

Cassandra

Cassandra

<https://cassandra.apache.org> és una de les bases de dades NoSQL més utilitzades.

1.- Crea un contenidor temporal per copiar el fitxer de configuració.

```
● box@docker:~/docker$ docker run --rm -d --name tmp \
> cassandra:4.1
70932be81978bd863b497e3ef3b0f08f01719468632a2f6a20c47a0898ecd1f3
○ box@docker:~/docker$
● box@docker:~/docker$ docker cp tmp:/etc/cassandra etc_cassandra
Successfully copied 164kB to /home/box/docker/etc_cassandra
● box@docker:~/docker$ docker stop tmp
tmp
○ box@docker:~/docker$
```

```
services:
  cassandra-1:
    image: cassandra:4.1
    environment: &environment
    CASSANDRA_SEEDS: cassandra-1, cassandra-2, cassandra-3
    volumes:
      - data-1:/var/lib/cassandra/data
  cassandra-2:
    image: cassandra:4.1
    environment: *environment
    volumes:
      - data-1:/var/lib/cassandra/data
  cassandra-3:
    image: cassandra:4.1
    environment: *environment
    volumes:
      - data-1:/var/lib/cassandra/data
volumes:
  data-1:
  data-2:
  data-3:
```

<https://blog.digitalis.io/containerized-cassandra-cluster-for-local-testing-60d24d70dcc4>

1.- Crear un volum cassandra :

```
box@docker:~$ docker volume create cassandra
cassandra
box@docker:~$
```

2.

no està associat a cap contenidor; és només un fragment de disc amb nom al qual es pot accedir mitjançant contenidors. El volum que acabeu de crear s'anomena **cass compartit**. En aquest cas, heu afegit una etiqueta al volum amb la clau **exemple** i el valor **cassandra**. Afegir metadades d'etiquetes als vostres volums us pot ajudar a organitzar i netejar els volums més tard. Utilitzareu aquest volum quan creeu un contenidor nou amb Cassandra:

```
Docker run -d \
  --volume cass-shared:/var/lib/cassandra/data \
  --nom cass1 \
  cassandra:2.2
```

Després que Docker estira el **cassandra:2.2** imatge de Docker Hub, crea un contenidor nou amb el **cass compartit** volum muntat a **/var/lib/cassandra/data**. A continuació, inicieu un contenidor des del **cassandra:2.2** imatge, però executeu una eina de client Cassandra (CQLSH) i connecteu-vos al vostre servidor en execució:

```
docker run -it --rm \
  --enllaç cass1:cass \
  cassandra:2.2 cqlsh cass
```

Ara podeu inspeccionar o modificar la vostra base de dades Cassandra des de l'ordre CQLSH

línia. Primer, cerqueu un espai de claus anomenat **docker_hello_world**:

```
selecciona *
system.schema_keyspaces
activat keyspace_name = 'docker_hello_world';
```

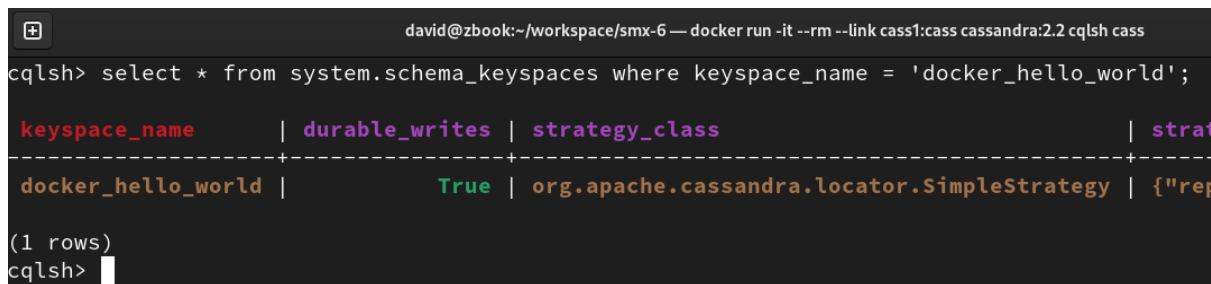
Cassandra hauria de tornar una llista buida. Això vol dir que la base de dades no ha estat modificada per l'exemple. A continuació, creeu aquest espai de tecles amb l'ordre següent:

```
crear un espai de claus docker_hello_world
amb replicació = {
  'classe' : "Estratègia senzilla",
  'factor_de_replica': 1
};
```

Ara que heu modificat la base de dades, hauríeu de poder tornar a fer la mateixa consulta per veure els resultats i verificar que s'han acceptat els vostres canvis. L'ordre següent és la mateixa que la que va executar anteriorment:

```
selecciona *
system.schema_keyspaces
activat keyspace_name = 'docker_hello_world';
```

Aquesta vegada, la Cassandra hauria de retornar una única entrada amb les propietats que va especificar quan va crear l'espai de tecles.



