Instalacja PostgreSQL z własnym katalogiem danych

Typ: procedura **Dotyczy: RHEL / Rocky Linux / CentOS, repozytorium PGDG**

Czas: ~30 min

Efekt: instancja PostgreSQL wybranej wersji głównej, zainstalowana i uruchomiona z katalogiem danych w lokalizacji wskazanej przez nas, a nie w domyślnej ścieżce dystrybucji.

Skróty użyte na tej stronie (4)

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

Zanim zaczniemy

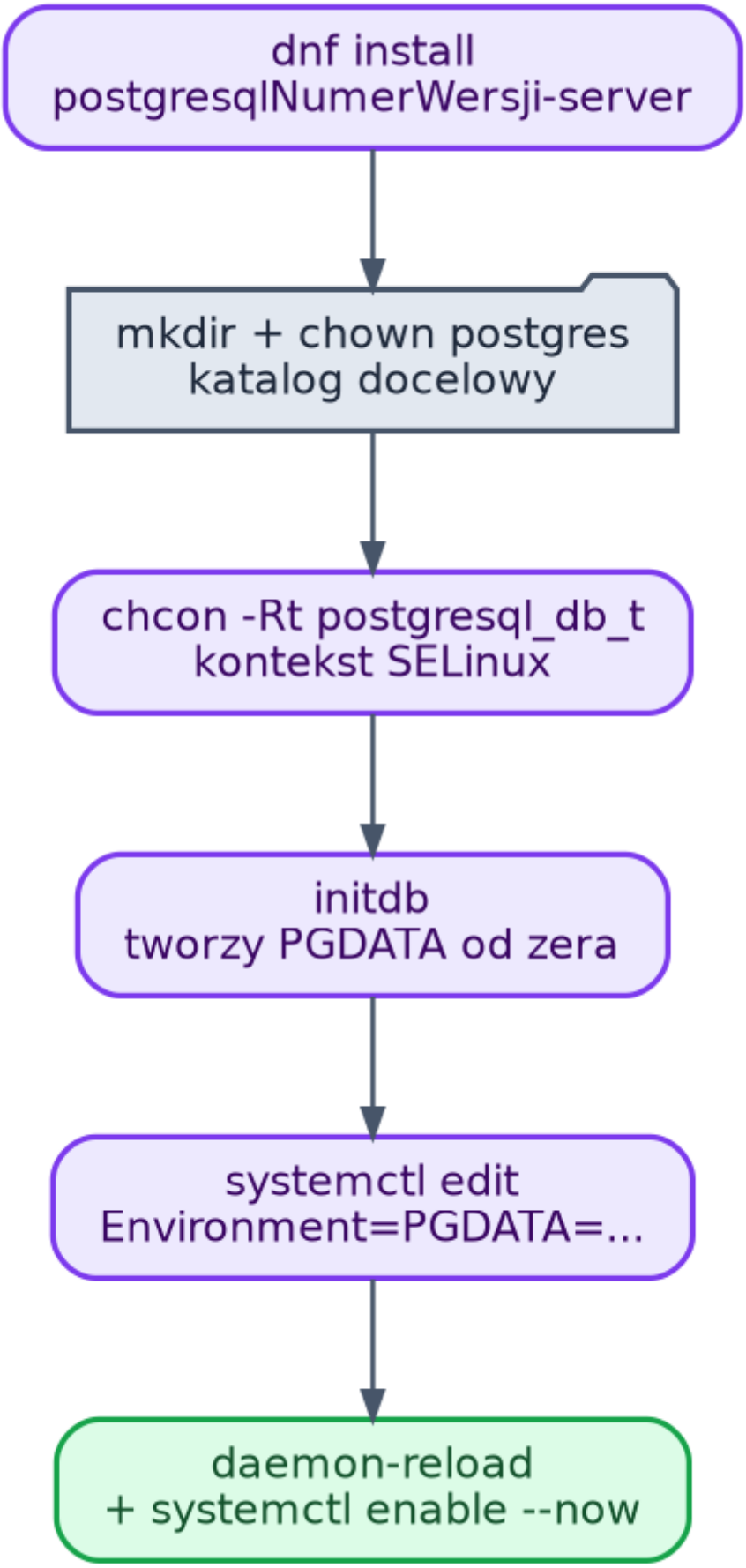
> **Z ŻYCIA WZIĘTE**
>
> Piątek, 15:40, Tomek stawia na pg01 drugą instancję PostgreSQL do testów panelu Oli, kopiując komendy ze starej notatki bez zmiany portu. `initdb` kończy się bez błędu, ale usługa nie startuje, bo port 5432 zajęty jest już przez instancję produkcyjną. Uczy się sprawdzać `SHOW port`, zanim uruchomi drugi klaster obok pierwszego.

