Node.js Static Site with Docker & Jenkins CI/CD Pipeline

Project Overview

This project demonstrates a complete **CI/CD pipeline** for a Node.js static website using:

- **Node.js** (backend serving HTML, CSS, JS)
- **Docker & Docker Compose** (containerization)
- **Jenkins** (automation pipeline)
- **AWS EC2** (deployment server)

The pipeline automates the build, test, and deployment processes, enabling rapid and reliable delivery of the application.

Features

- Static site served by a lightweight Node.js server
- Dockerized application for environment consistency
- Jenkins pipeline automates cloning, building, and deploying
- Hosted on AWS EC2 with proper security and port configurations
- Supports easy updates via GitHub commits triggering Jenkins

---

Prerequisites

- AWS account with EC2 instance running Ubuntu
- Jenkins installed on EC2 with Docker and Docker Compose
- GitHub repository for your project
- Security groups configured to allow ports 22, 3000, and 8080

Project Structure

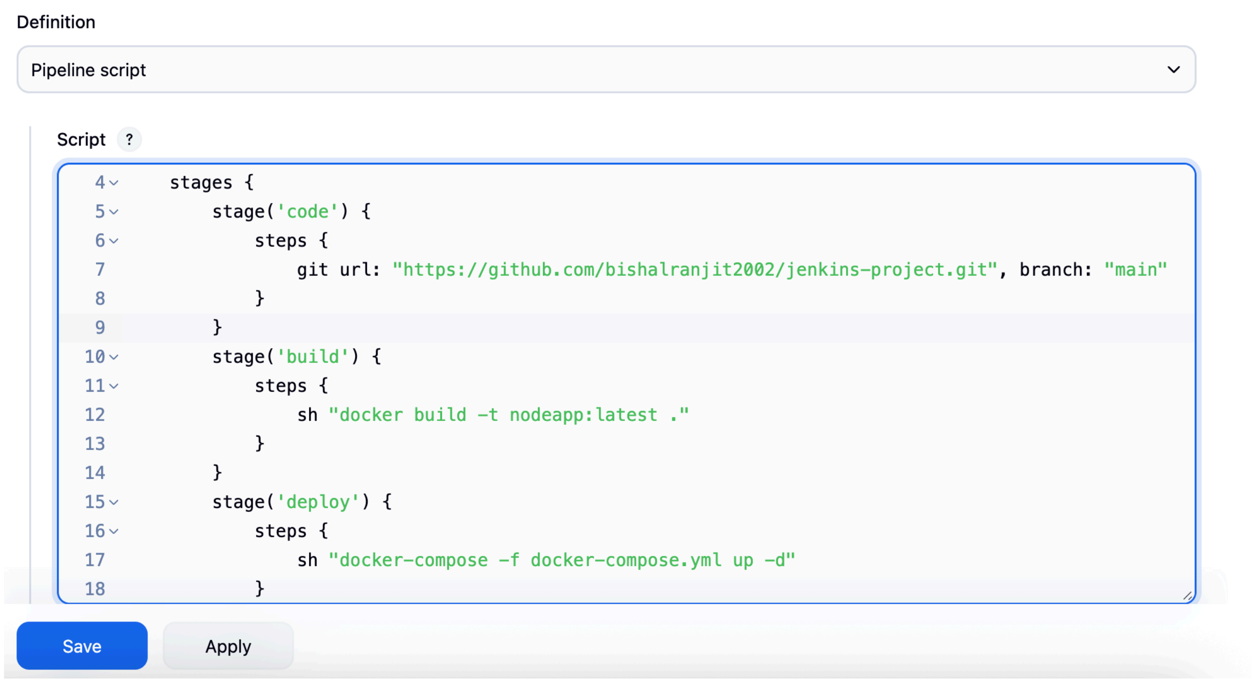
```
node-static-site/
├── public/
│   ├── index.html
│   ├── styles.css
│   └── script.js
├── server.js
├── package.json
├── Dockerfile
├── docker-compose.yml
└── README.md
```

Jenkins Pipeline

The Jenkinsfile automates:

- Cloning the GitHub repo
- Building the Docker image
- Deploying the app with Docker Compose

Pipeline Script:



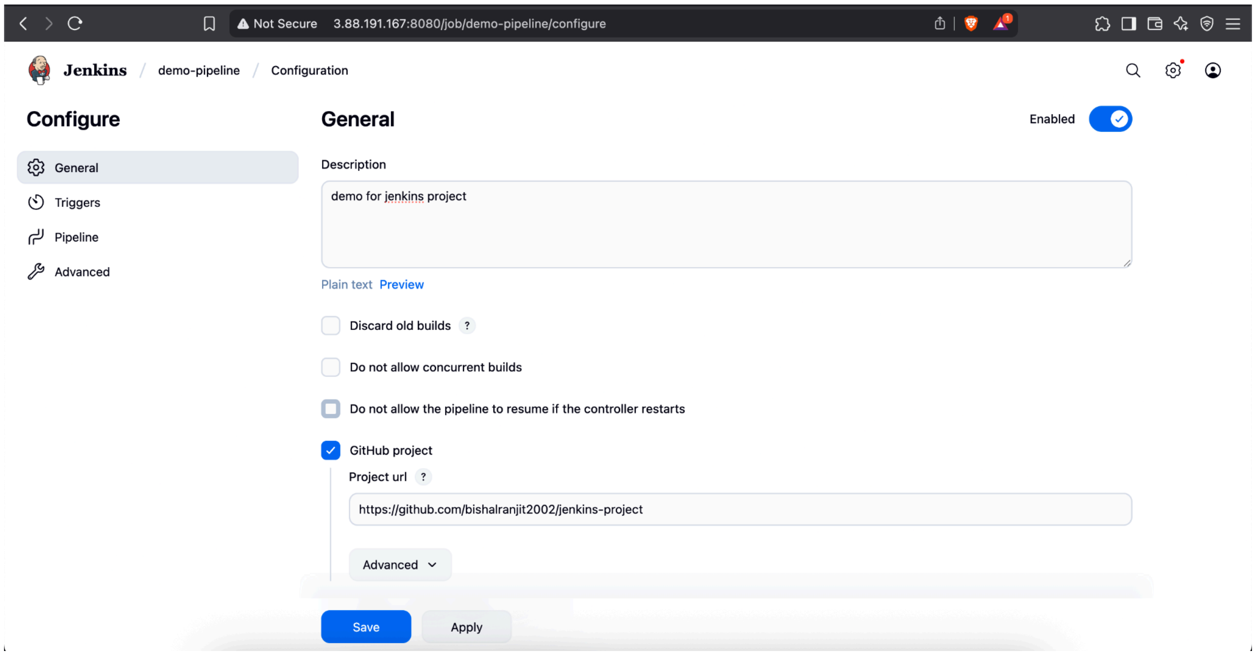
```
    stages {
        stage('code') {
            steps {
                git url: "https://github.com/bishalranjit2002/jenkins-project.git", branch: "main"
            }
        }
        stage('build') {
            steps {
                sh "docker build -t nodeapp:latest ."
            }
        }
        stage('deploy') {
            steps {
                sh "docker-compose -f docker-compose.yml up -d"
            }
```

AWS EC2 Setup

- Ubuntu instance with security group allowing ports 22 (SSH), 3000 (app), and 8080 (Jenkins)
- Installed Jenkins, Docker, Docker Compose, and Java
- Added Jenkins user to Docker group for permission

Accessing the Application

- Jenkins UI: `http://<EC2_PUBLIC_IP>:8080`
- Node.js app: `http://<EC2_PUBLIC_IP>:3000`


Not Secure 3.88.191.167:8080/job/demo-pipeline/configure

Jenkins / demo-pipeline / Configuration

Configure

General

Enabled

Description

demo for jenkins project

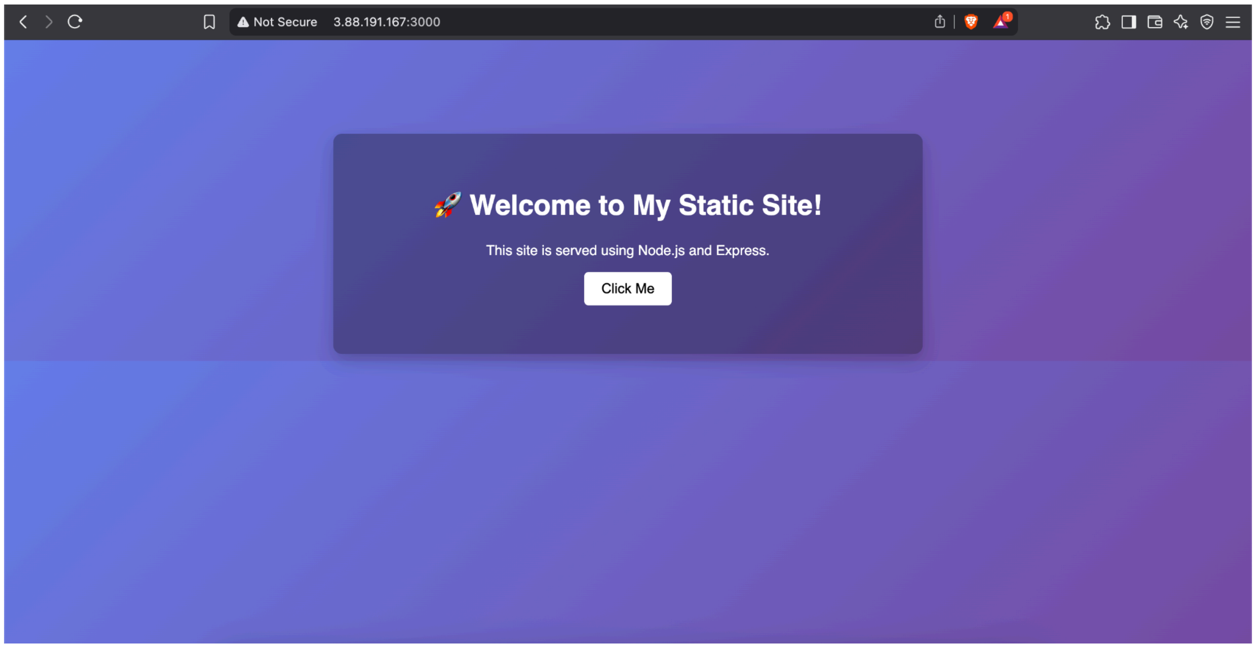
Discard old builds: not checked

Do not allow concurrent builds: not checked

Do not allow the pipeline to resume if the controller restarts: not checked

GitHub project: ✓

Project url: https://github.com/bishalranjit2002/jenkins-project


Not Secure 3.88.191.167:3000

🚀 Welcome to My Static Site!

This site is served using Node.js and Express.

Future Improvements

- Add automated tests and linting in Jenkins pipeline
- Implement multi-stage Dockerfile for optimized builds
- Use Nginx as a reverse proxy for better security and performance
- Configure SSL certificates for HTTPS access