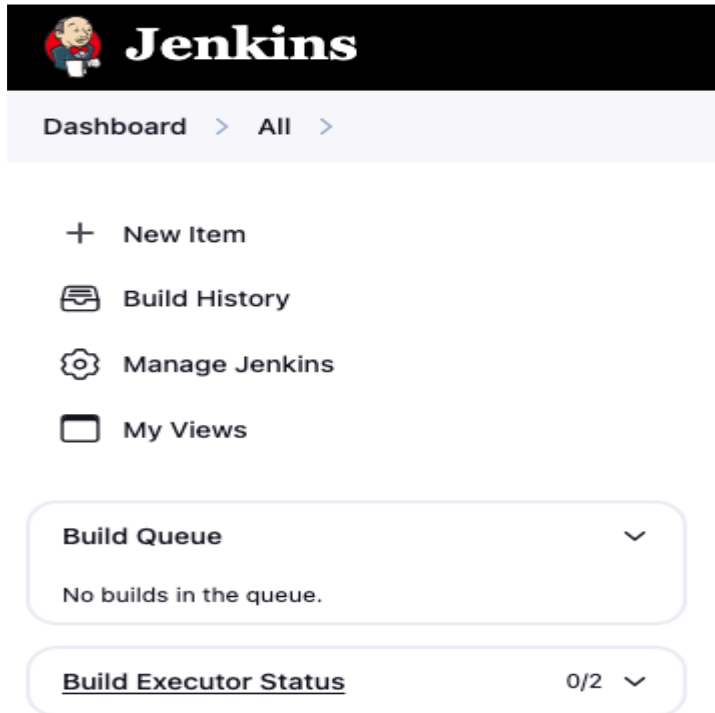


Jenkins freestyle project job running

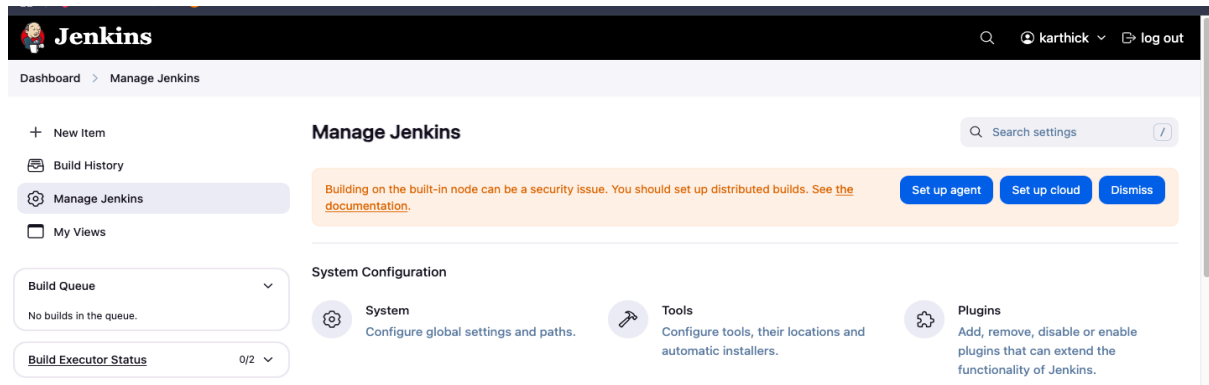
Step 1:

Click on manage jenkins



Step 2:

Click on tools section



Step 3:

Go to maven installation section

Set maven name and click check box install automatically maven

And chose maven version

Maven installations ^ [Edited](#)

Add Maven

Maven

Name

mymaven

☒ Install automatically ?

