

---

# 《虚拟化与云计算实践》

(2024-2025 学年第 2 学期)

## 实验一

---



| 姓名  | 学号            | 班级        |
|-----|---------------|-----------|
| 江欣怡 | 2022337621242 | 22 计科 2 班 |
| 周洪蕊 | 2022337621245 | 22 计科 2 班 |
| 祝云佳 | 2022337621086 | 22 计科 3 班 |

2025 年 6 月 6 日

---

### 1 新建 VMware Workstation 虚拟机

1. 启动 VMware Workstation，右击左上角“文件”点击“新建虚拟机”，选择“经典模式”，点击“下一步”：



2. 选择系统镜像文件，点击“下一步”：



3. 输入 Linux 计算机全名、用户名、密码及确认密码，然后点击“下一步”：

新建虚拟机向导

简易安装信息  
这用于安装 Ubuntu 64 位。

个性化 Linux

全名(F): u18

用户名(U): user

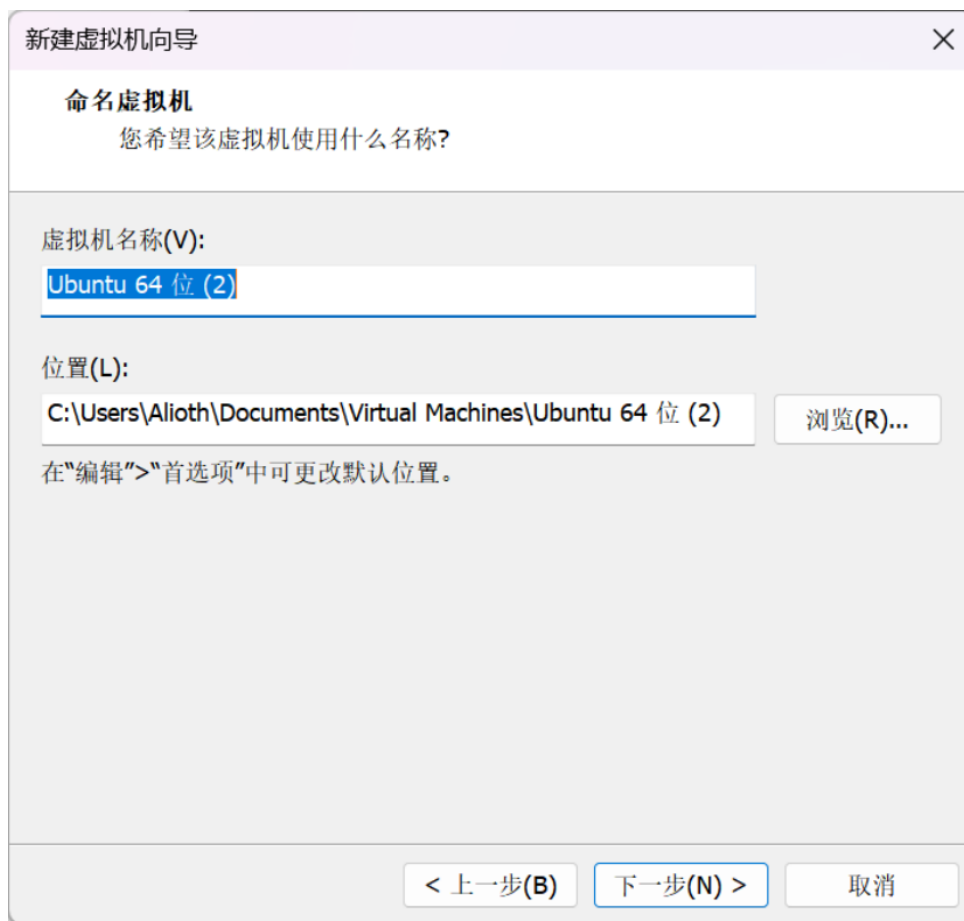
密码(P): ●●●●●●

确认(C): ●●●●●●

帮助 < 上一步(B) 下一步(N) > 取消

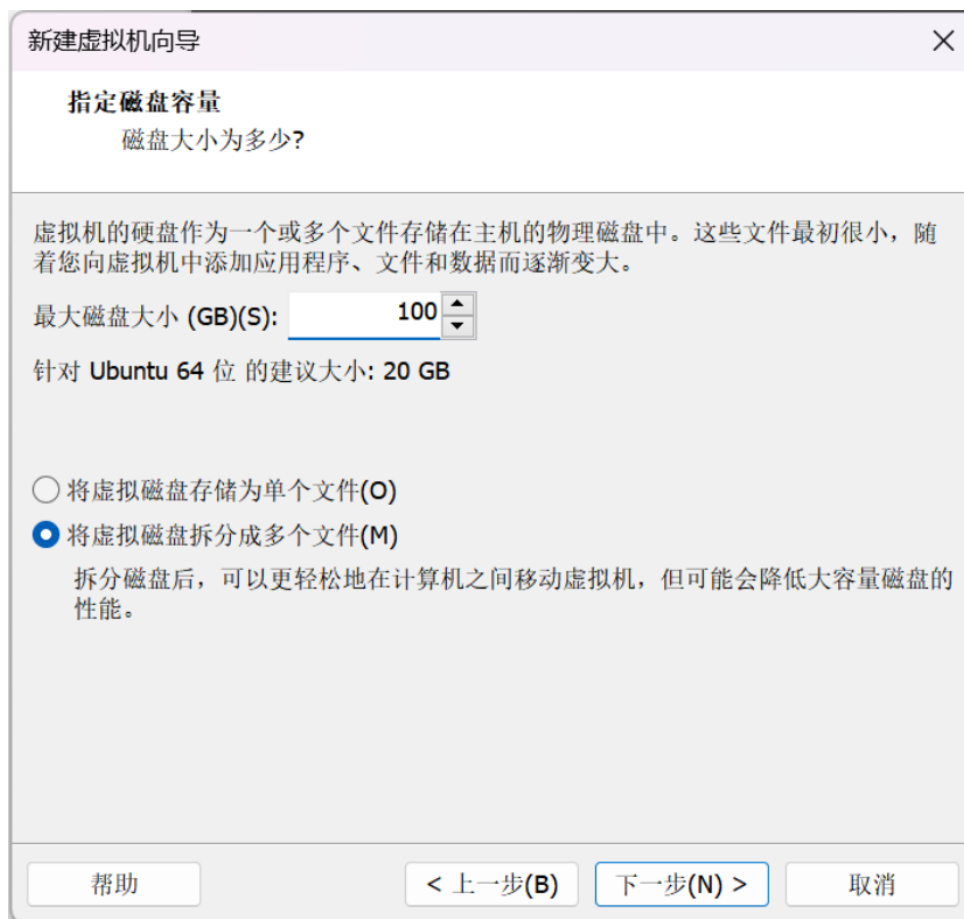
- 全名：这是计算机的名字，在命令行终端会显示，建议简短以系统版本命名。
- 用户名：默认的用户名，会在命令行终端显示，建议简短，英文。
- 密码：建议简短，方便经常输入。

