1 前言

1、安装 VMware station，虚拟机

2、配置 CentOS

3、安装 JDK 8

4、安装 MySQL

5、安装 Tomcat

6、安装 Xshell

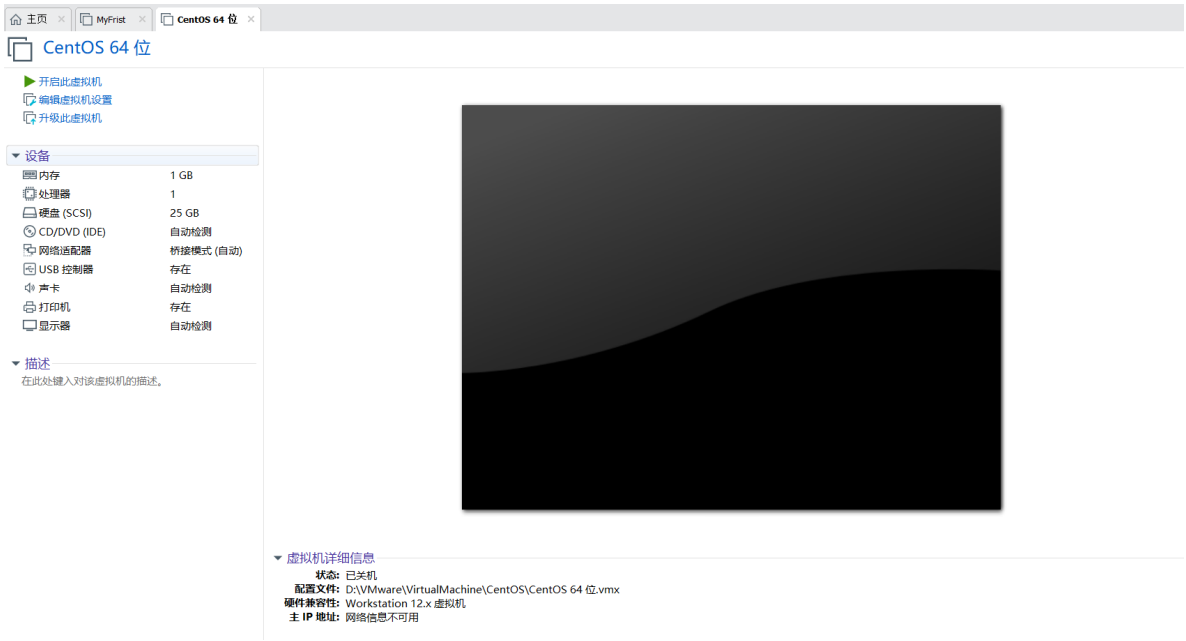
7、安装 Xftp

2 安装 VMware

3 导入 CentOS 7

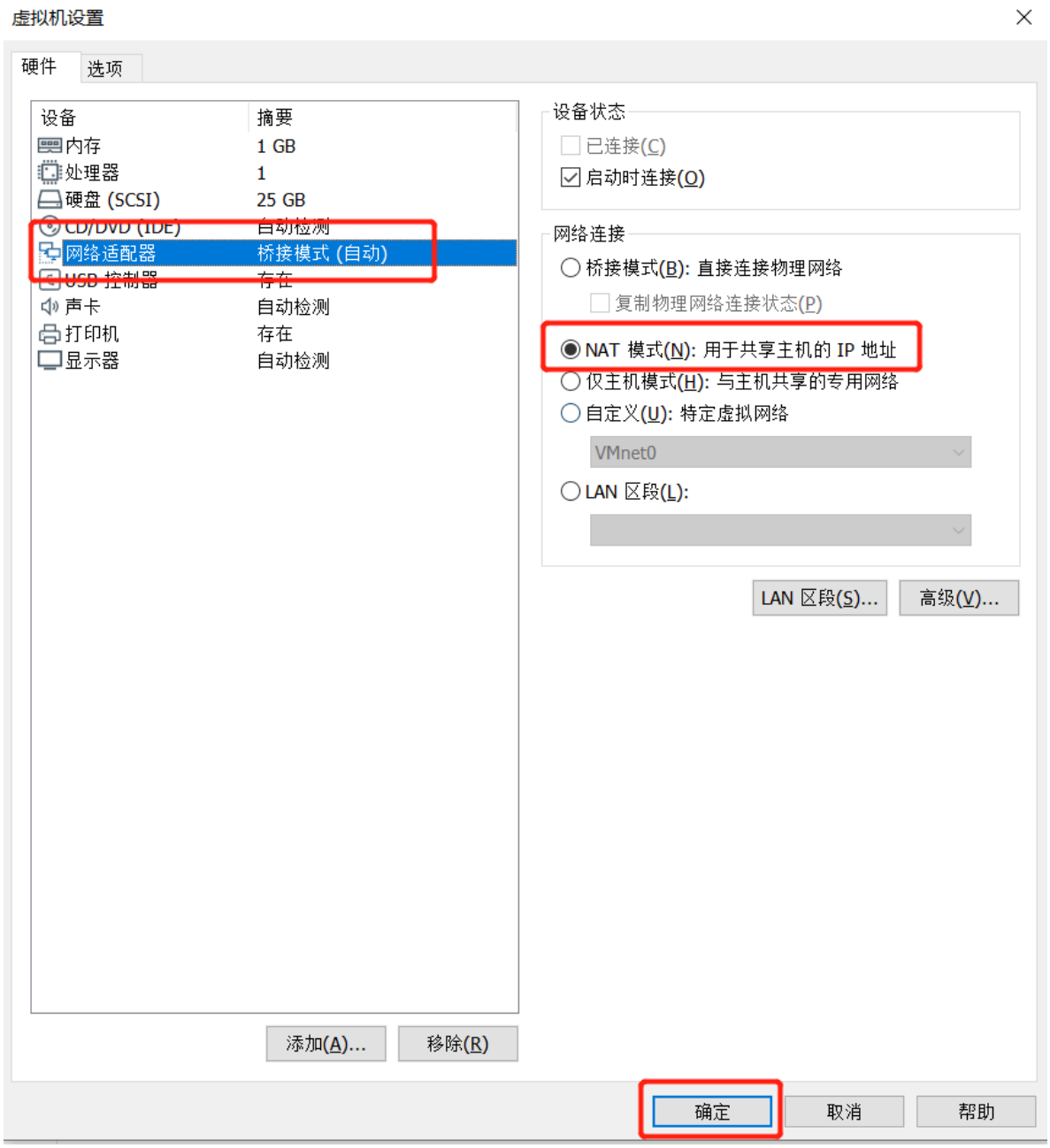
1、解压 CentOS

2、打开 VM，导入 CentOS.vmx



将网络设置为 NAT 模式。

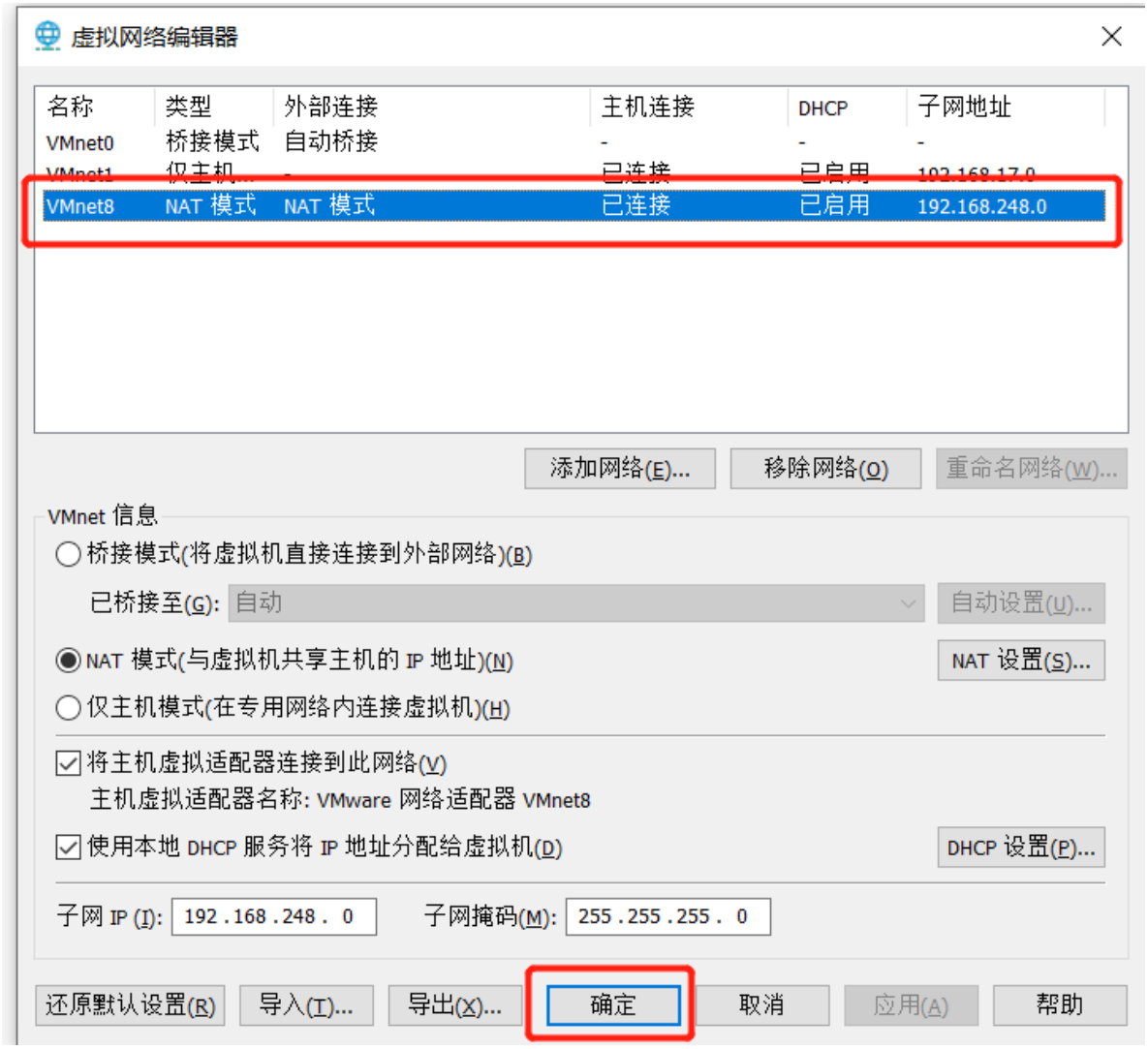
3、虚拟机设置



4、在虚拟网络编辑器中修改网络设置，必须以管理员身份修改。

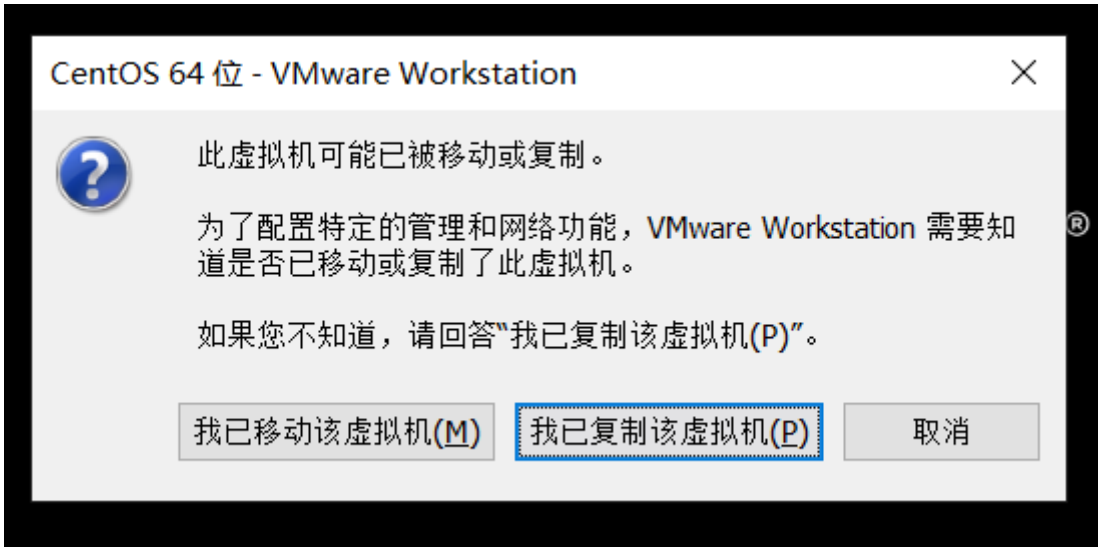

虚拟网络编辑器
名称 类型 外部连接 主机连接 DHCP 子网地址
VMnet1 仅主机... - 已连接 已启用 192.168.17.0
VMnet8 NAT 模式 NAT 模式 已连接 已启用 192.168.248.0
添加网络(E)...
移除网络(O)
重命名网络(W)...
VMnet 信息
桥接模式(将虚拟机直接连接到外部网络)(B)
已桥接至(G):
自动设置(U)...
NAT 模式(与虚拟机共享主机的 IP 地址)(N)
NAT 设置(S)...
仅主机模式(在专用网络内连接虚拟机)(H)
将主机虚拟适配器连接到此网络(V)
主机虚拟适配器名称: VMware 网络适配器 VMnet1
使用本地 DHCP 服务将 IP 地址分配给虚拟机(D)
DHCP 设置(P)...
子网 IP (I): 192 . 168 . 17 . 0
子网掩码(M): 255 . 255 . 255 . 0
需要具备管理员特权才能修改网络配置。
更改设置(C)
还原默认设置(R)
导入(T)...
导出(X)...
确定
取消
应用(A)
帮助

