

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”



Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Table des matières

Préface.....	3
Historique.....	4
Evolution du tutoriel.....	4
Contexte.....	5
But.....	5
Fonctionnement.....	5
Systèmes et versions supportés.....	5
Prérequis.....	7
Dépendances.....	7
Configuration des dépôts.....	7
Configuration couchdb / cozy-stack / cozy.....	8
CouchDB.....	8
Cozy stack.....	8
Cozy.....	10
Configuration de l'instance Cozy.....	11
D.N.S.....	11
Ouverture des ports 80 et 443.....	13
ACME (Let's Encrypt).....	14
Configuration pour une utilisation en local.....	15
Créer des instances.....	16
Sauvegarde de notre instance COZY.....	17
Migration de COZY v2 vers COZY v3.....	19
Plus loin.....	20

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Préface

Bonjour à tous,

Si vous êtes ici c'est que vous avez l'intention de faire le voyage vers COZY. Cette solution libre (expliqué ici en mode “auto-hébergé”) permettra de stocker vos documents, photos et bien plus

Ma motivation pour créer ce guide m'est venu de la complexité d'une telle mise en place (vocabulaire technique,) et par conséquent de son éventuel abandon pour des personnes non initiées alors que COZY est un projet grandissant et ouvert : libérer vous des solutions propriétaire !!

Je sais par avance que ce tuto ne plaira pas à tout le monde, certain(e)s le trouveront trop expliqué (parties en hors-sujet sur COZY), d'autres au contraire, ne le trouveront pas assez développé. J'ai essayé de trouver le juste milieu en proposant un tuto pour guider les personnes ayant un niveau débutant et en partant d'une installation propre.

Si vous apercevez des erreurs ou des parties à ajouter, n'hésitez pas à les partager, veuillez simplement le préciser dans le chapitre “historique”

Bon courage et comme dirait Dora : Let's Go !!!!

N.B. J'ai effectué mes tests sur une machine virtuelle (V.M.) sous Debian stretch x64 et avec un nom de domaine (hébergeur) pour pouvoir y accéder depuis Internet.

L'équipe Cozy et moi-même ne pourrait être tenu responsable de tous dommages ou perte de données que ce tutoriel pourrait occasionner. Pensez donc à sauvegarder vos données critiques !!

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Historique

Date	Version	Auteur(s)	Description
06/01/2018	1	<i>gudbes</i>	→ Déploiement de COZY en mode “auto-hébergé”
19/02/2018	2	<i>_aeris_ gudbes N`</i>	→ Correction des fautes d'orthographe → Suppression de la partie “Génération des certificats LE” car générer automatiquement lors de la création d'une instance. → Optimisation de la partie “D.N.S.”
21/02/2018	3	<i>gudbes lunanova</i>	→ Correctif au niveau des konectors
06/03/2018	4	<i>Clochix gudbes nono</i>	→ Mise à jour de la partie “Migration V2 vers V3”

Evolution du tutoriel

- Explication sur “un cas” de changement de nom de domaine.
- Mise en place d'un serveur S.M.T.P. pour permettre l'envoi de mail par COZY.
- Procédure de sauvegarde officialisée par COZY ou optimisation de celle proposée.

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Contexte

But

Ce guide doit vous permet de mettre en place un environnement fonctionnel de Cozy V3 utilisant Let's Encrypt (certificat SSL) et nginx consultable en interne (réseau local) et en externe (réseau Internet)

Fonctionnement

Cozy est divisé comme cela :

- `cozy-couchdb`: [CouchDB](#) → moteur de base de données utilisé par Cozy
- `cozy-nsjail`: [NSJail](#) → outil d'isolation utilisé par les connecteurs
- `cozy-stack`: [Cozy core](#) : → coeur de Cozy
- `cozy-coclyco`: [CLI](#) → gestion des serveurs virtuels (vhosts) et des certificats
- `COZY`: regroupement des paquets ci-dessus permettant l'installation “facile” d'un environnement Cozy auto-hébergé

