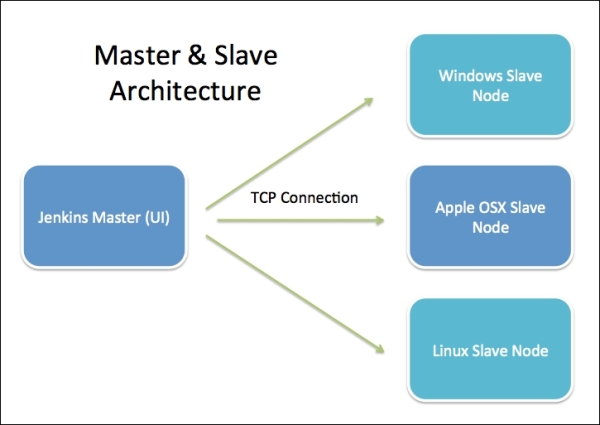
**What is Jenkins?**

* Jenkins is an open-source automation tool written in Java used mainly for Continuous Integration (CI) and Continuous Delivery (CD).
* It helps developers to automatically build and test code every time changes are made, catch errors early in the development cycle and Deploy applications faster by automating the entire process.
* It Supports Plugins for tools like Git, Maven, Docker, HTML Publisher, EC2, etc.
* Works with any programming language and also enables Pipeline automation for build, test, deploy, and more.

**What is Continuous Integration?**

* Continuous Integration (CI) is a software development practice where developers frequently commit their code changes to a shared repository.
* Each time a change is committed, an automated build and test process runs to ensure that the new code integrates well with the existing codebase.
* This practice helps detect errors early, improves code quality, and reduces integration issues.

**Jenkins Architecture:** Jenkins follows Master-Slave architecture to manage distributed builds. In this architecture, slave and master communicate through TCP/IP protocol.



**Jenkins Master**

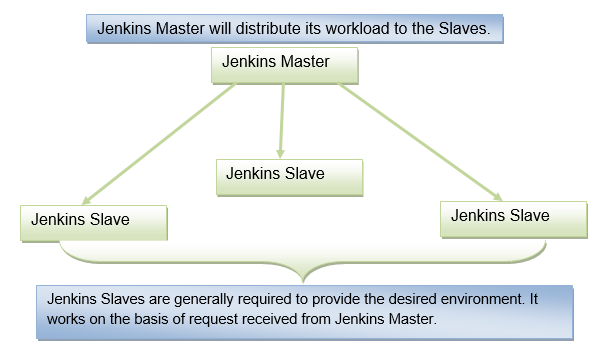
* Jenkins Master is the main server responsible for managing the overall Jenkins environment.
* It provides a web dashboard interface (usually on port 8080) which is launched from a WAR file. Through this dashboard, users can create and configure jobs or projects.
* The master handles key tasks like scheduling build jobs, assigning them to available slave nodes for execution, and recording the results of those builds.
* It also monitors the health and availability of slave nodes, bringing them online or offline as needed.
* Although Jenkins Master mainly coordinates tasks, it can also run builds directly when required.
* Additional nodes can be added using their IP address, username, and password. They can connect through methods like SSH, JNLP, or Webstart.

**Jenkins Slave:**

* Jenkins Slave (also called Node) is a machine configured to run build jobs assigned by the Jenkins Master. It does not have a web interface and is purely used for executing tasks.
* A specific project can be set to run on a particular slave, a type of slave, or Jenkins can automatically assign the job to the next available one.
* Slaves help distribute the workload and enable parallel job execution, improving performance and scalability.
* Since Jenkins is Java-based, both master and slave machines can run on any operating system, including Linux, Windows, or Mac.
* Slaves connect to the master using methods like SSH, JNLP, or Webstart, based on configuration.

**Jenkins Web Interface**

* The **Jenkins UI** is accessible through a web browser. Default URL: http://localhost:8080
* Through the UI, users can:
  + Create and configure jobs, Install and manage plugins
  + View build results and logs
  + Manage credentials, nodes, and system configuration



**How Jenkins Works (Workflow):**

1. Developer commits code to a version control system (like Git).
2. Jenkins detects the change (via webhook or polling).
3. It triggers a job on a selected agent.
4. The agent executes the job (build/test/deploy).
5. Results are sent back to the master, and users can view the status on the dashboard.
6. Jenkins can notify teams via email/Slack/etc

**Advantages of Jenkins**

* It is an open-source tool.
* It is free of cost.
* Easily configurable.
* It is built in java and hence it is portable.
* It is platform independent.

**Disadvantages of Jenkins**

* Its interface is out dated and not user friendly compared to current user interface trends.
* Requires some skills as server administrator to monitor its activity.
* CI regularly breaks due to some small setting changes.