

# Self-Hosted

Applications et services auto-hébergés

- [Diacamma](#)
  - [Diacamma - Clôture de l'exercice en cours et report à nouveau](#)
  - [Diacamma - mise à jour des modules et sauvegarde des instances](#)
  - [Installation de Diacamma sur serveur Linux Ubuntu/Debian](#)
- [Nextcloud](#)
  - [Installation et mise à jour de Collabora pour Nextcloud](#)
  - [Mise à jour manuelle serveur Nextcloud](#)
- [Installer, configurer et mettre à jour FlexGet](#)

# Diacamma

Diacamma Syndic est un logiciel libre qui permet de gérer simplement une copropriété

# Diacamma - Clôture de l'exercice en cours et report à nouveau

Guide rapide pour clôturer l'exercice en cours, créer un nouvel exercice comptable et effectuer le report à nouveau.

Dans `Comptabilité > Écritures comptables`, contrôler et clôturer toutes les écritures comptables de l'exercice en cours.

Dans `Administration > Configuration comptable`, créer un nouveau plan comptable en important le précédent afin de conserver tous les codes et comptes du plan comptable précédent.

Dans `Comptabilité > Plan comptable` :

1. clôturer l'exercice en cours et choisir comment ventiler le résultat de l'exercice
2. choisir le nouvel exercice créé précédemment et faire le `Report à nouveau`
3. cliquer sur `Commencer` pour démarrer le nouvel exercice

Dans `Administration > Configuration comptable`, activer le nouvel exercice.

Dans `Gestion > Catégories de charges`, reporter et corriger si nécessaire les budgets sur le nouvel exercice.

Le nouvel exercice comptable est prêt à être utilisé et les appels de fonds peuvent être créés

Diacamma

# Diacamma - mise à jour des modules et sauvegarde des instances

Diacamma est fourni avec un script, disponible dans `/var/lucterios2` (`launch_lucterios.sh`), permettant d'effectuer les opérations courantes de maintenance telles que ajouter et modifier des instances, mettre à jour les modules, archiver et restaurer une instance.

## Mise à jour des modules

Vérifier la présence de nouvelles versions des modules

```
cd /var/lucterios2
./launch_lucterios.sh check
```

Exemple de sortie de la commande check

```
check list:

lucterios 2.3.0.18073018 => 2.3.0.18073018
lucterios-standard 2.3.0.18070322 => 2.3.0.18070322
lucterios-documents 2.3.0.18070422 => 2.3.0.18070422
lucterios-contacts 2.3.0.18070609 => 2.3.0.18070609
diacamma-syndic 2.3.0.18073020 => 2.3.0.18073020
diacamma-financial 2.3.0.18072813 => 2.3.0.18072813
diacamma-asso 2.3.0.18070322 => 2.3.0.18070322
```

```
=> No upgrade
```

Lister en détails les modules installés

```
./launch_lucterios.sh installed
```

Mettre à jour tous les modules installés

```
./launch_lucterios.sh update
```

Suite à la mise à jour des modules il faut redémarrer gunicorn pour prendre en compte les nouvelles versions

```
sudo killall gunicorn  
cd ~  
./Diacamma/lancer.sh
```

Script lancer.sh à créer dans /home/username/Diacamma

```
#!/bin/bash  
  
cd /var/lucterios2/  
  
source virtual_for_lucterios/bin/activate  
  
DJANGO_SETTINGS_MODULE='copro.settings' gunicorn lucterios.framework.wsgi --  
bind=127.0.0.1:8000 --daemon --timeout 120  
  
deactivate
```

Rendre le script exécutable

```
chmod +x Diacamma/lancer.sh
```

## Sauvegarde des instances

Lister les instances actives

```
./launch_lucterios.sh listing
```

Archiver (sauvegarder) une instance (copro)

```
./launch_lucterios.sh archive -n copro -f save_copro.tar.gz
```

