**Advance Database Management System**

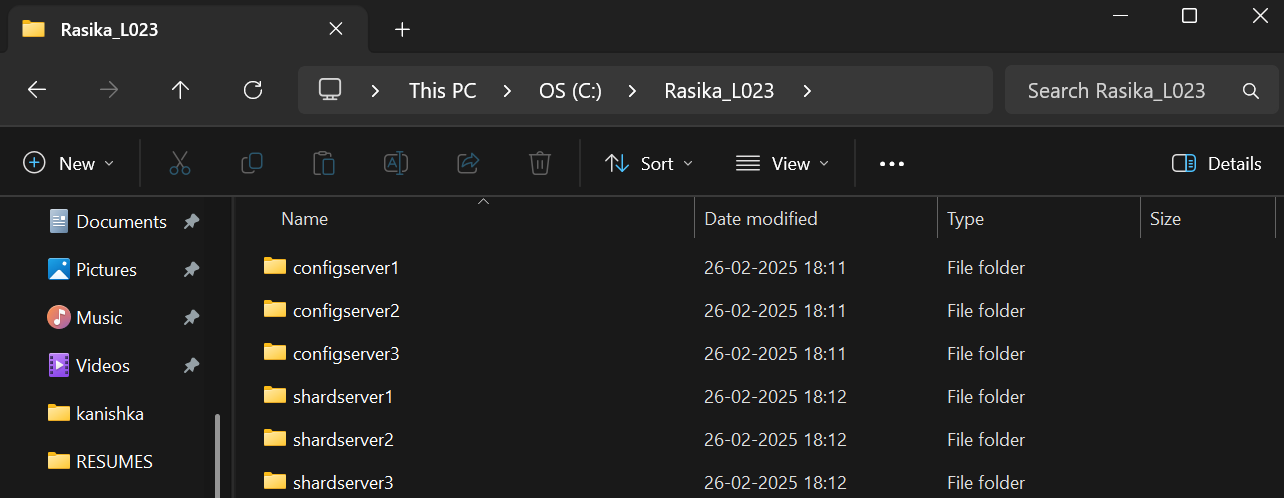
**Practical 7: Sharding in MongoDB**

You are a database administrator tasked with setting up MongoDB sharding to handle a growing dataset. You need to configure a MongoDB sharded cluster with the following requirements:

* One Config Server (Replica set)
* Two Shard Servers
* One Mongos Router
* Sharding enabled for a database and collection

1. **Create Directories:**

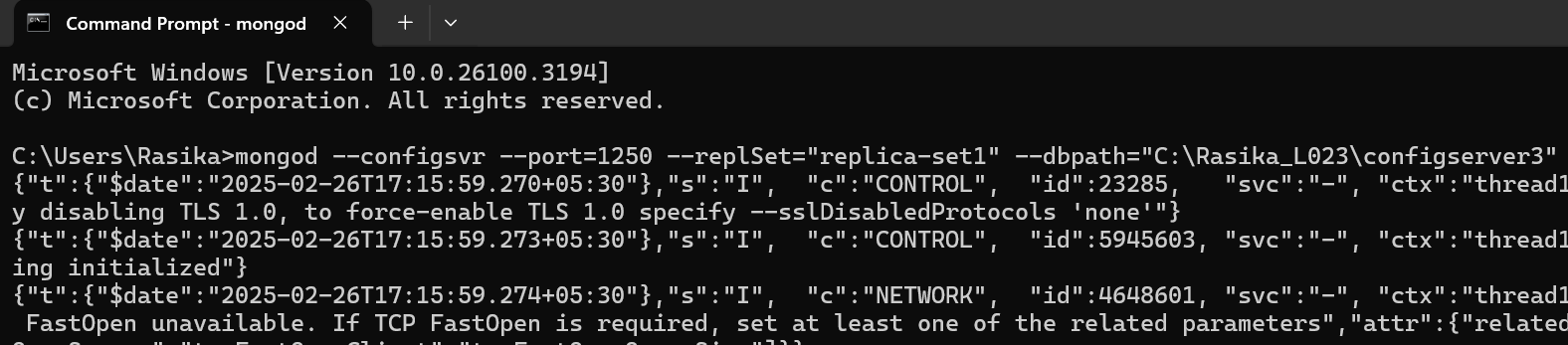
* Create the following directories.



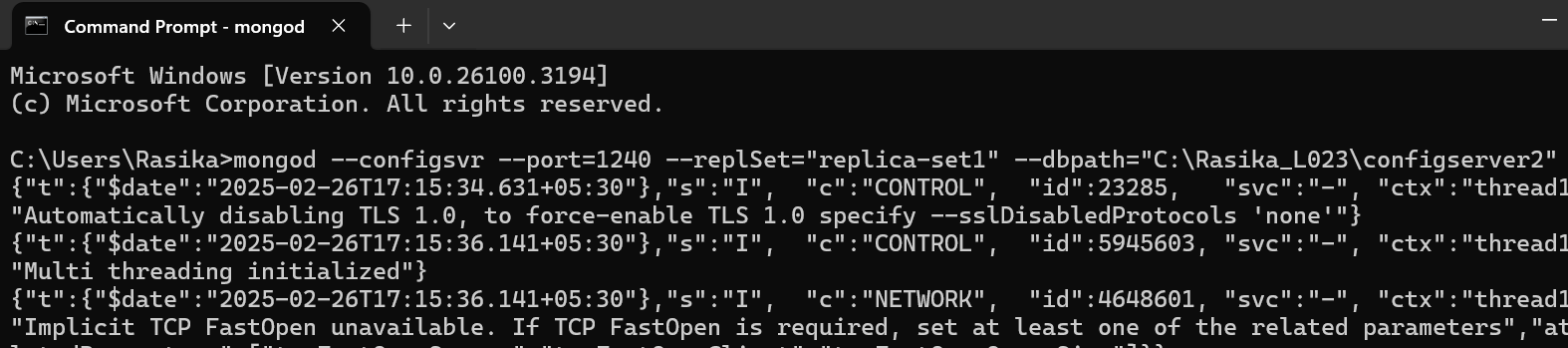
1. **Configure and Start Config Servers:**

* **Start the config servers:** Open three separate command prompt/terminal windows. Run these commands in each:

