

GUIDE D'INSTALLATION

PostgreSQL

- 2021 -

BRAHIMAN Koulibaly

Table des matières

1	PostgreSQL	3
1.1	Prérequis	3
1.2	Installation de PostgreSQL	3
1.3	Configuration de PostgreSQL	4
1.3.1	Basculer sur le compte utilisateur postgres	4
1.3.2	Accéder à une invite Postgres sans changer de compte	5
1.4	Création d'un nouveau rôle	5
1.5	Création d'une nouvelle base de données	6
1.6	Utiliser votre nouvelle utilisateur	6
1.7	Changer de base de données	7
1.8	Commandes utiles avec psql	7
2	pgAdmin4	7
2.1	Installation de pgAdmin 4	7
2.2	Configuration de pgAdmin4	8
2.3	Accéder à pgAdmin4	8
3	Pagila	11
4	Références	12

1 PostgreSQL

1.1 Prérequis

Avant d'installer PostgreSQL, il va falloir vérifier que la liste des paquets disponibles par le système de paquet ubuntu est bien à jour. Pour cela, il faudra faire la commande suivante :

```
1 # sudo apt update
```

A la suite de cette commande, vous devriez obtenir un résultat de cette nature :

```
(base) akuma06@akuma-pasukon:~$ sudo apt update
Atteint :1 http://ppa.launchpad.net/gnumdk/lollypop/ubuntu focal InRelease
Atteint :2 http://repo.mysql.com/apt/ubuntu focal InRelease
Atteint :3 https://dl.winehq.org/wine-builds/ubuntu focal InRelease
Atteint :4 http://dl.google.com/linux/chrome/deb stable InRelease
Atteint :5 http://fr.archive.ubuntu.com/ubuntu focal InRelease
Atteint :6 http://fr.archive.ubuntu.com/ubuntu focal-updates InRelease
Atteint :7 http://fr.archive.ubuntu.com/ubuntu focal-backports InRelease
Atteint :8 http://security.ubuntu.com/ubuntu focal-security InRelease
Atteint :9 https://packages.microsoft.com/repos/ms-teams stable InRelease
Atteint :10 https://download.docker.com/linux/ubuntu focal InRelease
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
```

FIGURE 1 – Résultat de la commande apt update

1.2 Installation de PostgreSQL

Une fois la liste des paquets mise à jour, il est possible d'installer PostgreSQL et le paquet *-contrib* qui ajoute quelques utilitaires et fonctionnalités supplémentaires :

```
1 # sudo apt install postgresql postgresql-contrib
```

```
Creating config file /etc/default/sysstat with new version
update-alternatives: utilisation de « /usr/bin/sar.sysstat » pour fournir
/bin/sar » (sar) en mode automatique
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.serv
ib/systemd/system/sysstat.service.
Paramétrage de postgresql-contrib (12+214ubuntu0.1) ...
Paramétrage de postgresql (12+214ubuntu0.1) ...
Traitement des actions différées (« triggers ») pour systemd (245.4-4ubu
...
Traitement des actions différées (« triggers ») pour man-db (2.9.1-1) ..
Traitement des actions différées (« triggers ») pour libc-bin (2.31-0ubu
```

FIGURE 2 – Résultat de la commande apt install

Il est possible de vérifier que tout s'est bien passé et que le service PostgreSQL est actif en tapant la commande :

```
1 # systemctl status postgresql
```

Il est aussi possible de vérifier que le service postgres est bien en position d'accepter des connexions en tapant la commande :

```

■ postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor preset: enabled)
   Active: active (exited) since Thu 2020-07-16 19:29:27 UTC; 11min ago
     Main PID: 3885 (code=exited, status=0/SUCCESS)
       Tasks: 0 (limit: 1075)
      Memory: 0B
      CGroup: /system.slice/postgresql.service

```

FIGURE 3 – Résultat de la commande `systemctl`

```

1 # sudo pg_isready

```

```

/var/run/postgresql:5432 - accepting connections

```

FIGURE 4 – Résultat de la commande `pgisready`

1.3 Configuration de PostgreSQL

Par défaut, Postgres utilise un concept appelé « rôles » pour gérer l'authentification et l'autorisation. Ils sont, à certains égards, semblables aux comptes classiques de type Unix, mais Postgres ne fait pas de distinction entre utilisateurs et groupes et préfère le terme plus souple de « rôles ».

