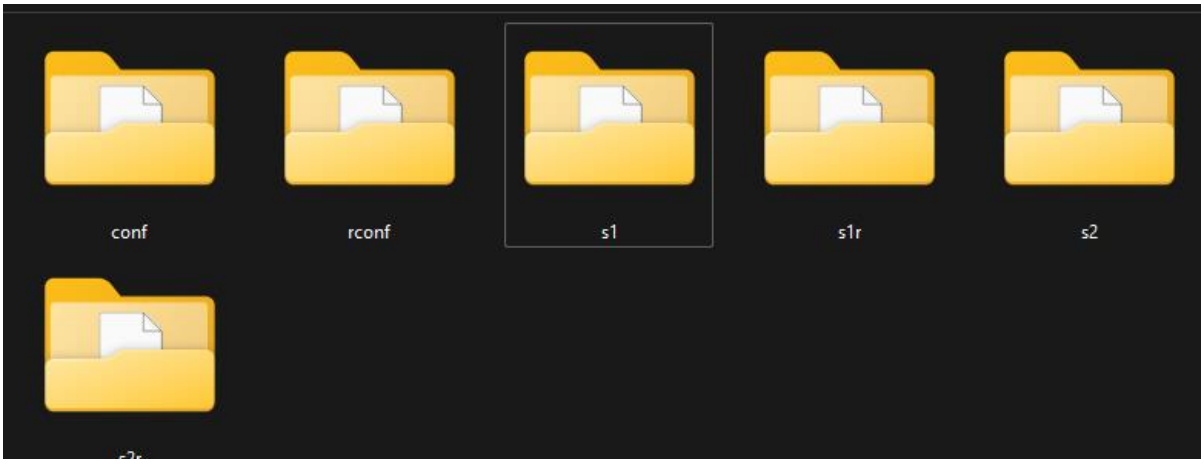
Practical 5:-Sharding using Mongodb
Name :- Arpit Chauhan
Roll no :- 40

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

create other folders for config server ,shard server and replication set like c:\data\db1 and db2



Sharding setup (locahost):

Config server:

```
C:\Users\Abhay>mongod --configsvr --port 27018 --replSet rs1 --dbpath C:\data\conf
{"t":{"$date":"2024-03-06T23:36:56.958+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "ctx":
```

```
PS C:\Users\Abhay> mongod --configsvr --port 27019 --replSet rs1 --dbpath C:\data\rconf
{"t":{"$date":"2024-03-06T23:14:15.133+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,   "c
{"t":{"$date":"2024-03-06T23:14:15.967+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701,  "c
```

```
PS C:\Users\Abhay> mongosh -port 27018
Current Mongosh Log ID: 65e8ab963793b87ff365dfb6
Connecting to:          mongodb://127.0.0.1:27018/?di
Using MongoDB:          6.0.13
```

```
test> rs.initiate({_id:'rs1',members:[{_id:0,host:'localhost:27018'},{_id:1,host:'localhost:27019'}]})

rs1 [direct: other] test> rs.status()
{
  set: 'rs1',
  date: ISODate('2024-03-06T18:08:30.243Z'),
  myState: 2,
  term: Long('3'),
  syncSourceHost: 'localhost:27019',
  syncSourceId: 1,
  configsvr: true,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 2,
  writableVotingMembersCount: 2,
  optimes: {
```

```
members: [
  {
    _id: 0,
    name: 'localhost:27018',
    health: 1,
    state: 2,
    stateStr: 'SECONDARY',
    uptime: 94,
    optime: { ts: Timestamp({ t: 1709748509, i: 1 }), t: Long('3') },
    optimeDate: ISODate('2024-03-06T18:08:29.000Z'),
    lastAppliedWallTime: ISODate('2024-03-06T18:08:29.417Z'),
    lastDurableWallTime: ISODate('2024-03-06T18:08:29.417Z'),
    syncSourceHost: 'localhost:27019',
    syncSourceId: 1,
    infoMessage: '',
    configVersion: 1,
    configTerm: 3,
    self: true,
    lastHeartbeatMessage: ''
  },
  {
    _id: 1,
    name: 'localhost:27019',
    health: 1,
    state: 1,
    stateStr: 'PRIMARY',
    uptime: 92,
    optime: { ts: Timestamp({ t: 1709748508, i: 1 }), t: Long('3') },
    optimeDurable: { ts: Timestamp({ t: 1709748508, i: 1 }), t: Long('3') },
    optimeDate: ISODate('2024-03-06T18:08:28.000Z'),
    optimeDurableDate: ISODate('2024-03-06T18:08:28.000Z'),
    lastAppliedWallTime: ISODate('2024-03-06T18:08:28.867Z'),
    lastDurableWallTime: ISODate('2024-03-06T18:08:28.867Z'),
    lastHeartbeat: ISODate('2024-03-06T18:08:29.334Z'),
    lastHeartbeatRecv: ISODate('2024-03-06T18:08:28.838Z'),
    pingMs: Long('0'),
    lastHeartbeatMessage: '',
    syncSourceHost: '',
    syncSourceId: -1,
    infoMessage: '',
    electionTime: Timestamp({ t: 1709748424, i: 1 }),
    electionDate: ISODate('2024-03-06T18:07:04.000Z'),
    configVersion: 1,
    configTerm: 3
  }
```

