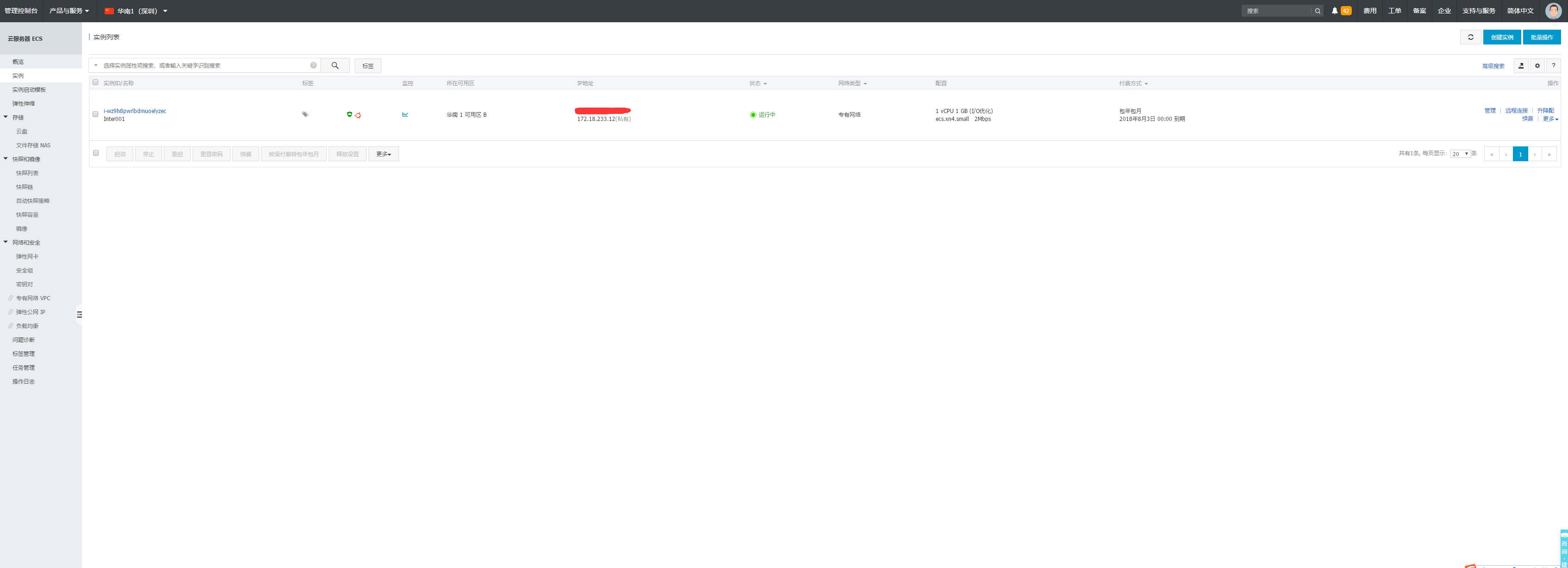
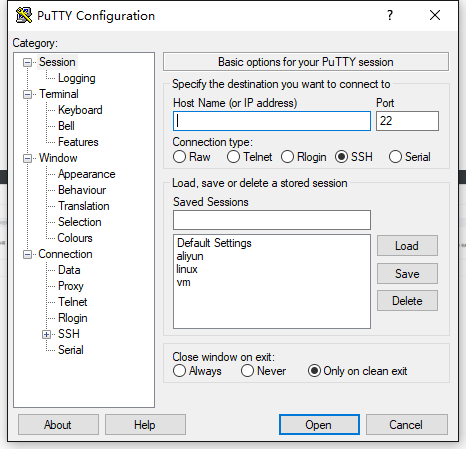
Linux学习笔记

# 安装与连接

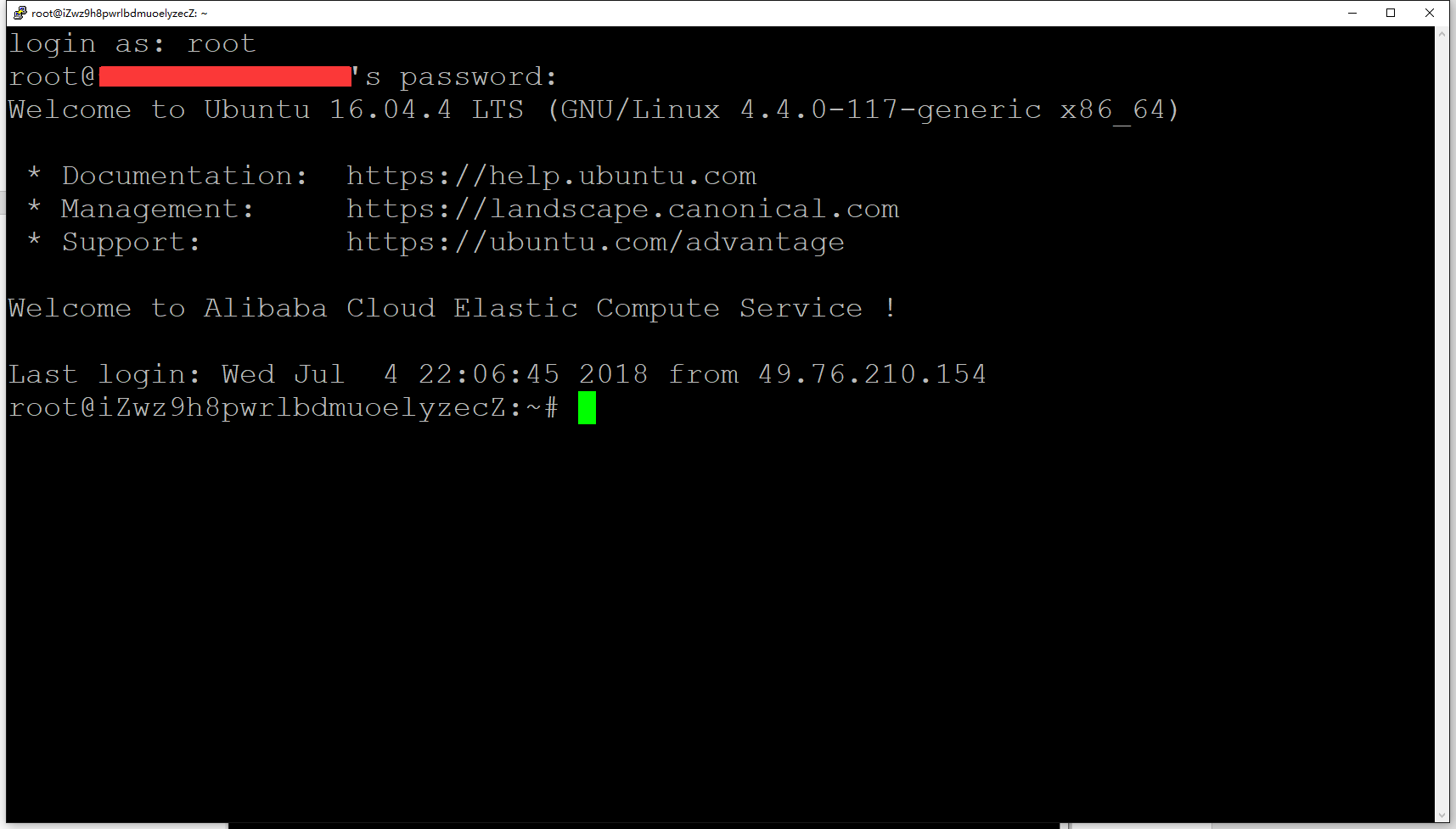
1. 购买云服务器



1. 安装putty



1. 连接服务器

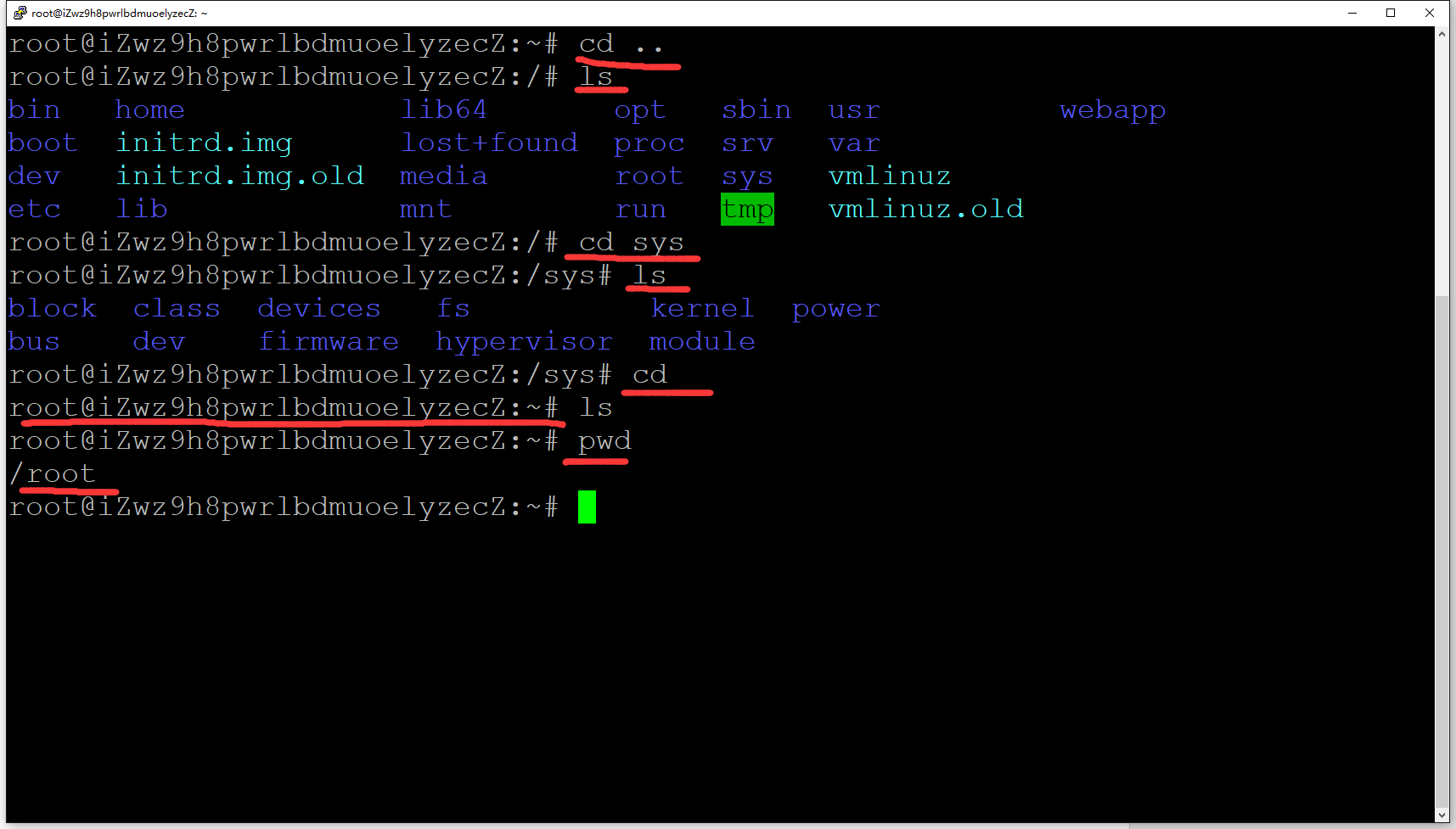


# 基础命令

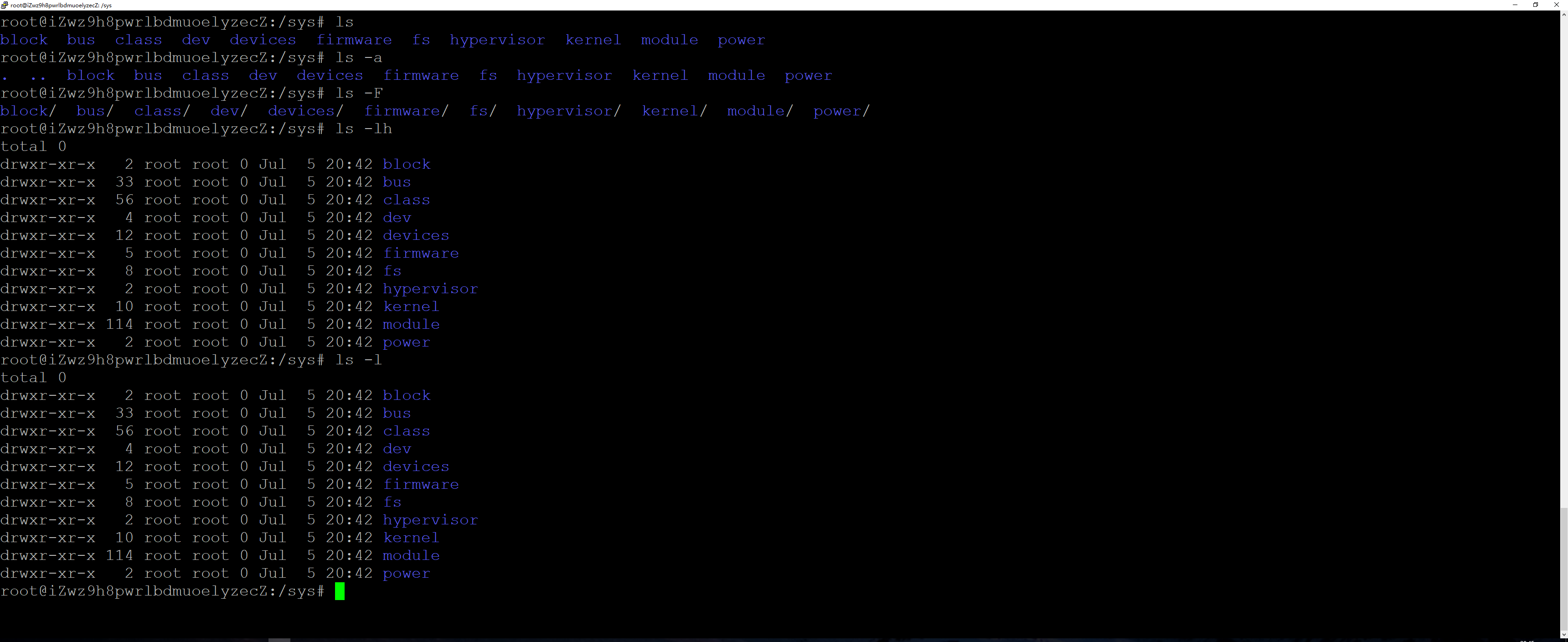
1. 文件基本操作

pwd 打印当前⼯作⽬录 print working directory  
cd 切换⽬录 change directory

cd ~ 宿主⽬录 或直接cd  
cd - 上次的⽬录  
cd .. 上级⽬录 (.代表当前⽬录)



