# Redis 4 Cluster 集群

作者在进行企业培训时，发现有的企业还在生产环境使用Redis 4集群，说明Redis 4集群还是有一定市场的，所以把Redis 4集群的配置也做一个总结，读者可自行比较与Redis 5集群配置的区别。



## 集群环境

本集群环境使用1台节点服务器，每台服务器开启6个Redis实例，开启了6个端口，模拟3个Master(主节点)和3个Slave(从节点）环境，组成一个Redis Cluster(集群）。本集群实验采用三主三从，每个Master(主节点)处理各自的数据，提供读写能力，Slave(从节点)异步复制主节点的数据。

Redis 4 集群的实验环境

|  |  |  |  |
| --- | --- | --- | --- |
| IP地址 | 端口 | 版本 | 系统 |
| 192.168.11.10 | 7001 | Redis 4.0.6 | CentOS7 64位 |
| 7002 |
| 7003 |
| 7004 |
| 7005 |
| 7006 |

以root用户登录LInux服务器。

## 开始Redis 4 集群搭建

**第一步：安装Redis 4**

1，首先，安装gcc基础依赖包，使用如下命令。

$ yum install gcc-c++

2, 下载Redis 4并解压缩

$ mkdir /upload

$ cd /upload

$ wget http://download.redis.io/releases/redis-4.0.6.tar.gz

$ tar -zxvf redis-4.0.6.tar.gz

3，编译安装Redis 4

进入解压后的目录

$ cd redis-4.0.6/

使用make命令编译Redis源文件。

$ make MALLOC=libc

编译成功后，安装Redis

$ make install PREFIX=/usr/local/redis

安装成功后，需要将Redis的命令所在目录添加到系统参数PATH中，修改/etc/profile文件。

vi /etc/profile

在 /etc/profile文件最后一行添加以下内容。

export PATH=$PATH:/usr/local/redis/bin

然后使用source命令让/etc/profile这个文件立即生效。

$ source /etc/profile

安装成功后，需要对Redis进行部署，把Redis的配置文件 redis.conf拷贝到 “/usr/local/redis/conf”目录下。

//创建集群目录

$ mkdir /usr/local/redis/conf

$ cp redis.conf /usr/local/redis/conf

至此， Redis在Linux上的安装和配置就结束了。

第二步：在 /usr/local下创建文件夹redis-cluster，然后在其下面创建6个文件夾如下。

$ mkdir -p /usr/local/redis\_cluster

$ cd /usr/local/redis\_cluster/

$ mkdir 7001 7002 7003 7004 7005 7006

Redis集群需要至少要三个master节点，我们使用一台服务器，在这台服务器搭建三个master节点，并且给每个master再搭建一个slave节点，总共6个redis节点。

第三步：把之前的redis.conf配置文件copy到7001-7006目录下。

//创建7001节点为例，拷贝到7001目录

$ cp /usr/local/redis/conf/redis.conf /usr/local/redis\_cluster/7001

修改redis.conf为如下内容。

1. port 7001 修改为主机的端口
2. bind 192.168.1.10 修改为主机的IP，默认ip为127.0.0.1 ，需要改为其他节点机器可访问的ip
3. daemonize yes Redis后台启动
4. pidfile /var/run/redis\_7001.pid
5. cluster-enabled yes 开启集群，把注释#去掉
6. cluster-config-file nodes\_7001.conf 集群节点信息文件，这里nodes-7001.conf最好和port对应上
7. cluster-node-timeout 5000 请求超时 ，单位是毫秒，设置5秒够了
8. appendonly yes AOP日志开启，会每次写操作时都会记录一条日志

第四步：把修改后的redis.conf配置文件，复制到7002-7006，修改第1、2、4、6项里的端口号和配置文件名称。

$ cd /usr/local/redis-cluster/7001

//拷贝到7002目录

$ cp redis.conf /usr/local/redis\_cluster/7002

//拷贝到7003目录

$ cp redis.conf /usr/local/redis\_cluster/7003

//拷贝到7004目录

$ cp redis.conf /usr/local/redis\_cluster/7004

//拷贝到7005目录

$ cp redis.conf /usr/local/redis\_cluster/7005

//拷贝到7006目录

$ cp redis.conf /usr/local/redis\_cluster/7006

第五步：分别启动6个Redis实例，然后检查是否启动成功。

$ redis-server /usr/local/redis\_cluster/7001/redis.conf

$ redis-server /usr/local/redis\_cluster/7002/redis.conf

$ redis-server /usr/local/redis\_cluster/7003/redis.conf

$ redis-server /usr/local/redis\_cluster/7004/redis.conf

$ redis-server /usr/local/redis\_cluster/7005/redis.conf

$ redis-server /usr/local/redis\_cluster/7006/redis.conf

执行启动Redis实例命令的返回结果如下所示。



图 6‑27 启动6个Redis节点

使用命令ps -ef | grep redis，查看Redis是否启动成功。

[root@localhost 7006]# ps -ef | grep redis

root 5170 1 0 15:00 ? 00:00:00 redis-server 127.0.0.1:7001 [cluster]

