## 第四十二章：MongoDB数据库复制集

**一、复制集概述；**

**二、复制集原理；**

**三、案例：搭建Mongodb实现应用复制集；**

**一、复制集概述：**

**组成：**

Mongodb复制集（副本集replica set）由一组Mongod实例（进程）组成，包含一个Primary节点和多个Secondary节点，Mongodb Driver（客户端）的所有数据都写入Primary，Secondary通过oplog来同步Primary的数据，保证主节点和从节点数据的一致性，复制集在完成主从复制的基础上，通过心跳机制，一旦primary节点出现宕机，则触发选举一个新的主节点，剩下的secondary节点指向新的primary，时间应该在10-30s内完成感知primary节点故障，实现高可用数据库集群；

**特点：**

主是唯一的，但不是固定的；

通过oplog同步数据保证数据的一致性；

从库无法写入（默认情况下，不使用驱动连接时，读也是不能查询的）；

相对于传统的主从结构，复制集可以自动容灾；

**二、复制集原理：**

**角色（按是否存储数据划分）：**

Primary：主节点，由选举产生，负责客户端的写操作，产生oplog日志文件；

Secondary：从节点，负责客户端的读操作，提供数据的备份和故障的切换；

Arbiter：仲裁节点，只参与选举的投票，不会成为primary，也不向Primary同步数据，若部署了一个2个节点的复制集，1个Primary，1个Secondary，任意节点宕机，复制集将不能提供服务了（无法选出Primary），这时可以给复制集添加一个Arbiter节点，即使有节点宕机，仍能选出Primary；

**角色（按类型区分）：**

Standard（标准）：这种是常规节点，它存储一份完整的数据副本，参与投票选举，有可能成为主节点；

Passive（被动）：存储完整的数据副本，参与投票，不能成为活跃节点。

Arbiter（投票）：仲裁节点只参与投票，不接收复制的数据，也不能成为活跃节点。

注：每个参与节点（非仲裁者）有个优先权（0-1000），优先权（priority）为0则是被动的，不能成为活跃节点，优先权不为0的，按照由大到小选出活跃节点，优先值一样的则看谁的数据比较新；

注：Mongodb 3.0里，复制集成员最多50个，参与Primary选举投票的成员最多7个；

**选举：**

每个节点通过优先级定义出节点的类型（标准、被动、投票）；

标准节点通过对比自身数据进行选举出primary节点或者secondary节点；

影响选举的因素：

1.心跳检测：复制集内成员每隔两秒向其他成员发送心跳检测信息，若10秒内无响应，则标记其为不可用；

2.连接：在多个节点中，最少保证两个节点为活跃状态，如果集群中共三个节点，挂掉两个节点，那么剩余的节点无论状态是primary还是处于选举过程中，都会直接被降权为secondary；