Domyślna instalacja z repozytorium dystrybucji zakłada katalog danych w standardowej ścieżce systemowej, zwykle na tym samym wolumenie co system operacyjny. Jeśli baza ma rosnąć, dane bazy powinny leżeć na osobnym wolumenie logicznym — konwencja `vgPostgres` / `lvPostgres` opisana na stronie Instalacja Rocky Linux z własnym układem LVM zakłada dokładnie taki podział. Ta procedura instaluje silnik tak, żeby `PGDATA` od razu wskazywał na docelowy katalog, zamiast przenosić dane po fakcie.

Wymaganie	Jak sprawdzić
Wolumen pod dane bazy zamontowany i pisalny	`df -h SciezkaDoPliku`
Wybrana wersja główna PostgreSQL	`dnf module list postgresql` albo strona pobierania PGDG
Dostęp `sudo` / root	`sudo -v`
SELinux w trybie `enforcing` czy `permissive`	`getenforce`

Uważaj: wybór wersji głównej jest decyzją na lata — przejście na inną wersję główną później to migracja opisana na stronie Upgrade silnika PostgreSQL, nie prosta podmiana pakietu.

Przebieg


dnf install
postgresqlNumerWersji-server
mkdir + chown postgres
katalog docelowy
chcon -Rt postgresql_db_t
kontekst SELinux
initdb
tworzy PGDATA od zera
systemctl edit
Environment=PGDATA=...
daemon-reload
+ systemctl enable --now

Sześć etapów instalacji PostgreSQL z własnym katalogiem danych, od instalacji pakietu do uruchomienia usługi (kliknięcie otwiera pełny rozmiar)

1. Instalacja pakietu serwera

Repozytorium PGDG (postgresql.org) trzyma równolegle kilka wersji głównych pod osobnymi nazwami pakietów, co pozwala zainstalować dokładnie tę wersję, jakiej potrzebujemy, niezależnie od tego, co akurat oferuje repozytorium systemowe:

```
sudo dnf install -y postgresqlNumerWersji-server
```

Przykład dla wersji 15:

```
sudo dnf install -y postgresql15-server
```

Jeśli zamiast repozytorium PGDG korzystamy z modułu dostarczanego przez samą dystrybucję, wersję główną wybieramy włączeniem modułu `dnf`, zanim zainstalujemy pakiet:

```
sudo dnf module enable postgresql:NumerWersji
sudo dnf install -y postgresql-server
```

2. Utworzenie katalogu danych i nadanie uprawnień

Katalog docelowy musi istnieć i należeć do konta systemowego `postgres`, zanim uruchomimy inicjalizację:

```
sudo mkdir -p SciezkaDoPliku
```

```
sudo chown -R postgres:postgres SciezkaDoPliku
```

Przykład dla katalogu `/postgres/15/data`:

```
sudo mkdir -p /postgres/15/data
sudo chown -R postgres:postgres /postgres
```

3. Nadanie kontekstu SELinux katalogowi danych

Katalog spoza domyślnej ścieżki (`/var/lib/pgsql`) nie ma automatycznie właściwego kontekstu SELinux — bez tego kroku usługa odmówi startu na systemie z `enforcing`:

```
sudo chcon -Rt postgresql_db_t SciezkaDoPliku
```

Kontekst nadany w ten sposób nie przetrwa relabelingu całego systemu plików — trwały wpis dodajemy regułą `semanage fcontext`, jeśli katalog ma zostać na stałe poza `/var/lib/pgsql`.

4. Inicjalizacja bazy (initdb)

> **Pojęcie: initdb.** Narzędzie, które tworzy od zera zawartość `PGDATA` opisaną na stronie Architektura PostgreSQL: klaster, baza, schemat, procesy — katalogi `base`, `global`, `pg_wal`, domyślne bazy `template0` / `template1` / `postgres` i pliki konfiguracyjne z wartościami startowymi. Bez `initdb` katalog danych jest po prostu pustym katalogiem, nieużytecznym dla żadnej instancji.

Pakiety PGDG dostarczają gotowy skrypt inicjalizujący pod nazwą wersji:

```
sudo /usr/pgsql-NumerWersji/bin/postgresqlNumerWersji-setup initdb
```

Przykład dla wersji 15:

```
sudo /usr/pgsql-15/bin/postgresql-15-setup initdb
```

Na systemie z pakietem modułowym dystrybucji odpowiednikiem jest:

```
sudo postgresql-setup --initdb
```

Uważaj: oba warianty inicjalizują katalog domyślny (`/var/lib/pgsql/NumerWersji/data`), a nie ten utworzony w kroku 2 — ścieżkę docelową wskazujemy dopiero w konfiguracji usługi w następnym kroku, zanim pierwszy raz uruchomimy `initdb` na docelowej ścieżce, albo wskazujemy ją bezpośrednio zmienną `PGDATA` przy wywołaniu skryptu.

