集群管理工具redis-trib.rb已经被废弃，所以不用安装ruby啥的了，当时redis-trib.rb的功能，现在已经集成到了redis-cli中，可以通过./redis-cli --cluster help查看使用方式。

环境

```
[root@mysql6 src]# ps -ef|grep redis
root     121195    1  0 09:54 ?        00:00:00 ./redis-server 127.0.0.1:6379 [cluster]
root     121203    1  0 09:55 ?        00:00:00 ./redis-server 127.0.0.1:6380 [cluster]
root     121209    1  0 09:55 ?        00:00:00 ./redis-server 127.0.0.1:6381 [cluster]
root     121214  3413  0 09:55 pts/1   00:00:00 grep redis
[root@mysql6 src]#
```

#创建集群#

./redis-cli --cluster create 192.168.1.172:6379 192.168.1.172:6380 192.168.1.172:6381

```
>>> Performing hash slots allocation on 3 nodes...
Master[0] -> Slots 0 - 5460
Master[1] -> Slots 5461 - 10922
Master[2] -> Slots 10923 - 16383
M: b97cde23f3c1a1b13e42728562180355b985831a 192.168.1.172:6379
   slots:[0-5460],[12571] (5461 slots) master
M: 6a1c0cfb9ed4abce0c0fee1525a62a6ca88e86e2 192.168.1.172:6380
   slots:[5461-10922] (5462 slots) master
M: 3333a86fb67b43d81e45b3b272deb49f9f99ea44 192.168.1.172:6381
   slots:[10923-16383] (5461 slots) master
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 192.168.1.172:6379)
M: b97cde23f3c1a1b13e42728562180355b985831a 192.168.1.172:6379
   slots:[0-5460] (5461 slots) master
M: 6a1c0cfb9ed4abce0c0fee1525a62a6ca88e86e2 127.0.0.1:6380
   slots:[5461-10922] (5462 slots) master
M: 3333a86fb67b43d81e45b3b272deb49f9f99ea44 127.0.0.1:6381
   slots:[10923-16383] (5461 slots) master
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
```

查看所有槽已经均匀分配

```
[root@mysql6 src]# ./redis-cli -p 6379
127.0.0.1:6379> cluster nodes
6a1c0cfb9ed4abce0c0fee1525a62a6ca88e86e2 127.0.0.1:6380@16380 master - 0 1540605540000 2 connected 5461-10922
b97cde23f3c1a1b13e42728562180355b985831a 192.168.1.172:6379@16379 myself,master - 0 1540605536000 1 connected 0-5460
3333a86fb67b43d81e45b3b272deb49f9f99ea44 127.0.0.1:6381@16381 master - 0 1540605540886 3 connected 10923-16383
127.0.0.1:6379> cluster info
cluster_state:ok
cluster_slots_assigned:16384
cluster_slots_ok:16384
cluster_slots_pfail:0
cluster_slots_fail:0
cluster_known_nodes:3
cluster_size:3
cluster_current_epoch:3
cluster_my_epoch:1
cluster_stats_messages_ping_sent:45
cluster_stats_messages_pong_sent:45
cluster_stats_messages_sent:90
cluster_stats_messages_ping_received:43
cluster_stats_messages_pong_received:45
cluster_stats_messages_meet_received:2
cluster_stats_messages_update_received:1
cluster_stats_messages_received:91
```

#检查集群#

./redis-cli --cluster check 192.168.1.172:6379

```
192.168.1.172:6379 (b97cde23...) -> 0 keys | 5461 slots | 0 slaves.
127.0.0.1:6380 (6a1c0cfb...) -> 0 keys | 5462 slots | 0 slaves.
127.0.0.1:6381 (3333a86f...) -> 0 keys | 5461 slots | 0 slaves.
[OK] 0 keys in 3 masters.
0.00 keys per slot on average.
>>> Performing Cluster Check (using node 192.168.1.172:6379)
M: b97cde23f3c1a1b13e42728562180355b985831a 192.168.1.172:6379
   slots:[0-5460] (5461 slots) master
M: 6a1c0cfb9ed4abce0c0fee1525a62a6ca88e86e2 127.0.0.1:6380
   slots:[5461-10922] (5462 slots) master
M: 3333a86fb67b43d81e45b3b272deb49f9f99ea44 127.0.0.1:6381
   slots:[10923-16383] (5461 slots) master
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.                          @51CTO博客
```

查看集群一切正常

#查看集群key、slot、slave分布信息#

./redis-cli --cluster info 192.168.1.172:6379

```
192.168.1.172:6379 (b97cde23...) -> 0 keys | 5461 slots | 0 slaves.
127.0.0.1:6380 (6a1c0cfb...) -> 0 keys | 5462 slots | 0 slaves.
127.0.0.1:6381 (3333a86f...) -> 0 keys | 5461 slots | 0 slaves.
[OK] 0 keys in 3 masters.
0.00 keys per slot on average.                         @51CTO博客
```

#在线迁移槽#

./redis-cli --cluster reshard 192.168.1.172:6379

选择一个目标节点的id

源选择all

```
[root@mysql6 src]# ./redis-cli --cluster info 192.168.1.172:6379
192.168.1.172:6379 (b97cde23...) -> 0 keys | 10461 slots | 0 slaves.
127.0.0.1:6380 (6a1c0cfb...) -> 0 keys | 2961 slots | 0 slaves.
127.0.0.1:6381 (3333a86f...) -> 0 keys | 2962 slots | 0 slaves.
[OK] 0 keys in 3 masters.
0.00 keys per slot on average.                         @51CTO博客
```