触发选举的情况：

1.初始化状态 2.从节点们无法与主节点进行通信 3.主节点辞职

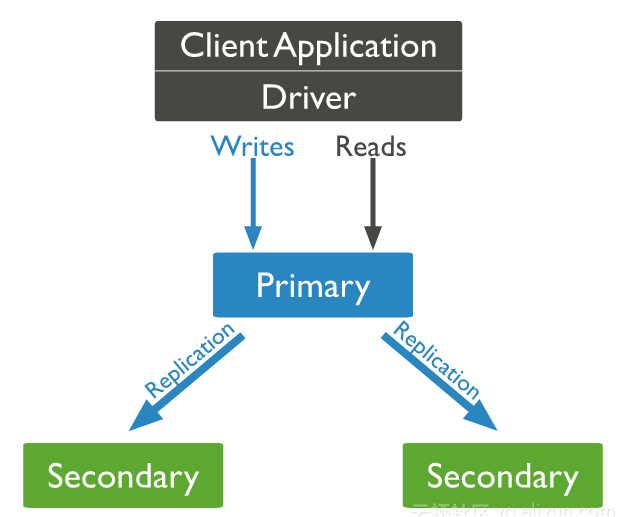
主节点辞职的情况：

1.在接收到replSetStepDown命令后；

2.在现有的环境中，其他secondary节点的数据落后于本身10s内，且拥有更高优先级；

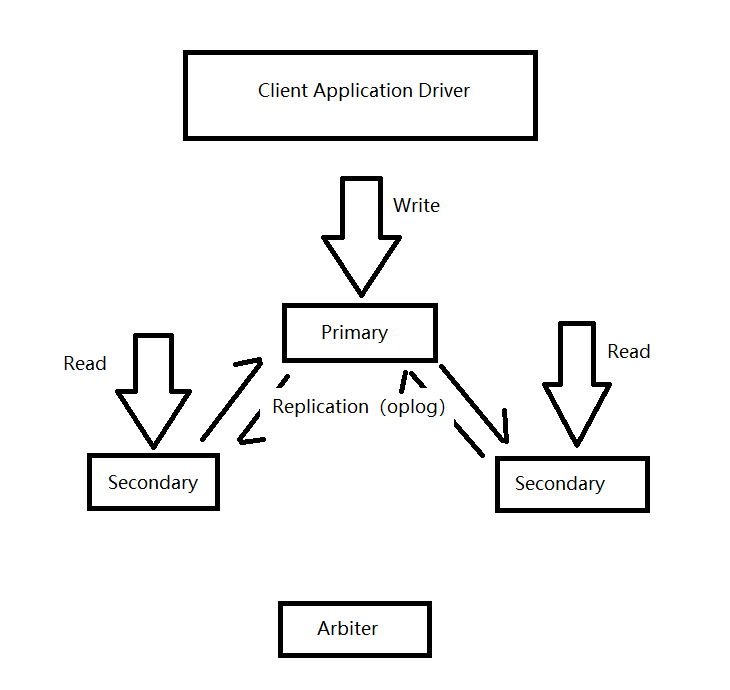
3.当主节点无法与群集中多数节点通信；

注：当主节点辞职后，主节点将关闭自身所有的连接，避免出现客户端在从节点进行写入操作；



**三、案例：搭建Mongodb实现应用复制集；**

**案例拓扑：**



**异常处理：**

当Primary宕机时，如果有数据未同步到Secondary，当Primary重新加入时，如果新的Primary上已经发生了写操作，则旧Primary需要回滚部分操作，以保证数据集与新的Primary一致。旧Primary将回滚的数据写到单独的rollback目录下，数据库管理员可根据需要使用mongorestore进行恢复。

**案例环境：**

|  |  |  |  |
| --- | --- | --- | --- |
| 系统 | 主机名 | IP地址 | 所需软件 |
| Centos 7.4 | mongodb.benet.com | 192.168.100.101 | mongodb-linux-x86\_64-rhel70-3.6.3.tgz |

**实验步骤：**

* 安装mongodb；
* 创建并启动四个实例；
* 配置实例；
* 创建复制集并添加节点；
* 模拟Primary节点出现故障，查看角色切换情况；
* 手动切换Primary角色；
* 指定节点的优先级，验证角色重选情况；
* 将标准节点统统停掉，被动节点也不会成为主节点；
* 查询复制集状态以及查看oplog日志文件的大小；
* 部署用户认证登录（密钥对）的复制集；
* **安装mongodb；**

下载mongodb软件包；

[root@mongodb ~]# tar zxvf mongodb-linux-x86\_64-rhel70-3.6.3.tgz

[root@mongodb ~]# mv mongodb-linux-x86\_64-rhel70-3.6.3 /usr/local/mongodb

[root@mongodb ~]# echo "export PATH=/usr/local/mongodb/bin:\$PATH" >>/etc/profile

[root@mongodb ~]# source /etc/profile

[root@mongodb ~]# ulimit -n 25000

[root@mongodb ~]# ulimit -u 25000

[root@mongodb ~]# echo 0 >/proc/sys/vm/zone\_reclaim\_mode

[root@mongodb ~]# sysctl -w vm.zone\_reclaim\_mode=0

[root@mongodb ~]# echo never >/sys/kernel/mm/transparent\_hugepage/enabled

[root@mongodb ~]# echo never >/sys/kernel/mm/transparent\_hugepage/defrag

* **创建并启动四个实例；**

[root@mongodb ~]# cd /usr/local/mongodb/bin/

[root@mongodb bin]# mkdir {../mongodb1,../mongodb2,../mongodb3,../mongodb4}

[root@mongodb bin]# mkdir ../logs

[root@mongodb bin]# touch ../logs/mongodb{1..4}.log

[root@mongodb bin]# chmod 777 ../logs/mongodb\*

[root@mongodb bin]# cat <<END >>/usr/local/mongodb/bin/mongodb1.conf

bind\_ip=192.168.100.101

port=27017

dbpath=/usr/local/mongodb/mongodb1/

logpath=/usr/local/mongodb/logs/mongodb1.log

logappend=true

fork=true

maxConns=5000

replSet=haha

#replication name

END

[root@mongodb bin]# cat <<END >>/usr/local/mongodb/bin/mongodb2.conf

bind\_ip=192.168.100.101

port=27018

dbpath=/usr/local/mongodb/mongodb2/

logpath=/usr/local/mongodb/logs/mongodb2.log

logappend=true

fork=true

maxConns=5000

replSet=haha

END

[root@mongodb bin]# cat <<END >>/usr/local/mongodb/bin/mongodb3.conf

bind\_ip=192.168.100.101

port=27019

dbpath=/usr/local/mongodb/mongodb3/

logpath=/usr/local/mongodb/logs/mongodb3.log

logappend=true

fork=true

maxConns=5000

replSet=haha

END

[root@mongodb bin]# cat <<END >>/usr/local/mongodb/bin/mongodb4.conf

bind\_ip=192.168.100.101

port=27020

dbpath=/usr/local/mongodb/mongodb4/

logpath=/usr/local/mongodb/logs/mongodb4.log

logappend=true

fork=true

maxConns=5000

replSet=haha

END

[root@mongodb bin]# cd

[root@mongodb ~]# mongod -f /usr/local/mongodb/bin/mongodb1.conf

[root@mongodb ~]# mongod -f /usr/local/mongodb/bin/mongodb2.conf

[root@mongodb ~]# mongod -f /usr/local/mongodb/bin/mongodb3.conf

[root@mongodb ~]# mongod -f /usr/local/mongodb/bin/mongodb4.conf

[root@mongodb ~]# netstat -utpln |grep mongod

tcp 0 0 192.168.100.101:27019 0.0.0.0:\* LISTEN 2271/mongod

tcp 0 0 192.168.100.101:27020 0.0.0.0:\* LISTEN 15260/mongod

tcp 0 0 192.168.100.101:27017 0.0.0.0:\* LISTEN 2440/mongod

tcp 0 0 192.168.100.101:27018 0.0.0.0:\* LISTEN 1412/mongod

[root@mongodb ~]# echo -e "/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/bin/mongodb1.conf \n/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/bin/mongodb2.conf\n/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/bin/mongodb3.conf\n/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/bin/mongodb4.conf">>/etc/rc.local

