

## PRACTICAL -05

Step1: create a folder and in that folder make 9 files/folders with name(primary, secondry1, secondry2, server1, server2, server3, shard1, shard2, shard3) and run the commands mentioned below

Name	Date modified	Type	Size
primary	15-03-2024 23:07	File folder	
secondry1	15-03-2024 23:10	File folder	
secondry2	15-03-2024 23:11	File folder	
server1	15-03-2024 23:11	File folder	
server2	15-03-2024 23:11	File folder	
server3	15-03-2024 23:11	File folder	
shard1	15-03-2024 23:11	File folder	
shard2	15-03-2024 23:11	File folder	
shard3	15-03-2024 23:12	File folder	

### REPLICATION

Step2: open cmd and run commands

cd (path in which you created your folder)

```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

E:\MSc\ADBMS>start mongod --port=50000 --repSet="Server1" --dbpath="E:\MSc\ADBMS\primary"

E:\MSc\ADBMS>

C:\Program Files\MongoDB\Server\6.0\bin\mongod.exe
{"connections":{"attr":{"port":50000,"ssl":"off"}}
{"t":{"date":"2024-03-15T23:19:17.558+05:30"},"s":"I", "c":"CONTROL", "id":8423403, "ctx":"initandlisten","msg":"mongod startup complete","attr":{"Summary of time elapsed":{"Startup from clean shutdown?":true,"Statistics":{"Transport layer setup":"0 ms","Run initial syncer crash recovery":"0 ms","Create storage engine lock file in the data directory":"0 ms","Create storage engine lock file in the data directory":"0 ms","Get metadata describing storage engine":"0 ms","Create storage engine":"0 ms","Create storage engine":"37 ms","Write current PID to file":"0 ms","Write current PID to file":"0 ms","Write a new metadata for storage engine":"0 ms","Write a new metadata for storage engine":"0 ms","Initialize FCV before rebuilding indexes":"0 ms","Initialize FCV before rebuilding indexes":"0 ms","Drop abandoned idents and get back indexes that need to be rebuilt or builds that need to be restarted":"0 ms","Rebuild indexes for collections":"0 ms","Rebuild indexes for collections":"0 ms","Build user and roles graph":"0 ms","Set up the background thread pool responsible for waiting for opTimes to be majority committed":"0 ms","Start up cluster time keys manager with a local/direct keys client":"0 ms","Start up the replication coordinator":"37 ms","Create an oplog view for tenant migrations":"0 ms","Start transport layer":"0 ms","_initAndListen total elapsed time":"322 ms"}}}}
{"t":{"date":"2024-03-15T23:19:17.719+05:30"},"s":"I", "c":"-", "id":4939300, "ctx":"monitoring-keys-for-HPAC", "msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-local read concern until replica set is finished initializing.", "nextWakeUpMillis":400}}
{"t":{"date":"2024-03-15T23:19:18.121+05:30"},"s":"I", "c":"-", "id":4939300, "ctx":"monitoring-keys-for-HPAC", "msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-local read concern until replica set is finished initializing.", "nextWakeUpMillis":600}}
{"t":{"date":"2024-03-15T23:19:18.723+05:30"},"s":"I", "c":"-", "id":4939300, "ctx":"monitoring-keys-for-HPAC", "msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-local read concern until replica set is finished initializing.", "nextWakeUpMillis":800}}
{"t":{"date":"2024-03-15T23:19:19.527+05:30"},"s":"I", "c":"-", "id":4939300, "ctx":"monitoring-keys-for-HPAC", "msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-local read concern until replica set is finished initializing.", "nextWakeUpMillis":1000}}
{"t":{"date":"2024-03-15T23:19:20.529+05:30"},"s":"I", "c":"-", "id":4939300, "ctx":"monitoring-keys-for-HPAC", "msg":"Failed to refresh key cache","attr":{"error":"NotYetInitialized: Cannot use non-local read concern until replica set is finished initializing.", "nextWakeUpMillis":1200}}

