

Перевірити правильність конфігурації за допомогою *nodetool status*

```
olekthunder@mellon ~ → docker exec -it c1 bash
root@4279c54d5280:/# nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens         Owns (effective)  Host ID
   Rack
UN  10.254.0.4   85.18 KiB     256            64.1%             89714a85-4584-4ada-b6f7-043a
732a4d85 rack1
UN  10.254.0.2   70.7 KiB      256            66.2%             7083c38c-66a5-4049-b260-e1b8
56548357 rack1
UN  10.254.0.3   85.21 KiB     256            69.6%             5c9415fb-3b66-4cf9-8840-7676
e044c9f7 rack1

root@4279c54d5280:/#
```

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

```
cqlsh> create keyspace rep1 with replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> create keyspace rep2 with replication = {'class': 'SimpleStrategy', 'replication_factor': 2};
cqlsh> create keyspace rep3 with replication = {'class': 'SimpleStrategy', 'replication_factor': 3};
root@4279c54d5280:/#
```

В кожному з кейспейсів створити таблиці

```
cqlsh> use rep1;
cqlsh:rep1> create table items (name text primary key, price);
SyntaxException: line 1:48 no viable alternative at input ')' (...name text primary key, price[])...)
cqlsh:rep1> create table items (name text primary key, price int);
cqlsh:rep1> use rep2;
cqlsh:rep2> create table items (name text primary key, price int);
cqlsh:rep2> use rep3;
cqlsh:rep3> create table items (name text primary key, price int);
cqlsh:rep3>
```

Спробуйте писати і читати на / та з різних нод

```
olekthunder@mellon ~ → docker exec -it c3 cqlsh
Connected to cassandra-cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use rep1;insert into items (name, price) values ('Item1',
...
cqlsh> use rep1;
cqlsh:rep1> insert into items (name, price) values ('Item1', 1000);
cqlsh:rep1> select * from items;

name | price
-----+-----
Item1 | 1000

(1 rows)
cqlsh:rep1> 
```

```
olekthunder@mellon ~ → docker exec -it c1 cqlsh
Connected to cassandra-cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use rep1;
cqlsh:rep1> insert into items (name, price) values ('Item2', 1000);
cqlsh:rep1> select * from items;

name | price
-----+-----
Item2 | 1000
Item1 | 1000

(2 rows)
cqlsh:rep1> 
```

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

```
root@4279c54d5280:/# nodetool status rep1
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens      Owns (effective)  Host ID                               Rack
UN  10.254.0.4    71.18 KiB     256         32.4%             89714a85-4584-4ada-b6f7-043a732a4d85 rack1
UN  10.254.0.2    71.16 KiB     256         31.8%             7083c38c-66a5-4049-b260-e1b856548357 rack1
UN  10.254.0.3    71.16 KiB     256         35.8%             5c9415fb-3b66-4cf9-8840-7676e044c9f7 rack1

root@4279c54d5280:/# nodetool status rep2
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens      Owns (effective)  Host ID                               Rack
UN  10.254.0.4    71.18 KiB     256         64.1%             89714a85-4584-4ada-b6f7-043a732a4d85 rack1
UN  10.254.0.2    71.16 KiB     256         66.2%             7083c38c-66a5-4049-b260-e1b856548357 rack1
UN  10.254.0.3    71.16 KiB     256         69.6%             5c9415fb-3b66-4cf9-8840-7676e044c9f7 rack1

root@4279c54d5280:/# nodetool status rep3
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens      Owns (effective)  Host ID                               Rack
UN  10.254.0.4    71.18 KiB     256         100.0%            89714a85-4584-4ada-b6f7-043a732a4d85 rack1
UN  10.254.0.2    71.16 KiB     256         100.0%            7083c38c-66a5-4049-b260-e1b856548357 rack1
UN  10.254.0.3    71.16 KiB     256         100.0%            5c9415fb-3b66-4cf9-8840-7676e044c9f7 rack1

root@4279c54d5280:/#
```

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

```
root@4279c54d5280:/# nodetool getendpoints rep1 items 'Item2';
10.254.0.4
root@4279c54d5280:/# nodetool getendpoints rep2 items 'Item2';
10.254.0.4
10.254.0.3
root@4279c54d5280:/# nodetool getendpoints rep3 items 'Item2';
10.254.0.4
10.254.0.3
10.254.0.2
root@4279c54d5280:/#
```

