



Name	DevOps Pipeline as Code: Java WebApp
URL	https://www.attackdefense.com/challengedetails?cid=2065
Type	Pipeline Basics: Web Applications

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Challenge Description

DevOps practices are to combine software development (Dev) and IT operations (Ops) in order to improve the delivery process. DevOps pipelines are chained tasks and components that run in a sequence to cover different phases of software compilation, packaging, automated testing, and test deployment.

In this lab, we have a simple DevOps pipeline for a sample Java-based web application. The pipeline consists of the following components (and tasks):

- Kali machine (For pulling, modifying, and pushing the code)
- GitLab server (For hosting code)
- Jenkins server (For integrating all parts: building java project using maven, deploying with Ansible, and dynamic testing with Selenium)
- Test server (For test deployment)

Objective: Run the pipeline and observe/understand the DevOps process!

Instructions:

- The GitLab server is reachable with the name 'gitlab'

- Gitlab credentials:

Username	Password
root	welcome123

- The Jenkins server is reachable with the name 'jenkins'
- Jenkins credentials:

Username	Password
admin	welcome123

- The test deployment server is reachable by the name "test-server"
- Test server SSH credentials:

Username	Password
tomcat	password1

Lab Setup

On starting the lab, the following interface will be accessible to the user.



Kali Jenkins GitLab Test Server

Kali

Jenkins

GitLab

Test Server

On choosing (clicking the text in the center) top left panel, **KALI CLI** will open in a new tab

```
root@kali-cli:~#
```

Similarly on selecting the top right panel, a web UI of **Jenkins** will open in a new tab.



Welcome to Jenkins!

Sign in

☐ Keep me signed in

On selecting the bottom left panel, a web UI of **Gitlab** will open in a new tab.



GitLab Community Edition

Open source software to collaborate on code

Manage Git repositories with fine-grained access controls that keep your code secure. Perform code reviews and enhance collaboration with merge requests. Each project can also have an issue tracker and a wiki.

Sign in	Register
Username or email	
<input type="text"/>	
Password	
<input type="password"/>	
<input type="checkbox"/> Remember me	Forgot your password?
Sign in	

And on selecting the bottom right panel, a web UI of **Test Server** will open in a new tab.

PENTESTER ACADEMY

WebApp will appear once deployed!

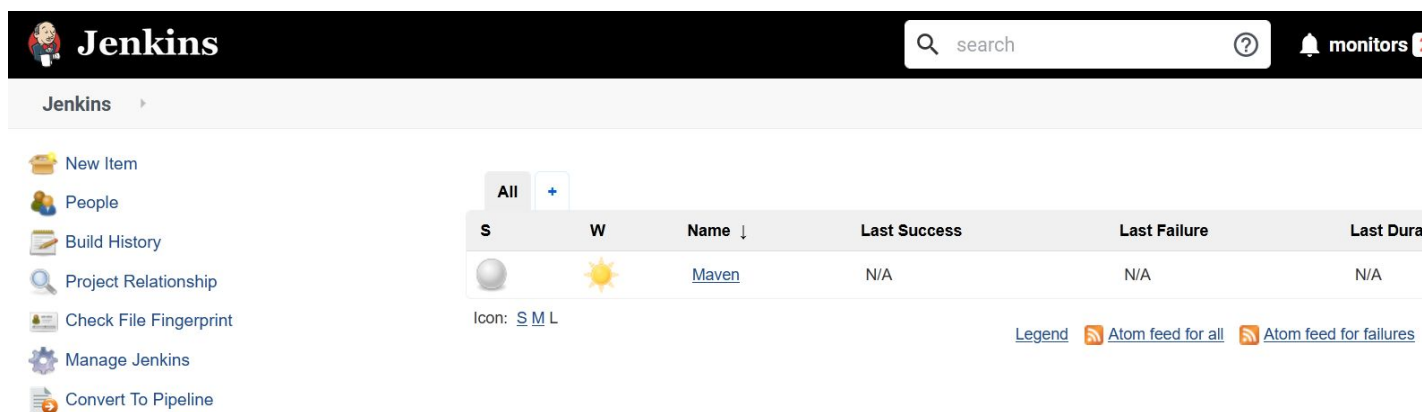
The page will reload until the test-server has started running the web service at port 8080

Solution

Step 1: Login into the Jenkins, The credentials are provided in the challenge description.

