

**Министерство науки и высшего образования Российской Федерации**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ**  
**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**  
**ITMO University**

# Лабораторная работа №1

## HA Postgres Cluster

**Обучающиеся** Колмогорова А.С., Тихонова М.С., Шевченко С.О.

Преподаватель Самохин Н.Ю.

Санкт-Петербург, 2025

## Содержание

1 Цель работы.....	3
2 Ход работы.....	4
3 Ответы на вопросы.....	18
4 Вывод.....	18

## **1 Цель работы**

В рамках лабораторной работы необходимо развернуть и настроить высокодоступный кластер Postgres.

## 2 Ход работы

Код из описания лабораторной с небольшими модификациями (продублировала pg\_hba в postgresql часть файла, обновила wal\_keep\_segments на wal\_keep\_size и т.д.)

Докерфайл:

```
FROM postgres:15
RUN apt-get update -y && \
    apt-get install -y netcat-openbsd python3-pip curl python3-psycopg2
python3-venv iputils-ping
RUN python3 -m venv /opt/patroni-venv && \
    /opt/patroni-venv/bin/pip install --upgrade pip && \
    /opt/patroni-venv/bin/pip install patroni[zookeeper] psycopg2-binary
COPY postgres0.yml /postgres0.yml
COPY postgres1.yml /postgres1.yml
ENV PATH="/opt/patroni-venv/bin:$PATH"
USER postgres
```

Докер компоуз:

```
version: '3.8'
services:
  pg-master:
    build: .
    image: localhost/postgres:patroni
    container_name: pg-master
    restart: no
    hostname: pg-master
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
      PGDATA: /var/lib/postgresql/data/pgdata
    expose:
      - "8008"
    ports:
      - "5433:5432"
    volumes:
      - pg-master:/var/lib/postgresql/data
    command: patroni /postgres0.yml
```

```

pg-slave:
  build: .
  image: localhost/postgres:patroni
  container_name: pg-slave
  restart: no
  hostname: pg-slave
  environment:
    POSTGRES_USER: postgres
    POSTGRES_PASSWORD: postgres
    PGDATA: /var/lib/postgresql/data/pgdata
  expose:
    - "8008"
  ports:
    - "5434:5432"
  volumes:
    - pg-slave:/var/lib/postgresql/data
  command: patroni /postgres1.yml

zoo:
  image: confluentinc/cp-zookeeper:7.7.1
  container_name: zoo
  restart: always
  hostname: zoo
  ports:
    - "2181:2181"
  environment:
    ZOOKEEPER_CLIENT_PORT: 2181
    ZOOKEEPER_TICK_TIME: 2000

volumes:
  pg-master:
  pg-slave:

```

## Постгресы

```

scope: my_cluster
name: postgresql0
restapi:
  listen: pg-master:8008
  connect_address: pg-master:8008
zookeeper:
  hosts:
    - zoo:2181

