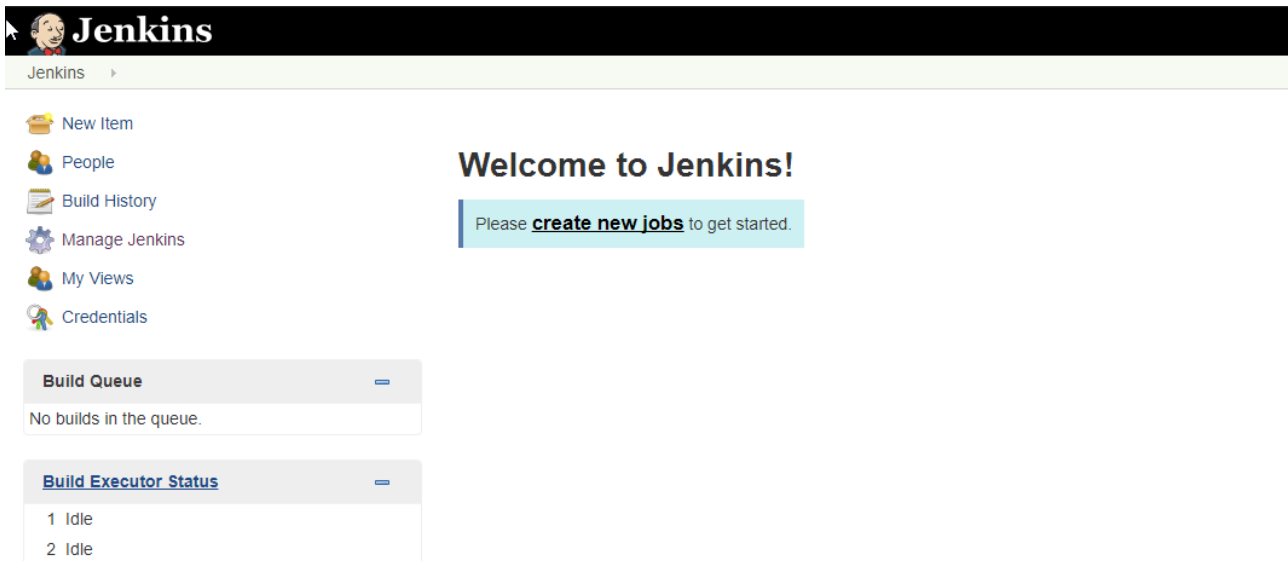


PROJECT SETUP

Choose 'Create new job'



The Jenkins Welcome Screen features a black header with the Jenkins logo and name. Below the header is a navigation menu with links: New Item, People, Build History, Manage Jenkins, My Views, and Credentials. A large blue box in the center says 'Welcome to Jenkins!' and 'Please create new jobs to get started.' Below this are two sections: 'Build Queue' showing 'No builds in the queue.' and 'Build Executor Status' showing two 'Idle' executors.

Jenkins

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Credentials

Welcome to Jenkins!

Please **create new jobs** to get started.