选择 NAT 模式。

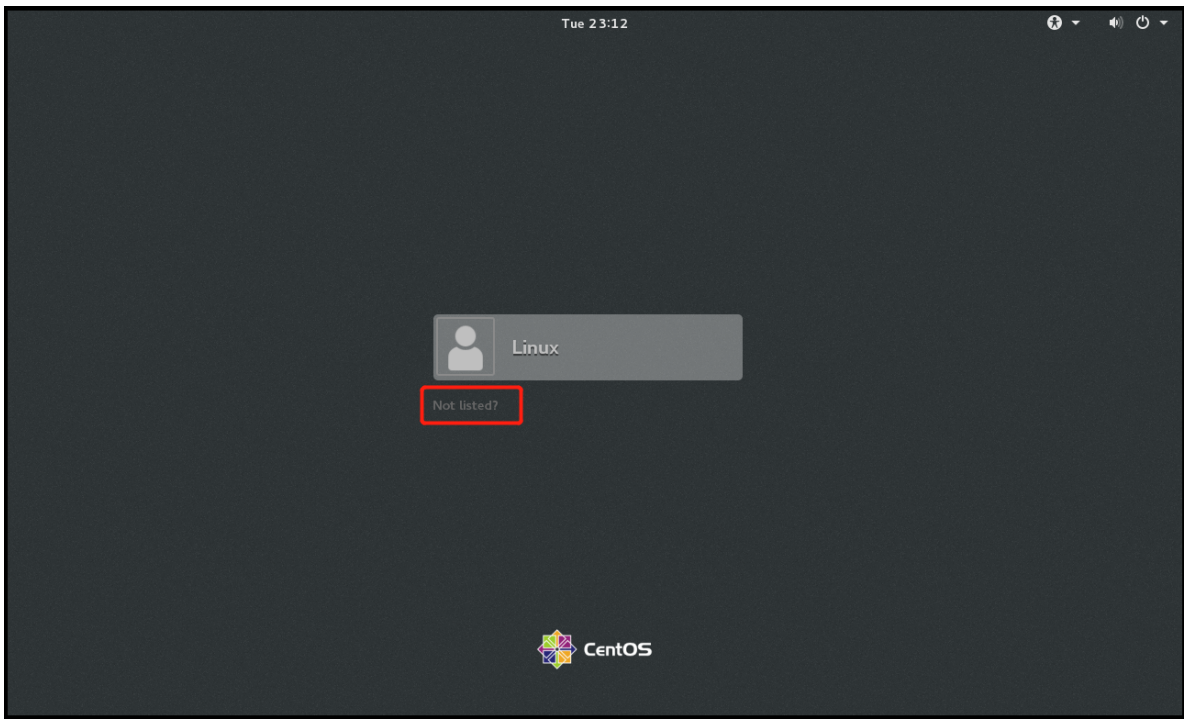

虚拟网络编辑器
名称 类型 外部连接 主机连接 DHCP 子网地址
VMnet0 桥接模式 自动桥接 - - -
VMnet8 NAT 模式 NAT 模式 已连接 已启用 192.168.248.0
添加网络(E)...
移除网络(O)
重命名网络(W)...
VMnet 信息
桥接模式(将虚拟机直接连接到外部网络)(B)
已桥接至(G): 自动
自动设置(U)...
NAT 模式(与虚拟机共享主机的 IP 地址)(N)
NAT 设置(S)...
仅主机模式(在专用网络内连接虚拟机)(H)
将主机虚拟适配器连接到此网络(V)
主机虚拟适配器名称: VMware 网络适配器 VMnet8
使用本地 DHCP 服务将 IP 地址分配给虚拟机(D)
DHCP 设置(P)...
子网 IP (I): 192 . 168 . 248 . 0
子网掩码(M): 255 . 255 . 255 . 0
还原默认设置(R)
导入(T)...
导出(X)...
确定
取消
应用(A)
帮助