[root@mongodb ~]# chmod +x /etc/rc.local

[root@mongodb ~]# cat <<END >>/etc/init.d/mongodb

#!/bin/bash

INSTANCE=\$1

ACTION=\$2

case "\$ACTION" in

'start')

/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/bin/"\$INSTANCE".conf;;

'stop')

/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/bin/"\$INSTANCE".conf --shutdown;;

'restart')

/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/bin/"\$INSTANCE".conf --shutdown

/usr/local/mongodb/bin/mongod -f /usr/local/mongodb/bin/"\$INSTANCE".conf;;

esac

END

[root@mongodb ~]# chmod +x /etc/init.d/mongodb

[root@mongodb ~]# /etc/init.d/mongodb mongodb1 stop

killing process with pid: 1301

[root@mongodb ~]# /etc/init.d/mongodb mongodb1 start

about to fork child process, waiting until server is ready for connections.

forked process: 1457

child process started successfully, parent exiting

[root@mongodb ~]# mongo --port 27017 --host 192.168.100.101

* **配置实例；**

[root@mongodb ~]# mongo --port 27017 --host 192.168.100.101

> show dbs

2018-04-19T04:34:35.871+0800 E QUERY [thread1] Error: listDatabases failed:{

"ok" : 0,

"errmsg" : "not master and slaveOk=false",

"code" : 13435,

"codeName" : "NotMasterNoSlaveOk"

} :

\_getErrorWithCode@src/mongo/shell/utils.js:25:13

Mongo.prototype.getDBs@src/mongo/shell/mongo.js:65:1

shellHelper.show@src/mongo/shell/utils.js:816:19

shellHelper@src/mongo/shell/utils.js:706:15

@(shellhelp2):1:1

> rs.status()

{

"info" : "run rs.initiate(...) if not yet done for the set",

"ok" : 0,

"errmsg" : "no replset config has been received",

"code" : 94,

"codeName" : "NotYetInitialized"

}

> rs.isMaster() ##查看复制集节点

{

"ismaster" : false,

"secondary" : false,

"info" : "Does not have a valid replica set config",

"isreplicaset" : true,

"maxBsonObjectSize" : 16777216,

"maxMessageSizeBytes" : 48000000,

"maxWriteBatchSize" : 100000,

"localTime" : ISODate("2018-04-18T20:36:31.698Z"),

"minWireVersion" : 0,

"maxWireVersion" : 6,

"readOnly" : false,

"ok" : 1

}

> exit

[root@mongodb ~]# mongo --port 27018 --host 192.168.100.101

> show dbs

2018-04-19T04:34:56.884+0800 E QUERY [thread1] Error: listDatabases failed:{

"ok" : 0,

"errmsg" : "not master and slaveOk=false",

"code" : 13435,

"codeName" : "NotMasterNoSlaveOk"

} :

\_getErrorWithCode@src/mongo/shell/utils.js:25:13

Mongo.prototype.getDBs@src/mongo/shell/mongo.js:65:1

shellHelper.show@src/mongo/shell/utils.js:816:19

shellHelper@src/mongo/shell/utils.js:706:15

@(shellhelp2):1:1

> exit

* **创建复制集并添加节点；**

[root@mongodb ~]# mongo --port 27017 --host 192.168.100.101

>cfg={"\_id":"haha","members":[{"\_id":0,"host":"192.168.100.101:27017"},{"\_id":1,"host":"192.168.100.101:27018"},{"\_id":2,"host":"192.168.100.101:27019"}]} ##添加节点

{

"\_id" : "haha",

"members" : [

{

"\_id" : 0,

"host" : "192.168.100.101:27017"

},

{

"\_id" : 1,

"host" : "192.168.100.101:27018"

},

{

"\_id" : 2,

"host" : "192.168.100.101:27019"

}

]

}

> rs.initiate(cfg) ##初始化节点

{

"ok" : 1,

"operationTime" : Timestamp(1524083843, 1),

"$clusterTime" : {

"clusterTime" : Timestamp(1524083843, 1),

"signature" : {

"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),

"keyId" : NumberLong(0)

}

}

}

haha:OTHER>

haha:PRIMARY>

haha:PRIMARY> rs.status()

{

"set" : "haha",

"date" : ISODate("2018-04-18T20:37:54.095Z"),

"myState" : 1,

"term" : NumberLong(1),

"heartbeatIntervalMillis" : NumberLong(2000),

"optimes" : {

"lastCommittedOpTime" : {

"ts" : Timestamp(1524083855, 5),

"t" : NumberLong(1)

},

"readConcernMajorityOpTime" : {

"ts" : Timestamp(1524083855, 5),

"t" : NumberLong(1)

