

Module 7: Jenkins Pipeline

Demo Document - 2

edureka!

© Brain4ce Education Solutions Pvt. Ltd.

Demo: Configuring Jenkins pipeline plugin

Problem statement: Commands to configure Jenkins pipeline plugin

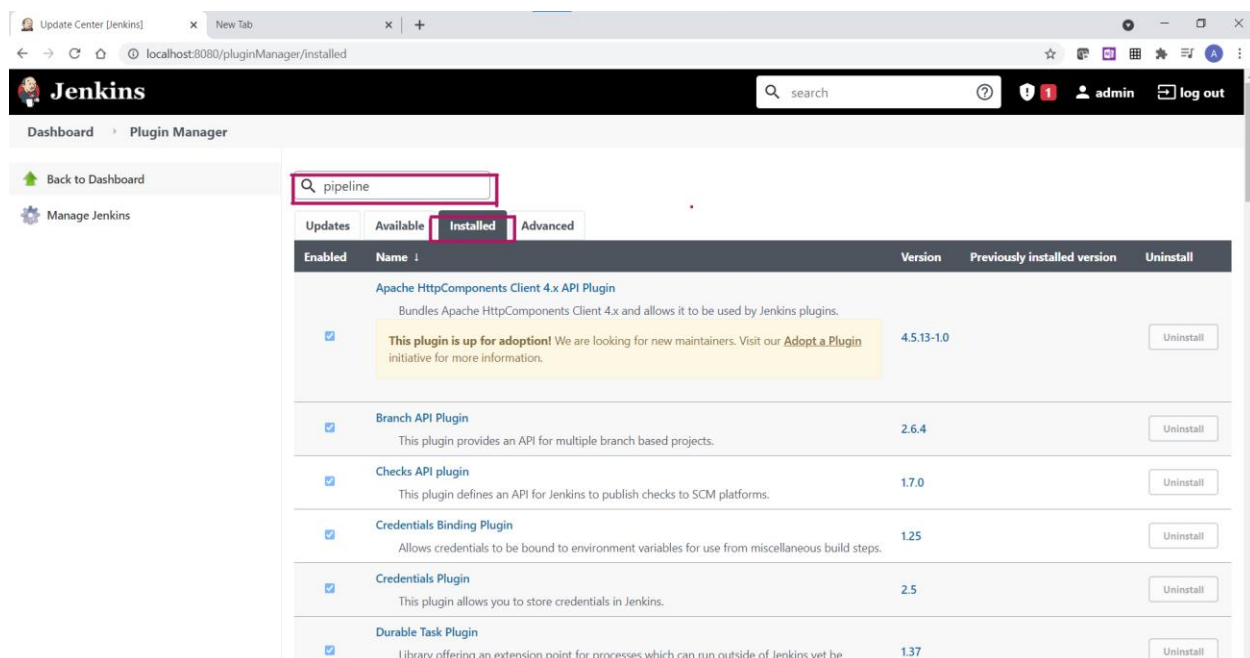
Solution:

Step 1: Check installed plugins for Pipeline.

URL to access: <http://localhost:8080/pluginManager/installed>

Please filter the list of installed plugins using 'pipeline'.

