

Module 6: Automated and Continuous Deployment

Demo Document - 1

edureka!

© Brain4ce Education Solutions Pvt. Ltd.

Demo: Deployment of Java App as per Jenkins free style project

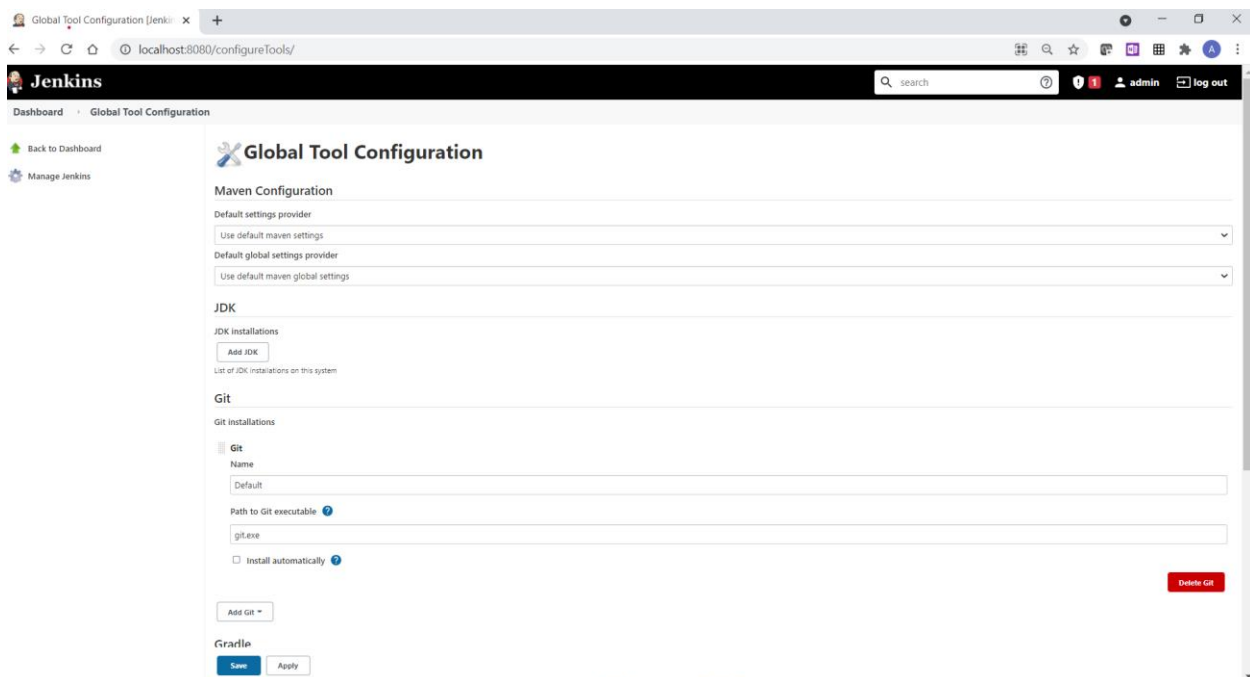
Problem statement: Execute commands to do deployment of JAVA application using Jenkins freestyle project

Solution:

Step 1: This demo will help the participants to understand how to build JAVA application using freestyle project in Jenkins.

Please start Jenkins using preferred option. We will be configuring JDK and Maven globally for the Jenkins environment.

URL to access: <http://localhost:8080/configureTools/>



The screenshot shows the Jenkins web interface at the URL `localhost:8080/configureTools/`. The page title is "Global Tool Configuration". On the left sidebar, there are links for "Back to Dashboard" and "Manage Jenkins". The main content area is titled "Global Tool Configuration" and contains several sections:

- Maven Configuration:** Includes two dropdown menus for "Default settings provider" and "Default global settings provider", both currently set to "Use default maven settings".
- JDK:** Includes a section for "JDK installations" with an "Add JDK" button and a note "List of JDK installations on this system".
- Git:** Includes a section for "Git installations" with a "Add Git" button. Below this, there are input fields for "Name" (set to "Default"), "Path to Git executable" (set to "git.exe"), and a checkbox for "Install automatically" (unchecked). A "Delete Git" button is visible on the right.
- Gradle:** Includes a section with "Save" and "Apply" buttons.

The top of the page features the Jenkins logo, a search bar, and user information (admin) with a "log out" link.

Module 6 – Automated and Continuous Deployment

Step 2: In this step, you will configure JDK pointing to local JDK available on your workstation. You do have an option of installing the tools automatically. But we won't be using it.

