

## Activité 1

### Point 3 : Installer OCSinventory et procédez à une première configuration assistée.

Passage administrateur:

```
-su  
-debian
```

Mise à jour du serveur:

```
-apt update  
-apt install software-properties-common gnupg2  
-apt upgrade  
-reboot
```

Installation des premiers paquets:

```
-apt -y install git make cmake gcc make build-essential
```

Installation de mariadb:

```
-apt install mariadb-server  
-mysql_secure_installation
```

Installation des modules apache:

```
-apt install libapache2-mod-perl2 libapache-dbi-perl libapache-db-perl  
libapache2-mod-php
```

Installation des modules php:

```
-apt install php  
-apt install php-zip php-pclzip php-gd php-mysql php-soap php-curl php-json php-xml  
php-mbstring
```

Installation de perl et des modules perl:

```
-apt install -y 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  
-cpan install XML::Entities Apache2::SOAP Net::IP Apache::DBI Mojolicious Switch  
Plack::Handler Archive::Zip  
-cpan install Parse::EDID Compress::Zlib
```

Configuration de php:

```
-nano /etc/php/7.0/apache2/php.ini  
    short_open_tag = On  
    max_execution_time = 360  
    memory_limit = 256M  
    post_max_size = 100M  
    file_uploads = On  
    upload_max_filesize = 100M  
    date.timezone = Europe/Paris
```

-Création de la base de donnée OCS:

- mysql -u root -p
- create database ocsdb;
- create user ocsuser@localhost identified by 'ocspwd';
- grant all on ocsdb.\* to ocsuser@localhost;
- flush privileges;
- exit

-Téléchargement de OCSInventory server

- cd /opt
- git clone https://github.com/OCSInventory-NG/OCSInventory-Server.git
- cd OCSInventory-Server
- git clone https://github.com/OCSInventory-NG/OCSInventory-ocsreports.git

ocsreports

- cd ocsreports

Installation de composer:

- curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin

--filename=composer

- composer install
- cd ..
- nano setup.sh
- DB\_SERVER\_USER="ocs" -> "ocsuser"
- DB\_SERVER\_PWD="ocs" -> "ocspwd"
- ./setup.sh
- dire oui à tout
- ln -s /etc/apache2/conf-available/ocsinventory-reports.conf

/etc/apache2/conf-enabled/ocsinventory-reports.conf

- ln -s /etc/apache2/conf-available/z-ocsinventory-server.conf

/etc/apache2/conf-enabled/z-ocsinventory-server.conf

- ln -s /etc/apache2/conf-available/zz-ocsinventory-restapi.conf

/etc/apache2/conf-enabled/zz-ocsinventory-restapi.conf

-Configuration de OCSInventory:

- chown -R www-data:www-data /var/lib/ocsinventory-reports
- systemctl restart apache2
- Trouver son IP via ip a (192.168.139.169/24 le 25/03)
- 192.168.139.169/ocsreports/install.php sur navigateur
- Remplir les champs :
  - MySQL login : ocsuser
  - MySQL password: ocspwd
  - Name of Database: ocsdb
  - MySQL HostName: localhost

-Pas la peine de remplir les champs SSL pour l'instant et appuyer sur le bouton send

- pb rencontrer: je n'avais pas écrit la commande des privileges de ocsuser

- On reçoit un compte pour ocsreports

- login= admin et password= admin

- on clique sur le lien "Click here to enter OCS-NG GUI" qui nous renvoie vers  
192.168.139.169/ocsreports/index.php

- la page demande une mise à jour que l'on accepte avec "Perform the update"
- On click sur le nouveau lien "Click here to enter OCS-NG GUI"
- se connecter avec le compte admin admin
  - un message d'erreur apparaît car le fichier install.php n'a pas été renommé

et n'importe qui peut reconfigurer ocsreport !

