山东大学 计算机科学与技术 学院

云计算技术 课程实验报告

学号：202200130048	姓名： 陈静雯	班级： 6 班
实验题目：虚拟化技术练习四 Docker		
实验学时：2	实验日期： 2025.4.2	

实验目的：熟悉 Docker 虚拟化环境。
具体包括：了解 Docker 虚拟化环境的配置和部署，完成实验环境及实验工具的熟悉，包括使用官方镜像运行容器，以及借助官方镜像构建、运行自己的镜像和容器，例如将之前设计的个人博客或者网站应用，通过 Dockerfile 构建镜像，并运行，同时了解和对比官方和自己构建的镜像文件，撰写实验报告。

硬件环境：
联网的计算机一台

软件环境：
Vmware，ubuntu，docker

实验步骤：

1. 安装 Docker 引擎
2. 配置用户权限
3. 验证安装
4. 使用官方镜像运行容器
（1）拉取并运行 Nginx 容器
（2）访问默认网址验证服务
5. 将之前的个人博客构建自定义镜像
（1）复制 html 文件到文件夹
（2）编写 dockerfile
（3）构建并运行镜像
（4）访问网站进行验证

前置知识：

一、Docker 核心概念

1. 镜像（Image）
   - 定义：镜像是只读的模板，包含运行应用所需的代码、运行时环境、库和配置。
   - 特点：镜像分层存储，每一层对应 Dockerfile 中的一条指令，复用层可减少存储占用。
   - 来源：
     - 官方镜像（如 nginx、mysql）托管在 Docker Hub。
     - 用户可自定义镜像并推送至私有仓库（如阿里云容器镜像服务）。
2. 容器（Container）
   - 定义：容器是镜像的运行实例，具有独立的文件系统、网络和进程空间。

- 生命周期：
  - 创建：docker create
  - 启动：docker start
  - 停止：docker stop
  - 删除：docker rm
- 特点：容器是临时的，默认情况下停止后数据不保留（需通过卷持久化）。

3. 仓库（Registry）
   - 定义：存储和分发镜像的平台。
   - 分类：
     - 公共仓库：如 Docker Hub、GitHub Container Registry。
     - 私有仓库：如 Harbor、AWS ECR、阿里云 ACR。

二、Docker 常用命令扩展

1. 容器操作
   - # 查看容器日志
     docker logs <容器 ID>
   - # 进入运行中的容器
     docker exec -it <容器 ID> sh
   - # 查看容器资源占用
     docker stats <容器 ID>
   - # 导出/导入容器快照
     docker export <容器 ID> > my-container.tar
     docker import my-container.tar my-image:latest
2. 镜像管理
   - # 查看镜像详细信息
     docker inspect <镜像 ID>
   - # 标记镜像并推送到仓库
     docker tag my-image:latest username/my-image:v1
     docker push username/my-image:v1
   - # 删除所有悬空镜像（未被任何容器引用的中间层）
     docker image prune
3. 网络管理
   - # 创建自定义网络
   - docker network create my-network
   - # 查看容器 IP 地址
   - docker inspect -f '{{.NetworkSettings.IPAddress}}' <容器 ID>
   - # 容器间通过别名通信
     docker run --network=my-network --name=app1 -d my-app
     docker run --network=my-network --name=app2 -d my-app
   - # 在 app2 中可通过 "ping app1" 访问
4. 数据卷（Volume）管理
   - # 创建卷
     docker volume create my-volume
   - # 挂载卷到容器
     docker run -v my-volume:/data my-app