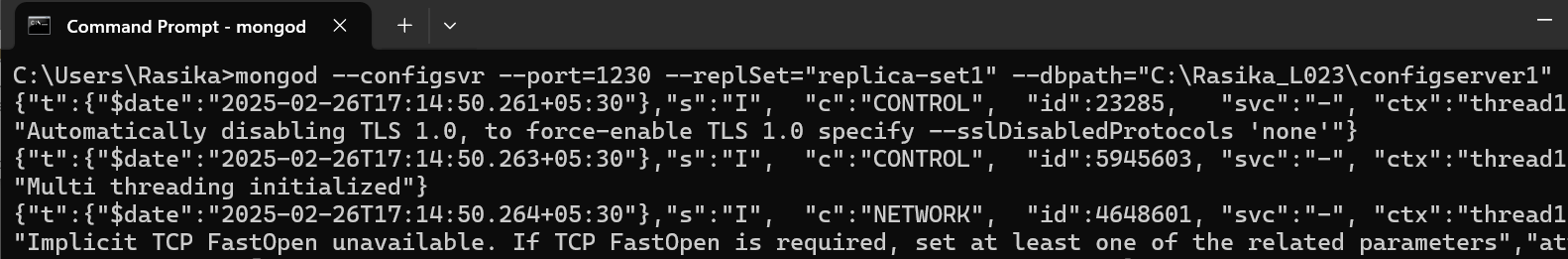
1. mongod --configsvr --port=1230 --replSet="replica-set1" --dbpath="C:\Rasika\_L023\configserver1"



1. mongod --configsvr --port=1240 --replSet="replica-set1" --dbpath="C:\Rasika\_L023\configserver2"

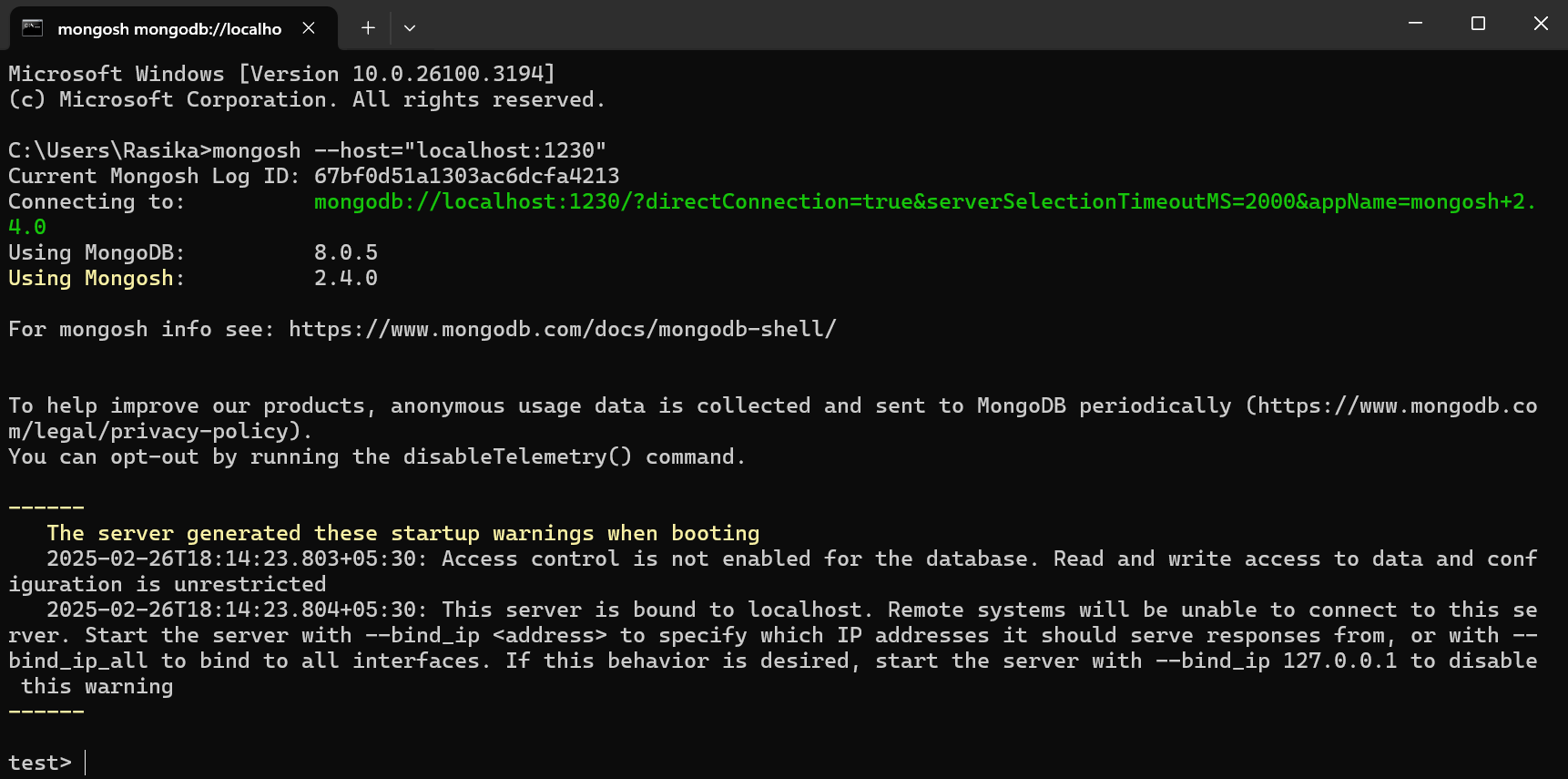


1. mongod --configsvr --port=1250 --replSet="replica-set1" --dbpath="C:\Rasika\_L023\configserver3"



**3. Connect to the first config server:**

* Open another command prompt/terminal and connect using mongosh:



4. **Initiate the config server replica set:**

Run the following command in the mongosh session:

rs.initiate({

\_id:"replica-set1",

configsvr:true,

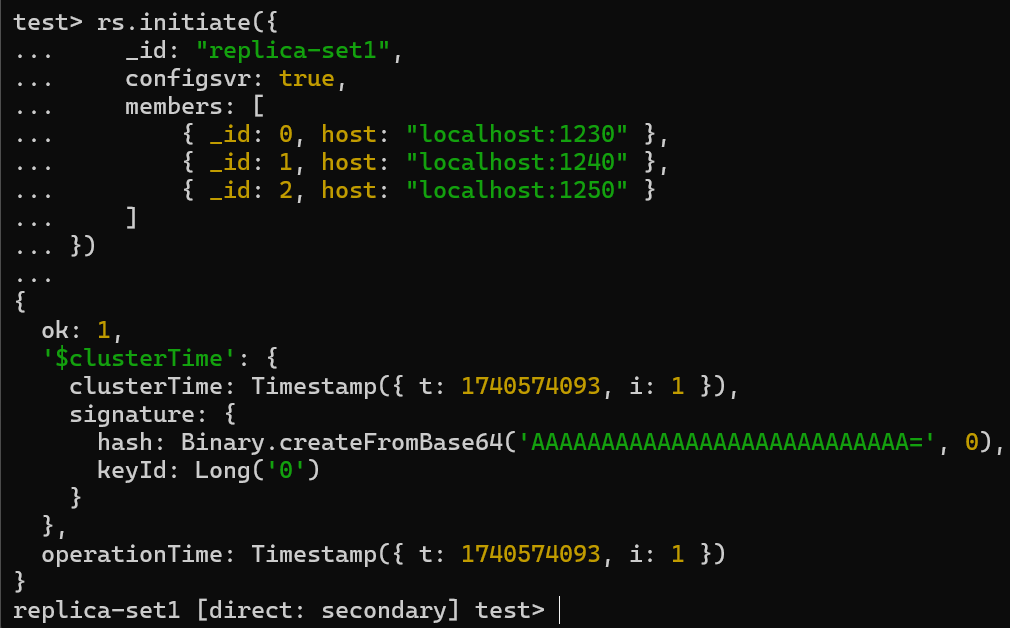
members:[

{\_id:0, host:"localhost:1230"},

{\_id:1, host:"localhost:1240"},

{\_id:2, host:"localhost:1250"}]