- cd /usr/share/ocsinventory-reports/ocsreports
- mv install.php

/usr/share/ocsinventory-reports/ocsreports/install.php.bak

- systemctl restart apache2
- cd etc/apache2/conf-available
- nano z-ocsinventory-server.conf
  - PerlSetEnv OCS\_DB\_NAME ocsweb -> ocscdb
  - PerlSetEnv OCS\_DB\_LOCAL ocsweb -> ocscdb
  - PerlSetEnv OCS\_DB\_USER ocs -> ocscuser
  - PerlSetEnv OCS\_DB\_NAME ocs -> ocspwd
- sudo nano zz-ocsinventory-restapi.conf
  - \$ENV{OCS\_DB\_LOCAL} = 'ocsweb' ; -> 'ocscdb' ;
  - \$ENV{OCS\_DB\_USER} = 'ocs' ; -> 'ocscuser' ;
  - \$ENV{OCS\_DB\_PWD} = 'ocs' ; -> 'ocspwd' ;
- sudo systemctl restart apache2

## Etape 3 :

### Point 1 :

Installation de l'agent OCS pour Ubuntu Avant de télécharger et d'installer l'agent nous allons mettre à jour les repository Ubuntu.

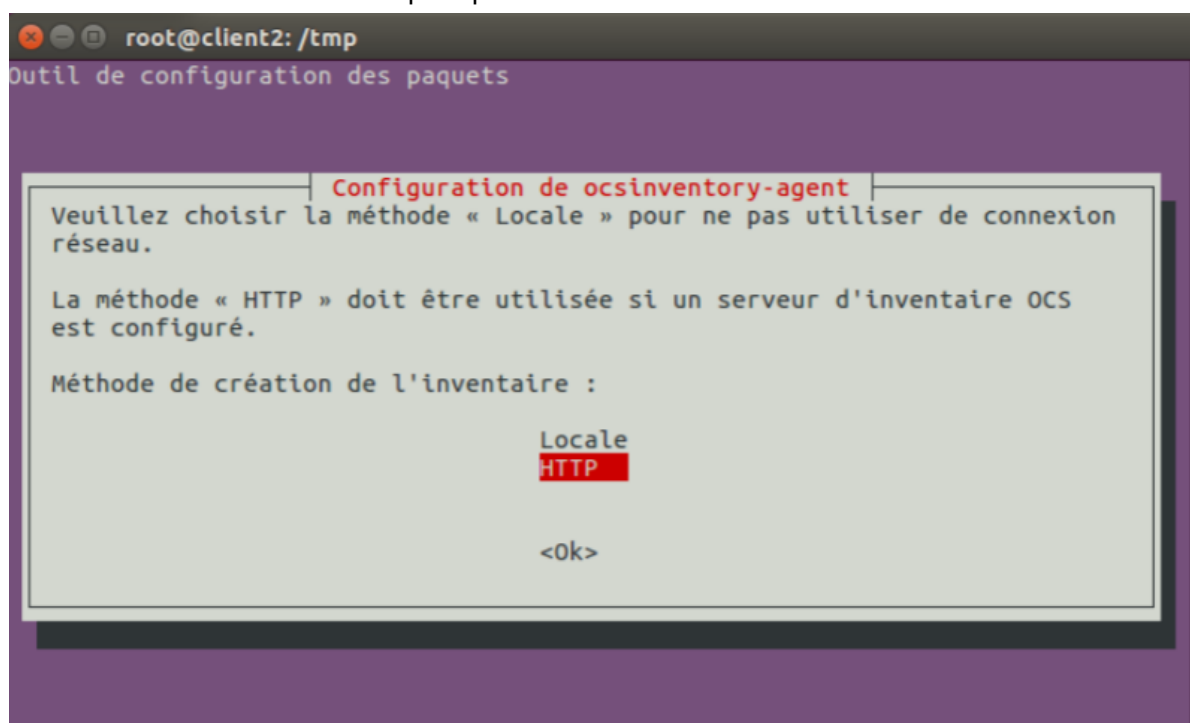
Pour cela tapez les commandes suivant :

- sudo apt-get update (saisissez le mot de passe super-utilisateur – « 1234Azerty » dans notre cas
- sudo apt-get upgrade (saisissez le mot de passe super-utilisateur – « 1234Azerty » dans notre cas Une fois la mise à jour terminée nous allons installer l'agent OCS Inventory Agent, pour cela taper ceci :

