

7.2

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 - **7. Отказоустойчивые решения на основе физической репликации**, 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: Имитация сбоя основного сервера

Можете запустить скрипт на стенде `~/practice/lecture3/2.changing_master.sh`, который спикер запускал в видео.

КУПЛЕНО НА
SKLADCHIK.COM

Или пройтись по шагам текущей практики

- https://gitlab.slurm.io/postgres/slurm_course/

[/blob/main/practice/lecture3/2.changing_master.md](#), также представленной ниже по тексту.

Сбой основного сервера

1. Останавливаем сервер на порту 5434. Убираем синхронную репликацию и поменяем минимальное количество wal файлов. Для чего это сделано будет объяснено ниже

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5434 stop
postgres$ nano /var/lib/pgsql/12/main5432/postgresql.auto.conf
postgres$ echo "min_wal_size=500MB" >> ~/12/main5432/postgresql.conf
```

2. Перезапустим PostgreSQL на порту 5432, чтобы изменения вступили в силу

```
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 restart
waiting for server to stop
waiting for server to start....2021-04-12 14:01:10.615 UTC [3064] LOG:  starting
PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-44), 64-bit
2021-04-12 14:01:10.615 UTC [3064] LOG:  listening on IPv4 address "127.0.0.1", port
5432
2021-04-12 14:01:10.619 UTC [3064] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-12 14:01:10.626 UTC [3064] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-12 14:01:10.641 UTC [3064] LOG:  redirecting log output to logging collector
process
2021-04-12 14:01:10.641 UTC [3064] HINT:  Future log output will appear in directory
"log".
done
server started
```

3. Проверяем, что мы инициализировали кластер с включенными контрольными суммами

```
postgres$ psql
5432=> \x
5432=> select * from pg_settings where name like '%checksum%';
-[ 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
```

4. Проверяем, что реплика работает в режиме чтения

```
5432=> \q
postgres$ psql -p5433 -d synch_replica
5433=> select pg_is_in_recovery();
 pg_is_in_recovery
-----
 t
(1 row)
5433=> select * from test;
 id | name
----+-----
  1 | name1
  2 | name2
(2 rows)

5433=> create table t1(id int);
ERROR:  cannot execute CREATE TABLE in a read-only transaction
```

4. Теперь предположим, что у нас моргнула сеть и наше выбранное решение, обеспечивающее высокодоступный кластер PostgreSQL, решило сделать реплику лидером и исключить "бывшего" лидера из пула доступных адресов. Промоутируем нашу реплику на порту 5433 до лидера

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

5. Проверим, что реплика стала лидером

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

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

6. Но как всегда бывает, у нас за время принятия решения остались соединения, которые успели сделать изменения в лидере. Протремаем их

```
postgres$ psql -d synch_replica
You are now connected to database "synch_replica" as user "postgres".

5432=> insert into test values(3,'name3');
INSERT 0 1

5432=> select * from test;
 id | name
----+-----
  1 | name1
  2 | name2
  3 | name3
(3 rows)
```

7. А теперь на "новом" лидере удалим записи из таблицы test

```
5432=> \q
```

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

```
5433=> delete from test;
DELETE 2
```

```
5433=> select * from test;
 id | name
----+-----
(0 rows)
```

8. Сейчас мы получаем "нехорошую" ситуацию. У нас два лидера и оба могут принимать подключения и менять данные.

Давайте избавимся от этого. Для этого остановим "бывший" лидер на порту 5432

```
5433=> \q
```

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

9. Запустим утилиту `pg_rewind`, которая прочитает все наши файлы `wal` из источника (сервер на порту 5433) и проиграт эти изменения на порту 5432

```
postgres$ pg_rewind -D /var/lib/pgsql/12/main5432 --source-server="port=5433" -P
pg_rewind: connected to server
pg_rewind: servers diverged at WAL location 0/3000610 on timeline 1
pg_rewind: rewinding from last common checkpoint at 0/2000060 on timeline 1
pg_rewind: reading source file list
pg_rewind: reading target file list
pg_rewind: reading WAL in target
pg_rewind: need to copy 53 MB (total source directory size is 80 MB)
54325/54325 kB (100%) copied
pg_rewind: creating backup label and updating control file
pg_rewind: syncing target data directory
pg_rewind: Done!
```

10. Всё, утилита отработала. Но есть проблема - для её работы нужны `wal` файлы предыдущей контрольной точки. Поэтому мы в начале ставили большой `min_wal_size` и запустить наш "бывший" лидер просто так не сможем. В папке `main5432` теперь конфигурация с папки 5433. Меняем порт и запускаем сервер

```
postgres$ echo "port=5432" >> /var/lib/pgsql/12/main5432/postgresql.conf
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 start
waiting for server to start....2021-04-12 13:57:59.888 UTC [3014] LOG:  starting
PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-44), 64-bit
2021-04-12 14:03:59.888 UTC [3014] LOG:  listening on IPv4 address "127.0.0.1", port
5432
2021-04-12 14:03:59.892 UTC [3014] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-12 14:03:59.899 UTC [3014] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-12 14:03:59.916 UTC [3014] LOG:  redirecting log output to logging collector
process
2021-04-12 14:03:59.916 UTC [3014] HINT:  Future log output will appear in directory
"log".
done
```

```
server started
```

11. Проверим данные в таблице test

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

```
5432=> select * from test;
 id | name
----+-----
(0 rows)
5432=>/q
```

Кластер на порту 5432 проиграл wal файлы с "нового" лидера. Теперь давайте поднимем на нем реплику.

12. Подключаем сервер на порту 5432 к "новому" лидеру

```
postgres$ psql -p5433
5433=> SELECT pg_create_physical_replication_slot('synch_slot5432');
 pg_create_physical_replication_slot
-----
(synch_slot5432,)
(1 row)
```

13. Меняем строку подключения к мастеру на сервере 5432.

Нам нужно подключаться теперь не к серверу на порту 5432, а к серверу на порту 5433 и слот репликации тоже нужно поменять с `synch_slot5433` на `synch_slot5432`

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

14. Создаем файл, указывающий что наш сервер теперь работает в режиме реплики, и перезагружаем сервер на порту 5432

