

Compte rendu d'Installation

« OwnCloud »

Table des matières

Objectif(s) :	2
Légende :	2
OwnCloud :	3
1) Configuration :	3
2) Mise en place de OwnCloud :	4
Installation des différents paquets :	4
Téléchargement de OwnCloud 9 et SHA256.....	4
Vérification de l'intégrité des paquets téléchargés :	6
Extraction du paquet OwnCloud :	6
Configuration de OwnCloud :	6
Configuration du OwnCloud Client	13
Sécurisation du trafic en SSL.....	14
Synchronisation Owncloud et l'Active directory :	16
Configuration Active directory :	16
Configuration Owncloud :	20

Objectif(s) :

L'objectif de ce tutoriel est de configurer OwnCloud afin d'accéder, de partager des fichiers, des calendriers, des contacts, des mails, à partir de n'importe quel appareils.

Légende :

- Les commandes ou les chemins (absolue/relatif) sont en gras, souligné et en italique ex :
 - ***Apt-get update***
- Des captures d'écrans ont été prises afin de faciliter la compréhension du lecteur.

Machine	Os	Distribution	Version	C/S	IP
OwnCloud	Debian	Linux	8.7	S	192.168.1.141

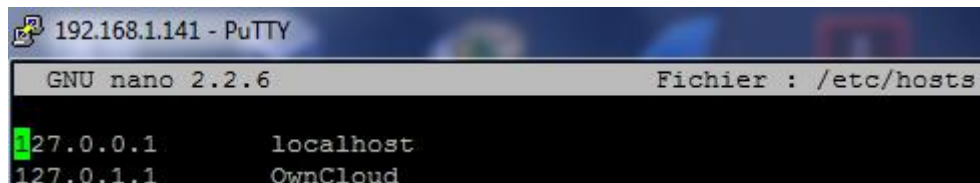
OwnCloud :

1) Configuration :

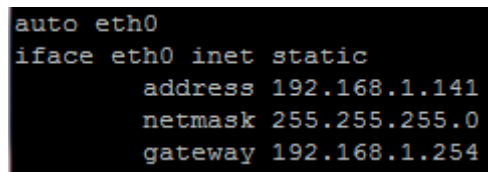
On va configurer pour commencer la machine :



```
192.168.1.141 - PuTTY
GNU nano 2.2.6 Fichier : /etc/hostname
OwnCloud
```



```
192.168.1.141 - PuTTY
GNU nano 2.2.6 Fichier : /etc/hosts
127.0.0.1 localhost
127.0.1.1 OwnCloud
```



```
auto eth0
iface eth0 inet static
    address 192.168.1.141
    netmask 255.255.255.0
    gateway 192.168.1.254
```

Avant de Commencer l'installation, nous allons mettre à jour la VM et les différents paquets :

apt-get update

apt-get upgrade

apt-getdist-upgrade

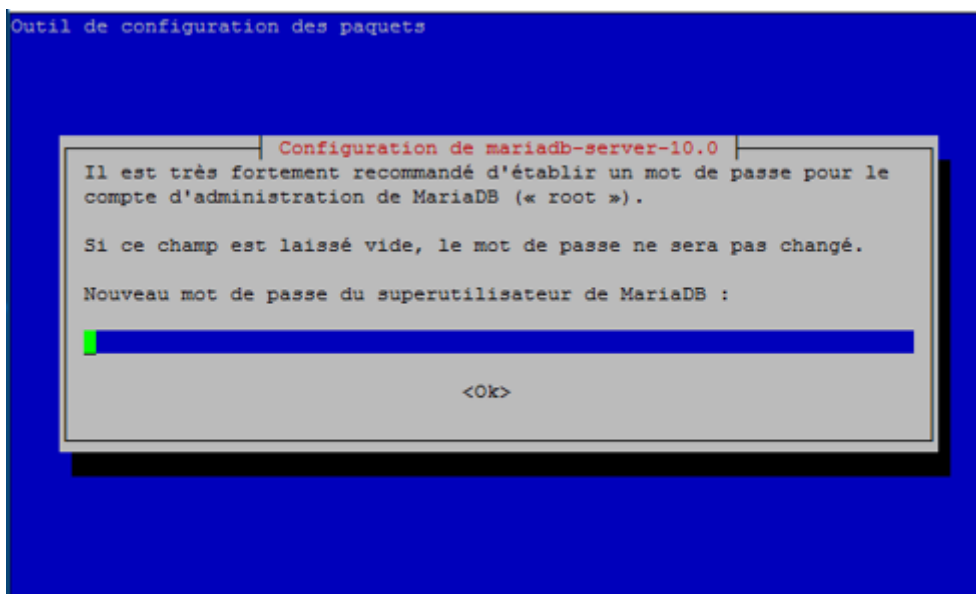
2) Mise en place de OwnCloud :

Installation des différents paquets :

On va tout d'abord installer tous les paquets :

apt-get install mariadb-server mariadb-client apache2 libapache2-mod-php5 php5-json php5-gd php5-mysql php5-curl php5-intl php5-mcrypt php5-imagick

La configuration de maria dB-server se lance donc peu de temps après la demande d'installation de paquets.



Le Mot de passe sera Root1234

Téléchargement de OwnCloud 9 et SHA256

On va maintenant télécharger OwnCloud 9 et aussi SHA256 hash, ce logiciel permet de vérifier et d'authentifier OwnCloud.

wget <https://download.owncloud.org/community/owncloud-9.0.0.tar.bz2>

```

root@OwnCloud:/home/letort# wget https://download.owncloud.org/community/owncloud-9.0.0.tar.bz2
--2017-02-02 14:11:50-- https://download.owncloud.org/community/owncloud-9.0.0.tar.bz2
Résolution de download.owncloud.org (download.owncloud.org)... 144.76.105.220, 46.4.80.187, 148.251.209.106, ...
Connexion à download.owncloud.org (download.owncloud.org)|144.76.105.220|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 22678208 (22M) [application/x-bzip2]
Sauvegarde en : « owncloud-9.0.0.tar.bz2 »

owncloud-9.0.0.tar. 100%[=====>] 21,63M 754KB/s ds 52s

2017-02-02 14:12:43 (423 KB/s) - « owncloud-9.0.0.tar.bz2 » sauvegardé [22678208/22678208]

