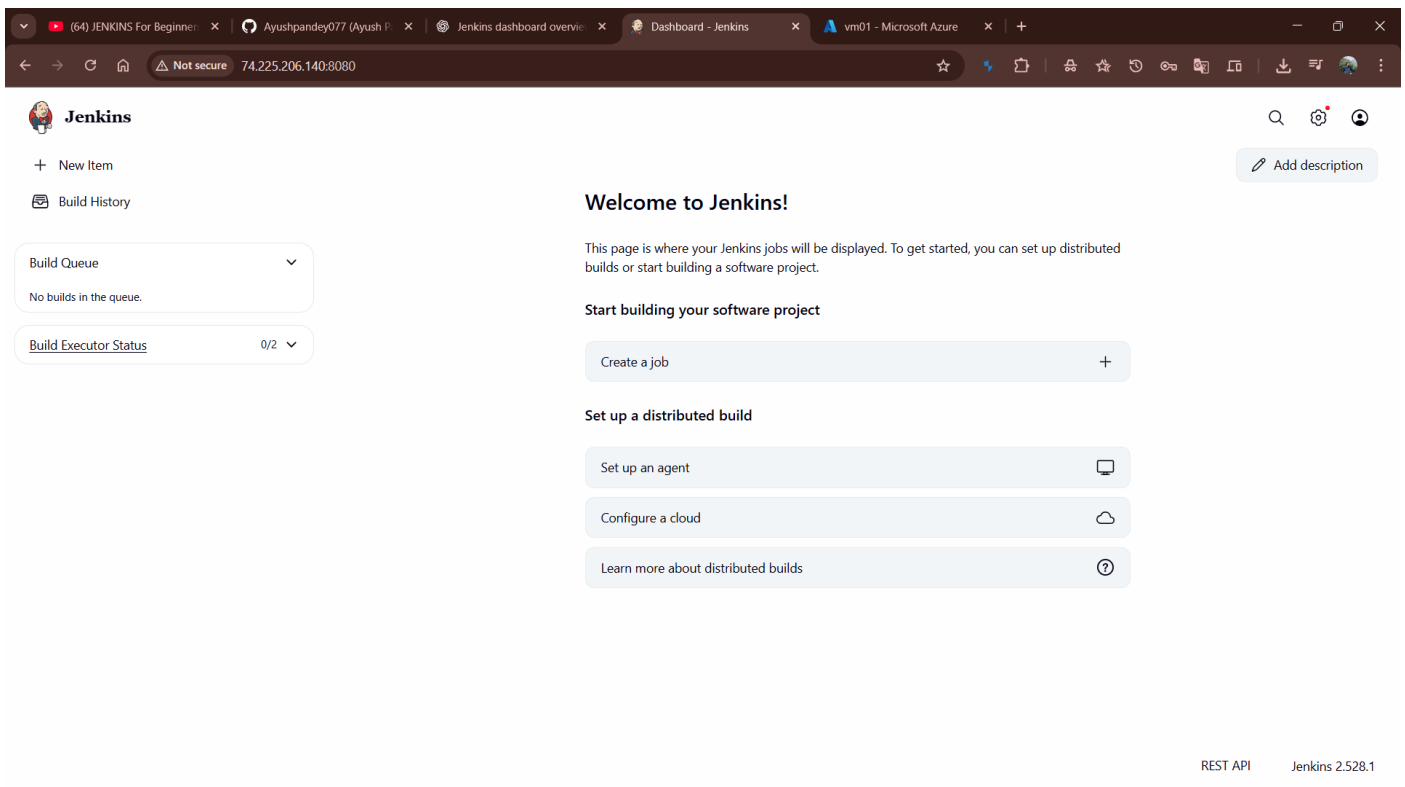


# Working with GitHub And Upload a Website template through Jenkins

## 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)**

