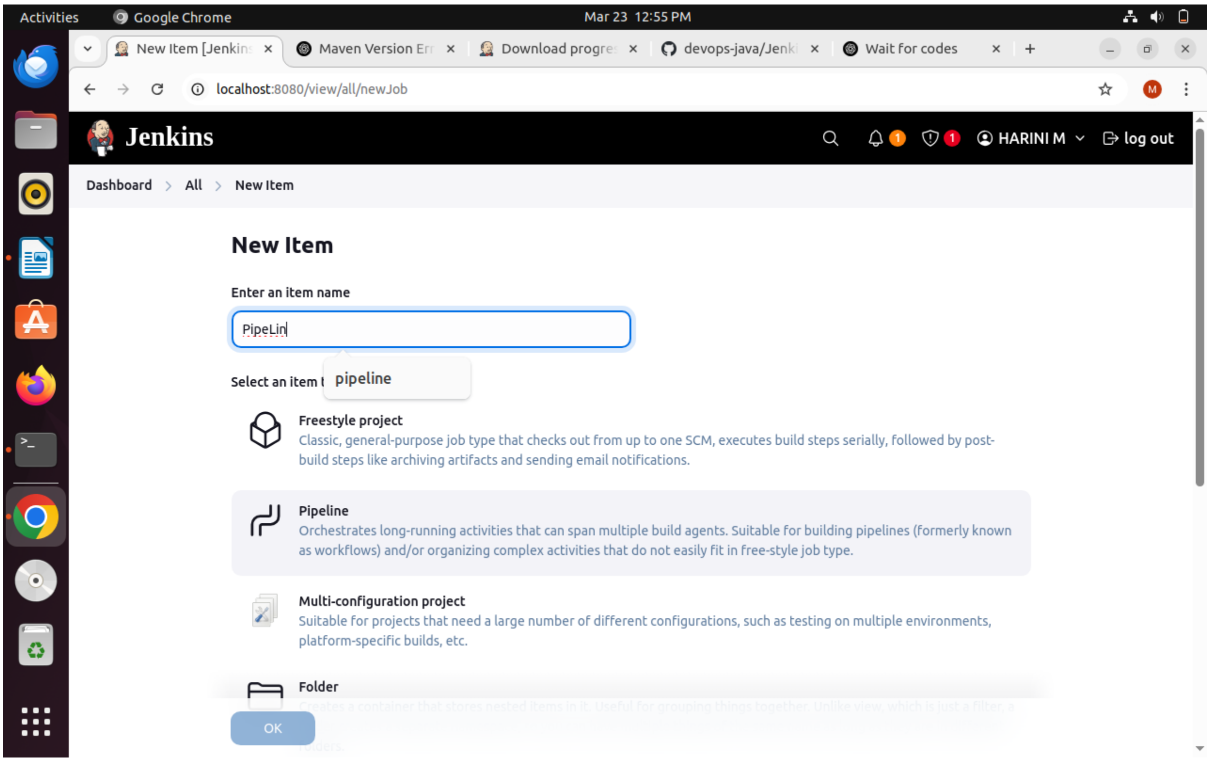
TASK -5: PIPELINE CREATION USING MAVEN AND DOCKER

Step 1: Create a New Item (Pipeline):

- **Action:**
  - Go to your Jenkins dashboard and click on **New Item**.
  - Name the item **Pipeline** and select the **Pipeline** type.
  - Click **OK** to proceed.



