

Dominic M Brause

Jing-Hua Ye

NoSQL Databases

24. May 2018

Assignment III - Replica design

Part 1 - Questions about replica sets

Explain how this replica set behaves in terms of automated failover?

All created replicas consist of individual nodes that hold exactly the same data across one replica set. The primary node, is the only one that is able to receive data and therefore passes it on to all secondaries via OPLOG.

In the event of a primary failure, the replica set needs to come to a qualified majority when deciding for a new primary node. Until this majority has been found, the set remains in read-only mode to protect the consistency of the set. Other shards / replica sets are not affected by this write-lock.

How to increase the size of a replica OPLOG to 25GB?

The information for a replica's OPLOG is stored on each individual member of a replica set. To adjust the size, we connect to our primary node and enter the following command to verify the current size of the OPLOG

```
> use local  
  
> db.oplog.rs.stats().maxSize
```

Once verified, we can adjust the size to 25GB like so:

```
db.adminCommand({ replSetResizeOplog: 1, size: 25000 })
```

```

Lab 3 : dominic ~ mongo --port 40020
MongoDB shell version v3.6.4
connecting to: mongod://127.0.0.1:40020/
MongoDB server version: 3.6.4
Server has startup warnings:
2018-05-27T12:09:54.614+0100 I CONTROL [initandlisten]
2018-05-27T12:09:54.614+0100 I CONTROL [initandlisten] ** WARNING: Access control is not enabled fo
r the database.
2018-05-27T12:09:54.614+0100 I CONTROL [initandlisten] **          Read and write access to data an
d configuration is unrestricted.
2018-05-27T12:09:54.614+0100 I CONTROL [initandlisten]
2018-05-27T12:09:54.614+0100 I CONTROL [initandlisten] ** WARNING: This server is bound to localhos
t.
2018-05-27T12:09:54.614+0100 I CONTROL [initandlisten] **          Remote systems will be unable to
connect to this server.
2018-05-27T12:09:54.614+0100 I CONTROL [initandlisten] **          Start the server with --bind_ip
<address> to specify which IP
2018-05-27T12:09:54.614+0100 I CONTROL [initandlisten] **          addresses it should serve respon
ses from, or with --bind_ip_all to
2018-05-27T12:09:54.614+0100 I CONTROL [initandlisten] **          bind to all interfaces. If this
behavior is desired, start the
2018-05-27T12:09:54.614+0100 I CONTROL [initandlisten] **          server with --bind_ip 127.0.0.1
to disable this warning.
2018-05-27T12:09:54.614+0100 I CONTROL [initandlisten]
cork:PRIMARY> show dbs
admin 0.0806GB
config 0.0806GB
data 0.0956GB
local 0.0806GB
cork:PRIMARY> use local
switched to db local
cork:PRIMARY> db.oplog.rs.s
db.oplog.rs.save(          db.oplog.rs.setWriteConcern( db.oplog.rs.stats(
db.oplog.rs.setSlaveOk(    db.oplog.rs.shellPrint(      db.oplog.rs.storageSize(
cork:PRIMARY> db.oplog.rs.stats().maxsize
cork:PRIMARY> db.oplog.rs.stats()
{
  "ns" : "local.oplog.rs",
  "size" : 13692028,
  "count" : 25820,
  "avgObjSize" : 534,
  "storageSize" : 6144000,
  "capped" : true,
  "max" : -1,
  "maxSize" : 201326592,
  "sleepCount" : 0,
  "sleepMS" : 0,
  "wiredTiger" : {
    "metadata" : {
      "formatVersion" : 1,
      "oplogKeyExtractionVersion" : 1
    },
    "creationString" : "access_pattern_hint=none,allocation_size=4KB,app_metadata={forma
tVersion=1,oplogKeyExtractionVersion=1},assert={commit_timestamp=none,read_timestamp=none},block_all

