**阶段四：企业应用部署**

**搭建企业应用部署环境**

# 一、安装VMWare虚拟机

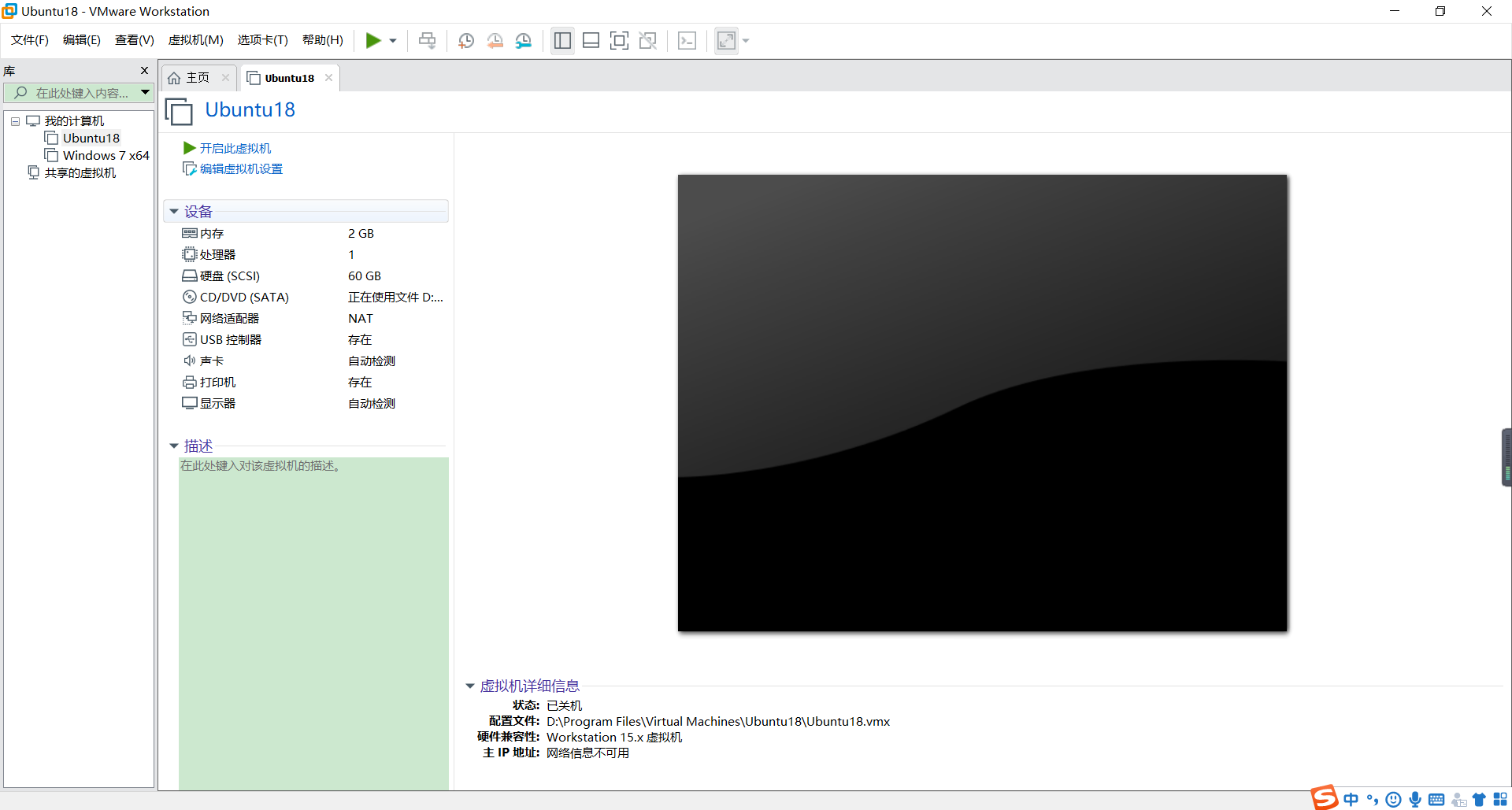


图1.1 VMWare虚拟机主页面

# 二、在虚拟机环境中安装Ubunut操作系统

2.1 在VMWare虚拟机主界面点击创建新的虚拟机

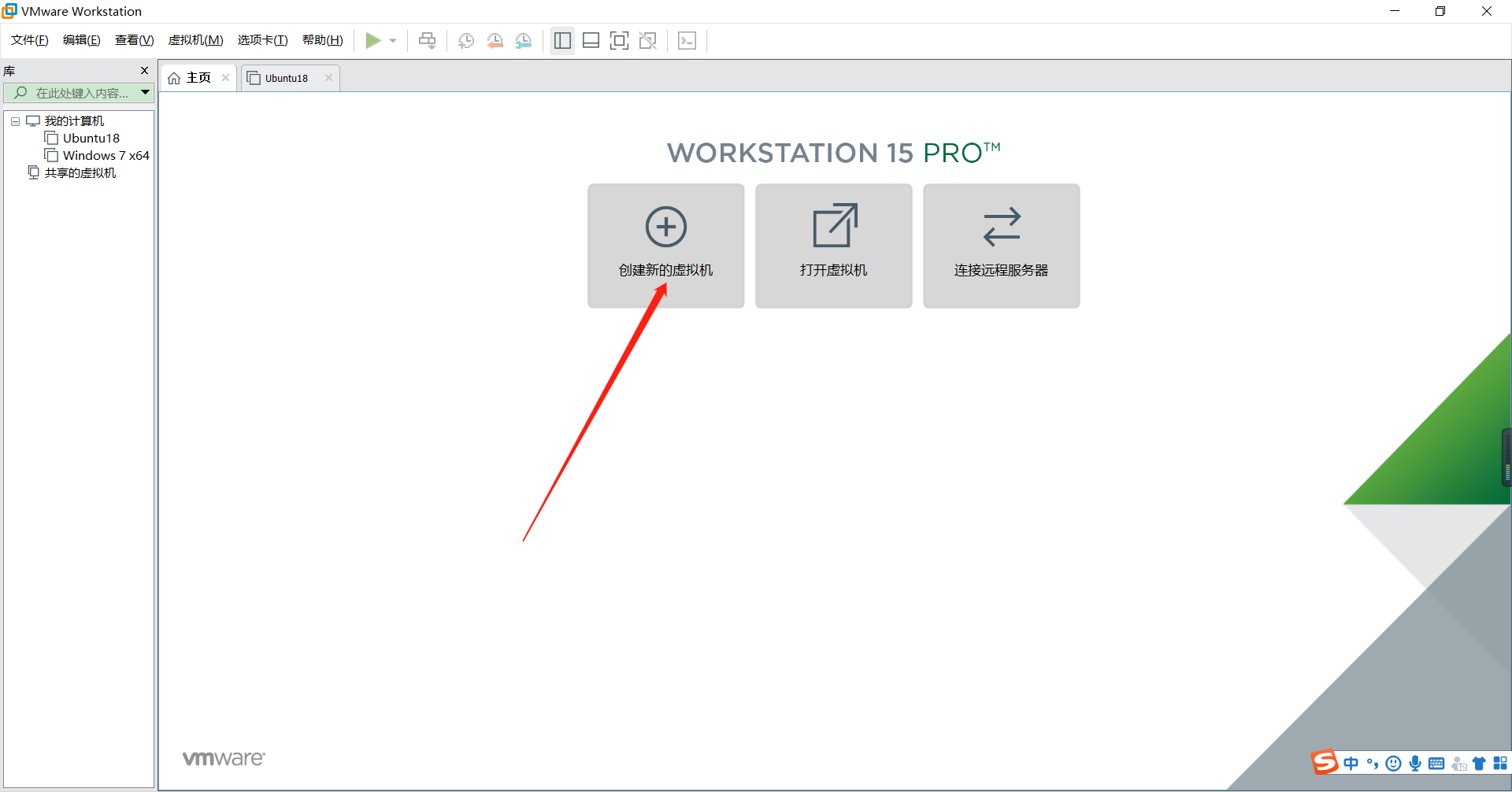


图2.1 选择创建新的虚拟机

2.2 选择Ubuntu镜像文件所在位置

