

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

Факультет инфокоммуникационных технологий

Отчет по дисциплине: **«Администрирование компьютерных сетей»**
Практическая работа №3 **«HA Postgres Cluster »**

Команда: Web_holders

Выполнили студенты:

Буякова Е.С. К3344

Кибенко Д.В. К3344

Лазебный В.В. К3344

Шемелева Е.С. К3342

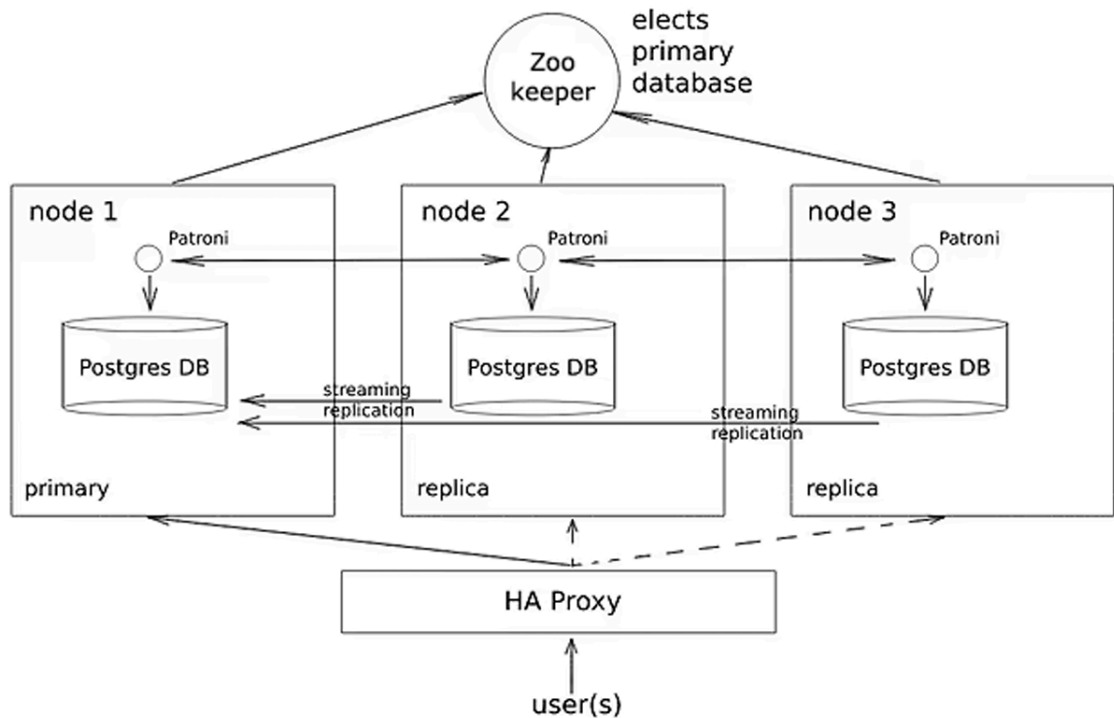
Лобова П.М. К3344

Проверил: Самохин Н.Ю.

Санкт-Петербург
2024

Введение

Задача лабораторной работы: Развернуть и настроить высокодоступный кластер Postgres:



Работа выполнялась локально, Docker Desktop был установлен до выполнения работы. Однако для работы устанавливался PostgreSQL, чтобы работать через консоль. В ходе установки была допущена ошибка с портами. Было принято решение изменить порты для контейнеров, на выполнение задачи никак не повлияло.