#平衡各节点槽数量#

./redis-cli --cluster rebalance --cluster-threshold 1 192.168.1.172:6379

```
>>> Performing cluster Check (using node 192.168.1.172:6379)
[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 = 3.00
Moving 2501 slots from 192.168.1.172:6379 to 127.0.0.1:6380
##########################################################################
##########################################################################
##########################################################################
##########################################################################
##########################################################################
##########################################################################
##########################################################################
##########################################################################
##########################################################################
Moving 2499 slots from 192.168.1.172:6379 to 127.0.0.1:6381
##########################################################################
##########################################################################
##########################################################################
##########################################################################
##########################################################################
##########################################################################
##########################################################################
##########################################################################
##########################################################################
##########################################################################
```

已平衡

```
[root@mysql6 src]# ./redis-cli --cluster info 192.168.1.172:6379
192.168.1.172:6379 (b97cde23...) -> 0 keys | 5461 slots | 0 slaves.
127.0.0.1:6380 (6a1c0cfb...) -> 0 keys | 5462 slots | 0 slaves.
127.0.0.1:6381 (3333a86f...) -> 0 keys | 5461 slots | 0 slaves.
[OK] 0 keys in 3 masters.
0.00 keys per slot on average.
```

#删除集群节点#

./redis-cli --cluster del-node 192.168.1.172:6379

b97cde23f3c1a1b13e42728562180355b985831a

这里必须是没有槽的节点，所以必须先移除槽，否则报如下错误

```
>>> Removing node b97cde23f3c1a1b13e42728562180355b985831a from cluster 19
[ERR] Node 192.168.1.172:6379 is not empty! Reshard data away and try again.
```

通过reshard迁移走槽后，删除成功，并且关闭了该节点

```
>>> Removing node b97cde23f3c1a1b13e42728562180355b985831a from cluster 192.168.1.172:6379
>>> Sending CLUSTER FORGET messages to the cluster...
>>> SHUTDOWN the node.
```

被删除的node重启后，依然记得集群中的其它节点，这是需要执行cluster forget nodeid来

忘记其它节点

#添加集群节点#

./redis-cli --cluster add-node 192.168.1.172:6379 192.168.1.172:6380

再平衡各节点slot数量

#将集群外部redis实例中的数据导入到集群中去#

./redis-cli --cluster import 192.168.1.172:6379 --cluster-from 192.168.1.172:6382 --cluster-copy

```
>>> Importing data from 192.168.1.172:6382 to cluster 192.168.1.172:6379
>>> Performing Cluster Check (using node 192.168.1.172:6379)
M: b97cde23f3c1a1b13e42728562180355b985831a 192.168.1.172:6379
   slots:[0-2500],[5000-7960] (5462 slots) master
M: 6a1c0cfb9ed4abce0c0fee1525a62a6ca88e86e2 127.0.0.1:6380
   slots:[7961-13421] (5461 slots) master
M: 3333a86fb67b43d81e45b3b272deb49f9f99ea44 127.0.0.1:6381
   slots:[2501-4999],[13422-16383] (5461 slots) master
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
*** Importing 5 keys from DB 0
Migrating name to 192.168.1.172:6379: OK
Migrating name2 to 192.168.1.172:6379: OK
Migrating name3 to 127.0.0.1:6381: OK
Migrating name4 to 127.0.0.1:6380: OK
Migrating name5 to 127.0.0.1:6380: OK
                                                    @51CTO博客
```

Cluster-from后面跟外部redis的ip和port

如果只使用cluster-copy，则要导入集群中的key不能在，否则如下：

```
*** Importing 5 keys from DB 0
Migrating name to 192.168.1.172:6379: Source 192.168.1.172:6382 rep@51CTO博客:
ERR Target instance replied with error: BUSYKEY Target key name already exists.
```

如果集群中已有同样的key，如果需要替换，可以cluster-copy和cluster-replace联用，这样集群中的key就会被替换为外部的

```
>>> Importing data from 192.168.1.172:6382 to cluster 192.168.1.172:6379
>>> Performing Cluster Check (using node 192.168.1.172:6379)
M: b97cde23f3c1a1b13e42728562180355b985831a 192.168.1.172:6379
   slots:[0-2500],[5000-7960] (5462 slots) master
M: 6a1c0cfb9ed4abce0c0fee1525a62a6ca88e86e2 127.0.0.1:6380
   slots:[7961-13421] (5461 slots) master
M: 3333a86fb67b43d81e45b3b272deb49f9f99ea44 127.0.0.1:6381
   slots:[2501-4999],[13422-16383] (5461 slots) master
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
*** Importing 5 keys from DB 0
Migrating name to 192.168.1.172:6379: OK
Migrating name2 to 192.168.1.172:6379: OK
Migrating name3 to 127.0.0.1:6381: OK
Migrating name4 to 127.0.0.1:6380: OK
Migrating name5 to 127.0.0.1:6380: OK
                                                    @51CTO博客
```