root 5175 1 0 15:00 ? 00:00:00 redis-server 127.0.0.1:7002 [cluster]

root 5180 1 0 15:00 ? 00:00:00 redis-server 127.0.0.1:7003 [cluster]

root 5185 1 0 15:01 ? 00:00:00 redis-server 127.0.0.1:7004 [cluster]

root 5221 1 0 15:01 ? 00:00:00 redis-server 127.0.0.1:7005 [cluster]

root 5226 1 0 15:01 ? 00:00:00 redis-server 127.0.0.1:7006 [cluster]

root 5247 4061 0 15:03 pts/1 00:00:00 grep --color=auto redis

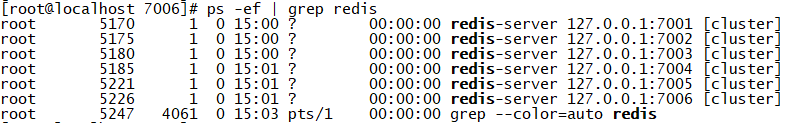


图 6‑28 查看Redis 进程

可以看到6个节点全部启动成功。

第六步：用redis-cli创建整个Redis集群

Redis4及以前的版本集群是依靠ruby脚本redis-trib.rb实现。关闭防火墙服务和selinx，避免与Redis的端口冲突，这部分的内容请参考13.2.1关闭防火墙。

已经准备好了搭建集群的Redis节点，接下来我们要把这些节点都串连起来搭建集群。redis-trib.rb是Redis官方推出的管理redis集群的工具，集成在Redis的源码src目录下，是基于Redis提供的集群命令封装成简单、便捷和实用的操作工具。redis-trib.rb是Redis作者用ruby完成的。所以要执行redis-trib.rb命令，需要安装Ruby环境。

下载编译Ruby需要的安装包，在主机上执行以下命令。

$ yum -y install ruby ruby-devel rubygems rpm-build

下载Ruby依赖的redis.gem，可以去rubygems官网下载最新的redis.gem源码。

$ wget https://rubygems.org/downloads/redis-3.3.1.gem

在本地安装gem环境。

$ gem install --local redis-3.3.1.gem

接下来运行一下redis-trib.rb 文件, redis-trib.rb在Redis的源码src文件中。确认所有的节点都启动，接下来使用参数create 创建 (在192.168.11.11中来创建)

$ cd /upload/redis-4.0.6/src

$ **./redis-trib.rb create --replicas 1 192.168.11.10:7001 192.168.11.10:7002 192.168.11.10:7003 192.168.11.10:7004 192.168.11.10:7005 192.168.11.10:7006**

redis-trib.rb参数说明：

* create 表示创建一个Redis cluster集群。
* --replicas 1 表示为集群中每一个主节点(master)指定一个从节点(slave)，即一比一的复制。

本例有6个节点，程序会按照一定规则生成 3个master（主节点）3个slave(从节点)。

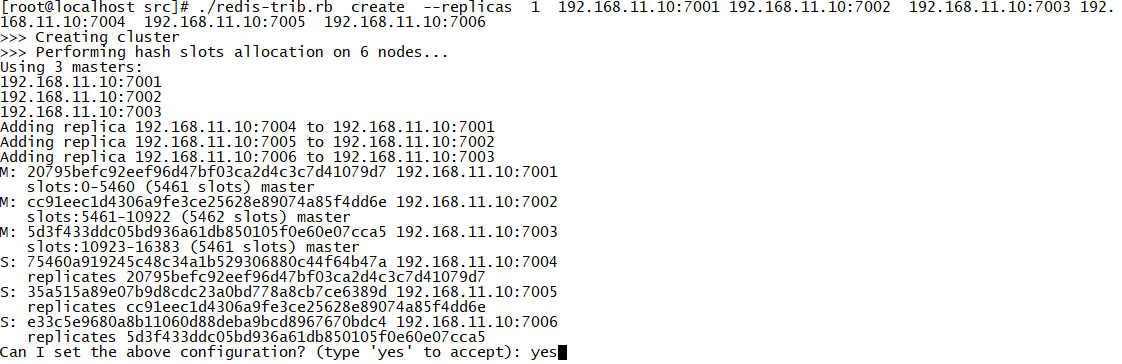
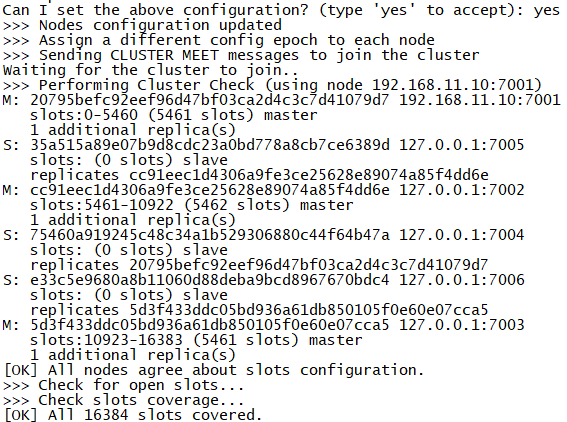