Credentials:

- **Username:** admin
- **Password:** welcome123

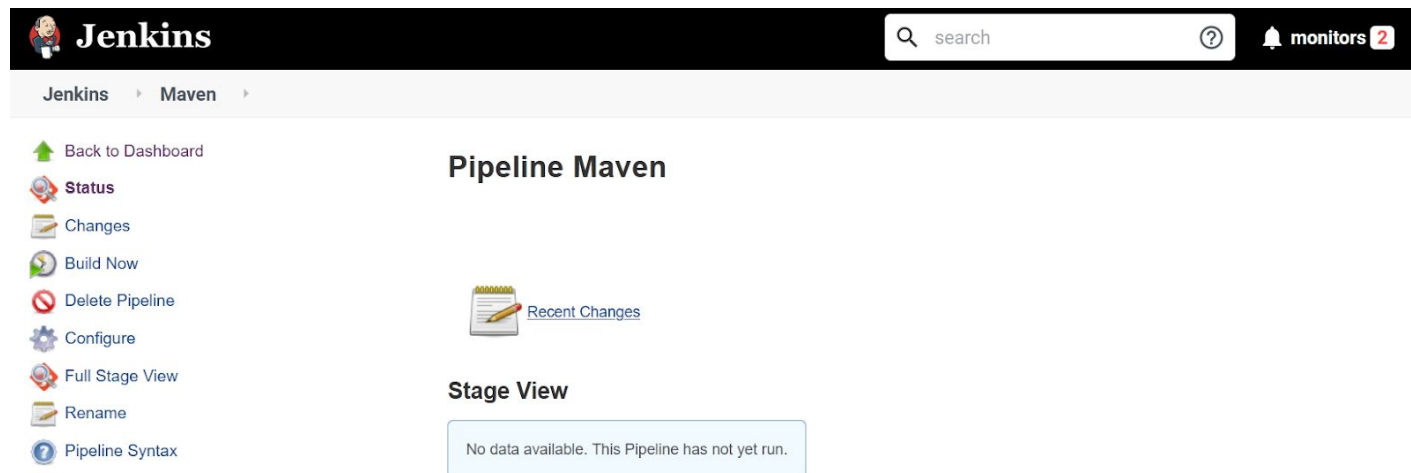


The screenshot shows the Jenkins web interface. At the top is a black header with the Jenkins logo, a search bar, and a 'monitors' button. Below the header is a sidebar with navigation links: 'Jenkins', 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'Convert To Pipeline'. The main content area displays a table of build items. The table has columns for 'S' (Status), 'W' (Weather icon), 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. A single item is listed with a status icon, a sun weather icon, and the name 'Maven'. Below the table, there are links for 'Icon: S M L' and 'Legend'. At the bottom right, there are links for 'Atom feed for all' and 'Atom feed for failures'.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		Maven	N/A	N/A	N/A

