

BTS SIO – SISR

Travaux Pratiques – Gestion de Parc et Helpdesk

# Installation, Configuration et Utilisation de GLPI

---

**Touil Ayoub**

BTS SIO SISR – 2ème année

# 1. Présentation de GLPI

---

GLPI (Gestionnaire Libre de Parc Informatique) est une solution open-source de gestion de parc informatique et de helpdesk. Elle permet d'inventorier les équipements (ordinateurs, imprimantes, périphériques réseau), de gérer les tickets d'incidents, de créer une base de connaissances et d'administrer les utilisateurs selon des rôles différenciés.

Dans ce TP, GLPI est déployé sur une machine virtuelle Debian 12 avec la pile LAMP (Apache2, MariaDB, PHP 8.2) et configuré pour gérer des utilisateurs, des tickets d'incidents et une base de connaissances.

```
On va d'abord commencer par cloner notre machine debian 12 dans virtualbox
TP GLPI
↳ SA fonction
Ensuite on se connecte en root/root puis on va faire :
sudo apt-get update && sudo apt-get upgrade
Et on va installer apache2 et php8.2 :
sudo apt-get install apache2 php mariadb-server
Après, on va installer les extensions nécessaires :
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
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
On va préparer une base de données pour GLPI
sudo mysql_secure_installation
```

Figure 1 – Machine virtuelle TP GLPI démarrée sous VirtualBox

## 2. Installation de la stack LAMP et de GLPI

### 2.1 Prérequis : Apache2, PHP et MariaDB

Après clonage de la machine Debian 12 de base et mise à jour du système, les composants de la stack LAMP sont installés :

```
sudo apt-get install apache2 php mariadb-server
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 php-ldap
```

### 2.2 Sécurisation de MariaDB et création de la base

L'assistant `mysql_secure_installation` est exécuté pour sécuriser l'installation MariaDB (suppression des utilisateurs anonymes, désactivation de la connexion root à distance, suppression de la base de test). Ensuite, la base de données GLPI est créée :

```
CREATE DATABASE db23_glpi;
GRANT ALL PRIVILEGES ON db23_glpi.* TO glpi_adm@localhost IDENTIFIED BY
'MotDePasseRobuste';
FLUSH PRIVILEGES;
```

```
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'.
... skipping.
Change the root password? [Y/n] n
... skipping.
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
so 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!

On va se connecter à Mariadb :

mysql -u root -p
root@debian12:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 37
Server version: 10.11.14-MariaDB-0+deb12u2 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)]>
```

Figure 2 – Sécurisation de MariaDB et création de la base db23\_glpi

```

Ensuite, on va faire :
CREATE DATABASE db23_glpi;
GRANT ALL PRIVILEGES ON db23_glpi.* TO glpi_admin@localhost IDENTIFIED BY
"MotDePasseRobuste";
FLUSH PRIVILEGES;
EXIT

Thanks for using MariaDB!
MariaDB [root] root@localhost:~#
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 57
Server version: 10.11.14-MariaDB-0-debian2 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 [root] root@localhost:~# CREATE DATABASE db23_glpi;
Query OK, 1 row affected (0.001 sec)
MariaDB [root] root@localhost:~# GRANT ALL PRIVILEGES ON db23_glpi.* TO glpi_admin@localhost IDENTIFIED BY "root";
Query OK, 0 rows affected (0.000 sec)
MariaDB [root] root@localhost:~# FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.000 sec)
MariaDB [root] root@localhost:~# EXIT
Bye
root@deb1012:~#

```

Une fois cela fait, on va faire

```
wget https://github.com/glpi-project/glpi/releases/download/10.0.20/glpi-10.0.20.tgz
```

Puis on va faire :

```
sudo tar -xzvf glpi-10.0.10.tgz -C /var/www/
```

Pour donner la propriété des fichiers, on fait :

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

On va créer le répertoire glpi et lui donner les droits :

```
sudo mkdir /etc/glpi
sudo chown www-data /etc/glpi/
```

Puis on va déplacer le répertoire vers le répertoire qu'on vient de créer

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

sudo mkdir /var/lib/glpi existe, donc pas besoin de le faire  
ON va donner quand même les droits :

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

sudo mv /var/www/glpi/files /var/lib/glpi ne marche pas  
On va refaire la même que avant pour un autre répertoire :

