Redis3.2.6集群环境搭建

技术文档

（版本：V1.0.0）

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 日期 | 说明 | 作者 |
| V1.0.0 | 2016/12/27 | 创建 | Simon Hoo |
|  |  |  |  |

目 录

一． 环境准备 - 3 -

二． 软件准备 - 4 -

2.1 下载Redis3.2.6 - 4 -

2.2 将软件上传到服务器 - 4 -

三． 安装Redis - 5 -

3.1 安装系统组件 - 5 -

3.2 安装Redis - 5 -

四． 配置集群模式 - 8 -

4.1 配置redis.conf - 8 -

4.2 安装Redis集群所需的Ruby工具 - 9 -

4.3 本地编译发布 - 9 -

五． 在OSS中发布构件 - 11 -

5.1 登录Sonatype OSS - 11 -

5.2 确认及发布构件 - 11 -

5.3 查看发布后的构构 - 14 -

## Redis集群方案

Redis集群方案采用Redis3.0支持的redis-cluster集群架构：



架构方案：

1. 所有的redis节点彼此互联（PING-PONG机制），内部使用二进制协议优化传输速度和带宽；
2. 节点的fail是通过集群中超过半数的节点检测失效时才生效；
3. 客户端与redis节点直连，不需要中间proxy层，客户端不需要连接集群所有的节点，连接集群中任何一个可用的节点即可。
4. redis-cluster把所有的物理节点映射到[0~16383]slot上，cluster负责维护node<->slot<->value。

容错机制：

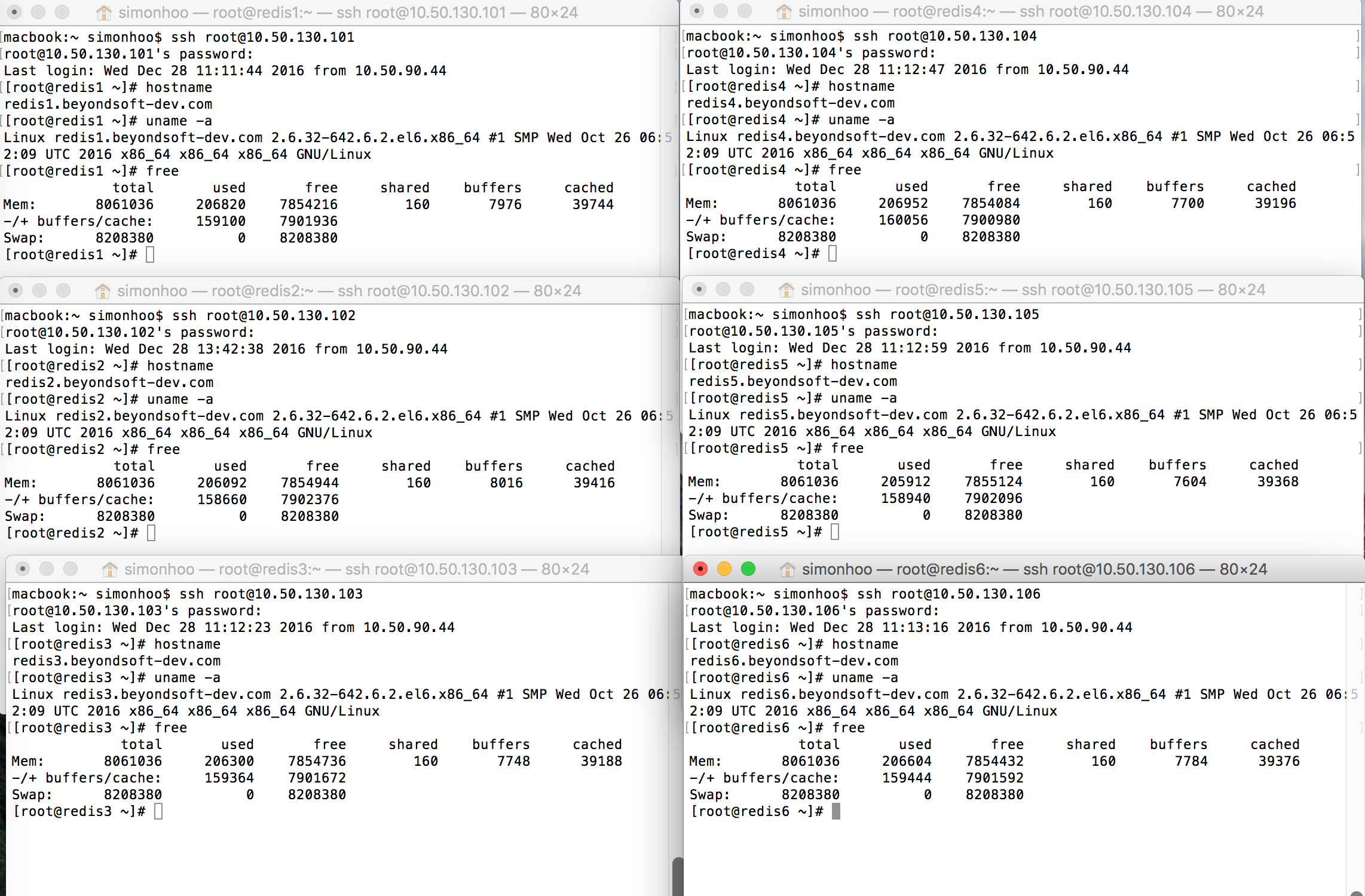


1. redis-cluster的容错机制是通过选举产生master,选择过程是集群中所有的master参与，如果半数以上的master节点与master节点的通信超时（cluster-node-timeout）,则认为当前master节点挂掉。
2. 什么时候整个集群不可用（cluster\_state:fail）,当集群不可用时，所有对集群的操作均不可用，收到ERRORQ错误。

## 环境准备

### 服务器清单

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | IP地址 | HOSTNAME | 配置 | 备注 |
| 1 | 10.50.130.101 | redis1.beyondsoft-dev.com | CPU:2\*2  内存：8GB |  |
| 2 | 10.50.130.102 | redis2.beyondsoft-dev.com | CPU:2\*2  内存：8GB |  |
| 3 | 10.50.130.103 | redis3.beyondsoft-dev.com | CPU:2\*2  内存：8GB |  |
| 4 | 10.50.130.104 | redis4.beyondsoft-dev.com | CPU:2\*2  内存：8GB |  |
| 5 | 10.50.130.105 | redis5.beyondsoft-dev.com | CPU:2\*2  内存：8GB |  |
| 6 | 10.50.130.106 | redis6.beyondsoft-dev.com | CPU:2\*2  内存：8GB |  |