图 6‑29 生成Redis4的集群

创建集群时，需要按照提示输入同意yes。从屏幕输出可以很容易的看出那些是主节点(master)，那些是从节点(slave)。



查看集群的主节点

$ **redis-cli -c -h 192.168.11.10 -p 7001 cluster nodes | grep master**

cc91eec1d4306a9fe3ce25628e89074a85f4dd6e 127.0.0.1:7002@17002 master - 0 1570173937238 2 connected 5461-10922

5d3f433ddc05bd936a61db850105f0e60e07cca5 127.0.0.1:7003@17003 master - 0 1570173937539 3 connected 10923-16383

20795befc92eef96d47bf03ca2d4c3c7d41079d7 192.168.11.10:7001@17001 myself,master - 0 1570173937000 1 connected 0-5460

查看集群的从节点

$ redis-cli -c -h 192.168.11.10 -p 7001 cluster nodes | grep slave

35a515a89e07b9d8cdc23a0bd778a8cb7ce6389d 127.0.0.1:7005@17005 slave cc91eec1d4306a9fe3ce25628e89074a85f4dd6e 0 1570173995000 5 connected

75460a919245c48c34a1b529306880c44f64b47a 127.0.0.1:7004@17004 slave 20795befc92eef96d47bf03ca2d4c3c7d41079d7 0 1570173996561 4 connected

e33c5e9680a8b11060d88deba9bcd8967670bdc4 127.0.0.1:7006@17006 slave 5d3f433ddc05bd936a61db850105f0e60e07cca5 0 1570173996662 6 connected

以上集群节点信息可以转换成如下的表格。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 节点ID | IP | 端口 | 角色 | 插槽范围 |
| 20795befc92eef96d47bf03ca2d4c3c7d41079d7 | 192.168.11.10 | 7001 | master | 0-5460 |
| cc91eec1d4306a9fe3ce25628e89074a85f4dd6e | 7002 | master | 5461-10922 |
| 5d3f433ddc05bd936a61db850105f0e60e07cca5 | 7003 | master | 10923-16383 |
| 35a515a89e07b9d8cdc23a0bd778a8cb7ce6389d | 7005 | slave | 7002 |
| 75460a919245c48c34a1b529306880c44f64b47a | 7004 | slave | 7001 |
| e33c5e9680a8b11060d88deba9bcd8967670bdc4 | 7006 | slave | 7003 |

