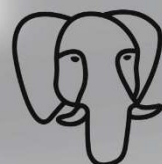




# JSYSTEMS

## Administracja PostgreSQL

**Archiwizacja  
ciągła, gorący  
backup z  
pg\_basebackup i  
jego odtwarzanie  
ćwiczenia**



PostgreSQL

# Archiwizacja ciągła, gorący backup z pg\_basebackup i jego odtwarzanie

<b>1</b>	Stwórz katalog “/archs” i zadбай o ustawienie dla niego odpowiednich uprawnień i właściciela.
<b>2</b>	Włącz archiwizację ciągłą w taki sposób, by archiwizowane pliki WAL trafiły do katalogu “/archs”.
<b>3</b>	Uruchom osobną sesję do serwera, przełącz się na użytkownika “postgres”. Umieść okno konsoli tak by było cały czas widoczne. Uruchom na nim poniższe polecenie: <pre>watch -n 1 'find /archs -type f -exec du -m {} +   awk "{sum += \\$1} END {print \"Liczba plików: \", NR; print \"Całkowity rozmiar w MB: \", sum}"'</pre>
<b>4</b>	Polecenie z powyższego punktu będzie co sekundę sprawdzać i wyświetlać ilość i sumaryczną wielkość plików WAL w katalogu “/archs”. Miej to okno na oku wykonując następujące kroki. W ramach osobnej sesji zaloguj się do psql jako “postgres” do bazy “postgres”. Stwórz tabelę i załaduj do niej trochę danych: <pre>create table test_archiwizacji(x integer, y text); insert into test_archiwizacji(x) select generate_series(1,10000000);</pre>
<b>5</b>	Zanotuj ile się utworzyło plików i ile miejsca zajmują.
<b>6</b>	Skasuj zawartość katalogu “archs”: <pre>rm /archs/*</pre>
<b>7</b>	Wykonaj update na tabeli: <pre>update test_archiwizacji set y='element numer '  x;</pre>
<b>8</b>	Ile plików wytworzyło się tym razem? Ile zajmują miejsca? Odpowiedz na pytanie: skąd taka różnica?
<b>9</b>	Jako użytkownik systemowy “postgres” stwórz katalog “/backups/hot_backup”.
<b>10</b>	Zaloguj się do psql, sprawdź jakie masz tabele i ewentualnie usuń duże niepotrzebne tabele, żeby przyspieszyć wykonywanie backupów i by nie zajmowały wiele miejsca.
<b>11</b>	Wykonaj kopię zapasową klastra za pomocą pg_basebackup do katalogu “/backups/hot_backup” w taki sposób, by w trakcie wykonywania backupu pokazywany był progres, a checkpoint wykonał się natychmiast.
<b>12</b>	Jako użytkownik systemowy “postgres” stwórz katalog “/backups/second_backup”.
<b>13</b>	Dodaj do pliku pg_hba.conf wpis pozwalający na wykonywanie kopii zapasowych z wewnętrznej sieci i przeładuj konfigurację. Jeśli korzystasz z Azure to Twoja podsieć to najprawdopodobniej “10.0.0.0/24”.
<b>14</b>	Wykonaj kopię zapasową klastra do katalogu “/backups/second_backup” korzystając z wewnętrzzięciowego adresu IP.
<b>15</b>	W poprzednim punkcie zostałeś poproszony o hasło przy próbie wykonania backupu. Nie chcemy by tak się działo, ponieważ zamierzamy zautomatyzować wywołanie robienia backupu. Stwórz więc odpowiedni plik “.pgpass” i wprowadź do niego niezbędne dane, tak by wykonywanie backupu nie wiązało się z podawaniem hasła.
<b>16</b>	Skonfiguruj odpowiednio parametr “restore_command”, w taki sposób by pobierał pliki WAL z “/archs”.