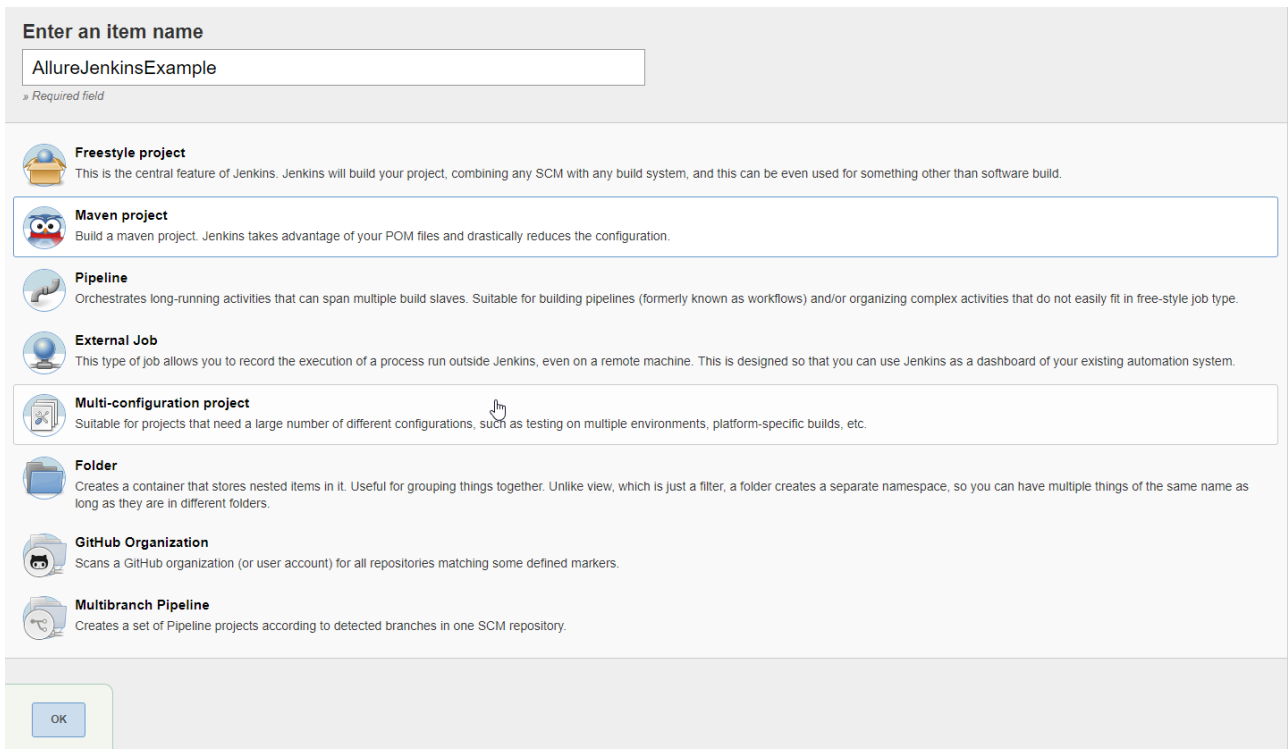
Build Queue

No builds in the queue.

Build Executor Status

- 1 Idle
- 2 Idle

Choose Maven Project.



The Jenkins Job Creation Screen shows a form to enter an item name, with 'AllureJenkinsExample' entered. Below the form is a list of job types: Freestyle project, Maven project, Pipeline, External Job, Multi-configuration project, Folder, GitHub Organization, and Multibranch Pipeline. The 'Maven project' option is highlighted. An 'OK' button is at the bottom.

Enter an item name

AllureJenkinsExample

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**
Orchestrates long-running activities that can span multiple build slaves. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
- External Job**
This type of job allows you to record the execution of a process run outside Jenkins, even on a remote machine. This is designed so that you can use Jenkins as a dashboard of your existing automation system.
- Multi-configuration project**
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
- Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- GitHub Organization**
Scans a GitHub organization (or user account) for all repositories matching some defined markers.
- Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

Click 'OK'

If maven is not installed by default, Install the following plugins from the Manage Plugins Section.