```

```
bootstrap:
  dcs:
    ttl: 30
    loop_wait: 10
    retry_timeout: 10
    maximum_lag_on_failover: 10485760
    master_start_timeout: 300
    synchronous_mode: true
  postgresql:
    use_pg_rewind: true
    use_slots: true
    parameters:
      wal_level: replica
      hot_standby: "on"
      wal_keep_size: 128MB
      max_wal_senders: 10
      max_replication_slots: 10
      wal_log_hints: "on"
      archive_mode: "always"
      archive_timeout: 1800s
      archive_command: mkdir -p /tmp/wal_archive && test ! -f
/tmp/wal_archive/%f && cp %p /tmp/wal_archive/%f
    pg_hba:
      - host replication replicator 0.0.0.0/0 md5
      - host all all 0.0.0.0/0 md5
postgresql:
  listen: 0.0.0.0:5432
  connect_address: pg-master:5432
  data_dir: /var/lib/postgresql/data/postgresql0
  bin_dir: /usr/lib/postgresql/15/bin
  pgpass: /tmp/pgpass0
  authentication:
    replication:
      username: replicator
      password: rep-pass
    superuser:
      username: postgres
      password: postgres
  pg_hba:
    - host replication replicator 0.0.0.0/0 md5
    - host all all 0.0.0.0/0 md5
  parameters:
    unix_socket_directories: '.'
```

```
watchdog:
```

```
  mode: off
```

```
tags:
```

```
  nofailover: false
```

```
  noloadbalance: false
```

```
  clonefrom: false
```

```
  nosync: false
```

```
scope: my_cluster # Имя нашего кластера
```

```
name: postgresql1 # Имя второй ноды
```

```
restapi: # Адреса второй ноды
```

```
  listen: pg-slave:8008
```

```
  connect_address: pg-slave:8008
```

```
zookeeper:
```

```
  hosts:
```

```
    - zoo:2181 # Адрес Zookeeper
```

```
bootstrap:
```

```
  dcs:
```

```
    ttl: 30
```

```
    loop_wait: 10
```

```
    retry_timeout: 10
```

```
    maximum_lag_on_failover: 10485760
```

```
    master_start_timeout: 300
```

```
    synchronous_mode: true
```

```
  postgresql:
```

```
    use_pg_rewind: true
```

```
    use_slots: true
```

```
    parameters:
```

```
      wal_level: replica
```

```
      hot_standby: "on"
```

```
      wal_keep_size: 128MB
```

```
      max_wal_senders: 10
```

```
      max_replication_slots: 10
```

```
      wal_log_hints: "on"
```

```
      archive_mode: "always"
```

```
      archive_timeout: 1800s
```

```
      archive_command: mkdir -p /tmp/wal_archive && test ! -f
```

```
/tmp/wal_archive/%f && cp %p /tmp/wal_archive/%f
```

```
    pg_hba:
```

```
      - host replication replicator 0.0.0.0/0 md5
```

```
      - host all all 0.0.0.0/0 md5
```

```
postgresql:
  listen: 0.0.0.0:5432
  connect_address: pg-slave:5432 # Адрес второй ноды
  data_dir: /var/lib/postgresql/data/postgresql1 # Место хранения
данных второй ноды
  bin_dir: /usr/lib/postgresql/15/bin
  pgpass: /tmp/pgpass1
  authentication:
    replication: # логотип для репликации, при желании можно поменять
      username: replicator
      password: rep-pass
    superuser: # админский логотип, при желании можно поменять (в том
числе в файле compose)
      username: postgres
      password: postgres
  pg_hba:
    - host replication replicator 0.0.0.0/0 md5
    - host all all 0.0.0.0/0 md5
  parameters:
    unix_socket_directories: '.'

watchdog:
  mode: off

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



## 1. Деплоим написанные конфигурации

```
[+] Building 1/1
✓localhost/postgres:patroni Built
PS C:\Users\Marina\itmo\net_adm> docker-compose up -d
time="2025-12-23T04:07:44+03:00" level=warning msg="C:\\Us
'version' is obsolete, it will be ignored, please remove it
[+] Running 14/14
✓zoo Pulled
✓878348106a95 Pull complete
✓732c9ebb730c Pull complete
✓00ded6dd259e Pull complete
✓e7688095d1e6 Pull complete
✓8eab815b3593 Pull complete
✓296f622c8150 Pull complete
✓ed746366f1b8 Pull complete
✓10894799ccd9 Pull complete
✓8d377259558c Pull complete
✓4ee3050cff6b Pull complete
✓98acab318002 Pull complete
✓56f27190e824 Pull complete
✓8e70b9b9b078 Pull complete
[+] Running 6/6
✓Network net_adm_default Created
✓Volume net_adm_pg-master Created
✓Volume net_adm_pg-slave Created
✓Container zoo Started
✓Container pg-slave Started
✓Container pg-master Started
PS C:\Users\Marina\itmo\net_adm> |
8008
:
5434.5432"
```

2. Проверяем по логам, где мастер-сегмент. В нашем случае, они поменялись местами (по названиям)