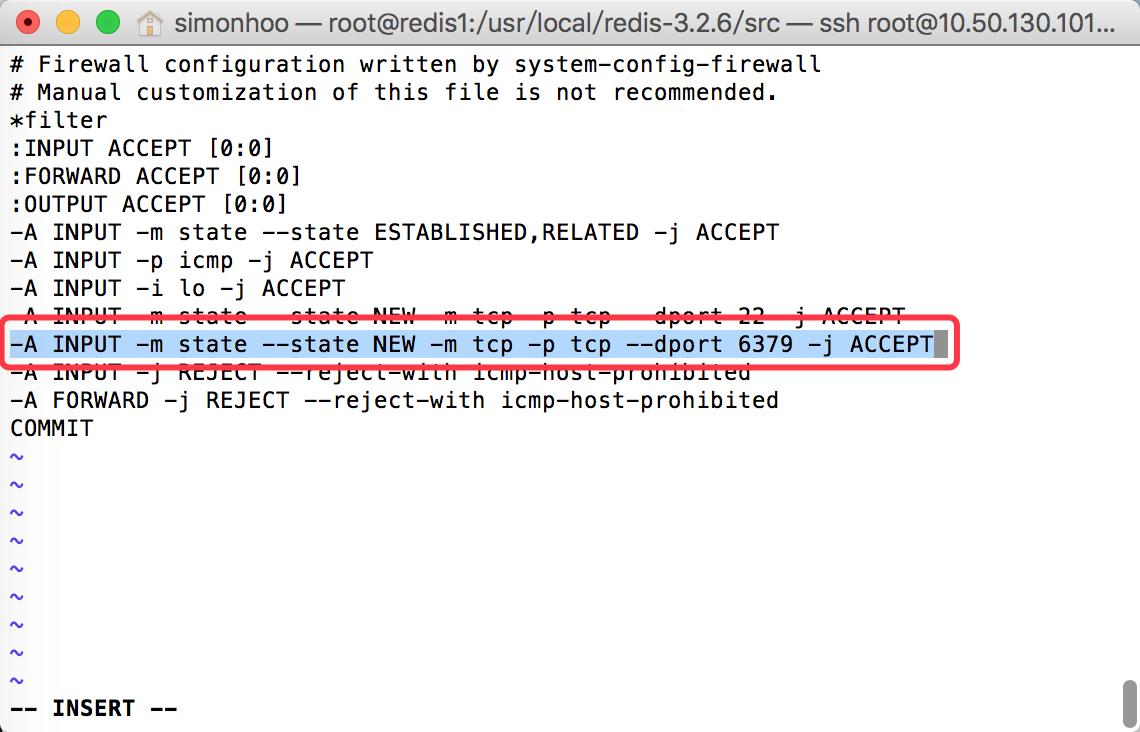
*注：本文档中所演示的操作系统为CentOS 6.8 64位操作系统。*

### 服务器设置

Step1. 将6379端口在防火墙看开启：

[root@redis1 ~]# vi /etc/sysconfig/iptables

-A INPUT -m state --state NEW -m tcp -p tcp --dport 6379 -j ACCEPT

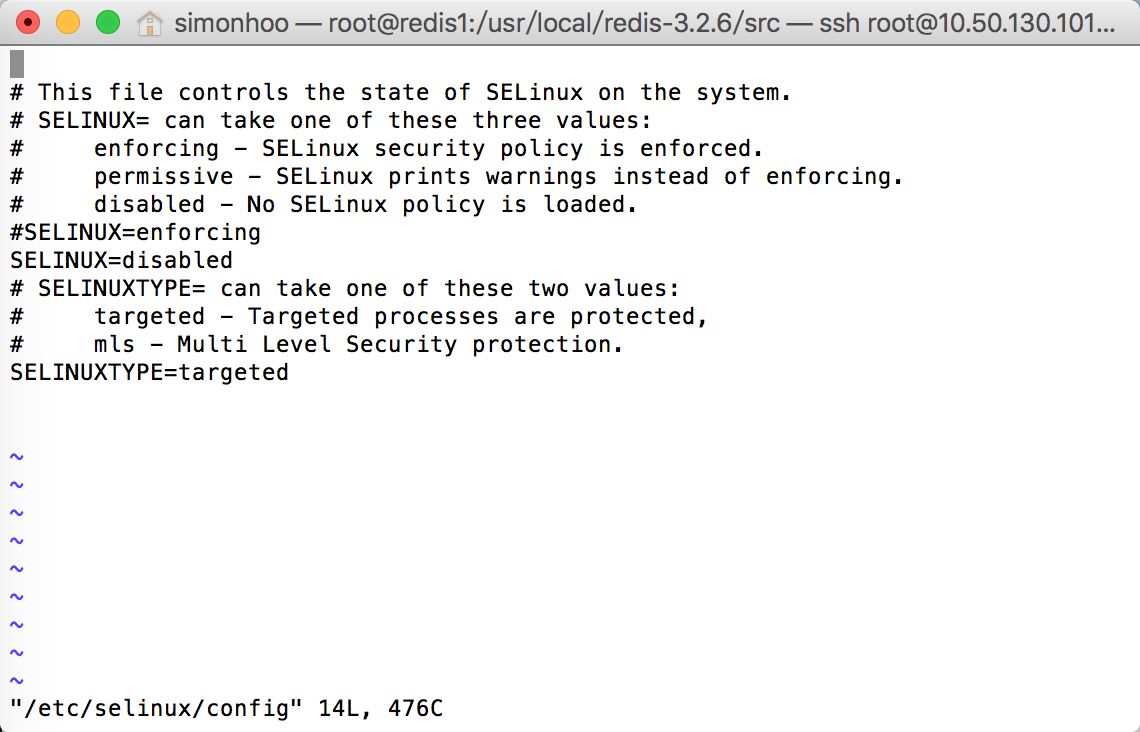


*注：必须加在REJECT前面。*

Step 2. 修改Selinux参数：

[root@redis1 ~]# vi /etc/selinux/config

SELINUX=disabled

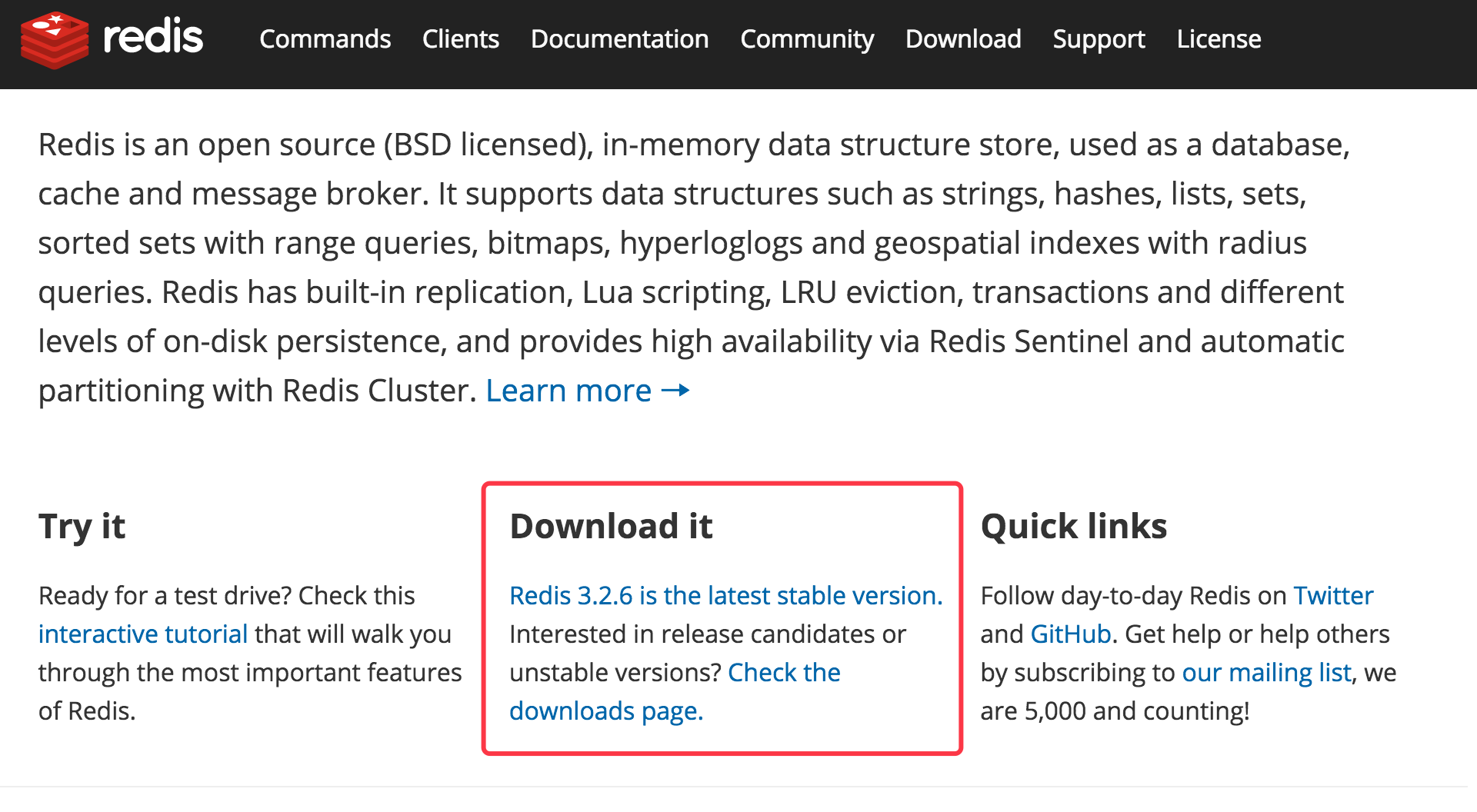


*注：分别在其它几个节点上同样配置防火墙和SELINUX。*

## 软件准备

### 下载Redis3.2.6

<https://redis.io/>



