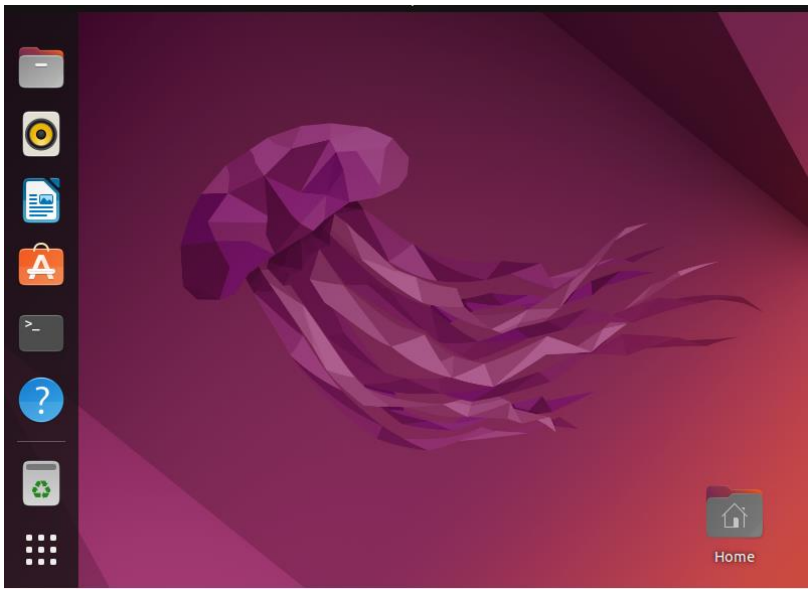
Przedmiot: Organizacja Systemów Zarządzania Baz Danych
Laboratorium 1: PgSQL – Replikacja strumieniowa
Autor: Bartłomiej Jamiołkowski, Adrianna Bodziony

Faza III - Pierwsze wprawki

a) Credentiale do maszyny wirtualne ubuntu/ubuntu. Jeśli potrzebujesz uprawnień roota użytkownik ten dodany jest do sudo



b) Do operowania na bazach postgres – mamy dedykowanego usera: postgres. Przełączenie z usera ubuntu na postgres: sudo -i -u postgres

Polecenie:

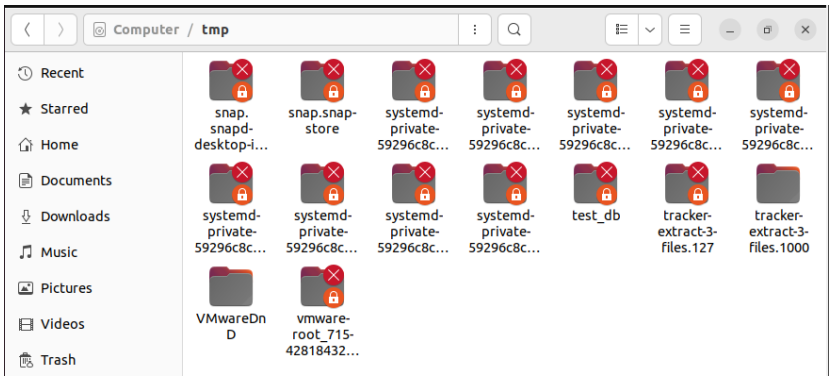
```
ubuntu@ubuntu-2204:~$ sudo -i -u postgres
postgres@ubuntu-2204:~$
```

c) Wszelkie narzędzia postgresowe zainstalowane są w następującej lokalizacji:

Polecenie:

```
ubuntu@ubuntu-2204:~$ sudo -i -u postgres
postgres@ubuntu-2204:~$ ls -l /usr/lib/postgresql/15/bin
total 12924
-rwxr-xr-x 1 root root   56048 Feb  6 07:37 clusterdb
-rwxr-xr-x 1 root root   60376 Feb  6 07:37 createdb
-rwxr-xr-x 1 root root   56400 Feb  6 07:37 createuser
-rwxr-xr-x 1 root root   51888 Feb  6 07:37 dropdb
-rwxr-xr-x 1 root root   51824 Feb  6 07:37 dropuser
-rwxr-xr-x 1 root root  118432 Feb  6 07:37 initdb
-rwxr-xr-x 1 root root   43848 Feb  6 07:37 oid2name
-rwxr-xr-x 1 root root   81464 Feb  6 07:37 pg_amcheck
-rwxr-xr-x 1 root root   39224 Feb  6 07:37 pg_archivecleanup
-rwxr-xr-x 1 root root  122368 Feb  6 07:37 pg_basebackup
-rwxr-xr-x 1 root root  163608 Feb  6 07:37 pgbench
-rwxr-xr-x 1 root root   56024 Feb  6 07:37 pg_checksums
-rwxr-xr-x 1 root root   43400 Feb  6 07:37 pg_config
-rwxr-xr-x 1 root root   55720 Feb  6 07:37 pg_controldata
-rwxr-xr-x 1 root root   68360 Feb  6 07:37 pg_ctl
-rwxr-xr-x 1 root root  389488 Feb  6 07:37 pg_dump
-rwxr-xr-x 1 root root   98176 Feb  6 07:37 pg_dumpall
-rwxr-xr-x 1 root root   39472 Feb  6 07:37 pg_isready
-rwxr-xr-x 1 root root   76816 Feb  6 07:37 pg_receivewal
-rwxr-xr-x 1 root root   56440 Feb  6 07:37 pg_recvlogical
-rwxr-xr-x 1 root root   60168 Feb  6 07:37 pg_resetwal
-rwxr-xr-x 1 root root  155416 Feb  6 07:37 pg_restore
-rwxr-xr-x 1 root root  109464 Feb  6 07:37 pg_rewind
-rwxr-xr-x 1 root root   43440 Feb  6 07:37 pg_test_fsync
-rwxr-xr-x 1 root root   31112 Feb  6 07:37 pg_test_timing
-rwxr-xr-x 1 root root  142328 Feb  6 07:37 pg_upgrade
-rwxr-xr-x 1 root root   96952 Feb  6 07:37 pg_verifybackup
-rwxr-xr-x 1 root root   97272 Feb  6 07:37 pg_waldump
-rwxr-xr-x 1 root root 9797112 Feb  6 07:37 postgres
lrwxrwxrwx 1 root root       8 Feb  6 07:37 postmaster -> postgres
-rwxr-xr-x 1 root root  761432 Feb  6 07:37 psql
-rwxr-xr-x 1 root root   64432 Feb  6 07:37 reindexdb
-rwxr-xr-x 1 root root   68816 Feb  6 07:37 vacuumdb
-rwxr-xr-x 1 root root   39592 Feb  6 07:37 vacuumlo
```

