



Zaawansowana administracja PostgreSQL: High Availability, Disaster Recovery, monitoring, skalowanie

Numer usługi 2026/02/05/202247/3310888

5 535,00 PLN brutto
4 500,00 PLN netto
158,14 PLN brutto/h
128,57 PLN netto/h
133,33 PLN cena rynkowa ⓘ

JSYSTEMS SPÓŁKA
Z OGRANICZONĄ
ODPOWIEDZIALNOŚ
CIĄ

Brak ocen dla tego dostawcy

- 📄 Usługa szkoleniowa
- 📺 zdalna w czasie rzeczywistym
- 🕒 35:00 h
- 📅 15.06.2026 do 19.06.2026

Informacje podstawowe

Kategoria	Informatyka i telekomunikacja / Bazy danych
Grupa docelowa usługi	Przeznaczone dla osób administrujących lub utrzymujących środowiska PostgreSQL, które chcą pogłębić wiedzę z zakresu replikacji, wysokiej dostępności, backupu, monitoringu oraz optymalizacji wydajności baz danych.
Minimalna liczba uczestników	6
Maksymalna liczba uczestników	15
Data zakończenia rekrutacji	11-06-2026
Forma prowadzenia usługi	zdalna w czasie rzeczywistym
Liczba godzin usługi	35
Podstawa uzyskania wpisu do BUR	Certyfikat systemu zarządzania jakością wg. ISO 9001:2015 (PN-EN ISO 9001:2015) - w zakresie usług szkoleniowych

Cel

Cel edukacyjny

Nabycie przez uczestników zaawansowanych umiejętności administrowania środowiskami PostgreSQL z naciskiem na wysoką dostępność, odtwarzanie po awarii, monitoring i skalowanie, umożliwiających samodzielne zarządzanie enterprise'owymi środowiskami PostgreSQL.

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Konfiguruje rozwiązania wysokiej dostępności PostgreSQL (Patroni, repmgr).	Uczestnik konfiguruje klaster HA PostgreSQL przy użyciu Patroni, symuluje awarię primary i weryfikuje automatyczne przełączenie (failover) na replikę.	Test teoretyczny z wynikiem generowanym automatycznie
Planuje i wdraża strategię Disaster Recovery dla PostgreSQL Monitoruje wydajność klastra PostgreSQL przy użyciu pg_stat_* i narzędzi zewnętrznych.	Uczestnik konfiguruje WAL archiving i PITR, wykonuje backup, symuluje utratę danych i przeprowadza odtwarzanie do wskazanego punktu w czasie, weryfikując kompletność danych. Uczestnik konfiguruje monitoring PostgreSQL eksportujący metryki przez postgres_exporter do Prometheus, tworzy dashboard w Grafanie i definiuje alert na długo działające zapytania	Test teoretyczny z wynikiem generowanym automatycznie Test teoretyczny z wynikiem generowanym automatycznie
Optymalizuje i skaluje środowisko PostgreSQL dla dużych obciążeń	Uczestnik identyfikuje wąskie gardło wydajnościowe przy użyciu pg_stat_activity i EXPLAIN ANALYZE, wdraża optymalizację (indeks, partycjonowanie lub connection pooler) i mierzy efekt.	Test teoretyczny z wynikiem generowanym automatycznie

Kwalifikacje

Kompetencje

Usługa prowadzi do nabycia kompetencji.

Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji zawiera opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument potwierdza, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji?

TAK

Pytanie 3. Czy dokument potwierdza zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

TAK

Program

1. Narzędzia kopii zapasowych

o Parametry bazy potrzebne przy konfiguracji kopii zapasowych

☒ archive_mode

☒ archive_command

☒ archive_timeout

☒ recovery_command

o Barman (Backup And Recovery MANager)

☒ Konfiguracja Barmana - backup strumieniowany

☒ Wykonywanie oraz zarządzanie kopiami zapasowymi

☒ Odzyskiwanie danych do wybranego punktu w czasie

☒ Odzyskiwanie wszystkich dostępnych danych

☒ Konfiguracja Barmana - backup rsync

o PgBackRest

☒ Konfiguracja pgBackRest

☒ Wykonanie oraz zarządzanie kopiami zapasowymi

☒ Backup pełny

☒ Backup przyrostowy

☒ Backup różnicowy

☒ Konfiguracja repozytorium w chmurze na przykładzie Azure

☒ Odzyskiwanie danych do punktu w czasie

☒ Odzyskiwanie wszystkich dostępnych danych

2. Replikacja

o Replikacja fizyczna - strumieniowa, hot standby

☒ Replikacja asynchroniczna

☒ Replikacja synchroniczna

o Replikacja fizyczna - wal-shiping, ciągle odtwarzanie, warm standby

o Opóźniona replikacja fizyczna

o Zatrzymywanie i wznowianie odtwarzania replikacji

o Failover, failback, switchover z fizyczną repliką

☒ promocja instancji standby na primary

☒ ponowne podłączenie starej instancji primary jako standby (recovery_target_timeline / pg_rewind)

o Replikacja logiczna

☒ Sposób pierwszy - z użyciem pg_dumpall

☒ Sposób drugi - z użyciem pgbackrest

☒ wykorzystanie kopii zapasowej lub serwera standby przy konfiguracji replikacji logicznej

