Table of Contents

[1](#_Toc532554454)

[Table of Contents 2](#_Toc532554455)

[1 Lab Details: 3](#_Toc532554456)

[1 DevOps Tools: Guided Exercises 3](#_Toc532554457)

[1.1 Jenkins 3](#_Toc532554458)

[Guided Exercise 1: 4](#_Toc532554459)

[Guided Exercise 2: 12](#_Toc532554460)

# Lab Details:

You will be provided with the below DevOps Lab for practicing the guided exercise in this document.

Enrol for DevOps Tools - Practice Lab to practice these guided exercises.

# DevOps Tools: Guided Exercises

## Jenkins

A leading travel agency in Chennai is planning to build a new web application to improve their business. They provide travel and tourism related services to the public on behalf of their suppliers such as activities, airlines, car rentals, cruise lines, hotels, railways, travel insurance, and package tours.

The application is planned to be developed and deployed in a DevOps environment. There are many project builds which needs to be updated to the GIT repository. Jenkins is used as a continuous integration and delivery server in the environment.

**Scope:**

You have been assigned the task of managing the automated build and testing of the source codes submitted by the developers to a SCM Repository .Given the repository details, you need to perform the following main tasks.

* Get the project builds from the SCM repository
* Automate project Build process
* Automate Test cases execution
* Display the test reports

**Steps*:***

1. Install and configure Jenkins
2. Install required plugins in Jenkins
3. Jenkins configuration with Maven and Git
4. Create a Maven Project in Jenkins
5. Configure Junit Reports in Jenkins
6. Build the Application on Jenkins
7. Verify the unit test reports

**Pre- Requisites**

* Maven 3.5
* Jenkins
* Git

Guided Exercise 1: **Install Jenkins**

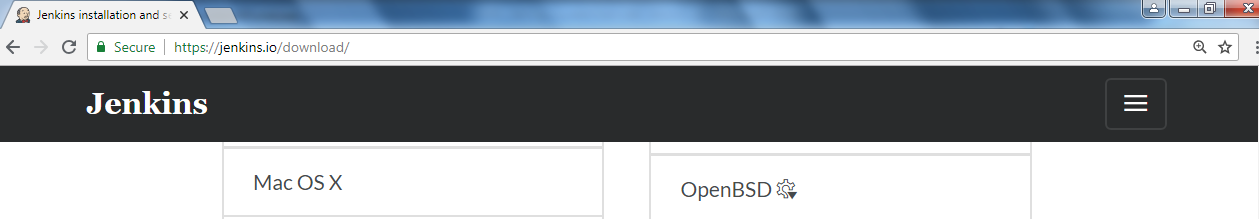
**Estimated Completion Time:** 10 Minutes

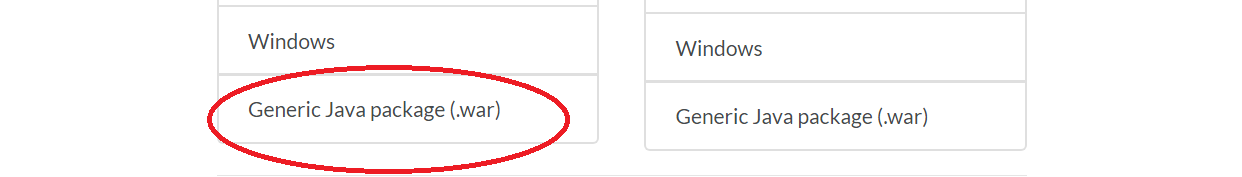
**Objective**: To Install and configure Jenkins on Windows Environment.

**Solution: Follow the given steps**

**Step 1:**

* + Download Jenkins and Set up Jenkins Environment.
  + Download Jenkins using the following URL [[**https://jenkins.io/download/**](https://jenkins.io/download/)](https://maven.apache.org/download.cgi) as shown in Fig 1.1

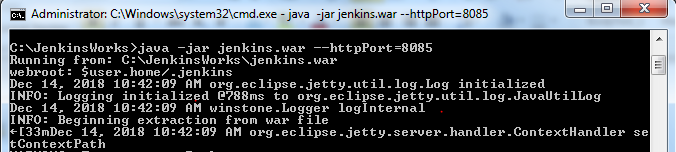




[Fig 1.1]

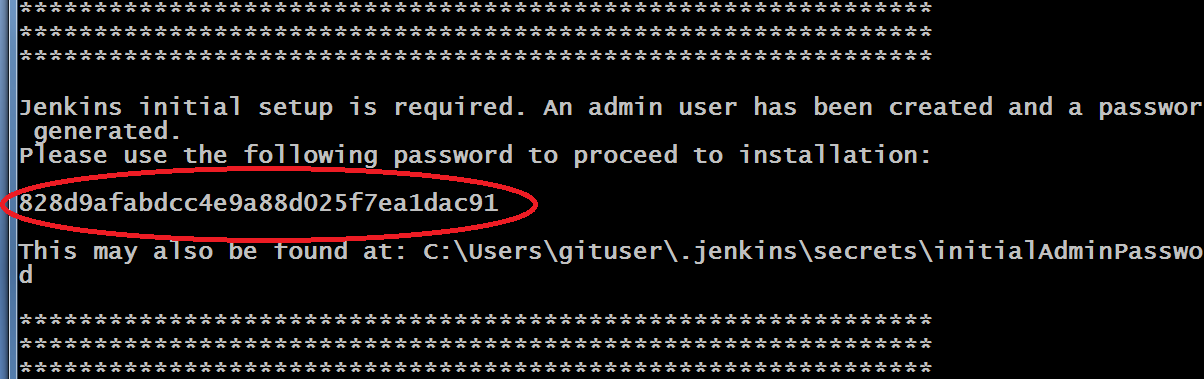
* + Copy the Jenkins.war to any folder. **C:\JenkinsWorks** is the path used in this example.
  + The default port to start Jenkins is 8080. Use –httpPort option to change the port number.
  + Open new command prompt
  + Navigate to the directory where jenkins.war is copied.
  + Use **java –jar jenkins.war –httpPort=8085** command to start Jenkins.

Refer Fig 1.2 and 1.3.



[Fig 1.2]

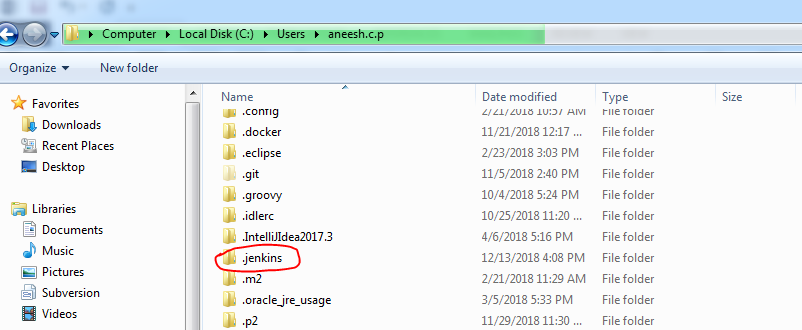
After Jenkins is started, it creates an initialAdminPassword by default. User can change/use this password for further reference.



