REDIS集群

环境：redhat7.5操作系统，关闭防火墙，selinux，配置好yum源

说明：由于环境机器限制问题，本处采用redis实例的方式模拟集群

软件：redis-4.0.8.tar.gz redis-3.2.1.gem

算法说明：redis集群采用投票制度，半数以上的节点投票才能时得fail的节点被踢出集群，如果时两个节点，一个节点挂掉的时候只有半数投票，不足以将挂掉的节点踢出，所以主节点至少需要有3个，此外，集群采用主从结构，要求每个主节点至少有1个从节点，保证主节点挂掉以后从能升级成主，所以还需要至少3个从节点，宗上，构建redis集群最少需要6个节点。

[root@server10 ~]# ls

redis-3.2.1.gem redis-4.0.8.tar.gz ruby-devel-2.0.0.648-30.el7.x86\_64.rpm

[root@server10 ~]# yum -y install gcc gcc-c++

[root@server10 ~]# tar -xf redis-4.0.8.tar.gz

[root@server10 ~]# cd redis-4.0.8/

[root@server10 redis-4.0.8]# make

[root@server10 redis-4.0.8]# make install PREFIX=/usr/local/redis-cluster

[root@server10 redis-4.0.8]# cd /usr/local/redis-cluster/

[root@server10 redis-cluster]# ls

bin

[root@server10 redis-cluster]# cp /root/redis-4.0.8/redis.conf bin/

#创建第一个实例

[root@server10 redis-cluster]# mv bin/ redis01

#修改配置文件

[root@server10 redis-cluster]# ls redis01/

redis-benchmark redis-check-aof redis-check-rdb redis-cli redis.conf redis-sentinel redis-server

[root@server10 redis-cluster]# vim redis01/redis.conf

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis01/redis.conf

port 7001

daemonize yes

pidfile /var/run/redis\_7001.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7001.conf

cluster-node-timeout 5000

#复制多个实例并修改每个实例的配置文件

[root@server10 redis-cluster]# ls

redis01

[root@server10 redis-cluster]# for i in {2..6}

> do

> cp -r redis01/ redis0$i

> done

[root@server10 redis-cluster]# ls

redis01 redis02 redis03 redis04 redis05 redis06

[root@server10 redis-cluster]# vim redis02/redis.conf

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis02/redis.conf

port 7002

daemonize yes

pidfile /var/run/redis\_7002.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7002.conf

cluster-node-timeout 5000

[root@server10 redis-cluster]# vim redis03/redis.conf

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis03/redis.conf

port 7003

daemonize yes

pidfile /var/run/redis\_7003.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7003.conf

cluster-node-timeout 5000

[root@server10 redis-cluster]# vim redis04/redis.conf

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis04/redis.conf

port 7004

daemonize yes

pidfile /var/run/redis\_7004.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7004.conf

cluster-node-timeout 5000

[root@server10 redis-cluster]# vim redis05/redis.conf

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis05/redis.conf

port 7005

daemonize yes

pidfile /var/run/redis\_7005.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7005.conf

cluster-node-timeout 5000

[root@server10 redis-cluster]# vim redis06/redis.conf

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis06/redis.conf

port 7006

daemonize yes

pidfile /var/run/redis\_7006.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7006.conf

cluster-node-timeout 5000

#拷贝集群管理脚本并安装ruby

[root@server10 redis-cluster]# cp /root/redis-4.0.8/src/redis-trib.rb .

[root@server10 redis-cluster]# ls

redis01 redis02 redis03 redis04 redis05 redis06 redis-trib.rb

[root@server10 redis-cluster]# yum -y install ruby rubygems

[root@server10 redis-cluster]# gem install /root/redis-3.2.1.gem

Successfully installed redis-3.2.1

Parsing documentation for redis-3.2.1

Installing ri documentation for redis-3.2.1

1 gem installed

#编写启动脚本

[root@server10 redis-cluster]# vim redis-clu.sh

[root@server10 redis-cluster]# cat redis-clu.sh

#!/bin/bash

case $1 in

start)

for i in {1..6}

do

cd redis0$i

./redis-server redis.conf &> /dev/null

cd ..

done

;;

stop)

pkill -9 redis

;;

status)

netstat -antpu | grep redis &> /dev/null

if [ $? -eq 0 ];then

echo "Running..."

else

echo "Not Running"

fi

;;

\*)

echo "Usage {start|stop|status}"

esac

#启动所有实例

[root@server10 redis-cluster]# ./redis-clu.sh start

[root@server10 redis-cluster]# ./redis-clu.sh status

Running...

[root@server10 redis-cluster]# netstat -antpu | grep redis

tcp 0 0 127.0.0.1:7004 0.0.0.0:\* LISTEN 16203/./redis-serve

tcp 0 0 127.0.0.1:7005 0.0.0.0:\* LISTEN 16208/./redis-serve

tcp 0 0 127.0.0.1:7006 0.0.0.0:\* LISTEN 16213/./redis-serve

tcp 0 0 127.0.0.1:17001 0.0.0.0:\* LISTEN 16188/./redis-serve

tcp 0 0 127.0.0.1:17002 0.0.0.0:\* LISTEN 16193/./redis-serve

tcp 0 0 127.0.0.1:17003 0.0.0.0:\* LISTEN 16198/./redis-serve

tcp 0 0 127.0.0.1:17004 0.0.0.0:\* LISTEN 16203/./redis-serve

tcp 0 0 127.0.0.1:17005 0.0.0.0:\* LISTEN 16208/./redis-serve

tcp 0 0 127.0.0.1:17006 0.0.0.0:\* LISTEN 16213/./redis-serve

tcp 0 0 127.0.0.1:7001 0.0.0.0:\* LISTEN 16188/./redis-serve

tcp 0 0 127.0.0.1:7002 0.0.0.0:\* LISTEN 16193/./redis-serve

tcp 0 0 127.0.0.1:7003 0.0.0.0:\* LISTEN 16198/./redis-serve

#注意，此处启动7001-7006端口表示每个实例启动成功，redis集群通信端口是默认端口+10000，所以此处启动12个端口才是正确结果

#注意，如果所有的节点不是全新安装的redis，一定确保任何一个redis节点的任何一个数据库中都没有数据，否则无法创建集群，报错某个节点is not empty

清空已有节点的数据的时候，既要把内存中的数据清掉，还要把磁盘文件清掉，然后重启redis服务

#创建集群

[root@server10 redis-cluster]# ./redis-trib.rb create --replicas 1 127.0.0.1:7001 127.0.0.1:7002 127.0.0.1:7003 127.0.0.1:7004 127.0.0.1:7005 127.0.0.1:7006

>>> Creating cluster

>>> Performing hash slots allocation on 6 nodes...

