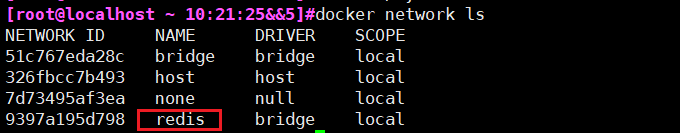
增加节点和删除节点会涉及到已有的槽位重新分配及数据迁移

# **创建网卡**

# **创建网卡**

|  |
| --- |
| # docker network create redis --subnet 172.38.0.0/16  # docker network ls |



# **创建redis配置**

# 通过脚本创建8个redis配置，其中2个备用扩容用。

|  |
| --- |
| for port in $(seq 1 8); \ do \ mkdir -p /mydata/redis/node-${port}/conf touch /mydata/redis/node-${port}/conf/redis.conf cat  << EOF >> /mydata/redis/node-${port}/conf/redis.conf port 6379  bind 0.0.0.0 cluster-enabled yes  cluster-config-file nodes.conf cluster-node-timeout 5000 cluster-announce-ip 172.38.0.1${port} cluster-announce-port 6379 cluster-announce-bus-port 16379 appendonly yes EOF done |

# **启动8个redis服务**

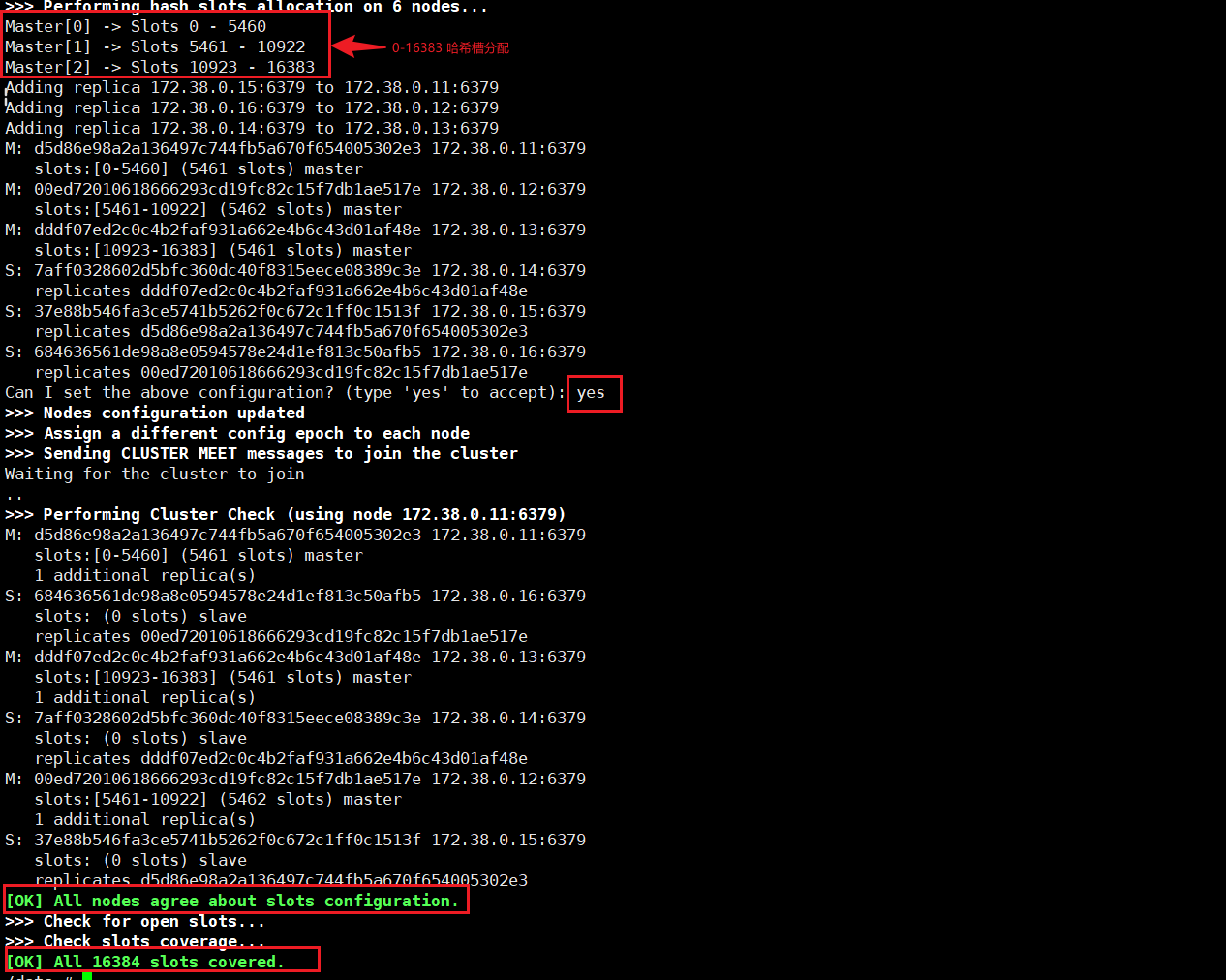
|  |
| --- |
| for port in $(seq 1 8); \ do \ docker run -p 637${port}:6379 -p 1637${port}:16379 \ --name redis-${port} \ --privileged=true \ -v /mydata/redis/node-${port}/data:/data \ -v /mydata/redis/node-${port}/conf/redis.conf:/etc/redis/redis.conf \ -d --net redis --ip 172.38.0.1${port} redis:5.0.9-alpine3.11 redis-server /etc/redis/redis.conf done |