Une fois installé, Postgres est configuré pour utiliser l'authentification ident, ce qui signifie qu'il associe les rôles Postgres à un compte système Unix/Linux correspondant. Si un rôle existe au sein de Postgres, un nom d'utilisateur Unix/Linux portant le même nom peut se connecter à ce rôle.

La procédure d'installation a créé un compte utilisateur nommé `postgres` qui est associé au rôle Postgres par défaut. Pour utiliser Postgres, vous pouvez vous connecter à ce compte.

Il y a deux façons d'utiliser ce compte pour accéder à Postgres. La première, un peu plus complexe, consiste à changer le compte utilisateur courant de votre terminal. La deuxième consiste à exécuter `postgres` en tant qu'utilisateur `postgres`.

1.3.1 Basculer sur le compte utilisateur `postgres`

Tout d'abord il va falloir basculer sur le compte utilisateur `postgres`. Pour ce faire, il faudra utiliser la commande :

```

1 # sudo -i -u postgres

```

```

(base) akuma06@akuma-pasukon:~$ sudo -i -u postgres
postgres@akuma-pasukon:~$

```

FIGURE 5 – Résultat de la commande `sudo -i -u postgres`

Si tout se passe bien, l'utilisateur dans votre terminal sera maintenant `postgres`. Vous pouvez maintenant accéder à l'invite PostgreSQL immédiatement en tapant :

```

1 # psql

```

De là, vous êtes libre d'interagir avec le système de gestion de la base de données si nécessaire. Sortez de l'invite PostgreSQL en tapant :

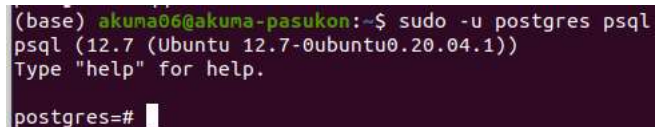
```
1 # postgres=# \q
```

1.3.2 Accéder à une invite Postgres sans changer de compte

Vous pouvez également exécuter la commande que vous souhaitez avec le compte postgres directement en utilisant `sudo`.

Par exemple, dans l'exemple précédent, il vous a été indiqué de vous rendre à l'invite Postgres en basculant d'abord sur l'utilisateur postgres puis en exécutant `psql` pour ouvrir l'invite Postgres. Vous pouvez faire cela en une seule étape en exécutant la commande unique `psql` en tant qu'utilisateur postgres avec privilèges `sudo`, comme ceci :

```
1 # sudo -u postgres psql
```



```
(base) akuma06@akuma-pasukon:~$ sudo -u postgres psql
psql (12.7 (Ubuntu 12.7-0ubuntu0.20.04.1))
Type "help" for help.

postgres=#
```

FIGURE 6 – Résultat de la commande `sudo -u postgres psql`

Cela vous connectera directement dans Postgres sans passer par le shell bash intermédiaire. Pour chaque commande que vous voudrez utiliser avec postgres (par la suite de ce tutoriel), il faudra ajouter dans ce cas au début : `sudo -u postgres`

Là encore, vous pouvez quitter la session Postgres interactive en tapant :

```
1 # postgres=# \q
```

1.4 Création d'un nouveau rôle

Actuellement, le rôle postgres est le seul à être configuré dans la base de données. Vous pouvez créer de nouveaux rôles à partir de la ligne de commande avec la commande `createuser`. L'indicateur `-interactive` vous invitera à indiquer le nom du nouveau rôle et vous demandera également si celui-ci doit disposer des autorisations de superutilisateur.

