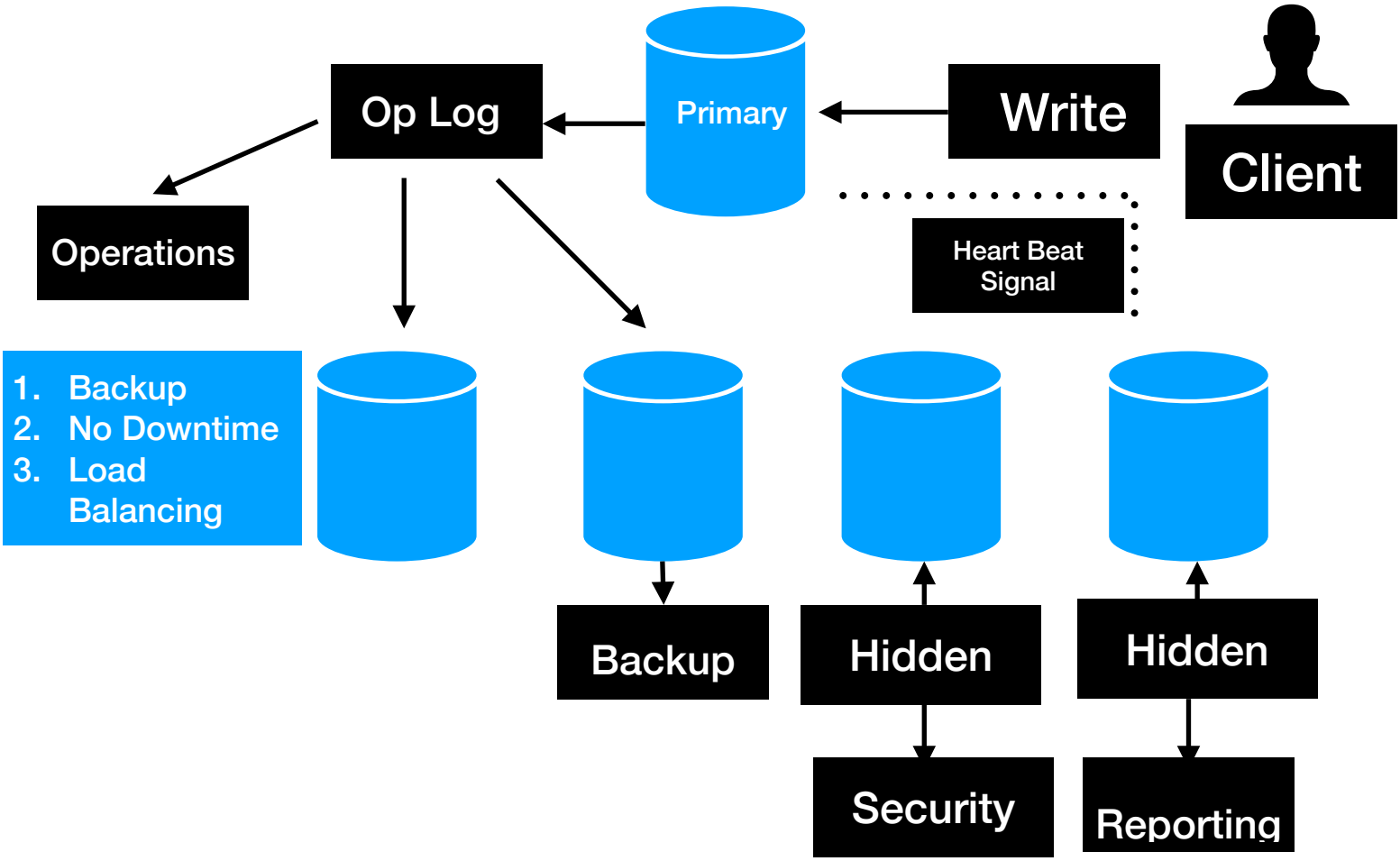
Replica set:

1. Primary Instance can have Maximum of 50 Instance.
2. Primary & Secondary instance which is called Replica set.
3. Primary always send the Heart Beat Signal to Secondary and Vice Versa. Because of that both of know about their availability status.