# **创建3主3从，redis集群扩缩容配置案例**

**4.1自动分配**

# docker exec -it redis-1 /bin/sh

/data # ls  
appendonly.aof  nodes.conf  
/data # redis-cli --cluster create 172.38.0.11:6379 172.38.0.12:6379 172.38.0.13:6379 172.38.0.14:6379 172.38.0.15:6379 172.38.0.16:6379 --cluster-replicas 1



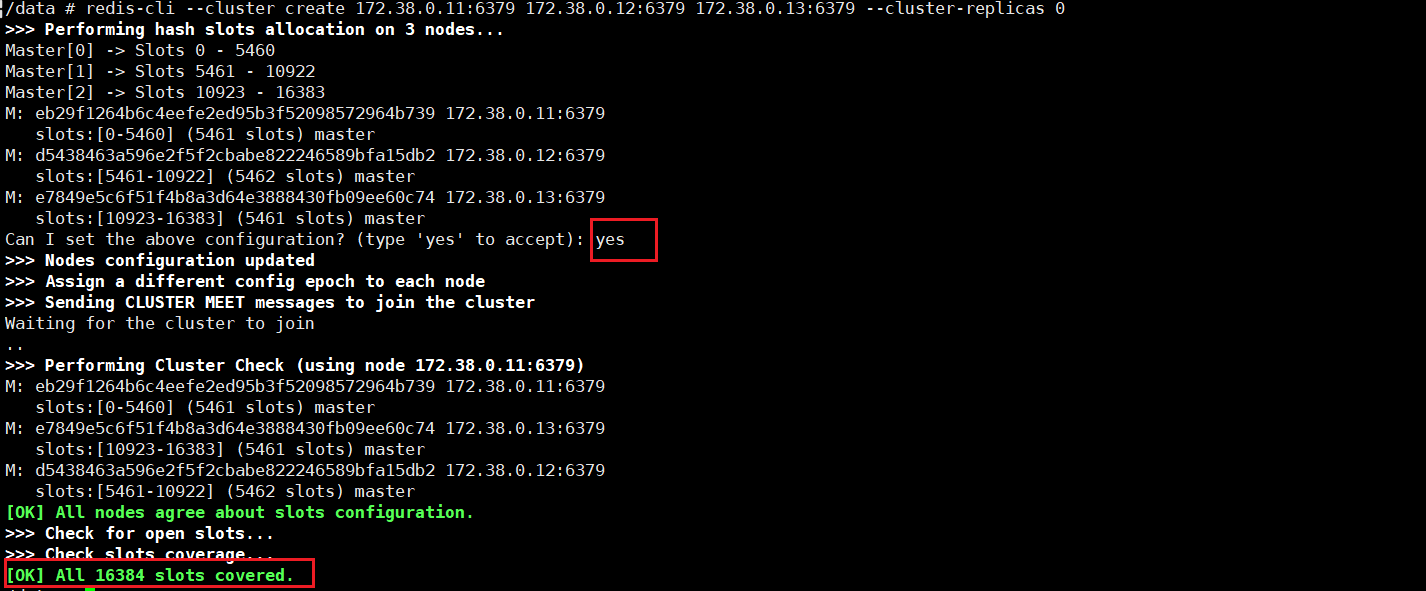
**4.2 自定义**

有时候，当我们只有3台物理节点时，就想要指定主从关系，从而保证高可用，而指定主从关系，就需要手动操作了

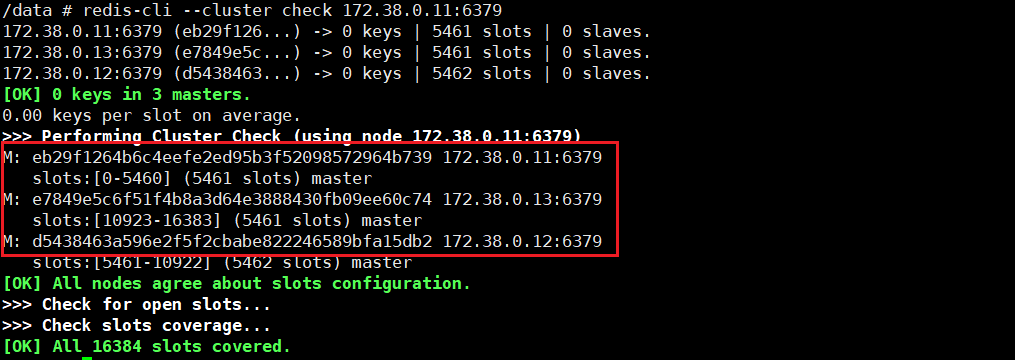
#docker exec -it redis-1 sh

**使用以下命令创建主节点**

/data # redis-cli --cluster create 172.38.0.11:6379 172.38.0.12:6379 172.38.0.13:6379 --cluster-replicas 0

