# 一、创建集群

## 1.1 规划

10.1.5.205:7001 master --> 10.1.5.205:7004 slave

10.1.5.205:7002 master --> 10.1.5.205:7005 slave

10.1.5.205:7003 master --> 10.1.5.205:7006 slave

## 1.2 创建集群

这里不介绍集群的部署，相关文档有很多，只介绍运维人员可能需要了解的相关集群操作。

create host1:port1 ... hostN:portN

--replicas <arg>

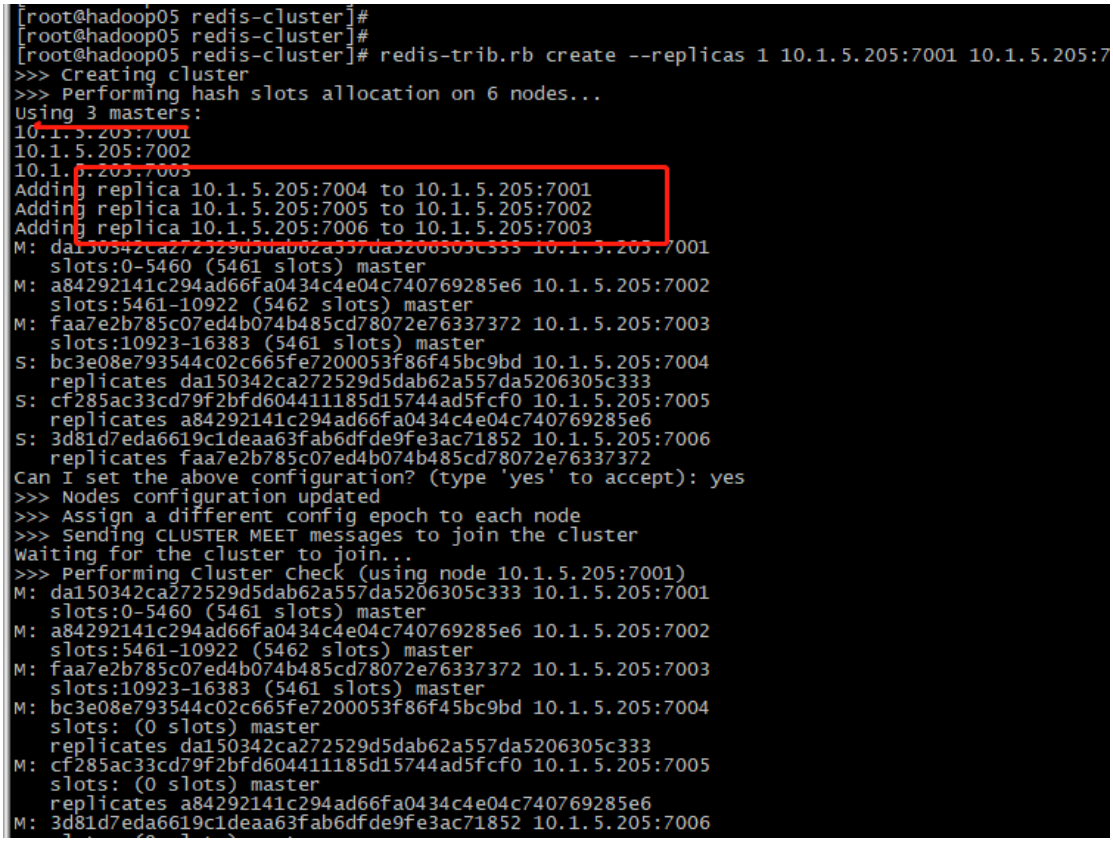
举例：

--replicas 1 每个 master 创建副本的数量是 1

redis-trib.rb create --replicas 1 10.1.5.205:7001 10.1.5.205:7002 10.1.5.205:7003

10.1.5.205:7004 10.1.5.205:7005 10.1.5.205:7006

这条命令，会默认将前三个设置为 master，后三个设置为 slave，且一一对应，如果没问题，输入 yes 即可创建完成。


[root@hadoop05 redis-cluster]#
[root@hadoop05 redis-cluster]#
[root@hadoop05 redis-cluster]# redis-trib.rb create --replicas 1 10.1.5.205:7001 10.1.5.205:7
>>> Creating cluster
>>> Performing hash slots allocation on 6 nodes...
Using 3 masters:
10.1.5.205:7001
10.1.5.205:7002
10.1.5.205:7003
Adding replica 10.1.5.205:7004 to 10.1.5.205:7001
Adding replica 10.1.5.205:7005 to 10.1.5.205:7002
Adding replica 10.1.5.205:7006 to 10.1.5.205:7003
M: da150342ca272529d5dab62a557da5206305c333 10.1.5.205:7001
slots:0-5460 (5461 slots) master
M: a84292141c294ad66fa0434c4e04c740769285e6 10.1.5.205:7002
slots:5461-10922 (5462 slots) master
M: faa7e2b785c07ed4b074b485cd78072e76337372 10.1.5.205:7003
slots:10923-16383 (5461 slots) master
S: bc3e08e793544c02c665fe7200053f86f45bc9bd 10.1.5.205:7004
replicates da150342ca272529d5dab62a557da5206305c333
S: cf285ac33cd79f2bfd604411185d15744ad5fcf0 10.1.5.205:7005
replicates a84292141c294ad66fa0434c4e04c740769285e6
S: 3d81d7eda6619c1deaa63fab6dfde9fe3ac71852 10.1.5.205:7006
replicates faa7e2b785c07ed4b074b485cd78072e76337372
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 10.1.5.205:7001)
M: da150342ca272529d5dab62a557da5206305c333 10.1.5.205:7001
slots:0-5460 (5461 slots) master
M: a84292141c294ad66fa0434c4e04c740769285e6 10.1.5.205:7002
slots:5461-10922 (5462 slots) master
M: faa7e2b785c07ed4b074b485cd78072e76337372 10.1.5.205:7003
slots:10923-16383 (5461 slots) master
M: bc3e08e793544c02c665fe7200053f86f45bc9bd 10.1.5.205:7004
slots: (0 slots) master
replicates da150342ca272529d5dab62a557da5206305c333
M: cf285ac33cd79f2bfd604411185d15744ad5fcf0 10.1.5.205:7005
slots: (0 slots) master
replicates a84292141c294ad66fa0434c4e04c740769285e6
M: 3d81d7eda6619c1deaa63fab6dfde9fe3ac71852 10.1.5.205:7006

## 二、集群信息查看

### 2.1 使用 check 检查集群

当出现下面一行时，证明集群工作正常，可使用 zabbix 等监控此信息来判断集群状态：

[OK] All 16384 slots covered.