New Item

Enter an item name

PipeLin

Select an item t pipeline

Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

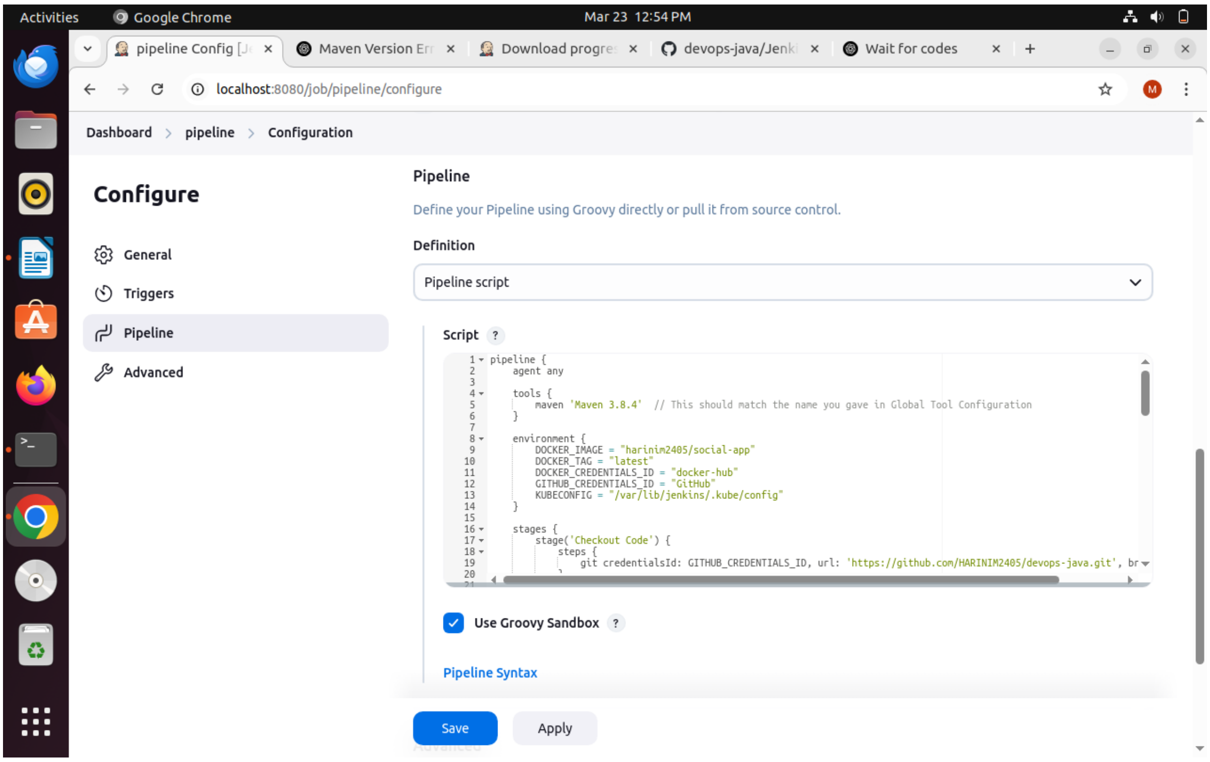
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder...

Step 2: Configure Pipeline Script:

- **Action:**
  - In the configuration section, input your desired **Pipeline script**.
  - After entering the script, click **Apply** to apply the changes.
  - Then, click **Save** to finalize the configuration



Pipeline

Define your Pipeline using Groovy directly or pull it from source control.

Definition

Pipeline script

Script

```
pipeline {
    agent any

    tools {
        maven 'Maven 3.8.4'  // This should match the name you gave in Global Tool Configuration
    }

    environment {
        DOCKER_IMAGE = "harinim2405/social-app"
        DOCKER_TAG = "latest"
        DOCKER_CREDENTIALS_ID = "docker-hub"
        GITHUB_CREDENTIALS_ID = "GitHub"
        KUBECONFIG = "/var/lib/jenkins/.kube/config"
    }

    stages {
        stage('Checkout Code') {
            steps {
                git credentialsId: GITHUB_CREDENTIALS_ID, url: 'https://github.com/HARINIM2405/devops-java.git', br
```

