Conceptos y Comandos básicos del particionamiento en bases de datos NoSQL

&

Pruebas de particionamiento de bases de datos NoSQL

Asignatura

Bases de Datos Avanzadas

Presenta

Jhony Steven Villareal Noguera

Docente

Jorge Castañeda

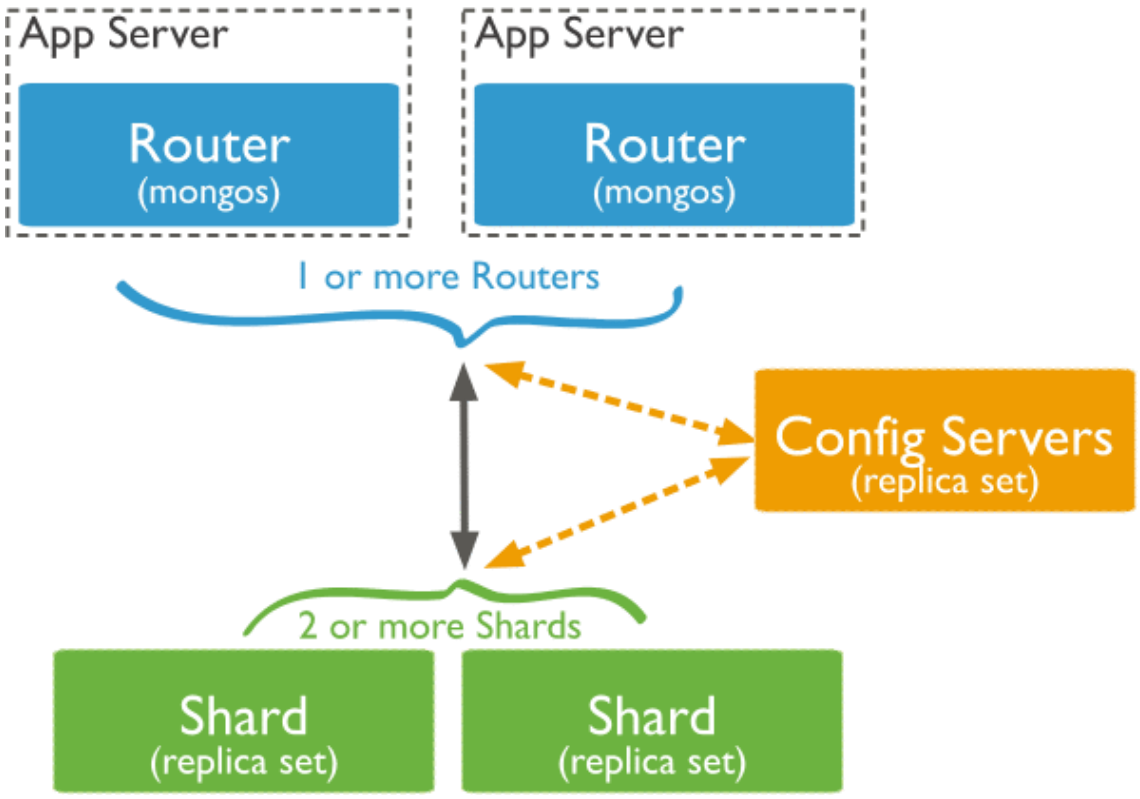
Ipiales – Nariño

2024

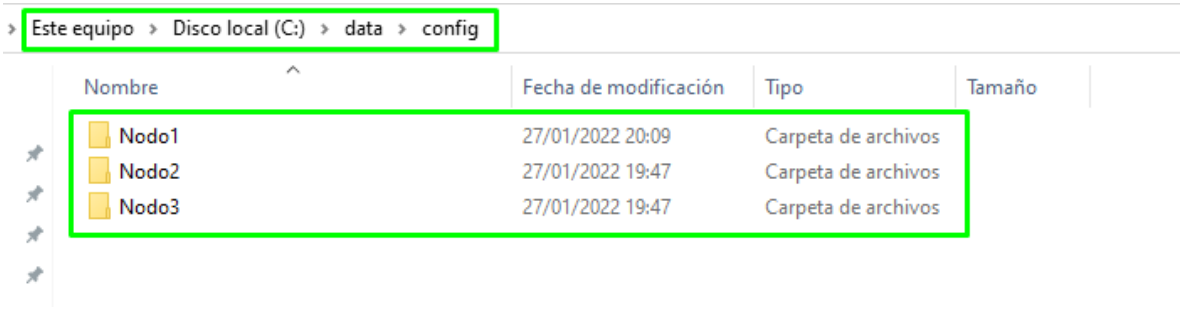
Requerimientos No Funcionales

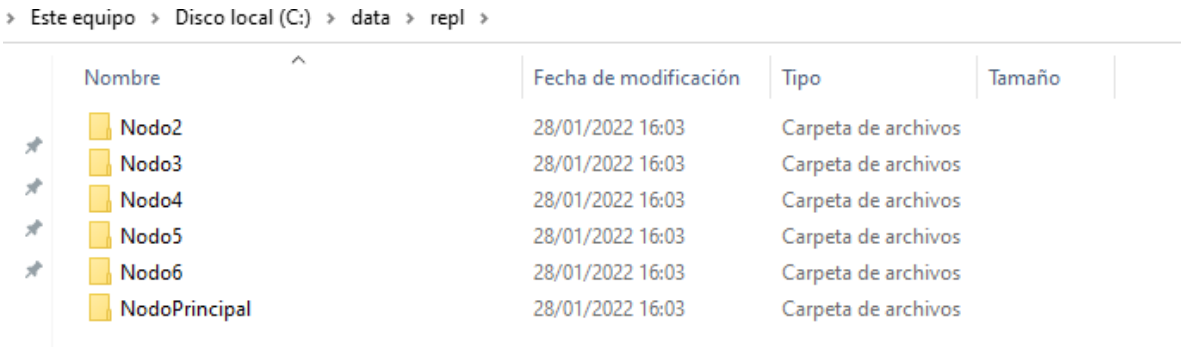
- ✓ El sistema de particionamiento debe poseer como mínimo 2 bases de replicación.
- ✓ Cada una de las réplicas asignadas, deberán poseer como mínimo 3 nodos y con diferentes puertos configurados, para su conexión.
- ✓ Todos los nodos configurados poseerán el mismo hostname o ip, para la comunicación entre estos.
- ✓ Se tendrá la necesidad de ocupar un servidor de configuración, que contenga así mismo 3 nodos, que ocupen los datos, para la transmisión de estos.
- ✓ Se debe ocupar de un enrutador para el manejo sobre los nodos de configuración.
- ✓ Se deben fragmentar la base de datos, una de las colecciones y las réplicas configuradas.

