Homework #3 - Containerization with Podman

In this homework you will use podman to wrap applications into containers and run them. The process with Docker would be relatively simple and configuration file would differ not that much

Pre-setup: Podman installation

You need Podman and Podman Compose installed on your system.

Windows

The recommended way is to install Podman Desktop.

It also requires Windows Subsystem for Linux, if you do not have it yet (WSL). To install it, run Windows Terminal and execute

wsl --install

macOS

The recommended way is to use brew

brew install podman podman-compose

Task #1 - Wrap single application in container

For this task, you need to:

1. Create Dockerfile for your application. Create it in the root directory of your project.

2. Run podman build to create an image of your application.

3. Use podman run to run the application container with your image.

4. Make a test request to your service.

5. Check service logs.

6. Clean working environment.

Task #2 - Multi-container setup with a local network.

This task will give you skills to set up subnetworks with compose files, run several containers at the same time and allow them to “speak” with each other.

1. Create a second fastAPI application. Make sure it has the URL to the first service stored securely.

2. The logic of the second application should be plain simple: it should call the first one on a scheduled basis, let’s say every 10 seconds.

3. Create a compose file that describes both of your applications, for example

You can find an example here.

4. Run podman compose in the directory with compose file and according services.

5. Verify both containers started.

6. Verify that the scheduled container is able to access the main application.

7. Verify that the main application correctly processes input requests.

8. Do not forget to clean the working environment :)

Submission form

Please submit as GitHub link, it would be beneficial if it would be build on top of your application from HW#2.
If possible, submit both tasks in one repository.