



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Проектування високонавантажених систем

Лабораторна робота №5

**Робота з базовими функціями БД типу column
family на прикладі Cassandra**

Перевірив:
Родіонов А. М.

Виконав:
студент I курсу
групи ФБ-41мп
Сахній Н. Р.

Київ 2024

Мета роботи: Дослідити роботу бази даних Cassandra на прикладі кластеру з 3-х нод, налаштувати реплікацію для забезпечення відмовостійкості, а також оцінити ефективність роботи бази даних під час виконання основних операцій читання та запису. Дослідити конфлікт даних при втраті зв'язку між нодами, а також визначити, за яким принципом він був вирішений.

Частина 1. Робота зі структурами даних у Cassandra

Завдання до виконання:

0. INITIALIZATION of Cassandra:

1) `cat ./docker-compose.yml`

```
1  services:
2      cassandra:
3          image: bitnami/cassandra:latest
4          container_name: cassandra_column
5          ports:
6              - '7000:7000' # Внутрішньокластерна комунікація
7              - '9042:9042' # Підключення до БД як CQL-клієнт
8          volumes:
9              - 'cassandra:/bitnami'
10         environment:
11             - CASSANDRA_SEEDS=cassandra
12             - CASSANDRA_PASSWORD_SEEDER=yes
13             - CASSANDRA_PASSWORD=UIOP1234
14     volumes:
15         cassandra:
16             driver: local
```

2) `docker-compose up`

```
t-1000@DESKTOP-DR10PBB MINGW64 /d/KPI/5 курс Марістрат/Проектування ВС/Task_5-Cassandra_Replication
$ docker-compose up
Creating network "task_5-cassandra_replication_default" with the default driver
Creating volume "task_5-cassandra_replication_cassandra" with local driver
Creating cassandra_column ...
Creating cassandra_column ... done
Attaching to cassandra_column
cassandra_column | cassandra 21:08:40.14 INFO ==>
cassandra_column | cassandra 21:08:40.14 INFO ==> Welcome to the Bitnami cassandra container
cassandra_column | cassandra 21:08:40.14 INFO ==> Subscribe to project updates by watching https://github.com/bitnami/containers
cassandra_column | cassandra 21:08:40.14 INFO ==> Submit issues and feature requests at https://github.com/bitnami/containers/issues
cassandra_column | cassandra 21:08:40.14 INFO ==> Upgrade to Tanzu Application Catalog for production environments to access custom-cont
cassandra_column | cassandra 21:08:40.14 INFO ==> naining Software Bill of Materials (SBOM), CVE scan result reports, and VEX documents. To learn more, visit https://bitnami.com/enterprise
cassandra_column | cassandra 21:08:40.15 INFO ==>
cassandra_column | cassandra 21:08:40.15 INFO ==> ** Delaying Cassandra start by 10 seconds **
cassandra_column | cassandra 21:08:50.15 INFO ==> ** Starting Cassandra setup **
cassandra_column | cassandra 21:08:50.18 WARN ==> DB_HOST not set, defaulting to system hostname
cassandra_column | cassandra 21:08:50.18 INFO ==> Validating settings in DB_* env vars..
cassandra_column | cassandra 21:10:08.85 INFO ==> Updating the password for the "cassandra" user...
cassandra_column | cassandra 21:10:09.76 INFO ==> Trying to access CQL server @ 6d465a840cc9
cassandra_column | cassandra 21:10:16.20 INFO ==> Accessed CQL server successfully
cassandra_column | cassandra 21:10:16.21 INFO ==> Password updated successfully
cassandra_column | cassandra 21:10:16.21 INFO ==> ** Cassandra setup finished! **
```

+ CREATING a KEYSPACE

```
t-1000@DESKTOP-DRIOPBB MINGW64 /d/KPI/5 курс Марістрат/Проектування ВС/Task_5-Cassandra_Replication
$ winpty docker exec -it cassandra_column bash
I have no name!@6d465a840cc9:/ $ cqlsh -u cassandra -p UIOP1234

Warning: Using a password on the command line interface can be insecure.
Recommendation: use the credentials file to securely provide the password.

Connected to My Cluster at 127.0.0.1:9042
[cqlsh 6.2.0 | Cassandra 5.0.2 | CQL spec 3.4.7 | Native protocol v5]
Use HELP for help.
cassandra@cqlsh> CREATE KEYSPACE IF NOT EXISTS shop WITH REPLICATION = {'class' : 'SimpleStrategy', 'replication_factor' : 1};
cassandra@cqlsh> DESCRIBE KEYSPACES;

shop      system_auth      system_schema      system_views
system     system_distributed system_traces      system_virtual_schema

cassandra@cqlsh> USE shop;
cassandra@cqlsh:shop>
```

I. WORKING with ITEMS

- Створення таблиці “**items**” (товари), яка міститиме такий набір ознак, як: “**id**” (ідентифікатор), “**genre**” (жанр), “**album**” (назва), “**performer**” (ім’я виконавця), “**price**” (ціна), а також map-структура “**properties**” (додаткові властивості у форматі “ключ-значення”).

```
cassandra@cqlsh:shop> CREATE TABLE items (id uuid, genre text, album text, performer text,
... price decimal, properties map<text, text>, PRIMARY KEY(genre, price, id));
```

```
cassandra@cqlsh:shop> SELECT * FROM items;

genre | price | id | album | performer | properties
-----+-----+---+-----+-----+-----
(0 rows)
```