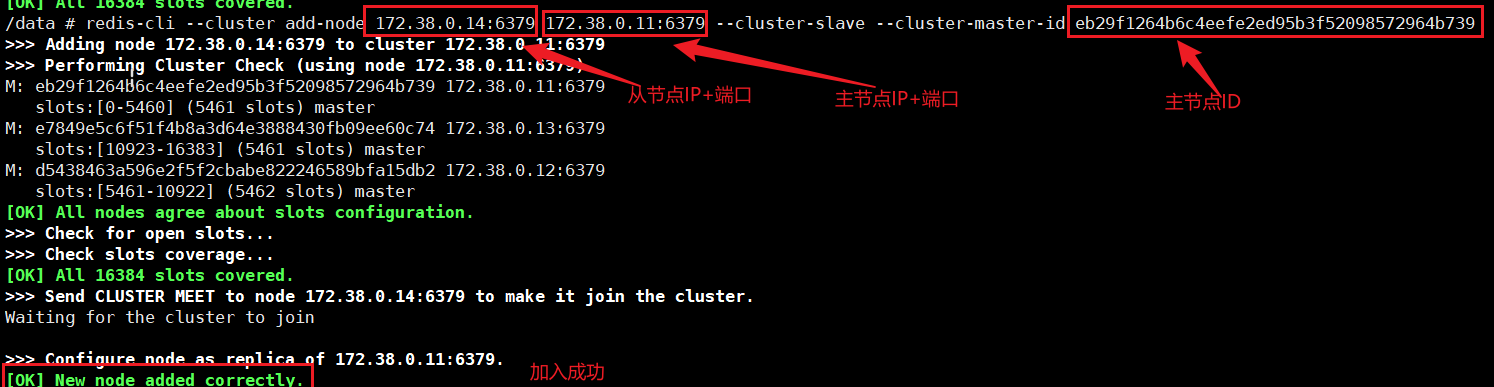


/data # redis-cli --cluster check 172.38.0.11:6379



**依次增加从节点**

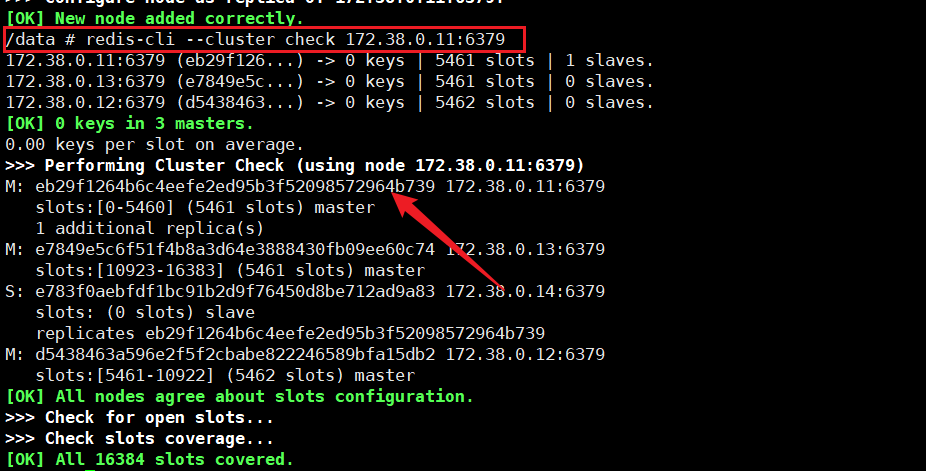
/data # redis-cli --cluster add-node 172.38.0.14:6379 172.38.0.11:6379 --cluster-slave --cluster-master-id eb29f1264b6c4eefe2ed95b3f52098572964b739



* 172.38.0.14:6379 要添加的从节点
* 172.38.0.11:6379 要添加到哪一个主节点。
* --cluster-slave 表示要添加从节点
* --cluster-master-id 要添加到哪一个主节点，id是\*\*\*\*\*

ok，这样添加完后得到的就是指定的想要的节点架构。

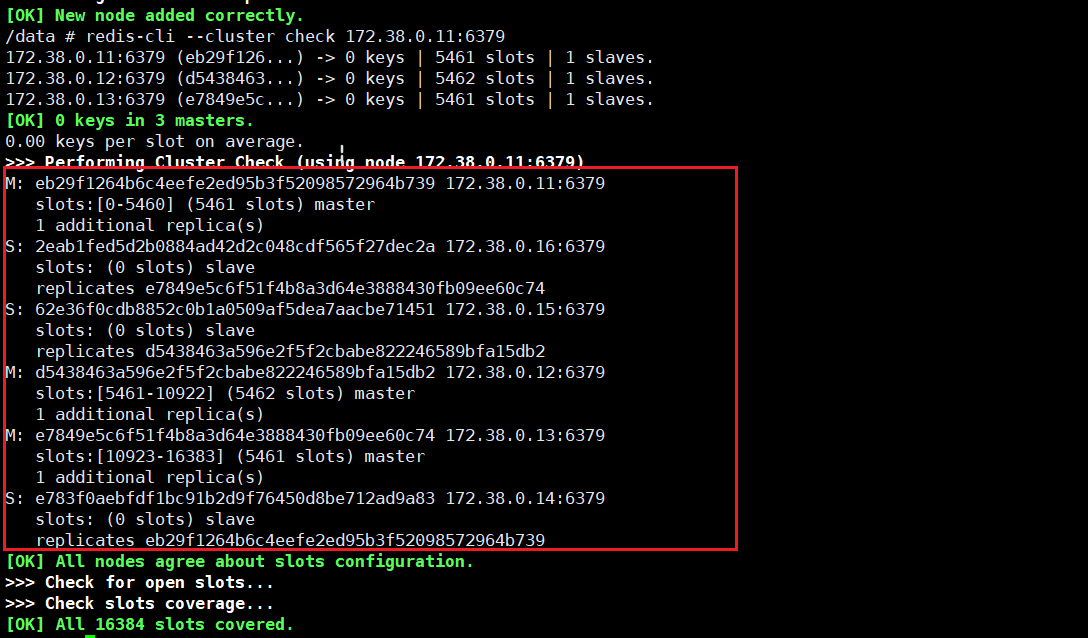
/data # redis-cli --cluster check 172.38.0.11:6379



