## ****安装需要的依赖包****

[root@localhost ~]# yum install gcc gcc-c++ kernel-devel automake autoconf libtool make tcl ruby rubygems  openssl -y

## 2. ****下载redis并安装****

[root@localhost src]# cd /usr/local/

[root@localhost local]# wget http://download.redis.io/releases/redis-4.0.1.tar.gz

[root@localhost local]# tar xzf redis-4.0.1.tar.gz

[root@localhost local]# cd redis-4.0.1

[root@localhost redis-4.0.1]# make

## 3. 创建集群需要的目录

mkdir -p /usr/local/cluster

cd /usr/local/cluster

mkdir –p /usr/local/cluster/7000

mkdir –p /usr/local/cluster/7001

mkdir –p /usr/local/cluster/7002

mkdir –p /usr/local/cluster/7003

mkdir –p /usr/local/cluster/7004

mkdir –p /usr/local/cluster/7005

## 4. 修改配置文件redis.conf

port 7000（7002,7003,7004,7005）

daemonize yes

dbfilename dump-700X.rdb

logfile "log-700X.txt"

cluster-enabled yes

cluster-config-file nodes\_700X.conf

cluster-node-timeout 5000

cluster-migration-barrier 1

cluster-require-full-coverage no

appendonly yes

appendfilename "appendonly\_700X.aof"

pidfile /var/run/redis\_700X.pid

bind 0.0.0.0

masterauth 123456

requirepass 123456

这个配置文件分别拷贝到7000/7001/7002/7003/7004/7005节点目录。

## 5. 分别启动这6个redis实例

/usr/local/redis-4.0.1/src/redis-server ./7000/redis.conf

/usr/local/redis-4.0.1/src/redis-server ./7001/redis.conf

/usr/local/redis-4.0.1/src/redis-server ./7002/redis.conf

/usr/local/redis-4.0.1/src/redis-server ./7003/redis.conf

/usr/local/redis-4.0.1/src/redis-server ./7004/redis.conf

/usr/local/redis-4.0.1/src/redis-server ./7005/redis.conf

并查看是否成功：ps -ef|grep redis

## 6. 执行redis的创建集群命令创建集群

[root@localhost cluster]# cd /usr/local/redis-4.0.1/src

[root@localhost src]# ./redis-trib.rb  create --replicas 1 127.0.0.1:7000 127.0.0.1:7001 127.0.0.1:7002 127.0.0.1:7003 127.0.0.1:7004 127.0.0.1:7005

到这一步因为前面第1步装了依赖包，未提示ruby和rubygems的错误，但还是会报错，提示不能加载redis，是因为缺少redis和ruby的接口，使用gem 安装  
错误内容：  
/usr/lib/ruby/site\_ruby/1.8/rubygems/custom\_require.rb:31:in `gem\_original\_require': no such file to load -- redis (LoadError)  
        from /usr/lib/ruby/site\_ruby/1.8/rubygems/custom\_require.rb:31:in `require'  
        from ./redis-trib.rb:25  
解决：gem install redis

还可能会提示ruby版本太低，需要安装高版本。

wget https://cache.ruby-china.org/pub/ruby/ruby-2.3.1.tar.gz

tar xzvf ruby-2.3.1.tar.gz

cd ruby-2.3.1

./configure -prefix=/usr/local/ruby

make && make install

可能还会提示缺啥openssl，解决办法参考：

<http://blog.csdn.net/huang930528/article/details/51027915>

【注意】--replicas 1代表1个slave的意思，6379是客户端的连接端口，6379+10000=16379是集群内部的通信端口，必须要开启。

## 7. 重新执行创建集群的命令

。。。。。。。

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

## 8. ****测试集群****

[root@iZuf6h90t5awxwr3uqp6f0Z src]# ./redis-cli -c -p 7000

127.0.0.1:7000> set username xjs

-> Redirected to slot [14315] located at 127.0.0.1:7002

OK

[root@iZuf6h90t5awxwr3uqp6f0Z ~]# /usr/local/redis-4.0.1/src/redis-cli -c -p 7005