- Додавання записів (прикладів даних) до щойно створеної таблиці.

```
cassandra@cqlsh:shop> INSERT INTO items (id, genre, album, performer, price, properties) VALUES
... (uuid(), 'Rock', 'From Zero', 'Linkin Park', 3000, {'release_year': '2024', 'availability': 'in stock'});
cassandra@cqlsh:shop> INSERT INTO items (id, genre, album, performer, price, properties) VALUES
... (uuid(), 'Rock', 'Metemora', 'Linkin Park', 2000, {'format': 'Vinyl', 'release_year': '2003', 'availability': 'out of stock'});
cassandra@cqlsh:shop> INSERT INTO items (id, genre, album, performer, price, properties) VALUES
... (uuid(), 'Rock', 'Hybrid Theory', 'Linkin Park', 2000, {'release_year': '2000', 'availability': 'in stock'});
cassandra@cqlsh:shop> INSERT INTO items (id, genre, album, performer, price, properties) VALUES
... (uuid(), 'Pop', 'Радіо Любов', 'Скрябін', 1500, {'format': 'CD', 'release_year': '2012'});
cassandra@cqlsh:shop> INSERT INTO items (id, genre, album, performer, price, properties) VALUES
... (uuid(), 'Pop', 'Добряк', 'Скрябін', 1500, {'format': 'CD', 'release_year': '2013', 'availability': 'limited'});
cassandra@cqlsh:shop> INSERT INTO items (id, genre, album, performer, price, properties) VALUES
... (uuid(), 'Folk', 'Пісні самонаведення', 'Kozak System & Тарас Чубай', 1000, {'format': 'Digital', 'release_year': '2014'});
cassandra@cqlsh:shop> INSERT INTO items (id, genre, album, performer, price, properties) VALUES
... (uuid(), 'Folk', 'Пісні УПА', 'Львівські Музики', 500, {'format': 'CD', 'availability': 'out of stock'});
```

```
cassandra@cqlsh:shop> SELECT * FROM items;

genre | price | id | album | performer | properties
-----+-----+---+-----+-----+-----
Folk | 500 | cfd3921-b7a8-470f-bff3-cb90a99802d | Пісні УПА | Львівські Музики | {'availability': 'out of stock', 'format': 'CD'}
Folk | 1000 | 8760cc0d-2ed9-4ce4-9414-3c0a29c3ccdd | Пісні самонаведення | Kozak System & Тарас Чубай | {'format': 'Digital', 'release_year': '2014'}
Pop | 1500 | 7993cac7-c58e-4a2c-9ec5-8f48db4357f9 | Добряк | Скрябін | {'availability': 'limited', 'format': 'CD', 'release_year': '2013'}
Pop | 1500 | ab1c3cb5-f819-4468-b985-ea009a7ba9ec | Радіо Любов | Скрябін | {'format': 'CD', 'release_year': '2012'}
Rock | 2000 | 902f52d5-9578-49e2-ba6d-23923b0c83e7 | Hybrid Theory | Linkin Park | {'availability': 'in stock', 'release_year': '2000'}
Rock | 2000 | f0f5a2e7-358b-4392-a004-97bdf334487b | Metemora | Linkin Park | {'availability': 'out of stock', 'format': 'Vinyl', 'release_year': '2003'}
Rock | 3000 | e63a9e9f-b194-486f-8661-e9013041c040 | From Zero | Linkin Park | {'availability': 'in stock', 'release_year': '2024'}

(7 rows)
```

1) Напишіть запит, який показує структуру створеної таблиці:

```
cassandra@cqlsh:shop> DESCRIBE items;

CREATE TABLE shop.items (
  genre text,
  price decimal,
  id uuid,
  album text,
  performer text,
  properties map<text, text>,
  PRIMARY KEY (genre, price, id)
) WITH CLUSTERING ORDER BY (price ASC, id ASC)
AND additional_write_policy = '99p'
AND allow_auto_snapshot = true
AND bloom_filter_fp_chance = 0.01
AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
AND cdc = false
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.UnifiedCompactionStrategy', 'max_sstables_to_compact': '64', 'min_sstable_size': '100MiB', 'scaling_parameters': 'T4', 'sstable_growth': '0.3333333333333333', 'target_sstable_size': '1GiB'}
AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
AND memtable = 'default'
AND crc_check_chance = 1.0
AND default_time_to_live = 0
AND extensions = {}
AND gc_grace_seconds = 864000
AND incremental_backups = true
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair = 'BLOCKING'
AND speculative_retry = '99p';
```

2) Напишіть запит, який виводить усі товари в певній категорії, що будуть відсортовані за ціною (порядок за замовчуванням):

```
cassandra@cqlsh:shop> SELECT * FROM items WHERE genre = 'Rock' ORDER BY price;
```

genre	price	id	album	performer	properties
Rock	2000	221227c6-15f8-4daf-9866-3b3ac11837a0	Meteora	Linkin Park	{'availability': 'out of stock', 'format': 'Vinyl', 'release_year': '2003'}
Rock	2000	f4cfda29-1710-44a0-8f3a-5cc709fd36da	Hybrid Theory	Linkin Park	{'availability': 'in stock', 'release_year': '2000'}
Rock	3000	f4e02e60-618f-4aef-83e0-d98744730620	From Zero	Linkin Park	{'availability': 'in stock', 'release_year': '2024'}

