第 5 章

CHAPTER 5

云服务器的配置和使用

5.1 云服务器简介

币安、欧易等交易所的私有 API 都需要绑定 IP 地址才可使用，绑定方法在 3.3 节有介绍，家庭宽带的 IP 地址是经常变化的，使用代理地址也不很稳定，最好的方式是购买有独立 IP 地址的云主机。

云主机需要选择在国外的主机，可以选择亚马逊云平台（Amazon AWS EC2）、谷歌云平台（Google Cloud Platform）、微软的 Azure 主机、阿里云国际等，这些云平台在全球都有服务器部署。如果要使用币安平台，则不要选择美国 IP 的服务器，因为美国政府禁止币安在美开展业务。尽量选择新加坡 IP、日本 IP 的主机。

有了云服务器后，需要指定云服务器的操作系统，建议选择最新版的 Ubuntu，Ubuntu 是 Linux 的一个发行版本，此外 CentOS 也是 Linux 流行的一个发行版本，也可以选择。无论是 Ubuntu 还是 CentOS，创建主机后，系统就自带 Python 运行环境了，无须再下载并安装 Python，可以直接运行自己编写的 Python 程序。

本书的 Python 程序运行环境：服务器使用的是微软的 Azure 主机，操作系统是 Ubuntu Server 22.04 LTS，1CPU，1GB 内存，Python 版本 3.10.12。

5.2 亚马逊 AWS EC2 主机申请

阿里云主机是先付费后使用，Google Cloud 可以免费试用 3 个月、微软的 Azure 主机有一年试用期、亚马逊的 AWS EC2 主机新用户可以免费试用一年，下面我们讲解如何申请和配置 AWS EC2 主机。

