

1. Set up Shards

Shard Controller	mongos	27021	N/A
Config Server	mongod	27022	/db/config/data
Shard0	mongod	27023	/db/shard0/data
Shard1	mongod	27024	/db/shard1/data

2. Set up a config replica set

(a) Create a directory

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mkdir -p /db/config/data
```

(b) Start a mongod serving as a config server

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongod --port 27022 --dbpath /db/config/data --  
configsvr --replSet config
```

(c) In a new window, initialize the replica set

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongo --port 27022
```

```
> rs.initiate()  
{  
  "info2" : "no configuration specified. Using a default configuration for the set",  
  "me" : "localhost:27022",  
  "ok" : 1,  
  "operationTime" : Timestamp(1552887471, 1),  
  "$gleStats" : {  
    "lastOpTime" : Timestamp(1552887471, 1),  
    "electionId" : ObjectId("000000000000000000000000")  
  },  
  "lastCommittedOpTime" : Timestamp(0, 0),  
  "$clusterTime" : {  
    "clusterTime" : Timestamp(1552887471, 1),  
    "signature" : {  
      "hash" : BinData(0, "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),  
      "keyId" : NumberLong(0)  
    }  
  }  
}
```

3. Set up the shard controller (mongos) in a new window

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongos --configdb config/127.0.0.1:27022 --port  
27021
```

Set the chunk size:

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongo 127.0.0.1:27021  
MongoDB shell version v4.0.6  
connecting to: mongod://127.0.0.1:27021/test?gssapiServiceName=mongodb  
Implicit session: session { "id" : UUID("06319946-6a48-4edd-b1aa-7dba8078b458") }  
MongoDB server version: 4.0.6  
Server has startup warnings:  
2019-03-18T05:53:12.339+0000 I CONTROL [main] ** WARNING: Access control is not enabled for the database.  
2019-03-18T05:53:12.340+0000 I CONTROL [main] ** Read and write access to data and configuration is unrestricted.  
2019-03-18T05:53:12.341+0000 I CONTROL [main] ** WARNING: This server is bound to localhost.  
2019-03-18T05:53:12.342+0000 I CONTROL [main] ** Remote systems will be unable to connect to this server.  
2019-03-18T05:53:12.343+0000 I CONTROL [main] ** Start the server with --bind_ip <address> to specify which IP  
2019-03-18T05:53:12.344+0000 I CONTROL [main] ** addresses it should serve responses from, or with --bind_ip_all  
2019-03-18T05:53:12.344+0000 I CONTROL [main] ** bind to all interfaces. If this behavior is desired, start the  
2019-03-18T05:53:12.344+0000 I CONTROL [main] ** server with --bind_ip 127.0.0.1 to disable this warning.  
2019-03-18T05:53:12.345+0000 I CONTROL [main]  
mongos> use config  
switched to db config  
mongos> db.settings.save({_id:"chunksize", value:1})  
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : "chunksize" })  
mongos>
```

4. Bring up the two shard servers

(1) Open two new terminal, one for each server

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mkdir -p /db/shard0/data
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongod --shardsvr --port 27023 --dbpath /db/shard0/data
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mkdir -p /db/shard0/data
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongod --shardsvr --port 27023 --dbpath /db/shard0/data
2019-03-18T06:05:34.982+0000 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2019-03-18T06:05:34.996+0000 I CONTROL [initandlisten] MongoDB starting : pid=7802 port=27023 dbpath=/db/shard0/data 64-bit host=vagrant-ubuntu-trusty-64
2019-03-18T06:05:34.996+0000 I CONTROL [initandlisten] db version v4.0.6
2019-03-18T06:05:34.997+0000 I CONTROL [initandlisten] git version: caa42a1f75a56c7643d0b68d3880444375ec42e3
2019-03-18T06:05:35.007+0000 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1f 6 Jan 2014
2019-03-18T06:05:35.008+0000 I CONTROL [initandlisten] allocator: tcmalloc
2019-03-18T06:05:35.008+0000 I CONTROL [initandlisten] modules: none
2019-03-18T06:05:35.008+0000 I CONTROL [initandlisten] build environment:
2019-03-18T06:05:35.008+0000 I CONTROL [initandlisten] distmod: ubuntu1404
2019-03-18T06:05:35.009+0000 I CONTROL [initandlisten] distarch: x86_64
2019-03-18T06:05:35.009+0000 I CONTROL [initandlisten] target_arch: x86_64
2019-03-18T06:05:35.009+0000 I CONTROL [initandlisten] options: { net: { port: 27023 }, sharding: { clusterRole: "shardsvr" }, storage: { dbPath: "/db/shard0/data" } }
2019-03-18T06:05:35.012+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2019-03-18T06:05:35.013+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-03-18T06:05:35.013+0000 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=256M,session_max=20000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2019-03-18T06:05:37.534+0000 I STORAGE [initandlisten] WiredTiger message [1552889137:533998][7802:0x7f666e1da80], txn-recover: Set global recovery timestamp: 0
2019-03-18T06:05:38.069+0000 I RECOVERY [initandlisten] WiredTiger recoveryTimestamp. Ts: Timestamp(0, 0)
2019-03-18T06:05:38.916+0000 I CONTROL [initandlisten]
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mkdir -p /db/shard1/data
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongod --shardsvr --port 27024 --dbpath /db/shard1/data
```

```
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mkdir -p /db/shard1/data
vagrant@vagrant-ubuntu-trusty-64:~$ sudo mongod --shardsvr --port 27024 --dbpath /db/shard1/data
2019-03-18T06:08:25.961+0000 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2019-03-18T06:08:25.975+0000 I CONTROL [initandlisten] MongoDB starting : pid=7838 port=27024 dbpath=/db/shard1/data 64-bit host=vagrant-ubuntu-trusty-64
2019-03-18T06:08:25.976+0000 I CONTROL [initandlisten] db version v4.0.6
2019-03-18T06:08:25.976+0000 I CONTROL [initandlisten] git version: caa42a1f75a56c7643d0b68d3880444375ec42e3
2019-03-18T06:08:25.977+0000 I CONTROL [initandlisten] OpenSSL version: OpenSSL 1.0.1f 6 Jan 2014
2019-03-18T06:08:25.978+0000 I CONTROL [initandlisten] allocator: tcmalloc
2019-03-18T06:08:25.978+0000 I CONTROL [initandlisten] modules: none
2019-03-18T06:08:25.979+0000 I CONTROL [initandlisten] build environment:
2019-03-18T06:08:25.981+0000 I CONTROL [initandlisten] distmod: ubuntu1404
2019-03-18T06:08:25.981+0000 I CONTROL [initandlisten] distarch: x86_64
2019-03-18T06:08:25.981+0000 I CONTROL [initandlisten] target_arch: x86_64
2019-03-18T06:08:25.982+0000 I CONTROL [initandlisten] options: { net: { port: 27024 }, sharding: { clusterRole: "shardsvr" }, storage: { dbPath: "/db/shard1/data" } }
2019-03-18T06:08:25.983+0000 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2019-03-18T06:08:25.984+0000 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2019-03-18T06:08:25.984+0000 I STORAGE [initandlisten] wiredtiger_open config: create,cache_size=256M,session_max=20000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),statistics_log=(wait=0),verbose=(recovery_progress),
2019-03-18T06:08:30.695+0000 I STORAGE [initandlisten] WiredTiger message [1552889310:694991][7838:0x7f666b691a80], txn-recover: Set global recovery timestamp: 0
2019-03-18T06:08:30.695+0000 I STORAGE [initandlisten] WiredTiger message [1552889310:694991][7838:0x7f666b691a80], txn-recover: Set global recovery timestamp: 0
2019-03-18T06:08:31.376+0000 I RECOVERY [initandlisten] WiredTiger recoveryTimestamp. Ts: Timestamp(0, 0)
```

5. Tell the sharding system (mongos) where the shard servers are located. Check the shards:

```

mongos> sh.addShard("127.0.0.1:27023")
{
  "shardAdded" : "shard0000",
  "ok" : 1,
  "operationTime" : Timestamp(1552889904, 2),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1552889904, 2),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  }
}
mongos> sh.addShard("127.0.0.1:27024")
{
  "shardAdded" : "shard0001",
  "ok" : 1,
  "operationTime" : Timestamp(1552889909, 2),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1552889909, 2),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  }
}

