Project Idea: Automated Deployment of a Simple Web Application

Project Overview:

The project involves creating an automated pipeline for deploying a static web application using Jenkins for continuous integration and Docker for containerization. The pipeline will automatically build, test, and deploy the application, ensuring a smooth and consistent deployment process.

Objectives:

Automate the Build Process:

Use Jenkins to automate the build process whenever code changes are pushed to the repository.

Containerize the Application:

Package the web application in a Docker container for consistent deployment.

Automate Deployment:

Deploy the Docker container to a local Docker environment automatically via Jenkins.

Technologies Used:

Jenkins:

For continuous integration and continuous deployment (CI/CD).

Docker:

For containerizing the application.

Git:

For version control.

Docker Compose:

For managing multi-container Docker applications (optional, if needed).

Shell Scripting:

For deployment scripts.

Project Setup and Configuration:

Step 1: Prepare the Application

- Create a simple web application (e.g., an HTML/CSS/JS website).
- Ensure the application is working locally.

Step 2: Set Up Docker

- Create a Dockerfile: Write a Dockerfile to build a Docker image for the web application.

```Dockerfile

```
# Use an official Nginx image as a parent image
FROM nginx:alpine
# Copy the application files to the container
COPY . /usr/share/nginx/html
```

Build Docker Image:

- **Run `docker build -t my-web-app .` to build the Docker image.**

Step 3: Set Up Jenkins

Install Jenkins:

Install Jenkins on your machine or use a hosted version.

Install Plugins:

Install necessary plugins like *Git*, *Docker*, and *Pipeline*.

Create a New Job:

Set up a new Jenkins job (e.g., a freestyle project or a pipeline).

Source Code Management:

Configure the job to pull code from a Git repository.

Build Triggers

Set up triggers (e.g., on code push to Git).

Build Steps:

Add steps to build the Docker image.

```shell
docker build -t my-web-app .
```

Post-Build Actions:

Introduction:

This document outlines the process and setup for automating the deployment of a simple web application using Jenkins and Docker. The goal is to demonstrate an automated CI/CD pipeline.

Setup Instructions:

1. Clone the Repository:

```bash
git clone [repository-url]
```

2. Build Docker Image:

```bash
docker build -t my-web-app .
```

3. Run Docker Container:

```bash
docker run -d -p 80:80 my-web-app
```

4. Configure Jenkins:

- Install necessary plugins.
- Create a Jenkins job to automate the build and deployment process.

Troubleshooting:

- Issue: Docker container fails to start.
- Solution: Check Dockerfile for errors and ensure all files are correctly copied.

Future Improvements:

- Enhance Security: Implement security best practices for Docker and Jenkins.
- Scale Deployment:Explore Kubernetes for scaling the application.

Conclusion:

This project successfully demonstrates the automation of a web application deployment using Jenkins and Docker, showcasing fundamental DevOps skills.