5、开启虚拟机

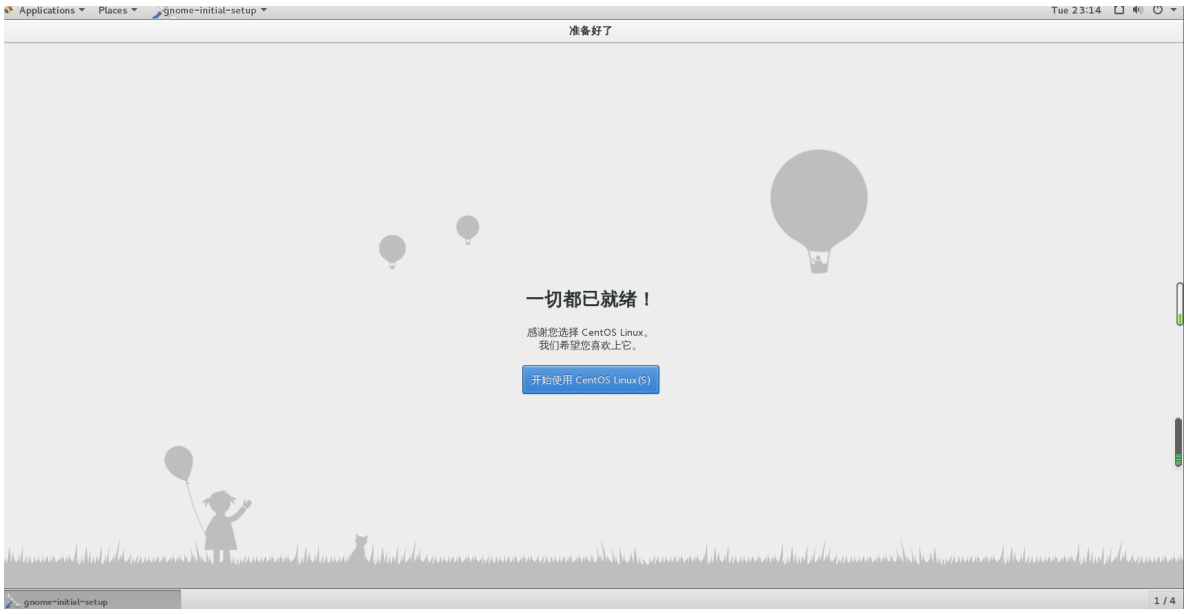
6、点击我已复制该虚拟机

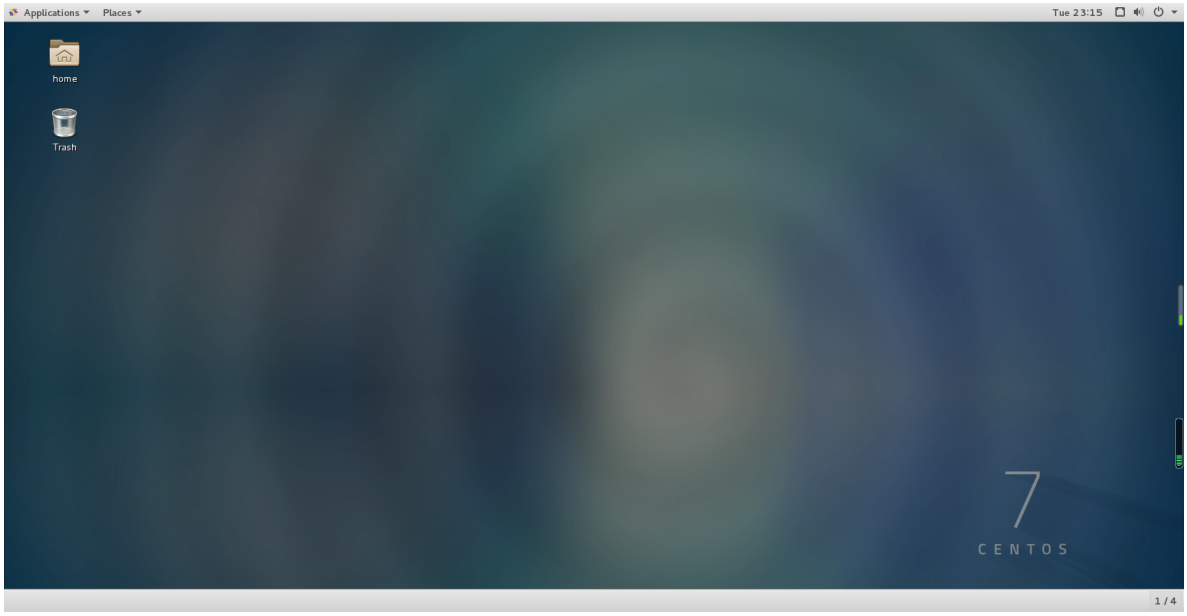


7、点击 Not listed，使用 root 权限登录，输入用户名（root）和密码（123456）。



8、初始化选择语言，输入法等设置，完毕之后来到欢迎页面，点击开始使用 CentOS Linux(S) 进入桌面。

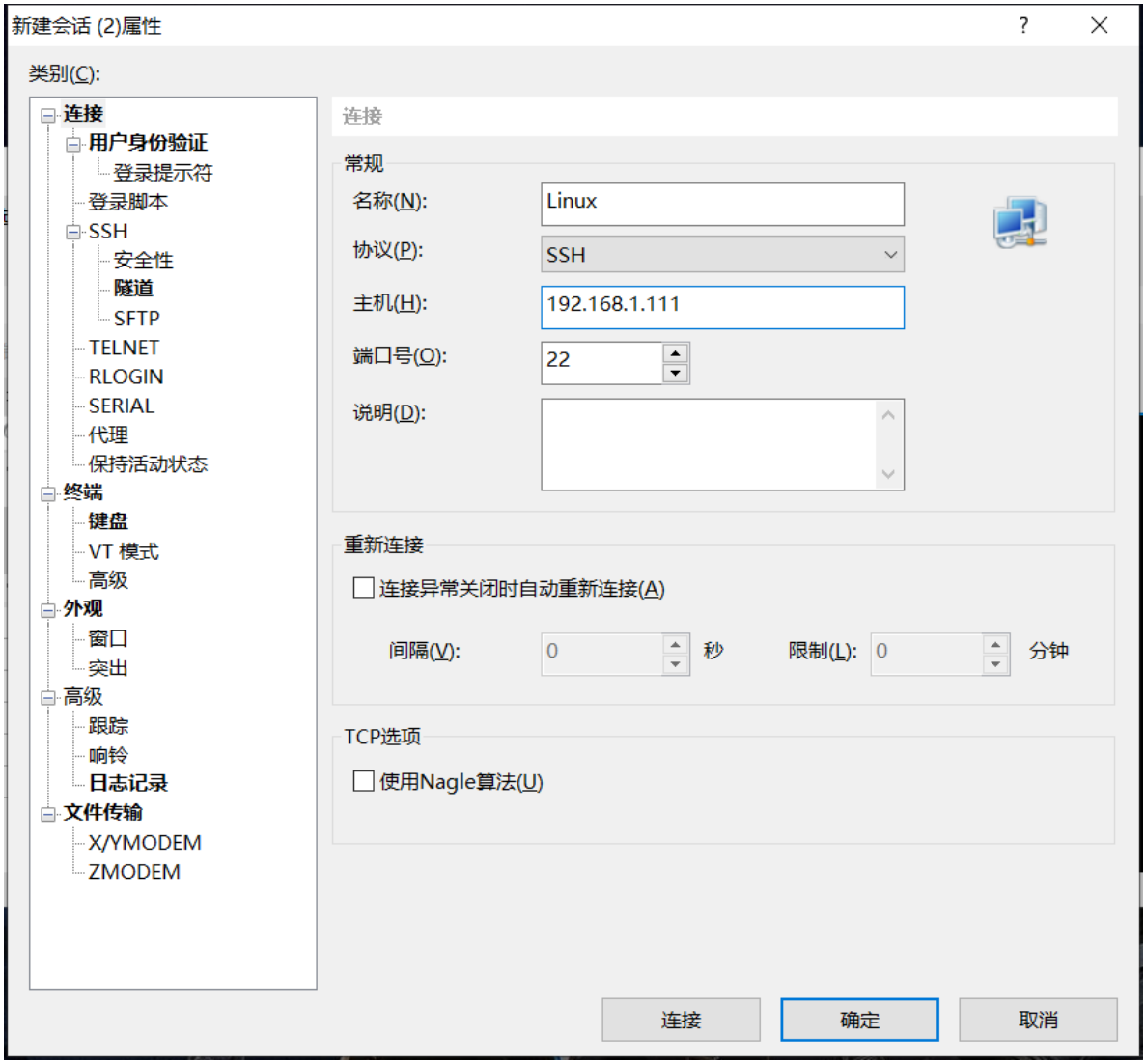




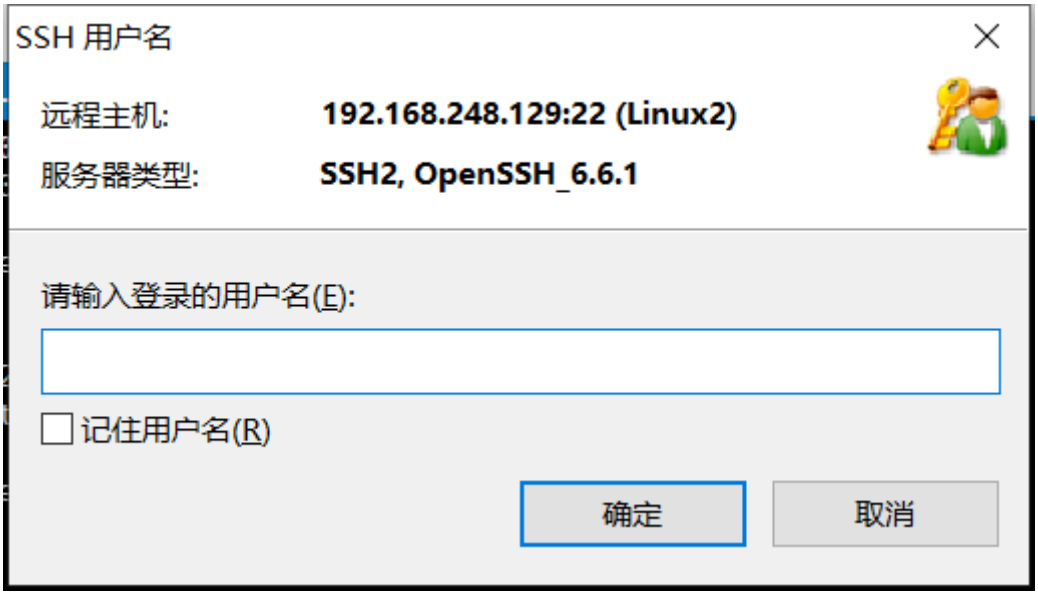
9、通过终端命令查看 CentOS IP 地址，如 192.168.1.111。

4 安装 Xshell

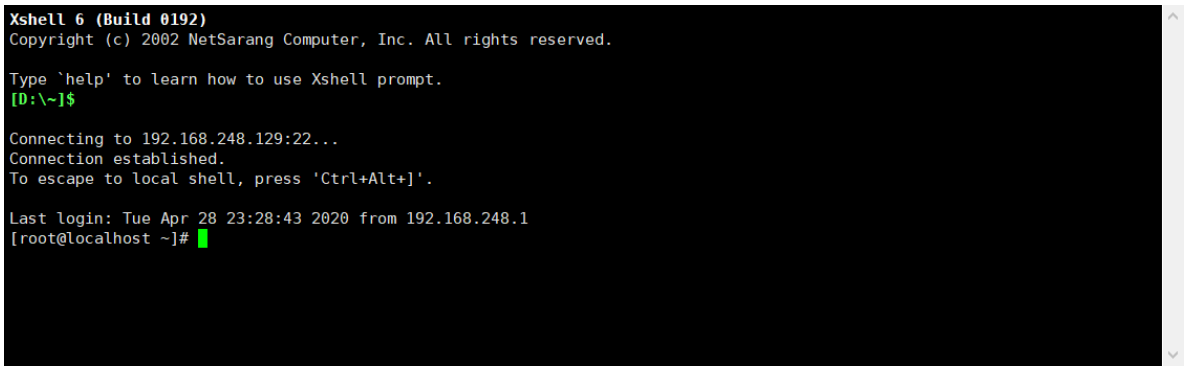
1、打开 Xshell，新建会话，输入 CentOS IP，端口默认为 22，点击连接。


新建会话 (2)属性
类别(C):
连接
用户身份验证
登录提示符
登录脚本
SSH
安全性
隧道
SFTP
TELNET
RLOGIN
SERIAL
代理
保持活动状态
终端
键盘
VT 模式
高级
外观
窗口
突出
高级
跟踪
响铃
日志记录
文件传输
X/YMODEM
ZMODEM
连接
常规
名称(N):
Linux
协议(P):
SSH
主机(H):
192.168.1.111
端口号(O):
22
说明(D):
重新连接
连接异常关闭时自动重新连接(A)
间隔(V):
0
秒
限制(L):
0
分钟
TCP选项
使用Nagle算法(U)
连接
确定
取消

2、输入用户名密码。


SSH 用户名
远程主机:
192.168.248.129:22 (Linux2)
服务器类型:
SSH2, OpenSSH_6.6.1
请输入登录的用户名(E):
记住用户名(R)
确定
取消

3、连接成功。



5 安装 JDK

1、卸载 CentOS 自带的 OpenJDK，可以通过命令查看。

```
rpm -qa | grep jdk
```

```
Xshell 6 (Build 0192)
Copyright (c) 2002 NetSarang Computer, Inc. All rights reserved.

Type `help' to learn how to use Xshell prompt.
[D:\~]$

Connecting to 192.168.248.129:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Last login: Tue Apr 28 23:28:43 2020 from 192.168.248.1
[root@localhost ~]# rpm -qa | grep jdk
java-1.8.0-openjdk-headless-1.8.0.65-3.b17.el7.x86_64
java-1.8.0-openjdk-1.8.0.65-3.b17.el7.x86_64
java-1.7.0-openjdk-1.7.0.91-2.6.2.3.el7.x86_64
java-1.7.0-openjdk-headless-1.7.0.91-2.6.2.3.el7.x86_64
[root@localhost ~]#
```

2、通过命令删除。

```
rpm -e --nodeps java-1.8.0-openjdk-1.8.0.65-3.b17.el7.x86_64
rpm -e --nodeps java-1.8.0-openjdk-headless-1.8.0.65-3.b17.el7.x86_64
rpm -e --nodeps java-1.7.0-openjdk-1.7.0.91-2.6.2.3.el7.x86_64
rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.91-2.6.2.3.el7.x86_64
```

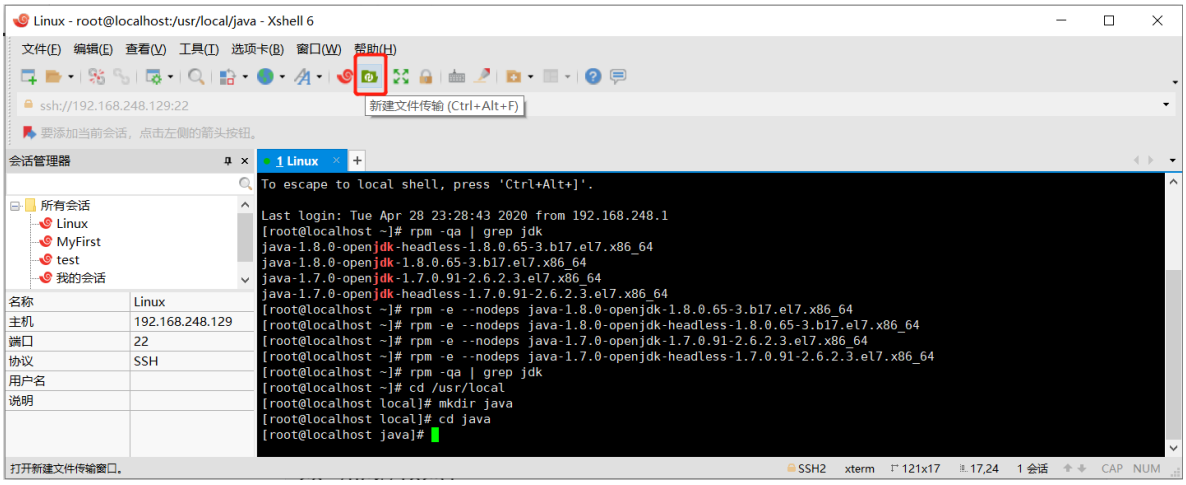
3、查看是否删除成功。

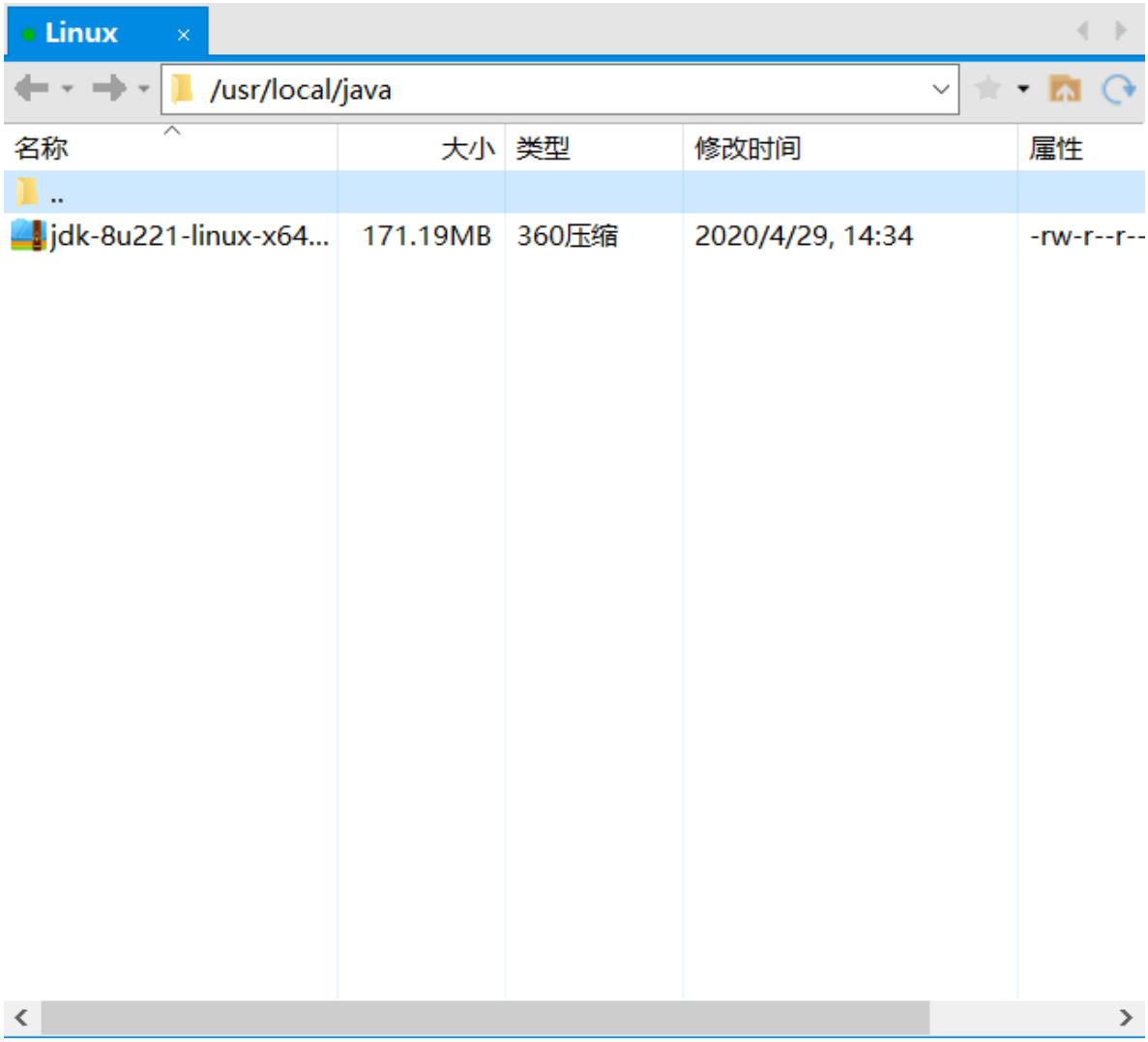
```
[root@localhost ~]# rpm -qa | grep jdk
[root@localhost ~]#
```

4、在 /usr/local 目录下新建 java 文件夹，并进入

```
cd /usr/local
mkdir java
cd java
```

5、打开 Xftp，将 JDK 安装文件拷贝到 java 目录。





6、使用命令进行安装

```
rpm -ivh jdk-8u221-linux-x64.rpm
```

```
[root@localhost local]# mkdir java
[root@localhost local]# cd java
[root@localhost java]# rpm -ivh jdk-8u221-linux-x64.rpm
warning: jdk-8u221-linux-x64.rpm: Header V3 RSA/SHA256 Signature, key ID ec551f03: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:jdk1.8-2000:1.8.0_221-fcs        ################################# [100%]
Unpacking JAR files...
        tools.jar...
        plugin.jar...
        javaws.jar...
        deploy.jar...
        rt.jar...
        jsse.jar...
        charsets.jar...
        localedata.jar...
[root@localhost java]#
```

