

Création d'un Cloud privé avec Raspberry Pi

Table of Contents

- [Création d'un Cloud privé avec Raspberry Pi](#)
 - [Préparation Raspberry Pi](#)
 - [ownCloud](#)
 - [Installation de Nginx, PHP7 et du serveur MySQL](#)
 - [Installation de Apache 2, PHP 5 et SQLite](#)
 - [Installation de ownCloud](#)
 - [Configuration de ownCloud](#)
 - [Nextcloud](#)
 - [Installation de nextcloud](#)
 - [Configuration de Nextcloud](#)
 - [CasaOS](#)
 - [Installation de CasaOs](#)
 - [Désinstallation de CasaOS](#)

L'utilisation d'un espace de stockage en ligne via des plateformes de Cloud personnel comme Dropbox, Google Drive ou Amazon Drive devient de plus en plus populaire. Avec ces plateformes, il est possible de stocker les données sur le Cloud et de les consulter en tout temps. En plus du logiciel, la seule chose dont on a besoin sera un accès internet et un ordinateur (ou support mobile). En ce qui concerne le choix de l'hébergeur Cloud, beaucoup de critères sont à prendre en compte. Un des aspects les plus critiqués repose sur le fait que les clients ignorent souvent si des tiers peuvent également accéder à leurs données ou si celles-ci disparaissent définitivement des serveurs une fois qu'elles ont été supprimées par leurs détenteurs. C'est notamment un point important à soulever en ce qui concerne le stockage de données sensibles.

Pour les personnes qui souhaitent avoir un contrôle total sur leurs données, une option consiste à créer et gérer son propre espace sur le Cloud, pour cela des solutions existent.

Ce tutoriel montre comment installer **nextcloudpi**, **ownCloud** ou **CasaOs** pour configurer Raspberry Pi afin d'héberger un Cloud privé.

Préparation Raspberry Pi

Il est important de s'assurer que le Raspberry Pi soit configuré de manière optimale pour l'utilisation avec la solution retenue. Si on n'a pas encore changé le nom d'utilisateur standard « Pi » et le mot de passe par défaut « Rapsberry », il faut procéder à la modification. Il serait risqué pour la sécurité des données de garder l'identifiant et le mot de passe par défaut. Sur Raspbian cela, est possible avec l'utilitaire **raspi-config**:

```
sudo raspi-config
```

Comme avant toute installation sur un Raspberry Pi, il est préférable de mettre à jour l'OS et les piles logicielles installées. Dans Raspbian, saisir les commandes suivantes dans la console :

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

Pour que le Cloud soit accessible, activer la redirection de port du routeur (« Port Forwarding ») et rediriger les requêtes du routeur du port 433 sur le Raspberry Pi. Cette méthode varie de routeur à routeur, mais de nombreux tutoriels sont disponibles en fonction de chaque routeur. Avec une redirection de port, le Raspberry Pi est accessible sur le Web sous son adresse IP (par exemple sous son nom de domaine si on utilise DDNS). Sans redirection de port, on n'aura accès à votre serveur Web que sur le réseau local.



Deskpi Pro est un kit matériel pour convertir une Raspberry PI 4 standard à partir d'un SBC nu, avec un stockage limité, en un mini PC complet avec un bouton d'alimentation, un refroidissement, de meilleurs ports et un Mass Storage via SATA USB3 ou M.2. Pour l'installer sur Raspbian 64bit il faut installer git en premier `apt-get update && apt-get -y install git` puis procéder ainsi:

```
cd ~  
git clone https://github.com/DeskPi-Team/deskpi.git  
cd ~/deskpi/  
./install.sh
```

ownCloud

ownCloud est un logiciel libre, populaire et simple à utiliser. Avec Raspberry Pi il est possible de bénéficier d'un serveur (hôte) pour utiliser **ownCloud**.

Il faut en premier lieu installer un serveur Web pour exécuter **OwnCloud**.