(3 rows)

3) Напишіть запити, які вибирають товари за різними критеріями в межах певної категорії (треба використовувати “Materialized views”):

Щоб при виконанні запитів не отримувати помилку «InvalidRequest: Error from server: code=2200 [Invalid query] message="Materialized views are disabled. Enable in cassandra.yaml to use.», потрібно змінити конфігурації:

```
t-1000@DESKTOP-DRI0PBB MINGW64 /d/KPI/5 курс Магістрат/Проектування ВС/Task_5-Cassandra_Replication
$ winpty docker exec -u root -it cassandra_column bash
root@6d465a840cc9:/# find / -name "cassandra.yaml"
/opt/bitnami/cassandra/conf/cassandra.yaml
/opt/bitnami/cassandra/conf.default/cassandra.yaml
find: '/proc/1/map_files': Permission denied
find: '/proc/268/map_files': Permission denied
find: '/proc/2052/map_files': Permission denied

root@6d465a840cc9:/# grep -iH "materialized_views_enabled" /opt/bitnami/cassandra/*/cassandra.yaml
/opt/bitnami/cassandra/conf.default/cassandra.yaml:1901:materialized_views_enabled: true
/opt/bitnami/cassandra/conf/cassandra.yaml:1902:materialized_views_enabled: true
```

○ Назва товару

```
cassandra@cqlsh:shop> CREATE MATERIALIZED VIEW items_by_album AS
... SELECT * FROM shop.items
... WHERE album IS NOT NULL
... AND genre IS NOT NULL
... AND price IS NOT NULL
... AND id IS NOT NULL
... PRIMARY KEY (album, genre, price, id);
```

Warnings :
Materialized views are experimental and are not recommended for production use.

```
cassandra@cqlsh:shop> SELECT * FROM items_by_album WHERE album = 'Пісні самонаведення';
```

album	genre	price	id	performer	properties
Пісні самонаведення	Folk	1000	cc9ed0ee-e903-48c0-bb1a-2fa2e9e8eb7d	Kozak System & Тарас Чубай	{'format': 'Digital', 'release_year': '2014'}

(1 rows)

○ Ціна (в проміжку)

```
cassandra@cqlsh:shop> CREATE MATERIALIZED VIEW items_by_price_in_genre AS
... SELECT * FROM shop.items
... WHERE price IS NOT NULL
... AND genre IS NOT NULL
... AND id IS NOT NULL
... PRIMARY KEY (genre, price, id);
```

Warnings :
Materialized views are experimental and are not recommended for production use.

```
cassandra@cqlsh:shop> SELECT * FROM items_by_price_in_genre WHERE genre = 'Pop' AND price > 1000 AND price < 2500;
```

genre	price	id	album	performer	properties
Pop	1500	16ab72bf-c96f-4d67-809b-30ee25717041	Добряк	Скрябін	{'availability': 'limited', 'format': 'CD', 'release_year': '2013'}
Pop	1500	3f172736-eefd-43bb-864b-cdd36096245a	Радіо Любов	Скрябін	{'format': 'CD', 'release_year': '2012'}

(2 rows)

○ Ціна та виробник

```
cassandra@cqlsh:shop> CREATE MATERIALIZED VIEW items_by_price_in_performer AS
... SELECT * FROM shop.items
... WHERE price IS NOT NULL
... AND performer IS NOT NULL
... AND genre IS NOT NULL
... AND id IS NOT NULL
... PRIMARY KEY (performer, price, genre, id);
```

Warnings :
Materialized views are experimental and are not recommended for production use.

```
cassandra@cqlsh:shop> SELECT * FROM items_by_price_in_performer WHERE performer = 'Львівські Музики' AND price < 1000;
```

performer	price	genre	id	album	properties
Львівські Музики	500	Folk	843dc3fa-b091-4103-b061-88bb09861dbd	Пісні УПА	{'availability': 'out of stock', 'format': 'CD'}

(1 rows)

II. WORKING with ORDERS

- Створення таблиці “**orders**” (замовлення), яка міститиме такий набір ознак, як (тобто інформація про замовлення): “**id**” (ідентифікатор),

“customer” (ім’я покупця), “items” (перелік товарів), “order_cost” (загальна вартість замовлення), “order_date” (дата замовлення):

```
cassandra@cqlsh:shop> CREATE TABLE orders (id uuid, customer text, items list<uuid>,
... order_cost decimal, order_date date, PRIMARY KEY(customer, order_date));
```

```
cassandra@cqlsh:shop> SELECT * FROM orders;
```

```
customer | order_date | id | items | order_cost
-----+-----+---+-----+-----
(0 rows)
```

