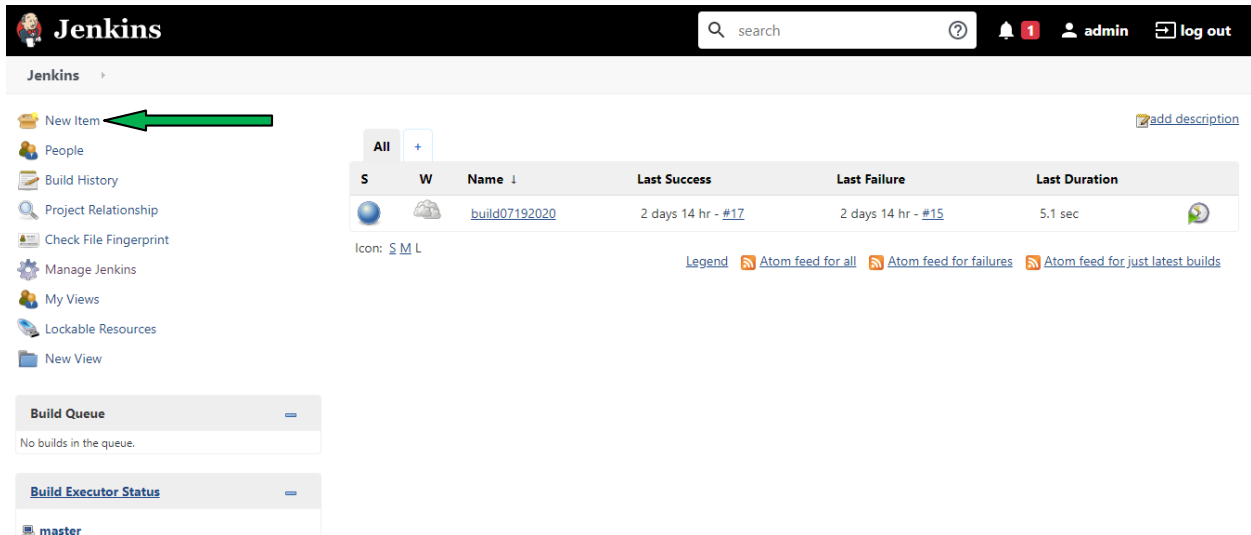


Jenkins Freestyle Job Step-by-Step Procedure

Configuring Freestyle Job in Jenkins:

Jenkins is an Automation / orchestration tool. Using Jenkins we can automate almost any type of task that we perform in SDLC as part of Development, QA, and Operation team's responsibilities.

- 1) TO start creating a Freestyle Job in Jenkins, click on the [New Item](#) link on the home page.



The screenshot shows the Jenkins home page. The left sidebar contains a list of links: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Lockable Resources, and New View. A green arrow points to the 'New Item' link. The main content area displays a table of jobs. The table has columns for Status (S), W, Name, Last Success, Last Failure, and Last Duration. A job named 'build07192020' is listed with a status of 'S' and a last success time of '2 days 14 hr - #17'. Below the table, there are links for 'Icon', 'Legend', and 'Atom feed'.


S	W	Name	Last Success	Last Failure	Last Duration
S		build07192020	2 days 14 hr - #17	2 days 14 hr - #15	5.1 sec


- 2) In the Job Template selection page as shown below, enter unique name (combination of char, number in small case) and select 'freestyle project' option and click on [OK](#) button.


Jenkins Freestyle Job Step-by-Step Procedure


Enter an item name

» Required field

**Freestyle project**
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**Maven project**
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**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.

- 3) On the job configuration page, we get different configurable items displayed in the tabs, '**General**', '**Source Code Management**', '**Build triggers**' etc. The first configuration that we need to update is in General configuration tab.
- 4) Select '**Discard Old Builds**' options for setting a policy of Log rotation. Here provide a value in number of days that we want to keep the logs for and number of builds. See below image for example.