Using 3 masters:

127.0.0.1:7001

127.0.0.1:7002

127.0.0.1:7003

Adding replica 127.0.0.1:7005 to 127.0.0.1:7001

Adding replica 127.0.0.1:7006 to 127.0.0.1:7002

Adding replica 127.0.0.1:7004 to 127.0.0.1:7003

>>> Trying to optimize slaves allocation for anti-affinity

[WARNING] Some slaves are in the same host as their master

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:0-5460 (5461 slots) master

M: 3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002

slots:5461-10922 (5462 slots) master

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

slots:10923-16383 (5461 slots) master

S: dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

S: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

replicates 3f1669fa42398841746f35278862d7c2cf991baf

S: 3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

Can I set the above configuration? (type 'yes' to accept): yes

>>> Nodes configuration updated

>>> Assign a different config epoch to each node

>>> Sending CLUSTER MEET messages to join the cluster

Waiting for the cluster to join...

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:0-5460 (5461 slots) master

1 additional replica(s)

S: dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

slots:10923-16383 (5461 slots) master

1 additional replica(s)

M: 3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002

slots:5461-10922 (5462 slots) master

1 additional replica(s)

S: 3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

S: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

slots: (0 slots) slave

replicates 3f1669fa42398841746f35278862d7c2cf991baf

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

#见到这一句，所有的哈希槽被覆盖则集群创建成功，提示信息中带有M字样的表示被分配成主节点，带有S字样的表示被分配成从节点

#查看集群配置文件

[root@server10 redis-cluster]# cat redis01/node7001.conf

dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001@17001 myself,master - 0 1544093423000 1 connected 0-5460

dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004@17004 slave dd2e53829cd35965bfce094fc930016ca8c29d6d 0 1544093423071 4 connected

b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003@17003 master - 0 1544093424000 3 connected 10923-16383

3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002@17002 master - 0 1544093424206 2 connected 5461-10922

3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006@17006 slave b5dd7b2a67e3c27790d17903fc7b319125a79141 0 1544093423000 6 connected

235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005@17005 slave 3f1669fa42398841746f35278862d7c2cf991baf 0 1544093424509 5 connected

vars currentEpoch 6 lastVoteEpoch 0

#命令行查看集群信息

[root@server10 redis-cluster]# ./redis01/redis-cli -h 127.0.0.1 -p 7001

127.0.0.1:7001> CLUSTER NODES

dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001@17001 myself,master - 0 1544098872000 1 connected 0-5460

dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004@17004 slave dd2e53829cd35965bfce094fc930016ca8c29d6d 0 1544098872014 4 connected

b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003@17003 master - 0 1544098871512 3 connected 10923-16383

3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002@17002 master - 0 1544098872519 2 connected 5461-10922

3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006@17006 slave b5dd7b2a67e3c27790d17903fc7b319125a79141 0 1544098873024 6 connected

235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005@17005 slave 3f1669fa42398841746f35278862d7c2cf991baf 0 1544098872000 5 connected

127.0.0.1:7001> CLUSTER INFO

cluster\_state:ok

cluster\_slots\_assigned:16384

cluster\_slots\_ok:16384

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:6

cluster\_size:3

cluster\_current\_epoch:6

cluster\_my\_epoch:1

cluster\_stats\_messages\_ping\_sent:9785

cluster\_stats\_messages\_pong\_sent:9987

cluster\_stats\_messages\_sent:19772

cluster\_stats\_messages\_ping\_received:9982

cluster\_stats\_messages\_pong\_received:9785

cluster\_stats\_messages\_meet\_received:5

cluster\_stats\_messages\_received:19772

#测试集群

#客户端访问任意一台master角色的节点，查询或存储数据

##存入数据，根据crc16计算数据u存储于哪个节点，会有提示信息

[root@server10 redis-cluster]# ./redis01/redis-cli -c -h 127.0.0.1 -p 7001

127.0.0.1:7001> keys \*

(empty list or set)

127.0.0.1:7001> set name zhangsan

-> Redirected to slot [5798] located at 127.0.0.1:7002

OK

127.0.0.1:7002> set age 19

-> Redirected to slot [741] located at 127.0.0.1:7001

OK

##查询数据

[root@server10 ~]# cd /usr/local/redis-cluster/

[root@server10 redis-cluster]# vim redis-trib.rb

[root@server10 redis-cluster]#

[root@server10 redis-cluster]# ./redis01/redis-cli -h 127.0.0.1 -p 7002

127.0.0.1:7002> keys \*

1) "name"

127.0.0.1:7002> get name

"zhangsan"

127.0.0.1:7002> exit

[root@server10 redis-cluster]# ./redis01/redis-cli -h 127.0.0.1 -p 7001

127.0.0.1:7001> keys \*

1) "age"

127.0.0.1:7001> get age

"19"

127.0.0.1:7001> exit

[root@server10 redis-cluster]# ./redis01/redis-cli -h 127.0.0.1 -p 7004

127.0.0.1:7004> keys \*

1) "age"

127.0.0.1:7004> get age

(error) MOVED 741 127.0.0.1:7001

#单独登陆节点，可查询到该节点存储的数据，从节点默认只做备份，不支持写和查

[root@server10 redis-cluster]# ./redis01/redis-cli -c -h 127.0.0.1 -p 7001

127.0.0.1:7001> keys \*

1) "age"

127.0.0.1:7001> get name

-> Redirected to slot [5798] located at 127.0.0.1:7002

"zhangsan"

127.0.0.1:7002> get age

-> Redirected to slot [741] located at 127.0.0.1:7001

"19"

#集群中能查询到所有key，并且提示该key所在的节点

##管理集群

#master选举测试

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:0-5460 (5461 slots) master

1 additional replica(s)

M: 3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002

slots:5461-10922 (5462 slots) master

1 additional replica(s)

S: 3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

S: dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

slots:10923-16383 (5461 slots) master

1 additional replica(s)

S: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

slots: (0 slots) slave

replicates 3f1669fa42398841746f35278862d7c2cf991baf

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

[root@server10 redis-cluster]# ps aux | grep redis

root 11228 0.2 1.1 151452 11936 ? Ssl 12:04 0:13 ./redis-server 127.0.0.1:7001 [cluster]

root 11233 0.2 1.1 151452 11964 ? Ssl 12:04 0:12 ./redis-server 127.0.0.1:7002 [cluster]

root 11238 0.2 1.1 151452 11896 ? Ssl 12:04 0:13 ./redis-server 127.0.0.1:7003 [cluster]

root 11243 0.2 1.1 151452 11948 ? Ssl 12:04 0:12 ./redis-server 127.0.0.1:7004 [cluster]

root 11253 0.2 1.1 151452 11912 ? Ssl 12:04 0:12 ./redis-server 127.0.0.1:7006 [cluster]

root 12068 0.3 1.1 151452 11948 ? Ssl 13:22 0:00 ./redis-server 127.0.0.1:7005 [cluster]

root 12116 0.0 0.0 112724 988 pts/0 S+ 13:23 0:00 grep --color=auto redis

[root@server10 redis-cluster]# kill 11233

[root@server10 redis-cluster]# ps aux | grep redis

root 11228 0.2 1.1 151452 11936 ? Ssl 12:04 0:13 ./redis-server 127.0.0.1:7001 [cluster]

root 11238 0.2 1.1 151452 11896 ? Ssl 12:04 0:13 ./redis-server 127.0.0.1:7003 [cluster]

root 11243 0.2 1.1 151452 11948 ? Ssl 12:04 0:12 ./redis-server 127.0.0.1:7004 [cluster]

root 11253 0.2 1.1 151452 11912 ? Ssl 12:04 0:12 ./redis-server 127.0.0.1:7006 [cluster]

root 12068 0.3 1.1 151452 11948 ? Ssl 13:22 0:00 ./redis-server 127.0.0.1:7005 [cluster]

root 12118 0.0 0.0 112720 988 pts/0 R+ 13:24 0:00 grep --color=auto redis

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:0-5460 (5461 slots) master

1 additional replica(s)

