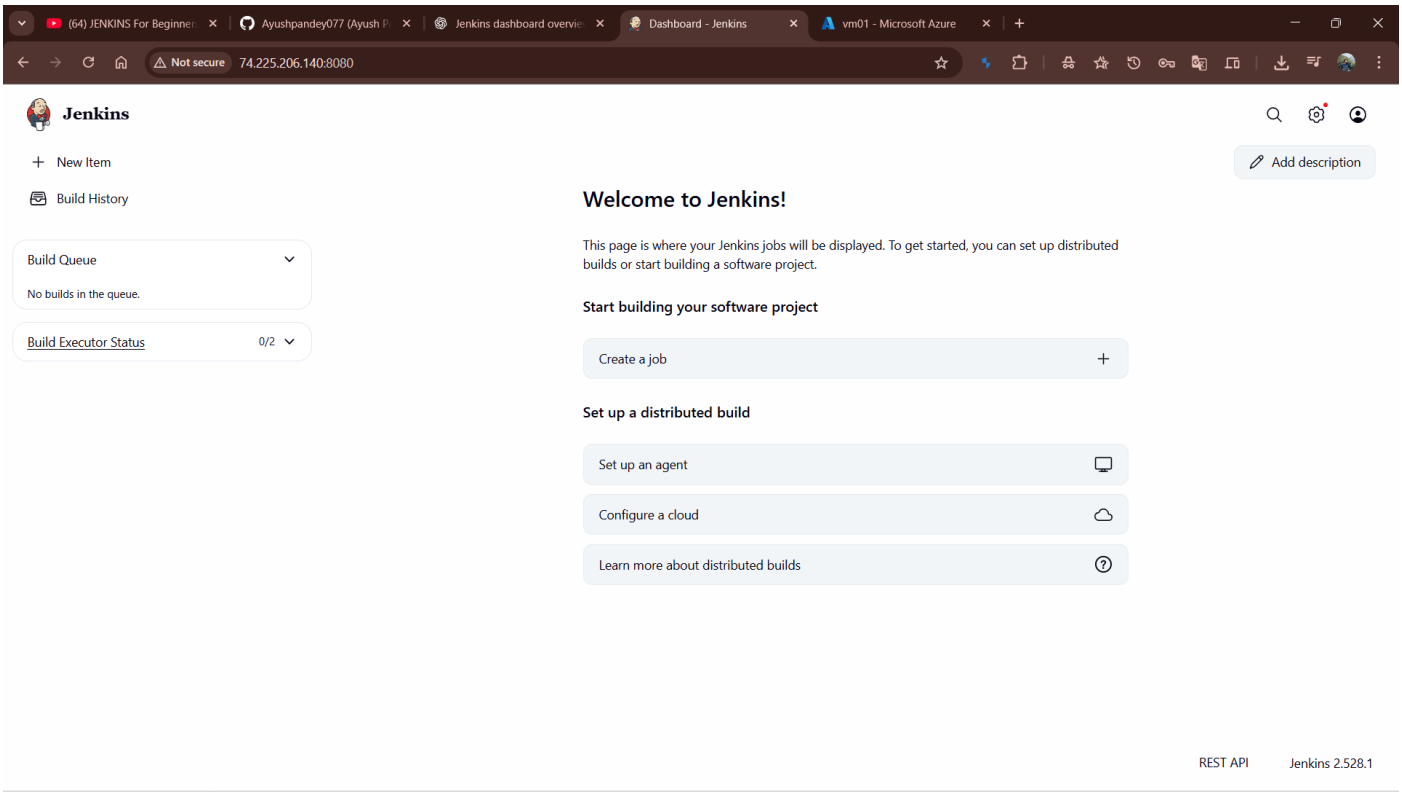
# Working with GitHub And Upload a Website template through Jenkins

## 🪜 Step-by-Step Jenkins Setup

## 🧩 Step 1: Open Jenkins

- Open in browser → http://localhost:8080
- Login with your Jenkins credentials.




