

Table of Contents

[1](#_Toc532577960)

[Table of Contents 2](#_Toc532577961)

[1 Lab Details: 3](#_Toc532577962)

[2 DevOps Tools: Guided Exercises 3](#_Toc532577963)

[2.1 Jenkins 3](#_Toc532577964)

[Guided Exercise 1: 4](#_Toc532577965)

[Guided Exercise 2: 11](#_Toc532577966)

# 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 hospital in India is working on their expansion plans and also planning to rebuild their existing website as an enterprise application on DevOps environment integrating with additional apps for various utilities. The hospital is having huge customers all over India and they are planning to provide apps that allows the users to check their wellness.

The entire application is planned to develop in DevOps environment which enables continuous integration and delivery. There are many developers working on it and the daily builds are 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 Freestyle Project in Jenkins
5. Configure Junit Reports in Jenkins
6. Build the Application on Jenkins
7. Verify the unit test reports

**Pre- Requisites**

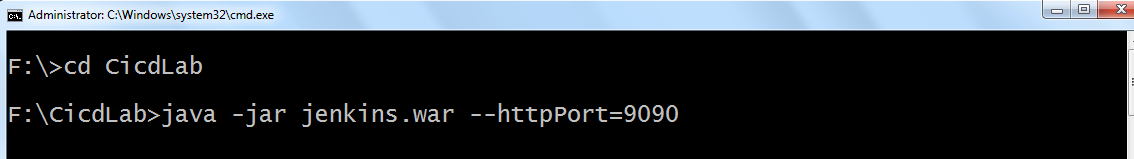
* Maven 3.5
* Jenkins
* Git

Guided Exercise 1: **Setup Jenkins**

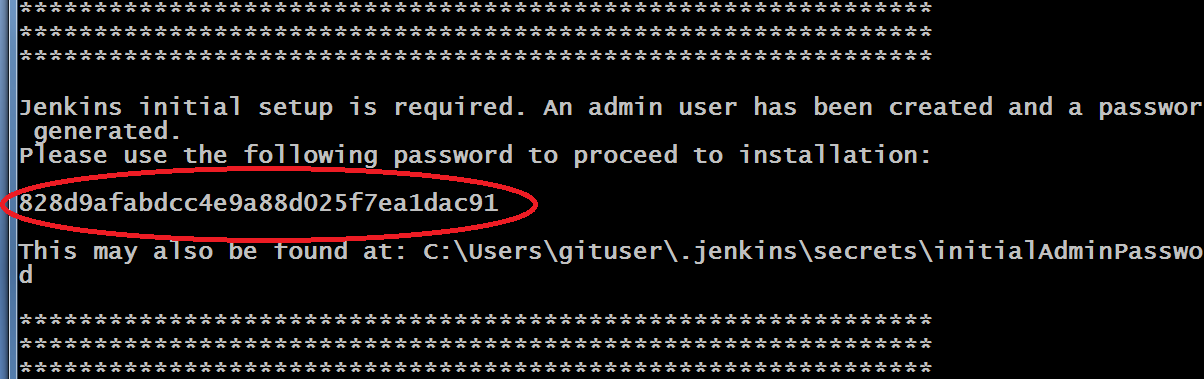
* + Verify the jenkins.war file is present in the path C:\Program Files (x86)\Jenkins
  + Open command prompt and navigate to the Jenkins folder mentioned above.
  + Start Jenkins on port 9090 by executing the below java command in the command prompt (default port is 8080) Result is shown in [Fig 1.9-2.1]

java –jar jenkins.war --httpPort=9090

* As part of the initial setup, observe the logs to get the admin password.

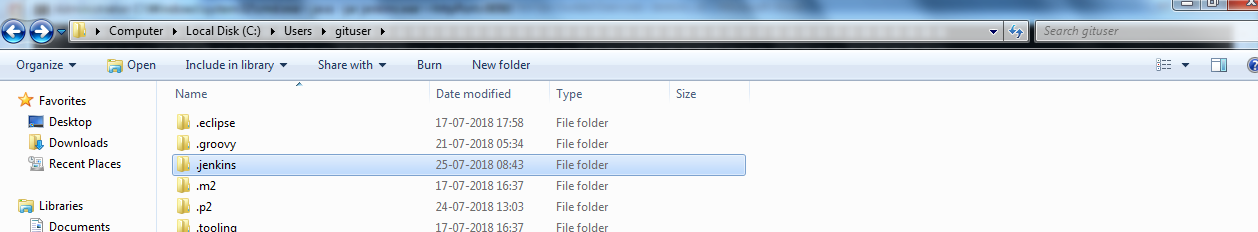


[Fig 1.9]



[Fig 2.1]

* Jenkins creates folder (.jenkins) in users profile (c:\users\username\.jenkins) Ref[Fig 2.2]