Updates Available Installed Advanced		
Install		
	Name	Version
<input checked="" type="checkbox"/>	Maven Artifact ChoiceListProvider (Nexus) A ChoiceListProvider to read Maven artifact information from a Nexus repository or MavenCentral	1.1.3
<input checked="" type="checkbox"/>	Maven Metadata Plugin for Jenkins CI server	1.5.0
<input checked="" type="checkbox"/>	Maven Integration plugin This plug-in provides deep integration of Jenkins and Maven. This functionality used to be part of the Jenkins core. Now it is a plug-in that is installed by default, but can be disabled.	2.16
<input type="checkbox"/>	Maven Dependency Update Trigger	1.5
<input type="checkbox"/>	Unleash Maven Plugin Provides Maven release functionality using the Unleash Maven Plugin.	2.3.0
<input type="checkbox"/>	Maven Release Plug-in Plug-in A plug-in that enables you to perform releases using the maven-release-plugin from Jenkins.	0.14.0
<input type="checkbox"/>	Config File Provider Plugin Ability to provide configuration files (e.g. settings.xml for maven, XML, groovy, custom files,...) loaded through the UI which will be copied to the job workspace	2.16.0
<input type="checkbox"/>	Jira Issue Updater Jenkins plugin which can update Jira issues in a maven or a freestyle job.	1.18
<input checked="" type="checkbox"/>	Maven Invoker plugin Reports on Maven Invoker it tests	1.2
<input type="checkbox"/>	Maven Repository Scheduled Cleanup Plugin	1.2
<input type="checkbox"/>	Repository Connector Repository Connector adds a build step which allows to resolve artifacts from a maven repository like nexus.	1.1.3
<input checked="" type="checkbox"/>	Maven Info Plugin	0.2.0

Installing Plugins/Upgrades

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Maven Metadata Plugin for Jenkins CI server	Downloaded Successfully. Will be activated during the next boot
Javadoc Plugin	Downloaded Successfully. Will be activated during the next boot
Maven Integration plugin	Downloaded Successfully. Will be activated during the next boot
Maven Info Plugin	Downloaded Successfully. Will be activated during the next boot
Maven Integration plugin	Success
Maven Invoker plugin	Downloaded Successfully. Will be activated during the next boot
Extensible Choice Parameter plugin	Downloaded Successfully. Will be activated during the next boot
Maven Artifact ChoiceListProvider (Nexus)	Installing
Restarting Jenkins	Pending

➡ [Go back to the top page](#)
(you can start using the installed plugins right away)

➡ ☒ Restart Jenkins when installation is complete and no jobs are running

Now, Validate the project title

GeneralMaven Info Plugin ConfigurationSource Code ManagementBuild TriggersBuild EnvironmentPre StepsBuildPost StepsBuild SettingsPost-build Actions

Maven project nameAllureJenkinsExample

Description

[Plain text] Preview

Maven Info Plugin Configuration

☐ Assign Visible Name

☐ Assign Description

☐ Discard old builds

☐ GitHub project

☐ This project is parameterized

☐ Throttle builds

☐ Disable this project

☐ Execute concurrent builds if necessary

Advanced...Advanced...

Now, Enter the URL of your Remote Repository.

Source Code Management

☐ None
☒ Git

Repositories

Repository URL

Credentials [Add](#)

[Advanced...](#)
[Add Repository](#)

Branches to build

Branch Specifier (blank for 'any') [Add Branch](#)

Repository browser

Additional Behaviours [Add](#)

☐ Subversion

Now, the Build Steps

Build

Root POM

Goals and options

MAVEN_OPTS

☐ Incremental build - only build changed modules
☐ Disable automatic artifact archiving
☐ Disable automatic site documentation artifact archiving
☐ Disable automatic fingerprinting of consumed and produced artifacts
☒ Enable triggering of downstream projects
☒ Block downstream trigger when building

☐ Build modules in parallel
☐ Use private Maven repository
☐ Resolve Dependencies during Pom parsing
☐ Run Headless
☐ Process Plugins during Pom parsing