Jenkins > build07222020 >

General | Source Code Management | Build Triggers | Build Environment | Build | Post-build Actions

[Plain text] [Preview](#)

Jira site

☒ Discard old builds

Strategy

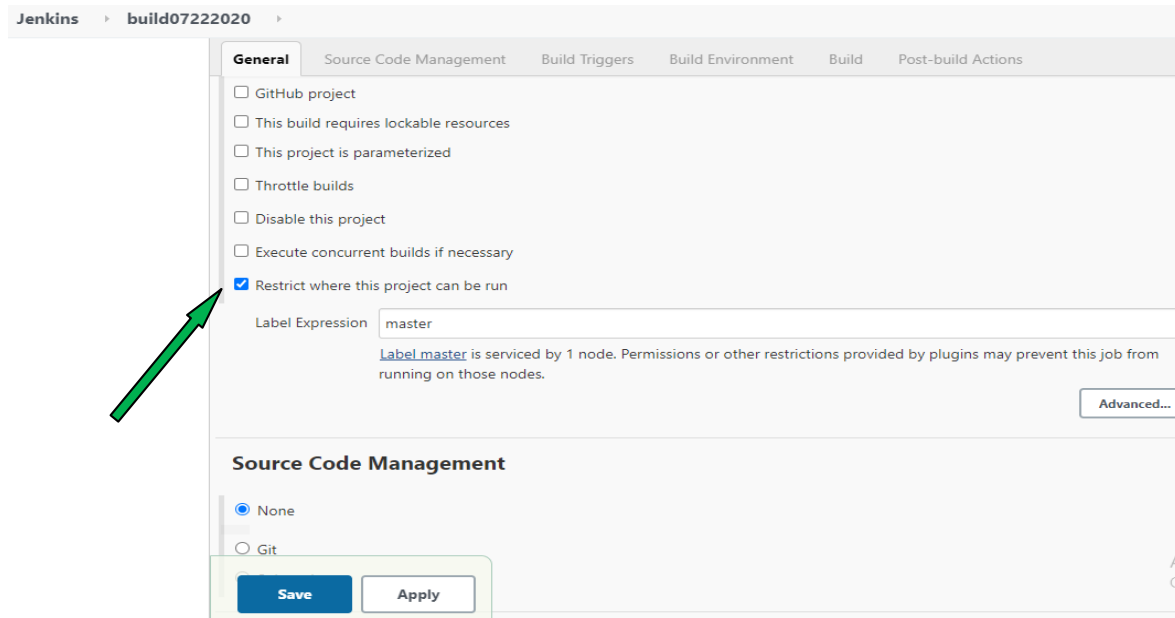
Days to keep builds
if not empty, build records are only kept up to this number of days

Max # of builds to keep
if not empty, only up to this number of build records are kept

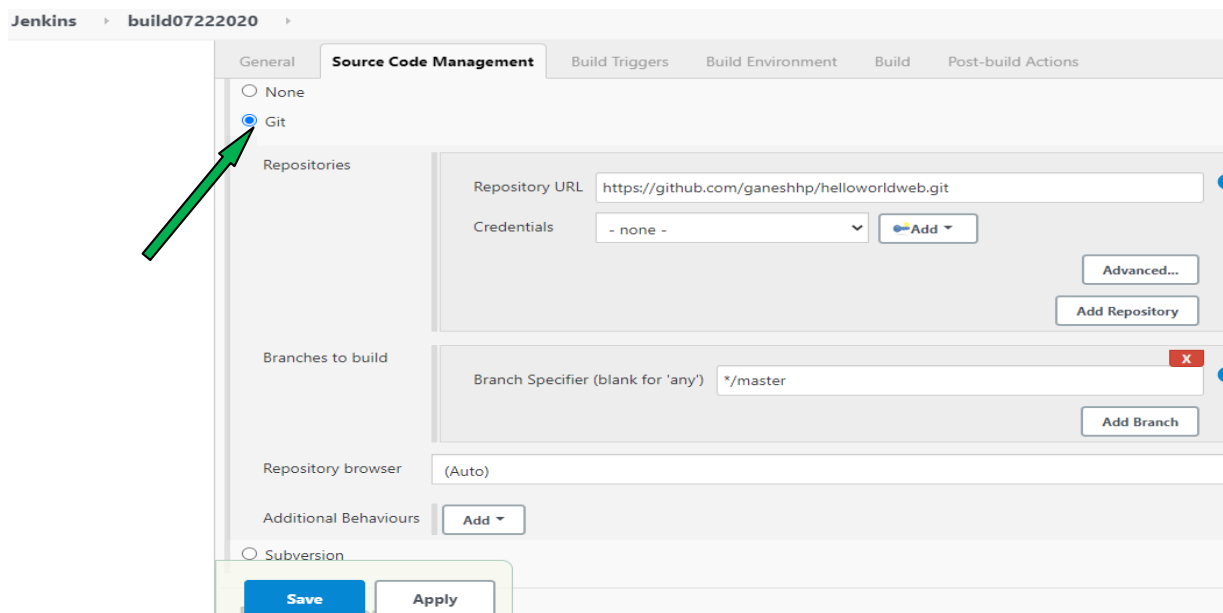
☐ GitHub project
☐ This build requires lockable resources
☐ This project is parameterized
☐ Throttle builds