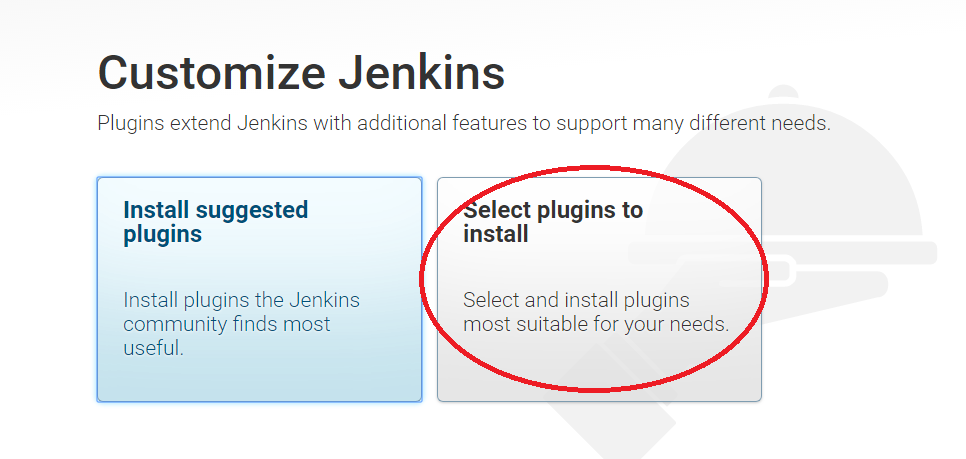
[Fig 2.2]

* Admin password is stored in initialAdminPassword file in users profile

C:\Users\username\.jenkins\secrets\initialAdminPassword

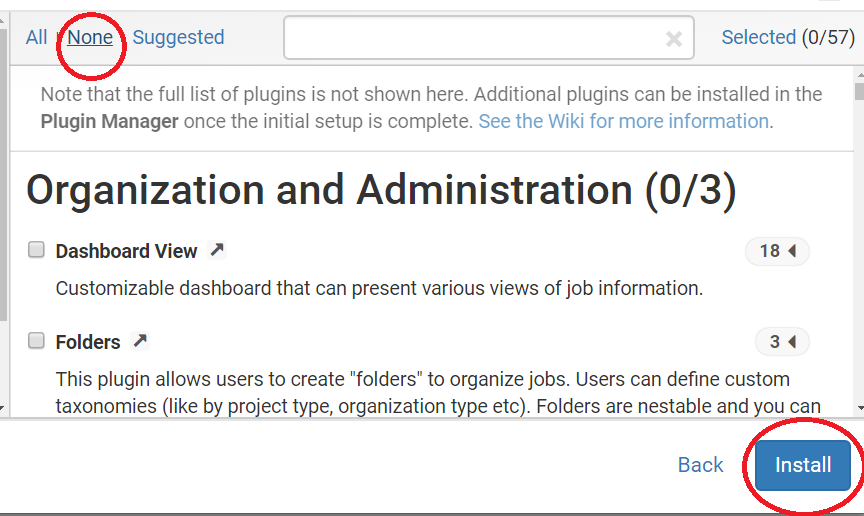
**Step 9: Set up the Jenkins Environment**

* Open <http://localhost:9090> in a web browser.
* Enter the admin password received as part of the initial setup, to Unlock Jenkins.
* Next, click on Select plugins to install [Reference [Fig 2.3 ]



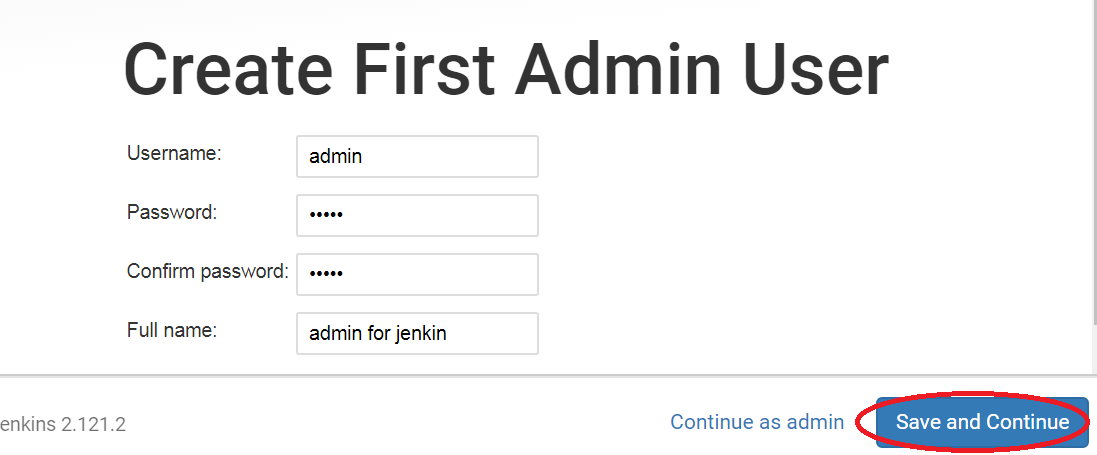
[Fig 2.3]