})

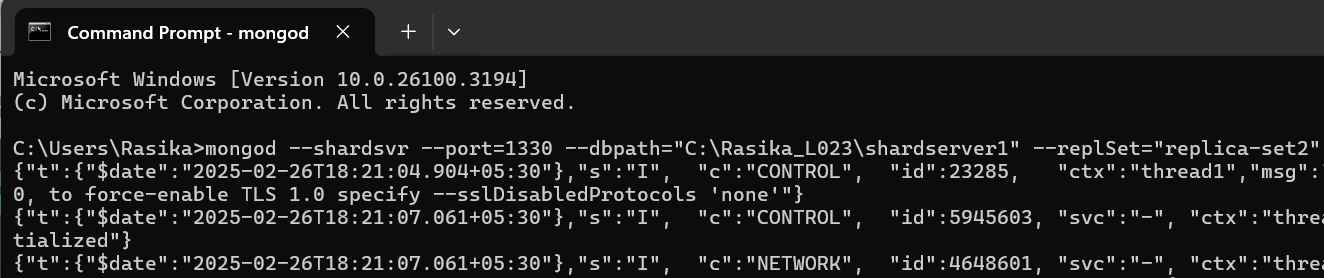


5. **Configure and Start Shard Servers:**

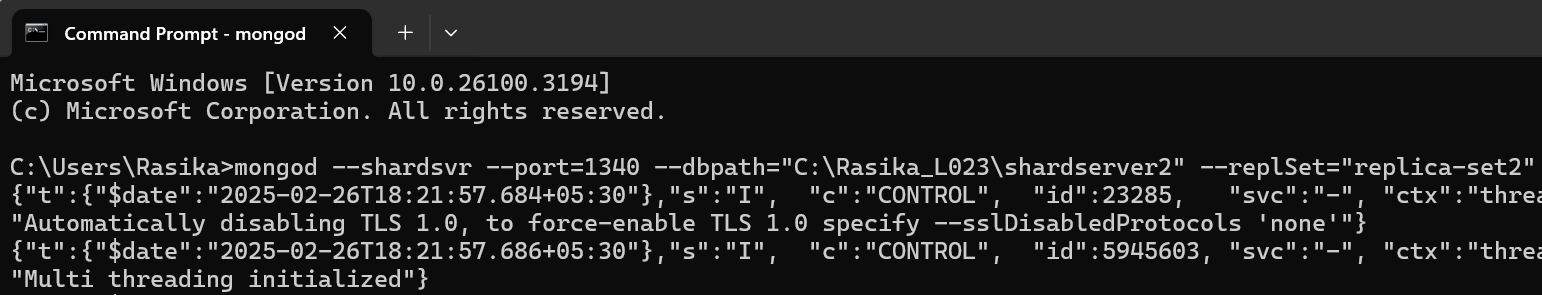
* **Start the shard servers:**

Open three new command prompt/terminal windows and run these commands:

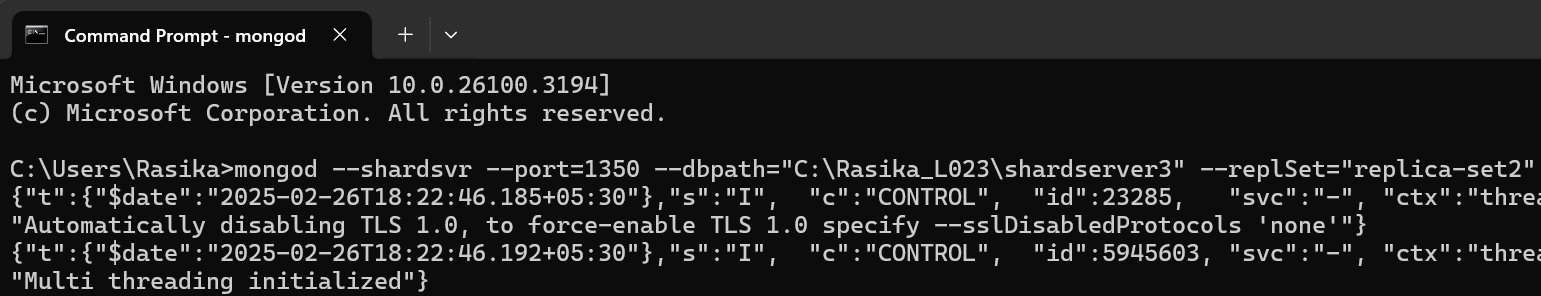
mongod --shardsvr --port=1330 --dbpath="C:\Rasika\_L023\shardserver1" --replSet="replica-set2"



mongod --shardsvr --port=1340 --dbpath="C:\Rasika\_L023\shardserver2" --replSet="replica-set2"

