

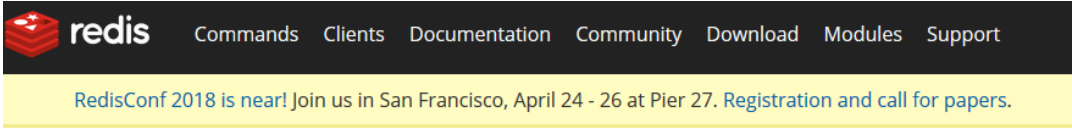
Zhan

博客园 首页 新随笔 联系 订阅 管理

## Linux安装redis和部署

### 第一步：下载安装包

访问<https://redis.io/download> 到官网进行下载。这里下载最新的4.0版本。





## Download

Redis uses a standard practice for its versioning: **major.minor.patchlevel**. An even **minor** marks a **stable** release, like 1.2, 2.0, 2.2, 2.4, 2.6, 2.8. Odd minors are used for **unstable** releases, for example 2.9.x releases are the unstable versions of what will be Redis 3.0 once stable.

### Stable (4.0)


Redis 4.0 was released as GA in July 2017. Contains several big improvements: a modules system, much better replication (PSYNC2), improvements to eviction policies, threaded DEL/FLUSH, mixed RDB+AOF format, Raspberry Pi support as primary platform, the new MEMORY command, Redis Cluster support for Nat/Docker, active memory defragmentation, memory usage and performance improvements, much faster Redis Cluster key creation, many other smaller features and a number of behavior fixed.

Release notes

Download 4.0.9


### Unstable

This is where all the development happens. Only for hard-core hackers. Use only if you need to test the latest features or performance improvements. This is going to be the next Redis release in a few months.

Download unstable

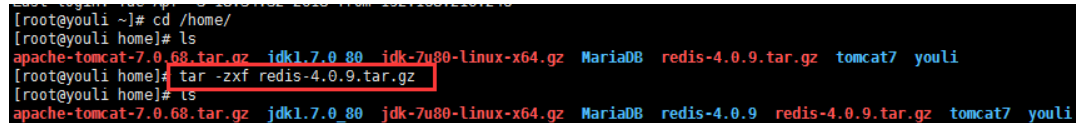
### Docker

It is possible to get Docker images of Redis from the Docker Hub. Multiple versions are available, usually updated in a short time after a new release is available.

Download

### 第二步：安装

- 1.通过远程管理工具，将压缩包拷贝到Linux服务器中，执行解压操作



tar -zxvf redis-4.0.9.tar.gz

- 2.进入解压文件目录使用make对解压的Redis文件进行编译

图示：这里因为我redis的安装目录在 /usr/local/ 目录下，因此会有如下操作



### 公告

昵称： 达兔哥  
园龄： 3年6个月  
粉丝： 35  
关注： 14  
+加关注

<	2020年2月				
日	一	二	三	四	
26	27	28	29	30	
2	3	4	5	6	
9	10	11	12	13	
16	17	18	19	20	
23	24	25	26	27	
1	2	3	4	5	

### 搜索

### 我的标签

Filter(1)

Java代码重构简化(1)

Java跨域请求(1)

Maven(1)

Redis(1)

### 随笔分类

Cookie(1)

Docker(1)

EasyUI学习笔记

Eclipse报错异常(1)

ELK

Exception(2)

