

Installation de owncloud sur un raspberry pi !

Nous allons voir comment installer owncloud sur le raspberry pi. On utilisera nginx au lieu de apache comme serveur web ¹⁾. Il est préféré ici car moins gourmand mais un peu plus difficile à optimiser que son homologue. Nous verrons aussi comment mettre owncloud en https.

Définition de owncloud



OwnCloud ²⁾ est un logiciel libre offrant une plateforme de services de stockage et d'applications diverses en ligne.

Fonctionnalités :

- Synchronisation de fichiers entre différents ordinateurs
- Stockage sécurisé (chiffrement des fichiers)
- Partage de fichiers entre utilisateurs ou publiquement
- Lecteur de musique en ligne
- Serveur de fichiers WebDAV
- Calendrier (permettant la synchronisation CalDAV)
- Gestionnaire de Contacts (CardDAV)
- Visionneuse de documents en ligne (pdf, open document)
- Galerie d'images, qui permet de visualiser ses photos et de les classer en albums.
- ...

Installation

1. On met à jour la base de packages de raspbian

```
apt-get update  
apt-get upgrade
```

2. On installe les packages nécessaires

```
apt-get install php5 php5-json php5-gd php5-sqlite curl libcurl3 libcurl3-  
dev php5-curl php5-common php-xml-parser sqlite
```



On choisit **sqlite** comme base de donnée. Si on a moins de 10 utilisateurs, c'est largement suffisant.

3. On installe php-apc (optimisation des performances) et php5-fpm (nécessaire à nginx pour renvoyer le résultat de page PHP en HTML)

```
apt-get install php-apc
apt-get install php5-fpm
```

4. On vérifie que nginx est bien lancé par défaut au boot :

Rem : sur une installation debian il faut ajouter le paquet nginx

```
service --status-all
...
[ - ]  apache2
...
[ + ]  nginx
...
```

Si ce n'est pas le cas, on l'ajoute :

```
update-rc.d nginx enable
```

5. On télécharge owncloud

<http://owncloud.org/>

6. On décompresse l'archive dans **/var/www**

```
tar xvf owncloud-5.0.5.tar.bz2
```

7. On change le propriétaire et groupe du répertoire owncloud

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

8. On crée le répertoire qui va recevoir les certificats pour activer owncloud en https

```
mkdir /etc/nginx/certs
```

9. On génère les certificats

```
cd /etc/nginx/certs
openssl genrsa -des3 -out owncloud.key 1024
openssl req -new -key owncloud.key -out owncloud.csr
cp owncloud.key owncloud.key.org
openssl rsa -in owncloud.key.org -out owncloud.key
openssl x509 -req -days 365 -in owncloud.csr -signkey owncloud.key -out
owncloud.crt
```



Attention : Il faut utiliser un mot de passe robuste pour générer le certificat. Le certificat est valable 365 jours. Il faudra en générer un nouveau dans un an



10. On supprime les fichiers initules

```
rm owncloud.csr owncloud.key.org
```

11. On configure nginx pour charger owncloud. On crée le fichier **/etc/nginx/sites-available/owncloud** ³⁾

```
# 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 /etc/nginx/certs/owncloud.crt;
    ssl_certificate_key /etc/nginx/certs/owncloud.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/caldav /remote.php/caldav/ redirect;
    rewrite ^/.well-known/calDav /remote.php/caldav/ redirect;
    rewrite ^/apps/calendar/caldav.php /remote.php/caldav/ last;
    rewrite ^/apps/contacts/caldav.php /remote.php/caldav/ 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;
}
}

```

12. On active le site

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

13. On modifie php pour permettre l'envoi de fichiers de maximum 900M. On modifie **/etc/php5/fpm/php.ini** ⁴⁾

```
upload_max_filesize = 900M
post_max_size = 1000M
```

upload_max_filesize : La taille maximale en octets d'un fichier à charger.

post_max_size : Définit la taille maximale des données reçues par la méthode POST. Pour charger de gros fichiers, cette valeur doit être plus grande que la valeur de upload_max_filesize.

