

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,  
МЕХАНИКИ И ОПТИКИ ФАКУЛЬТЕТ  
ИНФОКОММУНИКАЦИОННЫХ  
ТЕХНОЛОГИЙ

Отчет  
по курсу «Администрирование компьютерных сетей»  
Лабораторная 3

Выполнили:  
Дьячкова А., КОМСЕТ 3.1  
Сеничев С., КОМСЕТ 3.1

Санкт-Петербург  
2024 г.

## Часть 1 Поднимаем Postgres

Устанавливаем docker-compose (docker уже установлен)

```
root@copium-box ~ (1.275s)
sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Dload  % Upload   Total   Spent    Left     Speed
  0     0    0     0    0     0      0      0  --:--:-- --:--:-- --:--:--    0
100 12.1M 100 12.1M    0     0 10.1M    0  0:00:01 0:00:01 --:--:-- 69.0M

root@copium-box ~ (0.046s)
sudo chmod +x /usr/local/bin/docker-compose

root@copium-box ~ (0.702s)
docker-compose --version
docker-compose version 1.29.2, build 5becea4c
```

Создаем в директории файлы Dockerfile, конфигурацию для двух нод (postgres0.yml, postgres1.yml), compose-файл. В postgres1.yml меняем ряд настроек на pg-slave и переименовываем директорию для хранения данных.

Запускаем кластер постгреса, проверяем, что кластер запущен и работает

```
Status: Downloaded newer image for ...
Creating pg-master ... done
Creating zoo ... done
Creating pg-slave ... done

root@copium-box ~/lab3 (0.841s)
docker-compose ps

```

Name	Command	State	Ports
pg-master	docker-entrypoint.sh patro ...	Up	0.0.0.0:5433->5432/tcp, :::5433->5432/tcp, 8008/tcp
pg-slave	docker-entrypoint.sh patro ...	Up	0.0.0.0:5434->5432/tcp, :::5434->5432/tcp, 8008/tcp
zoo	/etc/confluent/docker/run	Up	0.0.0.0:2181->2181/tcp, :::2181->2181/tcp, 2888/tcp, 3888/tcp

## Часть 2. Проверяем репликацию

Выполняем подключение к обоим нодам через psql. Попробуем создать таблицу данных test\_table и добавить записи в таблицу через pg-master:

```
root@copium-box ~/lab3
psql -h 127.0.0.1 -p 5433 -U postgres -d postgres
Password for user postgres:
psql (14.13 (Ubuntu 14.13-0ubuntu0.22.04.1), server 15.10 (Debian 15.10-1.pgdg120+1))
WARNING: psql major version 14, server major version 15.
Some psql features might not work.
Type "help" for help.

postgres=# CREATE TABLE test_table (
  id SERIAL PRIMARY KEY,
  name VARCHAR(50),
  lastname VARCHAR(50)
);
CREATE TABLE
postgres=# INSERT INTO test_table (name, lastname) VALUES ('Sergey', 'Senichev');
INSERT 0 1
postgres=# INSERT INTO test_table (name, lastname) VALUES ('Alla', 'Diachkova');
INSERT 0 1
postgres=#
```

Проверка репликации, видим те же записи при подключении к pg-slave

```
root@copium-box ~/lab3
psql -h 127.0.0.1 -p 5434 -U postgres -d postgres
Password for user postgres:
psql (14.13 (Ubuntu 14.13-0ubuntu0.22.04.1), server 15.10 (Debian 15.10-1.pgdg120+1))
WARNING: psql major version 14, server major version 15.
Some psql features might not work.
Type "help" for help.

postgres=# SELECT * FROM test_table;
 id | name  | lastname
----+-----+-----
  1 | Sergey | Senichev
  2 | Alla  | Diachkova
(2 rows)

postgres=#
```

При попытке редактирования данные на pg-slave, получаем ошибку: действительно, тк pg-slave имеет read-only доступ

```
postgres=# INSERT INTO test_table (name, lastname) VALUES ('Nikita', 'Samokhin');
ERROR:  cannot execute INSERT in a read-only transaction
postgres=#
```

### Часть 3. Делаем высокую доступность

Добавляем в Dockerfile описание для контейнера HAProxy, создаем конфиг-файл с настройками для HAProxy. Останавливаем работу контейнера, перезапускаем.

Активные контейнеры:

```
root@copium-box ~/lab3 (0.068s)
docker ps
CONTAINER ID   IMAGE                                COMMAND
NAMES
c56d697f74a6   haproxy:3.0                        "docker-entrypoint.s..."
:::7000->7000/tcp   postgres_entrypoint
3598ebd7db97    confluentinc/cp-zookeeper:7.7.1   "/etc/confluent/dock..."
zoo
46d99c6a9fd3   localhost/postgres:patroni         "docker-entrypoint.s..."
pg-slave
19d5f09de45f   localhost/postgres:patroni         "docker-entrypoint.s..."
pg-master
```