```
2025-12-23 01:39:04.009 UTC [28] FATAL: the database system is starting up
localhost:5432 - rejecting connections
2025-12-23 01:39:04.202 UTC [25] LOG: entering standby mode
2025-12-23 01:39:04.202 UTC [25] LOG: starting backup recovery with redo LSN 0/2000028, checkpoint LSN 0/2000060, on timeline ID 1
2025-12-23 01:39:04.207 UTC [25] LOG: redo starts at 0/2000028
2025-12-23 01:39:04.210 UTC [25] LOG: completed backup recovery with redo LSN 0/2000028 and end LSN 0/2000100
2025-12-23 01:39:04.210 UTC [25] LOG: consistent recovery state reached at 0/2000100
2025-12-23 01:39:04.210 UTC [21] LOG: database system is ready to accept read-only connections
2025-12-23 01:39:04.225 UTC [29] LOG: started streaming WAL from primary at 0/3000000 on timeline 1
localhost:5432 - accepting connections
2025-12-23 01:39:05,043 INFO: Lock owner: postgresql1; I am postgresql0
2025-12-23 01:39:05,043 INFO: establishing a new patroni heartbeat connection to postgres
2025-12-23 01:39:05,097 INFO: no action. I am (postgresql0), a secondary, and following a leader (postgresql1)
2025-12-23 01:39:15,049 INFO: no action. I am (postgresql0), a secondary, and following a leader (postgresql1)
2025-12-23 01:39:26,946 INFO: no action. I am (postgresql0), a secondary, and following a leader (postgresql1)
PS C:\Users\Marina\itmo\net_adm> |
```

### 3. А вот и мастер (логи)!

```
24 kB, estimate=11224 kB
2025-12-23 01:39:03.336 UTC [47] WARNING: skipping special file "./.s.PGSQL.5432"
2025-12-23 01:39:03.343 UTC [47] WARNING: skipping special file "./.s.PGSQL.5432"
2025-12-23 01:39:04.028 INFO: Lock owner: postgresql1; I am postgresql1
2025-12-23 01:39:04.042 INFO: Enabled synchronous replication
2025-12-23 01:39:04.063 INFO: no action. I am (postgresql1), the leader with the lock
2025-12-23 01:39:14.030 INFO: Lock owner: postgresql1; I am postgresql1
2025-12-23 01:39:14.043 INFO: Assigning synchronous standby status to ['postgresql0']
server signaled
2025-12-23 01:39:14.045 UTC [32] LOG: received SIGHUP, reloading configuration files
2025-12-23 01:39:14.046 UTC [32] LOG: parameter "synchronous_standby_names" changed to "postgresql0"
2025-12-23 01:39:14.250 UTC [49] LOG: standby "postgresql0" is now a synchronous standby with priority 1
2025-12-23 01:39:14.250 UTC [49] STATEMENT: START_REPLICATION SLOT "postgresql0" 0/30000000 TIMELINE 1
2025-12-23 01:39:18.054 INFO: Synchronous standby status assigned to ['postgresql0']
2025-12-23 01:39:18.058 INFO: no action. I am (postgresql1), the leader with the lock
```

4. Подключаемся к первому инстансу с помощью psql, уже изнутри проверяем, какую роль исполняет сегмент. Получаем ответом t, значит, мы на слейве.

```

PS C:\Users\Marina\itmo\net_adm> psql -h localhost -p 5433
-U postgres -d postgres
Пароль пользователя postgres:

psql (17.5, сервер 15.15 (Debian 15.15-1.pgdg13+1))
ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается
от основной
                страницы Windows (1251).
                8-битовые (русские) символы могут отобража
                ться некорректно.
                Подробнее об этом смотрите документацию ps
                ql, раздел
                "Notes for Windows users".
Введите "help", чтобы получить справку.

postgres=# SELECT pg_is_in_recovery();
 pg_is_in_recovery
-----
 t
(1 row)

```

5. Пробуем создать таблицу на нем. Ожидаемо, не получилось, отсюда можно только читать!

