

---

# Procédure d'installation OCS Inventory NG

*Open Computers and Software Inventory – Serveur + ocsreports*

---

<b>Établissement</b>	Direction des Systèmes d'Information
<b>Date</b>	Mardi 21 décembre 2021
<b>Auteur</b>	Adelphe Borges-Silva
<b>OS cible</b>	Linux/Ubuntu 20.04
<b>Dépendances</b>	Apache2, MariaDB, PHP 7.4, Perl

# 1. Présentation d'OCS Inventory NG

OCS Inventory NG (Open Computers and Software Inventory Next Generation) est un outil open source permettant de réaliser l'inventaire automatique du parc informatique : matériel, logiciels, systèmes d'exploitation, etc.

Il fonctionne en mode client/serveur : un agent léger est installé sur chaque poste client et remonte automatiquement ses informations vers le serveur OCS via une interface web (ocsreports). Ces données peuvent ensuite être importées dans GLPI.

<b>Logiciel</b>	OCS Inventory NG (Open Computers and Software Inventory)
<b>Rôle</b>	Inventaire automatique du parc informatique (matériel + logiciels)
<b>Composants</b>	Serveur OCS + Agent OCS (installé sur les postes clients)
<b>Base de données</b>	MariaDB / MySQL
<b>Interface web</b>	ocsreports (PHP + Apache)
<b>Intégration</b>	Compatible GLPI via import OCS
<b>OS serveur</b>	Linux/Ubuntu 20.04
<b>Dépôt GitHub</b>	<a href="https://github.com/OCSInventory-NG/OCSInventory-Server">https://github.com/OCSInventory-NG/OCSInventory-Server</a>

## 2. Architecture du système



Schéma : flux Agent OCS (postes clients) → Serveur OCS → Import GLPI

- L'agent OCS est installé sur chaque poste du domaine (Windows ou Linux).
- Il communique avec le serveur OCS via HTTP/HTTPS sur le port 80 ou 443.
- Le serveur OCS stocke les données dans MariaDB et les expose via ocsreports.
- GLPI importe ces données pour enrichir son inventaire de parc.

## 3. Prérequis – Composants nécessaires

Composant	Rôle	Statut
Apache2 + mod_perl	Serveur web + support Perl	✓ Requis

MariaDB	Base de données SQL	✔ Requis
PHP 7.4 + extensions	Interface web ocsreports	✔ Requis
Perl + modules CPAN	Communication OCS serveur	✔ Requis
Git + curl + unzip	Téléchargement des sources OCS	✔ Requis
Composer	Gestion des dépendances PHP	✔ Requis
make / cmake / gcc	Compilation des composants OCS	✔ Requis

**⚠ Tous ces composants doivent être installés et opérationnels avant de lancer le script setup.sh.**

## 4. Installation pas à pas

### 01 Mise à jour du système

Commencer par mettre à jour les paquets du système :

```
$ sudo apt update
$ sudo apt upgrade
```

### 02 Installation des paquets de base

Installer les outils de compilation, Apache, MariaDB, PHP 7.4 et ses extensions :

```
$ sudo apt install make cmake gcc git curl unzip -y
$ sudo apt-get install apache2 mariadb-server libapache2-mod-perl2 \
libapache2-dbi-perl libapache2-db-perl php7.4 libapache2-mod-php7.4 \
php7.4-common php7.4-sqlite3 php7.4-mysql php7.4-gmp php7.4-curl \
php7.4-mbstring php7.4-gd php7.4-cli php7.4-xml php7.4-zip \
php7.4-soap php7.4-json php-pclzip composer
```

### 03 Configuration de PHP (php.ini)

Modifier le fichier php.ini pour ajuster la mémoire, la taille des uploads et le fuseau horaire :

```
$ sudo nano /etc/php/7.4/apache2/php.ini
```

Rechercher et modifier les valeurs suivantes :

```
memory_limit = 256M
post_max_size = 100M
upload_max_filesize = 100M
max_execution_time = 360
date.timezone = Europe/Paris
```

