Создаем стенд

Текущий стенд состоит из 3-ех
узлов: `vs01.sXXXXXX`, `vs02.sXXXXXX`, `vs03.sXXXXXX` (XXXXXX - ваш номер студента).
Работа стенда 6 часов. 2 попытки запуска стенда.

1. Над текстом нажмите кнопку "Создать стенд". Каждый стенд создан под определенную тему курса. Сейчас вы находитесь в теме - **8. Логическая репликация** , этот стенд не подойдет под практические занятия из других пунктов курса.

Запуск обычно идёт до 10 минут, в редких случаях - до 30 минут.

2. После создания стенда вам нужно зайти по SSH на **sbox.slurm.io** с логином и паролем из личного кабинета (https://edu.slurm.io/):

```
ssh s000001@sbox.slurm.io
```

s000001 нужно заменить на ваш номер студента

Далее вам нужно перейти на сам стенд (sbox является просто jump-хостом и не понадобится кроме как для входа):

```
ssh vs01.s000001
```

s000001 снова нужно заменить на ваш личный номер студента

На хосте vs01 у вас есть полные права (включая беспарольный sudo):

```
sudo -i
```

Будьте осторожны и не сломайте стенд до прохождения всех заданий :)

Практика: Знакомство с логической репликацией

Можете запустить скрипт на стенде `~/practice/lecture4/1.log_replica.sh`, который спикер запускал в видео.

Или пройтись по шагам текущей практики
- https://gitlab.slurm.io/postgres/slurm_course/-/blob/main/practice/lecture4/1.log_replica.md, также представленной ниже по тексту.

Инициализация кластера

Мы зашли на стенд. Теперь все действия с СУБД PostgreSQL мы будем выполнять от локального пользователя postgres. Для этого выполним команду

```
student$ sudo -u postgres -i
```

1. Инициализируем кластер

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 -o "--data-checksums" initdb

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 enabled.
creating directory /var/lib/pgsql/12/main5432 ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... UTC
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
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:
    /usr/pgsql-12/bin/pg_ctl -D /var/lib/pgsql/12/main5432 -l logfile start
```

2. Запускаем сервер на порту 5432

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 start

waiting for server to start....2021-04-14 15:03:13.103 UTC [27608] LOG:  starting
PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-39), 64-bit
2021-04-14 15:03:13.103 UTC [27608] LOG:  listening on IPv4 address "127.0.0.1", port
5432
2021-04-14 15:03:13.107 UTC [27608] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-14 15:03:13.119 UTC [27608] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-14 15:03:13.133 UTC [27608] LOG:  redirecting log output to logging collector
process
2021-04-14 15:03:13.133 UTC [27608] HINT:  Future log output will appear in directory
"log".
 done
server started
```

3. Смотрим, какой уровень информации записывается в wal

```
postgres$ psql
psql (12.5)
```

```
Type "help" for help.

5432=> \x
Expanded display is on.

5432=> select * from pg_settings where name ='wal_level';
-[ RECORD 1 ]---+---------------------------------------------
name            | wal_level
setting         | replica
unit            |
category        | Write-Ahead Log / Settings
short_desc      | Set the level of information written to the WAL.
extra_desc      |
context         | postmaster
vartype         | enum
source          | default
min_val         |
max_val         |
enumvals        | {minimal,replica,logical}
boot_val        | replica
reset_val       | replica
sourcefile      |
sourceline      |
pending_restart | f
```

4. Для логической репликации этого недостаточно. Повышаем уровень записи до logical

```
5432=> alter system set wal_level=logical;
```

5. Создаем новую базу данных и таблицу в ней. Вставляем в таблицу 2 записи

```
5432=> create database logical_replica;
CREATE DATABASE

5432=> \c logical_replica

5432=> create table test(id int primary key, name text);
CREATE TABLE

5432=> insert into test values (1, 'Value 1'), (2, 'Value 2');
INSERT 0 2

5432=> \q
```

6. Перезапускаем сервер PostgreSQL

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 restart

waiting for server to shut down.... done
server stopped
waiting for server to start....2021-04-14 16:26:12.493 UTC [28392] LOG:  starting
PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-39), 64-bit
2021-04-14 16:26:12.493 UTC [28392] LOG:  listening on IPv4 address "127.0.0.1", port
5432
2021-04-14 16:26:12.497 UTC [28392] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-14 16:26:12.505 UTC [28392] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-14 16:26:12.525 UTC [28392] LOG:  redirecting log output to logging collector
process
```

```
2021-04-14 16:26:12.525 UTC [28392] HINT:  Future log output will appear in directory
"log".
 done
server started
```

7. Создаем копию кластера

```
postgres$ pg_basebackup -D /var/lib/pgsql/12/main5433 -v

pg_basebackup: initiating base backup, waiting for checkpoint to complete
pg_basebackup: checkpoint completed
pg_basebackup: write-ahead log start point: 0/2000028 on timeline 1
pg_basebackup: starting background WAL receiver
pg_basebackup: created temporary replication slot "pg_basebackup_27705"
pg_basebackup: write-ahead log end point: 0/2000100
pg_basebackup: waiting for background process to finish streaming ...
pg_basebackup: syncing data to disk ...
pg_basebackup: base backup completed
```

8. Проверяем, что созданный нами сервер не инициализован как реплика

```
postgres$ ls /var/lib/pgsql/12/main5433/

total 136
drwx------ 20 postgres postgres  4096 Apr 14 15:18 ./
drwx------  6 postgres postgres  4096 Apr 14 15:18 ../
-rw-------  1 postgres postgres   224 Apr 14 15:18 backup_label
drwx------  6 postgres postgres  4096 Apr 14 15:18 base/
-rw-------  1 postgres postgres    30 Apr 14 15:18 current_logfiles
drwx------  2 postgres postgres  4096 Apr 14 15:18 global/
drwx------  2 postgres postgres  4096 Apr 14 15:18 log/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_commit_ts/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_dynshmem/
-rw-------  1 postgres postgres  4760 Apr 14 15:18 pg_hba.conf
-rw-------  1 postgres postgres  1636 Apr 14 15:18 pg_ident.conf
drwx------  4 postgres postgres  4096 Apr 14 15:18 pg_logical/
drwx------  4 postgres postgres  4096 Apr 14 15:18 pg_multixact/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_notify/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_replslot/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_serial/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_snapshots/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_stat/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_stat_tmp/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_subtrans/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_tblspc/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_twophase/
-rw-------  1 postgres postgres     3 Apr 14 15:18 PG_VERSION
drwx------  3 postgres postgres  4096 Apr 14 15:18 pg_wal/
drwx------  2 postgres postgres  4096 Apr 14 15:18 pg_xact/
-rw-------  1 postgres postgres    88 Apr 14 15:18 postgresql.auto.conf
-rw-------  1 postgres postgres 26612 Apr 14 15:18 postgresql.conf
```

9. Запускаем кластер на порту 5433

```
postgres$ echo "port=5433" >> /var/lib/pgsql/12/main5433/postgresql.conf

postgres$ pg_ctl -D /var/lib/pgsql/12/main5433 start

waiting for server to start....2021-04-14 15:17:21.248 UTC [27721] LOG:  starting
PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-39), 64-bit
2021-04-14 15:17:21.248 UTC [27721] LOG:  listening on IPv4 address "127.0.0.1", port
5433
```

```
2021-04-14 15:17:21.252 UTC [27721] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5433"
2021-04-14 15:17:21.260 UTC [27721] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5433"
2021-04-14 15:17:21.274 UTC [27721] LOG:  redirecting log output to logging collector
process
2021-04-14 15:17:21.274 UTC [27721] HINT:  Future log output will appear in directory
"log".
 done
server started
```

10. Проверяем, что оба кластера запущены как самостоятельные сервера

```
postgres$ psql
psql (12.5)
Type "help" for help.