*注：本文中演示的Redis版本为Redis 3.2.6，通用于Redis 3.x的所有版本，如果Redis版本为2.x的，则有所不同。*

### 将软件上传到服务器

Step 1. 分别在6个服务器上创建software目录，用于存放待安装的软件：

[root@redis1 ~]# mkdir /root/software

[root@redis2 ~]# mkdir /root/software

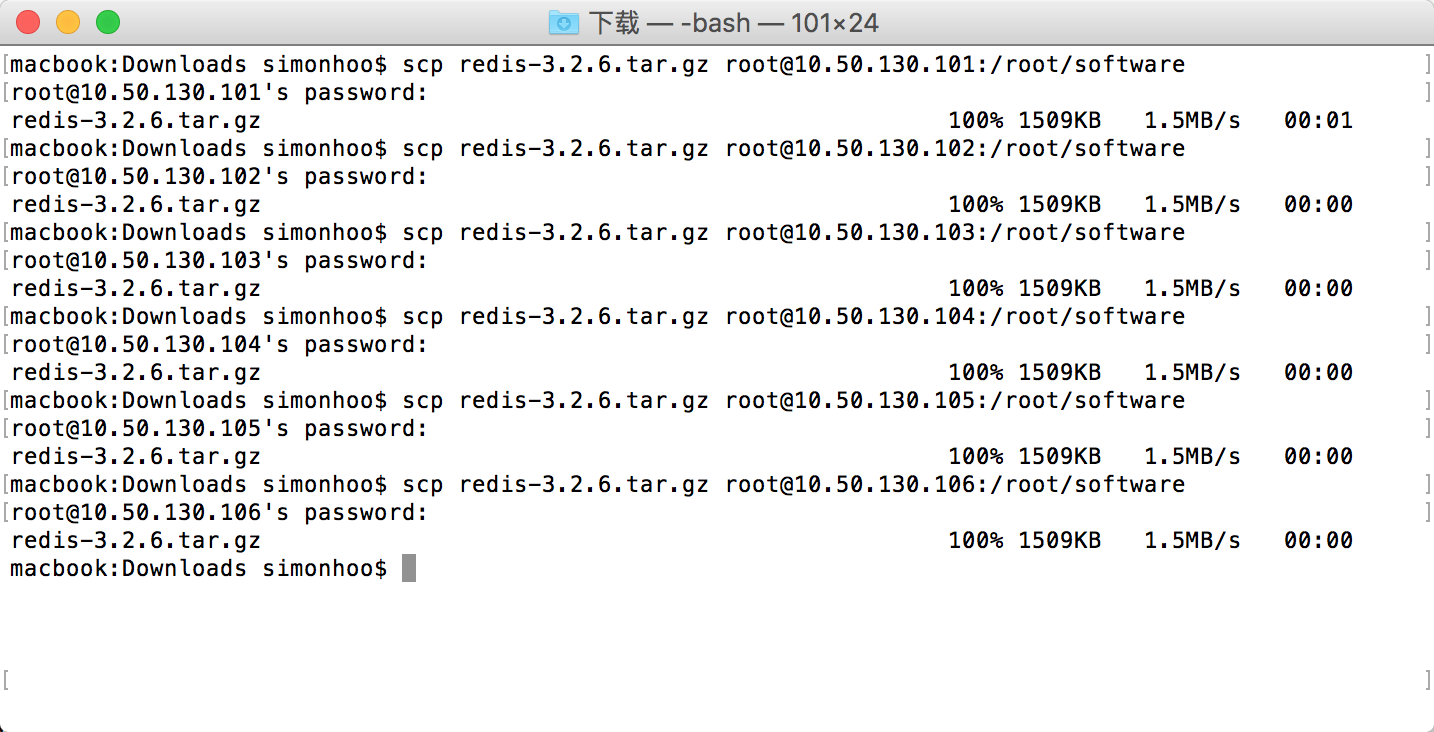
[root@redis3 ~]# mkdir /root/software

[root@redis4 ~]# mkdir /root/software

[root@redis5 ~]# mkdir /root/software

[root@redis6 ~]# mkdir /root/software

Step 2. 将本地下载好的软件SCP到服务器：



## 安装Redis

### 安装系统组件

Step 1. 安装gcc

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

Step 2. 安装tcl

