

# 一、Windows环境下运行 MongoDB 服务

## 1.1 MongoDB下载地址

<https://www.mongodb.com/try/download/community>

## 1.2 安装后的配置

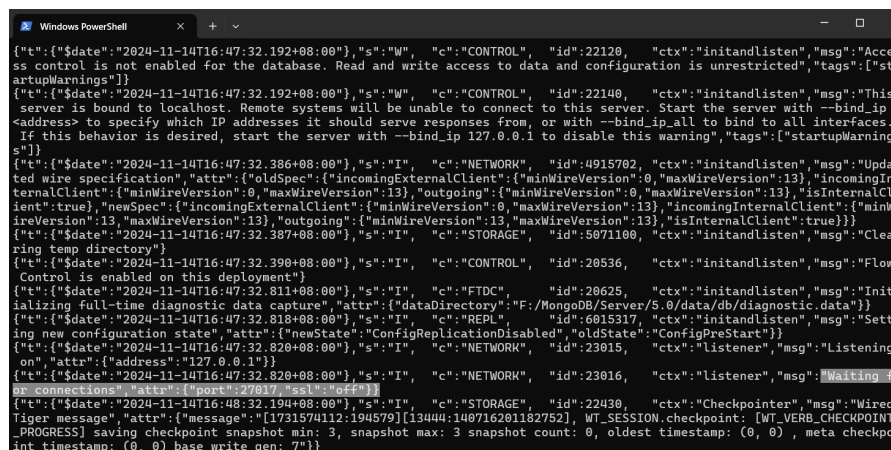
在你data的目录下，创建一个db文件夹；类似  
于：‘F:\MongoDB\Server\5.0\data\db’

因为启动 MongoDB 服务之前必须创建数据库文件的存放文件夹，否则命令不会自动创建，而且不能启动成功。

启动cmd，进入安装 MongoDB 的 db 文件，输入命令，来启动MongoDB 服务：

```
mongod --dbpath F:\MongoDB\Server\5.0\data\db
```

报“Waiting for connections”,“attr":{"port":27017,"ssl":"off"} 表示数据库已经启动。如下图所示。



```
Windows PowerShell
{"t":{"$date":"2024-11-14T16:47:32.192+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-14T16:47:32.192+08:00"},"s":"W", "c":"CONTROL", "id":22140, "ctx":"initandlisten","msg":"This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-14T16:47:32.386+08:00"},"s":"I", "c":"NETWORK", "id":4915702, "ctx":"initandlisten","msg":"Updated wire specification","attr":{"oldSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVersion":0,"maxWireVersion":13},"isInternalClient":true},"newSpec":{"incomingExternalClient":{"minWireVersion":0,"maxWireVersion":13},"incomingInternalClient":{"minWireVersion":13,"maxWireVersion":13},"outgoing":{"minWireVersion":13,"maxWireVersion":13},"isInternalClient":true}}}
{"t":{"$date":"2024-11-14T16:47:32.387+08:00"},"s":"I", "c":"STORAGE", "id":5871100, "ctx":"initandlisten","msg":"Cleaning temp directory"}
{"t":{"$date":"2024-11-14T16:47:32.399+08:00"},"s":"I", "c":"CONTROL", "id":28536, "ctx":"initandlisten","msg":"Flow Control is enabled on this deployment"}
{"t":{"$date":"2024-11-14T16:47:32.811+08:00"},"s":"I", "c":"FTDC", "id":28625, "ctx":"initandlisten","msg":"Initializing full-time diagnostic data capture","attr":{"dataDirectory":"F:/MongoDB/Server/5.0/data/db/diagnostic.data"}}
{"t":{"$date":"2024-11-14T16:47:32.818+08:00"},"s":"I", "c":"REPL", "id":6015317, "ctx":"initandlisten","msg":"Setting new configuration state","attr":{"newState":"ConfigReplicationDisabled","oldState":"ConfigPreStart"}}
{"t":{"$date":"2024-11-14T16:47:32.820+08:00"},"s":"I", "c":"NETWORK", "id":23015, "ctx":"listener","msg":"Listening on","attr":{"address":"127.0.0.1"}}
{"t":{"$date":"2024-11-14T16:47:32.820+08:00"},"s":"I", "c":"NETWORK", "id":23016, "ctx":"listener","msg":"Waiting for connections"}
{"t":{"$date":"2024-11-14T16:48:32.194+08:00"},"s":"I", "c":"STORAGE", "id":22430, "ctx":"Checkpointner","msg":"Wired Tiger message","attr":{"message":["[1731574112:194579][13444:140716201182752], WT_SESSION.checkpoint: [WT_VERB_CHECKPOINT_PROGRESS] saving checkpoint snapshot min: 3, snapshot max: 3 snapshot count: 0, oldest timestamp: (0, 0), meta checkpoint timestamp: (0, 0) base write gen: 7"]}}
```

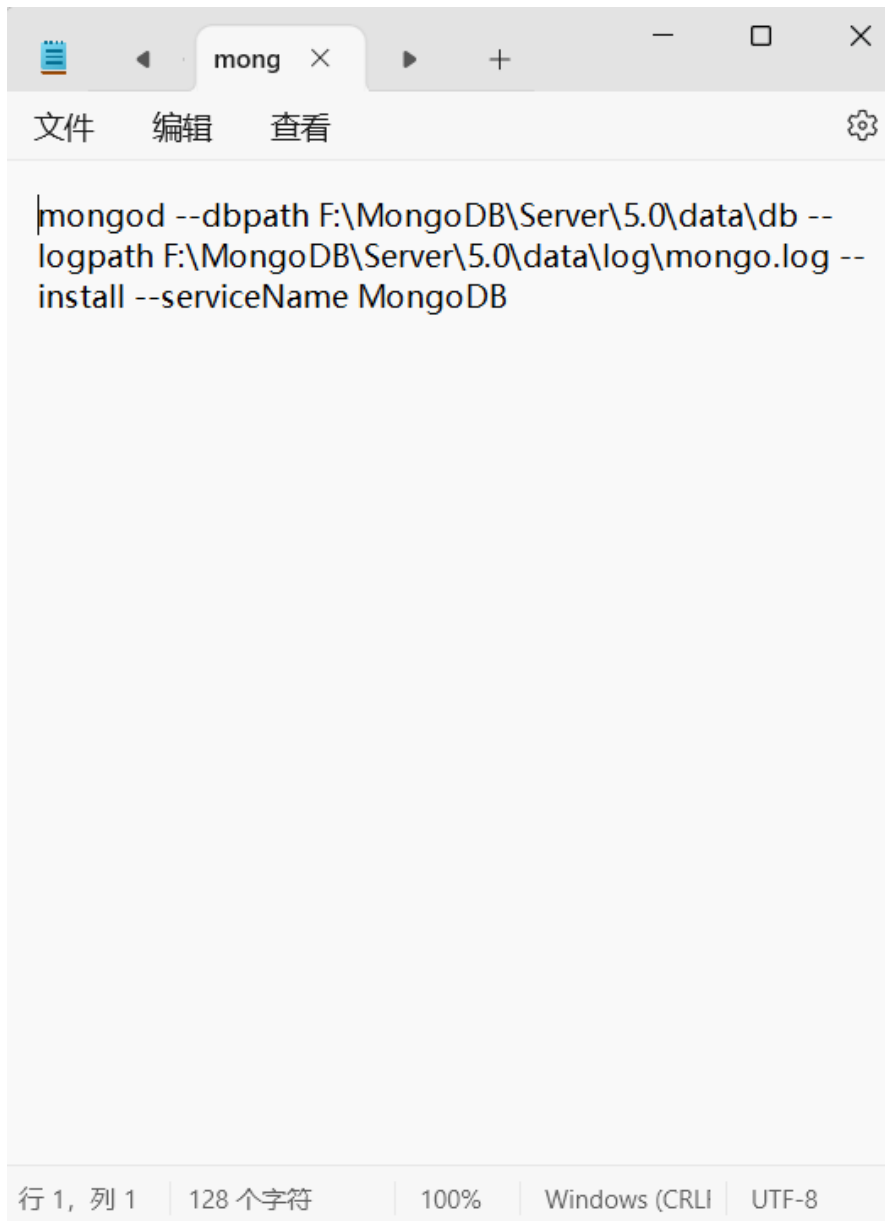
## 1.3 如何结束服务

一般是按两次的 ‘Ctrl + C’，就结束了该次进程：

## 1.4 配置本地 Windows MongoGB 服务

设置为开机自启动，可直接手动启动关闭，可通过命令行net start MongoDB启动。

- 1) 在 data 文件下创建新文件夹log; (用来存放日志文件);
- 2) 在 MongoGB 中新建配置文件 mongo.config; (与 bin 目录同级)
- 3) 用记事本打开 mongo.config 文件，并输入下面两个命令，然后保存;



The image shows a Notepad window with the title bar 'mong'. The menu bar includes '文件' (File), '编辑' (Edit), '查看' (View), and a settings icon. The text area contains the following commands:

```
|mongod --dbpath F:\MongoDB\Server\5.0\data\db --  
logpath F:\MongoDB\Server\5.0\data\log\mongo.log --  
install --serviceName MongoDB
```

The status bar at the bottom indicates '行 1, 列 1' (Line 1, Column 1), '128 个字符' (128 characters), '100%', 'Windows (CRLF)', and 'UTF-8'.

