Redis是一个开源的使用ANSI C语言编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语言的API。和Memcached类似，但它支持存储的value类型相对更多，包括string（字符串）、list（链表）、set（集合）、zset（sorted set -有序集合）和hash（哈希类型）。与Memcached相同的是，为了保证访问效率，数据都是缓存在内存中；与Memcached不同的是，redis会周期性的把更新的数据写入磁盘或者写入追加的记录文件，并且在此基础上实现了master-slave（主从）同步。
Redis的数据可以从master（主）服务器向任意数量的slave（从）服务器上同步，从服务器也可以是关联其他从服务器的主服务器，执行单层树复制，是Redis集群的一个重要特性。Redis集群最少要求6个节点，共需要6台机器，由于资源有限，我这边准备了3台机器，每台机器上配置两个节点。

系统环境

主机名	IP地址	操作系统	节点
Redis1	192.168.2.206	Centos 7.x 64位	Master port 7001、slave port 7002
Redis2	192.168.2.209	Centos 7.x 64位	Master port 7001、slave port 7002
Redis3	192.168.2.210	Centos 7.x 64位	Master port 7001、slave port 7002

一、安装redis

(注：目前Redis最新稳定版为5.0.2，可将以下5.0.0 改为 5.0.2)

查看centos版本：# cat /etc/redhat-release

修改centos主机名：# hostnamectl --static set-hostname redis1

重启centos系统：# reboot -f

1、安装编译相关软件包

[root@redis1 ~]# yum -y install make gcc gcc-c++ wget

2、下载redis并解压

[root@redis1 ~]# cd /tmp

[root@redis1 tmp]# wget http://download.redis.io/releases/redis-5.0.0.tar.gz（只在Redis1执行）

[root@redis1 tmp]# scp redis-5.0.0.tar.gz root@192.168.2.209:/tmp/（只在Redis1执行）

[root@redis1 tmp]# scp redis-5.0.0.tar.gz root@192.168.2.210:/tmp/（只在Redis1执行）

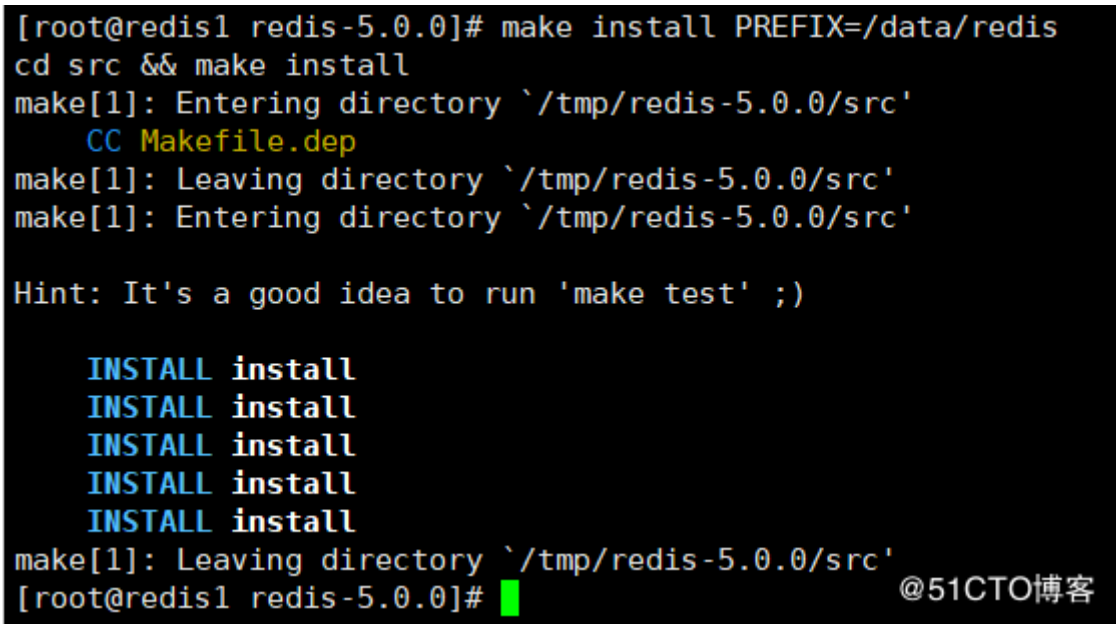
[root@redis1 tmp]# tar zxvf redis-5.0.0.tar.gz