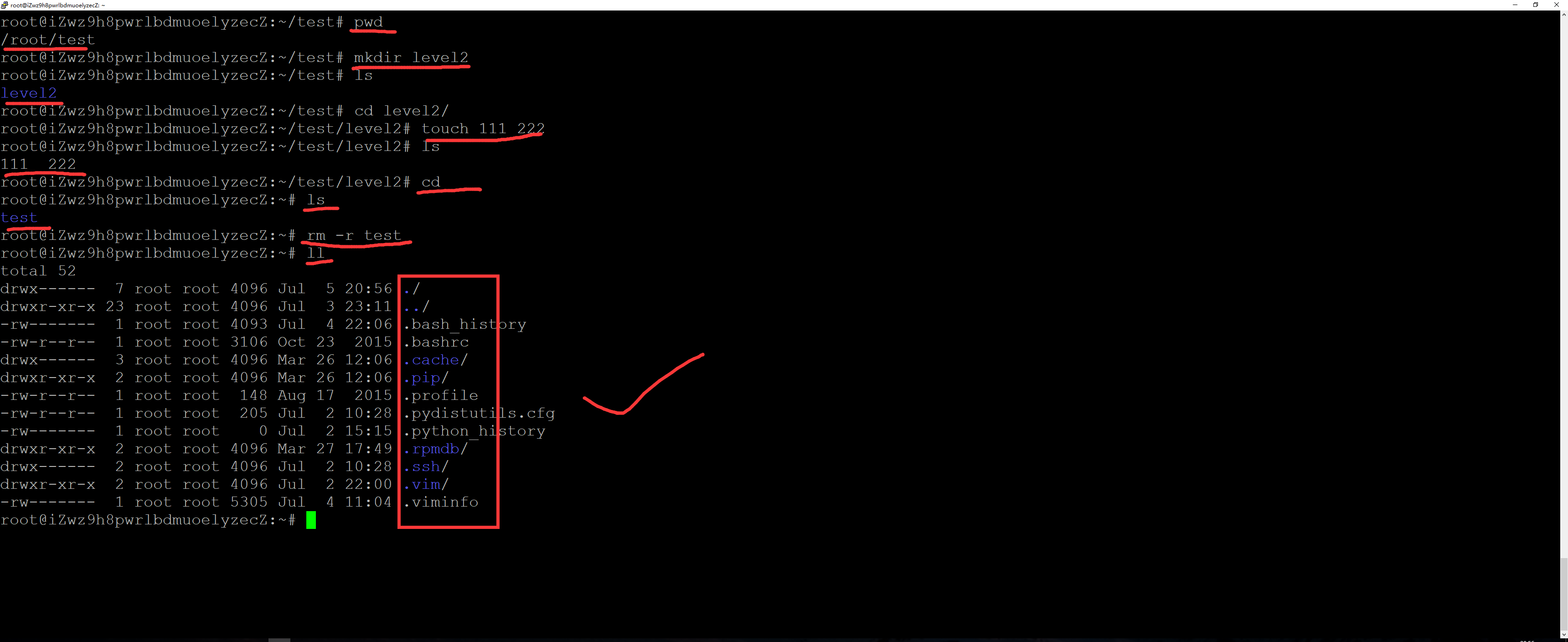
ls -l 详细信息  
ls -a 显示所有⽂件(包含以点开头的⽂件)  
ls -F 在⽬录后加斜线显示  
ls -lh ⼈性化单位显示⼤⼩



mkdir 创建⽬录 make directory  
touch 创建⼀个空⽂件  
rm 删除 remove

rm -r 删除⽬录  
rm -i 删除前提示 (CentOS默认)  
rm -f 删除⽂件不提示  
rm -rf 删除⽬录(不提示) (rm -r -f)  
rm -rf ./\* 删除当前⽬录所有⽂件



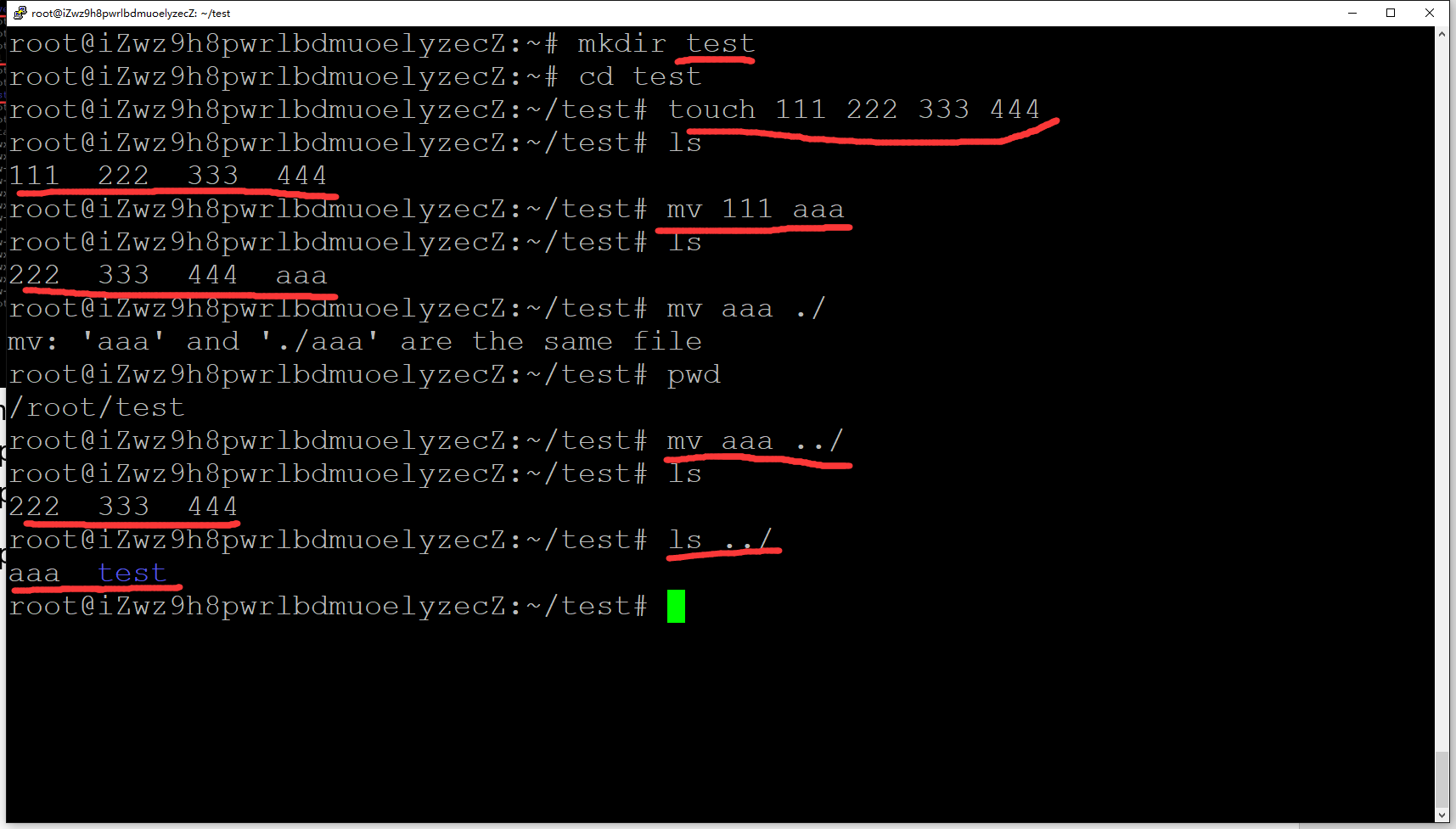


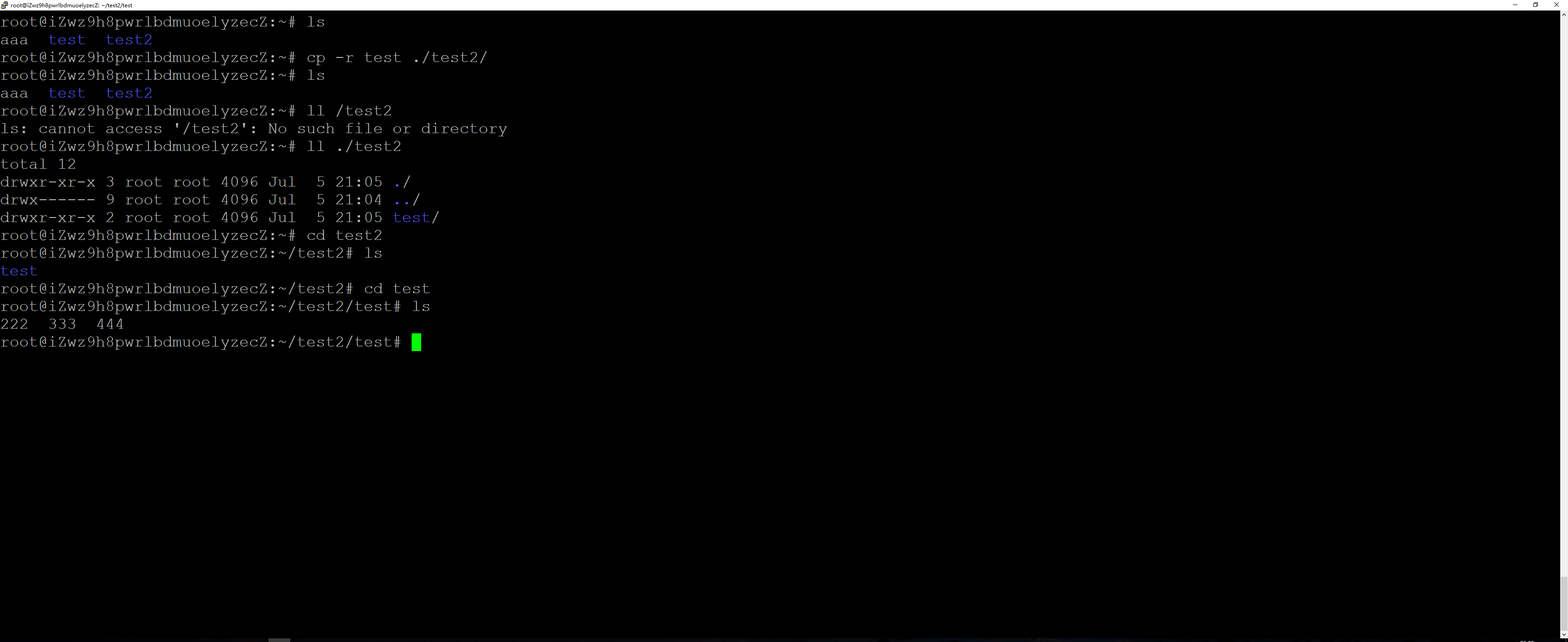
mv 移动 move (重命名也是这个命令)

cp 复制 copy

cp /etc/issue ./

cp -r 递归复制⽬录





find 查找 find 路径 -name 文件名（模糊查询,\*?支持正则？）

-size 根据尺寸查找

which 查找文件路径 which python3

cat 查看文件内容，适合于小文件

more 查看文件内容，大文件 回车==》一行 空格==》一页 q退出

head 看文件头几行 head -n 5 /etc/services 默认10行

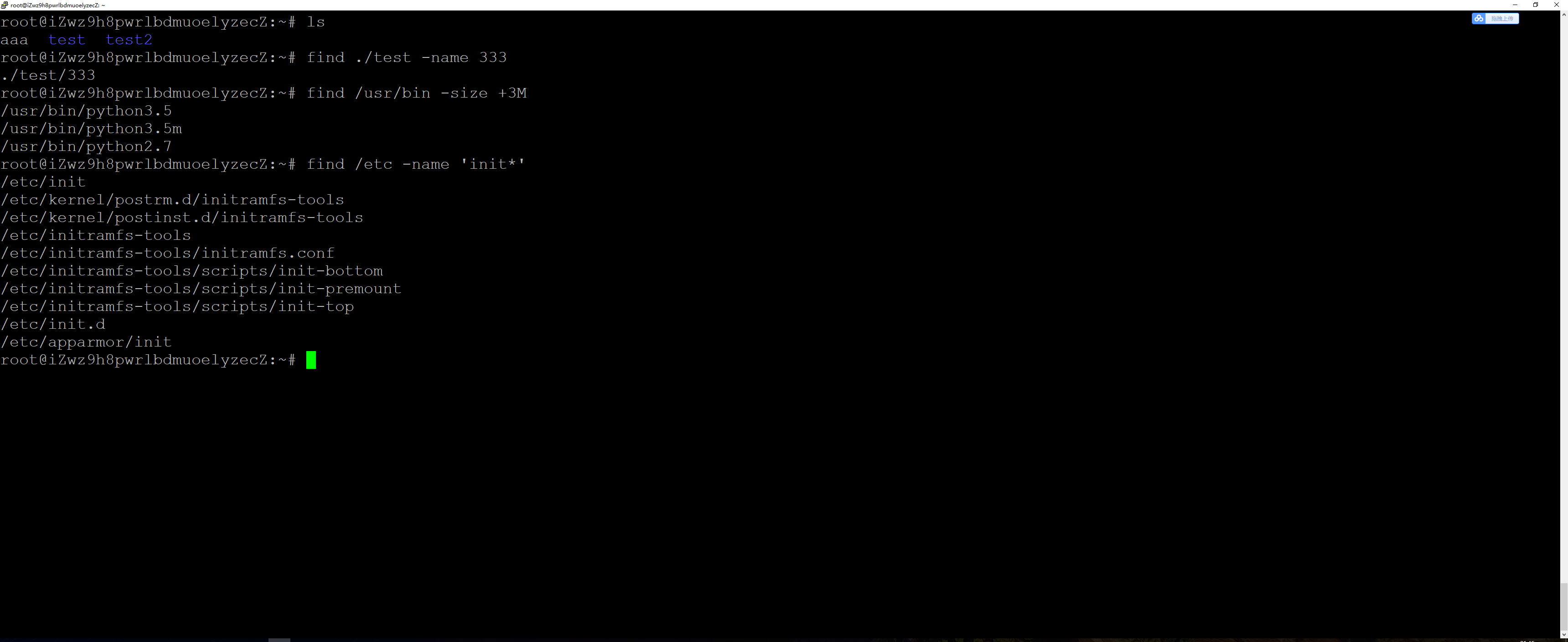
tail 看文件最后几行 tail -n 5 /etc/services -f 查看文件变化