(注意：以自己的实际安装的文件地址为准)

- 4) 用管理员身份打开 cmd，然后找到 bin 文件地址为："  
F:\MongoDB\Server\5.0\bin"，并输入代码为：

```
mongod --dbpath F:\MongoDB\Server\5.0\data\db --logpath
F:\MongoDB\Server\5.0\data\log\mongo.log --install --serviceName
MongoDB
```

```
F:\MongoDB\Server\5.0\bin>mongod --dbpath F:\MongoDB\Server\5.0\data\db --logpath F:\MongoDB\Server\5.0\data\log\mongo.
log --install --serviceName MongoDB
{"t":{"$date":"2024-11-14T09:10:42.366Z"},"s":"I", "c":"CONTROL", "id":20697, "ctx":"","msg":"Renamed existing log
file", "attr":{"oldLogPath":"F:\\MongoDB\\Server\\5.0\\data\\log\\mongo.log", "newLogPath":"F:\\MongoDB\\Server\\5.0\\data
\\log\\mongo.log.2024-11-14T09-10-42"}}
```

## 1.5、在cmd 管理员中启动和关闭 MongoDB 服务

其一、启动 MongoDB 命令为：

```
net start MongoDB
```

```
F:\MongoDB\Server\5.0>net start MongoDB
MongoDB Server (MongoDB) 服务正在启动。
MongoDB Server (MongoDB) 服务已经启动成功。
```

<http://localhost:27017/>



其三、关闭 MongoDB 命令为：

```
net stop MongoDB
```

```
F:\MongoDB\Server\5.0>net stop MongoDB
MongoDB Server (MongoDB) 服务正在停止。
MongoDB Server (MongoDB) 服务已成功停止。
```

## 二、Linux环境下分布式部署 MongoDB

### 1.1 下载与环境变量的配置

#### 1.2.1 到MongoDB官网下载(3台服务器执行相同操作)

[https://fastdl.mongodb.org/linux/mongodb-linux-x86\\_64-ubuntu1804-6.0.19.tgz](https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-ubuntu1804-6.0.19.tgz)

### 1.2.2 解压到/home/zzs并创建软链接(3台服务器执行相同操作)

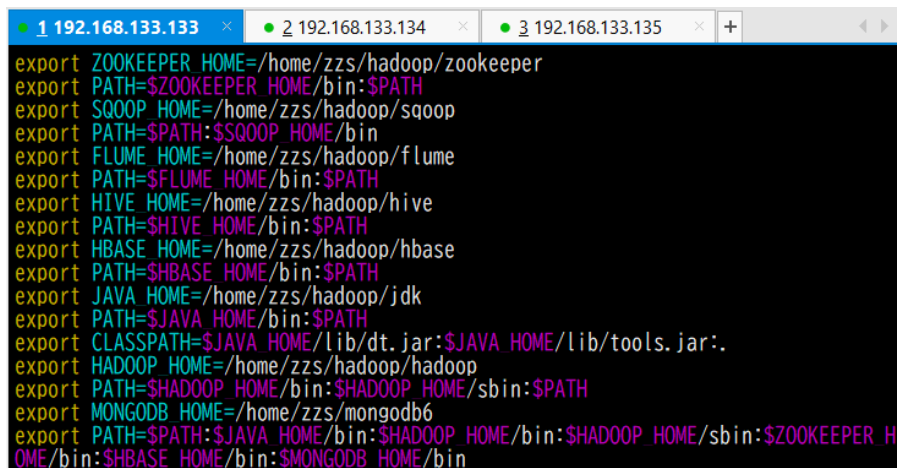
```
cd /home/zzs
tar -zxvf mongodb-linux-x86_64-ubuntu1804-6.0.19.tgz
ln -s mongodb-linux-x86_64-ubuntu1804-6.0.19 mongodb6
```

### 1.2.3配置环境变量(3台服务器执行相同操作)

```
sudo vi ~/.bashrc
```

```
export MONGODB_HOME=/home/zzs/mongodb6
export
PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:
in:$ZOOKEEPER_HOME/bin:$HBASE_HOME/bin:$MONGODB_HOME/bin
```

```
source ~/.bashrc
```



A terminal window with three tabs labeled 1 192.168.133.133, 2 192.168.133.134, and 3 192.168.133.135. The terminal displays the following environment variable configurations:

```
export ZOOKEEPER_HOME=/home/zzs/hadoop/zookeeper
export PATH=$ZOOKEEPER_HOME/bin:$PATH
export SQOOP_HOME=/home/zzs/hadoop/sqoop
export PATH=$PATH:$SQOOP_HOME/bin
export FLUME_HOME=/home/zzs/hadoop/flume
export PATH=$FLUME_HOME/bin:$PATH
export HIVE_HOME=/home/zzs/hadoop/hive
export PATH=$HIVE_HOME/bin:$PATH
export HBASE_HOME=/home/zzs/hadoop/hbase
export PATH=$HBASE_HOME/bin:$PATH
export JAVA_HOME=/home/zzs/hadoop/jdk
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:.
export HADOOP_HOME=/home/zzs/hadoop/hadoop
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export MONGODB_HOME=/home/zzs/mongodb6
export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$ZOOKEEPER_H
OME/bin:$HBASE_HOME/bin:$MONGODB_HOME/bin
```

```
1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
export ZOOKEEPER_HOME=/home/zzs/hadoop/zookeeper
export PATH=$ZOOKEEPER_HOME/bin:$PATH
export SQOOP_HOME=/home/zzs/hadoop/sqoop
export PATH=$PATH:$SQOOP_HOME/bin
export FLUME_HOME=/home/zzs/hadoop/flume
export PATH=$FLUME_HOME/bin:$PATH
export HIVE_HOME=/home/zzs/hadoop/hive
export PATH=$HIVE_HOME/bin:$PATH
export HBASE_HOME=/home/zzs/hadoop/hbase
export PATH=$HBASE_HOME/bin:$PATH
export JAVA_HOME=/home/zzs/hadoop/jdk
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:
export HADOOP_HOME=/home/zzs/hadoop/hadoop
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export MONGODB_HOME=/home/zzs/mongodb6
export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$ZOOKEEPER_H
OME/bin:$HBASE_HOME/bin:$MONGODB_HOME/bin
```

```
1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
export ZOOKEEPER_HOME=/home/zzs/hadoop/zookeeper
export PATH=$ZOOKEEPER_HOME/bin:$PATH
export SQOOP_HOME=/home/zzs/hadoop/sqoop
export PATH=$PATH:$SQOOP_HOME/bin
export FLUME_HOME=/home/zzs/hadoop/flume
export PATH=$FLUME_HOME/bin:$PATH
export HIVE_HOME=/home/zzs/hadoop/hive
export PATH=$HIVE_HOME/bin:$PATH
export HBASE_HOME=/home/zzs/hadoop/hbase
export PATH=$HBASE_HOME/bin:$PATH
export JAVA_HOME=/home/zzs/hadoop/jdk
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:
export HADOOP_HOME=/home/zzs/hadoop/hadoop
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export MONGODB_HOME=/home/zzs/mongodb6
export PATH=$PATH:$JAVA_HOME/bin:$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$ZOOKEEPER_H
OME/bin:$HBASE_HOME/bin:$MONGODB_HOME/bin
```