* Select None and then click on Install [ Reference [Fig 2.4 ]

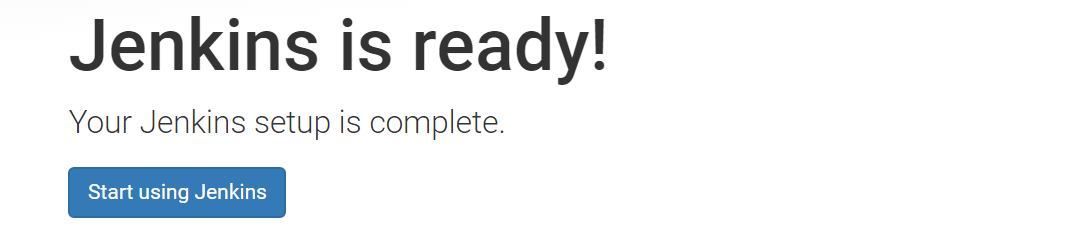


[Fig 2.4]

* Create the First Admin User account by specifying a username and password (Example : username – admin, password – admin)
* Click on Save and Continue [ Reference [Fig 2.5 - 2.6] ]

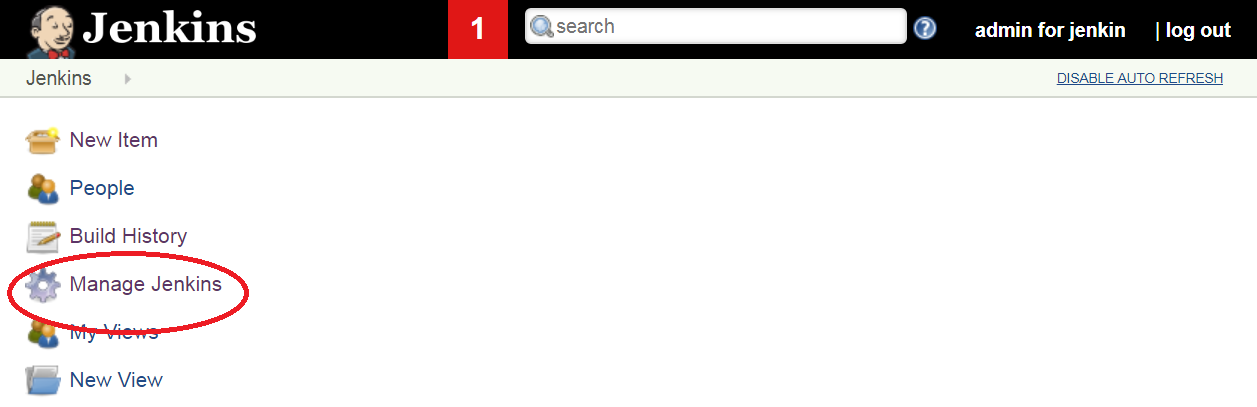


[Fig 2.5]



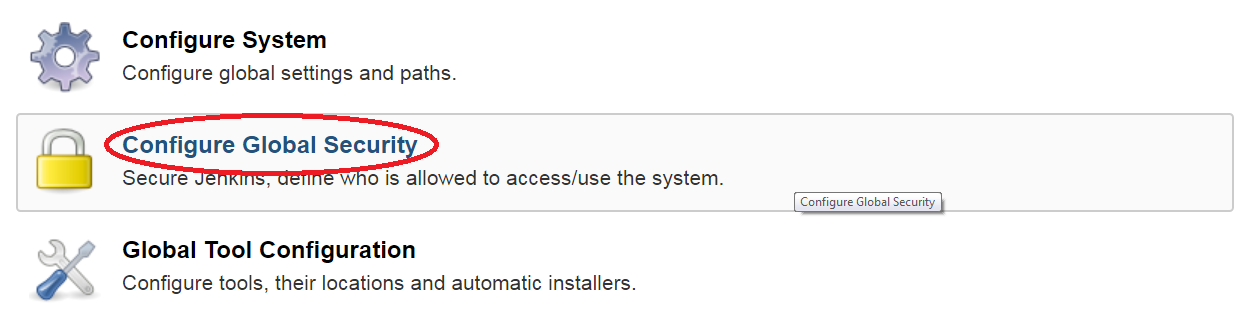
[Fig 2.6]

