**注意事项（坑）：**

1、必须允许其他的mongo服务器ip访问或者关闭防火墙，我已经把开启端口的命令做出一个.sh文件放在我的电脑上

2、在使用xshell通过windows向linux传 .conf .service文件时，注意文件的编码，这里我修改文件编码为utf-8时反而出错，使用windows直接新建文本文档默认的ASCII编码反而正确。

报错内容大致是说你的文件有不识别的行

unknow line '' 这样的错误

3、配置文件在我的电脑上有，因为我是root启动的，并且安装路径也不一样，因此对配置文件均有修改

配置参考文档：<https://blog.csdn.net/ilovemilk/article/details/79336951>

**以下为配置参考文档全文**

mongodb是最常用的noSql数据库，在数据库排名中已经上升到了前五。这篇文章介绍如何搭建高可用的mongodb（分片+副本）集群。

在搭建集群之前，需要首先了解几个概念：路由，分片、副本集、配置服务器等。

相关概念

mongodb集群架构图：

从图中可以看到有四个组件：mongos、config server、shard、replica set。

mongos，数据库集群请求的入口，所有的请求都通过mongos进行协调，不需要在应用程序添加一个路由选择器，mongos自己就是一个请求分发中心，它负责把对应的数据请求请求转发到对应的shard服务器上。在生产环境通常有多mongos作为请求的入口，防止其中一个挂掉所有的mongodb请求都没有办法操作。

config server，顾名思义为配置服务器，存储所有数据库元信息（路由、分片）的配置。mongos本身没有物理存储分片服务器和数据路由信息，只是缓存在内存里，配置服务器则实际存储这些数据。mongos第一次启动或者关掉重启就会从 config server 加载配置信息，以后如果配置服务器信息变化会通知到所有的 mongos 更新自己的状态，这样 mongos 就能继续准确路由。在生产环境通常有多个 config server 配置服务器，因为它存储了分片路由的元数据，防止数据丢失！

shard，分片（sharding）是指将数据库拆分，将其分散在不同的机器上的过程。将数据分散到不同的机器上，不需要功能强大的服务器就可以存储更多的数据和处理更大的负载。基本思想就是将集合切成小块，这些块分散到若干片里，每个片只负责总数据的一部分，最后通过一个均衡器来对各个分片进行均衡（数据迁移）。

replica set，中文翻译副本集，其实就是shard的备份，防止shard挂掉之后数据丢失。复制提供了数据的冗余备份，并在多个服务器上存储数据副本，提高了数据的可用性， 并可以保证数据的安全性。

仲裁者（Arbiter），是复制集中的一个MongoDB实例，它并不保存数据。仲裁节点使用最小的资源并且不要求硬件设备，不能将Arbiter部署在同一个数据集节点中，可以部署在其他应用服务器或者监视服务器中，也可部署在单独的虚拟机中。为了确保复制集中有奇数的投票成员（包括primary），需要添加仲裁节点做为投票，否则primary不能运行时不会自动切换primary。

简单了解之后，我们可以这样总结一下，应用请求mongos来操作mongodb的增删改查，配置服务器存储数据库元信息，并且和mongos做同步，数据最终存入在shard（分片）上，为了防止数据丢失同步在副本集中存储了一份，仲裁在数据存储到分片的时候决定存储到哪个节点。

**环境准备**

系统系统 centos 7.4

三台服务器：192.168.56.101、node2(192.168.56.102)、node3(192.168.56.103)

安装包：yum安装

服务器规划

|  |  |  |
| --- | --- | --- |
| 服务器node1(192.168.56.101) | 服务器node2(192.168.56.102) | 服务器node3(192.168.56.103) |
| mongos | mongos | mongos |
| config server | config server | config server |
| shard server1 主节点 | shard server1 副节点 | shard server1 仲裁 |
| shard server2 仲裁 | shard server2 主节点 | shard server2 副节点 |
| shard server3 副节点 | shard server3 仲裁 | shard server3 主节点 |