5432=> select pg_is_in_recovery();
 pg_is_in_recovery
-------------------
 f
(1 row)

5432=> \q

postgres$ psql -p5433
psql (12.5)
Type "help" for help.

5433=> select pg_is_in_recovery();
 pg_is_in_recovery
-------------------
 f
(1 row)

5433=> \q
```

11. Создаем публикацию на сервере 5432

```
postgres$ psql -d logical_replica
psql (12.5)
Type "help" for help.

5432=> CREATE PUBLICATION pub_tbl FOR TABLE test;
CREATE PUBLICATION

5432=> \q
```

12. На сервере 5433 создаем подписку

```
postgres$ psql -p5433 -d logical_replica
psql (12.5)
Type "help" for help.

5432=> CREATE SUBSCRIPTION sub_tbl CONNECTION 'dbname=logical_replica port=5432'
PUBLICATION pub_tbl WITH (create_slot = true, slot_name = slot5432);
CREATE SUBSCRIPTION

5432=> \q
```

13. Проверяем, что подписчик подписался на изменения

```
postgres$ psql logical_replica
```

```
psql (12.5)
Type "help" for help.

5432=> \x
Expanded display is on.

5432=> select * from pg_stat_replication;
-[ RECORD 1 ]----+------------------------------
pid              | 29998
usesysid         | 10
usename          | postgres
application_name | sub_tbl
client_addr      |
client_hostname  |
client_port      | -1
backend_start    | 2021-04-14 17:21:47.552849+00
backend_xmin     |
state            | streaming
sent_lsn         | 0/5036560
write_lsn        | 0/5036560
flush_lsn        | 0/5036560
replay_lsn       | 0/5036560
write_lag        |
flush_lag        |
replay_lag       |
sync_priority    | 0
sync_state       | async
reply_time       | 2021-04-14 17:22:48.372962+00
```

14. Теперь вставляем запись на сервере публикации в таблицу `test`

```
5432=> INSERT INTO test VALUES (3,'Value 3');
INSERT 0 1

5432=> \x
Expanded display is off.

5432=> select * from test ;
 id |   name
----+----------
  1 | Value 1
  2 | Value 2
  3 | Value 3
(3 rows)

5432=> \q
```

15. Смотрим, какая ситуация на сервере на порту 5433

```
postgres$ psql -p5433 -d logical_replica
psql (12.5)
Type "help" for help.

5433=> select * from test;
 id |   name
----+----------
  1 | Value 1
  2 | Value 2
(3 rows)

5433=> \q
```

Но почему не появилась запись? Ваше мнение.

16. Смотрим, что в логах на сервере 5433

```
postgres$ tail /var/lib/pgsql/12/main5433/log/postgresql-*

2021-04-14 17:44:28.853 UTC [30647] LOG:  logical replication table synchronization
worker for subscription "sub_tbl", table "test" has started
2021-04-14 17:44:28.892 UTC [30647] ERROR:  duplicate key value violates unique
constraint "test_pkey"
2021-04-14 17:44:28.892 UTC [30647] DETAIL:  Key (id)=(1) already exists.
2021-04-14 17:44:28.892 UTC [30647] CONTEXT:  COPY test, line 1
2021-04-14 17:44:28.894 UTC [28438] LOG:  background worker "logical replication
worker" (PID 30647) exited with exit code 1
2021-04-14 17:44:33.909 UTC [30650] LOG:  logical replication table synchronization
worker for subscription "sub_tbl", table "test" has started
2021-04-14 17:44:33.947 UTC [30650] ERROR:  duplicate key value violates unique
constraint "test_pkey"
2021-04-14 17:44:33.947 UTC [30650] DETAIL:  Key (id)=(1) already exists.
2021-04-14 17:44:33.947 UTC [30650] CONTEXT:  COPY test, line 1
2021-04-14 17:44:33.948 UTC [28438] LOG:  background worker "logical replication
worker" (PID 30650) exited with exit code 1
```

17. Стало ясно? Хорошо. Исправим эту ситуацию

```
postgres$ psql -p5433
psql (12.5)
Type "help" for help.

5433=> \c logical_replica
You are now connected to database "logical_replica" as user "postgres".

5433=> delete from test ;
```

18. Подождем немного...

```
5433=> select * from test ;
 id |   name
----+----------
  1 | Value 1
  2 | Value 2
  3 | Value 3
(3 rows)
```

Записи появились.

Внутри логической репликации

19. Давайте попробуем разобраться, что внутри логической репликации. Удаляем нашего подписчика

```
5433=> drop subscription sub_tbl ;
NOTICE:  dropped replication slot "slot5432" on publisher
DROP SUBSCRIPTION

5433=> \q
```

20. Заходим на сервер 5432 и создаем новую БД и логический слот репликации в ней

```
postgres$ psql
psql (12.5)
Type "help" for help.
```

+1

```
5432=> create database test_logical;
CREATE DATABASE

5432=> \c test_logical
You are now connected to database "test_logical" as user "postgres".