[Fig 1.3]

* Jenkins creates a folder (.jenkins) in user’s profile (c:\users\username\.Jenkins).

Refer Figure 1.4

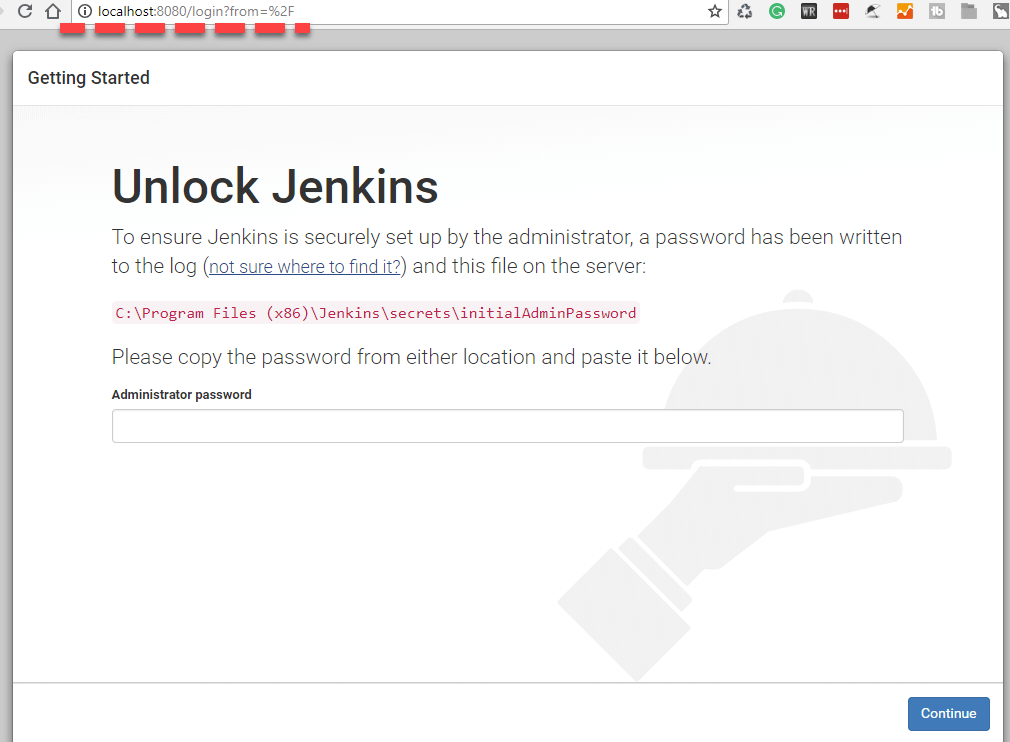


[Fig 1.4]

* Admin password is also stored in initialAdminPassword file in user’s profile. C:\Users\username\.jenkins\secrets\initialAdminPassword

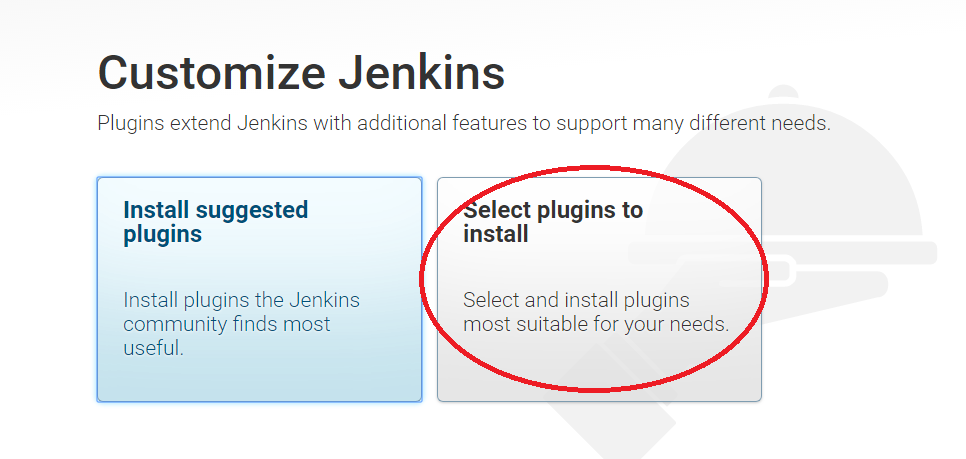
**Step 2: Set up the Jenkins Environment**

* Open the web browser and enter the URL: **http://localhost:8085** to open the Jenkins web page. For the first time the page asks for an admin password to unlock it as shown in Fig 1.5



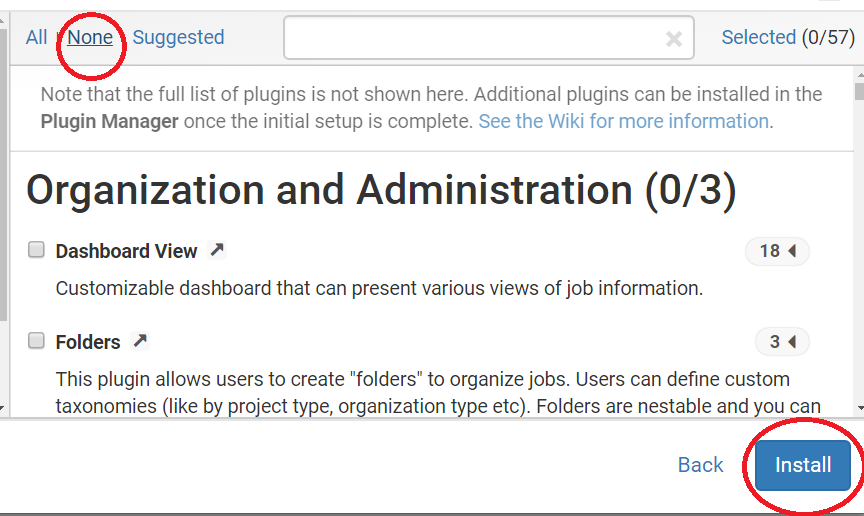
[Fig 1.5]

* Give the admin password to Unlock Jenkins either copied from the Jenkins startup console or from the initialAdminPassword file generated.
* Under Customize Jenkins, click on **Select plugins to install** option. Refer Fig 1.6. If you select the Install Suggested Plugins option the mandatory plugins will get installed before the home page is displayed. Refer Fig 1.8



