MongoDB学习指南

一、MongoDB产品简介

分布式
MongoDB提供了一个面向文档存储方式
文档型数据库
---体验操作即可
RDS----阿里云数据库---非关系型数据库

replication副本
sharding分片

应用案例：

游戏场景

物流场景

社交场景

物联网场景

视频直播

二、MongoDB基本概念

文档 document

集合 collection

MongoDB 和关系型数据库的对比

Database	database
Table	collection
Row	

数据库文件类型

命名空间文件

日志文件类型

系统日志

慢查询

oplog类似于binlog

MongoDB 数据类型

MongoDB元素命名规则

集合命名规范

不能含有空字符串

不能含有system.开头

不能含有出现$

MongoDB安装部署

```
[root@localhost ~]# iptables -F
[root@localhost ~]# setenforce 0
setenforce: SELinux is disabled
[root@localhost ~]# systemctl stop firewalld
```

临时调整

永久调整：/etc/profile　 /etc/rc.local

```
[root@localhost ~]# ulimit -n 65535
[root@localhost ~]# ulimit -u 65535
[root@localhost ~]# rz -E
rz waiting to receive.
```

```
[root@localhost ~]# tar xf mongodb-linux-x86_64-rhel70-4.0.6.tgz
[root@localhost ~]# mv mongodb-linux-x86_64-rhel70-4.0.6 /usr/local/mongodb
[root@localhost ~]# ls /usr/local/mongodb/
bin  LICENSE-Community.txt  MPL-2  README  THIRD-PARTY-NOTICES
[root@localhost ~]# ls /usr/local/mongodb/bin
bsondump        mongo  mongodump   mongofiles  mongoreplay  mongos    mongotop
install_compass  mongod  mongoexport  mongoimport  mongorestore  mongostat
[root@localhost ~]# ln -s /usr/local/mongodb/bin/* /bin/
[root@localhost ~]# mkdir -p /data/mongodb1
[root@localhost ~]# mkdir -p /data/logs/mongodb
[root@localhost ~]# touch /data/logs/mongodb/mongodb1.log
[root@localhost ~]# cd /usr/local/mongodb
[root@localhost mongodb]# ls
bin  LICENSE-Community.txt  MPL-2  README  THIRD-PARTY-NOTICES
[root@localhost mongodb]# mkdir conf
[root@localhost mongodb]# vim conf/mongodb1.conf
port=27017                              监听端口
dbpath=/data/mongodb1                   指定数据目录
logpath=/data/logs/mongodb/mongodb1.log             指定日志文件路径
logappend=true                  允许写入日志
fork=true                                       允许创建子进程
maxConns=5000                   最大连接数
storageEngine=mmapv1                    存储引擎
[root@localhost ~]# mongod -f /usr/local/mongodb/conf/mongodb1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 1783
child process started successfully, parent exiting
[root@localhost ~]# netstat -lnpt | grep mongod
tcp       0     0 127.0.0.1:27017        0.0.0.0:*           LISTEN      1783/mongod
[root@localhost ~]# ps aux | grep mongod
root              1783      1.8      4.7   1501232   95536   ?              Sl        04:13        0:02   mongod   -f
/usr/local/mongodb/conf/mongodb1.conf
root       1826  0.0  0.0 112720   984 pts/0    S+   04:15   0:00 grep --color=auto mongod
[root@localhost ~]# vim /etc/rc.local
rm -f /data/mongodb1/mongod.lock
/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/mongodb1.conf
[root@localhost ~]# ls -l /etc/rc.local
lrwxrwxrwx. 1 root root 13 1月   1 03:34 /etc/rc.local -> rc.d/rc.local
```

```
[root@localhost ~]# ls -l /etc/rc.d/rc.local
-rw-r--r--. 1 root root 619 5月  23 04:18 /etc/rc.d/rc.local
[root@localhost ~]# chmod u+x /etc/rc.d/rc.local
[root@localhost ~]# mongo
> show databases;
admin   0.078GB
config  0.078GB
local   0.078GB
> show dbs;
admin   0.078GB
config  0.078GB
local   0.078GB
> exit
bye
```

1. MongoDB配置管理

方法一：

```
[root@localhost ~]# mongo
> use admin
switched to db admin
> db.shutdownServer();
server should be down...
2020-05-23T04:26:50.626+0800 I NETWORK  [js] trying reconnect to 127.0.0.1:27017 failed
2020-05-23T04:26:50.627+0800 I NETWORK  [js] reconnect 127.0.0.1:27017 failed failed
> exit
bye
2020-05-23T04:26:55.227+0800 I NETWORK  [js] trying reconnect to 127.0.0.1:27017 failed
2020-05-23T04:26:55.227+0800 I NETWORK  [js] reconnect 127.0.0.1:27017 failed failed
2020-05-23T04:26:55.227+0800 I QUERY    [js] Failed to end session { id: UUID("1a70969e-d96c-421b-b712-92a56b84aa17") } du
e to SocketException: socket exception [CONNECT_ERROR] server [couldn't connect to server 127.0.0.1:27017, connection attempt failed: SocketException: Error connecting to 127.0.0.1:27017 :: caused by :: Connection refused]
[root@localhost ~]# netstat -lnpt |grep mongod
```

方法二：

```
[root@localhost ~]# mongod -f /usr/local/mongodb/conf/mongodb1.conf --shutdown
killing process with pid: 2005
[root@localhost ~]# netstat -lnpt |grep mongod
```

方法三：

```
[root@localhost ~]# mongod -f /usr/local/mongodb/conf/mongodb1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 2042
child process started successfully, parent exiting
[root@localhost ~]# ps aux | grep mongod
root        2042  0.4  4.4 1500188 89868 ?        Sl   04:30   0:00 mongod -f /usr/local/mongodb/conf/mongodb1.conf
root      2108  0.0  0.0 112720   980 pts/0    S+   04:31   0:00 grep --color=auto mongod
[root@localhost ~]# kill -9 2042
[root@localhost ~]# netstat -lnpt |grep mongod
```

MongoDB多实例配置

```
[root@localhost ~]# cd /usr/local/mongodb/conf/
[root@localhost conf]# ls
mongodb1.conf
[root@localhost conf]# cp mongodb{1,2}.conf
[root@localhost conf]# ls
mongodb1.conf  mongodb2.conf
[root@localhost conf]# vim mongodb2.conf
port=27018
dbpath=/data/mongodb2
logpath=/data/logs/mongodb/mongodb2.log
logappend=true
fork=true
maxConns=5000
storageEngine=mmapv1
[root@localhost conf]# mkdir /data/mongodb2
[root@localhost conf]# touch /data/logs/mongodb/mongodb2.log
[root@localhost conf]# mongod -f /usr/local/mongodb/conf/mongodb2.conf
about to fork child process, waiting until server is ready for connections.
forked process: 2192
child process started successfully, parent exiting
[root@localhost ~]# !net
```

```
netstat -lnpt |grep mongod
tcp        0      0 127.0.0.1:27018         0.0.0.0:*               LISTEN
2192/mongod
```

编写启停脚本

```
[root@localhost ~]# vim /etc/init.d/mongodb
#!/bin/bash
INSTANCE=$1
ACTION=$2
case "$ACTION" in
'start')
/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/"$INSTANCE".conf;;
'stop')
/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/"$INSTANCE".conf --shutdown;;
'restart')
/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/"$INSTANCE".conf --shutdown
/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/"$INSTANCE".conf;;
esac
```

```
[root@localhost ~]# /etc/init.d/mongodb mongodb1 restart
There doesn't seem to be a server running with dbpath: /data/mongodb1
about to fork child process, waiting until server is ready for connections.
forked process: 2427
child process started successfully, parent exiting
[root@localhost ~]# /etc/init.d/mongodb mongodb2 stop
killing process with pid: 2192
```

基本操作

```
[root@localhost ~]# mongo
MongoDB shell version v4.0.6
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
[root@localhost ~]# mongo --port 27018
MongoDB shell version v4.0.6
connecting to: mongodb://127.0.0.1:27018/?gssapiServiceName=mongodb
> show dbs
admin   0.078GB
config  0.078GB
```

```
local   0.078GB
> show databases;
admin   0.078GB
config  0.078GB
local   0.078GB
> db
test
> use admin
switched to db admin
> db
admin
> use sofia
switched to db sofia
> show dbs
admin   0.078GB
config  0.078GB
local   0.078GB

> db.user.insert({"id":1,"name":"sofia"});
WriteResult({ "nInserted" : 1 })
> show dbs
admin   0.078GB
cloud   0.078GB
config  0.078GB
local   0.078GB
> use cloud
switched to db cloud
> db.user.insert({"id":1,"name":"sofia"});
WriteResult({ "nInserted" : 1 })
> show dbs
admin   0.078GB
cloud   0.078GB
config  0.078GB
local   0.078GB
> show tables
user
> show collections
```

```
user
> db.user.find()
{ "_id" : ObjectId("5ec8467b43d53389391e8077"), "id" : 1, "name" : "sofia" }
[root@localhost ~]#
[root@localhost ~]# ll /data/mongodb1/
总用量 327688
-rw------- 1 root root 67108864 5月  23 04:13 admin.0
-rw------- 1 root root 16777216 5月  23 04:13 admin.ns
-rw------- 1 root root 67108864 5月  23 05:39 cloud.0
-rw------- 1 root root 16777216 5月  23 05:39 cloud.ns
-rw------- 1 root root 67108864 5月  23 07:34 config.0
-rw------- 1 root root 16777216 5月  23 07:34 config.ns
drwx------ 2 root root      239 5月  23 07:35 diagnostic.data
drwx------ 2 root root       29 5月  23 05:24 journal
-rw------- 1 root root 67108864 5月  23 04:59 local.0
-rw------- 1 root root 16777216 5月  23 04:59 local.ns
-rw------- 1 root root        5 5月  23 04:59 mongod.lock
-rw------- 1 root root       69 5月  23 04:13 storage.bson
drwx------ 2 root root        6 5月  23 05:39 _tmp
> help
	db.help()                    help on db methods
	db.mycoll.help()             help on collection methods
	sh.help()                    sharding helpers
	rs.help()                    replica set helpers
	help admin                   administrative help
	help connect                 connecting to a db help
	help keys                    key shortcuts
	help misc                    misc things to know
	help mr                      mapreduce

	show dbs                     show database names
	show collections             show collections in current database
	show users                   show users in current database
	show profile                 show most recent system.profile entries with time >= 1ms
	show logs                    show the accessible logger names
	show log [name]              prints out the last segment of log in memory, 'global' is default
	use <db_name>                set current database
```

```
        db.foo.find()                list objects in collection foo
        db.foo.find( { a : 1 } )     list objects in foo where a == 1
        it                           result of the last line evaluated; use to further iterate
        DBQuery.shellBatchSize = x   set default number of items to display on shell
        exit                         quit the mongo shell
> db.version()
4.0.6
> db.stats();
{
        "db" : "test",
        "collections" : 0,
        "views" : 0,
        "objects" : 0,
        "avgObjSize" : 0,
        "dataSize" : 0,
        "storageSize" : 0,
        "numExtents" : 0,
        "indexes" : 0,
        "indexSize" : 0,
        "fileSize" : 0,
        "fsUsedSize" : 0,
        "fsTotalSize" : 0,
        "ok" : 1
}
>
> use cloud
switched to db cloud
> use cloud
switched to db cloud
> show tables
user
> db.user.help()
> show databases;
admin   0.078GB
cloud   0.078GB
config  0.078GB
local   0.078GB
```

数据类型：

```
> use study
switched to db study
> db.t1.insert({"id":1});
WriteResult({ "nInserted" : 1 })
> show tables;
t1
> show dbs
admin   0.078GB
cloud   0.078GB
config  0.078GB
local   0.078GB
study   0.078GB
> db.t1.insert({"id":2,"name":"Tom","isadmin":true,"gender":null,"favorite":
["apple","banana","orange",1,2,3],"regtime":new Date()});
> db.t1.find();
{ "_id" : ObjectId("5ec8661c0a221f9a9efc3206"), "id" : 1 }
{ "_id" : ObjectId("5ec867c60a221f9a9efc3207"), "id" : 2, "name" : "Tom", "isadmin" : true, "gender" :
null, "favorite" :
[ "apple", "banana", "orange", 1, 2, 3 ], "regtime" : ISODate("2020-05-23T00:01:10.610Z") }
> db.t1.findOne({'id':2});
{
	"_id" : ObjectId("5ec867c60a221f9a9efc3207"),
	"id" : 2,
	"name" : "Tom",
	"isadmin" : true,
	"gender" : null,
	"favorite" : [
		"apple",
		"banana",
		"orange",
		1,
		2,
		3
	],
	"regtime" : ISODate("2020-05-23T00:01:10.610Z")
}
> a=db.t1.findOne({'id':2});
{
        "_id" : ObjectId("5ec867c60a221f9a9efc3207"),
        "id" : 2,
        "name" : "Tom",
        "isadmin" : true,
        "gender" : null,
        "favorite" : [
                "apple",
                "banana",
                "orange",
                1,
                2,
                3
        ],
        "regtime" : ISODate("2020-05-23T00:01:10.610Z")
}
> typeof(a.id)
number
> typeof(a.name)
string
> typeof(a.isadmin)
boolean
> typeof(a.favorite)
object
> typeof(a.regtime)
object
>
db.t1.insert({"id":3,"salary":666666666666666666666666666,"rx":1.888888888888888888888888888});
WriteResult({ "nInserted" : 1 })
> a=db.t1.findOne({'id':3});
{
        "_id" : ObjectId("5ec92cfd1f8acb370e97cf77"),
        "id" : 3,
        "salary" : 6.666666666666666e+26,
        "rx" : 1.8888888888888888
}
```

```
> b=db.t1.findOne({'id':3});
{
        "_id" : ObjectId("5ec92cfd1f8acb370e97cf77"),
        "id" : 3,
        "salary" : 6.666666666666666e+26,
        "rx" : 1.8888888888888888
}
> typeof(b.salary)
number
> typeof(b.rx)
number
```

在MongoDB中，没有整数与浮点数的区别，都是number类型

关于null空

```
> db.t1.find({"gender":null});
{ "_id" : ObjectId("5ec8661c0a221f9a9efc3206"), "id" : 1 }
{ "_id" : ObjectId("5ec867c60a221f9a9efc3207"), "id" : 2, "name" : "Tom", "isadmin" : true, "gender" :
null, "favorite" :
[ "apple", "banana", "orange", 1, 2, 3 ], "regtime" : ISODate("2020-05-23T00:01:10.610Z") }{ "_id" :
ObjectId("5ec92e801f8acb370e97cf78"), "id" : 3, "salary" : 6.666666666666666e+26, "rx" :
1.8888888888888888 }
> db.t1.find({"gender":{"$exists":true}});
{ "_id" : ObjectId("5ec867c60a221f9a9efc3207"), "id" : 2, "name" : "Tom", "isadmin" : true, "gender" :
null, "favorite" :
[ "apple", "banana", "orange", 1, 2, 3 ], "regtime" : ISODate("2020-05-23T00:01:10.610Z") }
> db.t2.insert({"id":1,"id":2});
WriteResult({ "nInserted" : 1 })
> db.t2.find();
{ "_id" : ObjectId("5ec92fcf1f8acb370e97cf79"), "id" : 2 }
> db.t2.insert({"id":1,"id":2});
WriteResult({ "nInserted" : 1 })
> db.t2.find();
{ "_id" : ObjectId("5ec92fcf1f8acb370e97cf79"), "id" : 2 }
> db.t2.insert({"id":1,"Id":2});
WriteResult({ "nInserted" : 1 })
> db.t2.find();
{ "_id" : ObjectId("5ec92fcf1f8acb370e97cf79"), "id" : 2 }
{ "_id" : ObjectId("5ec930011f8acb370e97cf7a"), "id" : 1, "Id" : 2 }
> db.t2.insert({"id":2,"Id":1});
WriteResult({ "nInserted" : 1 })
> db.t2.find();
{ "_id" : ObjectId("5ec92fcf1f8acb370e97cf79"), "id" : 2 }
{ "_id" : ObjectId("5ec930011f8acb370e97cf7a"), "id" : 1, "Id" : 2 }
{ "_id" : ObjectId("5ec930161f8acb370e97cf7b"), "id" : 2, "Id" : 1 }
> db.t2.insert({"":3});
WriteResult({ "nInserted" : 1 })
> db.t2.find();
{ "_id" : ObjectId("5ec92fcf1f8acb370e97cf79"), "id" : 2 }
{ "_id" : ObjectId("5ec930011f8acb370e97cf7a"), "id" : 1, "Id" : 2 }
{ "_id" : ObjectId("5ec930161f8acb370e97cf7b"), "id" : 2, "Id" : 1 }
{ "_id" : ObjectId("5ec930441f8acb370e97cf7c"), "" : 3 }
> db.t2.insert({"$id":23});                    #  不能以$开头
WriteResult({
        "nInserted" : 0,
        "writeError" : {
                "code" : 2,
                "errmsg" : "Document can't have $ prefixed field names: $id"
        }
})
> db.t2.insert({"i$d":23});            #  可放与中间
WriteResult({ "nInserted" : 1 })
> db.t2.find();
{ "_id" : ObjectId("5ec92fcf1f8acb370e97cf79"), "id" : 2 }
{ "_id" : ObjectId("5ec930011f8acb370e97cf7a"), "id" : 1, "Id" : 2 }
{ "_id" : ObjectId("5ec930161f8acb370e97cf7b"), "id" : 2, "Id" : 1 }
{ "_id" : ObjectId("5ec930441f8acb370e97cf7c"), "" : 3 }
{ "_id" : ObjectId("5ec930851f8acb370e97cf7e"), "i$d" : 23 }
> db.t2.insert({"\0":4});         #  不能以\0命名
2020-05-23T22:18:58.851+0800 E QUERY    [js] Error: JavaScript property (name) contains a null char which is not allowed i
n BSON.  :Bulk/addToOperationsList@src/mongo/shell/bulk_api.js:611:28
Bulk/this.insert@src/mongo/shell/bulk_api.js:654:20
DBCollection.prototype.insert@src/mongo/shell/collection.js:318:13
@(shell):1:1
> db..insert({"id":1});        #  表名不能为空
```

```
2020-05-23T22:20:04.680+0800 E QUERY    [js] SyntaxError: missing name after . operator @(shell):1:3
```

