Actividad 4 - Pruebas e informe de replicación en Bases de Datos NoSQL

Brayan Steven Bonilla Castellanos

Juan Carlos Monsalve Gómez

Corporación Universitaria Iberoamericana

Ingeniería de Software

Bases de datos avanzadas

Enlace Repositorio GIT

https://github.com/jcmonsalveg/Actividad-4---PruebasInformeReplicacion-

Enlace Video

https://youtu.be/6r7YDplp-g0

Casos de pruebas

- **La base de datos debe estar compuesta por 3 nodos**

  Creamos el Nodo 1 (Debe ser el principal, puerto 27017)

```
C:\Users\Admin>mongod --port 27017 --dbpath="C:\Users\Admin\Documents\IBERO\Semestre 3\Bases de datos avanzadas\Unidad 2
\Actividad 3\replicas" --replSet rs0
```

  Creamos el Nodo 2

```
C:\Users\Admin>mongod --port 27018 --dbpath="C:\Users\Admin\Documents\IBERO\Semestre 3\Bases de datos avanzadas\Unidad 2
\Actividad 3\replicas2" --replSet rs0
```

  Creamos el Nodo 3

```
C:\Users\Admin>mongod --port 27019 --dbpath="C:\Users\Admin\Documents\IBERO\Semestre 3\Bases de datos avanzadas\Unidad 2
\Actividad 3\replicas3" --replSet rs0
```

- **Los nodos secundarios deben estar agregados en el nodo principal**

  Ejecutamos el comando
  rs.initiate()
  Agregamos el nodo 2 al nodo primario

```
rs0 [direct: primary] test> rs.add("localhost:27018")
```

  Agregamos el nodo 3 al nodo primario

```
rs0 [direct: primary] test> rs.add("localhost:27019")
```

- **El estado de los nodos deben reflejar su asignación y puerto**

  Verificamos el estado

```
rs0 [direct: primary] test> rs.status()
```

```
members: [
  {
    _id: 0,
    name: 'localhost:27017',
    health: 1,
    state: 1,
    stateStr: 'PRIMARY',
    uptime: 456,
    optime: { ts: Timestamp({ t: 1664158785, i: 1 }), t: Long("1") },
    optimeDate: ISODate("2022-09-26T02:19:45.000Z"),
    lastAppliedWallTime: ISODate("2022-09-26T02:19:45.007Z"),
    lastDurableWallTime: ISODate("2022-09-26T02:19:45.007Z"),
    syncSourceHost: '',
    syncSourceId: -1,
    infoMessage: '',
    electionTime: Timestamp({ t: 1664158354, i: 2 }),
    electionDate: ISODate("2022-09-26T02:12:34.000Z"),
    configVersion: 5,
    configTerm: 1,
    self: true,
    lastHeartbeatMessage: ''
  },
  {
    _id: 1,
    name: 'localhost:27018',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 134,
    optime: { ts: Timestamp({ t: 1664158785, i: 1 }), t: Long("1") },
    optimeDurable: { ts: Timestamp({ t: 1664158785, i: 1 }), t: Long("1") },
    optimeDate: ISODate("2022-09-26T02:19:45.000Z"),
    optimeDurableDate: ISODate("2022-09-26T02:19:45.000Z"),
    lastAppliedWallTime: ISODate("2022-09-26T02:19:45.007Z"),
    lastDurableWallTime: ISODate("2022-09-26T02:19:45.007Z"),
    lastHeartbeat: ISODate("2022-09-26T02:19:50.530Z"),
    lastHeartbeatRecv: ISODate("2022-09-26T02:19:50.544Z"),
    pingMs: Long("0"),
```

```
    optimeDurable: { ts: Timestamp({ t: 1664158785, i: 1 }), t: Long("1") },
    optimeDate: ISODate("2022-09-26T02:19:45.000Z"),
    optimeDurableDate: ISODate("2022-09-26T02:19:45.000Z"),
    lastAppliedWallTime: ISODate("2022-09-26T02:19:45.007Z"),
    lastDurableWallTime: ISODate("2022-09-26T02:19:45.007Z"),
    lastHeartbeat: ISODate("2022-09-26T02:19:50.530Z"),
    lastHeartbeatRecv: ISODate("2022-09-26T02:19:50.544Z"),
    pingMs: Long("0"),
    lastHeartbeatMessage: '',
    syncSourceHost: 'localhost:27017',
    syncSourceId: 0,
    infoMessage: '',
    configVersion: 5,
    configTerm: 1
  },
  {
    _id: 2,
    name: 'localhost:27019',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 105,
    optime: { ts: Timestamp({ t: 1664158785, i: 1 }), t: Long("1") },
    optimeDurable: { ts: Timestamp({ t: 1664158785, i: 1 }), t: Long("1") },
    optimeDate: ISODate("2022-09-26T02:19:45.000Z"),
    optimeDurableDate: ISODate("2022-09-26T02:19:45.000Z"),
    lastAppliedWallTime: ISODate("2022-09-26T02:19:45.007Z"),
    lastDurableWallTime: ISODate("2022-09-26T02:19:45.007Z"),
    lastHeartbeat: ISODate("2022-09-26T02:19:50.531Z"),
    lastHeartbeatRecv: ISODate("2022-09-26T02:19:51.050Z"),
    pingMs: Long("0"),
    lastHeartbeatMessage: '',
    syncSourceHost: 'localhost:27018',
    syncSourceId: 1,
    infoMessage: '',
    configVersion: 5,
```