URL to access: <http://localhost:8080/configureTools/>

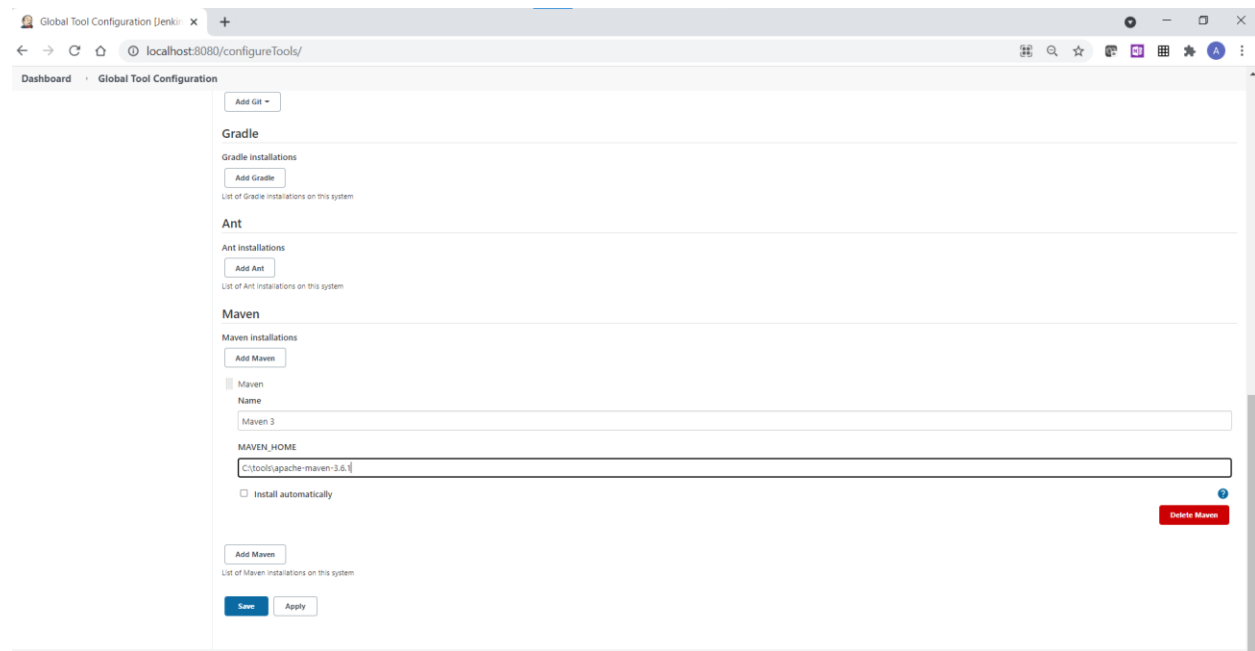
The screenshot shows the Jenkins 'Global Tool Configuration' page. The 'Maven Configuration' section is active, showing default settings for the Maven settings provider. Below this, the 'JDK' section is expanded, displaying a list of JDK installations. One installation is visible with the name 'JDK', version 'JAVA 8', and a path to the JDK executable. The 'Install automatically' checkbox is unchecked. At the bottom of the JDK section, there are 'Add JDK' and 'Delete JDK' buttons. The 'Git' section is also visible, showing a list of Git installations with a 'Default' name and a path to the Git executable.

Step 3: In this step, you will configure Maven pointing to local Maven available on your workstation.

URL to access: <http://localhost:8080/configureTools/>

The screenshot shows the Jenkins 'Global Tool Configuration' page with the 'Git' section expanded. It displays a list of Git installations with a 'Default' name and a path to the 'git.exe' executable. The 'Install automatically' checkbox is unchecked. Below the Git section, the 'Gradle' section is visible, showing a list of Gradle installations with an 'Add Gradle' button. The 'Ant' section is also visible, showing a list of Ant installations with an 'Add Ant' button. At the bottom, the 'Maven' section is visible, showing a list of Maven installations with an 'Add Maven' button. The 'Save' and 'Apply' buttons are at the bottom of the page.

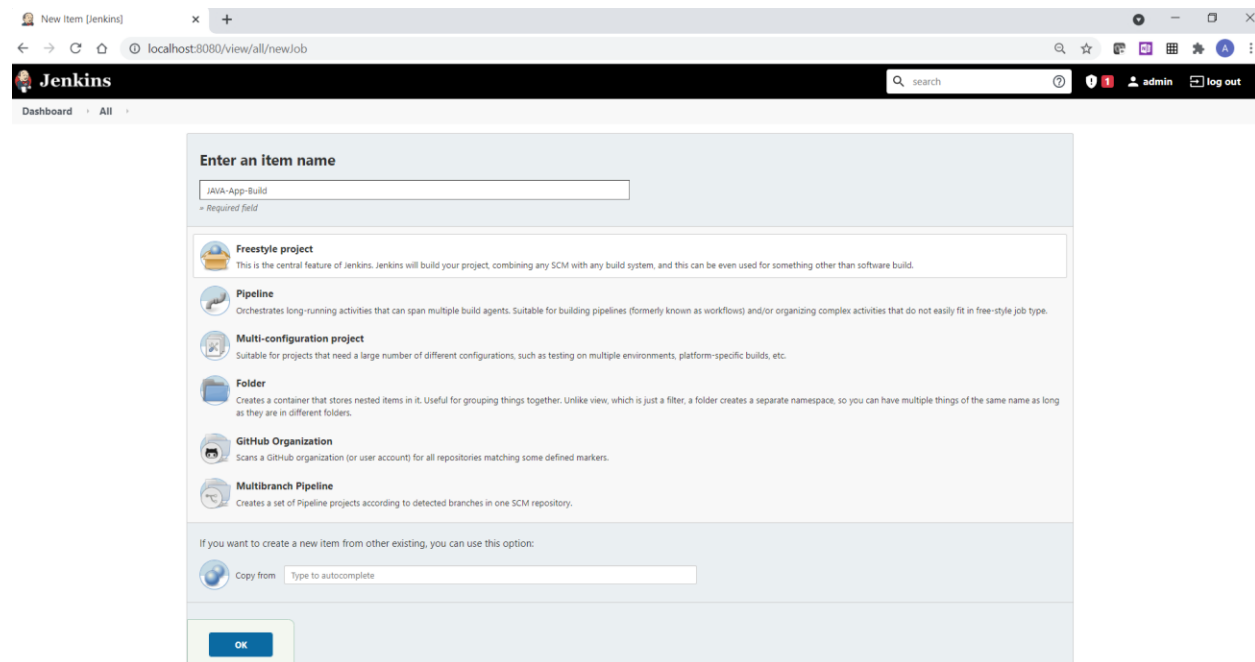
Module 6 – Automated and Continuous Deployment