Si vous êtes connecté en tant que compte postgres, vous pouvez créer un nouvel utilisateur en tapant :

```
1 # createuser -P --interactive
```

Dans le cas où vous ne changez pas de compte (en utilisant la méthode en 3.2), vous devrez taper la commande suivante :

```
1 # sudo -u postgres createuser -P --interactive
```

La commande vous demandera donc le nom du "rôle" (nom d'utilisateur) que vous voulez créer, ensuite un mot de passe de connexion pour cet utilisateur et enfin si vous voulez lui accorder des droits super-utilisateurs (pour l'administration de la BDD entièrement).

```
(base) akuma06@akuma-pasukon:~$ sudo -u postgres createuser -P --interactive
Enter name of role to add: akuma06
Enter password for new role:
Enter it again:
Shall the new role be a superuser? (y/n) y
```

FIGURE 7 – Résultat de la commande `sudo -u postgres createuser -P --interactive`

Par défaut, chaque utilisateur que vous allez créer auront besoin d'un utilisateur ubuntu correspondant. Ainsi si l'utilisateur que vous venez de créer sur Postgres s'appelle "simplon", vous devez posséder un utilisateur ubuntu qui s'appelle "simplon" (par défaut, les ordinateurs simplon ont un utilisateur simplon).

Dans le cas contraire, vous allez devoir créer ce nouvel utilisateur sur votre machine avec la commande ci-dessous (en remplaçant "simplon" par le nom d'utilisateur postgres) :

```
1 # sudo adduser simplon
```

1.5 Création d'une nouvelle base de données

Une autre hypothèse posée par défaut par le système d'authentification Postgres est que tout rôle utilisé pour se connecter disposera d'une base de données du même nom à laquelle il pourra accéder.

Cela signifie que si l'utilisateur que vous avez créé dans la dernière section est appelé *simplon*, ce rôle tentera de se connecter à une base de données qui est également appelée *simplon* par défaut. Vous devrez donc créer la base de données appropriée avec la commande `createdb`.

Si vous êtes connecté en tant que compte postgres, vous devez taper :

```
1 # createdb simplon
```

Si vous utilisez la méthode définie en 3.2, vous allez devoir taper :

```
1 # sudo -u postgres createdb simplon
```

Si vous créez une base de données d'un nom différent de l'utilisateur, vous pouvez lui octroyer des privilèges d'accès en écriture et lecture en tapant dans `psql`, la commande SQL *GRANT PRIVILEGES*. On lance `psql` (comme en 3.1 ou 3.2) puis on tape la commande SQL *GRANT* suivante :

```
1 # CREATE DATABASE simplondb;
2 # GRANT ALL PRIVILEGES ON DATABASE simplondb to simplon;
```

La première est une autre manière de créer une base de données via SQL puis on grant tous les privilèges sur cette nouvelle base de données avec *GRANT*. Il n'est pas obligé de créer la base de données via SQL, on peut la créer avec les méthodes précédentes.

1.6 Utiliser votre nouvelle utilisateur

Une fois ce nouveau compte disponible, vous pouvez démarrer postgres de la même manière que défini en 3.1 ou en 3.2 :

```
1 # sudo -u simplon psql
```

Cependant, si votre compte utilisateur ubuntu possède le même nom que l'utilisateur postgres que vous venez de créer, vous pouvez le lancer simplement en tapant simplement la commande :

```
1 # psql
```

1.7 Changer de base de données

Vous pouvez changer de base de données en précisant l'argument `-d` avec la commande `psql` :

```
1 # psql -d simplon
```

1.8 Commandes utiles avec psql

- `\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 `psql` ou la commande en cours d'exécution
- `\l` : liste les bases de données existantes
- `\du` : liste les rôles (utilisateurs ou groupes)
- `\c <database> <username>` : connecte à la base `<database>` avec l'utilisateur `<username>`
- `\lt` : liste les tables de la base courante
- `\d <table>` : Description de la table `<table>` (colonne, type, ...)
- `\dn` : Liste tout le schéma de la base courante
- `\h ALTER TABLE` : affiche l'aide pour la commande `ALTER TABLE`

