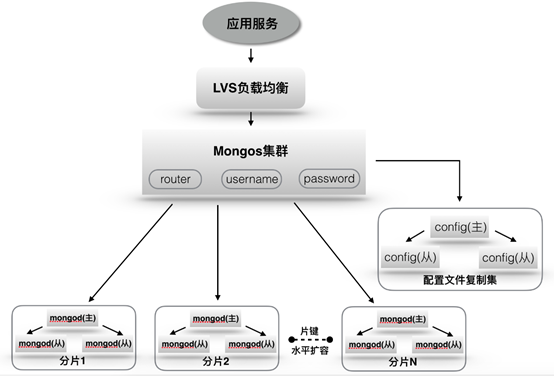
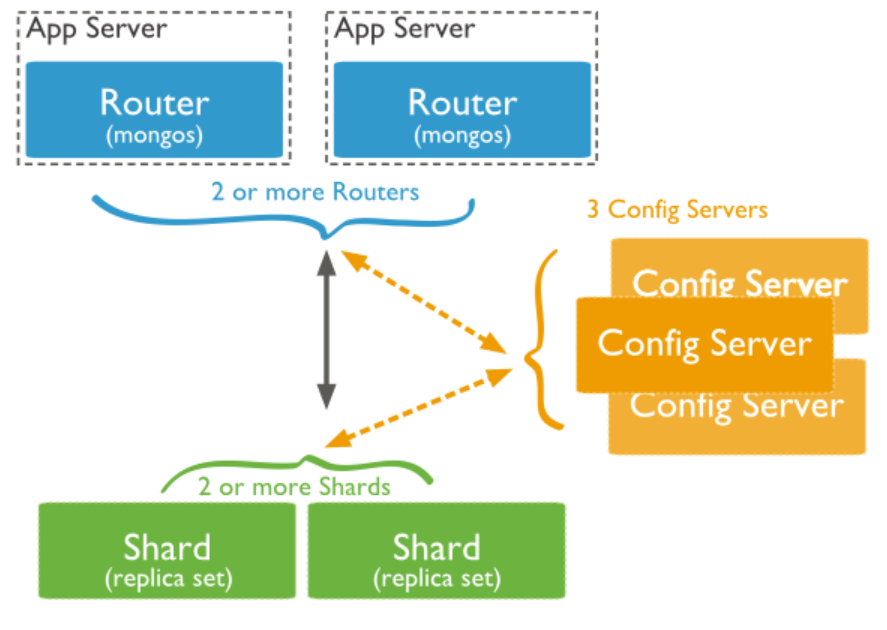
Mongodb分片集群(带认证)安装搭建详细过程

本文主要介绍mongodb分配集群的搭建安装过程，对于初学使用mongodb的同学有一定帮助，本文选取mongodb-3.6.3版本为例进行详细安装说明。

# Mongodb分片集群架构





如上图所示，整个mongodb分片集群由lvs、mongos、mongo-config集群、mongodb复制集组成，其对应的功能如下:

Lvs: 通过vip，提供负载均衡功能，保证后端的Mongodb集群扩容、缩容、数据迁移等过程对业务无感知。

Mongos: 提供路由功能

Mongo-config集群：配置服务器相当于集群的大脑，保存在集群和路由分片的元数据，包括集群有哪些分片、分片的是哪些集合、以及数据块的分布。

Mongodb复制集：用于多个mongod实例之间的数据复制，维护mongod集群的稳定性，实现故障转移，故障切换，故障恢复。

有了该架构，可以很方便的实现增加可用RAM、增加可用磁盘空间、减轻单台服务器的负载、处理单个mongod集群无法承受的吞吐量。

说明:本文只提供mongos+mongo\_cfg集群+mongod集群的安装使用，LVS环节忽略。

以下按照mongod复制集、mongo-config集群、mongos代理的顺序进行安装介绍。其地址信息分别如下：

mongos节点: 127.0.0.1:9000，127.0.0.1:9000

mongo-cfg集群节点: 127.0.0.1:8000，127.0.0.1:8001，127.0.0.1:8002

mongod复制集分片1: 127.0.0.1: 27017，127.0.0.1: 27018，127.0.0.1: 27019

mongod复制集分片2: 127.0.0.1: 27027，127.0.0.1: 27028，127.0.0.1: 27029

# 分片集群(不带认证)安装构建过程

以下为安装过程，本文以mongod

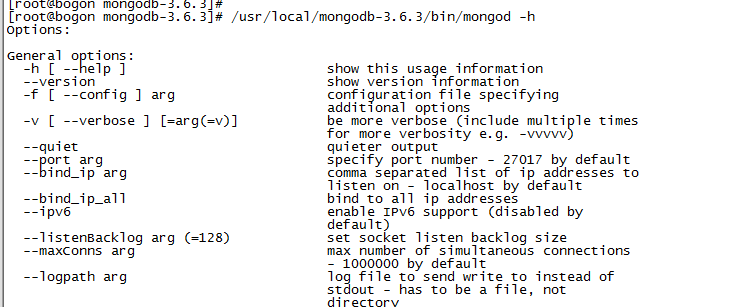
## 2.1下载mongodb安装包

到<https://www.mongodb.org/dl/linux/x86_64-rhel62>

<https://www.mongodb.org/dl/linux/x86_64-rhel70>下载对应的mongodb二进制安装包，然后解压到安装目录，例如本文指定安装目录为/usr/local/，如下图所示:



安装完成后，检查mongo二进制文件是否可以正常运行，如果能正常运行，则会有相应提示，如下:



## 2.2 mongod分片复制集安装过程

### 2.2.1 不带认证分片集群搭建过程

#### 2.2.1.1创建mongo分片复制集节点相关目录

假设数据库信息存放到/home/mongodb目录，则根据下面步骤创建相关的子目录，同时修改配置文件。

cd /home/mongodb

mkdir taxi\_business //代表这个集群是什么业务

cd taxi\_business

mkdir shardcluster1 //mongo赋值集分片1的数据全部存放在这里面，如果有多个分片，则这里还会创建shardcluster2，shardcluster-n，根据自己实际需要创建

cd shardcluster1

mkdir node1 node2 node3 //该复制集分片有3个节点，相关数据分布存入到node1、node2、node3节点中

cd node1

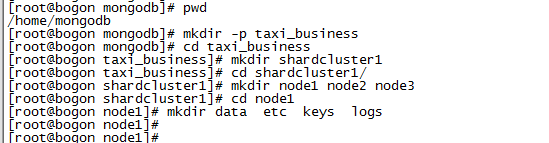
mkdir data etc keys logs //创建node1节点的相关配置、数据、日志、key目录

cd node2

mkdir data etc keys logs

cd node3

mkdir data etc keys logs



#### 2.2.1.2 修改配置mongod文件

processManagement:

fork: true

systemLog:

destination: file

# 指定mongod服务日志文件目录，如果node2则把node1改为node2,类推

path: /home/mongodb/taxi\_business/shardcluster1/node1/logs/mongodb.log

logAppend: true

storage:

journal:

enabled: true

# 指定数据存放的路径，如果node2则把node1改为node2,类推

dbPath: /home/mongodb/taxi\_business/shardcluster1/node1/logs/data/

directoryPerDB: true

engine: wiredTiger #选择存储引擎

wiredTiger:

engineConfig:

cacheSizeGB: 20 #指定存储引擎的cache大小

directoryForIndexes: true

collectionConfig:

blockCompressor: snappy

indexConfig:

prefixCompression: true

net:

port: 27028 #设置mongod监听端口

maxIncomingConnections: 10000 #设置最大连接数

bindIpAll: true

operationProfiling:

slowOpThresholdMs: 100 #设置慢日志时间

mode: slowOp

sharding: #是否支持分片，本集群需要支持分片，因此需要加上配置

clusterRole: shardsvr

archiveMovedChunks: true

replication:

oplogSizeMB: 10240

replSetName: featdoc\_1 #表示这是featdoc集群的第一个分片，该复制集中的所有node节点这个名字要一样，如果是第二个复制集，这里可以取名featdoc\_2

#不带认证需要屏蔽一下配置

#security:

# keyFile: /home/mongodb/taxi\_business/shardcluster1/node1/keys/keyfile

# clusterAuthMode: keyFile

# authorization: enabled

**注意:把该配置文件拷贝到所有mongod集群的etc配置文件中，并根据实际情况修改端口和路径。**

#### 2.2.1.3 启动mongod服务

1. 根据前面的配置启动mongod服务

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/shardcluster1/node1/etc/mongodb.cnf –fork

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/shardcluster1/node2/etc/mongodb.cnf –fork

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/shardcluster1/node3/etc/mongodb.cnf –fork

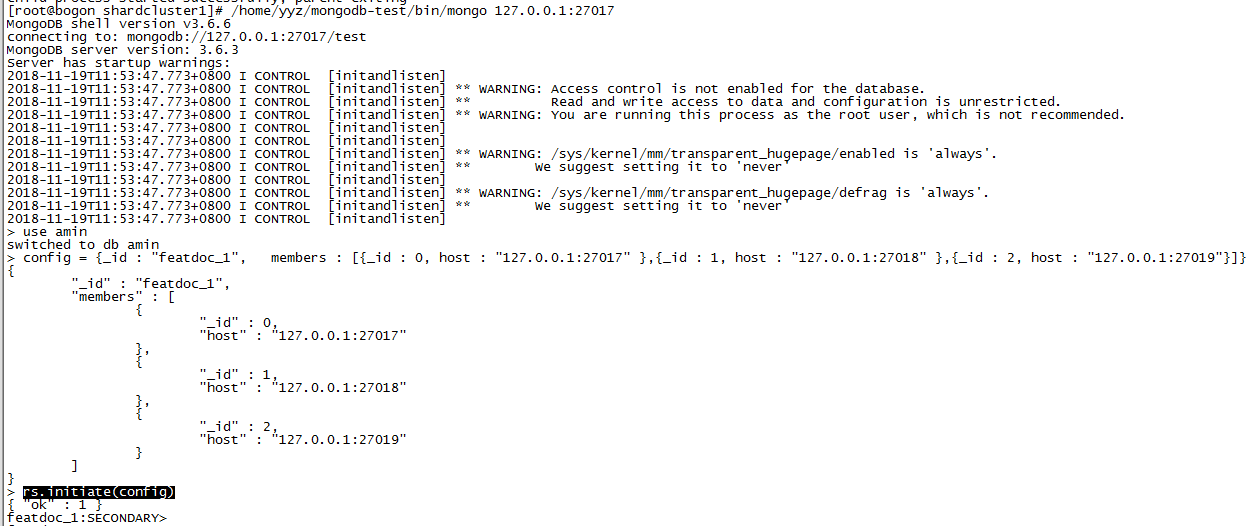
注意: mongodb服务端使用numactl --interleave=all 可以成倍的提高导入速度

#### 2.2.1.4 构建node1 node2 node3到同一个复制集

/usr/local/mongodb-3.6.3/bin/mongo --port 27017

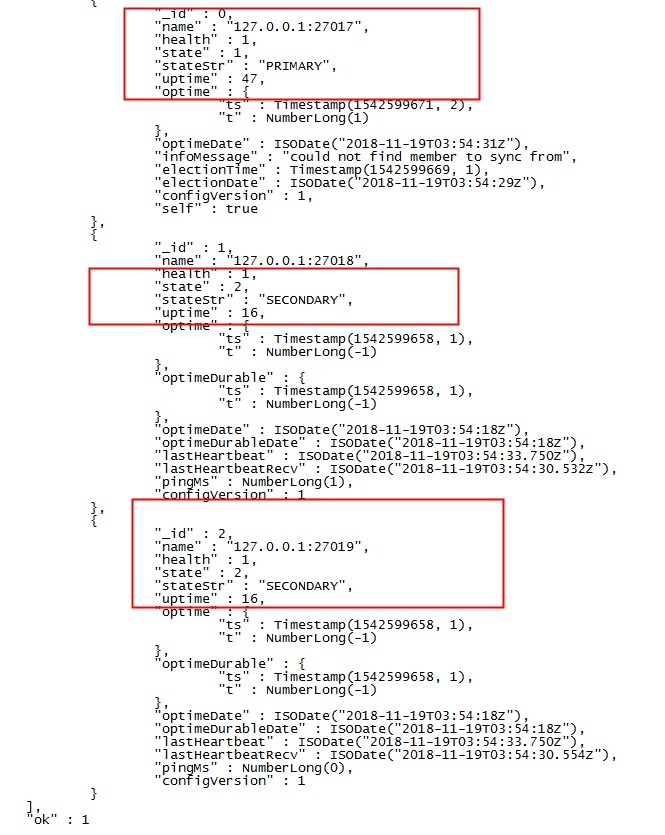
config = {\_id : "featdoc\_1", members : [{\_id : 0, host : "127.0.0.1:27017" },{\_id : 1, host : "127.0.0.1:27018" },{\_id : 2, host : "127.0.0.1:27019"}]}

