Practical 5:-Sharding using Mongodb

Name :- Waqas Ansari
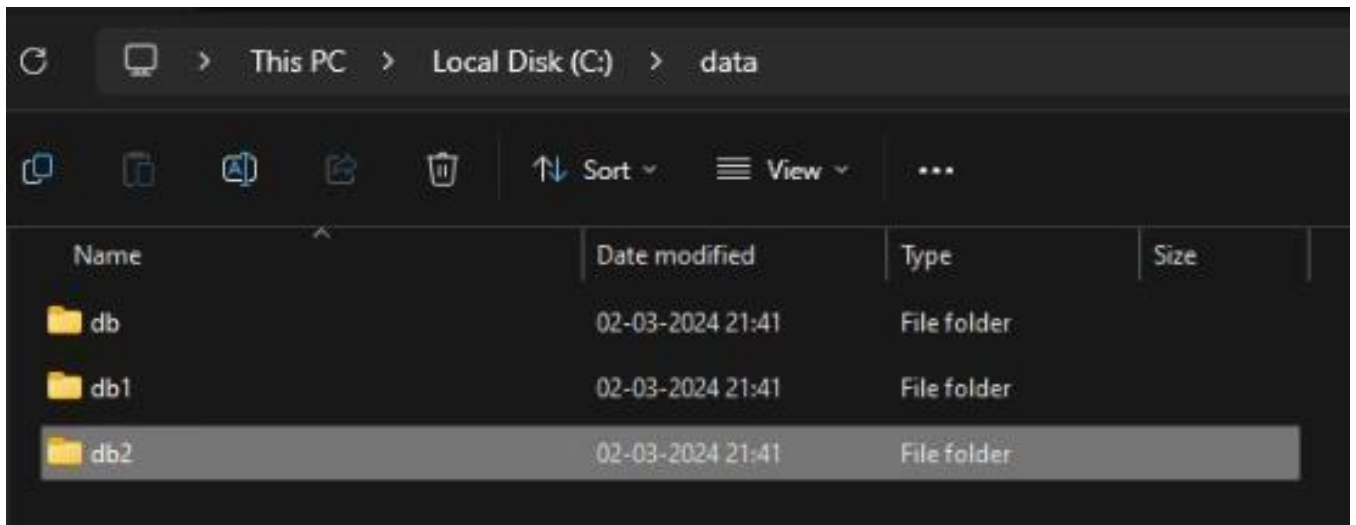
Roll no :- 37

Step 1 :-

create c:\data\db folders otherwise mongo will generate error at startup

STEP 2:-

create other folders for replica set like c:\data\db1 and db2

STEP 3:-

Go to mongodb default folder location and find mongod.conf file windows default: C:\Program Files\MongoDB\Server\7.0\data open file in edit mode and find replication: remove # from `replication` and in the next line add
``` cmd
replication:
 replSetName: "your_replica_set_name"

```
#replication:
replication:
    replSetName:"rs0"
```

open cmd for each replication folder like db, db1, db2 and so on...

```
Command Prompt          ×    +   ∨

Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Abhay>mongod --port 27017 --replSet rs0 --dbpath "C:\data\db" --bind_ip localhost
```

STEP 5:-

create other folders for replica set like c:\data\db1 and db2

type the following command to execute each replica node
Cmd > mongod --port 27017 --replSet your_replica_set_name --dbpath "C:\data\db" --bind_ip localhost

```
Command Prompt          ×    +   ∨

Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Abhay>mongod --port 27017 --replSet rs0 --dbpath "C:\data\db" --bind_ip localhost
```

repeat above step for all other replica set node with diffrenet ports like 27018, 27019 and so on and db folder names like db1, db2 and so on

STEP 6:-

initiate the settings to add members to replica set
cmd
rs.initiate({_id: "rs0", members: [{_id: 0, host: "localhost:27017"}, {_id: 1, host: "localhost:27018"}, {_id: 2, host: "localhost:27019"}]});

Step 7 :-

check the status cmd> rs.status()

