# mongo部署和备份恢复

此文档包含MongoDB单实例部署、数据的备份及恢复、MongoDB集群部署三部分。

# 单实例部署

## 下载安装

1、官网下载地址：<https://www.mongodb.com/>

网盘下载链接：https://pan.baidu.com/s/1lqP4TJ42fMMU-HeX0UM33Q

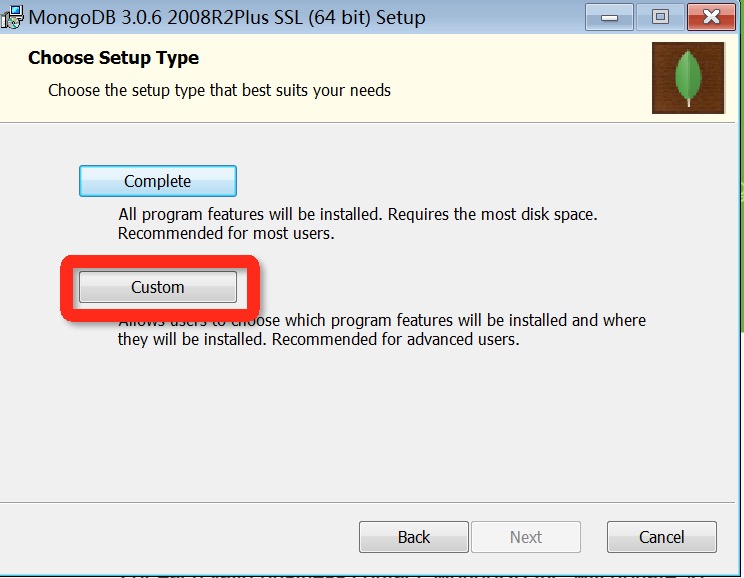
提取码：jp7c

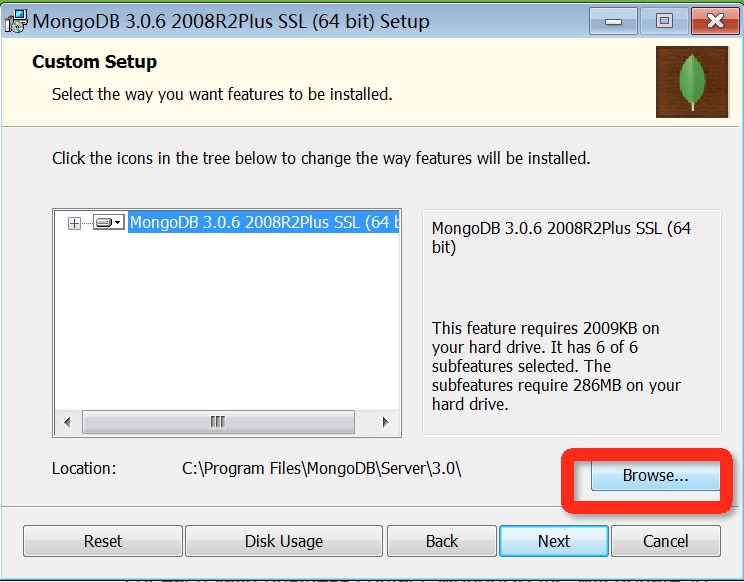
2、安装教程

安装手册：<https://www.runoob.com/mongodb/mongodb-window-install.html>

根据你的系统下载 32 位或 64 位的 .msi 文件，下载后双击该文件，按操作提示安装即可。

安装过程中，你可以通过点击 "Custom(自定义)" 按钮来设置你的安装目录。





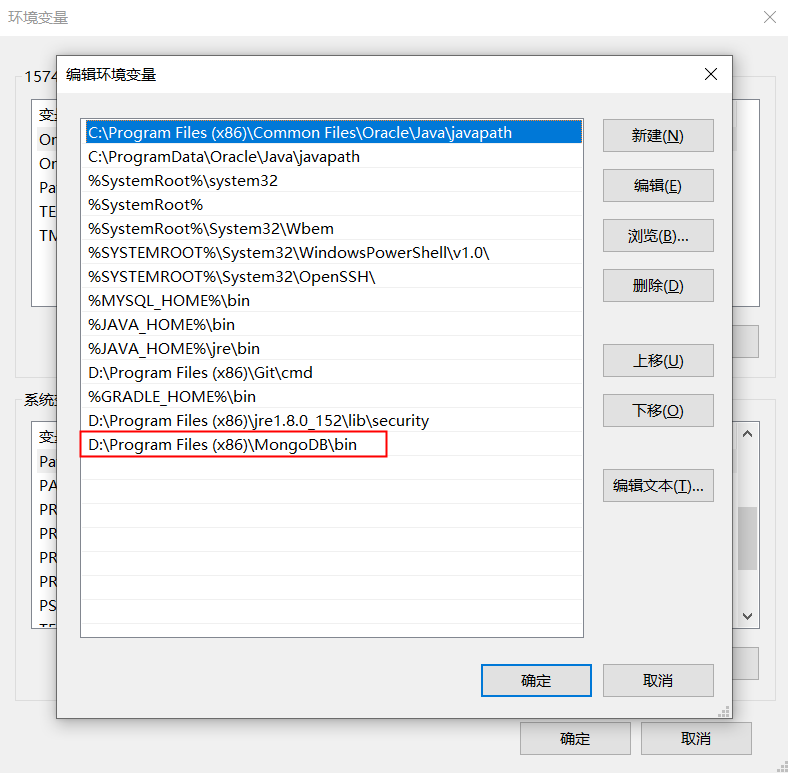
下一步安装 **"install mongoDB compass"** 不勾选（当然你也可以选择安装它，可能需要更久的安装时间），MongoDB Compass 是一个图形界面管理工具，我们可以在后面自己到官网下载安装，下载地址：<https://www.mongodb.com/download-center/compass>。