查询操作

```
[root@localhost ~]# mongoimport -d cloud -c list -f id,name,age,dept --file list.csv --type csv
2020-05-23T22:27:38.024+0800    connected to: localhost
2020-05-23T22:27:38.028+0800    imported 102 documents
[root@localhost ~]# mongo
> use cloud
switched to db cloud
> show tables
list
user
> db.list.find();
> db.list.count()
102
> db.list.insert({"id":103,"name":"诸葛亮","age":1000,"dept":"行政部"})
WriteResult({ "nInserted" : 1 })
> db.list.find({"name":"诸葛亮"})
{ "_id" : ObjectId("5ec933dfe8035add00fa10ca"), "id" : 103, "name" : "诸葛亮", "age" : 1000, "dept" : "行政部" }
> db.list.update({"name":"诸葛亮"},{"$set":{"id":150}});                  #  更新
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.list.find({"name":"诸葛亮"})
{ "_id" : ObjectId("5ec933dfe8035add00fa10ca"), "id" : 150, "name" : "诸葛亮", "age" : 1000, "dept" : "行政部" }
> db.list.remove({"name":"诸葛亮"})              #  删除
WriteResult({ "nRemoved" : 1 })
> db.list.find({"name":"诸葛亮"})
```

//排序，按年龄从小到大（-1为从大到小），前十个

```
> db.list.find({"name":"诸葛亮"})
> db.list.find().sort({age:1}).limit(10)
{ "_id" : ObjectId("5ec932da930cacaa1c7175c6"), "id" : 93, "name" : "贾宝珠", "age" : 15, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175c2"), "id" : 88, "name" : "崔兴龙", "age" : 17, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175c7"), "id" : 94, "name" : "阎君", "age" : 17, "dept" : "客服部"
}
{ "_id" : ObjectId("5ec932da930cacaa1c7175ba"), "id" : 81, "name" : "王晨旭", "age" : 18, "dept" : "客服
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175cf"), "id" : 102, "name" : "靳璠", "age" : 18, "dept" : "客服部"
}
{ "_id" : ObjectId("5ec932da930cacaa1c717594"), "id" : 42, "name" : "杨硕", "age" : 20, "dept" : "工程部"
}
{ "_id" : ObjectId("5ec932da930cacaa1c7175a0"), "id" : 55, "name" : "于忠鑫", "age" : 20, "dept" : "客服
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71759b"), "id" : 50, "name" : "高慧颖", "age" : 21, "dept" : "教师
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175d0"), "id" : 100, "name" : "杨强", "age" : 21, "dept" : "客服
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717574"), "id" : 11, "name" : "蔡明珠", "age" : 22, "dept" : "会计
部" }
> db.list.find().sort({age:1}).limit(12)
{ "_id" : ObjectId("5ec932da930cacaa1c7175c6"), "id" : 93, "name" : "贾宝珠", "age" : 15, "dept" : "客服
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175c2"), "id" : 88, "name" : "崔兴龙", "age" : 17, "dept" : "客服
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175c7"), "id" : 94, "name" : "阎君", "age" : 17, "dept" : "客服部"
}
{ "_id" : ObjectId("5ec932da930cacaa1c7175ba"), "id" : 81, "name" : "王晨旭", "age" : 18, "dept" : "客服
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175cf"), "id" : 102, "name" : "靳璠", "age" : 18, "dept" : "客服部"
}
{ "_id" : ObjectId("5ec932da930cacaa1c717594"), "id" : 42, "name" : "杨硕", "age" : 20, "dept" : "工程部"
}
{ "_id" : ObjectId("5ec932da930cacaa1c7175a0"), "id" : 55, "name" : "于忠鑫", "age" : 20, "dept" : "客服
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71759b"), "id" : 50, "name" : "高慧颖", "age" : 21, "dept" : "教师
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175d0"), "id" : 100, "name" : "杨强", "age" : 21, "dept" : "客服
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717574"), "id" : 11, "name" : "蔡明珠", "age" : 22, "dept" : "会计
部" }
```

```
{ "_id" : ObjectId("5ec932da930cacaa1c717584"), "id" : 27, "name" : "王希远", "age" : 22, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71759e"), "id" : 53, "name" : "张萌萌", "age" : 22, "dept" : "教师部" }
> db.list.find().sort({age:-1}).limit(12)
{ "_id" : ObjectId("5ec932da930cacaa1c7175c3"), "id" : 90, "name" : "杨志超", "age" : 67, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71756c"), "id" : 3, "name" : "刘嘉", "age" : 63, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175bd"), "id" : 83, "name" : "张静", "age" : 63, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175c1"), "id" : 87, "name" : "孙洁", "age" : 63, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175ad"), "id" : 68, "name" : "李佳", "age" : 60, "dept" : "工程部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175cb"), "id" : 98, "name" : "槐莎莎", "age" : 60, "dept" : "工程部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71757f"), "id" : 21, "name" : "吴秋月", "age" : 58, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717587"), "id" : 30, "name" : "宋涛", "age" : 58, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71757d"), "id" : 1, "name" : "孙美伶", "age" : 57, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717586"), "id" : 29, "name" : "齐琪", "age" : 56, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717572"), "id" : 9, "name" : "曹铮铮", "age" : 52, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71757b"), "id" : 18, "name" : "常莎", "age" : 52, "dept" : "会计部" }
>
> db.list.find().sort({age:-1}).skip(10).limit(12)              #  跳过头十个，从第11个开始
{ "_id" : ObjectId("5ec932da930cacaa1c717572"), "id" : 9, "name" : "曹铮铮", "age" : 52, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71757b"), "id" : 18, "name" : "常莎", "age" : 52, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71758b"), "id" : 34, "name" : "王芳", "age" : 52, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175aa"), "id" : 65, "name" : "王君", "age" : 51, "dept" : "工程部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71758a"), "id" : 33, "name" : "乔磊", "age" : 50, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71758e"), "id" : 37, "name" : "吴桐", "age" : 50, "dept" : "工程部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71759c"), "id" : 51, "name" : "肖玄", "age" : 50, "dept" : "工程部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175b4"), "id" : 62, "name" : "何志宽", "age" : 50, "dept" : "工程部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175be"), "id" : 84, "name" : "王继蔚", "age" : 49, "dept" : "工程部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71756b"), "id" : 2, "name" : "付建梅", "age" : 48, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71758c"), "id" : 35, "name" : "江丽", "age" : 48, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717592"), "id" : 41, "name" : "刘冉", "age" : 47, "dept" : "工程部" }
> db.list.find({"age":20})
{ "_id" : ObjectId("5ec932da930cacaa1c717594"), "id" : 42, "name" : "杨硕", "age" : 20, "dept" : "工程部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175a0"), "id" : 55, "name" : "于忠鑫", "age" : 20, "dept" : "客服部" }
> db.list.find({"age":20},{"name":1,"age":1})
{ "_id" : ObjectId("5ec932da930cacaa1c717594"), "name" : "杨硕", "age" : 20 }
{ "_id" : ObjectId("5ec932da930cacaa1c7175a0"), "name" : "于忠鑫", "age" : 20 }
> db.list.find({"age":{$gt:60}})
{ "_id" : ObjectId("5ec932da930cacaa1c71756c"), "id" : 3, "name" : "刘嘉", "age" : 63, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175bd"), "id" : 83, "name" : "张静", "age" : 63, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175c1"), "id" : 87, "name" : "孙洁", "age" : 63, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175c3"), "id" : 90, "name" : "杨志超", "age" : 67, "dept" : "市场部" }
```

年龄除以10余1的，即11 21 31 41....

```
> db.list.find({"age":{$mod:[10,1]}})
{ "_id" : ObjectId("5ec932da930cacaa1c717571"), "id" : 8, "name" : "闫金花", "age" : 41, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717599"), "id" : 48, "name" : "顾洁", "age" : 41, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71759b"), "id" : 50, "name" : "高慧颖", "age" : 21, "dept" : "教师
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175aa"), "id" : 65, "name" : "王君", "age" : 51, "dept" : "工程部"
}
{ "_id" : ObjectId("5ec932da930cacaa1c7175ce"), "id" : 101, "name" : "杨譞", "age" : 31, "dept" : "客服
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175d0"), "id" : 100, "name" : "杨强", "age" : 21, "dept" : "客服
部" }
> db.list.find({"dept":"会计部"})
{ "_id" : ObjectId("5ec932da930cacaa1c71756b"), "id" : 2, "name" : "付建梅", "age" : 48, "dept" : "会计
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71756d"), "id" : 4, "name" : "刘晓慧", "age" : 33, "dept" : "会计
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71756e"), "id" : 5, "name" : "谢娜", "age" : 23, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71756f"), "id" : 6, "name" : "翟迪", "age" : 27, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717572"), "id" : 9, "name" : "曹铮铮", "age" : 52, "dept" : "会计
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717573"), "id" : 10, "name" : "杨雪飞", "age" : 44, "dept" : "会计
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717574"), "id" : 11, "name" : "蔡明珠", "age" : 22, "dept" : "会计
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717575"), "id" : 12, "name" : "刘海霞", "age" : 35, "dept" : "会计
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717578"), "id" : 15, "name" : "晋红柳", "age" : 35, "dept" : "会计
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71757b"), "id" : 18, "name" : "常莎", "age" : 52, "dept" : "会计部"
}
{ "_id" : ObjectId("5ec932da930cacaa1c71757d"), "id" : 1, "name" : "孙美伶", "age" : 57, "dept" : "会计
部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717585"), "id" : 28, "name" : "吴迪", "age" : 38, "dept" : "会计部"
}
{ "_id" : ObjectId("5ec932da930cacaa1c71758d"), "id" : 36, "name" : "高雪", "age" : 46, "dept" : "会计部"
}
{ "_id" : ObjectId("5ec932da930cacaa1c717596"), "id" : 44, "name" : "杨柳", "age" : 23, "dept" : "会计部"
}
{ "_id" : ObjectId("5ec932da930cacaa1c7175b1"), "id" : 72, "name" : "肖瑞", "age" : 38, "dept" : "会计部"
}
```

```
{ "_id" : ObjectId("5ec932da930cacaa1c7175b2"), "id" : 73, "name" : "许博远", "age" : 30, "dept" : "会计部" }
> db.list.find({"dept":{"$in":["会计部","客服部"]}})
{ "_id" : ObjectId("5ec932da930cacaa1c71756b"), "id" : 2, "name" : "付建梅", "age" : 48, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71756d"), "id" : 4, "name" : "刘晓慧", "age" : 33, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71756e"), "id" : 5, "name" : "谢娜", "age" : 23, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71756f"), "id" : 6, "name" : "翟迪", "age" : 27, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717572"), "id" : 9, "name" : "曹铮铮", "age" : 52, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717573"), "id" : 10, "name" : "杨雪飞", "age" : 44, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717574"), "id" : 11, "name" : "蔡明珠", "age" : 22, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717575"), "id" : 12, "name" : "刘海霞", "age" : 35, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717578"), "id" : 15, "name" : "晋红柳", "age" : 35, "dept" : "会计部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71757b"), "id" : 18, "name" : "常莎", "age" : 52, "dept" : "会计部" }
> db.list.find({name:/^张/})
{ "_id" : ObjectId("5ec932da930cacaa1c71757a"), "id" : 17, "name" : "张研", "age" : 45, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71759a"), "id" : 49, "name" : "张才旺", "age" : 30, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71759e"), "id" : 53, "name" : "张萌萌", "age" : 22, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175ab"), "id" : 66, "name" : "张园", "age" : 32, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175bd"), "id" : 83, "name" : "张静", "age" : 63, "dept" : "市场部" }
```

姓张或姓王的，按姓名排序

```
> db.list.find({name:/^[张王]/}).sort({name:1})
{ "_id" : ObjectId("5ec932da930cacaa1c7175ab"), "id" : 66, "name" : "张园", "age" : 32, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71759a"), "id" : 49, "name" : "张才旺", "age" : 30, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71757a"), "id" : 17, "name" : "张研", "age" : 45, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71759e"), "id" : 53, "name" : "张萌萌", "age" : 22, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175bd"), "id" : 83, "name" : "张静", "age" : 63, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175aa"), "id" : 65, "name" : "王君", "age" : 51, "dept" : "工程部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175a6"), "id" : 61, "name" : "王子龙", "age" : 33, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717584"), "id" : 27, "name" : "王希远", "age" : 22, "dept" : "教师部" }
{ "_id" : ObjectId("5ec932da930cacaa1c717570"), "id" : 7, "name" : "王新宇", "age" : 36, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175ba"), "id" : 81, "name" : "王晨旭", "age" : 18, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175cd"), "id" : 89, "name" : "王朋", "age" : 38, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175be"), "id" : 84, "name" : "王继蔚", "age" : 49, "dept" : "工程部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71758b"), "id" : 34, "name" : "王芳", "age" : 52, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175a8"), "id" : 63, "name" : "王超鑫", "age" : 28, "dept" : "客服部" }
> db.list.find({$or:[{"age":{$gt:60}},{"age":{$lt:20}}]}).sort({age:1})
{ "_id" : ObjectId("5ec932da930cacaa1c7175c6"), "id" : 93, "name" : "贾宝珠", "age" : 15, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175c2"), "id" : 88, "name" : "崔兴龙", "age" : 17, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175c7"), "id" : 94, "name" : "阎君", "age" : 17, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175ba"), "id" : 81, "name" : "王晨旭", "age" : 18, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175cf"), "id" : 102, "name" : "靳璠", "age" : 18, "dept" : "客服部" }
{ "_id" : ObjectId("5ec932da930cacaa1c71756c"), "id" : 3, "name" : "刘嘉", "age" : 63, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175bd"), "id" : 83, "name" : "张静", "age" : 63, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175c1"), "id" : 87, "name" : "孙洁", "age" : 63, "dept" : "市场部" }
{ "_id" : ObjectId("5ec932da930cacaa1c7175c3"), "id" : 90, "name" : "杨志超", "age" : 67, "dept" : "市场部"
```

distingct去重

```
> db.list.distinct("dept")
[ "会计部", "市场部", "教师部", "工程部", "客服部" ]
```

数据备份与恢复

备份案例

```
[root@localhost yum.repos.d]# yum -y install mariadb-server mariadb-devel mariadb-devel
```

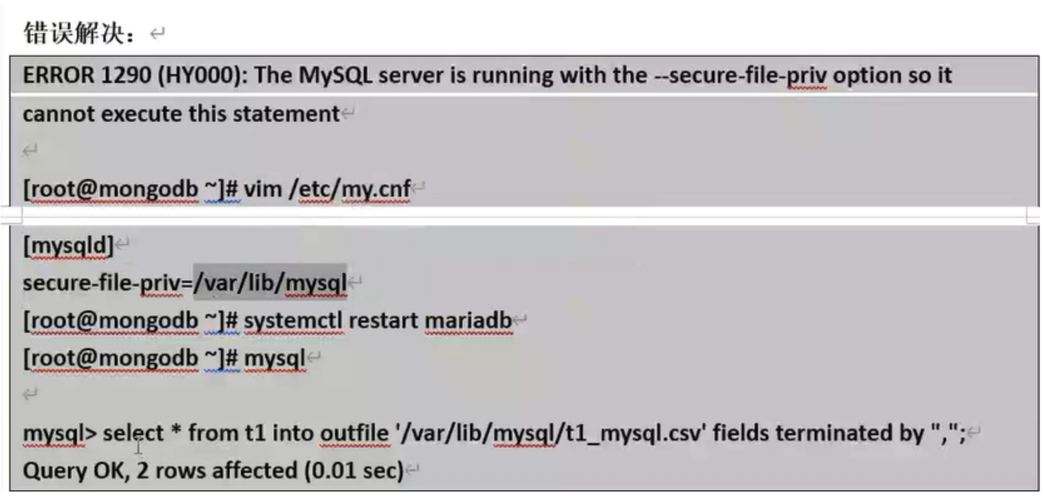
将mysql数据库内容导入MongoDB环境

```
[root@localhost ~]# systemctl start mariadb
[root@localhost ~]# mysql
MariaDB [(none)]>
MariaDB [(none)]> create database cloud;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> use cloud
Database changed
MariaDB [cloud]> create table t1(id int,name varchar(20));
Query OK, 0 rows affected (0.00 sec)
MariaDB [cloud]> insert into t1 values(1,"jack");
Query OK, 1 row affected (0.00 sec)
MariaDB [cloud]> insert into t1 values(2,"Rose");
Query OK, 1 row affected (0.00 sec)
MariaDB [cloud]> select * from t1;
+------+------+
| id   | name |
+------+------+
|    1 | jack |
|    2 | Rose |
+------+------+
2 rows in set (0.00 sec)
```

