Jenkins

Jenkins is an open-source automation server widely used for building, deploying, and automating any project. It facilitates Continuous Integration (CI) and Continuous Deployment (CD) by automating the software development processes.

Key Features:

- Automation: Jenkins automates various parts of the software development process, from building code to testing and deployment.
- Extensibility: A vast number of plugins are available, allowing Jenkins to integrate with a wide range of tools and technologies.
- Distributed Builds: Jenkins supports the distribution of build and test tasks across multiple machines, improving efficiency.
- Easy Configuration: It uses a simple web interface for configuration, making it accessible to both developers and administrators.
- Monitoring and Notifications: Jenkins provides real-time monitoring of builds and can notify teams about build results or failures.

Core Concepts:

- Job: The fundamental unit of work in Jenkins. A job defines a single task, such as building code, running tests, or deploying applications.
- Build: The process of compiling source code, running tests, and producing executable software artifacts.
- Node: A machine (physical or virtual) that executes tasks or builds. Jenkins can distribute tasks to multiple nodes for parallel execution.
- Workspace: The directory where Jenkins stores files for a specific job during its execution.

Continuous Integration (CI):

Jenkins promotes CI by automatically building and testing code changes whenever a developer commits code to the version control system (e.g., Git). This helps identify integration issues early in the development process, improving code quality.

Continuous Deployment (CD):

Jenkins supports CD by automating the deployment process after successful builds and tests. It can deploy applications to various environments, such as development, staging, and production.

Jenkins Workflow:

- Code Commit: Developers commit code changes to the version control system.
- Trigger: Jenkins monitors the version control system for changes and triggers a build when new code is committed.
- Build: Jenkins compiles the source code, runs tests, and produces artifacts.
- Test: Automated tests are executed to ensure code quality and identify issues.
- Deploy: If tests pass, Jenkins can deploy the application to the specified environment.
- Monitor: Jenkins monitors the deployed application and notifies teams about the build status.

Jenkins Pipeline:

Jenkins Pipeline is a suite of plugins that allows the definition and management of code delivery pipelines. It enables the creation of complex, multi-stage pipelines with code written in a domain-specific language (DSL).