

# MONGODB SHARDING SETUP GUIDE

## Chapter 8: Sharding in MongoDB Efficiently Scaling and Managing Large Datasets

Instructor: Houssein Dhayne ([houssein.dhayne@net.usj.edu.lb](mailto:houssein.dhayne@net.usj.edu.lb))

### INTRODUCTION

In this tutorial, you will learn how to work with MongoDB, a popular NoSQL database. We will cover the basics of creating a collection, configuring and starting MongoDB instances, initializing replica sets, setting up sharding, and finally, sharding a collection based on date ranges.

### SETUP GUIDE

#### Step 1: Download and Install MongoDB and Tools

To get started, please download and install the following applications:

- 1.1. MongoDB (Database Server):  
Download: `mongodb-windows-x86_64-7.0.5`
- 1.2. MongoDB Shell (Command-Line Interface):  
Download: `mongosh-2.1.1`
- 1.3. NoSQLBooster (MongoDB GUI Tool):  
Download: `nosqlbooster4mongo-8.1.6`

Once downloaded, follow the installation instructions for each application. This will set up MongoDB, the MongoDB Shell (mongosh), and NoSQLBooster, providing you with a comprehensive environment for working with MongoDB databases.

#### Step 2: Create Directories for MongoDB Instances

- 2.1. Create separate directories for each MongoDB instance, e.g., `shard1`, `shard2`, `shard3`, and `master`.

### Step 3: Create Configuration Files

#### 3.1. Create configuration files for each MongoDB instance:

##### mongod\_config\_master.conf:

```
systemLog:
  destination: file
  path: "E:\\courses\\shards\\master\\mongod.log"
  logAppend: true
storage:
  dbPath: "E:\\courses\\shards\\master\\data"
net:
  bindIp: 127.0.0.1
  port: 27010
sharding:
  clusterRole: configsvr
replication:
  replSetName: "configReplSet"
```

##### mongod\_config\_shard1.conf:

```
systemLog:
  destination: file
  path: "E:\\courses\\shards\\shard1\\mongod.log"
  logAppend: true
storage:
  dbPath: "E:\\courses\\shards\\shard1\\data"
net:
  bindIp: 127.0.0.1
  port: 27021
replication:
  replSetName: "shard1"
```

```
sharding:

  clusterRole: shardsvr
```

3.2. Repeat the above for shard2 and shard3, adjusting the paths, dbPaths, and ports accordingly.

#### Step 4: Start MongoDB Instances

4.1. Open command prompt windows and start each MongoDB instance:

```
mongod --config E:\courses\shards\master\mongod_config_master.conf --replSet configReplSet
mongod --config E:\courses\shards\shard1\mongod_config_shard.conf --replSet shard1
mongod --config E:\courses\shards\shard2\mongod_config_shard.conf --replSet shard2
mongod --config E:\courses\shards\shard3\mongod_config_shard.conf --replSet shard3
```

#### Step 5: Connect to MongoDB Instances and Initialize Replica Sets

5.1. Open a new command prompt window and connect to each MongoDB instance using the MongoDB shell:

```
mongosh --port 27010
```

5.2. In the MongoDB shell, initiate the replica sets:

For the master (config server):

```
rs.initiate();
```

For shard1:

```
mongosh --port 27021
rs.initiate({_id: "shard1", members: [{_id: 0, host: "localhost:27021"}]});
```

Repeat the above for shard2 and shard3, changing the port accordingly.

#### Step 6: Start MongoDB Router (mongos)

6.1. Open a new command prompt window and start the MongoDB router (mongos):

```
mongos --configdb configReplSet/localhost:27010 --port 27011
mongosh --port 27011
```

6.2. Once the replica set is correctly configured, go back to the mongos instance and try adding the shard again:

```
use mytest

sh.addShard("shard1/localhost:27021");

sh.addShard("shard2/localhost:27022");

sh.addShard("shard3/localhost:27023");
```

### Step 7: Choose a Sharding Key

7.1. Let's assume you have a "date" field in your "Product" collection. Choose this field as the sharding key.

### Step 8: Create Index on Sharding Key

8.1. Create an index on the "date" field. This index is required for sharding:

```
db.Product.createIndex({ "date": 1 });
```

### Step 9: Enable Sharding for Database

9.1. Enable sharding for your database if you haven't already:

```
sh.enableSharding("mytest");
```

### Step 10: Shard the Collection Based on Date Ranges

10.1. Shard the "Product" collection based on date ranges. Create three date range shards:

```
sh.shardCollection("mytest.Product", { "date": 1 });

sh.addShardTag("shard1", "tag1");

sh.addShardTag("shard2", "tag2");

sh.addShardTag("shard3", "tag3");

sh.addTagRange("mytest.Product", { "date": ISODate("2024-01-01") }, { "date":
ISODate("2024-01-31") }, "tag1");
```

```
sh.addTagRange("mytest.Product", { "date": ISODate("2024-02-01") }, { "date":  
ISODate("2024-02-28") }, "tag2");  
  
sh.addTagRange("mytest.Product", { "date": ISODate("2024-03-01") }, { "date":  
ISODate("2024-03-31") }, "tag3");
```

### Step 11: Insert Sample Documents into Sharded "Product" Collection

Now that we have set up our sharded environment and configured date range shards, let's proceed to insert sample documents into the sharded "Product" collection. We will consider the date ranges we've defined for each shard.

// Insert documents into shard1 (1-1-2024 to 31-1-2024)

```
db.Product.insertMany([  
  { name: "Product1", description: "Description1", price: 10, date: ISODate("2024-01-15")  
},  
  { name: "Product2", description: "Description2", price: 20, date: ISODate("2024-01-25")  
},  
  { name: "Product3", description: "Description3", price: 15, date: ISODate("2024-01-05")  
}  
]);
```

// Insert documents into shard2 (1-2-2024 to 28-2-2024)

```
db.Product.insertMany([  
  { name: "Product4", description: "Description4", price: 30, date: ISODate("2024-02-10")  
},  
  { name: "Product5", description: "Description5", price: 25, date: ISODate("2024-02-20")  
},  
  { name: "Product6", description: "Description6", price: 18, date: ISODate("2024-02-05")  
}  
]);
```

// Insert documents into shard3 (1-3-2024 to 31-3-2024)

```
db.Product.insertMany([  
  { name: "Product7", description: "Description7", price: 40, date: ISODate("2024-03-15")  
},  
]);
```

```
{ name: "Product8", description: "Description8", price: 35, date: ISODate("2024-03-25")
},
{ name: "Product9", description: "Description9", price: 28, date: ISODate("2024-03-05")
}
]);
```