Lab 2: Docker

Learning objectives

By the end of this lab, students will be able to:

- Understand and explain key Docker concepts: images, containers, and registries.
- Create a Dockerfile to containerize a simple Node.js or Python web app.
- Build and tag Docker images using the CLI.
- Run containers and map ports between host and container.
- Push and pull images to and from Docker Hub.
- Use volumes and bind mounts for persistent storage.
- Define and deploy multi-container applications with Docker Compose.

Pre-requisites (check before starting)

- Docker Desktop installed and running (Windows, macOS, or Linux)
- Docker Compose installed (included with Docker Desktop).
- A Docker Hub account for image sharing.
- Node.js or Python installed locally for sample app (confirm with `node -v` or `python --version`).
- Text editor (e.g., VS Code) for editing `Dockerfile` and `docker-compose.yml`.
- Basic command-line skills (navigate folders, run commands).

Lab Instructions: Docker Workshop

In this lab, you will follow the **official Docker "Get Started" Workshop** available at:
🔗 https://docs.docker.com/get-started/workshop/

You will complete each section of the workshop and document your progress with screenshots and notes.
At the end, you will submit a report containing all required proof of work.

Tasks

Follow the steps from the workshop to:

Part 1: Containerize an application
Part 2: Update the application
Part 3: Share the application
Part 4: Persist the DB
Part 5: Use bind mounts
Part 6: Multi-container apps

Part 7: Use Docker Compose
Part 8: Image-building best practices
Part 9: What next

Submission Requirements

Submit a report that includes:

1. **Project files in a GitHub repository:**
   - Dockerfile(s)
   - docker-compose.yml
   - Application source code
   - lab-submission.txt (brief notes of what you learned in each step)
2. **Required screenshot:**
   - `docker build` completed successfully
   - `docker images` listing your image
   - `docker run` running your container and web app in a browser in both your local machine and on ***Play With Docker***
   - `docker push` output and your Docker Hub repo page
   - `docker ps` showing multiple containers (for Compose)
   - `docker volume ls` for persistence proof
   - Browser view of the working app after Docker Compose up