Shard server:

```
PS C:\Users\Abhay> mongod --shardsvr --port 27020 --replSet rs2 --dbpath C:\data\s1
{"t":{"$date":"2024-03-06T23:18:40.150+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,
,"msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisable
"}
```

```
PS C:\Users\Abhay> mongod --shardsvr --port 27021 --replSet rs2 --dbpath C:\data\s1r
{"t":{"$date":"2024-03-06T23:18:49.405+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,
:"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocc
```

```
test> rs.initiate({_id:'rs2',members:[{_id:0,host:'localhost:27020'},{_id:1,host:'localhost:27021'}]})

{ ok: 1 }
rs2 [direct: other] test> rs.status()
{
  set: 'rs2',
  date: ISODate('2024-03-06T18:12:30.341Z'),
  myState: 1,
  term: Long('1'),
```

```
 },
 members: [
   {
     _id: 0,
     name: 'localhost:27020',
     health: 1,
     state: 1,
     stateStr: 'PRIMARY',
     uptime: 1431,
     optime: { ts: Timestamp({ t: 1709748748, i: 1 }), t: Long('1') },
     optimeDate: ISODate('2024-03-06T18:12:28.000Z'),
     lastAppliedWallTime: ISODate('2024-03-06T18:12:28.493Z'),
     lastDurableWallTime: ISODate('2024-03-06T18:12:28.493Z'),
     syncSourceHost: '',
     syncSourceId: -1,
     infoMessage: '',
     electionTime: Timestamp({ t: 1709747388, i: 1 }),
     electionDate: ISODate('2024-03-06T17:49:48.000Z'),
     configVersion: 1,
     configTerm: 1,
     self: true,
     lastHeartbeatMessage: ''
   },
   {
     _id: 1,
     name: 'localhost:27021',
     health: 1,
     state: 2,
     stateStr: 'SECONDARY',
     uptime: 1372,
     optime: { ts: Timestamp({ t: 1709748748, i: 1 }), t: Long('1') },
     optimeDurable: { ts: Timestamp({ t: 1709748748, i: 1 }), t: Long('1') },
     optimeDate: ISODate('2024-03-06T18:12:28.000Z'),
     optimeDurableDate: ISODate('2024-03-06T18:12:28.000Z'),
     lastAppliedWallTime: ISODate('2024-03-06T18:12:28.493Z'),
     lastDurableWallTime: ISODate('2024-03-06T18:12:28.493Z'),
     lastHeartbeat: ISODate('2024-03-06T18:12:29.032Z'),
     lastHeartbeatRecv: ISODate('2024-03-06T18:12:30.032Z'),
     pingMs: Long('0'),
     lastHeartbeatMessage: '',
     syncSourceHost: 'localhost:27020',
     syncSourceId: 0,
     infoMessage: '',
     configVersion: 1,
     configTerm: 1
```

