« 3.2 Docker containers (/css/2022/containers/exerc... 3.4 Assignment: create a python build environment ...

CS-E4190 (/css/2022/) / 3. Software containers (/css/2022/containers/) / 3.3 Assignment: Hello Docker!

Assignment: Hello Docker!

This assignment introduces the basic concepts behind creating and running a container image with Docker. The goal is to create a container image by writing a simple Dockerfile.

Warning

The activities in this course are individual work. **Do not read or copy solutions from other students**. **Do not share solutions**. Remember that episodes of plagiarism and collusion are fraudulent means in studying according to the Aalto University Code of Academic Integrity (https://into.aalto.fi/display/ensaannot/Aalto+University+Code+of+Academic+Integrity+and+Handling+Violations+Thereof) which may result in caution or suspension. See also the code of conduct (https://mycourses.aalto.fi/mod/page/view.php?id=916404) of the course.

Tip

Carefully read the related instructions (https://mycourses.aalto.fi/mod/page/view.php?id=916406) before submitting the assignment(s).

Task

Your task is to create a Docker image that is able to successfully output a Fibonacci sequence until a certain number of terms as well as say hello to Docker.

Tip

You should first get familiar with Docker by following the related tutorials (https://mycourses.aalto.fi/course/view.php?id=37032&section=3#tutorials) in MyCourses.

Tip

If you have troubles installing Docker (Desktop), you could try using the Google Cloud Shell (https://cloud.google.com/shell/docs) to carry out the assignment. You can follow this guide (https://cloud.google.com/shell/docs/accessing-cloud-shell-with-gcloud) if you prefer using a terminal application installed locally instead of a web-based session on the Google Cloud Console (https://cloud.google.com/shell/docs/launching-cloud-shell#launching_from_the_console).

In particular, the container should write **0 1 1 2 3 5 8 13 21 34** from the Fibonacci sequence. The sequence is generated by **`this shell script`** (https://gitmanager.cs.aalto.fi/static/CS-E4190_2022Autumn/_downloads/fibonacci.sh), which obtains the number of terms to print through an environment variable called `VAR` .

The next task is to say hello to Docker, the container should write a specific text **Hello Docker!**, The shell script `helloDocker.sh` takes input through the environment variable called `MSG` and then simply just prints it **`as in this script`** (https://gitmanager.cs.aalto.fi/static/CS-E4190_2022Autumn/_downloads/helloDocker.sh).

Both the scripts must be copied into the container image.

Attention

The `fibonacci.sh` and `helloDocker.sh` files are available to the grader in the same directory of the Dockerfile, so you do not have to upload it somewhere beforehand to be able to access it from the Dockerfile.

Grading

Submission are evaluated by means of an automated system based on the Container Structure Tests (https://github.com/GoogleContainerTools/container-structure-test) tool. You only need to submit a single file, namely, the Dockerfile for the container image that satisfies the requirements above.

Attention

Make sure to test your Dockerfile locally before submitting it. This is especially useful to obtain detailed error messages in case of issues.

Make sure that your Dockerfile fulfills the following requirements.

- The base image **must** be alpine:3.12 (https://hub.docker.com/layers/library/alpine/3.12/images/sha256-cb64bbe7fa613666c234e1090e91427314ee18ec6420e9426cf4e7f314056813?context=explore).
- It specifies `/usr/app` as the working directory.
- It includes a `VAR` environment variable appropriately set to a value that fulfills the task above.
- It includes an `MSG` environment variable appropriately set to a value that fulfills the task above.
- It assumes that the command to be executed is named `fibonacci.sh`.
- `helloDocker.sh` need not be executed directly by the container as grader will execute this automatically.

This course has already ended.

Note

The assignment runs multiple unit tests which give fractional points based on how the requirements in task are fulfilled, according to the table below.

Test	Points
The scripts exist in the container and the working directory is correct	8
The `VAR` and `MSG` environment variables exist and its values are correctly set	8
The scripts are successfully executed and produces the intended outputs	8

⚠ The deadline for the assignment has passed (Wednesday, 26 October 2022, 14:30).

Docker Assignment: Hello Docker

Upload the Dockerfile

Upload your Dockerfile as the solution.

Dockerfile

Choose File No file chosen

Submit

« 3.2 Docker containers (/css/2022/containers/exerc...

3.4 Assignment: create a python build environment ...