FastDfs

Filter过滤器相关(3)

```
[root@youli local]# cd redis/
[root@youli redis]# make
cd src && make all
make[1]: Entering directory `/usr/local/redis/src'
  CC Makefile.dep
make[1]: Leaving directory `/usr/local/redis/src'
make[1]: Entering directory `/usr/local/redis/src'
rm -rf redis-server redis-sentinel redis-cli redis-benchmark redis-check-rdb redis-check-aof *.o *.gcda *.gcnv *.gcov redis.info lcov
-html Makefile.dep dict-benchmark
(cd ../deps && make distclean)
make[2]: Entering directory `/usr/local/redis/deps'
(cd hiredis && make clean) > /dev/null || true
(cd linenoise && make clean) > /dev/null || true
(cd lua && make clean) > /dev/null || true
(cd jemalloc && [ -f Makefile ] && make distclean) > /dev/null || true
(rm -f .make-*)
make[2]: Leaving directory `/usr/local/redis/deps'
(rm -f .make-*)
echo STD=-std=c99 -pedantic -DREDIS_STATIC='' >> .make-settings
echo WARN=-Wall -W -Wno-missing-field-initializers >> .make-settings
echo OPT=-O2 >> .make-settings
echo MALLOC=jemalloc >> .make-settings
echo CFLAGS= >> .make-settings
echo LDFLAGS= >> .make-settings
echo REDIS_CFLAGS= >> .make-settings
echo REDIS_LDFLAGS= >> .make-settings
echo PREV_FINAL_CFLAGS=-std=c99 -pedantic -DREDIS_STATIC='' -Wall -W -Wno-missing-field-initializers -O2 -g -ggdb -I../deps/hiredis
-I../deps/linenoise -I../deps/lua/src -DUSE_JEMALLOC -I../deps/jemalloc/include >> .make-settings
echo PREV_FINAL_LDFLAGS=-g -ggdb -rdynamic >> .make-settings
(cd ../deps && make hiredis linenoise lua jemalloc)
make[2]: Entering directory `/usr/local/redis/deps'
(cd hiredis && make clean) > /dev/null || true
(cd linenoise && make clean) > /dev/null || true
(cd lua && make clean) > /dev/null || true
(cd jemalloc && [ -f Makefile ] && make distclean) > /dev/null || true
(rm -f .make-*)
(echo "" > .make-ldflags)
(echo "" > .make-cflags)
```

注：如果在编译过程中出现问题，有可能是安装包下载的有误，这里可以尝试的用别人下载的安装包或者直接用

wget http://download.redis.io/releases/redis-4.0.9.tar.gz

如果发现上述读不能解决问题，请参照该链接：<https://www.cnblogs.com/liu2-/p/6914159.html>

编译完成之后，可以看到解压文件redis-3.0.7 中会有对应的src、conf等文件夹

```
[root@youli redis-4.0.9]# ll
total 304
-rw-rw-r--. 1 root root 157632 Mar 27 00:04 00-RELEASENOTES
-rw-rw-r--. 1 root root 53 Mar 27 00:04 BUGS
-rw-rw-r--. 1 root root 1815 Mar 27 00:04 CONTRIBUTING
-rw-rw-r--. 1 root root 1487 Mar 27 00:04 COPYING
drwxrwxr-x. 6 root root 4096 Apr 4 09:55 deps
-rw-rw-r--. 1 root root 11 Mar 27 00:04 INSTALL
-rw-rw-r--. 1 root root 151 Mar 27 00:04 Makefile
-rw-rw-r--. 1 root root 4223 Mar 27 00:04 MANIFESTO
-rw-rw-r--. 1 root root 20543 Mar 27 00:04 README.md
-rw-rw-r--. 1 root root 58766 Mar 27 00:04 redis.conf
-rwxrwxr-x. 1 root root 271 Mar 27 00:04 runtest
-rwxrwxr-x. 1 root root 280 Mar 27 00:04 runtest-cluster
-rwxrwxr-x. 1 root root 281 Mar 27 00:04 runtest-sentinel
-rw-rw-r--. 1 root root 7606 Mar 27 00:04 sentinel.conf
drwxrwxr-x. 3 root root 4096 Apr 4 10:03 src
drwxrwxr-x. 10 root root 4096 Mar 27 00:04 tests
drwxrwxr-x. 8 root root 4096 Mar 27 00:04 utils
[root@youli redis-4.0.9]#
```