剩下两个也是如此

# redis-cli --cluster add-node 172.38.0.15:6379 172.38.0.12:6379 --cluster-slave --cluster-master-id d5438463a596e2f5f2cbabe822246589bfa15db2

# redis-cli --cluster add-node 172.38.0.16:6379 172.38.0.13:6379 --cluster-slave --cluster-master-id e7849e5c6f51f4b8a3d64e3888430fb09ee60c74



最终redis-4,5,6 分别为redis-1,2,3的从节点。

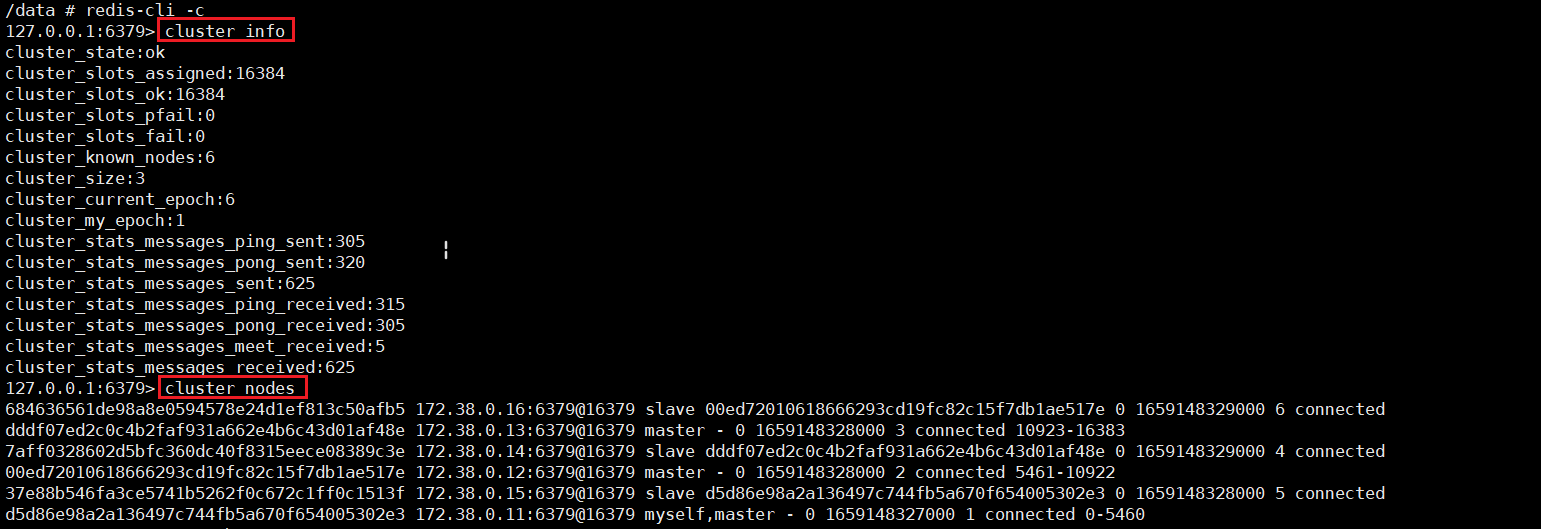
<https://blog.csdn.net/guotianqing/article/details/119778684>

