Zmiana katalogu danych działającej instancji

□□ Typ: procedura **□□ Dotyczy: PostgreSQL 9.6+, Linux**

□ Czas: ~20 min + czas kopiowania danych

Efekt: ta sama instancja, z tymi samymi bazami i rolami, działająca dalej na nowym wolumenie — bez reinstalacji i bez utraty danych.

□□ Skróty użyte na tej stronie (3)

Skrót	Co znaczy
PGDG	PostgreSQL Global Development Group: oficjalne repozytorium pakietów PostgreSQL dla systemów Linux
SELinux	Security-Enhanced Linux: dodatkowa warstwa uprawnień w jądrze RHEL
SQL	Structured Query Language: język zapytań do relacyjnych baz danych

Zanim zaczniemy

> **□□ Z ŻYCIA WZIĘTE**
>
> Baza aplikacji Oli na pg01 rośnie szybciej, niż zakładano, i wolumen z danymi kończy się w środę rano. Przenosimy `PGDATA` na nowy wolumen zgodny z konwencją `vgPostgres`/`lvPostgres`, kopiując dane poleceniem `cp` zamiast `mv`, żeby stary katalog został nietknięty do czasu pełnej weryfikacji. Po starcie na nowej ścieżce Ola sprawdza swój panel i potwierdza, że wszystko działa jak wcześniej.

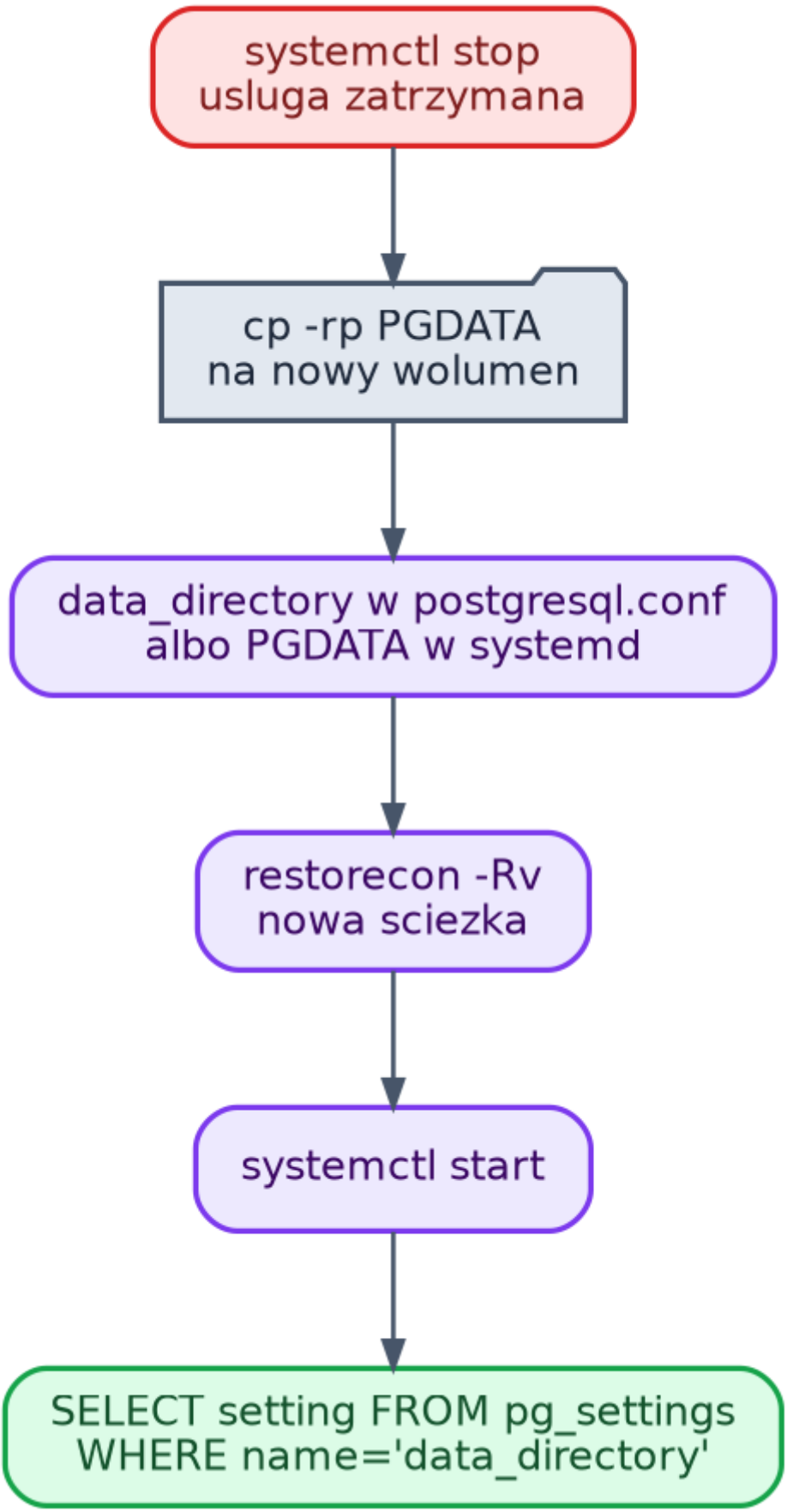
Przenosimy `PGDATA` w całości na nowy wolumen — najczęściej dlatego, że baza wyrosła z miejsca przydzielonego jej przy pierwszej instalacji, albo dlatego, że dane mają trafić na osobny, szybszy wolumen logiczny zgodny z konwencją `vgPostgres`/`lvPostgres`. Cała procedura wymaga zatrzymania instancji na czas kopiowania — to jest okno serwisowe, nie operacja na żywo.

