

Task 6 - Реплікація у Cassandra

Сконфігурувати кластер з 3-х нод

```
root@hypnos:~# docker run --name cas1 -e CASSANDRA_CLUSTER_NAME=cluster -e CASSANDRA_ENDPOINT_SNITCH=GossipingPropertyFileSnitch -d cassandra
id31baa54aab04240defad0f780331f9ff43e9821b3141fc4a5020a28f562387
root@hypnos:~# docker run --name cas2 -e CASSANDRA_SEEDS="$(docker inspect --format='{{ .NetworkSettings.IPAddress }}' cas1)" -e CASSANDRA_CLUSTER_NAME=cluster -e CASSANDRA_ENDPOINT_SNITCH=GossipingPropertyFileSnitch -d cassandra
e79530fb58b29d1cf87e6f5bffa165f6f8a661025fff7077dc67b1493709f3d2
root@hypnos:~# docker run --name cas3 -e CASSANDRA_SEEDS="$(docker inspect --format='{{ .NetworkSettings.IPAddress }}' cas1)" -e CASSANDRA_CLUSTER_NAME=cluster -e CASSANDRA_ENDPOINT_SNITCH=GossipingPropertyFileSnitch -d cassandra
c7ea6060bc0da2f4d792bc36799732fcee036a93c846bc09da2e0aa1fc4aa9c
root@hypnos:~#
```

```
~$ docker network connect net cas1
~$ docker network connect net cas2
~$ docker network connect net cas3
```

Перевірити правильність конфігурації за допомогою
nodetool status

```
root@hypnos:~# docker exec -it cas1 nodetool status
Datacenter: dc1
=====
Status=Up/Down
// State=Normal/Leaving/Joining/Moving
-- Address      Load       Tokens     Owns (effective)  Host ID                               Rack
UN 172.17.0.4    144.6 KiB   16         59.3%             c7a32f03-93ba-4c7f-a93c-b28fc17e0baf rack1
UN 172.17.0.3    74.03 KiB   16         76.0%             6ec1f95e-d588-4dce-be3d-97045606a07b rack1
UN 172.17.0.2    74.1 KiB    16         64.7%             0057c126-7ef0-462b-a771-b8e1465d410d rack1
```

Використовуючи *cqlsh*, створити три *Keyspace* з replication factor 1, 2, 3

```
root@hypnos:~# docker exec -it cas1 cqlsh
Connected to cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.1 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE ks1 WITH replication = { 'class' : 'NetworkTopologyStrategy', 'replication_factor':1};
SyntaxException: line 1:7 no viable alternative at input 'CREATE' ([CREATE] CREATE...)
cqlsh> CREATE KEYSPACE ks1 WITH replication = { 'class' : 'NetworkTopologyStrategy', 'replication_factor':1};
cqlsh> CREATE KEYSPACE ks2 WITH replication = { 'class' : 'NetworkTopologyStrategy', 'replication_factor':2};
cqlsh> CREATE KEYSPACE ks3 WITH replication = { 'class' : 'NetworkTopologyStrategy', 'replication_factor':3};
cqlsh>
```

1. В кожному з кейспейсів створити таблиці

```
cqlsh> CREATE TABLE ks1.table1 (id int primary key, name text);
cqlsh> CREATE TABLE ks2.table1 (id int primary key, name text);
cqlsh> CREATE TABLE ks3.table1 (id int primary key, name text);
```

Спробуйте писати і читати на / та з різних нод.

Node 1:

```

root@hypnos:~# docker exec -it cas1 cqlsh
Connected to Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.1 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> INSERT INTO ks1.table1(id, name) VALUES (1, 'Text 1');
cqlsh> INSERT INTO ks1.table1(id, name) VALUES (2, 'Text 2');
cqlsh> INSERT INTO ks1.table1(id, name) VALUES (3, 'Text 3');
cqlsh> INSERT INTO ks2.table1(id, name) VALUES (4, 'Text 4');
cqlsh> INSERT INTO ks3.table1(id, name) VALUES (5, 'Text 5');
cqlsh> SELECT * FROM ks1.table1;

 id | name
-----+-----
  1 | Text 1
  2 | Text 2
  3 | Text 3

(3 rows)
cqlsh>