➤ Додавання записів (прикладів даних) до щойно створеної таблиці.

```
cassandra@cqlsh:shop> INSERT INTO orders (id, customer, items, order_date, order_cost) VALUES
... (uuid(), 'Назар Сахній', [221227c6-15f8-4daf-9866-3b3ac11837a0, 843dc3fa-b091-4103-b061-88bb09861dbd], '2024-10-14', 2500);
cassandra@cqlsh:shop> INSERT INTO orders (id, customer, items, order_date, order_cost) VALUES
... (uuid(), 'Назар Сахній', [cc9ed0ee-e903-48c0-bb1a-2fa2e9e8eb7d], '2024-10-27', 1000);
cassandra@cqlsh:shop> INSERT INTO orders (id, customer, items, order_date, order_cost) VALUES
... (uuid(), 'Назар Сахній', [f4cfda29-1710-44a0-8f3a-5cc709fd36da, f4e02e60-618f-4aef-83e0-d98744730620], '2024-11-15', 5000);
cassandra@cqlsh:shop> INSERT INTO orders (id, customer, items, order_date, order_cost) VALUES
... (uuid(), 'Тарас Шевченко', [cc9ed0ee-e903-48c0-bb1a-2fa2e9e8eb7d, 16ab72bf-c96f-4d67-809b-30ee25717041], '2024-11-09', 2500);
cassandra@cqlsh:shop> INSERT INTO orders (id, customer, items, order_date, order_cost) VALUES
... (uuid(), 'Андрій Мельник', [3f172736-eeef-d43bb-864b-cdd36096245a, 221227c6-15f8-4daf-9866-3b3ac11837a0], '2024-11-30', 3500);
cassandra@cqlsh:shop> INSERT INTO orders (id, customer, items, order_date, order_cost) VALUES
... (uuid(), 'Андрій Мельник', [843dc3fa-b091-4103-b061-88bb09861dbd], '2024-12-18', 500);
```

```
cassandra@cqlsh:shop> SELECT * FROM orders;
```

customer	order_date	id	items	order_cost
Тарас Шевченко	2024-11-09	ce3ebc9d-cf64-49f9-a8f7-08781c5aalea	[cc9ed0ee-e903-48c0-bb1a-2fa2e9e8eb7d, 16ab72bf-c96f-4d67-809b-30ee25717041]	2500
Назар Сахній	2024-10-14	fe56e40b-b50c-43b2-bb66-228a33539344	[221227c6-15f8-4daf-9866-3b3ac11837a0, 843dc3fa-b091-4103-b061-88bb09861dbd]	2500
Назар Сахній	2024-10-27	8801154d-5728-4249-81da-0d5857bbcfb	[cc9ed0ee-e903-48c0-bb1a-2fa2e9e8eb7d]	1000
Назар Сахній	2024-11-15	b47ca8c1-0cde-4b8a-9c94-79e4c8502a21	[f4cfda29-1710-44a0-8f3a-5cc709fd36da, f4e02e60-618f-4aef-83e0-d98744730620]	5000
Андрій Мельник	2024-11-30	c49b9933-fbae-4a12-bf00-00d20571d138	[3f172736-eeef-d43bb-864b-cdd36096245a, 221227c6-15f8-4daf-9866-3b3ac11837a0]	3500
Андрій Мельник	2024-12-18	1c0d6a9d-e3f5-4d01-a19d-e69ad892a0e6	[843dc3fa-b091-4103-b061-88bb09861dbd]	500

(6 rows)

1) Напишіть запит, який показує структуру створеної таблиці.

```
cassandra@cqlsh:shop> DESCRIBE orders;
```

```
CREATE TABLE shop.orders (
  customer text,
  order_date date,
  id uuid,
  order_cost decimal,
  items list<uuid>,
  PRIMARY KEY (customer, order_date)
) WITH CLUSTERING ORDER BY (order_date ASC)
AND additional_write_policy = '99p'
AND allow_auto_snapshot = true
AND bloom_filter_fp_chance = 0.01
AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
AND cdc = false
AND comment = ''
AND compaction = {'class': 'org.apache.cassandra.db.compaction.UnifiedCompactionStrategy', 'max_sstables_to_compact': '64', 'min_sstable_size': '100MiB', 'scaling_parameters': 'T4', 'sstable_growth': '0.3333333333333333', 'target_sstable_size': '1GiB'}
AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
AND memtable = 'default'
AND crc_check_chance = 1.0
AND default_time_to_live = 0
AND extensions = {}
AND gc_grace_seconds = 864000
AND incremental_backups = true
AND max_index_interval = 2048
AND memtable_flush_period_in_ms = 0
AND min_index_interval = 128
AND read_repair = 'BLOCKING'
AND speculative_retry = '99p';
```

2) Для деякого покупця вивести відсортовані за датою його замовлення.

```
cassandra@cqlsh:shop> SELECT * FROM orders WHERE customer = 'Назар Сахній' ORDER BY order_date;
```

customer	order_date	id	items	order_cost
Назар Сахній	2024-10-14	fe56e40b-b50c-43b2-bb66-228a33539344	[221227c6-15f8-4daf-9866-3b3ac11837a0, 843dc3fa-b091-4103-b061-88bb09861dbd]	2500
Назар Сахній	2024-10-27	8801154d-5728-4249-81da-0d55857bbcfb	[cc9ed0ee-e903-48c0-bb1a-2fa2e9e8eb7d]	1000
Назар Сахній	2024-11-15	b47ca8c1-0cde-4b8a-9c94-79e4c8502a21	[f4cfda29-1710-44a0-8f3a-5cc709fd36da, f4e02e60-618f-4aef-83e0-d98744730620]	5000

(3 rows)

```
cassandra@cqlsh:shop> SELECT * FROM orders WHERE customer = 'Тарас Шевченко' ORDER BY order_date;
```

customer	order_date	id	items	order_cost
Тарас Шевченко	2024-11-09	ce3ebc9d-cf64-49f9-a8f7-08781c5aa1ea	[cc9ed0ee-e903-48c0-bb1a-2fa2e9e8eb7d, 16ab72bf-c96f-4d67-809b-30ee25717041]	2500