4. 输入虚拟机名称、安装位置，点击“下一步”：

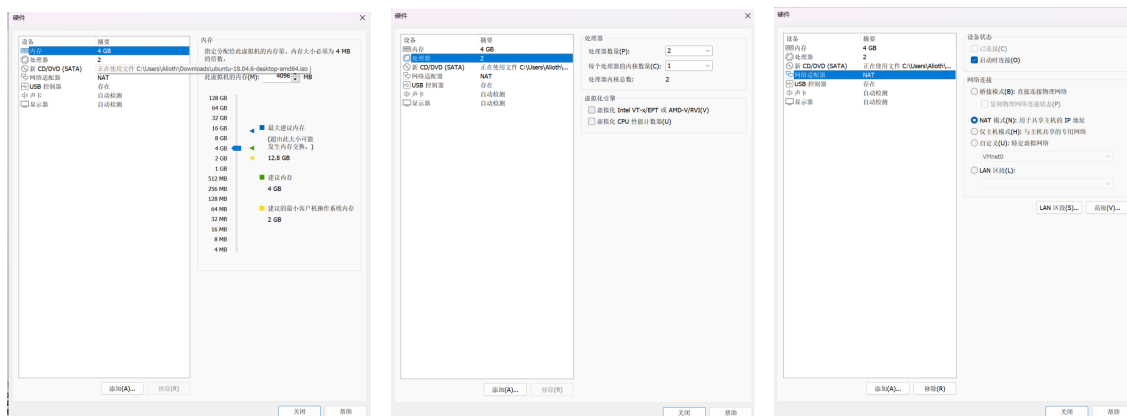


- 虚拟机名称：这里的虚拟机名称是 VMware 管理虚拟机时使用的，不是虚拟机内部的名称。
- 安装位置：选择一个空间足够的位置即可，尽量不要包含中文、特殊符号、空格等。

5. 设置最大磁盘大小，并勾选“将虚拟磁盘分为多个文件”，点击“下一步”：



6. 点击自定义硬件，根据自身宿主机的硬件情况进行配置：



内存

处理器

网络适配器

配置完成后点击“关闭”按钮，确认后点击“完成”，虚拟机会自动启动，等待系统安装完成。

## 2 Docker 安装步骤 (Ubuntu)

1. 更新系统软件包：

## 2 DOCKER 安装步骤 (UBUNTU)

```
sudo apt update
```

```
root@ubuntu:/home/user# sudo apt update
命中:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
命中:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
命中:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
命中:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease
获取:5 http://us.archive.ubuntu.com/ubuntu bionic/main Translation-zh_CN [67.7 kB]
获取:6 http://us.archive.ubuntu.com/ubuntu bionic/restricted Translation-zh_CN [1,188 B]
获取:7 http://us.archive.ubuntu.com/ubuntu bionic/universe Translation-zh_CN [174 kB]
获取:8 http://us.archive.ubuntu.com/ubuntu bionic/multiverse Translation-zh_CN [4,768 B]
已下载 248 kB, 耗时 3秒 (74.5 kB/s)
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
所有软件包均为最新。
```

### 2. 安装依赖包【用于通过 HTTPS 来获取仓库】:

```
sudo apt install apt-transport-https ca-certificates
```

```
root@ubuntu:/home/user# sudo apt install apt-transport-https ca-certificates
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
ca-certificates 已经是最新版 (20230311ubuntu0.18.04.1)。
ca-certificates 已设置为手动安装。
下列软件包是自动安装的并且现在不需要了:
 fonts-liberation2 fonts-opensymbol gir1.2-goa-1.0
 gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0
 gir1.2-snapd-1 gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3
 libboost-date-time1.65.1 libboost-filesystem1.65.1 libboost-iostreams1.65.1
 libboost-locale1.65.1 libcdt-0.1-1 libclucene-contribs1v5 libclucene-core1v5
 libcms-0.5-5v5 libcolamd2 libdazzle-1.0-0 libe-book-0.1-1
 libedataserverui-1.2-2 libeot libepubgen-0.1-1 libetonyek-0.1-1
 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2 libfreerdp2-2 libgck2
 libgee-0.8-2 libgexiv2-2 libgom-1.0-0 libgpgmepp6 libgpgod-common libgpgod4
 liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0
 libmspub-0.1-1 libodfgen-0.1-1 libqwing2v5 libraw16 libvenge-0.0-0
 libsguttl2-2 libssh-4 libsuitesparseconfig5 libvncclient1 libwinpr2-2
 libxapian30 libxmlsec1-nss lp-solve media-player-info python3-mako
 python3-markupsafe syslinux syslinux-common syslinux-legacy
 usb-creator-common
使用'sudo apt autoremove'来卸载它(它们)。
下列【新】软件包将被安装:
 apt-transport-https
升级了 0 个软件包, 新安装了 1 个软件包, 要卸载 0 个软件包, 有 0 个软件包未被升级。
需要下载 1,692 B 的归档。
解压缩后会消耗 155 kB 的额外空间。
您希望继续执行吗? [Y/n] y
获取:1 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 apt-transport-https all 1.6.17 [1,692 B]
已下载 1,692 B, 耗时 1秒 (1,721 B/s)
正在选中未选择的软件包 apt-transport-https。
(正在读取数据库 ... 系统当前共安装有 161926 个文件和目录。)
准备解包 .../apt-transport-https 1.6.17_all.deb ...
正在解包 apt-transport-https (1.6.17) ...
正在设置 apt-transport-https (1.6.17) ...
```

```
curl software-properties-common
```

```
root@ubuntu:/home/user# curl software-properties-common
curl: (6) Could not resolve host: software-properties-common
```

```
sudo snap install curl #如果提示找不到 curl 命令, 则执行
```

```
root@ubuntu:/home/user# sudo snap install curl
curl 8.13.0 from hideo aoyama (aoilinux) installed
```

