

OwnCloud : logiciel de cloud computing personnel

- Objet : Le cloud computing personnel avec ownCloud
- Niveau requis :
[débutant](#), [avisé](#)
- Commentaires : Installation de ownCloud
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[à-placer](#)
 - Création par [nifseg](#) et [chuugar](#) le 02/01/2014
 - Testé par [lr0nsh007er](#) le 13/07/2015
 - Ajout du depot pour owncloud stable (v9.x) [cemoi](#) le 31/07/2016
 - Ajout de la gestion du cache par [cemoi](#) le 1/08/2016
 - Ajout de la conf pour le fichier ../config.php (à terminer) par [cemoi](#) le 1/08/2016
 - Testé par [cemoi](#) le 1/08/2016
- Commentaires sur le forum : [ici](#) ¹⁾

Introduction

OwnCloud est un logiciel de services de stockage et d'applications variées en nuage sous licence AGPLv3. Plus communément appelé Cloud Computing.

Ce projet a été initialement lancé par Frank Karlitschek (développeur du projet KDE) en janvier 2010 . OwnCloud fait donc partie du projet KDE et de sa vaste communauté.

Contrairement à Dropbox, SpiderOak ou HubiC pour ne citer qu'eux, ownCloud permet de faire du Cloud Computing personnel avec son propre serveur. C'est à dire que vos données en ligne ne sont pas soumises à une entité commerciale mais gérées par vous même, sur un disque dur, chez-vous, bien au chaud !

L'inconvénient, c'est que ownCloud nécessite d'avoir un serveur web fonctionnel d'installé, PHP et une base de données (SGBD)!

Il existe deux versions de ownCloud, une version gratuite et une version payante. Nous traiterons dans ce tutoriel de la version gratuite.

Owncloud: tour d'horizon

Si Owncloud est conçu au départ pour fournir un équivalent à Dropbox, il peut aussi faire beaucoup d'autres choses.

En effet, Owncloud est modulaire; c'est à dire que vous pouvez rajouter à votre serveur des fonctionnalités qui ne sont pas présentes par défaut en installant des modules.

Ainsi vous pouvez transformer votre serveur en lecteur vidéo/musique/image à distance, vous pourrez gérer vos contacts, maintenir votre agenda, envoyez des mails etc...

Libre à vous d'ajouter de nouveaux modules présents sur le site [Owncloud Apps](#); ou, si rien ne vous

convient, de créer vos propres applications.

Les possibilités sont limitées par votre imagination (et par vos talents en PHP 😊).

Utilisation

Toutes les informations utiles concernant l'utilisation d'ownCloud.

- [Owncloud - Utilisation](#)

Installation de ownCloud

Owncloud n'est plus disponible dans les dépôts officiels Debian pour Wheezy mais présent pour Jessie, Sid et Experimental.

Installation de ownCloud en ajoutant un dépôt



Dans ce tuto, volontairement je ne vais pas utiliser les opérateurs “&&” et “;” dans le but d'expliquer la marche à suivre en détails, étape par étape!

Comme énoncé précédemment, pour fonctionner, ownCloud a besoin d'un serveur LAMP (Linux+Apache+MySQL+PHP).

Installez les paquets suivants :

Avec apt-get :

```
apt-get update && apt-get install apache2 libapache2-mod-php5 php5 mysql-server mysql-client php5-mysql git
```

Si vous souhaitez de plus ample information sur [un serveur LAMP](#) référez-vous dans le wiki.



Il n'est pas recommandé d'utiliser des dépôts externes au projet Debian. La méthode suivante est donc à utiliser à vos risques et périls.

Pour l'installation de ownCloud (version stable Jessie), nous allons ajouter un nouveau dépôt dans </etc/apt/sources.list.d/> :

```
echo 'deb
http://download.opensuse.org/repositories/isv:/ownCloud:/community/Debian_8.
0/ /' >> /etc/apt/sources.list.d/owncloud.list
```



Ce dépôt installera la version 8.x de owncloud

Cette commande va créer un nouveau fichier "owncloud.list" avec comme référence l'adresse URL du dépôt de ownCloud.