14. On relance nginx et php5-fpm

```
service nginx restart
```

```
service php5-fpm restart
```

Configuration

1. On se connecte sur owncloud. Logiquement nginx va rediriger la requête du port 80 vers le port 443.

```
http://adresse_ip_du_raspberry_pi
```

Rem : dans le fichier de configuration “/etc/nginx/sites-available/owncloud” nous avons indiqué le nom de notre serveur comme : cloud.example.com Lorsque nous utilisons l'adresse IP du serveur, nginx redirige vers l'url <http://cloud.example.com> Pour pouvoir effectuer des tests sans avoir une url valide (serveur DNS), nous pouvons modifier notre fichier /etc/hosts avec

```
sudo vi /etc/hosts
```

```
IP DE NOTRE SERVEUR    cloud.example.com
```

Lors de du premier accès nous avons le message suivant au sujet de la certification du site web.



Nous pouvons passer par “Je comprends les risques” et “Ajouter une exception...” et “Confirmer l'exception de sécurité”

Nous voilà sur l'écran de login :



The image 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: "Nom d'utilisateur" (Username) and "Mot de passe" (Password), with a toggle icon for password visibility. Below these is a dropdown menu labeled "Avancé ▾". Underneath is a section for "Répertoire des données" (Data directory) with a text input field containing "/var/www/owncloud/data". Below that is a section for "Configurer la base de données" (Configure the database) with the text "SQLite sera utilisé." and a text input field containing "localhost". A large blue button labeled "Terminer l'installation" (Finish installation) is at the bottom. At the very bottom, a footer line reads "ownCloud – services web sous votre contrôle".

On entre un login et un mot de passe. Le premier compte créé aura les droits d'administration dans owncloud.

On peut changer l'endroit où sera stocker les données de owncloud.

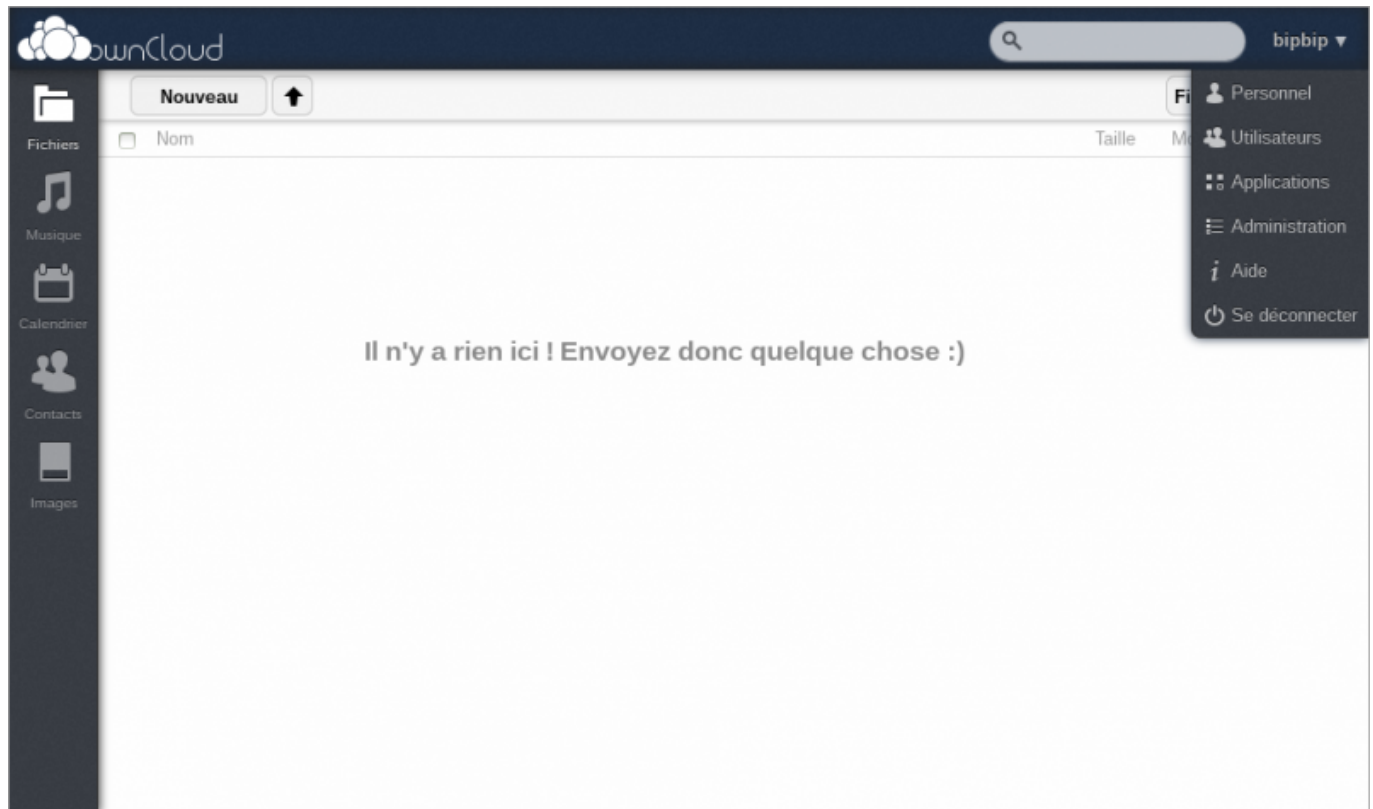
Une fois tous les champs remplis, on clique sur "Terminer l'installation".