Jenkins Freestyle Job Step-by-Step Procedure

- 5) In the general Tab, we then select the option '*Restrict where this project can be Run*' that will allow us to select the Node on which we want the Build Job to run. In below image, master node, i.e. Jenkins server has been selected. This will make the build to run on the Jenkins master server. The default location where the job execution will happen (Job Workspace) is </var/lib/Jenkins/workspace/<buildname>/>.

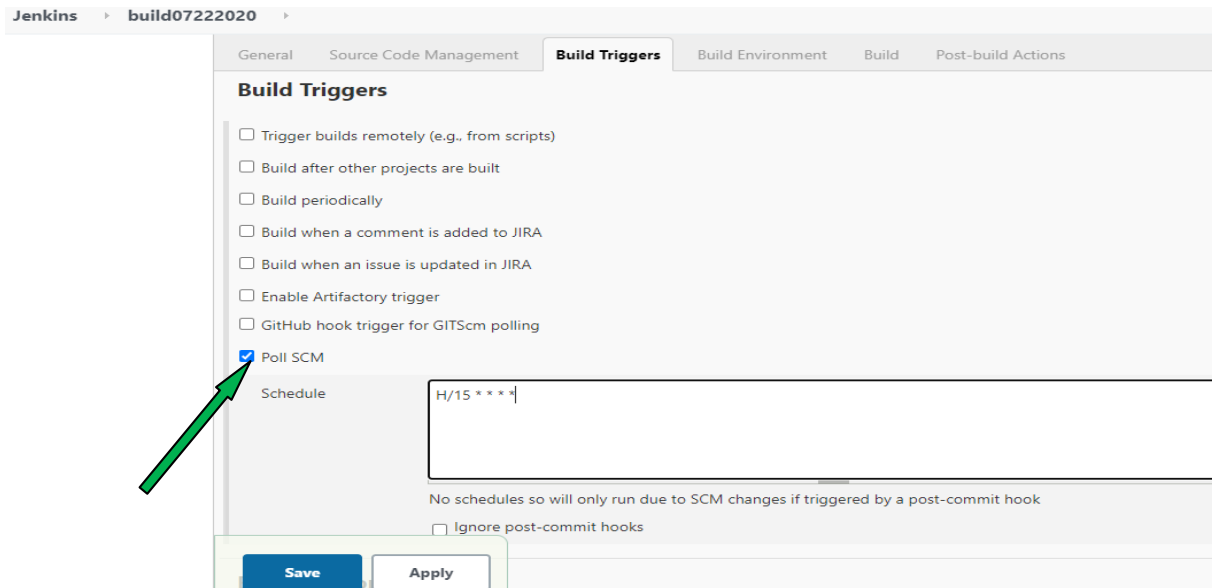


- 6) The next option in the menu tabs is *Source Code Management*. As we are using GitHub as our remote Source code management repository, we select '*git*' and provide github remote repository url.