- # 查看卷详情
  docker volume inspect my-volume

实验内容：

1. **安装 docker**

```
orange@orange-VMware-Virtual-Platform:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs cgroupfs-mount | cgroup-lite debootstrap docker-buildx
  | zfsutils git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb g
The following NEW packages will be installed:
  bridge-utils containerd docker.io git git-man liberror-perl pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 248 not upgraded.
Need to get 83.0 MB of archives.
After this operation, 325 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://mirrors.tuna.tsinghua.edu.cn/ubuntu noble/universe amd64 pigz amd64 2.8-1 [6
Get:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu noble/main amd64 bridge-utils amd64 1.7.
Get:3 http://mirrors.tuna.tsinghua.edu.cn/ubuntu noble-updates/main amd64 runc amd64 1.1.
Get:4 http://cn.archive.ubuntu.com/ubuntu noble-updates/main amd64 containerd amd64 1.7.2
Get:5 http://cn.archive.ubuntu.com/ubuntu noble-updates/universe amd64 docker.io amd64 26
Get:6 http://mirrors.tuna.tsinghua.edu.cn/ubuntu noble/main amd64 liberror-perl all 0.170
Get:7 http://mirrors.tuna.tsinghua.edu.cn/ubuntu noble-updates/main amd64 git-man all 1:2
Get:8 http://mirrors.tuna.tsinghua.edu.cn/ubuntu noble-updates/main amd64 git amd64 1:2.4
Get:9 http://mirrors.tuna.tsinghua.edu.cn/ubuntu noble/universe amd64 ubuntu-fan all 0.12
Fetched 83.0 MB in 2min 26s (570 kB/s)
```

2. **将当前用户加入用户组**

```
orange@orange-VMware-Virtual-Platform:~$ sudo groupadd docker
groupadd: group 'docker' already exists
orange@orange-VMware-Virtual-Platform:~$ sudo usermod -aG docker $USER
orange@orange-VMware-Virtual-Platform:~$ newgrp docker
orange@orange-VMware-Virtual-Platform:~$ groups
docker adm cdrom sudo dip plugdev users lpadmin orange
```

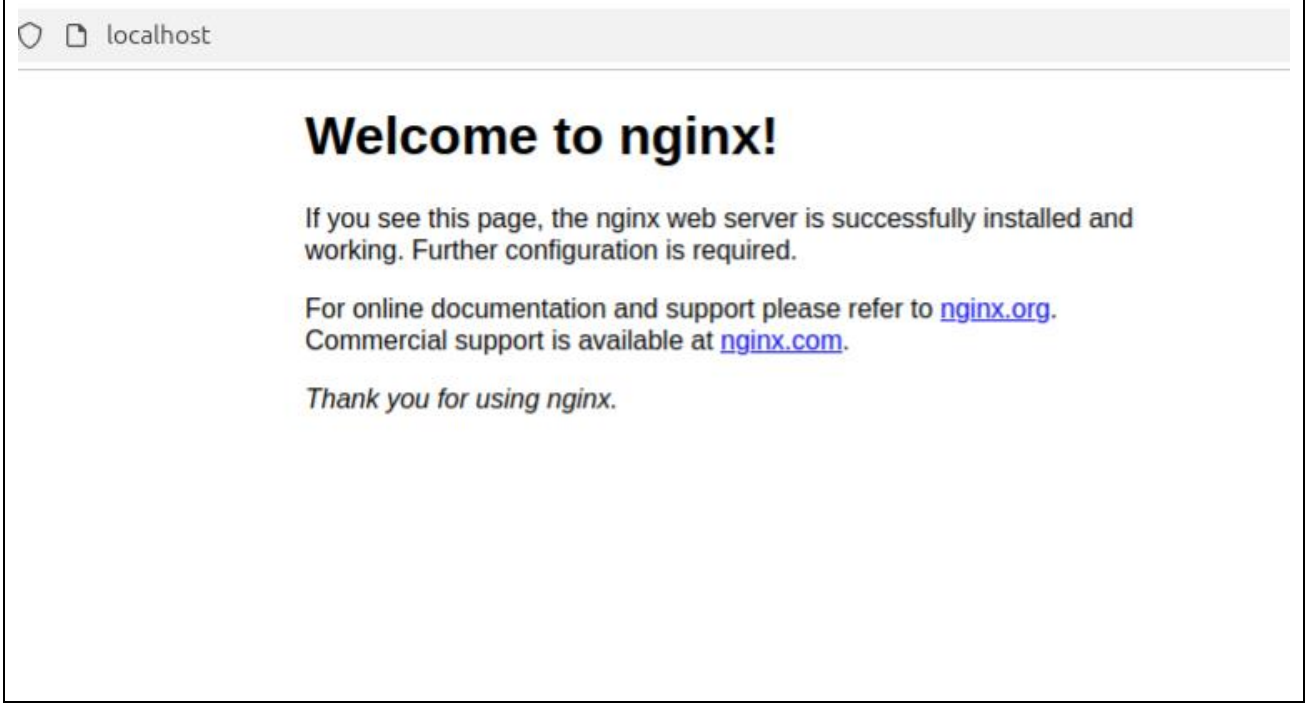
3. **验证安装**

```
orange@orange-VMware-Virtual-Platform:~$ docker --version
Docker version 26.1.3, build 26.1.3-0ubuntu1~24.04.1
```