S: 3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

M: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

slots:5461-10922 (5462 slots) master

0 additional replica(s)

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

slots:10923-16383 (5461 slots) master

1 additional replica(s)

S: dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

#当7002宕机以后，7005升级成主

#集群中一对主从都挂掉以后，集群停止工作，因为有节点所有数据丢失

#修复7002节点，启动服务以后7002节点会自动成为7005的从

[root@server10 redis-cluster]# cd redis02/

[root@server10 redis02]# ./redis-server redis.conf

12185:C 09 Dec 13:27:08.985 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

12185:C 09 Dec 13:27:08.985 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=12185, just started

12185:C 09 Dec 13:27:08.986 # Configuration loaded

[root@server10 redis02]# ps aux | grep redis

root 11228 0.2 1.1 151452 11936 ? Ssl 12:04 0:14 ./redis-server 127.0.0.1:7001 [cluster]

root 11238 0.2 1.1 151452 11896 ? Ssl 12:04 0:14 ./redis-server 127.0.0.1:7003 [cluster]

root 11243 0.2 1.1 151452 11948 ? Ssl 12:04 0:13 ./redis-server 127.0.0.1:7004 [cluster]

root 11253 0.2 1.1 151452 11912 ? Ssl 12:04 0:13 ./redis-server 127.0.0.1:7006 [cluster]

root 12068 0.4 1.1 151452 11964 ? Ssl 13:22 0:01 ./redis-server 127.0.0.1:7005 [cluster]

root 12186 0.3 1.1 151452 11952 ? Ssl 13:27 0:00 ./redis-server 127.0.0.1:7002 [cluster]

root 12193 0.0 0.0 112720 992 pts/0 R+ 13:27 0:00 grep --color=auto redis

[root@server10 redis02]# cd ..

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:0-5460 (5461 slots) master

1 additional replica(s)

S: 3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002

slots: (0 slots) slave

replicates 235812f9998bc95bbd5e524593d98758db769237

S: 3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

M: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

slots:5461-10922 (5462 slots) master

1 additional replica(s)

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

slots:10923-16383 (5461 slots) master

1 additional replica(s)

S: dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

#注意，此处如果按照我这种多实例的做法，会有一个坑，重新启动7002节点的时候，需要保证跟其他节点启动服务的方式一样，比如脚本里写的是./redis-server redis.conf，此处写的是./redis02/redis-server redis02/redis.conf，虽然redis实例启动服务没有问题，但是集群不会识别该节点，如果是使用虚拟机，用脚本控制服务，不存在该问题。

##向集群中新增节点

#添加主

[root@server10 redis-cluster]# cp -r redis01 redis07

[root@server10 redis-cluster]# cp -r redis01 redis08

[root@server10 redis-cluster]# rm -rf redis07/appendonly.aof redis07/node7001.conf redis07/dump.rdb

[root@server10 redis-cluster]# rm -rf redis08/appendonly.aof redis08/node7001.conf redis08/dump.rdb

[root@server10 redis-cluster]# vim redis07/redis.conf

[root@server10 redis-cluster]# vim redis08/redis.conf

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis07/redis.conf

port 7007

daemonize yes

pidfile /var/run/redis\_7007.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7007.conf

cluster-node-timeout 5000

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis08/redis.conf

port 7008

daemonize yes

pidfile /var/run/redis\_7008.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7008.conf

cluster-node-timeout 5000

#复制两个实例，删除源文件中的集群配置，修改配置文件

##为了规避上边的坑，启动新实例的方式一定跟其他实例保持一致

[root@server10 redis-cluster]# cd redis07/

[root@server10 redis07]# ./redis-server redis.conf

12603:C 09 Dec 13:45:20.499 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

12603:C 09 Dec 13:45:20.499 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=12603, just started

12603:C 09 Dec 13:45:20.499 # Configuration loaded

[root@server10 redis07]# netstat -antpu | grep 7007

tcp 0 0 127.0.0.1:7007 0.0.0.0:\* LISTEN 12604/./redis-serve

tcp 0 0 127.0.0.1:17007 0.0.0.0:\* LISTEN 12604/./redis-serve

[root@server10 redis07]# ./redis-cli -p 7007

127.0.0.1:7007> keys \*

(empty list or set)

127.0.0.1:7007> exit

[root@server10 redis07]# cd ../redis08/

[root@server10 redis08]# ./redis-server redis.conf

12617:C 09 Dec 13:46:19.227 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

12617:C 09 Dec 13:46:19.227 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=12617, just started

12617:C 09 Dec 13:46:19.227 # Configuration loaded

[root@server10 redis08]# netstat -antpu | grep 7008

tcp 0 0 127.0.0.1:7008 0.0.0.0:\* LISTEN 12618/./redis-serve

tcp 0 0 127.0.0.1:17008 0.0.0.0:\* LISTEN 12618/./redis-serve

[root@server10 redis08]# ./redis-cli -p 7008

127.0.0.1:7008> keys \*

(empty list or set)

127.0.0.1:7008> exit

[root@server10 redis08]# ps aux | grep redis

root 11228 0.2 1.1 151452 11936 ? Ssl 12:04 0:17 ./redis-server 127.0.0.1:7001 [cluster]

root 11238 0.2 1.1 151452 11896 ? Ssl 12:04 0:18 ./redis-server 127.0.0.1:7003 [cluster]

root 11243 0.2 1.1 151452 11948 ? Ssl 12:04 0:17 ./redis-server 127.0.0.1:7004 [cluster]

root 11253 0.2 1.1 151452 11912 ? Ssl 12:04 0:17 ./redis-server 127.0.0.1:7006 [cluster]

root 12068 0.3 1.1 151452 11964 ? Ssl 13:22 0:04 ./redis-server 127.0.0.1:7005 [cluster]

root 12186 0.2 1.1 151452 11956 ? Ssl 13:27 0:03 ./redis-server 127.0.0.1:7002 [cluster]

root 12604 0.2 0.9 147356 9652 ? Ssl 13:45 0:00 ./redis-server 127.0.0.1:7007 [cluster]

root 12618 0.3 0.9 147356 9648 ? Ssl 13:46 0:00 ./redis-server 127.0.0.1:7008 [cluster]

root 12626 0.0 0.0 112720 992 pts/0 R+ 13:46 0:00 grep --color=auto redis

#启动两个新实例的服务，确认新实例上所有库均为空

[root@server10 redis-cluster]# ./redis07/redis-cli -p 7007

127.0.0.1:7007> cluster info

cluster\_state:fail

cluster\_slots\_assigned:0

cluster\_slots\_ok:0

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:1

cluster\_size:0

cluster\_current\_epoch:0

cluster\_my\_epoch:0

cluster\_stats\_messages\_sent:0

cluster\_stats\_messages\_received:0

127.0.0.1:7007> cluster nodes

cbe858f670ebe7c6a2fa869f934ab00fe20d7569 :7007@17007 myself,master - 0 0 0 connected