5432=> SELECT * FROM pg_create_logical_replication_slot('slot5432', 'test_decoding');
 slot_name |    lsn
-----------+-----------
 slot5432  | 0/503BF68
(1 row)
```

21. Создадим таблицу и внесем одну запись туда

```
5432=> create table test(id serial primary key,name text);
CREATE TABLE

5432=> insert into test select 1;
INSERT 0 1
```

22. Функция `pg_logical_slot_peek_changes` позволяет посмотреть, какие изменения ждут передачи по слоту репликации без их применения

```
5432=> SELECT * FROM pg_logical_slot_peek_changes('slot5432', NULL, NULL);
    lsn    | xid |                                data
-----------+-----+-----------------------------------------------------
 0/503BF98 | 505 | BEGIN 505
 0/5064C88 | 505 | COMMIT 505
 0/5064C88 | 506 | BEGIN 506
 0/5064C88 | 506 | table public.test: INSERT: id[integer]:1 name[text]:null
 0/5064D98 | 506 | COMMIT 506
(5 rows)
```

23. Функция `pg_logical_slot_get_changes` показывает то же самое, но она проигрывает эти изменения

```
5432=> SELECT * FROM pg_logical_slot_get_changes('slot5432', NULL, NULL);
    lsn    | xid |                                data
-----------+-----+-----------------------------------------------------
 0/503BF98 | 505 | BEGIN 505
 0/5064C88 | 505 | COMMIT 505
 0/5064C88 | 506 | BEGIN 506
 0/5064C88 | 506 | table public.test: INSERT: id[integer]:1 name[text]:null
 0/5064D98 | 506 | COMMIT 506
(5 rows)
```

24. То есть если вызвать еще раз, то в слоте не будет никаких изменений

```
5432=> SELECT * FROM pg_logical_slot_get_changes('slot5432', NULL, NULL);
 lsn | xid | data
-----+-----+------
(0 rows)
```

Примерно на этой основе работает утилита `wal2json`, которая декодирует изменения данных, записанные в wal, в json.

Практика: Минорное обновление отказоустойчивого кластера

Можете запустить скрипт на стенде `~/practice/lecture4/2.upgrade_cluster.sh`, который спикер запускал в видео.

Или пройтись по шагам текущей практики - https://gitlab.slurm.io/postgres/slurm_course/-/blob/main/practice/lecture4/2.upgrade_cluster.md , также представленной ниже по тексту.

Обновление контрольных сумм

1. Возвращаемся в исходное состояние. Отключаем контрольные суммы на сервере 5432 и меняем параметр `min_wal_size`

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 stop
waiting for server to shut down.... done
server stopped
postgres$ pg_checksums -D /var/lib/pgsql/12/main5432 -d
pg_checksums: syncing data directory
pg_checksums: updating control file
Checksums disabled in cluster

postgres$ echo "min_wal_size=500MB" >> /var/lib/pgsql/12/main5432/postgresql.conf

postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 start
waiting for server to start....2021-04-15 05:48:32.356 UTC [2630] LOG:  starting
PostgreSQL 12.6 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-44), 64-bit
2021-04-15 05:48:32.356 UTC [2630] LOG:  listening on IPv4 address "127.0.0.1", port
5432
2021-04-15 05:48:32.360 UTC [2630] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-15 05:48:32.368 UTC [2630] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-15 05:48:32.387 UTC [2630] LOG:  redirecting log output to logging collector
process
2021-04-15 05:48:32.387 UTC [2630] HINT:  Future log output will appear in directory
"log".
 done
server started
```

2. Удаляем кластер на порту 5433

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5433 stop
waiting for server to shut down.... done
server stopped

postgres$ rm -rf /var/lib/pgsql/12/main5433
```

3. Заново инициализируем кластер на порту 5433 с выключенными контрольными суммами, меняем порт и запускаем его

```
postgres$ pg_basebackup -D /var/lib/pgsql/12/main5433 -R -v
pg_basebackup: initiating base backup, waiting for checkpoint to complete
pg_basebackup: checkpoint completed
```

```
pg_basebackup: write-ahead log start point: 0/4000028 on timeline 1
pg_basebackup: starting background WAL receiver
pg_basebackup: created temporary replication slot "pg_basebackup_2670"
pg_basebackup: write-ahead log end point: 0/4000138
pg_basebackup: waiting for background process to finish streaming ...
pg_basebackup: syncing data to disk ...
pg_basebackup: base backup completed
postgres$ echo "port=5433" >> /var/lib/pgsql/12/main5433/postgresql.conf
postgres$ pg_ctl -D /var/lib/pgsql/12/main5433 start
waiting for server to start....2021-04-15 05:51:27.190 UTC [2690] LOG:  starting
PostgreSQL 12.6 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-44), 64-bit
2021-04-15 05:51:27.190 UTC [2690] LOG:  listening on IPv4 address "127.0.0.1", port
5433
2021-04-15 05:51:27.193 UTC [2690] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5433"
2021-04-15 05:51:27.200 UTC [2690] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5433"
2021-04-15 05:51:27.218 UTC [2690] LOG:  redirecting log output to logging collector
process
2021-04-15 05:51:27.218 UTC [2690] HINT:  Future log output will appear in directory
"log".
 done
server started
```

4. Проверяем, что репликация запущена

```
postgres$ psql
psql (12.6)
Type "help" for help.

5432=> \x
Expanded display is on.

5432=> select * from pg_stat_replication ;
-[ RECORD 1 ]----+------------------------------
pid              | 2697
usesysid         | 10
usename          | postgres
application_name | walreceiver
client_addr      |
client_hostname  |
client_port      | -1
backend_start    | 2021-04-15 05:51:27.338752+00
backend_xmin     |
state            | streaming
sent_lsn         | 0/5000060
write_lsn        | 0/5000060
flush_lsn        | 0/5000060
replay_lsn       | 0/5000060
write_lag        | 00:00:00.0837
flush_lag        | 00:00:00.084988
replay_lag       | 00:00:00.085032
sync_priority    | 0
sync_state       | async
reply_time       | 2021-04-15 05:51:27.42534+00

5432=> \q
```

5. Проверяем, что контрольные суммы на кластере 5433 выключены

```
postgres$ psql -p5433
psql (12.6)
```

```
Type "help" for help.

5433=> \x
Expanded display is on.

5433=> select * from pg_settings where name ='data_checksums';
-[ RECORD 1 ]---+-----------------------------------------------------------
name            | data_checksums
setting         | off
unit            |
category        | Preset Options
short_desc      | Shows whether data checksums are turned on for this cluster.
extra_desc      |
context         | internal
vartype         | bool
source          | override
min_val         |
max_val         |
enumvals        |
boot_val        | off
reset_val       | off
sourcefile      |
sourceline      |
pending_restart | f

5433=> \q
```

6. Останавливаем реплику, включаем контрольные суммы на ней

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5433 stop
waiting for server to shut down.... done
server stopped

postgres$ pg_checksums -D /var/lib/pgsql/12/main5433 -e
Checksum operation completed
Files scanned:  1570
Blocks scanned: 5127
pg_checksums: syncing data directory
pg_checksums: updating control file
Checksums enabled in cluster
```

7. Запускаем кластер на порту 5433

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5433 start
waiting for server to start....2021-04-15 05:52:48.870 UTC [2716] LOG:  starting
PostgreSQL 12.6 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-44), 64-bit
2021-04-15 05:52:48.870 UTC [2716] LOG:  listening on IPv4 address "127.0.0.1", port
5433
2021-04-15 05:52:48.874 UTC [2716] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5433"
2021-04-15 05:52:48.886 UTC [2716] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5433"
2021-04-15 05:52:48.903 UTC [2716] LOG:  redirecting log output to logging collector
process
2021-04-15 05:52:48.903 UTC [2716] HINT:  Future log output will appear in directory
"log".
 done
server started
```

8. Проверяем, что на реплике включены контрольные суммы и к ней приходят изменения

```
postgres$ psql -p5433
psql (12.6)
Type "help" for help.