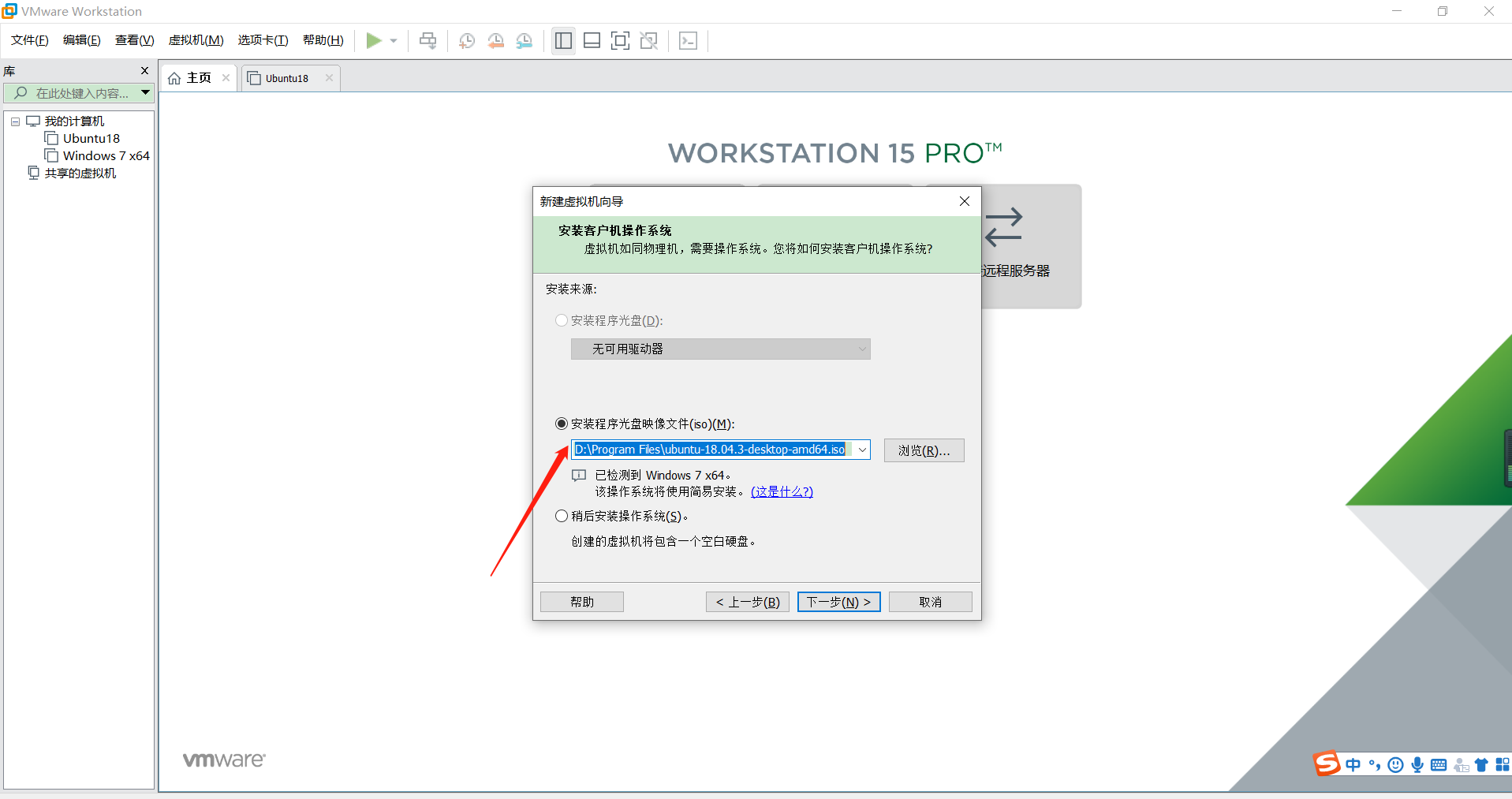


图2.2 选择Ubuntu镜像文件所在位置

2.3 按照提示进行安装即可

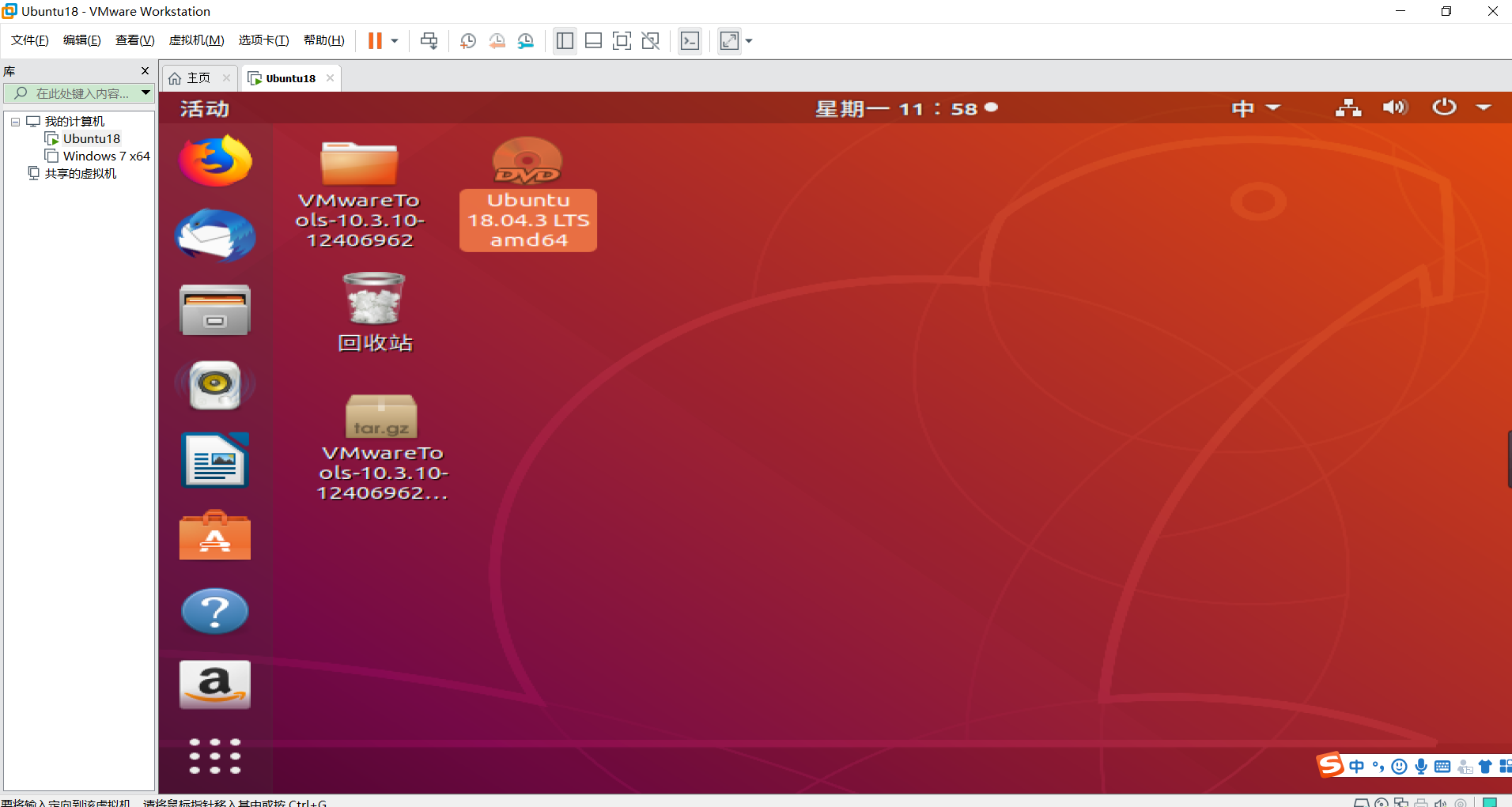


图2.3 Ubuntu主界面

# 三、在Linux系统中安装JDK并完成环境变量配置

3.1 官网下载JDK。

进入网址：

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

选择Linux x64版本的后缀位tar.gz的压缩包进行下载。

3.2 进行解压

1.在Ubuntu的根目录下创建 /java 文件夹，首先需要更改用户权限：

1. 输入命令：**cd/ ，**进入到根目录，
2. 输入命令：**sudo passwd root，**终端会提示你让你给root用户设置密码，设置新密码后继续确认新密码。密码是隐藏的，所以不会显示出来。
3. 设置好root用户后输入命令：**su root，**输入root用户密码；当输入框前面的“$”符号变成“#”时，就可以使用root权限了。
4. 输入命令：**mkdir /java，**在根目录下创建/java目录