Install from Apache

Version

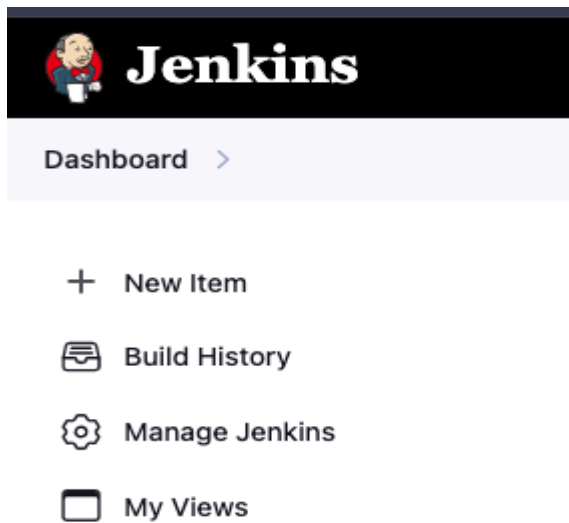
3.9.9

Add Installer ▾

Add Maven

Step 4:

Click on + new item



Step 5:

Enter job name and freestyle project and click ok button

New Item

Enter an item name

Select an item type

**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 creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK

Step 6:

Enter the description of job

General

Enabled



Description

kkkk

Plain text [Preview](#)

Step7:

Chose git options and give repository URL

repository URL: <https://github.com/KarthickSuresh09/jenkins-maven.git>

If repository is public means no need to provide credentials

You need to change **Branch Specifier master** to **main**

Git ?

Repositories ?

Repository URL ?

Credentials ?

- none -

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

Step 8:

In build step you need to choose **Invoke top-level Maven targets**

Now you need to choose which maven version you chose and maven name in maven check box

In goals you can give compile

Click on advanced option and you need to mention pom.xml file path

Build Steps

Automate your build process with ordered tasks like code compilation, testing, and deployment.



The screenshot shows a configuration window for a build step titled "Invoke top-level Maven targets". It includes a "Maven Version" dropdown set to "mymaven", a "Goals" dropdown set to "compile", and an "Advanced" section with a "POM" field set to "java-tomcat-sample/pom.xml". There is also a "Properties" field at the bottom.

Invoke top-level Maven targets ?

Maven Version

mymaven

Goals

compile

Advanced ^ Edited

POM ?

java-tomcat-sample/pom.xml

Properties ?

Step 9:

Apply and save the job

Click on build now button and your job will run



Status



Changes



Workspace



Build Now



Configure



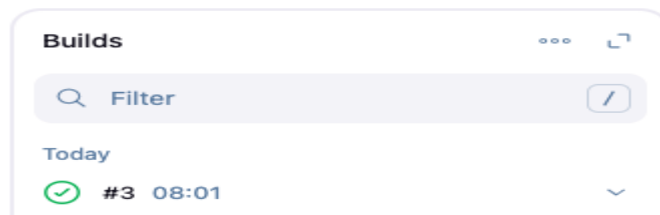
Delete Project



Rename

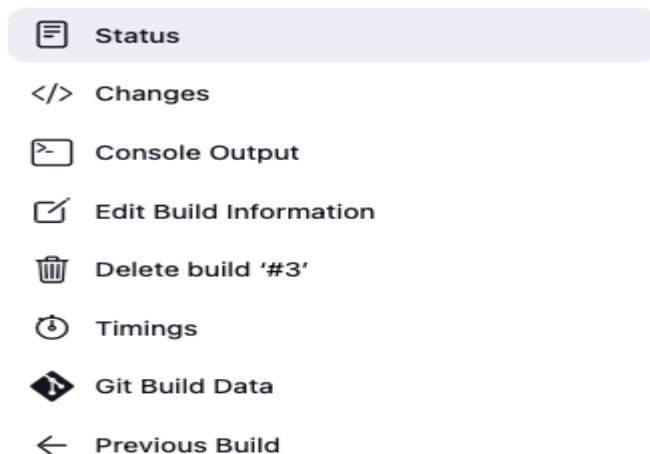
Step 10:

In builds section you get job is succeed or not



Step 11:

If you clicked job timings and you will enter details and click console output



Step 12 :

After clicking console output you will get the logs of the job

