# 安装redis

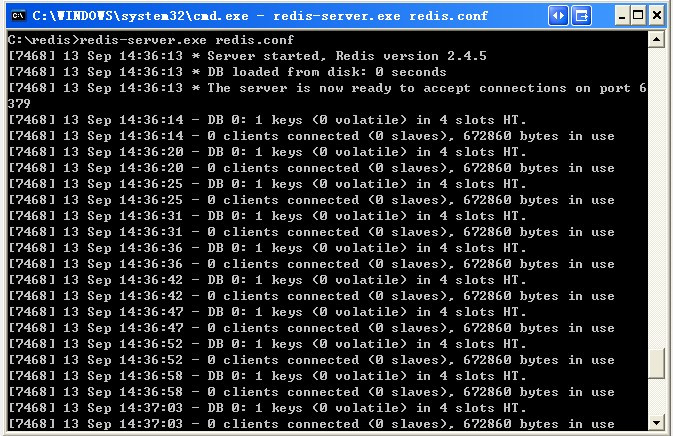
## Windows下安装redis

**下载地址：**[https://github.com/dmajkic/redis/downloads](https://github.com/dmajkic/redis/downloads" \t "_blank)

下载到的Redis支持32bit和64bit。根据自己实际情况选择，将64bit的内容cp到自定义盘符安装目录取名redis。 如 C:\reids

打开一个cmd窗口 使用cd命令切换目录到 C:\redis 运行 **redis-server.exe redis.conf** 。

如果想方便的话，可以把redis的路径加到系统的环境变量里，这样就省得再输路径了，后面的那个redis.conf可以省略，如果省略，会启用默认的。输入之后，会显示如下界面：



这时候另启一个cmd窗口，原来的不要关闭，不然就无法访问服务端了。

切换到redis目录下运行 **redis-cli.exe -h 127.0.0.1 -p 6379** 。

设置键值对 **set myKey abc**

取出键值对 **get myKey**

## Linux下安装redis

**下载地址：**[http://www.redis.net.cn/download/](http://www.redis.net.cn/download/" \t "_blank)，下载最新文档版本。

1. $ wget http://download.redis.io/releases/redis-2.8.17.tar.gz
2. $ tar xzf redis-2.8.17.tar.gz
3. $ cd redis-2.8.17
4. $ make

make完后 redis-2.8.17目录下会出现编译后的redis服务程序redis-server,还有用于测试的客户端程序redis-cli

下面启动redis服务.

1. $ ./redis-server

注意这种方式启动redis 使用的是默认配置。也可以通过启动参数告诉redis使用指定配置文件使用下面命令启动。

1. $ ./redis-server redis.conf

redis.conf是一个默认的配置文件。我们可以根据需要使用自己的配置文件。

启动redis服务进程后，就可以使用测试客户端程序redis-cli和redis服务交互了。 比如：

1. $ ./redis-cli
2. redis> set foo bar
3. OK
4. redis> get foo
5. "bar"

拷贝文件redis-benchmark redis-cli redis-server redis-check-aof redis.conf redis-trib.rb

redis-check-rdb redis-sentinel 到/usr/local/redis目录下

编辑/etc/profile 设置环境变量 export PATH=$PATH:/usr/local/redis

执行命令 Source export PATH=$PATH:/usr/local/redis 立即生效。（只是对当前shell有效、重启后才正式生效）。

# 集群搭建

## 1.介绍安装环境与版本

Redis：redis-4.0.2。

Centos 7：一台虚拟机模拟6个节点（redis要求一个集群至少需要6个节点）。

## 2.创建 Redis 节点

首先进入目录/usr/local/redis创建cluster\_conf文件夹

mkdir cluster\_conf

将redis.conf 拷贝6份到cluster\_conf目录中，并重命名为redis\_7000.conf、redis\_7001.conf、redis\_7002.conf…redis\_7006.conf。