#### 1.2.4创建路由、配置、分片等的相关目录与文件(3台服务器执行相同操作)

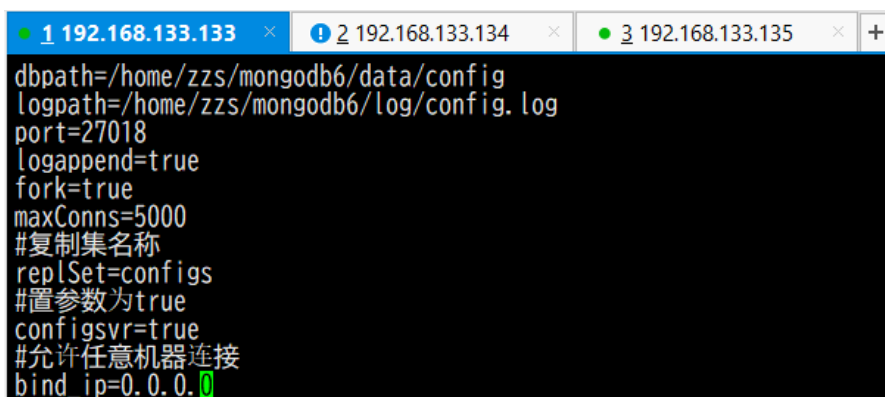
```
mkdir -p /home/zzs/mongodb6/conf
mkdir -p /home/zzs/mongodb6/log
mkdir -p /home/zzs/mongodb6/data/config
mkdir -p /home/zzs/mongodb6/data/shard1
mkdir -p /home/zzs/mongodb6/data/shard2
mkdir -p /home/zzs/mongodb6/data/shard3
touch /home/zzs/mongodb6/log/config.log
touch /home/zzs/mongodb6/log/mongos.log
touch /home/zzs/mongodb6/log/shard1.log
touch /home/zzs/mongodb6/log/shard2.log
touch /home/zzs/mongodb6/log/shard3.log
touch /home/zzs/mongodb6/conf/config.conf
```

#### 1.2.5配置服务器部署(3台服务器执行相同操作)

```
vi /home/zzs/mongodb6/conf/config.conf
```

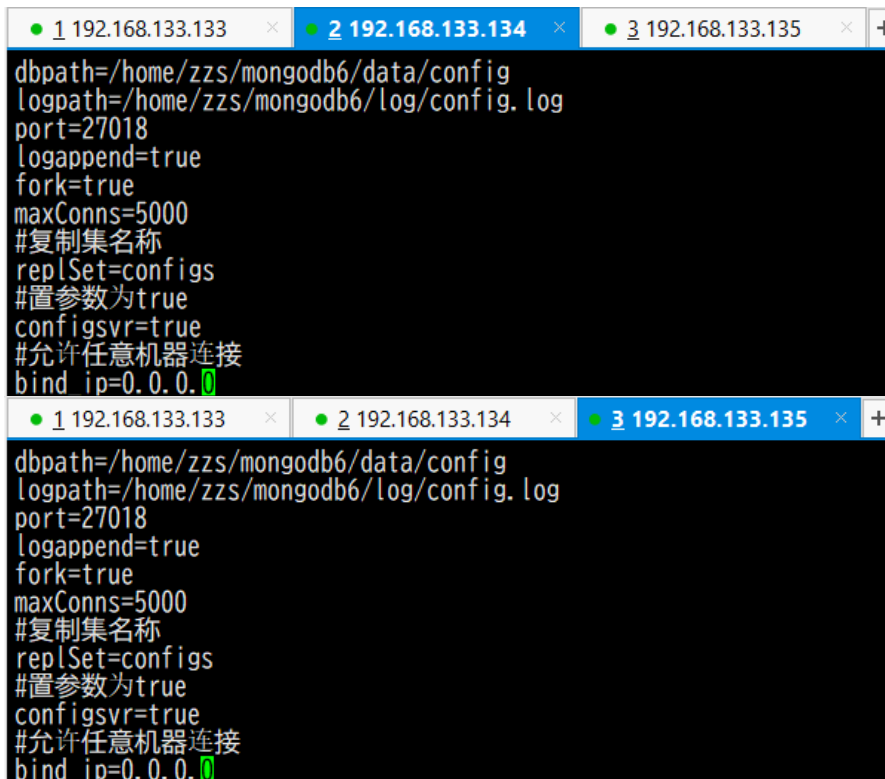
配置/home/zzs/mongodb6/conf/config.conf如下:

```
dbpath=/home/zzs/mongodb6/data/config
logpath=/home/zzs/mongodb6/log/config.log
port=27018
logappend=true
fork=true
maxConns=5000
#复制集名称
replSet=configs
#置参数为true
configsvr=true
#允许任意机器连接
bind_ip=0.0.0.0
```



A terminal window with three tabs: 1 192.168.133.133, 2 192.168.133.134 (active), and 3 192.168.133.135. The active tab displays the same MongoDB configuration as the first block, with a green cursor at the end of the bind\_ip line.

```
dbpath=/home/zzs/mongodb6/data/config
logpath=/home/zzs/mongodb6/log/config.log
port=27018
logappend=true
fork=true
maxConns=5000
#复制集名称
replSet=configs
#置参数为true
configsvr=true
#允许任意机器连接
bind_ip=0.0.0.0
```



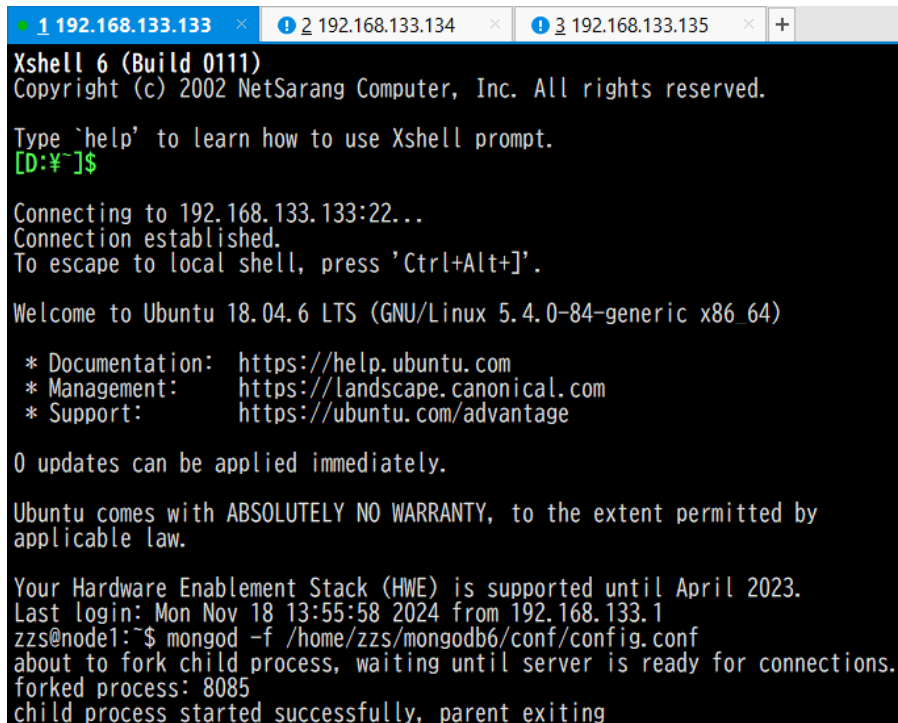
A terminal window with three tabs: 1 192.168.133.133, 2 192.168.133.134, and 3 192.168.133.135 (active). The active tab displays the same MongoDB configuration as the first block, with a green cursor at the end of the bind\_ip line.

```
dbpath=/home/zzs/mongodb6/data/config
logpath=/home/zzs/mongodb6/log/config.log
port=27018
logappend=true
fork=true
maxConns=5000
#复制集名称
replSet=configs
#置参数为true
configsvr=true
#允许任意机器连接
bind_ip=0.0.0.0
```

## 1.2配置复制集

### 1.2.1 分别启动三台服务器的配置服务

```
cd /home/zs/mongodb6/bin
chmod -R 777 /home/zs/mongodb6/
sudo apt-get install libcurl4 #根据报错自行修改
sudo apt install mongodb-server-core
mongod -f /home/zs/mongodb6/conf/config.conf
```



Xshell 6 (Build 0111)  
Copyright (c) 2002 NetSarang Computer, Inc. All rights reserved.