**4.3 集群查看验证**

/data # redis-cli -c # -c 连接集群

127.0.0.1:6379> cluster info # 查看集群信息

127.0.0.1:6379> cluster nodes # 查看集群节点信息



127.0.0.1:6379> set a b # 这里的值被设置到 redis-3

-> Redirected to slot [15495] located at 172.38.0.13:6379

OK

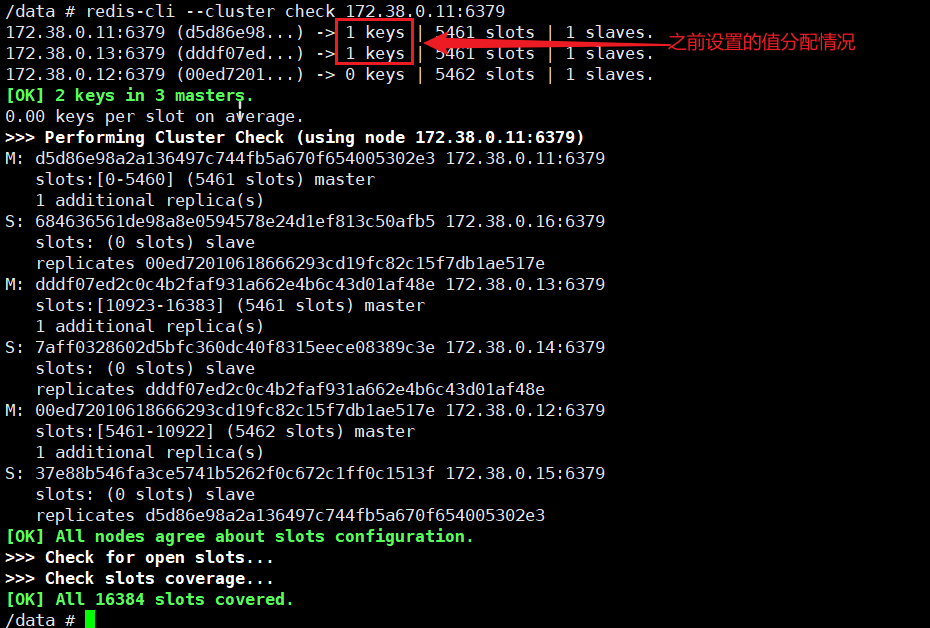
172.38.0.13:6379> set b c # 这里的值被设置到 redis-1

-> Redirected to slot [3300] located at 172.38.0.11:6379

OK

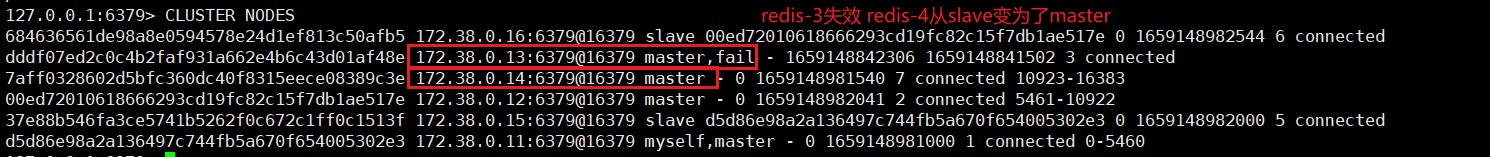


/data # redis-cli --cluster check 172.38.0.11:6379 # 进行集群的检查

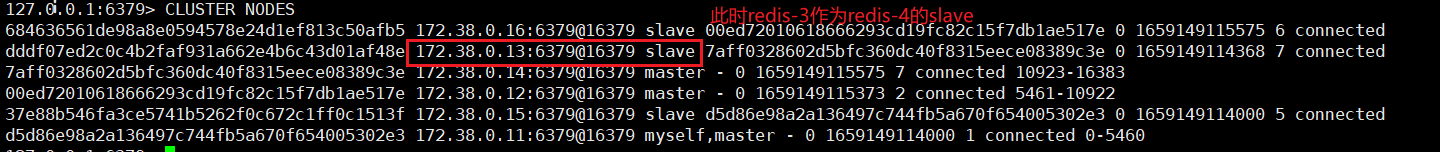


# **容错切换迁移**

# docker stop redis-3  # 停止掉redis  
redis-3  
127.0.0.1:6379> get a  # 从新获取， 从redis-4中获取到了，redis-3对应的从机redis-4成功上位，成为master。  
-> Redirected to slot [15495] located at 172.38.0.14:6379  
"b"



# docker start redis-3  
redis-3  
# 重启redis-3,此时redis-3仍为slave  
# 如果想还原之前的三主三从，先停掉redis-4、在启动redis-4



# **主从扩容**

准备将redis-7、redis-8加入集群，并且redis-8作为slave挂在redis-7下。槽位该如何分配?

