

# MISE EN SERVICE D'UN SERVEUR LAMP GLPI 10

---



ITSM

---

---

Machine Hôte

Système	Debian 11
Utilisateur root Debian	glpi
Mot de passe Debian	Btssio32
Serveur WEB	Apache 2
Serveur PHP	PHP 7.4
Serveur BDD	MariaDB

## Installation Apache

On commence par mettre à jour le cache des paquets :

```
sudo apt-get update
```

Ensuite, on installe le paquet "apache2" afin d'obtenir la dernière version d'Apache 2.4.

```
sudo apt-get install -y apache2
```

Pour qu'Apache démarre automatiquement en même temps que Debian, saisissez la commande ci-dessous (même si normalement c'est déjà le cas) :

```
systemctl enable apache2
```

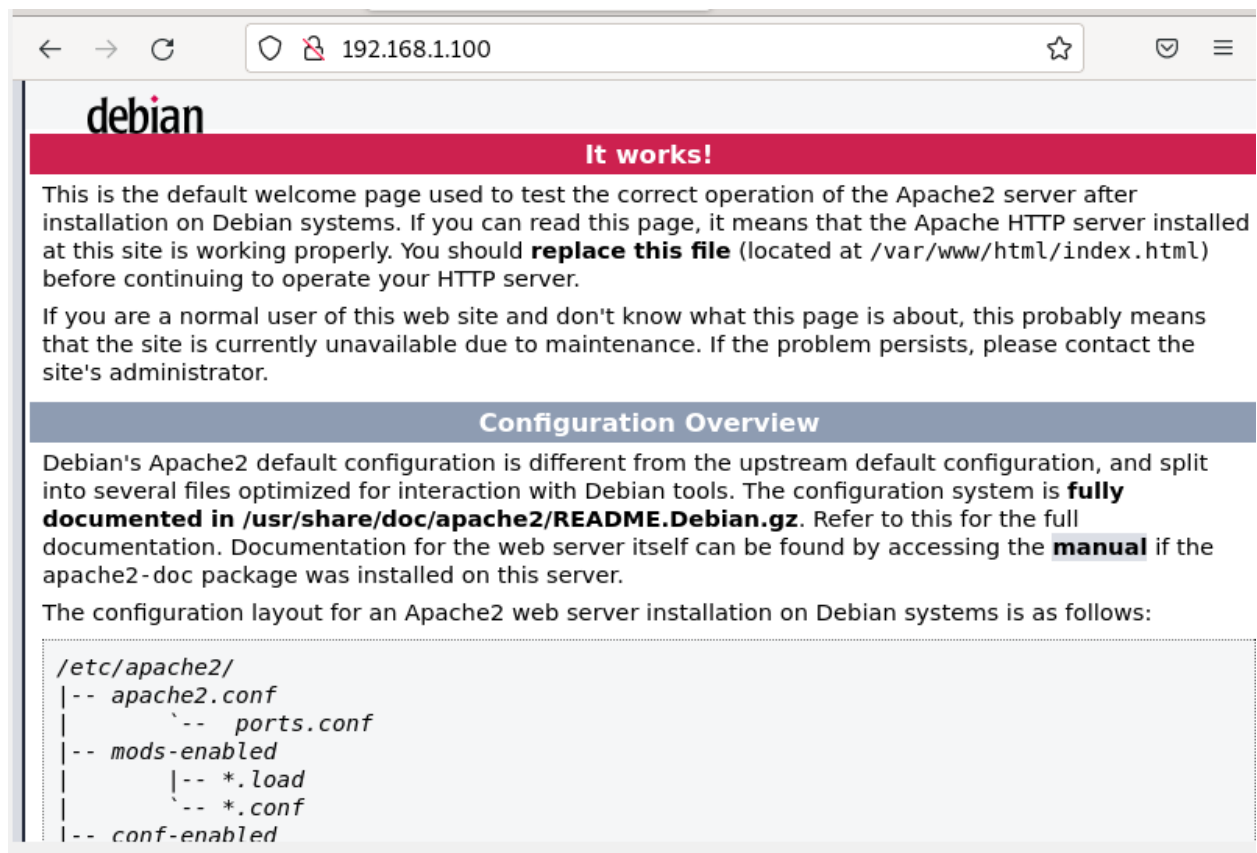
Suite à l'installation du paquet, le serveur Apache démarre directement. On devrait pouvoir accéder à sa page par défaut. Pour cela, il suffit de récupérer l'adresse IP du serveur :

```
ip address
```