```

postgres=# CREATE TABLE test_rep (
postgres(# id SERIAL PRIMARY KEY,
postgres(# name TEXT NOT NULL
postgres(# );
ERROR:  cannot execute CREATE TABLE in a read-only transac
tion

```

6. Прodelываем ту же процедуру с мастером. Получаем f - действительно мастер.

```
postgres=# \q
PS C:\Users\Marina\itmo\net_adm> psql -h localhost -p 5434
-U postgres -d postgres
Пароль пользователя postgres:

psql (17.5, сервер 15.15 (Debian 15.15-1.pgdg13+1))
ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается
от основной
                страницы Windows (1251).
                8-битовые (русские) символы могут отобража
тятся некорректно.
                Подробнее об этом смотрите документацию ps
ql, раздел
                "Notes for Windows users".
Введите "help", чтобы получить справку.

postgres=# SELECT pg_is_in_recovery();
 pg_is_in_recovery 
-----
 f
(1 ò€Ёър)
```

7. Успешно создаем таблицу на мастере и вставляем в нее запись.

```
postgres=# CREATE TABLE test (
postgres(# id SERIAL PRIMARY KEY,
postgres(# name TEXT NOT NULL
postgres(# );
CREATE TABLE
postgres=# INSERT INTO test (name) VALUES ('Hello from mas
ter!');
INSERT 0 1
postgres=# SELECT * FROM test;
 id |          name
----+-----
  1 | Hello from master!
(1 ò€Ёър)
```

8. Проверяем слейва. На него все реплицировалось!

```
PS C:\Users\Marina\itmo\net_adm> psql -h localhost -p 5433
-U postgres -d postgres
Пароль пользователя postgres:

psql (17.5, сервер 15.15 (Debian 15.15-1.pgdg13+1))
ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается
от основной
                страницы Windows (1251).
                8-битовые (русские) символы могут отобража
                ться некорректно.
                Подробнее об этом смотрите документацию ps
                ql, раздел
                "Notes for Windows users".
Введите "help", чтобы получить справку.

postgres=# SELECT * FROM test;
 id |          name
----+-----
  1 | Hello from master!
(1 ò€Ёюър)
```

9. Тестируем фейловер. Отключаем мастер-сегмент.

```
postgres=# \q
PS C:\Users\Marina\itmo\net_adm> docker stop pg-slave
pg-slave
```

10. Проверяем логи бывшего слейва. Он повысился до мастера, увидев, что старый мастер лег!

```
2025-12-23 02:32:39,089 INFO: promoted self to leader by a
acquiring session lock
2025-12-23 02:32:39.093 UTC [26] LOG:  received promote re
quest
2025-12-23 02:32:39.093 UTC [26] LOG:  redo done at 0/3079
8D8 system usage: CPU: user: 0.00 s, system: 0.01 s, elaps
ed: 365.93 s
2025-12-23 02:32:39.093 UTC [26] LOG:  last completed tran
saction was at log time 2025-12-23 02:31:11.589237+00
server promoting
2025-12-23 02:32:39.101 UTC [26] LOG:  selected new timeli
ne ID: 3
2025-12-23 02:32:39.216 UTC [26] LOG:  archive recovery co
mplete
2025-12-23 02:32:39.226 UTC [24] LOG:  checkpoint starting
: force
2025-12-23 02:32:39.231 UTC [22] LOG:  database system is
ready to accept connections
2025-12-23 02:32:39.247 UTC [24] LOG:  checkpoint complete
: wrote 2 buffers (0.0%); 0 WAL file(s) added, 0 removed,
0 recycled; write=0.005 s, sync=0.003 s, total=0.021 s; sy
nc files=2, longest=0.002 s, average=0.002 s; distance=0 k
B, estimate=166 kB
2025-12-23 02:32:40,125 INFO: no action. I am (postgresql0
), the leader with the lock
2025-12-23 02:32:50,116 INFO: no action. I am (postgresql0
), the leader with the lock
2025-12-23 02:33:00,112 INFO: no action. I am (postgresql0
), the leader with the lock
```

11. Вносим корректировки, чтобы добавить балансировщик (добавляем в компоуз и создаем новый файл)