```

```

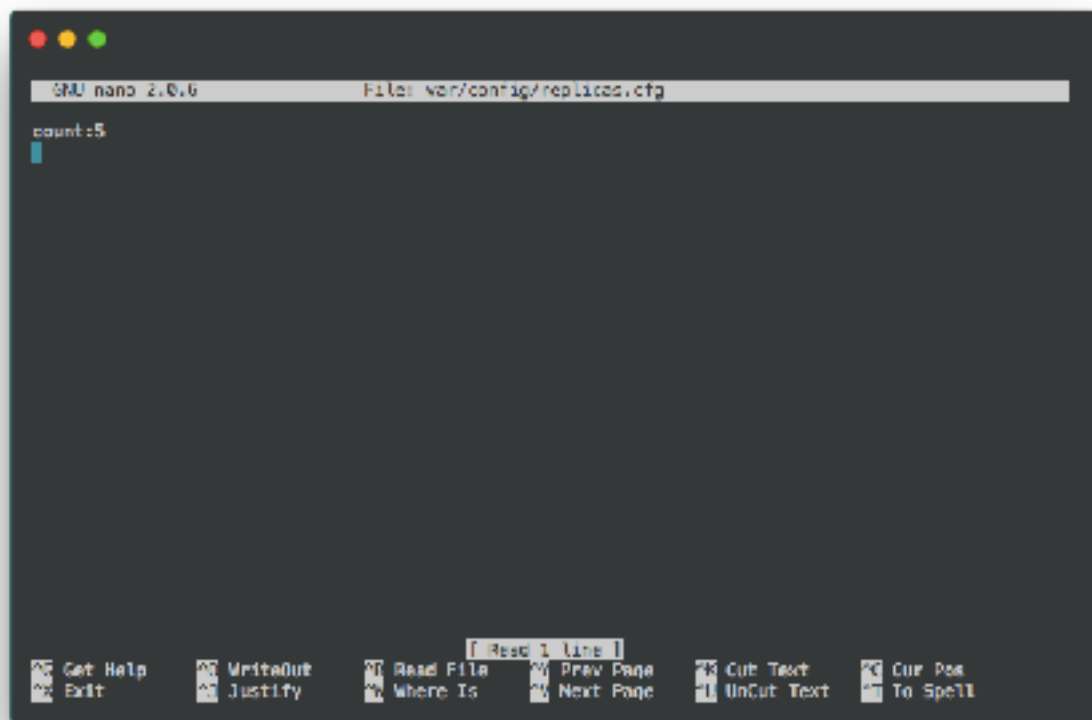
    }
  },
  "nindexes" : 0,
  "totalIndexSize" : 0,
  "indexSizes" : {

  },
  "ok" : 1,
  "operationTime" : Timestamp(1527424066, 1),
  "stats" : {
    "lastOpTime" : Timestamp(0, 0),
    "electionId" : ObjectId("7fffffff0000000000000002")
  },
  "clusterTime" : {
    "clusterTime" : Timestamp(1527424066, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  },
  "configServerState" : {
    "opTime" : {
      "ts" : Timestamp(1527424063, 1),
      "t" : NumberLong(2)
    }
  }
}
cork:PRIMARY> db.oplog.rs.stats().maxSize
201326592
cork:PRIMARY> db.adminCommand({ replSetResizeOplog: 1, size: 25000 })
{
  "ok" : 1,
  "operationTime" : Timestamp(1527424166, 1),
  "stats" : {
    "lastOpTime" : Timestamp(0, 0),
    "electionId" : ObjectId("7fffffff0000000000000002")
  },
  "clusterTime" : {
    "clusterTime" : Timestamp(1527424171, 2),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  },
  "configServerState" : {
    "opTime" : {
      "ts" : Timestamp(1527424171, 2),
      "t" : NumberLong(2)
    }
  }
}
cork:PRIMARY> db.oplog.rs.stats().maxSize
NumberLong("26214400000")
cork:PRIMARY>

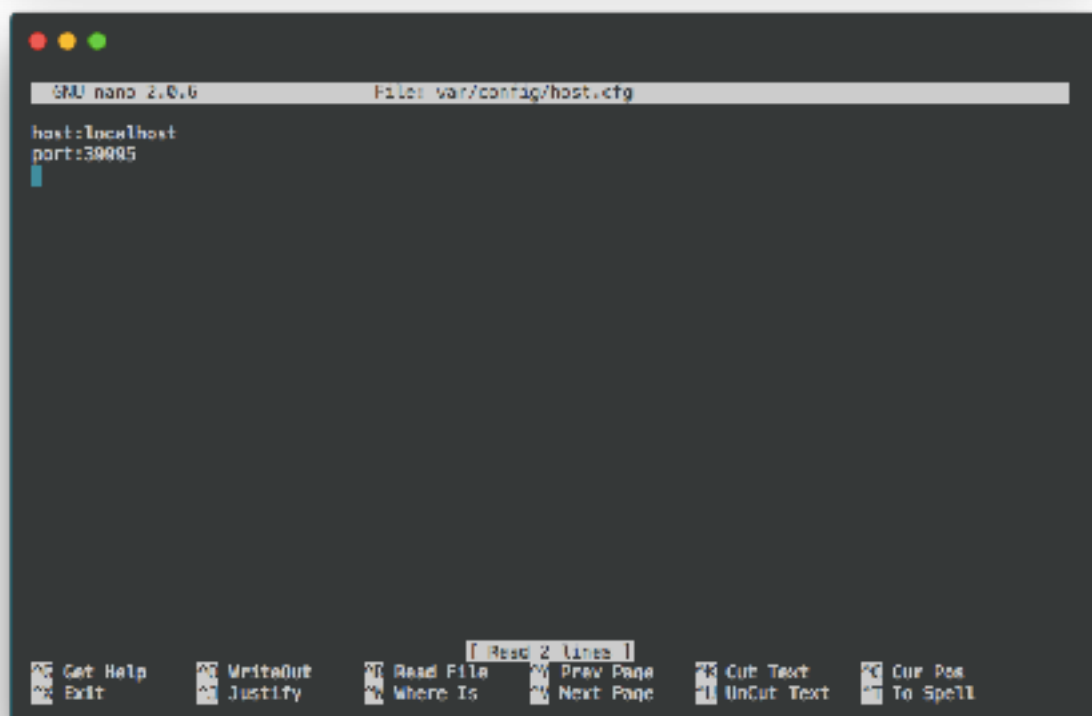
```

Part 2 - Creating an example replica set:

A predefined `/var/config/` folder holds information that my start-up script consumes. When ingested by the script, we will start spawning our mongod nodes for each individual data centre. Code for creating replica sets and clusters can be found inside the compressed zip.



```
GNU nano 2.0.6 File: /var/config/replicas.cfg
count:5
[Resc 1 line 1]
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
```

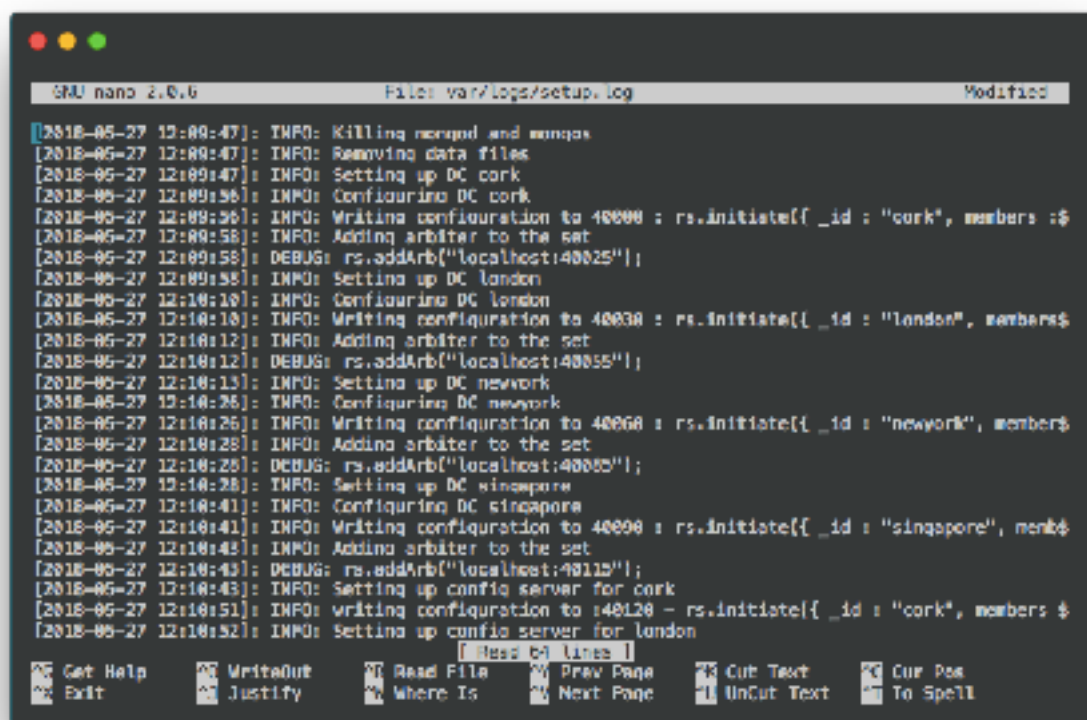


```
GNU nano 2.0.6 File: /var/config/host.cfg
host:localhost
port:39005
[Resc 2 lines 1]
Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where Is Next Page UnCut Text To Spell
```

Per my configuration, we will spawn 5 database nodes, plus an additional arbiter. Once every node has come into existence, we collect all processes and ports and add them to one replica set. The progress is persisted in a log file which became my best friend when debugging the script.