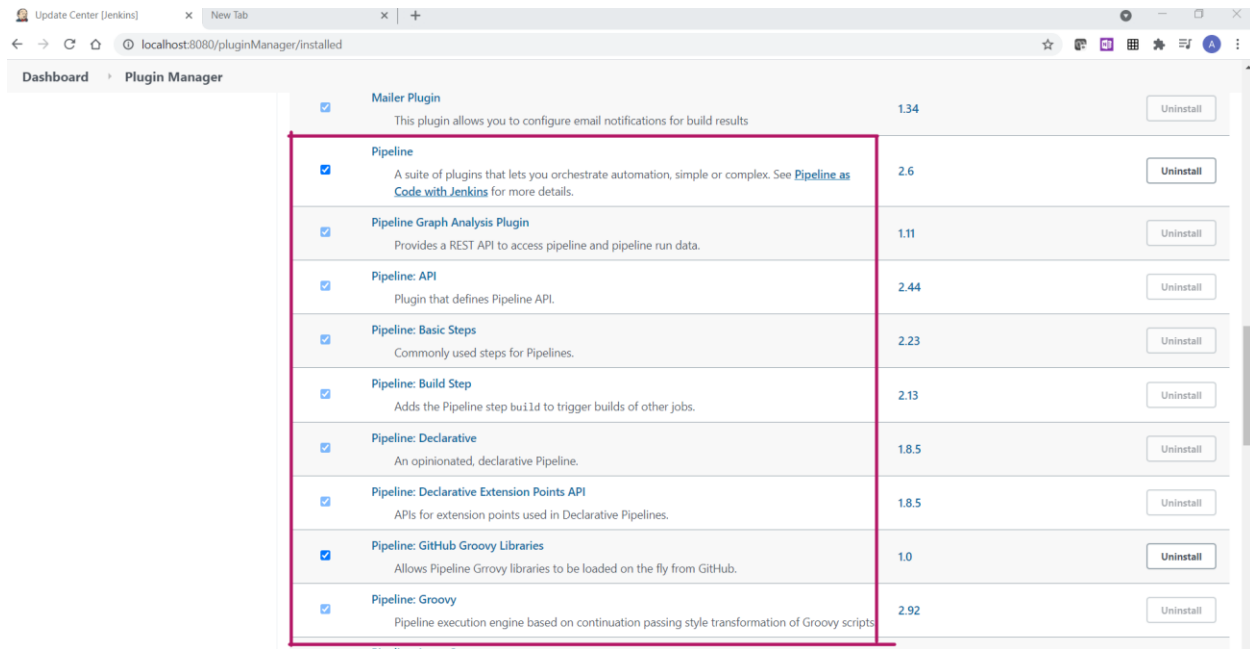
If you scroll down the list, you would see list of plugins installed. These plugins got installed since we installed suggested plugins in one of the previous lab/demo.



The screenshot shows the Jenkins Plugin Manager interface. The search bar at the top contains the text 'pipeline'. The 'Installed' tab is selected, displaying a list of installed plugins. The table has columns for 'Enabled', 'Name', 'Version', 'Previously installed version', and 'Uninstall'. The following table represents the data shown in the screenshot:

Enabled	Name	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	Apache HttpComponents Client 4.x API Plugin Bundles Apache HttpComponents Client 4.x and allows it to be used by Jenkins plugins. This plugin is up for adoption! We are looking for new maintainers. Visit our Adopt a Plugin initiative for more information.	4.5.13-1.0		<button>Uninstall</button>
<input checked="" type="checkbox"/>	Branch API Plugin This plugin provides an API for multiple branch based projects.	2.6.4		<button>Uninstall</button>
<input checked="" type="checkbox"/>	Checks API plugin This plugin defines an API for Jenkins to publish checks to SCM platforms.	1.7.0		<button>Uninstall</button>
<input checked="" type="checkbox"/>	Credentials Binding Plugin Allows credentials to be bound to environment variables for use from miscellaneous build steps.	1.25		<button>Uninstall</button>
<input checked="" type="checkbox"/>	Credentials Plugin This plugin allows you to store credentials in Jenkins.	2.5		<button>Uninstall</button>
<input checked="" type="checkbox"/>	Durable Task Plugin Library offering an extension point for processes which can run outside of Jenkins yet be	1.37		<button>Uninstall</button>

