

# PostgreSQL для администраторов баз данных и разработчиков



# Защита проекта

**Тема:** «Создание и тестирование высоконагруженного отказоустойчивого кластера PostgreSQL на базе Patroni»



**Александр Мыскин**

SRE - инженер

# План защиты

Цель и задачи проекта

Какие технологии использовались

Что получилось

Выводы

Вопросы и рекомендации

# Цель и задачи проекта

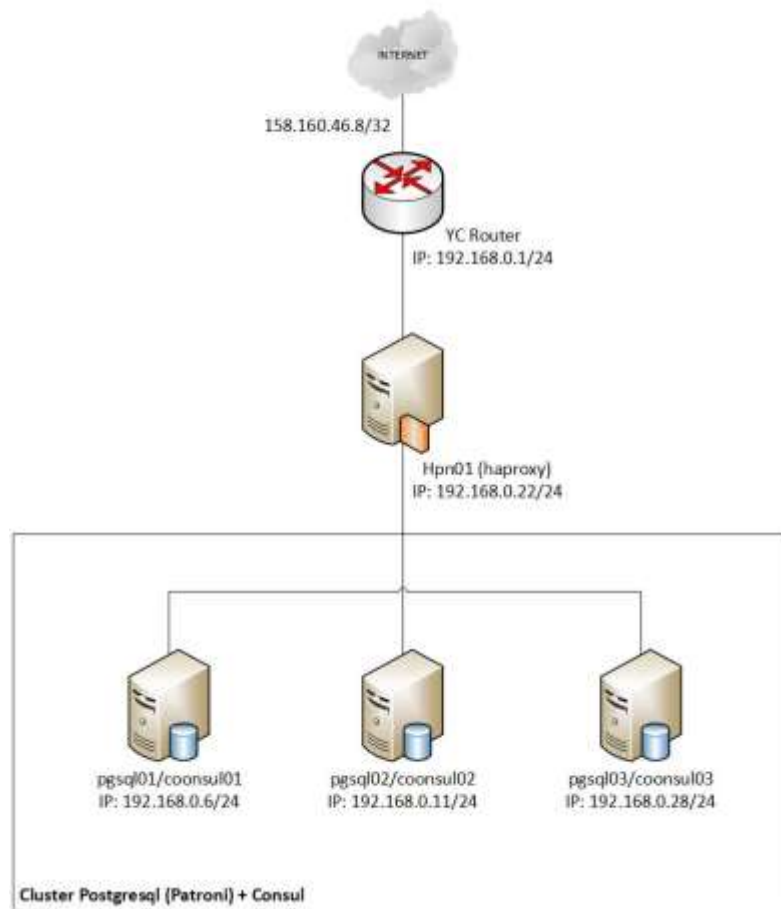
**Цель проекта: создать автоматизированную установку кластера PostgreSQL на базе Patroni и протестировать его на отказоустойчивость**

- 1. Развернуть виртуальные машины в Yandex Cloud**
- 2. Установить кластер ключей Consul**
- 3. Установить и настроить кластера PostgreSQL на базе Patroni**
- 4. Установить и настроить Nginx**
- 5. Протестировать кластер на отказоустойчивость**

# Какие технологии использовались

1. **Yandex Cloud**
2. **Consul**
3. **PostgreSQL**
4. **Patroni**
5. **HAProxy**

## Схема построения кластера PostgreSQL На базе Patroni



# Репозиторий проекта на GitHUB

Конфигурационные файлы установки можно увидеть тут:

[https://github.com/mrAlexbody/otus\\_postgre-dba-2024-07/tree/main/Project](https://github.com/mrAlexbody/otus_postgre-dba-2024-07/tree/main/Project)



# Этапы настройки

## 1. Создание сетевую инфраструктуру в Yandex Cloud

```
supervisor@LAPTOP:~$ yc vpc network create --name otus-net --description "otus-net"
id: enpmren6u9oe7acuac9v
folder_id: b1gtl3nfrisb20i58jtb
created_at: "2024-12-22T08:08:52Z"
name: otus-net
description: otus-net
default_security_group_id: enpsrv5qulvr9tcj400j
```

```
a_myskin@LAPTOP-DASh-F15:~$ yc vpc network list
```

