

# Онлайн образование

• REC

Проверить, идет ли запись

Меня хорошо видно  
&& слышно?



Тема вебинара

# Построение кластера Patroni



**Коробков Виктор**

ООО «ИТ ИКС5 Технологии»

Telegram: @Korobkov\_Viktor



# Правила вебинара



Активно  
участвуем



Off-topic обсуждаем  
в Telegram



Задаем вопрос  
в чат или голосом



Вопросы вижу в чате,  
могу ответить не сразу

## Условные обозначения



Индивидуально



Время, необходимое  
на активность



Пишем в чат



Говорим голосом



Документ



Ответьте себе или  
задайте вопрос

# Маршрут вебинара



# Цели вебинара

После занятия вы сможете

1. Понимать, что такое автоматический failover и как он реализуется
2. Настраивать кластер Patroni
3. Создавать кластер базы данных с высокой доступностью

# High availability – высокая доступность

## Распределенное хранилище

- NFS NAS/SAN - <https://habr.com/ru/post/137938/>
- DRBD - <https://habr.com/ru/post/417473/>
- iSCSI (+ LVM)

## Мульти-мастер

- BDR
- Bucardo

## Логическая репликация

- pglogical
- slony
- в postgresql с 10 версии

## Физическая репликация

- в postgresql начиная с 9.6

## Облака

- Yandex Cloud



# Варианты

## Встроенные решения

- Patroni
  - проксирует все запросы в мастер ноду, нельзя давать нагрузку на реплики;
  - мастер выбирается самостоятельно при switchover-e.
- repmgr:
  - нет защиты от двойного мастера (split brain);
  - нет нужды в DCS.
- Citus pg\_auto\_failover
- Slony

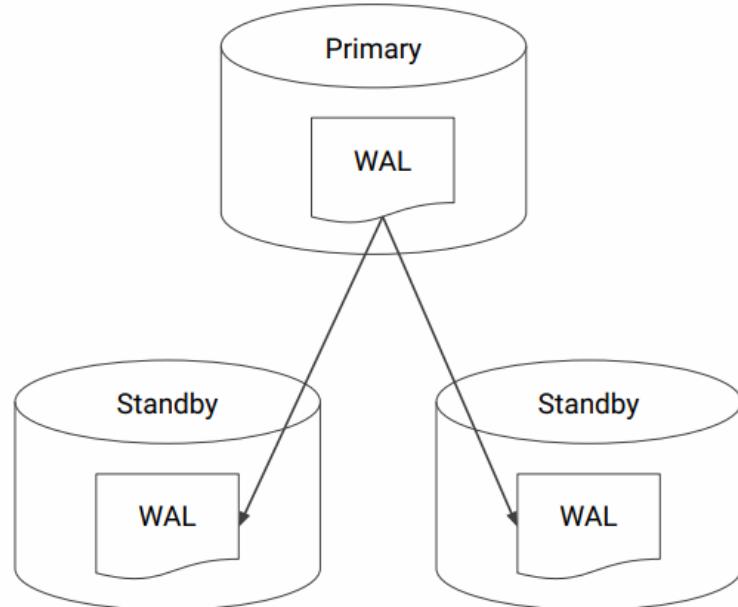
# Физическая репликация

Плюсы:

- встроенная фича;
- минимальная задержка;
- идентичные копии.

Минусы:

- нужны одинаковые мажорные версии;
- нет автоматического failover.

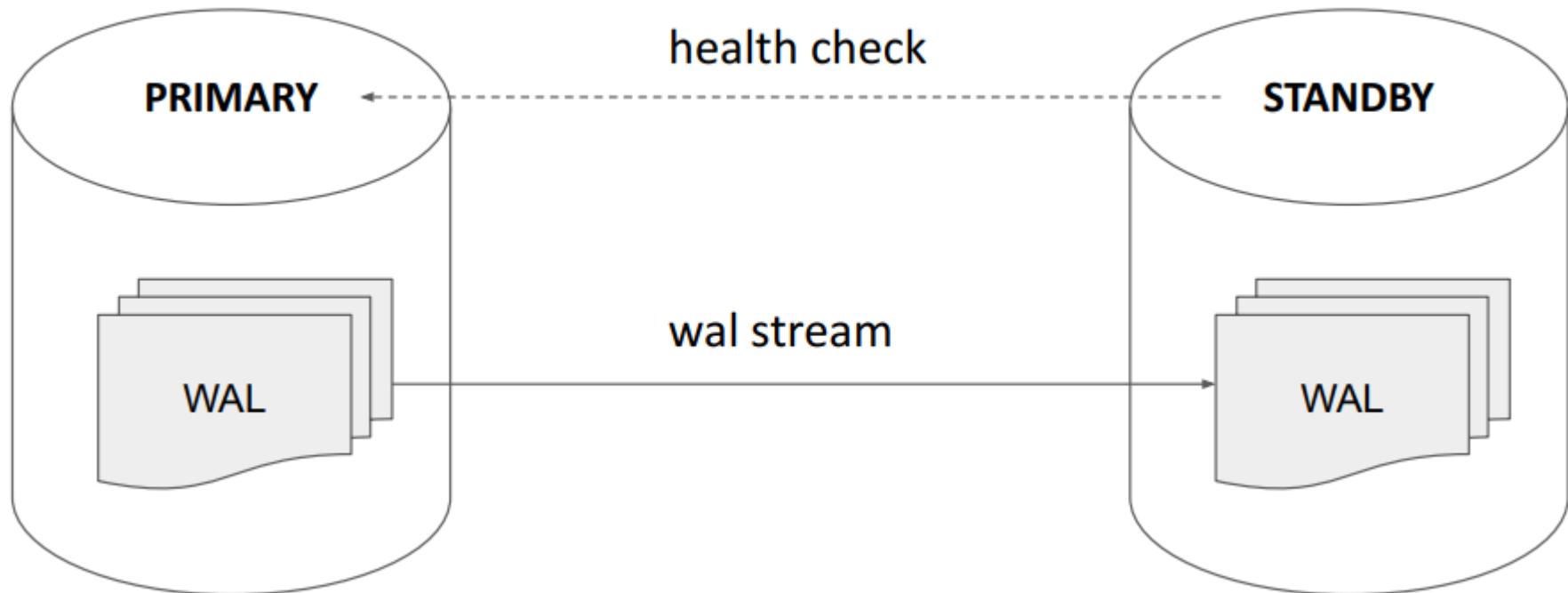


# В чем проблема ?

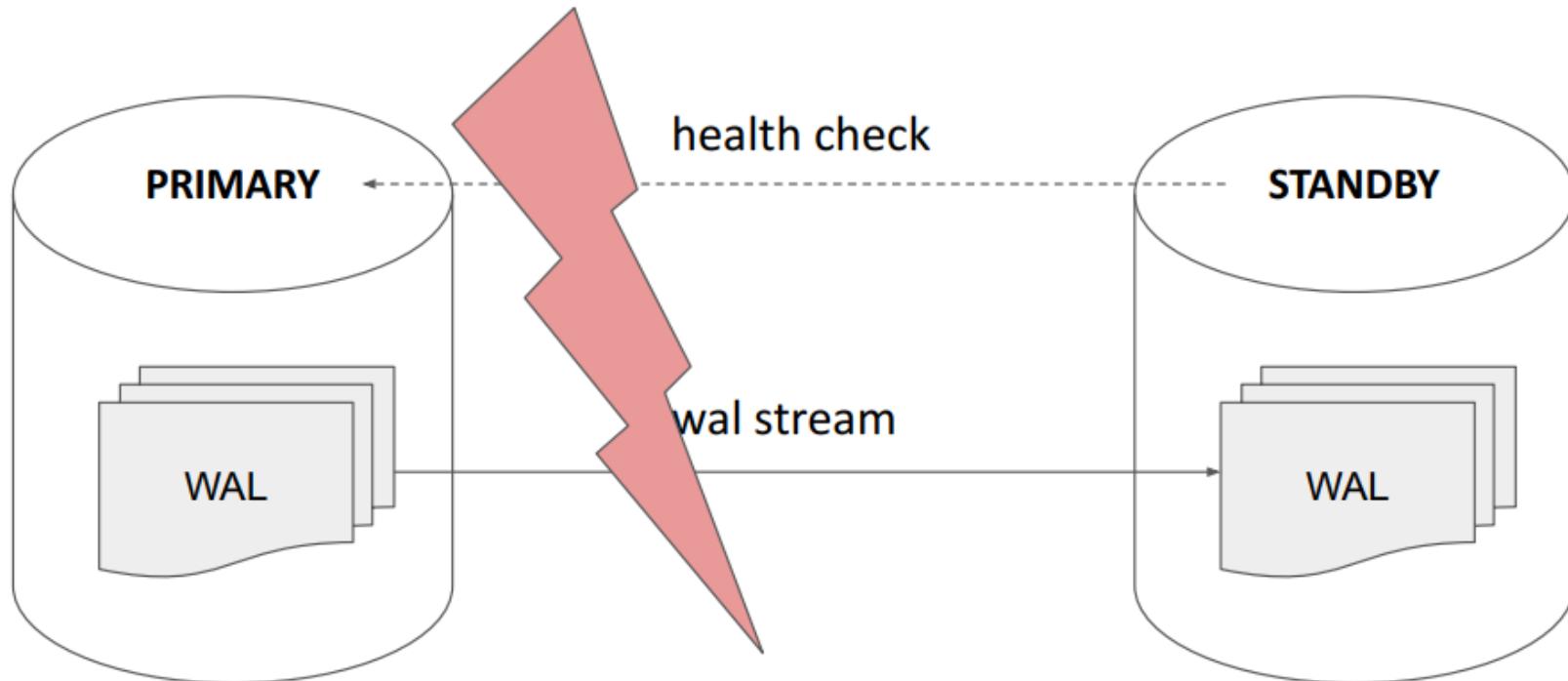


1. Запросы должны автоматически отправляться на активный сервер.
2. В случае наличия hot standby запросы на чтение могут автоматически отправляться на standby сервер.
3. У Постгреса нет какого либо решения по автоматическому фейловеру из коробки