Step 4: Now we will create a Jenkins freestyle project.

URL to access: <http://localhost:8080/view/all/newJob>

Please create a freestyle project named as 'JAVA-App-Build'.



This step will create a freestyle project. In freestyle project type, we need to use specific set of configurations provided by Jenkin tool.

Step 5: Let's configure GIT repository for the freestyle project.

Module 6 – Automated and Continuous Deployment

URL to access: <http://localhost:8080/job/JAVA-App-Build/configure>

The screenshot shows the Jenkins configuration page for a job named 'JAVA-App-Build'. The 'General' tab is selected, displaying the following sections:

- Description:** A text area for describing the job.
- [Plain text] Preview:** A preview of the job's configuration.
- Discard old builds:** A checkbox to discard old builds.
- GitHub project:** A checkbox to enable GitHub project integration.
- This build requires lockable resources:** A checkbox to require lockable resources.
- This project is parameterized:** A checkbox to parameterize the project.
- Throttle builds:** A checkbox to throttle builds.
- Disable this project:** A checkbox to disable the project.
- Execute concurrent builds if necessary:** A checkbox to execute concurrent builds if necessary.
- Advanced...** A button to expand advanced options.
- Source Code Management:** A section with a radio button for 'None' and a radio button for 'Git'.
- Build Triggers:** A section with checkboxes for 'Trigger builds remotely (e.g., from scripts)', 'Build after other projects are built', 'Build periodically', 'GitHub hook trigger for GITSCM polling', and 'Poll SCM'.
- Build Environment:** A section with a checkbox for 'Delete workspace before build starts'.
- Save** and **Apply** buttons at the bottom.

Git url - <https://github.com/JodaOrg/joda-time.git>

The screenshot shows the Jenkins configuration page for a job named 'JAVA-App-Build', with the 'Source Code Management' tab selected. The configuration is as follows:

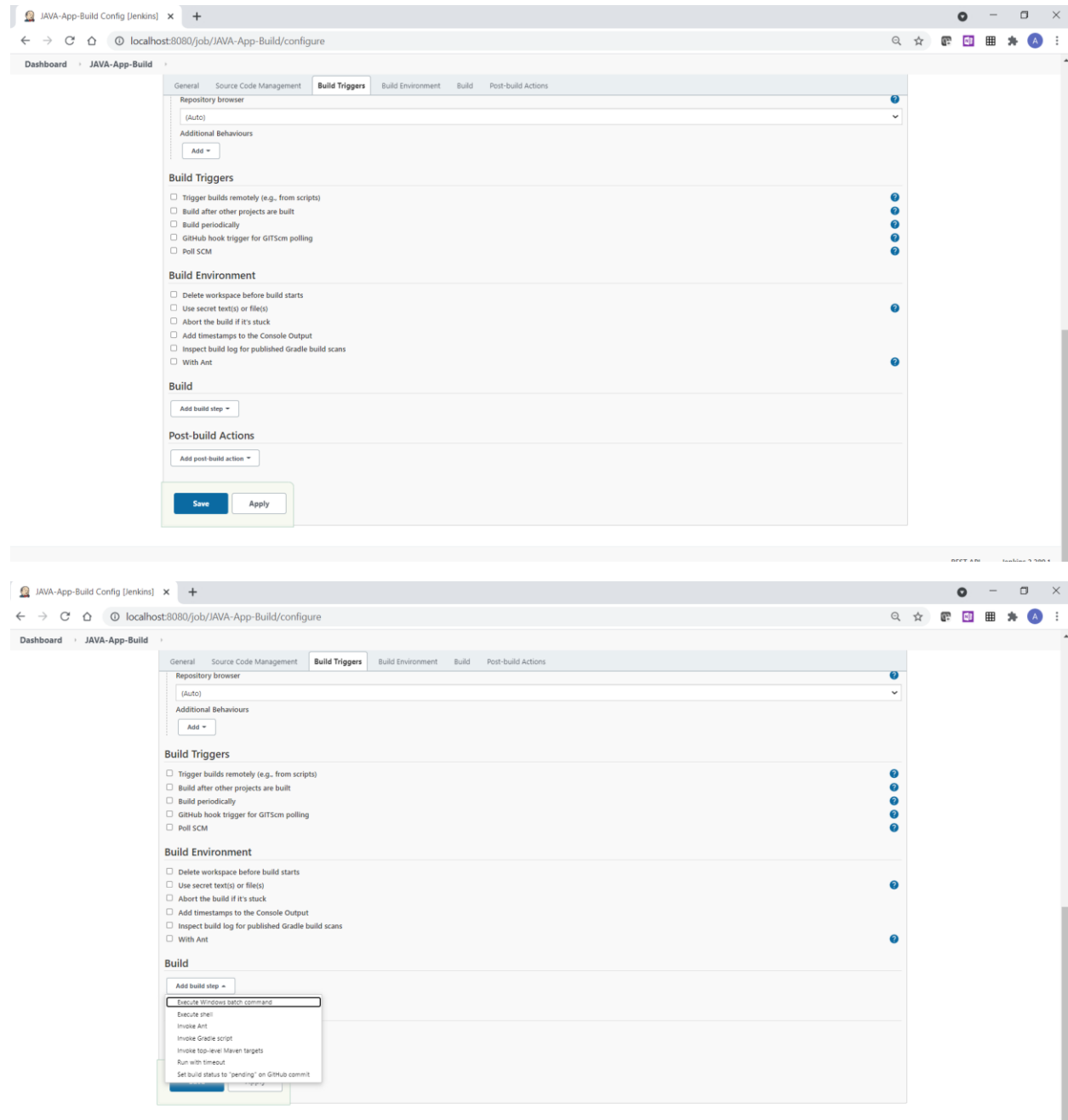
- Source Code Management:** The 'Git' radio button is selected.
- Repositories:** A section containing a 'Repository URL' field with the value 'https://github.com/JodaOrg/joda-time.git', a 'Credentials' dropdown menu set to 'none', and an 'Add Repository' button.
- Branches to build:** A section with a 'Branch Specifier (blank for \'any\')' field containing '*/master' and an 'Add Branch' button.
- Repository browser:** A dropdown menu set to '(Auto)'.
- Additional Behaviours:** A section with an 'Add' button.
- Save** and **Apply** buttons at the bottom.

Module 6 – Automated and Continuous Deployment

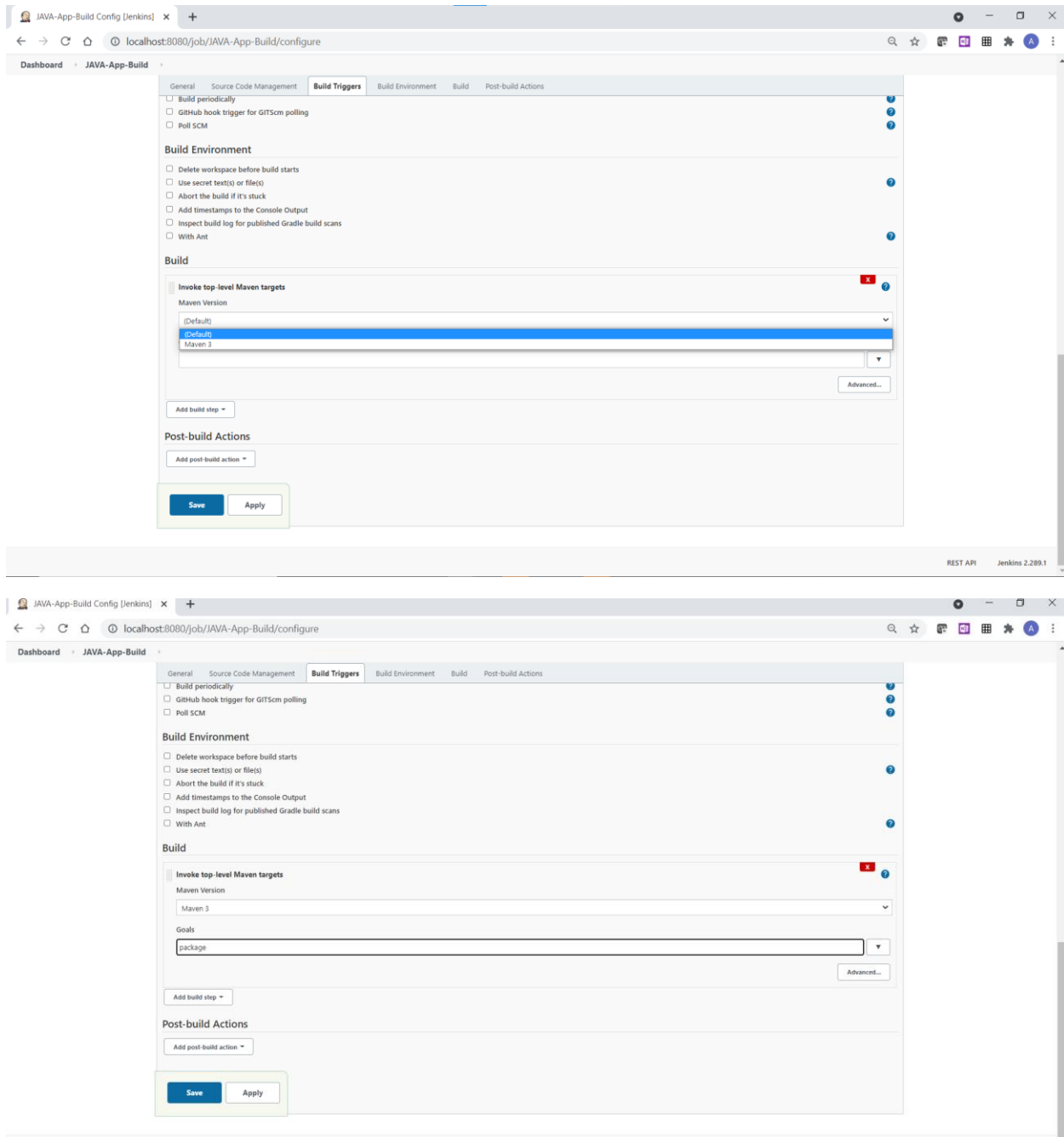
In this step, we are trying to refer a JAVA project which implements Joda-Time. Joda-Time provides replacement for JAVA date and time classes.

