

# Migration d'une version majeure de PostgreSQL à une autre

**NB** : instructions pour le passage de PostgreSQL 15 à PostgreSQL 17 (Debian Trixie). Voir [ici](#) pour de 9.1 à 9.4 (Debian Jessie), [ici](#) pour de 9.4 à 9.6 (Debian Stretch), [ici](#) pour de 9.6 à 11 (Debian Buster), [ici](#) pour de 11 à 13 (Debian Bullseye) et [ici](#) pour de 13 à 15 (Debian Bookworm).

**NB** : mettez vous dans un `tmux` avant de commencer la procédure. Prudence est mère de sûreté.

**NB** : si vous utilisez l'extension [PostGis](#), optez pour la [méthode moins rapide](#). N'oubliez pas d'installer le paquet de l'extension `postgis` pour la nouvelle version de PostgreSQL !

**Attention** : si vous avez des fichiers de configuration dans `/etc/postgresql/XX/main/conf.d`, ils ne seront pas copiés dans le dossier de configuration de la nouvelle version par `pg_upgrade_cluster` ! Pensez à les copier à la main.

## Pré-requis

Vérifiez les paquets PostgreSQL installés sur votre système avec la commande suivante :

```
dpkg -l | grep postgresql
```

Ça vous permettra de voir si vous avez des extensions installées, comme `postgresql-15-repack`, et donc d'installer la version qui va bien pour la nouvelle version de PostgreSQL (en l'occurrence, `postgresql-17-repack`).

## Méthode rapide

On stoppe les clusters PostgreSQL

```
systemctl stop postgresql
```

On vire le cluster de la nouvelle version (normalement vide si on vient juste de l'installer : faire gaffe à ne pas laisser passer de temps entre l'installation de la nouvelle version et la migration des données, pour que personne n'utilise le nouveau cluster)

```
pg_dropcluster --stop 17 main
```

On migre les données

```
pg_upgradecluster -m upgrade 15 main
```

## ATTENTION

Si vous avez mis des `shared_preload_libraries` dans la configuration de votre ancien cluster, il y a des chances que `pg_upgradecluster -m upgrade 15 main` se foire (mais pas si on utilise la méthode `dump` [décrite plus bas](#)).

La solution est simple : créez le répertoire `/etc/postgresql/17/main/conf.d` et mettez-y un fichier dont le nom se termine par `.conf` (genre `shared_preload_libraries.conf`).

Dans ce fichier, mettez la configuration de vos `shared_preload_libraries` et ça devrait être bon.

Il faut savoir que cette commande copie les données de l'ancien cluster vers le nouveau. Il vous faut donc avoir au moins une fois la place de `/var/lib/postgresql/13` de disponible. Un contournement est d'utiliser l'option `--link` qui utilisera des [hard links](#) plutôt qu'une copie. Par contre, si quelque chose foire, vous foirez votre ancien cluster avec, c'est donc dangereux.

On redémarre le cluster (le 17 pour le coup) :

```
systemctl start postgresql
```

On lance l'analyse du nouveau cluster :

```
sudo -u postgres /usr/lib/postgresql/17/bin/vacuumdb --all --analyze-in-stages
```

Si vous utilisiez des extensions, allez dans `/var/log/postgresql`, vous aurez un dossier qui commence par `pg_upgrade` et qui contiendra un script un autre pour supprimer l'ancien cluster et un autre pour mettre à jour vos extensions. Faites alors (**Attention** : je n'ai pas encore testé cette partie, ça vient des tutos des versions précédentes mais vu que la commande précédente a changé, il est possible que celle-ci aussi) :

```
sudo -u postgres psql -f /var/log/postgresql/pg_upgradecluster-15-17-  
main*/update_extensions.sql
```

# Méthode moins rapide

Cette méthode fait un `pg_dump` et un `pg_restore`. C'est infiniment plus long quand on a de grosses bases de données, mais ça donne un cluster bien propre. Tellement propre que des fois ça foire pour cause de clés dupliquées ?

Vous aurez compris, je n'aime pas tellement cette méthode. Elle a cependant l'avantage d'éviter les problème d'index, vu que ça reconstruit les indexes (ce qui participe à la lenteur de la méthode).

```
systemctl start postgresql  
pg_upgradecluster -m dump 15 main
```

# Fin de migration, partie commune aux deux méthodes

On teste les applis qui utilisent PostgreSQL.

Si ça fonctionne, on vire les anciennes données

```
pg_dropcluster 15 main --stop
```

On vire l'ancienne version de PostgreSQL

```
apt-get autoremove --purge postgresql-15 postgresql-client-15
```

[Source](#)

---

Révision #21

Créé 2020-01-23 16:25:03 CET par Luc

Mis à jour 2025-09-15 14:55:43 CEST par Luc