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

On va créer le fichier de config glpi :

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

On ajoute ça dans le fichier :

Figure 3 – Création de la base de données et de l'utilisateur glpi\_admin

## 2.3 Téléchargement et déploiement de GLPI

GLPI 10.0.20 est téléchargé depuis le dépôt GitHub officiel et extrait dans /var/www/ :

```

wget https://github.com/glpi-project/glpi/releases/download/10.0.20/glpi-
10.0.20.tgz
sudo tar -xzvf glpi-10.0.20.tgz -C /var/www/
sudo chown www-data /var/www/glpi/ -R

```

Les répertoires de configuration, données et logs sont créés hors du DocumentRoot et placés dans leurs emplacements dédiés selon les bonnes pratiques de sécurité :

```

sudo mkdir /etc/glpi && sudo chown www-data /etc/glpi/
sudo mv /var/www/glpi/config /etc/glpi
sudo chown www-data /var/lib/glpi/ && sudo mkdir /var/log/glpi && sudo
chown www-data /var/log/glpi

```

## 2.4 Configuration d'Apache2 et PHP-FPM

Un VirtualHost Apache2 dédié est créé (support.it-connect.tech.conf) pointant vers /var/www/glpi/public avec le module rewrite activé. Le site par défaut est désactivé et PHP 8.2 FPM est configuré avec la directive session.cookie\_httponly = On pour renforcer la sécurité des sessions.

```

sudo a2ensite support.it-connect.tech.conf && sudo a2dissite 000-
default.conf
sudo a2enmod rewrite && sudo a2enmod proxy_fcgi setenvif
sudo a2enconf php8.2-fpm && sudo systemctl reload apache2

```

```
GNU nano 7.2
<?php
define('GLPI_CONFIG_DIR', '/etc/glpi/');
if (file_exists(GLPI_CONFIG_DIR . '/local_define.php')) {
    require_once GLPI_CONFIG_DIR . '/local_define.php';
}
}
```

On va créer le 2nd fichier :

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

Et on met ça :

```
GNU nano 7.2
define('GLPI_VAR_DIR', '/var/lib/glpi/files');
define('GLPI_LOG_DIR', '/var/log/glpi');
```

On va préparer la configuration apache2

```
nano /etc/apache2/sites-available/support.it-connect.tech.conf (le nom peut se remplacer support.it-connect.tech)
```

On met ça dans le fichier :

```
<VirtualHost *:80>
    ServerName support.it-connect.tech

    DocumentRoot /var/www/glpi/public

    <Directory /var/www/glpi/public>
        Require all granted

        RewriteEngine On

        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ index.php [QSA,L]
    </Directory>
</VirtualHost>
```

support.it-connect.tech peut se remplacer si le nom du fichier est différent

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*).

Figure 4 – Fichiers de configuration GLPI (downstream.php, local\_define.php) et VirtualHost Apache2

```
sudo a2enmod rewrite
```

Ensuite on va restart apache2 pour appliquer les modifications

```
systemctl restart apache2
```

ON va ensuite installer php8.2

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

ON va activer 2 modules dans apache et on va le redemarrer :

```
sudo a2enmod proxy_fcgi setenvif
sudo a2enconf php8.2-fpm
sudo systemctl reload apache2
```

On va modifier ce fichier :

```
sudo nano /etc/php/8.2/fpm/php.ini
On va s'assurer que cette ligne est bien comme sur la photo :
session.cookie_httponly = on
```

Ensuite, on va restart php

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

ON va s'assurer que notre fichier support.it-connect ressemble à ça :

```
###
### VirtualHost
###
<VirtualHost *:80>
    ServerName support.it-connect.tech
    ServerAlias support.it-connect.tech

    DocumentRoot /var/www/glpi/public

    <Directory /var/www/glpi/public>
        Require all granted

        RewriteEngine On

        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ index.php [QSA,L]
    </Directory>

    <FilesMatch \.php$>
        <IncludeOptional conf.d/php8.2-fpm.conf
    </FilesMatch>
</VirtualHost>
```

On restart apache2

```
sudo systemctl restart apache2
```

On se connecte à GLPI :

Figure 5 – Configuration PHP-FPM (session.cookie\_httponly) et VirtualHost final

### 3. Installation via l'assistant web

---

L'assistant d'installation GLPI est accessible via le navigateur. Les 6 étapes sont parcourues successivement :

Étape 1 — Sélection de la langue (Français).