4. All the Write Operation from the Client will always happen to the Primary Instance.
5. Whenever write operation happen Primary will generate an Op Log, and based one the configuration happens to the secondary Op Log will copy and execute. After execute this Op Log, secondary instance having the exact content of what the Primary having.
6. Actually, Once of the Secondary have reporting and Security. This is based on the Companies Interest.



Setting Up Replica Set Servers:

Here, let assuming that, you have installed MongoDB Server and now you trying to create a db.

```
centos@ip-172-26-8-241:~
[centos@ip-172-26-8-241 ~]$ mkdir replica01
[centos@ip-172-26-8-241 ~]$ mkdir replica02
[centos@ip-172-26-8-241 ~]$ mkdir replica03
[centos@ip-172-26-8-241 ~]$ chmod -R 777 replica01
[centos@ip-172-26-8-241 ~]$ chmod -R 777 replica02
[centos@ip-172-26-8-241 ~]$ chmod -R 777 replica03
[centos@ip-172-26-8-241 ~]$
```

```
centos@ip-172-26-8-241:~
[centos@ip-172-26-8-241 ~]$ mongo --port 27020
MongoDB shell version v4.0.5
connecting to: mongodb://127.0.0.1:27020/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("cf21f1bd-b4dd-4d1b-95b0-57c17f1898a9") }
MongoDB server version: 4.0.5
Server has startup warnings:
2019-02-04T04:33:11.046+0000 I CONTROL  [initandlisten]
2019-02-04T04:33:11.046+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database
.
2019-02-04T04:33:11.046+0000 I CONTROL  [initandlisten] **          Read and write access to data and configuratio
n is unrestricted.
2019-02-04T04:33:11.046+0000 I CONTROL  [initandlisten]
MongoDB Enterprise >
```

Now, I have created all three servers

```
centos@ip-172-26-8-241:~
[centos@ip-172-26-8-241 ~]$ ps -ef | grep mongo
centos   16407     1  0 04:33 ?        00:00:01 mongod --dbpath replica01 --port 27020 --logpath replica01/log --l
ogappend --fork --bind_ip_all --replSet myreplicaset01
centos   16460     1  1 04:35 ?        00:00:01 mongod --dbpath replica02 --port 27021 --logpath replica02/log --l
ogappend --fork --bind_ip_all --replSet myreplicaset01
centos   16491     1  6 04:36 ?        00:00:00 mongod --dbpath replica03 --port 27022 --logpath replica03/log --f
ork --logappend --bind_ip_all --replSet myreplicaset01
centos   16520 16375  0 04:36 pts/0    00:00:00 grep --color=auto mongo
[centos@ip-172-26-8-241 ~]$
```

Initiating Replica set and adding members.

```
MongoDB Enterprise > rs.initiate()
{
        "info2" : "no configuration specified. Using a default configuration for the set",
        "me" : "ip-172-26-8-241.ap-south-1.compute.internal:27020",
        "ok" : 1,
        "operationTime" : Timestamp(1549255229, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1549255229, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
MongoDB Enterprise myreplicaset01:SECONDARY>
MongoDB Enterprise myreplicaset01:PRIMARY>
```

When you fire this command, rs.config() -> you can see at this point of time only one member is there. And, also you can rs.status().

Now, I am adding members. After successful adding we can check the rs.config() and rs.status(

```
MongoDB Enterprise myreplicaset01:PRIMARY> rs.add("ip-172-26-8-241.ap-south-1.compute.internal:27021")
{
        "ok" : 1,
        "operationTime" : Timestamp(1549255355, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1549255355, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

```
MongoDB Enterprise myreplicaset01:PRIMARY> rs.add("ip-172-26-8-241.ap-south-1.compute.internal:27022")
{
        "ok" : 1,
        "operationTime" : Timestamp(1549255414, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1549255414, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
MongoDB Enterprise myreplicaset01:PRIMARY>
```

Now, I am adding at add documents in Primary.