集群的主从节点如下图所示，有三个主节点(master)和三个从节点(slave)，一个主节点对应一个从节点，成一对一的对应关系，如下图所示。

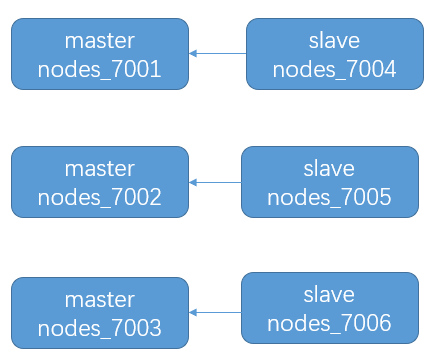


图 6‑30 集群的主从节点

redis-cli命令连接Redis集群，连接集群时需要加上参数 **–c** 。

$ redis-cli -c -h 192.168.11.10 -p 7001

$ redis-cli -c -h 192.168.11.10 -p 7002

$ redis-cli -c -h 192.168.11.10 -p 7003

$ redis-cli -c -h 192.168.11.10 -p 7004

$ redis-cli -c -h 192.168.11.10 -p 7005

$ redis-cli -c -h 192.168.11.10 -p 7006

测试Redis集群是否正常。

[root@localhost src]# redis-cli -c -h 192.168.11.10 -p 7001

192.168.11.10:7001> SET name xinping

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

OK

127.0.0.1:7002> SET age 33

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

OK

关闭集群一个节点，比如关闭集群中端口号为7005的节点使用如下命令。

[root@localhost conf]# redis-cli -h 192.168.11.10 -p 7005

192.168.11.11:7005> shutdown

也可以使用如下命令。

[root@localhost conf]# redis-cli -c -h 192.168.11.10 -p 7005 shutdown

再查看集群状态会发现 7005节点已经关闭了。

127.0.0.1:7001> cluster nodes

79a9bc3df0c769e3d57957c86b8420df83155afc 127.0.0.1:7003@17003 master - 0 1570176074728 3 connected 10923-16383

0c67831bea7d77f0dfb18ac8a6478d75535780f4 127.0.0.1:7005@17005 slave,**fail** fd5993d2accfefc7ea5759cff4a88e62a6392323 1570176068682 1570176066592 5 disconnected

fd5993d2accfefc7ea5759cff4a88e62a6392323 127.0.0.1:7002@17002 master - 0 1570176073000 2 connected 5461-10922

d0e99032ec0cd963160c5e47dbef0b1ed5b56ad3 127.0.0.1:7004@17004 slave 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 0 1570176073722 4 connected

03e6f7b06aea39302b48cbb29f1b3209df0fde70 127.0.0.1:7006@17006 slave 79a9bc3df0c769e3d57957c86b8420df83155afc 0 1570176074000 6 connected

2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 192.168.11.10:7001@17001 myself,master - 0 1570176074000 1 connected 0-5460

重新启动 7005节点，需要在主机192.168.11.11上面执行以下命令

$ redis-server /usr/local/redis\_cluster/7005/redis.conf

关闭集群中的节点。

$ redis-cli -h 192.168.11.10 -p 7001 shutdown

$ redis-cli -h 192.168.11.10 -p 7002 shutdown

$ redis-cli -h 192.168.11.10 -p 7003 shutdown

$ redis-cli -h 192.168.11.10 -p 7004 shutdown

$ redis-cli -h 192.168.11.10 -p 7005 shutdown

$ redis-cli -h 192.168.11.10 -p 7006 shutdown

至此，Redis4集群环境搭建完毕。

## 新增集群节点

一， 增加主节点

在主机(192.168.11.10)新添加一个节点7007，执行以下操作。

$ cd /usr/local/redis\_cluster

$ mkdir 7007

拷贝一份redis.conf文件到7007目录。

$ cp /usr/local/redis/conf/redis.conf /usr/local/redis\_cluster/7007

修改7007文件加下的redis.conf文件。

$ cd /usr/local/redis\_cluster/7007

$ vi redis.conf

修改以下内容。

port 7007

bind 192.168.11.10

daemonize yes

pidfile /var/run/redis\_7007.pid

# 开启集群 把注释#去掉

cluster-enabled yes

cluster-config-file nodes\_7006.conf

# 请求超时设置为5秒

cluster-node-timeout 5000

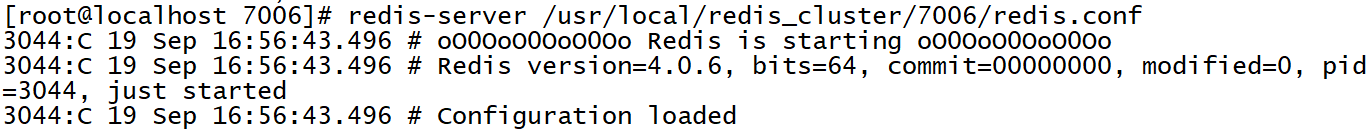
# aof日志开启,会每次写操作都记录一条日志

appendonly yes

1. 启动新添加的节点

在主机192.168.11.10执行以下命令。

$ redis-server /usr/local/redis\_cluster/7007/redis.conf



2，增加主节点，动态增加一个节点 nodes\_7007

$ cd /upload/redis-4.0.6/src

$ **./redis-trib.rb add-node 192.168.11.10:7007 192.168.11.10:7001**

参数说明：

第一个参数Add-node , 添加Redis节点。

第二个参数192.168.11.10:7007 , 是Redis新增节点。

第三个参数192.168.11.10:7001 ，Redis可以选集群中任何一个节点。

如果添加的是主节点，只需指定源节点和目标节点的地址即可。

[root@localhost src]# ./redis-trib.rb add-node 192.168.11.10:7007 192.168.11.10:7001

