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

Jhon Freddy Ortiz Gómez.

Corporación Universitaria Iberoamericana.
Ingeniería de Software.
Bases de datos avanzadas
Abril del 2022

Requisito no funcional de replicación

Con el fin de garantizar la disponibilidad de la información de la base de datos se deberá contar con como mínimo 3 nodos de replicación, para ello se tendrá 1 nodo primario y tres restantes como secundarios.

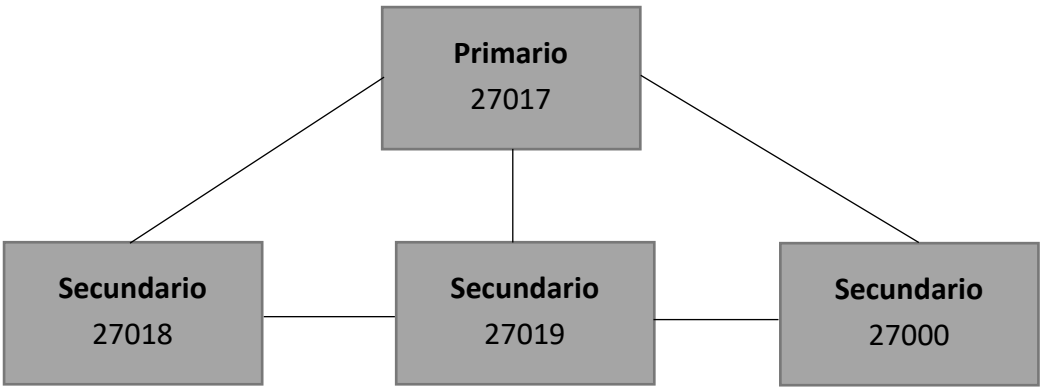
En caso de que el nodo principal quede fuera de servicio por alguna razón, la carga deberá distribuirse en los nodos restantes con el fin de mantener la disponibilidad de la información

Para tal fin se solicita la creación de los nodos de forma local, pero en puertos diferentes con el fin de correr el servicio de mongo en dichos nodos

Los puertos que se usarán son:

- **27017** – Primario.
- **27019** – Secundario
- **27018** – Secundario
- **27000** – Secundario

Para el entendimiento de lo que se necesita se muestra la siguiente gráfica en donde se evidencia el comportamiento del flujo de los datos en la replicación



Scripts para crear la replica:

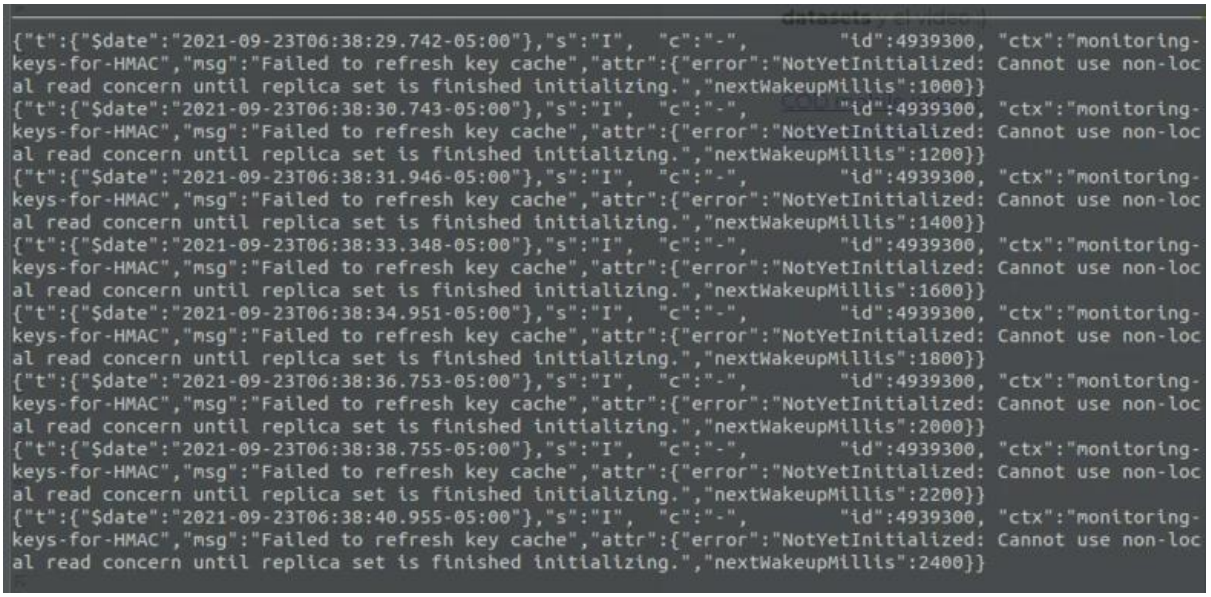
- Iniciar servidor de mongodb
    - sudo systemctl start mongod