同时如果出现：

0 additional replica(s)

证明有从节点挂了或则本身没有配置从节点，此时虽然不影响正常工作，但应该引起管理员注意。

```
[root@hadoop05 redis-cluster]# redis-trib.rb check 10.1.5.205:7001
>>> Performing Cluster Check (using node 10.1.5.205:7001)
M: da150342ca272529d5dab62a557da5206305c333 10.1.5.205:7001
   slots:1366-5461 (4096 slots) master
   1 additional replica(s)
M: 9987adaef51933673afe6f5d7c0ea292c94f2182 10.1.5.205:7007
   slots:0-1365,5462-6826,10923-12287 (4096 slots) master
   0 additional replica(s)
M: faa7e2b785c07ed4b074b485cd78072e76337372 10.1.5.205:7003
   slots:12288-16383 (4096 slots) master
   1 additional replica(s)
S: bc3e08e793544c02c665fe7200053f86f45bc9bd 10.1.5.205:7004
   slots: (0 slots) slave
   replicates da150342ca272529d5dab62a557da5206305c333
S: cf285ac33cd79f2bfd604411185d15744ad5fcf0 10.1.5.205:7005
   slots: (0 slots) slave
   replicates a84292141c294ad66fa0434c4e04c740769285e6
M: a84292141c294ad66fa0434c4e04c740769285e6 10.1.5.205:7002
   slots:6827-10922 (4096 slots) master
   1 additional replica(s)
S: 3d81d7eda6619c1deaa63fab6dfde9fe3ac71852 10.1.5.205:7006
   slots: (0 slots) slave
   replicates faa7e2b785c07ed4b074b485cd78072e76337372
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
```

当其中一个 master 挂掉，且没有从顶替时，使用 check 查看：

[ERR] Not all 16384 slots are covered by nodes.

```
[root@hadoop05 redis-cluster]# redis-trib.rb check 10.1.5.205:7001
>>> Performing Cluster Check (using node 10.1.5.205:7001)
M: da150342ca272529d5dab62a557da5206305c333 10.1.5.205:7001
   slots:1366-5461 (4096 slots) master
   1 additional replica(s)
M: faa7e2b785c07ed4b074b485cd78072e76337372 10.1.5.205:7003
   slots:12288-16383 (4096 slots) master
   1 additional replica(s)
S: bc3e08e793544c02c665fe7200053f86f45bc9bd 10.1.5.205:7004
   slots: (0 slots) slave
   replicates da150342ca272529d5dab62a557da5206305c333
S: cf285ac33cd79f2bfd604411185d15744ad5fcf0 10.1.5.205:7005
   slots: (0 slots) slave
   replicates a84292141c294ad66fa0434c4e04c740769285e6
M: a84292141c294ad66fa0434c4e04c740769285e6 10.1.5.205:7002
   slots:6827-10922 (4096 slots) master
   1 additional replica(s)
S: 3d81d7eda6619c1deaa63fab6dfde9fe3ac71852 10.1.5.205:7006
   slots: (0 slots) slave
   replicates faa7e2b785c07ed4b074b485cd78072e76337372
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[ERR] Not all 16384 slots are covered by nodes.
```

## 2.2 使用 info 查看集群信息

```
[root@hadoop05 redis-cluster]#
[root@hadoop05 redis-cluster]# redis-trib.rb info 10.1.5.205:7001
10.1.5.205:7001 (da150342...) -> 0 keys | 4096 slots | 1 slaves.
10.1.5.205:7007 (9987adae...) -> 0 keys | 4096 slots | 0 slaves.
10.1.5.205:7003 (faa7e2b7...) -> 0 keys | 4096 slots | 1 slaves.
10.1.5.205:7002 (a8429214...) -> 0 keys | 4096 slots | 1 slaves.
[OK] 0 keys in 4 masters.
0.00 keys per slot on average.
[root@hadoop05 redis-cluster]#
```

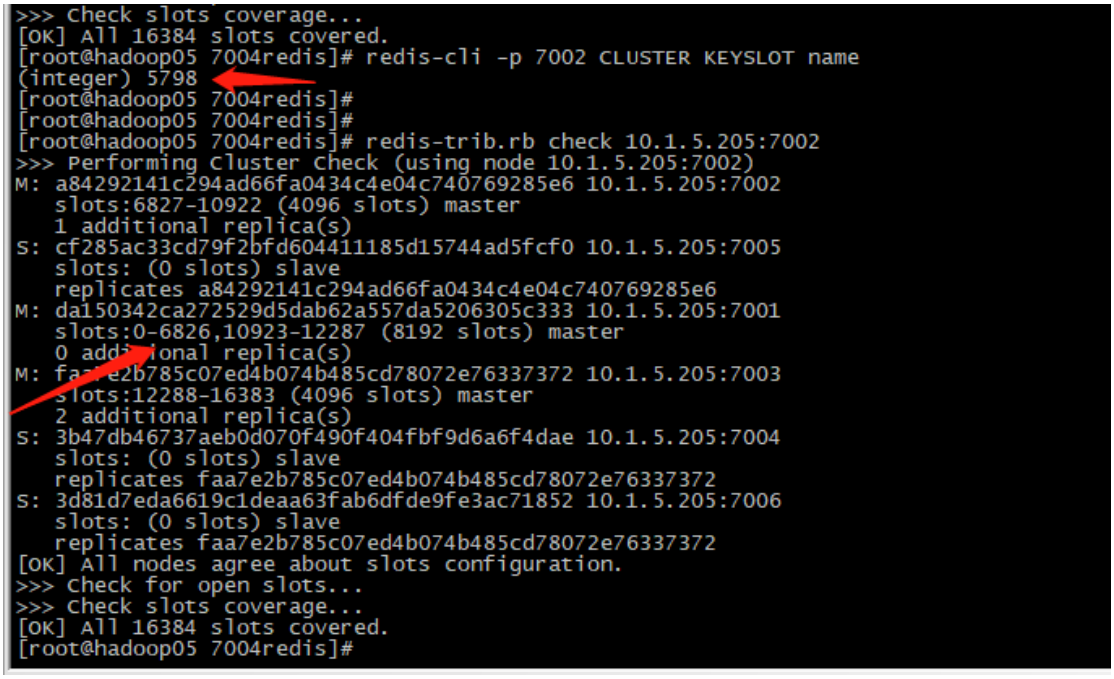
当其中一个 master 挂掉，且没有从顶替时，使用 info 查看：

[ERR] Sorry, can't connect to node 10.1.5.205:7007

```
[root@hadoop05 redis-cluster]# redis-trib.rb info 10.1.5.205:7001
[ERR] Sorry, can't connect to node 10.1.5.205:7007
10.1.5.205:7001 (da150342...) -> 0 keys | 4096 slots | 1 slaves.
10.1.5.205:7003 (faa7e2b7...) -> 0 keys | 4096 slots | 1 slaves.
10.1.5.205:7002 (a8429214...) -> 0 keys | 4096 slots | 1 slaves.
[OK] 0 keys in 3 masters.
0.00 keys per slot on average.
[root@hadoop05 redis-cluster]#
```

