

Logiciel

# PostgreSQL sur un Raspberry Pi : une base de données SQL

## Introduction

## Pré-requis

- Sur un Raspberry Pi :
  - le Raspberry Pi doit être accessible en local <sup>1)</sup> ou sur internet (<http://monsite.tld>)
  - et un serveur **LLSP** <sup>2)</sup> tourne dessus.

## Installation

### Installation de PostgreSQL

Nous sommes sur le Raspberry Pi via ssh (ou dans une console).

Sur le Raspberry Pi, installez **postgresql** en lançant :

```
$ sudo apt-get update
$ sudo apt-get install postgresql
```

Acceptez d'installer tous les paquets <sup>3)</sup>.

Après un certain temps, il s'affiche :

```
Paramétrage de postgresql-common (13+1wheezy1) ...  
Ajout de l'utilisateur postgres au groupe ssl-cert  
Building PostgreSQL dictionaries from installed myspell/hunspell packages...  
insserv: warning: script 'K01tightvncserver' missing LSB tags and overrides  
insserv: warning: script 'tightvncserver' missing LSB tags and overrides  
Paramétrage de postgresql-9.1 (9.1.12-0wheezy1) ...  
Creating new cluster (configuration: /etc/postgresql/9.1/main, data: /var/lib/postgresql/9.1/main)...  
Moving configuration file /var/lib/postgresql/9.1/main/postgresql.conf to /etc/postgresql/9.1/main...  
Moving configuration file /var/lib/postgresql/9.1/main/pg_hba.conf to /etc/postgresql/9.1/main...  
Moving configuration file /var/lib/postgresql/9.1/main/pg_ident.conf to /etc/postgresql/9.1/main...  
Configuring postgresql.conf to use port 5432...  
update-alternatives: utilisation de « /usr/share/postgresql/9.1/man/man1/postmaster.1.gz » pour fournir « /usr/share/man/man1/postmaster.1.gz » (postmaster.1.gz) en mode automatique  
[ ok ] Starting PostgreSQL 9.1 database server: main.  
Paramétrage de postgresql (9.1+134wheezy4) ...  
pi@framboise ~ $
```

Un utilisateur système postgres a été créé. Il n'a pas de mot de passe : c'est un utilisateur bloqué et le mieux est qu'il le reste.

**utilisateur postgres** Au départ, toutes les opérations d'administration se font donc avec l'utilisateur **postgres** qui seul peut se connecter.

Pour vous connecter en tant qu'utilisateur **postgres**, utilisez la commande :

```
$ sudo -i -u postgres
```

L'invite de commande mentionne que vous êtes actif en tant que postgres :

```
pi@framboise ~ $ sudo -i -u postgres  
postgres@framboise:~$
```



A la fin de cette session d'administration dans PostgreSQL, il suffira de taper

```
$ exit
```

pour reprendre la main en tant qu'utilisateur du système.

On peut aussi lancer une commande (ici **psql**) en tant que postgres :

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

Dès que cette commande se termine, on se retrouve avec l'utilisateur habituel.

Le serveur PostgreSQL est en place.

## Configuration

### Création d'un nouvel utilisateur pour gérer postgresql

Nous allons créer l'utilisateur qui nous permettra de gérer postgresql.

Pour cela, lancez l'outil d'administration de PostgreSQL en tant qu'utilisateur **postgres** :

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

```
pi@framboise ~ $ sudo -i -u postgres
postgres@framboise:~$ psql
psql (9.1.12)
Saisissez « help » pour l'aide.

postgres=#
```

Vous êtes dans une interface en ligne de commande, en tant que **root** (# en bout de ligne).

Pour créer un environnement plus sûr, créez un rôle pour l'administration, avec toutes les autorisations sauf l'héritage des droits :

```
$ CREATE ROLE <admin> LOGIN ENCRYPTED PASSWORD '<mot_de_passe>'
SUPERUSER NOINHERIT CREATEDB CREATEROLE;
```



<admin>  
nom d'utilisateur  
<mot\_de\_passe>  
mot de passe

N'oubliez pas le point-virgule en fin de ligne !

Quittez l'environnement psql :

```
$ \q
```

Vérifiez en essayant de vous connecter avec l'utilisateur que vous venez de créer :

```
$ sudo -i -u postgres
```

### Installation de l'interface web PhpPgAdmin

L'interface web **PhpPgAdmin** pour PostgreSQL simplifie la mise en place et l'administration.

