

Module 3: Plugins and its uses

Case Study Solution

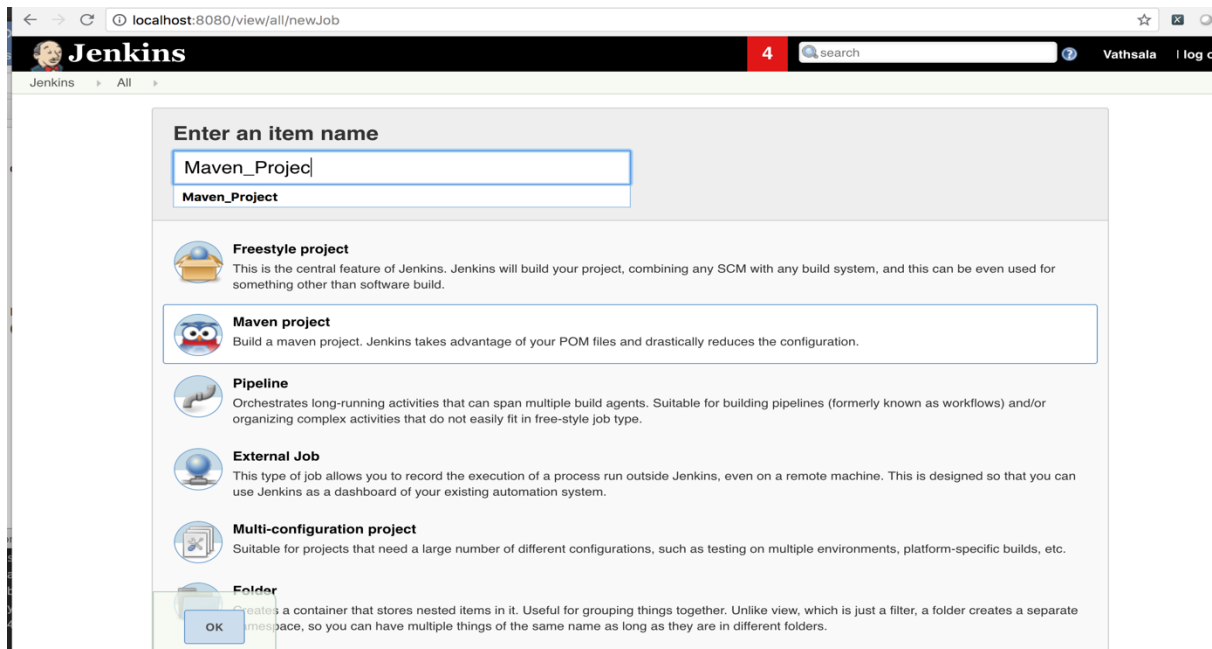
edureka!

edureka!

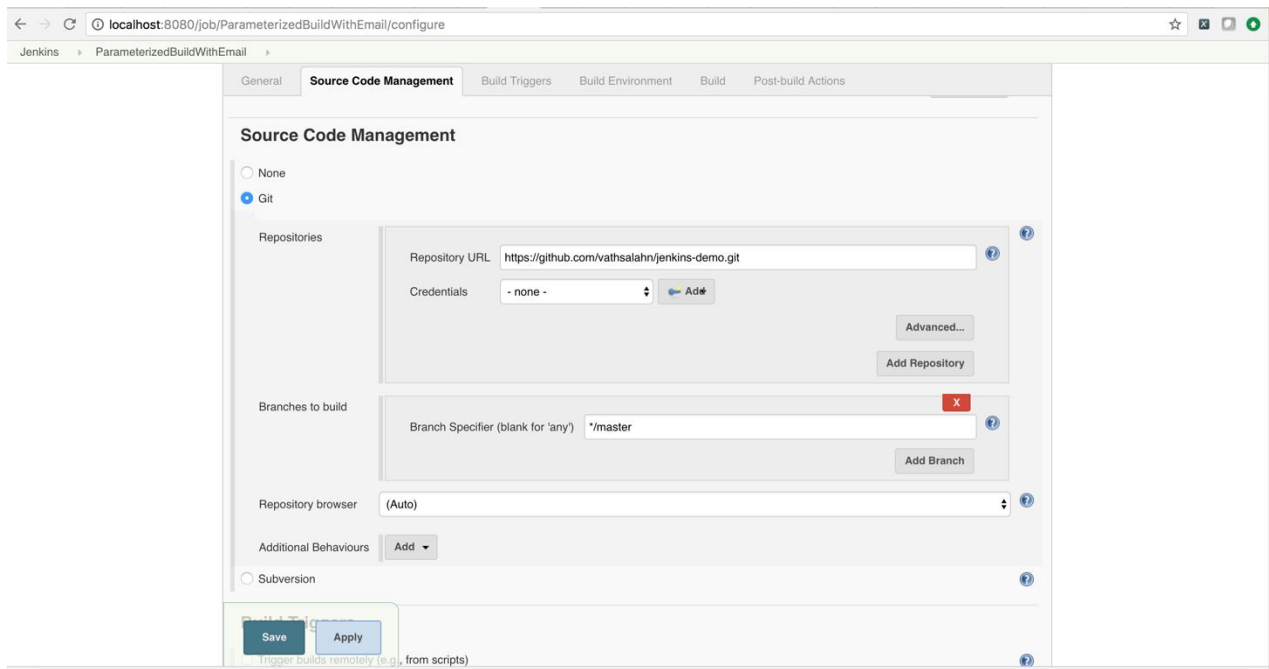
© Brain4ce Education Solutions Pvt. Ltd.