```

```

mongos> db.printShardingStatus()
--- Sharding Status ---
  sharding version: {
    "_id" : 1,
    "minCompatibleVersion" : 5,
    "currentVersion" : 6,
    "clusterId" : ObjectId("5c8f2eba79c881cd78e6c074")
  }
  shards:
    [ { "_id" : "shard0000", "host" : "127.0.0.1:27023", "state" : 1 }
      { "_id" : "shard0001", "host" : "127.0.0.1:27024", "state" : 1 }
    ]
  active mongoses:
    "4.0.6" : 1
  autosplit:
    Currently enabled: yes
  balancer:
    Currently enabled: yes
    Currently running: no
    Failed balancer rounds in last 5 attempts: 0
    Migration Results for the last 24 hours:
      No recent migrations
  databases:
    [ { "_id" : "config", "primary" : "config", "partitioned" : true }
    ]

```

```

vagrant@vagrant-ubuntu-trusty-64:~$ ps -af | grep mongo
root      6811   6790   0 05:36 pts/0    00:00:00 sudo mongod --port 27022 --dbpath /db/config/data --configsvr --replSet
config
root      6812   6811   1 05:36 pts/0    00:00:37 mongod --port 27022 --dbpath /db/config/data --configsvr --replSet confi
s
vagrant   6938   6922   0 05:37 pts/2    00:00:00 mongo --port 27022
vagrant   7375   7357   0 05:53 pts/3    00:00:00 mongos --configdb config/127.0.0.1:27022 --port 27021
vagrant   7490   7474   0 05:54 pts/5    00:00:00 mongo 127.0.0.1:27021
root      7801   7571   0 06:05 pts/6    00:00:00 sudo mongod --shardsvr --port 27023 --dbpath /db/shard0/data
root      7802   7801   0 06:05 pts/6    00:00:07 mongod --shardsvr --port 27023 --dbpath /db/shard0/data
root      7837   7784   0 06:08 pts/7    00:00:00 sudo mongod --shardsvr --port 27024 --dbpath /db/shard1/data
root      7838   7837   0 06:08 pts/7    00:00:07 mongod --shardsvr --port 27024 --dbpath /db/shard1/data
vagrant   8039   8020   0 06:23 pts/8    00:00:00 grep --color=auto mongo
vagrant@vagrant-ubuntu-trusty-64:~$

```

6. Explicitly tell both database and collection that you want them to be sharded.

- (a) Enabling a database for sharding is a prerequisite to sharding on of its collection
- (b) The given collection is going to be split into chunks.

```
mongos> sh.enableSharding("testdb")
{
  "ok" : 1,
  "operationTime" : Timestamp(1552890290, 6),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1552890290, 6),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
mongos> sh.shardCollection("testdb.testcollection", {testkey:1})
{
  "collectionsharded" : "testdb.testcollection",
  "collectionUUID" : UUID("58114fa0-ce93-48a0-8399-31b066db12a3"),
  "ok" : 1,
  "operationTime" : Timestamp(1552890331, 4),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1552890331, 4),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}
mongos>
```

7. Populate data in testdb.testCollection

```
mongos> db = (new Mongo("localhost:27021")).getDB("testdb")
testdb
mongos> db.testcollection.count()
0
mongos> for(var i=0; i < 10; i++){
... db.testcollection.insert({"username": "user"+i})
... }
WriteResult({
  "nInserted" : 0,
  "writeError" : {
    "code" : 61,
    "errmsg" : "document { '_id': ObjectId('5c8f3b18a218bdb3a346bdfe'), username: 'user9' } does not contain
shard key for pattern { testkey: 1.0 }"
  }
})
mongos> for(var i=0; i < 10; i++){ db.testcollection.insert({"testkey": "user"+i}) }
WriteResult({ "nInserted" : 1 })
mongos> db.testcollection.count()
10
mongos> for(var i=0; i < 100000; i++){
... db.testcollection.insert({"testkey": "key"+i})
... }
WriteResult({ "nInserted" : 1 })WriteResult({ "nInserted" : 1 })
mongos>
```

