# 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。

| Select a Suppor   | rt Plan   |                                 |  |
|---|---|---------------------------------|--|
| <text><image/><section-header><section-header></section-header></section-header></text> | And plans to meet your needs<br>be Learn more<br>Developer Plan<br>For \$29month<br>For \$29month<br>For \$29month<br>For \$29month<br>For \$29month<br>For \$29month<br>For \$29month<br>For \$29month<br>For \$29month<br>For \$29month<br>Support during business<br>hours<br>I primary contact can<br>open an unimited<br>number of support cases<br>I primary contact can<br>open an unimited<br>number of support cases<br>I systems<br>Plans the support for additional information on n<br>on AWS (starting at \$15,000/more) | e. Choose the support plan that |  |
|   |   |                                 |  |

图 5-1 选择 AWS 服务等级

| AWS Services - Resource Grou   | ps v t⊧   |  | 🗘 tomy 🕶 Ohio 🔺   | Support 👻 |
|--|---|--|---|-----------|
| AWS Manager  | ment Console  |  | US East (N. Virginia)<br>US East (Ohio)<br>US West (N. California)<br>US West (Oregon)                              |           |
| AWS services   |   | Access re  | Asia Pacific (Mumbai)<br>Asia Pacific (Seoul)   |           |
| Find Services<br>You can enter names, keywords or acronyms.<br>Q. Example: Relational Database Serv  | ice, database, RDS  | P Ac th Ar   | Asia Pacific (Singapore)<br>Asia Pacific (Sydney)<br>Asia Pacific (Tokyo)<br>Canada (Central)                       | g         |
| <ul> <li>▼ Recently visited services</li> <li>□ EC2</li> <li>Billing</li> </ul>  | ill AWS Organizations   | Explore<br>Amazon S  | EU (Frankfurt)<br>EU (Ireland)<br>EU (London)   |           |
| ✓ All services     ☐ Compute     EC2     Lightsail [2]     ECR     ECS     ECS     EKS     Https://eu-central=1.console@vanazon.com/console/home?. | Machine Learning<br>Amazon SageMaker<br>Amazon Comprehend<br>AWS DeepLens<br>Amazon Lex<br>Machine Learning<br>Amazon Polly | Machine le<br>scientist. Le<br>Open Dist<br>A 100% op<br>distributior<br>grade secun<br>features Lea | EU (Fans)<br>EU (Stockholm)<br>Middle East (Bahrain)<br>South America (São Paulo)<br>iny and atercing<br>inn more 🛃 |           |

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

| aws Services → Resource Groups → 🛧   | 众 tomy ▼ Tokyo ▼ Support ▼   |
|--|--|
| AWS Management Console   | 2  |
| AWS services   | Access resources on the go   |
| Find Services<br>You can enter names, keywords or acronyms.  | Access the Management Console using the AWS Console Mobile   |
| Q Example: Relational Database Service, database, RDS  | App. Learn more 🖸  |
| ▼ Recently visited services  | Explore AWS  |
| EC2     AWS Organizations     Billing  | Visit AWS around the world at a Summit<br>AWS Global Summits bring the cloud   |
| ▼ All services      Compute      Machine Learning  | computing community together to connect,<br>collaborate, and learn about<br>AWS. Learn more 🛃  |
| ECZ     Amazon SageMaker       Lightsail IZ     Amazon Comprehend       ECR     AWS DeepLens       ECS     Amazon Lex       EKS     Machine Learning       Lambda     Amazon Polly | Open Distro for Elasticsearch<br>A 100% open-source, communit<br>distribution of Elasticsearch with<br>grade security and alerting<br>foblume faam more få |