☒ Use custom workspace
Directory

Maven Validation Level

Settings file

Global Settings file

Now, Post-Step

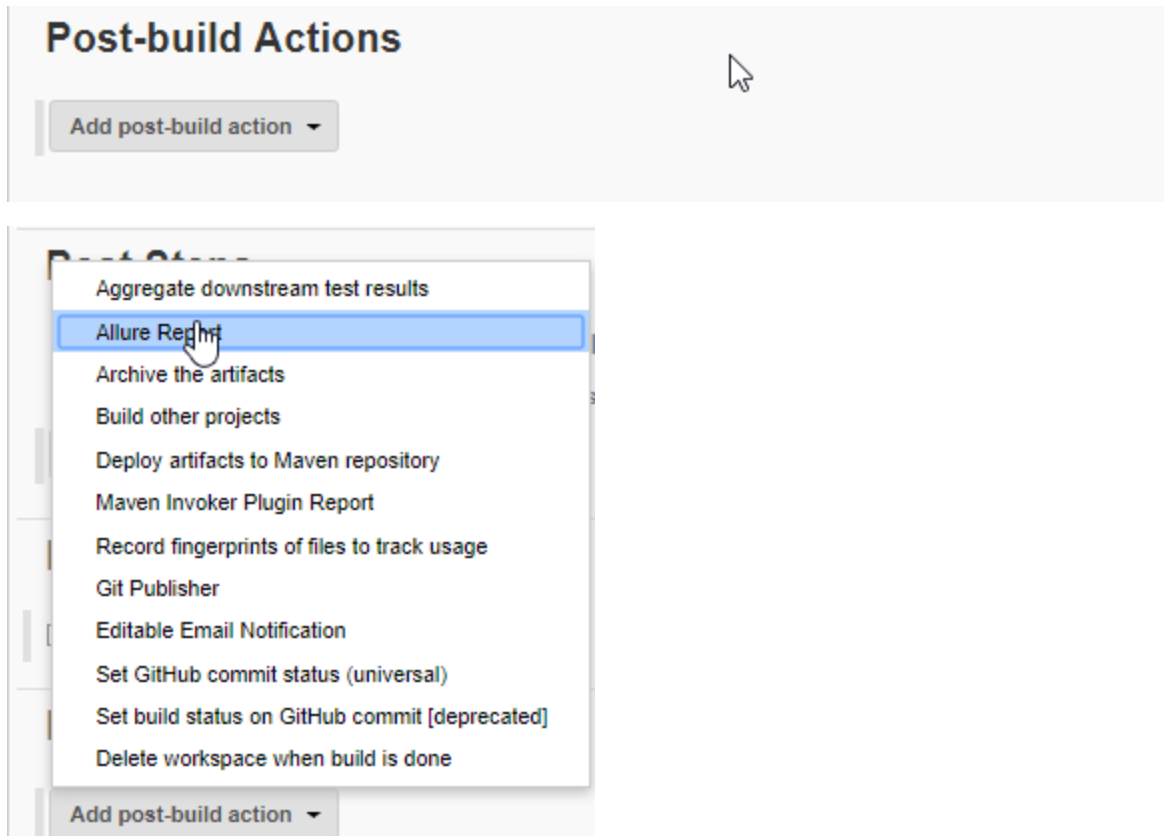
Post Steps

☐ Run only if build succeeds ☐ Run only if build succeeds or is unstable ☒ Run regardless of build result

Should the post-build steps run only for successful builds, etc.

[Add post-build step](#)

Finally, Setting up Allure Reports.



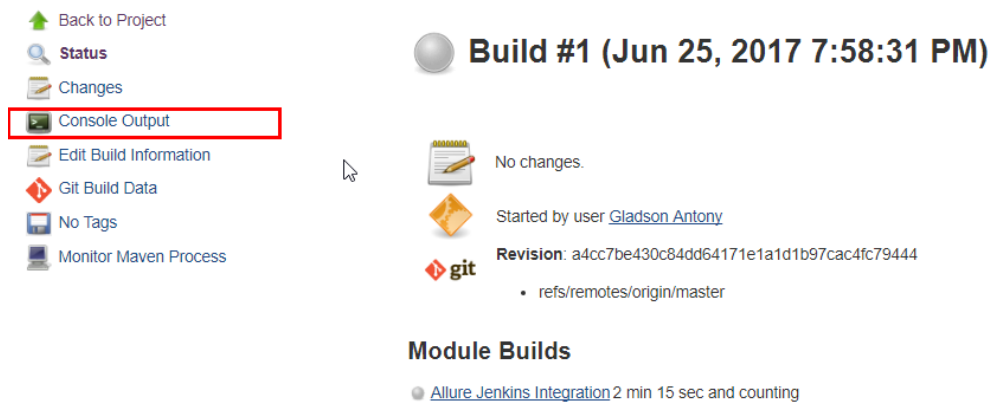
Change the allure reports path based on you project