Diagrama del particionamiento usado en forma horizontal



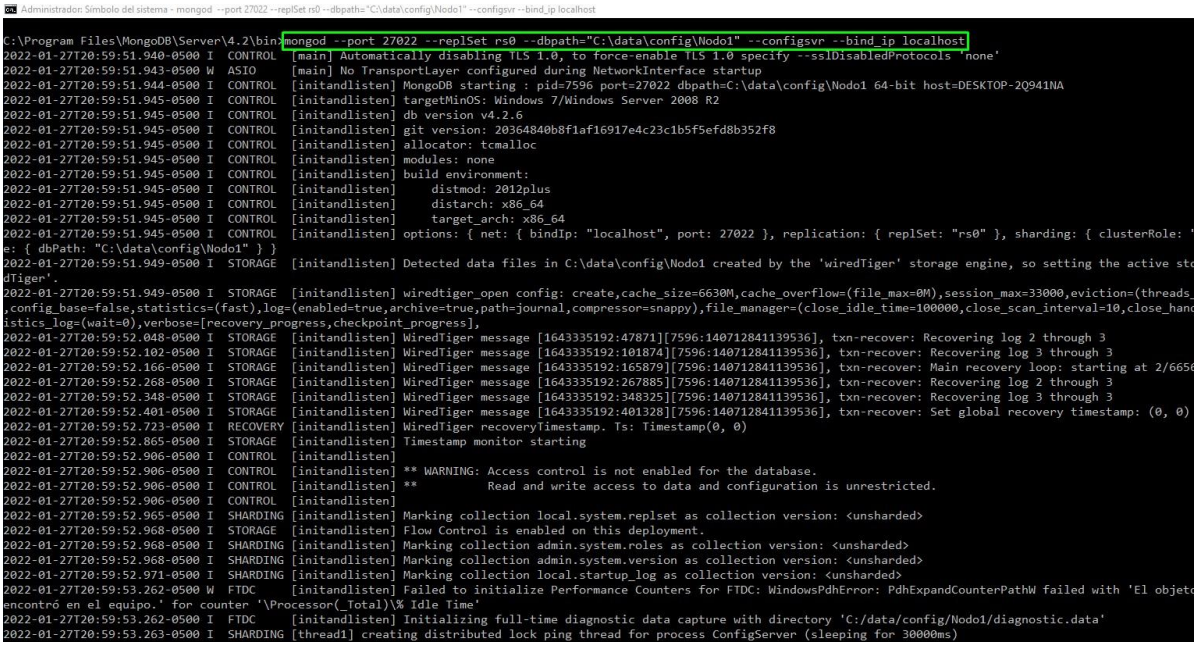
- Para inicar necesitamos que cada uno de los servidores de configuración le sean creados directorios de almacenamiento o carpetas, esto gracias al modo de que cada uno tendra el objetivo de guardar y organizar los datos.



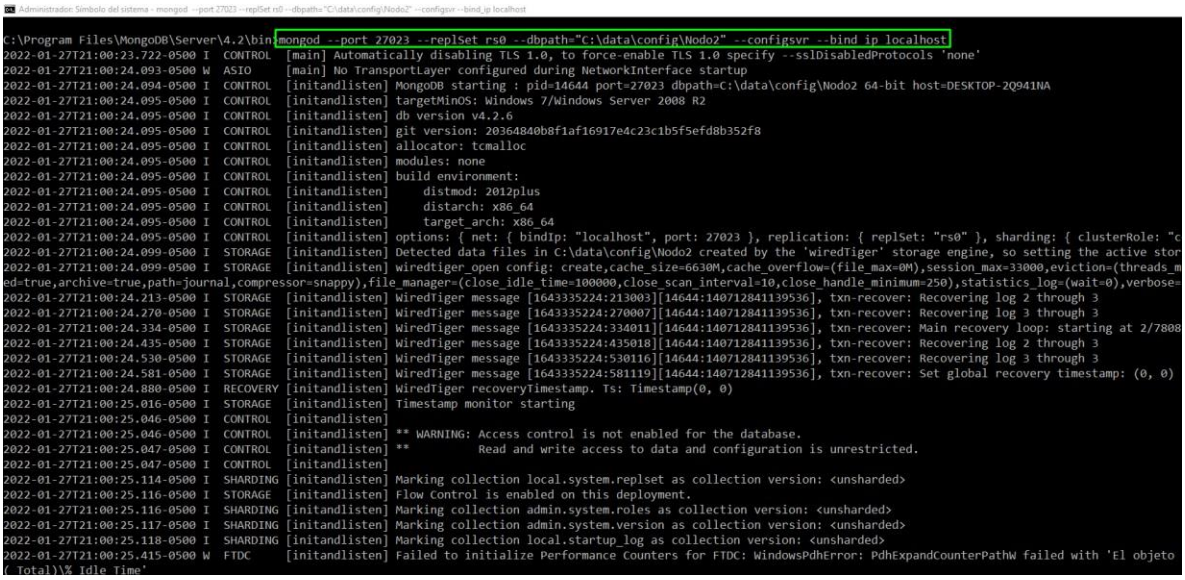


- Ahora procederemos a activar cada servidor de configuración. Para que funcione en una única máquina, hay que darlo de alta en la misma IP de la máquina en la que se ha dado de alta el replica set esto gracias al comando - -bind_ip:

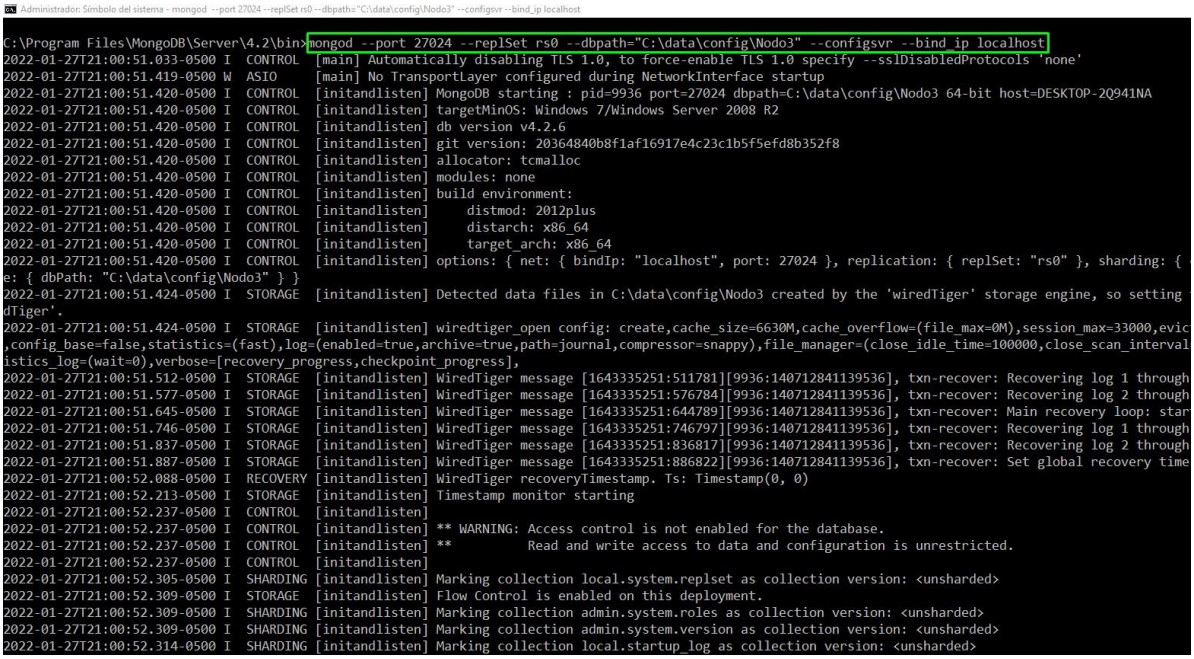
Activamos el servidor 1 con el puerto 27022