8. Check the shard status (after populating data)

```

mongos> db.printShardingStatus()
--- Sharding Status ---
  sharding version: {
    "_id" : 1,
    "minCompatibleVersion" : 5,
    "currentVersion" : 6,
    "clusterId" : ObjectId("5c8f2eba79c881cd78e6c074")
  }
  shards:
    [ { "_id" : "shard0000", "host" : "127.0.0.1:27023", "state" : 1 }
      { "_id" : "shard0001", "host" : "127.0.0.1:27024", "state" : 1 } ]
  active mongoses:
    "4.0.6" : 1
  autosplit:
    Currently enabled: yes
  balancer:
    Currently enabled: yes
    Currently running: no
    Failed balancer rounds in last 5 attempts: 0
    Migration Results for the last 24 hours:
      3 : Success
  databases:
    [ { "_id" : "config", "primary" : "config", "partitioned" : true }
      { "_id" : "testdb", "primary" : "shard0001", "partitioned" : true, "version" : { "uuid" : UUID("353f72d6-5e7e-4cb0-b131-852135881486"), "lastMod" : 1 } } ]
    config.system.sessions
      shard key: { "_id" : 1 }
      unique: false
      balancing: true
      chunks:
        shard0000 1
        { "_id" : { "$minKey" : 1 } } --> { "_id" : { "$maxKey" : 1 } } on : shard0000 Timestamp(1, 0)
    testdb.testcollection
      shard key: { "testkey" : 1 }
      unique: false
      balancing: true
      chunks:
        shard0000 3
        shard0001 4
        { "testkey" : { "$minKey" : 1 } } --> { "testkey" : "key1" } on : shard0000 Timestamp(2, 0)
        { "testkey" : "key1" } --> { "testkey" : "key20871" } on : shard0000 Timestamp(3, 0)
        { "testkey" : "key20871" } --> { "testkey" : "key4270" } on : shard0000 Timestamp(4, 0)
        { "testkey" : "key4270" } --> { "testkey" : "key53673" } on : shard0001 Timestamp(3, 2)
        { "testkey" : "key53673" } --> { "testkey" : "key64647" } on : shard0001 Timestamp(3, 3)
        { "testkey" : "key64647" } --> { "testkey" : "key8560" } on : shard0001 Timestamp(3, 4)
        { "testkey" : "key8560" } --> { "testkey" : { "$maxKey" : 1 } } on : shard0001 Timestamp(4, 1)

```

9. Connect to the shards and see how many documents are stored in each shard.

mongo 127.0.0.1:27021

```

mongos> use testdb
switched to db testdb
mongos> db.testcollection.count()
136346
mongos>

```

mongo 127.0.0.1:27023

```

> use testdb
switched to db testdb
> db.testcollection.count()
36336
>

```

mongo 127.0.0.1:27024

```

> use testdb
switched to db testdb
> db.testcollection.count()
100010
>