# Автоматический failover



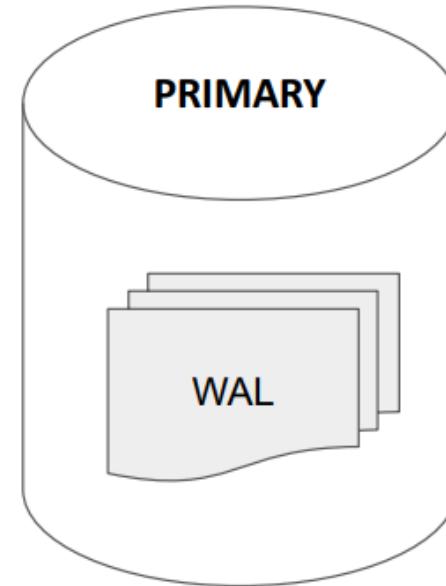
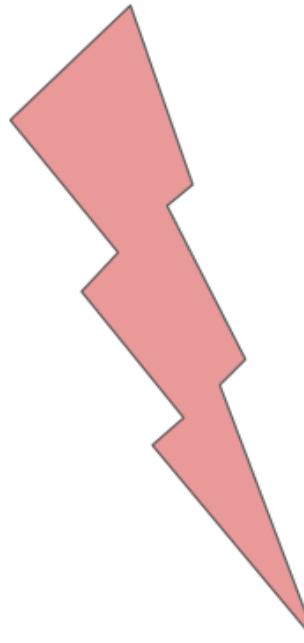
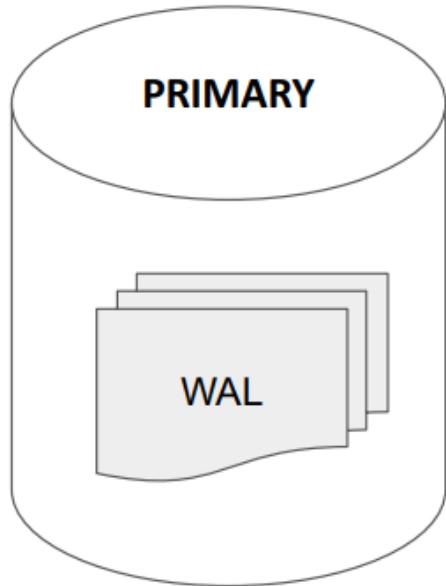
# Автоматический failover



Что произойдет при обрыве сети?