Wymaganie	Jak sprawdzić

Nowy wolumen zamontowany i z wystarczającą ilością miejsca	`df -h SciezkaDoPliku`
Rozmiar bieżącego `PGDATA`	`du -sh SciezkaDoPliku` na starej lokalizacji
Okno serwisowe uzgodnione z użytkownikami bazy	—
Świeży backup przed operacją	strona pgBackRest: podstawowe komendy albo pg_dump, pg_dumpall, pg_restore

Uważaj: dopóki nie usuniemy starego katalogu danych, mamy pełną drogę powrotu — procedura poniżej celowo kopiuje dane zamiast je przenosić (`mv`), żeby oryginał został nietknięty aż do potwierdzonej weryfikacji.

Przebieg


systemctl stop
usluga zatrzymana
cp -rp PGDATA
na nowy wolumen
data_directory w postgresql.conf
albo PGDATA w systemd
restorecon -Rv
nowa sciezka
systemctl start
SELECT setting FROM pg_settings
WHERE name='data_directory'

Sześć etapów przenoszenia katalogu danych działającej instancji na nowy wolumen (kliknięcie otwiera pełny rozmiar)

1. Zatrzymanie usługi

```
sudo systemctl stop postgresql
```

Na instalacji z pakietu PGDG, gdzie jednostka nosi nazwę wersji, zatrzymujemy konkretną jednostkę:

```
sudo systemctl stop postgresql-15.service
```

Weryfikujemy, że proces `postgres` faktycznie zniknął z listy procesów, zanim przejdziemy dalej — kopiowanie plików pod działającą instancją kończy się uszkodzonymi danymi:

```
ps -ef | grep postgres
```

2. Skopiowanie katalogu danych na nowy wolumen

Kopiujemy z zachowaniem uprawnień i znaczników czasu (`-p`), rekurencyjnie:

```
sudo cp -rp SciezkaDoPliku SciezkaDoNowegoKatalogu
```

Przykład przeniesienia z domyślnej ścieżki dystrybucji na nowy wolumen:

```
sudo cp -rp /var/lib/pgsql/15/data /postgres/15/
```

Dla dużych baz kopiowanie samym `cp` bywa zauważalnie wolniejsze niż `rsync -a` — mechanizm jest ten sam, ale `rsync` lepiej radzi sobie z wieloma małymi plikami, jakich w `PGDATA` (zwłaszcza w `base/`) jest zwykle bardzo dużo.

3. Wskazanie nowej lokalizacji instancji

Są dwa równoważne sposoby wskazania nowego `PGDATA` — wybieramy jeden, nie oba naraz.

Sposób pierwszy, przez parametr w `postgresql.conf` — otwieramy plik na starej ścieżce (ten w nowej lokalizacji jeszcze nie jest czytany) i zmieniamy wpis:

```
data_directory = 'SciezkaDoPliku'
```

na:

```
data_directory = 'SciezkaDoNowegoKatalogu'
```

Przykład zmiany:

```
data_directory = '/postgres/15/data'
```

Uważaj: ten wpis trzeba zmienić w pliku `postgresql.conf`, który usługa wczyta przy starcie — jeśli usługa startuje z parametrem `-D` wskazującym starą ścieżkę, silnik i tak przeczyta stary plik konfiguracyjny ze starym wpisem `data_directory`, a nowy katalog zostanie zignorowany.