127.0.0.1:7007> exit

#确认新实例启动集群配置

[root@server10 redis-cluster]# ./redis-trib.rb reshard 127.0.0.1:7007

#重新分配哈希槽，交互式，依次是挪动的哈希槽数量，被添加节点的id，从哪些节点取出哈希槽，是否接受

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:1365-5460 (4096 slots) master

1 additional replica(s)

S: 3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002

slots: (0 slots) slave

replicates 235812f9998bc95bbd5e524593d98758db769237

M: cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007

slots:0-1364,5461-6826,10923-12287 (4096 slots) master

0 additional replica(s)

S: 3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

M: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

slots:6827-10922 (4096 slots) master

1 additional replica(s)

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

slots:12288-16383 (4096 slots) master

1 additional replica(s)

S: dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

#7007节点已经被分配了哈希槽

##添加从节点

[root@server10 redis-cluster]# ./redis08/redis-cli -p 7008

127.0.0.1:7008> cluster info

cluster\_state:fail

cluster\_slots\_assigned:0

cluster\_slots\_ok:0

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:1

cluster\_size:0

cluster\_current\_epoch:0

cluster\_my\_epoch:0

cluster\_stats\_messages\_sent:0

cluster\_stats\_messages\_received:0

127.0.0.1:7008> exit

#确认新节点状态，已经启动集群配置

[root@server10 redis-cluster]# ./redis-trib.rb add-node --slave 127.0.0.1:7008 127.0.0.1:7001

>>> Adding node 127.0.0.1:7008 to cluster 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:1365-5460 (4096 slots) master

1 additional replica(s)

S: 3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002

slots: (0 slots) slave

replicates 235812f9998bc95bbd5e524593d98758db769237

M: cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007

slots:0-1364,5461-6826,10923-12287 (4096 slots) master

0 additional replica(s)

S: 3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

M: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

slots:6827-10922 (4096 slots) master

1 additional replica(s)

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

slots:12288-16383 (4096 slots) master

1 additional replica(s)

S: dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

Automatically selected master 127.0.0.1:7007

>>> Send CLUSTER MEET to node 127.0.0.1:7008 to make it join the cluster.

Waiting for the cluster to join.

>>> Configure node as replica of 127.0.0.1:7007.

[OK] New node added correctly.

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:1365-5460 (4096 slots) master

1 additional replica(s)

S: 3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002

slots: (0 slots) slave

replicates 235812f9998bc95bbd5e524593d98758db769237

M: cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007

slots:0-1364,5461-6826,10923-12287 (4096 slots) master

1 additional replica(s)

S: 3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

S: d8f0e7a8d5727aa7523739a2e15ee4dd2e240029 127.0.0.1:7008

slots: (0 slots) slave

replicates cbe858f670ebe7c6a2fa869f934ab00fe20d7569

M: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

slots:6827-10922 (4096 slots) master

1 additional replica(s)

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

slots:12288-16383 (4096 slots) master

1 additional replica(s)

S: dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

#新节点自动成为从数量最少主节点的从节点

##新增四个实例，分别作为当前四个主节点的从节点

#创建四个新实例

[root@server10 redis-cluster]# cp -r redis01/ redis09

[root@server10 redis-cluster]# rm -rf redis09/appendonly.aof redis09/dump.rdb redis09/node7001.conf

[root@server10 redis-cluster]# cp -r redis09 redis10

[root@server10 redis-cluster]# cp -r redis09 redis11

[root@server10 redis-cluster]# cp -r redis09 redis12

[root@server10 redis-cluster]# vim redis09/redis.conf

[root@server10 redis-cluster]# vim redis10/redis.conf

[root@server10 redis-cluster]# vim redis11/redis.conf

[root@server10 redis-cluster]# vim redis12/redis.conf

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis09/redis.conf

port 7009

daemonize yes

pidfile /var/run/redis\_7009.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7009.conf

cluster-node-timeout 5000

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis10/redis.conf

port 7010

daemonize yes

pidfile /var/run/redis\_7010.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7010.conf

cluster-node-timeout 5000

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis11/redis.conf

port 7011

daemonize yes

pidfile /var/run/redis\_7011.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7011.conf

cluster-node-timeout 5000

[root@server10 redis-cluster]# sed -n '92p;136p;158p;672p;814p;822p;828p' redis12/redis.conf

port 7012

daemonize yes

pidfile /var/run/redis\_7012.pid

appendonly yes

cluster-enabled yes

cluster-config-file node7012.conf

cluster-node-timeout 5000

#启动四个新实例的服务

[root@server10 redis-cluster]# cd redis09/

[root@server10 redis09]# ./redis-server redis.conf

13834:C 09 Dec 15:13:17.764 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

13834:C 09 Dec 15:13:17.764 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=13834, just started

13834:C 09 Dec 15:13:17.764 # Configuration loaded

[root@server10 redis09]# cd ..

[root@server10 redis-cluster]# cd redis10/

[root@server10 redis10]# ./redis-server redis.conf

13845:C 09 Dec 15:13:29.999 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

13845:C 09 Dec 15:13:29.999 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=13845, just started

13845:C 09 Dec 15:13:29.999 # Configuration loaded

[root@server10 redis10]# cd ..

[root@server10 redis-cluster]# cd redis11/

[root@server10 redis11]# ./redis-server redis.conf

13850:C 09 Dec 15:13:41.098 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

13850:C 09 Dec 15:13:41.098 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=13850, just started

13850:C 09 Dec 15:13:41.098 # Configuration loaded

[root@server10 redis11]# cd ..

[root@server10 redis-cluster]# cd redis12/

[root@server10 redis12]# ./redis-server redis.conf

13855:C 09 Dec 15:13:49.137 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

13855:C 09 Dec 15:13:49.138 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=13855, just started

13855:C 09 Dec 15:13:49.138 # Configuration loaded

[root@server10 redis12]# cd ..