☑ Use Groovy Sandbox

Pipeline Syntax

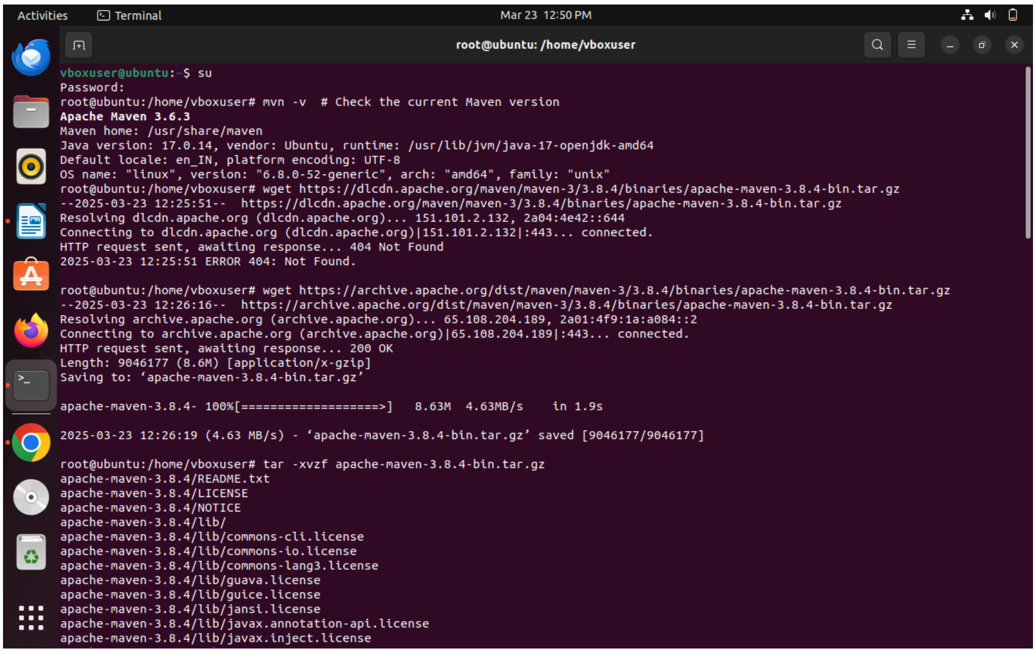
Save Apply

Step 3: Maven Installation and Configuration

- mvn -v
- wget https://archive.apache.org/dist/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz
- tar -xvzf apache-maven-3.8.4-bin.tar.gz
- sudo mv apache-maven-3.8.4 /opt/
- nano ~/.bashrc
- export M2_HOME=/opt/apache-maven-3.8.4
- export MAVEN_HOME=$M2_HOME
- export PATH=$M2_HOME/bin:$PATH
- source ~/.bashrc
- sudo systemctl restart Jenkins

1. mvn -v → Checks the current Maven version installed on the system.
2. wget ... → Downloads Maven 3.8.4 from the Apache archive.
3. tar -xvzf ... → Extracts the downloaded Maven archive.
4. sudo mv ... → Moves the extracted Maven folder to /opt/ for system-wide use.
5. nano ~/.bashrc → Opens the .bashrc file to update environment variables.
6. export ... → Sets Maven's home directory and updates the system path.

7. source ~/.bashrc → Applies the changes made to the environment variables.