## 2 DOCKER 安装步骤 (UBUNTU)

curl 是一个用于在命令行中传输数据的工具, 支持多种协议, 包括 HTTP、HTTPS、FTP 等。

### 3. 添加 Docker 官方 GPG 密钥:

```
sudo -i #用来在终端中以 root 用户的身份启动一个新的登录 shell
```

```
root@ubuntu:/home/user# sudo -i
```

```
curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg  
| gpg --dearmor -o /etc/apt/trusted.gpg.d/docker-ce.gpg #添  
加 Docker 官方 GPG 密钥
```

```
root@ubuntu:~# curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | gpg --dearmor -o /etc/apt/trusted.gpg.d/docker-ce.gpg
```

### 4. 验证密钥:

```
sudo apt-key fingerprint 0EBFCD88
```

```
root@ubuntu:~# sudo apt-key fingerprint 0EBFCD88  
pub   rsa4096 2017-02-22 [SCEA]  
      9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88  
uid   [ 未知 ] Docker Release (CE deb) <docker@docker.com>  
sub   rsa4096 2017-02-22 [S]
```

### 5. 添加 Docker 阿里稳定版软件源:

```
sudo add-apt-repository "deb [arch=amd64] https://mirrors.aliyun  
.com/docker-ce/linux/ubuntu $(lsb_release -cs) stable"
```

```
root@ubuntu:~# sudo add-apt-repository "deb [arch=amd64] https://mirrors.aliyun.com/docker-ce/linux/ubuntu $(lsb_release -cs) stable"  
获取:1 https://mirrors.aliyun.com/docker-ce/linux/ubuntu bionic InRelease [64.4 kB]  
命中:2 http://security.ubuntu.com/ubuntu bionic-security InRelease  
命中:3 http://us.archive.ubuntu.com/ubuntu bionic InRelease  
获取:4 https://mirrors.aliyun.com/docker-ce/linux/ubuntu bionic/stable amd64 Packages [39.0 kB]  
命中:5 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease  
命中:6 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease  
已下载 103 kB, 耗时 2秒 (58.8 kB/s)  
正在读取软件包列表... 完成
```

### 6. 更新并安装 Docker:

```
sudo apt update #再次更新软件包
```

```
root@ubuntu:~# sudo apt update  
命中:1 https://mirrors.aliyun.com/docker-ce/linux/ubuntu bionic InRelease  
命中:2 http://security.ubuntu.com/ubuntu bionic-security InRelease  
命中:3 http://us.archive.ubuntu.com/ubuntu bionic InRelease  
命中:4 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease  
命中:5 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease  
正在读取软件包列表... 完成  
正在分析软件包的依赖关系树  
正在读取状态信息... 完成  
所有软件包均为最新。
```



