

# Module 4: Continuous Integration With Jenkins

---

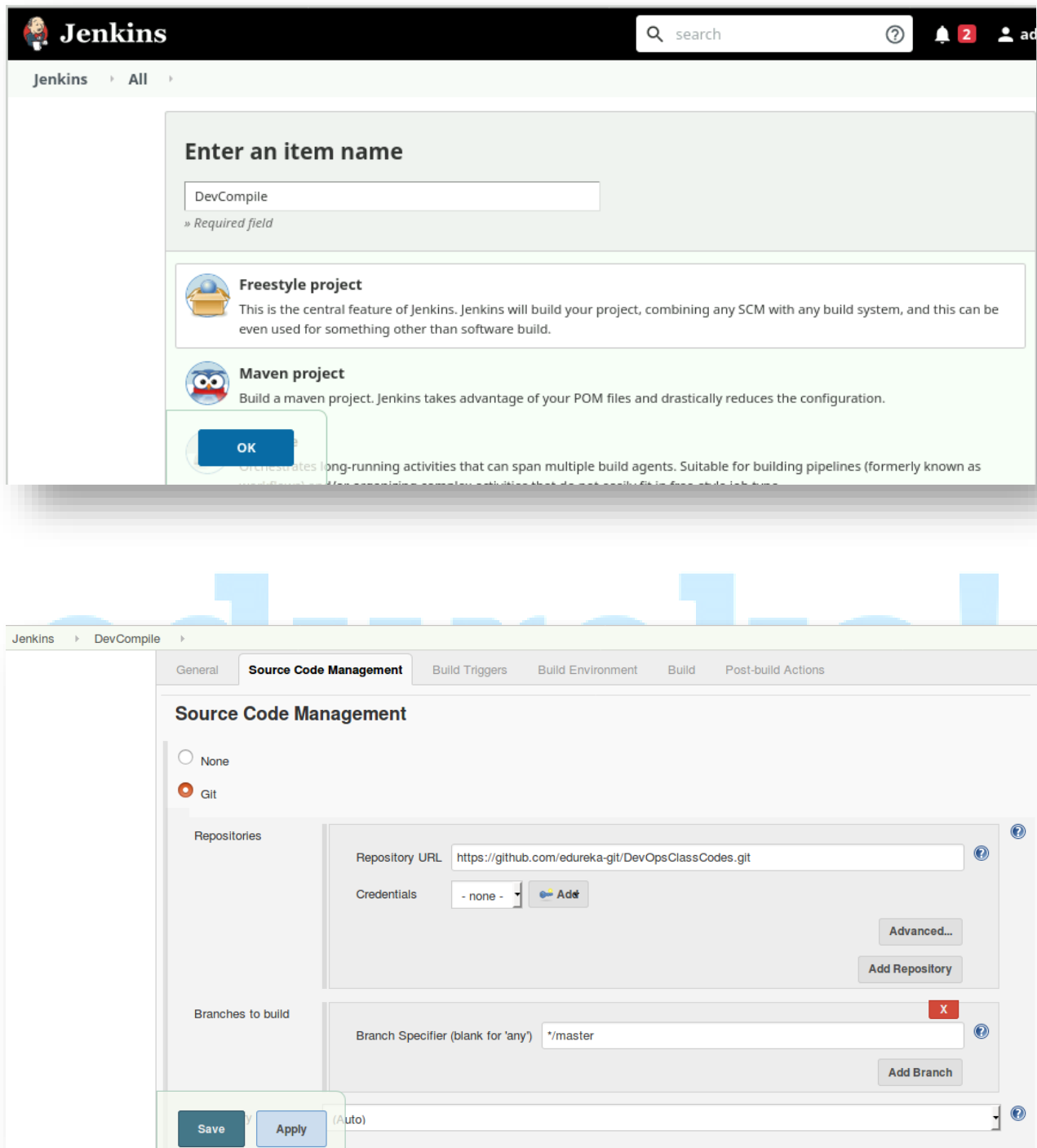
Demo Document - 1

edureka!

**edureka!**

© Brain4ce Education Solutions Pvt. Ltd.

## Demo – 1: Create Pipeline view using DevCompile and QUnitTest



The screenshot shows the Jenkins web interface. At the top, the Jenkins logo and a search bar are visible. Below the header, the breadcrumb navigation shows 'Jenkins > All'. The main content area is titled 'Enter an item name'. A text input field contains 'DevCompile', with a small note below it stating '» Required field'. Below the input field, there are two project type options: 'Freestyle project' and 'Maven project'. The 'Freestyle project' option is selected, indicated by a blue icon and a description: '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.' The 'Maven project' option is also visible with its description: 'Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.' At the bottom of the form, there is an 'OK' button.

Jenkins > All >

### Enter an item name

DevCompile  
» 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.

OK

Jenkins > DevCompile >

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

### Source Code Management

☐ None  
☒ Git

**Repositories**

Repository URL:  ?

Credentials:  Add ?

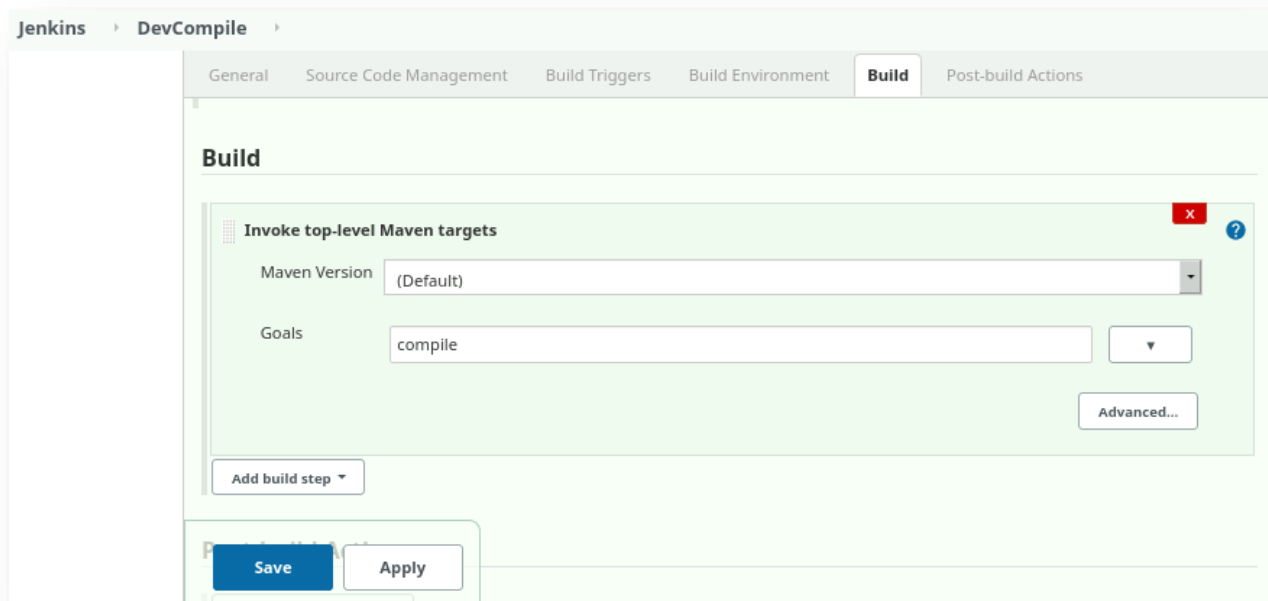
Advanced...  
Add Repository

**Branches to build**

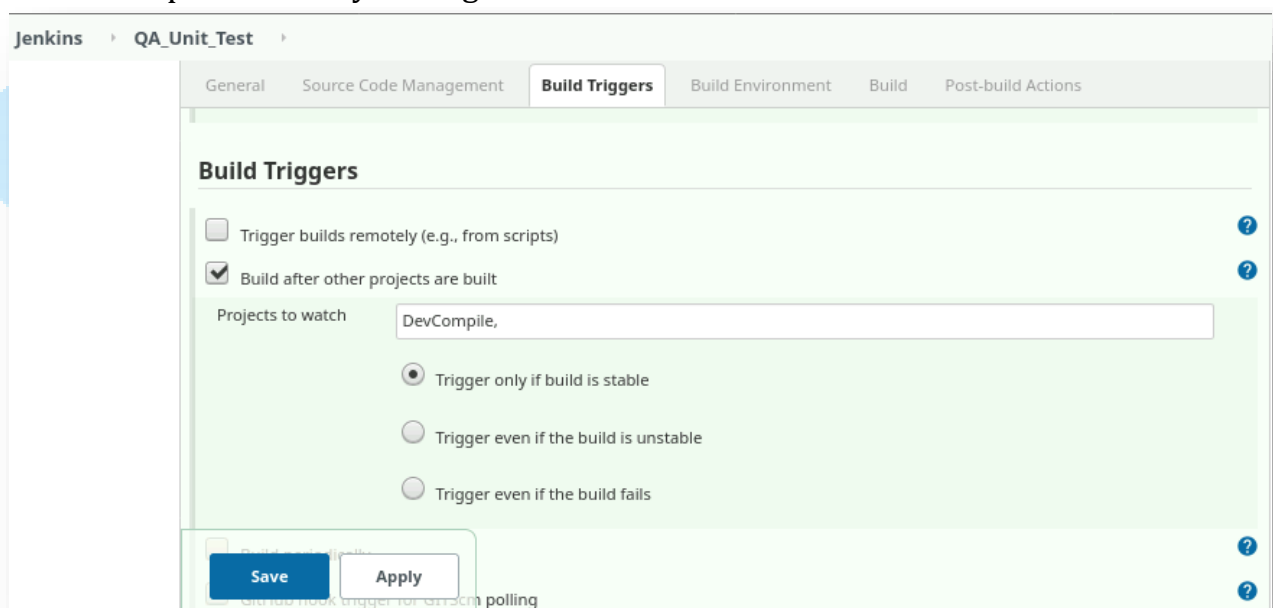
Branch Specifier (blank for 'any'):  ?