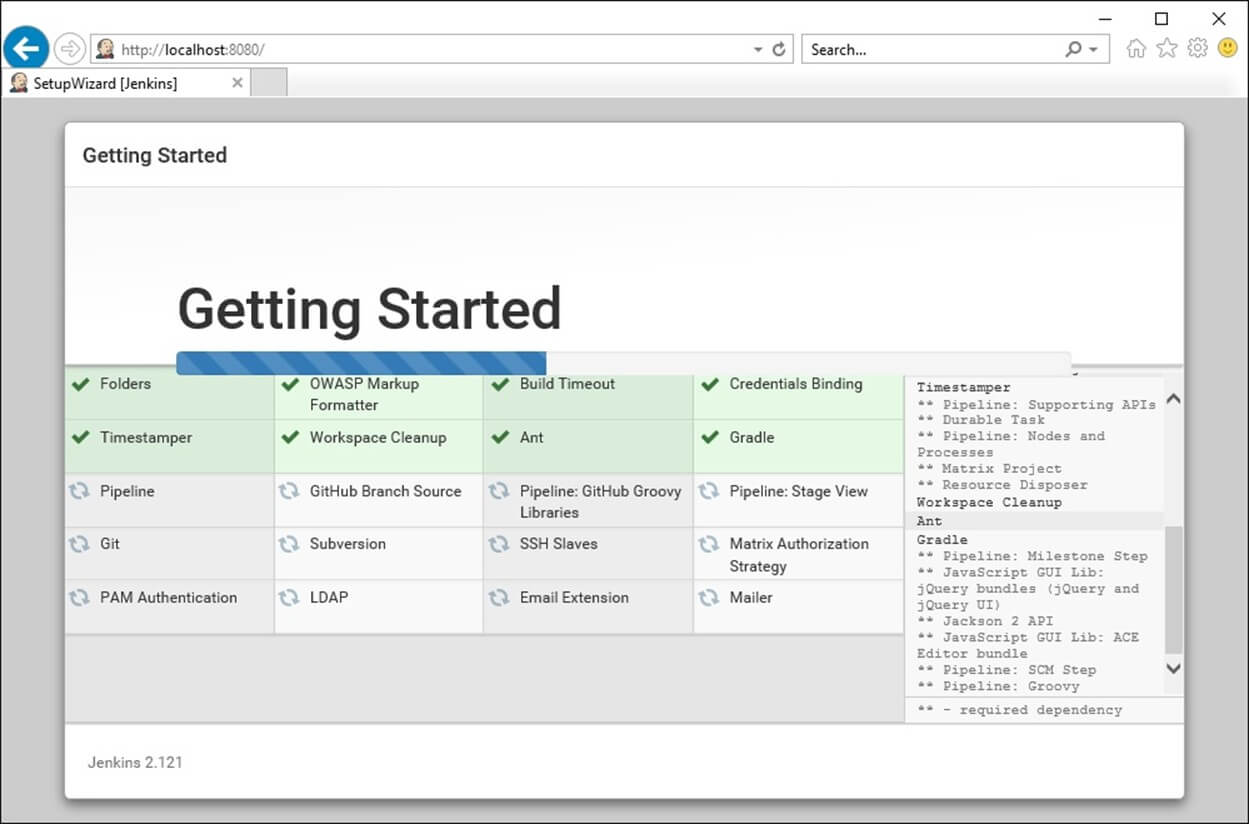
[Fig 1.6]

* Select **None** and click **install** as shown in Fig 1.7



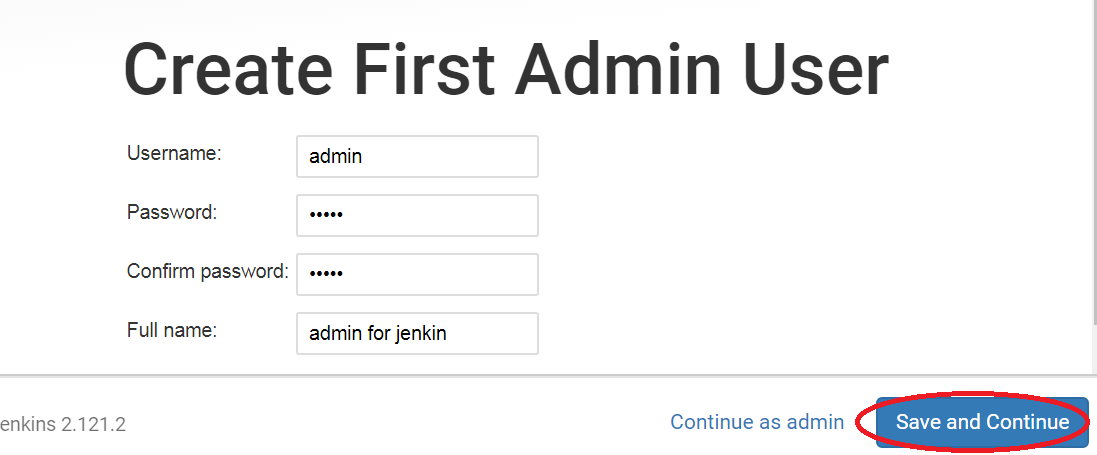
[Fig 1.7]

If you select suggested plugins the default plugins will get installed as shown in Figure 1.8



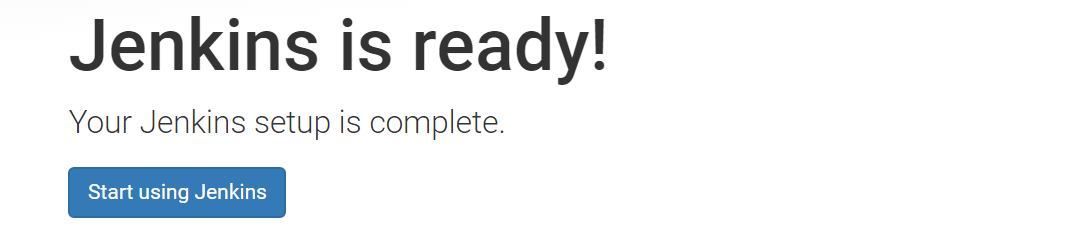
[Fig 1.8]

* Add user name and password (e.g. user admin password admin )
* Select save and continue Reference [Fig 1.9 -1.10]