>>> Adding node 192.168.11.10:7007 to cluster 192.168.11.10:7001

>>> Performing Cluster Check (using node 192.168.11.10:7001)

M: 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 192.168.11.10:7001

slots:0-5460 (5461 slots) master

1 additional replica(s)

M: 79a9bc3df0c769e3d57957c86b8420df83155afc 127.0.0.1:7003

slots:10923-16383 (5461 slots) master

1 additional replica(s)

S: 0c67831bea7d77f0dfb18ac8a6478d75535780f4 127.0.0.1:7005

slots: (0 slots) slave

replicates fd5993d2accfefc7ea5759cff4a88e62a6392323

M: fd5993d2accfefc7ea5759cff4a88e62a6392323 127.0.0.1:7002

slots:5461-10922 (5462 slots) master

1 additional replica(s)

S: d0e99032ec0cd963160c5e47dbef0b1ed5b56ad3 127.0.0.1:7004

slots: (0 slots) slave

replicates 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a

S: 03e6f7b06aea39302b48cbb29f1b3209df0fde70 127.0.0.1:7006

slots: (0 slots) slave

replicates 79a9bc3df0c769e3d57957c86b8420df83155afc

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

>>> Send CLUSTER MEET to node 192.168.11.10:7007 to make it join the cluster.

[OK] New node added correctly.

已经添加成功！查看集群信息，可以看到7006的Redis节点状态为master。

[root@localhost src]# **redis-cli -c -h 192.168.11.10 -p 7001 cluster nodes**

79a9bc3df0c769e3d57957c86b8420df83155afc 127.0.0.1:7003@17003 master - 0 1570178323303 3 connected 10923-16383

0c67831bea7d77f0dfb18ac8a6478d75535780f4 127.0.0.1:7005@17005 slave fd5993d2accfefc7ea5759cff4a88e62a6392323 0 1570178323504 5 connected

fd5993d2accfefc7ea5759cff4a88e62a6392323 127.0.0.1:7002@17002 master - 0 1570178323000 2 connected 5461-10922

5c8e074636cfb8f67a44d9a55982299565b6e9e1 127.0.0.1:7007@17007 master - 0 1570178323000 0 connected

d0e99032ec0cd963160c5e47dbef0b1ed5b56ad3 127.0.0.1:7004@17004 slave 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 0 1570178323504 4 connected

03e6f7b06aea39302b48cbb29f1b3209df0fde70 127.0.0.1:7006@17006 slave 79a9bc3df0c769e3d57957c86b8420df83155afc 0 1570178323806 6 connected

2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 192.168.11.10:7001@17001 myself,master - 0 1570178323000 1 connected 0-5460

7006发现没有插槽数，接下来需要给7006这个节点将7000的一部分（1000个）插槽分配给7006，用下面的命令可以重新分配插槽(slot)。

$ cd /upload/redis-4.0.6/src

[root@localhost src]# **./redis-trib.rb reshard 192.168.11.10:7001**

>>> Performing Cluster Check (using node 192.168.11.10:7001)

M: 9ed6d860785f2e4a0bf7761abe9a0b6f3280a4d6 192.168.11.10:7001

slots:0-5460 (5461 slots) master

0 additional replica(s)

M: fba16de5a20c73eb52f8d482f616fd8f2443368d 192.168.11.11:7006

slots: (0 slots) master

0 additional replica(s)

S: 2113489e118b5ae2ee6ac4ec1b88a703cde8b331 192.168.11.11:7004

slots: (0 slots) slave

replicates e4bafc33f3dbc2aa234c482f997428e91159b45a

M: e4bafc33f3dbc2aa234c482f997428e91159b45a 192.168.11.10:7001

slots:10923-16383 (5461 slots) master

2 additional replica(s)

S: 5dc15d549306570a6e71f33c1c0f302bebe103a4 192.168.11.10:7002

slots: (0 slots) slave

replicates df7b3ff4bc7735bc4ddc0933374d628f75cc250f

M: df7b3ff4bc7735bc4ddc0933374d628f75cc250f 192.168.11.11:7003

slots:5461-10922 (5462 slots) master

1 additional replica(s)