## 🆕 Step 2: Create a New Job

1. Click on **“New Item”**
2. Enter a name → for example: WebApp-Build
3. Select **Freestyle project**
4. Click **OK**

**Point to be Remember: -** Install Node.js on Jenkins Server (Ubuntu)

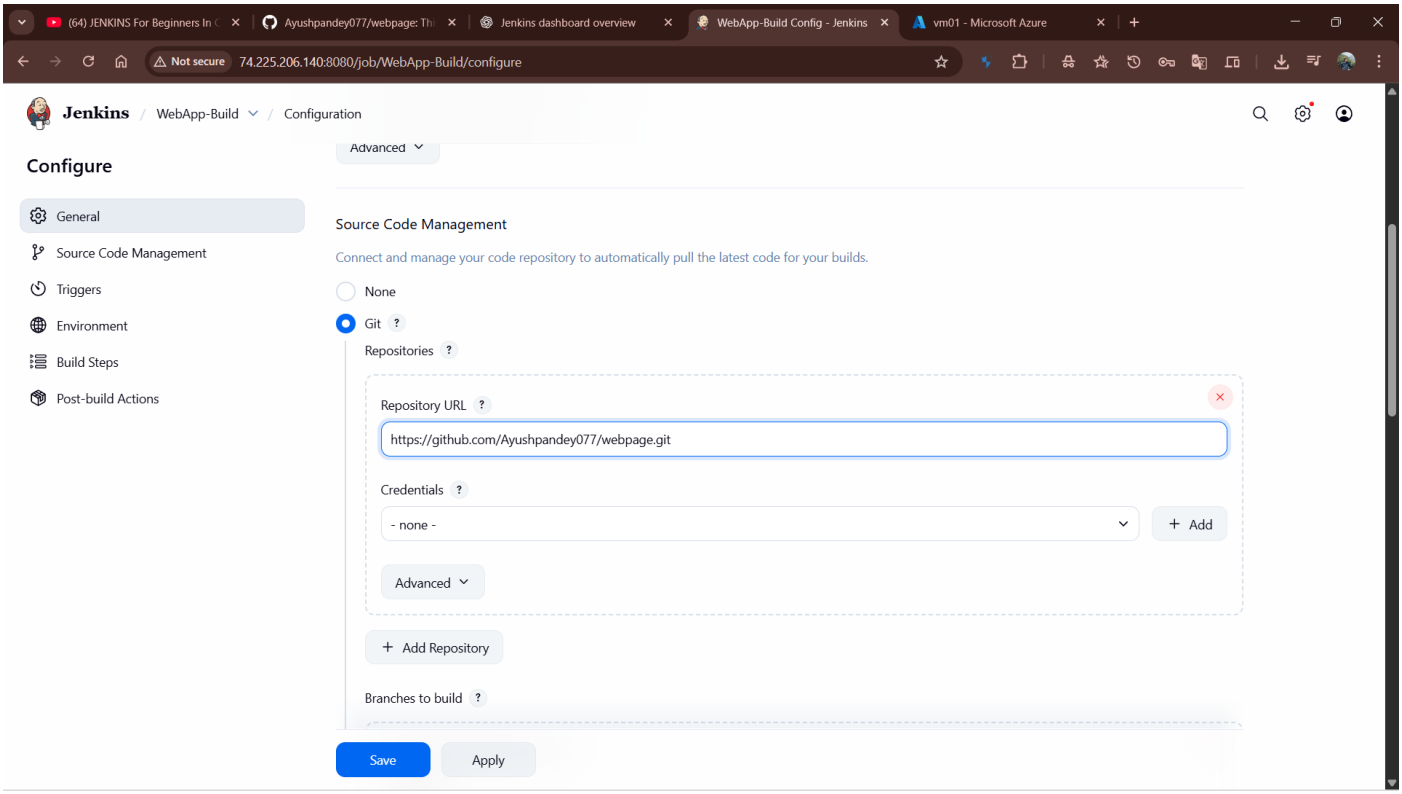
**Run these commands in your Ubuntu terminal (not in Jenkins):**

1. sudo apt update
2. sudo apt install -y nodejs npm

## ⚙️ Step 3: Configure the Job

- Source Code Management

1. Select **Git**
2. In the **Repository URL** field, paste your repo link:
3. If it's private, click **Add Credentials** → enter your GitHub username and token.

