



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий

Кафедра математического обеспечения и стандартизации информационных
технологий

ОТЧЁТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №2
по дисциплине «Распределенные системы управления базами данных»

Тема: «Знакомство с Apache Cassandra»

Выполнил студент группы

ИКБО-07-22

Буренин А. А.

Принял преподаватель

Красников С. А.

Работа выполнена

«__» _____ 20__

(Подпись студента)

Зачтено

«__» _____ 20__

(Подпись преподавателя)

Москва 2025

1. Ход работы

1.1. Создание кластера Cassandra

Для создания кластера Cassandra можно воспользоваться docker compose. Нам необходимо создать 3 ноды, 2 в дата-центре 1 и одну в дата-центре 2. Использование docker compose в данном случае поможет избежать использование ip-адресов, а так же упростит поиск ошибок.

Листинг 1 — Файл compose.yaml

```
x-cassandra: &cassandra-commons
  image: 'cassandra:latest'
  expose:
    - 7000
    - 9042
  networks:
    - cassandra

x-common-environments: &cassandra-common-environments
  CASSANDRA_SEEDS: cassandra-dc-1-node-1
  CASSANDRA_CLUSTER_NAME: burenin
  CASSANDRA_ENDPOINT_SNITCH: GossipingPropertyFileSnitch
  CASSANDRA_DC: dc-1
  MAX_HEAP_SIZE: 500M
  HEAP_NEWSIZE: 500M

services:

  cassandra-dc-1-node-1:
    <<: *cassandra-commons
    environment:
      <<: *cassandra-common-environments
      CASSANDRA_SEEDS: ""
    container_name: cassandra-dc-1-node-1
    hostname: cassandra-dc-1-node-1

  cassandra-dc-1-node-2:
    <<: *cassandra-commons
    container_name: cassandra-dc-1-node-2
    hostname: cassandra-dc-1-node-2
    environment:
      <<: *cassandra-common-environments
    depends_on:
      - cassandra-dc-1-node-1

  cassandra-dc-2-node-1:
    <<: *cassandra-commons
    container_name: cassandra-dc-2-node-1
    hostname: cassandra-dc-2-node-1
    environment:
      <<: *cassandra-common-environments
      CASSANDRA_SEEDS: cassandra-dc-1-node-1
      CASSANDRA_DC: dc-2
```

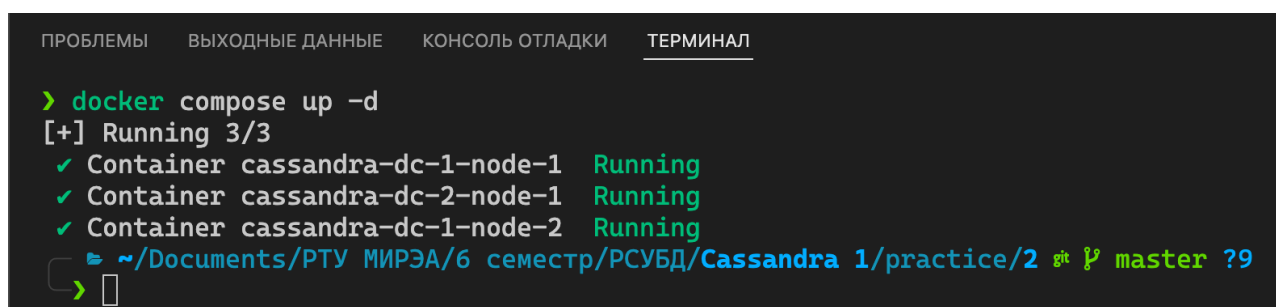
Продолжение листинга 1

```
depends_on:
  - cassandra-dc-1-node-1

networks:
  cassandra:
    driver: bridge
```

На листинге 1 представлен compose файл, в котором создаются 3 контейнера Cassandra.

Для создания и запуска кластера необходимо использовать команду `docker compose up -d`. Ниже на рисунке можно увидеть результат работы этой команды



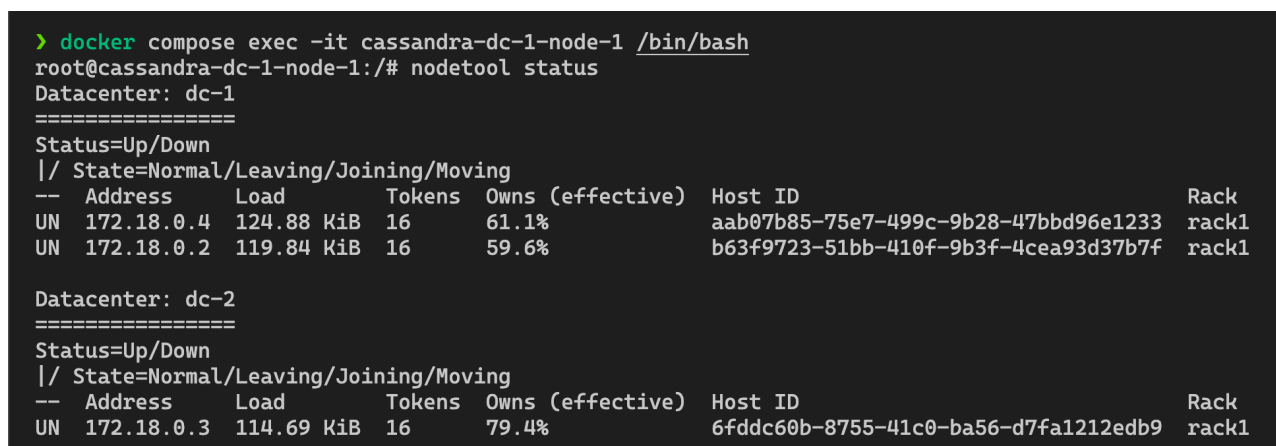
```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ

> docker compose up -d
[+] Running 3/3
✓ Container cassandra-dc-1-node-1  Running
✓ Container cassandra-dc-2-node-1  Running
✓ Container cassandra-dc-1-node-2  Running
~/Documents/ПТУ МИРЭА/6 семестр/PCУБД/Cassandra 1/practice/2 git master ?9
> █
```

