
POSTGRE

Installation et configuration du serveur PostgreSQL

Installation Postgre

Pour pouvoir installer et configurer notre serveur postgresQL, il faut dans un premier temps installer PostgreSql. Pour cela nous avons effectué la commande suivante :

```
root@constance:/home/constance# usermod -g sudo constance
root@constance:/home/constance# exit
constance@constance:~$ sudo apt-get install postgresql
```

Par la suite il nous faut créer un utilisateur, une base de données et également mettre en place l'outil pgAdmin4 qui nous permettra d'administrer nos bases de données.

De cette façon pour créer un utilisateur, nous avons effectué la commande suivante : `createuser appli_web --pwprompt --interactive` qui nous permet d'avoir une installation interactive lors de la création de l'utilisateur

```
postgres@constance:~$ createuser appli_web --pwprompt --interactive
Saisir le mot de passe pour le nouveau rôle :
Le saisir de nouveau :
Le nouveau rôle est-il super-utilisateur ? (o/n) n
Le nouveau rôle est-il autorisé à créer des bases de données ? (o/n) o
Le nouveau rôle est-il autorisé à créer de nouveaux rôles ? (o/n) o
```

L'utilisateur étant créé, nous pouvons ensuite créer notre base données.

```
postgres@constance:~$ createdb -O appli_web appli_web
```

Nous pouvons nous connecter sur cette dernière avec l'user créé précédemment. Il suffit donc d'effectuer la commande suivante:

```
psql -U nom_utilisateur -d nom_de_la_base
```

```
root@constance:/# psql -U appli_web appli_web
Mot de passe pour l'utilisateur appli_web :
psql (9.6.6)
Saisissez « help » pour l'aide.

appli_web=> help
Vous utilisez psql, l'interface en ligne de commande 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
```

Installation outil pgAdmin4

Il nous faut maintenant installé l'outil pgAdmin4, pour pouvoir administrer nos bases de données. Pour cela nous avons tout d'abord du installé plusieurs packages nécessaire au fonctionnement de pgadmin4, nous avons donc effectuer la commande suivante:

```
constance@constance:~$ sudo apt install build-essential libssl-dev libffi-dev libbz2-dev virtualenv python-pip libpq-dev python-dev
```

Une fois ces package installé, on crée l'environnement nécessaire au bon fonctionnement de pgadmin4:

```
constance@constance:~$ cd /opt/
constance@constance:/opt$ mkdir enviromentpy
mkdir: impossible de créer le répertoire « enviromentpy »: Permission non accordée
constance@constance:/opt$ sudo mkdir enviromentpy
constance@constance:/opt$ virtualenv pgadmin4
```

Une fois l'environnement configurer, il nous suffit de récupérer le package de pgadmin4 via l'adresse suivante (il suffit juste de changer la version voulu de pgadmin4 pour avoir la plus récente, ici l'exemple montre 1.3 mais nous avons pris la version la plus récente a savoir la 2.1):

```
constance@constance:/opt/pgadmin4$ source bin/activate
(pgadmin4) constance@constance:/opt/pgadmin4$ wget https://ftp.postgresql.org/pub/pgadmin/pgadmin4/c1.3/pip/pgadmin4-1.3-py2.py3-none-any.whl
```

Ensuite on instal pgadmin4 grâce a la commande suivante:

```
$pip install pgadmin4-1.3-py2.py3-none-any.whl
```

Une fois l'installation terminée, il faut créer le fichier de configuration locale de pg admin. On réalise ce ci grâce aux commande suivante:

-dans un premier temps on se déplace au bon endroit ou l'on souhaite que le fichier de configuration soit crée:

```
$cd lib/python2.7/site-packages/pgadmin4
```

-Puis on créer et on modifie le fichier de configuration:

```
$touch config_local.py
```

Une fois le fichier créé, on ajoute la ligne suivante dedans:

```
# echo « SERVER_MODE = False » >> lib/python2.7/site-packages/pgadmin4/config-local.py
```

On lance ensuite pgadmin4,

```
$python setup.py
```

Une fois lancé on nous demande de renseigner une adresse mail ainsi qu'un mot de passe.