2 pgAdmin4

2.1 Installation de pgAdmin 4

pgAdmin4 n'est pas disponible dans les repositories d'Ubuntu. Il faut donc l'installer à partir du repository APT pgAdmin4. Il faut donc l'ajouter et le configurer. Ajoutez la clé publique du référentiel et créez le fichier de configuration du référentiel avec les deux commandes suivantes :

```
1 # curl https://www.pgadmin.org/static/packages_pgadmin_org.pub | sudo apt-key  
  add  
2 # sudo sh -c 'echo "deb https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$(  
  lsb_release -cs) pgadmin4 main" > /etc/apt/sources.list.d/pgadmin4.list &&  
  apt update'
```

Puis vous pouvez installer pgAdmin4 avec `apt` :

```
1 # apt install pgadmin4
```

La commande ci-dessus installera de nombreux packages requis, y compris le serveur Web Apache2 pour servir l'application `pgadmin4-web` en mode Web.

Une fois l'installation terminée, exécutez le script de configuration Web fourni avec le package binaire `pgadmin4` pour configurer le système afin qu'il s'exécute en mode Web. Vous serez invité à créer un e-mail et un mot de passe de connexion pgAdmin4, comme indiqué dans la capture d'écran ci-dessous.

Ce script configurera Apache2 pour servir l'application Web pgAdmin4, ce qui implique d'activer le module WSGI et de configurer l'application pgAdmin pour qu'elle se monte sur pgadmin4 sur le serveur Web afin que vous puissiez y accéder à l'adresse : *http://localhost/pgadmin4*

2.2 Configuration de pgAdmin4

Pour plus de sécurité et de facilité, il est préférable de configurer un utilisateur pour pgAdmin4. Pour cela, il suffit de se laisser guider et de répondre aux questions en exécutant le script :

```
1 # sudo /usr/pgadmin4/bin/setup-web.sh
```

```
(base) akuma06@akuma-pasukon:~$ sudo /usr/pgadmin4/bin/setup-web.sh
Setting up pgAdmin 4 in web mode on a Debian based platform...
Creating configuration database...
NOTE: Configuring authentication for SERVER mode.

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

Email address: demon.akuma06@gmail.com
Password:
Retype password:
pgAdmin 4 - Application Initialisation
=====

Creating storage and log directories...
We can now configure the Apache Web server for you. This involves enabling the wsgi module and configuring the pgAdmin 4 application to mount at /pgadmin4. Do you wish to continue (y/n)? y
The Apache web server is running and must be restarted for the pgAdmin 4 installation to complete. Continue (y/n)? y
Apache successfully restarted. You can now start using pgAdmin 4 in web mode at http://127.0.0.1/pgadmin4
```

FIGURE 8 – Résultat de setup-web.sh

2.3 Accéder à pgAdmin4

Une fois la page de connexion chargée, entrez l'adresse e-mail et le mot de passe que vous avez créés dans la section précédente lors de la configuration de pgAdmin4 pour qu'il s'exécute en mode Web (figure 9).

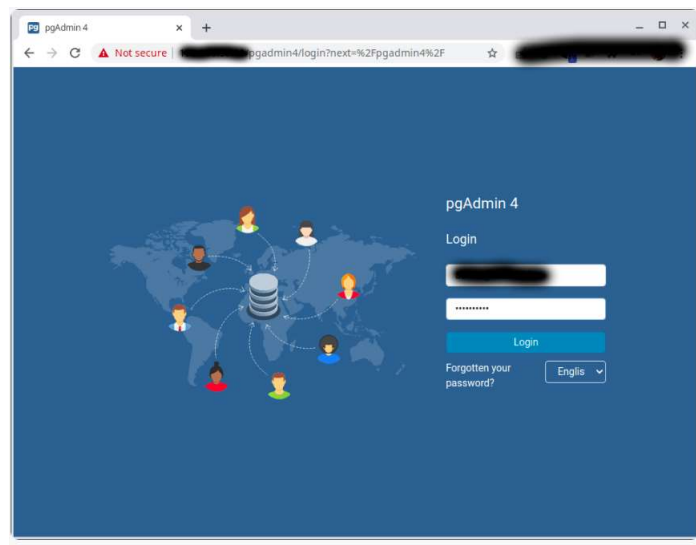


FIGURE 9 – Page de connexion pgAdmin4

Après une connexion réussie, vous arriverez dans le tableau de bord de l'application Web pgAdmin4. Pour vous connecter à un serveur, cliquez sur Ajouter un nouveau serveur comme indiqué dans la capture d'écran suivante (figure 10).