```

wget <https://download.owncloud.org/community/owncloud-9.0.0.tar.bz2.sha256>

```

root@OwnCloud:/home/letort# wget https://download.owncloud.org/community/owncloud-9.0.0.tar.bz2
--2017-02-02 14:11:50-- https://download.owncloud.org/community/owncloud-9.0.0.tar.bz2
Résolution de download.owncloud.org (download.owncloud.org)... 144.76.105.220, 46.4.80.187, 148.251.209.106, ...
Connexion à download.owncloud.org (download.owncloud.org)|144.76.105.220|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 22678208 (22M) [application/x-bzip2]
Sauvegarde en : « owncloud-9.0.0.tar.bz2 »

owncloud-9.0.0.tar.b 28%[=====>] 6,12M 619KB/s eta 27s

```

wget <https://owncloud.org/owncloud.asc>

```

root@OwnCloud:/home/letort# wget https://download.owncloud.org/community/owncloud-9.0.0.tar.bz2.sha256
--2017-02-02 14:14:16-- https://download.owncloud.org/community/owncloud-9.0.0.tar.bz2.sha256
Résolution de download.owncloud.org (download.owncloud.org)... 46.4.80.187, 85.10.210.219, 148.251.209.106, ...
Connexion à download.owncloud.org (download.owncloud.org)|46.4.80.187|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 89 [application/x-bzip2]
Sauvegarde en : « owncloud-9.0.0.tar.bz2.sha256 »

owncloud-9.0.0.tar. 100%[=====>] 89 --.-KB/s ds 0s

2017-02-02 14:14:17 (3,04 MB/s) - « owncloud-9.0.0.tar.bz2.sha256 » sauvegardé [89/89]

```

wget <https://download.owncloud.org/community/owncloud-9.0.0.tar.bz2.sha256>

```

root@OwnCloud:/home/letort# wget https://owncloud.org/owncloud.asc
--2017-02-02 14:15:18-- https://owncloud.org/owncloud.asc
Résolution de owncloud.org (owncloud.org)... 213.239.207.28, 2a01:4f8:130:806f::5
Connexion à owncloud.org (owncloud.org)|213.239.207.28|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 3117 (3,0K) [application/octet-stream]
Sauvegarde en : « owncloud.asc »

owncloud.asc      100%[=====>]   3,04K  --.-KB/s   ds 0s
2017-02-02 14:15:19 (62,3 MB/s) - « owncloud.asc » sauvegardé [3117/3117]

```

Vérification de l'intégrité des paquets téléchargés :

On va maintenant vérifier l'intégrité du téléchargement du paquet grâce au sha256sum :

sha256sum -c owncloud-9.0.0.tar.bz2.sha256

```

root@OwnCloud:/home/letort# sha256sum -c owncloud-9.0.0.tar.bz2.sha256
owncloud-9.0.0.tar.bz2: Réussi

```

Si le test échoué il ne faut pas continuer, ça veut dire que l'archive téléchargé n'est pas complet !

La prochaine étape est d'utiliser GnuPG pour vérifier l'authenticité du software et pour cela on va importer une clé PGP public dans notre GnuPG :

gpg --import owncloud.asc

```

root@OwnCloud:/home/letort# gpg --import owncloud.asc
gpg: répertoire « /root/.gnupg » créé
gpg: nouveau fichier de configuration « /root/.gnupg/gpg.conf » créé
gpg: Attention : les options de « /root/.gnupg/gpg.conf » ne sont pas encore actives cette fois
gpg: le porte-clefs « /root/.gnupg/secring.gpg » a été créé
gpg: le porte-clefs « /root/.gnupg/pubring.gpg » a été créé
gpg: /root/.gnupg/trustdb.gpg : base de confiance créée
gpg: clef F6978A26 : clef publique « ownCloud <info@owncloud.com> » importée
gpg:      Quantité totale traitée : 1
gpg:      importées : 1 (RSA: 1)

```

Extraction du paquet OwnCloud :

Nous allons maintenant extraire le package software :

tar -xjvf owncloud-9.0.0.tar.bz2

Configuration de OwnCloud :

On va ensuite copier les dossiers ownCloud dans les documents root d'Apache 2. Les documents qu'utilise le daemon Apache2 se situe sur le Debian 8 sur [/var/www/html](#)

`cp -r owncloud /var/www/html`

```
cp -r owncloud /var/www/html
```

On va maintenant ajouter OwnCloud dans les sites disponibles d'Apache2. Avant de toucher quoi que ce soit et de tout casser on va d'abord faire une copie du fichier « configuration default site ».

```
root@OwnCloud:~# cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/owncloud.conf
```

```
root@OwnCloud:~# ls -l /etc/apache2/sites-available
total 16
-rw-r--r-- 1 root root 1332 janv. 25 22:16 000-default.conf
-rw-r--r-- 1 root root 6437 févr. 11 09:57 default-ssl.conf
-rw-r--r-- 1 root root 1534 avril  4 11:17 owncloud.conf
```

On va maintenant aller dans le fichier `/etc/apache2/sites-available/owncloud.conf`

Ici on va :

- S'assurer que le `ServerName` est bien décommenté
- Il va ensuite sur cette même ligne mettre le nom de hostname de la machine, soit `OwnCloud`.
- Puis on va rajouter à la fin document tout ce qu'il y a en jaune sur le screen ci-dessous.

```
GNU nano 2.2.6 Fichier : /etc/apache2/sites-available/owncloud.conf
<VirtualHost *:80>
# The ServerName directive sets the request scheme, hostname and port that
# the server uses to identify itself. This is used when creating
# redirection URLs. In the context of virtual hosts, the ServerName
# specifies what hostname must appear in the request's Host: header to
# match this virtual host. For the default virtual host (this file) this
# value is not decisive as it is used as a last resort host regardless.
# However, you must set it for any further virtual host explicitly.
ServerName OwnCloud

ServerAdmin webmaster@localhost
DocumentRoot /var/www/html

# Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
# error, crit, alert, emerg.
# It is also possible to configure the loglevel for particular
# modules, e.g.
#LogLevel info ssl:warn

ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined

<Directory /var/www/owncloud/>
Options +FollowSymlinks
AllowOverride All

<IfModule mod_dav.c>
Dav off
</IfModule>

SetEnv HOME /var/www/owncloud
SetEnv HTTP_HOME /var/www/owncloud

</Directory>

# For most configuration files from conf-available/, which are
# enabled or disabled at a global level, it is possible to
# include a line for only one particular virtual host. For example the
# following line enables the CGI configuration for this host only
# after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Une fois le fichier OwnCloud configuré on va aller dans les fichiers [/etc/apache2/sites-enabled](#) et faire un lien symbolique !

[ln -s /etc/apache2/sites-available/owncloud.conf /etc/apache2/sites-enabled/owncloud.conf](#)