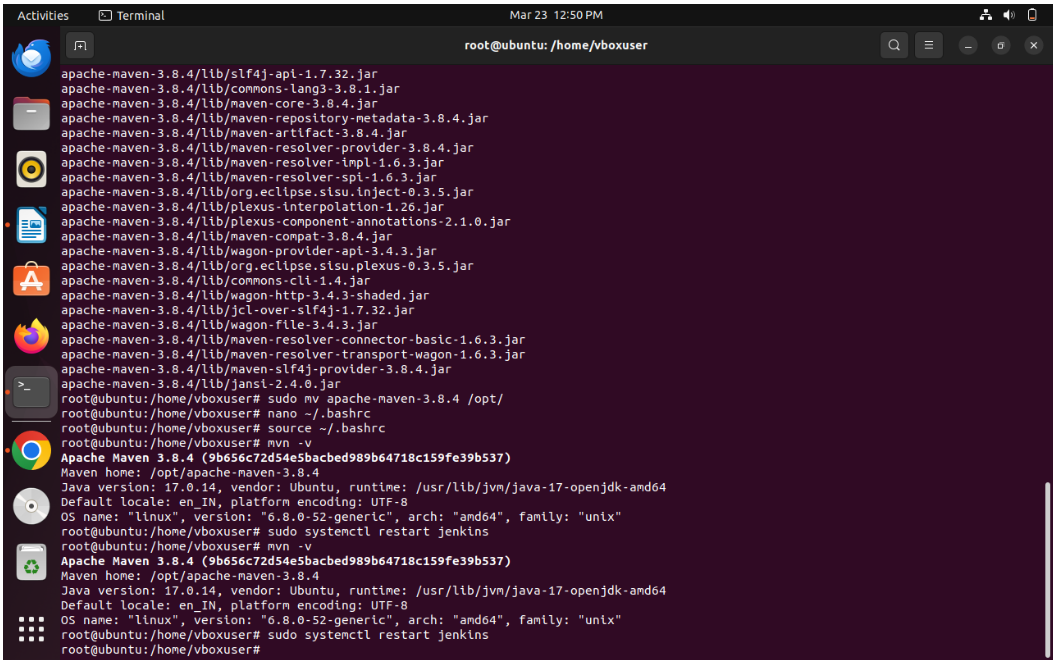
```
vboxuser@ubuntu:~$ su
Password:
root@ubuntu:/home/vboxuser# mvn -v  # Check the current Maven version
Apache Maven 3.6.3
Maven home: /usr/share/maven
Java version: 17.0.14, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-52-generic", arch: "amd64", family: "unix"
root@ubuntu:/home/vboxuser# wget https://dlcdn.apache.org/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz
--2025-03-23 12:25:51--  https://dlcdn.apache.org/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 404 Not Found
2025-03-23 12:25:51 ERROR 404: Not Found.

root@ubuntu:/home/vboxuser# wget https://archive.apache.org/dist/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz
--2025-03-23 12:26:16--  https://archive.apache.org/dist/maven/maven-3/3.8.4/binaries/apache-maven-3.8.4-bin.tar.gz
Resolving archive.apache.org (archive.apache.org)... 65.108.204.189, 2a01:4f9:1a:a084::2
Connecting to archive.apache.org (archive.apache.org)|65.108.204.189|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 9046177 (8.6M) [application/x-gzip]
Saving to: ‘apache-maven-3.8.4-bin.tar.gz’

apache-maven-3.8.4- 100%[===================>]   8.63M  4.63MB/s    in 1.9s

2025-03-23 12:26:19 (4.63 MB/s) - ‘apache-maven-3.8.4-bin.tar.gz’ saved [9046177/9046177]

root@ubuntu:/home/vboxuser# tar -xvzf apache-maven-3.8.4-bin.tar.gz
apache-maven-3.8.4/README.txt
apache-maven-3.8.4/LICENSE
apache-maven-3.8.4/NOTICE
apache-maven-3.8.4/lib/
apache-maven-3.8.4/lib/commons-cli.license
apache-maven-3.8.4/lib/commons-io.license
apache-maven-3.8.4/lib/commons-lang3.license
apache-maven-3.8.4/lib/guava.license
apache-maven-3.8.4/lib/guice.license
apache-maven-3.8.4/lib/jansi.license
apache-maven-3.8.4/lib/javax.annotation-api.license
apache-maven-3.8.4/lib/javax.inject.license
```