## MongoDB配置

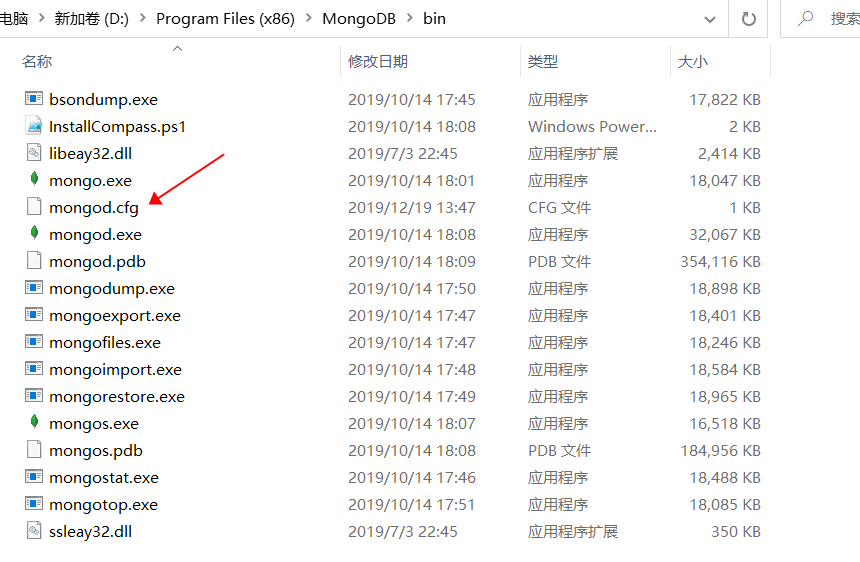
1. 配置环境变量

我的电脑—右击属性—高级系统设置—环境变量—Path—编辑，如下图：



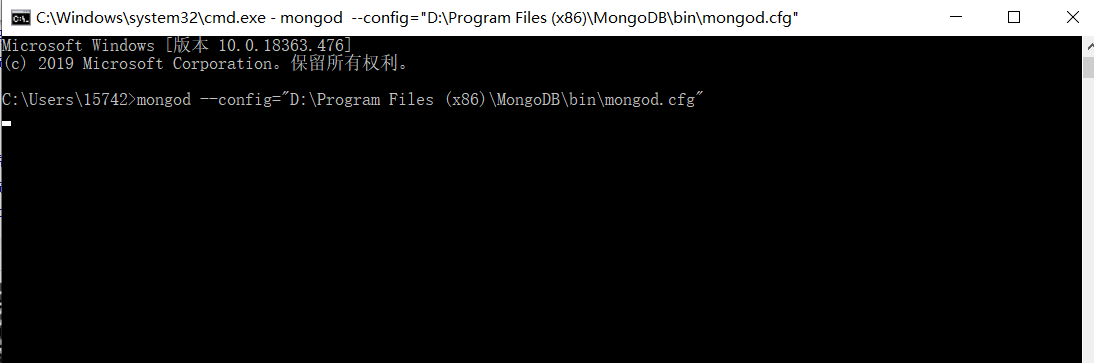
1. 启动mongo实例

安装成功之后bin目录下有一个默认的配置文件

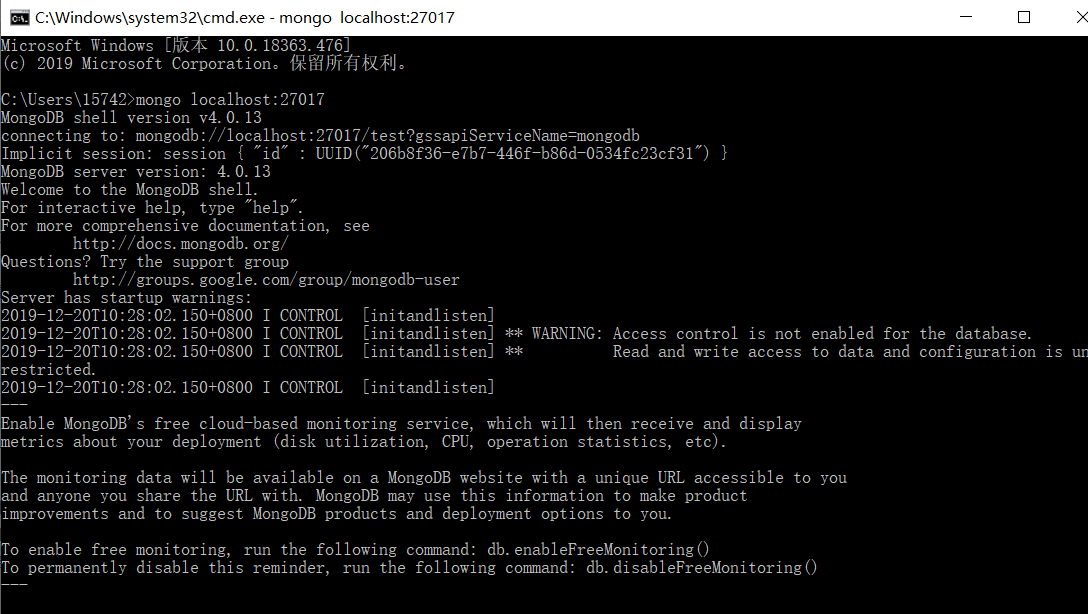


MongoDB启动服务的时候会默认读取该配置文件启动，配置了环境变量后，启动cmd运行命令：**mongod --config=“配置文件路径”。**其中配置文件路径为: D:\Program Files (x86)\MongoDB\bin\mongod.cfg

来启动一个mongod实例，如下图：



1. 创建用户和数据库
2. 启动完成后，另开一个窗口连接（ip默认是localhost）：mongo localhost:27017



1. 连接成功后，创建/切换数据库。foshan\_db为数据库名称，命令如下：

use foshan\_db

1. 然后为foshan\_db数据库创建一个管理员账户，角色为dbOwner：

db.createUser(

    {

        user: "augur",

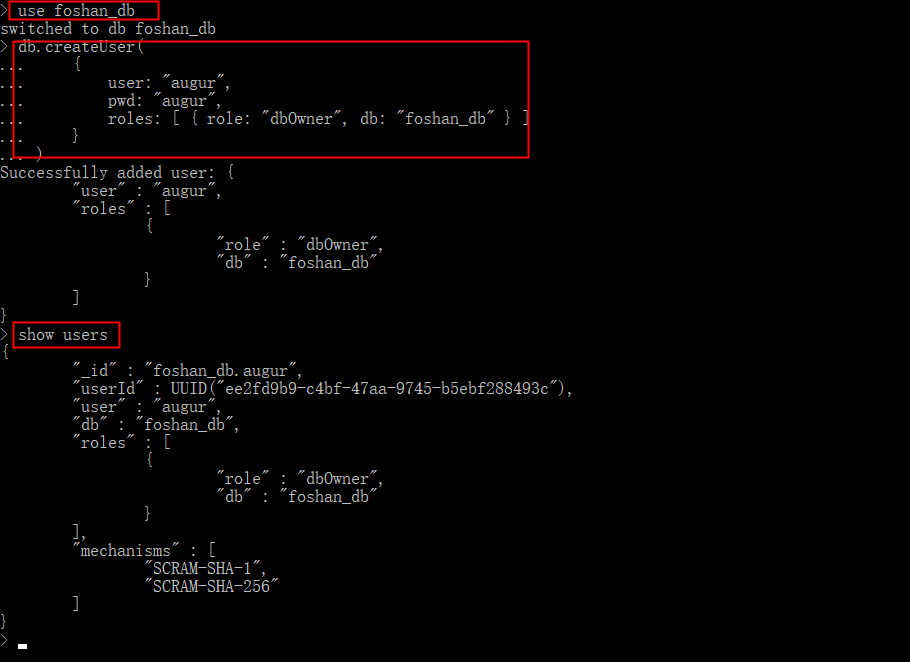
        pwd: "augur",

        roles: [ { role: "dbOwner", db: "foshan\_db" } ]

