

Tutoriel PostgreSQL

Configuration et gestion des utilisateurs

Projet BDD CIPA4, ISEN, 2025-2026

Connexion au serveur	3
Modifier le mot de passe de l'utilisateur Linux admin	3
Paramétrage de sécurité des accès du serveur Postgresql	3
Localiser le fichier pg_hba.conf.....	3
Ouvrir le fichier pour l'éditer.....	3
Comprendre la configuration	3
Après modification	4
Création de la base de données du projet	5
Connexion à la base de données depuis les outils GUI.....	6
Dans DBeaver	6
Dans DataGrip.....	6
Sauvegardes et restaurations	7
Sauvegarder une base de données sur le serveur (en local)	7
Restaurer une base de données sur le serveur	7
Bonnes pratiques pour les sauvegardes.....	7
Transférer un fichier vers le serveur avec SCP	7
Autres commandes utiles	8
Gestion du service PostgreSQL.....	8
Gestion des utilisateurs Linux	8
Gestion des utilisateurs PostgreSQL	9

Connexion au serveur

Les serveurs sont accessibles depuis le réseau de l'ISEN. Pour se connecter, ouvrir le terminal et saisir :

```
ssh admin@10.30.51.xxx
```

puis saisir le mot de passe. Vous arrivez ensuite sur le serveur.

Modifier le mot de passe de l'utilisateur Linux admin

On commence par modifier le mot de passe de l'utilisateur admin :

```
sudo passwd admin
```

Le système vous demande de saisir l'ancien mot de passe puis deux fois le nouveau mot de passe. **N'oubliez pas votre nouveau mot de passe !**

Quitter la session ssh, notez votre mot de passe et reconnectez-vous avec le nouveau mot de passe :

```
exit
```

```
ssh admin@10.30.51.181
```

Paramétrage de sécurité des accès du serveur Postgresql

Ce paramétrage est réalisé dans le fichier pg_hba.conf.

Localiser le fichier pg_hba.conf

```
# Sortir de psql si nécessaire
\q

# Trouver l'emplacement du fichier
sudo -u postgres psql -c "SHOW hba_file;"
```

Vous devriez trouver cet emplacement : /etc/postgresql/13/main/pg_hba.conf.

Ouvrir le fichier pour l'éditer

```
sudo nano /etc/postgresql/13/main/pg_hba.conf
```

Comprendre la configuration

Ordre important : PostgreSQL lit pg_hba.conf de haut en bas et applique la **première règle correspondante**. Mettez les règles les plus spécifiques en premier.

Configuration recommandée type

```
# TYPE DATABASE USER ADDRESS METHOD
# BLOQUER l'accès distant pour postgres
host all postgres 0.0.0.0/0 reject
host all postgres ::/0 reject

# Connexions locales (socket Unix)
local all postgres peer
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

# replication privilege.
local replication all scram-sha-256
host replication all 127.0.0.1/32 scram-sha-256
host replication all ::1/128 scram-sha-256
```

Recommandations par méthode d'authentification

À privilégier :

- **scram-sha-256** - méthode la plus sécurisée pour l'authentification par mot de passe
- **peer** - excellente pour les connexions locales Unix (vérifie l'utilisateur système)

À éviter :

- **trust** - ⚠ DANGEREUX : aucune authentification, à n'utiliser QUE temporairement en développement local
- **md5** - obsolète, utiliser **scram-sha-256** à la place
- **password** - transmet le mot de passe en clair, jamais en production

Quelques bonnes pratiques

1. **Principe du moindre privilège** : limitez l'accès à des bases/utilisateurs/IPs spécifiques
2. **Séparez les utilisateurs** : un utilisateur applicatif \neq postgres superuser
3. **Désactivez l'accès postgres à distance** (premières lignes de la configuration ci-dessus)

Après modification

Rechargez la configuration :

```
sudo systemctl reload postgresql
# ou
sudo -u postgres psql -c "SELECT pg_reload_conf();"

```

Création de la base de données du projet

Aller dans psql

```
sudo -u postgres psql
```

Créer l'utilisateur "isen-projet"

```
CREATE USER "isen-projet" WITH PASSWORD 'mot_de_passe_projet' LOGIN;
```

Notez le mot de passe choisi.

Créer la base de données avec cet utilisateur comme owner

```
CREATE DATABASE "isen-projet" OWNER "isen-projet";
```

Vérifier la création

```
\l
```

Vous devriez voir votre base de données avec `isen-projet` comme propriétaire.

Se connecter à la base de données

```
\c isen-projet
```

Vous êtes alors connecté à `isen-projet` en tant qu'utilisateur `postgres` car vous êtes entré dans PG avec cet utilisateur.