Restaurer la sauvegarde d'une instance (copro)

```
./launch_lucterios.sh restore -n copro -f save_copro.tar.gz
```

```
./launch_lucterios.sh --help
```

Usage:

```
lucterios_admin.py <check|installed|listing|refreshall|update>
```

```
lucterios_admin.py <add|archive|clear|delete|modif|read|refresh|restore|security> [option]
```

Options:

```
--version          show program's version number and exit
-h, --help         show this help message and exit
-n NAME, --name=NAME Instance name
-p APPLI, --appli=APPLI
                    Instance application
-d DATABASE, --database=DATABASE
                    Database configuration 'sqlite', 'MySQL:...' or
                    'PostgreSQL:...'
-m MODULE, --module=MODULE
                    Modules to add (comma separator)
-e EXTRA, --extra=EXTRA
                    extra parameters (<name>=value,...).For 'security':
                    'MODE=<M>,PASSWORD=<xxx>'(<M> equals to 0,1 or 2)
-f FILENAME, --file=FILENAME
                    file name for restor or archive
-i INSTANCE_PATH, --instance_path=INSTANCE_PATH
                    Directory of instance storage
```

# Installation de Diacamma sur serveur Linux Ubuntu/Debian

## Installation de Diacamma

---

### Prérequis

Avoir un serveur PostgreSQL installé et configuré avec une base de données créée et vide (DB diacamma).

Installer PostgreSQL et les bibliothèques spécifiques suivantes :

```
sudo apt-get install postgresql python-psycopg2 libpq-dev libssl-dev
```

### Créer la base de données

Connexion à l'utilisateur postgres et à l'invite de commande PostgreSQL.

```
sudo -i -u postgres  
sudo psql
```

Sur l'invite de commande de PostgreSQL ("postgres=#" est l'invite de commande, il ne faut pas le retaper), création d'un utilisateur "diacamma" et attribution de droits de création de DB.

```
postgres=# CREATE USER diacamma;  
postgres=# ALTER ROLE diacamma WITH CREATEDB;
```

Création de la base de données "diacamma" associée à l'utilisateur "diacamma".

```
postgres=# CREATE DATABASE diacamma OWNER diacamma;
```

Attribution d'un mot de passe à l'utilisateur pour qu'il puisse se connecter à la DB.  
Bien mettre le mot de passe entre ' '.

```
postgres=# ALTER USER diacamma WITH ENCRYPTED PASSWORD 'mot_de_passe';
```

Quitter l'invite de commande PostgreSQL et vérifier la création de la DB, l'encodage doit être en UTF8.

```
postgres=# \q  
psql -l
```

Se déconnecter de l'utilisateur postgres.

```
exit
```

## Installation via les scripts pour linux

Télécharger les scripts d'installation / désinstallation depuis l'adresse [https://download.sd-libre.fr/Diacamma\\_setup.tar.gz](https://download.sd-libre.fr/Diacamma_setup.tar.gz) et décompresser l'archive.

```
wget http://www.sd-libre.fr/download/Diacamma_setup.tar.gz
tar -xf Diacamma_setup.tar.gz
```

Rendre les scripts exécutables.

```
cd Diacamma
chmod +x install.sh uninstall.sh
```

Lancer l'installation de Diacamma en exécutant le script install.sh

```
sudo ./install.sh
```

Installer le driver PostgreSQL pour Django.

```
cd /var/lucterios2/
source virtual_for_lucterios/bin/activate
pip install psycopg2
deactivate
```

## Créer et modifier une instance

Pour créer une instance "copro", activer les modules nécessaires et configurer la base de données PostgreSQL, utiliser la commande suivante.

Ne pas oublier de renseigner le mot de passe PostgreSQL.

```
./launch_lucterios.sh add -n copro -m  
"lucterios.contacts,lucterios.documents,lucterios.mailing,diacamma.condominium,diacamma.accounting,diacamma.payoff" -p diacamma.syndic -d  
"PostgreSQL:name=diacamma,user=diacamma,password=mot_de_passe,host=localhost"
```

Pour corriger une instance existante, remplacer le paramètre "add" par "modif".