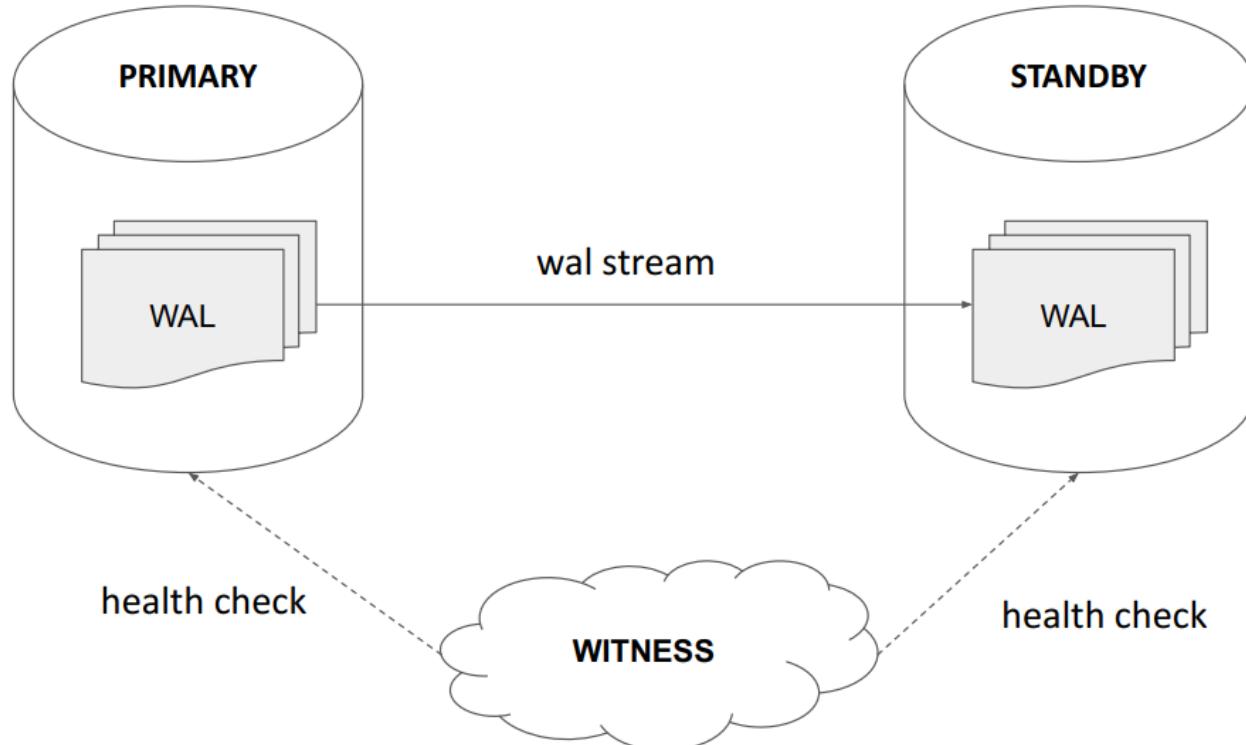


# Автоматический failover



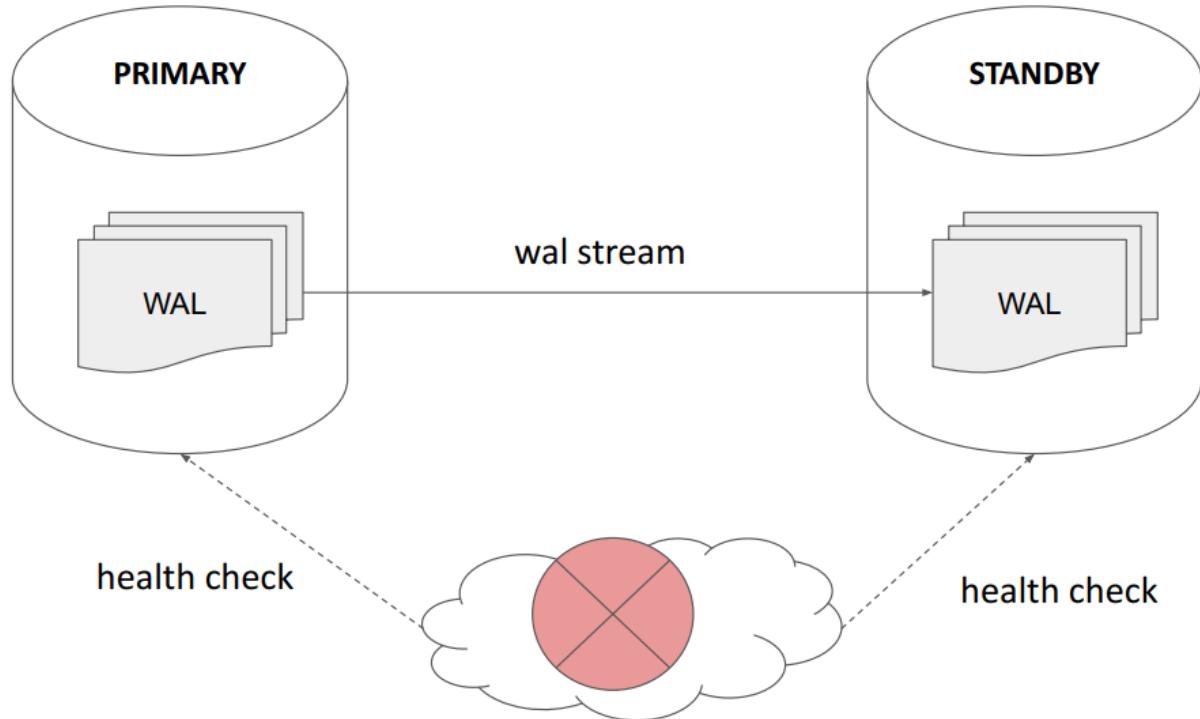
**Split Brain !!!**

# Вариант решения: кластер со свидетелем



**В чем опять проблема ?**

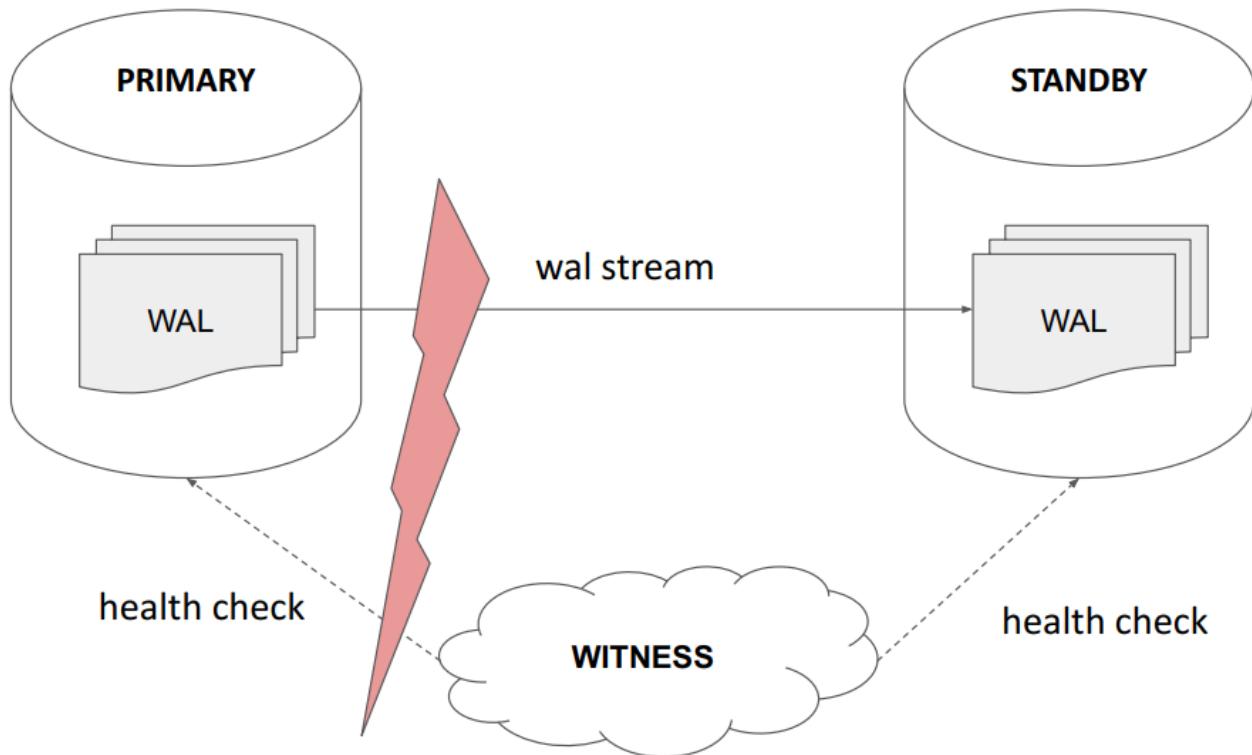
# Вариант решения: кластер со свидетелем



**Свидетеля может не стать**



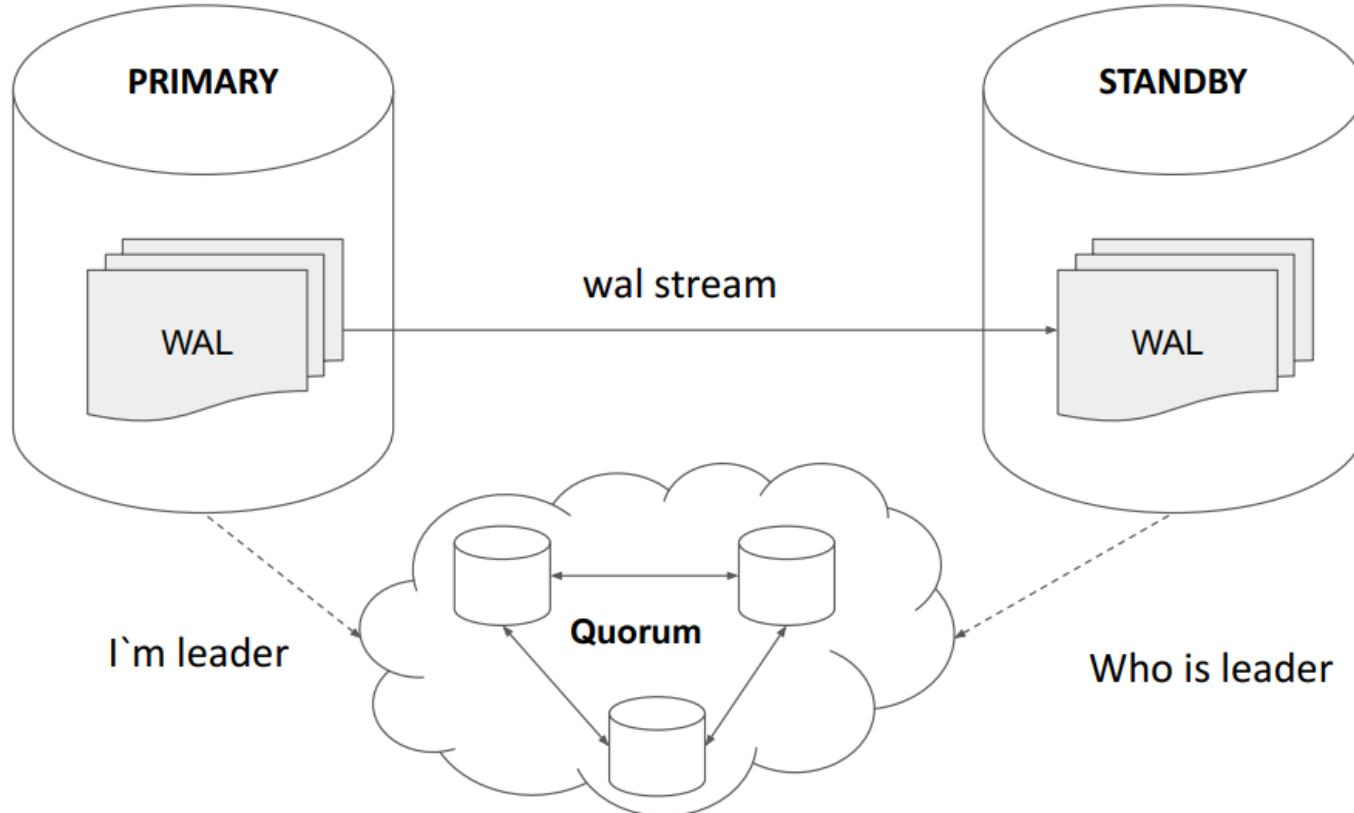
# Вариант решения: кластер со свидетелем