(1 rows)

```
cassandra@cqlsh:shop> SELECT * FROM orders WHERE customer = 'Андрій Мельник' ORDER BY order_date;
```

customer	order_date	id	items	order_cost
Андрій Мельник	2024-11-30	c49b9933-fbae-4a12-bf00-00d20571d138	[3f172736-eeef-43bb-864b-cdd36096245a, 221227c6-15f8-4daf-9866-3b3ac11837a0]	3500
Андрій Мельник	2024-12-18	1c0d6a9d-e3f5-4d01-a19d-e69ad892a0e6	[843dc3fa-b091-4103-b061-88bb09861dbd]	500

(2 rows)

3) Для кожного покупця розрахувати загальну суму всіх його замовлень.

```
cassandra@cqlsh:shop> SELECT customer, SUM(order_cost) FROM orders GROUP BY customer;
```

customer	system.sum(order_cost)
Тарас Шевченко	2500
Назар Сахній	8500
Андрій Мельник	4000

(3 rows)

4) Для кожного замовлення вивести час, коли його ціна була зафіксована в базі даних (використовуючи запит “SELECT WRITETIME”).

```
cassandra@cqlsh:shop> SELECT customer, order_date, order_cost, WRITETIME(order_cost) FROM orders;
```

customer	order_date	order_cost	writetime(order_cost)
Тарас Шевченко	2024-11-09	2500	1734522726946656
Назар Сахній	2024-10-14	2500	1734522685002012
Назар Сахній	2024-10-27	1000	1734522698675932
Назар Сахній	2024-11-15	5000	1734522712284907
Андрій Мельник	2024-11-30	3500	1734522739839897
Андрій Мельник	2024-12-18	500	1734522754883668

(6 rows)

Частина 2. Налаштування реплікації у Cassandra

Завдання до виконання:

1. Сконфігурувати кластер, що складатиметься із 3-х нод:

```
(root@saan24)-[/home/nazar]
# docker network create labReplicaSet
448ea522c1b323e959c70510e60b541fc0aa26e4702e8f47e3061ab35abc858d

(root@saan24)-[/home/nazar]
# docker run --name cassandra-node1 -d \
--network labReplicaSet \
--volume cassandra-node1 \
cassandra:latest
374e61b5d2f3ce041550592a4b6dd0c6b32b86bfd8fe3e7cf7521169db92f3b6

(root@saan24)-[/home/nazar]
# docker run --name cassandra-node2 -d \
--network labReplicaSet \
--volume cassandra-node2 \
-e CASSANDRA_SEEDS=cassandra-node1 \
cassandra:latest
22d6f5530209c7caa17410297e5d6dcac441e8b427877b788ff8c4e196682134

(root@saan24)-[/home/nazar]
# docker run --name cassandra-node3 -d \
--network labReplicaSet \
--volume cassandra-node3 \
-e CASSANDRA_SEEDS=cassandra-node1 \
cassandra:latest
e550c93a2d3e8aed528bc9f252e1187b754873c83339f740ff89a146d381e88
```

```
(root@saan24)-[/home/nazar]
# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                                                                                               NAMES
e550c93a2d3e   cassandra:late  "docker-entrypoint.s..." About a minute ago Up About a minute 7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp             cassandra-node3
22d6f5530209   cassandra:late  "docker-entrypoint.s..." About a minute ago Up About a minute 7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp             cassandra-node2
374e61b5d2f3   cassandra:late  "docker-entrypoint.s..." About a minute ago Up About a minute 7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp             cassandra-node1
```

2. Перевірити правильність конфігурації за допомогою:

➤ nodetool status

```
(root@saan24)-[/home/nazar]
# docker exec -it cassandra-node1 nodetool status
Datacenter: datacenter1

Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens    Owns (effective)  Host ID                               Rack
UN 172.25.0.2    119.83 KiB    16        64.7%             f701469a-ab34-4892-98f6-3a77c5b44683 rack1
UN 172.25.0.4    126.36 KiB    16        76.0%             522f4f0f-6ea9-4be3-a7b0-39bab5e182cb rack1
UN 172.25.0.3    114.67 KiB    16        59.3%             295d9815-8b3a-431b-a9c3-743213e85844 rack1
```

```
(root@saan24)-[/home/nazar]
# docker exec -it cassandra-node1 nodetool status system_schema
Datacenter: datacenter1

Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens    Owns  Host ID                               Rack
UN 172.25.0.2    119.83 KiB    16        ?     f701469a-ab34-4892-98f6-3a77c5b44683 rack1
UN 172.25.0.4    126.36 KiB    16        ?     522f4f0f-6ea9-4be3-a7b0-39bab5e182cb rack1
UN 172.25.0.3    114.67 KiB    16        ?     295d9815-8b3a-431b-a9c3-743213e85844 rack1
```