4. 配置镜像源，拉取并运行 Nginx 官方镜像

```
orange@orange-VMware-Virtual-Platform:~$ sudo vim /etc/docker/daemon.json
orange@orange-VMware-Virtual-Platform:~$ sudo systemctl restart docker
orange@orange-VMware-Virtual-Platform:~$ docker pull nginx:latest
latest: Pulling from library/nginx
6e909acdb790: Pull complete
5eaa34f5b9c2: Pull complete
417c4bccf534: Pull complete
e7e0ca015e55: Pull complete
373fe654e984: Pull complete
97f5c0f51d43: Pull complete
c22eb46e871a: Pull complete
Digest: sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc19
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
```

```
orange@orange-VMware-Virtual-Platform:~$ docker pull nginx:latest
latest: Pulling from library/nginx
6e909acdb790: Pull complete
5eaa34f5b9c2: Pull complete
417c4bccf534: Pull complete
e7e0ca015e55: Pull complete
373fe654e984: Pull complete
97f5c0f51d43: Pull complete
c22eb46e871a: Pull complete
Digest: sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc19
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
orange@orange-VMware-Virtual-Platform:~$ docker run -d -p 80:80 --name my-nginx
nginx
5ce0d49ec707c6a47bf268c81dec0acc52db8bb1f6de5544142200d86ba738be
orange@orange-VMware-Virtual-Platform:~$
```