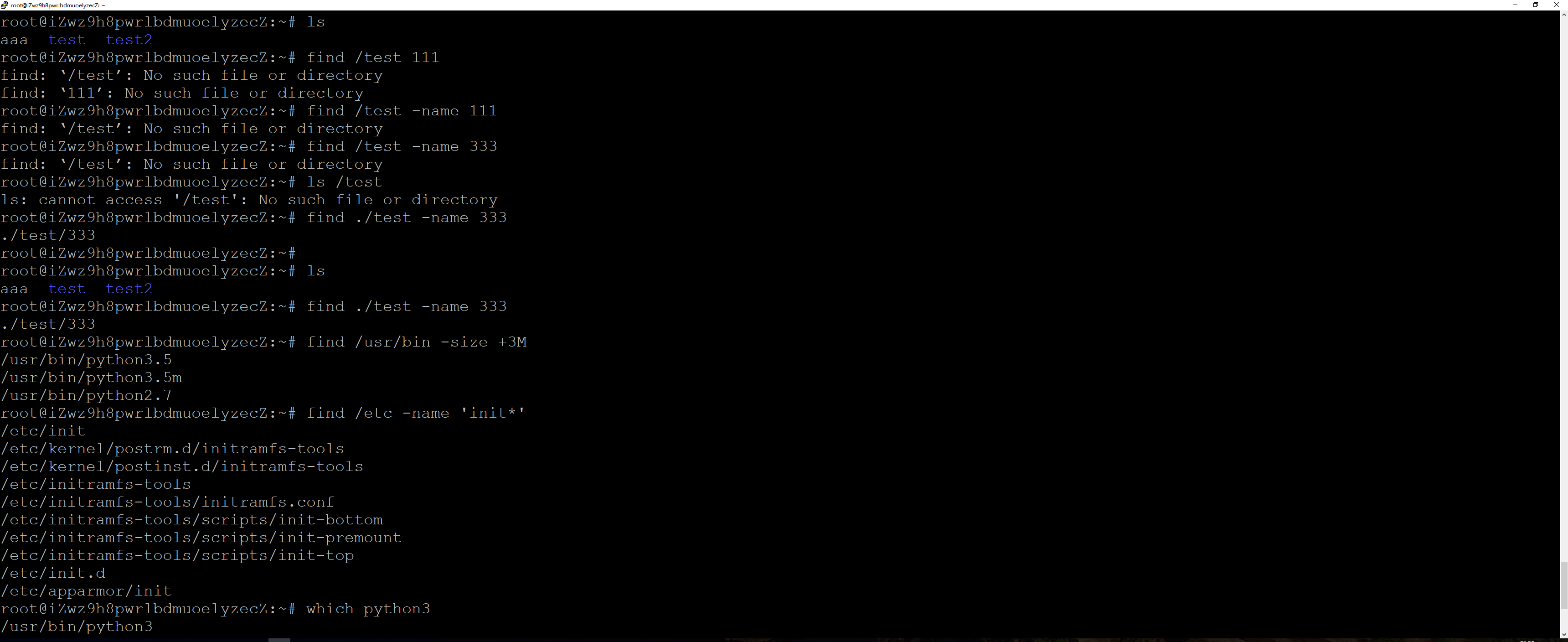
ln 创建链接 -s创建软链接 ln -s readme.txt hello.txt

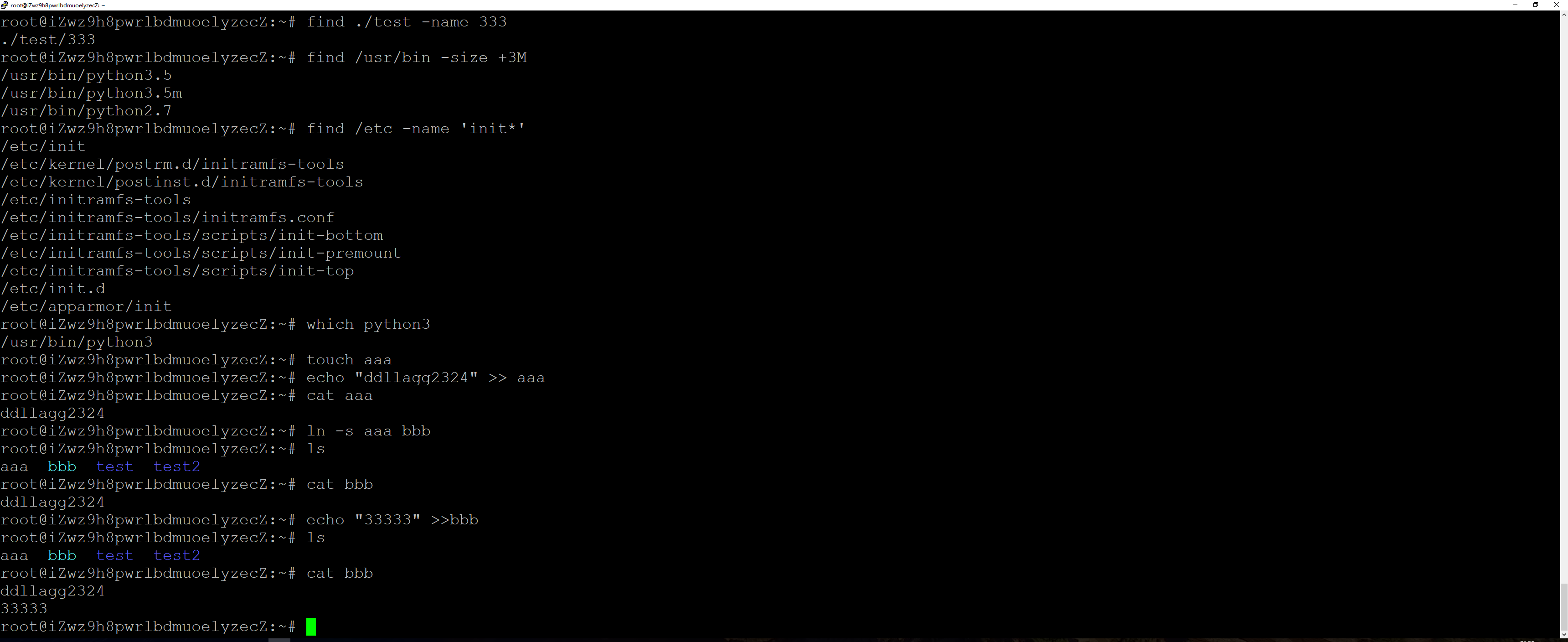
不加参数则为硬链接,硬链接相当于创建副本，大小内容都会同时变化

ln -s 源⽂件 新⽂件 (创建软链接)  
ln 源⽂件 新⽂件 （创建硬链接）

echo "字符串">> readme.txt 输出重定向







# 解包、打包、安装、源码编译安装

tar -zcf 文件 -z gzip -c create -f 文件 -v显示内容 压缩

tar -zxf 文件 -z gzip -x xtarct -f 文件 -v显示内容 解压缩

bz2格式 将z换成j

zip apt install zip

zip xx.zip 文件 unzip xx.zip

实战：Tomcat

1.下载JAVA

<http://download.oracle.com/otn-pub/java/jdk/8u172-b11/a58eab1ec242421181065cdc37240b08/jdk-8u172-linux-x64.tar.gz>

注意：jdk包先下载至Windows再传至Linux，直接wget解压会报错

* 1. 在usr/lib目录下创建jdk1.8目录，并解压缩
  2. 编辑配置文件，创建环境变量：

vim /etc/profile 最前面添加

export JAVA\_HOME=/usr/lib/jdk1.8

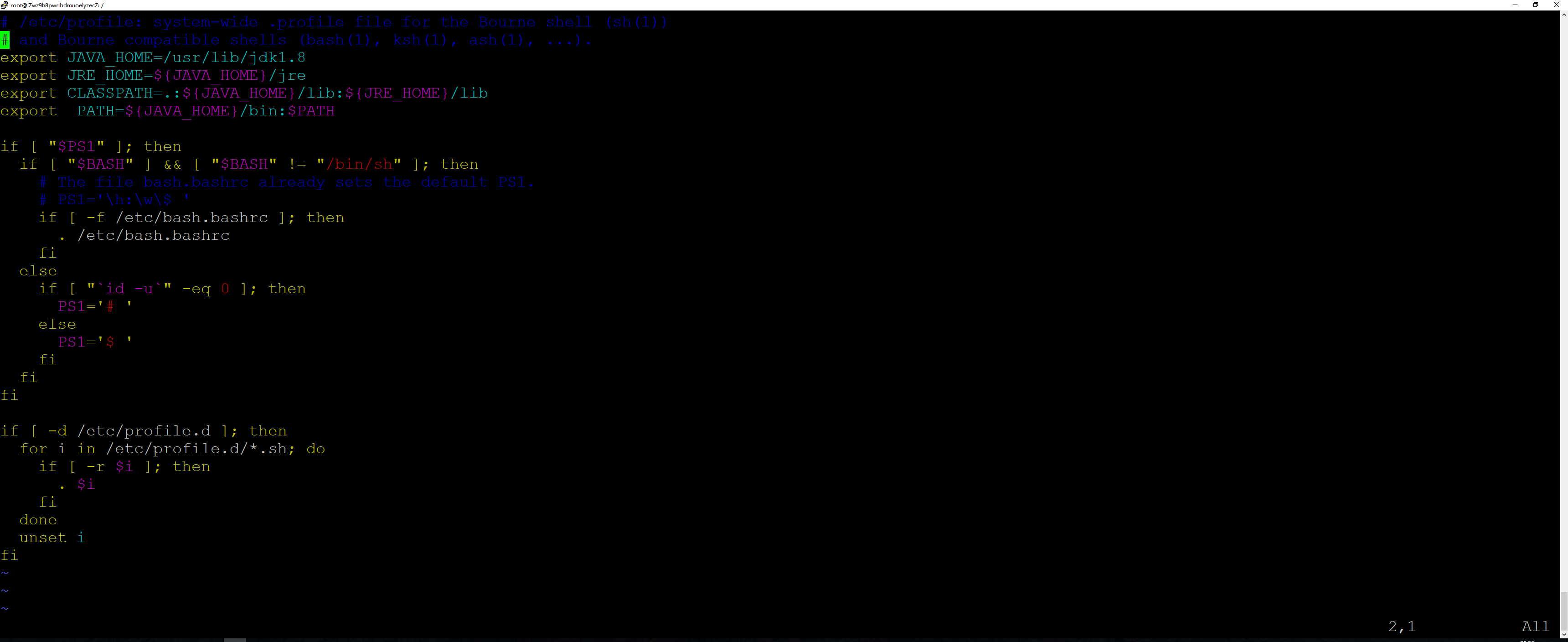
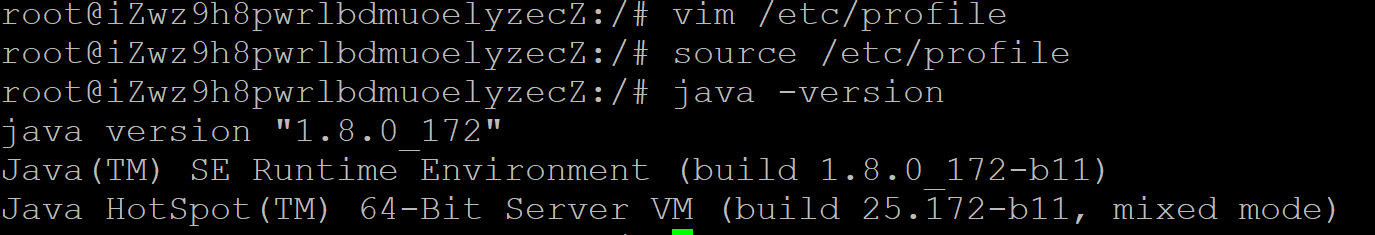
export JRE\_HOME=${JAVA\_HOME}/jre