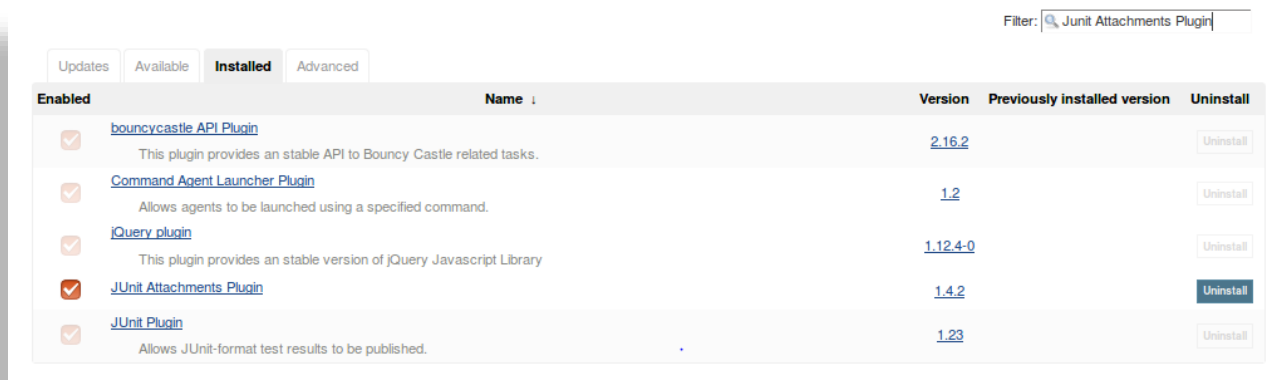
Add Branch

Save Apply (Auto) ?

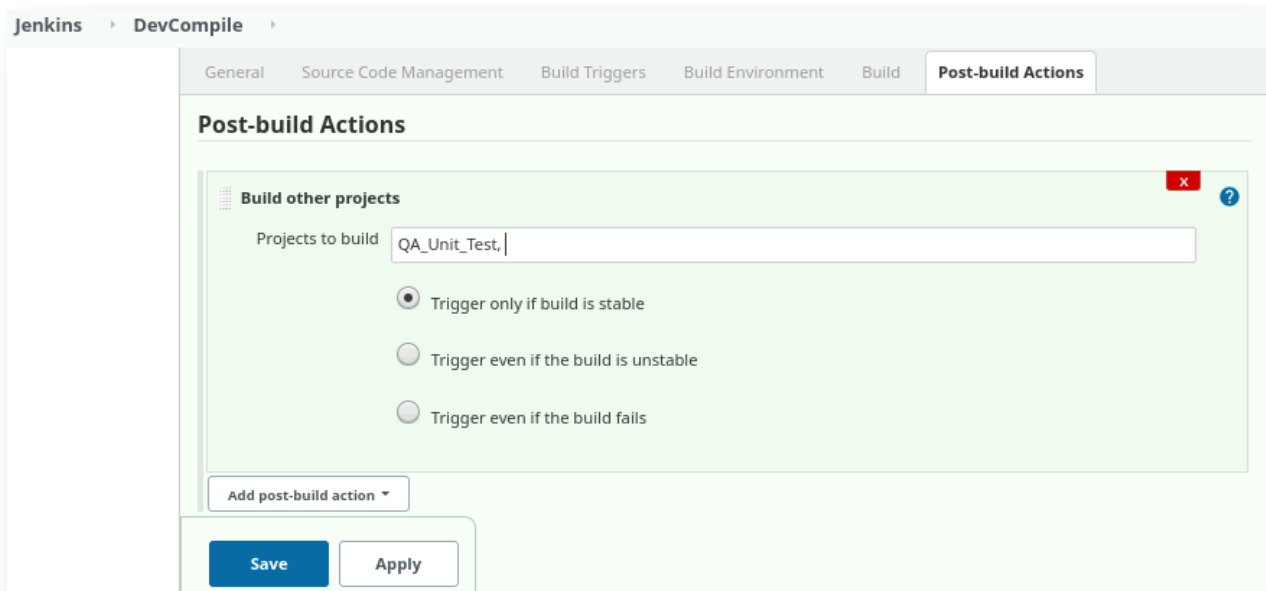


- Create QAunitTest project and provide the same SCM path and execute it after DevCompile Action by setting build.

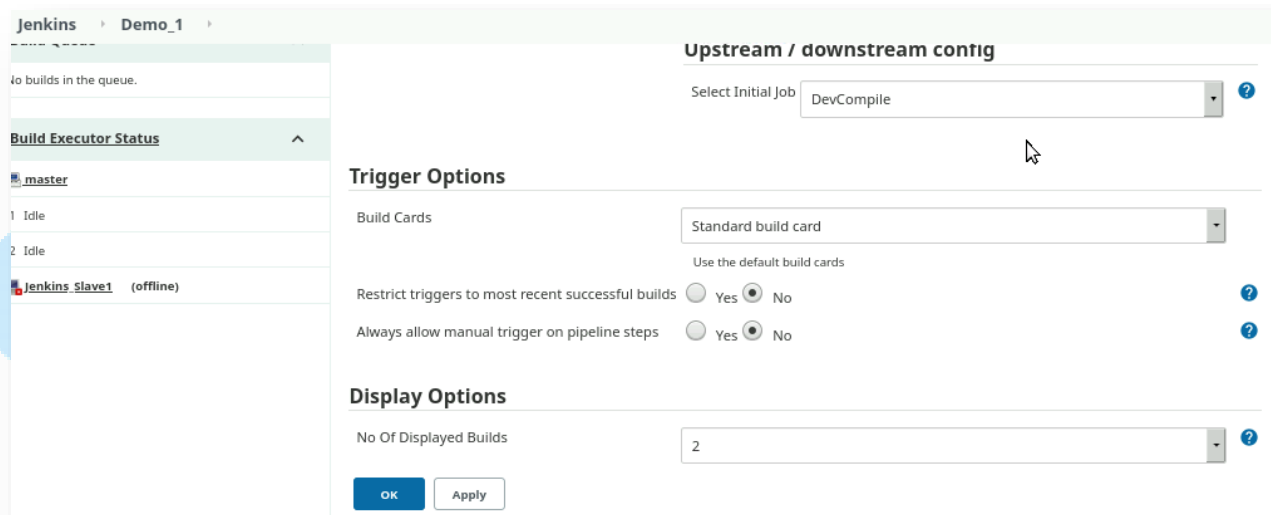
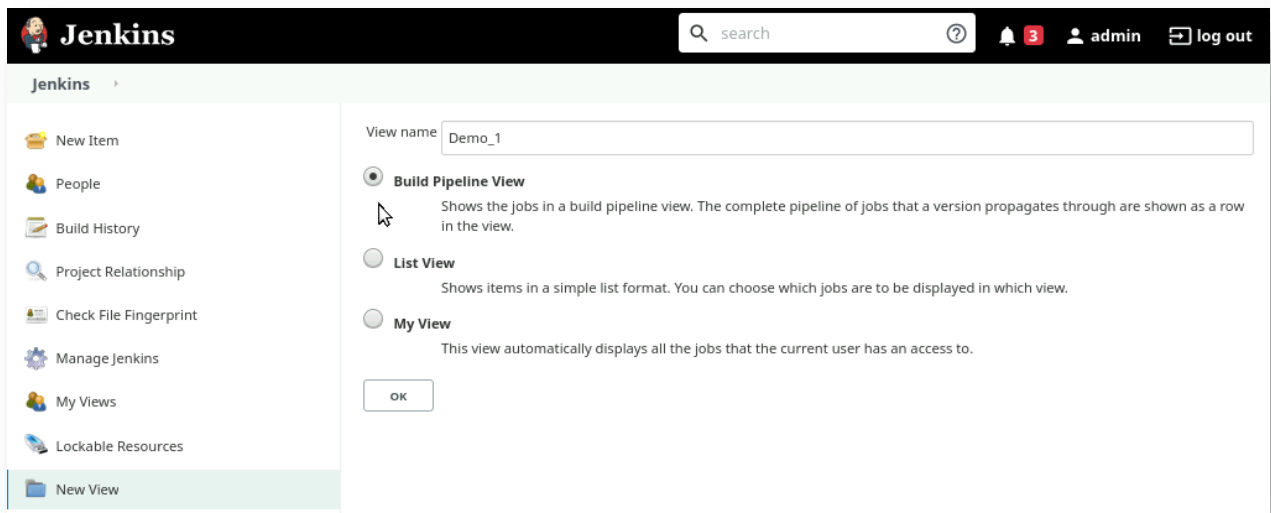


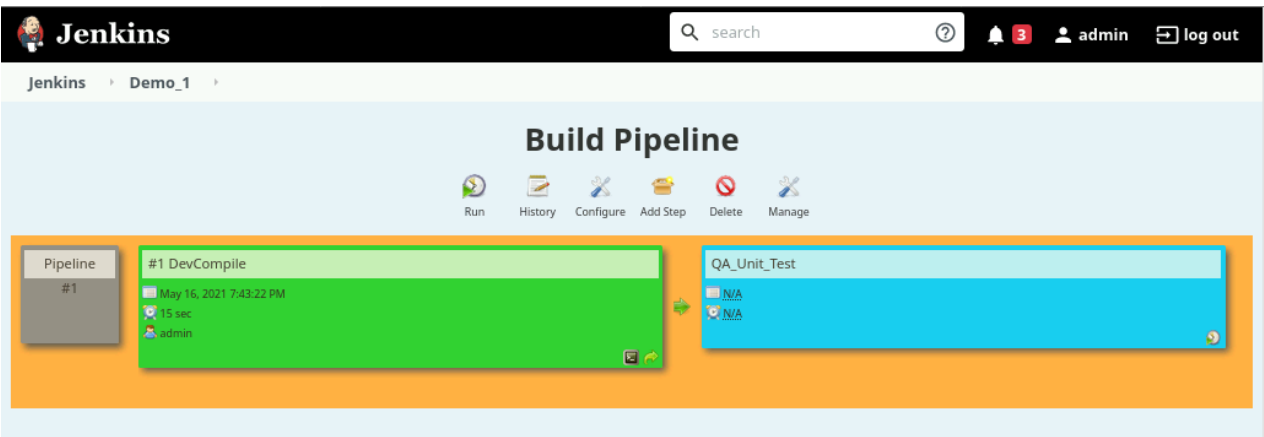
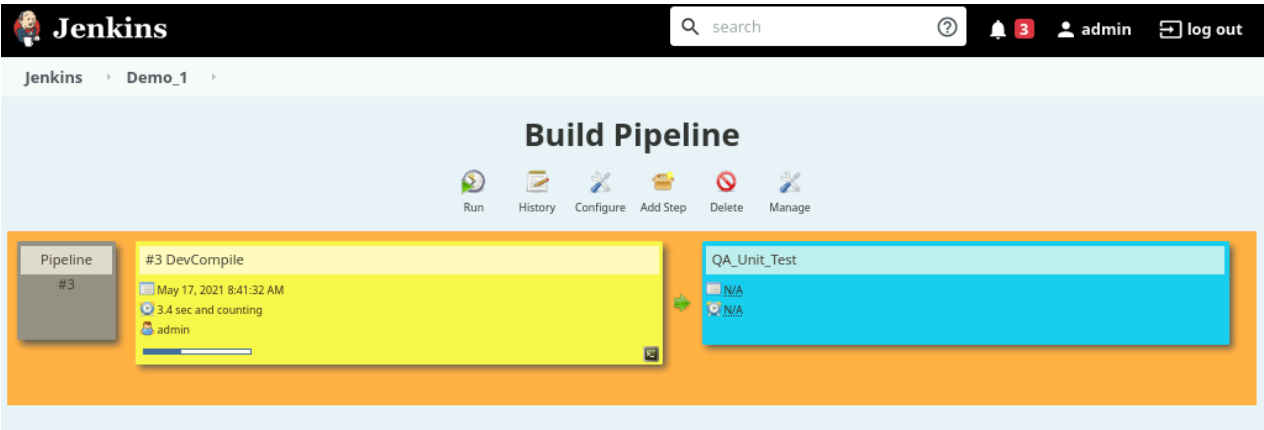