Installation de Nginx, PHP7 et du serveur MySQL

Il faut ajouter de nouvelles sources à APT afin d'installer **PHP 7.1**. Exécuter la commande suivante :

```
sudo apt-get install apt-transport-https lsb-release ca-certificates
```

Il faut maintenant modifier la liste de sources afin d'installer **Php 7** en utilisant APT. Télécharger une clé de sécurité afin de poursuivre l'installation :

```
wget -O /etc/apt/trusted.gpg.d/php.gpg https://packages.sury.org/php/apt.gpg
```

Il faut maintenant créer un fichier `/etc/apt/sources.list.d/php.list` pour que la source **PHP** réside :

```
sudo vi /etc/apt/sources.list.d/php.list
```

Ajouter ce qui suit au fichier :

```
deb https://packages.sury.org/php/ jessie main
```

Enregistrer et quitter, enfin, mettre à jour Apt pour enregistrer les nouvelles sources :

```
sudo apt-get update
```

Il y a quelques paquets dont on aura besoin. Exécuter la commande suivante pour installer les fichiers principaux :

```
sudo apt-get install nginx php7.0-fpm php7.0-curl php7.0-xml php7.0-json  
php7.0-zip php7.0-mb php7.0-mysql php7.0-mcrypt php7.0- dieu
```

Essayer maintenant de visiter l'adresse IP du Raspberry Pi, on doit voir la page par défaut **Nginx** par défaut.

La prochaine étape consiste à installer **MySQL Database Server**. Pour ce faire, exécuter la commande suivante :

```
sudo apt-get install mysql-server mysql-client
```

Une invite interactive commencera à créer la configuration initiale. Fournir un mot de passe fort pour le serveur **MySQL** et continuer jusqu'à la fin de l'installation. Ensuite, il faut se connecter au serveur **MySQL** et créer une base de données et un utilisateur pour l'installation **OwnCloud**.

Se connecter à l'aide de la commande suivante :

```
mysql -uroot -p
```

Il sera demandé le mot de passe root créé lors de l'installation. Une fois connecté au serveur **MySQL** local, exécuter la commande suivante pour créer votre base de données OwnCloud :

```
CREATE DATABASE owncloud;
```

Enfin, créer un nouvel utilisateur (changer le 'a-secure-password' en fonction, il est conseillé de ne pas utiliser le mot de passe root et que de le noter, car on en aura besoin plus tard) :

```
GRANT ALL PRIVILEGES ON owncloud.* TO 'owncloud'@'localhost' IDENTIFIED BY
```

```
'a-secure-password' ;
```

On est prêts à démarrer l'installation d'**OwnCloud**. Quitter en tapant quit.

Tout d'abord, il faut renforcer **PHP** pour garantir sa sécurité. Modifier le fichier de configuration pour PHP à l'aide de cette commande :

```
sudo vi /etc/php/7.0/fpm/php.ini
```

Rechercher ;cgi.fix_pathinfo=1, modifier cette ligne pour qu'elle ressemble à ceci en supprimant le point-virgule et l'espace au début et en changeant la valeur à 0 :

```
cgi.fix_pathinfo=0
```

Modifier le site par défaut à l'aide de la commande suivante :

```
sudo vi /etc/nginx/sites-available/default
```

Modifier le fichier jusqu'à ce qu'il ressemble à ceci:

```
server {
    listen 80
    default_server;
    listen [::]:80
    default_server;
    root /var/www/html;
    index index.php index.html
    index.htm; server_name _;
    location / {
        try_files $uri /index.php$is_args$args;
    }
    location ~ \.php(/|$) {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/run/php/php7.0-fpm.sock;
        fastcgi_read_timeout 300
    }
    location ~ /\.ht {
        deny all;
    }
}
```

Enregistrer et tester la configuration à l'aide de la commande suivante :

```
sudo nginx-t
```

On doit voir la confirmation de la configuration réussie. Tester PHP en créant un nouveau fichier :

```
sudo vi /var/www/html/index.php
```

Entrer ce qui suit dans l'éditeur:

```
<?php phpinfo();
```

Enregistrer puis visiter l'adresse IP du Raspberry Pi cette fois, on doit voir la page d'informations **PHP**

Il faut maintenant étendre le temps d'exécution maximal de **PHP-FPM** à 300 afin de donner à **Nginx** et **PHP** suffisamment de temps pour passer par le processus d'installation. Modifier la configuration **FPM** avec la commande suivante :

```
sudo vi /etc/php/7.0/fpm/pool.d/www.conf
```

Rechercher `request_terminate_timeout` et modifier la ligne se lit comme suit :

```
request_terminate_timeout=300
```

Enregistrer.

Installation de Apache 2, PHP 5 et SQLite

Au départ, le serveur **Apache HTTP Server** sera installé de la manière suivante :

```
sudo apt-get install apache2
```



Pour découvrir si l'installation du serveur Web est réussie, on peut vérifier en saisissant l'adresse IP du Raspberry-Pi dans l'URL d'un navigateur. Si une page Web s'affiche par la suite avec la mention « it works ! », l'installation a bien fonctionné.