d) Wszelkie nowe instancje o których będzie mowa poniżej proponuje tworzyć w katalogu /tmp



e) Stwórz instancje serwera w lokalizacji /tmp/test_db (polecenie initdb z opcją -D)

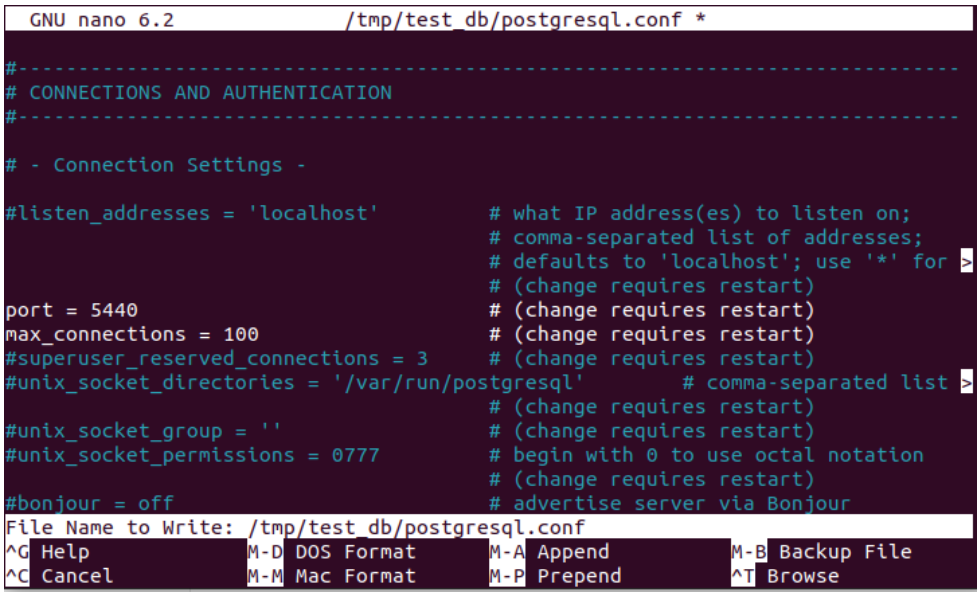
Polecenie: initdb –D /tmp/test_db

```
postgres@ubuntu-2204:~$ export PATH=$PATH:/usr/lib/postgresql/15/bin/
postgres@ubuntu-2204:~$ initdb -D /tmp/test_db
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

creating directory /tmp/test_db ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... America/New_York
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
```

f) Przestaw port na którym uruchomiona będzie ta instancja na 5440 (zmiana w pliku postgresql.conf)

Polecenie:

```
postgres@ubuntu-2204:/tmp/test_db$ nano /tmp/test_db/postgresql.conf
```



g) Uruchom tą instancję (narzędzie pg_ctl)

Polecenie:

```
postgres@ubuntu-2204:/tmp/test_db$ pg_ctl start -D /tmp/test_db -l logfile
waiting for server to start.... done
server started
postgres@ubuntu-2204:/tmp/test_db$
```

h) Połącz się z tą instancją z domyślną bazą postgres

Polecenie:

```
postgres@ubuntu-2204:/tmp/test_db$ psql -p 5440 -U postgres
psql (15.6 (Ubuntu 15.6-1.pgdg22.04+1))
Type "help" for help.

postgres=#
```

i) Utworz tabele tbl o strukturze: int id PK, varchar name

Polecenie:

```
postgres=# CREATE TABLE tbl (
    id SERIAL PRIMARY KEY,
    name VARCHAR
);
CREATE TABLE
```

j) Wyświetl schemat tabeli tbl;

Polecenie:

```
postgres=# \d tbl
```

```
                                Table "public.tbl"
 Column |       Type        | Collation | Nullable |              Default
--------+-------------------+-----------+----------+---------------------------
-----
 id     | integer           |           | not null | nextval('tbl_id_seq'::regcl
ass)
 name   | character varying |           |          |
Indexes:
    "tbl_pkey" PRIMARY KEY, btree (id)

(END)
```

k) Wstaw do tabeli tbl kilka przykładowych rekordów

Polecenia:

```
postgres=# INSERT INTO tbl (name) VALUES ('Bartek'), ('Filip'), ('Piotrek');
INSERT 0 3
postgres=# INSERT INTO tbl (name) VALUES ('Jacek'), ('Jasiek'), ('Wojtek');
INSERT 0 3
```

l) Pobierz wszystkie rekordy

Polecenie:

```
postgres=# SELECT * FROM tbl;
 id |  name
----+---------
  1 | Bartek
  2 | Filip
  3 | Piotrek
  4 | Jacek
  5 | Jasiek
  6 | Wojtek
(6 rows)
```

m) Usuń rekordy o id mniejszym od 3

Polecenie:

```
postgres=# DELETE FROM tbl WHERE id < 3;
```

```
DELETE 2
postgres=# SELECT * FROM tbl;
 id |  name
----+---------
  3 | Piotrek
  4 | Jacek
  5 | Jasiek
  6 | Wojtek
(4 rows)
```

n) Usuń z tabeli wszystkie rekordy

Polecenie:

```
postgres=# DELETE FROM tbl;
```

```
DELETE 4
postgres=# SELECT * FROM tbl;
 id | name
----+------
(0 rows)
```

o) Usuń tabele tbl;

