

http://www.scub-foundation.org/fr/doku.php?id=tutorial_postgresql

Tutorial PostgreSQL

Introduction

PostgreSQL est un [système de gestion de base de données relationnelle et objet](#) open source (en anglais [object-relational database management system](#)).

Site officiel

Vous trouverez le site officiel de PostgreSQL à l'adresse suivante : <http://www.postgresql.org/>

Je vous recommande aussi d'aller faire un tour sur le site de la communauté francophone de PostgreSQL sur : <http://postgresql.fr/>

Wikipédia

Je vous invite à regarder les articles [français](#) ou [anglais](#) de la célèbre encyclopédie Wikipédia.

Les fonctionnalités de PostgreSQL

Le site francophone <http://www.postgresql.fr/> dans la [préface de la documentation](#) de PostgreSQL liste les fonctionnalités de PostgreSQL suivantes :

- requêtes complexes
- clés étrangères
- triggers
- vues
- intégrité transactionnelle
- contrôle des versions concurrentes (MVCC ou multiversion concurrency control)

De plus, PostgreSQL™ peut être étendu par l'utilisateur de multiples façons. En ajoutant, par exemple :

- de nouveaux types de données
- de nouvelles fonctions
- de nouveaux opérateurs
- de nouvelles fonctions d'agrégat
- de nouvelles méthodes d'indexage
- de nouveaux langages de procédure

Tutoriel

Nous allons maintenant présenter comment installer et utiliser PostgreSQL sous Ubuntu et Windows dans deux parties distinctes. Suivant le système d'exploitation que vous utilisez, référez vous à la partie correspondante.

PostgreSQL sous Ubuntu

Je vous conseille de jeter un œil à la [page](#) dédiée à PostgreSQL du site de la communauté francophone des utilisateurs d'Ubuntu.

Téléchargement/Installation

Nous allons détailler ici comment installer PostgreSQL sur votre machine. On utilise le gestionnaire de paquet de Ubuntu qui va faire tout le travail pour nous. Pour cela on ouvre une console et on tape la commande suivante :

```
sudo apt-get install postgresql
```

Il va vous demander de saisir votre mot de passe utilisateur pour continuer l'installation. Puis télécharger les paquets, et procéder à l'installation. Une fois l'installation achevée, on va vérifier qu'on possède bien la dernière version en tapant la commande suivante dans la console :

```
psql -V
```

qui renvoie normalement un résultat du genre :

```
psql (PostgreSQL) x.x  
contient une gestion avancée de la ligne de commande
```

x.x est la version de PostgreSQL installée. Retenez bien la version utilisée, vous en aurez besoin pour la suite de ce tutorial en remplaçant tous les x.x contenus dans les commandes par le numéro de version.

Utilisation

Nous allons maintenant voir les principales commandes pour utiliser PostgreSQL.

Je vous recommande si vous voulez approfondir vos connaissances sur PostgreSQL le tutoriel très détaillé disponible sur le site de la communauté francophone de PostgreSQL sur cette [page](#).

Changement du mot de passe utilisateur postgres

A l'issue de l'installation de PostgreSQL, un utilisateur *postgres* a été créé. La première chose à faire, c'est de définir un nouveau mot de passe pour cet utilisateur. Pour cela on saisit la commande suivante dans une console :

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

On accède alors au prompt de PostgreSQL, on va définir comme mot de passe "scub" pour le super utilisateur *postgres* en tapant les commandes suivantes :

```
ALTER USER postgres WITH ENCRYPTED PASSWORD 'scub';  
\q
```

Connexion à Postgresql

Pour se connecter au SGBD PostgreSQL, il faut dans un premier temps exécuter les commandes en tant que l'utilisateur *postgres*. Pour cela il faut taper la commande suivante dans une console :

```
sudo -s -u postgres
```

La commande *sudo* permet à un utilisateur autorisé d'exécuter une commande en tant que superutilisateur (*root*) ou un autre utilisateur :

- l'option *-s* lance le shell et non juste la commande suivante avec *sudo*
- l'option *-u* spécifie qu'on souhaite utiliser un autre utilisateur que *root*, ici on souhaite utiliser l'utilisateur *postgres*.

