Actividad 4 : Pruebas de particionamiento de bases de datos NoSQL

Juan Sebastián Vacca Peña

ID 100098569

Corporación Universitaria Iberoamericana

Bases de Datos Avanzadas

13 diciembre de 2023

Contenido

Requerimientos funcionales

Escalabilidad

La base de datos debe ser capaz de escalar horizontalmente, aumentando su tamaño en función al aumento de usuarios, sin generar interrupciones ni conflictos al momento de generar consultas.

Rendimiento

La base de datos deberá mantener un rendimiento eficiente manejando tráfico de diferentes magnitudes.

Disponibilidad

El sistema debe estar en funcionamiento continuo durante todo el evento, sin interrupciones, y estar disponible en todo momento para los usuarios.

Seguridad

La base de datos debe ser segura y confiable, para conservar temas de confidencialidad dentro del campeonato.

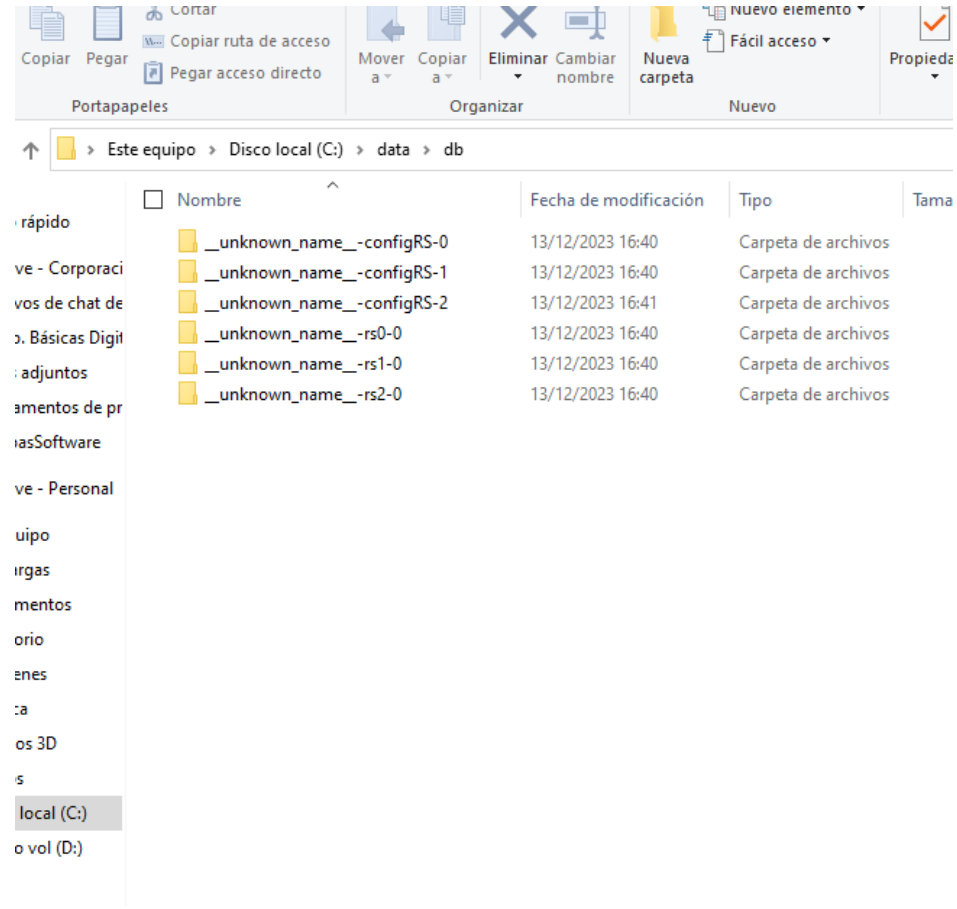
Respaldo

La base de datos tiene respaldo de información al ser replicada y alojada en nodos secundarios que garantizan la información de la base de datos.