Installez le paquet :

```
$ sudo apt install phppgadmin
```

Éditez avec les droits d'administration le fichier **/etc/lighttpd/lighttpd.conf** pour ajouter la ligne suivante à la fin du fichier :

</etc/lighttpd/lighttpd.conf>

```
...
alias.url += ( "/phpPgAdmin" => "/usr/share/phpPgAdmin/" )
```

Ainsi, l'adresse <http://monsite.tld/phpPgAdmin> pointe sur le bon répertoire interne.

Dans le fichier **/etc/lighttpd/lighttpd.conf**, assurez-vous que le module **mod\_alias** est activé (dé-commenté)

et activez les modules **mod\_fastcgi** et **mod\_cgi** :

```
$ sudo lighty-enable-mod fastcgi cgi
```

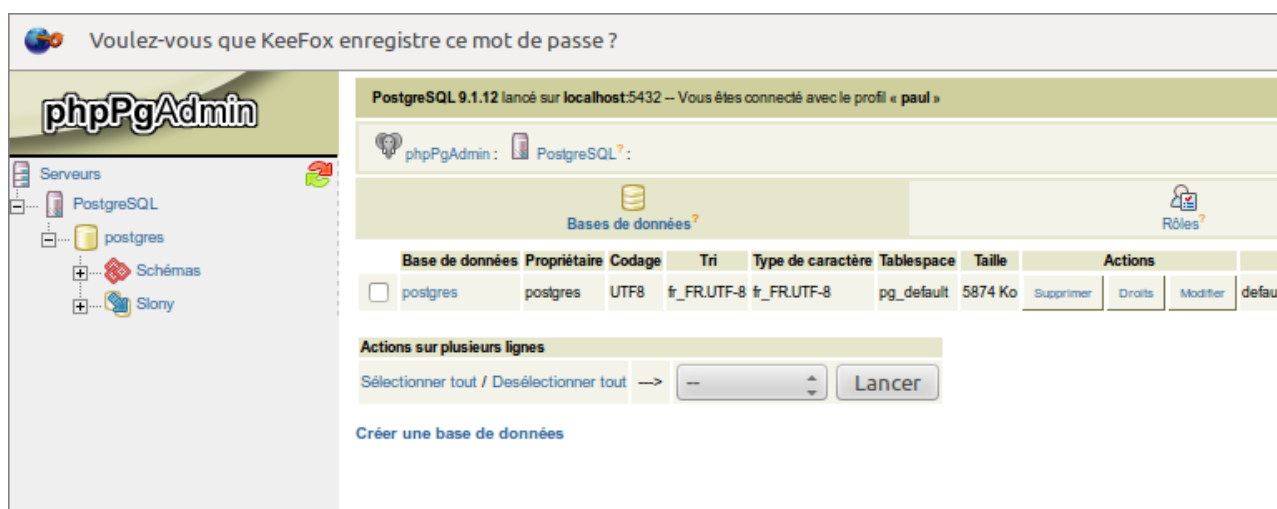
Redémarrez le serveur :

```
$ sudo service lighttpd restart
```

Allez à la page de phpPgAdmin : <http://monsite.tld/phpPgAdmin>

Cliquez sur **PostgreSQL** sur le côté gauche.

Dans la fenêtre de connexion, entrez le nom et le mot de passe de l'utilisateur créé plus haut pour gérer postgresql. Cela devrait vous connecter



Pour ajouter un rôle (utilisateur)

Cliquez sur **rôles** dans la zone du milieu en haut de la page

Cliquez sur **Créer Rôle**

Créer nom d'utilisateur / mot de passe et donner toutes les autorisations autres que **privileges Inherited**.

Ignorer les autres options dans les cases à cocher

Cliquez sur **Créer**

## Sous Apache

Installez le paquet :

```
$ sudo apt-get install phppgadmin
```

Ajouter la directive :

```
# Mes alias  
Alias /phpPgadmin /usr/share/phpPgadmin/
```

## Sécurisation de l'interface web

Nous allons maintenant configurer un utilisateur pour l'interface Web.