Rem : lors du premier démarrage et dans l'onglet d'administration on peut retrouver le message suivante :

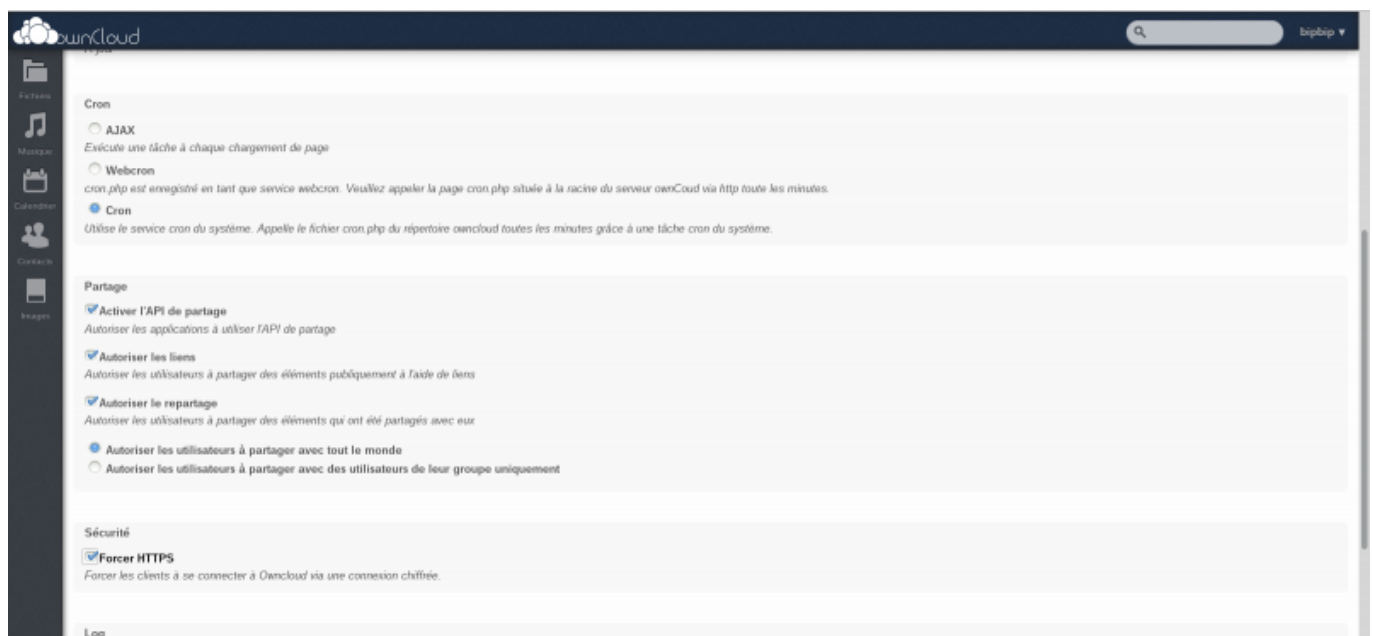
Votre serveur web, n'est pas correctement configuré pour permettre la synchronisation des fichiers, car l'interface WebDav ne fonctionne pas comme il faut.

