TRUNG NGUYEN
THE No.1 COFFEE!

PROJECT BRIEF REPORT

Prepared for :

TRUNG NGUYEN COFFEE CORP.

Prepared by : Group 1

Dang Vi Luan - 103802759
Nguyen Linh Dan – 103488557
Tran Bao Huy – 103505799
Vo Thanh Tam – 103487596

TABLE OF CONTENTS

1. PROJECT BACKGROUND

Trung Nguyen Coffee Corporation is a prominent Vietnamese distributor of coffee-related products that is seeking to expand into international markets. Trung Nguyen Coffee is planning to develop an e-commerce platform to grow its sales and purchasing capabilities across digital channels, rather than relying solely on traditional sales methods.

Implementing an e-commerce platform can provide Trung Nguyen Coffee Corporation with several advantages in business operations and customer reach.

The online platform enables direct and personalized customer engagement through features such as account registration, interactions with products (viewing, purchasing, reviewing, etc.), and targeted marketing. These features allow Trung Nguyen Coffee to better understand their customers and adapt the user experience to meet their evolving demands.

From the sales process perspective, an e-commerce platform is a solution to modernize and simplify administrative tasks such as inventory tracking, order management, and secure payment processing. This automation not only reduces unwanted errors but also provides customers with a 24/7 shopping experience.

The biggest difference between e-commerce and traditional platforms is the ability to expand Trung Nguyen's market reach far beyond its physical store locations. By selling products directly to national and international customers, the corporation can access new customer segments that may not have been accessible through traditional retail channels.

The purpose of this project is to develop and implement an e-commerce platform to optimize current business operations and unlock new potential growth opportunities in both domestic and international markets. However, an e-commerce platform that simply meets the functionalities specified in the Request for Proposal (RFP) is not enough on its own to ensure smooth automation of the backend processes. Therefore, our project team has suggested Trung Nguyen Coffee develop an additional part for their e-commerce platform: a CI/CD pipeline to manage the development and deployment of any changes to the platform.

2. PROBLEM STATEMENT

Trung Nguyen Coffee Corporation, a renowned Vietnamese coffee brand, is planning to develop an e-commerce platform to reach more potential customers and increase revenue. They are facing the challenge of inefficient code development, testing, and deployment processes for their platform. Trung Nguyen's current software development lifecycle (SDLC) is manual, time-consuming, prone to errors, and difficult to adapt to the rapidly evolving e-commerce landscape.

The key problems Trung Nguyen Coffee is facing can be summarized as follows:

- **Problem 1: Inconsistent and insecure deployment**

Trung Nguyen's existing deployment process is heavily manual, which leads to delays in updating the platform with new features and bug fixes. This slow update process can make some customers

confused when they accidentally encounter errors on the platform before they can be resolved. The manual and insecure deployment slows down Trung Nguyen's ability to provide a smooth, error-free experience for their customers.

- **Problem 2: No automated testing**

Currently, Trung Nguyen is performing all steps in the testing process manually, from writing the test cases to gathering code from several sources and running the test cases. This leads to a lengthy and inefficient quality assurance process. Moreover, manual actions also increase the risk of regressions and bugs in the production environment, which can negatively destroy the customer experience.

- **Problem 3: Limited collaboration and observation**

The lack of a centralized and automated system for managing the SDLC makes it hard for Trung Nguyen's e-commerce development team to collaborate effectively when updating new versions or making any changes to the platform. Without a CI/CD pipeline, the team cannot see the deployment output quickly, which limits their ability to identify and address issues timely.

- **Problem 4: Inflexible scalability**

Trung Nguyen Coffee's e-commerce platform needs to handle thousands of customer visits per day. However, the current manual processes and lack of automated infrastructure provisioning make the platform unsustainable. This lack of robust automation can cause the platform to become unstable and potentially corrupted when workloads increase, as the manual processes are unable to scale effectively to meet the demand.

To address the above problems and ensure the development and management of a successful e-commerce platform, Trung Nguyen Coffee needs to implement a robust CI/CD pipeline. The pipeline will enable the company to automate the software development lifecycle, especially the testing phase, to provide consumers and business owners with a reliable e-commerce solution.

3. PROJECT GOALS AND OBJECTIVES

After discussing with Trung Nguyen Coffee, the team proposed the following objectives that need to be satisfied:

- **Prepare the cloud infrastructure**

Prepare the cloud infrastructure for deployment to deliver high availability, reliability, and scalability for the e-commerce platform of Trung Nguyen Coffee Corporation.

- **Integrate the CI/CD Pipeline**

Working with the Agile Scrum methodology requires continued delivery. Therefore, building the CI/CD pipeline is necessary to ensure smooth operation between the development and deployment stages. Additionally, the automation of the deployment process will reduce IT spending costs.