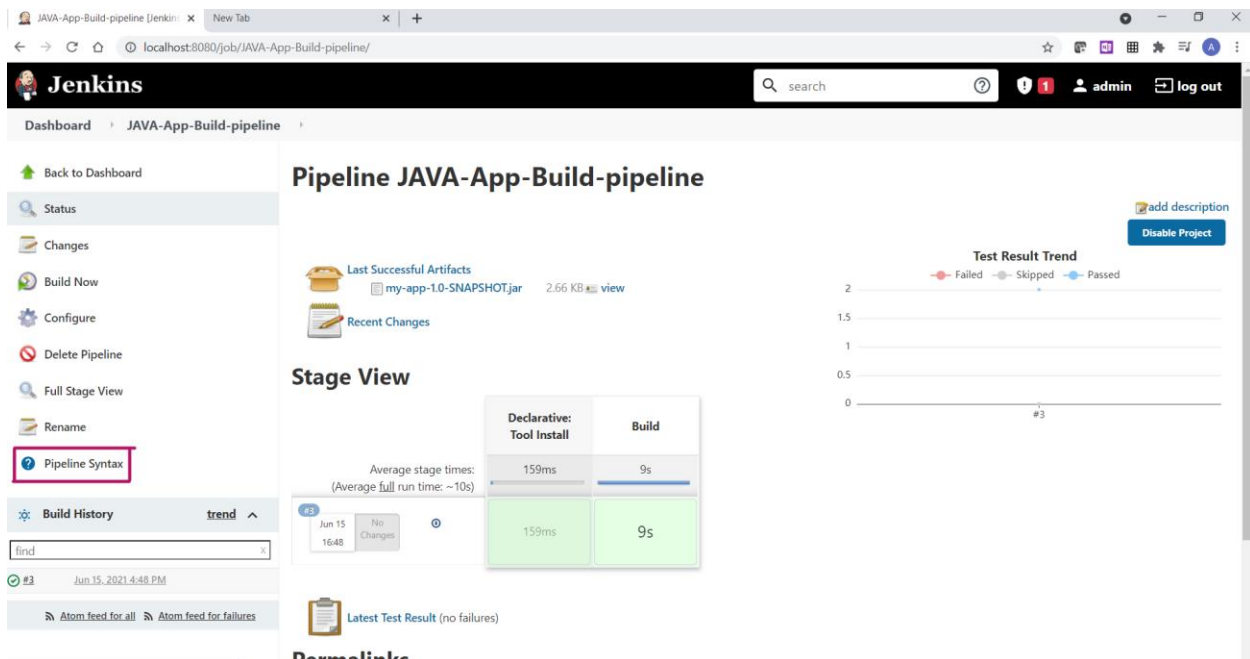
Module 7 – Jenkins Pipeline



Step 2: Try Pipeline syntax option provided by Jenkins.

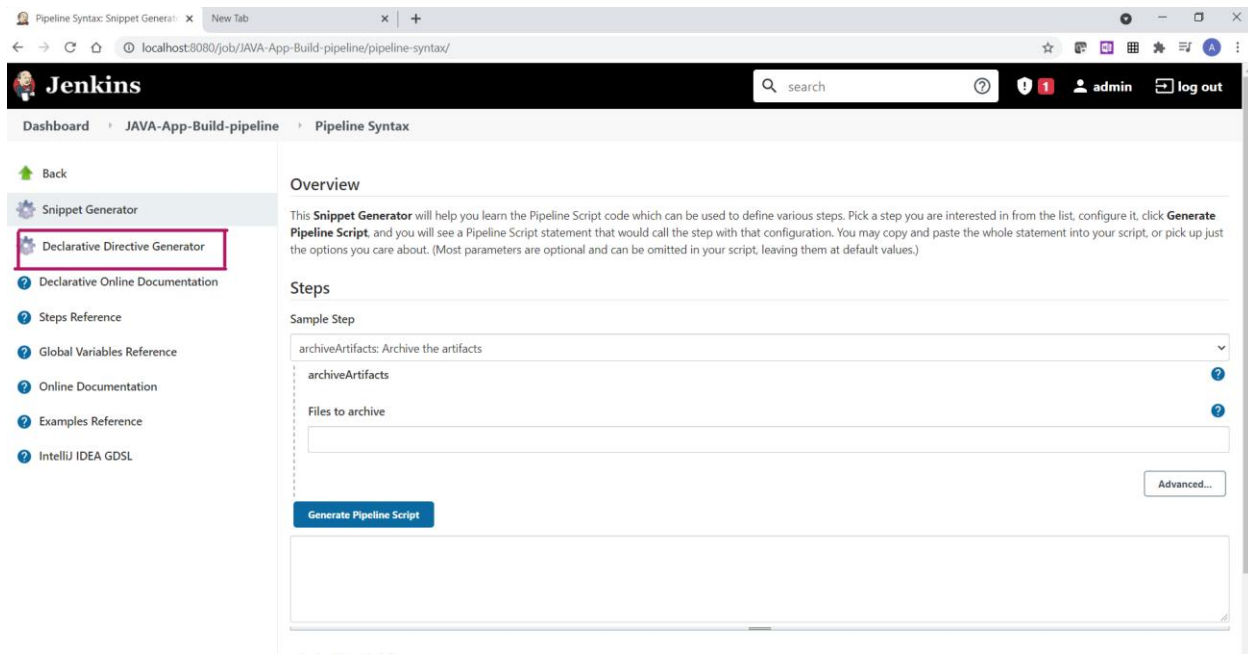
URL to access: <http://localhost:8080/job/JAVA-App-Build-pipeline/>