，

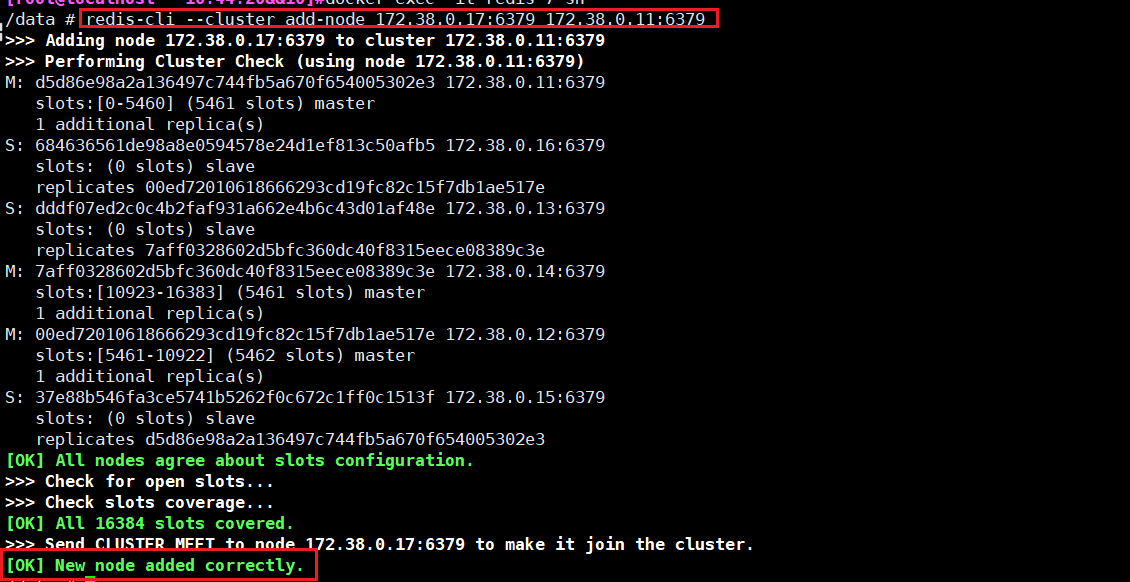
新建2个节点

进入redis-7容器内部，将redis-7节点作为master节点加入到集群中

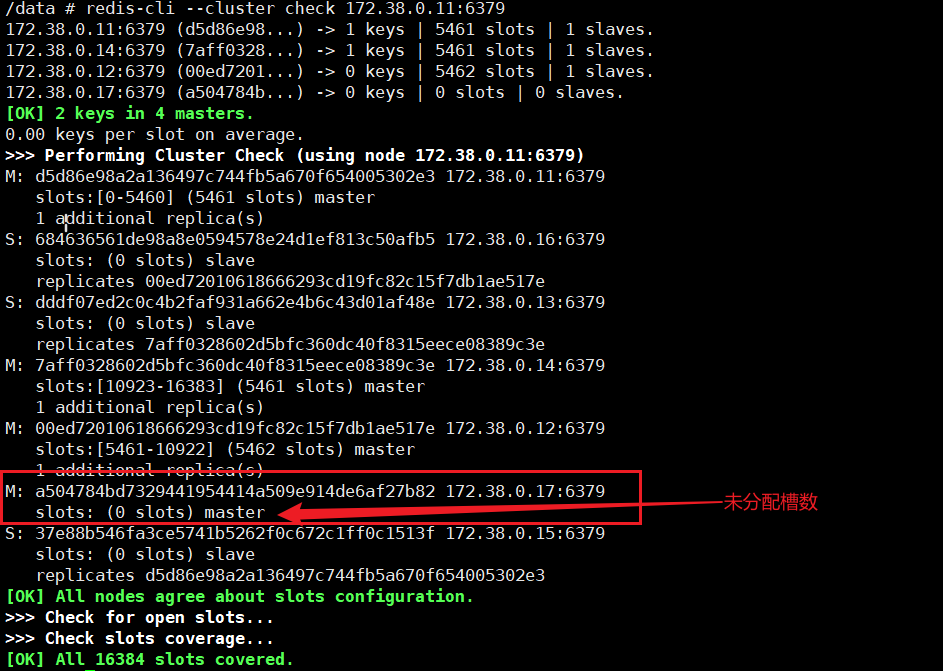
172.38.0.11:6379

原来节点里面的引路人

/data # redis-cli --cluster add-node 172.38.0.17:6379 172.38.0.11:6379

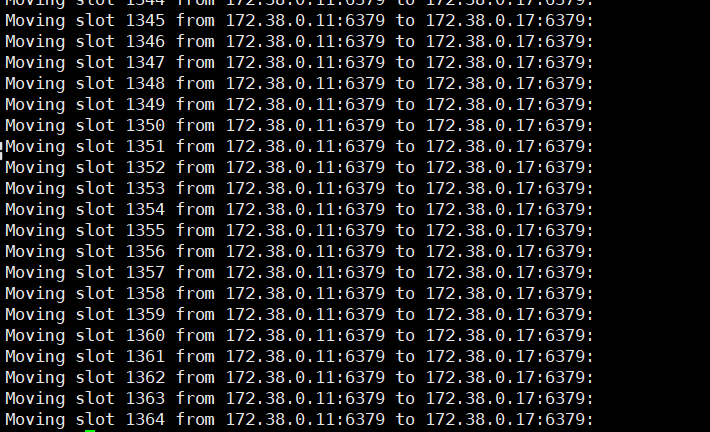


/data # redis-cli --cluster check 172.38.0.11:6379



重新分配哈希槽

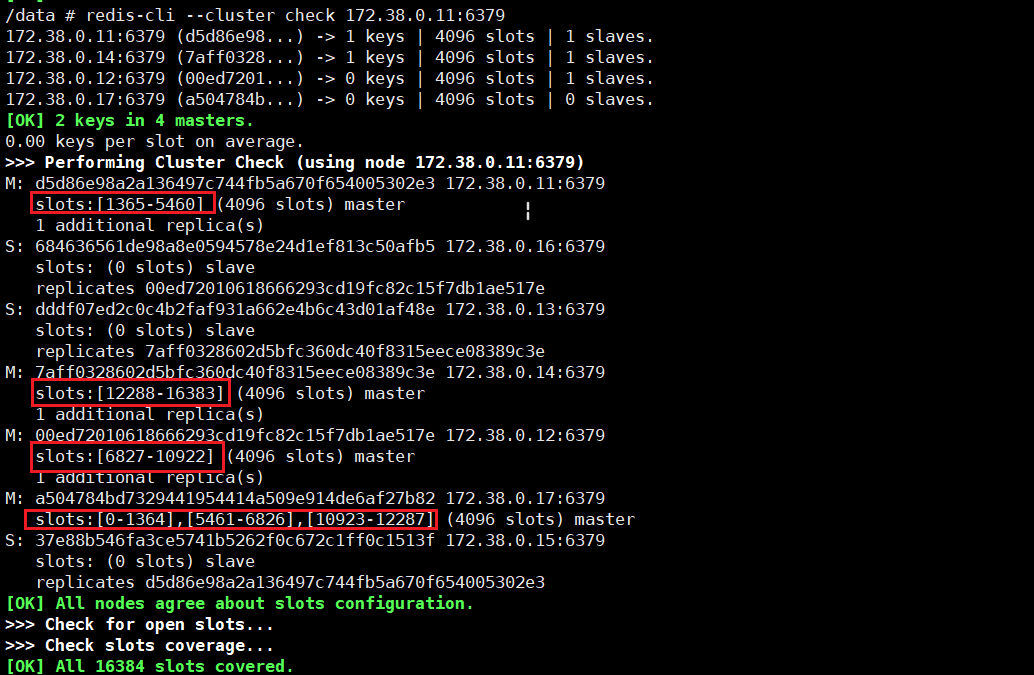
|  |
| --- |
| /data # redis-cli --cluster reshard 172.38.0.11:6379  # 平均分配槽数：16384/4 = 4096 #4为master的节点数  How many slots do you want to move(from 1 to 16384)? 4096  ...  # 输入需要分配节点的ID，即redis-7的ID  What is the receiving node ID? a504784bd7329441954414a509e914de6af27b82  ...  # 输入all  Source node #1: all  ...  # 输入yes  Do you want to proceed with the proposed reshard plan(yes/no)? yes |



槽号分派说明：为什么redis-7是三个新的区间？

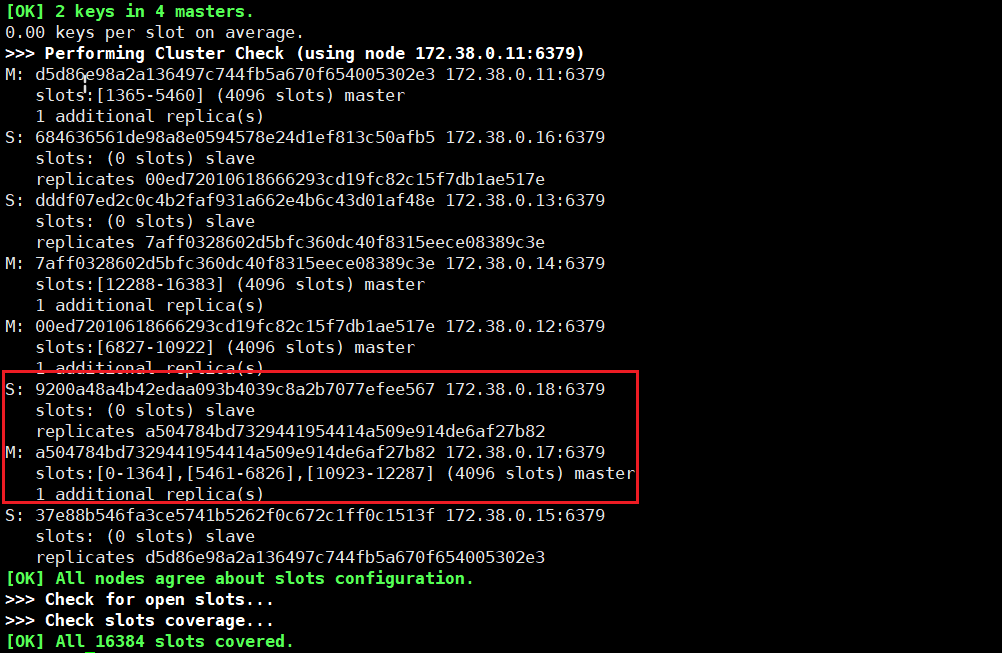
重新分配成本太高，所以前三个各自匀出来一部分，分给新的节点。

，并非平均分配。



为主节点分配从节点redis-8

/data # redis-cli --cluster add-node 172.38.0.18:6379 172.38.0.17:6379 --cluster-slave --cluster-master-id a504784bd7329441954414a509e914de6af27b82



# **主从缩容**

停掉redis-7和redis-8,恢复3主3从，

思考：

槽位该重新分配？

原路返回？还是都分配给某一台机器？

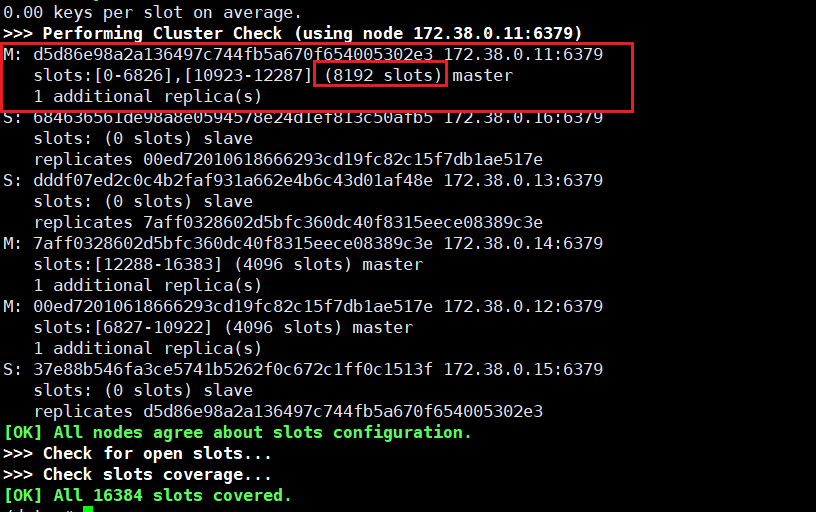
手动根据分配给指定的节点。本案例全部分给redis-1

先删除redis-7 还是redis-8?