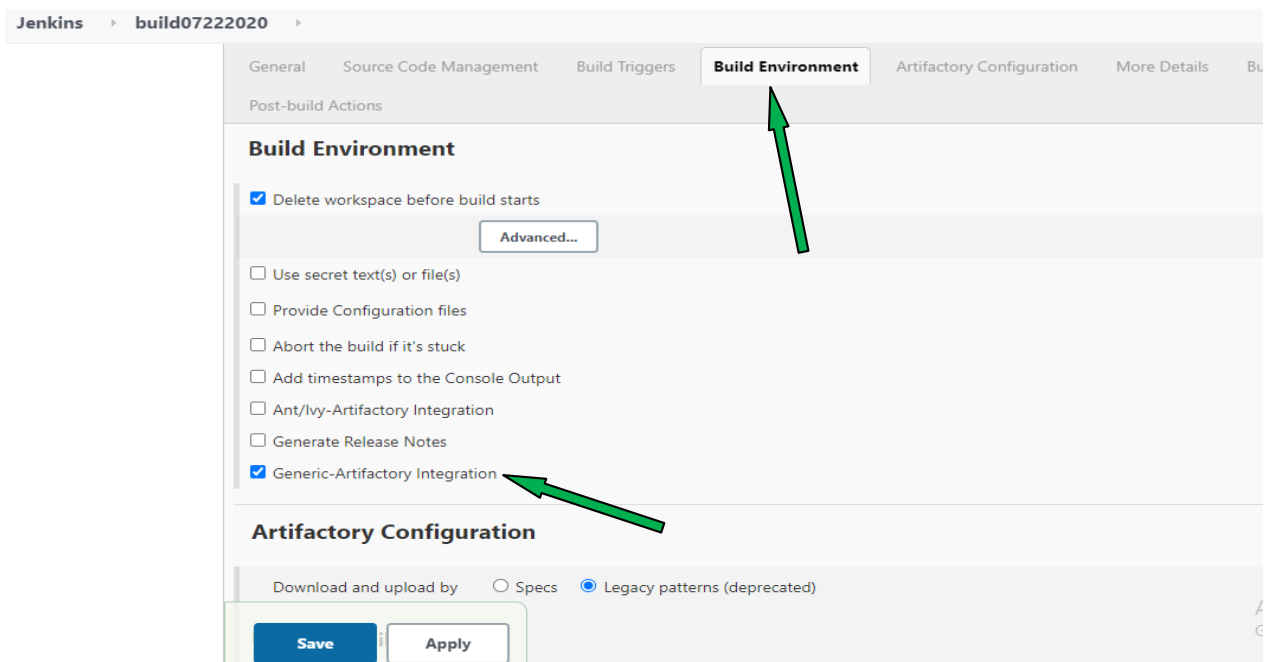


Jenkins Freestyle Job Step-by-Step Procedure

- 7) In the below image, the next option is to select how we want our Jenkins job to start, the *trigger*. If we want the job to start after a new version has been updated on the remote SCM repository, we have to select the '*Poll SCM*' option and provide the polling schedule in *CRON* syntax, '*H/5 * * * **'.

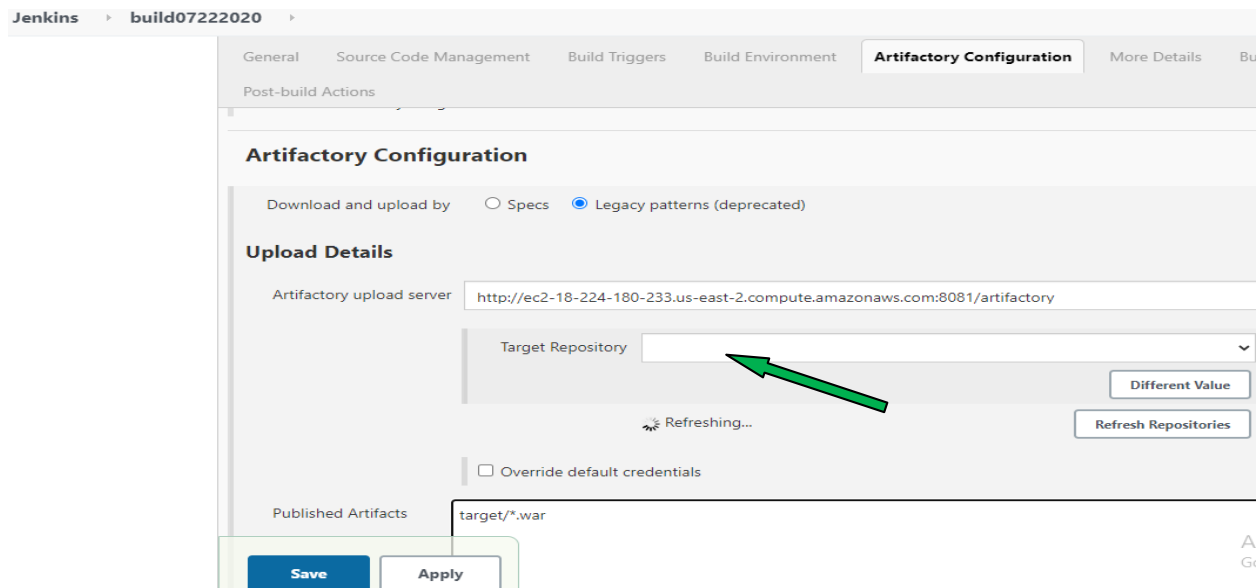


- 8) The next option in *Build environment* which allows to manage the build workspace. '*Delete workspace before build starts*' will help to get rid of any files that are created from earlier build.