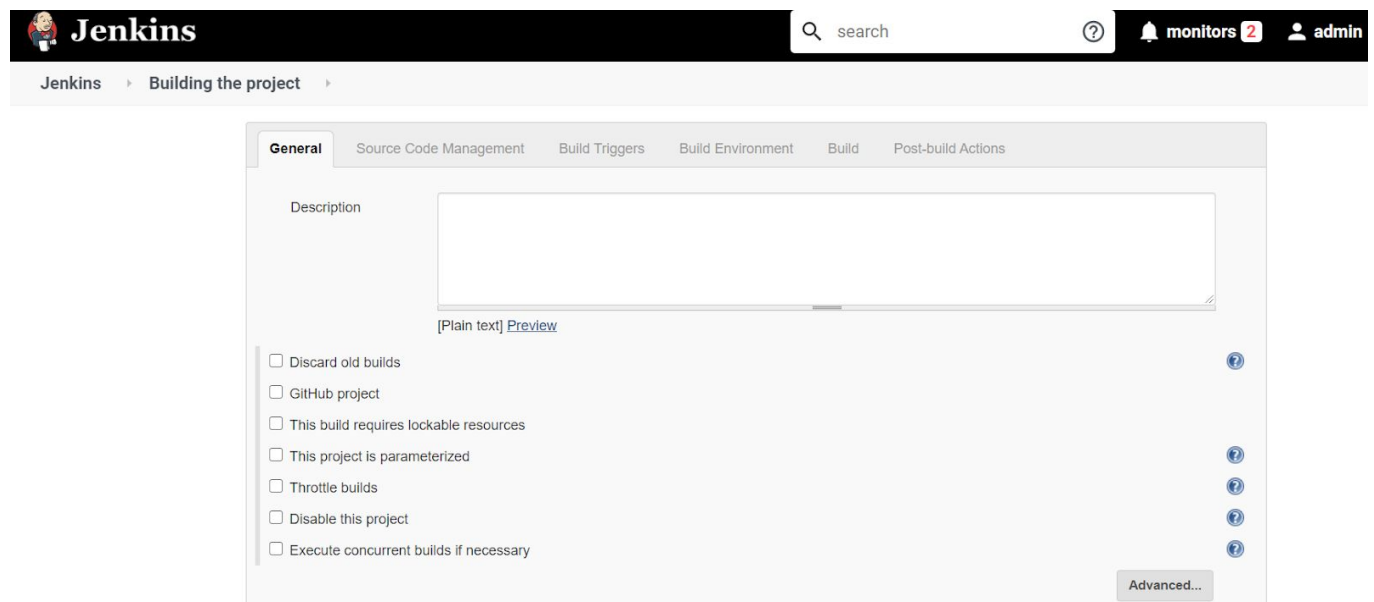
There is only one job present in the Jenkins Interface, We will take one job at a time to study.

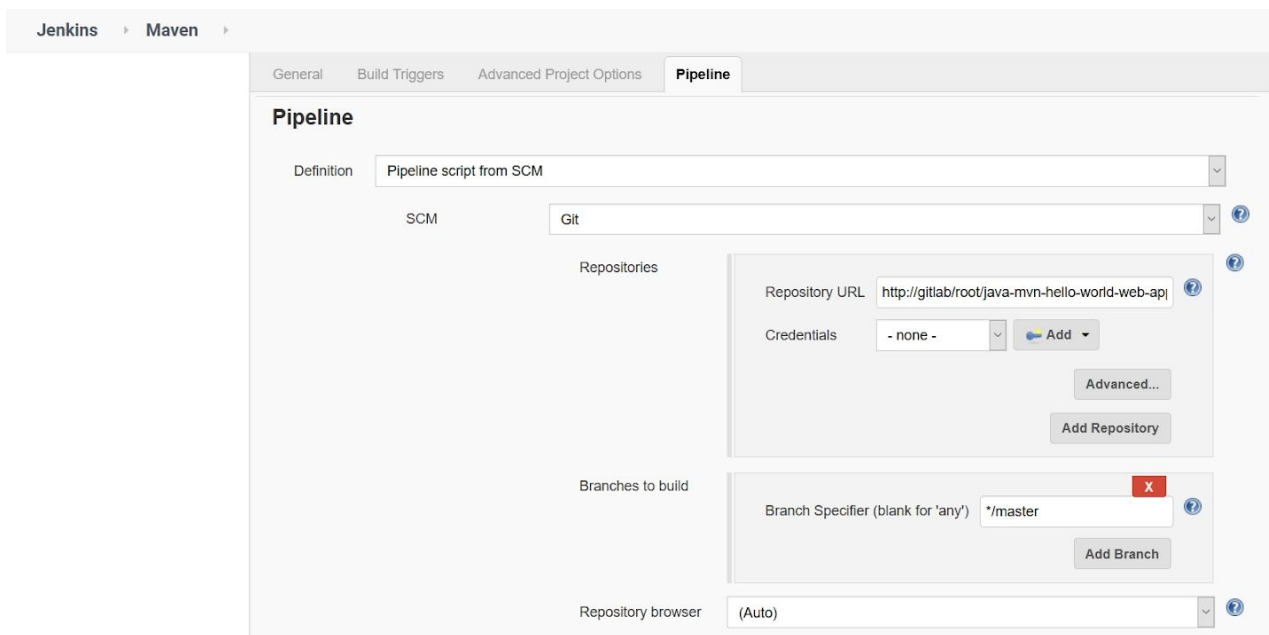
Step 2: Click on the “Maven” job.