```
root@OwnCloud:~# ln -s /etc/apache2/sites-available/owncloud.conf /etc/apache2/sites-enabled/owncloud.conf
```

La prochaine étape est d'activer les modules d'Apache2 grâce à la commande [a2enmod](#).

[a2enmod rewrite](#)

De plus les [modules headers, env, dir, mime](#) sont recommandés, nous allons donc les activer :

[a2enmod headers](#)

[a2enmod env](#)

[a2enmod dir](#)

[a2enmod mime](#)

```
root@OwnCloud:/home/letort# a2enmod headers
Enabling module headers.
To activate the new configuration, you need to run:
  service apache2 restart
root@OwnCloud:/home/letort# a2enmod env
Module env already enabled
root@OwnCloud:/home/letort# a2enmod dir
Module dir already enabled
root@OwnCloud:/home/letort# a2enmod mime
Module mime already enabled
root@OwnCloud:/home/letort# █
```

Nous allons maintenant activer notre site OwnCloud et redémarrer Apache2 afin que les configurations préalablement faits soient pris en compte.

[a2ensite owncloud](#)

[service apache2 restart](#)

```
root@OwnCloud:/home/letort# a2ensite owncloud
Site owncloud already enabled
root@OwnCloud:/home/letort# service apache2 restart
root@OwnCloud:/home/letort# █
```

La prochaine manipulation sera de mettre les bonnes permissions sur les dossiers d'ownCloud. En effet pour que tout soit correct il faut que HTTP user/group aient les permissions de changer dans les fichiers de ownCloud. Sur Debian le HTTP user/group c'est www-data. On va donc donner aux fichiers de owncloud le http user/group, www-data.

Voici les droits avant de les changer :

```
root@OwnCloud:/var/www/html# ls -l
total 16
-rw-r--r--  1 root root 10701 févr.  2 12:23 index.html
drwxr-xr-x 14 root root  4096 févr.  2 15:04 owncloud
root@OwnCloud:/var/www/html#
```

chown -R www-data:www-data /var/www/html/owncloud/

Voici les nouveaux droits HTTP user/group du dossier :

```
root@OwnCloud:/var/www/html# chown -R www-data:www-data /var/www/html/owncloud/
root@OwnCloud:/var/www/html# ls -l
total 16
-rw-r--r--  1 root  root  10701 févr.  2 12:23 index.html
drwxr-xr-x 14 www-data www-data 4096 févr.  2 15:04 owncloud
root@OwnCloud:/var/www/html#
```

Différentes permissions sont ensuite nécessaires pour le bon déroulement du OwnCloud :

- ✓ All files should be read-write for the file owner, read-only for the group owner, and not accessible to others
- ✓ All directories should be executable, read-write for the directory owner, and read-only for the group owner
- ✓ The **apps/** directory should be owned by **[HTTP user]:[HTTP group]**
- ✓ The **config/** directory should be owned by **[HTTP user]:[HTTP group]**
- ✓ The **themes/** directory should be owned by **[HTTP user]:[HTTP group]**
- ✓ The **assets/** directory should be owned by **[HTTP user]:[HTTP group]**
- ✓ The **data/** directory should be owned by **[HTTP user]:[HTTP group]**
- ✓ The **[ocpath]/.htaccess** file should be owned by **root:[HTTP group]**
- ✓ The **data/.htaccess** file should be owned by **root:[HTTP group]**
- ✓ Both **.htaccess** files should be read-write for the file owner, read-only for the group owner, and not accessible to others

Pour appliquer toutes ces permissions un script a été créé. Nous allons le placer dans et par la suite le rendre exécutable et l'exécuter !

```
#!/bin/bash
ocpath='/var/www/html/owncloud'
htuser='www-data'
htgroup='www-data'
rootuser='root'

printf "Creating possible missing Directories\n"
mkdir -p $ocpath/data
mkdir -p $ocpath/assets

printf "chmod Files and Directories\n"
find ${ocpath}/ -type f -print0 | xargs -0 chmod 0640
find ${ocpath}/ -type d -print0 | xargs -0 chmod 0750

printf "chown Directories\n"
```

```

chown -R ${rootuser}:${htgroup} ${ocpath}/
chown -R ${htuser}:${htgroup} ${ocpath}/apps/
chown -R ${htuser}:${htgroup} ${ocpath}/config/
chown -R ${htuser}:${htgroup} ${ocpath}/data/
chown -R ${htuser}:${htgroup} ${ocpath}/themes/
chown -R ${htuser}:${htgroup} ${ocpath}/assets/

chmod +x ${ocpath}/occ

printf "chmod/chown .htaccess\n"
if [ -f ${ocpath}/.htaccess ]
then
    chmod 0644 ${ocpath}/.htaccess
    chown ${rootuser}:${htgroup} ${ocpath}/.htaccess
fi
if [ -f ${ocpath}/data/.htaccess ]
then
    chmod 0644 ${ocpath}/data/.htaccess
    chown ${rootuser}:${htgroup} ${ocpath}/data/.htaccess
fi

```

```

GNU nano 2.2.6 Fichier : OC-perms.sh

#!/bin/bash
ocpath='/var/www/html/owncloud'
htuser='www-data'
htgroup='www-data'
rootuser='root'

printf "Creating possible missing Directories\n"
mkdir -p $ocpath/data
mkdir -p $ocpath/assets

printf "chmod Files and Directories\n"
find ${ocpath}/ -type f -print0 | xargs -0 chmod 0640
find ${ocpath}/ -type d -print0 | xargs -0 chmod 0750

printf "chown Directories\n"
chown -R ${rootuser}:${htgroup} ${ocpath}/
chown -R ${htuser}:${htgroup} ${ocpath}/apps/
chown -R ${htuser}:${htgroup} ${ocpath}/config/
chown -R ${htuser}:${htgroup} ${ocpath}/data/
chown -R ${htuser}:${htgroup} ${ocpath}/themes/
chown -R ${htuser}:${htgroup} ${ocpath}/assets/

chmod +x ${ocpath}/occ

printf "chmod/chown .htaccess\n"
if [ -f ${ocpath}/.htaccess ]
then
    chmod 0644 ${ocpath}/.htaccess
    chown ${rootuser}:${htgroup} ${ocpath}/.htaccess
fi
if [ -f ${ocpath}/data/.htaccess ]
then
    chmod 0644 ${ocpath}/data/.htaccess
    chown ${rootuser}:${htgroup} ${ocpath}/data/.htaccess
fi

```

Pour rendre le script exécutable :