5433=> \x
Expanded display is on.

5433=> select * from pg_settings where name ='data_checksums';
-[ RECORD 1 ]---+-----------------------------------------------------------
name            | data_checksums
setting         | on
unit            |
category        | Preset Options
short_desc      | Shows whether data checksums are turned on for this cluster.
extra_desc      |
context         | internal
vartype         | bool
source          | override
min_val         |
max_val         |
enumvals        |
boot_val        | off
reset_val       | on
sourcefile      |
sourceline      |
pending_restart | f

5433=> \q
```

9. На лидере создаем новую БД и таблицу с данными

```
postgres$ psql
psql (12.6)
Type "help" for help.

5432=> create database db_checksums;
CREATE DATABASE

5432=> \c db_checksums
You are now connected to database "db_checksums" as user "postgres".

5432=> create table tt(id int);
CREATE TABLE

5432=> insert into tt select 1;
INSERT 0 1

5432=> \q
```

10. Проверяем, что до реплики изменения доходят

```
postgres$ psql -p5433
psql (12.6)
Type "help" for help.

5433=> \c db_checksums
You are now connected to database "db_checksums" as user "postgres".

5433=> select * from tt;
 id
----
  1
(1 row)
```

```
5433=> \q
```

11. Переключаемся на реплику. Для этого делаем целую цепочку событий

12. Выполняем контрольную точку

```
postgres$ psql
psql (12.6)
Type "help" for help.

5432=> checkpoint;
5432=> \q
```

13. Останавливаем лидер

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 stop
waiting for server to shut down.... done
server stopped
```

14. Переключаем реплику на лидер

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5433 promote
waiting for server to promote.... done
server promoted
```

15. Меняем на "новом" лидере порт с 5433 на 5432

```
postgres$ echo "port=5432" >> /var/lib/pgsql/12/main5433/postgresql.conf
```

16. Перестартовываем "новый" лидер, чтобы переподключиться на порт 5432

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5433 restart
waiting for server to shut down.... done
server stopped
waiting for server to start....2021-04-15 05:55:27.910 UTC [2765] LOG:  starting
PostgreSQL 12.6 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-44), 64-bit
2021-04-15 05:55:27.910 UTC [2765] LOG:  listening on IPv4 address "127.0.0.1", port
5432
2021-04-15 05:55:27.914 UTC [2765] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-15 05:55:27.922 UTC [2765] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-15 05:55:27.943 UTC [2765] LOG:  redirecting log output to logging collector
process
2021-04-15 05:55:27.943 UTC [2765] HINT:  Future log output will appear in directory
"log".
 done
server started
```

17. Проверяем работу лидера и что контрольные суммы включены

```
postgres$ psql
psql (12.6)
Type "help" for help.

5432=> \x
Expanded display is on.

5432=> select * from pg_settings where name ='data_checksums';
-[ RECORD 1 ]---+-----------------------------------------------------------
name            | data_checksums
```

```
setting         | on
unit            |
category        | Preset Options
short_desc      | Shows whether data checksums are turned on for this cluster.
extra_desc      |
context         | internal
vartype         | bool
source          | override
min_val         |
max_val         |
enumvals        |
boot_val        | off
reset_val       | on
sourcefile      |
sourceline      |
pending_restart | f

5432=> \q
```

18. Возвращаем в работу "старый" лидер в виде реплики. Меняем порт с 5432 на 5433. Включаем контрольные суммы. Заносим информацию о "новом" лидере и создаем файл standby.signal

```
postgres$ echo "port=5433" >> /var/lib/pgsql/12/main5432/postgresql.conf

postgres$ pg_checksums -D /var/lib/pgsql/12/main5432 -e
Checksum operation completed
Files scanned:  1871
Blocks scanned: 6134
pg_checksums: syncing data directory
pg_checksums: updating control file
Checksums enabled in cluster

postgres$ cat /var/lib/pgsql/12/main5433/postgresql.auto.conf
# Do not edit this file manually!
# It will be overwritten by the ALTER SYSTEM command.
wal_level = 'logical'
primary_conninfo = 'user=postgres passfile=''/var/lib/pgsql/.pgpass'' port=5432
sslmode=prefer sslcompression=0 gssencmode=prefer krbsrvname=postgres
target_session_attrs=any'

postgres$ nano /var/lib/pgsql/12/main5432/postgresql.auto.conf

postgres$ touch /var/lib/pgsql/12/main5432/standby.signal
```

19. Включаем реплику

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 start
waiting for server to start....2021-04-15 05:58:31.689 UTC [2845] LOG:  starting
PostgreSQL 12.6 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-44), 64-bit
2021-04-15 05:58:31.690 UTC [2845] LOG:  listening on IPv4 address "127.0.0.1", port
5433
2021-04-15 05:58:31.693 UTC [2845] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5433"
2021-04-15 05:58:31.700 UTC [2845] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5433"
2021-04-15 05:58:31.717 UTC [2845] LOG:  redirecting log output to logging collector
process
2021-04-15 05:58:31.717 UTC [2845] HINT:  Future log output will appear in directory
"log".
 done
```

```
server started
```

20. Проверяем работу реплики

```
postgres$ psql -p5433
psql (12.6)
Type "help" for help.

5433=> select pg_is_in_recovery();
 pg_is_in_recovery
-------------------
 t
(1 row)

5433=> create table tt(id int);
ERROR:  cannot execute CREATE TABLE in a read-only transaction

5433=> \q
```

21. Проверяем, что работает цепочка лидер-реплика

```
postgres$ psql
psql (12.6)
Type "help" for help.

5432=> \c db_checksums
You are now connected to database "db_checksums" as user "postgres".

5432=> insert into tt select 2;
INSERT 0 1

5432=> select * from tt ;
 id
----
  1
  2
(2 rows)
5432=> \q

postgres$ psql -p5433
psql (12.6)
Type "help" for help.

5433=> \c db_checksums
You are now connected to database "db_checksums" as user "postgres".

