Installer et Configurer GLPI 12

1. Présentation et prérequis

Architecture

Élément	Choix
Système	Debian 13 « Trixie »
Serveur web	Apache 2.4 avec PHP-FPM
Base de données	MariaDB
Application	GLPI 12.0.0
Usage E6	Gestion des incidents et gestion des configurations

GLPI 12 exige PHP 8.3 ou plus récent. Debian 11 et 12 n'ont que PHP 7.4 et 8.2, donc elles ne conviennent pas. Si votre sandbox impose Debian 11, utilisez GLPI 10, ou une VM Debian 13.

Un retour d'installation de GLPI 12.0.0 sur Debian 13 mentionne PHP 8.4, Apache 2.4 et MariaDB 11.8. La version 12.0.1 corrige des failles de sécurité ; remplacez `VERSION` dans les commandes pour l'utiliser.

Matériel conseillé pour la VM

- 2 vCPU, 4 Go de RAM, 30 Go de disque.
- Une adresse IP fixe.
- Un accès internet pendant l'installation.

2. Préparer le système

```
sudo apt update && sudo apt upgrade -y
sudo hostnamectl set-hostname glpi
sudo timedatectl set-timezone Europe/Paris
cat /etc/os-release
```

Installez les paquets. Les extensions obligatoires pour GLPI 11 et plus incluent `curl`, `gd`, `intl`, `mysqli`, `bcmath`, `mbstring` et `openssl`. `ldap` sert à l'authentification Active Directory, `zip` et `bz2` au marketplace.

```
sudo apt install -y apache2 mariadb-server wget tar \
  php php-cli php-fpm libapache2-mod-fcgid \
  php-curl php-gd php-intl php-mysql php-xml php-mbstring \
  php-bcmath php-zip php-bz2 php-ldap php-opcache php-apcu

php -v
```

La version affichée doit être 8.3 ou supérieure.

3. Installer et sécuriser MariaDB

```
sudo systemctl enable --now mariadb
sudo mysql_secure_installation
```

Créez la base et l'utilisateur, après avoir remplacé `MOT_DE_PASSE_GLPI` :

```
sudo mysql
```

```
CREATE DATABASE glpi CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci;
CREATE USER 'glpi'@'localhost' IDENTIFIED BY 'MOT_DE_PASSE_GLPI';
GRANT ALL PRIVILEGES ON glpi.* TO 'glpi'@'localhost';
GRANT SELECT ON mysql.time_zone_name TO 'glpi'@'localhost';
FLUSH PRIVILEGES;
EXIT;
```

Chargez les fuseaux horaires :

```
mariadb-tzinfo-to-sql /usr/share/zoneinfo | sudo mysql mysql
```

Vérifiez la connexion avec le mot de passe défini :

```
mysql -u glpi -p glpi -e "SELECT 1;"
```

4. Télécharger GLPI

Utilisez l'archive `glpi-VERSION.tgz` des releases, pas les archives « Source code » de GitHub.

```
cd /tmp
VERSION=12.0.0
wget https://github.com/glpi-project/glpi/releases/download/${VERSION}/glpi-${VERSION}.tgz
sudo tar -xzf glpi-${VERSION}.tgz -C /var/www/
ls /var/www/glpi
```

5. Séparer code, configuration et données

La documentation recommande de placer la configuration, les fichiers de données et les journaux hors du code, selon la norme FHS.

```
sudo mkdir -p /etc/glpi /var/lib/glpi /var/log/glpi
sudo cp -a /var/www/glpi/config /etc/glpi/
sudo cp -a /var/www/glpi/files /var/lib/glpi/
```

Créez `/var/www/glpi/inc/downstream.php` :

```
<?php
define('GLPI_CONFIG_DIR', '/etc/glpi/');

if (file_exists(GLPI_CONFIG_DIR . '/local_define.php')) {
   require_once GLPI_CONFIG_DIR . '/local_define.php';
}
```

Créez `/etc/glpi/local_define.php` :

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

Appliquez les droits :

```
sudo chown -R root:root /var/www/glpi
sudo chown -R www-data:www-data /etc/glpi /var/lib/glpi /var/log/glpi
sudo chown -R www-data:www-data /var/www/glpi/marketplace
sudo find /var/www/glpi -type d -exec chmod 755 {} \;
sudo find /var/www/glpi -type f -exec chmod 644 {} \;
```

6. Configurer Apache et PHP

Le `DocumentRoot` doit pointer vers le dossier `public`, pas vers la racine de GLPI.