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

| 1. Choose AMI 2. Choose Instance | e Type 3. Config              | gure Instance 4. Add Storage 5. Add Tegs 6. Configure Security Group 7. Review   |                                      |
|----------------------------------|-------------------------------|--|--------------------------------------|
| Step 1: Choose an A              | mazon Ma                      | achine Image (AMI)   | Cancel and Exit                      |
|                                  | 9                             | Ubuntu Server 16.04 LTS HVM), SSD Volume Type - aml-db710fa3   | Select                               |
|                                  | Free tier eligible            | Ubuntu Server 16.04 LTS (MM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (http://www.ubuntu.com/cloud/services).<br>Rot device type: eba: Vinalization type: htm: ENA Enabled: Nei   | 64-bit                               |
|                                  | Amazon RDS                    | Are you launching a database instance? Try Amazon RDS.<br>Amazon Relational Database Service (RDS) markes it easy to set up, operate, and scale your database on AWS by automating time-consuming<br>management tasks: With RDS, you can easily deloy, Amazon Aurona, Ama | Hide<br>database<br>n AWS. Aurora is |
|                                  | Ry                            | Microsoft Windows Server 2016 Base - ami-3703414f  | Select                               |
|                                  | Windows<br>Free tier eligible | Microsoft Windows 2016 Datacenter edition. [English]<br>Roct device type: eba Vritalization type: hm. DNA Enabled: Nes   | 64-bit                               |
|                                  | 0                             | Deep Learning AMI (Ubuntu) Version 10.0 - ami-e580c79d   | Select                               |

#### 图 5-5 选择主机的操作系统

| 1. Chor                            | use AMI 2. Choose Insta  | ance Type 3. Co  | onfigure Instance  | 4. Add Storage   | 5. Add Tags 6. Confi   | aure Security Group 7.                               | Review   | apport                                      |
|------------------------------------|--|--|--|--|--|--|--|---|
| Step<br>mazon<br>ombina<br>bout in | 2: Choose an<br>EC2 provides a wide sel<br>titions of CPU, memory, si<br>stance types and how the<br>r: All instance types                               | Instance T<br>lection of instance<br>torage, and netwo<br>ey can meet your c<br>Curren   | ype<br>types optimized t<br>rking capacity, an<br>computing needs. | o fit different use c<br>d give you the flex<br>Show/Hide Co | ases. Instances are virtu<br>ibility to choose the app<br>plumns   | ual servers that can run a ropriate mix of resources | applications. They have a for your applications. L   | varying<br>earn more                        |
| Curre<br>Note:                     | ntly selected: t2.micro (\<br>The vendor recommends  | Variable ECUs, 1 v<br>s using a <b>t2.micro</b>  | CPUs, 2.5 GHz,<br>instance (or large                               | Intel Xeon Family,<br>er) for the best exp                   | 1 GiB memory, EBS only<br>erience with this product  | y)<br>t.   |  | IPv6  |
|                                    | Family 👻   | Туре -   | vCPUs (i) *  | (CIP)  | Instance Storage   | EBS-Optimized  | Retwork  | Support -                                   |
|                                    | Family *<br>General purpose  | Type -   | vCPUs (i) *  | (GiB)  | EBS only   | Available ()   | Performance (i)  | Support +<br>(i)<br>Yes                     |
|                                    | Family  General purpose General purpose  | Type •<br>t2.nano<br>t2.micro  | vCPUs (i) ~<br>1<br>1  | (GiB)<br>0.5   | EBS only   | Available ()   | Performance ①<br>Low to Moderate   | Support *                                   |
|                                    | Family        General purpose        General purpose        General purpose  | Type<br>12.nano<br>12.micro<br>Free tier eligible<br>12.small  | vCPUs () -<br>1<br>1<br>1  | (GiB)<br>0.5<br>1<br>2                                       | EBS only<br>EBS only   | Available ()   | Performance () Control | Support *<br>(i)<br>Yes<br>Yes<br>Yes       |
|                                    | Family        General purpose        General purpose        General purpose        General purpose   | Type Turner of the second seco | vCPUs () ~<br>1<br>1<br>1<br>2                                     | (GiB)<br>0.5<br>1<br>2<br>4                                  | Instance Storage     (GB) ①       EBS only       EBS only       EBS only       EBS only       EBS only   | Available ()   | Network<br>Performance () *<br>Low to Moderate<br>Low to Moderate<br>Low to Moderate   | Support *<br>()<br>Yes<br>Yes<br>Yes<br>Yes |
|                                    | Family     P       General purpose     G       General purpose     G       General purpose     G       General purpose     G       General purpose     G | Type -<br>12.nano<br>12.micro<br>Free tier eligible<br>12.small<br>12.medium<br>12.large   | vCPUs () ~<br>1<br>1<br>1<br>2<br>2                                | (GiB)<br>0.5<br>1<br>2<br>4<br>8                             | Instance storage     (GB) ()       EBS only       EBS only | EBS-Optimized<br>Available ① *<br>-<br>-<br>-        | Performance ① *<br>Low to Moderate<br>Low to Moderate<br>Low to Moderate<br>Low to Moderate<br>Low to Moderate   | Support *<br>()<br>Yes<br>Yes<br>Yes<br>Yes |

