

Przeniesienie klastra na Linuxie

Przygotowania starej usługi postgresa

1. **Przed całym procesem należy wykonać kopię bezpieczeństwa!**

2. Przechodzimy do katalogu:

```
sudo nano /etc/postgresql/10/main/pg_hba.conf
```

Struktura powinna wyglądać następująco:

```
# 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 trust
host all all 0.0.0.0/0 md5
# IPv6 local connections:
host all all ::1/128 trust
```

3. Restartujemy usługę postgresa

```
sudo systemctl restart postgresql
```

Przygotowania nowej usługi postgresa

1. Aktualizujemy pakiety i instalujemy Postgresa 12

```
sudo apt-get update
```

```
sudo apt-get install postgresql-12
```

2. Edytujemy plik, w którym są zawarte zezwolenia do połączenia z bazą danych:

```
sudo nano /etc/postgresql/12/main/pg_hba.conf
```

Struktura powinna wyglądać następująco:

```
# 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 trust
host all all 0.0.0.0/0 trust
# IPv6 local connections:
host all all ::1/128 trust
```

3. Edytujemy plik konfiguracyjny: (usuwamy "#" przed listen_addresses, port i max connections)

```
sudo nano /etc/postgresql/12/main/postgresql.conf
```

Zmieniamy konfigurację według poniższego przykładu

```
#-----  
# CONNECTIONS AND AUTHENTICATION  
#-----  
# - Connection Settings -  
  
listen_addresses = '*'           # what IP address(es) to listen on;  
                                # comma-separated list of addresses;  
                                # defaults to 'localhost'; use '*' for all  
                                # (change requires restart)  
port = 5454                      # (change requires restart)  
max_connections = 300           # (change requires restart)
```

4. Restartujemy usługę postgresa, aby zmiany weszły w życie

```
sudo systemctl restart postgresql
```

5. Zatrzymujemy usługę postgresa

```
sudo systemctl stop postgresql
```

Przeniesienie i optymalizacja nowego klastra

1. Logujemy się na użytkownika postgresa

```
sudo su postgres
```

2. Przechodzimy do lokalizacji, w której użytkownik postgres ma uprawnienia do tworzenia plików np. do katalogu /tmp

```
cd /tmp
```

3. Wykonujemy skrypt, który wykona migracje i potrzebne pliki.

```
/usr/lib/postgresql/12/bin/pg_upgrade \  
--old-datadir=/var/lib/postgresql/10/main \  
--new-datadir=/var/lib/postgresql/12/main \  
--old-bindir=/usr/lib/postgresql/10/bin \  
--new-bindir=/usr/lib/postgresql/12/bin \  
--old-options '-c config_file=/etc/postgresql/10/main/postgresql.conf' \  
--new-options '-c config_file=/etc/postgresql/12/main/postgresql.conf'
```

Po poprawnym wykonaniu powinny być następujące komunikaty:

```
Performing Consistency Checks  
-----  
Checking cluster versions                               ok  
Checking database user is the install user             ok  
Checking database connection settings                 ok  
Checking for prepared transactions                   ok  
Checking for system-defined composite types in user tables ok  
Checking for reg* data types in user tables           ok  
Checking for contrib/isn with bigint-passing mismatch ok  
Checking for tables WITH OIDS                        ok  
Checking for invalid "sql_identifier" user columns   ok  
Creating dump of global objects                       ok  
Creating dump of database schemas                    ok  
  
Checking for presence of required libraries           ok  
Checking database user is the install user             ok  
Checking for prepared transactions                   ok
```

```
Checking for new cluster tablespace directories          ok
If pg_upgrade fails after this point, you must re-initdb the
new cluster before continuing.

Performing Upgrade
-----
Analyzing all rows in the new cluster                  ok
Freezing all rows in the new cluster                  ok
Deleting files from new pg_xact                       ok
Copying old pg_xact to new server                     ok
Setting oldest XID for new cluster                    ok
Setting next transaction ID and epoch for new cluster ok
Deleting files from new pg_multixact/offsets          ok
Copying old pg_multixact/offsets to new server        ok
Deleting files from new pg_multixact/members          ok
Copying old pg_multixact/members to new server        ok
Setting next multixact ID and offset for new cluster  ok
Resetting WAL archives                               ok
Setting frozenxid and minmxid counters in new cluster ok
Restoring global objects in the new cluster           ok
Restoring database schemas in the new cluster         ok

Copying user relation files                           ok

Setting next OID for new cluster                      ok
Sync data directory to disk                           ok
Creating script to analyze new cluster                ok
Creating script to delete old cluster                 ok
Checking for extension updates                       notice
```

```
Your installation contains extensions that should be updated
with the ALTER EXTENSION command. The file
  update_extensions.sql
when executed by psql by the database superuser will update
these extensions.
```

```
Upgrade Complete
-----
```

```
Optimizer statistics are not transferred by pg_upgrade so,
once you start the new server, consider running:
  ./analyze_new_cluster.sh
```

```
Running this script will delete the old cluster's data files:
  ./delete_old_cluster.sh
```

4. Przełączamy się z powrotem na standardowego użytkownika

```
exit
```

5. Jeżeli nie wyświetlił się komunikat o pliku update_extensions.sql (komunikat z punktu 3, zaznaczony na czerwono) to pomijamy ten krok.