Выполнение лабораторной работы

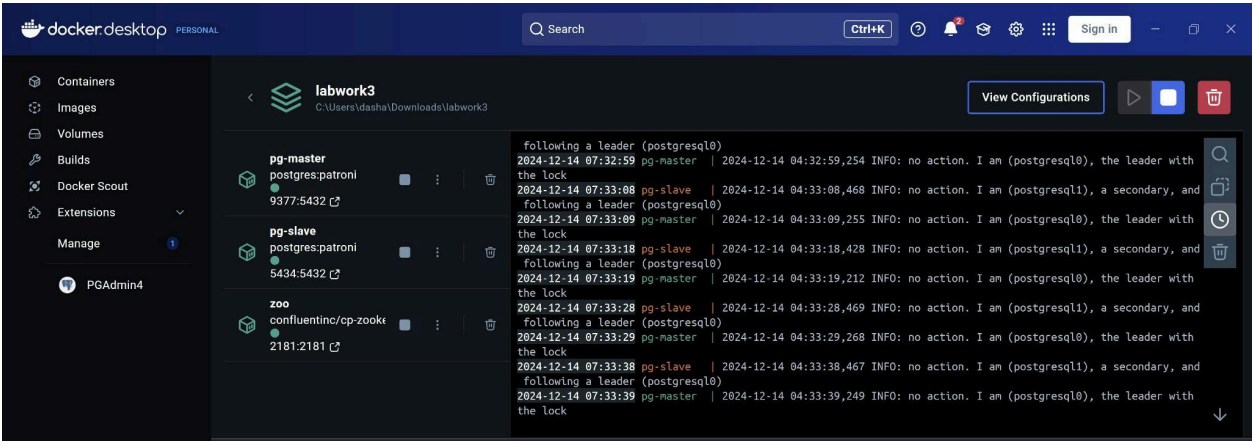
Часть 1. Поднимаем Postgres

Подготавливаем Dockerfile для нашего постгреса. Файлы при создании не подвергались правкам.

После мы деплоим, результат которого покажет, что pg-master будет нодой постгрес, ставшей лидером.

```
Terminal
PS C:\Users\dasha\Downloads\labwork3> docker-compose up -d
time="2024-12-14T07:34:57+03:00" level=warning msg="C:\\Users\\dasha\\Downloads\\labwork3\\docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 4/4
  ✓ Container pg-master           Running      0.0s
is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 4/4
  ✓ Container pg-master           Running      0.0
is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 4/4
  ✓ Container pg-master           Running      0.
is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 4/4
  ✓ Container pg-master           Running      0
is obsolete, it will be ignored, please remove it to avoid potential confusion"
is obsolete, it will be ignored, please remove it to avoid potential confusion"
is obsolete, it will be ignored, please remove it to avoid potential confusion"
is obsolete, it will be ignored, please remove it to avoid potential confusion"
is obsolete, it will be ignored, please remove it to avoid potential confusion"
[+] Running 4/4
  ✓ Container pg-master           Running
```

```
Terminal
s obsolete, it will be ignored, please remove it to avoid po
tential confusion"
[+] Running 4/4
  ✓ Container zoo                 R...      0.0s
  ✓ Container pg-master           Running    0.0s
  ✓ Container pg-master           Running    0.0
s
  ✓ Container pg-master           Running    0.0s
  ✓ Container pg-slave            Running    0.0s
```



```
PS C:\Users\dasha\Downloads\labwork3> docker container ls
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
08ada469d36c   haproxy:3.0                         "docker-entrypoint.s..." 2 hours ago   Up 2 hours   0.0.0.0:7000->7000/tcp, 0.0.0.0:9
327->5432/tcp   postgres_entrypoint
3c89f3a7b9f1   confluentinc/cp-zookeeper:7.7.1    "/etc/confluent/dock..." 11 hours ago  Up 11 hours   2888/tcp, 0.0.0.0:2181->2181/tcp,
3888/tcp
bc685b8f121c   localhost/postgres:patroni         "docker-entrypoint.s..." 11 hours ago  Up 11 hours   8008/tcp, 0.0.0.0:5434->5432/tcp
pg-slave
7ea276a01e63   localhost/postgres:patroni         "docker-entrypoint.s..." 11 hours ago  Up 11 hours   8008/tcp, 0.0.0.0:9377->5432/tcp
pg-master
```

Часть 2. Проверяем репликацию

Именно на этом этапе возникла ошибка с портами. Из-за того, что порт 5432 и 5433 были заняты, невозможно было через постгрес клиента подключиться к ноде pg-master.

```
C:\Program Files\PostgreSQL\15\bin>psql -h localhost -p 5433 -U postgres
Пароль пользователя postgres:
psql: ошибка: подключиться к серверу "localhost" (::1), порту 5433 не удалось: ВАЖНО: пользователь "postgres" не прошёл проверку подлинности (по паролю)
```