```

13634 = 36336 + 100010

9. Adding a new shard to the cluster

Make a directory and launch mongod for the shard 2

Sudo mkdir -p /db/shard2/data


```

vagrant@vagrant-ubuntu-trusty-64:~$ mongo 127.0.0.1:27021
MongoDB shell version v4.0.6
connecting to: mongodb://127.0.0.1:27021/test?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("82fbf995-a8f2-4722-b961-7d774f1cf682") }
MongoDB server version: 4.0.6
Server has startup warnings:
2019-03-18T05:53:12.339+0000 I CONTROL [main]
2019-03-18T05:53:12.340+0000 I CONTROL [main] ** WARNING:
2019-03-18T05:53:12.340+0000 I CONTROL [main] **
ed.
2019-03-18T05:53:12.341+0000 I CONTROL [main]
2019-03-18T05:53:12.341+0000 I CONTROL [main] ** WARNING:
2019-03-18T05:53:12.342+0000 I CONTROL [main] **
2019-03-18T05:53:12.343+0000 I CONTROL [main] **

2019-03-18T05:53:12.344+0000 I CONTROL [main] **
ll to
2019-03-18T05:53:12.344+0000 I CONTROL [main] **
e
2019-03-18T05:53:12.344+0000 I CONTROL [main] **
2019-03-18T05:53:12.345+0000 I CONTROL [main]
mongos> use testdb
switched to db testdb
mongos> db.testcollection.count()
112202

vagrant@vagrant-ubuntu-trusty-64:~$ mongo 127.0.0.1:27024
MongoDB shell version v4.0.6
connecting to: mongodb://127.0.0.1:27024/test?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("8a5cf638-39b5-48d2-b961-7d774f1cf682") }
MongoDB server version: 4.0.6
Server has startup warnings:
2019-03-18T06:08:25.983+0000 I STORAGE [initandlisten]
2019-03-18T06:08:25.984+0000 I STORAGE [initandlisten] **
2019-03-18T06:08:25.984+0000 I STORAGE [initandlisten] **
2019-03-18T06:08:32.279+0000 I CONTROL [initandlisten]
2019-03-18T06:08:32.279+0000 I CONTROL [initandlisten] **
2019-03-18T06:08:32.280+0000 I CONTROL [initandlisten] **
2019-03-18T06:08:32.281+0000 I CONTROL [initandlisten] **
2019-03-18T06:08:32.281+0000 I CONTROL [initandlisten]
2019-03-18T06:08:32.282+0000 I CONTROL [initandlisten] **
2019-03-18T06:08:32.282+0000 I CONTROL [initandlisten] **
2019-03-18T06:08:32.283+0000 I CONTROL [initandlisten] **
2019-03-18T06:08:32.284+0000 I CONTROL [initandlisten] **
2019-03-18T06:08:32.284+0000 I CONTROL [initandlisten] **
2019-03-18T06:08:32.284+0000 I CONTROL [initandlisten] **
2019-03-18T06:08:32.285+0000 I CONTROL [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which
metrics about your deployment (disk utilization, CPU, oper

The monitoring data will be available on a MongoDB website
and anyone you share the URL with. MongoDB may use this in
improvements and to suggest MongoDB products and deployment

To enable free monitoring, run the following command: db.e
To permanently disable this reminder, run the following co
---
> use testdb
switched to db testdb
> db.testcollection.count()
63674

```

```
vagrant@vagrant-ubuntu-trusty-64: $ mongo 127.0.0.1:27025
MongoDB shell version v4.0.6
connecting to: mongodb://127.0.0.1:27025/test?gssapiServiceName=mongodb
Implicit session: session { "id": "UUID("13c7fc51-b0e1-47f8-a9d9-d72e87d4135e")"}
MongoDB server version: 4.0.6
Server has startup warnings:
2019-03-18T06:51:47.978+0000 I STORAGE [initandlisten] 
2019-03-18T06:51:47.978+0000 I STORAGE [initandlisten] **
2019-03-18T06:51:47.978+0000 I STORAGE [initandlisten] **
2019-03-18T06:51:53.157+0000 I CONTROL [initandlisten] 
2019-03-18T06:51:53.157+0000 I CONTROL [initandlisten] **
2019-03-18T06:51:53.157+0000 I CONTROL [initandlisten] **
2019-03-18T06:51:53.157+0000 I CONTROL [initandlisten] **
2019-03-18T06:51:53.157+0000 I CONTROL [initandlisten] **
2019-03-18T06:51:53.157+0000 I CONTROL [initandlisten] **
2019-03-18T06:51:53.157+0000 I CONTROL [initandlisten] **
2019-03-18T06:51:53.157+0000 I CONTROL [initandlisten] **
2019-03-18T06:51:53.157+0000 I CONTROL [initandlisten] **
2019-03-18T06:51:53.157+0000 I CONTROL [initandlisten] **
2019-03-18T06:51:53.157+0000 I CONTROL [initandlisten] **
-----
Enable MongoDB's free cloud-based monitoring service, which collects
metrics about your deployment (disk utilization, CPU, operati
The monitoring data will be available on a MongoDB website
and anyone you share the URL with. MongoDB may use this info
improvements and to suggest MongoDB products and deployments
To enable free monitoring, run the following command: db.setEnvi
To permanently disable this reminder, run the following command:
----

> use testdb
switched to db testdb
> db.testcollection.count()
12192
```

```

vagrant@vagrant-ubuntu-trusty-64:~$ mongo 127.0.0.1:27023
MongoDB shell version v4.0.6
connecting to: mongod://127.0.0.1:27023/test?gssapiService
Implicit session: session { "id" : UUID("7b4d3f95-689e-41e
MongoDB server version: 4.0.6
Server has startup warnings:
2019-03-18T06:05:35.012+0000 I STORAGE [initandlisten]
2019-03-18T06:05:35.013+0000 I STORAGE [initandlisten] **
2019-03-18T06:05:35.013+0000 I STORAGE [initandlisten] **
2019-03-18T06:05:38.916+0000 I CONTROL [initandlisten]
2019-03-18T06:05:38.916+0000 I CONTROL [initandlisten] **
2019-03-18T06:05:38.917+0000 I CONTROL [initandlisten] **
2019-03-18T06:05:38.917+0000 I CONTROL [initandlisten] **
2019-03-18T06:05:38.918+0000 I CONTROL [initandlisten]
2019-03-18T06:05:38.919+0000 I CONTROL [initandlisten] **
2019-03-18T06:05:38.919+0000 I CONTROL [initandlisten] **
2019-03-18T06:05:38.920+0000 I CONTROL [initandlisten] **
2019-03-18T06:05:38.921+0000 I CONTROL [initandlisten] **
2019-03-18T06:05:38.921+0000 I CONTROL [initandlisten] **
2019-03-18T06:05:38.921+0000 I CONTROL [initandlisten] **
---
Enable MongoDB's free cloud-based monitoring service, which
metrics about your deployment (disk utilization, CPU, oper
The monitoring data will be available on a MongoDB website
and anyone you share the URL with. MongoDB may use this in
improvements and to suggest MongoDB products and deployment
To enable free monitoring, run the following command: db.e
To permanently disable this reminder, run the following co
---
> use testdb
switched to db testdb
> db.testcollection.count()
36336