#### 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。

| tep 6: C |                   |                                  |                          | re about Amazon   | EC2 security gr | roups.  |            |                    |      |
|----------|-------------------|----------------------------------|--------------------------|-------------------|-----------------|---|------------|--------------------|------|
|          | A                 | ssign a security gro             | up: Create a new se      | curity group      |                 |   |            |                    |      |
|          |                   | Security group nam<br>Descriptio | n: CentOS 7 -x86         | _64 with Updat    | ed by AWS Mark  | 01-AutogenByAWSMP-<br>ketplace and is based on reco | nn         |                    |      |
| ype 🕕    |                   | Protocol (i)                     | Port Range (j)           | Source (i)        |                 |   | Descripti  | on (i)             |      |
| SSH      | ۲                 | TCP                              | 22                       | Custom            | 0.0.0.0/0       | G   | e.g. SSH   | for Admin Desktop  | 8    |
| HTTP     | ۲                 | TCP                              | 80                       | Custom •          | 0.0.0/0, ::/0   |   | e.g. SSH   | for Admin Desktop  | 8    |
| HTTPS    | ۲                 | TCP                              | 443                      | Custom            | 0.0.0/0, ::/0   |   | e.g. SSH   | for Admin Desktop  | 8    |
| Add Rule | arning<br>es with | source of 0.0.0/0 al             | llow all IP addresses to | access your insta | nce. We recomr  | mend setting security group rul                     | es to allo | w access from know | n IP |

图 5-7 主机的防火墙设置

| aws                                  | ervices 👻 Resource Groups 👻 🕏                              | Δ  | tomy 👻 Tokyo 👻 Support 👻                   |
|--------------------------------------|--|--|--|
| INSTANCES Instances Launch Templates | Connect Actions  | •  | <b>▲ ↔ ♦ Ø</b><br>❷ K < 1to1of1 > >        |
| Spot Requests                        | Name - Instance ID - In                                    | stance Type 👻 Availability Zone 👻 Insta                  | nce State - Status Checks - Alarm Statu    |
| Reserved Instances                   | i-0c68cf2c5a3c37078 t2                                     | .micro ap-northeast-1a 🌒 rr                              | unning Ø 2/2 checks None                   |
| Dedicated Hosts                      |  |  |  |
| Capacity<br>Reservations             |  |  |  |
| IMAGES                               |  |  |  |
| AMIs                                 | 4  |  | ٢  |
| Bundle Tasks                         | Instance: i-0c68cf2c5a3c37078 Public DN                    | 5: ec2-54-199-17-108.ap-northeast-1.comp                 | ute.amazonaws.com                          |
| ELASTIC BLOCK<br>STORE               | Description Status Checks Monitoring                       | Tags Usage Instructions                                  |  |
| Volumes                              | Instance ID i-0c68cf2c5a3                                  | c37078 Public DNS  | (IPv4) ec2-54-199-17-108.ap-               |
| Snapshots                            |  |  | northeast-                                 |
| Lifecycle Manager                    | Instance state   | IBv4 Pu  | 1.compute.amazonaws.com                    |
| - NETWORK &                          | Instance type t2.micro                                     | IP V4 P0   | v6 IPs -                                   |
| SECURITY                             | Elastic IPs  | Private  | DNS ip-172-31-34-53.ap-                    |
| Security Groups                      |  |  | northeast-                                 |
| Elastic IPs                          |  |  | 1.compute.internal                         |
| Placement Groups                     | Availability zone ap-northeast-                            | a Priva  | te IPs 172.31.34.53                        |
|                                      | <ul> <li>Security groups</li> <li>CentOS 7 -x80</li> </ul> | Secondary priva  | •  |
| 🔍 Feedback 🔇                         | English (US) © 2008 - 21                                   | 119, Amazon Web Services, Inc. or its affiliates. All ri | ghts reserved. Privacy Policy Terms of Use |

