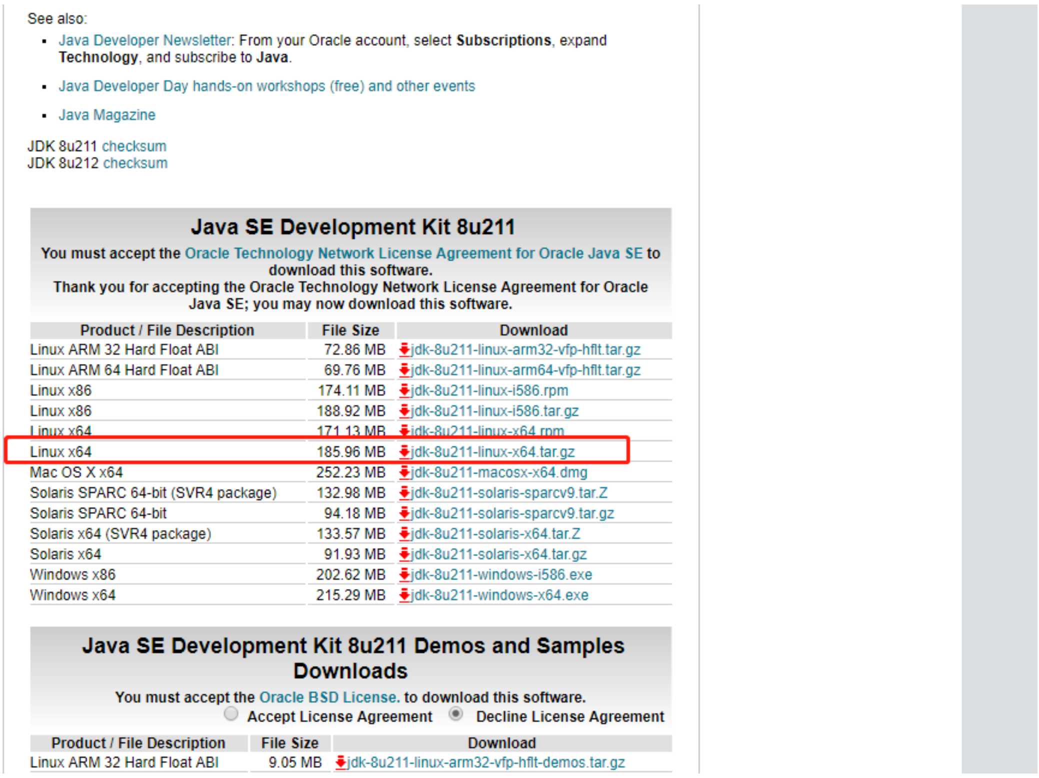
中间件安装

Linux系统

jdk

1. 进入Orcal官网，下载适用于lunux系统的jdk**.tar.gz，如



See also:

- Java Developer Newsletter: From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to **Java**.
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

JDK 8u211 checksum
JDK 8u212 checksum

Java SE Development Kit 8u211

You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software.
Thank you for accepting the Oracle Technology Network License Agreement for Oracle Java SE; you may now download this software.

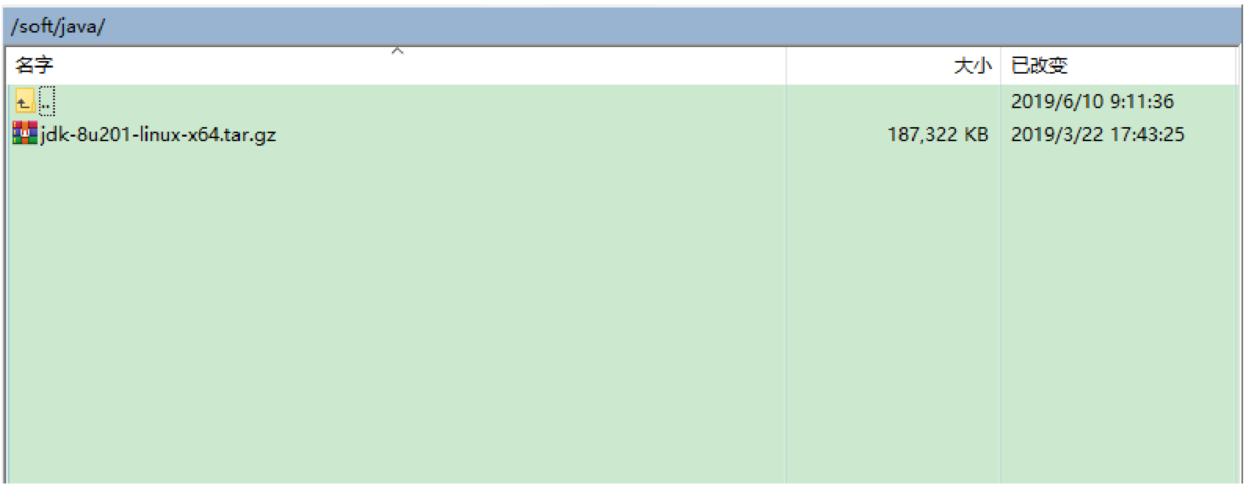
Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.86 MB	jdk-8u211-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.76 MB	jdk-8u211-linux-arm64-vfp-hflt.tar.gz
Linux x86	174.11 MB	jdk-8u211-linux-i586.rpm
Linux x86	188.92 MB	jdk-8u211-linux-i586.tar.gz
Linux x64	171.13 MB	jdk-8u211-linux-x64.rpm
Linux x64	185.96 MB	jdk-8u211-linux-x64.tar.gz
Mac OS X x64	252.23 MB	jdk-8u211-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	132.98 MB	jdk-8u211-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.18 MB	jdk-8u211-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.57 MB	jdk-8u211-solaris-x64.tar.Z
Solaris x64	91.93 MB	jdk-8u211-solaris-x64.tar.gz
Windows x86	202.62 MB	jdk-8u211-windows-i586.exe
Windows x64	215.29 MB	jdk-8u211-windows-x64.exe

Java SE Development Kit 8u211 Demos and Samples Downloads

You must accept the Oracle BSD License. to download this software.
○ Accept License Agreement ● Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	9.05 MB	jdk-8u211-linux-arm32-vfp-hflt-demos.tar.gz

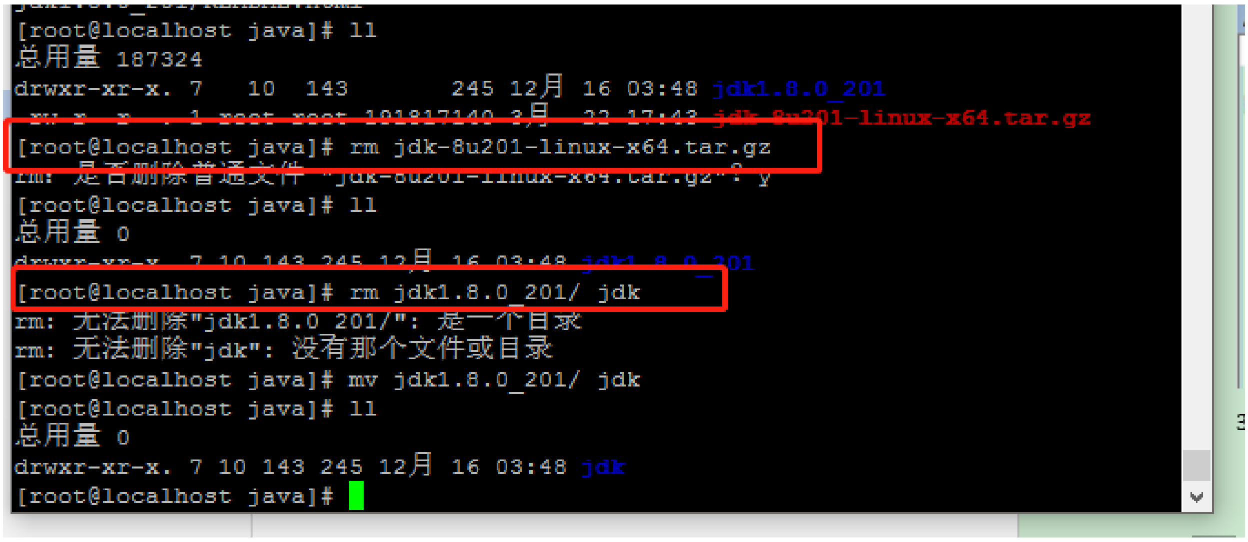
2. 使用linux远程工具，如winscp连接至linux，创建/soft/java文件夹，将下载好的jdk上传至/soft/java下，如



/soft/java/

名字	大小	已改变
..		2019/6/10 9:11:36
jdk-8u201-linux-x64.tar.gz	187,322 KB	2019/3/22 17:43:25

3. 使用命令行工具进入上一步的目录下，对jdk的压缩包使用tar zxvf jdk***.tar.gz命令进行解压缩，删除无用的压缩包，将解压后的文件夹重命名为jdk1.8