导出t1表里的内容到/var/lib/mysql/t1_mysql.csv文件，以“，”逗号分割

```
MariaDB [cloud]> select * from t1 into outfile '/var/lib/mysql/t1_mysql.csv' fields terminated by ",";
Query OK, 2 rows affected (0.00 sec)
[root@localhost ~]# ls /var/lib/mysql/
t1_mysql.csv
[root@localhost ~]# cat /var/lib/mysql/t1_mysql.csv
1,jack
2,Rose
```



错误解决：

```
ERROR 1290 (HY000): The MySQL server is running with the --secure-file-priv option so it
cannot execute this statement

[root@mongodb ~]# vim /etc/my.cnf
[mysqld]
secure-file-priv=/var/lib/mysql
[root@mongodb ~]# systemctl restart mariadb
[root@mongodb ~]# mysql

mysql> select * from t1 into outfile '/var/lib/mysql/t1_mysql.csv' fields terminated by ",";
Query OK, 2 rows affected (0.01 sec)
```

将/var/lib/mysql/t1_mysql.csv文件导入到MongoDB的cloud数据库下的tt1表，字段名称为id和name，文件类型为csv

```
[root@localhost ~]# mongoimport -d cloud -c tt1 -f id,name --file /var/lib/mysql/t1_mysql.csv --type csv
2020-05-24T00:06:00.702+0800    connected to: localhost
2020-05-24T00:06:00.711+0800    imported 2 documents
[root@localhost ~]# mongo
> db.tt1.find()
{ "_id" : ObjectId("5ec949e8930cacaa1c717640"), "id" : 2, "name" : "Rose" }
{ "_id" : ObjectId("5ec949e8930cacaa1c717641"), "id" : 1, "name" : "jack" }
> db.user.find()
{ "_id" : ObjectId("5ec8467b43d53389391e8077"), "id" : 1, "name" : "sofia" }
[root@localhost ~]# mongoexport -d cloud -c user -o /tmp/user.json
2020-05-24T00:08:45.149+0800    connected to: localhost
2020-05-24T00:08:45.150+0800    exported 1 record
[root@localhost ~]# cat /tmp/user.json
{"_id":{"$oid":"5ec8467b43d53389391e8077"},"id":1.0,"name":"sofia"}
```

mongodump备份

```
[root@localhost ~]# mkdir /backup
[root@localhost ~]# mongodump -d cloud -o /backup/
2020-05-24T00:12:00.906+0800    writing cloud.list to
2020-05-24T00:12:00.907+0800    writing cloud.tt1 to
2020-05-24T00:12:00.907+0800    writing cloud.user to
2020-05-24T00:12:00.908+0800    done dumping cloud.tt1 (2 documents)
2020-05-24T00:12:00.909+0800    done dumping cloud.list (102 documents)
2020-05-24T00:12:00.910+0800    done dumping cloud.user (1 document)
[root@localhost ~]# ll /backup/cloud/
总用量 28
-rw-r--r-- 1 root root 7899 5月  24 00:12 list.bson
-rw-r--r-- 1 root root  124 5月  24 00:12 list.metadata.json
-rw-r--r-- 1 root root   90 5月  24 00:12 tt1.bson
-rw-r--r-- 1 root root  123 5月  24 00:12 tt1.metadata.json
-rw-r--r-- 1 root root   50 5月  24 00:12 user.bson
-rw-r--r-- 1 root root  124 5月  24 00:12 user.metadata.json
```

mongorestore还原

```
[root@localhost ~]# mongorestore -d sifia --dir=/backup/cloud          # sifia写错了
2020-05-24T00:14:37.466+0800    the --db and --collection args should only be used when restoring from a BSON file. Other
uses are deprecated and will not exist in the future; use --nsInclude instead2020-05-24T00:14:37.466+0800    building a list of collections to restore from /backup/cloud dir
2020-05-24T00:14:37.469+0800    reading metadata for sifia.list from /backup/cloud/list.metadata.json
2020-05-24T00:14:37.471+0800    reading metadata for sifia.tt1 from /backup/cloud/tt1.metadata.json
2020-05-24T00:14:37.565+0800    restoring sifia.tt1 from /backup/cloud/tt1.bson
2020-05-24T00:14:37.576+0800    restoring sifia.list from /backup/cloud/list.bson
2020-05-24T00:14:37.615+0800    reading metadata for sifia.user from /backup/cloud/user.metadata.json
2020-05-24T00:14:37.710+0800    no indexes to restore
2020-05-24T00:14:37.710+0800    finished restoring sifia.tt1 (2 documents)
2020-05-24T00:14:37.715+0800    restoring sifia.user from /backup/cloud/user.bson
2020-05-24T00:14:37.754+0800    no indexes to restore
2020-05-24T00:14:37.754+0800    finished restoring sifia.user (1 document)
2020-05-24T00:14:37.757+0800    no indexes to restore
2020-05-24T00:14:37.757+0800    finished restoring sifia.list (102 documents)
```

```
2020-05-24T00:14:37.757+0800    done

[root@localhost ~]# mongo
> use sifia        #  将错就错
switched to db sifia
> show tables;
list
tt1
user
```

复制数据库

```
> db.copyDatabase("sofia","sofia2","localhost");
WARNING: db.copyDatabase is deprecated. See http://dochub.mongodb.org/core/copydb-clone-deprecation
{
    "note" : "Support for the copydb command has been deprecated. See http://dochub.mongodb.org/core/copydb-clone-depr
ecation",     "ok" : 1
}
> show dbs
admin   0.078GB
cloud   0.078GB
config  0.078GB
local   0.078GB
sifia   0.078GB
sofia2  0.078GB
study   0.078GB
test    0.078GB
```

两个MongoDB实例之间的数据导入

克隆集合

```
> use db1
switched to db db1
> db.student.insert({"id":1,"name":"zhangsan"});
WriteResult({ "nInserted" : 1 })
> exit
```

```
bye
[root@localhost ~]# mongo --port 27017
> db.runCommand({"cloneCollection":"db1.student","from":"localhost:27018"});
{ "ok" : 1 }
> show dbs
admin   0.078GB
cloud   0.078GB
config  0.078GB
db1     0.078GB
local   0.078GB
sifia   0.078GB
sofia2  0.078GB
study   0.078GB
test    0.078GB
> use db1
switched to db db1
> show tables;
student
> db.student.find();
{ "_id" : ObjectId("5ec94e13239698b2a7025767"), "id" : 1, "name" : "zhangsan" }
```

练习案例：

批量插入1万条数据

```
[root@localhost ~]# mongo
> use cloud
switched to db cloud
> show collections
list
tt1
user
> for(var i=1;i<=10000;i++) db.test.insert({"id":i,"name":"name"+i});
WriteResult({ "nInserted" : 1 })
```

仅显示5行信息

```
> db.test.find().limit(5)
{ "_id" : ObjectId("5ec950388cf30f63d5839cd2"), "id" : 1, "name" : "name1" }
{ "_id" : ObjectId("5ec950388cf30f63d5839cd3"), "id" : 2, "name" : "name2" }
{ "_id" : ObjectId("5ec950388cf30f63d5839cd4"), "id" : 3, "name" : "name3" }
{ "_id" : ObjectId("5ec950388cf30f63d5839cd5"), "id" : 4, "name" : "name4" }
{ "_id" : ObjectId("5ec950388cf30f63d5839cd6"), "id" : 5, "name" : "name5" }
```

条件导出，导出test表中id大于5000的文档

```
> db.test.find({"id":{"$gt":5000}}).limit(3);
{ "_id" : ObjectId("5ec9503a8cf30f63d583b05a"), "id" : 5001, "name" : "name5001" }
{ "_id" : ObjectId("5ec9503a8cf30f63d583b05b"), "id" : 5002, "name" : "name5002" }
{ "_id" : ObjectId("5ec9503a8cf30f63d583b05c"), "id" : 5003, "name" : "name5003" }
```

统计大于5000的行数量

```
> db.test.find({"id":{"$gt":5000}}).count();
5000
```

导出test集合到/tmp/test.json文件

```
[root@localhost ~]#
[root@localhost ~]# /usr/local/mongodb/bin/mongoexport -d cloud -c test -q
2020-05-24T00:40:29.372+0800    error parsing command line options: expected argument for flag `-q, --query'
2020-05-24T00:40:29.372+0800    try 'mongoexport --help' for more information
[root@localhost ~]# /usr/local/mongodb/bin/mongoexport -d cloud -c test -q '{"id":{"$gt":5000}}' -o /tmp/test.json
2020-05-24T00:43:34.332+0800    connected to: localhost
2020-05-24T00:43:34.406+0800    exported 5000 records
[root@localhost ~]# ll /tmp/test.json
-rw-r--r-- 1 root root 370002 5月  24 00:43 /tmp/test.json
[root@localhost ~]# wc -l /tmp/test.json
5000 /tmp/test.json
```

MongoDB监控管理

MongoDB安全管理

1. 配置MongoDB实例监听特定IP地址

```
[root@localhost ~]#
[root@localhost ~]# netstat -lnpt | grep mongod
tcp        0      0 127.0.0.1:27017         0.0.0.0:*               LISTEN      2427/mongod
tcp        0      0 127.0.0.1:27018         0.0.0.0:*               LISTEN      2862/mongod
[root@localhost ~]# vim /usr/local/mongodb/conf/mongodb1.conf
port=27017
dbpath=/data/mongodb1
logpath=/data/logs/mongodb/mongodb1.log
logappend=true
```

```
fork=true
maxConns=5000
storageEngine=mmapv1
slowms=1            # 慢查询日志，超过1秒的查询请求将被记录
profile=1           # 0=off，1=slow，2=all
bind_ip=192.168.200.110       # 限制监听特定IP
[root@localhost ~]# /etc/init.d/mongodb mongodb1 restart
killing process with pid: 2427
about to fork child process, waiting until server is ready for connections.
forked process: 7358
ERROR: child process failed, exited with error number 48
To see additional information in this output, start without the "--fork" option.
[root@localhost ~]# netstat -lnpt | grep mongod
tcp      0      0 192.168.200.110:27017   0.0.0.0:*            LISTEN      7379/mongod
tcp      0      0 127.0.0.1:27018       0.0.0.0:*            LISTEN      2862/mongod

[root@localhost ~]# mongo --host 192.168.200.110
```

配置实例授权启动，连接时必须验证用户密码

```
[root@localhost ~]# mongo --host 192.168.200.110
> db.createUser({"user":"root","pwd":"123456","roles":["root"]})
Successfully added user: { "user" : "root", "roles" : [ "root" ] }
{
	"_id" : "admin.root",
	"user" : "root",
	"db" : "admin",
	"credentials" : {
		"SCRAM-SHA-1" : {
			"iterationCount" : 10000,
			"salt" : "Y2Kodg47sMXTk4ELH53AGw==",
			"storedKey" : "6trjOoDVAA7xurxNrju770pVGzc=",
			"serverKey" : "f4Qquu3awwO3ErPGC2OurlEEiU4="
		},
		"SCRAM-SHA-256" : {
			"iterationCount" : 15000,
			"salt" : "ULqdqQUeIaDTTyHbqttK41OODo2ExCfZIRUGOA==",
			"storedKey" : "XZdPyUwbeoTJImaomiD3NeHRJiJMMVjY1mHPNcIIF7k=",
            "serverKey" : "jkIRB2KLR1w7N3J/CNsJmZUp0wVlDblhNZScOYuoJYg="
        }
    },
    "roles" : [
        {
            "role" : "root",
            "db" : "admin"
        }
    ]
}
```

限定IP地址和验证最好配一下

```
[root@localhost ~]# vim /usr/local/mongodb/conf/mongodb1.conf
port=27017
dbpath=/data/mongodb1
logpath=/data/logs/mongodb/mongodb1.log
logappend=true
fork=true
maxConns=5000
storageEngine=mmapv1
slowms=1
profile=1
bind_ip=192.168.200.110
auth=true
[root@localhost ~]# /etc/init.d/mongodb mongodb1 restart
killing process with pid: 7513
about to fork child process, waiting until server is ready for connections.
forked process: 7555
child process started successfully, parent exiting
[root@localhost ~]# mongo --host 192.168.200.110
MongoDB shell version v4.0.6
connecting to: mongodb://192.168.200.110:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("d2134aea-e4d5-434a-b4fd-0d443ee6298e") }
MongoDB server version: 4.0.6
> show dbs
> use admin
switched to db admin
> db.auth("root","123456")
1
```

还原环境

```
> db.dropUser("root")
true
[root@localhost ~]# vim /usr/local/mongodb/conf/mongodb1.conf
#bind_ip=192.168.200.110
#auth=true
```

MongoDB角色管理

built-ln roles　内置角色

MongoDB（内部的）进程管理

查看当前正在运行的进程：

```
[root@localhost ~]# mongo
> db.currentOp()
{
    "inprog" : [
        {
            "host" : "localhost:27017",
            "desc" : "conn3",
            "connectionId" : 3,
            "client" : "127.0.0.1:33188",
            "appName" : "MongoDB Shell",
            "clientMetadata" : {
                "application" : {
                    "name" : "MongoDB Shell"
                },
                "driver" : {
                    "name" : "MongoDB Internal Client",
                    "version" : "4.0.6"
                },
                "os" : {
                    "type" : "Linux",
                    "name" : "CentOS Linux release 7.5.1804 (Core) ",
                    "architecture" : "x86_64",
                    "version" : "Kernel 3.10.0-862.el7.x86_64"
                }
            },
            "active" : true,
            "currentOpTime" : "2020-05-24T08:08:39.859+0800",
            "opid" : 4547,
            "lsid" : {
                "id" : UUID("ccbd78b0-e68e-451a-809b-88c841204419"),
                "uid" : BinData(0,"47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=")
            },
            "secs_running" : NumberLong(0),
            "microsecs_running" : NumberLong(501),
            "op" : "command",
            "ns" : "admin.$cmd.aggregate",
            "command" : {
                "currentOp" : 1,
                "lsid" : {
                    "id" : UUID("ccbd78b0-e68e-451a-809b-88c841204419")
                },
                "$db" : "admin"
            },
            "numYields" : 0,
            "locks" : {

            },
            "waitingForLock" : false,
            "lockStats" : {

            }
        }
    ],
    "ok" : 1
}
```

杀掉当前正在运行的高消耗资源的进程：

```
>
> db.killOp(4547)
{ "info" : "attempting to kill op", "ok" : 1 }
```

MongoDB监控管理

查看数据库实例状态信息：

```
> db.serverStatus()
{
        "host" : "localhost",
        "version" : "4.0.6",
        "process" : "mongod",
        "pid" : NumberLong(7650),
        "uptime" : 25488,
        "uptimeMillis" : NumberLong(25487753),
        "uptimeEstimate" : NumberLong(25487),
        "localTime" : ISODate("2020-05-24T00:16:20.347Z"),
--------------------------------后面省略若干行
```

查看当前数据库统计信息:

```
> db.stats()
{
        "db" : "test",
        "collections" : 3,
        "views" : 0,
        "objects" : 5,
        "avgObjSize" : 86.4,
        "dataSize" : 432,
        "storageSize" : 20480,
        "numExtents" : 3,
        "indexes" : 1,
        "indexSize" : 8176,
        "fileSize" : 67108864,
        "nsSizeMB" : 16,
        "extentFreeList" : {
                "num" : 0,
                "totalSize" : 0
        },
        "dataFileVersion" : {
                "major" : 4,
                "minor" : 22
        },
        "fsUsedSize" : 6503837696,
        "fsTotalSize" : 53660876800,
        "ok" : 1
}
```

