

Migration d'une version majeure de PostgreSQL à une autre

NB : instructions pour le passage de PostgreSQL 9.6 à PostgreSQL 11 (Debian Buster). Voir [ici](#) pour de 9.1 à 9.4 (Debian Jessie) et [ici](#) pour de 9.4 à 9.6 (Debian Stretch).

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](#), ça vous évitera bien des soucis.

Attention !

Quand on fait la mise à jour vers Debian Buster, il est nécessaire de réindexer les bases. On a d'ailleurs ce message à l'installation de PostgreSQL 11 :

```
Existing PostgreSQL databases need to be reindexed
```

```
When upgrading from stretch to buster, the glibc locale data is upgraded. Specifically, this changes how PostgreSQL sorts data in text indexes. To avoid corruption, such indexes need to be REINDEXed immediately after upgrading the locales or locales-all packages, before putting back the database into production.
```

```
Suggested command: sudo -u postgres reindexdb --all
```

```
Alternatively, upgrade the databases to PostgreSQL 11 using pg_upgradecluster. (This uses pg_dump by default which will rebuild all indexes. Using -m upgrade or pg_upgrade is not safe because it preserves the now-wrong index ordering.)
```

```
Refer to the PostgreSQL Wiki for more information:
```

```
https://wiki.postgresql.org/wiki/Locale\_data\_changes
```

Donc, **dès la fin du apt dist-upgrade** on réindexe les bases :

```
sudo -u postgres reindexdb --all
```

Méthode rapide

On stoppe les clusters PostgreSQL

```
service postgresql stop
```

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 11 main
```

On migre les données

```
pg_upgradecluster -m upgrade 9.6 main
```

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 9.6 main` se foire (mais pas si on utilise la méthode `dump` décrite plus bas).

La solution est simple : créez le répertoire `/etc/postgresql/11/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/9.6` 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 11 pour le coup) :

```
systemctl start postgresql
```

On a normalement déjà réindexé la base en 9.6, donc la copie en 11 devrait avoir un index correct. Dans le doute (je préfère perdre du temps qu'un cluster PostgreSQL) :

```
sudo -u postgres reindexdb --all
```

Allez dans `/var/log/postgresql`, vous aurez un dossier qui commence par `pg_upgrade` et qui contiendra deux scripts. Un pour réanalyser votre nouveau cluster, un autre pour supprimer l'ancien. On supprime l'ancien avec `apt-get` plus loin, vous n'avez donc qu'à lancer le script de réanalyse :

```
sudo -u postgres /var/log/postgresql/pg_upgradecluster-9.6-11-main.6bmx/analyze_new_cluster.sh
```

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).

```
service postgresql start  
pg_upgradecluster -m dump 9.6 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 9.6 main --stop
```

On vire l'ancienne version de PostgreSQL

```
apt-get autoremove --purge postgresql-9.6
```

[Source](#)

From:

<https://wiki.fiat-tux.fr/> - **Wiki Fiat Tux**

Permanent link:

<https://wiki.fiat-tux.fr/admin:bdd:pgsql:migration-version-majeure>

Last update: **2019/10/14 11:47**