****

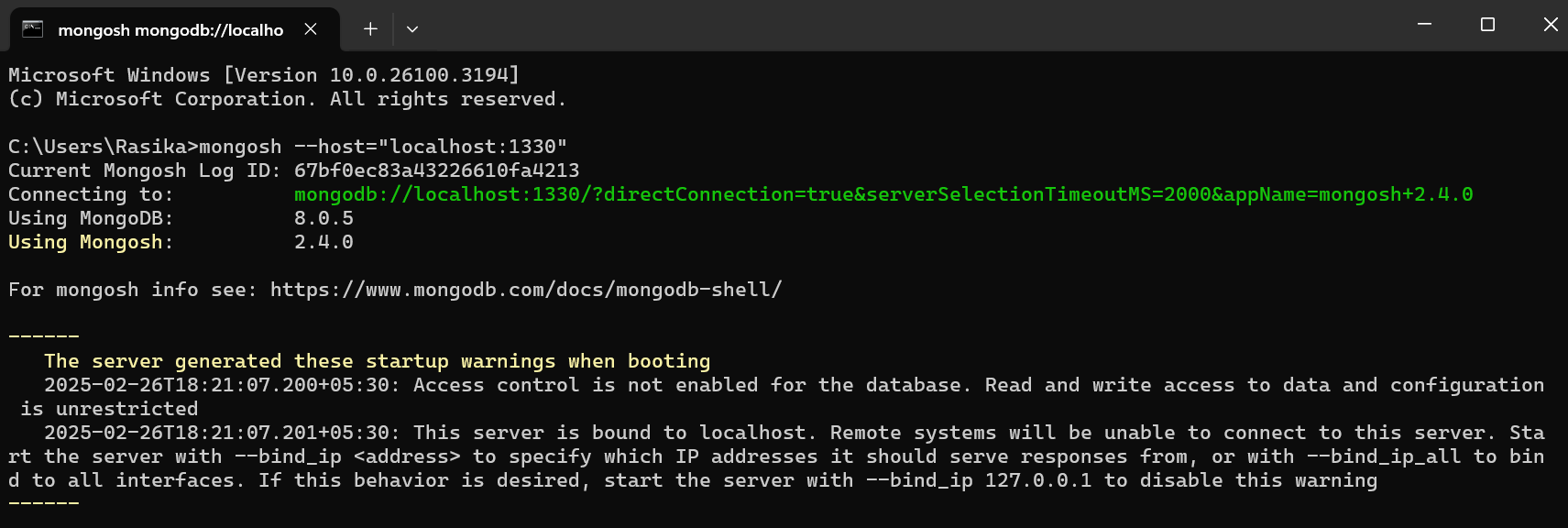
mongod --shardsvr --port=1350 --dbpath="C:\Rasika\_L023\shardserver3" --replSet="replica-set2"



**6.** **Connect to the first shard server:**

* Open another command prompt/terminal and connect using mongosh:

mongosh --host="localhost:1330"



**7. Initiate the shard server replica set:**

* Run the following in the mongosh session:

rs.initiate({

\_id:"replica-set2",

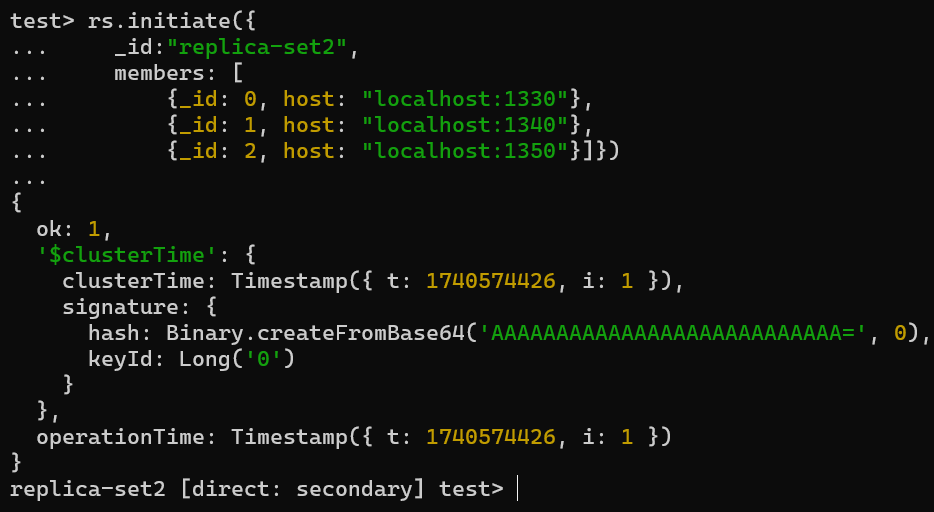
members: [

{\_id: 0, host: "localhost:1330"},

{\_id: 1, host: "localhost:1340"},

{\_id: 2, host: "localhost:1350"}]

})

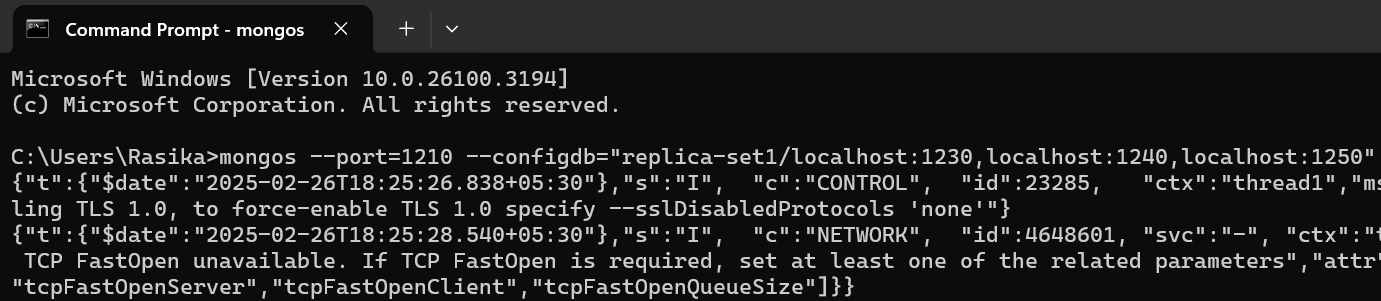


**8. Configure and Start the Query Router (mongos)**

* **Start the mongos router:**

Open a new command prompt/terminal and run:

mongos --port=1210 --configdb="replica-set1/localhost:1230,localhost:1240,localhost:1250"