查看集合统计信息

```
> use cloud
```

```
switched to db cloud
> show tables;
list
test
tt1
user
> db.user.stats()
{
        "ns" : "cloud.user",
        "size" : 112,
        "count" : 1,
        "avgObjSize" : 112,
        "numExtents" : 1,
        "storageSize" : 8192,
        "lastExtentSize" : 8192,
        "paddingFactor" : 1,
        "paddingFactorNote" : "paddingFactor is unused and unmaintained in 3.0. It remains hard cod
ed to 1.0 for compatibility only.",     "userFlags" : 1,
        "capped" : false,
        "nindexes" : 1,
        "totalIndexSize" : 8176,
        "indexSizes" : {
                "_id_" : 8176
        },
        "ok" : 1
}
```

查看集合的大小：

```
> db.user.dataSize()
112
```

mongostat是mongodb自带的状态检测工具----数据库变慢及其他问题

间隔固定时间获取mongodb的当前运行状态，并输出

```
[root@localhost ~]# /usr/local/mongodb/bin/mongostat
[root@localhost ~]# mongostat
insert query update delete getmore command flushes mapped vsize  res faults qrw arw net_in net_out
conn              time
    *0    *0    *0    *0      0    2|0      0       2.38G 194M      0 0|0 0|0   158b   32.0k    2 May 24 08:30:10.729
```

```
       *0    *0     *0     *0       0     2|0       0        2.38G 194M      0 0|0 0|0   158b    32.0k     2 May 24
08:30:11.728
后面还有很多----------------
[root@localhost ~]# mongotop
2020-05-24T08:31:04.310+0800      connected to: 127.0.0.1

                          ns    total    read    write    2020-05-24T08:31:05+08:00
  admin.$cmd.aggregate     0ms     0ms      0ms
  admin.system.indexes     0ms     0ms      0ms
admin.system.namespaces     0ms     0ms      0ms
  admin.system.profile     0ms     0ms      0ms
    admin.system.roles     0ms     0ms      0ms
    admin.system.users     0ms     0ms      0ms
  admin.system.version     0ms     0ms      0ms
            cloud.list     0ms     0ms      0ms
  cloud.system.indexes     0ms     0ms      0ms
cloud.system.namespaces     0ms     0ms      0ms
```

输出结果字段说明

ns: 数据库命名空间（数据库名称和集合）

db: 数据库名称，.的数据库针对全局锁定，而非特定

total: mongod花费的时间工作在这个命名空间提供总额

read: 读

write: 写

MongoDB复制集集群部署及管理

MongoDB复制

将一个数据库实例中的所有数据改变复制到另一个独立的数据库实例的过程，默认是
主从复制集群(未来不再使用)。
缺点是一旦主库出现故障，需要手动把主库角色切换到最可靠的从库上，而其他从库还需配置从新的主库去同步。
复制是将数据同步在多个服务器的过程。
●复制提供了数据的冗余备份，并在多个服务器上存储数据副本，提高了数据的可用性，

并可以保证数据的安全性。

●复制还允许您从硬件故障和服务中断中恢复数据。

复制的特征

● 保障 数据的安全性

● 数据高可用性(7*24)？

● 数据灾难恢复

● 无需停机维护 (如备份，重建索引，压缩)

● 分布式读取数据

MongoDB复制集ReplSet (副本集)

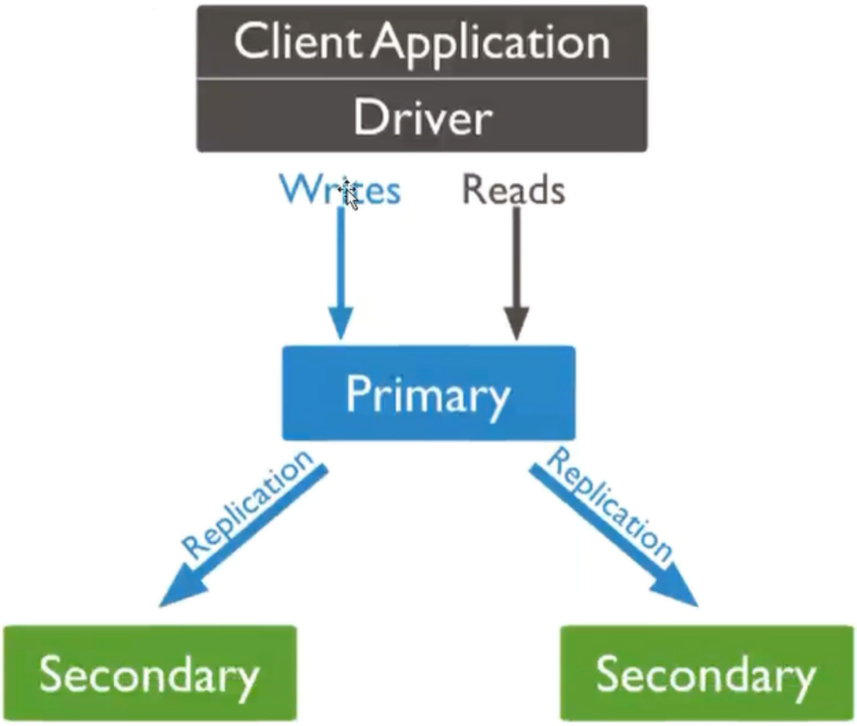
原理上也是MongoDB主从复制技术，但当主库出现故障时，能**自动**实现主从切换，从而故障得以恢复，其他从库自动从新的主库上同步数据，整个过程不需要手动干预。类似于MySQL中的MHA技术。

● MongoDB的复制至少需要两个节点。其中一个是主节点，负责处理客户端请求，其余的都是从节点，负责复制主节点上的数据，达到数据- -致性。

● MongoDB各个节点常见的搭配方式为:一主一从、一主多从。

● 主节点记录在其上的所有操作oplog,从节点定期轮询主节点获取这些操作，然后对自己的数据副本执行这些操作，从而保证从节点的数据与主节点一致。

MongoDB复制结构图如下所示:



以上结构图中，客户端从主节点读取数据，在客户端写入数据到主节点时，主节点与

从节点进行数据交互保障数据的一致性。

复制集的特征

- N个节点的集群
- 任何节点可作为主节点
- 所有写入操作都在主节点上
- 自动故障转移
- 自动故障恢复

构建MongoDB复制集集群

删除之前的实例

```
[root@localhost ~]# /etc/init.d/mongodb mongodb1 stop
killing process with pid: 7650
[root@localhost ~]# /etc/init.d/mongodb mongodb2 stop
killing process with pid: 2862
[root@localhost ~]# rm -rf /data/
```

配置4个MongoDB实例

```
[root@localhost ~]# vim /usr/local/mongodb/conf/mongodb1.conf
port=27017
dbpath=/data/mongodb1
logpath=/data/logs/mongodb/mongodb1.log
logappend=true
fork=true
maxConns=5000
storageEngine=mmapv1
slowms=1
profile=1
replSet=sofia        // 集群的名字；名字可以随意起；这里面的实例指定的名字一定要相同
[root@localhost ~]# vim /usr/local/mongodb/conf/mongodb2.conf
port=27018
dbpath=/data/mongodb2
logpath=/data/logs/mongodb/mongodb2.log
logappend=true
fork=true
maxConns=5000
storageEngine=mmapv1
slowms=1
```

```
profile=1
replSet=sofia
[root@localhost ~]# vim /usr/local/mongodb/conf/mongodb3.conf
port=27019
dbpath=/data/mongodb3
logpath=/data/logs/mongodb/mongodb3.log
logappend=true
fork=true
maxConns=5000
storageEngine=mmapv1
slowms=1
profile=1
replSet=sofia
[root@localhost ~]# vim /usr/local/mongodb/conf/mongodb4.conf
port=27020
dbpath=/data/mongodb4
logpath=/data/logs/mongodb/mongodb4.log
logappend=true
fork=true
maxConns=5000
storageEngine=mmapv1
slowms=1
profile=1
replSet=sofia
[root@localhost ~]# mkdir /data/mongodb{1..4} -p
[root@localhost ~]# mkdir /data/logs/mongodb -p
[root@localhost ~]# ls /data
logs  mongodb1  mongodb2  mongodb3  mongodb4
[root@localhost ~]# touch /data/logs/mongodb/mongodb{1..4}.log
[root@localhost ~]# chmod 777 /data/logs/mongodb/mongodb*
[root@localhost ~]# ll /data/logs/mongodb/mongodb*
-rwxrwxrwx 1 root root 0 5月  24 09:25 /data/logs/mongodb/mongodb1.log
-rwxrwxrwx 1 root root 0 5月  24 09:25 /data/logs/mongodb/mongodb2.log
-rwxrwxrwx 1 root root 0 5月  24 09:25 /data/logs/mongodb/mongodb3.log
-rwxrwxrwx 1 root root 0 5月  24 09:25 /data/logs/mongodb/mongodb4.log
```

```
[root@localhost ~]# /etc/init.d/mongodb mongodb1 start
about to fork child process, waiting until server is ready for connections.
forked process: 10996
child process started successfully, parent exiting
[root@localhost ~]# /etc/init.d/mongodb mongodb2 start
about to fork child process, waiting until server is ready for connections.
forked process: 11030
child process started successfully, parent exiting
[root@localhost ~]# /etc/init.d/mongodb mongodb3 start
about to fork child process, waiting until server is ready for connections.
forked process: 11057
child process started successfully, parent exiting
[root@localhost ~]# /etc/init.d/mongodb mongodb4 start
about to fork child process, waiting until server is ready for connections.
forked process: 11083
child process started successfully, parent exiting
[root@localhost ~]# netstat -lnpt | grep mongod
tcp        0      0 127.0.0.1:27017          0.0.0.0:*               LISTEN
10996/mongod
tcp        0      0 127.0.0.1:27018          0.0.0.0:*               LISTEN
11030/mongod
tcp        0      0 127.0.0.1:27019          0.0.0.0:*               LISTEN
11057/mongod
tcp        0      0 127.0.0.1:27020          0.0.0.0:*               LISTEN
11083/mongod
```

在生产环境中要限定IP，而不是用127.0.0.1

```
[root@localhost ~]# mongo
> rs.help()              # 查看复制集帮助指令
> rs.status()
{
        "operationTime" : Timestamp(0, 0),
        "ok" : 0,
        "errmsg" : "no replset config has been received",
        "code" : 94,
        "codeName" : "NotYetInitialized",
        "$clusterTime" : {
                "clusterTime" : Timestamp(0, 0),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

配置集群

```
>              cfg={"_id":"sofia","members":[{"_id":0,"host":"127.0.0.1:27017"},
{"_id":1,"host":"127.0.0.1:27018"},{"_id":2,"host":"127.0.0.1:27019"}
]}{
        "_id" : "sofia",
        "members" : [
                {
                        "_id" : 0,
                        "host" : "127.0.0.1:27017"
                },
                {
                        "_id" : 1,
                        "host" : "127.0.0.1:27018"
                },
                {
                        "_id" : 2,
                        "host" : "127.0.0.1:27019"
                }
        ]
}
> rs.initiate(cfg)                    # 一主两从
{
        "ok" : 1,
        "operationTime" : Timestamp(1590284592, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590284592, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
sofia:SECONDARY>
sofia:PRIMARY>        # 当前主机别选为主
sofia:PRIMARY> rs.status()
{
        "set" : "sofia",
        "date" : ISODate("2020-05-24T01:44:10.800Z"),
        "myState" : 1,
        "term" : NumberLong(1),
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1590284644, 1),
                        "t" : NumberLong(1)
                },
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1590284644, 1),
                        "t" : NumberLong(1)
                },
                "appliedOpTime" : {
                        "ts" : Timestamp(1590284644, 1),
                        "t" : NumberLong(1)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1590284644, 1),
                        "t" : NumberLong(1)
                }
        },
        ″members″ : [
                {
                        ″_id″ : 0,
                        ″name″ : ″127.0.0.1:27017″,
                        ″health″ : 1,
                        ″state″ : 1,
                        ″stateStr″ : ″PRIMARY″,
                        ″uptime″ : 933,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590284644, 1),
                                ″t″ : NumberLong(1)
                        },
                        ″optimeDate″ : ISODate(″2020-05-24T01:44:04Z″),
                        ″syncingTo″ : ″″,
                        ″syncSourceHost″ : ″″,
                        ″syncSourceId″ : -1,
                        ″infoMessage″ : ″could not find member to sync from″,
                        ″electionTime″ : Timestamp(1590284603, 1),
                        ″electionDate″ : ISODate(″2020-05-24T01:43:23Z″),
                        ″configVersion″ : 1,
                        ″self″ : true,
                        ″lastHeartbeatMessage″ : ″″
                },
                {
                        ″_id″ : 1,
                        ″name″ : ″127.0.0.1:27018″,
                        ″health″ : 1,
                        ″state″ : 2,
                        ″stateStr″ : ″SECONDARY″,
                        ″uptime″ : 58,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590284644, 1),
                                ″t″ : NumberLong(1)
                        },
                        ″optimeDurable″ : {
```

```
                "ts" : Timestamp(1590284644, 1),
                "t" : NumberLong(1)
            },
            "optimeDate" : ISODate("2020-05-24T01:44:04Z"),
            "optimeDurableDate" : ISODate("2020-05-24T01:44:04Z"),
            "lastHeartbeat" : ISODate("2020-05-24T01:44:09.758Z"),
            "lastHeartbeatRecv"     :     ISODate("2020-05-
24T01:44:09.678Z"),
            "pingMs" : NumberLong(0),
            "lastHeartbeatMessage" : "",
            "syncingTo" : "127.0.0.1:27017",
            "syncSourceHost" : "127.0.0.1:27017",
            "syncSourceId" : 0,
            "infoMessage" : "",
            "configVersion" : 1
        },
        {
            "_id" : 2,
            "name" : "127.0.0.1:27019",
            "health" : 1,      # 健康状态；为1表示节点健康；0表示当
机
            "state" : 2,        # 1 表示主；2表示从
            "stateStr" : "SECONDARY",
            "uptime" : 58,
            "optime" : {
                "ts" : Timestamp(1590284644, 1),
                "t" : NumberLong(1)
            },
            "optimeDurable" : {
                "ts" : Timestamp(1590284644, 1),
                "t" : NumberLong(1)
            },
            "optimeDate" : ISODate("2020-05-24T01:44:04Z"),
            "optimeDurableDate" : ISODate("2020-05-24T01:44:04Z"),
            "lastHeartbeat" : ISODate("2020-05-24T01:44:09.759Z"),
                        "lastHeartbeatRecv" : ISODate("2020-05-24T01:44:09.785Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "127.0.0.1:27017",
                        "syncSourceHost" : "127.0.0.1:27017",
                        "syncSourceId" : 0,
                        "infoMessage" : "",
                        "configVersion" : 1
                }
        ],
        "ok" : 1,
        "operationTime" : Timestamp(1590284644, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590284644, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

添加节点

```
sofia:PRIMARY> rs.add("127.0.0.1:27020")
{
        "ok" : 1,
        "operationTime" : Timestamp(1590284889, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590284889, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
sofia:PRIMARY> rs.status()
```

此处省略数行：

```
        {
                        ″_id″ : 3,
                        ″name″ : ″127.0.0.1:27020″,
                        ″health″ : 1,
                        ″state″ : 2,
                        ″stateStr″ : ″SECONDARY″,
                        ″uptime″ : 26,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590284914, 1),
                                ″t″ : NumberLong(1)
                        },
                        ″optimeDurable″ : {
                                ″ts″ : Timestamp(1590284914, 1),
                                ″t″ : NumberLong(1)
                        },
                        ″optimeDate″ : ISODate(″2020-05-24T01:48:34Z″),
                        ″optimeDurableDate″ : ISODate(″2020-05-24T01:48:34Z″),
                        ″lastHeartbeat″ : ISODate(″2020-05-24T01:48:35.810Z″),
                        ″lastHeartbeatRecv″     :     ISODate(″2020-05-
24T01:48:35.342Z″),
                        ″pingMs″ : NumberLong(0),
                        ″lastHeartbeatMessage″ : ″″,
                        ″syncingTo″ : ″127.0.0.1:27018″,
                        ″syncSourceHost″ : ″127.0.0.1:27018″,
                        ″syncSourceId″ : 1,
                        ″infoMessage″ : ″″,
                        ″configVersion″ : 2
                }
        ],
        ″ok″ : 1,
        ″operationTime″ : Timestamp(1590284914, 1),
        ″$clusterTime″ : {
                ″clusterTime″ : Timestamp(1590284914, 1),
                ″signature″ : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

到此为止：一主三从

删除节点：

```
sofia:PRIMARY> rs.remove("127.0.0.1:27020")
{
        "ok" : 1,
        "operationTime" : Timestamp(1590285057, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590285057, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
sofia:PRIMARY> rs.status()              # 已查看不到27020的节点了
```

查看实例的状态