Step 6: Now we will add build step to invoke maven targets.

URL to access: <http://localhost:8080/job/JAVA-App-Build/configure>



Module 6 – Automated and Continuous Deployment



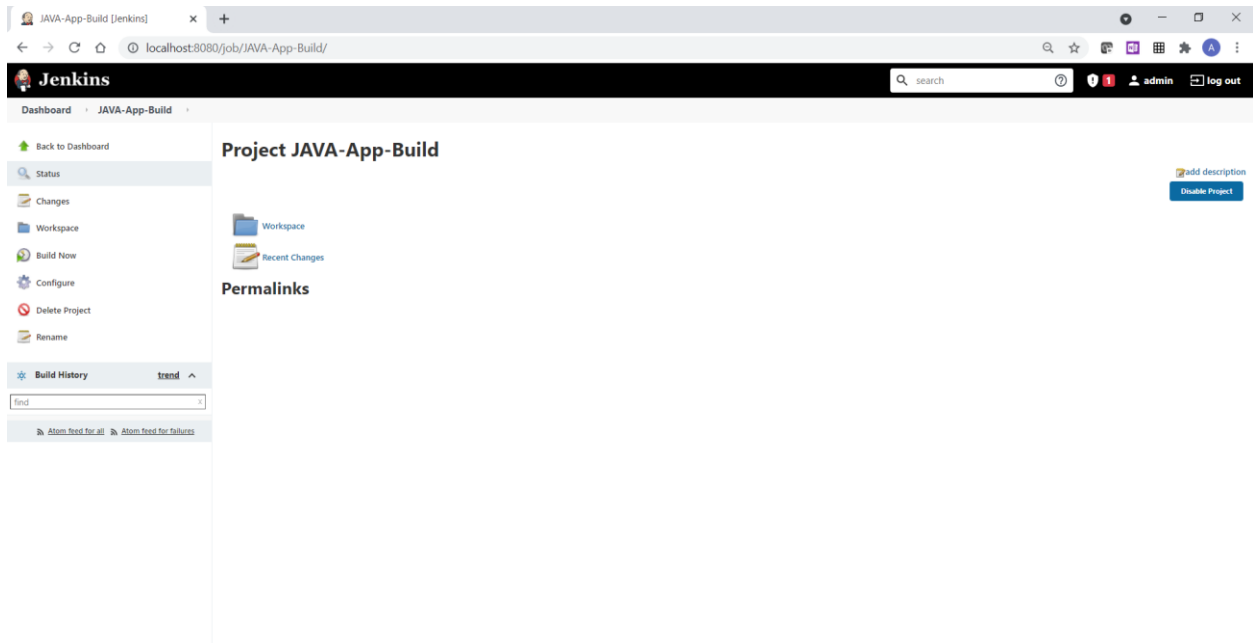
In the above step, we have asked Jenkins to invoke 'package' maven goal against the codebase.