```

Step3:

start mongod --port=50001 --replSet="Server1" --dbpath="(folder path)/secondary1"

start mongod --port=50002 --replSet="Server1" --dbpath="(folder path)/secondary2"

```
E:\MSc\ADBMS>start mongod --port=50001 --replSet="Server1" --dbpath="E:\MSc\ADBMS\secondary1"
E:\MSc\ADBMS>start mongod --port=50002 --replSet="Server1" --dbpath="E:\MSc\ADBMS\secondary2"
```

Step4:

mongosh --port=50000

```
test> rs.initiate({
...   _id:"Server1",
...   members:[
...     { _id:0,host:"localhost:50000"},
...     { _id:1,host:"localhost:50001"},
...     { _id:2,host:"localhost:50002"}
...   ]
... });
{ ok: 1 }
Server1 [direct: other] test> rs.status()
{
  set: 'Server1',
  date: ISODate('2024-03-15T18:45:03.695Z'),
  myState: 1,
  term: Long('1'),
  syncSourceHost: '',
  syncSourceId: -1,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1710528298, i: 1 }), t: Long('1') },
    lastCommittedWallTime: ISODate('2024-03-15T18:44:58.148Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1710528298, i: 1 }), t: Long('1') },
    appliedOpTime: { ts: Timestamp({ t: 1710528298, i: 1 }), t: Long('1') },
    durableOpTime: { ts: Timestamp({ t: 1710528298, i: 1 }), t: Long('1') },
    lastAppliedWallTime: ISODate('2024-03-15T18:44:58.148Z'),
    lastDurableWallTime: ISODate('2024-03-15T18:44:58.148Z')
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1710528268, i: 1 }),
  electionCandidateMetrics: {
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate('2024-03-15T17:58:47.793Z'),
    electionTerm: Long('1'),
    lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1710525516, i: 1 }), t: Long('-1') },
    lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1710525516, i: 1 }), t: Long('-1') },
    numVotesNeeded: 2,
    priorityAtElection: 1,
    electionTimeoutMillis: Long('10000'),
    numCatchUpOps: Long('0'),
    newTermStartDate: ISODate('2024-03-15T17:58:47.826Z'),
    wMajorityWriteAvailabilityDate: ISODate('2024-03-15T17:58:48.919Z')
  },
  members: [
```

## CONFIG SERVER

Step5:

```
mongod --configsvr --port=1030 --replSet="Server1" --dbpath="(folder path)/server1"
```

```
E:\MSc\ADBMS>mongod --configsvr --port=1030 --replSet="Server1" --dbpath="E:\MSc\ADBMS\server1"
{"t":{"$date":"2024-03-15T23:31:59.370+05:30"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"-", "msg":"Initialized wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true}}}}
{"t":{"$date":"2024-03-15T23:31:59.371+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1", "msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-03-15T23:31:59.371+05:30"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1", "msg":"Implicit TCP FastOpen in use."}
{"t":{"$date":"2024-03-15T23:31:59.373+05:30"},"s":"I", "c":"REPL", "id":5123008, "ctx":"thread1", "msg":"Implicitly setting the replication factor to 1 because the configuration server is not available."}
```

```
mongod --configsvr --port=1040 --replSet="Server1" --dbpath="(folder path)/server2"
```

```
E:\MSc\ADBMS>mongod --configsvr --port=1040 --replSet="Server1" --dbpath="E:\MSc\ADBMS\server2"
{"t":{"$date":"2024-03-15T23:33:26.702+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"-", "msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-03-15T23:33:27.775+05:30"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"thread1", "msg":"Initialized wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true}}}}
{"t":{"$date":"2024-03-15T23:33:27.778+05:30"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1", "msg":"Implicit TCP FastOpen in use."}
```

```
mongod --configsvr --port=1050 --replSet="Server1" --dbpath="(folder path)/server3"
```

```
E:\MSc\ADBMS>mongod --configsvr --port=1050 --replSet="Server1" --dbpath="(folder path)/server3"
{"t":{"$date":"2024-03-15T23:34:37.809+05:30"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"-", "msg":"Initialized wire specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true}}}}
{"t":{"$date":"2024-03-15T23:34:37.812+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"thread1", "msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-03-15T23:34:38.879+05:30"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1", "msg":"Implicit TCP FastOpen in use."}
```

```
test> rs.initiate({
...   _id:"Server1",
...   configsvr:true,
...   members:[
...     {_id:0, host:"localhost:1030"},
...     {_id:1, host:"localhost:1040"},
...     {_id:2, host:"localhost:1050"}
...   ]
... })
{ ok: 1, lastCommittedOpTime: Timestamp({ t: 1710526309, i: 1 }) }
Server1 [direct: other] test> ■
```