 端口分配：

mongos：20000

config：21000

shard1：27001

shard2：27002

shard3：27003

开放端口

firewall-cmd --add-port=21000/tcp --permanent

firewall-cmd --add-port=21000/tcp

firewall-cmd --add-port=27001/tcp --permanent

firewall-cmd --add-port=27001/tcp

firewall-cmd --add-port=27002/tcp --permanent

firewall-cmd --add-port=27002/tcp

firewall-cmd --add-port=27003/tcp --permanent

firewall-cmd --add-port=27003/tcp

firewall-cmd --add-port=20000/tcp --permanent

firewall-cmd --add-port=20000/tcp

关闭selinux

vi /etc/selinux/config

SELINUX=disabled

重启系统

**集群搭建**

**1、安装mongodb**

#yum安装mongodb

cat /etc/yum.repos.d/mongodb.repo

[mongodb-org]

name=MongoDB Repository

baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.6/x86\_64/

gpgcheck=1

enabled=1

gpgkey=https://www.mongodb.org/static/pgp/server-3.6.asc

yum install mongodb-org

禁用自带服务：

systemctl disable mongod

systemctl stop mongod

路径规划并创建

分别在每台机器建立conf、mongos、config、shard1、shard2、shard3目录，因为mongos不存储数据，只需要建立日志文件目录即可。

配置文件路径：

mkdir -p /etc/mongod/conf.d

pid文件路径

/var/run/mongodb

数据存储路径

#config server数据存储路径

mkdir -p /var/lib/mongo/config/data

#shard server数据存储路径

mkdir -p /var/lib/mongo/shard1/data

mkdir -p /var/lib/mongo/shard2/data

mkdir -p /var/lib/mongo/shard3/data

chown -R mongod:mongod /var/lib/mongo

日志文件路径

/var/log/mongodb

**2、config server配置服务器**

mongodb3.4以后要求配置服务器也创建副本集，不然集群搭建不成功。

添加配置文件

vi /etc/mongod/conf.d/config.conf

## 配置文件内容

systemLog:

  destination: file

  logAppend: true

  path: /var/log/mongodb/configsvr.log

storage:

  dbPath: /var/lib/mongo/config/data

  journal:

    enabled: true

processManagement:

  fork: true

  pidFilePath: /var/run/mongodb/configsvr.pid

  timeZoneInfo: /usr/share/zoneinfo

net:

  port: 21000

  bindIp: 0.0.0.0

  maxIncomingConnections: 20000

replication:

  replSetName: csReplSet

sharding:

  clusterRole: configsvr

创建systemctl unit文件mongod-configsvr.service

**?如何使用.service文件？**

**$ cd /etc/systemd/system**

**拷贝.service文件到这个文件夹目录下**

**$ systemctl enable mongod-configsvr.service**

**这就注册了一个service， 通过以下语句来启动service**

**systemctl start mongd-configsrv**

[Unit]

Description=Mongodb Config Server

After=network.target

Documentation=https://docs.mongodb.org/manual

[Service]

User=mongod

Group=mongod

Environment="OPTIONS=-f /etc/mongod/conf.d/config.conf"

ExecStart=/usr/bin/mongod $OPTIONS

ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb

ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb

ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb

PermissionsStartOnly=true

PIDFile=/var/run/mongodb/configsvr.pid

Type=forking

# file size

LimitFSIZE=infinity

# cpu time

LimitCPU=infinity

# virtual memory size

LimitAS=infinity

# open files

LimitNOFILE=64000

# processes/threads

LimitNPROC=64000

# locked memory

LimitMEMLOCK=infinity

# total threads (user+kernel)

TasksMax=infinity

TasksAccounting=false

# Recommended limits for for mongod as specified in

# http://docs.mongodb.org/manual/reference/ulimit/#recommended-settings

[Install]

WantedBy=multi-user.target

启动三台服务器的config server

systemctl enable mongod-configsvr

systemctl start mongod-configsvr

登录任意一台配置服务器，初始化配置副本集

#连接

mongo 127.0.0.1:21000

#config变量

config = {

        \_id : "csReplSet",

        members : [

          {\_id : 1, host : "192.168.56.101:21000" },

          {\_id : 2, host : "192.168.56.102:21000" },

          {\_id : 3, host : "192.168.56.103:21000" }

        ]

}

#初始化副本集

rs.initiate(config)

其中，"\_id" : " csReplSet "应与配置文件中配置的 replicaction.replSetName 一致，"members" 中的 "host" 为三个节点的 ip 和 port

**3、配置分片副本集(三台机器)**

1) 设置第一个分片副本集

添加配置文件

vi /etc/mongod/conf.d/shard1.conf

## 配置文件内容

systemLog:

  destination: file

  logAppend: true

  path: /var/log/mongodb/shard1.log

storage:

  dbPath: /var/lib/mongo/shard1/data

  journal:

    enabled: true

processManagement:

  fork: true

  pidFilePath: /var/run/mongodb/shard1.pid

  timeZoneInfo: /usr/share/zoneinfo

net:

  port: 27001

  bindIp: 0.0.0.0

  maxIncomingConnections: 20000

replication:

  replSetName: shard1

sharding:

  clusterRole: shardsvr

创建systemctl unit文件mongod-shard1.service

[Unit]

