

2808ICT/7623ICT
Secure Development Operations
Assignment 1

Containerize Full-stack Projects

Due Date: 23:59 on 15th September
Total Marks: 100

Group Assignment
Weighting: 30%

Rationale:

The aim of this assignment is that you show your understanding of the theory presented in lectures and the practical exercises conducted in workshops about using containers for the development and deployment of software projects.

Teamwork:

This assignment is a group assessment. Each group should have 2 or 3 students, and you cannot mix undergrads and postgrads in a group.

Each group member must be involved in completing or refining the solution to each task. The workload should be evenly distributed, and by default, each member receives the same mark. If there is a concern about work contribution, please contact the teaching team for mark adjustment.

Tasks:

This assignment has three main tasks. The first task is about building images and running docker containers individually. The second task requires you to configure and run the whole project using docker-compose. The third task requires you to configure and run the whole project using Kubernetes.

In this assignment, you can either use the full-stack software project provided via the links below or find a web application project from GitHub. Either way, you must use a single project for all the tasks in this assignment.

If you use the provided software project, you can download them here:

- 2808ICT: <http://formal-analysis.com/tmp/pizzeria.zip>
- 7623ICT: <http://formal-analysis.com/tmp/nodejs-shop.zip>

If you find a project by yourself, it must be from GitHub and satisfy the following conditions:

- The project must include a front-end Web application and a back-end database, and allow you to run the four containers (including a graphical interface for the database and a Nginx proxy) described in Task 1.
- The project must run correctly and allow users to use the front-end Web application to add, update, and delete data in the database, which is reflected in the front-end application.
- The GitHub project itself does not provide any Dockerfile nor yaml files for containerizing the project.
- It must be a public GitHub project and you need to provide the link to it.
- The version of the project cannot be later than July 2023.

For 7623ICT students

You will need to do some research and write one additional section on the benefits of Containers and related technologies in software development operations (DevOps). In particular, you need to

- briefly describe the technologies you used in the assignment,
- discuss how they fit into the bigger picture of DevOps, e.g., who will use them at what stages of the software development and deployment,
- what are the major benefits of these technologies compared to more traditional ways of software development

Submission:

- 1) *Each team* should submit *one* report to describe the process and your results. In particular, the report must include
 - the student numbers and names of team members;
 - the content of all the files you created, including the Dockerfiles and yaml files;
 - sufficient screenshots of the build processes showing what you have done and the results;
 - a section with details of the self-learning resources you have used, such as online forums, articles, videos, and ChatGPT, including examples of questions/ keywords you used
 - a section named Revision History (using the table below); and

Date of Change	Contributor	Summary of Change

- a list of references with proper formatting.
- 2) *Each team* should submit *one* video (no longer than 10 minutes, compressed to max 100 MB) where each team member takes turns to describe what they did in the assignment and demonstrate the results.
 - Each member describes their name and contributions (less than 1 minute), and demonstrates the results, i.e., explain how you know it is working (around 2 minutes).
 - Make sure the video and sound can play, otherwise, it will be considered unsubmitted.
 - The goal of the video is to demonstrate your understanding and the quality of your work to help the evaluation of your work.

Task 1. Run the Project with Individual Containers (30 Marks)

For this task, you must create docker images (when necessary) and containers that will allow you to build and run the full-stack project, including at least four docker containers:

- one that runs the front-end Web application,
- one that runs the back-end database,
- one that runs a graphical interface for the back-end database (such as Mongo Express), and
- one that runs an Nginx proxy to allow secure HTTPS connections for the project.

Note that in Task 1, you are to run docker containers without using docker-compose. You should demonstrate that your containers are working together.

Task 2. Run the Project using Docker Compose (30 Marks)

For the second task, you must create a docker-compose.yml file that will allow you to automate the construction of the full-stack project services. This task can be completed independently from Task 1.

The docker-compose.yml need to control at least four containers as described in Task 1 to run the project. There must be internal docker networks, exposed ports and volumes defined. You must show that you can configure containers through environment variables. You should demonstrate that your containers work well together.

Task 3. Run the Project using Minikube and Kubectl (30 Marks)

For the second task, you must create a Kubernetes cluster with one node with Minikube that will allow you to automate the construction of the full-stack project services. This task can be completed independently from Tasks 1 and 2.

The cluster needs to contain at least four containers as described in Task 1 to run the project. There must be internal and external services with exposed ports for pods' communication and external access to the application. You must show that you can configure containers through environment variables. You should demonstrate that your containers work well together.