# Redis cluster集群搭建

集群至少需要6个节点（3主3从模式），每一个节点可以搭建在同一台机器上，也可以搭建在不同的服务器上。

1. **安装redis节点，指定端口**

**wget http://download.redis.io/releases/redis-3.2.0.tar.gz**

1. [root@localhost redis-3.2.0]# tar xzf redis-3.2.0.tar.gz
2. [root@localhost redis-3.2.0]# cd redis-3.2.0
3. [root@localhost redis-3.2.0]# make
4. [root@localhost redis01]# make install PREFIX=/usr/local/redis-cluster

在redis-cluster下 修改bin文件夹为redis01,复制redis.conf配置文件

配置redis的配置文件redis.conf：

daemonize yes #后台启动

port 7001 #修改端口号，从7001到7006

cluster-enabled yes #开启cluster，去掉注释

cluster-config-file nodes.conf # 节点配置文件

cluster-node-timeout 15000

appendonly yes #开启aof

上面记得把注释都去掉，不然会报错

然后复制六份，修改对应的端口号

1. **安装redis-trib（集群管理工具）所需的ruby脚本**

复制redis解压文件src下的redis-trib.rb文件到redis-cluster目录

1. [root@localhost redis-cluster]# cp /usr/local/redis/redis-3.2.0/src/redis-trib.rb ./

安装ruby环境：

1. [root@localhost redis-cluster]# yum install ruby
2. [root@localhost redis-cluster]# yum install rubygems

安装redis-trib.rb运行依赖的ruby的包redis-3.2.2.gem（网上下载，对应版本最好跟redis一致）

目前gem install redis是无法运行的，因为国外镜像ruby gem被墙，

到国内镜像下载即可，附上链接<https://gems.ruby-china.com/> 选择redis-3.2.2.gem下载，然后传输进来，运行下面命令即可

1. [root@localhost redis-cluster]# gem install redis-3.2.2.gem
2. **启动所有的redis节点**

可以写一个命令脚本start-all.sh

touch start-all.sh

最好不要复制下面粘贴进去，有时会出问题

1. cd redis01
2. ./redis-server redis.conf
3. cd ..
4. cd redis02
5. ./redis-server redis.conf
6. cd ..
7. cd redis03
8. ./redis-server redis.conf
9. cd ..
10. cd redis04
11. ./redis-server redis.conf
12. cd ..
13. cd redis05
14. ./redis-server redis.conf
15. cd ..
16. cd redis06
17. ./redis-server redis.conf
18. cd ..
19. 设置权限启动
20. [root@localhost redis-cluster]# chmod 777 start-all.sh
21. [root@localhost redis-cluster]# ./start-all.sh
22. 查看redis进程启动状态
23. [root@localhost redis-cluster]# ps -ef | grep redis
25. root       4547      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7001 [cluster]
26. root       4551      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7002 [cluster]
27. root       4555      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7003 [cluster]
28. root       4559      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7004 [cluster]
29. root       4563      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7005 [cluster]
30. root       4567      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7006 [cluster]
31. root       4840   4421  0 23:26 pts/1    00:00:00 grep --color=auto redis

### 使用redis-trib.rb创建集群

1. ./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

使用create命令 --replicas 1 参数表示为每个主节点创建一个从节点，其他参数是实例的地址集合。

## 三、redis的测试

1、测试存取值

客户端连接集群redis-cli需要带上 -c ，redis-cli -c -p 端口号

1. [root@localhost redis01]# ./redis-cli -c -p 7001
2. 127.0.0.1:7001> set name andy
3. -> Redirected to slot [5798] located at 127.0.0.1:7002
4. OK
5. 127.0.0.1:7002> get name
6. "andy"
7. 127.0.0.1:7002>

根据redis-cluster的key值分配，name应该分配到节点7002[5461-10922]上，上面显示redis cluster自动从7001跳转到了7002节点。

 我们可以测试一下7006从节点获取name值

1. [root@localhost redis06]# ./redis-cli -c -p 7006
2. 127.0.0.1:7006> get name
3. -> Redirected to slot [5798] located at 127.0.0.1:7002
4. "andy"
5. 127.0.0.1:7002>