```

Node 2:

```

root@hypnos:~# docker exec -it cas2 cqlsh
Connected to Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.1 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> SELECT * FROM ks1.table1;

 id | name
-----+-----
  1 | Text 1
  2 | Text 2
  3 | Text 3

(3 rows)
cqlsh> SELECT * FROM ks2.table1;

 id | name
-----+-----
  4 | Text 4

(1 rows)
cqlsh> SELECT * FROM ks3.table1;

 id | name
-----+-----
  5 | Text 5

(1 rows)
cqlsh> INSERT INTO ks1.table1(id, name) VALUES (6, 'Text 6');
cqlsh> INSERT INTO ks3.table1(id, name) VALUES (7, 'Text 7');
cqlsh> SELECT * FROM ks1.table1;

 id | name
-----+-----
  1 | Text 1
  2 | Text 2
  6 | Text 6
  3 | Text 3

(4 rows)
cqlsh>

```

Node 3:

```

root@hypnos:~# docker exec -it cas3 cqlsh
Connected to Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.1 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> SELECT * FROM ks1.table1;

 id | name
-----+-----
  1 | Text 1
  2 | Text 2
  6 | Text 6
  3 | Text 3

(4 rows)
cqlsh> SELECT * FROM ks2.table1;

 id | name
-----+-----
  4 | Text 4

(1 rows)
cqlsh> SELECT * FROM ks3.table1;

 id | name
-----+-----
  5 | Text 5
  7 | Text 7

(2 rows)
cqlsh>

```

Вставте дані в створені таблиці і подивіться на їх розподіл по вузлах кластера (для кожного з кейспесов - *nodetool status*)

```

root@hypnos:~# docker exec -it cas1 nodetool status ks1;
Datacenter: dc1
=====
Status=Up/Down
-- State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens    Owns (effective)  Host ID                               Rack
UN 172.17.0.4    143.26 KiB    16        31.6%             c7a32f03-93ba-4c7f-a93c-b28fc17e0baf rack1
UN 172.17.0.3    79.5 KiB      16        35.7%             6ec1f95e-d588-4dce-be3d-97045606a07b rack1
UN 172.17.0.2    79.58 KiB     16        32.7%             0057c126-7ef0-462b-a771-b8e1465d410d rack1

root@hypnos:~# docker exec -it cas1 nodetool status ks2
Datacenter: dc1
=====
Status=Up/Down
-- State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens    Owns (effective)  Host ID                               Rack
UN 172.17.0.4    143.26 KiB    16        59.3%             c7a32f03-93ba-4c7f-a93c-b28fc17e0baf rack1
UN 172.17.0.3    79.5 KiB      16        76.0%             6ec1f95e-d588-4dce-be3d-97045606a07b rack1
UN 172.17.0.2    79.58 KiB     16        64.7%             0057c126-7ef0-462b-a771-b8e1465d410d rack1

root@hypnos:~# docker exec -it cas1 nodetool status ks3
Datacenter: dc1
=====
Status=Up/Down
-- State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens    Owns (effective)  Host ID                               Rack
UN 172.17.0.4    143.26 KiB    16        100.0%            c7a32f03-93ba-4c7f-a93c-b28fc17e0baf rack1
UN 172.17.0.3    79.5 KiB      16        100.0%            6ec1f95e-d588-4dce-be3d-97045606a07b rack1
UN 172.17.0.2    79.58 KiB     16        100.0%            0057c126-7ef0-462b-a771-b8e1465d410d rack1

```

Для якогось запису з кожного з кейспейсу виведіть ноди на яких зберігаються дані

```

root@hypnos:~# docker exec -it cas1 nodetool getendpoints ks1 table1 1
172.17.0.3
root@hypnos:~# docker exec -it cas1 nodetool getendpoints ks1 table1 2
172.17.0.4
root@hypnos:~# docker exec -it cas1 nodetool getendpoints ks1 table1 3
172.17.0.2
root@hypnos:~# docker exec -it cas1 nodetool getendpoints ks3 table1 5
172.17.0.4
172.17.0.2
172.17.0.3
root@hypnos:~# docker exec -it cas1 nodetool getendpoints ks3 table1 7
172.17.0.3
172.17.0.2
172.17.0.4
root@hypnos:~# docker exec -it cas1 nodetool getendpoints ks2 table1 4
172.17.0.3
172.17.0.4
root@hypnos:~#

