The "Build a CI/CD Pipeline with Git, Jenkins, and EC2" project abstracts the complex process of software delivery into an automated, repeatable workflow. At its core, it represents the transformation of source code changes into deployed applications with minimal human intervention.

Abstraction:

- **Source Code Changes → Deployed Application:** This is the fundamental abstraction. The intricate steps of compilation, testing, packaging, and deployment are hidden behind a continuous, automated flow triggered by a simple code commit.
- **Version Control as the Trigger:** The Git repository acts as the single, authoritative source of truth. Any modification to this source is the abstract signal that initiates the entire pipeline.
- **Jenkins as the Orchestrator:** Jenkins abstracts the various build, test, and deployment tools and scripts into a unified automation engine. It handles the scheduling, execution, and monitoring of all pipeline stages.
- **EC2 as the Execution and Deployment Environment:** EC2 abstracts the underlying physical hardware, providing scalable and configurable virtual machines for hosting Jenkins itself and for serving as the target environment for the deployed application.
- **Pipeline as Code:** When using declarative pipelines, the Jenkinsfile abstracts the entire CI/CD workflow into a version-controlled script, making the process itself a programmable and auditable artifact.

In essence, the project provides an abstracted view where developers focus on writing code, and the pipeline ensures that their changes are reliably and efficiently delivered to end-users, minimizing the cognitive load associated with manual deployment processes.

S.V.Raghavendra

24P31F00G1

Aditya College of Engineering and Technology