- Configure DevCompile project and set post-build action as QAUnitTest



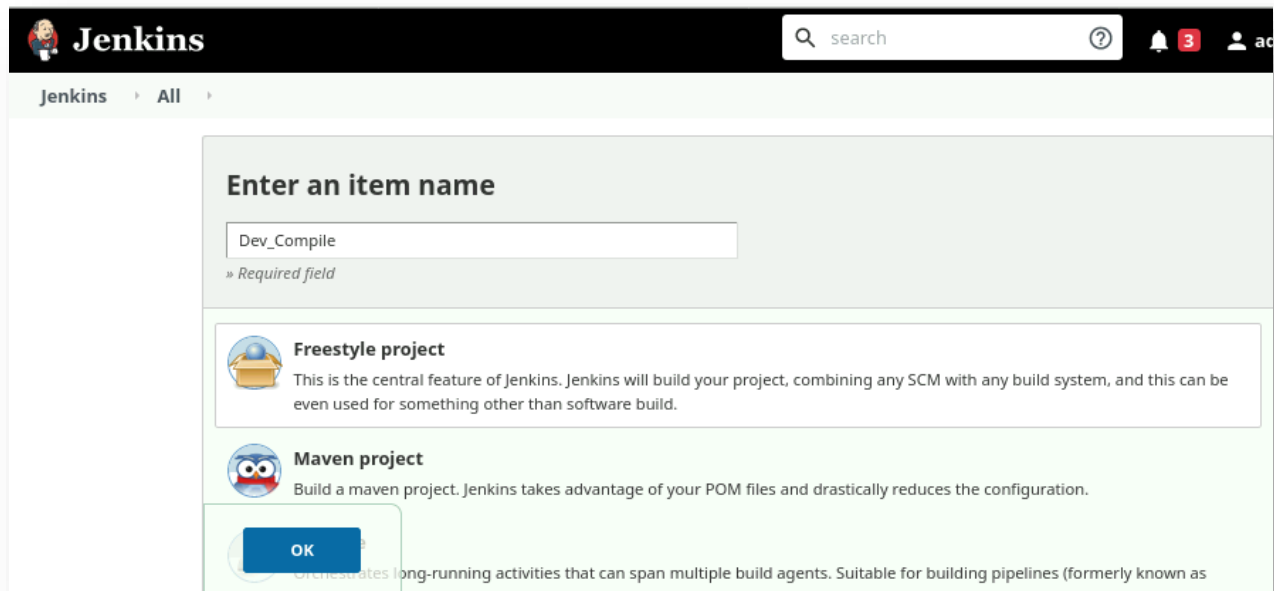
- Go to Jenkins Dashboard and click on the “+” button. That button is for adding a view.
- You will be redirected to the following screen.
- Give the View name and Select the build Pipeline View radio button, and press OK.



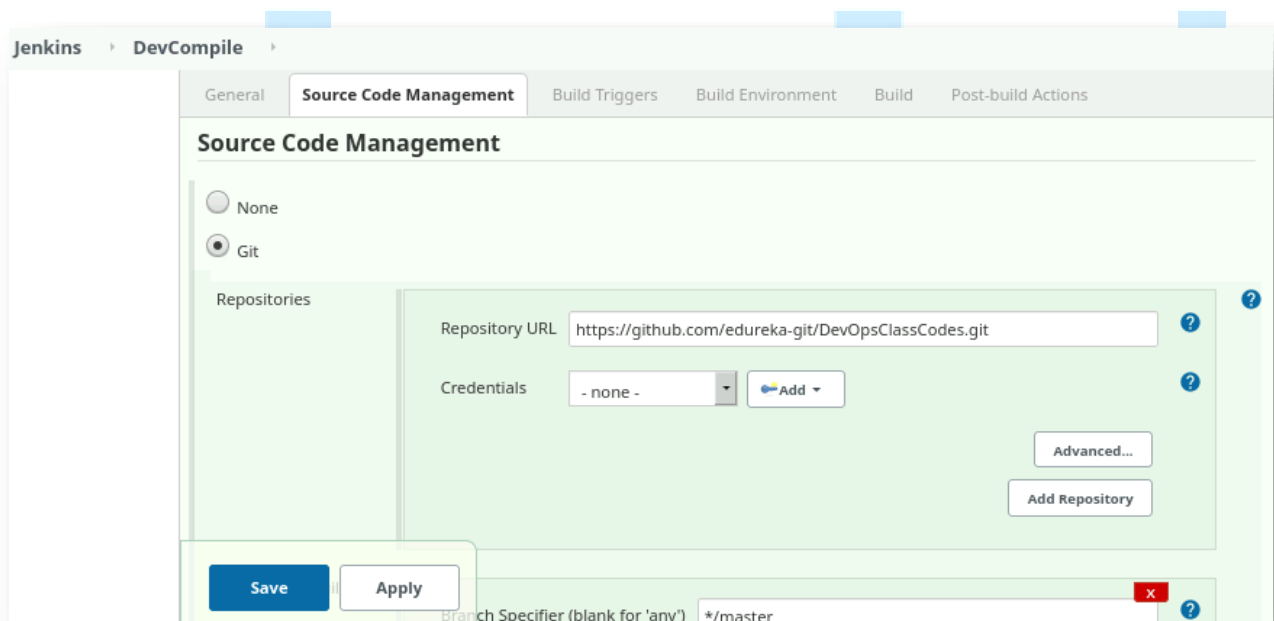


ccatena

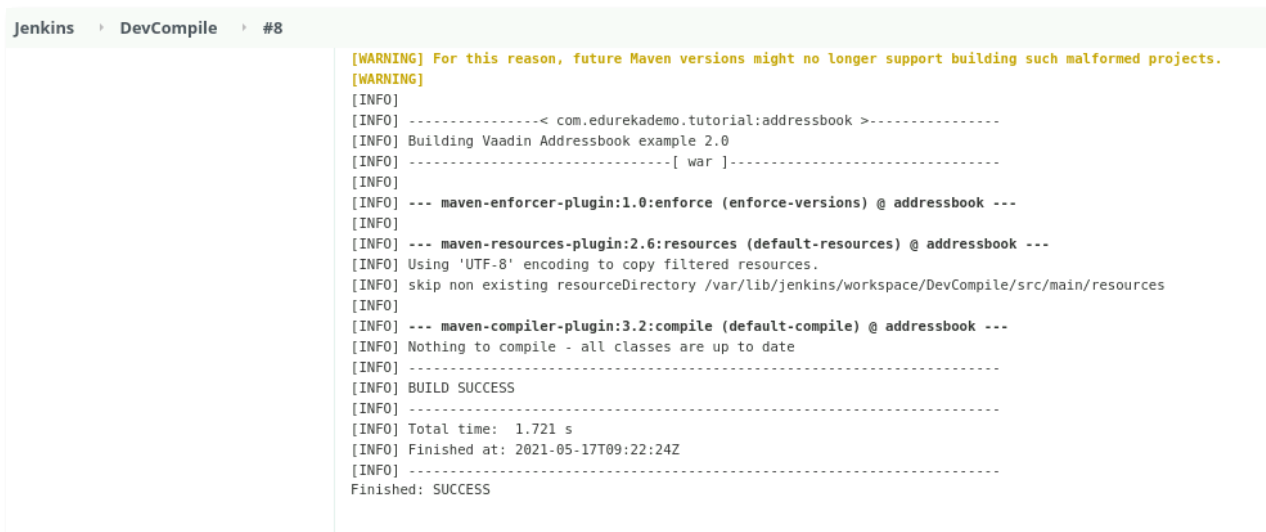
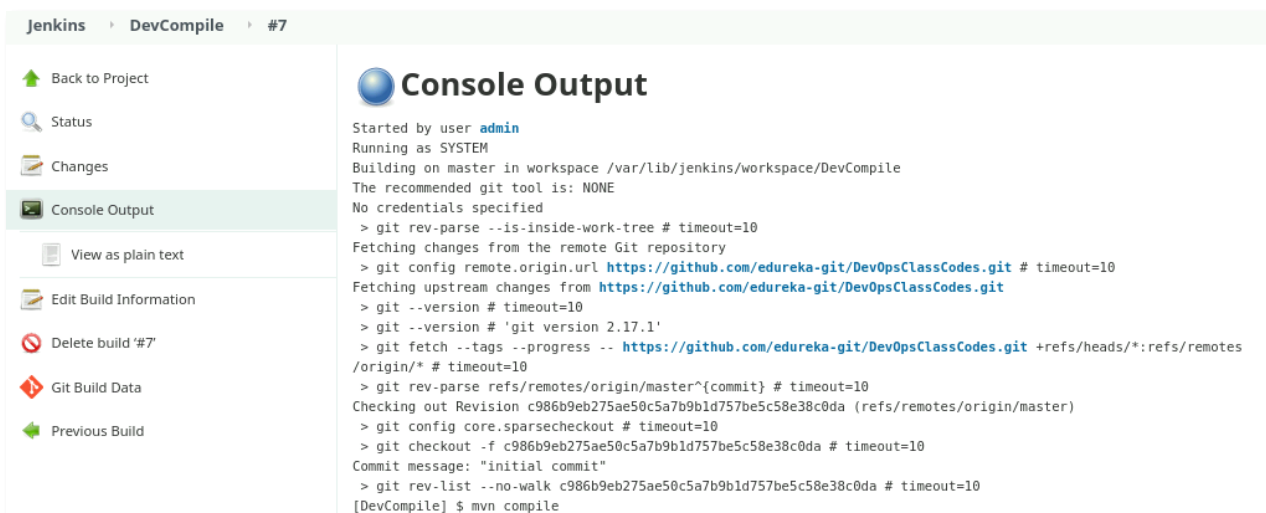
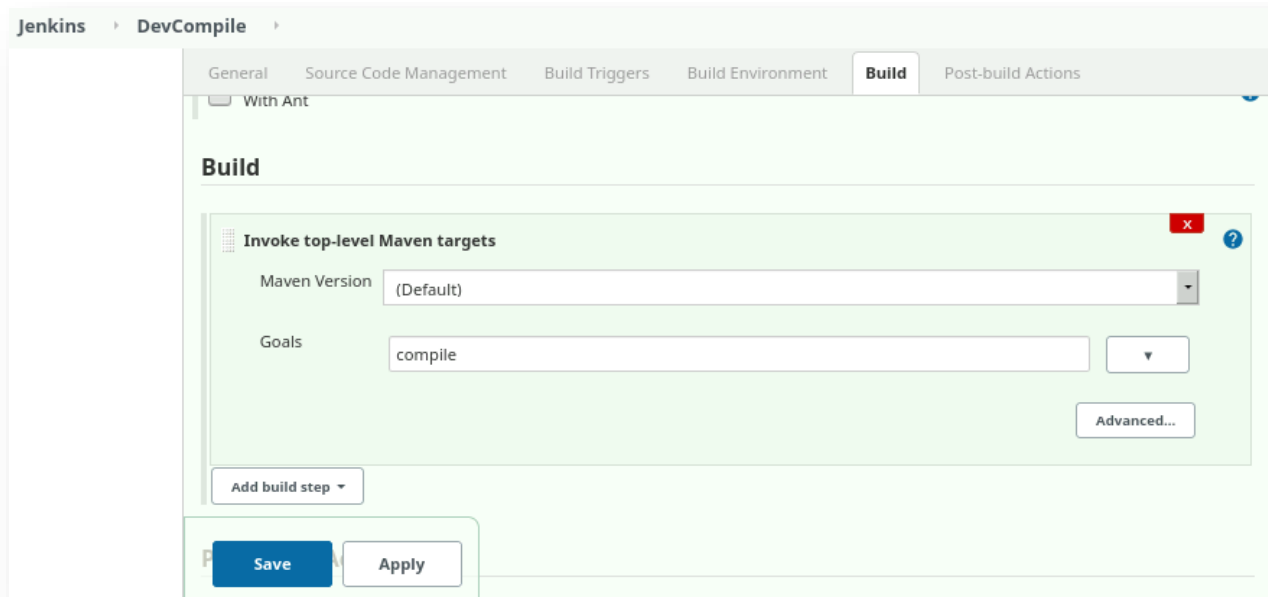
## Demo – 2 : Create The Whole Delivery Pipeline View