  7006位7003的从节点，从上面也是自动跳转至7002获取值，这也是redis cluster的特点，它是去中心化，每个节点都是对等的，连接哪个节点都可以获取和设置数据。

## 四、集群节点选举

现在模拟将7002节点挂掉，按照redis-cluster原理会选举会将 7002的从节点7005选举为主节点。

1. [root@localhost redis-cluster]# ps -ef | grep redis
2. root       7950      1  0 12:50 ?        00:00:28 ./redis-server 127.0.0.1:7001 [cluster]
3. root       7952      1  0 12:50 ?        00:00:29 ./redis-server 127.0.0.1:7002 [cluster]
4. root       7956      1  0 12:50 ?        00:00:29 ./redis-server 127.0.0.1:7003 [cluster]
5. root       7960      1  0 12:50 ?        00:00:29 ./redis-server 127.0.0.1:7004 [cluster]
6. root       7964      1  0 12:50 ?        00:00:29 ./redis-server 127.0.0.1:7005 [cluster]
7. root       7966      1  0 12:50 ?        00:00:29 ./redis-server 127.0.0.1:7006 [cluster]
8. root      11346  10581  0 14:57 pts/2    00:00:00 grep --color=auto redis
9. [root@localhost redis-cluster]# kill 7952

在查看集群中的7002节点

1. [root@localhost redis-cluster]#
2. [root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7002
3. [ERR] Sorry, can't connect to node 127.0.0.1:7002
4. [root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7005
5. >>> Performing Cluster Check (using node 127.0.0.1:7005)
6. M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005
7. slots:5461-10922 (5462 slots) master
8. 0 additional replica(s)
9. S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004
10. slots: (0 slots) slave
11. replicates dd19221c404fb2fc4da37229de56bab755c76f2b
12. M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003
13. slots:10923-16383 (5461 slots) master
14. 1 additional replica(s)
15. M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001
16. slots:0-5460 (5461 slots) master
17. 1 additional replica(s)
18. S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006
19. slots: (0 slots) slave
20. replicates f9886c71e98a53270f7fda961e1c5f730382d48f
21. [OK] All nodes agree about slots configuration.
22. >>> Check for open slots...
23. >>> Check slots coverage...
24. [OK] All 16384 slots covered.
25. [root@localhost redis-cluster]#

可以看到集群连接不了7002节点，而7005有原来的S转换为M节点，代替了原来的7002节点。我们可以获取name值：

1. [root@localhost redis01]# ./redis-cli -c -p 7001
2. 127.0.0.1:7001> get name
3. -> Redirected to slot [5798] located at 127.0.0.1:7005
4. "andy"
5. 127.0.0.1:7005>
6. 127.0.0.1:7005>

从7001节点连入，自动跳转到7005节点，并且获取name值。

现在我们将7002节点恢复，看是否会自动加入集群中以及充当的M还是S节点。

1. [root@localhost redis-cluster]# cd redis02
2. [root@localhost redis02]# ./redis-server redis.conf
3. [root@localhost redis02]#

 在check一下7002节点

1. [root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7002
2. >>> Performing Cluster Check (using node 127.0.0.1:7002)
3. S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002
4. slots: (0 slots) slave
5. replicates a5db243087d8bd423b9285fa8513eddee9bb59a6
6. M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003
7. slots:10923-16383 (5461 slots) master
8. 1 additional replica(s)
9. M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005
10. slots:5461-10922 (5462 slots) master
11. 1 additional replica(s)
12. S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004
13. slots: (0 slots) slave
14. replicates dd19221c404fb2fc4da37229de56bab755c76f2b
15. S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006
16. slots: (0 slots) slave
17. replicates f9886c71e98a53270f7fda961e1c5f730382d48f
18. M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001
19. slots:0-5460 (5461 slots) master
20. 1 additional replica(s)
21. [OK] All nodes agree about slots configuration.
22. >>> Check for open slots...
23. >>> Check slots coverage...
24. [OK] All 16384 slots covered.
25. [root@localhost redis-cluster]#

