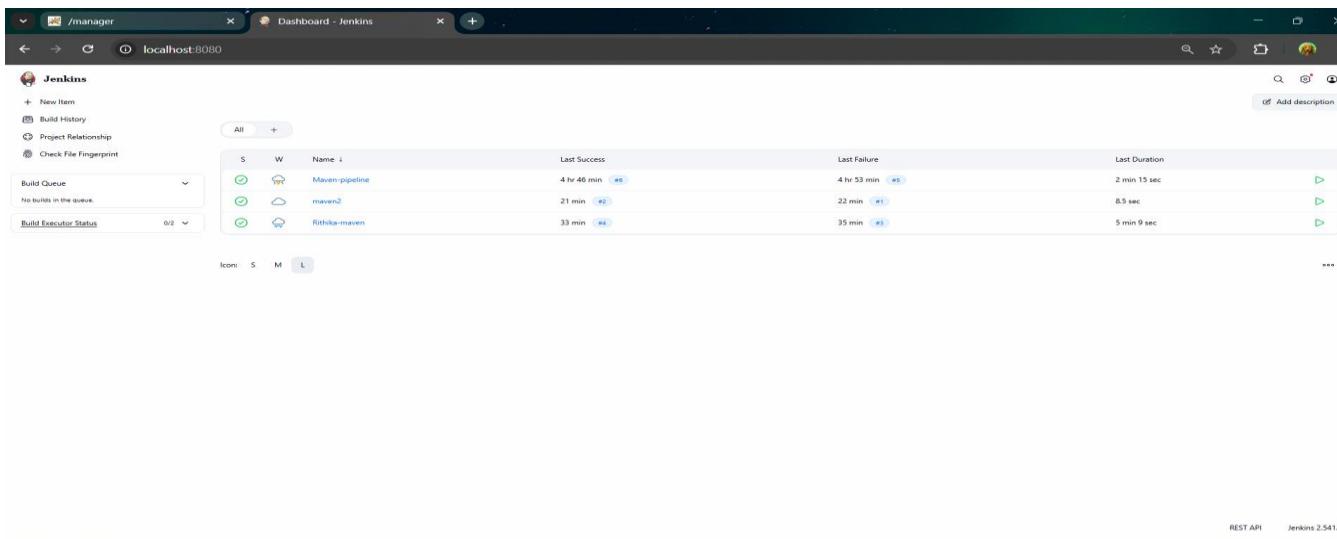


JENKINS PIPELINE

1) Open the Jenkins



The screenshot shows the Jenkins dashboard at localhost:8080. The left sidebar has links for New Item, Build History, Project Relationship, and Check File Fingerprint. The main area displays a table of builds:

S	V	Name	Last Success	Last Failure	Last Duration
Green	Cloud	Maven-pipeline	4 hr 46 min	4 hr 53 min	2 min 15 sec
Green	Cloud	maven2	21 min	22 min	8.5 sec
Green	Cloud	Rithika-maven	33 min	35 min	5 min 9 sec

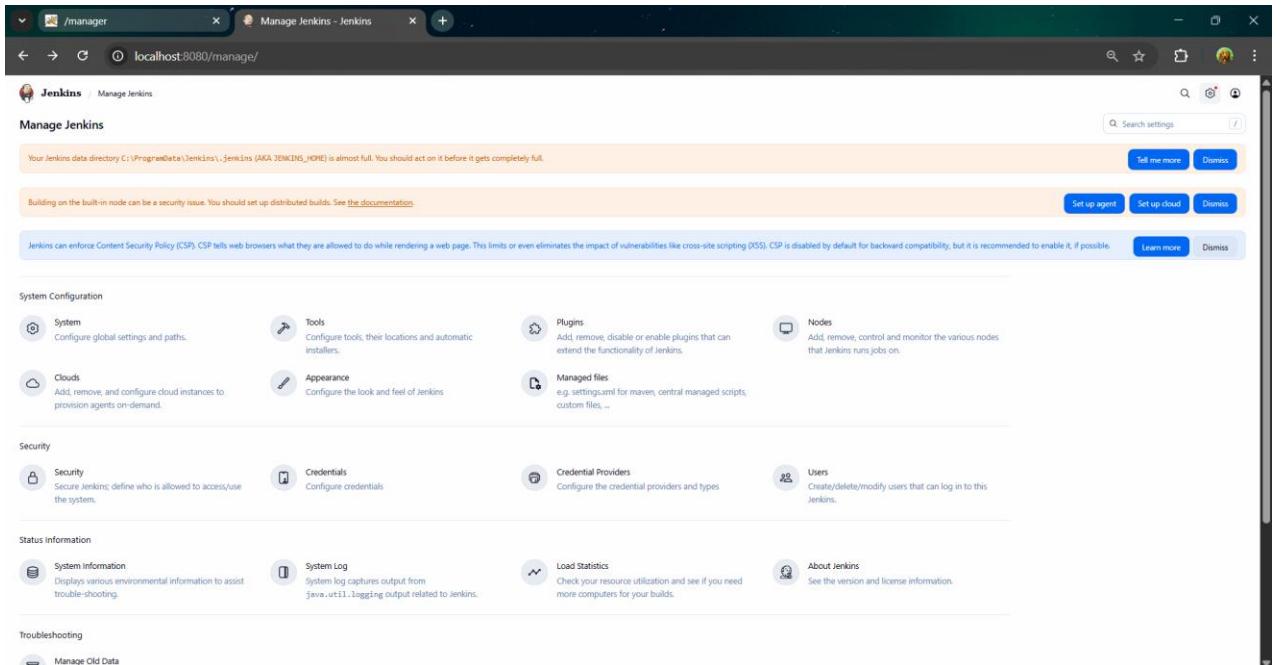
Build Queue: No builds in the queue.

