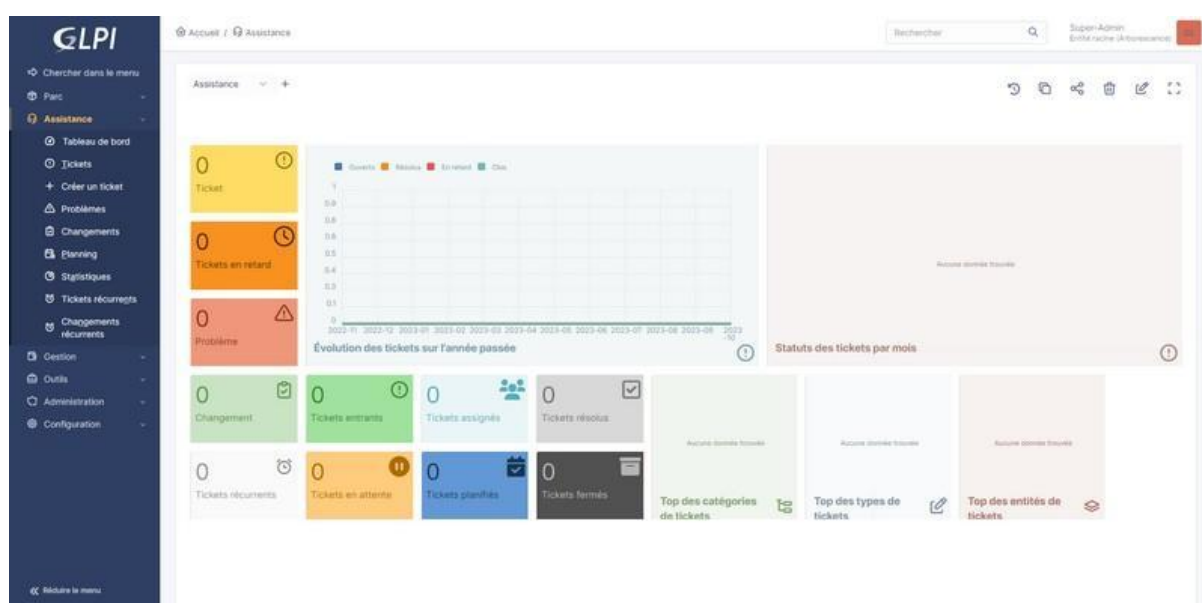


I. Présentation

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.

- [Site officiel de 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 :

```

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
... Success!

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
... Success!

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...
... Success!
- Removing privileges on test database...
... Success!

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
installation should now be secure.

Thanks for using MariaDB!

```

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 "Nom de la base de donnée"** 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
```

Ce qui donne :

```
Enter password:  
Welcome to the MariaDB monitor.  Commands end with ; or \g.  
Your MariaDB connection id is 41  
Server version: 10.11.3-MariaDB-1 Debian 12  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> CREATE DATABASE db23_glpi;  
Query OK, 1 row affected (0.001 sec)  
  
MariaDB [(none)]> GRANT ALL PRIVILEGES ON db23_glpi.* TO glpi_adm@localhost IDENTIFIED BY  
Query OK, 0 rows affected (0.003 sec)  
  
MariaDB [(none)]> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.001 sec)  
  
MariaDB [(none)]> EXIT  
Bye  
glpi_adm@SRV-GLPI:~$
```

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/
```

Puis, nous allons déplacer le répertoire "config" de GLPI vers ce nouveau dossier :

```
sudo mv /var/www/glpi/config /etc/glpi
```

■ Le répertoire /var/lib/glpi

Répetons la même opération avec la création du répertoire **"/var/lib/glpi"** :

```
sudo mkdir /var/lib/glpi  
sudo chown www-data /var/lib/glpi/
```

Dans lequel nous déplaçons également le dossier "**files**" qui contient la majorité des fichiers de GLPI : CSS, plugins, etc.

```
sudo mv /var/www/glpi/files /var/lib/glpi
```

