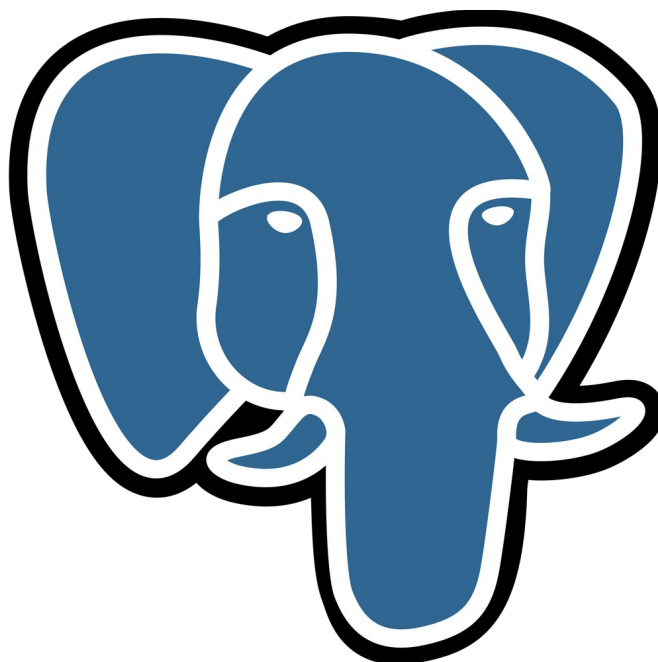


PostgreSQL



Sommaire

1 - Problématique.....	2
2 - Mise en place de PostgreSQL.....	2
2.1 - Installation du service.....	2
3 - Utilisation PostgreSQL.....	3
3.1 - Première connexion au serveur PostgreSQL.....	3
3.1.1 - Installation Client Linux.....	3
3.1.2 - Installation Client Windows.....	3
3.1.3 - Paramétrage de PostgreSQL pour connexion externe.....	4
3.1.4 - Création utilisateurs (locaux et externes) et connexion avec ces utilisateurs.....	6
3.2 - Créer une base de donnée.....	6
3.3 - Donner des privilèges à un utilisateur.....	7
4 - Annexes.....	8

1 - Problématique

Pour plusieurs services, une base de données est nécessaire afin de mettre en ordre les informations qui permettent aux services de fonctionner.

PostgreSQL est un service permettant de mettre en place un serveur SQL qui permet justement de remplir la tâche précédemment décrite.

2 – Mise en place de PostgreSQL

2.1 – Installation du service

Pour installer PostgreSQL, utiliser les commandes suivantes :

```
sudo sh -c 'echo "deb https://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'

wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

apt update

apt install postgresql
```

Ensuite, vérifier que l'installation s'est bien déroulé en vérifiant le statut du service de PostgreSQL :

```
sudo systemctl status postgresql
```

Si l'installation est correcte alors, il devrait donner comme retour « active(running) ».

3 – Utilisation PostgreSQL

PostgreSQL reste au bout du compte un serveur basé sur du MySQL, une grande partie des commandes reposent donc sur le langage SQL, nous allons donc les voir.

En parallèle, nous verrons des paramètres du serveur PostgreSQL.

3.1 – Première connexion au serveur PostgreSQL

Pour vous connecter au serveur PostgreSQL, vous aurez besoin d'un client postgresql, si vous restez sur machine serveur, le client est déjà installé.

En local, vous n'avez qu'à faire :

```
sudo -u postgres psql
```

Après avoir rentré le mot de passe que vous avez paramétré précédemment, vous serez connectés en root au serveur. Nous verrons la création d'autres utilisateurs après.

3.1.1 – Installation Client Linux

Cependant, si vous êtes sur une machine externe Linux, il faut utiliser les commandes suivantes pour installer le client :

```
sudo apt update  
sudo apt install postgresql-client
```

Il n'y a rien d'autre à faire.

3.1.2 – Installation Client Windows

Pour une machine basée sur Windows, installer le packet Postgres en suivant les consignes de l'installateur, vous trouverez le paquet à l'adresse suivante :

```
https://www.enterprisedb.com/downloads/postgres-postgresql-downloads
```

Une fois l'installation terminée, vous aurez accès à la commande « mysqlsh » sur le powershell de Windows, je ne m'étendrais pas plus dessus vis-à-vis de son utilisation, mais elle reste similaire à celle sur Linux.

3.1.3 – Paramétrage de PostgreSQL pour connexion externe

Sur la machine de PostgreSQL, nous allons modifier sur quelle adresse le serveur écoute, car initialement, elle ne s'écoute qu'elle-même, cela peut être un problème si vous essayez d'y accéder depuis l'extérieur.

Pour modifier ce paramètre, faites :

```
sudo nano /etc/postgresql/15/main/postgresql.conf
```

Puis changer « bind-address » de localhost à *:

```
#-----  
# CONNECTIONS AND AUTHENTICATION  
#-----  
  
# - Connection Settings -  
  
#listen_addresses = 'localhost'      # what IP address(es) to listen on;  
#                                     # comma-separated list of addresses;  
#                                     # defaults to 'localhost'; use '*' for all  
#                                     # (change requires restart)  
port = 5432                          # (change requires restart)  
max_connections = 100                # (change requires restart)  
#superuser_reserved_connections = 3  # (change requires restart)  
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories  
#                                     # (change requires restart)
```

Après avoir quitté et sauvegarder le fichier, redémarrer le serveur :

```
sudo systemctl restart postgresql
```

PostgreSQL écoute désormais toutes les connexions qui essaient d'accéder à lui. Mais vous devez encore créer des utilisateurs qui ont le droit de se connecter à lui.

