



Actividad 2 – Comandos de replicación



Reybert Andrés Peñuela Sepúlveda

Corporación universitaria Iberoamericana

Facultad de Ingeniería de software

Bases de datos avanzadas

Profesor William Ruiz

Cali, Colombia

03 de abril del 2023

Comandos o directivas paso a paso para replicar las bases de datos del evento deportivo en mínimo 3 nodos

En este caso vamos a realizar todos los ejemplos con contenedores de Docker, esto debido a la comodidad y sencilla que es la implementación de todo tipo de herramientas en Docker. Para esto debemos tener presente que debemos tener previamente instalado Docker Desktop en el caso de tener un sistema operativo Windows, y debemos tener instalado WSL (Distribución de Linux) con sistema operativo Ubuntu. Esto para poder usar Docker Desktop en Windows.

Cabe aclarar que en toda la explicación dentro de este documento usaremos PowerShell de Windows.

Paso 1: Descarga de la imagen de MongoDB en Docker

Ejecuta el comando `docker pull mongo`

La imagen debe descargarse y el resultado debería ser más o menos el siguiente:

```
Windows PowerShell
PS C:\Users\Reybert> docker pull mongo
Using default tag: latest
latest: Pulling from library/mongo
74ac377868f8: Pull complete
7bf92c33f8cb: Pull complete
3ad057cae032: Pull complete
94195d632f1b: Pull complete
df4a989974eb: Pull complete
a94fc3cc2ce1: Pull complete
e31c2ebced96: Pull complete
7af2dc62bb4b: Pull complete
ec5239bce583: Pull complete
Digest: sha256:cc4522f3f5c0d3435046eb51b1d8a633d8e24d8e661b6ba127a98e5519d11bde
Status: Downloaded newer image for mongo:latest
docker.io/library/mongo:latest
PS C:\Users\Reybert> |
```

Paso 2: Crear una red para el conjunto de réplicas

Crea una red de Docker para que los contenedores del conjunto de réplicas puedan comunicarse entre sí. Ejecutando el comando a continuación

`docker network create mongo-replicaset`

Como resultado este debe retornar el id de la red creada

```
PS C:\Users\Reybert> docker network create mongo-replicaset
7a2cc9ad9fa18846b43572690c2aa416a77d736481f3eece618bb029964ae774
```

Paso 3: Ejecutar contenedores de MongoDB

Ejecuta tres contenedores de MongoDB en la red creada en el paso anterior. Cada contenedor actuará como un miembro del conjunto de réplicas. Los comandos para ejecutar cada contenedor son los siguientes.

- Comando para crear contenedor **mongo1**:

```
docker run -d --name mongo1 --net mongo-replicaset -p 27017:27017 -e MONGO_INITDB_ROOT_USERNAME=admin -e MONGO_INITDB_ROOT_PASSWORD=adminpass mongo --replSet rs0
```

- Comando para crear contenedor **mongo2**:

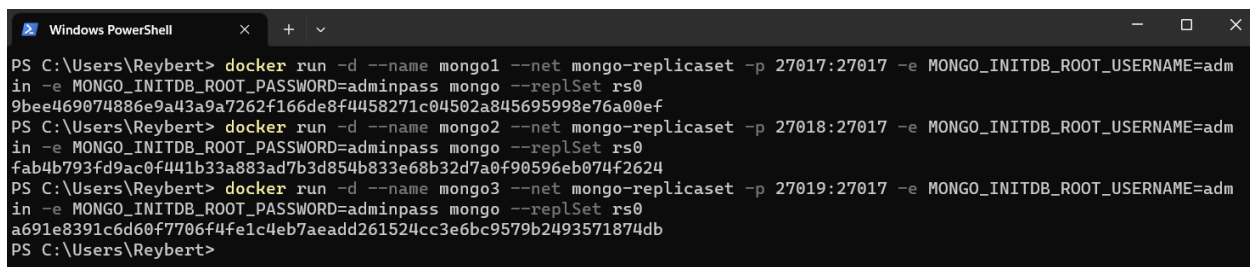
```
docker run -d --name mongo2 --net mongo-replicaset -p 27018:27017 -e MONGO_INITDB_ROOT_USERNAME=admin -e MONGO_INITDB_ROOT_PASSWORD=adminpass mongo --replSet rs0
```