■ Le répertoire `/var/log/glpi`

Terminons par la création du répertoire "`/var/log/glpi`" destiné à stocker les journaux de GLPI. Toujours sur le même principe :

```
sudo mkdir /var/log/glpi
```

```
sudo chown www-data /var/log/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

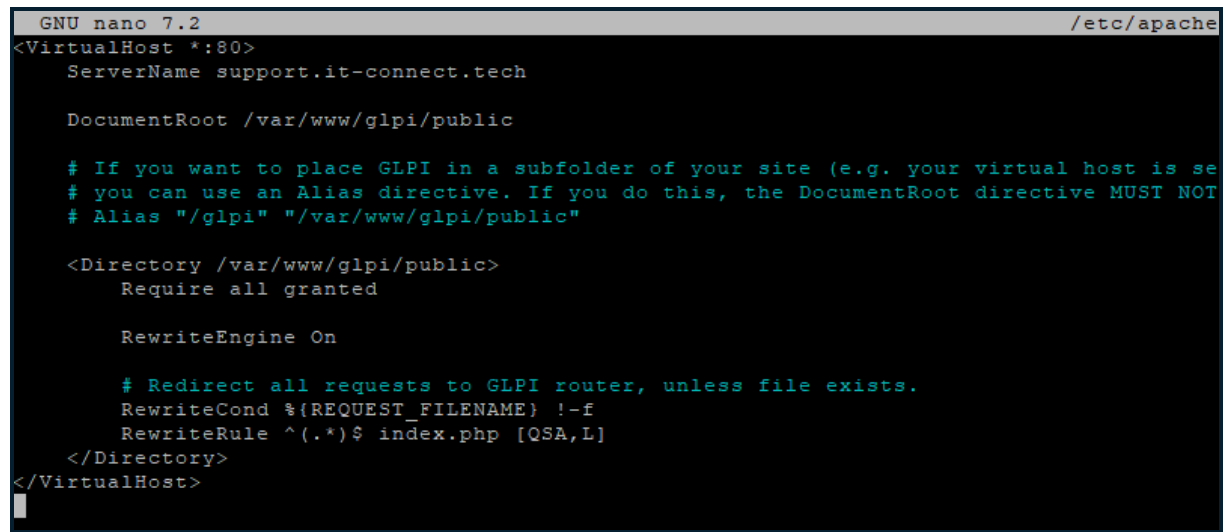
    # If you want to place GLPI in a subfolder of your site (e.g. your
    virtual host is serving multiple applications),
    # you can use an Alias directive. If you do this, the DocumentRoot
    directive MUST NOT target the GLPI directory itself.
    # Alias "/glpi" "/var/www/glpi/public"

    <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>
```



```
GNU nano 7.2 /etc/apache2
<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 se
  # you can use an Alias directive. If you do this, the DocumentRoot directive MUST NOT
  # 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 veillez 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 :



Même connecté sur le profil root il faut mettre sudo devant les 3

commandes suivantes pour qu'elles fonctionnent

```
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.

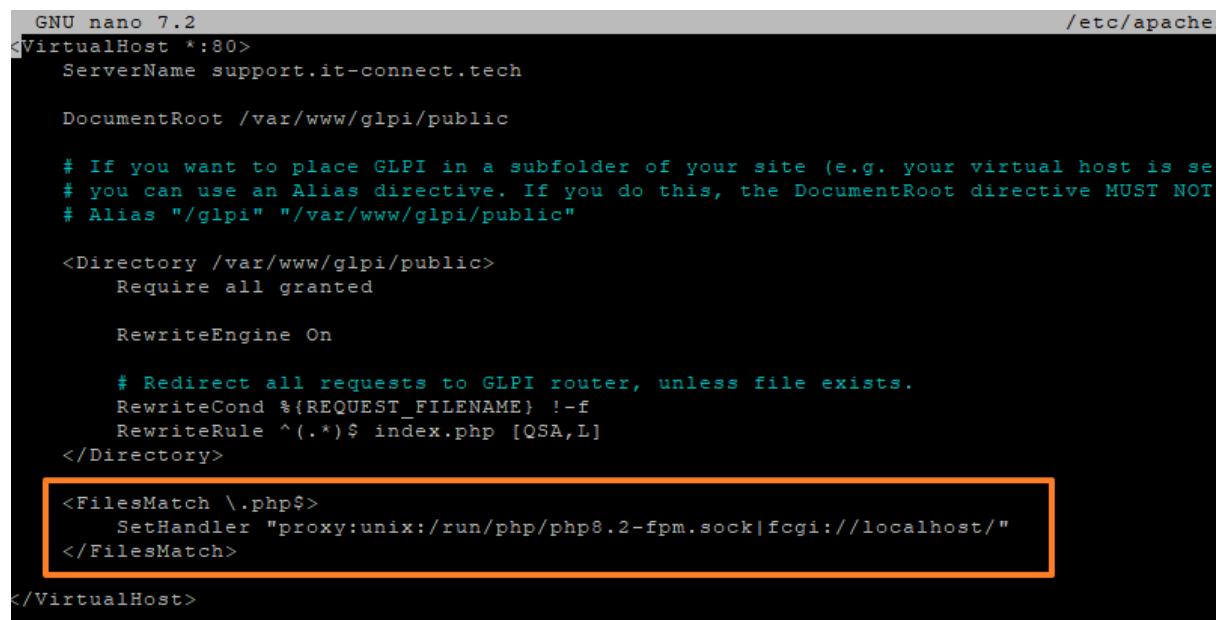
Pour appliquer les modifications, nous devons redémarrer PHP-FPM :

```
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>
```