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。

#### 126 ◀ Python区块链量化交易

| en 7. Review In   |  |  |   |
|---|--|--|---|
|   | Select an existing key pair or create a  | new key pair X   |   |
| Improve your in<br>AutogenByAW<br>Your instances may<br>addresses only.<br>You can also open<br>Edit security group | A key pair consists of a <b>public key</b> that AWS stores, and a <b>p</b><br>allow you to connect to your instance securely. For Windows<br>obtain the password used to log into your instance. For Linux<br>securely SSH into your instance.<br>Note: The selected key pair will be added to the set of keys a<br>about removing existing key pairs from a public AMI. | rivate key file that you store. Together, they<br>AMIs, the private key file is required to<br>AMIs, the private key file allows you to<br>uthorized for this instance. Learn more | I -<br>m known IP<br>80) for web servers. |
| AMI Details   | Create a new key pair  | •  | Edit AMI                                  |
| Jun Dotano  | Key pair name<br>tomv's vm   |  |   |
| Instance Type   |  | Download Key Pair  | Edit instance type                        |
| Security Groups   |  |  | Edit security groups                      |
| Instance Details  | You have to download the private key file (*.per<br>it in a secure and accessible location. You will<br>again after it's created.  | n file) before you can continue. <b>Store</b><br>I not be able to download the file  | Edit instance details                     |
|   |  |  | Editetorago                               |





图 5-10 找到 C 盘用户目录





| Lenovo  |   |   |   | -   | - × |
|---|---|---|---|---|-----|
| 文件 主页 共享 查看   |   |   |   |   | ~ 0 |
| ★ 日本 10 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   |   | □ 新建项目 •<br>1 经松访问 •<br>健<br>件共   | <ul> <li>●打开・</li> <li>■</li> <li>■</li> <li>編辑</li> <li>●</li> <li>の</li> <li>の&lt;</li></ul> | <ul> <li>全部选择</li> <li>全部取消</li> <li>反向选择</li> </ul>      | -   |
| 剪贴板   | 组织  | 新建  | 打开  | 选择  |     |
| ← → ∽ ↑ 🔲 « Windows_O   | S (C:) > 用户 > Lenovo v さ  | 在 Lenovo 中搜索  | like'   |   | Q   |
| ▼下戦 ★ ▲ ▲ ○ 文档 ★ ■ 回片 ★ ■ new ■ pyExcel ■ share ■ ubuntu ● OneDrive - Personal ■ 电子邮件期件 ■ 图片 ※ 文档 ※ 文档 ※ 公档 ※ 公档 ※ 公档 ※ 公档 ※ 公相 ※ ※ ※ ※ ※ ※ ※ ※ ※ ※ ※ ※ ※ ※ ※ ※ ※ ※ ※ | 2%<br>Tracing<br>verv<br>WeChatProjects<br>伊存的游戏<br>軍務人<br>種類<br>位数<br>使業人<br>準<br>構築<br>一<br>構築<br>一<br>電数人<br>一<br>電数人<br>一<br>電数人<br>一<br>電数人<br>一<br>電数人<br>一<br>電数人<br>一<br>電数人<br>一<br>電数人<br>一<br>電数人<br>一<br>電数人<br>一<br>電数人<br>一<br>電動<br>一<br>の<br>の<br>の<br>の<br>の<br>の<br>の<br>の<br>の<br>の<br>の<br>の<br>の | (神波日期)<br>2022/7/22 10:33<br>2020/4/19 15:46<br>2022/2/13 23:21<br>2021/2/15 12:01<br>2021/2/15 12:01<br>2021/2/15 12:01<br>2021/2/15 12:01<br>2021/2/15 12:01<br>2021/2/15 12:01<br>2021/2/17 16:04<br>2022/8/17 16:04<br>2022/8/17 16:04<br>2022/8/17 17:14 | 类型<br>文件夹<br>文件夹<br>文件夹<br>文件夹<br>文件夹<br>文件夹<br>文件夹<br>文件夹<br>文件夹<br>文件夹  | 大小<br>3 KB<br>1 KB<br>1 KB<br>2 KB                        | ^   |
| <ul> <li>3D 对象</li> <li>III 规模</li> <li>III 规模</li> <li>III 文档</li> <li>III 文档</li> <li>III 下载</li> <li>小音乐</li> </ul>  | ppmrc<br>python_history<br>Python_history<br>Pontes-20.5rc<br>electrum-private-keys.csv<br>   | 2020/6/4 19:55<br>2024/4/17 15:23<br>2021/12/19 19:54<br>2021/4/9 16:01<br>2024/3/19 11:37<br>2021/11/20 16:28<br>2024/4/11 15:58   | NPMRC 文件<br>PYTHON_HISTO<br>SRC 文件<br>CSV 文件<br>PEM 文件<br>BOX 文件<br>DAT 文件  | 1 KB<br>2 KB<br>1 KB<br>4 KB<br>2 KB<br>0 KB<br>15,104 KB |     |
| ● 日本<br>■ 貞面<br>● Windows_OS (C:)<br>■ 本地磁盘 (D:)<br>65 个项目 选中 1 个项目 1.63 KB   | ]] package-lock.json<br>] sinxcosx3.osp<br>]] userslenovoappdatalocaltemptmp3j5<br>] 未命名项目.osp  | 2021/9/11 11:21<br>2021/12/22 8:54<br>2022/2/8 21:31<br>2021/7/18 12:36   | JSON 文件<br>OSP 文件<br>文件<br>OSP 文件   | 1 KB<br>213 KB<br>0 KB<br>331 KB                          | •   |