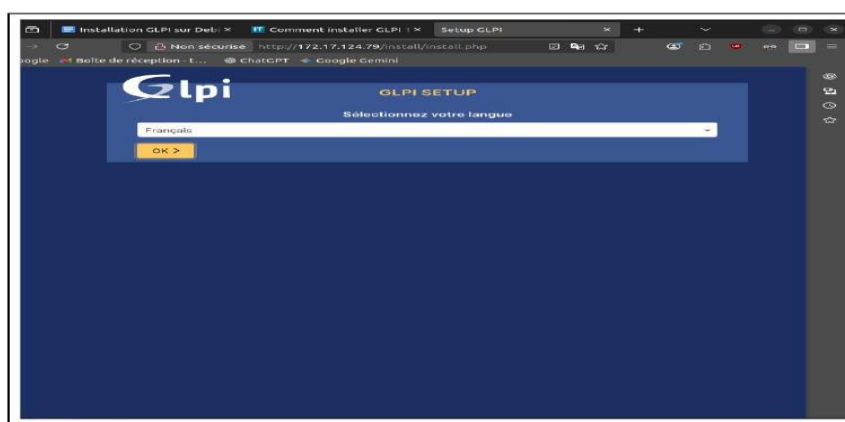
Étape 2 — Acceptation de la licence GNU GPL v3.

Étape 3 — Choix du type d'installation (nouvelle installation).

Étape 4 — Vérification des prérequis (une alerte pour l'extension LDAP non installée, ignorée pour ce TP).

Étape 5 — Configuration de la connexion à la base de données (serveur localhost, utilisateur gpi\_adm, sélection de db23\_gpi).

Étape 6 — Initialisation réussie de la base de données.



On suit le setup.

Figure 6 – Assistant GLPI : sélection de langue et début de l'installation



Figure 7 – Vérification des prérequis (alerte LDAP non bloquante) et connexion à la base

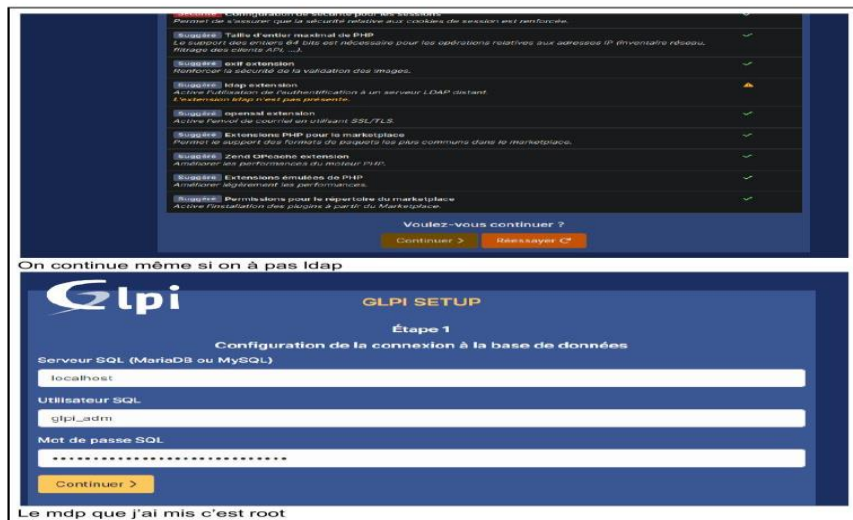


Figure 8 – Étape 1 : configuration de la connexion SQL (glpi\_admin / db23\_glpi)