rs.initiate(config)



Rs.initiate(config)执行后会进行主从选举，选举成功后可以通过rs.status()查看复制集集群状态

rs.status()



然后，通过rs.status()查看集群状态，可以看到27017节点被选举为主节点，该复制集1集群搭建完成。

**如果要搭建其他分片复制集2，过程类似，以此内推。**

### 2.2.2 mongo-cfg不带集群搭建

#### 2.2.2.1 创建mongo-cfg集群相关节点目录

cd /home/mongodb/taxi\_business

mkdir mongocfg\_cluster

mkdir node1 node2 node3

cd node1

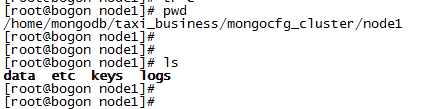
mkdir data etc keys logs

cd ../node2

mkdir data etc keys logs

cd ../node3

mkdir data etc keys logs



#### 2.2.2.2 添加mongo-cfg配置文件

以node1为例，在etc目录下面创建mongocfg.cnf配置文件内容如下：

processManagement:

fork: true

systemLog:

destination: file

#日志路径，node2 node3节点需要把node1改为node2或者node3

path: /home/mongodb/taxi\_business/mongocfg\_cluster/node1/logs/mongodb.log

logAppend: true

storage:

journal:

enabled: true

#数据路径，node2 node3节点需要把node1改为node2或者node3

dbPath: /home/mongodb/taxi\_business/mongocfg\_cluster/node1/data/

directoryPerDB: true

engine: wiredTiger

wiredTiger:

engineConfig:

cacheSizeGB: 20

directoryForIndexes: true

collectionConfig:

blockCompressor: snappy

indexConfig:

prefixCompression: true

net:

port: 8000 #监听的端口

maxIncomingConnections: 10000

bindIpAll: true

operationProfiling:

slowOpThresholdMs: 100

mode: slowOp

sharding:

clusterRole: configsvr #这里是重点，表示该mongod进程是提供mongo-cfg服务

archiveMovedChunks: true

replication:

oplogSizeMB: 10240

replSetName: featdoc #需要和mongos configDB配置中的名字一致

#不认证集群需要屏蔽以下配置

#security:

#key路径，node2 node3节点需要把node1改为node2或者node3

# keyFile: /home/mongodb/taxi\_business/mongocfg\_cluster/node1/keys/keyfile

# clusterAuthMode: keyFile

# authorization: enabled

**说明:**

1. mongo-cfg集群和mongod复制集都是由mongod进程提供服务，他们的配置文件几乎完全一致，唯一的区别是clusterRole角色不一样，复制集配置中角色为shardsvr(表示分片)，mongo-cfg集群角色为configsvr
2. mongo-cfg配置中的replSetName名称必须和mongos configDB配置名称一样
3. 同理，node2和node3中的etc下面的配置文件修改过程类似
4. Mongo-cfg目录etc下面的配置文件最好取名为mongocfg.cnf，这样可以很方便的从进程名就能知道是mongo分片集群还是mongo-cfg集群

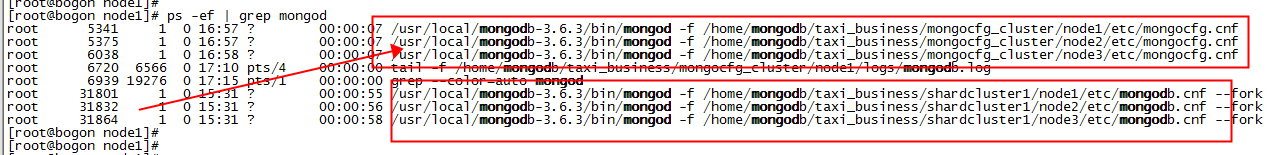
**注意:把该配置文件拷贝到所有mongo-config集群node节点的etc配置文件中，并根据实际情况修改端口和路径。**