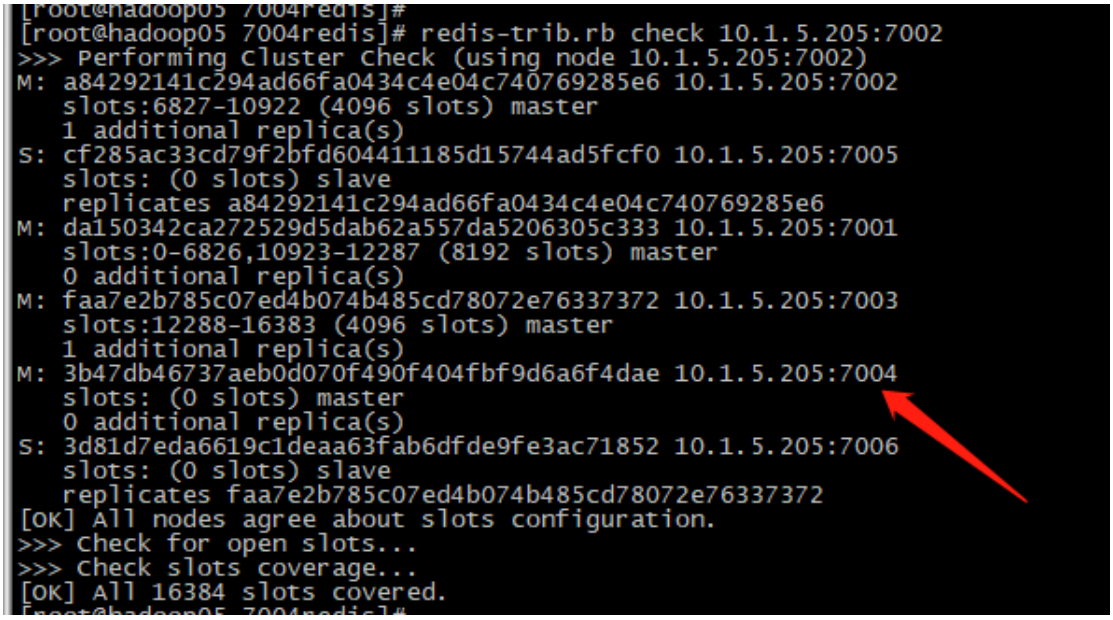
## 2.3 查看 key 的 slot

先使用 CLUSTER KEYSLOT key 计算出 slot，再使用 info 查看对应的 master


>>> Check slots coverage...
[OK] All 16384 slots covered.
[root@hadoop05 7004redis]# redis-cli -p 7002 CLUSTER KEYSLOT name
(integer) 5798
[root@hadoop05 7004redis]#
[root@hadoop05 7004redis]#
[root@hadoop05 7004redis]# redis-trib.rb check 10.1.5.205:7002
>>> Performing Cluster Check (using node 10.1.5.205:7002)
M: a84292141c294ad66fa0434c4e04c740769285e6 10.1.5.205:7002
slots:6827-10922 (4096 slots) master
1 additional replica(s)
S: cf285ac33cd79f2bfd604411185d15744ad5fcf0 10.1.5.205:7005
slots: (0 slots) slave
replicates a84292141c294ad66fa0434c4e04c740769285e6
M: da150342ca272529d5dab62a557da5206305c333 10.1.5.205:7001
slots:0-6826,10923-12287 (8192 slots) master
0 additional replica(s)
M: faa7e2b785c07ed4b074b485cd78072e76337372 10.1.5.205:7003
slots:12288-16383 (4096 slots) master
2 additional replica(s)
S: 3b47db46737aeb0d070f490f404fbf9d6a6f4dae 10.1.5.205:7004
slots: (0 slots) slave
replicates faa7e2b785c07ed4b074b485cd78072e76337372
S: 3d81d7eda6619c1deaa63fab6dfde9fe3ac71852 10.1.5.205:7006
slots: (0 slots) slave
replicates faa7e2b785c07ed4b074b485cd78072e76337372
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
[root@hadoop05 7004redis]#