**⚠ Le fuseau horaire correct est 'Europe/Paris' et non 'France/Paris' (valeur invalide qui provoque des erreurs).**

## 04 Création de la base de données OCS

Se connecter à MariaDB en tant que root et créer la base de données ainsi que l'utilisateur dédié :

```
$ sudo mysql -u root -p
```

```
CREATE DATABASE ocs;
CREATE USER 'userocs'@'localhost' IDENTIFIED BY 'Mot_de_passe_complexe';
GRANT ALL ON ocs.* TO 'userocs'@'localhost' WITH GRANT OPTION;
FLUSH PRIVILEGES;
EXIT;
```

⚠ **Remplacez 'Mot\_de\_passe\_complexe' par un mot de passe fort. Notez-le, il sera nécessaire pour configurer setup.sh.**

## 05 Installation des modules Perl

Installer les modules Perl requis par OCS via apt :

```
$ sudo apt install perl libxml-simple-perl libcompress-zlib-perl libdbi-
perl \
  libdbd-mysql-perl libnet-ip-perl libsoap-lite-perl libio-compress-perl \
  libapache-dbi-perl libapache2-mod-perl2 libapache2-mod-perl2-dev -y
```

Activer ensuite les modules complémentaires via CPAN :

```
$ sudo perl -MCPAN -e 'install Apache2::SOAP'
$ sudo perl -MCPAN -e 'install XML::Entities'
$ sudo perl -MCPAN -e 'install Net::IP'
$ sudo perl -MCPAN -e 'install Apache::DBI'
$ sudo perl -MCPAN -e 'install Mojolicious'
$ sudo perl -MCPAN -e 'install Switch'
$ sudo perl -MCPAN -e 'install Plack::Handler'
```

## 06 Clonage et installation depuis GitHub

Cloner les dépôts OCS Inventory depuis GitHub dans le répertoire /opt :

```
$ sudo apt install git curl
$ cd /opt
$ sudo git clone https://github.com/OCSInventory-NG/OCSInventory-Server.git
$ cd /opt/OCSInventory-Server
$ git clone https://github.com/OCSInventory-NG/OCSInventory-ocsreports.git
$ cd ocsreports
$ sudo curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin --filename=composer
```

Si l'installation de Composer est réussie, le message suivant s'affiche :

```
All settings correct for using Composer
Downloading...
Composer (version 2.0.4) successfully installed to: /usr/local/bin/composer
Use it: php /usr/local/bin/composer
```

## 07 Configuration du fichier setup.sh

Revenir dans le répertoire OCSInventory-Server et éditer le fichier de configuration :

```
$ cd /opt/OCSInventory-Server
$ sudo nano setup.sh
```

Renseigner les informations de la base de données créée à l'étape 4 :

```
DB_SERVER_HOST="localhost"
DB_SERVER_PORT="3306"
DB_SERVER_USER="userocs"
DB_SERVER_PWD="Mot_de_passe_complexe"
```

Sauvegarder et quitter le fichier (Ctrl+X → O → Entrée).

## 08 Lancement du script d'installation

Lancer le script setup.sh pour installer OCS Inventory sur le serveur :

```
$ sudo ./setup.sh
```

Le script est interactif : valider chaque étape en appuyant sur Entrée ou en tapant 'y'. Si tout est correct, le message de bienvenue OCS s'affiche.

WARNING: You will not be able to build any deployment package with size greater than 100MB  
You must raise both `post_max_size` and `upload_max_filesize` in your `httpd` configuration to increase this limit.

WARNING: If you change default database name (`ocsweb`) or user (`ocs`), don't forget to update the file `'3-ocsinventory-server.conf'` in your `Apache` configuration directory

ERROR: Can't connect to MySQL. Please enter a valid login/password. (host=localhost login=userocs password=ocspassword password=3306 password=0 password= password= password=)  
ERROR: MySQL connection problem 1045  
Access denied for user 'userocs'@'localhost' (using password: YES)

MySQL login:

MySQL password:

Name of Database:

MySQL HostName:

MySQL Port:

Enable SSL:

SSL mode:

SSL key path:

SSL certificate path:

CA certificate path:

Capture : script `setup.sh` en cours d'exécution

**i** Si le script signale un module Perl manquant, l'installer via `cpan` puis relancer `setup.sh` :

```
$ sudo cpan Nom::Du::Module # ex: sudo cpan Net::IP  
$ sudo ./setup.sh # relancer après installation du module
```

## 09 Activation des accès Apache

Créer les liens symboliques pour activer la configuration OCS dans Apache :

```
$ sudo ln -s /etc/apache2/conf-available/ocsinventory-reports.conf \
    /etc/apache2/conf-enabled/ocsinventory-reports.conf
$ sudo ln -s /etc/apache2/conf-available/z-ocsinventory-server.conf \
    /etc/apache2/conf-enabled/z-ocsinventory-server.conf
$ sudo ln -s /etc/apache2/conf-available/zz-ocsinventory-restapi.conf \
    /etc/apache2/conf-enabled/zz-ocsinventory-restapi.conf
```

Définir les permissions sur le répertoire de rapports OCS :

```
$ sudo chown -R www-data:www-data /var/lib/ocsinventory-reports
```

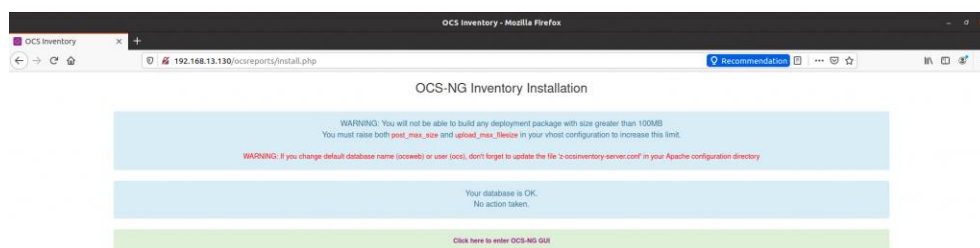
Redémarrer Apache pour appliquer toutes les modifications :

```
$ sudo systemctl restart apache2
```

## 10 Finalisation via l'interface web

Ouvrir un navigateur et accéder à l'interface d'installation web d'OCS :

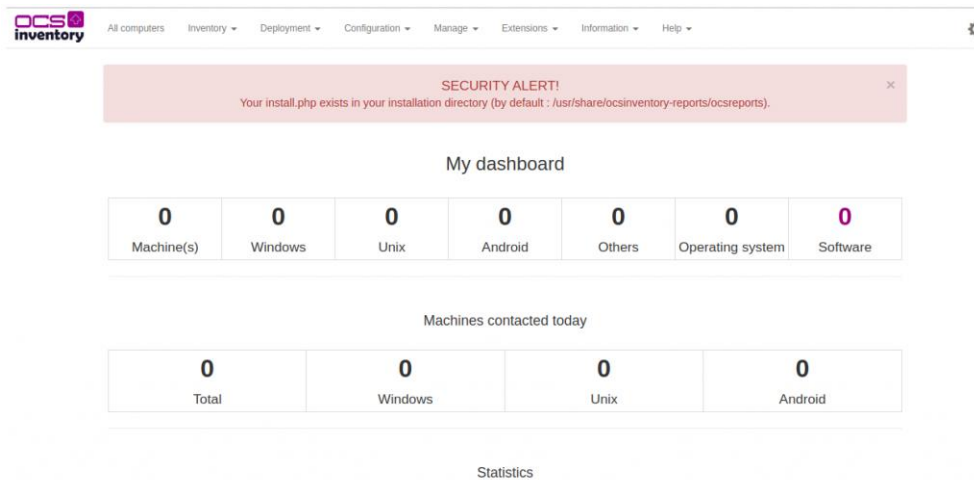
```
http://localhost/ocsreports/install.php
```



Capture : interface d'installation web d'OCS Inventory

**i** Si la page affiche une page blanche, exécuter les commandes suivantes puis rafraîchir le navigateur :

```
$ cd /usr/share/ocsinventory-reports/ocsreports  
$ composer install
```



Capture : interface OCS Inventory opérationnelle

## 5. Sécurisation post-installation

Une fois OCS Inventory installé et fonctionnel, plusieurs actions de sécurisation sont obligatoires avant toute mise en production.