```
[root@localhost ~]# ps aux | grep mongod
root        10996  0.7  3.1 10334960 63936 ?        Sl   09:28   0:10
/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/mongodb1.conf
root        11030  0.8  3.0 6113788 60912 ?        Sl   09:28   0:11
/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/mongodb2.conf
root        11057  0.8  2.7 6046200 56136 ?        Sl   09:28   0:11
/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/mongodb3.conf
root        11083  0.6  2.4 6011380 50304 ?        Sl   09:28   0:09
/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/conf/mongodb4.conf
root      11592  0.0  0.0 112720   984 pts/1    S+   09:52   0:00 grep --
color=auto mongod
[root@localhost ~]# kill -9 10996          # 模拟主节点出现故障

[root@localhost ~]# mongo --port 27018
```

```
sofia:PRIMARY>
sofia:PRIMARY> rs.status()
{
        ″set″ : ″sofia″,
        ″date″ : ISODate(″2020-05-24T02:00:41.489Z″),
        ″myState″ : 1,
        ″term″ : NumberLong(2),
        ″syncingTo″ : ″″,
        ″syncSourceHost″ : ″″,
        ″syncSourceId″ : -1,
        ″heartbeatIntervalMillis″ : NumberLong(2000),
        ″optimes″ : {
                ″lastCommittedOpTime″ : {
                        ″ts″ : Timestamp(1590285635, 1),
                        ″t″ : NumberLong(2)
                },
                ″readConcernMajorityOpTime″ : {
                        ″ts″ : Timestamp(1590285635, 1),
                        ″t″ : NumberLong(2)
                },
                ″appliedOpTime″ : {
                        ″ts″ : Timestamp(1590285635, 1),
                        ″t″ : NumberLong(2)
                },
                ″durableOpTime″ : {
                        ″ts″ : Timestamp(1590285635, 1),
                        ″t″ : NumberLong(2)
                }
        },
        ″members″ : [
                {
                        ″_id″ : 0,
                        ″name″ : ″127.0.0.1:27017″,
                        ″health″ : 0,
                        ″state″ : 8,
                        "stateStr" : "(not reachable/healthy)",
                        "uptime" : 0,
                        "optime" : {
                                "ts" : Timestamp(0, 0),
                                "t" : NumberLong(-1)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(0, 0),
                                "t" : NumberLong(-1)
                        },
                        "optimeDate" : ISODate("1970-01-01T00:00:00Z"),
                        "optimeDurableDate" : ISODate("1970-01-01T00:00:00Z"),
                        "lastHeartbeat" : ISODate("2020-05-24T02:00:39.559Z"),
                        "lastHeartbeatRecv" : ISODate("2020-05-
24T01:59:21.941Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "Error connecting to
127.0.0.1:27017 :: caused by :: Connection refused",
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "configVersion" : -1
                },
                {
                        "_id" : 1,
                        "name" : "127.0.0.1:27018",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 1916,
                        "optime" : {
                                "ts" : Timestamp(1590285635, 1),
                                "t" : NumberLong(2)
                        },
                        "optimeDate" : ISODate("2020-05-24T02:00:35Z"),
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "could not find member to sync from",
                        "electionTime" : Timestamp(1590285573, 1),
                        "electionDate" : ISODate("2020-05-24T01:59:33Z"),
                        "configVersion" : 3,
                        "self" : true,
                        "lastHeartbeatMessage" : ""
                },
                {
                        "_id" : 2,
                        "name" : "127.0.0.1:27019",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 1047,
                        "optime" : {
                                "ts" : Timestamp(1590285635, 1),
                                "t" : NumberLong(2)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1590285635, 1),
                                "t" : NumberLong(2)
                        },
                        "optimeDate" : ISODate("2020-05-24T02:00:35Z"),
                        "optimeDurableDate" : ISODate("2020-05-24T02:00:35Z"),
                        "lastHeartbeat" : ISODate("2020-05-24T02:00:39.533Z"),
                        "lastHeartbeatRecv" : ISODate("2020-05-24T02:00:40.317Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "127.0.0.1:27018",
                        "syncSourceHost" : "127.0.0.1:27018",
                        "syncSourceId" : 1,
                        "infoMessage" : "",
                        "configVersion" : 3
                }
        ],
        "ok" : 1,
        "operationTime" : Timestamp(1590285635, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590285635, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
-----------主节点变更为27018
[root@localhost ~]# /etc/init.d/mongodb mongodb1 start
about to fork child process, waiting until server is ready for connections.
forked process: 11802
child process started successfully, parent exiting
[root@localhost ~]# mongo         # 默认连接到27017
sofia:SECONDARY> rs.status()
{
        "set" : "sofia",
        "date" : ISODate("2020-05-24T02:03:37.104Z"),
        "myState" : 2,
        "term" : NumberLong(2),
        "syncingTo" : "127.0.0.1:27019",
        "syncSourceHost" : "127.0.0.1:27019",
        "syncSourceId" : 2,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1590285815, 1),
                        "t" : NumberLong(2)
                },
                ″readConcernMajorityOpTime″ : {
                        ″ts″ : Timestamp(1590285815, 1),
                        ″t″ : NumberLong(2)
                },
                ″appliedOpTime″ : {
                        ″ts″ : Timestamp(1590285815, 1),
                        ″t″ : NumberLong(2)
                },
                ″durableOpTime″ : {
                        ″ts″ : Timestamp(1590285815, 1),
                        ″t″ : NumberLong(2)
                }
        },
        ″members″ : [
                {
                        ″_id″ : 0,
                        ″name″ : ″127.0.0.1:27017″,
                        ″health″ : 1,
                        ″state″ : 2,
                        ″stateStr″ : ″SECONDARY″,
                        ″uptime″ : 67,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590285815, 1),
                                ″t″ : NumberLong(2)
                        },
                        ″optimeDate″ : ISODate(″2020-05-24T02:03:35Z″),
                        ″syncingTo″ : ″127.0.0.1:27019″,
                        ″syncSourceHost″ : ″127.0.0.1:27019″,
                        ″syncSourceId″ : 2,
                        ″infoMessage″ : ″″,
                        ″configVersion″ : 3,
                        ″self″ : true,
                        ″lastHeartbeatMessage″ : ″″
                },
                {
                        ″_id″ : 1,
                        ″name″ : ″127.0.0.1:27018″,
                        ″health″ : 1,
                        ″state″ : 1,
                        ″stateStr″ : ″PRIMARY″,
                        ″uptime″ : 66,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590285815, 1),
                                ″t″ : NumberLong(2)
                        },
                        ″optimeDurable″ : {
                                ″ts″ : Timestamp(1590285815, 1),
                                ″t″ : NumberLong(2)
                        },
                        ″optimeDate″ : ISODate(″2020-05-24T02:03:35Z″),
                        ″optimeDurableDate″ : ISODate(″2020-05-24T02:03:35Z″),
                        ″lastHeartbeat″ : ISODate(″2020-05-24T02:03:35.686Z″),
                        ″lastHeartbeatRecv″    :    ISODate(″2020-05-
        24T02:03:35.711Z″),
                        ″pingMs″ : NumberLong(0),
                        ″lastHeartbeatMessage″ : ″″,
                        ″syncingTo″ : ″″,
                        ″syncSourceHost″ : ″″,
                        ″syncSourceId″ : -1,
                        ″infoMessage″ : ″″,
                        ″electionTime″ : Timestamp(1590285573, 1),
                        ″electionDate″ : ISODate(″2020-05-24T01:59:33Z″),
                        ″configVersion″ : 3
                },
                {
                        ″_id″ : 2,
                        ″name″ : ″127.0.0.1:27019″,
                        ″health″ : 1,
                        ″state″ : 2,
                        ″stateStr″ : ″SECONDARY″,
                        ″uptime″ : 66,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590285815, 1),
                                ″t″ : NumberLong(2)
                        },
                        ″optimeDurable″ : {
                                ″ts″ : Timestamp(1590285815, 1),
                                ″t″ : NumberLong(2)
                        },
                        ″optimeDate″ : ISODate(″2020-05-24T02:03:35Z″),
                        ″optimeDurableDate″ : ISODate(″2020-05-24T02:03:35Z″),
                        ″lastHeartbeat″ : ISODate(″2020-05-24T02:03:35.692Z″),
                        ″lastHeartbeatRecv″ : ISODate(″2020-05-24T02:03:36.525Z″),
                        ″pingMs″ : NumberLong(0),
                        ″lastHeartbeatMessage″ : ″″,
                        ″syncingTo″ : ″127.0.0.1:27018″,
                        ″syncSourceHost″ : ″127.0.0.1:27018″,
                        ″syncSourceId″ : 1,
                        ″infoMessage″ : ″″,
                        ″configVersion″ : 3
                }
        ],
        ″ok″ : 1,
        ″operationTime″ : Timestamp(1590285815, 1),
        ″$clusterTime″ : {
                ″clusterTime″ : Timestamp(1590285815, 1),
                ″signature″ : {
                        ″hash″ : BinData(0,″AAAAAAAAAAAAAAAAAAAAAAAAAAA=″),
                        ″keyId″ : NumberLong(0)
                }
        }
}
[root@localhost ~]# mongo --port 27017
```

```
sofia:PRIMARY> rs.freeze(30)          # 暂停30秒不参加选举27018
sofia:SECONDARY>  rs.freeze(30)          # 暂停30秒不参加选举
{
        ″ok″ : 1,
        ″operationTime″ : Timestamp(1590286208, 1),
        ″$clusterTime″ : {
                ″clusterTime″ : Timestamp(1590286208, 1),
                ″signature″ : {
                        ″hash″ : BinData(0,″AAAAAAAAAAAAAAAAAAAAAAAAAAA=″),
                        ″keyId″ : NumberLong(0)
                }
        }
}
[root@localhost ~]# /etc/init.d/mongodb mongodb3 stop
killing process with pid: 11057
sofia:SECONDARY> rs.status()
{
        ″set″ : ″sofia″,
        ″date″ : ISODate(″2020-05-24T02:11:46.847Z″),
        ″myState″ : 2,
        ″term″ : NumberLong(3),
        ″syncingTo″ : ″127.0.0.1:27019″,
        ″syncSourceHost″ : ″127.0.0.1:27019″,
        ″syncSourceId″ : 2,
        ″heartbeatIntervalMillis″ : NumberLong(2000),
        ″optimes″ : {
                ″lastCommittedOpTime″ : {
                        ″ts″ : Timestamp(1590286298, 1),
                        ″t″ : NumberLong(3)
                },
                ″readConcernMajorityOpTime″ : {
                        ″ts″ : Timestamp(1590286298, 1),
                        ″t″ : NumberLong(3)
                },
                ″appliedOpTime″ : {
```

```
				″ts″ : Timestamp(1590286298, 1),
				″t″ : NumberLong(3)
			},
			″durableOpTime″ : {
				″ts″ : Timestamp(1590286298, 1),
				″t″ : NumberLong(3)
			}
		},
		″members″ : [
			{
				″_id″ : 0,
				″name″ : ″127.0.0.1:27017″,
				″health″ : 1,
				″state″ : 2,
				″stateStr″ : ″SECONDARY″,
				″uptime″ : 556,
				″optime″ : {
					″ts″ : Timestamp(1590286298, 1),
					″t″ : NumberLong(3)
				},
				″optimeDate″ : ISODate(″2020-05-24T02:11:38Z″),
				″syncingTo″ : ″127.0.0.1:27019″,
				″syncSourceHost″ : ″127.0.0.1:27019″,
				″syncSourceId″ : 2,
				″infoMessage″ : ″″,
				″configVersion″ : 3,
				″self″ : true,
				″lastHeartbeatMessage″ : ″″
			},
			{
				″_id″ : 1,
				″name″ : ″127.0.0.1:27018″,
				″health″ : 0,
				″state″ : 8,
				″stateStr″ : ″(not reachable/healthy)″,
                        ″uptime″ : 0,
                        ″optime″ : {
                                ″ts″ : Timestamp(0, 0),
                                ″t″ : NumberLong(-1)
                        },
                        ″optimeDurable″ : {
                                ″ts″ : Timestamp(0, 0),
                                ″t″ : NumberLong(-1)
                        },
                        ″optimeDate″ : ISODate(″1970-01-01T00:00:00Z″),
                        ″optimeDurableDate″ : ISODate(″1970-01-01T00:00:00Z″),
                        ″lastHeartbeat″ : ISODate(″2020-05-24T02:11:45.233Z″),
                        ″lastHeartbeatRecv″    :    ISODate(″2020-05-
24T02:09:35.853Z″),
                        ″pingMs″ : NumberLong(0),
                        ″lastHeartbeatMessage″  :  ″Error  connecting  to
127.0.0.1:27018 :: caused by :: Connection refused″,
                        ″syncingTo″ : ″″,
                        ″syncSourceHost″ : ″″,
                        ″syncSourceId″ : -1,
                        ″infoMessage″ : ″″,
                        ″configVersion″ : -1
                },
                {
                        ″_id″ : 2,
                        ″name″ : ″127.0.0.1:27019″,
                        ″health″ : 1,
                        ″state″ : 1,
                        ″stateStr″ : ″PRIMARY″,
                        ″uptime″ : 134,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590286298, 1),
                                ″t″ : NumberLong(3)
                        },
                        ″optimeDurable″ : {
                                "ts" : Timestamp(1590286298, 1),
                                "t" : NumberLong(3)
                        },
                        "optimeDate" : ISODate("2020-05-24T02:11:38Z"),
                        "optimeDurableDate" : ISODate("2020-05-24T02:11:38Z"),
                        "lastHeartbeat" : ISODate("2020-05-24T02:11:45.159Z"),
                        "lastHeartbeatRecv" : ISODate("2020-05-24T02:11:46.183Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "electionTime" : Timestamp(1590286186, 1),
                        "electionDate" : ISODate("2020-05-24T02:09:46Z"),
                        "configVersion" : 3
                }
        ],
        "ok" : 1,
        "operationTime" : Timestamp(1590286298, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590286298, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

解除暂停选举的30秒

```
sofia:SECONDARY> rs.freeze(0)
{
        "info" : "unfreezing",
        "ok" : 1,
        "operationTime" : Timestamp(1590286398, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590286398, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
[root@localhost ~]# mongo
sofia:SECONDARY> rs.status()
{
        "set" : "sofia",
        "date" : ISODate("2020-05-24T02:14:53.584Z"),
        "myState" : 2,
        "term" : NumberLong(3),
        "syncingTo" : "127.0.0.1:27019",
        "syncSourceHost" : "127.0.0.1:27019",
        "syncSourceId" : 2,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1590286488, 1),
                        "t" : NumberLong(3)
                },
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1590286488, 1),
                        "t" : NumberLong(3)
                },
                "appliedOpTime" : {
                        "ts" : Timestamp(1590286488, 1),
                        "t" : NumberLong(3)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1590286488, 1),
                        ″t″ : NumberLong(3)
                }
        },
        ″members″ : [
                {
                        ″_id″ : 0,
                        ″name″ : ″127.0.0.1:27017″,
                        ″health″ : 1,
                        ″state″ : 2,
                        ″stateStr″ : ″SECONDARY″,
                        ″uptime″ : 743,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590286488, 1),
                                ″t″ : NumberLong(3)
                        },
                        ″optimeDate″ : ISODate(″2020-05-24T02:14:48Z″),
                        ″syncingTo″ : ″127.0.0.1:27019″,
                        ″syncSourceHost″ : ″127.0.0.1:27019″,
                        ″syncSourceId″ : 2,
                        ″infoMessage″ : ″″,
                        ″configVersion″ : 3,
                        ″self″ : true,
                        ″lastHeartbeatMessage″ : ″″
                },
                {
                        ″_id″ : 1,
                        ″name″ : ″127.0.0.1:27018″,
                        ″health″ : 1,
                        ″state″ : 2,
                        ″stateStr″ : ″SECONDARY″,
                        ″uptime″ : 16,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590286488, 1),
                                ″t″ : NumberLong(3)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1590286488, 1),
                                "t" : NumberLong(3)
                        },
                        "optimeDate" : ISODate("2020-05-24T02:14:48Z"),
                        "optimeDurableDate" : ISODate("2020-05-24T02:14:48Z"),
                        "lastHeartbeat" : ISODate("2020-05-24T02:14:53.392Z"),
                        "lastHeartbeatRecv"    :    ISODate("2020-05-
24T02:14:51.790Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "127.0.0.1:27019",
                        "syncSourceHost" : "127.0.0.1:27019",
                        "syncSourceId" : 2,
                        "infoMessage" : "",
                        "configVersion" : 3
                },
                {
                        "_id" : 2,
                        "name" : "127.0.0.1:27019",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 321,
                        "optime" : {
                                "ts" : Timestamp(1590286488, 1),
                                "t" : NumberLong(3)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1590286488, 1),
                                "t" : NumberLong(3)
                        },
                        "optimeDate" : ISODate("2020-05-24T02:14:48Z"),
                        "optimeDurableDate" : ISODate("2020-05-24T02:14:48Z"),
                        "lastHeartbeat" : ISODate("2020-05-24T02:14:53.229Z"),
                        ″lastHeartbeatRecv″ : ISODate(″2020-05-24T02:14:52.252Z″),
                        ″pingMs″ : NumberLong(0),
                        ″lastHeartbeatMessage″ : ″″,
                        ″syncingTo″ : ″″,
                        ″syncSourceHost″ : ″″,
                        ″syncSourceId″ : -1,
                        ″infoMessage″ : ″″,
                        ″electionTime″ : Timestamp(1590286186, 1),
                        ″electionDate″ : ISODate(″2020-05-24T02:09:46Z″),
                        ″configVersion″ : 3
                }
        ],
        ″ok″ : 1,
        ″operationTime″ : Timestamp(1590286488, 1),
        ″$clusterTime″ : {
                ″clusterTime″ : Timestamp(1590286488, 1),
                ″signature″ : {
                        ″hash″ : BinData(0,″AAAAAAAAAAAAAAAAAAAAAAAAAAA=″),
                        ″keyId″ : NumberLong(0)
                }
        }
}
==========================================
```

