Clément MONTMAYEUR Mathieu URGIN

Installation et Configuration de GLPI

AssurMer V1

Date de réalisation : 02/04/2024 Validé par DSI AssuMer

Sommaire :

P0 – Présentation et prérequis3
P1 – Installation
- P1.1 Service Base de données
- P1.2 Service PHP
- P1.3 Service Base de données
- P1.4 Installation GLPI
P2 – Configuration GLPI4

P0 – Présentation et prérequis

GLPI est application web permettant la gestions des services informatiques et la gestion des services d'assistance pour les entreprises, elle est édité en PHP et permet en premier temps de crée une base de données regroupant des ressources techniques et de gestion, mais aussi de fournir au utilisateurs une fonctionnalité de gestion d'assistance (helpdesk),

Nous déploierons cette solutions pour Assumer dans le but de réaliser l'inventaires de ressources de l'entreprise mais aussi pour fournir une assistance de ticketing aux utilisateurs,

GLPI et un service pouvant s'installer sur divers systèmes d'exploitation Linux notamment :

- Ubuntu / Debian (Facile à configurer, grande communauté, gratuit)
- CentOS / RockyLinux (Plus adapté au professionnels car souvent plus sécurisé)
- Red Hat Entreprise (Version

GLPI fonctionne sur Windows Serveur mais il est moins bien supporté,

De plus, pour fonctionner, GLPI nécessite plusieurs dépendances :

- Serveur Web (Ex : Apache 2 ou Nginx)
- PHP (PHP 7.4.X minimum)
- Base de données (Ex : MariaDB 10.2+ ou MySQL 5.7+)

Dans cette procédure nous utiliserons Ubuntu serveur, une version optimiser de Ubuntu sans interface graphique, peu utile dans notre cas,

Si vous utilisez cette OS, voici la configuration minimale nécessaire :

- CPU 2 Cœurs 64Bits
- RAM 2Go
- Stockage 25Gb

Information Machine SRV-GLPI : OS : Ubuntu Serveur 24.04 IP : 172.16.0.11 Specs : 2CPU/4Go/200Go

P1.1 – Installation et configuration base de données

Dans un premier temps, effectuez les mises à jour logiciel de base avec : sudo apt update && apt upgrade

The following upgrades have been deferred due to phasing: apport apport-core-dump-handler python3-apport python3-problem-report sosreport ubuntu-drivers-common) mis à jour, 0 nouvellement installés, 0 à enlever et 6 non mis à jour. coot@srv-glpi:~#

Redémarrez si nécessaire

Ensuite installons la base de données, dans notre cas, nous installerons MariaDB Tapez : apt install -y mariadb-server

Puis pour passer à la configuration tapez : mysql_secure_installation Et faite comme la capture suivante :

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.

Ensuite configurez la timezone : mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql mysql root@srv-glpi:~# mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql mysql root@srv-glpi:~#

Créons maintenant un utilisateur et une base de données pour GLPI :

Utilisez ce modèle comme référence, pensez a changer le mot de passe par quelque chose de fort

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;

Cela fait, vous devriez obtenir ce message :



P1.2 – Installation et configuration fichiers GLPI

Ensuite passons à l'installation des fichiers de GLPI,

P1.3 – Installation et configuration serveur Web

Puis, installez le serveur Web (ici Apache 2) via la commande : apt install -y apache2 php php-{apcu,cli,common,curl,gd,imap,ldap,mysql,xmlrpc,xml,mbstring,bcmath,intl,zip,redis,bz2} libapache2-mod-php php-soap php-cas

Ensuite, on configure le serveur Web, Tout dabord on crée une interface reseau virtuel dédié a GLPI : nano /etc/apache2/sites-available/glpi.conf

On ajoute ceci, en prenant soin de modifier ce qui est en vert <VirtualHost *:80> ServerName yourglpi.yourdomain.com DocumentRoot /var/www/html/glpi/public <Directory /var/www/html/glpi/public> Require all granted RewriteEngine On RewriteCond %{HTTP:Authorization} ^(.+)\$ RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Authorization}] RewriteCond %{REQUEST_FILENAME} !-f RewriteRule ^(.*)\$ index.php [QSA,L] </VirtualHost>

Exemple :

GNU nano 7.2	/etc/apache2
<virtualhost *:80=""></virtualhost>	
ServerName srv-glpi	
DocumentRoot /var/www/html/glpi/public	
<directory glpi="" html="" public="" var="" www=""></directory>	
Require all granted	
. RewriteEngine On	
RewriteCond %{HTTP:Authorization} ^(.+)\$	
RewriteRule .* - [E=HTTP_AUTHORIZATION:%{HTTP:Author	ization}]
RewriteCond %{REQUEST_FILENAME} !-f	
RewriteRule ^(.*)\$ index.php [QSA,L]	
AG Help AO Write Out AW Where Is AK Cut	
AX Exit AR Read File AN Replace AU Paste	
Puis enregistrez	

Nous effectuerons la suite de la configuration plus tard,