The image shows the Jenkins 'Enter an item name' dialog box. At the top, there's a search bar and a user profile icon. Below the header, the text 'Jenkins > All' is visible. The main section is titled 'Enter an item name'. There is a text input field containing 'Dev\_Compile' and a 'Required field' message below it. Below the input field, there are two project type options: 'Freestyle project' and 'Maven project'. The 'Freestyle project' option is selected and highlighted with a blue border. It includes a description: '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.' The 'Maven project' option is also visible with its description: 'Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.' At the bottom, there is an 'OK' button.

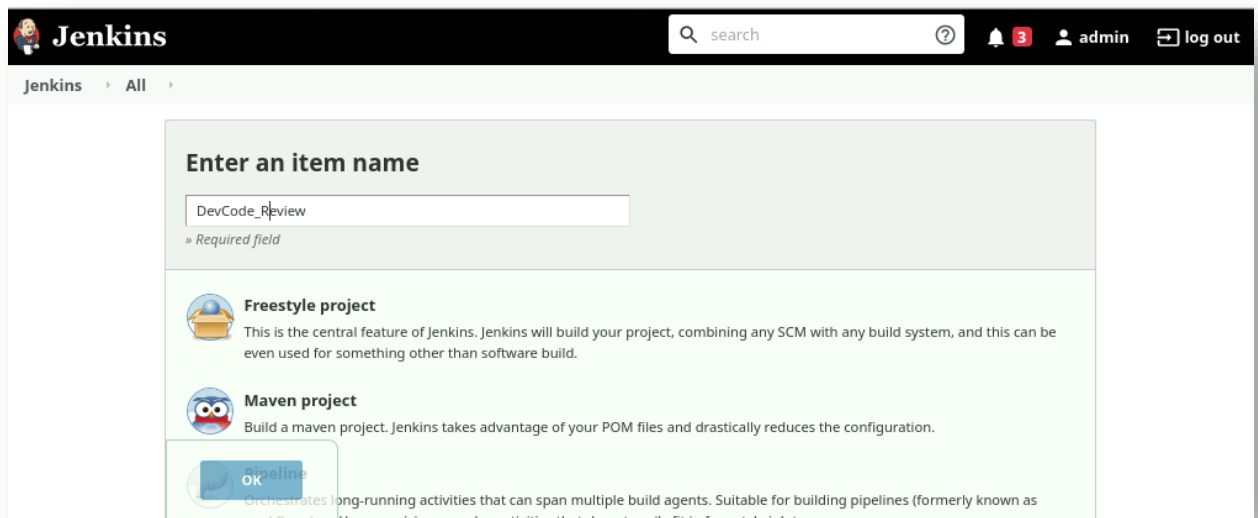


The image shows the Jenkins 'Source Code Management' configuration page for a project named 'DevCompile'. The page has tabs for 'General', 'Source Code Management', 'Build Triggers', 'Build Environment', 'Build', and 'Post-build Actions'. The 'Source Code Management' tab is active. Under the 'Source Code Management' heading, there are radio buttons for 'None' and 'Git'. The 'Git' option is selected. Below this, there is a 'Repositories' section. It contains a 'Repository URL' field with the value 'https://github.com/edureka-git/DevOpsClassCodes.git', a 'Credentials' dropdown menu set to '- none -', and an 'Add' button. There are also 'Advanced...' and 'Add Repository' buttons. At the bottom, there is a 'Branch Specifier (blank for 'any')' field with the value '\*/master'. There are 'Save' and 'Apply' buttons at the bottom left.