Télécharger [avec wget](#) la clé GPG du dépôt :

```
wget  
http://download.opensuse.org/repositories/isv:ownCloud:community/Debian_8.0/  
Release.key
```

Mettre à jour la liste des paquets pour prendre en compte le nouveau dépôt :

```
apt-get update
```

Installer la clé GPG :

```
apt-key add - < Release.key
```

Pour vérifier que la clé est bien intégrée faites :

```
apt-get update
```

Si il n'y a pas de message d'erreur, tout est ok! Sinon essayez à nouveau d'installer la clé.

Vous pouvez utiliser ce dépôt à la place si vous voulez la version 9.x (la dernière version stable de owncloud):

```
echo 'deb  
http://download.owncloud.org/download/repositories/stable/Debian_8.0/ /' >>  
/etc/apt/sources.list.d/owncloud.list
```

Cette commande va créer un nouveau fichier "owncloud.list" avec comme référence l'adresse URL du dépôt de ownCloud.

Placez vous dans /tmp. Télécharger [avec wget](#) la clé GPG du dépôt :

```
wget  
http://download.owncloud.org/download/repositories/stable/Debian_8.0/Release  
.key
```

Installer la clé GPG :

```
apt-key add - < Release.key
```

Pour vérifier que la clé est bien intégrée faites :

```
apt-get update
```

Si il n'y a pas de message d'erreur, tout est ok! Sinon essayez à nouveau d'installer la clé.

Vérifiez la version du paquet owncloud avec:

apt-cache policy owncloud

```
owncloud:
  Installed: (none)
  Candidate: 9.1.0-1.1
  Version table:
     9.1.0-1.1 0
               500
http://download.owncloud.org/download/repositories/stable/Debian_8.0/
Packages
  7.0.4+dfsg-4~deb8u4 0
                    500 http://mirrors.online.net/debian/ jessie/main amd64 Packages
  7.0.4+dfsg-4~deb8u3 0
                    500 http://security.debian.org/ jessie/updates/main amd64 Packages
```



Il vous faut installer un serveur de base de donnée avant l'installation de owncloud
voir la doc [ici](#)

Enfin, installez ownCloud :

```
apt-get install owncloud
```

Configuration du cache

Il y a plusieurs façon de configurer me cache pour que owncloud soit plus fluide notamment sur le chargement des images. Celle décrite ici a été testé, elle est basée sur [la doc officielle](#) de owncloud 9.xx

Controlez la version du module php5 pour le serveur redis:

```
apt-cache policy php5-redis
```

```
php5-redis:
  Installed: 2.2.5-1
  Candidate: 2.2.5-1
  Version table:
*** 2.2.5-1 0
        500 http://toto.net/debian/ jessie/main amd64 Packages
        100 /var/lib/dpkg/status
```

Le module doit être au moins en 2.2.5.x

Si c'est le cas vous pouvez installer le serveur redis:

```
apt-get install redis-server
```

et le module php qui va bien:

```
apt-get install php5-redis
```

Il vous faut un autre module php:

```
apt-get install php5-apcu
```

Vous pouvez controlez si le serveur redis tourne avec:

```
ps ax | grep redis
```

```
22203 ? Ssl      0:00 /usr/bin/redis-server 127.0.0.1:6379
```

Il faut redémarrer le serveur http ici j'utilise apache2.4.xx

```
service apache2 restart
```

Il faut maintenant éditer le fichier `/var/www/owncloud/config/config.php`

```
nano /var/www/owncloud/config/config.php
```

Y ajouter:

</var/www/owncloud/config/config.php>

```
'memcache.local' => '\OC\Memcache\APCu',  
'memcache.locking' => '\OC\Memcache\Redis',  
'redis' => array(  
    'host' => 'localhost',  
    'port' => 6379,  
),
```

Connectez vous en admin sur votre owncloud via votre navigateur internet, allez dans le panel d'administration normalement vous ne devriez plus avoir le message:

Aucun cache de la mémoire n'est configuré. Si possible, configurez un "memcache" pour augmenter les performances. Pour plus d'information consultez la documentation.

