

# COMPACT - Expurgo

Após expurgo de dados, o mongo não libera o espaço do disco que não está mais em uso, pois ele considera que você tende a voltar a consumi-lo.

Para liberar mesmo o recurso (e poder reduzir o disco), é necessário rodar o `compact`

O compact faz a desfragmentação dos dados e dos índices.

Pontos de atenção:

1. Fazer snapshot antes para garantir em caso de problemas maiores.
2. Se o processo do compact for terminado de forma forçada (restart da instância ou Kill), tenha ciência de:
  - Se tiver journaling habilitado, os dados continuarão válidos independente do estado do compact. Mas, você pode precisar recriar os índices
  - Se não tiver o journaling habilitado e o compact terminar de forma forçada, não é possível garantir que o dado esteja consistente.
3. Para saber o quanto de espaço foi liberado, rode `collStats` antes e depois do compact

EX:

```
1 use notification
2 db.runCommand( { collStats : "notification", scale: 1024*1024*1024 } )
3 //=> olhar o campo freeStorageSize (já está trazendo o valor em GB)
4
```

4. O compact pode necessitar de espaço adicional livre para ser rodado.
5. O compact só roda no mongod, então, para sharding, é necessário rodar em cada shard separadamente (não é possível rodar no mongos)
6. Permissão necessária

```
1 use admin
2 db.createRole(
3   {
4     role: "myCustomCompactRole",
5     privileges: [
6       {
7         resource: { "db" : "<database>" , "collection" : "<collection>" },
8         actions: [ "compact" ]
9       }
10    ],
11    roles: []
12  }
13 )
14 use admin
15 db.grantRolesToUser("myCompactUser", [ "dbAdmin" | "myCustomCompactRole" ] )
16 use admin
17 db.createUser(
18   {
19     user: "myCompactUser",
20     pwd: "myCompactUserPassword",
```

```

21     roles: [
22         { role: "dbAdmin", db: "<database>" } | "myCustomCompactRole"
23     ]
24 }
25 )

```

7. O compact deve rodar no secundário. Rodar no primário não propaga para os secundários

#### 7.1 Para descobrir o host de cada nó, usando mongo Atlas, rode o seguinte comando:

7.1.1 - **No Windows**, rodar o comando abaixo para achar o nome dos nós do cluster:

```
1 nslookup -q=srv _mongodb._tcp.ENDPOINT_DO_CLUSTER
```

EX: `nslookup -q=srv _mongodb._tcp.eng-prd-mas-db-mongo-33-pri.xbctp.mongodb.net`

7.1.2 -- **No Linux ou no Mac**, rodar o comando abaixo para achar o nome dos nós do cluster:

```
1 dig srv _mongodb._tcp.ENDPOINT_DO_CLUSTER
```

EX: `dig srv _mongodb._tcp.chg0036745-backup-220-pri.figdw.mongodb.net`

8. Conectar no secundário:

Precisa ser via terminal (pelo Studio 3T não funcionou):

```
1 mongosh "mongodb://USER@ENDPOINT:27017" --authenticationDatabase admin --tls
```

EX:

```

1 mongosh "mongodb://dba1@meuhost-shard-00-00-pri.xbctp.mongodb.net:27017" --authenticationDatabase admin --tls
2 mongosh "mongodb://dba1@meuhost-shard-00-01-pri.xbctp.mongodb.net:27017" --authenticationDatabase admin --tls
3 mongosh "mongodb://dba1@meuhost-shard-00-02-pri.xbctp.mongodb.net:27017" --authenticationDatabase admin --tls

```

Comando para rodar compact:

```
1 db.runCommand({compact: <collection name>});
```

OBS: Tem que estar conectado no secundário!

Após rodar em todos os secundários, para chavear o nó, pelo atlas, basta ir no atlas, no cluster em “...” e escolher “test Outage”

EX:

```
1 db.runCommand({compact:"notification"});
```