3.编译成功后，进入src文件夹，执行make install进行Redis安装。

如下图示安装完成，界面如下：

```
[root@youli redis-4.0.9]# cd src/
[root@youli src]# make install

Hint: It's a good idea to run 'make test' ;)

INSTALL install
INSTALL install
INSTALL install
INSTALL install
INSTALL install
[root@youli src]# ll
total 52016
-rw-rw-r--. 1 root root 10291 Mar 27 00:04 adlist.c
-rw-rw-r--. 1 root root 3512 Mar 27 00:04 adlist.h
-rw-r--r--. 1 root root 17016 Apr 4 10:02 adlist.o
-rw-rw-r--. 1 root root 17537 Mar 27 00:04 ae.c
-rw-rw-r--. 1 root root 4846 Mar 27 00:04 ae_epoll.c
-rw-rw-r--. 1 root root 10939 Mar 27 00:04 ae_evport.c
-rw-rw-r--. 1 root root 5308 Mar 27 00:04 ae.h
-rw-rw-r--. 1 root root 4567 Mar 27 00:04 ae_queue.c
-rw-r--r--. 1 root root 35312 Apr 4 10:02 ae.o
-rw-rw-r--. 1 root root 3828 Mar 27 00:04 ae_select.c
-rw-rw-r--. 1 root root 20701 Mar 27 00:04 anet.c
-rw-rw-r--. 1 root root 3562 Mar 27 00:04 anet.h
-rw-r--r--. 1 root root 49320 Apr 4 10:02 anet.o
-rw-rw-r--. 1 root root 64450 Mar 27 00:04 aof.c
-rw-rw-r--. 1 root root 141976 Apr 4 10:02 aof.o
-rw-rw-r--. 1 root root 2833 Mar 27 00:04 asciilogo.h
-rw-rw-r--. 1 root root 5367 Mar 27 00:04 atomicvar.h
-rw-rw-r--. 1 root root 10822 Mar 27 00:04 bio.c
-rw-rw-r--. 1 root root 2120 Mar 27 00:04 bio.h
```

第三步：部署

1.为了方便管理，将Redis文件中的conf配置文件和常用命令移动到统一文件中

1)、创建bin和etc文件

如图示：

```
[root@youli redis]# ls
00-RELEASENOTES CONTRIBUTING deps Makefile README.md runtest runtest-sentinel src utils
BUGS COPYING INSTALL MANIFESTO redis.conf runtest-cluster sentinel.conf tests
[root@youli redis]# mkdir etc
[root@youli redis]# mkdir bin
[root@youli redis]# ls
00-RELEASENOTES BUGS COPYING etc Makefile README.md runtest runtest-cluster runtest-sentinel src utils
bin CONTRIBUTING deps INSTALL MANIFESTO redis.conf runtest-cluster sentinel.conf tests
```

2)、回到刚刚安装目录，找到redis.conf，将其复制移动到 /usr/local/redis/ect 下

执行命令如下：

git(4)
IDEA(2)
IOS(1)
Java代码重构优化(3)
JAVA泛型(2)
Java跨域(1)
Java线程并发合集(5)
JTA(1)
Jvm调优(2)
Linux_Xshell(2)
Maven集合(3)
Mybatis
Netty读书笔记(1)
Nginx(4)
NIO学习笔记
Nosql
P2P(1)
Python系列(1)
Redis(6)
Shiro(2)
SOA_Dubbo(6)
Solr
Spring(2)
Spring AOP巧用大全(4)
Spring MVC(3)
Spring-Activity(1)
SpringBoot(4)
Spring源码浅析系列(1)
SQL
Swagger2(2)
zookeeper(2)
从零开始改造淘淘商城系列(2

mv redis.conf /usr/local/redis/etc/

```
[root@youli redis]# mv redis.conf /usr/local/redis/etc/
```

进入src目录，移动mkreleashdr.sh redis-benchmark redis-check-aof redis-check-rdb redis-cli redis-server 到/usr/local/redis/bin/