Cela ne semble pas générer de problème... a vérifier.

2. On sélectionne l'onglet administration.



- On peut choisir la taille maximum pour les fichiers ZIP
- On choisit la façon dont owncloud va rafraichir sa base de donnée ...
- On force le https
- ...

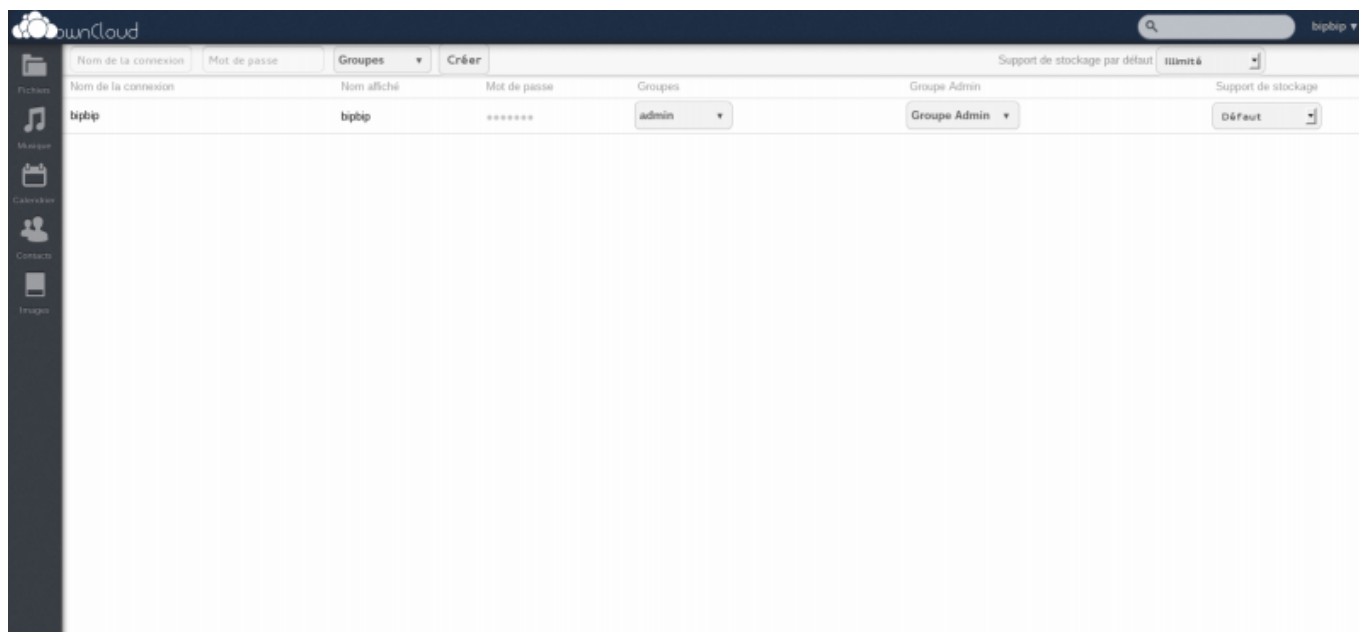


Pour finaliser le rafraichissement de owncloud en ajoutant cette commande en crontab.

```
* /15 * * * * php -f /var/www/owncloud/cron.php >> /var/log/nginx/owncloud.log 2>&1
```

Cela permet de faire du nettoyage dans la base de donnée + d'autres choses toutes les 15 minutes.