ID	NAME
enpmren6u9oe7acuac9v	otus-net





# Этапы настройки

## 1. Создание сетевую инфраструктуру в Yandex Cloud

- Создание подсети

```
supervisor@LAPTOP:~$ yc vpc subnet create --name otus-subnet --range 192.168.0.0/24 --network-name otus-net --description "otus-subnet"
id: e9b2e8chn2srkmr8e2tq
folder_id: b1gtl3nfrisb20i58jtb
created_at: "2024-12-22T08:11:28Z"
name: otus-subnet
description: otus-subnet
network_id: enpmren6u9oe7acuac9v
zone_id: ru-central1-a
v4_cidr_blocks:
  - 192.168.0.0/24
```

```
supervisor@LAPTOP:~$ yc vpc subnet list
```

ID	NAME	NETWORK ID	ROUTE TABLE ID	ZONE	RANGE
e9b2e8chn2srkmr8e2tq	otus-subnet	enpmren6u9oe7acuac9v		ru-central1-a	[192.168.0.0/24]

# Этапы настройки

## 1. Создание сетевую инфраструктуру в Yandex Cloud

- Создание подсети
- Настройка DNS зоны

```
supervisor@LAPTOP:~$ yc dns zone create --name otus-dns --zone staging. --private-visibility network-ids="e9b2e8chn2srkmr8e2tq"
id: dns5epssmqvtite5tlvb
folder_id: b1gtl3nfrisb20i58jtb
created_at: "2024-12-22T08:14:50.828Z"
name: otus-dns
zone: staging.
private_visibility: {}

supervisor@LAPTOP:~$ yc dns zone list
```

ID	NAME	ZONE	VISIBILITY	DESCRIPTION
dns5epssmqvtite5tlvb	otus-dns	staging.		

# Этапы настройки

## 2. Развёртывание 3-х виртуальных машин для PostgreSQL

ID	NAME	ZONE ID	STATUS	EXTERNAL IP	INTERNAL IP
fhm7qul3v52bp3k6loc9	pgsql3	ru-central1-a	RUNNING	51.250.13.152	192.168.0.28
fhmg2q9speqs0d6jaa2k	pgsql1	ru-central1-a	RUNNING	51.250.92.84	192.168.0.6
fhmu5e662752dee46e4n	pgsql2	ru-central1-a	RUNNING	51.250.76.168	192.168.0.11

# Этапы настройки

## 3. Установка и настройка кластера ключей Consul

- Установка Consul на все машины вирт. машины
- Создание учётной записи, от которой будет работать сервис Consul
- Создание конфигурационного файла для Consul(a)

```
root@pgsqli:~# cat /etc/consul.d/config.json
{
  "bind_addr": "0.0.0.0",
  "bootstrap_expect": 3,
  "client_addr": "0.0.0.0",
  "data_dir": "/var/lib/consul",
  "enable_script_checks": true,
  "acl_config": {
    "enable_truncate": true,
    "only_passing": true
  },
  "enable_syslog": true,
  "encrypt": "*****",
  "leave_on_terminate": true,
  "log_level": "INFO",
  "rejoin_after_leave": true,
  "retry_join": [
    "pgsqli",
    "pgsqli",
    "pgsqli"
  ],
  "server": true,
  "start_join": [
    "pgsqli",
    "pgsqli",
    "pgsqli"
  ],
  "ui_config": { "enabled": true }
}
```

```
out@pgsqli:~# cat /etc/consul.d/config.json
{
  "bind_addr": "0.0.0.0",
  "bootstrap_expect": 3,
  "client_addr": "0.0.0.0",
  "data_dir": "/var/lib/consul",
  "enable_script_checks": true,
  "acl_config": {
    "enable_truncate": true,
    "only_passing": true
  },
  "enable_syslog": true,
  "encrypt": "*****",
  "leave_on_terminate": true,
  "log_level": "INFO",
  "rejoin_after_leave": true,
  "retry_join": [
    "pgsqli",
    "pgsqli",
    "pgsqli"
  ],
  "server": true,
  "start_join": [
    "pgsqli",
    "pgsqli",
    "pgsqli"
  ],
  "ui_config": { "enabled": true }
}
```

```
root@pgsqli:~# cat /etc/consul.d/config.json
{
  "bind_addr": "0.0.0.0",
  "bootstrap_expect": 3,
  "client_addr": "0.0.0.0",
  "data_dir": "/var/lib/consul",
  "enable_script_checks": true,
  "acl_config": {
    "enable_truncate": true,
    "only_passing": true
  },
  "enable_syslog": true,
  "encrypt": "*****",
  "leave_on_terminate": true,
  "log_level": "INFO",
  "rejoin_after_leave": true,
  "retry_join": [
    "pgsqli",
    "pgsqli",
    "pgsqli"
  ],
  "server": true,
  "start_join": [
    "pgsqli",
    "pgsqli",
    "pgsqli"
  ],
  "ui_config": { "enabled": true }
}
```