Si le message y est encore un redémarrage de apache devrait régler le problème:

```
service apache2 restart
```

Configuration du fichier de config (`/var/www/owncloud/config/config.php`)

Pour éviter d'avoir une periode de rétention trop importante dans le module activity d'owncloud il faut ajouter:

</var/www/owncloud/config/config.php>

```
'activity_expire_days' => 150,
```

Ici la rétention des activités sont sur 150 jours; votre bas de donnée va vous dire merci ;)

Un coup de F5 après ça pour recharger la page du panel d'administration et le message ne devrait plus y être.

Installation de ownCloud en passant par git

Se rendre dans le dossier du serveur apache :

```
cd /var/www
```

Télécharger ownCloud :

```
git clone https://github.com/owncloud/core.git owncloud
```

```
git clone https://github.com/owncloud/apps.git apps
```

```
git clone https://github.com/owncloud/3rdparty.git 3rdparty
```

Création du dossier “/var/www/owncloud/data” :

```
mkdir owncloud/data
```

Se rendre dans le dossier “/var/www/owncloud” :

```
cd /var/www/owncloud
```

Changer l'utilisateur et le groupe des dossiers “data”, “config” et “apps” :

```
chown -R www-data:www-data data/ config/ apps/
```

Installer Owncloud sur un serveur Nginx et le configurer avec un disque dur externe



— [chuugar](#) 01/01/2015 13:59 : Cette section n'est plus à jour et n'est pas utilisable en l'état.

Présentation

Nginx est un concurrent relativement récent (2002) de Apache. Au départ un peu confidentiel, Nginx ne cesse de croître depuis 2006; aujourd'hui ce serveur tient la troisième position des serveurs les plus déployés dans le monde avec 4% d'utilisation.

Face à Apache, Nginx a la réputation d'être plus léger, plus modulaire, mais aussi plus complexe à configurer.

Sur des vieux systèmes, sur des petites configurations ou encore sur des systèmes visant à être très optimisés il peut-être intéressant de le préférer à Apache.

Objectifs

Dans les lignes qui suivent nous montrerons comment déployer le serveur Nginx, puis nous configurons Owncloud de sorte à ce que les fichiers présents sur un disque dur externe soient synchronisés. Nous déploierons ici une base de données légère avec SQLite.

I - Installation

1 - Services

On vérifie au préalable si Apache est installé sur la machine:

```
aptitude show apache2 | grep "État"
```

Si le paquet est installé, on désactivera Apache:

```
update-rc.d -f apache2 remove
```

2 - Paquets nécessaire

Pour commencer nous allons installer Nginx, PHP-FPM, SQLite ainsi que leurs dépendances:

```
apt-get install php5-fpm php-apc curl sqlite nginx
```

3 - Téléchargement des sources

La deuxième étape consiste à récupérer les sources directement depuis le [site de Owncloud](http://owncloud.org). Pour récupérer la version 6.0:

```
wget http://download.owncloud.org/community/owncloud-6.0.0a.tar.bz2
```

La décompression des sources se fera dans le répertoire **/var/www**; il se peut que vous ayez à créer

le répertoire avant cela:

```
mkdir -p /var/www/
```

```
tar xvfC owncloud-6.0.0a.tar.bz2 /var/www/
```

II - Configuration pour nginx

1 - Chown

Il est nécessaire de changer la propriété du répertoire **/var/www/owncloud** où ne venons de décompresser les sources, pour ce faire on utilise la commande [chown](#). Nginx nécessite que ses utilisateurs soient associés au groupe *"www-data"*; vous devez donc ajouter votre nom d'utilisateur à ce groupe. Pour ce faire nous utiliserons la commande:

```
usermod -a -G www-data "VOTRE_LOGIN"
```



N'oubliez pas de remplacer *"VOTRE_LOGIN"* par... Votre login (sans les guillemets) oui. 😊

Puis nous utiliserons chown:

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

2 - Génération des certificats de sécurité

Rendons nous tout d'abord dans le répertoire où nous allons travailler:

```
cd /usr/local/nginx/conf
```

On génère ensuite la clé (privée) du serveur:

```
openssl genrsa -out server.key 2048
```

Puis l'on crée le [CSR](#):

```
openssl req -new -key server.key -out server.csr
```

Enfin, on signe le certificat en utilisant le CSR et la clé que l'on a créée:

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```


3 - Édition du fichier de configuration Nginx

Vous suivez? On y est presque courage! 😊 Il va falloir maintenant configurer Nginx avec Owncloud; pour cela on crée le fichier **/etc/nginx/sites-available/owncloud** contenant les informations suivantes:

[fichier /etc/nginx/sites-available/owncloud](#)

```
# redirect http to https.
server {
    listen 80;
    server_name cloud.example.com;
    rewrite ^ https://$server_name$request_uri? permanent; # enforce
https
    access_log /var/log/nginx/owncloud.access.log;
    error_log /var/log/nginx/owncloud.error.log;
}

# owncloud (ssl/tls)
server {
    listen 443 ssl;
    ssl_certificate /usr/local/nginx/conf/server.crt;
    ssl_certificate_key /usr/local/nginx/conf/server.key;
    server_name cloud.example.com;
    root /var/www/owncloud;
    index index.php;
    client_max_body_size 900M; # set maximum upload size
    fastcgi_buffers 64 4K;
    access_log /var/log/nginx/owncloud.access.log;
    error_log /var/log/nginx/owncloud.error.log;

    rewrite ^/caldav((/|$).*)$ /remote.php/caldav$1 last;
    rewrite ^/carddav((/|$).*)$ /remote.php/carddav$1 last;
    rewrite ^/webdav((/|$).*)$ /remote.php/webdav$1 last;

    index index.php;
    error_page 403 = /core/templates/403.php;
    error_page 404 = /core/templates/404.php;

    location = /robots.txt {
        allow all;
        log_not_found off;
        access_log off;
    }

    location ~
^/(data|config|\.ht|db_structure\.xml|README|AUTHORS|COPYING-
AGPL|COPYING-README) {
        deny all;
    }
}
```

```
}

location / {
    rewrite ^/.well-known/host-meta /public.php?service=host-meta last;
    rewrite ^/.well-known/host-meta.json /public.php?service=host-meta-
json last;
    rewrite ^/.well-known/carddav /remote.php/carddav/ redirect;
    rewrite ^/.well-known/caldav /remote.php/caldav/ redirect;
    rewrite ^/apps/calendar/caldav.php /remote.php/caldav/ last;
    rewrite ^/apps/contacts/carddav.php /remote.php/carddav/ last;
    rewrite ^/apps/([^/]*)(/*.*\.css|php))$
/index.php?app=$1&getfile=$2 last;

    rewrite ^(/core/doc/[^/]+)/$ $1/index.html;

    try_files $uri $uri/ index.php;
}

location ~ ^(?<script_name>.+?\.php)(?<path_info>/.*)?$ {
    try_files $script_name = 404;
    include fastcgi_params;
    fastcgi_param PATH_INFO $path_info;
    fastcgi_param HTTPS on;
    fastcgi_pass unix:/var/run/php5-fpm.sock;
}

location ~* ^.+.(jpg|jpeg|gif|bmp|ico|png|css|js|swf)$ {
    expires 30d;
    # Optional: Don't log access to assets
    access_log off;
}
}
```

On édite le **server_name** du fichier ci-dessus: remplacez le *cloud.example.com* par votre adresse IP local.



Pour connaître son IP local:

```
ifconfig eth0 | grep "inet ad" | cut -f2 -d: | awk '{print $1}'
```



Faites attention à bien modifier 2 fois `server_name`: une fois pour la redirection (en haut du fichier); puis la seconde pour le https (un peu plus bas)

4 - Activation du site

```
ln -s /etc/nginx/sites-available/owncloud /etc/nginx/sites-enabled/owncloud
```

5 - Paramètres PHP

Il faut maintenant définir la **taille maximum** des fichiers que nous allons échanger avec notre Owncloud. 2 variables sont à modifier dans le fichier **/etc/php5/fpm/php.ini** .