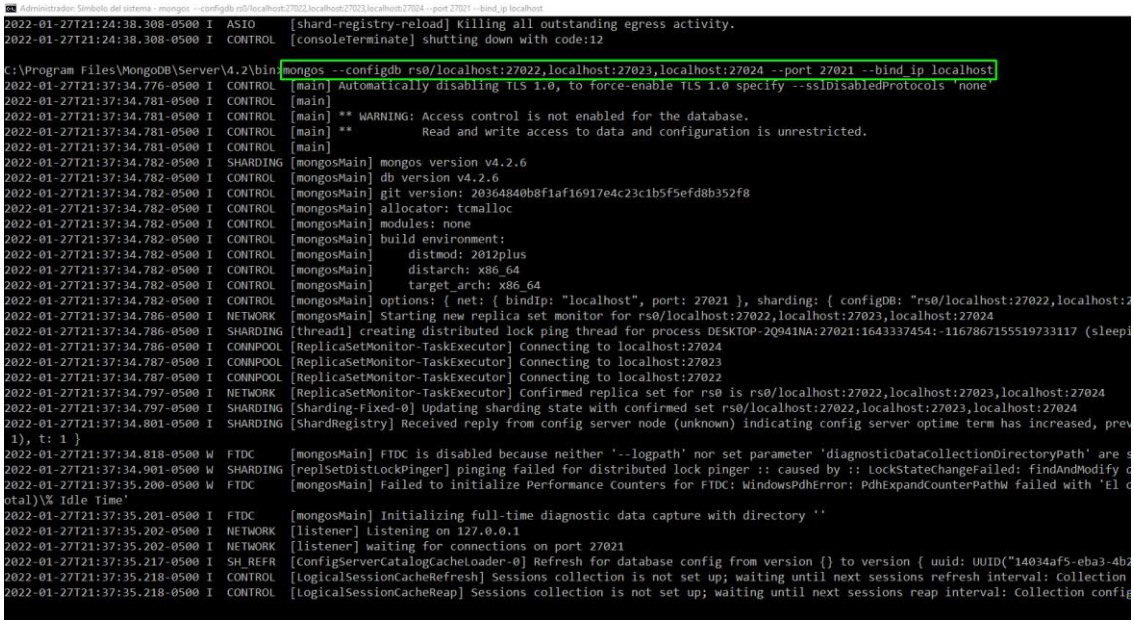
Activamos el servidor 2 con el puerto 27023



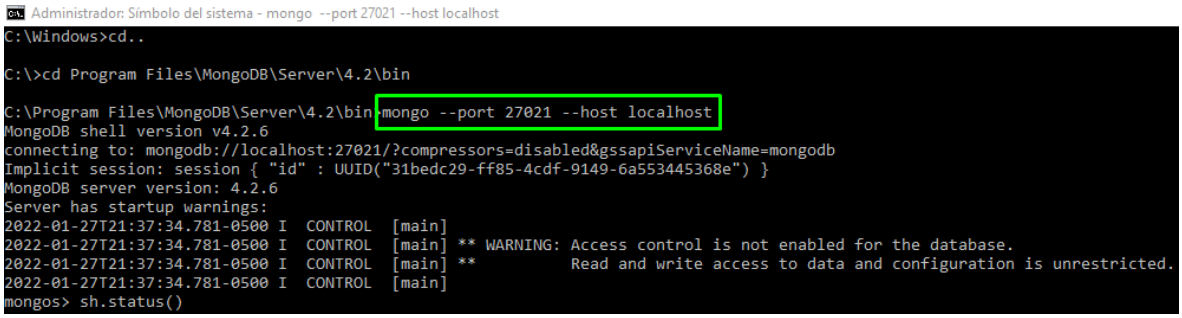
Activamos el servidor 3 con el puerto 27024



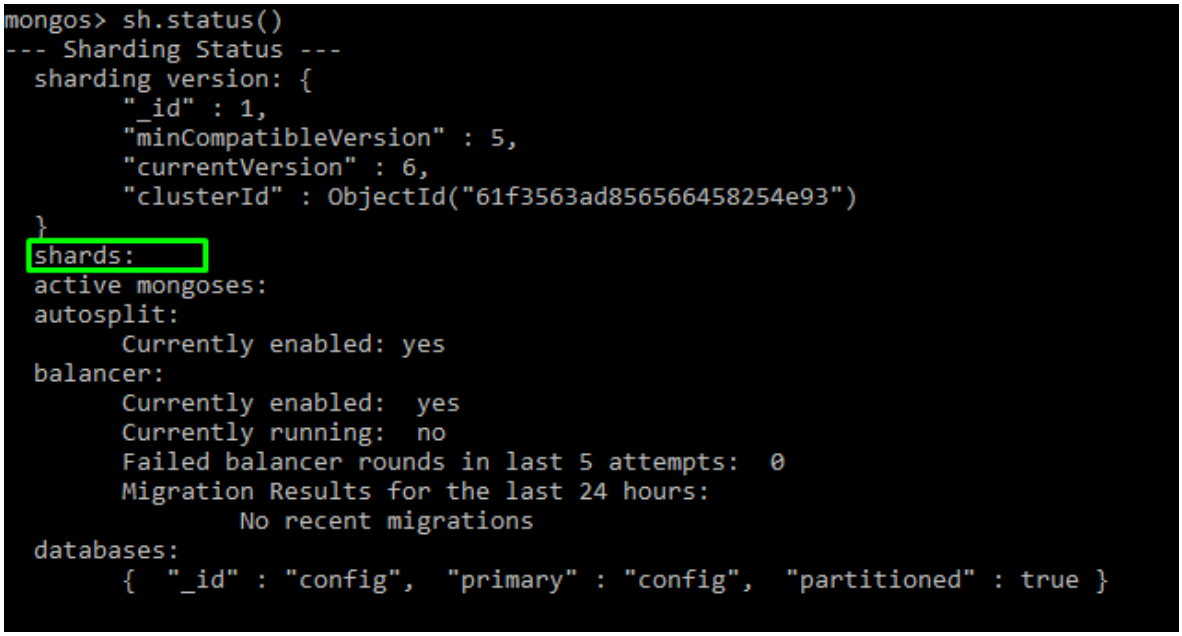
- Ahora sera importante continuar con la activación de nuestro enrutador sobbre la misma dirección Ip, anteriormente mencionada, y adicionalmente asignando los servidores sobre este.