[root@redis1 ~]# yum install –y tcl

### 安装Redis

Step 1. 解压Redis到/usr/local目录下

[root@redis1 ~]# tar -zxvf /root/software/redis-3.2.6.tar.gz -C /usr/local/

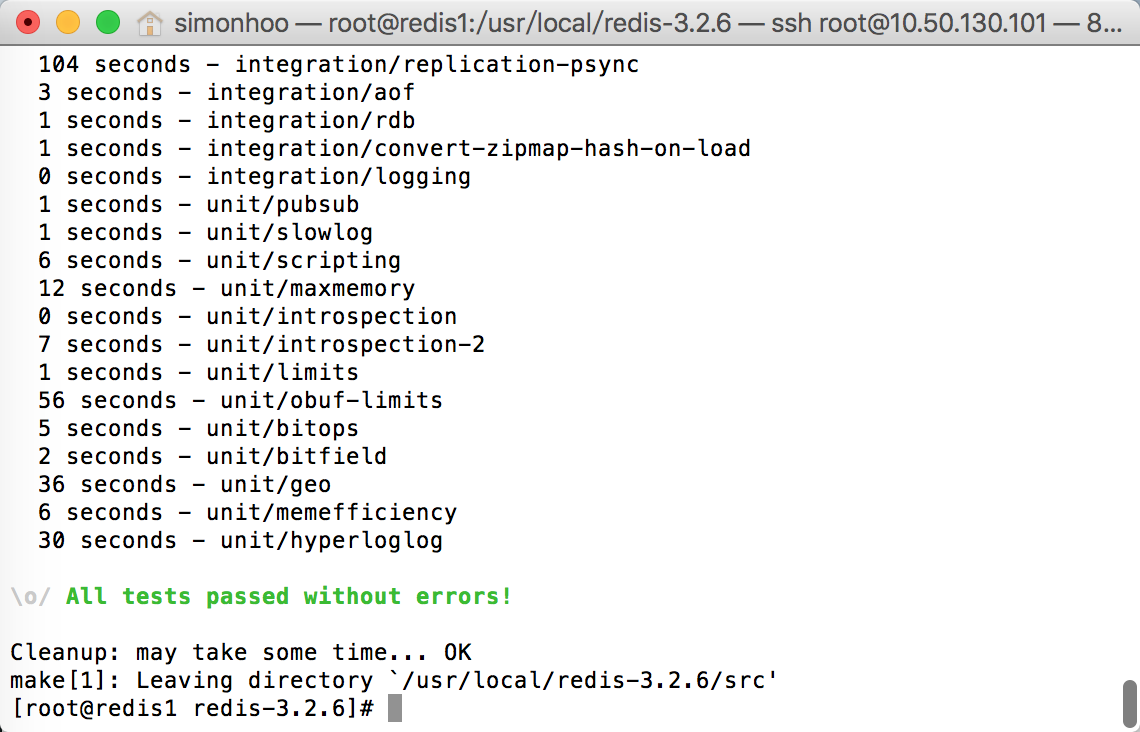
Step 2. 在解压后的目录中进行make和make test

[root@redis1 ~]# cd /usr/local/redis-3.2.6

[root@redis1 redis-3.2.6]# make

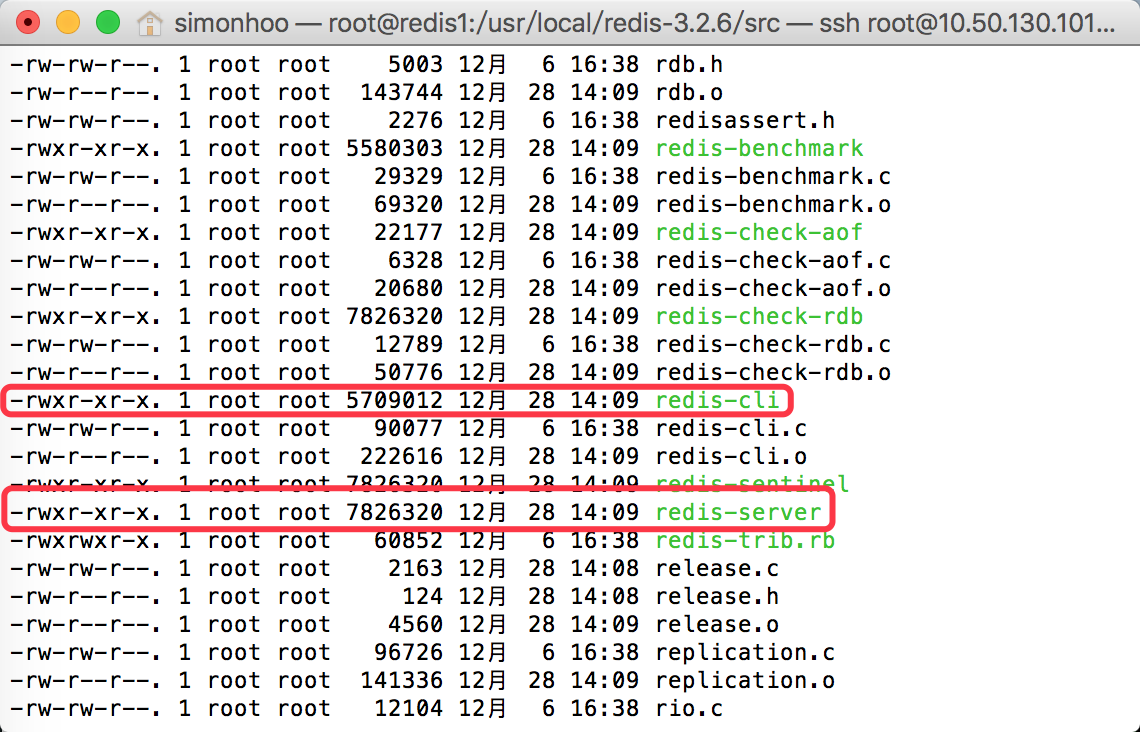
[root@redis1 redis-3.2.6]# make test

*注：要检查make和make test的结果是否都正确，如果报错，针对性检查并安装系统缺少的组件。*



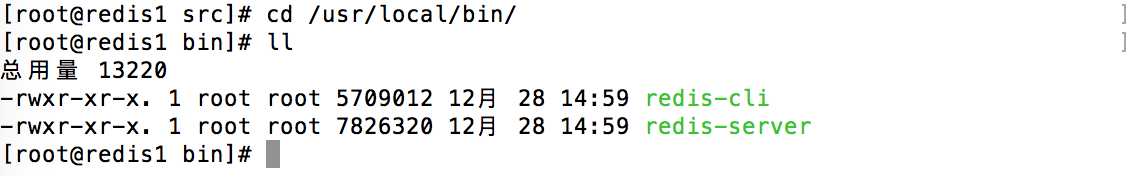
Step 3. 复制redis-server和redis-cli到/usr/local/bin目录下：

