



Jenkins

By
Venkat
Rise 'n' Shine Technologies

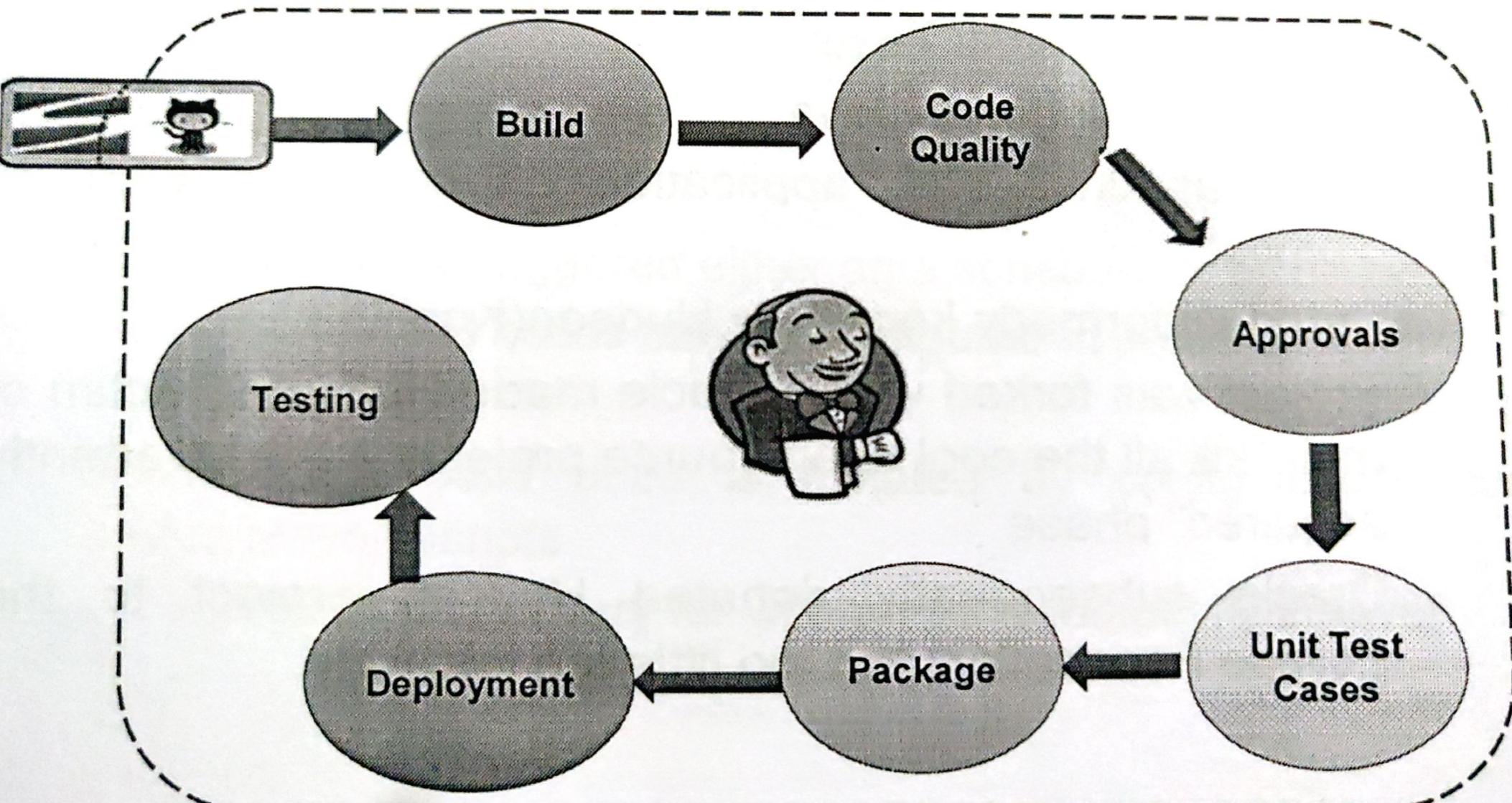
What is Continuous Integration and Delivery?

- **Continuous Integration (CI):**
 - It is a practice that requires developers to integrate code into a shared repository several times a day.
 - Every check in will result into an automated build
- **Continuous Delivery:**
 - It is the capability to do automated deployments

Why we use CI?

- CI can help you keep tabs on the health of your code base
- Automatically monitoring code quality
- Automatically monitoring code coverage metrics
- runs unit tests and rejects the build if they fail
- Deploy builds to other servers
- Keep maintenance costs low
- Automate approval process

CI Server:



Tools

- **Build Automation**
 - Maven
- **Continuous Integration**
 - Jenkins/Hudson, Cruise Control, Team City, Bamboo, Build Forge
- **Deployment Automation**
 - Linux
 - Shell Scripting
 - DevOps – Ansible, Puppet, Chef
- **Application Server**
 - Apache Tomcat, WebLogic
- **Version Control System**
 - Subversion, Git
- **Code Quality**
 - Sonar Qube, Cobertura

What's Jenkins?