- Comando para crear contenedor **mongo3**:

```
docker run -d --name mongo3 --net mongo-replicaset -p 27019:27017 -e MONGO_INITDB_ROOT_USERNAME=admin -e MONGO_INITDB_ROOT_PASSWORD=adminpass mongo --replSet rs0
```

Esto iniciará tres contenedores llamados "mongo1", "mongo2" y "mongo3", cada uno ejecutando MongoDB en diferentes puertos (27017, 27018 y 27019) y formando parte del conjunto de réplicas llamado "rs0". Los contenedores están configurados con un usuario "admin" y una contraseña "adminpass".

Los resultados obtenidos son los siguientes.



```
Windows PowerShell
PS C:\Users\Reybert> docker run -d --name mongo1 --net mongo-replicaset -p 27017:27017 -e MONGO_INITDB_ROOT_USERNAME=admin -e MONGO_INITDB_ROOT_PASSWORD=adminpass mongo --replSet rs0
9bee469074886e9a43a9a7262f166de8f4458271c04502a845695998e76a00ef
PS C:\Users\Reybert> docker run -d --name mongo2 --net mongo-replicaset -p 27018:27017 -e MONGO_INITDB_ROOT_USERNAME=admin -e MONGO_INITDB_ROOT_PASSWORD=adminpass mongo --replSet rs0
fab4b793fd9ac0f441b33a883ad7b3d854b833e68b32d7a0f90596eb074f2624
PS C:\Users\Reybert> docker run -d --name mongo3 --net mongo-replicaset -p 27019:27017 -e MONGO_INITDB_ROOT_USERNAME=admin -e MONGO_INITDB_ROOT_PASSWORD=adminpass mongo --replSet rs0
a691e8391c6d60f7706f4fe1c4eb7aeadd261524cc3e6bc9579b2493571874db
PS C:\Users\Reybert>
```

Paso 4: Iniciar el conjunto de réplicas

Conéctate al primer contenedor de MongoDB utilizando el siguiente comando:

```
docker exec -it mongo1 mongo -u admin -p adminpass --authenticationDatabase admin
```

Es probable que se retorne el mensaje a continuación

```
Error response from daemon: Container
9bee469074886e9a43a9a7262f166de8f4458271c04502a845695998e76a00ef is not running
```

El mensaje de error indica que el contenedor "mongo1" no se está ejecutando. Asegúrate de que hayas creado y ejecutado el contenedor "mongo1" antes de intentar conectarte al shell de MongoDB.

Para verificar si el contenedor "mongo1" se está ejecutando, ejecuta el siguiente comando en PowerShell:

```
docker ps
```

Si el contenedor "mongo1" no aparece en la lista, verifica si está presente pero detenido, ejecutando:

```
docker ps -a
```

Si el contenedor "mongo1" está presente pero detenido, puedes iniciarlo utilizando el siguiente comando:

```
docker start mongo1
```

Una vez que el contenedor esté en ejecución, intenta conectarte al shell de MongoDB nuevamente con el comando:

```
docker exec -it mongo1 mongo -u admin -p adminpass --authenticationDatabase admin
```

Una vez dentro del shell de MongoDB, inicia el conjunto de réplicas utilizando el siguiente comando

```
rs.initiate()
```

Esto creará un conjunto de réplicas con un solo miembro (mongo1).

Paso 5: Agregar miembros al conjunto de réplicas

Aún en el shell de MongoDB, agrega los otros dos contenedores como miembros del conjunto de réplicas utilizando el siguiente comando

```
rs.add("mongo2:27017")
rs.add("mongo3:27017")
```

Esto agregará "mongo2" y "mongo3" como miembros secundarios del conjunto de réplicas.

Paso 6: Verificar el estado del conjunto de réplicas

Comprueba el estado del conjunto de réplicas utilizando el comando

```
rs.status()
```

Esto mostrará información sobre el conjunto de réplicas, incluyendo los miembros y sus roles (primario o secundario).

Ahora tenemos un conjunto de réplicas de MongoDB funcionando en un único servidor utilizando Docker.