Jenkins

Last updated by | Shakibe Hasan | Jun 19, 2023 at 4:58 PM GMT+6

What is Jenkins?

Jenkins is an open-source automation server that is widely used for continuous integration and continuous delivery (CI/CD) processes in software development. It helps automate the building, testing, and deployment of software applications, making it easier to maintain and deliver high-quality software efficiently.

What is Jenkins used for

- **Continuous Integration:** Jenkins can automatically build and test software projects whenever changes are committed to a version control system (e.g., Git). It helps identify integration issues early by regularly merging code changes from multiple developers into a shared repository and running tests against the integrated codebase.

- **Continuous Delivery/Deployment:** Jenkins enables automating the delivery and deployment of applications to various environments, such as development, staging, and production. It can trigger the deployment process based on predefined conditions, allowing for faster and more reliable software releases.

- **Build and Test Automation:** Jenkins provides a wide range of plugins and integrations with other tools, allowing you to automate various build and test tasks. It can compile source code, run unit tests, perform static code analysis, generate reports, and execute custom scripts or commands as part of the build process.

- **Scalability and Distribution:** Jenkins supports distributed builds, enabling you to distribute build and test workloads across multiple machines or agents. This helps in scaling your automation infrastructure to handle large projects or multiple concurrent builds.

- **Extensibility and Customization:** Jenkins has a vast ecosystem of plugins that extend its functionality and allow integration with various tools, such as version control systems, bug tracking systems, and cloud platforms. It offers flexibility to customize and configure the automation workflows to match specific project requirements.

How Does Jenkins Work?

Jenkins is an automation server that follows a master-agent architecture. The core of Jenkins is the Jenkins master, which manages the system configuration, schedules and coordinates jobs, and monitors the overall status of the automation environment. The Jenkins master also provides a web interface for users to interact with and configure the system.

1. **Setup and Configuration:** After installing Jenkins, you set it up by configuring system settings, such as specifying the location of your source code repository, defining build tools and environments, and installing plugins for additional functionality.

2. **Creating Jobs:** In Jenkins, you define jobs, which are the units of work that automate specific tasks in your software development process. A job typically includes instructions on how to retrieve source code, build the application, run tests, and perform other actions. Jobs can be created through the Jenkins web interface or by defining them programmatically using Jenkinsfile or pipeline scripts.

3. **Triggers and Scheduling:** Jenkins allows you to define triggers for jobs to determine when they should be executed. Jobs can be triggered based on events such as source code changes (e.g., commits to a version control system) or scheduled to run at specific times or intervals.

4. **Distributed Builds:** Jenkins supports distributed builds by leveraging agent nodes. Agent nodes can be configured on different machines or environments to offload the build and test workload from the Jenkins

master. Jenkins assigns jobs to available agent nodes, which then execute the tasks defined in the job configuration.

5. **Build Execution:** When a job is triggered or scheduled, Jenkins retrieves the necessary source code and dependencies, sets up the build environment, and executes the build steps defined in the job configuration. This may involve compiling code, running tests, performing static code analysis, generating reports, and more.

6. **Monitoring and Reporting:** Throughout the build process, Jenkins captures and displays real-time logs, output, and test results. It provides visibility into the status and progress of jobs, allowing developers and administrators to monitor the build and identify any issues or failures.

7. **Notifications and Integrations:** Jenkins can send notifications to relevant parties via email, instant messaging, or other communication channels to keep them informed about build status or failures. It also integrates with other tools and services, such as issue tracking systems, version control systems, and cloud platforms, enabling seamless interactions and automation across the development ecosystem.