- **El nodo primario debe tener un prioridad determinada para tolerancia a fallos**

Cambiamos la prioridad al nodo principal para tolerancia a fallos pueda ser instanciado de nuevo como principal

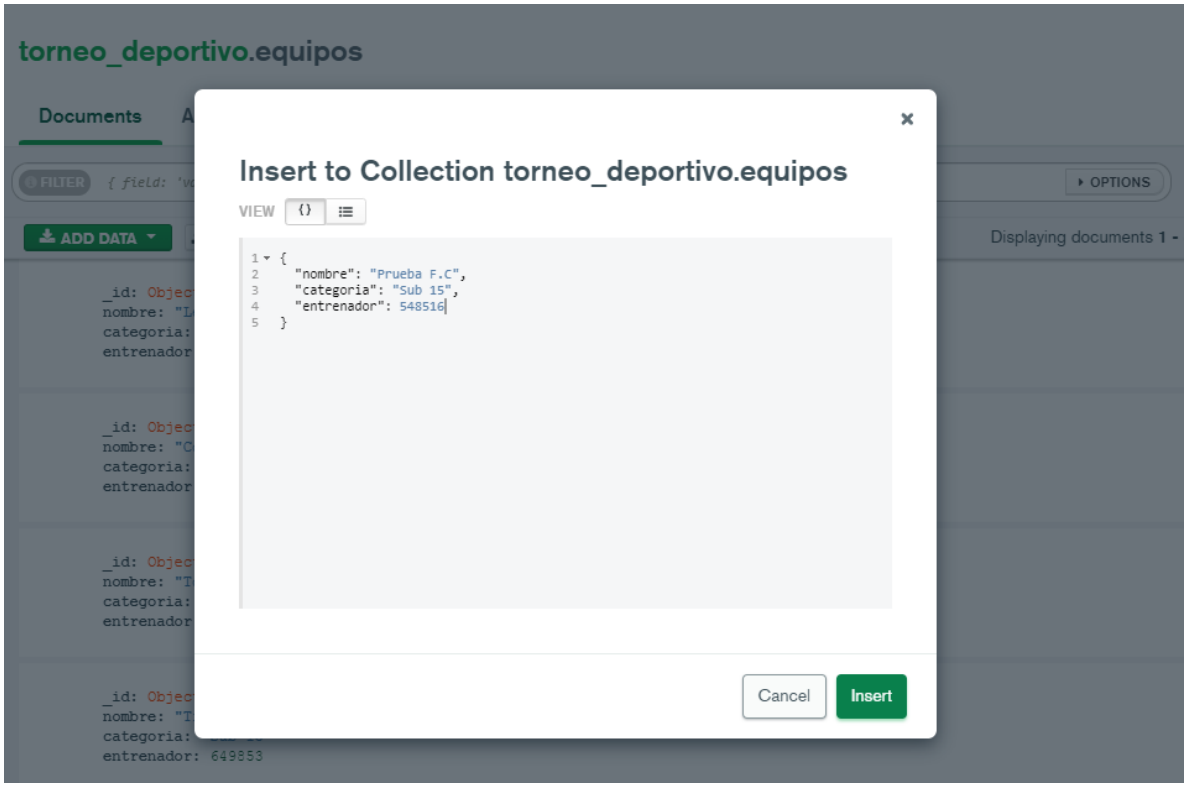
```
rs0 [direct: primary] test> rconf.members[0].priority=2
2
```