可以看到7002节点变成了a5db243087d8bd423b9285fa8513eddee9bb59a6 7005的从节点。

## 五、集群节点添加

### 1、新增主节点

       新增一个节点7007作为主节点修改配置文件

1. [root@localhost redis-cluster]# cp -r  redis01 redis07
2. [root@localhost redis-cluster]# cd redis07/
3. [root@localhost redis07]# sed -i "s/7001/7007/g" ./redis.conf

删除redis07下aof、rdb、nodes.conf等本地备份文件删除；

同时将新Node的集群配置文件删除,即：删除你redis.conf里面cluster-config-file所在的文件；

启动7007redis服务

1. [root@localhost redis07]# ./redis-server redis.conf
2. [root@localhost redis07]# netstat -anp | grep 7007
3. tcp        0      0 127.0.0.1:17007         0.0.0.0:\*               LISTEN      13441/./redis-serve
4. tcp        0      0 127.0.0.1:7007          0.0.0.0:\*               LISTEN      13441/./redis-serve
5. [root@localhost redis07]#

  上面可以看到，7007已经启动，现在加入集群中。添加使用redis-trib.rb的add-node命令

1. ./redis-trib.rb add-node 127.0.0.1:7007 127.0.0.1:7002

    add-node是加入集群节点，127.0.0.1:7007为要加入的节点，127.0.0.1:7002 表示加入的集群的一个节点，用来辨识是哪个集群，理论上那个集群的节点都可以。

    执行以下add-node

1. [root@localhost redis-cluster]# ./redis-trib.rb add-node 127.0.0.1:7007 127.0.0.1:7002
2. >>> Adding node 127.0.0.1:7007 to cluster 127.0.0.1:7002
3. >>> Performing Cluster Check (using node 127.0.0.1:7002)
4. S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002
5. slots: (0 slots) slave
6. replicates a5db243087d8bd423b9285fa8513eddee9bb59a6
7. M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003
8. slots:10923-16383 (5461 slots) master
9. 1 additional replica(s)
10. M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005
11. slots:5461-10922 (5462 slots) master
12. 1 additional replica(s)
13. S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004
14. slots: (0 slots) slave
15. replicates dd19221c404fb2fc4da37229de56bab755c76f2b
16. S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006
17. slots: (0 slots) slave
18. replicates f9886c71e98a53270f7fda961e1c5f730382d48f
19. M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001
20. slots:0-5460 (5461 slots) master
21. 1 additional replica(s)
22. [OK] All nodes agree about slots configuration.
23. >>> Check for open slots...
24. >>> Check slots coverage...
25. [OK] All 16384 slots covered.
26. >>> Send CLUSTER MEET to node 127.0.0.1:7007 to make it join the cluster.
27. [OK] New node added correctly.
28. [root@localhost redis-cluster]#

     可以看到7007加入这个Cluster，并成为一个新的节点。

     可以check以下7007节点状态

1. [root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7007
2. >>> Performing Cluster Check (using node 127.0.0.1:7007)
3. M: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007
4. slots: (0 slots) master
5. 0 additional replica(s)
6. S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006
7. slots: (0 slots) slave
8. replicates f9886c71e98a53270f7fda961e1c5f730382d48f
9. M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001
10. slots:0-5460 (5461 slots) master
11. 1 additional replica(s)
12. M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003
13. slots:10923-16383 (5461 slots) master
14. 1 additional replica(s)
15. S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002
16. slots: (0 slots) slave
17. replicates a5db243087d8bd423b9285fa8513eddee9bb59a6
18. M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005
19. slots:5461-10922 (5462 slots) master
20. 1 additional replica(s)
21. S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004
22. slots: (0 slots) slave
23. replicates dd19221c404fb2fc4da37229de56bab755c76f2b
24. [OK] All nodes agree about slots configuration.
25. >>> Check for open slots...
26. >>> Check slots coverage...
27. [OK] All 16384 slots covered.
28. [root@localhost redis-cluster]#