```
chmod u+x /root /OC-perms.sh
```

```
/root/OC-perms.sh
```

Exécutons-le maintenant :

```
/home/letort/oc-perms.sh
```

```
root@OwnCloud:/var/www/html# /home/letort/oc-perms.sh
Creating possible missing Directories
chmod Files and Directories
chown Directories
chmod/chown .htaccess
root@OwnCloud:/var/www/html#
```

Nous allons maintenant nous occuper de la Database. Nous avons installé dès le début MariaDB. Nous allons donc rentrer dessus :

```
mysql -u root -p
```

Le mdp je le rappelle est le suivant : Root1

```
root@OwnCloud:/var/www/html# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.0.29-MariaDB-0+deb8u1 (Debian)

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
MariaDB [(none)]>
```

On va commencer par créer une database celle d'OwnCloud :

```
MariaDB [(none)]> create database owncloudDB
-> ;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| owncloudDB |
| performance_schema |
+-----+
4 rows in set (0.00 sec)
```

On va ensuite créer un utilisateur autorisé à se connecter à distance :

Grant all privileges on *.* to 'remote_user'@'192.168.1.141' identified by 'user_password';

- *.* représente l'accès à l'ensemble des bases de données, si vous voulez autoriser l'accès de l'utilisateur à une seule base de données, remplacez *.* par le nom de la base de données.
- **remote_user** est le nom de l'utilisateur qui sera créé.
- **192.168.1.141** représente l'IP à partir de laquelle nous pourrions nous connecter à distance.
- Pour autoriser l'accès venant de toutes machines (limité par l'authentification), remplacer l'ip par le symbole '%'
- Pour attribuer un mot de passe à l'utilisateur, changer **user_password** par le nouveau mot de passe.

Pour rendre ces privilèges effectifs :

#flush privileges;

On a donc fait :

```
MariaDB [(none)]> use owncloudDB;
Database changed
MariaDB [owncloudDB]> grant all privileges on owncloudDB to 'root'@'192.168.1.141' identified by 'Password654123';
Query OK, 0 rows affected (0.00 sec)

MariaDB [owncloudDB]> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```


Une fois cela fait on va quitter la database

quit

Configuration du OwnCloud Client

Nous allons nous diriger sur notre navigateur et nous allons écrire cela :

<http://192.168.1.141/owncloud>



The screenshot shows the OwnCloud installation configuration interface. At the top is the OwnCloud logo. Below it, the text "Créer un compte administrateur" is displayed. There are two input fields: the first contains "admin" and the second contains a masked password ".....". A red error message "Mot de passe très faible" is shown below the password field. Below the password field is a dropdown menu for "Stockage & base de données" with a downward arrow. Underneath is the text "Répertoire des données" followed by an input field containing "/var/www/html/owncloud/d". Below this is the text "Configurer la base de données". A message states: "Seul(e) MySQL/MariaDB est disponible. Installez et activez les modules PHP additionnels adéquats pour choisir d'autres types de base de données. Consultez la documentation pour plus de détails." Below this message are three input fields: the first contains "root", the second contains a masked password ".....", the third contains "owncloudDB", and the fourth contains "172.17.0.199". At the bottom is a large blue button labeled "Terminer l'installation". At the very bottom, there is a link: "i Besoin d'aide ? Lire la documentation ↗".

On arrive donc sur cette page :

Les identifiants sont les suivant : Admin / Password1234

Il faut ensuite mettre le nom d'utilisateur/MDP/et la base.

A la place de mettre 172.17.0.199, on va mettre **localhost**.

Un endroit sûr pour toutes vos données

Consultez et partagez vos fichiers, agendas, carnets d'adresses, emails et bien plus depuis les appareils de votre choix, sous vos conditions.

Obtenez les applications vous permettant de synchroniser vos fichiers



Connectez vos applications de bureau à ownCloud

📅 Connectez votre calendrier 📇 Connectez vos contacts 🖱️ Accédez à vos fichiers via WebDAV

Des informations complémentaires sont disponibles dans la documentation et sur notre site web.
Si vous aimez ownCloud, recommandez-le à vos amis et faites passer le mot !

La configuration est finit et l'utilisation de OwnCloud et maintenant possible.

Sécurisation du trafic en SSL

Pour sécurié noter OwnCloud nous allons Activé le SSL sur Apache2.

a2enmod ssl

a2ensite default-ssl

Nous allons ensuite redémarrer Apache 2 pour activer la nouvelle configuration

service apache2 restart

Une configuration par SSL par défaut est déjà présente dans apache2 **/etc/apache2/sites-available/default-ssl.conf**

Nous allons donc copier l'intérieur de ce fichier et le coller en haut de notre **/etc/apache2/sites-enabled/owncloud.conf**

Dans notre nouveau document il faut rajouter une ligne ServerName 'myhostname' et il faut que notre DocumentRoot pointe vers /var/www/html/

```
GNU nano 2.2.6 Fichier : /etc/apache2/sites-enabled/owncloud.conf
IfModule mod_ssl.c>
  <VirtualHost _default_:443>
    ServerAdmin webmaster@localhost

    DocumentRoot /var/www/html

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn
    ServerName OwnCloud
```

Une fois cela fait la connexion en HTTPS est maintenant possible :

<https://192.168.1.141/owncloud>

Nous arrivons sur cette page :

La connexion n'est pas sécurisée

Les propriétaires de 192.168.1.141 ont mal configuré leur site web. Pour éviter que vos données ne soient dérobées, Firefox ne s'est pas connecté à ce site web.

[En savoir plus...](#)

[Retour](#) [Avancé](#)

Signaler les erreurs similaires pour aider Mozilla à identifier et bloquer les sites malveillants

Il suffit simplement de faire avancer, c'est un certificat auto-signé et donc « pas sûr ». Cependant nous sommes les créateurs donc nous savons qu'il est sûr. Nous allons donc continuer sur le site.

https://172.17.0.199/owncloud/index.php

Nom d'utilisateur

Mot de passe →

Rester connecté

Nous sommes donc maintenant en https. Il suffit maintenant de rentrer les ID et nous serons sur notre OwnCloud.