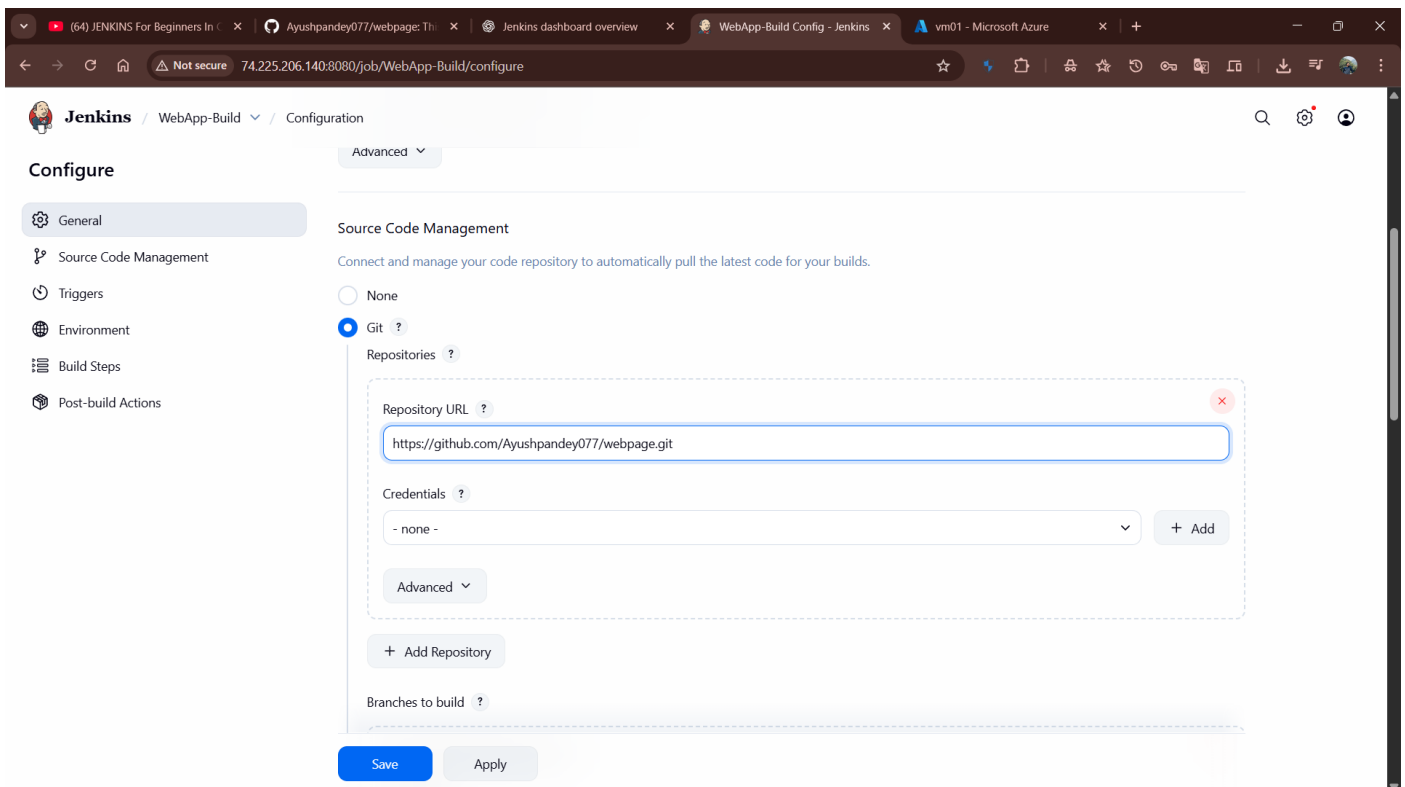
Run these commands in your Ubuntu terminal (not in Jenkins):

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.