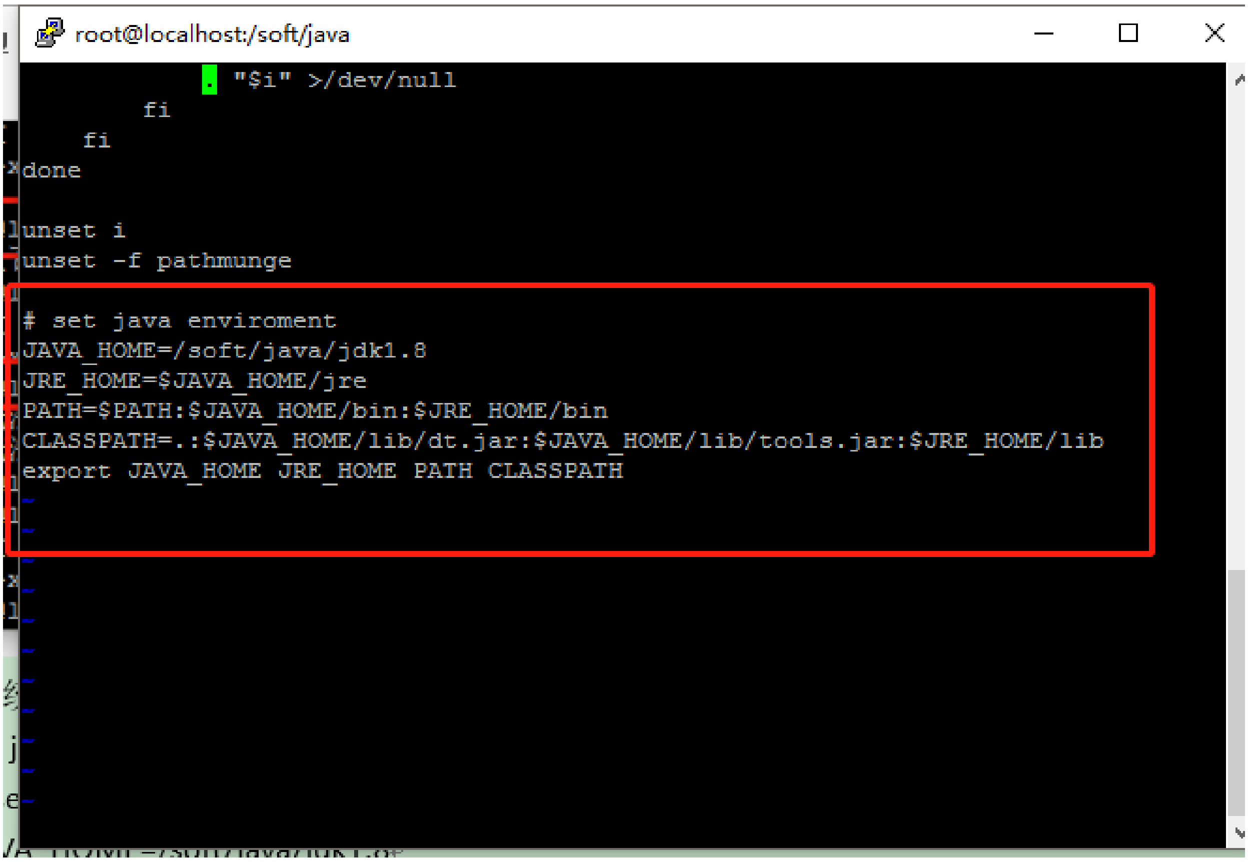


```
[root@localhost java]# ll
总用量 187324
drwxr-xr-x. 7   10  143       245 12月 16 03:48 jdk1.8.0_201
-rw-r--r--. 1 root root 191817140 3月  22 17:43 jdk-8u201-linux-x64.tar.gz
[root@localhost java]# rm jdk-8u201-linux-x64.tar.gz
rm: 是否删除普通文件 "jdk-8u201-linux-x64.tar.gz"? y
[root@localhost java]# ll
总用量 0
drwxr-xr-x. 7 10 143 245 12月 16 03:48 jdk1.8.0_201
[root@localhost java]# rm jdk1.8.0_201/ jdk
rm: 无法删除"jdk1.8.0_201/": 是一个目录
rm: 无法删除"jdk": 没有那个文件或目录
[root@localhost java]# mv jdk1.8.0_201/ jdk
[root@localhost java]# ll
总用量 0
drwxr-xr-x. 7 10 143 245 12月 16 03:48 jdk
[root@localhost java]#
```

4. 配置系统环境变量：vi /etc/profile 在该文件最后增加以下配置信息，其中JAVA_HOME是真实的java所在位置

```
# set java enviroment
JAVA_HOME=/soft/java/jdk1.8
JRE_HOME=$JAVA_HOME/jre
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib
export JAVA_HOME JRE_HOME PATH CLASSPATH
```

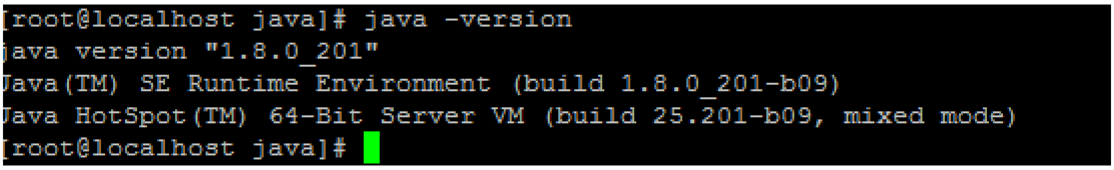
如图：



```
                . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge

# set java enviroment
JAVA_HOME=/soft/java/jdk1.8
JRE_HOME=$JAVA_HOME/jre
PATH=$PATH:$JAVA_HOME/bin:$JRE_HOME/bin
CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:$JRE_HOME/lib
export JAVA_HOME JRE_HOME PATH CLASSPATH
~
~
~
~
~
~
~
~
~
~
```

5. 使配置生效：source /etc/profile

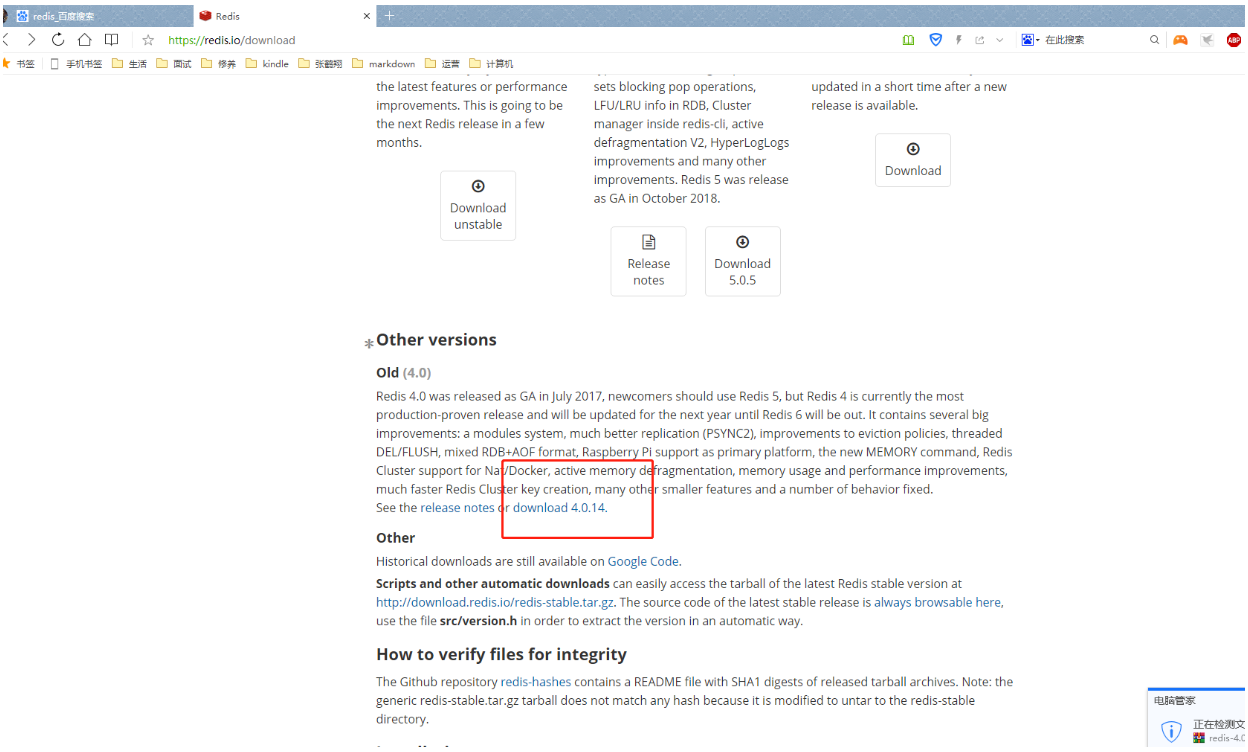
6. 输入java －version 查看java是否配置生效



```
[root@localhost java]# java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
[root@localhost java]# 
```

Redis

1. 进入redis官网下载redisxxx.tar.gz



the latest features or performance improvements. This is going to be the next Redis release in a few months.

sets blocking pop operations, LFU/LRU info in RDB, Cluster manager inside redis-cli, active defragmentation V2, HyperLogLogs improvements and many other improvements. Redis 5 was release as GA in October 2018.

updated in a short time after a new release is available.

Other versions

Old (4.0)

Redis 4.0 was released as GA in July 2017, newcomers should use Redis 5, but Redis 4 is currently the most production-proven release and will be updated for the next year until Redis 6 will be out. It 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.
See the release notes or download 4.0.14.

Other

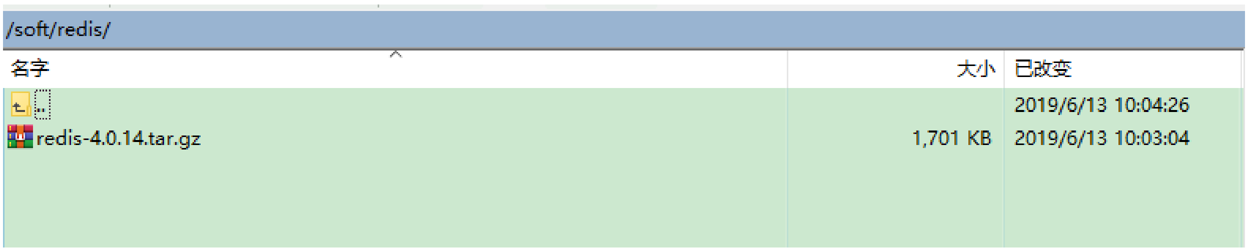
Historical downloads are still available on Google Code.

Scripts and other automatic downloads can easily access the tarball of the latest Redis stable version at http://download.redis.io/redis-stable.tar.gz. The source code of the latest stable release is always browsable here, use the file **src/version.h** in order to extract the version in an automatic way.

How to verify files for integrity

The Github repository redis-hashes contains a README file with SHA1 digests of released tarball archives. Note: the generic redis-stable.tar.gz tarball does not match any hash because it is modified to untar to the redis-stable directory.

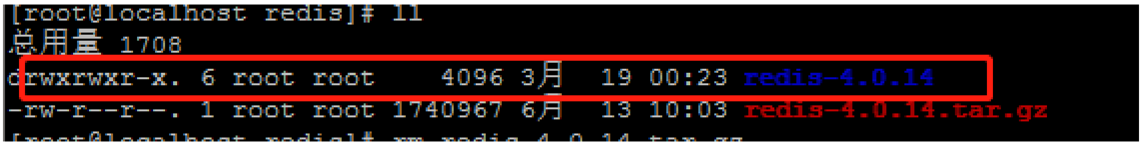
2. 上传redis至centos的soft/redis文件夹下



/soft/redis/

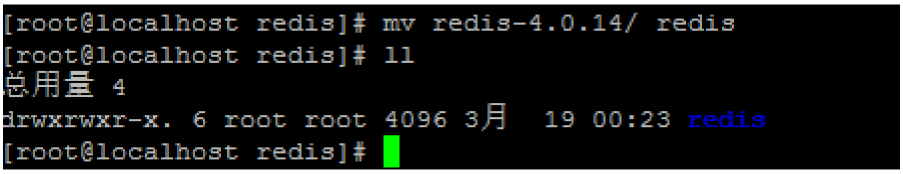
名字	大小	已改变
..		2019/6/13 10:04:26
redis-4.0.14.tar.gz	1,701 KB	2019/6/13 10:03:04

3. 解压redis：tar -zxvf redisxxx.tar.gz



```
[root@localhost redis]# ll
总用量 1708
drwxrwxr-x. 6 root root    4096 3月  19 00:23 redis-4.0.14
-rw-r--r--. 1 root root 1740967 6月  13 10:03 redis-4.0.14.tar.gz
[root@localhost redis]# rm redis-4.0.14.tar.gz
```