Polecenie:

```
postgres=# DROP TABLE tbl;
DROP TABLE
```

p) Zamknij sesje

Polecenie:

```
postgres=# \q
```

q) Zatrzymaj instancje test_db

Polecenie:

```
postgres@ubuntu-2204:/tmp/test_db$ pg_ctl stop -D /tmp/test_db
waiting for server to shut down.... done
server stopped
```

Faza VI – Przebieg ćwiczenia

a) Na potrzeby ćwiczenia utwórz nową instancję serwera w lokalizacji /tmp/primary_db

```
postgres@ubuntu-2204:~$ export PATH=$PATH:/usr/lib/postgresql/15/bin/
postgres@ubuntu-2204:~$ initdb -D /tmp/primary_db
```

b) Efekt jakiego się spodziewamy

c) .

```
The files belonging to this database system will be owned by user "postgres"
.
This user must also own the server process.

The database cluster will be initialized with locale "en_US.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

Data page checksums are disabled.

creating directory /tmp/primary_db ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... America/New_York
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

initdb: warning: enabling "trust" authentication for local connections
initdb: hint: You can change this by editing pg_hba.conf or using the option
 -A, or --auth-local and --auth-host, the next time you run initdb.

Success. You can now start the database server using:

    pg_ctl -D /tmp/primary_db -l logfile start
```

d) Aby przygotować instancję primary do replikacji należy wykonać następujące kroki:

i. Zmiana konfiguracji (w postgresql.conf) na potrzeby komunikacji sieciowej
ii. Stworzenie dedykowanego na potrzeby replikacji użytkownika (dobra praktyka)
iii. Zezwolenie na zdalny dostęp do instancji (pg_hba.conf) (w naszym ćwiczeniu nie jest to konieczne)
iv. Uruchomienie/restart instancji primary_db

e) Ad d.i

i. Zmień w pliku konfiguracyjnym instancji primary parametr odpowiedzialny za akceptację połączeń przychodzących (listen_address) tak aby możliwe było łączenie się z tą instancja z dowolnego adresu

Polecenie:

```
postgres@ubuntu-2204:~$ cd /tmp/primary_db
postgres@ubuntu-2204:/tmp/primary_db$ nano /tmp/primary_db/postgresql.conf
```

```
  GNU nano 6.2          /tmp/primary_db/postgresql.conf *
# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = ''                 # write an extra PID file
                                        # (change requires restart)


#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -
listen_addresses = '0.0.0.0'
#listen_addresses = 'localhost'         # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' >
                                        # (change requires restart)
#port = 5432                            # (change requires restart)
max_connections = 100                   # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
#unix_socket_directories = '/var/run/postgresql'        # comma-separated l>
                                        # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                        # (change requires restart)
#bonjour = off                          # advertise server via Bonjour
                                        # (change requires restart)
#bonjour_name = ''                      # defaults to the computer name

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify
```

ii. Ponieważ obie instancje (primary oraz replika) będą uruchamiane na tym samych hoście – zmieniamy adres na którym uruchomiona będzie instancja primary na 5433

```
  GNU nano 6.2          /tmp/primary_db/postgresql.conf *
# If external_pid_file is not explicitly set, no extra PID file is written.
#external_pid_file = ''                 # write an extra PID file
                                        # (change requires restart)


#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -
listen_addresses = '0.0.0.0'
#listen_addresses = 'localhost'         # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' >
                                        # (change requires restart)
port = 5433                             # (change requires restart)
max_connections = 100                   # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
#unix_socket_directories = '/var/run/postgresql'        # comma-separated l>
                                        # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                        # (change requires restart)
#bonjour = off                          # advertise server via Bonjour
                                        # (change requires restart)
#bonjour_name = ''                      # defaults to the computer name

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify
```

iii. Po tych zmianach uruchamiamy instancję primary (narzędzie pg_ctl). Po tej czynności, w logach serwera powinniśmy zobaczyć informację że serwer został uruchomiony, jest gotowy do przyjmowania połączeń i że nasłuchuje połączeń z dowolnego adresu (0.0.0.0) na porcie 5433

```
postgres@ubuntu-2204:/tmp/primary_db$ pg_ctl start -D /tmp/primary_db -l log
file
waiting for server to start.... done
server started
postgres@ubuntu-2204:/tmp/primary_db$
```

f) Ad d.ii

i. Łączymy się z domyślną bazą danych postgres na uruchomionej instancji primary_db (narzędzie psql, dla przypomnienia port na którym nasłuchuje ta instancja to 5433)

Polecenie:

```
postgres@ubuntu-2204:/tmp/primary_db$ psql -p 5433 -U postgres
psql (15.6 (Ubuntu 15.6-1.pgdg22.04+1))
Type "help" for help.

postgres=#
```

ii. Tworzymy użytkownika repuser z ustawioną flagą „replication”

Polecenie:

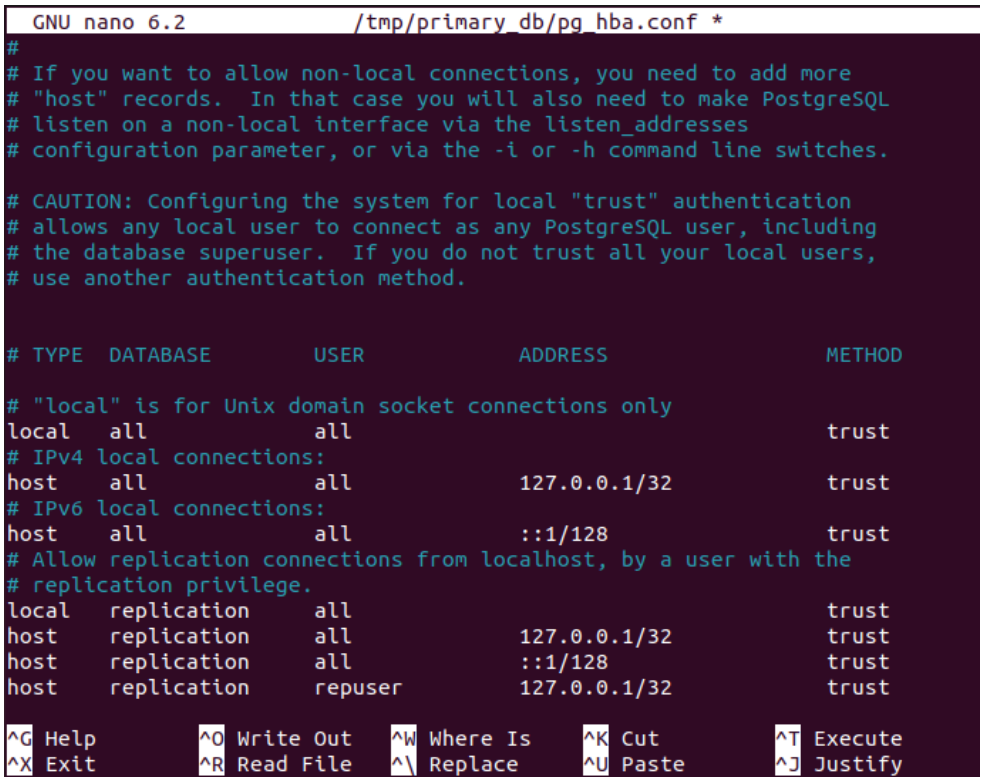
```
postgres=# CREATE USER repuser REPLICATION LOGIN;
CREATE ROLE
postgres=#
```

g) Ad d.iii

i. Edytujemy plik konfiguracyjny pg_hba.conf instancji primary_db i dodajemy możliwość łączenia się do niej użytkownika repuser z maszyny localhost (ponieważ mamy tam dodaną możliwość łączenia się dowolnego użytkownika z maszyny localhost krok ten jest w naszym przypadku nadmiarowy, niemniej w normalnych warunkach należałoby go wykonać, stąd proszę na potrzeby przećwiczenia o wykonanie tego kroku)

Polecenie:

```
postgres@ubuntu-2204:/tmp/primary_db$ nano /tmp/primary_db/pg_hba.conf
```



ii. Uruchamiamy / restartujemy instancję serwera primary_db

Polecenie:

```
postgres@ubuntu-2204:/tmp/primary_db$ pg_ctl restart -D /tmp/primary_db -l l
ogfile
waiting for server to shut down.... done
server stopped
waiting for server to start.... done
server started
```

h) Następnie przygotowujemy instancję repliki. W ramach ćwiczenia wykonamy to korzystając z narzędzia pg_basebackup

i. No więc wywołujemy narzędzie pg_basebackup tworząc kopie instancji primary a zatem łącząc się z localhostem (opcja -h), na użytkowniku repuser (opcja -U) na porcie 5433 (opcja -p) tworząc nową instancję w lokalizacji /tmp/replica_db (opcja -D), konfigurując ją od razu na potrzeby replikacji (opcja -R), ustawiając „recykling” plików WAL kiedy replika je w pełni przetworzy (opcja -C), tworząc slot na potrzeby utrzymania plików WAL (opcja --slot) z dowolną nazwą (np. slot_name) i ustawiając częstotliwość uaktualniania plików log (opcja –checkpoint) na „fast”

Polecenie:

```
postgres@ubuntu-2204:/tmp/primary_db$ pg_basebackup -h localhost -U repuser
-p 5433 -D /tmp/replica_db -R -C -S slot_name --checkpoint=fast
postgres@ubuntu-2204:/tmp/primary_db$
```

ii. W efekcie prawidłowo wykonanej komendy pg_basebackup zgodnie z opisem powyżej w lokalizacji /tmp/replica_db powinny pojawić się pliki typowe dla każdej instancji serwera pgsql (de facto kopia tego co mieliśmy w instancji

primary_db) plus m.in. dodatkowy plik/flaga standby.signal informujący o tym że instancją ta będzie instancją typu standby na potrzeby replikacji:

iii. .

Polecenie:

```
postgres@ubuntu-2204:/tmp/replica_db$ ls -al
total 272
drwx------ 19 postgres postgres   4096 Apr 29 09:27 .
drwxrwxrwt 23 root     root       4096 Apr 29 09:27 ..
-rw-------  1 postgres postgres    225 Apr 29 09:27 backup_label
-rw-------  1 postgres postgres 137177 Apr 29 09:27 backup_manifest
drwx------  5 postgres postgres   4096 Apr 29 09:27 base
drwx------  2 postgres postgres   4096 Apr 29 09:27 global
-rw-------  1 postgres postgres   3146 Apr 29 09:27 logfile
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_commit_ts
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_dynshmem
-rw-------  1 postgres postgres   4835 Apr 29 09:27 pg_hba.conf
-rw-------  1 postgres postgres   1636 Apr 29 09:27 pg_ident.conf
drwx------  4 postgres postgres   4096 Apr 29 09:27 pg_logical
drwx------  4 postgres postgres   4096 Apr 29 09:27 pg_multixact
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_notify
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_replslot
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_serial
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_snapshots
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_stat
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_stat_tmp
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_subtrans
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_tblspc
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_twophase
-rw-------  1 postgres postgres      3 Apr 29 09:27 PG_VERSION
drwx------  3 postgres postgres   4096 Apr 29 09:27 pg_wal
drwx------  2 postgres postgres   4096 Apr 29 09:27 pg_xact
-rw-------  1 postgres postgres    441 Apr 29 09:27 postgresql.auto.conf
-rw-------  1 postgres postgres  29620 Apr 29 09:27 postgresql.conf
-rw-------  1 postgres postgres      0 Apr 29 09:27 standby.signal
```

iv. Oraz z plikiem postgresql.auto.conf zawierającym parametry połączenia do instancji primary:

v. .