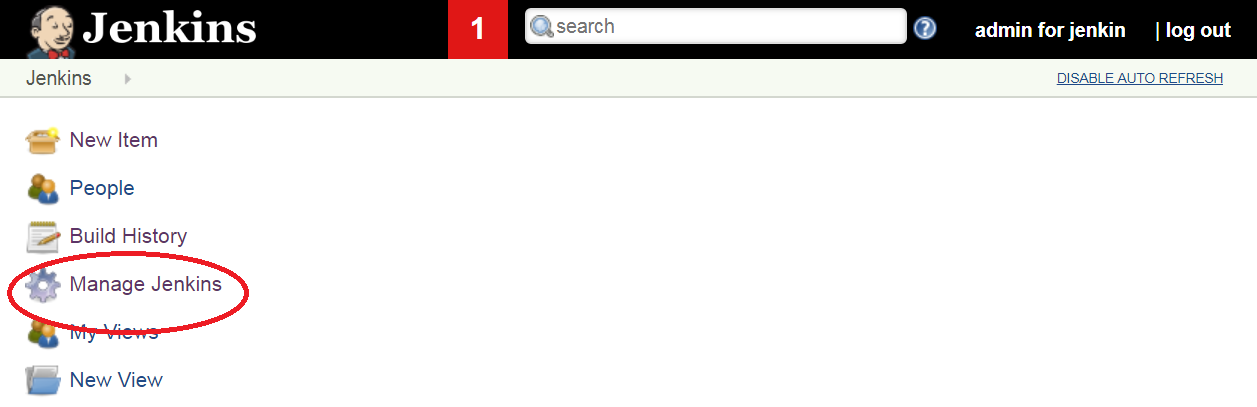
[Fig 1.9]

Save and Continue will redirect to Jenkins Ready page as shown in Figure 1.9



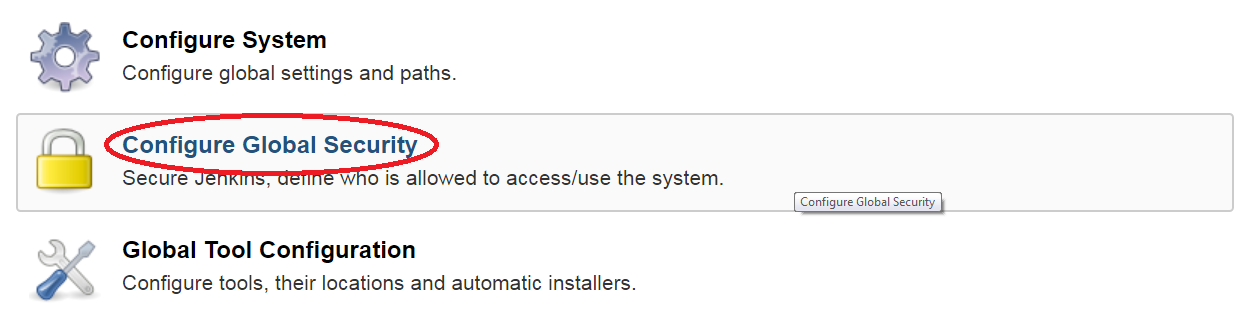
[Fig 1.10]

The following figure shows the Jenkins Homes Screen. Now the security permissions can be changed in Jenkins by using the **Manage Jenkins -> Configure Global Security** option. Refer Fig 1.11 and 1.12



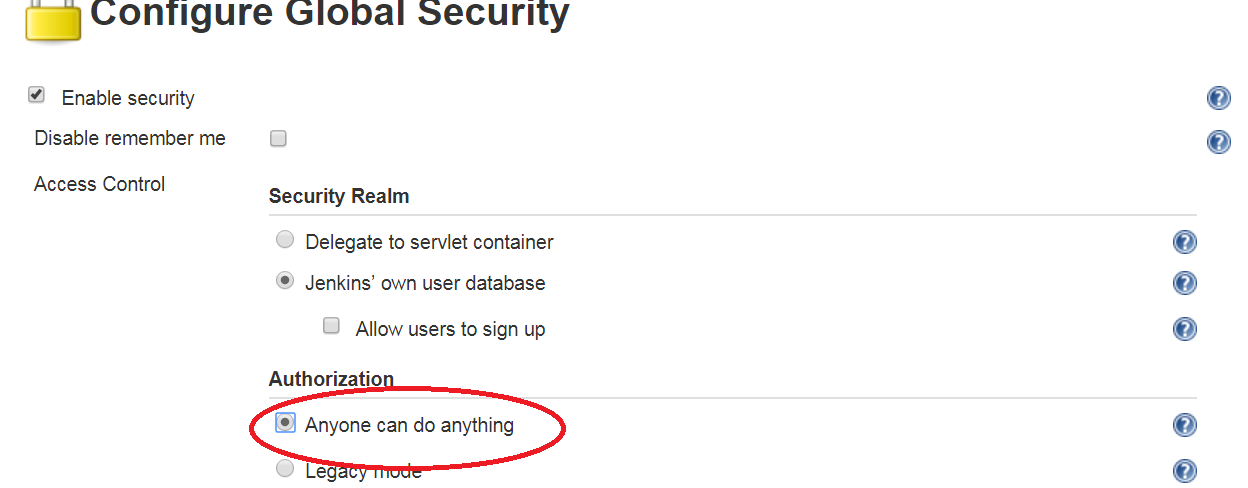
[Fig 1.11]

* Select Configure Global Security

****

[Fig 1.12]

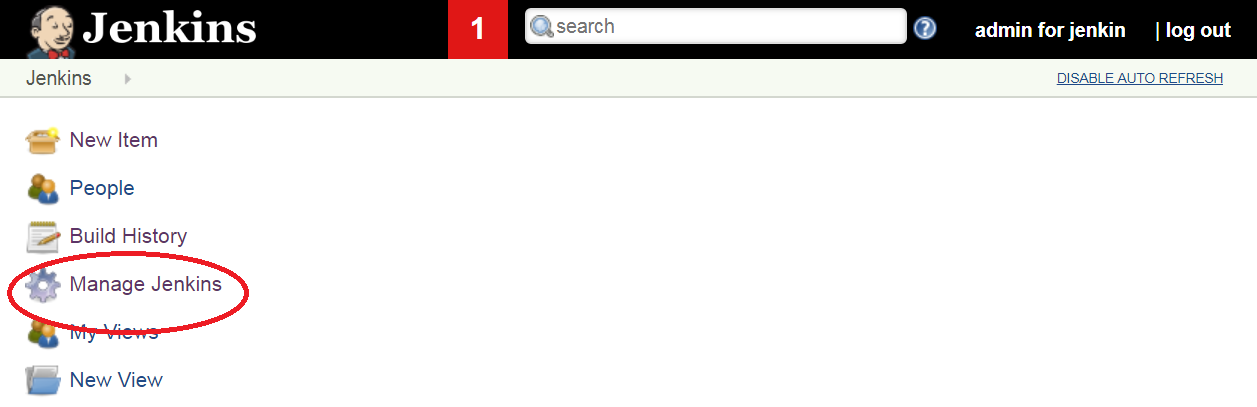
* In Authorization, select an option **Anyone can do anything** (apply and save the setting). Refer Fig 1.13