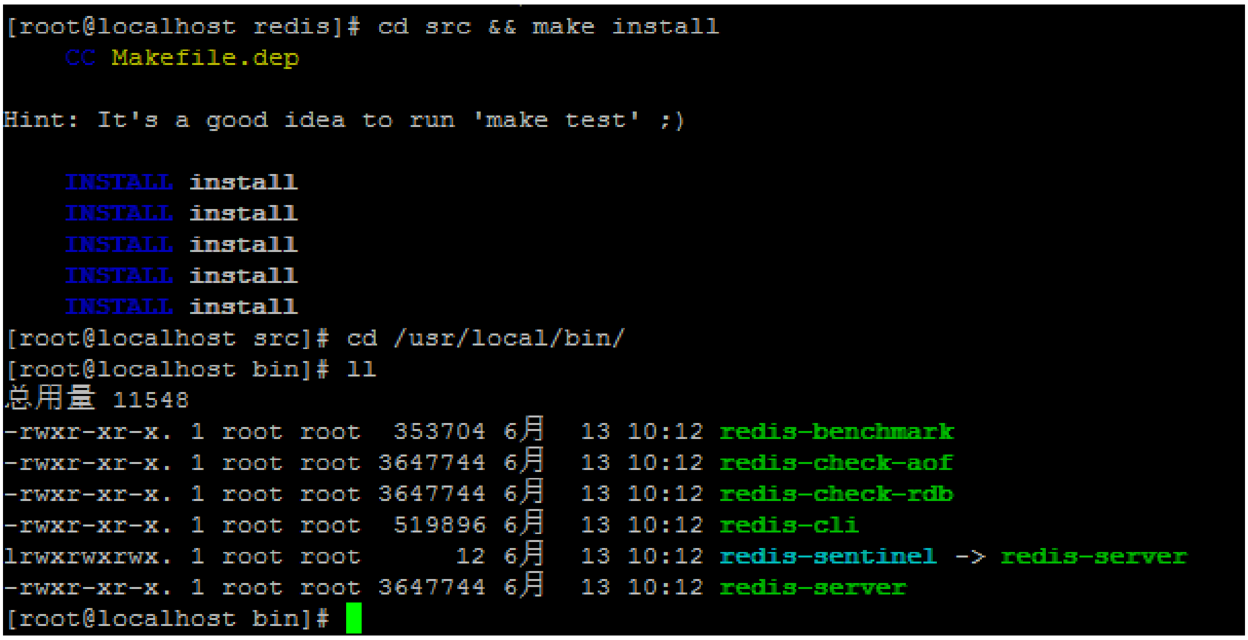
4. 删除无用的压缩包：rm -rf redisxxx.tar.gz
5. 将解压后的文件夹重命名为redis：mv redisxxx redis



```
[root@localhost redis]# mv redis-4.0.14/ redis
[root@localhost redis]# ll
总用量 4
drwxrwxr-x. 6 root root 4096 3月  19 00:23 redis
[root@localhost redis]# 
```

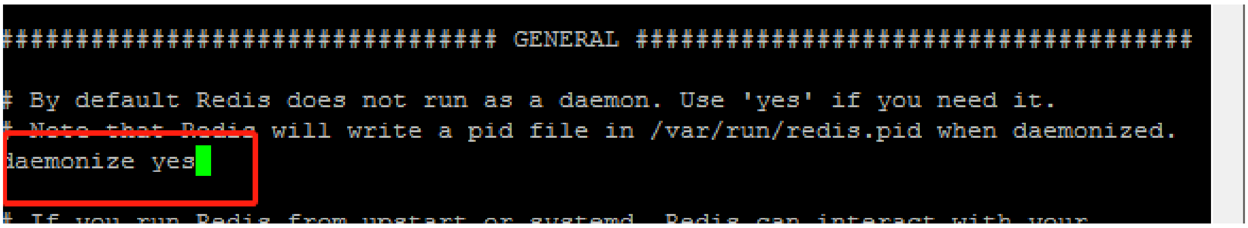
6. 安装gcc依赖：yum install gcc
7. 进入解压后的redis目录，编译安装redis：make MALLOC=libc

8. 将/soft/redis/redis/src目录的文件加到/usr/local/bin目录，在解压后的redis目录中执行cd src && make install



```
[root@localhost redis]# cd src && make install
    CC Makefile.dep

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

    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
    INSTALL install
[root@localhost src]# cd /usr/local/bin/
[root@localhost bin]# ll
总用量 11548
-rwxr-xr-x. 1 root root  353704 6月  13 10:12 redis-benchmark
-rwxr-xr-x. 1 root root 3647744 6月  13 10:12 redis-check-aof
-rwxr-xr-x. 1 root root 3647744 6月  13 10:12 redis-check-rdb
-rwxr-xr-x. 1 root root  519896 6月  13 10:12 redis-cli
lrwxrwxrwx. 1 root root      12 6月  13 10:12 redis-sentinel -> redis-server
-rwxr-xr-x. 1 root root 3647744 6月  13 10:12 redis-server
[root@localhost bin]#
```

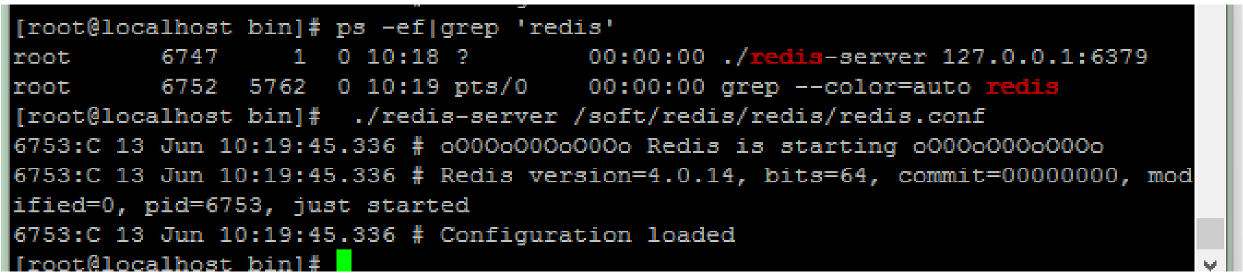
9. 设置redis为后台启动，修改redis.conf文件：vi /soft/redis/redis/redis.conf，找到daemonize no，将no修改为yes，保存并退出



```
################################# GENERAL #####################################

# By default Redis does not run as a daemon. Use 'yes' if you need it.
# Note that Redis will write a pid file in /var/run/redis.pid when daemonized.
daemonize yes

# If you run Redis from upstart or systemd, Redis can interact with your
```

10. 进入usr/local/bin目录下，执行./redis-server /soft/redis/redis/redis.conf

11. 查看redis是否执行成功：ps-ef|greo ‘redis’

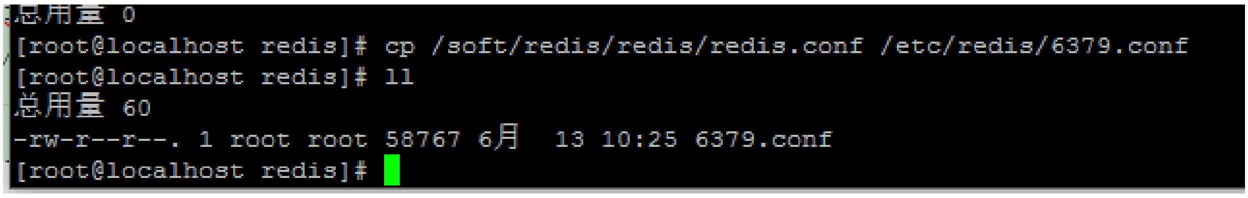


```
[root@localhost bin]# ps -ef|grep 'redis'
root      6747     1  0 10:18 ?        00:00:00 ./redis-server 127.0.0.1:6379
root      6752  5762  0 10:19 pts/0    00:00:00 grep --color=auto redis
[root@localhost bin]#  ./redis-server /soft/redis/redis/redis.conf
6753:C 13 Jun 10:19:45.336 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo
6753:C 13 Jun 10:19:45.336 # Redis version=4.0.14, bits=64, commit=00000000, mod
ified=0, pid=6753, just started
6753:C 13 Jun 10:19:45.336 # Configuration loaded
[root@localhost bin]#
```

12. 杀死redis进程：kill -9 6747

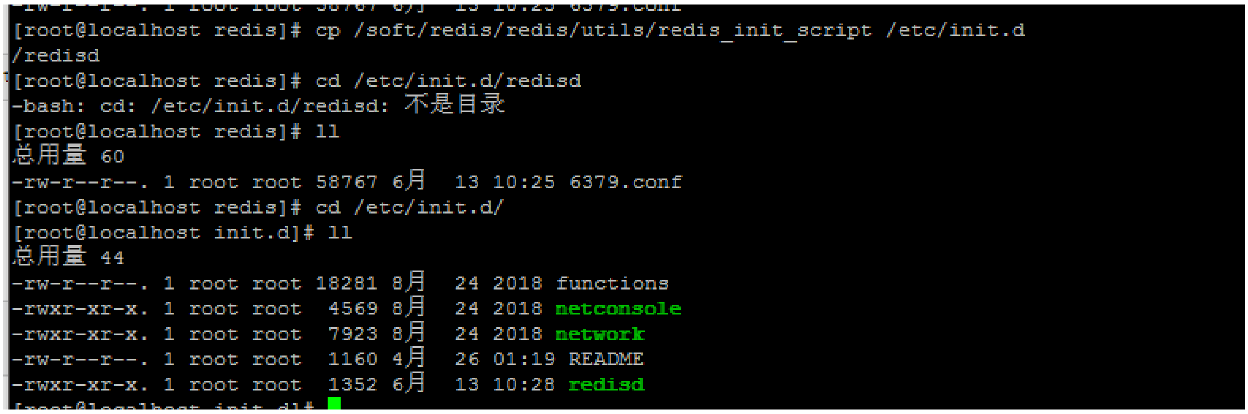
13. 将redis配置开机启动：在/etc目录下创建redis目录：makdir /etc/redis

14. 将/soft/redis/redis/redis.conf文件复制一份至/etc/redis目录下，且重命名为6379.conf



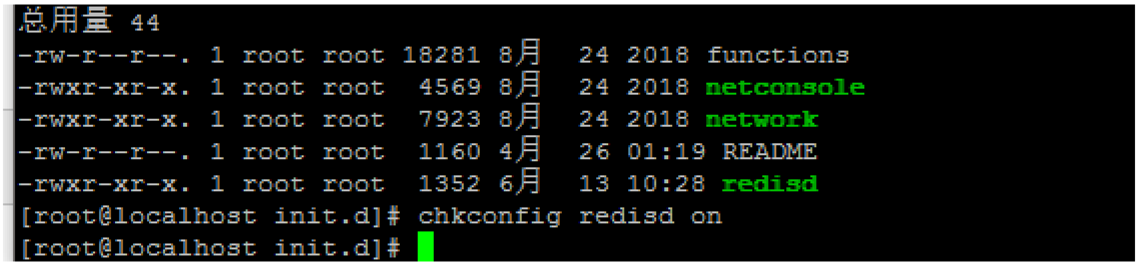
```
总用量 0
[root@localhost redis]# cp /soft/redis/redis/redis.conf /etc/redis/6379.conf
[root@localhost redis]# ll
总用量 60
-rw-r--r--. 1 root root 58767 6月  13 10:25 6379.conf
[root@localhost redis]#
```