127.0.0.1:7005> get username

-> Redirected to slot [14315] located at 127.0.0.1:7002

"xjs"

127.0.0.1:7001> cluster node #查看集群的状态

(error) ERR Wrong CLUSTER subcommand or number of arguments

127.0.0.1:7001> cluster nodes

3c4b6483135cc38650a31c331eecc997b6e88190 127.0.0.1:7002@17002 master - 0 1504779228400 3 connected 10923-16383

c25ea8b61a8aa1d3ffefa4de38ac841f8cc6c358 127.0.0.1:7004@17004 slave 57f4e347168da74ca684ebf36a28a2fc09083ac2 0 1504779228000 5 connected

fdc803212f1cf652f24ff829e577cd3d1910287c 127.0.0.1:7003@17003 master - 0 1504779229000 7 connected 0-5460

57f4e347168da74ca684ebf36a28a2fc09083ac2 127.0.0.1:7001@17001 myself,master - 0 1504779227000 2 connected 5461-10922

458c8db15585fa4ecc487a0d532990293a3451a2 127.0.0.1:7000@17000 slave fdc803212f1cf652f24ff829e577cd3d1910287c 0 1504779228600 7 connected

7fff66e34ca2087602f46dd49d3e9dd20d7658e7 127.0.0.1:7005@17005 slave 3c4b6483135cc38650a31c331eecc997b6e88190 0 1504779229401 6 connected

127.0.0.1:7001>

参考：

<http://blog.csdn.net/nuli888/article/details/52134117>

<http://www.cnblogs.com/feiyun126/p/7248989.html>

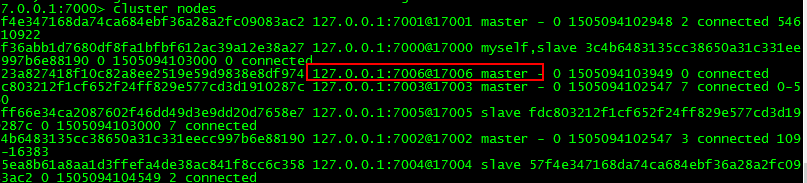
## 9.运维相关

参考：<https://redis.io/topics/cluster-tutorial>

### 添加一个master节点7006：

1. 启动一个新节点./redis-server /usr/local/cluster/7006/redis.conf
2. ./redis-trib.rb add-node 127.0.0.1:7006 127.0.0.1:7000

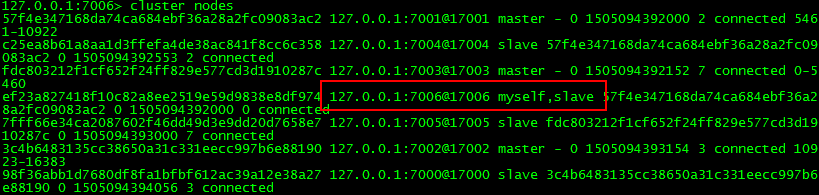
127.0.0.1:7006代表新的节点的ip和端口，127.0.0.1:7000代表老的集群



### 修改7006为7001的slave：

（1）进入7006：./redis-cli -c -p 7006

（2）cluster replicate 57f4e347168da74ca684ebf36a28a2fc09083ac2（7001的id）



### 删除节点7006：

（1）./redis-trib.rb del-node 127.0.0.1:7000 ef23a827418f10c82a8ee2519e59d9838e8df974

127.0.0.1:7000代表集群的ip和端口，ef23a827418f10c82a8ee2519e59d9838e8df974是要删除的节点的id

### 添加一个slave节点7006：

1. 启动7006节点
2. ./redis-trib.rb add-node --slave --master-id 3c4b6483135cc38650a31c331eecc997b6e88190 127.0.0.1:7006 127.0.0.1:7000