5433=> select * from tt;
 id
----
  1
  2
(2 rows)

5433=> \q
```

8.5

Практика: Мажорное обновление отказоустойчивого кластера

Можете запустить скрипт на стенде `~/practice/lecture4/3.major_upgrade_cluster.sh`, который спикер запускал в видео.

Или пройтись по шагам текущей практики - https://gitlab.slurm.io/postgres/slurm_course/-/blob/main/practice/lecture4/3.major_upgrade_cluster.md, также представленной ниже по тексту.

Мажорное обновление кластера

1. Возвращаем всё в первоначальное состояние:

- Проигрываем кластер на порту 5433 с реплики до лидера.
- Останавливаем кластер на порту 5432.
- Меняем порт для кластера 5433 на 5432 и перезапускаем кластер, чтобы изменения применились.
- Удаляем каталог main5433, так как он нам сейчас не нужен

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 promote
waiting for server to promote.... done
server promoted

postgres$ pg_ctl -D /var/lib/pgsql/12/main5433 stop
waiting for server to shut down.... done
server stopped

postgres$ echo "port=5432" >> /var/lib/pgsql/12/main5432/postgresql.conf

postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 restart
waiting for server to shut down.... done
server stopped
waiting for server to start....2021-04-15 07:36:45.180 UTC [15725] LOG:  starting
PostgreSQL 12.6 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-44), 64-bit
2021-04-15 07:36:45.180 UTC [15725] LOG:  listening on IPv4 address "127.0.0.1", port
5432
2021-04-15 07:36:45.184 UTC [15725] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-15 07:36:45.191 UTC [15725] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-15 07:36:45.209 UTC [15725] LOG:  redirecting log output to logging collector
process
2021-04-15 07:36:45.209 UTC [15725] HINT:  Future log output will appear in directory
"log".
 done
server started
```

```
postgres$ rm -rf /var/lib/pgsql/12/main5433
```

2. Проверяем, что все у нас работает

```
postgres$ psql
psql (12.6)
Type "help" for help.

postgres=# \q
```

3. Устанавливаем новый пакет PostgreSQL 13 версии. Для этого заходим под root и добавляем новый репозиторий

```
postgres$ exit
logout

student$ sudo -i
root# yum install -y https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-
x86_64/pgdg-redhat-repo-latest.noarch.rpm

...

Installed:
  pgdg-redhat-repo.noarch 0:42.0-14

Complete!
```

4. Устанавливаем пакет Postgresql 13 версии

```
root# yum install -y postgresql13-server

...

Installed:
  postgresql13-server.x86_64 0:13.2-1PGDG.rhel7

Dependency Installed:
  postgresql13.x86_64 0:13.2-1PGDG.rhel7
postgresql13-libs.x86_64 0:13.2-1PGDG.rhel7

Complete!
```

5. Добавляем пакеты с postgresql13-contrib

```
root# yum install -y postgresql13-contrib

...

Installed:
  postgresql13-contrib.x86_64 0:13.2-1PGDG.rhel7

Complete!
```

6. Возвращаемся к пользователю postgres

```
root# exit
logout

student$ sudo -u postgres -i
```

7. Проверяем, что у нас не запущен PostgreSQL для 13 версии

```
postgres$ ps -ef | grep postgres
```

```
postgres 15725     1  0 07:36 ?        00:00:00 /usr/pgsql-12/bin/postgres -D
/var/lib/pgsql/12/main5432
postgres 15726 15725  0 07:36 ?        00:00:00 postgres: logger
postgres 15728 15725  0 07:36 ?        00:00:00 postgres: checkpointer
postgres 15729 15725  0 07:36 ?        00:00:00 postgres: background writer
postgres 15730 15725  0 07:36 ?        00:00:00 postgres: walwriter
postgres 15731 15725  0 07:36 ?        00:00:00 postgres: autovacuum launcher
postgres 15732 15725  0 07:36 ?        00:00:00 postgres: stats collector
postgres 15733 15725  0 07:36 ?        00:00:00 postgres: logical replication
launcher
root     16450  2139  0 07:46 pts/0    00:00:00 sudo -u postgres -i
postgres 16451 16450  0 07:46 pts/0    00:00:00 -bash
postgres 16492 16451  0 07:46 pts/0    00:00:00 ps -ef
postgres 16493 16451  0 07:46 pts/0    00:00:00 grep --color=auto postgres
```

8. Смотрим наш домашний каталог. У нас появилась папка для 13 версии

```
postgres$ ls ~
total 32
drwx------   5 postgres postgres 4096 Apr 15 07:45 ./
drwxr-xr-x. 27 root     root     4096 Apr 15 05:28 ../
drwx------   5 postgres postgres 4096 Apr 15 07:37 12/
drwx------   4 postgres postgres 4096 Apr 15 07:45 13/
drwx------   3 postgres postgres 4096 Apr 15 05:28 .ansible/
-rw-------   1 postgres postgres 3065 Apr 15 07:45 .bash_history
-rwx------   1 postgres postgres  266 Apr 15 07:45 .bash_profile*
-rw-------   1 postgres postgres 2392 Apr 15 07:36 .psql_history
```

9. Инициализируем кластер для PostgreSQL 13 версии

```
postgres$ /usr/pgsql-13/bin/pg_ctl -D /var/lib/pgsql/13/main5432 -o "--data-
checksums" initdb
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 enabled.

creating directory /var/lib/pgsql/13/main5432 ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... UTC
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
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:

    /usr/pgsql-13/bin/pg_ctl -D /var/lib/pgsql/13/main5432 -l logfile start
```

10. Меняем порт с 5432 на порт 5433

```
postgres$ echo "port=5433" >> /var/lib/pgsql/13/main5432/postgresql.conf
```

11. Запускаем кластер

```
postgres$ /usr/pgsql-13/bin/pg_ctl -D /var/lib/pgsql/13/main5432 start
waiting for server to start....2021-04-15 07:50:29.808 UTC [16599] LOG:  redirecting
log output to logging collector process
2021-04-15 07:50:29.808 UTC [16599] HINT:  Future log output will appear in directory
"log".
 done
server started
```

12. Проверяем, что два PostgreSQL у нас запущены на порту 5432 (12 версия) и 5433 (13 версия)

```
postgres$ ps -ef | grep postgres
postgres 15725     1  0 07:36 ?        00:00:00 /usr/pgsql-12/bin/postgres -D
/var/lib/pgsql/12/main5432
postgres 15726 15725  0 07:36 ?        00:00:00 postgres: logger
postgres 15728 15725  0 07:36 ?        00:00:00 postgres: checkpointer
postgres 15729 15725  0 07:36 ?        00:00:00 postgres: background writer
postgres 15730 15725  0 07:36 ?        00:00:00 postgres: walwriter
postgres 15731 15725  0 07:36 ?        00:00:00 postgres: autovacuum launcher
postgres 15732 15725  0 07:36 ?        00:00:00 postgres: stats collector
postgres 15733 15725  0 07:36 ?        00:00:00 postgres: logical replication
launcher
root     16450  2139  0 07:46 pts/0    00:00:00 sudo -u postgres -i
postgres 16451 16450  0 07:46 pts/0    00:00:00 -bash
postgres 16599     1  0 07:50 ?        00:00:00 /usr/pgsql-13/bin/postgres -D
/var/lib/pgsql/13/main5432
postgres 16600 16599  0 07:50 ?        00:00:00 postgres: logger
postgres 16602 16599  0 07:50 ?        00:00:00 postgres: checkpointer
postgres 16603 16599  0 07:50 ?        00:00:00 postgres: background writer
postgres 16604 16599  0 07:50 ?        00:00:00 postgres: walwriter
postgres 16605 16599  0 07:50 ?        00:00:00 postgres: autovacuum launcher
postgres 16606 16599  0 07:50 ?        00:00:00 postgres: stats collector
postgres 16607 16599  0 07:50 ?        00:00:00 postgres: logical replication
launcher
postgres 16608 16451  0 07:50 pts/0    00:00:00 ps -ef
postgres 16609 16451  0 07:50 pts/0    00:00:00 grep --color=auto postgres
```