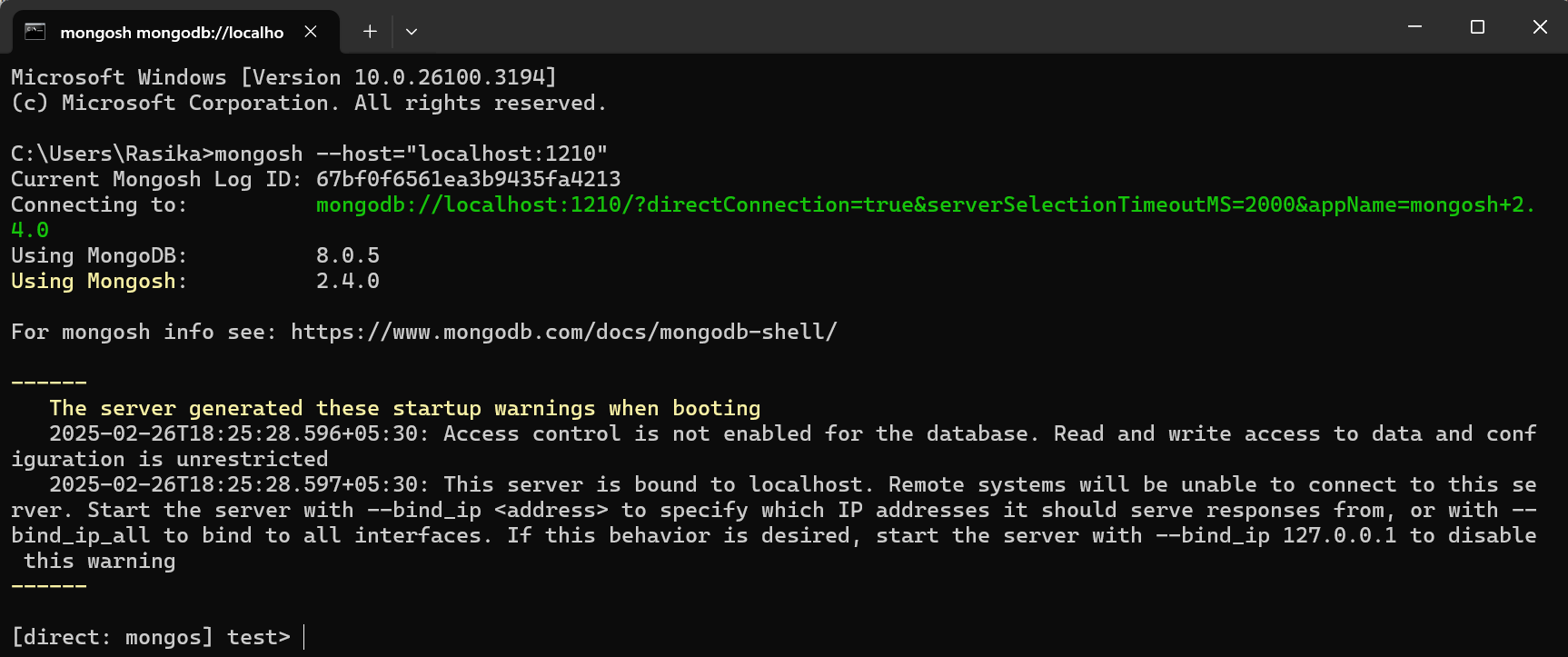


**9.** **Connect to the Sharded Cluster and Add the Shard**

* **Connect to the mongos router:**

Open a new command prompt/terminal and connect using mongosh:

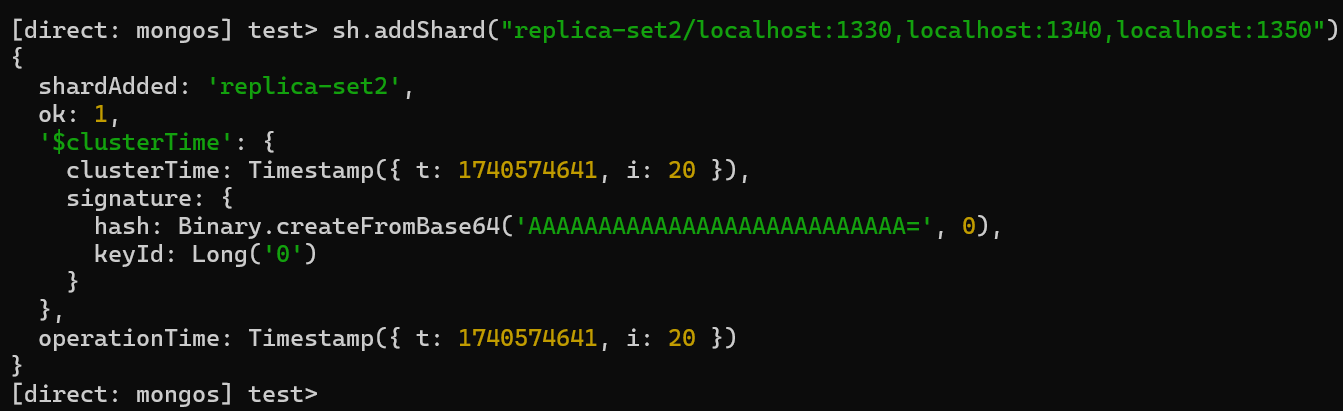
mongosh --host="localhost:1210"



**10. Add the shard to the cluster:**

Run the following command in the mongosh session:

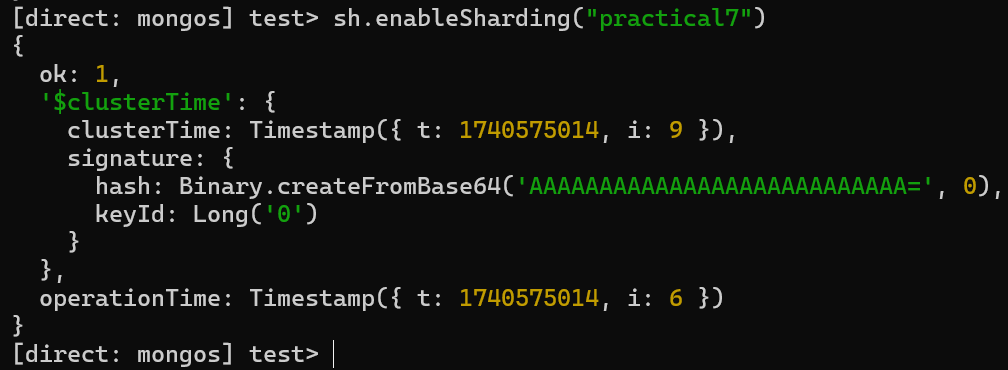
sh.addShard("replica-set2/localhost:1330,localhost:1340,localhost:1350")