- sudo apt-get install ocsinventory-agent

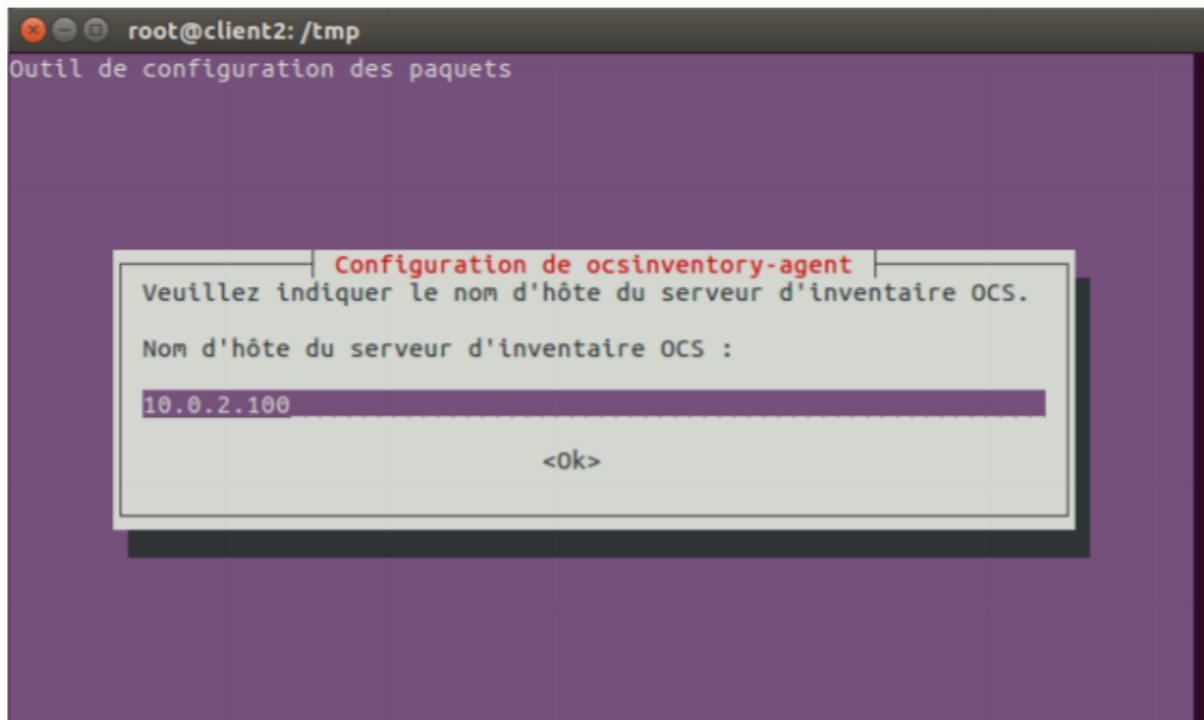
Une fois le téléchargement et l'installation terminée, vous arriverez sur cette fenêtre :

- choisissez la méthode HTTP puis presser « Entrer »



La fenêtre suivante apparaît :

- saisissez l'adresse IP du serveur qui héberge OCS Inventory NG Serveur .



Si vous souhaitez enregistrer un TAG pour votre machine au moment de la remontée d'information vers le serveur il vous suffit de taper la commande suivante afin de re-configurer l'agent :

Afin de faire remonter immédiatement les informations vers le serveur utilisez la commande suivante :

- `sudo ocsinventory-agent`

L'agent est alors installé, configuré et les informations sont enregistrées sur le serveur OCS.

## **Point 2 : Insérez chaque élément du parc informatique de la MLH.**

création d'une VM ubuntu

Changement du nom de la machine : MLH\_1

mise à jour ainsi que `apt update/upgrade`

`sudo apt install ocsinventory-agent`

- dire que la méthode est http dans prompt

- donner l'adresse du serveur ocs (192.168.139.254)

`sudo dpkg-reconfigure ocsinventory-agent`

- revérifié la config et choisir un tag si voulu (non)

`sudo ocsinventory-agent`

- informe le serveur

Clone lié de MLH\_1, renommé MLH\_2

`sudo ocsinventory-agent`

répéter jusqu'à MLH\_9

## Activité 2 :

### Point 1 :

On commence par copier le certificat dans le répertoire de l'interface web car on en a besoin dans le serveur et les clients pour que le télédéploiement soit disponible.

```
-cp server.crt /usr/share/ocsinventory-reports/ocsreports/
```