Systemes et versions supportés

Ce référentiel supporte actuellement:

- **Debian Stretch** (9.x): amd64 (conseillé et utilisé pour le reste du tuto)
- **Ubuntu Xenial** (16.04 LTS): amd64
- **Raspbian Stretch** (9.x): armhf

Les dépôts disponibles sont:

- **stable**: versions officielles et supportées (mise à jour tous les 6 mois : pas de prise en charge de fix bug)
 - Debian Stretch (9.x)
 - deb <https://apt.cozy.io/debian/stretch/stable>
 - Ubuntu Xenial (16.04 LTS)
 - deb <https://apt.cozy.io/ubuntu/xenial/stable>
 - Raspbian Stretch (9.x)

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

- deb <https://apt.cozy.io/raspbian/> stretch stable
- **testing**: futures versions officielles, à des fins de test. (mise à jour ± deux fois par mois)
 - Debian Stretch (9.x)
 - deb <https://apt.cozy.io/debian/> stretch testing
 - Ubuntu Xenial (16.04 LTS)
 - deb <https://apt.cozy.io/ubuntu/> xenial testing
 - Raspbian Stretch (9.x)
 - deb <https://apt.cozy.io/raspbian/> stretch testing
- **unstable**: expérimentale, à utiliser avec précaution
 - Debian Stretch (9.x)
 - deb <https://apt.cozy.io/nightly/debian/> stretch unstable
 - Ubuntu Xenial (16.04 LTS)
 - deb <https://apt.cozy.io/nightly/ubuntu/> xenial unstable
 - Raspbian Stretch (9.x)
 - deb <https://apt.cozy.io/nightly/raspbian/> stretch unstable

△ *cozy-couchdb et cozy-nsjail sont des paquets temporaires. Ils seront retirés lorsque couchdb et nsjail seront disponibles en version officielle (paquets validés par Debian).*

△ *Vous pouvez choisir d'installer cozy-couchdb sur le même hôte que cozy-stack ou d'utiliser un serveur CouchDB distant. Cozy n'a besoin que d'un CouchDB 2.x (1.x non supporté).*

△ *Comme pour CouchDB, vous pouvez choisir d'installer votre reverse proxy sur le même hôte ou d'utiliser un serveur distant. Dans ce cas, cozy-coclyco supporte seulement nginx (local). Si vous voulez utiliser apache2 ou un proxy inversé distant, vous devez le configurer manuellement pour l'émission de certificats vhost ou TLS.*

△ *Toutes les commandes indiquées dans ce guide seront à taper avec un utilisateur ayant les droits suffisants.*

△ *Pensez à utiliser Mozilla Firefox plutôt qu'Internet Explorer (petit bug en cours de correction)*

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Prérequis

Dépendances

Voici les dépendances à installer pour satisfaire l'installation de Cozy depuis une installation propre :

```
apt install -y curl apt-transport-https
```

Configuration des dépôts

Adaptez votre fichier de sources (*/etc/apt/sources.list/cozy.list*) en indiquant la version que vous souhaitez utiliser : TESTING (recommandé pour le moment), STABLE ou UNSTABLE (voir partie *systèmes et versions supportés du chapitre CONTEXTE*) et récupérez la clé de signature GPG Cozy

```
# A Adapter au besoin : testing ou stable
echo "deb https://apt.cozy.io/debian/ stretch testing" > \
/etc/apt/sources.list.d/cozy.list

# Récupération de la clé GPG
curl https://apt.cozy.io/cozy.gpg | apt-key --keyring \
/etc/apt/trusted.gpg.d/cozy.gpg add -

# Mise à jour de la liste des paquets
apt update
```

Si vous voulez utiliser les version **unstable/nightly**, vous devez accepter une autre clé (faible et sans mot de passe de notre côté à cause des compilations automatisées non surveillées)

```
# Dépôt pour la version unstable
echo "deb https://apt.cozy.io/debian/ stretch unstable" > \
/etc/apt/sources.list.d/cozy.list

# Récupération de la clé GPG de la version unstable
curl https://apt.cozy.io/nightly/cozy.gpg | \ \
  apt-key --keyring /etc/apt/trusted.gpg.d/cozy.gpg add -

# Mise à jour de la liste des paquets
apt update
```

△ Pensez à faire des mises à jour régulières, la version testing évolue toutes les 2 semaines.