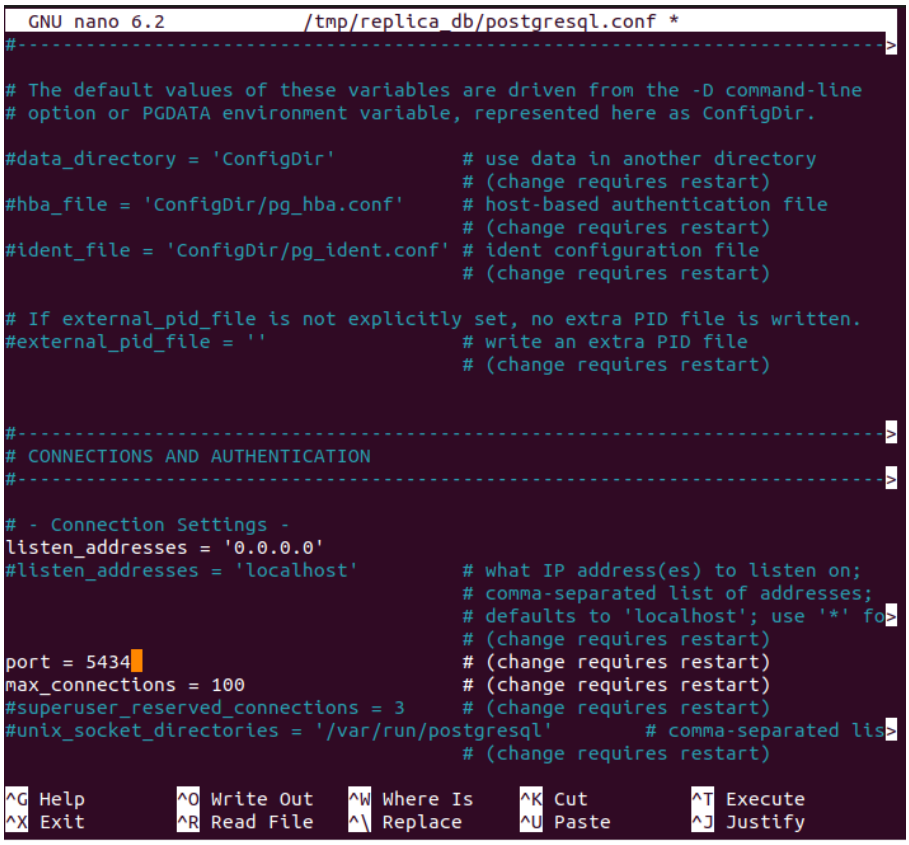
Polecenie:

```
postgres@ubuntu-2204:/tmp/replica_db$ cat postgresql.auto.conf
# Do not edit this file manually!
# It will be overwritten by the ALTER SYSTEM command.
primary_conninfo = 'user=repuser passfile=''/var/lib/postgresql/.pgpass'' chan
nel_binding=prefer host=localhost port=5433 sslmode=prefer sslcompression=0 ss
lcertmode=allow sslsni=1 ssl_min_protocol_version=TLSv1.2 gssencmode=prefer kr
bsrvname=postgres gssdelegation=0 target_session_attrs=any load_balance_hosts=
disable'
primary_slot_name = 'slot_name'
```

vi. Ponieważ uruchamiamy obie instancje na jednym hoście potrzebujemy przestawić port (na 5434) na którym uruchamiana będzie nasza instancja repliki – proszę dokonać odpowiedniej zmiany w pliku konfiguracyjnym repliki

Polecenie:

```
postgres@ubuntu-2204:/tmp/replica_db$ nano /tmp/replica_db/postgresql.conf
```



vii. Po tych zmianach uruchamiamy naszą replikę (narzędzie pg_ctl). Jeżeli wszystko wykonane zostało poprawnie po uruchomieniu repliki w jej logach powinniśmy zobaczyć informacje jak poniżej:

Polecenie:

```
postgres@ubuntu-2204:/tmp/replica_db$ pg_ctl -D /tmp/replica_db -l logfile sta
rt
waiting for server to start.... done
server started
```

```
.pgdg22.04+1) on x86_64-pc-linux-gnu, compiled by gcc (Ubuntu 11.4.0-1ubuntu1~22
.04) 11.4.0, 64-bit
2024-04-29 11:12:18.836 EDT [4751] LOG:  listening on IPv4 address "0.0.0.0", po
rt 5434
2024-04-29 11:12:18.836 EDT [4751] LOG:  listening on Unix socket "/var/run/post
gresql/.s.PGSQL.5434"
2024-04-29 11:12:18.839 EDT [4754] LOG:  database system was interrupted; last k
nown up at 2024-04-29 09:27:44 EDT
2024-04-29 11:12:18.845 EDT [4754] LOG:  entering standby mode
2024-04-29 11:12:18.847 EDT [4754] LOG:  starting backup recovery with redo LSN
0/2000028, checkpoint LSN 0/2000060, on timeline ID 1
2024-04-29 11:12:18.849 EDT [4754] LOG:  redo starts at 0/2000028
2024-04-29 11:12:18.849 EDT [4754] LOG:  completed backup recovery with redo LSN
 0/2000028 and end LSN 0/2000138
2024-04-29 11:12:18.849 EDT [4754] LOG:  consistent recovery state reached at 0/
2000138
2024-04-29 11:12:18.849 EDT [4751] LOG:  database system is ready to accept read
-only connections
2024-04-29 11:12:18.858 EDT [4755] LOG:  started streaming WAL from primary at 0
/3000000 on timeline 1
2024-04-29 11:17:18.924 EDT [4752] LOG:  restartpoint starting: time
2024-04-29 11:17:18.926 EDT [4752] LOG:  restartpoint complete: wrote 1 buffers
(0.0%); 0 WAL file(s) added, 0 removed, 1 recycled; write=0.001 s, sync=0.001 s,
 total=0.002 s; sync files=0, longest=0.000 s, average=0.000 s; distance=16384 k
B, estimate=16384 kB
2024-04-29 11:17:18.926 EDT [4752] LOG:  recovery restart point at 0/3000060
postgres@ubuntu-2204:/tmp/replica_db$
```

viii. .

