

PRACTICAL NO. : 05

AIM: Sharding using MongoDB

STEPS:

**Step1:** create a folder and in that folder make 9 files/folders with name(primary, secondary1, secondary2, server1, server2, server3, shard1, shard2, shard3) and run the commands mentioned below

**Step2:**

replication

open cmd and run commands

cd (path in which you created your folder)

e.g(cd C:\Users\RESHMA KEESARI\OneDrive\Desktop\Pract5)

start mongod --port=50000 --replSet="Server1" --dbpath="(folder path)/primary"

e.g(start mongod --port=50000 --replSet="Server1" --dbpath="C:/Users/RESHMA  
KEESARI/OneDrive/Desktop/Pract5/primary"  
)

"MAKE SURE TO CHANGE PATH"

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

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

open cmd and run

mongosh --port=50000

```
rs.initiate({  
  _id:"Server1",  
  members:[  
    { _id:0, host:"localhost:50000"},  
    { _id:1, host:"localhost:50001"},  
    { _id:2, host:"localhost:50002"}  
  ]  
});
```

**Step3:**

config server

open cmd and run commands

cd (path in which you created your folder)

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

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

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

open cmd and run

mongosh --host="localhost:1030"

```
rs.initiate({  
  _id:"Server1",  
  configsvr:true,  
  members:[  
    { _id:0, host:"localhost:1030"},
```

RESHMA KEESARI

L036

```
    { _id:1, host:"localhost:1040"},  
    { _id:2, host:"localhost:1050"}  
  ]  
})
```

#### Step4:

shards

open cmd and run commands

cd (path in which you created your folder)

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

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

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

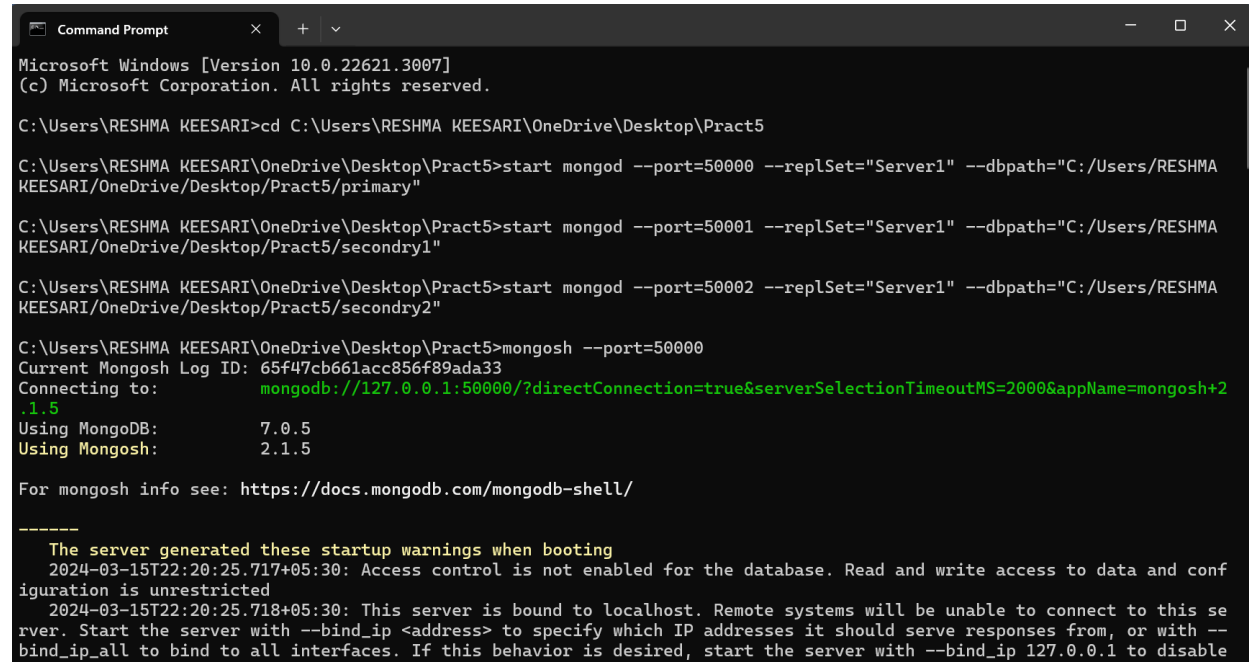
open cmd and run

```
mongosh --host="localhost:1130"
```