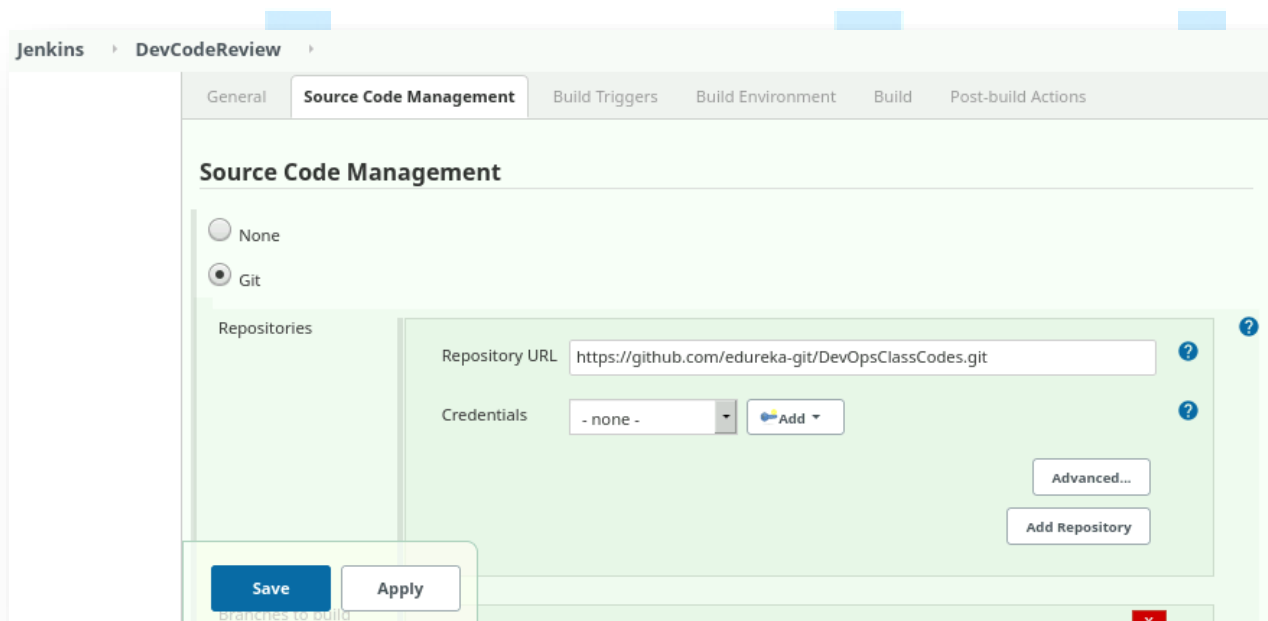


- Now, create a new Project DevCodeReview

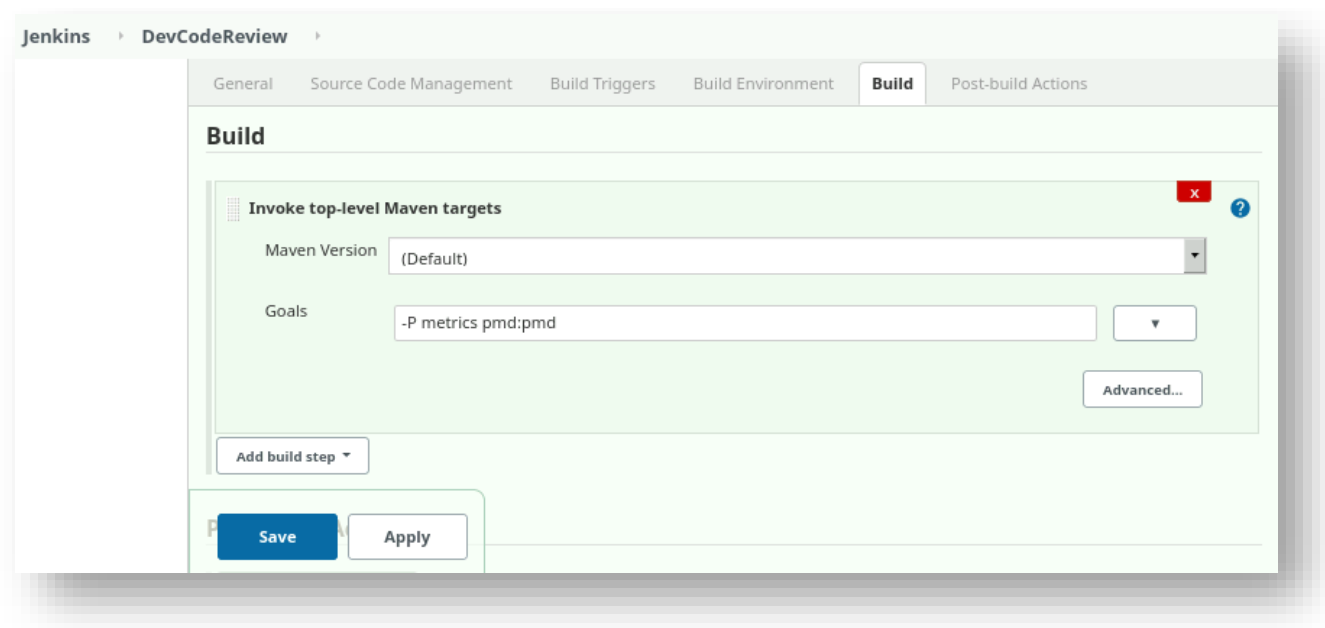
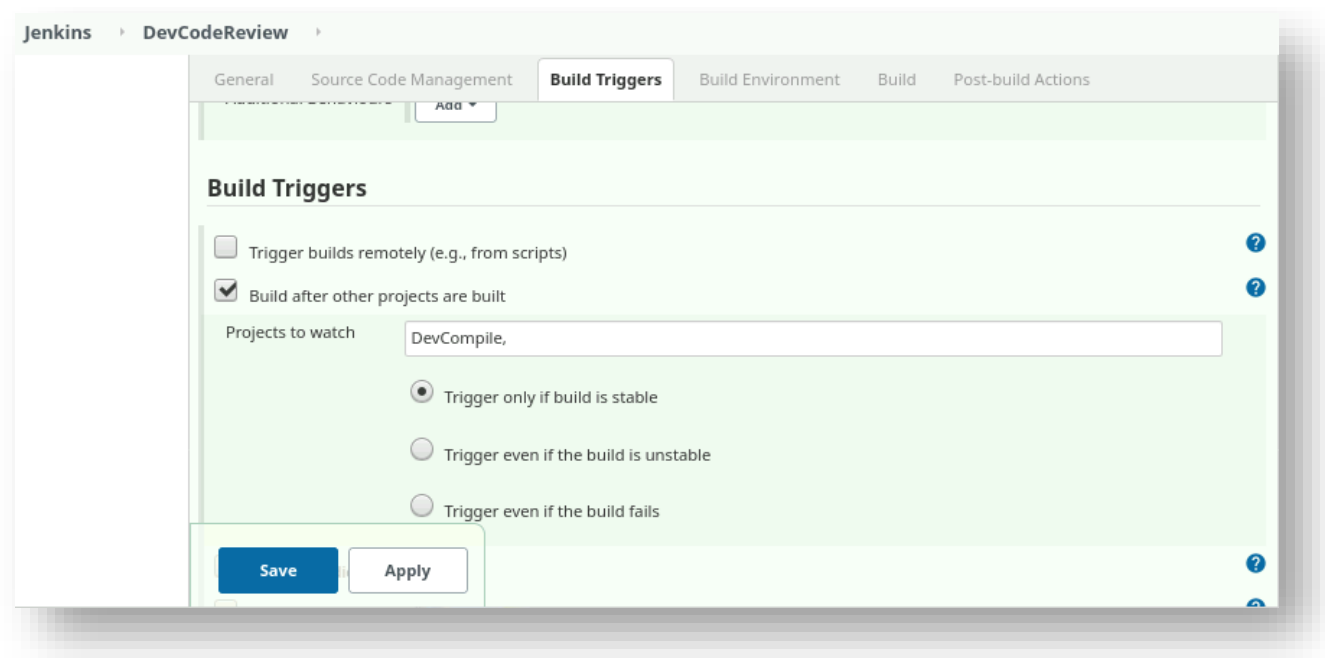


The screenshot shows the Jenkins 'Enter an item name' form. The header includes the Jenkins logo, a search bar, and user information (admin, log out). The main form area has a text input field containing 'DevCode\_Review' with a 'Required field' message below it. Below the input field, there are three project type options: 'Freestyle project' (described as the central feature of Jenkins), 'Maven project' (described as building a maven project), and 'Pipeline' (described as orchestrating long-running activities). The 'Pipeline' option is highlighted with a blue border and an 'OK' button.

- Provide the same Source Code Management Path to DevCodereview



The screenshot shows the Jenkins 'Source Code Management' configuration page for a project named 'DevCodeReview'. The page has tabs for 'General', 'Source Code Management' (selected), 'Build Triggers', 'Build Environment', 'Build', and 'Post-build Actions'. Under 'Source Code Management', there are radio buttons for 'None' and 'Git' (selected). Below this, there is a 'Repositories' section with a 'Repository URL' field containing 'https://github.com/edureka-git/DevOpsClassCodes.git' and a 'Credentials' dropdown menu set to '- none -'. There are 'Advanced...' and 'Add Repository' buttons. At the bottom, there are 'Save' and 'Apply' buttons. A 'Branches to build' section is partially visible at the bottom left.



## Module 4: Continuous Integration With Jenkins

Filter:

Updates	Available	Installed	Advanced		
Enabled					
		Name	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>		<a href="#">bouncycastle API Plugin</a>	2.16.2		<a href="#">Uninstall</a>
		This plugin provides an stable API to Bouncy Castle related tasks.			
<input checked="" type="checkbox"/>		<a href="#">Command Agent Launcher Plugin</a>	1.2		<a href="#">Uninstall</a>
		Allows agents to be launched using a specified command.			
<input checked="" type="checkbox"/>		<a href="#">Matrix Project Plugin</a>	1.12		<a href="#">Uninstall</a>
		Multi-configuration (matrix) project type.			
<input checked="" type="checkbox"/>		<a href="#">Maven Integration plugin</a>	3.1		<a href="#">Uninstall</a>
		This plug-in provides, for better and for worse, a deep integration of Jenkins and Maven: Automatic triggers between projects depending on SNAPSHOTS, automated configuration of various Jenkins publishers (JUnit, ...).			
<input checked="" type="checkbox"/>		<a href="#">PMD Plug-in</a>	3.50		<a href="#">Uninstall</a>
		This plug-in collects the <a href="#">PMD</a> analysis results of the project modules and visualizes the found warnings.			
<input checked="" type="checkbox"/>		<a href="#">Static Analysis Utilities</a>	1.94		<a href="#">Uninstall</a>
		This plug-in provides utilities for the static code analysis plug-ins.			
<input checked="" type="checkbox"/>		<a href="#">Token Macro Plugin</a>	2.3		<a href="#">Uninstall</a>
		This plug-in adds reusable macro expansion capability for other plug-ins to use.			

Jenkins > DevCodeReview > #2

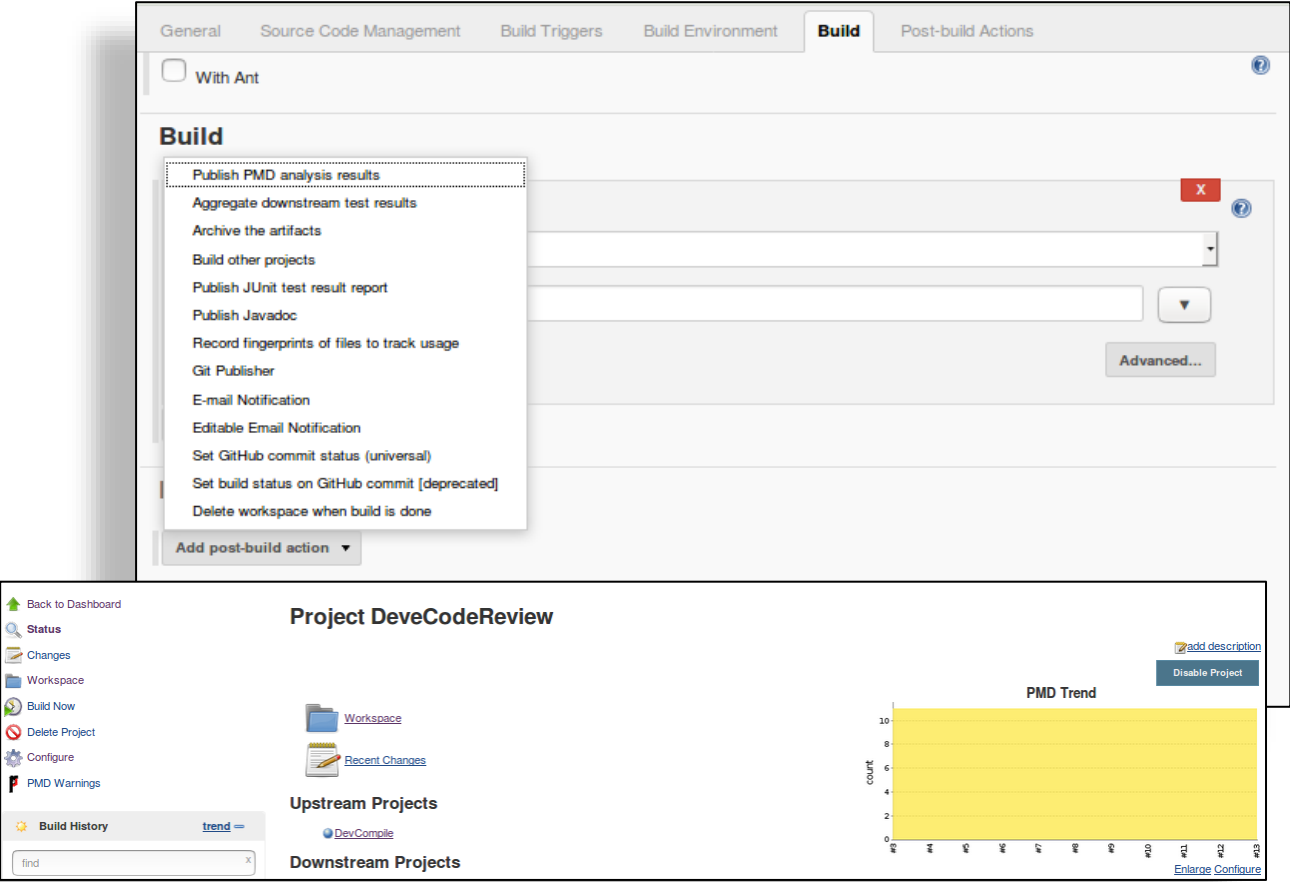
[Back to Project](#)  
[Status](#)  
[Changes](#)  
[Console Output](#)  
[View as plain text](#)  
[Edit Build Information](#)  
[Delete build '#2'](#)  
[Git Build Data](#)  
[Previous Build](#)