},

"appliedOpTime" : {

"ts" : Timestamp(1524083855, 5),

"t" : NumberLong(1)

},

"durableOpTime" : {

"ts" : Timestamp(1524083855, 5),

"t" : NumberLong(1)

}

},

"members" : [

{

"\_id" : 0,

"name" : "192.168.100.101:27017",

"health" : 1,

"state" : 1,

"stateStr" : "PRIMARY",

"uptime" : 329,

"optime" : {

"ts" : Timestamp(1524083855, 5),

"t" : NumberLong(1)

},

"optimeDate" : ISODate("2018-04-18T20:37:35Z"),

"infoMessage" : "could not find member to sync from",

"electionTime" : Timestamp(1524083854, 1),

"electionDate" : ISODate("2018-04-18T20:37:34Z"),

"configVersion" : 1,

"self" : true

},

{

"\_id" : 1,

"name" : "192.168.100.101:27018",

"health" : 1,

"state" : 2,

"stateStr" : "SECONDARY",

"uptime" : 30,

"optime" : {

"ts" : Timestamp(1524083855, 5),

"t" : NumberLong(1)

},

"optimeDurable" : {

"ts" : Timestamp(1524083855, 5),

"t" : NumberLong(1)

},

"optimeDate" : ISODate("2018-04-18T20:37:35Z"),

"optimeDurableDate" : ISODate("2018-04-18T20:37:35Z"),

"lastHeartbeat" : ISODate("2018-04-18T20:37:54.043Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T20:37:52.499Z"),

"pingMs" : NumberLong(0),

"syncingTo" : "192.168.100.101:27017",

"configVersion" : 1

},

{

"\_id" : 2,

"name" : "192.168.100.101:27019",

"health" : 1,

"state" : 2,

"stateStr" : "SECONDARY",

"uptime" : 30,

"optime" : {

"ts" : Timestamp(1524083855, 5),

"t" : NumberLong(1)

},

"optimeDurable" : {

"ts" : Timestamp(1524083855, 5),

"t" : NumberLong(1)

},

"optimeDate" : ISODate("2018-04-18T20:37:35Z"),

"optimeDurableDate" : ISODate("2018-04-18T20:37:35Z"),

"lastHeartbeat" : ISODate("2018-04-18T20:37:54.043Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T20:37:52.500Z"),

"pingMs" : NumberLong(0),

"syncingTo" : "192.168.100.101:27017",

"configVersion" : 1

}

],

"ok" : 1,

"operationTime" : Timestamp(1524083855, 5),

"$clusterTime" : {

"clusterTime" : Timestamp(1524083855, 5),

"signature" : {

"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),

"keyId" : NumberLong(0)

}

}

}

haha:PRIMARY> rs.addArb("192.168.100.101:27020") ##添加仲裁节点

{

"ok" : 1,

"operationTime" : Timestamp(1524083905, 1),

"$clusterTime" : {

"clusterTime" : Timestamp(1524083905, 1),

"signature" : {

"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),

"keyId" : NumberLong(0)

}

}

}

haha:PRIMARY> rs.status()

{

"set" : "haha",

"date" : ISODate("2018-04-18T20:38:41.468Z"),

"myState" : 1,

"term" : NumberLong(1),

"heartbeatIntervalMillis" : NumberLong(2000),

"optimes" : {

"lastCommittedOpTime" : {

"ts" : Timestamp(1524083915, 1),

"t" : NumberLong(1)

},

"readConcernMajorityOpTime" : {

"ts" : Timestamp(1524083915, 1),

"t" : NumberLong(1)

},

"appliedOpTime" : {

"ts" : Timestamp(1524083915, 1),

"t" : NumberLong(1)

},

"durableOpTime" : {

"ts" : Timestamp(1524083915, 1),

"t" : NumberLong(1)

}

},

"members" : [

{

"\_id" : 0,

"name" : "192.168.100.101:27017",

"health" : 1,

"state" : 1,

"stateStr" : "PRIMARY",

"uptime" : 376,

"optime" : {

"ts" : Timestamp(1524083915, 1),

"t" : NumberLong(1)

},

"optimeDate" : ISODate("2018-04-18T20:38:35Z"),

"infoMessage" : "could not find member to sync from",

"electionTime" : Timestamp(1524083854, 1),

"electionDate" : ISODate("2018-04-18T20:37:34Z"),

"configVersion" : 2,

"self" : true

},

{

"\_id" : 1,

"name" : "192.168.100.101:27018",

"health" : 1,

"state" : 2,

"stateStr" : "SECONDARY",

"uptime" : 78,

"optime" : {

"ts" : Timestamp(1524083905, 1),

"t" : NumberLong(1)

},

"optimeDurable" : {

"ts" : Timestamp(1524083905, 1),

"t" : NumberLong(1)

},

"optimeDate" : ISODate("2018-04-18T20:38:25Z"),

"optimeDurableDate" : ISODate("2018-04-18T20:38:25Z"),

"lastHeartbeat" : ISODate("2018-04-18T20:38:41.134Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T20:38:40.134Z"),

"pingMs" : NumberLong(0),

"configVersion" : 2

},

{

"\_id" : 2,

"name" : "192.168.100.101:27019",

"health" : 1,

"state" : 2,

"stateStr" : "SECONDARY",

"uptime" : 78,

"optime" : {

"ts" : Timestamp(1524083905, 1),

"t" : NumberLong(1)