图 5-12 把密钥文件复制到这个目录

### 128 ◀ I Python区块链量化交易



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 -1,指令如下:

ls -1

运行结果如图 5-17 所示。

| BB ubuntu@VM-4-12-ubuntu: ~/book  | _              |                  | $\times$ |
|---|----------------|------------------|----------|
| ubuntu@VM-4-12-ubuntu:~/book\$ ls<br>binanceFutureGetOrd.py binanceFuturesKline.py binanceSpotKline.py okDepth.py<br>binanceFutureMacd.py binanceSpotDepth.py okBalance.py okKline.py<br>ubuntu@VM-4-12-ubuntu:~/book\$ ls -l<br>total 44   | okMar<br>okTic | gin.py<br>ker.py | ^        |
| -rw-rw-r 1 ubuntu ubuntu 434 Apr 23 09:47 binanceFutureGetOrd.py<br>-rw-rw-r 1 ubuntu ubuntu 6335 Apr 23 09:46 binanceFutureMacd.py<br>-rw-rw-r 1 ubuntu ubuntu 586 Apr 23 09:58 binanceSpotDepth.py<br>-rw-rw-r 1 ubuntu ubuntu 436 Apr 23 10:02 binanceSpotDepth.py<br>-rw-rw-r 1 ubuntu ubuntu 585 Apr 23 09:59 okBalance.py<br>-rw-rw-r 1 ubuntu ubuntu 585 Apr 23 09:59 okBalance.py<br>-rw-rw-r 1 ubuntu ubuntu 202 Apr 23 09:59 okDepth.py<br>-rw-rw-r 1 ubuntu ubuntu 65 Apr 23 10:00 okkline.py<br>-rw-rw-r 1 ubuntu ubuntu 87 Apr 23 10:00 okkline.py<br>-rw-rw-r 1 ubuntu ubuntu 1263 Apr 23 09:44 okTicker.py<br>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 test0kLeverage.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 get0k
```

运行结果如图 5-22 所示。

| 🚥 ubuntu@VM-4-12-ubuntu: ~/book                                   | _ | $\times$ |
|---|---|----------|
| ubuntu@VM-4-12-ubuntu:~/book\$ ps ax   grep getOk<br>473145 pts/3 |   | ^        |
| 473434 pts/3 S+ 0:00 grepcolor=auto get0k                         |   |          |
| ubuntu@VM-4-12-ubuntu:~/book\$                                    |   |          |
|   |   |          |
|   |   |          |
|   |   |          |

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

| <b>git</b> distribut   | ted-is-the-new-centralized   | Q Search entire site   |
|--|--|--|
| About<br>Documentation<br>Downloads  | Downloads  | Latest source Release  |
| GUI Clients<br>Logos<br>Community  | <ul> <li>Mac OS X</li> <li>↓ Linux/Unix</li> </ul>   | <b>2.24.0</b><br>Release Notes (2019-11-04)<br>Download 2.24.0 for Windows                     |
| The entire <b>Pro Git book</b><br>written by Scott Chacon and<br>Ben Straub is available to read | Older releases are available and the Git source<br>repository is on GitHub.                                    |  |
| online for free. Dead tree<br>versions are available on<br>Amazon.com.                           | GUI Clients<br>Git comes with built-in GUI tools (git-gui,<br>cith) but there are served third participals for | Logos<br>Various Git logos in PNG (bitmap) and EPS<br>(water) formate are available for use in |

