

# Documentation d'installation GLPI

## Prérequis pour l'installation

- Un server Linux Ubuntu Server
- Un serveur web (nous utiliserons ici Apache)
- Un interpréteur PHP
- Un SGBD (nous utiliserons ici MariaDB)

## 1 – Installation des composants

Avant l'installation, nous devons mettre à jour le serveur :

```
sudo apt update && apt upgrade
```

Ensuite, nous installons Apache, PHP et ses dépendances ainsi que MariaDB :

```
sudo apt install -y apache2 php php-  
{apcu,cli,common,curl,gd,imap,ldap,mysql,xmldrpc,xmldbstring,bcmath,intl,zip,r  
edis,bz2} libapache2-mod-php php-soap php-cas mariadb-server
```

## 2 - Configuration de la base de données

MariaDB, par défaut, ne possède pas de mot de passe associé au compte root. De plus, il possède une base de données de test, ainsi qu'une configuration de privilèges qu'il convient d'adapter à nos besoins. Le mieux est de créer un nouvel utilisateur MariaDB, une nouvelle base de données et d'associer les privilèges de cette base au nouveau compte.

Secure MariaDB Installation :

```
mysql_secure_installation
```

**Recommandations minimales :**

- Changer le mot de passe root
- Supprimer les utilisateurs anonymes

- Interdire la connexion en root à distance
- Supprimer la base de données de test
- Mettre à jour les privilèges

Comme GLPI peut être utilisé par plusieurs personnes dans le monde au même moment, il serait intéressant d'enregistrer la time zone de l'utilisateur pour l'heure et la date de l'application :

```
mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql mysql
```

Créons un utilisateur pour la base de données GLPI :

```
mysql -uroot -pmysql
```

```
CREATE DATABASE glpi;
CREATE USER 'glpi'@'localhost' IDENTIFIED BY 'yourstrongpassword';
GRANT ALL PRIVILEGES ON glpi.* TO 'glpi'@'localhost';
GRANT SELECT ON 'mysql'.'time_zone_name' TO 'glpi'@'localhost';
FLUSH PRIVILEGES;
```

### 3 - Préparation pour l'installation de GLPI

Une fois la base de données créée, nous devons installer les fichiers de GLPI. Comme nous souhaitons installer la dernière version, nous allons récupérer les fichiers directement depuis le répertoire Github :

```
cd /var/www/html;
wget https://github.com/glpi-project/glpi/releases/download/[version]/glpi-[version].tgz;
tar -xvzf glpi-[version].tgz;
```

Une fois décompressé, les fichiers se trouvent dans les répertoires suivants :

- **/var/www/html/glpi/config** : pour les fichiers de configuration
- **/var/www/html/glpi/files** : pour le code source
- **/var/lib/glpi** : pour les fichiers variables
- **/var/lib/glpi/\_log** : pour les fichiers de log

Cependant, nous aimerions stocker les informations de GLPI dans les dossiers suivants, afin de rendre la hiérarchie de dossiers plus claire :

- **/etc/glpi** : pour les fichiers de configuration
- **/var/www/html/glpi** : pour le code source
- **/var/lib/glpi** : pour les fichiers variables
- **/var/log/glpi** : pour les fichiers de log

Pour cela, nous devons indiquer à l'application où se trouve ces répertoires, et notamment le dossier des fichiers de configuration. Pour cela, nous allons créer le fichier `downstream.php` (GLPI reconnaît ce nom de fichier) pour lui indiquer où se situe le répertoire de configuration.

Créons le fichier `downstream.php` :

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

Déclarons maintenant le chemin vers `'/etc/glpi/` :

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

Nous pouvons maintenant déplacer nos fichiers vers les nouveaux chemins d'accès :

```
mv /var/www/html/glpi/config /etc/glpi;
mv /var/www/html/glpi/files /var/lib/glpi;
mv /var/lib/glpi/_log /var/log/glpi;
```

Une fois le nouveau `"GLPI_CONFIG_DIR"` créé avec `downstream.php`, on peut créer un autre fichier, `"local_define.php"` dans le répertoire **/etc/glpi** pour indiquer à GLPI tous les autres dossiers qu'il devra utiliser :

Créons `local_define.php` :