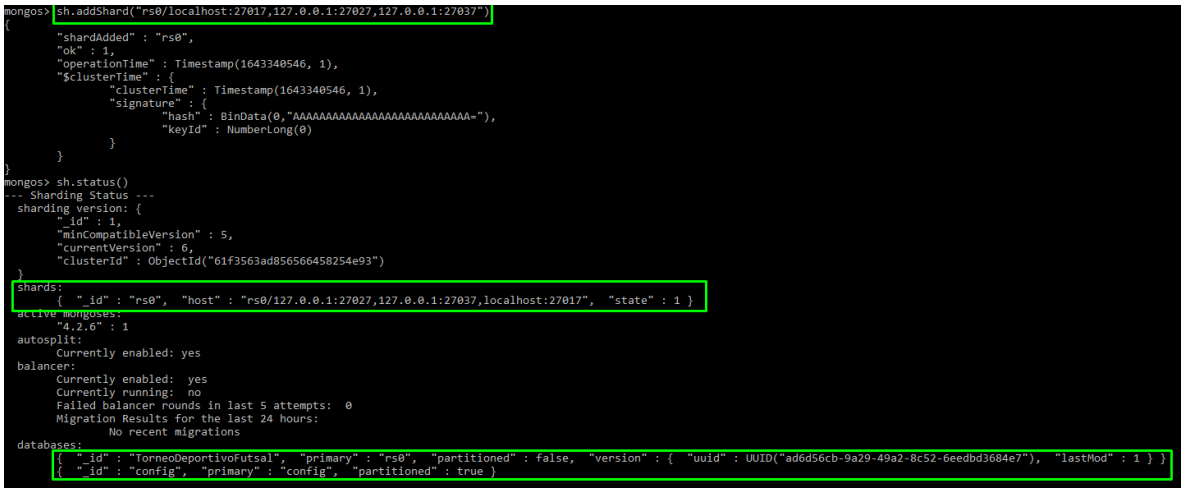
- Ahora necesitaremos añadir el replica set inicial que se ha configurado como un shard. Se se ejecuta el puerto del enrutador para dar de alta el replica set como un shard del clúster, siendo asi la validación exitosa de la activación del enrutador y funcionamiento de este.



Ahora debemos vertificar el estado inicial de nuestro shard, ya que encontraremos que se encuentra vacio sin ningun replica set, para luego poder añadirselo.

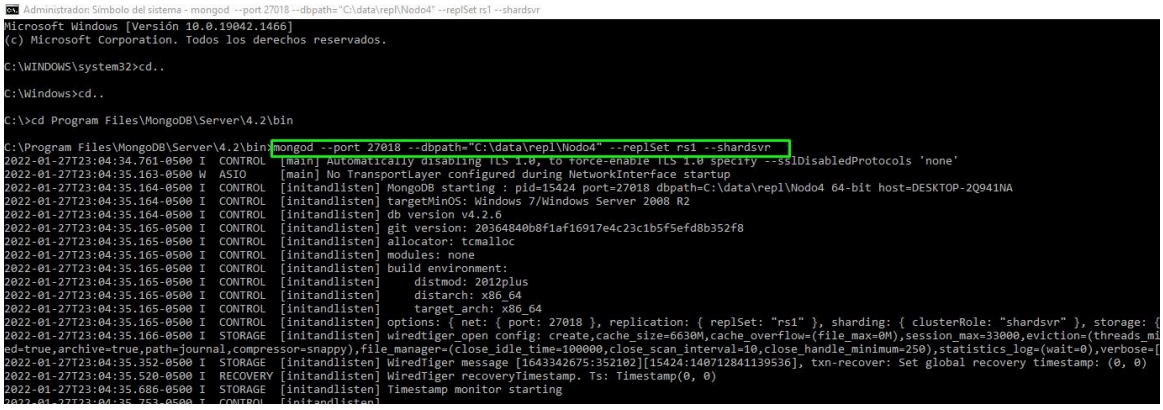


Para cuando le hayamos añadido el replica set sobre nuestro shard, tendremos:



Asi mismo ocuparemos el uso de nuestra base de datos, sobre el shard ya configurado, en base de la replica añadida.

- Desde este punto ocupamos ya dentro de nuestra estructura de particionamiento, los nodos de configuración y los nodos de una replica set ya definida, ahora necesitamos crear un nuevo replica set para añadir a nuestro shard y estariamos manejando 2 shard para nuestra base de datos.