（1）登录亚马逊 AWS 官方网站，进行注册，注册时需要邮箱和信用卡，注册成功后会先扣除 1 美元来验证信用卡，几天后再退回。

（2）注册后选择不同等级的服务，例如选择 Free 计划，如图 5-1 所示。

（3）AWS 主机遍布全球，主机位置建议选择新加坡或东京，如图 5-2 所示。

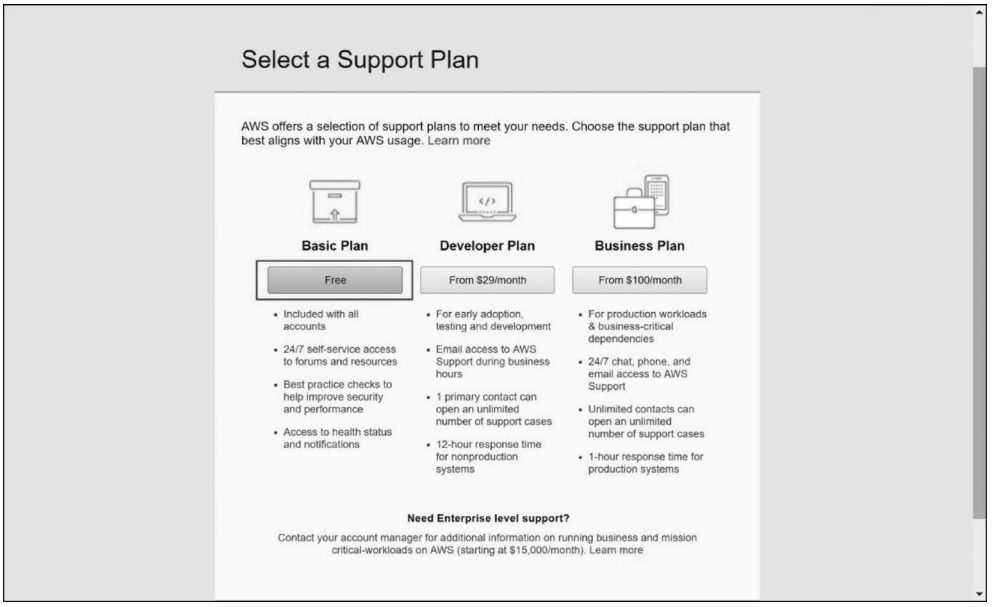

图 5-1　选择 AWS 服务等级

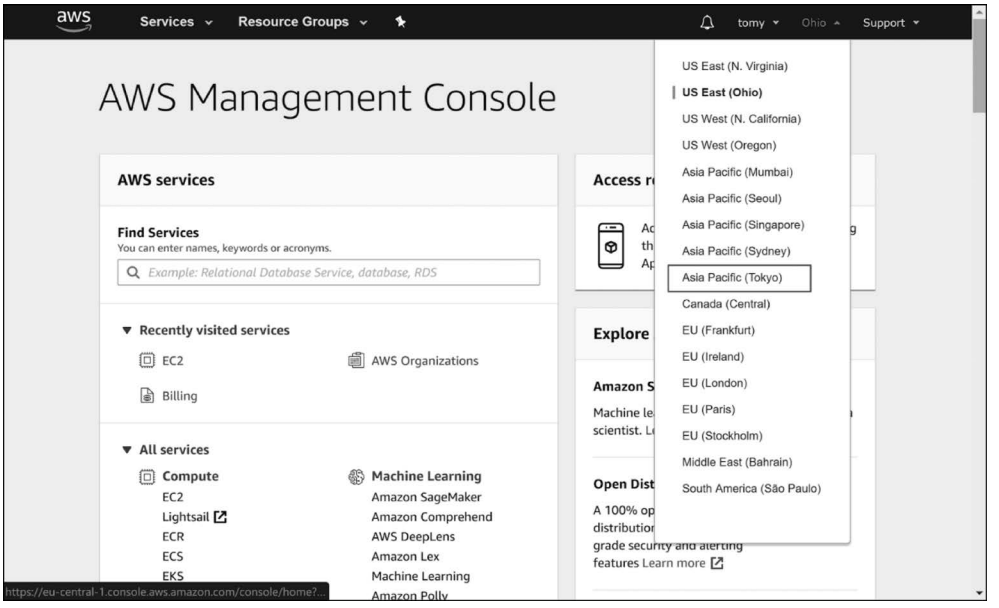

图 5-2　选择 AWS 主机的区域

(4) 选择 EC2 服务,创建云主机,如图 5-3 所示。

(5) 进入 EC2 主控制台,单击 Launch Instance 按钮,启动实例,AWS 中的实例就是云主机的意思,如图 5-4 所示。

(6) 选择主机的操作系统,建议大家选择 Ubuntu 系统,版本选择 22 或 20,如图 5-5

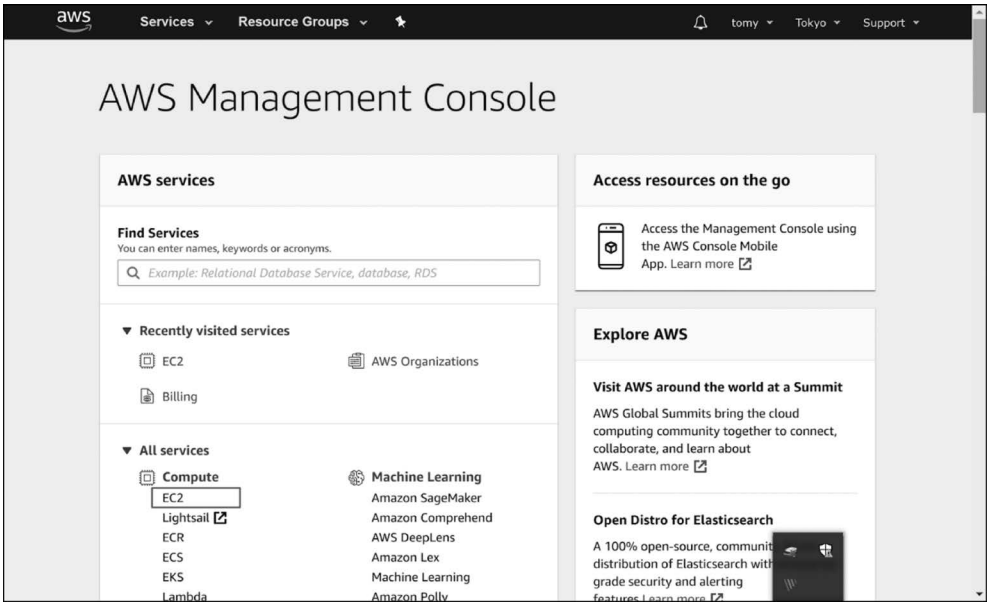