15. 将redis的启动脚本复制一份放到/etc/init.d目录下：cp /soft/redis/redis/utils/redis_init_script /etc/init.d/，并重命名为redisd



```
-rw-r--r--. 1 root root 58767 6月  13 10:25 6379.conf
[root@localhost redis]# cp /soft/redis/redis/utils/redis_init_script /etc/init.d
/redisd
[root@localhost redis]# cd /etc/init.d/redisd
-bash: cd: /etc/init.d/redisd: 不是目录
[root@localhost redis]# ll
总用量 60
-rw-r--r--. 1 root root 58767 6月  13 10:25 6379.conf
[root@localhost redis]# cd /etc/init.d/
[root@localhost init.d]# ll
总用量 44
-rw-r--r--. 1 root root 18281 8月  24 2018 functions
-rwxr-xr-x. 1 root root  4569 8月  24 2018 netconsole
-rwxr-xr-x. 1 root root  7923 8月  24 2018 network
-rw-r--r--. 1 root root  1160 4月  26 01:19 README
-rwxr-xr-x. 1 root root  1352 6月  13 10:28 redisd
[root@localhost init.d]#
```

16. 切换到/etc/init.d目录下,执行自启动命令：chkconllfig redisd on



```
总用量 44
-rw-r--r--. 1 root root 18281 8月  24 2018 functions
-rwxr-xr-x. 1 root root  4569 8月  24 2018 netconsole
-rwxr-xr-x. 1 root root  7923 8月  24 2018 network
-rw-r--r--. 1 root root  1160 4月  26 01:19 README
-rwxr-xr-x. 1 root root  1352 6月  13 10:28 redisd
[root@localhost init.d]# chkconfig redisd on
[root@localhost init.d]#
```

如果出现service redisd does not support chkconfig ，说明redisd不支持chkconfig，使用vim编辑redisdredisd文件，在第一行加入下面两行注释保存并退出：

# chkconfig: 2345 90 10

# description: Redis is a persistent key-value database

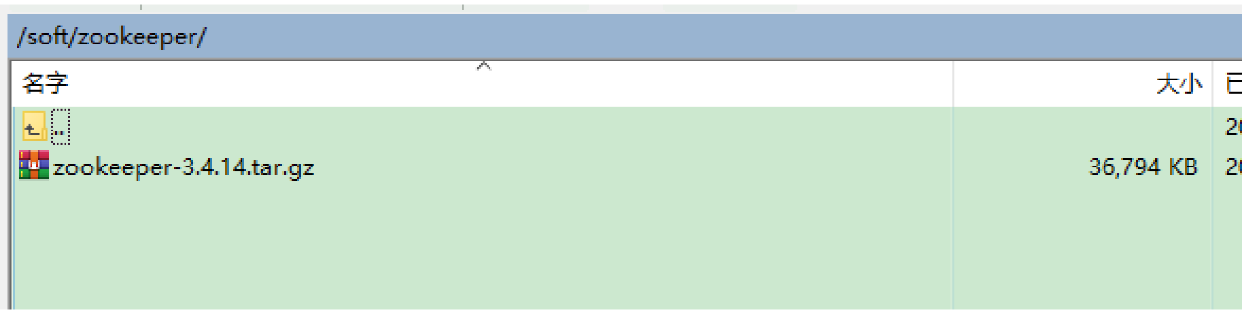
再次执行自启动命令：chkconfig redisd on

17. 安装完成，执行service redisd start启动redis，执行service redisd stop停止redis

18. 如果出现/var/redis/run/redis_6379.pid exists, process is already running or crashed错误信息，先执行一下service redisd stop，然后稍等一会执行service redisd start启动redis，如果依然出现该问题，重启一下服务器 shutdown －r now即可，启动之后之后redis会自动启动，并且redis启动和关闭命令均可正常使用

Zookeeper

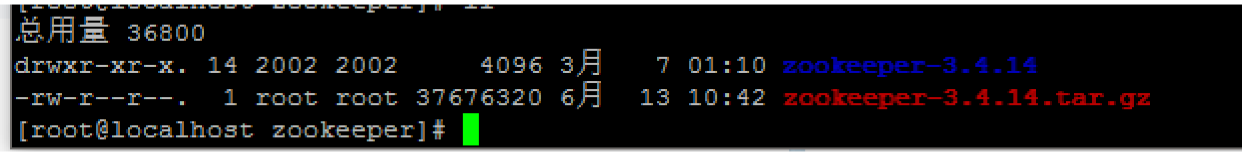
1. 从官网下载zookeeperxxx.tar.gz，并上传至服务器目录，/soft/zookeeper



/soft/zookeeper/

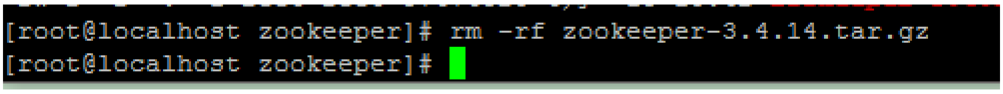
名字	大小	E
..		2
zookeeper-3.4.14.tar.gz	36,794 KB	2

2. 解压缩zookeeperxxx.tar.gz： tar －zxvf zookeeperxxx.tar.gz



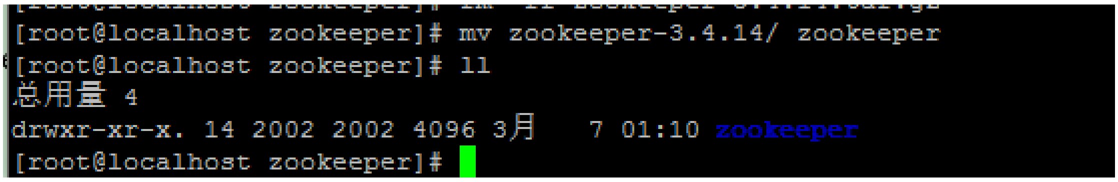
```
总用量 36800
drwxr-xr-x. 14 2002 2002      4096 3月   7 01:10 zookeeper-3.4.14
-rw-r--r--.  1 root root 37676320 6月  13 10:42 zookeeper-3.4.14.tar.gz
[root@localhost zookeeper]#
```

3. 删除压缩包：rm －rf zookeeperxxx.tar.gz



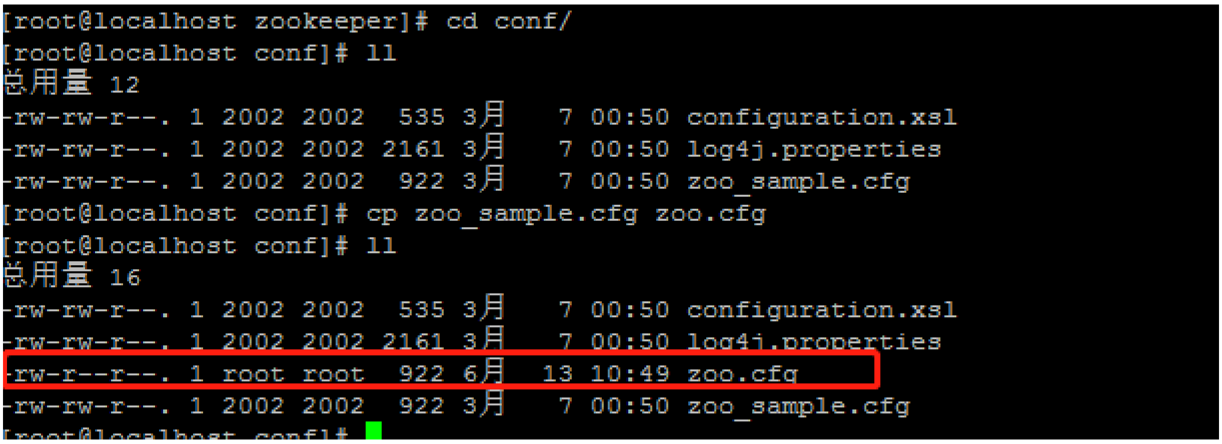
```
[root@localhost zookeeper]# rm -rf zookeeper-3.4.14.tar.gz
[root@localhost zookeeper]# 
```

4. 重命名解压后的文件夹为zookeeper



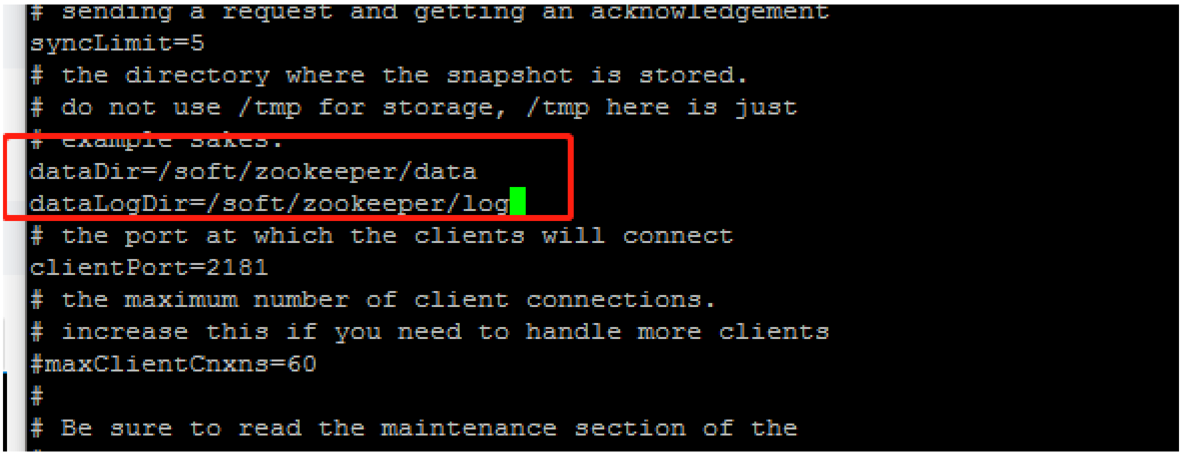
```
[root@localhost zookeeper]# mv zookeeper-3.4.14/ zookeeper
[root@localhost zookeeper]# ll
总用量 4
drwxr-xr-x. 14 2002 2002 4096 3月   7 01:10 zookeeper
[root@localhost zookeeper]# 
```