Build Executor Status: 0/2

Icons: S M L

REST API Jenkins 2.541

2) Goto manage Jenkins:



The screenshot shows the Manage Jenkins page at localhost:8080/manage/. It includes several sections:

- System Configuration**: Includes System (Configure global settings and paths), Tools (Configure tools, their locations and automatic installers), Clouds (Add, remove, and configure cloud instances to provision agents on-demand), Appearance (Configure the look and feel of Jenkins), Plugins (Add, remove, disable or enable plugins that can extend the functionality of Jenkins), Managed files (e.g. settings.xml for maven, central managed scripts, custom files, ...), Nodes (Add, remove, control and monitor the various nodes that Jenkins runs jobs on).
- Security**: Includes Security (Secure Jenkins: define who is allowed to access/use the system), Credentials (Configure credentials), Credential Providers (Configure the credential providers and types), and Users (Create/delete/modify users that can log in to this Jenkins).
- Status Information**: Includes System information (Displays various environmental information to assist trouble-shooting), System Log (System log captures output from java.util.logging output related to Jenkins), Load Statistics (Check your resource utilization and see if you need more computers for your builds), and About Jenkins (See the version and license information).
- Troubleshooting**: Includes Manage Old Data.

3) Goto Plugins and search Deploy to container and install it :

The screenshot shows the Jenkins Plugins page. A search bar at the top contains the text "deploy". Below it, a table lists four plugins:

- Deploy to container 1.17** (Released 8 mo 19 days ago, 87 reviews): This plugin allows you to deploy a war to a container after a successful build. Glassfish 3.x remote deployment.
- Docker Pipeline 634.vedc7242b_eda_7** (Released 3 mo 16 days ago, 76 reviews): Build and use Docker containers from pipelines. This plugin is up for adoption! We are looking for new maintainers. Visit our [Adopt a Plugin](#) initiative for more information.
- Artifactory 4.0.8** (Released 1 yr 6 mo ago, 82 reviews): This plugin allows your build jobs to deploy artifacts and resolve dependencies to and from Artifactory, and then have them linked to the build job that created them. The plugin includes a vast collection of features, including a rich pipeline API library and release management for Maven and Gradle builds with Staging and Promotion.
- Ansible 635.v34b_48e979cd6** (Released 14 days ago, 100 reviews): pipeline External Site/Tool Integrations DevOps Build Tools Deployment

4) Now go to the system

The screenshot shows the Jenkins Manage Jenkins page. At the top, there is a message: "Building on the built-in node can be a security issue. You should set up distributed builds. See [the documentation](#)". Below this, there is a note about Content Security Policy (CSP): "Jenkins can enforce Content Security Policy (CSP). CSP tells web browsers what they are allowed to do while rendering a web page. This limits or even eliminates the impact of vulnerabilities like cross-site scripting (XSS). CSP is disabled by default for backward compatibility, but it is recommended to enable it, if possible." There are three buttons: "Set up agent", "Set up cloud", and "Dis".

The main area is titled "System Configuration" and contains several sections:

- System**: Configure global settings and paths.
- Tools**: Configure tools, their locations and auto installers.
- Plugins**: Add, remove, disable or enable plugins that can
- Nodes**: Add, remove, control and monitor the va

and choose Global credentials and add some credentials.