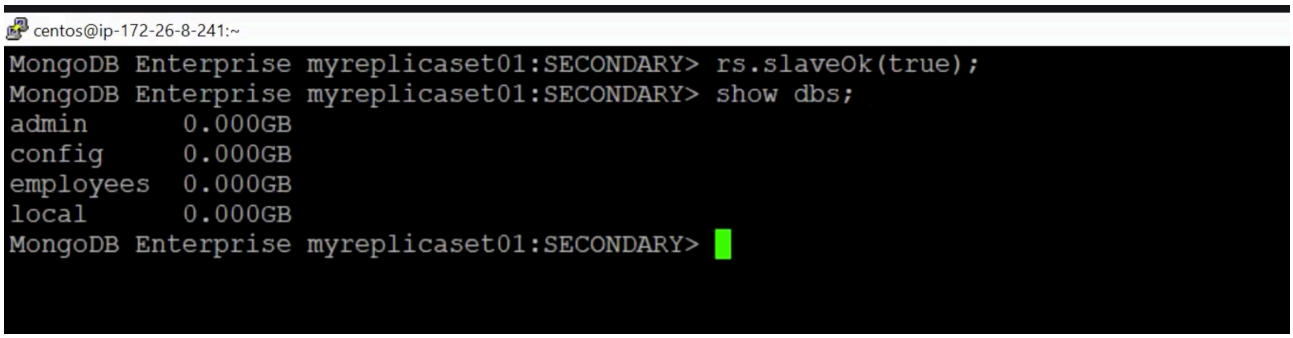
```
centos@ip-172-26-8-241:~
MongoDB Enterprise myreplicaset01:PRIMARY> use employees
switched to db employees
MongoDB Enterprise myreplicaset01:PRIMARY> db.myemployees.insert({_id:1, name:"myemp1", sal:500});
WriteResult({ "nInserted" : 1 })
MongoDB Enterprise myreplicaset01:PRIMARY> db.myemployees.find();
{ "_id" : 1, "name" : "myemp1", "sal" : 500 }
MongoDB Enterprise myreplicaset01:PRIMARY>
```

After successfully added, Now, I am going to check from secondary server.

```
[centos@ip-172-26-8-241 ~]$ mongo --port 27021
MongoDB shell version v4.0.5
connecting to: mongodb://127.0.0.1:27021/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("6bc017e8-3364-4848-a30f-b7114b2e7f8e") }
MongoDB server version: 4.0.5
Server has startup warnings:
2019-02-04T04:35:10.642+0000 I CONTROL  [initandlisten]
2019-02-04T04:35:10.643+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the databas
.
2019-02-04T04:35:10.643+0000 I CONTROL  [initandlisten] **          Read and write access to data and configurati
n is unrestricted.
2019-02-04T04:35:10.643+0000 I CONTROL  [initandlisten]
MongoDB Enterprise myreplicaset01:SECONDARY> show dbs;
2019-02-04T04:50:01.437+0000 E QUERY    [js] Error: listDatabases failed:{
        "operationTime" : Timestamp(1549255801, 1),
        "ok" : 0,
        "errmsg" : "not master and slaveOk=false",
        "code" : 13435,
        "codeName" : "NotMasterNoSlaveOk",
        "$clusterTime" : {
                "clusterTime" : Timestamp(1549255801, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
} :
_getErrorWithCode@src/mongo/shell/utils.js:25:13
Mongo.prototype.getDBs@src/mongo/shell/mongo.js:124:1
shellHelper.show@src/mongo/shell/utils.js:876:19
shellHelper@src/mongo/shell/utils.js:766:15
```

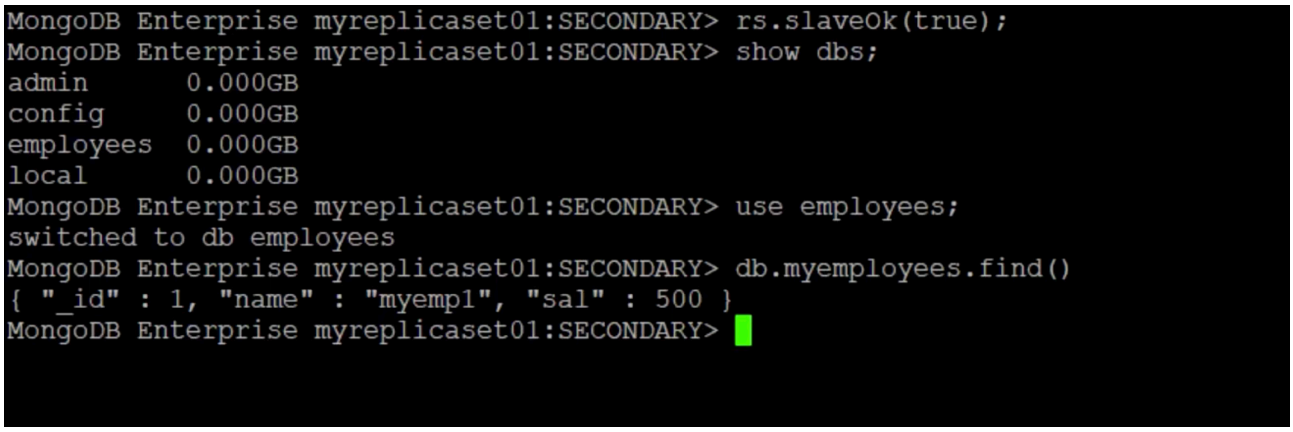
I am getting this error because by default you will not access Secondary node.