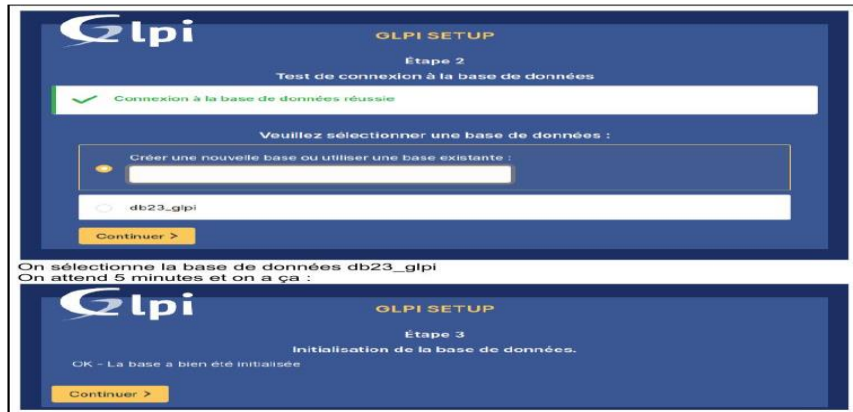


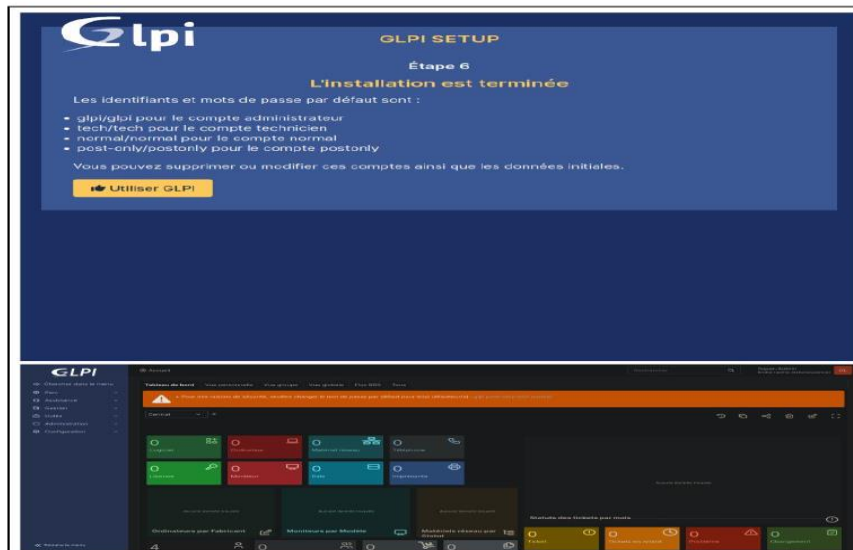
Figure 9 – Sélection de db23\_gipi et initialisation réussie (Étape 3)



Figure 10 – Étape 4 : désactivation des statistiques d'usage



Figure 11 – Étape 5 : informations GLPI-Network



Le mdp pour se connecter à glpi c'est : glpi/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)

Figure 12 – Installation terminée (Étape 6) : identifiants par défaut et tableau de bord GLPI

Après connexion (glpi/glpi), le fichier d'installation est supprimé pour éviter tout risque de réinstallation accidentelle :

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

## 4. Réinitialisation du mot de passe depuis MariaDB

Le professeur ayant modifié les identifiants GLPI, la réinitialisation du mot de passe est effectuée directement en base de données. Un nouveau hash bcrypt est généré via PHP, puis mis à jour dans la table `glpi_users` :

```
php -r 'echo password_hash("bts123", PASSWORD_BCRYPT) . "\n";'  
USE db23_glpi;  
UPDATE glpi_users SET password = '<hash>', authtype = 1, is_active = 1,  
password_last_update = NOW() WHERE name = 'glpi';
```

La connexion est ensuite rétablie avec le compte `glpi / bts123`. L'accès Super-Admin est confirmé dans l'interface.

```
Le professeur a changé nos logins sur glpi, notre mission est de pouvoir accéder à la  
session GLPI  
On va donc devoir utiliser la base de données pour pouvoir réinitialiser le mot de passe  
On va d'abord générer le mot de passe  
root@debian12:~# php -r 'echo password_hash("bts123", PASSWORD_BCRYPT) . "\n";'  
$2y#10#26CT1KQ2P/pD21pIfXd0.DafHbkhuWqTguFb8xTlQvTq0968DI0S  
ensuite , on va se rendre dans mariadb avec : mariadb et on va faire :  
USE db23_glpi  
SELECT name, password FROM glpi_users WHERE name = 'glpi';  
MariaDB [db23_glpi]> USE db23_glpi  
Database changed  
MariaDB [db23_glpi]> SELECT name, password FROM glpi_users WHERE name = 'glpi';  
+-----+-----+  
| name | password |  
+-----+-----+  
| glpi | $2y#10#26CT1KQ2P/pD21pIfXd0.DafHbkhuWqTguFb8xTlQvTq0968DI0S |  
+-----+-----+  
Et après on fait :  
UPDATE glpi_users SET password = 'le hashé', authtype = 1, is_active =  
1,password_last_update = NOW () WHERE name = 'glpi';  
MariaDB [db23_glpi]> UPDATE glpi_users SET password = 'le hashé', authtype = 1, is_active = 1, password_last_update = NOW () WHERE name = 'glpi';  
Query OK, 1 row affected (0.0006 sec)  
Ensuite, on va sur glpi :
```