```

$$112202 = 63674 + 12192 + 36336$$

10. Removing a shard from the cluster

```
vagrant@vagrant-ubuntu-trusty-64:~$ mongo 127.0.0.1:27021
MongoDB shell version v4.0.6
connecting to: mongodb://127.0.0.1:27021/test?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("3024239c-cbd3-432e-99ad-10e1251635f5") }
MongoDB server version: 4.0.6
Server has startup warnings:
2019-03-18T05:53:12.339+0000 I CONTROL [main]
2019-03-18T05:53:12.340+0000 I CONTROL [main] ** WARNING: Access control is
2019-03-18T05:53:12.340+0000 I CONTROL [main] ** Read and write acc
2019-03-18T05:53:12.341+0000 I CONTROL [main]
2019-03-18T05:53:12.341+0000 I CONTROL [main] ** WARNING: This server is bou
2019-03-18T05:53:12.342+0000 I CONTROL [main] ** Remote systems wil
2019-03-18T05:53:12.343+0000 I CONTROL [main] ** Start the server w
2019-03-18T05:53:12.344+0000 I CONTROL [main] ** addresses it shoul
2019-03-18T05:53:12.344+0000 I CONTROL [main] ** bind to all interf
2019-03-18T05:53:12.344+0000 I CONTROL [main] ** server with --bind
2019-03-18T05:53:12.345+0000 I CONTROL [main]
mongos> use admin
switched to db admin
mongos> db.runCommand({removeShard:"127.0.0.1:27025"})
{
  "msg" : "draining started successfully",
  "state" : "started",
  "shard" : "shard0002",
  "note" : "you need to drop or movePrimary these databases",
  "dbsToMove" : [ ],
  "ok" : 1,
  "operationTime" : Timestamp(1552893356, 2),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1552893356, 2),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  }
}
mongos> db.runCommand({removeShard:"127.0.0.1:27025"})
{
  "msg" : "removeshard completed successfully",
  "state" : "completed",
  "shard" : "shard0002",
  "ok" : 1,
  "operationTime" : Timestamp(1552893393, 2),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1552893393, 2),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  }
}
```

```

mongos> db.runCommand({listshards:1})
{
  "shards" : [
    {
      "_id" : "shard0000",
      "host" : "127.0.0.1:27023",
      "state" : 1
    },
    {
      "_id" : "shard0001",
      "host" : "127.0.0.1:27024",
      "state" : 1
    }
  ],
  "ok" : 1,
  "operationTime" : Timestamp(1552893500, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1552893500, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}

mongos> db.runCommand({isdbgrid:1})
{
  "isdbgrid" : 1,
  "hostname" : "vagrant-ubuntu-trusty-64",
  "ok" : 1,
  "operationTime" : Timestamp(1552893611, 3),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1552893611, 3),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  }
}

```