ix. A zatem powinniśmy dostać potwierdzenie że instancja replica uruchomiona została na porcie 5434, w trybie standby, że akceptuje wyłącznie połączenia typu read-only, i że rozpoczęte zostało streamowanie plików WAL z instancji primary. Dodatkowo w widoku pg_stat_replication instancji primary powinien pojawić się wpis jak poniżej:

```
postgres=# \x
Expanded display is on.
postgres=# SELECT * FROM pg_stat_replication;
-[ RECORD 1 ]----+------------------------------
pid              | 4756
usesysid         | 16388
usename          | repuser
application_name | walreceiver
client_addr      | 127.0.0.1
client_hostname  |
client_port      | 52998
backend_start    | 2024-04-29 11:12:18.85433-04
backend_xmin     |
state            | streaming
sent_lsn         | 0/3000148
write_lsn        | 0/3000148
flush_lsn        | 0/3000148
replay_lsn       | 0/3000148
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 0
sync_state       | async
reply_time       | 2024-04-29 12:11:53.724124-04

postgres=#
```

Zaś w widoku pg_stat_wal_receiver powinien pojawić się wpis jak poniżej:

```
postgres=# \x
Expanded display is on.
postgres=# SELECT * FROM pg_stat_wal_receiver;
-[ RECORD 1 ]---------+----------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-----------------------------------------------------
pid                   | 4755
status                | streaming
receive_start_lsn     | 0/3000000
receive_start_tli     | 1
written_lsn           | 0/3000148
flushed_lsn           | 0/3000148
received_tli          | 1
last_msg_send_time    | 2024-04-29 12:17:54.249116-04
last_msg_receipt_time | 2024-04-29 12:17:54.249131-04
latest_end_lsn        | 0/3000148
latest_end_time       | 2024-04-29 11:12:18.858631-04
slot_name             | slot_name
sender_host           | localhost
sender_port           | 5433
conninfo              | user=repuser passfile=/var/lib/postgresql/.pgpass channe
l_binding=prefer dbname=replication host=localhost port=5433 fallback_applicatio
n_name=walreceiver sslmode=prefer sslcompression=0 sslcertmode=allow sslsni=1 ss
l_min_protocol_version=TLSv1.2 gssencmode=prefer krbsrvname=postgres gssdelegati
on=0 target_session_attrs=any load_balance_hosts=disable
```

i. W celu przetestowania:

i. Utwórz na instancji primary przykładową tabelę z kilkoma przykładowymi rekordami. Zweryfikuj czy tabela wraz z nowymi rekordami została zreplikowana

Polecenie:

```
postgres@ubuntu-2204:/$ psql -p 5433 -U postgres
psql (15.6 (Ubuntu 15.6-1.pgdg22.04+1))
Type "help" for help.

postgres=# CREATE TABLE example_table (
    id SERIAL PRIMARY KEY,
    name VARCHAR(50)
);
CREATE TABLE
postgres=# INSERT INTO example_table (name) VALUES ('Bartek'), ('Janek'), ('Piotrek'
);
INSERT 0 3
```

```
postgres@ubuntu-2204:/$ psql -p 5434 -U postgres
psql (15.6 (Ubuntu 15.6-1.pgdg22.04+1))
Type "help" for help.

postgres=# SELECT * FROM example_table;
 id |  name
----+---------
  1 | Bartek
  2 | Janek
  3 | Piotrek
(3 rows)
```

ii. Dodatkowo sprawdź działanie replikacji dla operacji delete oraz truncate

Polecenie delete:

```
postgres@ubuntu-2204:/$ psql -p 5433 -U postgres
psql (15.6 (Ubuntu 15.6-1.pgdg22.04+1))
Type "help" for help.

postgres=# DELETE FROM example_table;
DELETE 3
```

```
postgres@ubuntu-2204:/$ psql -p 5434 -U postgres
psql (15.6 (Ubuntu 15.6-1.pgdg22.04+1))
Type "help" for help.

postgres=# SELECT * FROM example_table;
 id | name
----+------
(0 rows)
```

Polecenie truncate:

```
postgres@ubuntu-2204:/$ psql -p 5433 -U postgres
psql (15.6 (Ubuntu 15.6-1.pgdg22.04+1))
Type "help" for help.

postgres=# INSERT INTO example_table (name) VALUES ('Bartek'), ('Janek'), ('Piotrek'
);
INSERT 0 3
postgres=# TRUNCATE TABLE example_table;
TRUNCATE TABLE
```

```
postgres@ubuntu-2204:/$ psql -p 5434 -U postgres
psql (15.6 (Ubuntu 15.6-1.pgdg22.04+1))
Type "help" for help.

postgres=# SELECT * FROM example_table;
 id | name
----+------
(0 rows)
```

j. Zasymuluj awarię serwera podstawowego i wykonaj operację recznego fail-overa mianując dotychczasową replikę do roli serwera podstawowego

Polecenie:

```
postgres@ubuntu-2204:/$ pg_ctl stop -D /tmp/primary_db
waiting for server to shut down.... done
server stopped
```

```
postgres@ubuntu-2204:~$ pg_ctl start -D /tmp/replica_db -l logfile
waiting for server to start.... done
server started
```

VIII. Zadanie dodatkowe / domowe

a. Rozszerz konfigurację o dwa dodatkowe serwery replikacji i skonfiguruj całość tak aby replikacja odbywała się:

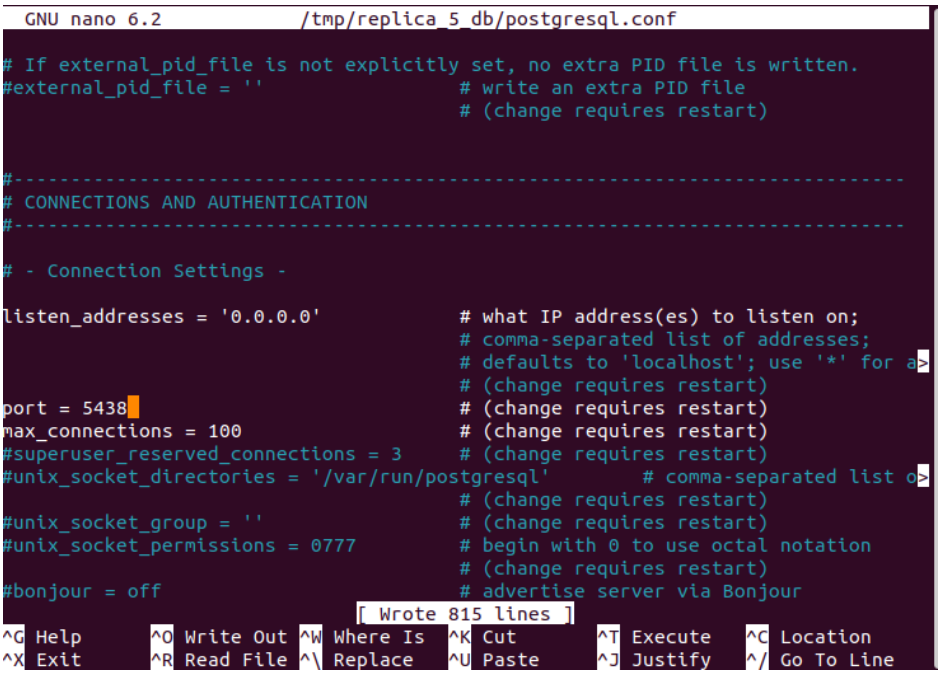
i. Bezpośrednio z serwera primary do wszystkich serwerów zapasowych (tryb multi-standby setup)

Polecenie:

```
postgres@ubuntu-2204:/tmp/primary_db$ pg_basebackup -h localhost -U repuser -p 54
33 -D /tmp/replica_4_db -R -C -S slot_name_4 --checkpoint=fast
```

```
postgres@ubuntu-2204:/tmp/primary_db$ pg_basebackup -h localhost -U repuser -p 54
33 -D /tmp/replica_5_db -R -C -S slot_name_5 --checkpoint=fast
```

```
  GNU nano 6.2              /tmp/replica_4_db/postgresql.conf
#external_pid_file = ''                 # write an extra PID file
                                        # (change requires restart)


#------------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#------------------------------------------------------------------------------

# - Connection Settings -

listen_addresses = '0.0.0.0'            # what IP address(es) to listen on;
                                        # comma-separated list of addresses;
                                        # defaults to 'localhost'; use '*' for a>
                                        # (change requires restart)
port = 5437                             # (change requires restart)
max_connections = 100                   # (change requires restart)
#superuser_reserved_connections = 3     # (change requires restart)
#unix_socket_directories = '/var/run/postgresql'        # comma-separated list o>
                                        # (change requires restart)
#unix_socket_group = ''                 # (change requires restart)
#unix_socket_permissions = 0777         # begin with 0 to use octal notation
                                        # (change requires restart)
#bonjour = off                          # advertise server via Bonjour
                                        # (change requires restart)
#bonjour_name = ''                      # defaults to the computer name
                        [ Wrote 815 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```




```
postgres=# \x
Expanded display is on.
postgres=# SELECT * FROM pg_stat_wal_receiver ;
-[ RECORD 1 ]---------+----------------------------------------------------------
----------------------------------------------------------------------------------
----------------------------------------------------------------------------------
----------------------------------------------------------------------------------
-------------------------------------------------
pid                   | 3841
status                | streaming
receive_start_lsn     | 0/5000000
receive_start_tli     | 1
written_lsn           | 0/7000148
flushed_lsn           | 0/7000148
received_tli          | 1
last_msg_send_time    | 2024-05-06 09:38:54.397006-04
last_msg_receipt_time | 2024-05-06 09:38:54.397063-04
latest_end_lsn        | 0/7000148
latest_end_time       | 2024-05-06 09:38:24.163302-04
slot_name             | slot_name_4
sender_host           | localhost
sender_port           | 5433
conninfo              | user=repuser passfile=/var/lib/postgresql/.pgpass channel
_binding=prefer dbname=replication host=localhost port=5433 fallback_application_
name=walreceiver sslmode=prefer sslcompression=0 sslcertmode=allow sslsni=1 ssl_m
in_protocol_version=TLSv1.2 gssencmode=prefer krbsrvname=postgres gssdelegation=0
 target_session_attrs=any load_balance_hosts=disable
```

```
postgres=# \x
Expanded display is on.
postgres=# SELECT * FROM pg_stat_wal_receiver ;
-[ RECORD 1 ]---------+-----------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------------------------------------
-------------------------------------------------
pid                   | 3865
status                | streaming
receive_start_lsn     | 0/7000000
receive_start_tli     | 1
written_lsn           | 0/7000148
flushed_lsn           | 0/7000148
received_tli          | 1
last_msg_send_time    | 2024-05-06 09:40:41.696687-04
last_msg_receipt_time | 2024-05-06 09:40:41.696808-04
latest_end_lsn        | 0/7000148
latest_end_time       | 2024-05-06 09:40:41.696687-04
slot_name             | slot_name_5
sender_host           | localhost
sender_port           | 5433
conninfo              | user=repuser passfile=/var/lib/postgresql/.pgpass channel
_binding=prefer dbname=replication host=localhost port=5433 fallback_application_
name=walreceiver sslmode=prefer sslcompression=0 sslcertmode=allow sslsni=1 ssl_m
in_protocol_version=TLSv1.2 gssencmode=prefer krbsrvname=postgres gssdelegation=0
 target_session_attrs=any load_balance_hosts=disable
```