Description=Mongodb Shard1 Server

After=network.target

Documentation=https://docs.mongodb.org/manual

[Service]

User=mongod

Group=mongod

Environment="OPTIONS=-f /etc/mongod/conf.d/shard1.conf"

ExecStart=/usr/bin/mongod $OPTIONS

ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb

ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb

ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb

PermissionsStartOnly=true

PIDFile=/var/run/mongodb/shard1.pid

Type=forking

# file size

LimitFSIZE=infinity

# cpu time

LimitCPU=infinity

# virtual memory size

LimitAS=infinity

# open files

LimitNOFILE=64000

# processes/threads

LimitNPROC=64000

# locked memory

LimitMEMLOCK=infinity

# total threads (user+kernel)

TasksMax=infinity

TasksAccounting=false

# Recommended limits for for mongod as specified in

# http://docs.mongodb.org/manual/reference/ulimit/#recommended-settings

[Install]

WantedBy=multi-user.target

启动三台服务器的shard1 server

systemctl enable mongod-shard1

systemctl start mongod-shard1

登陆任意一台非仲裁节点服务器，初始化副本集

mongo 127.0.0.1:27001

#使用admin数据库

use admin

#定义副本集配置，第三个节点的 "arbiterOnly":true 代表其为仲裁节点。

config = {

       \_id : "shard1",

       members : [

            {\_id : 1, host : "192.168.56.101:27001" , priority:2},

            {\_id : 2, host : "192.168.56.102:27001" , priority:1},

            {\_id : 3, host : "192.168.56.103:27001" , arbiterOnly: true }

        ]

    }

#初始化副本集配置

rs.initiate(config);

2) 设置第二个分片副本集

添加配置文件

vi /etc/mongod/conf.d/shard2.conf

## 配置文件内容

systemLog:

  destination: file

  logAppend: true

  path: /var/log/mongodb/shard2.log

storage:

  dbPath: /var/lib/mongo/shard2/data

  journal:

    enabled: true

processManagement:

  fork: true

  pidFilePath: /var/run/mongodb/shard2.pid

  timeZoneInfo: /usr/share/zoneinfo

net:

  port: 27002

  bindIp: 0.0.0.0

  maxIncomingConnections: 20000

replication:

  replSetName: shard2

sharding:

  clusterRole: shardsvr

创建systemctl unit文件mongod-shard2.service

[Unit]

Description=Mongodb Shard2 Server

After=network.target

Documentation=https://docs.mongodb.org/manual

[Service]

User=mongod

Group=mongod

Environment="OPTIONS=-f /etc/mongod/conf.d/shard2.conf"

ExecStart=/usr/bin/mongod $OPTIONS

ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb

ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb

ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb

PermissionsStartOnly=true

PIDFile=/var/run/mongodb/shard2.pid

Type=forking

# file size

LimitFSIZE=infinity

# cpu time

LimitCPU=infinity

# virtual memory size

LimitAS=infinity

# open files

LimitNOFILE=64000

# processes/threads

LimitNPROC=64000

# locked memory

LimitMEMLOCK=infinity

# total threads (user+kernel)

TasksMax=infinity

TasksAccounting=false

# Recommended limits for for mongod as specified in

# http://docs.mongodb.org/manual/reference/ulimit/#recommended-settings

[Install]

WantedBy=multi-user.target

启动三台服务器的shard2 server

systemctl enable mongod-shard2

systemctl start mongod-shard2

登陆任意一台非仲裁节点服务器，初始化副本集

mongo 127.0.0.1:27002

#使用admin数据库

use admin

#定义副本集配置，第一个节点的 "arbiterOnly":true 代表其为仲裁节点。

config = {

       \_id : "shard2",

       members : [

            {\_id : 1, host : "192.168.56.101:27002" , arbiterOnly: true },

            {\_id : 2, host : "192.168.56.102:27002" , priority:2},

            {\_id : 3, host : "192.168.56.103:27002" , priority:1}

        ]

    }

#初始化副本集配置

rs.initiate(config);

3) 设置第三个分片副本集

添加配置文件

vi /etc/mongod/conf.d/shard3.conf

## 配置文件内容

systemLog:

  destination: file

  logAppend: true

  path: /var/log/mongodb/shard3.log

storage:

  dbPath: /var/lib/mongo/shard3/data

  journal:

    enabled: true

processManagement:

  fork: true

  pidFilePath: /var/run/mongodb/shard3.pid

  timeZoneInfo: /usr/share/zoneinfo

net:

  port: 27003

  bindIp: 0.0.0.0

  maxIncomingConnections: 20000

replication:

  replSetName: shard3

sharding:

  clusterRole: shardsvr

创建systemctl unit文件mongod-shard3.service

[Unit]