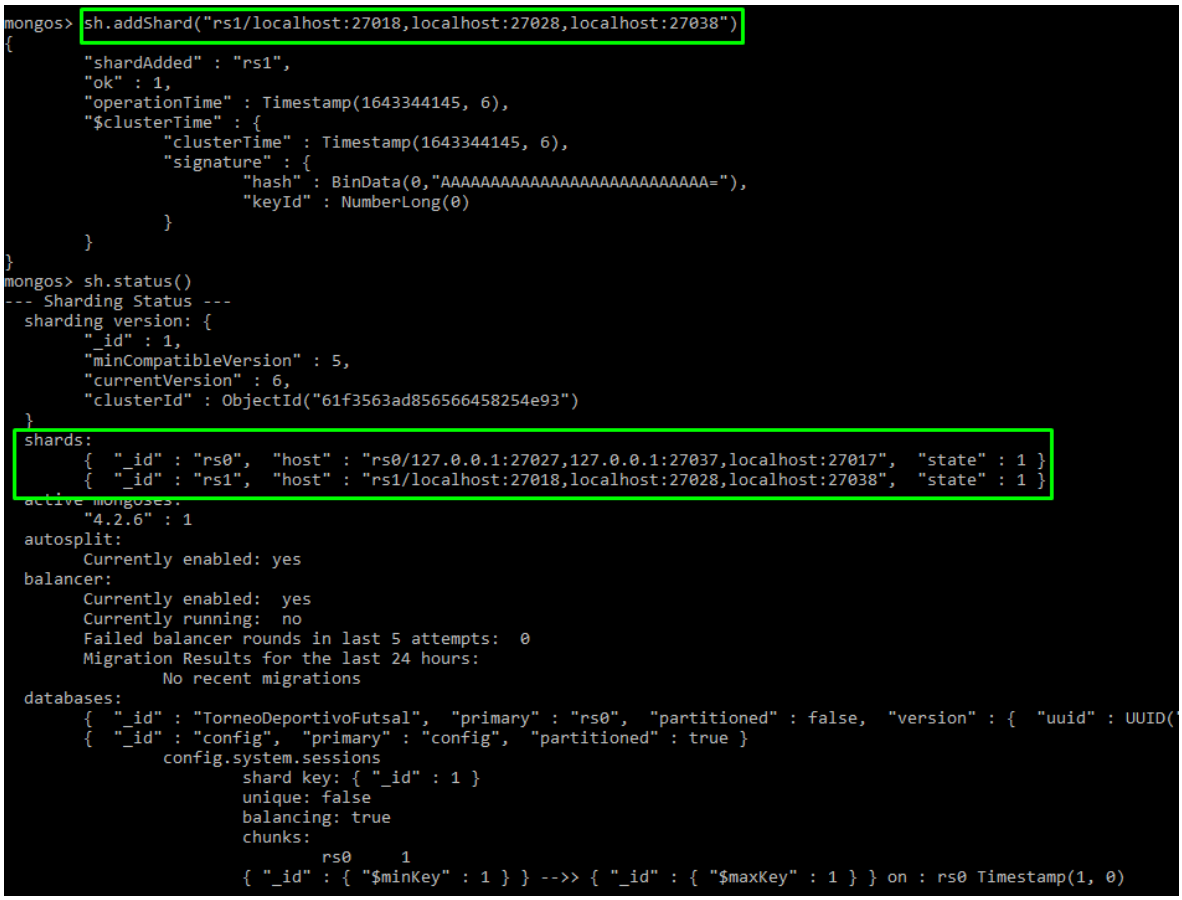
```
Administrador: Símbolo del sistema - mongod  --port 27028 --dbpath="C:\data\repl\Nodo5" --replSet rs1 --shardsvr
Microsoft Windows [Versión 10.0.19042.1466]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>cd..

C:\Windows>cd..

C:\>cd Program Files\MongoDB\Server\4.2\bin

C:\Program Files\MongoDB\Server\4.2\bin>mongod --port 27028 --dbpath="C:\data\repl\Nodo5" --replSet rs1 --shardsvr
2022-01-27T23:21:18.047-0500 I  CONTROL  [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2022-01-27T23:21:18.447-0500 W  ASIO     [main] No TransportLayer configured during NetworkInterface startup
2022-01-27T23:21:18.448-0500 I  CONTROL  [initandlisten] MongoDB starting : pid=9716 port=27028 dbpath=C:\data\repl\Nodo5 64-bit host=DESKTOP-2Q941NA
2022-01-27T23:21:18.448-0500 I  CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2022-01-27T23:21:18.448-0500 I  CONTROL  [initandlisten] db version v4.2.6
2022-01-27T23:21:18.448-0500 I  CONTROL  [initandlisten] git version: 20364840b8f1af16917e4c23c1b5f5efd8b352f8
2022-01-27T23:21:18.448-0500 I  CONTROL  [initandlisten] allocator: tcmalloc
2022-01-27T23:21:18.448-0500 I  CONTROL  [initandlisten] modules: none
2022-01-27T23:21:18.448-0500 I  CONTROL  [initandlisten] build environment:
2022-01-27T23:21:18.448-0500 I  CONTROL  [initandlisten]     distmod: 2012plus
2022-01-27T23:21:18.448-0500 I  CONTROL  [initandlisten]     distarch: x86_64
2022-01-27T23:21:18.448-0500 I  CONTROL  [initandlisten]     target_arch: x86_64
2022-01-27T23:21:18.448-0500 I  CONTROL  [initandlisten] options: { net: { port: 27028 }, replication: { replSet: "rs1" }, sharding: { clusterRole: "shardsvr" }, storage:
2022-01-27T23:21:18.449-0500 I  STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=6630M,cache_overflow=(file_max=0M),session_max=33000,eviction=(threads_
ed=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose
2022-01-27T23:21:18.760-0500 I  STORAGE  [initandlisten] WiredTiger message [1643343678:760366][9716:140712841139536], txn-recover: Set global recovery timestamp: (0, 0)
2022-01-27T23:21:19.028-0500 I  RECOVERY [initandlisten] WiredTiger recoveryTimestamp. Ts: Timestamp(0, 0)
2022-01-27T23:21:19.242-0500 I  STORAGE  [initandlisten] Timestamp monitor starting
2022-01-27T23:21:19.319-0500 I  CONTROL  [initandlisten]
2022-01-27T23:21:19.319-0500 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-01-27T23:21:19.321-0500 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
```

```
Administrador: Símbolo del sistema - mongod  --port 27038 --dbpath="C:\data\repl\Nodo6" --replSet rs1 --shardsvr
Microsoft Windows [Versión 10.0.19042.1466]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>cd..

C:\Windows>cd..

C:\>cd Program Files\MongoDB\Server\4.2\bin

C:\Program Files\MongoDB\Server\4.2\bin>mongod --port 27038 --dbpath="C:\data\repl\Nodo6" --replSet rs1 --shardsvr
2022-01-27T23:22:22.993-0500 I  CONTROL  [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2022-01-27T23:22:23.384-0500 W  ASIO     [main] No TransportLayer configured during NetworkInterface startup
2022-01-27T23:22:23.385-0500 I  CONTROL  [initandlisten] MongoDB starting : pid=2184 port=27038 dbpath=C:\data\repl\Nodo6 64-bit host=DESKTOP-2Q941NA
2022-01-27T23:22:23.385-0500 I  CONTROL  [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2022-01-27T23:22:23.385-0500 I  CONTROL  [initandlisten] db version v4.2.6
2022-01-27T23:22:23.385-0500 I  CONTROL  [initandlisten] git version: 20364840b8f1af16917e4c23c1b5f5efd8b352f8
2022-01-27T23:22:23.385-0500 I  CONTROL  [initandlisten] allocator: tcmalloc
2022-01-27T23:22:23.385-0500 I  CONTROL  [initandlisten] modules: none
2022-01-27T23:22:23.385-0500 I  CONTROL  [initandlisten] build environment:
2022-01-27T23:22:23.385-0500 I  CONTROL  [initandlisten]     distmod: 2012plus
2022-01-27T23:22:23.385-0500 I  CONTROL  [initandlisten]     distarch: x86_64
2022-01-27T23:22:23.385-0500 I  CONTROL  [initandlisten]     target_arch: x86_64
2022-01-27T23:22:23.385-0500 I  CONTROL  [initandlisten] options: { net: { port: 27038 }, replication: { replSet: "rs1" }, sharding: { clusterRole: "shardsvr" }, storage:
2022-01-27T23:22:23.386-0500 I  STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=6630M,cache_overflow=(file_max=0M),session_max=33000,eviction=(threads_
ed=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000,close_scan_interval=10,close_handle_minimum=250),statistics_log=(wait=0),verbose
2022-01-27T23:22:23.756-0500 I  STORAGE  [initandlisten] WiredTiger message [1643343743:756060][2184:140712841139536], txn-recover: Set global recovery timestamp: (0, 0)
2022-01-27T23:22:24.051-0500 I  RECOVERY [initandlisten] WiredTiger recoveryTimestamp. Ts: Timestamp(0, 0)
```

