ChatGPT

Jenkins Beginner Guide: Learn Jenkins from Scratch with Exercises and Projects

---

Introduction to Jenkins

What is Jenkins?

- Jenkins is an open-source automation server.
- It helps automate parts of software development: building, testing, and deploying.
- Core to DevOps and CI/CD pipelines.

Why Use Jenkins?

- Automates repetitive tasks
- Works with Git, Docker, Kubernetes, Ansible
- Has thousands of plugins

Jenkins in Real Life (Analogy)

- Imagine Jenkins as a factory manager:
- Jobs: Tasks Jenkins performs (like making coffee, welding parts)
- Triggers: Alarms that start tasks (e.g., new commit)
- Pipelines: Assembly lines (build, test, deploy)

Installation & Setup

Prerequisites:

- Linux (Ubuntu/Debian or CentOS)
- Java (JDK 11 or above)

Ubuntu/Debian Installation:

```
sudo apt update
sudo apt install openjdk-11-jdk -y
wget -q -O - https://pkg.jenkins.io/debian/jenkins.io.key | sudo apt-key add -
sudo sh -c 'echo deb https://pkg.jenkins.io/debian-stable binary/ > /etc/apt/
sources.list.d/jenkins.list'
sudo apt update
sudo apt install jenkins -y
sudo systemctl start jenkins
sudo systemctl enable jenkins
```

Access Jenkins:

- URL: `http://localhost:8080`
- Password:

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

Your First Jenkins Job

Goal: Hello World

1. Click **New Item** > **Freestyle Project**
2. Name: `hello-job`
3. In **Build** > **Execute Shell**:

```
echo "Hello, Jenkins!"
```

1. Save & Build

Output:

- View Console Output to see the message.

💻 Automating with Git

Goal: Clone repo and run a script

1. Create Freestyle Project
2. Under **Source Code Management**, select **Git**
3. Enter Git URL
4. Under **Build Steps** > Execute Shell:

```
bash run.sh
```

1. Save and Build

Pipeline as Code

Jenkinsfile Example:

```
pipeline {
  agent any
  stages {
    stage('Build') {
      steps {
        echo 'Building...'
      }
    }
    stage('Test') {
      steps {
        echo 'Testing...'
      }
    }
    stage('Deploy') {
      steps {
        echo 'Deploying...'
      }
    }
  }
}
```

Create Pipeline Job:

- Choose **Pipeline** project
- In **Pipeline** section > use "Pipeline script from SCM" if using GitHub

---

Exercises

Exercise 1: Create a Job to List Files

- Create Freestyle Job `list-files`
- Build step: `ls -la`
- Run & check output

Exercise 2: Run Script from Git Repo

- Host a repo with `hello.sh`
- Job clones the repo and runs `bash hello.sh`

Exercise 3: Build with Parameters

- Add String Parameter `USERNAME`
- Script:

```
echo "Hello, $USERNAME!"
```

Exercise 4: Basic Pipeline

- Write Jenkinsfile that has 3 stages: Setup, Build, Finish

Projects

Project 1: Jenkins CI Pipeline for a Node.js App

- Clone from GitHub
- Install dependencies
- Run tests (`npm test`)
- Archive test results

Project 2: Docker Integration

- Jenkinsfile builds Docker image
- Push to Docker Hub

Project 3: Notification with Email or Slack

- Add post-build step to notify team

Project 4: GitHub Webhook Integration

- Auto-trigger job on GitHub push

🎓 Learning Plan Summary

Week	Topics
1	Jenkins setup, first job, UI navigation
2	Git integration, Shell scripts, Build triggers
3	Jenkinsfiles, Pipelines, Parameters
4	Docker, Email, Webhooks, Real Projects

Next Steps

- Learn Jenkins plugins (JUnit, GitHub, Docker, etc.)
- Integrate with GitHub Actions or Ansible
- Move to Jenkins on Kubernetes

📚 References

- https://www.jenkins.io/doc/
- https://plugins.jenkins.io/
- https://github.com/jenkinsci

You're now ready to build your DevOps automation career with Jenkins!