

School of Technology, Business and Arts  
University of Suffolk

MSc Computer Science

Level: 7

Module: DevOps

Assessment: Case Study  
Module Leader: Dr. Godwin Dzvapatsva / Dr. Kakia Chatsiou  
Weighting in Module: **Total 100%** (component 2: **60%**)  
Handed out: August 2025  
Deadline: **on or before 12 noon Friday 19<sup>th</sup> of September 2025**  
Submission via Brightspace module pages at

**Required:**

a) a **3,000-word case study report** providing a critical evaluation identifying appropriate DevOps techniques and tools to support a DevOps implementation approach for a project. You are given a specific scenario to study. [See Case Study](#) in this document for a full brief. Your report to include source code, data extracts, diagrams as appropriate to support the report.

| Component number | Form of assessment | Assessment size | Weighting (%) | Learning outcomes assessed | Late Sub <sup>n</sup> | Core or non-core |
|------------------|--------------------|-----------------|---------------|----------------------------|-----------------------|------------------|
| 2                | Case Study Report  | 3,000 words     | 60%           | 1 ,2 and 3                 | Yes                   | Core             |

**Learning outcomes to be assessed:**

1. A systematic and comprehensive knowledge and understanding of the DevOps approach and techniques
2. ability to critically evaluate DevOp techniques for appropriateness in a particular setting
3. skill in using a variety of industry-standard software tools used to implement DevOps

**Assessment & Grading Criteria:**

1. The ability to choose from appropriate DevOps techniques and tools, and articulate any major limitations
2. The ability to design a DevOps pipeline
3. The ability to implement aspects of the designed DevOps solution
4. The ability to draw some relevant conclusions and reflect on any major limitations on implementation and design.

See also the attached assessment criteria for Level 7.

## Case Study: Implementing DevOps in E-commerce Application Development

### Background

Ipswich Retail, a mid-sized e-commerce company selling some products online, currently uses a monolithic application to manage its online retail operations. They have a small team of developers and a small operations team who are responsible for the current application used by the company. However, the monolithic architecture has led to numerous issues, including slow deployment cycles, lengthy bug fixes, and frequent downtime during updates. The application downtime has had customers complains almost once every week and this especially happens whenever there are new features added on the application. Maintenance of the application seems to be a challenge for the developers themselves. Further, the development and operations teams are independent, resulting in friction and delays in releasing features.

### Problem Statement

The current application is slow to adapt to new market demands and has not been able to scale with the growing user base. Bugs and new feature requests pile up, and the process of updating the system involves considerable manual work. Each deployment cycle requires extensive downtime, impacting the user experience and the business's reputation. In addition, the disjointed efforts of the development and operations teams lead to slower delivery times and inconsistent application performance. The company wants to replace its current system with a modern, Model-View-Template Django e-commerce application, developed using DevOps principles to foster better collaboration between development and operations and achieve improved efficiency and customer experience.

### Objectives

Ipswich Retail requires a Proof-of-Concept (PoC) solution for a new e-commerce application developed using the **Django framework** (+/-4 pages excluding admin page) and guided by DevOps principles. This PoC should address the challenges faced by the monolithic architecture and how implementing model-view-template will address the problem, continuous integration and deployment (CI/CD), and automated testing.

You can choose the type of products you would like to have on the site. Also feel free to use templates. Check on CodePen for most of your frontend designs.

### The solution should demonstrate how you addressed DevOps concepts such as:

1. **Planning-** In this phase, you will plan the design of your frontend (Mockup, Wireframe) and the backend architecture. (Some common tools to use include Ms Visio, Draw.io, Lucidchart etc.). You can also mention team composition being mindful of the requirements.
2. **Modular Architecture:** Develop the application with a Model-View-Template architecture using Django to increase scalability and flexibility.
3. **Automated CI/CD Pipeline:** Implement a CI/CD pipeline that facilitates regular, automated builds, testing, and deployment. This should include tools like Jenkins, GitHub Actions, **OR** GitLab CI/CD.
4. **Containerization:** Use Docker to containerize services, ensuring consistent environments across development, staging, and production.