## Configurer Diacamma en mode serveur

Installer l'outil Gunicorn

```
cd /var/lucterios2/  
source virtual_for_lucterios/bin/activate  
pip install gunicorn  
deactivate
```

Ajouter les options ci-dessous dans le fichier de configuration copro/settings.py pour ne pas avoir une erreur d'hôte non autorisé et l'interface en français.

```
ALLOWED_HOSTS = ['192.168.1.50', 'localhost', '127.0.0.1']  
LANGUAGE_CODE = 'fr'
```

Lancer l'instance "copro" en tâche de fond avec les commandes suivantes.

```
cd /var/lucterios2/  
source virtual_for_lucterios/bin/activate  
DJANGO_SETTINGS_MODULE='copro.settings' gunicorn lucterios.framework.wsgi --  
bind=127.0.0.1:8000 --daemon --timeout 120  
deactivate
```

L'instance devrait être accessible à l'adresse locale du serveur sur le port 8000.

<http://192.168.1.50:8000>

# Nextcloud

Nextcloud est un logiciel libre de site d'hébergement de fichiers et une plateforme de collaboration.

# Installation et mise à jour de Collabora pour Nextcloud

## Installation de Collabora (version CODE) pour Nextcloud

---

### Prérequis

Liste des prérequis pour installer Collabora Online Development Edition et l'utiliser avec un serveur Nextcloud :

- Un serveur capable de faire tourner un conteneur Docker
- Un sous-domaine pour le serveur Collabora Online
- Un serveur Apache
- Un certificat SSL pour le sous-domaine Collabora Online
- Un certificat SSL pour le sous-domaine Nextcloud

### Installer le serveur Collabora Online

Étapes pour télécharger et lancer le docker Collabora Online.

```
docker pull collabora/code
docker run -t -d -p 127.0.0.1:9980:9980 -e 'domain=nextcloud\\.choukajohn\\.me' --
dns=192.168.1.50 --restart always --cap-add MKNOD collabora/code
```

Avec `domain=domaine serveur Nextcloud` et `--dns=adresse serveur dns` (si serveur dns local ou modifié sur l'hôte Nextcloud).

## Installer le reverse proxy Apache

Étapes pour installer Apache et activer les modules nécessaires.

```
sudo apt-get install apache2
a2enmod proxy
a2enmod proxy_wstunnel
a2enmod proxy_http
a2enmod ssl
```

Ensuite, configurer un Virtualhost pour rediriger le trafic.

```
sudo nano /etc/apache2/sites-available/office.conf
```

```
<VirtualHost *:443>
ServerName office.nextcloud.com:443

# SSL configuration, you may want to take the easy route instead and use Lets Encrypt!
SSLEngine on
SSLCertificateFile /path/to/signed_certificate
SSLCertificateChainFile /path/to/intermediate_certificate
SSLCertificateKeyFile /path/to/private/key
SSLProtocol          all -SSLv2 -SSLv3
SSLCipherSuite ECDHE-ECDSA-CHACHA20-POLY1305:ECDHE-RSA-CHACHA20-POLY1305:ECDHE-ECDSA-AES128-
GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-GCM-
SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-SHA256:ECDHE-
```

```
RSA-AES128-SHA256:ECDSA-AES128-SHA:ECDSA-AES256-SHA384:ECDSA-AES128-SHA:ECDSA-
ECDSA-AES256-SHA384:ECDSA-AES256-SHA:ECDSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-
AES128-SHA:DHE-RSA-AES256-SHA256:DHE-RSA-AES256-SHA:ECDSA-DES-CBC3-SHA:ECDSA-DES-
CBC3-SHA:EDH-RSA-DES-CBC3-SHA:AES128-GCM-SHA256:AES256-GCM-SHA384:AES128-SHA256:AES256-
SHA256:AES128-SHA:AES256-SHA:DES-CBC3-SHA:!DSS
SSLHonorCipherOrder    on

# Encoded slashes need to be allowed
AllowEncodedSlashes NoDecode

# Container uses a unique non-signed certificate
SSLProxyEngine On
SSLProxyVerify None
SSLProxyCheckPeerCN Off
SSLProxyCheckPeerName Off

# keep the host
ProxyPreserveHost On

# static html, js, images, etc. served from loolwsd
# loleaflet is the client part of LibreOffice Online
ProxyPass                /loleaflet https://127.0.0.1:9980/loleaflet retry=0
ProxyPassReverse         /loleaflet https://127.0.0.1:9980/loleaflet

# WOPI discovery URL
ProxyPass                /hosting/discovery https://127.0.0.1:9980/hosting/discovery retry=0
ProxyPassReverse         /hosting/discovery https://127.0.0.1:9980/hosting/discovery

# Main websocket
ProxyPassMatch "/lool/(.*)/ws$" wss://127.0.0.1:9980/lool/$1/ws nocanon

# Admin Console websocket
ProxyPass    /lool/adminws wss://127.0.0.1:9980/lool/adminws

# Download as, Fullscreen presentation and Image upload operations
ProxyPass                /lool https://127.0.0.1:9980/lool
ProxyPassReverse         /lool https://127.0.0.1:9980/lool
</VirtualHost>
```