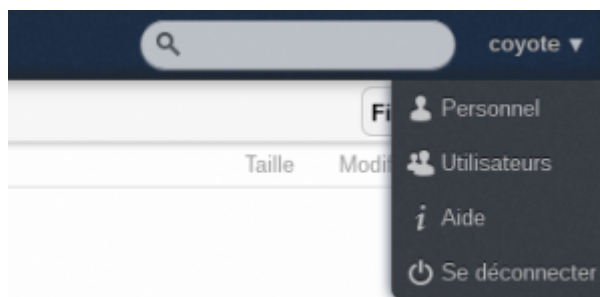
3. On sélectionne l'onglet utilisateur.



On entre le login et le mot de passe des utilisateurs qui auront accès à owncloud.

Lorsque l'on clique sur la flèche dans groupe, on a la possibilité de créer un nouveau groupe "user" par exemple. On l'active en le choisissant dans "Groupe admin". Ce qui va réduire les droits d'accès à l'utilisateur dans le menu (Voir image ci-dessous).

On peut aussi fixer le quota de l'utilisateur.



4. On configure l'envoi d'email par owncloud.

Il faut d'abord installer un client pour envoyer un email. Pour cela on utilise ssmtp. ssmtp permet à des applications d'envoyer des courriels via la commande `/usr/sbin/sendmail`. Il est beaucoup plus facile à mettre en place que sendmail.

```
apt-get install ssmtp
```

On édite le fichier `/etc/ssmtp/ssmtp.conf`

```
#
# Config file for sSMTP sendmail
#
# The person who gets all mail for userids < 1000
# Make this empty to disable rewriting.
root=mon_adresse_gmail@gmail.com
```

```
# The place where the mail goes. The actual machine name is required no
# MX records are consulted. Commonly mailhosts are named mail.domain.com
mailhub=smtp.gmail.com:587

# Where will the mail seem to come from?
rewriteDomain=gmail.com

# The full hostname
hostname=raspberrypi

# Are users allowed to set their own From: address?
# YES - Allow the user to specify their own From: address
# NO - Use the system generated From: address
FromLineOverride=YES

UseTLS=YES
UseSTARTTLS=YES
AuthUser=mon_adresse_gamil@gmail.com
AuthPass=mon_mot_de_passe_gmail
```

Il faut éditer le fichier **/var/www/owncloud/config/config.php** On ajoute les 5 dernières lignes.

```
<?php
$CONFIG = array (
    'instanceid' => '46903df875e5b',
    'passwordsalt' => '74b88d768969f8d0676346770e1fac',
    'datadirectory' => '/var/www/owncloud/data',
    'dbtype' => 'sqlite3',
    'version' => '5.0.6',
    'installed' => true,
    'maxZipInputSize' => 943718400,
    'allowZipDownload' => true,
    'forcessl' => true,
    'mail_smtpmode' => 'smtp',
    'mail_smtphost' => 'ssl://smtp.gmail.com:465',
    'mail_smtpauth' => true,
    'mail_smtpname' => 'mon_adresse_gmail@gmail.com',
    'mail_smtppassword' => 'mon_mot_de_passe_gmail',
);
```

Client synchro linux

Il est possible d'utiliser un client de synchronisation des fichiers en local sous différents OS.

Pour l'installation linux suivre le lien suivant :

<http://software.opensuse.org/download/package?project=isv:ownCloud:devel&package=owncloud-client>

Les fichiers synchronisés seront stockés dans le répertoire "clientsync" sur votre compte owncloud.



Le client ne supporte pas la synchro des fichiers avec le caractère ":" ex : Capture 1:1.png

Utilisation

[Démonstration de owncloud](#)

Références :

[owncloud sur raspberry pi](#)

1)

[apache vs nginx](#)

2)

[Référence wikipédia](#)

3)

[Configuration de nginx](#)

4)

[Description de php.ini](#)

From:

<https://www.loligrub.be/wiki/> - **LoLiGrUB**

Permanent link:

https://www.loligrub.be/wiki/owncloud_raspberry_pi?rev=1368217528

Last update: **2014/12/27 08:13**