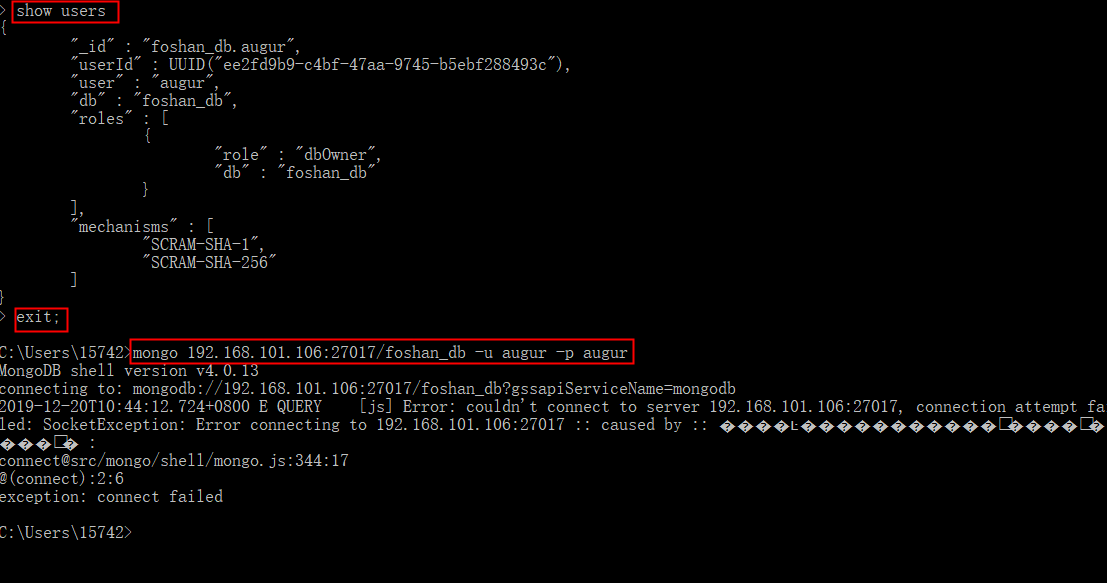
    }

)