Case study Solution

Step 1: Create a Jenkins “Maven Project” and name the project

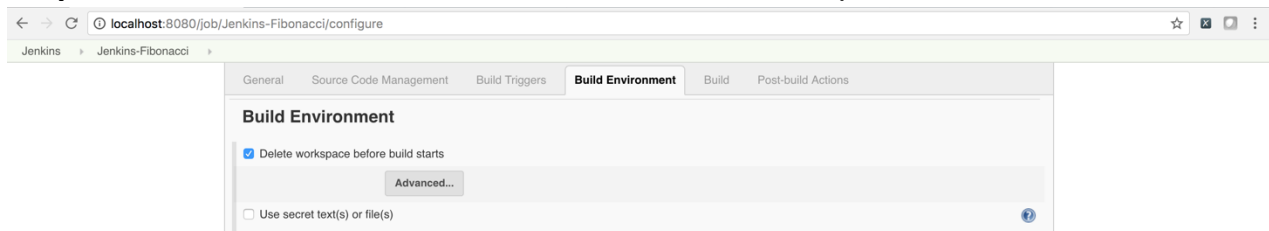


Step 2: Enable “Git” in Source Code Management and Provide the repository url (<https://github.com/edureka-devops/jenkins-demo.git>)



edureka!

Step 3: Enable the Build Environment to “Delete workspace before build starts”



Step 5: In Build select run the maven script by giving the path of pom.xml (MavenProject/pom.xml) and set Goals and Options to “ -B -DskipTests cleanpackage”

The screenshot shows the Jenkins configuration page for a job named 'Maven_Project'. The 'Build' tab is selected. Under 'Post-build Actions', the 'Build' section is expanded. The 'Root POM' field is set to 'MavenProject/pom.xml'. The 'Goals and options' field is set to '-B -DskipTests clean package'. Below this, the 'Post Steps' section is visible, showing a radio button selection for 'Run only if build succeeds' (which is selected). There is also a checkbox for 'Should the post-build steps run only for successful builds, etc.' which is unchecked. A 'Post-build step' is added with the title 'Invoke top-level Maven targets'. In this step, the 'Maven Version' is set to '(Default)' and the 'Goals' are set to 'test'. At the bottom, there are 'Save' and 'Apply' buttons.

Step 6: Update the Post-build Actions to archive the generated artifacts and trigger a downstream job for deploying the archived artifacts.

The screenshot shows the Jenkins configuration page for the same job 'Maven_Project', but now the 'Post-build Actions' tab is selected. The 'Post-build Actions' section is expanded. The first action is 'Archive the artifacts', with 'Files to archive' set to 'MavenProject/multi3/target/*.war, MavenProject/pom.xml'. The second action is 'Build other projects', with 'Projects to build' set to 'Mavendeploy'. Under this action, the radio button 'Trigger only if build is stable' is selected. The third action is 'Publish Maven Artifacts'. At the bottom, there are 'Save' and 'Apply' buttons.

Step 7: Apply and save the job

Step 8: Create a new maven job for deployment and copy the artifacts archived in the previous job under pre-steps.

The screenshot shows the Jenkins configuration page for a job named 'Mavendeploy'. The 'Pre Steps' tab is selected. Under the 'Pre Steps' section, there is a configuration for 'Copy artifacts from another project'. The 'Project name' is set to 'Maven_Project'. A warning message states: 'Artifacts will be copied from all modules of this Maven project; click the help icon to learn about selecting a particular module.' The 'Which build' dropdown is set to 'Upstream build that triggered this job'. There is an unchecked checkbox for 'Use "Last successful build" as fallback'. Below these are fields for 'Artifacts to copy', 'Artifacts not to copy', 'Target directory', and 'Parameter filters'. At the bottom, there are checkboxes for 'Flatten directories', 'Optional', and 'Fingerprint Artifacts' (which is checked). 'Advanced...' buttons are present on the right side of the configuration area. At the bottom left, there are 'Save' and 'Apply' buttons.

Step 9: Upload the artifacts to a webserver (tomcat) and trigger an email to notify the status.