13. Обращаемся к утилите `pg_upgrade` и запускаем ее в режиме проверки. Обратите внимание на путь к утилите `pg_upgrade`

```
postgres$ /usr/pgsql-13/bin/pg_upgrade -d /var/lib/pgsql/12/main5432/ -D
/var/lib/pgsql/13/main5432/ -b /usr/pgsql-12/bin -B /usr/pgsql-13/bin/ --link --check
Performing Consistency Checks on Old Live Server
------------------------------------------------
Checking cluster versions                                   ok
Checking database user is the install user                  ok
Checking database connection settings                       ok
Checking for prepared transactions                          ok
Checking for reg* data types in user tables                 ok
Checking for contrib/isn with bigint-passing mismatch       ok
Checking for presence of required libraries                 ok
Checking database user is the install user                  ok
Checking for prepared transactions                          ok
Checking for new cluster tablespace directories             ok

*Clusters are compatible*
```

14. Убираем ключик `--check` и уже запускаем в рабочем режиме

```
postgres$ /usr/pgsql-13/bin/pg_upgrade -d /var/lib/pgsql/12/main5432/ -D
/var/lib/pgsql/13/main5432/ -b /usr/pgsql-12/bin -B /usr/pgsql-13/bin/ --link

There seems to be a postmaster servicing the old cluster.
Please shutdown that postmaster and try again.
Failure, exiting
```

15. Появилось предупреждение, что кластер на порту 5432 нужно остановить.

Делаем это

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 stop
waiting for server to shut down.... done
server stopped
```

16. Запускаем миграцию

```
postgres$ /usr/pgsql-13/bin/pg_upgrade -d /var/lib/pgsql/12/main5432/ -D
/var/lib/pgsql/13/main5432/ -b /usr/pgsql-12/bin -B /usr/pgsql-13/bin/ --link
Performing Consistency Checks
-----------------------------
Checking cluster versions                                   ok
Checking database user is the install user                  ok
Checking database connection settings                       ok
Checking for prepared transactions                          ok
Checking for reg* data types in user tables                 ok
Checking for contrib/isn with bigint-passing mismatch       ok
Creating dump of global objects                             ok
Creating dump of database schemas
                                                            ok
Checking for presence of required libraries                 ok
Checking database user is the install user                  ok
Checking for prepared transactions                          ok
Checking for new cluster tablespace directories             ok

If pg_upgrade fails after this point, you must re-initdb the
new cluster before continuing.

Performing Upgrade
------------------
Analyzing all rows in the new cluster                       ok
Freezing all rows in the new cluster                        ok
Deleting files from new pg_xact                             ok
Copying old pg_xact to new server                           ok
Setting next transaction ID and epoch for new cluster       ok
Deleting files from new pg_multixact/offsets                ok
Copying old pg_multixact/offsets to new server              ok
Deleting files from new pg_multixact/members                ok
Copying old pg_multixact/members to new server              ok
Setting next multixact ID and offset for new cluster        ok
Resetting WAL archives                                      ok
Setting frozenxid and minmxid counters in new cluster       ok
Restoring global objects in the new cluster                 ok
Restoring database schemas in the new cluster
                                                            ok
Adding ".old" suffix to old global/pg_control               ok

If you want to start the old cluster, you will need to remove
the ".old" suffix from /var/lib/pgsql/12/main5432/global/pg_control.old.
Because "link" mode was used, the old cluster cannot be safely
started once the new cluster has been started.

Linking user relation files
```

```
                                                        ok
Setting next OID for new cluster                        ok
Sync data directory to disk                             ok
Creating script to analyze new cluster                  ok
Creating script to delete old cluster                   ok

Upgrade Complete
----------------
Optimizer statistics are not transferred by pg_upgrade so,
once you start the new server, consider running:
    ./analyze_new_cluster.sh

Running this script will delete the old cluster's data files:
    ./delete_old_cluster.sh
```

17. Скрипт сообщил, что для нас создал два файлика. Первый - сбор статистики на новом сервере. Второй - удаление старого кластера. Но вначале запустим наш новый сервер. Меняем порт с 5433 на 5432

```
postgres$ echo "port=5432" >> /var/lib/pgsql/13/main5432/postgresql.conf
```

18. Запускаем

```
postgres$ /usr/pgsql-13/bin/pg_ctl -D /var/lib/pgsql/13/main5432 start

waiting for server to start....2021-04-15 08:00:34.215 UTC [17029] LOG:  redirecting
log output to logging collector process
2021-04-15 08:00:34.215 UTC [17029] HINT:  Future log output will appear in directory
"log".
 done
server started
```

19. Проверяем, что сервер работает

```
5432=> psql
psql (13.2)
Type "help" for help.

5432=> \q
```

20. Выполняем скрипты. Скрипт `analyze_new_cluster` выполняется в три стадии, сначала собирается быстрая статистика с target равным 1, потом 10, а уже потом с полным сбором

```
postgres$ ./analyze_new_cluster.sh

This script will generate minimal optimizer statistics rapidly
so your system is usable, and then gather statistics twice more
with increasing accuracy.  When it is done, your system will
have the default level of optimizer statistics.

If you have used ALTER TABLE to modify the statistics target for
any tables, you might want to remove them and restore them after
running this script because they will delay fast statistics generation.

If you would like default statistics as quickly as possible, cancel
this script and run:
    "/usr/pgsql-13/bin/vacuumdb" --all --analyze-only

vacuumdb: processing database "db_checksums": Generating minimal optimizer statistics
(1 target)
```

```
vacuumdb: processing database "logical_replica": Generating minimal optimizer
statistics (1 target)
vacuumdb: processing database "postgres": Generating minimal optimizer statistics (1
target)
vacuumdb: processing database "template1": Generating minimal optimizer statistics (1
target)
vacuumdb: processing database "test_logical": Generating minimal optimizer statistics
(1 target)
vacuumdb: processing database "db_checksums": Generating medium optimizer statistics
(10 targets)
vacuumdb: processing database "logical_replica": Generating medium optimizer
statistics (10 targets)
vacuumdb: processing database "postgres": Generating medium optimizer statistics (10
targets)
vacuumdb: processing database "template1": Generating medium optimizer statistics (10
targets)
vacuumdb: processing database "test_logical": Generating medium optimizer statistics
(10 targets)
vacuumdb: processing database "db_checksums": Generating default (full) optimizer
statistics
vacuumdb: processing database "logical_replica": Generating default (full) optimizer
statistics
vacuumdb: processing database "postgres": Generating default (full) optimizer
statistics
vacuumdb: processing database "template1": Generating default (full) optimizer
statistics
vacuumdb: processing database "test_logical": Generating default (full) optimizer
statistics