- **DevSecOps Integrated**

During the CI/CD process, there will be some security risks that we do not expect. By integrating DevSecOps techniques, the team will deliver a secured CI/CD pipeline for best practices.

- **Performance testing**

Perform the testing on the platform (with manual and automation methods) to guarantee that the CI/CD pipeline works as expected.

- **Provide training documentation**

Preparing necessary documents for the Trung Nguyen Coffee Corporation employees to maintain the e-commerce platform, CI/CD pipeline, and cloud infrastructure in the handover minutes.

4. SELECTED TOOLS AND JUSTIFICATION

Terraform for managing cloud infrastructure AWS

Instead of manually configuring the cloud platform on the management console, the team decided to use Terraform to manipulate the cloud infrastructure. Infrastructure as Code is well implemented in this project for easy maintenance, provisioning of cloud resources, and troubleshooting. In the handover minute, the Trung Nguyen Coffee Corporation employees can check the configuration of Terraform files to understand the cloud infrastructure.

Git for Version-Source Control

It requires the use of version-control software in any software development project. The developer and operation team must be familiar with Git to keep track of the code version in SDLC to figure out the bug, troubleshoot, and even revert to the code version we want.

CI/CD Pipeline

Continuous Integration Tools

- **Github Action:** The team will use GitHub as the platform to store the remote source code. Any changes in the source code, GItHub Action will trigger the CI pipeline to begin the CI/CD process.
- **Docker:** When the CI pipeline is triggered, a new Docker image will be built and pushed to the Docker Hub. Using container technology such as Docker will avoid any conflict between development and deployment environments.

Continuous Delivery Tools

- **Kubernetes:** All Docker containers will run on Kubernetes (K8S). Using K8S allows us to increase the usage of containers in order to achieve high availability. Assuming that the system handles a million users, K8S will distribute the user traffic to different pods, which is an abstract concept referring to a group of containers, to increase the performance and availability of the system.
- **ArgoCD:** The ArgoCD serves as the controller for our K8S cluster. It will synchronize with the GitHub repository to detect any changes in the source code before deploying the

application to the K8s cluster. Overall, using ArgoCD is a way to manifest our K8S environment.

DevSecOp Tools

- **AWS Secret Manager:** Protect any credentials related to the cloud platform (accessed key, etc.).
- **Git Secret:** Protect any credential related to the application (database password, database endpoint, etc.).
- **Sonarqube:** Automatically detect security risks in the application source code
- **Clair:** Securing the Infrastructure as Code (Terraform)
- **Grafana and Prometheus:** Provide visualization and alerts during the CI/CD process.

5. SUCCESS METRICS

To measure the performance and success of the CI/CD pipeline for Trung Nguyen Coffee, it is essential to track and analyze several key metrics that provide insights into the efficiency and reliability of the development process.

- **Cycle Time:** This metric measures the time taken for a feature or bug fix to move from development to production. Optimizing this time ensures rapid delivery of value to customers, ideally in 30 minutes
- **Lead Time:** This encompasses the entire process from feature identification to deployment. Minimizing lead time, with a target of 2 hours, helps streamline the development process.
- **Deployment Frequency:** High deployment frequency, aiming for around 10 deployments per week, indicates continuous delivery of improvements and new features.
- **Mean Time to Recover (MTTR):** Keeping the MTTR low, preferably within 1 hour, is crucial for maintaining service reliability and customer satisfaction after production issues.
- **Change Failure Rate (CFR):** Monitoring the CFR, with a goal of keeping it below 5%, helps identify and mitigate risks associated with new deployments.
- **Code Coverage:** Ensuring comprehensive automated testing with a robust code coverage target of at least 80% reduces the likelihood of defects in production.
- **Test Execution Time:** Optimizing test execution time to around 15 minutes provides quick feedback to developers, facilitating a fast and efficient development cycle.

By diligently tracking these metrics, Trung Nguyen Coffee can enhance its CI/CD pipeline, ensuring it is robust, efficient, and aligned with business goals. This approach ultimately drives better service delivery and customer satisfaction.

6. PROJECT SCOPE

6.1. Functional requirements

A. DevOps Pipeline

A1. Version Control

- Implement Git to manage source code changes to the code base, enabling collaboration and code management.
- Prepare the repositories and any needed infrastructure code using GitHub, DockerHub, Terraform Registry.