On peut ensuite taper la commande suivante pour accéder à PostgreSQL :

```
psql
```

On aura alors accès au shell de postgres qui devrait afficher :

```
Bienvenue dans psql x.x, l'interface interactive de PostgreSQL.
```

```
Saisissez:
```

```
\copyright pour les termes de distribution  
\h pour l'aide-mémoire des commandes SQL  
\? pour l'aide-mémoire des commandes psql  
\g ou point-virgule en fin d'instruction pour exécuter la requête  
\q pour quitter
```

```
postgres=#
```

Création d'un utilisateur

L'utilisateur par défaut *postgres* dans PostgreSQL est l'équivalent du super utilisateur *root* de Linux. Il est recommandé de l'utiliser pour l'administration du système uniquement. C'est pourquoi nous allons créer un nouvel utilisateur.

Tout d'abord, il faut modifier le fichier *pg_hba.conf*. On ouvre le fichier depuis une nouvelle console avec la commande :

```
sudo gedit /etc/postgresql/x.x/main/pg_hba.conf
```

Il faut remplacer *x.x* par la version installée de PostgreSQL.

On va à la fin du document et pour autoriser les connexions avec mots de passe cryptés on remplace *ident sameuser* par *md5*

On devrait avoir un fichier qui ressemble à :

```
# Database administrative login by UNIX sockets
local  all          postgres          md5

# TYPE  DATABASE  USER          CIDR-ADDRESS  METHOD

# "local" is for Unix domain socket connections only
local  all          all            md5
# IPv4 local connections:
host   all          all            127.0.0.1/32  md5
# IPv6 local connections:
host   all          all            ::1/128       md5
```

On sauvegarde le fichier, et on redémarre PostgreSQL à partir de la même console avec la commande :

```
sudo /etc/init.d/postgresql-x.x restart
```

Qui affichera si tout se passe bien :

```
* Restarting PostgreSQL x.x database server
[ OK ]
```

On peut désormais créer notre utilisateur à partir de la console où est logué l'utilisateur *postgres* avec la commande :

```
createuser -PrlSED test_user
```

On aura alors à saisir un mot de passe pour notre nouvel utilisateur, on prendra un mot de passe simple comme “test” pour notre exemple, qu'on retape une deuxième fois, puis on saisit le mot de passe de l'utilisateur *postgres*, “scub”, et on valide en tapant sur la touche *entrer* et on devrait avoir l'affichage suivant :

```
postgres@scub-nprouteau:~$ createuser -PrlSED test_user
Saisissez le mot de passe pour le nouvel rôle :
Saisissez-le à nouveau :
Mot de passe :
CREATE ROLE test_user ENCRYPTED PASSWORD
'md5ce65fb6038e348eb70a17aa9a6ce13e8' NOSUPERUSER NOCREATEDB NOCREATEROLE
INHERIT LOGIN;
postgres@scub-nprouteau:~$
```

Création d'une base de données

Maintenant qu'on a créé un utilisateur, on peut créer une base de données proprement avec la commande :

```
postgres@scub-nprouteau:~$ createdb -O test_user -E UNICODE test_base
```

Se connecter à une base de données en mode console

Pour se connecter avec un utilisateur en mode console, il faut modifier le fichier `/etc/postgresql/X.X/main/pg_hba.conf` pour y ajouter (X.X est à remplacer par le version de PostgreSQL):

```
local all all password
```

Puis relancer le serveur en faisant dans la console :

```
sudo /etc/init.d/postgresql-x.x restart
```

Vous pouvez ensuite vous connecter à la base avec :

```
postgres@scub-nprouteau:~$ psql -d test_base
```

Sauvegarde / Restauration d'une base de données

Pour sauvegarder la base il suffit d'utiliser la commande *pg_dump* avec les options suivantes :

```
pg_dump -h localhost -U test_user -Ft test_base > sauvegarde.tar
```

Pour restaurer une base, on utilisera *pg_restore* :

```
pg_restore -d test_base -U test_user sauvegarde.tar
```

Faire un export de base depuis un serveur distant

Il faut spécifier l'adresse ip du serveur distant avec l'option -h :

```
pg_dump -h 192.168.1.x base -U user > fichier.sql
```

Renommer un utilisateur

A lancer depuis un terminal postgres :

```
alter USER ancien_nom RENAME TO nouveau_nom;
```

Changer le mot de passe d'un utilisateur

A lancer depuis un terminal postgres :

```
alter USER nom_utilisateur ENCRYPTED PASSWORD nouveau_password;
```

Renommer une base existante

A lancer depuis un terminal postgres :

```
alter DATABASE ancien_nom_bdd RENAME TO nouveau_nom_bdd;
```