图 5-3　选择 EC2 服务

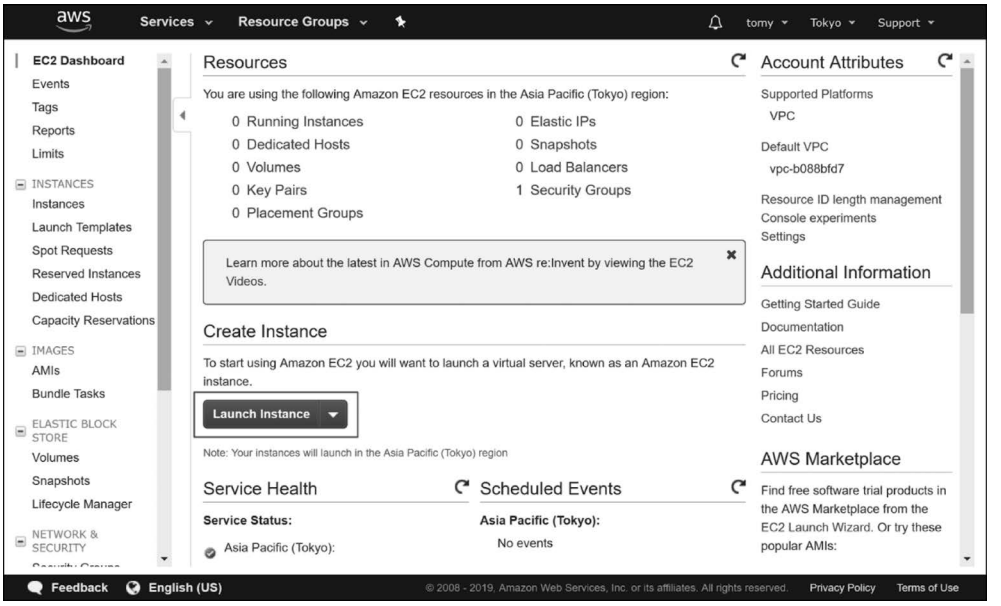

图 5-4　启动实例

所示。

(7) 选择主机的规格,带有 Free tier eligible 绿色标签提醒的是免费的规格,其他规格是要收费的,如图 5-6 所示。

(8) 下面需要设置安全组,也就是设置主机的防火墙,确定主机对外开放的服务和端口

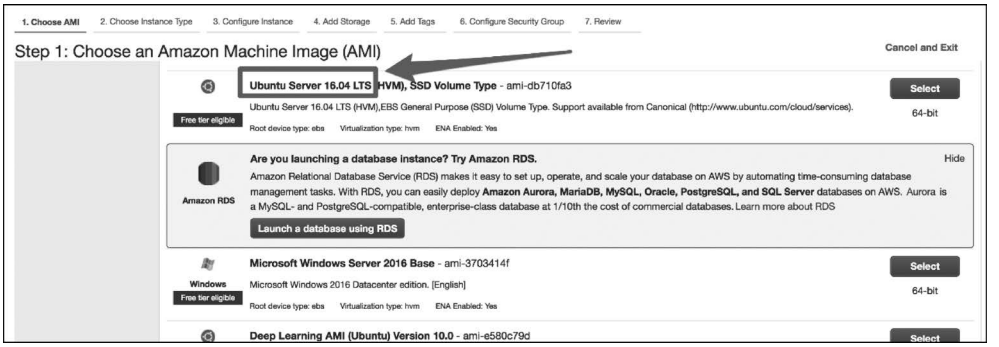

图 5-5　选择主机的操作系统

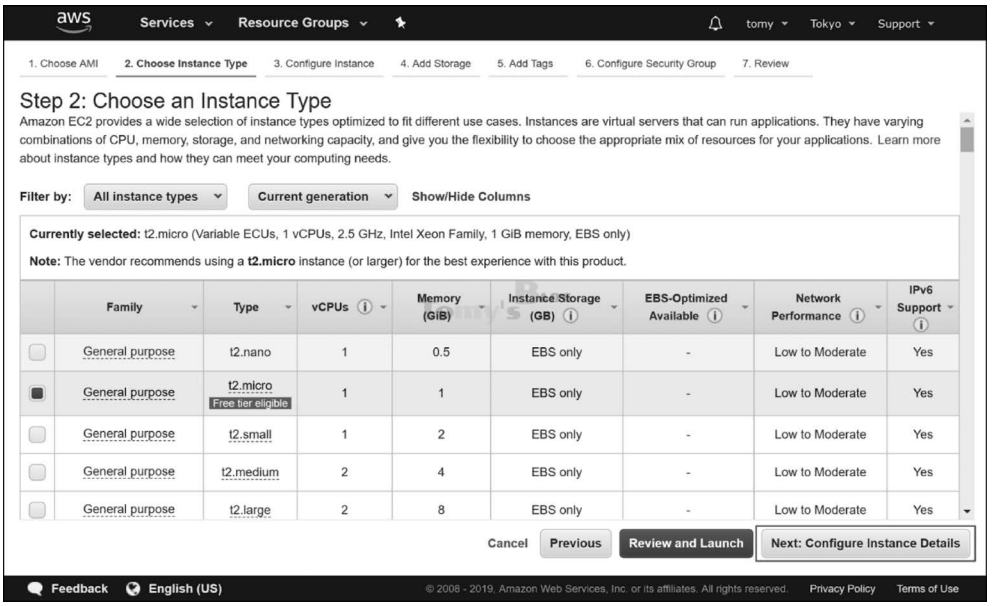

图 5-6　选择主机的规格

号，安全组中的端口会对外开放，没有出现在安全组的端口则外界无法访问。

如果主机需要用到网站 Web 服务，则需要开放 80 和 443 端口，本书讲到的编写交易程序完全不用 Web 服务，只需开放一个 SSH 登录服务器的 22 号端口，如图 5-7 所示。