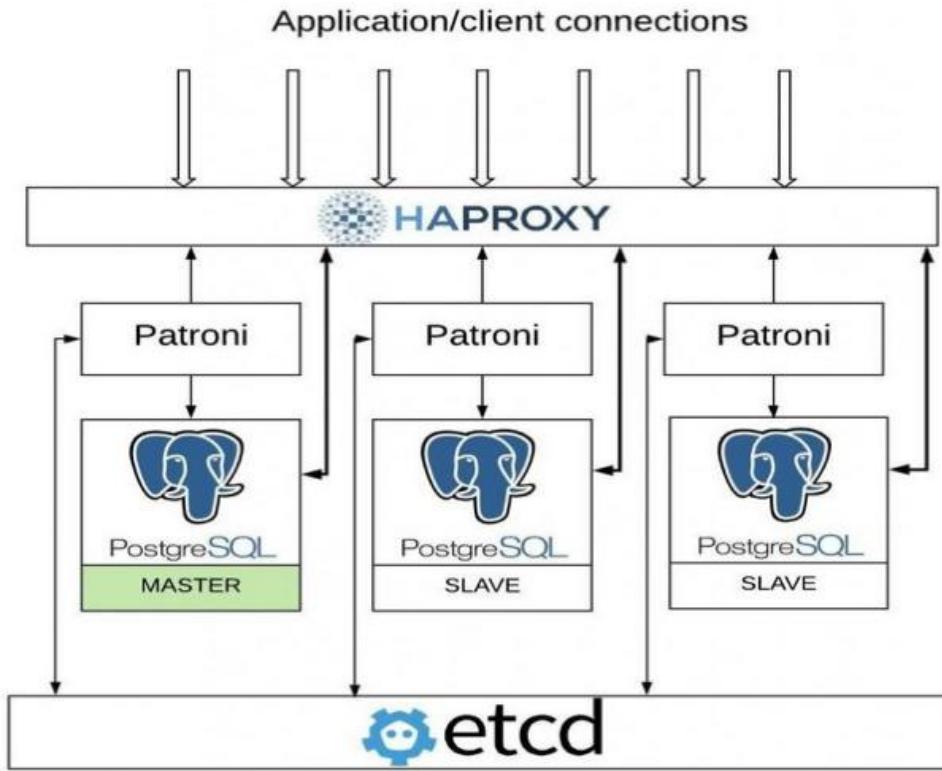
Обрыв соединения и опять Split Brain



# Вариант решения: кластер с кворумом



# Вариант решения: Patroni



Еще можно добавить:  
Pgbounce  
Keepalived  
2-й HAProxy для НА



# Patroni

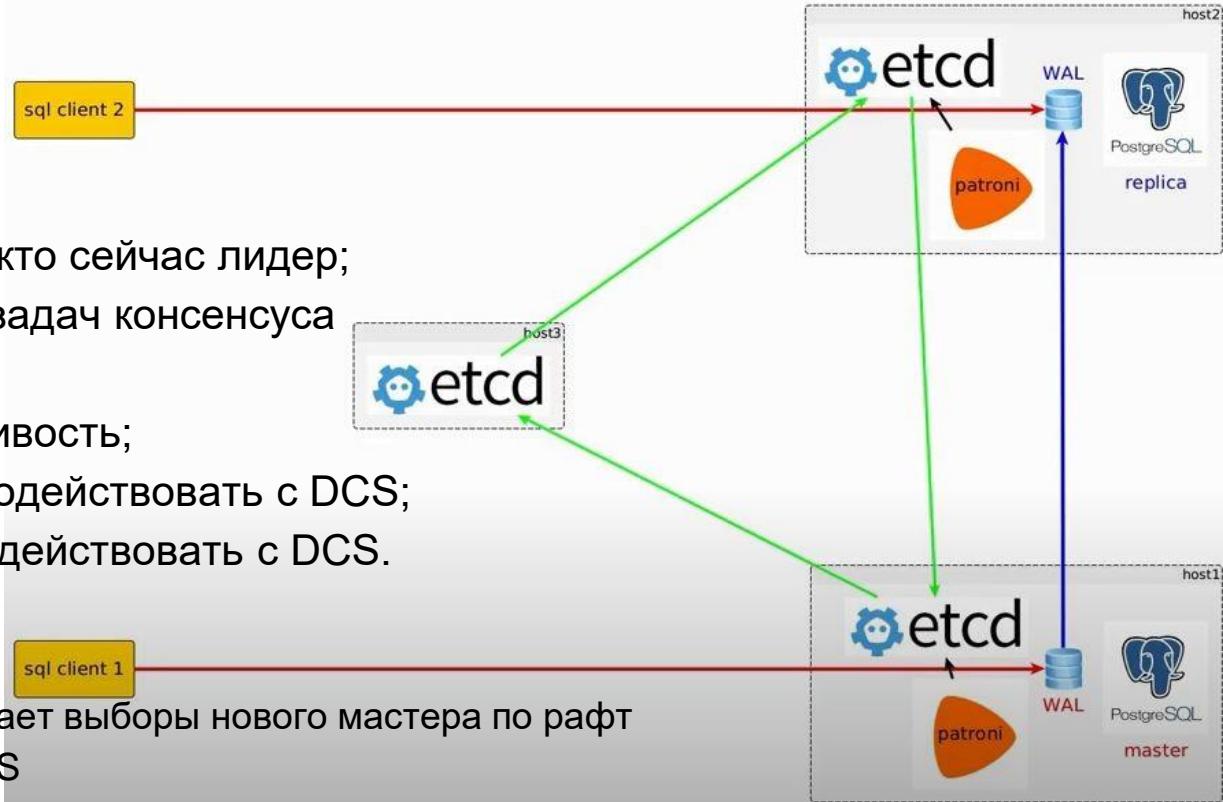
- демон Patroni будет запущен рядом с PostgreSQL;
- Patroni управляет PostgreSQL;
- демон принимает решение promotion/demotion;
- TTL для ключа или сессии лидера.



# DCS (распределенное хранилище данных)

## etcd / Consul / Zookeeper

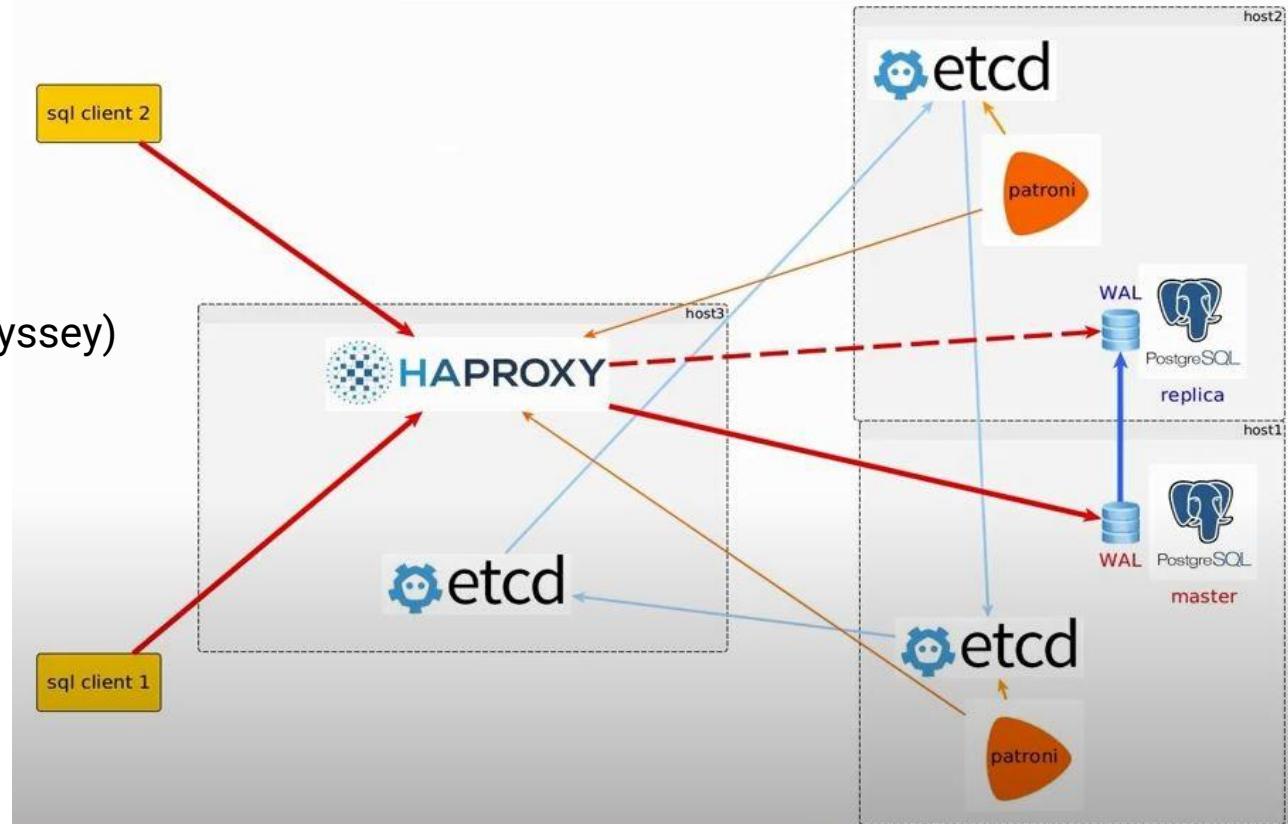
- принцип: key – value;
- хранит информацию о том, кто сейчас лидер;
- имеет алгоритмы решения задач консенсуса ([RAFT](#), [PAXOS](#));
- обеспечивает отказоустойчивость;
- PostgreSQL не умеет взаимодействовать с DCS;
- демон Patroni умеет взаимодействовать с DCS.



С версии 2.0 в Patroni поддерживает выборы нового мастера по рафт протоколу без использования DCS

# Направление клиентов

- HAProxy
- Pgbounce (pgPool, Odyssey)
- KeepaliveD
- TCP Proxy (NGINX)



# ETCD

Это распределенное хранилище данных вида «ключ-значение».

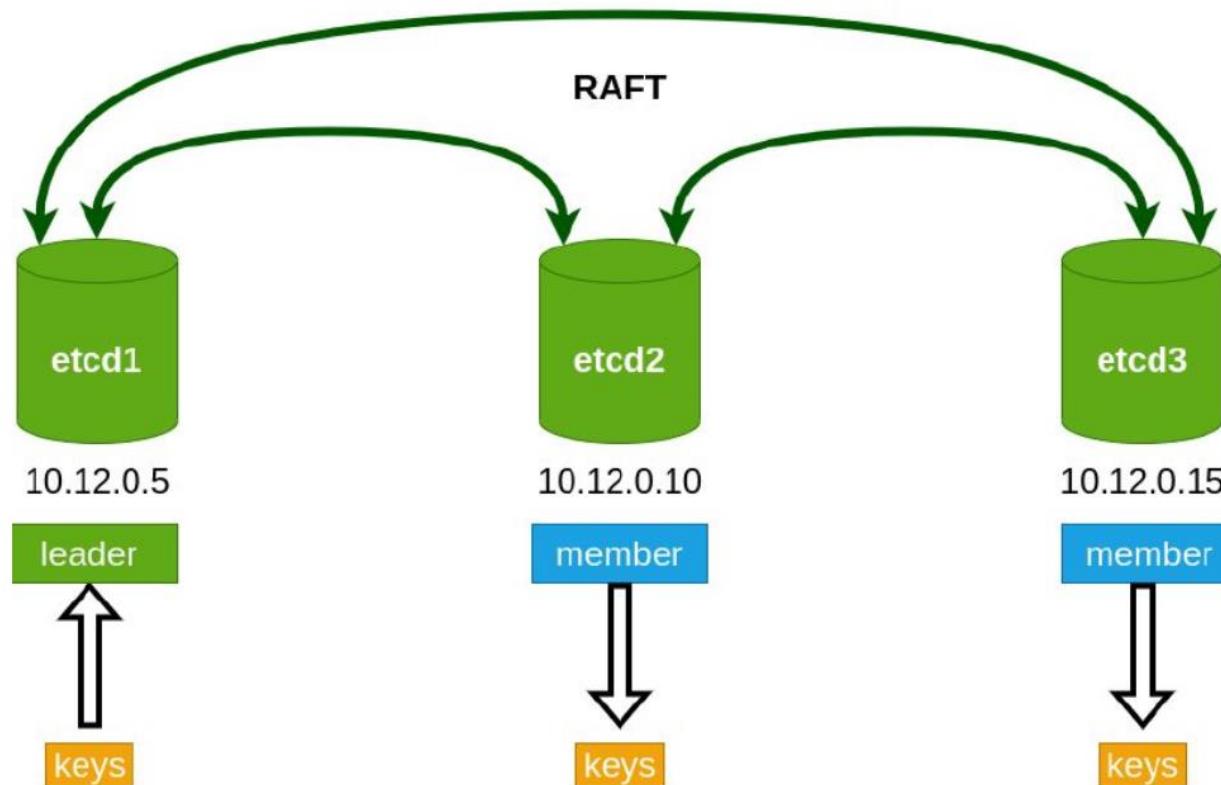
Особенности:

- хранение небольших объемов данных в виде ключей небольшого размера;
- полная репликация между нодами и высокая степень доступности;
- все данные пишутся на диск, *in-memory* отсутствует;
- использует для работы алгоритм консенсуса RAFT;
- написан на Go, кроссплатформенный, имеет небольшой размер;
- минимально отказоустойчивый кластер можно собрать из 3 нод;
- теоретически количество нод не ограничено, но надо помнить, что любое изменение данных согласуют все ноды кластера;
- задержка записи-чтения (так как используется запись на диск) приводит к нестабильной работе кластера и постоянным переизбраниям мастера;
- так как плохо переживает нагрузку на дисковую подсистему - не рекомендуется размещать ноды кластера на используемых уже в продакшен ВМ.

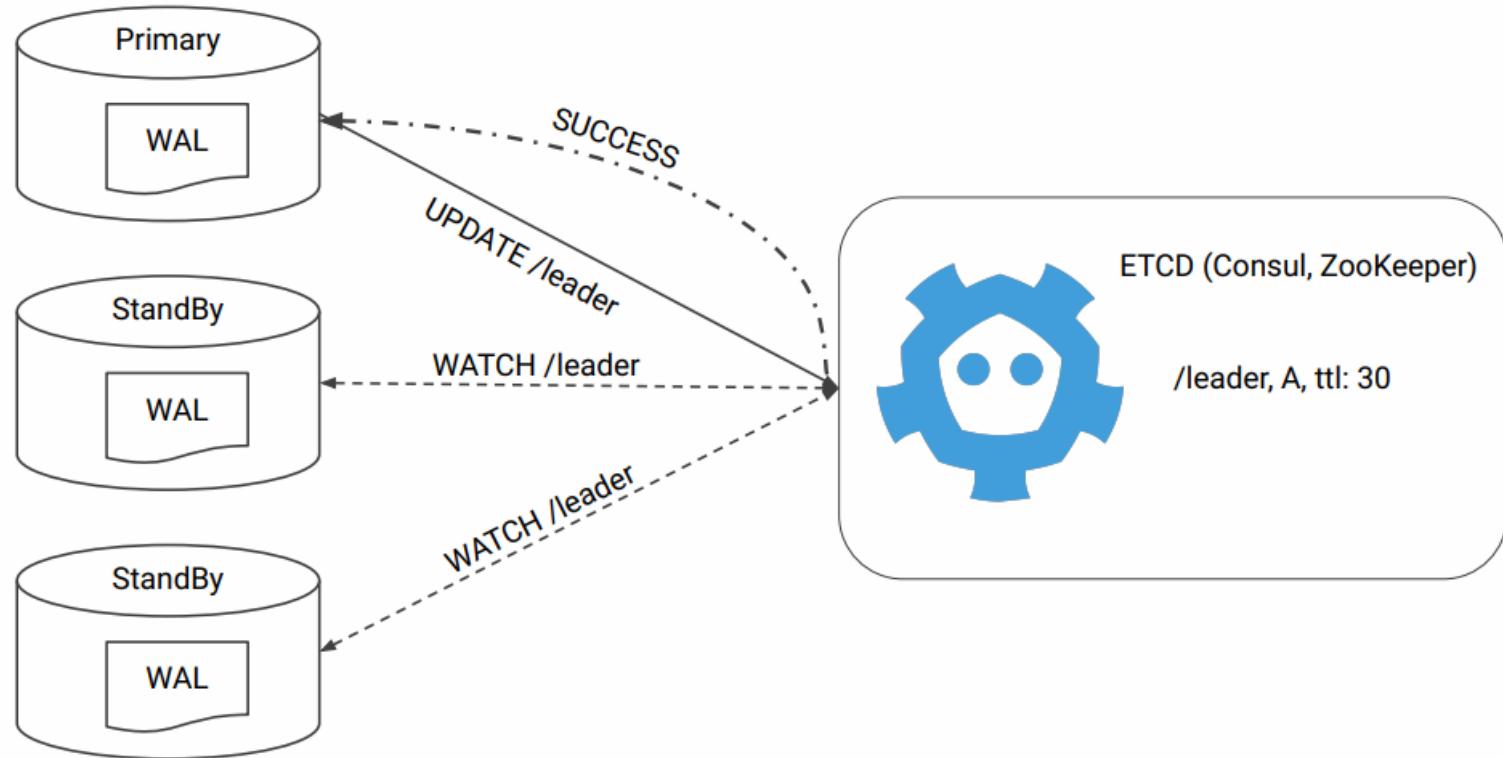


# ETCD

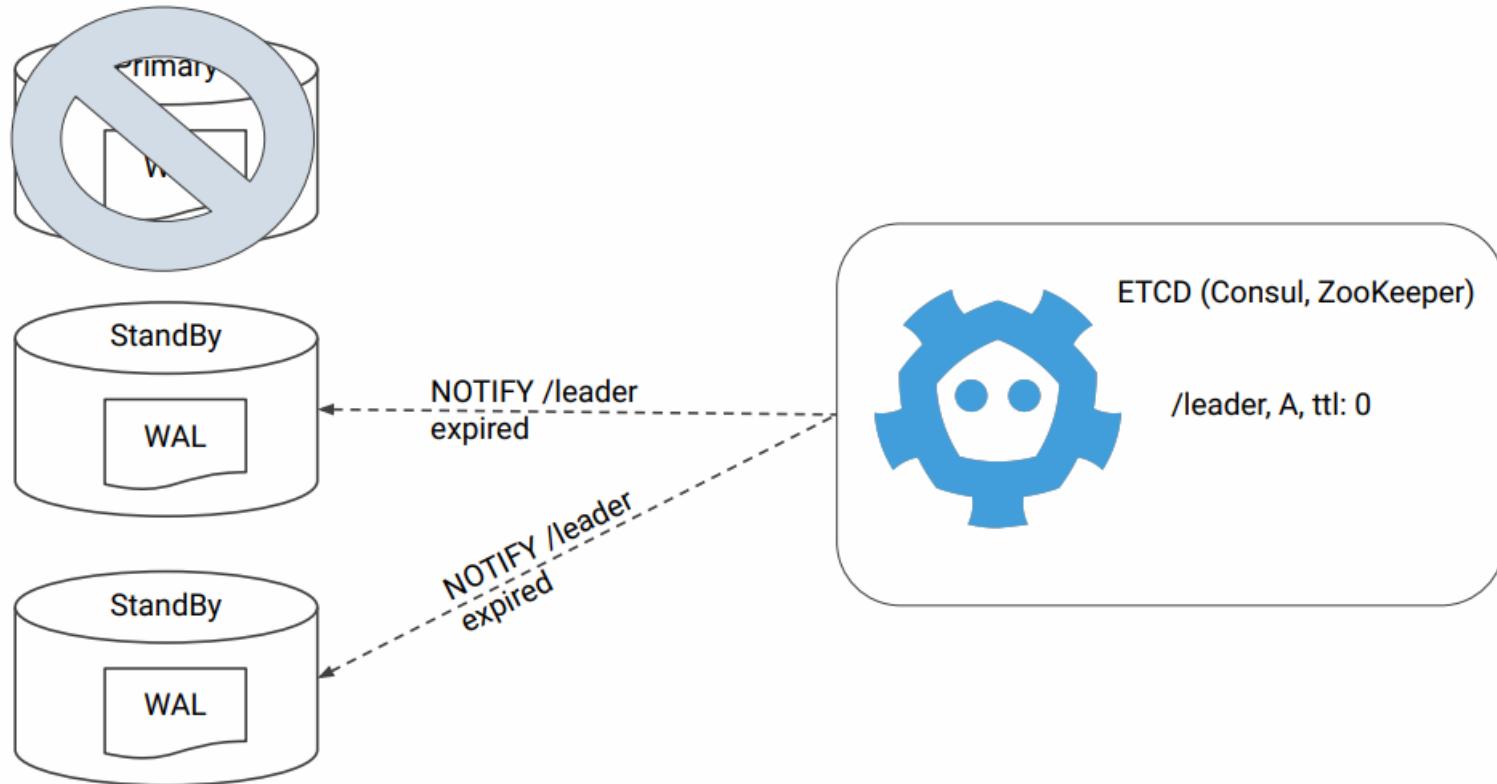
Схема тестового стенда:



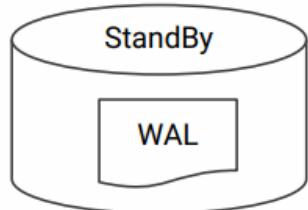
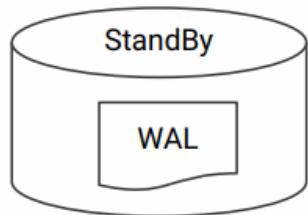
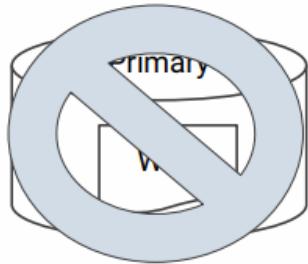
# Автоматическая репликация



# Автоматическая репликация



# Автоматическая репликация



Node B:

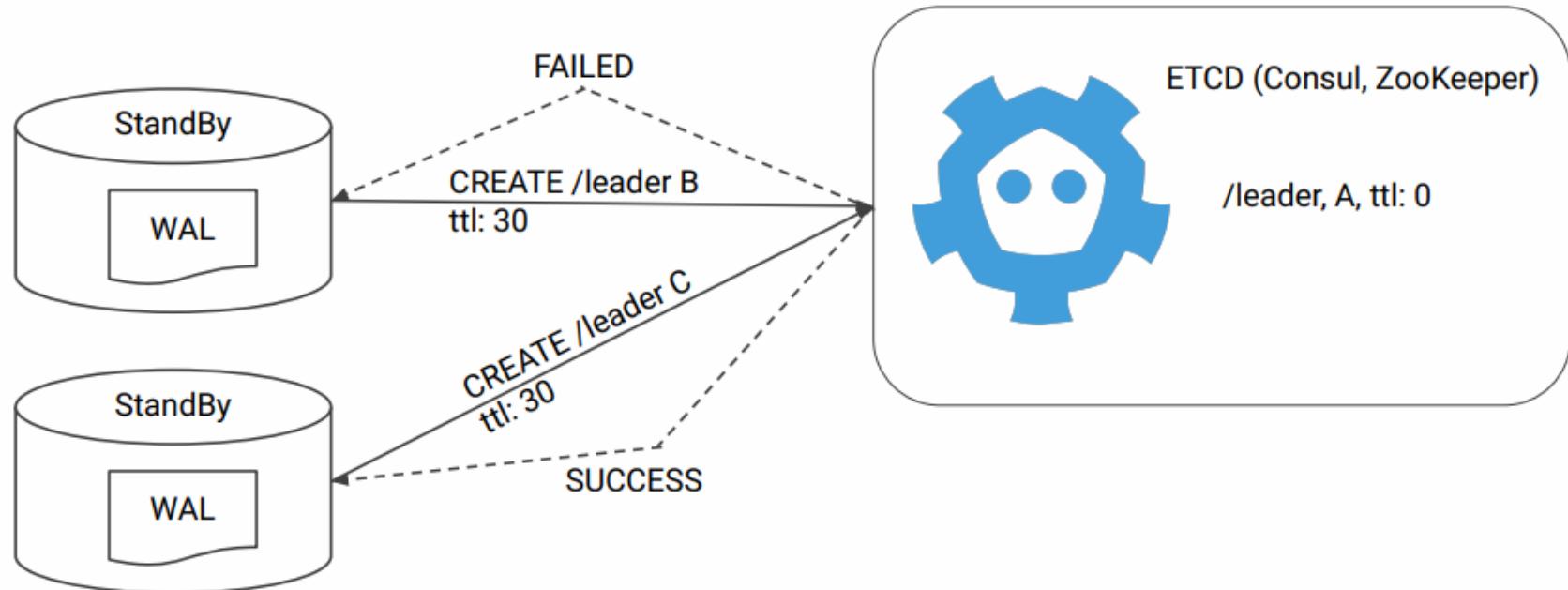
GET hostA:patroni -> Timeout  
GET hostB:patroni -> wal\_position: 200  
GET hostC:patroni -> wal\_position: 100

Node C:

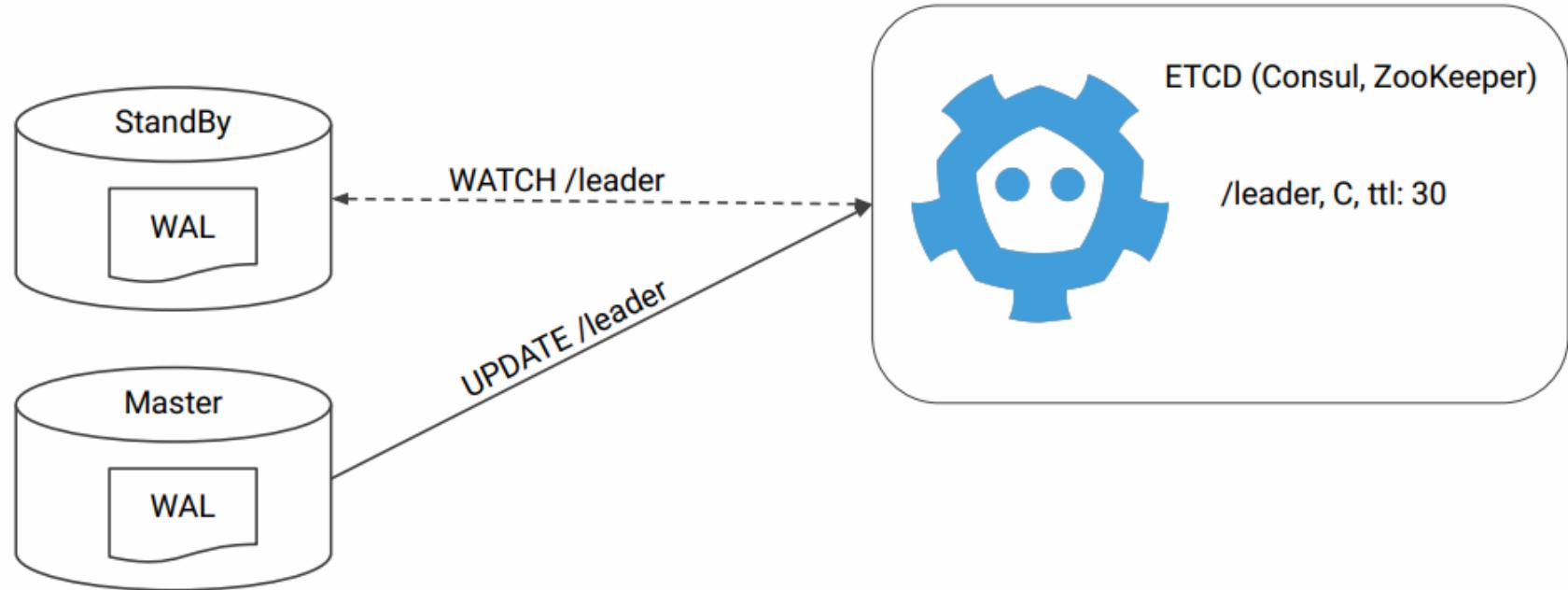
GET hostA:patroni -> Timeout  
GET hostB:patroni -> wal\_position: 200  
GET hostC:patroni -> wal\_position: 100



# Автоматическая репликация



# Автоматическая репликация



# Настройка ETCD

Установка: apt -y install etcd-server && apt -y install etcd-client

vi /etc/default/etcd

```
ETCD_NAME="etcd-Name-1"
ETCD_LISTEN_CLIENT_URLS="http://192.168.1.14:2379,http://localhost:2379"
ETCD_ADVERTISE_CLIENT_URLS="http://hostname1:2379"
ETCD_LISTEN_PEER_URLS="http://192.168.1.14:2380"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://hostname1:2380"
ETCD_INITIAL_CLUSTER_TOKEN="etcd_Name_Claster"
ETCD_INITIAL_CLUSTER="etcd-Name-1=http://hostname1:2380, etcd-Name-2=http://hostname2:2380, etcd-Name-3 =
http://hostname3:2380"
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_DATA_DIR="/var/lib/etcd«
ETCD_ELECTION_TIMEOUT="10000"
ETCD_HEARTBEAT_INTERVAL="2000"
ETCD_INITIAL_ELECTION_TICK_ADVANCE="false"
ETCD_ENABLE_V2="true"
```

# Настройка ETCD

**ETCD\_DATA\_DIR** указывает расположение каталога данных кластера.

**ETCD\_LISTEN\_PEER\_URLS** задаёт схему и точку подключения для остальных узлов кластера, по шаблону scheme://IP:port. если в качестве IP адреса указано 0.0.0.0, то указанный порт будет прослушиваться на всех интерфейсах.

**ETCD\_LISTEN\_CLIENT\_URLS** задаёт схему и точку подключения для клиентов кластера. В остальном совпадает с **ETCD\_LISTEN\_PEER\_URLS**.

**ETCD\_NAME** имя узла кластера. Должно быть уникально в кластере. Для первого узла может быть любым. Для последующих должно совпадать с именем, указанным при добавлении узла.

**ETCD\_HEARTBEAT\_INTERVAL** время в миллисекудах, между рассылками лидером оповещений о том, что он всё ещё лидер. Рекомендуется задавать с учётом сетевой задержки между узлами кластера.

**ETCD\_ELECTION\_TIMEOUT** время в миллисекундах, которое проходит между последним принятым оповещением от лидера кластера, до попытки захватить роль лидера на ведомом узле. Рекомендуется задавать его в несколько раз большим, чем **ETCD\_HEARTBEAT\_INTERVAL**.