Voici un exemple :



```
GNU nano 7.2 /etc/apache2  
<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 se  
    # you can use an Alias directive. If you do this, the DocumentRoot directive MUST NOT  
    # 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>  
  
    <FilesMatch \.php$>  
        SetHandler "proxy:unix:/run/php/php8.2-fpm.sock|fcgi://localhost/"  
    </FilesMatch>  
  
</VirtualHost>
```

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.



GLPI SETUP

Étape 0

Vérification de la compatibilité de votre environnement avec l'exécution de GLPI

TESTS EFFECTUÉS	RÉSULTATS
Requis Parser PHP	✓
Requis Configuration des sessions	✓
Requis Mémoire allouée	✓
Requis mysqli extension	✓
Requis Extensions du noyau de PHP	✓
Requis curl extension <i>Requis pour l'accès à distance aux ressources (requêtes des agents d'inventaire, Marketplace, flux RSS, ...).</i>	✓
Requis gd extension <i>Requis pour le traitement des images.</i>	✓
Requis intl extension <i>Requis pour l'internationalisation.</i>	✓
Requis zlib extension <i>Requis pour la gestion de la communication compressée avec les agents d'inventaire, l'installation de paquets gzip à partir du Marketplace et la génération de PDF.</i>	✓
Requis Libsodium ChaCha20-Poly1305 constante de taille <i>Activer l'utilisation du cryptage ChaCha20-Poly1305 requis par GLPI. Il est fourni par libsodium à partir de la version 1.0.12.</i>	✓
Requis Permissions pour les fichiers de log	✓
Requis Permissions pour les dossiers de données	✓
Suggéré Version de PHP supportée <i>Une version officiellement supportée de PHP devrait être utiliser pour bénéficier des correctifs de sécurité et de bogues.</i>	✓
Suggéré Configuration sécurisée du dossier racine du serveur web <i>La configuration du dossier racine du serveur web devrait être `/var/www/glpi/public` pour s'assurer que les fichiers non publics ne peuvent être accessibles.</i>	✓
Suggéré Configuration de sécurité pour les sessions <i>Permet de s'assurer que la sécurité relative aux cookies de session est renforcée.</i>	✓

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.



GLPI SETUP

Étape 1

Configuration de la connexion à la base de données

Serveur SQL (MariaDB ou MySQL)

Utilisateur SQL

Mot de passe SQL

[Continuer >](#)

Après avoir cliqué sur "**Continuer**", nous devons choisir la base de données "**db23_glpi**" créée précédemment.



GLPI SETUP

Étape 2

Test de connexion à la base de données

 Connexion à la base de données réussie

Veuillez sélectionner une base de données :