(9) 至此云主机的配置完成，看一下主机的详情信息，记录主机的 IP 地址。把这个地址写到交易所 API 设置页面中的 IP 白名单中(见 2.3 节)，如图 5-8 所示。

(10) 登录主机使用 SSH 方式，需要创建一个密钥对，务必保存好这个密钥对文件，一台主机只能创建一次，如图 5-9 所示。

(11) 生成的密钥对文件是一个以 pem 为扩展名的文件，把这个文件复制到计算机 C 盘的用户目录，在其他位置用密钥对登录会报权限错误，首先打开 C 盘，找到“用户”文件夹，如图 5-10 所示。

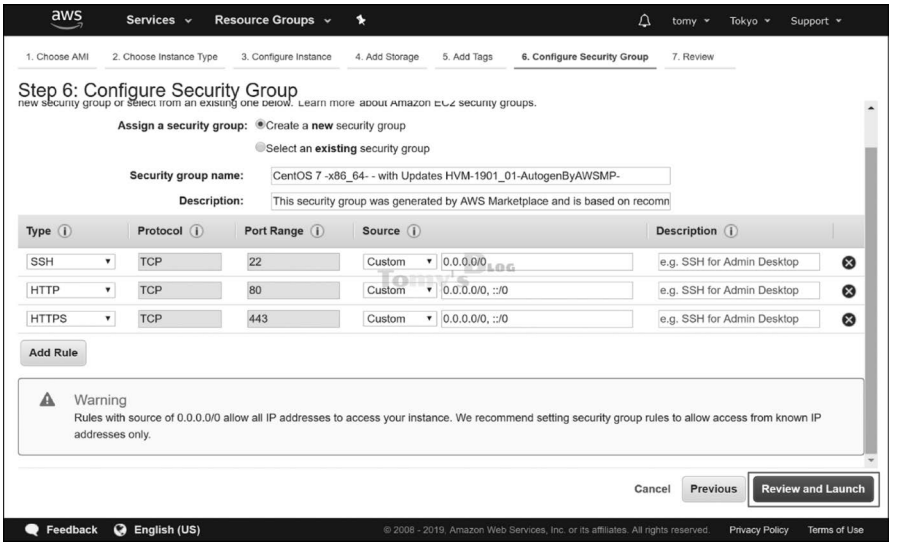

图 5-7　主机的防火墙设置

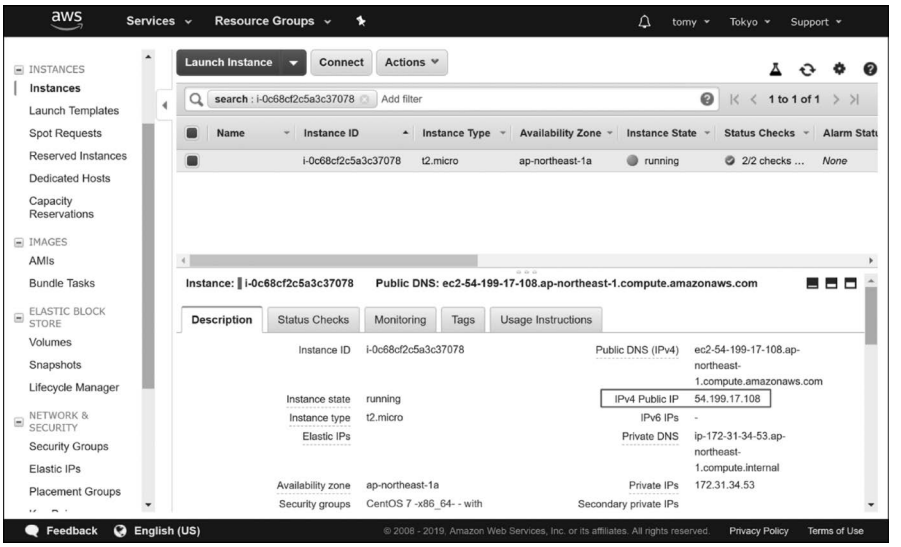

图 5-8　查看主机的 IP 地址

(12) 找到计算机中的用户名目录,如图 5-11 所示。

(13) 把密钥文件复制到这个目录,如图 5-12 所示。

(14) 打开命令行窗口,单击 Windows 窗口最下面的任务栏中的搜索图标,输入 cmd,然后选择“命令行提示符系统”,如图 5-13 所示。

(15) SSH 登录云主机的命令格式：ssh -i 密钥对文件名 ubuntu@IP 地址,如图 5-14 所示。

(16) SSH 登录成功界面如图 5-15 所示。

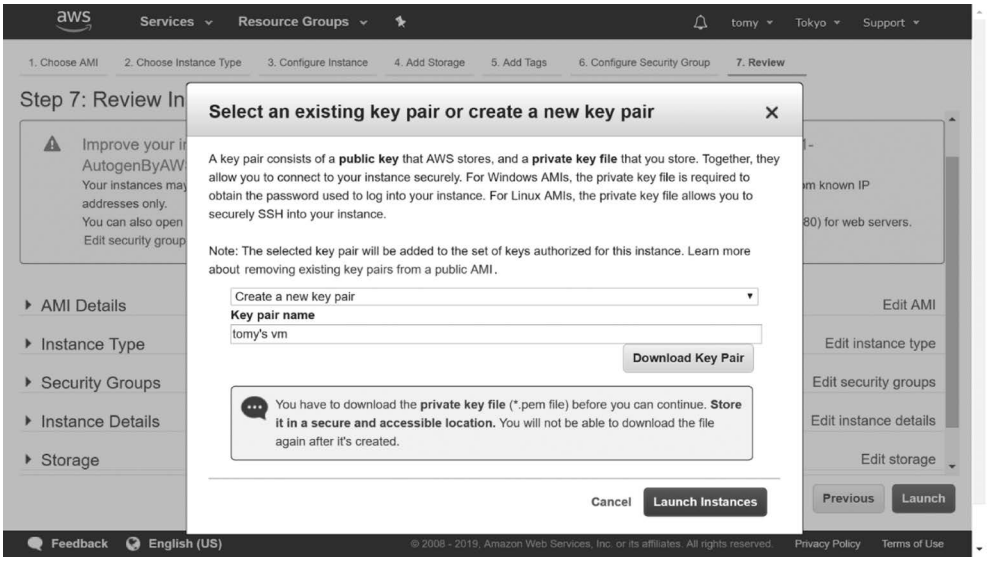

图 5-9　创建登录主机用的密钥对

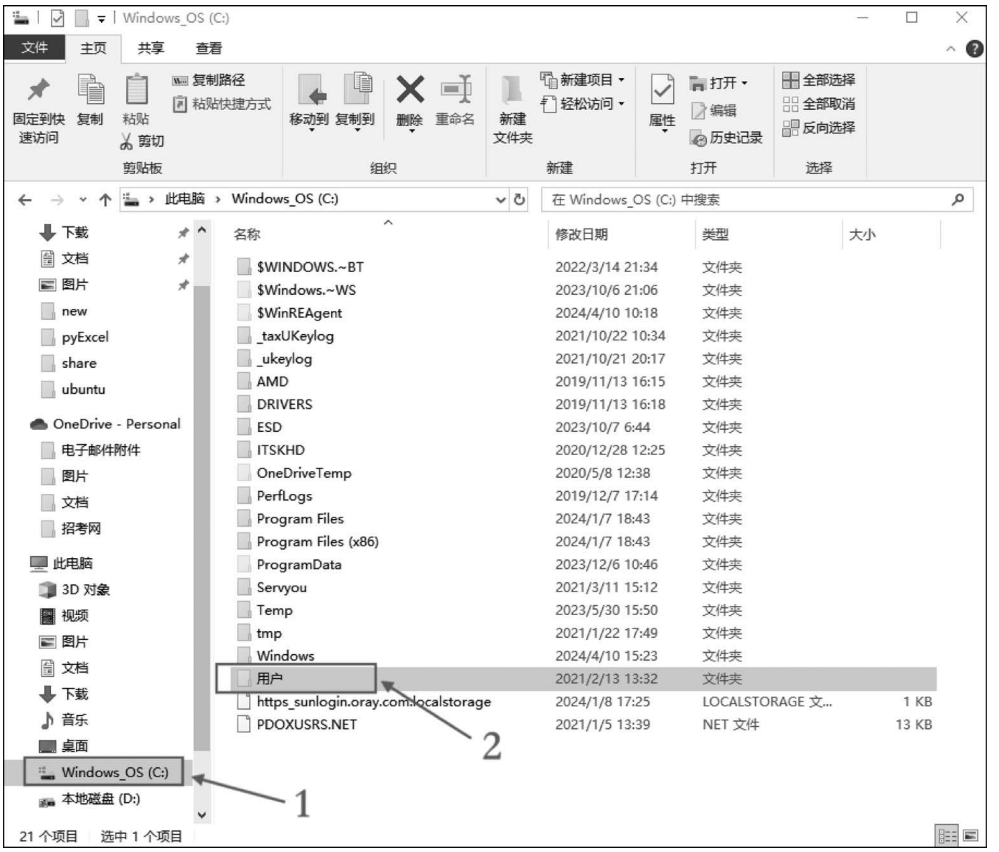

图 5-10　找到 C 盘用户目录

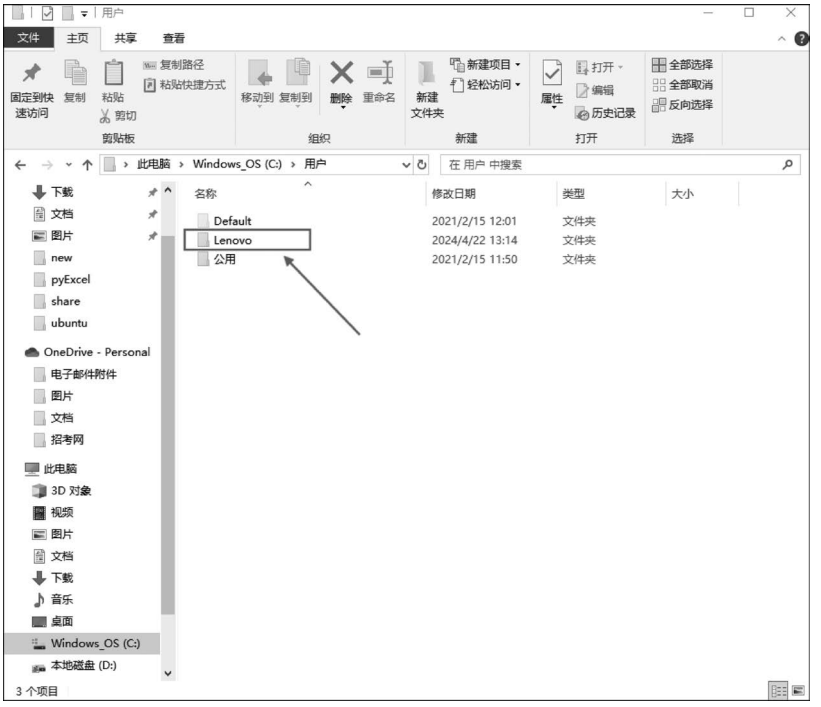

图 5-11　找到计算机中的用户名目录

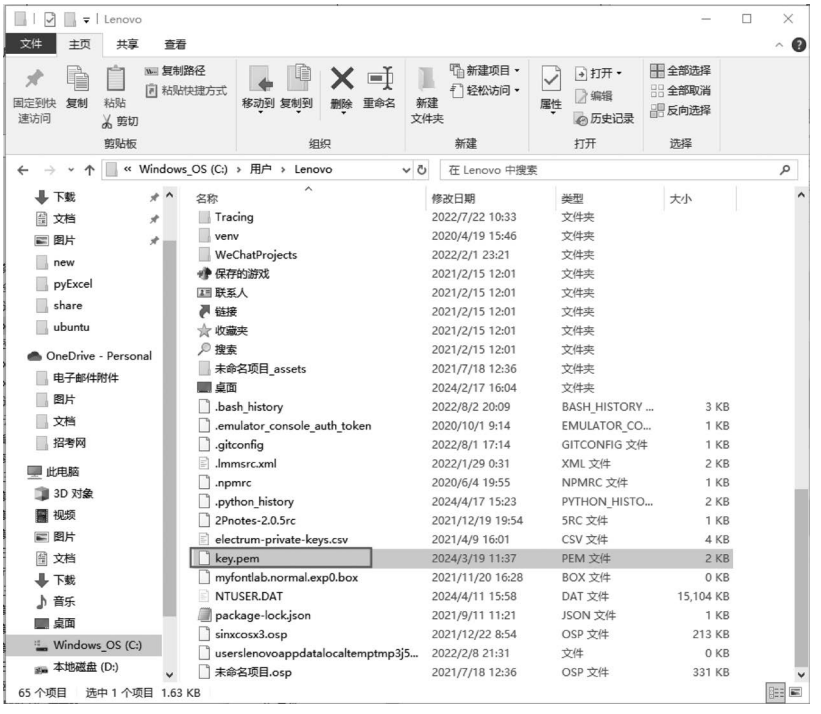

图 5-12　把密钥文件复制到这个目录

图 5-13　打开命令行窗口

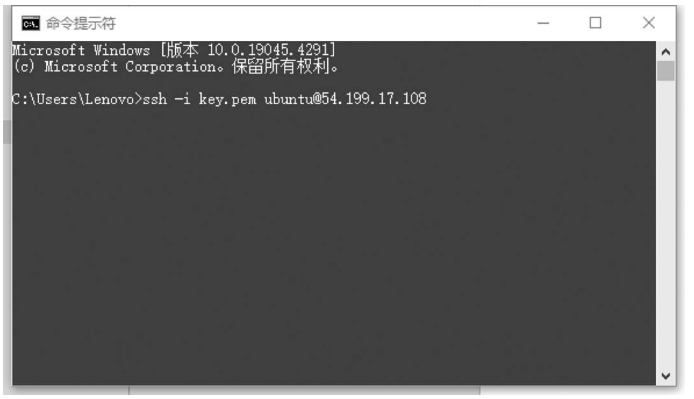

图 5-14　SSH 登录云主机

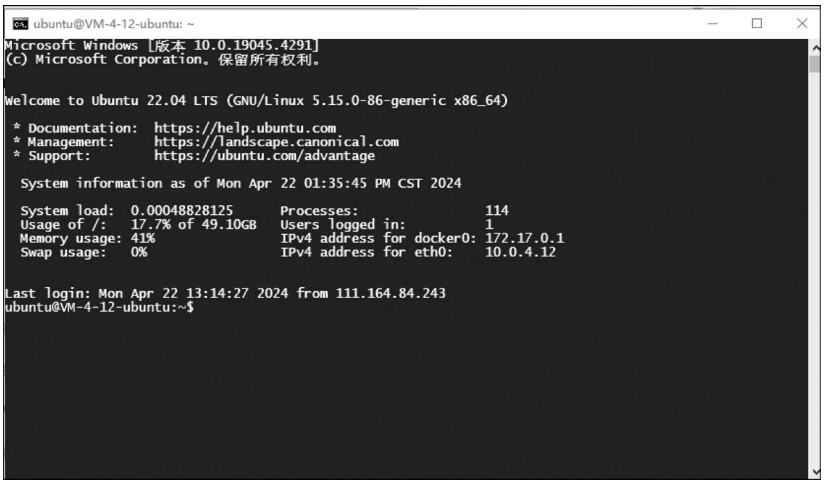

图 5-15　SSH 登录云主机成功界面

5.3　Linux 系统简介

Linux 是一个开源的操作系统，Linux 遵循 GNU 通用公共许可证(GPL)，任何个人和机构都可以自由地使用 Linux 的所有底层源代码，也可以自由地修改和再发布。

5.4　Linux 系统目录结构

不同于 Windows 系统把磁盘分为 C 盘和 D 盘的模式，Linux 的文件就是一个完整的树形结构，如图 5-16 所示。

