EECS 348: Software Engineering, Spring 2025

Lab 11: Docker Programming

Task 1. Docker installation

Docker has different operating system versions, so you should download and install Docker with the correct version. You can not install Docker on the lab machine.

1. Visit the docker website and select the appropriate version for your operating system. You must make sure that you have the privilege to install Docker.
2. Follow the installation instructions for your operating system. The installation process may vary depending on your system, but generally, you will need to run an installer and accept the license agreement.
3. After the installation is complete, you can test that Docker is working by opening a terminal or command prompt and running the command docker version. This should display information about the Docker client and server.

Task 2. Dockerize a Web Application

Create a simple web page using HTML, CSS, JavaScript, and PHP (you can use what you have done for Lab 4). Suppose you save the file in the folder **lab11**.

Create a file named **Dockerfile** (use the exact file name) in the same **lab11** directory. Copy the following code into the file. Some commands contain dashes or underscores, which may be lost or misinterpreted when copying from a PDF file. To avoid errors, it's recommended to type the commands manually or carefully check them after copy-pasting.

```
FROM php:7.4-apache
COPY . /var/www/html/
EXPOSE 80
```

1. *FROM php:7.4-apache*: This line specifies the base image that we will use to build our new image. In this case, we are using the official PHP 7.4 image that comes with the Apache web server.
2. *COPY . /var/www/html/*: This line copies the contents of the current directory (the . symbol) to the /var/www/html/ directory inside the container. This is where Apache looks for files to serve over the web.
3. *EXPOSE 80*: This line tells Docker that our container will be listening on port 80, which is the default port for HTTP traffic (i.e., web traffic).

Open a terminal and navigate to the **lab11** directory. Run the following command to build the Docker image.

```
docker build -t webapp .
```

This command will build a Docker image named ”webapp” from the Dockerfile in the current directory. Run the following command to start a Docker container from the ”webapp” image.

```
docker run -d -p 8080:80 --name webapp_container webapp
```

This command will start a Docker container named ”webapp-container” from the ”webapp” image, map port 8080 on the host machine to port 80 in the container and run the container in detached mode.

Now, you can open a web browser and navigate to http://localhost:8080 to test the web application. You should be able to see the web page that you created.

To push the docker image to Docker Hub, you need to sign up for a Docker Hub account if you don't have one. After getting an account, sign in to the Docker. Run the following commands to push the Docker image to Docker Hub. Make sure to replace “username” with your Docker Hub username. The tag_name is optional, and the default value will be ”latest” if you don't assign it. If you don't want to assign it, you should also not type the colon before it. After pushing your Docker image, you can see it on your Docker Hub.

```
docker tag webapp username/webapp:tag_name
docker push username/webapp:tag_name
```

Task 3. Dockerize a C Program

You should have a C program for this task. You can write a new Hello World program or use the C program in Lab 7 to Lab 10.

Then, create a file named ”Dockerfile” in the same directory and copy the following code into the file. But the quotation mark around *./main* should be the normal English quotation mark, and you should correct it if you copy rather than type it.

```
FROM gcc:latest
COPY program.c .
RUN gcc -o main program.c
CMD [”./main”]
```

Open a terminal or command prompt and navigate to your directory. Run the following command to build the Docker image.

```
docker build -t cprogram .
```

Run the following command to start a Docker container from the **eclab2_cprogram** image.

```
docker run -it cprogram
```

Then, you also need to push this image to the Docker Hub using the following commands.

```
docker tag cprogram username/cprogram:tag_name
docker push username/cprogram:tag_name
```

Task 4. Publish to GitHub and Docker Hub

Create a repository in the folder **lab11**, and push all files to GitHub. Make sure you submit both links.