****

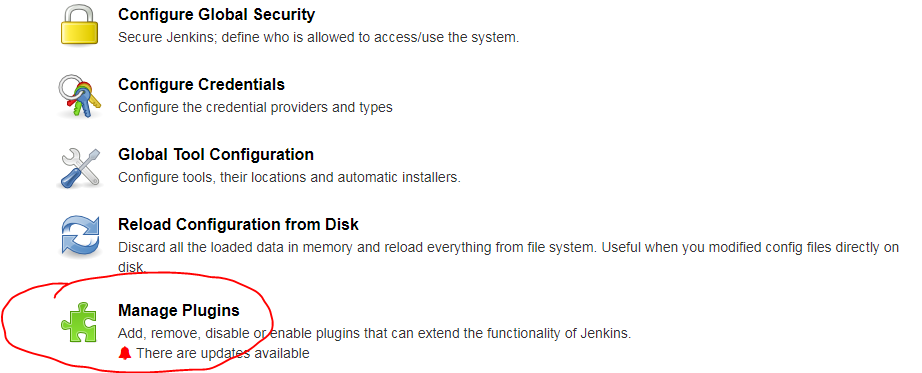
[Fig 1.13]

**Step 3:** Install or verify the required plugins.

To install new plugins, navigate to **Manage Jenkins -> Manage Plugins** option. Refer Fig 1.14 and 1.15



[Fig 1.14]



[Fig 1.15]

Navigate to available tab and check the plugin in the checkbox to be installed and use **Install without Restart** button to install the plugin without a Jenkins server restart.

Add the following plugins using the above mentioned step.

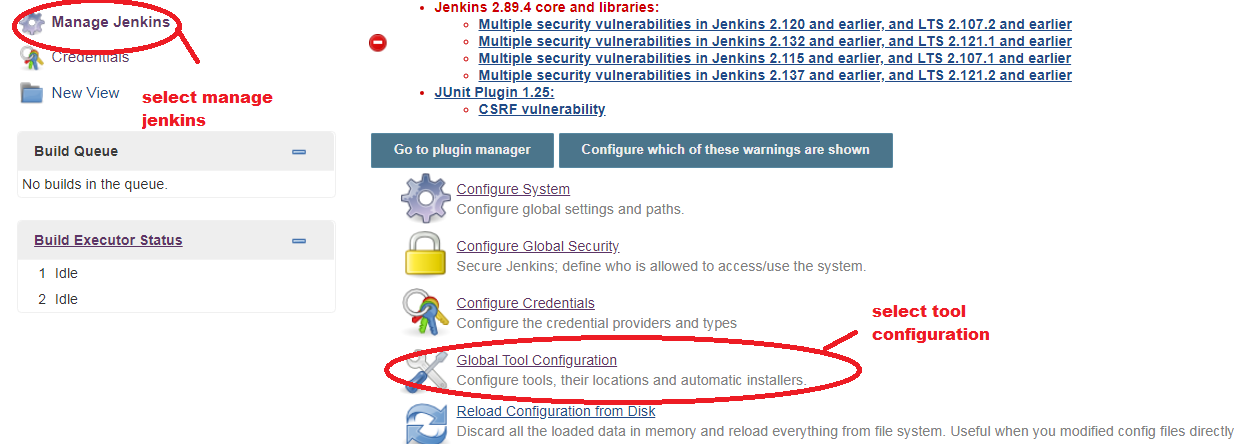
Plugins to be installed or verified for existence.

* Unleash maven plugin
* GitHub plugin

**Step 4: Configure JDK, Maven and Git on Jenkins.**

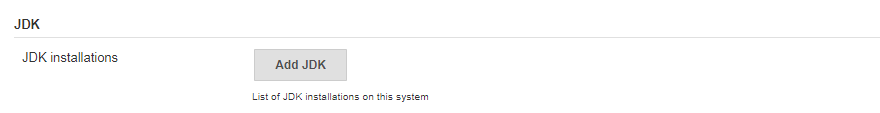
The installed plugins has to be configured to use it with Jenkins.

* Navigate to **Manage Jenkins ->Global Tool Configuration.** Refer Fig 1.16

****

[Fig 1.16]

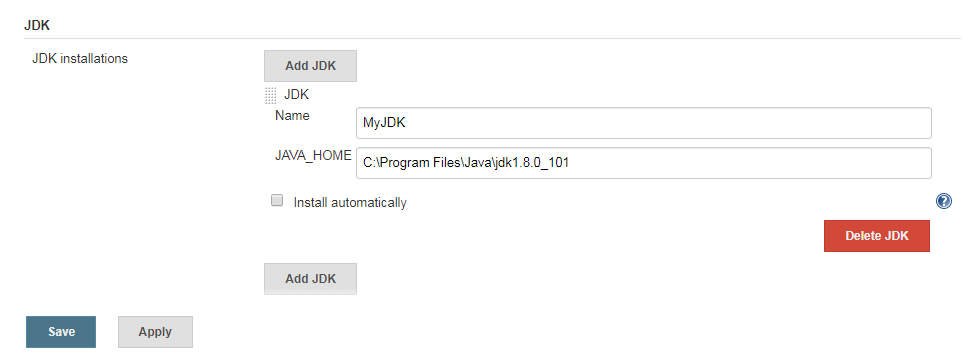
* Set up JDK. Click on JDK installations -> Add JDK. Refer Fig 1.17



[Fig 1.17]

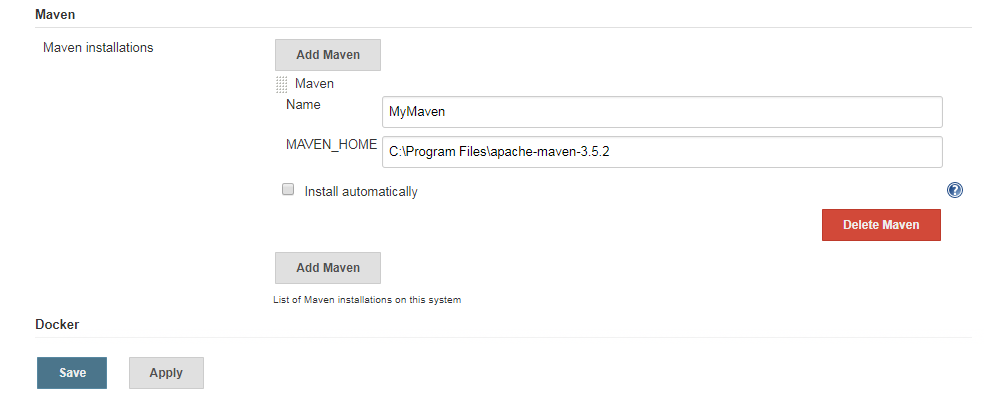
Specify the JDK Home path as per your JDK Installation.

