

# Installation des prérequis pour GLPI

Une fois l'installation de Debian terminée, je me rends sur le site de GLPI afin de voir les prérequis pour qu'il fonctionne

La documentation officielle de GLPI m'informe qu'il me faut :

- **Un serveur Web** (Comme j'ai choisi d'installer un serveur web a l'installation de Debian, je dispose déjà de Apache 2.4.59)
- **PHP** avec comme version minimum 7.4
- **Une base de données**, la documentation de GLPI m'indique que uniquement MySQL et MariaDB sont supportées par GLPI, j'ai choisi d'utiliser MariaDB (pour changer de mon installation Windows ou j'avais utilisé MySQL)



## Mise en place d'une IP Fixe

Avant de faire quoi que ce soit, je m'assure que ma machine dispose d'une adresse IP fixe. J'effectue la commande `ip a` dans le but d'obtenir le nom actuel de mon interface réseau.

```
root@glpi:/home# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
   qlen 1000
    link/ether 52:54:00:48:a4:3e brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.26/24 brd 192.168.122.255 scope global dynamic enp1s0
        valid_lft 3440sec preferred_lft 3440sec
    inet6 fe80::5054:ff:fe48:a43e/64 scope link
        valid_lft forever preferred_lft forever
```

Ici le nom de mon interface est **enp1s0**.

Je consulte ensuite le fichier `/etc/network/interfaces`.

```
This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp1s0
iface enp1s0 inet dhcp
```

Je vois que mon interface est actuellement en **DHCP**, je passe donc l'adressage en **fixe** en modifiant ce fichier de cette manière :

```
iface enp1s0 inet static
    address 192.168.122.26
    netmask 255.255.255.0
    gateway 192.168.122.1
    dns-nameservers 8.8.8.8 8.8.4.4
```

Je lui ai tout simplement "Fixé" l'adresse qu'il avait déjà, je n'ai plus qu'à redémarrer le service réseau afin d'appliquer mes modifications avec la commande `systemctl restart networking`

Il faut maintenant que je fasse une exclusion ou une réservation sur mon serveur DHCP. Dans mon étendue DHCP, j'indique qu'il faut exclure l'adresse **192.168.122.26** de la plage.

## Installation des paquets manquants

Pour installer les paquets manquants, je m'assure d'abord que tout est à jour en faisant la commande

```
apt update && apt upgrade -y
```

J'installe ensuite les deux paquets qui me manquent c'est-à-dire **PHP** et la base de données **MariaDB**.

```
apt install php mariadb-server
```

Les dépôts de Debian m'installent PHP 8.2 ainsi que MariaDB 10.11.6.

## Configuration initiale de PHP

Dans la documentation de GLPI, on nous indique qu'il faut activer des extensions obligatoires dans le bon fonctionnement du service.

“ Ces extensions PHP suivantes sont requises pour que l'application fonctionne correctement :

- `dom`, `fileinfo`, `filter`, `libxml`, `json`, `simplexml`, `xmlreader`, `xmlwriter`: these PHP extensions are enable by default and are used for various operations;
- `curl`: used for remote access to resources (inventory agent requests, marketplace, RSS feeds, ...);
- `gd`: used for images handling;
- `intl`: used for internationalization;
- `mysqli`: used for database connection;
- `session`: used for sessions support;
- `zlib`: used for handling of compressed communication with inventory agents, installation of gzip packages from marketplace and PDF generation.

Je vais donc éditer le fichier `php.ini` présent dans le répertoire `/etc/php/8.2/apache2/`

Je me rends dans la catégorie "Dynamic Extensions" et enlève-le ; devant chaque extension que je souhaite activer.

## Configuration de la base de données MariaDB



Avant de faire quoi que ce soit, je lance le script :

```
mysql_secure_installation
```

Il me permet de définir plusieurs options de sécurité comme supprimer le mot de passe root et définir les options de connexion à distance, etc.

```
root@glpi:~# mysql_secure_installation
```

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB  
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
```

```
In order to log into MariaDB to secure it, we'll need the current  
password for the root user. If you've just installed MariaDB, and  
haven't set the root password yet, you should just press enter here.
```

```
Enter current password for root (enter for none):
```

```
OK, successfully used password, moving on...
```

```
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] 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  
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] n
```

```
... skipping.
```

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!

Je vais ensuite créer une base de données dédiée à GLPI, je me connecte donc en tant que superutilisateur à MariaDB.