- Continuous Integration (CI) server
 - (Vcs + Build + Unit Tests)
- Open source, written in Java
 - Runs as Java web application on container such as Tomcat
- The product formerly known as Hudson(Koshuko)
 - Project was forked when Oracle made Hudson a victim of their "kill all the cool open source projects we inadvertently acquired" phase
 - Oracle subsequently donated Hudson project to the Eclipse Foundation, but too little too late

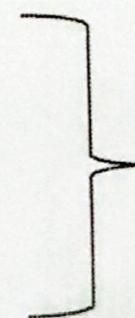
Advantages

- Build Automation
- Integration with VCS and Build Tools
- Scheduling
- Reporting
- History
- Artifacts Availability

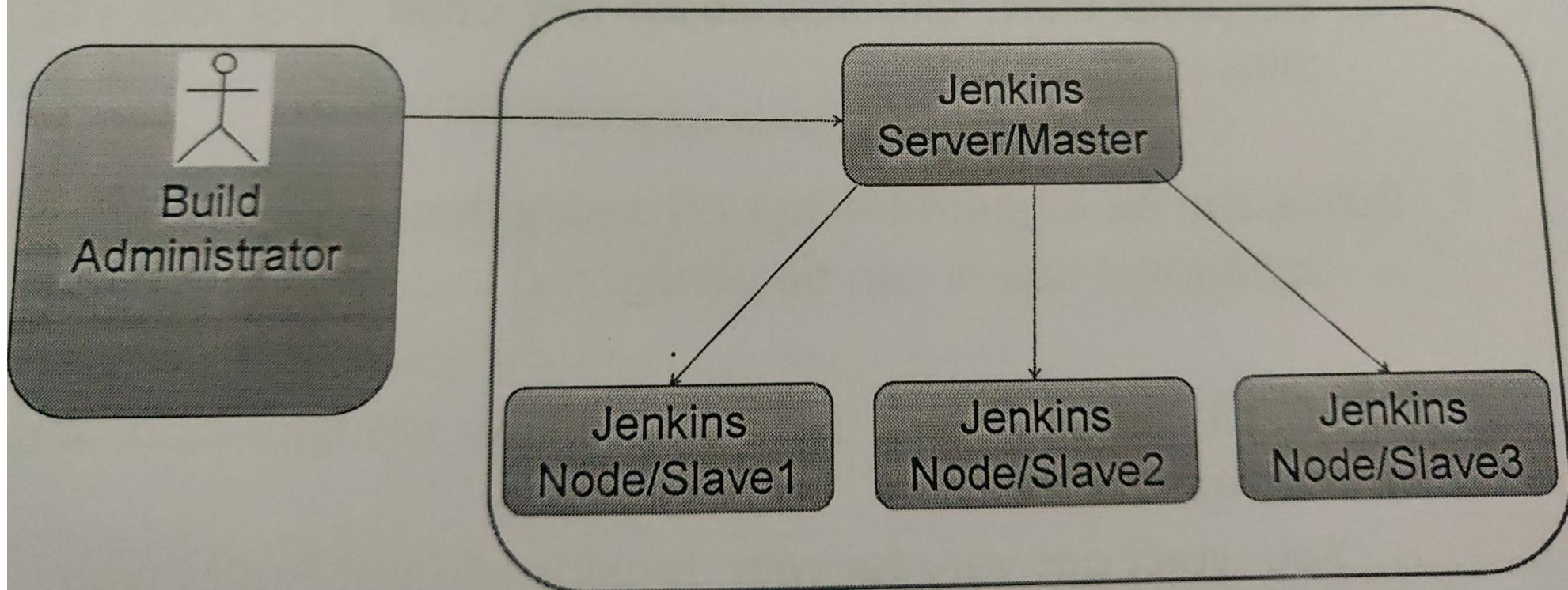
How Does Jenkins Work?

- Integrates with source control
 - Supports SVN, Git, and about any other SCM tool you can think of via plugins
- Builds can be triggered either on a schedule or by hitting a URL
 - Scheduled builds can be configured in Jenkins
- What the build does is dictated by configuration plus an Ant/Maven scripts
 - You won't get very far with Jenkins without Ant/Maven