## 三、从节点切换

新节点加入时，如果未指定，默认是主节点，且无 slot 分配。

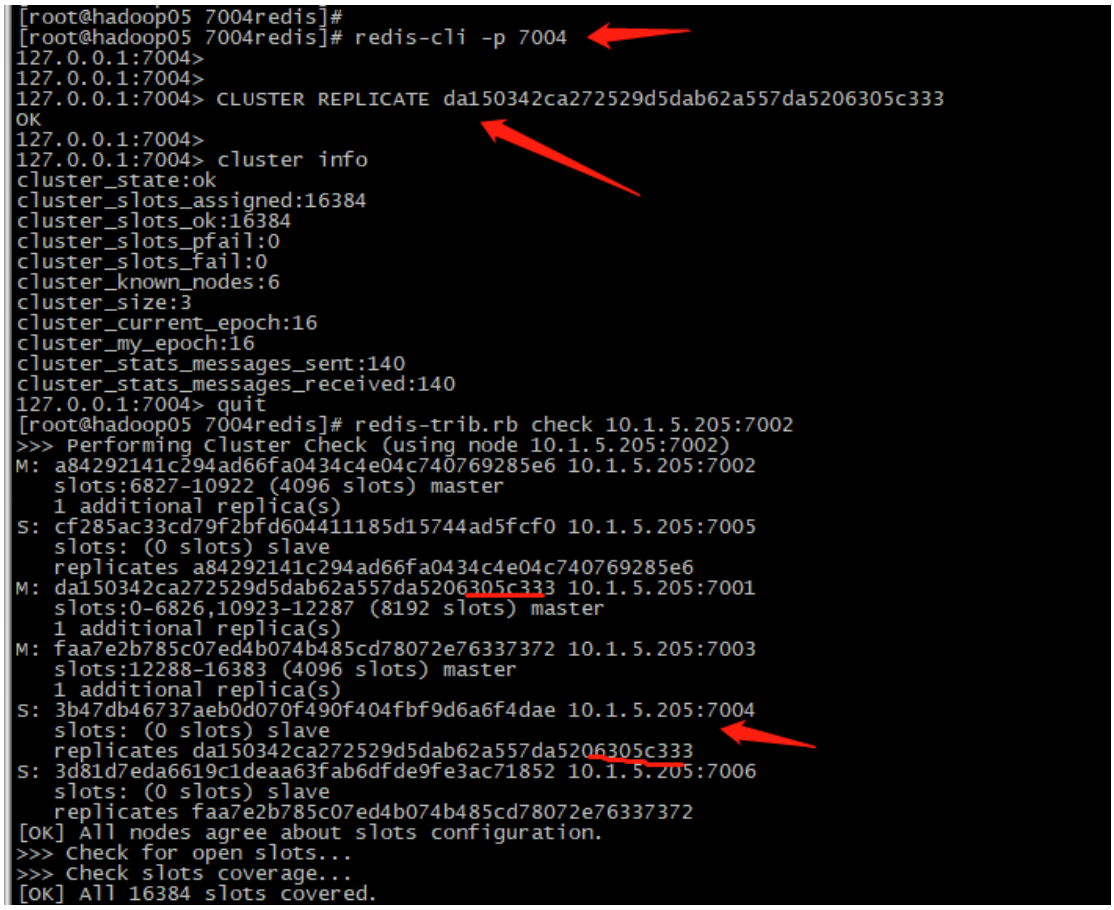



此时可以通过 CLUSTER REPLICATE 命令进去指定为某个节点的从。

直连 7004，

CLUSTER REPLICATE da150342ca272529d5dab62a557da5206305c333

其中 da150342ca272529d5dab62a557da5206305c333 为 7001 的 id




同样，即使 7004 已经有 master，也可以重新指定其他 master 代替当前 master，比如下图，重新指定后,7003 有两个 slave

7001 master -->无 slave

7002 master -->7005 slave

7003 master --> 7004/7006slave

注意：这种操作对初始创建集群时分配不合理，而又不能停止服务，需要重新调整时特别有用。

```
[root@hadoop05 7004redis]#
[root@hadoop05 7004redis]#
[root@hadoop05 7004redis]# redis-cli -p 7004
127.0.0.1:7004>
127.0.0.1:7004>
127.0.0.1:7004>
127.0.0.1:7004> CLUSTER REPLICATE faa7e2b785c07ed4b074b485cd78072e76337372
OK
127.0.0.1:7004> quit
[root@hadoop05 7004redis]# redis-trib.rb check 10.1.5.205:7002
>>> Performing Cluster Check (using node 10.1.5.205:7002)
M: a84292141c294ad66fa0434c4e04c740769285e6 10.1.5.205:7002
   slots:6827-10922 (4096 slots) master
   1 additional replica(s)
S: cf285ac33cd79f2bfd604411185d15744ad5fcf0 10.1.5.205:7005
   slots: (0 slots) slave
   replicates a84292141c294ad66fa0434c4e04c740769285e6
M: da150342ca272529d5dab62a557da5206305c333 10.1.5.205:7001
   slots:0-6826,10923-12287 (8192 slots) master
   0 additional replica(s)
M: faa7e2b785c07ed4b074b485cd78072e76337372 10.1.5.205:7003
   slots:12288-16383 (4096 slots) master
   2 additional replica(s)
S: 3b47db46737aeb0d070f490f404fbf9d6a6f4dae 10.1.5.205:7004
   slots: (0 slots) slave
   replicates faa7e2b785c07ed4b074b485cd78072e76337372
S: 3d81d7eda6619c1deaa63fab6dfde9fe3ac71852 10.1.5.205:7006
   slots: (0 slots) slave
   replicates faa7e2b785c07ed4b074b485cd78072e76337372
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
```

使用 info 也可以查看到主从相关信息。

```
[root@hadoop05 7004redis]# redis-trib.rb info 10.1.5.205:7002
10.1.5.205:7002 (a8429214...) -> 0 keys | 4096 slots | 1 slaves.
10.1.5.205:7001 (da150342...) -> 0 keys | 8192 slots | 0 slaves.
10.1.5.205:7003 (faa7e2b7...) -> 0 keys | 4096 slots | 2 slaves.
[OK] 0 keys in 3 masters.
0.00 keys per slot on average.
```

## 四 观察主从切换

停止前：

```
127.0.0.1:7001> cluster nodes
a3bc24765d7c8e1d5de86db2978c091fd81758c9 10.1.5.205:7006 slave 10518d23d2a78052c1c164ee68c197797651bf28 0 1531131093146 6 connected
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 slave cef32f21f38fe366352ec43f3e271a221d34d712 0 1531131094149 4 connected
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531131095151 3 connected 10923-16383
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 myself,master - 0 0 1 connected 0-5460
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 1531131092144 5 connected
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 master - 0 1531131094148 2 connected 5461-10922
```

kill 掉 7001 进程：

```
127.0.0.1:7002> cluster nodes
a3bc24765d7c8e1d5de86db2978c091fd81758c9 10.1.5.205:7006 slave 10518d23d2a78052c1c164ee68c197797651bf28 0 1531131282803 6 connected
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 myself,master - 0 0 2 connected 5461-10922
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531131286839 3 connected 10923-16383
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 master - 0 1531131285816 7 connected 0-5460
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 1531131284813 5 connected
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 master,fail - 1531131261836 1531131258610 1 disconnected
127.0.0.1:7002>
```

发现 7001 处于 fail 状态，且 7001 的从 7004 变成了主。使用 redis-trib.rb 查看也提示

7004 有 0 个额外的副本，且另个两个 master 都有 1 个额外的副本。

```
[root@hadoop05 7003redis]# redis-trib.rb check 127.0.0.1:7006
>>> Performing Cluster Check (using node 127.0.0.1:7006)
M: a3bc24765d7c8e1d5de86db2978c091fd81758c9 127.0.0.1:7006
   slots:10923-16383 (5461 slots) master
   1 additional replica(s)
S: 72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005
   slots: (0 slots) slave
   replicates 5289506a768c2253c10c91311b7135e5bee86bbe
S: 10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003
   slots: (0 slots) slave
   replicates a3bc24765d7c8e1d5de86db2978c091fd81758c9
M: a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004
   slots:0-5460 (5461 slots) master
   0 additional replica(s)
M: 5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002
   slots:5461-10922 (5462 slots) master
   1 additional replica(s)
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
```

重启 7001

```
127.0.0.1:7002> cluster nodes
a3bc24765d7c8e1d5de86db2978c091fd81758c9 10.1.5.205:7006 slave 10518d23d2a78052c1c164ee68c197797651bf28 0 1531131398406 6 connected
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 myself,master - 0 0 2 connected 5461-10922
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531131401413 3 connected 10923-16383
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 master - 0 1531131402415 7 connected 0-5460
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 1531131399409 5 connected
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 slave a627628c4bd24c1e34454ff67cafe86858ccc84f 0 1531131400411 7 connected
127.0.0.1:7002>
```

发现 7004 仍然是主，但 7001 变成了 7004 的从

备注：

a:如果集群任意 master 挂掉,且当前 master 没有 slave.集群进入 fail 状态,也可以理解成

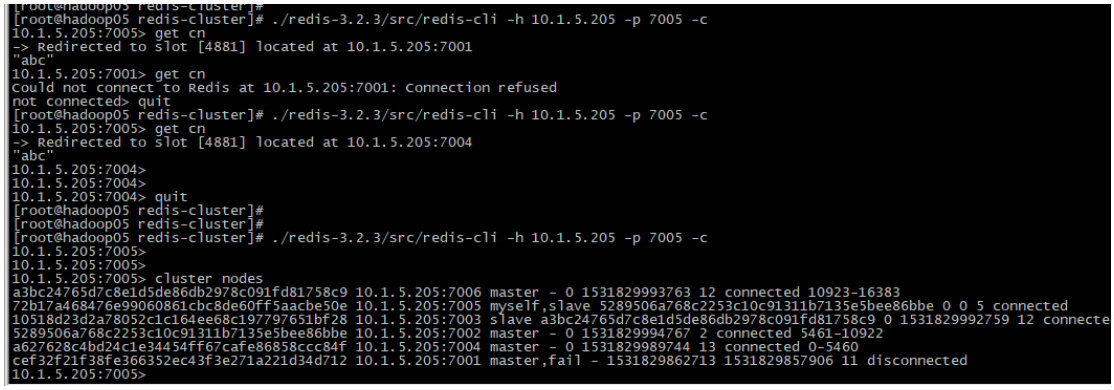
进群的 slot 映射[0-16383]不完全时，集群进入 fail 状态，当有 slave 进入时，自动转换为 master。

b:如果进群超过半数以上 master 挂掉，无论是否有 slave 集群都将进入 fail 状态。所以，如果三个 master 节点中，两个 master 挂掉，集群不可恢复，无论有多少个 slave。

c.当对应的 master、slave 同时挂时，集群不可通过新增节点恢复，除非修改所有点的 nodes.conf 信息，这种现象概率很低，当然也需要管理员监控好集群状态，防止这种情况出现。

## 五 观察主从切换时，数据是否丢失

对于 key:cn，value:abc。开始存储在 7001 上，当 7001 停止后，发现 7004 成了 master，且该 key 可以在 7004 上查到。