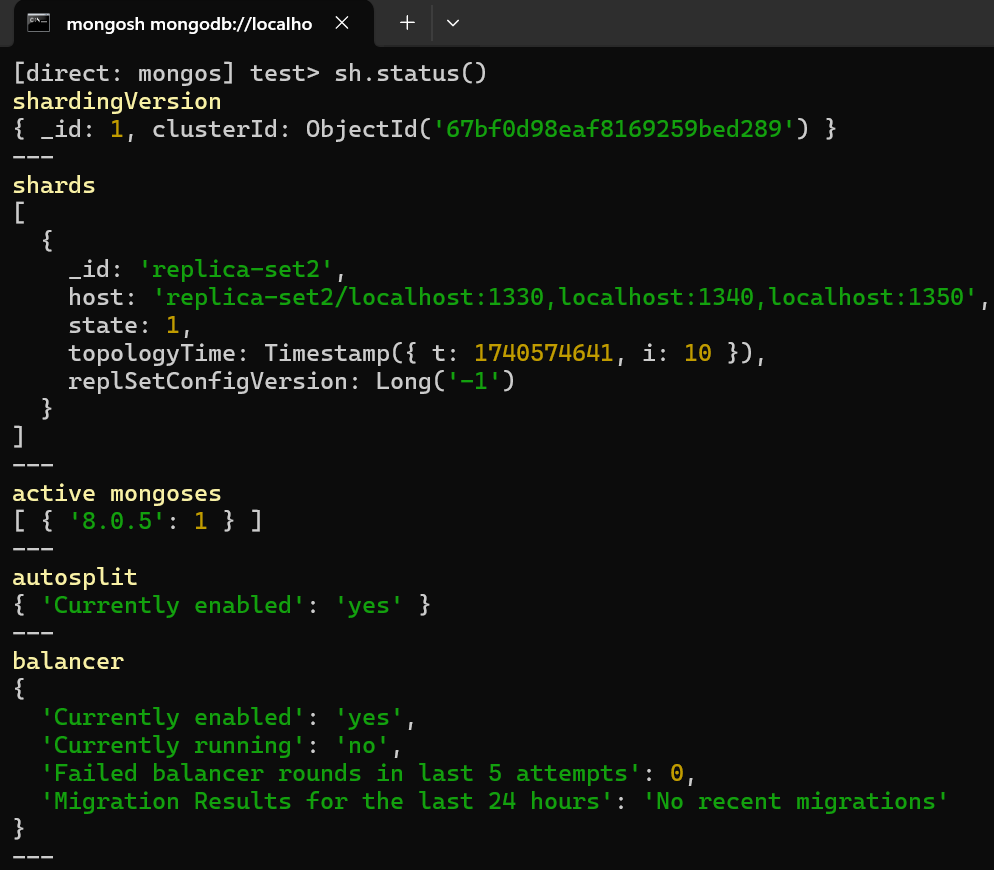


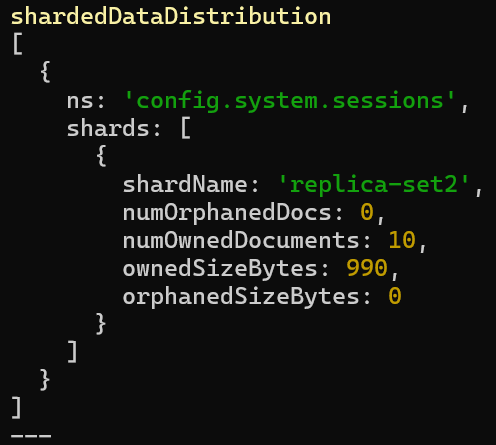
**11. Enable Sharding on a Specific Database:**

1. **Enable sharding for the database:** Connect to the mongos router (mongosh --host="localhost:1210"). Then:

sh.enableSharding("practical7")