5. 赋值一份zookeeper/conf下的zoo_sample.cfg，并重命名为zoo.cfg



```
[root@localhost zookeeper]# cd conf/
[root@localhost conf]# ll
总用量 12
-rw-rw-r--. 1 2002 2002  535 3月   7 00:50 configuration.xsl
-rw-rw-r--. 1 2002 2002 2161 3月   7 00:50 log4j.properties
-rw-rw-r--. 1 2002 2002  922 3月   7 00:50 zoo_sample.cfg
[root@localhost conf]# cp zoo_sample.cfg zoo.cfg
[root@localhost conf]# ll
总用量 16
-rw-rw-r--. 1 2002 2002  535 3月   7 00:50 configuration.xsl
-rw-rw-r--. 1 2002 2002 2161 3月   7 00:50 log4j.properties
-rw-r--r--. 1 root root  922 6月  13 10:49 zoo.cfg
-rw-rw-r--. 1 2002 2002  922 3月   7 00:50 zoo_sample.cfg
[root@localhost conf]# 
```

6. 修改zoo.cfg的配置，日志地址为/soft/zookeeper/log，数据地址为/soft/zookeeper/data，保存并退出，并且创建好对应的文件夹



```
# sending a request and getting an acknowledgement
syncLimit=5
# the directory where the snapshot is stored.
# do not use /tmp for storage, /tmp here is just
# example sakes.
dataDir=/soft/zookeeper/data
dataLogDir=/soft/zookeeper/log
# the port at which the clients will connect
clientPort=2181
# the maximum number of client connections.
# increase this if you need to handle more clients
#maxClientCnxns=60
#
# Be sure to read the maintenance section of the
```

7. 配置环境变量：vi /etc/profile，在最后增加下面两行：

export ZOOKEEPER=/soft/zookeeper/zookeeper

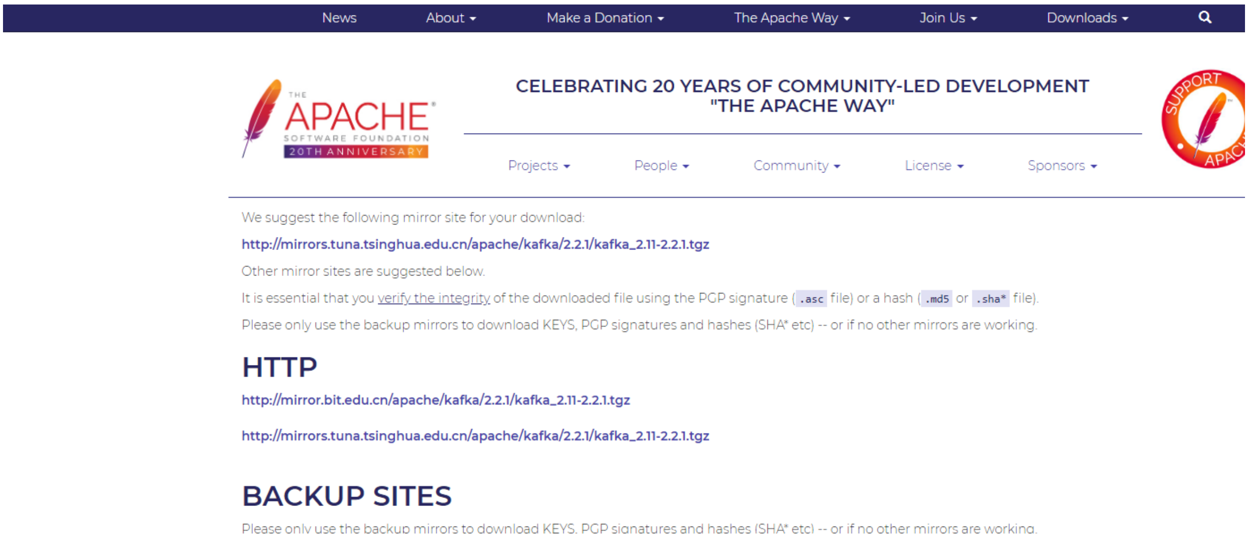
export PATH=\$PATH:\$ZOOKEEPER/bin

8. 使配置生效：source /etc/profile

9. 启动zookeeper：zkServer.sh start；查看运行状态：zkServer.sh status；启动客户端：zkCli.sh；停止zookeeper：zkServer.sh stop

Kafka

1. kafka官网下载kafka，上传压缩包至/soft/kafka目录下



We suggest the following mirror site for your download:

http://mirrors.tuna.tsinghua.edu.cn/apache/kafka/2.2.1/kafka_2.11-2.2.1.tgz

Other mirror sites are suggested below.

It is essential that you verify the integrity of the downloaded file using the PGP signature (`.asc` file) or a hash (`.md5` or `.sha*` file).

Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

HTTP

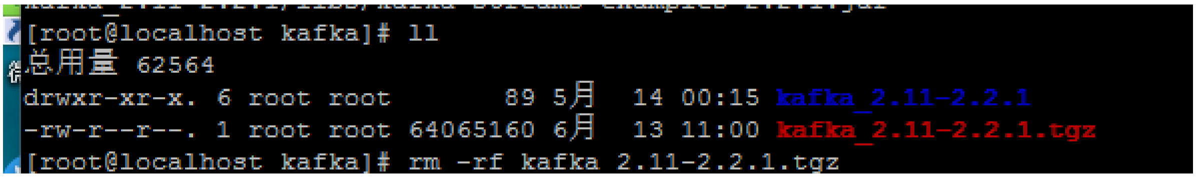
http://mirror.bit.edu.cn/apache/kafka/2.2.1/kafka_2.11-2.2.1.tgz

http://mirrors.tuna.tsinghua.edu.cn/apache/kafka/2.2.1/kafka_2.11-2.2.1.tgz

BACKUP SITES

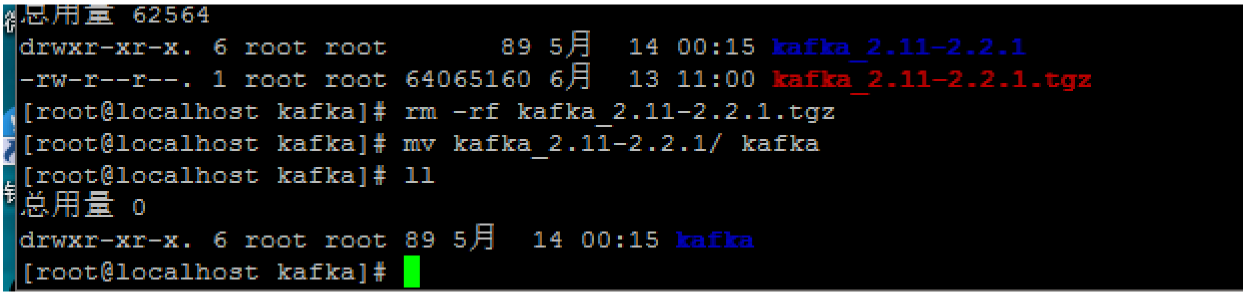
Please only use the backup mirrors to download KEYS, PGP signatures and hashes (SHA* etc) -- or if no other mirrors are working.

2. 解压缩压缩包：tar －zxvf kafkaxxx.tgz



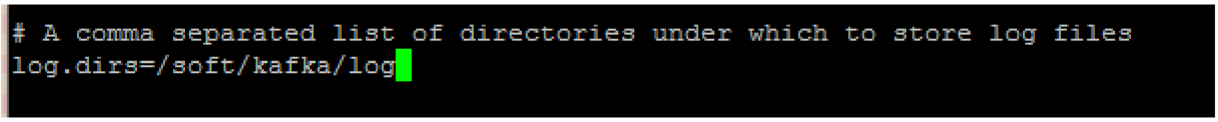
```
[root@localhost kafka]# ll
总用量 62564
drwxr-xr-x. 6 root root       89 5月  14 00:15 kafka_2.11-2.2.1
-rw-r--r--. 1 root root 64065160 6月  13 11:00 kafka_2.11-2.2.1.tgz
[root@localhost kafka]# rm -rf kafka 2.11-2.2.1.tgz
```

3. 删除无用压缩包，解压后的文件夹重命名为kafka：rm -rf kafkaxxx.tgz，mv kafkaxxx/ kafka



```
总用量 62564
drwxr-xr-x. 6 root root       89 5月  14 00:15 kafka_2.11-2.2.1
-rw-r--r--. 1 root root 64065160 6月  13 11:00 kafka_2.11-2.2.1.tgz
[root@localhost kafka]# rm -rf kafka_2.11-2.2.1.tgz
[root@localhost kafka]# mv kafka_2.11-2.2.1/ kafka
[root@localhost kafka]# ll
总用量 0
drwxr-xr-x. 6 root root 89 5月  14 00:15 kafka
[root@localhost kafka]#
```

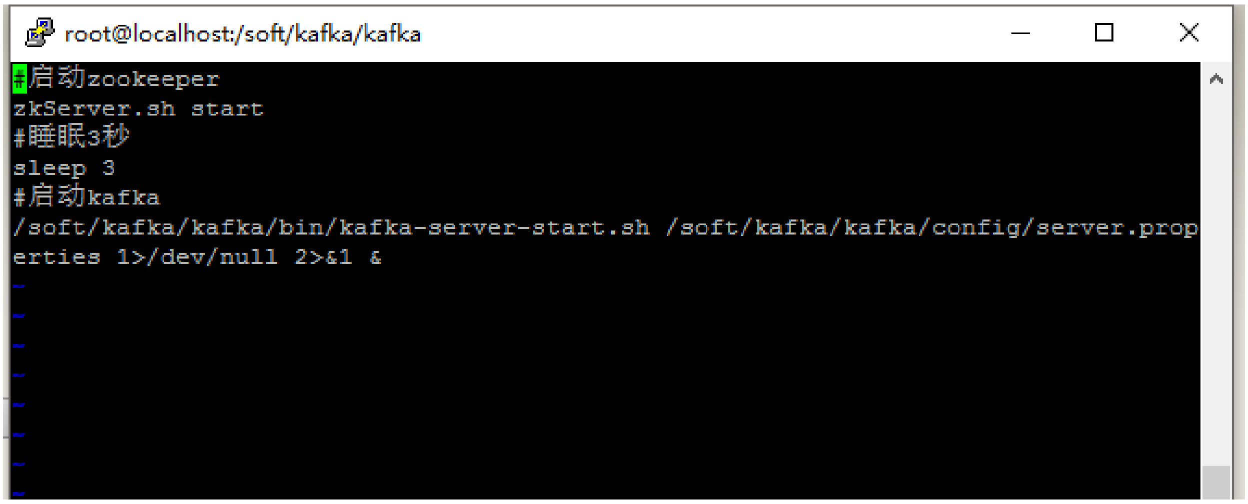
4. 修改/soft/kafka/kafka/config/server.properties修改配置文件



```
# A comma separated list of directories under which to store log files
log.dirs=/soft/kafka/log
```

