Step 1: Create a Python script

Create a file named file\_operations.py and add the following code to it. This script will read from and write to a text file named data.txt.

python

# file\_operations.py

# Read from the file

try:

with open("data.txt", "r") as file:

content = file.read()

print("Content read from file:")

print(content)

except FileNotFoundError:

print("File not found.")

# Write to the file

with open("data.txt", "a") as file:

new\_content = "\nNew line added"

file.write(new\_content)

print("Content written to file:", new\_content)

# Keep the container running

while True:

pass

Step 2: Create a Dockerfile

Create a file named Dockerfile (without any file extension) and add the following code to it. This Dockerfile defines the instructions for building the Docker image.

Dockerfile

# Dockerfile

FROM python:3.9

# Set the working directory inside the container

WORKDIR /app

# Copy the Python script and data file to the container

COPY file\_operations.py .

COPY data.txt .

# Define the command to run when the container starts

CMD ["python", "file\_operations.py"]

Step 3: Create a data.txt file

Create a file named data.txt in the same directory as the Dockerfile and file\_operations.py files. Add some content to this file.

Step 4: Build the Docker image

Open a terminal and navigate to the directory where the Dockerfile, file\_operations.py, and data.txt files are located. Run the following command to build the Docker image:

shell

docker build -t file-operations-image .

This command builds the Docker image using the Dockerfile and tags it with the name file-operations-image.

Step 5: Run the Docker container and keep it running

After building the Docker image, you can run a container based on it. Run the following command to launch the container and keep it running:

shell

docker run -d --name file-operations-container file-operations-image

This command runs the container in detached mode (-d) and assigns it a name (--name file-operations-container). The container will continue running until manually stopped.

Step 6: Start the stopped container in interactive mode

To execute commands inside the stopped container, we need to start it in interactive mode. Run the following command to start the container interactively:

shell

docker start -i file-operations-container

This command starts the container (docker start) and connects to its interactive mode (-i).

Step 7: Read the text file from inside the container

Once the container is running in interactive mode, you can run the following command inside the container to read the contents of the text file:

shell

cat data.txt

This command displays the contents of the data.txt file from within the container.

Step 8: Read the text file from the host system

To read the contents of the text file from the host system, run the following command:

shell

docker cp file-operations-container:/app/data.txt .

This command uses the docker cp command to copy the data.txt file from the running container to the current directory on the host system. You will find the data.txt file in the same directory as the Dockerfile, file\_operations.py, and data.txt files.

Now, you have the container running and can read the data.txt file from both inside and outside the container.