Perfect timing, Harish 👍  
Learning **Jenkins** is a must for an automation engineer, especially if you’re targeting CI/CD + DevOps awareness.  
Let me guide you step by step from **zero → hands-on**.

**🚀 Jenkins Learning Roadmap (From Scratch)**

**🔹 1. What is Jenkins?**

* Jenkins is an **open-source automation server**.
* It helps automate **build → test → deploy** processes.
* Mostly used for **Continuous Integration (CI)** and **Continuous Delivery (CD)** pipelines.

👉 Example:  
You commit Selenium code → Jenkins pulls it → runs Maven → executes TestNG tests → generates reports → optionally deploys results.

**🔹 2. Install Jenkins**

**Option A: Local installation (best for beginners)**

* Install **Java 11/17** (Jenkins requires Java).
* Download Jenkins .war → run:
* java -jar jenkins.war
* Open browser → http://localhost:8080

**Option B: Docker installation (for practice with containers)**

docker run -p 8080:8080 -p 50000:50000 jenkins/jenkins:lts

**🔹 3. Basic Jenkins Setup**

* Unlock Jenkins → create Admin user.
* Install **Suggested Plugins** (Git, Maven, Email, Pipeline, TestNG).
* Configure **JDK, Maven, Git** inside Jenkins (Manage Jenkins → Global Tool Configuration).

**🔹 4. Create First Job**

👉 Go to **New Item → Freestyle Project**  
Example: “Smoke Tests Job”

* **Source Code Management** → connect GitHub repo.
* **Build Tool** → choose Maven/Gradle.
* **Build Step** → mvn clean test.
* **Post-Build Action** → Publish TestNG or JUnit Reports.

Now every time you **click Build Now**, Jenkins will pull your repo, run tests, and show results.

**🔹 5. Run Selenium/TestNG Framework in Jenkins**

* Push your Java Selenium + TestNG project to GitHub.
* Create a Jenkins job with:
* mvn clean test -DsuiteXmlFile=testng-smoke.xml
* Reports appear in Jenkins console + HTML reports if configured.

**🔹 6. Automate with Triggers**

* **Poll SCM** → checks GitHub repo every X mins.
* **GitHub Webhook** → triggers Jenkins build immediately after commit.
* **Schedule Builds** (like cron jobs) → run daily/nightly regression at 0 0 \* \* \*.

**🔹 7. Pipeline (Jenkinsfile)**

Instead of GUI jobs, you can write **Pipeline as Code** (Jenkinsfile).

Example:

pipeline {

agent any

stages {

stage('Checkout') {

steps {

git 'https://github.com/harish/selenium-framework.git'

}

}

stage('Build') {

steps {

sh 'mvn clean install'

}

}

stage('Test') {

steps {

sh 'mvn test -DsuiteXmlFile=testng-regression.xml'

}

}

stage('Report') {

steps {

publishHTML([reportDir: 'test-output', reportFiles: 'index.html', reportName: 'TestNG Report'])

}

}

}

}

**🔹 8. Reports & Notifications**

* Integrate **Extent/Allure reports** → show fancy dashboards.
* Add **Email/Slack notification plugin** → send pass/fail results to team.

**🔹 9. Advanced (After Basics)**

* Distributed Builds → run Selenium on multiple nodes (Grid/Docker).
* Parameterized Jobs → pass environment, browser, or test type (Smoke/Regression).
* Integration with Docker, Kubernetes, AWS → CI/CD at scale.

**✅ Suggested 7-Day Plan for You**

**Day 1-2:** Install Jenkins, create Freestyle jobs.  
**Day 3:** Integrate with Git + Maven + TestNG project.  
**Day 4:** Add triggers (SCM, schedule).  
**Day 5:** Generate reports (Extent/Allure, HTML).  
**Day 6:** Write your first Pipeline (Jenkinsfile).  
**Day 7:** Add notifications + parameterized jobs.

👉 Harish, do you want me to prepare a **hands-on practice project for you** (like a Git repo with a simple Selenium TestNG framework + Jenkins integration), so you can directly practice without wasting time setting things up?