cp redis.conf cluster\_conf/redis\_7000.conf

cp redis.conf cluster\_conf/redis\_7001.conf

cp redis.conf cluster\_conf/redis\_7002.conf

cp redis.conf cluster\_conf/redis\_7003.conf

cp redis.conf cluster\_conf/redis\_7004.conf

cp redis.conf cluster\_conf/redis\_7005.conf

分别修改这六个配置文件，修改如下内容

port 7000 //端口7000,7002,7003

bind 本机ip //默认ip为127.0.0.1 需要改为其他节点机器可访问的ip 否则创建集群时无法访问对应的端口，无法创建集群

daemonize yes //redis后台运行

pidfile /var/run/redis\_7000.pid //pidfile文件对应7000,7001,7002

cluster-enabled yes //开启集群 把注释#去掉

cluster-config-file nodes\_7000.conf //集群的配置 配置文件首次启动自动生成 7000,7001,7002

cluster-node-timeout 15000 //请求超时 默认15秒，可自行设置

appendonly yes //aof日志开启 有需要就开启，它会每次写操作都记录一条日志

## 3.启动各个节点

redis-server cluster\_conf/redis\_7000.conf

redis-server cluster\_conf/redis\_7001.conf

redis-server cluster\_conf/redis\_7002.conf

redis-server cluster\_conf/redis\_7003.conf

redis-server cluster\_conf/redis\_7004.conf

redis-server cluster\_conf/redis\_7005.conf

## 4.检查 redis 启动情况

[root@localhost redis]# ps -ef | grep redis

[root@localhost redis]# ps -ef | grep redis

root 11555 1 0 14:41 ? 00:00:15 redis-server 0.0.0.0:7000 [cluster]

root 11557 1 0 14:41 ? 00:00:14 redis-server 0.0.0.0:7001 [cluster]

root 11565 1 0 14:41 ? 00:00:14 redis-server 0.0.0.0:7002 [cluster]

root 11570 1 0 14:41 ? 00:00:15 redis-server 0.0.0.0:7003 [cluster]

root 11572 1 0 14:41 ? 00:00:16 redis-server 0.0.0.0:7004 [cluster]

root 11577 1 0 14:41 ? 00:00:15 redis-server 0.0.0.0:7005 [cluster]

[root@localhost redis]# netstat -tnlp | grep redis

tcp 0 0 0.0.0.0:17002 0.0.0.0:\* LISTEN 11565/redis-server

tcp 0 0 0.0.0.0:17003 0.0.0.0:\* LISTEN 11570/redis-server

tcp 0 0 0.0.0.0:17004 0.0.0.0:\* LISTEN 11572/redis-server

tcp 0 0 0.0.0.0:17005 0.0.0.0:\* LISTEN 11577/redis-server

tcp 0 0 0.0.0.0:7000 0.0.0.0:\* LISTEN 11555/redis-server

tcp 0 0 0.0.0.0:7001 0.0.0.0:\* LISTEN 11557/redis-server

tcp 0 0 0.0.0.0:7002 0.0.0.0:\* LISTEN 11565/redis-server

tcp 0 0 0.0.0.0:7003 0.0.0.0:\* LISTEN 11570/redis-server

tcp 0 0 0.0.0.0:7004 0.0.0.0:\* LISTEN 11572/redis-server

tcp 0 0 0.0.0.0:7005 0.0.0.0:\* LISTEN 11577/redis-server

tcp 0 0 0.0.0.0:17000 0.0.0.0:\* LISTEN 11555/redis-server

tcp 0 0 0.0.0.0:17001 0.0.0.0:\* LISTEN 11557/redis-server

## 5.创建集群

Redis 官方提供了 redis-trib.rb 这个工具，就在解压目录的 src 目录中，我们已将其拷贝到/usr/local/redis/目录中。

[root@localhost redis]#redis-trib.rb create --replicas 1 192.168.11.41:7000 192.168.11.41:7001 192.168.11.41:7002 192.168.11.41:7003 192.168.11.41:7004 192.168.11.41:7005

节点ip:port

等等，出错了！这个工具是用 ruby 实现的，所以需要安装ruby。

使用yum安装ruby

安装命令如下：

yum –y install ruby ruby-devel rubygems rpm-build

使用源码安装ruby

下载地址：<http://www.ruby-lang.org/en/downloads>

下载最新版的：ruby-2.4.2.tar.gz(redis要求最低版本为2.2.2)

解压、make、make install

cd ruby-2.4.2

./configure

make

make install

安装完ruby

执行命令：gem install redis

gem install redis时报错

#gem install redis  
ERROR:  Loading command: install (LoadError)  
    no such file to load -- zlib  
ERROR:  While executing gem ... (NameError)  
    uninitialized constant Gem::Commands::InstallCommand

