Week-4 Day-4 Mentor Material

### **Assignment: Teaching Sharding and Replication in MongoDB**

#### **Objective:**

The objective of this assignment is to guide learners through setting up MongoDB sharding and replication. This will involve creating a sharded cluster, setting up replication, and performing operations on a dataset to observe the behavior of the sharded and replicated environment.

#### **Prerequisites:**

* MongoDB installed on multiple instances or machines (minimum 4 instances for this exercise).
* Basic understanding of MongoDB operations and configurations.
* Python installed for generating the dataset. ( If you are using this method)

#### **Steps:**

### **Part 1: Setting Up Replication**

**Start MongoDB Instances** Start three MongoDB instances on separate ports (e.g., 27017, 27018, 27019). Use the following commands:  
  
mongod --replSet rs0 --port 27017 --dbpath /data/db1 --bind\_ip localhost

mongod --replSet rs0 --port 27018 --dbpath /data/db2 --bind\_ip localhost

mongod --replSet rs0 --port 27019 --dbpath /data/db3 --bind\_ip localhost

**Initialize the Replica Set** Connect to one of the instances and initiate the replica set:  
  
mongo --port 27017

In the MongoDB shell, run:  
  
rs.initiate()

rs.add("localhost:27018")

rs.add("localhost:27019")

**Verify Replica Set** Check the status of the replica set:  
  
rs.status()

### **Part 2: Setting Up Sharding**

**Start Config Server** Start a config server for the sharded cluster:  
  
mongod --configsvr --replSet configReplSet --port 27019 --dbpath /data/configdb --bind\_ip localhost

**Initialize Config Server Replica Set** Connect to the config server instance and initiate the replica set:  
  
mongo --port 27019

In the MongoDB shell, run:  
  
rs.initiate()

**Start Shard Servers** Start three shard servers on separate ports:  
  
mongod --shardsvr --replSet shardReplSet1 --port 27020 --dbpath /data/shard1 --bind\_ip localhost

mongod --shardsvr --replSet shardReplSet2 --port 27021 --dbpath /data/shard2 --bind\_ip localhost

mongod --shardsvr --replSet shardReplSet3 --port 27022 --dbpath /data/shard3 --bind\_ip localhost

**Initialize Shard Server Replica Sets** Connect to each shard server instance and initiate the replica sets:  
  
mongo --port 27020

rs.initiate()

1. Repeat for the other shard servers (27021 and 27022).

**Start MongoS Router** Start a MongoS router to route queries to the shards:  
  
mongos --configdb configReplSet/localhost:27019 --port 27023 --bind\_ip localhost

**Add Shards to the Cluster** Connect to the MongoS instance and add the shards:  
  
mongo --port 27023

In the MongoDB shell, run:  
  
sh.addShard("shardReplSet1/localhost:27020")

sh.addShard("shardReplSet2/localhost:27021")

sh.addShard("shardReplSet3/localhost:27022")

### **Part 3: Using a Dataset**

You may use another way to store the data .

**Create a Dataset** Use the following Python script to generate a sample dataset:  
  
import json

import random

import names

data = []

for i in range(1000):

record = {

"name": names.get\_full\_name(),

"age": random.randint(18, 70),

"gender": random.choice(["M", "F"]),

"address": {

"city": random.choice(["New York", "Los Angeles", "Chicago", "Houston", "Phoenix"]),

"state": random.choice(["NY", "CA", "IL", "TX", "AZ"]),

"zip": random.randint(10000, 99999)

},

"employment": {

"company": random.choice(["Company A", "Company B", "Company C"]),

"position": random.choice(["Engineer", "Manager", "Director", "Analyst"]),

"salary": random.randint(40000, 120000)

}

}

data.append(record)

with open("employees.json", "w") as f:

json.dump(data, f)

**Import Dataset into MongoDB** Use the following command to import the dataset into MongoDB:  
  
mongoimport --host localhost --port 27023 --db company --collection employees --file employees.json

**Enable Sharding on the Database and Collection** Connect to the MongoS instance and enable sharding:  
  
mongo --port 27023

In the MongoDB shell, run:  
  
sh.enableSharding("company")

sh.shardCollection("company.employees", {"address.state": 1})

### **Part 4: Performing Operations**

**Insert Data** Insert a new document into the employees collection:  
  
db.employees.insertOne({

"name": "John Doe",

"age": 45,

"gender": "M",

"address": {

"city": "Miami",

"state": "FL",

"zip": 33101

},

"employment": {

"company": "Company D",

"position": "Consultant",

"salary": 90000

}

})

**Read Data** Find all employees who work in Company A:  
  
db.employees.find({"employment.company": "Company A"})

**Update Data** Update the salary of an employee named John Doe:  
  
db.employees.updateOne({"name": "John Doe"}, {"$set": {"employment.salary": 95000}})

**Delete Data** Delete an employee record by name:  
  
db.employees.deleteOne({"name": "John Doe"})

### **Conclusion**

This assignment covers the complete process of setting up sharding and replication in MongoDB, from installing MongoDB instances to performing CRUD operations on a sharded and replicated database. This hands-on experience will help learners understand the concepts and practical implementation of sharding and replication in MongoDB.