[root@server10 redis-cluster]# ps aux | grep redis

root 11228 0.3 1.1 151452 11988 ? Ssl 12:04 0:37 ./redis-server 127.0.0.1:7001 [cluster]

root 11238 0.3 1.1 151452 11896 ? Ssl 12:04 0:37 ./redis-server 127.0.0.1:7003 [cluster]

root 11243 0.2 1.1 151452 11968 ? Ssl 12:04 0:31 ./redis-server 127.0.0.1:7004 [cluster]

root 11253 0.2 1.1 151452 11912 ? Ssl 12:04 0:30 ./redis-server 127.0.0.1:7006 [cluster]

root 12068 0.3 1.1 151452 11992 ? Ssl 13:22 0:24 ./redis-server 127.0.0.1:7005 [cluster]

root 12186 0.2 1.1 151452 11964 ? Ssl 13:27 0:17 ./redis-server 127.0.0.1:7002 [cluster]

root 12604 0.4 1.1 149404 11932 ? Ssl 13:45 0:24 ./redis-server 127.0.0.1:7007 [cluster]

root 12618 0.2 1.1 149404 11924 ? Rsl 13:46 0:11 ./redis-server 127.0.0.1:7008 [cluster]

root 13835 0.2 0.9 147356 9652 ? Ssl 15:13 0:00 ./redis-server 127.0.0.1:7009 [cluster]

root 13846 0.1 0.9 147356 9648 ? Ssl 15:13 0:00 ./redis-server 127.0.0.1:7010 [cluster]

root 13851 0.2 0.9 147356 9648 ? Ssl 15:13 0:00 ./redis-server 127.0.0.1:7011 [cluster]

root 13856 0.4 0.9 147356 9652 ? Ssl 15:13 0:00 ./redis-server 127.0.0.1:7012 [cluster]

root 13861 0.0 0.0 112720 992 pts/0 R+ 15:13 0:00 grep --color=auto redis

#确认四个新实例的状态

[root@server10 redis-cluster]# ./redis01/redis-cli -p 7009

127.0.0.1:7009> keys \*

(empty list or set)

127.0.0.1:7009> cluster info

cluster\_state:fail

cluster\_slots\_assigned:0

cluster\_slots\_ok:0

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:1

cluster\_size:0

cluster\_current\_epoch:0

cluster\_my\_epoch:0

cluster\_stats\_messages\_sent:0

cluster\_stats\_messages\_received:0

127.0.0.1:7009> exit

[root@server10 redis-cluster]# ./redis01/redis-cli -p 7010

127.0.0.1:7010> keys \*

(empty list or set)

127.0.0.1:7010> cluster info

cluster\_state:fail

cluster\_slots\_assigned:0

cluster\_slots\_ok:0

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:1

cluster\_size:0

cluster\_current\_epoch:0

cluster\_my\_epoch:0

cluster\_stats\_messages\_sent:0

cluster\_stats\_messages\_received:0

127.0.0.1:7010> exit

[root@server10 redis-cluster]# ./redis01/redis-cli -p 7011

127.0.0.1:7011> keys \*

(empty list or set)

127.0.0.1:7011> cluster info

cluster\_state:fail

cluster\_slots\_assigned:0

cluster\_slots\_ok:0

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:1

cluster\_size:0

cluster\_current\_epoch:0

cluster\_my\_epoch:0

cluster\_stats\_messages\_sent:0

cluster\_stats\_messages\_received:0

127.0.0.1:7011> exit

[root@server10 redis-cluster]# ./redis01/redis-cli -p 7012

127.0.0.1:7012> keys \*

(empty list or set)

127.0.0.1:7012> cluster info

cluster\_state:fail

cluster\_slots\_assigned:0

cluster\_slots\_ok:0

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:1

cluster\_size:0

cluster\_current\_epoch:0

cluster\_my\_epoch:0

cluster\_stats\_messages\_sent:0

cluster\_stats\_messages\_received:0

127.0.0.1:7012> exit

#将新实例添加成从节点

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep M

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

M: cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007

M: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

[root@server10 redis-cluster]# ./redis-trib.rb add-node --slave --master-id dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7009 127.0.0.1:7001 &> /dev/null

[root@server10 redis-cluster]# ./redis-trib.rb add-node --slave --master-id cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7010 127.0.0.1:7001 &> /dev/null

[root@server10 redis-cluster]# ./redis-trib.rb add-node --slave --master-id 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7011 127.0.0.1:7001 &> /dev/null

[root@server10 redis-cluster]# ./redis-trib.rb add-node --slave --master-id b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7012 127.0.0.1:7001 &> /dev/null

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:1365-5460 (4096 slots) master

2 additional replica(s)

S: 2be821257021c435e2ef1a3478491baaa52322e5 127.0.0.1:7009

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

S: 3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

M: cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007

slots:0-1364,5461-6826,10923-12287 (4096 slots) master

2 additional replica(s)

S: d8f0e7a8d5727aa7523739a2e15ee4dd2e240029 127.0.0.1:7008

slots: (0 slots) slave

replicates cbe858f670ebe7c6a2fa869f934ab00fe20d7569

M: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

slots:6827-10922 (4096 slots) master

2 additional replica(s)

S: 85a6483f814992032f0785a549c3185fb321228e 127.0.0.1:7010

slots: (0 slots) slave

replicates cbe858f670ebe7c6a2fa869f934ab00fe20d7569

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

slots:12288-16383 (4096 slots) master

2 additional replica(s)

S: 3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002

slots: (0 slots) slave

replicates 235812f9998bc95bbd5e524593d98758db769237

S: 170bac6d2525b09b96931c4b9d7e30ad47498aad 127.0.0.1:7012

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

S: 8869272a10bb1eeebdc6eb866edfd26b91ffb369 127.0.0.1:7011

slots: (0 slots) slave

replicates 235812f9998bc95bbd5e524593d98758db769237

S: dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

#新结构为一主两从

##移除节点

#移除从节点

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep -e 7008

S: d8f0e7a8d5727aa7523739a2e15ee4dd2e240029 127.0.0.1:7008

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep -e 7010

S: 85a6483f814992032f0785a549c3185fb321228e 127.0.0.1:7010

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep S | wc -l

8

#此时从节点8个

[root@server10 redis-cluster]# ./redis-trib.rb del-node 127.0.0.1:7001 d8f0e7a8d5727aa7523739a2e15ee4dd2e240029

>>> Removing node d8f0e7a8d5727aa7523739a2e15ee4dd2e240029 from cluster 127.0.0.1:7001

>>> Sending CLUSTER FORGET messages to the cluster...

>>> SHUTDOWN the node.

[root@server10 redis-cluster]# ./redis-trib.rb del-node 127.0.0.1:7001 85a6483f814992032f0785a549c3185fb321228e

>>> Removing node 85a6483f814992032f0785a549c3185fb321228e from cluster 127.0.0.1:7001

>>> Sending CLUSTER FORGET messages to the cluster...

>>> SHUTDOWN the node.