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

#### OUTPUT :

#### REPLICATION:



```
Microsoft Windows [Version 10.0.22621.3007]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Users\RESHMA KEESARI>cd C:\Users\RESHMA KEESARI\OneDrive\Desktop\Pract5  
  
C:\Users\RESHMA KEESARI\OneDrive\Desktop\Pract5>start mongod --port=50000 --replSet="Server1" --dbpath="C:/Users/RESHMA  
KEESARI/OneDrive/Desktop/Pract5/primary"  
  
C:\Users\RESHMA KEESARI\OneDrive\Desktop\Pract5>start mongod --port=50001 --replSet="Server1" --dbpath="C:/Users/RESHMA  
KEESARI/OneDrive/Desktop/Pract5/secondary1"  
  
C:\Users\RESHMA KEESARI\OneDrive\Desktop\Pract5>start mongod --port=50002 --replSet="Server1" --dbpath="C:/Users/RESHMA  
KEESARI/OneDrive/Desktop/Pract5/secondary2"  
  
C:\Users\RESHMA KEESARI\OneDrive\Desktop\Pract5>mongosh --port=50000  
Current Mongosh Log ID: 65f47cb661acc856f89ada33  
Connecting to:      mongodb://127.0.0.1:50000/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2  
.1.5  
Using MongoDB:      7.0.5  
Using Mongosh:      2.1.5  
  
For mongosh info see: https://docs.mongodb.com/mongosh-shell/  
  
-----  
The server generated these startup warnings when booting  
2024-03-15T22:20:25.717+05:30: Access control is not enabled for the database. Read and write access to data and conf  
figuration is unrestricted  
2024-03-15T22:20:25.718+05:30: 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
```

```

Server1 [direct: primary] test> rs.initiate({
...   _id:"Server1",
...   members:[
...     {_id:0, host:"localhost:50000"},
...     {_id:1, host:"localhost:50001"},
...     {_id:2, host:"localhost:50002"}
...   ]
... });
MongoServerError[AlreadyInitialized]: already initialized
Server1 [direct: primary] test> rs.status()
{
  set: 'Server1',
  date: ISODate('2024-03-15T16:52:52.124Z'),
  myState: 1,
  term: Long('2'),
  syncSourceHost: '',
  syncSourceId: -1,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1710521572, i: 1 }), t: Long('2') },
    lastCommittedWallTime: ISODate('2024-03-15T16:52:52.063Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1710521572, i: 1 }), t: Long('2') },
    appliedOpTime: { ts: Timestamp({ t: 1710521572, i: 1 }), t: Long('2') },
    durableOpTime: { ts: Timestamp({ t: 1710521572, i: 1 }), t: Long('2') },
    lastAppliedWallTime: ISODate('2024-03-15T16:52:52.063Z'),
    lastDurableWallTime: ISODate('2024-03-15T16:52:52.063Z')
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1710521542, i: 1 }),
  members: [
    {
      _id: 0,
      name: 'localhost:50000',
      health: 1,
      state: 1,
      stateStr: 'PRIMARY',
      uptime: 150,
      optime: { ts: Timestamp({ t: 1710521572, i: 1 }), t: Long('2') },
      optimeDate: ISODate('2024-03-15T16:52:52.000Z'),
      lastAppliedWallTime: ISODate('2024-03-15T16:52:52.063Z'),
      lastDurableWallTime: ISODate('2024-03-15T16:52:52.063Z'),
      syncSourceHost: '',
      syncSourceId: -1,
      infoMessage: 'Could not find member to sync from',
      electionTime: Timestamp({ t: 1710521489, i: 1 }),
      electionDate: ISODate('2024-03-15T16:51:29.000Z'),
      configVersion: 1,
      configTerm: 2,
      self: true,
      lastHeartbeatMessage: ''
    },
  ],
}

```

```

    },
    {
      _id: 1,
      name: 'localhost:50001',
      health: 1,
      state: 2,
      stateStr: 'SECONDARY',
      uptime: 86,
      optime: { ts: Timestamp({ t: 1710521562, i: 1 }), t: Long('2') },
      optimeDurable: { ts: Timestamp({ t: 1710521562, i: 1 }), t: Long('2') },
      optimeDate: ISODate('2024-03-15T16:52:42.000Z'),
      optimeDurableDate: ISODate('2024-03-15T16:52:42.000Z'),
      lastAppliedWallTime: ISODate('2024-03-15T16:52:52.063Z'),
      lastDurableWallTime: ISODate('2024-03-15T16:52:52.063Z'),
      lastHeartbeat: ISODate('2024-03-15T16:52:50.790Z'),
      lastHeartbeatRecv: ISODate('2024-03-15T16:52:51.064Z'),
      pingMs: Long('0'),
      lastHeartbeatMessage: '',
      syncSourceHost: 'localhost:50000',
      syncSourceId: 0,
      infoMessage: '',
      configVersion: 1,
      configTerm: 2
    },
  ],
  ok: 1,
  '$clusterTime': {
    clusterTime: Timestamp({ t: 1710521572, i: 1 }),
    signature: {
      hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
      keyId: Long('0')
    }
  },
  operationTime: Timestamp({ t: 1710521572, i: 1 })
}