- Crear carpetas de las bases de datos
    - mkdir set1 **&&** mkdir set2 **&&** mkdir set3

```
COD_mobile_stadistics_mongo_project/datasets on  main took 7s
➜ ls -la
total 20
drwxrwxr-x 5 carlosleoncode carlosleoncode 4096 sep 23 06:36 .
drwxrwxr-x 8 carlosleoncode carlosleoncode 4096 sep 22 01:43 ..
drwxrwxr-x 2 carlosleoncode carlosleoncode 4096 sep 23 06:01 set1
drwxrwxr-x 2 carlosleoncode carlosleoncode 4096 sep 23 06:35 set2
drwxrwxr-x 2 carlosleoncode carlosleoncode 4096 sep 23 05:54 set3

COD_mobile_stadistics_mongo_project/datasets on  main
➜
```

- Crear réplicas
    - mongod --port 27018 --dbpath --replSet myReplica



```
{"t":{"$date":"2021-09-23T06:38:29.742-05:00"},"s":"I",  "c":"-",        "id":4939300, "ctx":"monitoring-
keys-for-HMAC","msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-loc
al read concern until replica set is finished initializing.","nextWakeupMillis":1000}}
{"t":{"$date":"2021-09-23T06:38:30.743-05:00"},"s":"I",  "c":"-",        "id":4939300, "ctx":"monitoring-
keys-for-HMAC","msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-loc
al read concern until replica set is finished initializing.","nextWakeupMillis":1200}}
{"t":{"$date":"2021-09-23T06:38:31.946-05:00"},"s":"I",  "c":"-",        "id":4939300, "ctx":"monitoring-
keys-for-HMAC","msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-loc
al read concern until replica set is finished initializing.","nextWakeupMillis":1400}}
{"t":{"$date":"2021-09-23T06:38:33.348-05:00"},"s":"I",  "c":"-",        "id":4939300, "ctx":"monitoring-
keys-for-HMAC","msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-loc
al read concern until replica set is finished initializing.","nextWakeupMillis":1600}}
{"t":{"$date":"2021-09-23T06:38:34.951-05:00"},"s":"I",  "c":"-",        "id":4939300, "ctx":"monitoring-
keys-for-HMAC","msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-loc
al read concern until replica set is finished initializing.","nextWakeupMillis":1800}}
{"t":{"$date":"2021-09-23T06:38:36.753-05:00"},"s":"I",  "c":"-",        "id":4939300, "ctx":"monitoring-
keys-for-HMAC","msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-loc
al read concern until replica set is finished initializing.","nextWakeupMillis":2000}}
{"t":{"$date":"2021-09-23T06:38:38.755-05:00"},"s":"I",  "c":"-",        "id":4939300, "ctx":"monitoring-
keys-for-HMAC","msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-loc
al read concern until replica set is finished initializing.","nextWakeupMillis":2200}}
{"t":{"$date":"2021-09-23T06:38:40.955-05:00"},"s":"I",  "c":"-",        "id":4939300, "ctx":"monitoring-
keys-for-HMAC","msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-loc
al read concern until replica set is finished initializing.","nextWakeupMillis":2400}}
```