Для кожного з кейспейсів відключивши одну з нод визначить чи можемо гарантувати strong consistency, для читання та запису, змінюючи рівень consistency

```
olekthunder@mellon ~ → docker stop c2
c2
olekthunder@mellon ~ → docker exec -it c1 cqlsh
Connected to cassandra-cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use rep1;
cqlsh:rep1> CONSISTENCY THREE
Consistency level set to THREE.
cqlsh:rep1> insert into items (name, price) values ('Item100', 1000);
NoHostAvailable:
cqlsh:rep1> select * from items;
NoHostAvailable:
cqlsh:rep1> use rep2;
cqlsh:rep2> insert into items (name, price) values ('Item100', 1000);
NoHostAvailable:
cqlsh:rep2> select * from items;
NoHostAvailable:
cqlsh:rep2> use rep3;
cqlsh:rep3> insert into items (name, price) values ('Item100', 1000);
NoHostAvailable:
cqlsh:rep3> select * from items;
NoHostAvailable:
cqlsh:rep3> CONSISTENCY TWO
Consistency level set to TWO.
cqlsh:rep3> insert into items (name, price) values ('Item100', 1000);
cqlsh:rep3> select * from items;

name      | price
-----+-----
Item4     | 1000
Item3     | 1000
Item2     | 1000
Item7     | 1000
Item9     | 1000
Item8     | 1000
Item100   | 1000
Item0     | 1000
Item5     | 1000
Item6     | 1000

(10 rows)
cqlsh:rep3> use rep2;
cqlsh:rep2> insert into items (name, price) values ('Item100', 1000);
cqlsh:rep2> insert into items (name, price) values ('Item101', 1000);
NoHostAvailable:
cqlsh:rep2> select * from items;
NoHostAvailable:
cqlsh:rep2> □
```

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

```
olekthunder@mellon ~ ❌ docker network disconnect cnet c1
olekthunder@mellon ~ ➔ docker network disconnect cnet c2
olekthunder@mellon ~ ➔ □
```

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

```
olekthunder@mellon ~ ➔ docker exec -it c1 cqlsh
Connected to cassandra-cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use rep3;
cqlsh:rep3> insert into items (name, price) values ('test', 5000);
cqlsh:rep3> CONSISTENCY ONE ;
Consistency level set to ONE.
cqlsh:rep3>
```

```
olekthunder@mellon ~ ➔ docker exec -it c2 cqlsh
Connected to cassandra-cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use rep3;
cqlsh:rep3> CONSISTENCY ONE
Consistency level set to ONE.
cqlsh:rep3> insert into items (name, price) values ('test', 6000);
cqlsh:rep3>
```

```
olekthunder@mellon ~ ➔ docker exec -it c3 cqlsh
Connected to cassandra-cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> insert into items (name, price) values ('test', 7000);
cqlsh> use rep3;
cqlsh:rep3> insert into items (name, price) values ('test', 7000);
cqlsh:rep3> □
```

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

```
olekthunder@mellon ~ → docker network connect cnet c2
olekthunder@mellon ~ → docker network connect cnet c3
olekthunder@mellon ~ → docker exec -it c1 bash
root@4279c54d5280:/# nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
// State=Normal/Leaving/Joining/Moving
-- Address      Load       Tokens     Owns    Host ID                               Rack
UN  10.254.0.4    71.18 KiB  256        ?       89714a85-4584-4ada-b6f7-043a732a4d85 rack1
UN  10.254.0.2    71.16 KiB  256        ?       7083c38c-66a5-4049-b260-e1b856548357 rack1
UN  10.254.0.3    176.9 KiB  256        ?       5c9415fb-3b66-4cf9-8840-7676e044c9f7 rack1

Note: Non-system keyspaces don't have the same replication settings, effective ownership information is meaningless
root@4279c54d5280:/# cqlsh
Connected to cassandra-cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.10 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cqlsh> use rep3;
cqlsh:rep3> select * from items;

 name      | price
-----+-----
 Item4     | 1000
 Item3     | 1000
 Item2     | 1000
 test      | 7000
 Item7     | 1000
 Item9     | 1000
 Item8     | 1000
 Item100   | 1000
 Item0     | 1000
 Item5     | 1000
 Item6     | 1000

(11 rows)
cqlsh:rep3> █
```

Last-write-wins conflict resolution - останній, хто писав - істинний.