Installer **PHP**, **SQLite** et d'autres paquets nécessaires :

```
sudo apt-get install php5 php5-gd sqlite php5-sqlite php5-curl php5-curl
```

Après avoir installé tous les programmes, redémarrer le serveur Web **Apache** :

```
sudo service apache2 restart
```

Installation de ownCloud

Une fois que toutes ces étapes préliminaires ont été mises en place, il est temps d'installer **ownCloud** sur le Raspberry Pi. Le logiciel est disponible par exemple dans la section de téléchargement du site officiel. Télécharger dans le menu « archive file » le dossier .zip et le copier dans le Raspberry Pi. De manière alternative, il est possible d'installer le programme via une console. Pour cela, il convient de saisir la version **ownCloud**. Pour ce tutoriel, **ownCloud 9.1.0** a été utilisé.

```
wget https://download.owncloud.org/community/owncloud-9.1.0.zip
```

Une autre version peut être installée simplement en changeant les chiffres de ladite version ([...] community/owncloud-X.X.X.zip), mais peu importe quelle version on choisi, après avoir téléchargé le dossier .zip **ownCloud**, il convient de le déposer dans le bon dossier et de le décompresser :

```
sudo mv owncloud-9.0.4.zip /var/www/html
cd /var/www/html
sudo unzip -q owncloud-9.0.4.zip
```

Créer le dossier data qui devar être indiqué dans la section “dossier de données” de la page de configuration de **ownCloud**:

```
sudo mkdir /var/www/html/owncloud/data
sudo chown www-data:www-data /var/www/html/owncloud/data
sudo chmod 750 /var/www/html/owncloud/data
```

Après que le dossier ait été créé, redémarrer Raspberry Pi :

```
sudo reboot
```

Configuration de ownCloud

Après que le programme ait été installé et que la bonne structure de dossier a été définie, il convient de mettre en place un compte privé **ownCloud**. Pour cela, saisir l'adresse IP du raspberry-pi dans la barre URL et ajouter « /owncloud » (par exemple 192.168.X.X/owncloud). L'écran de connexion **ownCloud** devrait apparaître.

Entrer le chemin du répertoire data créé précédemment dans la section “dossier de données” de la page de configuration.

Saisir les comptes pour se connecter à la base de données, par exemple pour **MySQL**:

```
Username: owncloud
Password: your chosen password
Database: owncloud
Server: localhost
```

Pour configurer **ownCloud** en utilisant la base de données **SQLite** pour une utilisation limitée, on

peut "Terminer l'installation" sans étapes supplémentaires.



Si jamais un avertissement concernant la performance apparaît concernant **SQLite**, il n'est pas nécessaire de le prendre en compte.

Créer un compte administrateur en saisissant un nom d'utilisateur et un mot de passe.

Enregistrer les données saisies.

Le **ownCloud** personnel est dès à présent disponible sous le compte administrateur. Les données peuvent être importées et téléchargées sous le menu « file ».

Nextcloud

Nextcloud est une solution cloud open source qui peut être considérée comme une alternative aux services commerciaux tels que DropBox, Google Cloud ou aux services cloud de Microsoft (par exemple OneDrive) et d'Apple (iCloud). Le système de cloud a été mis sur le marché en 2016 par Frank Karlitschek, qui avait auparavant quitté le projet précédent **Owncloud** avec un certain nombre d'autres collaborateurs. Les développeurs de **Nextcloud** attachent une importance particulière à ce que le logiciel soit conçu pour une utilisation la plus large possible et ce gratuitement..

NextCloudPi désigne un projet communautaire qui fournit une version **Nextcloud** presque prête à l'emploi pour Raspberry Pi, y compris le programme d'installation. L'instance spéciale **NextCloudPi** est adaptée de manière optimale au matériel partiellement spécifique des nano-ordinateurs, afin de garantir l'interaction la plus fluide possible de tous les composants.

Installation de nextcloud

NextcloudPi peut être installé dans n'importe quelle architecture exécutant la dernière Debian

Pour lancer l'installation, utiliser le script suivant:

```
curl -sSL  
https://raw.githubusercontent.com/nextcloud/nextcloudpi/master/install.sh |  
bash
```

Ou lorsqu'on n'est pas root:

```
curl -sSL  
https://raw.githubusercontent.com/nextcloud/nextcloudpi/master/install.sh |  
sudo bash
```



NextCloudPi permet également de produire des fichiers images prêtes à l'emploi.

