



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

Numer usługi 2026/02/12/202247/3330909

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

★★★★★ 4,6 / 5

5 ocen

- 👤 Usługa szkoleniowa
- 📄 zdalna w czasie rzeczywistym
- 👥 Zajęcia grupowe
- 🕒 35:00 h
- 📅 23.11.2026 do 27.11.2026

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	19-11-2026
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

Po ukończeniu tego szkolenia uczestnik będzie potrafił:

- stosować różne rodzaje replikacji baz danych PostgreSQL
- tworzyć kopie zapasowe i odzyskiwać dane w wielu konfiguracjach

- zarządzać klastrem HA przy użyciu Patroni
- prowadzić monitoring baz PostgreSQL, skalowanie i analizę wydajności.

Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
Po ukończeniu tego szkolenia uczestnik będzie potrafił: • stosować różne rodzaje replikacji baz danych PostgreSQL • tworzyć kopie zapasowe i odzyskiwać dane w wielu konfiguracjach • zarządzać klastrem HA przy użyciu Patroni • prowadzić monitoring baz PostgreSQL, skalowanie i analizę wydajności.	Każde omawiane zagadnienie jest utrwalane rzędem ćwiczeń o wzrastającym poziomie trudności. Podczas szkolenia uczestnicy wykonują ponad 50 warsztatów. W szkoleniu duży nacisk jest kładziony na dobre zrozumienie zasad działania wykorzystywanych technik, a nie ich odtwórcze stosowanie.	Obserwacja w warunkach rzeczywistych

Kwalifikacje

Kompetencje

Usługa prowadzi do nabycia kompetencji.

Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem zawierają opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji i zgodnie z zaplanowanymi metodami walidacji?

TAK

Pytanie 3. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają 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-shipping, ciągłe 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 Częstość 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: 0

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
Brak wyników.					

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

Mateusz na co dzień jest DBA w firmie Percona świadczącej support i consulting z zakresu PostgreSQL. Odpowiada za utrzymanie, tuning i monitoring klastrów PostgreSQL klientów firmy. Rozwiązuje problemy klientów na całym świecie. Ma szerokie, wieloletnie doświadczenie z produkcyjnymi środowiskami PostgreSQL od małych jednoinstancyjnych do wielkich, wielonodowych i przetwarzających terabajty danych. Ma pod swoją opieką klastry PostgreSQL z SLA 99,999%. Takie SLA oznacza, że serwer może być niedostępny przez 5 minut i 13 sekund w ciągu całego roku!

Informacje dodatkowe

Informacje o materiałach dla uczestników usługi

Uczestnicy otrzymają komplet materiałów PDF

Warunki uczestnictwa

Podstawowa znajomość administracji baz danych PostgreSQL, umiejętność korzystania z komputera

Informacje dodatkowe

Szkolenie kończy się wydaniem certyfikatu imiennego

Warunki techniczne

Dostęp do internetu, laptop/komputer, wideoaplikacja ZOOM lub dostęp do wersji w przeglądarce, możliwość podłączenia się do hostów przygotowanych na pulpicie zdalnym.

Kontakt



Biuro Obsługi Klienta

E-mail biuro@jsystems.pl

Telefon (+48) 534 506 503