



链滴

# [Redis] 在一台服务器上模拟集群搭建

作者: [lin772662623](#)

原文链接: <https://ld246.com/article/1530433538166>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## 一台机模拟Cluster

Redis集群中至少应该有三个节点(由投票机制决定的, 节点的fail是通过集群中超过半数的节点检测失时才生效)。要保证集群的高可用, 需要每个节点有一个备份机。

Redis集群至少需要6台服务器。

目的:

- 1.使用一台虚拟机运行6个redis实例。
- 2.需要修改redis的端口号7001-7006

### 1.环境准备

系统方面, 我使用的是Centos6.9

#### 1.1 安装redis 3.0.0

准备好redis-3.0.0.tar.gz,

```
tar -xvf redis-3.0.0.tar.gz
```

```
cd redis-3.0.0
```

```
make
```

```
sudo make install PREFIX=/usr/local/redis
```

执行完后, /usr/local/redis/下应该就有redis相关命令了,

```
-rwxr-xr-x 1 root root 22193 Jul 1 11:16 redis-check-aof  
-rwxr-xr-x 1 root root 4588878 Jul 1 11:16 redis-benchmark
```

```
-rwxr-xr-x 1 root root 45403 Jul 1 11:16 redis-check-dump
-rwxr-xr-x 1 root root 4691793 Jul 1 11:16 redis-cli
-rwxr-xr-x 1 root root 6450193 Jul 1 11:16 redis-sentinel
-rwxr-xr-x 1 root root 6450193 Jul 1 11:16 redis-server
-rw-r--r-- 1 root root 41402 Jul 1 12:27 redis.conf
-rw-r--r-- 1 root root 769 Jul 1 14:08 nodes.conf
```

有了这些文件之后，我们就可以简单的来测试一下了。

## 1.2 运行redis

需要先修改redis.conf，将daemonize 设为yes，设置为后台运行

```
sudo vi /usr/local/redis/redis.conf
```

找到daemonize，修改为yes，保存退出，然后就可以运行redis

```
cd /usr/local/redis/
```

```
sudo ./redis-server redis.conf
```

检查一下

```
ps -ef | grep -i redis
```

有以下内容则表示正常启动了

```
root 5190 0.1 0.3 33936 1712 ? Ssl 18:23 0:00 ./redis-server *:6379
```

这里没启成功的话，检查一下权限，权限不够启不起来的

## 1.3 测试redis

执行redis目录下的redis-cli就可以了

```
cd /usr/local/redis/
```

```
sudo ./redis-cli [-h IP] -p 6379
```

-h: 连接的服务器的地址 -p: 服务的端口号

用本地127.0.0.1的话就不需要加-h IP了。

```
127.0.0.1:6379> set str1 hello
```

```
OK
```

```
127.0.0.1:6379> get str1
```

```
"hello"
```

```
127.0.0.1:6379> quit
```

这样就可以了，能存能取。

然后我们停掉这个redis

```
sudo ./redis-cli -p 6379 shutdown
```

## 1.4 设置多节点

在一台机上部署集群的话，我们需要在一台机上弄6个redis-server，  
执行

```
cd /usr/local/  
mkdir redis-cluster  
sudo cp -r redis redis-cluster/redis01
```

这里需要检查一下redis01里有没有 dump.rdb，有的话就删了。

```
sudo cp -r redis01/ redis02  
sudo cp -r redis01/ redis03  
sudo cp -r redis01/ redis04  
sudo cp -r redis01/ redis05  
sudo cp -r redis01/ redis06
```

在这里还需要再复制一个文件，就从redis的源码包里取

```
sudo cp redis-3.0.0/src/redis-trib.rb /usr/local/redis-cluster/
```

复制完之后，每个redis的配置文件都要修改以下内容：

```
1.enable cluster  
cluster-enabled yes  
2.修改端口  
port 7001(按照123456这样分)
```

每个都弄好之后，我们就要启动他们了，怎么启动呢。可以写个脚本

```
sudo vi /usr/local/redis-cluster/start-all.sh
```

```
cd redis01  
./redis-server redis.conf  
cd ..  
cd redis02  
./redis-server redis.conf  
cd ..  
cd redis03  
./redis-server redis.conf  
cd ..  
cd redis04  
./redis-server redis.conf  
cd ..  
cd redis05  
./redis-server redis.conf  
cd ..  
cd redis06  
./redis-server redis.conf  
cd ..
```

保存退出后，赋予权限，

```
sudo chmod +x start-all.sh
```

执行

```
sudo ./start-all.sh
```

检查一下

```
ps -ef | grep -i redis
```

```
root  33946  1 0 14:07 ?        00:00:06 ./redis-server *:7001 [cluster]
root  33950  1 0 14:07 ?        00:00:05 ./redis-server *:7002 [cluster]
root  33954  1 0 14:07 ?        00:00:06 ./redis-server *:7003 [cluster]
root  33956  1 0 14:07 ?        00:00:05 ./redis-server *:7004 [cluster]
root  33962  1 0 14:07 ?        00:00:05 ./redis-server *:7005 [cluster]
root  33964  1 0 14:07 ?        00:00:06 ./redis-server *:7006 [cluster]
llq   34348 32038 0 15:47 pts/4  00:00:00 grep -i redis
```

有上面这些进程就代表成功了。下面可以做集群搭建了

## 2. 集群搭建

### 2.1 安装软件

官方提供的脚本，是要用ruby执行的，所以需要安装ruby

```
sudo yum install ruby
```

```
sudo yum install rubygems
```

```
sudo gem install redis-3.0.0.gem
```

这个redis-3.0.0.gem文件，从网上可以下载，也可以选择直接指定版本安装

```
gem install redis --version 3.0.0
```

