Assessment Brief

Subject Code and Name	DEV1004 - DevOps
Assessment Number	1
Assessment Title	Containerise an Existing Application
Assessment Type	Individual Programming Project
Words, Size or Duration	1 programming project
Subject Learning Outcomes	SLO2, SLO3
Submission Date / Time	Due by 11:55pm AEST Sunday end of Module 2.
Weighting	20%

Assessment Purpose

Modern software development projects need to be as available and reliable as possible, and there's a variety of technologies available to achieve those goals. Containerisation is a technique that helps a developer—regardless of their working environment—create applications that run in a specified, common container environment that can easily scale applications to support extremely dynamic usage demands with minimal fuss.

To solidify your knowledge of modern software development concepts and show your ability to work with containers at a fundamental level, you should be able to write code to appropriately containerise an application.

Assessment Task / Item

For this assessment, you must submit an application that meets the design and programming requirements to showcase your skills as a software developer.

Assessment Instructions

Using a containerisation platform, write build files in a format appropriate to that platform that:

1. Contains instructions to containerise an existing application suitable for a development environment.
2. Executes an automated build of a container image when run by the platform.

Project requirements:

- Complete a planning stage before developing the application, which requires the development of these items:
    - An application architecture diagram (AAD) that represents the application and its relation to any services or infrastructure that contribute to the standard execution of the application.
    - An explanation of the AAD, including justification for its components and explanation of how the components interact with each other.

- Use an appropriate “ignore” file for any containerisation process (such as “.dockerignore”)

Design requirements

The build files must:

- Create container images with a consistent naming and tagging scheme that includes appropriate data about the images.
- Appropriately set environment variables for the containerised application. At a minimum, specify an environment for the application (such as development, testing, or production).

Programming requirements

The build files must:

- Be syntactically correct and execute successfully
- Use DRY coding principles
- Facilitate the application working as intended
- Implement basic cyber security techniques such as environment variables or secret variables

Submission

All work must be submitted via Canvas, in the assignments section appropriate to this brief. Please ensure the above mentioned submission date and/or time are adhered to, or penalties may apply.

When submitting your work, please save your files using the naming convention below.

[Student_ID]_[Surname]_[First Name]_[SubjectCode]_[Assessment_#].zip

E.g.: **1234_Singh_Visha_PRG1002_Assessment_01.zip**

For more information on late submission, please see the **Assessment Policy**.

Academic Integrity

The integrity of the assessment process is fundamental for ensuring appropriate evaluation at AIT. All work submitted should be your own, and where additional resources are used, they must be referenced according to the Harvard style. Additionally, TurnItIn is available in the LMS to test plagiarism in your writing.

For more information on academic integrity, please see the **Academic Integrity** and **Academic Integrity Penalties Policies**.

Appeals

Fair application of the assessment rubric, rules and guidelines should be administered for each assessment. If you feel an evaluation requires further consideration, you may be entitled to an appeal.

For more information on your right to an appeal, please see the **Assessment Appeals Procedure and Policy.**

Policies

For access to the policies mentioned above and related to education at AIT, please see the footer of the AIT website, and follow the link named **Education Policies and Procedures.**

Website: https://www.ait.edu.au

Assessment Rubric

Task Descriptor	**(HD)** **High Distinction** *(85-100%)*	**(D)** **Distinction** *(75-84%)*	**(C)** **Credit** *(65-74%)*	**(P)** **Pass** *(50-64%)*	**(F)** **Fail** *(0-49%)*
DEVELOPS suitable application architecture diagrams to represent an application **20%** SLO 2	MULTIPLE application architecture diagrams provided, representing ALL appropriate pieces of an application's architecture.	MULTIPLE application architecture diagrams provided, representing MOST appropriate pieces of an application's architecture.	ONE application architecture diagram provided, representing ALL appropriate pieces of an application's architecture.	ONE application architecture diagram provided, representing MOST appropriate pieces of an application's architecture.	Diagram either not provided or not depicting any or much relevant appropriate data about the application's architecture.
EXPLAINS an application architecture diagram and its components **30%** SLO 2	EXTENSIVE explanation of ALL components of an application architecture diagram & how the components interact with each other.	MODERATELY-DETAILED explanation of ALL components of an application architecture diagram & how the components interact with each other.	EXTENSIVE explanation of SOME components of an application architecture diagram & how the components interact with each other.	MODERATELY-DETAILED explanation of SOME components of an application architecture diagram & how the components interact with each other.	Explanation provided is either completely incorrect, not containing relevant details, or not provided at all.
DEVELOPS semantically and syntactically valid & complete containerisation files. **10%** SLO 3	Implements at least one containerisation files or help files which are COMPLETELY semantically & syntactically valid.	Implements at least one containerisation files or help files which are ALMOST COMPLETELY semantically & syntactically valid.	Implements at least one containerisation files or help files which are MOSTLY semantically & syntactically valid.	Implements at least one containerisation files or help files which are SOMEWHAT semantically & syntactically valid.	The relevant containerisation files either do not exist or are almost entirely semantically & syntactically invalid.
CREATES appropriate files or commands to optimise a build or versioning process **10%** SLO 3	Meets D criteria, with the relevant "ignore" files using optimised strings & techniques to minimise file content.	Meets C criteria, with the relevant "ignore" files themselves referring to only files or folders found within the project.	Uses version control "ignore"-type files to optimise versioning, and uses containerisation "ignore"-type files to optimise the build process.	Uses version control "ignore"-type files to optimise versioning.	Fails to use any "ignore"-type files in a project, or uses syntactically-invalid "ignore" files.
CREATES appropriate files to configure a secure containerised application with valid environment & other settings **10%** SLO 3	The containerisation files configure an application using secrets configured by a continuous integration/continuous delivery (CI/CD) platform or cloud platform in	The containerisation files configure an application with appropriate arguments, environment variables, and secrets.	The containerisation files configure an application with appropriate environment variables and make appropriate use of arguments in combination with	The containerisation files configure an application with appropriate environment variables.	The containerisation files do not configure an application with any relevant data.

	combination with typical containerization file arguments, environment variables & secrets.		environment variables.		
GENERATES consistent & appropriate container image names & tags that includes relevant data about the image **10%** SLO 3	Application container images are named using a naming & tagging scheme that contains relevant data about the application's versioning, environment, version control , and container registry.	Application container images are named using a naming & tagging scheme that contains some relevant data about the application's versioning, environment, and version control information.	Application container images are named using a naming & tagging scheme that contains some relevant data about the application's versioning.	Application container images are given a specific name and/or follow a basic naming scheme, but the naming scheme doesn't contain much appropriate or relevant data about the image.	Application container images follow no consistent naming scheme, or use default names.
CREATES appropriate build files that facilitate the application working as intended **10%** SLO 3	Meets D criteria, plus the containerised app has working images for production environments.	Meets C criteria, plus the containerised app has working images for development and testing environments.	Meets P criteria, plus the containerised app runs with zero errors overall.	The containerised application runs in a development environment with zero errors that don't occur when the app runs outside of a container.	The containerised application runs with errors that don't occur when the app runs outside of a container.