Vous pouvez sortir de `psql` et essayer de vous connecter en tant qu'utilisateur `isen-projet` :

```
\q
```

```
# en dehors de psql
psql -U isen-projet -d isen-projet
```

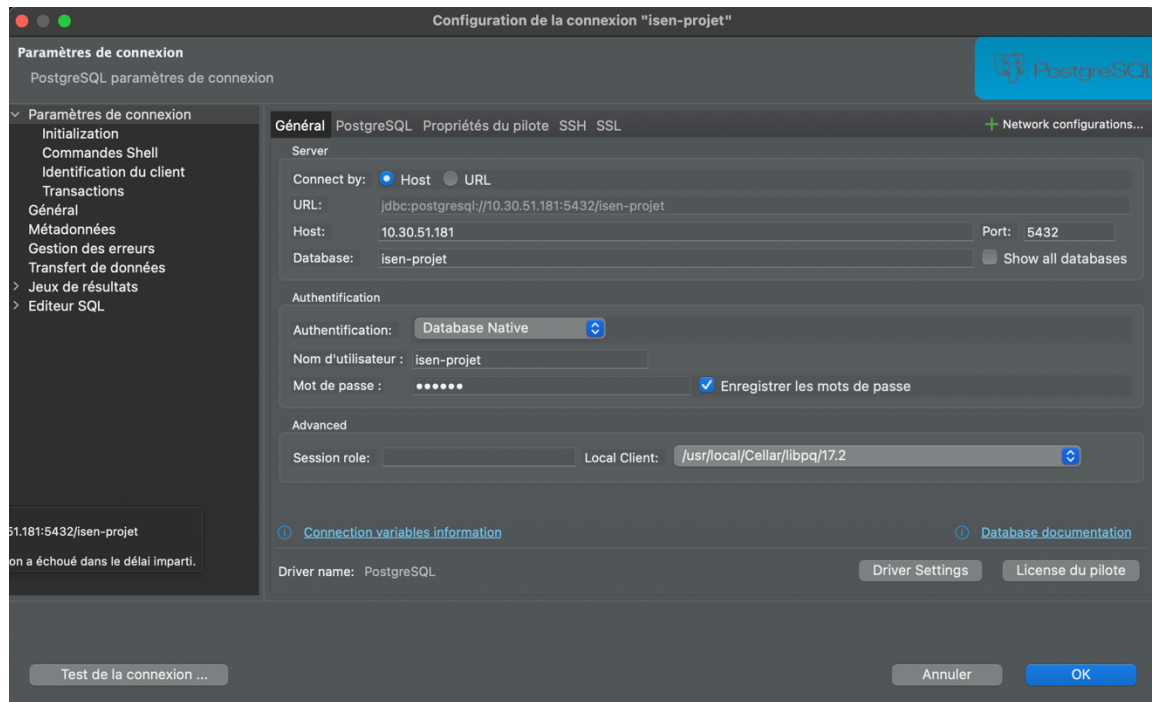
Cela ne devrait pas fonctionner car nous avons choisi le mode `peer` pour les connexions locales. Il regarde donc s'il existe un user Linux `"isen-projet"`, et ce n'est pas le cas.

Par contre, si vous ajoutez l'option `-h localhost`, cela devrait fonctionner car on passe alors par notre paramétrage `host` :

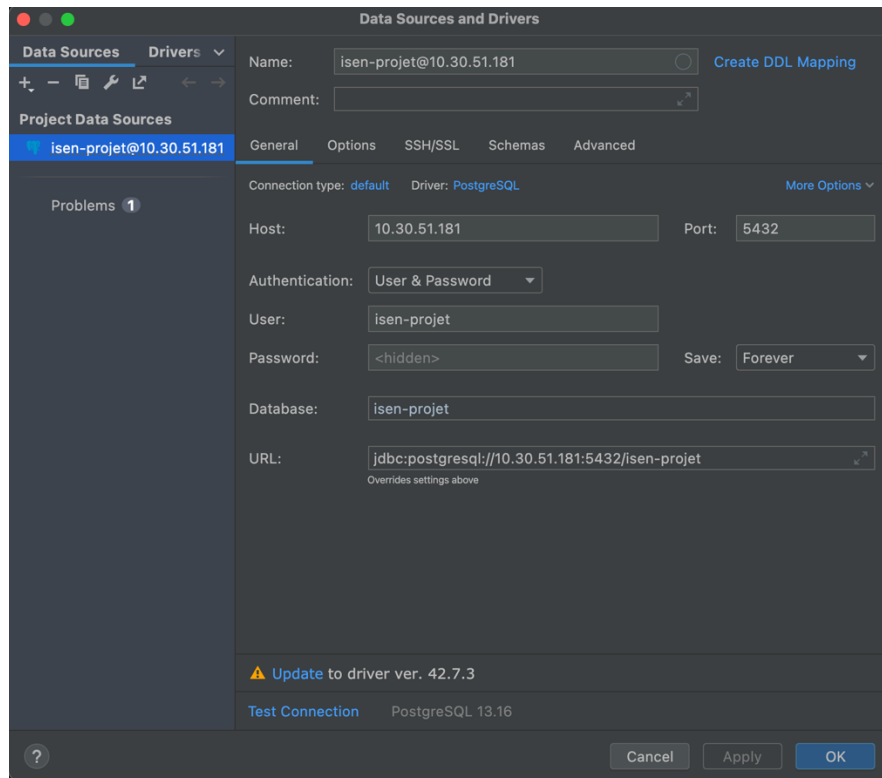
```
psql -U isen-projet -d isen-projet -h localhost
```

