**Jenkins**

**Introduction to Jenkins**

Jenkins is an open-source automation tool that facilitates Continuous Integration (CI) and Continuous Deployment (CD). It automates various stages of the software development lifecycle, eliminating manual intervention and ensuring faster and more reliable software delivery.

**Key Features of Jenkins**

* **Automation Tool** – Automates projects without human interference.
* **CI/CD Tool** – Supports Continuous Integration and Continuous Deployment.
* **Developed by Hudson** – Jenkins was initially introduced by the Hudson project.
* **Plugin-Based Architecture** – Offers 1000+ plugins to integrate with various DevOps tools.
* **Master-Slave Architecture** – Uses multiple nodes/slaves for efficient workload distribution.
* **Security and Backup** – Provides high security, user management, and backup capabilities.
* **Supports Multiple Environments** – Integrates with Development, QA, UAT, and Production environments.

**Understanding CI/CD in Jenkins**

**Continuous Integration (CI)**

CI is the process of integrating code changes frequently, ensuring that each change is tested before deployment. It involves:

* Automating code integration with various DevOps tools.
* Running unit tests, functional tests, and regression tests.
* Providing rapid feedback to developers.

**Continuous Deployment (CD)**

CD is the process of deploying applications to the production environment seamlessly. It involves:

* Deploying code to end users without manual intervention.
* Ensuring an error-free application.
* Enhancing productivity and customer satisfaction.

**Need for CI/CD**

* **Overcomes Drawbacks of Waterfall & Agile Models** – Ensures faster requirement adaptation and acceptance.
* **Heart of DevOps** – Jenkins integrates multiple DevOps tools for automation.
* **Reduces Deployment Time** – Manual deployment takes months, while CI/CD pipelines complete deployment in days.
* **Ensures Code Quality** – Conducts static code analysis, vulnerability checks, and end-to-end automation testing.
* **Supports Various DevOps Tools** – Integrates with Git, Maven, Docker, Kubernetes, Ansible, Terraform, SonarQube, and Nexus.

**Jenkins Jobs and Builds**

* **Jobs/Projects** – Define automation tasks.
* **Builds** – Execution of jobs, also known as development.
* **Types of Jobs in Jenkins**:
  1. **Freestyle Jobs** – Configured using the Jenkins dashboard.
  2. **Pipeline Jobs** – Created using Groovy scripts and Jenkinsfiles in GitHub.
  3. **Parameterized Jobs** – Allow user-defined inputs before execution.
* **Sequential (Upstream & Downstream) and Parallel Jobs** – Supports both execution types for efficiency.

**Jenkins Architecture**

* **Master-Node (Slave) Architecture**
  + **Master** – Controls and schedules jobs.
  + **Nodes/Slaves** – Execute assigned tasks.

**Jenkins Configuration Files**

1. **Jenkins Home Directory** – /var/lib/jenkins
2. **Installed Plugins** – /var/lib/jenkins/plugins
3. **Created Jobs List** – /var/lib/jenkins/workspace
4. **Nodes Info** – /var/lib/jenkins/nodes
5. **Jenkins Log Info** – /var/lib/jenkins/log
6. **List of Jobs** – /var/lib/jenkins/jobs
7. **Created Users List** – /var/lib/jenkins/users

**Real-Time Jenkins Environments**

1. **Development** – Code development and initial integration.
2. **Quality Assurance (QA)** – Ensures software quality.
3. **User Acceptance Testing (UAT)** – Confirms software meets user expectations.
4. **Production** – Fully tested application deployed to end users.

**Importance of Environments**

* Ensures an error-free application.
* Improves productivity.
* Enhances customer satisfaction.

**Jenkins Installation Steps (AWS EC2 Instance)**

1. **Launch an EC2 Instance** (Ubuntu or Amazon Linux recommended).
2. **Install Java** – Required for Jenkins.
3. **Install Git** – For version control.
4. **Install Maven** – For build management.
5. **Install Jenkins** – Download and install the latest Jenkins version.
6. **Start Jenkins** and retrieve the initial admin password:
7. cat /var/lib/jenkins/secrets/initialAdminPassword
8. **Configure Jenkins** – Set up plugins, users, and jobs.

**CI/CD Pipeline Key Features**

**CI Pipeline**

* **Automatic Build & Tests** – Automates testing and feedback.
* **Rapid Feedback** – Ensures quick identification of issues.

**CD Pipeline**

* **Manual Deployment** – Provides control over deployments.
* **Deploy-Ready Code** – Ensures the code is ready for production.

**Jenkins Plugin Installation Methods**

1. **From Jenkins UI** – Using the dashboard.
2. **Manually Uploading Plugins** – Upload .hpi or .jpi files.
3. **Using CLI Commands** – Automate plugin installation via scripts.