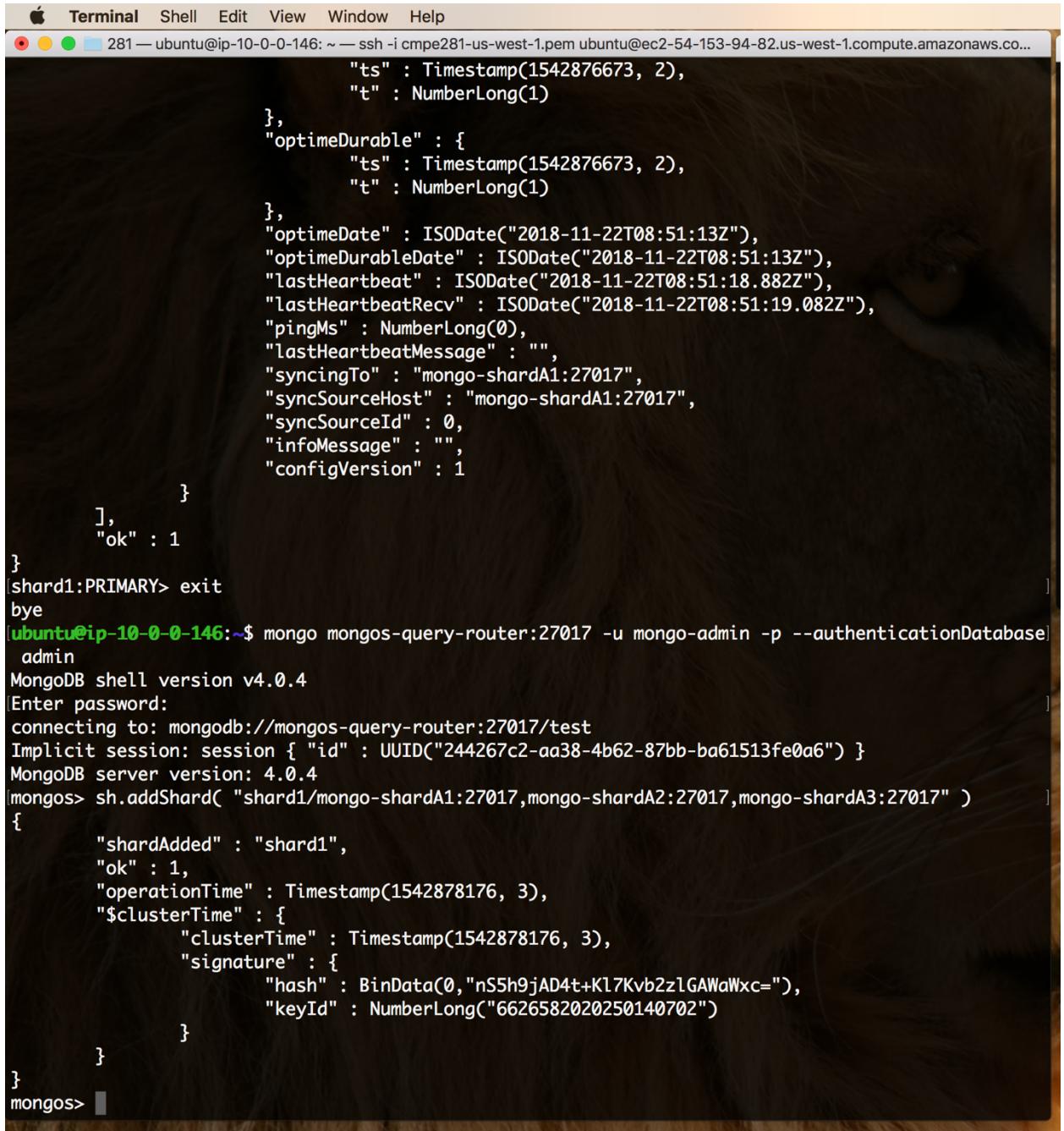


MongoDB-Sharding

1) Adding shards clusters from Mongos

Screenshot of adding shard cluster for Shard1 from Mongos shell



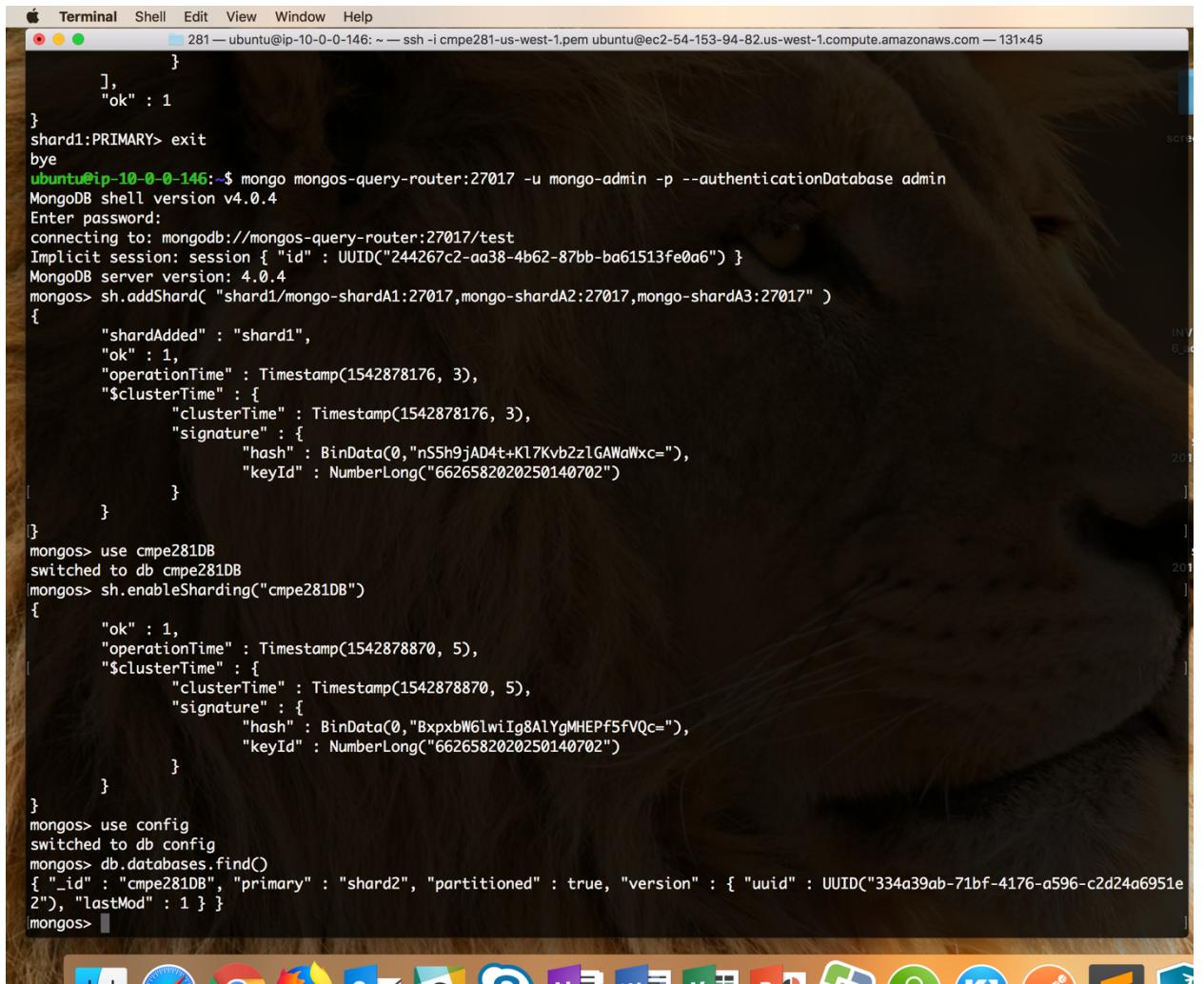
The screenshot shows a terminal window titled "Terminal" on a Mac OS X desktop. The window contains a command-line session for the MongoDB mongos shell. The user has run the command "sh.addShard" to add a shard cluster consisting of three servers: shard1/mongo-shardA1:27017, mongo-shardA2:27017, and mongo-shardA3:27017. The output of this command is displayed in the terminal, showing the shard's configuration and the "ok" status.

```
Terminal Shell Edit View Window Help
281 — ubuntu@ip-10-0-0-146: ~ — ssh -i cmpe281-us-west-1.pem ubuntu@ec2-54-153-94-82.us-west-1.compute.amazonaws.co...
"ts" : Timestamp(1542876673, 2),
      "t" : NumberLong(1)
},
"optimeDurable" : {
      "ts" : Timestamp(1542876673, 2),
      "t" : NumberLong(1)
},
"optimeDate" : ISODate("2018-11-22T08:51:13Z"),
"optimeDurableDate" : ISODate("2018-11-22T08:51:13Z"),
"lastHeartbeat" : ISODate("2018-11-22T08:51:18.882Z"),