## 2 DOCKER 安装步骤 (UBUNTU)

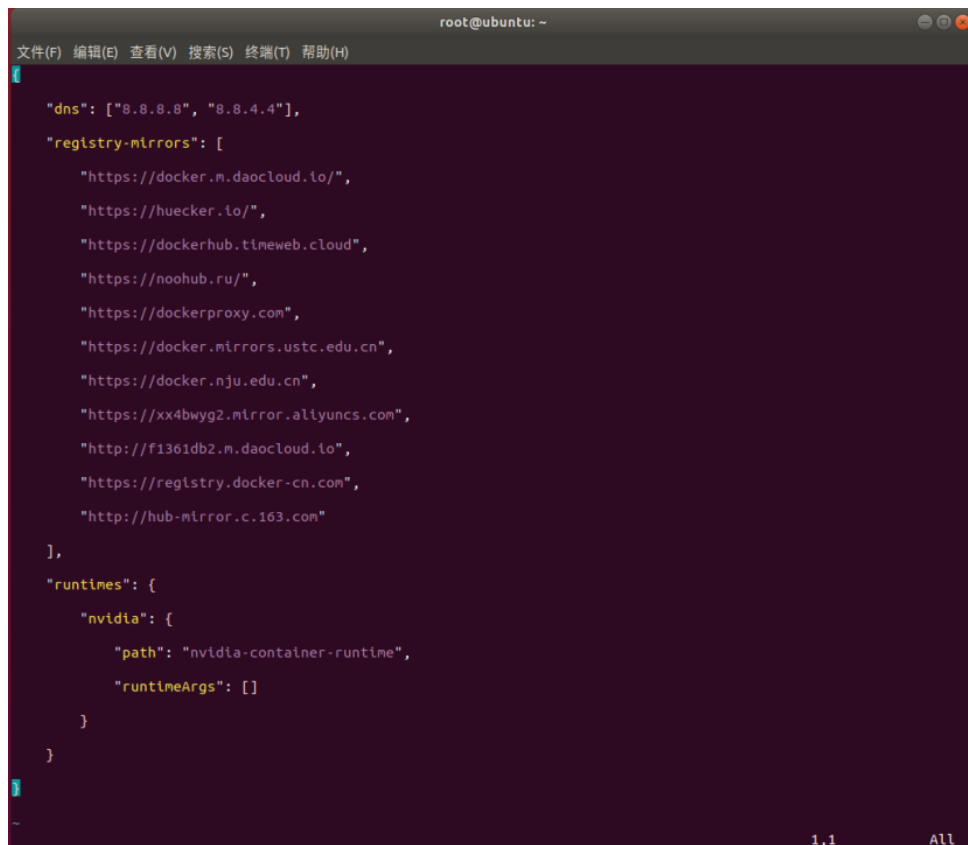
```
sudo apt install docker-ce docker-ce-cli containerd.io #安装 Docker
```

```
root@ubuntu:~# sudo apt install docker-ce docker-ce-cli containerd.io
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了:
 fonts-liberation2 fonts-opensymbol gir1.2-goa-1.0
 gir1.2-gst-plugins-base-1.0 gir1.2-gstreamer-1.0 gir1.2-gudev-1.0
 gir1.2-snapd-1 gir1.2-udisks-2.0 grilo-plugins-0.3-base gstreamer1.0-gtk3
 libboost-date-time1.65.1 libboost-filesystem1.65.1 libboost-iostreams1.65.1
 libboost-locale1.65.1 libcdt-0.1.1 libclucene-contrib1b1v5 libclucene-core1v5
 libcnis-0.5-5v5 libcolamd2 libdazzle-1.0-0 libe-book-0.1-1
 libdataserverui-1.2-2 libeot0 libepubgen-0.1-1 libetonyek-0.1-1
 libevent-2.1-6 libexiv2-14 libfreerdp-client2-2 libfreerdp2-2 libgck1
 libgce-0.8-2 libgettext2-2 libgsm-1.0-0 libgpgmepp6 libgpgod-common libgpgod4
 liblangtag-common liblangtag1 liblirc-client0 liblua5.3-0 libmediaart-2.0-0
 libmspub-0.1-1 libodfgen-0.1-1 libqwing2v5 libraw16 librevenge-0.0-0
 libsgutil2-2 libssh-4 libsuitesparseconfig5 libvncclient1 libwinpr2-2
 libxapian30 libxmisc1-nss lp-solve media-player-info python3-nako
 python3-markupsafe syslinux syslinux-common syslinux-legacy
 usb-creator-common
使用 'sudo apt autoremove' 来卸载它(它们)。
将会同时安装下列软件:
 docker-buildx-plugin docker-ce-rootless-extras docker-compose-plugin pigz
建议安装:
 aufs-tools cgroupfs-mount | cgroup-lite
推荐安装:
 slirp4netns
下列【新】软件包将被安装:
 containerd.io docker-buildx-plugin docker-ce docker-ce-cli
 docker-ce-rootless-extras docker-compose-plugin pigz
升级了 0 个软件包，新安装了 7 个软件包，要卸载 0 个软件包，有 0 个软件包未被升级。
需要下载 111 MB 的归档。
解压后还会消耗 401 MB 的额外空间。
您希望继续执行吗? [Y/n] y
获取 21 https://mirrors.aliyun.com/docker-ce/linux/ubuntu bionic/stable amd64 containerd.io amd64 1.6.21-1 [28.3 MB]
获取 12 https://us.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
获取 13 https://mirrors.aliyun.com/docker-ce/linux/ubuntu bionic/stable amd64 docker-buildx-plugin amd64 0.10.5-1-ubuntu.18.04-bionic [26.1 MB]
获取 14 https://mirrors.aliyun.com/docker-ce/linux/ubuntu bionic/stable amd64 docker-ce-cli amd64 5:24.0.2-1-ubuntu.18.04-bionic [13.3 MB]