Sposób drugi, przez zmienną środowiskową usługi systemd — zamiast edytować `postgresql.conf`, wskazujemy nową ścieżkę bezpośrednio jednostce:

```
sudo systemctl edit postgresql-15.service
```

```
[Service]
Environment=PGDATA=SciezkaDoNowegoKatalogu
```

Ten wariant jest wygodniejszy, gdy `PGDATA` na starej ścieżce ma pozostać nietknięty do czasu pełnej weryfikacji nowej lokalizacji — nic w starym katalogu się wtedy nie zmienia.

4. Odtworzenie kontekstu SELinux na nowej ścieżce

Skopiowane pliki dziedziczą kontekst SELinux ze źródła, który na nowym punkcie montowania bywa niepoprawny:

```
sudo restorecon -Rv SciezkaDoNowegoKatalogu
```

Jeśli nowa ścieżka nigdy wcześniej nie była używana pod PostgreSQL, kontekst trzeba nadać jawnie, tak jak przy pierwszej instalacji:

```
sudo chcon -Rt postgresql_db_t SciezkaDoNowegoKatalogu
```

5. Przeładowanie systemd i uruchomienie usługi

```
sudo systemctl daemon-reload
```

```
sudo systemctl start postgresql-15.service
```

Weryfikacja

Logujemy się do instancji i sprawdzamy, że katalog danych rzeczywiście wskazuje na nową lokalizację:

```
SELECT setting FROM pg_settings WHERE name = 'data_directory';
```

Przykład:

```
        setting
------------------------
 /postgres/15/data
(1 row)
```

Alternatywą z poziomu bash, bez logowania do bazy, jest odczyt skonfigurowanej domyślnie ścieżki z opcji kompilacji binarki:

```
pg_config --configure | grep -- '--datadir'
```

Uważaj: `pg_config --configure` pokazuje domyślną ścieżkę wbudowaną w binarkę przy kompilacji, nie bieżącą wartość `data_directory` uruchomionej instancji — jeśli katalog wskazaliśmy przez `postgresql.conf` albo zmienną `PGDATA`, to zapytanie SQL wyżej jest jedynym pewnym źródłem prawdy o tym, gdzie instancja faktycznie czyta dane.

Sprawdzamy też, że dane są kompletne — porównujemy liczbę baz i wielkość katalogu ze stanem sprzed przeniesienia:

```
\l+
```

Wycofanie zmian

Dopóki stary katalog danych nie został usunięty, wycofanie polega na przywróceniu poprzedniej konfiguracji:

```
sudo systemctl stop postgresql-15.service
```

Cofamy wpis `data_directory` w `postgresql.conf` do starej ścieżki albo usuwamy nadpisanie `PGDATA`:

```
sudo systemctl revert postgresql-15.service
```

```
sudo systemctl start postgresql-15.service
```

Stary katalog danych usuwamy dopiero po kilku dniach pracy na nowej lokalizacji, nie od razu po pierwszym udanym starcie.

Gdy coś nie zadziała

Objaw	Przyczyna	Naprawa
Usługa startuje, ale `data_directory` wciąż pokazuje starą ścieżkę	zmieniony `postgresql.conf` leży w katalogu, którego usługa nie czyta	sprawdzamy `systemctl cat postgresql-15.service` i realną ścieżkę `-D` albo `PGDATA`
`FATAL: data directory ... has invalid permissions`	uprawnienia po `cp` nie są `0700` albo właściciel to nie `postgres`	`chmod 0700 SciezkaDoNowegoKatalogu`, `chown -R postgres:postgres`
`FATAL: lock file "postmaster.pid" already exists`	plik blokady skopiowany razem z resztą danych z katalogu, który wciąż uznawany jest za zajęty	usuwamy `postmaster.pid` w nowym katalogu tylko po pewności, że stary proces faktycznie już nie działa
SELinux blokuje start na nowej ścieżce	brak kontekstu `postgresql_db_t` po skopiowaniu na nowy wolumen	`restorecon -Rv` albo `chcon -Rt postgresql_db_t`
Baza rusza, ale brakuje w niej najnowszych danych	kopiowanie wykonane przed pełnym zatrzymaniem usługi	zatrzymujemy usługę, potwierdzamy brak procesu `postgres`, kopiujemy ponownie

Powiązane: Architektura PostgreSQL: klaster, baza, schemat, procesy · Instalacja PostgreSQL z własnym katalogiem danych · pgBackRest: podstawowe komendy · Instalacja Rocky Linux z własnym układem LVM

Źródła: dokumentacja PostgreSQL, rozdział „Database File Layout" · `man 8 restorecon` · `man 8 chcon`

Revision #43
Created 2026-09-24 23:07:59 UTC by Tomasz Konieczny
Updated 2026-10-26 14:12:51 UTC by Tomasz Konieczny