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Configuration couchdb / cozy-stack / cozy

Pour le reste de ce document, nous supposons que vous installez les composants un par un pour permettre une vérification intermédiaire.

Pour un environnement local complet (couchDB + nginx + cozy), il suffit d'installer le paquet cozy qui peut installer tous les paquets nécessaires en un seul clic.

CouchDB

```
# Installation du paquet cozy-couchdb  
apt install -y cozy-couchdb
```

1- Installer CouchDB en mode autonome (**standalone**)

2- Configurer CouchDB pour écouter sur **127.0.0.1**

3 - Choisissez un **mot de passe administrateur**

⚠ Ce mot de passe est utilisé par les scripts shell, évitez donc actuellement d'en utiliser un avec des guillemets simples ou doubles ou d'autres symboles ayant une signification dans un shell . Nous vous conseillons d'en choisir un avec seulement des caractères alphanumériques pour éviter les problèmes.

A ce stade, vous devez avoir une instance CouchDB fonctionnelle

```
# Permet de vérifier le bon fonctionnement de couchdb  
curl http://localhost:5984/  
{  
  "couchdb": "Welcome",  
  "version": "2.1.0",  
  "features": ["scheduler"],  
  "vendor": {"name": "The Apache Software Foundation"}  
}
```

Cozy stack

```
# Installation du paquet cozy-stack  
apt install -y cozy-stack
```

Cozy a besoin de créer un administrateur CouchDB et de se connecter en tant qu'administrateur CouchDB. Remplissez ces paramètres obligatoires pour permettre cette création:

- Adresse (address) : par défaut, c'est **localhost: 5984**
- Nom du nœud (Node Name) : par défaut, c'est **couchdb@localhost**
- Utilisateur Admin (Admin user) : par défaut, c'est **admin**
- Mot de passe administrateur (Admin password): : **mettez celui que vous choisissez lors de l'installation de CouchDB**
- Utilisateur cozy (Cozy user) : par défaut, c'est **cozy**
- Mot de passe cozy (Cozy password): : **choisissez un mot de passe**

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

⚠ Ce mot de passe est utilisé par les scripts shell, évitez donc actuellement d'en utiliser un avec des guillemets simples ou doubles ou d'autres symboles ayant une signification dans un shell . Nous vous conseillons d'en choisir un avec seulement des caractères alphanumériques pour éviter les problèmes.

- Pour la gestion de “stack” (créer des instances, installer des applications...), [Cozy a besoin d'un mot de passe administrateur](#). Alors, choisissez-en un nouveau.

Lorsque vous invoquez cozy-stack (ou cozy-coclyco qui l'utilise), vous devez définir la variable d'environnement COZY_ADMIN_PASSWORD avec ce mot de passe. Vous pouvez le mettre sur votre .bashrc pour plus de simplicité si vous voulez.

```
# A ajouter à votre fichier ~/.bashrc
echo "export COZY_ADMIN_PASSWORD=<votre-mot-de-passe-administrateur>" > \
~/.bashrc
# Prise en compte du fichier .bashrc
. ~/.bashrc
```

⚠ Il semble préférable que la variable COZY_ADMIN_PASSWORD soit ajouté au fichier “.bashrc” d'un utilisateur courant.