  M: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007  
   slots: (0 slots) master  
   0 additional replica(s)

  上面信息可以看到有4个M节点，3个S节点，7007成为了M主节点，它没有附属的从节点，而且Cluster并未给7007分配哈希卡槽（0 slots）。

   可以从客户端连接集群查看一下，集群节点的连接情况

1. [root@localhost redis-cluster]# cd redis07/
2. [root@localhost redis07]# ./redis-cli -c -p 7007
3. 127.0.0.1:7007> cluster nodes
4. 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006 slave f9886c71e98a53270f7fda961e1c5f730382d48f 0 1462955393326 3 connected
5. dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001 master - 0 1462955388247 1 connected 0-5460
6. ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007 myself,master - 0 0 0 connected
7. f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003 master - 0 1462955390270 3 connected 10923-16383
8. 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002 slave a5db243087d8bd423b9285fa8513eddee9bb59a6 0 1462955394334 7 connected
9. a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005 master - 0 1462955392309 7 connected 5461-10922
10. 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004 slave dd19221c404fb2fc4da37229de56bab755c76f2b 0 1462955389663 1 connected
11. 127.0.0.1:7007>

     redis-cluster在新增节点时并未分配卡槽，需要我们手动对集群进行重新分片迁移数据，需要重新分片命令 reshard

     redis-trib.rb reshard 127.0.0.1:7005

     这个命令是用来迁移slot节点的，后面的127.0.0.1:7005是表示是哪个集群，端口填［7000-7007］都可以，执行结果如下：

1. [root@localhost redis-cluster]# ./redis-trib.rb reshard 127.0.0.1:7005
2. >>> Performing Cluster Check (using node 127.0.0.1:7005)
3. M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005
4. slots:5461-10922 (5462 slots) master
5. 1 additional replica(s)
6. S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004
7. slots: (0 slots) slave
8. replicates dd19221c404fb2fc4da37229de56bab755c76f2b
9. M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003
10. slots:10923-16383 (5461 slots) master
11. 1 additional replica(s)
12. S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002
13. slots: (0 slots) slave
14. replicates a5db243087d8bd423b9285fa8513eddee9bb59a6
15. M: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007
16. slots: (0 slots) master
17. 0 additional replica(s)
18. M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001
19. slots:0-5460 (5461 slots) master
20. 1 additional replica(s)
21. S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006
22. slots: (0 slots) slave
23. replicates f9886c71e98a53270f7fda961e1c5f730382d48f
24. [OK] All nodes agree about slots configuration.
25. >>> Check for open slots...
26. >>> Check slots coverage...
27. [OK] All 16384 slots covered.
28. How many slots do you want to move (from 1 to 16384)?

      它提示我们需要迁移多少slot到7007上，我们平分16384个哈希槽给4个节点：16384/4 = 4096，我们需要移动4096个槽点到7007上。

1. [OK] All 16384 slots covered.
2. How many slots do you want to move (from 1 to 16384)? 4096
3. What is the receiving node ID?

     需要输入7007的节点id，ee3efb90e5ac0725f15238a64fc60a18a71205d7

1. Please enter all the source node IDs.
2. Type 'all' to use all the nodes as source nodes for the hash slots.
3. Type 'done' once you entered all the source nodes IDs.
4. Source node #1:

      redis-trib 会向你询问重新分片的源节点（source node），即，要从特点的哪个节点中取出 4096 个哈希槽，还是从全部节点提取4096个哈希槽， 并将这些槽移动到7007节点上面。  
  
     如果我们不打算从特定的节点上取出指定数量的哈希槽，那么可以向redis-trib输入 all，这样的话， 集群中的所有主节点都会成为源节点，redis-trib从各个源节点中各取出一部分哈希槽，凑够4096个，然后移动到7007节点上：

