# Advanced Software Engineering - Lab test

January 17th, 2024

NAME: _____ SURNAME: _____ ID (MATRICOLA): _____
ORAL EXAM TODAY OR TOMORROW (IF YOU PASS)? Y: ☐ N: ☐
ORAL EXAM FOR CONTINUOUS ASSESSMENT? Y: ☐ N: ☐
YOUR CODE (Needed in some exercises): 123456

## Instructions

It is mandatory to put the number (YOUR CODE) of the header in the first line of the *solutions.txt* file. A missing or misspelt code is considered an automatic fail.

For example, if your code is 123456 the first line of the *solution.txt* must be:

Your code =:= 123456

You will have 60 minutes to deliver your solution. The deadline is strict, in the course Moodle you will not be able to upload anything after the deadline. If you deliver the solution, you must also give back this sheet.

You can access the slides of the course from the Moodle and the documentation website of PIP (`https://pip.pypa.io/en/stable/`), Docker (`https://docs.docker.com/`), Bandit (`https://bandit.readthedocs.io/en/latest/`) and Locust (`https://docs.locust.io/en/stable/`).

Any other material (your old code included), website or application (generative AI included, e.g. Copilot) consulted will result in an automatic fail of the test.

You have to download the zip file of the test and upload an archive file with the requested files as a solution before the deadline.

## Delivery - 6 points needed to pass

Put all the files (also the unmodified ones) in a `.zip`, `.tar`, or `.tar.gz` file and upload it on the Moodle delivery. You can avoid uploading the cache folders created during the execution. The name of the file must be `name_surname` indicating your name and your surname.

**The solutions will be automatically evaluated** by an offline script, so be careful to not modify the structure of files to fill. Add only the answers to the exercises and do not insert new lines, comments, or anything else. To pass this test you need to reach 6 points. Every exercise but the `Dockerfile` one, will grant points for partial solutions.

## Material description

Download *material.zip* file from the Moodle.

It contains the following:

- folder *code*: folder containing code and files of a multi-service application (do not modify);
- *docker-compose.yml*, the docker compose file you have to complete;
- *solutions.txt*: text file you have to fill with answers, add only text after the =:= symbols without adding new lines, escape symbols, or quotes;
- *locustfile.py*: the file containing the performance tests for locust (do not modify).

# 1 Dockerfile (3 points)

In the extracted root folder, create and write the `Dockerfile` to make the following image:

- based on the image `python:3.9.18-slim`,
- put the content of the *code* folder inside a folder called *YOUR_CODE* where YOUR_CODE is the number in the exam header,
- make that folder the working directory,
- install Bandit with `pip`,
- set the command to run Bandit on the working directory as the initial command of the container.

Do not add unnecessary commands, e.g. `apt-get update`. Do not create a directory with `mkdir`.

# 2 Docker commands and Bandit (4 points)

In the *solution.txt* write the commands (without `sudo`) to

- create the docker image calling it as `aselab`,
- run the container based on such an image and calling it `cbandit`,
- check the log of the last 11 lines of the output of Bandit via docker.

In the same file, answer the two questions about the output of Bandit.

# 3 Docker compose and Locust(3 points)

Complete the *docker-compose.yml* file by adding the service called `frontend` and make it reachable from port 5005 (look in the frontend Dockerfile which port you have to bind).

Run the application with docker compose. In a different terminal, run `locust` and use a web browser to reach locust's web service to run performance tests on the application. Analyse the statistics and fill the file *solutions.txt* with the requested results (wait at least 10 seconds to collect significant statistics).

# 4 Extra (1 point)

Perform an HTTP GET to the frontend endpoint `/secret?X=code`, where `code` is YOUR CODE of the exam header. Look for the **cookie** named `secret` in the HTTP response and put its value in the *solution.txt* file (without any quotes) answering the *Secret* question.