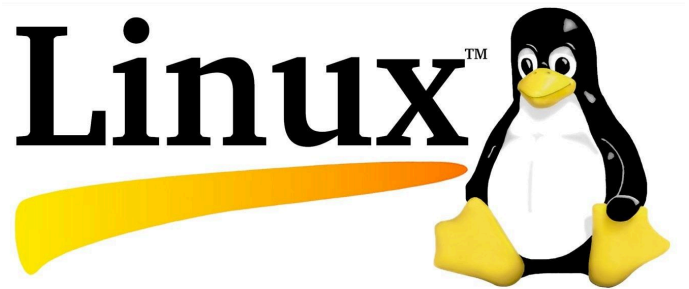


CONFIGURER UN SERVEUR GLPI SUR LINUX



Sommaire

Installation Paquets	4
GLPI	5
MariaDB :	5
Téléchargement GLPI et préparation pour l'installation :	7
Apache2	10
Configuration Serveur Web :	10
PHP8.2-FPM :	12
Installation GLPI	14

Introduction

La société r3st0.fr qui a développé une application web permettant de mettre en ligne les avis des clients à propos des restaurants où ils ont mangé connaît depuis ces derniers mois un succès grandissant. Cette notoriété soudaine a entraîné une expansion exponentielle de ses équipes administrative et informatique.

Le parc de postes de travail et d'équipements actifs (routeurs et commutateurs) a donc grandement augmenté et le DSI (Directeur des Systèmes d'Information) aimerait mettre en place une solution ITSM conforme au référentiel ITIL.

La solution GLPI a été choisie pour répondre à ces différentes problématiques. GLPI est un logiciel libre de gestion des services informatiques et de gestion des services d'assistance.

Il faudra installer le serveur GLPI et ses prérequis (BDD et serveur Web), utiliser GLPI comme inventaire du parc informatique, configurer l'outil avec le contexte ci-dessus (comptes utilisateurs, habilitations, différentes équipes...), configurer des catégories et des modèles pour des tickets récurrents (exemple : demande de matériel informatique, oubli ou verrouillage de compte AD...), définir également des SLAs pour ces types de ticket.

Pour ne pas surcharger la hotline informatique, il faudra donner un accès au portail GLPI directement aux utilisateurs pour qu'ils puissent signaler un incident via leur navigateur sans appeler la hotline. Configurer l'outil GLPI pour que les cadres d'équipes puissent créer des tickets incidents. Ils ne doivent pas être admin de la plateforme et voir seulement leur ticket.

Installation Paquets

Commencer 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`**

La première 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.

Installer ces 3 paquets :

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

Puis 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`**

GLPI

MariaDB :

MariaDB est un système de gestion de base de données relationnelle (SGBDR) open source. Il sert à stocker, organiser, interroger et gérer des données à l'aide du langage SQL.

Il faudra 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 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, créer une base de données dédiée pour GLPI qui sera accessible par un utilisateur dédié. Hors de question d'utiliser le compte root de MariaDB : *une base de données = un utilisateur que vous venez de définir à l'étape précédente.*

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`**

Téléchargement GLPI et préparation pour l'installation :

Cette 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.

Lien : [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`

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/`

Définir l'utilisateur `www-data` correspondant à `Apache2` en tant que propriétaire sur les fichiers GLPI.

Ensuite, créer plusieurs dossiers et sortir des données de la racine Web (`/var/www/glpi`) de manière à les stocker dans les nouveaux dossiers que vous allez 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. Il faut donner 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`

- Le répertoire */var/lib/glpi*

Répétons 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 il faut également déplacer 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*

Terminer 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`**

- Créer les fichiers de configuration

Configurer GLPI pour qu'il sache où aller chercher les données. Autrement dit, il faut déclarer les nouveaux répertoires créés.

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, créer ce second fichier :

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

Afin d'ajouter le contenu ci-dessous permettant de déclarer deux variables qui permet de préciser les chemins vers les répertoires *files* et *log* que vous avez préparés précédemment.

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

Apache2

Apache2 est un serveur web open source qui permet d'héberger et distribuer des sites web ou applications web sur Internet ou sur un réseau local. Il reçoit les requêtes des navigateurs (Chrome, Firefox, etc.) et renvoie les pages web demandées en HTML, images, fichiers CSS, ou exécute des applications dynamiques.

Configuration Serveur Web :

Créer un nouveau fichier de configuration qui va permettre de configurer le *VirtualHost* dédié à GLPI. Dans le cas ici présent, 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) :

```
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>
```

 **N'oubliez pas d'enregistrer le fichier quand la configuration est prête.**

Puis, activer ce nouveau site dans Apache2 :

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

Désactiver le site par défaut car il est inutile :

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

Activer le module *rewrite* (pour les règles de réécriture) car il est utilisé dans le fichier de configuration du VirtualHost (*RewriteCond / RewriteRule*).