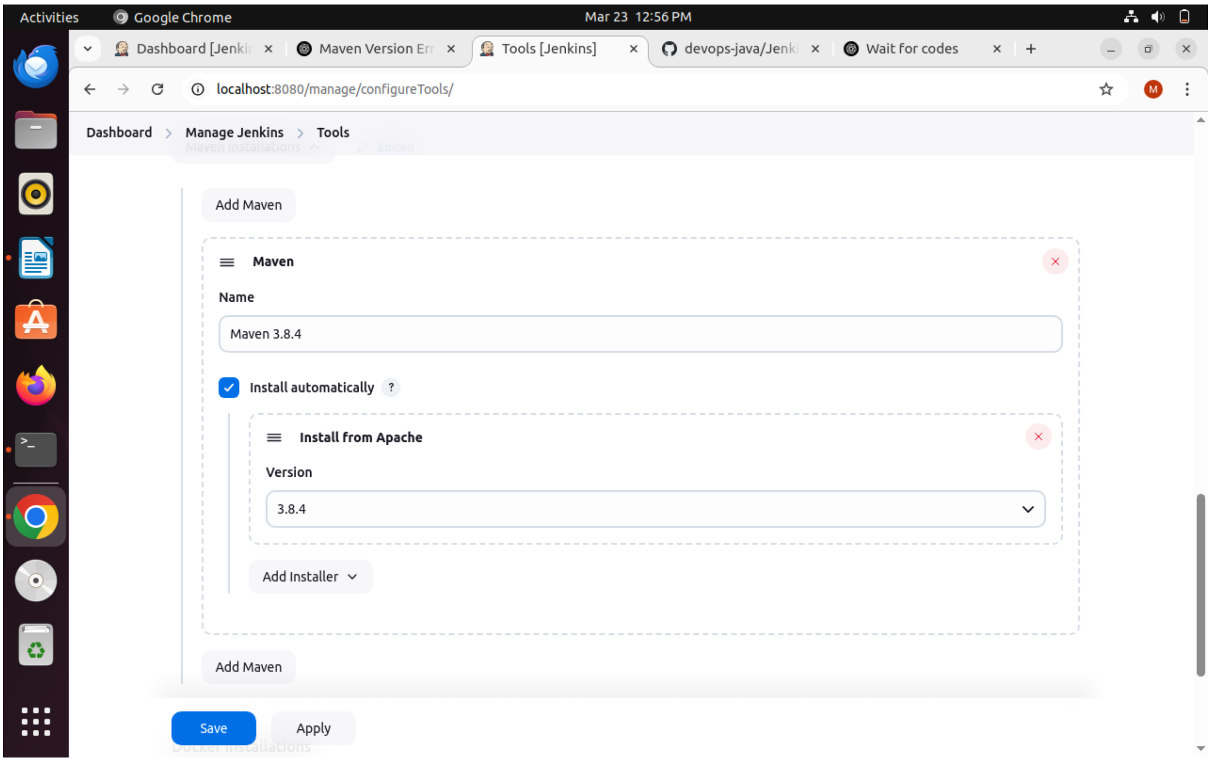
```
apache-maven-3.8.4/lib/slf4j-api-1.7.32.jar
apache-maven-3.8.4/lib/commons-lang3-3.8.1.jar
apache-maven-3.8.4/lib/maven-core-3.8.4.jar
apache-maven-3.8.4/lib/maven-repository-metadata-3.8.4.jar
apache-maven-3.8.4/lib/maven-artifact-3.8.4.jar
apache-maven-3.8.4/lib/maven-resolver-provider-3.8.4.jar
apache-maven-3.8.4/lib/maven-resolver-impl-1.6.3.jar
apache-maven-3.8.4/lib/maven-resolver-spi-1.6.3.jar
apache-maven-3.8.4/lib/org.eclipse.sisu.inject-0.3.5.jar
apache-maven-3.8.4/lib/plexus-interpolation-1.26.jar
apache-maven-3.8.4/lib/plexus-component-annotations-2.1.0.jar
apache-maven-3.8.4/lib/maven-compat-3.8.4.jar
apache-maven-3.8.4/lib/wagon-provider-api-3.4.3.jar
apache-maven-3.8.4/lib/org.eclipse.sisu.plexus-0.3.5.jar
apache-maven-3.8.4/lib/commons-cli-1.4.jar
apache-maven-3.8.4/lib/wagon-http-3.4.3-shaded.jar
apache-maven-3.8.4/lib/jcl-over-slf4j-1.7.32.jar
apache-maven-3.8.4/lib/wagon-file-3.4.3.jar
apache-maven-3.8.4/lib/maven-resolver-connector-basic-1.6.3.jar
apache-maven-3.8.4/lib/maven-resolver-transport-wagon-1.6.3.jar
apache-maven-3.8.4/lib/maven-slf4j-provider-3.8.4.jar
apache-maven-3.8.4/lib/jansi-2.4.0.jar
root@ubuntu:/home/vboxuser# sudo mv apache-maven-3.8.4 /opt/
root@ubuntu:/home/vboxuser# nano ~/.bashrc
root@ubuntu:/home/vboxuser# source ~/.bashrc
root@ubuntu:/home/vboxuser# mvn -v
Apache Maven 3.8.4 (9b656c72d54e5bacbed989b64718c159fe39b537)
Maven home: /opt/apache-maven-3.8.4
Java version: 17.0.14, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-52-generic", arch: "amd64", family: "unix"
root@ubuntu:/home/vboxuser# sudo systemctl restart jenkins
root@ubuntu:/home/vboxuser# mvn -v
Apache Maven 3.8.4 (9b656c72d54e5bacbed989b64718c159fe39b537)
Maven home: /opt/apache-maven-3.8.4
Java version: 17.0.14, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en_IN, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-52-generic", arch: "amd64", family: "unix"
root@ubuntu:/home/vboxuser# sudo systemctl restart jenkins
root@ubuntu:/home/vboxuser#
```