5. **Continuous Delivery (CD)** - Ensuring code can be deployed to production quickly and safely with each code change.
6. **Monitoring and Logging:** Integrate monitoring tools such as Prometheus and Grafana to track application performance, uptime, and resource usage. You can also implement tracking using GitHub Issues, incorporate applications such as Slack to monitor any deploys, pull/merge requests just to mention a few.
7. **Automated Testing:** Incorporate unit tests, integration tests, and end-to-end tests within the CI/CD pipeline to ensure reliable deployments.
8. **Version Control:** Use Git and GitHub/GitLab/BitBucket to manage the source code and configuration files, with branching strategies to support collaborative development. Also link with communication tools such as Slack or Teams

**NB You can use Render.com, Supabase, Railway.com, PythonAnywhere, DigitalOcean for deployment etc. Please use open-source platforms-You must not pay for any resources.**

## Required

In this assignment, you will be challenged to take up the role of a DevOps expert working closely with an organisation to plan, develop and deploy an e-commerce application implementing DevOps practices and tools as a solution to the existing challenge which Ipswich retail is facing. You are then asked to compile report of no more than 3000 words outlining the following:

- a) Discuss the need of adopting DevOps in response to the current system's inefficiencies. Address how DevOps principles can break down silos between the development and operations teams, enabling faster and more reliable deployments.
- b) Development of a PoC e-commerce solution using Django with a M-V-T architecture, showcasing best practices in code modularity, CI/CD, containerization, and infrastructure automation.
- c) Provide a DevOps workflow plan detailing the tools, practices, and stages they implemented in the pipeline (e.g., code commit, build, test, deploy, monitor). (Include screenshots and notes that you recorded as you developed the PoC)
- d) Summarizing the implementation, lessons learned, and challenges faced, with recommendations for full-scale deployment of the PoC.

**SUBMISSION INSTRUCTIONS:**

- a. Please submit your reports and code (screenshots and GitHub links) as appropriate.
- b. If your report exceeds the word count by up to 10% then there will be no penalty applied. Submissions that exceed the word count by more than 10% will be applied a fixed penalty of 5 percentage points (i.e., 5 marks). In all cases, the penalised mark will not be reduced below a pass level, assuming the work merits a pass. Tables, diagrams (including associated legends), appendices, reference lists, tables of contents, footnotes, and endnotes are excluded from the word count however should be used appropriately. It is for the Module Leader to decide if there is an excessive or inappropriate use of components excluded from the word count.
- c. Ensure that your work (submitted electronically via the online submission portal) is bundled into a suitable file (Word, pdf,) with the filename matching the pattern **sXXXXXX-DevOps\_II** where sXXXXXX is your UoS user\_id.
- d. You must include the **GitHub link** for source code and **live link** for the deployed app in your report with the username and password. The GitHub links must be public and functional.
- e. Submit your document to the Brightspace module for this course (link at the top of this document) under 'Assessment' > 'Submission Folder for Report' or similar. Please note that this folder will become available a couple of weeks prior to the deadline.
- f. Note that whilst you will need to be as thorough in terms of your research as with any academic piece, the target reader of this report (especially the case study report) is a chief executive or manager of the company, so remember to adjust your tone accordingly.
- g. All bibliographies must be formatted according to the University of Suffolk Harvard Style. More information about citation and referencing is available here:  
<https://libguides.uos.ac.uk/academic/referencing/Harvard>
- h. Cite your references carefully – remember that the University of Suffolk has strict rules dealing with plagiarism and the university reserves the right to call any student to a viva examination of any piece of assessed work.
- i. The coursework must be your own individual work and not copied (**avoid plagiarism or use of large language models e.g., ChatGPT**)

**Notes:**

- What we are more interested in seeing in the report is your practice, your experience building the pipeline, your reflections and thoughts of the process. Take screenshots as you are implementing the project, especially if you are configuring a part of the DevOps process.
- Make sure you demonstrate advanced, critically justified DevOps practices throughout your report.
- Feel free to compare with other tools that you have used in your practice or have tried during the module this term. Remember that open-source tools are not the best for everything - if you know of a better way of creating that pipeline, say so. This could be for example on aspects of scalability, open source, flexibility and interoperability etc.

