Introduction

The assessment of the Cloud DevOpsSec module is based on two assessment components:

a) *Project* which represents 60% of the module assessment, and
b) *Terminal Exam* which represents 40% of the module assessment.

The learning outcomes of the Cloud DevOpsSec module are as follows:

LO1. Critically analyse different techniques to perform code analysis, plan and implement static code analysis.
LO2. Develop and implement a plan for provisioning and configuration of software applications through CI/CD.
LO3. Critically evaluate and implement methodologies for secure application development and secure execution in production.

The Project assesses LO1, LO2, and LO3.

Project Description

For this assignment, you are required to develop a dynamic cloud-based application. You are required to build, deploy, monitor, and update your web application by setting up a full lifecycle CI/CD pipeline. Your application must be deployed and hosted on a *public cloud provider*.

Your dynamic web-based application should:

- Accept input from the user and validate that input
- Provide CRUD functionalities
- Use a suitable data storage solution (i.e., based on the requirements of your application)
- Deploy your application to a suitable public cloud platform. The deployed application must not be modified after the submission deadline. The examiner should be able to view your deployed application without having to register for any account with the public cloud provider where you deployed your application (i.e., the application **(not its source code!)** should be publicly accessible via a URL). This publicly accessible URL should not be shared with anyone else but the examiners of this instance of the Cloud DevOpsSec module at NCI.

You must conduct some independent research and include any relevant bibliography in the accompanying report.

Please refrain from sharing or reusing code from other students or external stakeholders unless otherwise instructed.

On completion, you will document the process and reflect on it through the deliverables listed in the next section.

Project Deliverables

You are required to document the process of developing the cloud-based application and the CI/CD pipeline set up, and reflect on it through the following deliverables:

1. A **project report** (7-8 pages formatted using the IEEE Conference double-column template[1]) which should include:
   - NCI Project Submission Sheet/ Project Cover Sheet
   - Headline: title of the report, your name, student number, module, programme, and date
   - Abstract – a 150-250-word executive summary of the project and the main results
   - Section 1: Introduction – motivation for your project and its main objectives. A brief description of your application.

[1] https://www.overleaf.com/read/bxnmxqvgkcnb

- Section 2: Continuous integration, continuous delivery and deployment of your application
  - Document the CI/CD pipeline including a diagram of the entire workflow; the diagram should indicate where the different tools/cloud-based services are used to support the CI/CD pipeline, and the relationships between these services/tools. Ensure that you distinguish between the CI and CD in the diagram. The diagram should be fully explained in text. Note that the diagram should be created by you based on the pipeline and stages you set up for your project.
  - Include the URL to your deployed application
  - Document the CI/CD pipeline in action – document how a change in the source code of your application flows through the pipeline. Ensure that you include relevant screenshots of the pipeline you set up for your project that capture that code change flow.

    Note that at all times you **must use** a **private repository** for versioning control (e.g., GitHub, AWS CodeCommit, etc.)
- Section 3: Critically analyse and document the approach you took for performing static code analysis, including security vulnerabilities analysis. Document your findings from performing static code analysis and security vulnerabilities analysis, together with their fixes.
- Section 4: Conclusions including findings/interpretations – what did you learn and find out? Include a short reflection on developing this project. If you were to implement this project again, what would you do differently?
- Section 5: References – a complete list of academic works and/or online materials used in the project. References should be included as in-text citations using the IEEE referencing style. Note that a good starting point to find academic works is the NCI Library Guide on Cloud Computing at https://libguides.ncirl.ie/cloudcomputing

Note that the report should include for all the previously mentioned elements demonstration snippets and/or screenshots of the commands and tools used and findings from static code analysis and security vulnerability analysis, where appropriate.

IMPORTANT: Anything after Page 8 will not be considered i.e., it WILL NOT BE MARKED! (Note that the NCI Project Submission Sheet/ Project Cover Sheet is not considered as part of the page count.)

