
Kopie zapasowe i odzyskiwanie danych

Wydanie 1.0

Miłosz Śmieja

12 cze 2025

Contents

1	Kopie zapasowe i odzyskiwanie danych w PostgreSQL	1
1.1	Wprowadzenie	2
1.2	Mechanizmy wbudowane do tworzenia kopii zapasowych całego systemu PostgreSQL	2
1.3	Mechanizmy wbudowane do tworzenia kopii zapasowych poszczególnych baz danych	3
1.4	Odzyskiwanie usuniętych lub uszkodzonych danych	5
1.5	Dedykowane oprogramowanie i skrypty zewnętrzne do automatyzacji	6
1.6	Podsumowanie	8
2	Indeksy i tabele	9

Kopie zapasowe i odzyskiwanie danych w PostgreSQL

Autor

Miłosz Śmieja

Wersja

1.0

Data

12 czerwca 2025

Spis treści

- *Wprowadzenie*
- *Mechanizmy wbudowane do tworzenia kopii zapasowych całego systemu PostgreSQL*
 - *pg_basebackup*
 - *Continuous Archiving (Point-in-Time Recovery)*
 - *Streaming Replication*
 - *File System Level Backup*
- *Mechanizmy wbudowane do tworzenia kopii zapasowych poszczególnych baz danych*
 - *pg_dump*
 - *pg_dumpall*
 - *COPY command*
 - *pg_dump z opcjami selektywnymi*
- *Odzyskiwanie usuniętych lub uszkodzonych danych*
 - *Odzyskiwanie z kopii logicznych*
 - *Point-in-Time Recovery (PITR)*

- *Odzyskiwanie tabel z tablespaces*
- *Transaction log replay*
- *Odzyskiwanie na poziomie klastra*
- *Dedykowane oprogramowanie i skrypty zewnętrzne do automatyzacji*
 - *pgBackRest*
 - *Barman (Backup and Recovery Manager)*
 - *WAL-E i WAL-G*
 - *Skrypty shell i cron jobs*
 - *Narzędzia automatyzacji infrastruktury*
 - *Monitoring i alertowanie*
- *Podsumowanie*
 - *Kluczowe wnioski*
 - *Najważniejsze zalecenia*

1.1 Wprowadzenie

System zarządzania bazą danych PostgreSQL oferuje kompleksowy zestaw narzędzi i mechanizmów służących do tworzenia kopii zapasowych oraz odzyskiwania danych. Skuteczne zarządzanie kopiami zapasowymi stanowi fundament bezpieczeństwa danych i ciągłości działania systemów bazodanowych.

PostgreSQL dostarcza zarówno mechanizmy wbudowane, jak i możliwość integracji z zewnętrznymi narzędziami automatyzacji.

1.2 Mechanizmy wbudowane do tworzenia kopii zapasowych całego systemu PostgreSQL

PostgreSQL oferuje kilka mechanizmów tworzenia kopii zapasowych na poziomie całego systemu, które zapewniają kompleksową ochronę wszystkich baz danych w klastrze.

1.2.1 pg_basebackup

pg_basebackup stanowi podstawowe narzędzie do tworzenia fizycznych kopii zapasowych całego klastra PostgreSQL.

Kluczowe cechy:

- Działa w trybie online - możliwość wykonywania kopii zapasowych bez zatrzymywania działania serwera
- Tworzy dokładną kopię wszystkich plików danych
- Zawiera pliki konfiguracyjne, dzienniki transakcji oraz wszystkie bazy danych w klastrze

1.2.2 Continuous Archiving (Point-in-Time Recovery)

Continuous Archiving reprezentuje zaawansowany mechanizm tworzenia ciągłych kopii zapasowych poprzez archiwizację dzienników WAL (Write-Ahead Logging).

Zalety:

- Umożliwia odtworzenie stanu bazy danych w dowolnym momencie czasowym
- Szczególnie wartościowe w środowiskach produkcyjnych wymagających minimalnej utraty danych
- Zapewnia wysoką granularność odzyskiwania danych

1.2.3 Streaming Replication

Streaming Replication może służyć jako mechanizm kopii zapasowych poprzez utrzymywanie synchronicznych lub asynchronicznych replik głównej bazy danych.

Funkcjonalności:

- Repliki funkcjonują jako kopie zapasowe w czasie rzeczywistym
- Oferuje możliwość szybkiego przełączenia w przypadku awarii systemu głównego
- Wspiera zarówno tryb synchroniczny, jak i asynchroniczny

1.2.4 File System Level Backup

File System Level Backup polega na tworzeniu kopii zapasowych na poziomie systemu plików.