```
global
maxconn 100
defaults
log global
mode tcp
retries 3
timeout client 30m
timeout connect 4s
timeout server 30m
timeout check 5s
listen stats
```

```

mode http
bind *:7000
stats enable
stats uri /
listen postgres
bind *:5432 # Выбранный порт из docker-compose.yml
option httpchk
http-check expect status 200 # Описываем нашу проверку доступности (в
данном случае обычный HTTP-пинг)
default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
server postgresql_pg_master_5432 pg-master:5432 maxconn 100 check port
8008 # Адрес первой ноды постгреса
server postgresql_pg_slave_5432 pg-slave:5432 maxconn 100 check port
8008 # Адрес второй ноды постгреса

```

12. Для разнообразия посмотрим, что поднялся контейнер с хайпрокси через Docker Desktop

<input type="checkbox"/>	▼	net_adm	-	-	-	1.85%	5 second		:	
<input type="checkbox"/>		zoo	457a3a5fab22	<a href="#">confluentinc</a>	<a href="#">2181:2181</a>	1.51%	2 minut		:	
<input type="checkbox"/>		postgres_entr	4b36a87efa90	<a href="#">haproxy:3.0</a>	<a href="#">5432:5432</a> <a href="#">Show all ports (2)</a>	0%	6 second		:	
<input type="checkbox"/>		pg-slave	89f7f80f87a8	<a href="#">postgres:pg</a>	<a href="#">5434:5432</a>	0.22%	6 second		:	
<input type="checkbox"/>		pg-master	ff76e6beeb6f	<a href="#">postgres:pg</a>	<a href="#">5433:5432</a>	0.12%	6 second		:	

13. Подключение успешно

```

PS C:\Users\Marina\itmo\net_adm> psql -h localhost -p 5432
-U postgres -d postgres
Пароль пользователя postgres:

psql (17.5, сервер 15.15 (Debian 15.15-1.pgdg13+1))
ПРЕДУПРЕЖДЕНИЕ: Кодовая страница консоли (866) отличается
от основной
                    страницы Windows (1251).
                    8-битовые (русские) символы могут отобража
тятся некорректно.
                    Подробнее об этом смотрите документацию ps
ql, раздел
                    "Notes for Windows users".
Введите "help", чтобы получить справку.

postgres=# |

```



14. И это даже мастер, все как надо

```
postgres=# SELECT pg_is_in_recovery();
pg_is_in_recovery
-----
f
(1 row)
```

### 3 Ответы на вопросы

Порты 8008 и 5432 вынесены в разные директивы, expose и ports. По сути, если записать 8008 в ports, то он тоже станет exposed. В чем разница?

- expose - внутренняя коммуникация между контейнерами, а ports - для доступа с хоста. Если стоит ports, то expose для того же порта избыточен.

При обычном перезапуске композ-проекта, будет ли сбилден заново образ? А если предварительно отредактировать файлы postgresX.yml? А если содержимое самого Dockerfile? Почему?

- Нет, образ заново не сбилдится. Перезапуск компоуз-проекта просто перезапустит контейнеры, используя уже готовый образ. Изменения в файлах замечены не будут.

### 4 Вывод

В ходе данной лабораторной нам удалось ближе познакомиться с механизмом кластеризации БД и работой балансировщиков, а также лишний раз потренироваться в работе с Docker и Postgresql. Первоначальная цель работы “развернуть и настроить высокодоступный кластер Postgres” выполнена!