1. 创建成功后可以通过命令：show users查看用户信息

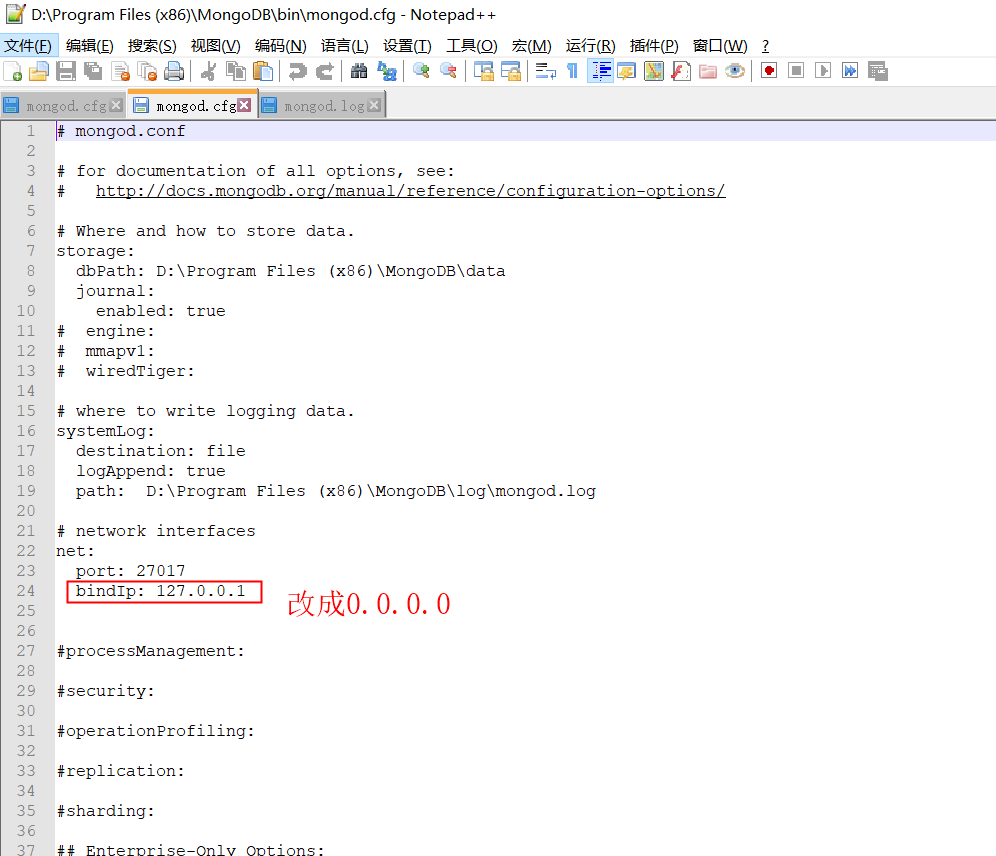


1. 用账户和密码连接foshan\_db数据库



此时会报错，是因为ip跟配置文件mongod.cfg的bindIP不一致，用127.0.0.1可以连接成功。开放所有端口连接，将bindIP改成0.0.0.0

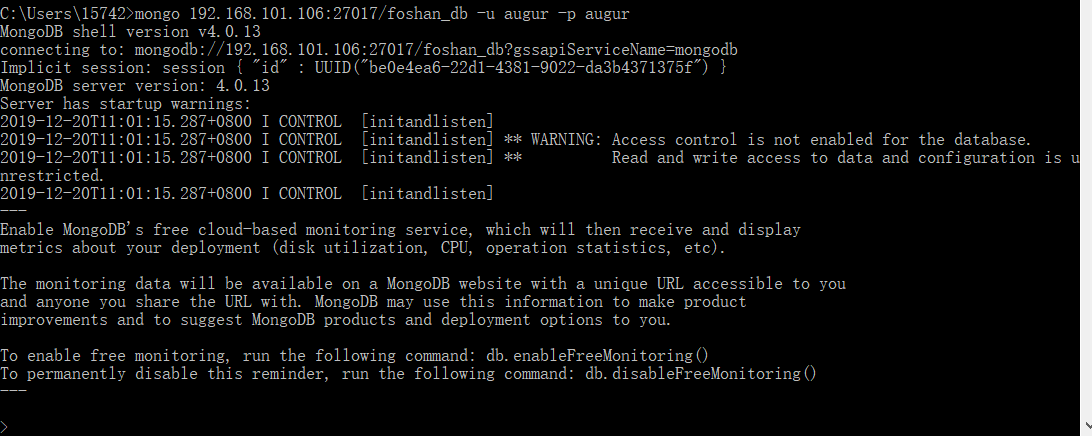
关闭mongod实例窗口。修改配置文件，如下图：



再次启动实例：

mongod --config="C:\Program Files\MongoDB\Server\4.0\bin\mongod.cfg"

实例启动后，连接数据库：

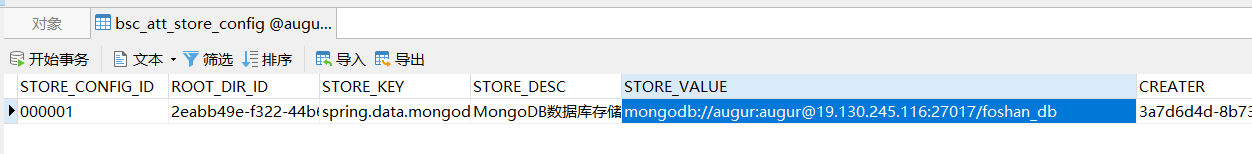


1. 数据库配置

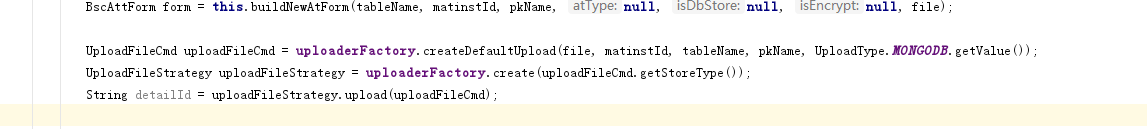
将账户连接信息配置到数据库表：bsc\_att\_store\_config中，更改bsc\_att\_store\_config表的store\_value字段

mongodb://augur:augur@19.130.245.116:27017/foshan\_db

注：mongodb://用户名:密码@ip:端口/数据库名称



上传文件的时候通过注入的UploaderFactory对象，根据相关业务信息创建一个文件上传命令对象UploadFileCmd，然后通过uploaderFactory的create方法获取到对象文件上传实现类对象，最后调用upload方法传入文件命令对象即可完成上传。上传成功返回一个detailId。



# 数据备份

参考地址：<https://www.runoob.com/mongodb/mongodb-mongodump-mongorestore.html>

1. 备份命令说明

使用mongodump命令来备份MongoDB数据。该命令可以导出所有数据到指定目录中

mongodump -h dbhost -d dbname -o dbdirectory

* **-h：**

MongDB所在服务器地址，例如：127.0.0.1，当然也可以指定端口号：127.0.0.1:27017

* **-d：**

需要备份的数据库实例，例如：test

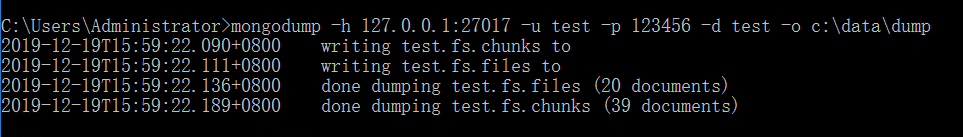
* **-o：**

备份的数据存放位置，例如：c:\data\dump，当然该目录需要提前建立，在备份完成后，系统自动在dump目录下建立一个test目录，这个目录里面存放该数据库实例的备份数据。

