# Docker Concepts Lab Report

## 1. Introduction

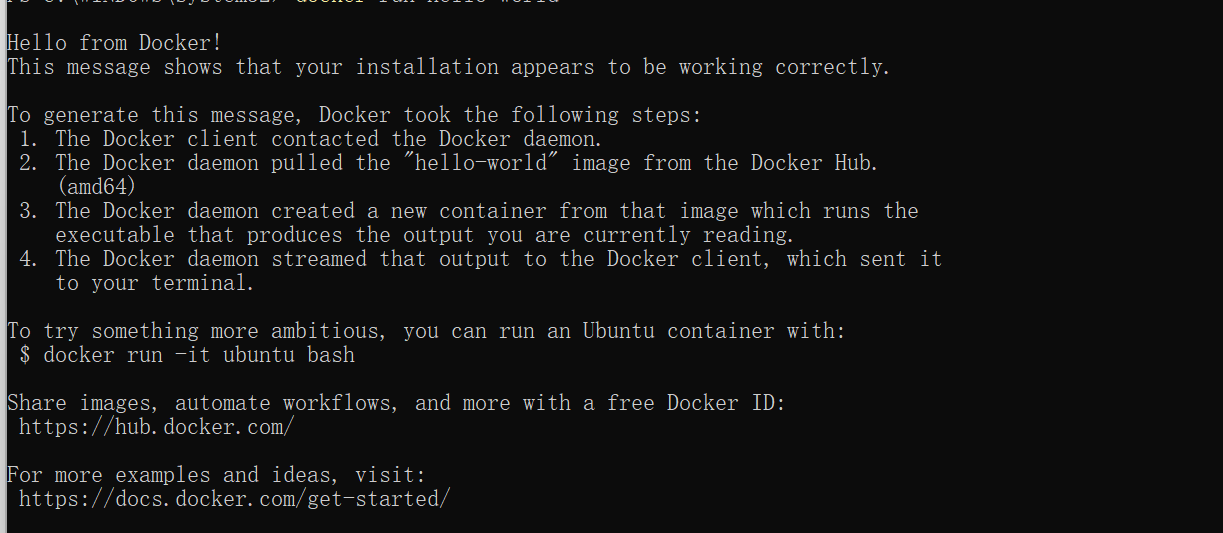
This lab introduces the core concepts of Docker, including containers, images, ports, Docker Compose, and data persistence. The goal is to gain practical experience in running and managing containers, publishing ports, sharing local files, and deploying multi-container applications.

## 2. Environment Setup

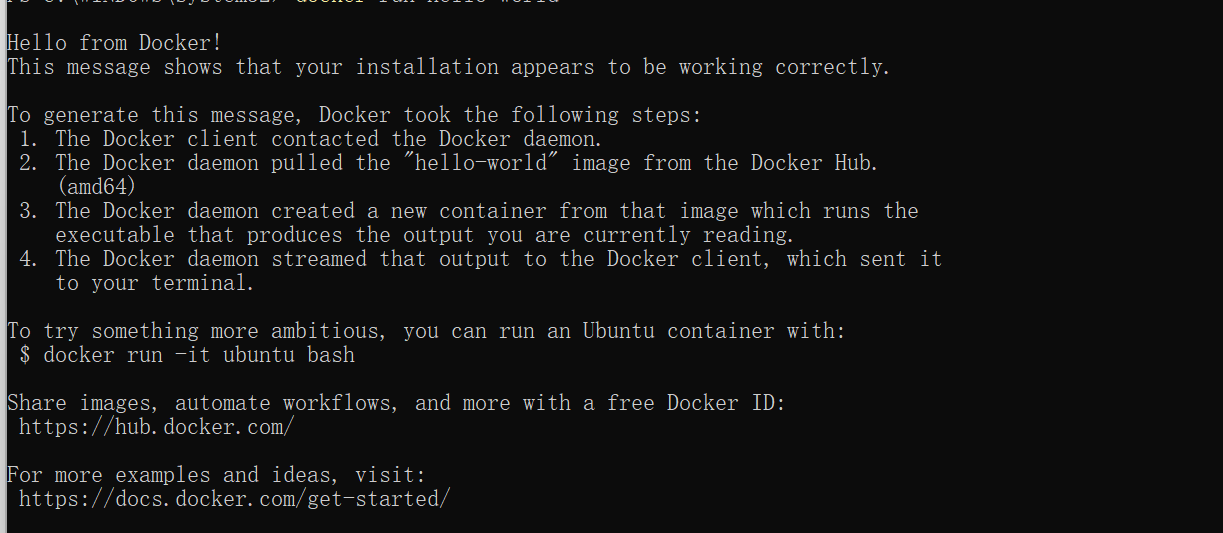
Operating System: Windows 11  
Docker Desktop: Latest version  
WSL 2: Enabled  
Tools: Docker CLI, Visual Studio Code, PowerShell/Command Prompt