A2. Continuous Integration

- Implement automatic build and test processes to ensure the continuous development flow by combining the selected tools (GitHub and Docker)
- Design unit tests, and any necessary code validation to ensure the application's functionality.
- Use ArgoCD to automatically trigger build upon code commits.
- Inform developers of the status of build/test/deploy processes (successful/failed)

A3. Deployment

- Use Docker to generate and push container images to a registry, use versioning schemes to maintain traceability
- Implement Dockerfiles for services and components and use CI processes to automatically build and publish Docker images.

A4. Operation

- Implement Terraform to set up cloud infrastructure, integrating Terraform code into the CI/CD pipeline for consistent provisioning across environments.
- Deploy the container of application to a Kubernetes cluster. Kubernetes will manage the lifecycle of the application containers and monitor the application performance.

B. DevSecOps Integration

B1. Version Control

- Use the Git Secrets tool to set up the pre-commit hooks to protect the source code and prevent unintentional commits of sensitive data, such as credentials or access keys. This will automatically scan files before they are committed and block the commit if any prohibited patterns are detected.
- Use AWS Secret Manager to centralize the management of passwords, API keys, and other sensitive credentials.

B2. Continuous Integration

- Integrate SonarQube, a SAST tool, to scan the source code for security vulnerabilities. It helps minimize false positives and improve the efficiency of the security risk analysis.
- Combine Source Composition Analysis (SCA) with GitHub to scan for vulnerable dependencies in the code. This tool helps identify and address potential security risks hidden in third-party services.
- Use OWASP ZAP for Dynamic Application Security Testing (DAST) to identify risks in the running applications. This dynamic testing approach complements the SAST performed by SonarQube to boost the effectiveness of security assessment.

B3. Deployment

- Integrate Clair, an open-source vulnerability static analysis tool for containers, with the CI/CD pipeline to scan the Docker images.

B4. Integration

- Configure Prometheus to centralize the metrics collections from different sources, like application containers or infrastructure resources as the time-series data to analyze historical data patterns.
- Connect Grafana to your Prometheus data source to visualize the collected metrics on a customizable dashboard to have a broader vision of the system's performance.
- Configure Prometheus to trigger alerts when predefined thresholds are exceeded, or strange patterns are detected.
- Define alerting rules in Prometheus to monitor errors in pipeline operations.
- Link the alerting system to Telegram to send notifications to the relevant team members or channels when alerts are triggered to ensure a timely response to the incidents.

6.2. Non-functional requirements

Performance

- Scalability: The pipeline must be able to handle and facilitate a large growing number of updates and functionalities to the website. The pipeline must support horizontal scaling if needed.
- Efficiency: The pipeline must be able to perform a significant number of builds, tests, deployments simultaneously without any bottlenecks.
- Latency: The pipeline must ensure minimal latency in build, test, deploy stage to reduce the overall development duration.

Reliability

- Availability: The pipeline must be highly available with continuous operation and minimal downtime.
- Fault Tolerance: The pipeline must be able to recover after failures without data loss or causing interruption to the service.

Integrability

- The pipeline must ensure compatibility with tools such as: Github for version control, ArgoCD for CI tool, Dockerhub for container registries.

Performance monitoring

- The pipeline uses Prometheus to collect performance and health metrics of the system.
- The pipeline uses Grafana to visualize real-time metrics collected from the running application on the dashboards.
- Prometheus is configured with Telegram to create an alert system, which sends real-time alerts, notifications of metrics crossing defined thresholds.

6.3. Out of scope

Training for Cloud Services and relevant applied tools: The project's scope only includes the user manual, training documentation, and training sessions for the operation of the project outcome—the e-commerce platform and the CI/CD pipeline built to support the platform. The training and documentation for each individual tool used in the project, such as AWS Services, Docker, or detailed Terraform workflows, are not within the scope of this project.

Additional features or functionality: This project only delivers core functionality and features required by the customers. It does not cover any additional features, integrations, customizations, or ongoing maintenance beyond the initial project requirements. Any requests for extra functionality or enhancements will be evaluated and require a new contract.

Third-party tools and services management: All ongoing maintenance, management, updates, or troubleshooting support related to the system after the delivery to the customer are out of scope. Any requests or issues involving third-party tools or services are also not supported.

Marketing Plan and User Adoption: The project team will not be responsible for any marketing, promotion, or user adoption efforts related to the built e-commerce platform and CI/CD pipeline. Ensuring that the customer's business developers and end-users successfully adapt to and operate the new system is out of scope.

7. PROJECT DELIVERABLES

By the end of the project, the following deliverables will be delivered to Trung Nguyen Coffee Corporation:

- **A secured CI/CD pipeline design and development**

This deliverable outlines the design of a comprehensive CI/CD pipeline to support the function development and deployment of the e-commerce platform. The pipeline integrates basic functionalities such as version control, continuous integration (CI), continuous deployment (CD), and automation in the SDLC. DevSecOps methodologies are also integrated to establish a secure and streamlined development and deployment process.