[root@redis1 redis-3.2.6]# cd src



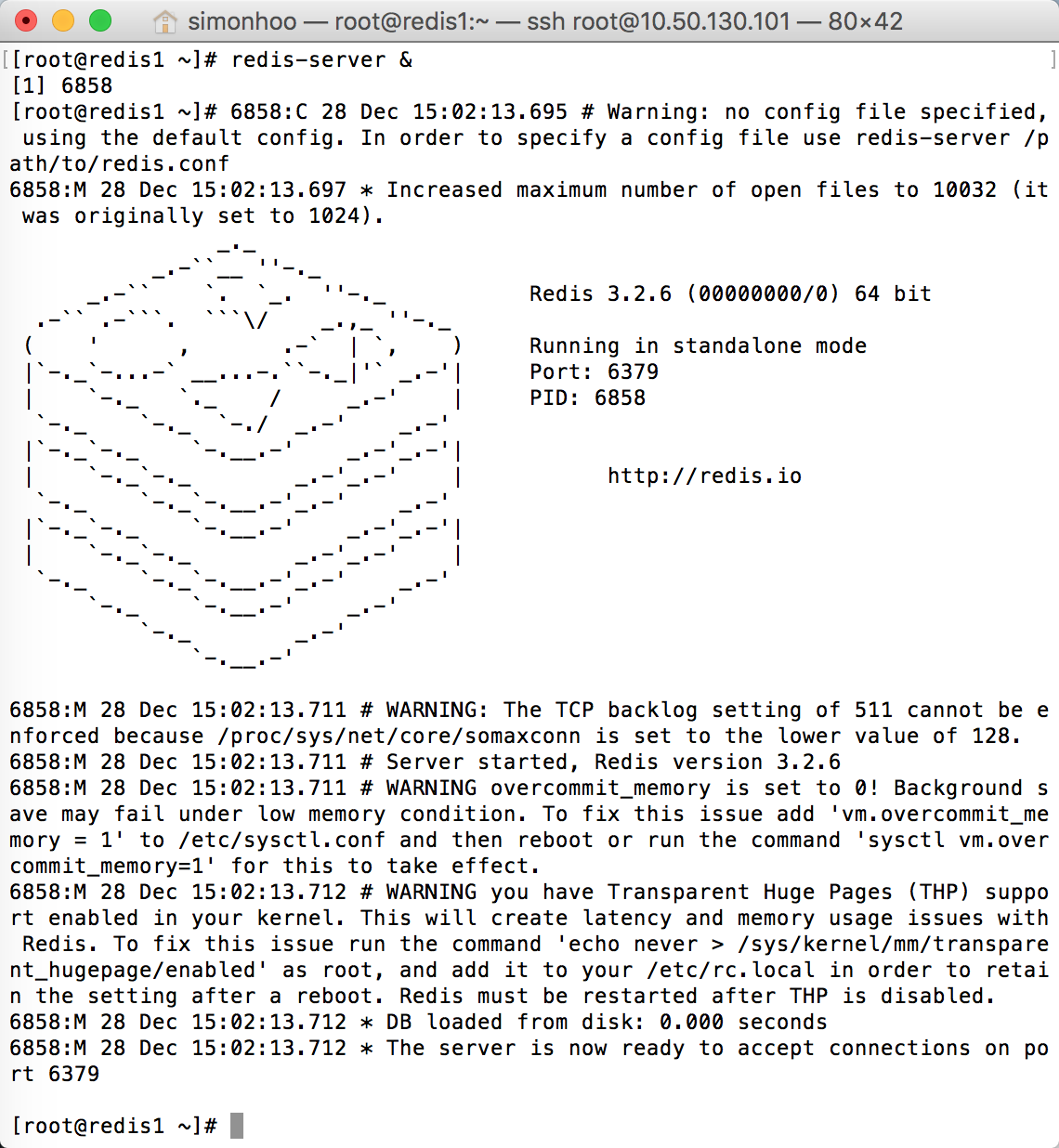
[root@redis1 src]# cp redis-server /usr/local/bin/

[root@redis1 src]# cp redis-cli /usr/local/bin/

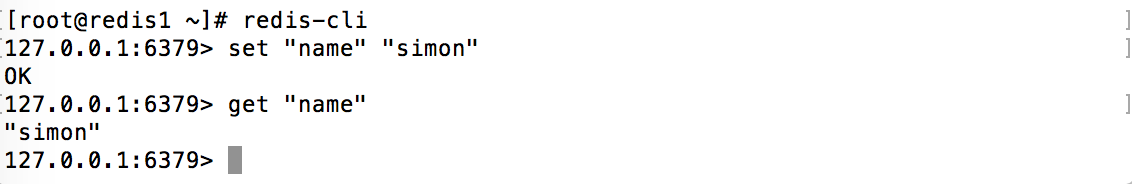


Step 3. 验证Redis安装是否成功：

[root@redis1 ~]# redis-server



[root@redis1 ~]# redis-cli



*注：安装其它5台服务器，重复本章节的4.1~4.2。*

## 配置集群模式

### 配置redis.conf

Step 1. 配置redis.conf

[root@redis1 ~]# mkdir /etc/redis

[root@redis1 ~]# cd /etc/redis

[root@redis ~]# vi redis.conf

port 6379

daemonize yes

cluster-enabled yes

cluster-config-file /etc/redis/nodes.conf

cluster-node-timeout 5000

appendonly yes

requirepass Ab123456

*注1：cluster-node-timeout是集群中各节点相互通讯时，允许“失联”的最大毫秒数，本演示中配置的为5秒，如果超过5秒某个节点没有向其它节点汇报成功，认为该节点挂了。*

*注2：requirepass是Redis访问密码，为了安全起见，该参数建议必须配置，从但客户端Jedis版本必须使用2.8.x以上的版本，否则需要通过扩展JedisCluster来实现对密码访问的支持。此外几个Redis节点的密码应该设置为相同的。*