Installer curl sur un Client

```
-sudo apt update  
-sudo apt upgrade  
-sudo apt install curl
```

Télécharger le certificat

```
-cd /var/lib/ocsinventory-agent/http:___192.168.139.254_ocsinventory  
-sudo curl -o server.crt http://192.168.139.254/server.crt
```

Pour transmettre le certificat chez les ocsinventory-agent des clients, il faut:

```
-archiver le certificat en tar.gz  
-mkdir certificat  
-cp server.crt ./certificat/server.crt  
-tar -czvf certificat.tar.gz ./certificat
```

Nous avons ensuite la possibilité d'utiliser la fonction de télédéploiement d'un paquet. Pour ce faire, depuis l'interface web de ocsinventory-server, création d'un télédéploiement d'un paquet

Nom et description : peu importe

choisir le fichier certificat.tar.gz

l'action est stocker

le chemin est /var/lib/ocsinventory-agent/http:\_\_\_192.168.139.254\_ocsinventory

(attendre que Gaëtan est vérifié que ça marche, autre solutions potentiellement nécessaire)

## Activité 3 :

### Point 1 : Installation et configuration de GLPI

Liste des commandes

Installation

```
-cd /usr/share  
-su  
-debian  
-wget https://github.com/glpi-project/glpi/releases/download/9.5.4/glpi-9.5.4.tgz  
-tar -xvzf glpi-9.5.4.tgz  
-chown -R www-data ./glpi
```

Configuration

```
-mysql -u root -p  
-debian  
-create database glpidb;  
-grant all privileges on glpidb.* to glpiuser@localhost identified by "glpipwd";  
-exit
```

(le site n'est pas encore accessible, donc je recherche)

Liste des commandes troubleshooting

- apt update
- apt upgrade

Le serveur français d'apt de debian n'est pas accessible: solution téléchargement manuel des paquets

- wget

[http://ftp.de.debian.org/debian/pool/main/a/apcupsd/apcupsd-doc\\_3.14.14-2\\_all.deb](http://ftp.de.debian.org/debian/pool/main/a/apcupsd/apcupsd-doc_3.14.14-2_all.deb)

- dpkg -i apcupsd-doc\_3.14.14-2\_all.deb
- wget <https://pecl.php.net/get/apcu-5.1.20.tgz>
- tar -xvzf apcu-5.1.20.tgz

Suivis du tutoriel

[https://www.youtube.com/watch?v=57NXlwrM2yA&ab\\_channel=SebastienMazzei](https://www.youtube.com/watch?v=57NXlwrM2yA&ab_channel=SebastienMazzei)

- su
- debian
- wget <https://github.com/glpi-project/glpi/releases/download/9.5.4/glpi-9.5.4.tgz>
- tar -xvzf glpi-9.5.4.tgz
- rm glpi-9.5.4.tgz
- chown -R www-data glpi/
- mysql -u root -p
- debian
  - create user 'userglpi'@'%' identified by 'pwdglpi';
  - grant all privileges on \*.\* to 'userglpi'@'%';
  - flush privileges;
  - exit
- cd /etc/apache2/sites-available/
- cp 000-default.conf 001-GLPI.conf
- nano 0001-GLPI.conf
  - DocumentRoot /var/www/html -> /var/www/html/glpi
- a2ensite
  - 001-GLPI
- systemctl reload apache2
- a2dissite 000-default.conf
- systemctl reload apache2

<https://www.quora.com/How-do-I-change-the-server-that-Debian-uses-to-download-package-s>

pour apt

Pour mettre à jour de php

- apt install apt-transport-https lsb-release ca-certificates wget -y
- wget -O /etc/apt/trusted.gpg.d/php.gpg <https://packages.sury.org/php/apt.gpg>
- sh -c 'echo "deb <https://packages.sury.org/php/> \$(lsb\_release -sc) main" >

/etc/apt/sources.list.d/php.list"

- apt update

```
-apt install php8.0-common php8.0-cli -y
-apt install libapache2-mod-php8.0 php8.0 php8.0-curl php8.0-gd php8.0-mbstring
php8.0-mysql php8.0-opcache php8.0-readline php8.0-soap php8.0-xml php8.0-zip
-apt install php8.0-ldap php8.0-xmlrpc php8.0-APCu
```