Please Note: Write Operation Always Primary and Read Operation by default is Primary. You need to set slaveOk = true Changs in the Secondary server then, you can actually go ahead and do the read operations



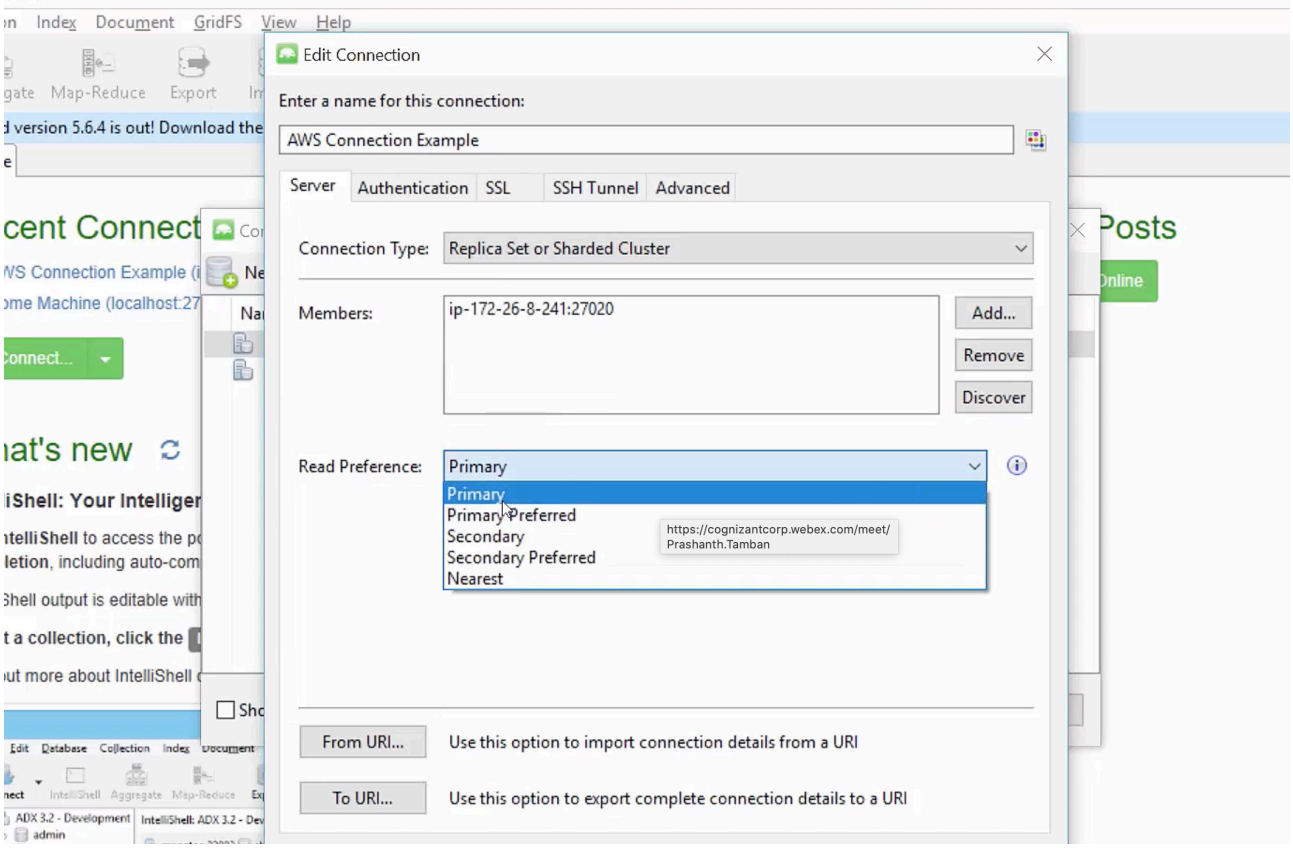
Now, you see show dbs; shows the output and find the myemployees output.