This page is for “Pipeline maven” job, The Pipeline is appended in front of the Job name because this is a “Pipeline” type job in which it accepts a ‘Jenkinsfile’ which has all the commands and configuration of the pipeline.

Step 2: Click on the “Configure” option to check the configuration of the Job.



A screenshot of the Jenkins web interface showing the 'Pipeline' configuration tab for a project named 'Maven'. The 'Definition' is set to 'Pipeline script from SCM'. The 'SCM' is set to 'Git'. Under 'Repositories', the 'Repository URL' is 'http://gitlab/root/java-mvn-hello-world-web-app', 'Credentials' is '- none -', and there are buttons for 'Add', 'Advanced...', and 'Add Repository'. Under 'Branches to build', the 'Branch Specifier (blank for \'any\')' is set to '*/master', and there is an 'Add Branch' button. The 'Repository browser' is set to '(Auto)'.

Jenkins » Maven »

General Build Triggers Advanced Project Options **Pipeline**

Pipeline

Definition Pipeline script from SCM

SCM Git

Repositories

Repository URL http://gitlab/root/java-mvn-hello-world-web-app

Credentials - none - Add Advanced... Add Repository

Branches to build

Branch Specifier (blank for 'any') */master Add Branch

Repository browser (Auto)

The “Pipeline” sections accept Jenkinsfile directly or a source such as Gitlab where the code and Jenkinsfile are stored for the project.

The code is hosted on GitLab instance at this path
“http://gitlab/root/java-mvn-hello-world-web-app.git”

A screenshot of the GitLab web interface showing the 'Projects' page. The 'Your projects' section shows 1 project, 'Administrator / java-mvn-hello-world-web-app', which is a 'Maintainer' project. The page includes a search bar, filters for 'All' and 'Personal' projects, and a list of project statistics (stars, forks, etc.).

GitLab Projects Groups More

Search or jump to...

Projects

Your projects 1 Starred projects 0 Explore projects Filter by name...

All Personal

J Administrator / java-mvn-hello-world-web-app Maintainer

★ 0 🍴 0 📄 0

Step 3: Open the project on Gitlab and check the Jenkinsfile to build the pipeline.


```

5     stage ('Building the project - Checkout') {
6         checkout([$class: 'GitSCM', branches: [[name: '*/master']], doGenerateSubmoduleConfigurations: false, extensions: [], submoduleCfg: [], userRe
7     }
8     stage ('Building the project - Build') {
9         // Maven build step
10        withMaven {
11            if(isUnix()) {
12                sh "mvn compile "
13            } else {
14                bat "mvn compile "
15            }
16        } // Maven build step
17        withMaven {
18            if(isUnix()) {
19                sh "mvn test "
20            } else {
21                bat "mvn test "
22            }
23        } // Maven build step
24        withMaven {
25            if(isUnix()) {
26                sh "mvn package "
27            } else {
28                bat "mvn package "
29            }
30        } // Shell build step
31        sh """
32        mv target/*.war /tmp/ROOT.war
33        """
34    }

35    stage ('Tomcat Installation - Build') {
36        // Shell build step
37
38        ansiblePlaybook(
39            inventory: '',
40            playbook: '/ansible-scripts/tomcat.yml',
41        )
42
43        sh """
44        sleep 20
45        """
46    }
47    stage ('Selenium Testing - Build') {
48        // Shell build step
49        sh """
50        pytest --capture=no --verbose /selenium_script.py
51        """
52    }
53
54
55
56 }
57 }