7、配置环境变量

```
vim /etc/profile
```

```
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
"/etc/profile" 76L, 1750C                                                 1,1           Top
```

按 i 进行输入，复制如下内容

```
JAVA_HOME=/usr/java/jdk1.8.0_221-amd64
CLASSPATH=%JAVA_HOME%/lib:%JAVA_HOME%/jre/lib
PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin
export PATH CLASSPATH JAVA_HOME
```

```
for i in /etc/profile.d/*.sh ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
JAVA_HOME=/usr/java/jdk1.8.0_221-amd64
CLASSPATH=%JAVA_HOME%/lib:%JAVA_HOME%/jre/lib
PATH=$PATH:$JAVA_HOME/bin:$JAVA_HOME/jre/bin
export PATH CLASSPATH JAVA_HOME
-- INSERT --                                                              80,32         Bot
```

按 Esc 退出，输入 :wq，回车退出保存。

8、让配置生效

```
source /etc/profile
```

9、检查安装是否成功

```
java -version
```

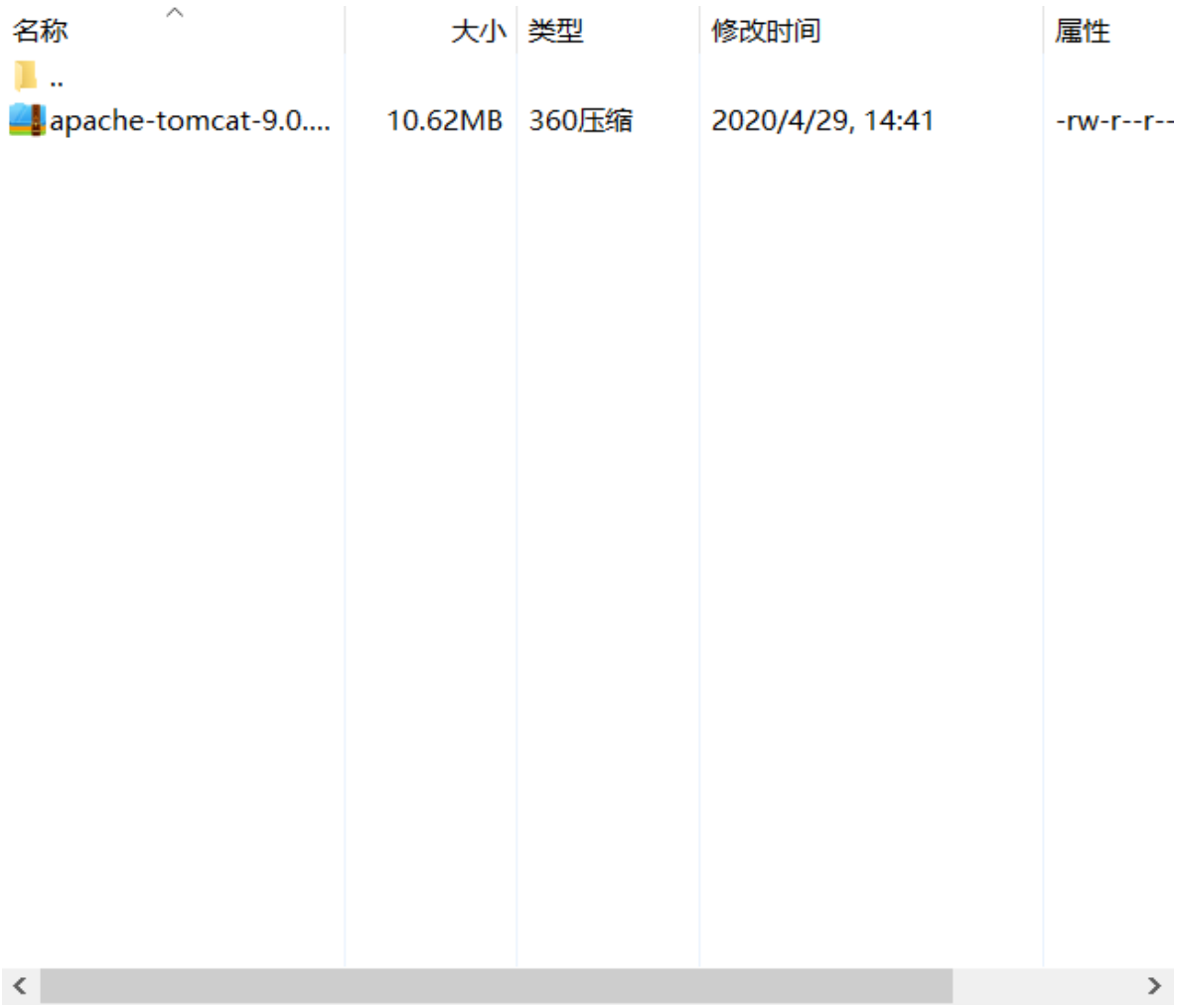
```
   1:jdk1.8-2000:1.8.0_221-fcs        ################################# [100%]
Unpacking JAR files...
        tools.jar...
        plugin.jar...
        javaws.jar...
        deploy.jar...
        rt.jar...
        jsse.jar...
        charsets.jar...
        localedata.jar...
[root@localhost java]# vim /etc/profile
[root@localhost java]# source /etc/profile
[root@localhost java]# java -version
java version "1.8.0_221"
Java(TM) SE Runtime Environment (build 1.8.0_221-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.221-b11, mixed mode)
[root@localhost java]#
```

6 安装 Tomcat

1、在 /usr/local 目录下新建 tomcat 文件夹，并进入。

```
cd /usr/local
mkdir tomcat
cd tomcat
```

2、打开 Xftp，将 Tomcat 压缩文件拷贝到 tomcat 目录。



3、解压缩。

```
tar -zxvf apache-tomcat-9.0.34.tar.gz
```

```
apache-tomcat-9.0.34/webapps/manager/images/asf-logo.svg
apache-tomcat-9.0.34/webapps/manager/images/tomcat.gif
apache-tomcat-9.0.34/webapps/manager/index.jsp
apache-tomcat-9.0.34/webapps/manager/status.xsd
apache-tomcat-9.0.34/webapps/manager/xform.xsl
apache-tomcat-9.0.34/bin/catalina.sh
apache-tomcat-9.0.34/bin/ciphers.sh
apache-tomcat-9.0.34/bin/configtest.sh
apache-tomcat-9.0.34/bin/daemon.sh
apache-tomcat-9.0.34/bin/digest.sh
apache-tomcat-9.0.34/bin/makebase.sh
apache-tomcat-9.0.34/bin/setclasspath.sh
apache-tomcat-9.0.34/bin/shutdown.sh
apache-tomcat-9.0.34/bin/startup.sh
apache-tomcat-9.0.34/bin/tool-wrapper.sh
apache-tomcat-9.0.34/bin/version.sh
[root@localhost tomcat]#
```

4、进入 /apache-tomcat-9.0.34/bin 路径启动 Tomcat。

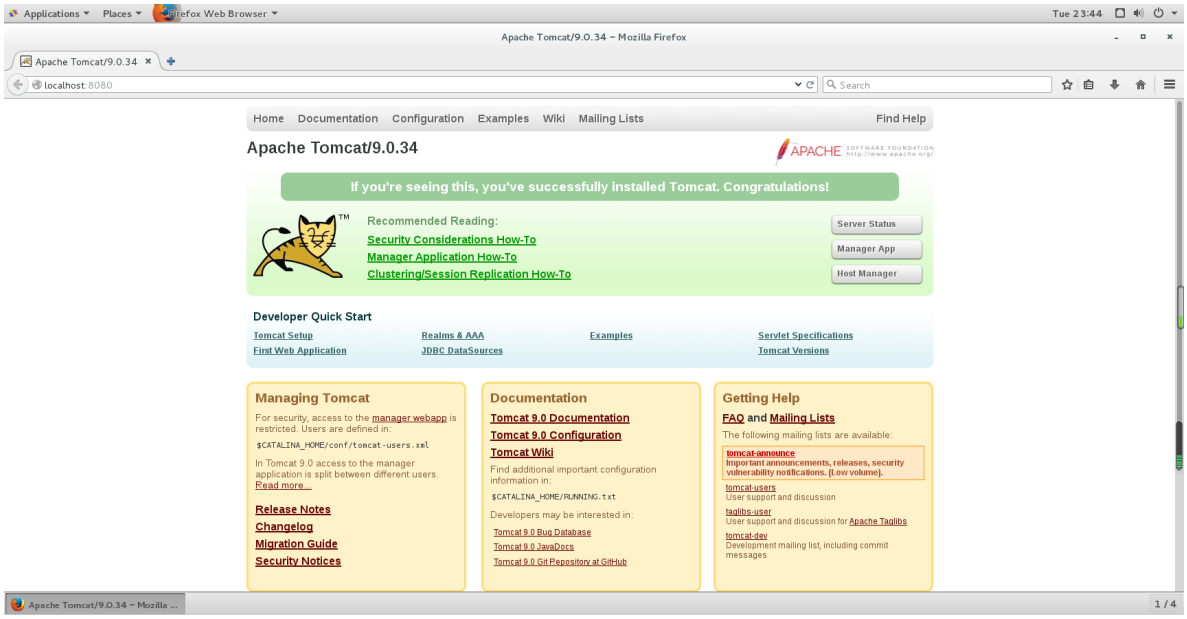
```
./startup.sh
```

```
BUILDING.txt  CONTRIBUTING.md  LICENSE  NOTICE  RELEASE-NOTES  temp        work
[root@localhost apache-tomcat-9.0.34]# cd bin
[root@localhost bin]# ls
bootstrap.jar       ciphers.sh                daemon.sh     setclasspath.bat  startup.sh            version.bat
catalina.bat        commons-daemon.jar        digest.bat    setclasspath.sh   tomcat-juli.jar       version.sh
catalina.sh         commons-daemon-native.tar.gz  digest.sh     shutdown.bat      tomcat-native.tar.gz
catalina-tasks.xml  configtest.bat            makebase.bat  shutdown.sh       tool-wrapper.bat
ciphers.bat         configtest.sh             makebase.sh   startup.bat       tool-wrapper.sh
[root@localhost bin]# ./startup.sh
Using CATALINA_BASE:   /usr/local/tomcat/apache-tomcat-9.0.34
Using CATALINA_HOME:   /usr/local/tomcat/apache-tomcat-9.0.34
Using CATALINA_TMPDIR: /usr/local/tomcat/apache-tomcat-9.0.34/temp
Using JRE_HOME:        /usr/java/jdk1.8.0_221-amd64
Using CLASSPATH:       /usr/local/tomcat/apache-tomcat-9.0.34/bin/bootstrap.jar:/usr/local/tomcat/apache-tomcat-9.0.34/bi
n/tomcat-juli.jar
Tomcat started.
[root@localhost bin]#
```

关闭 Tomcat

```
./shutdown.sh
```

启动成功之后，可以在 CentOS 中直接访问。

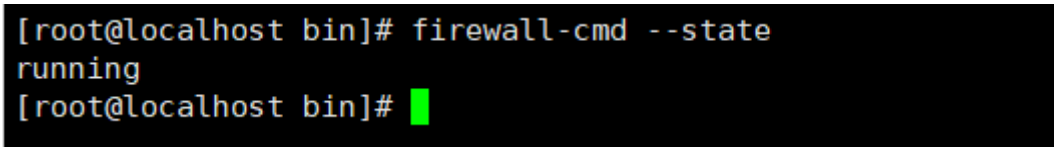


但是在本地无法访问，这是因为 CentOS 没有开放 8080 端口导致的。

5、CentOS 开放 8080 端口。

1、查看防火墙状态

```
firewall-cmd --state
```



runing 表示开启，not runing 表示关闭，如果关闭，执行

```
systemctl start firewalld.service
```

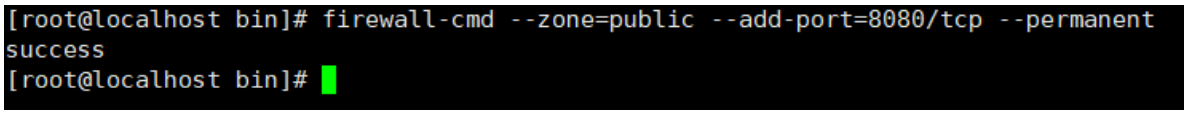
2、开启 8080 端口

```
firewall-cmd --zone=public --add-port=8080/tcp --permanent
```