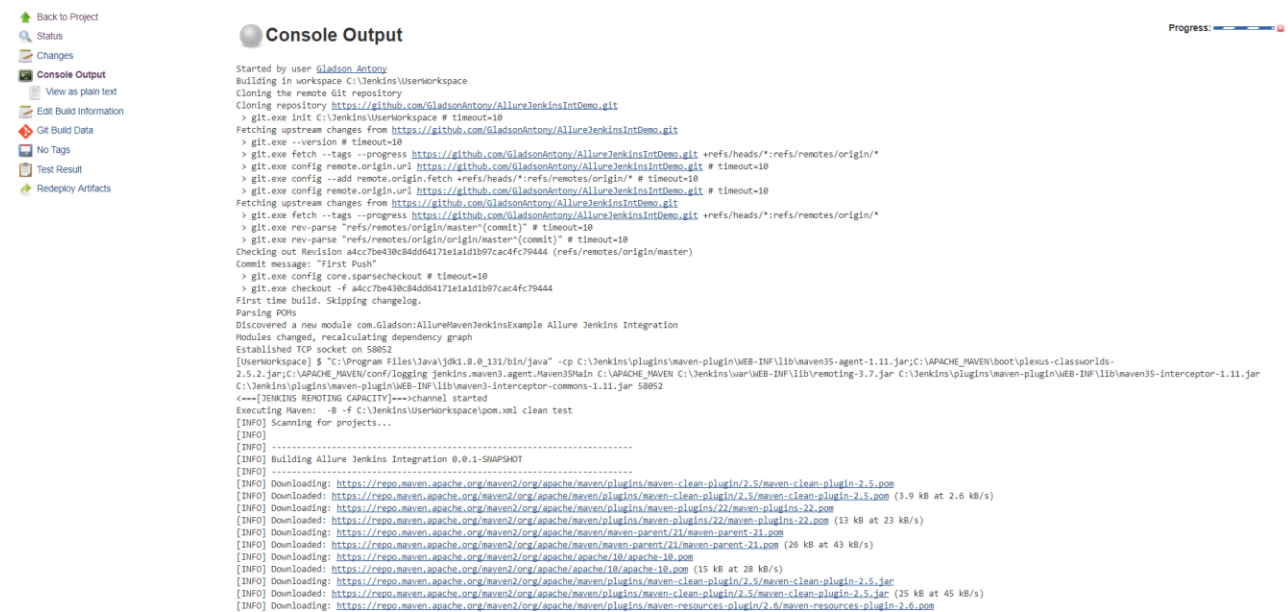
Now. Save the contents. Navigate to the Dashboard, choose your project and select Build Now. You'll see the status.



Now, click on build #1 and choose to see the Console Output. This will help you find the project status.



The screenshot shows the Jenkins interface for Build #1 (Jun 25, 2017 7:58:31 PM). On the left sidebar, the 'Console Output' link is highlighted with a red box. The main area shows the build status as 'No changes' and 'Started by user Gladson Antony'. Below this, the revision is listed as 'a4cc7be430c84dd64171e1a1d1b97cac4fc79444' with a link to 'refs/remotes/origin/master'. The 'Module Builds' section shows 'Allure Jenkins Integration' with a duration of '2 min 15 sec and counting'.

