



## STEP 1: Update Your Fedora System

Open terminal and run:

```
sudo dnf update -y
```

👉 Updates all system packages.

---



## STEP 2: Install Java (Required for Jenkins)

Jenkins requires Java.

Check if Java is installed:

```
java -version
```

If not installed, install OpenJDK 17:

```
sudo dnf install java-17-openjdk -y
```

Verify again:

```
java -version
```

✓ You should see Java 17 version details.

---



## STEP 3: Add Jenkins Repository

Add Jenkins repo:

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

Import Jenkins key:

```
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

---



## STEP 4: Install Jenkins

```
sudo dnf install jenkins -y
```

---



## STEP 5: Start and Enable Jenkins

Start Jenkins:

```
sudo systemctl start jenkins
```

Enable at boot:

```
sudo systemctl enable jenkins
```

Check status:

```
sudo systemctl status jenkins
```

If you see **active (running)** → correct.

---

## 🔥 STEP 6: Allow Firewall (Very Important in Fedora)

Fedora blocks port 8080 by default.

```
sudo firewall-cmd --permanent --add-port=8080/tcp  
sudo firewall-cmd --reload
```

---



## STEP 7: Access Jenkins in Browser

Open browser inside VM:

<http://localhost:8080>



**Testing not crt one so  
don't use this**

**Perfect 🔥** Now we'll create a Jenkins project that pulls code from GitHub step-by-step in your Fedora VM.

We'll connect Jenkins with GitHub.

---

### PART 1: Install Git (If Not Installed)

First check:

```
git --version
```

If not installed:

```
sudo dnf install git -y
```

---

### PART 2: Install Git Plugin in Jenkins

#### **1** Open Jenkins in browser

<http://localhost:8080>

#### **2** Go to:

Dashboard → Manage Jenkins → Manage Plugins

<b>3</b>	Under	Available	Plugins
Search:			
	<b>Git Plugin</b>		
<input checked="" type="checkbox"/>	Install it (if not already installed)		

### PART 3: Create New Project (Freestyle Project)

<b>1</b>	Click	New	Item
<b>2</b>	Enter	project	name

Example:

**MyGitProject**

#### **3** Select:

**Freestyle Project**

#### **4** Click OK

---

### PART 4: Connect GitHub Repository

Inside project configuration:

- ◆ Step 1: Source Code Management

- Select Git
- In Repository URL paste your GitHub repo link

Example:

<https://github.com/username/repository.git>

If repo is public → no credentials needed.

If repo is private → Add credentials:

- Click Add
- Enter GitHub username + Personal Access Token

---

◆ Step 2: Branch

Default branch usually:

\* /main

(or \* /master if old repo)

---

✓ PART 5: Add Build Step (Optional for Lab)

Scroll down → Build → Add build step

Select:

Execute shell

Add simple command:

echo "Build Started"

ls

This will show pulled files.

Click Save

---

✓ PART 6: Build the Project

1	Click	Build	Now
2	Click build number → Console Output		

You should see:

Cloning the remote Git repository

If successful → repository files are pulled 🎉

---

🧠 Viva Questions (Very Important)

1 What is SCM?

SCM = Source Code Management.

2 How does Jenkins pull code?

Using Git plugin and repository URL.

3 Difference between public and private repo?

Public	→	no	credentials
Private → needs authentication			

4 What is a webhook?

Webhook automatically triggers Jenkins when code is pushed.

**5** Where does Jenkins store pulled code?

In:

/var/lib/jenkins/workspace/<project-name>

---

**BONUS (For Extra Marks in Lab)**

You can enable automatic build:

Go to project → Configure → Build Triggers  
Select:

**Poll SCM**

OR connect webhook from GitHub:

Repo → Settings → Webhooks → Add Jenkins URL

---

**Common Errors**

Repository URL wrong

Branch name wrong

Git not installed

Private repo without credentials

---

If you want next, I can teach:

- Setup GitHub Webhook
- Create Jenkins Pipeline Project
- CI/CD project example
- Connect Jenkins with Docker

Tell me what your lab next asks 😊💻🔥