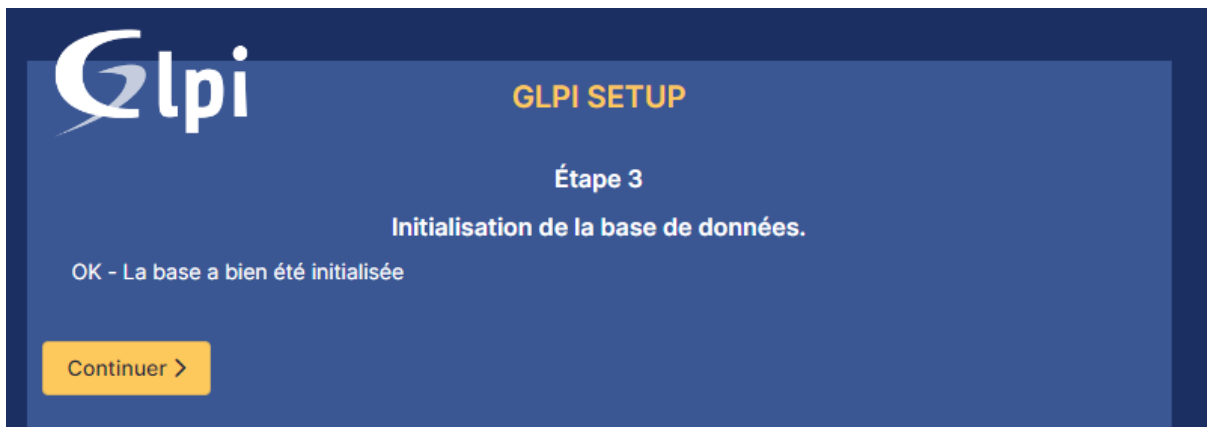
Créer une nouvelle base ou utiliser une base existante :

☐

☒ db23_glpi

[Continuer >](#)

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**" !



GLPI SETUP

Étape 6

L'installation est terminée

Les identifiants et mots de passe par défaut sont :

- glpi/glpi pour le compte administrateur
- tech/tech pour le compte technicien
- normal/normal pour le compte normal
- post-only/postonly pour le compte postonly

Vous pouvez supprimer ou modifier ces comptes ainsi que les données initiales.

 Utiliser GLPI

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



Connexion à votre compte

Identifiant

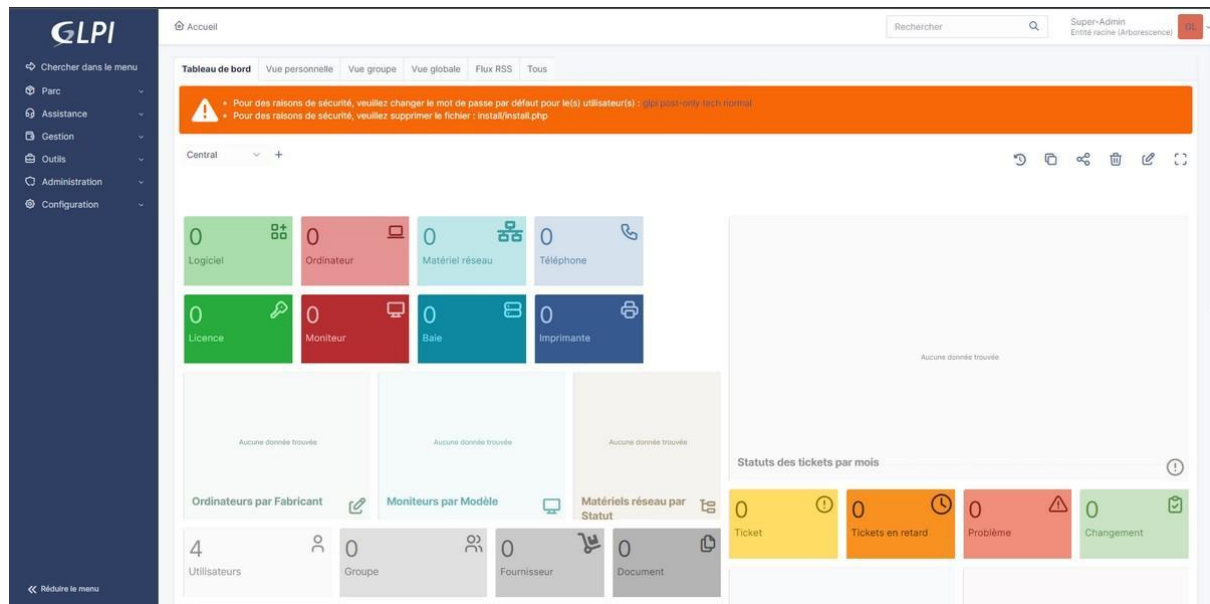
Mot de passe

Source de connexion

☒ Se souvenir de moi

Se connecter

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...).

id : glpi
mdp : Btssio64