# Настройка ETCD

**ETCD\_INITIAL\_ADVERTISE\_PEER\_URLS** URL-адрес, по которым его могут найти остальные узлы кластера. Используется только при первом запуске нового узла кластера.

**ETCD\_ADVERTISE\_CLIENT\_URLS** URL-адрес, по которым его могут найти остальные узлы кластера. Используется для передачи данных по кластеру. Может содержать доменные имена.

**ETCD\_INITIAL\_CLUSTER** Список узлов кластера на момент запуска. Используется только при первом запуске нового узла кластера.

**ETCD\_INITIAL\_CLUSTER\_TOKEN** Токен кластера. Должен совпадать на всех узлах кластера. Используется только при первом запуске нового узла кластера.

**ETCD\_INITIAL\_CLUSTER\_STATE** может принимать два значения «new» и «existing». Значение «new» используется при первом запуске первого узла в кластере. При значении «existing», узел при старте будет пытаться установить связь с остальными узлами кластера.

# Настройка ETCD

**После создания кластера есть рекомендация в конфиге поправить new->existing**

**Добавить участника в кластер:**

```
etcdctl member add etcd4 --peer-urls=http://10.12.0.20:2380
```

```
ETCD_NAME="etcd4"
```

```
ETCD_INITIAL_CLUSTER="etcd3=http://10.12.0.15:2380,etcd2=http://10.12.0.10:2380 "
```

```
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://hostname4:2380"
```

```
ETCD_INITIAL_CLUSTER_STATE="existing"
```

**Удалить участника:**

```
etcdctl member remove <member_id>
```



# Настройка ETCD

Команды:

systemctl status etcd

systemctl start etcd

systemctl stop etcd

systemctl is-enabled etcd

systemctl restart etcd

etcd --version

ETCDCTL\_API=2

echo \$ETCDCTL\_API

journalctl -e посмотреть информацию об ошибке

etcdctl endpoint health --cluster

rm -R /var/lib/etcd/member/

etcdctl endpoint status --cluster -w table

ENDPOINT	ID	VERSION	DB SIZE	IS LEADER
http://10.0.0.19:2379	53531d8291f44edb	3.4.30	20 kB	true
http://10.0.0.24:2379	aab377b8295d285c	3.4.30	20 kB	false
http://10.0.0.10:2379	f05c4b4306b0ecf5	3.4.30	20 kB	false



# Patroni

Разработчики: Александр Кукушкин, Алексей Клюкин (Zalando SE)

Документация: <https://patroni.readthedocs.io/en/latest/index.html>

Репозиторий: <https://github.com/zalando/patroni>

Выступление на конференции: <https://www.youtube.com/watch?v=IMPYerAYEVs&t=8109s>

# Кластер Patroni

Name	IP-address	Purpose
Node1	192.168.1.11	PostgreSQL, Patroni
Node2	192.168.1.12	PostgreSQL, Patroni
Etcd	192.168.1.14	etcd



# Настройка PostgreSQL

1. Установка PostgreSQL (на каждой ноде):

```
apt -y install postgresql
```

2. Создать пользователя replicator:

```
create user replicator replication login encrypted password 'password';
```

3. Отредактировать файл pg\_hba.conf:

```
host all all 0.0.0.0/0 scram-sha-256
```

```
host replication replicator 0.0.0.0/0 scram-sha-256
```

4. Отредактировать файл postgresql.conf:

```
listen_address = '*'
```

5. Перезапустить PostgreSql:

```
sudo systemctl restart postgresql
```

6. На второй и третьей ноде удалить содержимое каталога pgdata:

```
sudo systemctl stop postgresql
```

```
rm -rf /var/lib/postgresql/17/main/*
```



# Настройка кластера Patroni

Установка на каждой ноде (вариант 1):

- sudo apt -y install python3 python3-pip python3-dev python3-psycopg2 libpq-dev
- sudo pip3 install launchpadlib --break-system-packages
- sudo pip3 install --upgrade setuptools --break-system-packages
- sudo pip3 install psycopg2 --break-system-packages
- sudo pip3 install python-etcd --break-system-packages
- sudo apt -y install patroni
  
- sudo systemctl stop patroni
- sudo systemctl disable patroni

# Настройка кластера Patroni

Установка на каждой ноде (вариант 2):

Новый механизмом безопасности в Ubuntu 22.04+ (и Debian 12+) блокирует глобальную установку Python-пакетов через pip, чтобы не повредить системные пакеты.

1. Ставим модуль для создания виртуальных окружений

```
sudo apt install python3.12-venv
```

2. Создаём каталог для Patroni.

```
sudo mkdir -p /opt/patroni
```

3. Передаём владение каталогом пользователю postgres

```
sudo chown postgres:postgres /opt/patroni
```

4. Создаём виртуальное окружение от имени postgres.

```
sudo -u postgres python3 -m venv /opt/patroni/venv
```

5. Устанавливаем Patroni с поддержкой etcd3.

```
sudo -u postgres /opt/patroni/venv/bin/pip install 'patroni[etcd3]'
```

6. Устанавливаем драйвер для работы Patroni с PostgreSQL.

```
sudo -u postgres /opt/patroni/venv/bin/pip install 'psycopg2-binary'
```



# Patroni.yml

- vi /etc/patroni.yml

scope: **Name\_Cluster**

namespace: /db/

name: **Node1**

restapi:

  listen: **192.168.1.11 :8008**

  connect\_address: **192.168.1.11 :8008**

etcd:

  hosts: **hostname1:2379, hostname2:2379,**

**hostname3:2379**

bootstrap:

  dcs:

    ttl: 30

    loop\_wait: 10

    retry\_timeout: 10

    maximum\_lag\_on\_failover: 1048576

...

DCS:

- **loop\_wait** - минимальный промежуток в секундах между попытками обновить ключ лидера.

- **ttl** - время жизни ключа лидера, рекомендуется как минимум **loop\_wait + retry\_timeout \* 2**

- **retry-timeout** - общее время всех попыток внутри одной операции

- **maximum\_lag\_on\_failover** - максимальное отставание ноды от лидера для того, чтобы участвовать в выборах



# Patroni.yml

```
...
postgresql:
    use_pg_rewind: true
    parameters:
        autovacuum_analyze_scale_factor: 0.01
        ...
initdb:
    - encoding: UTF8
    pg_hba:
        - host replication replicator 127.0.0.1/8 md5
        - host replication replicator 192.168.1.11 md5
        - host replication replicator 192.168.1.12 md5
        - host all all 0.0.0.0/0 md5
users:
    admin:
        password: Пароль админа
        options:
            - createrole
            - createdb
```

```
...
postgresql:
    listen: 127.0.0.1, 192.168.1.11 :5432
    connect_address: 192.168.1.11 :5432
    data_dir: /var/lib/postgresql/14/main
    bin_dir: /usr/lib/postgresql/14/bin
    authentication:
        replication:
            username: replicator
            password: Пароль
        superuser:
            username: postgres
            password: Пароль
    rewind:
        username: rewind_user
        password: Пароль
parameters:
    unix_socket_directories: ''
```



# Patroni.yml

```
...  
tags:  
 nofailover: false  
noloadbalance: false  
clonefrom: false  
nosync: false
```

Tags:

- **nofailover** (true/false) - в положении true нода никогда не станет мастером
- **noloadbalance** (true/false) - replica всегда возвращает код 503
- **clonefrom** (true/false) - patronictl выберет предпочтительную ноду для pgbasebackup
- **nosync** (true/false) - нода никогда не станет синхронной репликой
- **replicatefrom** (node name) - указать реплику с которой снимать реплику



# Настройка кластера Patroni

Определяем Patroni как службу (на каждой ноде):

- vi /etc/systemd/system/patroni.service

```
[Unit]
Description=High availability PostgreSQL Cluster
After=syslog.target network.target
[Service]
Type=simple:
User=postgres
Group=postgres
ExecStart=/usr/local/bin/patroni /etc/patroni.yml
KillMode=process
TimeoutSec=30
Restart=no
[Install]
WantedBy=multi-user.target
```



# Настройка кластера Patroni

Перевод Patroni в автозапуск, старт и проверка:

```
systemctl daemon-reload
```

```
systemctl enable patroni
```

```
systemctl start patroni
```

```
systemctl status patroni
```

Помощь:

```
patronictl --help
```

Просмотр состояния кластера:

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

# Автоматический failover

# systemctl stop patroni – или любой другой способ протестировать failover =)

1. 30 секунд по умолчанию на истечение ключа в DCS.
2. После чего Patroni стучится на каждую ноду в кластере и спрашивает, не мастер ли ты, проверяет WAL логи, насколько близки они к мастеру. В итоге если WAL логи у всех одинаковые то, промоутится следующий по порядку.
3. Опрос нод идёт параллельно.



