

# Distribuidas

A.D.G.

Hacer un tutorial breve de cómo montar un cluster mysql (Máximo 3 hojas).

Podéis seguir los tutoriales del aula para hacerlo con docker, también podéis usar un docker compose como este (modificando lo que necesitéis)

<https://github.com/mlucasdasilva/cluster-mysql/blob/master/docker-compose.yml>

Debe tener:

1 nodo de gestión

2 nodos de datos

2/3 nodos mysql

Si solo tiene un nodo mysql la nota de la práctica será máximo un 8,5 dependiendo de la calidad de la explicación.

Una vez funcionando se debe mostrar un show del nodo de gestión:

ndb\_mgm> show

Y por otro lado se debe mostrar alguna sentencia sql ejecutada en la base de datos

---

Usaré MobaXterm por comodidad mediante SSH, y teniendo ya instalado docker en el Ubuntu:

## Creamos la red docker:

Es necesaria para el cluster MySQL. Lo creamos con:

```
root@102:~# docker network create cluster --subnet=192.168.4.0/24
7b2b0fafdb64b55db8698977af1d4feb305c79fccde1f28416ee5de0919f0d81
```

## Obtenemos el repositorio mysql de docker.

Para ello, vamos a clonar desde este repositorio, luego nos situamos dentro y nos movemos a la rama del mysql-cluster:

```
root@102:~# git clone https://github.com/mysql/mysql-docker.git
```

```
root@102:~# cd mysql-docker/
```

```
root@102:~/mysql-docker# git checkout -b mysql-cluster
```

Abrimos el archivo:

```
root@102:~/mysql-docker# nano mysql-cluster/8.0/cnf/mysql-cluster.cnf
```

Y lo modificaremos de la siguiente manera, para tener 1 nodo de gestión, 2 de datos y 3 de MySQL:

```
GNU nano 6.2 mysql-cluster/8.0/cnf/mysql-cluster.cnf
[ndbd default]
NoOfReplicas=2
DataMemory=80M
IndexMemory=18M

[ndb_mgmd]
NodeId=1
hostname=192.168.4.2
datadir=/var/lib/mysql

[ndbd]
NodeId=2
hostname=192.168.4.3
datadir=/var/lib/mysql

[ndbd]
NodeId=3
hostname=192.168.4.4
datadir=/var/lib/mysql

[mysqld]
NodeId=4
hostname=192.168.4.10

[mysqld]
NodeId=5
hostname=192.168.4.11

[mysqld]
NodeId=6
hostname=192.168.4.12
```

Abrimos este otro archivo:

```
root@102:~/mysql-docker# nano mysql-cluster/8.0/cnf/my.cnf
GNU nano 6.2 mysql-cluster/8.0/cnf/my.cnf *
[mysqld]
ndbcluster
ndb-connectstring=192.168.4.2
user=mysql

[mysql_cluster]
ndb-connectstring=192.168.4.2
```

Creamos la imagen docker:

```
root@102:~/mysql-docker# docker build -t mysql-cluster mysql-cluster/8.0
Successfully built 28f636fd5a83
Successfully tagged mysql-cluster:latest
```

Creamos el nodo para manejar:

```
root@102:~/mysql-docker# docker run -d --net=cluster --name=management1 --ip=192.168.4.2 mysql-cluster ndb_mgmd
```

Creamos los nodos de datos:

```
root@102:~/mysql-docker# docker run -d --net=cluster --name=ndb1 --ip=192.168.4.3 mysql-cluster ndbd
bdb74bb493782b183e190c40534b972169c7f617574a4dcf4f87a90add446451
root@102:~/mysql-docker# docker run -d --net=cluster --name=ndb2 --ip=192.168.4.4 mysql-cluster ndbd
ae1e822b4436348f62c48df211150a6f4b1e5f0178808f65926f29a00c3195f3
```