Из двух подключений выбираем нужный контейнер/базу данных и создаем таблицу с предложенной из инструкции структурой, потом записываем в нее данные из примера.

```
C:\Program Files\PostgreSQL\15\bin>psql -h 127.0.0.1 -p 9377 -U postgres -d postgres
Пароль пользователя postgres:
psql (15.4, сервер 15.10 (Debian 15.10-1.pgdg120+1))
Введите "help", чтобы получить справку.

postgres=# CREATE TABLE my_first_replication ( id INT, my_data VARCHAR, my_comment VARCHAR);
CREATE TABLE
postgres=# INSERT INTO my_first_replication (id, my_data, my_comment)VALUES (1, 'my important data', 'is this line replicated?');
INSERT 0 1
postgres=# SELECT * FROM my_first_replication;
 id |      my_data      |      my_comment
-----+-----+-----
  1 | my important data | is this line replicated?
(1 строка)

postgres=#
```

```
postgres=# psql -h 127.0.0.1 -p 5434 -U postgres -d postgres
postgres=# SELECT pg_is_in_recovery();
ERROR:  syntax error at or near "psql"
СТРОКА 1: psql -h 127.0.0.1 -p 5434 -U postgres -d postgres
          ^
postgres=# \q

C:\Program Files\PostgreSQL\15\bin>psql -h 127.0.0.1 -p 9377 -U postgres -d postgres
Пароль пользователя postgres:
psql (15.4, сервер 15.10 (Debian 15.10-1.pgdg120+1))
Введите "help", чтобы получить справку.

postgres=# SELECT pg_is_in_recovery();
 pg_is_in_recovery
-----
 f
(1 строка)

postgres=#
```

В подключении pg-slave пробуем провести операцию на редактирование. Получим отказ, ведь эта нода работает в режиме slave/readonly.

```
C:\Program Files\PostgreSQL\15\bin>psql -h 127.0.0.1 -p 5434 -U postgres -d postgres
Пароль пользователя postgres:
psql (15.4, сервер 15.10 (Debian 15.10-1.pgdg120+1))
Введите "help", чтобы получить справку.
```

```
postgres=# SELECT pg_is_in_recovery();
pg_is_in_recovery
-----
t
(1 строка)
```

```
postgres=# CREATE TABLE my_first_replication ( id INT, my_data VARCHAR, my_comment VARCHAR);
ERROR: cannot execute CREATE TABLE in a read-only transaction
postgres=# CREATE TABLE my_first_replication (id INT, my_data VARCHAR, my_comment VARCHAR);
ERROR: cannot execute CREATE TABLE in a read-only transaction
postgres=#
```

Часть 3. Делаем среднего роста/высокую доступность

Мы создаем специальное ПО для балансировки трафика. Используем для этого HAProxy.

Перезапускаем проект в Docker и убеждаемся, что:

- Обе ноды корректно поднялись и распределили между собой роли мастера и слейва.
- Зукипер подцепился к кластеру без ошибок.
- Хапрокси поднялась без ошибок.

The screenshot shows the 'labwork3' Docker management interface. On the left, a list of containers is displayed:

- pg-master**: postgres:patroni, 9377:5432
- pg-slave**: postgres:patroni, 5434:5432
- zoo**: confluentinc/cp-zoo, 2181:2181
- postgres_entrypoint**: haproxy:3.0, 9327:5432

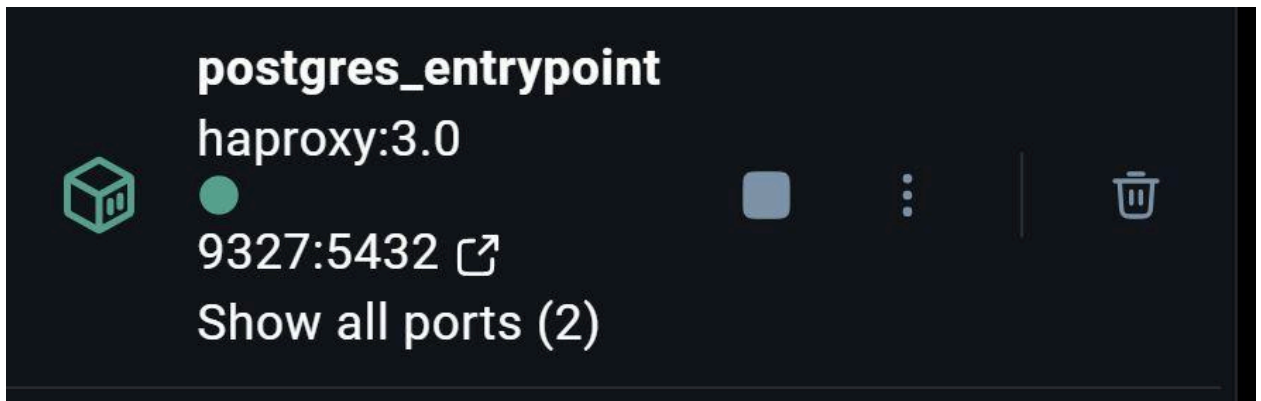
On the right, the logs for the 'pg-master' container are visible, showing PostgreSQL startup messages and replication status. At the bottom, a 'Terminal' window displays the status of the containers:

Container	Status	Time
Container zoo	Running	0.0s
Container pg-master	Running	0.0s
Container pg-master	Running	0.0
Container pg-master	Running	0.0s
Container pg-slave	Running	0.0s
Container postgres_entrypoint	Started	1.4s

System resources at the bottom: RAM 4.77 GB, CPU 0.00%, Disk 237.34 GB avail. of 269.43 GB. The interface is running on v4.36.0.

Подключаемся к появившейся базе данных, в качестве адреса используется имя контейнера с HAProxy. Проверяем, что подключение успешное.

В данном случае тоже понадобилось поменять порт.



```
PS C:\Users\dasha\Downloads\labwork3> docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
08ada469d36c	haproxy:3.0	"docker-entrypoint.s..."	2 hours ago	Up 2 hours	0.0.0.0:7000->7000/tcp, 0.0.0.0:9
327->5432/tcp	postgres_entrypoint				
3c89f3a7b9f1	confluentinc/cp-zookeeper:7.7.1	"/etc/confluent/dock..."	11 hours ago	Up 11 hours	2888/tcp, 0.0.0.0:2181->2181/tcp,
3888/tcp	zoo				
bc685b8f121c	localhost/postgres:patroni	"docker-entrypoint.s..."	11 hours ago	Up 11 hours	8008/tcp, 0.0.0.0:5434->5432/tcp
	pg-slave				
7ea276a01e63	localhost/postgres:patroni	"docker-entrypoint.s..."	11 hours ago	Up 11 hours	8008/tcp, 0.0.0.0:9377->5432/tcp
	pg-master				

```
C:\Program Files\PostgreSQL\15\bin>psql -h 127.0.0.1 -p 5432 -U postgres -d postgres
Пароль пользователя postgres:
psql: ошибка: подключиться к серверу "127.0.0.1", порту 5432 не удалось: ВАЖНО: пользователь "pos
tgres" не прошёл проверку подлинности (по паролю)
```

```
C:\Program Files\PostgreSQL\15\bin>psql -h localhost -p 9327 -U postgres -d postgres
Пароль пользователя postgres:
psql (15.4, сервер 15.10 (Debian 15.10-1.pgdg120+1))
Введите "help", чтобы получить справку.

postgres=#
```

```
C:\Program Files\PostgreSQL\15\bin>psql -h localhost -p 9327 -U postgres -d postgres
Пароль пользователя postgres:
psql (15.4, сервер 15.10 (Debian 15.10-1.pgdg120+1))
Введите "help", чтобы получить справку.

postgres=# SELECT pg_is_in_recovery();
 pg_is_in_recovery
-----
 f
(1 строка)

postgres=#
```

Ответы на вопросы

1. 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?

Ответ: Директива expose делает порты доступными только для контейнеров одной сети Docker, т.е. порты не будут открыты на хосте. Директива ports, в свою очередь, делает порты доступными на хосте, таким образом внешние приложения могут получить доступ к контейнеру по записанному порту.

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

Ответ: Если образ был уже сбилден ранее и совершается перезапуск без каких-либо изменений - образ не будет собираться заново, только запускаться.

Редактирование файла postgresX.yml повлияет только на конфигурацию контейнера при запуске. Изменения в этом файле не затрагивают процесс сборки образа, поэтому он не будет собираться заново.

Редактирование Dockerfile напрямую влияет на конфигурацию образа, поэтому образ будет собран заново в соответствии с изменениями.