export CLASSPATH=.:${JAVA\_HOME}/lib:${JRE\_HOME}/lib

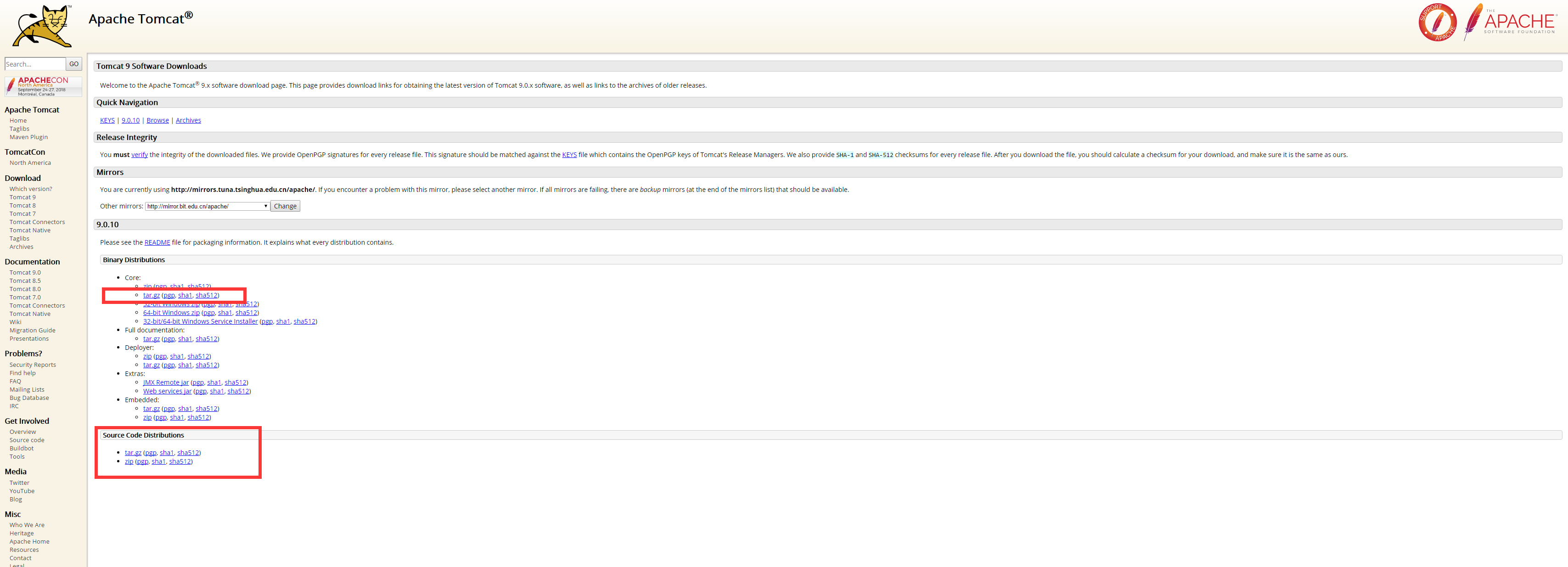
export PATH=${JAVA\_HOME}/bin:$PATH

* 1. source /etc/profile

java -version

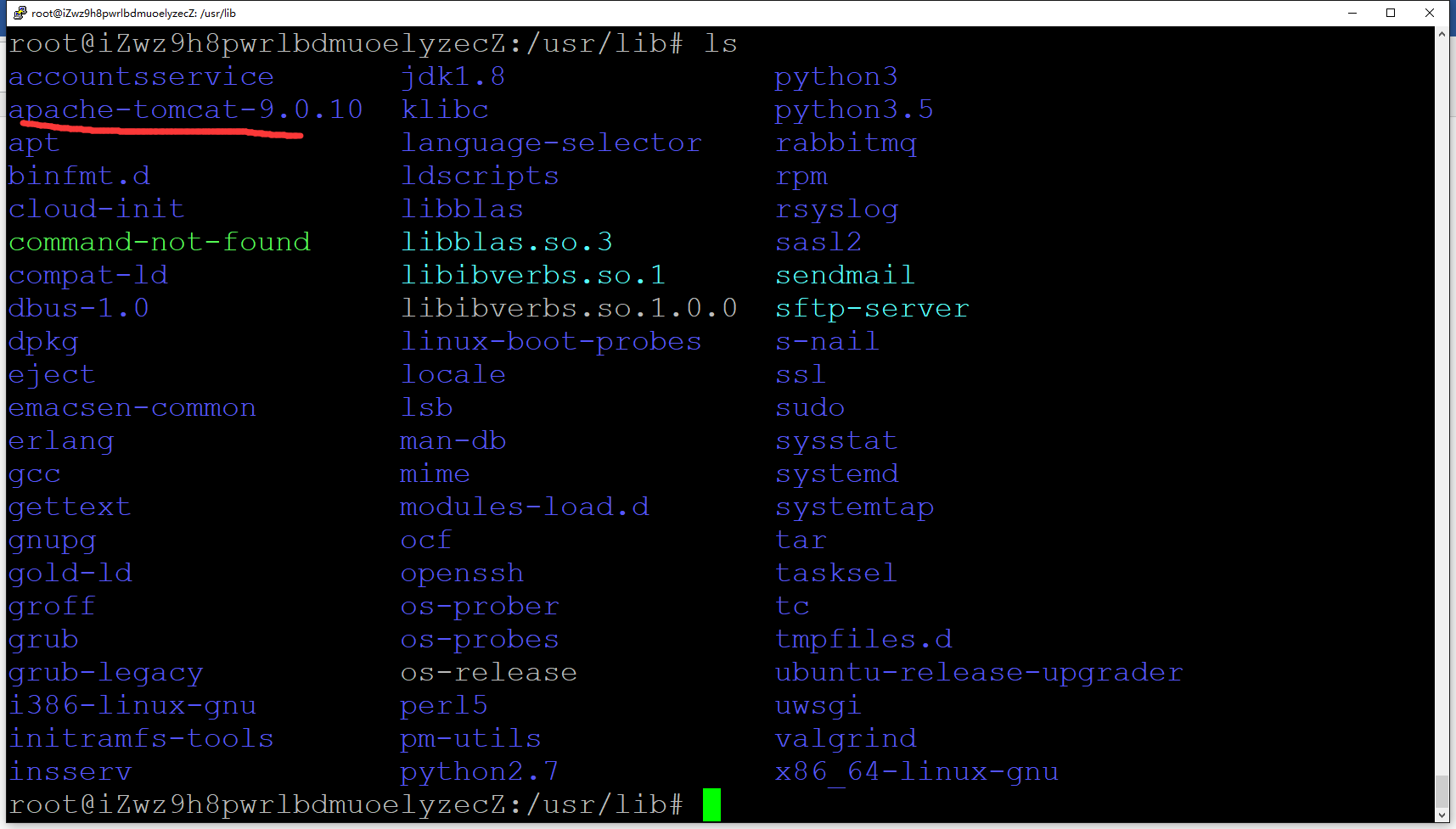
2.下载TOMCAT



安装：<http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-9/v9.0.10/bin/apache-tomcat-9.0.10.tar.gz>

源码：<http://mirrors.tuna.tsinghua.edu.cn/apache/tomcat/tomcat-9/v9.0.10/src/apache-tomcat-9.0.10-src.tar.gz>