},

"optimeDurable" : {

"ts" : Timestamp(1524083905, 1),

"t" : NumberLong(1)

},

"optimeDate" : ISODate("2018-04-18T20:38:25Z"),

"optimeDurableDate" : ISODate("2018-04-18T20:38:25Z"),

"lastHeartbeat" : ISODate("2018-04-18T20:38:41.134Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T20:38:40.136Z"),

"pingMs" : NumberLong(0),

"configVersion" : 2

},

{

"\_id" : 3,

"name" : "192.168.100.101:27020",

"health" : 1,

"state" : 7,

"stateStr" : "ARBITER",

"uptime" : 16,

"lastHeartbeat" : ISODate("2018-04-18T20:38:41.137Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T20:38:40.140Z"),

"pingMs" : NumberLong(0),

"configVersion" : 2

}

],

"ok" : 1,

"operationTime" : Timestamp(1524083915, 1),

"$clusterTime" : {

"clusterTime" : Timestamp(1524083915, 1),

"signature" : {

"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),

"keyId" : NumberLong(0)

}

}

}

注：rs.add()和rs.remove()命令分别用于添加和删除标准节点

haha:PRIMARY> show dbs

admin 0.000GB

config 0.000GB

local 0.000GB

haha:PRIMARY> use cloud

switched to db cloud

haha:PRIMARY> db.users.insert({"id":"1","name":"xiaoming"})

WriteResult({ "nInserted" : 1 })

haha:PRIMARY> db.users.find()

{ "\_id" : ObjectId("5ad7b245f6308759d4605b5c"), "id" : "1", "name" : "xiaoming" }

haha:PRIMARY> show dbs

admin 0.000GB

cloud 0.000GB

config 0.000GB

local 0.000GB

haha:PRIMARY> exit

[root@mongodb ~]# mongo --port 27018 --host 192.168.100.101

haha:SECONDARY> rs.status()

{

"set" : "haha",

"date" : ISODate("2018-04-18T20:50:50.975Z"),

"myState" : 2,

"term" : NumberLong(1),

"syncingTo" : "192.168.100.101:27017",

"heartbeatIntervalMillis" : NumberLong(2000),

"optimes" : {

"lastCommittedOpTime" : {

"ts" : Timestamp(1524084645, 1),

"t" : NumberLong(1)

},

"readConcernMajorityOpTime" : {

"ts" : Timestamp(1524084645, 1),

"t" : NumberLong(1)

},

"appliedOpTime" : {

"ts" : Timestamp(1524084645, 1),

"t" : NumberLong(1)

},

"durableOpTime" : {

"ts" : Timestamp(1524084645, 1),

"t" : NumberLong(1)

}

},

"members" : [

{

"\_id" : 0,

"name" : "192.168.100.101:27017",

"health" : 1,

"state" : 1,

"stateStr" : "PRIMARY",

"uptime" : 805,

"optime" : {

"ts" : Timestamp(1524084645, 1),

"t" : NumberLong(1)

},

"optimeDurable" : {

"ts" : Timestamp(1524084645, 1),

"t" : NumberLong(1)

},

"optimeDate" : ISODate("2018-04-18T20:50:45Z"),

"optimeDurableDate" : ISODate("2018-04-18T20:50:45Z"),

"lastHeartbeat" : ISODate("2018-04-18T20:50:49.966Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T20:50:49.997Z"),

"pingMs" : NumberLong(0),

"electionTime" : Timestamp(1524083854, 1),

"electionDate" : ISODate("2018-04-18T20:37:34Z"),

"configVersion" : 2

},

{

"\_id" : 1,

"name" : "192.168.100.101:27018",

"health" : 1,

"state" : 2,

"stateStr" : "SECONDARY",

"uptime" : 1497,

"optime" : {

"ts" : Timestamp(1524084645, 1),

"t" : NumberLong(1)

},

"optimeDate" : ISODate("2018-04-18T20:50:45Z"),

"syncingTo" : "192.168.100.101:27017",

"configVersion" : 2,

"self" : true

},

{

"\_id" : 2,

"name" : "192.168.100.101:27019",

"health" : 1,

"state" : 2,

"stateStr" : "SECONDARY",

"uptime" : 805,

"optime" : {

"ts" : Timestamp(1524084645, 1),

"t" : NumberLong(1)

},

"optimeDurable" : {

"ts" : Timestamp(1524084645, 1),

"t" : NumberLong(1)

},

"optimeDate" : ISODate("2018-04-18T20:50:45Z"),

"optimeDurableDate" : ISODate("2018-04-18T20:50:45Z"),

"lastHeartbeat" : ISODate("2018-04-18T20:50:49.811Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T20:50:49.965Z"),

"pingMs" : NumberLong(0),

"syncingTo" : "192.168.100.101:27017",

"configVersion" : 2

},

{

"\_id" : 3,

"name" : "192.168.100.101:27020",

"health" : 1,

"state" : 7,

"stateStr" : "ARBITER",

"uptime" : 745,

"lastHeartbeat" : ISODate("2018-04-18T20:50:49.966Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T20:50:50.378Z"),