➤ nodetool info

```
(root@sazan24)-[/home/nazar]
# docker exec -it cassandra-node2 nodetool info
ID : 295d9815-8b3a-431b-a9c3-743213e85844
Gossip active : true
Native Transport active: true
Load : 114.67 KiB
Uncompressed load : 151.94 KiB
Generation No : 1734637989
Uptime (seconds) : 742
Heap Memory (MB) : 786.33 / 2000.00
Off Heap Memory (MB) : 0.00
Data Center : datacenter1
Rack : rack1
Exceptions : 0
Key Cache : entries 10, size 896 bytes, capacity 100 MiB, 111 hits, 128 requests, 0.867
Row Cache : entries 0, size 0 bytes, capacity 0 bytes, 0 hits, 0 requests, NaN recent hi
Counter Cache : entries 0, size 0 bytes, capacity 50 MiB, 0 hits, 0 requests, NaN recent hit
Network Cache : size 8 MiB, overflow size: 0 bytes, capacity 125 MiB
Percent Repaired : 100.0%
Token : (invoke with -T/--tokens to see all 16 tokens)
Bootstrap state : COMPLETED
Bootstrap failed : false
Decommissioning : false
Decommission failed : false
```

➤ nodetool describcluster

```
(root@sazan24)-[/home/nazar]
# docker exec -it cassandra-node3 nodetool describcluster
Cluster Information:
  Name: Test Cluster
  Snitch: org.apache.cassandra.locator.SimpleSnitch
  DynamicEndPointSnitch: enabled
  Partitioner: org.apache.cassandra.dht.Murmur3Partitioner
  Schema versions:
    d03783d7-b468-3c1a-82f1-8e30b2edde8b: [172.25.0.3, 172.25.0.2, 172.25.0.4]

Stats for all nodes:
  Live: 3
  Joining: 0
  Moving: 0
  Leaving: 0
  Unreachable: 0

Data Centers:
  datacenter1 #Nodes: 3 #Down: 0

Database versions:
  5.0.2: [172.25.0.3:7000, 172.25.0.2:7000, 172.25.0.4:7000]

Keyspaces:
  system_auth → Replication class: SimpleStrategy {replication_factor=1}
  system_distributed → Replication class: SimpleStrategy {replication_factor=3}
  system_traces → Replication class: SimpleStrategy {replication_factor=2}
  system_schema → Replication class: LocalStrategy {}
  system → Replication class: LocalStrategy {}
```

3. Створити кейспейси з “replication_factor={1, 2, 3}” та “SimpleStrategy”.

```
cqlsh> CREATE KEYSPACE replication_factor1 WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> CREATE KEYSPACE replication_factor2 WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 2};
cqlsh> CREATE KEYSPACE replication_factor3 WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 3};
```

4. В кожному з кейспейсів створити прості таблиці.

```
cqlsh> CREATE TABLE IF NOT EXISTS replication_factor1.lab(record text, PRIMARY KEY (record));
cqlsh> CREATE TABLE IF NOT EXISTS replication_factor2.lab(record text, PRIMARY KEY (record));
cqlsh> CREATE TABLE IF NOT EXISTS replication_factor3.lab(record text, PRIMARY KEY (record));
```

5. Спробувати “писати” й “читати” в ці таблиці, підключаючись на різні ноди.

❖ `cassandra_node1 cqlsh`

```
cqlsh> INSERT INTO replication_factor1.lab(record) VALUES ('[1] Nazar Sakhnii');
cqlsh>
cqlsh> SELECT * FROM replication_factor1.lab;

record
-----
[1] Nazar Sakhnii
(1 rows)
```

❖ `cassandra_node2 cqlsh`

```
cqlsh> INSERT INTO replication_factor2.lab(record) VALUES ('[2] Nazar Sakhnii');
cqlsh>
cqlsh> SELECT * FROM replication_factor2.lab;

record
-----
[2] Nazar Sakhnii
(1 rows)
```

❖ `cassandra_node3 cqlsh`

```
cqlsh> INSERT INTO replication_factor3.lab(record) VALUES ('[3] Nazar Sakhnii');
cqlsh>
cqlsh> SELECT * FROM replication_factor3.lab;

record
-----
[3] Nazar Sakhnii
(1 rows)
```

6. Вставити дані в створені таблиці і поглянути на їх розподіл по нодах кластера для кожного з кейспесів (команда “`nodetool status`”).

```
(root@saan24)-[/home/nazar]
# docker exec -it cassandra-node1 nodetool status replication_factor1
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens    Owns (effective)  Host ID                               Rack
UN 172.25.0.2    122.81 KiB    16        32.7%             f701469a-ab34-4892-98f6-3a77c5b44683 rack1
UN 172.25.0.4    145.17 KiB    16        35.7%             522f4f0f-6ea9-4be3-a7b0-39bab5e182cb rack1
UN 172.25.0.3    117.66 KiB    16        31.6%             295d9815-8b3a-431b-a9c3-743213e85844 rack1
```

```
(root@sazan24)-[/home/nazar]
# docker exec -it cassandra-node1 nodetool status replication_factor2
Datacenter: datacenter1
```

Status=Up/Down							
/ State=Normal/Leaving/Joining/Moving							
--	Address	Load	Tokens	Owns (effective)	Host ID	Rack	
UN	172.25.0.2	122.81 KiB	16	64.7%	f701469a-ab34-4892-98f6-3a77c5b44683	rack1	
UN	172.25.0.4	145.17 KiB	16	76.0%	522f4f0f-6ea9-4be3-a7b0-39bab5e182cb	rack1	
UN	172.25.0.3	117.66 KiB	16	59.3%	295d9815-8b3a-431b-a9c3-743213e85844	rack1	

```
(root@sazan24)-[/home/nazar]
# docker exec -it cassandra-node1 nodetool status replication_factor3
Datacenter: datacenter1
```