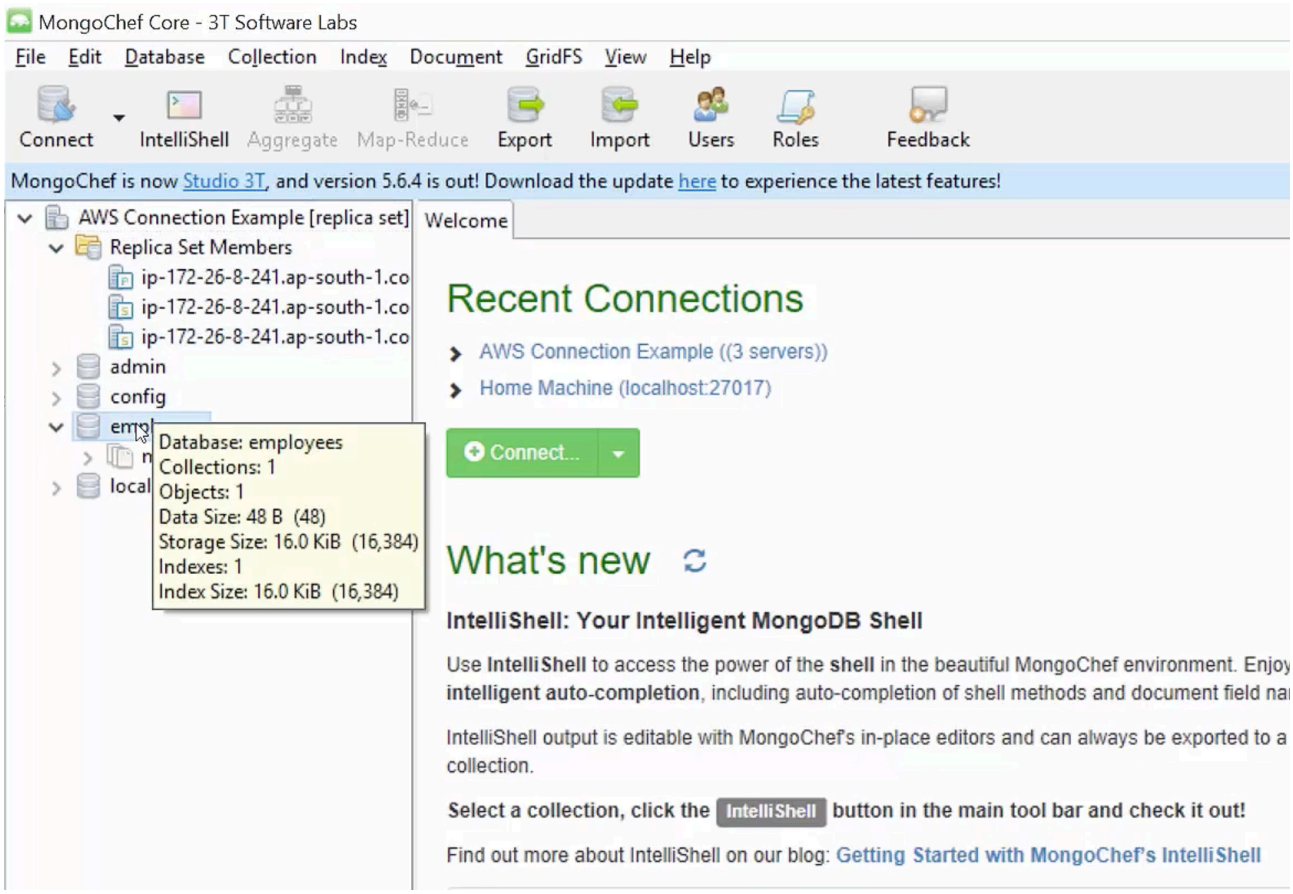
Importantly, Secondary Server Only Read Operation is allowed. And you must need to set rs.slaveOk(true) for every secondary node.