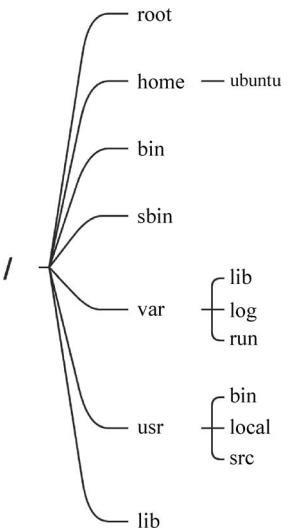

图 5-16　Linux 文件目录结构

Linux 的目录系统非常庞大，层次很深，这里只介绍一些最常用的目录，见表 5-1。

表 5-1　Linux 目录介绍

目 录 名 称	描　　述
/	根目录
/root	系统管理员主目录
/home	普通用户主目录都在这个目录下面
/home/ubuntu	Ubuntu 用户的主目录
/bin	存放系统命令程序
/sbin	存放系统管理员使用的命令
/var	存放经常修改的数据，例如程序运行的日志文件
/usr	存放用户应用程序，类似于 Windows 系统的 Program Files 目录
/lib	存放基本代码库，例如 C++库文件

下面介绍几个目录操作指令，登录 Ubuntu 系统后，当前位置是/home/ubuntu。显示当前目录，首先输入 cmd，然后按键盘的 Enter 键，指令如下：

```
cmd
```

运行结果如下：

```
/home/ubuntu
```

回到上一级目录，首先输入 cd . .，然后按键盘的 Enter 键，指令如下：

```
cd ..
```

当前目录就可变为/home。

进入下一级目录，首先输入 cd 目录名，然后按键盘的 Enter 键，指令如下：

```
cd ubuntu
```

当前目录就变为/home/ubuntu。

上面两个命令，使用的都是基于当前位置的相对路径，如果使用绝对路径(绝对路径就是包含了从根目录开始的完整路径)，则可以进入任意目录，现在当前路径是/home/ubuntu，我们想进入/var/log 目录，输入 cd 带有根目录的完整目录名，然后按键盘的 Enter 键，指令如下：

```
cd /var/log
```

当前目录就变为/var/log。

5.5 Linux 常用操作指令

Linux 系统没有 Windows 系统那样的图形界面，所有操作都需要通过在命令行输入指令来完成，指令非常多，用法也比较复杂，本节仅介绍与创建交易程序、运行交易程序相关的指令。

5.5.1 创建目录指令

创建目录的指令为 mkdir，用这个指令创建一个 book 目录，指令如下：

```
mkdir book
```

5.5.2 改变目录指令

进入下一级的 book 目录，指令如下：

```
cd book
```

返回上一级目录，指令如下：

```
cd ..
```

跳到任意目录，需要使用绝对路径，绝对路径就是从根目录起始的一个完整的路径，例如要进入/home/ubuntu这个目录，无论当前处于哪个目录内都可以使用绝对径路来跳转目录，指令如下：

```
cd /home/ubuntu
```