```
postgres$ touch /var/lib/pgsql/12/main5432/standby.signal
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 restart
waiting for server to shut down..... done
server stopped
waiting for server to start....2021-04-12 15:57:20.918 UTC [2840] LOG:  starting
PostgreSQL 12.5 on x86_64-pc-linux-gnu, compiled by gcc (GCC) 4.8.5 20150623 (Red Hat
4.8.5-44), 64-bit
2021-04-12 15:57:20.918 UTC [2840] LOG:  listening on IPv4 address "127.0.0.1", port
5432
2021-04-12 15:57:20.922 UTC [2840] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-12 15:57:20.930 UTC [2840] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-12 15:57:20.946 UTC [2840] LOG:  redirecting log output to logging collector
process
2021-04-12 15:57:20.946 UTC [2840] HINT:  Future log output will appear in directory
"log".
done
server started
```

15. Проверяем

```
postgres$ psql
psql (12.5)
Type "help" for help.
5432=> select pg_is_in_recovery();
 pg_is_in_recovery
```

```

-----
t
(1 row)
5432=> \q
postgres$ psql -p5433
5433=> \x
5433=> select * from pg_stat_replication;
-[ RECORD 1 ]-----+-----
pid                | 2979
usesysid           | 10
username           | postgres
application_name   | walreceiver
client_addr        |
client_hostname    |
client_port        | -1
backend_start      | 2021-04-12 16:01:05.972088+00
backend_xmin       |
state              | streaming
sent_lsn           | 0/3024F60
write_lsn          | 0/3024F60
flush_lsn          | 0/3030CE0
replay_lsn         | 0/3030CE0
write_lag          | 00:00:00.001146
flush_lag          | 00:00:00.001146
replay_lag         | 00:00:00.001146
sync_priority      | 0
sync_state         | async
reply_time         | 2021-04-12 16:03:36.304406+00

```

7.3

Домашнее задание.

Сбой репликации и восстановления отказоустойчивого решения" (Вариант 1)

1. Настроить асинхронную репликацию с архивом. Остановить реплику. На лидере внести изменения и checkpoint, так чтобы посматстер сгенерил wal файл и чекпоинт удалил старые wal файлы.
2. Запустить реплику и посмотреть логи postgresql. Убедится, что реплика "отстала" и не может присоединиться к лидеру.
3. В конфиги реплики добавить команду восстановления из архива и перезапустить реплику.
4. Убедится, что все wal файлы были получены и реплика "догнала" лидер.

Чек-лист домашнего задания лекция 7-1

1. Настроена асинхронная репликация с архивом. Остановлена реплика.
2. Запущена реплика и она отстает от лидера.
3. В конфигурации реплики добавлена команда восстановления из архива;
4. Получено подтверждение, что реплика "догнала" лидер.

Решение по ссылке в [конце файла](#)

Решение:

1. Настроить асинхронную репликацию с архивом. Остановить реплику. На лидере внести изменения и checkpoint, так чтобы посматстер сгенерил wal файл и чекпоинт удалил старые wal файлы.

```
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 12:24:25.308 UTC [2265] 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 12:24:25.309 UTC [2265] LOG: listening on IPv4 address
"127.0.0.1", port 5432
2021-04-24 12:24:25.313 UTC [2265] LOG: listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-24 12:24:25.321 UTC [2265] LOG: listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-24 12:24:25.334 UTC [2265] LOG: redirecting log output to logging
collector process
2021-04-24 12:24:25.334 UTC [2265] HINT: Future log output will appear in
directory "log".
done
server started
postgres$ echo "wal_level=replica" >>
/var/lib/pgsql/12/main5432/postgresql.auto.conf
postgres$ echo "archive_mode=on" >>
/var/lib/pgsql/12/main5432/postgresql.auto.conf
postgres$ echo "archive_command='cp %p /var/lib/pgsql/12/backups/wal/%f >>
/var/lib/pgsql/12/main5432/log/archive_command.log 2>&1' " >>
/var/lib/pgsql/12/main5432/postgresql.auto.conf
postgres$ echo "restore_command='cp /var/lib/pgsql/12/backups/wal/%f %p >>
/var/lib/pgsql/12/main5432/log/restore_command.log 2>&1' " >>
/var/lib/pgsql/12/main5432/postgresql.auto.conf
postgres$ cat ~/12/main5432/postgresql.auto.conf
# Do not edit this file manually!
# It will be overwritten by the ALTER SYSTEM command.
wal_level=replica
archive_mode=on
archive_command='cp %p /var/lib/pgsql/12/backups/wal/%f >>
/var/lib/pgsql/12/main5432/log/archive_command.log 2>&1'
restore_command='cp /var/lib/pgsql/12/backups/wal/%f %p >>
/var/lib/pgsql/12/main5432/log/restore_command.log 2>&1'
postgres$ mkdir /var/lib/pgsql/12/backups/wal
postgres$ pg_ctl -D /var/lib/pgsql/12/main5432 restart
waiting for server to shut down.... done
server stopped
waiting for server to start....2021-04-24 12:25:17.695 UTC [2283] 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 12:25:17.695 UTC [2283] LOG:  listening on IPv4 address
"127.0.0.1", port 5432
2021-04-24 12:25:17.699 UTC [2283] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5432"
2021-04-24 12:25:17.707 UTC [2283] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5432"
2021-04-24 12:25:17.725 UTC [2283] LOG:  redirecting log output to logging
collector process
2021-04-24 12:25:17.725 UTC [2283] HINT:  Future log output will appear in
directory "log".
done
server started
postgres$ psql
psql (12.6)
Type "help" for help.