--master-id 3c4b6483135cc38650a31c331eecc997b6e88190是master的id

127.0.0.1:7006是新加入的节点的ip和端口

127.0.0.1:7000是集群的IP和端口

## 10.实践相关

### 三主三从改进成三主四从

三主三从的缺点：如果一个node的主节点和从节点同时挂掉，集群不可用

改进方案：

1、三主六从，每个node设置多个从节点，这样会造成内存资源或者服务器资源的浪费。

2、给其中一个主，配置多个从

cluster-migration-barrier 参数

#一个主节点在拥有多少个好的从节点的时候就要割让一个从节点出来给其他没有从节点或者从节点挂掉的主节点

### 一组节点挂掉不影响别的节点

# By default Redis Cluster nodes stop accepting queries if they detect there

# is at least an hash slot uncovered (no available node is serving it).

# This way if the cluster is partially down (for example a range of hash slots

# are no longer covered) all the cluster becomes, eventually, unavailable.

# It automatically returns available as soon as all the slots are covered again.

# However sometimes you want the subset of the cluster which is working,

# to continue to accept queries for the part of the key space that is still

# covered. In order to do so, just set the cluster-require-full-coverage

# option to no.

cluster-require-full-coverage no

### 集群设置密码

修改redis.conf:

masterauth 123456

requirepass 123456

【注意】，不能先设置密码再创建集群，而是先用无密码的方式创建集群，然后再挨个节点设置密码然后重启。

### Jedis客户端连接带密码的Redis集群

（1）Redis集群创建的时候，需要用真实的IP，而不能用127.0.0.1。原因是Jedis在初始化的时候会向集群发送cluser slots命令来查询各个机器上的slot，但是同时用命令中返回的host和ip来创建JedisPool，如果用127.0.0.1，那么创建的JedisPool去获取连接的时候会去连接127.0.0.1而不是实际的集群。

JedisClusterInfoCache：

public void discoverClusterNodesAndSlots(Jedis jedis) {

w.lock();

try {

reset();

List<Object> slots = jedis.clusterSlots();

for (Object slotInfoObj : slots) {

List<Object> slotInfo = (List<Object>) slotInfoObj;

。。。

// hostInfos

int size = slotInfo.size();

for (int i = MASTER\_NODE\_INDEX; i < size; i++) {

List<Object> hostInfos = (List<Object>) slotInfo.get(i);

。。。。

HostAndPort targetNode = generateHostAndPort(hostInfos);

setupNodeIfNotExist(targetNode);

if (i == MASTER\_NODE\_INDEX) {

assignSlotsToNode(slotNums, targetNode);

}

}

}

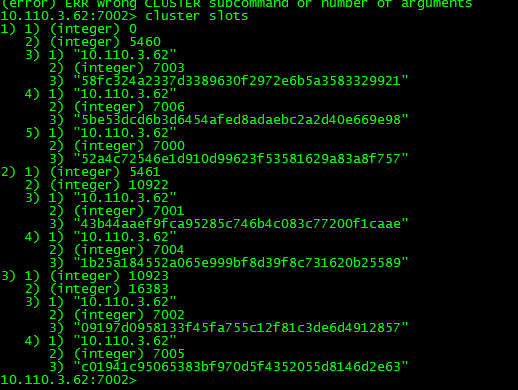
} finally {

w.unlock();

}

}

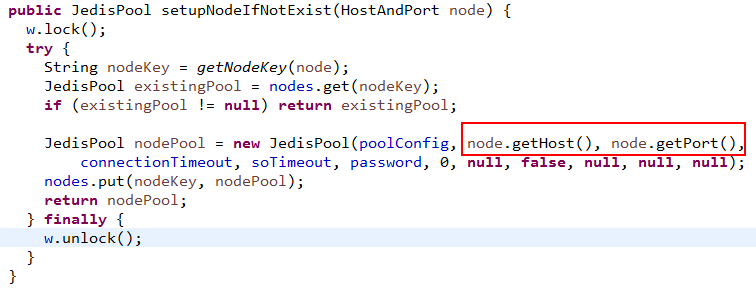
jedis.clusterSlots();向服务端发送了一个cluster slots命令，输出的结果是类似于：