---

Puis, à l'aide d'une machine équipée d'un navigateur, on peut accéder à notre serveur Apache :

`http://192.168.1.100`



Il est nécessaire d'activer quelques modules d'Apache qui sont indispensables, notamment pour faire tourner un site Internet.

---

Commençons par le module utilisé pour la réécriture d'URL :

```
a2enmod rewrite
```

Après avoir activé ou désactivé un module, ou modifié la configuration d'Apache, il faut redémarrer le service apache2 :

```
systemctl restart apache2
```

Le fichier de configuration d'Apache 2 est le suivant :

```
/etc/apache2/apache2.conf
```

## Installation php 7.4.30

**Importation du référentiel PHP Ondřej Surý :**

PHP va venir se greffer sur notre serveur Apache, comme une extension, afin de pouvoir traiter les scripts intégrés aux pages ".php". Afin d'y aller progressivement, installons le paquet "php" en lui-même :

```
sudo apt-get install -y php
```

On peut voir que cette commande va installer une multitude de paquets :

```
libapache2-mod-php7.4 libsodium23 php-common php7.4 php7.4-cli  
php7.4-common php7.4-json php7.4-opcache php7.4-readline
```

Avant d'aller plus loin, nous allons installer quelques paquets supplémentaires pour compléter l'installation de PHP sur notre serveur. Par exemple, pour permettre les interactions entre PHP et notre instance **MariaDB**.

```
sudo apt-get install -y php-pdo php-mysql php-zip php-gd php-mbstring  
php-curl php-xml php-pear php-bcmath
```

Maintenant, pour nous assurer que notre moteur de script PHP est bien actif, nous allons créer un fichier "phpinfo.php" (ou un autre nom) à la racine de notre site Web :

```
sudo nano /var/www/html/phpinfo.php
```

Dans ce fichier, indiquez le code suivant :

```
<?php
```

```
phpinfo();
```

```
?>
```

Elle sera accessible à partir de cette adresse :

```
http://192.168.1.100/phpinfo.php
```

PHP Version 7.4.30	
System	Linux Machine-gipi 5.10.0-18-amd64 #1 SMP Debian 5.10.140-1 (2022-09-02) :
Build Date	Jul 7 2022 15:51:43
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pd /7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, / /conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/i /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext. /7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/p /20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/cor /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sock /7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.i /7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.i
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902

---

Ajoutez des extensions nécessaire pour permettre à **glpi** de fonctionner.

```
apt install php-ldap php-imap php-apcu php-xmldr php-cas php-mysqli  
php-mbstring php-curl php-gd php-simplexml php-xml php-intl php-zip  
php-bz2 -y
```

## Installation Mysql/MariaDB

Pour installer MariaDB sous Debian 11, voici la commande à exécuter :

```
sudo apt-get install -y mariadb-server
```

Suite à l'installation, je vous invite à exécuter le script "mariadb-secure-installation" afin de sécuriser un minimum votre installation de MariaDB.

```
sudo mariadb-secure-installation
```

Pour obtenir le numéro de version de MariaDB, on peut utiliser cette commande :

```
mariadb -V  
mariadb Ver 15.1 Distrib 10.5.12-MariaDB, for debian-linux-gnu (x86_64)  
using EditLine wrapper
```

Se connecter à MariaDB en tant que root :

```
sudo mariadb -u root -p
```

---

Saisir le mot de passe "root".