## SHARDING

Step6:

```
mongod --shardsvr --port=1130 --dbpath="(folder path)/shard1" --replSet="Server1"
```

```
E:\MSc\ADBMS\shard1>mongod --shardsvr --port=1130 --dbpath="E:\MSc\ADBMS\shard1" --replSet="Server1"
{"t":{"$date":"2024-03-15T23:51:25.829+05:30"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"-", "msg":"Initialized wire
specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClie
nt":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true
}}}
{"t":{"$date":"2024-03-15T23:51:27.010+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"-", "msg":"Automatically
sabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-03-15T23:51:27.012+05:30"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1", "msg":"Implicit
CP FastOpen in use "}
```

```
mongod --shardsvr --port=1140 --dbpath="(folder path)/shard2" --replSet="Server1"
```

```
E:\MSc\ADBMS\shard2>mongod --shardsvr --port=1140 --dbpath="E:\MSc\ADBMS\shard2" --replSet="Server1"
{"t":{"$date":"2024-03-15T23:52:28.971+05:30"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"-", "msg":"Initialized wire
specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClie
nt":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true
}}}
{"t":{"$date":"2024-03-15T23:52:30.116+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"-", "msg":"Automatically d
sabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-03-15T23:52:30.117+05:30"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1", "msg":"Implicit
CP FastOpen in use "}
```

```
mongod --shardsvr --port=1150 --dbpath="(folder path)/shard3" --replSet="Server1"
```

```
E:\MSc\ADBMS\shard3>
E:\MSc\ADBMS\shard3>mongod --shardsvr --port=1150 --dbpath="E:\MSc\ADBMS\shard3" --replSet="Server1"
{"t":{"$date":"2024-03-15T23:56:09.871+05:30"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"-", "msg":"Initialized wire
specification", "attr":{"spec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClie
nt":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion":17},"isInternalClient":true
}}}
{"t":{"$date":"2024-03-15T23:56:09.874+05:30"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"-", "msg":"Automatically di
sabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
{"t":{"$date":"2024-03-15T23:56:11.025+05:30"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":"thread1", "msg":"Implicit T
CP FastOpen in use "}
```

```
test> rs.initiate({
... _id:"Server1",
... members:[
... {_id: 0, host:"localhost:1130"},
... {_id: 1, host:"localhost:1140"},
... {_id: 2, host:"localhost:1150"}
... ]
... })
{ ok: 1 }
```



