

Documentation : Installation et configuration d'OpenHalo

Pré-requis

Il faut posséder un environnement Linux. Par exemple, sur Windows il est possible de télécharger WSL en ouvrant PowerShell en administrateur :

```
wsl --install
```

Par défaut, un terminal Linux, Ubuntu, sera installé.

Maintenant, il faut ouvrir un terminal Linux, le mettre à jour et installer les outils nécessaires à l'installation d'OpenHalo :

```
sudo apt-get update
sudo apt install build-essential gcc g++ make cmake autoconf uuid-dev libicu-dev
zlib1g-dev libreadline-dev -y
```

Ainsi un compilateur C et les librairies nécessaires ont été installées.

Il faut également posséder MySQL client. Voici comment l'installer :

```
sudo apt install mysql-client-core-8.0
```

Téléchargement et compilation d'OpenHalo

Désormais, il faut cloner les données d'OpenHalo présentes sur GitHub :

```
git clone https://github.com/HaloTech-Co-Ltd/openHalo.git
```

Les données sont clonées sur l'ordinateur. Il faut maintenant installer depuis le code source, et il faut se placer dans le dossier openHalo :

```
cd openHalo
./configure --prefix=/home/halo/openhalo/1.0 --enable-debug --with-uuid=ossdp --
with-icu CFLAGS=-O2
```

Cette ligne a préparé la compilation et a vérifié les bibliothèques. La compilation et l'installation se font avec :

```
make && make install
```

Pour compiler les modules supplémentaires, il faut se placer dans le dossier contrib :

```
cd contrib  
make && make install
```

Configuration du serveur

Il faut créer le groupe halo avec l'identifiant ID 1000 :

```
groupadd -g 1000 halo
```

Si l'ID est déjà pris, il est possible d'en prendre un autre :

```
groupadd -g 1500 halo
```

Il faut ensuite ajouter l'utilisateur halo au bon ID :

```
useradd -u 1000 -g halo halo
```

Pour vérifier ces créations :

```
id halo est censé donner uid=1000(halo) gid=1000(halo) groups=1000(halo)
```

Il est préférable d'utiliser OpenHalo avec l'utilisateur halo. Pour cela, il faut se placer en tant qu'halo avec :

```
su - halo
```

(Si un mot de passe est demandé, il faut le créer préalablement avec `passwd halo`)

Il faut maintenant préparer les variables d'environnement. Il faut créer un dossier pour les sockets :

```
mkdir /var/run/openhalo  
chown halo:halo /var/run/openhalo
```

Ensuite, il faut définir les variables d'environnement qui permettent au shell de savoir où sont les binaires d'OpenHalo, où se trouvent les données, et comment trouver les librairies. Pour cela, il est préférable de modifier le fichier `~/.bashrc` : Pour l'ouvrir :

```
nano ~/.bashrc
```

Maintenant il faut ajouter les lignes suivantes au fichier :

```
export HALO_HOME=/home/halo/openhalo/1.0
export PGDATA=/home/halo/ohdata
export PATH=$HALO_HOME/bin:$PATH
export LD_LIBRARY_PATH=$HALO_HOME/lib:$LD_LIBRARY_PATH
export PGHOST=/var/run/openhalo
alias pg_ctl='/home/halo/openhalo/1.0/bin/pg_ctl -D /home/halo/ohdata'
```

Pour enregistrer les modifications : Ctrl+O, puis Entrée et quitter avec Ctrl+X Pour appliquer les changements :

```
. ~/.bashrc
```

Il est possible de vérifier :

```
echo $PATH
echo $PGDATA
which pg_ctl
```

Cette dernière ligne doit afficher `/home/halo/openhalo/1.0/bin/pg_ctl`

Initialisation de la base de données

Il faut initialiser la base de données avec cette commande :

```
pg_ctl init -D $PGDATA
```

L'environnement du serveur a bien été configuré, maintenant il est possible de regarder le statut, de lancer, arrêter ou relancer le serveur :

```
pg_ctl status
pg_ctl start
pg_ctl restart
pg_ctl stop
```

Configuration de OpenHalo/PostgreSQL

Pour cela, il faut modifier le fichier postgresql.conf. Pour l'ouvrir :

```
nano $PGDATA/postgresql.conf
```

Dans ce fichier, il faut ajouter, modifier ou enlever le # commentaire de ces lignes :

```
listen_addresses = '*'  
port = 5432  
database_compat_mode = 'mysql'  
mysql.listener_on = true  
mysql.port = 3306
```