Module 6 – Automated and Continuous Deployment

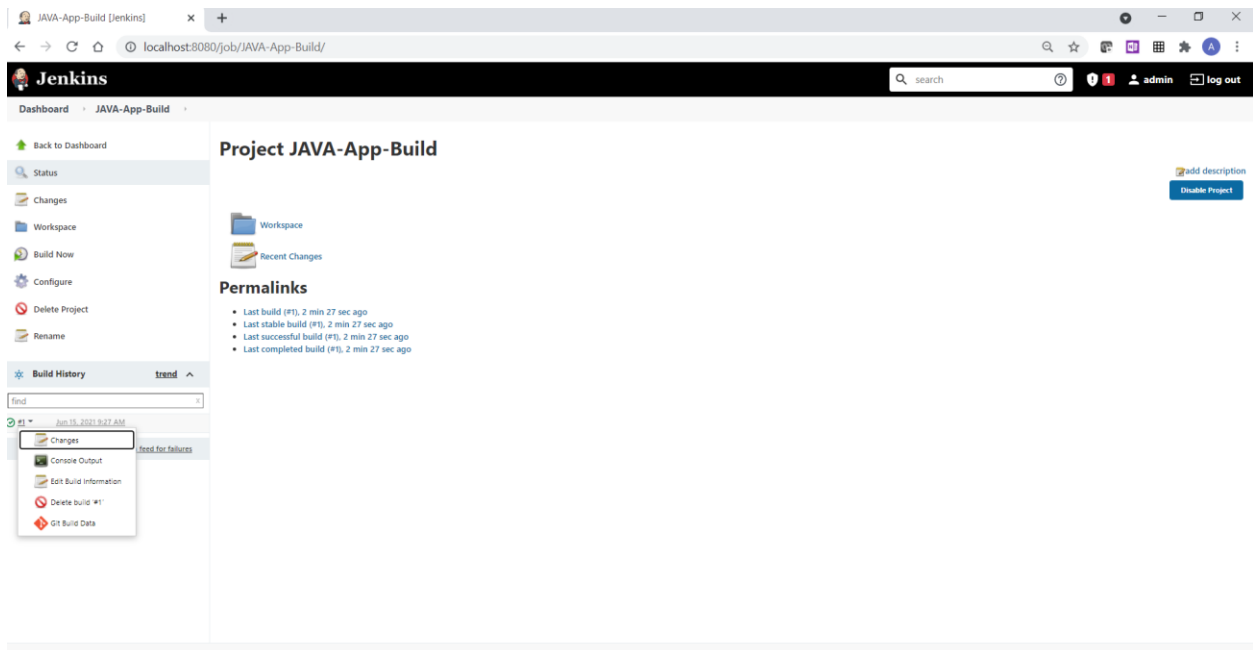
Step 7: Now we will build the project.

URL to access: <http://localhost:8080/job/JAVA-App-Build/>

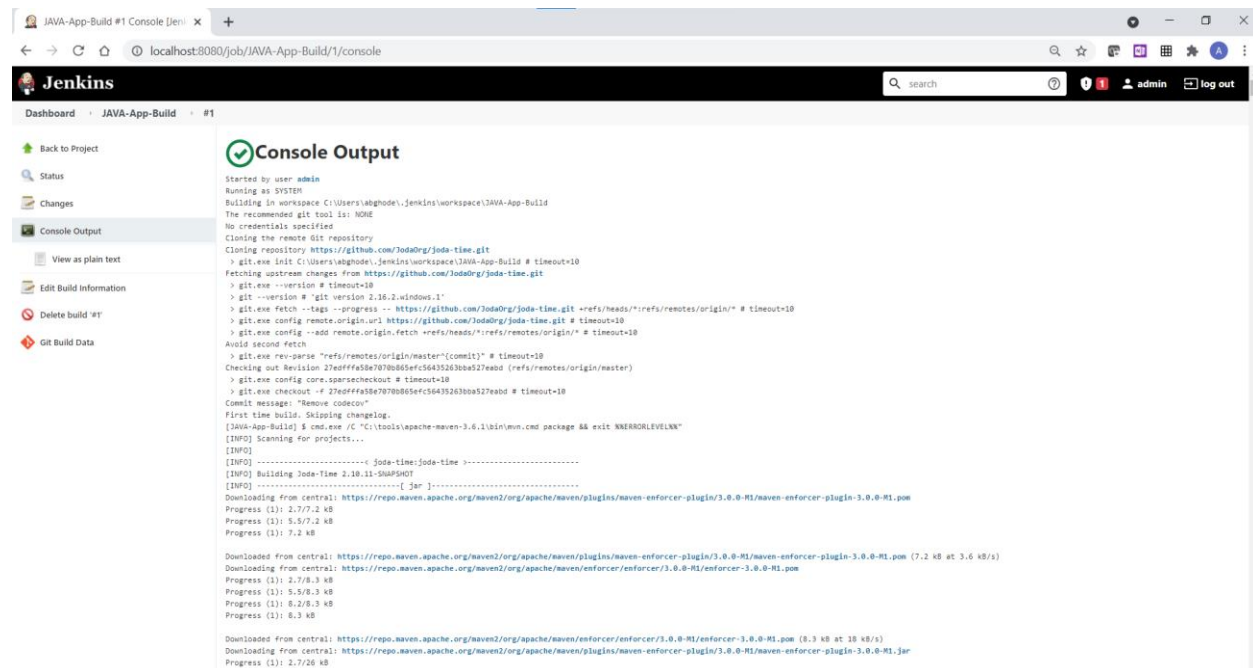
Click on ‘Build Now’



