

# CI/CD AND Jenkins Learning Document

## What is CI/CD?

CI/CD stands for Continuous Integration and Continuous Delivery/Deployment. It is a set of practices that automate the process of software development.

## Continuous Integration (CI)

- Developers frequently push code to a shared repository (like GitHub).
- Jenkins checks and builds the code automatically after every push.
- Helps find and fix bugs early.

## Continuous Delivery (CD)

- Once CI is successful, the next step is to automatically prepare the app for release.
- Jenkins can run tests and package the application for deployment.

## Continuous Deployment

- Similar to CD but takes it one step further.
- Automatically deploys the application to production after successful tests.
- No manual approval required.

## Jenkins Role in CI/CD

- Jenkins automates the entire CI/CD pipeline.
- You can configure pipelines to build, test, and deploy code automatically.
- Helps reduce human error and speeds up software delivery.

## What is Jenkins?

Jenkins is an open-source automation tool. It helps developers to:

- Automatically build, test, and deploy their code
- Speed up software development using Continuous Integration (CI) and Continuous Delivery (CD)

## Jenkins Setup:

### Step 1: Check if Java is Installed

Command: `java -version`

Why: Jenkins runs on Java. You need Java installed.

### Step 2: Create a Jenkins Folder

Command: `mkdir C:\Jenkins`

Why: Keeps Jenkins files organized in one place.

### Step 3: Move into Jenkins Folder

Command: `cd C:\Jenkins\`

Why: So you can run Jenkins from this folder.

### Step 4: Start Jenkins with WAR File

Command: `java -jar jenkins.war`

Why: This starts Jenkins in your terminal.

Alternate (if port 8080 is busy): `java -jar jenkins.war --httpPort=9090`

### Step 5: Open Jenkins in Your Browser

Go to: `http://localhost:8080`

Use the initial password displayed in terminal or located at:

`C:\Users\<YourName>\.jenkins\secrets\initialAdminPassword`

## Jenkins Basics

### 1. Jenkins Dashboard

The main screen you see after logging in. Shows all jobs, their status, and build buttons. Like a control panel.

### 2. What is a Jenkins Job?

A Job is a task Jenkins performs (build, test, deploy). Freestyle is the most basic job type.

### 3. What is a Pipeline?

Pipeline is a script that defines the CI/CD flow (e.g., Build → Test → Deploy). Written in Groovy.

### 4. Master and Agent (Node)

Master: Main Jenkins server that controls everything.

Agent: Extra machines that help run jobs.

### 5. Scheduling Jobs – CRON Syntax

Use CRON to auto-run jobs:

H/15 \* \* \* \* → Every 15 minutes

@hourly → Every hour

@daily → Every day

## 6. Jenkins Plugins

Plugins add features (Git, Docker, etc). Install via: Dashboard → Manage Jenkins → Manage Plugins.

## 7. Jenkins Workspace

Each job has its own folder where it saves files, code, and results.

## Summary Table

Feature	Description
Dashboard	Main screen with job list.
Job	A task Jenkins runs(build, test, deploy).
Pipeline	Scripted automation of stages
Master	Main Jenkins controller
Agent	Worker that executes jobs
Plugin	Extra feature (like Git or Docker)
Workspace	Folder where job files are stored
CRON	Auto-schedule job timings