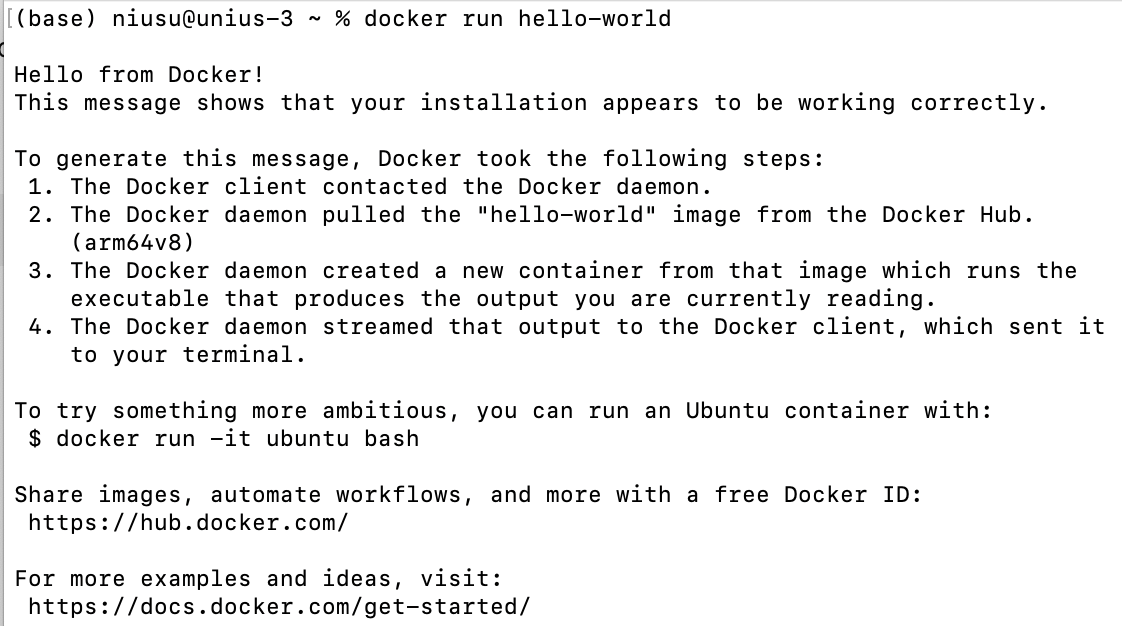
1. What is a container

**Command:**

docker run hello-world

**Note:**

A container is an isolated runtime environment that packages an application with all its dependencies.  
It ensures that software runs the same way across different systems.



***Figure1: Docker container***

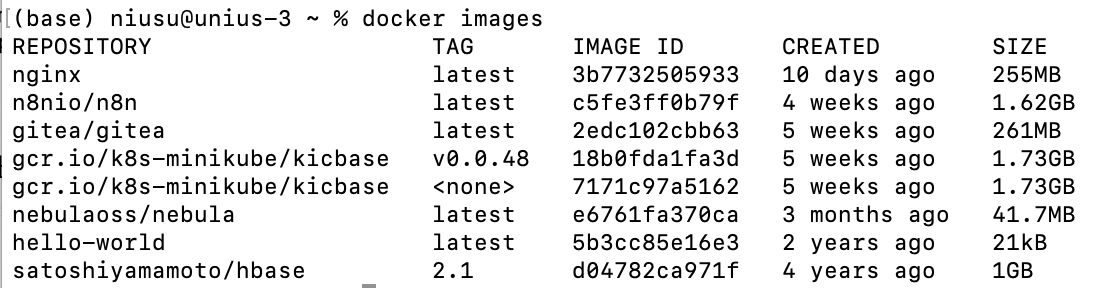
2. What is an image

**Command:**

docker images

**Note:**

A Docker image is a read-only template used to create containers.  
It contains the application code, runtime, libraries, and environment variables.



