# redis4.0.1集群安装

1. **两台服务器先设置hosts**

192.168.137.200 node1

192.168.137.124 node2

1. 下载redis-4.0.1并解压

cd /usr/local/

wget http://download.redis.io/releases/redis-4.0.1.tar.gz

tar -zxvf redis-4.0.1.tar.gz

建立软连接：ln -s redis-4.0.1 redis

1. **编译安装**

cd /usr/local/redis

centos:yum -y install gcc

ubuntu:apt-get -y install gcc

make && make install

如果报错则 make MALLOC=libc

1. **创建配制节点**

每台机器配置三个节点

mkdir -p /usr/local/redis/redis\_cluster/7000

mkdir -p /usr/local/redis/redis\_cluster/7001

mkdir -p /usr/local/redis/redis\_cluster/7002

cp /usr/local/redis/redis.conf /usr/local/redis/redis\_cluster/7000

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

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

1. **修改redis.conf进行基础配置**

#基础配置

port 7000 #端口根据对应的文件夹去配制端口

bind 本机ip #根据本机所在的IP

daemonize yes #redis后台运行

pidfile /var/run/redis\_7000.pid #pidfile文件对应

cluster-enabled yes #开启集群 把注释#去掉

cluster-config-file nodes\_7000.conf #集群的配置 配置文件首次启动自动生成 cluster-node-timeout 15000 #请求超时 默认15秒，可自行设置

#淘汰策略

#默认maxmemory-policy volatile-lru

volatile-lru  #在设置了过期时间的键中，优先移除最近未使用的key

volatile-ttl  #在设置了过期时间的键中，优先移除将要过期的key

volatile-random   #在设置了过期时间的键中，随机移除某个key

allkeys-lru  #在所有键中，优先移除最近未使用的key

allkeys-random  #在所有键中，随机移除某个key

noeviction  #当内存使用达到阈值的时候，所有引起申请内存的命令会报错

#持久化配置，需要持久化可以配置

#AOF方式

dir "/data/dbs/redis/6381" #AOF文件存放目录

appendonly yes #开启AOF持久化，默认关闭

appendfilename "appendonly.aof" #AOF文件名称（默认）

appendfsync no #AOF持久化策略，always、everysec和no

auto-aof-rewrite-percentage 100 #触发AOF文件重写的条件（默认）auto-aof-rewrite-min-size 64mb #触发AOF文件重写的条件（默认）

#RDB方式

save 900 1 #每900秒钟里redis数据库有一条数据被修改则触发RDB

dbfilename "dump.rdb" #持久化文件名称

dir "/data/dbs/redis/6381" #持久化数据文件存放的路径

1. **启动节点**

/usr/local/redis/src/redis-server /usr/local/redis/redis\_cluster/7000/redis.conf

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

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

1. **检查各节点是否启动**

查看进程

ps -ef | grep redis

查看端口

netstat -tnlp | grep redis

1. **其它服务器做同样配置**
2. **安装ruby环境**

选择其中一台安装就行

centos:

yum install -y ruby

yum install -y rubygems

ubuntu：

apt-get install -y ruby

apt-get install -y rubygems

1. **安装ruby的包**

wget https://rubygems.org/downloads/redis-4.0.0.rc1.gem

gem install -l ./redis-4.0.0.rc1.gem

1. **创建集群**

安装了ruby的服务器进入/usr/local/redis/src目录下,执行以下脚本

cd /usr/local/redis/src

./redis-trib.rb create --replicas 1 node1:7000 node1:7001 node1:7002 node2:7003 node2:7004 node2:7005

--replicas 1 表示自动为每一个master节点分配一个slave节点，在上面有6个节点，程序会按照一定规则生成 3个master（主）3个slave(从)

运行中，提示Can I set the above configuration? (type 'yes' to accept): yes //输入yes

如果创建失败，开放防火墙监听的端口。

1. **验证集群**

redis-cli -h node1 -p 7000 -c //任何个节点都可以

设置

192.168.137.200:7000> set name yeyang

-> Redirected to slot [5798] located at 192.168.137.124:7003

OK

获取

92.168.137.124:7003> get name

"yeyang"

查看集群信息：

./redis-trib.rb check 192.168.137.200:7000 //任何一个节点都可以

>>> Performing Cluster Check (using node 192.168.137.200:7000)

M: 9e40e764084293d48e59408b213e6238d37c1676 192.168.137.200:7000

slots:0-5460 (5461 slots) master

1 additional replica(s)

M: e82c1a839fd29faac5628a41374f58b07e9aae08 192.168.137.200:7001

slots:10923-16383 (5461 slots) master

1 additional replica(s)

S: 6241699d402f7d8d5e68ead189ccc7cde0d65b6c 192.168.137.124:7005

slots: (0 slots) slave

replicates e82c1a839fd29faac5628a41374f58b07e9aae08

S: d0ef8d6c2e7ca4f2b97381dfc37ef6e59babb926 192.168.137.200:7002

slots: (0 slots) slave

replicates 4d4e09b0c46dd8682d722b89c3c701e1d327534b

M: 4d4e09b0c46dd8682d722b89c3c701e1d327534b 192.168.137.124:7003

slots:5461-10922 (5462 slots) master

1 additional replica(s)

S: f5c962cf35ecbbbd9f8cb87e339d27c51eb2c989 192.168.137.124:7004

slots: (0 slots) slave

replicates 9e40e764084293d48e59408b213e6238d37c1676

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

到此整个集群就搭建完成！

其它命令：

#添加主节点，第二个参数是集群中任意节点

./redis-trib.rb add-node 127.0.0.1:7006 127.0.0.1:7000

#添加从节点，第二个参数是集群中任意节点

./redis-trib.rb add-node --slave 127.0.0.1:7006 127.0.0.1:7000

#添加从节点指定主节点，第二个参数是集群中任意节点

./redis-trib.rb add-node --slave --master-id 3c3a0c74aae0b56170ccb03a76b60cfe7dc1912e 127.0.0.1:7006 127.0.0.1:7000

# 修复插槽

./redis-trib.rb fix 192.168.3.15:6379

****问题解决方案：****

****错误提示是****

slot插槽被占用了（这是 搭建集群前时，以前redis的旧数据和配置信息没有清理干净。）

****解决方案是****

用redis-cli 登录到每个节点执行  flushall  和 cluster reset  就可以了。

然后重新执行群集脚本命令