## 3. Containers and Images

Command: docker run hello-world

Note: Containers are lightweight, isolated environments that package an application and all its dependencies together.

Command: docker pull nginx  
docker images

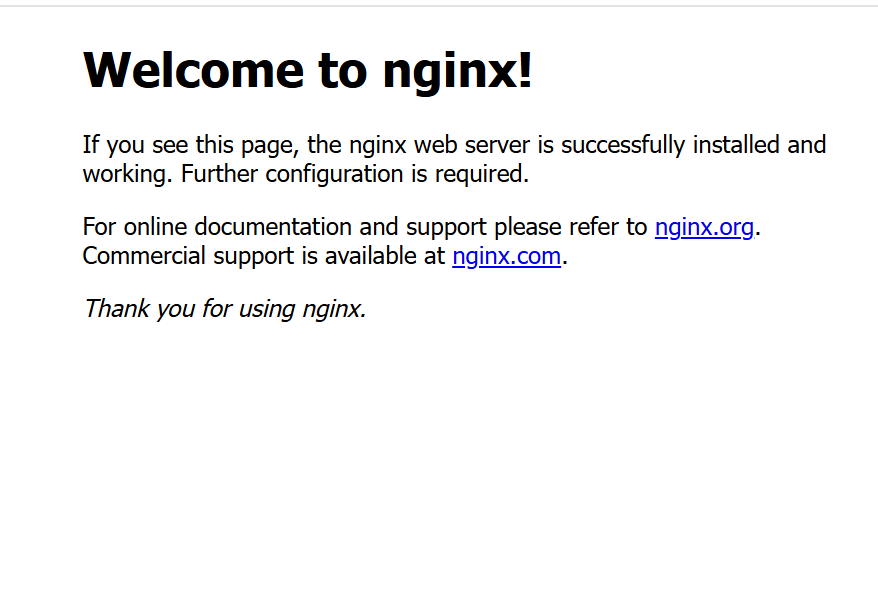
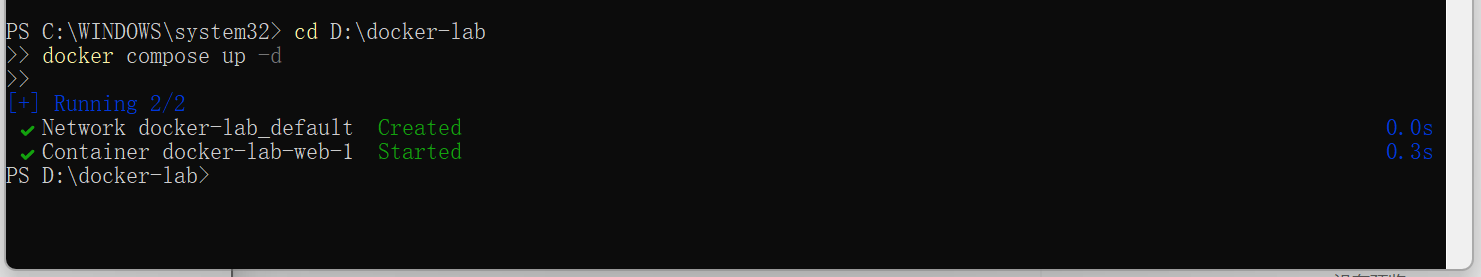
Note: Images are templates used to create containers.