Click on 'Pipeline syntax'



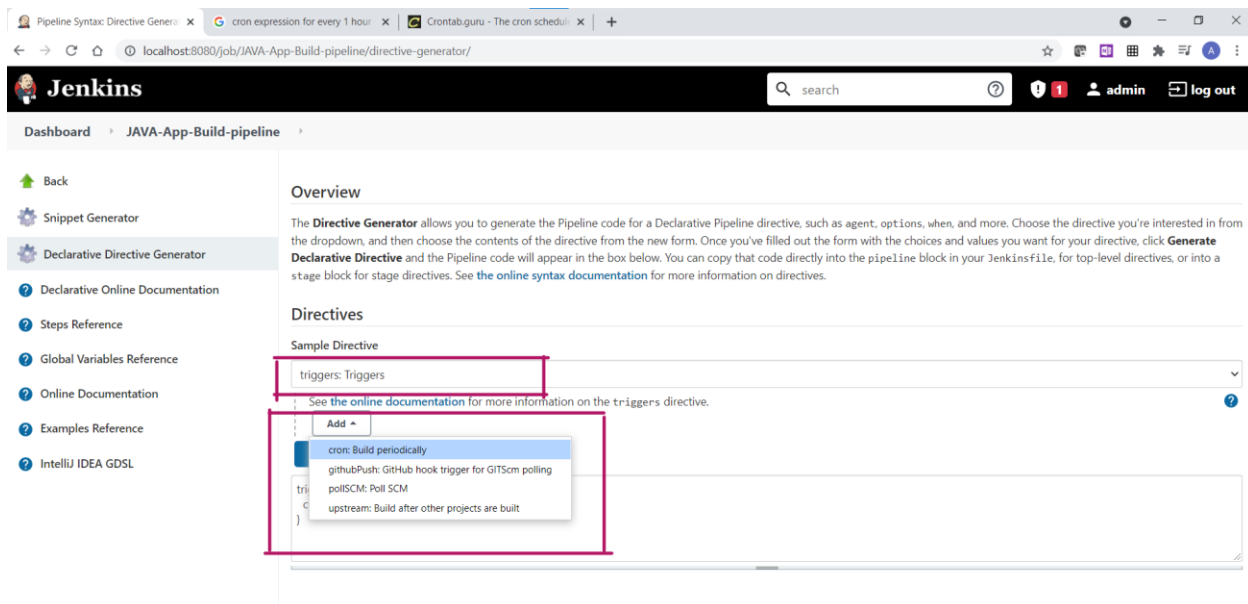
Module 7 – Jenkins Pipeline

Now, click on 'Declarative directive generator'.



You can try generating directive for triggers.

URL to access: `http://localhost:8080/job/JAVA-App-Build-pipeline/directive-generator/`



Module 7 – Jenkins Pipeline

Please enter below details while configuring directive.

Please select triggers as sample directive.

Inside schedule, put below expression **H 2 * * ***

Click on 'Generate Declarative directive'

Copy the directive generated. We will use it subsequent step.