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 Figure 1.11 and 1.12



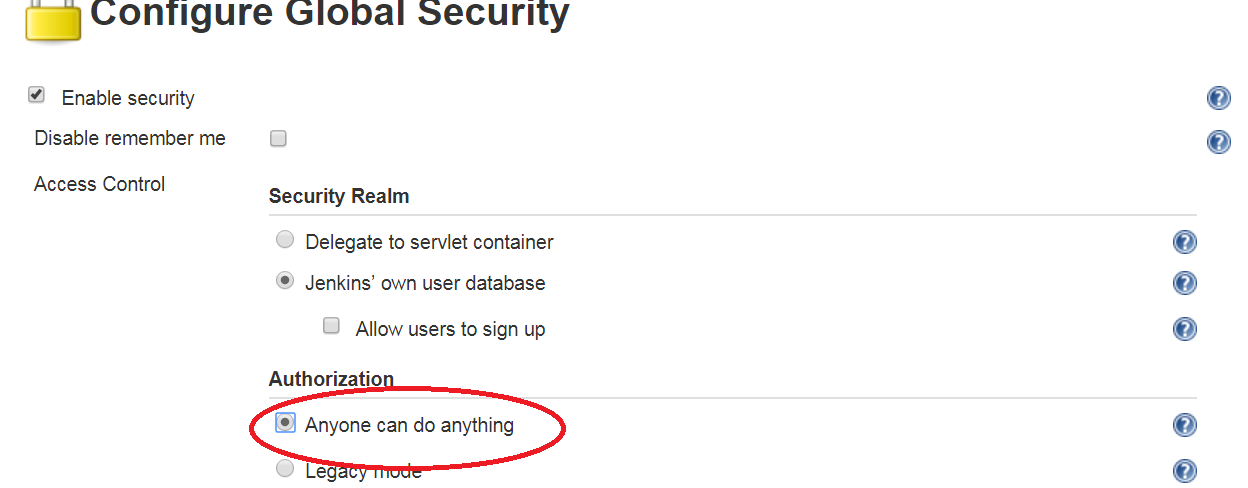
[Fig 1.11]

* 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]

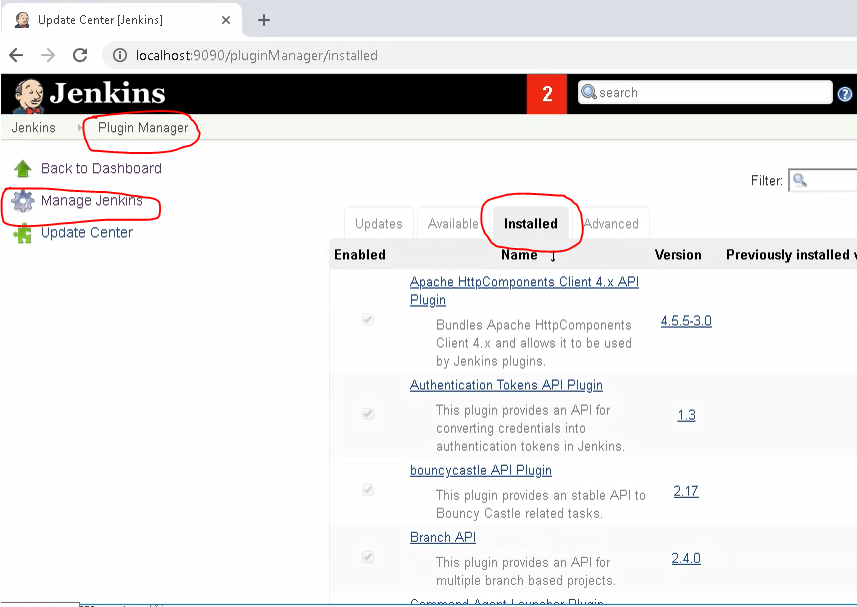
**Step 3** **Install the required plugins.**

*Here, we will learn how to manually install plugins in Jenkins without an active internet connection*

1. If your Jenkins instance is already running, please stop the process first. You can do this by hitting Ctrl+C in the Windows Command Prompt window where Jenkins was started.
2. Copy all the contents from C:\Program Files (x86)\Jenkins\plugins into C:\Users\<your-id>\**.**jenkins\plugins
3. Restart Jenkins, by executing in the Windows Command Prompt, the same java command as before:

**java –jar Jenkins.war –httpPort=9090**

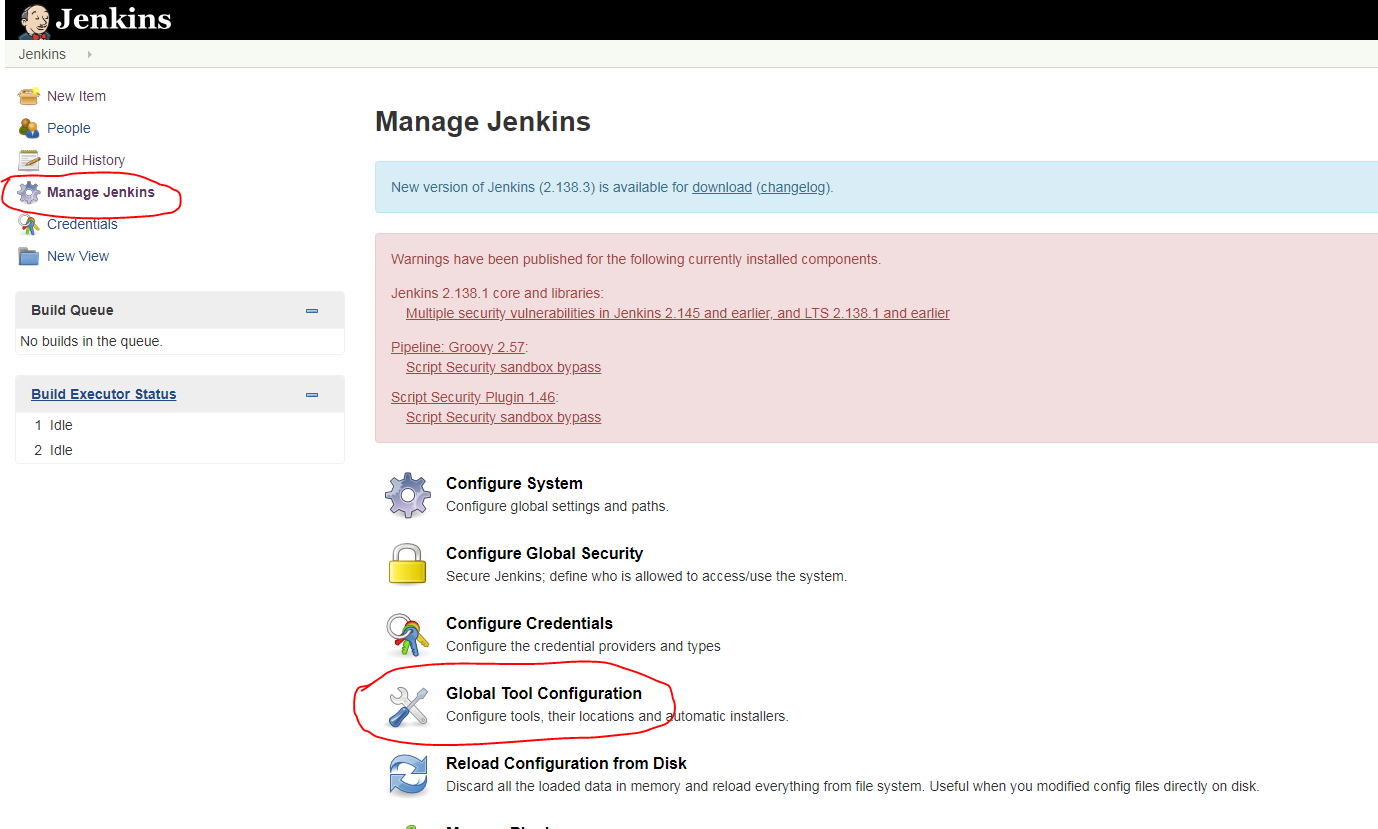
1. In a browser, navigate to the URL, <http://localhost:9090>
2. Click on Manage Jenkins -> Manage Plugins -> Installed
3. Here you will see a list of all the installed plugins.



**Step 2: Configure the installations inside Jenkins.**

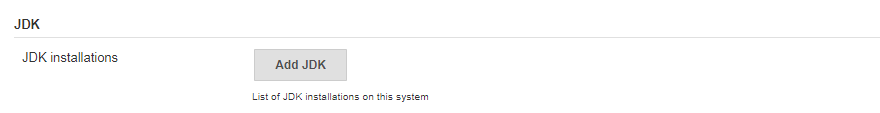
The installed plugins and build environment configurations are as follows.

1. Navigate to Manage Jenkins - > Global Tool Configuration. Refer Fig 3.9



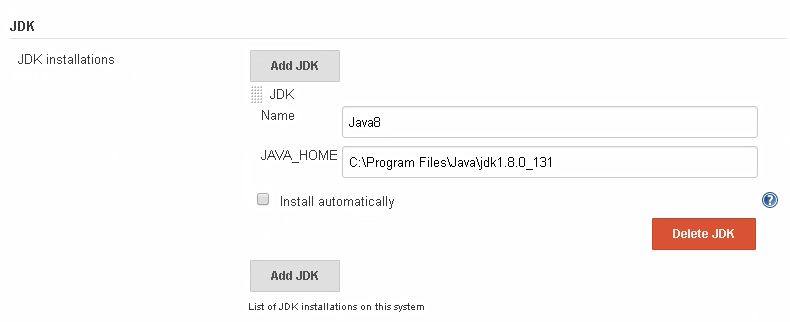
[Fig 3.9]

