# docker-compose部署mongodb分片集群（2个单分片）

## 一 前期准备

### 目录结构

# tree

.

├── conf

│   ├── configsvr1

│   │   ├── config.conf

│   │   └── key.file

│   ├── configsvr2

│   │   ├── config.conf

│   │   └── key.file

│   ├── configsvr3

│   │   ├── config.conf

│   │   └── key.file

│   ├── mongos

│   │   ├── key.file

│   │   └── mongos.conf

│   ├── shard1

│   │   ├── key.file

│   │   └── shard.conf

│   ├── shard2

│   │   ├── key.file

│   │   └── shard.conf

│   └── shard3

│   ├── key.file

│   └── shard.conf

├── data

│   ├── configsvr1

│   ├── configsvr2

│   ├── configsvr3

│   ├── shard1

│   └── shard2

├── del.sh

├── docker-compose.yml

├── log

│   ├── configsvr1

│   │   └── config.log

│   ├── configsvr2

│   │   └── config.log

│   ├── configsvr3

│   │   └── config.log

│   ├── mongos

│   │   ├── mongos.diagnostic.data

│   │   │   ├── metrics.2022-10-10T07-03-55Z-00000

│   │   │   └── metrics.interim

│   │   └── mongos.log

│   ├── shard1

│   │   └── shard.log

│   └── shard2

│   └── shard.log

└── mkdir.sh

### 创建网络

docker network create --subnet=10.20.0.0/24 mongo-network

### docker-compose.yml

version: '3.3'  
networks:   
 fzmtr4-test-network:  
 external: true  
services:  
 # 配置服务器configsvr  
 fzmtr4\_test\_mongo\_cfgsvr1:  
 image: mongo:5.0  
 networks:  
 - fzmtr4-test-network  
 container\_name: fzmtr4\_test\_mongo\_cfgsvr1  
 restart: always  
 ports:   
 - 37019:27019  
 command: mongod --config /etc/mongo/config.conf --logpath /log/configsvr/config.log  
 volumes:  
 - /etc/localtime:/etc/localtime:ro  
 - ${PWD}/data/configsvr1/:/data/configdb/  
 - ${PWD}/log/configsvr1/:/log/configsvr/  
 - ${PWD}/conf/configsvr1:/etc/mongo/  
   
 fzmtr4\_test\_mongo\_cfgsvr2:  
 image: mongo:5.0  
 networks:  
 - fzmtr4-test-network  
 container\_name: fzmtr4\_test\_mongo\_cfgsvr2  
 restart: always  
 ports:  
 - 38019:27019  
 command: mongod --config /etc/mongo/config.conf --logpath /log/configsvr/config.log  
 volumes:  
 - /etc/localtime:/etc/localtime:ro  
 - ${PWD}/data/configsvr2/:/data/configdb/  
 - ${PWD}/log/configsvr2/:/log/configsvr/  
 - ${PWD}/conf/configsvr2:/etc/mongo/  
  
 fzmtr4\_test\_mongo\_cfgsvr3:  
 image: mongo:5.0  
 networks:  
 - fzmtr4-test-network  
 container\_name: fzmtr4\_test\_mongo\_cfgsvr3  
 restart: always  
 ports:  
 - 39019:27019  
 command: mongod --config /etc/mongo/config.conf --logpath /log/configsvr/config.log  
 volumes:  
 - /etc/localtime:/etc/localtime:ro  
 - ${PWD}/data/configsvr3/:/data/configdb/  
 - ${PWD}/log/configsvr3/:/log/configsvr/  
 - ${PWD}/conf/configsvr3:/etc/mongo/  
  
  
#分片1  
 fzmtr4\_test\_mongo\_shard1:  
 image: mongo:5.0  
 networks:  
 - fzmtr4-test-network  
 container\_name: fzmtr4\_test\_mongo\_shard1  
 restart: always  
 ports:  
 - 37018:27018  
 command: mongod --config /etc/mongo/shard.conf --logpath /log/shardsvr/shard.log  
 volumes:  
 - /etc/localtime:/etc/localtime:ro  
 - ${PWD}/data/shard1:/data/db/  
 - ${PWD}/log/shard1:/log/shardsvr/  
 - ${PWD}/conf/shard1:/etc/mongo/  
 #- ${PWD}/init:/init/  
 depends\_on:  
 - fzmtr4\_test\_mongo\_cfgsvr1  
 - fzmtr4\_test\_mongo\_cfgsvr2  
 - fzmtr4\_test\_mongo\_cfgsvr3  
  