# Команды Patroni

systemctl stop patroni

patronictl -c /etc/patroni.yml list

```
vic@node1:~$ patronictl -c /etc/patroni.yml list
+ Cluster: postgres (6995624626377153828) ---+-----+
| Member | Host           | Role    | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| node1  | 192.168.1.11 | Leader  | running | 16 |          |
| node2  | 192.168.1.12 | Replica | running | 16 |      0 |
+-----+-----+-----+-----+-----+-----+
vic@node1:~$ sudo systemctl stop patroni
[sudo] password for vic:
vic@node1:~$ patronictl -c /etc/patroni.yml list
+ Cluster: postgres (6995624626377153828) ---+-----+
| Member | Host           | Role    | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| node1  | 192.168.1.11 | Replica | stopped |     | unknown |
| node2  | 192.168.1.12 | Leader  | running | 17 |          |
+-----+-----+-----+-----+-----+-----+
```

systemctl start patroni

patronictl -c /etc/patroni.yml list

```
vic@node1:~$ sudo systemctl start patroni
vic@node1:~$ patronictl -c /etc/patroni.yml list
+ Cluster: postgres (6995624626377153828) ---+-----+
| Member | Host           | Role    | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| node1  | 192.168.1.11 | Replica | running | 17 |      0 |
| node2  | 192.168.1.12 | Leader  | running | 17 |          |
+-----+-----+-----+-----+-----+-----+
```

# Команды Patroni

```
curl -v http://192.168.1.11:8008/patroni | master | replica
```

```
vic@node1:~$ curl -v http://192.168.1.11:8008/patroni
*   Trying 192.168.1.11:8008...
* TCP_NODELAY set
* Connected to 192.168.1.11 (192.168.1.11) port 8008 (#0)
> GET /patroni HTTP/1.1
> Host: 192.168.1.11:8008
> User-Agent: curl/7.68.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
* HTTP 1.0, assume close after body
< HTTP/1.0 200 OK
< Server: BaseHTTP/0.6 Python/3.8.10
< Date: Tue, 24 Aug 2021 18:11:22 GMT
< Content-Type: application/json
<
* Closing connection 0
{"state": "running", "postmaster_start_time": "2021-08-24 21:09:48.944547+03:00", "role": "replica", "server_version": 120008, "cluster_unlocked": false, "xlog": {"received_location": 100664256, "replayed_location": 100664256, "timestamp": null, "paused": false}, "timeline": 17, "database_system": "6995624626377153828", "patroni": {"version": "2.1.0", "scope": "all", "last_lsn": 100664256}}
```

```
vic@node2:~$ curl -v http://192.168.1.12:8008/replica
*   Trying 192.168.1.12 (192.168.1.12) port 8008 (#0)
* TCP_NODELAY set
* Connected to 192.168.1.12 (192.168.1.12) port 8008 (#0)
> GET /replica HTTP/1.1
> Host: 192.168.1.12:8008
> User-Agent: curl/7.68.0
> Accept: */*
>
* Mark bundle as not supporting multiuse
* HTTP 1.0, assume close after body
< HTTP/1.0 503 Service Unavailable
< Server: BaseHTTP/0.6 Python/3.8.10
< Date: Tue, 24 Aug 2021 18:13:52 GMT
< Content-Type: application/json
<
```

В PostgreSQL:

```
select pg_is_in_recovery();
```

true – replica

false – master

# **Switchover vs failover**

## **Failover**

- Экстренное переключение Мастера на новую ноду.
- Происходит автоматически.
- Ручной вариант - manual failover - только когда система не может решить на кого переключать.

## **Switchover**

- Переключение роли Мастера на новую ноду. Делается вручную, по сути плановые работы.

# Switchover

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

```
vic@node1:~$ patronictl -c /etc/patroni.yml switchover
Master [node2]:
Candidate ['node1'] []:
When should the switchover take place (e.g. 2021-08-24T22:23) [now]:
Current cluster topology
+ Cluster: postgres (6995624626377153828) +-----+
| Member | Host      | Role    | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+
| node1  | 192.168.1.11 | Replica | running | 17 |          0 |
| node2  | 192.168.1.12 | Leader   | running | 17 |          |
+-----+-----+-----+-----+-----+
Are you sure you want to switchover cluster postgres, demoting current master node2? [y/N]: y
2021-08-24 21:23:58.98114 Successfully switched over to "node1"
+ Cluster: postgres (6995624626377153828) +-----+
| Member | Host      | Role    | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+
| node1  | 192.168.1.11 | Leader   | running | 17 |          |
| node2  | 192.168.1.12 | Replica | stopped |     | unknown   |
+-----+-----+-----+-----+-----+
```

# Глобальная конфигурация

```
patronictl -c /etc/patroni.yml edit-config
```

```
vtc@node1:~$ patronictl -c /etc/patroni.yml edit-config
--+
+++ @@ -2,7 +2,7 @@
maximum_lag_on_failover: 1048576
postgresql:
  parameters:
-   max_connections: 100
+   max_connections: 101
  use_pg_rewind: true
  retry_timeout: 10
  synchronous_mode: false

Apply these changes? [y/N]: y
Configuration changed
vtc@node1:~$ patronictl -c /etc/patroni.yml list
+ Cluster: postgres (6995624626377153828) +-----+
+ | Member | Host           | Role      | State    | TL | Lag in MB | Pending restart |
+-----+-----+-----+-----+-----+-----+
+ | node1  | 192.168.1.11 | Leader    | running  | 18 |             | *                |
+ | node2  | 192.168.1.12 | Replica   | running  | 18 | 0          | *                |
+-----+-----+-----+-----+-----+-----+
```

# Глобальная конфигурация

patronictl -c /etc/patroni.yml restart ИМЯ\_клестера

```
vic@node1:~$ patronictl -c /etc/patroni.yml restart postgres
+ Cluster: postgres (6995624626377153828)  +-----+
+ Member | Host      | Role     | State    | TL | Lag in MB | Pending restart
+-----+-----+-----+-----+-----+-----+
+ node1 | 192.168.1.11 | Leader   | running  | 18 |           | *
| node2 | 192.168.1.12 | Replica  | running  | 18 |          0 | *
+-----+-----+-----+-----+-----+-----+
+
When should the restart take place (e.g. 2021-08-24T22:29) [now]:
Are you sure you want to restart members node2, node1? [y/N]: y
Restart if the PostgreSQL version is less than provided (e.g. 9.5.2) []:
Success: restart on member node2
Success: restart on member node1
vic@node1:~$ patronictl -c /etc/patroni.yml list
+ Cluster: postgres (6995624626377153828)  +-----+
| Member | Host      | Role     | State    | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| node1 | 192.168.1.11 | Leader   | running  | 18 |           |
| node2 | 192.168.1.12 | Replica  | running  | 18 |          0 |
+-----+-----+-----+-----+-----+
```

# Локальная конфигурация

Что делать если нужно поменять конфигурацию PostgreSQL только локально:

- patroni.yml
- postgresql.base.conf
- ALTER SYSTEM SET - имеет наивысший приоритет

Параметры : max\_connections, max\_locks\_per\_transaction, wal\_level, max\_wal\_senders,

max\_prepared\_transactions, max\_replication\_slots, max\_worker\_processes

не могут быть переопределены локально - Patroni их перезаписывает



# Пользовательские скрипты

postgresql:

callbacks:

on\_start: /opt/pgsql/pg\_start.sh

on\_stop: /opt/pgsql/pg\_stop.sh

on\_restart: /opt/pgsql/pg\_restart.sh

on\_role\_change: /opt/pgsql/pg\_role\_change.sh



# Реинициализация

`patronictl -c /etc/patroni.yml reinit имя_кластера имя_ноды` - реинициализирует ноду в кластере.

Т.е. по сути удаляет директорию и делает pg\_basebackup

# Режим паузы

patronictl -c /etc/patroni.yml pause|resume - отключается | включается  
автоматический failover

Ставится глобальная пауза на все ноды

Проведение плановых работ, например с etcd или обновление PostgreSQL

Тем ни менее:

- можно создавать реплики;
- ручной switchover возможен.

# Синхронная репликация

**synchronous\_mode: true/false** - не делает failover ни на какую реплику кроме синхронной

**synchronous\_mode\_strict: true/false** - если синхронная реплика пропала, то мастер не принимает новые записи пока она не вернется

**synchronous\_commit to local / off** – установка асинхронного режима для транзакции даже при общем синхронном режиме

# Синхронная репликация

```
vic@node1:~$ patronictl -c /etc/patroni.yml edit-config
---
+++
@@ -5,5 +5,5 @@
     max_connections: 101
     use_pg_rewind: true
     retry_timeout: 10
-    synchronous_mode: false
+    synchronous_mode: true
     ttl: 30
```

```
Apply these changes? [y/N]: y
Configuration changed
```

```
vic@node1:~$ patronictl -c /etc/patroni.yml list
+ Cluster: postgres (6995624626377153828) -----
| Member | Host           | Role        | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| node1  | 192.168.1.11 | Leader      | running | 18 |          |
| node2  | 192.168.1.12 | Sync Standby | running | 18 |          0 |
+-----+-----+-----+-----+-----+-----+
```

# Дополнительный материал

Patroni + Consul: [https://gitlab.com/otus\\_linux/patroni](https://gitlab.com/otus_linux/patroni)  
<https://github.com/lalbrekht/otus-patroni>

Patroni + Zookeeper: <https://temofeev.ru/info/articles/zaryazhay-patroni-testiruem-patroni-zookeeper-klaster-chast-pervaya/>

Etcd: <https://github.com/coreos/etcd>

# Рефлексия

# Рефлексия



Как Вам Patroni ?

**Заполните, пожалуйста,  
опрос о занятии  
по ссылке в чате**

Спасибо за внимание!

# Продолжение следует ...



Коробков Виктор