Give a name and uncheck “**install automatically**” .Refer Fig 1.18



[Fig 1.18]

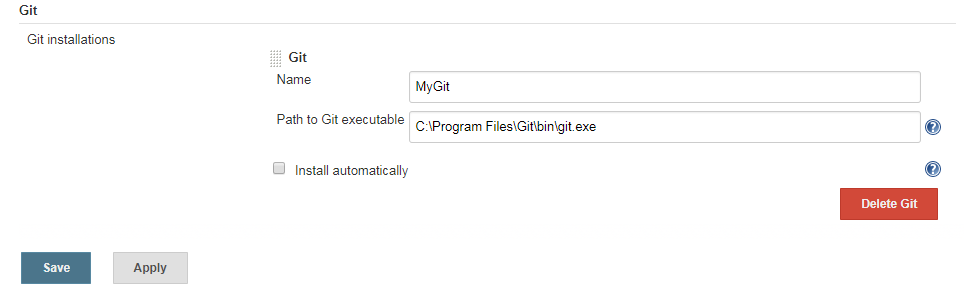
* Set up Maven as same as done in the previous step. Refer Fig 1.19



[Fig 1.19]

* Set up GIT configuration.

Configure the GIT installation path. Refer Fig 1.20



[Fig 1.20]

Guided Exercise 2: **Create a Maven Project in Jenkins**

**Estimated Completion Time:** 20 Minutes

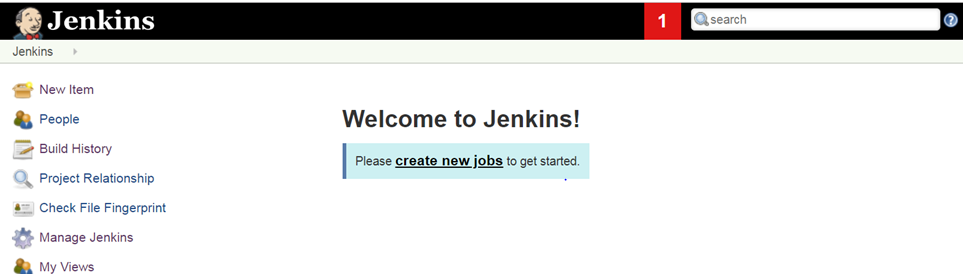
**Objective**: To create a Maven project in Jenkins which pulls the project build from Git, builds and tests the project

**Solution: Follow the given steps**

**Step 1:** Create a new Maven project in Jenkins.

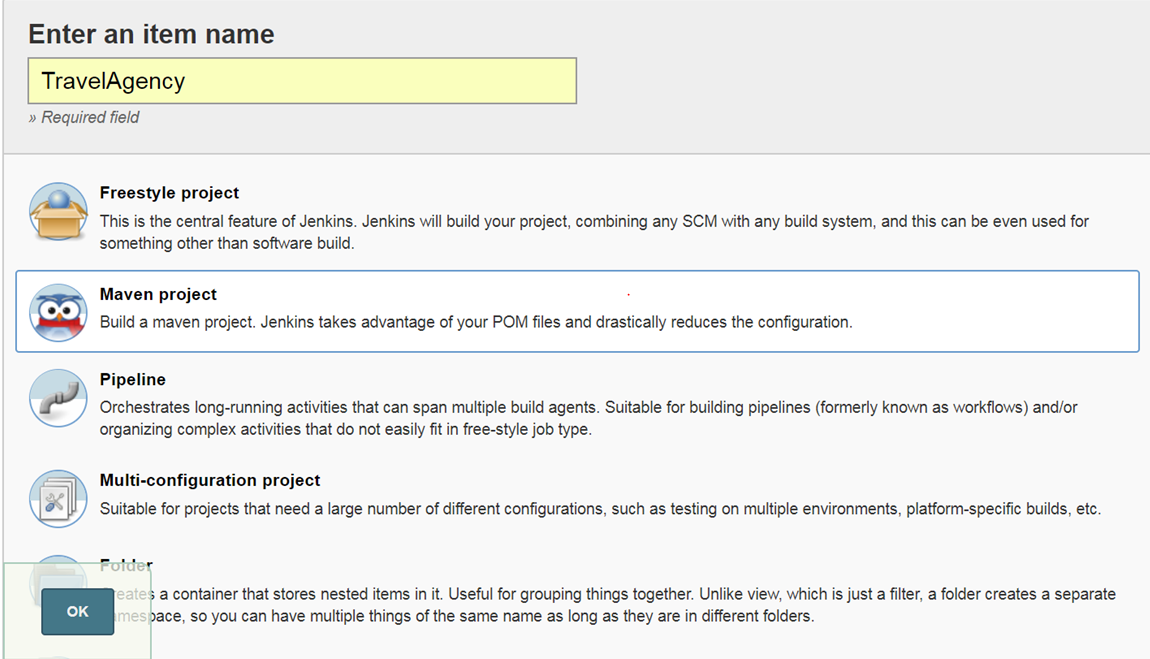
Follow the steps to create a new project.

* On Jenkins home page, select **New Item** or click **create new jobs**. Refer Fig 2.1



[Fig 2.1]

* Enter the item name as TravelAgency and select **Maven project** as the project template and click OK. Refer Fig 2.2



[Fig 2.2]