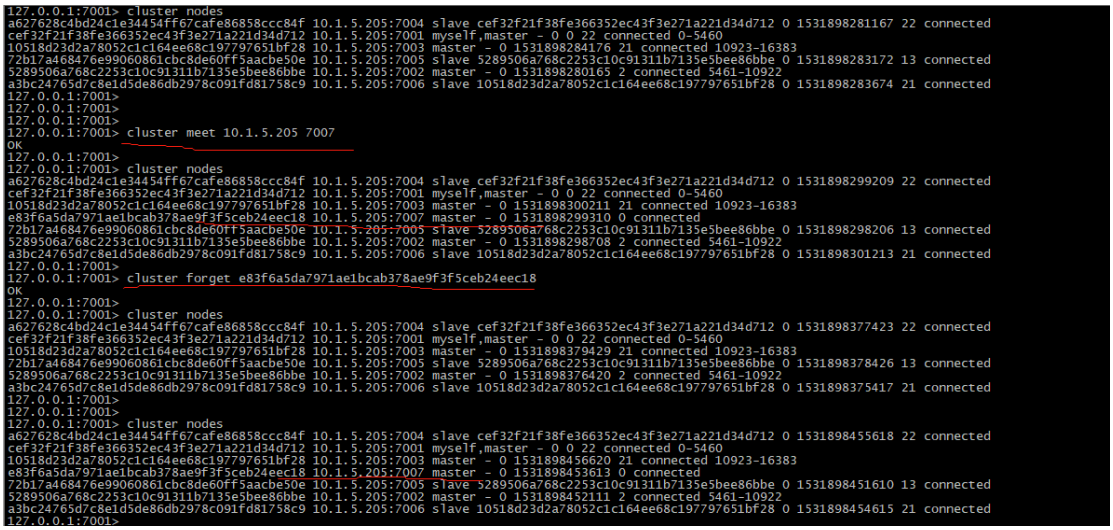


# 六、 集群中添加和删除节点

## 6.1 cluster 方法

使用 redis-cli，连接，-c 与否均可。

cluster meet <ip> <port> ：将 ip 和 port 所指定的节点添加到集群当中，让它成为集群的一份子，nodes.conf 文件也会变化。
cluster forget <node_id> ：从集群中移除 node_id 指定的节点，nodes.conf 文件中不会删除。




注意：这种移除，其实并没有真正移除，因为各节点的 nodes.conf 中并没有删除该节点信息，只要该节点在线，稍等一会就会自动加入。

## 6.2 redis-trib.rb 方法

使用 redis-trib.rb 脚本
语法：
add-node　　new_host:new_port existing_host:existing_port
　　--slave，加入时默认是 master 节点。一个主可以有多个从。
　　--master-id <arg>
例如：redis-trib.rb add-node 10.1.5.205:7007 10.1.5.205:7002，
默认是 master 节点，且无 slot 分配，该操作也会直接修改 nodes.conf 文件。

```
[root@hadoop05 7001redis]# cat nodes.conf
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 slave cef32f21f38fe366352ec43f3e271a221d34d712 0 1531898685239 22 c
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 myself,master - 0 0 22 connected 0-5460
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531898689250 21 connected 10923-16383
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 1531898684237 13 c
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 master - 0 1531898687245 2 connected 5461-10922
a3bc24765d7c8e1d5de86db2978c091fd81758c9 10.1.5.205:7006 slave 10518d23d2a78052c1c164ee68c197797651bf28 0 1531898688248 21 c
vars currentEpoch 22 lastVoteEpoch 12
[root@hadoop05 7001redis]# redis-trib.rb add-node 10.1.5.205:7007 10.1.5.205:7002
>>> Adding node 10.1.5.205:7007 to cluster 10.1.5.205:7002
>>> Performing Cluster Check (using node 10.1.5.205:7002)
M: 5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002
   slots:5461-10922 (5462 slots) master
   1 additional replica(s)
S: a3bc24765d7c8e1d5de86db2978c091fd81758c9 10.1.5.205:7006
   slots: (0 slots) slave
   replicates 10518d23d2a78052c1c164ee68c197797651bf28
M: 10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003
   slots:10923-16383 (5461 slots) master
   1 additional replica(s)
S: a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004
   slots: (0 slots) slave
   replicates cef32f21f38fe366352ec43f3e271a221d34d712
S: 72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005
   slots: (0 slots) slave
   replicates 5289506a768c2253c10c91311b7135e5bee86bbe
M: cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001
   slots:0-5460 (5461 slots) master
   1 additional replica(s)
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
>>> Send CLUSTER MEET to node 10.1.5.205:7007 to make it join the cluster.
[OK] New node added correctly.
[root@hadoop05 7001redis]# cat nodes.conf
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 slave cef32f21f38fe366352ec43f3e271a221d34d712 0 1531898899030 22 c
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 myself,master - 0 0 22 connected 0-5460
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531898902038 21 connected 10923-16383
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 1531898903041 13 c
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 master - 0 1531898900034 2 connected 5461-10922
a3bc24765d7c8e1d5de86db2978c091fd81758c9 10.1.5.205:7006 slave 10518d23d2a78052c1c164ee68c197797651bf28 0 1531898901036 21 c
fc4c63986c1f2355a867382a8912bea49fdb5e68 10.1.5.205:7007 master - 0 1531898903144 0 connected
vars currentEpoch 22 lastVoteEpoch 12
[root@hadoop05 7001redis]#
```

添加节点时，使用--slave 选项，优先选择 slave 不足的主作为主

```
[root@hadoop05 redis-cluster]# cat 7001redis/nodes.conf
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 slave cef32f21f38fe366352ec43f3e271a221d34d712 0 1531901343742 22 co
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 myself,master - 0 0 22 connected 0-5460
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531901341737 21 connected 10923-16383
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 1531901346749 13 co
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 master - 0 1531901345747 2 connected 5461-10922
vars currentEpoch 22 lastVoteEpoch 12
[root@hadoop05 redis-cluster]#
[root@hadoop05 redis-cluster]# redis-trib.rb add-node 10.1.5.205:7007 10.1.5.205:7002 --slave
[ERR] Wrong number of arguments for specified sub command
[root@hadoop05 redis-cluster]# redis-trib.rb add-node --slave 10.1.5.205:7007 10.1.5.205:7002
>>> Adding node 10.1.5.205:7007 to cluster 10.1.5.205:7002
>>> Performing Cluster Check (using node 10.1.5.205:7002)
M: 5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002
   slots:5461-10922 (5462 slots) master
   1 additional replica(s)
M: 10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003
   slots:10923-16383 (5461 slots) master
   0 additional replica(s)
S: a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004
   slots: (0 slots) slave
   replicates cef32f21f38fe366352ec43f3e271a221d34d712
S: 72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005
   slots: (0 slots) slave
   replicates 5289506a768c2253c10c91311b7135e5bee86bbe
M: cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001
   slots:0-5460 (5461 slots) master
   1 additional replica(s)
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
Automatically selected master 10.1.5.205:7003
>>> Send CLUSTER MEET to node 10.1.5.205:7007 to make it join the cluster.
Waiting for the cluster to join.
>>> Configure node as replica of 10.1.5.205:7003.
[OK] New node added correctly.
[root@hadoop05 redis-cluster]# cat 7001redis/nodes.conf
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 slave cef32f21f38fe366352ec43f3e271a221d34d712 0 1531901455075 22 co
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 myself,master - 0 0 22 connected 0-5460
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531901457078 21 connected 10923-16383
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 1531901456078 13 co
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 master - 0 1531901458081 2 connected 5461-10922
908c580f36779339c299115ba3a9cbecbfb3c0ba 10.1.5.205:7007 slave 10518d23d2a78052c1c164ee68c197797651bf28 0 1531901456881 21 co
vars currentEpoch 22 lastVoteEpoch 12
```

