环境介绍

1、软件版本

```
VMware® Workstation 15 Pro 15.5.6 build-16341506
虚机系统 SentOS7 CentOS-7-x86_64-DVD-1503-01
shell工具 XShell5
redis-5.0.5.tar.gz
```

2、虚机作用

IP	主机名	端口使用
192.168.41.140	redis1	7001做master 7002做slaver
192.168.41.141	redis2	7001做master 7002做slaver
192.168.41.142	redis3	7001做master 7002做slaver
192.168.41.143	redis4	7001做master 7002做slaver

3、搭建步骤（三主三从）

1）安装依赖包(Redis是C语言开发，需要安装gcc依赖环境)

```
yum install -y gcc gcc-c++
```

2）下载最新版redis安装包并解压

```
cd /usr/local/src
wget http://download.redis.io/releases/redis-5.0.5.tar.gz
tar -zxvf redis-5.0.5.tar.gz
mkdir -p /usr/local/redis/{etc,data}
cd redis-5.0.5/
```

3）集群部署

以192.168.41.140这一台主机为例，其余三台主机在安装、配置的方法是相同的

第一步：创建7001实例，复制配置文件

```
mkdir -p /usr/local/redis-cluster/7001/{etc,data}

cd /usr/local/src/redis-5.0.5/
make install PREFIX=/usr/local/redis-cluster/7001

cp /usr/local/src/redis-5.0.5/redis.conf /usr/local/redis-cluster/7001/redis.conf
```

第二步：编辑配置文件

```
vim /usr/local/redis-cluster/7001/redis.conf

# 注释本机ip，让外部ip可以访问
#bind 127.0.0.1
# 关闭受保护模式，可以免密码连接
protected-mode no
port 7001
# 开启后台运行
deamonize yes
 # 开启集群模式
 cluster-enabled yes
```

第三步：复制7001，创建7002实例，注意端口修改
注意：创建实例，即拷贝单机版安装时，生成的bin目录：7001目录。

```
#复制目录
cp -R /usr/local/redis-cluster/7001 /usr/local/redis-cluster/7002
#修改7002的配置文件，port
vim /usr/local/redis-cluster/7002/redis.conf # port 7002
```

第四步：创建start.sh，启动主机上的实例

```
cd /usr/local/redis-cluster/
cd 7001/bin
./redis-server ../redis.conf

cd ../../
cd 7002/bin
./redis-server ../redis.conf

# 查看启动的进程
 ps -ef | grep redis
```

chmod u+x start.sh (赋写和执行的权限)

