



Page left blank intentionally

**INDEX**

[**1 SUMMARY 1**](#_30j0zll)

[**2 INTRODUCTION 1**](#_1fob9te)

[2.1 Background 1](#_3znysh7)

[2.2 Stakeholders 2](#_2et92p0)

[2.3 Objectives 3](#_tyjcwt)

[**3 METHODOLOGY 4**](#_3dy6vkm)

[3.1 Considerations & Assumption 4](#_1t3h5sf)

[3.2 Approach 5](#_4d34og8)

[3.3 Activities 5](#_2s8eyo1)

[**4 TARGETED V/S ACHIEVED OUTPUT 6**](#_17dp8vu)

[**5 CONCLUSION 7**](#_3rdcrjn)

[**6 APPENDICES 9**](#_26in1rg)

[6.1 Appendix A – CI/CD Pipeline Overview for Web Application Deployment 9](#_lnxbz9)

**PROJECT DETAILS**

| **Project Name** | Deploy Fusion | | |
| --- | --- | --- | --- |
| **Project Sponsor** | Mohammed Umar (Intern at Cloud Counselage) | | |
| **Project Manager** | Mohammed Umar (Intern) | | |
| **Start Date** | 12/09/2024 | **Completion Date** | 26/09/2024 |

# **SUMMARY**

The purpose of the Deploy Fusion project was to ease the process of deploying a web application by creating a fully automated Continuous Integration/Continuous Deployment (CI/CD) pipeline using GitHub Actions and Vercel for hosting. The undertaking was significant in solving the problem of the current system of Web Application deployment processes that was strictly manual, prone to delays, errors and discrepancies in updates delivery.

The project focused at automating some of the processes within the software development life cycle namely: integration and testing of code and the deployment of the software. Thus with the purpose of implementing that CI/CD pipeline, the project was looking to increase the rate of releases and reduce the rate of failures in those releases making the development process more efficient.

Some potential positive impacts of Deploy Fusion that can be seen within the organization after a long period of time include increased inter team working, a decrease in the errors made during deployments, invention of new features in a faster manner. The continuous automations encourages such organizations to be always on the move as there is constant change in business trends. In the end, this project will be beneficial because it enables the company to grasp new opportunities, fulfill customer expectations and should remain competitive in the line of business.

# **INTRODUCTION**

## Background

The need for Deploy Fusion was due to the inefficacy and the unreliability of the existing deployment process. The traditional ways of deploying web applications manually had their advantages, however, they were considerably slow and high risk hence frequent deployment failures and anomalies. This stifled the development team’s efforts of timely and efficiently pushing updates hence the anger from the engineers and the management.

Some of the critical points in the discussions that were raised in the requirement elicitation meetings include the following:

Inadequacies in the Current Process: Team members underscored the burdensome manual deployments where there are multiple steps that might not be followed. This always incurred additional delays and had to use other resources to solve them.

Automation is Needed: There was uniform agreement on the need to introduce automation to facilitate the integration and deployment processes. The integration of a CI/CD pipeline was to ensure code changes are tested and deployed with no need for human beings to touch them.

Collaboration is Improved: The participants recommended that focusing on developers in this manner must change. There should be a single CI/CD pipeline that allows for such transitions to happen without the teams being out of sync.

Scalability Requirements: Towards the end of the workshop, the organizational expansion plans made it obvious that the existing deployment techniques must be capable of scaling greatly. The system being installed has to support a higher rate of code and change requests without degradation in service or speed of delivery.

Feedback Loop: A faster feedback loop was also considered necessary in the course of the meetings. The team intended to get instantaneous availability of test results by putting testing in the CI/CD process.

## Stakeholders

The project Deploy Fusion was solely undertaken by Mohammed Umar, myself being the stake holder, project manager and developer at the same time. Control over the entire project from inception through to planning, execution and deployment was in my hands.

Nevertheless, even if no external stakeholders were directly involved, a few indirect stakeholders managed to surface during the requirement elicitation meetings:

Development Team: Even though their role did not include active participation in the project, the wider development team would be a beneficiary of the enhancement in the deployment process. Their requirements and challenges were addressed to ensure that the solution does not disappoint the team.

Management: In order to meet the bounds of the project, the management at Cloud Counselage was interested in this endeavor. Indeed, an efficient CI/CD pipeline would save time and increase the firm's ability to respond to the market demands.

Clients/Users: It is also worthy to note that the users of the web application under development are also stakeholders. The way they feel and their level of contentment would be higher with more stable and quicker releases of new updates that will translate to the applications effectiveness.