*注3：分别在其它几个节点上创建与上面相同的redis.conf文件，内容也相同。*

*注4：重启/重建Redis集群时，必须删除去*/etc/redis/nodes.conf文件。

Step 2. 以次启动所有节点

[root@redis1 ~]# redis-server /etc/redis/redis.conf

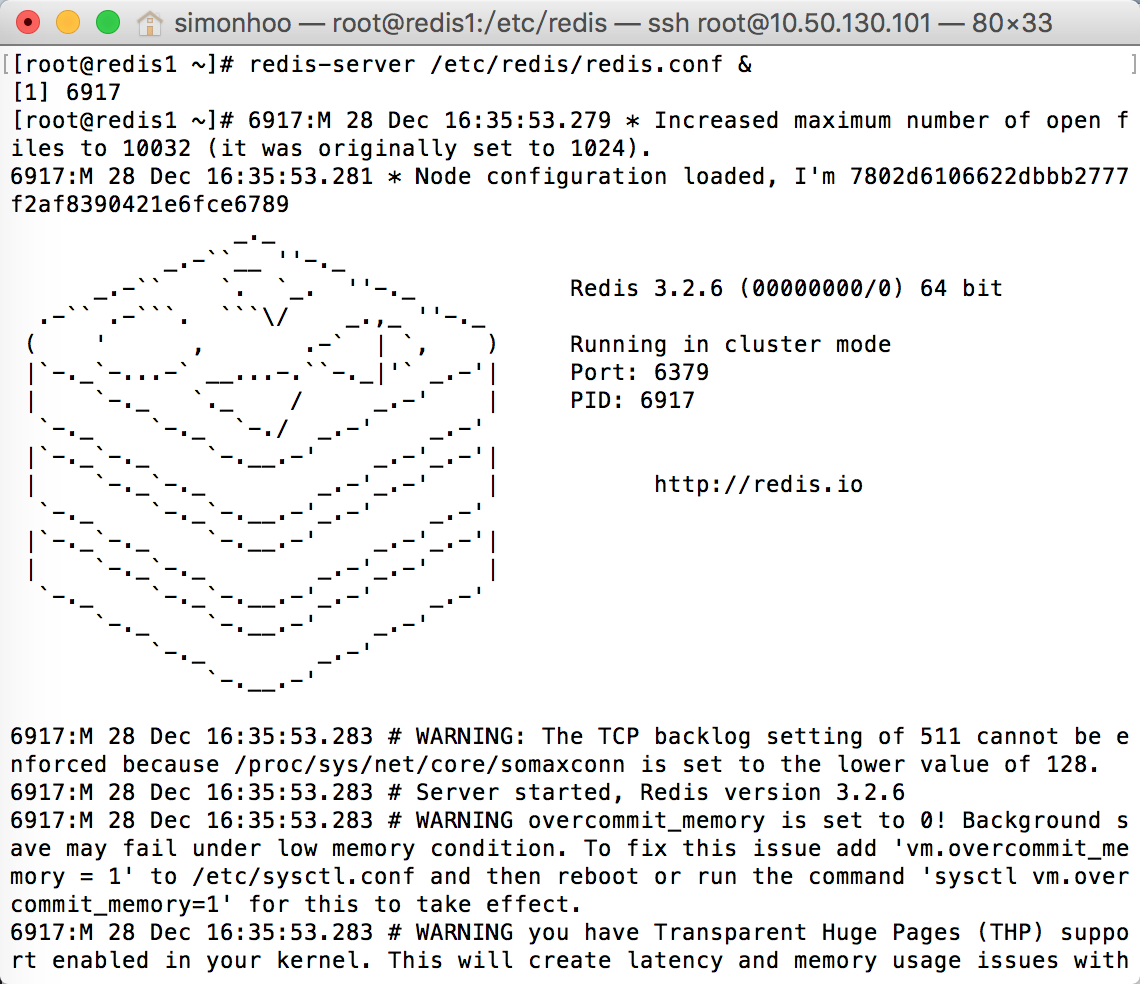
[root@redis2 ~]# redis-server /etc/redis/redis.conf

[root@redis3 ~]# redis-server /etc/redis/redis.conf

[root@redis4 ~]# redis-server /etc/redis/redis.conf

[root@redis5 ~]# redis-server /etc/redis/redis.conf

[root@redis6 ~]# redis-server /etc/redis/redis.conf



### 安装Redis集群所需的Ruby工具

Step 1. 安装Ruby工具：

Redis集群需要借助其它工具将相关节点加入到Cluster中，而这个工具是由Redis提供一个名为redis-trib.rb的ruby脚本，否则接下来创建cluster会失败。

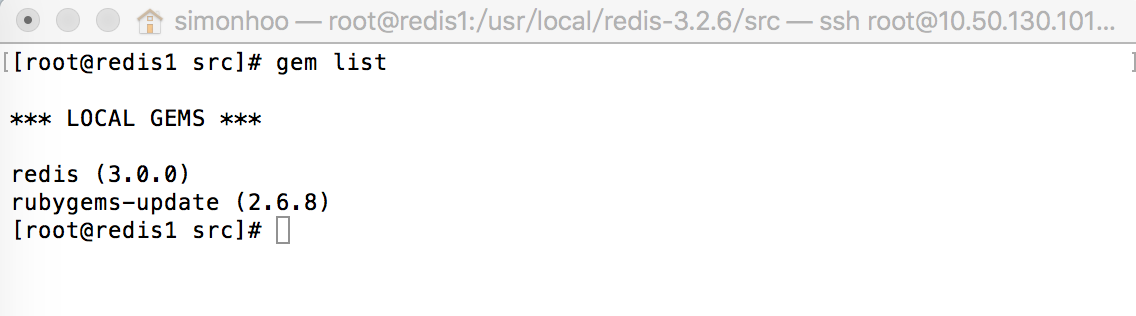
[root@redis1 ~]# cd /usr/local/redis-3.2.6/src