```

RESHMA KEESARI

L036

## CONFIG SERVER:

```
mongosh mongodb://localho x + v
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RESHMA KEESARI>mongosh --host="localhost:1030"
Current Mongosh Log ID: 65f47ff5087a36f5083b860b
Connecting to:      mongodb://localhost:1030/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.5
Using MongoDB:      7.0.5
Using Mongosh:      2.1.5
mongosh 2.2.0 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
2024-03-15T22:33:22.890+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2024-03-15T22:33:22.892+05:30: This server is bound to localhost. Remote systems 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, 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
-----

Server1 [direct: primary] test> rs.initiate({
...   _id:"Server1",
...   configsvr:true,
...   members:[
...     { _id:0, host:"localhost:1030"},
...     { _id:1, host:"localhost:1040"},
...     { _id:2, host:"localhost:1050"}
...   ]
... });
MongoServerError[AlreadyInitialized]: already initialized
```

```
mongosh mongodb://localhost × + ▾
Server1 [direct: primary] test> rs.status()
{
  set: 'Server1',
  date: ISODate('2024-03-15T17:06:58.419Z'),
  myState: 1,
  term: Long('2'),
  syncSourceHost: '',
  syncSourceId: -1,
  configsvr: true,
  heartbeatIntervalMillis: Long('2000'),
  majorityVoteCount: 2,
  writeMajorityCount: 2,
  votingMembersCount: 3,
  writableVotingMembersCount: 3,
  optimes: {
    lastCommittedOpTime: { ts: Timestamp({ t: 1710522418, i: 1 }), t: Long('2') },
    lastCommittedWallTime: ISODate('2024-03-15T17:06:58.265Z'),
    readConcernMajorityOpTime: { ts: Timestamp({ t: 1710522418, i: 1 }), t: Long('2') },
    appliedOpTime: { ts: Timestamp({ t: 1710522418, i: 1 }), t: Long('2') },
    durableOpTime: { ts: Timestamp({ t: 1710522418, i: 1 }), t: Long('2') },
    lastAppliedWallTime: ISODate('2024-03-15T17:06:58.265Z'),
    lastDurableWallTime: ISODate('2024-03-15T17:06:58.265Z')
  },
  lastStableRecoveryTimestamp: Timestamp({ t: 1710522382, i: 1 }),
  electionCandidateMetrics: {
    lastElectionReason: 'electionTimeout',
    lastElectionDate: ISODate('2024-03-15T17:05:00.442Z'),
    electionTerm: Long('2'),
    lastCommittedOpTimeAtElection: { ts: Timestamp({ t: 0, i: 0 }), t: Long('-1') },
    lastSeenOpTimeAtElection: { ts: Timestamp({ t: 1709983155, i: 1 }), t: Long('1') },
    numVotesNeeded: 2,
    priorityAtElection: 1,
  },
  members: [
    {
      _id: 0,
      name: 'localhost:1030',
      health: 1,
      state: 1,
      stateStr: 'PRIMARY',
      uptime: 219,
      optime: { ts: Timestamp({ t: 1710522418, i: 1 }), t: Long('2') },
      optimeDate: ISODate('2024-03-15T17:06:58.000Z'),
      lastAppliedWallTime: ISODate('2024-03-15T17:06:58.265Z'),
      lastDurableWallTime: ISODate('2024-03-15T17:06:58.265Z'),
      syncSourceHost: '',
      syncSourceId: -1,
      infoMessage: 'Could not find member to sync from',
      electionTime: Timestamp({ t: 1710522300, i: 1 }),
      electionDate: ISODate('2024-03-15T17:05:00.000Z'),
      configVersion: 1,
      configTerm: 2,
      self: true,
      lastHeartbeatMessage: ''
    },
  ],
}
```



## SHARDS:

```

Command Prompt
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RESHMA KEESARI>cd C:\Users\RESHMA KEESARI\OneDrive\Desktop\Pract5

C:\Users\RESHMA KEESARI\OneDrive\Desktop\Pract5>mongod --shardsvr --port=1130 --dbpath="C:/Users/RESHMA KEESARI/OneDrive/Desktop/Pract5/server1" --replSet="Server1"

```

```

Command Prompt
C:\Users\RESHMA KEESARI\OneDrive\Desktop\Pract5>mongod --shardsvr --port=1140 --dbpath="C:/Users/RESHMA KEESARI/OneDrive/Desktop/Pract5/server2" --replSet="Server1"