3、安装redis

[root@redis1 tmp]# cd redis-5.0.0/

[root@redis1 redis-5.0.0]# make

[root@redis1 redis-5.0.0]# make install PREFIX=/data/redis



出现上图画面表示安装成功

二、配置集群环境

1、复制配置文件

注：每台机器两个实例，故需要复制两份对应的配置文件

[root@redis1 redis-5.0.0]# mkdir /data/redis/conf

[root@redis1 redis-5.0.0]# cp redis.conf /data/redis/conf/redis-7001.conf

[root@redis1 redis-5.0.0]# cp redis.conf /data/redis/conf/redis-7002.conf

2、修改配置文件

[root@redis1 redis-5.0.0] #vim /data/redis/conf/redis-7001.conf

[root@redis1 redis-5.0.0] #vim /data/redis/conf/redis-7002.conf

修改时，可使用查找的方式；鼠标定位光标：:set mouse=a

修改以下地方

bind 192.168.2.206 #redis监听的本地IP地址

port 7001 #监听端口，另一个节点改为7002

daemonize yes #开启后台运行，no表示运行在前台

pidfile /var/run/redis_7001.pid #pid文件，另一个节点改为7002

appendonly yes #开启aof日志，每次写操作都会记录一条日志

cluster-enabled yes #开启集群，把注释#去掉

cluster-config-file nodes-7001.conf #集群的配置文件，首次启动会自动创建，另一个节点改为7002

cluster-node-timeout 15000 #集群节点连接超时时间，15秒

3、添加firewalld防火墙允许端口

[root@redis1 redis-5.0.0]# firewall-cmd --permanent --add-port=7001-7002/tcp

[root@redis1 redis-5.0.0]# firewall-cmd --permanent --add-port=17001-17002/tcp

[root@redis1 redis-5.0.0]# firewall-cmd --reload

4、启动redis服务

[root@redis1 redis-5.0.0]# cd /root

[root@redis1 ~]# vim redis-all.sh（只在Redis1执行）

创建启动脚本，redis-all.sh

```
#!/bin/bash
/data/redis/bin/redis-server /data/redis/conf/redis-7001.conf
/data/redis/bin/redis-server /data/redis/conf/redis-7002.conf
```

将脚本分发到另外两台主机上

[root@redis1 ~]# scp redis-all.sh root@192.168.2.209:/root/（只在Redis1执行）

[root@redis1 ~]# scp redis-all.sh root@192.168.2.210:/root/（只在Redis1执行）

运行脚本，启动redis，每个节点都要启动

[root@redis1 ~]# sh redis-all.sh

查看Redis运行状态

[root@redis1 ~]# ps -ef |grep redis

按照以上步骤：在Redis2、Redis3执行同样命令

5、启动集群

[root@redis1 ~]# vim redis-cluster.sh（只在Redis1执行）

创建启动集群脚本，redis-cluster.sh

```
#!/bin/bash
/data/redis/bin/redis-cli --cluster create 192.168.2.206:7001 192.168.2.209:7001 192.168.2.210:7001
192.168.2.206:7002 192.168.2.209:7002 192.168.2.210:7002 --cluster-replicas 1
```