5432=> select pg_switch_wal();
 pg_switch_wal
-----
 0/18C2258
(1 row)

5432=> select pg_switch_wal();
 pg_switch_wal
-----
 0/2000000
(1 row)
5432=>\q
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: 0/5000028 on timeline 1
pg_basebackup: starting background WAL receiver
pg_basebackup: created temporary replication slot "pg_basebackup_2312"
pg_basebackup: write-ahead log end point: 0/5000100
pg_basebackup: waiting for background process to finish streaming ...
pg_basebackup: syncing data to disk ...
pg_basebackup: base backup completed
postgres$ sed -i 's/#port = 5432/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 12:26:07.427 UTC [2326] 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 12:26:07.427 UTC [2326] LOG:  listening on IPv4 address
"127.0.0.1", port 5433
2021-04-24 12:26:07.431 UTC [2326] LOG:  listening on Unix socket
"/var/run/postgresql/.s.PGSQL.5433"
2021-04-24 12:26:07.439 UTC [2326] LOG:  listening on Unix socket
"/tmp/.s.PGSQL.5433"
2021-04-24 12:26:07.452 UTC [2326] LOG:  redirecting log output to logging
collector process
2021-04-24 12:26:07.452 UTC [2326] HINT:  Future log output will appear in
directory "log".
done
server started
postgres$ psql -p5433
psql (12.6)
Type "help" for help.

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

```

(1 row)

5433=> \x

Expanded display is on.

5433=> select * from pg_stat_wal_receiver ;

-[RECORD 1]-----+-----

```
pid                | 2338
status             | streaming
receive_start_lsn | 0/6000000
receive_start_tli | 1
received_lsn       | 0/6000060
received_tli       | 1
last_msg_send_time | 2021-04-24 12:26:07.619552+00
last_msg_receipt_time | 2021-04-24 12:26:07.619629+00
latest_end_lsn     | 0/6000060
latest_end_time    | 2021-04-24 12:26:07.619552+00
slot_name          |
sender_host        | /var/run/postgresql
sender_port        | 5432
conninfo           | user=postgres passfile=/var/lib/pgsql/.pgpass
dbname=replication port=5432 fallback_application_name=walreceiver
sslmode=prefer sslcompression=0 gssencmode=prefer krbsrvname=postgres
target_session_attrs=any
```

5433=> \q

postgres\$ psql

psql (12.6)

Type "help" for help.

5432=> \x

Expanded display is on.

5432=> select * from pg_stat_replication;

-[RECORD 1]-----+-----

```
pid                | 2339
usesysid           | 10
username           | postgres
application_name   | walreceiver
client_addr        |
client_hostname    |
client_port        | -1
backend_start      | 2021-04-24 12:26:07.617048+00
backend_xmin       |
state              | streaming
sent_lsn           | 0/6000060
write_lsn          | 0/6000060
flush_lsn          | 0/6000060
replay_lsn         | 0/6000060
write_lag          |
flush_lag          |
replay_lag         |
sync_priority      | 0
sync_state         | async
reply_time         | 2021-04-24 12:26:27.734615+00
```

7.4

Домашнее задание.

Сбой репликации и восстановления отказоустойчивого решения" (Вариант 2)

1. Создать каскадную репликацию по следующей схеме лидер -> реплика -> реплика.
2. Схема "лидер -> реплика" - это потоковая репликация со слотом репликации
3. Схема "реплика-реплика" - это репликация с архивом
4. Промоутить первую реплику до лидера
5. Подключить "бывший" лидер к "новому" лидеру
6. Проверить, что вторая реплика принимает изменения с лидера

Чек-лист

1. Создана каскадная репликация по следующей схеме лидер -> реплика -> реплика;
2. Схема "лидер -> реплика" - это потоковая репликация со слотом репликации;
3. Схема "реплика-реплика" - это репликация с архивом;
4. Переключена первая реплика до лидера;
5. Подключен "бывший" лидер к "новому" лидер;
6. Проверена, что вторая реплика принимает изменения с лидера.

Решение по ссылке в [конце файла](#)

7.5

Практика: Высокодоступный кластер

Можете запустить скрипт на стенде `~/practice/lecture3/patroni.sh`, который спикер запускал в видео.

Или пройтись по шагам текущей практики

- https://gitlab.slurm.io/postgres/slurm_course/-

</blob/main/practice/lecture3/3.patroni.md>, также представленной ниже по тексту.

Установка etcd

ЗАМЕЧАНИЕ!!!

Далее по тексту вместо переменных

- `{HOST1_IP}` нужно вставить ip vs01

- `{HOST2_IP}` нужно вставить ip vs02

- `{HOST3_IP}` нужно вставить ip vs03

1. Устанавливаем требуемые пакеты

```
vs01$ yum install etcd patroni patroni-etcd -y
```

```
vs02$ yum install patroni patroni-etcd -y
```

```
vs03$ yum install patroni patroni-etcd -y
```

2. Проверяем установку

```
vs01$ yum list installed etcd patroni patroni-etcd
```

```
Loaded plugins: fastestmirror, protectbase
```

```
Installed Packages
```

etcd.x86_64	3.3.11-2.el7.centos	@extras
patroni.x86_64	2.1.1-1.rhel7	@pgdg-common
patroni-etcd.x86_64	2.1.1-1.rhel7	@pgdg-common

```
vs02$ yum list installed patroni patroni-etcd
```

```
Loaded plugins: fastestmirror, protectbase
```

```
Installed Packages
```

patroni.x86_64	2.1.1-1.rhel7	@pgdg-common
patroni-etcd.x86_64	2.1.1-1.rhel7	@pgdg-common

```
vs03$ yum list installed patroni patroni-etcd
```

```
Loaded plugins: fastestmirror, protectbase
```

```
Installed Packages
```

patroni.x86_64	2.1.1-1.rhel7	@pgdg-common
patroni-etcd.x86_64	2.1.1-1.rhel7	@pgdg-common

3. Кроме etcd patroni поддерживает и другие распределенные хранилища.