[root@redis1 src]# yum install –y ruby

[root@redis1 src]# yum install -y rubygems

[root@redis1 src]# gem install redis --version 3.0.0

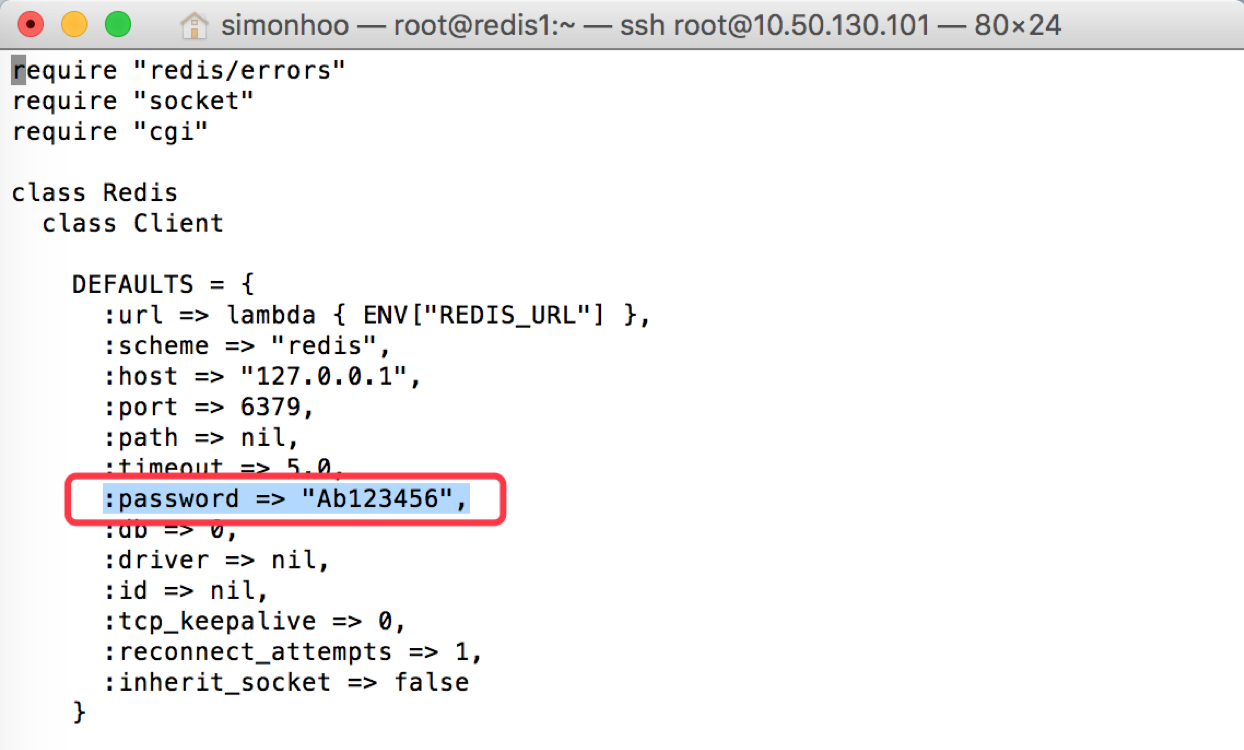
[root@redis1 src]# gem list



Step 2. 设置Ruby连接Redis的密码：

[root@redis1 ~]# vi /usr/lib/ruby/gems/1.8/gems/redis-3.0.0/lib/redis/client.rb

:password => "Ab123456"

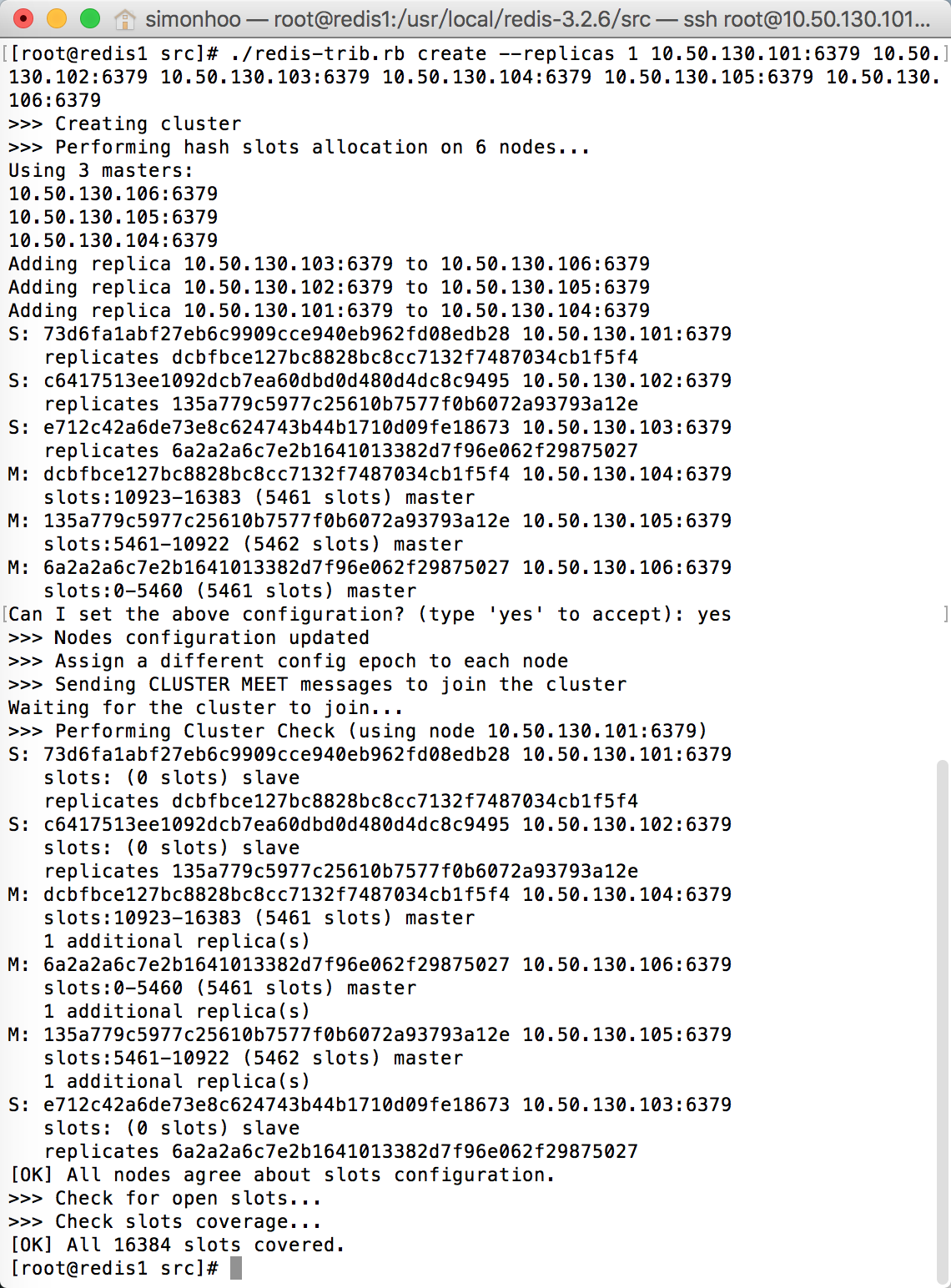


*注：分别在其它几个节点上用同样的方式安装好Ruby工具。*

### 利用redis-trib.rb创建Redis集群

Step 1. 在src目录下运行以下脚本：  
[root@redis1 ~]# cd /usr/local/redis-3.2.6/src

[root@redis1 src]# ./redis-trib.rb create --replicas 1 10.50.130.101:6379 10.50.130.102:6379 10.50.130.103:6379 10.50.130.104:6379 10.50.130.105:6379 10.50.130.106:6379



*注1：只需在其中某个个节点执行以上脚本（本例在第一个节点执行）。*

*注2：利用redis-trib创建Cluster，只需要操作一次即可，假设系统关机、重启，把所有的节点全部关闭之后，下次重启后，即自动进入Cluster模式，不用现次执行redis-trib.rb cteate命令。*

