redis集群搭建步骤

```
#准备redis安装包，放到/usrl/local下 解压
tar -zxvf redis-6.0.9.tar.gz
cd redis-6.0.9
#编译并安装
make
make install
#在/usrl/local/etc下创建redis集群节点
mkdir -p redis/cluster/7001
mkdir -p redis/cluster/7002
mkdir -p redis/cluster/7003
mkdir -p redis/cluster/7004
mkdir -p redis/cluster/7005
mkdir -p redis/cluster/7006
#将redis-6.0.9文件下redis.conf文件复制到各个节点下
cd /usr/local/etc/redis/cluster/7001
cp /usr/local/redis/redis.conf
#其他节点一样
#修改配置文件
vim redis.conf
port 7001
daemonize yes
cluster-enabled yes
cluster-config-file nodes-7001.conf
cluster-node-timeout 15000
appendonly yes
pidfile /var/run/redis_7001.pid
logfile "/usr/local/etc/redis/cluster/7006/7006-log"
appendfilename "appendonly-7006.aof"
#保存，并同步修改其他节点
#分别进入六台主机后启动redis节点
cd /usr/local/etc/redis/cluster/7001
redis-server redis.conf
...
```

```
[wujinhua@wujinhua 7006 % ps -ef|grep redis
  501 66581     1   0  2:55下午 ??         0:00.37 redis-server *:7001 [cluster]
  501 66583     1   0  2:55下午 ??         0:00.33 redis-server *:7002 [cluster]
  501 66585     1   0  2:55下午 ??         0:00.30 redis-server *:7003 [cluster]
  501 66590     1   0  2:55下午 ??         0:00.28 redis-server *:7004 [cluster]
  501 66592     1   0  2:56下午 ??         0:00.24 redis-server *:7005 [cluster]
  501 66612     1   0  2:56下午 ??         0:00.08 redis-server *:7006 [cluster]
  501 66618 60905   0  2:56下午 ttys002    0:00.00 grep redis
```

```
#创建集群
redis-cli --cluster create 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 --cluster-replicas 1
```

```
[wujinhua@wujinhua 7006 % redis-cli --cluster create 127.0.0.1:7001 127.0
r-replicas 1
>>> Performing hash slots allocation on 6 nodes...
Master[0] -> Slots 0 - 5460
Master[1] -> Slots 5461 - 10922
Master[2] -> Slots 10923 - 16383
Adding replica 127.0.0.1:7005 to 127.0.0.1:7001
Adding replica 127.0.0.1:7006 to 127.0.0.1:7002
Adding replica 127.0.0.1:7004 to 127.0.0.1:7003
>>> Trying to optimize slaves allocation for anti-affinity
[WARNING] Some slaves are in the same host as their master
M: 79b9c449b96ed7cdd5eb7e386cdc2b64c695f0d2 127.0.0.1:7001
   slots:[0-5460] (5461 slots) master
M: 745059e48faa038aeafb594ab2eee71f0eff4730 127.0.0.1:7002
   slots:[5461-10922] (5462 slots) master
M: 6d492f22a041007e5a2c8faa9c5f3efca0094f35 127.0.0.1:7003
   slots:[10923-16383] (5461 slots) master
S: dde556fdfb9fb8382ce02f7742a709efe88b5987 127.0.0.1:7004
   replicates 6d492f22a041007e5a2c8faa9c5f3efca0094f35
S: 722846bf4ca79dc25352710ebd2924454ff7e541 127.0.0.1:7005
   replicates 79b9c449b96ed7cdd5eb7e386cdc2b64c695f0d2
S: b841ae391e652bd42d604e9e8a31d8c3076e7de9 127.0.0.1:7006
   replicates 745059e48faa038aeafb594ab2eee71f0eff4730
Can I set the above configuration? (type 'yes' to accept):
```

```
>>> Performing Cluster Check (using node 127.0.0.1:7001)
M: 79b9c449b96ed7cdd5eb7e386cdc2b64c695f0d2 127.0.0.1:7001
   slots:[0-5460] (5461 slots) master
   1 additional replica(s)
M: 745059e48faa038aeafb594ab2eee71f0eff4730 127.0.0.1:7002
   slots:[5461-10922] (5462 slots) master
   1 additional replica(s)
S: 722846bf4ca79dc25352710ebd2924454ff7e541 127.0.0.1:7005
   slots: (0 slots) slave
   replicates 79b9c449b96ed7cdd5eb7e386cdc2b64c695f0d2