An excerpt from said log is looking like this:

```
[2018-05-27 12:09:56]: INFO: Writing configuration to 40000 :
rs.initiate({ _id : "cork", members : [{ _id : 0, host : "local-
host:40000" }, { _id : 1, host : "localhost:40005" }, { _id : 2,
host : "localhost:40010" }, { _id : 3, host : "localhost:40015" },
{ _id : 4, host : "localhost:40020" } ]})
```



```
GNU nano 2.0.6 File: var/logs/setup.log Modified
[2018-05-27 12:09:47]: INFO: Killing mongod and mongos
[2018-05-27 12:09:47]: INFO: Removing data files
[2018-05-27 12:09:47]: INFO: Setting up DC cork
[2018-05-27 12:09:56]: INFO: Configuring DC cork
[2018-05-27 12:09:56]: INFO: Writing configuration to 40000 : rs.initiate({ _id : "cork", members : $
[2018-05-27 12:09:58]: INFO: Adding arbiter to the set
[2018-05-27 12:09:58]: DEBUG: rs.addArb("localhost:40025");
[2018-05-27 12:09:58]: INFO: Setting up DC london
[2018-05-27 12:10:10]: INFO: Configuring DC london
[2018-05-27 12:10:10]: INFO: Writing configuration to 40030 : rs.initiate({ _id : "london", members $
[2018-05-27 12:10:12]: INFO: Adding arbiter to the set
[2018-05-27 12:10:12]: DEBUG: rs.addArb("localhost:40055");
[2018-05-27 12:10:13]: INFO: Setting up DC newyork
[2018-05-27 12:10:25]: INFO: Configuring DC newyork
[2018-05-27 12:10:25]: INFO: Writing configuration to 40060 : rs.initiate({ _id : "newyork", members $
[2018-05-27 12:10:28]: INFO: Adding arbiter to the set
[2018-05-27 12:10:28]: DEBUG: rs.addArb("localhost:40085");
[2018-05-27 12:10:28]: INFO: Setting up DC singapore
[2018-05-27 12:10:41]: INFO: Configuring DC singapore
[2018-05-27 12:10:41]: INFO: Writing configuration to 40090 : rs.initiate({ _id : "singapore", memb$
[2018-05-27 12:10:43]: INFO: Adding arbiter to the set
[2018-05-27 12:10:43]: DEBUG: rs.addArb("localhost:40115");
[2018-05-27 12:10:43]: INFO: Setting up config server for cork
[2018-05-27 12:10:51]: INFO: Writing configuration to 40120 : rs.initiate({ _id : "cork", members $
[2018-05-27 12:10:52]: INFO: Setting up config server for london
```

Once the replica set is created, we can proceed importing the data set. For this exercise we do it manually:

```
> mongoimport --db data --collection restaurants --drop --file ./
```

```
raw/data.json --port 40000
```

```

host:localhost
port:39995Lab 3 : dominic * -
Lab 3 : dominic * -
Lab 3 : dominic * - cat var/config/replicas.cfg
count:5Lab 3 : dominic * -
Lab 3 : dominic * -
Lab 3 : dominic * -
Lab 3 : dominic * - cat var/config/replicas.cfg
count:5Lab 3 : dominic * - nano var/config/replicas.cfg
Lab 3 : dominic * - nano var/config/host.cfg
Lab 3 : dominic * - nano var/config/node.map
Lab 3 : dominic * - nano var/loca/s
setup.log singapore/
Lab 3 : dominic * - nano var/loca/setup.log
Lab 3 : dominic * - mongoimport --db data --collection restaurants --drop --file ./r
raw/ route.sh run.sh
Lab 3 : dominic * - mongoimport --db data --collection restaurants --drop --file ./raw/data.json --p
ort 40000
2018-05-27T13:02:41.119+0100 connected to: localhost:40000
2018-05-27T13:02:41.123+0100 dropping: data.restaurants
2018-05-27T13:02:44.060+0100 [+++++::.....] data.restaurants 4.13MB/11.3MB (36.4%
)
2018-05-27T13:02:47.058+0100 [+++++::+++++.....] data.restaurants 6.78MB/11.3MB (59.2%
)
2018-05-27T13:02:50.101+0100 [+++++::+++++::+++++...] data.restaurants 10.1MB/11.3MB (89.3%
)
2018-05-27T13:02:52.360+0100 [+++++::+++++::+++++::] data.restaurants 11.3MB/11.3MB (100.0
%)
2018-05-27T13:02:52.360+0100 imported 25359 documents
Lab 3 : dominic * -

```

Part 3 - Multiple replica sets

A single replica set does not match our given scenario and therefore, please find screenshots that show the `rs.status()` across all replica sets.