以上四步操作，在192.168.41.141和192.168.41.142中同样执行

第五步：创建Redis集群（创建时Redis里不要有数据）

```
# 进入任一应用
cd /usr/local/redis-cluster/
cd 7001/bin
# 创建主从：前三台机器为主，后三台为从； cluster-replicas 1 表示一主一从
./redis-cli --cluster create 192.168.41.140:7001 192.168.41.141:7001 192.168.41.142:7001 192.168.41.140:7002 //换行
 192.168.41.141:7002 192.168.41.142:7002 --cluster-replicas 1
```

```
[root@mysql-slave1 bin]# cd /usr/local/redis-cluster/
[root@mysql-slave1 redis-cluster]# cd 7001/bin
[root@mysql-slave1 bin]# ./redis-cli --cluster create 192.168.41.140:7001 192.168.41.141:7001 192.168.41.142:7001 //换行
 192.168.41.140:7002 192.168.41.141:7002 192.168.41.142:7002 --cluster-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 192.168.41.141:7002 to 192.168.41.140:7001
Adding replica 192.168.41.142:7002 to 192.168.41.141:7001
Adding replica 192.168.41.140:7002 to 192.168.41.142:7001
>>> Trying to optimize slaves allocation for anti-affinity
[WARNING] Some slaves are in the same host as their master
M: 667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 192.168.41.140:7001
   slots:[0-5460] (5461 slots) master
M: d0f238234a279a65e714a017fb1d91f372c16c26 192.168.41.141:7001
   slots:[5461-10922] (5462 slots) master
M: 149a15e4c836c09b4297090e26d30948c6740a04 192.168.41.142:7001
   slots:[10923-16383] (5461 slots) master
S: 81eb93e3b2639e8ea461d65676f8dc088edc820a 192.168.41.140:7002
   replicates d959677decab18a363424e4089f50d3856f35ecb
S: 438a9e0f199eb77568ae4e0c084e8150b38e8881 192.168.41.141:7002
   replicates d0f238234a279a65e714a017fb1d91f372c16c26
S: a7086771a3c99781172fcd1f6e8e2cdd5162a251 192.168.41.142:7002
   replicates 149a15e4c836c09b4297090e26d30948c6740a04
```

```
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 192.168.41.140:7001)
M: d959677decab18a363424e4089f50d3856f35ecb 192.168.41.140:7001
 slots:[0-5460] (5461 slots) master
 1 additional replica(s)
M: fcd75dbfc87aa90fe9461b75377dc15080e4953a 192.168.41.142:7001
 slots:[10923-16383] (5461 slots) master
 1 additional replica(s)
S: 7557a4a824fe28db55c6c504abca2f5414aa1270 192.168.41.142:7002
 slots: (0 slots) slave
 replicates 149a15e4c836c09b4297090e26d30948c6740a04
S: a07582a6023e9ab9abc2fbdc747a93b4c1e8c814 192.168.41.140:7002
 slots: (0 slots) slave
 replicates d959677decab18a363424e4089f50d3856f35ecb
M: 59c6fe1ffb30f29a9b0789c70ffe373e62a79232 192.168.41.141:7001
 slots:[5461-10922] (5462 slots) master
 1 additional replica(s)
S: 5671b67729da77ed6a390188525f3e2b1a11a3cb 192.168.41.141:7002
 slots: (0 slots) slave
 replicates d0f238234a279a65e714a017fb1d91f372c16c26
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
```

第六步：测试连接

```
[root@localhost bin]# ./redis-cli -h 192.168.41.140 -p 7001 -c
192.168.41.140:7001> set name aaa
-> Redirected to slot [5798] located at 192.168.41.141:7002
OK
192.168.41.141:7002> get name
-> Redirected to slot [5798] located at 192.168.41.141:7001
"aaa"
```

可以看到：数据插入时，发生自动跳转，数据存入到了7002节点。这是 数据的自动路由; 根据键取模槽位数，来定位存储的节点

```
# 查看集群状态
192.168.41.141:7001> 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:2
 cluster_stats_messages_ping_sent:573
 cluster_stats_messages_pong_sent:589
 cluster_stats_messages_meet_sent:4
 cluster_stats_messages_sent:1166
 cluster_stats_messages_ping_received:587
 cluster_stats_messages_pong_received:577
 cluster_stats_messages_meet_received:2
 cluster_stats_messages_received:1166
```

```
# 查看集群中的节点
192.168.41.141:7001> cluster nodes
667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 192.168.41.140:7001@17001 master - 0 1604441488992 1 connected 0-5460
7557a4a824fe28db55c6c504abca2f5414aa1270 192.168.41.142:7002@17002 slave 59c6fe1ffb30f29a9b0789c70ffe373e62a79232 0 1604441487000 6 connected
fcd75dbfc87aa90fe9461b75377dc15080e4953a 192.168.41.142:7001@17001 master - 0 1604441487000 3 connected 10923-16383
81eb93e3b2639e8ea461d65676f8dc088edc820a 192.168.41.140:7002@17002 slave fcd75dbfc87aa90fe9461b75377dc15080e4953a 0 1604441487981 4 connected
5671b67729da77ed6a390188525f3e2b1a11a3cb 192.168.41.141:7002@17002 slave 667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 0 1604441490000 5 connected
59c6fe1ffb30f29a9b0789c70ffe373e62a79232 192.168.41.141:7001@17001 myself,master - 0 1604441490000 2 connected 5461-10922
```

以上，集群部署完成

4、集群扩容

1）添加主节点

先创建192.168.41.143:7001节点 （无数据）

```
#在192.168.41.143主机上
mkdir -p /usr/local/redis-cluster/7001/{etc,data}

cd /usr/local/src/redis-5.0.5/
make install PREFIX=/usr/local/redis-cluster/7001

cp /usr/local/src/redis-5.0.5/redis.conf /usr/local/redis-cluster/7001/redis.conf
```