一个主可以有多个从，当从已经够时，自主协商哪一个作为主：

```
[root@hadoop05 redis-cluster]# cat 7001redis/nodes.conf
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 slave cef32f21f38fe366352ec43f3e271a221d34d7
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 myself,master - 0 0 22 connected 0-5460
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531901457078 21 connected 10923-
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86b
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 master - 0 1531901458081 2 connected 5461-10
908c580f36779339c299115ba3a9cbecbfb3c0ba 10.1.5.205:7007 slave 10518d23d2a78052c1c164ee68c197797651bf
vars currentEpoch 22 lastVoteEpoch 12
[root@hadoop05 redis-cluster]#
[root@hadoop05 redis-cluster]#
[root@hadoop05 redis-cluster]# redis-trib.rb add-node 10.1.5.205:7006 10.1.5.205:7002 --slave
[ERR] Wrong number of arguments for specified sub command
[root@hadoop05 redis-cluster]# redis-trib.rb add-node --slave 10.1.5.205:7006 10.1.5.205:7002
>>> Adding node 10.1.5.205:7006 to cluster 10.1.5.205:7002
>>> Performing Cluster Check (using node 10.1.5.205:7002)
M: 5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002
   slots:5461-10922 (5462 slots) master
   1 additional replica(s)
M: 10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003
   slots:10923-16383 (5461 slots) master
   1 additional replica(s)
S: a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004
   slots: (0 slots) slave
   replicates cef32f21f38fe366352ec43f3e271a221d34d712
S: 72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005
   slots: (0 slots) slave
   replicates 5289506a768c2253c10c91311b7135e5bee86bbe
S: 908c580f36779339c299115ba3a9cbecbfb3c0ba 10.1.5.205:7007
   slots: (0 slots) slave
   replicates 10518d23d2a78052c1c164ee68c197797651bf28
M: cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001
   slots:0-5460 (5461 slots) master
   1 additional replica(s)
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
Automatically selected master 10.1.5.205:7002
>>> Send CLUSTER MEET to node 10.1.5.205:7006 to make it join the cluster.
Waiting for the cluster to join.
>>> Configure node as replica of 10.1.5.205:7002.
[OK] New node added correctly.
[root@hadoop05 redis-cluster]#
```

```
127.0.0.1:7001>
127.0.0.1:7001> cluster nodes
e89d78469c680039d1052c774ea52f938f548cbc 10.1.5.205:7006 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 1531901619506 2 connected
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 slave cef32f21f38fe366352ec43f3e271a221d34d712 0 1531901619004 22 connected
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 myself,master - 0 0 22 connected 0-5460
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531901617498 21 connected 10923-16383
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 1531901615493 13 connected
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 master - 0 1531901618503 2 connected 5461-10922
908c580f36779339c299115ba3a9cbecbfb3c0ba 10.1.5.205:7007 slave 10518d23d2a78052c1c164ee68c197797651bf28 0 1531901616497 21 connected
127.0.0.1:7001>
```

使用--master-id 选项，加入的为从节点时，指定主节点

```
[root@hadoop05 7006redis]#
[root@hadoop05 7006redis]# redis-trib.rb add-node --slave --master-id cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7006 10.1.5.205:7002
>>> Adding node 10.1.5.205:7006 to cluster 10.1.5.205:7002
>>> Performing Cluster Check (using node 10.1.5.205:7002)
M: 5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002
   slots:5461-10922 (5462 slots) master
   1 additional replica(s)
M: 10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003
   slots:10923-16383 (5461 slots) master
   1 additional replica(s)
S: a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004
   slots: (0 slots) slave
   replicates cef32f21f38fe366352ec43f3e271a221d34d712
S: 72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005
   slots: (0 slots) slave
   replicates 5289506a768c2253c10c91311b7135e5bee86bbe
S: 908c580f36779339c299115ba3a9cbecbfb3c0ba 10.1.5.205:7007
   slots: (0 slots) slave
   replicates 10518d23d2a78052c1c164ee68c197797651bf28
M: cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001
   slots:0-5460 (5461 slots) master
   1 additional replica(s)
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
>>> Send CLUSTER MEET to node 10.1.5.205:7006 to make it join the cluster.
Waiting for the cluster to join.
>>> Configure node as replica of 10.1.5.205:7001.
[OK] New node added correctly.
[root@hadoop05 7006redis]#
[root@hadoop05 7006redis]#
[root@hadoop05 7006redis]# cat nodes.conf
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 master - 0 1531902339260 22 connected 0-5460
ada7e6d3edcc409ab4c1ec30dbea58eb53c308b7 10.1.5.205:7006 myself,slave cef32f21f38fe366352ec43f3e271a221d34d712 0 0 0 connected
908c580f36779339c299115ba3a9cbecbfb3c0ba 10.1.5.205:7007 slave 10518d23d2a78052c1c164ee68c197797651bf28 0 1531902339361 21 connected
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 1531902339259 2 connected
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531902339259 21 connected 10923-16383
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 slave cef32f21f38fe366352ec43f3e271a221d34d712 0 1531902339461 22 connected
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 master - 0 1531902339159 2 connected 5461-10922
vars currentEpoch 22 lastVoteEpoch 0
[root@hadoop05 7006redis]#
```

删除节点：

语法：

del-node　　host:port node_id

举例：

redis-trib.rb del-node 10.1.5.205:7007

fc4c63986c1f2355a867382a8912bea49fdb5e68

该操作会直接修改 nodes.conf，永久生效，和方法一不同

```
[root@hadoop05 7001redis]# cat nodes.conf
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 slave cef32f21f38fe366352ec43f3e271a221d34d712 0 15318
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 myself,master - 0 0 22 connected 0-5460
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531898902038 21 connected 10923-16383
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 15318
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 master - 0 1531898900034 2 connected 5461-10922
a3bc24765d7c8e1d5de86db2978c091fd81758c9 10.1.5.205:7006 slave 10518d23d2a78052c1c164ee68c197797651bf28 0 15318
fc4c63986c1f2355a867382a8912bea49fdb5e68 10.1.5.205:7007 master - 0 1531898903144 0 connected
vars currentEpoch 22 lastVoteEpoch 12
[root@hadoop05 7001redis]#
[root@hadoop05 7001redis]# redis-trib.rb del-node 10.1.5.205:7007 fc4c63986c1f2355a867382a8912bea49fdb5e68
>>> Removing node fc4c63986c1f2355a867382a8912bea49fdb5e68 from cluster 10.1.5.205:7007
>>> Sending CLUSTER FORGET messages to the cluster...
>>> SHUTDOWN the node.
[root@hadoop05 7001redis]# cat nodes.conf
a627628c4bd24c1e34454ff67cafe86858ccc84f 10.1.5.205:7004 slave cef32f21f38fe366352ec43f3e271a221d34d712 0 15318
cef32f21f38fe366352ec43f3e271a221d34d712 10.1.5.205:7001 myself,master - 0 0 22 connected 0-5460
10518d23d2a78052c1c164ee68c197797651bf28 10.1.5.205:7003 master - 0 1531899584865 21 connected 10923-16383
72b17a468476e99060861cbc8de60ff5aacbe50e 10.1.5.205:7005 slave 5289506a768c2253c10c91311b7135e5bee86bbe 0 15318
5289506a768c2253c10c91311b7135e5bee86bbe 10.1.5.205:7002 master - 0 1531899580852 2 connected 5461-10922
a3bc24765d7c8e1d5de86db2978c091fd81758c9 10.1.5.205:7006 slave 10518d23d2a78052c1c164ee68c197797651bf28 0 15318
vars currentEpoch 22 lastVoteEpoch 12
[root@hadoop05 7001redis]#
```