## Console Output

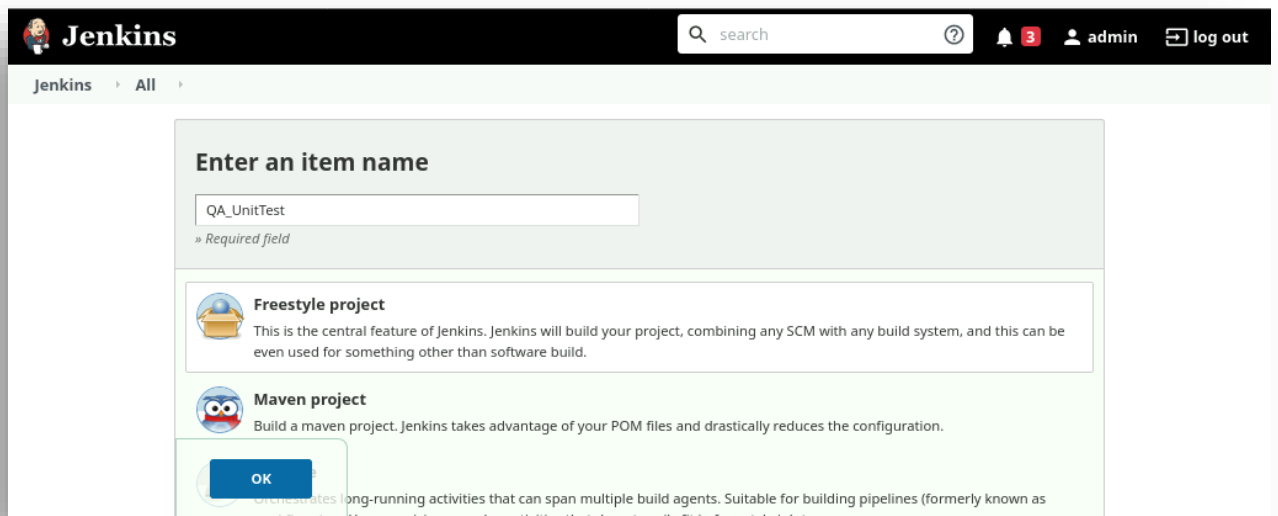
Started by user [admin](#)  
Running as SYSTEM  
Building on master in workspace /var/lib/jenkins/workspace/DevCodeReview  
The recommended git tool is: NONE  
No credentials specified  
> git rev-parse --is-inside-work-tree # timeout=10  
Fetching changes from the remote Git repository  
> git config remote.origin.url <https://github.com/edureka-git/DevOpsClassCodes.git> # timeout=10  
Fetching upstream changes from <https://github.com/edureka-git/DevOpsClassCodes.git>  
> git --version # timeout=10  
> git --version # 'git version 2.17.1'  
> git fetch --tags --progress -- <https://github.com/edureka-git/DevOpsClassCodes.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10  
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10  
Checking out Revision c986b9eb275ae50c5a7b9b1d757be5c58e38c0da (refs/remotes/origin/master)  
> git config core.sparsecheckout # timeout=10  
> git checkout -f c986b9eb275ae50c5a7b9b1d757be5c58e38c0da # timeout=10

Jenkins > DevCodeReview > #2

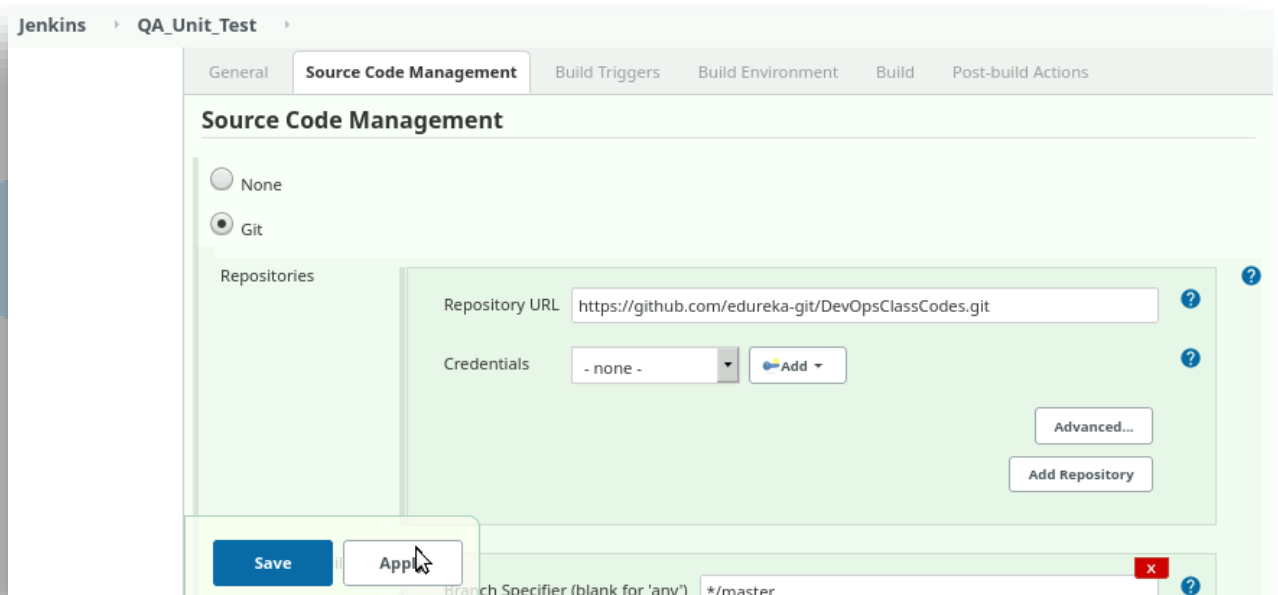
```
Commit message: "initial commit"
> git rev-list --no-walk c986b9eb275ae50c5a7b9b1d757be5c58e38c0da # timeout=10
[DevCodeReview] $ mvn -P metrics pmd:pmd
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.edurekademo.tutorial:addressbook >-----
[INFO] Building Vaadin Addressbook example 2.0
[INFO] -----[ war ]-----
[INFO]
[INFO] --- maven-pmd-plugin:3.2:pmd (default-cli) @ addressbook ---
[WARNING] Unable to locate Source XRef to link to - DISABLED
[WARNING] Unable to locate Source XRef to link to - DISABLED
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.024 s
[INFO] Finished at: 2021-05-17T09:33:58Z
[INFO] -----
Finished: SUCCESS
```



- Now, create a New project QaUnitTest



The screenshot shows the Jenkins 'Enter an item name' dialog. At the top, there's a search bar and user information (admin, log out). Below the title 'Enter an item name', there's a text input field containing 'QA\_UnitTest' with a 'Required field' note. Underneath, there are two project type options: 'Freestyle project' (described as the central feature of Jenkins) and 'Maven project' (described as a build system for Maven projects). An 'OK' button is visible at the bottom left of the dialog.



The screenshot shows the Jenkins 'Source Code Management' configuration page for a project named 'QA\_Unit\_Test'. The page has tabs for 'General', 'Source Code Management' (active), 'Build Triggers', 'Build Environment', 'Build', and 'Post-build Actions'. Under 'Source Code Management', there are radio buttons for 'None' and 'Git' (selected). Below this, there's a 'Repositories' section with a 'Repository URL' field containing 'https://github.com/edureka-git/DevOpsClassCodes.git' and a 'Credentials' dropdown set to '- none -'. There are 'Advanced...' and 'Add Repository' buttons. At the bottom, there's a 'Branch Specifier (blank for 'any')' field with the value '\*/master'. A 'Save' button and an 'Apply' button (with a mouse cursor) are at the bottom left.

Jenkins > QA\_Unit\_Test >

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

### Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?

☒ Build after other projects are built ?

Projects to watch

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

☐ Build periodically ?

Jenkins > QA\_Unit\_Test >

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

☐ With Ant ?

### Build

☒ Invoke top-level Maven targets ?

Maven Version

Goals

Jenkins > QA\_Unit\_Test > #9

[Back to Project](#)  
[Status](#)  
[Changes](#)  
**Console Output**  
[View as plain text](#)  
[Edit Build Information](#)  
[Delete build '#9'](#)  
[Git Build Data](#)  
[Previous Build](#)

## Console Output

Started by user [admin](#)  
Running as SYSTEM  
Building on master in workspace /var/lib/jenkins/workspace/QA\_Unit\_Test  
The recommended git tool is: NONE  
No credentials specified  
> git rev-parse --is-inside-work-tree # timeout=10  
Fetching changes from the remote Git repository  
> git config remote.origin.url <https://github.com/edureka-git/DevOpsClassCodes.git> # timeout=10  
Fetching upstream changes from <https://github.com/edureka-git/DevOpsClassCodes.git>  
> git --version # timeout=10  
> git --version # 'git version 2.17.1'  
> git fetch --tags --progress -- <https://github.com/edureka-git/DevOpsClassCodes.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10  
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10  
Checking out Revision c986b9eb275ae50c5a7b9b1d757be5c58e38c0da (refs/remotes/origin/master)  
> git config core.sparsecheckout # timeout=10  
> git checkout -f c986b9eb275ae50c5a7b9b1d757be5c58e38c0da # timeout=10  
Commit message: "initial commit"  
First time build. Skipping changelog.  
[QA\_Unit\_Test] \$ /var/lib/jenkins/tools/hudson.tasks.Maven\_MavenInstallation/My\_Maven/bin/mvn test

Jenkins > QA\_Unit\_Test > #9

Testing using junit runner into maven

Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.037 sec - in com.edurekademo.utilities.TestHexAsciiConversion  
Running com.edurekademo.utilities.TestGenericComparator  
Tests run: 14, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.031 sec - in com.edurekademo.utilities.TestGenericComparator

Results :

Tests run: 23, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----  
[INFO] BUILD SUCCESS  
[INFO] -----  
[INFO] Total time: 3.371 s  
[INFO] Finished at: 2021-05-17T09:42:18Z  
[INFO] -----  
Finished: SUCCESS

[Back to Project](#)  
[Status](#)  
[Changes](#)  
[Console Output](#)  
[Edit Build Information](#)  
[History](#)  
[Git Build Data](#)  
[No Tags](#)  
**Test Result**  
[Previous Build](#)

