Rúbrica 5: Sigue el tutorial Cassandra

También tenemos que tener mínimo 4 GB de RAM y 4 CPUs para crear el contenedor. Creamos el archivo docker-compose.yml y la carpeta donde guardarlo:

```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19045.2846]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\aleja>cd BBDD/

C:\Users\aleja\BBDD>mkdir cassandra-docker

C:\Users\aleja\BBDD>cd cassandra-docker

C:\Users\aleja\BBDD\cassandra-docker>touch docker-compose.yml

C:\Users\aleja\BBDD\cassandra-docker>nano docker-compose.yml

C:\Users\aleja\BBDD\cassandra-docker>cat docker-compose.yml
version: '3'
services:
# The first node and config in the first datacenter.
    node1:
        image: datastax/dse-server:6.8.16-ubi7
        container_name: DSE-6_node1
        hostname: node1
        # use static ip address
        networks:
            dc1ring:
                ipv4_address: 172.30.0.2
        # Maps cassandra exercises to a local folder.
        # This preserves data across container restarts.
        volumes:
            - ./musicdb:/opt/dse/musicdb
        # Docker container environment variable. We are using the
        # CASSANDRA_CLUSTER_NAME to name the cluster. This needs to be the same
        # across clusters. We are also declaring that node1 is a seed node etc.
        environment:
            - DS_LICENSE=accept
            - SEEDS=node1
            - START_RPC=false
            - CLUSTER_NAME=dse51_cluster
            - NUM_TOKENS=3
            - DC=DC1
            - RACK=RAC1
            - MAX_HEAP_SIZE=1000000000
```

Símbolo del sistema

```
            - RACK=RAC1
            - MAX_HEAP_SIZE=1000000000
        # Exposing ports for inter cluster communication
        expose:
            # Intra-node communication
            - 7000
            # TLS intra-node communication
            - 7001
            # JMX
            - 7199
            # CQL
            - 9042
            # CQL SSL
            - 9142
        ports:
            - 9042:9042
        ulimits:
            memlock: -1
            nproc: 32768
            nofile: 100000
    node2:
        image: datastax/dse-server:6.8.16-ubi7
        container_name: DSE-6_node2
        hostname: node2
        networks:
            dc1ring:
                ipv4_address: 172.30.0.3
        volumes:
            - ./musicdb:/opt/dse/musicdb
        environment:
            - DS_LICENSE=accept
            - SEEDS=node1
            - START_RPC=false
            - CLUSTER_NAME=dse51_cluster
            - NUM_TOKENS=3
            - DC=DC2
            - RACK=RAC1
            - MAX_HEAP_SIZE=1000000000
        expose:
```

Símbolo del sistema

```
            - RACK=RAC1
            - MAX_HEAP_SIZE=1000000000
        expose:
            - 7000
            - 7001
            - 7199
            - 9042
            - 9142
        ports:
            - 9043:9042
        ulimits:
            memlock: -1
            nproc: 32768
            nofile: 100000
        depends_on:
            - node1
    node3:
        image: datastax/dse-server:6.8.16-ubi7
        container_name: DSE-6_node3
        hostname: node3
        networks:
            dc1ring:
                ipv4_address: 172.30.0.4
        volumes:
            - ./musicdb:/opt/dse/musicdb
        environment:
            - DS_LICENSE=accept
            - SEEDS=node1
            - START_RPC=false
            - CLUSTER_NAME=dse51_cluster
            - NUM_TOKENS=3
            - DC=DC1
            - RACK=RAC1
            - MAX_HEAP_SIZE=1000000000
        expose:
            - 7000
            - 7001
            - 7199
            - 9042
            - 9142
        ports:
```