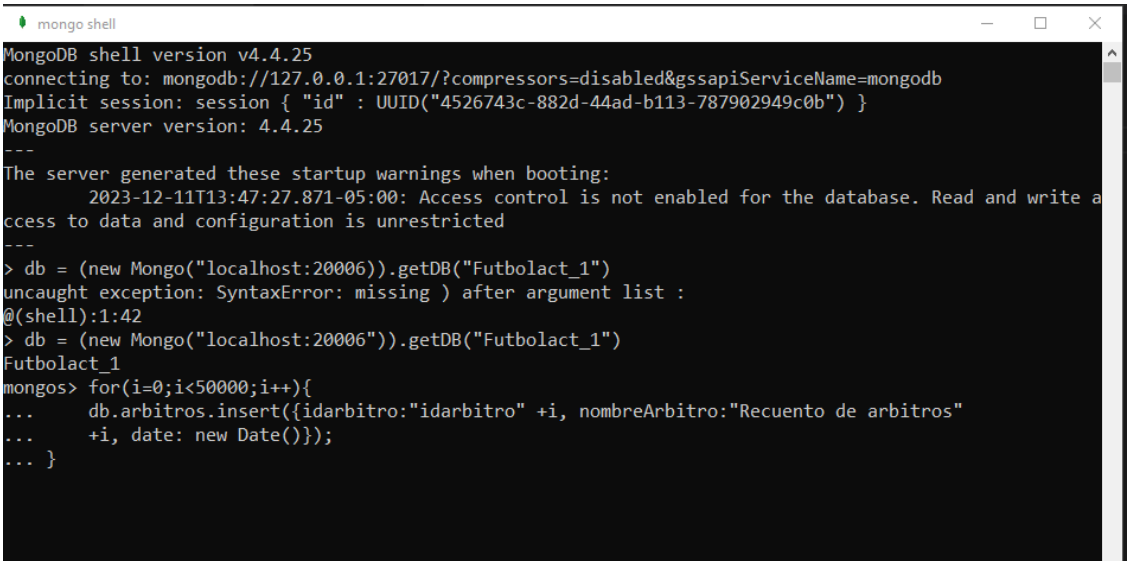
Pruebas de particionamiento

1. Levantamiento del cluster: cluster=new ShardingTest ({shards: 3, chunksize:1})

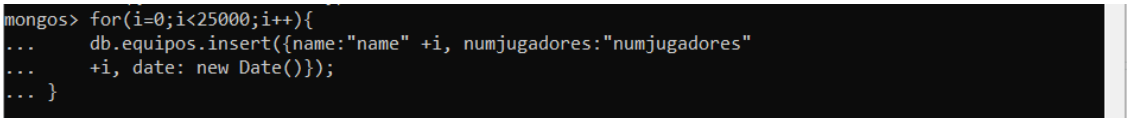
```
    }, msg, timeout);
},
        "getSecondaries" : function(timeout) {
    var master = this.getPrimary(timeout);
    var secs = [];
    for (var i = 0; i < this.nodes.length; i++) {
        if (this.nodes[i] != master) {
            secs.push(this.nodes[i]);
        }
    }

    return secs;
},
        "getSecondary" : function(timeout) {
    return this.getSecondaries(timeout)[0];
},
        "getArbiters" : function() {
    let arbiters = [];
    for (let i = 0; i < this.nodes.length; i++) {
        const node = this.nodes[i];

        let isArbiter = false;

        assert.retryNoExcept(() => {
            isArbiter = isNodeArbiter(node);
            return true;
        }, `Could not call 'isMaster' on ${node}.`, 3, 1000);
```



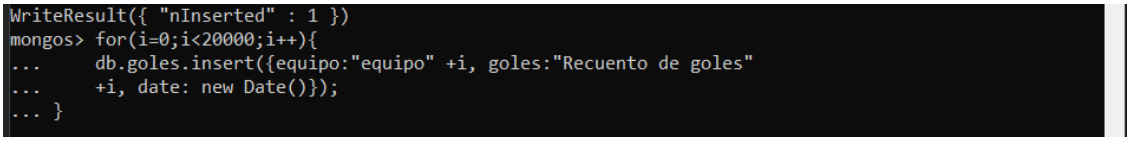
2. Hacemos ingreso de una cantidad de registros aleatoria en diferentes tablas alojadas en la base de datos “Futbolact_1”. Con esto comprobamso el renidmiento de la base de datos con cantidades grandes de datso superiores a un dato, en este caso 50000 datos, 25000 y 20000 repsectivamente.