**Step 2:** Select TravelAgency from the Dashboard to configure Maven project. Refer Fig 2.3

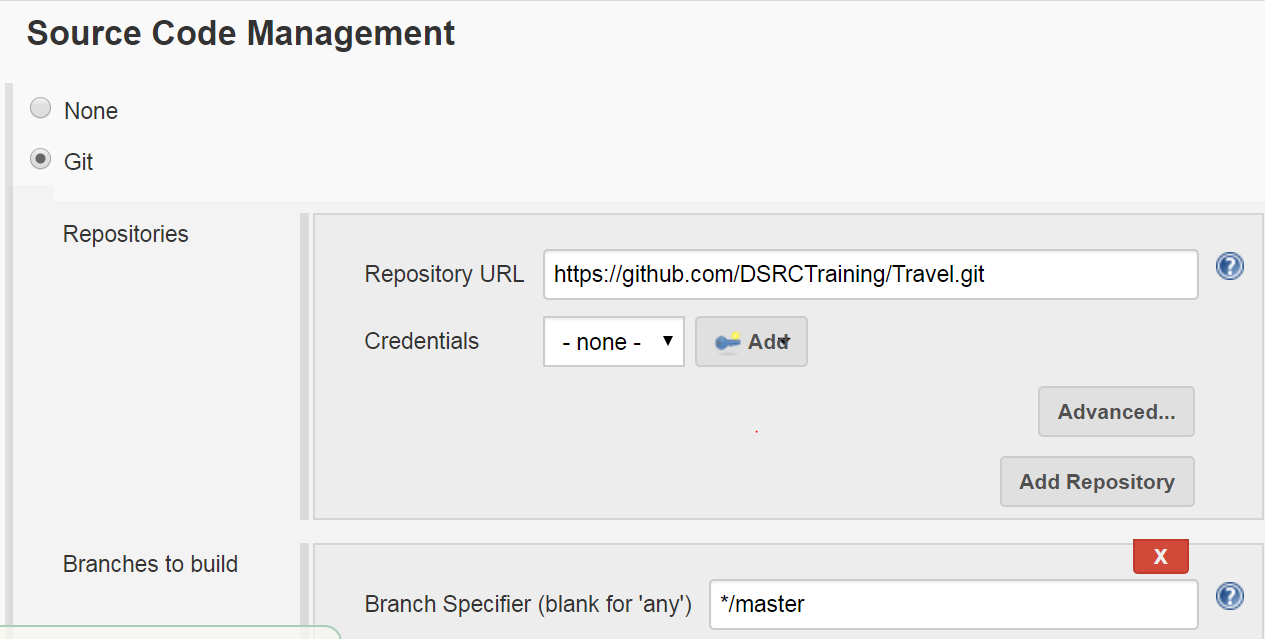


[Fig 2.3]

**Step 3:** Go to My Views and click TravelAgency and select configure.

In source code management option, click on Git radio button and specify the repository URL where the project is located in github. URL: <https://github.com/DSRCTraining/Travel.git>

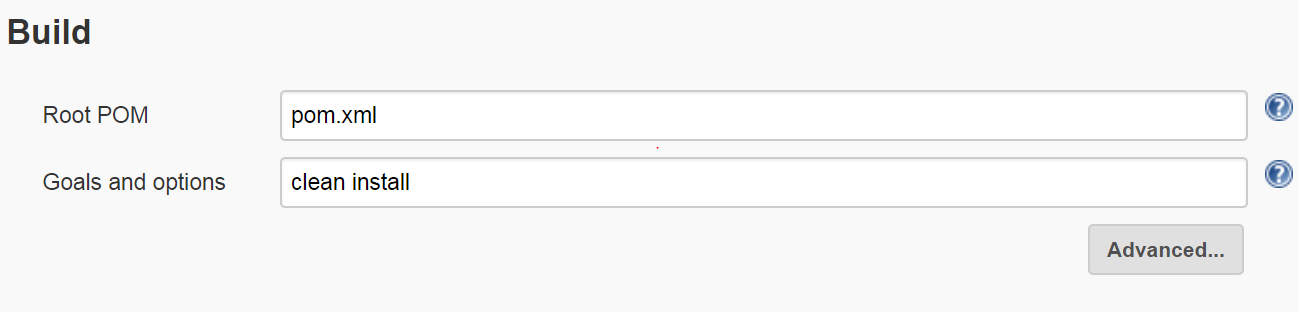
Note: The url given is a public repository and can be used for your testing. If you want to perform any branch operations, the branch name can be given inside **branches to build** section. Refer Fig 2.4



[Fig 2.4]

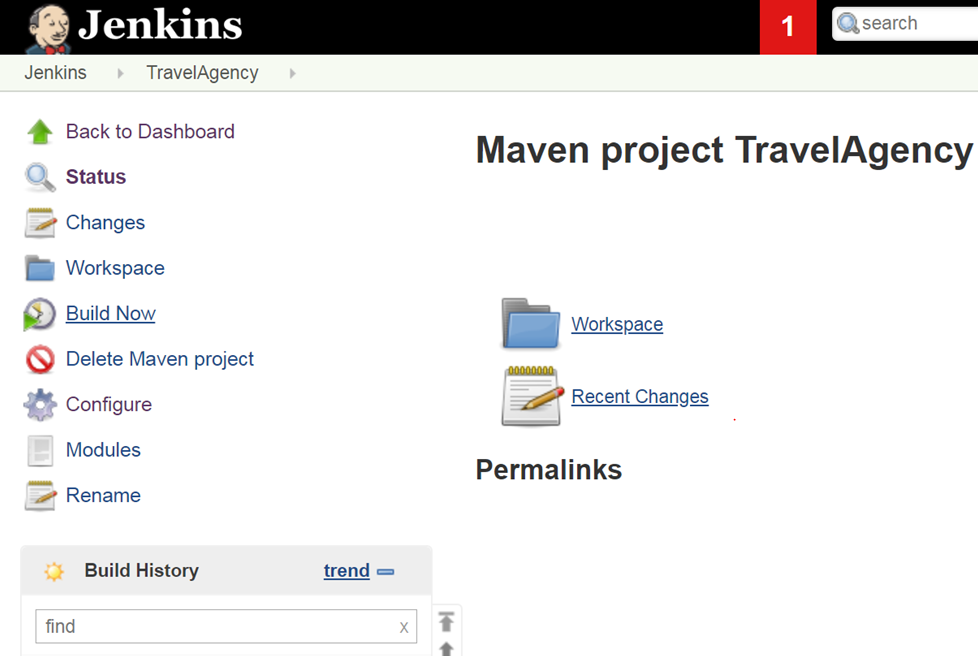
* Set the build step for the project. In Goals and options, give **clean and install**.

Refer Fig 2.5



[Fig 2.5]

* Save the settings and the page will get redirected to the project home page.
* To build the project, select **Build Now** option from the left panel as shown in Figure 2.6

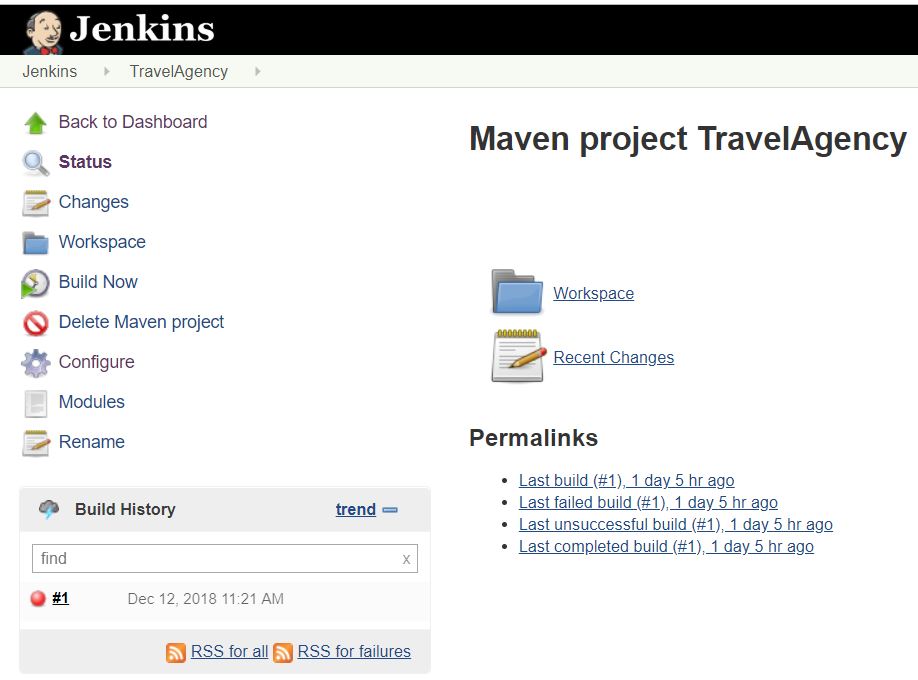


[Fig 2.6]