Une fois le compte créer dans notre navigateur on ouvre notre localhost:5050 et on se connecte a notre PostgreSQL Server :

```
$python pgAdmin4.py
```

```
(.pgadmin4) root@constance:~/.pgadmin4/lib/python2.7/site-packages/pgadmin4# python setup.py
pgAdmin 4 - Application Initialisation
=====

The configuration database - '/root/.pgadmin/pgadmin4.db' does not exist.
Entering initial setup mode...
NOTE: Configuring authentication for SERVER mode.

Enter the email address and password to use for the initial pgAdmin user account
:

Email address: appli_web@mail.com
Password:
Retype password:
Password must be at least 6 characters. Please try again.
Password:
Retype password:

The configuration database has been created at /root/.pgadmin/pgadmin4.db
(.pgadmin4) root@constance:~/.pgadmin4/lib/python2.7/site-packages/pgadmin4# python pgAdmin4.py
Starting pgAdmin 4. Please navigate to http://localhost:5050 in your browser.
```

/!\ bien noter l'adresse mail utilisée ainsi que le mot de passe si oubli faire:

```
constance@constance:~/.local/lib/python2.7/site-packages/pgadmin4$ rm /home/constance/.pgadmin/pgadmin4.db
constance@constance:~/.local/lib/python2.7/site-packages/pgadmin4$ python2.7 setup.py
```

Résumé des commandes pour pgAdmin4:

```
35 apt install python-dev virtualenv python-pip libpq-dev
36 virtualenv .pgadmin4
37 cd .pgadmin4/
38 source bin/activate
39 wget https://ftp.postgresql.org/pub/pgadmin/pgadmin4/v1.3/pip/pgadmin4-1.3-py2.py3-none-any.whl
40 pip install pgadmin4-1.3-py2.py3-none-any.whl
41 ls
42 cd lib/python2.7/site-packages/pgadmin4
43 ls
44 python setup.py
45 python pgAdmin4.py
```

Installation et configuration du server apache 2

Pour installer notre server apache2, c'est très simple, il nous suffit d'exécuter la commande suivante : `$apt install apache2`

Pour voir si l'installation s'est bien effectuée, il suffit d'aller à l'adresse de notre server.

Pour commettre l'ip de notre server il suffit de taper la commande : `$ip addr`

Création de l'application PHP avec CRUD

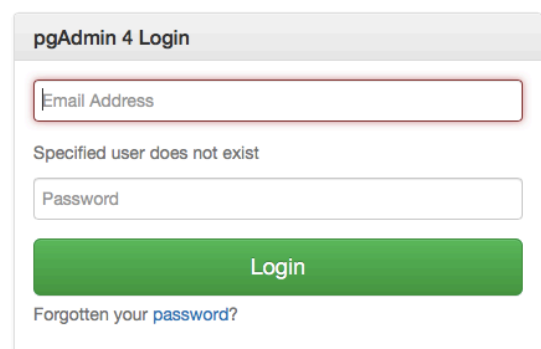
Pour la création de notre application, nous avons récupéré notre projet fait en laravel 4.2 qui présente un site de recettes de cuisine avec la possibilité de créer, modifier et supprimer des recettes ainsi que des ingrédients. Pour cela, nous avons dû changer les paramètres dans notre application. Il a fallu changer notamment changer les paramètres de la database car celle-ci prenait par défaut la connexion sur mysql. En redirigeant les paramètres de connexion sur notre base de données nous avons effectué les changements suivants vu que notre application était prévue pour être utilisée avec mysql:

```
'pgsql' => array(
  'driver' => 'pgsql',
  'host' => '10.211.55.9',
  'database' => 'appli_web',
  'username' => 'appli_web',
  'password' => 'appli',
  'charset' => 'utf8',
  'prefix' => '',
  'schema' => 'public',
),
```