[root@server10 redis-cluster]# ps aux | grep redis

root 11228 0.3 1.1 151452 11988 ? Ssl 12:04 0:51 ./redis-server 127.0.0.1:7001 [cluster]

root 11238 0.3 1.1 151452 11924 ? Rsl 12:04 0:51 ./redis-server 127.0.0.1:7003 [cluster]

root 11243 0.2 1.1 151452 11968 ? Ssl 12:04 0:44 ./redis-server 127.0.0.1:7004 [cluster]

root 11253 0.2 1.1 151452 11936 ? Ssl 12:04 0:44 ./redis-server 127.0.0.1:7006 [cluster]

root 12068 0.3 1.1 151452 11992 ? Ssl 13:22 0:38 ./redis-server 127.0.0.1:7005 [cluster]

root 12186 0.2 1.1 151452 11964 ? Ssl 13:27 0:30 ./redis-server 127.0.0.1:7002 [cluster]

root 12604 0.3 1.1 149404 11948 ? Ssl 13:45 0:37 ./redis-server 127.0.0.1:7007 [cluster]

root 13835 0.2 1.1 149404 11916 ? Ssl 15:13 0:12 ./redis-server 127.0.0.1:7009 [cluster]

root 13851 0.2 1.1 149404 11912 ? Ssl 15:13 0:12 ./redis-server 127.0.0.1:7011 [cluster]

root 13856 0.2 1.1 149404 11916 ? Ssl 15:13 0:12 ./redis-server 127.0.0.1:7012 [cluster]

root 14703 0.0 0.0 112720 992 pts/0 R+ 16:25 0:00 grep --color=auto redis

#此时7008和7010节点已经宕机

#重新启动7008和7010节点

[root@server10 redis-cluster]# ./redis01/redis-cli -p 7008

127.0.0.1:7008> cluster info

cluster\_state:ok

cluster\_slots\_assigned:16384

cluster\_slots\_ok:16384

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:12

cluster\_size:4

cluster\_current\_epoch:10

cluster\_my\_epoch:10

cluster\_stats\_messages\_ping\_sent:612

cluster\_stats\_messages\_sent:612

cluster\_stats\_messages\_pong\_received:161

cluster\_stats\_messages\_received:161

127.0.0.1:7008> cluster nodes

dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001@17001 master - 0 1544344073742 1 connected 1365-5460

3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006@17006 slave b5dd7b2a67e3c27790d17903fc7b319125a79141 0 1544344073034 3 connected

170bac6d2525b09b96931c4b9d7e30ad47498aad 127.0.0.1:7012@17012 slave b5dd7b2a67e3c27790d17903fc7b319125a79141 0 1544344073000 3 connected

3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002@17002 slave 235812f9998bc95bbd5e524593d98758db769237 0 1544344073000 9 connected

d8f0e7a8d5727aa7523739a2e15ee4dd2e240029 127.0.0.1:7008@17008 myself,slave cbe858f670ebe7c6a2fa869f934ab00fe20d7569 0 1544344025288 0 connected

235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005@17005 master - 0 1544344074000 9 connected 6827-10922

85a6483f814992032f0785a549c3185fb321228e 127.0.0.1:7010@17010 slave,fail? cbe858f670ebe7c6a2fa869f934ab00fe20d7569 1544344025291 1544344025289 10 disconnected

cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007@17007 master - 0 1544344074247 10 connected 0-1364 5461-6826 10923-12287

2be821257021c435e2ef1a3478491baaa52322e5 127.0.0.1:7009@17009 slave dd2e53829cd35965bfce094fc930016ca8c29d6d 0 1544344073000 1 connected

b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003@17003 master - 0 1544344073540 3 connected 12288-16383

8869272a10bb1eeebdc6eb866edfd26b91ffb369 127.0.0.1:7011@17011 slave 235812f9998bc95bbd5e524593d98758db769237 0 1544344073034 9 connected

dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004@17004 slave dd2e53829cd35965bfce094fc930016ca8c29d6d 0 1544344074044 1 connected

127.0.0.1:7008> exit

[root@server10 redis-cluster]# cd redis10/

[root@server10 redis10]# ./redis-server redis.conf

14795:C 09 Dec 16:29:21.798 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

14795:C 09 Dec 16:29:21.799 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=14795, just started

14795:C 09 Dec 16:29:21.799 # Configuration loaded

[root@server10 redis10]# cd ..

[root@server10 redis-cluster]# ./redis01/redis-cli -p 7010

127.0.0.1:7010> cluster info

cluster\_state:ok

cluster\_slots\_assigned:16384

cluster\_slots\_ok:16384

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:11

cluster\_size:4

cluster\_current\_epoch:10

cluster\_my\_epoch:10

cluster\_stats\_messages\_ping\_sent:75

cluster\_stats\_messages\_pong\_sent:8

cluster\_stats\_messages\_sent:83

cluster\_stats\_messages\_ping\_received:8

cluster\_stats\_messages\_pong\_received:75

cluster\_stats\_messages\_received:83

127.0.0.1:7010> cluster nodes

dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001@17001 master - 0 1544344182522 1 connected 1365-5460

3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006@17006 slave b5dd7b2a67e3c27790d17903fc7b319125a79141 0 1544344184552 3 connected

8869272a10bb1eeebdc6eb866edfd26b91ffb369 127.0.0.1:7011@17011 slave 235812f9998bc95bbd5e524593d98758db769237 0 1544344184552 9 connected

235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005@17005 master - 0 1544344183000 9 connected 6827-10922

cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007@17007 master - 0 1544344184000 10 connected 0-1364 5461-6826 10923-12287

85a6483f814992032f0785a549c3185fb321228e 127.0.0.1:7010@17010 myself,slave cbe858f670ebe7c6a2fa869f934ab00fe20d7569 0 1544344161805 0 connected

2be821257021c435e2ef1a3478491baaa52322e5 127.0.0.1:7009@17009 slave dd2e53829cd35965bfce094fc930016ca8c29d6d 0 1544344184552 1 connected

b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003@17003 master - 0 1544344184554 3 connected 12288-16383

170bac6d2525b09b96931c4b9d7e30ad47498aad 127.0.0.1:7012@17012 slave b5dd7b2a67e3c27790d17903fc7b319125a79141 0 1544344182928 3 connected

dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004@17004 slave dd2e53829cd35965bfce094fc930016ca8c29d6d 0 1544344183944 1 connected

3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002@17002 slave 235812f9998bc95bbd5e524593d98758db769237 0 1544344183432 9 connected

127.0.0.1:7010> exit

#重启服务后集群节点状态正常

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep S | wc -l

6

#此时从节点6个

##移除7007主节点

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep 7007

M: cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007

[root@server10 redis-cluster]# ./redis-trib.rb del-node 127.0.0.1:7001 cbe858f670ebe7c6a2fa869f934ab00fe20d7569

>>> Removing node cbe858f670ebe7c6a2fa869f934ab00fe20d7569 from cluster 127.0.0.1:7001

[ERR] Node 127.0.0.1:7007 is not empty! Reshard data away and try again.