"lastHeartbeatRecv" : ISODate("2018-11-22T08:51:19.082Z"),
"pingMs" : NumberLong(0),
"lastHeartbeatMessage" : "",
"syncingTo" : "mongo-shardA1:27017",
"syncSourceHost" : "mongo-shardA1:27017",
"syncSourceId" : 0,
"infoMessage" : "",
"configVersion" : 1
}
],
"ok" : 1
}
[shard1:PRIMARY> exit
bye
[ubuntu@ip-10-0-0-146:~$ mongo mongos-query-router:27017 -u mongo-admin -p --authenticationDatabase]
admin
MongoDB shell version v4.0.4
[Enter password:
connecting to: mongodb://mongos-query-router:27017/test
Implicit session: session { "id" : UUID("244267c2-aa38-4b62-87bb-ba61513fe0a6") }
MongoDB server version: 4.0.4
[mongos> sh.addShard( "shard1/mongo-shardA1:27017,mongo-shardA2:27017,mongo-shardA3:27017" )
{
      "shardAdded" : "shard1",
      "ok" : 1,
      "operationTime" : Timestamp(1542878176, 3),
      "$clusterTime" : {
            "clusterTime" : Timestamp(1542878176, 3),
            "signature" : {
                  "hash" : BinData(0,"nS5h9jAD4t+Kl7Kvb2z1GAWaWxc="),
                  "keyId" : NumberLong("6626582020250140702")
            }
      }
}
mongos>
```