1 - **upload_max_size**: taille maximum des fichiers acceptés. Chez moi:

```
upload_max_filesize = 1500M
```

2 - **post_max_size**: la valeur doit être plus grande que upload_max_size. Chez moi:

```
post_max_size = 1700M
```

6 - Redémarrage du serveur

Le plus simple est de redémarrer votre machine; mais vous pouvez aussi exécutez les commandes suivantes:

```
service nginx restart
```

```
service php5-fpm restart
```

III - Préparation du disque dur externe

Notre serveur est désormais configuré, il ne reste plus qu'à lui indiquer que vous souhaitez stocker vos données sur un disque dur externe branché en USB. Pour cela votre disque dur doit être vierge et au format ext4.

1 - Formater le disque dur

On se réfère à la très bonne documentation de Debian-Facile: [mkfs](#) .

2 - Monter le disque dur automatiquement au démarrage

Encore une fois on va sur le wiki du site: [/etc/fstab](#) . 🤪

3 - Changer les droits d'accès au disque dur

Comme précédemment il faut que les utilisateurs du groupe **www-data** puissent avoir accès au disque dur, vous l'aurez compris on re-lance la commande chown:

```
chown -R www-data:www-data /media/"VOTRE_DISQUE_DUR"
```

IV - Owncloud

Si tout c'est bien passé nous devrions maintenant avoir accès à notre site, dans votre navigateur rentrez votre adresse IP:

```
https://mon.ip.local/
```

Une page d'accueil vous demande de définir le mot de passe administrateur. Faites le; puis cliquez sur "Avancé"; là changez le répertoire des données (par défaut **/var/www/owncloud/data**) par le point de montage de votre disque dur: **/media/"VOTRE_DISQUE_DUR"**.

Puis cliquez sur "Terminez l'installation"

SMS via Owncloud

Description

OwnCloud SMS est une application qui envoie vos SMS Android vers votre serveur ownCloud.

Vous pouvez lire directement vos conversations SMS. Les numéros de téléphone sont résolus à partir de votre carnet d'adresses (contacts). Nous vous suggérons d'utiliser un connecteur CardDAV qui redirigera à votre ownCloud.

Les conversations actives seront rafraîchies automatiquement sur votre serveur.

Exigences

- Un navigateur avec JavaScript activé
- L'application ownCloud SMS sur votre appareil Android (via Google Play Store) et l'accès aux données (2G / 3G / 4G / Wi-Fi)
- L'application ownCloud contact activée sur votre serveur ownCloud
- L'application ownCloud

Installation

- Télécharger le fichier [ocsms-1.0.5.zip](#)
- Extraire l'archive de github vers /apps/
- Renommer le dossier par "ocsms"
- Activez l'application dans votre serveur via le lien
https://VOTRE_SERVEUR/index.php/settings/apps?installed

- Téléchargez l'appli mobile sur [Google Play](#) et l'activer sur votre téléphone intelligent

Liens

- L'application Android OwnCloud SMS:
https://play.google.com/store/apps/details?id=fr.unix_experience.owncloud_sms
- Questions sur l'application ownCloud SMS: <https://github.com/nerzhul/ownCloud-SMS-App/issues>
- La documentation de l'application OwnCloud SMS:
<https://github.com/nerzhul/ownCloud-SMS-App/wiki>
- Le code source de l'application web OwnCloud SMS: <https://github.com/nerzhul/ocsms>
- Questions sur l'application web OwnCloud SMS: <https://github.com/nerzhul/ocsms/issues>

Conclusion

Vous disposez désormais d'un cloud fonctionnel, n'oubliez pas cependant de créer des utilisateurs simples, ne disposants pas des droits d'administrateur sur votre Owncloud. Amusez-vous bien. 😊

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:reseau:owncloud>

Last update: **30/05/2023 13:05**