"pingMs" : NumberLong(0),

"configVersion" : 2

}

],

"ok" : 1,

"operationTime" : Timestamp(1524084645, 1),

"$clusterTime" : {

"clusterTime" : Timestamp(1524084645, 1),

"signature" : {

"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),

"keyId" : NumberLong(0)

}

}

}

haha:SECONDARY> show dbs ##secondary节点默认无法读取，可以通过以下方式或者驱动方式实现

2018-04-19T04:52:42.813+0800 E QUERY [thread1] Error: listDatabases failed:{

"operationTime" : Timestamp(1524084755, 1),

"ok" : 0,

"errmsg" : "not master and slaveOk=false",

"code" : 13435,

"codeName" : "NotMasterNoSlaveOk",

"$clusterTime" : {

"clusterTime" : Timestamp(1524084755, 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:65:1

shellHelper.show@src/mongo/shell/utils.js:816:19

shellHelper@src/mongo/shell/utils.js:706:15

@(shellhelp2):1:1

haha:SECONDARY> db.getMongo().setSlaveOk();

haha:SECONDARY> show dbs

admin 0.000GB

cloud 0.000GB

config 0.000GB

local 0.000GB

haha:SECONDARY> use cloud

switched to db cloud

haha:SECONDARY> db.users.insert({"id":"2","name":"xiaohong"}) ##secondary节点无法写入

WriteResult({ "writeError" : { "code" : 10107, "errmsg" : "not master" } })

haha:SECONDARY> db.users.find()

{ "\_id" : ObjectId("5ad7b245f6308759d4605b5c"), "id" : "1", "name" : "xiaoming" }

haha:SECONDARY> exit

* **模拟Primary节点出现故障，查看角色切换情况；**

[root@mongodb ~]# mongod -f /usr/local/mongodb/bin/mongodb1.conf --shutdown

killing process with pid: 1457

[root@mongodb ~]# mongo --port 27018 --host 192.168.100.101

haha:SECONDARY> rs.status()

{

"set" : "haha",

"date" : ISODate("2018-04-18T21:16:22.623Z"),

"myState" : 2,

"term" : NumberLong(2),

"syncingTo" : "192.168.100.101:27019",

"heartbeatIntervalMillis" : NumberLong(2000),

"optimes" : {

"lastCommittedOpTime" : {

"ts" : Timestamp(1524086155, 1),

"t" : NumberLong(1)

},

"readConcernMajorityOpTime" : {

"ts" : Timestamp(1524086155, 1),

"t" : NumberLong(1)

},

"appliedOpTime" : {

"ts" : Timestamp(1524086174, 1),

"t" : NumberLong(2)

},

"durableOpTime" : {

"ts" : Timestamp(1524086174, 1),

"t" : NumberLong(2)

}

},

"members" : [

{

"\_id" : 0,

"name" : "192.168.100.101:27017",

"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("2018-04-18T21:16:19.030Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T21:16:03.344Z"),

"pingMs" : NumberLong(0),

"lastHeartbeatMessage" : "Connection refused",

"configVersion" : -1

},

{

"\_id" : 1,

"name" : "192.168.100.101:27018",

"health" : 1,

"state" : 2,

"stateStr" : "SECONDARY",

"uptime" : 3029,

"optime" : {

"ts" : Timestamp(1524086174, 1),

"t" : NumberLong(2)

},

"optimeDate" : ISODate("2018-04-18T21:16:14Z"),

"syncingTo" : "192.168.100.101:27019",

"infoMessage" : "syncing from: 192.168.100.101:27019",

"configVersion" : 2,

"self" : true

},

{

"\_id" : 2,

"name" : "192.168.100.101:27019",

"health" : 1,

"state" : 1,

"stateStr" : "PRIMARY",

"uptime" : 2337,

"optime" : {

"ts" : Timestamp(1524086174, 1),

"t" : NumberLong(2)

},

"optimeDurable" : {

"ts" : Timestamp(1524086174, 1),

"t" : NumberLong(2)

},

"optimeDate" : ISODate("2018-04-18T21:16:14Z"),

"optimeDurableDate" : ISODate("2018-04-18T21:16:14Z"),

"lastHeartbeat" : ISODate("2018-04-18T21:16:19.030Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T21:16:21.575Z"),

"pingMs" : NumberLong(0),

"electionTime" : Timestamp(1524086173, 1),

"electionDate" : ISODate("2018-04-18T21:16:13Z"),

"configVersion" : 2

},

{

"\_id" : 3,

"name" : "192.168.100.101:27020",

"health" : 1,

"state" : 7,

"stateStr" : "ARBITER",

"uptime" : 2277,

"lastHeartbeat" : ISODate("2018-04-18T21:16:19.030Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T21:16:20.894Z"),

"pingMs" : NumberLong(0),

"configVersion" : 2

}

],

"ok" : 1,

"operationTime" : Timestamp(1524086174, 1),

"$clusterTime" : {

"clusterTime" : Timestamp(1524086174, 1),

"signature" : {

"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),

"keyId" : NumberLong(0)