## Test Result

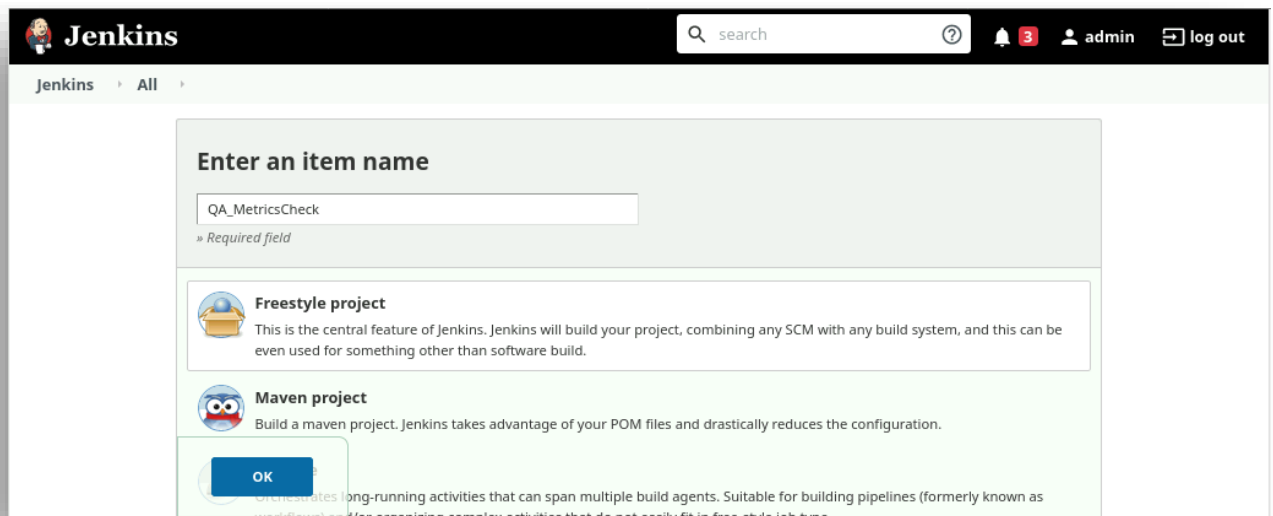
0 failures

23 tests  
Took 0.38 sec.  
[add description](#)

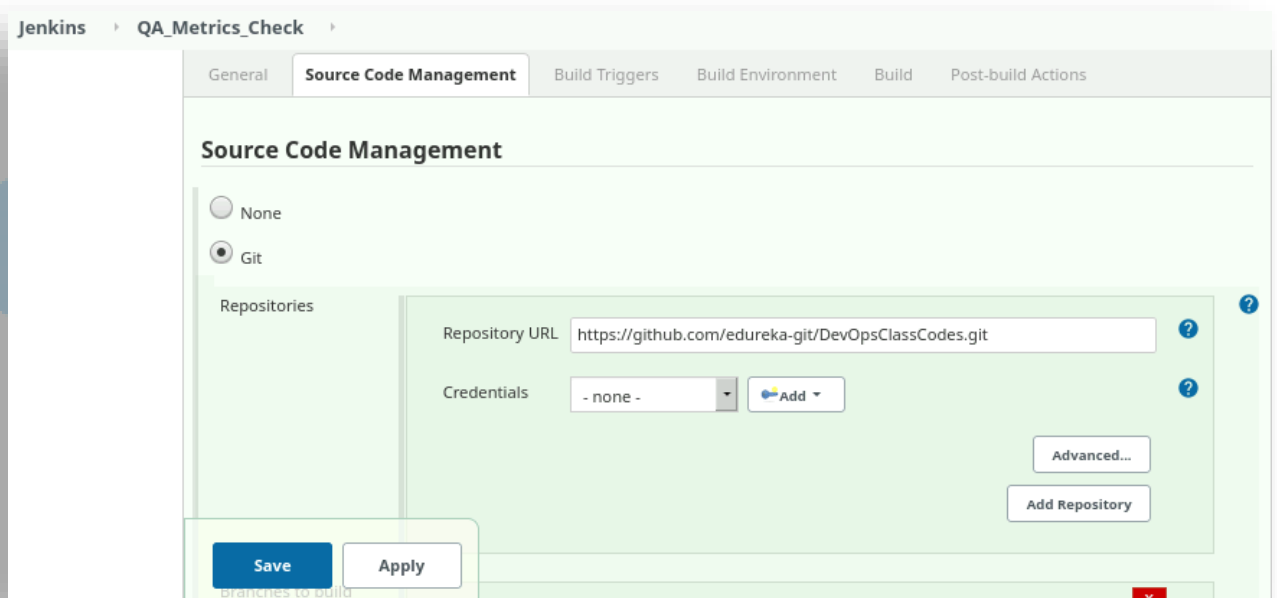
### All Tests

Package	Duration	Fail (d/f)	Skip (d/f)	Pass (d/f)	Total (d/f)
<a href="#">com.edurekademo.utilities</a>	0.38 sec	0	0	23 +23	23 +23

- Now, create a New project QAMerticCheck



The screenshot shows the Jenkins 'Enter an item name' configuration page. The top navigation bar includes the Jenkins logo, a search bar, and user information (admin, log out). The breadcrumb trail shows 'Jenkins > All >'. The main section is titled 'Enter an item name' and contains a text input field with the value 'QA\_MetricsCheck'. Below the input field is a 'Required field' message. There are three project type options: 'Freestyle project' (described as the central feature of Jenkins), 'Maven project' (described as a build system for Maven), and 'Pipeline' (described as long-running activities that can span multiple build agents). A blue 'OK' button is visible at the bottom left of the project type selection area.



The screenshot shows the 'Source Code Management' configuration page for the 'QA\_Metrics\_Check' project. The top navigation bar includes the Jenkins logo and the breadcrumb trail 'Jenkins > QA\_Metrics\_Check >'. The page has tabs for 'General', 'Source Code Management' (selected), 'Build Triggers', 'Build Environment', 'Build', and 'Post-build Actions'. The 'Source Code Management' section has two radio buttons: 'None' and 'Git' (selected). Below the radio buttons is a 'Repositories' section. It contains a 'Repository URL' input field with the value 'https://github.com/edureka-git/DevOpsClassCodes.git' and a 'Credentials' dropdown menu with the value '- none -'. There are 'Advanced...' and 'Add Repository' buttons. At the bottom, there are 'Save' and 'Apply' buttons. A red 'x' icon is visible in the bottom right corner.



Jenkins > QA\_Metrics\_Check >

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

### Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?

☒ Build after other projects are built ?

Projects to watch

☒ Trigger only if build is stable

☐ Trigger even if the build is unstable

☐ Trigger even if the build fails

?

Jenkins > QA\_Metrics\_Check >

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

### Build

**Invoke top-level Maven targets** x ?

Maven Version

Goals  ▼

Updates Available Installed Advanced					
Enabled	Name	Version	Previously installed version	Uninstall	
<input checked="" type="checkbox"/>	<a href="#">bouncycastle API Plugin</a> This plugin provides an stable API to Bouncy Castle related tasks.	2.16.2		<input type="button" value="Uninstall"/>	
<input checked="" type="checkbox"/>	<a href="#">Cobertura Plugin</a> This plugin integrates <a href="#">Cobertura coverage reports</a> to Jenkins.	1.12		<input type="button" value="Uninstall"/>	
<input checked="" type="checkbox"/>	<a href="#">Command Agent Launcher Plugin</a> Allows agents to be launched using a specified command.	1.2		<input type="button" value="Uninstall"/>	
<input checked="" type="checkbox"/>	<a href="#">Structs Plugin</a> Library plugin for DSL plugins that need names for Jenkins objects.	1.10		<input type="button" value="Uninstall"/>	

**Jenkins**   **QA\_Metrics\_Check**   #2

**Console Output**

```

Started by user admin
Running as SYSTEM
Building on master in workspace /var/lib/jenkins/workspace/QA_Metrics_Check
The recommended git tool is: NONE
No credentials specified
> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/edureka-git/DevOpsClassCodes.git # timeout=10
Fetching upstream changes from https://github.com/edureka-git/DevOpsClassCodes.git
> git --version # timeout=10
> git --version # 'git version 2.17.1'
> git fetch --tags --progress -- https://github.com/edureka-git/DevOpsClassCodes.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision c986b9eb275ae50c5a7b9b1d757be5c58e38c0da (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f c986b9eb275ae50c5a7b9b1d757be5c58e38c0da # timeout=10
  
```

**Jenkins**   **QA\_Metrics\_Check**   #2

```

net.sourceforge.cobertura.reporting.ComplexityCalculator.getCCNForPackageInternal(ComplexityCalculator.java:208)
at net.sourceforge.cobertura.reporting.ComplexityCalculator.getCCNForProject(ComplexityCalculator.java:174)
at net.sourceforge.cobertura.reporting.html.HTMLReport.generateTableRowForTotal(HTMLReport.java:606)
at net.sourceforge.cobertura.reporting.html.HTMLReport.generateOverview(HTMLReport.java:290)
at net.sourceforge.cobertura.reporting.html.HTMLReport.generateOverviews(HTMLReport.java:231)
at net.sourceforge.cobertura.reporting.html.HTMLReport.<init>(HTMLReport.java:74)
at net.sourceforge.cobertura.reporting.html.HTMLReportFormatStrategy.save(HTMLReportFormatStrategy.java:18)
at net.sourceforge.cobertura.reporting.NativeReport.export(NativeReport.java:31)
at net.sourceforge.cobertura.reporting.CompositeReport.export(CompositeReport.java:19)
at net.sourceforge.cobertura.reporting.ReportMain.parseArgumentsAndReport(ReportMain.java:91)
at net.sourceforge.cobertura.reporting.ReportMain.generateReport(ReportMain.java:141)
at net.sourceforge.cobertura.reporting.ReportMain.main(ReportMain.java:151)

[INFO] Cobertura Report generation was successful.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 8.819 s
[INFO] Finished at: 2021-05-17T09:49:15Z
[INFO] -----
Finished: SUCCESS
  