The screenshot shows the Jenkins configuration page for the same job, now on the 'Post-build Actions' tab. Under the 'Post-build Actions' section, there is a configuration for 'Deploy war/ear to a container'. The 'WAR/EAR files' field is set to '**/*.war'. The 'Context path' field is empty. Under the 'Containers' section, 'Tomcat 7.x' is selected. The 'Credentials' dropdown is set to 'tomcat/***** (apache)'. The 'Tomcat URL' is set to 'http://localhost:8080'. There is an 'Add Container' button. Below this, there is an unchecked checkbox for 'Deploy on failure'. Under the 'Editable Email Notification' section, there is an unchecked checkbox for 'Disable Extended Email Publisher'. The 'Project From' field is set to 'vathsala.hn22@gmail.com'. The 'Project Recipient List' field is also set to 'vathsala.hn22@gmail.com'. 'Save' and 'Apply' buttons are at the bottom left.

Step 10: Upon triggering the job, the execution has to be success and a report regarding disk usage has to be generated as below.

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Maven project

Configure

Modules

Rebuild Last

Rename

Maven project MavenTesting

This is a Maven Project Example

edit description

Disable Project

Workspace

Last Successful Artifacts

Recent Changes

Latest Test Result (no failures)

Disk Usage

Latest Test Result (no failures)

multi1-3.7-SNAPSHOT-sources.jar817 Bview

multi1-3.7-SNAPSHOT-tests.jar1.04 KBview

multi1-3.7-SNAPSHOT.jar1.01 KBview

Job560 KB

All builds560 KB

Locked builds-

All workspaces5 MB

Slave workspaces5 MB

Non-slave workspaces-

Build History

trend

find

#828-May-2018 19:2073 KB

#728-May-2018 19:1969 KB

#628-May-2018 19:1670 KB

#528-May-2018 18:4770 KB

#428-May-2018 18:4670 KB

#328-May-2018 18:4571 KB

#228-May-2018 00:5169 KB

Disk Usage Trend

Test Result Trend

localhost:8080/job/MavenTesting/8/console

JenkinsMavenTesting#8

[INFO] -----

[INFO] Reactor Summary:

[INFO]

[INFO] Simple Multi Modules Build 3.7-SNAPSHOT SUCCESS [0.399 s]

[INFO] Multi 1 SUCCESS [2.556 s]

[INFO] Multi 2 SUCCESS [1.161 s]

[INFO] Multi 3 3.7-SNAPSHOT SUCCESS [0.846 s]

[INFO] -----

[INFO] BUILD SUCCESS

[INFO] -----

[INFO] Total time: 5.438 s

[INFO] Finished at: 2018-05-28T19:21:04+05:30

[INFO] -----

Waiting for Jenkins to finish collecting data

[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/MavenTesting/MavenProject/pom.xml to org.jfrog.test/multi/3.7-SNAPSHOT/multi-3.7-SNAPSHOT.pom

[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/MavenTesting/MavenProject/multi3/pom.xml to org.jfrog.test/multi3/3.7-SNAPSHOT/multi3-3.7-SNAPSHOT.pom

[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/MavenTesting/MavenProject/multi3/target/multi3-3.7-SNAPSHOT.war to org.jfrog.test/multi3/3.7-SNAPSHOT/multi3-3.7-SNAPSHOT.war

[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/MavenTesting/MavenProject/multi1/pom.xml to org.jfrog.test/multi1/3.7-SNAPSHOT/multi1-3.7-SNAPSHOT.pom

[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/MavenTesting/MavenProject/multi1/target/multi1-3.7-SNAPSHOT.jar to org.jfrog.test/multi1/3.7-SNAPSHOT/multi1-3.7-SNAPSHOT.jar

[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/MavenTesting/MavenProject/multi1/target/multi1-3.7-SNAPSHOT-sources.jar to org.jfrog.test/multi1/3.7-SNAPSHOT/multi1-3.7-SNAPSHOT-sources.jar

[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/MavenTesting/MavenProject/multi1/target/multi1-3.7-SNAPSHOT-tests.jar to org.jfrog.test/multi1/3.7-SNAPSHOT/multi1-3.7-SNAPSHOT-tests.jar

[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/MavenTesting/MavenProject/multi2/pom.xml to org.jfrog.test/multi2/3.7-SNAPSHOT/multi2-3.7-SNAPSHOT.pom

[JENKINS] Archiving /Users/vatnaray/.jenkins/workspace/MavenTesting/MavenProject/multi2/target/multi2-3.7-SNAPSHOT.jar to org.jfrog.test/multi2/3.7-SNAPSHOT/multi2-3.7-SNAPSHOT.jar

channel stopped

Archiving artifacts

Started calculate disk usage of build

Finished calculate disk usage of build in 0 seconds

Started calculate disk usage of workspace

Finished calculate disk usage of workspace in 0 seconds