当模拟mongodb1实例故障时，由于mongodb2实例不参与选举，因此mongodb3也就是27019将会成为新的主节点

现在集群的状态：一主三从；27018为主，其余为从

模拟故障：

方法一：

```
kill -9 进程号
```

方法二：

```
sofia:PRIMARY>rs.stepDown(60,30)  # 手动修改为从节点
```

4.4、 复制集选举原理

1、复制的原理

MongoDB复制是基于操作日志oplog. 实现，oplog. 相当于mysqL中的二:进制日志，只记录数据发生的改变操作。

2、选举的原理

(1)节点类型: **标准节点，被动节点，仲裁节点**

　　只有标准节点可能被选举为活跃(主)节点，有选举权

　　被动节点有完整副本，不可能成为活跃节点，有选举权

　　仲裁节点不复制数据，不可能成为活跃节点，只有选举权

(2)标准节点与被动节点的区别

　　priority值高者是标准节点，低者则为被动节点

(3)选举规则

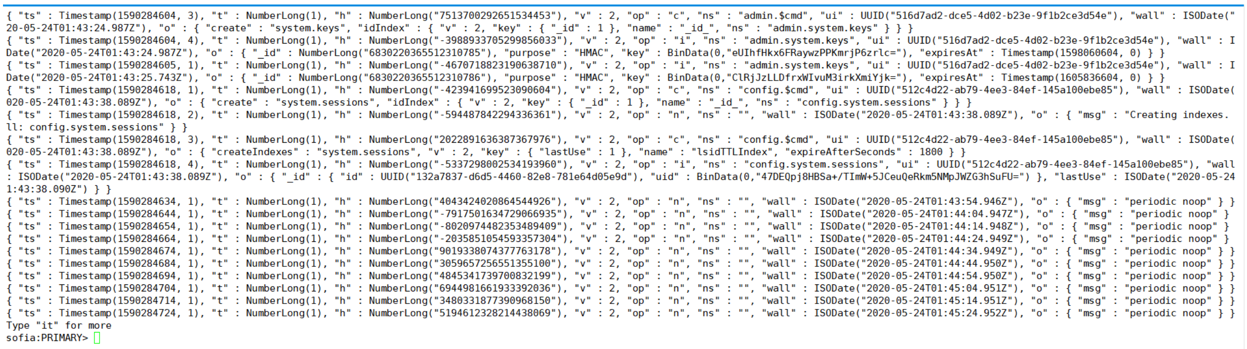
　　票数高者获胜，priority 是优先权0-1000值，相当于额外增加0-1000的票数。

　　选举结果:票数高者获胜;若票数相同，数据新者获胜

```
[root@localhost ~]# mongo
sofia:SECONDARY> rs.status()
{
        ″set″ : ″sofia″,
        ″date″ : ISODate(″2020-05-24T02:45:30.356Z″),
        ″myState″ : 2,
        ″term″ : NumberLong(5),
        ″syncingTo″ : ″127.0.0.1:27017″,
        ″syncSourceHost″ : ″127.0.0.1:27017″,
        ″syncSourceId″ : 0,
        ″heartbeatIntervalMillis″ : NumberLong(2000),
        ″optimes″ : {
                ″lastCommittedOpTime″ : {
                        ″ts″ : Timestamp(1590288324, 1),
                        ″t″ : NumberLong(5)
                },
                ″readConcernMajorityOpTime″ : {
                        ″ts″ : Timestamp(1590288324, 1),
                        ″t″ : NumberLong(5)
                },
                ″appliedOpTime″ : {
                        ″ts″ : Timestamp(1590288324, 1),
                        ″t″ : NumberLong(5)
                },
                ″durableOpTime″ : {
                        ″ts″ : Timestamp(1590288324, 1),
                        ″t″ : NumberLong(5)
                }
        },
        ″members″ : [
                {
                        ″_id″ : 0,
                        ″name″ : ″127.0.0.1:27017″,
                        ″health″ : 1,
                        ″state″ : 1,
                        ″stateStr″ : ″PRIMARY″,
                        ″uptime″ : 1641,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590288324, 1),
                                ″t″ : NumberLong(5)
                        },
                        ″optimeDurable″ : {
                                ″ts″ : Timestamp(1590288324, 1),
                                ″t″ : NumberLong(5)
                        },
                        ″optimeDate″ : ISODate(″2020-05-24T02:45:24Z″),
                        ″optimeDurableDate″ : ISODate(″2020-05-24T02:45:24Z″),
                        ″lastHeartbeat″ : ISODate(″2020-05-24T02:45:28.615Z″),
                        ″lastHeartbeatRecv″    :    ISODate(″2020-05-
24T02:45:28.996Z″),
                        ″pingMs″ : NumberLong(0),
                        ″lastHeartbeatMessage″ : ″″,
                        ″syncingTo″ : ″″,
                        ″syncSourceHost″ : ″″,
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "electionTime" : Timestamp(1590288292, 1),
                        "electionDate" : ISODate("2020-05-24T02:44:52Z"),
                        "configVersion" : 3
                },
                {
                        "_id" : 1,
                        "name" : "127.0.0.1:27018",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 19,
                        "optime" : {
                                "ts" : Timestamp(1590288324, 1),
                                "t" : NumberLong(5)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1590288324, 1),
                                "t" : NumberLong(5)
                        },
                        "optimeDate" : ISODate("2020-05-24T02:45:24Z"),
                        "optimeDurableDate" : ISODate("2020-05-24T02:45:24Z"),
                        "lastHeartbeat" : ISODate("2020-05-24T02:45:28.648Z"),
                        "lastHeartbeatRecv"     :     ISODate("2020-05-
24T02:45:29.115Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "127.0.0.1:27019",
                        "syncSourceHost" : "127.0.0.1:27019",
                        "syncSourceId" : 2,
                        "infoMessage" : "",
                        "configVersion" : 3
                },
                {
```

```
				"_id" : 2,
				"name" : "127.0.0.1:27019",
				"health" : 1,
				"state" : 2,
				"stateStr" : "SECONDARY",
				"uptime" : 1642,
				"optime" : {
					"ts" : Timestamp(1590288324, 1),
					"t" : NumberLong(5)
				},
				"optimeDate" : ISODate("2020-05-24T02:45:24Z"),
				"syncingTo" : "127.0.0.1:27017",
				"syncSourceHost" : "127.0.0.1:27017",
				"syncSourceId" : 0,
				"infoMessage" : "",
				"configVersion" : 3,
				"self" : true,
				"lastHeartbeatMessage" : ""
			}
		],
		"ok" : 1,
		"operationTime" : Timestamp(1590288324, 1),
		"$clusterTime" : {
			"clusterTime" : Timestamp(1590288324, 1),
			"signature" : {
				"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
				"keyId" : NumberLong(0)
			}
		}
}
[root@localhost ~]# mongo
sofia:PRIMARY> show dbs
admin   0.078GB
config  0.078GB
local   4.076GB
```

```
sofia:PRIMARY> use cloud
switched to db cloud
sofia:PRIMARY> db.t1.insert({"id":1,"name":"Tom"})
WriteResult({ "nInserted" : 1 })
sofia:PRIMARY> db.t1.insert({"id":2,"name":"Bob"})
WriteResult({ "nInserted" : 1 })
sofia:PRIMARY> db.t1.find()
{ "_id" : ObjectId("5ec9e0781927d2f2ef29bf3d"), "id" : 1, "name" : "Tom" }
{ "_id" : ObjectId("5ec9e0871927d2f2ef29bf3e"), "id" : 2, "name" : "Bob" }
sofia:PRIMARY> db.t1.remove({"id":2})
WriteResult({ "nRemoved" : 1 })
sofia:PRIMARY> db.t1.find()
{ "_id" : ObjectId("5ec9e0781927d2f2ef29bf3d"), "id" : 1, "name" : "Tom" }
sofia:PRIMARY> use local
switched to db local
sofia:PRIMARY> show collections
oplog.rs
replset.election
replset.minvalid
replset.oplogTruncateAfterPoint
startup_log
sofia:PRIMARY> db.oplog.rs.find()                    # 记录增删改的操作
```



```
{ "ts" : Timestamp(1590284604, 3), "t" : NumberLong(1), "h" : NumberLong("7513700292651534453"), "v" : 2, "op" : "c", "ns" : "admin.$cmd", "ui" : UUID("516d7ad2-dce5-4d02-b23e-9f1b2ce3d54e"), "wall" : ISODate("
20-05-24T01:43:24.987Z"), "o" : { "create" : "system.keys", "idIndex" : { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_", "ns" : "admin.system.keys" } } }
{ "ts" : Timestamp(1590284604, 4), "t" : NumberLong(1), "h" : NumberLong("-3988933705299856033"), "v" : 2, "op" : "i", "ns" : "admin.system.keys", "ui" : UUID("516d7ad2-dce5-4d02-b23e-9f1b2ce3d54e"), "wall" : I
Date("2020-05-24T01:43:24.987Z"), "o" : { "_id" : NumberLong("6830220365512310785"), "purpose" : "HMAC", "key" : BinData(0,"eUIhfHkx6FRaywzPPKmrjP6zrlc="), "expiresAt" : Timestamp(1598060604, 0) } }
{ "ts" : Timestamp(1590284605, 1), "t" : NumberLong(1), "h" : NumberLong("-4670718823190638710"), "v" : 2, "op" : "i", "ns" : "admin.system.keys", "ui" : UUID("516d7ad2-dce5-4d02-b23e-9f1b2ce3d54e"), "wall" : I
Date("2020-05-24T01:43:25.743Z"), "o" : { "_id" : NumberLong("6830220365512310786"), "purpose" : "HMAC", "key" : BinData(0,"ClRjJzLLDfrxWIvuM3irkXmiYjk="), "expiresAt" : Timestamp(1605836604, 0) } }
{ "ts" : Timestamp(1590284618, 1), "t" : NumberLong(1), "h" : NumberLong("-423941699523090604"), "v" : 2, "op" : "c", "ns" : "config.$cmd", "ui" : UUID("512c4d22-ab79-4ee3-84ef-145a100ebe85"), "wall" : ISODate(
020-05-24T01:43:38.089Z"), "o" : { "create" : "system.sessions", "idIndex" : { "v" : 2, "key" : { "_id" : 1 }, "name" : "_id_", "ns" : "config.system.sessions" } } }
{ "ts" : Timestamp(1590284618, 2), "t" : NumberLong(1), "h" : NumberLong("-5944878422943363б1"), "v" : 2, "op" : "n", "ns" : "", "wall" : ISODate("2020-05-24T01:43:38.089Z"), "o" : { "msg" : "Creating indexes.
ll: config.system.sessions" } }
{ "ts" : Timestamp(1590284618, 3), "t" : NumberLong(1), "h" : NumberLong("2022891636387367976"), "v" : 2, "op" : "c", "ns" : "config.$cmd", "ui" : UUID("512c4d22-ab79-4ee3-84ef-145a100ebe85"), "wall" : ISODate(
020-05-24T01:43:38.089Z"), "o" : { "createIndexes" : "system.sessions", "v" : 2, "key" : { "lastUse" : 1 }, "name" : "lsidTTLIndex", "expireAfterSeconds" : 1800 } }
{ "ts" : Timestamp(1590284618, 4), "t" : NumberLong(1), "h" : NumberLong("-5337298002534193960"), "v" : 2, "op" : "i", "ns" : "config.system.sessions", "ui" : UUID("512c4d22-ab79-4ee3-84ef-145a100ebe85"), "wall
: ISODate("2020-05-24T01:43:38.089Z"), "o" : { "_id" : { "id" : UUID("132a7837-d6d5-4460-82e8-781e64d05e9d"), "uid" : BinData(0,"47DEQpj8HBSa+/TImW+5JCeuQeRkm5NMpJWZG3hSuFU=") }, "lastUse" : ISODate("2020-05-24
1:43:38.090Z") } }
{ "ts" : Timestamp(1590284634, 1), "t" : NumberLong(1), "h" : NumberLong("4043424020864544926"), "v" : 2, "op" : "n", "ns" : "", "wall" : ISODate("2020-05-24T01:43:54.946Z"), "o" : { "msg" : "periodic noop" } }
{ "ts" : Timestamp(1590284644, 1), "t" : NumberLong(1), "h" : NumberLong("-7917501634729066935"), "v" : 2, "op" : "n", "ns" : "", "wall" : ISODate("2020-05-24T01:44:04.947Z"), "o" : { "msg" : "periodic noop" }
{ "ts" : Timestamp(1590284654, 1), "t" : NumberLong(1), "h" : NumberLong("-8020974482353489409"), "v" : 2, "op" : "n", "ns" : "", "wall" : ISODate("2020-05-24T01:44:14.948Z"), "o" : { "msg" : "periodic noop" }
{ "ts" : Timestamp(1590284664, 1), "t" : NumberLong(1), "h" : NumberLong("-2035851054593357304"), "v" : 2, "op" : "n", "ns" : "", "wall" : ISODate("2020-05-24T01:44:24.949Z"), "o" : { "msg" : "periodic noop" }
{ "ts" : Timestamp(1590284674, 1), "t" : NumberLong(1), "h" : NumberLong("9019338074377763178"), "v" : 2, "op" : "n", "ns" : "", "wall" : ISODate("2020-05-24T01:44:34.949Z"), "o" : { "msg" : "periodic noop" } }
{ "ts" : Timestamp(1590284684, 1), "t" : NumberLong(1), "h" : NumberLong("3059657256551355100"), "v" : 2, "op" : "n", "ns" : "", "wall" : ISODate("2020-05-24T01:44:44.950Z"), "o" : { "msg" : "periodic noop" } }
{ "ts" : Timestamp(1590284694, 1), "t" : NumberLong(1), "h" : NumberLong("4845341739700832199"), "v" : 2, "op" : "n", "ns" : "", "wall" : ISODate("2020-05-24T01:44:54.950Z"), "o" : { "msg" : "periodic noop" } }
{ "ts" : Timestamp(1590284704, 1), "t" : NumberLong(1), "h" : NumberLong("6944981661933392036"), "v" : 2, "op" : "n", "ns" : "", "wall" : ISODate("2020-05-24T01:45:04.951Z"), "o" : { "msg" : "periodic noop" } }
{ "ts" : Timestamp(1590284714, 1), "t" : NumberLong(1), "h" : NumberLong("3480331877390968150"), "v" : 2, "op" : "n", "ns" : "", "wall" : ISODate("2020-05-24T01:45:14.951Z"), "o" : { "msg" : "periodic noop" } }
{ "ts" : Timestamp(1590284724, 1), "t" : NumberLong(1), "h" : NumberLong("5194612328214438069"), "v" : 2, "op" : "n", "ns" : "", "wall" : ISODate("2020-05-24T01:45:24.952Z"), "o" : { "msg" : "periodic noop" } }
Type "it" for more
sofia:PRIMARY> 
```

重新定义集群，将27020节点设置为仲裁节点

```
[root@localhost ~]# /etc/init.d/mongodb mongodb1 stop
killing process with pid: 11802
[root@localhost ~]# /etc/init.d/mongodb mongodb2 stop
killing process with pid: 13016
[root@localhost ~]# /etc/init.d/mongodb mongodb3 stop
killing process with pid: 12406
```

```
[root@localhost ~]# /etc/init.d/mongodb mongodb4 stop
killing process with pid: 11083
```