**TENTATIVE REPORT STRUCTURE**

1. Title Page - including your S number **not your name**.
2. Introduction:
  - A short overview of the e-commerce project. You will need to discuss why you have opted for DevOps in delivering your solution
3. DevOps workflow: An explanation and justification of the DevOps workflow followed. You can split the sections of DevOps from planning to maintenance. Make sure you highlight and discuss the principles and the associated tools implemented. You can also discuss the alternatives. You should also consider:
  - Make sure you include screenshots of the steps/tools for the project.[Include only key screenshots]
  - your thoughts/experience building that pipeline: what went ok, what went wrong
  - diagrams of the DevOps optimised application
4. Reflection on the solution implemented and future work@ An honest, reflective account of the overall process. Make sure to include
  - benefits and limitations of using the open-source software as the main tools for building your pipeline
  - comparisons or references to alternative tools that you could have included if necessary
  - Key challenges to DevOps as a practice
  - Future of DevOps in software development
5. References
  - include them for all the works that you have cited in your report. Make sure to cite references which you have consulted.

## Grading Grid – School of Technology, Business, and Arts (TBA) - University of Suffolk

**LEVEL 7**

In accordance with the FHEQ, at the end of Level 7 students should have a systematic understanding of knowledge, and a critical awareness of current problems and/or new insights, much of which is at, or informed by, the forefront of their academic discipline, field of study or area of professional practice. They will be able to demonstrate originality in the application of knowledge, together with a practical understanding of how established techniques of research and enquiry are used to create and interpret knowledge in the discipline. They should have a conceptual understanding that enables them to evaluate critically current research and advanced scholarship in the discipline and to evaluate methodologies and develop critiques of them and, where appropriate, to propose new hypotheses. They will also be able to deal with complex issues both systematically and creatively, make sound judgements in the absence of complete data, and communicate their conclusions clearly to specialist and non-specialist audiences. In addition, they will be able to demonstrate self-direction and originality in tackling and solving problems, and act autonomously in planning and implementing tasks at a professional or equivalent level.

| First Name | Last Name | Student Number |
|------------|-----------|----------------|
| Anonymous  | User 5    | S              |

| Module Name            | Module Leader | Semester | Academic Year |
|------------------------|---------------|----------|---------------|
| DevOps – (Component 2) | Dr Dzvapatsva | SEM1     | 2025-2026     |

| General Feedback    | Overall mark (%) |
|---------------------|------------------|
|                     | %                |
| Assessment category |                  |

|                                  |            |  |   |  |   |  |   |   |
|----------------------------------|------------|--|---|--|---|--|---|---|
|                                  |            | The introduction clearly outlines the flaws of current systems and provides a well-documented solution aligned with DevOps principles, covering CI/CD, version control, automation, monitoring, security, and collaboration. (maximum marks = 10%)   | Functionality of the Model-View_Template solution (maximum marks = 20%)   | *Continuous steps in the infinity DevOps lifecycle (plan, code, build, and test. delivery, deploy, communication and monitor). (maximum marks = 20%)   | Balance of benefits and limitations of the created pipeline (maximum marks = 20%)   | Lessons learnt from the project/Challenges/Alternative tools/ Future of DevOps ( maximum marks = 20%)  | Presentations/Structure: Labelling of screenshots or diagrams( maximum marks = 6%)  | Conclusion ( maximum marks = 4%)  |
| Feedback for Assessment Category |            |  |   |  |   |  |   |   |
| First (1:1)                      | 70% – 100% | The introduction clearly outlines the flaws of current systems and provides a well-documented solution aligned with DevOps principles, covering CI/CD, version control, automation, monitoring, security and collaboration. Title of report included | The Model-View-Template solution is fully functional, with all components (Model, View, Template) working correctly together. The application is efficient and handles errors well. | The application is highly integrated into the continuous DevOps lifecycle, with all steps (planning, coding, building, testing, delivery, deployment, communication, and monitoring) well-defined and automated. Tools and practices are consistently applied, | The report demonstrates an exceptional balance in the assessment, presenting a nuanced understanding of the interplay between the benefits and limitations of the created pipeline. | The report demonstrates exceptional experience implemented on the Proof of Concept with solid application of principles and tools. Future directions are clearly articulated with a focus on continuous improvement. | The report demonstrates high-quality presentation, with well-labelled visuals, clear organization, and correct referencing. | The conclusion is very clear with impressive insights on the overall practice and reflections originality, and is strongly supported by the learner experience. |