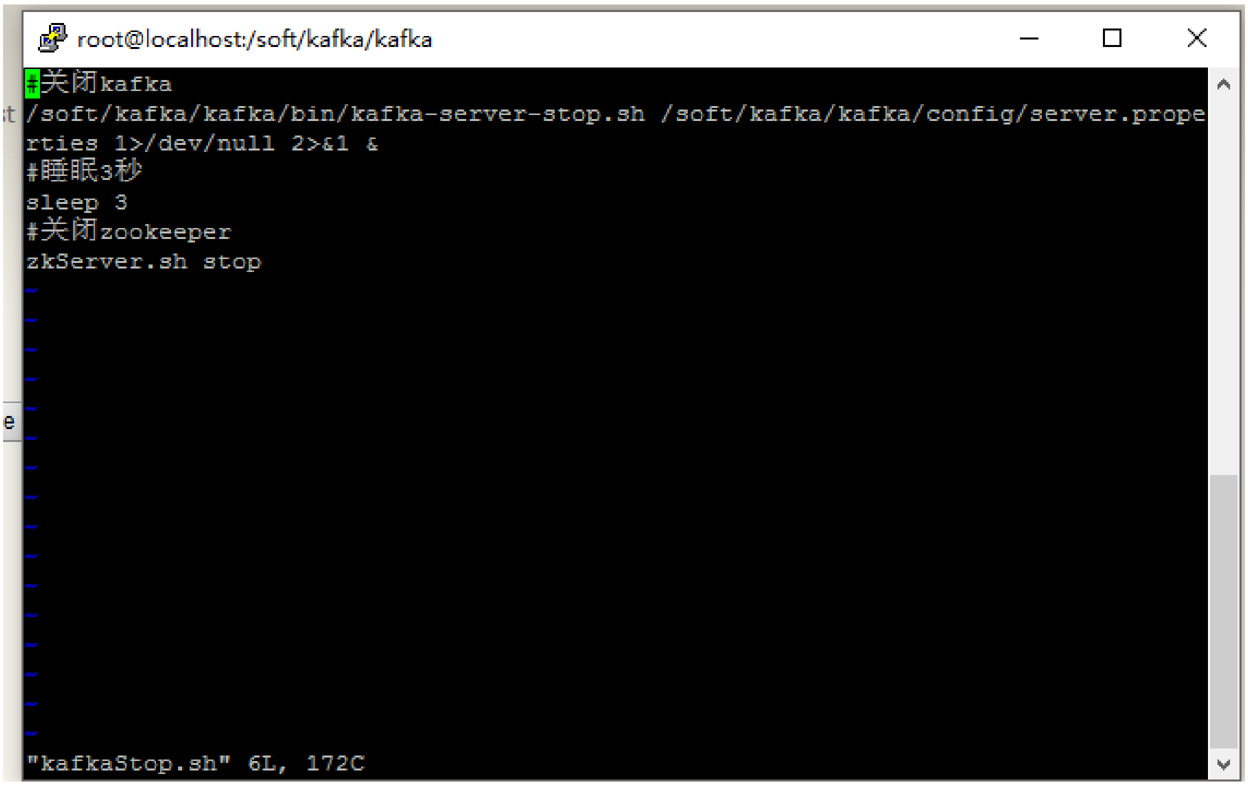
创建好对应的目录

5. 编写启动脚本：startKafka.sh：/soft/kafka/kafka/bin/kafka-server-start.sh /soft/kafka/kafka/config/server.properties 1>dev/null 2>&1 &



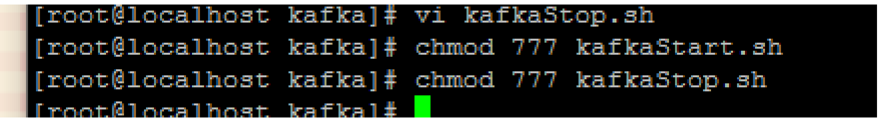
```
#启动zookeeper
zkServer.sh start
#睡眠3秒
sleep 3
#启动kafka
/soft/kafka/kafka/bin/kafka-server-start.sh /soft/kafka/kafka/config/server.prop
erties 1>/dev/null 2>&1 &
~
~
~
~
~
~
~
~
```

6. 编写停止脚本：stopKafka.sh



```
#关闭kafka
/soft/kafka/kafka/bin/kafka-server-stop.sh /soft/kafka/kafka/config/server.prope
rties 1>/dev/null 2>&1 &
#睡眠3秒
sleep 3
#关闭zookeeper
zkServer.sh stop
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"kafkaStop.sh" 6L, 172C
```

7. 获取两个脚本的执行权限

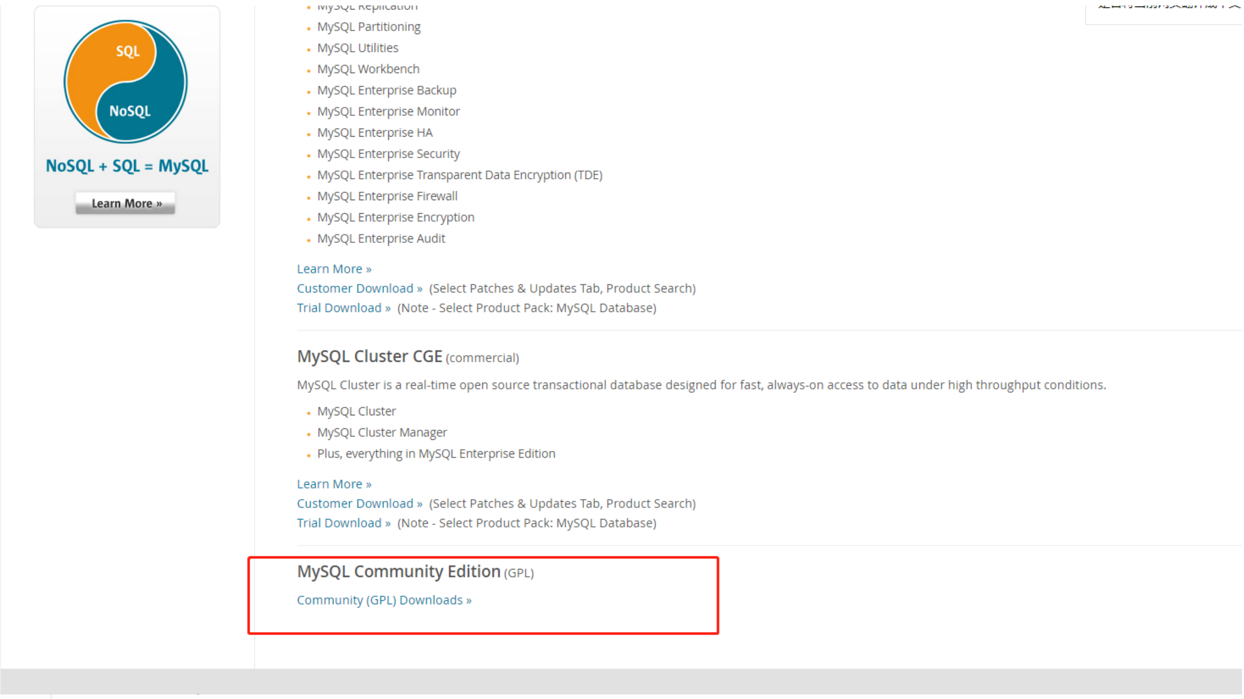


```
[root@localhost kafka]# vi kafkaStop.sh
[root@localhost kafka]# chmod 777 kafkaStart.sh
[root@localhost kafka]# chmod 777 kafkaStop.sh
[root@localhost kafka]# 
```

8.测试启动和停止：./kafkaStart.sh ./kafkaStop.sh

Mysql

1. 官网下载mysql程序，进入mysql官网，进入社区版下载页面，下载MySQL Community Server社区办服务，下载5.6/5.7版本。



- MySQL Replication
- MySQL Partitioning
- MySQL Utilities
- MySQL Workbench
- MySQL Enterprise Backup
- MySQL Enterprise Monitor
- MySQL Enterprise HA
- MySQL Enterprise Security
- MySQL Enterprise Transparent Data Encryption (TDE)
- MySQL Enterprise Firewall
- MySQL Enterprise Encryption
- MySQL Enterprise Audit

Learn More »
Customer Download » (Select Patches & Updates Tab, Product Search)
Trial Download » (Note - Select Product Pack: MySQL Database)

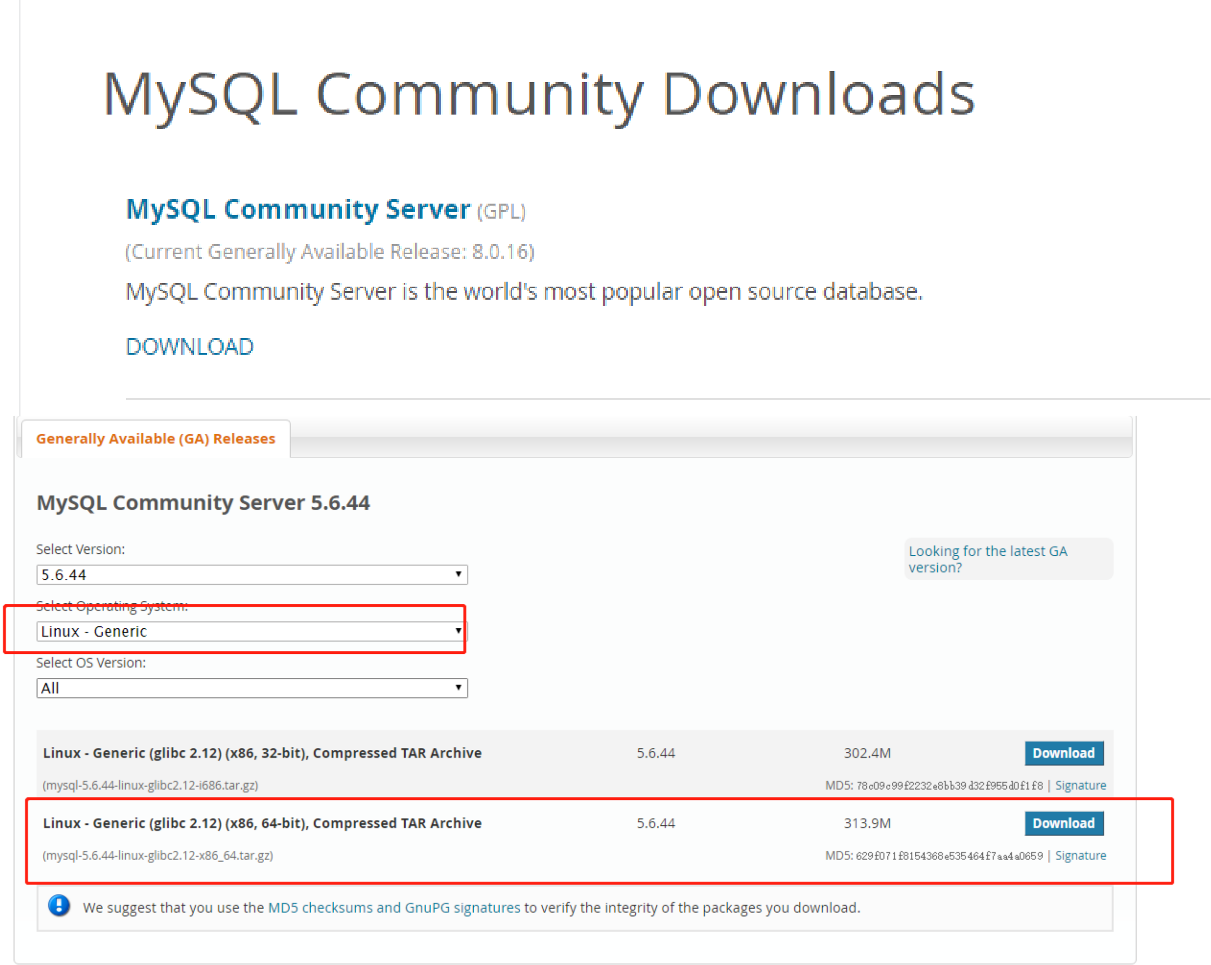
MySQL Cluster CGE (commercial)

MySQL Cluster is a real-time open source transactional database designed for fast, always-on access to data under high throughput conditions.