```
mysql -u root -p
```

Je crée ma base de données avec comme nom "glpi".

```
CREATE DATABASE glpi;
```

Je crée un utilisateur "**administrateurglpi**" avec comme mot de passe "**motdepasse**"

```
CREATE USER 'administrateurglpi'@'localhost' IDENTIFIED BY 'motdepasse';
```

Je donne tous les droits à mon utilisateur sur la base de données **glpi**

```
GRANT ALL PRIVILEGES ON glpi.* TO 'administrateurglpi'@'localhost';
```

Et je recharge la table des privilèges pour appliquer mes modifications et je quitte

```
FLUSH PRIVILEGES;  
EXIT;
```

Je dispose désormais de la configuration suivante :

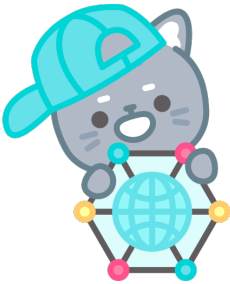
**Base de données** : glpi

**Utilisateur ayant les droits sur la base de données** : administrateurglpi

**Mot de passe de l'utilisateur** : motdepasse

**Adresse de la base de données** : localhost/192.168.122.26

## Configuration d'Apache



Il faut que je configure un hôte virtuel dans Apache pour mon GLPI. Un hôte virtuel me permettra d'ajouter plusieurs services plus tard sur le même serveur Apache si je le souhaite.

Je crée le fichier de configuration pour GLPI

```
nano /etc/apache2/sites-available/glpi.conf
```

Et j'ajoute la configuration recommandée par la documentation GLPI dans ce fichier.

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

# 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 file exists.
RewriteCond %{REQUEST_FILENAME} !-f
RewriteRule ^(.*)$ index.php [QSA,L]

</Directory>
</VirtualHost>
```

**J'active le module rewrite**, car il est demandé dans le fichier de configuration (Ligne 13).

```
a2enmod rewrite
```

**J'active le site** que je viens de créer.

```
a2ensite glpi.conf
```

**J'en profite pour désactiver le site par défaut.**

```
a2dissite 000-default.conf
```

**Je redémarre** ensuite le service apache2

```
systemctl restart apache2
```

## Téléchargement de GLPI

Il ne me reste plus qu'à télécharger GLPI et à le placer à la racine de mon serveur web, pour faire cela, je me rends sur le site officiel de GLPI afin de récupérer l'URL de téléchargement et de télécharger l'archive avec wget

```
wget https://github.com/glpi-project/glpi/releases/download/10.0.15/glpi-10.0.15.tgz
```

J'extrait ensuite l'archive dans le répertoire **/var/www**

```
tar -xvf glpi-10.0.15.tgz -C /var/www/
```

Et donne les permissions à Apache d'accéder au répertoire **/var/www/glpi**

```
chown -R www-data:www-data /var/www/glpi/  
chmod -R 755 /var/www/glpi/
```

## Création des dossiers de configuration

La documentation de GLPI nous indique qu'il faut que la configuration de GLPI soit stockée dans `/etc/glpi` et déplacer la configuration actuelle dans celui-ci. Je donne également les permissions d'accès au répertoire

```
mkdir /etc/glpi  
mv /var/www/glpi/config/ /etc/glpi/  
chown -R www-data /etc/glpi/
```

Il nous indique qu'il faut faire pareil avec le contenu du dossier files dans `/var/lib/glpi/`

```
mkdir /var/lib/glpi  
mv /var/www/glpi/files/ /var/lib/glpi/  
chown -R www-data /var/lib/glpi/
```

Il faudra aussi créer un dossier `/var/log/glpi` afin d'y accueillir les logs, rien à copier ici

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

Il faudra également créer un fichier `inc/downstream.php` dans le dossier de GLPI avec le contenu indiqué sur la documentation de GLPI