Screenshot of adding shard cluster for Shard2 from Mongos shell

```
Terminal Shell Edit View Window Help
281 — ubuntu@ip-10-0-0-99: ~ — ssh -i cmpe281-us-west-1.pem ubuntu@ec2-54-67-74-102.us-west-1.compute.amazonaws...
},
"optimeDurable" : {
    "ts" : Timestamp(1542877565, 1),
    "t" : NumberLong(-1)
},
"optimeDate" : ISODate("2018-11-22T09:06:05Z"),
"optimeDurableDate" : ISODate("2018-11-22T09:06:05Z"),
"lastHeartbeat" : ISODate("2018-11-22T09:06:16.203Z"),
"lastHeartbeatRecv" : ISODate("2018-11-22T09:06:16.268Z"),
"pingMs" : NumberLong(0),
"lastHeartbeatMessage" : "",
"syncingTo" : "",
"syncSourceHost" : "",
"syncSourceId" : -1,
"infoMessage" : "",
"configVersion" : 1
}
],
"ok" : 1
}
shard2:SECONDARY>
[shard2:PRIMARY> exit
bye
[ubuntu@ip-10-0-0-99:~$ mongo mongos-query-router:27017 -u mongo-admin -p --authenticationData]
base admin
MongoDB shell version v4.0.4
[Enter password:
connecting to: mongodb://mongos-query-router:27017/test
Implicit session: session { "id" : UUID("3789d821-4d4a-4c85-a945-6f05613b352d") }
MongoDB server version: 4.0.4
[mongos> sh.addShard( "shard2/mongo-shardB1:27017,mongo-shardB2:27017,mongo-shardB3:27017" )
{
    "shardAdded" : "shard2",
    "ok" : 1,
    "operationTime" : Timestamp(1542878196, 2),
    "$clusterTime" : {
        "clusterTime" : Timestamp(1542878196, 2),
        "signature" : {
            "hash" : BinData(0,"1iJlrUCfxgQipZ7eU7N2iK1rgZ0="),
            "keyId" : NumberLong("6626582020250140702")
        }
    }
}
mongos> ]
```