获取 15 https://mirrors.aliyun.com/docker-ce/linux/ubuntu bionic/stable amd64 docker-ce amd64 5:24.0.2-1-ubuntu.18.04-bionic [22.0 MB]
获取 16 https://mirrors.aliyun.com/docker-ce/linux/ubuntu bionic/stable amd64 docker-ce-rootless-extras amd64 5:24.0.2-1-ubuntu.18.04-bionic [9,014 kB]
获取 17 https://mirrors.aliyun.com/docker-ce/linux/ubuntu bionic/stable amd64 docker-compose-plugin amd64 2.18.1-1-ubuntu.18.04-bionic [10.9 MB]
已下载 111 MB，耗时 45秒 (2,429 kB/s)
正在选中未选择的软件包 pigz。
(正在读取数据库... 系统当前共安装有 161930 个文件和目录。)
正准备解包 .../0-pigz-2.4-1_amd64.deb ...
正在解包 pigz (2.4-1) ...
正在选中未选择的软件包 containerd.io。
正准备解包 .../1-containerd.io-1.6.21-1_amd64.deb ...
正在解包 containerd.io (1.6.21-1) ...
正在选中未选择的软件包 docker-buildx-plugin。
正准备解包 .../2-docker-buildx-plugin-0.10.5-1-ubuntu.18.04-bionic_amd64.deb ...
正在解包 docker-buildx-plugin (0.10.5-1-ubuntu.18.04-bionic) ...
正在选中未选择的软件包 docker-ce-cli。
正准备解包 .../3-docker-ce-cli-5:24.0.2-1-ubuntu.18.04-bionic_amd64.deb ...
正在解包 docker-ce-cli (5:24.0.2-1-ubuntu.18.04-bionic) ...
正在选中未选择的软件包 docker-ce。
正准备解包 .../4-docker-ce-5:24.0.2-1-ubuntu.18.04-bionic_amd64.deb ...
正在解包 docker-ce (5:24.0.2-1-ubuntu.18.04-bionic) ...
正在选中未选择的软件包 docker-ce-rootless-extras。
正准备解包 .../5-docker-ce-rootless-extras-5:24.0.2-1-ubuntu.18.04-bionic_amd64.deb ...
正在解包 docker-ce-rootless-extras (5:24.0.2-1-ubuntu.18.04-bionic) ...
正在选中未选择的软件包 docker-compose-plugin。
正准备解包 .../6-docker-compose-plugin-2.18.1-1-ubuntu.18.04-bionic_amd64.deb ...
正在解包 docker-compose-plugin (2.18.1-1-ubuntu.18.04-bionic) ...
正在设置 containerd.io (1.6.21-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
正在设置 docker-ce-rootless-extras (5:24.0.2-1-ubuntu.18.04-bionic) ...
正在设置 docker-buildx-plugin (0.10.5-1-ubuntu.18.04-bionic) ...
正在设置 docker-compose-plugin (2.18.1-1-ubuntu.18.04-bionic) ...
正在设置 docker-ce-cli (5:24.0.2-1-ubuntu.18.04-bionic) ...
正在设置 pigz (2.4-1) ...
正在设置 docker-ce (5:24.0.2-1-ubuntu.18.04-bionic) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
正在处理用于 systemd (337-ubuntu18.07) 的触发器 ...
正在处理用于 nan-db (2.8.3-2ubuntu0.1) 的触发器 ...
正在处理用于 ureadahead (0.100-0-21) 的触发器 ...
root@ubuntu:~# sudo docker run hello-world
```