***Figure2: Docker images***

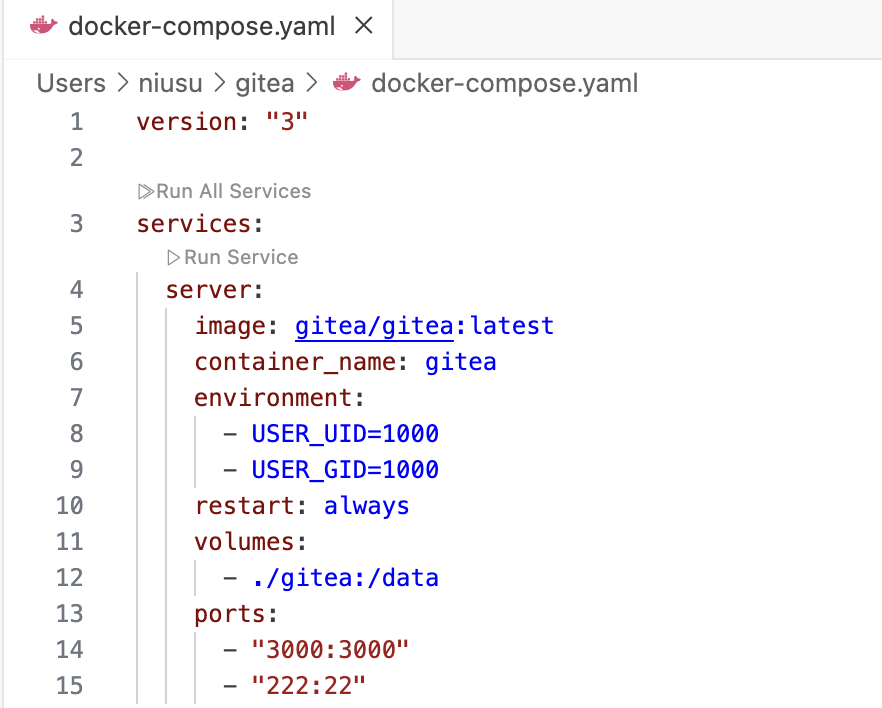
3. What-is-docker-compose

**Command:**

docker compose up -d

**Note:**

Docker Compose simplifies multi-container management.  
It allows developers to define and run multiple services in one file.



***Figure3: Docker compose***

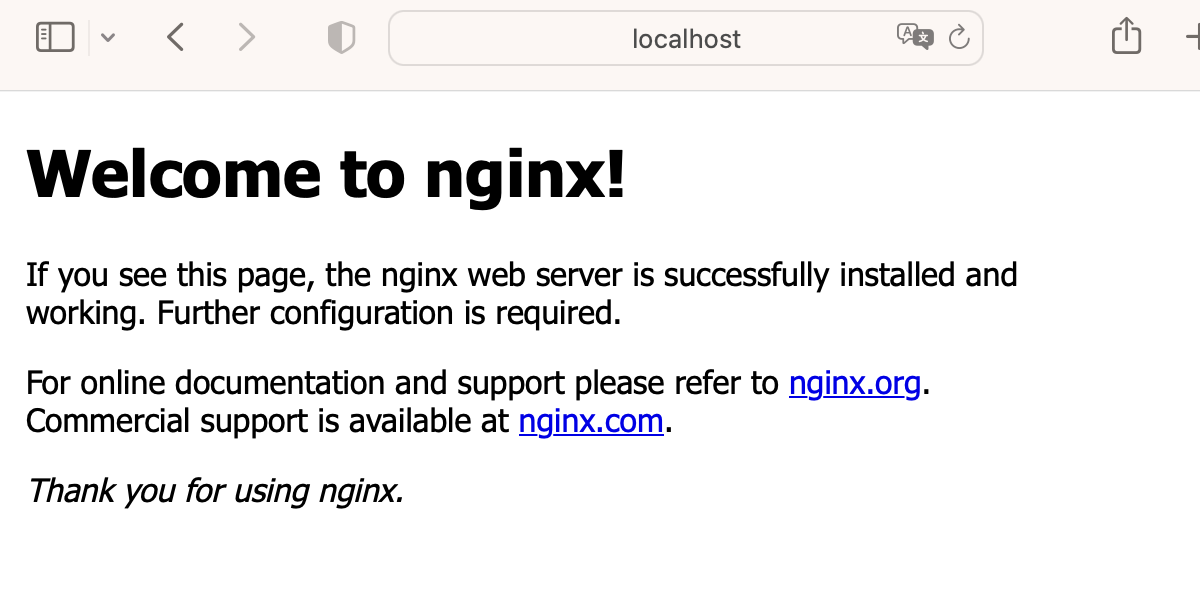
4. Publishing-ports

**Command:**

docker run -d -p 8080:80 nginx

**Note:**

The container’s internal port 80 is mapped to host port 8080.  
When visiting http://localhost:8080, the same Nginx page is visible.



***Figure4: Publishing ports***

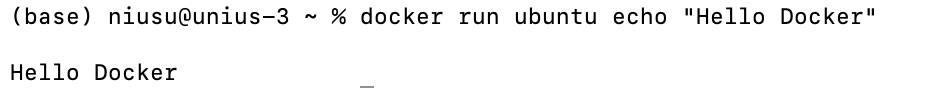
5. Overriding-container-defaults

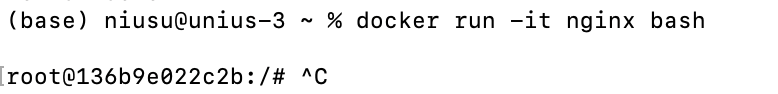
**Command:**

docker run ubuntu echo "Hello Docker"/ docker run -it nginx bash

**Note:**

This shows that containers can override the default startup command using custom commands or arguments.