5.5.3　显示目录中包含的文件和子目录的指令

显示当前目录下的文件及目录内容，首先输入ls，然后按Enter键，当ls后面没有参数时，只显示文件名或目录名的简要信息，指令如下：

```
ls
```

如果需要以列表形式显示当前目录中的文件和目录细节信息，则需要在ls后加上参数-l，指令如下：

```
ls -l
```

运行结果如图5-17所示。

```
ubuntu@VM-4-12-ubuntu: ~/book
ubuntu@VM-4-12-ubuntu:~/book$ ls
binanceFutureGetOrd.py  binanceFuturesKline.py  binanceSpotKline.py  okDepth.py  okMargin.py
binanceFutureMacd.py    binanceSpotDepth.py     okBalance.py         okKline.py  okTicker.py
ubuntu@VM-4-12-ubuntu:~/book$ ls -l
total 44
-rw-rw-r-- 1 ubuntu ubuntu  434 Apr 23 09:47 binanceFutureGetOrd.py
-rw-rw-r-- 1 ubuntu ubuntu 6335 Apr 23 09:46 binanceFutureMacd.py
-rw-rw-r-- 1 ubuntu ubuntu  589 Apr 23 09:57 binanceFuturesKline.py
-rw-rw-r-- 1 ubuntu ubuntu  586 Apr 23 09:58 binanceSpotDepth.py
-rw-rw-r-- 1 ubuntu ubuntu  436 Apr 23 10:02 binanceSpotKline.py
-rw-rw-r-- 1 ubuntu ubuntu  585 Apr 23 09:39 okBalance.py
-rw-rw-r-- 1 ubuntu ubuntu  202 Apr 23 09:59 okDepth.py
-rw-rw-r-- 1 ubuntu ubuntu   65 Apr 23 10:00 okKline.py
-rw-rw-r-- 1 ubuntu ubuntu   87 Apr 23 10:01 okMargin.py
-rw-rw-r-- 1 ubuntu ubuntu 1263 Apr 23 09:44 okTicker.py
ubuntu@VM-4-12-ubuntu:~/book$
```

图5-17　显示目录内容

5.5.4　创建Python程序文件指令

Linux系统自带一个编辑工具，叫作vim，可以用来创建和编辑文本文件及代码文件，早期的Linux内置的编辑工具是vi，vim是vi的升级版，下面我们在云服务器上创建一段设置欧易杠杆倍数的代码，并在服务器端运行。按快捷键Ctrl+C复制源代码，代码如下：

```
from okx import Account
```

```
apiKey = "2a076334 - 82ca - 44a8 - 9971 - fbf556862d44"
apiSecretKey = "EE51E9F072DE8B6DB7A41F4EF5E3CFB5"
passphrase = "Hello2020!"
#flag:0 实盘;flag:1 模拟盘
accountAPI = Account.AccountAPI(apiKey, apiSecretKey, passphrase, False, flag = "1")
#instId:交易对;lever:杠杆倍数; mgnMode:逐仓模式
result = accountAPI.set_leverage(instId = "BTC - USDT", lever = "5", mgnMode = "isolated")
print(result)
```

在服务器端的命令行窗口，输入 vim 指令创建这个 Python 程序文件，Python 程序的扩展名都是.py，指令如下：

```
vim testOkLeverage.py
```

运行结果如图 5-18 所示，现在就创建了一个代码编辑界面，在这个界面里录入代码。

图 5-18　vim 编辑界面

vim 有 3 种模式：命令模式、输入模式和底线命令模式。输入 vim 指令后看到的上面界面就处于命令模式，命令模式下无法输入字符。

注意：此时计算机端的输入法要保持英文输入法状态，然后按键盘的 I 键，I 就是 vim 的输入指令，此时编辑区最下面一行出现了 INSERT，此时就可以录入字符了，如图 5-19 所示。

然后右击便可粘贴代码，粘贴代码后 Python 程序源代码就出现在 vim 编辑区了，如图 5-20 所示。

代码粘贴完成后，按键盘的 Esc 键，令 vim 切换回命令模式，然后按键盘的组合键 Shift+：进入底线命令行模式，然后在：后面输入 wq 指令，保存并退出，w 是 write 的首字母，保存的意思，q 是 quit 的首字母，退出的意思，如图 5-21 所示。

图 5-19　vim 输入模式界面

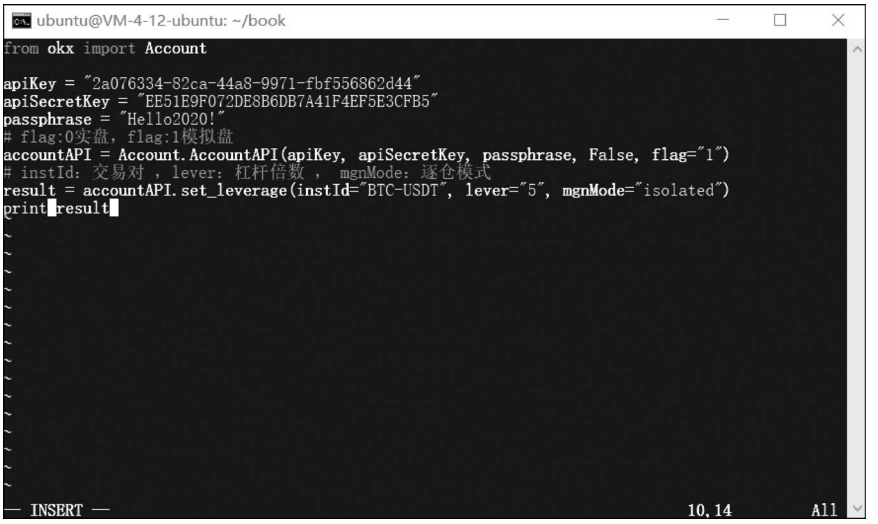

图 5-20　vim 粘贴代码界面

5.5.5　运行 Python 程序文件指令

Ubuntu 系统下的 Python 程序名是 Python3，而 CentOS 下是 Python，运行 5.5.4 节创建的代码文件 testOkLeverage.py，程序运行结果会显示在服务器命令行窗口中，注意：python3 和要运行的程序文件名之间要有一个空格，输入完成后按 Enter 键，即可运行这个 Python 程序，指令如下：

```
python3 testOkLeverage.py
```