下载安装包，并解压到、usr/lib下



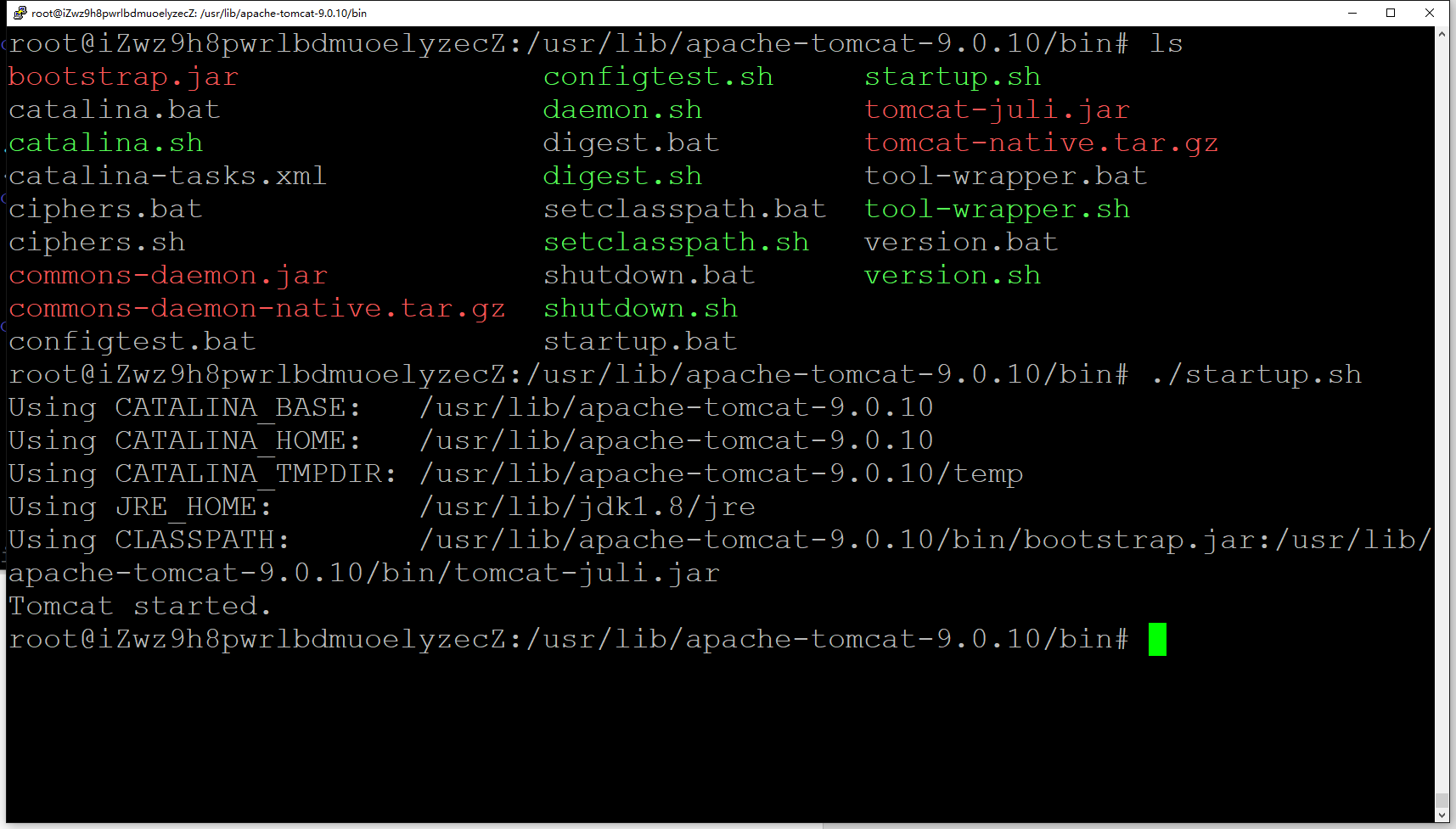
进入目录后./startup.sh

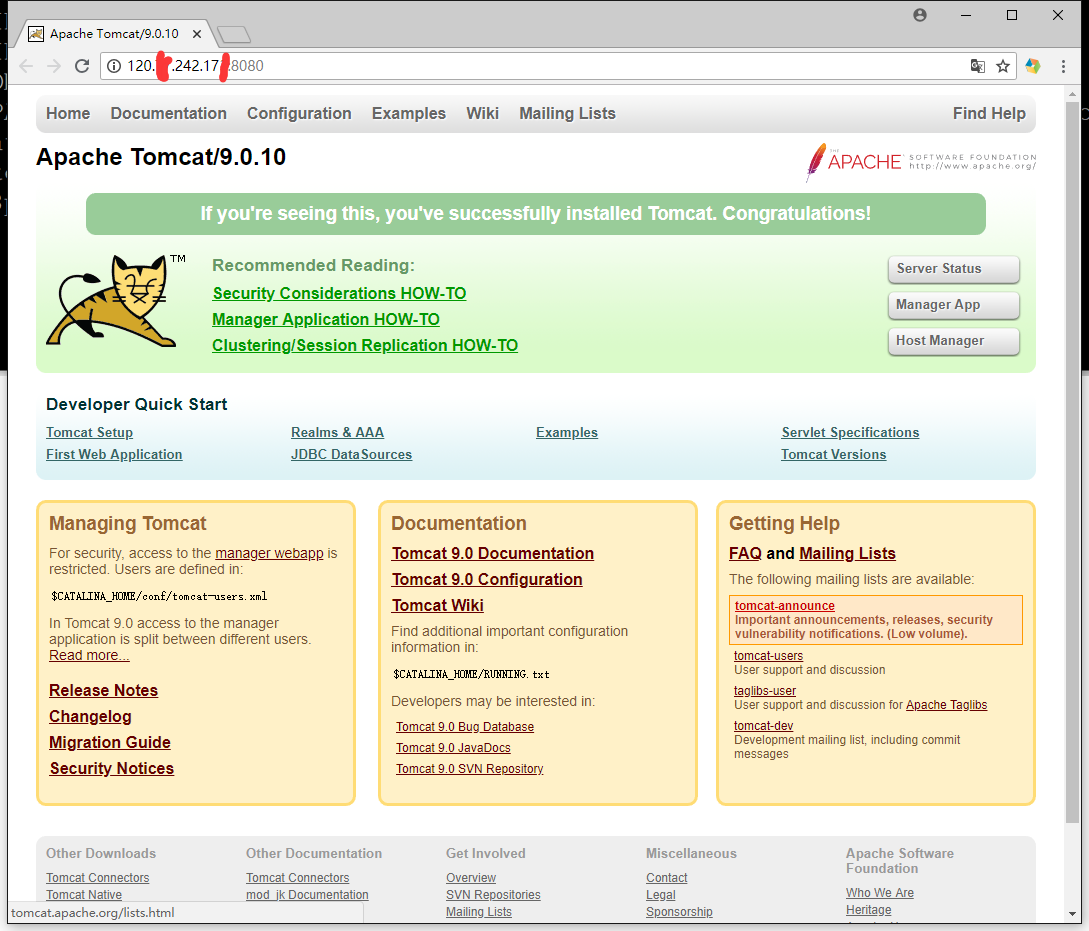
开启8080端口：

netstate -anp

iptables -A INPUT -ptcp --dport 8080 -j ACCEPT

阿里云服务器需要再管理中开放相关端口





# 权限管理

参考资料：http://man.linuxde.net/chmod

r 4 读

w 2 写

x 1 执行

u 拥有者

g 小组

o 其他人

a：所有人

-R 递归文件夹

groupadd 创建组 groupadd python

useradd 创建用户 useradd qiuguochang

-m 同时创建用户目录

-g 添加到对应组

useradd -m -g python qiuguochang

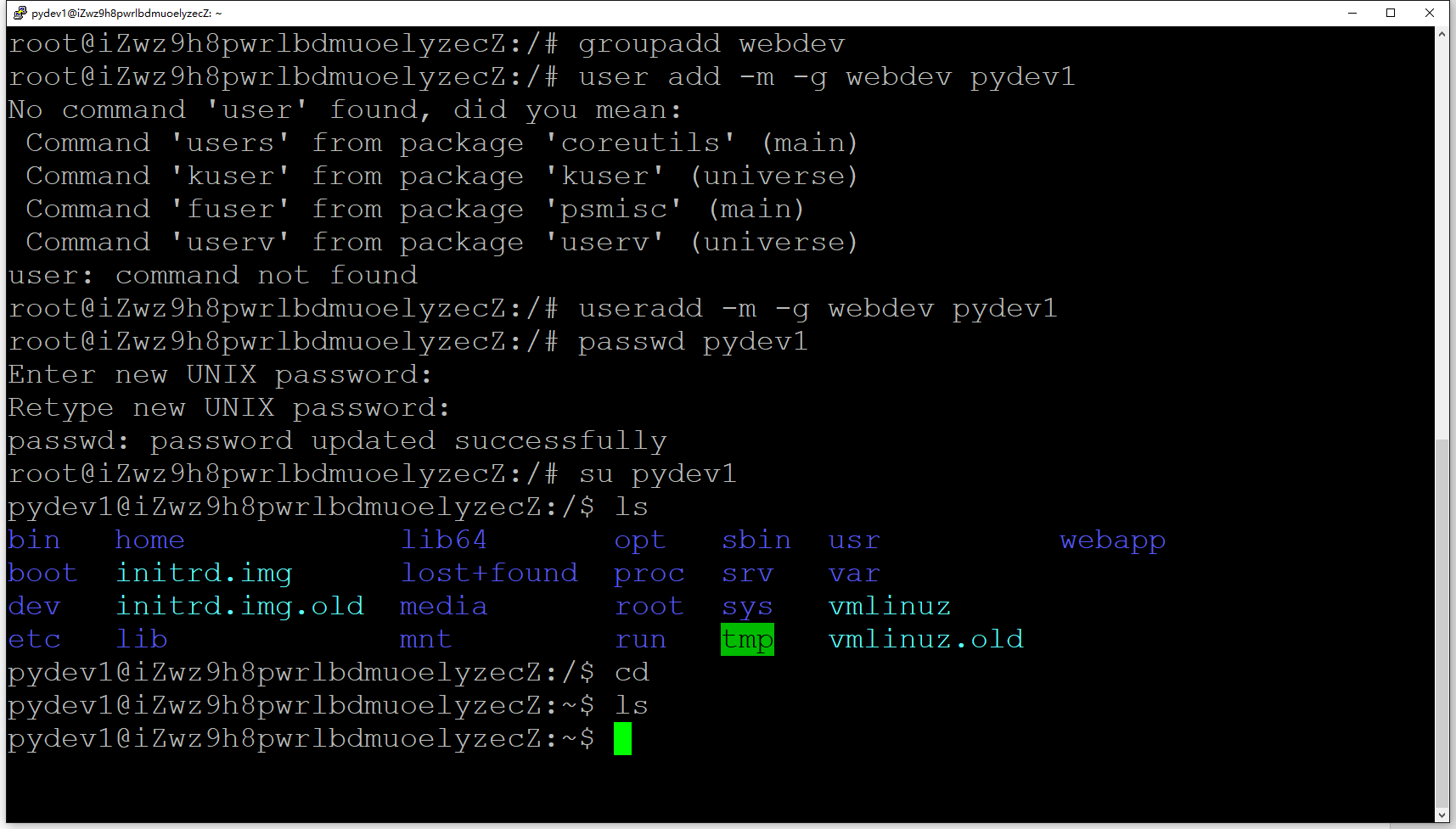
不加参数，默认加入用户同名组

id 用户名 查看该用户的组信息

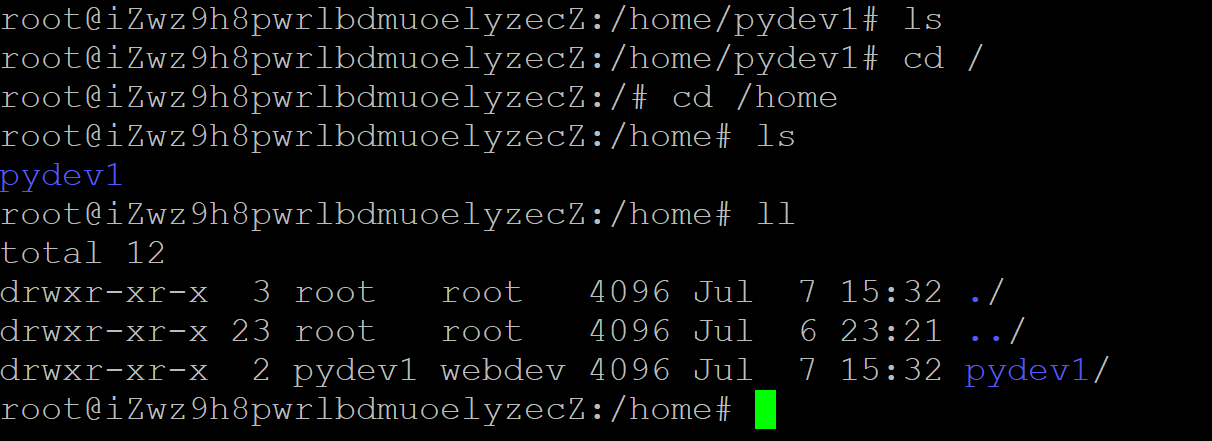
userdel 删除用户

passwd 用户名 密码 设置用户密码

chown 用户名 文件夹 转让文件夹



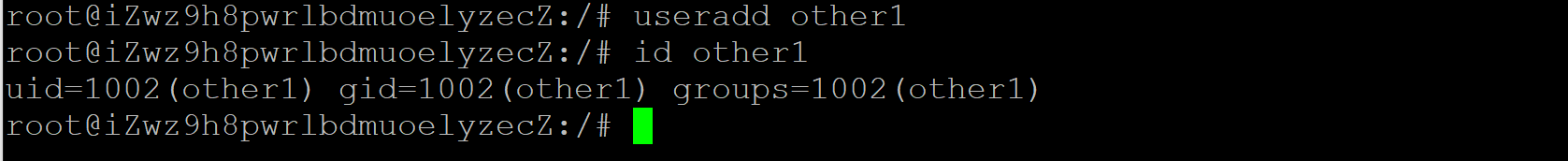
创建组：webdev ，创建pydev1并加入webdev组:



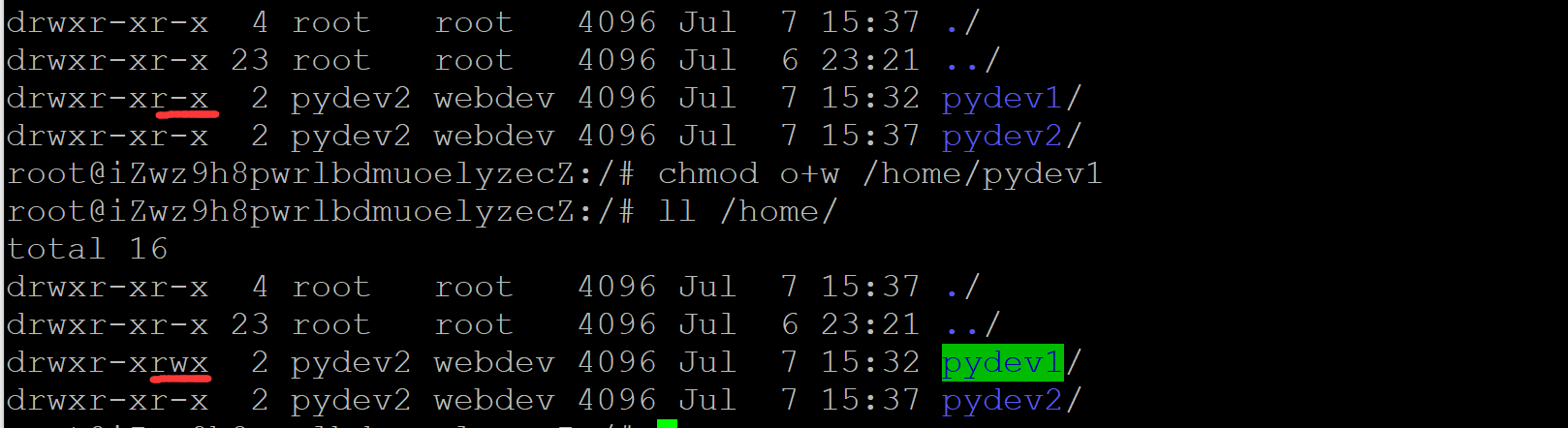
创建pydev2用户，并将/home/pydev1 目录所有权转让给pydev2：