## 七、在线迁移 slot

```
reshard        host:port
               --from <arg>
               --to <arg>
               --slots <arg>
               --yes
               --timeout <arg>
               --pipeline <arg>
```

**host:port**：这个是必传参数，用来从一个节点获取整个集群信息，相当于获取集群信息的入口。

**--from <arg>**：需要从哪些源节点上迁移 slot，可从多个源节点完成迁移，以逗号隔开，传递的是节点的 node id，还可以直接传递--from all，这样源节点就是集群的所有节点，不传递该参数的话，则会在迁移过程中提示用户输入。

**--to <arg>**：slot 需要迁移的目的节点的 node id，目的节点只能填写一个，不传递该参数的话，则会在迁移过程中提示用户输入。

**--slots <arg>**：需要迁移的 slot 数量，不传递该参数的话，则会在迁移过程中提示用户输入。

**--yes**：设置该参数，可以在打印执行 reshard 计划的时候，提示用户输入 yes 确认后再执行 reshard。

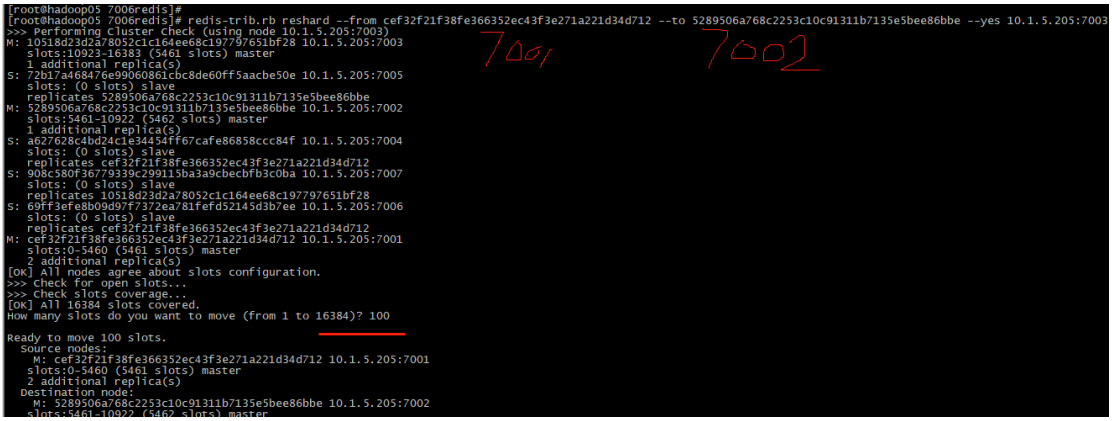
**--timeout <arg>**：设置 migrate 命令的超时时间。

**--pipeline <arg>**：定义 cluster getkeysinslot 命令一次取出的 key 数量，不传的话使用默认值为 10。

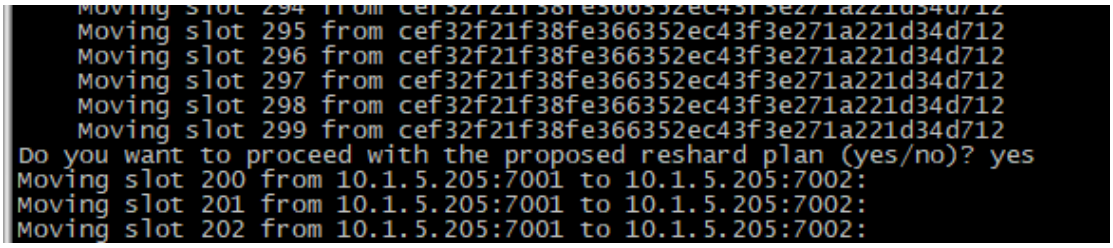
比如：从 7001 移动 100 个槽位到 7002，默认是从小到大，100 就是 0-99，可使用--slots 指定移动的个数。

举例：

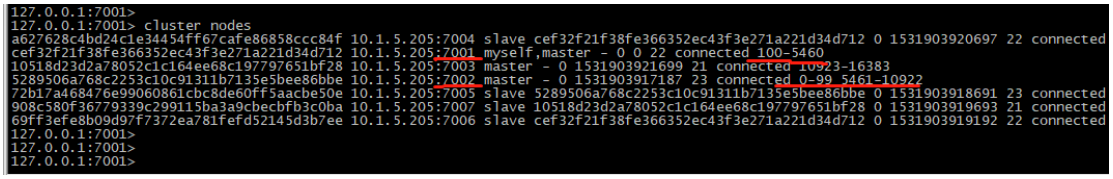
redis-trib.rb reshard --from cef32f21f38fe366352ec43f3e271a221d34d712 --to 5289506a768c2253c10c91311b7135e5bee86bbe --yes 10.1.5.205:7003



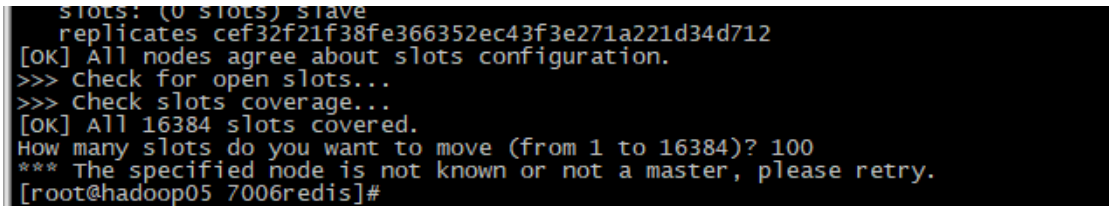

如果不加-yes，系统会提示如下：




移动后：




注意：to 的节点必须是 master，否则出现错误



## 八 平衡集群节点的 slot 数量

```
rebalance       host:port
                --weight <arg>
                --auto-weights
                --use-empty-masters
                --timeout <arg>
                --simulate
                --pipeline <arg>
                --threshold <arg>
```

host:port：这个是必传参数，用来从一个节点获取整个集群信息，相当于获取集群信息的入口。

--weight <arg>：节点的权重，格式为 node_id=weight，如果需要为多个节点分配权重的话，需要添加多个--weight <arg>参数，即--weight b31e3a2e=5 --weight 60b8e3a1=5，node_id 可为节点名称的前缀，只要保证前缀位数能唯一区分该节点即可。没有传递–weight 的节点的权重默认为 1。

--auto-weights：这个参数在 rebalance 流程中并未用到。

--threshold <arg>：只有节点需要迁移的 slot 阈值超过 threshold，才会执行 rebalance 操作。具体计算方法可以参考下面的 rebalance 命令流程的第四步。

--**use-empty-masters**：rebalance 是否考虑没有节点的 master，默认没有分配 slot 节点的 master 是不参与 rebalance 的，设置--use-empty-masters 可以让没有分配 slot 的节点参与 rebalance，这个参数在扩容集群 master 数量时特别有用。