➤ **`sudo a2enmod rewrite`**

Pour finir, redémarrer le service Apache2 :

➤ **`sudo systemctl restart apache2`**

PHP8.2-FPM :

PHP est un langage de programmation principalement utilisé pour le développement web côté serveur. Il permet de créer des pages web dynamiques, c'est-à-dire des sites dont le contenu change selon l'utilisateur, la base de données ou les actions réalisées. PHP s'exécute sur le serveur web puis envoie au navigateur le résultat en HTML.

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*).
- Utiliser le module *PHP-FPM*.

PHP-FPM signifie PHP FastCGI Process Manager. C'est un gestionnaire de processus pour PHP qui permet d'exécuter les scripts PHP de manière plus rapide et plus performante. PHP-FPM optimise le traitement des requêtes PHP en maintenant plusieurs processus prêts à répondre, ce qui améliore :

- **les performances ;**
- **la gestion de plusieurs visiteurs simultanés ;**
- **la stabilité du serveur ;**
- **la configuration avancée des ressources.**

Il est donc recommandé d'utiliser PHP-FPM.

- ❖ **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.**

Commencer par installer *PHP8.2-FPM* avec la commande suivante :

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

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, éditer le fichier `/etc/php/8.2/fpm/php.ini` :

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

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, redémarrer PHP-FPM :

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

Pour finir, modifier le `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>
```

```
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`**

Installation GLPI

Pour effectuer l'installation de GLPI, il faut 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. Commencer par choisir la langue.



Puisqu'il s'agit d'une nouvelle installation, cliquer sur *Installer*.



- ❖ **GLPI vérifie la configuration du serveur pour déterminer si tous les prérequis sont respectés. Si certains ne le sont pas, contrôler la partie de la configuration qui pose problème et essayer de le régler en suivant les étapes précédentes de ce guide. Si tout est bon, vous pouvez passer à la prochaine étape.**



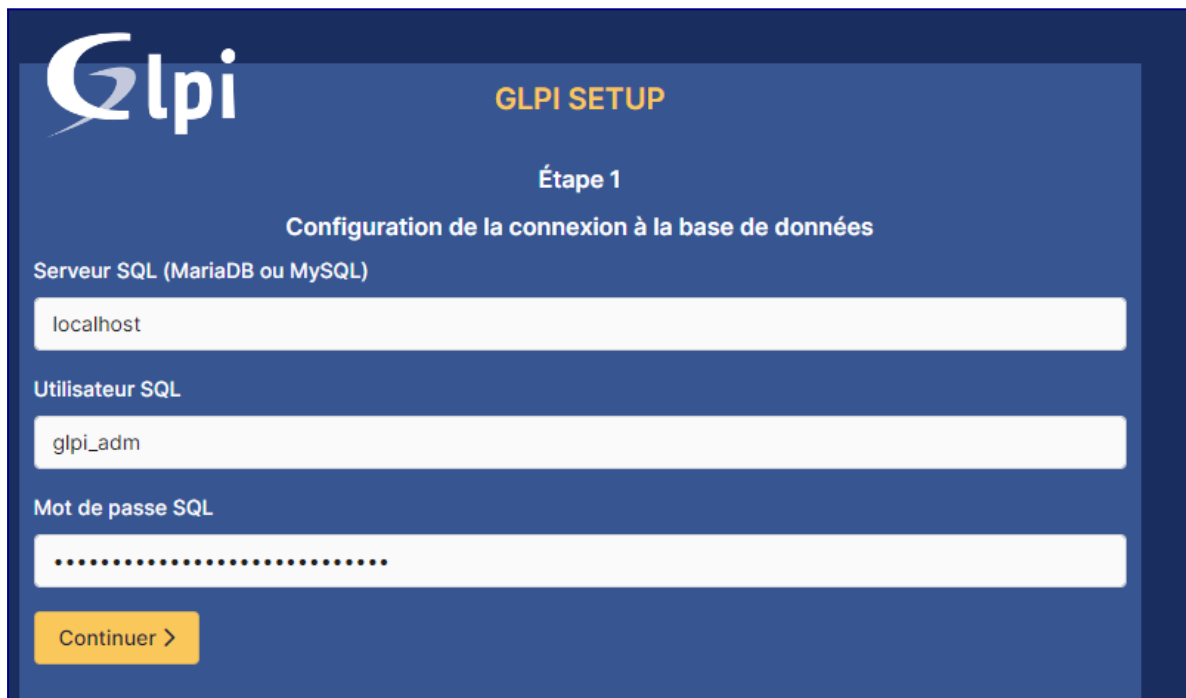
GLPI SETUP

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 <code>`/var/www/glpi/public`</code> 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>	✓

Ensuite, renseigner les informations pour se connecter à la base de données. Indiquer *localhost* en tant que serveur SQL puisque MariaDB est installé en local, sur le même serveur que GLPI. Puis, entrer l'utilisateur *glpi_adm* et le mot de passe associé.