都停的目的是为了让它原来的集群销毁

```
[root@localhost ~]# /etc/init.d/mongodb mongodb1 start
about to fork child process, waiting until server is ready for connections.
forked process: 13301
child process started successfully, parent exiting
[root@localhost ~]# /etc/init.d/mongodb mongodb2 start
about to fork child process, waiting until server is ready for connections.
forked process: 13354
child process started successfully, parent exiting
[root@localhost ~]# /etc/init.d/mongodb mongodb3 start
about to fork child process, waiting until server is ready for connections.
forked process: 13422
child process started successfully, parent exiting
[root@localhost ~]# /etc/init.d/mongodb mongodb4 start
about to fork child process, waiting until server is ready for connections.
forked process: 13481
child process started successfully, parent exiting
[root@localhost ~]# mongo
sofia:PRIMARY> cfg={"_id":"sofia","protocolVersion":1,"members":
[{"_id":0,"host":"127.0.0.1:27017","priority":100},{"_id":1,"host":"
127.0.0.1:27018","priority":100},
{"_id":2,"host":"127.0.0.1:27019","priority":0},
{"_id":3,"host":"127.0.0.1:27020","arbiterOnly":true}]}
# 两个标准节点，一个被动节点，一个仲裁节点
{
        "_id" : "sofia",
        "protocolVersion" : 1,
        "members" : [
                {
                        "_id" : 0,
                        "host" : "127.0.0.1:27017",
                        "priority" : 100
                },
                {
                        ″_id″ : 1,
                        ″host″ : ″127.0.0.1:27018″,
                        ″priority″ : 100
                },
                {
                        ″_id″ : 2,
                        ″host″ : ″127.0.0.1:27019″,
                        ″priority″ : 0
                },
                {
                        ″_id″ : 3,
                        ″host″ : ″127.0.0.1:27020″,
                        ″arbiterOnly″ : true
                }
        ]
}
```

此时，集群还未生效需要进行初始化（重新加载配置）

```
sofia:PRIMARY> rs.reconfig(cfg)
{
        ″ok″ : 1,
        ″operationTime″ : Timestamp(1590289800, 1),
        ″$clusterTime″ : {
                ″clusterTime″ : Timestamp(1590289800, 1),
                ″signature″ : {
                        ″hash″ : BinData(0,″AAAAAAAAAAAAAAAAAAAAAAAAAAA=″),
                        ″keyId″ : NumberLong(0)
                }
        }
}
sofia:SECONDARY> rs.status()
{
        ″set″ : ″sofia″,
        ″date″ : ISODate(″2020-05-24T03:14:34.427Z″),
        ″myState″ : 2,
        ″term″ : NumberLong(8),
        ″syncingTo″ : ″127.0.0.1:27018″,
        ″syncSourceHost″ : ″127.0.0.1:27018″,
        ″syncSourceId″ : 1,
        ″heartbeatIntervalMillis″ : NumberLong(2000),
        ″optimes″ : {
                ″lastCommittedOpTime″ : {
                        ″ts″ : Timestamp(1590290069, 1),
                        ″t″ : NumberLong(8)
                },
                ″readConcernMajorityOpTime″ : {
                        ″ts″ : Timestamp(1590290069, 1),
                        ″t″ : NumberLong(8)
                },
                ″appliedOpTime″ : {
                        ″ts″ : Timestamp(1590290069, 1),
                        ″t″ : NumberLong(8)
                },
                ″durableOpTime″ : {
                        ″ts″ : Timestamp(1590290069, 1),
                        ″t″ : NumberLong(8)
                }
        },
        ″members″ : [
                {
                        ″_id″ : 0,
                        ″name″ : ″127.0.0.1:27017″,
                        ″health″ : 1,
                        ″state″ : 2,
                        ″stateStr″ : ″SECONDARY″,
                        ″uptime″ : 40,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590290069, 1),
                                ″t″ : NumberLong(8)
                        },
                        "optimeDate" : ISODate("2020-05-24T03:14:29Z"),
                        "syncingTo" : "127.0.0.1:27018",
                        "syncSourceHost" : "127.0.0.1:27018",
                        "syncSourceId" : 1,
                        "infoMessage" : "",
                        "configVersion" : 4,
                        "self" : true,
                        "lastHeartbeatMessage" : ""
                },
                {
                        "_id" : 1,
                        "name" : "127.0.0.1:27018",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 27,
                        "optime" : {
                                "ts" : Timestamp(1590290069, 1),
                                "t" : NumberLong(8)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1590290069, 1),
                                "t" : NumberLong(8)
                        },
                        "optimeDate" : ISODate("2020-05-24T03:14:29Z"),
                        "optimeDurableDate" : ISODate("2020-05-24T03:14:29Z"),
                        "lastHeartbeat" : ISODate("2020-05-24T03:14:33.922Z"),
                        "lastHeartbeatRecv" : ISODate("2020-05-24T03:14:34.060Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "electionTime" : Timestamp(1590290058, 1),
                        "electionDate" : ISODate("2020-05-24T03:14:18Z"),
                        "configVersion" : 4
                },
                {
                        "_id" : 2,
                        "name" : "127.0.0.1:27019",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 23,
                        "optime" : {
                                "ts" : Timestamp(1590290069, 1),
                                "t" : NumberLong(8)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1590290069, 1),
                                "t" : NumberLong(8)
                        },
                        "optimeDate" : ISODate("2020-05-24T03:14:29Z"),
                        "optimeDurableDate" : ISODate("2020-05-24T03:14:29Z"),
                        "lastHeartbeat" : ISODate("2020-05-24T03:14:33.922Z"),
                        "lastHeartbeatRecv"    :    ISODate("2020-05-
24T03:14:33.922Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "127.0.0.1:27018",
                        "syncSourceHost" : "127.0.0.1:27018",
                        "syncSourceId" : 1,
                        "infoMessage" : "",
                        "configVersion" : 4
                },
                {
                        "_id" : 3,
                        "name" : "127.0.0.1:27020",
                        ″health″ : 1,
                        ″state″ : 7,
                        ″stateStr″ : ″ARBITER″,
                        ″uptime″ : 17,
                        ″lastHeartbeat″ : ISODate(″2020-05-24T03:14:33.922Z″),
                        ″lastHeartbeatRecv″ : ISODate(″2020-05-24T03:14:32.546Z″),
                        ″pingMs″ : NumberLong(0),
                        ″lastHeartbeatMessage″ : ″″,
                        ″syncingTo″ : ″″,
                        ″syncSourceHost″ : ″″,
                        ″syncSourceId″ : -1,
                        ″infoMessage″ : ″″,
                        ″configVersion″ : 4
                }
        ],
        ″ok″ : 1,
        ″operationTime″ : Timestamp(1590290069, 1),
        ″$clusterTime″ : {
                ″clusterTime″ : Timestamp(1590290069, 1),
                ″signature″ : {
                        ″hash″ : BinData(0,″AAAAAAAAAAAAAAAAAAAAAAAAAAA=″),
                        ″keyId″ : NumberLong(0)
                }
        }
}
sofia:SECONDARY> rs.isMaster()
{
        ″hosts″ : [
                ″127.0.0.1:27017″,
                ″127.0.0.1:27018″
        ],
        ″passives″ : [
                ″127.0.0.1:27019″
        ],
        "arbiters" : [
                "127.0.0.1:27020"
        ],
        "setName" : "sofia",
        "setVersion" : 4,
        "ismaster" : false,
        "secondary" : true,
        "primary" : "127.0.0.1:27018",
        "me" : "127.0.0.1:27017",
        "lastWrite" : {
                "opTime" : {
                        "ts" : Timestamp(1590290189, 1),
                        "t" : NumberLong(8)
                },
                "lastWriteDate" : ISODate("2020-05-24T03:16:29Z"),
                "majorityOpTime" : {
                        "ts" : Timestamp(1590290189, 1),
                        "t" : NumberLong(8)
                },
                "majorityWriteDate" : ISODate("2020-05-24T03:16:29Z")
        },
        "maxBsonObjectSize" : 16777216,
        "maxMessageSizeBytes" : 48000000,
        "maxWriteBatchSize" : 100000,
        "localTime" : ISODate("2020-05-24T03:16:38.004Z"),
        "logicalSessionTimeoutMinutes" : 30,
        "minWireVersion" : 0,
        "maxWireVersion" : 7,
        "readOnly" : false,
        "ok" : 1,
        "operationTime" : Timestamp(1590290189, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590290189, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        ″keyId″ : NumberLong(0)
                }
        }
}
```

现在的主节点是27018

模拟主节点出现故障

```
[root@localhost ~]# /etc/init.d/mongodb mongodb2 stop
killing process with pid: 13905
[root@localhost ~]# mongo --port 2017
sofia:PRIMARY>                              # 27017变为主节点
```

被动节点不可能成为活跃节点

```
[root@localhost ~]# /etc/init.d/mongodb mongodb2 stop
killing process with pid: 14222
[root@localhost ~]# /etc/init.d/mongodb mongodb1 stop
killing process with pid: 13847
[root@localhost ~]# mongo
MongoDB shell version v4.0.6
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
2020-05-24T11:24:16.618+0800 E QUERY    [js] Error: couldn't connect to server 127.0.0.1:27017, connection attempt failed: SocketExc
eption: Error connecting to 127.0.0.1:27017 :: caused by :: Connection refused :
connect@src/mongo/shell/mongo.js:343:13
@(connect):1:6
exception: connect failed
[root@localhost ~]# mongo --port 27019
sofia:SECONDARY> rs.status()
{
        ″set″ : ″sofia″,
        ″date″ : ISODate(″2020-05-24T03:24:50.564Z″),
        ″myState″ : 2,
        ″term″ : NumberLong(9),
        ″syncingTo″ : ″″,
        ″syncSourceHost″ : ″″,
"syncSourceId" : -1,
"heartbeatIntervalMillis" : NumberLong(2000),
"optimes" : {
        "lastCommittedOpTime" : {
                "ts" : Timestamp(1590290628, 1),
                "t" : NumberLong(9)
        },
        "readConcernMajorityOpTime" : {
                "ts" : Timestamp(1590290628, 1),
                "t" : NumberLong(9)
        },
        "appliedOpTime" : {
                "ts" : Timestamp(1590290648, 1),
                "t" : NumberLong(9)
        },
        "durableOpTime" : {
                "ts" : Timestamp(1590290648, 1),
                "t" : NumberLong(9)
        }
},
"members" : [
        {
                "_id" : 0,
                "name" : "127.0.0.1:27017",
                "health" : 0,
                "state" : 8,
                "stateStr" : "(not reachable/healthy)",
                "uptime" : 0,
                "optime" : {
                        "ts" : Timestamp(0, 0),
                        "t" : NumberLong(-1)
                },
                "optimeDurable" : {
                        "ts" : Timestamp(0, 0),
                        "t" : NumberLong(-1)
                        },
                        ″optimeDate″ : ISODate(″1970-01-01T00:00:00Z″),
                        ″optimeDurableDate″ : ISODate(″1970-01-01T00:00:00Z″),
                        ″lastHeartbeat″ : ISODate(″2020-05-24T03:24:50.069Z″),
                        ″lastHeartbeatRecv″ : ISODate(″2020-05-
24T03:24:08.761Z″),
                        ″pingMs″ : NumberLong(0),
                        ″lastHeartbeatMessage″ : ″Error connecting to
127.0.0.1:27017 :: caused by :: Connection refused″,
                        ″syncingTo″ : ″″,
                        ″syncSourceHost″ : ″″,
                        ″syncSourceId″ : -1,
                        ″infoMessage″ : ″″,
                        ″configVersion″ : -1
                },
                {
                        ″_id″ : 1,
                        ″name″ : ″127.0.0.1:27018″,
                        ″health″ : 0,
                        ″state″ : 8,
                        ″stateStr″ : ″(not reachable/healthy)″,
                        ″uptime″ : 0,
                        ″optime″ : {
                                ″ts″ : Timestamp(0, 0),
                                ″t″ : NumberLong(-1)
                        },
                        ″optimeDurable″ : {
                                ″ts″ : Timestamp(0, 0),
                                ″t″ : NumberLong(-1)
                        },
                        ″optimeDate″ : ISODate(″1970-01-01T00:00:00Z″),
                        ″optimeDurableDate″ : ISODate(″1970-01-01T00:00:00Z″),
                        ″lastHeartbeat″ : ISODate(″2020-05-24T03:24:50.069Z″),
                        ″lastHeartbeatRecv″ : ISODate(″2020-05-
24T03:23:52.543Z″),
                        ″pingMs″ : NumberLong(0),
                        ″lastHeartbeatMessage″ : ″Error connecting to 127.0.0.1:27018 :: caused by :: Connection refused″,
                        ″syncingTo″ : ″″,
                        ″syncSourceHost″ : ″″,
                        ″syncSourceId″ : -1,
                        ″infoMessage″ : ″″,
                        ″configVersion″ : -1
                },
                {
                        ″_id″ : 2,
                        ″name″ : ″127.0.0.1:27019″,
                        ″health″ : 1,
                        ″state″ : 2,
                        ″stateStr″ : ″SECONDARY″,
                        ″uptime″ : 640,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590290648, 1),
                                ″t″ : NumberLong(9)
                        },
                        ″optimeDate″ : ISODate(″2020-05-24T03:24:08Z″),
                        ″syncingTo″ : ″″,
                        ″syncSourceHost″ : ″″,
                        ″syncSourceId″ : -1,
                        ″infoMessage″ : ″could not find member to sync from″,
                        ″configVersion″ : 4,
                        ″self″ : true,
                        ″lastHeartbeatMessage″ : ″″
                },
                {
                        ″_id″ : 3,
                        ″name″ : ″127.0.0.1:27020″,
                        ″health″ : 1,
                        ″state″ : 7,
                        ″stateStr″ : ″ARBITER″,
                        "uptime" : 633,
                        "lastHeartbeat" : ISODate("2020-05-24T03:24:50.534Z"),
                        "lastHeartbeatRecv" : ISODate("2020-05-24T03:24:48.742Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "configVersion" : 4
                }
        ],
        "ok" : 1,
        "operationTime" : Timestamp(1590290648, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590290648, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
[root@localhost ~]# /etc/init.d/mongodb mongodb1 start
about to fork child process, waiting until server is ready for connections.
forked process: 14371
child process started successfully, parent exiting
[root@localhost ~]# /etc/init.d/mongodb mongodb2 start
about to fork child process, waiting until server is ready for connections.
forked process: 14438
child process started successfully, parent exiting
[root@localhost ~]# mongo
sofia:PRIMARY>  rs.status()
{
        "set" : "sofia",
        "date" : ISODate("2020-05-24T03:27:11.387Z"),
        "myState" : 1,
        "term" : NumberLong(10),
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1590290825, 1),
                        "t" : NumberLong(10)
                },
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1590290825, 1),
                        "t" : NumberLong(10)
                },
                "appliedOpTime" : {
                        "ts" : Timestamp(1590290825, 1),
                        "t" : NumberLong(10)
                },
                "durableOpTime" : {
                        "ts" : Timestamp(1590290825, 1),
                        "t" : NumberLong(10)
                }
        },
        "members" : [
                {
                        "_id" : 0,
                        "name" : "127.0.0.1:27017",
                        "health" : 1,
                        "state" : 1,
                        "stateStr" : "PRIMARY",
                        "uptime" : 19,
                        "optime" : {
                                "ts" : Timestamp(1590290825, 1),
                                "t" : NumberLong(10)
                        },
                        "optimeDate" : ISODate("2020-05-24T03:27:05Z"),
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "could not find member to sync from",
                        "electionTime" : Timestamp(1590290823, 1),
                        "electionDate" : ISODate("2020-05-24T03:27:03Z"),
                        "configVersion" : 4,
                        "self" : true,
                        "lastHeartbeatMessage" : ""
                },
                {
                        "_id" : 1,
                        "name" : "127.0.0.1:27018",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 12,
                        "optime" : {
                                "ts" : Timestamp(1590290825, 1),
                                "t" : NumberLong(10)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1590290825, 1),
                                "t" : NumberLong(10)
                        },
                        "optimeDate" : ISODate("2020-05-24T03:27:05Z"),
                        "optimeDurableDate" : ISODate("2020-05-24T03:27:05Z"),
                        "lastHeartbeat" : ISODate("2020-05-24T03:27:09.870Z"),
                        "lastHeartbeatRecv"     :     ISODate("2020-05-
24T03:27:10.315Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "127.0.0.1:27017",
                        "syncSourceHost" : "127.0.0.1:27017",
                        "syncSourceId" : 0,
                        "infoMessage" : "",
                        "configVersion" : 4
                },
                {
                        "_id" : 2,
                        "name" : "127.0.0.1:27019",
                        "health" : 1,
                        "state" : 2,
                        "stateStr" : "SECONDARY",
                        "uptime" : 18,
                        "optime" : {
                                "ts" : Timestamp(1590290825, 1),
                                "t" : NumberLong(10)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1590290825, 1),
                                "t" : NumberLong(10)
                        },
                        "optimeDate" : ISODate("2020-05-24T03:27:05Z"),
                        "optimeDurableDate" : ISODate("2020-05-24T03:27:05Z"),
                        "lastHeartbeat" : ISODate("2020-05-24T03:27:09.866Z"),
                        "lastHeartbeatRecv"    :    ISODate("2020-05-
24T03:27:10.055Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "127.0.0.1:27017",
                        "syncSourceHost" : "127.0.0.1:27017",
                        "syncSourceId" : 0,
                        "infoMessage" : "",
                        "configVersion" : 4
                },
                {
                        "_id" : 3,
                        "name" : "127.0.0.1:27020",
                        "health" : 1,
                        "state" : 7,
                        "stateStr" : "ARBITER",
                        "uptime" : 18,
                        "lastHeartbeat" : ISODate("2020-05-24T03:27:09.866Z"),
                        "lastHeartbeatRecv" : ISODate("2020-05-24T03:27:10.880Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                        "configVersion" : 4
                }
        ],
        "ok" : 1,
        "operationTime" : Timestamp(1590290825, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590290825, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

注意：两个标准节点，谁先启动谁就拿到主节点，也即活跃节点

查看配置：