1 inserción de datos a tabla arbitro



2 *inserción de datos a tabla equipos*

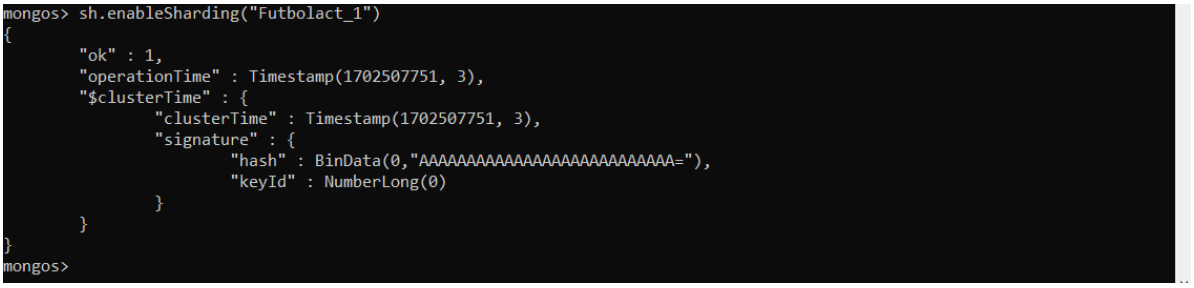


3inserción de datos a tabla goles

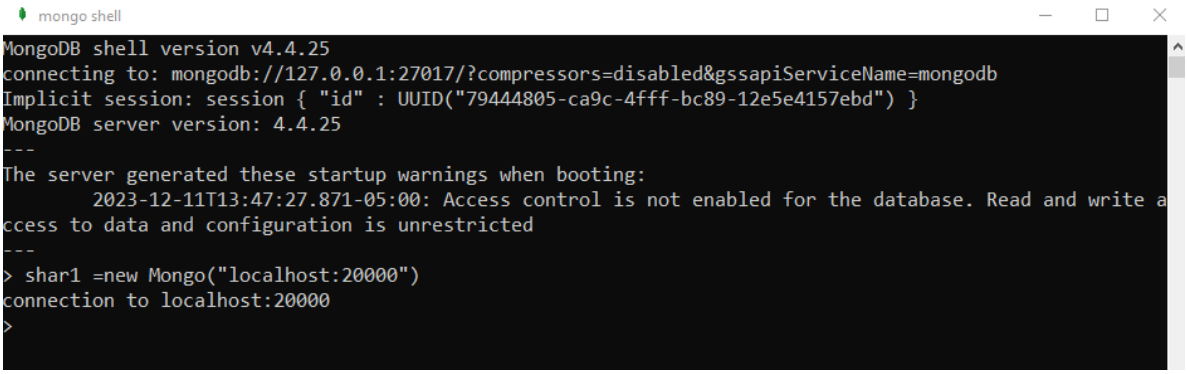
Verificamos la inserción de los datos: db.arbitros.count(), db.equipos.count(), db.goles.count(), con esto verifiacmos el funcionamiento de la base de datos y la captura delas inserciones registradas anteriormente

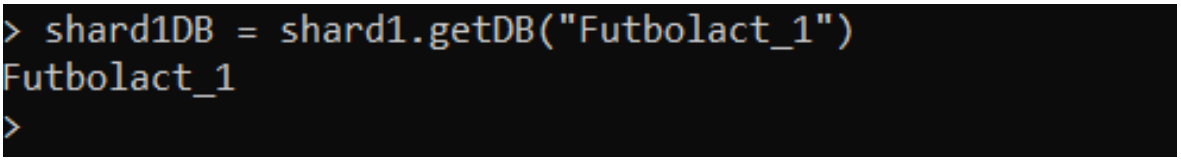
3. Activación del sharding: para este proceso nos tenemos que redireccionar a la consola mongo Shell donde se creo el balanceador que alojamos en el puerto 20006 shard1 = new Mongo("localhost:20006")