## 4. Docker Compose and Port Publishing

docker-compose.yml:

services:  
 web:  
 image: nginx  
 ports:  
 - '8080:80'

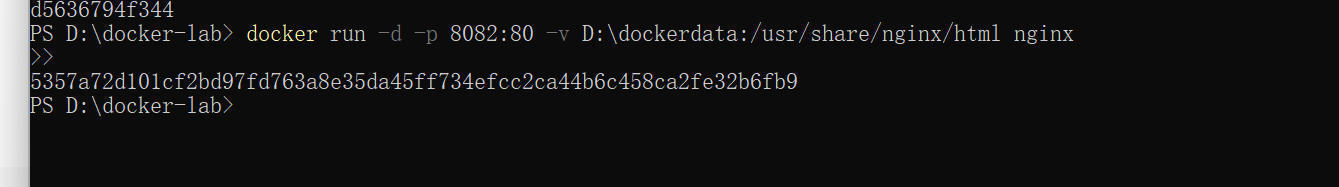
Command: docker compose up -d

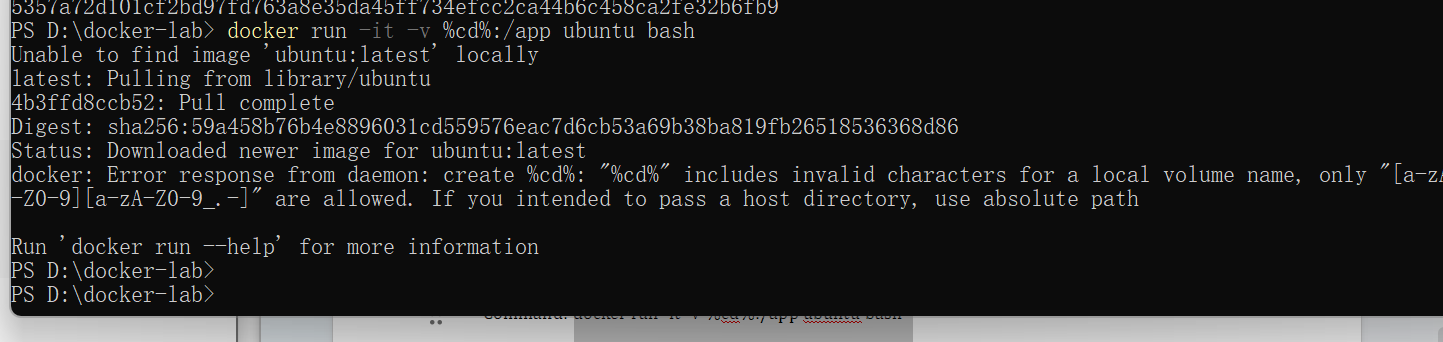


Note: Docker Compose simplifies running multi-container applications by defining them in a YAML file.

## 5. Persisting and Sharing Data

Command: docker run -d -p 8082:80 -v D:\dockerdata:/usr/share/nginx/html nginx

Command: docker run -it -v %cd%:/app ubuntu bash

Note: Volumes and bind mounts allow persistent storage and real-time file sharing between host and container.

## 6. Multi-Container Applications

docker-compose.yml:  
services:  
 web:  
 image: nginx  
 ports:  
 - '8083:80'  
 db:  
 image: mysql  
 environment:  
 MYSQL\_ROOT\_PASSWORD: root

Command: docker compose up -d

## 7. Conclusion

Through this lab, I learned how Docker containers and images work, how to expose ports, and how to persist data. I also gained practical experience using Docker Compose to run multi-container environments.