## 7. 测试 Docker 安装:

```
sudo nano /etc/docker/daemon.json #编辑配置文件 (没有这个文件则进行创建)
```

## 2 DOCKER 安装步骤 (UBUNTU)



```
root@ubuntu: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

{
  "dns": ["8.8.8.8", "8.8.4.4"],
  "registry-mirrors": [
    "https://docker.m.daocloud.io/",
    "https://huecker.io/",
    "https://dockerhub.tineweb.cloud",
    "https://noohub.ru/",
    "https://dockerproxy.com",
    "https://docker.mirrors.ustc.edu.cn",
    "https://docker.nju.edu.cn",
    "https://xx4bwyg2.mirror.aliyuncs.com",
    "http://f1361db2.m.daocloud.io",
    "https://registry.docker-cn.com",
    "http://hub-mirror.c.163.com"
  ],
  "runtimes": {
    "nvidia": {
      "path": "nvidia-container-runtime",
      "runtimeArgs": []
    }
  }
}
```

```
sudo systemctl daemon-reload #重新加载
```

```
root@ubuntu:~# sudo systemctl daemon-reload
```

```
sudo systemctl restart docker #重新启动docker
```

```
root@ubuntu:~# sudo systemctl restart docker
```

```
sudo docker run hello-world
```

```
root@ubuntu:~# sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
e6590344b1a5: Pull complete
Digest: sha256:dd01f97f252193ae3210da231b1dca0cffab4aad3566692d6730bf93f123a48
Status: Downloaded newer image for hello-world:latest

hello from Docker!

This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/
```

### 8. 配置用户组:

```
sudo usermod -aG docker 用户名
```

```
root@ubuntu:~# sudo usermod -aG docker user
```

```
su - 用户名 #刷新shell状态
```

```
root@ubuntu:~# su - user
```

```
docker images #验证用户组配置
```

```
user@ubuntu:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world    latest    74cc54e27dc4   4 months ago   10.1kB
```

### 常用 Docker 运行命令:

```
sudo systemctl status docker #查看 Docker 状态
sudo systemctl start docker #启动 Docker
sudo systemctl enable docker #设置 Docker 开机自启
sudo systemctl stop docker #停止 Docker 服务
```

## 3 部署 DeepSeek 模型

### 1. 安装 Ollama:

```
docker pull ollama/ollama #拉取 Ollama 的 Docker 镜像
```

```
root@ubuntu:~# docker pull ollama/ollama
Using default tag: latest
latest: Pulling from ollama/ollama
13b7e930469f: Pull complete
97ca0261c313: Pull complete
bdb00c53f489: Pull complete
54fa3bd5e7f0: Downloading [=====>] 244.2MB/1.71GB
54fa3bd5e7f0: Downloading [=====>] 459.8MB/1.71GB
54fa3bd5e7f0: Downloading [=====>] 564.6MB/1.71GB
^C
```

