

Arnau Muñoz Barrera, NIU: 1665982

José Ortín López, NIU: 1667573

Albert Vacas Martínez, NIU: 1665473

Grup: GABD - grup15

Pràctica 3

Exercici 1:

Per a completar la resolució d'aquest exercici vam de, primerament, crear i generar tota la distribució i jerarquia de rols, d'usuaris i de privilegis en les bases de dades corresponents. A continuació, vam reiniciar les màquines perquè estiguessin en funcionament amb el flag d'autenticació activat.

Per altra banda, vam finalitzar el script d'inserció i vam afegir la informació corresponent a allCountries, deixant córrer la màquina fins a arribar al 65% com es pot veure a la següent imatge:

```
El fitxer c:\Users\arnau\OneDrive\Escritorio\allCountries.txt ja s'ha descarregat.  
Descomprimit a c:\Users\arnau\OneDrive\Escritorio  
allCountries: 65%|██████████ | 1117394829/1715455451 [21:38:59<12:36:00, 13184.64it/s]
```

```
if client.isStarted:  
    client.close()
```



Per finalitzar l'exercici 1, amb tota la informació ja inserida, vam realitzar les consultes i observar un gran temps d'espera per observar el resultat, com que les consultes es realitzen sense índexs.

Exercici 2:

L'objectiu d'aquest exercici és implementar un sistema de replicació en un entorn MongoDB. La replicació de les bases de dades es fa per assegurar la disponibilitat de les dades en cas de caiguda d'un dels servidors. Vam configurar el replica set rs0 amb tres nodes: un node principal, encarregat de gestionar totes les operacions d'escriptura, i dos nodes secundaris, encarregats de replicar les dades i estar preparats per actuar com a principal en el cas que es produís una fallada.

La configuració va començar modificant els servidors MongoDB perquè escoltessin en el port 37017 en lloc del predeterminat 27017. Amb la comanda **`mongod --port 37017 --dbpath /path/to/db --bind_ip_all`** ens vam assegurar que cada node estava accessible en el port 37017. Seguidament, vam generar el fitxer de claus per garantir la seguretat de la comunicació entre nodes. Per crear aquest fitxer en el node principal ho vam fer mitjançant l'ús d'OpenSSL i scp per copiar als nodes secundaris amb les eines de transferència segura. Es van reiniciar els servidors configurant l'autenticació basada en el fitxer de claus. També es va modificar el fitxer de configuració de MongoDB, afegint paràmetres que indicaven el port i la ubicació del fitxer de claus, a més d'habilitar l'autenticació. Amb la configuració dels nodes ja completada, es va inicialitzar el replica set des del node principal, especificant els detalls dels tres membres del conjunt: el node principal (main) i els dos secundaris (mongo-1 i mongo-2).

Amb el replica set ja inicialitzada es va utilitzar la comanda **`rs.status()`** per verificar l'estat del replica set. Els resultats d'aquesta comanda van ser positius, on tots els nodes havien replicat correctament les dades. El node principal, identificat com main.grup15.gabd:37017, havia estat escollit automàticament com a PRIMARY i estava sincronitzat amb els nodes secundaris mongo-1.grup15.gabd:37017 i mongo-2.grup15.gabd:37017.

Del resultat de la comanda `rs.status()` voldríem destacar algunes observacions de gran importància:

- La sincronització de tots els nodes és correcta segons els òptims observats en el resultat de la comanda assegurant que les operacions de lectura en els nodes secundaris i possibles falles mantindran la consistència de les dades.
- Un factor crucial pel correcte funcionament en cas de fallada es la quasi negligència de la latència dels **heartbeats** entre els nodes la qual cosa reflexa una xarxa estable i eficient pel manteniment de la comunicació entre els membres del replica set.
- Tots els nodes secundaris estan capacitats per assumir el rol de PRIMARY en cas de que el node principal (main) falli.
- El node PRIMARY és l'únic capacitat per realitzar operacions d'escriptura, i els nodes SECONDARY repliquen les operacions de manera quasi immediata.

En conclusió, els objectius d'aquesta pràctica han estat assolits gràcies a aquesta implementació que ha preparat el sistema per a un funcionament robust i tolerant davant falles.

Exercici 3:

En aquest exercici, s'ha implementat un sistema de sharding en MongoDB seguint els passos indicats a l'enunciat i al tutorial oficial de MongoDB.

Primerament, vam crear la carpeta `/home/student/data/shardDB-2` a les màquines `main`, `mongo-1` i `mongo-2`, on es va desar la informació necessària.