Status=Up/Down							
/ State=Normal/Leaving/Joining/Moving							
--	Address	Load	Tokens	Owns (effective)	Host ID	Rack	
UN	172.25.0.2	122.81 KiB	16	100.0%	f701469a-ab34-4892-98f6-3a77c5b44683	rack1	
UN	172.25.0.4	145.17 KiB	16	100.0%	522f4f0f-6ea9-4be3-a7b0-39bab5e182cb	rack1	
UN	172.25.0.3	117.66 KiB	16	100.0%	295d9815-8b3a-431b-a9c3-743213e85844	rack1	

7. Для якогось запису з кейспейсів вивести ноди, на яких зберігаються дані.

```
cqlsh> INSERT INTO replication_factor1.lab(record) VALUES ('FirstPoint');
cqlsh> INSERT INTO replication_factor2.lab(record) VALUES ('SecondPoint');
cqlsh> INSERT INTO replication_factor3.lab(record) VALUES ('ThirdPoint');
```

```
root@374e61b5d2f3:/# nodetool getendpoints replication_factor1 lab "FirstPoint"
172.25.0.3
```

```
root@374e61b5d2f3:/# nodetool getendpoints replication_factor2 lab "SecondPoint"
172.25.0.2
172.25.0.4
```

```
root@374e61b5d2f3:/# nodetool getendpoints replication_factor3 lab "ThirdPoint"
172.25.0.3
172.25.0.2
172.25.0.4
```

8. Відключити одну з нод. Для кожного з кейспейсів перевірити, з якими рівнями **consistency** можемо “писати” й “читати”:

```
(root@sazan24)-[/home/nazar]
# docker stop cassandra-node3
cassandra-node3
```

```
(root@sazan24)-[/home/nazar]
# docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
22d6f5530209	cassandra:latest	"docker-entrypoint.s..."	About an hour ago	Up About an hour	7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp	cassandra-node2
374e61b5d2f3	cassandra:latest	"docker-entrypoint.s..."	About an hour ago	Up About an hour	7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp	cassandra-node1

```
(root@sazan24)-[/home/nazar]
#
```

– для KEYSpace 3 “replication_factor=1”: CONSISTENCY ONE

```
cqlsh> INSERT INTO replication_factor1.lab(record) VALUES ('1-CONSISTENCY_ONE');
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>:
consistency level ONE" info={\'consistency\': \'ONE\', \'required_replicas\': 1, \'alive_replicas\': 0}})})
cqlsh>
cqlsh> SELECT * FROM replication_factor1.lab;
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>:
consistency level ONE" info={\'consistency\': \'ONE\', \'required_replicas\': 1, \'alive_replicas\': 0}})})
```

```
cqlsh> INSERT INTO replication_factor2.lab(record) VALUES ('2-CONSISTENCY_ONE');
cqlsh>
cqlsh> SELECT * FROM replication_factor2.lab;
```

```
record
-----
2-CONSISTENCY_ONE
[2] Nazar Sakhnii
    SecondPoint
(3 rows)
```

```
cqlsh> INSERT INTO replication_factor3.lab(record) VALUES ('3-CONSISTENCY_ONE');
cqlsh>
cqlsh> SELECT * FROM replication_factor3.lab;
```

```
record
-----
[3] Nazar Sakhnii
    ThirdPoint
3-CONSISTENCY_ONE
(3 rows)
```

– для KEYSpace 3 “replication_factor=2”: CONSISTENCY TWO

```
cqlsh> INSERT INTO replication_factor1.lab(record) VALUES ('1-CONSISTENCY_TWO');
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>:
consistency level TWO" info={\'consistency\': \'TWO\', \'required_replicas\': 2, \'alive_replicas\': 1}})})
cqlsh>
cqlsh> SELECT * FROM replication_factor1.lab;
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>:
consistency level TWO" info={\'consistency\': \'TWO\', \'required_replicas\': 2, \'alive_replicas\': 0}})})
```

```
cqlsh> INSERT INTO replication_factor2.lab(record) VALUES ('2-CONSISTENCY_TWO');
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>:
consistency level TWO" info={\'consistency\': \'TWO\', \'required_replicas\': 2, \'alive_replicas\': 1}})})
cqlsh>
cqlsh> SELECT * FROM replication_factor2.lab;
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>:
consistency level TWO" info={\'consistency\': \'TWO\', \'required_replicas\': 2, \'alive_replicas\': 1}})})
```

```
cqlsh> INSERT INTO replication_factor3.lab(record) VALUES ('3-CONSISTENCY_TWO');
cqlsh>
cqlsh> SELECT * FROM replication_factor3.lab;
```

```
record
-----
[3] Nazar Sakhnii
    ThirdPoint
3-CONSISTENCY_TWO
3-CONSISTENCY_ONE
(4 rows)
```

– для KEYSpace 3 “replication_factor=3”: CONSISTENCY THREE

```
cqlsh> CONSISTENCY ONE/TWO/THREE
Improper CONSISTENCY command.
```



```
cqlsh> INSERT INTO replication_factor1.lab(record) VALUES ('1-CONSISTENCY_THREE');
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>: Un
consistency level THREE" info={'consistency': 'THREE', 'required_replicas': 3, 'alive_replicas': 1}}))
cqlsh>
cqlsh> SELECT * FROM replication_factor1.lab;
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>: Un
consistency level THREE" info={'consistency': 'THREE', 'required_replicas': 3, 'alive_replicas': 0}}))
```