Sobrescribimos la configuración

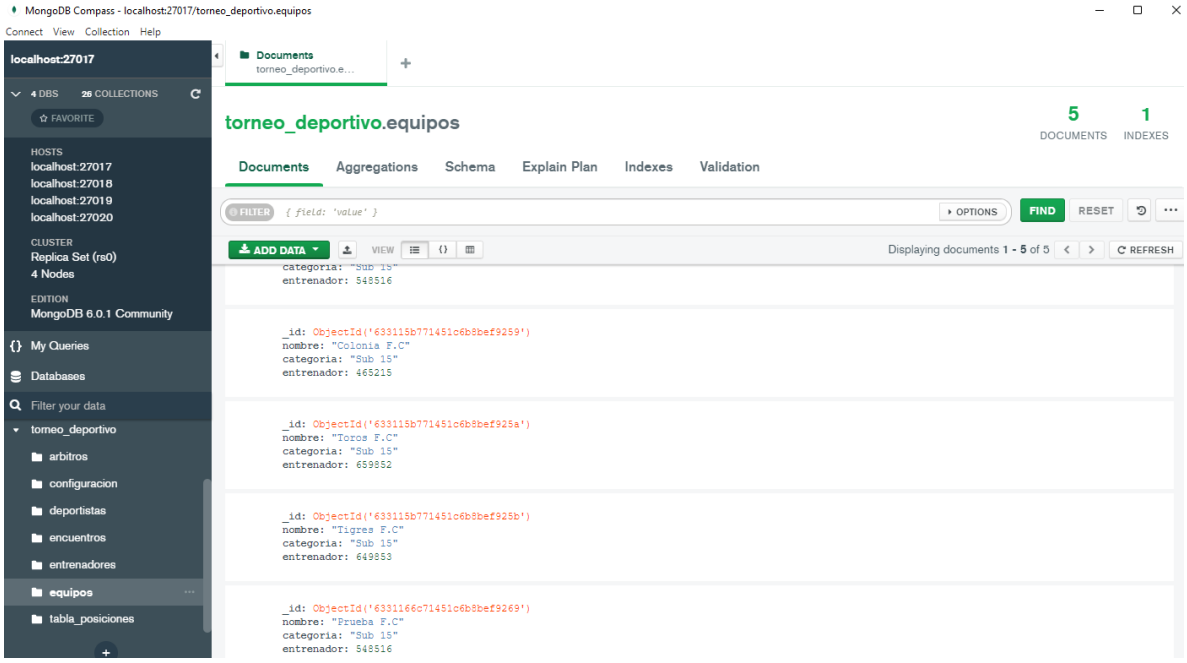
```
rs0 [direct: primary] test> rs.reconfig(rconf)
```

- **La insercion de datos debe reflejarse en los nodos secundarios**

Insertamos un nuevo equipo



Ahora tenemos 5 registros



- **La caida del nodo primario debe ser suplida por un nodo secundario**

Cerramos la conexión del nodo principal

```
C:\Users\Admin>mongosh --port 27017
Current Mongosh Log ID: 633118a361dd27d6f2e63c51
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1
.6.0
MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27017

C:\Users\Admin>
```

Verificamos cual nodo tomo el lugar de primario (Tolerancia a fallos), el cual fue el siguiente

```
C:\Users\Admin>mongosh --port 27018
Current Mongosh Log ID: 633118cd583580a48b26f62a
Connecting to:          mongodb://127.0.0.1:27018/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+
.6.0
Using MongoDB:          6.0.1
Using Mongosh:          1.6.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2022-09-25T21:16:08.188-05:00: Access control is not enabled for the database. Read and write access to data and con
iguration is unrestricted
   2022-09-25T21:16:08.189-05:00: This server is bound to localhost. Remote systems will be unable to connect to this s
rver. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --
bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable
 this warning
------

------
   Enable MongoDB's free cloud-based monitoring service, which will then receive and display
   metrics about your deployment (disk utilization, CPU, operation statistics, etc).

   The monitoring data will be available on a MongoDB website with a unique URL accessible to you
   and anyone you share the URL with. MongoDB may use this information to make product
   improvements and to suggest MongoDB products and deployment options to you.

   To enable free monitoring, run the following command: db.enableFreeMonitoring()
   To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
------
rs0 [direct: primary] test>
```

Verificamos que el nodo restante quedo como secundario

```
C:\Users\Admin>mongosh --port 27019
Current Mongosh Log ID: 63311905026e3c171e08c138
Connecting to:          mongodb://127.0.0.1:27019/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1
.6.0
Using MongoDB:          6.0.1
Using Mongosh:          1.6.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2022-09-25T21:16:35.756-05:00: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
   2022-09-25T21:16:35.757-05:00: This server is bound to localhost. Remote systems will be unable to connect to this se
rver. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --
bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable
 this warning
------

------
   Enable MongoDB's free cloud-based monitoring service, which will then receive and display
   metrics about your deployment (disk utilization, CPU, operation statistics, etc).

   The monitoring data will be available on a MongoDB website with a unique URL accessible to you
   and anyone you share the URL with. MongoDB may use this information to make product
   improvements and to suggest MongoDB products and deployment options to you.

   To enable free monitoring, run the following command: db.enableFreeMonitoring()
   To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
------

Warning: Found ~/.mongorc.js, but not ~/.mongoshrc.js. ~/.mongorc.js will not be loaded.
  You may want to copy or rename ~/.mongorc.js to ~/.mongoshrc.js.
rs0 [direct: secondary] test>
```

- **El reestablecimiento del nodo primario debe tomar lugar como nodo princpial**

Reestablecemos la instancia del nodo principal y verificamos que es primario