然后就是解析返回结果，

generateHostAndPort(hostInfos)：根据返回信息创建host和port

setupNodeIfNotExist(targetNode);创建JedisPool，同时缓存起来，



### gem install redis 报错：Unable to require openssl。。。。

参考：<http://blog.csdn.net/huang930528/article/details/51027915>

**问题**：  
前面已经在/usr/local/src安装了ruby-2.3.0.tar.gz、rubygems-2.4.2.tar.gz。  
在配置 redis-3.1.1 群集中，使用gem install 安装 ruby redis 接口时报：

**[plain]** [view plain](http://blog.csdn.net/huang930528/article/details/51027915) [copy](http://blog.csdn.net/huang930528/article/details/51027915)

1. # [root@localhost src]# gem install redis --version 3.0.0
2. # # 由于源的原因，可能下载失败，就手动下载下来安装
3. [root@localhost src]# gem install /usr/local/src/redis-3.2.1.gem
4. ERROR: While executing gem ... (Gem::Exception)
5. Unable to require openssl, install OpenSSL and rebuild ruby (preferred) or use non-HTTPS sources

**原因**：  
缺少openssl，需要安装openssl包，我现在这里来安装一个openssl-1.0.1s.tar.gz（下载链接：http://www.openssl.org/source/）  
  
**解决方法步骤**：  
1.解压在/usr/local/src目录下，进入/usr/local/src/openssl-1.0.1s目录准备安装：

**[plain]** [view plain](http://blog.csdn.net/huang930528/article/details/51027915) [copy](http://blog.csdn.net/huang930528/article/details/51027915)

1. [root@localhost src]# tar -xzvf openssl-1.0.1s.tar.gz
2. [root@localhost src]# cd openssl-1.0.1s
3. [root@localhost openssl-1.0.1s]# ./config -fPIC --prefix=/usr/local/openssl enable-shared
4. [root@localhost openssl-1.0.1s]# ./config -t
5. [root@localhost openssl-1.0.1s]# make && make install

openssl的配置文件必须要配置-fPIC参数，如果没有该参数下面的安装中会出现问题！  
安装完成，可以检测一下是否安装成功：

**[plain]** [view plain](http://blog.csdn.net/huang930528/article/details/51027915) [copy](http://blog.csdn.net/huang930528/article/details/51027915)

1. [root@localhost openssl-1.0.1s]# openssl version

2.进入ruby源码[/usr/local/src/ruby-2.3.0]目录下的ext/openssl 目录：

**[plain]** [view plain](http://blog.csdn.net/huang930528/article/details/51027915) [copy](http://blog.csdn.net/huang930528/article/details/51027915)

1. [root@localhost openssl-1.0.1s]# cd ../ruby-2.3.0
2. [root@localhost ruby-2.3.0]# cd ext/openssl
3. [root@localhost openssl]# ruby extconf.rb
4. checking for t\_open() in -lnsl... no
5. checking for socket() in -lsocket... no
6. checking for assert.h... yes
7. checking for openssl/ssl.h... no

提示没有找到ssl.h, 因为出现了错误：openssl/ssl.h：没有那个文件或目录

**[plain]** [view plain](http://blog.csdn.net/huang930528/article/details/51027915) [copy](http://blog.csdn.net/huang930528/article/details/51027915)

1. [root@localhost openssl]# ruby extconf.rb --with-openssl-include=/usr/local/openssl/include/ --with-openssl-lib=/usr/local/openssl/lib
2. checking for t\_open() in -lnsl... no
3. checking for socket() in -lsocket... no
4. checking for assert.h... yes
5. checking for openssl/ssl.h... yes
6. .............................................中间略.............................................
7. checking for EVP\_CTRL\_GCM\_GET\_TAG in openssl/evp.h... yes
8. creating extconf.h
9. creating Makefile

成功。  
  
3.接下来并且将ruby 源码目录下的include目录软链接到 / 目录下：

**[plain]** [view plain](http://blog.csdn.net/huang930528/article/details/51027915) [copy](http://blog.csdn.net/huang930528/article/details/51027915)