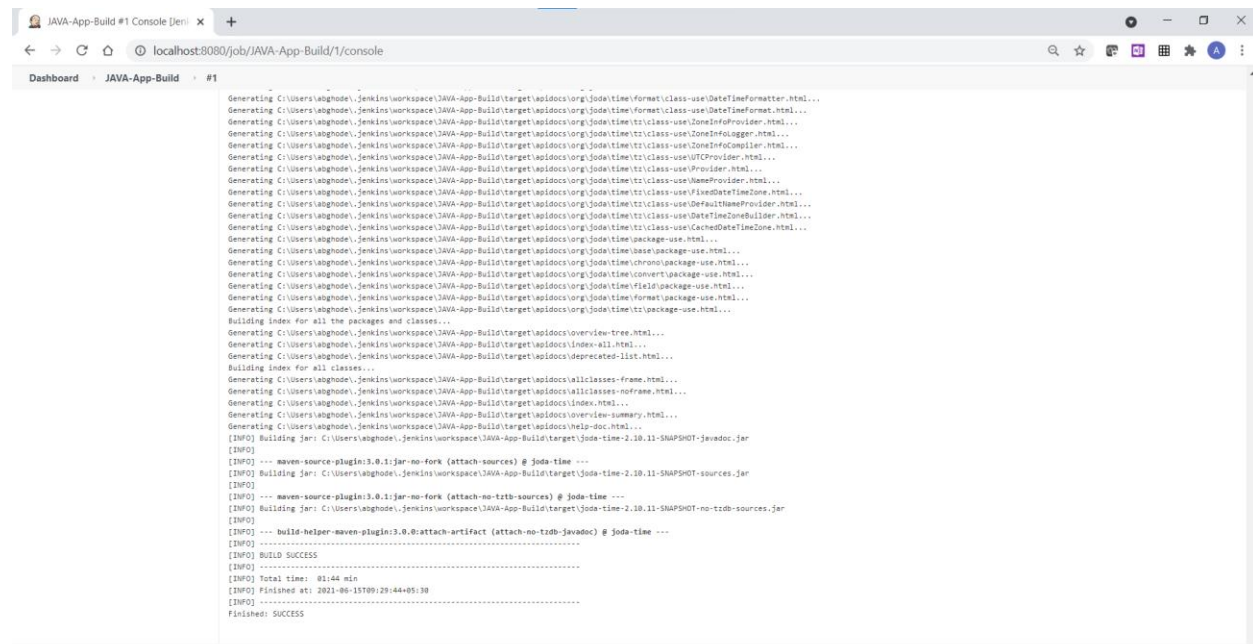
You can see that first build is initiated. To observe the build, you can click on ‘Console output’.



Module 6 – Automated and Continuous Deployment



Please scroll to the end. You should be able to see a message 'Finished: Success'



In this step, we have now executed 'package' maven goal. Please note this build is being performed in Jenkins environment.

