



**Administracja
PostgreSQL**



**ARCHIWIZACJA
CIĄGŁA, GORĄCY
BACKUP Z
PG_BASEBACKUP
I JEGO ODTWARZANIE
cheatsheet**

Archiwizacja ciągła, gorący backup z pg_basebackup i jego odtwarzanie

Włączenie archiwizacji ciągłej	<pre>#włączenie archiwizacji plików WAL alter system set archive_mode=on; pg_ctl -D /data_pg/ restart #stworzenie katalogu na zarchiwizowane pliki WAL sudo mkdir /archs sudo chown postgres:postgres /archs/ #ustawienie komendy archiwizującej pliki WAL alter system set archive_command='cp %p /archs/%f'; select pg_reload_conf();</pre>
Przykładowy kod generujący dużo plików WAL na potrzeby sprawdzenia archiwizacji ciągłej	<pre>create table archiwizacja(x integer,y text); do \$\$ begin for x in 1..1000000 loop insert into archiwizacja values (x,'element numer ' x); end loop; end \$\$;</pre>
Sprawdzenie ilości miejsca zajmowanego przez pliki WAL	<pre>du -sh /data_pg/pg_wal/</pre>
Stworzenie katalogu na potrzeby kopii zapasowej	<pre>sudo mkdir /backups sudo mkdir /backups/backup_1 sudo chown -R postgres:postgres /backups/</pre>
Najprostsza forma wykonania kopii zapasowej do katalogu „/backups/backup_1”. Łączenie domyślnie do localhosta na porcie 5432	<pre>pg_basebackup -D /backups/backup_1/</pre>
Stworzenie kopii zapasowej z jednoczesnym mapowaniem przestrzeni tabel znajdujących się poza PGDATA	<pre>pg_basebackup -D /backups/backup_2/ --tablespace- mapping=/other_disk=/backups/backup_2/other_disk</pre>
Stworzenie kopii zapasowej zdalnego hosta (przełącznik -h)	<pre>pg_basebackup -D /backups/backup_3/ -h 13.81.53.1</pre>
Konfiguracja pliku haseł, byśmy nie musieli podawać hasła przy każdym wywołaniu kopii zapasowej	<pre>cd ~ touch .pgpass chmod 600 .pgpass nano .pgpass #w pliku wpis wg systematyki: #ip_hosta_którego_robimy_backup:port:baza_danych:użytkownik:hasło #np: 13.81.53.1:5432:postgres:postgres:super_tajne_hasło</pre>
Stworzenie kopii zapasowej z pokazywaniem postępu prac	<pre>pg_basebackup -D /backups/backup_5/ -P</pre>

Stworzenie skompresowanej kopii zapasowej	<code>pg_basebackup -D /backups/backup_6/ -Ft -z</code>
Stworzenie kopii zapasowej z wywołaniem natychmiastowego checkpointa	<code>pg_basebackup --checkpoint=fast -D /backups/backup_7/</code>
Czyszczenie niepotrzebnych zarchiwizowanych plików WAL	<code>ls /archs/ -la grep backup</code> <code>pg_archivecleanup -d /archs/ 00000001000000070000005A.00000028.backup</code>
Odtwarzanie klastra z kopii zapasowej	<pre> #kopiujemy aktualne pliki WAL cp /data_pg/pg_wal/* /archs/ #zatrzymujemy klaster, czyściami PGDATA i odzyskujemy z kopii: pg_ctl -D /data_pg/ stop rm -rf /data_pg/* cp -R /backups/backup_8/* /data_pg nano /data_pg/postgresql.conf #w tym pliku konfigurujemy: restore_command = 'cp /archs/%f %p' #tworzymy plik sygnału odtwarzania: touch /data_pg/recovery.signal #startujemy klaster: pg_ctl -D /data_pg/ start </pre>
Odtwarzanie do punktu w czasie z kopii zapasowej	<pre> #kopiujemy aktualne pliki WAL cp /data_pg/pg_wal/* /archs/ #zatrzymujemy klaster, czyściami PGDATA i odzyskujemy z kopii: pg_ctl -D /data_pg/ stop rm -rf /data_pg/* cp -R /backups/backup_8/* /data_pg nano /data_pg/postgresql.conf #w tym pliku konfigurujemy: restore_command = 'cp /archs/%f %p' recovery_target_time = '2023-10-24 13:59:52.455783+00' #tworzymy plik sygnału odtwarzania: touch /data_pg/recovery.signal #startujemy klaster: pg_ctl -D /data_pg/ start #przejdźcie w tryb read-write psql -c "select pg_wal_replay_resume()" </pre>