Type `help` to learn how to use Xshell prompt.  
[D:¥~]\$

Connecting to 192.168.133.133:22...  
Connection established.  
To escape to local shell, press 'Ctrl+Alt+J'.

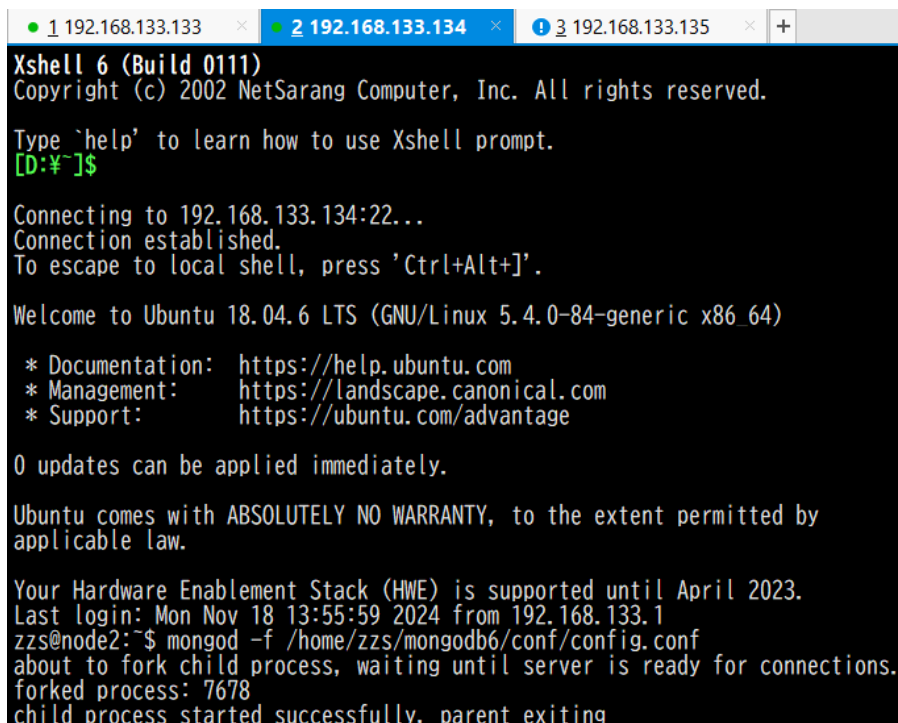
Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-84-generic x86\_64)

\* Documentation: <https://help.ubuntu.com>  
\* Management: <https://landscape.canonical.com>  
\* Support: <https://ubuntu.com/advantage>

0 updates can be applied immediately.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Your Hardware Enablement Stack (HWE) is supported until April 2023.  
Last login: Mon Nov 18 13:55:58 2024 from 192.168.133.1  
zs@node1:~\$ mongod -f /home/zs/mongodb6/conf/config.conf  
about to fork child process, waiting until server is ready for connections.  
forked process: 8085  
child process started successfully, parent exiting



Xshell 6 (Build 0111)  
Copyright (c) 2002 NetSarang Computer, Inc. All rights reserved.

Type `help` to learn how to use Xshell prompt.  
[D:¥~]\$

Connecting to 192.168.133.134:22...  
Connection established.  
To escape to local shell, press 'Ctrl+Alt+J'.

Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-84-generic x86\_64)

\* Documentation: <https://help.ubuntu.com>  
\* Management: <https://landscape.canonical.com>  
\* Support: <https://ubuntu.com/advantage>

0 updates can be applied immediately.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by applicable law.

Your Hardware Enablement Stack (HWE) is supported until April 2023.  
Last login: Mon Nov 18 13:55:59 2024 from 192.168.133.1  
zs@node2:~\$ mongod -f /home/zs/mongodb6/conf/config.conf  
about to fork child process, waiting until server is ready for connections.  
forked process: 7678  
child process started successfully, parent exiting

```
1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
Xshell 6 (Build 0111)
Copyright (c) 2002 NetSarang Computer, Inc. All rights reserved.

Type 'help' to learn how to use Xshell prompt.
[D:¥~]$

Connecting to 192.168.133.135:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+J'.

Welcome to Ubuntu 18.04.6 LTS (GNU/Linux 5.4.0-84-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 updates can be applied immediately.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

Your Hardware Enablement Stack (HWE) is supported until April 2023.
Last login: Mon Nov 18 13:55:59 2024 from 192.168.133.1
zxs@node3:~$ mongod -f /home/zxs/mongodb6/conf/config.conf
about to fork child process, waiting until server is ready for connections.
forked process: 7657
child process started successfully, parent exiting
```

1.2.2连接mongo,只需在任意一台机器执行即可

```
sudo apt install mongodb-clients
mongo --host 192.168.133.133 --port 27018
```

### 1.2.3切换数据库

```
use admin
```

### 1.2.4初始化复制

```
rs.initiate({_id:"configs",members:[{_id:0,host: "192.168.133.133:27018"},
{_id:1,host: "192.168.133.134:27018"},{_id:2,host:
"192.168.133.135:27018"}]})
```



```

zzs@node1:~$ mongo --host 192.168.133.133 --port 27018
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.133:27018/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T13:58:26.500+08:00"},"s":"I", "c":"STORAGE", "id":22297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/production-notes-filesystem","tags":["startupWarnings"]}}

{"t":{"$date":"2024-11-18T13:58:30.976+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}}

{"t":{"$date":"2024-11-18T13:58:30.976+08:00"},"s":"W", "c":"CONTROL", "id":22184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too low","attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarnings"]}}

configs:SECONDARY> use admin
switched to db admin
configs:SECONDARY> rs.initiate({ id:"configs",members:[{ id:0,host: "192.168.133.133:27018"},{ id:1,host: "192.168.133.134:27018"},{ id:2,host: "192.168.133.135:27018"}]})
{
  "ok" : 0,
  "errmsg" : "already initialized",
  "code" : 23,
  "codeName" : "AlreadyInitialized",
  "lastCommittedOpTime" : Timestamp(1731909606, 1),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1731909606, 1),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA="),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1731909606, 1)
}

```

查看状态:

```
rs.status()
```

```

1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
}
configs:SECONDARY> rs.status()
{
  "set" : "configs",
  "date" : ISODate("2024-11-18T06:00:18.350Z"),
  "myState" : 2,
  "term" : NumberLong(9),
  "syncSourceHost" : "192.168.133.135:27018",
  "syncSourceId" : 2,
  "configsvr" : true,
  "heartbeatIntervalMillis" : NumberLong(2000),
  "majorityVoteCount" : 2,
  "writeMajorityCount" : 2,
  "votingMembersCount" : 3,
  "writableVotingMembersCount" : 3,
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1731909618, 1),
      "t" : NumberLong(9)
    },
    "lastCommittedWallTime" : ISODate("2024-11-18T06:00:18.404Z"),
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1731909618, 1),
      "t" : NumberLong(9)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1731909618, 1),
      "t" : NumberLong(9)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1731909618, 1),
      "t" : NumberLong(9)
    },
    "lastAppliedWallTime" : ISODate("2024-11-18T06:00:18.404Z"),
    "lastDurableWallTime" : ISODate("2024-11-18T06:00:18.404Z")
  },
  "lastStableRecoveryTimestamp" : Timestamp(1731909570, 1),
  "electionParticipantMetrics" : {
    "votedForCandidate" : true,
    "electionTerm" : NumberLong(9),
    "lastVoteDate" : ISODate("2024-11-18T05:58:44.358Z"),
    "electionCandidateMemberId" : 1,
    "voteReason" : "",
    "lastAppliedOpTimeAtElection" : {
      "members" : [
        {
          "id" : 0,
          "name" : "192.168.133.133:27018",
          "health" : 1,
          "state" : 2,
          "stateStr" : "SECONDARY",
          "uptime" : 112,
          "optime" : {
            "ts" : Timestamp(1731909618, 1),
            "t" : NumberLong(9)
          },
          "optimeDate" : ISODate("2024-11-18T06:00:18Z"),
          "lastAppliedWallTime" : ISODate("2024-11-18T06:00:18.404Z"),
          "lastDurableWallTime" : ISODate("2024-11-18T06:00:18.404Z"),
          "syncSourceHost" : "192.168.133.135:27018",
          "syncSourceId" : 2,
          "infoMessage" : "",
          "configVersion" : 1,
          "configTerm" : 9,
          "self" : true,
          "lastHeartbeatMessage" : ""
        },
        {
          "id" : 1,
          "name" : "192.168.133.134:27018",
          "health" : 1,
          "state" : 1,
          "stateStr" : "PRIMARY",
          "uptime" : 106,
          "optime" : {
            "ts" : Timestamp(1731909617, 1),
            "t" : NumberLong(9)
          }
        }
      ]
    }
  }
}