--timeout <arg>：设置 migrate 命令的超时时间。

--simulate：设置该参数，可以模拟 rebalance 操作，提示用户会迁移哪些 slots，而不会真正执行迁移操作。

--pipeline <arg>：与 reshar 的 pipeline 参数一样，定义 cluster getkeysinslot 命令一

次取出的 key 数量，不传的话使用默认值为 10。

## 8.1 没有新 master 加入

**平衡前：**

```
[root@hadoop05 redis-cluster]# cat 7001redis/nodes.conf
faa7e2b785c07ed4b074b485cd78072e76337372 10.1.5.205:7003 master - 0 1531969323639 8 connected 100-199 10923-16383
bc3e08e793544c02c665fe7200053f86f45bc9bd 10.1.5.205:7004 slave da150342ca272529d5dab62a557da5206305c333 0 1531969319605 4 connected
cf285ac33cd79f2bfd604411185d15744ad5fcf0 10.1.5.205:7005 slave a84292141c294ad66fa0434c4e04c740769285e6 0 1531969324641 7 connected
da150342ca272529d5dab62a557da5206305c333 10.1.5.205:7001 myself,master - 0 0 1 connected 200-5460
a84292141c294ad66fa0434c4e04c740769285e6 10.1.5.205:7002 master - 0 1531969322635 7 connected 0-99 5461-10922
3d81d7eda6619c1deaa63fab6dfde9fe3ac71852 10.1.5.205:7006 slave faa7e2b785c07ed4b074b485cd78072e76337372 0 1531969321610 8 connected
vars currentEpoch 8 lastVoteEpoch 0
[root@hadoop05 redis-cluster]#
```

**执行平衡命令：**

redis-trib.rb rebalance  10.1.5.205:7001

```
[root@hadoop05 redis-cluster]# redis-trib.rb rebalance 10.1.5.205:7001
>>> Performing Cluster Check (using node 10.1.5.205:7001)
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
>>> Rebalancing across 3 nodes. Total weight = 3
Moving 101 slots from 10.1.5.205:7002 to 10.1.5.205:7001
#########################################################################################################
Moving 100 slots from 10.1.5.205:7003 to 10.1.5.205:7001
#########################################################################################################
```

可以明显的看到从 7002 和 7003 各移动 100 个 slot 到 7001

**平衡后：**

```
[root@hadoop05 redis-cluster]# cat 7001redis/nodes.conf
faa7e2b785c07ed4b074b485cd78072e76337372 10.1.5.205:7003 master - 0 1531969450955 8 connected 10923-16383
bc3e08e793544c02c665fe7200053f86f45bc9bd 10.1.5.205:7004 slave da150342ca272529d5dab62a557da5206305c333 0 1531969452963 9 connected
cf285ac33cd79f2bfd604411185d15744ad5fcf0 10.1.5.205:7005 slave a84292141c294ad66fa0434c4e04c740769285e6 0 1531969451975 7 connected
da150342ca272529d5dab62a557da5206305c333 10.1.5.205:7001 myself,master - 0 0 9 connected 0-5461
a84292141c294ad66fa0434c4e04c740769285e6 10.1.5.205:7002 master - 0 1531969449952 7 connected 5462-10922
3d81d7eda6619c1deaa63fab6dfde9fe3ac71852 10.1.5.205:7006 slave faa7e2b785c07ed4b074b485cd78072e76337372 0 1531969448951 8 connected
vars currentEpoch 9 lastVoteEpoch 0
```

## 8.2 有新的 master 加入

多加一个 master 节点进行平衡：

平衡前：

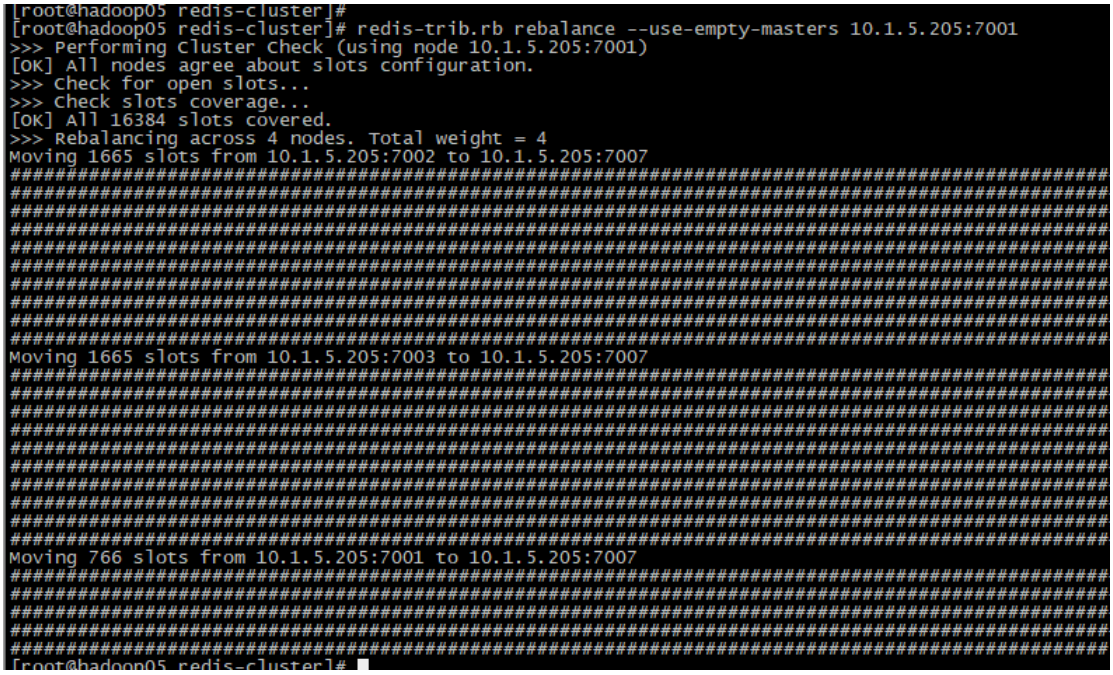
处于不平衡状态，且新加入的 7007 是 master，且未分配任何 slot。

```
[root@hadoop05 redis-cluster]# cat 7001redis/nodes.conf
9987adaef51933673afe6f5d7c0ea292c94f2182 10.1.5.205:7007 master - 0 1531970538226 0 connected
faa7e2b785c07ed4b074b485cd78072e76337372 10.1.5.205:7003 master - 0 1531970539228 13 connected 0-299 10923-16383
bc3e08e793544c02c665fe7200053f86f45bc9bd 10.1.5.205:7004 slave da150342ca272529d5dab62a557da5206305c333 0 1531970541
cf285ac33cd79f2bfd604411185d15744ad5fcf0 10.1.5.205:7005 slave a84292141c294ad66fa0434c4e04c740769285e6 0 1531970541
da150342ca272529d5dab62a557da5206305c333 10.1.5.205:7001 myself,master - 0 0 12 connected 600-5461
a84292141c294ad66fa0434c4e04c740769285e6 10.1.5.205:7002 master - 0 1531970542321 14 connected 300-599 5462-10922
3d81d7eda6619c1deaa63fab6dfde9fe3ac71852 10.1.5.205:7006 slave faa7e2b785c07ed4b074b485cd78072e76337372 0 1531970540
vars currentEpoch 14 lastVoteEpoch 0
```

**执行平衡命令：**

redis-trib.rb rebalance --**use-empty-masters** 10.1.5.205:7001

可以看到从 7002 7003 7001 各移动了一定数量的 slot 到 7007




**平衡后：**

新增的 master 节点也接管了一部分 slot，且 7007 无 slave