2. Set up JDK. Click on JDK installations -> Add JDK. Refer Fig 3.10



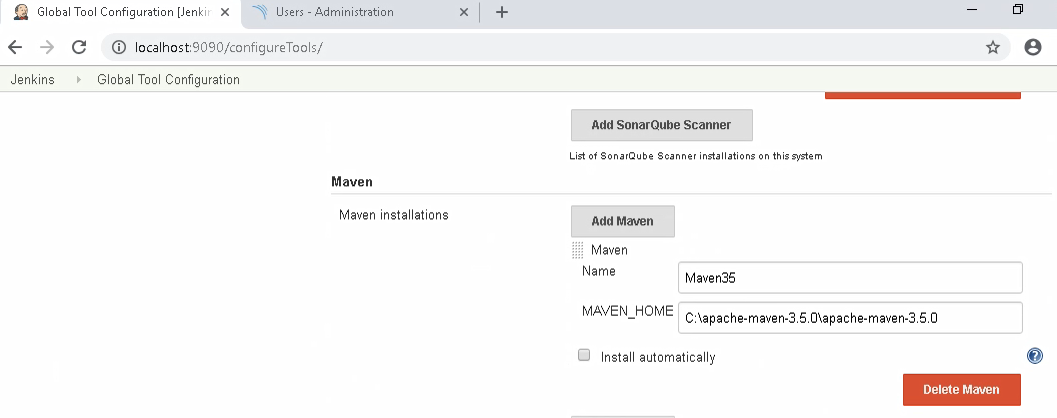
[Fig 3.10]

Specify the JDK Home path as per your JDK Installation. Give a name and uncheck “**install automatically**” .Refer Fig 4.0



[Fig 4.0]

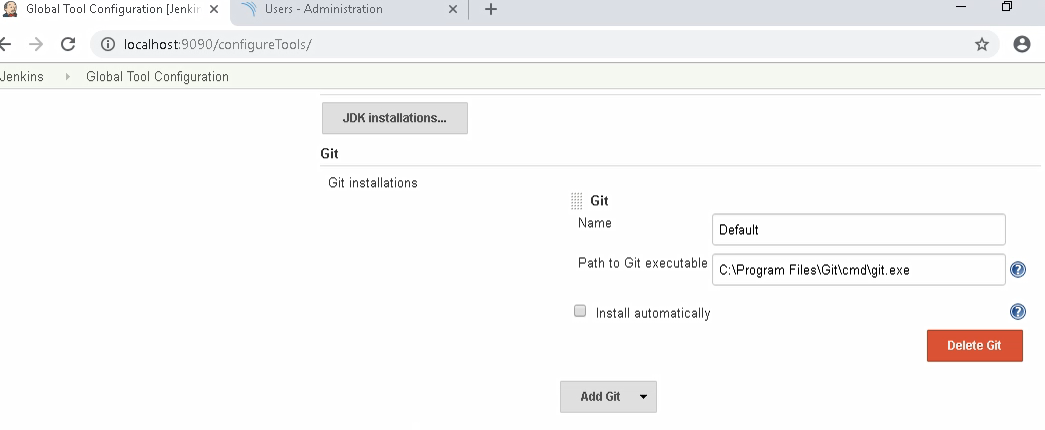
3. Set up Maven as same as done in the previous step. Refer Fig 4.1



[Fig 4.1]

4. Set up GIT.

Configure the GIT installation path. Refer Fig 4.2



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

**Estimated Completion Time:** 20 Minutes

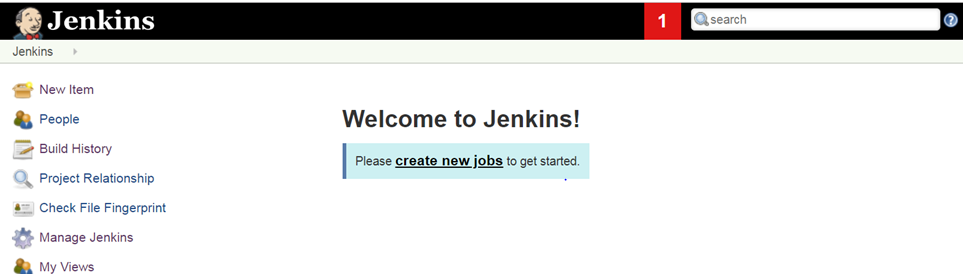
**Objective**: To create a freestyle project in Jenkins which pulls the project build from Git, builds and tests the project.