<b>17</b>	Stwórz katalog “/backups/test” i utwórz w nim kopię zapasową, weryfikując czy plik “.pgpass” zadziałał i udało się stworzyć kopię zapasową bez podawania hasła - łącząc się po sieci.
<b>18</b>	Wyczyść wszystkie zarchiwizowane pliki WAL starsze niż ostatni backup.
<b>19</b>	Podłącz się za pomocą “psql” do bazy “postgres” jako użytkownik “postgres” i stwórz tabelkę wypełniając ją danymi: <pre>create table table4restore(x integer); Insert into table4restore select generate_series(1,1000000);</pre>
<b>20</b>	Sprawdź i zapisz ilość elementów w tabeli: <pre>select count(*) from table4restore;</pre>
<b>21</b>	Nowo powstałej i wypełnionej tabeli nie ma w stworzonej wcześniej kopii zapasowej. Wszystkie informacje na jej temat znajdują się wyłącznie w plikach WAL. Odzyskamy teraz klaster wraz z tabelą.
<b>22</b>	Wyłącz klaster i skasuj całą zawartość PGDATA (/postgresql)
<b>23</b>	Odzyskaj klaster do “/postgresql” z backupu znajdującego się w “/backups/test”. Pamiętaj o stworzeniu pliku “recovery.signal”, ponieważ bez niego odtwarzanie ze zarchiwizowanych plików WAL nie dojdzie do skutku.
<b>24</b>	Podnieś odzyskany klaster za pomocą “pg_ctl”. Jeśli parametr “restore_command” masz poprawnie ustawiony, to odtworzone powinny zostać też zarchiwizowane pliki WAL - te, w których znajduje się informacja o tabelce “table4restore” i jej zawartości.
<b>25</b>	Zajrzyj do loga odzyskanego klastra i upewnij się, że nastąpiło odtwarzanie ze zarchiwizowanych plików WAL.
<b>26</b>	Sprawdź ilość elementów w tabeli “table4restore” i upewnij się, że jest ona taka sama jak przed odtwarzaniem.
<b>27</b>	Zatrzymaj odzyskany klaster za pomocą “pg_ctl” i uruchom go jako usługę.
<b>28</b>	Przejdziemy teraz scenariusz odzyskiwania danych po błędzie ludzkim. Podłącz się do klastra, sprawdź i zapisz aktualny czas: <pre>select current_timestamp;</pre>
<b>29</b>	Skasuj tabelę “table4restore”. Upewnij się, że została skasowana i nie ma jej w bazie.
<b>30</b>	Wywołaj "select pg_switch_wal()" żeby sobie zagwarantować, że wszystkie wymagane pliki WAL będą dostępne w archiwum
<b>31</b>	Stwórz katalog “/data_pg_restored” i zadбай o odpowiednie uprawnienia i właściciela niezbędne do odtworzenia do niego klastra.
<b>32</b>	Odzyskaj do katalogu “/data_pg_restored” ostatni backup - ten z “/backups/test”. Stwórz też w tym katalogu plik “recovery.signal”.
<b>33</b>	Wyedytuj plik “postgresql.conf” i ustaw w parametrze “recovery_target_time” punkt czasowy przed skasowaniem tabeli. Zmień też parametr “port” na wartość “5433”, tak byś mógł mieć uruchomione równolegle zarówno oryginalny klaster, jak i ten odzyskany.
<b>34</b>	Upewnij się, że w pliku “postgresql.auto.conf” klastra znajdującego się pod “/data_pg_restored” nie ma konfiguracji nadpisującej ustawienie portu z “postgresql.conf”. Jeśli jest, usuń wpis lub zmień port w “postgresql.auto.conf”.
<b>35</b>	Podnieś odzyskany klaster znajdujący się w “/data_pg_restored” za pomocą “pg_ctl”.
<b>36</b>	Podłącz się do odzyskanego klastra za pomocą “psql” (na porcie 5433) i sprawdź czy istnieje tabelka, którą skasowaliśmy na oryginalnym klastrze.
<b>37</b>	Za pomocą “pg_dump” i “psql” przepompuj w locie tabelę “table4restore” z odzyskanego do oryginalnego klastra:

	<code>pg_dump -p 5433 -t table4restore   psql</code>
<b>38</b>	Upewnij się, że odzyskana tabela “table4restore” znajduje się z powrotem w oryginalnym klastrze.