A ce stade, Cozy Stack devrait fonctionner

```
# Permet de vérifier le bon fonctionnement de cozy stack
curl http://localhost:8080/version
{"build_mode":"production","build_time":"2017-09-
28T10:26:03Z","runtime_version":"go1.8.1","version":"0.1.0"}#
```

Si, comme moi, vous rencontrez l'erreur : `curl: (7) Failed to connect to localhost port 8080: Connexion refusée`, je vous invite à suivre les indications ci-dessous (merci à [etiennecrb](#) pour cette solution).

```
# Ajout de l'utilisateur COZY et de son mot de passe à la fin du fichier
local.ini
nano /opt/couchdb/etc/local.ini
[admins]
;admin = mysecretpassword
cozy = VotreMotDePasse
# On relance le service CouchDB
systemctl restart couchdb
# On relance le service Cozy-stack
systemctl restart cozy-stack
```

Si vous voulez utiliser des connecteurs, vous devez initialiser le chroot de NodeJS

⚠ Actuellement, ce script ne fonctionne que pour [Debian](#) et sera bientôt adapté pour [Ubuntu](#) et [Raspbian](#)

```
/usr/share/cozy/konnector-create-chroot.sh
```

N.B. Si vous rencontrez l'erreur suivant : `E: No such script: /usr/share/debootstrap/scripts/Debian`. Procédez comme suit :

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

```
# Modification du fichier konnector-create-chroot.sh
nano /usr/share/cozy/konnector-create-chroot.sh
#!/bin/bash
set -xe
CHROOT_DIRECTORY="/usr/share/cozy/chroot"
DEBIAN_INCLUDE_PACKAGES="gnupg,apt-transport-https,ca-certificates,curl"
// Remplacer dans la ligne ci-dessous par lsb_release -si par lsb_release -cs
RELEASE="$(lsb_release -cs)"

mkdir -p "${CHROOT_DIRECTORY}"
debootstrap --variant=minbase --include="${DEBIAN_INCLUDE_PACKAGES}" \
"${RELEASE}" "${CHROOT_DIRECTORY}"

chroot "${CHROOT_DIRECTORY}" /bin/bash <<EOF
echo "deb https://deb.nodesource.com/node_8.x ${RELEASE} main" >
"/etc/apt/sources.list.d/nodesource.list"
curl -s "https://deb.nodesource.com/gpgkey/nodesource.gpg.key" | apt-key
--keyring "/etc/apt/trusted.gpg.d/nodesource.gpg" add -
apt update
apt install -y nodejs
EOF
```

N.B. Pour corriger un problème de synchronisation avec certains connecteurs, procédez comme suit :

```
# Vérification du paramètre système "kernel.unprivileged_usersns_clone"
sysctl -n kernel.unprivileged_usersns_clone

# La valeur retournée doit être 1. Si la valeur est 0, exécutez la commande ci
dessous :
sysctl -w kernel.unprivileged_usersns_clone=1
```

Cozy

```
# Mise à jour et installation du paquet cozy
apt update && apt upgrade -y && apt install -y cozy
```

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Configuration de l'instance Cozy

D.N.S.

Cozy s'appuie sur des sous-domaines pour chaque application que vous installez sur votre instance. Pour une instance `cozy.domaine.fr`, `<app>.cozy.domaine.fr` doit être disponible aussi. Actuellement, vous avez besoin d'au moins :

- `onboarding.cozy.domaine.fr`
- `settings.cozy.domaine.fr`
- `drive.cozy.domaine.fr`
- `photos.cozy.domaine.fr`
- `collect.cozy.domaine.fr`
- `store.cozy.domaine.fr`
- `<app>.cozy.domaine.fr` pour chaque application que vous utilisez

Suivez votre méthode habituelle pour créer ces entrées sur votre zone de domaine. La façon la plus simple de gérer cela est d'utiliser une entrée wildcard* : si elle est supportée par votre hébergeur de domaine.

```
cozy 1h IN A x.x.x.x
*.cozy 1h IN CNAME cozy
```

* Les certificats SSL wildcard - ou omnidomaine - permettent de sécuriser plusieurs sous-domaines d'un même domaine malheureusement mon hébergeur ne propose pas cette solution gratuitement. Je ne la développerais donc pas ici.