- MySQL Cluster
- MySQL Cluster Manager
- Plus, everything in MySQL Enterprise Edition

Learn More »
Customer Download » (Select Patches & Updates Tab, Product Search)
Trial Download » (Note - Select Product Pack: MySQL Database)

MySQL Community Edition (GPL)
Community (GPL) Downloads »



MySQL Community Downloads

MySQL Community Server (GPL)
(Current Generally Available Release: 8.0.16)
MySQL Community Server is the world's most popular open source database.
DOWNLOAD

Generally Available (GA) Releases

MySQL Community Server 5.6.44

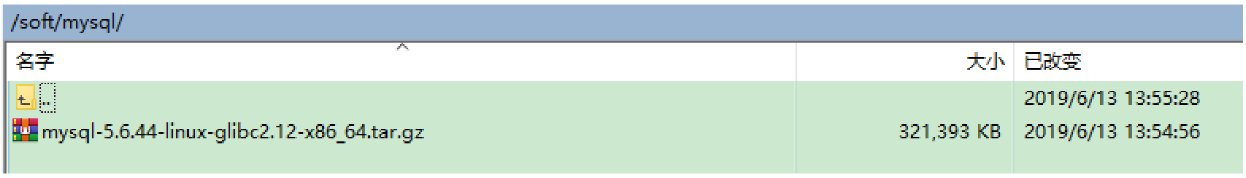
Select Version: 5.6.44
Select Operating System: Linux - Generic
Select OS Version: All

Looking for the latest GA version?

Linux - Generic (glibc 2.12) (x86, 32-bit), Compressed TAR Archive (mysql-5.6.44-linux-glibc2.12-i686.tar.gz)	5.6.44	302.4M	Download MD5: 78c09c99f2232e8bb39d32f955d0f1f8 | Signature
Linux - Generic (glibc 2.12) (x86, 64-bit), Compressed TAR Archive (mysql-5.6.44-linux-glibc2.12-x86_64.tar.gz)	5.6.44	313.9M	Download MD5: 629f071f8154368e535464f7aa4a0659 | Signature

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

2. 将下载的mysqlxxx.tar.gz上传至服务器/soft/mysql

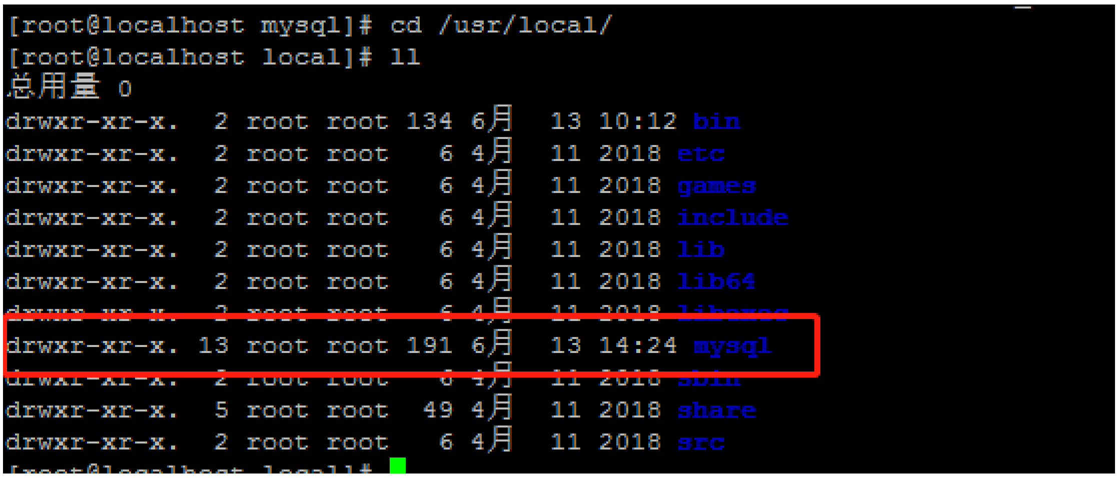


/soft/mysql/

名字	大小	已改变
..		2019/6/13 13:55:28
mysql-5.6.44-linux-glibc2.12-x86_64.tar.gz	321,393 KB	2019/6/13 13:54:56

3. 解压缩：tar －zxvf mysqlxxx.tar.gz

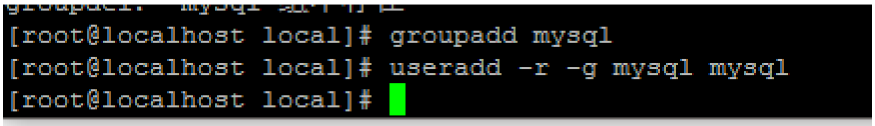
4. 将解压后的文件复制到/usr/local/mysql下：cp mysqlxxx /usr/local/mysql -r



```
[root@localhost mysql]# cd /usr/local/
[root@localhost local]# ll
总用量 0
drwxr-xr-x.  2 root root 134 6月  13 10:12 bin
drwxr-xr-x.  2 root root   6 4月  11 2018 etc
drwxr-xr-x.  2 root root   6 4月  11 2018 games
drwxr-xr-x.  2 root root   6 4月  11 2018 include
drwxr-xr-x.  2 root root   6 4月  11 2018 lib
drwxr-xr-x.  2 root root   6 4月  11 2018 lib64
drwxr-xr-x.  2 root root   6 4月  11 2018 libexec
drwxr-xr-x. 13 root root 191 6月  13 14:24 mysql
drwxr-xr-x.  2 root root   6 4月  11 2018 sbin
drwxr-xr-x.  5 root root  49 4月  11 2018 share
drwxr-xr-x.  2 root root   6 4月  11 2018 src
[root@localhost local]#
```

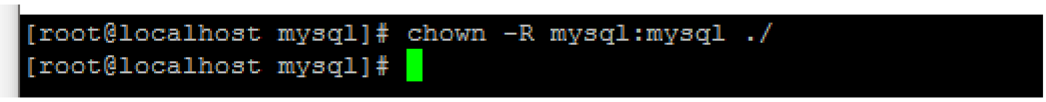
5. 删除压缩包：rm －rf mysqlxxx.tar.gz，和解压后的文件

6. 在mysql上一层目录[即在/usr/local目录]添加系统mysql组和mysql用户：groupadd mysql和useradd －g mysql mysql



```
[root@localhost local]# groupadd mysql
[root@localhost local]# useradd -r -g mysql mysql
[root@localhost local]#
```

7. 进入mysql软件目录：cd /usr/local/mysql，修改当前目录拥有者为mysql用户：chown －R mysql:mysql ./



```
[root@localhost mysql]# chown -R mysql:mysql ./
[root@localhost mysql]#
```

8. 安装数据库：./scripts/mysql_install_db －user=mysql

a.若提示-bash: ./scripts/mysql_install_db: /usr/bin/perl: 坏的解释器: 没有那个文件或目录，则执行yum -y install perl perl-devel，

b.若之后提醒FATAL ERROR: please install the following Perl modules before executing ./scripts/mysql_install_db:

Data::Dumper

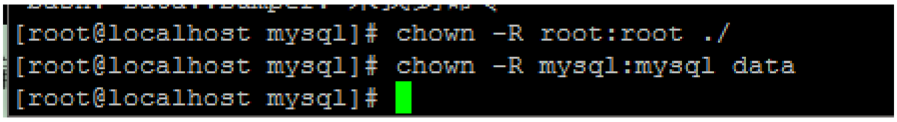
则执行yum install -y perl-Data-Dumper 即可。

c.若提示Installing MySQL system tables..../bin/mysqld: error while loading shared libraries: libaio.so.1: cannot open shared object file: No such file or directory，原因是没有安装libaio.so.1，执行yum install -y libaio即可

c.解决上述问题后执行./scripts/mysql_install_db －user=mysql安装数据库

9.修改当前目录拥有者为root用户：chown -R root:root ./

10.修改当前data目录拥有者为mysql用户：chown -R mysql:mysql data

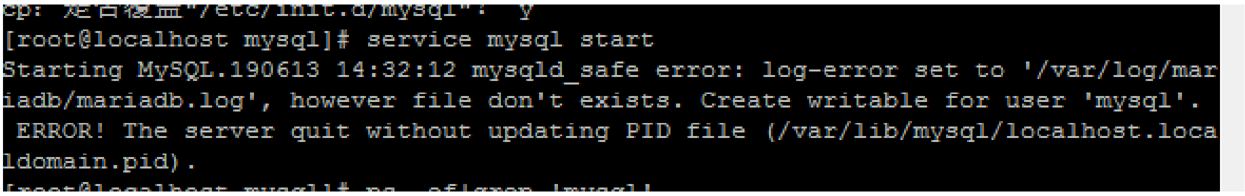


```
[root@localhost mysql]# chown -R root:root ./
[root@localhost mysql]# chown -R mysql:mysql data
[root@localhost mysql]#
```