创建用户other1,不属于任何组



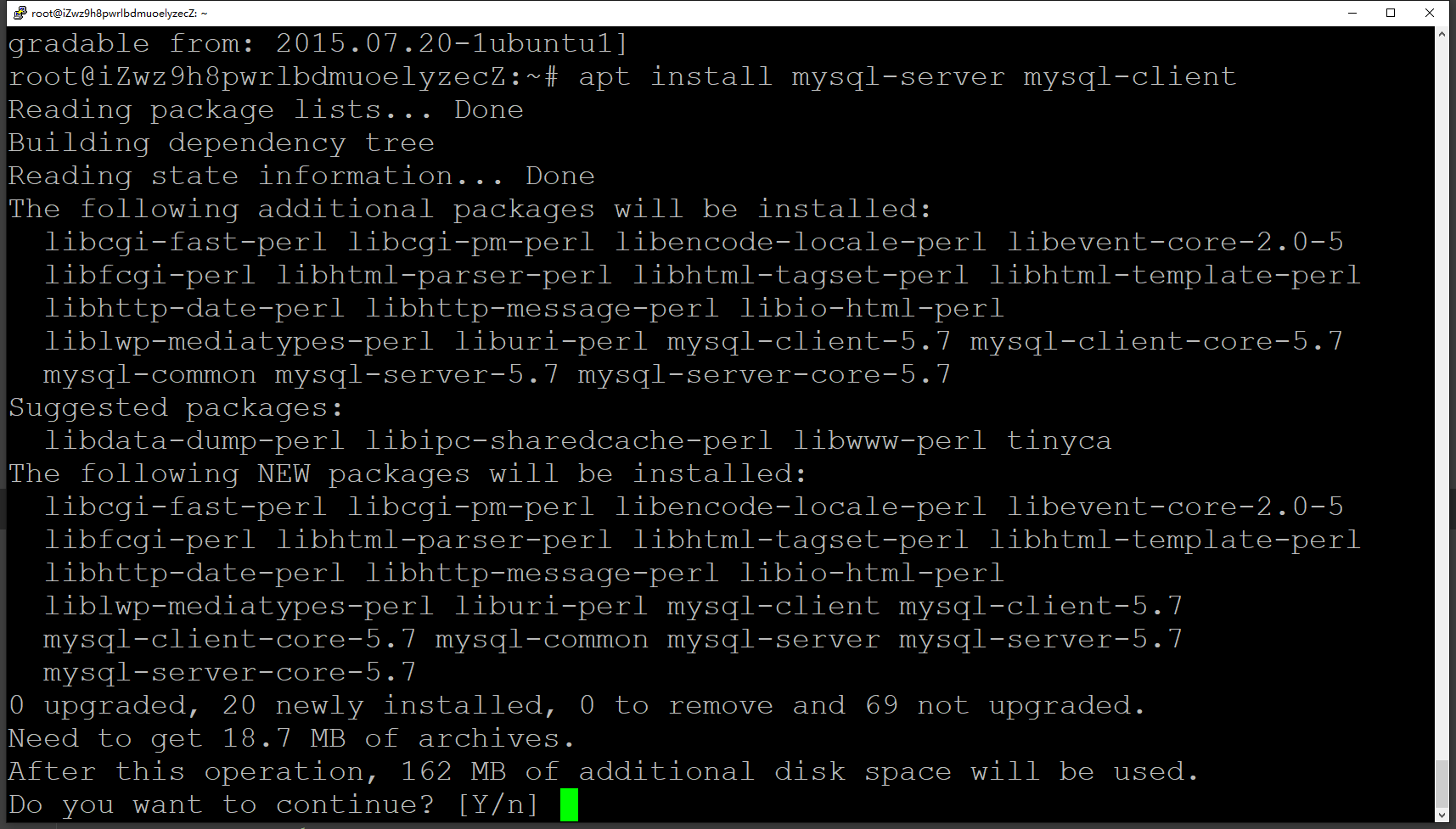
给other1 授予/home/pydev1的写权限

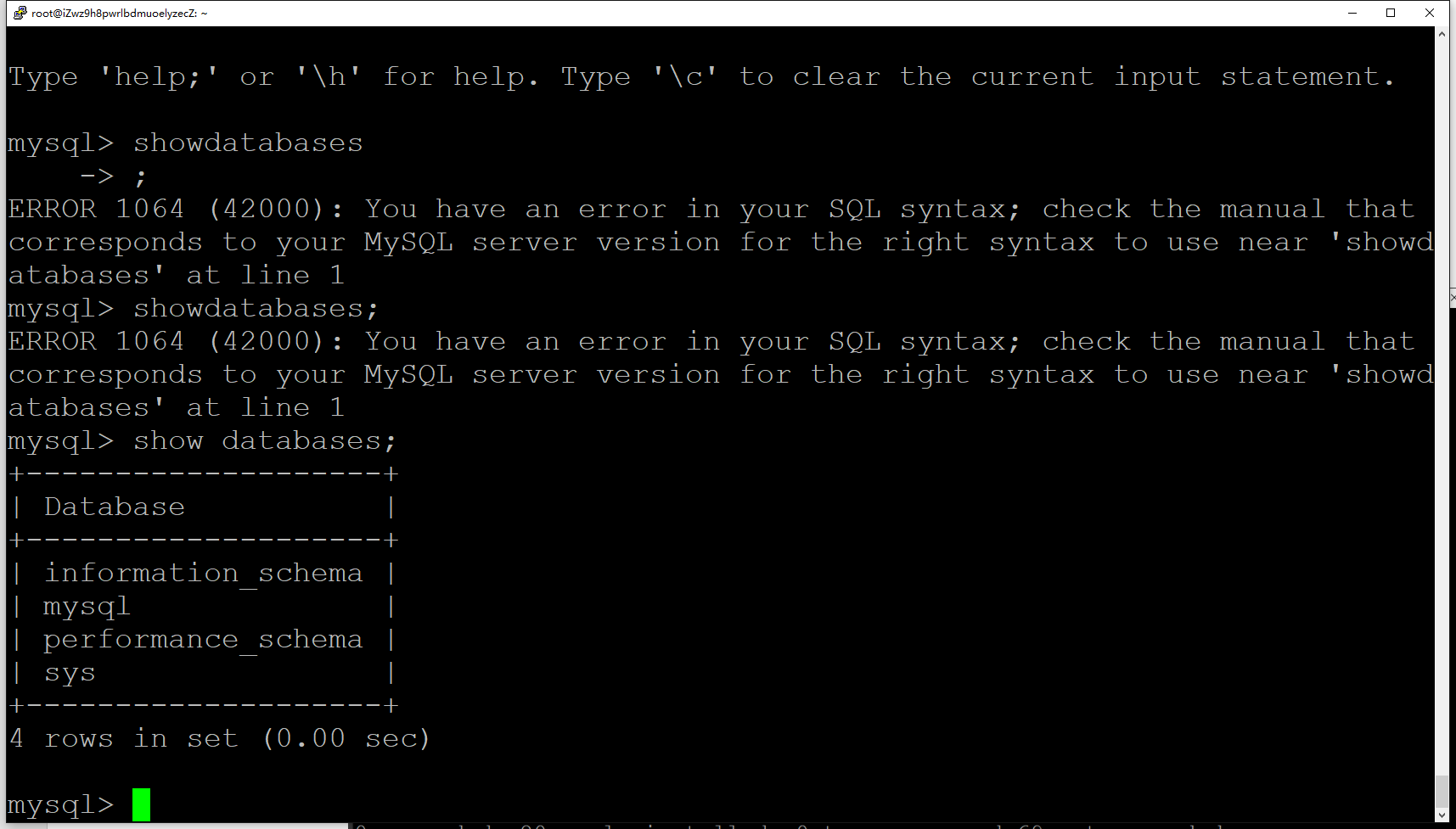


# Django商城项目部署

1. Mysql数据库安装

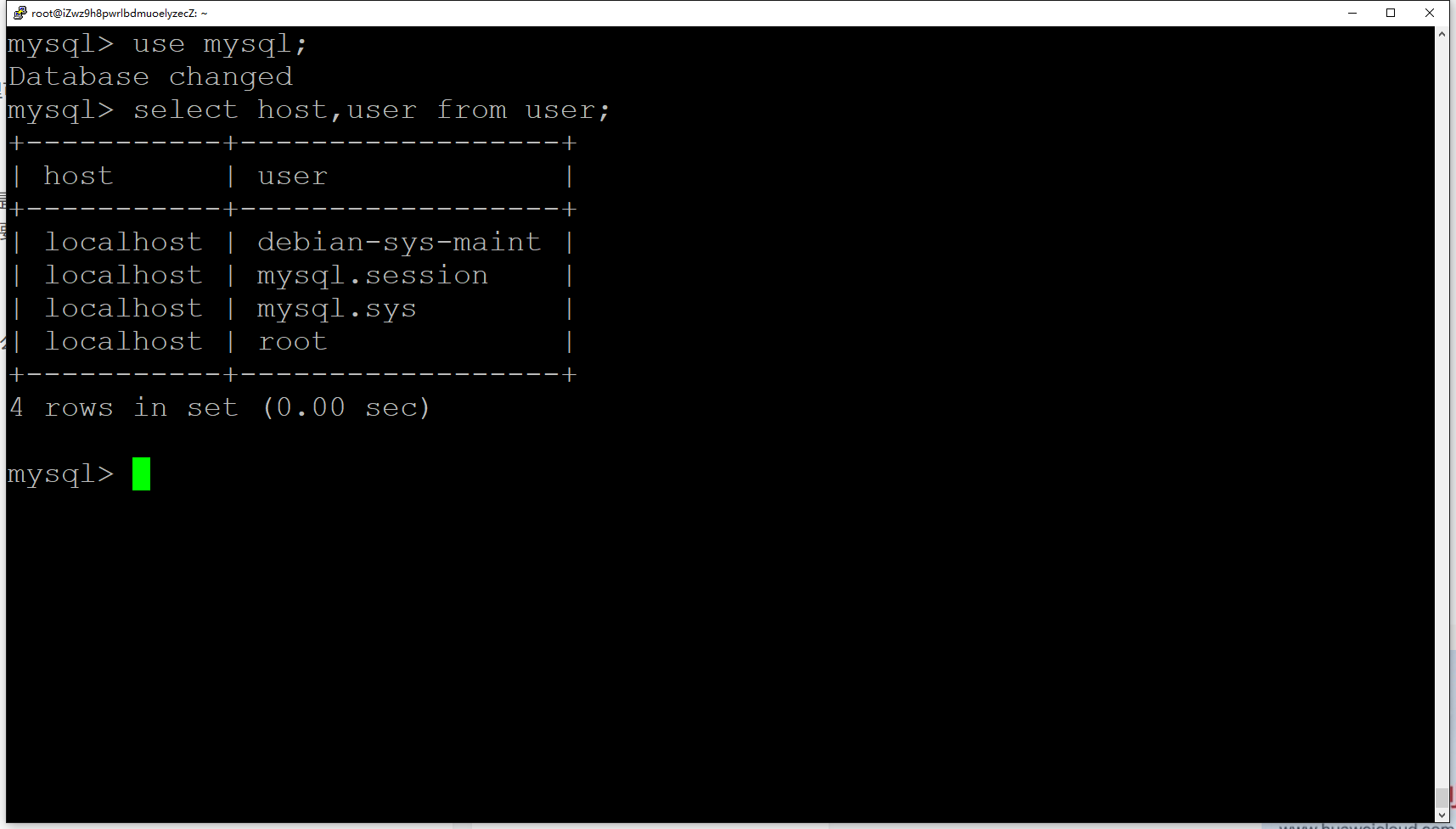
apt install mysql-server mysql-client





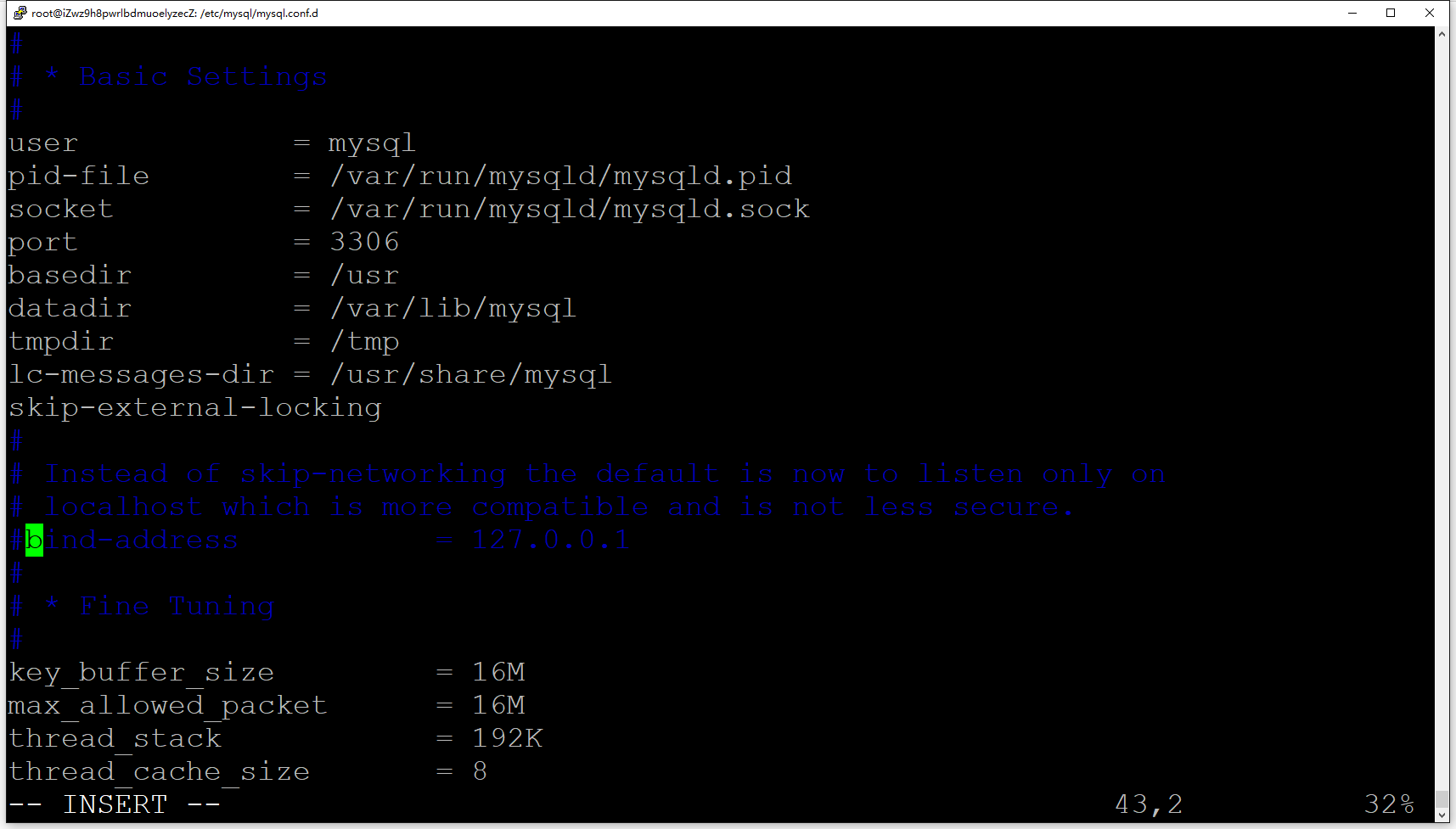
数据库导入可通过2种方式，一种是通过dump文件上传到服务器后导入

一种直接通过本地workbench远程连接服务器数据库进行导入



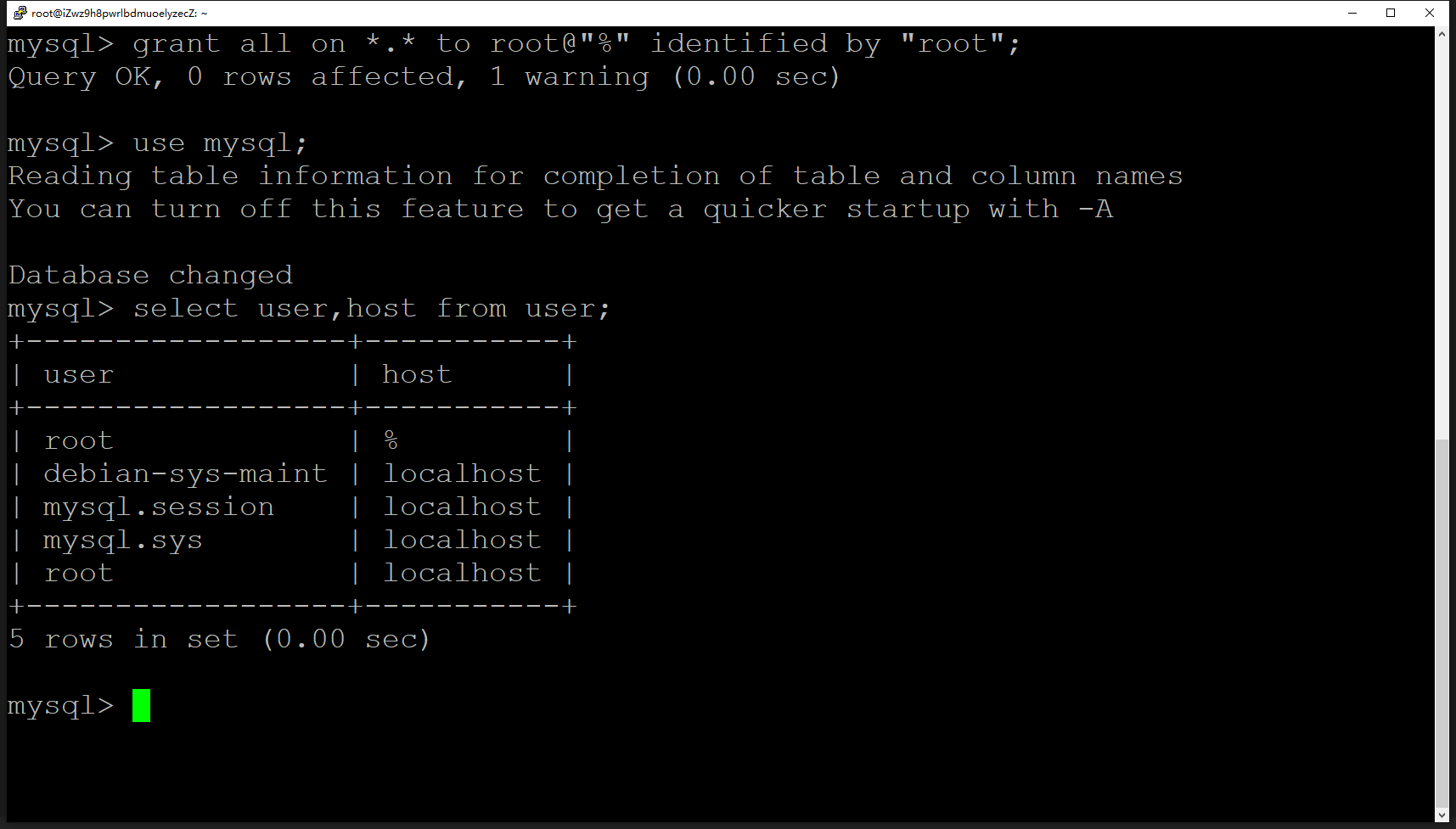
在user表中目前只有localhost的用户，需要添加远程访问的用户

1. 在MySQL配置文件中，注释掉绑定地址
2. etc/mysql/mysql.conf.d

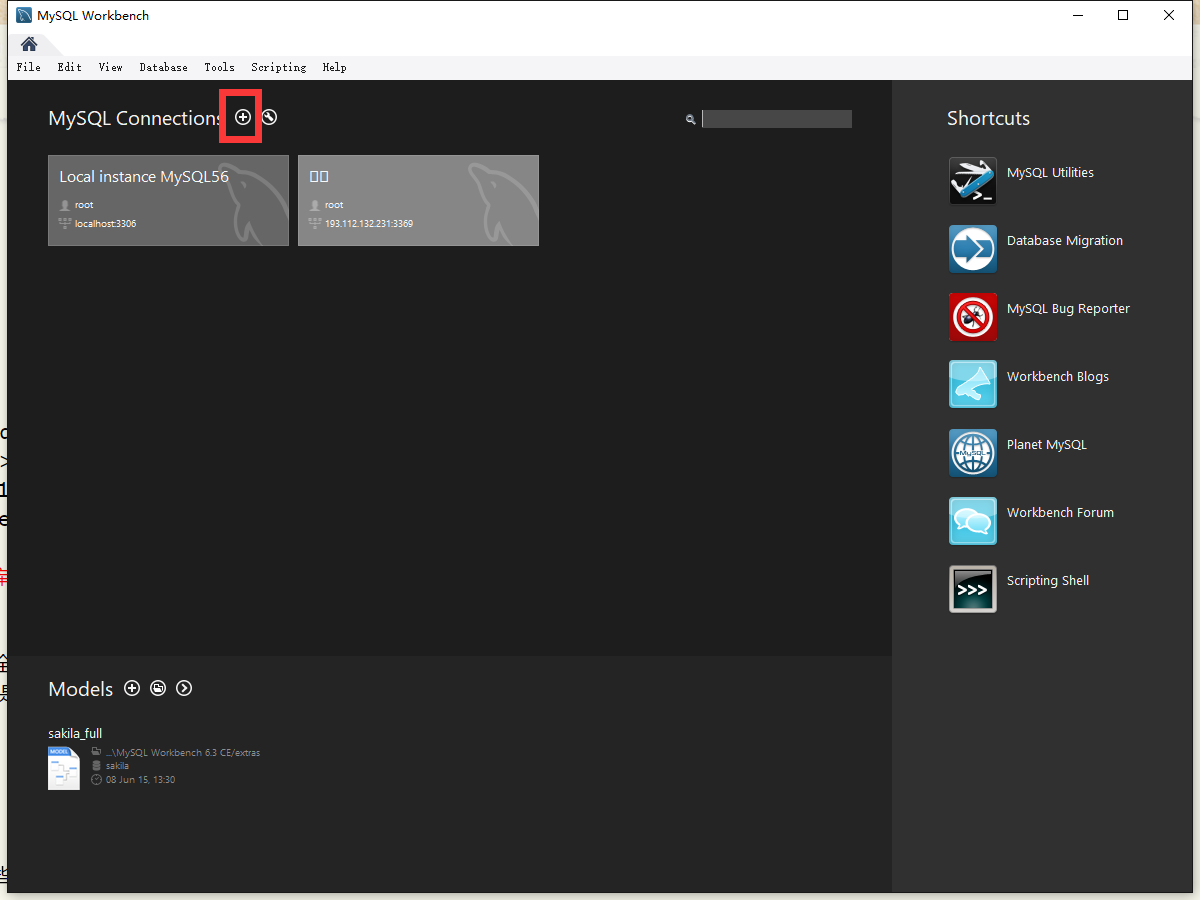


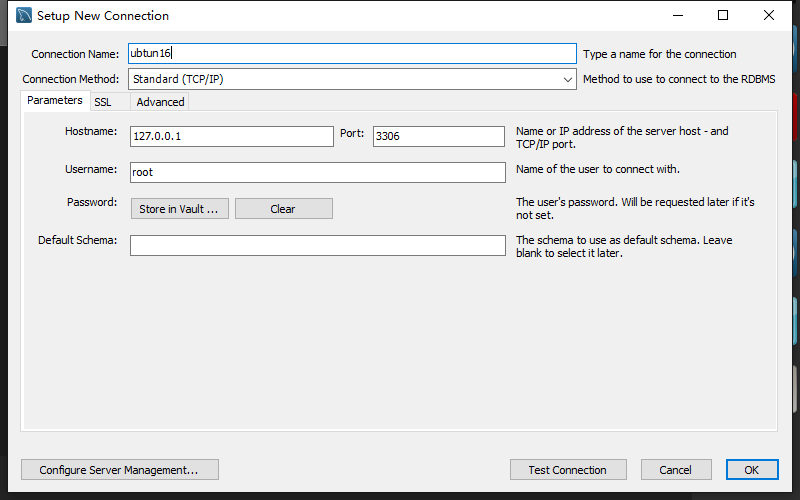
grant all on \*.\* to root@"%" identified by "root"