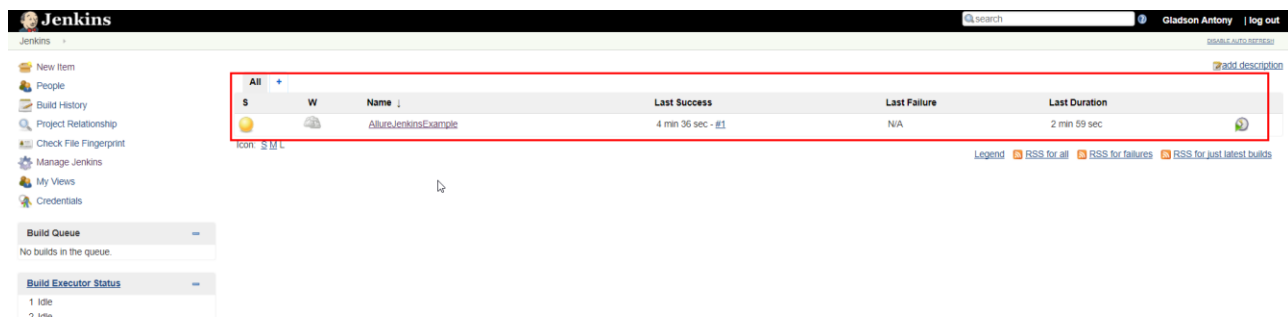


The screenshot shows the Jenkins Console Output for Build #1. The output text is as follows:


```
Started by user Gladson Antony
Building in workspace C:\Jenkins\workspace
Cloning the remote Git repository
Cloning repository https://github.com/GladsonAntony/AllureJenkinsIntDemo.git
> git.exe init C:\Jenkins\workspace # timeout=10
Fetching upstream changes from https://github.com/GladsonAntony/AllureJenkinsIntDemo.git
> git.exe --version # timeout=10
> git.exe fetch --tags --progress https://github.com/GladsonAntony/AllureJenkinsIntDemo.git +refs/heads/*:refs/remotes/origin/*
> git.exe config remote.origin.url https://github.com/GladsonAntony/AllureJenkinsIntDemo.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/GladsonAntony/AllureJenkinsIntDemo.git # timeout=10
Fetching upstream changes from https://github.com/GladsonAntony/AllureJenkinsIntDemo.git
> git.exe fetch --tags --progress https://github.com/GladsonAntony/AllureJenkinsIntDemo.git +refs/heads/*:refs/remotes/origin/*
> git.exe rev-parse 'refs/remotes/origin/master^{commit}' # timeout=10
> git.exe rev-parse 'refs/remotes/origin/master^{commit}' # timeout=10
Checking out Revision a4cc7be430c84dd64171e1a1d1b97cac4fc79444 (refs/remotes/origin/master)
Commit message: "First Push"
> git.exe config core.sparsecheckout # timeout=10
> git.exe checkout -f a4cc7be430c84dd64171e1a1d1b97cac4fc79444
First time build. Skipping changelog.
Parsing POMs
Discovered a new module com.Gladson:AllureJenkinsExample Allure Jenkins Integration
Modules changed, recalculating dependency graph
Established TCP socket on 58852
[Userworkspace] $ "C:\Program Files\Java\jdk1.8.0_131\bin\java" -cp C:\Jenkins\plugins\maven-plugin\WEB-INF\lib\maven3-agent-1.11.jar;C:\VAPACHE_MAVEN\boot\plexus-classworlds-2.5.2.jar;C:\VAPACHE_MAVEN\conf\logging_jenkins.maven3.agent\maven3-jar;C:\VAPACHE_MAVEN\c:\Jenkins\war\WEB-INF\lib\remoting-3.7.jar C:\Jenkins\plugins\maven-plugin\WEB-INF\lib\maven3-interceptor-1.11.jar C:\Jenkins\plugins\maven-plugin\WEB-INF\lib\maven3-interceptor-commons-1.11.jar 58852
<==[JENKINS REMOTING CAPACITY]==>channel started
Executing Maven: -B -f C:\Jenkins\workspace\pom.xml clean test
[INFO] Scanning for projects...
[INFO]
[INFO]
[INFO] Building Allure Jenkins Integration 0.0.1-SNAPSHOT
[INFO]
[INFO]
[INFO] Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom
[INFO] Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom (3.9 kB at 2.6 kB/s)
[INFO] Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-plugins-22.pom
[INFO] Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-plugins-22.pom (13 kB at 23 kB/s)
[INFO] Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/21/maven-parent-21.pom
[INFO] Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/21/maven-parent-21.pom (26 kB at 43 kB/s)
[INFO] Downloading: https://repo.maven.apache.org/maven2/org/apache/apache/10/apache-10.pom
[INFO] Downloaded: https://repo.maven.apache.org/maven2/org/apache/apache/10/apache-10.pom (15 kB at 28 kB/s)
[INFO] Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.jar
[INFO] Downloaded: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.jar (25 kB at 45 kB/s)
[INFO] Downloading: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resources-plugin/2.6/maven-resources-plugin-2.6.pom
```