Why Maven?:

Maven is a build automation tool primarily for Java applications. It manages dependencies, builds projects, and handles packaging and deployment. It ensures your project builds are consistent and repeatable.

Step 4 : Add Maven in the Manage Plugins Section:

- Go to the **Manage Plugins** section under **Tools**.
- Add **Maven** to the list of available tools.



Maven

Name: Maven 3.8.4

☑ Install automatically

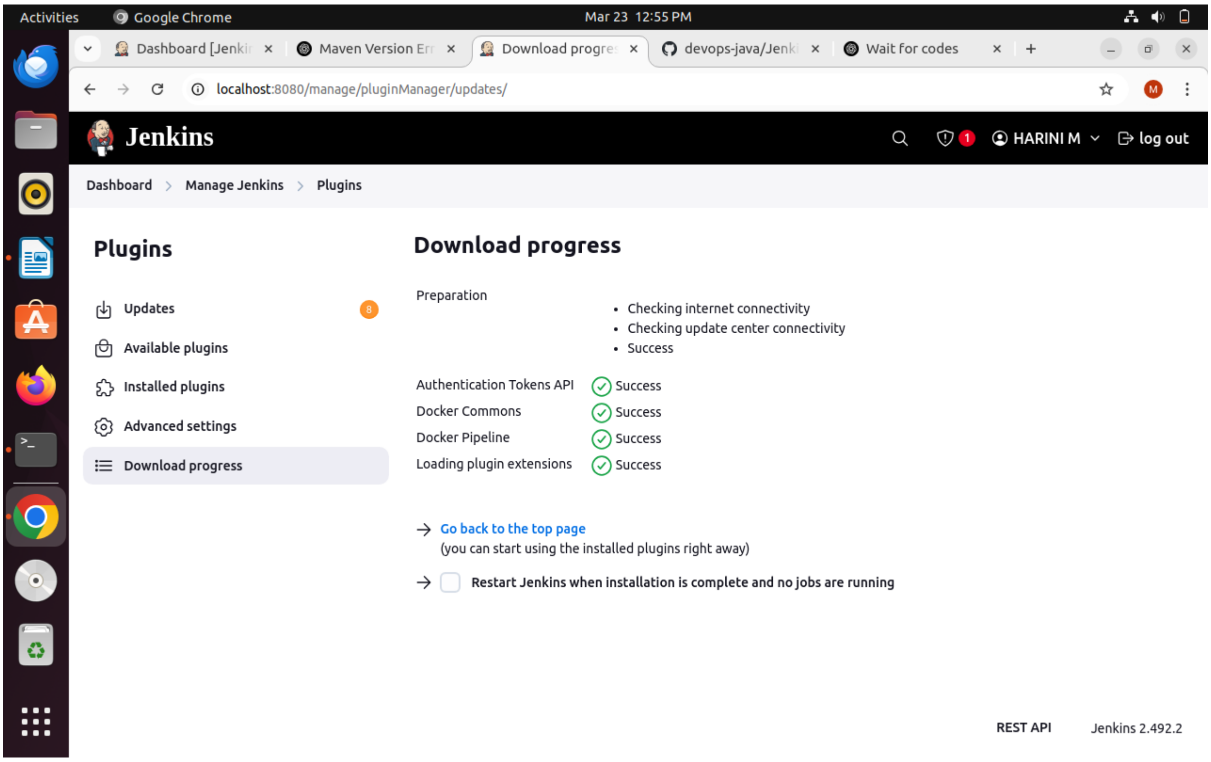
Install from Apache

Version: 3.8.4

Step 5 : Install Docker Pipeline Plugin (if required):

- **Action:**
  - Go to **Manage Jenkins** > **Manage Plugins**.
  - In the **Available** tab, search for **Docker Pipeline** and install the plugin if it is not already installed.