2）复制7001节点 （无数据） 到 7002节点

```
cd /usr/local/redis-cluster/
cp -r 7001 7002

vim /usr/local/redis-cluster/7002/redis.conf # port 7002
```

3）启动192.168.41.143:7001节点, 并向集群添加该新节点

```
# 启动7007
cd /usr/local/redis-cluster/
cd 7001/bin
./redis-server ../redis.conf

# 进入192.168.41.140节点
cd /usr/local/redis-cluster/7001/bin
# 添加新节点
./redis-cli --cluster add-node 192.168.41.143:7001 192.168.41.140:7001
```

```
[root@localhost bin]# ./redis-cli --cluster add-node 192.168.41.143:7001 192.168.41.140:7001
>>> Adding node 192.168.41.143:7001 to cluster 192.168.41.140:7001
>>> Performing Cluster Check (using node 192.168.41.140:7001)
M: 667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 192.168.41.140:7001
   slots:[0-5460] (5461 slots) master
   1 additional replica(s)
S: 5671b67729da77ed6a390188525f3e2b1a11a3cb 192.168.41.141:7002
   slots: (0 slots) slave
   replicates 667e7a3d0989171ef285f4ca45b7ce5d3d828ebe
S: 7557a4a824fe28db55c6c504abca2f5414aa1270 192.168.41.142:7002
```

```
   slots: (0 slots) slave
   replicates 59c6fe1ffb30f29a9b0789c70ffe373e62a79232
S: 81eb93e3b2639e8ea461d65676f8dc088edc820a 192.168.41.140:7002
   slots: (0 slots) slave
   replicates fcd75dbfc87aa90fe9461b75377dc15080e4953a
M: 59c6fe1ffb30f29a9b0789c70ffe373e62a79232 192.168.41.141:7001
   slots:[5461-10922] (5462 slots) master
   1 additional replica(s)
M: fcd75dbfc87aa90fe9461b75377dc15080e4953a 192.168.41.142:7001
   slots:[10923-16383] (5461 slots) master
   1 additional replica(s)
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
>>> Send CLUSTER MEET to node 192.168.41.143:7001 to make it join the cl
uster.
[OK] New node added correctly.
```

如上，可以看到添加成功，使用命令查看状态：

```
# 注意：-c 表示是以redis集群方式进行连接
# 查看集群状态
[root@localhost bin]# ./redis-cli -h 192.168.41.140 -p 7001 -c
192.168.41.140:7001> cluster info
cluster_state:ok
cluster_slots_assigned:16384
cluster_slots_ok:16384
cluster_slots_pfail:0
cluster_slots_fail:0
 cluster_known_nodes:7
 cluster_size:3
 cluster_current_epoch:6
 cluster_my_epoch:1
 cluster_stats_messages_ping_sent:995
 cluster_stats_messages_pong_sent:1055
 cluster_stats_messages_sent:2050
 cluster_stats_messages_ping_received:1049
 cluster_stats_messages_pong_received:995
 cluster_stats_messages_meet_received:6
 cluster_stats_messages_received:2050
 # 查看集群节点状态
```

```
192.168.41.140:7001> cluster nodes
5671b67729da77ed6a390188525f3e2b1a11a3cb 192.168.41.141:7002@17002 slave 667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 0 1604441910712 5 connected
7557a4a824fe28db55c6c504abca2f5414aa1270 192.168.41.142:7002@17002 slave 59c6fe1ffb30f29a9b0789c70ffe373e62a79232 0 1604441910000 6 connected
81eb93e3b2639e8ea461d65676f8dc088edc820a 192.168.41.140:7002@17002 slave fcd75dbfc87aa90fe9461b75377dc15080e4953a 0 1604441909000 4 connected
59c6fe1ffb30f29a9b0789c70ffe373e62a79232 192.168.41.141:7001@17001 master - 0 1604441908000 2 connected 5461-10922
667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 192.168.41.140:7001@17001 myself,master - 0 1604441910000 1 connected 0-5460
3ceebba0df9422ac20587b2292904601f3065ce3 192.168.41.143:7001@17001 master - 0 1604441911722 0 connected
fcd75dbfc87aa90fe9461b75377dc15080e4953a 192.168.41.142:7001@17001 master - 0 1604441912731 3 connected 10923-16383