#### 2.2.2.3 启动mongo-cfg服务

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/shardcluster1/node1/etc/mongodb.cnf --fork

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/shardcluster1/node2/etc/mongodb.cnf --fork

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/shardcluster1/node3/etc/mongodb.cnf –fork

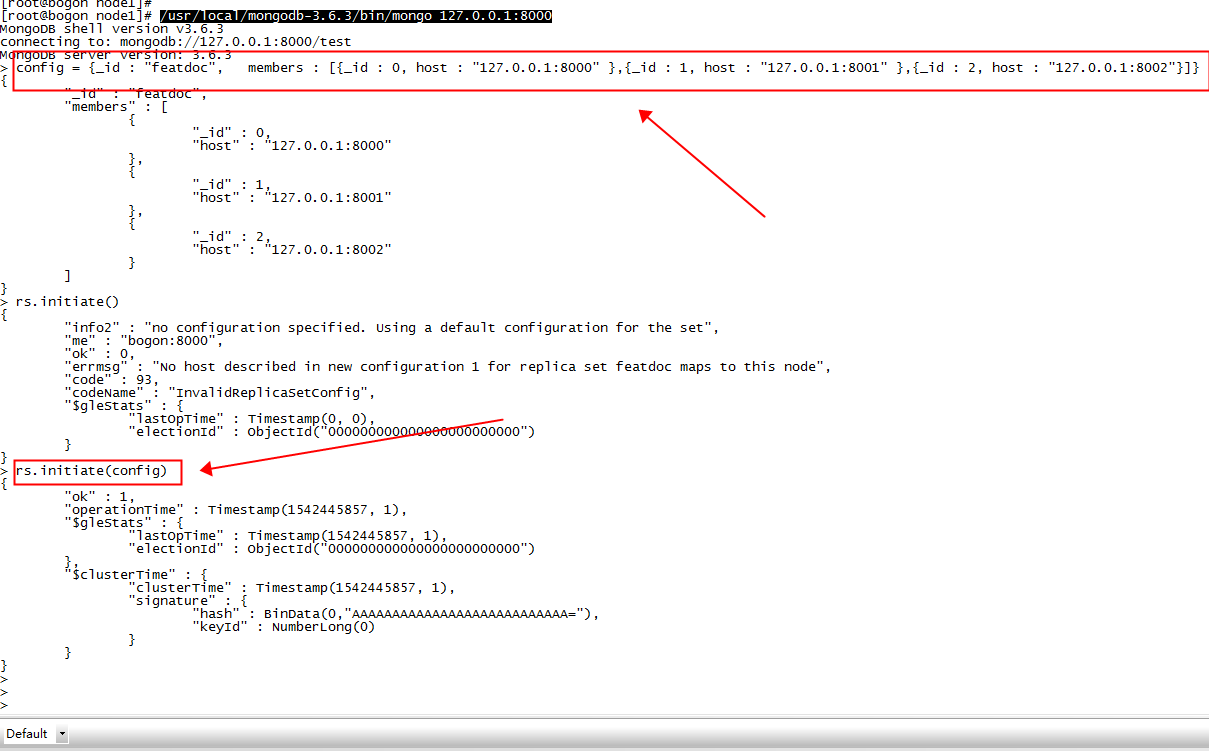


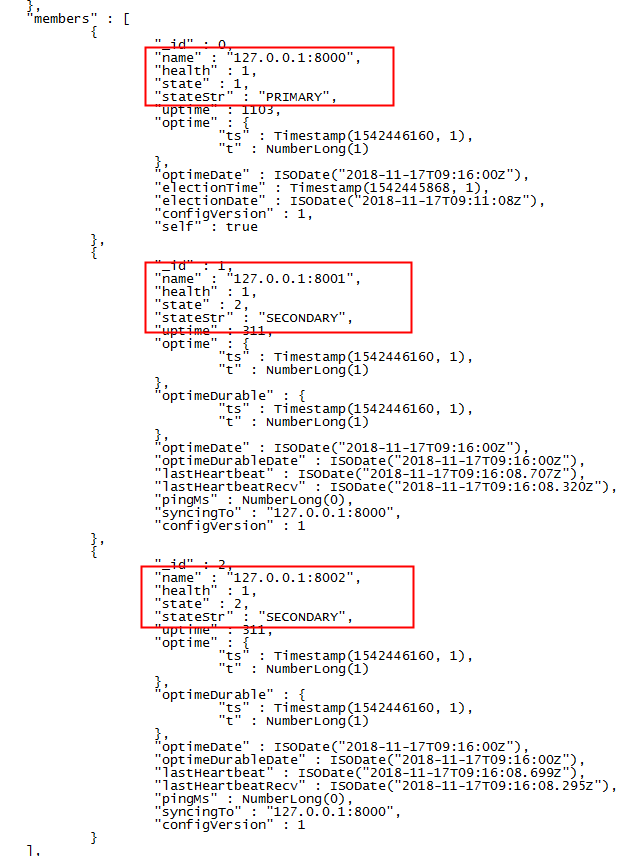
#### 2.2.2.4 构建不带认证的mongo-cfg集群

登录任何一个cfg节点，构建集群：

config = {\_id : "featdoc", members : [{\_id : 0, host : "127.0.0.1:8000" },{\_id : 1, host : "127.0.0.1:8001" },{\_id : 2, host : "127.0.0.1:8002"}]}

rs.initiate(config)





### 2.2.3 mongos不带认证代理搭建

#### 2.2.3.1 创建mongos相关的目录

cd /home/mongodb/taxi\_business

mkdir mongos

cd mongos

mkdir mongos\_1 mongos\_2

cd mongos\_1

mkdir etc keys logs

cd mongos\_2

mkdir etc keys logs

cd /home/mongodb/taxi\_business/mongos/mongos\_1/etc

在该目录创建mongos.cnf

#### 2.2.3.2 添加mongos配置文件

#不带认证需要屏蔽这两行配置

#security:

# keyFile: /home/mongodb/taxi\_business/mongos/mongos\_1/keys/keyfile

systemLog:

destination: file

logAppend: true

path: /home/mongodb/taxi\_business/mongos/mongos\_1/logs/mongos.log

processManagement:

fork: true