**Solution: Follow the given steps**

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

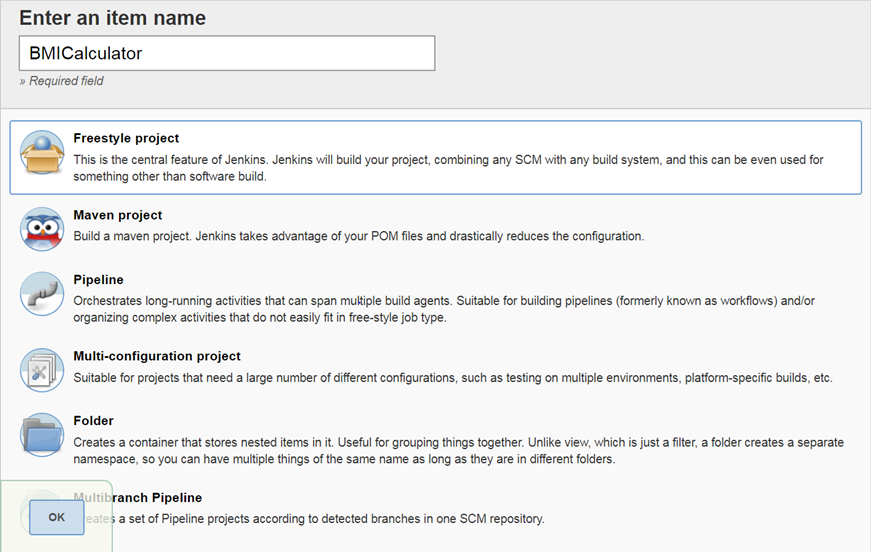
Follow the steps to create a new project.

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



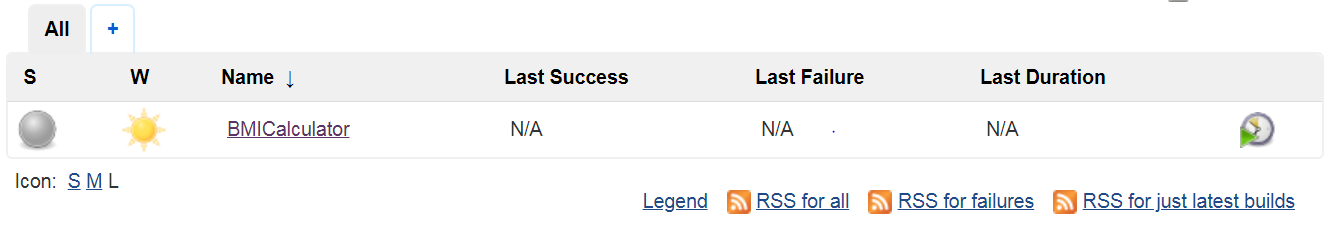
[Fig 2.1]

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



[Fig 2.2]

**Step 2:** Select BMICalculator from the Dashboard to configure freestyle project. Refer Fig 2.3

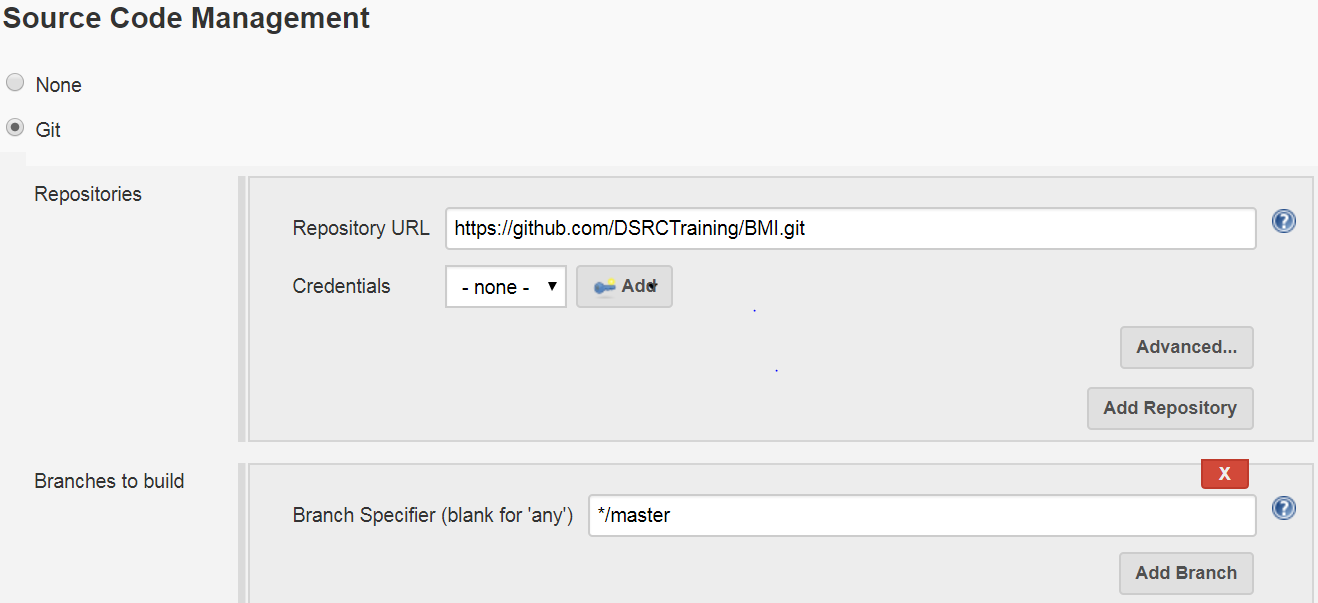


[Fig 2.3]