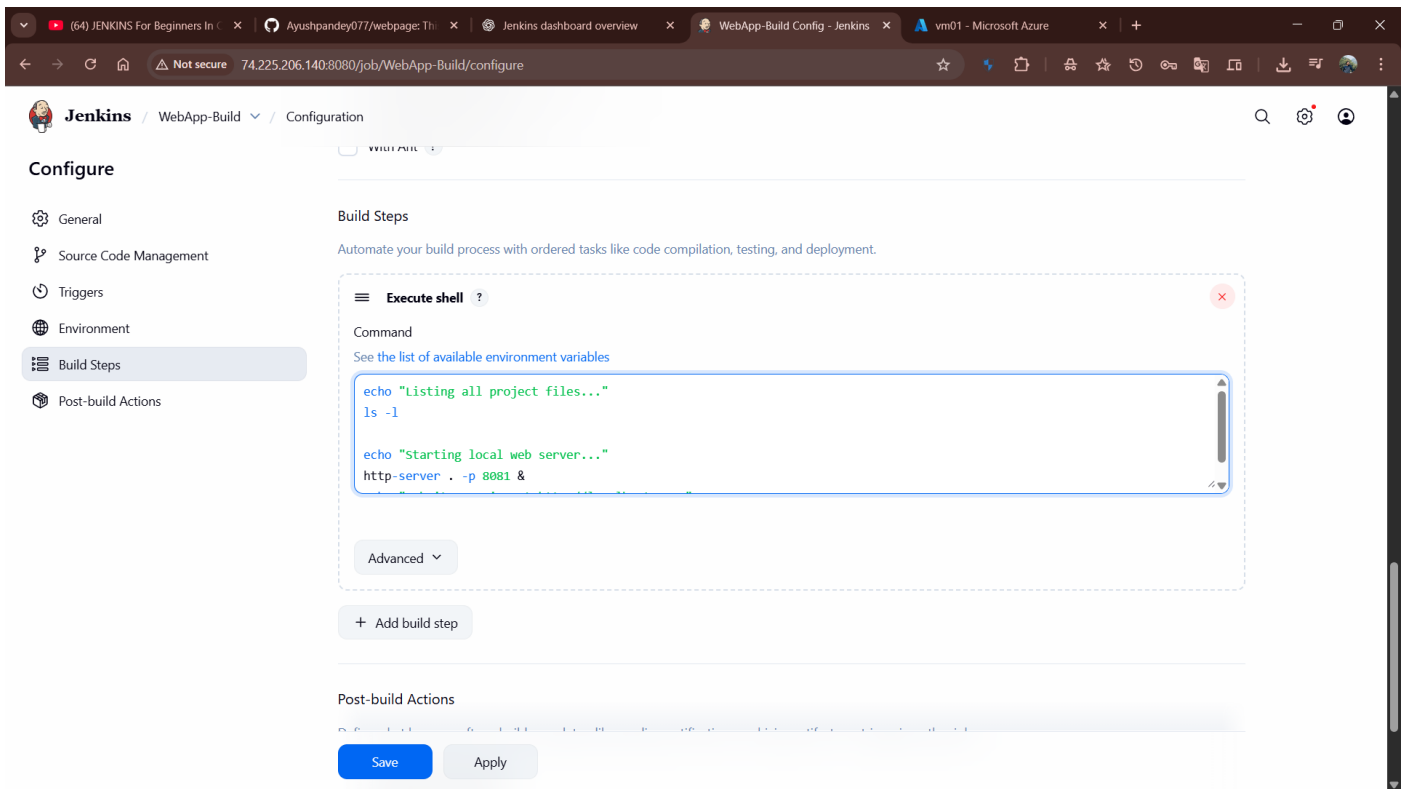


- Build Section

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.

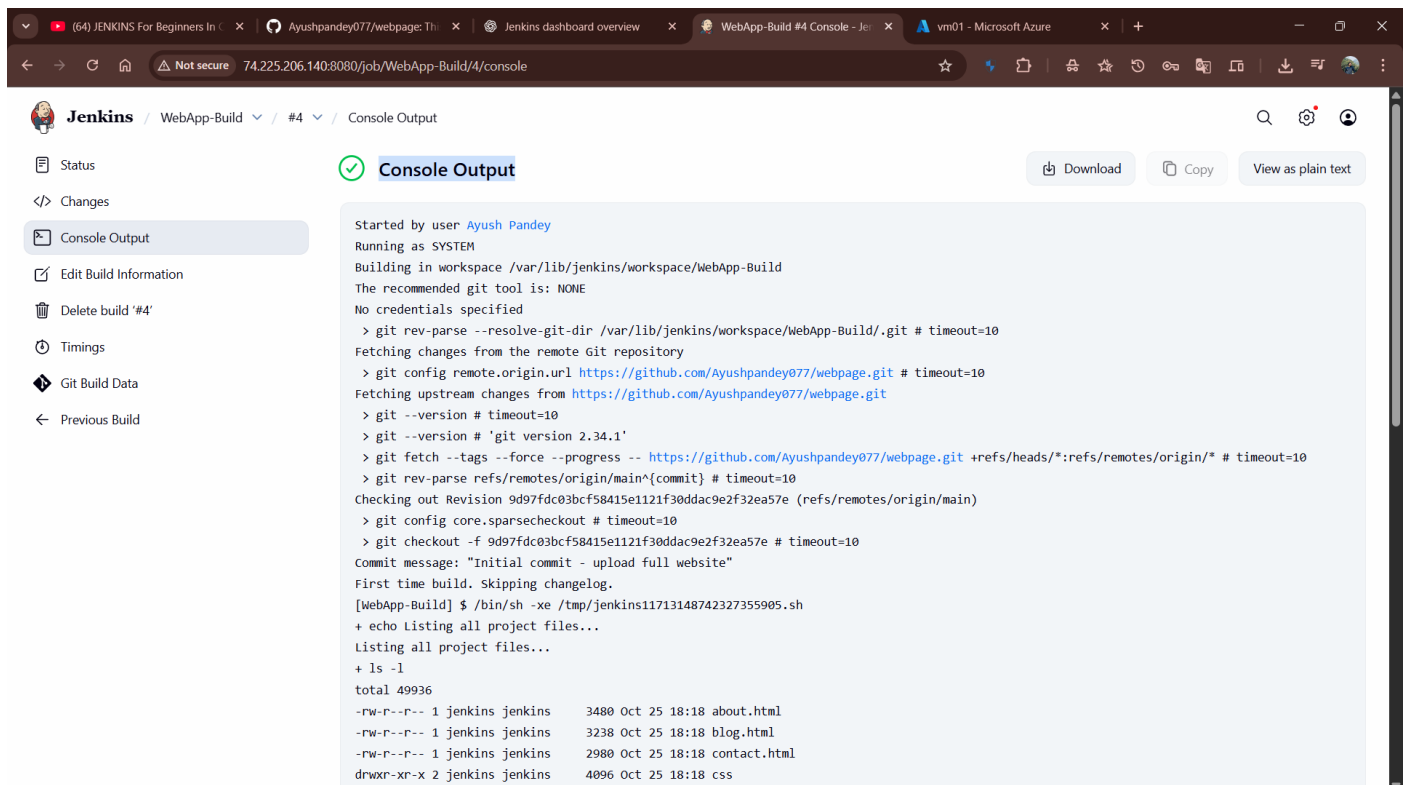
To keep it running after the job ends, you can run it like this:

- ❖ `nohup http-server. -p 8081 &`

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

## Result: -

**Console Output:- Success**



The screenshot shows the Jenkins web interface for a build named 'WebApp-Build #4'. The 'Console Output' tab is selected, displaying the following text:

```
Started by user Ayush Pandey
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/WebApp-Build
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/WebApp-Build/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Ayushpandey077/webpage.git # timeout=10
Fetching upstream changes from https://github.com/Ayushpandey077/webpage.git
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git fetch --tags --force --progress -- https://github.com/Ayushpandey077/webpage.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 9d97fdc03bcf58415e1121f30ddac9e2f32ea57e (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 9d97fdc03bcf58415e1121f30ddac9e2f32ea57e # timeout=10
Commit message: "Initial commit - upload full website"
First time build. Skipping changelog.
[WebApp-Build] $ /bin/sh -xe /tmp/jenkins11713148742327355905.sh
+ echo Listing all project files...
Listing all project files...
+ ls -l
total 49936
-rw-r--r-- 1 jenkins jenkins 3480 Oct 25 18:18 about.html
-rw-r--r-- 1 jenkins jenkins 3238 Oct 25 18:18 blog.html
-rw-r--r-- 1 jenkins jenkins 2980 Oct 25 18:18 contact.html
drwxr-xr-x 2 jenkins jenkins 4096 Oct 25 18:18 css
```









WebApp-Build #4 Console - Jenkins

```
-rw-r--r-- 1 jenkins jenkins    2982 Oct 25 18:18 singlepost.html
+ echo Starting local web server...
Starting local web server...
+ echo Website running at http://localhost:8081
Website running at http://localhost:8081
+ http-server . -p 8081
events.js:291
    throw er; // Unhandled 'error' event
    ^

Error: listen EADDRINUSE: address already in use 0.0.0.0:8081
    at Server.setupListenHandle [as _listen2] (net.js:1310:16)
    at listenInCluster (net.js:1358:12)
    at doListen (net.js:1495:7)
    at processTicksAndRejections (internal/process/task_queues.js:85:21)
Emitted 'error' event on Server instance at:
    at emitErrorNT (net.js:1337:8)
    at processTicksAndRejections (internal/process/task_queues.js:84:21) {
  code: 'EADDRINUSE',
  errno: 'EADDRINUSE',
  syscall: 'listen',
  address: '0.0.0.0',
  port: 8081
}
Finished: SUCCESS
```

REST API Jenkins 2.528.1

Index of /

	(drwx-----) 25-Oct-2025 17:32	<a href="#">.cache/</a>
	(drwx-----) 25-Oct-2025 17:29	<a href="#">.ssh/</a>
	(-rw-r--r--) 06-Jan-2022 16:23 220B	<a href="#">.bash_logout</a>
	(-rw-r--r--) 06-Jan-2022 16:23 3.7k	<a href="#">.bashrc</a>
	(-rw-r--r--) 06-Jan-2022 16:23 807B	<a href="#">.profile</a>
	(-rw-r--r--) 25-Oct-2025 17:32 0B	<a href="#">.sudo_as_admin_successful</a>
	(-rw-r--r--) 25-Oct-2025 17:32 50B	<a href="#">.Xauthority</a>
	(-rw-r--r--) 25-Oct-2025 18:23 1.4k	<a href="#">nohup.out</a>

Node.js v12.22.9/ [http-server](#) server running @ 74.225.206.140:8081

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