# Этапы настройки

## 3. Установка и настройка кластера ключей Consul

- Установка Consul на все машины вирт. машины
- Создание учётной записи, от которой будет работать сервис Consul
- Создание конфигурационного файла для Consul(a)
- Включение авторизации в Consul для входа на сайт и работы из командной строке

```
...  
"ui_config": { "enabled": true },  
  "acl": {  
    "enabled": true,  
    "default_policy": "deny",  
    "enable_token_persistence": true  
  }  
....
```

# Этапы настройки

## 3. Установка и настройка кластера ключей Consul

- Установка Consul на все машины вирт. машины
- Создание учётной записи, от которой будет работать сервис Consul
- Создание конфигурационного файла для Consul(a)
- Включение авторизации в Consul для входа на сайт и работы из командной строке

```
root@pgsql1:~# consul acl bootstrap
```

# Этапы настройки

## 3. Установка и настройка кластера ключей Consul

- Установка Consul на все машины вирт. машины
- Создание учётной записи, от которой будет работать сервис Consul
- Создание конфигурационного файла для Consul(a)
- Включение авторизации в Consul для входа на сайт и работы из командной строке

```
root@pgsql1:~# export CONSUL_HTTP_TOKEN=*****_****_****_****_*****
root@pgsql2:~# export CONSUL_HTTP_TOKEN=*****_****_****_****_*****
root@pgsql3:~# export CONSUL_HTTP_TOKEN=*****_****_****_****_*****
```

# Этапы настройки

## 3. Установка и настройка кластера ключей Consul

- Установка Consul на все машины вирт. машины
- Создание учётной записи, от которой будет работать сервис Consul
- Создание конфигурационного файла для Consul(a)
- Включение авторизации в Consul для входа на сайт и работы из командной строке

```
yc-user@pgsql3:~$ consul members
```

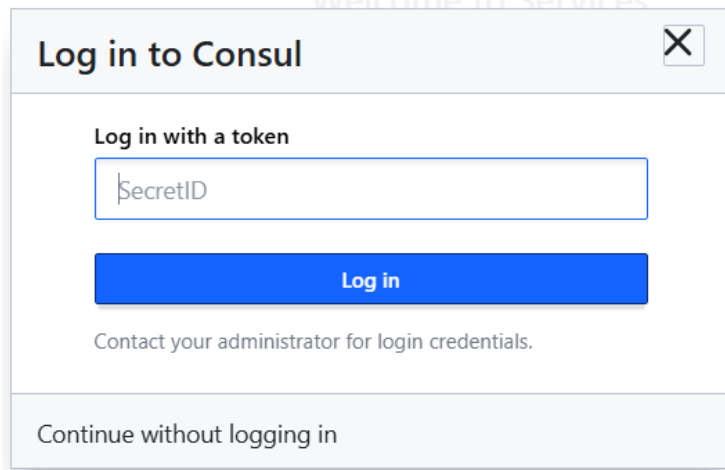
Node	Address	Status	Type	Build	Protocol	DC	Partition	Segment
consul1	192.168.0.6:8301	alive	server	1.14.3	2	yc1a	default	<all>
consul2	192.168.0.11:8301	alive	server	1.14.3	2	yc1a	default	<all>
consul3	192.168.0.28:8301	alive	server	1.14.3	2	yc1a	default	<all>



# Этапы настройки

## 3. Установка и настройка кластера ключей Consul

- Установка Consul на все машины вирт. машины
- Создание учётной записи, от которой будет работать сервис Consul
- Создание конфигурационного файла для Consul(a)
- Включение авторизации в Consul для входа на сайт и работы из командной строке



The image shows a 'Log in to Consul' dialog box. It has a title bar with a close button (X). The main content area is titled 'Log in with a token' and contains a text input field with the placeholder text 'SecretID'. Below the input field is a blue 'Log in' button. At the bottom of the dialog, there is a link that says 'Continue without logging in'. A faint watermark 'WELCOME TO SERVICES' is visible in the background of the dialog box.