1. Source node #1:all

    然后开始从别的主节点迁移哈希槽，并且确认。

1. Moving slot 1343 from dd19221c404fb2fc4da37229de56bab755c76f2b
2. Moving slot 1344 from dd19221c404fb2fc4da37229de56bab755c76f2b
3. Moving slot 1345 from dd19221c404fb2fc4da37229de56bab755c76f2b
4. Moving slot 1346 from dd19221c404fb2fc4da37229de56bab755c76f2b
5. Moving slot 1347 from dd19221c404fb2fc4da37229de56bab755c76f2b
6. Moving slot 1348 from dd19221c404fb2fc4da37229de56bab755c76f2b
7. Moving slot 1349 from dd19221c404fb2fc4da37229de56bab755c76f2b
8. Moving slot 1350 from dd19221c404fb2fc4da37229de56bab755c76f2b
9. Moving slot 1351 from dd19221c404fb2fc4da37229de56bab755c76f2b
10. Moving slot 1352 from dd19221c404fb2fc4da37229de56bab755c76f2b
11. Moving slot 1353 from dd19221c404fb2fc4da37229de56bab755c76f2b
12. Moving slot 1354 from dd19221c404fb2fc4da37229de56bab755c76f2b
13. Moving slot 1355 from dd19221c404fb2fc4da37229de56bab755c76f2b
14. Moving slot 1356 from dd19221c404fb2fc4da37229de56bab755c76f2b
15. Moving slot 1357 from dd19221c404fb2fc4da37229de56bab755c76f2b
16. Moving slot 1358 from dd19221c404fb2fc4da37229de56bab755c76f2b
17. Moving slot 1359 from dd19221c404fb2fc4da37229de56bab755c76f2b
18. Moving slot 1360 from dd19221c404fb2fc4da37229de56bab755c76f2b
19. Moving slot 1361 from dd19221c404fb2fc4da37229de56bab755c76f2b
20. Moving slot 1362 from dd19221c404fb2fc4da37229de56bab755c76f2b
21. Moving slot 1363 from dd19221c404fb2fc4da37229de56bab755c76f2b
22. Moving slot 1364 from dd19221c404fb2fc4da37229de56bab755c76f2b
23. Do you want to proceed with the proposed reshard plan (yes/no)? yes

     确认之后，redis-trib就开始执行分片操作，将哈希槽一个一个从源主节点移动到7007目标主节点。

     重新分片结束后我们可以check以下节点的分配情况。

1. [root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001
2. >>> Performing Cluster Check (using node 127.0.0.1:7001)
3. M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001
4. slots:1365-5460 (4096 slots) master
5. 1 additional replica(s)
6. M: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007
7. slots:0-1364,5461-6826,10923-12287 (4096 slots) master
8. 0 additional replica(s)
9. M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005
10. slots:6827-10922 (4096 slots) master
11. 1 additional replica(s)
12. S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006
13. slots: (0 slots) slave
14. replicates f9886c71e98a53270f7fda961e1c5f730382d48f
15. M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003
16. slots:12288-16383 (4096 slots) master
17. 1 additional replica(s)
18. S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002
19. slots: (0 slots) slave
20. replicates a5db243087d8bd423b9285fa8513eddee9bb59a6
21. S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004
22. slots: (0 slots) slave
23. replicates dd19221c404fb2fc4da37229de56bab755c76f2b
24. [OK] All nodes agree about slots configuration.
25. >>> Check for open slots...
26. >>> Check slots coverage...
27. [OK] All 16384 slots covered.
28. [root@localhost redis-cluster]#

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

   可以看到7007节点分片的哈希槽片不是连续的，间隔的移动。

1. [root@localhost redis-cluster]# cd redis07/
2. [root@localhost redis07]# ./redis-cli -c 7007
3. Could not connect to Redis at 127.0.0.1:6379: Connection refused
4. [root@localhost redis07]# ./redis-cli -c -p 7007
5. 127.0.0.1:7007> keys \*
6. 1) "name"
7. 2) "age"
8. 127.0.0.1:7007>
9. 127.0.0.1:7007>

    可以看到将7001的age[741]和name[5798]移动到7007节点上，

    主节点7007添加成功。