```
vs01$ yum search patroni
```

```
Loaded plugins: fastestmirror, protectbase
```

```
Loading mirror speeds from cached hostfile
```

```
* epel: fedora-epel.koyanet.lv
```

```
0 packages excluded due to repository protections
```

```
===== N/S matched: patroni =====
```

```
patroni-aws.x86_64 : Related components to use patroni on AWS
patroni-consul.x86_64 : Related components to use patroni with Consul
patroni-etcd.x86_64 : Related components to use patroni with etcd
patroni-zookeeper.x86_64 : Related components to use patroni with Zookeeper
patroni.x86_64 : A Template for PostgreSQL HA with ZooKeeper, etcd or Consul
```

Name and summary matches only, use "search all" for everything.

4. Настраиваем etcd. Заменяем все упоминания localhost на IP сервера. А в параметре `ETCD_LISTEN_CLIENT_URLS` ставим слушать локальный адрес и IP адрес

```
vs01$ sed -i 's/localhost/{HOST1_IP}/' /etc/etcd/etcd.conf
```

```
vs01$ sed -i
's/ETCD_LISTEN_CLIENT_URLS=\"http:///${HOST1_IP}:2379\"/ETCD_LISTEN_CLIENT_URLS=\"ht
tp:///${HOST1_IP}:2379,http://localhost:2379\"/' /etc/etcd/etcd.conf
```

ЗАМЕЧАНИЕ!!! Вместо переменной `${HOST1_IP}` нужно поставить ip адреса сервера vs01!!!

5. Проверяем, что заменили правильно

```
vs01$ cat /etc/etcd/etcd.conf | grep ETCD_LISTEN_CLIENT_URLS
ETCD_LISTEN_CLIENT_URLS="http://172.25.27.10:2379,http://localhost:2379"
```

6. Запускаем службу etcd

```
vs01$ service etcd start
Redirecting to /bin/systemctl start etcd.service
```

7. Ждем 10 секунд, когда запустится сервис и проверяем его статус

```
vs01$ sleep 10
vs01$ service etcd status
Redirecting to /bin/systemctl status etcd.service
• etcd.service - Etcd Server
  Loaded: loaded (/usr/lib/systemd/system/etcd.service; disabled; vendor preset:
disabled)
  Active: active (running) since Thu 2021-09-23 17:47:27 UTC; 16s ago
  Main PID: 2210 (etcd)
  CGroup: /system.slice/etcd.service
          └─2210 /usr/bin/etcd --name=default --data-dir=/var/lib/etcd/default.etcd
--listen-client-urls=http://172.25.27.10:2379,http://localhost:2379

Sep 23 17:47:27 vs01.s013162.slurm.io etcd[2210]: 8e9e05c52164694d became leader at
term 2
Sep 23 17:47:27 vs01.s013162.slurm.io etcd[2210]: raft.node: 8e9e05c52164694d elected
leader 8e9e05c52164694d at term 2
Sep 23 17:47:27 vs01.s013162.slurm.io etcd[2210]: setting up the initial cluster
version to 3.3
Sep 23 17:47:27 vs01.s013162.slurm.io etcd[2210]: set the initial cluster version to
3.3
Sep 23 17:47:27 vs01.s013162.slurm.io etcd[2210]: enabled capabilities for version
3.3
Sep 23 17:47:27 vs01.s013162.slurm.io etcd[2210]: published {Name:default
ClientURLs:[http://172.25.27.10:2379]} to cluster cdf818194e3a8c32
Sep 23 17:47:27 vs01.s013162.slurm.io etcd[2210]: ready to serve client requests
Sep 23 17:47:27 vs01.s013162.slurm.io etcd[2210]: serving insecure client requests on
172.25.27.10:2379, this is strongly discouraged!
Sep 23 17:47:27 vs01.s013162.slurm.io etcd[2210]: ready to serve client requests
Sep 23 17:47:27 vs01.s013162.slurm.io etcd[2210]: serving insecure client requests on
127.0.0.1:2379, this is strongly discouraged!
```

Установка patroni на vs01

8. Настраиваем конфигурацию patroni. Создаем папку `/etc/patroni` и файл `/etc/patroni/patroni.yml`

```
vs01$ rm -rf /etc/patroni
vs01$ mkdir /etc/patroni
vs01$ touch /etc/patroni/patroni.yml
```

9. Вносим в конфигурационный файл `/etc/patroni/patroni.yml` следующие данные

```

vs01$ echo 'scope: cluster
#namespace: /service/
name: postgresql0

restapi:
  listen: ${HOST1_IP}:8008
  connect_address: ${HOST1_IP}:8008

etcd:
  #Provide host to do the initial discovery of the cluster topology:
  host: ${HOST1_IP}:2379
  #Or use "hosts" to provide multiple endpoints
  #Could be a comma separated string:
  #hosts: host1:port1,host2:port2
  #or an actual yaml list:
  #hosts:
  #- host1:port1
  #- host2:port2
  #Once discovery is complete Patroni will use the list of advertised clientURLs
  #It is possible to change this behavior through by setting:
  #use_proxies: true

bootstrap:
  # this section will be written into Etcd:/<namespace>/<scope>/config after
  # initializing new cluster
  # and all other cluster members will use it as a ''global configuration''
  dcs:
    ttl: 30
    loop_wait: 10
    retry_timeout: 10
    maximum_lag_on_failover: 1048576
  #   master_start_timeout: 300
  #   synchronous_mode: false
  #standby_cluster:
  #  #host: 127.0.0.1
  #  #port: 1111
  #  #primary_slot_name: patroni
  postgresql:
    use_pg_rewind: true
  #   use_slots: true
    parameters:
  #     wal_level: hot_standby
  #     hot_standby: "on"
  #     max_connections: 100
  #     max_worker_processes: 8
  #     wal_keep_segments: 8
  #     max_wal_senders: 10
  #     max_replication_slots: 10
  #     max_prepared_transactions: 0
  #     max_locks_per_transaction: 64
  #     wal_log_hints: "on"
  #     track_commit_timestamp: "off"
  #     archive_mode: "on"
  #     archive_timeout: 1800s
  #     archive_command: mkdir -p ../wal_archive && test ! -f ../wal_archive/%f &&
  # cp %p ../wal_archive/%f
  #     recovery_conf:
  #       restore_command: cp ../wal_archive/%f %p

