

Creación de un cluster

Para realizar las pruebas con Cassandra crearemos un cluster de 3 nodos en local. Para ello utilizaremos Docker y Docker Compose. Los nodos del cluster serán los siguientes: cass1, cass2 y cass3. Los nodos cass1 y cass2 serán los nodos semilla del cluster. El nodo cass3 se unirá al cluster posteriormente.

Obteniendo la imagen de Cassandra

Usaremos la imagen oficial de Cassandra que se encuentra en Docker Hub:

https://hub.docker.com/_/cassandra.

Para obtener la imagen ejecutaremos el siguiente comando:

```
docker pull cassandra:latest
```

Creando el `docker-compose.yml`

```
version: "3.8"
networks:
  cassandra:
services:
  cass1:
    image: cassandra:latest
    container_name: cass1
    hostname: cass1
    mem_limit: 2g
    healthcheck:
      test: ["CMD", "cqlsh", "-e", "describe keyspaces"]
      interval: 5s
      timeout: 5s
      retries: 60
    networks:
      - cassandra
    ports:
      - "9042:9042"
    volumes:
      - ./data/cass1:/var/lib/cassandra # Para almacenar los datos de la base de
datos.
      - ./etc/cass1:/etc/cassandra # Para poder editar los archivos de
configuración.
      - ./scripts:/scripts # Para escribir y ejecutar scripts de CQL.

    environment: &environment
      CASSANDRA_SEEDS: "cass1,cass2"
      CASSANDRA_CLUSTER_NAME: SolarSystem
      CASSANDRA_DC: Mars
      CASSANDRA_RACK: West
      CASSANDRA_ENDPOINT_SNITCH: GossipingPropertyFileSnitch
      CASSANDRA_NUM_TOKENS: 128

  cass2:
    image: cassandra:latest
```

```

container_name: cass2
hostname: cass2
mem_limit: 2g
healthcheck:
  test: ["CMD", "cqlsh", "-e", "describe keyspaces"]
  interval: 5s
  timeout: 5s
  retries: 60
networks:
  - cassandra
ports:
  - "9043:9042"
volumes:
  - ./data/cass2:/var/lib/cassandra
  - ./etc/cass2:/etc/cassandra
  - ./scripts:/scripts
environment:
  <<: *environment
depends_on:
  cass1:
    condition: service_healthy
cass3:
  image: cassandra:latest
  container_name: cass3
  hostname: cass3
  mem_limit: 2g
  healthcheck:
    test: ["CMD", "cqlsh", "-e", "describe keyspaces"]
    interval: 5s
    timeout: 5s
    retries: 60
  networks:
    - cassandra
  ports:
    - "9045:9042"
  volumes:
    - ./data/cass3:/var/lib/cassandra
    - ./etc/cass3:/etc/cassandra
    - ./scripts:/scripts
  environment:
    <<: *environment
  depends_on:
    cass2:
      condition: service_healthy

```

Este archivo de configuración crea un *cluster* de 3 nodos de Cassandra en local.

Cada nodo tiene tres volúmenes asociados:

- `/data/cass1:/var/lib/cassandra`: Contiene los ficheros de la base de datos del nodo.
- `/etc/cass1:/etc/cassandra`: Contiene los archivos de configuración del nodo.
- `./scripts:/scripts`: Contendrá los scripts de CQL que vayamos escribiendo.

Antes de arrancar el *cluster* hemos de crear los directorios `data/cass1`, `data/cass2` y `data/cass3` y los directorios `etc/cass1`, `etc/cass2` y `etc/cass3`.

A continuación copiaremos los archivos de configuración de Cassandra en los directorios `etc/cass1`, `etc/cass2` y `etc/cass3`.

Para ello hemos primero de obtener los archivos de configuración de Cassandra. Para ello los copiaremos de un contenedor de Cassandra que se ejecutará temporalmente.

```
docker run --rm -d --name temp cassandra:latest
docker cp temp:/etc/cassandra .
docker stop temp
```

las opciones `--rm` y `-d` indican que el contenedor se elimine automáticamente al pararlo y que se ejecute en segundo plano.

Ahora tendremos un directorio `cassandra` con los archivos de configuración que hemos copiado de la máquina temporal.

Copiar los archivos de configuración de Cassandra

Copiaremos los archivos de configuración de Cassandra en los directorios `etc/cass1`, `etc/cass2` y `etc/cass3`:

```
cp -a ./cassandra ./etc/cass1/
cp -a ./cassandra ./etc/cass2/
cp -a ./cassandra ./etc/cass3/
```

La opción `-a` de `cp` indica que se copien los archivos de forma recursiva y que se conserven los permisos, propietarios y fechas de los archivos.

Si estamos en Windows hemos de hacer lo mismo copiando directamente el contenido del directorio `cassandra` a `etc/cass1` y los demás.

Todo esto lo hacemos para que podamos **modificar los archivos de configuración de Cassandra** de cada nodo de forma independiente editando los archivos de los directorios locales `etc/cass1`, `etc/cass2` y `etc/cass3`.

Nota respecto al volumen `scripts`: El volumen `scripts` se ha creado para poder escribir y ejecutar scripts de CQL desde dentro de los contenedores. Para ello hemos de copiar los scripts de CQL en el directorio `scripts` del directorio raíz del proyecto. Los scripts de CQL se ejecutarán desde dentro de los contenedores con el siguiente comando:

```
docker exec -it cass1 cqlsh -f /scripts/script.cql
```

No es necesario crear el directorio `/scripts` en los contenedores ya que se crea automáticamente al crear el volumen `scripts`.

Sí es necesario crear el directorio `scripts` en el host... *creo*.

Iniciar el *cluster*

Los nodos `cass1` y `cass2` serán los nodos designados como semilla del *cluster*.

Para iniciar el *cluster* ejecutaremos el siguiente comando:

```
docker-compose up -d
```

y el resultado debería ser el siguiente:

```
> docker-compose up -d
[+] Running 3/3
 ✓ Container cass1 Healthy
 0.0s
 ✓ Container cass2 Healthy
 0.5s
 ✓ Container cass3 Started
 0.7s
>
```

Para comprobar que los contenedores se han iniciado correctamente ejecutaremos el siguiente comando:

Revisar esto.

```
docker-compose ps
```

Abrir una consola de CQLSH

Para empezar a trabajar con Cassandra hemos de abrir una consola de CQLSH que nos permitirá ejecutar comandos CQL.

Para abrir una consola de CQLSH ejecutaremos el siguiente comando:

```
docker exec -it cass1 cqlsh
```

Con este último paso estaremos listos para empezar a trabajar con Cassandra.