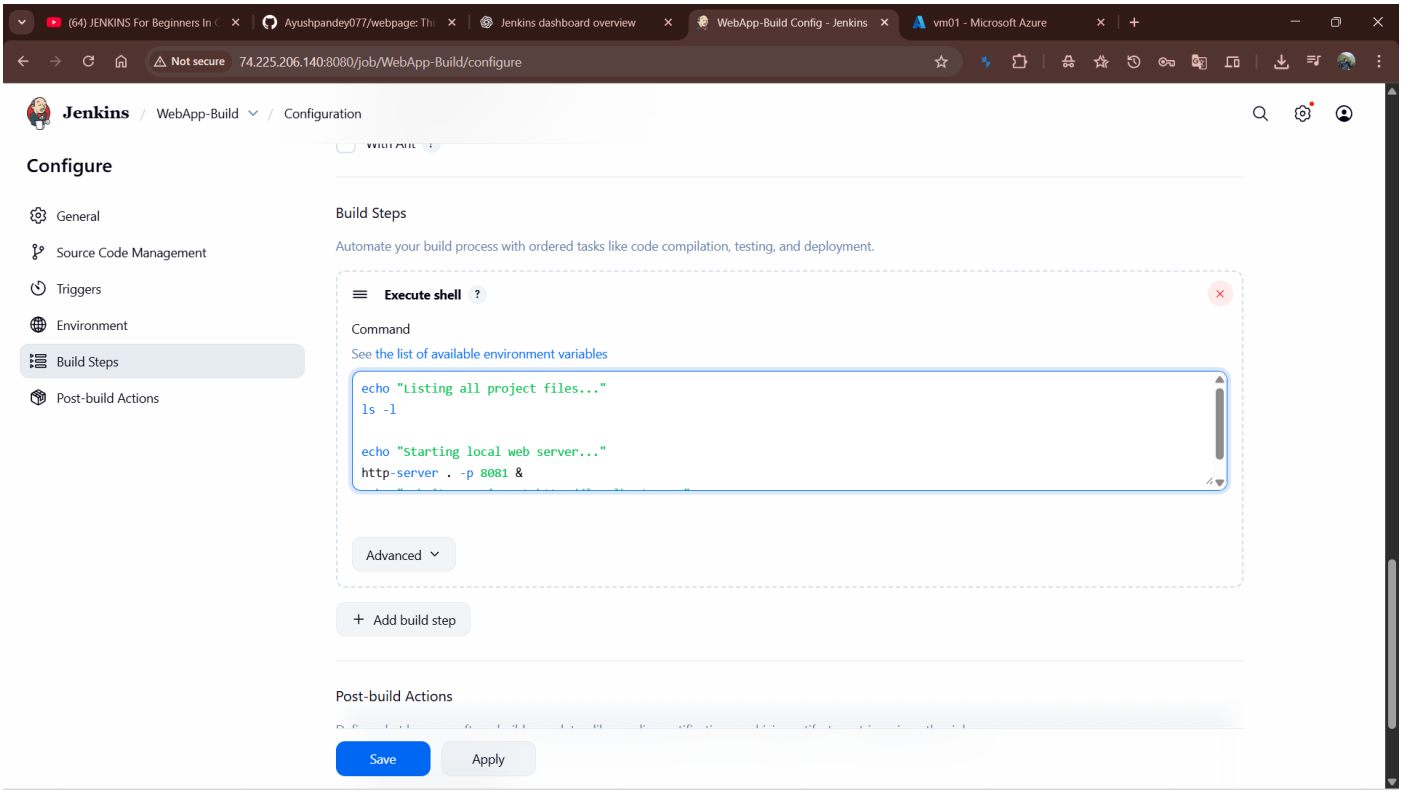



- Build Section

1. Now click on **Add build step → Execute shell**
2. In the box, type the commands below 👇

## 💻 Shell Commands

If your project is **pure HTML/CSS/JS (no Node.js)**:




## 📝 Explanation:

1. ls -l → lists all your files (for verification)
2. npm install -g http-server → installs a simple Node-based web server
3. http-server. -p 8081 & → serves your site in background on port 8081