pidFilePath: /home/mongodb/taxi\_business/mongos/mongos\_1/mongos.pid

net:

port: 9000 #端口

maxIncomingConnections: 20000

sharding:

# 这里的的featdoc必须和mongs configDB配置名称一致

# 这里的三个地址为mongo-cfg集群的地址

configDB: featdoc/127.0.0.1:8000,127.0.0.1:8001,127.0.0.1:8002

#### 2.2.3.3 启动mongos服务

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongos -f /home/mongodb/taxi\_business/mongos/mongos\_1/etc/mongos.cnf

#### 2.2.3.4 给mongos代理添加分片信息

sh.addShard("featdoc\_1/127.0.0.1:27017,127.0.0.1:27018,127.0.0.1:27019")

### 2.2.3 mongos不带认证代理搭建

#### 2.2.3.1 创建mongos相关的目录

cd /home/mongodb/taxi\_business

mkdir mongos

cd mongos

mkdir mongos\_1 mongos\_2

cd mongos\_1

mkdir etc keys logs

cd mongos\_2

mkdir etc keys logs

cd /home/mongodb/taxi\_business/mongos/mongos\_1/etc

在该目录创建mongos.cnf

#### 2.2.3.2 添加mongos配置文件

#不带认证需要屏蔽这两行配置

#security:

# keyFile: /home/mongodb/taxi\_business/mongos/mongos\_1/keys/keyfile

systemLog:

destination: file

logAppend: true

path: /home/mongodb/taxi\_business/mongos/mongos\_1/logs/mongos.log

processManagement:

fork: true

pidFilePath: /home/mongodb/taxi\_business/mongos/mongos\_1/mongos.pid

net:

port: 9000 #端口

maxIncomingConnections: 20000

sharding:

# 这里的的featdoc必须和mongs configDB配置名称一致

# 这里的三个地址为mongo-cfg集群的地址

configDB: featdoc/127.0.0.1:8000,127.0.0.1:8001,127.0.0.1:8002

#### 2.2.3.3 启动mongos服务

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongos -f /home/mongodb/taxi\_business/mongos/mongos\_1/etc/mongos.cnf

#### 2.2.3.4 给mongos代理添加分片信息

sh.addShard("featdoc\_1/127.0.0.1:27017,127.0.0.1:27018,127.0.0.1:27019")



注意: featdoc\_1必须和该分片对应的复制集的replSetName名字一致

**如果要再搭建一个mongos，搭建步骤类似**

# 分片集群(带认证)安装构建过程

接前面的第2章节，带认证分片集群搭建过程，首先需要搭建好不带认证分片集群(mongos+mongo-cfg集群+mongo复制集集群)，参考前面。

然后为mongos和mongod集群添加账号信息，添加成功后退出服务，然后修改配置加上认证配置，重新启动服务即可，注意退出服务顺序为:

1. stop mongos
2. stop mongod
3. Stop mongo-cfg

重启服务的启动顺序为:

1. start mongo-cfg
2. start mongod
3. start mongos

## 3.1 为不带认证的mongo复制集和mongos添加账号

1. 登录mongo复制集集群主节点，为mongo复制集集群添加账号

/home/yyz/mongodb-test/bin/mongo 127.0.0.1:27017

use admin

db.createUser({user: "root", pwd: "123456", roles: [{role: "root", db: "admin"}]});

db.createUser( { user: "admin", pwd: "123456", roles: [{role: "userAdminAnyDatabase", db: "admin"}]});

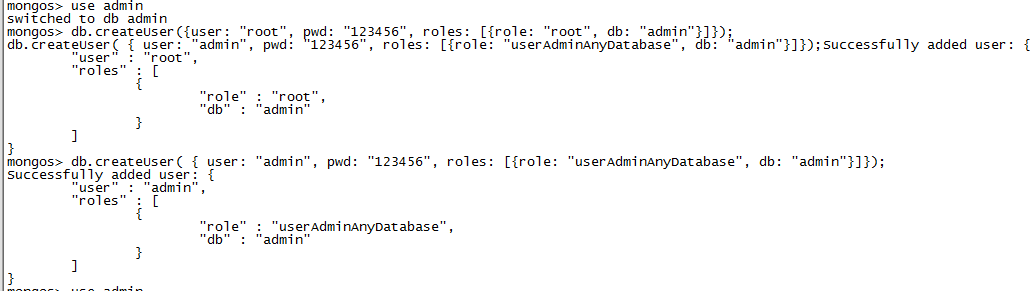
2. 为mongos代理添加账号

/home/yyz/mongodb-test/bin/mongo 127.0.0.1:9000

use admin

db.createUser({user: "root", pwd: "123456", roles: [{role: "root", db: "admin"}]});