|                          |           |  |  |   |  |   |  |   |
|--------------------------|-----------|--|--|---|--|---|--|---|
|                          |           |  |  | and the development process is streamlined, efficient and fully aligned with DevOps principles.   |  |   |  |   |
| Upper Second (2:1)       | 60% – 69% | The introduction offers a good solution overview but lacks detail and clarity in some DevOps areas like CI/CD, monitoring and full reason behind the implementation of the current solution in favour of the existing.                 | The MVT solution is mostly functional, with minor issues or inefficiencies in some parts of the system (e.g., slight problems with data flow or user interaction).                             | Most steps in the DevOps lifecycle are addressed, but there may be gaps or areas for improvement in automation or communication. The lifecycle is not fully integrated and some phases (e.g., testing, deployment) may be manual or underdeveloped. | The report provides a good assessment of the benefits and limitations of the created pipeline, demonstrating above average consideration of both aspects but with room for improvements. | The report provided a strong evaluation of lessons learnt in the implementation but could benefit from more depth or specificity in principles/tools.   | The report is well-presented but may have minor areas that need improvement in clarity, organisation, or referencing.          | The conclusion is well formulated, insightful, and supported by the results, but with room for improvement. |
| Lower Second (2:2)       | 50% – 59% | The introduction lacks detail and clarity, failing to connect key DevOps practices like CI/CD, infrastructure as code and monitoring to the Django solution. The mention of existing case study challenge is fair but can be improved. | The solution is partially functional but suffers from several significant issues in data flow, logic, or user interaction. The MVT components may be implemented incorrectly or ineffectively. | The report presented average discussion on DevOps lifecycle and implementation of tools.  | The report provides a fairly balanced assessment of the benefits and limitations of the created pipeline, demonstrating a fair consideration of both aspects.                            | The report includes basic reflections but lacks depth. There is limited discussion of alternative tools, principles or future trends and the lessons learned are general or surface-level. The future of DevOps has not been explored sufficiently. | The report meets the basic requirements but it lacks clarity or structure in some areas and may have citation inconsistencies. | The conclusion is moderately insightful and is only partially supported by the results.                     |
| Third (3 <sup>rd</sup> ) | 40% – 49% | The introduction lacks coherence, fails to justify the system's weaknesses and   | The solution is largely non-functional, with major issues in basic functionality   | The report missed several steps in the DevOps lifecycle, or there is significant manual intervention  | The report is heavily skewed towards either the benefits or limitations,   | The presented reflections are below average and they are not related to the case study and Proof of concept. The report does not detail the future  | Significant improvement is needed in presentation, organisation and referencing.   | The conclusion has a lot of gaps and is misleading especially on  |

|      |          |   |  |   |  |  |   |                             |
|------|----------|---|--|---|--|--|---|-----------------------------|
|      |          | provides little to no explanation of how the solution aligns with DevOps principles, missing key aspects like automation, version control and deployment pipelines. | such as weak data flow, poor model design, or incomplete user interaction. | at various stages. Automation and communication practices are inconsistent. Key areas like monitoring or testing are missing or incomplete from the report.   | lacking a balanced perspective.  | trends of DevOps this could be on tools/practices. |   | future directions of DevOps |
| Fail | 0% - 39% | There is no introduction and title for the report with no outline on why the current solution has been proposed   | There is lack of understanding of the software architecture                | The DevOps lifecycle is poorly implemented or missing significant components, such as testing, deployment automation, or monitoring. The process is not integrated, with major gaps in continuous delivery, feedback, or collaboration. | Not provided or is unacceptably poor for the expected academic standards at level 7. | Not provided or is unacceptably poor               | The presentation lacks clarity and coherence, making it difficult to follow. The organisation of ideas is disjointed. A thorough review and revision of these areas are necessary to meet academic standards for level 7. | No conclusion given.        |

\* If your essay submission exceeds the word count limit (3000-word) by up to 10% then there will be no penalty applied. Submissions that exceed the word count by more than 10% will be applied a fixed penalty of 5 percentage points (i.e., 5 marks). In all cases, the penalised mark will not be reduced below a pass level, assuming the work merits a pass. Tables, diagrams (including associated legends), appendices, reference lists, tables of contents, footnotes, and endnotes are excluded from the word count however should be used appropriately.