```
-[ RECORD 1 ]----+------------------------------
pid              | 3451
usesysid         | 16388
usename          | repuser
application_name | walreceiver
client_addr      | 127.0.0.1
client_hostname  |
client_port      | 34730
backend_start    | 2024-05-06 08:58:46.799799-04
backend_xmin     |
state            | streaming
sent_lsn         | 0/7000148
write_lsn        | 0/7000148
flush_lsn        | 0/7000148
replay_lsn       | 0/7000148
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 0
sync_state       | async
reply_time       | 2024-05-06 09:42:16.665914-04
-[ RECORD 2 ]----+------------------------------
pid              | 3842
usesysid         | 16388
usename          | repuser
application_name | walreceiver
client_addr      | 127.0.0.1
client_hostname  |
:
```

```
sync_state        | async
reply_time        | 2024-05-06 09:42:16.665914-04
-[ RECORD 2 ]----+------------------------------
pid               | 3842
usesysid          | 16388
usename           | repuser
application_name  | walreceiver
client_addr       | 127.0.0.1
client_hostname   |
client_port       | 34744
backend_start     | 2024-05-06 09:38:24.160664-04
backend_xmin      |
state             | streaming
sent_lsn          | 0/7000148
write_lsn         | 0/7000148
flush_lsn         | 0/7000148
replay_lsn        | 0/7000148
write_lag         |
flush_lag         |
replay_lag        |
sync_priority     | 0
sync_state        | async
reply_time        | 2024-05-06 09:42:24.637218-04
-[ RECORD 3 ]----+------------------------------
pid               | 3866
```

```
sync_state        | async
reply_time        | 2024-05-06 09:42:24.637218-04
-[ RECORD 3 ]----+------------------------------
pid               | 3866
usesysid          | 16388
usename           | repuser
application_name  | walreceiver
client_addr       | 127.0.0.1
client_hostname   |
client_port       | 34746
backend_start     | 2024-05-06 09:40:41.692825-04
backend_xmin      |
state             | streaming
sent_lsn          | 0/7000148
write_lsn         | 0/7000148
flush_lsn         | 0/7000148
replay_lsn        | 0/7000148
write_lag         |
flush_lag         |
replay_lag        |
sync_priority     | 0
sync_state        | async
reply_time        | 2024-05-06 09:42:21.809499-04

(END)
```

ii. Kaskadowo z serwera primary do jednego z serwerów zapasowych, stamtąd do kolejnego serwera zapasowego etc (tryb cascade setup)

Utworzono kolejną replikę na podstawie 2 repliki (port 5434) na porcie 5436:

```
postgres@LAPTOP-P9AVKL9O:/usr/lib/postgresql/16/bin$ ./pg_basebackup -h localhost -U repuser -p 5434 -D /tmp/replica_db_4 -R -C --slot slot_name
_4 --checkpoint=fast
postgres@LAPTOP-P9AVKL9O:/usr/lib/postgresql/16/bin$ nano /tmp/replica_db_4/postgresql.conf
postgres@LAPTOP-P9AVKL9O:/usr/lib/postgresql/16/bin$ ./pg_ctl -D /tmp/replica_db_4 -l /tmp/replica_db_4/logfile start
waiting for server to start.... done
server started
```

Polecenie select * from pg_stat_replication; wykonane na replice:

```
postgres=# select * from pg_stat_replication;
-[ RECORD 1 ]----+------------------------------
pid              | 97128
usesysid         | 16388
usename          | repuser
application_name | walreceiver
client_addr      | 127.0.0.1
client_hostname  |
client_port      | 47006
backend_start    | 2024-05-06 14:20:35.425621+02
backend_xmin     |
state            | streaming
sent_lsn         | 0/6017610
write_lsn        | 0/6017610
flush_lsn        | 0/6017610
replay_lsn       | 0/6017610
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 0
sync_state       | async
reply_time       | 2024-05-06 14:40:25.825236+02
```

Oraz polecenie na kolejnej replice - select * from pg_stat_wal_receiver;

```
postgres=# select * from pg_stat_wal_receiver;
-[ RECORD 1 ]---------+----------------------------------------------------------------------------------------------------------------------------------
pid                   | 97127
status                | streaming
receive_start_lsn     | 0/6000000
receive_start_tli     | 1
written_lsn           | 0/6017610
flushed_lsn           | 0/6000000
received_tli          | 1
last_msg_send_time    | 2024-05-06 14:41:35.819155+02
last_msg_receipt_time | 2024-05-06 14:41:35.819519+02
latest_end_lsn        | 0/6017610
latest_end_time       | 2024-05-06 14:20:35.431594+02
slot_name             | slot_name_4
sender_host           | localhost
sender_port           | 5434
conninfo              | user=repuser passfile=/var/lib/postgresql/.pgpass channel_binding=prefer dbname=replication host=localhost port=5434 fal
lback_application_name=walreceiver sslmode=prefer sslcompression=0 sslcertmode=allow sslsni=1 ssl_min_protocol_version=TLSv1.2 gssencmode=prefer
 krbsrvname=postgres gssdelegation=0 target_session_attrs=any load_balance_hosts=disable
```

Potwierdzenie że kolejna replika ma taką samą zawartość:

```
postgres@LAPTOP-P9AVKL9O:/usr/lib/postgresql/16/bin$ ./psql  -p 5436  -d postgres
psql (16.2 (Ubuntu 16.2-1.pgdg22.04+1))
Type "help" for help.

postgres=# select * from tmp_table;
 id
----
(0 rows)
```

Istnieje tabela, utworzona na primary_db.