Figure 13 – Génération du hash bcrypt et mise à jour en base MariaDB



On rentre sur glpi on rentre nos logins et on met se connecter  
On voit bien qu'on est en super admin qui est l'utilisateur dont on devait initialement entrer sur le compte :



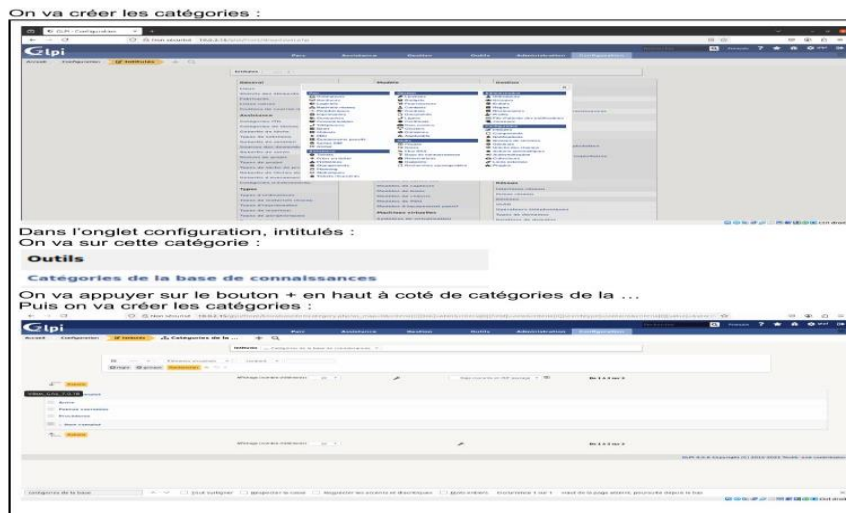
Figure 14 – Reconnexion réussie avec gpi/bts123 : accès Super-Admin confirmé

## 5. Création des utilisateurs et gestion des rôles

### 5.1 Structure des utilisateurs

Les utilisateurs sont créés depuis Administration → Utilisateurs. Chaque compte reçoit un rôle spécifique dans GLPI, définissant ses droits et son interface :

- M. CASORLA (identifiant : admin) — Administrateur GLPI, supervise l'ensemble du système.
- M. LANGLOIS, M. LUZZI, M. CHANTRIAUX — Techniciens, traitent les tickets d'incidents.
- M. LACOUR — Self-Service, peut créer des tickets et consulter la base de connaissances.



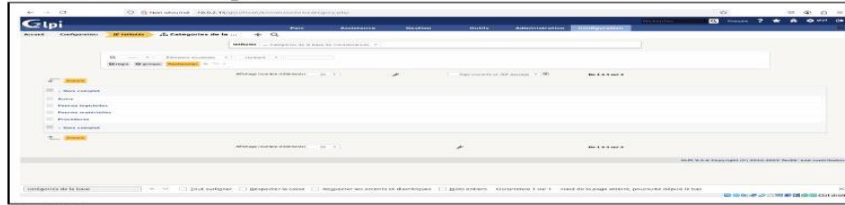
Travail à faire :  
Suite du TP GLPI Tickets d'incidents précédent.  
Vous avez réussi à créer 5 utilisateurs :  
M. CASORLA est Administrateur,  
M. LANGLOIS, M. LUZZI et M. CHANTRIAUX sont Techniciens dans le serveur GLPI.  
M. LACOUR est le Self-Service.  
Vous êtes stagiaire au service informatique de cette structure, vous travailler sous la