1. 备份操作

命令如下：

mongodump -h 127.0.0.1:27017 -u test -p 123456 -d test -o c:\data\dump



1. 备份后，会自动生成数据库名foshan\_db的文件夹



# 数据恢复

1. 恢复数据命令说明

mongorestore -h <hostname><:port> -d dbname <path>

* **--host <:port>, -h <:port>：**

MongoDB所在服务器地址，默认为： localhost:27017

* **--db , -d ：**

需要恢复的数据库实例，例如：test，当然这个名称也可以和备份时候的不一样，比如test2

* **--drop：**

恢复的时候，先删除当前数据，然后恢复备份的数据。就是说，恢复后，备份后添加修改的数据都会被删除，慎用哦！

* **<path>：**

mongorestore 最后的一个参数，设置备份数据所在位置，例如：c:\data\dump\test。

你不能同时指定 <path> 和 --dir 选项，--dir也可以设置备份目录。

* **--dir：**

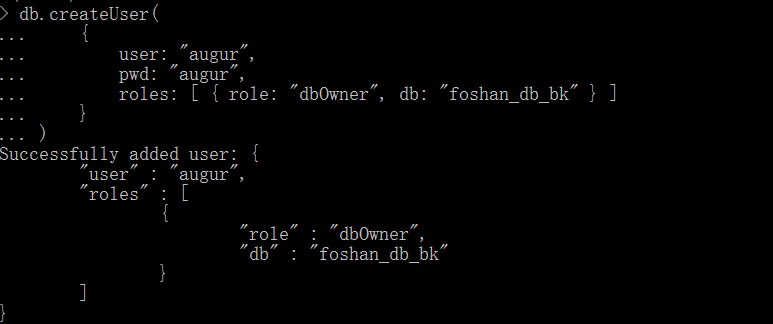
指定备份的目录

你不能同时指定 <path> 和 --dir 选项。

1. 恢复操作

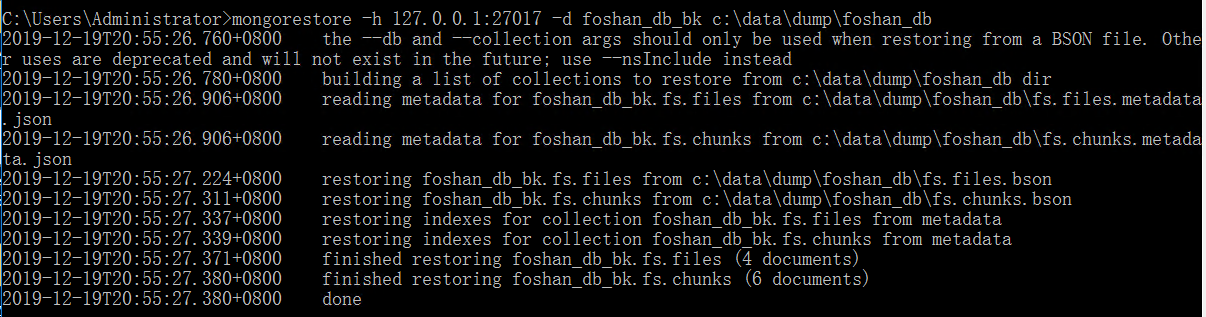
恢复操作到另一个数据库foshan\_db\_bk，需要新建数据库foshan\_db\_bk





新建成功后，恢复数据命令如下：

mongorestore -h 127.0.0.1:27017 -d foshan\_db\_bk c:\data\dump\foshan\_db



1. 检测是否恢复成功
2. 在原数据库上传一份材料后，检查是否可以预览和下载，预览下载成功后。
3. 再修改数据库配置，mongodb改成foshan\_db\_bk数据库。



1. 重启审批系统和coverpdf程序，再次检查是否可以预览和下载。若均可以，则备份成功。

# 集群部署

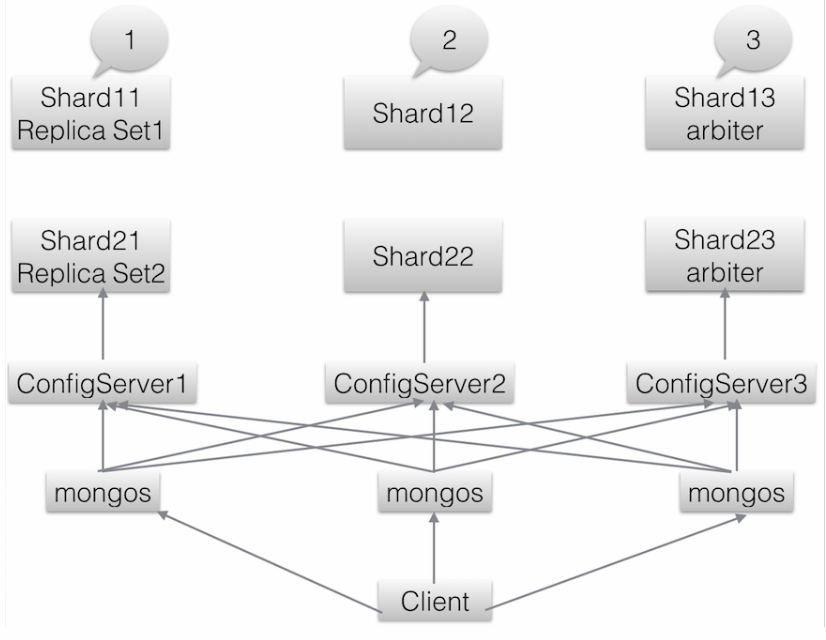
## 集群原理

一、集群组成：

1、配置服务器（是一个副本集）

2、路由服务器

3、分片服务器（副本集通常包含三个节点：主节点、备节点、仲裁节点）



这个图所表达的意思：

**Shard服务器：**使用Replica Sets确保每个数据节点都具有备份、自动容错转移、自动恢复的能力。

3个分片进程:Shard11,Shard12,Shard13组成一个副本集,提供分片中 shard1 的功能。

3个分片进程:Shard21,Shard22,Shard23组成一个副本集,提供分片中 Shard2 的功能。

**配置服务器：**使用使用3个配置服务器确保元数据完整性。

**路由进程：**使用3个路由进程实现平衡,提高客户端接入性能。

3个配置服务器进程和3个路由器进程。

构建一个 mongoDB Sharding Cluster 需要三种角色:shard 服务器(ShardServer)、配置服务器(config Server)、路由进程(Route Process)

一些说明：

**Shard 服务器**

　　shard 服务器即存储实际数据的分片,每个 shard 可以是一个 mongod 实例, 也可以是一组 mongod 实例构成的 Replica Sets.为了实现每个 Shard 内部的故障 自动转换,MongoDB 官方建议每个 shard 为一组 Replica Sets.  
**配置服务器**

　　为了将一个特定的 collection 存储在多个 shard 中,需要为该 collection 指定 一个 shard key,决定该条记录属于哪个 chunk,配置服务器可以存储以下信息, 每个shard节点的配置信息,每个chunk的shard key范围,chunk在各shard 的分布情况,集群中所有 DB 和 collection 的 sharding 配置信息。  
**路由进程**

　　它是一个前段路由,客户端由此接入,首先询问配置服务器需要到哪个 shard 上查询或保存记录,然后连接相应的 shard 执行操作,最后将结果返回给客户端,客 户端只需要将原本发给 mongod 的查询或更新请求原封不动地发给路由进程,而 不必关心所操作的记录存储在哪个 shard 上。