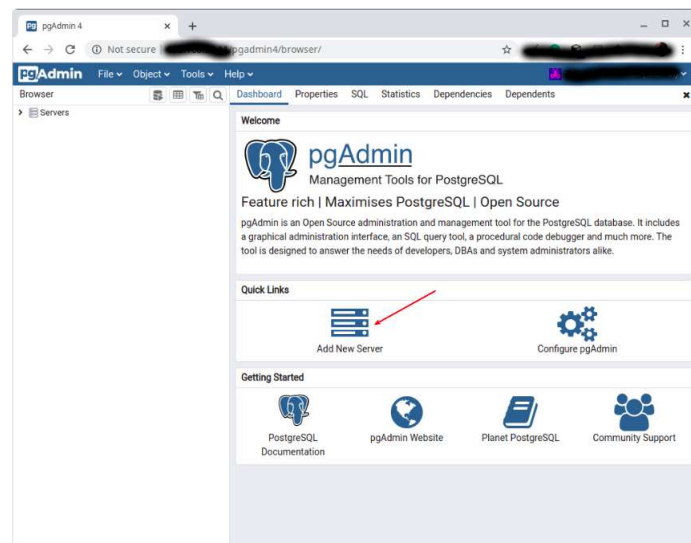


FIGURE 10 – Tableau de bord pgAdmin4

Ensuite, entrez la connexion dans les paramètres généraux (Nom, Groupe de serveurs et un commentaire). Cliquez ensuite sur Connexions comme indiqué dans la capture d'écran suivante (figure 11).

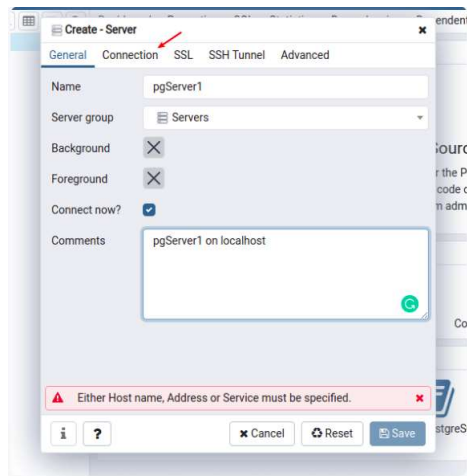


FIGURE 11 – Ajouter une connexion au serveur

Ensuite, entrez le nom d'hôte/l'adresse du serveur de base de données PostgreSQL, le numéro de port (laissez 5432 pour utiliser la valeur par défaut), sélectionnez la base de données de maintenance (qui devrait être postgres), entrez le nom d'utilisateur et le mot de passe de la base de données (figure 12).

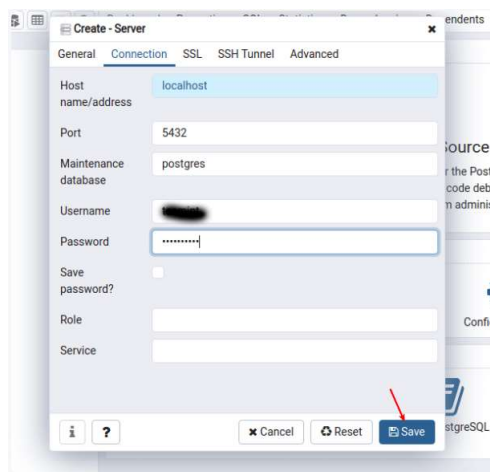


FIGURE 12 – Informations de connexion serveur postgres

Si les informations d'identification d'accès à la base de données sont correctes et que la configuration d'authentification serveur-client l'est également, pgAdmin4 doit se connecter avec succès au serveur de base de données (figure 13).