```

```
    "t" : NumberLong(9)
  },
  "optimeDurable" : {
    "ts" : Timestamp(1731909617, 1),
    "t" : NumberLong(9)
  },
  "optimeDate" : ISODate("2024-11-18T06:00:17Z"),
  "optimeDurableDate" : ISODate("2024-11-18T06:00:17Z"),
  "lastAppliedWallTime" : ISODate("2024-11-18T06:00:17.404Z"),
}
```

## 1.3配置分片服务部署

### 1.3.1分片服务部署(3台服务器执行相同操作)

在/home/zzs/mongodb6/conf目录创建shard1.conf、shard2.conf、shard3.conf。

```
touch /home/zzs/mongodb6/conf/shard1.conf
touch /home/zzs/mongodb6/conf/shard2.conf
touch /home/zzs/mongodb6/conf/shard3.conf
```

```
vi /home/zzs/mongodb6/conf/shard1.conf
vi /home/zzs/mongodb6/conf/shard2.conf
vi /home/zzs/mongodb6/conf/shard3.conf
```

内容如下：

```
dbpath=/home/zzs/mongodb6/data/shard1 #其他2个分片对应修改为shard2、
shard3文件夹
logpath=/home/zzs/mongodb6/log/shard1.log #其他2个分片对应修改为
shard2.log、shard3.log
port=27001 #其他2个分片对应修改为27002、27003
logappend=true
fork=true
maxConns=5000
#storageEngine=mmapv1(必须注释掉，否则无法运行)
shardsvr=true
replSet=shard1 #其他2个分片对应修改为shard2、shard3
bind_ip=0.0.0.0
```

```

1 192.168.133.133 x 2 192.168.133.134 3 192.168.133.135 x +
dbpath=/home/zzs/mongodb6/data/shard1 #其他2个分片对应修改为shard2、shard3文件夹
logpath=/home/zzs/mongodb6/log/shard1.log #其他2个分片对应修改为shard2.log、shar
d3.log
port=27001 #其他2个分片对应修改为27002、27003
logappend=true
fork=true
maxConns=5000
#storageEngine=mmapv1(必须注释掉, 否则无法运行)
shardsvr=true
replSet=shard1 #其他2个分片对应修改为shard2、shard3
bind_ip=0.0.0.0
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~/mongodb6/conf/shard1.conf 10L, 442C 10, 15 全部

```

```
dbpath=/home/zzs/mongodb6/data/shard2 #其他2个分片对应修改为shard2、shard3文件夹  
logpath=/home/zzs/mongodb6/log/shard2.log #其他2个分片对应修改为shard2.log、shar  
d3.log  
port=27002 #其他2个分片对应修改为27002、27003  
logappend=true  
fork=true  
maxConns=5000  
#storageEngine=mmapv1(必须注释掉，否则无法运行)  
shardsvr=true  
replSet=shard2 #其他2个分片对应修改为shard2、shard3  
bind_ip=0.0.0.0
```



```
1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
zszs@node2:~$ mongod -f /home/zszs/mongodb6/conf/shard1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 7945
child process started successfully, parent exiting
```

```
1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
zszs@node3:~$ mongod -f /home/zszs/mongodb6/conf/shard1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 7927
child process started successfully, parent exiting
```

### 1.3.2.2连接mongo

```
mongo --host 192.168.133.133 --port 27001
mongo --host 192.168.133.134 --port 27001
mongo --host 192.168.133.135 --port 27001
```

以上命令要分别在三台机器上执行。

```
1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
zszs@node1:~$ mongod -f /home/zszs/mongodb6/conf/shard1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 8362
child process started successfully, parent exiting
zszs@node1:~$ mongo --host 192.168.133.133 --port 27001
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.133:27001/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:07:48.200+08:00"},"s":"I", "c":"STORAGE", "id":22
297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recomme
nded with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prod
notes-filesystem","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:07:50.088+08:00"},"s":"W", "c":"CONTROL", "id":22
120, "ctx":"initandlisten","msg":"Access control is not enabled for the databa
se. Read and write access to data and configuration is unrestricted","tags":["st
artupWarnings"]}
{"t":{"$date":"2024-11-18T14:07:50.089+08:00"},"s":"W", "c":"CONTROL", "id":22
184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too l
ow","attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarn
ings"]}
shard1:PRIMARY>
```

```
1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
zsz@node2:~$ mongod -f /home/zsz/mongodb6/conf/shard1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 7945
child process started successfully, parent exiting
zsz@node2:~$ mongo --host 192.168.133.133 --port 27001
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.133:27001/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:07:48.200+08:00"},"s":"I", "c":"STORAGE", "id":22
297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recomme
nded with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prod
notes-filesystem","tags":["startupWarnings"]}

{"t":{"$date":"2024-11-18T14:07:50.088+08:00"},"s":"W", "c":"CONTROL", "id":22
120, "ctx":"initandlisten","msg":"Access control is not enabled for the databa
se. Read and write access to data and configuration is unrestricted","tags":["st
artupWarnings"]}

{"t":{"$date":"2024-11-18T14:07:50.089+08:00"},"s":"W", "c":"CONTROL", "id":22
184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too l
ow","attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarn
ings"]}

shard1:PRIMARY>
```

```
1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
zsz@node3:~$ mongod -f /home/zsz/mongodb6/conf/shard1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 7927
child process started successfully, parent exiting
zsz@node3:~$ mongo --host 192.168.133.133 --port 27001
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.133:27001/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:07:48.200+08:00"},"s":"I", "c":"STORAGE", "id":22
297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recomme
nded with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prod
notes-filesystem","tags":["startupWarnings"]}

{"t":{"$date":"2024-11-18T14:07:50.088+08:00"},"s":"W", "c":"CONTROL", "id":22
120, "ctx":"initandlisten","msg":"Access control is not enabled for the databa
se. Read and write access to data and configuration is unrestricted","tags":["st
artupWarnings"]}

{"t":{"$date":"2024-11-18T14:07:50.089+08:00"},"s":"W", "c":"CONTROL", "id":22
184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too l
ow","attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarn
ings"]}

shard1:PRIMARY>
```

### 1.3.2.3切换数据库

use admin

以上命令一台机器上执行即可。



```
1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
zzs@node1:~$ mongod -f /home/zzs/mongodb6/conf/shard1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 8362
child process started successfully, parent exiting
zzs@node1:~$ mongo --host 192.168.133.133 --port 27001
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.133:27001/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:07:48.200+08:00"},"s":"I", "c":"STORAGE", "id":22
297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recomme
nded with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prod
notes-filesystem","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:07:50.088+08:00"},"s":"W", "c":"CONTROL", "id":22
120, "ctx":"initandlisten","msg":"Access control is not enabled for the databa
se. Read and write access to data and configuration is unrestricted","tags":["st
artupWarnings"]}
{"t":{"$date":"2024-11-18T14:07:50.089+08:00"},"s":"W", "c":"CONTROL", "id":22
184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too l
ow","attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarn
ings"]}
shard1:PRIMARY> use admin
switched to db admin
shard1:PRIMARY>
```