}

}

}

haha:SECONDARY> exit

[root@mongodb ~]# mongo --port 27019 --host 192.168.100.101

haha:PRIMARY> exit

[root@mongodb ~]# mongod -f /usr/local/mongodb/bin/mongodb1.conf

* **手动切换Primary角色；**

[root@mongodb ~]# mongo --port 27019 --host 192.168.100.101

haha:PRIMARY> rs.stepDown(60,30) 指示副本集的[主](https://docs.mongodb.com/v3.6/reference/glossary/#term-primary)节点成为 [辅助节点](https://docs.mongodb.com/v3.6/reference/glossary/#term-secondary)。在主要步骤失败后，eligble辅助人员将举行[小学选举](https://docs.mongodb.com/v3.6/core/replica-set-elections/#replica-set-election-internals)。

haha:SECONDARY> exit

[root@mongodb ~]# mongo --port 27018 --host 192.168.100.101

haha:PRIMARY> exit

* **指定节点的优先级，验证角色重选情况；**

[root@mongodb ~]# mongo --port 27018 --host 192.168.100.101

haha:PRIMARY> cfg={"\_id":"haha","members":[{"\_id":0,"host":"192.168.100.101:27017","priority":100},{"\_id":1,"host":"192.168.100.101:27018","priority":100},{"\_id":2,"host":"192.168.100.101:27019","priority":10}]}

{

"\_id" : "haha",

"members" : [

{

"\_id" : 0,

"host" : "192.168.100.101:27017",

"priority" : 100

},

{

"\_id" : 1,

"host" : "192.168.100.101:27018",

"priority" : 100

},

{

"\_id" : 2,

"host" : "192.168.100.101:27019",

"priority" : 10

}

]

}

haha:PRIMARY> rs.reconfig(cfg)

{

"ok" : 1,

"operationTime" : Timestamp(1524086716, 1),

"$clusterTime" : {

"clusterTime" : Timestamp(1524086716, 1),

"signature" : {

"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),

"keyId" : NumberLong(0)

}

}

}

haha:PRIMARY> rs.isMaster()

{

"hosts" : [

"192.168.100.101:27017",

"192.168.100.101:27018",

"192.168.100.101:27019"

],

"setName" : "haha",

"setVersion" : 3,

"ismaster" : true,

"secondary" : false,

"primary" : "192.168.100.101:27018",

"me" : "192.168.100.101:27018",

"electionId" : ObjectId("7fffffff0000000000000003"),

"lastWrite" : {

"opTime" : {

"ts" : Timestamp(1524086733, 1),

"t" : NumberLong(3)

},

"lastWriteDate" : ISODate("2018-04-18T21:25:33Z"),

"majorityOpTime" : {

"ts" : Timestamp(1524086733, 1),

"t" : NumberLong(3)

},

"majorityWriteDate" : ISODate("2018-04-18T21:25:33Z")

},

"maxBsonObjectSize" : 16777216,

"maxMessageSizeBytes" : 48000000,

"maxWriteBatchSize" : 100000,

"localTime" : ISODate("2018-04-18T21:25:42.122Z"),

"logicalSessionTimeoutMinutes" : 30,

"minWireVersion" : 0,

"maxWireVersion" : 6,

"readOnly" : false,

"ok" : 1,

"operationTime" : Timestamp(1524086733, 1),

"$clusterTime" : {

"clusterTime" : Timestamp(1524086733, 1),

"signature" : {

"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),

"keyId" : NumberLong(0)

}

}

}

haha:PRIMARY> exit

[root@mongodb ~]# mongod -f /usr/local/mongodb/bin/mongodb2.conf --shutdown

[root@mongodb ~]# mongo --port 27017 --host 192.168.100.101

haha:PRIMARY> exit

* **将标准节点统统停掉，被动节点也不会成为主节点；**

[root@mongodb ~]# mongod -f /usr/local/mongodb/bin/mongodb1.conf --shutdown

[root@mongodb ~]# mongod -f /usr/local/mongodb/bin/mongodb2.conf --shutdown

[root@mongodb ~]# mongo --port 27019 --host 192.168.100.101

haha:SECONDARY> rs.status()

{

"set" : "haha",

"date" : ISODate("2018-04-18T21:29:42.973Z"),

"myState" : 2,

"term" : NumberLong(5),

"heartbeatIntervalMillis" : NumberLong(2000),

"optimes" : {

"lastCommittedOpTime" : {

"ts" : Timestamp(1524086961, 1),

"t" : NumberLong(5)

},

"readConcernMajorityOpTime" : {

"ts" : Timestamp(1524086961, 1),

"t" : NumberLong(5)

},

"appliedOpTime" : {

"ts" : Timestamp(1524086961, 1),

"t" : NumberLong(5)

},

"durableOpTime" : {

"ts" : Timestamp(1524086961, 1),

"t" : NumberLong(5)

}

},

"members" : [

{

"\_id" : 0,

"name" : "192.168.100.101:27017",

"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("2018-04-18T21:29:38.566Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T21:29:26.484Z"),

"pingMs" : NumberLong(0),

"lastHeartbeatMessage" : "Connection refused",

"configVersion" : -1

},

{

"\_id" : 1,

"name" : "192.168.100.101:27018",

"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("2018-04-18T21:29:38.566Z"),