1. [root@localhost openssl]# ln -s /usr/local/src/ruby-2.2.3/include /

接着再执行make，如果在配置openssl时没有-fPIC参数时就会出现如下错误

**[plain]** [view plain](http://blog.csdn.net/huang930528/article/details/51027915) [copy](http://blog.csdn.net/huang930528/article/details/51027915)

1. [root@localhost openssl]# make
2. gcc -shared -o openssl.so ossl\_x509.o ossl\_x509cert.o ossl\_x509crl.o ossl\_pkcs7.o ossl\_cipher.o ossl\_pkey\_dh.o ossl\_ssl\_session.o ossl\_x509attr.o ossl\_x509ext.o ossl\_ocsp.o ossl\_asn1.o ossl\_x509name.o ossl\_x509revoked.o ossl\_rand.o ossl\_pkcs12.o ossl\_x509store.o ossl\_ssl.o ossl.o ossl\_digest.o ossl\_config.o ossl\_pkcs5.o ossl\_bio.o ossl\_pkey\_dsa.o ossl\_pkey.o ossl\_x509req.o ossl\_pkey\_ec.o ossl\_hmac.o openssl\_missing.o ossl\_ns\_spki.o ossl\_pkey\_rsa.o ossl\_engine.o ossl\_bn.o -L. -L/usr/local/ruby/lib -Wl,-R/usr/local/ruby/lib -L/usr/local/ssl/lib -Wl,-R/usr/local/ssl/lib -L. -rdynamic -Wl,-export-dynamic -lssl -lcrypto -lrt -ldl -lcrypt -lm -lc
3. /usr/bin/ld: /usr/local/ssl/lib/libssl.a(s2\_meth.o): relocation R\_X86\_64\_32 against `a local symbol'
4. can not be used when making a shared object; recompile with -fPIC
5. /usr/local/ssl/lib/libssl.a: could not read symbols: Bad value
6. collect2: ld returned 1 exit status
7. make: \*\*\* [openssl.so] Error 1

有配置-fPIC参数时就继续走下去：

**[plain]** [view plain](http://blog.csdn.net/huang930528/article/details/51027915) [copy](http://blog.csdn.net/huang930528/article/details/51027915)

1. [root@localhost openssl]# make
2. compiling ossl\_pkey\_dsa.c
3. compiling openssl\_missing.c
4. compiling ossl\_x509name.c
5. compiling ossl\_pkey\_rsa.c
6. .............................................中间略.............................................
7. compiling ossl\_bn.c
8. compiling ossl.c
9. linking shared-object openssl.so

**[plain]** [view plain](http://blog.csdn.net/huang930528/article/details/51027915) [copy](http://blog.csdn.net/huang930528/article/details/51027915)

1. [root@localhost openssl]# make install
2. /usr/bin/install -c -m 0755 openssl.so /usr/local/ruby/lib/ruby/site\_ruby/2.2.0/x86\_64-linux
3. installing default openssl libraries

成功。  
  
4.回过头来，这时候我们已经把配置集群遇到的各种问题已经解决好了，再次使用gem install 安装 ruby redis 接口：

**[plain]** [view plain](http://blog.csdn.net/huang930528/article/details/51027915) [copy](http://blog.csdn.net/huang930528/article/details/51027915)

1. # [root@localhost src]# gem install redis --version 3.0.0
2. # # 由于源的原因，可能下载失败，就手动下载下来安装
3. [root@localhost src]# gem install /usr/local/src/redis-3.2.1.gem
4. Successfully installed redis-3.2.1
5. Parsing documentation for redis-3.2.1
6. Installing ri documentation for redis-3.2.1
7. Done installing documentation for redis after 0 seconds
8. WARNING:  Unable to pull data from 'https://rubygems.org/': SSL\_connect returned=1 errno=0 state=error: certificate verify failed (https://rubygems.org/specs.4.8.gz)
9. 1 gem installed

可以看到这时ruby redis接口已经安装成功了；