Creamos los nodos mysql e iniciamos sesión interactiva en el nodo de gestión del cluster en un cluster de MySQL NDB:

```
root@102:~/mysql-docker# docker run -d --net=cluster --name=mysql1 --ip=192.168.4.10 -e MYSQL_RANDOM_ROOT_PASSWORD=true mysql-cluster mysqld
d0a8e6ae8ffc6fbb900cd7666131be4403177fb42f9856e8f3b7ff9babbf9258
root@102:~/mysql-docker# docker run -d --net=cluster --name=mysql2 --ip=192.168.4.11 -e MYSQL_RANDOM_ROOT_PASSWORD=true mysql-cluster mysqld
8c24297626ad084d44cba6d5f0fd1bbc28e2f375d621bd7157b095d8df2c60f7
root@102:~/mysql-docker# docker run -d --net=cluster --name=mysql3 --ip=192.168.4.12 -e MYSQL_RANDOM_ROOT_PASSWORD=true mysql-cluster mysqld
517285a6cb64666c29d01e93dde9a999cc3a1462439a91f1fef7c13ea5219ac2
root@102:~/mysql-docker# docker run -it --net=cluster mysql-cluster ndb_mgmd
[Entrypoint] MySQL Docker Image 8.0.32-1.2.11-cluster
[Entrypoint] Starting ndb_mgm
-- NDB Cluster -- Management Client --
ndb_mgm>
```

Aquí podemos ver lo que tenemos, tal cual lo pide el ejercicio, incluyendo 3 nodos MySQL:

```
ndb_mgm> show
Connected to Management Server at: 192.168.4.2:1186
Cluster Configuration
-----
[ndbd(NDB)] 2 node(s)
id=2 @192.168.4.3 (mysql-8.0.32 ndb-8.0.32, Nodegroup: 0, *)
id=3 @192.168.4.4 (mysql-8.0.32 ndb-8.0.32, Nodegroup: 0)

[ndb_mgmd(MGM)] 1 node(s)
id=1 @192.168.4.2 (mysql-8.0.32 ndb-8.0.32)

[mysqld(API)] 3 node(s)
id=4 @192.168.4.10 (mysql-8.0.32 ndb-8.0.32)
id=5 @192.168.4.11 (mysql-8.0.32 ndb-8.0.32)
id=6 @192.168.4.12 (mysql-8.0.32 ndb-8.0.32)

ndb_mgm>
```

Obtenemos las contraseñas para los nodos MySQL que se crearon por defecto:

```
root@102:~/# docker logs mysql1 2>&1 | grep PASSWORD
[Entrypoint] GENERATED ROOT PASSWORD: h1_J8Xa4#8WMsqG#W=,=?,j17XaH7z7S
root@102:~/# docker logs mysql2 2>&1 | grep PASSWORD
[Entrypoint] GENERATED ROOT PASSWORD: N4A_l;e^+2r5ki04&06QjD_24_f&wjIM
root@102:~/# docker logs mysql3 2>&1 | grep PASSWORD
[Entrypoint] GENERATED ROOT PASSWORD: XS3j3T_4D_ob;;ke3a=99&MEB#wf#80q
```

h1\_J8Xa4#8WMsqG#W=,=?,j17XaH7z7S  
N4A\_l;e^+2r5ki04&06QjD\_24\_f&wjIM  
XS3j3T\_4D\_ob;;ke3a=99&MEB#wf#80q



Nos logueamos con el primer nodo MySQL por ejemplo:

```
root@102:~# docker exec -it mysql1 mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 60
Server version: 8.0.32-cluster MySQL Cluster Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Creamos una base de datos y alguna tabla con algún que otro campo:

```
mysql> create schema andreiprueba_db;
Query OK, 1 row affected (0.21 sec)

mysql> use andreiprueba_db
Database changed
mysql> CREATE TABLE practicas (idalumno INT NOT NULL, nombreempresa VARCHAR(255), tipotrabajo VARCHAR(255), PRIMARY KEY (idalumno));
Query OK, 0 rows affected (0.09 sec)

mysql> INSERT INTO practicas (idalumno, nombreempresa, tipotrabajo) VALUES (102, 'Proximamente', 'IT HelpDesk');
Query OK, 1 row affected (0.05 sec)

mysql> SHOW COLUMNS FROM practicas;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idalumno   | int           | NO   | PRI | NULL    |       |
| nombreempresa | varchar(255) | YES  |     | NULL    |       |
| tipotrabajo | varchar(255) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> SELECT * FROM practicas;
+-----+-----+-----+
| idalumno | nombreempresa | tipotrabajo |
+-----+-----+-----+
| 102      | Proximamente  | IT HelpDesk |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Podemos ver el nombre de la base de datos desde otro nodo MySQL:

```
mysql> ^DBye
root@102:~# docker exec -it mysql2 mysql -uroot -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 156
Server version: 8.0.32-cluster MySQL Cluster Community Server - GPL

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| andreiprueba_db |
| information_schema |
| mysql |
| ndbinfo |
| performance_schema |
| sys |
+-----+
6 rows in set (0.01 sec)

mysql> use andreiprueba_db;
Database changed
mysql>
```