    - mongod --port 27019 --dbpath --replSet myReplica
    - mongod --port 27000 --dbpath --replSet myReplica
- Ingresar a mongo, para inicializar la réplica y agregarlas.
    - mongo
    - rs.initiate()

```
ess to data and configuration is unrestricted
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and displa
        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
        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.disableFreeMonitorin
---
> rs.status()
{
        "ok" : 0,
        "errmsg" : "no replset config has been received",
        "code" : 94,
        "codeName" : "NotYetInitialized"
}
> rs.initiate()=
... ^C

> rs.initiate()
{
        "info2" : "no configuration specified. Using a default configuration for the set",
        "me" : "127.0.0.1:27017",
        "ok" : 1
}
myReplica:OTHER>
```

- rs.add("localhost:27018")
- rs.add("localhost:27019")
- rs.add("localhost:27000")

```
        },
        "optimeDate" : ISODate("2021-09-23T11:44:42Z"),
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "electionTime" : Timestamp(1632397178, 1),
        "electionDate" : ISODate("2021-09-23T11:39:38Z"),
        "configVersion" : 7,
        "configTerm" : 1,
        "self" : true,
        "lastHeartbeatMessage" : ""
    },
    {
        "_id" : 1,
        "name" : "localhost:27018",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 130,
        "optime" : {
                "ts" : Timestamp(1632397482, 1),
                "t" : NumberLong(1)
        },
        "optimeDurable" : {
                "ts" : Timestamp(1632397482, 1),
                "t" : NumberLong(1)
        },
        "optimeDate" : ISODate("2021-09-23T11:44:42Z"),
        "optimeDurableDate" : ISODate("2021-09-23T11:44:42Z"),
        "lastHeartbeat" : ISODate("2021-09-23T11:44:51Z"),
```

Actividad 4:

CASOS DE PRUEBA

1.**CREACIÓN DE COLECCIONES**

Vamos a crear las colecciones de la base de datos desde la réplica del puerto **27018.**

```
COD_mobile_stadistics_mongo_project/COD_mobile_stadistics on  main [?] took 2m 7s
➜ mongosh --port 27018
```

```
myReplica [direct: primary] cod_mobile> db.createCollection('game_matches')
{ ok: 1 }
myReplica [direct: primary] cod_mobile> db.createCollection('game_seasons')
{ ok: 1 }
myReplica [direct: primary] cod_mobile> db.createCollection('game_user_stats')
{ ok: 1 }
```

2. INSERTAR DATOS

Insertamos un documento desde la réplica del puerto **27018.**

```
myReplica [direct: primary] cod_mobile> db.countries.insertOne({"name": "Colombia", "latitude": "131321",
 "longitude": "-132564", "created_at": new Date()})
{
  acknowledged: true,
  insertedId: ObjectId("614c76889cc53d2dda27508c")
}
```

Bajamos la réplica para ver cómo se comporta el set.

```
myReplica [direct: primary] cod_mobile> rs.stepDown(60)
```

Luego observamos inmediatamente que pasó de estado **Primario** a **Secundario.**

Antes

```
myReplica [direct: primary] cod_mobile>
```

Después

```
myReplica [direct: secondary] cod_mobile>
```

Intentamos hacer una consulta, pero nos devuelve un error ya que la réplica no es primaria.

```
myReplica [direct: secondary] cod_mobile> db.countries.find()
MongoServerError: not primary and secondaryOk=false - consider using db.getMongo().setReadPref() or readP
reference in the connection string
myReplica [direct: secondary] cod_mobile>
```

3. CONECTAR A LA NUEVA RÉPLICA PRIMARIA

Ahora las réplicas tienen el siguiente estado.

- **27017** - Secundaria
- **27018** - Secundaria
- **27019** - **Primaria**
- **27020** - Secundaria

Nos conectamos a la nueva réplica primaria

```
→ mongosh --port 27019
```

Consultamos en la colección que insertamos en la réplica anterior.

```
myReplica [direct: primary] cod_mobile> db.countries.find()
[
  {
    _id: ObjectId("614c76889cc53d2dda27508c"),
    name: 'Colombia',
    latitude: '131321',
    longitude: '-132564',
    created_at: ISODate("2021-09-23T12:43:52.682Z")
  }
]
```

Como podemos observar, la consulta nos retorna el documento que creamos anteriormente desde la replica nueva evidenciando el correcto funcionamiento

Repositorio de git Hub donde se encuentran los documentos, la bd con las réplicas y el set de datos y la documentación : https://github.com/jhofrorgo/data