Why Docker Pipeline Plugin is needed:
The Docker Pipeline plugin allows Jenkins to interact with Docker containers as part of the pipeline process. This is essential for running jobs inside containers or working with Docker images in CI/CD workflows



Plugins

- Updates 8
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

Download progress

Preparation
- Checking internet connectivity
- Checking update center connectivity
- Success

Authentication Tokens API	✓ Success
Docker Commons	✓ Success
Docker Pipeline	✓ Success
Loading plugin extensions	✓ Success

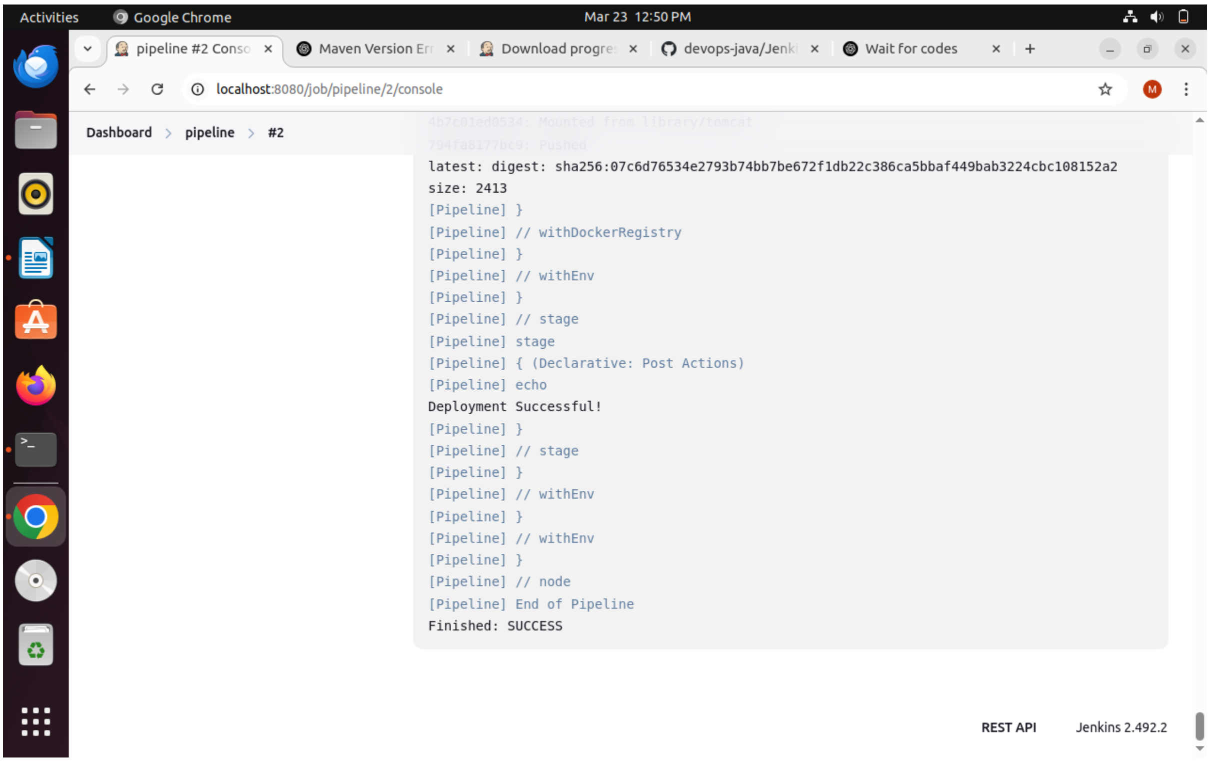
→ Go back to the top page
(you can start using the installed plugins right away)