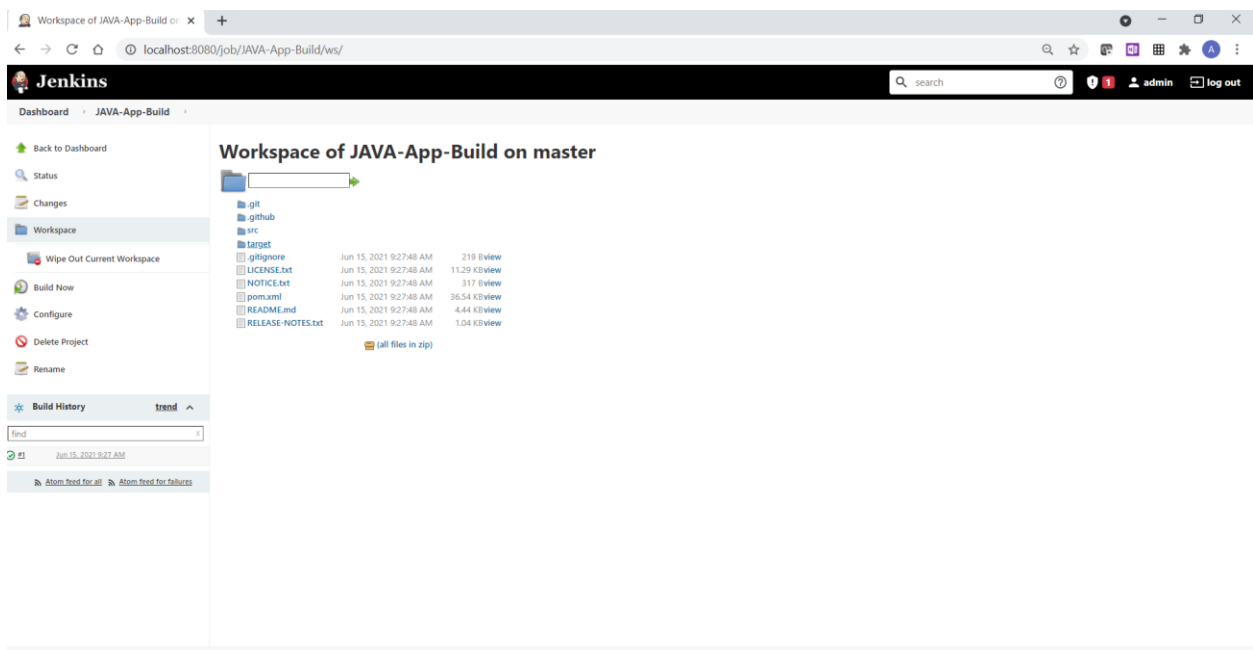
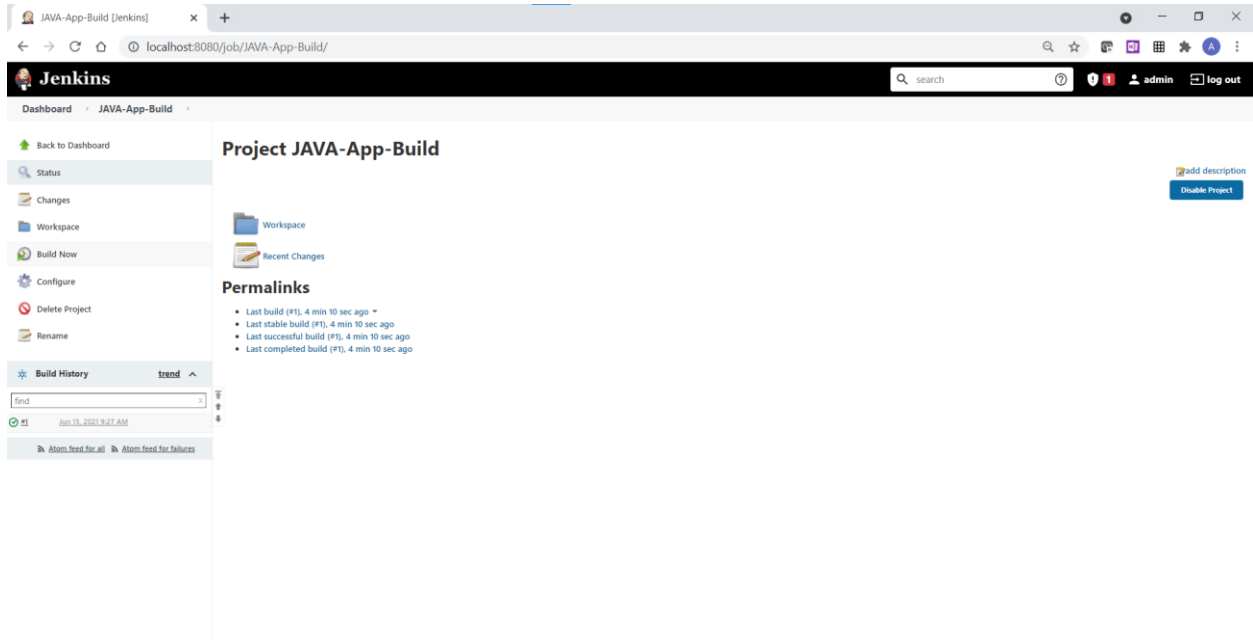
So you can see that there is a parity between this build and the build that you may have performed on your DEV environment.

Module 6 – Automated and Continuous Deployment

Step 8: Watch the workspace of your freestyle project for the application artifact (*.jar)

URL to access: <http://localhost:8080/job/JAVA-App-Build/>

Click on 'Workspace'



Module 6 – Automated and Continuous Deployment

The screenshot displays the Jenkins web interface in a browser. The address bar shows the URL `localhost:8080/job/JAVA-App-Build/ws/target/`. The Jenkins logo and navigation menu are visible at the top. The main content area is titled "Workspace of JAVA-App-Build on master" and shows a file tree for the `target/` directory. The file tree includes folders like `apidocs`, `classes`, `javadoc-bundle-options`, `maven-archiver`, `maven-status/maven-compiler-plugin`, `surefire-reports`, and `test-classes`. Below the file tree, a table lists build artifacts for the `joda-time-2.10.11-SNAPSHOT` project, including `jar`, `jar`, `no-tzdb.jar`, `no-tzdb-sources.jar`, and `sources.jar`, along with their build times and sizes.

File Name	Build Time	Size	Action
joda-time-2.10.11-SNAPSHOT.jar	Jun 15, 2021 9:29:12 AM	629.33 KB	view
joda-time-2.10.11-SNAPSHOT-javadoc.jar	Jun 15, 2021 9:29:40 AM	1.33 MB	view
joda-time-2.10.11-SNAPSHOT-no-tzdb.jar	Jun 15, 2021 9:29:13 AM	427.32 KB	view
joda-time-2.10.11-SNAPSHOT-no-tzdb-sources.jar	Jun 15, 2021 9:29:44 AM	814.96 KB	view
joda-time-2.10.11-SNAPSHOT-sources.jar	Jun 15, 2021 9:29:42 AM	815.70 KB	view

Your Jenkins build has created set of artifacts in the targets folder of your codebase now.