

a) Installation des composant requis :

a) Installation des composant requis :

Dans un premier temps on installe les composant de base :

Apache2 pour le serveur web et mariadb-server pour le serveur de base de données

```
sudo apt-get install apache2 php8.4-fpm mariadb-server
```

```
root@GLPI-VM:/home/glpi# sudo apt-get install apache2 php mariadb-server
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
 apache2-bin apache2-data apache2-utils galera-4 gawk libapache2-mod-php8.4 libapr1t64 libaprutil1-dbd-sqlite3
 libaprutil1-ldap libaprutil1t64 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl libdbd-mariadb-perl
 libdbi-perl libfcgi-bin libfcgi-perl libfcgi0t64 libhtml-template-perl liblua5.4-0 libmariadb3 libpcre2-posix3
 libsigsegv2 libterm-readkey-perl mariadb-client mariadb-client-core mariadb-common mariadb-plugin-provider-bzip2
 mariadb-plugin-provider-lz4 mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo mariadb-plugin-provider-snappy
 mariadb-server-core mysql-common php-common php8.4 php8.4-cli php8.4-common php8.4-opcache php8.4-readline pv rsync
 socat
Paquets suggérés :
 apache2-doc apache2-suexec-pristine | apache2-suexec-custom ufw gawk-doc php-pear libmldbm-perl libnet-daemon-perl
 libsql-statement-perl libipc-sharedcache-perl mailx mariadb-test netcat-openbsd doc-base python3-braceexpand
Les NOUVEAUX paquets suivants seront installés :
 apache2 apache2-bin apache2-data apache2-utils galera-4 gawk libapache2-mod-php8.4 libapr1t64
 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64 libcgi-fast-perl libcgi-pm-perl libconfig-inifiles-perl
 libdbd-mariadb-perl libdbi-perl libfcgi-bin libfcgi-perl libfcgi0t64 libhtml-template-perl liblua5.4-0 libmariadb3
 libpcre2-posix3 libsigsegv2 libterm-readkey-perl mariadb-client mariadb-client-core mariadb-common
 mariadb-plugin-provider-bzip2 mariadb-plugin-provider-lz4 mariadb-plugin-provider-lzma mariadb-plugin-provider-lzo
 mariadb-plugin-provider-snappy mariadb-server mariadb-server-core mysql-common php php-common php8.4 php8.4-cli
 php8.4-common php8.4-opcache php8.4-readline pv rsync socat
0 mis à jour, 46 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de prendre 27,6 MB dans les archives.
```

Après on installe les composant pour l'exécution de glpi (pour GLPI ce sont des modules PHP) :

```
sudo apt install php8.4-{curl,gd,intl,mysql,zip,bcmath,mbstring,xml,bz2,ldap}
```

```
Installation de :
  php8.4-bcmath  php8.4-curl  php8.4-intl  php8.4-mbstring  php8.4-xml
  php8.4-bz2     php8.4-gd    php8.4-ldap  php8.4-mysql     php8.4-zip

Installation de dépendances :
  libonig5  libzip5

Sommaire :
  Mise à niveau de : 0. Installation de : 12Supprimé : 0. Non mis à jour : 0
Taille du téléchargement : 1 699 kB
  Espace nécessaire : 6 506 kB / 56,7 GB disponible

Continuer ? [0/n] o|
```

Puis configuration de mysql pour la base de données :

```
sudo mysql_secure_installation
```

```
gipi@GLPI-VH:~$ sudo mariadb-secure-installation
NOTE: MariaDB is secure by default in Debian. Running this script is
useless at best, and misleading at worst. This script will be
removed in a future MariaDB release in Debian. Please read
mariadb-server.README.Debian for details.

Enter root user password or leave blank:

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] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
On change le mot de
passe root, dans la
simplicité je remets le
même de debian

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.
On interdit les utilisateur anonyme

Remove anonymous users? [Y/n] y
SQL executed without errors!
The operation might have been successful, or it might have not done anything.

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
SQL executed without errors!
The operation might have been successful, or it might have not done anything.
On interdit à root de se connecté
à distance

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 ...
SQL executed without errors!
The operation might have been successful, or it might have not done anything.
- Removing privileges on test database ...
SQL executed without errors!
The operation might have been successful, or it might have not done anything.
On enlève les base
de donnée d'exemple

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y
... Success!
On applique toutes
les modifs au table

Cleaning up ...

All done! If you've completed all of the above steps, your MariaDB
```

Revision #2

Created 2025-12-22 21:15:39 UTC by Renard

Updated 2025-12-23 18:06:16 UTC by Renard