Description=Mongodb Shard3 Server

After=network.target

Documentation=https://docs.mongodb.org/manual

[Service]

User=mongod

Group=mongod

Environment="OPTIONS=-f /etc/mongod/conf.d/shard3.conf"

ExecStart=/usr/bin/mongod $OPTIONS

ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb

ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb

ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb

PermissionsStartOnly=true

PIDFile=/var/run/mongodb/shard3.pid

Type=forking

# file size

LimitFSIZE=infinity

# cpu time

LimitCPU=infinity

# virtual memory size

LimitAS=infinity

# open files

LimitNOFILE=64000

# processes/threads

LimitNPROC=64000

# locked memory

LimitMEMLOCK=infinity

# total threads (user+kernel)

TasksMax=infinity

TasksAccounting=false

# Recommended limits for for mongod as specified in

# http://docs.mongodb.org/manual/reference/ulimit/#recommended-settings

[Install]

WantedBy=multi-user.target

启动三台服务器的shard3 server

systemctl enable mongod-shard3

systemctl start mongod-shard3

登陆任意一台非仲裁节点服务器，初始化副本集

mongo 127.0.0.1:27003

#使用admin数据库

use admin

#定义副本集配置，第二个节点的 "arbiterOnly":true 代表其为仲裁节点。

config = {

       \_id : "shard3",

       members : [

            {\_id : 1, host : "192.168.56.101:27003" , priority:1},

            {\_id : 2, host : "192.168.56.102:27003" , arbiterOnly: true},

            {\_id : 3, host : "192.168.56.103:27003" , priority:2}

        ]

    }

#初始化副本集配置

rs.initiate(config);

**4、配置路由服务器 mongos**

先启动配置服务器和分片服务器,后启动路由实例启动路由实例:（三台机器）

vi /etc/mongod/conf.d/mongos.conf

## 配置文件内容

systemLog:

  destination: file

  logAppend: true

  path: /var/log/mongodb/mongos.log

processManagement:

  fork: true

  pidFilePath: /var/run/mongodb/mongos.pid

  timeZoneInfo: /usr/share/zoneinfo

net:

  port: 20000

  bindIp: 0.0.0.0

  maxIncomingConnections: 20000

sharding:

  configDB: csReplSet/192.168.56.101:21000, 192.168.56.102:21000, 192.168.56.103:21000

#注意监听的配置服务器,只能有1个或者3个 csReplSet为配置服务器的副本集名字

configDB: csReplSet/192.168.56.101:21000, 192.168.56.102:21000, 192.168.56.103:21000

(只是做个提示，这句不放进 .conf 文件)

创建systemctl unit文件mongod-mongos.service

[Unit]

Description=Mongodb Mongos Server

After=network.target mongod-shard1.service mongod-shard2.service mongod-shard3.service

Documentation=https://docs.mongodb.org/manual

[Service]

User=mongod

Group=mongod

Environment="OPTIONS=-f /etc/mongod/conf.d/mongos.conf"

ExecStart=/usr/bin/mongos $OPTIONS

ExecStartPre=/usr/bin/mkdir -p /var/run/mongodb

ExecStartPre=/usr/bin/chown mongod:mongod /var/run/mongodb

ExecStartPre=/usr/bin/chmod 0755 /var/run/mongodb

PermissionsStartOnly=true

PIDFile=/var/run/mongodb/mongos.pid

Type=forking

# file size

LimitFSIZE=infinity

# cpu time

LimitCPU=infinity

# virtual memory size

LimitAS=infinity

# open files

LimitNOFILE=64000

# processes/threads

LimitNPROC=64000

# locked memory

LimitMEMLOCK=infinity

# total threads (user+kernel)

TasksMax=infinity

TasksAccounting=false

# Recommended limits for for mongod as specified in

# http://docs.mongodb.org/manual/reference/ulimit/#recommended-settings

[Install]

WantedBy=multi-user.target

启动三台服务器的mongos server

systemctl enable mongod-mongos

systemctl start mongod-mongos

**5、启用分片**

目前搭建了mongodb配置服务器、路由服务器，各个分片服务器，不过应用程序连接到mongos路由服务器并不能使用分片机制，还需要在程序里设置分片配置，让分片生效。