Done
```

21. Удаляем старый каталог данных

```
postgres$ ./delete_old_cluster.sh
```

22. Проверяем, что в папке `/var/lib/pgsql/12/` папки main5432 нет, а в папке `/var/lib/pgsql/13/` есть

```
postgres$ ls ~/12/
total 20
drwx------  4 postgres postgres 4096 Apr 15 08:00 ./
drwx------  5 postgres postgres 4096 Apr 15 07:58 ../
drwx------  2 postgres postgres 4096 Feb 11 01:16 backups/
drwx------ 20 postgres postgres 4096 Apr 15 05:28 data/
-rw-------  1 postgres postgres  911 Apr 15 05:28 initdb.log
postgres$ ls ~/13/
total 20
drwx------  5 postgres postgres 4096 Apr 15 07:49 ./
drwx------  5 postgres postgres 4096 Apr 15 07:58 ../
drwx------  2 postgres postgres 4096 Feb 11 01:42 backups/
drwx------  2 postgres postgres 4096 Feb 11 01:42 data/
drwx------ 20 postgres postgres 4096 Apr 15 08:00 main5432/
```

23. Пользуемся новым сервером

```
postgres$ psql
psql (13.2)
Type "help" for help.

5432=> select version();
                                                 version
-----------------------------------------------------------------------------------------
--------------------
```

```
 PostgreSQL 13.2 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red
Hat 4.8.5-44), 64-bit
(1 row)

5432=> \q
```

8.6

Домашняя работа.
Минорное обновление кластера с помощью физической репликации

1. Создайте отказоустойчивый кластер по схеме "лидер-реплика"
2. Установите пакет с версией PostgreSQL 12.9 (может быть и старше)
3. Перезагрузите сначала реплику и убедитесь, что поменялась версия PostgreSQL. И она принимает изменения и "догнала" лидера
4. Переключите лидер на реплику
5. Перезагрузите "старый" лидер и убедитесь, что на нем поменялась версия PostgreSQL
6. Подключите "старый" лидер как реплику к "новому" лидеру

Решение по ссылке в конце файла

Решение:

1. Создайте отказоустойчивый кластер по схеме "лидер-реплика"

```
student$ sudo -u postgres -i

postgres$ pg_ctl initdb "-D" "/var/lib/pgsql/12/main5432" -o "--data-
checksums"

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 enabled.
creating directory /var/lib/pgsql/12/main5432 ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... UTC
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
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:
    /usr/pgsql-12/bin/pg_ctl -D /var/lib/pgsql/12/main5432 -l logfile start

postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 start

waiting for server to start....2021-04-24 11:17:57.923 UTC [2254] LOG:
starting PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5
20150623 (Red Hat 4.8.5-39), 64-bit
2021-04-24 11:17:57.924 UTC [2254] LOG:  listening on IPv4 address
"127.0.0.1", port 5432
2021-04-24 11:17:57.927 UTC [2254] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-24 11:17:57.935 UTC [2254] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-24 11:17:57.961 UTC [2254] LOG:  redirecting log output to logging
collector process
2021-04-24 11:17:57.961 UTC [2254] HINT:  Future log output will appear in
directory "log".
 done
server started

postgres$ pg_basebackup -D /var/lib/pgsql/12/main5433 -R -v

pg_basebackup: initiating base backup, waiting for checkpoint to complete
pg_basebackup: checkpoint completed
pg_basebackup: write-ahead log start point: 0/2000028 on timeline 1
pg_basebackup: starting background WAL receiver
pg_basebackup: created temporary replication slot "pg_basebackup_2270"
pg_basebackup: write-ahead log end point: 0/2000100
pg_basebackup: waiting for background process to finish streaming ...
pg_basebackup: syncing data to disk ...
pg_basebackup: base backup completed

postgres$ echo "port=5433" >> /var/lib/pgsql/12/main5433/postgresql.conf

postgres$ pg_ctl -D /var/lib/pgsql/12/main5433 start

waiting for server to start....2021-04-24 11:18:27.867 UTC [2274] LOG:
starting PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5
20150623 (Red Hat 4.8.5-39), 64-bit
2021-04-24 11:18:27.867 UTC [2274] LOG:  listening on IPv4 address
"127.0.0.1", port 5433
2021-04-24 11:18:27.871 UTC [2274] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5433"
2021-04-24 11:18:27.879 UTC [2274] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5433"
2021-04-24 11:18:27.896 UTC [2274] LOG:  redirecting log output to logging
collector process
```

```
2021-04-24 11:18:27.896 UTC [2274] HINT:  Future log output will appear in
directory "log".
 done
server started

postgres$ ps -ef | grep postgres

root      2197  2050  0 11:17 pts/0    00:00:00 sudo -u postgres -i
postgres  2198  2197  0 11:17 pts/0    00:00:00 -bash
postgres  2254     1  0 11:17 ?        00:00:00 /usr/pgsql-12/bin/postgres -D
/var/lib/pgsql/12/main5432
postgres  2255  2254  0 11:17 ?        00:00:00 postgres: logger
postgres  2257  2254  0 11:17 ?        00:00:00 postgres: checkpointer
postgres  2258  2254  0 11:17 ?        00:00:00 postgres: background writer
postgres  2259  2254  0 11:17 ?        00:00:00 postgres: walwriter
postgres  2260  2254  0 11:17 ?        00:00:00 postgres: autovacuum launcher
postgres  2261  2254  0 11:17 ?        00:00:00 postgres: stats collector
postgres  2262  2254  0 11:17 ?        00:00:00 postgres: logical replication
launcher
postgres  2274     1  0 11:18 ?        00:00:00 /usr/pgsql-12/bin/postgres -D
/var/lib/pgsql/12/main5433
postgres  2275  2274  0 11:18 ?        00:00:00 postgres: logger
postgres  2276  2274  0 11:18 ?        00:00:00 postgres: startup
recovering 000000010000000000000003
postgres  2277  2274  0 11:18 ?        00:00:00 postgres: checkpointer
postgres  2278  2274  0 11:18 ?        00:00:00 postgres: background writer
postgres  2279  2274  0 11:18 ?        00:00:00 postgres: stats collector
postgres  2280  2274  0 11:18 ?        00:00:00 postgres: walreceiver
streaming 0/3000060
postgres  2281  2254  0 11:18 ?        00:00:00 postgres: walsender postgres
[local] streaming 0/3000060
postgres  2282  2198  0 11:18 pts/0    00:00:00 ps -ef
postgres  2283  2198  0 11:18 pts/0    00:00:00 grep --color=auto postgres

postgres$ psql

psql (12.5)
Type "help" for help.

5432=> create database minor_upgrade;

CREATE DATABASE

5432=> \c minor_upgrade

You are now connected to database "minor_upgrade" as user "postgres".