Si votre hébergeur ne propose pas l'utilisation d'entrée wildcard, il faudra créer chaque sous domaine à la main. Je vous propose de vous montrer les étapes que j'ai effectué chez mon hébergeur. Pour vous, cela peut différer, mais les étapes restent généralement les mêmes.

⚠ Je prendrais en compte que votre domaine : `cozy.domaine.fr` pointe déjà sur votre adresse I.P. WAN* . Pensez à bien enregistrer les modifications pour chaque étape.

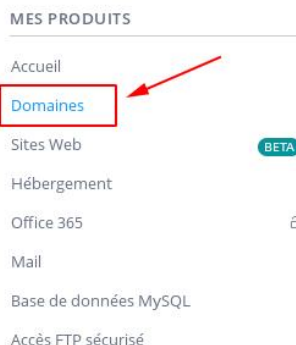
* Vous pouvez obtenir votre adresse I.P. W.A.N. sur ce site par exemple : <http://www.mon-ip.fr/>. Renseignez vous avant pour savoir si elle est fixe ou pas (sinon votre connexion depuis l'extérieur risque d'être temporaire).

1 – Connectez vous, avec vos identifiants, à l'espace client de votre hébergeur.

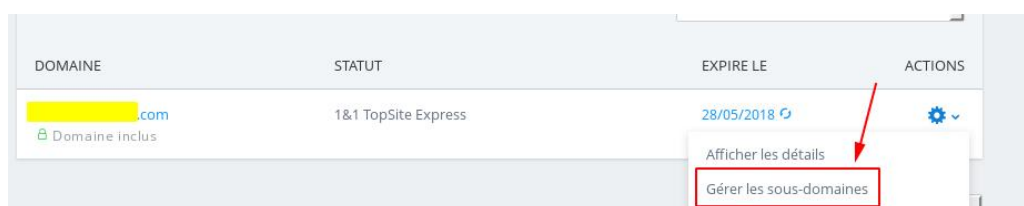
Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

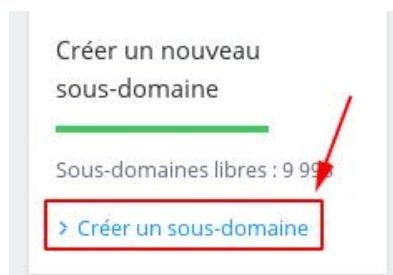
2- Sélectionnez le menu : **Domaines**



3 – Sélectionnez l'option **gérer les sous-domaines** de votre domaine (ex. domaine.fr)



4 – Procédez à la création du nouveau domaine via le menu : **créer un sous domaine**



5 – Créez tous les sous domaines demandées au dessus (ex. onboarding.cozy.domaine.fr).



N.B. Répétez les étapes 4 et 5 pour les autres sous domaine : settings.cozy.domaine.fr, drive.cozy.domaine.fr, photos.cozy.domaine.fr, collect.cozy.domaine.fr, store.cozy.domaine.fr (6 domaines minimum doivent être créés).

6 – Pour chaque sous domaine, **modifier les paramètres DNS**



Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

7 – Indiquez votre adresse I.P. Internet (WAN) * pour la faire pointer vers chaque sous domaine.

Enregistrements A/AAAA et CNAME ?

Adresse IP (A-record) : Adresse IP 1&1
 Autre adresse IP
 CNAME

Alias

Je confirme avoir lu, compris et accepté que tous les paramètres DNS actuels, adresses emails et redirections seront désactivés.

***N.B.** Répétez les étapes 6 et 7 pour les autres sous domaine : settings.cozy.domaine.fr, drive.cozy.domaine.fr, photos.cozy.domaine.fr, collect.cozy.domaine.fr, store.cozy.domaine.fr (6 modifications de DNS minimum doivent être faite).*

Ouverture des ports 80 et 443

J'ai bien conscience que cette partie est complètement hors-sujet mais je me permet quand même de la développer furtivement pour ceux qui en aurait besoin.