登陆任意一台mongos

mongo 127.0.0.1:20000

#使用admin数据库

user  admin

#串联路由服务器与分配副本集

sh.addShard("shard1/192.168.56.101:27001,192.168.56.102:27001,192.168.56.103:27001")

sh.addShard("shard2/192.168.56.101:27002,192.168.56.102:27002,192.168.56.103:27002")

sh.addShard("shard3/192.168.56.101:27003,192.168.56.102:27003,192.168.56.103:27003")

#查看集群状态

sh.status()

mongos> sh.status()

--- Sharding Status ---

  sharding version: {

  "\_id" : 1,

  "minCompatibleVersion" : 5,

  "currentVersion" : 6,

  "clusterId" : ObjectId("5a86b6255d128f35cb22de20")

  }

  shards:

        {  "\_id" : "shard1",  "host" : "shard1/192.168.56.101:27001,192.168.56.102:27001",  "state" : 1 }

        {  "\_id" : "shard2",  "host" : "shard2/192.168.56.102:27002,192.168.56.103:27002",  "state" : 1 }

        {  "\_id" : "shard3",  "host" : "shard3/192.168.56.101:27003,192.168.56.103:27003",  "state" : 1 }

  active mongoses:

        "3.6.2" : 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 }

**6、测试**

目前配置服务、路由服务、分片服务、副本集服务都已经串联起来了，但我们的目的是希望插入数据，数据能够自动分片。连接在mongos上，准备让指定的数据库、指定的集合分片生效。

mongo 127.0.0.1:20000

#设置分片chunk大小

use config

db.settings.save({ "\_id" : "chunksize", "value" : 1 })

设置1M是为了测试，否则要插入大量数据才能分片。

#指定test分片生效

sh.enableSharding("test")

#指定数据库里需要分片的集合和片键

use test

db.users.createIndex({user\_id : 1})

use admin

sh.shardCollection("test.users", {user\_id: 1})

我们设置testdb的 table1 表需要分片，根据 user\_id 自动分片到 shard1 ，shard2，shard3 上面去。要这样设置是因为不是所有mongodb 的数据库和表 都需要分片！

测试分片配置结果

mongo  127.0.0.1:20000

use test;

for (var i = 1; i <=1000000; i++){

db.users.save({user\_id: i, username: "user"+i});

}

#查看分片情况如下，部分无关信息省掉了

sh.status()

--- Sharding Status ---

  sharding version: {

  "\_id" : 1,

  "minCompatibleVersion" : 5,

  "currentVersion" : 6,

  "clusterId" : ObjectId("5a8919101e5f8edd5feddf36")

  }

  shards:

        {  "\_id" : "shard1",  "host" : "shard1/192.168.56.101:27001,192.168.56.102:27001",  "state" : 1 }

        {  "\_id" : "shard2",  "host" : "shard2/192.168.56.102:27002,192.168.56.103:27002",  "state" : 1 }

        {  "\_id" : "shard3",  "host" : "shard3/192.168.56.101:27003,192.168.56.103:27003",  "state" : 1 }

  active mongoses:

        "3.6.2" : 3

  autosplit:

        Currently enabled: yes

  balancer:

        Currently enabled:  yes

        Currently running:  no

        Collections with active migrations:

                test.users started at Sun Feb 18 2018 21:14:30 GMT+0800 (CST)

        Failed balancer rounds in last 5 attempts:  0

        Migration Results for the last 24 hours:

                43 : Success

                1 : Failed with error 'aborted', from shard1 to shard3

                139 : Failed with error 'aborted', from shard1 to shard2

  databases:

…

        {  "\_id" : "test",  "primary" : "shard3",  "partitioned" : true }

                test.users

                        shard key: { "user\_id" : 1 }

                        unique: false

                        balancing: true

                        chunks:

                                shard1 34

                                shard2 31

                                shard3 31

                        too many chunks to print, use verbose if you want to force print

可以看到数据分到3个分片，各自分片chunk 数为： shard1 : 34，shard2 : 31，shard3 :  31。已经成功了！

**后期运维**

**启动**

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.

systemctl start mongod-configsvr

systemctl start mongod-shard1

systemctl start mongod-shard2

systemctl start mongod-shard3

systemctl start mongod-mongos

关闭:

systemctl stop mongod-mongos

systemctl stop mongod-shard3

systemctl stop mongod-shard2

systemctl stop mongod-shard1

systemctl stop mongod-configsvr

--------------------- 本文来自 ilovemilk 的CSDN 博客 ，全文地址请点击：https://blog.csdn.net/ilovemilk/article/details/79336951?utm\_source=copy

片键选择参考文档：<https://blog.csdn.net/long12310225/article/details/77689460>