--zone=public：表示作用域为公共的；

--add-port=8080/tcp：添加tcp协议的端口8080；

--permanent：永久生效，如果没有此参数，则只能维持当前服务生命周期内，重新启动后失效；



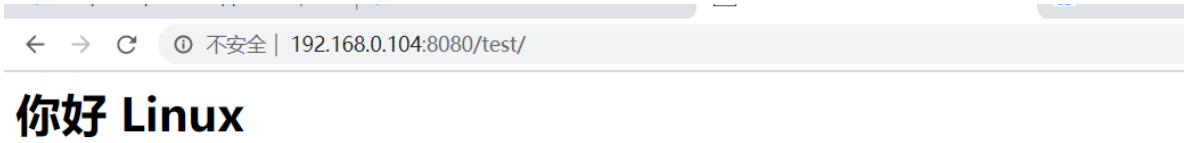
3、输入命令重启防火墙

```
systemctl restart firewalld.service
```

4、输入命令重新载入配置

```
firewall-cmd --reload
```

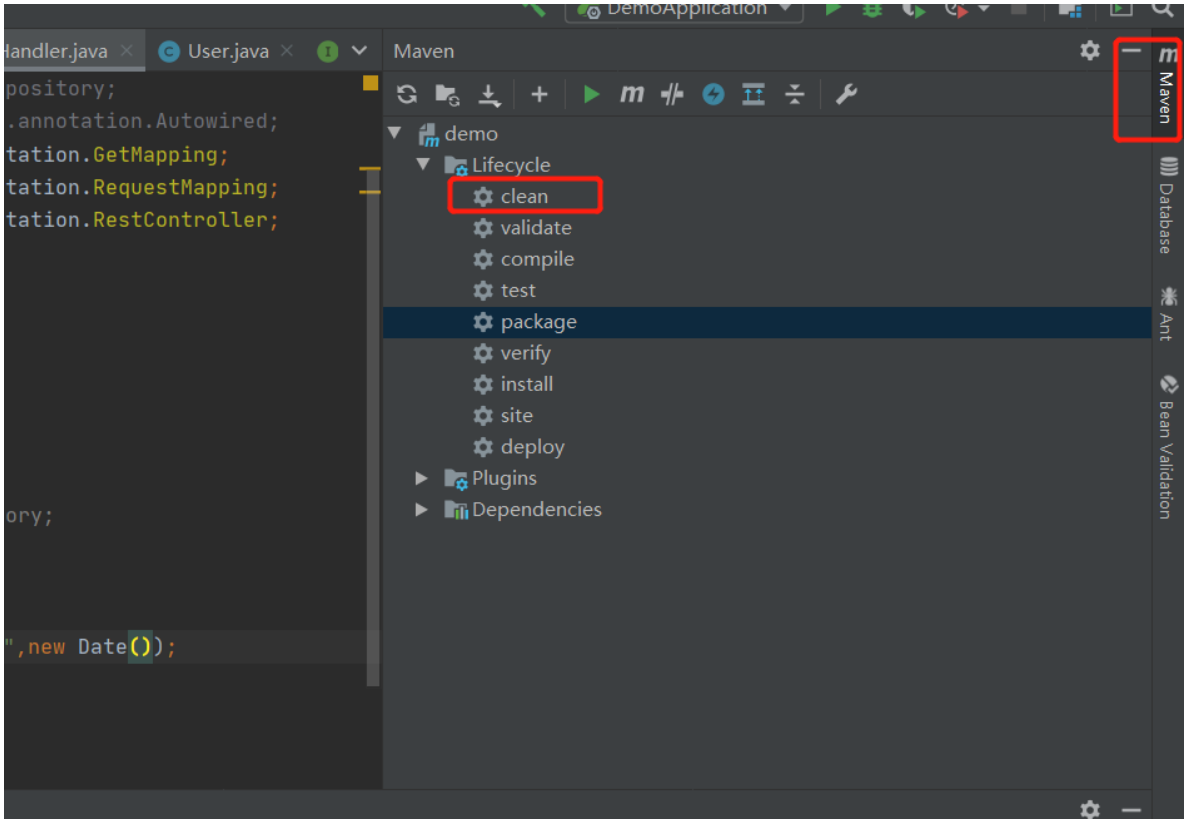
5、打开浏览器，访问成功。



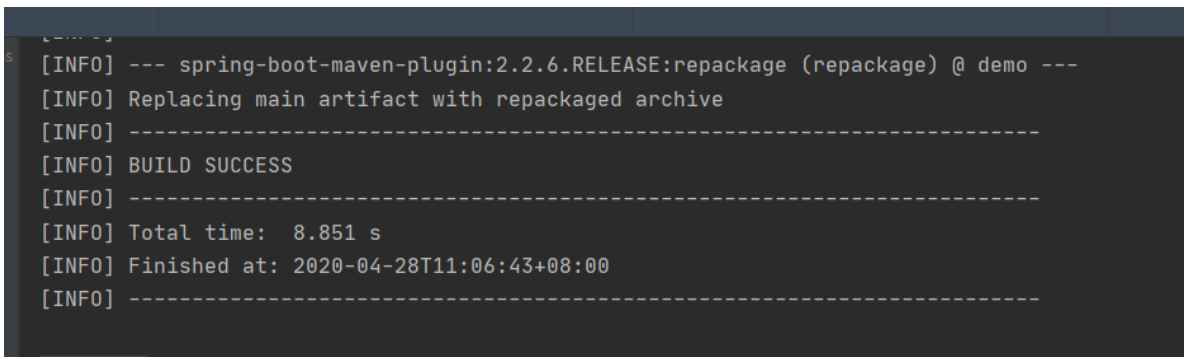
6、将打包之后的 war 拷贝到 Tomcat/webapps 下，启动访问即可。

7 部署 Spring Boot 应用

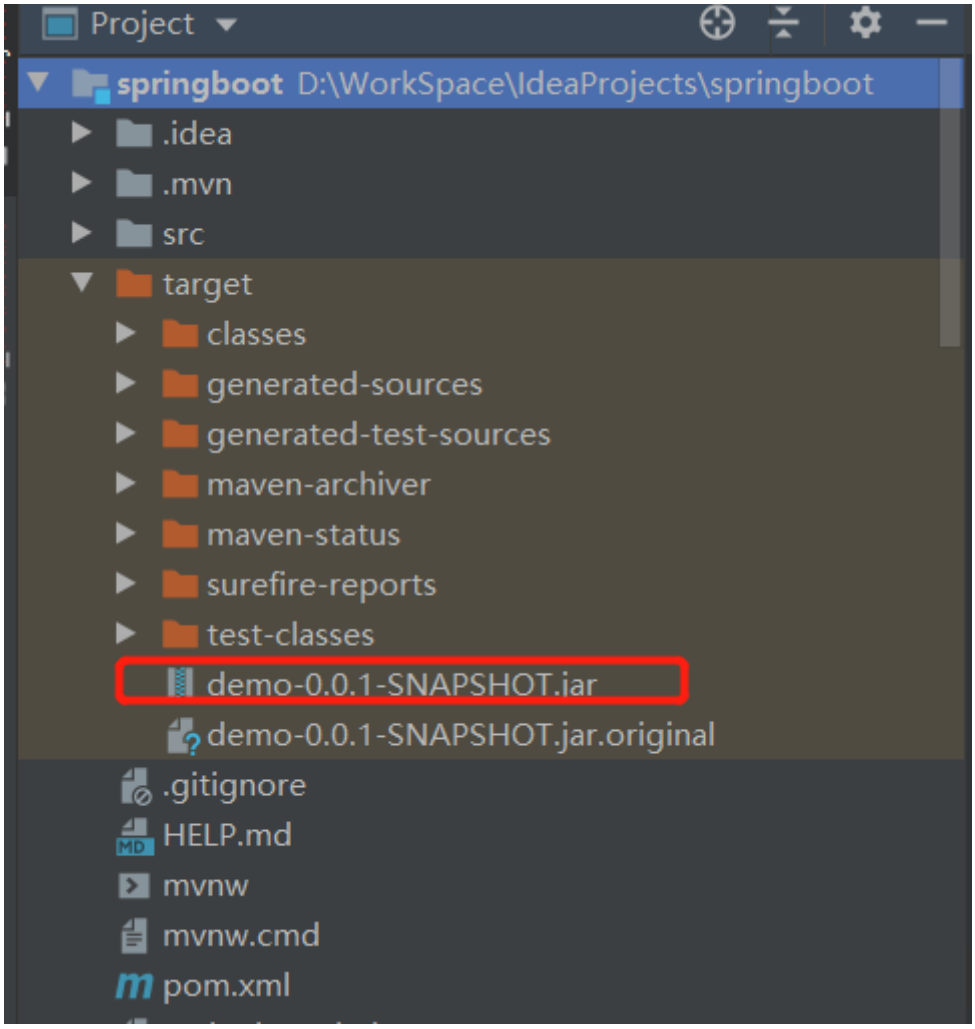
1、本地新建 Spring Boot 应用，打包，先执行 clean。



2、执行 package 进行打包。



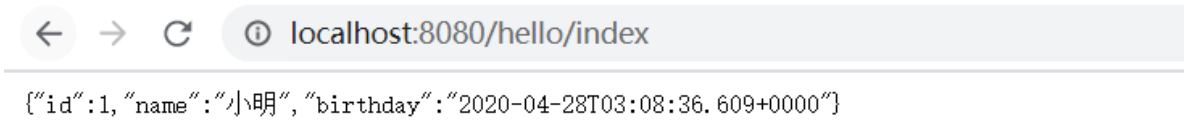
打包成功，target 目录下可以看到 jar 文件。



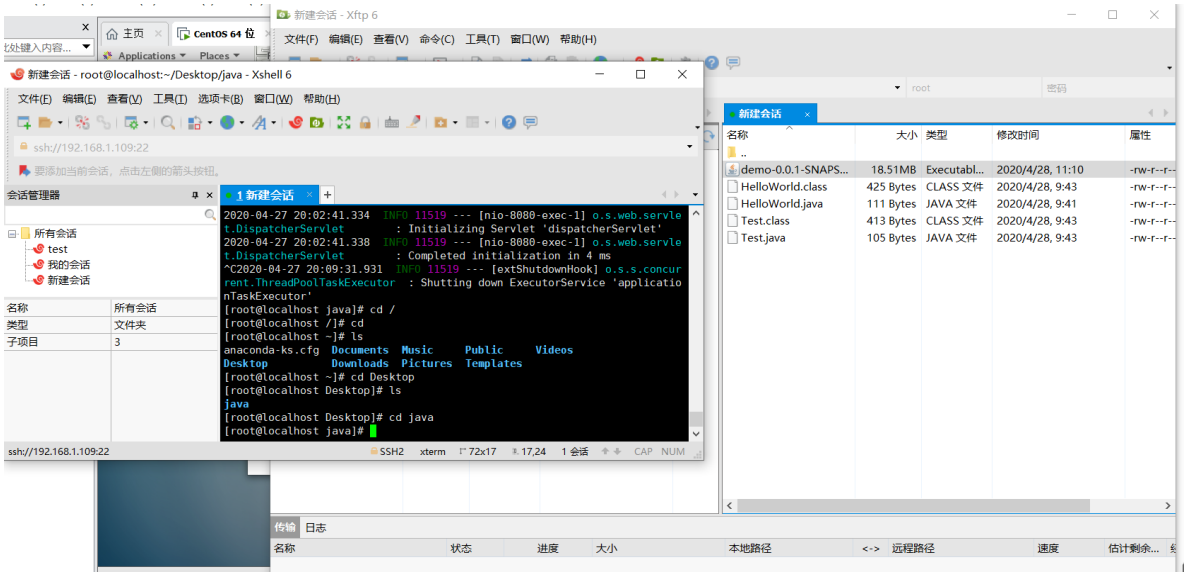
3、先测试本地发布，CMD 进入 jar 目录，执行

```
java -jar demo-0.0.1-SNAPSHOT.jar
```