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

Et colle le contenu suivant

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

Je fais la même chose pour le fichier `/etc/glpi/local_define.php`

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

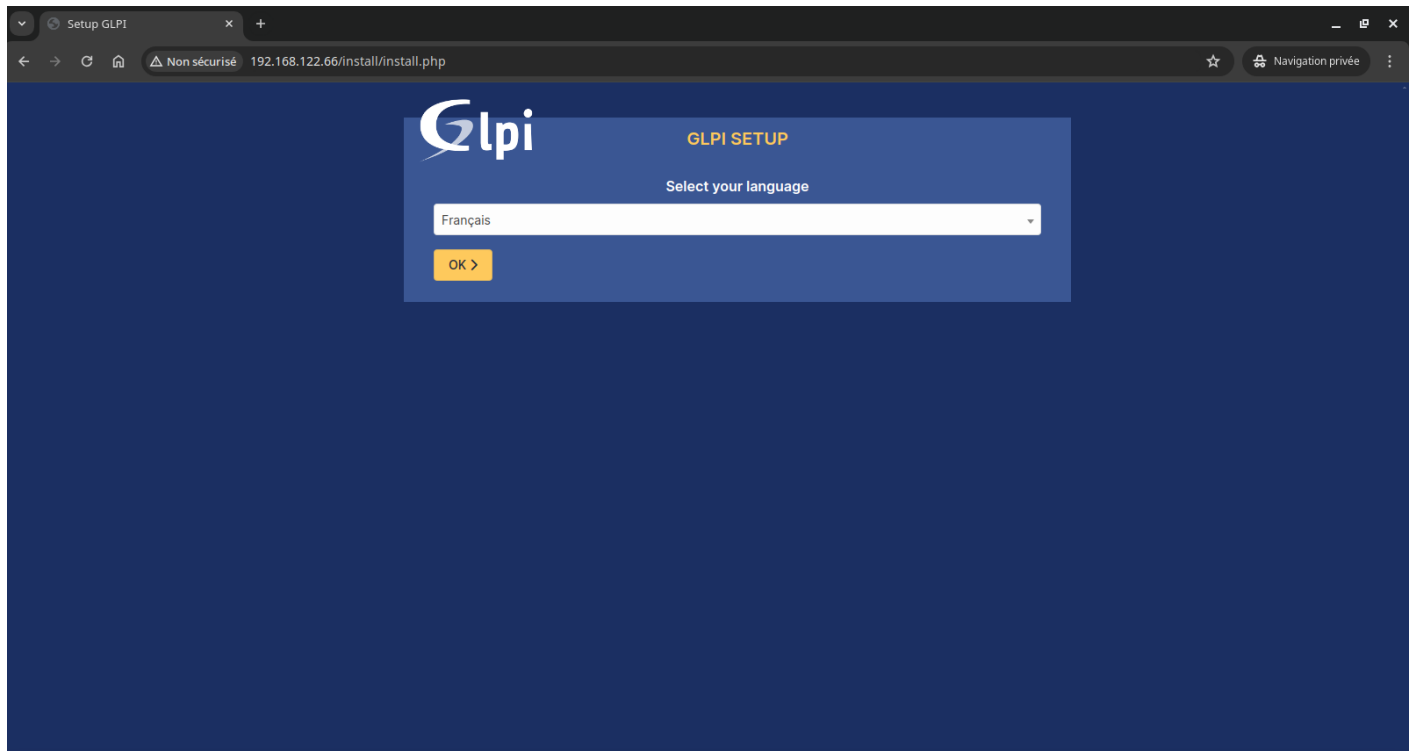
Et je colle le contenu suivant

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

## Vérification de la configuration



Pour vérifier que j'ai bien tout configuré, je tape l'adresse IP de ma machine virtuelle dans mon navigateur internet.



J'arrive bien sur la page d'installation de GLPI je peux maintenant passer à l'étape suivante !

Révision #8

Créé 2024-05-22 08:43:18 UTC par Renard

Mis à jour 2024-05-22 16:12:04 UTC par Renard