```

以上可以看到，新增 master节点 还没有分配槽位

3ceebba0df9422ac20587b2292904601f3065ce3 192.168.41.143:7001@17001 master - 0 1604441911722 0 connected

给刚添加的 结点分配槽

第一步：连接上集群（连接集群中任意一个可用结点都行）

```
[root@mysql-slave1 bin]# ./redis-cli --cluster reshard 192.168.41.143:7001
>>> Performing Cluster Check (using node 192.168.41.143:7001)
M: 3ceebba0df9422ac20587b2292904601f3065ce3 192.168.41.143:7001
   slots: (0 slots) master
M: 667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 192.168.41.140:7001
   slots:[0-5460] (5461 slots) master
   1 additional replica(s)
S: 5671b67729da77ed6a390188525f3e2b1a11a3cb 192.168.41.141:7002
   slots: (0 slots) slave
   replicates 667e7a3d0989171ef285f4ca45b7ce5d3d828ebe
S: 7557a4a824fe28db55c6c504abca2f5414aa1270 192.168.41.142:7002
   slots: (0 slots) slave
   replicates 59c6fe1ffb30f29a9b0789c70ffe373e62a79232
S: 81eb93e3b2639e8ea461d65676f8dc088edc820a 192.168.41.140:7002
   slots: (0 slots) slave
   replicates fcd75dbfc87aa90fe9461b75377dc15080e4953a
M: 59c6fe1ffb30f29a9b0789c70ffe373e62a79232 192.168.41.141:7001
   slots:[5461-10922] (5462 slots) master
   1 additional replica(s)
```

```
M: fcd75dbfc87aa90fe9461b75377dc15080e4953a 192.168.41.142:7001
 slots:[10923-16383] (5461 slots) master
 1 additional replica(s)
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
### 你想分配多少个槽位（1-16384）？
How many slots do you want to move (from 1 to 16384)? 3000
### 你想把槽位分配给谁？ 7007的节点id
What is the receiving node ID? d8f50e50e278730f44d43ee7987f2212342e7ce2
Please enter all the source node IDs.
 Type 'all' to use all the nodes as source nodes for the hash slots.
 Type 'done' once you entered all the source nodes IDs.
### 你想从哪些节点分配槽位： all-所有主节点
Source node #1: all

Ready to move 3000 slots.
 Source nodes:
 M: 59c6fe1ffb30f29a9b0789c70ffe373e62a79232 192.168.41.141:7001
 slots:[5461-10922] (5462 slots) master
 1 additional replica(s)
 M: fcd75dbfc87aa90fe9461b75377dc15080e4953a 192.168.41.142:7001
 slots:[10923-16383] (5461 slots) master
 1 additional replica(s)
 M: 667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 192.168.41.140:7001
 slots:[0-5460] (5461 slots) master
 1 additional replica(s)
 Destination node:
 M: d8f50e50e278730f44d43ee7987f2212342e7ce2 192.168.1.202:7007
 slots: (0 slots) master
 Resharding plan:
 Moving slot 5461 from d0f238234a279a65e714a017fb1d91f372c16c26
 .......

 ### 你想要开始移动槽位吗？ yes
 Do you want to proceed with the proposed reshard plan (yes/no)? yes
 .......
```

```
# 注意：-c 表示是以redis集群方式进行连接
```

```
# 查看集群状态
[root@mysql-slave1 bin]# ./redis-cli -h 127.0.0.1 -p 7001 -c
# 查看集群节点状态
127.0.0.1:7001> cluster nodes
5671b67729da77ed6a390188525f3e2b1a11a3cb 192.168.41.141:7002@17002 slave 667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 0 1604441910712 5 connected
7557a4a824fe28db55c6c504abca2f5414aa1270 192.168.41.142:7002@17002 slave 59c6fe1ffb30f29a9b0789c70ffe373e62a79232 0 1604441910000 6 connected
81eb93e3b2639e8ea461d65676f8dc088edc820a 192.168.41.140:7002@17002 slave fcd75dbfc87aa90fe9461b75377dc15080e4953a 0 1604441909000 4 connected
59c6fe1ffb30f29a9b0789c70ffe373e62a79232 192.168.41.141:7001@17001 master - 0 1604441908000 2 connected 5461-10922
667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 192.168.41.140:7001@17001 myself,master - 0 1604441910000 1 connected 0-5460
3ceebba0df9422ac20587b2292904601f3065ce3 192.168.41.143:7001@17001 master - 0 1599923619032 7 connected 0-998 5461-6461 10923-11921
fcd75dbfc87aa90fe9461b75377dc15080e4953a 192.168.41.142:7001@17001 master - 0 1604441912731 3 connected 10923-16383
```