2.将下载好的JDK文件移动到/java目录下

输入命令：**sudo mv jdk-8u241-linux-x64.tar.gz /java**

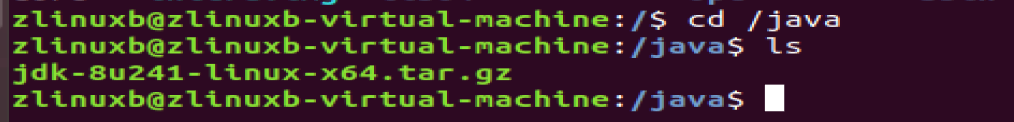


图3.1 压缩文件已成功移动到

3. 解压文件

输入命令：**tar -zxvf jdk-8u241-linux-x64.tar.gz**

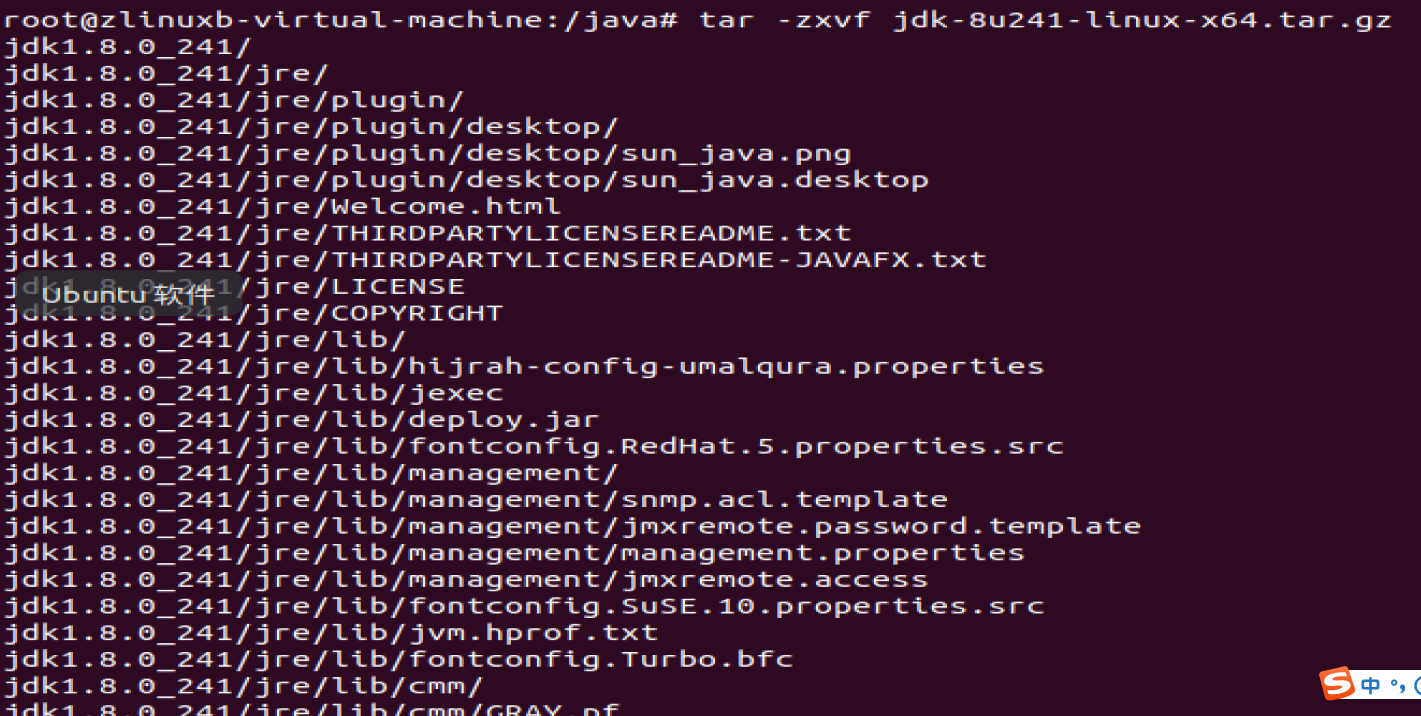


图3.2 文件已解压成功

3.3 配置环境变量

（1）输入命令：**$sudo gedit ~/.bashrc**

（2）然后把如下命令复制到最底部：

**export JAVA\_HOME=/java/ jdk1.8.0\_241**

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

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

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

（3）保存文件

（4）输入命令：**$sudo source ~/.bashrc ，**使文件生效

3.4 测试JDK是否安装成功

输入命令：**java -version**

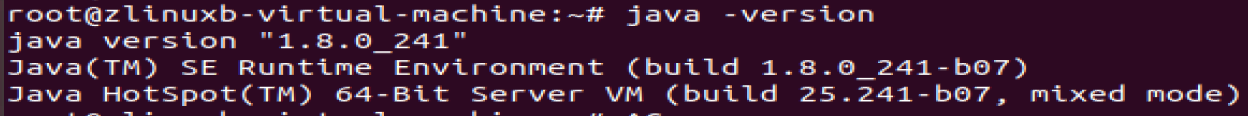


图3.3 查看JDK版本