Wymagania:

- Zatrzymanie serwera PostgreSQL lub zapewnienie spójności
- Wykorzystanie mechanizmów snapshot systemu plików:
 - LVM snapshots
 - ZFS snapshots

1.3 Mechanizmy wbudowane do tworzenia kopii zapasowych poszczególnych baz danych

PostgreSQL dostarcza precyzyjne narzędzia umożliwiające tworzenie kopii zapasowych pojedynczych baz danych lub ich wybranych elementów.

1.3.1 pg_dump

pg_dump stanowi najczęściej wykorzystywane narzędzie do tworzenia logicznych kopii zapasowych pojedynczych baz danych.

Charakterystyka:

- Tworzy skrypt SQL zawierający wszystkie polecenia niezbędne do odtworzenia struktury bazy danych oraz jej danych
- Oferuje liczne opcje konfiguracji:

- Możliwość wyboru formatu wyjściowego
- Filtrowanie obiektów
- Kontrola nad poziomem szczegółowości kopii zapasowej

1.3.2 pg_dumpall

pg_dumpall rozszerza funkcjonalność **pg_dump** o możliwość tworzenia kopii zapasowych wszystkich baz danych w klastrze.

Dodatkowe funkcje:

- Backup obiektów globalnych:
 - Role użytkowników
 - Tablespace
 - Ustawienia konfiguracyjne na poziomie klastra

1.3.3 COPY command

COPY command umożliwia eksport danych z poszczególnych tabel do plików w różnych formatach.

Obsługiwane formaty:

- CSV
- Text
- Binary

Zastosowania:

- Tworzenie selektywnych kopii zapasowych dużych tabel
- Migracje danych

1.3.4 pg_dump z opcjami selektywnymi

pg_dump z opcjami selektywnymi pozwala na tworzenie kopii zapasowych wybranych obiektów bazy danych.

Możliwości filtrowania:

- Konkretne tabele
- Schematy
- Sekwencje

Funkcjonalność ta jest nieoceniona w scenariuszach wymagających granularnej kontroli nad procesem tworzenia kopii zapasowych.

1.4 Odzyskiwanie usuniętych lub uszkodzonych danych

PostgreSQL oferuje różnorodne mechanizmy odzyskiwania danych w zależności od rodzaju i zakresu uszkodzeń.

1.4.1 Odzyskiwanie z kopii logicznych

Odzyskiwanie z kopii logicznych wykonanych przy użyciu `pg_dump` realizowane jest poprzez `psql` lub `pg_restore`.

Proces odzyskiwania:

- Wykonanie skryptów SQL
- Przywrócenie plików dump w odpowiednim formacie

Zaawansowane opcje `pg_restore`:

- Selektywne przywracanie obiektów
- Równoległe przetwarzanie
- Kontrola nad kolejnością przywracania

1.4.2 Point-in-Time Recovery (PITR)

Point-in-Time Recovery (PITR) umożliwia przywrócenie bazy danych do konkretnego momentu w czasie.

Wykorzystywane komponenty:

- Kombinacja kopii bazowej
- Archiwalne dzienniki WAL

Zastosowania:

- Cofnięcie zmian do momentu poprzedzającego wystąpienie błędu
- Odzyskiwanie po uszkodzeniu danych

Informacja: PITR jest szczególnie wartościowy w przypadkach, gdy konieczne jest cofnięcie zmian do momentu poprzedzającego wystąpienie błędu lub uszkodzenia.

1.4.3 Odzyskiwanie tabel z tablespaces

Odzyskiwanie tabel z tablespaces może wymagać specjalnych procedur w przypadku uszkodzenia przestrzeni tabel.

Możliwości PostgreSQL:

- Odtworzenie tablespaces
- Przeniesienie tabel między różnymi lokalizacjami
- Odzyskiwanie danych nawet w przypadku częściowego uszkodzenia systemu plików

1.4.4 Transaction log replay

Transaction log replay wykorzystuje dzienniki WAL do odtworzenia zmian wprowadzonych po utworzeniu kopii zapasowej.

Charakterystyka:

- Automatycznie wykorzystywany podczas standardowych procedur odzyskiwania
- Możliwość ręcznej kontroli w szczególnych sytuacjach

1.4.5 Odzyskiwanie na poziomie klastra

Odzyskiwanie na poziomie klastra przy wykorzystaniu `pg_basebackup` wymaga przywrócenia wszystkich plików klastra oraz odpowiedniej konfiguracji parametrów `recovery`.

Zakres procesu:

- Odtworzenie całego środowiska PostgreSQL
- Konfiguracja ról i uprawnień
- Przywrócenie ustawień systemowych

1.5 Dedykowane oprogramowanie i skrypty zewnętrzne do automatyzacji

Automatyzacja procesów tworzenia kopii zapasowych stanowi kluczowy element profesjonalnego zarządzania bazami danych PostgreSQL.