#### 1.3.2.4初始化复制集

```
rs.initiate({_id:"shard1",members:[{_id:0,host: "192.168.133.133:27001"},
{_id:1,host: "192.168.133.134:27001"},{_id:2,host:
"192.168.133.135:27001"}]})
```

```

1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
forked process: 8362
child process started successfully, parent exiting
zsz@node1:~$ mongo --host 192.168.133.133 --port 27001
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.133:27001/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:07:48.200+08:00"},"s":"I", "c":"STORAGE", "id":22297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem","tags":["startupWarnings"]}

{"t":{"$date":"2024-11-18T14:07:50.088+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}

{"t":{"$date":"2024-11-18T14:07:50.089+08:00"},"s":"W", "c":"CONTROL", "id":22184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too low", "attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarnings"]}

shard1:PRIMARY> use admin
switched to db admin
shard1:PRIMARY> rs.initiate({ id:"shard1",members:[{ id:0,host: "192.168.133.133:27001"},{ id:1,host: "192.168.133.134:27001"},{ id:2,host: "192.168.133.135:27001"}]})
{
  "ok" : 0,
  "errmsg" : "already initialized",
  "code" : 23,
  "codeName" : "AlreadyInitialized",
  "lastCommittedOpTime" : Timestamp(1731910243, 2),
  "$clusterTime" : {
    "clusterTime" : Timestamp(1731910243, 2),
    "signature" : {
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"),
      "keyId" : NumberLong(0)
    }
  },
  "operationTime" : Timestamp(1731910243, 2)
}

```

```
1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
shard1:PRIMARY> rs.status()
{
  "set" : "shard1",
  "date" : ISODate("2024-11-18T06:11:17.624Z"),
  "myState" : 1,
  "term" : NumberLong(8),
  "syncSourceHost" : "",
  "syncSourceId" : -1,
  "heartbeatIntervalMillis" : NumberLong(2000),
  "majorityVoteCount" : 2,
  "writeMajorityCount" : 2,
  "votingMembersCount" : 3,
  "writableVotingMembersCount" : 3,
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1731910273, 2),
      "t" : NumberLong(8)
    },
    "lastCommittedWallTime" : ISODate("2024-11-18T06:11:13.719Z"),
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1731910273, 2),
      "t" : NumberLong(8)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1731910273, 2),
      "t" : NumberLong(8)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1731910273, 2),
      "t" : NumberLong(8)
    },
    "lastAppliedWallTime" : ISODate("2024-11-18T06:11:13.719Z"),
    "lastDurableWallTime" : ISODate("2024-11-18T06:11:13.719Z")
  },
  "lastStableRecoveryTimestamp" : Timestamp(1731910243, 2),
  "electionCandidateMetrics" : {
    "lastElectionReason" : "electionTimeout",
    "lastElectionDate" : ISODate("2024-11-18T06:08:03.686Z"),
    "electionTerm" : NumberLong(8),
    "lastCommittedOpTimeAtElection" : {
      "ts" : Timestamp(0, 0),
      "t" : NumberLong(-1)
    },
    "lastSeenOpTimeAtElection" : {
```

```
1 192.168.133.133 x 2 192.168.133.134 x 3 192.168.133.135 x +
"members" : [
  {
    "id" : 0,
    "name" : "192.168.133.133:27001",
    "health" : 1,
    "state" : 1,
    "stateStr" : "PRIMARY",
    "uptime" : 209,
    "optime" : {
      "ts" : Timestamp(1731910273, 2),
      "t" : NumberLong(8)
    },
    "optimeDate" : ISODate("2024-11-18T06:11:13Z"),
    "lastAppliedWallTime" : ISODate("2024-11-18T06:11:13.719Z"),
    "lastDurableWallTime" : ISODate("2024-11-18T06:11:13.719Z"),
    "syncSourceHost" : "",
    "syncSourceId" : -1,
    "infoMessage" : "",
    "electionTime" : Timestamp(1731910083, 1),
    "electionDate" : ISODate("2024-11-18T06:08:03Z"),
    "configVersion" : 1,
    "configTerm" : 8,
    "self" : true,
    "lastHeartbeatMessage" : ""
  },
  {
    "id" : 1,
    "name" : "192.168.133.134:27001",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 205,
    "optime" : {
      "ts" : Timestamp(1731910273, 2),
      "t" : NumberLong(8)
    },
    "optimeDurable" : {
      "ts" : Timestamp(1731910273, 2),
      "t" : NumberLong(8)
    },
    "optimeDate" : ISODate("2024-11-18T06:11:13Z"),
    "optimeDurableDate" : ISODate("2024-11-18T06:11:13Z"),
  }
]
```

### 1.3.2.5启动shard1.conf分片服务

在3台机器的相同端口形成一个分片的复制集，由于3台机器都需要这3个文件，所以根据这9个配置文件分别启动分片服务

```
mongod -f /home/zs/mongodb6/conf/shard2.conf
```

以上命令要分别启动三台服务器。

```
zss@node1:~$ mongod -f /home/zs/mongodb6/conf/shard2.conf
about to fork child process, waiting until server is ready for connections.
forked process: 8660
child process started successfully, parent exiting
```

```
zss@node2:~$ mongod -f /home/zs/mongodb6/conf/shard2.conf
about to fork child process, waiting until server is ready for connections.
forked process: 8194
child process started successfully, parent exiting
```

```
zss@node3:~$ mongod -f /home/zs/mongodb6/conf/shard2.conf
about to fork child process, waiting until server is ready for connections.
forked process: 8201
child process started successfully, parent exiting
```

### 1.3.2.6连接mongo

```
mongo --host 192.168.133.133 --port 27002
mongo --host 192.168.133.134 --port 27002
mongo --host 192.168.133.135 --port 27002
```

```
zsz@node1:~$ mongo --host 192.168.133.133 --port 27002
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.133:27002/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:13:40.710+08:00"},"s":"I", "c":"STORAGE", "id":22297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/production-notes-filesystem","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:13:42.098+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:13:42.098+08:00"},"s":"W", "c":"CONTROL", "id":22184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too low", "attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarnings"]}
```

```
zsz@node2:~$ mongo --host 192.168.133.134 --port 27002
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.134:27002/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:13:43.247+08:00"},"s":"I", "c":"STORAGE", "id":22297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/production-notes-filesystem","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:13:44.359+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:13:44.360+08:00"},"s":"W", "c":"CONTROL", "id":22184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too low", "attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarnings"]}
```

```
zsz@node3:~$ mongo --host 192.168.133.135 --port 27002
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.135:27002/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:13:44.547+08:00"},"s":"I", "c":"STORAGE", "id":22297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/production-notes-filesystem","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:13:46.030+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:13:46.030+08:00"},"s":"W", "c":"CONTROL", "id":22184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too low", "attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarnings"]}
```