Figure 15 – Menu Administration → Utilisateurs et ajout d'un nouvel utilisateur

responsabilité de M. CASORLA qui vous demande de créer une base de connaissances. Il vous demande de créer 2 articles pour des pannes possibles et de les associés aux catégories que vous allez créer qui sont les suivantes :

1. Pannes matérielles,
2. Pannes logicielles,
3. Procédures,
4. Autres.

L'objectif c'est d'avoir deux articles par catégories.  
 Vous allez donner accès à 4 articles à M. LACOUR concernant les pannes matérielles et les pannes logicielles.  
 Vous allez donner uniquement accès aux articles de la catégories Procédures pour les techniciens.  
 Les articles associés aux « Autres » ne sont pas encore publiés.  
 J'ai donc créé les 4 catégories :

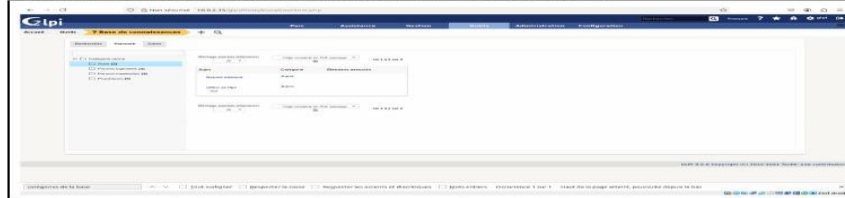


Ensuite :  
 J'ai fait des groupes avec les utilisateurs dedans pour que ça soit plus simple :

Figure 16 – Création de l'administrateur CASORLA Quentin

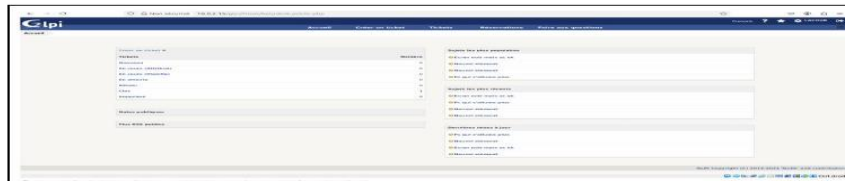


Ensuite je vais sur base de connaissances pour créer les articles :



Et ensuite je vais tester avec les utilisateurs LACOUR et un technicien :  
 LACOUR :

Figure 17 – Création du technicien LANGLOIS Victor



On voit bien les articles donc c'est ok !



Il ne voit que les 2 articles que j'ai décidé donc c'est bon  
Fin du tp

Figure 18 – Création du technicien LUZZI Evan

On va aller dans les 3 petits tirets à droite puis dans l'onglet administration :

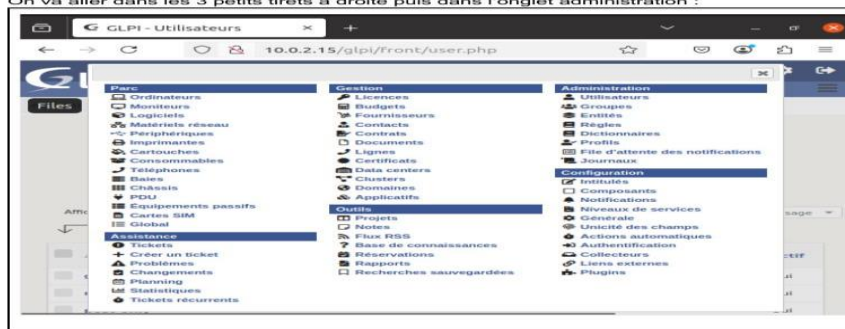


Figure 19 – Interface helpdesk de LACOUR (Self-Service) et création d'un ticket

## 6. Gestion des tickets d'incidents

### 6.1 Cycle de vie d'un ticket

Le test du workflow de ticketing est réalisé end-to-end entre l'utilisateur LACOUR et le technicien LANGLOIS :

1. LACOUR (Self-Service) crée un ticket d'incident intitulé « Recherche stage » depuis l'interface helpdesk.
2. LANGLOIS (Technicien) visualise le ticket entrant dans sa file, consulte le contenu et rédige une solution via le bouton Solution.
3. LACOUR se reconnecte, consulte la solution proposée et l'approuve, ce qui clôture automatiquement le ticket.