执行命令：mv mkreleashdr.sh redis-benchmark redis-check-aof redis-check-rdb redis-cli redis-server /usr/local/redis/bin/

```
[root@youli redis]# mv redis.conf /usr/local/redis/etc/
[root@youli redis]# cd src
[root@youli src]# ls
adlist.c      childinfo.o  evict.c      lzf.c.c      pubsub.c      redis-cli     server.o      t_set.o
adlist.h      cluster.c    evict.o      lzf.c.o      pubsub.o      redis-cli.c   setproctitle.c t_string.c
adlist.o      cluster.h    expire.c     lzf.d.c      quicklist.c   redis-cli.o   setproctitle.o t_string.o
ae.c          cluster.o    expire.o     lzf.d.o      quicklist.h   redismodule.h shal.c        t_zset.c
ae_epoll.c    config.c     fmacros.h    lzf.h        quicklist.o   shal.h        t_zset.o
ae_evport.c   config.h     geo.c        lzfP.h       rand.c        shal.o        util.c
ae.h          config.o     geo.h        Makefile     rand.h        siphash.c     util.h
ae_kqueue.c   crc16.c     geohash.c    Makefile.dep rand.o        siphash.o     util.o
ae.o          crc16.o     geohash.h    memtest.c    rax.c        release.h     slowlog.c     valgrind.sup
ae_select.c   crc64.c     geohash_helper.c memtest.o    rax.h        release.o     slowlog.h     version.h
anet.c        crc64.o     geohash_helper.h mkreleashdr.sh rax_malloc.h replication.c  slowlog.o     ziplist.c
anet.h        crc64.o     geohash_helper.o module.c      rax.o        replication.o solarisfixes.h ziplist.h
anet.o        db.c        geohash.o    module.o     rdb.c        rio.c         sort.c        ziplist.o
aof.c         db.o        geo.o        modules      rdb.h        rio.h         sort.o        zipmap.c
aof.o         debug.c     help.h       multi.c      rdb.o        rio.o        sparkline.c   zipmap.h
asciilogo.h   debugmacro.h hyperloglog.c multi.o      redisassert.h scripting.c    sparkline.h   zipmap.o
atomicvar.h   debug.o     hyperloglog.o networking.c  redis-benchmark scripting.o    sparkline.o   zmalloc.c
bio.c         defrag.c    intset.c     notify.c     redis-benchmark.c sdsalloc.h   syncio.c     zmalloc.h
bio.h         defrag.o    intset.o     notify.o     redis-benchmark.o sds.c        syncio.o     zmalloc.o
bio.o         dict.c     intset.c     object.c     redis-check-aof sds.h        testhelp.h
bitops.c     dict.h     latency.c    object.o     redis-check-aof.c sds.o        t_hash.c
bitops.o     dict.o     latency.h    object.o     redis-check-aof.o sentinel.c    t_hash.o
blocked.c    endianconv.c lazyfree.c   pqsort.c     redis-check-rdb sentinel.o    t_list.c
blocked.o    endianconv.h lazyfree.o   pqsort.h     redis-check-rdb.c server.c      t_list.o
childinfo.c  endianconv.o lazyfree.o   pqsort.o     redis-check-rdb.o server.h      t_set.c
[root@youli src]# mv mkreleashdr.sh redis-benchmark redis-check-aof redis-check-rdb redis-cli redis-server /usr/local/redis/bin/
```

依次查看移动后文件

```
[root@youli src]# cd /usr/local/redis/bin/
[root@youli bin]# ls
mkreleashdr.sh redis-benchmark redis-check-aof redis-check-rdb redis-cli redis-server
[root@youli bin]# cd ..
[root@youli redis]# cd ect/
[root@youli ect]# ls
redis.conf
[root@youli ect]# █
```

2、执行redis-server 启动redis

```
[root@youli bin]# ./redis-server
5884:C 04 Apr 11:35:30.516 # o090o090o090o Redis is starting o090o090o090o
5884:C 04 Apr 11:35:30.516 # Redis version=4.0.9, bits=64, commit=0909090900, modified=0, pid=5884, just started
5884:C 04 Apr 11:35:30.516 # Warning: no config file specified, using the default config. In order to specify a config file use ./redis-server /path/to/redis.conf
5884:M 04 Apr 11:35:30.519 * Increased maximum number of open files to 10032 (it was originally set to 1024).

Redis 4.0.9 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 5884

http://redis.io

5884:M 04 Apr 11:35:30.539 # WARNING: The TCP backlog setting of 511 cannot be enforced because /proc/sys/net/core/somaxconn is set to the lower value of 128.
5884:M 04 Apr 11:35:30.539 # Server initialized
5884:M 04 Apr 11:35:30.539 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to /etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
5884:M 04 Apr 11:35:30.540 # WARNING you have Transparent Huge Pages (THP) support enabled in your kernel. This will create latency and memory usage issues with Redis. To fix this issue run the command 'echo never > /sys/kernel/mm/transparent_hugepage/enabled' as root, and add it to your /etc/rc.local in order to retain the setting after a reboot. Redis must be restarted after THP is disabled.
5884:M 04 Apr 11:35:30.540 * Ready to accept connections
```