11.添加开机启动mysql服务：cp support-files/mysql.server /etc/init.d/mysql，把启动脚本放到开机目录

12.启动mysql服务：service mysql start

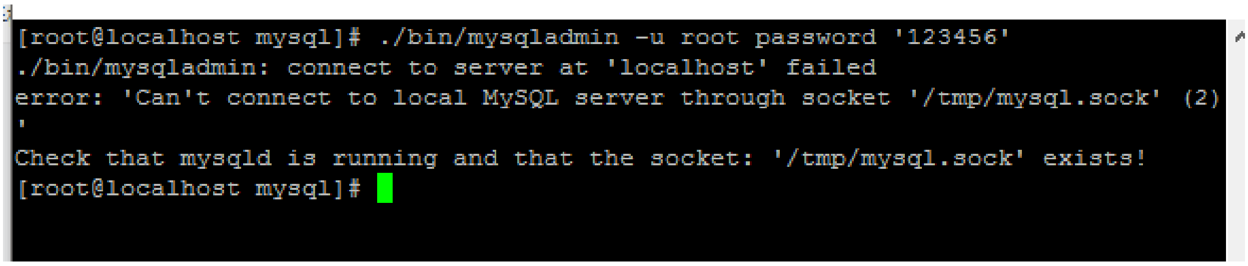
若提示一下错误，则只需创建对应的的文件夹和文件即可：cd /var/log/，mkdir mariadb，cd mariadb/，touch mariadb.log；然后service mysql start 启动mysql



```
cp: 是否覆盖"/etc/init.d/mysql"? y
[root@localhost mysql]# service mysql start
Starting MySQL.190613 14:32:12 mysqld_safe error: log-error set to '/var/log/mar
iadb/mariadb.log', however file don't exists. Create writable for user 'mysql'.
 ERROR! The server quit without updating PID file (/var/lib/mysql/localhost.loca
ldomain.pid).
[root@localhost mysql]# ps -ef|grep 'mysql'
```

13. 修改mysql的root密码，进入usr/local/mysql目录，默认初始化密码为空：./bin/mysqladmin -u root password ’密码’

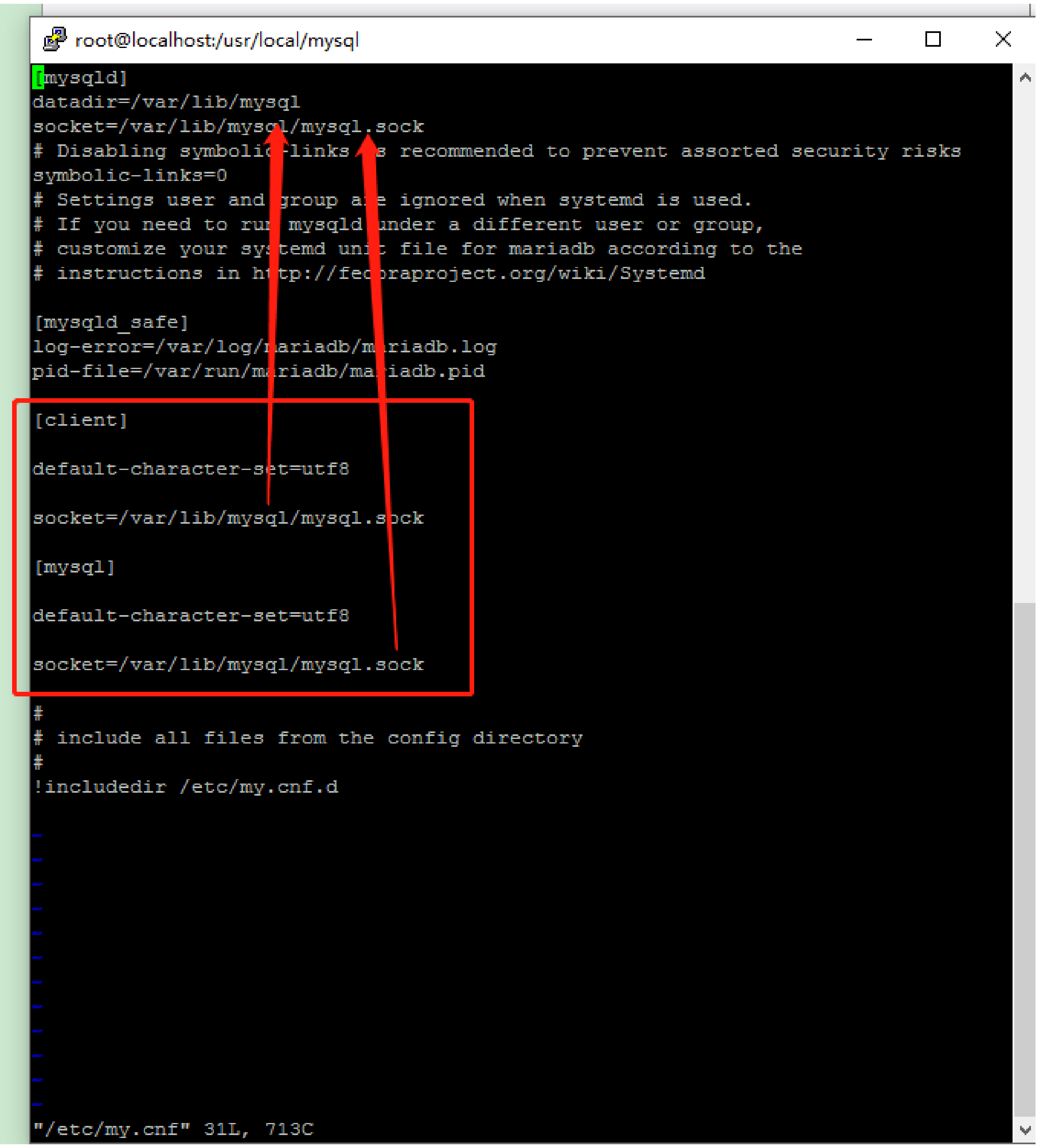
若出现以下错误：



```
[root@localhost mysql]# ./bin/mysqladmin -u root password '123456'
./bin/mysqladmin: connect to server at 'localhost' failed
error: 'Can't connect to local MySQL server through socket '/tmp/mysql.sock' (2)
'
Check that mysqld is running and that the socket: '/tmp/mysql.sock' exists!
[root@localhost mysql]# 
```

则需完善mysql配置文件，编辑/etc/my.cnf文件：vi /etc/my.cnf，在配置文件添加[client]和[mysql]选项，如图

```
[client]

default-character-set=utf8

socket=/var/lib/mysql/mysql.sock

[mysql]

default-character-set=utf8

socket=/var/lib/mysql/mysql.sock
```



```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
# Disabling symbolic-links is recommended to prevent assorted security risks
symbolic-links=0
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd

[mysqld_safe]
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid

[client]

default-character-set=utf8

socket=/var/lib/mysql/mysql.sock

[mysql]

default-character-set=utf8

socket=/var/lib/mysql/mysql.sock

#
# include all files from the config directory
#
!includedir /etc/my.cnf.d

~
~
~
~
~
~
~
~
~
~
~
~
"/etc/my.cnf" 31L, 713C
```

15.解决完上述问题后继续修改密码：./bin/mysqladmin -u root password ’密码’

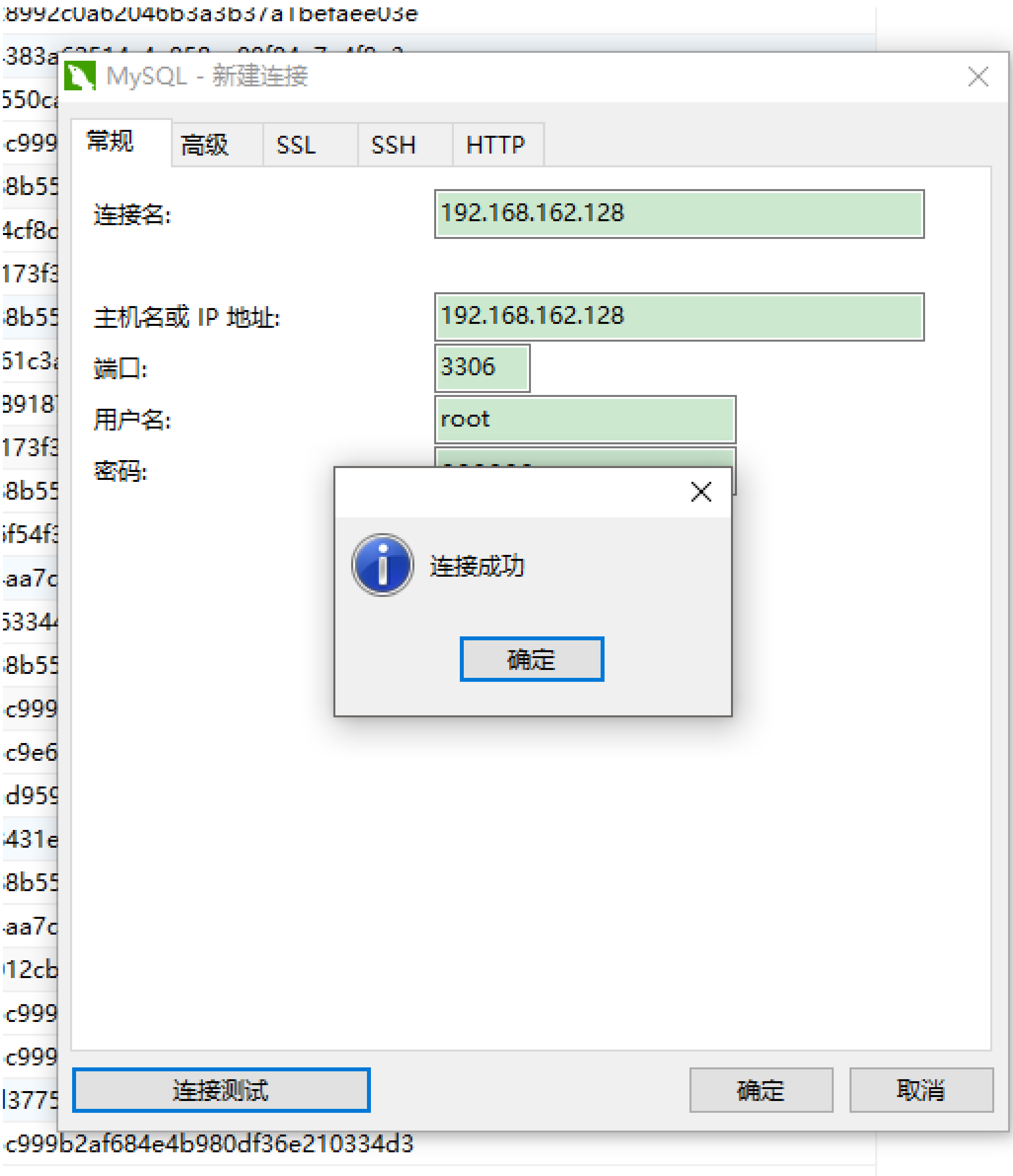
14.设置mysql允许远程连接

进入mysql命令行：/usr/local/mysql/bin/mysql －u root －p 123456

grant all privileges on *.* to ’root’@’%’ identified by ’密码’ with grant option;

flush privileges;

15.测试是否能够连接上



MySQL - 新建连接

连接名: 192.168.162.128

主机名或 IP 地址: 192.168.162.128

端口: 3306

用户名: root

密码:

连接成功

c999b2af684e4b980df36e210334d3

Centos7查看防火墙状态： firewall-cmd －state； 关闭防火墙： systemctl stop firewalld.service；禁止firewall开机启动：systemctl disable firewalld.service