## ✅ Step 4: Save and Build

1. Click **Save**
2. On the left side, click **Build Now**
3. Check **Console Output**

---

**Point to be Remember: -** Before Run Above Command Run given below Command on the server.

**Run these commands on your Ubuntu terminal (not inside Jenkins):**

- ❖ sudo npm install -g http-server
- ❖ which http-server

**You should see something like:**

/usr/bin/http-server

**Then click Build Now again → it should succeed ✅**

## Network Rules: -

Server Running on Cloud (like AWS or Azure)
**If Jenkins is hosted on a cloud VM:**

- Go to your VM's security group / network rules
- Allow Inbound rule for TCP Port 8081

## Confirm http-server is Running

- sudo lsof -i :8081

If you see a process with node or http-server, that means it's active.

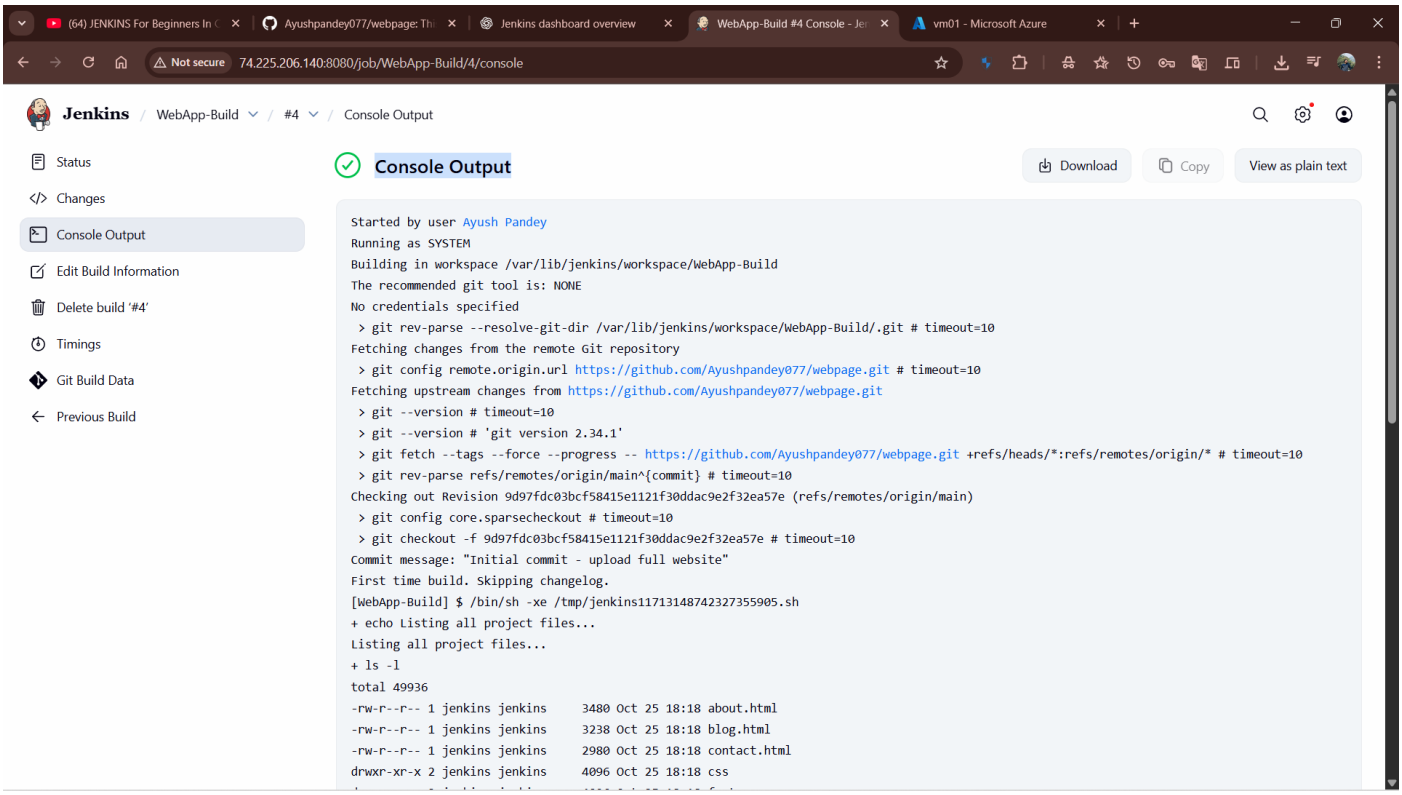
If not, it may have stopped after Jenkins finished the job.
To keep it running after the job ends, you can run it like this:

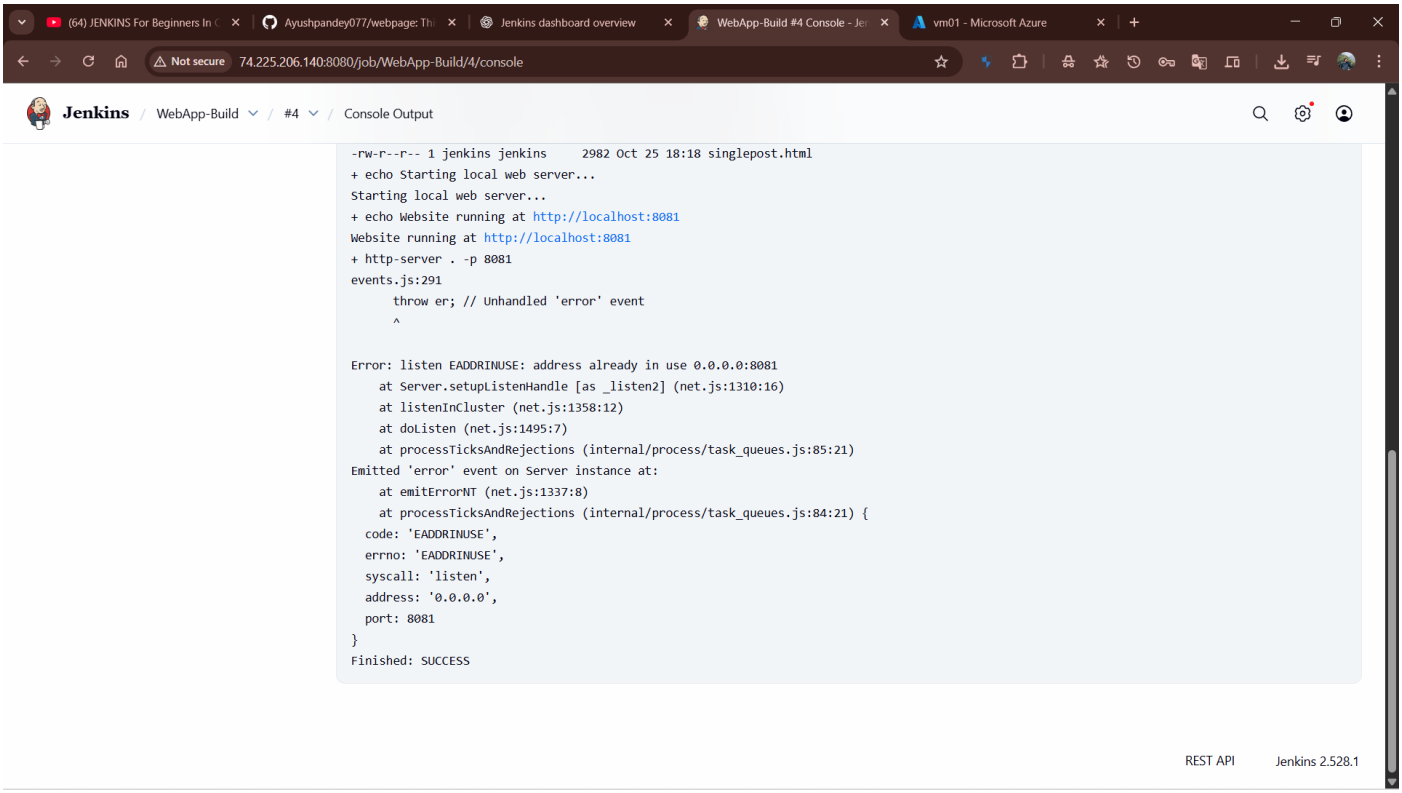
- nohup http-server. -p 8081 &

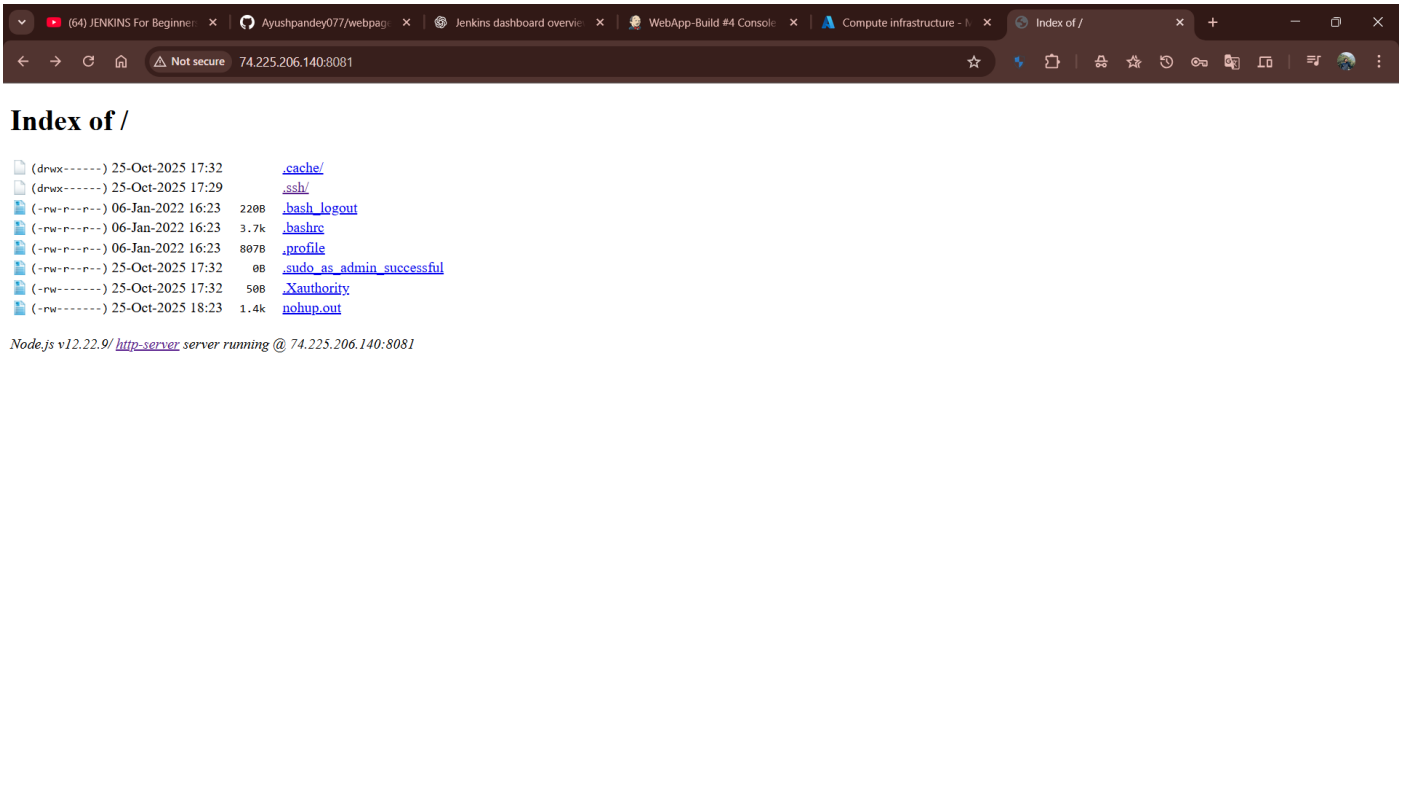
**This keeps it running even after Jenkins exits.**

## Result: -

**Console Output:-** Success






**Perfect 👏 — that output actually means your Jenkins job worked successfully this time!**