Pour enregistrer puis quitter : Ctrl+O, puis Entrée et Ctrl+X

De même, il faut modifier le fichier pg_hba.conf. Pour l'ouvrir :

```
nano $PGDATA/pg_hba.conf
```

Dans ce fichier, il faut ajouter, modifier ou enlever le # commentaire de ces lignes :

```
# IPv4 local connections:  
host      all             all             127.0.0.1/32      trust  
host      all             all             0.0.0.0/0         md5
```

Pour enregistrer puis quitter : Ctrl+O, puis Entrée et Ctrl+X

Maintenant, il est possible de relancer le serveur :

```
pg_ctl restart
```

Pour vérifier si ça tourne :

```
pg_ctl status
```

Activation de l'extension MySQL

Il faut se connecter à PostgreSQL :

```
psql -p 5432
```

Dans cette interface :

```
CREATE EXTENSION aux_mysql CASCADE;
```

Puis il faut créer un utilisateur pour MySQL :

```
SET password_encryption = 'mysql_native_password';  
CREATE USER test PASSWORD 'test';  
SELECT * FROM pg_shadow WHERE username='test';
```

Ensuite, il faut quitter psql :

```
\q
```

Maintenant, il est possible de tester la connexion MySQL :

```
mysql -P 3306 -h 127.0.0.1
```

Le mot de passe est test. Pour quitter mysql :

```
\q
```

Voici des tests pour comprendre comment le lien entre MySQL et PostgreSQL se fait :

```
[jet@halodev-jet-01 oh1]$ mysql -P 3306 -h 127.0.0.1  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 1  
Server version: 5.7.32-log MySQL Server (GPL)  
  
Copyright (c) 2000, 2021, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
mysql> SHOW databases;  
+-----+  
| Database |
```

```

+-----+
| information_schema |
| mys_sys            |
| mysql              |
| performance_schema |
| sys                |
+-----+
5 rows in set (0.01 sec)

mysql> USE mysql;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> CREATE TABLE hello_openhalo(a VARCHAR(30));
Query OK, 0 rows affected (0.01 sec)

mysql> INSERT INTO hello_openhalo VALUES('Hello openHalo!');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM hello_openhalo;
+-----+
| a                |
+-----+
| Hello openHalo!  |
+-----+
1 row in set (0.00 sec)

mysql> quit
Bye
[jet@halodev-jet-01 oh1]$ psql -p 5432
psql (1.0.14.10)
Type "help" for help.

halo0root=# SELECT * FROM mysql.hello_openhalo;
          a
-----
Hello openHalo!
(1 row)

halo0root=#

```

Vérification finale

Pour voir si PostgreSQL écoute bien sur 5432 et MySQL listener sur 3306

```

pg_ctl status
ss -tlnp | grep 5432
ss -tlnp | grep 3306

```

Arrêt et redémarrage du serveur OpenHalo

Quand la session est finie, il faut arrêter PostgreSQL/OpenHalo :

```
pg_ctl stop
```

Puis vérifier l'état et si le serveur n'écoute plus :

```
pg_ctl status  
ss -tlnp | grep 5432  
ss -tlnp | grep 3306
```

Lors d'une prochaine utilisation, il faut tout d'abord ouvrir une session avec l'utilisateur halo :

```
su - halo
```

Recharger les variables d'environnement :

```
././bashrc
```

Lancer le serveur et vérifier l'état :

```
pg_ctl start  
pg_ctl status
```

Une fois le serveur lancé, il est possible d'accéder à PostgreSQL :

```
psql -p 5432
```

Et à MySQL (mot de passe test):

```
mysql -P 3306 -h 127.0.0.1
```

Création d'autres utilisateurs

Il est aussi possible de créer d'autres utilisateurs MySQL compatibles dans PostgreSQL :

```
psql -p 5432  
SET password_encryption = mysql_native_password;  
CREATE USER nom PASSWORD 'mon_mot_de_passe';  
CREATE DATABASE projetdb OWNER nom;
```

```
\q  
mysql -P 3306 -h 127.0.0.1 -unom -p
```

Chargement d'une base de données PostgreSQL dans OpenHalo

Si l'on veut charger une base de données en PostgreSQL (par exemple fichier.sql) :

```
psql -p 5432 -U halo -d postgres -f chemin/vers/backup.sql
```