## 部署说明

1、配置服务器（3个）：

30001，30002，30003

2、路由服务器（3个）：

40000，40001，40002

3、分片服务器（每个副本集3个端口，2个分片6个）：

分片1：20001，20002，20003

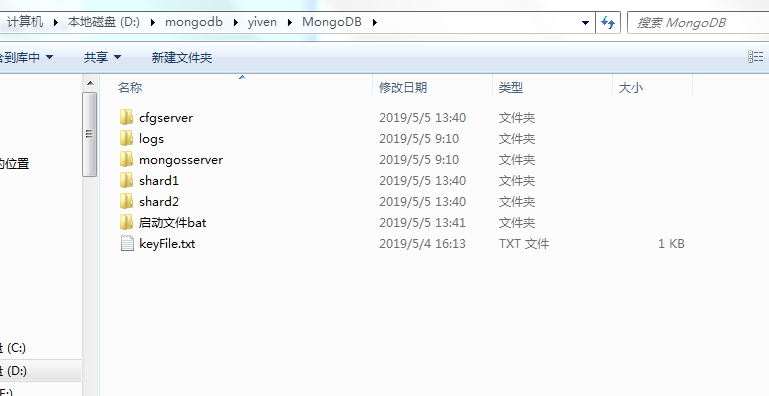
分片2：20004，20005，20006

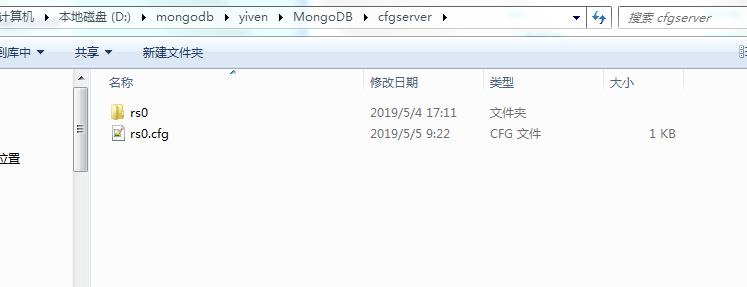
|  |  |  |
| --- | --- | --- |
| 服务器 | Host | 服务和端口 |
| 1 | 192.168.15.178 | Shard11:20001，shard21:20004，configSvr:30001，mongos:40000 |
| 2 | 192.168.20.34 | Shard12:20002，shard22:20005，configSvr:30002，mongos:40001 |
| 3 | 192.168.20.212 | Shard13:20003，shard23:20006，configSvr:30003，mongos:40002 |

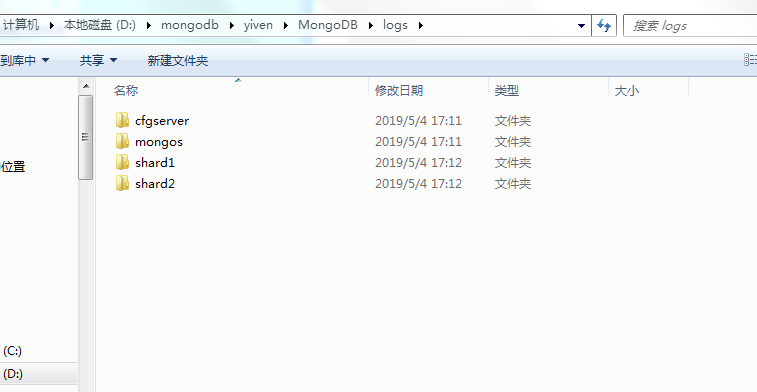
## 集群搭建

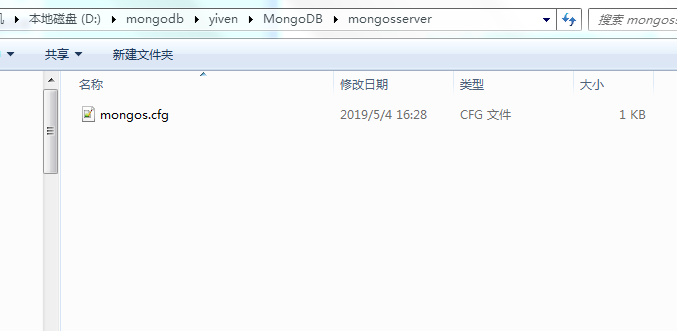
### 创建文件

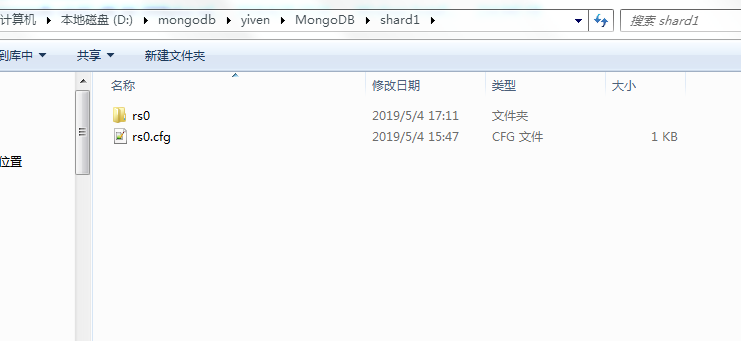
1、在服务器上创建每个副本集对应的文件夹（用来存储数据文件和存储日志文件的路径）

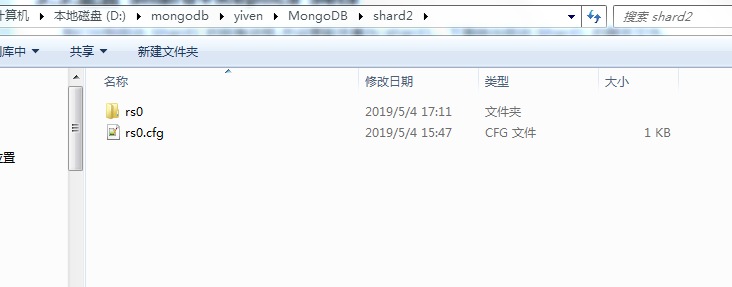












### 配置文件

1. 创建分片shard1副本集配置文件
   1. 主节点配置

# shard1

# mongod.conf

# for documentation of all options, see:

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

# Where and how to store data.

storage:

dbPath: D:\Program Files\MongoDB\shard1\rs0

journal:

enabled: true

# Where to write logging data.

systemLog:

destination: file

logAppend: true

path: D:\Program Files\MongoDB\logs\shard1\rs0.log

# network interfaces

net:

port: 20001

bindIp: 192.168.15.178

#security:

# keyFile: D:\Program Files\MongoDB\keyFile.txt

# authorization : "enabled"

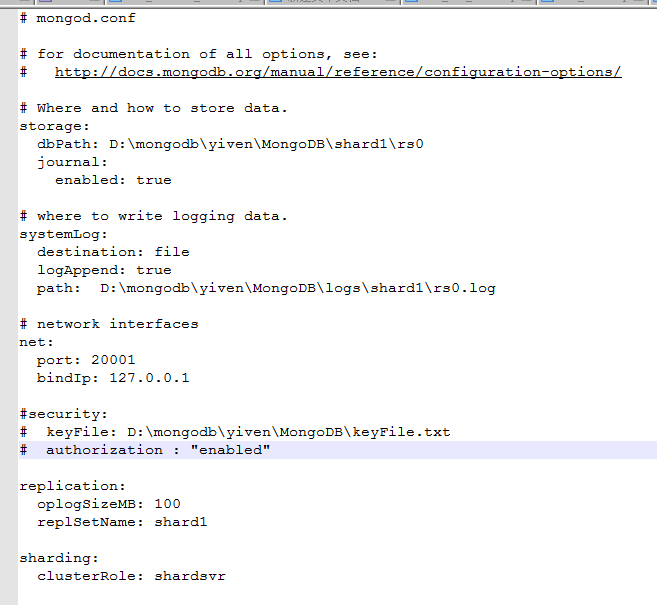
replication:

oplogSizeMB: 100

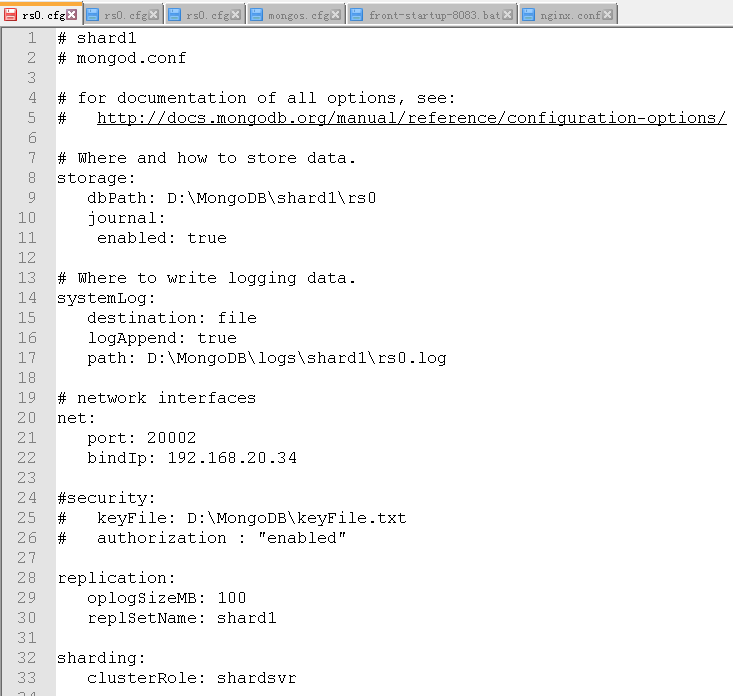
replSetName: shard1

sharding:

clusterRole: shardsvr



* 1. 备节点配置



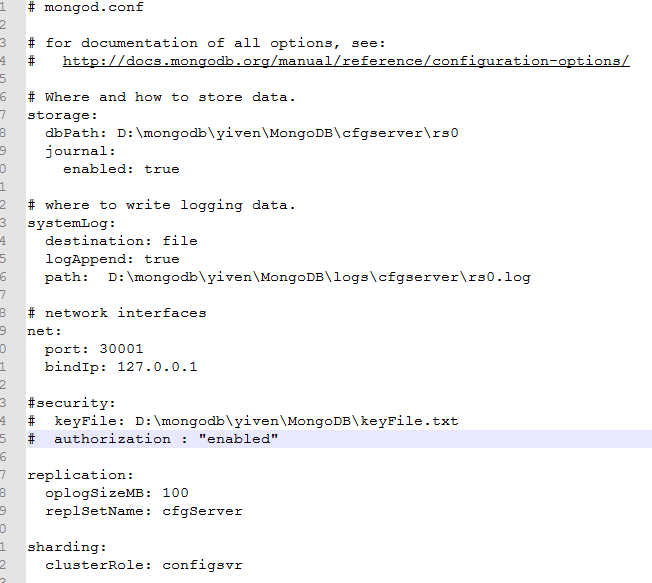
* 1. 仲裁节点配置

参考上面的配置。

1. 创建分片shard2副本集配置文件

参考上面的配置。

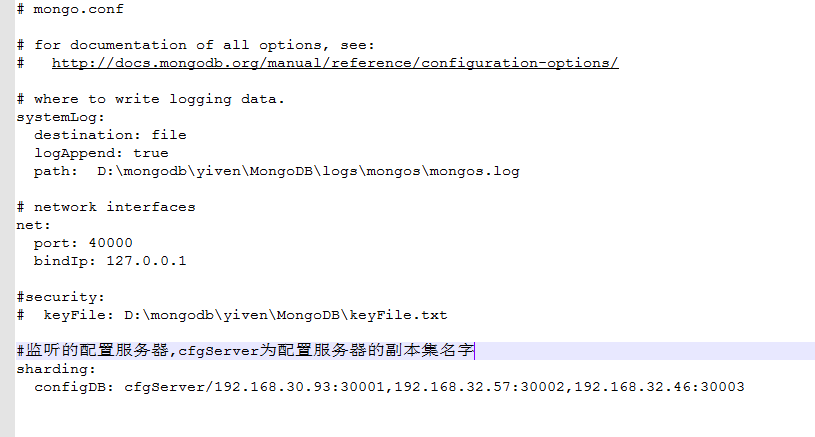
1. 创建配置服务器的配置文件
   1. 配置服务器1



* 1. 配置服务器2和配置服务器3。

参考上面的配置。

1. 创建路由服务器的配置文件



提醒：

1. 配置文件不要有tab键。
2. bin目录下默认创建的mongod.cfg要删掉。

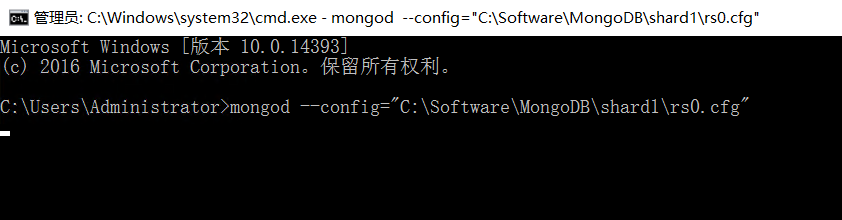
### 启动服务器

1. 启动分片服务器，创建分片管理员（方便远程单独登录分片服务器）
2. 启动分片1（三个窗口）

mongod --config="D:\Program Files\MongoDB\shard1\rs0.cfg"

mongod --config="D:\MongoDB\shard1\rs0.cfg"

mongod --config="D:\MongoDB\shard1\rs0.cfg"



1. 另开一个窗口连接主节点，执行初始化

mongo 192.168.15.178:20001

use admin

rs.initiate();

rs.add("192.168.20.34:20002");

rs.addArb("192.168.20.212:20003");

rs.status()

1. 创建分片管理员

use admin

db.createUser(

   {

     user: "admin",

     pwd: "admin",

     roles: [ "root" ]