# 四、安装Nginx服务器并启动

4.1 官网下载“nginx-1.9.9.tar.gz”文件，进入下载网址：

<https://nginx.org/download/>

4.2 解压文件

（1）首先在根目录下建立/nginx目录，并将压缩文件移动到该目录下：

输入命令：**mkdir /nginx**

输入命令：**sudo mv nginx-1.9.9.tar.gz /nginx**

（2）解压文件：

输入命令：**tar -zxvf nginx-1.9.9.tar.gz**

4.3 配置

（1）如果linux中没有安装gcc编译器，首先需要安装gcc编译器，否则nginx无法编译安装。

安装gcc：**sudo apt-get install build-essential**

查看gcc版本：**gcc –-version**

（2）安装相关依赖包openssl、libssl-dev，否则后面的命令会报错，ubuntu中使用命令：

输入命令：**sudo apt-get install openssl**

输入命令：**sudo apt-get install libssl-dev**

输入命令：**sudo apt-get install ruby**

输入命令：**sudo apt-get install zlib1g-dev**

注意：如果是CentOS 则使用yum源进行安装，ubuntu中默认使用apt源进行安装。

（3）进入到nginx解压后所在的文件夹下，执行命令：**./configure**

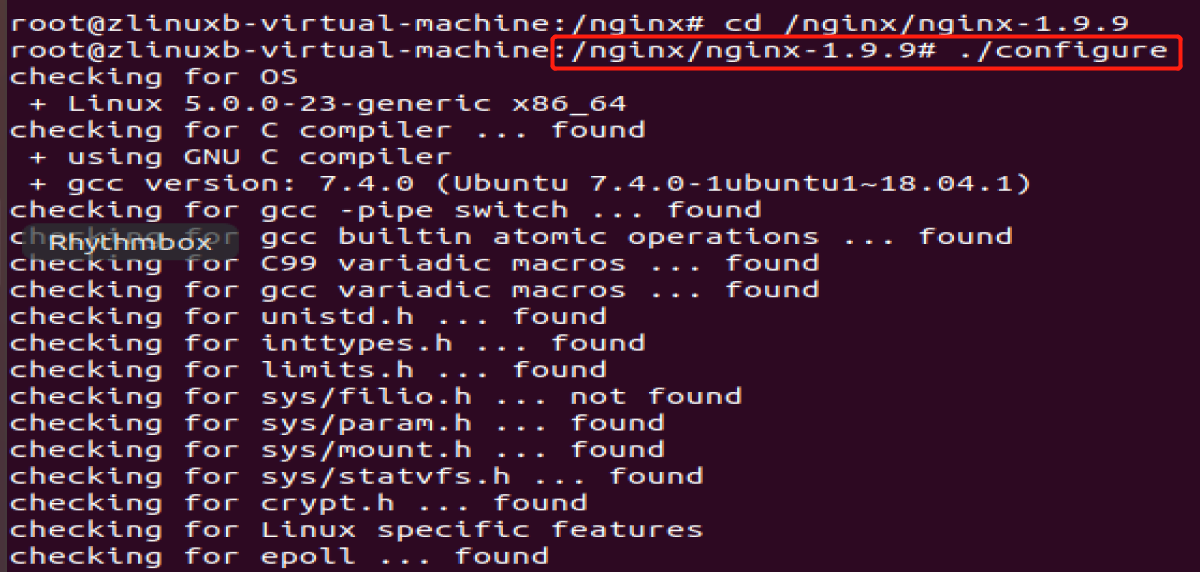


图3.4 配置



图3.5 配置成功

4.4 安装

（1）执行make、make install

输入命令：**make**

输入命令：**make install**

4.5 启动Nginx

输入命令：**/usr/local/nginx/sbin/nginx**

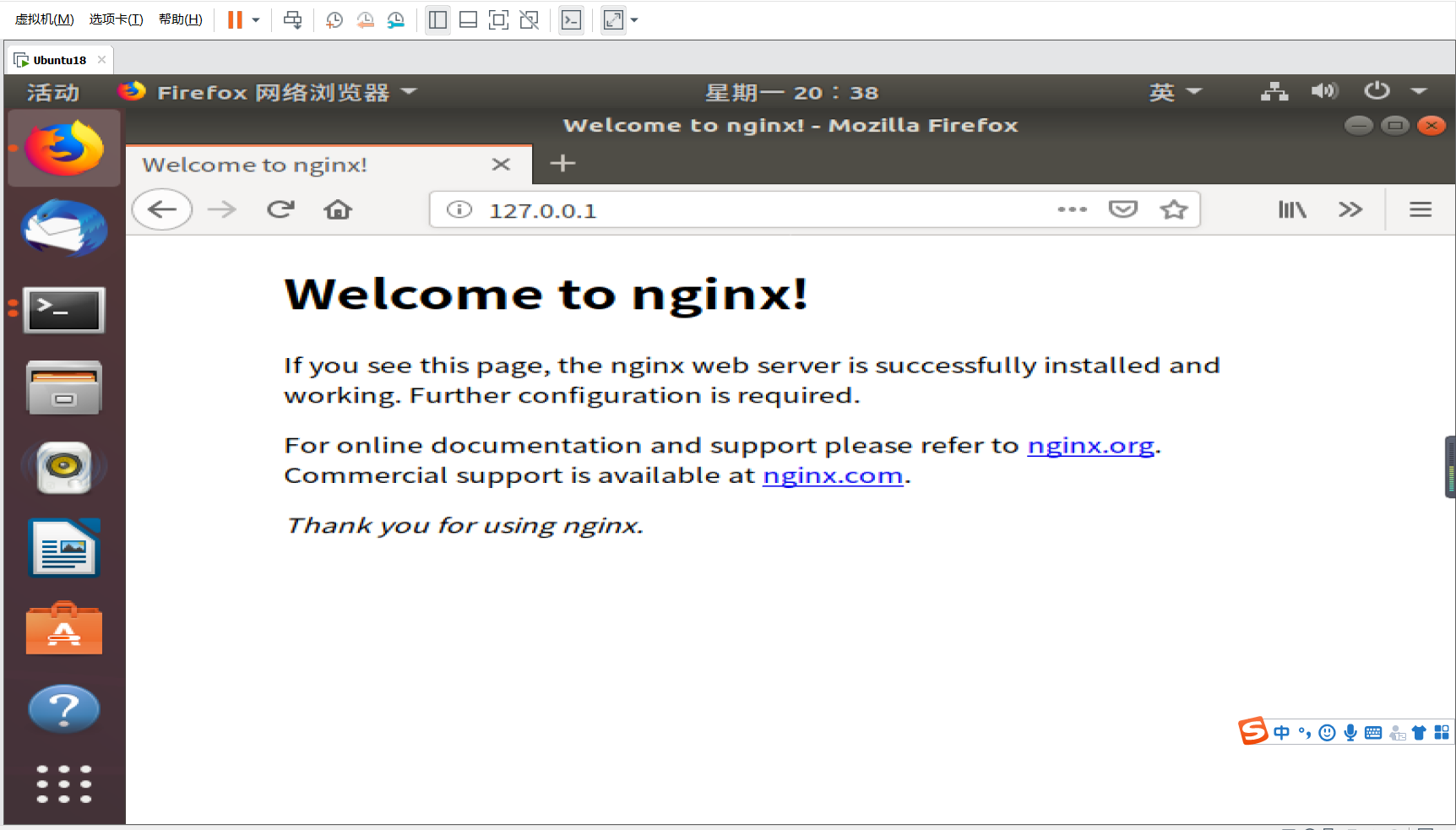


图3.6 Nginx已成功启动

4.6 windows 系统中访问 Linux 中 nginx

（1）查看Linux的ip：

输入命令：**ifconfig -a**

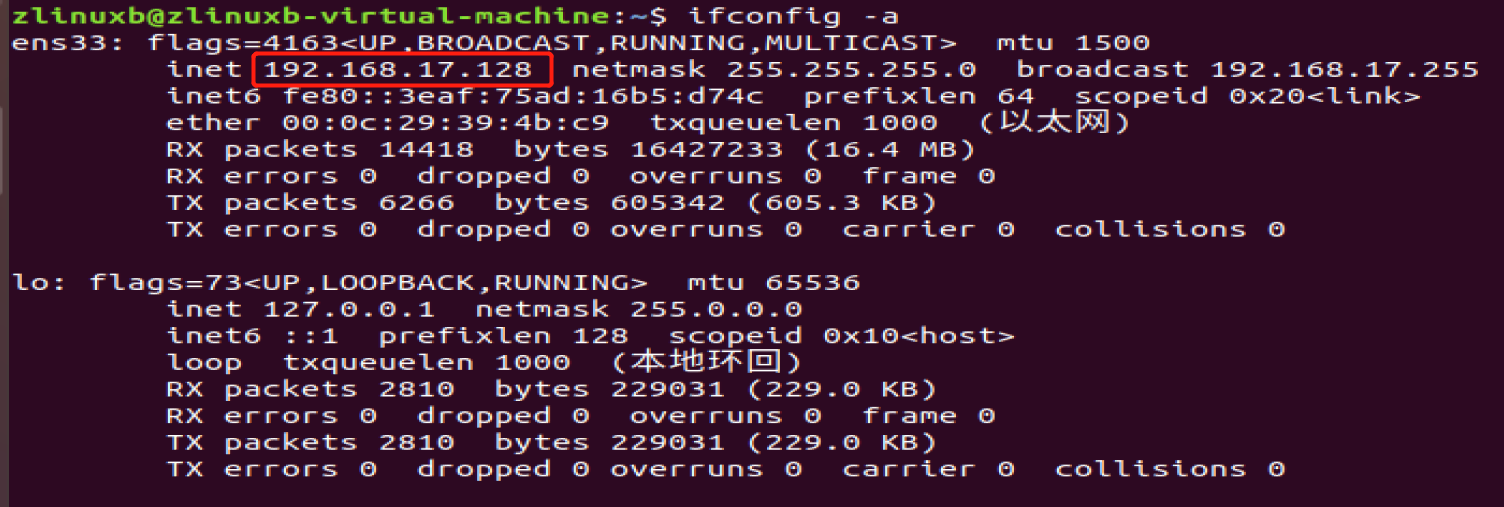


图3.7 查看Linux的IP地址

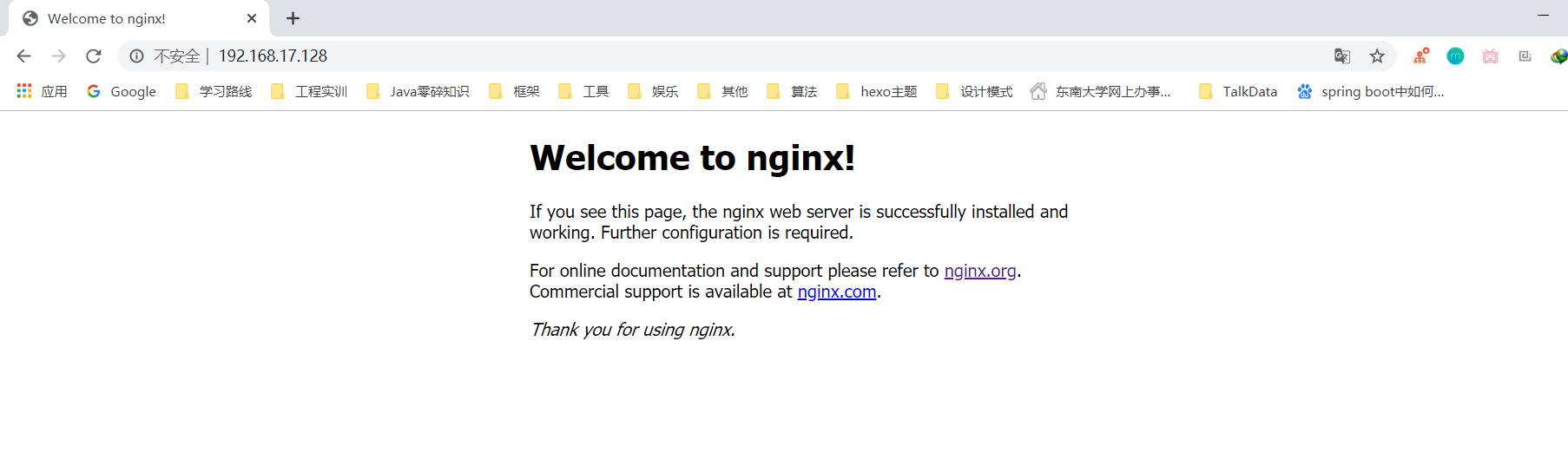


图3.8 Windows中访问Linux中的nginx

注意：若windows中无法访问，可能是防火墙的原因，此时需要关闭Linux的防火墙，并开放访问的端口号：80 端口。

# 五、将京东首页部署至SpringBoot项目，并启动三个不同Web服务器

5.1 创建SpringBoot项目，并导入相关的pom依赖，将京东首页放到resources/static目录下即可

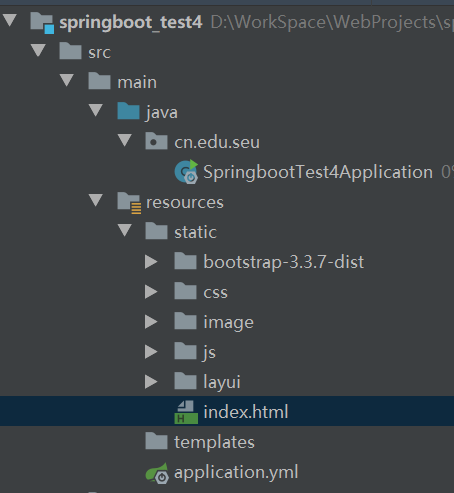


图 5.1 项目目录

5.2 IDEA启动多个SpringBoot项目