☒ publikacje

☒ subskrypcje

❏ pg_create_logical_replication_slot

❏ replication origin

❏ pg_replication_origin_advance

o Failover z logiczną repliką

❏ promocja instancji standby na primary

❏ ponowne podłączenie starej instancji primary jako standby (recovery_target_timeline / pg_rewind)

o Monitorowanie replikacji

❏ pg_stat_replication (sent, write, flush, replay)

❏ pg_replication_slot

❏ log_recovery_conflict_waits (PG14)

❏ pg_publication

❏ pg_subscription

o Częste problemy z replikacją

o Parametry PostgreSQL przydatne przy replikacji

❏ wal_level

❏ hot_standby

❏ max_wal_senders

❏ max_replication_slots

❏ wal_keep_segments/wal_keep_size

❏ max_standby_archive_delay

❏ max_standby_streaming_delay

❏ idle_in_transaction_session_timeout

❏ hot_standby_feedback

❏ vacuum_defer_cleanup_age

❏ synchronous_commit

❏ recovery_target_timeline

3. HA z użyciem Patroni - PostgreSQL o wysokiej dostępności

o Co to Patroni?

o ETCD - rozproszony magazyn klucz-wartość

o Watchdog / softdog

o YAML

o HAProxy

o Konfiguracja ETCD

o Konfiguracja Patroni z etcd na Ubuntu

o Konfiguracja Patroni z etcd na CentOS

o Uruchomienie usługi watchdog

- o Przygotowanie pliku konfiguracyjnego patroni.yml
- o Pierwsze uruchomienie klastra
- o Konfiguracja kopii zapasowych i archiwizacji plików WAL
- o Konfiguracja pgBackRest dla klastra Patroni
- o automatyczne połączenie do serwera primary
- o Zarządzanie klastrem Patroni
- ☒ Przeglądanie instancji w klastrze Patroni
- ☒ Przełączanie serwera lidera w klastrze Patroni
- ☒ Wstrzymywanie instancji w klastrze Patroni
- ☒ Przywracanie wstrzymanej instancji w klastrze Patroni
- ☒ Failover w klastrze Patroni
- ☒ Switchover w klastrze Patroni
- ☒ Reinicjalizacja instancji w klastrze Patroni
- o zmiana parametrów Patroni / PostgreSQL
- o ponowne podłączenie instancji do klastra
- o odbudowa instancji

4. Skalowanie PostgreSQL z wykorzystaniem pgBouncer i HAProxy

- o Kiedy implementować
- o Konfiguracja
- o Jakie korzyści można osiągnąć
- o Konfiguracja pgBouncer
- o PgBouncer CLI
- o Loadbalancing i wysoka dostępność z HAProxy i pgbouncerem

5. Autovacuum - zaawansowane zagadnienia i tuning

- o Zapytania pomocne przy szukaniu potencjalnie problematycznych tabel
- o Tuning parametrów autovacuum
- o mechanizm działania
- o autovacuum_max_workers
- o autovacuum_freeze_max_age
- o autovacuum_naptime
- o autovacuum_vacuum_cost_delay
- o autovacuum_vacuum_cost_limit
- o autovacuum_work_mem
- o vacuum_cost_delay
- o vacuum_cost_limit
- o vacuum_cost_page_dirty

o vacuum_cost_page_hit

o vacuum_cost_page_miss

o Heap Only Tuple - czyli wydajniejszy update

🗒 Wyszukiwanie często aktualizowanych tabel

🗒 Szacowanie fillfactora dla tabel na podstawie dostępnych danych statystycznych

o pg_repack - bloat, czym jest, jak się go pozbyć online, migracja tabel i indeksów pomiędzy przestrzeniami tabel online

6. Monitoring PostgreSQL

o pg_stat_monitor

🗒 Instalacja pg_stat_monitor na Ubuntu

🗒 Instalacja pg_stat_monitor na CentOS

🗒 Utworzenie rozszerzenia w postgresie

🗒 PMM - Percona Monitoring and Management

🗒 Integracja z pg_stat_statements i pg_stat_monitor

🗒 Customowe zapytania

🗒 Konfiguracja monitoringu Patroni na podstawie danych z RestAPI /metrics

🗒 Monitoring baz PostgreSQL i analiza wydajności - na podstawie stacku VictoriaMetrics + Grafana oraz parsera logów pgBadger