S: b1cfe7a57700ac29666d825c20db65c52123d8fc 192.168.11.11:7005

slots: (0 slots) slave

replicates e4bafc33f3dbc2aa234c482f997428e91159b45a

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

# 从192.168.11.10重新分配1000个槽

How many slots do you want to move (from 1 to 16384)? **1000**

# 7006的节点id

What is the receiving node ID? 5c8e074636cfb8f67a44d9a55982299565b6e9e1

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:**all**

然后根据提示输入yes,在查看节点情况：

[root@localhost src]# redis-cli -c -h 192.168.11.10 -p 7001 cluster nodes

79a9bc3df0c769e3d57957c86b8420df83155afc 127.0.0.1:7003@17003 master - 0 1570178454217 3 connected 11256-16383

0c67831bea7d77f0dfb18ac8a6478d75535780f4 127.0.0.1:7005@17005 slave fd5993d2accfefc7ea5759cff4a88e62a6392323 0 1570178454720 5 connected

fd5993d2accfefc7ea5759cff4a88e62a6392323 127.0.0.1:7002@17002 master - 0 1570178454000 2 connected 5795-10922

5c8e074636cfb8f67a44d9a55982299565b6e9e1 127.0.0.1:7007@17007 master - 0 1570178455527 7 connected 0-332 5461-5794 10923-11255

d0e99032ec0cd963160c5e47dbef0b1ed5b56ad3 127.0.0.1:7004@17004 slave 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 0 1570178454000 4 connected

03e6f7b06aea39302b48cbb29f1b3209df0fde70 127.0.0.1:7006@17006 slave 79a9bc3df0c769e3d57957c86b8420df83155afc 0 1570178455727 6 connected

2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 192.168.11.10:7001@17001 myself,master - 0 1570178454000 1 connected 333-5460

可以看到节点id为5c8e074636cfb8f67a44d9a55982299565b6e9e1的7006的插槽范围为0-332，5461-5794和10923-11255。

计算节点分配的插槽 (11255 – 10923 +1 ) + (5794 – 5461 + 1) + (332+1) = 1000

二， 增加从节点

在主机(192.168.11.10)新添加一个节点7008，执行以下操作。

$ cd /usr/local/redis\_cluster

$ mkdir 7008

拷贝一份redis.conf文件到7008。

$ cp /usr/local/redis/conf/redis.conf /usr/local/redis\_cluster/7008

修改7008文件加下的redis.conf文件。

$ cd /usr/local/redis\_cluster/7008

$ vi redis.conf

修改redis.conf的以下内容。

port 7008

bind 192.168.11.10

daemonize yes

pidfile /var/run/redis\_7008.pid

# 开启集群，需要把注释#去掉

cluster-enabled yes

cluster-config-file nodes\_7008.conf

# 请求超时设置为5秒

cluster-node-timeout 5000

# aof日志开启，会每次写操作都记录一条日志

appendonly yes

1. 启动新添加的节点

在主机192.168.11.10执行以下命令。

$ redis-server /usr/local/redis\_cluster/7008/redis.conf

输入 ps -ef | grep redis命令可以看出进程为7000的Redis已经启动了。

[root@localhost 7008]# ps -ef | grep redis

root 2663 1 0 15:59 ? 00:00:04 redis-server 127.0.0.1:7001 [cluster]

root 2668 1 0 15:59 ? 00:00:04 redis-server 127.0.0.1:7002 [cluster]

root 2673 1 0 15:59 ? 00:00:04 redis-server 127.0.0.1:7003 [cluster]

root 2678 1 0 15:59 ? 00:00:04 redis-server 127.0.0.1:7004 [cluster]

root 2695 1 0 15:59 ? 00:00:04 redis-server 127.0.0.1:7006 [cluster]

root 2777 1 0 16:01 ? 00:00:04 redis-server 127.0.0.1:7005 [cluster]

root 3103 1 0 16:31 ? 00:00:02 redis-server 127.0.0.1:7007 [cluster]

root 3256 1 0 16:45 ? 00:00:00 redis-server 127.0.0.1:7008 [cluster]

root 3323 2579 0 16:50 pts/0 00:00:00 grep --color=auto redis

3，增加从节点，动态增加一个节点 nodes\_7008

$ cd /upload/redis-4.0.6/src

$ **./redis-trib.rb add-node --slave 192.168.11.10:7008 192.168.11.10:7001**

新增从节点7008。

[root@localhost src]# ./redis-trib.rb add-node --slave 192.168.11.10:7008 192.168.11.10:7001

>>> Adding node 192.168.11.10:7008 to cluster 192.168.11.10:7001

>>> Performing Cluster Check (using node 192.168.11.10:7001)

M: 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 192.168.11.10:7001

slots:333-5460 (5128 slots) master

1 additional replica(s)

M: 79a9bc3df0c769e3d57957c86b8420df83155afc 127.0.0.1:7003

slots:11256-16383 (5128 slots) master

1 additional replica(s)

