Step 1: Create Directories

powershell

Copy

Download

# Config Server

mkdir C:\data\cfg1, C:\data\cfg2, C:\data\cfg3

# Shard Servers (2 shards, 3 nodes each)

mkdir C:\data\shard1\_1, C:\data\shard1\_2, C:\data\shard1\_3

mkdir C:\data\shard2\_1, C:\data\shard2\_2, C:\data\shard2\_3

# Mongos router

mkdir C:\data\mongos

o/p:

Directory: C:\data

Mode LastWriteTime Length Name

---- ------------- ------ ----

d----- 13-Jun-25 6:35 PM cfg1

d----- 13-Jun-25 6:35 PM cfg2

d----- 13-Jun-25 6:35 PM cfg3

Directory: C:\data

Mode LastWriteTime Length Name

---- ------------- ------ ----

d----- 13-Jun-25 6:35 PM shard1\_1

d----- 13-Jun-25 6:35 PM shard1\_2

d----- 13-Jun-25 6:35 PM shard1\_3

Directory: C:\data

Mode LastWriteTime Length Name

---- ------------- ------ ----

d----- 13-Jun-25 6:35 PM shard2\_1

d----- 13-Jun-25 6:35 PM shard2\_2

d----- 13-Jun-25 6:35 PM shard2\_3

Directory: C:\data

Mode LastWriteTime Length Name

---- ------------- ------ ----

d----- 13-Jun-25 6:35 PM mongos

Step 2: Start Config Server Replica Set

Run in 3 separate PowerShell windows:

powershell

Copy

Download

# Config Server 1

mongod --configsvr --replSet configReplSet --port 27019 --dbpath C:\data\cfg1 --bind\_ip localhost

# Config Server 2

mongod --configsvr --replSet configReplSet --port 27020 --dbpath C:\data\cfg2 --bind\_ip localhost

# Config Server 3

mongod --configsvr --replSet configReplSet --port 27021 --dbpath C:\data\cfg3 --bind\_ip localhost

Initialize the config replica set:

powershell

Copy

Download

mongo --port 27019

javascript

Copy

Download

rs.initiate({

\_id: "configReplSet",

configsvr: true,

members: [

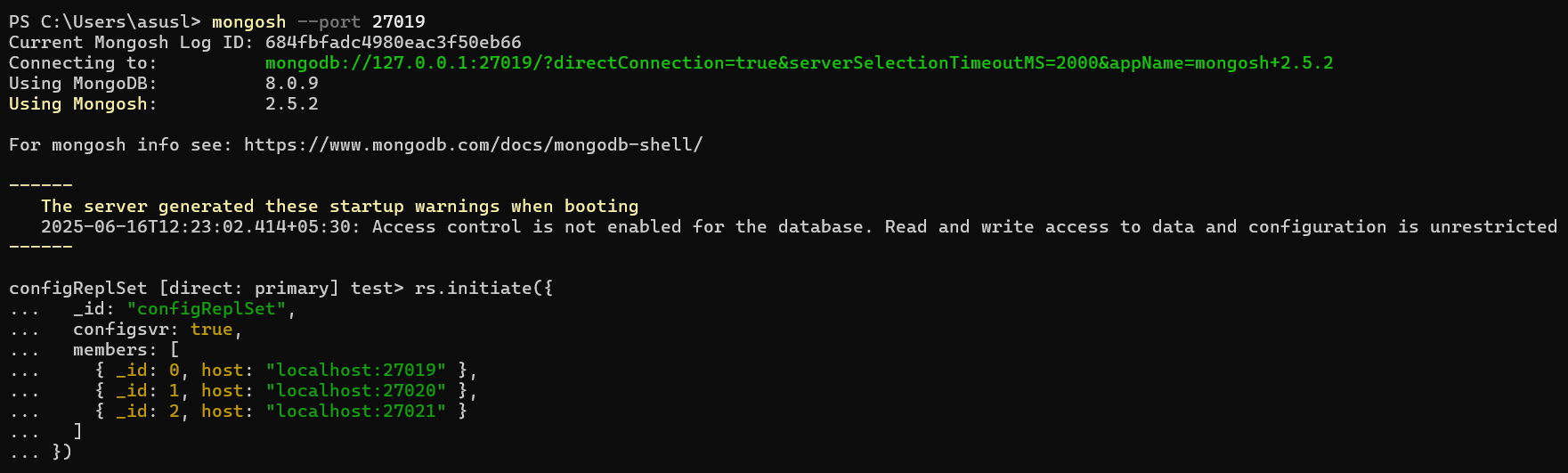
{ \_id: 0, host: "localhost:27019" },

{ \_id: 1, host: "localhost:27020" },

{ \_id: 2, host: "localhost:27021" }

]

})



Step 3: Start Shard Replica Sets

Shard 1 (3 nodes in separate PowerShell windows):

powershell

Copy

Download

# Shard1 Node1

mongod --shardsvr --replSet shard1ReplSet --port 27022 --dbpath C:\data\shard1\_1 --bind\_ip localhost

# Shard1 Node2

mongod --shardsvr --replSet shard1ReplSet --port 27023 --dbpath C:\data\shard1\_2 --bind\_ip localhost

# Shard1 Node3

mongod --shardsvr --replSet shard1ReplSet --port 27024 --dbpath C:\data\shard1\_3 --bind\_ip localhost

Initialize Shard 1:

powershell

Copy

Download

mongo --port 27022

javascript

Copy

Download

rs.initiate({

\_id: "shard1ReplSet",

members: [

{ \_id: 0, host: "localhost:27022" },

{ \_id: 1, host: "localhost:27023" },

{ \_id: 2, host: "localhost:27024" }

]

})

Shard 2 (3 nodes in separate PowerShell windows):

powershell

Copy

Download

# Shard2 Node1

mongod --shardsvr --replSet shard2ReplSet --port 27025 --dbpath C:\data\shard2\_1 --bind\_ip localhost

# Shard2 Node2

mongod --shardsvr --replSet shard2ReplSet --port 27026 --dbpath C:\data\shard2\_2 --bind\_ip localhost

# Shard2 Node3

mongod --shardsvr --replSet shard2ReplSet --port 27027 --dbpath C:\data\shard2\_3 --bind\_ip localhost

Initialize Shard 2:

powershell

Copy

Download

mongo --port 27025

javascript

Copy

Download

rs.initiate({

\_id: "shard2ReplSet",

members: [

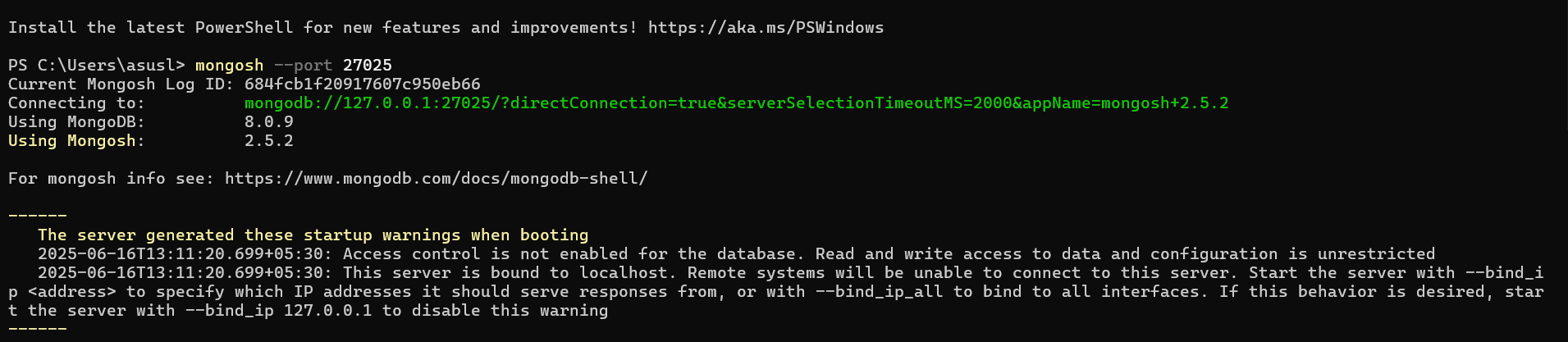
{ \_id: 0, host: "localhost:27025" },

{ \_id: 1, host: "localhost:27026" },

{ \_id: 2, host: "localhost:27027" }

]

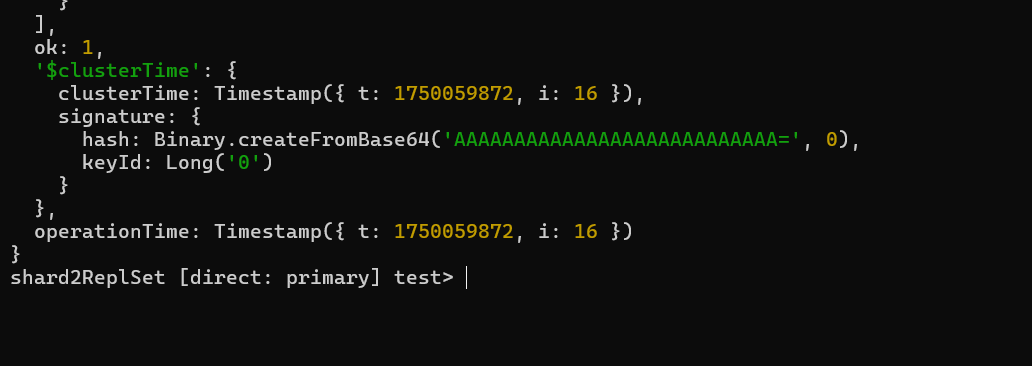
})











Step 4: Start Mongos Query Router

powershell

Copy

Download

mongos --configdb configReplSet/localhost:27019,localhost:27020,localhost:27021 --port 27018 --bind\_ip localhost

Step 5: Configure Sharding

Connect to mongos:

powershell

Copy

Download

mongo --port 27018

Add shards to cluster:

javascript

Copy

Download

sh.addShard("shard1ReplSet/localhost:27022,localhost:27023,localhost:27024")

sh.addShard("shard2ReplSet/localhost:27025,localhost:27026,localhost:27027")



Step 6: Enable Sharding for a Database and Collection

javascript

Copy

Download

// Enable sharding for a database

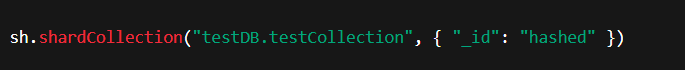
sh.enableSharding("testDB")

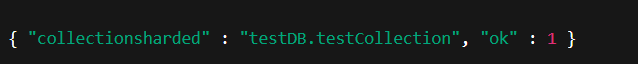
// Shard a collection with hashed shard key

sh.shardCollection("testDB.testCollection", { "\_id": "hashed" })

// Or with ranged sharding

sh.shardCollection("testDB.testCollection", { "userId": 1 })





Step 7: Verify Sharding Status

javascript

Copy

Download

sh.status()

Connecting from MongoDB Compass

Use this connection string:

text

Copy

Download

mongodb://localhost:27018/

Real-Time Data Insertion Test

javascript

Copy

Download

// Connect to mongos

use testDB

// Insert sample data (will distribute across shards)

for (let i = 0; i < 10000; i++) {

db.testCollection.insert({

userId: i,

data: "Sample data " + i,

timestamp: new Date()

})

}

// Check distribution

db.testCollection.getShardDistribution()