The screenshot shows the 'Pipeline Syntax: Directive Generator' web application. The browser tabs include 'cron expression for every 1 hour', 'Crontab.guru - The cron scheduler', and the application itself. The URL is 'localhost:8080/job/JAVA-App-Build-pipeline/directive-generator/'. The left sidebar has a 'Dashboard' link and a 'JAVA-App-Build-pipeline' link. The main content area is titled 'Directives'. Under 'Sample Directive', there is a dropdown menu set to 'triggers: Triggers'. Below it, a link points to 'the online documentation for more information on the triggers directive.' A section titled 'Build periodically' contains a 'Schedule' field with the cron expression 'H 2 * * *'. Below the schedule field, a message states: 'Would last have run at Wednesday, June 16, 2021 2:36:36 AM IST; would next run at Thursday, June 17, 2021 2:36:36 AM IST.' There is a 'Delete' button. At the bottom, there is an 'Add' button and a 'Generate Declarative Directive' button. Below the 'Generate Declarative Directive' button, the generated directive is shown:

```
triggers {  
  cron 'H 2 * * *'  
}
```

In this step, we are trying to perform builds periodically. You can try cron expression **cron 'H 2 * * *'**.

It indicates that build will be scheduled 2AM everyday.

Module 7 – Jenkins Pipeline

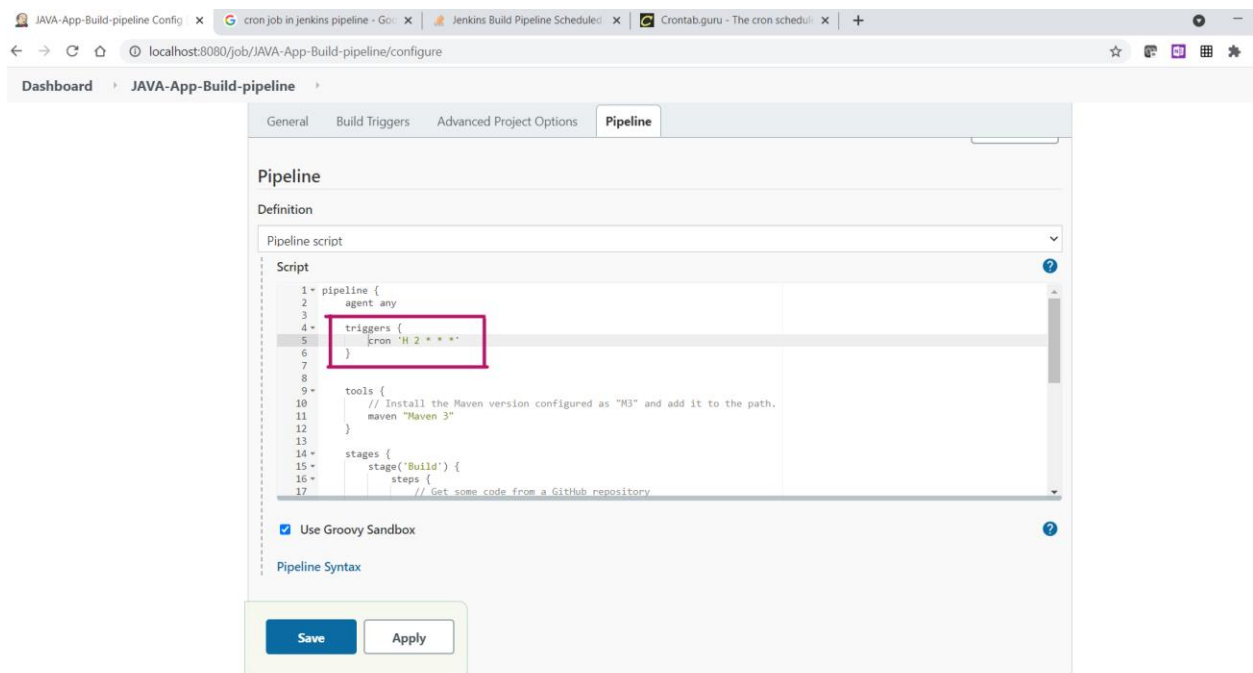
Step 3: Configure pipeline project to use trigger.

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

Click on 'Pipeline'

Add the directive generated in previous step.

Save the configuration.



Now, the build of this pipeline project will be triggered at scheduled interval.