由于网络问题，下载超时，从华为云容器镜像服务拉取 Docker 镜像，并重命名为标准镜像名以便本地使用。

```
docker pull swr.cn-north-4.myhuaweicloud.com/ddn-k8s/docker.io/ollama/ollama:0.3.13
```

```
root@ubuntu:~# docker pull swr.cn-north-4.myhuaweicloud.com/ddn-k8s/docker.io/ollama/ollama:0.3.13
0.3.13: Pulling from ddn-k8s/docker.io/ollama/ollama
7478e0ac0f23: Pull complete
75a058f715a3: Pull complete
99ef3d1416df: Pull complete
96587fbd30e: Pull complete
5fb1e92c3720: Pull complete
6043d7740b8f: Pull complete
048c8c4ad84e: Pull complete
d05d3a05f8c0: Downloading [=====>] 858.5MB/877MB
ll complete
Digest: sha256:b9bf6eb97f0dad240e628e95e8ea875fc1ddeed688a0d7f69f8ffbdddad1b966c
Status: Downloaded newer image for swr.cn-north-4.myhuaweicloud.com/ddn-k8s/docker.io/ollama/ollama:0.3.13
swr.cn-north-4.myhuaweicloud.com/ddn-k8s/docker.io/ollama/ollama:0.3.13
```

```
docker tag swr.cn-north-4.myhuaweicloud.com/ddn-k8s/docker.io/ollama/ollama:0.3.13 ollama/ollama:0.3.13
```

```
root@ubuntu:~# docker tag swr.cn-north-4.myhuaweicloud.com/ddn-k8s/docker.io/ollama/ollama:0.3.13 ollama/ollama:0.3.13
```

### 2. 创建 Ollama 容器:

```
docker run -d --name ollama -p 11434:11434 ollama/ollama:0.3.13
#启动 Ollama 容器并映射必要的端口（如 11434）
```

```
root@ubuntu:~# docker run -d --name ollama -p 11434:11434 ollama/ollama:0.3.13
2b2106f659061d97c42e7f0a3da36b4a5b15d731e3277af359323778e42ffc79
```

```
docker ps #确认是否运行成功
```

```
root@ubuntu:~# docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED        STATUS        PORTS                               NAMES
2b2106f65906   ollama/ollama:0.3.13 "/bin/ollama serve"     15 seconds ago Up 13 seconds 0.0.0.0:11434->11434/tcp, :::11434->11434/tcp   ollama
```

### 3 部署 DEEPSEEK 模型

#### 3. 下载并加载 DeepSeek 模型:

```
docker exec -it ollama bash # 进入 Ollama 容器
```

```
root@ubuntu:~# docker exec -it ollama bash
```

```
ollama run deepseek-r1:1.5b # 在容器内使用 Ollama 命令加载  
DeepSeek 模型 (以 deepseek:1.5b 为例)
```

```
root@2b2106f65906:/# ollama run deepseek-r1:1.5b  
pulling manifest  
pulling manifest  
pulling manifest  
pulling manifest  
pulling manifest  
pulling manifest  
pulling manifest  
pulling manifest  
pulling manifest  
pulling aabd4debf0c8... 100% 1.1 GB  
pulling 369ca498f347... 100% 387 B  
pulling 6e4c38e1172f... 100% 1.1 KB  
pulling f4d24e9138dd... 100% 148 B  
pulling a85fe2a2e58e... 100% 487 B  
verifying sha256 digest  
writing manifest  
success
```

#### 4. 与 DeepSeek 模型进行交互及退出方式:

```
>>> hello  
<think>  
</think>  
Hello! How can I assist you today? 😊  
  
>>> 你好  
<think>  
Alright, the user greeted me with "你好" and I responded with a greeting. I should check if this is okay  
or if there's something else they want to discuss.  
  
I think it would be better to keep the conversation open for further interaction. Maybe I can ask them how  
I can help today or suggest some topics they're interested in.  
  
Let me adjust my response accordingly.  
</think>  
你好! 有什么我可以帮助你的吗?  
  
>>>  
Use Ctrl + d or /bye to exit.  
>>> bye  
<think>  
嗯, "bye" 是你让我想到的。希望这些建议能对你的学习或生活有所帮助! 如果需要更多的信息, 随时告诉我哦- 😊  
>>>
```

‘Ctrl + D’: 退出模型对话, 回到容器命令行; 输入 ‘exit’: 退出容器, 回到本地终端。若需再次使用模型, 可通过以下命令重新启动容器并加载模型:

```
docker start ollama # 启动 ollama  
docker exec -it ollama bash# 进入 Ollama 容器  
ollama run deepseek-r1:1.5b
```

```
root@ubuntu:~# docker start ollama
ollama
root@ubuntu:~# docker exec -it ollama bash
root@2b2106f65906:/# ollama run deepseek-r1:1.5b
>>> Send a message (/? for help)
```