以上命令要分别在三台机器上执行。

### 1.3.2.7切换数据库

```
use admin
```

以上命令一台机器上执行即可。

```
shard2:SECONDARY> use admin  
switched to db admin
```

### 1.3.2.8初始化复制集

```
rs.initiate({_id:"shard2",members:[{_id:0,host: "192.168.133.133:27002"},  
{_id:1,host: "192.168.133.134:27002"},{_id:2,host:  
"192.168.133.135:27002"}]})
```

```
shard2:SECONDARY> rs.initiate({_id:"shard2",members:[{_id:0,host: "192.168.133.133:27002"},{_id:1,host: "192.168.133.134:27002"},{_id:2,host: "192.168.133.135:27002"}]})  
{  
  "ok" : 0,  
  "errmsg" : "already initialized",  
  "code" : 23,  
  "codeName" : "AlreadyInitialized",  
  "$clusterTime" : {  
    "clusterTime" : Timestamp(1731910736, 1),  
    "signature" : {  
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),  
      "keyid" : NumberLong(0)  
    },  
    "operationTime" : Timestamp(1731910736, 1)  
  },  
}
```

```
shard2:SECONDARY> rs.status()
{
  "set" : "shard2",
  "date" : ISODate("2024-11-18T06:19:43.466Z"),
  "myState" : 2,
  "term" : NumberLong(9),
  "syncSourceHost" : "192.168.133.133:27002",
  "syncSourceId" : 0,
  "heartbeatIntervalMillis" : NumberLong(2000),
  "majorityVoteCount" : 2,
  "writeMajorityCount" : 2,
  "votingMembersCount" : 3,
  "writableVotingMembersCount" : 3,
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1731910776, 1),
      "t" : NumberLong(9)
    },
    "lastCommittedWallTime" : ISODate("2024-11-18T06:19:36.530Z"),
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1731910776, 1),
      "t" : NumberLong(9)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1731910776, 1),
      "t" : NumberLong(9)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1731910776, 1),
      "t" : NumberLong(9)
    },
    "lastAppliedWallTime" : ISODate("2024-11-18T06:19:36.530Z"),
    "lastDurableWallTime" : ISODate("2024-11-18T06:19:36.530Z")
  },
  "lastStableRecoveryTimestamp" : Timestamp(1731910716, 1),
  "electionParticipantMetrics" : {
    "votedForCandidate" : true,
    "electionTerm" : NumberLong(9),
    "lastVoteDate" : ISODate("2024-11-18T06:13:56.853Z"),
    "electionCandidateMemberId" : 0,
    "voteReason" : ""
  }
}
```

```

    },
    "members" : [
      {
        "id" : 0,
        "name" : "192.168.133.133:27002",
        "health" : 1,
        "state" : 1,
        "stateStr" : "PRIMARY",
        "uptime" : 359,
        "optime" : {
          "ts" : Timestamp(1731910776, 1),
          "t" : NumberLong(9)
        },
        "optimeDurable" : {
          "ts" : Timestamp(1731910776, 1),
          "t" : NumberLong(9)
        },
        "optimeDate" : ISODate("2024-11-18T06:19:36Z"),
        "optimeDurableDate" : ISODate("2024-11-18T06:19:36Z"),
        "lastAppliedWallTime" : ISODate("2024-11-18T06:19:36.530Z"),
        "lastDurableWallTime" : ISODate("2024-11-18T06:19:36.530Z"),
        "lastHeartbeat" : ISODate("2024-11-18T06:19:42.114Z"),
        "lastHeartbeatRecv" : ISODate("2024-11-18T06:19:42.821Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncSourceHost" : "",
        "syncSourceId" : -1,
        "infoMessage" : "",
        "electionTime" : Timestamp(1731910436, 1),
        "electionDate" : ISODate("2024-11-18T06:13:56Z"),
        "configVersion" : 1,
        "configTerm" : 9
      },
      {
        "id" : 1,
        "name" : "192.168.133.134:27002",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 360,
        "optime" : {

```

### 1.3.2.9启动shard3.conf分片服务

在3台机器的相同端口形成一个分片的复制集，由于3台机器都需要这3个文件，所以根据这9个配置文件分别启动分片服务

```
mongod -f /home/zzs/mongodb6/conf/shard3.conf
```

以上命令要分别启动三台服务器。

```

zzs@node1:~$ mongod -f /home/zzs/mongodb6/conf/shard3.conf
about to fork child process, waiting until server is ready for connections.
forked process: 8930
child process started successfully, parent exiting

```

```

zzs@node2:~$ mongod -f /home/zzs/mongodb6/conf/shard3.conf
about to fork child process, waiting until server is ready for connections.
forked process: 8463
child process started successfully, parent exiting

```

```

zzs@node3:~$ mongod -f /home/zzs/mongodb6/conf/shard3.conf
about to fork child process, waiting until server is ready for connections.
forked process: 8473
child process started successfully, parent exiting

```



### 1.3.2.10连接mongo

```
mongo --host 192.168.133.133 --port 27003
mongo --host 192.168.133.134 --port 27003
mongo --host 192.168.133.135 --port 27003
```

以上命令要分别在三台机器上执行。

```
zsz@node1: $ mongo --host 192.168.133.133 --port 27003
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.133:27003/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:21:24.874+08:00"},"s":"I", "c":"STORAGE", "id":22297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/production-notes-filesystem","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:21:26.014+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:21:26.014+08:00"},"s":"W", "c":"CONTROL", "id":22184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too low", "attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarnings"]}
```

```
zsz@node2: $ mongo --host 192.168.133.134 --port 27003
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.134:27003/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:21:22.183+08:00"},"s":"I", "c":"STORAGE", "id":22297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/production-notes-filesystem","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:21:23.244+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:21:23.244+08:00"},"s":"W", "c":"CONTROL", "id":22184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too low", "attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarnings"]}
```

```
zsz@node3: $ mongo --host 192.168.133.135 --port 27003
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.135:27003/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:21:18.749+08:00"},"s":"I", "c":"STORAGE", "id":22297, "ctx":"initandlisten","msg":"Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/production-notes-filesystem","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:21:20.181+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"initandlisten","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
{"t":{"$date":"2024-11-18T14:21:20.181+08:00"},"s":"W", "c":"CONTROL", "id":22184, "ctx":"initandlisten","msg":"Soft rlimits for open file descriptors too low", "attr":{"currentValue":1024,"recommendedMinimum":64000},"tags":["startupWarnings"]}
```

### 1.3.2.11切换数据库

```
use admin
```

以上命令一台机器上执行即可。

```
shard3:PRIMARY> use admin  
switched to db admin
```

### 1.3.2.12初始化复制集

```
rs.initiate({_id:"shard3",members:[{_id:0,host: "192.168.133.133:27003"},  
{_id:1,host: "192.168.133.134:27003"},{_id:2,host:  
"192.168.133.135:27003"}]})
```

```
shard3:PRIMARY> rs.initiate({_id:"shard3",members:[{_id:0,host: "192.168.133.133:27003"},{_id:1,host: "192.168.133.134:27003"},{_id:2,host: "192.168.133.135:27003"}]})  
{  
  "ok" : 0,  
  "errmsg" : "already initialized",  
  "code" : 23,  
  "codeName" : "AlreadyInitialized",  
  "$clusterTime" : {  
    "clusterTime" : Timestamp(1731911011, 1),  
    "signature" : {  
      "hash" : BinData(0,"AAAAAAAAAAAAAAAAAAAAAAAAAAAA="),  
      "keyId" : NumberLong(0)  
    }  
  },  
  "operationTime" : Timestamp(1731911011, 1)  
}
```

```

shard3:PRIMARY> rs.status()
{
  "set" : "shard3",
  "date" : ISODate("2024-11-18T06:24:05.307Z"),
  "myState" : 1,
  "term" : NumberLong(7),
  "syncSourceHost" : "",
  "syncSourceId" : -1,
  "heartbeatIntervalMillis" : NumberLong(2000),
  "majorityVoteCount" : 2,
  "writeMajorityCount" : 2,
  "votingMembersCount" : 3,
  "writableVotingMembersCount" : 3,
  "optimes" : {
    "lastCommittedOpTime" : {
      "ts" : Timestamp(1731911041, 1),
      "t" : NumberLong(7)
    },
    "lastCommittedWallTime" : ISODate("2024-11-18T06:24:01.188Z"),
    "readConcernMajorityOpTime" : {
      "ts" : Timestamp(1731911041, 1),
      "t" : NumberLong(7)
    },
    "appliedOpTime" : {
      "ts" : Timestamp(1731911041, 1),
      "t" : NumberLong(7)
    },
    "durableOpTime" : {
      "ts" : Timestamp(1731911041, 1),
      "t" : NumberLong(7)
    },
    "lastAppliedWallTime" : ISODate("2024-11-18T06:24:01.188Z"),
    "lastDurableWallTime" : ISODate("2024-11-18T06:24:01.188Z")
  },
  "lastStableRecoveryTimestamp" : Timestamp(1731910991, 1),
  "electionCandidateMetrics" : {
    "lastElectionReason" : "electionTimeout",
    "lastElectionDate" : ISODate("2024-11-18T06:21:31.170Z"),
    "electionTerm" : NumberLong(7),
    "lastCommittedOpTimeAtElection" : {
      "ts" : Timestamp(0, 0),
      "t" : NumberLong(-1)
    }
  },
}