  # some desired options for ''initdb''
  initdb: # Note: It needs to be a list (some options need values, others are
  switches)

```

```

- encoding: UTF8
- data-checksums

pg_hba: # Add following lines to pg_hba.conf after running 'initdb'
# For kerberos gss based connectivity
#- host replication replicator 127.0.0.1/32 gss include_realm=0
#- host all all 0.0.0.0/0 gss include_realm=0
- host replication replicator 127.0.0.1/32 md5
- host replication replicator 0.0.0.0/0 md5
- host all all 0.0.0.0/0 md5
# - hostssl all all 0.0.0.0/0 md5

# Additional script to be launched after initial cluster creation (will be passed
the connection URL as parameter)
# post_init: /usr/local/bin/setup_cluster.sh

# Some additional users users which needs to be created after initializing new
cluster
users:
  admin:
    password: admin
    options:
      - createrole
      - createdb

postgresql:
  listen: ${HOST1_IP}:5432
  connect_address: ${HOST1_IP}:5432
  data_dir: /var/lib/pgsql/12/main
  bin_dir: /usr/pgsql-12/bin/
# config_dir:
pgpass: /tmp/pgpass0
authentication:
  replication:
    username: replicator
    password: replicator
  superuser:
    username: postgres
    password: postgres
  rewind: # Has no effect on postgres 10 and lower
    username: postgres
    password: postgres
# Server side kerberos spn
# krbsrvname: postgres
parameters:
  # Fully qualified kerberos ticket file for the running user
  # same as KRB5CCNAME used by the GSS
# krb_server_keyfile: /var/spool/keytabs/postgres
  unix_socket_directories: ''.'''
# Additional fencing script executed after acquiring the leader lock but before
promoting the replica
#pre_promote: /path/to/pre_promote.sh

#watchdog:
# mode: automatic # Allowed values: off, automatic, required
# device: /dev/watchdog
# safety_margin: 5

tags:
  nofailover: false
  noloadbalance: false
  clonefrom: false

```

```
nosync: false' > /etc/patroni/patroni.yml
```

ЗАМЕЧАНИЕ!!! Вместо переменной `{HOST1_IP}` нужно поставить ip адреса сервера vs01!!!

10. Проверяем, что ошибок при создании и внесении информации в файл `/etc/patroni/patroni.yml` нет

```
vs01$ ls -la /etc/patroni/patroni.yml
-rw-r--r-- 1 root root 3653 Sep 23 17:50 /etc/patroni/patroni.yml
```

11. Запускаем сервер patroni

```
vs01$ service patroni start
Redirecting to /bin/systemctl start patroni.service
```

12. Ждем 20 секунд и проверяем статус сервера patroni

```
vs01$ sleep 20
vs01$ service patroni status
Redirecting to /bin/systemctl status patroni.service
• patroni.service - Runners to orchestrate a high-availability PostgreSQL
  Loaded: loaded (/usr/lib/systemd/system/patroni.service; disabled; vendor preset: disabled)
  Active: active (running) since Thu 2021-09-23 17:50:58 UTC; 26s ago
  Main PID: 2351 (patroni)
  CGroup: /system.slice/patroni.service
          └─2351 /usr/bin/python3 /usr/bin/patroni /etc/patroni/patroni.yml
            └─2386 /usr/pgsql-12/bin/postgres -D /var/lib/pgsql/12/main --config-file=/var/lib/pgsql/12/main/postgresql.conf --listen_addresses=172.25.27.10 --port=5432 --cluster_name=cluster --wal_level=replica --hot_standby=on --max_connections=100 --max_wal_senders=10 --max_prepared_transactions=0 --max_locks_per_transaction=64 --track_commit_timestamp=off --max_replication_slots=10 --max_worker_processes=8 --wal_log_hints=on
              └─2388 postgres: cluster: logger
                └─2393 postgres: cluster: checkpointer
                  └─2394 postgres: cluster: background writer
                    └─2395 postgres: cluster: walwriter
                      └─2396 postgres: cluster: autovacuum launcher
                        └─2397 postgres: cluster: stats collector
                          └─2398 postgres: cluster: logical replication launcher
                            └─2401 postgres: cluster: postgres postgres 172.25.27.10(50054) idle

Sep 23 17:51:02 vs01.s013162.slurm.io patroni[2351]: 172.25.27.10:5432 - rejecting connections
Sep 23 17:51:02 vs01.s013162.slurm.io patroni[2351]: 172.25.27.10:5432 - rejecting connections
Sep 23 17:51:03 vs01.s013162.slurm.io patroni[2351]: 172.25.27.10:5432 - accepting connections
Sep 23 17:51:03 vs01.s013162.slurm.io patroni[2351]: 2021-09-23 17:51:03,939 INFO: establishing a new patroni connection to the postgres cluster
Sep 23 17:51:03 vs01.s013162.slurm.io patroni[2351]: 2021-09-23 17:51:03,952 INFO: running post_bootstrap
Sep 23 17:51:03 vs01.s013162.slurm.io patroni[2351]: 2021-09-23 17:51:03,964 WARNING: Could not activate Linux watchdog device: "Can't open watchdog device: [Errno 2] No such file or directory: '/dev/watchdog'"
Sep 23 17:51:03 vs01.s013162.slurm.io patroni[2351]: 2021-09-23 17:51:03,977 INFO: initialized a new cluster
Sep 23 17:51:13 vs01.s013162.slurm.io patroni[2351]: 2021-09-23 17:51:13,977 INFO: no action. I am (postgresql0) the leader with the lock
```