4、启动成功，浏览器访问 localhost:8080/hello/index

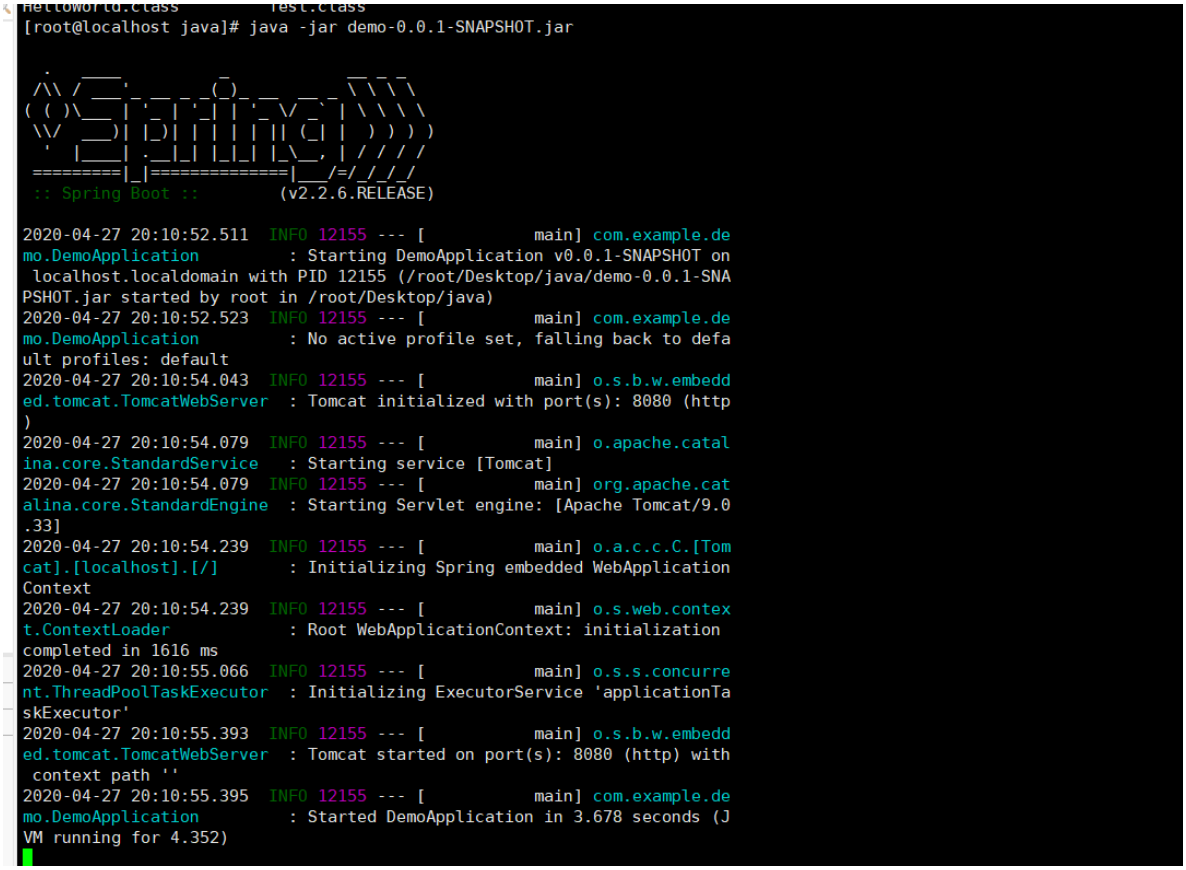


5、通过 Xftp 将 jar 包拷贝到 CentOS。

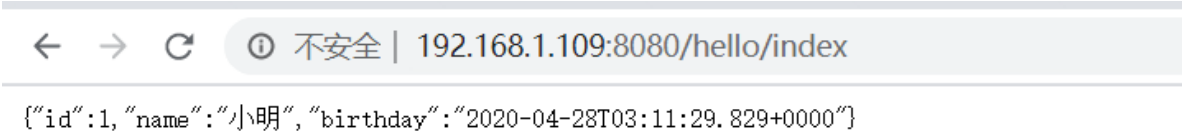


6、命令行启动 Spring Boot 应用。

```
java -jar demo-0.0.1-SNAPSHOT.jar
```

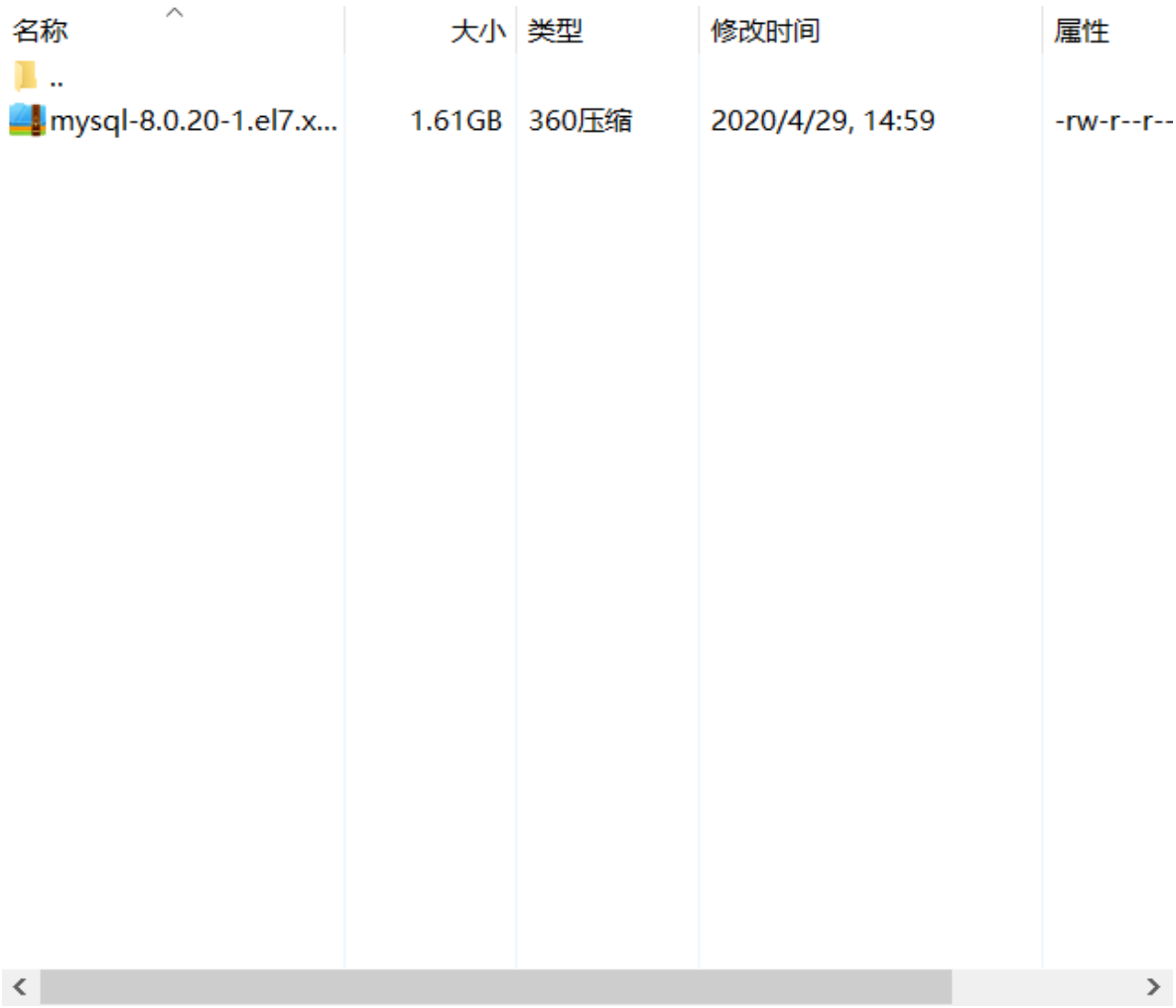


7、打开浏览器访问，注意，如果 Spring Boot 端口不是 8080，需要提前在 CentOS 中开放对应端口。



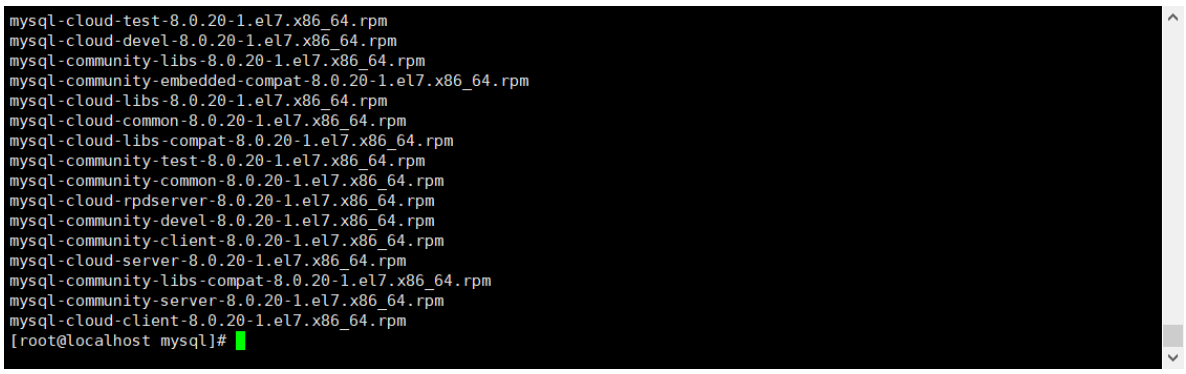
8 安装 MySQL

1、新建 /usr/local/mysql，将 MySQL 安装包拷贝过去。



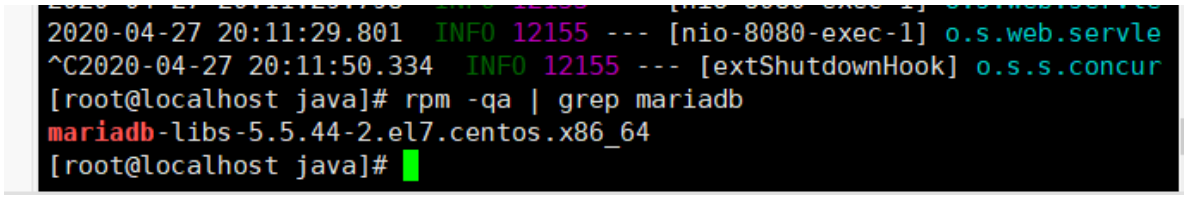
2、进入 /usr/local/mysql，解压。

```
tar -xvf mysql-8.0.20-1.el7.x86_64.rpm-bundle.tar
```



3、安装 MySQL 之前需要先删除 mariadb，检查是否存在 mariadb

```
rpm -qa | grep mariadb
```



4、卸载 mariadb

```
rpm -e mariadb-libs-5.5.44-2.el7.centos.x86_64 --nodeps
```

```
[root@localhost java]# rpm -qa | grep mariadb
mariadb-libs-5.5.44-2.el7.centos.x86_64
[root@localhost java]# rpm -e mariadb-libs-5.5.44-2.el7.centos.x86_64 --
nodeps
[root@localhost java]#
```

5、安装 common

```
mysql-community-common-8.0.20-1.el7.x86_64.rpm
```

```
rpm -ivh mysql-community-common-8.0.20-1.el7.x86_64.rpm --
nodeps --force
```

```
[root@localhost mysql]# rpm -ivh mysql-community-common-8.0.20-1.el7.x86_64.rpm --nodeps --force
warning: mysql-community-common-8.0.20-1.el7.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:mysql-community-common-8.0.20-1.e################################# [100%]
[root@localhost mysql]#
```

6、安装 libs

```
mysql-community-libs-8.0.20-1.el7.x86_64.rpm
```

```
rpm -ivh mysql-community-libs-8.0.20-1.el7.x86_64.rpm --
nodeps --force
```

```
   1:mysql-community-common-8.0.20-1.e################################# [100%]
[root@localhost mysql]# rpm -ivh mysql-community-libs-8.0.20-1.el7.x86_64.rpm --nodeps --force
warning: mysql-community-libs-8.0.20-1.el7.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:mysql-community-libs-8.0.20-1.el7################################# [100%]
[root@localhost mysql]#
```

7、安装 client

```
mysql-community-client-8.0.20-1.el7.x86_64.rpm
```

```
rpm -ivh mysql-community-client-8.0.20-1.el7.x86_64.rpm --
nodeps --force
```

```
   1:mysql-community-libs-8.0.20-1.el7################################# [100%]
[root@localhost mysql]# rpm -ivh mysql-community-client-8.0.20-1.el7.x86_64.rpm --nodeps --force
warning: mysql-community-client-8.0.20-1.el7.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:mysql-community-client-8.0.20-1.e################################# [100%]
[root@localhost mysql]#
```

8、安装 server

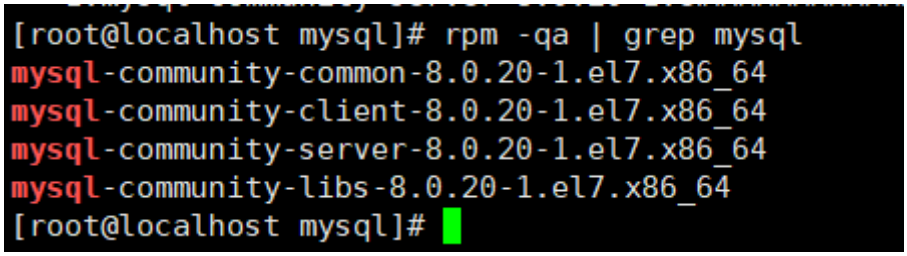
```
mysql-community-server-8.0.20-1.el7.x86_64.rpm
```

```
rpm -ivh mysql-community-server-8.0.20-1.el7.x86_64.rpm --
nodeps --force
```

```
   1:mysql-community-client-8.0.20-1.e################################# [100%]
[root@localhost mysql]# rpm -ivh mysql-community-server-8.0.20-1.el7.x86_64.rpm --nodeps --force
warning: mysql-community-server-8.0.20-1.el7.x86_64.rpm: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Preparing...                          ################################# [100%]
Updating / installing...
   1:mysql-community-server-8.0.20-1.e################################# [100%]
[root@localhost mysql]#
```