\*代表数据库及数据表，%代表主机地址，identified by 密码



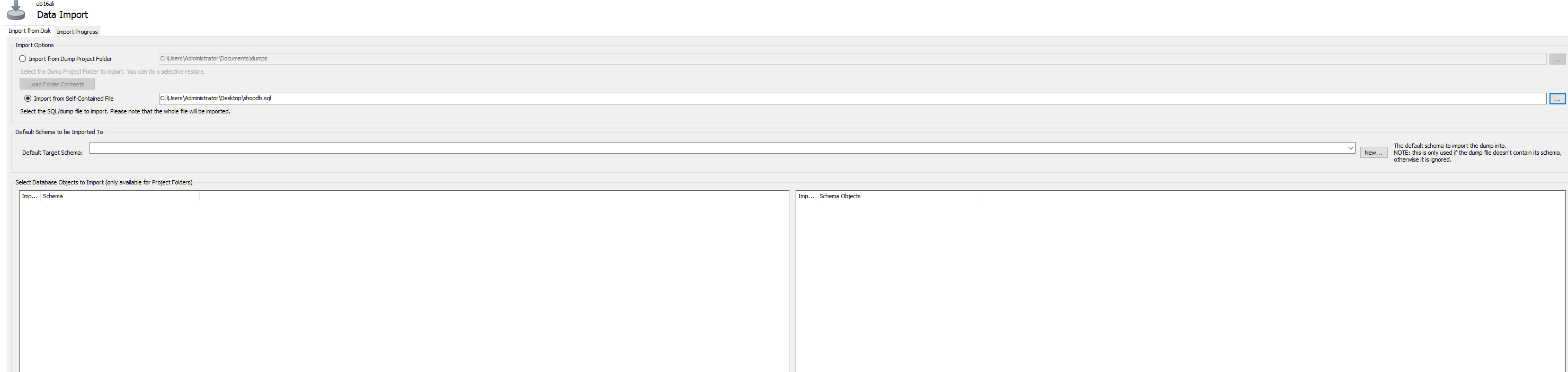
修改完成后需要重启MySQL服务！！



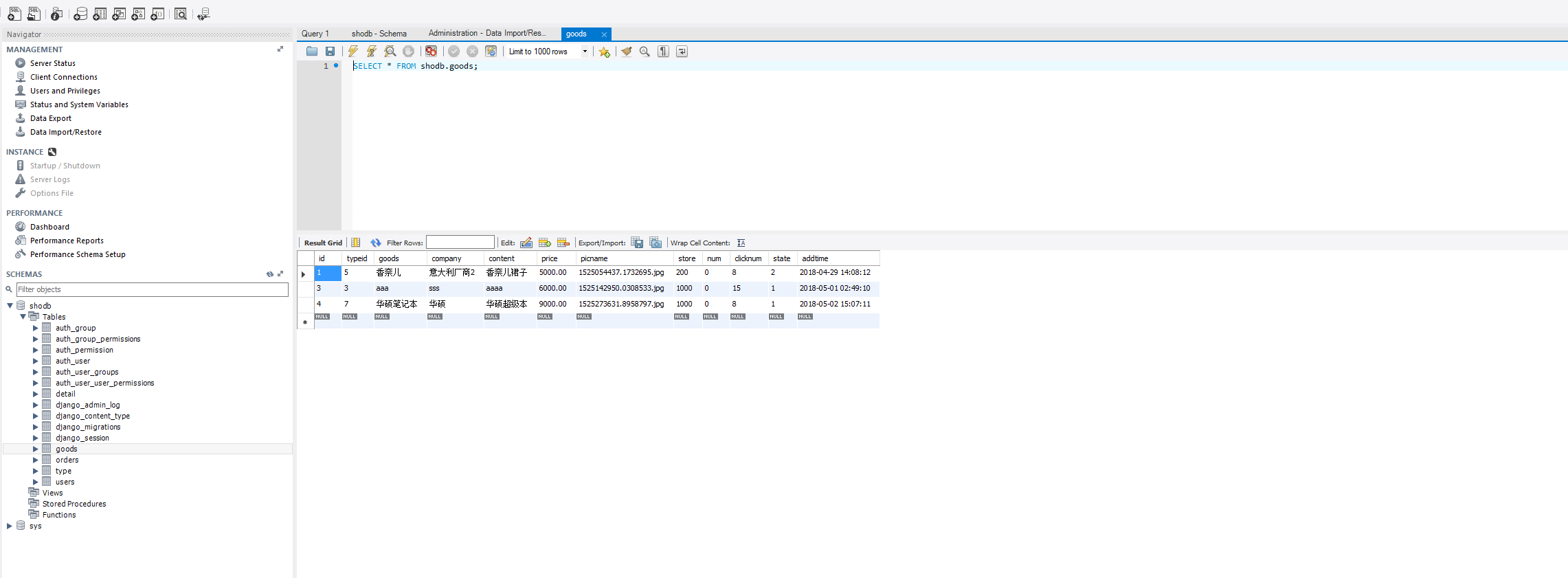




将原来的本地数据库导入远程数据库

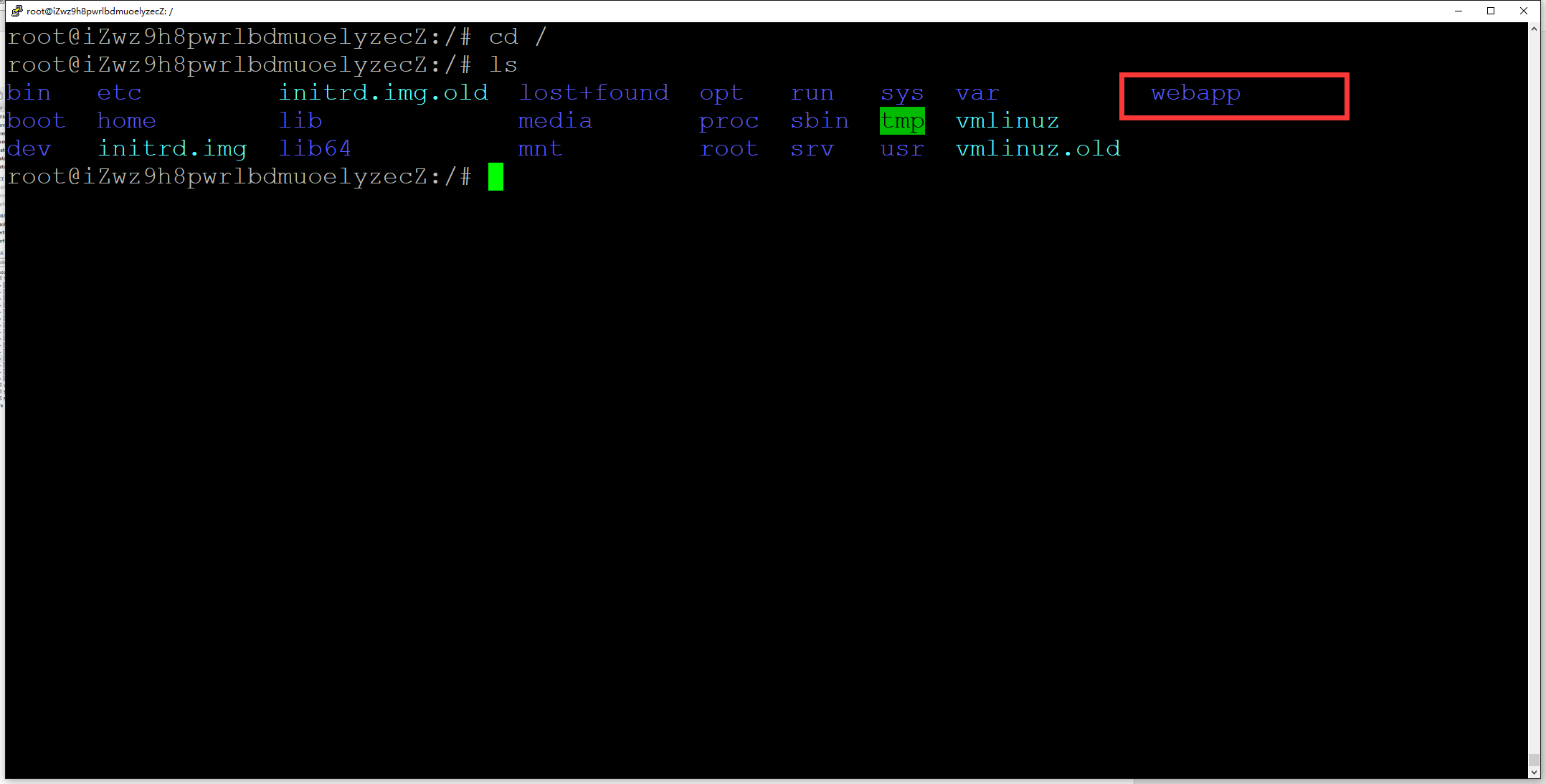


导入成功



1. uwsgi部署

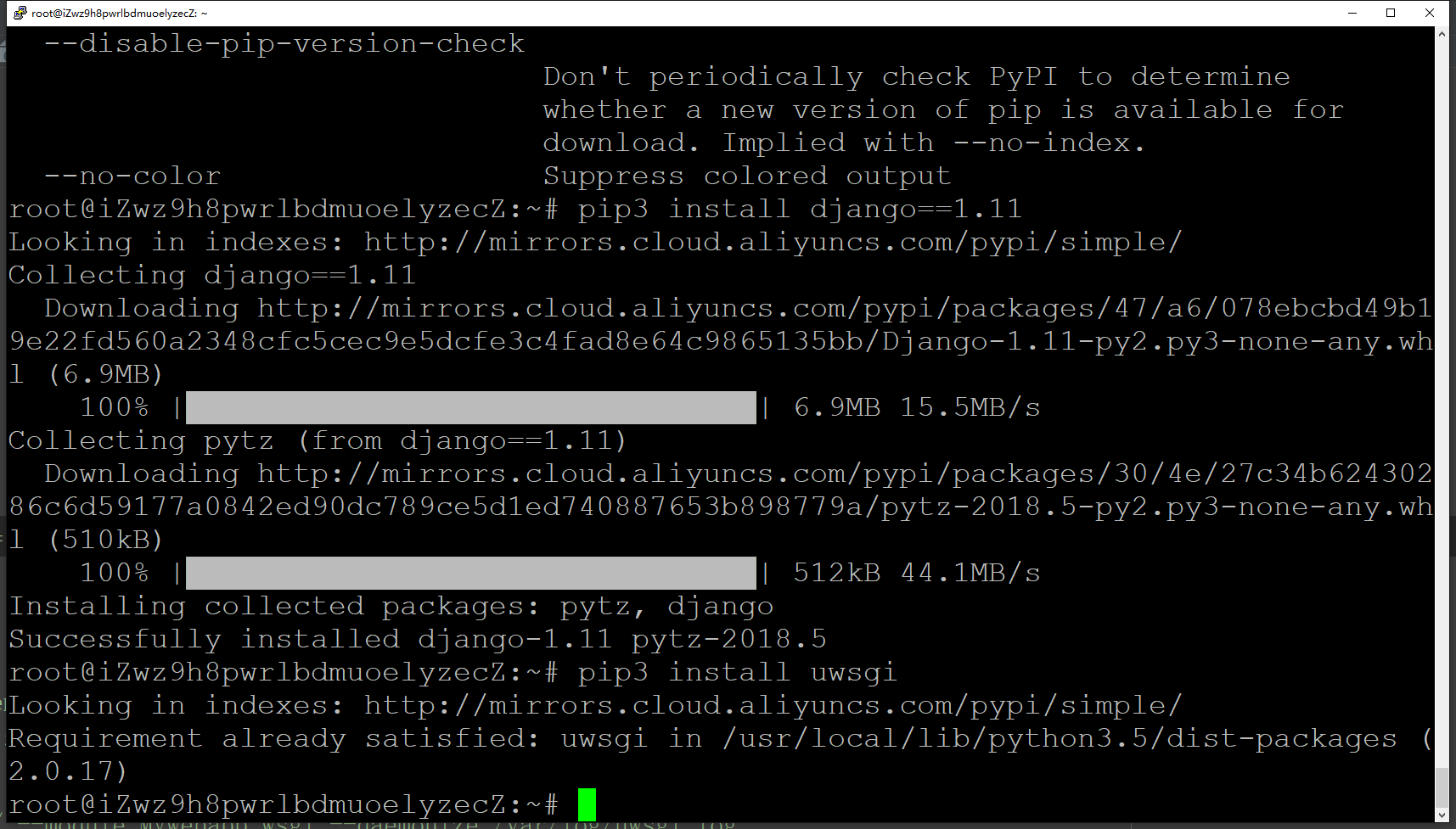
安装uwsgi: apt install uwsgi

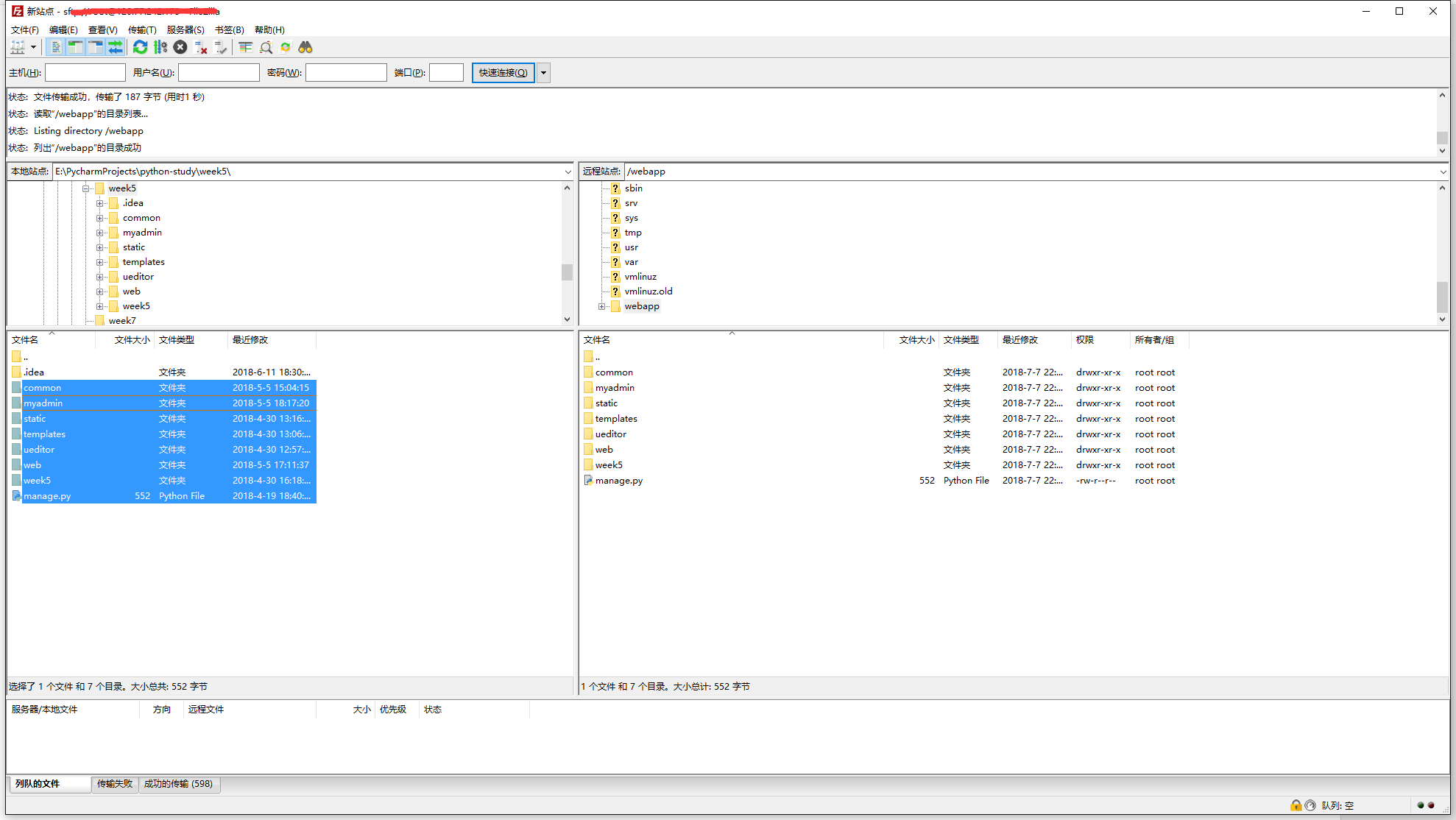


在根目录下创建webapp目录，存放dijango项目根目录

# 通过filezilla将项目文件传输至该目录

通过pip3 安装Django和uwsgi







1. 网站外部地址访问

uwsgi --http :80 --chdir /root/demo --module demo.wsgi --static-map=/static=static --daemonize /var/log/uwsgi.log

--home=/path/to/env 指定virtualenv 路径，如果没有可以去掉

--chdir 指定项目根目录(manage.py所在目录) --module 指定目录下的wsgi文件 --daemonizez 后台运行 指定log文件

--static-map=/static=static 做一个映射,指定静态文件，否则图片css无法显示

|  |  |
| --- | --- |
|  |  |

注意：完成后启动项目，会报错，需要安装相关包

apt install python3-pymysql

apt install Pillow

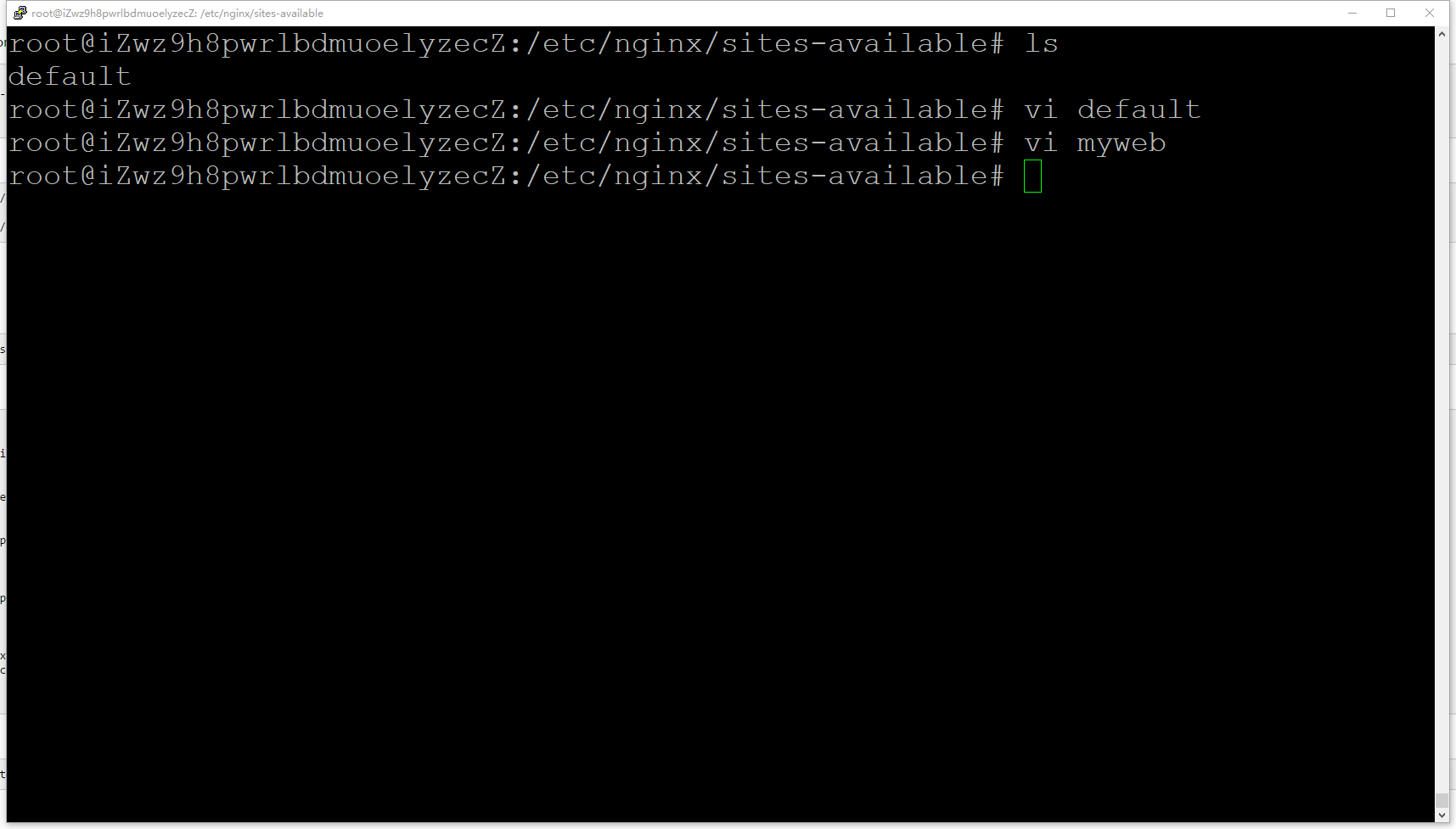
1. 配置nginx，静态资源优化

参考：<https://www.cnblogs.com/chenice/p/6921727.html>

