LAB 1: Running First Docker Container

Step 1: Setting Up the Project Directory

1. **Create a project directory**:

   ```
   mkdir my-docker-lab
   cd my-docker-lab
   ```

Step 2: Creating the Dockerfile

2. **Create a file named Dockerfile**:

   ```
   touch Dockerfile
   ```

3. **Open the Dockerfile in a text editor** (e.g., nano, vim, or any code editor of your choice):

   ```
   nano Dockerfile
   ```

4. **Add the following content to the Dockerfile**:

   ```
   # Use an official Ubuntu as a parent image
   FROM ubuntu:latest

   # Set the maintainer label (optional)
   LABEL maintainer="your-email@example.com"

   # Update the package repository and install Python
   RUN apt-get update && apt-get install -y python3

   # Set the working directory in the container
   WORKDIR /usr/src/app

   # Copy the current directory contents into the container at /usr/src/app
   COPY . .

   # Make port 80 available to the world outside this container
   EXPOSE 80

   # Define the command to run the application
   CMD ["python3", "-m", "http.server", "80"]
   ```

Step 3: Creating Content for the Container

5. **Create a simple HTML file to serve**:

   ```
   touch index.html
   ```

6. **Open the index.html file in a text editor** and add the following content:

   ```
   <!DOCTYPE html>
   <html lang="en">
   ```

```
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Welcome to Docker Lab</title>
</head>
<body>
   <h1>Welcome to Docker Lab</h1>
   <p>This is a simple web page served by a Docker container.</p>
</body>
</html>
```

Step 4: Building the Docker Image

7. **Build the Docker image from the Dockerfile**:

   ```
   docker build -t my-simple-webserver .
   ```

   Explanation:

   - `docker build`: Command to build an image from a Dockerfile.
   - `-t my-simple-webserver`: Tags the image with the name `my-simple-webserver`.
   - `.`: Specifies the current directory as the build context.

Step 5: Running the Docker Container

8. **Run a container from the image you just built**:

   ```
   docker run -d -p 80:80 --name my-webserver my-simple-webserver
   ```

   Explanation:

   - `-d`: Runs the container in detached mode (in the background).
   - `-p 80:80`: Maps port 80 on the host to port 80 in the container.
   - `--name my-webserver`: Names the container `my-webserver`.
   - `my-simple-webserver`: Specifies the image to use.