PostgreSQL: z Berkeley do każdej firmy

Poziom: podstawy **Dotyczy: PostgreSQL 9.6-17** **Czas: ~15 min**

Jedyny z czterech silników tej bazy wiedzy, który nie ma właściciela ani rachunku za licencję. Ta strona otwiera książkę „PostgreSQL": skąd wziął się ten silnik, jak trafił do Zielonej i co czeka na kolejnych stronach.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
BSD	Berkeley Software Distribution: rodzina Uniksów z uczelni Berkeley, inna gałąź niż Linux, stąd czasem inna składnia tych samych poleceń
DARPA	Defense Advanced Research Projects Agency: agencja badawcza amerykańskiej armii, jeden ze sponsorów projektu POSTGRES na Berkeley
SQL	Structured Query Language: język zapytań do relacyjnych baz danych

Skąd się wzięło

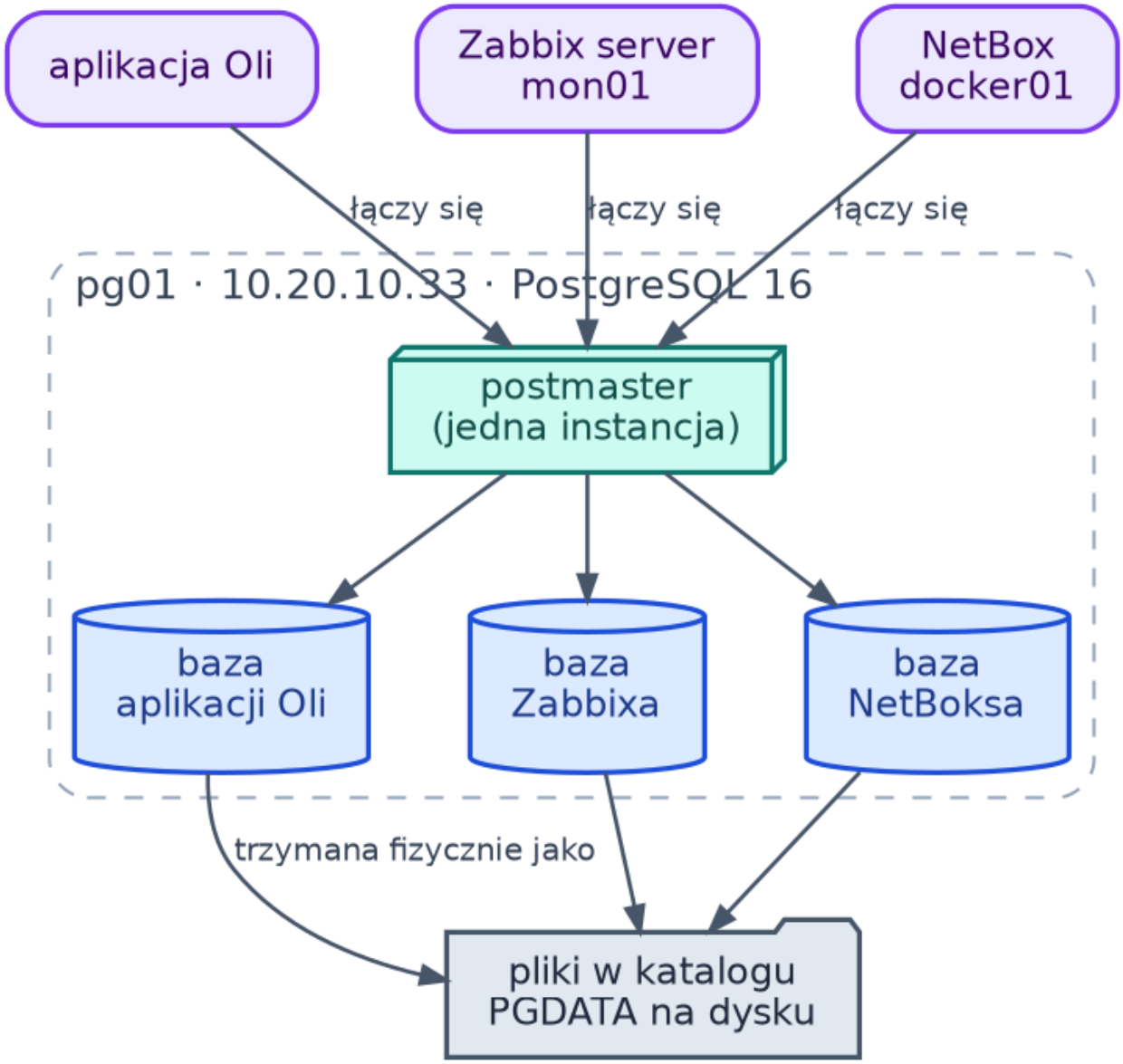
Zanim powstał PostgreSQL, na Uniwersytecie Kalifornijskim w Berkeley powstał Ingres, wczesny silnik relacyjny prowadzony przez Michaela Stonebrakera w latach siedemdziesiątych. Stonebraker skomercjalizował Ingres, sprzedał swoją firmę i wrócił na uczelnię z pomysłem na następcę: silnik, który zamiast trzymać się sztywno modelu relacyjnego, pozwoli programiście definiować własne typy danych i reguły. W 1986 roku ruszył na Berkeley projekt o nazwie POSTGRES, sfinansowany między innymi przez agencję badawczą DARPA. Nazwa mówiła wprost, czym ma być: tym, co po (post) Ingres.