Replica set #1: CORK

```
Lab 3 : dominic ~ - mongo --port 40000
MongoDB shell version v3.6.4
connecting to: mongodb://127.0.0.1:40000/
MongoDB server version: 3.6.4
Server has startup warnings:
2018-05-27T14:01:45.551+0100 I CONTROL [initandlisten]
2018-05-27T14:01:45.551+0100 I CONTROL [initandlisten] ** WARNING: Access control is not enabled fo
r the database.
2018-05-27T14:01:45.551+0100 I CONTROL [initandlisten] **          Read and write access to data an
d configuration is unrestricted.
2018-05-27T14:01:45.552+0100 I CONTROL [initandlisten]
2018-05-27T14:01:45.552+0100 I CONTROL [initandlisten] ** WARNING: This server is bound to localhos
t.
2018-05-27T14:01:45.552+0100 I CONTROL [initandlisten] **          Remote systems will be unable to
connect to this server.
2018-05-27T14:01:45.552+0100 I CONTROL [initandlisten] **          Start the server with --bind_ip
<address> to specify which IP
2018-05-27T14:01:45.552+0100 I CONTROL [initandlisten] **          addresses it should serve respon
ses from, or with --bind_ip_all to
2018-05-27T14:01:45.552+0100 I CONTROL [initandlisten] **          bind to all interfaces. If this
behavior is desired, start the
2018-05-27T14:01:45.552+0100 I CONTROL [initandlisten] **          server with --bind_ip 127.0.0.1
to disable this warning.
2018-05-27T14:01:45.552+0100 I CONTROL [initandlisten]
cork:PRIMARY> rs.status()
{
  "set" : "cork",
  "date" : ISODate("2018-05-27T13:05:00.428Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1527426299, 1),
      "t" : NumberLong(1)
    },
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1527426299, 1),
      "t" : NumberLong(1)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1527426299, 1),
      "t" : NumberLong(1)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1527426299, 1),
      "t" : NumberLong(1)
    }
  }
}
```

```
"members" : [
  {
    "_id" : 0,
    "name" : "localhost:40000",
    "health" : 1,
    "state" : 1,
    "stateStr" : "PRIMARY",
    "uptime" : 196,
    "optime" : {
      "ts" : Timestamp(1527426299, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2018-05-27T13:04:59Z"),
    "electionTime" : Timestamp(1527426127, 1),
    "electionDate" : ISODate("2018-05-27T13:02:07Z"),
    "configVersion" : 1,
    "self" : true
  },
  {
    "_id" : 1,
    "name" : "localhost:40005",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 184,
    "optime" : {
      "ts" : Timestamp(1527426299, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1527426299, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2018-05-27T13:04:59Z"),
    "optimeDurableDate" : ISODate("2018-05-27T13:04:59Z"),
    "lastHeartbeat" : ISODate("2018-05-27T13:05:00.189Z"),
    "lastHeartbeatRecv" : ISODate("2018-05-27T13:04:59.767Z"),
    "pingMs" : NumberLong(0),
    "syncingTo" : "localhost:40000",
    "configVersion" : 1
  },
  {
    "_id" : 2,
    "name" : "localhost:40010",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 184,
    "optime" : {
```



```
{
  "_id" : 2,
  "name" : "localhost:40010",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 184,
  "optime" : {
    "ts" : Timestamp(1527426299, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1527426299, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-05-27T13:04:59Z"),
  "optimeDurableDate" : ISODate("2018-05-27T13:04:59Z"),
  "lastHeartbeat" : ISODate("2018-05-27T13:05:00.188Z"),
  "lastHeartbeatRecv" : ISODate("2018-05-27T13:04:58.454Z"),
  "pingMs" : NumberLong(0),
  "syncingTo" : "localhost:40000",
  "configVersion" : 1
},
{
  "_id" : 3,
  "name" : "localhost:40015",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 184,
  "optime" : {
    "ts" : Timestamp(1527426299, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1527426299, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-05-27T13:04:59Z"),
  "optimeDurableDate" : ISODate("2018-05-27T13:04:59Z"),
  "lastHeartbeat" : ISODate("2018-05-27T13:05:00.189Z"),
  "lastHeartbeatRecv" : ISODate("2018-05-27T13:05:00.207Z"),
  "pingMs" : NumberLong(0),
  "syncingTo" : "localhost:40000",
  "configVersion" : 1
},
{
  "_id" : 4,
  "name" : "localhost:40020",
```

```

        "pingMs" : NumberLong(0),
        "syncingTo" : "localhost:40000",
        "configVersion" : 1
    },
    {
        "_id" : 4,
        "name" : "localhost:40020",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 184,
        "optime" : {
            "ts" : Timestamp(1527426299, 1),
            "t" : NumberLong(1)
        },
        "optimeDurable" : {
            "ts" : Timestamp(1527426299, 1),
            "t" : NumberLong(1)
        },
        "optimeDate" : ISODate("2018-05-27T13:04:59Z"),
        "optimeDurableDate" : ISODate("2018-05-27T13:04:59Z"),
        "lastHeartbeat" : ISODate("2018-05-27T13:05:00.188Z"),
        "lastHeartbeatRecv" : ISODate("2018-05-27T13:05:00.215Z"),
        "pingMs" : NumberLong(0),
        "syncingTo" : "localhost:40000",
        "configVersion" : 1
    }
],
"ok" : 1,
"operationTime" : Timestamp(1527426299, 1),
"soleStats" : {
    "lastOpTime" : Timestamp(0, 0),
    "electionId" : ObjectId("7fffffff0000000000000001")
},
"$clusterTime" : {
    "clusterTime" : Timestamp(1527426299, 1),
    "signature" : {
        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAA"),
        "keyId" : NumberLong(0)
    }
},
"sconfigServerState" : {
    "opTime" : {
        "ts" : Timestamp(1527426299, 1),
        "t" : NumberLong(1)
    }
}
}
cork:PRIMARY>