参数说明

--cluster create：表示创建redis集群

--cluster-replicas 1：表示为集群中的每一个主节点指定一个从节点，即一比一的复制。

将脚本分发到另外两台主机上

[root@redis1 ~]# scp redis-cluster.sh root@192.168.2.209:/root/（只在Redis1执行）

[root@redis1 ~]# scp redis-cluster.sh root@192.168.2.210:/root/（只在Redis1执行）

运行脚本，启动集群，只需要在其中一个节点启动即可

[root@redis1 ~]# sh redis-cluster.sh（只在Redis1执行）

注：中途需要输入yes来确认

```
>>> Trying to optimize slaves allocation for anti-affinity
[OK] Perfect anti-affinity obtained!
M: e56e21465b2bdf1097b7e690e726d72eb74468c2 192.168.2.206:7001
   slots:[0-5460] (5461 slots) master
M: 1da58bb3bcceac671d25c10ad66b8914e0e53173 192.168.2.209:7001
   slots:[5461-10922] (5462 slots) master
M: ff177612d01eccdef808403281449cc9ceeb7fa9 192.168.2.210:7001
   slots:[10923-16383] (5461 slots) master
S: a6507513f69292fd95d7ef80e90b88ad044c43ba 192.168.2.206:7002
   replicates ff177612d01eccdef808403281449cc9ceeb7fa9
S: 51b701546ddc33742bc89c4c156d8b6144ce0732 192.168.2.209:7002
   replicates e56e21465b2bdf1097b7e690e726d72eb74468c2
S: 63fac038a99d4012e98768f3e1ac9f52ac500c60 192.168.2.210:7002
   replicates 1da58bb3bcceac671d25c10ad66b8914e0e53173
Can I set the above configuration? (type 'yes' to accept): yes
>>> Nodes configuration updated
>>> Assign a different config epoch to each node
>>> Sending CLUSTER MEET messages to join the cluster
Waiting for the cluster to join
..
>>> Performing Cluster Check (using node 192.168.2.206:7001)
M: e56e21465b2bdf1097b7e690e726d72eb74468c2 192.168.2.206:7001
   slots:[0-5460] (5461 slots) master
   1 additional replica(s)
S: 51b701546ddc33742bc89c4c156d8b6144ce0732 192.168.2.209:7002
   slots: (0 slots) slave
   replicates e56e21465b2bdf1097b7e690e726d72eb74468c2
M: 1da58bb3bcceac671d25c10ad66b8914e0e53173 192.168.2.209:7001
   slots:[5461-10922] (5462 slots) master
   1 additional replica(s)
S: a6507513f69292fd95d7ef80e90b88ad044c43ba 192.168.2.206:7002
   slots: (0 slots) slave
   replicates ff177612d01eccdef808403281449cc9ceeb7fa9
M: ff177612d01eccdef808403281449cc9ceeb7fa9 192.168.2.210:7001
   slots:[10923-16383] (5461 slots) master
   1 additional replica(s)
S: 63fac038a99d4012e98768f3e1ac9f52ac500c60 192.168.2.210:7002
   slots: (0 slots) slave
   replicates 1da58bb3bcceac671d25c10ad66b8914e0e53173
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.                    @51CTO博客
```

三、测试

(注：可将以下密码：xuad123456 改为 tomtaw)

1、设置redis集群密码

注：所有节点都需要设置密码，且密码必须一致。

[root@redis1 ~]# /data/redis/bin/redis-cli -h 192.168.2.206 -p 7001 -c

192.168.2.206:7001> config set masterauth xuad123456

192.168.2.206:7001> config set requirepass xuad123456

192.168.2.206:7001> exit

2、将密码写入配置文件

注：所有节点都需要将密码写入配置文件

[root@redis1 ~]# /data/redis/bin/redis-cli -h 192.168.2.206 -p 7001 -c -a xuad123456

192.168.2.206:7001> config rewrite

192.168.2.206:7001> exit

```
[root@redis1 ~]# /data/redis/bin/redis-cli -h 192.168.2.209 -p 7001 -c
192.168.2.209:7001> config set masterauth xuad123456
OK
192.168.2.209:7001> config set requirepass xuad123456
OK
192.168.2.209:7001> config rewrite
(error) NOAUTH Authentication required.
192.168.2.209:7001> exit
[root@redis1 ~]# /data/redis/bin/redis-cli -h 192.168.2.209 -p 7001 -c -a xuad123456
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
192.168.2.209:7001> config rewrite
OK
192.168.2.209:7001> exit
[root@redis1 ~]#
```


3、查看密码

[root@redis1 ~]# /data/redis/bin/redis-cli -h 192.168.2.206 -p 7001 -c -a xuad123456

192.168.2.209:7001> config get masterauth

192.168.2.209:7001> config get requirepass

```
[root@redis1 ~]# /data/redis/bin/redis-cli -h 192.168.2.206 -p 7001 -c -a xuad123456
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
192.168.2.206:7001> get name
-> Redirected to slot [5798] located at 192.168.2.209:7001
(nil)
192.168.2.209:7001> config get masterauth
1) "masterauth"
2) "xuad123456"
192.168.2.209:7001> config get requirepass
1) "requirepass"
2) "xuad123456"
192.168.2.209:7001>
```


附录：

杀掉Redis进程：

[root@redis1 ~]# kill -9 pid (pid用实际的数据替代，可用 ps -ef |grep redis 查看)

打印集群的信息：

192.168.2.209:7001> cluster info

列出集群当前已知的所有节点（node），以及这些节点的相关信息：

`192.168.2.209:7001> cluster nodes`