02 Deploy a replica set

Directorios de datos para los tres servidores

Linux

mkdir data/rs{1,2,3}

Windows

md c:\data\server1
md c:\data\server2
md c:\data\server2

Levantar los 3 servidores

mongod --replSet clusterGetafe --dbpath data/server1 --port 27017
mongod --replSet clusterGetafe --dbpath data/server2 --port 27018
mongod --replSet clusterGetafe --dbpath data/server3 --port 27019

Configuración e inicialización del Replica set

Nos conectamos, por ejemplo, al primer servidor:

mongo --host localhost --port 27017

Creamos un objeto de configuración

```
> rsconfig = {
... _id: "clusterGetafe",
... members: [
... {_id: 0, host: "localhost:27017"},
... {_id: 1, host: "localhost:27018"},
... {_id: 2, host: "localhost:27019"}
... ]
... }


> rs.initiate(rsconfig)
```

```
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1580405618, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1580405618, 1)
}
clusterGetafe:SECONDARY>   // Pasará a estado primario
```

Comprobamos el estado del cluster con

```
rs.status()
```

Otro comando es

```
isMaster()
```

Operación de escritura en primario

Si el primario fuera otro servidor diferente nos conectamos a el y creamos una operación de escritura:

```
clusterGetafe:PRIMARY> use getafeTest
switched to db getafeTest

clusterGetafe:PRIMARY> for (i= 0; i < 1000; i++) { db.foo.insert({contador: i}) }
WriteResult({ "nInserted" : 1 })
```

Nos conectamos ahora en otra shell a un servidor secundario

```
clusterGetafe:SECONDARY> show dbs
2020-01-30T18:45:37.686+0100 E  QUERY    [js] uncaught exception: Error:
listDatabases failed:{
```

Nos devuelve error porque por defecto los servidores secundarios no permiten operaciones de lectura. Excepcionalmente podemos darle permiso de lectura para comprobar la réplica:

```
clusterGetafe:SECONDARY> db.setSlaveOk()
clusterGetafe:SECONDARY> use getafeTest
switched to db getafeTest

clusterGetafe:SECONDARY> db.setSlaveOk()
clusterGetafe:SECONDARY> db.foo.find()
{ "_id" : ObjectId("5e331552e1f3ba9744f8b243"), "contador" : 3 }
...
```

Si intentamos escribir en un secundario, provoca error

Automatic Failover

Apagamos el servidor primario y lanzamos en el secundario:

rs.status()

```
...
        "members" : [
                {
                        "_id" : 0,
                        "name" : "localhost:27017",
                        "health" : 0,
                        "state" : 8,
                        "stateStr" : "(not reachable/healthy)",
                        "uptime" : 0,
                        ...
                },
                {
                        "_id" : 1,
                        "name" : "localhost:27018",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",

...
```

Tenemos nuevo primario.

Operación de escritura en el nuevo primario

clusterGetafe:PRIMARY> db.libros.insert({autor: "Cervantes", titulo: "El Quijote"})
WriteResult({ "nInserted" : 1 })

Iniciamos el otro secundario para comprobar

mongo --port 27019

clusterGetafe:SECONDARY> use getafeTest
switched to db getafeTest
clusterGetafe:SECONDARY> db.setSlaveOk()
clusterGetafe:SECONDARY> show collections
foo
libros

Reincorporación del servidor caido

Levantamos el servidor caído (con el mismo comando que se inicializó la primera vez y que incluye los parámetros del cluster)

Comprobamos como en la consola hace referencia a la actualización de datos.

Reconectamos la shell a este servidor y:

clusterGetafe:SECONDARY> use getafeTest
switched to db getafeTest
clusterGetafe:SECONDARY> show collections
2020-01-30T19:04:27.115+0100 E QUERY   [js] uncaught exception: Error:
listCollections failed: {
	"operationTime" : Timestamp(1580407462, 1),
	"ok" : 0,
	"errmsg" : "not master and slaveOk=false",
	"code" : 13435,
	"codeName" : "NotMasterNoSlaveOk",
	"$clusterTime" : {
		"clusterTime" : Timestamp(1580407462, 1),
		"signature" : {
			"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
			"keyId" : NumberLong(0)
		}
	}

} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
DB.prototype._getCollectionInfosCommand@src/mongo/shell/db.js:834:15
DB.prototype.getCollectionInfos@src/mongo/shell/db.js:882:16
shellHelper.show@src/mongo/shell/utils.js:893:9
shellHelper@src/mongo/shell/utils.js:790:15
@(shellhelp2):1:1
clusterGetafe:SECONDARY> db.setSlaveOk()
clusterGetafe:SECONDARY> show collections
foo
libros

Aunque fuera primario en su momento ahora es secundario y debemos asignarle permisos de lectura.