Below is a summary of the key components and integrated tools in the pipeline:

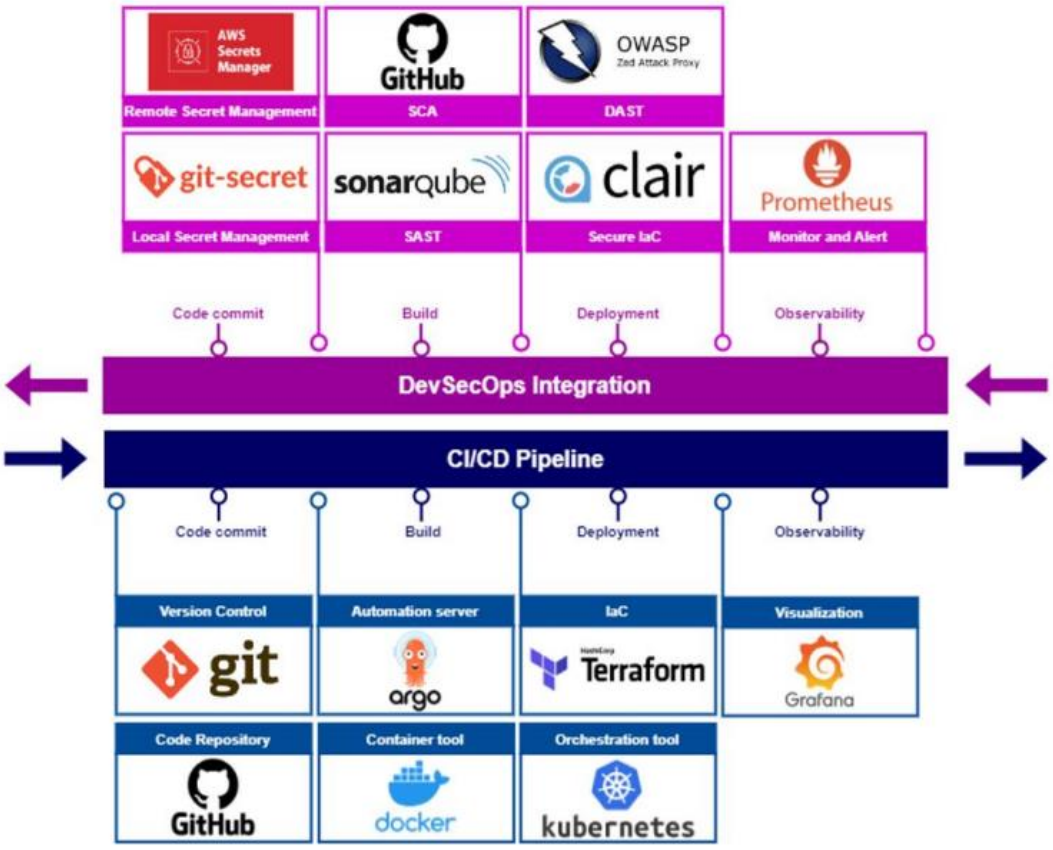

Figure 1: Summary of the CI/CD pipeline components

- **Project Final Report**

This report will summarize and give more insights into this CI/CD pipeline development to support the e-commerce project. It will outline the project requirements, objectives, and major activities carried out during the project, such as making project strategies, selecting tools, designing, and configuring the pipeline. The report will also include the project outcomes and project closure activities. The report also discusses the challenges faced, lessons learned, and recommendations for future improvements.

- **Relevant training materials and documentation of the developed CI/CD pipeline**

Relevant training materials and documentation of the developed CI/CD pipeline

These materials were developed and provided to ensure a thorough understanding of the pipeline's architecture, processes, and functionality. The key documentation artifacts include:

- CI/CD Pipeline Architecture Design (with diagrams and explanation)
- CI/CD Pipeline Training Book for Administrators and Developers
- CI/CD Pipeline Troubleshooting Guide to Resolve Common Issues

- **Setup documentation and User Knowledge Tranfer**

Comprehensive setup documentation and user knowledge transfer activities were included to ensure the successful onboarding of the CI/CD pipeline.

The setup documentation includes:

- Infrastructure as Code Templates
- CI/CD Tool Setup Guides
- Configuration files for a specific environment

The User Knowledge Transfer includes

- CI/CD Pipeline Architecture Design (with diagrams and explanation)
- Introductory sessions to familiarize the development and operations teams with the CI/CD pipeline
- Tutorial videos for common pipeline-related tasks