Pour cela, il faut d'abord retirer les sécurités. éditez avec les droits d'administration le fichier **/usr/share/phpPgadmin/conf/config.inc.php** pour mettre la ligne suivante à false :

[/usr/share/phpPgadmin/conf/config.inc.php](#)

```
$conf['extra_login_security'] = false;
```

Il faut encore modifier l'utilisateur postgres pour lui donner un mot de passe.

Lancer :

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

Au prompt, exécuter :

```
$ ALTER USER postgres password 'Mot_DE_PASSE';
```

et quitter par

```
$ \q
```

```
pi@framboise ~ $ sudo -u postgres psql
psql (9.1.12)
Saisissez « help » pour l'aide.

postgres=# ALTER ROLE postgres WITH ENCRYPTED PASSWORD 'raspberry'
postgres=# \q
```

### Arborescence

- fichiers de configuration : dans **/etc/postgresql/[VERSION]/main/** :
  - environment
  - postgresql.conf
  - pg\_hba.conf
  - pg\_ident.conf
  - start.conf

2. Les données se trouveront dans **"/var/lib/postgresql/[VERSION]/main"**

Pour déplacer le répertoire des données, éditer la ligne suivante du fichier **postgresql.conf** :

```
data_directory = ...
```



Le répertoire des données doit appartenir à "postgres:postgres" :

```
$ sudo chown postgres:postgres -R repertoiredata
```

Le port par défaut se configure dans le fichier **postgresql.conf**.

### Sécurisation

Maintenant, allez à la page de phpPgAdmin : <http://framboise.local/phppgadmin/>.

- Choisir la langue française.
- Cliquez sur **PostgreSQL** sur le côté gauche pour charger le serveur. Pour la connexion,
  - Utilisateur : **postgres**
  - Mot de passe : mot de passe défini ci-dessus
  - → Cela vous connecte.

Nous allons maintenant créer un nouveau compte d'utilisateur qui peut accéder à l'interface de phpPgAdmin en toute sécurité.



En effet pour l'instant, n'importe qui peut se connecter en utilisant le mot de passe avec l'utilisateur **postgres**.

Pour créer un environnement plus sûr

- Cliquez sur **Rôles** dans la zone du milieu en haut de la page
- Cliquez sur **Créer un rôle**
- Créez un nom d'utilisateur/mot de passe et donnez-lui toutes les autorisations autres que "Hérite des droits". Ignorez les autres options dans les cases à cocher
- Cliquez sur **Créer**
- Cliquez sur **Déconnexion** en haut à droite de la page

Maintenant que vous êtes déconnecté, Essayez de vous connecter avec l'utilisateur que vous venez de créer : Vous devez y arriver.

Il est temps de re-sécuriser notre serveur. éditez avec les droits d'administration le fichier **/usr/share/phpPgAdmin/conf/config.inc.php** pour mettre la ligne suivante à true :

</usr/share/phpPgAdmin/conf/config.inc.php>

```
$conf['extra_login_security'] = true;
```

C'est tout! Vous avez maintenant une page de phpPgAdmin sécurisé et une interface agréable pour configurer votre serveur PostgreSQL.

## Utilisation

### Aide-mémoire de psql


Voir :

- [Aide-mémoire des commandes psql](#)
- [Aide-mémoire des commandes SQL](#)



Petit guide de survie pour le client psql

\h

 \? pour l'aide-mémoire des commandes SQL

\g ou point-virgule en fin d'instruction pour exécuter la requête

\q pour quitter

## Désinstallation

## Voir aussi

- **(en)** <http://c-mobberley.com/wordpress/index.php/2013/10/18/raspberry-pi-installation-of-postgresql-its-simple/>

Basé sur « [Article](#) » par Auteur.

1)

par exemple sur <http://framboise.local> avec **avahi**

2)

lighttpd + php + SQLite

3)

les paquets supplémentaires suivants vont être installés : libpq5, lsb-release, postgresql-9.1, postgresql-client-9.1, postgresql-client-common, postgresql-common,,ssl-cert

From:  
<https://nfrappe.fr/doc-0/> - **Documentation du Dr Nicolas Frappé**

Permanent link:  
<https://nfrappe.fr/doc-0/doku.php?id=logiciel:sql:postgresql:rpi:start>

Last update: **2022/08/13 22:14**