```

Replica set #2: LONDON

```
Lab 3 : dominic * - mongo --port 40030
MongoDB shell version v3.6.4
connecting to: mongodb://127.0.0.1:40030/
MongoDB server version: 3.6.4
Server has startup warnings:
2018-05-27T14:01:59.355+0100 I CONTROL [initandlisten]
2018-05-27T14:01:59.375+0100 I CONTROL [initandlisten] ** WARNING: Access control is not enabled fo
r the database.
2018-05-27T14:01:59.393+0100 I CONTROL [initandlisten] **          Read and write access to data an
d configuration is unrestricted.
2018-05-27T14:01:59.394+0100 I CONTROL [initandlisten]
2018-05-27T14:01:59.415+0100 I CONTROL [initandlisten] ** WARNING: This server is bound to localhos
t.
2018-05-27T14:01:59.426+0100 I CONTROL [initandlisten] **          Remote systems will be unable to
connect to this server.
2018-05-27T14:01:59.437+0100 I CONTROL [initandlisten] **          Start the server with --bind_ip
<address> to specify which IP
2018-05-27T14:01:59.456+0100 I CONTROL [initandlisten] **          addresses it should serve respon
ses from, or with --bind_ip_all to
2018-05-27T14:01:59.476+0100 I CONTROL [initandlisten] **          bind to all interfaces. If this
behavior is desired, start the
2018-05-27T14:01:59.497+0100 I CONTROL [initandlisten] **          server with --bind_ip 127.0.0.1
to disable this warning.
2018-05-27T14:01:59.498+0100 I CONTROL [initandlisten]
london:PRIMARY> rs.status()
{
  "set" : "london",
  "date" : ISODate("2018-05-27T13:14:12.658Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1527426844, 1),
      "t" : NumberLong(1)
    },
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1527426844, 1),
      "t" : NumberLong(1)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1527426844, 1),
      "t" : NumberLong(1)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1527426844, 1),
      "t" : NumberLong(1)
    }
  },
  "members" : [
    {
      "_id" : 0,
      "name" : "localhost:40030",
      "health" : 1,
```

```
"members" : [
  {
    "_id" : 0,
    "name" : "localhost:40030",
    "health" : 1,
    "state" : 1,
    "stateStr" : "PRIMARY",
    "uptime" : 735,
    "optime" : {
      "ts" : Timestamp(1527426844, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2018-05-27T13:14:04Z"),
    "electionTime" : Timestamp(1527426142, 1),
    "electionDate" : ISODate("2018-05-27T13:02:22Z"),
    "configVersion" : 1,
    "self" : true
  },
  {
    "_id" : 1,
    "name" : "localhost:40035",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 721,
    "optime" : {
      "ts" : Timestamp(1527426844, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1527426844, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2018-05-27T13:14:04Z"),
    "optimeDurableDate" : ISODate("2018-05-27T13:14:04Z"),
    "lastHeartbeat" : ISODate("2018-05-27T13:14:11.908Z"),
    "lastHeartbeatRecv" : ISODate("2018-05-27T13:14:12.066Z"),
    "pingMs" : NumberLong(0),
    "syncingTo" : "localhost:40030",
    "configVersion" : 1
  },
  {
    "_id" : 2,
    "name" : "localhost:40040",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 721,
    "optime" : {
      "ts" : Timestamp(1527426844, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1527426844, 1),
```

```
{
  "_id" : 2,
  "name" : "localhost:40040",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 721,
  "optime" : {
    "ts" : Timestamp(1527426844, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1527426844, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-05-27T13:14:04Z"),
  "optimeDurableDate" : ISODate("2018-05-27T13:14:04Z"),
  "lastHeartbeat" : ISODate("2018-05-27T13:14:11.908Z"),
  "lastHeartbeatRecv" : ISODate("2018-05-27T13:14:12.063Z"),
  "pingMs" : NumberLong(0),
  "syncingTo" : "localhost:40030",
  "configVersion" : 1
},
{
  "_id" : 3,
  "name" : "localhost:40045",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 721,
  "optime" : {
    "ts" : Timestamp(1527426844, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1527426844, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-05-27T13:14:04Z"),
  "optimeDurableDate" : ISODate("2018-05-27T13:14:04Z"),
  "lastHeartbeat" : ISODate("2018-05-27T13:14:11.908Z"),
  "lastHeartbeatRecv" : ISODate("2018-05-27T13:14:12.353Z"),
  "pingMs" : NumberLong(0),
  "syncingTo" : "localhost:40030",
  "configVersion" : 1
},
{
  "_id" : 4,
  "name" : "localhost:40050",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 721,
  "optime" : {
```

```

    },
    "optimeDate" : ISODate("2018-05-27T13:14:04Z"),
    "optimeDurableDate" : ISODate("2018-05-27T13:14:04Z"),
    "lastHeartbeat" : ISODate("2018-05-27T13:14:11.908Z"),
    "lastHeartbeatRecv" : ISODate("2018-05-27T13:14:12.353Z"),
    "pingNs" : NumberLong(0),
    "syncingTo" : "localhost:40030",
    "configVersion" : 1
  },
  {
    "_id" : 4,
    "name" : "localhost:40050",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 721,
    "optime" : {
      "ts" : Timestamp(1527426844, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1527426844, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2018-05-27T13:14:04Z"),
    "optimeDurableDate" : ISODate("2018-05-27T13:14:04Z"),
    "lastHeartbeat" : ISODate("2018-05-27T13:14:11.908Z"),
    "lastHeartbeatRecv" : ISODate("2018-05-27T13:14:12.453Z"),
    "pingNs" : NumberLong(0),
    "syncingTo" : "localhost:40030",
    "configVersion" : 1
  }
],
"ok" : 1,
"operationTime" : Timestamp(1527426844, 1),
"$aleStats" : {
  "lastOpTime" : Timestamp(0, 0),
  "electionId" : ObjectId("7fffffff0000000000000001")
},
"$clusterTime" : {
  "clusterTime" : Timestamp(1527426844, 1),
  "signature" : {
    "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
    "keyId" : NumberLong(0)
  }
},
"$configServerState" : {
  "opTime" : {
    "ts" : Timestamp(1527426837, 1),
    "t" : NumberLong(1)
  }
}
}
London:PRIMARY>

```

Replica set #3: NEWYORK

```

Lab 3 : dominic @ - mongo --port 40060
MongoDB shell version v3.6.4
connecting to: mongodb://127.0.0.1:40060/
MongoDB server version: 3.6.4
Server has startup warnings:
2018-05-27T14:02:14.554+0100 I CONTROL [initandlisten]
2018-05-27T14:02:14.555+0100 I CONTROL [initandlisten] ** WARNING: Access control is not enabled fo
r the database.
2018-05-27T14:02:14.574+0100 I CONTROL [initandlisten] **          Read and write access to data an
d configuration is unrestricted.
2018-05-27T14:02:14.579+0100 I CONTROL [initandlisten]
2018-05-27T14:02:14.599+0100 I CONTROL [initandlisten] ** WARNING: This server is bound to localhos
t.
2018-05-27T14:02:14.607+0100 I CONTROL [initandlisten] **          Remote systems will be unable to
connect to this server.
2018-05-27T14:02:14.628+0100 I CONTROL [initandlisten] **          Start the server with --bind_ip
<address> to specify which IP
2018-05-27T14:02:14.637+0100 I CONTROL [initandlisten] **          addresses it should serve respon
ses from, or with --bind_ip_all to
2018-05-27T14:02:14.657+0100 I CONTROL [initandlisten] **          bind to all interfaces. If this
behavior is desired, start the
2018-05-27T14:02:14.667+0100 I CONTROL [initandlisten] **          server with --bind_ip 127.0.0.1
to disable this warning.
2018-05-27T14:02:14.719+0100 I CONTROL [initandlisten]
newyork:PRIMARY> rs.status()
{
  "set" : "newyork",
  "date" : ISODate("2018-05-27T13:22:36.711Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1527427349, 1),
      "t" : NumberLong(1)
    },
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1527427349, 1),
      "t" : NumberLong(1)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1527427349, 1),
      "t" : NumberLong(1)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1527427349, 1),
      "t" : NumberLong(1)
    }
  },
  "members" : [
    {
      "_id" : 0,
      "name" : "localhost:40060",
      "health" : 1,