```

- Now, create a New project QAPackage

**Jenkins**   **All**

**Enter an item name**

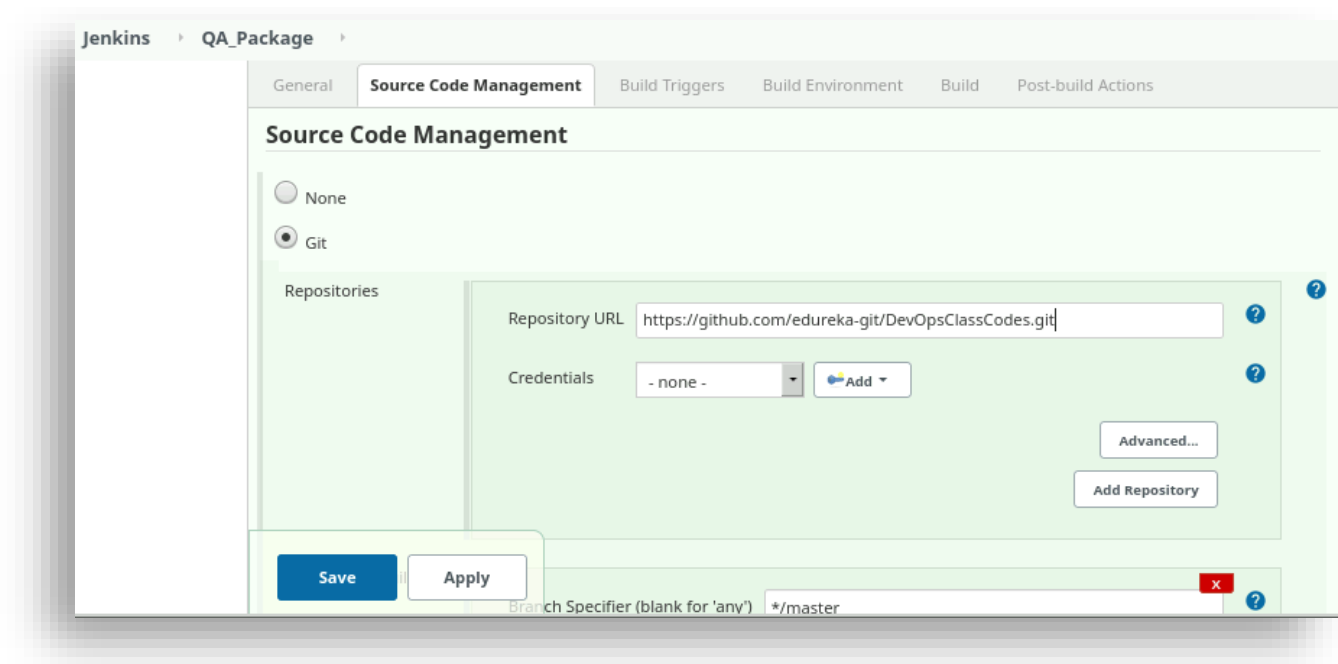
QA Package  
» Required field

**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.

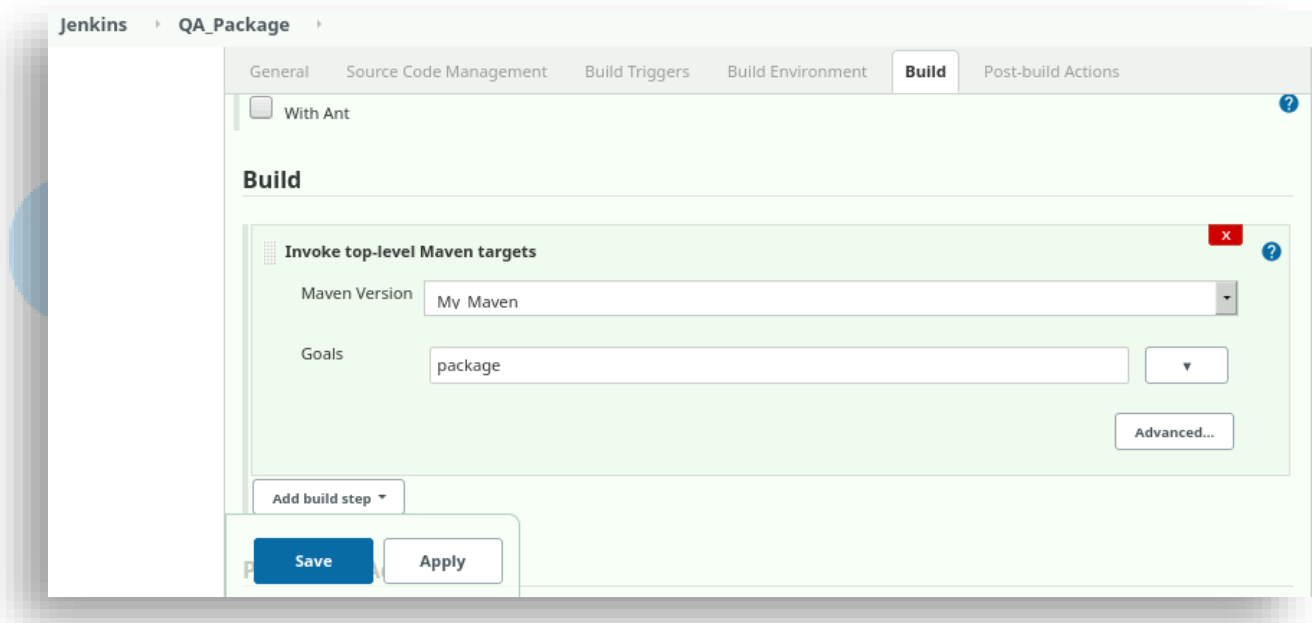
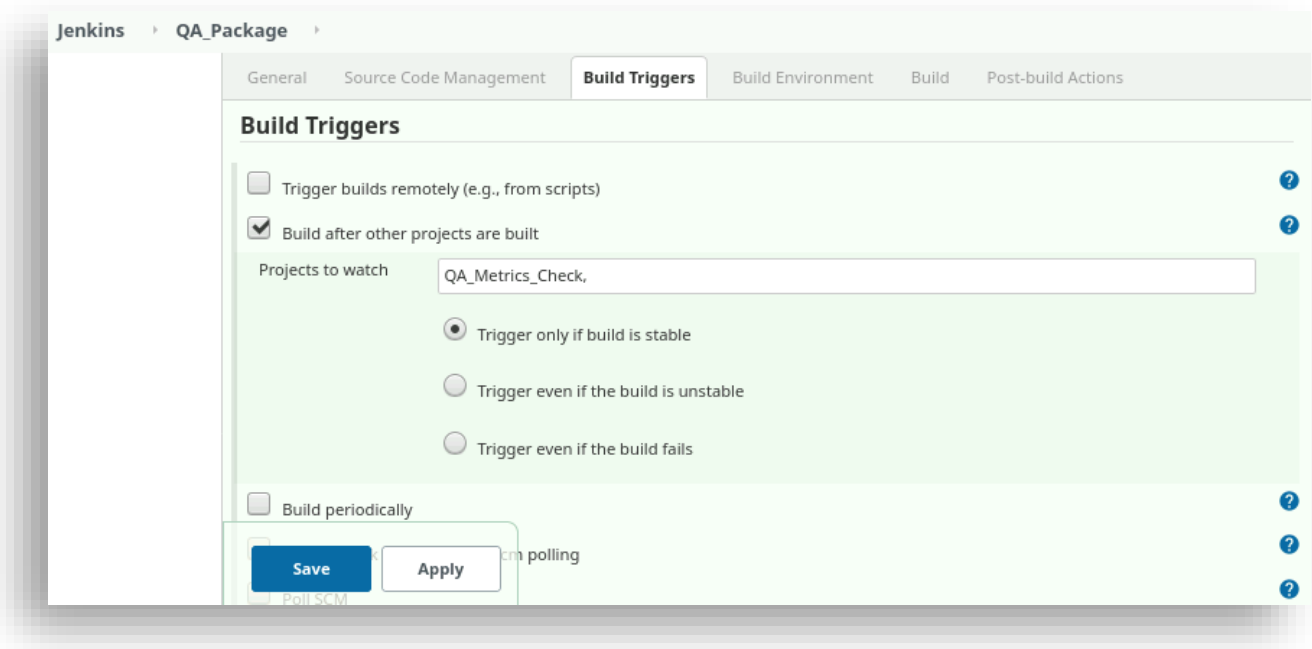
**Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Pipeline project**  
Orchestrate long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) for organizing complex activities that do not easily fit in free-style job type.

**OK**



edureka!



admin
log out

Jenkins > QA\_Package > #2

- Back to Project
- Status
- Changes
- Console Output**
  - View as plain text
- Edit Build Information
- Delete build '#2'
- Git Build Data
- Previous Build

## Console Output

Started by user **admin**  
Running as SYSTEM  
Building on master in workspace /var/lib/jenkins/workspace/QA\_Package  
The recommended git tool is: NONE  
No credentials specified  

```

> git rev-parse --is-inside-work-tree # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/edureka-git/DevOpsClassCodes.git # timeout=10
Fetching upstream changes from https://github.com/edureka-git/DevOpsClassCodes.git
> git --version # timeout=10
> git --version # 'git version 2.17.1'
> git fetch --tags --progress -- https://github.com/edureka-git/DevOpsClassCodes.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision c986b9eb275ae50c5a7b9b1d757be5c58e38c0da (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f c986b9eb275ae50c5a7b9b1d757be5c58e38c0da # timeout=10

```

Jenkins > QA\_Package > #2

```

Running com.edurekademo.utilities.TestGenericComparator
Tests run: 14, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.094 sec - in
com.edurekademo.utilities.TestGenericComparator

Results :

Tests run: 23, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-war-plugin:2.2:war (default-war) @ addressbook ---
[INFO] Packaging webapp
[INFO] Assembling webapp [addressbook] in [/var/lib/jenkins/workspace/QA_Package/target/addressbook]
[INFO] Processing war project
[INFO] Webapp assembled in [341 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/QA_Package/target/addressbook.war
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 5.495 s
[INFO] Finished at: 2021-05-17T09:53:21Z
[INFO] -----
Finished: SUCCESS

```

- Install the Build Pipeline Plugin

Filter:

Updates	Available	Installed	Advanced	
Enabled				
	Name	Version	Previously installed version	Uninstall
<input checked="" type="checkbox"/>	<a href="#">bouncycastle API Plugin</a> This plugin provides an stable API to Bouncy Castle related tasks.	<a href="#">2.16.2</a>		<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<a href="#">Build Pipeline Plugin</a> This plugin renders upstream and downstream connected jobs that typically form a build pipeline. In addition, it offers the ability to define manual triggers for jobs that require intervention prior to execution, e.g. an approval process outside of Jenkins.	<a href="#">1.5.8</a>		<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<a href="#">jQuery plugin</a> This plugin provides an stable version of JQuery Javascript Library	<a href="#">1.12.4-0</a>		<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<a href="#">Parameterized Trigger plugin</a>	<a href="#">2.35.2</a>		<a href="#">Uninstall</a>
<input checked="" type="checkbox"/>	<a href="#">Pipeline: Supporting APIs</a> Common utility implementations to build Pipeline Plugin	<a href="#">2.17</a>		<a href="#">Uninstall</a>

- Go to the Jenkins dashboard and create a Pipeline View

**Jenkins**  ? 3 admin log out

Jenkins >

- New Item
- People
- Build History
- Project Relationship
- Check File Fingerprint
- Manage Jenkins
- My Views
- Lockable Resources
- New View**

View name:

☒ **Build Pipeline View**  
Shows the jobs in a build pipeline view. The complete pipeline of jobs that a version propagates through are shown as a row in the view.

☐ **List View**  
Shows items in a simple list format. You can choose which jobs are to be displayed in which view.

☐ **My View**  
This view automatically displays all the jobs that the current user has an access to.