S: 0c67831bea7d77f0dfb18ac8a6478d75535780f4 127.0.0.1:7005

slots: (0 slots) slave

replicates fd5993d2accfefc7ea5759cff4a88e62a6392323

M: fd5993d2accfefc7ea5759cff4a88e62a6392323 127.0.0.1:7002

slots:5795-10922 (5128 slots) master

1 additional replica(s)

M: 5c8e074636cfb8f67a44d9a55982299565b6e9e1 127.0.0.1:7007

slots:0-332,5461-5794,10923-11255 (1000 slots) master

0 additional replica(s)

S: d0e99032ec0cd963160c5e47dbef0b1ed5b56ad3 127.0.0.1:7004

slots: (0 slots) slave

replicates 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a

S: 03e6f7b06aea39302b48cbb29f1b3209df0fde70 127.0.0.1:7006

slots: (0 slots) slave

replicates 79a9bc3df0c769e3d57957c86b8420df83155afc

[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 192.168.11.10:7008 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.

查看集群的Redis节点状态。

[root@localhost src]# redis-cli -c -p 7001 cluster nodes

79a9bc3df0c769e3d57957c86b8420df83155afc 127.0.0.1:7003@17003 master - 0 1570179131000 3 connected 11256-16383

0c67831bea7d77f0dfb18ac8a6478d75535780f4 127.0.0.1:7005@17005 slave fd5993d2accfefc7ea5759cff4a88e62a6392323 0 1570179132584 5 connected

fd5993d2accfefc7ea5759cff4a88e62a6392323 127.0.0.1:7002@17002 master - 0 1570179132584 2 connected 5795-10922

5c8e074636cfb8f67a44d9a55982299565b6e9e1 127.0.0.1:7007@17007 master - 0 1570179131577 7 connected 0-332 5461-5794 10923-11255

d0e99032ec0cd963160c5e47dbef0b1ed5b56ad3 127.0.0.1:7004@17004 slave 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 0 1570179132000 4 connected

9da2d0405a50f8609464b7f3f7f4e4653dfec8a6 127.0.0.1:7008@17008 slave 5c8e074636cfb8f67a44d9a55982299565b6e9e1 0 1570179132584 7 connected

03e6f7b06aea39302b48cbb29f1b3209df0fde70 127.0.0.1:7006@17006 slave 79a9bc3df0c769e3d57957c86b8420df83155afc 0 1570179131577 6 connected

2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 192.168.11.10:7001@17001 myself,master - 0 1570179132000 1 connected 333-5460

然后使用cluster replicat将当前节点设置为 node\_id 指定的节点的从节点。将IP地址为192.168.11.10，端口为7008的Redis的节点设置为7007节点” **5c8e074636cfb8f67a44d9a55982299565b6e9e1**”的从节点。

[root@localhost 7007]# redis-cli -c -h 192.168.11.10 -p 7008

192.168.11.11:7008> cluster replicate **5c8e074636cfb8f67a44d9a55982299565b6e9e1**

OK

再查看集群节点状态。

192.168.11.10:7008> cluster nodes

9da2d0405a50f8609464b7f3f7f4e4653dfec8a6 127.0.0.1:7008@17008 myself,slave 5c8e074636cfb8f67a44d9a55982299565b6e9e1 0 1570179252000 0 connected

fd5993d2accfefc7ea5759cff4a88e62a6392323 127.0.0.1:7002@17002 master - 0 1570179252000 2 connected 5795-10922

79a9bc3df0c769e3d57957c86b8420df83155afc 127.0.0.1:7003@17003 master - 0 1570179252572 3 connected 11256-16383

0c67831bea7d77f0dfb18ac8a6478d75535780f4 127.0.0.1:7005@17005 slave fd5993d2accfefc7ea5759cff4a88e62a6392323 0 1570179251566 2 connected

5c8e074636cfb8f67a44d9a55982299565b6e9e1 127.0.0.1:7007@17007 master - 0 1570179252572 7 connected 0-332 5461-5794 10923-11255

03e6f7b06aea39302b48cbb29f1b3209df0fde70 127.0.0.1:7006@17006 slave 79a9bc3df0c769e3d57957c86b8420df83155afc 0 1570179252000 3 connected

d0e99032ec0cd963160c5e47dbef0b1ed5b56ad3 127.0.0.1:7004@17004 slave 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 0 1570179253578 1 connected

2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 127.0.0.1:7001@17001 master - 0 1570179253075 1 connected 333-5460

## 删除集群节点

1. 删除从节点(slave)