```

```

Command Prompt
C:\Users\RESHMA KEESARI\OneDrive\Desktop\Pract5>mongod --shardsvr --port=1150 --dbpath="C:/Users/RESHMA KEESARI/OneDrive/Desktop/Pract5/server3" --replSet="Server1"

```

```

mongosh mongodb://localhost
Microsoft Windows [Version 10.0.22621.3007]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RESHMA KEESARI>mongosh --host="localhost:1130"
Current Mongosh Log ID: 65f4834badd4cbca8670d4d8
Connecting to:      mongodb://localhost:1130/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.5
Using MongoDB:      7.0.5
Using Mongosh:      2.1.5
mongosh 2.2.0 is available for download: https://www.mongodb.com/try/download/shell
For mongosh info see: https://docs.mongodb.com/mongosh-shell/

-----
The server generated these startup warnings when booting
2024-03-15T22:46:16.955+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2024-03-15T22:46:16.958+05:30: This server is bound to localhost. Remote systems 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, 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
-----

Server1 [direct: primary] test> rs.initiate({
...   _id: "Server1",
...   members: [
...     { _id: 0, host: "localhost:1130" },
...     { _id: 1, host: "localhost:1140" },
...     { _id: 2, host: "localhost:1150" }
...   ]
... });
MongoServerError[AlreadyInitialized]: already initialized

```

```

members: [
  {
    _id: 0,
    name: 'localhost:1130',
    health: 1,
    state: 1,
    stateStr: 'PRIMARY',
    uptime: 278,
    optime: { ts: Timestamp({ t: 1710523247, i: 1 }), t: Long('2') },
    optimeDate: ISODate('2024-03-15T17:20:47.000Z'),
    lastAppliedWallTime: ISODate('2024-03-15T17:20:47.155Z'),
    lastDurableWallTime: ISODate('2024-03-15T17:20:47.155Z'),
    syncSourceHost: '',
    syncSourceId: -1,
    infoMessage: 'Could not find member to sync from',
    electionTime: Timestamp({ t: 1710523151, i: 1 }),
    electionDate: ISODate('2024-03-15T17:19:11.000Z'),
    configVersion: 1,
    configTerm: 2,
    self: true,
    lastHeartbeatMessage: ''
  },
]

```



```

{
  _id: 1,
  name: 'localhost:1140',
  health: 1,
  state: 2,
  stateStr: 'SECONDARY',
  uptime: 102,
  optime: { ts: Timestamp({ t: 1710523247, i: 1 }), t: Long('2') },
  optimeDurable: { ts: Timestamp({ t: 1710523247, i: 1 }), t: Long('2') },
  optimeDate: ISODate('2024-03-15T17:20:47.000Z'),
  optimeDurableDate: ISODate('2024-03-15T17:20:47.000Z'),
  lastAppliedWallTime: ISODate('2024-03-15T17:20:47.155Z'),
  lastDurableWallTime: ISODate('2024-03-15T17:20:47.155Z'),
  lastHeartbeat: ISODate('2024-03-15T17:20:49.978Z'),
  lastHeartbeatRecv: ISODate('2024-03-15T17:20:49.920Z'),
  pingMs: Long('0'),
  lastHeartbeatMessage: '',
  syncSourceHost: 'localhost:1130',
  syncSourceId: 0,
  infoMessage: '',
  configVersion: 1,
  configTerm: 2
},
{
  _id: 2,
  name: 'localhost:1150',
  health: 1,
  state: 2,
  stateStr: 'SECONDARY',
  uptime: 55,
  optime: { ts: Timestamp({ t: 1710523247, i: 1 }), t: Long('2') },
  optimeDurable: { ts: Timestamp({ t: 1710523247, i: 1 }), t: Long('2') },
  optimeDate: ISODate('2024-03-15T17:20:47.000Z'),
  optimeDurableDate: ISODate('2024-03-15T17:20:47.000Z'),
  lastAppliedWallTime: ISODate('2024-03-15T17:20:47.155Z'),
  lastDurableWallTime: ISODate('2024-03-15T17:20:47.155Z'),
  lastHeartbeat: ISODate('2024-03-15T17:20:49.787Z'),
  lastHeartbeatRecv: ISODate('2024-03-15T17:20:50.368Z'),
  pingMs: Long('0'),
  lastHeartbeatMessage: '',
  syncSourceHost: 'localhost:1140',
  syncSourceId: 1,
  infoMessage: '',
  configVersion: 1,
  configTerm: 2
}
],
ok: 1,
'$clusterTime': {
  clusterTime: Timestamp({ t: 1710523247, i: 1 }),
  signature: {
    hash: Binary.createFromBase64('AAAAAAAAAAAAAAAAAAAAAAAAA=', 0),
    keyId: Long('0')
  }
},
operationTime: Timestamp({ t: 1710523247, i: 1 })

```