   }

 )

 db.auth("admin", "admin")

1. 启动分片2

参考上面分片1

1. 启动配置服务器

mongod --config="D:\Program Files\MongoDB\cfgserver\rs0.cfg"

mongod --config="D:\MongoDB\cfgserver\rs0.cfg"

mongod --config="D:\MongoDB\cfgserver\rs0.cfg"

初始化，连接其中一个配置服务器

mongo localhost:30001

use admin

rs.initiate({\_id:"cfgServer",members:[{\_id:0,host:" 192.168.15.178:30001"},

{\_id:1,host:" 192.168.20.34:30002"}, {\_id:2,host:" 192.168.20.212:30003"}]})

rs.status()

其中cfgServer是配置文件里面的副本集名称

1. 启动路由服务器

mongos --config="D:\Program Files\MongoDB\mongosserver\mongos.cfg"

1. 添加各个分片到集群

打开新窗口连接到mongos服务器

mongo 192.168.15.178:40000

添加第一个分片的主从节点：

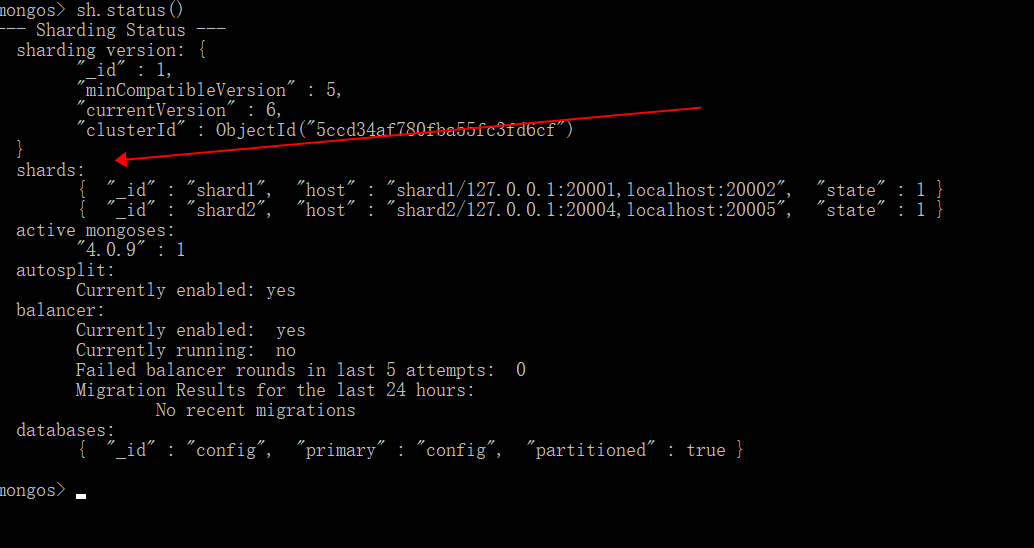
sh.addShard("shard1/192.168.15.178:20001,192.168.20.34:20002")

添加第二个分片的主从节点：

sh.addShard("shard2/192.168.15.178:20004,192.168.20.34:20005")

sh.status()

如图所示则添加成功了。



1. 设置超级管理员

use admin

db.createUser({user:"superman",pwd:"superman",roles:[ "clusterAdmin","userAdminAnyDatabase","dbAdminAnyDatabase","readWriteAnyDatabase" ]})

db.auth("superman","superman")

1. 至此，关闭全部副本集窗口。取消所有配置文件注释的：

security:

keyFile: F:\develop\MongoDB\mywindowmongodb\keyFile.txt

authorization : "enabled"

keyFile文件是一些base64编码的随机字符，具体了解：

<https://docs.mongodb.com/manual/core/security-internal-authentication/#inter-process-auth>

1. 打开验证后，重新启动分片服务器、配置服务器和路由服务器
2. 用超级管理员设置用户，客户端就用这个用户访问数据库。

首先连接路由：mongo 192.168.15.178:40000/admin -u superman -p superman

use tangshan\_dev2

db.createUser(

   {

     user: "tangshan\_dev2",

     pwd: "tangshan\_dev2",

     roles: [ "dbOwner"]

   }

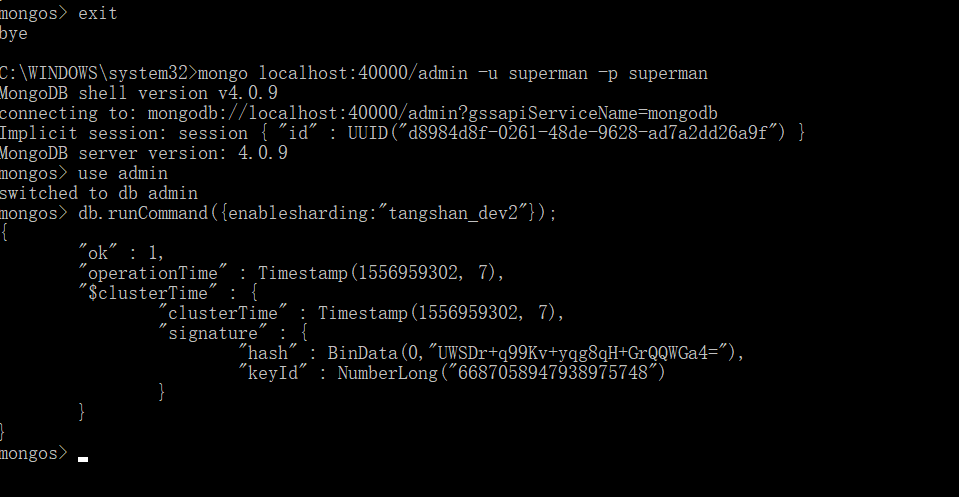
 )

db.auth("tangshan\_dev2","tangshan\_dev2")

Exit退出，重新使用superman 连接

激活数据库分片（tangshan\_dev2相当于数据库名）

db.runCommand({enablesharding:"tangshan\_dev2"});



Collecton分片（mongoAttInfo相当于表名）

db.runCommand( { shardcollection : "tangshan\_dev2.mongoAttInfo", key : { id : 1 } } );

参考地址：

<https://docs.mongodb.com/manual/reference/configuration-options/>

<https://www.cnblogs.com/zhanjun/p/4649277.html>

<https://www.cnblogs.com/silentjesse/p/4676440.html>

<https://www.cnblogs.com/zzw1787044/p/5773178.html>