```

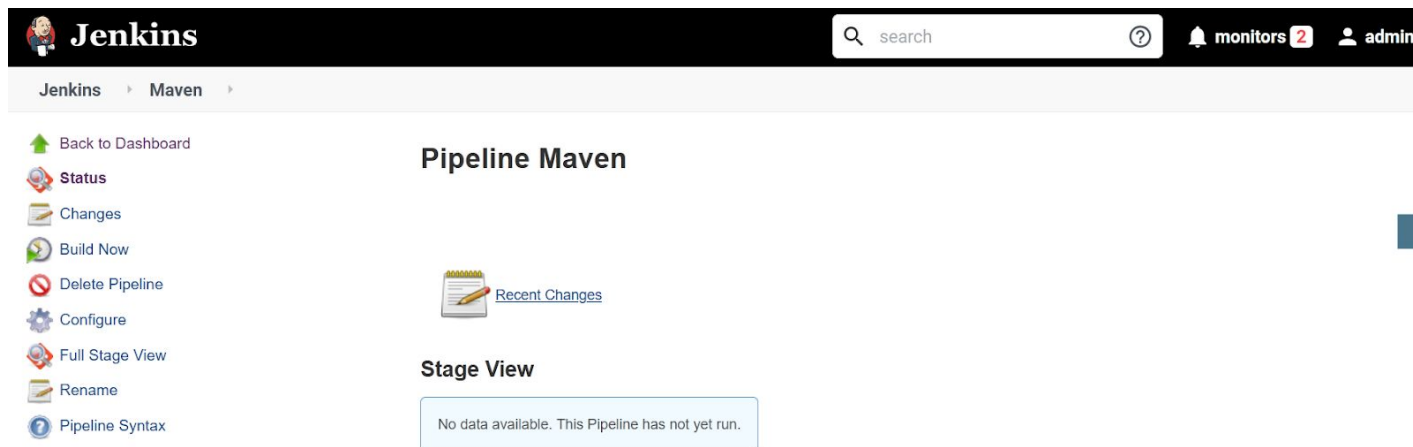
The file includes 'stages' that are a collection of steps, Each step performs a function which is explained below:-

Jenkinsfile Stages:

- **Building the project - checkout:** In this stage, the git repository will be checked for any updates or commits. If commits are found in the repository then the new files will be fetched from remote repository.
- **Building the project - Build:** In this stage, the maven will compile, test and package the source code to a distributable format like JAR, WAR
- **Tomcat Installation:** In this stage, the ansible will initiate the installation of Tomcat on the remote server (test-server).
- **Selenium Testing:** In this stage, the Jenkins will start checks on the newly deployed server to verify if the installation was successful or not.

Pipeline Execution

Step 1: Navigate to the Pipeline tab.



Step 2: Click on the “Build Now” button to start the Pipeline.

Jenkins

Jenkins
Maven

[Back to Dashboard](#)
[Status](#)
[Changes](#)
[Build Now](#)
[Delete Pipeline](#)
[Configure](#)
[Full Stage View](#)
[Rename](#)
[Pipeline Syntax](#)

Pipeline Maven

[Recent Changes](#)

Stage View

Building the project - Checkout

487ms

Building the project - Build

647ms

#1

Sep 26 13:40

No Changes

487ms

Build History

#1

Sep 26, 2020, 8:10 AM

[Atom feed for all](#)
[Atom feed for failures](#)

Reload the page to see the recent changes in the pipeline

Jenkins

Jenkins
Maven

[Back to Dashboard](#)
[Status](#)
[Changes](#)
[Build Now](#)
[Delete Pipeline](#)
[Configure](#)
[Full Stage View](#)
[Rename](#)
[Pipeline Syntax](#)

Pipeline Maven

[add description](#)

Last Successful Artifacts

mvn-hello-world-1.0-SNAPSHOT.pom

3.06 KB

view

mvn-hello-world-1.0-SNAPSHOT.war

10.89 MB

view

[Recent Changes](#)

Test Result Trend

Failed

Skipped

Passed

2

1.5

1

0.5

0

#1

Build History

#1

Sep 26, 2020, 10:12 AM

[Atom feed for all](#)
[Atom feed for failures](#)

Stage View

Average stage times:

458ms

16s

28s

47s

Average full run time: ~1min 35s

#1

Sep 26 15:42

No Changes

Building the project - Checkout

458ms

Building the project - Build

16s

Tomcat Installation - Build

28s

Selenium Testing - Build

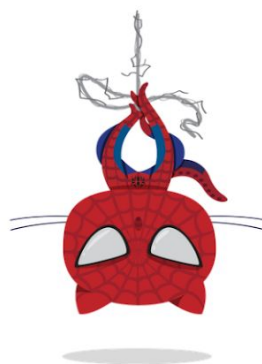
47s

The pipeline completed the execution successfully.

©PentesterAcademy.com

www.attackdefense.com

Step 3: Navigate to the deployed website.



Hello World!

This is a sample application

Learning

Working of a simple DevOps pipeline consisting of different components.