



DEPARTMENT OF APEX INSTITUTE OF TECHNOLOGY

PROJECT PROPOSAL

1. Project Title: - Create a Complete CI/CD pipeline for a web application using Jenkins, docker etc.

2. Project Scope: -

The purpose of this project is to create a complete CI/CD pipeline for a web application using Jenkins and Docker. The pipeline will automate the process of building, testing, and deploying the web application to production. The pipeline will ensure that the application is tested thoroughly before being deployed to production, reducing the risk of introducing bugs or other issues.

The web application will be developed using a modern web framework, such as React or Angular, and will include a backend server, such as Node.js or Django. The web application will be hosted on a cloud platform, such as Amazon Web Services (AWS) or Microsoft Azure.

The pipeline will start with the developer committing code changes to a source code repository, such as Git. Jenkins will then pull the changes from the repository and run a series of automated tests, including unit tests, integration tests, and end-to-end tests. The tests will be designed to verify that the application is functioning as expected and that new changes do not introduce any new issues.

If the tests pass, Jenkins will then build the application and package it into a Docker container. The container will include the application and its dependencies, making it easy to deploy and manage. The container will be pushed to a Docker registry, such as Docker Hub or Amazon Elastic Container Registry (ECR).

The pipeline will then deploy the container to a staging environment, where it will be further tested and reviewed by the team. If the application is approved, the pipeline will then deploy the container to the production environment. The production environment will be hosted on a cloud platform, such as AWS or Azure, and will be configured to handle traffic from users.

The pipeline will also include monitoring and logging to help identify and troubleshoot any issues that arise. The monitoring will include metrics such as CPU usage, memory usage, and network traffic. The logging will include detailed information about errors and events that occur within the application.

Overall, the goal of the project is to create a complete CI/CD pipeline that can be used to develop and deploy web applications with greater speed and reliability. The pipeline will automate the development, testing, and deployment process, helping to reduce the risk of errors and improve the overall quality of the application.

3. Requirements: -

The following are the key requirements for this project:

- **Continuous Integration:** The pipeline will include continuous integration, which means that code changes will be automatically built and tested every time they are pushed to the source code repository.
- **Automated Testing:** The pipeline will include automated testing, which means that tests will be automatically run on the application to ensure that it is functioning as expected. The testing should include unit tests, integration tests, and end-to-end tests.
- **Containerization:** The application will be containerized using Docker, which means that the application and its dependencies will be packaged in a container that can be run anywhere, making it easier to deploy and manage.
- **Continuous Delivery:** The pipeline will include continuous delivery, which means that changes will be automatically deployed to a staging environment for further testing and approval before being deployed to production.
- **Deployment to Cloud Platform:** The pipeline will include deployment to a cloud platform, such as AWS or Azure. The deployment should be automated and easily repeatable.
- **Monitoring and Logging:** The pipeline will include monitoring and logging, which means that the application will be monitored for errors and issues, and logs will be generated to help troubleshoot any problems that occur.
- **Security:** The pipeline will include security measures to ensure that the application is secure and protected from attacks. This includes setting up appropriate firewall rules, using HTTPS, and implementing other security best practices.

➤ Hardware Requirements

CPU	800 MHz or faster 32-bit or 64-bit processor	1GHz or faster 32-bit or 64-bit processor
Physical Memory (RAM)	4GB	8GB
Hard Disk Drive	20GB with minimum 15GB free disk space	40GB with minimum 15GB free disk space
Graphics Processor	SVGA adapter with DirectX-9 Support and m in of 800 × 600 resolution	Video adapter compatible with DirectX 9, WDDM driver support and minimum 128 MB RAM
Optical Drive	CD- or DVD-ROM Drive	DVD-ROM Drive

➤ Software Requirements

1. JENKINS - Jenkins plays a critical role in this CI/CD pipeline for a web application. As an open-source automation tool, Jenkins provides a flexible and extensible platform that enables the creation of a fully automated pipeline.

2. DOCKER - Docker is another critical component in this CI/CD pipeline for a web application. Docker is a containerization platform that enables the creation of lightweight, portable, and self-contained application packages called containers. Containers are isolated environments that include everything an application needs to run, including code, dependencies, and configuration.

3. DIGITAL OCEAN SERVER - In this CI/CD pipeline for a web application, Digital Ocean can be used as a cloud platform for hosting the web application in a production environment. Digital Ocean is a cloud infrastructure provider that offers virtual private servers (VPS) or "Droplets" that can be used to host applications in the cloud.

4. GIT - Git is an essential tool in this CI/CD pipeline for a web application. Git is a distributed version control system that enables developers to manage changes to the source code of the application.

5. GITHUB - GitHub is a web-based platform that provides hosting for Git repositories and offers several features and tools for managing the software development process. In this CI/CD pipeline for a web application, GitHub can be used as the Git repository hosting service, providing a central location to store the source code of the application.

6. JENKINS EXTENSIONS - Jenkins is a highly extensible tool, with a vast ecosystem of plugins that can be used to add functionality to the platform. Some of the most popular Jenkins extensions or plugins include:

These are Some useful extensions that we are going to use in our project:

Pipeline: Pipeline is a plugin that enables the creation of continuous delivery pipelines using a domain-specific language (DSL). It allows users to define the stages and steps of their pipeline in code, providing a powerful and flexible way to define their CI/CD workflows.

Docker: The Docker plugin enables Jenkins to interact with Docker and Docker Compose, allowing users to build and publish Docker images, and run Docker containers as part of their pipeline.

Git: The Git plugin provides Jenkins with the ability to integrate with Git repositories, allowing users to clone source code, branch, and merge changes as part of their pipeline.

GitHub: The GitHub plugin provides integration between Jenkins and GitHub, allowing users to trigger builds based on GitHub events, such as pull requests, commits, and releases.

- ***STUDENTS DETAILS***

<i>Name</i>	<i>UID</i>	<i>Signature</i>
<i>SHUBHAM</i>	<i>21CDO1062</i>	
<i>ANKIT RAJ</i>	<i>21CDO1058</i>	
<i>NIKHIL SINGH</i>	<i>21CDO1033</i>	
<i>DURGESH PATEL</i>	<i>21CDO1043</i>	

- ***APPROVAL AND AUTHORITY TO PROCEED***

We approve the project as described above, and authorize the team to proceed.

<i>Name</i>	<i>Title</i>	<i>Signature (With Date)</i>