  
#分片2  
 fzmtr4\_test\_mongo\_shard2:  
 image: mongo:5.0  
 networks:  
 - fzmtr4-test-network  
 container\_name: fzmtr4\_test\_mongo\_shard2  
 restart: always  
 ports:  
 - 47018:27018  
 command: mongod --config /etc/mongo/shard.conf --logpath /log/shardsvr/shard.log  
 volumes:  
 - /etc/localtime:/etc/localtime:ro  
 - ${PWD}/data/shard2:/data/db/  
 - ${PWD}/log/shard2:/log/shardsvr/  
 - ${PWD}/conf/shard2:/etc/mongo/  
 #- ${PWD}/init:/init/  
 depends\_on:  
 - fzmtr4\_test\_mongo\_cfgsvr1  
 - fzmtr4\_test\_mongo\_cfgsvr2  
 - fzmtr4\_test\_mongo\_cfgsvr3  
  
  
  
#mongos  
 fzmtr4\_test\_mongo\_router:  
 image: mongo:5.0  
 container\_name: fzmtr4\_test\_mongo\_router  
 command: mongos --config /etc/mongo/mongos.conf --logpath /log/mongos/mongos.log  
 #command: mongos --config /etc/mongo/mongos.conf  
 volumes:  
 - /etc/localtime:/etc/localtime:ro  
 - ${PWD}/log/mongos/:/log/mongos/  
 - ${PWD}/conf/mongos/:/etc/mongo/  
 ports:  
 - "37017:27017"  
 depends\_on:  
 - fzmtr4\_test\_mongo\_shard1  
 - fzmtr4\_test\_mongo\_shard2

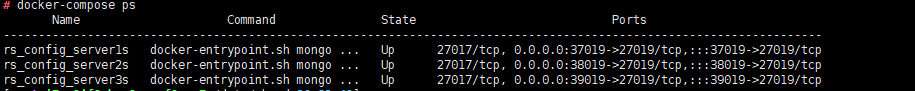
## 二 配置config-server

### 1 mongo.conf

sharding:  
 clusterRole: configsvr  
  
replication:  
 replSetName: rs\_configsvr  
  
storage:  
 dbPath: "/data/db"  
 # 关于存储引擎wiredTiger的参数设置  
 # 参考：https://docs.mongodb.com/manual/reference/configuration-options/#storage-options  
 wiredTiger:  
 engineConfig:  
 # cacheSizeGB: <number>  
 journalCompressor: "zstd"  
 collectionConfig:  
 blockCompressor: "zstd"  
  
security:  
 authorization: enabled  
 clusterAuthMode: "keyFile"  
 keyFile: "/etc/mongo/key.file"  
  
systemLog:  
 verbosity: 0  
 quiet: false  
 traceAllExceptions: false  
 destination: "file"  
 logAppend: true  
 logRotate: reopen  
  
net:  
 bindIpAll: true  
 port: 27019

### 2 启动mongo-config-server

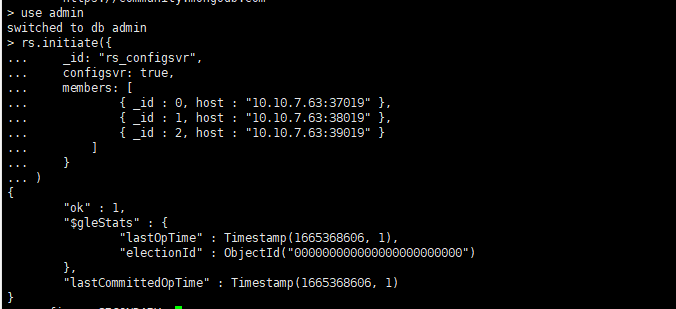
docker-compose up -d fzmtr4\_test\_mongo\_cfgsvr1 fzmtr4\_test\_mongo\_cfgsvr2 fzmtr4\_test\_mongo\_cfgsvr3



### 3 初始化配置服务复制集

docker exec -it fzmtr4\_test\_mongo\_cfgsvr1 mongo 127.0.0.1:27019

use admin  
rs.initiate({  
 \_id: "rs\_configsvr",  
 configsvr: true,  
 members: [  
 { \_id : 0, host : "10.10.7.63:37019" },  
 { \_id : 1, host : "10.10.7.63:38019" },  
 { \_id : 2, host : "10.10.7.63:39019" }  
 ]  
 }  
)



#查看状态rs.status()