Pour fonctionner correctement, nous devons ouvrir, au niveau de notre “box”, 2 ports pointant vers notre serveur COZY :

- **Port 80** (http) : permettant à Let's Encrypt de valider le nom de domaine pour la génération des certificats.
- **Port 443** (https) : permettant l'accès de façon “sécurisée” à notre instance COZY depuis l'extérieur (Internet).

1- Connecter vous à la page d'administration de votre “box” (généralement à l'adresse : <http://192.168.1.1>)

2- Une fois identifié, trouvez un menu intitulé “NAT/PAT” (généralement en mode avancé).

3 – Ajouter aux règles la configuration suivant :

Nom du service	Port interne	Port externe	Protocole	Adresse I.P.
Lets_Encrypt	80	80	TCP	Adresse_IP_Du_Serveur_Cozy *
Cozy_Https	443	433	TCP	Adresse_IP_Du_Serveur_Cozy *

* Vous pouvez trouver cette adresse en exécutant la commande “ip a” dans un terminal.

4 – Valider la configuration et voila notre serveur accessible depuis l'extérieur.

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

ACME (Let's Encrypt)

Comme D.N.S., chaque application utilisera un sous-domaine différent et demandera donc un certificat qui inclut tous les domaines nécessaires.

cozy-coclyco utilise Let's Encrypt et son protocole ACME pour prouver votre propriété sur le domaine où vous essayez d'émettre un certificat. Ce protocole nécessite que votre reverse proxy soit capable de satisfaire les requêtes `http://<app>.domaine.fr/.well-known/acme-challenge/` correctement.

La façon la plus simple d'y parvenir est de configurer votre reverse proxy avec une règle générique pour transférer n'importe quelle requête `/.well-known/acme-challenge/` vers le dossier `/etc/ssl/private/acme-challenge/` correspondant. Pour nginx, ceci peut être fait avec

```
# à mettre dans le fichier /etc/nginx/sites-available/default
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;
    server_name _;

    location /.well-known/acme-challenge/ {
        alias /etc/ssl/private/acme-challenge/;
    }

    location / {
        return 301 https://$host$request_uri;
    }
}
```

Autres cas : pour les certifications autres que par une autorité de certification (ex. Let's Encrypt)

*Si vous utilisez un certificat auto-signé ou une autorité de certification non officielle, vous devez déployer le certificat racine correspondant dans le répertoire :
`/usr/share/cozy/chroot/etc/ssl/certs/custom.crt`.*

Par exemple, si vous utilisez [Let's Encrypt staging environnement](#) à des fins de test:

```
wget -q https://letsencrypt.org/certs/fakelerootx1.pem \  
-O /usr/share/cozy/chroot/etc/ssl/certs/custom.crt
```

Pour l'installation du module SSL, tapez les commandes suivantes :

```
# Installation du paquet ssl-cert
apt install ssl-cert

# Ajout de l'utilisateur www-data au groupe ssl-cert
adduser www-data ssl-cert

# redémarrage du service nginx
systemctl restart nginx
```

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Configuration pour une utilisation en local

Pour pouvoir utiliser correctement notre future instance COZY en local, je préconise de venir compléter le fichier D.N.S. local : /etc/hosts

```
# Ajouter la configuration ci-dessous sur tous vos postes clients locaux (Adresse de mon serveur COZY ici : 10.10.1.200) pour les différents sous-domaines créés
nano /etc/hosts

127.0.0.1 localhost
10.10.1.200 cozy.domaine.fr
10.10.1.200 onboarding.cozy.domaine.fr
10.10.1.200 drive.cozy.domaine.fr
10.10.1.200 settings.cozy.domaine.fr
10.10.1.200 photos.cozy.domaine.fr
10.10.1.200 collect.cozy.domaine.fr
10.10.1.200 store.cozy.domaine.fr
```