The screenshot shows a terminal window with the title 'david@zbook: ~/workspace/smx-6 — docker run -it --rm --link cass1:cass cassandra:2.2 cqlsh cass'. The prompt is 'cqlsh>'. The user enters the query 'select * from system.schema_keyspaces where keyspace_name = 'docker_hello_world';'. The result is displayed as a table with four columns: 'keyspace_name', 'durable_writes', 'strategy_class', and 'strategy_options'. The first row shows 'docker_hello_world', 'True', 'org.apache.cassandra.locator.SimpleStrategy', and '{\"replication_factor\": 1}'. Below the table, it says '(1 rows)' and the prompt 'cqlsh>' is shown again.

keyspace_name	durable_writes	strategy_class	strategy_options
docker_hello_world	True	org.apache.cassandra.locator.SimpleStrategy	{\"replication_factor\": 1}

Si esteu satisfet d'haver connectat i modificat el vostre node Cassandra, sortiu del programa CQLSH per aturar el contenidor del client:

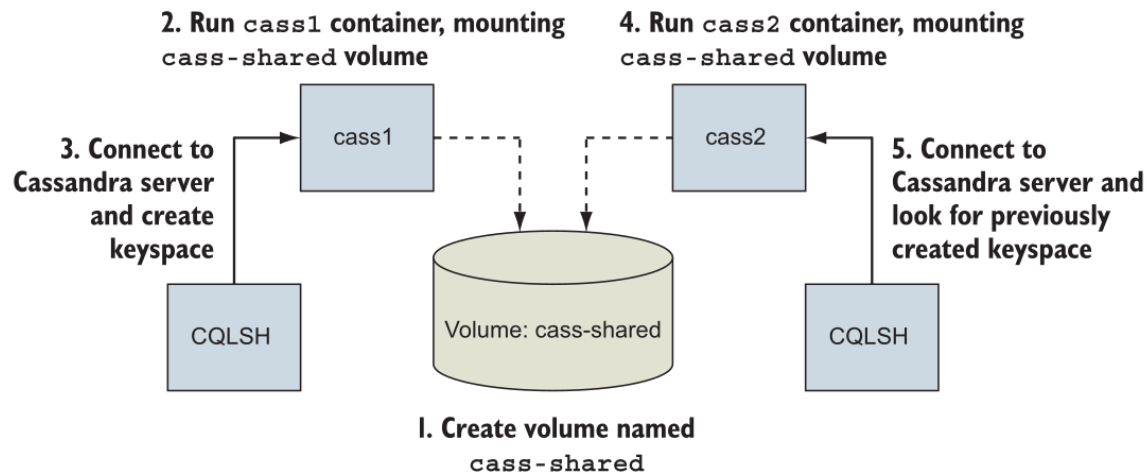
```
# Sortiu i atureu el contenidor actual
sortir
```

El contenidor del client es va crear amb el `--rm` bandera i es va eliminar automàticament quan es va aturar l'ordre. Continueu netejant la primera part d'aquest exemple aturant i eliminant el node Cassandra que heu creat:

```
docker stop cass1
docker rm -vf cass1
```

Tant el client Cassandra com el servidor que heu creat se suprimiran després d'executar aquestes ordres. Si les modificacions que heu fet persisteixen, l'únic lloc on podrien quedar-se és el **cas compartit** volum. Podeu provar-ho repetint aquests passos. Creeu un nou node Cassandra, adjunteu un client i consulteu l'espai de claus.

La següent figura il·lustra el sistema i el que haureu construït.



Els passos clau per crear i recuperar dades van persistir a un volum amb Cassandra

Les tres ordres següents proveen la recuperació de les dades:

```
Docker run -d \
  --volume cass-shared:/var/lib/cassandra/data \
  --nom cass2 \
  cassandra:2.2
```

```
docker run -it --rm \
  --enllaç cass2:cass \
  cassandra:2.2 \
  cqlsh cass
```

```
selecciona *
system.schema_keyspaces
activat keyspace_name = 'docker_hello_world';
```

L'última ordre d'aquest conjunt retorna una única entrada i coincideix amb l'espai de tecles que heu creat al contenidor anterior. Això confirma les afirmacions anteriors i demostra com es poden utilitzar els volums per crear sistemes duradors. Abans de continuar, sortiu del programa CQLSH i netegeu el vostre espai de treball. Assegureu-vos de treure aquest contenidor de volum també:

```
sortir  
docker rm -vf cass2 cass-shared
```

Aquest exemple mostra una manera d'utilitzar els volums sense entrar en com funcionen, els patrons en ús o com gestionar el cicle de vida dels volums. La resta d'aquest document aprofundeix en cada faceta dels volums, començant pels tipus disponibles.