## 三 配置shard

### 1 shard1.conf

sharding:  
 clusterRole: shardsvr  
replication:  
 replSetName: rs\_shardsvr1  
  
storage:  
 dbPath: "/data/db"  
 # 关于存储引擎wiredTiger的参数设置  
 # 参考：https://docs.mongodb.com/manual/reference/configuration-options/#storage-options  
 wiredTiger:  
 engineConfig:  
 # cacheSizeGB: <number>  
 journalCompressor: "zstd"  
 collectionConfig:  
 blockCompressor: "zstd"  
  
security:  
 authorization: enabled  
 clusterAuthMode: "keyFile"  
 keyFile: "/etc/mongo/key.file"  
 # transitionToAuth: <boolean>  
 # javascriptEnabled: <boolean>  
  
systemLog:  
 verbosity: 0  
 quiet: false  
 traceAllExceptions: false  
 destination: "file"  
 logAppend: true  
 logRotate: reopen  
  
net:  
 bindIpAll: true  
 port: 27018

### 2 shard2.conf

sharding:  
 clusterRole: shardsvr  
replication:  
 replSetName: rs\_shardsvr2  
  
storage:  
 dbPath: "/data/db"  
 # 关于存储引擎wiredTiger的参数设置  
 # 参考：https://docs.mongodb.com/manual/reference/configuration-options/#storage-options  
 wiredTiger:  
 engineConfig:  
 # cacheSizeGB: <number>  
 journalCompressor: "zstd"  
 collectionConfig:  
 blockCompressor: "zstd"  
  
security:  
 authorization: enabled  
 clusterAuthMode: "keyFile"  
 keyFile: "/etc/mongo/key.file"  
 # transitionToAuth: <boolean>  
 # javascriptEnabled: <boolean>  
  
systemLog:  
 verbosity: 0  
 quiet: false  
 traceAllExceptions: false  
 destination: "file"  
 logAppend: true  
 logRotate: reopen  
  
net:  
 bindIpAll: true  
 port: 27018

### 3 启动shard

docker-compose up -d fzmtr4\_test\_mongo\_shard1 fzmtr4\_test\_mongo\_shard2

### 

### 4 初始化分片服务器

docker exec -it fzmtr4\_test\_mongo\_shard1 mongo 127.0.0.1:27018

use admin  
rs.initiate(  
 {  
 \_id : "rs\_shardsvr1",  
 members: [  
 { \_id : 0, host : "192.168.0.155:37018"}  
 ]  
 }  
)

docker exec -it fzmtr4\_test\_mongo\_shard2 mongo 127.0.0.1:27018

use admin  
rs.initiate(  
 {  
 \_id : "rs\_shardsvr2",  
 members: [  
 { \_id : 0, host : "192.168.0.155:47018"}  
 ]  
 }  
)

## 四 配置mongos路由

### 1 mongos.conf

systemLog:  
 verbosity: 0  
 quiet: false  
 traceAllExceptions: true  
 destination: "file"  
 logAppend: true  
 logRotate: reopen  
  
security:  
 clusterAuthMode: "keyFile"  
 keyFile: "/etc/mongo/key.file"  
 # authorization: disabled #该配置项不支持mongos，仅支持mongod  
  
net:  
 bindIpAll: true  
 port: 27017  
  
sharding:  
 #定义为mongos配置服务器  
 configDB: rs\_configsvr/192.168.0.155:37019,192.168.0.155:38019,192.168.0.155:39019

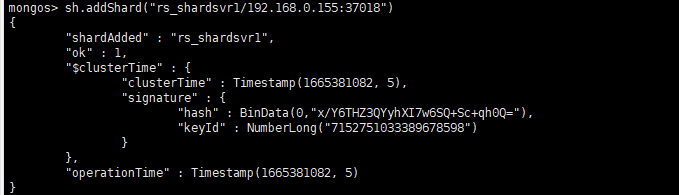
### 2 启动mongos

docker-compose up -d fzmtr4\_test\_mongo\_router

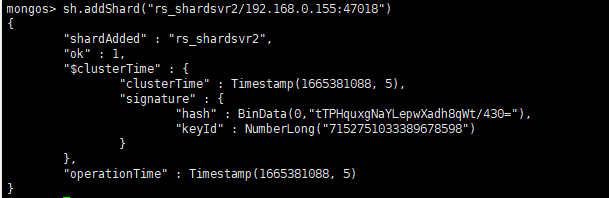
### 

### 3 初始化mongos

docker exec -it fzmtr4\_test\_mongo\_router mongo 127.0.0.1:27017  
mongos> use admin  
mongos> sh.addShard("rs\_shardsvr1/192.168.0.155:37018")



mongos> sh.addShard("rs\_shardsvr2/192.168.0.155:47018")