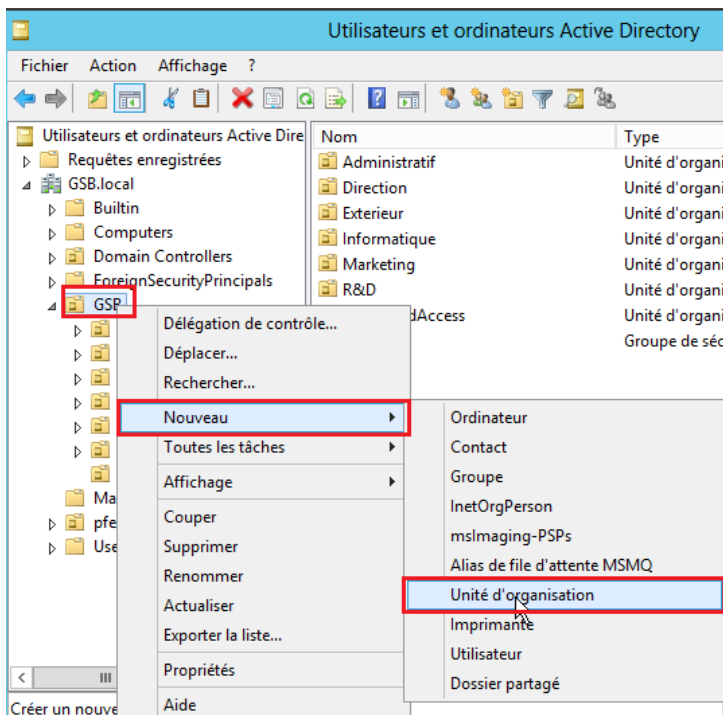
Synchronisation Owncloud et l'Active directory :

Les prérequis pour faire une synchronisation d'Owncloud et de l'Active directory :

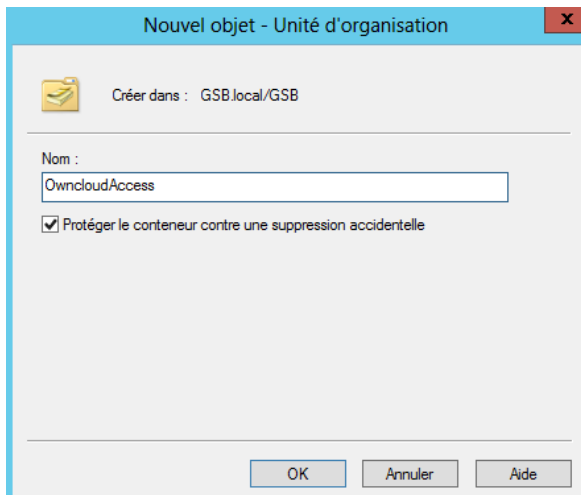
- ✓ Serveur Owncloud
- ✓ Active directory Windows
- ✓ Installation du paquet « php5-ldap ».

Configuration Active directory :

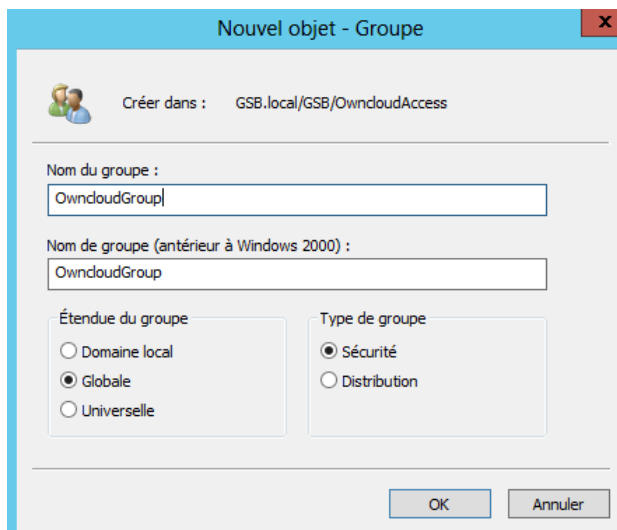
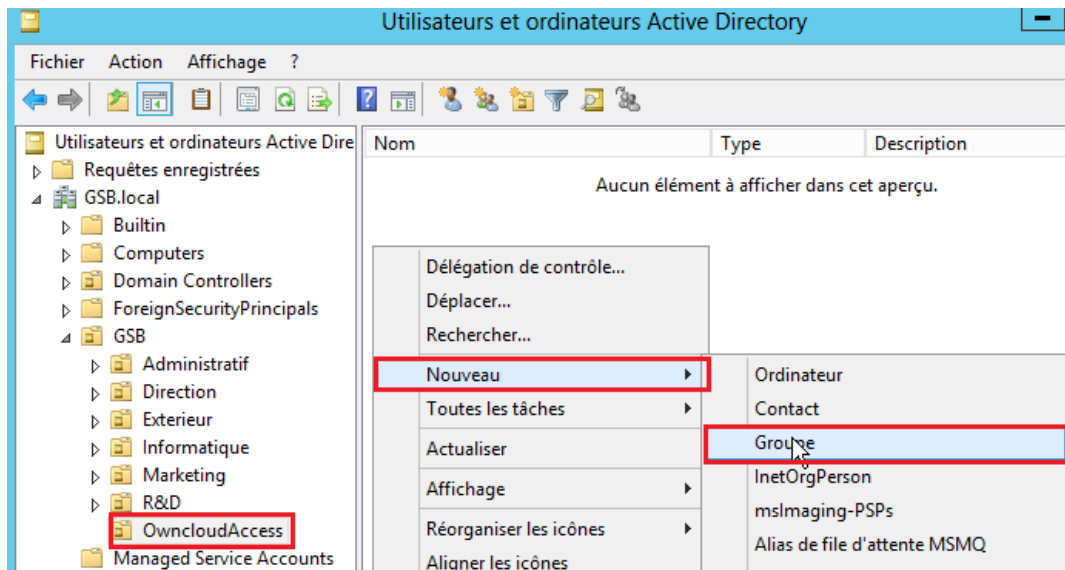
- Création d'une OU appelée "OwncloudAccess" :