Après avoir configuré le Virtualhost :

```
a2ensite office.conf  
sudo service apache2 stop  
sudo service apache2 start
```

## Configurer l'application dans Nextcloud

- Dans la section "Application", "Bureautique & texte", installer Collabora Online.
- Dans les paramètres, section "Collabora en ligne", configurer le sous-domaine du Virtualhost : <https://office.choukajohn.me>

## Mise à jour de Collabora Online

---

Télécharger la dernière image Docker :

```
docker pull collabora/code
```

Lister les images en cours d'exécution :

```
docker ps
```

Stopper et supprimer l'image en cours d'exécution :

```
docker stop NAME
```

```
docker rm NAME
```

Lister espace disque utilisé et nettoyer Docker :

```
docker system df
```

```
docker system prune
```

Démarrer la nouvelle image

```
docker run -t -d -p 127.0.0.1:9980:9980 -e 'domain=nextcloud\\.choukajohn\\.me' --  
dns=192.168.1.50 --restart always --cap-add MKNOD collabora/code
```

Nextcloud

# Mise à jour manuelle serveur Nextcloud

## Procédure de mise à jour manuelle

---

Procédure issue de la documentation officielle de [Nextcloud](#).

Télécharger la dernière version de **Nextcloud** sur <https://nextcloud.com/install>

```
cd /var/www  
sudo su  
wget https://download.nextcloud.com/server/releases/nextcloud-xx.x.x.zip
```

Arrêter le serveur web

```
service apache2 stop
```

Supprimer l'ancien dossier de sauvegarde et renommer le dossier actuel

```
rm -r nextcloud-old  
mv nextcloud nextcloud-old
```

Extraire la dernière version de Nextcloud dans **/var/www/nextcloud**

```
unzip nextcloud-xx.x.x.zip
```

Copier le fichier **config.php** depuis le répertoire **-old** vers le nouveau

```
cp nextcloud-old/config/config.php nextcloud/config/
```

Copier les applications tierces depuis le répertoire **-old** vers le nouveau

```
cp -r nextcloud-old/apps/bruteforcesettings nextcloud/apps  
cp -r nextcloud-old/apps/richdocuments nextcloud/apps
```

Ajuster le propriétaire et les permissions des fichiers

```
chown -R www-data:www-data nextcloud  
find nextcloud/ -type d -exec chmod 750 {} \  
find nextcloud/ -type f -exec chmod 640 {} \  

```

Redémarrer le serveur web

```
service apache2 start
```

Lancer la mise à jour depuis la console avec **occ**

```
cd nextcloud  
sudo -u www-data php occ upgrade
```

Attendre la fin de la mise à jour et se loguer pour vérifier la version.

