/ et logs/ hors du répertoire Web (par exemple sous /var/lib/glpi comme l'explique ce tutoriel) renforce la sécurité. Cette approche facilite aussi les mises à jour.

## Comment augmenter la taille des uploads de fichiers sur GLPI ?

GLPI s'appuie directement sur la configuration PHP pour déterminer la taille maximale des fichiers pouvant être téléversés (pièces jointes, documents, images). Bien souvent, il est nécessaire d'augmenter la taille, car la valeur par défaut (2M) est contraignante pour les pièces jointes des tickets.

Pour augmenter cette limite, il faut modifier votre fichier **php.ini** (dans notre cas : /etc/php/8.4/apache2/php.ini). Voici un exemple pour adapter la limite à 10 Mo : **upload\_max\_filesize = 10M**. Adaptez aussi cette valeur : **post\_max\_size = 10M**. Redémarrez le service PHP-FPM.

Si cela ne fonctionne toujours pas, vérifiez qu'il n'y ait pas une restriction au sein même de GLPI, à cet endroit : **Configuration > Générale > Gestion > Configuration des documents**.

## 6. Installation via l'interface web

URL d'accès : `http://IP_DU_SERVEUR/glpi`

Étapes :

1. Connexion à la base de données avec l'utilisateur créé
2. Sélectionner notre base de données
3. Se connecter avec l'utilisateur par défaut glpi / glpi (à changer par la suite)

## 7. Vérification du fonctionnement

Tests à réaliser :

- Accès interface web
- Connexion administrateur
- Test de création

Même si l'installation est terminée, nous avons encore quelques **actions indispensables** 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 (possibilité de relancer l'installation)**

```
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...).

## 8. Sécurisation

- Modifier les mots de passe par défaut
- Activer HTTPS
- Mettre en place des sauvegardes

## 9. Sauvegarde

Sauvegarde base de données : `mysqldump -u root -p db25_glpi > backup_glpi.sql`

Sauvegarde fichiers : `tar -czf glpi_backup.tar.gz /var/www/glpi`

## **10. Conclusion**

L'installation de GLPI est désormais terminée.

Le serveur peut maintenant être utilisé pour la gestion des tickets et du parc informatique.