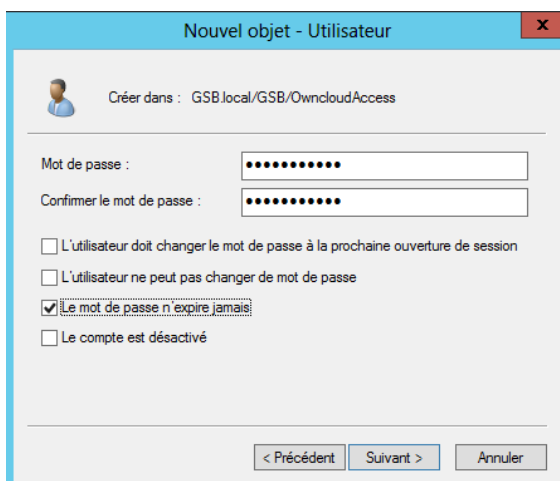
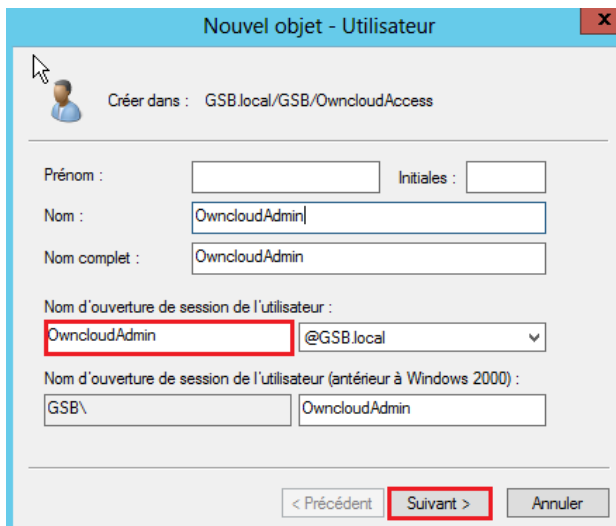
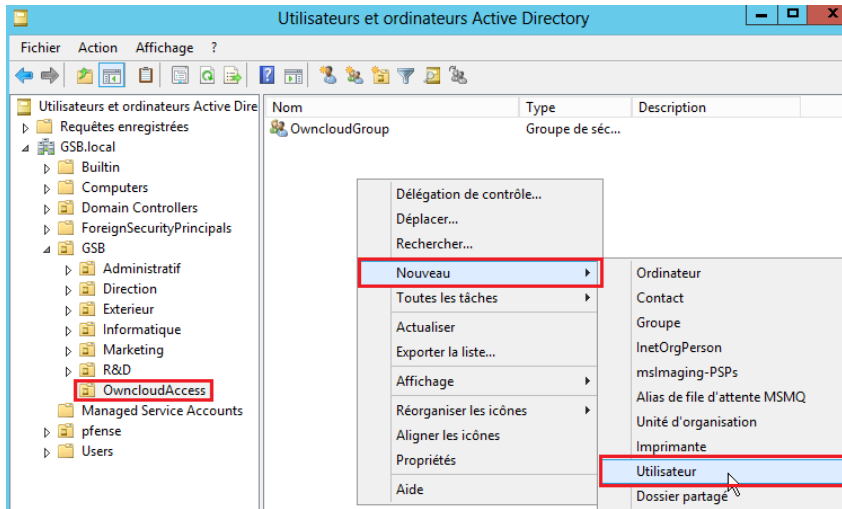
Il suffira ensuite de faire ok :



- Création d'un groupe appelé "OwncloudGroup" stocké dans OwncloudAccess



- Création d'un utilisateur appelé "OwncloudAdmin" stocké dans OwncloudAccess



MDP : Password123

Nous allons aussi créer un autre utilisateur leoletort avec le même mot de passe :

Nouvel objet - Utilisateur

Créer dans : GSB.local/GSB/OwncloudAccess

Prénom : leo Initiales :

Nom : letort

Nom complet : leo letort

Nom d'ouverture de session de l'utilisateur : leoletort @GSB.local

Nom d'ouverture de session de l'utilisateur (antérieur à Windows 2000) : GSB\ leoletort

< Précédent Suivant > Annuler

- Ajout des utilisateurs concernés par Owncloud et OwncloudAdmin dans le groupe "OwncloudGroup"

Nom	Type	Description
leo letort	Utilisateur	
OwncloudAdmin	Utilisateur	
OwncloudGroup	Groupe de sécurité - Global	

Propriétés de : OwncloudGroup

Général Membres Membre de Géré par

Membres :

Nom	Dossier Services de domaine Active Directory
leo letort	GSB.local/GSB/OwncloudAccess
OwncloudAd...	GSB.local/GSB/OwncloudAccess

Ajouter... Supprimer

OK Annuler Appliquer

Notre dernier prérequis était l'installation de « php5-ldap »

```
root@OwnCloud:~#  
root@OwnCloud:~# apt install php5-ldap_
```

Configuration Owncloud :

Il faut que notre machine possède le DNS de notre serveur AD :

```
OwnCloud_SyncLDAP [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Input  Périphériques  Aide
GNU nano 2.2.6      Fichier : /etc/network/interfaces

allow-hotplug eth0

#iface eth0 inet dhcp

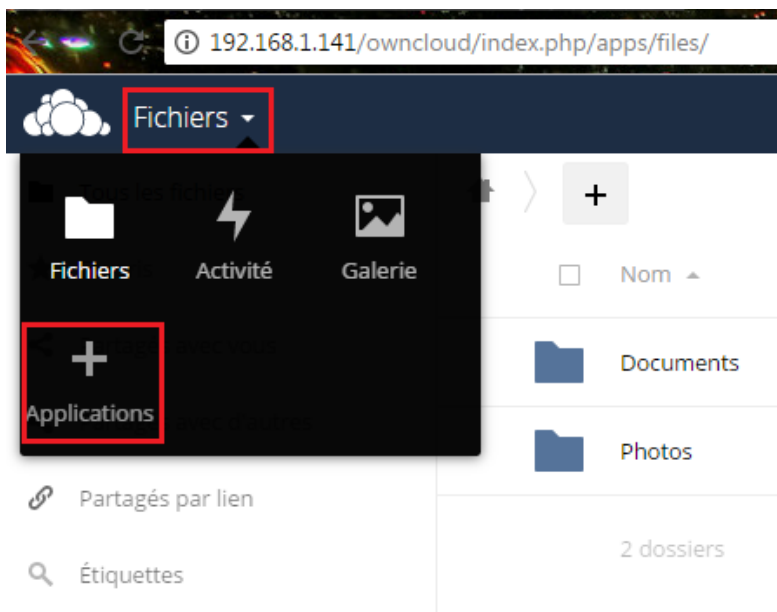
auto eth0
iface eth0 inet static
    address 192.168.1.141
    netmask 255.255.255.0
    gateway 192.168.1.254
    nameserver 192.168.1.130
    nameserver 192.168.1.110
```

```
OwnCloud_SyncLDAP [En fonction] - Oracle VM VirtualBox
Fichier  Machine  Écran  Input  Périphériques  Aide
GNU nano 2.2.6      Fichier : /etc/resolv.conf

domain gsb.local
search gsb.local
nameserver 192.168.1.130
nameserver 192.168.1.110
nameserver 192.168.1.49
nameserver 192.168.1.50
nameserver 8.8.8.8
```

On va maintenant se connecter sur la page Owncloud : 192.168.1.141/owncloud/

Allez ensuite dans « Application, et activer LDAP user and group backend » :





LDAP user and group backend 0.8.0

par Dominik Schmidt and Arthur Schiwon (Sous licence AGPL)

✓ Officielle

Afficher la description...