```
Command Prompt    ×   +  ∨                    www.BANDICAM.com                                           –  □  ×
Microsoft Windows [Version 10.0.22621.3155]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Abhay>rs.initiate({_id: "rs0", members: [{_id: 0, host: "localhost:27017"}, {_id: 1, host: "localhost:27018"}, {_id: 2, host: "localhost
:27019"}]});
```

```
    infoMessage: '',
    configVersion: 96329,
    configTerm: -1
  },
  {
    _id: 2,
    name: 'localhost:27019',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 19,
    optime: { ts: Timestamp({ t: 1709396225, i: 1 }), t: Long('1') },
    optimeDurable: { ts: Timestamp({ t: 1709396225, i: 1 }), t: Long('1') },
    optimeDate: ISODate('2024-03-02T16:17:05.000Z'),
    optimeDurableDate: ISODate('2024-03-02T16:17:05.000Z'),
    lastAppliedWallTime: ISODate('2024-03-02T16:17:05.502Z'),
    lastDurableWallTime: ISODate('2024-03-02T16:17:05.502Z'),
    lastHeartbeat: ISODate('2024-03-02T16:17:10.975Z'),
    lastHeartbeatRecv: ISODate('2024-03-02T16:17:12.575Z'),
    pingMs: Long('0'),
    lastHeartbeatMessage: '',
    syncSourceHost: 'localhost:27017',
    syncSourceId: 0,
    infoMessage: '',
    configVersion: 96329,
    configTerm: -1
  }
],
ok: 1,
'$clusterTime': {
  clusterTime: Timestamp({ t: 1709396225, i: 1 }),
  signature: {
    hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
    keyId: Long('0')
  }
},
operationTime: Timestamp({ t: 1709396225, i: 1 })
```

STEP 8:-

check the status cmd> rs.conf()

```
  },
  {
    _id: 1,
    host: 'localhost:27018',
    arbiterOnly: false,
    buildIndexes: true,
    hidden: false,
    priority: 1,
    tags: {},
    secondaryDelaySecs: Long('0'),
    votes: 1
  },
  {
    _id: 2,
    host: 'localhost:27019',
    arbiterOnly: false,
    buildIndexes: true,
    hidden: false,
    priority: 1,
    tags: {},
    secondaryDelaySecs: Long('0'),
    votes: 1
  }
],
protocolVersion: Long('1'),
writeConcernMajorityJournalDefault: true,
settings: {
  chainingAllowed: true,
  heartbeatIntervalMillis: 2000,
  heartbeatTimeoutSecs: 10,
  electionTimeoutMillis: 10000,
  catchUpTimeoutMillis: -1,
  catchUpTakeoverDelayMillis: 30000,
  getLastErrorModes: {},
  getLastErrorDefaults: { w: 1, wtimeout: 0 },
  replicaSetId: ObjectId('65e3509d6f936be55b29c05c')
}
```

Step 9 :-

now you can add more member using

```
rs0 [direct: primary] test> rs.add({host:"localhost:27020"})

{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1709396687, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1709396687, i: 1 })
}
```

remove memeber from replication set

```
rs0 [direct: primary] test> rs.remove("localhost:27020");
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1709396796, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1709396796, i: 1 })
}
```

---------------- WARNINGS -----------------if
you stepdown primary using:
`rs.stepDown()`
then use below command to check configuration and set primary back
`rs.conf()`

you can set previliges for primary and secondary
`db.getMongo().setReadPref('primaryPreferred')`
and when you on secondary node
`db.getMongo().setReadPref('secondary')`

REMOVING OF REPLICA SETS
_stop mongodb all running instances
_comment back the `replSetName` and `replication` in `mongod.conf` file

_now start an instance using simple command
```cmd
mongod --port 27017 --dbpath "path to db folder"
```

RECONFIG MEMBERS USING rs.initiate();
_remove all secondary nodes and then type:
```cmd
db.adminCommand({ replSetStepDown: 120, force: true })
```

_this will step down the primary node

RECONFIG THE CONFIGURATION
```cmd
rs.reconfig({
  _id: "your_replica_set_name",
   members: [
    { _id: 0, host: "localhost:27017" },
    // Add other members here if needed
  ]
}, { force: true });
```

NOTE: if you want now you can delete all other files if the testing is completed.