***Figure5: Overriding-container-defaults***

6. Persisting-container-data

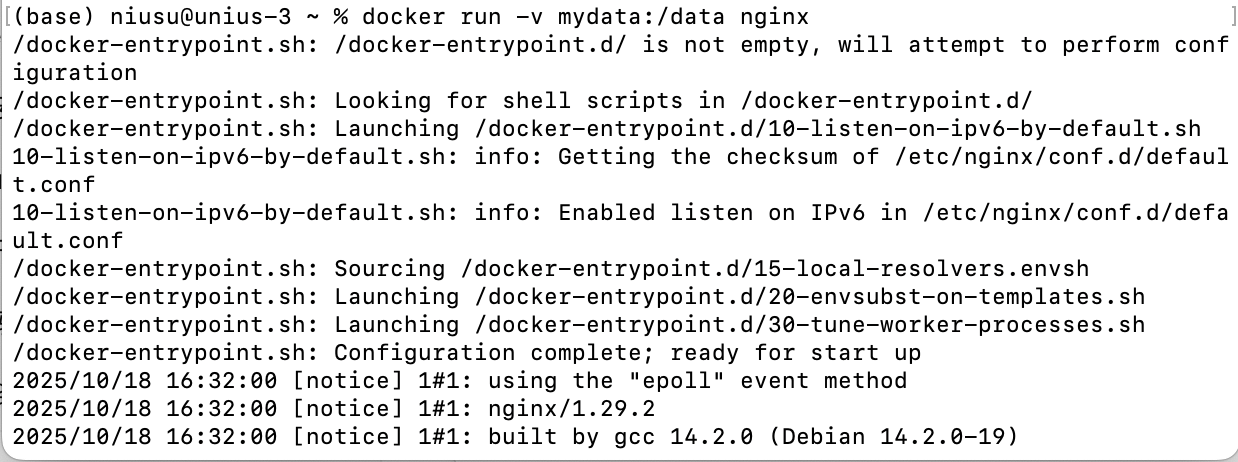
**Command:**

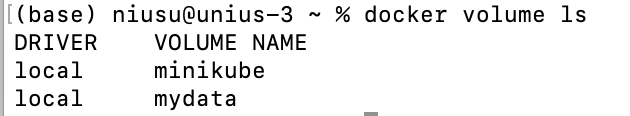
docker run -v mydata:/data nginx

docker volume ls

**Note:**

Volumes allow persistent storage so that data isn’t lost when a container stops or restarts.





***Figure6: Overriding-container-defaults***

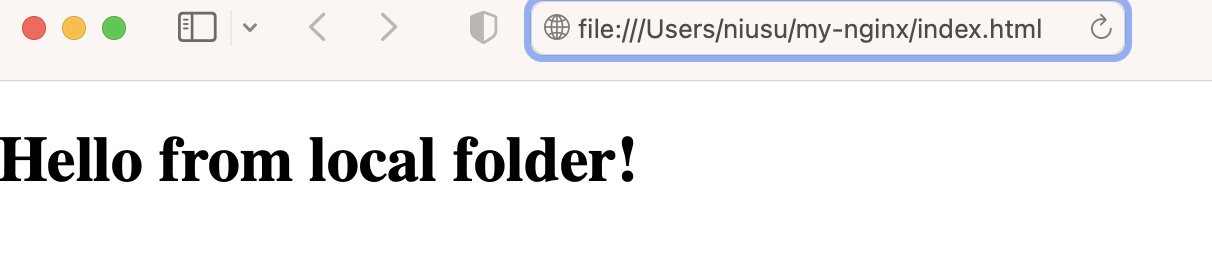
7. Sharing-local-files

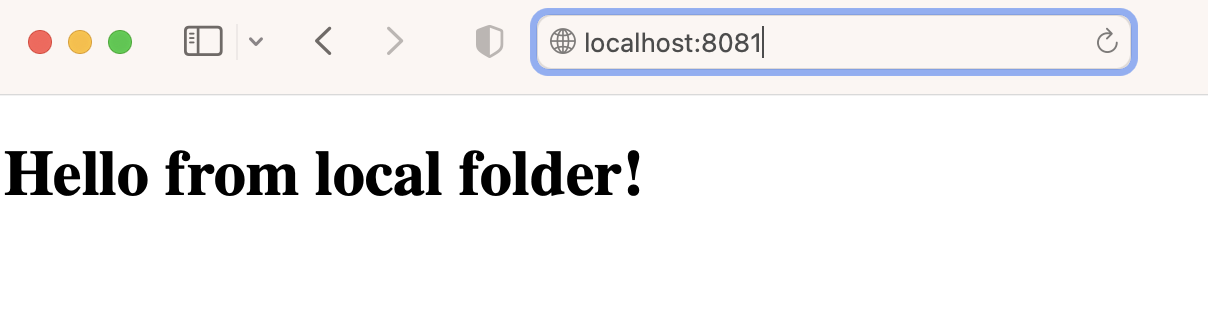
**Command:**

docker run -d -p 8081:80 -v $(pwd):/usr/share/nginx/html nginx

**Note:**

This demonstrates bind mounts, allowing developers to share files between the host and container in real time.