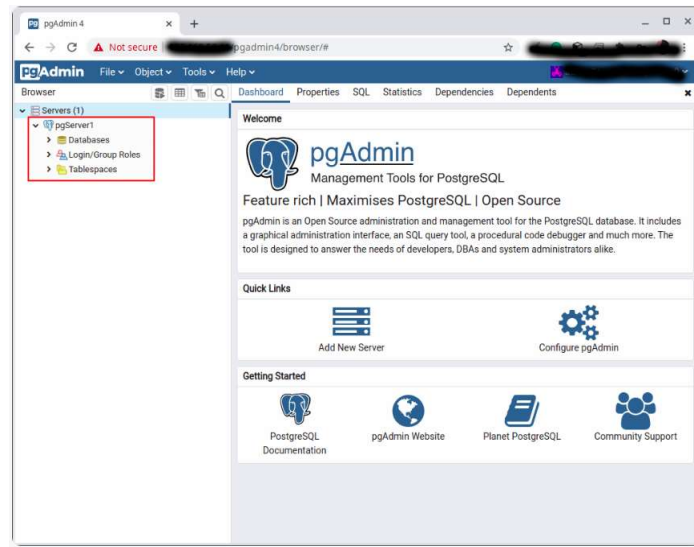


FIGURE 13 – Tableau de bord avec la connexion au serveur

3 Pagila

Depuis la page github (<https://github.com/devringunduz/pagila/>), vous pouvez télécharger le schéma et la data de la bdd sakila pour PostgreSQL (<https://raw.githubusercontent.com/devringunduz/pagila/master/insert-data.sql>).

Une fois téléchargés, aller sur pgAdmin4, cliquer sur le bouton d'exécution de requête SQL (1) puis sélectionner le bouton ouvrir un fichier sql (2) ensuite ajouter votre fichier SQL (3). Lorsque le fichier est ajouté (100%), il faudra penser à fermer la petite fenêtre et ouvrir le fichier pagila-insert-data.sql en cliquant de nouveau sur le bouton (2), en sélectionnant le fichier dans la liste qui apparaît et en appuyant sur le bouton select (4).

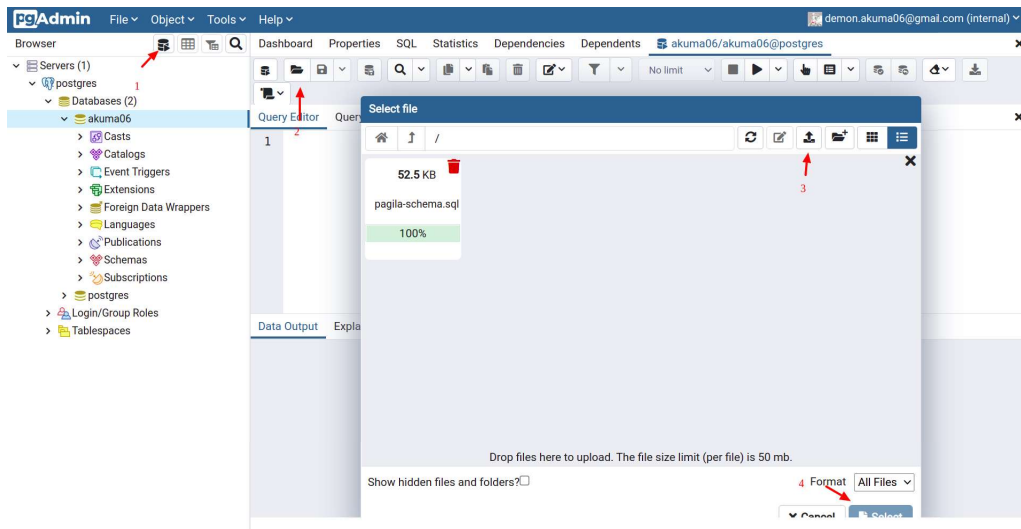


FIGURE 14 – Comment ajouter des fichiers sql sur pgAdmin4

Une fois le fichier ouvert, vous pouvez l'exécuter en cliquant sur le bouton "play".

4 Références

- Documentation PostgreSQL 12 (<https://www.postgresql.org/docs/12/index.html>)
- Documentation pgAdmin 4 (<https://www.pgadmin.org/docs/pgadmin4/4.23/index.html>)