```



```
"members" : [
  {
    "_id" : 0,
    "name" : "localhost:40060",
    "health" : 1,
    "state" : 1,
    "stateStr" : "PRIMARY",
    "uptime" : 1226,
    "optime" : {
      "ts" : Timestamp(1527427349, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2018-05-27T13:22:29Z"),
    "electionTime" : Timestamp(1527426156, 1),
    "electionDate" : ISODate("2018-05-27T13:02:38Z"),
    "configVersion" : 1,
    "self" : true
  },
  {
    "_id" : 1,
    "name" : "localhost:40065",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 1211,
    "optime" : {
      "ts" : Timestamp(1527427349, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1527427349, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2018-05-27T13:22:29Z"),
    "optimeDurableDate" : ISODate("2018-05-27T13:22:29Z"),
    "lastHeartbeat" : ISODate("2018-05-27T13:22:38.294Z"),
    "lastHeartbeatRecv" : ISODate("2018-05-27T13:22:38.491Z"),
    "pingMs" : NumberLong(0),
    "syncingTo" : "localhost:40060",
    "configVersion" : 1
  },
  {
    "_id" : 2,
    "name" : "localhost:40070",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 1211,
    "optime" : {
      "ts" : Timestamp(1527427349, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1527427349, 1),
```



```
{
  "_id" : 2,
  "name" : "localhost:40070",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 1211,
  "optime" : {
    "ts" : Timestamp(1527427349, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1527427349, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-05-27T13:22:29Z"),
  "optimeDurableDate" : ISODate("2018-05-27T13:22:29Z"),
  "lastHeartbeat" : ISODate("2018-05-27T13:22:38.294Z"),
  "lastHeartbeatRecv" : ISODate("2018-05-27T13:22:38.393Z"),
  "pingMs" : NumberLong(0),
  "syncingTo" : "localhost:40060",
  "configVersion" : 1
},
{
  "_id" : 3,
  "name" : "localhost:40075",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 1211,
  "optime" : {
    "ts" : Timestamp(1527427349, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1527427349, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-05-27T13:22:29Z"),
  "optimeDurableDate" : ISODate("2018-05-27T13:22:29Z"),
  "lastHeartbeat" : ISODate("2018-05-27T13:22:38.294Z"),
  "lastHeartbeatRecv" : ISODate("2018-05-27T13:22:36.965Z"),
  "pingMs" : NumberLong(0),
  "syncingTo" : "localhost:40070",
  "configVersion" : 1
},
{
  "_id" : 4,
  "name" : "localhost:40080",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 1211,
  "optime" : {
```

```

    },
    "optimeDate" : ISODate("2018-05-27T13:22:29Z"),
    "optimeDurableDate" : ISODate("2018-05-27T13:22:29Z"),
    "lastHeartbeat" : ISODate("2018-05-27T13:22:38.294Z"),
    "lastHeartbeatRecv" : ISODate("2018-05-27T13:22:36.965Z"),
    "pingMs" : NumberLong(0),
    "syncingTo" : "localhost:40070",
    "configVersion" : 1
  },
  {
    "_id" : 4,
    "name" : "localhost:40060",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 1211,
    "optime" : {
      "ts" : Timestamp(1527427349, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1527427349, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2018-05-27T13:22:29Z"),
    "optimeDurableDate" : ISODate("2018-05-27T13:22:29Z"),
    "lastHeartbeat" : ISODate("2018-05-27T13:22:38.294Z"),
    "lastHeartbeatRecv" : ISODate("2018-05-27T13:22:37.098Z"),
    "pingMs" : NumberLong(0),
    "syncingTo" : "localhost:40070",
    "configVersion" : 1
  }
],
"ok" : 1,
"operationTime" : Timestamp(1527427349, 1),
"SoleStats" : {
  "lastOpTime" : Timestamp(0, 0),
  "electionId" : ObjectId("7fffffff0000000000000001")
},
"ScclusterTime" : {
  "clusterTime" : Timestamp(1527427354, 1),
  "signature" : {
    "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
    "keyId" : NumberLong(0)
  }
},
"SconfiaServerState" : {
  "opTime" : {
    "ts" : Timestamp(1527427347, 1),
    "t" : NumberLong(1)
  }
}
}
newyork:PRIMARY>

```

Replica set #4: SINGAPORE

```

Lab 3 : dominic ~ - mongo --port 40090
MongoDB shell version v3.6.4
connecting to: mongod://127.0.0.1:40090/
MongoDB server version: 3.6.4
Server has startup warnings:
2018-05-27T14:02:30.226+0100 I CONTROL [initandlisten]
2018-05-27T14:02:30.226+0100 I CONTROL [initandlisten] ** WARNING: Access control is not enabled fo
r the database.
2018-05-27T14:02:30.226+0100 I CONTROL [initandlisten] **          Read and write access to data an
d configuration is unrestricted.
2018-05-27T14:02:30.227+0100 I CONTROL [initandlisten]
2018-05-27T14:02:30.227+0100 I CONTROL [initandlisten] ** WARNING: This server is bound to localhos
t.
2018-05-27T14:02:30.227+0100 I CONTROL [initandlisten] **          Remote systems will be unable to
connect to this server.
2018-05-27T14:02:30.227+0100 I CONTROL [initandlisten] **          Start the server with --bind_ip
<address> to specify which IP
2018-05-27T14:02:30.227+0100 I CONTROL [initandlisten] **          addresses it should serve respon
ses from, or with --bind_ip_all to
2018-05-27T14:02:30.227+0100 I CONTROL [initandlisten] **          bind to all interfaces. If this
behavior is desired, start the
2018-05-27T14:02:30.227+0100 I CONTROL [initandlisten] **          server with --bind_ip 127.0.0.1
to disable this warning.
2018-05-27T14:02:30.227+0100 I CONTROL [initandlisten]
singapore:PRIMARY> rs.status()
{
  "set" : "singapore",
  "date" : ISODate("2018-05-27T13:24:20.037Z"),
  "myState" : 1,
  "term" : NumberLong(1),
  "heartbeatIntervalMillis" : NumberLong(2000),
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1527427454, 1),
      "t" : NumberLong(1)
    },
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1527427454, 1),
      "t" : NumberLong(1)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1527427454, 1),
      "t" : NumberLong(1)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1527427454, 1),
      "t" : NumberLong(1)
    }
  },
  "members" : [
    {
      "_id" : 0,
      "name" : "localhost:40090",
      "health" : 1,