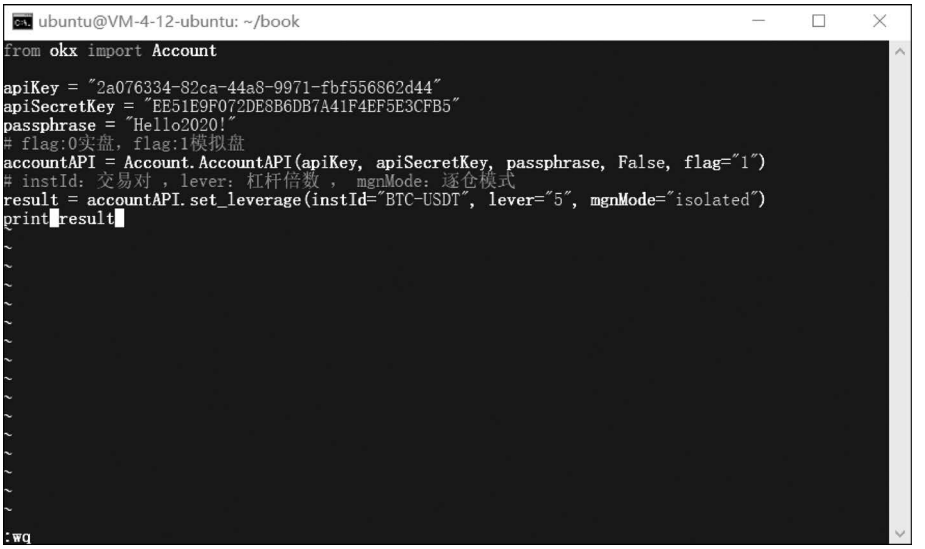

图 5-21 vim 底线命令行模式界面

5.5.6 程序运行结果保存到日志文件指令

如果需要把程序运行结果保存到日志文件中,则可以在运行程序指令的后面加上一个重定向符>,这样原本在窗口内输出的内容就会重定向到日志文件中,指令如下:

```
python3 testOkLeverage.py > 日志.log
```

5.5.7 中止程序运行

一般的程序,运行结束后就自动退出了,但有的程序是无限循环运行的,例如接收交易所行情推送指令的程序,中止程序运行可以关闭命令行窗口,也可以按键盘的快捷键 Ctrl+C。

5.5.8 程序后台运行指令

如果需要一个程序 24h 不间断地运行,并且关闭 SSH 登录窗口也不会中断程序的运行,就需要用后台运行指令 nohup 配合 & 符号,指令如下:

```
nohup python3 getOkKline.py > kline.log 2 > &1 &
```

5.5.9 查看后台运行程序的指令

查看后台程序的指令为 ps,Linux 系统的后台程序非常多,要在众多程序信息中筛选出需要看到的程序名,需要结合 grep 过滤指令,中间的|符号是管道符,功能是把左侧指令的运行结果作为右侧指令的输入,指令如下:

```
ps ax | grep getOk
```

运行结果如图 5-22 所示。

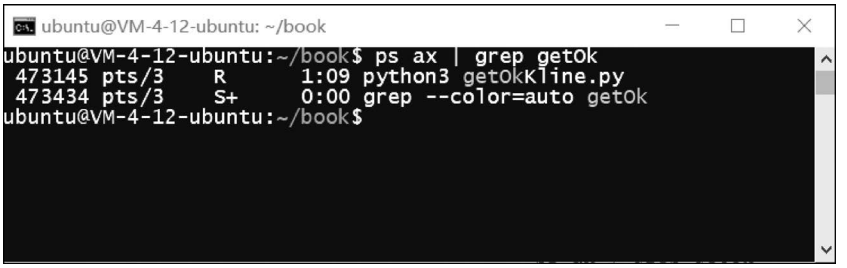

图 5-22　查看后台运行程序的界面

查询结果的第 1 行就是我们启动的 getOkline.py 程序的信息，473145 就是进程号。

5.5.10　关闭后台运行程序的指令

关闭一个后台程序的指令是 kill，后面加上该程序对应的进程号参数就可以关闭这个程序，指令如下：

```
kill 473145
```

5.5.11　删除文件或目录的指令

删除文件和目录都使用 rm 指令，删除文件的指令如下：

```
rm 文件名
#例子
rm okTest.py
```

删除目录是很危险的指令，使用时一定要小心，删除目录的指令如下：

```
rm -r 目录名
#例子
rm -r books
```

5.5.12　移动文件或目录的指令

移动文件（如果在同一个目录下移动文件，则实际效果就是给文件改名）和目录都使用 rm 指令，指令如下：

```
mv 旧文件名 新文件名
#改名例子
mv okTest.py okTest2.py          #将文件改名为 okTest2.py
```

```
#移动位置的例子,把 okTest2.py 文件移动到上一级目录
mv okTest2.py ..
```

5.5.13 查看文本文件内容指令

vim 是编辑文本文件内容的工具,而如果只想查看文本文件内容,不进行编辑修改,则可以使用 cat 指令,指令格式如下:

```
cat 文件名
#例子
cat telegramBot.log
```

5.5.14 查看文本文件头部内容指令

查看文本文件头部内容的指令如下:

```
head 文件名
#例子 1,默认显示前 10 行
head telegramBot.log
#例子 2,显示前 20 行
head -n 20 telegramBot.log
```

5.5.15 查看文本文件尾部内容指令

查看文本文件尾部内容的指令如下:

```
tail 文件名
#例子 1,默认显示最后 10 行
tail telegramBot.log
#例子 2,显示最后 20 行
tail -n 20 telegramBot.log
```

5.6 Git 指令介绍

在 Linux 系统下使用 vim 编辑代码,要记住大量的操作指令和快捷键,操作非常不方便。更好的方式是在我们的计算机上用集成开发工具 VS Code 编写好代码,然后利用 Git 指令把代码同步(push)到 Gitee 仓库,我们的云服务器端再从 Gitee 仓库拉取最新代码(pull),本节介绍 Git 的基本使用方法。操作流程如图 5-23 所示。