（1）设置IDEA，点击Edit Configuration

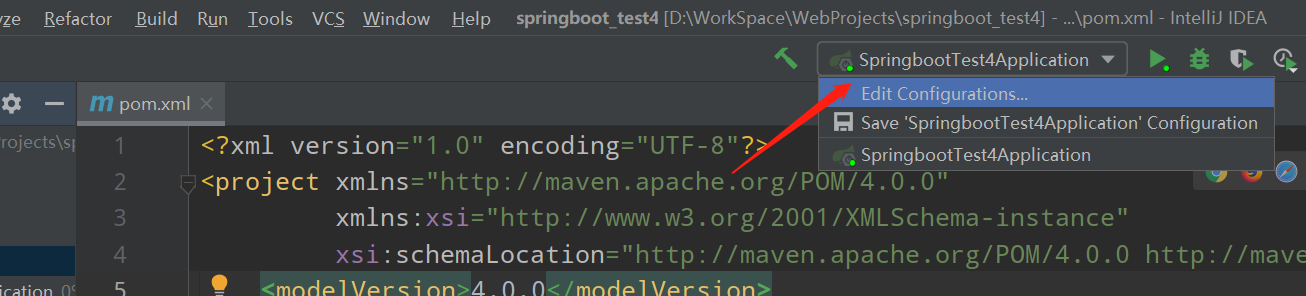


图 5.2 点击Edit Configuration

（2）打开配置后，将默认的Allow prarallel run打钩。

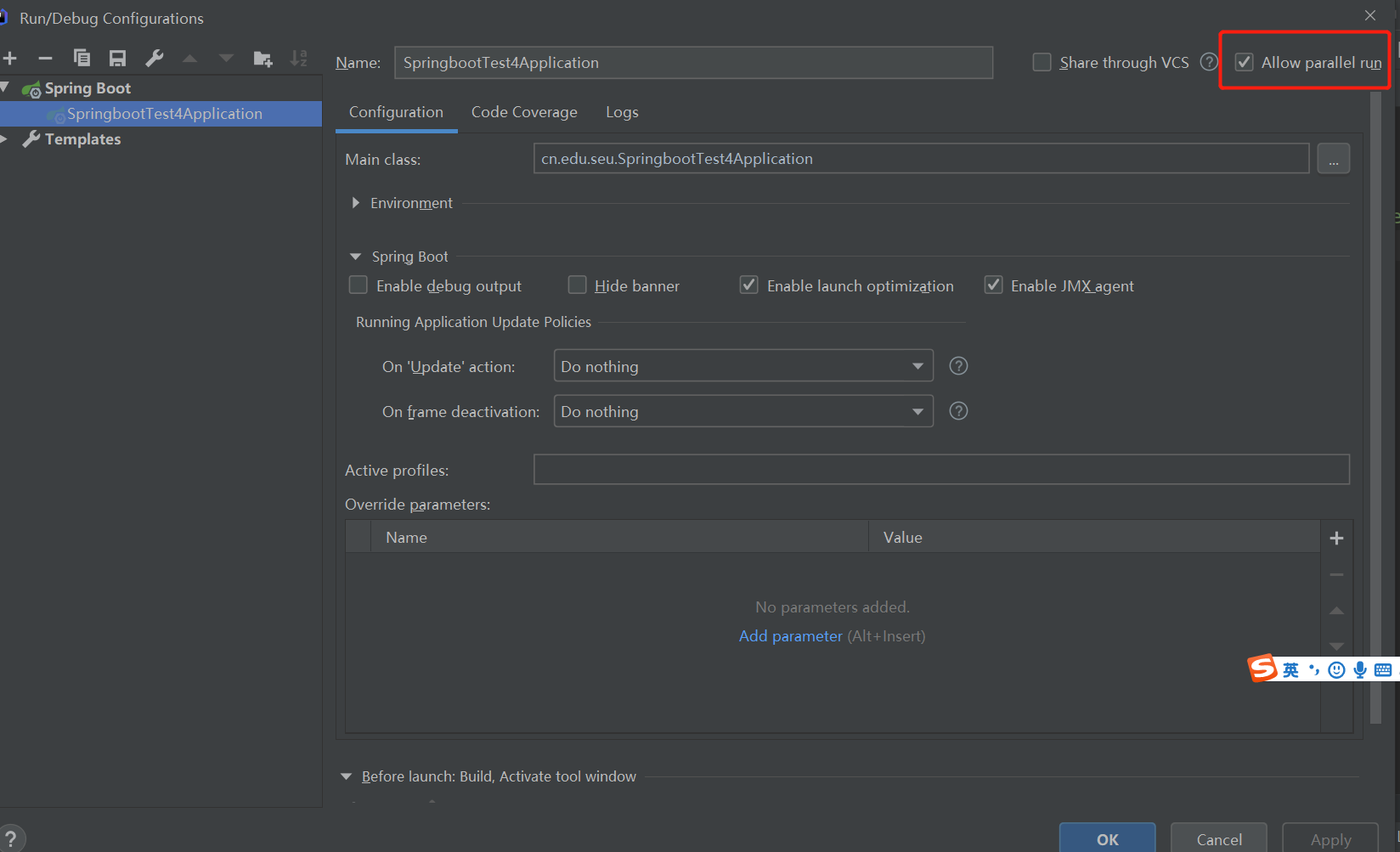


图 5.3 点击Allow prarallel run

（3）通过修改application文件的server.port的端口，启动。多个实例，需要多个端口，分别启动。

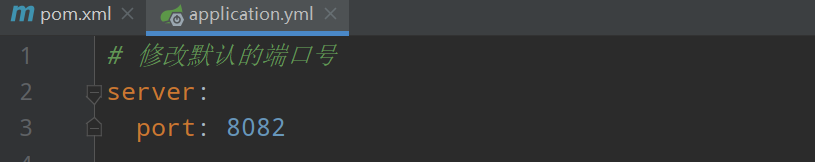


图 5.4 修改端口号

（4）分别启动三个不同的Web服务器（分别为8081、8082、8083）



图 5.5 启动服务器

# 六、使用Nginx实现负载均衡配置，并测试负载是否生效

6.1 在nginx的配置文件中进行负载均衡的配置

（1）修改nginx配置文件

输入命令：**vim /usr/local/nginx/conf/nginx.conf**

进入到nginx.conf配置文件后，按**i键**进入编辑模式；

在http模块中加入：

**upstream myserver{**

**server 192.168.32.1:8081;**

**server 192.168.32.1:8082;**

**server 192.168.32.1:8083;**

**}**

其中ip地址为windows中的本机ip地址

