



Software Engineering

Lab7 – Docker

Docker Lab

Lab objective

In this lab, you will containerize a simple Flask web application using Docker. The app serves a basic HTML template and runs on port 5000.

By the end of this lab, you will:

- Understand the fundamentals of containerization with Docker
- Learn how to write a Dockerfile to define your application environment
- Use Docker commands to build and run a container
- Access a running containerized web app through your browser

Folder structure

```
/lab-starter-code
├── app.py
├── requirements.txt
└── templates/
    └── index.html
```

TODO:

1: Create the Dockerfile

Inside the `lab-starter-code` directory, edit the file named `Dockerfile` (no extension), and add the following instructions:

```
# Use an Python base image
FROM python:3.10-slim

# Set the working directory inside the container
WORKDIR /app

# Copy files into the container
COPY . .

# Install dependencies
RUN pip install -r requirements.txt

# Expose the port Flask runs on
EXPOSE 5000

# Set the default command
CMD ["python", "app.py"]
```

2: Build the Docker Image

Open a terminal or PowerShell, navigate to the `lab-starter-code/` directory, and run:

```
docker build -t lab-starter-code .
```

3: Run the Docker Container

Now run your Flask app in a container:

```
docker run -p 5000:5000 lab-starter-code
```

You should now be able to open your browser and visit:

<http://localhost:5000>

4: Stop & Clean Up

To stop the container, use `Ctrl+C` in the terminal.

To remove all containers and images:

```
docker ps -a          # find container ID
```

```
docker rm <ID>        # remove container
```

```
docker rmi lab-starter-code # remove the image
```