Une fois ces paramètres changés, il nous suffit de refaire une migration pour créer les différentes tables de notre application. De cette façon nous avons maintenant une base de données sur pgsql.

```
$ php artisan migrate
```

Pour accéder il suffit d'entrer dans votre navigateur l'adresse de votre machine. Vous arriverez sur une page d'authentification :



pgAdmin 4 Login

Email Address

Specified user does not exist

Password

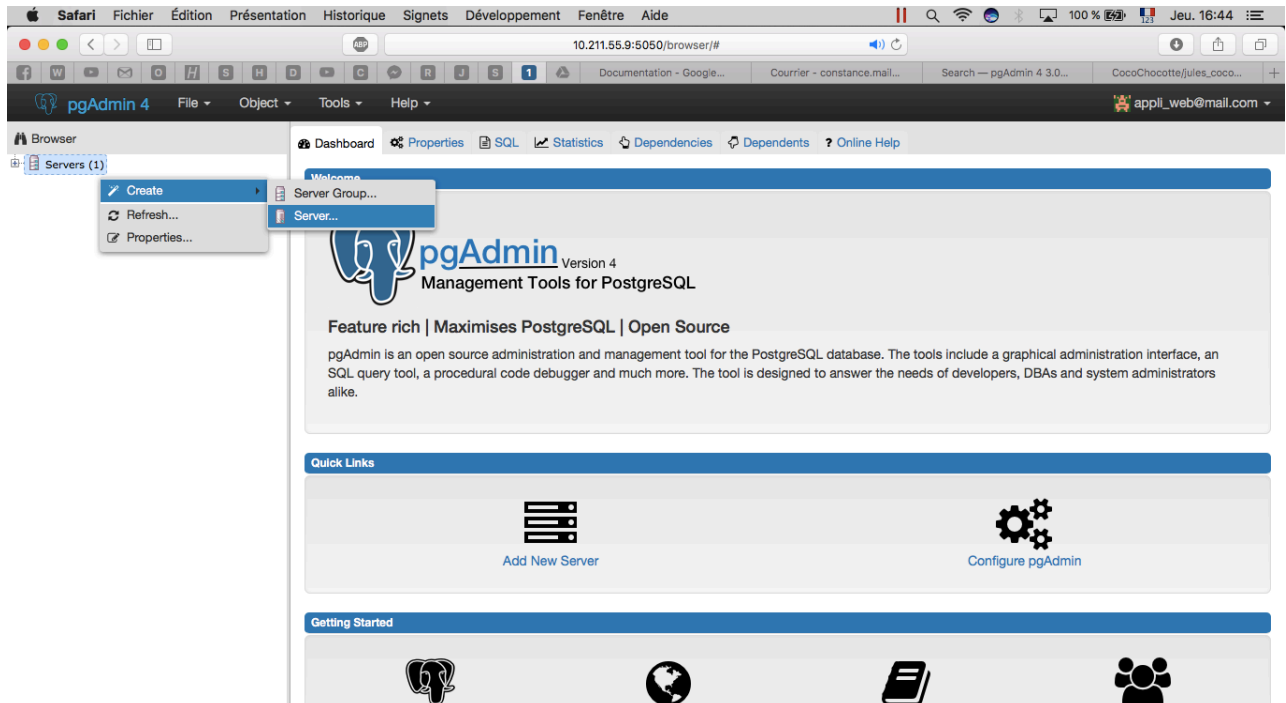
Login

[Forgotten your password?](#)

Entrez vos identifiants.