"lastHeartbeatRecv" : ISODate("2018-04-18T21:26:42.249Z"),

"pingMs" : NumberLong(0),

"lastHeartbeatMessage" : "Connection refused",

"configVersion" : -1

},

{

"\_id" : 2,

"name" : "192.168.100.101:27019",

"health" : 1,

"state" : 2,

"stateStr" : "SECONDARY",

"uptime" : 3826,

"optime" : {

"ts" : Timestamp(1524086961, 1),

"t" : NumberLong(5)

},

"optimeDate" : ISODate("2018-04-18T21:29:21Z"),

"infoMessage" : "could not find member to sync from",

"configVersion" : 3,

"self" : true

}

],

"ok" : 1,

"operationTime" : Timestamp(1524086961, 1),

"$clusterTime" : {

"clusterTime" : Timestamp(1524086961, 1),

"signature" : {

"hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAA="),

"keyId" : NumberLong(0)

}

}

}

haha:SECONDARY> exit

[root@mongodb ~]# mongod -f /usr/local/mongodb/bin/mongodb1.conf

[root@mongodb ~]# mongod -f /usr/local/mongodb/bin/mongodb2.conf

[root@mongodb ~]# netstat -utpln |grep mongod

tcp 0 0 192.168.100.101:27019 0.0.0.0:\* LISTEN 1357/mongod

tcp 0 0 192.168.100.101:27020 0.0.0.0:\* LISTEN 1385/mongod

tcp 0 0 192.168.100.101:27017 0.0.0.0:\* LISTEN 14363/mongod

tcp 0 0 192.168.100.101:27018 0.0.0.0:\* LISTEN 14442/mongod

* **查询复制集状态以及查看oplog日志文件的大小；**

[root@mongodb ~]# mongo --port 27018 --host 192.168.100.101

haha:PRIMARY> use local

haha:PRIMARY> rs.printSlaveReplicationInfo() ##查看节点信息

source: 192.168.100.101:27017

syncedTo: Thu Apr 19 2018 05:34:37 GMT+0800 (CST)

0 secs (0 hrs) behind the primary

source: 192.168.100.101:27019

syncedTo: Thu Apr 19 2018 05:34:37 GMT+0800 (CST)

0 secs (0 hrs) behind the primary

haha:PRIMARY> rs.printReplicationInfo()

configured oplog size: 990MB

log length start to end: 3464secs (0.96hrs)

oplog first event time: Thu Apr 19 2018 04:37:23 GMT+0800 (CST)

oplog last event time: Thu Apr 19 2018 05:35:07 GMT+0800 (CST)

now: Thu Apr 19 2018 05:35:16 GMT+0800 (CST)

haha:PRIMARY> db.oplog.rs.stats()

{

"ns" : "local.oplog.rs",

"size" : 50356,

"count" : 450,

"avgObjSize" : 111,

"storageSize" : 61440,

"capped" : true,

"max" : -1,

"maxSize" : 1038090240, ##单位是字节

...

* **部署用户认证登录（密钥对）的复制集；**

[root@mongodb ~]# mongo --port 27018 --host 192.168.100.101

haha:PRIMARY> use admin

switched to db admin

haha:PRIMARY> db.createUser({"user":"root","pwd":"123","roles":["root"]})

Successfully added user: { "user" : "root", "roles" : [ "root" ] }

haha:PRIMARY> exit

[root@mongodb ~]# echo -e "clusterAuthMode=keyFile\nkeyFile=/usr/local/mongodb/bin/cloudkey1" >>/usr/local/mongodb/bin/mongodb1.conf

[root@mongodb ~]# echo -e "clusterAuthMode=keyFile\nkeyFile=/usr/local/mongodb/bin/cloudkey2" >>/usr/local/mongodb/bin/mongodb2.conf

[root@mongodb ~]# echo -e "clusterAuthMode=keyFile\nkeyFile=/usr/local/mongodb/bin/cloudkey3" >>/usr/local/mongodb/bin/mongodb3.conf

[root@mongodb ~]# echo "haha key" >/usr/local/mongodb/bin/cloudkey1

[root@mongodb ~]# echo "haha key" >/usr/local/mongodb/bin/cloudkey2

[root@mongodb ~]# echo "haha key" >/usr/local/mongodb/bin/cloudkey3

[root@mongodb ~]# chmod 600 /usr/local/mongodb/bin/cloudkey\*

[root@mongodb ~]# /etc/init.d/mongodb mongodb1 restart

[root@mongodb ~]# /etc/init.d/mongodb mongodb2 restart

[root@mongodb ~]# /etc/init.d/mongodb mongodb3 restart

[root@mongodb ~]# mongo --port 27018 --host 192.168.100.101

haha:PRIMARY> show dbs

2018-04-19T12:02:57.720+0800 E QUERY [thread1] Error: listDatabases failed:{

"operationTime" : Timestamp(1524110569, 1),

"ok" : 0,

"errmsg" : "not authorized on admin to execute command { listDatabases: 1.0, $clusterTime: { clusterTime: Timestamp(1524110569, 1), signature: { hash: BinData(0, 85A5B587BFFE4B2FB534789EA87032CB263D8121), keyId: 6545890313586606081 } }, $db: \"admin\" }",

...

haha:PRIMARY> use admin

switched to db admin

haha:PRIMARY> db.auth("root","123")

1

haha:PRIMARY> show dbs

admin 0.000GB

cloud 0.000GB

config 0.000GB

local 0.000GB

haha:PRIMARY> exit