### 5.1 Changer le mot de passe par défaut

Lors de la première connexion, se rendre dans Administration > Utilisateurs et modifier immédiatement le mot de passe du compte admin (identifiants par défaut : admin / admin).

**⚠ Ne jamais laisser le mot de passe par défaut 'admin' en production. C'est une faille de sécurité critique.**

### 5.2 Supprimer le fichier install.php

Le fichier install.php ne doit plus être accessible une fois l'installation terminée. Le supprimer pour éviter les alertes sur le tableau de bord et empêcher une réinstallation accidentelle :

```
$ sudo rm -f /usr/share/ocsinventory-reports/ocsreports/install.php
```

### 5.3 Vérifier l'accès à MariaDB

S'assurer que l'utilisateur 'userocs' n'est accessible que depuis localhost :

```
$ sudo mysql -u root -p
SELECT user, host FROM mysql.user WHERE user='userocs';
EXIT;
```

Le champ 'host' doit afficher 'localhost' uniquement.

### 5.4 Configurer le pare-feu (UFW)

Restreindre les ports accessibles depuis l'extérieur :

```
$ sudo ufw allow 80/tcp      # HTTP (agents OCS)
$ sudo ufw allow 443/tcp    # HTTPS (optionnel)
```

```
$ sudo ufw allow 22/tcp      # SSH (administration)
$ sudo ufw deny 3306/tcp    # Bloquer MariaDB depuis l'extérieur
$ sudo ufw enable
```

## 6. Checklist de validation finale

Utiliser cette liste pour valider chaque étape avant de mettre OCS Inventory en production :

✓	Tâche	Phase	Statut
<input type="checkbox"/>	Système Linux/Ubuntu mis à jour (apt update + upgrade)	Prérequis	À valider
<input type="checkbox"/>	Paquets de base installés (gcc, make, cmake, git, curl)	Prérequis	À valider
<input type="checkbox"/>	Apache2 + mod_perl installés et opérationnels	Prérequis	À valider
<input type="checkbox"/>	PHP 7.4 et toutes ses extensions installés	Prérequis	À valider
<input type="checkbox"/>	php.ini configuré (mémoire, upload, fuseau Europe/Paris)	Configuration	À valider
<input type="checkbox"/>	MariaDB installé et base 'ocs' créée	Base de données	À valider
<input type="checkbox"/>	Utilisateur 'userocs' créé avec les bons privilèges	Base de données	À valider
<input type="checkbox"/>	Modules Perl installés et activés via CPAN	Installation	À valider
<input type="checkbox"/>	Dépôt OCS-NG cloné depuis GitHub dans /opt	Installation	À valider
<input type="checkbox"/>	Fichier setup.sh configuré avec les infos BDD	Installation	À valider
<input type="checkbox"/>	Script setup.sh exécuté avec succès	Installation	À valider
<input type="checkbox"/>	Liens symboliques Apache créés (conf-enabled)	Configuration	À valider
<input type="checkbox"/>	Permissions /var/lib/ocsinventory-reports définies	Configuration	À valider
<input type="checkbox"/>	Apache redémarré après configuration	Configuration	À valider
<input type="checkbox"/>	Interface web accessible sur /ocsreports/install.php	Tests	À valider
<input type="checkbox"/>	Installation finalisée via l'interface web	Tests	À valider
<input type="checkbox"/>	Fichier install.php supprimé (sécurité)	Sécurisation	À valider
<input type="checkbox"/>	Mot de passe par défaut OCS modifié	Sécurisation	À valider
<input type="checkbox"/>	Pare-feu UFW configuré (ports 80, 443, 22 ouverts)	Sécurisation	À valider
<input type="checkbox"/>	Premier agent OCS remonte bien dans l'interface	Tests finaux	À valider
<input type="checkbox"/>	Import OCS → GLPI fonctionnel	Tests finaux	À valider

**⚠ Ne mettre en production qu'une fois toutes les cases cochées. Penser à supprimer install.php et à changer le mot de passe admin.**