# Этапы настройки

## 4. Установка и настройка кластера PostgreSQL на базе Patroni

- Установка PostgreSQL
- Установка Patroni
- Настройка конфигурационного файла Patroni

```
root@pgsql1:~# cat /etc/patroni.yml
scope: patroni
name: pgsql1
restapi:
  listen: 192.168.0.6:8008
  connect_address: 192.168.0.6:8008
consul:
  host: "localhost:8500"
  register_service: true
  token: consul-acl-token
bootstrap:
  dcs:
    ttl: 30s
    loop_wait: 10s
    retry_timeout: 10s
    maximum_lag_on_follower: 1048576
    postgresql:
      use_pg_rewind: true
    parameters:
  initdb:
    - encoding: UTF8
    - data-checksums
  pg_hba:
    - host replication replicator 192.168.0.0/24 md5
    - host all all 192.168.0.0/24 md5
  users:
    admin:
      password: admin_121
      options:
        - createrole
        - createdb
```

```
postgresql:
  listen: 127.0.0.1, 192.168.0.6:5432
  connect_address: 192.168.0.6:5432
  data_dir: /var/lib/postgresql/14/main
  bin_dir: /usr/lib/postgresql/14/bin
  pgpass: /tmp/pgpass0
  authentication:
    replication:
      username: replicator
      password: *****
    superuser:
      username: postgres
      password: *****
    rewind:
      username: rewind_user
      password: *****
  parameters:
    unix_socket_directories: .
  tags:
    nofailover: false
    noloadbalance: false
    clonefrom: false
    nosync: false
```

# Этапы настройки

## 4. Установка и настройка кластера PostgreSQL на базе Patroni

- Установка PostgreSQL
- Установка Patroni
- Настройка конфигурационного файла Patroni
- Проверка конфигурации кластера

```
+ Cluster: patroni (7451367085296931935) -----+-----+
| Member | Host      | Role   | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| pgsql1  | 192.168.0.6 | Replica | streaming | 1 | 0 |
| pgsql2  | 192.168.0.11 | Replica | streaming | 1 | 0 |
| pgsql3  | 192.168.0.28 | Leader  | running  | 1 |  |
+-----+-----+-----+-----+-----+-----+
+ Cluster: patroni (7451367085296931935) -----+-----+
| Member | Host      | Role   | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| pgsql1  | 192.168.0.6 | Replica | streaming | 1 | 0 |
| pgsql2  | 192.168.0.11 | Replica | streaming | 1 | 0 |
| pgsql3  | 192.168.0.28 | Leader  | running  | 1 |  |
+-----+-----+-----+-----+-----+-----+
+ Cluster: patroni (7451367085296931935) -----+-----+
| Member | Host      | Role   | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| pgsql1  | 192.168.0.6 | Replica | streaming | 1 | 0 |
| pgsql2  | 192.168.0.11 | Replica | streaming | 1 | 0 |
| pgsql3  | 192.168.0.28 | Leader  | running  | 1 |  |
+-----+-----+-----+-----+-----+-----+
```

# Этапы настройки

#### 4. Установка и конфигурация Nargoxy

- Установка Наргоху
- Настройка конфигурационного файла Наргоху

```
root@pr01:~# cat /etc/haproxy/haproxy.cfg
global
    log /dev/log local0
    log /dev/log local1 notice
    chroot /var/lib/haproxy
    stats socket /var/haproxy/admin.sock mode 660 level admin expose fd listeners
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # default ssl material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # See: https://ssl-config.mozilla.org/#server=haproxy&version=1.4.3&config=intermediate
    ssl-default-bind-ciphers SCHE-ECDSA-AES128-GCM-SHA256:ECHE-PSA-AES128-GCM-SHA256:ECHE-SCD5A-AES256-GCM
    ssl-default-bind-ciphersuites TLS_AES_128_GCM_SHA256:TLS_AES_256_GCM_SHA384:TLS_CHACHA20_POLY1305_SHA256
    ssl-default-bind-options ssl-err-ver TLSv1.2 no-tls-tickets

defaults
    log global
    mode http
    option httplog
    option dontlognull
    timeout connect 3000
    timeout client 10000
    timeout server 10000

    errorfile 400 /etc/haproxy/errors/400.http
    errorfile 401 /etc/haproxy/errors/401.http
    errorfile 403 /etc/haproxy/errors/403.http
    errorfile 500 /etc/haproxy/errors/500.http
    errorfile 502 /etc/haproxy/errors/502.http
    errorfile 503 /etc/haproxy/errors/503.http
    errorfile 504 /etc/haproxy/errors/504.http
```