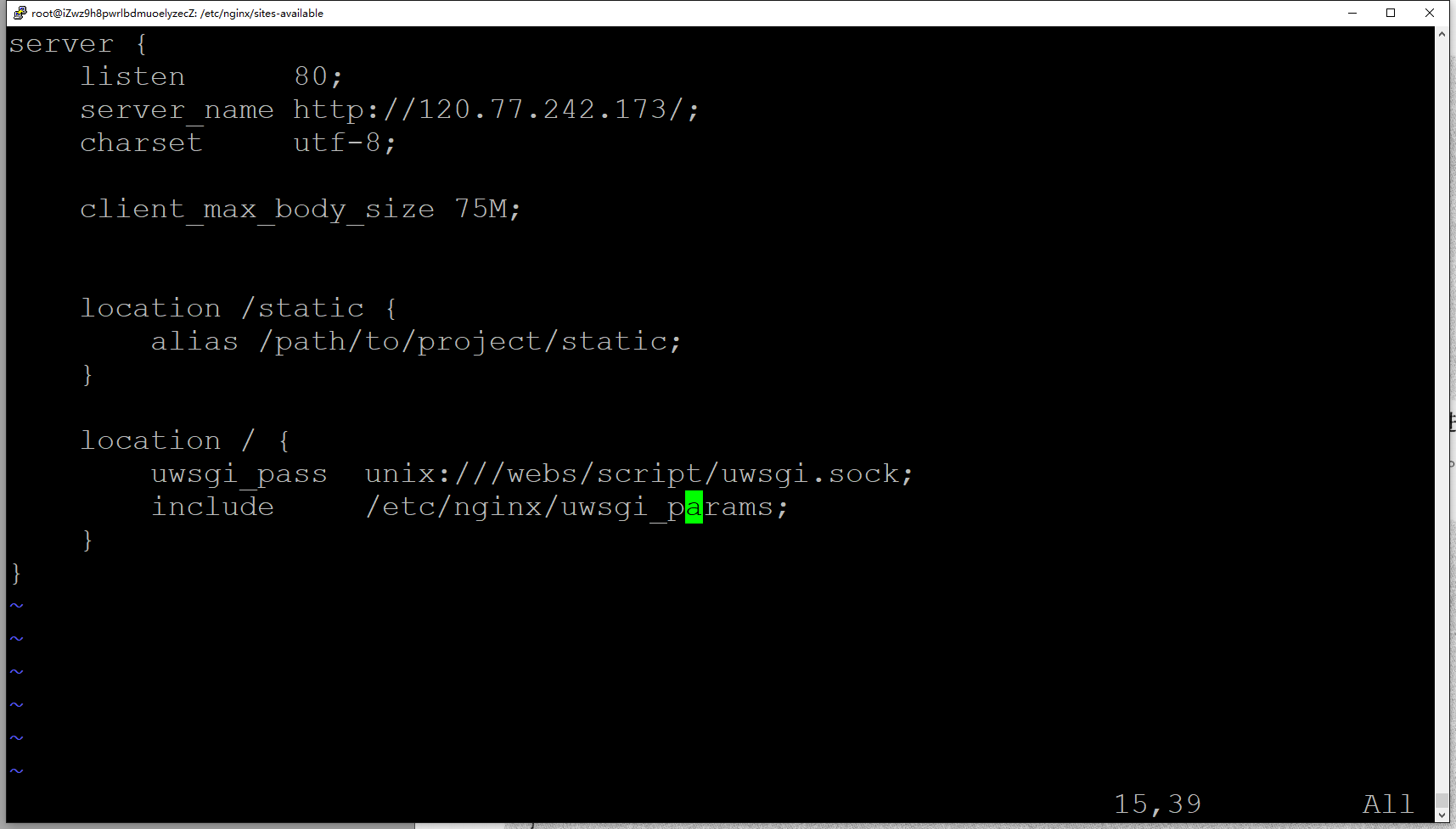
[uwsgi.ini]

*# uwsig使用配置文件启动  
# 使用命令：uwsig --ini -uwsgi.ini***[uwsgi]***# 项目目录*chdir=**/webs/webapp/***# 指定项目的application*module=**week5.wsgi:application***# 指定sock的文件路径*socket=**/webs/script/uwsgi.sock***# 进程个数*workers=**5**pidfile=**/webs/script/uwsgi.pid***# 指定IP端口  
http=:80  
  
# 指定静态文件*static-map=**/static=/webs/webapp/static***# 启动uwsgi的用户名和用户组*uid=**root**gid=**root***# 启用主进程*master=**true***# 自动移除unix Socket和pid文件当服务停止的时候*vacuum=**true***# 序列化接受的内容，如果可能的话*thunder-lock=**true***# 启用线程*enable-threads=**true***# 设置自中断时间*harakiri=**30***# 设置缓冲*post-buffering=**4096***# 设置日志目录*daemonize=**/webs/script/uwsgi.log**

uwsgi处理动态请求能力高，但对于静态请求（如static文件，css,js文件等）处理能力差，此时就要结合nginx一起使用



在Nginx 配置文件目录新建自己的配置文件 myweb



service nginx reload

service nginx restart

lsof -i:80

注意：请求的过程为，request→Nginx→socket→uwsgi

所以，在Nginx绑定80端口后，uwsgi中不需要再绑定80端口，否则会导致启动失败！！

商城地址如下：

http://120.77.242.173