Dans la console MariaDB / MySQL. Vous pouvez saisir vos requêtes SQL ici. Par exemple, pour lister les bases de données de votre instance :

```
show databases;
```

```
MariaDB [(none)]> show databases;
```

```
+-----+
| Database          |
+-----+
| glpi              |
| information_schema |
| mysql             |
| performance_schema |
+-----+
4 rows in set (0.000 sec)
```

```
MariaDB [(none)]> ■
```

---

# Installation GLPI 10

Maintenant on télécharge l'archive de GLPI depuis github

```
wget
```

```
https://github.com/glpi-project/glpi/releases/download/10.0.3/glpi-10.0.3.
```

```
tgz
```

On décompresse le fichier et le mettre dans le dossier /var/www/html

```
tar xzf glpi-10.0.3.tgz -C /var/www/html
```

On change les permissions sur le dossier de GLPI afin que le serveur web Apache puisse y accéder :

```
chown -R www-data:www-data /var/www/html/glpi
```

```
chmod -R 775 /var/www/html/glpi
```



---

# Installation et configuration MariaDB

On utilise MariaDB pour la partie base de données (SQL). Je crée une base de données, un utilisateur et je donne la permission à ce dernier de travailler sur ma base de données.

```
mysql -u root
```

Dans le terminal de MariaDB Créer la base glpi, Créer l'utilisateur "glpiuser" dont le mot de passe est "Btssio32" :

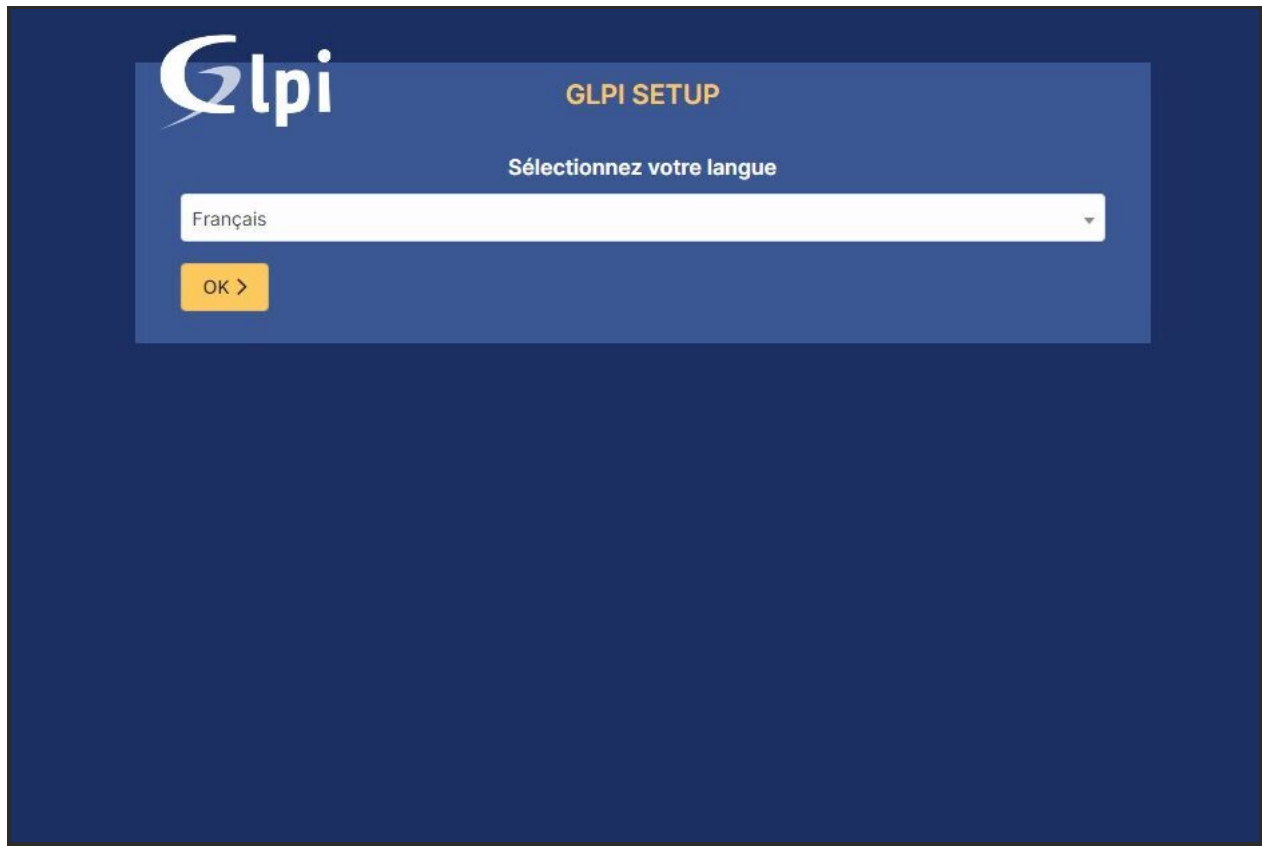
```
MariaDB [(none)]> create database glpi;  
Query OK, 1 row affected (0.000 sec)  
  
MariaDB [(none)]> create user glpiuser identified by 'Btssio32';  
Query OK, 0 rows affected (0.001 sec)  
  
MariaDB [(none)]> grant all privileges on glpi.* to glpiuser;  
Query OK, 0 rows affected (0.000 sec)  
  
MariaDB [(none)]> flush privileges;  
Query OK, 0 rows affected (0.000 sec)  
  
MariaDB [(none)]> exit;  
Bye
```