Símbolo del sistema

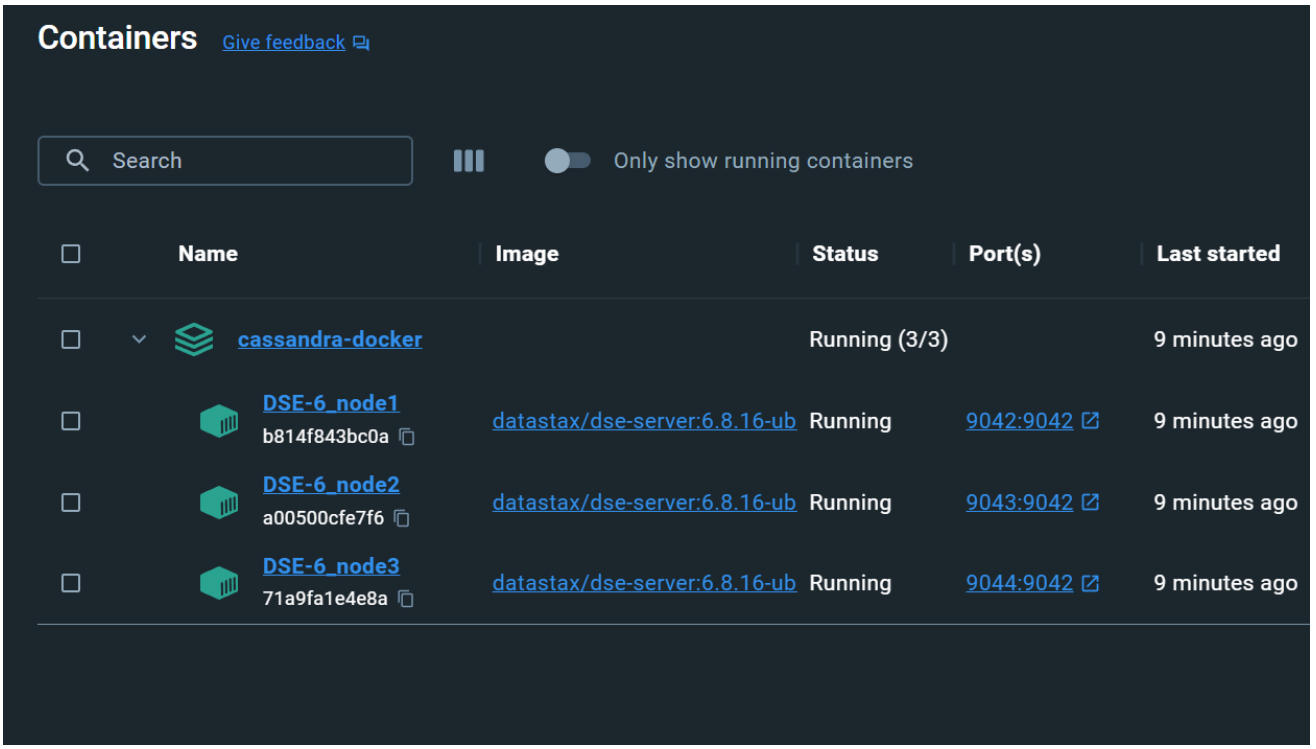
```
            - SEEDS=node1
            - START_RPC=false
            - CLUSTER_NAME=dse51_cluster
            - NUM_TOKENS=3
            - DC=DC1
            - RACK=RAC1
            - MAX_HEAP_SIZE=1000000000
        expose:
            - 7000
            - 7001
            - 7199
            - 9042
            - 9142
        ports:
            - 9044:9042
        ulimits:
            memlock: -1
            nproc: 32768
            nofile: 100000
        depends_on:
            - node1
networks:
    dc1ring:
        ipam:
            driver: default
            config:
                - subnet: 172.30.0.0/16
```