```
sofia:PRIMARY> rs.conf()
{
        "_id" : "sofia",
        "version" : 4,
        "protocolVersion" : NumberLong(1),
        "writeConcernMajorityJournalDefault" : true,
        "members" : [
                {
                        "_id" : 0,
                        "host" : "127.0.0.1:27017",
                        "arbiterOnly" : false,
                        "buildIndexes" : true,
                        "hidden" : false,
                        "priority" : 100,
                        "tags" : {

                        },
                        "slaveDelay" : NumberLong(0),
                        "votes" : 1
                },
                {
                        "_id" : 1,
                        "host" : "127.0.0.1:27018",
                        "arbiterOnly" : false,
                        "buildIndexes" : true,
                        "hidden" : false,
                        "priority" : 100,
                        "tags" : {

                        },
                        "slaveDelay" : NumberLong(0),
                        "votes" : 1
                },
                {
                        "_id" : 2,
                        "host" : "127.0.0.1:27019",
                        "arbiterOnly" : false,
                        "buildIndexes" : true,
                        "hidden" : false,
                        "priority" : 0,
                        ″tags″ : {

                        },
                        ″slaveDelay″ : NumberLong(0),
                        ″votes″ : 1
                },
                {
                        ″_id″ : 3,
                        ″host″ : ″127.0.0.1:27020″,
                        ″arbiterOnly″ : true,
                        ″buildIndexes″ : true,
                        ″hidden″ : false,
                        ″priority″ : 0,
                        ″tags″ : {

                        },
                        ″slaveDelay″ : NumberLong(0),
                        ″votes″ : 1
                }
        ],
        ″settings″ : {
                ″chainingAllowed″ : true,
                ″heartbeatIntervalMillis″ : 2000,
                ″heartbeatTimeoutSecs″ : 10,
                ″electionTimeoutMillis″ : 10000,
                ″catchUpTimeoutMillis″ : -1,
                ″catchUpTakeoverDelayMillis″ : 30000,
                ″getLastErrorModes″ : {

                },
                ″getLastErrorDefaults″ : {
                        ″w″ : 1,
                        ″wtimeout″ : 0
                },
                ″replicaSetId″ : ObjectId(″5ec9d1307e5c719a8c0878e4″)
        }
}

sofia:PRIMARY> use local
switched to db local
sofia:PRIMARY> show tables
oplog.rs
replset.election
replset.minvalid
replset.oplogTruncateAfterPoint
startup_log
sofia:PRIMARY> db.system.replset.findOne()             # 查看集群动态
等同于：sofia:PRIMARY> rs.status()
{
        ″_id″ : ″sofia″,
        ″version″ : 4,
        ″protocolVersion″ : NumberLong(1),
        ″writeConcernMajorityJournalDefault″ : true,
        ″members″ : [
                {
                        ″_id″ : 0,
                        ″host″ : ″127.0.0.1:27017″,
                        ″arbiterOnly″ : false,
                        ″buildIndexes″ : true,
                        ″hidden″ : false,
                        ″priority″ : 100,
                        ″tags″ : {

                        },
                        ″slaveDelay″ : NumberLong(0),
                        ″votes″ : 1
                },
                {
                        ″_id″ : 1,
                        ″host″ : ″127.0.0.1:27018″,
                        ″arbiterOnly″ : false,
                        ″buildIndexes″ : true,
                        ″hidden″ : false,
                        ″priority″ : 100,
                        ″tags″ : {

                        },
                        ″slaveDelay″ : NumberLong(0),
                        ″votes″ : 1
                },
                {
                        ″_id″ : 2,
                        ″host″ : ″127.0.0.1:27019″,
                        ″arbiterOnly″ : false,
                        ″buildIndexes″ : true,
                        ″hidden″ : false,
                        ″priority″ : 0,
                        ″tags″ : {

                        },
                        ″slaveDelay″ : NumberLong(0),
                        ″votes″ : 1
                },
                {
                        ″_id″ : 3,
                        ″host″ : ″127.0.0.1:27020″,
                        ″arbiterOnly″ : true,
                        ″buildIndexes″ : true,
                        ″hidden″ : false,
                        ″priority″ : 0,
                        ″tags″ : {

                        },
                        ″slaveDelay″ : NumberLong(0),
                        ″votes″ : 1
                }
        ],
        "settings" : {
                "chainingAllowed" : true,
                "heartbeatIntervalMillis" : 2000,
                "heartbeatTimeoutSecs" : 10,
                "electionTimeoutMillis" : 10000,
                "catchUpTimeoutMillis" : -1,
                "catchUpTakeoverDelayMillis" : 30000,
                "getLastErrorModes" : {

                },
                "getLastErrorDefaults" : {
                        "w" : 1,
                        "wtimeout" : 0
                },
                "replicaSetId" : ObjectId("5ec9d1307e5c719a8c0878e4")
        }
}
```

配置从节点可以读取数据

```
[root@localhost ~]# mongo --port 27018
sofia:SECONDARY> show dbs
2020-05-24T11:36:27.733+0800 E QUERY    [js] Error: listDatabases failed:{
        "operationTime" : Timestamp(1590291385, 1),
        "ok" : 0,
        "errmsg" : "not master and slaveOk=false",
        "code" : 13435,
        "codeName" : "NotMasterNoSlaveOk",
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590291385, 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:139:1
shellHelper.show@src/mongo/shell/utils.js:882:13
shellHelper@src/mongo/shell/utils.js:766:15
@(shellhelp2):1:1


sofia:SECONDARY> rs.slaveOk()                # 使从节点可以读取数据
sofia:SECONDARY> show dbs
admin   0.078GB
cloud   0.078GB
config  0.078GB
local   2.077GB
sofia:SECONDARY> rs.printSlaveReplicationInfo()             # 查看从的动态
source: 127.0.0.1:27018
        syncedTo: Sun May 24 2020 11:38:35 GMT+0800 (CST)
        0 secs (0 hrs) behind the primary
source: 127.0.0.1:27019
        syncedTo: Sun May 24 2020 11:38:35 GMT+0800 (CST)
        0 secs (0 hrs) behind the primary
sofia:SECONDARY> rs.printReplicationInfo()
configured oplog size:   2071.980453491211MB           # oplog 的大小，可手动修改
log length start to end: 7033secs (1.95hrs)
oplog first event time:  Sun May 24 2020 09:43:12 GMT+0800 (CST)
oplog last event time:   Sun May 24 2020 11:40:25 GMT+0800 (CST)
now:                     Sun May 24 2020 11:40:28 GMT+0800 (CST)
```

4.5、 复制集管理

配置oplog文件大小

生产常见中oplog文件默认大小是不能满足频繁的更新业务需求的，在配置复制集启动时，就应该针对oplog有大小预计，旧版本修改oplog大小必须重启主数据库，但是新版本不需要。

```
[root@localhost ~]# mongo
```

```
sofia:PRIMARY> use local
switched to db local
sofia:PRIMARY> db.oplog.rs.stats()
{
        "ns" : "local.oplog.rs",
        "size" : 85348,
        "count" : 731,
        "avgObjSize" : 116,
        "numExtents" : 2,
        "storageSize" : 2391474160,
        "lastExtentSize" : 245047296,
        "paddingFactor" : 1,
        "paddingFactorNote" : "paddingFactor is unused and unmaintained in 3.0.
It remains hard coded to 1.0 for compatibility only.
",      "userFlags" : 1,
        "capped" : true,
        "max" : NumberLong("9223372036854775807"),
        "maxSize" : 2391474160,
        "nindexes" : 0,
        "totalIndexSize" : 0,
        "indexSizes" : {

        },
        "ok" : 1,
        "operationTime" : Timestamp(1590292095, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590292095, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
```

修改oplog的大小 100M

```
sofia:PRIMARY> db.runCommand({"convertToCapped":"oplog.rs","size":102400000})
```

```
{
        "ok" : 1,
        "operationTime" : Timestamp(1590292275, 1),
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590292275, 1),
                "signature" : {
                        "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),
                        "keyId" : NumberLong(0)
                }
        }
}
sofia:PRIMARY> rs.printReplicationInfo()              # 这里重启了一遍服务
configured oplog size:   97.65625MB
log length start to end: 7857secs (2.18hrs)
oplog first event time:  Sun May 24 2020 09:43:12 GMT+0800 (CST)
oplog last event time:   Sun May 24 2020 11:54:09 GMT+0800 (CST)
now:                     Sun May 24 2020 11:54:14 GMT+0800 (CST)
```

部署认证的复制

复制集群以密钥认证，其他登录需要密码认证

```
sofia:PRIMARY> use admin
switched to db admin
sofia:PRIMARY> db.createUser({"user":"root","pwd":"123456","roles":["root"]})
Successfully added user: { "user" : "root", "roles" : [ "root" ] }
[root@localhost ~]# cat << END >> /usr/local/mongodb/conf/mongodb1.conf
> clusterAuthMode=keyFile
> keyFile=/usr/local/mongodb/conf/sofiakey
> END
[root@localhost ~]# cat << END >> /usr/local/mongodb/conf/mongodb2.conf
> clusterAuthMode=keyFile
> keyFile=/usr/local/mongodb/conf/sofiakey
> END
[root@localhost ~]# cat << END >> /usr/local/mongodb/conf/mongodb3.conf
> clusterAuthMode=keyFile
> keyFile=/usr/local/mongodb/conf/sofiakey
> END
```

```
[root@localhost ~]# cat << END >> /usr/local/mongodb/conf/mongodb4.conf
> clusterAuthMode=keyFile
> keyFile=/usr/local/mongodb/conf/sofiakey
> END
[root@localhost ~]# echo "123456 key" > /usr/local/mongodb/conf/sofiakey
[root@localhost ~]# chmod 600 /usr/local/mongodb/conf/sofiakey
[root@localhost ~]# /etc/init.d/mongodb mongodb1 restart
killing process with pid: 14829
about to fork child process, waiting until server is ready for connections.
forked process: 15308
child process started successfully, parent exiting
[root@localhost ~]# /etc/init.d/mongodb mongodb2 restart
killing process with pid: 14912
about to fork child process, waiting until server is ready for connections.
forked process: 15432
child process started successfully, parent exiting
[root@localhost ~]# /etc/init.d/mongodb mongodb3 restart
killing process with pid: 14992
about to fork child process, waiting until server is ready for connections.
forked process: 15532
child process started successfully, parent exiting
[root@localhost ~]# /etc/init.d/mongodb mongodb4 restart
killing process with pid: 15072
about to fork child process, waiting until server is ready for connections.
forked process: 15624
child process started successfully, parent exiting
[root@localhost ~]# mongo
MongoDB shell version v4.0.6
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("a081aad2-e9cc-406a-b693-3453dd088849")
}
MongoDB server version: 4.0.6
sofia:PRIMARY> rs.status()                  # 此时不能查看
{
        "operationTime" : Timestamp(1590293405, 1),
        "ok" : 0,
        "errmsg" : "command replSetGetStatus requires authentication",
        "code" : 13,
        "codeName" : "Unauthorized",
        "$clusterTime" : {
                "clusterTime" : Timestamp(1590293405, 1),
                "signature" : {
                        "hash" : BinData(0,"TVqBCh09K9gbrmPIiJBR7Jz0N8I="),
                        "keyId" : NumberLong("6830220365512310785")
                }
        }
}
sofia:PRIMARY> use admin
switched to db admin
sofia:PRIMARY> db.auth("root","123456")                          # 验证通过才可以查
看
1
sofia:PRIMARY> rs.status()
{
        "set" : "sofia",
        "date" : ISODate("2020-05-24T04:12:42.612Z"),
        "myState" : 1,
        "term" : NumberLong(14),
        "syncingTo" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "heartbeatIntervalMillis" : NumberLong(2000),
        "optimes" : {
                "lastCommittedOpTime" : {
                        "ts" : Timestamp(1590293555, 1),
                        "t" : NumberLong(14)
                },
                "readConcernMajorityOpTime" : {
                        "ts" : Timestamp(1590293555, 1),
                        "t" : NumberLong(14)
                },
                ″appliedOpTime″ : {
                        ″ts″ : Timestamp(1590293555, 1),
                        ″t″ : NumberLong(14)
                },
                ″durableOpTime″ : {
                        ″ts″ : Timestamp(1590293555, 1),
                        ″t″ : NumberLong(14)
                }
        },
        ″members″ : [
                {
                        ″_id″ : 0,
                        ″name″ : ″127.0.0.1:27017″,
                        ″health″ : 1,
                        ″state″ : 1,
                        ″stateStr″ : ″PRIMARY″,
                        ″uptime″ : 350,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590293555, 1),
                                ″t″ : NumberLong(14)
                        },
                        ″optimeDate″ : ISODate(″2020-05-24T04:12:35Z″),
                        ″syncingTo″ : ″″,
                        ″syncSourceHost″ : ″″,
                        ″syncSourceId″ : -1,
                        ″infoMessage″ : ″″,
                        ″electionTime″ : Timestamp(1590293233, 1),
                        ″electionDate″ : ISODate(″2020-05-24T04:07:13Z″),
                        ″configVersion″ : 4,
                        ″self″ : true,
                        ″lastHeartbeatMessage″ : ″″
                },
                {
                        ″_id″ : 1,
                        ″name″ : ″127.0.0.1:27018″,
                        ″health″ : 1,
                        ″state″ : 2,
                        ″stateStr″ : ″SECONDARY″,
                        ″uptime″ : 339,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590293555, 1),
                                ″t″ : NumberLong(14)
                        },
                        ″optimeDurable″ : {
                                ″ts″ : Timestamp(1590293555, 1),
                                ″t″ : NumberLong(14)
                        },
                        ″optimeDate″ : ISODate(″2020-05-24T04:12:35Z″),
                        ″optimeDurableDate″ : ISODate(″2020-05-24T04:12:35Z″),
                        ″lastHeartbeat″ : ISODate(″2020-05-24T04:12:41.277Z″),
                        ″lastHeartbeatRecv″    :    ISODate(″2020-05-
24T04:12:41.972Z″),
                        ″pingMs″ : NumberLong(0),
                        ″lastHeartbeatMessage″ : ″″,
                        ″syncingTo″ : ″127.0.0.1:27017″,
                        ″syncSourceHost″ : ″127.0.0.1:27017″,
                        ″syncSourceId″ : 0,
                        ″infoMessage″ : ″″,
                        ″configVersion″ : 4
                },
                {
                        ″_id″ : 2,
                        ″name″ : ″127.0.0.1:27019″,
                        ″health″ : 1,
                        ″state″ : 2,
                        ″stateStr″ : ″SECONDARY″,
                        ″uptime″ : 334,
                        ″optime″ : {
                                ″ts″ : Timestamp(1590293555, 1),
                                "t" : NumberLong(14)
                        },
                        "optimeDurable" : {
                                "ts" : Timestamp(1590293555, 1),
                                "t" : NumberLong(14)
                        },
                        "optimeDate" : ISODate("2020-05-24T04:12:35Z"),
                        "optimeDurableDate" : ISODate("2020-05-24T04:12:35Z"),
                        "lastHeartbeat" : ISODate("2020-05-24T04:12:41.277Z"),
                        "lastHeartbeatRecv"     :     ISODate("2020-05-
24T04:12:41.793Z"),
                        "pingMs" : NumberLong(0),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "127.0.0.1:27017",
                        "syncSourceHost" : "127.0.0.1:27017",
                        "syncSourceId" : 0,
                        "infoMessage" : "",
                        "configVersion" : 4
                },
                {
                        "_id" : 3,
                        "name" : "127.0.0.1:27020",
                        "health" : 1,
                        "state" : 7,
                        "stateStr" : "ARBITER",
                        "uptime" : 327,
                        "lastHeartbeat" : ISODate("2020-05-24T04:12:41.433Z"),
                        "lastHeartbeatRecv"     :     ISODate("2020-05-
24T04:12:41.410Z"),
                        "pingMs" : NumberLong(2),
                        "lastHeartbeatMessage" : "",
                        "syncingTo" : "",
                        "syncSourceHost" : "",
                        "syncSourceId" : -1,
                        "infoMessage" : "",
                ″configVersion″ : 4
        }
    ],
    ″ok″ : 1,
    ″operationTime″ : Timestamp(1590293555, 1),
    ″$clusterTime″ : {
        ″clusterTime″ : Timestamp(1590293555, 1),
        ″signature″ : {
            ″hash″ : BinData(0,″5kS7yi4j56MmbO3lbSWskou5qzE=″),
            ″keyId″ : NumberLong(″6830220365512310785″)
        }
    }
}
```