解决办法是：

进入ruby源码文件夹

安装ruby自身提供的zlib包

#cd ext/zlib  
#ruby ./extconf.rb  
#make  
#make install

问题解决

注意：若遇到 make: \*\*\* 没有规则可以创建“zlib.o”需要的目标“/include/ruby.h”

打开makefile 配置变量top\_srcdir=/opt/software/ruby/ruby-2.4.2 （源代码路径）或 make top\_srcdir=/opt/software/ruby/ruby-2.4.2

再次执行命令：gem install redis

有报错不过这次错误为：

ERROR: While executing gem ... (Gem::Exception)

Unable to require openssl, install OpenSSL and rebuild ruby (preferred) or use non-HTTPS sources

or 前面让安装OpenSSL 重新build ruby，or后面是让使用 http 别用https。

先看一下 $openssl version 是不是安装了OpenSSL

OpenSSL 0.9.8zh 14 Jan 2016 恩，有版本，还挺新的

那就更换源的地址

删除 https 源

$gem source -r https://rubygems.org/ to remove

添加 http 源

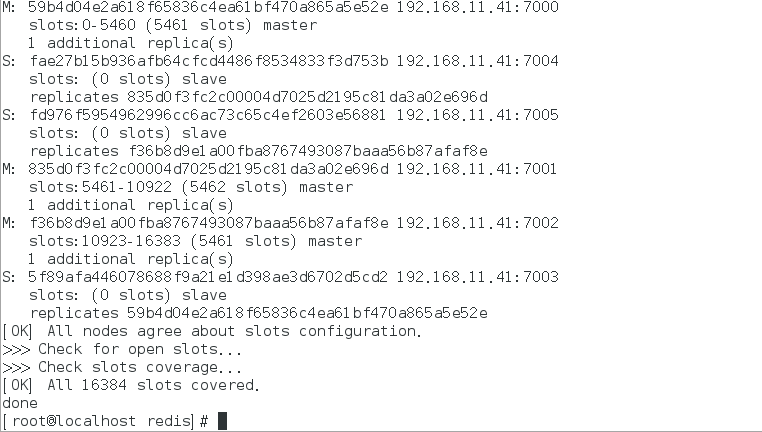
$gem source -a http://rubygems.org/ to read

确保添加正确

之后再运行 redis-trib.rb 命令，会出现如下提示：



输入 yes 即可，然后出现如下内容，说明安装成功。



## 6.集群验证

打开窗口连接到7002端口：

redis-cli -h 192.168.11.41 -c -p 7002  ,加参数 -C 可连接到集群，因为上面 redis.conf 将 bind 改为了ip地址，所以 -h 参数不可以省略。

[root@localhost redis]# redis-cli -h 192.168.11.41 -c -p 7002

192.168.11.41:7002> set name b

-> Redirected to slot [5798] located at 192.168.11.41:7001

OK

192.168.11.41:7001> get name

"b"

再打开一个窗口连接到7005端口：

[root@localhost redis]# redis-cli -h 192.168.11.41 -c -p 7005

192.168.11.41:7005> get name

-> Redirected to slot [5798] located at 192.168.11.41:7001

"b"

说明集群运作正常。

简单说一下原理

redis cluster在设计的时候，就考虑到了去中心化，去中间件，也就是说，集群中的每个节点都是平等的关系，都是对等的，每个节点都保存各自的数据和整个集群的状态。每个节点都和其他所有节点连接，而且这些连接保持活跃，这样就保证了我们只需要连接集群中的任意一个节点，就可以获取到其他节点的数据。

Redis 集群没有并使用传统的一致性哈希来分配数据，而是采用另外一种叫做哈希槽 (hash slot)的方式来分配的。redis cluster 默认分配了 16384 个slot，当我们set一个key 时，会用CRC16算法来取模得到所属的slot，然后将这个key 分到哈希槽区间的节点上，具体算法就是：CRC16(key) % 16384。所以我们在测试的时候看到set 和 get 的时候，直接跳转到了7000端口的节点。

Redis 集群会把数据存在一个 master 节点，然后在这个 master 和其对应的salve 之间进行数据同步。当读取数据时，也根据一致性哈希算法到对应的 master 节点获取数据。只有当一个master 挂掉之后，才会启动一个对应的 salve 节点，充当 master 。

需要注意的是：必须要3个或以上的主节点，否则在创建集群时会失败，并且当存活的主节点数小于总节点数的一半时，整个集群就无法提供服务了。