# Этапы настройки

## 4. Установка и конфигурация Наргоху

- Установка Наргоху
- Настройка конфигурационного файла Наргоху

```
framed consul
bind 0.0.0.0:80
mode http
default_backend consulweb

listen postgres_write
bind 0.0.0.0:5432
mode tcp
option tcplog
option httpchk
http-check connect
http-check send meth GET url /master
http-check expect status 200
default-server http 1s fail 3 on-marked-down shutdown-sessions on-marked-up shutdown-backup-sessions
server pgsql01 192.168.0.5:5432 check port 5432
server pgsql02 192.168.0.11:5432 check port 5432
server pgsql03 192.168.0.28:5432 check port 5432

listen postgres_read
bind 0.0.0.0:5433
mode tcp
option tcplog
option httpchk
http-check send meth GET url /replica
http-check expect status 200
default-server http 1s fail 3 on-marked-down shutdown-sessions on-marked-up shutdown-backup-sessions
server pgsql01 192.168.0.5:5432 check port 5432
server pgsql02 192.168.0.11:5432 check port 5432
server pgsql03 192.168.0.28:5432 check port 5432

backend consulweb
mode http
option httpchk
option /forwarder
balance roundrobin
server consul01 192.168.0.3:8300 check
server consul02 192.168.0.11:8300 check
server consul03 192.168.0.28:8300 check
```

# Этапы настройки

## 4. Установка и конфигурация Наргоху

- Установка Наргоху
- Настройка конфигурационного файла Наргоху
- Удаление внешних IP у нод кластера PostgreSQL в Yandex Cloud









Виртуальные машины

Фильтр по имени

Все статусы

Все платформы

Все зоны доступности

<input type="checkbox"/>	Имя <sup>1</sup>	Cloud Backup	Статус <sup>1</sup>	ОС	Платформа <sup>1</sup>	vCPU	Доля vCPU	RAM	Прерываемая	Размер дисков	Зона доступности <sup>1</sup>	Внутренний IPv4	Публичный IPv4	Дата создания <sup>1</sup>
<input type="checkbox"/>	pgsql3	 Не подключён	Running		Intel Cascade Lake	2	100%	4 GB	Нет	10 GB	ru-central1-a	192.168.0.28	—	22.12.2024, в 11:19
<input type="checkbox"/>	pgsql1	 Не подключён	Running		Intel Cascade Lake	2	100%	4 GB	Нет	10 GB	ru-central1-a	192.168.0.6	—	22.12.2024, в 11:19
<input type="checkbox"/>	hpr01	 Не подключён	Running		Intel Cascade Lake	2	100%	4 GB	Нет	10 GB	ru-central1-a	192.168.0.22	158.160.46.8	24.12.2024, в 10:53
<input type="checkbox"/>	pgsql2	 Не подключён	Running		Intel Cascade Lake	2	100%	4 GB	Нет	10 GB	ru-central1-a	192.168.0.11	—	22.12.2024, в 11:19