```

```

    },
    "members" : [
      {
        "id" : 0,
        "name" : "192.168.133.133:27003",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 159,
        "optime" : {
          "ts" : Timestamp(1731911041, 1),
          "t" : NumberLong(7)
        },
        "optimeDurable" : {
          "ts" : Timestamp(1731911041, 1),
          "t" : NumberLong(7)
        },
        "optimeDate" : ISODate("2024-11-18T06:24:01Z"),
        "optimeDurableDate" : ISODate("2024-11-18T06:24:01Z"),
        "lastAppliedWallTime" : ISODate("2024-11-18T06:24:01.188Z"),
        "lastDurableWallTime" : ISODate("2024-11-18T06:24:01.188Z"),
        "lastHeartbeat" : ISODate("2024-11-18T06:24:05.277Z"),
        "lastHeartbeatRecv" : ISODate("2024-11-18T06:24:04.285Z"),
        "pingMs" : NumberLong(0),
        "lastHeartbeatMessage" : "",
        "syncSourceHost" : "192.168.133.135:27003",
        "syncSourceId" : 2,
        "infoMessage" : "",
        "configVersion" : 1,
        "configTerm" : 7
      },
      {
        "id" : 1,
        "name" : "192.168.133.134:27003",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 161,
        "optime" : {
          "ts" : Timestamp(1731911041, 1),
          "t" : NumberLong(7)
        }
      }
    ]
  },
  {
    "id" : 2,
    "name" : "192.168.133.135:27003",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 161,
    "optime" : {
      "ts" : Timestamp(1731911041, 1),
      "t" : NumberLong(7)
    }
  }
]
}

```

## 1.4 路由服务部署(3台服务器执行相同操作)

### 1.4.1配置mongos.conf文件

在/home/zzs/mongodb6/conf目录创建mongos.conf。

```
touch /home/zzs/mongodb6/conf/mongos.conf
```

```
vi /home/zzs/mongodb6/conf/mongos.conf
```

内容如下：

```
logpath=/home/zzs/mongodb6/log/mongos.log
logappend = true
port = 27017
fork = true
configdb =
configs/192.168.133.133:27018,192.168.133.134:27018,192.168.133.135:2701
8
maxConns=20000
bind_ip=0.0.0.0
```

```
logpath=/home/zzs/mongodb6/log/mongos. log
logappend = true
port = 27017
fork = true
configdb = configs/192.168.133.133:27018,192.168.133.134:27018,192.168.133.135:27018
maxConns=20000
bind_ip=0.0.0.0
```

```
logpath=/home/zzs/mongodb6/log/mongos.log
logappend = true
port = 27017
fork = true
configdb = configs/192.168.133.133:27018,192.168.133.134:27018,192.168.133.135:27018
maxConns=20000
bind_ip=0.0.0.0
```

~/mongodb6/conf/mongos.conf 7L, 200C 7, 15 全部



### 1.4.3.1连接mongo

```
mongo --host 192.168.133.133 --port 27017
mongo --host 192.168.133.134 --port 27017
mongo --host 192.168.133.135 --port 27017
```

```
zszs@node1:~$ mongo --host 192.168.133.133 --port 27017
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.133:27017/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:26:59.584+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"main","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
mongos>
```

```
zszs@node2:~$ mongo --host 192.168.133.134 --port 27017
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.134:27017/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:27:04.153+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"main","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
mongos>
```

```
zszs@node3:~$ mongo --host 192.168.133.135 --port 27017
MongoDB shell version v3.6.3
connecting to: mongodb://192.168.133.135:27017/
MongoDB server version: 6.0.19
WARNING: shell and server versions do not match
Server has startup warnings:
{"t":{"$date":"2024-11-18T14:27:05.771+08:00"},"s":"W", "c":"CONTROL", "id":22120, "ctx":"main","msg":"Access control is not enabled for the database. Read and write access to data and configuration is unrestricted","tags":["startupWarnings"]}
mongos>
```

### 1.4.3.2切换数据库

```
use admin
```

```
mongos> use admin
switched to db admin
```

### 1.4.3.3添加分片(只需在一台机器执行即可)

```
sh.addShard("shard1/192.168.133.133:27001,192.168.133.134:27001,192.168.133.135:27001")
```

### 1.4.3.4查看集群状态



```
sh.status()
```

```
mongos> sh.status()
--- Sharding Status ---
  sharding version: {
    "_id" : 1,
    "minCompatibleVersion" : 5,
    "currentVersion" : 6,
    "clusterId" : ObjectId("6721882402554fb31e9e0b55")
  }
  shards:
    { "_id" : "shard1", "host" : "shard1/192.168.133.133:27001,192.168.133.134:27001,192.168.133.135:27001", "state" : 1, "topologyTime" : Timestamp(1730251847, 5) }
  active mongoses:
    "6.0.19" : 3
  autosplit:
    Currently enabled: yes
  balancer:
    Currently enabled: yes
    Currently running: no
    Failed balancer rounds in last 5 attempts: 0
    Migration Results for the last 24 hours:
      No recent migrations
  databases:
    { "_id" : "config", "primary" : "config", "partitioned" : true }
      config.system.sessions
        shard key: { "_id" : 1 }
        unique: false
        balancing: true
        chunks:
```

## 1.5后期运维

mongodb的启动顺序是，先启动配置服务器，在启动分片，最后启动 mongos。

```
mongod -f /home/zzs/mongodb6/conf/config.conf
```

```
mongod -f /home/zzs/mongodb6/conf/shard1.conf
```

```
mongod -f /home/zzs/mongodb6/conf/shard2.conf
```

```
mongod -f /home/zzs/mongodb6/conf/shard3.conf
```

```
mongos -f /home/zzs/mongodb6/conf/mongos.conf
```

```
zsz@node1:~$ mongod -f /home/zsz/mongodb6/conf/config.conf
about to fork child process, waiting until server is ready for connections.
forked process: 14877
child process started successfully, parent exiting
zsz@node1:~$ mongod -f /home/zsz/mongodb6/conf/shard1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 14972
child process started successfully, parent exiting
zsz@node1:~$ mongod -f /home/zsz/mongodb6/conf/shard2.conf
about to fork child process, waiting until server is ready for connections.
forked process: 15115
child process started successfully, parent exiting
zsz@node1:~$ mongod -f /home/zsz/mongodb6/conf/shard3.conf
about to fork child process, waiting until server is ready for connections.
forked process: 15239
child process started successfully, parent exiting
zsz@node1:~$ mongos -f /home/zsz/mongodb6/conf/mongos.conf
about to fork child process, waiting until server is ready for connections.
forked process: 15346
child process started successfully, parent exiting
```

```
zsz@node2:~$ mongod -f /home/zsz/mongodb6/conf/config.conf
about to fork child process, waiting until server is ready for connections.
forked process: 4897
child process started successfully, parent exiting
zsz@node2:~$ mongod -f /home/zsz/mongodb6/conf/shard1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 4990
ERROR: child process failed, exited with 48
To see additional information in this output, start without the "--fork" option.
zsz@node2:~$ mongod -f /home/zsz/mongodb6/conf/shard2.conf
about to fork child process, waiting until server is ready for connections.
forked process: 5014
child process started successfully, parent exiting
zsz@node2:~$ mongod -f /home/zsz/mongodb6/conf/shard3.conf
about to fork child process, waiting until server is ready for connections.
forked process: 5098
child process started successfully, parent exiting
zsz@node2:~$ mongos -f /home/zsz/mongodb6/conf/mongos.conf
about to fork child process, waiting until server is ready for connections.
forked process: 5194
child process started successfully, parent exiting
```

```
zsz@node3:~$ mongod -f /home/zsz/mongodb6/conf/config.conf
about to fork child process, waiting until server is ready for connections.
forked process: 10483
child process started successfully, parent exiting
zsz@node3:~$ mongod -f /home/zsz/mongodb6/conf/shard1.conf
about to fork child process, waiting until server is ready for connections.
forked process: 10586
child process started successfully, parent exiting
zsz@node3:~$ mongod -f /home/zsz/mongodb6/conf/shard2.conf
about to fork child process, waiting until server is ready for connections.
forked process: 10696
child process started successfully, parent exiting
zsz@node3:~$ mongod -f /home/zsz/mongodb6/conf/shard3.conf
about to fork child process, waiting until server is ready for connections.
forked process: 10790
child process started successfully, parent exiting
zsz@node3:~$ mongos -f /home/zsz/mongodb6/conf/mongos.conf
about to fork child process, waiting until server is ready for connections.
forked process: 10901
child process started successfully, parent exiting
```

关闭时，直接killall杀掉所有进程

```
killall mongod
```

```
killall mongos
```

```
zsz@node1:~$ killall mongod  
zsz@node1:~$ killall mongos  
zsz@node1:~$
```

```
zsz@node2:~$ killall mongod  
zsz@node2:~$ killall mongos  
zsz@node2:~$
```

```
zsz@node3:~$ killall mongod  
zsz@node3:~$ killall mongos  
zsz@node3:~$
```