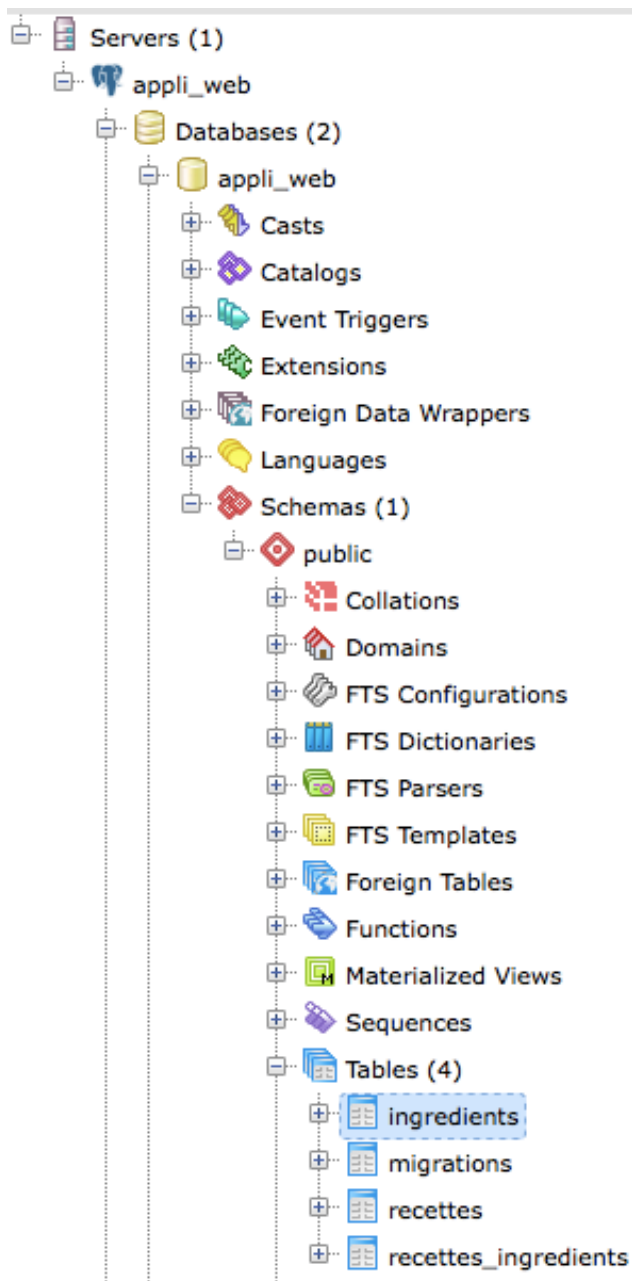
Vous arriver sur la page de Posgree.

Pour ajouter votre base :



Entrer les identifiants utilisées plus haut :

Create - Server	
General	Connection
Host name/address	127.0.0.1
Port	5432
Maintenance database	postgres
User name	appli_web
Password	
Save password?	<input type="checkbox"/>
Role	
SSL mode	Prefer



Vous apercevez que vos bases de données ont bien été créées.

Script de sauvegarde, suppression et restauration :

Pour exécuter ce script vous aurez besoin d'installer la commande date :

```
$ apt install date
```

```
1  #!/bin/bash
2  if [ ! -d "/backuppostgre" ];then
3  echo "Création de backuppostgre !";
4  sudo mkdir /backuppostgre
5  fi
6
7  while
8  echo "#####Voici mon menu#####"
9  echo "1: Sauvegarder"
10 echo "2: Restaurer"
11 echo "3: Supprimer +7j"
12 DATE=`date +%Y%m%d`
13 read choix
14 do
15 case $choix in
16     1) PGPASSWORD=appli pg_dump -h 127.0.0.1 -U appli_web > /backuppostgre/$DATE.backup$
17 echo "sauve";;
18     2) find /backuppostgre ls
19         echo "tappez votre choix dans la liste"
20         read choice
21         psql appli_web < $choice ;;
22     3) find . /backuppostgre $log -type f -mtime +7 -exec /bin/rm -f {} ;;
23
24     *) echo "tu t'es trompe";;
25 esac
26 done
```