以上可以看到，7007节点已经成功分配槽位：
3ceebba0df9422ac20587b2292904601f3065ce3 192.168.41.143:7001@17001 master - 0 1599923619032 7 connected 0-998 5461-6461 10923-11921

添加 192.168.41.143:7002 从结点，将 192.168.41.143:7002作为 192.168.41.143:7001的从结点

```
[root@localhost bin]# cd /usr/local/redis-cluster/7001/bin
[root@localhost bin]# ./redis-cli --cluster add-node 192.168.41.143:7002 192.168.41.143:7001 --cluster-slave --cluster-master-id 3ceebba0df9422ac20587b2292904601f3065ce3
>>> Adding node 192.168.41.143:7002 to cluster 192.168.41.143:7001
>>> Performing Cluster Check (using node 192.168.41.143:7001)
M: 3ceebba0df9422ac20587b2292904601f3065ce3 192.168.41.143:7001
   slots:[0-998],[5461-6461],[10923-11921] (2999 slots) master
M: 59c6fe1ffb30f29a9b0789c70ffe373e62a79232 192.168.41.141:7001
   slots:[6462-10922] (4461 slots) master
   1 additional replica(s)
S: 5671b67729da77ed6a390188525f3e2b1a11a3cb 192.168.41.141:7002
   slots: (0 slots) slave
   replicates 667e7a3d0989171ef285f4ca45b7ce5d3d828ebe
S: 81eb93e3b2639e8ea461d65676f8dc088edc820a 192.168.41.140:7002
   slots: (0 slots) slave
   replicates fcd75dbfc87aa90fe9461b75377dc15080e4953a
S: 7557a4a824fe28db55c6c504abca2f5414aa1270 192.168.41.142:7002
```

```
  slots: (0 slots) slave
  replicates 59c6fe1ffb30f29a9b0789c70ffe373e62a79232
M: 667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 192.168.41.140:7001
  slots:[999-5460] (4462 slots) master
  1 additional replica(s)
M: fcd75dbfc87aa90fe9461b75377dc15080e4953a 192.168.41.142:7001
  slots:[11922-16383] (4462 slots) master
  1 additional replica(s)
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
>>> Send CLUSTER MEET to node 192.168.41.143:7002 to make it join the cl
uster.
Waiting for the cluster to join

>>> Configure node as replica of 192.168.41.143:7001.
[OK] New node added correctly.

```

```
[root@localhost bin]# ./redis-cli -h 192.168.41.140 -p 7001 -c
192.168.41.140:7001> cluster nodes
5671b67729da77ed6a390188525f3e2b1a11a3cb 192.168.41.141:7002@17002 slave
667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 0 1604442504269 5 connected
8cf5aae94df5ae8e8932fc8c12b0da75b41c7a40 192.168.41.143:7002@17002 slave
3ceebba0df9422ac20587b2292904601f3065ce3 0 1604442502000 7 connected
7557a4a824fe28db55c6c504abca2f5414aa1270 192.168.41.142:7002@17002 slave
59c6fe1ffb30f29a9b0789c70ffe373e62a79232 0 1604442503000 6 connected
81eb93e3b2639e8ea461d65676f8dc088edc820a 192.168.41.140:7002@17002 slave
fcd75dbfc87aa90fe9461b75377dc15080e4953a 0 1604442500231 4 connected
59c6fe1ffb30f29a9b0789c70ffe373e62a79232 192.168.41.141:7001@17001 master
- 0 1604442505276 2 connected 6462-10922
667e7a3d0989171ef285f4ca45b7ce5d3d828ebe 192.168.41.140:7001@17001
myself,master - 0 1604442501000 1 connected 999-5460
3ceebba0df9422ac20587b2292904601f3065ce3 192.168.41.143:7001@17001 master
- 0 1604442503261 7 connected 0-998 5461-6461 10923-11921
fcd75dbfc87aa90fe9461b75377dc15080e4953a 192.168.41.142:7001@17001 maste
r - 0 1604442502251 3 connected 11922-16383