**方式一**

|  |
| --- |
| 1、先删除slave redis-8 ,master可能在进行读写操作  # 命令：redis-cli --cluster del-node ip:端口 节点id  /data # redis-cli --cluster del-node 172.38.0.18:6379 9200a48a4b42edaa093b4039c8a2b7077efee567    2、将redis-7的槽号清空，重新分配，将清出来的槽号都分给redis-1  **2.1 重新分配槽位（可以多次分配给不同的master节点，这里为了方便演示，一次性分配给了redis-1）**  /data # redis-cli --cluster reshard 172.38.0.11:6379  ...  **2.2 需要移动的槽位数：4096**  How many slots do you want to move (from 1 to 16384)? 4096  **2.3 接收槽位的节点id. redis-1**  What is the receiving node ID? d5d86e98a2a136497c744fb5a670f654005302e3  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.  **3.4 redis-7的节点id, 告知需要移除那个**  Source node #1: a504784bd7329441954414a509e914de6af27b82  Source node #2: done  ..  # yes  Do you want to proceed with the proposed reshard plan (yes/no)? yes  **2.5 删除redis-7节点**  /data # redis-cli --cluster del-node 172.38.0.17:6379 a504784bd7329441954414a509e914de6af27b82 |

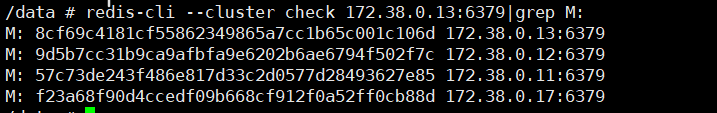
分配后redis-7的槽位变为0，4096个槽位都分配给了redis-1, 它变成了8192.



**方式二**

1、先删除slave redis-8 ,master可能在进行读写操作

2、将redis-7的槽号清空，重新分配，将清出来的槽号平均分配给redis-1 redis2 redis-3

redis-cli --cluster reshard --cluster-from f23a68f90d4ccedf09b668cf912f0a52ff0cb88d --cluster-to 57c73de243f486e817d33c2d0577d28493627e85 --cluster-slots 1365 172.38.0.11:6379

redis-cli --cluster reshard --cluster-from f23a68f90d4ccedf09b668cf912f0a52ff0cb88d --cluster-to 9d5b7cc31b9ca9afbfa9e6202b6ae6794f502f7c --cluster-slots 1365 172.38.0.11:6379

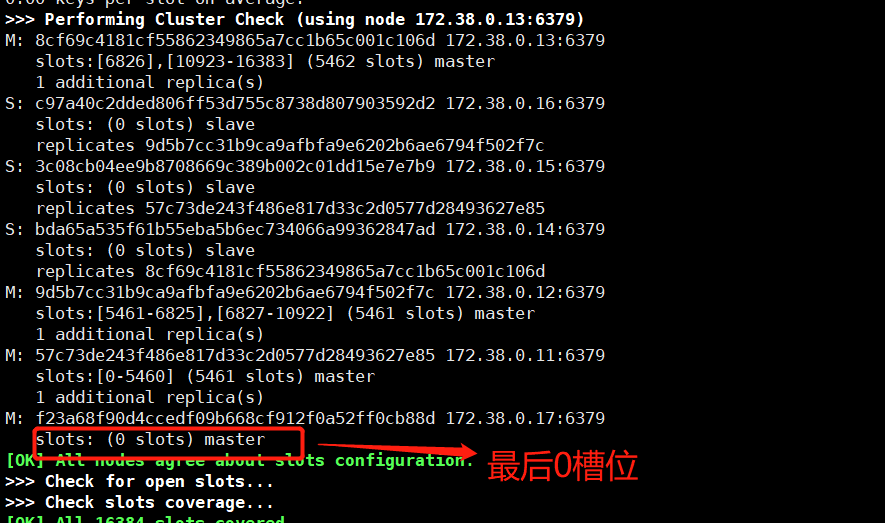
redis-cli --cluster reshard --cluster-from f23a68f90d4ccedf09b668cf912f0a52ff0cb88d --cluster-to 8cf69c4181cf55862349865a7cc1b65c001c106d --cluster-slots 1366 172.38.0.11:6379

--cluster-from：要删除的主节点id

--cluster-to：接收迁移槽的主节点id

--cluster-slots：迁移槽的数量

ip:端口：集群任意1个节点即可



**移除节点**

redis-cli --cluster del-node 172.38.0.11:6379 f23a68f90d4ccedf09b668cf912f0a52ff0cb88d

172.38.0.11:6379为集群中的任意一个节点即可

# **参考URL**

<https://www.modb.pro/db/325603>