4. Activamos el sharding en la base de datos: sh.enableSharding("Futbolact_1")



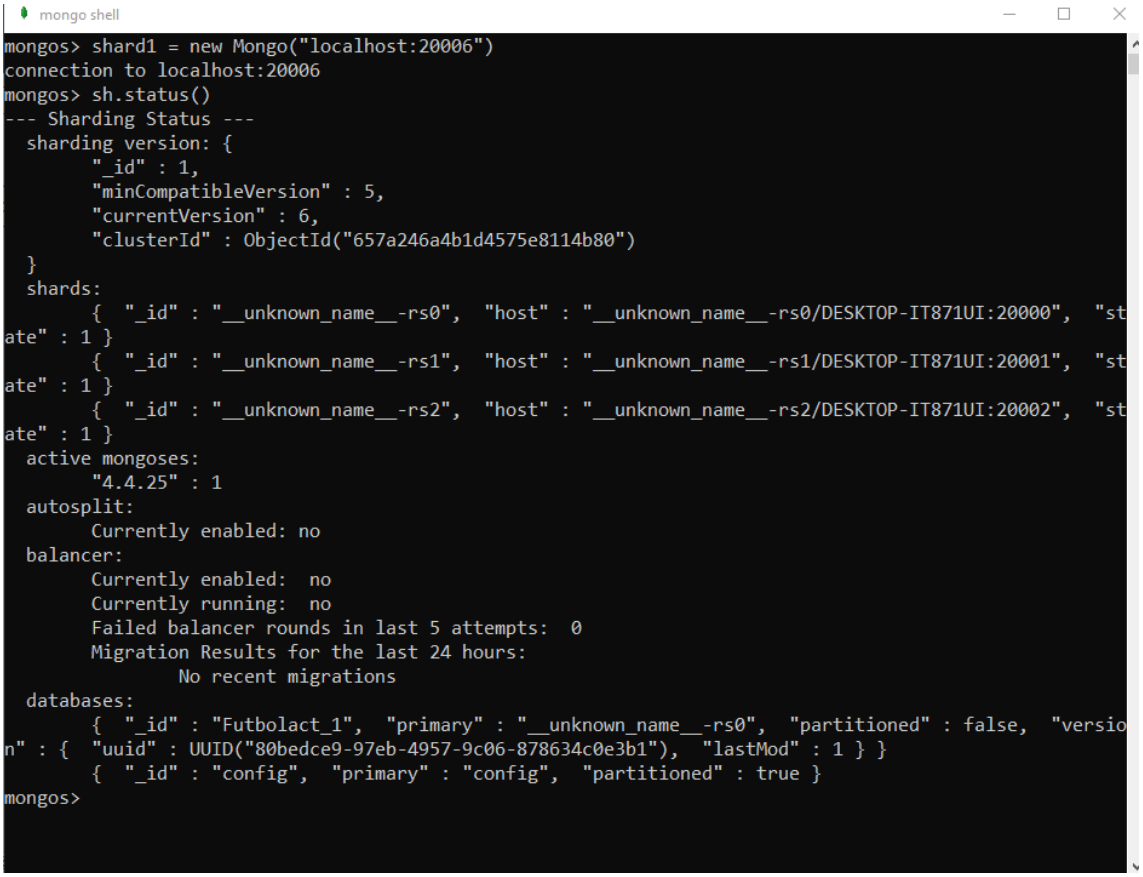
5. Comprobamos la distribución de los datos