- Ahora inicializamos nuestro nuevo conjunto de nodos sobre la replica nombrada “rs1”, para su posterior configuración sobre nuestro shard.

```
> rs.initiate( {_id : "rs1",members: [{ _id: 0, host: "localhost:27018" },{ _id: 1, host: "localhost:27028" },{ _id: 2, host: "localhost:27038" }]})
{
        "ok" : 1,
        "$clusterTime" : {
                "clusterTime" : Timestamp(1643344035, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        },
        "operationTime" : Timestamp(1643344035, 1)
}
```

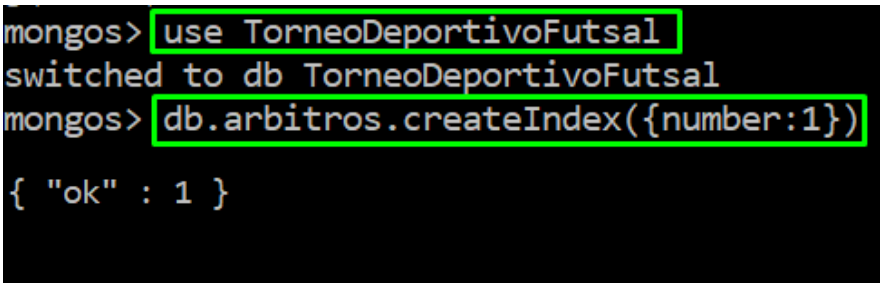
Y luego le añadiremos al shard, quedandose asi:



Casos de prueba

- Ahora necesitamos desarrollar la fragmentación de una de nuestras collecciones y poder validar que la configuración e implementación de particionamiento en forma horizontal, estara dispuesta a ejercer su labor correspondiente, es asi que antes de poder fragmentar una colección, primero se tendra que habilitar la fragmentación para la base de datos de nuestro torneo deportivo y de la colección que se vaya a elegir.

- Luego de esto sera necesario crear un indice para la clave de fragmentación, esto debido para que la colección sea fragmentada y determine la clave de fragmento, logrando que se distribuyan los documentos entre los fragmentos configurados.



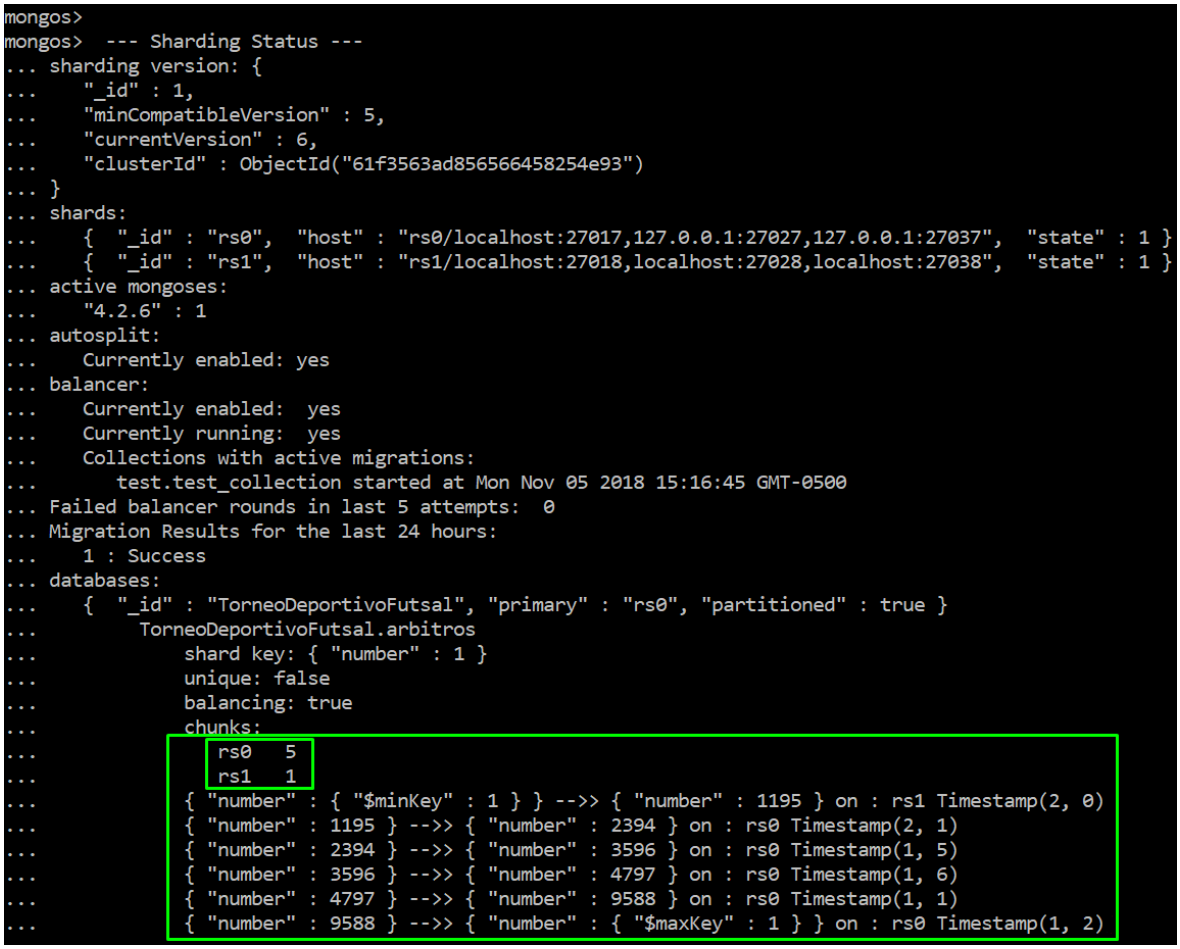
- Ahora si podremos empezar a fragmentar nuestra colección “arbitros” de nuestra base de datos, indicando siempre adiconalmente nuestra clave de fragmento anteriormente definida.

```
mongos>
mongos> sh.shardCollection("TorneoDeportivoFutsal.arbitros",{"number":1})

{"collectionsharded":"TorneoDeportivoFutsal.arbitros","ok" : 1}
```

