**Technical Report: Designing a Dynamic CI/CD Pipeline for Pet Adoption Web Application Using AWS CodePipeline**

**1. Introduction**

**1.1 Title**

Designing a dynamic CI/CD Pipeline using AWS Code Pipeline for Pet Adoption Web Application

**1.2 Team Members**

* Reshmi Harikumar Lathakumari
* Shrey Bhoir
* Vimmi Cherusserykkaran

**1.3 Goal**

The goal of this project is to establish a robust Continuous Integration and Continuous Deployment (CI/CD) pipeline for the "Pet Adoption" online application using AWS Code Pipeline. The aim is to streamline and improve the software development and deployment processes, ensuring efficient and reliable delivery of new features and updates to the web application. This project intends to accelerate updates, repairs, and deployments while ensuring thorough testing of each code change.

**1.4 Project Beneficiaries**

This project benefits various stakeholders, including Developers, Testers, Deployers, Business Owners, and Users.

**2. Project Scope**

The project's scope includes the design and implementation of a comprehensive CI/CD pipeline for the Pet Adoption web application. It encompasses CI/CD pipeline configuration, source code management, deployment automation, automated builds, and tests, testing and quality assurance, scalability, and flexibility. Technologies and services utilized include GitHub, AWS, MySQL, Python, and Docker.

**3. Project Approach**

The project starts by understanding the requirements of the Pet Adoption application. The approach involves customizing the CI/CD pipeline, integrating AWS Code Pipeline with AWS Code Build, and utilizing GitHub for version control. Docker ensures consistent deployments, Python for backend development, and MySQL for CRUD activities. Separate build and production environments are provided for rigorous testing, leading to a smooth transition to live deployment. Collaboration, customization, security, automation, and continuous improvement are emphasized for a successful CI/CD pipeline installation.

**4. Methodology**

**4.1 Steps**

The methodology involves building an effective CI/CD pipeline covering front-end, back-end, database, deployment, and Docker.

Key steps include:

* Needs analysis with Docker integration.
* Backend development using Python and MySQL for the database.
* Selection of AWS as a hosting provider
* Creation of an all-inclusive CI/CD pipeline
* GitHub for code management and AWS Code Pipeline for CI/CD

**5. Resources**

**5.1 Technologies and Tools**

* Cloud Platform: AWS
* CI/CD Tools: AWS Code Pipeline
* Version Control System: GitHub
* Docker: Docker for containerization
* Front-end: HTML, CSS, Js
* Back-end: Python
* Database System: MySQL
* Development Tools: PyCharm

**6. Tasks**

**6.1 Detailed Tasks**

* Docker Containerization:

Create Docker containers with configured volumes and networks.

* CI/CD Pipeline Setup:

Establish a comprehensive CI/CD pipeline for front-end, back-end, and database components, including Docker containers.

* Environment Setup:

Set up staging and production environments for testing and live deployment, incorporating Docker containers.

* Front and Back-end Development:

Develop front-end and back-end components based on specifications.

* Database Design and Implementation:

Create the database structure and perform CRUD operations.

* Documentation:

Create detailed documentation for code, database, Docker containers, and pipeline configuration.

**7. Summary**

The project aims to gain a deep understanding of CI/CD and its best practices, ensuring efficient and reliable software delivery. Expertise in deploying various components, such as front-end, back-end, and databases, will be automated through Docker and AWS. The CI/CD pipeline will be configured with stage settings covering the full application stack. The team will leverage GitHub for code management and implement automated testing, including unit tests, integration tests, and security testing, to ensure high code quality. The project will equip the team with valuable skills in CI/CD practices and tools.