Désactiver

On va ensuite se diriger vers Admin en haut a droite, puis administration, puis LDAP :

LDAP

2. Serveur + [icône] [icône]

Hôte Port Détecter le port

DN Utilisateur

Mot de passe

Un DN de base par ligne Détecter le DN de base Tester le DN de base

Saisir les filtres LDAP manuellement (recommandé pour les annuaires de grande ampleur)

Remplir avec les informations suivantes :

Hôte : ServeurAD.domaine

DN Utilisateur : cn=OwncloudAdmin,ou=OwncloudAccess,dc=domaine

Mot de passe : mot de passe de OwncloudAdmin soit Password123

Un DN racine par ligne : dc=domaine

Comme ceci :

LDAP

1. Serveur : LABANNU1.GSB.local + [icône] [icône]

LABANNU1.GSB.local 389 Détecter le port

CN=OwncloudAdmin,OU=OwncloudAccess,OU=GSB,DC=GSB,DC=local

.....

DC=GSB,DC=local Détecter le DN de base Tester le DN de base

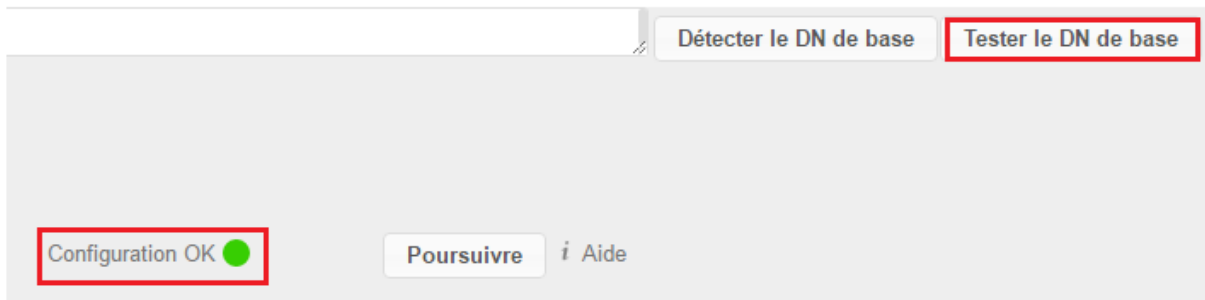
Saisir les filtres LDAP manuellement (recommandé pour les annuaires de grande ampleur)

Pour un accès anonyme, laisser le DN utilisateur et le mot de passe vides.

En faisant Détecter le port, on obtient cela :

389 Détecter le port

On va ensuite faire Tester le DN de base :



On va ensuite aller dans l'onglet « Utilisateurs ».

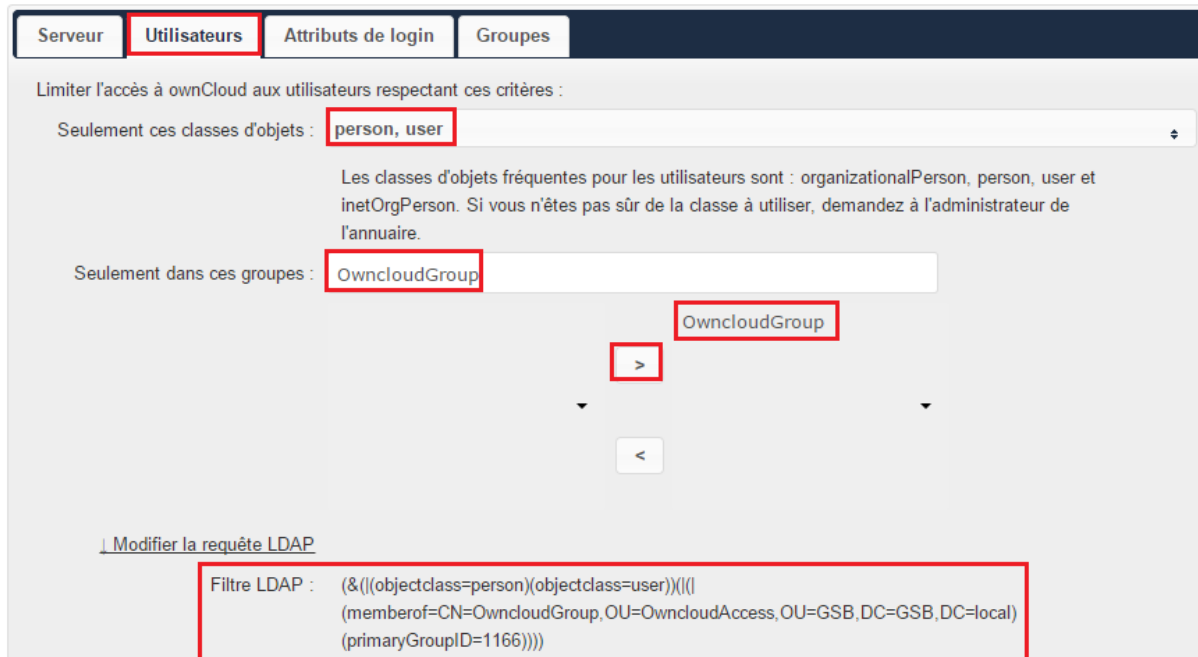
L'onglet `Utilisateurs` permet de configurer les paramètres de recherche / validation des utilisateurs sur l'annuaire.

On va limiter l'accès à Owncloud aux utilisateurs respectant ce critère :

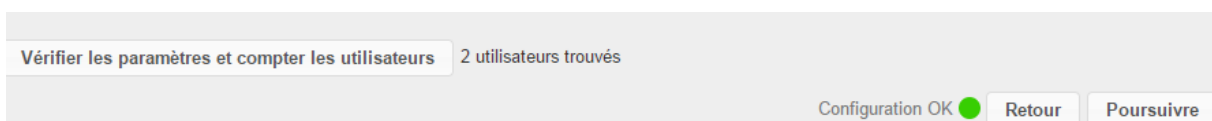
Seulement ces classes d'objets : Person, user.