Rendez vous sur un navigateur et allez à l'adresse suivante :  
<http://192.168.1.100/glpi>

---

# Interface GLPI

<http://192.168.1.100/glipi>



Il vous faudra ensuite accepter les termes de la Licence de GLPI.



Et cliquer sur installer.



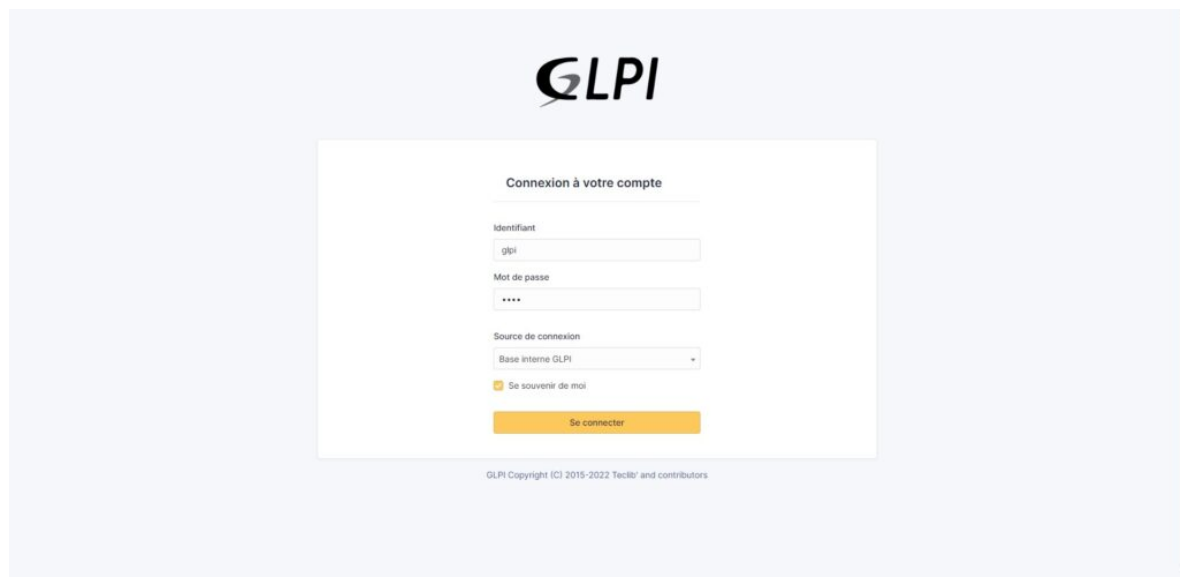
Un check des prérequis est fait, tout est coché hormis un petit triangle orange au milieu. L'erreur est normale, c'était déjà le cas sur GLPI 9, vous pouvez poursuivre l'installation.



Une fois connecté, sélectionnez la base de données nommée « glpi » précédemment créée.