A continuació, es va duplicar el fitxer de configuració `'Master-Slave.conf'` per utilitzar-lo en el segon shard, i es van actualitzar els paràmetres del fitxer, canviant el nom del replica set a `rs1`, el port a `47017` i el `dbpath` corresponent. També es va crear l'usuari `SYS`, juntament amb tots els altres definits a l'exercici 1 per prevenir possibles implementacions properes.

Es va activar l'autenticació i es van aixecar els tres servidors de MongoDB en les respectives màquines.

Posteriorment, es va procedir a configurar el replica set `rs1` seguint els mateixos passos que per a la creació de `rs0` en l'exercici anterior. En la segona part, vam reiniciar els sis servidors entre les màquines `main`, `mongo-1` i `mongo-2` per configurar-los com a shards, i es van fer els canvis necessaris als fitxers de configuració, seguint les indicacions de la secció `Restart the Replica Set as a Shard` del tutorial.

A continuació, es va crear un nou fitxer de configuració `'shard2.conf'` per als servidors de configuració i es va utilitzar el mateix fitxer per als tres nodes. Es va crear el directori `/home/student/data/configdb` a cada màquina i es va definir com a `dbpath` en el fitxer de configuració. Els servidors els vam iniciar escoltant al port `57017`, vam habilitar l'autenticació utilitzant els mateixos fitxers de claus que en l'exercici anterior. Es van fer els canvis necessaris al fitxer de configuració per definir aquests servidors com a servidors de configuració, especificant el rol de `configsvr`, i es van reiniciar els servidors en els tres nodes novament. Després es va iniciar el servei de mongos des del node `main` amb la instrucció que se'ns dona a l'enunciat i amb el fitxer de claus `mongodb.pem` que es va utilitzar des de la carpeta `ssl`. Un cop configurat el servei de mongos, es va accedir a ell amb les credencials de l'usuari `SYS` creat prèviament, i es van afegir els dos shards, utilitzant el format adequat per al replica set `rs0` i `rs1`, especificant els ports corresponents a cada servidor. Es va utilitzar la comanda `rs.status()` per comprovar l'estat dels replica sets i assegurar-se que la configuració era correcta.

A continuació, es va habilitar el sharding per a la base de dades `GABD` des del servei de mongos i es va procedir a fragmentar la col·lecció `geonames` seguint les indicacions de la secció `Shard a Collection` del tutorial.

Finalment, es van crear els usuaris `userTest` i `userGestor` al servidor de configuració per tal que poguessin connectar-se al servei de mongos.

Exercici 4:

En aquest exercici hem d'implementar una vista per a les consultes de l'exercici 1 i comparar els dos casos en els quals s'utilitzen índex per a les consultes i en el cas que no s'utilitza cap índex.

Primerament, vam haver de crear una vista per a les dues consultes:

- Obtenir totes les localitzacions amb "country code" "ES".
- Obtenir les localitzacions de més de 5 milions d'habitants que existeixen. (La informació de la població està en el camp "population")

Seguidament, vam guardar les definicions de les vistes amb la comanda `createview()`.

Per verificar els resultats obtinguts vam executar les vistes amb `noSqlBooster`. Amb la comanda `.explain("executionStats")` podem visualitzar informació detallada sobre el pla d'execució.

Al GitHub tenim penjat els dos documents amb la informació detallada de les consultes amb índex y sense, les conclusions obtingudes són les següents:

- En l'anàlisi de la consulta sense índexs observem que la consulta ha trigat aproximadament 12,7 segons , també ha analitzat tots els documents de la col·lecció (`totalDocsExamined: 9.227.211 documents`) i tot i analitzar milions de documents només s'han retornat 2 resultats(`nReturned: 2 documents`).
- En l'anàlisi de la consulta amb índexs podem observar que la consulta ha estat pràcticament instantània, l'índex només ha examinat 2 claus per localitzar els documents rellevants (`totalKeysExamined: 2 claus`), només s'han analitzat els documents corresponents a les claus trobades a l'índex (`totalDocsExamined: 2 documents`) i s'han retornat 2 documents correctes(`nReturned: 2 documents`).

Per tant, podem concloure que, per una banda, utilitzant l'índex la consulta ha estat extremadament eficient, amb un temps d'execució insignificant i un nombre mínim de documents analitzats.

Per l'altra banda, la consulta sense índex ha sigut molt ineficient, ja que analitza tots els documents de la col·lecció, independentment de si compleixen o no els criteris de cerca. Això genera temps de resposta llargs per a col·leccions grans.