```

以上，集群扩容完成

5、JAVA客户端调用

引入依赖

```
<dependency>
 <groupId>redis.clients</groupId>
 <artifactId>jedis</artifactId>
 <version>2.8.1</version>
</dependency>
```

通过JedisCluster向RedisCluster添加数据和取出数据

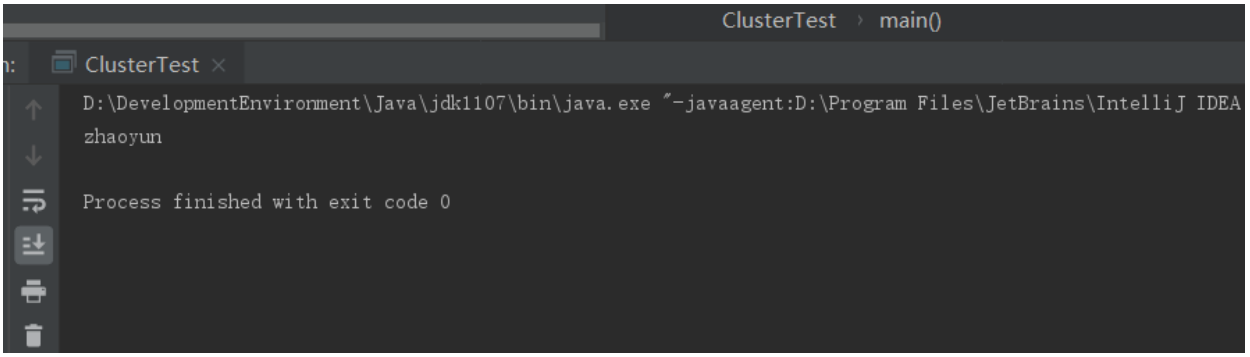
```
import redis.clients.jedis.HostAndPort;
import redis.clients.jedis.JedisCluster;
import redis.clients.jedis.JedisPoolConfig;

import java.util.HashSet;
import java.util.Set;

public class ClusterTest {
 public static void main(String[] args) {

  JedisPoolConfig config = new JedisPoolConfig();
  Set<HostAndPort> jedisClusterNode = new HashSet<>();
  jedisClusterNode.add(new HostAndPort("192.168.41.140", 7001));
  jedisClusterNode.add(new HostAndPort("192.168.41.140", 7002));
  jedisClusterNode.add(new HostAndPort("192.168.41.141", 7001));
  jedisClusterNode.add(new HostAndPort("192.168.41.141", 7002));
  jedisClusterNode.add(new HostAndPort("192.168.41.142", 7001));
  jedisClusterNode.add(new HostAndPort("192.168.41.142", 7002));
  jedisClusterNode.add(new HostAndPort("192.168.41.143", 7001));
  jedisClusterNode.add(new HostAndPort("192.168.41.143", 7002));
  JedisCluster jcd = new JedisCluster(jedisClusterNode, config);
  jcd.set("name:001", "zhaoyun");
  String value = jcd.get("name:001");
  System.out.println(value);
  }
}
```

运行结果：


ClusterTest › main()
ClusterTest
D:\DevelopmentEnvironment\Java\jdk1107\bin\java.exe "-javaagent:D:\Program Files\JetBrains\IntelliJ IDEA
zhaoyun
Process finished with exit code 0