#直接移除主节点报错，该主节点哈希槽没有释放，需要重新分片

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep M | wc -l

4

#此时有4个主节点

#把7007上的4096个哈希槽移动到7001节点

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep 7007

M: cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007

[root@server10 redis-cluster]# ./redis-trib.rb del-node 127.0.0.1:7001 cbe858f670ebe7c6a2fa869f934ab00fe20d7569

>>> Removing node cbe858f670ebe7c6a2fa869f934ab00fe20d7569 from cluster 127.0.0.1:7001

[ERR] Node 127.0.0.1:7007 is not empty! Reshard data away and try again.

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep -A 2 7007

M: cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007

slots:0-1364,5461-6826,10923-12287 (4096 slots) master

0 additional replica(s)

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep -A 2 7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:1365-5460 (4096 slots) master

2 additional replica(s)

[root@server10 redis-cluster]# ./redis-trib.rb reshard 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:1365-5460 (4096 slots) master

2 additional replica(s)

S: 2be821257021c435e2ef1a3478491baaa52322e5 127.0.0.1:7009

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

S: 3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

M: cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007

slots:0-1364,5461-6826,10923-12287 (4096 slots) master

0 additional replica(s)

M: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

slots:6827-10922 (4096 slots) master

2 additional replica(s)

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

slots:12288-16383 (4096 slots) master

2 additional replica(s)

S: 3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002

slots: (0 slots) slave

replicates 235812f9998bc95bbd5e524593d98758db769237

S: 170bac6d2525b09b96931c4b9d7e30ad47498aad 127.0.0.1:7012

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

S: 8869272a10bb1eeebdc6eb866edfd26b91ffb369 127.0.0.1:7011

slots: (0 slots) slave

replicates 235812f9998bc95bbd5e524593d98758db769237

S: dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

How many slots do you want to move (from 1 to 16384)? 4096

What is the receiving node ID? dd2e53829cd35965bfce094fc930016ca8c29d6d /接受节点

Please enter all the source node IDs.

Type 'all' to use all the nodes as source nodes for the hash slots.

Type 'done' once you entered all the source nodes IDs.

Source node #1:cbe858f670ebe7c6a2fa869f934ab00fe20d7569 /移除节点

Source node #2:done

#7007节点移除哈希槽后测试

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep -A 2 7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:0-6826,10923-12287 (8192 slots) master

2 additional replica(s)

#7001节点有8192个哈希槽

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep -A 2 7007

M: cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007

slots: (0 slots) master

0 additional replica(s)

#7007节点有0个哈希槽

#再移除7007节点

[root@server10 redis-cluster]# ./redis-trib.rb del-node 127.0.0.1:7001 cbe858f670ebe7c6a2fa869f934ab00fe20d7569

>>> Removing node cbe858f670ebe7c6a2fa869f934ab00fe20d7569 from cluster 127.0.0.1:7001

>>> Sending CLUSTER FORGET messages to the cluster...

>>> SHUTDOWN the node.

[root@server10 redis-cluster]# ps aux | grep 7007

root 15101 0.0 0.0 112720 992 pts/0 R+ 16:53 0:00 grep --color=auto 7007

#此时7007 节点宕机

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001 | grep M | wc -l

3

#此时仅有3个主节点

[root@server10 redis-cluster]# cd redis07/

[root@server10 redis07]# ./redis-server redis.conf

15113:C 09 Dec 16:56:03.689 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

15113:C 09 Dec 16:56:03.689 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=15113, just started

15113:C 09 Dec 16:56:03.689 # Configuration loaded

[root@server10 redis07]# ps aux | grep 7007

root 15114 0.2 0.9 151452 9760 ? Ssl 16:56 0:00 ./redis-server 127.0.0.1:7007 [cluster]

root 15119 0.0 0.0 112720 992 pts/0 R+ 16:56 0:00 grep --color=auto 7007

[root@server10 redis07]# ./redis-cli -p 7007

127.0.0.1:7007> cluster info

cluster\_state:ok

cluster\_slots\_assigned:16384

cluster\_slots\_ok:16384

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:10

cluster\_size:3

cluster\_current\_epoch:11

cluster\_my\_epoch:10

cluster\_stats\_messages\_ping\_sent:76

cluster\_stats\_messages\_pong\_sent:18

cluster\_stats\_messages\_sent:94

cluster\_stats\_messages\_ping\_received:18

cluster\_stats\_messages\_pong\_received:76

cluster\_stats\_messages\_received:94

127.0.0.1:7007> cluster nodes

3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002@17002 slave 235812f9998bc95bbd5e524593d98758db769237 0 1544345789000 9 connected

dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004@17004 slave dd2e53829cd35965bfce094fc930016ca8c29d6d 0 1544345787520 11 connected

3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006@17006 slave b5dd7b2a67e3c27790d17903fc7b319125a79141 0 1544345787822 3 connected

235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005@17005 master - 0 1544345789337 9 connected 6827-10922

8869272a10bb1eeebdc6eb866edfd26b91ffb369 127.0.0.1:7011@17011 slave 235812f9998bc95bbd5e524593d98758db769237 0 1544345789039 9 connected

cbe858f670ebe7c6a2fa869f934ab00fe20d7569 127.0.0.1:7007@17007 myself,master - 0 1544345763693 10 connected

b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003@17003 master - 0 1544345788000 3 connected 12288-16383

2be821257021c435e2ef1a3478491baaa52322e5 127.0.0.1:7009@17009 slave dd2e53829cd35965bfce094fc930016ca8c29d6d 0 1544345787117 11 connected

170bac6d2525b09b96931c4b9d7e30ad47498aad 127.0.0.1:7012@17012 slave b5dd7b2a67e3c27790d17903fc7b319125a79141 0 1544345788000 3 connected

dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001@17001 master - 0 1544345788527 11 connected 0-6826 10923-12287

127.0.0.1:7007> exit

#重新启动7007节点，集群配置依旧存在，但是以不再集群中

##如果需要删除被移除节点的集群配置，则需要删除集群配置文件，重启服务

[root@server10 redis07]# ps aux | grep 7007

root 15114 0.2 0.9 151452 9760 ? Ssl 16:56 0:00 ./redis-server 127.0.0.1:7007 [cluster]

root 15123 0.0 0.0 112720 988 pts/0 R+ 16:58 0:00 grep --color=auto 7007

[root@server10 redis07]# ps aux | grep 7007|7008

-bash: 7008: 未找到命令

[root@server10 redis07]# ps aux | grep 7007

root 15114 0.2 0.9 151452 9760 ? Ssl 16:56 0:00 ./redis-server 127.0.0.1:7007 [cluster]

root 15145 0.0 0.0 112720 992 pts/0 R+ 16:59 0:00 grep --color=auto 7007

[root@server10 redis07]# kill -9 151452

-bash: kill: (151452) - 没有那个进程

[root@server10 redis07]# ps aux | grep 7007

root 15114 0.2 0.9 151452 9760 ? Ssl 16:56 0:00 ./redis-server 127.0.0.1:7007 [cluster]