```
cqlsh> INSERT INTO replication_factor2.lab(record) VALUES ('2-CONSISTENCY_THREE');
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>: Un
consistency level THREE" info={'consistency': 'THREE', 'required_replicas': 3, 'alive_replicas': 1}}))
cqlsh>
cqlsh> SELECT * FROM replication_factor2.lab;
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>: Un
consistency level THREE" info={'consistency': 'THREE', 'required_replicas': 3, 'alive_replicas': 1}}))
```

```
cqlsh> INSERT INTO replication_factor3.lab(record) VALUES ('3-CONSISTENCY_THREE');
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>: Un
consistency level THREE" info={'consistency': 'THREE', 'required_replicas': 3, 'alive_replicas': 2}}))
cqlsh>
cqlsh> SELECT * FROM replication_factor3.lab;
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 datacenter1>: Un
consistency level THREE" info={'consistency': 'THREE', 'required_replicas': 3, 'alive_replicas': 2}}))
```

9. Зробити так, щоб три ноди працювали, але не бачили одна одну по мережі (тобто заблокувати чи відключити зв'язок між ними).

```
(root@sazan24)-[/home/nazar]
# docker start cassandra-node3
cassandra-node3
```

```
(root@sazan24)-[/home/nazar]
# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
e550c93a2d3e   cassandra:late "docker-entrypoint.s..." 3 hours ago   Up 3 minutes  7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp   cassandra-node3
22d6f5530209   cassandra:late "docker-entrypoint.s..." 3 hours ago   Up 3 hours    7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp   cassandra-node2
374e61b5d2f3   cassandra:late "docker-entrypoint.s..." 3 hours ago   Up 3 hours    7000-7001/tcp, 7199/tcp, 9042/tcp, 9160/tcp   cassandra-node1
```

```
(root@sazan24)-[/home/nazar]
# docker exec -it cassandra-node1 nodetool disablegossip

(root@sazan24)-[/home/nazar]
# docker exec -it cassandra-node2 nodetool disablegossip

(root@sazan24)-[/home/nazar]
# docker exec -it cassandra-node3 nodetool disablegossip
```

10. Для кейспейсу з **replication_factor=3** задати рівень **consistency=1**. Виконайте по черзі запис значення з однаковим Primary Key, але різними іншими значенням окремо на кожну з нод (тобто створіть конфлікт).

```
cqlsh:replication_factor3> CONSISTENCY ONE
Consistency level set to ONE.
```

❖ cassandra_node1 cqlsh

```
cqlsh:replication_factor3> INSERT INTO lab(record, value) VALUES ('conflict', 'who');
cqlsh:replication_factor3>
cqlsh:replication_factor3> SELECT record, value, writetime(value) FROM lab;
```

record	value	writetime(value)
[3] Nazar Sakhnii	null	null
ThirdPoint	null	null
3-CONSISTENCY_TWO	null	null
conflict	who	1734647854561602
3-CONSISTENCY_ONE	null	null

(5 rows)

❖ cassandra_node2 cqlsh

```
cqlsh:replication_factor3> SELECT record, value, writetime(value) FROM lab;
```

record	value	writetime(value)
[3] Nazar Sakhnii	null	null
ThirdPoint	null	null
3-CONSISTENCY_TWO	null	null
conflict	where	1734647884825341
3-CONSISTENCY_ONE	null	null

(5 rows)

❖ cassandra_node3 cqlsh

```
cqlsh:replication_factor3> INSERT INTO lab(record, value) VALUES ('conflict', 'when');
cqlsh:replication_factor3>
cqlsh:replication_factor3> SELECT record, value, writetime(value) FROM lab;
```

record	value	writetime(value)
[3] Nazar Sakhnii	null	null
ThirdPoint	null	null
3-CONSISTENCY_TWO	null	null
conflict	when	1734647915192812
3-CONSISTENCY_ONE	null	null

(5 rows)

11.Відновити зв'язок між нодами, і перевірити, що вони знову об'єдналися у кластер. Визначити яким чином була вирішений конфлікт даних та яке значення було прийнято кластером та за яким принципом.

```
(root@saan24)-[/home/nazar]
# docker exec -it cassandra-node1 nodetool enablegossip

(root@saan24)-[/home/nazar]
# docker exec -it cassandra-node2 nodetool enablegossip

(root@saan24)-[/home/nazar]
# docker exec -it cassandra-node3 nodetool enablegossip
```



```
cqlsh> SELECT record, value, writetime(value) FROM replication_factor3.lab;
```

record	value	writetime(value)
[3] Nazar Sakhnii	null	null
ThirdPoint	null	null
3-CONSISTENCY_TWO	null	null
conflict	who	1734647854561602
3-CONSISTENCY_ONE	null	null

(5 rows)

```
root@e550c93a2d3e:/# nodetool getendpoints replication_factor3 lab "when"
```

```
172.25.0.4  
172.25.0.2  
172.25.0.3
```

Отже, як можна помітити, Cassandra вирішила конфлікт таким чином, що реплікувала запис із найновішим таймштампом (“when”). Це стандартний механізм вирішення конфліктів у Cassandra за допомогою моделі **Last Write Wins** (LWW). Якщо на різних нодах є різні значення для одного й того самого запису, Cassandra автоматично вибирає той, який має найновіший таймштамп, встановлений при внесенні змін.