Step 2. 查看Cluster进程：

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

[root@redis2 ~]# ps -ef|grep redis

[root@redis3 ~]# ps -ef|grep redis

[root@redis4 ~]# ps -ef|grep redis

[root@redis5 ~]# ps -ef|grep redis

[root@redis6 ~]# ps -ef|grep redis

Step 3. 查看节点属性（Master/Slave）

[root@redis1 ~]# cd /usr/local/redis-3.2.6/src

[root@redis1 src]# ./redis-trib.rb check 10.50.130.101:6379

[root@redis1 src]# ./redis-trib.rb check 10.50.130.102:6379

[root@redis1 src]# ./redis-trib.rb check 10.50.130.103:6379

[root@redis1 src]# ./redis-trib.rb check 10.50.130.104:6379

[root@redis1 src]# ./redis-trib.rb check 10.50.130.105:6379

[root@redis1 src]# ./redis-trib.rb check 10.50.130.106:6379

Step 4. 查看节点/集群信息

redis-cli客户端登录到任一个节点，查看:



## Jedis测试Redis集群

### JAVA测试代码

package com.cottsoft.redis;

import java.util.HashSet;

import java.util.Set;

import org.apache.commons.pool2.impl.GenericObjectPoolConfig;

import redis.clients.jedis.HostAndPort;

import redis.clients.jedis.JedisCluster;

/\*\*

\* @author Simon.Hoo

\* @version v1.0.0

\* @since JDK1.7

\*/

public class RedisTest {

private static JedisCluster cluster;

/\*\*

\* @param args

\*/

public static void main(String[] args) {

HostAndPort host1 = new HostAndPort("10.50.130.101", 6379);

HostAndPort host2 = new HostAndPort("10.50.130.102", 6379);

HostAndPort host3 = new HostAndPort("10.50.130.103", 6379);

HostAndPort host4 = new HostAndPort("10.50.130.104", 6379);

HostAndPort host5 = new HostAndPort("10.50.130.105", 6379);

HostAndPort host6 = new HostAndPort("10.50.130.106", 6379);

Set<HostAndPort> jedisClusterNode = new HashSet<HostAndPort>();

jedisClusterNode.add(host1);

jedisClusterNode.add(host2);

jedisClusterNode.add(host3);

jedisClusterNode.add(host4);

jedisClusterNode.add(host5);

jedisClusterNode.add(host6);

GenericObjectPoolConfig poolConfig = new GenericObjectPoolConfig();

poolConfig.setMaxWaitMillis(-1);

poolConfig.setMaxTotal(1000);

poolConfig.setMinIdle(8);

poolConfig.setMaxIdle(100);

int connectionTimeout=10;

int soTimeout=10;

int maxAttempts = 10;

String password="Ab123456";

cluster = new JedisCluster(jedisClusterNode, connectionTimeout,soTimeout,maxAttempts,password,poolConfig);

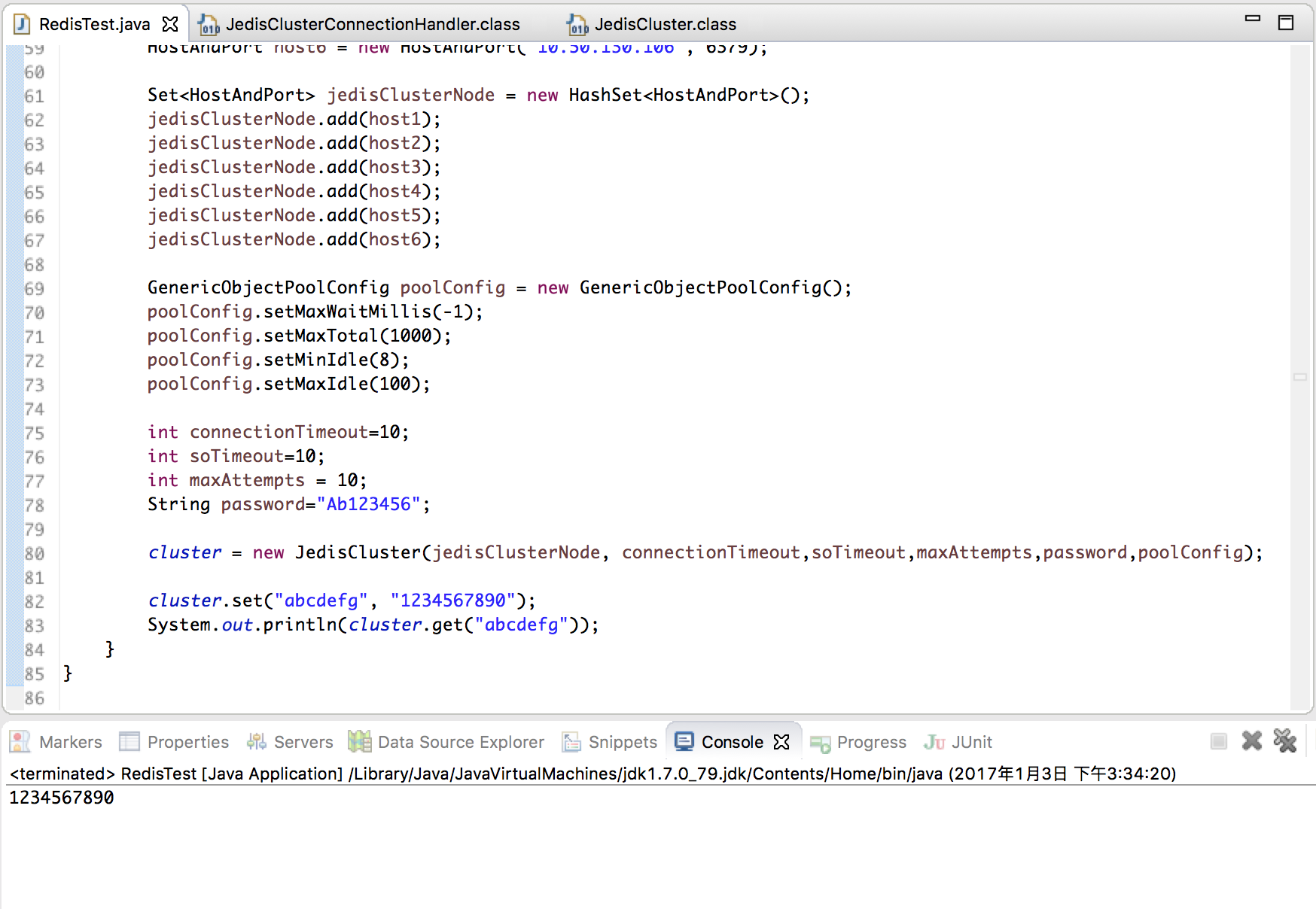
cluster.set("abcdefg", "1234567890");

System.out.println(cluster.get("abcdefg"));

}

}

### 测试结果



PASS.