Voici la magnifique page de connexion, l'utilisateur admin par défaut étant  
:identifiant : glpi    mot de passe : glpi

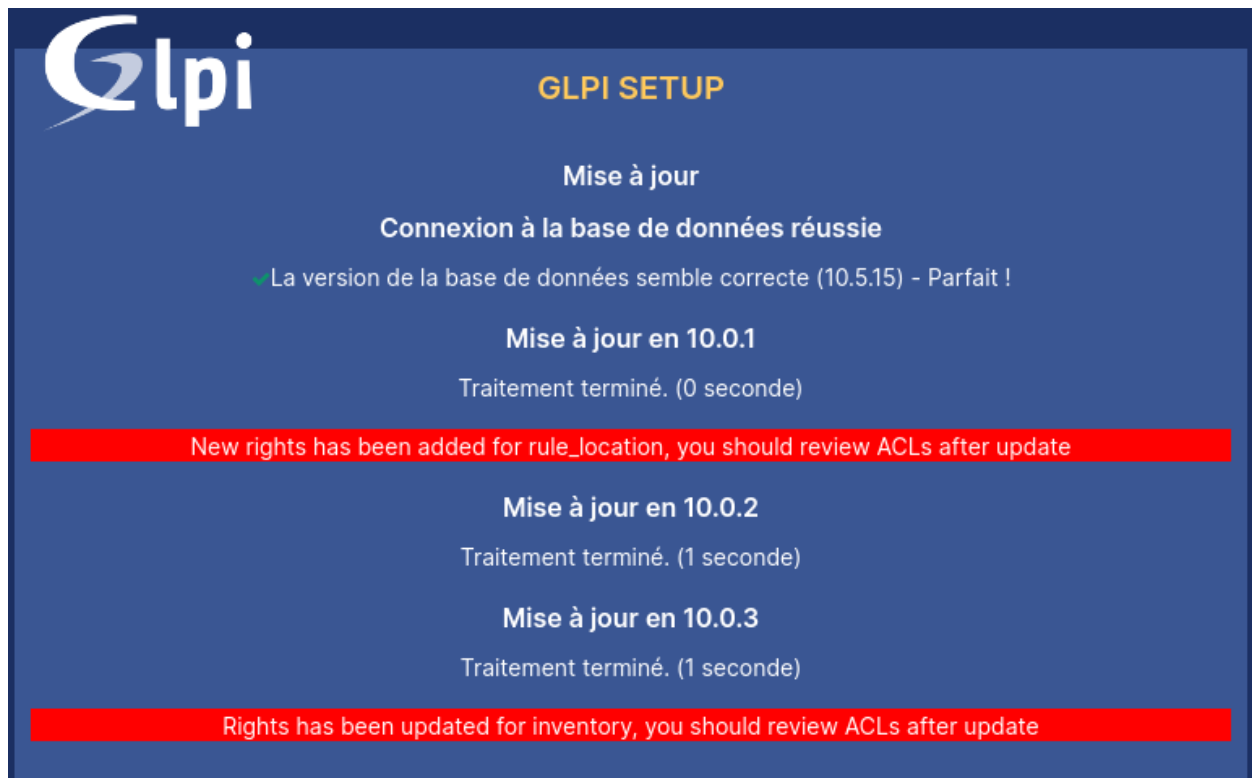


---

Nous avons ensuite installé la dernière mise à jour.

```
glpi@Machine-glpi:~$ su
Mot de passe :
root@Machine-glpi:/home/glpi# wget http://github.com/glpi-project/glpi/releases/download/10.0.3/glpi-10.0.3.tgz

root@Machine-glpi:/home/glpi# tar xzf glpi-10.0.3.tgz -C /var/www/html
root@Machine-glpi:/home/glpi# chown -R www-data:www-data /var/www/html/glpi
root@Machine-glpi:/home/glpi# chmod -R 775 /var/www/html/glpi
```



---

## Définir des utilisateurs :

Super-Admin mdp : Btssio32

Admin mdp : Btssio1

technician : Btssio2

Hotliner : Btssio3

Observer : Btssio4

Self-Service : Btssio5

Et nous avons aussi entré des machines

### - 3 Ordinateurs Bureautique :

<input type="checkbox"/> post 1	1044-1122	Dell	D 211	2022-10-17 08:51
<input type="checkbox"/> post 2	1133-7984	Dell	D 211	2022-10-17 08:55
<input type="checkbox"/> post 3	4555-2456	Dell	D 211	2022-10-17 08:56