```
Sep 23 17:51:14 vs01.s013162.slurm.io patroni[2351]: 2021-09-23 17:51:14,018 INFO: no
action. I am (postgresql0) the leader with the lock
Sep 23 17:51:24 vs01.s013162.slurm.io patroni[2351]: 2021-09-23 17:51:24,019 INFO: no
action. I am (postgresql0) the leader with the lock
```

13. Запросим версию PostgreSQL и проверим его работоспособность

```
vs01$ sudo -u postgres -i PGPASSWORD=postgres psql -h ${HOST1_IP} -c "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)
```

ЗАМЕЧАНИЕ!!! Вместо переменной `${HOST1_IP}` нужно поставить ip адреса сервера vs01!!!

Установка patroni на vs02

14. Создаем директорию `/etc/patroni` на втором сервере

```
vs02$ mkdir /etc/patroni
```

15. Копируем конфигурационный файл `patroni.yml` с первого сервера

```
vs02$ scp ${HOST1_IP}:/etc/patroni/patroni.yml /etc/patroni/patroni.yml
```

ЗАМЕЧАНИЕ!!! Вместо переменной `${HOST1_IP}` нужно поставить ip адреса сервера vs01!!!

16. Заменяем имя сервера с `postgresql0` на `postgresql1`

```
vs02$ sed -i 's/name: postgresql0/name: postgresql1/' /etc/patroni/patroni.yml
vs02$ sed -i 's/${HOST1_IP}:5432/${HOST2_IP}:5432/' /etc/patroni/patroni.yml
vs02$ sed -i 's/listen: ${HOST1_IP}:8008/listen: ${HOST2_IP}:8008/'
/etc/patroni/patroni.yml
vs02$ sed -i 's/connect_address: ${HOST1_IP}:8008/connect_address: ${HOST2_IP}:8008/'
/etc/patroni/patroni.yml
```

ЗАМЕЧАНИЕ!!! Вместо переменной `${HOST1_IP}` нужно поставить ip адреса сервера vs01. Вместо переменной `${HOST2_IP}` нужно поставить ip адреса сервера vs02!!!

17. Проверяем, что замена сработала

```
vs02$ cat /etc/patroni/patroni.yml | grep 'name: postgresql1'
vs02$ cat /etc/patroni/patroni.yml | grep '${HOST2_IP}:5432'
name: postgresql1
listen: 172.25.27.11:5432
connect_address: 172.25.27.11:5432
```

ЗАМЕЧАНИЕ!!! Вместо переменной `${HOST2_IP}` нужно поставить ip адреса сервера vs02!!!

18. Запускаем сервер `patroni` и проверяем статус

```

vs02$ service patroni start
vs02$ sleep 20
vs02$ service patroni status
Redirecting to /bin/systemctl start patroni.service
Redirecting to /bin/systemctl status patroni.service
• patroni.service - Runners to orchestrate a high-availability PostgreSQL
  Loaded: loaded (/usr/lib/systemd/system/patroni.service; disabled; vendor preset:
disabled)
  Active: active (running) since Thu 2021-09-23 17:59:15 UTC; 20s ago
  Main PID: 2274 (patroni)
  CGroup: /system.slice/patroni.service
          └─2274 /usr/bin/python3 /usr/bin/patroni /etc/patroni/patroni.yml
            └─2292 /usr/pgsql-12/bin/postgres -D /var/lib/pgsql/12/main --config-
file=/var/lib/pgsql/12/main/postgresql.conf --listen_addresses=172.25.27.11 --
port=5432 --cluster_name=cluster --wal_level=replica --hot_standby=on --
max_connections=100 --max_wal_senders=10 --max_prepared_transactions=0 --
max_locks_per_transaction=64 --track_commit_timestamp=off --max_replication_slots=10
--max_worker_processes=8 --wal_log_hints=on
            └─2294 postgres: cluster: logger
            └─2295 postgres: cluster: startup   recovering 00000001000000000000000003
            └─2299 postgres: cluster: checkpointer
            └─2300 postgres: cluster: background writer
            └─2301 postgres: cluster: stats collector
            └─2307 postgres: cluster: postgres postgres 172.25.27.11(41180) idle
            └─2309 postgres: cluster: walreceiver   streaming 0/3000060

```

```

Sep 23 17:59:19 vs02.s013162.slurm.io patroni[2274]: 2021-09-23 17:59:19.831 UTC
[2292] LOG: redirecting log output to logging collector process
Sep 23 17:59:19 vs02.s013162.slurm.io patroni[2274]: 2021-09-23 17:59:19.831 UTC
[2292] HINT: Future log output will appear in directory "log".
Sep 23 17:59:19 vs02.s013162.slurm.io patroni[2274]: 172.25.27.11:5432 - rejecting
connections
Sep 23 17:59:19 vs02.s013162.slurm.io patroni[2274]: 172.25.27.11:5432 - rejecting
connections
Sep 23 17:59:20 vs02.s013162.slurm.io patroni[2274]: 172.25.27.11:5432 - accepting
connections
Sep 23 17:59:20 vs02.s013162.slurm.io patroni[2274]: 2021-09-23 17:59:20,877 INFO:
Lock owner: postgresql0; I am postgresql1
Sep 23 17:59:20 vs02.s013162.slurm.io patroni[2274]: 2021-09-23 17:59:20,877 INFO:
establishing a new patroni connection to the postgres cluster
Sep 23 17:59:20 vs02.s013162.slurm.io patroni[2274]: 2021-09-23 17:59:20,902 INFO: no
action. I am a secondary (postgresql1) and following a leader (postgresql0)
Sep 23 17:59:24 vs02.s013162.slurm.io patroni[2274]: 2021-09-23 17:59:24,027 INFO: no
action. I am a secondary (postgresql1) and following a leader (postgresql0)
Sep 23 17:59:34 vs02.s013162.slurm.io patroni[2274]: 2021-09-23 17:59:34,027 INFO: no
action. I am a secondary (postgresql1) and following a leader (postgresql0)

```

19. Проверяем, что PostgreSQL подцепился репликой