注意：gem install redis --version 3.0.0 失败的话，需要修改一下gem的源

```
gem sources --remove https://rubygems.org/
```

```
gem sources -a https://ruby.taobao.org/
```

gem安装完redis后，可以开始运行ruby脚本了。

### 2.2 运行集群

同样的，我们写个脚本，把命令保存起来，下次就不用再敲了。

```
cd /usr/local/redis-cluster
```

```
vi setup-cluster.sh
```

```
./redis-trib.rb create --replicas 1 192.168.200.128:7001 192.168.200.128:7002 192.168.200.128:7003 192.168.200.128:7004 192.168.200.128:7005 192.168.200.128:7006
```

同时顺便写个关闭集群的脚本

```
vi shutdown-all.sh
```

```
redis01/redis-cli -p 7001 shutdown
redis01/redis-cli -p 7002 shutdown
redis01/redis-cli -p 7003 shutdown
redis01/redis-cli -p 7004 shutdown
redis01/redis-cli -p 7005 shutdown
redis01/redis-cli -p 7006 shutdown
```

好了，两个脚本都要赋予执行权限

```
sudo chmod +x shutdown-all.sh setup-cluster.sh
```

```
sudo ./setup-cluster.sh
```

具体跑的内容找不到了，随便贴一段，中间需要我们输入yes确认集群配置，yes之后就完成了。

```
>>> Creating cluster
Connecting to node 192.168.200.128:7001: OK
Connecting to node 192.168.200.128:7002: OK
Connecting to node 192.168.200.128:7003: OK
Connecting to node 192.168.200.128:7004: OK
Connecting to node 192.168.200.128:7005: OK
Connecting to node 192.168.200.128:7006: OK
>>> Performing hash slots allocation on 6 nodes...
Using 3 masters:
192.168.200.128:7001
192.168.200.128:7002
192.168.200.128:7003
Adding replica 192.168.200.128:7004 to 192.168.200.128:7001
Adding replica 192.168.200.128:7005 to 192.168.200.128:7002
Adding replica 192.168.200.128:7006 to 192.168.200.128:7003
M: 2e48ae301e9c32b04a7d4d92e15e98e78de8c1f3 192.168.200.128:7001
  slots:0-5460 (5461 slots) master
M: 8cd93a9a943b4ef851af6a03edd699a6061ace01 192.168.200.128:7002
  slots:5461-10922 (5462 slots) master
M: 2935007902d83f20b1253d7f43dae32aab9744e6 192.168.200.128:7003
  slots:10923-16383 (5461 slots) master
S: 74f9d9706f848471583929fc8bbde3c8e99e211b 192.168.200.128:7004
  replicates 2e48ae301e9c32b04a7d4d92e15e98e78de8c1f3
S: 42cc9e25ebb19dda92591364c1df4b3a518b795b 192.168.200.128:7005
  replicates 8cd93a9a943b4ef851af6a03edd699a6061ace01
S: 8b1b11d509d29659c2831e7a9f6469c060dfcd39 192.168.200.128:7006
  replicates 2935007902d83f20b1253d7f43dae32aab9744e6
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 192.168.200.128:7001)
M: 2e48ae301e9c32b04a7d4d92e15e98e78de8c1f3 192.168.200.128:7001
  slots:0-5460 (5461 slots) master
M: 8cd93a9a943b4ef851af6a03edd699a6061ace01 192.168.200.128:7002
  slots:5461-10922 (5462 slots) master
M: 2935007902d83f20b1253d7f43dae32aab9744e6 192.168.200.128:7003
```

```
slots:10923-16383 (5461 slots) master
M: 74f9d9706f848471583929fc8bbde3c8e99e211b 192.168.200.128:7004
slots: (0 slots) master
replicates 2e48ae301e9c32b04a7d4d92e15e98e78de8c1f3
M: 42cc9e25ebb19dda92591364c1df4b3a518b795b 192.168.200.128:7005
slots: (0 slots) master
replicates 8cd93a9a943b4ef851af6a03edd699a6061ace01
M: 8b1b11d509d29659c2831e7a9f6469c060dfcd39 192.168.200.128:7006
slots: (0 slots) master
replicates 2935007902d83f20b1253d7f43dae32aab9744e6
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
[root@localhost redis-cluster]#
```

下面我们可以写下代码测试一下集群

## 3 Jedis测试集群

### 3.1 添加依赖

这里不写maven工程创建的了，自己搞定哈。

创建完maven工程后，在pom.xml中添加依赖

```
<!-- Redis客户端 -->
<dependency>
  <groupId>redis.clients</groupId>
  <artifactId>jedis</artifactId>
</dependency>
<!-- 测试 -->
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
</dependency>
```

创建一个TestJedis.java，运行一下看看。

```
public class TestJedis {

    @Test
    public void testJedisCluster() throws Exception {
        //创建一个HostAndPort的Set集合
        Set<HostAndPort> nodes = new HashSet<>();
        nodes.add(new HostAndPort("192.168.200.128",7001));
        nodes.add(new HostAndPort("192.168.200.128",7002));
        nodes.add(new HostAndPort("192.168.200.128",7003));
        nodes.add(new HostAndPort("192.168.200.128",7004));
        nodes.add(new HostAndPort("192.168.200.128",7005));
        nodes.add(new HostAndPort("192.168.200.128",7006));
        //创建一个JedisCluster对象，参数是HostAndPort的Set集合
```

```
JedisCluster jedisCluster = new JedisCluster(nodes);  
//使用JedisCluster对象查询  
jedisCluster.set("jedis-cluster", "hello world");  
System.out.println(jedisCluster.get("jedis-cluster"));  
//关闭jediscluster  
jedisCluster.close();  
}  
}
```

到这里，简单的redis伪集群搭建就结束了。