1.5.1 pgBackRest

pgBackRest reprezentuje kompleksowe rozwiązanie do zarządzania kopiami zapasowymi PostgreSQL.

Zaawansowane funkcje:

- Incremental i differential backups
- Kompresja danych
- Szyfrowanie
- Weryfikacja integralności kopii
- Możliwość przechowywania kopii w chmurze
- Automatyzacja procesów zarządzania kopiami zapasowymi
- Uprozczone procedury odzyskiwania

Ważne: `pgBackRest` automatyzuje wiele procesów związanych z zarządzaniem kopiami zapasowymi i znacznie upraszcza procedury odzyskiwania.

1.5.2 Barman (Backup and Recovery Manager)

Barman stanowi dedykowane narzędzie stworzone przez 2ndQuadrant do zarządzania kopiami zapasowymi PostgreSQL w środowiskach enterprise.

Kluczowe funkcjonalności:

- Centralne zarządzanie kopiami zapasowymi wielu serwerów PostgreSQL
- Monitoring procesów backup
- Automatyczne testowanie procedur recovery
- Integracja z narzędziami monitorowania

1.5.3 WAL-E i WAL-G

WAL-E i WAL-G specjalizują się w archiwizacji dzienników WAL w środowiskach chmurowych.

Oferowane funkcje:

- Efektywna kompresja
- Szyfrowanie danych
- Przechowywanie kopii zapasowych w serwisach chmurowych:
 - Amazon S3
 - Google Cloud Storage
 - Azure Blob Storage

1.5.4 Skrypty shell i cron jobs

Skrypty shell i cron jobs stanowią tradycyjne podejście do automatyzacji kopii zapasowych.

Możliwości automatyzacji:

- Wykonywanie `pg_dump` i `pg_basebackup`
- Zarządzanie cyklem życia kopii zapasowych
- Rotacja i czyszczenie starych kopii

Wskazówka: Właściwie napisane skrypty mogą automatyzować wykonywanie `pg_dump`, `pg_basebackup` oraz zarządzanie cyklem życia kopii zapasowych, w tym rotację i czyszczenie starych kopii.

1.5.5 Narzędzia automatyzacji infrastruktury

Ansible, Puppet, Chef jako narzędzia automatyzacji infrastruktury mogą być wykorzystywane do zarządzania konfiguracją procesów backup na większą skalę.

Korzyści:

- Standaryzacja procedur backup w środowiskach wieloserwerowych
- Zapewnienie konsystencji konfiguracji
- Skalowalne zarządzanie infrastrukturą

1.5.6 Monitoring i alertowanie

Prometheus i Grafana w połączeniu z `postgres_exporter` umożliwiają monitoring procesów backup oraz alertowanie w przypadku niepowodzeń.

Zakres monitorowania:

- Śledzenie czasu wykonywania kopii
- Monitorowanie rozmiaru kopii zapasowych
- Wskaźnik sukcesu procesów backup
- Alertowanie w czasie rzeczywistym

1.6 Podsumowanie

Skuteczne zarządzanie kopiami zapasowymi w PostgreSQL wymaga kombinacji mechanizmów wbudowanych oraz zewnętrznych narzędzi automatyzacji. Wybór odpowiedniej strategii backup zależy od specyficznych wymagań organizacji, w tym:

- **RTO (Recovery Time Objective)** - maksymalny akceptowalny czas odzyskiwania
- **RPO (Recovery Point Objective)** - maksymalna akceptowalna utrata danych
- Dostępne zasoby
- Złożoność środowiska

1.6.1 Kluczowe wnioski

Mechanizmy wbudowane PostgreSQL, takie jak `pg_dump`, `pg_basebackup` czy PITR, oferują solidne podstawy dla większości scenariuszy backup i recovery.

W środowiskach produkcyjnych o wysokich wymaganiach dotyczących dostępności i niezawodności, integracja z dedykowanymi narzędziami takimi jak pgBackRest czy Barman staje się niezbędna.

1.6.2 Najważniejsze zalecenia

Ostrzeżenie: Kluczowym elementem każdej strategii backup jest regularne testowanie procedur odzyskiwania danych. Kopie zapasowe mają wartość tylko wtedy, gdy można z nich skutecznie odzyskać dane w sytuacji kryzysowej.

Kompleksowa strategia backup powinna obejmować:

1. Tworzenie kopii zapasowych
2. Regularne testy restore
3. Dokumentację procedur
4. Szkolenie personelu odpowiedzialnego za zarządzanie bazami danych

CHAPTER 2

Indeksy i tabele

- genindex
- modindex
- search