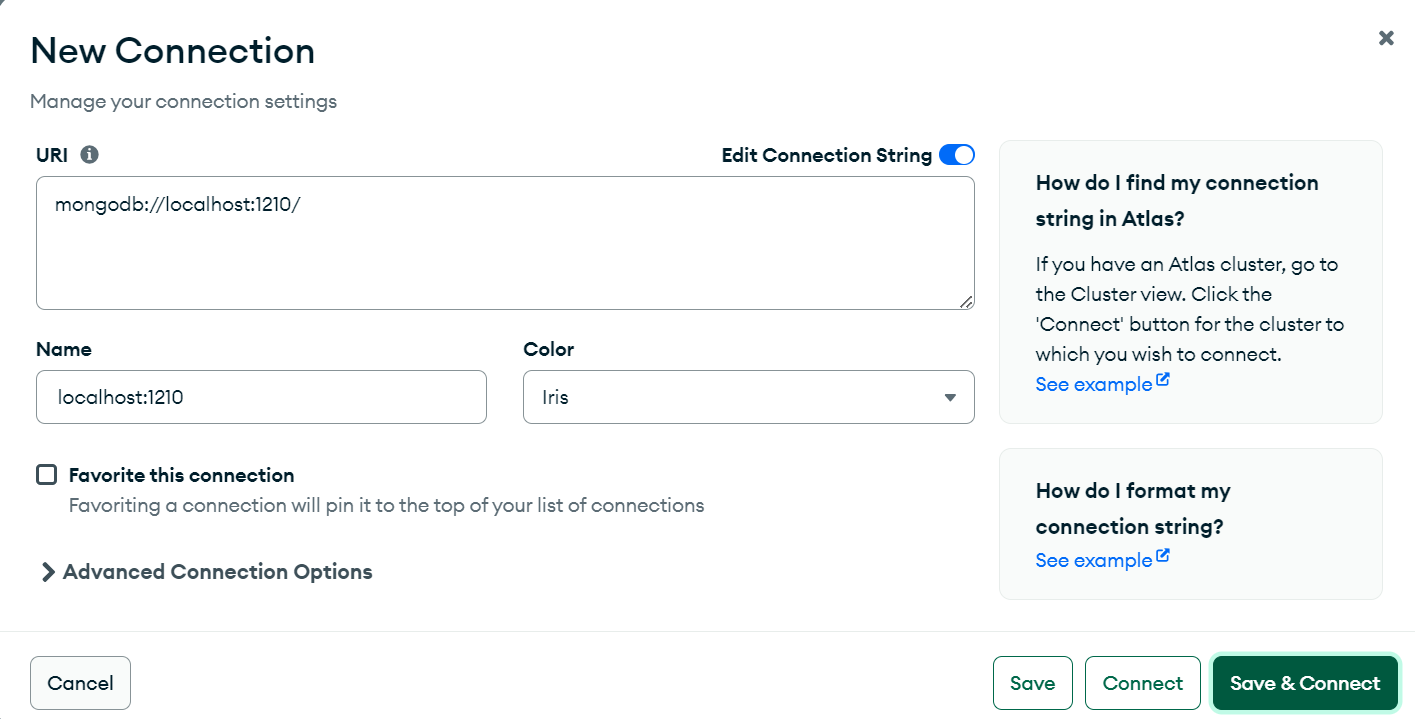


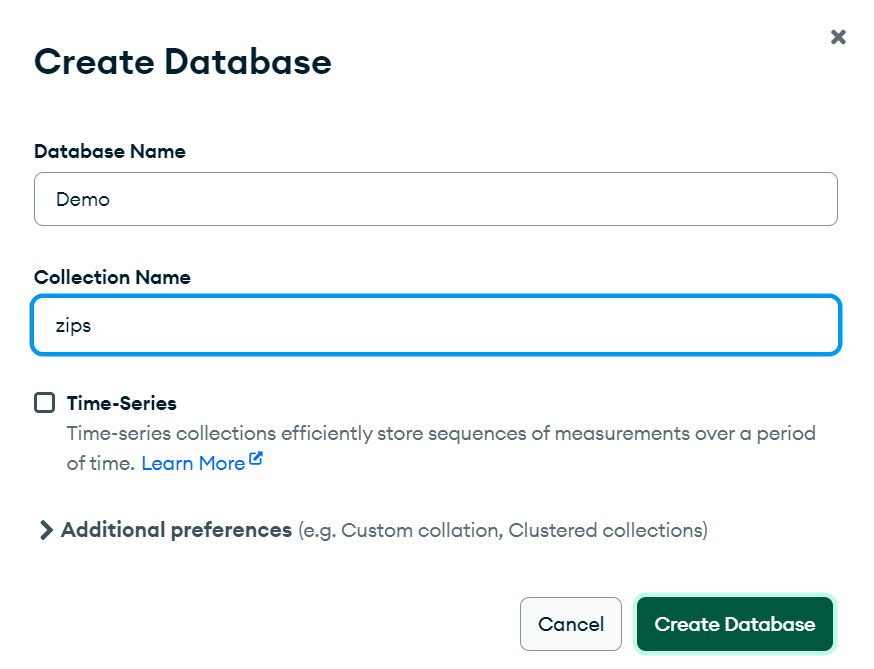




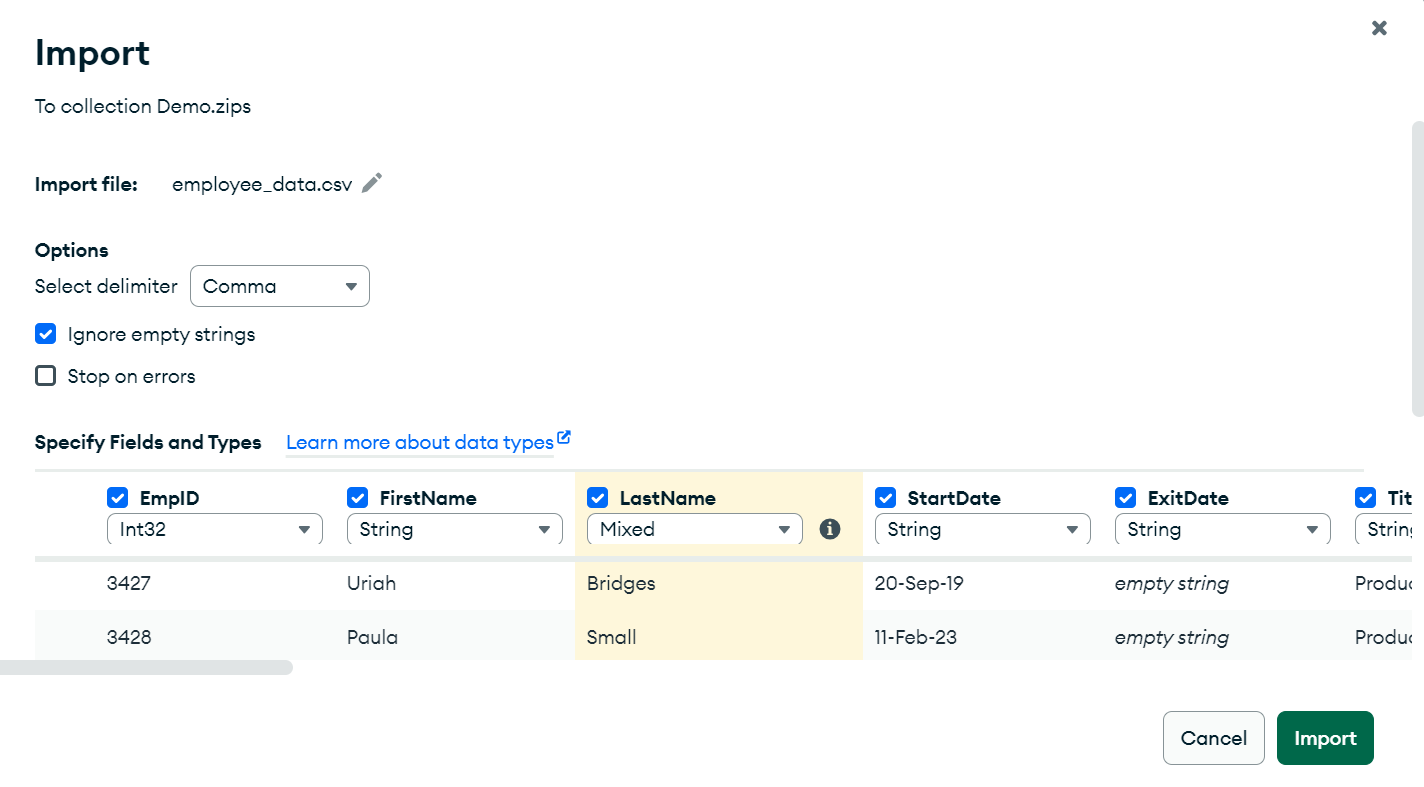
**12. Importing Data using MongoDB Compass**

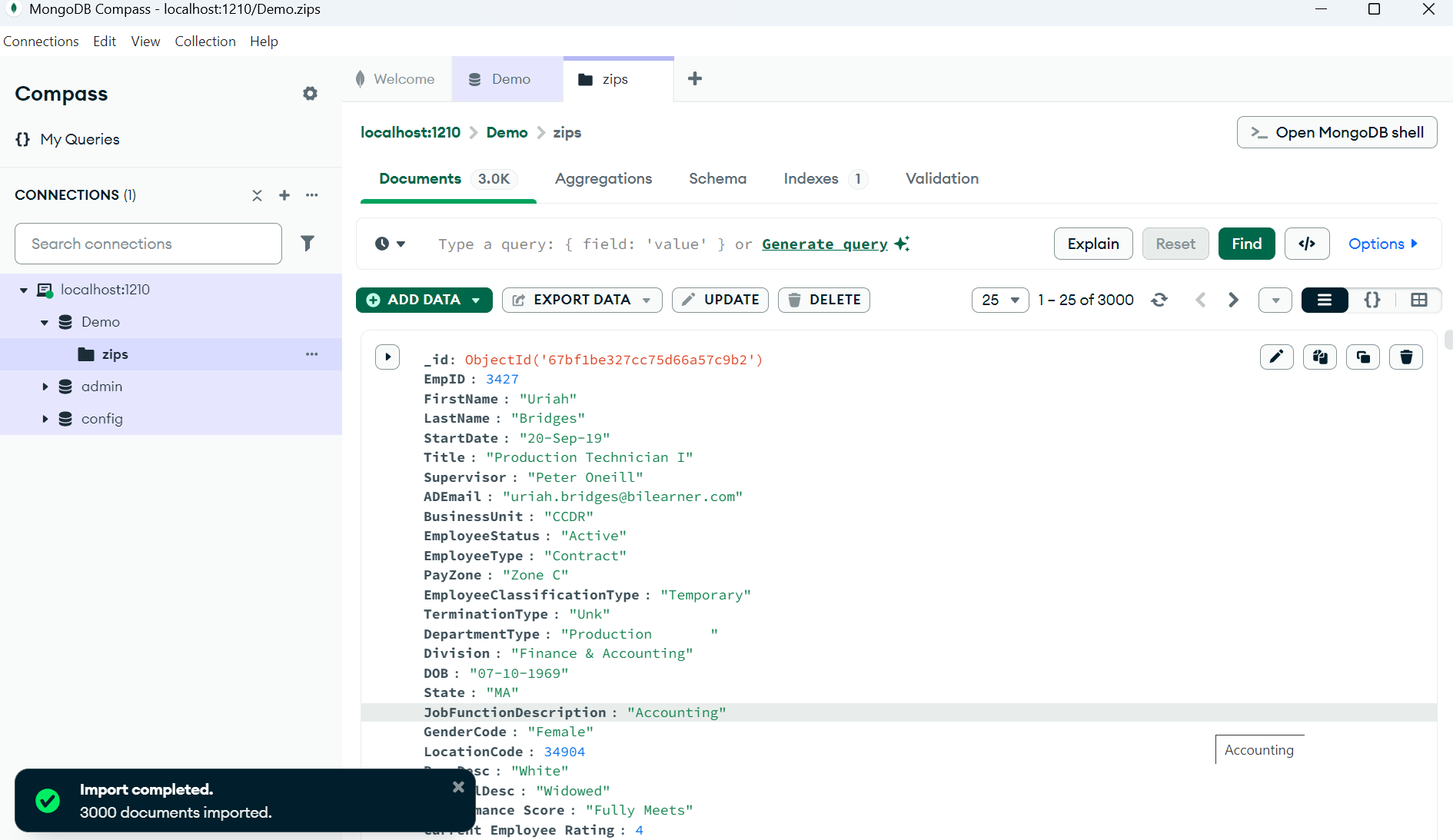
* **Launch MongoDB Compass.** Start by opening MongoDB Compass on your local machine.
* **Create a New Connection:** Then, set up a new connection by inputting the connection string mongodb://localhost:27017.
* **Input Database Name:** Create a database (“Demo”).
* **Input Collection Name:** Create a collection (“zips”).