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Créer des instances

Une fois que vous avez correctement configurés stack, votre D.N.S. et votre proxy inversé (nginx) , vous pouvez créer des instances (compte utilisateur) sur votre Cozy stack. N'oubliez pas de définir la variable d'environnement COZY_ADMIN_PASSWORD.

```
# Seulement si vous ne l'avez pas ajouter à votre fichier .bashrc
export COZY_ADMIN_PASSWORD=<votre-mot-de-passe-administrateur>

# création de l'instance : indiquer votre sous-domaine : cozy.domaine.fr et
votre adresse mail : me@adressemail.fr
cozy-coclyco create cozy.domaine.fr me@adressemail.fr
```

Pour une référence complète de Coclyco, référez-vous à la documentation de [coclyco](#).

N.B. : Dans mon cas, il a fallu que je créer une seconde fois mon instance pour que cela fonctionne correctement. Si cela devait vous arriver, il vous faudra détruire l'instance précédemment créer pour la régénérer de nouveau :

```
# Suppression de l'instance
cozy-stack instances destroy cozy.domaine.fr

# Régénération de l'instance
cozy-coclyco create cozy.domaine.fr me@adressemail.fr
```

Une fois l'instance créée, il faut activer le lien pour pouvoir passer à la configuration :

- Soit par mail (à l'adresse que vous avez indiqué) si vous avez configuré un serveur S.M.T.P. sur votre serveur.

- Soit en recopiant le lien indiqué à la fin de la commande de création d'instance (ex. <https://cozy.domaine.fr/?registerToken=ce24568ea5b3cbc54ddadb88b61bb32>).

Pour créer une nouvelle instance (nouvel utilisateur), il faut créer un nouveau sous-domaine (ex. cozy2.domaine.fr) et régénérer des certificats avec Let's Encrypt ainsi que la création des différents sous-domaines : photo.cozy2.domaine.fr, onboarding.cozy2.domaine.fr ...

⚠ Attention la creation d'instance fait appel à Let's Encrypt qui impose un quota. Si votre instance n'aboutit pas, **prenez votre mal en patience et retentez votre chance le lendemain.**

Plus loin : Je suis tombé sur ce lien permettant un sécurisation plus poussé de nginx : <https://www.blogdev.fr/tutoriels/nginx/securite-ssl-tls-qualite-a+>

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Sauvegarde de notre instance COZY

⚠ Cette partie n'a pas été vérifiée par mes soins et peut donc contenir des erreurs. Je vous invite donc à remonter toutes informations pour qu'elles puissent profiter à tout le monde.

Je propose ici un petit script (à utiliser avec un utilisateur ayant les droits d'accès aux dossiers à sauvegarder) utilisant le logiciel “BorgBackup” (merci Reventlo) pour permettre de sauvegarder votre instance COZY en attendant une procédure plus officielle. Le script est bien sur à adapter à votre réseau.

```
# Installation du paquet BorgBackup
apt install -y borgbackup
```

```
# Création du script de sauvegarde
touch /etc/cozy/backup_cozy.sh

# Application des droits d'exécution
chmod 755 /etc/cozy/backup_cozy.sh

# Initialisation du dossier de sauvegarde (REPOSITORY) et choisissez un mot de
# passe.
borg init /Chemin/De/La/Sauvegarde/

# Ouvrir le script (backup_cozy.sh) avec votre éditeur préféré (vi ou nano) et
# copier l'encadré ci-dessous dedans en modifiant la variable REPOSITORY en
# conséquence : nano /etc/cozy/backup_cozy.sh
```

```
##### Script de sauvegarde des instances COZY #####
#!/bin/sh
# Indiquez votre mot de passe ici (créer à l'étape précédente) pour que
# borgbackup ne vous le demande plus
export Borg_PASSPHRASE='XYZl0ngandsecurepa_55_phrasea&&123'

#Pour une sauvegarde sur un disque dur
REPOSITORY=/media/USBdisk/backup

#Pour une sauvegarde sur une machine distante
#REPOSITORY=backup@pcdistant:/backup/cozy

# Sauvegarde des dossiers/fichiers utilisés par COZY
borg create -v --stats --progress $REPOSITORY::'{hostname}-{now:%Y-%m-%d}'
/etc/cozy /var/lib/cozy /var/lib/couchdb /etc/hosts /etc/nginx

# Utiliser l'option `prune` pour maintenir (ici) 7 jours, 4 semaines et 6 mois
# d'archive de CETTE machine. Le préfixe '{hostname}-' est très important
# pour limiter l'opération aux archives de cette machines et ne pas l'appliquer
# aux autres archives d'autres machines.
borg prune -v --list $REPOSITORY --prefix '{hostname}-' --keep-daily=7 --keep-
weekly=4 --keep-monthly=6
```