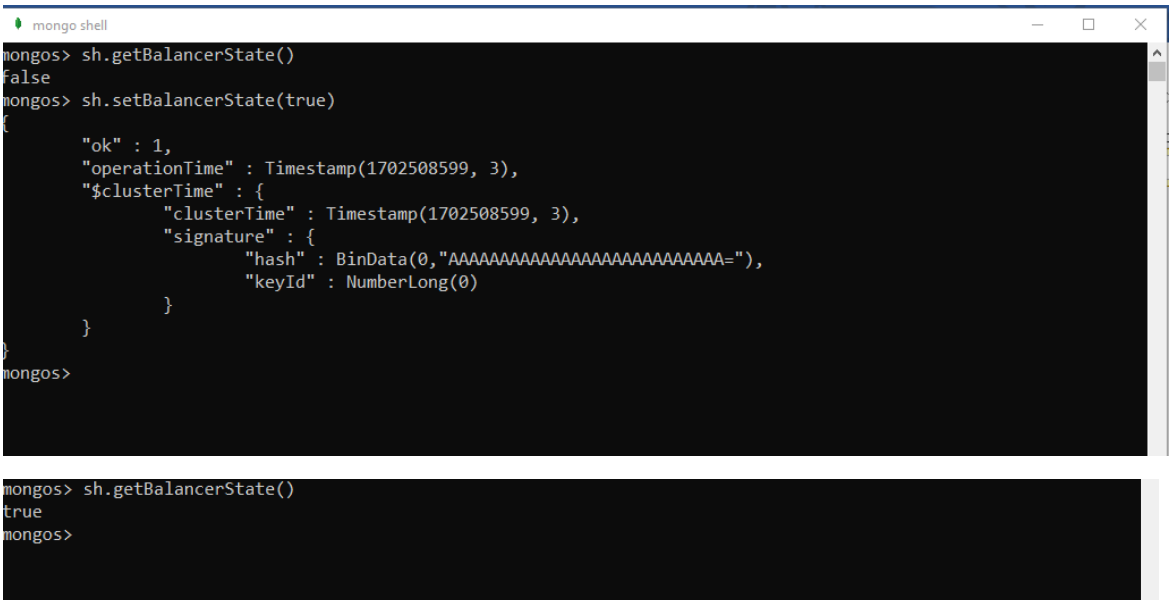
6. Activación del shard

7. Probamos la existencias del shard y su estado

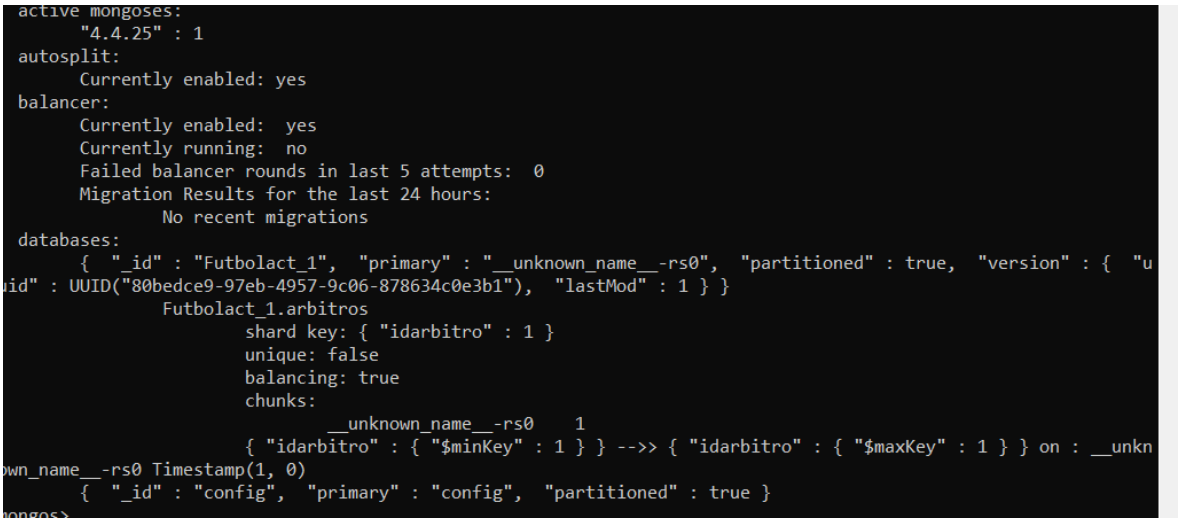


8. Prueba del balanceador para determinar si está activo o no

9. Se coloca en marcha el balanceador y se vuelve a realizar su prueba de actividad



10. Probamos el status del balanceador



11. Prueba de shards activos

```
false
mongos> db.adminCommand({listshards: 1})
{
        "shards" : [
                {
                        "_id" : "__unknown_name__-rs0",
                        "host" : "__unknown_name__-rs0/DESKTOP-IT871UI:20000",
                        "state" : 1
                },
                {
                        "_id" : "__unknown_name__-rs1",
                        "host" : "__unknown_name__-rs1/DESKTOP-IT871UI:20001",
                        "state" : 1
                },
                {
                        "_id" : "__unknown_name__-rs2",
                        "host" : "__unknown_name__-rs2/DESKTOP-IT871UI:20002",
                        "state" : 1
                }
        ],
        "ok" : 1,
        "operationTime" : Timestamp(1702509021, 3),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1702509021, 3),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
mongos>
```

Github

https://github.com/SebastianVacca/BD