3、设置绑定ip（注：该步骤如果不需要可省略）

```
#
# include /path/to/local.conf
# include /path/to/other.conf

##### MODULES #####

# Load modules at startup. If the server is not able to load modules
# it will abort. It is possible to use multiple loadmodule directives.
#
# loadmodule /path/to/my_module.so
# loadmodule /path/to/other_module.so

##### NETWORK #####

# By default, if no "bind" configuration directive is specified, Redis listens
# for connections from all the network interfaces available on the server.
# It is possible to listen to just one or multiple selected interfaces using
# the "bind" configuration directive, followed by one or more IP addresses.
#
# Examples:
#
# bind 192.168.1.100 10.0.0.1
# bind 127.0.0.1 ::1
#
# ~~~ WARNING ~~~ If the computer running Redis is directly exposed to the
# internet, binding to all the interfaces is dangerous and will expose the
# instance to everybody on the internet. So by default we uncommend the
# following bind directive, that will force Redis to listen only into
# the IPv4 loopback interface address (this means Redis will be able to
# accept connections only from clients running into the same computer it
# is running).
#
# IF YOU ARE SURE YOU WANT YOUR INSTANCE TO LISTEN TO ALL THE INTERFACES
# JUST COMMENT THE FOLLOWING LINE.
# ~~~~~
bind 127.0.0.1
# Protected mode is a layer of security protection, in order to avoid that
# Redis instances left open on the internet are accessed and exploited.
```

多线程(5)

分布式事务(1)

高并发处理方案(3)

工具类

工作(4)

公众号精品备份集合(6)

集合相关(1)

加密解密(1)

面试有感

七牛云(1)

设计模式(5)

消息队列(3)

小程序(1)

新年计划

正则表达式(1)

随笔档案

2020年1月(1)

2019年12月(1)

2019年11月(2)

2019年10月(1)

2019年9月(1)

2019年7月(5)

2019年6月(1)

2019年5月(1)

2019年4月(7)

2019年1月(2)

2018年11月(12)

2018年10月(1)

2018年9月(3)

2018年8月(3)

2018年7月(1)

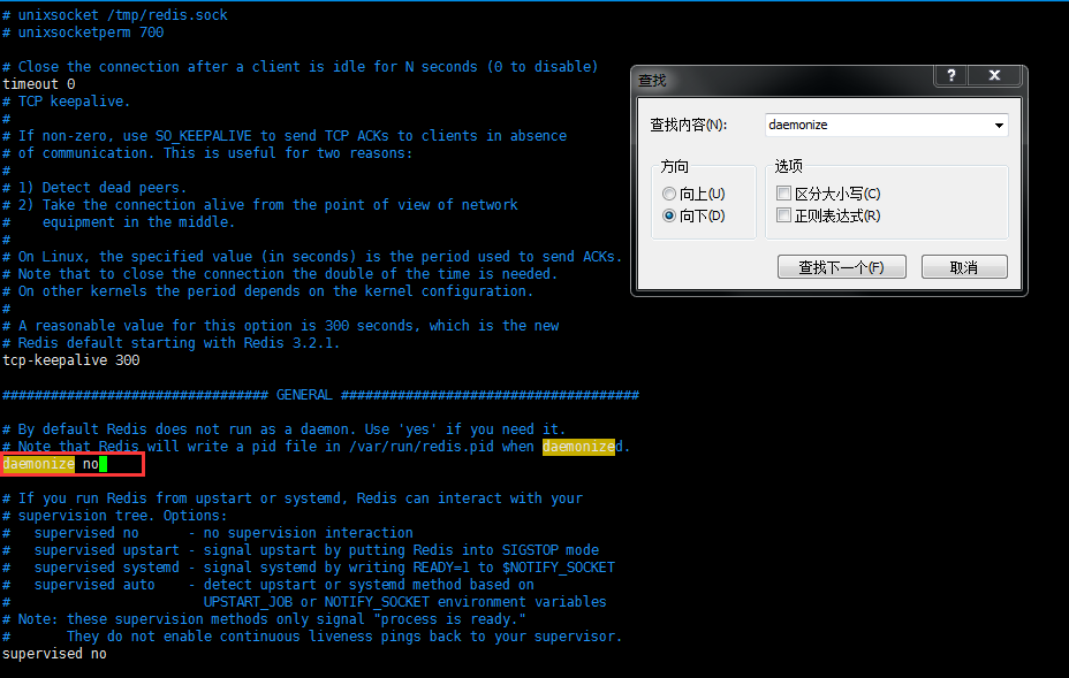
2018年6月(1)

