**Business Requirements Document (BRD)**

**Project Title:** Flask-based ML Model API Deployment with Docker and Kubernetes, and CICD Pipeline Integration with Jenkins

**1. Introduction** This document outlines the business requirements for deploying a Flask-based ML Model API using Docker and Kubernetes and integrating a CI/CD pipeline with Jenkins to automate the API building process triggered by new code changes in the Bitbucket repository.

**2. Objective** The objective of this project is to deploy the Flask-based ML Model API in a scalable and containerized environment using Docker and Kubernetes. Additionally, we aim to establish a CI/CD pipeline with Jenkins to automate the API building process and ensure efficient deployment of new code changes from the Bitbucket repository.

**3. Scope** The scope of this project includes:

* Deployment of Flask-based ML Model API using Docker and Kubernetes.
* Configuration of a scalable and reliable containerized environment.
* Integration of a CI/CD pipeline using Jenkins to automate the API building process.
* Triggering the pipeline on new code changes in the Bitbucket repository.
* Ensuring smooth and efficient deployment of the API.

**4. Requirements**

**4.1. Deployment Requirements**

* Containerization: The Flask-based ML Model API should be deployed using Docker containers to ensure consistency and portability.
* Orchestration: Kubernetes should be used to manage and orchestrate the Docker containers, providing scalability, high availability, and load balancing.
* Scaling: The solution should be capable of scaling the containers to handle increased API traffic efficiently.

**4.2. CI/CD Pipeline Requirements**

* Jenkins Integration: Jenkins should be integrated to establish a CI/CD pipeline for automating the API building process.
* Bitbucket Integration: The pipeline should be triggered on new code changes in the Bitbucket repository, ensuring seamless deployment of updates.
* Build Automation: Jenkins should automatically build the Docker image for the API, ensuring consistent and reliable deployment.
* Testing and Quality Assurance: The pipeline should include appropriate stages for unit testing, integration testing, and quality assurance checks.
* Deployment Automation: Jenkins should automate the deployment process by pushing the built Docker image to the Kubernetes cluster.
* Monitoring and Reporting: The pipeline should provide monitoring and reporting capabilities to track the status and success of each build.

**5. Deliverables**

* Deployed Flask-based ML Model API in a Docker and Kubernetes environment.
* Configured CI/CD pipeline using Jenkins for the API building process.
* Automated deployment of the API triggered by new code changes in the Bitbucket repository.
* Documentation on the setup, configuration, and maintenance of the deployment and CI/CD pipeline.

**6. Timeline and Milestones**

* Milestone 1: Dockerize the Flask-based ML Model API
* Milestone 2: Setup Kubernetes for container orchestration
* Milestone 3: Integrate Jenkins and Bitbucket for CI/CD pipeline
* Milestone 4: Implement automated deployment and testing stages
* Milestone 5: Finalize documentation and project handover

This BRD outlines the business requirements for deploying the Flask-based ML Model API using Docker and Kubernetes and integrating a CI/CD pipeline with Jenkins. The project aims to ensure efficient and automated deployment, scaling, and testing of the API, while enabling seamless integration with the Bitbucket repository.