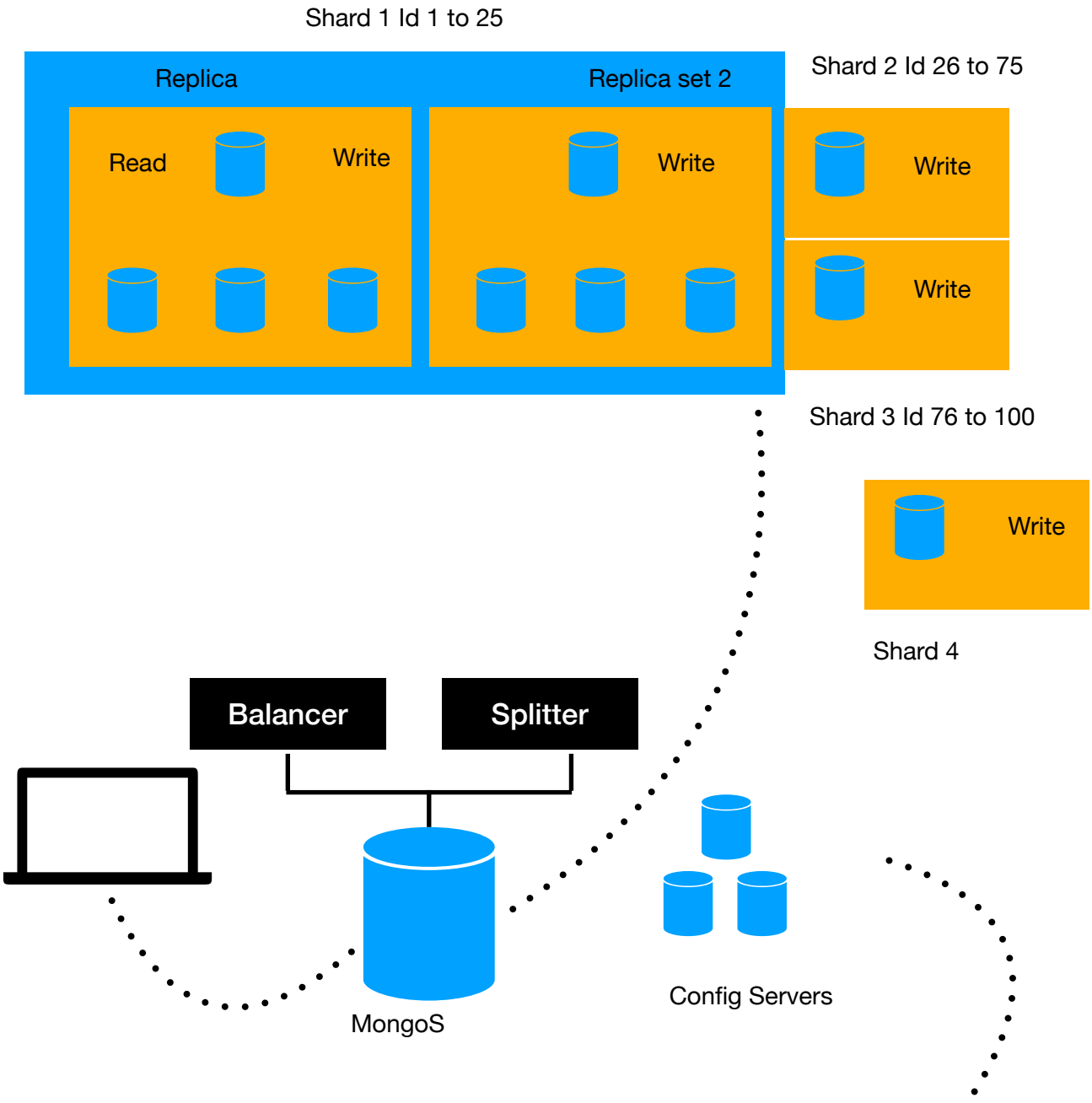
now, I am connecting through MongoChef Core - 3T, which IDE for MongoDB. Here, you have Read Preference. Meaning, first you have to set rs.slaveOk(true) and later when read operation comes, you check actually select the Read Preference which close to Primary.