***Figure7: Sharing-local-files***

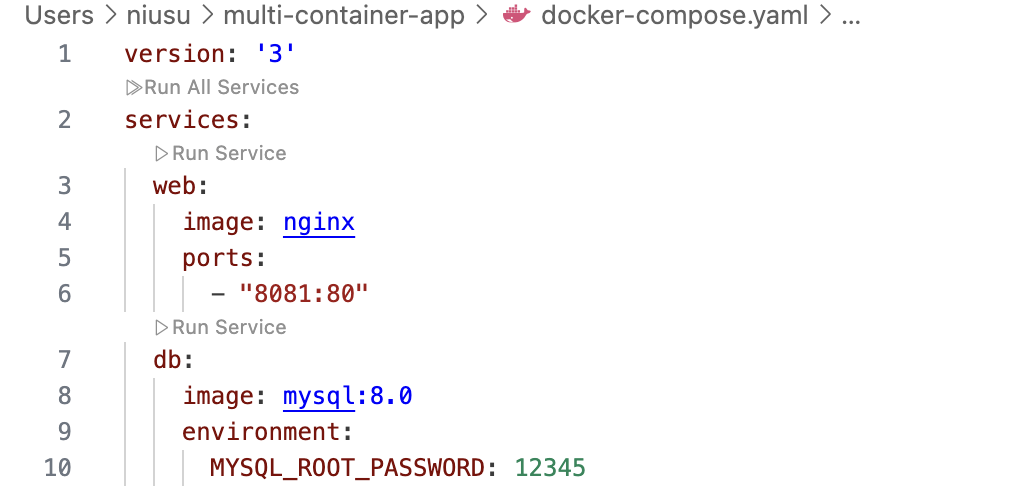
8. Multi-container-applications

**Command:**

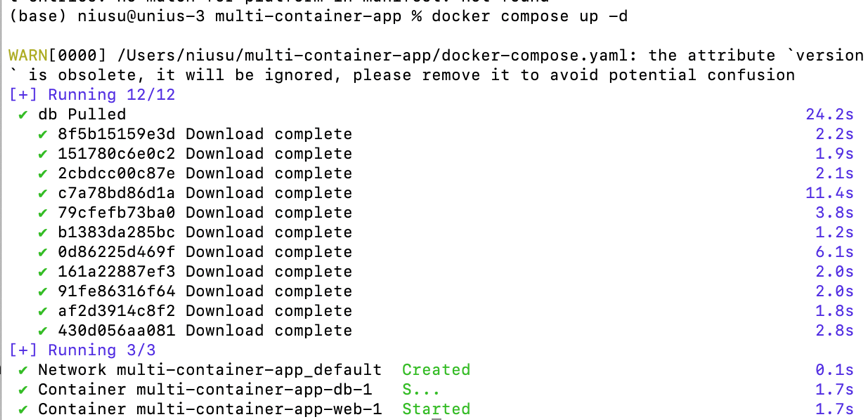
docker compose up -d

**Note:**

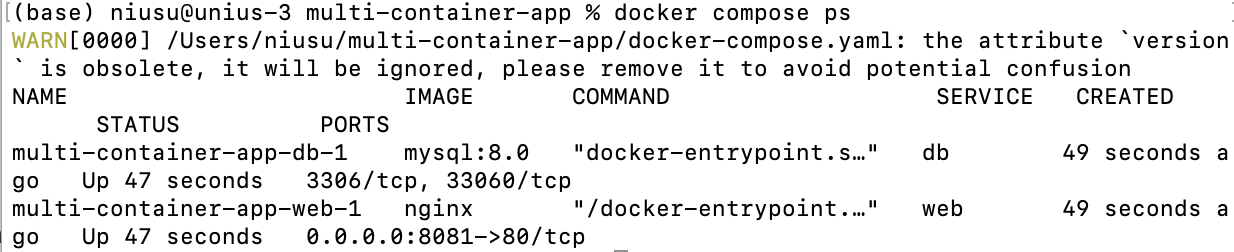
A multi-container application allows different services (frontend, backend, database) to work together through Compose.

****

***Figure8: Create compose file***



***Figure9: Start container***



***Figure10: Check container status***