The screenshot shows the 'GLPI SETUP' interface. At the top left is the GLPI logo. The main heading is 'GLPI SETUP' in orange. Below it, 'Étape 1' is centered, followed by 'Configuration de la connexion à la base de données'. There are three input fields: 'Serveur SQL (MariaDB ou MySQL)' with 'localhost', 'Utilisateur SQL' with 'glpi_adm', and 'Mot de passe SQL' with a masked password. A yellow 'Continuer >' button is at the bottom left.

A l'étape 2, choisir la base de données *db23_glpi* créée précédemment. Cliquer sur continuer jusqu'à arriver à l'étape finale qui confirme l'installation de GLPI avec tous les identifiants par défaut.



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

Maintenant que l'installation est finie, connectez vous à GLPI avec le compte administrateur *glpi/glpi*.



Connexion à votre compte

Identifiant

Mot de passe

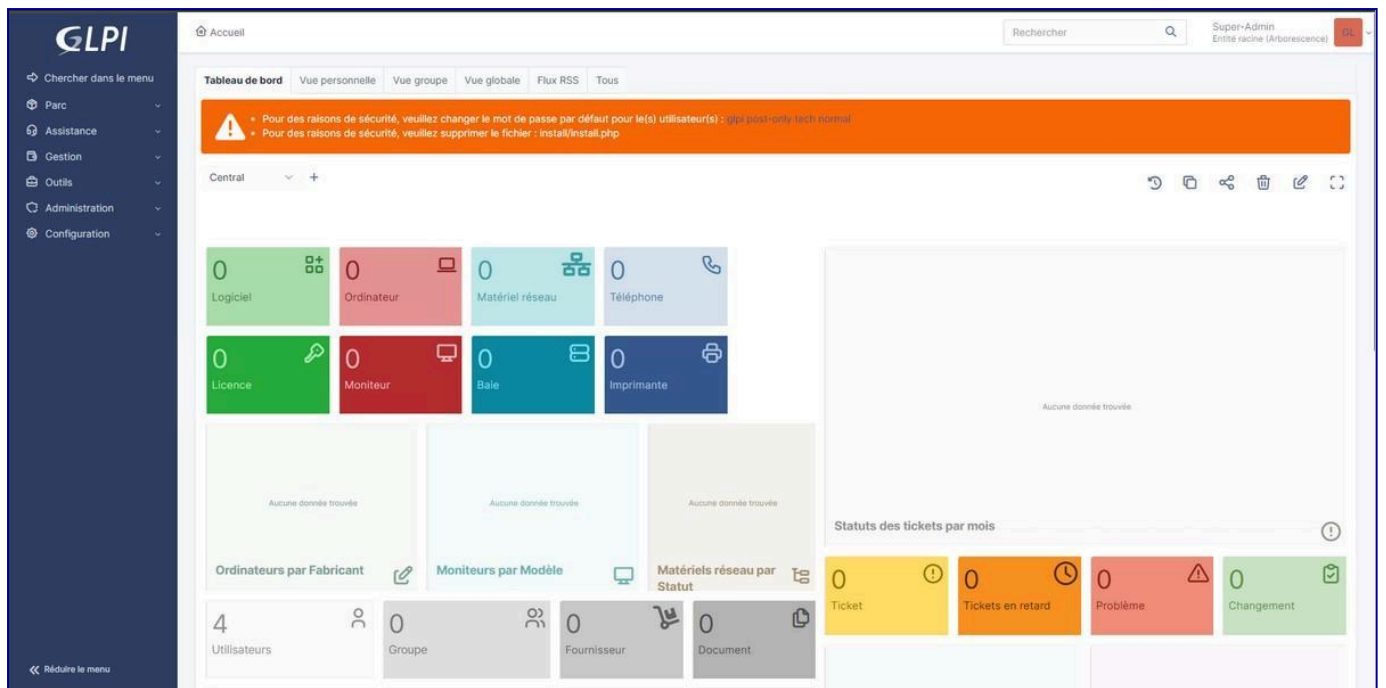
Source de connexion

Se souvenir de moi

Se connecter

GLPI Copyright (C) 2015-2023 Teclib' and contributors

Aperçu du site



Même si l'installation est terminée, il y a 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).