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

Déclarons maintenant les chemins d'accès :

```
<?php
define('GLPI_VAR_DIR', '/var/lib/glpi');
define('GLPI_DOC_DIR', GLPI_VAR_DIR);
define('GLPI_CRON_DIR', GLPI_VAR_DIR . '/_cron');
define('GLPI_DUMP_DIR', GLPI_VAR_DIR . '/_dumps');
define('GLPI_GRAPH_DIR', GLPI_VAR_DIR . '/_graphs');
define('GLPI_LOCK_DIR', GLPI_VAR_DIR . '/_lock');
define('GLPI_PICTURE_DIR', GLPI_VAR_DIR . '/_pictures');
define('GLPI_PLUGIN_DOC_DIR', GLPI_VAR_DIR . '/_plugins');
define('GLPI_RSS_DIR', GLPI_VAR_DIR . '/_rss');
define('GLPI_SESSION_DIR', GLPI_VAR_DIR . '/_sessions');
define('GLPI_TMP_DIR', GLPI_VAR_DIR . '/_tmp');
define('GLPI_UPLOAD_DIR', GLPI_VAR_DIR . '/_uploads');
define('GLPI_CACHE_DIR', GLPI_VAR_DIR . '/_cache');
define('GLPI_LOG_DIR', '/var/log/glpi');
```

## 4 - Permissions d'accès pour les dossiers et fichiers

Maintenant que tous les dossiers pour GLPI ont été configurés, il faut maintenant gérer les permissions. Voici une propositions des permissions nécessaires :

```
chown root:root /var/www/html/glpi/ -R;
chown www-data:www-data /etc/glpi -R;
chown www-data:www-data /var/lib/glpi -R;
chown www-data:www-data /var/log/glpi -R;
chown www-data:www-data /var/www/html/glpi/marketplace -Rf;
find /var/www/html/glpi/ -type f -exec chmod 0644 {};
find /var/www/html/glpi/ -type d -exec chmod 0755 {};
find /etc/glpi -type f -exec chmod 0644 {};
find /etc/glpi -type d -exec chmod 0755 {};
find /var/lib/glpi -type f -exec chmod 0644 {};
find /var/lib/glpi -type d -exec chmod 0755 {};
find /var/log/glpi -type f -exec chmod 0644 {};
find /var/log/glpi -type d -exec chmod 0755 {};
```

## 5 - Configurer le serveur web

Pour que GLPI fonctionne correctement, il est recommandé de créer un VirtualHost et une zone DNS pour simplifier les URLs. Pour cela, nous allons éditer créer le fichier glpi.conf.

Créons le fichier glpi.conf :

```
# emplacement à modifier en fonction des exigences du serveur
nano /etc/apache2/sites-available/glpi.conf
```

Créons maintenant le Virtual Host :

```
# Start of the VirtualHost configuration for port 80
<VirtualHost *:80>
    ServerName yourglpi.yourdomain.com
    # Specify the server's hostname
    DocumentRoot /var/www/html/glpi/public
    # The directory where the website's files are located
    # Start of a Directory directive for the website's directory
    <Directory /var/www/html/glpi/public>
        Require all granted
        # Allow all access to this directory
        RewriteEngine On
        # Enable the Apache rewrite engine
        # 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 the file
exists.

        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ index.php [QSA,L]
    </Directory>
    # End of the Directory directive for /var/www/glpi/public
</VirtualHost>
# End of the VirtualHost configuration for port 80
```

Informations à changer :

**ServerName** si vous avez une URL publique.

**DocumentRoot** si vous allez placer GLPI sur une page différente.

Une fois le VirtualHost créé, vous devez désactiver la configuration par défaut du site Apache, activer le module de réécriture et recharger le nouveau fichier :

```
a2dissite 000-default.conf; # Disable default apache site
a2enmod rewrite; # enable the rewrite module
a2ensite glpi.conf; # enable the new apache virtual host settings for your
glpi instance
systemctl restart apache2;
```

Pour que GLPI fonctionne correctement, il est recommandé de modifier les paramètres suivants dans votre fichier php.ini.

Ouvrons le fichier php.ini :

```
nano /etc/php/[version_php]/apache2/php.ini
```

Changeons les paramètres suivants :

- upload\_max\_filesize = 20M Maximum size for uploaded files is set to 20 megabytes.
- post\_max\_size = 20M Maximum size for POST data (e.g., form submissions) is also set to 20 megabytes.
- max\_execution\_time = 60 Maximum execution time for a PHP script is set to 60 seconds.
- max\_input\_vars = 5000 Maximum number of input variables (e.g., form fields) a script can accept is 5000.
- memory\_limit = 256M The maximum amount of memory a single PHP script can use is 256 megabytes.
- session.cookie\_httponly = On Sets the "HttpOnly" attribute for session cookies
- date.timezone = America/Sao\_Paulo Sets the default timezone for PHP to yours.

Pour ajouter son fuseau horaire, il faut se référer à la liste officielle des [fuseaux horaires supportés par PHP](#)

## 6 - Start Web Installation

Une fois terminé, nous pouvons ouvrir un navigateur web, saisir l'URL du serveur, cliquer sur le lien dans la page Apache et suivre les instructions d'installation de GLPI.