New credentials

Kind

Username with password

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

Username
Blank username; did you mean to use secret text credentials instead?
 Treat username as secret ?

Password

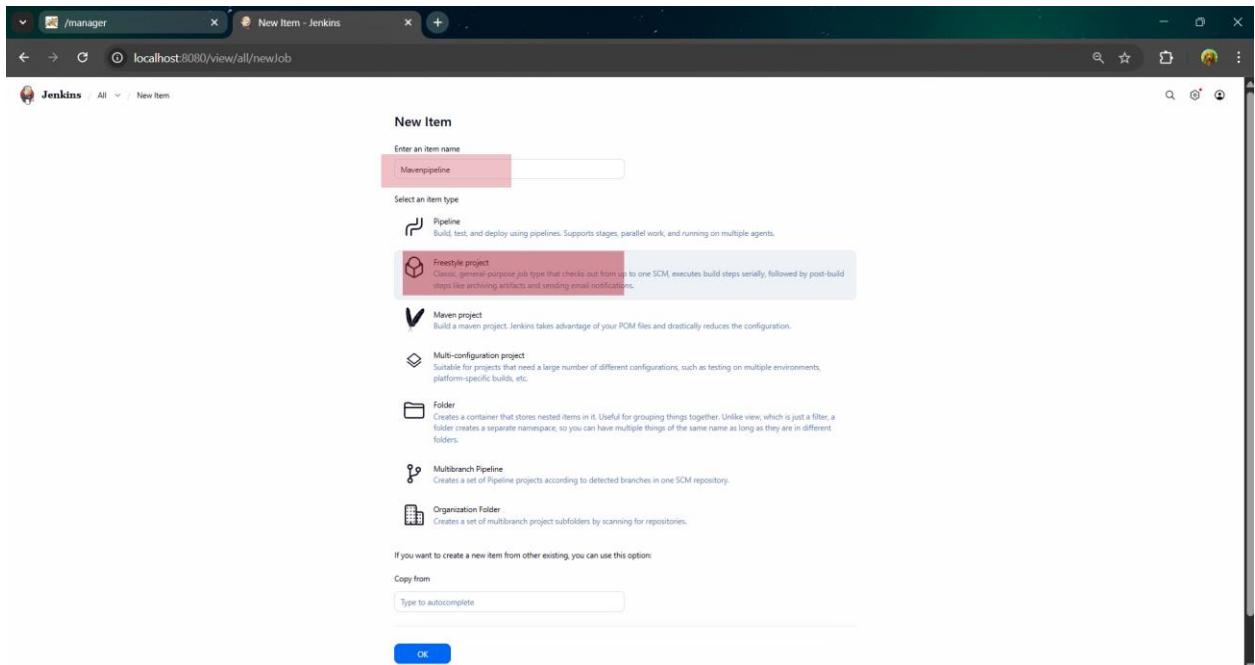
ID ?

Description ?

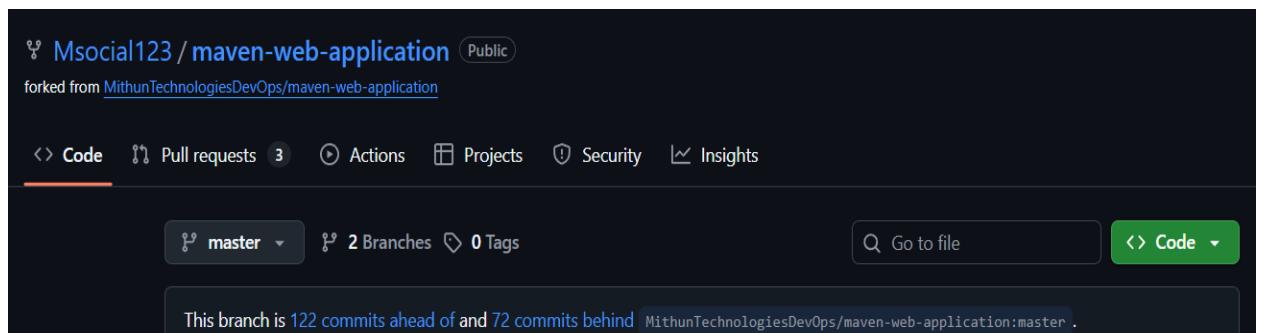
Create

Then will create a new credentials such as **username** , **password** , **ID** , **Description** and **Click create.**

- 5) Now goto the home page and click **New item** , enter the item name and choose the **Freestyle Project** and click OK.