9、检查安装结果

```
rpm -qa | grep mysql
```




```
rpm -ivh mysql-community-common-8.0.20-1.el7.x86_64.rpm --
nodeps --force
rpm -ivh mysql-community-libs-8.0.20-1.el7.x86_64.rpm --
nodeps --force
rpm -ivh mysql-community-client-8.0.20-1.el7.x86_64.rpm --
nodeps --force
rpm -ivh mysql-community-server-8.0.20-1.el7.x86_64.rpm --
nodeps --force
```

10、初始化 MySQL

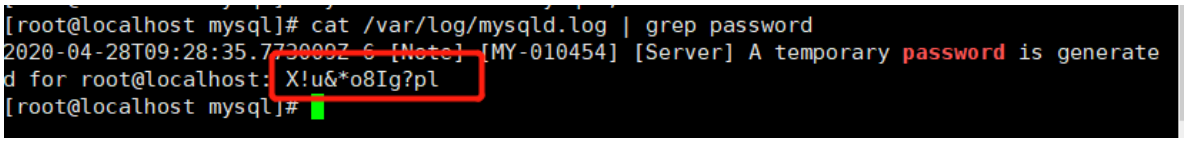
```
mysqld --initialize
```

11、授权防火墙

```
chown mysql:mysql /var/lib/mysql -R;
systemctl start mysqld.service;
systemctl enable mysqld;
```

12、查看数据库默认密码

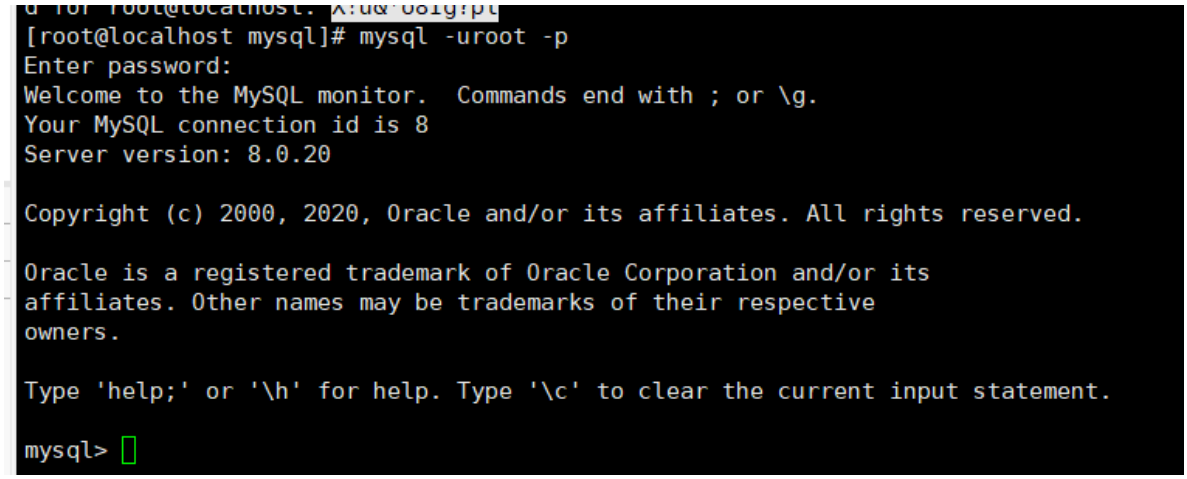
```
cat /var/log/mysqld.log | grep password
```



13、登录数据库

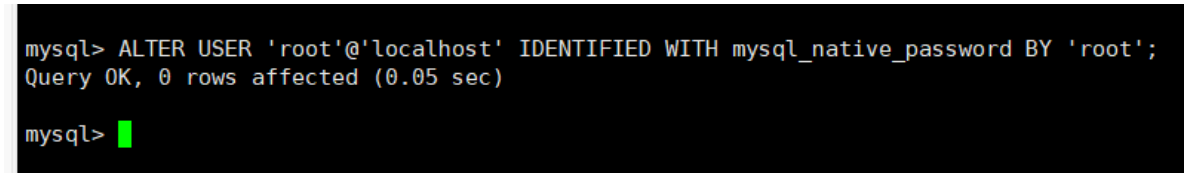
```
mysql -uroot -p
```

14、复制粘贴密码



15、登录成功，修改密码

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH
mysql_native_password BY 'root';
```



16、exit 退出，使用新密码重新登陆。

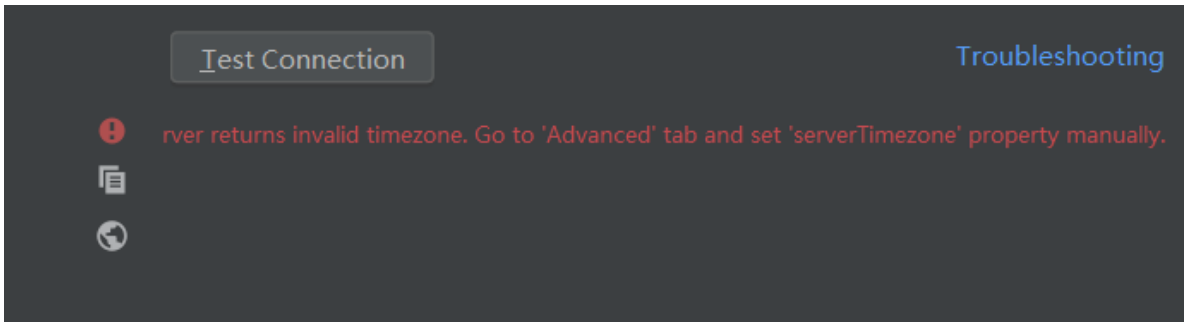
17、开启远程访问。

```
create user 'root'@'%' identified with
mysql_native_password by 'root';
grant all privileges on *.* to 'root'@'%' with grant
option;
flush privileges;
```

18、CentOS 开放 3306 端口。

```
firewall-cmd --zone=public --add-port=3306/tcp --permanent
systemctl restart firewalld.service
firewall-cmd --reload
```

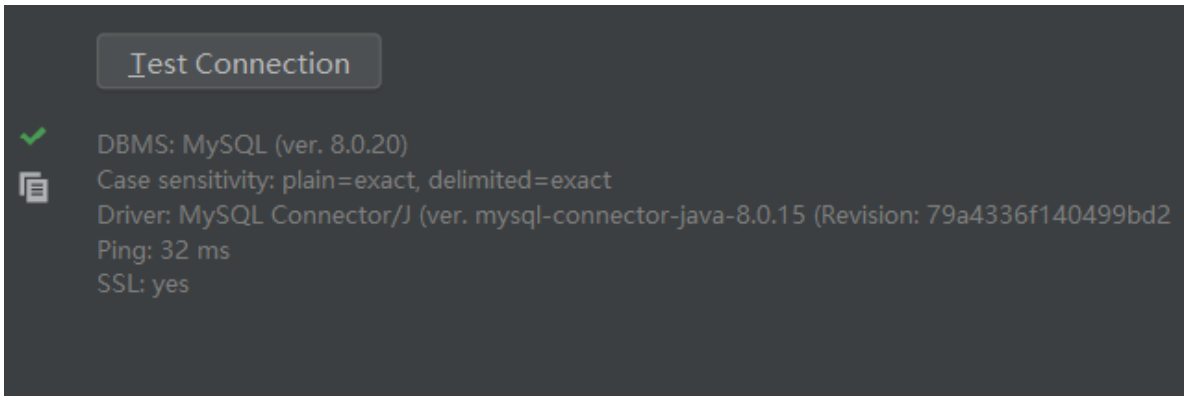
19、使用 DataGrip 连接，如果报如下错误，数据库时区问题，因为 MySQL 安装默认设置为美国时区，而北京时间比美国晚 8 小时。



在数据库中修改时区即可。

```
set global time_zone='+8:00';
```

再次连接，成功。



9 安装 Redis

1、进入 /usr/local

2、下载 Redis

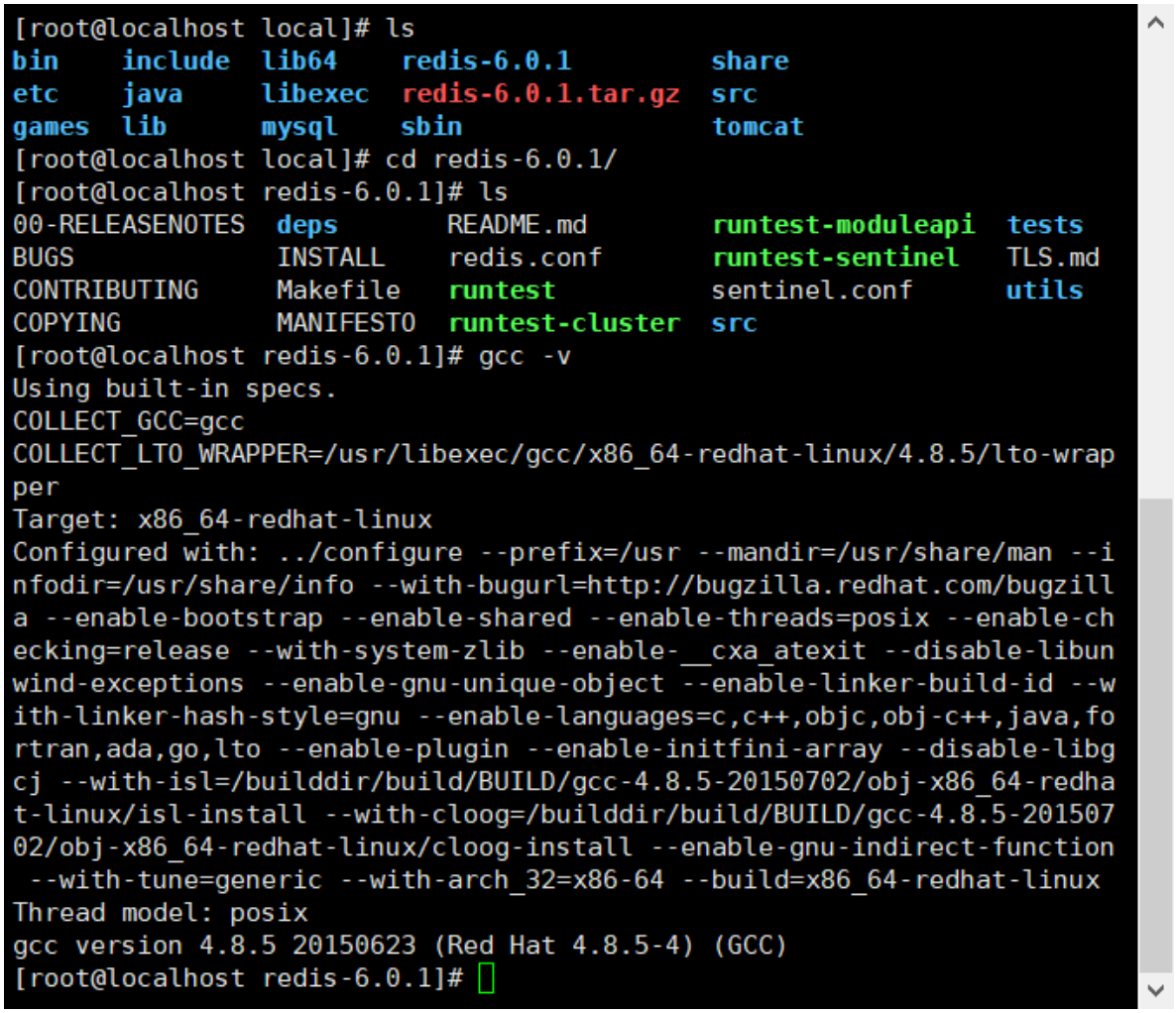
```
wget http://download.redis.io/releases/redis-5.0.5.tar.gz
```

3、解压缩