Using enableSharding() method enabling the shards at database level

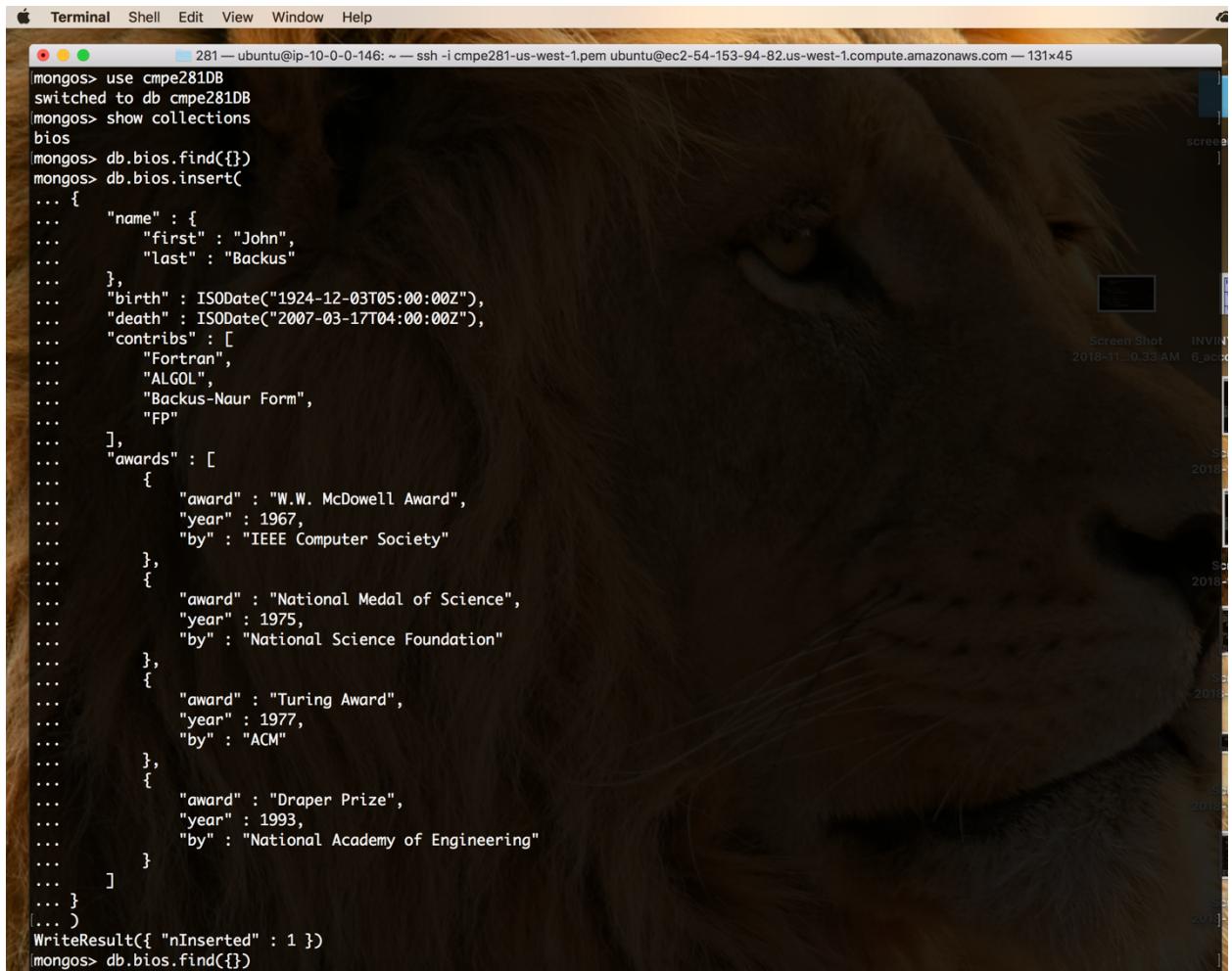


```
ubuntu@ip-10-0-0-146:~$ mongo mongos-query-router:27017 -u mongo-admin -p --authenticationDatabase admin
MongoDB shell version v4.0.4
Enter password:
connecting to: mongodb://mongos-query-router:27017/test
Implicit session: session { "id" : UUID("244267c2-aa38-4b62-87bb-ba61513fe0a6") }
MongoDB server version: 4.0.4
mongos> sh.addShard( "shard1/mongo-shardA1:27017,mongo-shardA2:27017,mongo-shardA3:27017" )
{
  "shardAdded" : "shard1",
  "ok" : 1,
  "operationTime" : Timestamp(1542878176, 3),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1542878176, 3),
    "signature" : {
      "hash" : BinData(0,"nS5h9jAD4t+Kl7Kvb2zlGAWaWxc="),
      "keyId" : NumberLong("6626582020250140702")
    }
  }
}
mongos> use cmpe281DB
switched to db cmpe281DB
mongos> sh.enableSharding("cmpe281DB")
{
  "ok" : 1,
  "operationTime" : Timestamp(1542878870, 5),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1542878870, 5),
    "signature" : {
      "hash" : BinData(0,"BpxxbW6lwiIg8AlYgMHEPf5fVQc="),
      "keyId" : NumberLong("6626582020250140702")
    }
  }
}
mongos> use config
switched to db config
mongos> db.databases.find()
{ "_id" : "cmpe281DB", "primary" : "shard2", "partitioned" : true, "version" : { "uuid" : UUID("334a39ab-71bf-4176-a596-c2d24a6951e2")}, "lastMod" : 1 }
mongos>
```




```
mongos> use cmpe281DB
switched to db cmpe281DB
mongos> sh.enableSharding("cmpe281DB")
{
  "ok" : 1,
  "operationTime" : Timestamp(1542878870, 5),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1542878870, 5),
    "signature" : {
      "hash" : BinData(0,"BpxxbW6lwiIg8AlYgMHEPf5fVQc="),
      "keyId" : NumberLong("6626582020250140702")
    }
  }
}
mongos> use config
switched to db config
mongos> db.databases.find()
{ "_id" : "cmpe281DB", "primary" : "shard2", "partitioned" : true, "version" : { "uuid" : UUID("334a39ab-71bf-4176-a596-c2d24a6951e2")}, "lastMod" : 1 }
```