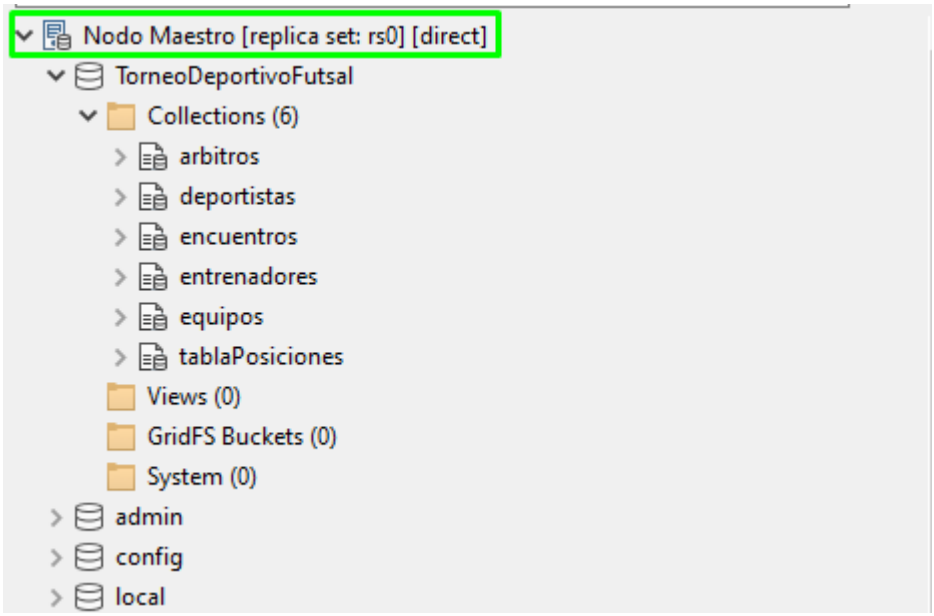
- El sistema ahora habra distribuido los documentos y podremos validar que se haya ejecutado de manera correcta el flujo de partición sobre los datos que poseemos actualmente en nuestra colección “arbitros”, en primer lugar lograremos visualizar el equilibrio que poseen nuestras replicas, nodos y fragmentos, es asi que:

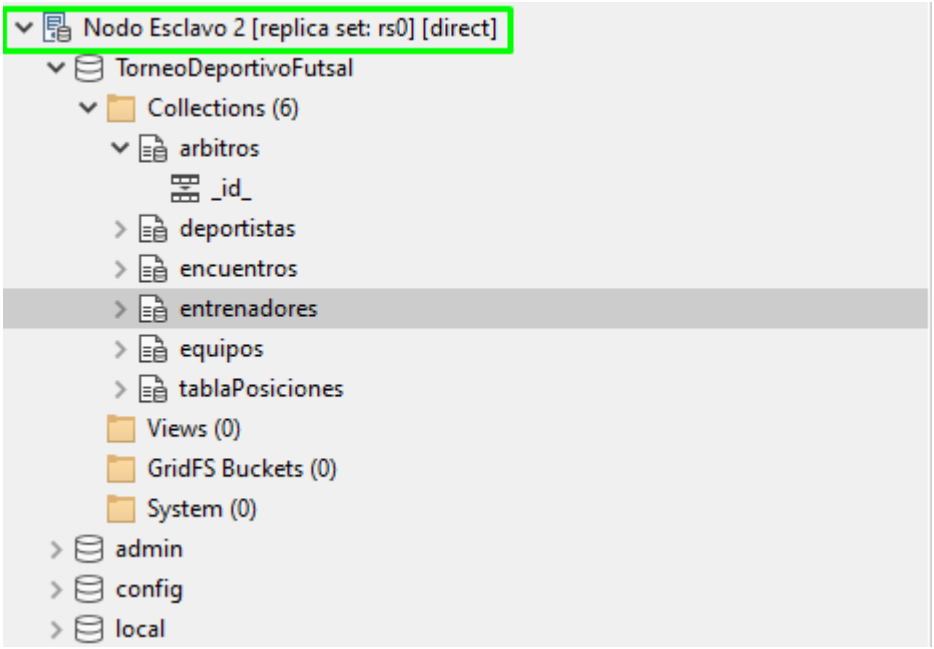
```
mongos>
mongos> {
...     "raw" : {
...         "rs0/localhost:27017,127.0.0.1:27027,127.0.0.1:27037" : {
...           "db" : "TorneoDeportivoFutsal",
...           "collections" : 1,
...           "views" : 0,
...           "objects" : 640545,
...           "avgObjSize" : 70.83200339949052,
...           "dataSize" : 45370913,
...           "storageSize" : 50438144,
...           "numExtents" : 0,
...           "indexes" : 2,
...           "indexSize" : 24502272,
...           "ok" : 1,
...           "$gleStats" : {
...                     "electionId" : ObjectId("61d783e7f5563ceee5822cfa")
...                   }
...         },
...         "rs1/localhost:27018,localhost:27028,localhost:27038" : {
...           "db" : "TorneoDeportivoFutsal",
...           "collections" : 1,
...           "views" : 0,
...           "objects" : 359455,
...           "avgObjSize" : 70.83259935179647,
...           "dataSize" : 25461132,
...           "storageSize" : 8630272,
...           "numExtents" : 0,
...           "indexes" : 2,
...           "indexSize" : 8151040,
...           "ok" : 1,
...           "$gleStats" : {
...             "electionId" : ObjectId("61d74ced33c9f0547630c562")
...           }
...
...         }
...     },
...     "objects" : 1000000,
...     "avgObjSize" : 70,
```

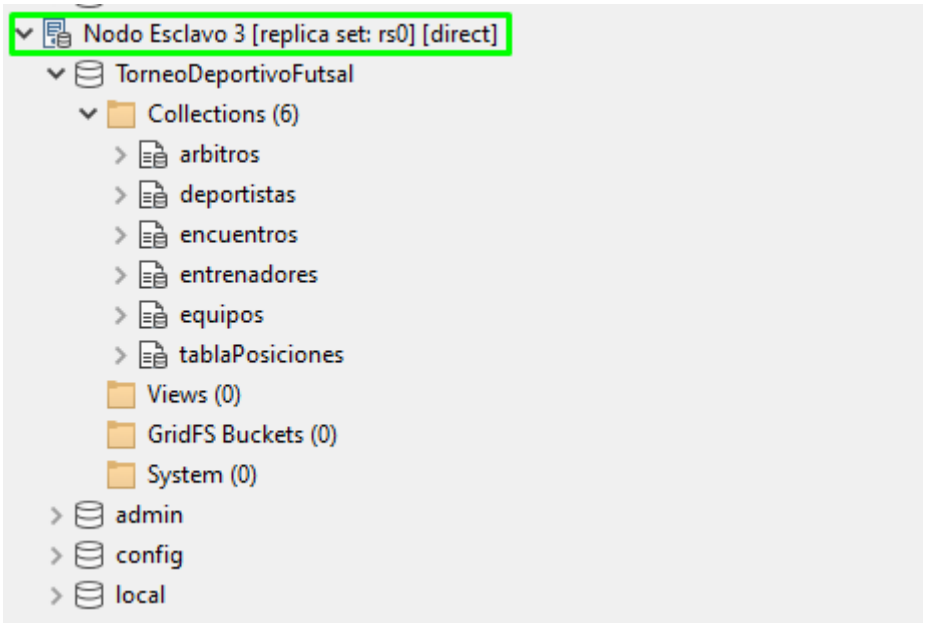
Ahora detallaremos el equilibrio para el particionamiento de nuestra colección, desde otro tipo de comando, para conocer cada uno de los estados en los que se encuentra nuestra fragmentación, de tal forma se muestr que:

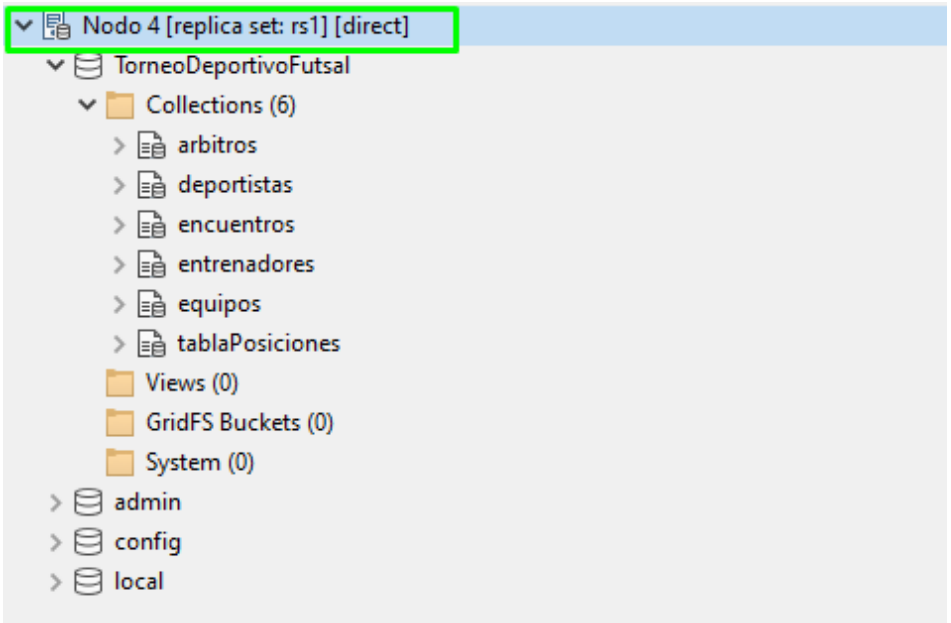


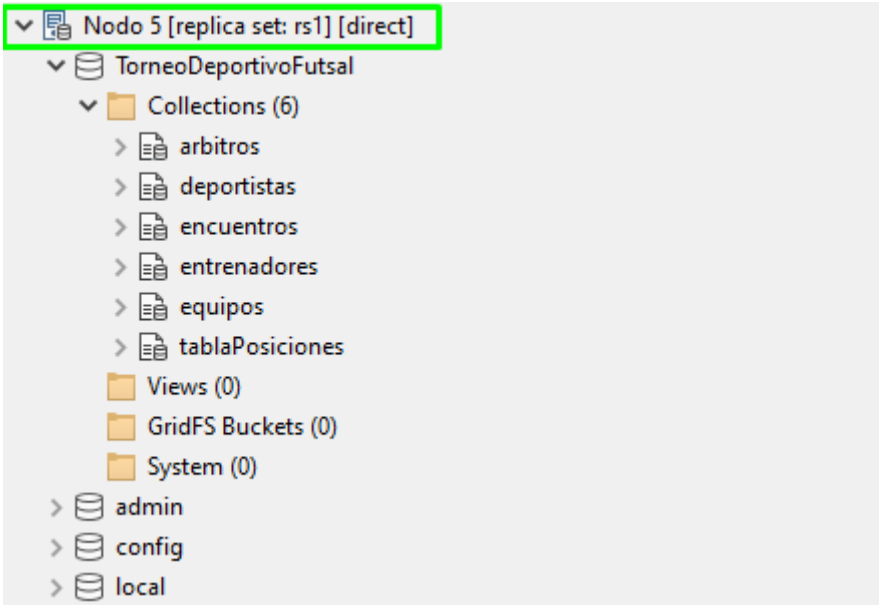
- Por utlimo graficamente podemos observar la distribución de nuestra base de datos sobre nuestras 2 replicas y cada uno de sus nodos definidos.

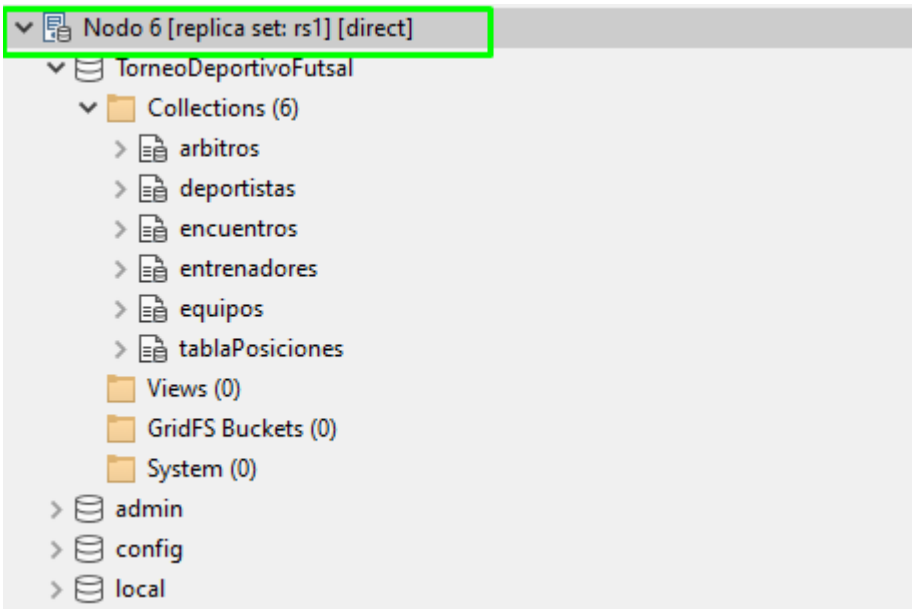





Nodo Esclavo 3 [replica set: rs0] [direct]
TorneoDeportivoFutsal
Collections (6)
arbitros
deportistas
encuentros
entrenadores
equipos
tablaPosiciones
Views (0)
GridFS Buckets (0)
System (0)
admin
config
local


Nodo 4 [replica set: rs1] [direct]
TorneoDeportivoFutsal
Collections (6)
arbitros
deportistas
encuentros
entrenadores
equipos
tablaPosiciones
Views (0)
GridFS Buckets (0)
System (0)
admin
config
local


Nodo 5 [replica set: rs1] [direct]
TorneoDeportivoFutsal
Collections (6)
arbitros
deportistas
encuentros
entrenadores
equipos
tablaPosiciones
Views (0)
GridFS Buckets (0)
System (0)
admin
config
local


Nodo 6 [replica set: rs1] [direct]
TorneoDeportivoFutsal
Collections (6)
arbitros
deportistas
encuentros
entrenadores
equipos
tablaPosiciones
Views (0)
GridFS Buckets (0)
System (0)
admin
config
local

Bibliografia

- Sarasa, A. (2016). Introducción a las bases de datos NoSQL usando MongoDB. Editorial UOC.
- https://docs.mongodb.com/manual/sharding/