- 6) Goto the **github** and choose the **Repository** and copy the code.



7) Now **open the Item** , goto configure

Jenkins / Maven-pipeline

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename

It's a maven

Permalink

and add description

Jenkins / Maven-pipeline / Configure

Configure

General

Description
It's a maven application

Plain text [Preview](#)

Discard old builds ?

GitHub project

This project is parameterized ?

Throttle builds ?

Execute concurrent builds if necessary ?

[Advanced](#) ▾

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

None

Git ?

Repositories ?

- 8) Goto the Source code management: paste the repo url and credential should be none

Jenkins / Maven-pipeline / Configure

Configure

Source Code Management

Connect and manage your code repository to automatically pull the latest code for your builds.

None

Git ?

Repositories ?

Repository URL ?
https://github.com/Msocial123/maven-web-application.git

Credential
- none -

[Advanced](#) ▾

+ Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?
*/master

+ Add Branch

Repository browser ?
(Auto)

Additional Behaviours

+ Add

And to check the branches to build should be as in the repository whether it is main or master in the left side. After that in build steps type **clean package** in **Goals**.

- 9) Goto the post build Actions and give war by clicking ? And choose the credentials and our tomcat url and click Apply and save.

The screenshot shows the Jenkins job configuration page for 'Maven-pipeline'. The 'Post-build Actions' section is selected. It contains a 'Deploy war/ear to a container' action. Under 'WAR/EAR files', the value '**/*.war' is highlighted. Under 'Containers', 'Tomcat 9.x Remote' is selected, with 'Credentials' set to 'Ritzym***** (Tomcat-credentials)' and 'Tomcat URL' set to 'http://localhost:8081'. A blue box highlights the 'Advanced' dropdown and the '+ Add post-build action' button. At the bottom are 'Save' and 'Apply' buttons.

- 10) Goto setting and click plugins ==> available plugins and search **Maven integration, config file provider, Pipeline Maven Plugin API, Pipeline Maven Integration**.

- 11) Goto **Tools** and scroll down to see maven installation.

The screenshot shows the Jenkins 'Tools' configuration page under 'Maven installations'. A new entry for 'Maven' is being added. The 'Name' field is set to 'Maven', and the 'Install automatically' checkbox is checked. The 'Version' is set to '3.9.12'. A blue box highlights the 'Install from Apache' link. At the bottom are 'Save' and 'Apply' buttons.

Here give the name as maven and default it shows the version .Click **apply and save**.

12) Here in configure , can see in the environment , it will show the maven version and have to give goals as **clean package** in the earlier when the starting process in Build steps.

The screenshot shows the Jenkins 'Configure' screen for a project named 'Maven-pipeline'. The 'Build Steps' section is highlighted with a blue box. Inside this box, under the 'Invoke top-level Maven targets' heading, there is a 'Maven Version' step with 'Maven' selected and a 'Goals' step with 'clean package' selected. A large blue arrow points from the 'Environment' section at the top left down to the 'Goals' step.

12)Now click Build run

The screenshot shows the Jenkins project page for 'Maven-pipeline'. The left sidebar has buttons for 'Status', 'Changes', 'Workspace', 'Build Now' (which is highlighted with a blue box), 'Configure', 'Delete Project', and 'Rename'. The main area shows a green checkmark icon and the text 'Maven-pipeline'. Below it says 'It's a maven application'. On the right, there is a 'Permalinks' section with a bulleted list of recent builds.

- Last build (#6), 6 hr 23 min ago
- Last stable build (#6), 6 hr 23 min ago
- Last successful build (#6), 6 hr 23 min ago
- Last failed build (#5), 6 hr 30 min ago
- Last unsuccessful build (#5), 6 hr 30 min ago
- Last completed build (#6), 6 hr 23 min ago

13) It will get success and can see in the tomcat apache.

The screenshot shows a web browser window with three tabs open:

- /manager
- Maven-pipeline Config - Jenkins
- Document 6.docx

The main content area is titled "Tomcat Web Applications". It includes a "Message:" input field with "OK" and a "Manager" button. Below is a table listing applications:

Applications		
Path	Version	Display Name
/	<i>None specified</i>	Welcome to Tomcat
/docs	<i>None specified</i>	Tomcat Documentation
/manager	<i>None specified</i>	Tomcat Manager Application
/maven-web-application	<i>None specified</i>	maven-web-application
/petclinic	<i>None specified</i>	