Рисунок 1 — Результат запуска кластера

1.2. Проверка кластера

Для проверки работоспособности кластера, подключимся к одной из его нод и выполним команду `nodetool status`. На рисунке 2 можно увидеть, что созданы три ноды: две в dc-1 и одна в dc-2



```
> docker compose exec -it cassandra-dc-1-node-1 /bin/bash
root@cassandra-dc-1-node-1:/# nodetool status
Datacenter: dc-1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens    Owns (effective)  Host ID                               Rack
UN 172.18.0.4    124.88 KiB    16        61.1%             aab07b85-75e7-499c-9b28-47bbd96e1233 rack1
UN 172.18.0.2    119.84 KiB    16        59.6%             b63f9723-51bb-410f-9b3f-4cea93d37b7f rack1

Datacenter: dc-2
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens    Owns (effective)  Host ID                               Rack
UN 172.18.0.3    114.69 KiB    16        79.4%             6fddc60b-8755-41c0-ba56-d7fa1212edb9 rack1
```

Рисунок 2 — Состояние нод в кластере

1.3. Создание пространства ключей

Создадим пространство ключей, которое будет иметь по одной реплике в dc-1 и в dc-2. На рисунке 3.

```
root@cassandra-dc-1-node-1:/# cqlsh
Connected to burenin at 127.0.0.1:9042
[cqlsh 6.2.0 | Cassandra 5.0.3 | CQL spec 3.4.7 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE mykeyspace WITH replication = {'class': 'NetworkTopologyStrategy', 'dc-1': 1, 'dc-2': 1}
...
cqlsh> |
```

Рисунок 3 — Создание пространства ключей «keyspace»

1.4. Создание таблицы users

Создадим таблицу «users» с полями id, first_name и last_name и заполним её тестовыми данными.

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ

cqlsh:mykeyspace> USE mykeyspace ;
cqlsh:mykeyspace> CREATE TABLE users(id int primary key, first_name VARCHAR, last_name VARCHAR);
cqlsh:mykeyspace> INSERT INTO users (id, first_name, last_name) VALUES (1, 'John', 'Doe')
...
cqlsh:mykeyspace> INSERT INTO users (id, first_name, last_name) VALUES (2, 'Jane', 'Doe');
cqlsh:mykeyspace> SELECT * FROM users;

 id | first_name | last_name
----+-----+-----
  1 |      John  |      Doe
  2 |      Jane  |      Doe

(2 rows)
cqlsh:mykeyspace> |
```

Рисунок 4 — Создание таблицы «users» и заполнение ее данными

1.5. Выборка данных из другой ноды

Проверим, что данные успешно реплицировались на другую ноду, для этого подключимся cassandra-dc-2-node-1 и попробуем запросить таблицу «users».

```
> docker compose exec -it cassandra-dc-2-node-1 /bin/bash
root@cassandra-dc-2-node-1:/# cqlsh
Connected to burenin at 127.0.0.1:9042
[cqlsh 6.2.0 | Cassandra 5.0.3 | CQL spec 3.4.7 | Native protocol v5]
Use HELP for help.
cqlsh> use mykeyspace ;
cqlsh:mykeyspace> select * from users;
```

id	first_name	last_name
1	John	Doe
2	Jane	Doe

```
(2 rows)
cqlsh:mykeyspace> █
```

Рисунок 5 — Выборка данных из таблицы «users» на другой ноде

2. Контрольные вопросы

Как Вы думаете, к какому типу РБД относится БД, реализованная в практической работе? Почему?

БД, реализованная в практической работе безусловно относится к типу гомогенных автономны, так как схема базы данных совпадает на всех нодах и все ноды «знают» о существовании всех остальных. В последнем можно убедиться, рассмотрев рисунок 2.

Какой командой устанавливается образ Cassandra с помощью Docker

Образ докер устанавливается командой `docker compose pull cassandra:latest`, однако эта команда не является обязательной. Если образа нет на машине во время выполнения команды `docker run`, он будет скачен автоматически.

3. Вывод

При выполнении практической работы был создан и настроен Cassndra кластер из трёх нод в двух дата-центрах. После этого была проверена работа репликации данных.