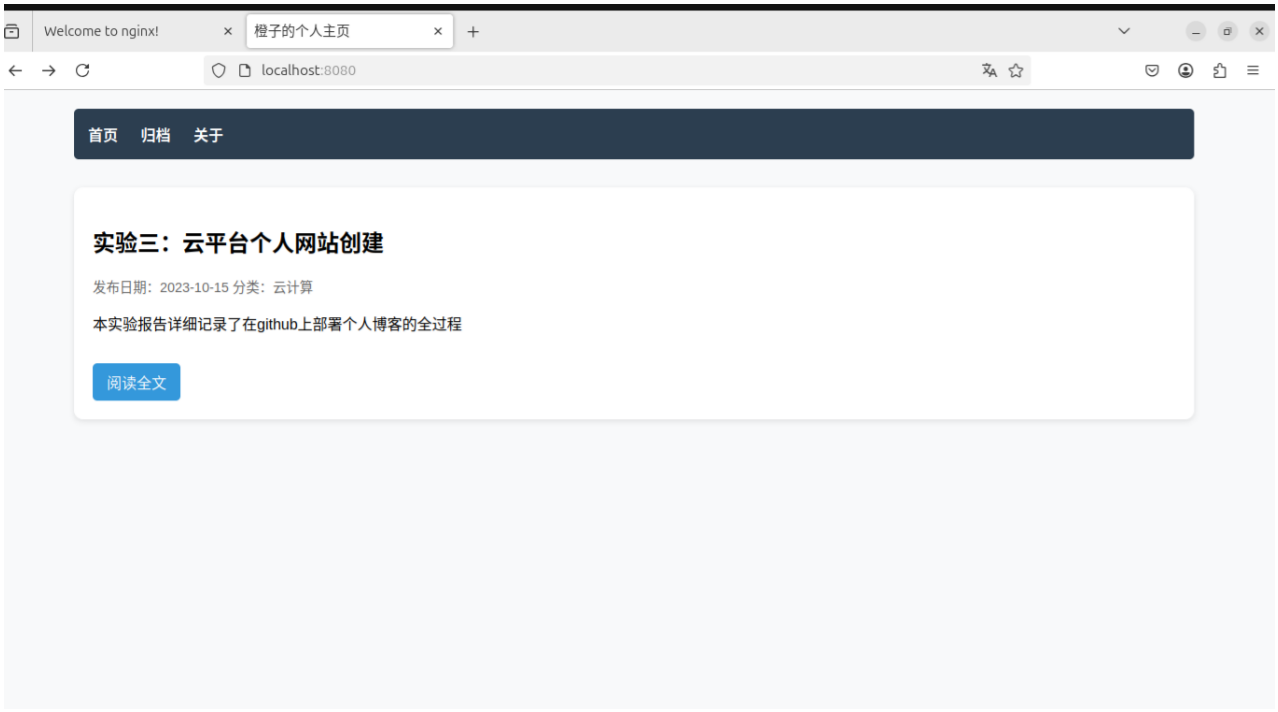


5. 复制个人博客的 html 到文件夹下，配置 dockerfile，构建镜像并运行


```
# 使用 Nginx 官方镜像
FROM nginx:alpine

# 删除默认的欢迎页面（可选）
RUN rm -rf /usr/share/nginx/html/*

# 将本地文件复制到 Nginx 的 Web 根目录
COPY index.html /usr/share/nginx/html/

# 暴露 80 端口
EXPOSE 80

# Nginx 默认已启动，无需 CMD 指令
```

```
orange@orange-VMware-Virtual-Platform:~/docker-test$ docker build -t my-static-site .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon   5.12kB
Step 1/4 : FROM nginx:alpine
 ---> 1ff4bb4faebc
Step 2/4 : RUN rm -rf /usr/share/nginx/html/*
 ---> Using cache
 ---> 385111d72e7a
Step 3/4 : COPY index.html /usr/share/nginx/html/
 ---> Using cache
 ---> 98d3151a91ad
Step 4/4 : EXPOSE 80
 ---> Running in be2f19a437d5
 ---> Removed intermediate container be2f19a437d5
 ---> e319e5b5fce4
Successfully built e319e5b5fce4
Successfully tagged my-static-site:latest
orange@orange-VMware-Virtual-Platform:~/docker-test$ docker run -d -p 8080:80 --name my-site my-static-site
26542bbaac0d869f143c2bf290bc8eecfd93a8dd55602fe42c78c8a309c89697
orange@orange-VMware-Virtual-Platform:~/docker-test$
```

7. 访问 http://localhost:8080 进行验证



结论分析与体会：

1. 了解和对比官方和自己构建的镜像文件

```
orange@orange-VMware-Virtual-Platform:~/docker-test$ docker images
REPOSITORY       TAG          IMAGE ID       CREATED          SIZE
my-static-site   latest       e319e5b5fce4   4 minutes ago    47.9MB
<none>           <none>       b18d492ef65d   58 minutes ago   118MB
nginx            latest       53a18edff809   7 weeks ago      192MB
nginx            alpine       1ff4bb4faebc   7 weeks ago      47.9MB
```

镜像名称	TAG	大小	说明
nginx (官方)	latest	192MB	官方默认镜像，包含完整功能模块。
nginx (官方)	alpine	47.9MB	基于 Alpine Linux，轻量精简。
my-static-site	latest	47.9MB	基于 nginx:alpine，体积与官方 Alpine 镜像一致。

对比项	官方镜像（**nginx:alpine**）	自定义镜像（**my-static-site**）
镜像体积	47.9MB	48.1MB（增加 HTML/CSS 文件）

层级数量	5 层	4 层（合并 COPY 步骤）
安全性	非 root 用户运行	继承官方配置
用途	通用 Web 服务器	专用静态网站托管

my-static-site 镜像与 nginx:alpine 体积相差很小，说明未引入冗余文件，符合轻量化预期。且自定义镜像层级更少，说明构建过程简洁。

2. 实验遇到的问题

（1）权限不足问题：

- 现象：docker run 时提示 permission denied。
- 解决：重新加入 docker 组并刷新权限（newgrp docker）

（2）pull 超时：

- 解决：配置镜像源时添加多个可用镜像源，docker 会自动选择最合适的

3. 实验总结

（1）成果：

- 成功部署 Nginx 服务并运行自定义静态网站。
- 掌握了镜像构建、容器管理和权限配置的核心技能。

（2）改进方向：

- 探索多阶段构建优化镜像体积。
- 结合 Docker Compose 实现多容器编排。

（3）Docker 的应用价值：

- 实现开发、测试、生产环境的一致性。
- 快速部署微服务架构，提升运维效率。