→ □ Restart Jenkins when installation is complete and no jobs are running

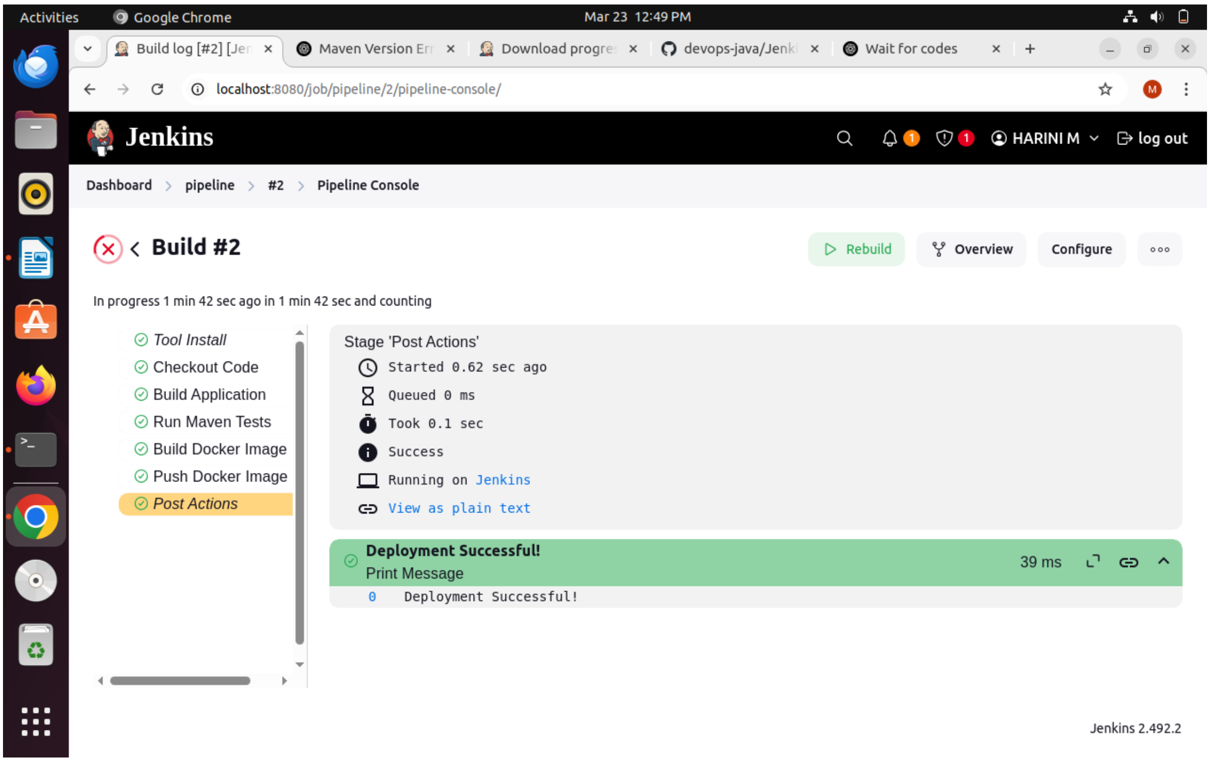
REST API Jenkins 2.492.2

Step 6 : Build the Pipeline:

- **Action:**
  - After configuration and installation of necessary tools, you can build the pipeline by selecting **Build Now**.
- **Explanation:**
  The build will execute the pipeline script, utilize Maven for the build process, and interact with Docker (if necessary).



```
4b7c01ed0534: Mounted from library/tomcat
794fa8177bc9: Pushed
latest: digest: sha256:07c6d76534e2793b74bb7be672f1db22c386ca5bbaf449bab3224cbc108152a2
size: 2413
[Pipeline] }
[Pipeline] // withDockerRegistry
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Declarative: Post Actions)
[Pipeline] echo
Deployment Successful!
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```



Build #2

In progress 1 min 42 sec ago in 1 min 42 sec and counting

- ✓ Tool Install
- ✓ Checkout Code
- ✓ Build Application
- ✓ Run Maven Tests
- ✓ Build Docker Image
- ✓ Push Docker Image
- ✓ Post Actions

Stage 'Post Actions'

```
Started 0.62 sec ago
Queued 0 ms
Took 0.1 sec
Success
Running on Jenkins
View as plain text
```

Deployment Successful!
Print Message 39 ms

```
0   Deployment Successful!
```