# Installer, configurer et mettre à jour FlexGet

FlexGet est un outil d'automatisation polyvalent pour tous les médias.

Il permet par exemple de télécharger des séries ou des films automatiquement à partir de fichiers torrent ou de flux RSS de torrent.

---

## Installation de FlexGet sous Linux

### Prérequis

#### Python

Versions de Python supportées :

- 3.10 | 3.11 | 3.12 | 3.13 | 3.14

Pour connaître la version de Python installée et installer une version spécifique :

```
python -V
sudo apt-get install python3.x
```

#### PIP

Le prérequis suivant est l'outil Python package manager dénommé PIP.

Installation :

```
sudo apt-get install python-pip
```

## Installer FlexGet

Commencer par mettre à jour l'outil setuptools puis installer FlexGet avec les commandes suivantes :

```
sudo pip install --upgrade setuptools
sudo pip install flexget
```

## Lancer FlexGet en mode daemon

FlexGet est conçu pour être exécuté depuis le user crontab.

Avec FlexGet en mode daemon, utiliser le plugin scheduler pour définir l'exécution des tâches directement dans le fichier de configuration.

Pour lancer le daemon au démarrage du système, éditer le fichier cron avec la commande :

```
crontab -e
```

Ajouter les lignes suivantes au crontab.

La seconde ligne est utile pour relancer le Daemon toutes les heures en cas de plantage.

```
@reboot /usr/local/bin/flexget daemon start -d  
0 */1 * * * /usr/local/bin/flexget daemon start -d >/dev/null 2>&1
```

## Gestion du daemon FlexGet

Pour lancer le daemon FlexGet utiliser la commande *start*.

Utiliser le paramètre *-d* pour lancer le daemon en arrière plan.

Utiliser le paramètre *--autoreload-config* pour forcer le rechargement du fichier de configuration avant l'exécution de chaque tâche.

```
flexget daemon start [-d, --autoreload-config]
```

Pour stopper le daemon en cours d'exécution :

```
flexget daemon stop
```

Pour vérifier l'état du daemon :

```
flexget daemon status
```

Pour recharger le fichier de configuration :

```
flexget daemon reload-config
```

Pour exécuter immédiatement les tâches du fichier de configuration :

```
flexget execute
```

# Configuration de FlexGet

## Introduction

Afin d'utiliser FlexGet il faudra créer un fichier de configuration.

FlexGet utilise le format [YAML](#) dans le fichier de configuration.

YAML est une façon de définir une suite de clés (des noms de plugin ou des options) et de valeurs (noms de séries ou répertoire de téléchargement).

## Localisation

Par défaut FlexGet essaye de trouver le fichier de configuration *config.yml* depuis plusieurs emplacements, dans l'ordre suivant :

1. Le répertoire courant
2. ~/.flexget/
3. ~/.config/flexget/

Possibilité de créer un lien symbolique dans ~/.flexget/ vers un fichier de configuration sur un emplacement réseau partagé afin de le modifier facilement.

## Config.yml

Exemple de fichier de configuration créé à partir de la documentation sur <https://flexget.com/Configuration>

```
schedules:
  - tasks: '*'
    interval:
      minutes: 10

tasks:
  Téléchargement auto torrents:

    rss:
      https://rss.URL.com/cats=118.119.107.1.2.100.4.5.7.3.8.6.94.95.12.33.31.9.104.13.15.14.110.32.101.109.34.30.98.17.16.20.24.96.99.116.102.103.25.111.27.112.26.28.29.117.21.22.23&type=dl&torrent_pass=xxx
      accept_all: yes

    filesystem:
      path:
        - /home/partage/Telechargement/torrents/
      mask: '*.torrent'

    plugin_priority:
      if: 2
      series: 1

    if:
      - "'1080p' in title":
          set:
            path: /home/partage/Telechargement/Films/
      - "'720p' in title":
          set:
            path: /home/partage/Telechargement/Films/
      - "'x264' in title":
          set:
            path: /home/partage/Telechargement/Films/
      - "'x265' in title":
          set:
```

```
    path: /home/partage/Telechargement/Films/
- "'eBook' in title":
  set:
    path: /home/partage/Telechargement/Magazines Journaux/
- "'.pdf' in title":
  set:
    path: /home/partage/Telechargement/Magazines Journaux/
- "'Epub' in title":
  set:
    path: /home/partage/Telechargement/Magazines Journaux/
- "'mp3' in title":
  set:
    path: /media/disqueUSB/Telechargement/Musique/
- "'MP3' in title":
  set:
    path: /media/disqueUSB/Telechargement/Musique/
- "'Mp3' in title":
  set:
    path: /media/disqueUSB/Telechargement/Musique/
```