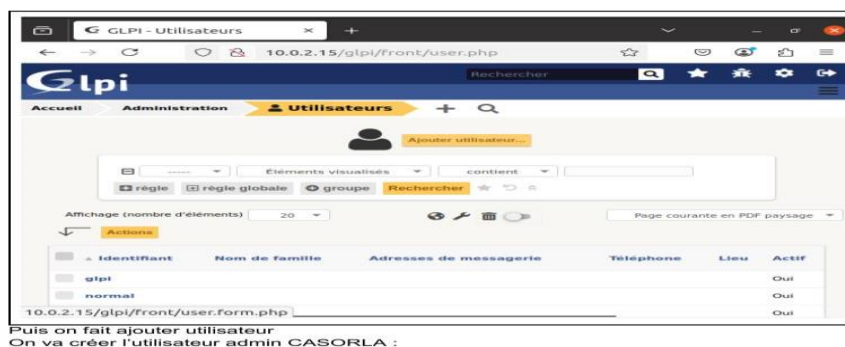


Figure 20 – Vue technicien LANGLOIS : ticket de LACOUR visible dans la file d'assistance



Et on clique sur ajouter en bas  
On va ajouter aussi les 3 techniciens :

Figure 21 – Technicien : rédaction de la solution (bouton Solution)

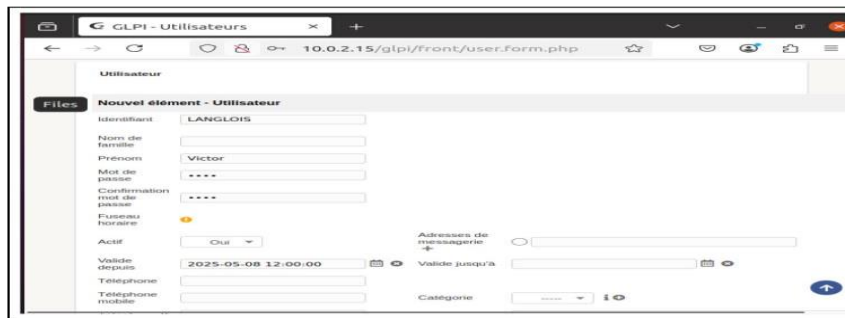


Figure 22 – LACOUR : approbation de la solution et clôture du ticket

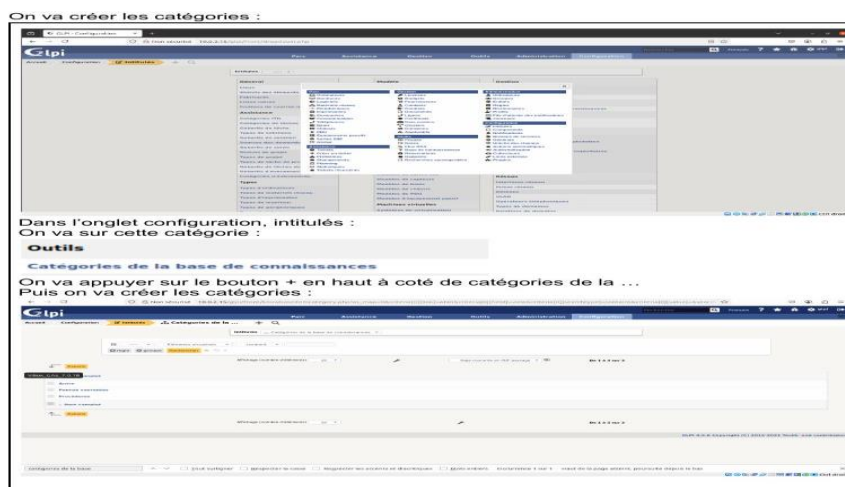
## 7. Base de connaissances

### 7.1 Création des catégories et des articles

La base de connaissances est organisée en 4 catégories, créées depuis Configuration → Intitulés → Catégories de la base de connaissances :

- Pannes matérielles — 2 articles (accessibles à M. LACOUR)
- Pannes logicielles — 2 articles (accessibles à M. LACOUR)
- Procédures — 2 articles (accessibles aux techniciens uniquement)
- Autres — 2 articles (non publiés)

Des groupes GLPI sont créés pour faciliter l'attribution des droits : groupe SelfService (LACOUR) et groupe Technicien (LANGLOIS, LUZZI, CHANTRIAUX).