### - 2 Stations CAO/DAO :

<input type="checkbox"/> Serveur 1				2022-10-17 09:27
<input type="checkbox"/> Serveur 2				2022-10-17 09:27

### - 2 Portables

<input type="checkbox"/> portable 1	Toshiba / Dynabook Satellite Pro C50-J-128	D 211	2022-10-17 09:16
<input type="checkbox"/> portable 2	Toshiba / Dynabook Satellite Pro C50-J-128	D 211	2022-10-17 09:17

## - 6 Écrans :

<input type="checkbox"/> écran 4	2022-10-17 09:39
<input type="checkbox"/> écran1	2022-10-17 09:31
<input type="checkbox"/> écran2	2022-10-17 09:34
<input type="checkbox"/> écran3	2022-10-17 09:38
<input type="checkbox"/> écran5	2022-10-17 09:40
<input type="checkbox"/> écran6	2022-10-17 09:41

## - 3 Tablettes :

<input type="checkbox"/> tablette 1	d111	Apple iPad Air(2022)	2022-10-17 09:44
<input type="checkbox"/> tablette 2		Apple iPad Air(2022)	2022-10-17 09:44
<input type="checkbox"/> tablette 3	d109	Apple iPad Air(2022)	2022-10-17 09:45

**Ci dessous, vous installer l'agent fusion dans le serveur Debian  
??!!**

```
root@Machine-glp: /home/glp# apt-get install fusioninventory-agent
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 :
  hdparm libfile-which-perl libnet-cups-perl libnet-ip-perl libparse-edid-perl
  libproc-daemon-perl libproc-processtable-perl libsocket-getaddrinfo-perl
  libtext-template-perl libuniversal-require-perl libxml-treepp-perl
  libxml-xpath-perl libyaml-libyaml-perl libyaml-perl libyaml-tiny-perl
  net-tools powermgmt-base
Paquets suggérés :
  smartmontools read-edid cups-common libyaml-shell-perl
Les NOUVEAUX paquets suivants seront installés :
  fusioninventory-agent hdparm libfile-which-perl libnet-cups-perl
  libnet-ip-perl libparse-edid-perl libproc-daemon-perl
  libproc-processtable-perl libsocket-getaddrinfo-perl libtext-template-perl
  libuniversal-require-perl libxml-treepp-perl libxml-xpath-perl
```



```
root@Machine-glpi:/home/glpi# gedit /etc/fusioninventory/agent.cfg

(gedit:14209): dconf-WARNING **: 08:46:25.739: failed to commit changes to dconf
: La connexion est fermée

(gedit:14209): dconf-WARNING **: 08:46:25.742: failed to commit changes to dconf
: La connexion est fermée
Error creating proxy: La connexion est fermée (g-io-error-quark, 18)
Error creating proxy: La connexion est fermée (g-io-error-quark, 18)
Error creating proxy: La connexion est fermée (g-io-error-quark, 18)
Error creating proxy: La connexion est fermée (g-io-error-quark, 18)
Error creating proxy: La connexion est fermée (g-io-error-quark, 18)

(gedit:14209): dconf-WARNING **: 08:46:25.941: failed to commit changes to dconf
: La connexion est fermée

(gedit:14209): dconf-WARNING **: 08:46:25.941: failed to commit changes to dconf
: La connexion est fermée
```

```
# all defined values match default
# all commented values are examples
```

```
#
# Target definition options
#
```

```
# send tasks results to an OCS server
#server = http://server.domain.com/ocsinventory
# send tasks results to a FusionInventory for GLPI server
#server = http://http://192.168.1.110S/glpi/plugins/fusioninventory/
# write tasks results in a directory
#local = /tmp
```

```
#
# Task definition options
#
```

```
# disable software deployment tasks
#tasks = deploy
#tasks = inventory,deploy,inventory
```

```
#
```