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Une fois terminée, il ne vous reste plus qu'à lancer la sauvegarde manuellement (commande : `./etc/cozy/backup_cozy.sh`) ou à l'intégrer à cron.

Compléments :

```
# Liste toutes les archives de votre repository
borg list /chemin/de/votre/repository
Monday                               Mon, 2016-02-15 19:14:44
Tuesday                              Tue, 2016-02-16 19:15:11

# Liste le contenu de l'archive "Lundi"
borg list /chemin/de/votre/repository::Lundi
drwxr-xr-x user  group          0 Mon, 2016-02-15 18:22:30 home/user/Documents
-rw-r--r-- user  group       7961 Mon, 2016-02-15 18:22:30 home/user/Documents/Import

# Restaure l'archive "lundi" dans le répertoire en cours
borg extract /chemin/de/votre/repository::Lundi
```

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Migration de COZY v2 vers COZY v3

Possédant une version 2 de COZY, je me suis posé la question de la migration de mes données et cela est possible via les commandes suivantes (merci aeris, nono, et Clochix) :

Je vous propose la version qui à fonctionner pour moi mais il existe des variantes plus simple :

```
## Depuis COZY v2
# Déplacement vers le dossier /tmp
cd /tmp
# Mise à jour de "cozy-monitor"
npm install cozy-monitor
# Affichage de la version utilisée de cozy-monitor (la dernière est la 1.4.5)
./node_modules/.bin/cozy-monitor version
# Génération de l'archive de migration
./node_modules/.bin/cozy-monitor export /home/cozy_migration.tar.gz
# Commande pour faire le ménage, une fois l'archive générée
rm -rf /tmp/node_modules
## Depuis COZY v3
# Depuis COZY v3, importation de l'archive précédement créée
cozy-stack instances import cozy_migration.tar.gz --domain cozy.domaine.fr
```

Cozy Cloud

Mise en place de la solution en mode “auto-hébergé”

Plus loin

Pour aller plus loin dans les explications ou pour trouver de l'aide n'hésitez pas à contacter l'équipe COZY par :

- **Le forum** : <https://forum.cozy.io/c/francais>
- **I.R.C.** : <https://webchat.freenode.net/?channels=cozycloud>

Pour toute demande sur ce tutoriel, n'hésitez pas à me contacter : gud*bes*@*proton*mail.com
(retirer les * pour avoir l'adresse valide)

Sources :

- <https://docs.cozy.io/fr/install/debian/>
- <https://www.blogdev.fr/tutoriels/nginx/https-nginx-letsencrypt>
- <https://forum.cozy.io/t/migration-impossible-depuis-une-version-2-5-x/5289/7>
- <https://forum.cozy.io/t/souci-d-installation-sous-debian/5166/6>
- <https://borgbackup.readthedocs.io/en/stable/quickstart.html>
- <https://forum.cozy.io/t/migration-des-donnees-la-v2-a-la-v3/5511>