CS-548: Cloud-native Software Architectures

Computer Science Department • University of Crete
Instructor: Antony Chazapis

Website: https://www.csd.uoc.gr/~hy548/

Version: 202501

Assignment 1: Docker

1. Nginx is one of the most popular web servers. Container images are readily available on Docker Hub. Write down the commands needed to:
    a. Download the images tagged `1.27.4` and `1.27.4-alpine` locally.
    b. Compare the sizes of the two images.
    c. Start `nginx:1.27.4` in the background, with the appropriate network settings to forward port 80 locally at port 8000 and use a browser (or `curl` or `wget`) to see that calls are answered. What is the answer?
    d. Confirm that the container is running in Docker.
    e. Get the logs of the running container.
    f. Stop the running container.
    g. Start the stopped container.
    h. Stop the container and remove it from Docker.

2. Following the previous exercise, start the Nginx container again and issue the commands needed to:
    a. Open a shell session inside the running container and change the first sentence of the default page to "Welcome to MY nginx!". Close the session. Validate the change.
    b. From your computer's terminal (outside the container) download the default page locally and upload another one in its place. Validate the change.
    c. Close the container, delete it and start another instance. Do you see the changes? Why?

    Create a simple HTML page in a local folder and issue the commands needed to:
    d. Start an Nginx container to serve the page from the local folder instead of the default page. Validate that the correct content is served.

3. An example of a very simple Django application is available in the course's repository on GitHub (https://github.com/chazapis/hy548/django). The code is based on the tutorial included in the documentation (https://docs.djangoproject.com/en/5.1/intro/tutorial01/). You are encouraged to follow the tutorial to get acquainted with the commands necessary for the following steps. Write down the commands needed to start a Python container, get a shell environment into it, copy the files in, install the necessary software (listed in `requirements.txt`), initialize the application, start it, validate that it works.

4. Following the previous exercise, create your own container image, based on `python:3.13.2`, that will contain the Django application and run it when started. Make sure that the state (the database) is kept in a separate volume. In your new image, also include vi, provided by package `vim-tiny`, in case someone needs to inspect or edit a file in a running container (note: Django needs `PYTHONUNBUFFERED=1` in the environment when run non-interactively in a Python container).
    a. Provide the updated codebase of the application, including the Dockerfile. Briefly mention any changes in the code or new files that were created (*i.e.*, scripts, ignore lists).
    b. Explain how much bigger your own image is compared to the image you were based on. Why? What have you done in the Dockerfile to keep the image as small as possible?
    c. Create a Docker Hub account and upload the image. Provide the command needed to upload the image to Docker Hub.
    d. Issue the commands needed to start the same container twice, one with "debug mode" enabled and one without. Validate that both are running and serving different content.

5. Now that you have the Django application with the corresponding Dockerfile in your GitHub repository and a Docker Hub account, create a GitHub Action that will automatically build and push the image (the workflow should be initiated by the user). Provide the YAML of the workflow you made.

Notes:

- The assignment is personal.
- All exercises contribute equally to the overall grade (unless individual percentages are defined).
- A day/time will be set for answering questions and giving clarifications.
- Write down your answers in a Markdown-formatted text file in either Greek or English and commit it (along with any other files) in a private GitHub repository before the exercise's deadline. Share the repository with the instructor (username "chazapis").