5432=> create table tt (id serial, name text);

CREATE TABLE

5432=> insert into tt (name) select 'name1';

INSERT 0 1

5432=> insert into tt (name) select 'name2';

INSERT 0 1

5432=> \q

postgres$ psql -p5433 minor_upgrade

psql (12.5)
```

```
Type "help" for help.

5433=> select * from tt;

 id | name
----+-------
  1 | name1
  2 | name2
(2 rows)
```

2. Установите пакет с версией PostgreSQL 12.6

```
5433=> \q

postgres$ exit
logout

student$ sudo -i

root$ yum install postgresql-12

root$ sudo -u postgres -i

postgres$ psql

psql (12.9, server 12.5)
Type "help" for help.

5432=> select version();
                                                  version
-------------------------------------------------------------------------------
 PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623
(Red Hat 4.8.5-39), 64-bit
(1 row)

postgres=# \q

postgres$ psql -p5433
psql (12.9, server 12.5)
Type "help" for help.

5433=> select version();
                                                  version
-------------------------------------------------------------------------------
-----------------------------
 PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623
(Red Hat 4.8.5-39), 64-bit
(1 row)
```

3. Перезагрузите сначала реплику и убедитесь, что поменялась версия PostgreSQL. И она принимает изменения и "догнала" лидера

```
postgres=# \q

postgres$ pg_ctl -D /var/lib/pgsql/12/main5433 restart

waiting for server to shut down.... done
server stopped
waiting for server to start....2021-04-24 11:20:56.417 UTC [2446] LOG:
starting PostgreSQL 12.6 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5
20150623 (Red Hat 4.8.5-44), 64-bit
```

```
2021-04-24 11:20:56.418 UTC [2446] LOG:  listening on IPv4 address
"127.0.0.1", port 5433
2021-04-24 11:20:56.422 UTC [2446] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5433"
2021-04-24 11:20:56.430 UTC [2446] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5433"
2021-04-24 11:20:56.445 UTC [2446] LOG:  redirecting log output to logging
collector process
2021-04-24 11:20:56.445 UTC [2446] HINT:  Future log output will appear in
directory "log".
 done
server started

[postgres@vs01.s013162.slurm.io ~]$ psql -p5433

psql (12.9)
Type "help" for help.

5433=> select version();
                                                  version
-----------------------------------------------------------------------------
----------------------------
 PostgreSQL 12.9 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623
(Red Hat 4.8.5-44), 64-bit
(1 row)

postgres=# \q

postgres$ psql minor_upgrade

psql (12.9, server 12.5)
Type "help" for help.

5432=> insert into tt (name) select 'name3';
INSERT 0 1
5432=> select * from tt;
 id | name
----+-------
  1 | name1
  2 | name2
  3 | name3
(3 rows)

5432=> \q

[postgres@vs01.s000000.slurm.io ~]$ psql minor_upgrade -p5433

psql (12.9)
Type "help" for help.

5433=> select * from tt;
 id | name
----+-------
  1 | name1
  2 | name2
  3 | name3
(3 rows)
```

4. Переключите лидер на реплику

```
5433=> \q
```

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 stop && pg_ctl -D
/var/lib/pgsql/12/main5433 promote && echo "port=5432" >>
/var/lib/pgsql/12/main5433/postgresql.conf && pg_ctl -D
/var/lib/pgsql/12/main5433 restart

waiting for server to shut down.... done
server stopped
waiting for server to promote.... done
server promoted
waiting for server to shut down.... done
server stopped
waiting for server to start....2021-04-24 11:24:32.487 UTC [2541] LOG:
starting PostgreSQL 12.6 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5
20150623 (Red Hat 4.8.5-44), 64-bit
2021-04-24 11:24:32.487 UTC [2541] LOG:  listening on IPv4 address
"127.0.0.1", port 5432
2021-04-24 11:24:32.491 UTC [2541] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-24 11:24:32.505 UTC [2541] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-24 11:24:32.526 UTC [2541] LOG:  redirecting log output to logging
collector process
2021-04-24 11:24:32.526 UTC [2541] HINT:  Future log output will appear in
directory "log".
 done
server started

postgres$ psql

psql (12.9)
Type "help" for help.

5432=> select version();
                                                  version
--------------------------------------------------------------------------------
----------------------------
 PostgreSQL 12.9 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623
(Red Hat 4.8.5-44), 64-bit
(1 row)

5432=> \q
```

5. Перезагрузите "старый" лидер и убедитесь, что на нем поменялась версия PostgreSQL
6. Подключите "старый" лидер как реплику к "новому" лидеру

```
postgres$ echo "port=5433" >> /var/lib/pgsql/12/main5432/postgresql.conf

postgres$ cat /var/lib/pgsql/12/main5433/postgresql.auto.conf
# Do not edit this file manually!
# It will be overwritten by the ALTER SYSTEM command.
primary_conninfo = 'user=postgres passfile=''/var/lib/pgsql/.pgpass''
port=5432 sslmode=prefer sslcompression=0 gssencmode=prefer
krbsrvname=postgres target_session_attrs=any'

postgres$ nano /var/lib/pgsql/12/main5432/postgresql.auto.conf

postgres$ touch /var/lib/pgsql/12/main5432/standby.signal

postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 start
```

```
waiting for server to start....2021-04-24 11:26:29.575 UTC [2572] LOG:
starting PostgreSQL 12.9 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5
20150623 (Red Hat 4.8.5-44), 64-bit
2021-04-24 11:26:29.575 UTC [2572] LOG:  listening on IPv4 address
"127.0.0.1", port 5433
2021-04-24 11:26:29.579 UTC [2572] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5433"
2021-04-24 11:26:29.588 UTC [2572] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5433"
2021-04-24 11:26:29.608 UTC [2572] LOG:  redirecting log output to logging
collector process
2021-04-24 11:26:29.608 UTC [2572] HINT:  Future log output will appear in
directory "log".
 done
server started

postgres$ psql -p5433 minor_upgrade

psql (12.9)
Type "help" for help.

5433=> select version();
                                                  version
--------------------------------------------------------------------------
---------------------------
 PostgreSQL 12.9 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623
(Red Hat 4.8.5-44), 64-bit
(1 row)


5433=> \x
Expanded display is on.

5433=> select * from pg_settings where name like '%data%';

5433=> select * from pg_settings where name = 'data_directory';

-[ RECORD 1 ]---+---------------------------------
name            | data_directory
setting         | /var/lib/pgsql/12/main5432
unit            |
category        | File Locations
short_desc      | Sets the server's data directory.
extra_desc      |
context         | postmaster
vartype         | string
source          | override
min_val         |
max_val         |
enumvals        |
boot_val        |
reset_val       | /var/lib/pgsql/12/main5432
sourcefile      |
sourceline      |
pending_restart | f

5433=> show port;

-[ RECORD 1 ]
port | 5433
```