```
tar -xvf redis-5.0.5.tar.gz
```

4、安装 gcc，检查 CentOS 是否已经安装

```
gcc -v
```



已经安装，并且版本是 4.8.5，如果没有安装，需要手动进行安装。

```
yum -y install gcc
```

如果 yum 安装报错 yum 被 packagekit 占用

将 /etc/yum/pluginconf.d/refresh-packagekit.conf 改为如下

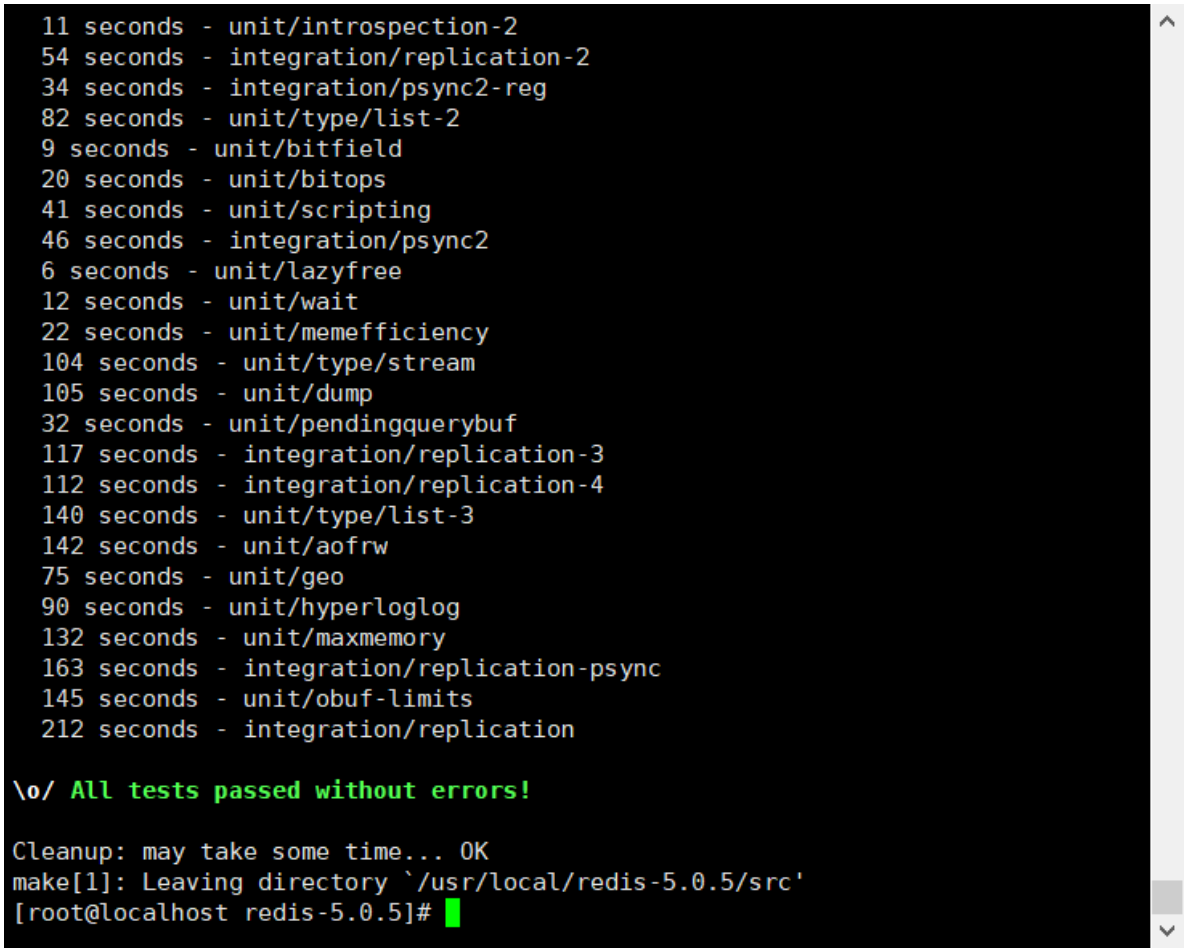
enabled=0

然后重启 Linux 即可

5、安装 Redis

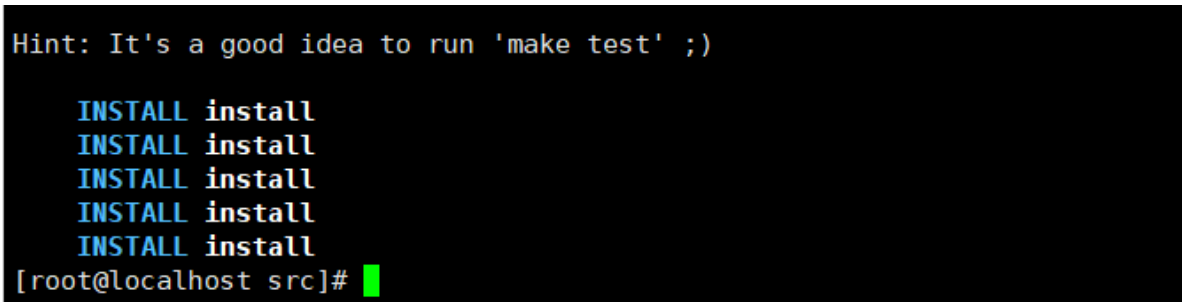
```
cd redis-5.0.5
#安装
make MALLOC=libc
#测试
make test
```

6、测试通过 需要一分钟左右



7、正式安装

```
cd src && make install
```



8、启动 Redis

```
./redis-server
```

```
0, modified=0, pid=9367, just started
9367:C 05 May 2020 23:55:02.683 # Warning: no config file specified, using the
 default config. In order to specify a config file use ./redis-server /path/to
/redis.conf
9367:M 05 May 2020 23:55:02.684 * Increased maximum number of open files to 10
032 (it was originally set to 1024).
                _.-``__ ''-._
           _.-``    `.  `_.  ''-._           Redis 5.0.5 (00000000/0) 64 bit
       .-`` .-```.  ```\/    _.,_ ''-._
      (    '      ,       .-`  | `,    )     Running in standalone mode
      |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
      |    `-._   `._    /     _.-'    |     PID: 9367
       `-._    `-._  `-./  _.-'    _.-'
      |`-._`-._    `-.__.-'    _.-'_.-'|
      |    `-._`-._        _.-'_.-'    |           http://redis.io
       `-._    `-._`-.__.-'_.-'    _.-'
      |`-._`-._    `-.__.-'    _.-'_.-'|
      |    `-._`-._        _.-'_.-'    |
       `-._    `-._`-.__.-'_.-'    _.-'
           `-._    `-.__.-'    _.-'
               `-._        _.-'
                   `-.__.-'

9367:M 05 May 2020 23:55:02.685 # WARNING: The TCP backlog setting of 511 cann
ot be enforced because /proc/sys/net/core/somaxconn is set to the lower value
of 128.
9367:M 05 May 2020 23:55:02.685 # Server initialized
9367:M 05 May 2020 23:55:02.685 # WARNING overcommit_memory is set to 0! Backg
round save may fail under low memory condition. To fix this issue add 'vm.over
```

9、配置为后台服务

修改 redis.conf 文件

```
cd ..
vim redis.conf
```

守护线程改为 yes 表示启动后台启动，保存退出

```
################################ 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 da
daemonize yes

# If you run Redis from upstart or systemd, Redis can interact with y
# supervision tree. Options:
#   supervised no      - no supervision interaction
```

10、将 redis.conf 复制到 /etc/redis 路径下，并改名为 6379.conf

```
cd /etc
mkdir redis
cd redis
cp /usr/local/redis-5.0.5/redis.conf 6379.conf
```

11、将启动文件 usr/local/redis-5.0.5/utils/redis_init_script 拷贝到 /etc/rc.d/init.d/

```
cp /usr/local/redis-5.0.5/utils/redis_init_script
/etc/rc.d/init.d/
```

改名字

```
mv redis_init_script redisd
```

12、修改复制后的 redisd 文件，让它成为服务

```
cd /etc/rc.d/init.d/
vim redisd
```

- 修改 EXEC、CLIEXEC 的路径

```
#原内容
EXEC=/usr/local/bin/redis-server
CLIEXEC=/usr/local/bin/redis-cli

#修改后的内容
EXEC=/usr/local/redis-5.0.5/src/redis-server
CLIEXEC=/usr/local/redis-5.0.5/src/redis-cli
```

- 在 $EXEC $CONF 后面加上 &，表示后台启动

```
#!/bin/sh
#
# Simple Redis init.d script conceived to work on Linux systems
# as it does use of the /proc filesystem.

### BEGIN INIT INFO
# Provides:     redis_6379
# Default-Start:        2 3 4 5
# Default-Stop:         0 1 6
# Short-Description:    Redis data structure server
# Description:          Redis data structure server. See https://redis.io
### END INIT INFO

REDISPORT=6379
EXEC=/usr/local/redis-5.0.5/src/redis-server
CLIEXEC=/usr/local/redis-5.0.5/src/redis-cli

PIDFILE=/var/run/redis_${REDISPORT}.pid
CONF="/etc/redis/${REDISPORT}.conf"

case "$1" in
    start)
        if [ -f $PIDFILE ]
        then
                echo "$PIDFILE exists, process is already running or crashed"
        else
                echo "Starting Redis server..."
                $EXEC $CONF &
        fi
-- INSERT --                                              28,30         Top
```

13、添加开机启动

```
chkconfig redisd on
```

14、启动 redis 服务

```
service redisd start
```

```
[root@localhost init.d]# vim redisd
[root@localhost init.d]# chkconfig redisd on
[root@localhost init.d]# service redisd start
Starting Redis server...
[root@localhost init.d]# 9914:C 06 May 2020 00:38:40.425 # oOOOoOOOoOOOo Redis
 is starting oOOOoOOOoOOOo
9914:C 06 May 2020 00:38:40.425 # Redis version=5.0.5, bits=64, commit=0000000
0, modified=0, pid=9914, just started
9914:C 06 May 2020 00:38:40.425 # Configuration loaded
```

Ctrl + c 退出，Reids 也不会关闭了，执行命令查看

```
ps -ef | grep redis
```

```
^C
[root@localhost init.d]# ps -ef | grep redis
root       3789   3626  0 May05 pts/1    00:00:00 wget http://download.redis.i
o/releases/redis-5.0.5.tar.gz
root       9915      1  0 00:38 ?        00:00:00 /usr/local/redis-5.0.5/src/r
edis-server 127.0.0.1:6379
root       9929   3626  0 00:39 pts/1    00:00:00 grep --color=auto redis
[root@localhost init.d]#
```

可以看到 Redis 服务已经后台启动了

15、关闭 Redis 服务

```
service redisd stop
```

16、客户端访问

```
cd /usr/local/redis-5.0.5/src
redis-cli
```

```
evict.c       networking.o       sds.h              zmalloc.o
[root@localhost src]# redis-cli
127.0.0.1:6379> set name tom
OK
127.0.0.1:6379> get name
"tom"
127.0.0.1:6379>
```

17、允许外部访问

- 添加端口

1、查看防火墙状态

```
firewall-cmd --state
```

```
[root@localhost bin]# firewall-cmd --state
running
[root@localhost bin]#
```

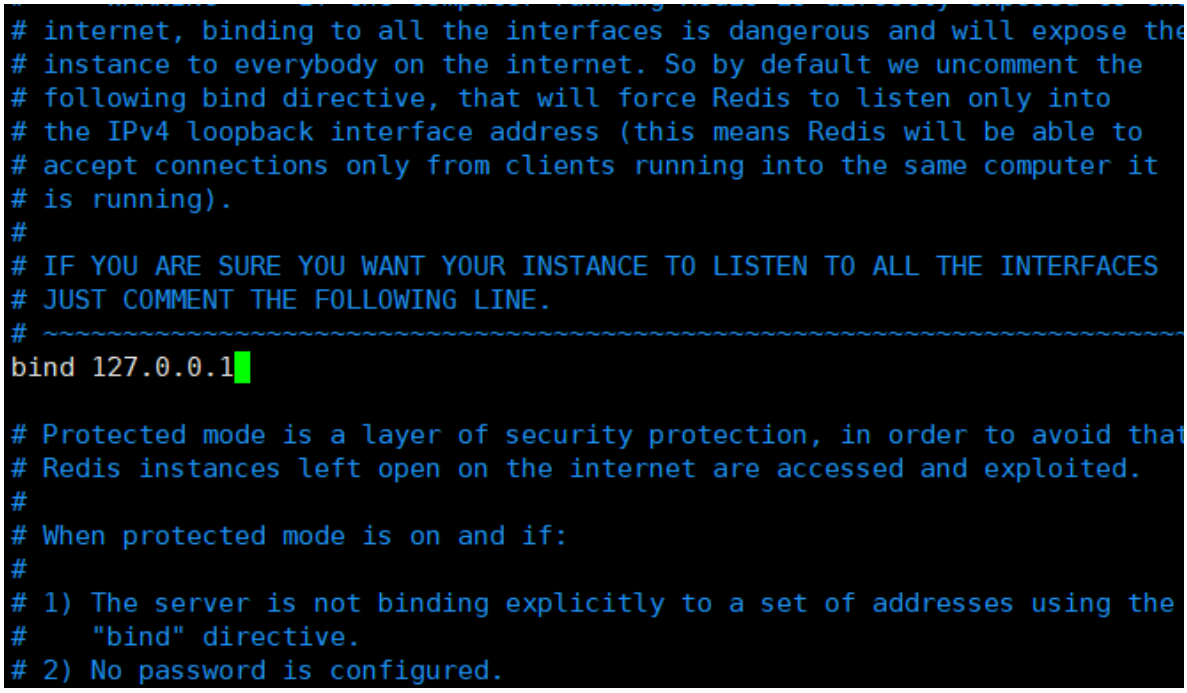
runing 表示开启，not runing 表示关闭，如果关闭，执行

```
systemctl start firewalld.service
```

```
firewall-cmd --zone=public --add-port=6379/tcp --permanent
systemctl restart firewalld.service
firewall-cmd --reload
```

- 修改配置文件

```
vim /etc/redis/6379.conf
```



bind 改为 0.0.0.0 表示任何 IP 都可以连接