Shard server:

```
PS C:\Users\Abhay>  mongod --shardsvr --port 27022 --replSet rs3 --dbpath C:\data\s2
{"t":{"$date":"2024-03-06T23:20:26.769+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,
:"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtoc
{"t":{"$date":"2024-03-06T23:20:27.605+05:30"},"s":"I",  "c":"NETWORK",  "id":4915701,
```

```
PS C:\Users\Abhay>  mongod --shardsvr --port 27023 --replSet rs3 --dbpath C:\data\s2r
{"t":{"$date":"2024-03-06T23:20:37.067+05:30"},"s":"I",  "c":"CONTROL",  "id":23285,
,"msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledP
"}
```

```
PS C:\Users\Abhay> mongosh --port 27022
Current Mongosh Log ID: 65e8acf7a2e448bff823dac9
Connecting to:          mongodb://127.0.0.1:27022/?di
&appName=mongosh+2.1.4
```

```
test> rs.initiate({_id:'rs3',members:[{_id:0,host:'localhost:27022'},{_id:1,host:'localhost:27023'}]})
{ ok: 1 }
rs3 [direct: other] test> rs.status()
{
  set: 'rs3',
```

```
  members: [
    {
      _id: 0,
      name: 'localhost:27022',
      health: 1,
      state: 1,
      stateStr: 'PRIMARY',
      uptime: 1426,
      optime: { ts: Timestamp({ t: 1709748847, i: 1 }), t: Long('1') },
      optimeDate: ISODate('2024-03-06T18:14:07.000Z'),
      lastAppliedWallTime: ISODate('2024-03-06T18:14:07.069Z'),
      lastDurableWallTime: ISODate('2024-03-06T18:14:07.069Z'),
      syncSourceHost: '',
      syncSourceId: -1,
      infoMessage: '',
      electionTime: Timestamp({ t: 1709747476, i: 1 }),
      electionDate: ISODate('2024-03-06T17:51:16.000Z'),
      configVersion: 1,
      configTerm: 1,
      self: true,
      lastHeartbeatMessage: ''
    },
    {
      _id: 1,
      name: 'localhost:27023',
      health: 1,
      state: 2,
      stateStr: 'SECONDARY',
      uptime: 1385,
      optime: { ts: Timestamp({ t: 1709748847, i: 1 }), t: Long('1') },
      optimeDurable: { ts: Timestamp({ t: 1709748847, i: 1 }), t: Long('1') },
      optimeDate: ISODate('2024-03-06T18:14:07.000Z'),
      optimeDurableDate: ISODate('2024-03-06T18:14:07.000Z'),
      lastAppliedWallTime: ISODate('2024-03-06T18:14:07.069Z'),
      lastDurableWallTime: ISODate('2024-03-06T18:14:07.069Z'),
      lastHeartbeat: ISODate('2024-03-06T18:14:11.602Z'),
      lastHeartbeatRecv: ISODate('2024-03-06T18:14:10.597Z'),
      pingMs: Long('0'),
      lastHeartbeatMessage: '',
      syncSourceHost: 'localhost:27022',
      syncSourceId: 0,
      infoMessage: '',
      configVersion: 1,
      configTerm: 1
    }
```

MongoS:

```
PS C:\Users\Abhay> mongos --configdb rs1/localhost:27018,localhost:27019 --port 27017
{"t":{"$date":"2024-03-06T17:52:26.014Z"},"s":"W",  "c":"SHARDING", "id":24132,   "ctx
ning a sharded cluster with fewer than 3 config servers should only be done for testing
```