删除从节点7008。

$ ./redis-trib.rb del-node 192.168.11.10:7008 '9da2d0405a50f8609464b7f3f7f4e4653dfec8a6'



1. 删除主节点(master)

1） 将这个节点上的所有插槽转移到其他节点上；

假设我们想要删除7007这个主节点，执行脚本：

$ ./redis-trib.rb reshard 192.168.11.10:7007

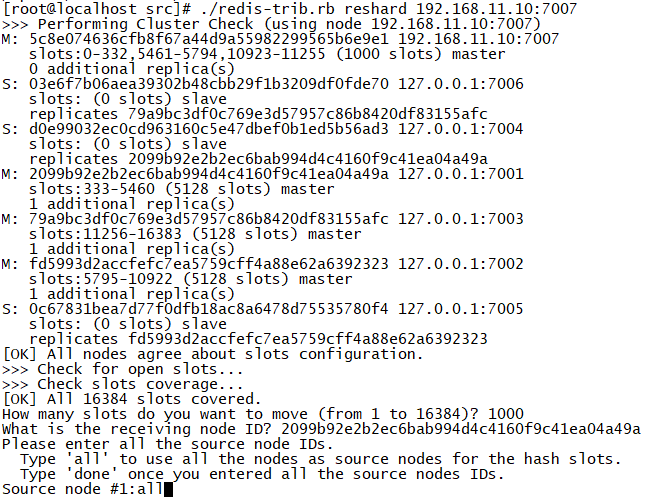


图 6‑31 Redis集群删除主节点

选择需要转移的插槽的数量，因为7007有1000个插槽，所以输入1000, 把插糟转移到7001节点，输入7001节点的id 2099b92e2b2ec6bab994d4c4160f9c41ea04a49ae ， 输入done，开始转移。

查看Redis集群信息，可以看到7007节点已经没有插槽了。

[root@localhost src]# redis-cli -c -h 192.168.11.10 -p 7001 cluster nodes

79a9bc3df0c769e3d57957c86b8420df83155afc 127.0.0.1:7003@17003 master - 0 1570186468609 3 connected

0c67831bea7d77f0dfb18ac8a6478d75535780f4 127.0.0.1:7005@17005 slave 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 0 1570186469615 8 connected

fd5993d2accfefc7ea5759cff4a88e62a6392323 127.0.0.1:7002@17002 master - 0 1570186468508 2 connected

5c8e074636cfb8f67a44d9a55982299565b6e9e1 127.0.0.1:**7007**@17007 master - 0 1570186470520 7 connected

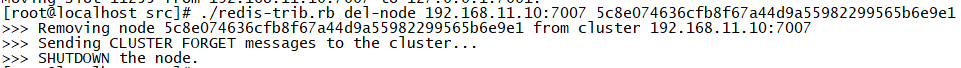
d0e99032ec0cd963160c5e47dbef0b1ed5b56ad3 127.0.0.1:7004@17004 slave 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 0 1570186469514 8 connected

03e6f7b06aea39302b48cbb29f1b3209df0fde70 127.0.0.1:7006@17006 slave 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 0 1570186469111 8 connected

2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 192.168.11.10:7001@17001 myself,master - 0 1570186469000 8 connected 0-16383

在使用redis-trib.rb删除节点，节点id 为5c8e074636cfb8f67a44d9a55982299565b6e9e1。

./redis-trib.rb del-node 192.168.11.10:7007 5c8e074636cfb8f67a44d9a55982299565b6e9e1



查看集群信息，可以看到已经没有7007这个节点了。

[root@localhost src]# redis-cli -c -h 192.168.11.10 -p 7001 cluster nodes

79a9bc3df0c769e3d57957c86b8420df83155afc 127.0.0.1:7003@17003 master - 0 1570186570546 3 connected

0c67831bea7d77f0dfb18ac8a6478d75535780f4 127.0.0.1:7005@17005 slave 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 0 1570186570000 8 connected

fd5993d2accfefc7ea5759cff4a88e62a6392323 127.0.0.1:7002@17002 master - 0 1570186570000 2 connected

d0e99032ec0cd963160c5e47dbef0b1ed5b56ad3 127.0.0.1:7004@17004 slave 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 0 1570186570747 8 connected

03e6f7b06aea39302b48cbb29f1b3209df0fde70 127.0.0.1:7006@17006 slave 2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 0 1570186570000 8 connected

2099b92e2b2ec6bab994d4c4160f9c41ea04a49a 192.168.11.10:7001@17001 myself,master - 0 1570186567000 8 connected 0-16383