Смотрим логи:

```
root@copium-box ~/lab3 (0.059s)
docker logs c56d697f74a6
[NOTICE] (1) : New worker (9) forked
[NOTICE] (1) : Loading success.
[WARNING] (9) : Server postgres/postgresql_pg_master_5432 is DOWN, reason: Layer4 connection pr
servers left. 0 sessions active, 0 queued, 0 remaining in queue.
[WARNING] (9) : Server postgres/postgresql_pg_slave_5432 is DOWN, reason: Layer4 connection pro
ervers left. 0 sessions active, 0 queued, 0 remaining in queue.
[ALERT] (9) : proxy 'postgres' has no server available!
[WARNING] (9) : Server postgres/postgresql_pg_master_5432 is UP, reason: Layer7 check passed, c
s queued, 0 total in queue.
```

Проверяем работу HAProxy

Видим IP-адрес контейнера HAProxy (postgres\_entrypoint), подключаемся:

```
root@copium-box ~/lab3 (0.058s)
docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' postgres_entrypoint
172.19.0.5

root@copium-box ~/lab3
psql -h 172.19.0.5 -p 5432 -U postgres -d postgres
Password for user postgres:
psql (14.13 (Ubuntu 14.13-0ubuntu0.22.04.1), server 15.10 (Debian 15.10-1.pgdg120+1))
WARNING: psql major version 14, server major version 15.
        Some psql features might not work.
Type "help" for help.
```

Проверка перенаправления на мастер-ноду:

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

Видим false - значит, подключены к мастер-ноде.

### Часть 3. Задание

Любым способом выключаем доступ до ноды, которая сейчас является мастером (например, через `docker stop`). Некоторое время ждем, после этого анализируем логи и так же пытаемся считать/записать что-то в БД через `entroyoint` подключение. Затем необходимо расписать, получилось или нет, а так же объяснить, что в итоге произошло после принудительного выключения мастера (со скриншотами).

Отключаем доступ до мастер ноды. Видим в логах, что слейв получил промоушн и стал мастером:

```
2024-11-24 11:52:10.835 WARNING: Request failed to postgresql0: GET http://pg-master:8008/patroni (HTTPConnectionPool(host='pg-master', port=8008): Max retries exceeded with url: /patroni (Caused by NameResolutionError("urllib3.connection.HTTPConnection object at 0x7f2908df77d0>: Failed to resolve 'pg-master' ([Errno -2] Name or service not known)"))
2024-11-24 11:52:10.850 INFO: promoted self to leader by acquiring session lock
2024-11-24 11:52:10.852 UTC [28] LOG: received promote request
```

2024-11-24 11:52:10.953 UTC [24] LOG: database system is ready to accept connections

Видим, что в HAProxy произошло переключение на слейв:

```
[WARNING] (9) : Server postgres/postgresql_pg_master_5432 is DOWN, reason: Layer4 timeout, check duration: 3002ms. 0 active and 0 backup servers left. 0 sessions active, 0 queued, 0 remaining in queue.
```

```
[ALERT] (9) : proxy 'postgres' has no server available!
```

```
[WARNING] (9) : Server postgres/postgresql_pg_slave_5432 is UP, reason: Layer7 check passed, code: 200, check duration: 2ms. 1 active and 0 backup servers online. 0 sessions queued, 0 total in queue.
```

Подключаемся к HAProxy и смотрим работоспособность. Результат f, подключены к новому мастеру.

```
root@copium-box ~/lab3
psql -h 172.19.0.5 -p 5432 -U postgres -d postgres
Password for user postgres:
psql (14.13 (Ubuntu 14.13-0ubuntu0.22.04.1), server 15.10 (Debian 15.10-1.pgdg120+1))
WARNING: psql major version 14, server major version 15.
        Some psql features might not work.
Type "help" for help.

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

Тестовая запись в таблицу проходит успешно:

```
postgres=# INSERT INTO test_table (name, lastname) VALUES ('Nikita', 'Samokhin');
INSERT 0 1
postgres=# SELECT * FROM test_table;
 id | name  | lastname
----+-----+-----
  1 | Sergey | Senichev
  2 | Alla  | Diachkova
 34 | Nikita | Samokhin
(3 rows)
```

#### Вывод по ч.4:

В итоге после отключения мастер-ноды произошло автоматическое присваивание мастер роли бывшей слейв-ноде, теперь HAProxy направляет запросы на новый мастер, бд остается доступной.

## Часть 5. 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?*

Разница между директивами expose и ports в Docker Compose заключается в их назначении и области действия: когда expose открывает порт для внутренней сети Docker (значит, другие контейнеры в одной сети Docker Compose могут обращаться к этому порту, но он недоступен внешне), ports пробрасывает сопоставляет порту из контейнера порт хост-машины (порт доступен для внешних запросов).

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

При обычном перезапуске без изменений:

- Docker Compose проверяет существование и состояние образов и контейнеров, описанных в docker-compose.yml. Если образы и контейнеры не изменены, они будут перезапущены без пересборки образов или пересоздания.

При перезапуске с отредактированными postgresX.yml:

- Docker Compose не отслеживает изменения в файлах, не связанных с процессом сборки образа (postgres0.yml и postgres1.yml). Проект не пересобирается, изменения в таких файлах вступают в силу без пересборки, так как Docker напрямую использует обновлённые файлы.

Перезапуск при измененном содержимом Dockerfile:

- Docker compose проверяет изменения в Dockerfile, образ будет пересобран. Для применения изменений потребуется пересборка образа, например, с помощью команды docker-compose up --build, и изменения в Dockerfile появятся.