5.6.1 计算机端安装 Git

打开 Git 官网,下载计算机操作系统对应的版本,下载界面如图 5-24 所示。

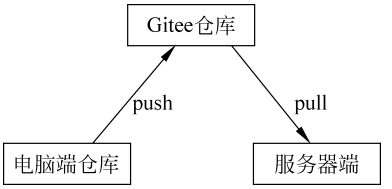

图 5-23　代码同步的操作流程

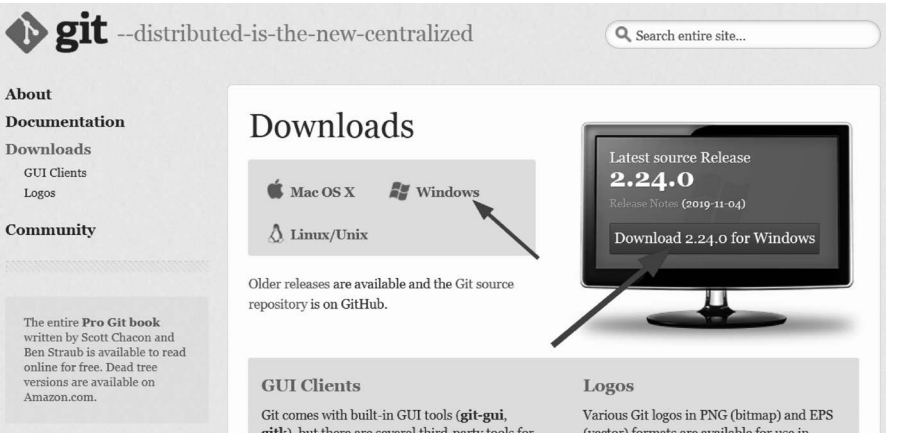

图 5-24　Git 官网下载界面

下载并安装成功后，打开命令行窗口，输入的指令如下：

```
git
```

如果看到如图 5-25 所示的界面，则说明安装 Git 成功。

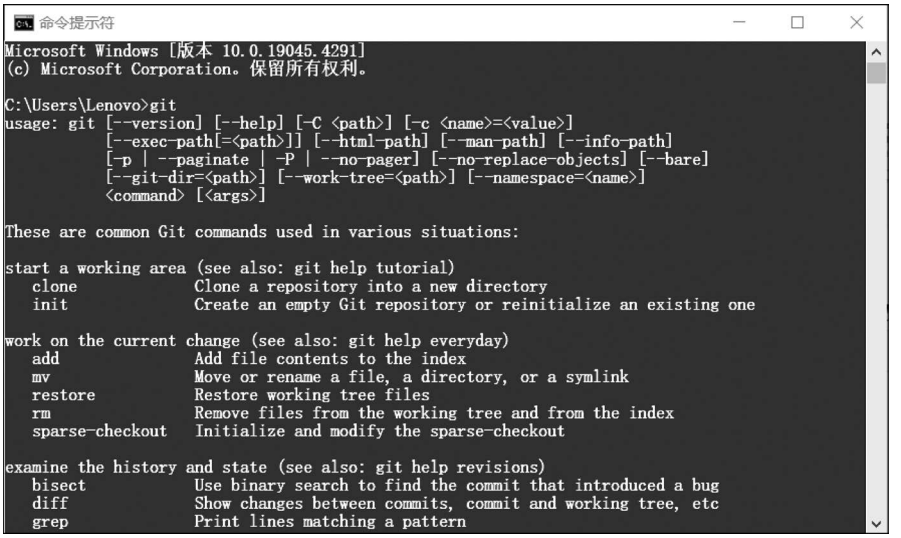

图 5-25　Git 运行界面

5.6.2 服务器端安装 Git

Ubuntu 下包管理器程序是 apt，安装软件的指令是 apt install 软件名，安装 Git 需要使用最高管理员(root)权限，所以在指令前要加 sudo，安装 Git 的指令如下：

```
sudo apt update
sudo apt install git
```

检查是否安装成功，输入 git --version 命令，运行结果如图 5-26 所示。

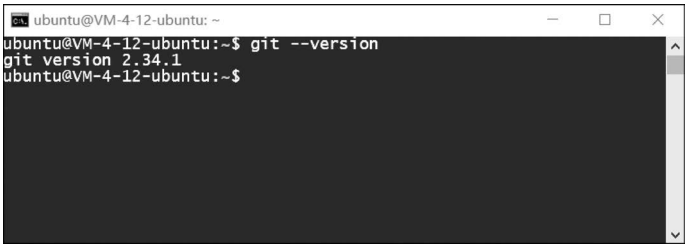

图 5-26 服务器端 Git 运行界面

5.6.3 注册 Gitee 账号并创建仓库

Gitee 是国内最大的软件代码托管平台，个人使用是完全免费的，注册好账号后，新建一个仓库，输入仓库名称，选择仓库类型，例如选择私有，分支模型选择单分支，最后单击“创建”按钮创建仓库，如图 5-27 所示。

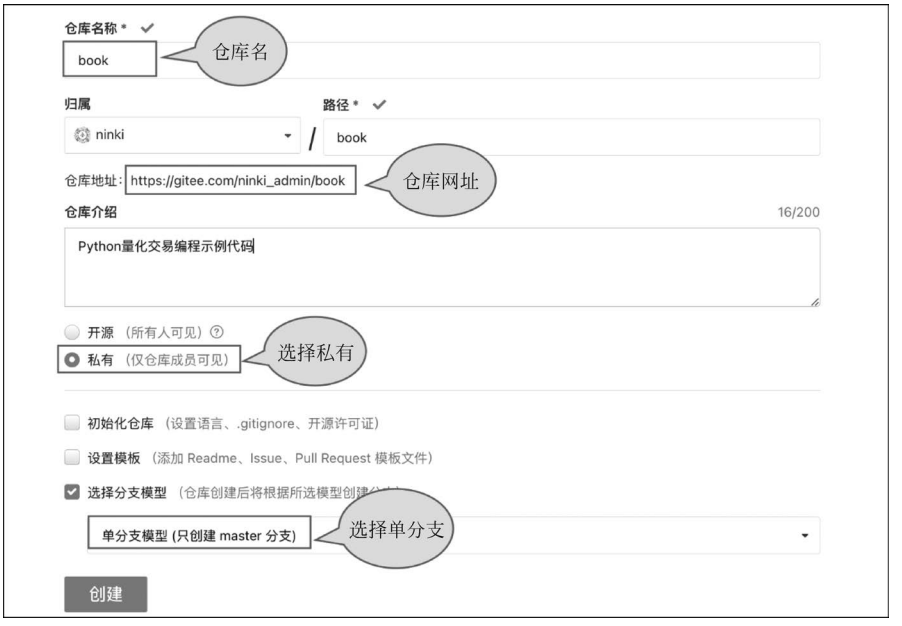

图 5-27 Gitee 创建仓库界面

5.6.4　计算机端创建仓库

首先进入代码所在文件夹，打开命令行窗口，输入以下 Git 的账号、邮箱等全局配置指令，指令如下：

```
git config --global user.name "ninki51"
git config --global user.email "2556792125@qq.com"
```

创建仓库的指令如下：

```
git init
git add .
git commit -m "第 1 次提交"
git remote add origin https://gitee.com/ninki_admin/book.git
git push -u origin master
```

5.6.5　服务器端拉取仓库代码

首次拉取代码使用 git clone 指令会在服务器建立一个以仓库名命名的目录，并把代码都下载到这个目录中，指令如下：

```
git clone https://gitee.com/ninki_admin/book.git
```

以后我们修改计算机端的本地仓库代码，并提交(push)到 Gitee 仓库后，服务器端可以使用 git pull 指令来同步最新的有变化的代码，指令如下：

```
git pull
```