Seulement dans ces groupes : OwncloudGroup.

LDAP



On va ensuite cliquer sur « vérifier les paramètres et compter les utilisateurs » :



Les deux utilisateurs trouvés sont donc : leoletort et OwncloudAdmin.

Nous allons donc pouvoir faire Poursuivre !

On arrive donc dans l'onglet Attribut de login :

L'onglet `Attributs de login` permet de configurer le mode de validation de l'utilisateur lors de la connexion. Encore une fois, cette fonctionnalité est plutôt bien pensée. En effet, il est possible de spécifier une authentification sur l'identifiant, l'adresse mail ou d'autres parmi la définition de la classe de filtre. Il est donc possible de l'effectuer sur l'id, des initiales etc.

- Nous allons donc cocher le nom d'utilisateur LDAP
- Et mettre dans les attributs : SamAccountName

LDAP

Serveur Utilisateurs **Attributs de login** Groupes

Au login, ownCloud cherchera l'utilisateur sur base de ces attributs :

Nom d'utilisateur LDAP / AD :

Adresse mail LDAP / AD :

Autres attributs : `sAMAccountName`

[Modifier la requête LDAP](#)

Filtre LDAP : `(&(&((objectclass=person)(objectclass=user))(|(memberof=CN=OwncloudGroup,OU=OwncloudAccess,OU=GSB,DC=GSB,DC=local)(primaryGroupID=1166))))|(samaccountname=%uid)(sAMAccountName=%uid))`

On va tester les paramètres avec l'utilisateur : OwncloudAdmin

OwncloudAdmin **Tester les paramètres**

Utilisateur trouvé et paramètres vérifiés.

On arrive donc dans l'onglet Groupes :

On va limiter l'accès à Owncloud aux groupes respectant ce critère :

LDAP

Les groupes respectant ces critères sont disponibles dans ownCloud :

Seulement ces classes d'objets : Sélectionner les classes d'objet

Seulement dans ces groupes : OwncloudGroup

OwncloudGroup

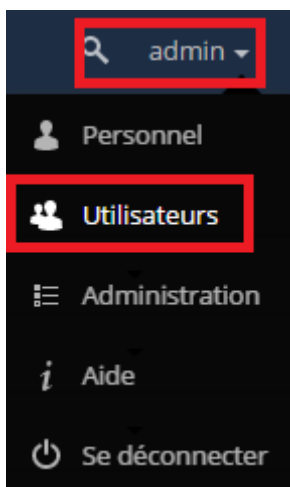
[Modifier la requête LDAP](#)

Filtre LDAP : ((cn=OwncloudGroup))

Vérifier les paramètres et compter les groupes 1 groupe trouvé

Configuration OK ● Retour

En allant dans Admin puis Utilisateurs :






Voici le résultat :

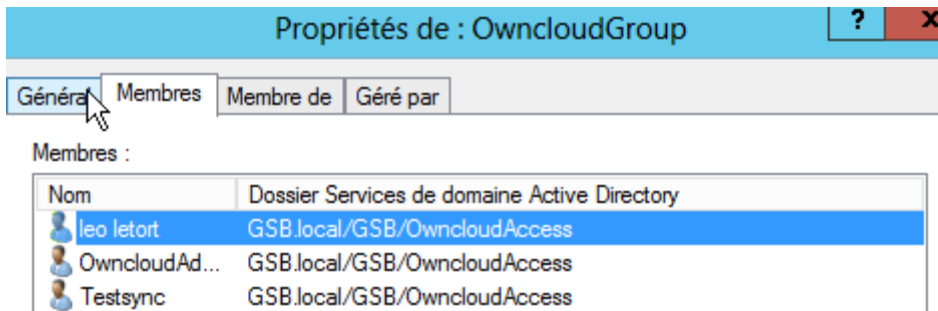
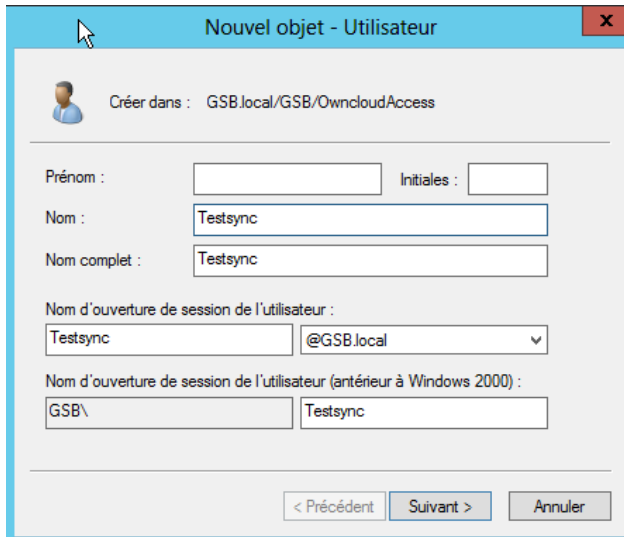
Nom d'utilisateur	Nom complet	Mot de passe	Groupes
4512C137-C1E3-478D-A395-2BABB5A17E6	leo letort	●●●●●●	OwncloudGroup
admin	admin	●●●●●●	admin
CC70CD0D-65B1-469F-B707-AE7A02F74622	OwncloudAdmin	●●●●●●	OwncloudGroup




On peut donc voir que notre Owncloud à synchroniser les utilisateurs de l'active Directory !

Test de vérification :





Nom	Type	Description
 leo letort	Utilisateur	
 OwncloudAdmin	Utilisateur	
 OwncloudGroup	Groupe de sécurité - Global	

On va donc créer un utilisateur « TestSync » sur l'AD puis on va l'ajouter au group OwncloudGroup



Nom	Dossier Services de domaine Active Directory
 leo letort	GSB.local/GSB/OwncloudAccess
 OwncloudAd...	GSB.local/GSB/OwncloudAccess
 Testsync	GSB.local/GSB/OwncloudAccess

On retourne donc sur notre Owncloud :

	Nom d'utilisateur	Nom complet	Mot de passe	Groupes
	4512C137-C1E3-478D-A395-2BABB5A17E6	leo letort	••••••	OwncloudGroup
	admin	admin	••••••	admin
	B0CAA5BE-C126-4FE4-9F39-786B691498E9	Testsync	••••••	OwncloudGroup
	CC70CD0D-65B1-469F-B707-AE7A02F74622	OwncloudAdmin	••••••	OwncloudGroup

Et la synchronisation s'est donc faite !