如需要，可将上图绑定ip改为指定ip。

4、设置后台启动redis

- 1)、首先编辑conf文件，将daemonize属性改为yes（表明需要在后台运行）

```
cd etc/  
vim redis.conf
```



将no修改为yes

- 2)、再次启动redis服务，并指定启动服务配置文件

```
redis-server /usr/local/redis/etc/redis.conf
```



配置后台运行成功。

不经一番寒彻骨,怎得梅花扑鼻香

分类： Redis

好文要顶

关注我

收藏该文





达兔哥  
关注 - 14  
粉丝 - 35

+加关注

- « 上一篇： 【原】 DjianGo Windows7下的安装
- » 下一篇： 【原】 通过多线程分批处理派发任务

posted @ 2019-01-18 17:12 达兔哥 阅读(16816) 评论(0) 编辑 收藏

20

2018年5月(3)

2018年4月(7)

2018年3月(7)

2018年2月(1)

2018年1月(5)

2017年12月(1)

2017年11月(5)

2017年10月(4)

2017年9月(12)

2017年8月(3)

2017年7月(2)

2017年5月(2)

2017年4月(2)

2017年3月(4)

2017年2月(8)

2017年1月(1)

2016年12月(3)

2016年11月(3)

2016年10月(6)

2016年9月(1)

最新评论

1. Re: 【转】使用Freemark静态化

6666

--规格严格

2. Re: 【原】 Redis实现生成5555

--规格严格

3. Re: 【转】 Java策略消除if @ liusen1也可以采用spi的方

4. Re: 【转】 Java策略消除if

@ 小明记spring @Autowric泛型集合，不需要手动put...



相关博文：

- [Linux安装redis和部署](#)
- [Linux上安装Redis教程](#)
- [Redis学习-----单节点安装部署](#)
- [LinuxCentOS7.2下安装Redis&&配置Redis开机自启动](#)
- [Ubuntu中安装部署IntelCSWebRTC](#)
- » [更多推荐...](#)

这6种编码方法，你掌握了几个？

最新 IT 新闻：

- [小米10/Pro手机MIUI 11开发版计划公布：2月17日起内测推送](#)
- [又一支付领域独角兽崭露头角？「Flywire」完成 1.2 亿美元E轮融资](#)
- [小米天猫超级品牌日终极战报：狂揽61项第一 销售额卖3亿元！](#)
- [苹果加入FIDO技术联盟 拟淘汰古老的密码](#)
- [红外热像技术如何实现体温筛查](#)
- » [更多新闻...](#)

5. Re: [【转】Java策略消除if](#)

如果增加了一种充值方式怎么  
yFactory 中的map是不是要  
t, 有没有更好的方式呢?

阅读排行榜

1. [Linux安装redis和部署](#)(16544)

2. [【原】Spring和Dubbo基  
整合过程](#)(16544)

3. [【原】通过Dubbo注解实现  
\(16373\)](#)

4. [【原】Nginx搭建FTP服务  
题](#)(12686)

5. [【原】ActiveMq实现分布  
性](#)(11586)

评论排行榜

1. [【原】继承AbstractRouti  
urce再通过AOP实现动态数据](#)

2. [【原】ActiveMq实现分布  
性](#)(5)

3. [【原】通过Dubbo注解实现  
\(5\)](#)

4. [【转】Java策略消除if els](#)

5. [【原】Spring AOP实现对  
存同步](#)(4)

推荐排行榜

1. [【转】浅谈Nginx负载均衡  
别](#)(4)

2. [【转】Cookie深度解析](#)(3)

3. [Linux安装redis和部署](#)(2)

4. [【原】通过Dubbo注解实现  
\(2\)](#)

5. [【原】Spring和Dubbo基  
整合过程](#)(2)