# Этапы настройки

## 5. Проверка кластера

- Проверим состояние кластера и вручную переключим

```
root@pgsql1:~# patronictl -c /etc/patroni.yml switchover
Current cluster topology
+ Cluster: patroni (7451367085296031935) -----+
| Member | Host           | Role   | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+
| pgsql1  | 192.168.0.6    | Replica | streaming | 1 | 0 |
| pgsql2  | 192.168.0.11   | Replica | streaming | 1 | 0 |
| pgsql3  | 192.168.0.20   | Leader  | running  | 1 |  |
+-----+-----+-----+-----+-----+

Primary [pgsql3]:
Candidate ['pgsql1', 'pgsql2'] []:
When should the switchover take place (e.g. 2024-12-24T11:20 ) [now]:
Are you sure you want to switchover cluster patroni, demoting current leader pgsql3? [y/N]: y
2024-12-24 10:21:14.396681 Successfully switched over to "pgsql1"
+ Cluster: patroni (7451367085296031935) -----+
| Member | Host           | Role   | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+
| pgsql1  | 192.168.0.6    | Leader  | running  | 1 |  |
| pgsql2  | 192.168.0.11   | Replica | running  | 1 |  |
| pgsql3  | 192.168.0.20   | Replica | stopped  |  | unknown |
+-----+-----+-----+-----+-----+

root@pgsql1:~# patronictl -c /etc/patroni.yml list
+ Cluster: patroni (7451367085296031935) -----+
| Member | Host           | Role   | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+
| pgsql1  | 192.168.0.6    | Leader  | running  | 1 |  |
| pgsql2  | 192.168.0.11   | Replica | streaming | 2 | 0 |
| pgsql3  | 192.168.0.20   | Replica | streaming | 2 | 0 |
+-----+-----+-----+-----+-----+
```

# Этапы настройки

## 5. Проверка кластера

- Проверим состояние кластера и вручную переключим
- Остановим одну ноду pgsql01 (Master) кластера

```
root@pgsql3:~# patronictl -c /etc/patroni.yml list
+ Cluster: patroni (7451367085296931935) -----+-----+-----+
| Member | Host          | Role   | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| pgsql2  | 192.168.0.11  | Leader | running | 3  |           |
| pgsql3  | 192.168.0.28  | Replica | streaming | 3  | 0         |
+-----+-----+-----+-----+-----+-----+
```



# Этапы настройки

## 5. Проверка кластера

- Проверим состояние кластера и вручную переключим
- Остановим одну ноду pgsql01 (Master) кластера
- Запустим обратно ноду pgsql01 кластера и проверим

```
root@pgsql3:~# patronictl -c /etc/patroni.yml list
+ Cluster: patroni (7451367085296931935) -----+-----+-----+
| Member | Host          | Role   | State   | TL | Lag in MB |
+-----+-----+-----+-----+-----+-----+
| pgsql1  | 192.168.0.6   | Replica | streaming | 3 | 0 |
| pgsql2  | 192.168.0.11  | Leader  | running  | 3 |  |
| pgsql3  | 192.168.0.28  | Replica | streaming | 3 | 0 |
+-----+-----+-----+-----+-----+-----+
```

# Этапы настройки

## 5. Проверка кластера

- Проверим состояние кластера и вручную переключим
- Остановим одну ноду pgsql01 (Master) кластера
- Запустим обратно ноду pgsql01 кластера и проверим
- Создадим БД на ноде Master

```
postgres=# CREATE DATABASE testdb;  
CREATE DATABASE  
postgres=# \l
```

```
                List of databases  
  Name      | Owner   | Encoding | Collate  | Ctype    | Access privileges  
-----+-----+-----+-----+-----+-----  
postgres    | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 |  
template0   | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 | =c/postgres      +  
              |          |           |              |              | postgres=CTc/postgres  
template1   | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 | =c/postgres      +  
              |          |           |              |              | postgres=CTc/postgres  
test         | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 |  
testdb      | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 |  
(5 rows)
```

# Этапы настройки

## 5. Проверка кластера

- Проверим состояние кластера и вручную переключим
- Остановим одну ноду pgsql01 (Master) кластера
- Запустим обратно ноду pgsql01 кластера и проверим
- Создадим БД на ноде Master
- Проверим на Replic

```
root@pgsql1:~# psql -h 127.0.0.1 -U postgres
psql (14.15 (Ubuntu 14.15-1.pgdg20.04+1))
Type "help" for help.
```

```
postgres=# \l
```

```
          List of databases
  Name      | Owner   | Encoding | Collate | Ctype   | Access privileges
-----+-----+-----+-----+-----+-----
 postgres   | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 |
 template0  | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 | =c/postgres      +
            |          |           |             |             | postgres=CTc/postgres
 template1  | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 | =c/postgres      +
            |          |           |             |             | postgres=CTc/postgres
 test       | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 |
 testdb     | postgres | UTF8      | en_US.UTF-8 | en_US.UTF-8 |
(5 rows)
```

# Вопросы и рекомендации

+ если есть вопросы

— если вопросов нет



Спасибо за внимание! ✨