Pour vos utilisateurs externes, vous aurez aussi besoin de rajouter une ligne dans pg_hba.conf

```
nano /etc/postgresql/15/main/pg_hba.conf
```

#	TYPE	DATABASE	USER	ADDRESS	METHOD
# "local" is for Unix domain socket connections only					
local	all		all		peer
# IPv4 local connections:					
host	all		all	127.0.0.1/32	scram-sha-256
host	all		all	0.0.0.0/0	scram-sha-256
# IPv6 local connections:					
host	all		all	::1/128	scram-sha-256
# Allow replication connections from localhost, by a user with the					
# replication privilege.					
local	replication		all		peer
host	replication		all	127.0.0.1/32	scram-sha-256
host	replication		all	::1/128	scram-sha-256

Maintenant, vous êtes sûr que vos utilisateurs pourront se connecter sans problème.

3.1.4 – Création utilisateurs (locaux et externes) et connexion avec ces utilisateurs

Pour créer des utilisateurs, soyez sur la machine ayant le serveur PostgreSQL et connectez-vous en root au serveur :

```
sudo -u postgres psql
```

Puis pour créer un utilisateur, faites : (répéter opération pour chaque utilisateur)
`CREATE USER <Nom utilisateur> WITH PASSWORD « <Mot de passe de l'utilisateur> » ;`

Enfin, pour, vous connectez depuis une machine externe, faites :

```
psql -U <Nom utilisateur> -h <IP du serveur Postgresql> -W
```

3.2 – Créer une base de donnée

Toujours en root sur le serveur PostgreSQL faites :

```
CREATE DATABASE <Nom de la database> ;
```

Puis pour vous mettre dans la database elle-même, faites :

```
Psql -h <IP PostgreSQL> -U <Utilisateur> -d <Database>
```

Pour créer des tables dans la base de données, faites :

```
CREATE TABLE <Nom de la table>
(
    <Nom colonne1> <Type de données>,
    <Nom colonne2> <Type de données>,
    <Nom colonne3> <Type de données>,
    <Nom colonne4> <Type de données>,
);
```

En type de données, vous pouvez utiliser ceux-ci principalement :

```
integer
varchar(<Longueur chaine de caractère>)
date
```

Dans une table, il faut aussi une colonne qui soit une « clé primaire », c'est elle qui sert de référence, il ne peut donc pas y avoir deux fois la même clé primaire ; cela mènerait à une erreur.

Pour l'exemple, voici une table qui n'a qu'une seule valeur qui est la clé primaire :

```
CREATE TABLE exemple
(
    id integer primary key not null,
);
```

Ainsi, la valeur de la colonne d'id sera toujours un nombre entier et la valeur de la case ne pourra pas être nul.

3.3 – Donner des privilèges à un utilisateur

Toujours en root de mysql, nous allons chercher à donner des droits à un utilisateur pour qu'il puisse modifier la ou les bases de données à sa guise.

Pour donner les pleins droits à un utilisateur, faites :

```
GRANT ALL PRIVILEGES ON ALL DATABASE TO '<Nom utilisateur>';
```

La désignation de la base de données se passe au niveau des *. Donc pour donner les pleins droits à une seule base de données à un utilisateur, il faut faire :

```
GRANT ALL PRIVILEGES ON DATABASE <Nom de la DB> TO '<Nom utilisateur>';
```

Les privilèges existant dans PostgreSQL sont SELECT, INSERT, UPDATE et DELETE ; ALL les applique tous.

4 - Annexes

Fiche de recette

Vérification de l'opérationnalité de la solution mise en œuvre : PostgreSQL

Description du test :

1. Vérification statut du service
2. Tenter de se connecter au serveur
3. Tenter de faire des modifications sur le serveur

Résultats Attendus :

1. Statut du service en active(running)
2. Connexion réussie
3. Modifications appliquées

Réception Globale : Serveur PostgreSQL
Auteurs: Timothée LIGNIERE

Date: 09/04/24

Reçu : ☐

Reçu avec réserve : ☐

Refusé : ☐

Commentaire :

Recette étape par étape *

** (pour chaque étape, vous devez élaborer dans un fichier distinct un scénario détaillé à faire appliquer au « client » venant valider votre solution)*

Réception Etape 1: Vérification du statut du service à l'aide de la commande « `sudo systemctl status postgresql` » qui devrait renvoyer « `active(running)` »

Reçu : ☐

Reçu avec réserve : ☐

Refusé : ☐

Commentaire :

Réception Etape 2 : Connectez vous de manière locale et distante au serveur

Reçu : ☐

Reçu avec réserve : ☐

Refusé : ☐

Commentaire :

Réception Etape 3 : Faites des modifications sur une base de données ou une table d'une base

Reçu : ☐

Reçu avec réserve : ☐

Refusé : ☐

Commentaire :