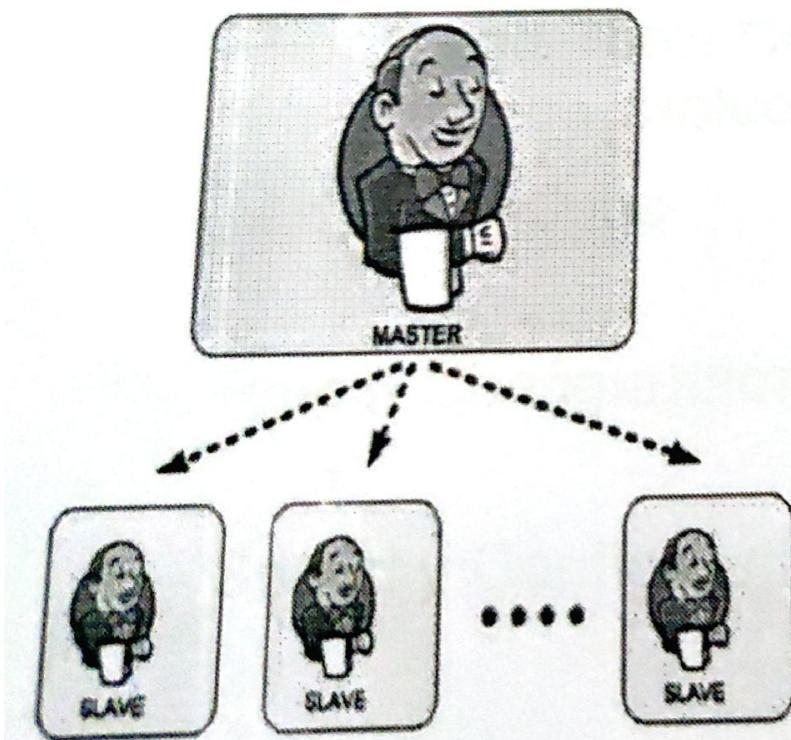
Jenkins Architecture

- Master - Slave Architecture
 - Example:
 - 12 Projects
 - The build Time of each project – 3 hrs
 - Sequence:
 - $12 * 3 = 36\text{hrs}$ (Almost one and half day)
 - Parallel:
 - 3 Machines
 - $1 + 1 + 1 = 3$ projects in 3hrs
 - $1 + 1 + 1 = 3$ projects in 3hrs
 - $1 + 1 + 1 = 3$ projects in 3hrs
 - $1 + 1 + 1 = 3$ projects in 3hrs
- 
- In 12 Hrs (all projects build can be completed)

Jenkins Master – Node/Slave



Jenkins Master – Node/Slave



Master :

- 1. Schedule Build Job**
- 2. Dispatches Builds to the Slave for Actual Job Execution**
- 3. Monitoring the Slave and recording the build Results.**

Slave :

- 1. Execute Builds Jobs dispatched by master**

Jenkins Jobs

- A job in Jenkins defines a sequence of tasks for Jenkins to perform typically project-oriented activities.
- When job is triggered, Jenkins looks for an Ant/ Maven script and runs it on the basis of process.
- Job tasks can be divided into:
 - Pre-tasks
 - Pull the code from repo, clean the workspace
 - Tasks
 - Perform the build activities
 - Post-Tasks
 - e.g. create WAR, copy files to another location, etc.

Jenkins : Executor

- ❖ Executor is separate Stream of Builds to be run on a Node in Parallel.
- ❖ A Node can have one or more Executors.

Jenkins Installation

- Download Jenkins from <https://jenkins.io>
- Pre-requisite: Java should be installed
 - \$ java –version
- Start the jenkins:
 - Go to jenkins.war file
 - Java –jar jenkins.war
 - Jenkins Console URL – http://ip_address:8080

Note: Default port is 8080 if it is required custom port then
pass –httpPort=9000
-- Default JENKINS_HOME=~/.jenkins/

Activities of Jenkins

- -- Master and Slave Configuration
- -- Global and Tools Configuration
- -- Managing Jobs
- -- Plugin Configurations
- -- Security Configurations

Steps to Create Master/Slave

- Prerequisite:
 - Master and Slave nodes should have the JDK installed
 - SSH should be configured between the Master and Slave
- How to configure Master with Slave:
- Go to Manage Jenkins --> Nodes --> New Node --> Name
- Create a Node
 - No of Executors
 - Remote FS Root
 - Launch Method with host and Credentials

Managing Jobs/Items

- Creating jobs
- Configuring Jobs
- Build and Report

Jenkins Job Scheduling

- Time Based Scheduler
- Poll based SCM
- Build after other projects build

Jenkins Job Scheduling

- 1. Time Based Schedule
 - It is like as crontab in Linux and Task Manager in windows
 - The format as follows...
 - * * * * *
 - 1 – Minutes (0 to 59)
 - 2 - Hours (0 to 23)
 - 3 - Day of the Month (1 to 31)
 - 4 - Month of the Year (1 to 12)
 - 5 - Day of the Week (0 to 7) 0 or 7 are sundays
 - Example: (Night 1 am) - * 1 * * *
 - Morning 4:30 - 30 4 * * *
 - Every Month 15th Morning 9 am - * 9 15 * *

Administration Tasks

- Security
 - Authentication (Valid Credentials)
 - Authorization (Privileges)
- Users
- Access
- Plug-in

Jenkin Labels