Once it's done.

Navigate to the Project location and select Allure report.



The screenshot shows the Jenkins Project Location page. The 'All' tab is selected, and a table lists the builds. The first build, 'Allure JenkinsExample', is highlighted with a red box. The table has columns for 'S', 'W', 'Name', 'Last Success', 'Last Failure', and 'Last Duration'. The 'Last Success' column shows '4 min 36 sec - #1'. The 'Last Failure' column shows 'N/A'. The 'Last Duration' column shows '2 min 59 sec'. Below the table, there are links for 'Build Queue' and 'Build Executor Status'.

 Jenkins

Jenkins > AllureJenkinsExample

Back to Dashboard

Status

Changes

Workspace

Build Now

Delete Maven project

Configure

Modules

Allure Report

Build History

trend

find

Jun 25, 2017 7:58 PM

RSS for all

RSS for failures

Maven project AllureJenkinsExample

Allure Report

Workspace

Last Successful Artifacts

allure-report.zip 957.38 KB view

Recent Changes

Latest Test Result (5 failures)

Latest Test Result (5 failures)

Permalinks

Last build (#1). 5 min 14 sec ago

Last successful build (#1). 5 min 14 sec ago

Last unstable build (#1). 5 min 14 sec ago

Last unsuccessful build (#1). 5 min 14 sec ago

Last completed build (#1). 5 min 14 sec ago

Proof!

Allure

Overview

Categories

xUnit

Graphs

Timeline

Behaviors

Packages

ALLURE REPORT 25/06/2017

25.01.11 - 25.01.12 (144ms)

10

test cases

50%

LAUNCHES

There is no information about launches

ENVIRONMENT

There are no environment variables

TREND

There is nothing to show

EXECUTORS

Jenkins AllureJenkinsExample #1

FEATURES BY STORIES 0 items total

GLADSON ANTONY

Menu Allure Report

Categories

name duration status Filter test cases by status

Product defects

expected [11] but found [10]

test1 0s

expected [12] but found [10]

test1 13ms

expected [13] but found [10]

test1 1ms

expected [14] but found [10]

test1 1ms

expected [15] but found [10]

test1 1ms

demo Demo1 test1

Failed test1

expected [11] but found [10]

