CI/CD Jenkins Pipeline-Based Web Application Deployment

College Name: BMS Institute of Technology and Management

Team Members:
1. Asrithya Vardhan - CAN_35608412

2. Chinmay Pawar - CAN_35699997

3. Ayush Prasad - CAN_35595462

4. Mushahid M - CAN_35595391

Overview:

This project focuses on deploying a web application using CI/CD pipelines implemented via Jenkins. The objective is to gain real-time exposure to modern software development practices involving continuous integration and continuous deployment on cloud infrastructure using EC2, GitHub, and Jenkins. The application is designed to bridge academic concepts with industry expectations by implementing DevOps principles, cloud tools, and AI-powered enhancements.

Functionality of the Application:

- End-to-end web application deployment using Jenkins pipelines.
- Integration with GitHub via Webhooks to trigger automated builds.
- Deployment onto EC2 instances using Amazon Linux.
- Demonstrates full development cycle: code → build → test → deploy.
- Hands-on practice with backend technologies (Java, Spring Boot) and AI tools.

Services to Use:

Service/Tool	Purpose
Jenkins	Automates CI/CD processes for build and deployment
GitHub	Code hosting, version control, and webhook triggers
Amazon EC2	Hosts the deployed application on scalable cloud VMs
Amazon Linux	OS environment for Jenkins master and agents
Java & Spring Boot	Backend development technologies

AI Tools (e.g., Gemini, GitHub Copilot)	Assist in coding and AI-powered software development

Tools:

Tool/Library	Purpose
Jenkins Dashboard	Visual interface for pipeline monitoring
GitHub Webhooks	Automatically triggers CI/CD jobs
Linux Networking	Basic server configuration and SSH setup
Prompt Engineering	Interacting with AI assistants effectively
Python, Java	Programming and scripting backend tools
CI/CD Pipelines	Ensures fast, reliable, and repeatable deployments

Steps for Project:

1. Set Up the Cloud Environment:
   - Launch EC2 Instances on AWS using Amazon Linux.
   - Set appropriate inbound rules for Jenkins and app ports.

2. Install Jenkins on EC2:
   - Configure Jenkins master server.
   - Install necessary plugins (Git, Pipeline, etc.).

3. Connect GitHub Repo:
   - Create and push source code to GitHub.
   - Configure GitHub Webhook to notify Jenkins on new commits.

4. Create CI/CD Pipeline:
   - Define Jenkinsfile for pipeline stages (build, test, deploy).
   - Configure Jenkins job to execute the pipeline.

5. Deploy Application:
   - Automatically pull latest code from GitHub.
   - Build and deploy the application to the target EC2 instance.

Reference Links:

- Jenkins Documentation: https://www.jenkins.io/doc/
- AWS EC2 Setup: https://docs.aws.amazon.com/ec2/
- GitHub Webhooks: https://docs.github.com/en/webhooks

- Spring Boot: https://spring.io/projects/spring-boot
- Prompt Engineering: https://platform.openai.com/docs/guides/prompt-engineering