Now inserting some data into the database that we created



```
mongos> use cmpe281DB
switched to db cmpe281DB
mongos> show collections
bios
mongos> db.bios.find({})
mongos> db.bios.insert(
... {
...   "name" : {
...     "first" : "John",
...     "last" : "Backus"
...   },
...   "birth" : ISODate("1924-12-03T05:00:00Z"),
...   "death" : ISODate("2007-03-17T04:00:00Z"),
...   "contribs" : [
...     "Fortran",
...     "ALGOL",
...     "Backus-Naur Form",
...     "FP"
...   ],
...   "awards" : [
...     {
...       "award" : "W.W. McDowell Award",
...       "year" : 1967,
...       "by" : "IEEE Computer Society"
...     },
...     {
...       "award" : "National Medal of Science",
...       "year" : 1975,
...       "by" : "National Science Foundation"
...     },
...     {
...       "award" : "Turing Award",
...       "year" : 1977,
...       "by" : "ACM"
...     },
...     {
...       "award" : "Draper Prize",
...       "year" : 1993,
...       "by" : "National Academy of Engineering"
...     }
...   ]
... }
... )
WriteResult({ "nInserted" : 1 })
mongos> db.bios.find({})
```

```

mongos> db.bios.insert(
... {
...     "name" : {
...         "first" : "John",
...         "last" : "McCarthy"
...     },
...     "birth" : ISODate("1927-09-04T04:00:00Z"),
...     "death" : ISODate("2011-12-24T05:00:00Z"),
...     "contribs" : [
...         "Lisp",
...         "Artificial Intelligence",
...         "ALGOL"
...     ],
...     "awards" : [
...         {
...             "award" : "Turing Award",
...             "year" : 1971,
...             "by" : "ACM"
...         },
...         {
...             "award" : "Kyoto Prize",
...             "year" : 1988,
...             "by" : "Inamori Foundation"
...         },
...         {
...             "award" : "National Medal of Science",
...             "year" : 1990,
...             "by" : "National Science Foundation"
...         }
...     ]
... }
... )
WriteResult({ "nInserted" : 1 })
mongos> db.bios.find({})
{ "_id" : ObjectId("5bf67ce2bcf7e83a7dfd0f45"), "name" : { "first" : "John", "last" : "Backus" }, "birth" : ISODate("1924-12-03T05:00:00Z"), "death" : ISODate("2007-03-17T04:00:00Z"), "contribs" : [ "Fortran", "ALGOL", "Backus-Naur Form", "FP" ], "awards" : [ { "award" : "W.W. McDowell Award", "year" : 1967, "by" : "IEEE Computer Society" }, { "award" : "National Medal of Science", "year" : 1975, "by" : "National Science Foundation" }, { "award" : "Turing Award", "year" : 1977, "by" : "ACM" }, { "award" : "Draper Prize", "year" : 1993, "by" : "National Academy of Engineering" } ] }
{ "_id" : ObjectId("5bf67cf0bcf7e83a7dfd0f46"), "name" : { "first" : "John", "last" : "McCarthy" }, "birth" : ISODate("1927-09-04T04:00:00Z"), "death" : ISODate("2011-12-24T05:00:00Z"), "contribs" : [ "Lisp", "Artificial Intelligence", "ALGOL" ], "awards" : [ { "award" : "Turing Award", "year" : 1971, "by" : "ACM" }, { "award" : "Kyoto Prize", "year" : 1988, "by" : "Inamori Foundation" }, { "award" : "National Medal of Science", "year" : 1990, "by" : "National Science Foundation" } ] }
mongos> db.bios.ensureIndex( { name : 1 } )
{