5. Wskazanie katalogu danych w usłudze systemd

Jednostka `systemd` instalowana przez pakiet PGDG czyta katalog danych ze zmiennej środowiskowej `PGDATA`, ustawianej w nadpisaniu (override), nie w samym pliku jednostki:

```
sudo systemctl edit postgresqlNumerWersji.service
```

Komenda otwiera edytor z pustym plikiem nadpisania. Wpisujemy w nim:

```
[Service]
Environment=PGDATA=SciezkaDoPliku
```

Przykład zawartości dla wersji 15 i katalogu `/postgres/15/data`:

```
[Service]
Environment=PGDATA=/postgres/15/data
```

Plik zapisuje się automatycznie w `/etc/systemd/system/postgresql-NumerWersji.service.d/override.conf`.

6. Przeładowanie systemd i uruchomienie usługi

```
sudo systemctl daemon-reload
```

```
sudo systemctl enable --now postgresqlNumerWersji.service
```

Przykład dla wersji 15:

```
sudo systemctl enable --now postgresql-15.service
```

7. Wskazanie PGDATA w profilu użytkownika postgres

Krok wygodnościowy, nie wymagany do działania usługi — dzięki niemu narzędzia takie jak `pg_ctl` czy `pg_config` uruchamiane ręcznie z konta `postgres` od razu widzą właściwy katalog bez podawania flagi `-D`:

```
sudo su - postgres -c "echo 'export PGDATA=SciezkaDoPliku' >> ~/.bash_profile"
```

Weryfikacja

Logujemy się do instancji lokalnym kontem systemowym `postgres`, co przy domyślnej regule `peer` w `pg_hba.conf` nie wymaga hasła:

```
sudo -u postgres psql
```

Przykład:

```
psql (15.5)
Type "help" for help.


postgres=#
```

Sprawdzamy z poziomu SQL, że katalog danych rzeczywiście wskazuje na naszą lokalizację:

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

Przykład:

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

Sprawdzamy też stan samej usługi:

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

Wycofanie zmian

Jeśli instalacja poszła w złą stronę i chcemy zacząć od nowa na tym samym serwerze, zatrzymujemy usługę i czyścimy katalog danych, zanim powtórzymy `initdb`:

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

```
sudo rm -rf SciezkaDoPliku/*
```

Nadpisanie systemd usuwamy komendą `systemctl revert`, jeśli chcemy wrócić do domyślnej ścieżki katalogu danych:

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

Gdy coś nie zadziała

Objaw	Przyczyna	Naprawa
Usługa nie startuje, `journalctl` pokazuje `Permission denied` na katalogu danych	katalog nie należy do `postgres` albo ma złe uprawnienia uniksowe	`chown -R postgres:postgres` i uprawnienia `0700` na katalogu danych
Usługa nie startuje, w logu SELinux `AVC denial`	katalog spoza `/var/lib/pgsql` bez kontekstu `postgresql_db_t`	`sudo chcon -Rt postgresql_db_t SciezkaDoPliku`, docelowo reguła `semanage fcontext -a`
`initdb` kończy się błędem `directory ... exists but is not empty`	katalog docelowy zawiera pozostałości po poprzedniej próbie	czyścimy katalog albo wskazujemy nowy, pusty

Objaw	Przyczyna	Naprawa
Po `systemctl edit` usługa wciąż startuje w starej lokalizacji	nadpisanie zapisane dla innej nazwy jednostki niż faktycznie używana (`postgresql.service` zamiast `postgresql-15.service`)	sprawdzamy `systemctl list-units
`psql: could not connect to server: No such file or directory`	usługa nie wystartowała albo gniazdo uniksowe w innej ścieżce niż oczekuje klient	`systemctl status` , `journalctl -u postgresqlNumerWersji.service`

Powiązane: Architektura PostgreSQL: klaster, baza, schemat, procesy · Zmiana katalogu danych działającej instancji · Instalacja Rocky Linux z własnym układem LVM · Upgrade silnika PostgreSQL

Źródła: dokumentacja PostgreSQL, rozdział „Installation from Binaries" · repozytorium PGDG, `download.postgresql.org` · `man 8 semanage-fcontext`

Revision #43
Created 2026-09-24 23:07:58 UTC by Tomasz Konieczny
Updated 2026-10-26 14:12:49 UTC by Tomasz Konieczny