```

vs02$ sudo -u postgres -i PGPASSWORD=postgres psql -h ${HOST2_IP} -c "select
version();"
 pid | status | receive_start_lsn | receive_start_tli | received_lsn |
received_tli | last_msg_send_time | last_msg_receipt_time |
latest_end_lsn | latest_end_time | slot_name | sender_host |
sender_port |
conninfo
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

КУПЛЕНО НА
SKLADCHIK.COM

```

2309 | streaming | 0/3000000 | | 1 | 0/3000060 |
1 | 2021-09-23 17:59:24.939901+00 | 2021-09-23 17:59:24.936167+00 | 0/3000060 |
2021-09-23 17:59:24.939901+00 | postgresql1 | 172.25.27.10 | 5432 |
user=replicator passfile=/tmp/pgpass0 dbname=replication host=172.25.27.10 port=5432
application_name=postgresql1 fallback_application_name=cluster sslmode=prefer
sslcompression=0 gssencmode=prefer krbsrvname=postgres target_session_attrs=any
(1 row)

```

ЗАМЕЧАНИЕ!!! Вместо переменной `${HOST2_IP}` нужно поставить ip адреса сервера vs02!!!

Установка patroni на vs03

20. Аналогичные действия для patroni проводим для 3 сервера

```

vs03$ mkdir /etc/patroni

vs03$ scp ${HOST1_IP}:/etc/patroni/patroni.yml /etc/patroni/patroni.yml
vs03$ sed -i 's/name: postgresql0/name: postgresql2/' /etc/patroni/patroni.yml
vs03$ sed -i 's/${HOST1_IP}:5432/${HOST3_IP}:5432/' /etc/patroni/patroni.yml
vs03$ sed -i 's/listen: ${HOST1_IP}:8008/listen: ${HOST3_IP}:8008/'
/etc/patroni/patroni.yml
vs03$ sed -i 's/connect_address: ${HOST1_IP}:8008/connect_address: ${HOST3_IP}:8008/'
/etc/patroni/patroni.yml
vs03$ cat /etc/patroni/patroni.yml | grep 'name: postgresql'
vs03$ cat /etc/patroni/patroni.yml | grep '${HOST3_IP}:5432'
vs03$ service patroni start
vs03$ sleep 20
vs03$ service patroni status
name: postgresql2
  listen: 172.25.27.12:5432
  connect_address: 172.25.27.12:5432
Redirecting to /bin/systemctl start patroni.service
Redirecting to /bin/systemctl status patroni.service
• patroni.service - Runners to orchestrate a high-availability PostgreSQL
  Loaded: loaded (/usr/lib/systemd/system/patroni.service; disabled; vendor preset:
disabled)
  Active: active (running) since Thu 2021-09-23 17:59:57 UTC; 20s ago
  Main PID: 2269 (patroni)
  CGroup: /system.slice/patroni.service
          └─2269 /usr/bin/python3 /usr/bin/patroni /etc/patroni/patroni.yml
          └─2300 /usr/pgsql-12/bin/postgres -D /var/lib/pgsql/12/main --config-
file=/var/lib/pgsql/12/main/postgresql.conf --listen_addresses=172.25.27.12 --
port=5432 --cluster_name=cluster --wal_level=replica --hot_standby=on --
max_connections=100 --max_wal_senders=10 --max_prepared_transactions=0 --
max_locks_per_transaction=64 --track_commit_timestamp=off --max_replication_slots=10
--max_worker_processes=8 --wal_log_hints=on
          └─2302 postgres: cluster: logger
          └─2303 postgres: cluster: startup   recovering 000000100000000000000005
          └─2304 postgres: cluster: checkpointer
          └─2305 postgres: cluster: background writer
          └─2306 postgres: cluster: stats collector
          └─2314 postgres: cluster: postgres postgres 172.25.27.12(48890) idle
          └─2316 postgres: cluster: walreceiver   streaming 0/5000060

Sep 23 18:00:01 vs03.s013162.slurm.io patroni[2269]: 2021-09-23 18:00:01.711 UTC
[2300] LOG:  listening on Unix socket "./.s.PGSQL.5432"
Sep 23 18:00:01 vs03.s013162.slurm.io patroni[2269]: 2021-09-23 18:00:01.732 UTC
[2300] LOG:  redirecting log output to logging collector process
Sep 23 18:00:01 vs03.s013162.slurm.io patroni[2269]: 2021-09-23 18:00:01.732 UTC
[2300] HINT:  Future log output will appear in directory "log".

```



```
/service/cluster/leader
/service/cluster/status
/service/cluster/members
/service/cluster/initialize
/service/cluster/config
```

24. Кто сейчас лидер можно увидеть в ключе leader

```
vs01$ etcdctl get /service/cluster/leader
postgresl0
```

25. Параметры лидера можно увидеть тут

```
vs01$ etcdctl get service/cluster/members/postgresql0
{"conn_url":"postgres://172.25.27.10:5432/postgres","api_url":"http://127.0.0.1:8008/patroni","state":"running","role":"master","version":"2.1.1","xlog_location":83886176,"timeline":1}
```

26. Эмулируем выход лидера из строя

```
vs01$ service patroni stop
Redirecting to /bin/systemctl stop patroni.service
```

27. Проверяем статуса кластера patroni

```
vs01$ patronictl -c /etc/patroni/patroni.yml list cluster
+ Cluster: cluster (7011188190946281796) -----+-----+-----+
| Member      | Host          | Role    | State  | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| postgresql0 | 172.25.27.10 | Replica | stopped |   | unknown   |
| postgresql1 | 172.25.27.11 | Leader  | running | 2 |           |
| postgresql2 | 172.25.27.12 | Replica | running | 2 |           |
+-----+-----+-----+-----+-----+-----+