M: 6d492f22a041007e5a2c8faa9c5f3efca0094f35 127.0.0.1:7003
   slots:[10923-16383] (5461 slots) master
   1 additional replica(s)
S: dde556fdfb9fb8382ce02f7742a709efe88b5987 127.0.0.1:7004
   slots: (0 slots) slave
   replicates 6d492f22a041007e5a2c8faa9c5f3efca0094f35
S: b841ae391e652bd42d604e9e8a31d8c3076e7de9 127.0.0.1:7006
   slots: (0 slots) slave
   replicates 745059e48faa038aeafb594ab2eee71f0eff4730
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
wujinhua@wujinhua 7006 %
```

```
#测试集群，查看集群信息

redis-cli -h 127.0.0.1 -p 7001 cluster info
```

```
wujinhua@wujinhua 7006 % redis-cli -h 127.0.0.1 -p 7001 cluster info
cluster_state:ok
cluster_slots_assigned:16384
cluster_slots_ok:16384
cluster_slots_pfail:0
cluster_slots_fail:0
cluster_known_nodes:6
cluster_size:3
cluster_current_epoch:6
cluster_my_epoch:1
cluster_stats_messages_ping_sent:273
cluster_stats_messages_pong_sent:270
cluster_stats_messages_sent:543
cluster_stats_messages_ping_received:265
cluster_stats_messages_pong_received:273
cluster_stats_messages_meet_received:5
cluster_stats_messages_received:543
wujinhua@wujinhua 7006 %
wujinhua@wujinhua 7006 %
wujinhua@wujinhua 7006 %
```

```
#进入一个节点
redis-cli -h 127.0.0.1 -p 7001 -c   #加“-c”参数，节点之间就可以互相跳转
CLUSTER SLOTS #查看节点的哈希槽编号范围
```

```
[wujinhua@wujinhua 7006 % redis-cli -h 127.0.0.1 -p 7001 -c
[127.0.0.1:7001> CLUSTER SLOTS
1) 1) (integer) 5461
   2) (integer) 10922
   3) 1) "127.0.0.1"
      2) (integer) 7002
      3) "745059e48faa038aeafb594ab2eee71f0eff4730"
   4) 1) "127.0.0.1"
      2) (integer) 7006
      3) "b841ae391e652bd42d604e9e8a31d8c3076e7de9"
2) 1) (integer) 10923
   2) (integer) 16383
   3) 1) "127.0.0.1"
      2) (integer) 7003
      3) "6d492f22a041007e5a2c8faa9c5f3efca0094f35"
   4) 1) "127.0.0.1"
      2) (integer) 7004
      3) "dde556fdfb9fb8382ce02f7742a709efe88b5987"
3) 1) (integer) 0
   2) (integer) 5460
   3) 1) "127.0.0.1"
      2) (integer) 7001
      3) "79b9c449b96ed7cdd5eb7e386cdc2b64c695f0d2"
   4) 1) "127.0.0.1"
      2) (integer) 7005
      3) "722846bf4ca79dc25352710ebd2924454ff7e541"
127.0.0.1:7001>
```

```
#创建key
127.0.0.1:7003> set k1 "hello word!"
OK
127.0.0.1:7003> CLUSTER KEYSLOT k1 #查看k1键的槽编号，注意，可以看到这里自动切换到7003了
(integer) 12706
#这条数据的槽编号对应为12706,对应在 7003和7004两台主机上
127.0.0.1:7003>
wujinhua@wujinhua 7006 % redis-cli -h 127.0.0.1 -p 7004 -c
127.0.0.1:7004> keys *
1) "k1"
127.0.0.1:7004> get k1
-> Redirected to slot [12706] located at 127.0.0.1:7003
"hello word!"
127.0.0.1:7003>
wujinhua@wujinhua 7006 % redis-cli -h 127.0.0.1 -p 7005 -c
127.0.0.1:7005> keys *
(empty array)
127.0.0.1:7005>
```

```
19 #7005主机上为空
```

```
[127.0.0.1:7001> set k1 hello word!
-> Redirected to slot [12706] located at 127.0.0.1:7003
(error) ERR syntax error
[127.0.0.1:7003> set k1 "hello word!"
OK
[127.0.0.1:7003> CLUSTER KEYSLOT k1
(integer) 12706
[127.0.0.1:7003>
[wujinhua@wujinhua 7006 % redis-cli -h 127.0.0.1 -p 7004 -c
[127.0.0.1:7004> keys *
1) "k1"
[127.0.0.1:7004> get k1
-> Redirected to slot [12706] located at 127.0.0.1:7003
"hello word!"
[127.0.0.1:7003>
[wujinhua@wujinhua 7006 % redis-cli -h 127.0.0.1 -p 7005 -c
[127.0.0.1:7005> keys *
(empty array)
127.0.0.1:7005>
```