Docker Assignment Steps

### 1. Objective

Write a Dockerfile that includes all major Dockerfile instructions and prints a message when the container starts. Make the application accessible from outside the container. Use both bind mounts and Docker volumes to store data, then verify which data remains after deleting the container.

### 2. Dockerfile

# Base image  
FROM ubuntu:22.04  
  
# Maintainer info  
LABEL maintainer="yourname@example.com"  
  
# Environment variable  
ENV APP\_DIR=/app  
  
# Arguments  
ARG APP\_VERSION=1.0  
  
# Create app directory  
RUN mkdir -p $APP\_DIR  
  
# Set working directory  
WORKDIR $APP\_DIR  
  
# Copy application files  
COPY ./app /app  
  
# Expose port to make container accessible  
EXPOSE 8080  
  
# Volume declaration (for persistent data)  
VOLUME ["/data"]  
  
# Entry point (runs when container starts)  
ENTRYPOINT ["bash", "-c", "echo 'Container started! Version: $APP\_VERSION'; exec bash"]  
  
# Default command (optional)  
CMD ["echo", "Hello from Docker!"]

### 3. Steps to Run Container

**Step 1: Create Docker volume**

docker volume create mydata

**Step 2: Run container with bind mount and volume**

docker run -it --name my\_app \  
 -v /host/data:/host\_data \ # Bind mount  
 -v mydata:/data \ # Docker volume  
 -p 8080:8080 \  
 my\_app\_image

**Step 3: Test data persistence inside container**

# Inside container  
echo "Hello Bind Mount" > /host\_data/file1.txt  
echo "Hello Volume" > /data/file2.txt

**Step 4: Delete container**

docker rm -f my\_app

**Step 5: Verify data persistence**

# Bind mount data on host  
ls /host/data # file1.txt will remain  
  
# Docker volume data  
docker run -it -v mydata:/data ubuntu ls /data # file2.txt will remain

### 4. Persistence Table

| Storage Type | Location | Data after container deletion |
| --- | --- | --- |
| Bind mount | Host directory (/host/data) | ✅ Remains on host |
| Docker volume | Managed by Docker (/data) | ✅ Remains in volume |
| Container internal | /app or other container paths | ❌ Lost |

### 5. Diagram: Bind Mount vs Volume Persistence

Container Deleted  
┌─────────────┐  
│ Container │  
│ ─────────── │  
│ /app │ <- lost after delete  
│ /data │ <- Docker volume, persists  
│ /host\_data │ <- Bind mount, persists on host  
└─────────────┘

### Notes

* Bind mounts store data on the host machine, so it persists even if the container is removed.
* Docker volumes are managed by Docker and also persist beyond container lifecycle.
* Any data created inside the container without a volume or bind mount will be lost when the container is deleted.

This document can be used for submission along with screenshots showing which data persisted after container deletion.