**Travail à faire :**  
Suite du TP GLPI Tickets d'incidents précédent.  
Vous avez réussi à créer 5 utilisateurs :  
M. CASORLA est Administrateur.  
M. LANGLOIS, M. LUZZI et M. CHANTRIAUX sont Techniciens dans le serveur GLPI.  
M. LACOUR est le Self-Service.  
Vous êtes stagiaire au service informatique de cette structure, vous travailler sous la

Figure 23 – Groupes GLPI : SelfService et Technicien

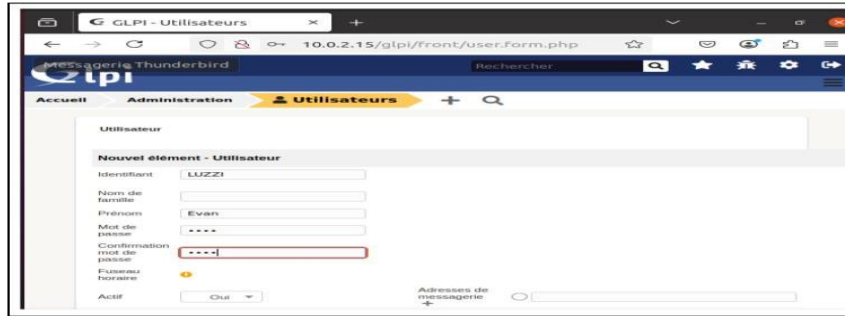


Figure 24 – Liste des 4 catégories créées dans la base de connaissances

## 7.2 Validation des droits d'accès

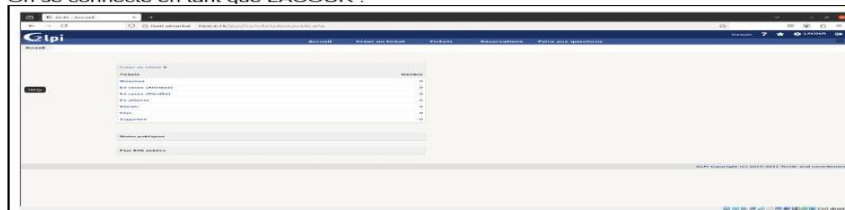
Les droits sont vérifiés en se connectant successivement avec les différents profils :

- LACOUR voit les articles des catégories Pannes matérielles et Pannes logicielles (4 articles au total).
- LANGLOIS (Technicien) ne voit que les 2 articles de la catégorie Procédures.
- Les articles de la catégorie Autres ne sont pas encore publiés et n'apparaissent pour aucun profil.



On va ajouter en tant que self-service : LACOUR

On se connecte en tant que LACOUR :

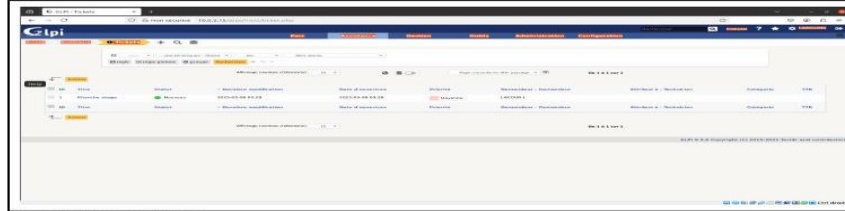


Et on va créer un ticket :

Figure 25 – Vue LACOUR : articles visibles (Pannes matérielles + logicielles)



On se connecte sur LANGLOIS :



On voit bien le ticket :

Figure 26 – Vue technicien LANGLOIS : 2 articles Procédures uniquement

## 8. Conclusion

---

Ce TP a permis de déployer GLPI de bout en bout : installation de la stack LAMP, configuration sécurisée d'Apache2 et PHP-FPM, initialisation de la base MariaDB, et parcours complet de l'assistant web. Les fonctionnalités clés ont été testées : gestion différenciée des rôles (Admin, Technicien, Self-Service), cycle de vie d'un ticket d'incident et contrôle des accès à la base de connaissances par groupes. GLPI constitue un outil professionnel incontournable pour la gestion de parc informatique et le support aux utilisateurs.