```

```
"members" : [
  {
    "_id" : 0,
    "name" : "localhost:40090",
    "health" : 1,
    "state" : 1,
    "stateStr" : "PRIMARY",
    "uptime" : 1312,
    "optime" : {
      "ts" : Timestamp(1527427454, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2018-05-27T13:24:14Z"),
    "electionTime" : Timestamp(1527426172, 1),
    "electionDate" : ISODate("2018-05-27T13:02:52Z"),
    "configVersion" : 1,
    "self" : true
  },
  {
    "_id" : 1,
    "name" : "localhost:40095",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 1298,
    "optime" : {
      "ts" : Timestamp(1527427454, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1527427454, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2018-05-27T13:24:14Z"),
    "optimeDurableDate" : ISODate("2018-05-27T13:24:14Z"),
    "lastHeartbeat" : ISODate("2018-05-27T13:24:19.091Z"),
    "lastHeartbeatRecv" : ISODate("2018-05-27T13:24:19.974Z"),
    "pingMs" : NumberLong(0),
    "syncingTo" : "localhost:40090",
    "configVersion" : 1
  },
  {
    "_id" : 2,
    "name" : "localhost:40100",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 1298,
    "optime" : {
      "ts" : Timestamp(1527427454, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1527427454, 1),
```

```
{
  "_id" : 2,
  "name" : "localhost:40100",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 1298,
  "optime" : {
    "ts" : Timestamp(1527427454, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1527427454, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-05-27T13:24:14Z"),
  "optimeDurableDate" : ISODate("2018-05-27T13:24:14Z"),
  "lastHeartbeat" : ISODate("2018-05-27T13:24:19.091Z"),
  "lastHeartbeatRecv" : ISODate("2018-05-27T13:24:19.346Z"),
  "pingMs" : NumberLong(0),
  "syncingTo" : "localhost:40090",
  "configVersion" : 1
},
{
  "_id" : 3,
  "name" : "localhost:40105",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 1298,
  "optime" : {
    "ts" : Timestamp(1527427454, 1),
    "t" : NumberLong(1)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1527427454, 1),
    "t" : NumberLong(1)
  },
  "optimeDate" : ISODate("2018-05-27T13:24:14Z"),
  "optimeDurableDate" : ISODate("2018-05-27T13:24:14Z"),
  "lastHeartbeat" : ISODate("2018-05-27T13:24:19.092Z"),
  "lastHeartbeatRecv" : ISODate("2018-05-27T13:24:19.357Z"),
  "pingMs" : NumberLong(0),
  "syncingTo" : "localhost:40090",
  "configVersion" : 1
},
{
  "_id" : 4,
  "name" : "localhost:40110",
  "health" : 1,
  "state" : 2,
  "stateStr" : "SECONDARY",
  "uptime" : 1296,
  "optime" : {
```

```

    },
    "optimeDate" : ISODate("2018-05-27T13:24:14Z"),
    "optimeDurableDate" : ISODate("2018-05-27T13:24:14Z"),
    "lastHeartbeat" : ISODate("2018-05-27T13:24:19.092Z"),
    "lastHeartbeatRecv" : ISODate("2018-05-27T13:24:19.357Z"),
    "pingMs" : NumberLong(0),
    "syncingTo" : "localhost:40090",
    "configVersion" : 1
  },
  {
    "_id" : 4,
    "name" : "localhost:40110",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 1298,
    "optime" : {
      "ts" : Timestamp(1527427454, 1),
      "t" : NumberLong(1)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1527427454, 1),
      "t" : NumberLong(1)
    },
    "optimeDate" : ISODate("2018-05-27T13:24:14Z"),
    "optimeDurableDate" : ISODate("2018-05-27T13:24:14Z"),
    "lastHeartbeat" : ISODate("2018-05-27T13:24:19.091Z"),
    "lastHeartbeatRecv" : ISODate("2018-05-27T13:24:19.108Z"),
    "pingMs" : NumberLong(0),
    "syncingTo" : "localhost:40090",
    "configVersion" : 1
  }
],
"ok" : 1,
"operationTime" : Timestamp(1527427454, 1),
"SoleStats" : {
  "lastOpTime" : Timestamp(0, 0),
  "electionId" : ObjectId("7fffffff0000000000000001")
},
"clusterTime" : {
  "clusterTime" : Timestamp(1527427459, 1),
  "signature" : {
    "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAA="),
    "keyId" : NumberLong(0)
  }
},
"SconfiaServerState" : {
  "opTime" : {
    "ts" : Timestamp(1527427459, 1),
    "t" : NumberLong(1)
  }
}
}
singapore:PRIMARY>

```

Part 4 - Sharding results (ref screencast)**Replica set #1: CORK**

```

cork:PRIMARY> use data
switched to db data
cork:PRIMARY> db.getCollection("restaurants").stats()
{
  "ns" : "data.restaurants",
  "size" : 5414468,
  "count" : 12821,
  "avgObjSize" : 422,
  "storageSize" : 2383972,
  "capped" : false,
  "wiredTiger" : {
    "metadata" : {
      "formatVersion" : 1
    },
    "creationString" : "access_pattern_hint=none,allocation_size=4KB,app_metadata=(formatVersion=1),assort=(commit_timestamp=none,read_timestamp=none),block_allocation=best,block_compressor=snappy,cache_resident=false,checksum=on,collgroups=,collator=,columns=,dictionary=0,encryption=(keyid=,name=),exclusive=false,extractor=,format=btree,huffman_keys=,huffman_values=,ignore_in_memory_cache_size=false,immutable=false,internal_item_max=0,internal_key_max=0,internal_key_truncate=true,internal_page_max=4KB,key_format=q,key_gap=10,leaf_item_max=0,leaf_key_max=0,leaf_page_max=32KB,leaf_value_max=64MB,log=(enabled=true),lsn=(auto_throttle=true,bloom=true,bloom_bit_count=16,bloom_config=,bloom_hash_count=8,bloom_oldest=false,chunk_count_limit=0,chunk_max=5GB,chunk_size=10MB,merge_custom=[prefix,start_generation=0,suffix=],merge_max=15,merge_min=0),memory_page_max=10n,os_cache_dirty_max=0,os_cache_max=0,prefix_compression=false,prefix_compression_min=4,source=,split_deepen_min_child=0,split_deepen_per_child=0,split_pct=90,type=file,value_format=u",
    "type" : "file",
    "uri" : "statistics:table:collection-22-6255001369053508746",
    "LSN" : {
      "bloom filter false positives" : 0,
      "bloom filter hits" : 0,
      "bloom filter misses" : 0,
      "bloom filter pages evicted from cache" : 0,
      "bloom filter pages read into cache" : 0,
      "bloom filters in the LSM tree" : 0,
      "chunks in the LSM tree" : 0,
      "highest merge generation in the LSM tree" : 0,
      "queries that could have benefited from a Bloom filter that did not exist" : 0,
      "sleep for LSM checkpoint throttle" : 0,
      "sleep for LSM merge throttle" : 0,
      "total size of bloom filters" : 0
    },
    "block-manager" : {