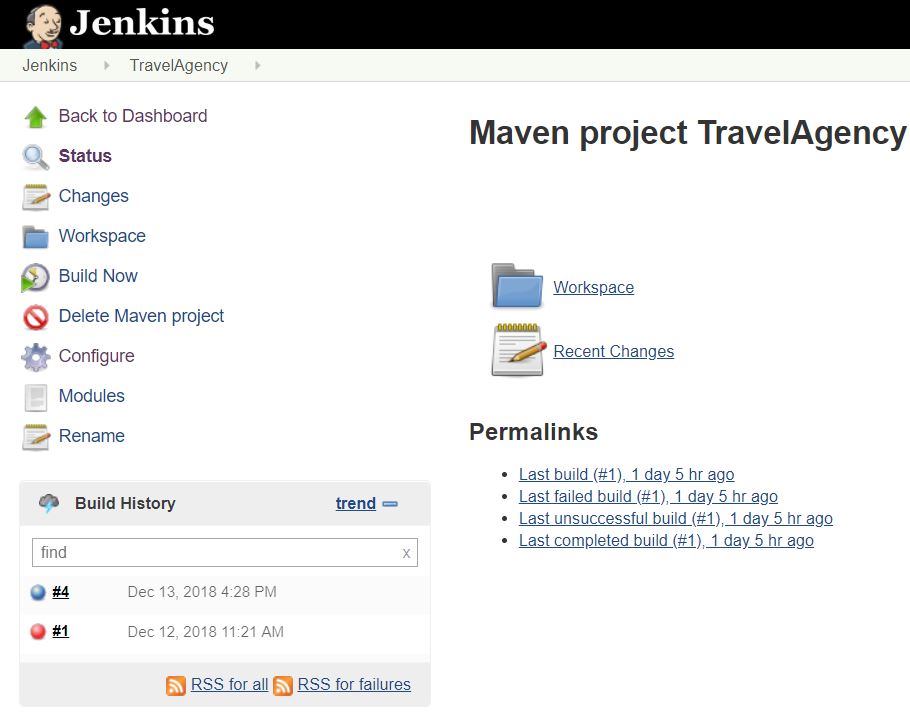
To view the build result in console click the icon in the build history. The icon colour indicates the build status.

**Red**- Build failure

**Blue**- Build Success

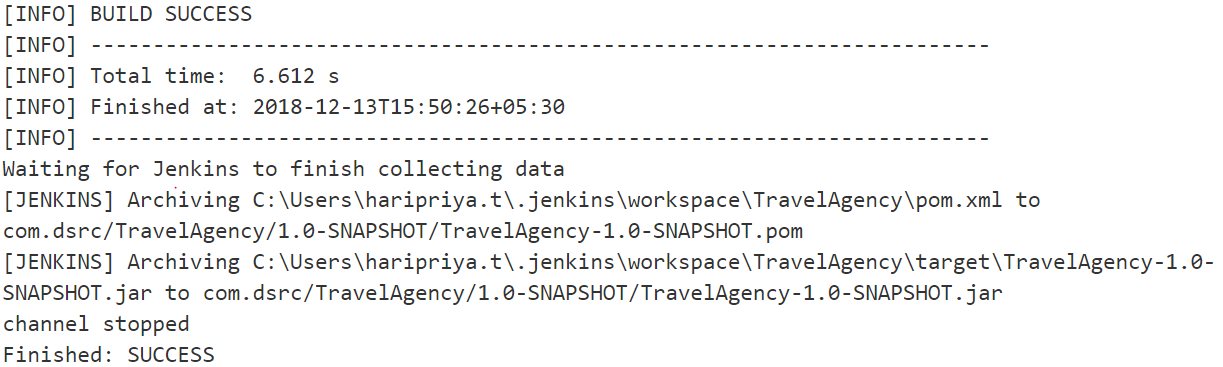


[Fig 2.7]



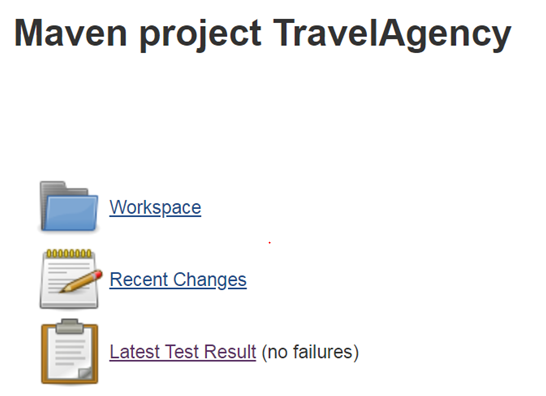
[Fig 2.8]

To view the detailed console output click on the build no and select Console Output . Your build result will be displayed as shown in Fig 2.9

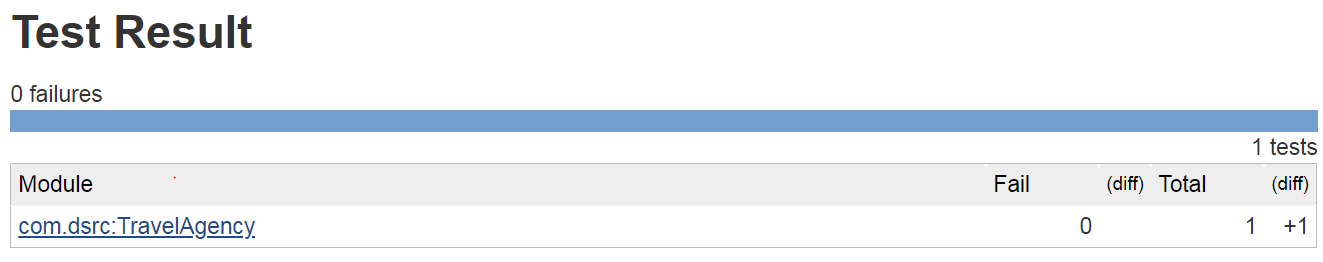


[Fig 2.9]

To view the test report of your project click on the Latest Test Result as shown in Fig 2.10 & 2.11



[Fig 2.10]



[Fig 2.11]

**Summary:**

You have learnt to install and configure Maven project in Jenkins