Connect to the Sharded Cluster

```
PS C:\Users\Abhay> mongosh --port 27017
Current Mongosh Log ID: 65e8ad6698b569c46a0944c9
Connecting to:          mongodb://127.0.0.1:27017/?
&appName=mongosh+2.1.4
Using MongoDB:          6.0.13
Using Mongosh:          2.1.4
mongosh 2.1.5 is available for download: https://ww
```

```
[direct: mongos] test> sh.addShard("rs2/localhost:27020,localhost:27021")
{
  shardAdded: 'rs2',
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1709747594, i: 7 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1709747594, i: 7 })
}
[direct: mongos] test> sh.status()
shardingVersion
{ _id: 1, clusterId: ObjectId('65e8ac5829d30d04c82ee780') }
---
shards
[
  {
    _id: 'rs2',
    host: 'rs2/localhost:27020,localhost:27021',
    state: 1,
    topologyTime: Timestamp({ t: 1709747594, i: 4 })
  }
]
---
active mongoses
[ { '6.0.13': 1 } ]
---
autosplit
{ 'Currently enabled': 'yes' }
---
balancer
{
  'Currently enabled': 'yes',
  'Failed balancer rounds in last 5 attempts': 0,
  'Currently running': 'no',
  'Migration Results for the last 24 hours': 'No recent migrations'
```

```
PS C:\Users\Abhay> mongosh --port 27017
Current Mongosh Log ID: 65e9e68b779faec973e4c
Connecting to:          mongodb://127.0.0.1:2
```

```
[direct: mongos] test> sh.status()
shardingVersion
{ _id: 1, clusterId: ObjectId('65e8ac5829d30d04c82ee780') }
---
shards
[
  {
    _id: 'rs2',
    host: 'rs2/localhost:27020,localhost:27021',
    state: 1,
    topologyTime: Timestamp({ t: 1709747594, i: 4 })
  }
]
---
active mongoses
[ { '6.0.13': 1 } ]
---
autosplit
```

```
[direct: mongos] test> use arpit
switched to db arpit
[direct: mongos] arpit> sh.enableSharding("arpit")
{
  ok: 1,
```

```
[direct: mongos] arpit> sh.status()
shardingVersion
{ _id: 1, clusterId: ObjectId('65e8ac5829d30d04c82ee780') }
---
shards
[
```

```
]
[direct: mongos] arpit> db.arp.createIndex({Pincode:1})
Pincode_1
[direct: mongos] arpit> sh.shardCollection("arpit.arp",{Pincode:1})
{
  collectionsharded: 'arpit.arp',
  ok: 1,
  '$clusterTime': {
```

```
]
[direct: mongos] arpit> sh.status()
shardingVersion
{ _id: 1, clusterId: ObjectId('65e8ac5829d30d04c82ee780') }
---
shards
```

```
]
[direct: mongos] arpit> sh.split
sh.splitAt    sh.splitFind

[direct: mongos] arpit> sh.splitAt("arpit.arp",{Pincode:600000})
{
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1709828903, i: 5 }),
    signature: {
```

```
[direct: mongos] arpit> sh.status()
shardingVersion
{ _id: 1, clusterId: ObjectId('65e8ac5829d30d04c82ee780') }
---
shards
```

```
[direct: mongos] arpit> sh.moveChunk("arpit.arp",{Pincode: MinKey()},"r
s2")
{
  millis: 336,
  ok: 1,
```

```
}
[direct: mongos] arpit> sh.getShardedDataDistribution()
[
  {
    ns: 'config.system.sessions',
    shards: [
      {
        shardName: 'rs2',
        numOrphanedDocs: 0,
        numOwnedDocuments: 18,
        ownedSizeBytes: 1782,
        orphanedSizeBytes: 0
      }
    ]
  },
  {
    ns: 'arpit.arp',
    shards: [
      {
        shardName: 'rs3',
        numOrphanedDocs: 11529,
        numOwnedDocuments: 7723,
        ownedSizeBytes: 648732,
        orphanedSizeBytes: 968436
      },
      {
        shardName: 'rs2',
```

```
[direct: mongos] arpit> db.arp.findOne({Pincode:{$gt:100000}})
{
  _id: ObjectId('65e9e845f933085d9ee635ae'),
  Pincode: 110001,
  District: 'CENTRAL DELHI',
  StateName: 'Delhi'
}
[direct: mongos] arpit> db.arp.findOne({Pincode:{$gt:610000}})
{
  _id: ObjectId('65e9e846f933085d9ee6648b'),
  Pincode: 610001,
  District: 'TIRUVARUR',
  StateName: 'Tamil Nadu'
}
[direct: mongos] arpit> db.arp.find().count(0
...
...
[direct: mongos] arpit> db.arp.find().count()
30781
```

```
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.
ms/PSWindows

PS C:\Users\Abhay> mongosh --port 27020
Current Mongosh Log ID: 65e9ec89c1ff83c8cafdd7d6
Connecting to:          mongodb://127.0.0.1:27020/?directConnection=true&serv
erSelectionTimeoutMS=2000&appName=mongosh+2.1.4
Using MongoDB:          6.0.13
Using Mongosh:          2.1.4
mongosh 2.1.5 is available for download: https://www.mongodb.com/try/download
/shell

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

------
   The server generated these startup warnings when booting
   2024-03-07T21:41:54.733+05:30: Access control is not enabled for the datab
ase. Read and write access to data and configuration is unrestricted
   2024-03-07T21:41:54.734+05:30: This server is bound to localhost. Remote s
ystems will be unable to connect to this server. Start the server with --bind
_ip <address> to specify which IP addresses it should serve responses from, o
r 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
------

rs2 [direct: secondary] test> show dbs
admin    72.00 KiB
arpit   908.00 KiB
config  808.00 KiB
local     1.28 MiB
rs2 [direct: secondary] test> use arpit
switched to db arpit
rs2 [direct: secondary] arpit> db.arp.find().count()
MongoServerError[NotPrimaryNoSecondaryOk]: not primary - consider using db.ge
tMongo().setReadPref() or readPreference in the connection string
rs2 [direct: secondary] arpit> db.getMongo().setReadPref('secondary')

rs2 [direct: secondary] arpit> db.arp.find().count()
11529
```