Introduction to Jenkins

What is Jenkins?

- Jenkins is an open-source automation server used for Continuous Integration and Continuous Deployment (CI/CD).
- It automates building, testing, and deploying software applications.
- Supports integration with various tools like Git, Docker, Kubernetes, and more.

Benefits of Jenkins

- Automates repetitive tasks in software development.
- Improves code quality with continuous testing.
- Allows integration with multiple tools.
- Provides a centralized platform for DevOps workflows.
- Scales easily to meet project requirements.

Jenkins Architecture

- Master-Slave architecture for distributed builds.
- Jenkins Master: Controls the execution of jobs and schedules builds.
- Jenkins Agents (Slaves): Execute build jobs assigned by the master.
- Uses web-based UI for job configuration and monitoring.

Installing Jenkins

1. Download Jenkins from the official website: https://www.jenkins.io
2. Install Java (JDK 11 or later recommended).
3. Run Jenkins as a service using `java -jar jenkins.war`.
4. Access Jenkins via `http://localhost:8080`.
5. Install necessary plugins for SCM, build tools, and deployment.

Jenkins Pipeline

- A Jenkins Pipeline is a script defining a sequence of CI/CD steps.
- It is written in Groovy and supports automation workflows.
- Two types:
  - Declarative Pipeline: Simpler and more structured.Scripted Pipeline: More flexible but complex.

Jenkinsfile Example

Example of a simple Declarative Pipeline:

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                echo 'Building the application...'
            }
        }
        stage('Test') {
            steps {
                echo 'Running tests...'
            }
}
        stage('Deploy') {
          steps {
                echo 'Deploying the application...'
            }
 }
   }
}
```

Integrating Jenkins with Git

- Install the Git plugin in Jenkins.
- Configure Git repository under 'Source Code Management'.
- Set up webhook for automatic build triggers.
- Example: `git clone <repository-url>` in a Jenkins job.

Jenkins with Docker

- Jenkins can be used to build and deploy Docker containers.
- Steps:
  - Install the Docker plugin in Jenkins.
  - Create a `Dockerfile` to define the application environment.
  - Use Jenkins pipeline to build and push Docker images.
  - Deploy containers to Kubernetes or cloud platforms.

Common Jenkins Plugins

- Git Plugin - Integrates Git repositories with Jenkins.
- Pipeline Plugin - Enables Pipeline as Code (Jenkinsfile).
- Docker Plugin - Supports containerized builds.
- SonarQube Plugin - Helps with code quality analysis.
- Blue Ocean Plugin - Provides a modern UI for pipelines.

Conclusion

- Jenkins automates CI/CD workflows for software development.
- Pipelines enable efficient build, test, and deployment automation.
- Integration with Git, Docker, and other tools enhances capabilities.
- Continuous learning and practice will help in mastering Jenkins.