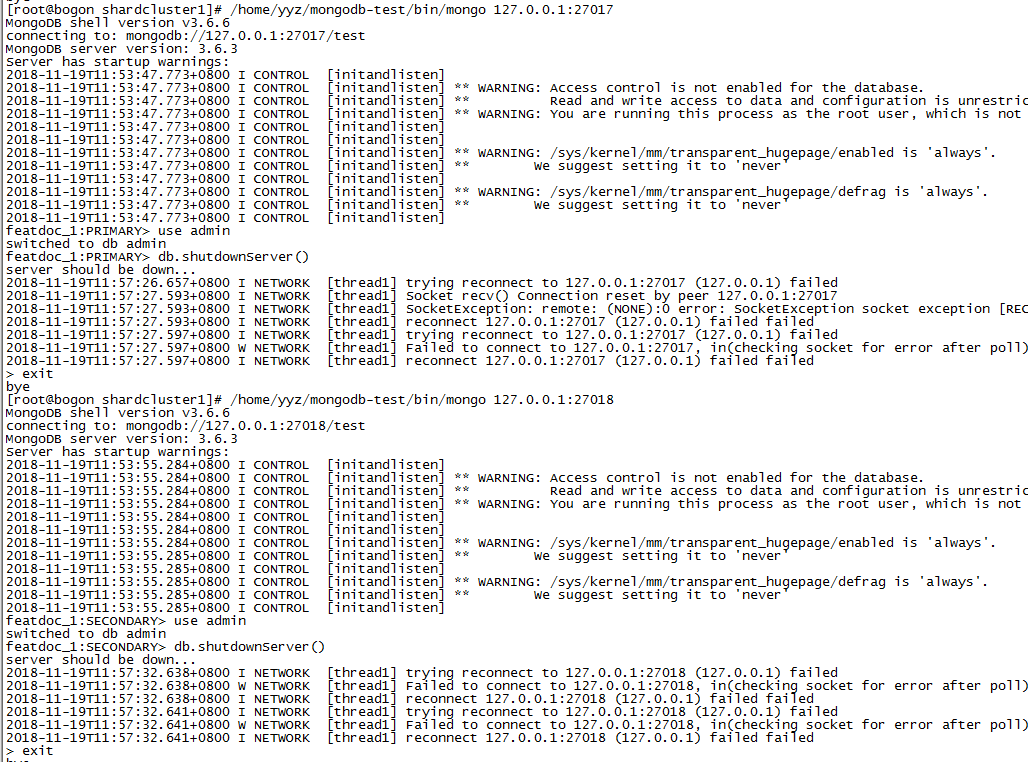
db.createUser( { user: "admin", pwd: "123456", roles: [{role: "userAdminAnyDatabase", db: "admin"}]});



## 3.2 shutdown关闭mongos、mongod集群、mongo-cfg

依次登录mongos、mongod节点、mongo-cfg节点，然后执行:  
use admin

db.shutdownServer()



## 3.3 生成key文件，并拷贝到所有node节点的keys目录

1. 生成key文件，并修改权限

openssl rand -base64 666 > keyfile

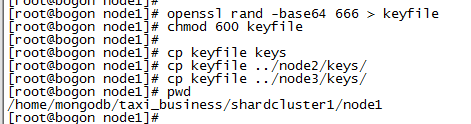
chmod 600 keyfile

2.拷贝keyfile到mongos、mongod集群节点、mongo-cfg集群节点的相应目录中

cp keyfile keys

cp keyfile ../node2/keys/

cp keyfile ../node3/keys/



## 3.4 修改配置文件，在mongod和mongo-cfg的配置文件中，使能认证配置

1. mongo-cfg和mongod集群配置中，把以下配置加上

security:

#key路径，node2 node3节点需要把node1改为node2或者node3

keyFile: /home/mongodb/taxi\_business/mongocfg\_cluster/node1/keys/keyfile

clusterAuthMode: keyFile

authorization: enabled

2. 在mongos配置文件中，把以下配置加上:

security:

keyFile: /home/mongodb/taxi\_business/mongos/mongos\_1/keys/keyfile

## 3.4 依次重启mongo-cfg集群、mongo复制集集群、mongos代理

1. 启动mong-cfg集群实例

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/mongocfg\_cluster/node1/etc/mongocfg.cnf

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/mongocfg\_cluster/node2/etc/mongocfg.cnf

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/mongocfg\_cluster/node3/etc/mongocfg.cnf

2.启动mongo复制集集群实例

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/shardcluster1/node1/etc/mongodb.cnf

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/shardcluster1/node2/etc/mongodb.cnf

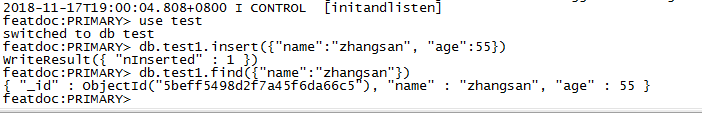
numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongod -f /home/mongodb/taxi\_business/shardcluster1/node3/etc/mongodb.cnf

3.启动mongos代理实例

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongos -f /home/mongodb/taxi\_business/mongos/mongos\_1/etc/mongos.cnf

numactl --interleave=all /usr/local/mongodb-3.6.3/bin/mongos -f /home/mongodb/taxi\_business/mongos/mongos\_2/etc/mongos.cnf

## 3.5带认证分片集群测试

通过上面的步骤整个分片集群已经安装完毕，进行简单测试： 

插入后查找，查找成功，说明带认证分片集群搭建完成。