series:

```
- Game of Thrones:
  set:
    path: /home/partage/Telechargement/Series TV/Game of Thrones/Saison 7
- Black Mirror:
  set:
    path: /home/partage/Telechargement/Series TV/Black Mirror/Saison 4
- Supernatural:
  set:
    path: /home/partage/Telechargement/Series TV/Supernatural/Saison 13
- The Originals:
  set:
    path: /home/partage/Telechargement/Series TV/The originals/Saison 5
- Doctor Who:
  set:
    path: /home/partage/Telechargement/Series TV/Doctor who/Saison 11
- The Handmaids Tale:
  set:
    path: /home/partage/Telechargement/Series TV/The Handmaids Tale/Saison 1
- Vikings:
```

```

    set:
      path: /home/partage/Telechargement/Series TV/Vikings/Saison 5

transmission:
  host: localhost
  port: 9091
  username: xxx
  password: xxx
  addpaused: no

notify:
  entries:
    title: "{{title}}" #Nouveau torrent en téléchargement
    message: "Téléchargement dans {{path}}"
    what: accepted
    via:
      - pushbullet:
          api_key: xxx
#   task:
#     always_send: no
#     via:
#       - pushbullet:
#           api_key: xxx
  abort:
    via:
      - pushbullet:
          api_key: xxx

```

## Top-Level Keys et Plugins

- `schedules` : top-level key permettant de définir l'intervalle d'exécution des tâches, utilisable en mode daemon uniquement.
- `tasks` : top-level key permettant de créer différentes tâches.
- `rss` : plugin permettant de lire un flux rss en entrée.
- `filesystem` : plugin permettant de lire les fichiers .torrent présents dans un répertoire.
- `plugin_priority` : option permettant de définir l'ordre d'exécution des plugins.
- `if` : plugin permettant de définir des paramètres en fonction d'un mot clé présent dans le titre du torrent.

- series : plugin permettant de définir des paramètres en fonction du titre de la série.
- transmission : plugin permettant de communiquer avec le daemon de transmission.
- notify : plugin permettant d'envoyer des notifications via Pushbullet.

## Vérifier et tester config.yml

Après avoir modifié ou créé le fichier *config.yml* il faut le recharger, vérifier la syntaxe et tester l'exécution avec les commandes suivantes :

```
flexget daemon reload-config  
flexget check  
flexget --test execute
```

## Mise à jour de FlexGet

### Vérifier la version actuelle

Commencer par vérifier la version actuellement installée, la commande donne également le numéro de la dernière version disponible.

```
flexget -V
```

### Mettre à jour avec PIP

Dans l'ordre, stopper le daemon FlexGet, mettre à jour le setuptools et enfin mettre à jour flexget. Redémarrer le daemon FlexGet.

```
flexget daemon stop  
sudo pip install --upgrade setuptools  
sudo pip install --upgrade flexget  
flexget daemon start -d
```

## Vérifier le fichier de configuration

Suite à une mise à jour, vérifier la validité du fichier de configuration, il faudra éventuellement lui apporter des corrections.

```
flexget check
```

Si le fichier de configuration échoue au test, vérifier sur <https://flexget.com/UpgradeActions> si des corrections sont nécessaires.