Hence, I request you to do the same as above in your MongoDB and check the output and after that do the necessary changes.


Edit Connection
Enter a name for this connection:
AWS Connection Example
Server
Authentication
SSL
SSH Tunnel
Advanced
Connection Type:
Replica Set or Sharded Cluster
Members:
ip-172-26-8-241:27020
Add...
Remove
Discover
Read Preference:
Primary
Primary
Primary Preferred
Secondary
Secondary Preferred
Nearest
From URI...
Use this option to import connection details from a URI
To URI...
Use this option to export complete connection details to a URI


MongoChef Core - 3T Software Labs
File Edit Database Collection Index Document GridFS View Help
Connect
IntelliShell
Aggregate
Map-Reduce
Export
Import
Users
Roles
Feedback
MongoChef is now Studio 3T, and version 5.6.4 is out! Download the update here to experience the latest features!
AWS Connection Example [replica set]
Replica Set Members
ip-172-26-8-241.ap-south-1.co
ip-172-26-8-241.ap-south-1.co
ip-172-26-8-241.ap-south-1.co
admin
config
Database: employees
Collections: 1
Objects: 1
Data Size: 48 B (48)
Storage Size: 16.0 KiB (16,384)
Indexes: 1
Index Size: 16.0 KiB (16,384)
Welcome
Recent Connections
AWS Connection Example ((3 servers))
Home Machine (localhost:27017)
Connect...
What's new
IntelliShell: Your Intelligent MongoDB Shell
Use IntelliShell to access the power of the shell in the beautiful MongoChef environment. Enjoy intelligent auto-completion, including auto-completion of shell methods and document field na
IntelliShell output is editable with MongoChef's in-place editors and can always be exported to a collection.
Select a collection, click the IntelliShell button in the main tool bar and check it out!
Find out more about IntelliShell on our blog: Getting Started with MongoChef's IntelliShell

Sharding Techniques:



Shard 1	1	25
Shard 2	26	75
Shard 3	76	100

1. Shard 1 has data from 1 to 25, Shard 2 has data from 26 to 75, Shard 3 has data from 76 to 100. This information is mentioning in Config Servers.
2. Whenever any request from Customer, it will goes to MongoS and contacting the Config Server and get the location based one the request has routed to either Shard 1 or 2 or 3.
3. Balancer and Splitter will ensure that, at any give point of them all the Shard all equally utilised.
4. If you are introducing the new Shard 4, then you don't have to worry because Balancer and Splitter will found that Shard 4 is newly added and which under performed. So then, all the data will shared to Shard 4 and make them ready to server the request. Which is called automatic scaling.
5. If you are simply calling find() then, mongos doesn't know which Shard need to call, therefore Mongos broadcast it to all Shard and find it. This is also possible reduce the performance.
6. Always, Operation does contain Shard Key, request it goes to the respected Shard and fetch the data.

Important: Whenever you are operations are Read Intensive, you can have Replica set at your rescue. Whenever you are operations are Write intensive, you can have Shard Cluster at your rescue. Together you have Read & Write powerful mechanism

Balancer & Splitter:

1. For e.g. Shard 1 have 10 GB Memory, Shard 2 & 3 also having 10 GB Memory. Assuming here, Shard 1 receiving multiple request and Shard1 memory consuming is 8 GB. In the given situation, Balancer & Splitter understand that other 2 Shard are not utilised hence, Shard 1 data will be splitting to two Shard so that, utilisation is equals at any given point of time.

Capped Collections:

When you create a collections, if you set capped = true, then, when retrieval process, data will be retrieved the way you interested. Non capped collection, there is no guaranteed that retrieval may you have to use Natural.