```

Відключіть одну з нод. Для кожного з кейспейсів визначить з якими рівнями *consistency* можемо читати та писати, і які з них забезпечують *strong consistency*

```

root@hypnos:~# docker stop cas3
cas3
root@hypnos:~# docker exec -ti cas1 nodetool status ks1
Datacenter: dc1
=====
Status=Up/Down
// State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens     Owns (effective)  Host ID                               Rack
DN  172.17.0.4    143.26 KiB    16         31.6%             c7a32f03-93ba-4c7f-a93c-b28fc17e0baf rack1
UN  172.17.0.3    79.5 KiB     16         35.7%             6ec1f95e-d588-4dce-be3d-97045606a07b rack1
UN  172.17.0.2    79.58 KiB    16         32.7%             0057c126-7ef0-462b-a771-b8e1465d410d rack1

cqlsh> USE ks1
... ;
cqlsh:ks1> CONSISTENCY
Current consistency level is ONE.
cqlsh:ks1> USE ks2 ;
cqlsh:ks2> CONSISTENCY
Current consistency level is ONE.
cqlsh:ks2> USE ks3 ;
cqlsh:ks3> CONSISTENCY
Current consistency level is ONE.

```

$R + W > N$

ks1: $N = 1, R = 1, W = 1$ - strong consistency

ks2: $N = 2, R = 1, W = 1$ - weak consistency

ks3: $N = 2, R = 1, W = 1$ - weak consistency

Зробіть так щоб три ноди працювали, але не бачили одна одну по мережі (відключити зв'язок між ними)

```

root@hypnos:~# docker network disconnect cass-net cas1
root@hypnos:~# docker network disconnect cass-net cas2
root@hypnos:~# docker network disconnect cass-net cas3

```

кейспейсу з *replication factor* 3 задайте рівень consistency рівним 1. Виконайте запис одного й того самого значення, з однаковим primary key, але різними іншими значенням на кожному з нод (тобто створіть конфлікт)

Node 1: (id 8, text 8)


```

root@hypnos:~# docker exec -it cas1 cqlsh
Connected to Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.1 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> SELECT * FROM ks1.table1;

id | name
-----+-----
3 | Text 3

(1 rows)
cqlsh> SELECT * FROM ks2.table1;

id | name
-----+-----

(0 rows)
cqlsh> SELECT * FROM ks3.table1;

id | name
-----+-----
5 | Text 5
7 | Text 7

(2 rows)
cqlsh> INSERT INTO ks3.table1 (id, name) VALUES (8, 'Text 8');
cqlsh> SELECT * FROM ks3.table1;

id | name
-----+-----
5 | Text 5
8 | Text 8
7 | Text 7

(3 rows)

```

Node 2: (id: 8, text 9)

```

root@hypnos:~# docker exec -it cas2 cqlsh
Connected to Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.1 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> INSERT INTO ks3.table1 (id, name) VALUES (8, 'Text 9');
cqlsh> SELECT * FROM ks3.table1;

id | name
-----+-----
5 | Text 5
8 | Text 9
7 | Text 7
9 | Text 9

(4 rows)
cqlsh> DELETE FROM ks3.table1 WHERE id = 9;
cqlsh> SELECT * FROM ks3.table1;

id | name
-----+-----
5 | Text 5
8 | Text 9
7 | Text 7

(3 rows)

```

Node 3:

```

root@hypnos:~# docker exec -it cas3 cqlsh
Connected to Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.1 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> SELECT * FROM ks3.table1;

id | name
-----+-----
5 | Text 5
7 | Text 7

(2 rows)
cqlsh> INSERT INTO ks3.table1 (id, name) VALUES (8, 'Text 10');
cqlsh> SELECT * FROM ks3.table1
... ;

id | name
-----+-----
5 | Text 5
8 | Text 10
7 | Text 7

(3 rows)

```

Об'єднайте ноди в кластер і визначте яке значення було прийнято кластером та за яким принципом

```

root@hypnos:~# docker exec -it cas1 cqlsh
Connected to Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.1 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> SELECT * FROM ks3.table1;

 id | name
----+-----
  5 | Text 5
  8 | Text 10
  7 | Text 7
(3 rows)

```

Було обрано найновіше значення з ноди 3.

Перевірте поведінку *lightweight transactions* для попередніх пунктів у розділеному на три частини кластері

```

cqlsh:ks3> CONSISTENCY ONE;
Consistency level set to ONE.
cqlsh:ks3> CONSISTENCY
Current consistency level is ONE.
cqlsh:ks3> INSERT INTO table1 (id, name) VALUES (10, 'Text 10') IF NOT EXISTS;
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 dc1>: Unavailable('Error from server: code=1000 [Unavailable exception] message="Cannot achieve consistency level QUORUM" info={\'consistency\': \'QUORUM\', \'required_replicas\': 2, \'alive_replicas\': 1})'})})
cqlsh:ks3>

```