root 15148 0.0 0.0 112720 992 pts/0 R+ 16:59 0:00 grep --color=auto 7007

[root@server10 redis07]# kill -9 15114

[root@server10 redis07]# ps aux | grep 7008

root 14727 0.3 1.3 151452 14020 ? Ssl 16:27 0:06 ./redis-server 127.0.0.1:7008 [cluster]

root 15150 0.0 0.0 112720 988 pts/0 R+ 16:59 0:00 grep --color=auto 7008

[root@server10 redis07]# kill -9 14727

[root@server10 redis07]# ps aux | grep 7010

root 14796 0.3 1.3 151452 14024 ? Ssl 16:29 0:05 ./redis-server 127.0.0.1:7010 [cluster]

root 15169 0.0 0.0 112720 988 pts/0 R+ 16:59 0:00 grep --color=auto 7010

[root@server10 redis07]# kill -9 14796

[root@server10 redis07]# ps aux | grep 7007

root 15171 0.0 0.0 112720 988 pts/0 R+ 17:00 0:00 grep --color=auto 7007

[root@server10 redis07]# ps aux | grep 7008

root 15173 0.0 0.0 112720 988 pts/0 S+ 17:00 0:00 grep --color=auto 7008

[root@server10 redis07]# ps aux | grep 7010

root 15175 0.0 0.0 112720 992 pts/0 S+ 17:00 0:00 grep --color=auto 7010

#停止三个被移除重新启动的节点

[root@server10 redis07]# ls

appendonly.aof node7007.conf redis-check-aof redis-cli redis-sentinel

dump.rdb redis-benchmark redis-check-rdb redis.conf redis-server

[root@server10 redis07]# rm -rf dump.rdb appendonly.aof node7007.conf

[root@server10 redis07]# ./redis-server redis.conf

15237:C 09 Dec 17:02:00.603 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

15237:C 09 Dec 17:02:00.603 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=15237, just started

15237:C 09 Dec 17:02:00.603 # Configuration loaded

[root@server10 redis07]# ./redis-cli -p 7007

127.0.0.1:7007> cluster info

cluster\_state:fail

cluster\_slots\_assigned:0

cluster\_slots\_ok:0

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:1

cluster\_size:0

cluster\_current\_epoch:0

cluster\_my\_epoch:0

cluster\_stats\_messages\_sent:0

cluster\_stats\_messages\_received:0

127.0.0.1:7007> cluster nodes

33153f39b1f4e11f8fac7692be74906475c7c48f :7007@17007 myself,master - 0 0 0 connected

127.0.0.1:7007> exit

#还原7007节点

[root@server10 redis07]# cd ../redis08

[root@server10 redis08]# ls

appendonly.aof node7008.conf redis-check-aof redis-cli redis-sentinel

dump.rdb redis-benchmark redis-check-rdb redis.conf redis-server

[root@server10 redis08]# rm -rf appendonly.aof dump.rdb node7008.conf

[root@server10 redis08]# ./redis-server redis.conf

15258:C 09 Dec 17:03:10.341 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

15258:C 09 Dec 17:03:10.341 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=15258, just started

15258:C 09 Dec 17:03:10.341 # Configuration loaded

[root@server10 redis08]# ./redis-cli -p 7008

127.0.0.1:7008> cluster info

cluster\_state:fail

cluster\_slots\_assigned:0

cluster\_slots\_ok:0

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:1

cluster\_size:0

cluster\_current\_epoch:0

cluster\_my\_epoch:0

cluster\_stats\_messages\_sent:0

cluster\_stats\_messages\_received:0

127.0.0.1:7008> cluster nodes

fb89413cf86b576e16624b64ee529c8fde270a68 :7008@17008 myself,master - 0 0 0 connected

127.0.0.1:7008> exit

#还原7008节点

[root@server10 redis08]# cd ../redis10/

[root@server10 redis10]# ls

appendonly.aof node7010.conf redis-check-aof redis-cli redis-sentinel

dump.rdb redis-benchmark redis-check-rdb redis.conf redis-server

[root@server10 redis10]# rm -rf appendonly.aof dump.rdb node7010.conf

[root@server10 redis10]# ./redis-server redis.conf

15303:C 09 Dec 17:04:10.784 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

15303:C 09 Dec 17:04:10.784 # Redis version=4.0.8, bits=64, commit=00000000, modified=0, pid=15303, just started

15303:C 09 Dec 17:04:10.784 # Configuration loaded

[root@server10 redis10]# ./redis-cli -p 7010

127.0.0.1:7010> cluster info

cluster\_state:fail

cluster\_slots\_assigned:0

cluster\_slots\_ok:0

cluster\_slots\_pfail:0

cluster\_slots\_fail:0

cluster\_known\_nodes:1

cluster\_size:0

cluster\_current\_epoch:0

cluster\_my\_epoch:0

cluster\_stats\_messages\_sent:0

cluster\_stats\_messages\_received:0

127.0.0.1:7010> cluster nodes

104e7df578319d69a45ddf6fb4f836824965ce5e :7010@17010 myself,master - 0 0 0 connected

127.0.0.1:7010> exit

#还原7010节点

[root@server10 redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd2e53829cd35965bfce094fc930016ca8c29d6d 127.0.0.1:7001

slots:0-6826,10923-12287 (8192 slots) master

2 additional replica(s)

S: 2be821257021c435e2ef1a3478491baaa52322e5 127.0.0.1:7009

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

S: 3968831027e5272f998fd2be7fe8530926a3cb68 127.0.0.1:7006

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

M: 235812f9998bc95bbd5e524593d98758db769237 127.0.0.1:7005

slots:6827-10922 (4096 slots) master

2 additional replica(s)

M: b5dd7b2a67e3c27790d17903fc7b319125a79141 127.0.0.1:7003

slots:12288-16383 (4096 slots) master

2 additional replica(s)

S: 3f1669fa42398841746f35278862d7c2cf991baf 127.0.0.1:7002

slots: (0 slots) slave

replicates 235812f9998bc95bbd5e524593d98758db769237

S: 170bac6d2525b09b96931c4b9d7e30ad47498aad 127.0.0.1:7012

slots: (0 slots) slave

replicates b5dd7b2a67e3c27790d17903fc7b319125a79141

S: 8869272a10bb1eeebdc6eb866edfd26b91ffb369 127.0.0.1:7011

slots: (0 slots) slave

replicates 235812f9998bc95bbd5e524593d98758db769237

S: dfd97ac48853853bb6640de727df0e988ac5caca 127.0.0.1:7004

slots: (0 slots) slave

replicates dd2e53829cd35965bfce094fc930016ca8c29d6d

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

#此时，集群中有三个主节点，六个从节点

##如果需要把从集群中移出来的节点重新添加回集群，启动节点以后连接到redis命令行

用cluster info查看集群状态，如果显示cluster\_state:fail则可以直接添加，如果显示状态是ok，则需要先执行cluster reset命令，重置节点集群信息，当状态显示为fail以后再执行添加动作。