Categories: Product defects

Severity: normal

Duration: 0s

Parameters

arg0: 10

arg1: 0

arg2: 11

Menu Allure Report

xUnit

name duration status Filter test cases by status

AllureJenkins

Test

demo Demo1

test1 9ms

test1 13ms

test1 1ms

test1 0s

test1 1ms

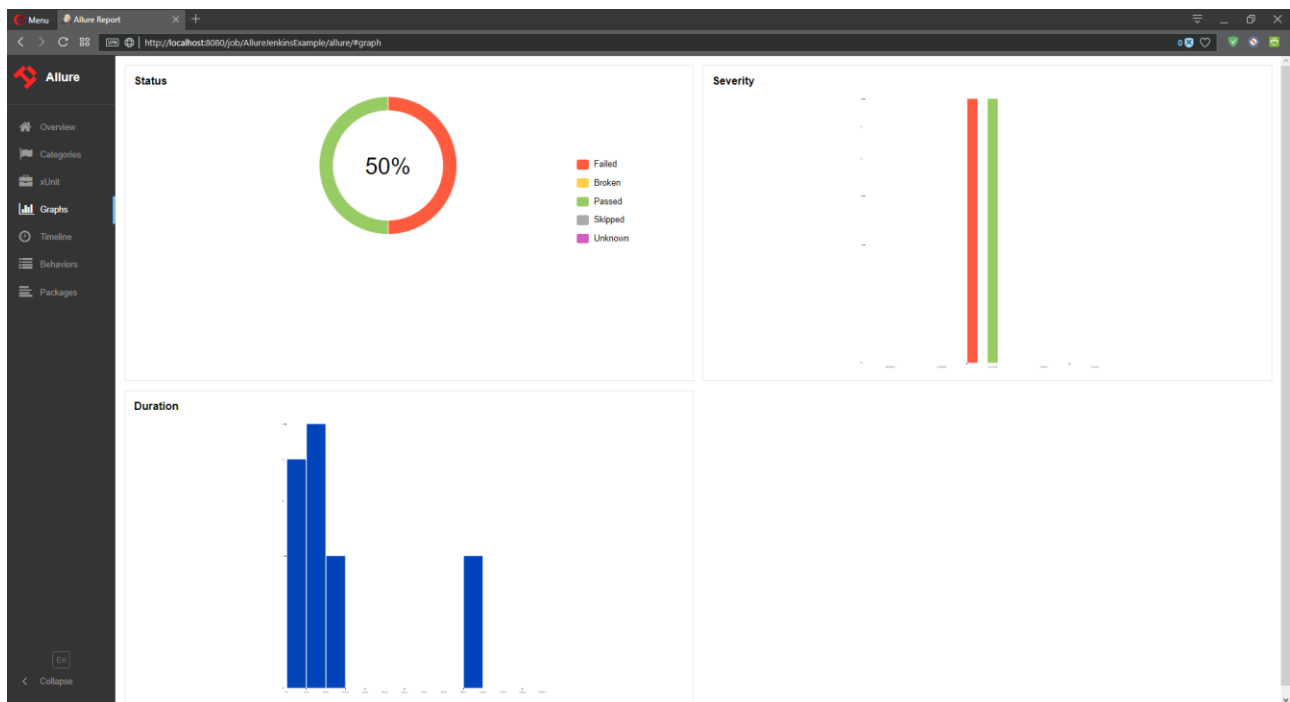
test1 0s

test1 1ms

test1 0s

test1 0s

test1 2ms



The Allure Report Behaviors page displays a table of test behaviors. The table has columns for name, duration, and status. The table is filtered by status, showing only 'Failed' and 'Passed' results.

name	duration	status
Default feature		
Default story		
test1	9ms	Failed
test1	13ms	Failed
test1	1ms	Failed
test1	1ms	Failed
test1	0s	Failed
test1	1ms	Failed
test1	0s	Failed
test1	1ms	Failed
test1	0s	Failed
test1	2ms	Failed

The Allure Report Packages page displays a table of test packages. The table has columns for name, duration, and status. The table is filtered by status, showing only 'Failed' and 'Passed' results.

name	duration	status
demo Demo1		
test1	9ms	Failed
test1	13ms	Failed
test1	1ms	Failed
test1	1ms	Failed
test1	0s	Failed
test1	1ms	Failed
test1	0s	Failed
test1	1ms	Failed
test1	0s	Failed
test1	2ms	Failed