2. The **source code artefacts** submission (packed as a compressed ZIP file) should include:
   - Source code of the solution (includes commented source code of the application together with any scripts used for automation and configuration)

     Note: Please include substantial meaningful comments in YOUR source code to document your ORIGINAL contributions

3. **Project presentation and demonstration, to be held in class during the submission week as per the CA schedule. It should include the following:**
   - A concise presentation of the motivation and high-level description of the project.
   - Demonstration – give a demonstration of your project highlighting the main features, which includes a demonstration of the CI/CD pipeline by making a change in one of the features of your application (i.e., modify its source code) and show how the change flows through the CI/CD pipeline.
   - Duration: Maximum 4 minutes, every 30 seconds over 4 minutes will incur a penalty of 20%
   - As part of the project presentation and demonstration you will be required to answer questions

   Notes:

   1. **You are required to submit the project presentation slide deck together with all the other deliverables by the submission deadline**
   2. **You must present/demo your project in order to be marked 'Present' for this project. Those who do not will be marked *'Absent without permission'* and the other deliverables will NOT BE GRADED.**

Assessment Criteria

The Project will be assessed based on the assessment criteria shown in Table 1 and marking rubric shown in Table 2.

Table 1 Assessment Criteria

The CI/CD pipeline	30%
Static code analysis, including security vulnerability analysis	35%
Conclusions and findings & References	10%
Project Demonstration & Answers to Questions	25%

Table 2 Marking Rubric

Grade Criterion	H1 (> 70%)	60-69%	50-59%	40-49%	Fail (< 40%)
The CI/CD pipeline: 30%	Excellent/very good application of CI/CD pipeline design and development in terms of appropriate methodology and tools/cloud-based services. Excellent/very good discussion and documentation of the CI/CD pipeline. Excellent/very good demonstration of implemented CRUD operations and input validation in the deployed application.	Good application of CI/CD pipeline design and development in terms of appropriate methodology and tools/cloud-based services. Good discussion and documentation of the CI/CD pipeline. Good demonstration of implemented CRUD operations and input validation in the deployed application.	Adequate application of CI/CD pipeline design and development in terms of appropriate methodology and tools/cloud-based services. Adequate discussion and documentation of the CI/CD pipeline. Adequate demonstration of implemented CRUD operations and input validation in the deployed application.	Weak application of CI/CD pipeline design and development in terms of appropriate methodology and tools/cloud-based services. Weak discussion and documentation of the CI/CD pipeline. Weak demonstration of implemented CRUD operations and input validation in the deployed application.	Poor application of CI/CD pipeline design and development in terms of appropriate methodology and tools/cloud-based services. Poor discussion and documentation of the CI/CD pipeline. Poor/Inexistent demonstration of implemented CRUD operations and input validation in the deployed application.
Static code analysis, including security vulnerability analysis: 35%	Excellent/very good critical analysis of substantive and relevant findings that incorporate both findings from static analysis and security vulnerability analysis.	Good critical analysis of substantive and relevant findings that incorporate both findings from static analysis and security vulnerability analysis.	Adequate critical analysis of relevant findings that incorporate findings from static analysis and/or security vulnerability analysis.	Limited critical analysis of relevant findings that incorporate findings from static analysis and/or security vulnerability analysis.	Very limited and poor/inexistent critical analysis of relevant findings that incorporate findings from static analysis and/or security vulnerability analysis.
Conclusions and Findings & References: 10%	Excellent/very good conclusions, insightful findings and reflection. References are complete, consistent, appropriately, and correctly used.	Good conclusions and good discussion of findings and reflection. Most references are complete, consistent, appropriately and correctly used.	Adequate conclusions and adequate discussion of findings and reflection. Adequate, consistent and appropriate referencing.	Limited/weak conclusions and limited/weak discussion of findings and reflection. References are few and/or mostly incomplete and/or inconsistent and/or inappropriately used.	Very limited and poor/inexistent conclusions. Very limited and poor/inexistent discussion of findings and reflection. References (if any) are incomplete and/or inconsistent and/or inappropriately used.
Project Demonstration & Answers to Questions: 25%	Excellent well directed presentation and demonstration with impeccable handling of questions.	Clear presentation and demonstration with good handling of questions.	Neat oral presentation and demonstration and acceptable handling of questions.	Poor oral presentation and demonstration and weak handling of questions.	Unacceptable oral presentation and demonstration and poor handling of questions.