Ahora levantamos el contenedor:

```
C:\Users\aleja\BBDD\cassandra-docker>docker compose -f docker-compose.yml up
[+] Running 19/19
 ⠿ node3 Pulled
 ⠿ node1 Pulled
 ⠿ node2 16 layers [⠿⠿⠿⠿⠿⠿⠿⠿⠿⠿⠿⠿⠿⠿⠿⠿]      0B/0B      Pulled
   ⠿ c13bd28f35dc Pull complete
   ⠿ 0dc69daaa449 Pull complete
   ⠿ 755d653b6228 Pull complete
   ⠿ a5ea20cc493e Pull complete
   ⠿ 4bf873279014 Pull complete
   ⠿ 44fde88c4818 Pull complete
   ⠿ f4e3008bfe4a Pull complete
   ⠿ ff9002066711 Pull complete
   ⠿ aa5021e1dd75 Pull complete
   ⠿ 0b4e5582d536 Pull complete
   ⠿ d3dcde48834f Pull complete
   ⠿ 54c750f038bd Pull complete
   ⠿ 5c0d7d13972c Pull complete
   ⠿ 93b60e864ca0 Pull complete
   ⠿ 8f8472b3ed49 Pull complete
   ⠿ 1a45cbded976 Pull complete
[+] Running 4/4
 ⠿ Network cassandra-docker_dc1ring  Created
 ⠿ Container DSE-6_node1             Created
 ⠿ Container DSE-6_node2             Created
 ⠿ Container DSE-6_node3             Created
Attaching to DSE-6_node1, DSE-6_node2, DSE-6_node3
DSE-6_node1  | Applying changes to /opt/dse/resources/cassandra/conf/cassandra.yaml ...
DSE-6_node1  | done.
DSE-6_node1  | Applying changes to /opt/dse/resources/cassandra/conf/cassandra-rackdc.properties ..
DSE-6_node1  | done.
DSE-6_node1  | Running dse cassandra -f -R
```

```
 ⠿ 0b4e5582d536 Pull complete
 ⠿ d3dcde48834f Pull complete
 ⠿ 54c750f038bd Pull complete
 ⠿ 5c0d7d13972c Pull complete
 ⠿ 93b60e864ca0 Pull complete
 ⠿ 8f8472b3ed49 Pull complete
 ⠿ 1a45cbded976 Pull complete
[+] Running 4/4
 ⠿ Network cassandra-docker_dc1ring  Created
 ⠿ Container DSE-6_node1             Created
 ⠿ Container DSE-6_node2             Created
 ⠿ Container DSE-6_node3             Created
Attaching to DSE-6_node1, DSE-6_node2, DSE-6_node3
DSE-6_node1  | Applying changes to /opt/dse/resources/cassandra/conf/cassandra.yaml ...
DSE-6_node1  | done.
DSE-6_node1  | Applying changes to /opt/dse/resources/cassandra/conf/cassandra-rackdc.properties ...
DSE-6_node1  | done.
DSE-6_node1  | Running dse cassandra -f -R
DSE-6_node2  | Applying changes to /opt/dse/resources/cassandra/conf/cassandra.yaml ...
DSE-6_node2  | done.
DSE-6_node2  | Applying changes to /opt/dse/resources/cassandra/conf/cassandra-rackdc.properties ...
DSE-6_node2  | done.
DSE-6_node2  | Running dse cassandra -f -R
DSE-6_node3  | Applying changes to /opt/dse/resources/cassandra/conf/cassandra.yaml ...
DSE-6_node3  | done.
DSE-6_node3  | Applying changes to /opt/dse/resources/cassandra/conf/cassandra-rackdc.properties ...
DSE-6_node3  | done.
DSE-6_node3  | Running dse cassandra -f -R
```

Comprobamos que estén levantados:



Con este comando nos conectaremos al nodo por ejemplo:

```
C:\Users\aleja\BBDD\cassandra-docker>docker exec -it DSE-6_node1 bash
```

Usamos este comando para poder comenzar a trabajar con Cassandra y hacer consultas:

```
cqlsh
dse51_cluster at 127.0.0.1:9042.
| DSE 6.8.16 | CQL spec 3.4.5 | DSE protocol v2]
help.
```

```
DESC keyspaces;

_virtual_schema  system_schema  dse_leases
stem_local       system_auth    system_backups
curity           system_views   dse_insights
```

Creamos el keyspace musicDB como ejemplo

```
CREATE KEYSPACE musicDb WITH replication = {'class': 'SimpleStrategy', 'replicati
 : '3'};
```

y luego creamos la tabla

```
USE musicdb ;
usicdb> CREATE TABLE musics_by_genre (
    ...   genre VARCHAR,
    ...   performer VARCHAR,
        ...   year INT,
    ...   title VARCHAR,
Y           ...   PRIMARY KEY ((genre), performer, year, title)
        ... ) WITH CLUSTERING ORDER BY (performer ASC, year DESC, title ASC);
```

Insertamos datos que nos dan de ejemplo:

```
INSERT INTO musics_by_genre (genre, performer, year, title) VALUES ('Rock',
, 'Smells Like Teen Spirit');
```

Ahora vamos a parar los nodos 2 y 3, y veremos como podremos acceder a nuestra tabla recién creada. Porque la Consistencia está a nivel 0 si estuviera a nivel ALL, no podríamos:

```
\aleja\BBDD\cassandra-docker>docker stop DSE-6_node2
```

```
)\cassandra-docker>docker stop DSE-6_node3
```

Ahora nos metemos a musicDb y dentro cambiamos la consistencia:

```
Consistency level set to ALL.
cqlsh:musicdb> SELECT * FROM musics_by_genre WHERE genre='Rock';
NoHostAvailable:
```

Vemos como no devuelve nada.
Pero si la cambiamos a 0 y hacemos la consulta de nuevo:

```
 genre | performer | year | title
-------+-----------+------+--------------------------
  Rock |   Nirvana | 1991 | Smells Like Teen Spirit
```

y ya habríamos acabado.