Wykonujemy wygenerowany SQL na bazie nowego postgresa kolejno poniższymi komendami:

```
sudo su postgres
```

```
psql -h 127.0.0.1 -p <port> -d <nazwa_usera_bazy>
```

np.

```
psql -h 127.0.0.1 -p 5432 -d softmedica
```

```
ALTER EXTENSION "hstore" UPDATE;
```

```
\q
```

```
exit
```

6. Edytujemy konfigurację nowego postgresa i zmieniamy na chwilę port na 5432 (dla starego postgresa nadajemy jakiś inny):

```
sudo nano /etc/postgresql/12/main/postgresql.conf
```

7. Uruchamiamy usługę postgresa:

```
sudo systemctl start postgresql
```

8. Przełączamy się na użytkownika postgres i przechodzimy do katalogu /tmp

```
sudo su postgres  
cd /tmp
```

9. Uruchamiamy wygenerowany skrypt

```
./analize_new_cluster.sh
```

Po poprawnym wykonaniu powinny być następujące komunikaty:

```
This script will generate minimal optimizer statistics rapidly  
so your system is usable, and then gather statistics twice more  
with increasing accuracy. When it is done, your system will  
have the default level of optimizer statistics.  
  
If you have used ALTER TABLE to modify the statistics target for  
any tables, you might want to remove them and restore them after  
running this script because they will delay fast statistics generation.  
  
If you would like default statistics as quickly as possible, cancel  
this script and run:  
    "C:/Program Files/Softmedica - Medycyna Pracy/PostgreSQL/12.5/bin/vacuumdb" -U softmedica --all --analyze-only  
  
Password:  
vacuumdb: processing database "postgres": Generating minimal optimizer statistics (1 target)  
vacuumdb: processing database "softmedica": Generating minimal optimizer statistics (1 target)  
vacuumdb: processing database "template1": Generating minimal optimizer statistics (1 target)  
vacuumdb: processing database "postgres": Generating medium optimizer statistics (10 targets)  
vacuumdb: processing database "softmedica": Generating medium optimizer statistics (10 targets)  
vacuumdb: processing database "template1": Generating medium optimizer statistics (10 targets)  
vacuumdb: processing database "postgres": Generating default (full) optimizer statistics  
vacuumdb: processing database "softmedica": Generating default (full) optimizer statistics  
vacuumdb: processing database "template1": Generating default (full) optimizer statistics  
  
Done
```

10. Wracamy na standardowego użytkownika

```
exit
```

11. Zmieniamy ustawienia nowego postgresa na docelowe (takie jakie były na starym funkcjonującym postgresie) w plikach konfiguracyjnych (zmiana wymaga restartu usługi)

```
sudo nano /etc/postgresql/12/main/postgresql.conf
```

Struktura pliku powinna wyglądać w podobny sposób:

```
-----  
#  
# CONNECTIONS AND AUTHENTICATION  
#  
# - Connection Settings -  
  
listen_addresses = '*'           # what IP address(es) to listen on;  
                                # comma-separated list of addresses;  
                                # defaults to 'localhost'; use '*' for all  
                                # (change requires restart)  
port = <docelowy_port>         # (change requires restart)  
max_connections = 300          # (change requires restart)
```

```
sudo nano /etc/postgresql/12/main/pg_hba.conf
```

Struktura pliku powinna wyglądać w podobny sposób:

- md5 - autoryzacja łączenia się z bazą, wymaga wpisania poprawnego hasła
- trust - można wpisać obojętnie jakie hasło

```
-----  
# 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    md5  
host      all         all         0.0.0.0/0       md5  
# IPv6 local connections:  
host      all         all         ::1/128         md5
```

Usuwanie starego klastra

Po pomyślnym przeniesieniu mamy możliwość usunięcia starego klastra wykonując po kolei poniższe komendy:

```
sudo su postgres
```

```
cd /tmp
```

```
./delete_old_cluster.sh
```

```
sudo apt-get remove postgresql-<wersja_postgres> postgresql-server-dev-<wersja_postgres>
```

np.

```
sudo apt-get remove postgresql-10 postgresql-server-dev-10
```

```
sudo rm -rf /etc/postgresql/10/
```