* **Import Data:**

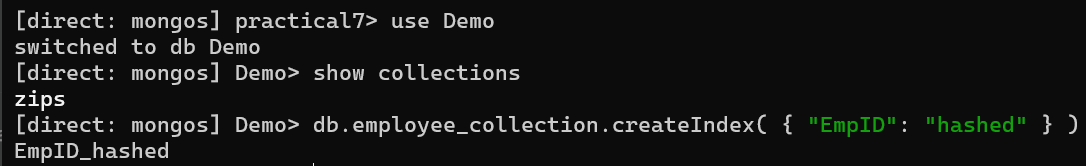


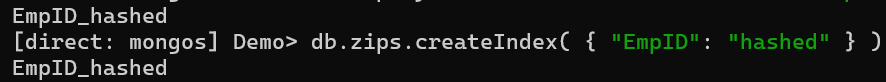


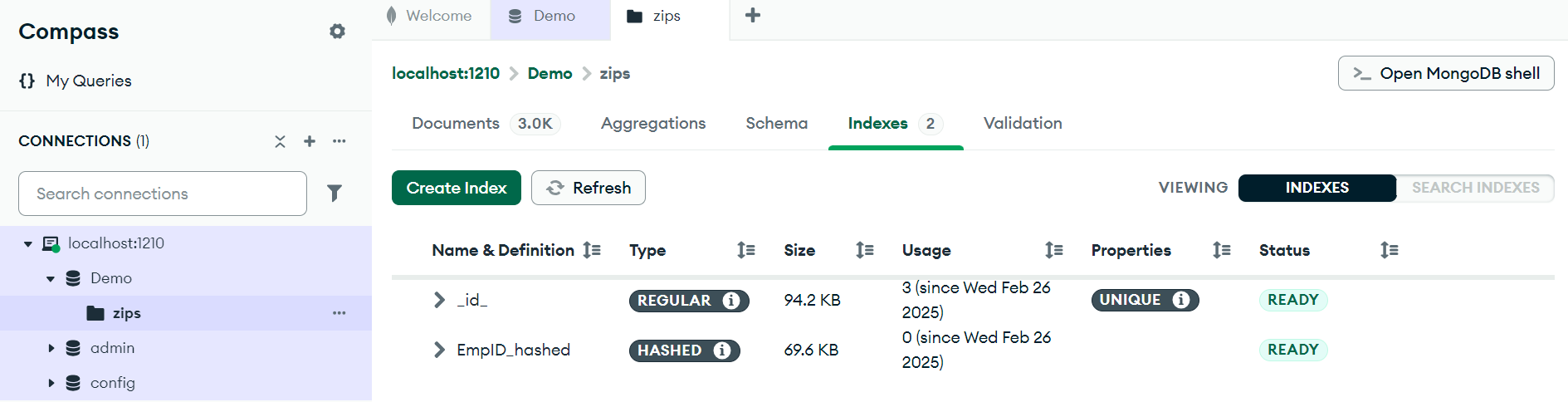
**13. Choose a Shard Key & Shard the Collection**

* **Analyze the Dataset:** Identify a suitable shard key. A shard key should have high cardinality (many distinct values) and be frequently used in queries. (“EmpId”).
* **Shard the Employee Collection:** Use the sh.shardCollection() command to shard the collection, specifying the shard key. Note: The shard key must have an index before sharding the collection. Create the index first:

db.zips.createIndex( { "EmpID": "hashed" } ) // Or 1 for ascending

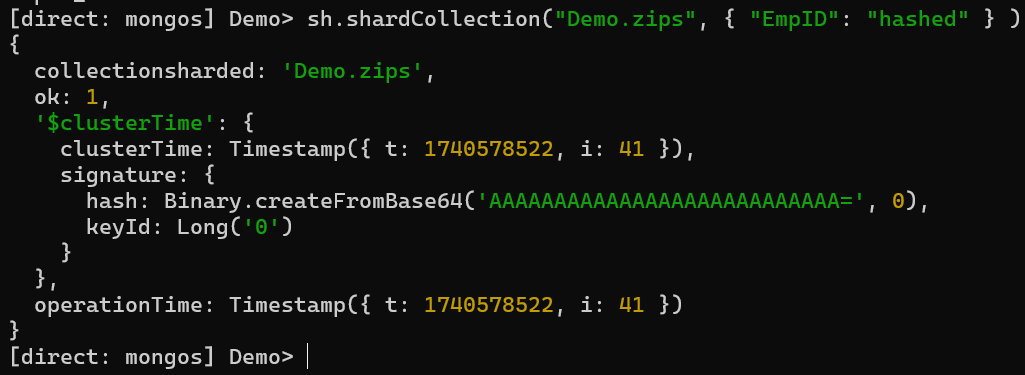






**Then shard the collection:**

sh.shardCollection("Demo.zips", { "EmpID": "hashed" } )



**14. Verify Sharding**

* **Check Sharding Status:** Use the sh.status() command in the mongosh shell connected to the mongos router to verify the sharding configuration.