```

Now as we have 2 data inserted into our bios collection, we are creating an index for our shard key

Demonstrating sharding with 'name' as a shard key-

In the project md file I have taken a hashed shard key _id. So, tried sharding with different keys

Using name key as the shard key as it will be easy to distribute the data so that same name or same first letter names will be collected together in one shard and easy to query the database.

```
|mongos> db.bios.ensureIndex( { name : 1 } )
{
  "raw" : [
    {
      "shard2/mongo-shardB1:27017,mongo-shardB2:27017,mongo-shardB3:27017" : {
        "numIndexesBefore" : 2,
        "numIndexesAfter" : 2,
        "note" : "all indexes already exist",
        "ok" : 1
      },
      "shard1/mongo-shardA1:27017,mongo-shardA2:27017,mongo-shardA3:27017" : {
        "ok" : 0,
        "errmsg" : "request doesn't allow collection to be created implicitly",
        "code" : 227,
        "codeName" : "CannotImplicitlyCreateCollection",
        "ns" : "cmpe281DB.bios"
      }
    },
    "ok" : 1,
    "operationTime" : Timestamp(1542880528, 1),
    "$clusterTime" : {
      "clusterTime" : Timestamp(1542880529, 4),
      "signature" : {
        "hash" : BinData(0,"GQoKTQdM/0YcgbNghF0DJl/kFJ0="),
        "keyId" : NumberLong("6626582020250140702")
      }
    }
  }
}
mongos>
```

```
|mongos> sh.shardCollection( "cmpe281DB.bios", { "name" : 1 } )
{
  "collectionsharded" : "cmpe281DB.bios",
  "collectionUUID" : UUID("836c1cea-0212-4420-ba23-d70a2b2cf773"),
  "ok" : 1,
  "operationTime" : Timestamp(1542880693, 4),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1542880693, 4),
    "signature" : {
      "hash" : BinData(0,"jCx/lvE+4QnJQolCM6Ws0F0QMSk="),
      "keyId" : NumberLong("6626582020250140702")
    }
  }
}
mongos>
```

When you do `getShardDistribution()` then you will get that shard2 is having 100% of the data because MongoDB only does balancing of data among shards when the number of chunks is greater than 20 and migration threshold is 2.

As we only have 10 records in Bios collection. All the data went to shard2 cluster. When the records will increase then the data distribution will be done among shards

Available at- <https://docs.mongodb.com/manual/core/sharding-balancer-administration/#sharding-migration-thresholds>

```
{  
mongos> db.bios.getShardDistribution()  
  
Shard shard2 at shard2/mongo-shardB1:27017,mongo-shardB2:27017,mongo-shardB3:27017  
  data : 3KiB docs : 10 chunks : 1  
  estimated data per chunk : 3KiB  
  estimated docs per chunk : 10  
  
Totals  
  data : 3KiB docs : 10 chunks : 1  
  Shard shard2 contains 100% data, 100% docs in cluster, avg obj size on shard : 352B  
  
mongos> █
```