## Objectives

The objectives identified in the Project Charter of Deploy Fusion project comprises of the following:

Establish a CI/CD Pipeline: The Process of developing a working integrated system will be made easier by building a working Develop Continuous Integration Continuous Development (CI/CD) using GitHub Actions and Vercel to house and publish the application.

Enhance Deployment Efficiency: The time taken to deploy an application will be reduced, and also the process in which the application goes through to be deployed will be improved in a way that little human efforts are required.

Improve Code Quality: There will be a shift towards testing at every stage of development rather than waiting till the end. This will enhance the quality of code and in most cases eliminate bugs at the later stage of development of this system.

Facilitate Rapid Iteration: Facilitate quicker user feedback mechanisms that can enable accommodating new changes and features requested by users in a very short period of time.

**Achievements:**

CI/CD Pipeline Established: A Need Fully functional CI/CD was developed through automation using GitHub Actions and hosting with Vercel. This pipeline integrates code and deploys it more easily.

Deployment Efficiency Improved: By the new system this process has automated the manual intervention and interaction to the extent that it has taken a few minutes for the process instead of hours.

Code Quality Enhanced: In the due course of the project, testing was automated and incorporated into the terminology simple enough for all team members. They created a code freeze for each milestone and allowed

only Nondestructive Changes for Integration to the Milestone.

Rapid Iteration Facilitated: Through enabling automated feedback loops, there has been improvement on how fast the user feedback is gathered and acted upon allowing the team to deliver updates and new functionalities faster.

**Changes to Objectives:**

Since the Project Charter was approved there were no alterations made to the project objectives. Additionally, all objectives remained constant during the project and the project’s outcomes achieved were in close alignment with each of the project objectives.

# **METHODOLOGY**

## Considerations & Assumption

**Constraints:**

Time Limitations: The project had a fixed constraint of two weeks, thus a laser sharp approach to the core functionalities was necessary without going on ramp.

Solo Contributor: I had to handle all development, testing and documentation among other activities on my own within specified timelines since I was the only developer.

Technical Familiarity: The entire project was dependent on my knowledge of GitHub Actions and Vercel. Therefore, it was very important to fix things very fast.

Platform Reliability: This project highly relied on both GitHub and Vercel platforms and any downtime on them would have set back the progress greatly.

**Challenges:**

Debugging of the Automated Workflows: The process of setting up GitHub Actions and integrating it into the deployment process was intricate because one misconfiguration could break the entire system.

Identifying an Appropriate Testing Framework: There were problems incurred in finding the right testing frameworks and integrating them as thorough studies were needed to ascertain the tests would not break the system.

Work-Life Balance: It was also difficult to manage development, testing and documentation tasks simultaneously as it called for proper prioritizing.

**Assumptions:**

Fixed Code: There was a belief that the current code base would not undergo any changes and thus would remain stable and therefore limit the integration risk.

Changes Will Be Encouraged by the Users: It was also believed that such changes would be welcome by the users because it would make deployment faster and more dependable.

Assurance of the Platforms: There was a consideration that GitHub and Vercel would suffice in terms of their services and tools for the duration of the project.

## Approach

In order to address the challenges of inefficient and unmanageable deployments, I incorporated an Agile methodology, which enabled iterative development and variation. It was this particular methodology that allowed feed-sourcing in the long run so that the most critical aspect of building the CI/CD pipeline was done in stages and could be improved depending on results out of testing and the end user’s feedback. Given the concentration on automation and its annealing with integration, simplifying the deployment process was the end goal so that efficiency would be increased as well as human errors minimized.

## Activities

Requirement Gathering: I performed a self- evaluation of the situation from in and identified issues with the current deployment processes in the easy way I knew.

Planning: Presented a project plan by giving the project objectives, the timelines and the needed gadgets while giving much attention to the core features of the CI/CD pipeline.

Setup and Configuration: GitHub Actions workflows were created, and Vercel was set up for hosting purposes, including all the integrations.

Automated Testing Integration: Automated testing frameworks have been chosen and implemented into the CI/CD pipeline in order to maintain quality of code and its functioning.

Testing and Debugging: The pipeline has been thoroughly refined and tested, fixing any bugs that occurred in the course of the implementation to achieve stable performance.

Deployment: The deployment of the web application by the CI/CD pipeline has been performed, as well as the supervision of its operation with a set of tolerable deviations.

Documentation: Archived the whole process with all configurations and insights for the sake of reference for subsequent works.

Review and Retrospective: The results of the project were evaluated against the set goals and suggestions for the improvement of future projects were considered.

# **TARGETED V/S ACHIEVED OUTPUT**

**CI/CD Pipeline Established:**

* **Target:** Implement a fully functional Continuous Integration/Continuous Deployment pipeline using GitHub Actions and Vercel to automate the deployment process of the web application.
* **Achieved:** Successfully established a CI/CD pipeline that integrates automated testing, builds, and deployments, resulting in a reliable deployment process.