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。

| 🖏 命令提示符  |  | -   | $\times$ |
|--|--|-----|----------|
| Microsoft Windows [版<br>(c) Microsoft Corpora  | 本 10. 0. 19045. 4291]<br>ation。保留所有权利。   |     | ^        |
| C:\Users\Lenovo>git<br>usage: git [version<br>[exec-p¤<br>[-p]pe<br>[git-din<br><command/> | n] [help] [-C <path>] [-c <name>=<value>]<br/>th[=<path>]] [html-path] [man-path] [info-path]<br/>aginate   -P  no-pager] [no-replace-objects] [bare]<br/>=<path>] [work-tree=<path>] [namespace=<name>]<br/>[<args>]</args></name></path></path></path></value></name></path> |     |          |
| These are common Git   | commands used in various situations:   |     |          |
| start a working area<br>clone<br>init  | (see also: git help tutorial)<br>Clone a repository into a new directory<br>Create an empty Git repository or reinitialize an existing   | one |          |
| work on the current of<br>add<br>mv<br>restore<br>rm<br>sparse-checkout                    | change (see also: git help everyday)<br>Add file contents to the index<br>Move or rename a file, a directory, or a symlink<br>Restore working tree files<br>Remove files from the working tree and from the index<br>Initialize and modify the sparse-checkout                 |     |          |
| examine the history a<br>bisect<br>diff<br>grep  | and state (see also: git help revisions)<br>Use binary search to find the commit that introduced a bug<br>Show changes between commits, commit and working tree, etc<br>Print lines matching a pattern   |     |          |

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。

| ubuntu@VM-4-12-ubuntu: ~                                   | - | × |
|--|---|---|
| ubuntu@VM-4-12-ubuntu:~\$ gitversion<br>git version 2.34.1 |   | ^ |
| ubuntu@VM-4-12-ubuntu:~\$                                  |   |   |
|  |   |   |
|  |   |   |
|  |   |   |
|  |   |   |
|  |   | ~ |

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。

| 😳 ninki  | - / book  |        |
|--|---|--------|
| 仓库地址: https://git  | ee.com/ninki_admin/book   |        |
| 仓库介绍 16/   |   | 16/200 |
| Python量化交易编  | 星示例代码   |        |
|  | ana mana ana  |        |
|  |   |        |
| ● 开源 (所有人可)  |   |        |
| 私有 (仅仓库成员)   | <sub>1可见)</sub> 选择私有 )  |        |
|  |   |        |
|  | 『语言aitianore. 开源许可证)  |        |
| 📄 初始化仓库 (设置  | and the second |        |
| <ul> <li>初始化仓库(设置</li> <li>设置模板(添加)</li> </ul>                   | Readme、Issue、Pull Request 模板文件)   |        |
| <ul> <li>初始化仓库(设置</li> <li>设置模板(添加)</li> <li>送置模板(添加)</li> </ul> | Readme、Issue、Pull Request 模板文件)   |        |

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