然后在server的location中加：

**proxy\_pass http://myserver;**



图 6.1 修改配置文件

添加完成后，按**ESC**键退出，然后输入**:wq** 即可保存成功。

6.2 添加负载均衡提示信息

在SpringBoot项目的Controller方法中，输出当前请求的时间，以及该项目的端口号：

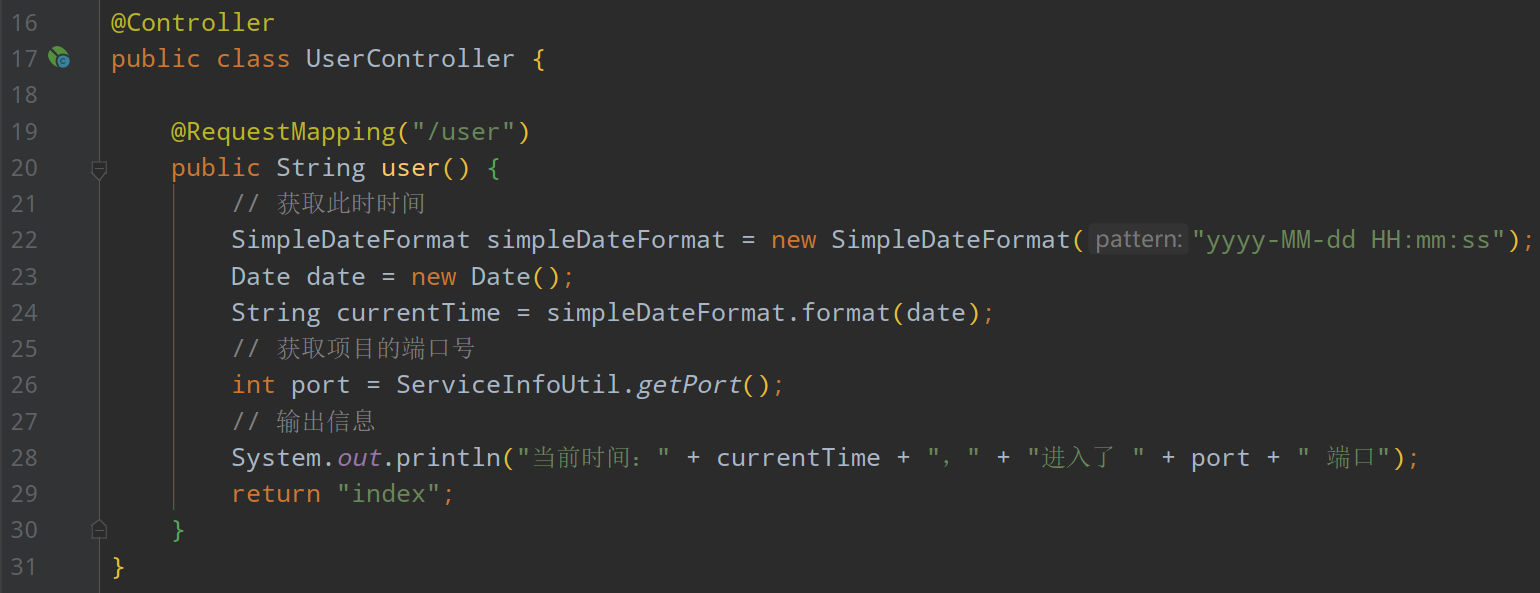


图 6.2 添加提示信息

6.3 测试负载均衡

（1）查看Linux的ip地址：

输入命令：**ifconfig -a**

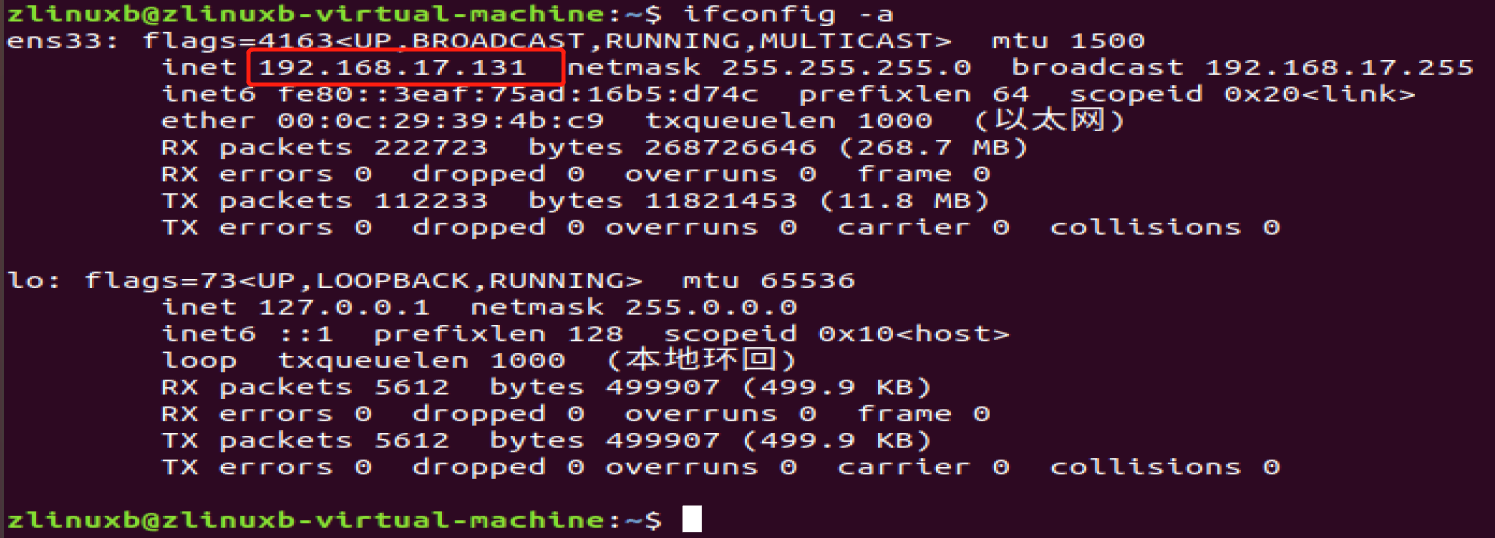


图 6.3 查看Linux的ip地址

（2）在windows的浏览器中输入网址：**http://192.168.17.131/user**

注意：默认是80端口，可不写。



图 6.4 访问成功

（3）多刷新几次当前页面，并查看SpringBoot项目的控制台的输出信息

可以发现：当向nginx服务器发出了3次相同的http请求时，此时分别访问的是8081、8082、8083端口。说明nginx已对请求做了负载均衡，将用户的请求分别分发到了不同的web服务器上，降低了单台服务器的压力，大大提升系统的吞吐量、提高网络的可用性和灵活性。



图 6.5 访问8081端口



图 6.6 访问8082端口



图 6.7 访问8083端口