**Deployment Efficiency Improved:**

* **Target:** Reduce deployment time to under 10 minutes and minimize manual interventions.
* **Achieved:** The deployment time was reduced to approximately 5 minutes, with significant reduction in manual steps required for deployment.

**Code Quality Assurance:**

* **Target:** Integrate a testing framework that ensures at least 90% code coverage.
* **Achieved:** Integrated automated testing with about 85% code coverage. While close to the target, further refinements are needed to achieve full coverage.

**Facilitate Rapid Iteration:**

* **Target:** Implement feedback loops to allow for rapid updates based on user feedback.
* **Achieved:** Established automated feedback mechanisms, enabling faster iterations and updates, though some feedback cycles were initially slower than anticipated.

**Reasons for Deviation:**

Limits of the Testing Framework:

The slight difference in the code coverages (85% against targeted 90%) could be attributed to some legacy having no full coverage by the testing framework. This was ignored in the first estimation and will help in the next projects to make a detailed evaluation of the legacy code.

Scheduling Problems:

Though the efficiency of deployment in the whole process was beyond the expectations, certain tasks were delayed more than anticipated because of debugging problems caused by GitHub Actions. This showed that more strategies to begin troubleshooting would be ideal and also careful planning and time allocation on testing and debugging would be necessary for the next projects.

Changes in the Feedback Loop:

The first level of feedback mechanism was fine tuned too much in order to cut down on response times. This change took more time than anticipated; however, it ended up contributing greatly to the system that will be used for the next version.

**Lessons Learned:**

All-encompassing Preliminary Evaluations: Make sure that the entire codebase is examined including the old ones, in order to detect gaps in the coverage as early as possible.

Debugging Strategies Adopted In Advance: Debugging and troubleshooting should be given ample time, as the automated workflow may not always be a walk in the park.

Optimize Feedback Mechanisms Early in Project Lifecycle: Introduce feedback mechanisms into the project lifecycle at the early stages for the purposes of enhancement and streamlining of the process to make sure they work and are fit for the purpose from the very beginning.

# **CONCLUSION**

The Deploy Fusion project was able to set a working Continuous Integration / Continuous Deployment (CI/CD) pipeline for the deployment of the web application, greatly improving the efficiency of operations. This project is useful especially for providers such as the development team and management at Cloud Counselage, as it facilitates easing the processes, Singapore shortening the duration of deployment, and providing a lesser scope of errors which can result from doing it manually.

With the new CI/CD pipeline in place, development teams can work in a much better collaborative environment since they can iterate faster and hand off work to each other with less friction. The inclusion of automated testing ensures that the quality of the code delivered is maintained, which in turn provides a better application for the users. Overall, management is also able to gain a degree of operational effectiveness that allows for quick adaptation to market changes and better use of resources.

**Future Scope:**

In the future, there are multiple suggestions for improvement and advancement.

Increasing Code Coverage: Future endeavors can aim to increase the code coverage to a scale of 100%, incorporating supplementary testing frameworks and strategies to guarantee that every aspect is well covered with tests.

Enhanced Surveillance Mechanisms: The introduction of superior observability and analytics capabilities in the CI/CD process will assist in monitoring the performance and deployment metrics of the application in real time for addressing issues beforehand.

Extensibility Improvements: In the event that the application expands, the pipeline can be extended to accommodate many services and microservices, thereby providing assurance that the system will meet future requirements.

Interactions with Other Applications: Additional development tools and project management applications can be investigated with a view to enhancing the interactivity of the system and its operations making the process more effective.

Training and Documentation: It will be important to prepare detailed training materials and instruction manuals on the new CI/CD process to ensure the development team will be able to make full use of the pipeline and train new team members effectively.

# **APPENDICES**

## Appendix A – CI/CD Pipeline Overview for Web Application Deployment

| **Component Name** | **Description** | **Version** | **Responsible Team/Member** |
| --- | --- | --- | --- |
| CI/CD Pipeline | GitHub Actions workflow for automated builds and deployments. | 2.1.0 | Mohammed Umar |
| CI Configuration | Configuration files for GitHub Actions (e.g., .github/workflows) | 1.0.0 | Mohammed Umar |
| Hosting Platform | Vercel configuration for deploying the web application. | 3.0.0 | Mohammed Umar |
| Deployment Script | Custom scripts for deploying to Vercel via CLI. | 1.0.0 | Mohammed Umar |
| Environment Variables | Configuration for environment variables in Vercel. | 1.0.0 | Mohammed Umar |