```

28. Возвращаем лидера в стой

```
vs01$ sleep 10
vs01$ service patroni start
Redirecting to /bin/systemctl start patroni.service
```

29. Проверяем статуса кластера patroni

```
vs01$ patronictl -c /etc/patroni/patroni.yml list cluster
+ Cluster: cluster (7011188190946281796) -----+-----+-----+
| Member      | Host          | Role    | State  | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| postgresql0 | 172.25.27.10 | Replica | running | 2 |           |
| postgresql1 | 172.25.27.11 | Leader  | running | 2 |           |
| postgresql2 | 172.25.27.12 | Replica | running | 2 |           |
+-----+-----+-----+-----+-----+-----+

```

Видим, что лидер переехал с postgresql0 на postgresql1

7.6

Домашняя работа

Высокодоступный кластер Patroni

1. Прогоняем скрипт, которые лежит в домашней директории сервера sbox.slurm.io

```
[s000000@sbox.slurm.io ~]$ ~/patroni.sh
... Показана концовка вывода файла ...
+ Cluster: cluster (7011154030226532723) -----+-----+-----+
| Member      | Host           | Role    | State  | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| postgresql0 | 172.25.27.10  | Replica | running | 2 |          0 |
| postgresql1 | 172.25.27.11  | Leader  | running | 2 |          0 |
| postgresql2 | 172.25.27.12  | Replica | running | 2 |          0 |
+-----+-----+-----+-----+-----+-----+

Видим, что лидер переехал с postgresql0 на postgresql1
Конец демонстрации
```

2. Проверяем, что кластер Патрони у нас работает и что лидер находится на сервере postgresql1. Это должен подтверждать статус кластера.
3. Заходим на первый сервер vs01
4. Смотрим помощь к утилите patronictl
5. Находим нужную команду для переезда
6. Запускаем ее, указав путь до конфигурационного файла
7. Подтверждаем, что лидер сейчас это postgresql1
8. Указываем нового кандидата на переезд postgresql0
9. Подтверждаем, что переезд должен быть сейчас
10. Подтверждаем сам факт переезда
11. Проверяем, что сервер переехал

Чек-лист домашнего задания. Лекция "Высокодоступный кластер"

1. Создан высокодоступный кластер Patroni;
2. Вызвана команда изменения мастера в высокодоступном кластере Patroni;

Решение по ссылке в [конце файла](#)

Решение:

1. Прогоняем скрипт, которые лежит в домашней директории сервера sbox.slurm.io

```
[s000000@sbox.slurm.io ~]$ ~/patroni.sh
```

```

... Показана концовка вывода файла ...
+ Cluster: cluster (7011154030226532723) -----+-----+-----+
| Member          | Host            | Role      | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| postgresql0    | 172.25.27.10   | Replica   | running | 2  |          0 |
| postgresql1    | 172.25.27.11   | Leader    | running | 2  |          |
| postgresql2    | 172.25.27.12   | Replica   | running | 2  |          0 |
+-----+-----+-----+-----+-----+-----+

```

Видим, что лидер переехал с postgresql0 на postgresql1
Конец демонстрации

2. Проверяем, что кластер Патрони у нас работает и что лидер находится на сервере postgresql1. Это должен подтвердить статус кластера.

```
| postgresql1 | 172.25.27.11 | Leader | running | 2 |
```

3. Заходим на первый сервер vs01

```

ssh root@vs01.s000000
Warning: Permanently added 'vs01.s000000,172.25.27.10' (ECDSA) to the list of
known hosts.
Last login: Thu Sep 23 15:23:17 2021 from 172.20.100.50
[root@vs01.s000000.slurm.io ~]#

```

4. Смотрим помощь к утилите patronictl

```

root$ patronictl --help
Usage: patronictl [OPTIONS] COMMAND [ARGS]...

Options:
  -c, --config-file TEXT  Configuration file
  -d, --dcs TEXT          Use this DCS
  -k, --insecure          Allow connections to SSL sites without certs
  --help                  Show this message and exit.

Commands:
  configure      Create configuration file
  dsn            Generate a dsn for the provided member,...
  edit-config    Edit cluster configuration
  failover      Failover to a replica
  flush         Discard scheduled events
  history       Show the history of failovers/switchovers
  list          List the Patroni members for a given Patroni
  pause         Disable auto failover
  query         Query a Patroni PostgreSQL member
  reinit        Reinitialize cluster member
  reload        Reload cluster member configuration
  remove        Remove cluster from DCS
  restart       Restart cluster member
  resume        Resume auto failover
  scaffold      Create a structure for the cluster in DCS
  show-config   Show cluster configuration
  switchover    Switchover to a replica
  topology      Prints ASCII topology for given cluster
  version       Output version of patronictl command or a...

```

5. Находим нужную команду для переезда

```
switchover    Switchover to a replica
```

6. Запускаем ее, указав путь до конфигурационного файла

```
patronictl -c /etc/patroni/patroni.yml switchover
```

7. Подтверждаем, что лидер сейчас это postgresql1

```
Master [postgresql1]:
```

8. Указываем нового кандидата на переезд postgresql0

```
Candidate ['postgresql0', 'postgresql2'] []: postgresql0
```

9. Подтверждаем, что переезд должен быть сейчас

```
When should the switchover take place (e.g. 2021-09-23T16:41 ) [now]:
```

10. Подтверждаем сам факт переезда

```
Are you sure you want to switchover cluster cluster, demoting current master postgresql1? [y/N]: y
```

11. Проверяем, что сервер переехал

```
patronictl -c /etc/patroni/patroni.yml list cluster
```

```
+ Cluster: cluster (7011154030226532723) -----+-----+-----+
| Member      | Host          | Role    | State  | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| postgresql0 | 172.25.27.10 | Leader  | running | 3 |           |
| postgresql1 | 172.25.27.11 | Replica | stopped |   | unknown   |
| postgresql2 | 172.25.27.12 | Replica | running | 2 |           0 |
+-----+-----+-----+-----+-----+-----+-----+
```