- Récupérer les sources git clone
`https://github.com/nextcloud/nextcloudpi.git`
- Se placer dans le répertoire `cd nextcloudpi`



Construction à partir des sources::

- Pour **Raspberry Pi**: `./build/build-SD-rpi.sh`
- Pour **Armbian**: `./build-SD-armbian.sh odroidxu4 # supported board code name`
- Pour **LXD**: `./build/build-LXD.sh`

Configuration de Nextcloud

Nextcloud fournit par défaut les services suivants:

- Raspberry Pi OS/Debian 11 (Bullseye)
- Nextcloud
- Apache, with HTTP2 enabled
- PHP 8.1
- MariaDB
- Redis memory cache
- ncp-config TUI for easy setup (RAM logs, USB drive and more)
- Automatic redirection to HTTPS
- ACPU PHP cache
- PHP Zend OPcache enabled with file cache
- HSTS
- Cron jobs for Nextcloud
- Sane configuration defaults
- Full emoji support
- Postfix email
- Secure

Les extras suivants peuvent être ajoutés:

- Setup wizard
- NextcloudPi Web Panel
- Wi-Fi ready
- Ram logs
- Automatic security updates, activated by default.
- Let's Encrypt for trusted HTTPS certificates.
- Fail2Ban protection against brute force attacks.
- UFW firewall
- Dynamic DNS support for no-ip.org
- Dynamic DNS support for freeDNS
- Dynamic DNS support for duckDNS
- Dynamic DNS support for spDYN
- Dynamic DNS support for Namecheap

- dnsmasq DNS server with DNS cache
- ModSecurity Web Application Firewall
- NFS ready to mount your files over LAN
- SAMBA ready to share your files with Windows/Mac/Linux
- USB automount
- Remote updates
- Automatic NCP updates
- Automatic Nextcloud updates
- Update notifications
- Nextcloud backup and restore
- Nextcloud online installation
- Format USB drive to BTRFS
- BTRFS snapshots
- Automatic BTRFS snapshots
- BTRFS snapshot auto sync
- scheduled rsync
- UPnP automatic port forwarding
- Security audits with Lynis and Debsecan
- ZRAM
- SMART hard drive health monitoring

Les extras peuvent être activés et configurés à l'aide de l'interface Web sur le port HTTPS 4443 , ou depuis la ligne de commande en utilisant:

```
sudo ncp-config
```

CasaOS

CasaOS permet de déployer un cloud à domicile sur lequel on peut gérer toutes les données, appareils et actifs de données très facilement.

- UI conçu pour la maison: Pas de code, pas de formulaires, intuitif, design intuitif
- Prise en charge du système de matériel et de base multiple: **CasaOs** soutient entièrement Zimaboard, Intel Nuc et Raspberry Pi. En outre, **CasaOS** est entièrement compatibles avec Ubuntu, Debian, Raspberry Pi OS et CentOS avec une installation en une ligne de commande
- Installation des applications sélectionnées dans l'App Store en un clic: NextCloud, Homeassistant, Adguard, Jellyfin, * arr et plus!
- Installation facile de nombreuses applications Docker: Plus de 100 000 applications de l'écosystème Docker peuvent être facilement installées
- Gestion élégante du lecteur et des fichiers: Ce qu'on voit est ce qu'on obtient. Aucune formation technique requise.
- Widgets système/application bien conçus: Ce qui tient à cœur, en un coup d'œil. Utilisation des ressources, statut d'application et plus encore!

Installation de CasaOs

Pour installer sur **CasaOs**, exécuter cette commande:

```
wget -qo- https://get.casaos.io | Sudo Bash
```

ou alors

```
curl -fssl https://get.casaos.io | Sudo Bash
```

Désinstallation de CasaOS

Pour la version v0.3.3 ou plus récent

```
Casaos-uninstall
```

Avant la version v0.3.3

```
curl -fsSL https://get.icewhale.io/casaos-uninstall.sh | sudo bash
```

From:

<http://www.ouarte.garden/> - **dwndoc**

Permanent link:

<http://www.ouarte.garden/doku.php?id=prive:rpi-cloud>

Last update: **2025/02/19 10:59**