Connexion à la base de données depuis les outils GUI

Dans DBeaver



Dans DataGrip



Sauvegardes et restaurations

Sauvegarder une base de données sur le serveur (en local)

Se connecter au serveur en SSH

```
ssh admin@10.30.51.xxx
```

Sauvegarder une base complète (format SQL)

```
pg_dump -U isen-projet -d isen-projet > sauvegarde.sql
```

La sauvegarde se trouve dans le répertoire sur lequel vous êtes positionné. Vous pouvez utiliser l'alternative suivante pour placer la sauvegarde dans le répertoire home de l'utilisateur (pour admin : /home/admin /sauvegarde.sql) :

```
pg_dump -U isen-projet -d isen-projet > ~/sauvegarde.sql
```

Sauvegarder depuis votre machine locale

```
ssh admin@10.30.51.xxx "pg_dump -U isen-projet -d isen-projet" > sauvegarde.sql
```

Restaurer une base de données sur le serveur

Depuis le serveur, créer la nouvelle base de données

```
sudo -u postgres createdb new-isen-projet
```

Restaurer depuis un fichier SQL (déjà sur le serveur)

```
psql -U isen-projet -d new-isen-projet < ~/sauvegarde.sql
```

Bonnes pratiques pour les sauvegardes

1. **Tester les restaurations** - une sauvegarde non testée n'est pas une sauvegarde
2. **Conserver des copies locales** - toujours télécharger les sauvegardes importantes sur votre machine
3. **Stocker les sauvegardes hors du serveur** - utiliser `scp` pour les récupérer régulièrement
4. **Conserver plusieurs versions** - ne pas écraser la sauvegarde précédente

Transférer un fichier vers le serveur avec SCP

Si vous travaillez depuis votre machine locale et devez transférer un fichier vers le serveur PostgreSQL :

Transférer un fichier unique

```
scp test.txt admin@10.30.51.181:/home/admin/
```

Autres commandes utiles

Gestion du service PostgreSQL

Démarrer PostgreSQL

```
sudo systemctl start postgresql
```

Arrêter PostgreSQL

```
sudo systemctl stop postgresql
```

Redémarrer PostgreSQL

```
sudo systemctl restart postgresql
```

Recharger la configuration (sans redémarrage complet)

```
sudo systemctl reload postgresql  
# ou  
sudo -u postgres psql -c "SELECT pg_reload_conf();"
```

Vérifier le statut de PostgreSQL

```
sudo systemctl status postgresql
```

Gestion des utilisateurs Linux

Voir les détails d'un utilisateur

```
id nom_utilisateur
```

Lister les groupes d'un utilisateur

```
groups nom_utilisateur
```


Lister les utilisateurs "réels"

```
grep -E '/bin/(bash|sh|zsh)$' /etc/passwd | cut -d: -f1
```

Créer un utilisateur


```
sudo adduser USERNAME
```

Transformer l'utilisateur en administrateur

```
sudo usermod -aG sudo USERNAME
```

Explication de la commande :

- `usermod` : modifie un compte utilisateur
- `-aG` : ajoute (a) l'utilisateur à un groupe (G) supplémentaire sans le retirer des autres groupes
- `sudo` : nom du groupe donnant les privilèges administrateur
- `USERNAME` : nom de l'utilisateur à modifier

 **IMPORTANT** : L'utilisateur doit **se déconnecter et se reconnecter** pour que les changements de groupe prennent effet !

Gestion des utilisateurs PostgreSQL

Lister les bases de données

```
\l
```

Lister les utilisateurs et leurs rôles

```
\du
```

Se connecter à une base spécifique

```
\c nom_base_de_donnees
```

Modifier le mot de passe d'un utilisateur

```
ALTER USER nom_utilisateur WITH PASSWORD 'nouveau_mot_de_passe';
```

Supprimer un utilisateur

```
DROP USER nom_utilisateur;
```

Supprimer une base de données

```
DROP DATABASE nom_base_de_donnees;
```

Révoquer des privilèges

```
REVOKE ALL PRIVILEGES ON DATABASE nom_base_de_donnees FROM nom_utilisateur;
```