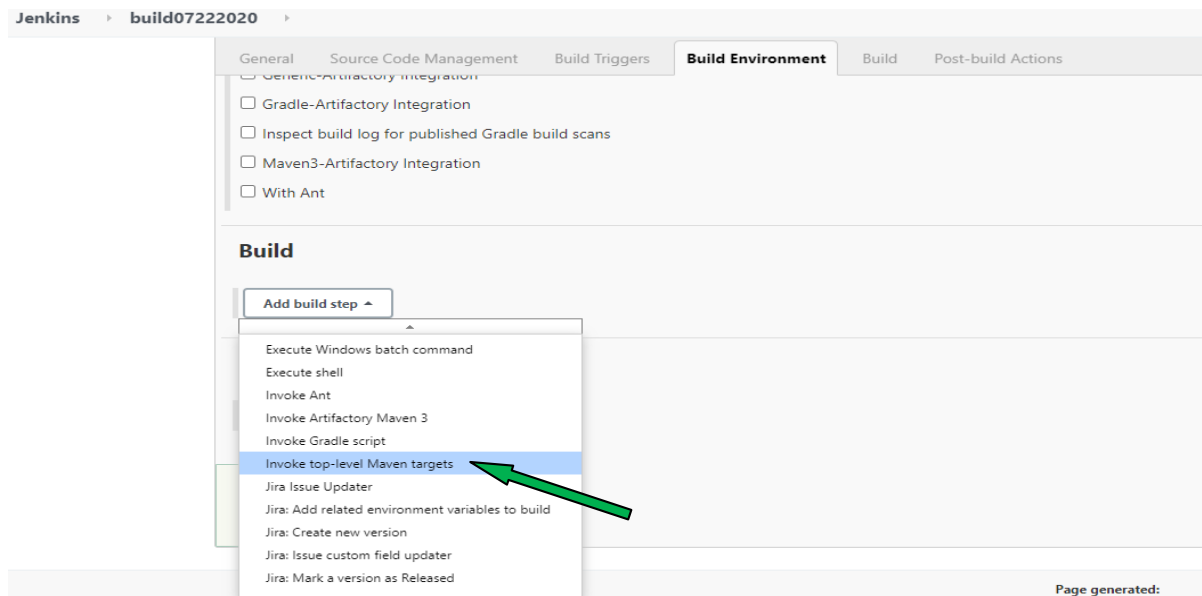


Jenkins Freestyle Job Step-by-Step Procedure

- 9) In this option select the '*Generic-Artifactory integration*' selection and provide details about repository name to which we want the binaries or the package to be deployed for sharing with other teams, like QA or Deployment. Here we also have to mentioned the file name we want to push to the remote repository on Artifactory.



- 10) In the build tab, we now select '*invoke Top Level Maven Targets*' and mention the Maven goal, e.g. '*clean*', '*test*', '*package*' etc.



- 11) In the *Post Build Action* tab, we can select option that will complement the selection in *Build* tab. So we select option to archive artifacts, binaries that gets created as an result of maven build task.

Jenkins Freestyle Job Step-by-Step Procedure

Jenkins > build07222020 >

General Source Code Management Build Triggers Build Environment **Build** Post-build Actions

Build

Invoke top-level Maven targets [X]

Goals [v] [Advanced...]

[Add build step v]

Post-build Actions

Archive the artifacts [X]

Files to archive [Advanced...]

[Add post-build action v]

[Save] [Apply]

To run the *Build manually*, we can click on the Build Now option.

Jenkins > build07222020 >

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Project

Configure

Git Polling Log

Rename

Project build07222020

Workspace

Recent Changes

Permalinks

Build History trend

Atom feed for all Atom feed for failures