Installation de ownCloud sur un raspberry pi

1/12

Nous allons voir comment installer ownCloud sur le Raspberry Pi. On utilisera Nginx au lieu de Apache comme serveur web¹⁾. 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, la version sécurisée du protocole web http.



Définition de ownCloud

OwnCloud²⁾ 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 (chiffrage 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.
- ...

OwnCloud se positionne en alternative aux solutions de Dropbox, Box.net Google Drive, Ubuntu One, en mettant en avant la flexibilité et la sécurité.



La partie "cloud" de son nom vient du fait que OwnCloud offre une alternative libre aux solutions propriétaires présentes sur le marché. Lors d'une installation en autohébergement, il est nécessaire d'assurer soi-même la sauvegarde et si nécessaire l'archivage des données !

Installation

0. Installer Raspbian pour votre Raspberry. Vous trouverez la procédure à suivre résumée sur cette page !



Pour la suite de l'installation, il faut disposer des droits d'administration. Cela se fait soit en appliquant d'abord la commande "sudo su" pour passer en mode "superutilisateur", soit en faisant précéder de "sudo" toutes les commandes décrites ci-après !.

1. On met à jour la base de packages de Raspbian :

apt-get update apt-get upgrade

2. On installe les packages nécéssaires

apt-get install php5 php5-json php5-gd php5-sqlite curl libcurl3 libcurl3dev 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 (nécéssaire à nginx pour renvoyer le résultat de page PHP en HTML)

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

4. On configure le système pour que le serveur web Nginx soit lancé par défaut au boot. On commence par désactiver apache2 actif par défaut :

update-rc.d -f apache2 remove

(Rem : tester si "update-rc.d apache2 disable" ne suffit pas ??)

On ajoute le paquet nginx :

apt-get install nginx

On active Nginx :

update-rc.d nginx enable

On vérifie que apache2 est bien désactivé et Nginx activé :

3/12

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

5. On télécharge ownCloud (tar or zip file) dans le répertoire /var/www. En ligne de commande, on peut y arriver comme suit :

cd /var/www
wget http://download.owncloud.org/community/owncloud-5.0.5.tar.bz2



le numéro 5.0.5 correspond à la version disponible le 12/05/2013. Il est à actualiser en consultant le site !

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
```



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 <sup>3)</sup>
```

```
# 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/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;
 }
}
```

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 Nx 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 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 voila sur l'écran de login :

ownCloud
Créer un compte administrateur
≗ Nom d'utilisateur
Mot de passe 💿
Avancé * Répertoire des données
/var/www/owncloud/data
Configurer la base de données SQLite sera utilisé.
localhost
Terminer l'installation
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 seront stockées 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
- ...

d Da		bipbip v
	Cron Cron ADAX Exicute une tâche à chaque chargement de page Webcron cron php sat enregistré en tant que service webcron. Veuillez appeler la page cron php située à la racine du serveur ourcCoud via http foute les minutes. Cron Cron Utilise le service cron du système. Appelle le fichier cron php du répertoire ouncloud foutes les minutes grâce à une tâche cron du système.	
breagers	Partage Partage Partage Partage Partage Partage Partage Autoriser les applications à utiliser l'API de partage Partage	
	Sécurité Forcer HTTPS Forcer HTTPS Log	

Pour finaliser le rafraîchissement de ownCloud en ajoutant cette commande dans /etc/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.

(Ô)	un(loud				٩	bipbip 🔻
	Nom de la connexion Mot de passe	Groupes v Créer			Support de stockage par défaut	6 <u> </u>
Fichiers	Nom de la connexion	Nom alliché	Mot de passe	Groupes	Groupe Admin	Support de stockage
Л	bipbip	bipbip		admin *	Groupe Admin *	Défaut 🔄
Musique						
Ë						
Calendrier						
-						
Contacts						
Images						

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 reduire 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</p>
# 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' => '74b88d768969f8d0676346770elfac',
'datadirectory' => '/var/www/owncloud/data',
'dbtype' => 'sqlite3',
'version' => '5.0.6',
'installed' => true,
'maxZipInputSize' => 943718400,
'allowZipDownload' => true,
'forcessl' => true,
'mail_smtpmode' => 'smtp',
'mail_smtpmode' => 'ssl://smtp.gmail.com:465',
'mail_smtphost' => 'ssl://smtp.gmail.com:465',
'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-clie nt

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



Utilisation

Démonstration de owncloud

Problèmes divers

Localisation

Problème : dans la partie administration vous voyez un message sur la localisation : *Localisation non fonctionnelle Ce serveur ownCloud ne peut pas ajuster la localisation du système en en_US.UTF-8/en_US.UTF8. Cela signifie qu'il pourra y avoir des problèmes avec certains caractères dans les noms de fichiers. Il est vivement recommandé d'installer les paquets requis pour le support de en_US.UTF-8/en_US.UTF8.*

Solution : exécuter la commande

```
sudo dpkg-reconfigure locales
```

choisir les locales suivantes :

• en GB.UTF-8

- en_US.UTF-8
- fr_BE.UTF-8
- fr FR.UTF-8

Finalement, indiquer comme défaut pour le système en_US.UTF-8

Une installation automatique de owncloud

Le développeur petRockBlog a développé un script pour effectuer l'installation automatiquement de owncloud et des dépendances.

Vous pouvez trouvez le script ICI : https://github.com/petrockblog/OwncloudPie

Procédure d'utilisation :

- Installer Raspbian sur votre Raspberry Pi (voir la procédure sur cette page)
- Connectez vous en mode console sur votre Raspberry Pi
 - Vous pouvez également vous connecter via le protocole ssh sur votre raspberry pi avec ssh pi@IP_DU_RASPBERRYPI
- Après avoir installé le système via l'image, la partition système sur votre carte SD ne fait que 2go. Si vous avez une carte SD de plus grosse capacité, elle n'est pas exploitée complètement. Voici comme agrandir la partition :

```
sudo raspi-config
-> choisir : expand_rootfs
-> Finish + Reboot now
ou bien : sudo shutdown -r now
```

Expand root partition to fill SD card

• En mode console sur votre Raspberry Pi commencer par la mise à jour des paquets

sudo apt-get update
sudo apt-get upgrade

- Effectuer l'installation de 2 paquets nécessaire pour l'utilisation du script
 - Git = récupération des sources du script
 - Dialog = système de gestion d'écran en mode console

```
sudo apt-get install -y git dialog
```

 Ce placer dans le répertoire home de l'utilisateur et lancer la récupération du script via la commande git

```
cd
```

git clone git://github.com/petrockblog/OwncloudPie.git

- Lancer maintenant le script pour installer owncloud.
 - Ce placer dans le répertoire du script
 - Donner les droits d'exécution à l'utilisateur pour le script
 - Lancer le script ;)

```
cd OwncloudPie
chmod +x owncloudpie_setup.sh
```

sudo ./owncloudpie_setup.sh

- La procédure d'installation commence par un écran de choix
 - Définir l'url de votre installation
 - Choisir le type d'installation

	Bet Aniver DAL New Installati See Installati Bedete exists S Redete Gwicles	(Grt, adropterver, God) on, Nclax based on, Apeche based o Gwecteud testallation dPie script	
l		d mercle.	

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=1368736098



Last update: 2014/12/27 08:13