Pierwsza działająca wersja POSTGRES trafiła do testów w 1987 roku, a projekt rozwijał się na uczelni przez kolejne sześć lat, doczekawszy się czterech kolejnych wydań. Silnik miał wtedy własny język zapytań, PostQUEL, niezgodny z SQL-em, który w tym czasie stawał się standardem branży. W 1994 roku dwóch studentów Berkeley, Andrew Yu i Jolly Chen, dopisało do POSTGRES interpreter SQL i wydało wynik pod nazwą Postgres95, jako darmowe, otwarte oprogramowanie.

W 1996 roku projekt zmienił nazwę na PostgreSQL, żeby oddać fakt, że silnik rozumie już SQL, zachowując jednocześnie oryginalne dziedzictwo POSTGRES. Numeracja wersji ruszyła od 6.0, kontynuując numerację z Berkeley. Od tego momentu PostgreSQL rozwija globalna społeczność ochotników i firm, nie jedna spółka: licencja PostgreSQL, bardzo zbliżona do BSD, pozwala każdemu używać, modyfikować i sprzedawać silnik bez opłaty, co odróżnia go od pozostałych trzech silników w tej bazie wiedzy.

Jak to działa w jednym obrazku

Niezależnie od tego, ile różnych aplikacji łączy się z PostgreSQL, droga jest ta sama: klient łączy się z jedną instancją, instancja kieruje połączenie do właściwej bazy w tym samym klastrze danych, a wynik pracy ląduje fizycznie jako plik w katalogu `PGDATA`.



Trzy aplikacje z Zielonej łączą się z jedną instancją PostgreSQL na pg01, każda do własnej bazy w tym samym klastrze danych (kliknięcie otwiera pełny rozmiar)

Dlaczego Zielona

> **Z ŻYCIA WZIĘTE**
>
> Kiedy zespół Oli postawił drugi wewnętrzny panel obok pierwszego, oba połączyły się bezpośrednio z tym samym `pg01`, nie sprawdzając, że w tej samej instancji siedzi już baza Zabbixa. W środku miesiąca limit `max_connections` omal nie odciął Zabbixowi monitoringu całej sieci, bo dwa nowe panele otwierały po kilkadziesiąt sesji naraz. Tomek dostał zadanie:

> postawić przed aplikacjami pulę połączeń, żeby kolejny nowy panel nie zabierał połączeń ważniejszej usłudze.

Na `pg01` PostgreSQL trzyma dziś bazy trzech różnych odbiorców: aplikacji zespołu Oli, Zabbixa monitorującego całą sieć i NetBoksa dokumentującego adresację. Żaden z nich nie musiał kupować licencji ani negocjować liczby rdzeni, bo PostgreSQL nie ma cennika. To właśnie ten brak tarcia sprawia, że w firmach takich jak Zielona PostgreSQL staje się domyślnym wyborem dla każdej nowej wewnętrznej aplikacji, nawet jeśli nikt formalnie tej decyzji nie podjął.

Jedna instancja obsługująca kilka niepowiązanych aplikacji ma jednak swoją cenę: awaria albo przeciążenie jednej bazy potrafi odbić się na sąsiadach w tym samym klastrze, bo dzielą jeden proces nadzorczy i jeden limit połączeń. Reszta tej książki pokazuje, jak rozpoznać taką sytuację, zanim zauważy ją Zabbix albo Marek z magazynu.

Mapa tej książki

Książka „PostgreSQL" składa się z dziewięciu rozdziałów, budowanych na słowniku z książki „Bazy danych: fundamenty".

Rozdział	Na jakie pytanie odpowiada
PostgreSQL: z Berkeley do każdej firmy	Skąd wziął się PostgreSQL i dlaczego trafia dziś do tylu firm
Architektura i instalacja	Z czego składa się klaster PostgreSQL i jak postawić go z własnym katalogiem danych
Połączenie i konfiguracja	Jak połączyć się klientem `psql` i co ustawić w plikach konfiguracyjnych
Rozpoznanie bazy	Jak sprawdzić rozmiary, obiekty i aktywność na nieznanej instancji
Użytkownicy i uprawnienia	Jak zarządzać rolami i kontami technicznymi
Backup i odtwarzanie	Jak działa WAL i jak zrobić kopię zapasową `pg_dump` albo pgBackRest
Wydajność i utrzymanie	Czym jest VACUUM, bloat i plan wykonania zapytania
Procedury	Instalacja, przeniesienie katalogu danych, upgrade silnika
Replikacja	Replikacja strumieniowa i logiczna
Awarie	Objaw, przyczyna, pierwsze kroki

Powiązane: Baza danych, silnik, instancja i schemat · Architektura PostgreSQL, klaster, baza, schemat, procesy

Źródła: dokumentacja PostgreSQL, „A Brief History of PostgreSQL", postgresql.org/docs/current/history.html · Wikipedia, „PostgreSQL" · Momjian B., „The History of PostgreSQL Open Source Development"

Revision #22
Created 2026-09-28 17:32:33 UTC by Tomasz Konieczny
Updated 2026-10-26 14:12:07 UTC by Tomasz Konieczny