```


Replica set #2: LONDON

```

2016-05-30T17:31:59.321+0100 I CONTROL [initandlisten]
London:PRIMARY> use data
switched to db data
London:PRIMARY> db.getCollection("restaurants").stats()
{
  "ns" : "data.restaurants",
  "size" : 959067,
  "count" : 2225,
  "avgObjSize" : 422,
  "storageSize" : 425884,
  "capped" : false,
  "wiredTiger" : {
    "metadata" : {
      "formatVersion" : 1
    },
    "creationString" : "access_pattern_hint=none,allocation_size=4KB,app_metadata=(formatVersion=1),assert=(commit_timestamp=none,read_timestamp=none),block_allocation=best,block_compressor=snappy,cache_resident=false,checksum=on,colgroups=,collator=,columns=,dictionary=0,encryption=(keyid=,none=),exclusive=false,extractors=,format=btree,huffman_key=,huffman_value=,ignore_in_memory_cache_size=false,immutable=false,internal_item_max=0,internal_key_max=0,internal_key_truncate=true,internal_page_max=4KB,key_format=q,key_gap=10,leaf_item_max=0,leaf_key_max=0,leaf_page_max=32KB,leaf_value_max=64MB,log=(enabled=true),lsn=(auto_throttle=true,bloom=true,bloom_bit_count=16,bloom_config=,bloom_hash_count=0,bloom_oldest=false,chunk_count_limit=0,chunk_max=5GB,chunk_size=10MB,merge_custom=(prefix=,start_generation=0,suffix=),merge_max=15,merge_min=0),memory_page_max=10n,os_cache_dirty_max=0,os_cache_max=0,prefix_compression=false,prefix_compression_min=4,sources=,split_deepen_min_child=0,split_deepen_per_child=0,split_pct=90,type=file,value_format=u",
    "type" : "file",
    "uri" : "statistics:tables:collection-21-641867172228159613",
    "LSN" : {
      "bloom filter false positives" : 0,
      "bloom filter hits" : 0,
      "bloom filter misses" : 0,
      "bloom filter pages evicted from cache" : 0,
      "bloom filter pages read into cache" : 0,
      "bloom filters in the LSN tree" : 0,
      "chunks in the LSN tree" : 0,
      "highest merge generation in the LSN tree" : 0,
      "queries that could have benefited from a Bloom filter that did not exist" : 0,
      "sleep for LSN checkpoint throttle" : 0,
      "sleep for LSN merge throttle" : 0,
      "total size of bloom filters" : 0
    }
  },

```


Replica set #3: NEWYORK

```

2018-05-30T17:32:11.603+0100 I CONTROL [initandlisten]
newyork:PRIMARY> use data
switched to db data
newyork:PRIMARY> db.getCollection("restaurants").stats()
{
  "ns" : "data.restaurants",
  "size" : 1984878,
  "count" : 4895,
  "avgObjSize" : 401,
  "storageSize" : 876544,
  "capped" : false,
  "wiredTiger" : {
    "metadata" : {
      "formatVersion" : 1
    },
    "creationString" : "access_pattern_hint=none,allocation_size=4KB,app_metadata=(formatVersion=1),asort=(commit_timestamp=none,read_timestamp=none),block_allocation=best,block_compressor=snappy,cache_resident=false,checksum=on,colgroups=,collator=,columns=,dictionary=0,encryption=(keyid=,none=),exclusive=false,extractors=,format=btree,huffman_key=,huffman_value=,ignore_in_memory_cache_size=false,immutable=false,internal_item_max=0,internal_key_max=0,internal_key_truncate=true,internal_page_max=4KB,key_format=q,key_gap=10,leaf_item_max=0,leaf_key_max=0,leaf_page_max=32KB,leaf_value_max=64MB,log=(enabled=true),lsn=(auto_throttle=true,bloom=true,bloom_bit_count=16,bloom_config=,bloom_hash_count=0,bloom_oldest=false,chunk_count_limit=0,chunk_max=5GB,chunk_size=10MB,merge_cursor=(prefix=,start_generation=0,suffix=),merge_max=15,merge_min=0),memory_page_max=100,os_cache_dirty_max=0,os_cache_max=0,prefix_compression=false,prefix_compression_min=4,sources=,split_deepen_min_child=0,split_deepen_per_child=0,split_pct=90,type=file,value_format=u",
    "type" : "file",
    "uri" : "statistics:table:collection-21-2173104882446639500",
    "LSN" : {
      "bloom filter false positives" : 0,
      "bloom filter hits" : 0,
      "bloom filter misses" : 0,
      "bloom filter pages evicted from cache" : 0,
      "bloom filter pages read into cache" : 0,
      "bloom filters in the LSM tree" : 0,
      "chunks in the LSM tree" : 0,
      "highest merge generation in the LSM tree" : 0,
      "queries that could have benefited from a Bloom filter that did not exist" : 0,
      "sleep for LSM checkpoint throttle" : 0,
      "sleep for LSM merge throttle" : 0,
      "total size of bloom filters" : 0
    }
  },

```

Replica set #4: SINGAPORE

```
is warning.
2018-05-30T17:32:24.000+0100 I CONTROL [initandlisten]
singapore:PRIMARY> use data
switched to db data
singapore:PRIMARY> db.getCollection("restaurants").stats()
{
  "ns" : "data.restaurants",
  "size" : 2311401,
  "count" : 5418,
  "avgObjSize" : 426,
  "storageSize" : 1828896,
  "capped" : false,
  "wiredTiger" : {
    "metadata" : {
      "formatVersion" : 1
    },
    "creationString" : "access_pattern_hint=none,allocation_size=4KB,app_metadata=(formatVersion=1),assert=(commit_timestamp=none,read_timestamp=none),block_allocation=best,block_compressor=snappy,cache_resident=false,checksum=on,colgroups=,collator=,columns=,dictionary=0,encryption=(keyid=,name=),exclusive=false,extractor=,format=btree,huffman_keys=huffman_values=,ignore_in_memory_cache_size=false,immutable=false,internal_item_max=0,internal_key_max=0,internal_key_truncate=true,internal_page_max=4KB,key_format=q,key_gap=10,leaf_item_max=0,leaf_key_max=0,leaf_page_max=32KB,leaf_value_max=64KB,log=(enabled=true),lsm=(auto_throttle=true,bloom=true,bloom_bit_count=16,bloom_config=,bloom_hash_count=8,bloom_oldest=false,chunk_count_limit=0,chunk_max=5GB,chunk_size=10MB,merge_custom=(prefix=,start_generation=0,suffix=),merge_max=15,merge_min=0),memory_page_max=10n,os_cache_dirty_max=0,os_cache_max=0,prefix_compression=false,prefix_compression_min=4,source=,split_deepen_min_child=0,split_deepen_per_child=0,split_pct=90,type=file,value_format=u",
    "type" : "file",
    "uri" : "statistics:table:collection-28-54334582393764229",
    "LSM" : {
      "bloom filter false positives" : 0,
      "bloom filter hits" : 0,
      "bloom filter misses" : 0,
      "bloom filter pages evicted from cache" : 0,
      "bloom filter pages read into cache" : 0,
      "bloom filters in the LSM tree" : 0,
      "chunks in the LSM tree" : 0,
      "highest merge generation in the LSM tree" : 0,
      "queries that could have benefited from a Bloom filter that did not exist" : 0,
      "sleep for LSM checkpoint throttle" : 0,
      "sleep for LSM merge throttle" : 0,
      "total size of bloom filters" : 0
    }
  }
}
```