Mongodb replica set部署

参考网址：

https://docs.mongodb.com/manual/tutorial/configure-a-non-voting-replica-set-member/

# 配置前的准备：

## 保持时间一致

保证各个机器的时间一致，前后差异不要大于30s，最好是一致。

执行一次时间同步：/usr/sbin/ntpdate time.nist.gov

在计划任务中增加

crontab –e

0 12 \* \* \* /usr/sbin/ntpdate time.nist.gov >/dev/null 2>&1

## 配置同时允许打开的文件最大数

查看系统允许同时打开文件的最大数。

ulimit -a

#查看系统允许的最大句柄文件数

cat /proc/sys/fs/file-max

#修改允许最大打开文件数【永久修改】

#在【/etc/security/limits.conf】中，增加下面的代码

\* soft nofile 65536

\* hard nofile 65536

然后重启生效

## 生成key文件，使用x.509 certificates.

openssl rand -base64 756 > <path-to-keyfile>

# 账号密码设置

## 创建账号，并且修改密码加密方式为【MONGODB-CR】

连接mongos，配置admin账号，实际上是配置在config节点上。

连接replica set的PRIMARY节点，配置admin账号，仲裁节点不用配置，也不允许配置。

#先增加一个账号，否则下面会没有记录

use admin;

db.createUser(

{

user: "sa",

pwd: "sa",

roles: [ "\_\_system","backup","clusterAdmin","dbAdminAnyDatabase","readWriteAnyDatabase","userAdminAnyDatabase" ]

}

);

#给管理员赋予【\_\_system】的角色

#切换到admin数据库

use admin

#执行下面的代码

var schema = db.system.version.findOne({"\_id" : "authSchema"})

schema.currentVersion = 3

#将会输出3

#保存该记录

db.system.version.save(schema)

#成功将输出WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

#接下来重启数据库，就可以开始创建用户了。

#创建之后，查看一下密码的加密方式

use admin

db.system.users.find({"user":"test"})

#结果类似如下即可

#{ "\_id" : "userdb.myuser", "user" : "test", "db" : "userdb", "credentials" : { "MONGODB-CR" : ... }

#原来的为

#{ "\_id" : "userdb.myuser", "user" : "test", "db" : "userdb", "credentials" : { "SCRAM-SHA-1" : ... }

#删除用户

use admin

db.system.users.remove({user:"java1"});

#修改用户密码（为已经存在的用户更改密码）

db.createUser(

{

user: "sa",

pwd: "sa",

roles: [ "\_\_system","backup","clusterAdmin","dbAdminAnyDatabase","readWriteAnyDatabase","userAdminAnyDatabase" ]

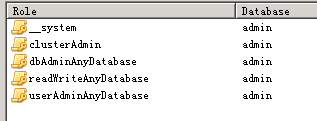
}

);

#用户授权

db.auth("java","java");

最终，增加这些角色权限



## 使用keyfile认证

### 生成keyfile

openssl rand -base64 741 > /var/mongo/mongodb-keyfile

#注意，一定要使用600权限，否则mongo认为文件访问权限太宽松，启动失败

chmod 600 mongodb-keyfile

主、从、仲裁节点都配置keyfile

security:

keyFile: /home/mongos27020/mongodb-keyfile

authorization: enabled

replication:

replSetName: mongodb30000

上面标红的名字，是可以随便更改的，保证【主】【从】【仲裁】的名字一致即可。

【仲裁】节点的特殊配置：（journal设置为false，节省空间。原因：仲裁不保存任何数据）

storage:

dbPath: /home/mongodb30000arb/data

journal:

enabled: false

directoryPerDB: true

重新启动主、从、仲裁、config、mongos，即可生效

# 配置Primary、 Secondary、仲裁

## 修改【主】【从】【仲裁】的配置文件，可根据需要调整配置

# mongod.conf

# for documentation of all options, see:

# http://docs.mongodb.org/manual/reference/configuration-options/

# where to write logging data.

systemLog:

destination: file

logAppend: true

path: /home/mongodb27020\_27025/log/mongod.log

# Where and how to store data.

storage:

dbPath: /home/mongodb27020\_27025/data

journal:

enabled: true

directoryPerDB: true

# engine:

# mmapv1:

# wiredTiger:

# how the process runs

processManagement:

fork: true # fork and run in background

pidFilePath: /home/mongodb27020\_27025/mongod.pid # location of pidfile

# network interfaces

net:

port: 27025

# bindIp: 127.0.0.1 # Listen to local interface only, comment to listen on all interfaces.

# make lower version driver can login

setParameter:

failIndexKeyTooLong: false

# authenticationMechanisms: MONGODB-CR

security:

keyFile: /home/mongos27020/mongodb-keyfile

authorization: enabled

#operationProfiling:

replication:

replSetName: mongodb30000

#sharding:

# clusterRole: shardsvr

## Enterprise-Only Options

#auditLog:

#snmp:

2. 下面我们开始配置3个mongodb，

mongodb27020\_27025: 主，ip为192.168.160.130

mongodb27020\_27026: 从，ip为192.168.160.130

mongodb27020arb27021：仲裁，ip为192.168.160.130

## 启动Primary和Secondary、arb

/home/mongodb/bin/mongod -f /home/mongodb27020\_27025/mongod.conf

/home/mongodb/bin/mongod -f /home/mongodb27020\_27026/mongod.conf

/home/mongodb/bin/mongod -f /home/mongodb27020arb27021/mongod.conf

## 配置主、从、仲裁

### 连接Primary

./mongo 127.0.0.1:27025/admin -u sa -p

### 初始化副本集配置

use admin;

config={\_id:"mongodb30000",members:[{\_id:0,host:"192.168.160.130:27025","priority":20}]}

rs.initiate(config);

确认返回的是{ "ok" : 1 }

上面config里面，是当前主节点对外的ip，即【从】节点能够访问到的ip。

#查看集群节点的状态

rs.status();

### 增加一个replica set 节点（即从节点）

rs.add("192.168.160.130:27026")

#查看集群节点的状态

rs.status();

### **增加一个arb 节点** （即仲裁节点）

rs.addArb("192.168.160.130:27022")

#查看集群节点的状态

rs.status();

### 偷懒办法（建议不要这样偷懒）

如果觉得上面的配置比较繁琐，可以在【主节点】上，一次性将【从节点】【仲裁节点】加入到配置中。

config={\_id:"mongodb30000",members:[{\_id:0,host:"192.168.160.130:27025","priority":20},{\_id:1,host:"192.168.160.130:27026"},{\_id:2,host:"192.168.160.130:27021",arbiterOnly:true}]}

#初始化副本集配置

rs.initiate(config);

确认返回的是{ "ok" : 1 }

#查看集群节点的状态

rs.status();

### 配置某一个从节点永远不会被选为【主节点】，并且是【只读】

#### 连接Primary

./mongo 127.0.0.1:27025/admin -u sa -p

#### 获取当前配置

cfg = rs.conf();

{…..,

"members" : [

{

"\_id" : 2,

"host" : "192.168.160.130:27026",

"arbiterOnly" : false,

"buildIndexes" : true,

"hidden" : false,

"priority" : 1,

"tags" : {

},

"slaveDelay" : NumberLong(0),

"votes" : 1

},...

],…

}

#### 修改想要【只读】的【从节点】的配置，并保存

cfg.members[1].priority=0

cfg.members[1].votes=0

rs.reconfig(cfg);

完成。

这样的话，从节点【192.168.160.130:27026】只读，并且永远不可能选举为【主节点】。

### 从节点上执行js的shell命令，有时候会提示没有权限。

#从节点需要使用下面命令开启slave模式，默认没有开启

#### 连接从节点

./mongo 127.0.0.1:27026/admin -u sa -p

#### 开启slave模式

use admin;

db.getMongo().setSlaveOk(); #推荐使用

然后再当前shell环境下，就可以执行查询命令了。

注意：一旦exit当前环境，下次需要重新开启。