🗒 konfiguracja logowania oraz generowanie i czytanie raportu pgBadger

🗒 istotne parametry monitoringu

🗒 identyfikacja i eliminacja wąskiego gardła za pomocą monitoringu i pgBadger

🗒 tuning na podstawie informacji z monitoringu oraz logu PostgreSQL

Harmonogram

Liczba pozycji harmonogramu: 16

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
1 z 16 Narzędzia kopii zapasowych	Mateusz Henicz	15-06-2026	09:00	12:30	03:30
2 z 16 Przerwa Obiadowa	Mateusz Henicz	15-06-2026	12:30	13:00	00:30
3 z 16 PgBackRest	Mateusz Henicz	15-06-2026	13:00	16:00	03:00
4 z 16 Replikacja	Mateusz Henicz	16-06-2026	09:00	12:30	03:30

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
5 z 16 Przerwa Obiadowa	Mateusz Henicz	16-06-2026	12:30	13:00	00:30
6 z 16 Parametry PostgreSQL przydatne przy replikacji	Mateusz Henicz	16-06-2026	13:00	16:00	03:00
7 z 16 HA z użyciem Patroni - PostgreSQL o wysokiej dostępności	Mateusz Henicz	17-06-2026	09:00	12:30	03:30
8 z 16 Przerwa Obiadowa	Mateusz Henicz	17-06-2026	12:30	13:00	00:30
9 z 16 Zarządzanie klastrem Patroni	Mateusz Henicz	17-06-2026	13:00	16:00	03:00
10 z 16 Skalowanie PostgreSQL z wykorzystaniem pgBouncer i HAProxy	Mateusz Henicz	18-06-2026	09:00	12:30	03:30
11 z 16 Przerwa Obiadowa	Mateusz Henicz	18-06-2026	12:30	13:00	00:30
12 z 16 Autovacuum - zaawansowane zagadnienia i tuning	Mateusz Henicz	18-06-2026	13:00	16:00	03:00
13 z 16 Monitoring PostgreSQL	Mateusz Henicz	19-06-2026	09:00	12:30	03:30
14 z 16 Przerwa Obiadowa	Mateusz Henicz	19-06-2026	12:30	13:00	00:30
15 z 16 Zajęcia szkoleniowe – podsumowanie	Mateusz Henicz	19-06-2026	13:00	15:30	02:30
16 z 16 Walidacja	Mateusz Henicz	19-06-2026	15:30	16:00	00:30

Cennik

Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	5 535,00 PLN
Koszt przypadający na 1 uczestnika netto	4 500,00 PLN
Koszt osobogodziny brutto	158,14 PLN
Koszt osobogodziny netto	128,57 PLN

Prowadzący

Liczba prowadzących: 1



1 z 1

Mateusz Henicz

Trener posiada wieloletnie doświadczenie w zaawansowanej administracji bazami danych PostgreSQL, przy czym kluczowe kwalifikacje w zakresie konfiguracji wysokiej dostępności, Disaster Recovery, monitoringu i skalowania PostgreSQL zostały zdobyte i są czynnie wykorzystywane w okresie ostatnich 5 lat (od 2021 roku do chwili obecnej). Potwierdzają to zrealizowane projekty wdrożeń PostgreSQL HA dla klientów enterprise oraz szkolenia w latach 2022–2026.

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

Informacje o materiałach dla uczestników usługi - Uczestnicy otrzymają komplet materiałów PDF. Każdy uczestnik otrzymuje kod dostępu i

dane logowania do platformy ZOOM na 7 dni przed datą rozpoczęcia szkolenia. Dane

przesyłane są na adres e-mail podany podczas rejestracji.

Warunki uczestnictwa

Podstawowa znajomość administracji baz danych PostgreSQL, umiejętność korzystania z komputera

Informacje dodatkowe

Warunkiem ukończenia szkolenia i otrzymania zaświadczenia jest uzyskanie minimalnej

frekwencji na poziomie 80% całkowitego czasu trwania usługi. Obecność uczestnika będzie

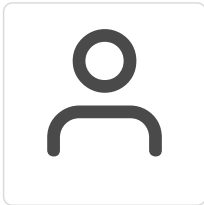
potwierdzana na podstawie codziennych list obecności lub logów z platformy online.

Warunki techniczne

Uczestnik musi dysponować sprzętem i łączem o parametrach:

- Procesor: min. 4-rdzeniowy (np. Intel i5/i7 lub odpowiednik AMD/M1/M2)
- Pamięć RAM: min. 16 GB
- Dysk: min. 20 GB wolnej przestrzeni
- System operacyjny: Windows 10/11 Pro, Linux lub macOS
- Multimedia: Sprawna kamera internetowa oraz mikrofon (wymagane do komunikacji i weryfikacji obecności)
- Łącze internetowe: Stabilne połączenie o minimalnej prędkości 10 Mbps (download) / 5 Mbps (upload)
- Oprogramowanie: Uprawnienia administratora pozwalające na instalację narzędzi

Kontakt



Biuro Obsługi Klienta

E-mail biuro@jssystem.com

Telefon (+48) 534 506 503