### 4 创建系统管理员账号

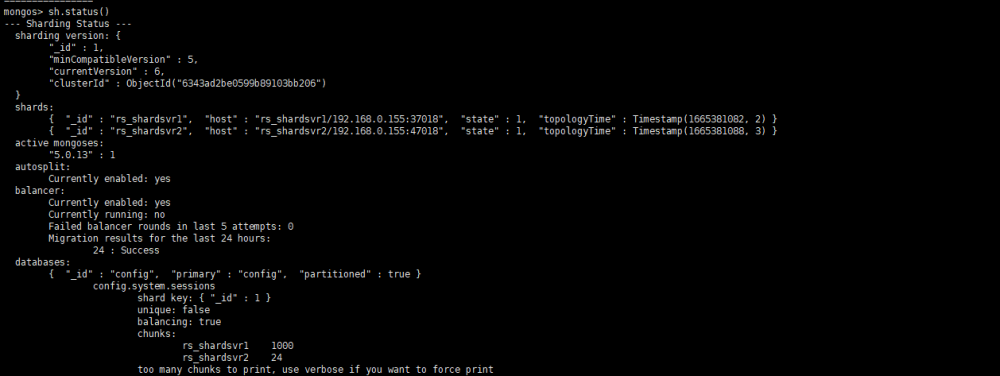
db.createUser({user:"system",pwd:"fzmtr4123456",roles:[{role:"root",db:"admin"}]})

## 五 验证

### 1 登陆mongos

docker exec -it fzmtr4\_test\_mongo\_router mongo 127.0.0.1:27017 -u system -p fzmtr4123456 --authenticationDatabase admin

### 2 查看集群状态



### 3 开启分片功能

mongos> use admin  
mongos> db.runCommand({"enablesharding":"testdb2"})  
mongos> db.runCommand({"shardcollection":"testdb2.person","key":{\_id:'hashed'}})

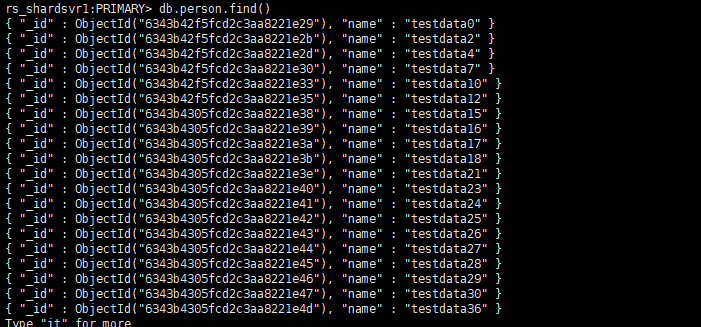
### 4 添加数据

mongos> use testdb2  
mongos> for(var i=0;i<100;i++){db.person.insert({name:"testdata"+i});}

### 5 查看数据

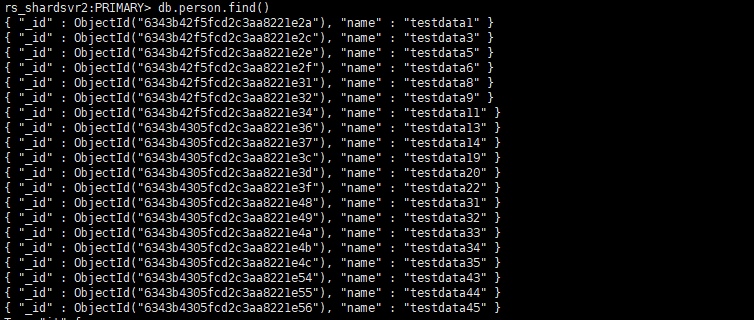
shard1查看分片数据

$ docker exec -it fzmtr4\_test\_mongo\_shard1 mongo 127.0.0.1:27018  
rs\_shardsvr1:PRIMARY> use admin  
rs\_shardsvr1:PRIMARY> db.createUser({user:"system",pwd:"fzmtr4123456",roles:[{role:"root",db:"admin"}]})  
rs\_shardsvr1:PRIMARY> db.auth("system","fzmtr4123456")  
rs\_shardsvr1:PRIMARY> use testdb2  
rs\_shardsvr1:PRIMARY> db.person.find()



shard2查看分片数据

$ docker exec -it fzmtr4\_test\_mongo\_shard2 mongo 127.0.0.1:27018  
rs\_shardsvr2:PRIMARY> use admin  
rs\_shardsvr2:PRIMARY> db.createUser({user:"system",pwd:"fzmtr4123456",roles:[{role:"root",db:"admin"}]})  
rs\_shardsvr2:PRIMARY> db.auth("system","fzmtr4123456")  
rs\_shardsvr2:PRIMARY> use testdb2  
rs\_shardsvr2:PRIMARY> db.person.find()



由以上数据可知，数据分散在shard1,shard2上