## STATUS

```

Server1 [direct: other] test> rs.status()
{
  set: 'Server1',
  date: ISODate('2024-03-15T18:38:40.267Z'),
  myState: 1,
  term: Long('1'),
  syncSourceHost: '',
  syncSourceId: -1,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1710527915, i: 1 }), t: Long('1') },
    lastCommittedWallTime: ISODate('2024-03-15T18:38:35.036Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1710527915, i: 1 }), t: Long('1') },
    appliedOpTime: { ts: Timestamp({ t: 1710527915, i: 1 }), t: Long('1') },
    durableOpTime: { ts: Timestamp({ t: 1710527915, i: 1 }), t: Long('1') },
    lastAppliedWallTime: ISODate('2024-03-15T18:38:35.036Z'),
    lastDurableWallTime: ISODate('2024-03-15T18:38:35.036Z')
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1710527875, i: 1 }),
  electionCandidateMetrics: {
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate('2024-03-15T18:29:14.924Z'),
    electionTerm: Long('1'),
    lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 1710527343, i: 1 }), t: Long('-1') },
    lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1710527143, i: 1 }), t: Long('-1') },
    numVotesNeeded: 2,
    priorityAtElection: 1,
    electionTimeoutMillis: Long('10000'),
    numCatchUpOps: Long('0'),
    newTermStartDate: ISODate('2024-03-15T18:29:14.949Z'),
    wMajorityWriteAvailabilityDate: ISODate('2024-03-15T18:29:16.255Z')
  },
  members: [
    {
      _id: 0,
      name: 'localhost:1130',
      health: 1,
      state: 1,
      stateStr: 'PRIMARY',
      uptime: 1035,
      optime: { ts: Timestamp({ t: 1710527915, i: 1 }), t: Long('1') },
      optimeDate: ISODate('2024-03-15T18:38:35.000Z'),
      lastAppliedWallTime: ISODate('2024-03-15T18:38:35.036Z'),
      lastDurableWallTime: ISODate('2024-03-15T18:38:35.036Z'),
      syncSourceHost: '',
    },
    {
      _id: 1,
      name: 'localhost:1140',
      health: 1,
      state: 2,
      stateStr: 'SECONDARY',
      uptime: 576,
      optime: { ts: Timestamp({ t: 1710527915, i: 1 }), t: Long('1') },
      optimeDurable: { ts: Timestamp({ t: 1710527915, i: 1 }), t: Long('1') },
      optimeDate: ISODate('2024-03-15T18:38:35.000Z'),
      optimeDurableDate: ISODate('2024-03-15T18:38:35.000Z'),
      lastAppliedWallTime: ISODate('2024-03-15T18:38:35.036Z'),
      lastDurableWallTime: ISODate('2024-03-15T18:38:35.036Z'),
      lastHeartbeat: ISODate('2024-03-15T18:38:39.277Z'),
      lastHeartbeatRecv: ISODate('2024-03-15T18:38:38.801Z'),
      pingMs: Long('0'),
      lastHeartbeatMessage: '',
      syncSourceHost: 'localhost:1130',
      syncSourceId: 0,
      infoMessage: '',
      configVersion: 1,
      configTerm: 1
    },
    {

```

```

{
  _id: 2,
  name: 'localhost:1150',
  health: 1,
  state: 2,
  stateStr: 'SECONDARY',
  uptime: 576,
  optime: { ts: Timestamp({ t: 1710527915, i: 1 }), t: Long('1') },
  optimeDurable: { ts: Timestamp({ t: 1710527915, i: 1 }), t: Long('1') },
  optimeDate: ISODate('2024-03-15T18:38:35.000Z'),
  optimeDurableDate: ISODate('2024-03-15T18:38:35.000Z'),
  lastAppliedWallTime: ISODate('2024-03-15T18:38:35.036Z'),
  lastDurableWallTime: ISODate('2024-03-15T18:38:35.036Z'),
  lastHeartbeat: ISODate('2024-03-15T18:38:39.275Z'),
  lastHeartbeatRecv: ISODate('2024-03-15T18:38:38.801Z'),
  pingMs: Long('0'),
  lastHeartbeatMessage: '',
  syncSourceHost: 'localhost:1130',
  syncSourceId: 0,
  infoMessage: '',
  configVersion: 1,
  configTerm: 1
}
],
...
},
ok: 1,
'$clusterTime': {
  clusterTime: Timestamp({ t: 1710527915, i: 1 }),
  signature: {
    hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
    keyId: Long('0')
  }
},
operationTime: Timestamp({ t: 1710527915, i: 1 })
}
Server1 [direct: primary] test>

```