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

```
<VirtualHost *:80>
    ServerName glpi.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>

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

Si `php-fpm.sock` n'apparaît pas dans `/run/php/`, utilisez le nom réel du socket, par exemple `php8.4-fpm.sock`.

Activez les modules et le site :

```
sudo a2enmod rewrite proxy_fcgi setenvif
sudo a2dissite 000-default
sudo a2ensite glpi
sudo apache2ctl configtest
```

Sécurisez les sessions PHP dans le `php.ini` de PHP-FPM, en remplaçant `8.4` par votre version :

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

```
session.cookie_httponly = On
session.cookie_samesite = Lax
; session.cookie_secure = On   (uniquement après l'activation du HTTPS)
```

```
sudo systemctl restart php8.4-fpm apache2
sudo systemctl enable php8.4-fpm apache2
```

7. Terminer l'installation

Par l'assistant web

Ouvrez `http://IP_DU_SERVEUR/`, puis :

1. Choisissez la langue et acceptez la licence.
2. Cliquez sur **Installer** et corrigez tout prérequis en rouge.
3. Renseignez la base : serveur `localhost`, utilisateur `glpi`, mot de passe `MOT_DE_PASSE_GLPI` .
4. Sélectionnez la base `glpi`.
5. Terminez l'installation.

Par la ligne de commande

```
cd /var/www/glpi
sudo -u www-data php bin/console db:install \
  --db-host=localhost --db-name=glpi \
  --db-user=glpi --db-password='MOT_DE_PASSE_GLPI' \
  --default-language=fr_FR
```

8. Comptes par défaut

L'installation crée quatre comptes : `glpi/glpi` administrateur, `tech/tech` technicien, `normal/normal` utilisateur normal et `post-only/postonly` utilisateur limité aux tickets.

Après la première connexion :

1. Créez un nouvel utilisateur avec le profil `super-admin`.
2. Changez le mot de passe de `glpi`, ou désactivez ce compte.
3. Supprimez ou désactivez `tech`, `normal` et `post-only`, ou changez leurs mots de passe.
4. Ne touchez pas à `glpi-system`, utilisé par les fonctions automatiques.

Supprimez aussi le script d'installation s'il est resté présent :

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

9. Tâches planifiées et HTTPS

```
sudo crontab -u www-data -e
```

```
* * * * * /usr/bin/php /var/www/glpi/front/cron.php &>/dev/null
```

Pour une sandbox, créez un certificat auto-signé :

```
sudo a2enmod ssl headers
sudo mkdir -p /etc/ssl/glpi
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/ssl/glpi/glpi.key -out /etc/ssl/glpi/glpi.crt \
  -subj "/CN=glpi.local"
```

Ajoutez un second `VirtualHost *:443` qui reprend la configuration précédente avec :

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

Redirigez ensuite le port 80 vers le HTTPS, puis activez `session.cookie_secure = On` et redémarrez PHP-FPM et Apache.

10. Configuration pour l'E6

Organisation

- Créez les entités, les profils, les groupes de techniciens et quelques utilisateurs de test.
- Configurez les catégories de tickets et les délais de prise en charge.

Authentification Active Directory

Dans la liste de votre épreuve, l'annuaire est un Active Directory sous Windows Server 2019.

- Allez dans Configuration > Authentification > Annuaire LDAP.
- Renseignez le serveur AD, le DN de base et un compte de lecture dédié.
- Testez la connexion, puis importez des utilisateurs.

Inventaire et gestion des configurations

- Activez l'inventaire natif dans Administration > Inventaire.
- Installez l'agent GLPI sur un poste Windows et un serveur Linux.
- Vérifiez que les machines apparaissent dans le parc.
- Documentez les logiciels, versions et propriétaires.

Gestion des incidents

- Configurez l'e-mail sortant pour les notifications.
- Créez un ticket de test, assignez-le, résolvez-le, puis documentez la solution.
- Associez le ticket à un équipement de l'inventaire.

Lien avec Zabbix

Simulez un service arrêté sur une machine supervisée : Zabbix déclenche l'alerte, vous ouvrez le ticket dans GLPI, puis vous le résolvez. Cela montre la supervision, l'incident et la configuration avec deux outils.

11. Sauvegarde

```
sudo mkdir -p /root/backups-glpi
sudo mysqldump --single-transaction --routines --triggers glpi \
  > "/root/backups-glpi/glpi-$(date +%F).sql"
sudo tar -czf "/root/backups-glpi/glpi-files-$(date +%F).tar.gz" \
  /etc/glpi /var/lib/glpi /var/www/glpi/marketplace
```

Si vous utilisez PBS, sauvegardez la VM entière sur un stockage séparé de l'hôte Proxmox, puis testez une restauration.

12. Validation et dépannage

Vérification	Commande ou action
Services actifs	`systemctl is-active apache2 mariadb php8.4-fpm`
Syntaxe Apache	`sudo apache2ctl configtest`
Journal Apache	`sudo tail -f /var/log/apache2/error.log`
Journal GLPI	`sudo tail -f /var/log/glpi/php-errors.log`
Cron GLPI	Page Configuration > Actions automatiques

Symptôme	Piste
Page Apache par défaut	`a2ensite glpi` et `a2dissite 000-default`
Erreur 500	Journal Apache, extension PHP manquante ou droits
Prérequis rouges	Installez le paquet `php-...` correspondant
Erreur d'écriture	Propriétaire `www-data` sur `/etc/glpi`, `/var/lib/glpi`, `/var/log/glpi`
Fuseaux horaires	Relancez `mariadb-tzinfo-to-sql` et le `GRANT SELECT`

Revision #1
Created 2026-11-04 14:14:37 UTC by Yoann Trevette
Updated 2026-11-04 14:15:58 UTC by Yoann Trevette