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

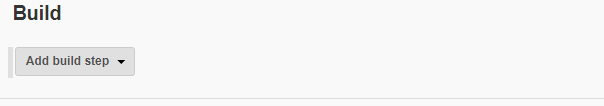
In source code management option, click on Git radio button and specify the repository URL where the project is located in github. URL: <https://github.com/DSRCTraining/BMI.git>

Note: The url given is a public repository and can be used for your testing. You can clone this repo and save the code in your personal repo at <https://code.cognizant.com>. If you want to perform any branch operations, the branch name can be given inside **branches to build** section. Refer Fig 2.4

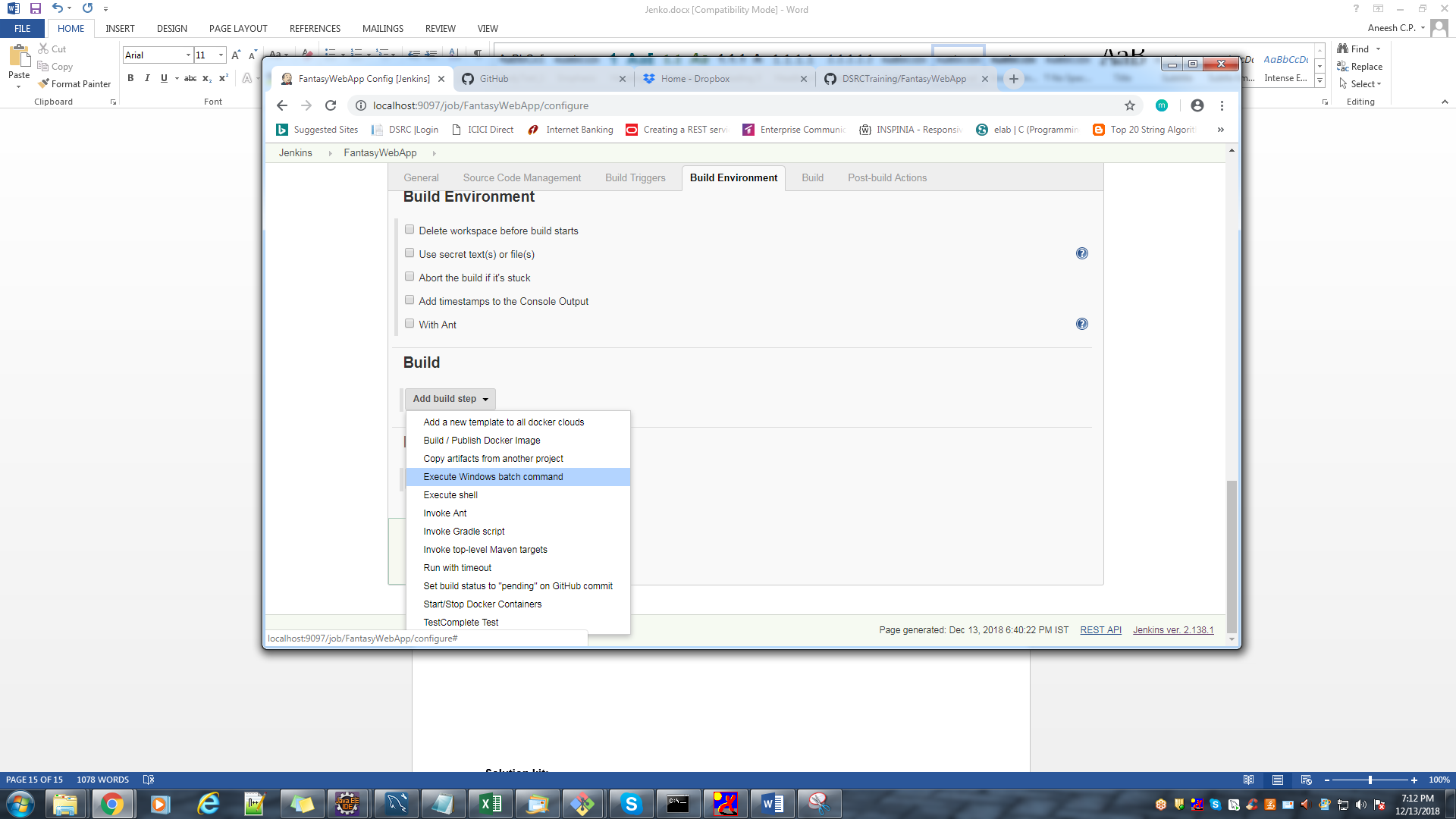
[Fig 2.4]

**Step 4:**  Specify the build options. Any build commands given in this section will get executed after the project is pulled from the repository.

* Navigate to build section and select **Execute Windows Batch Command** option. Refer Figure 2.5 and 2.6.

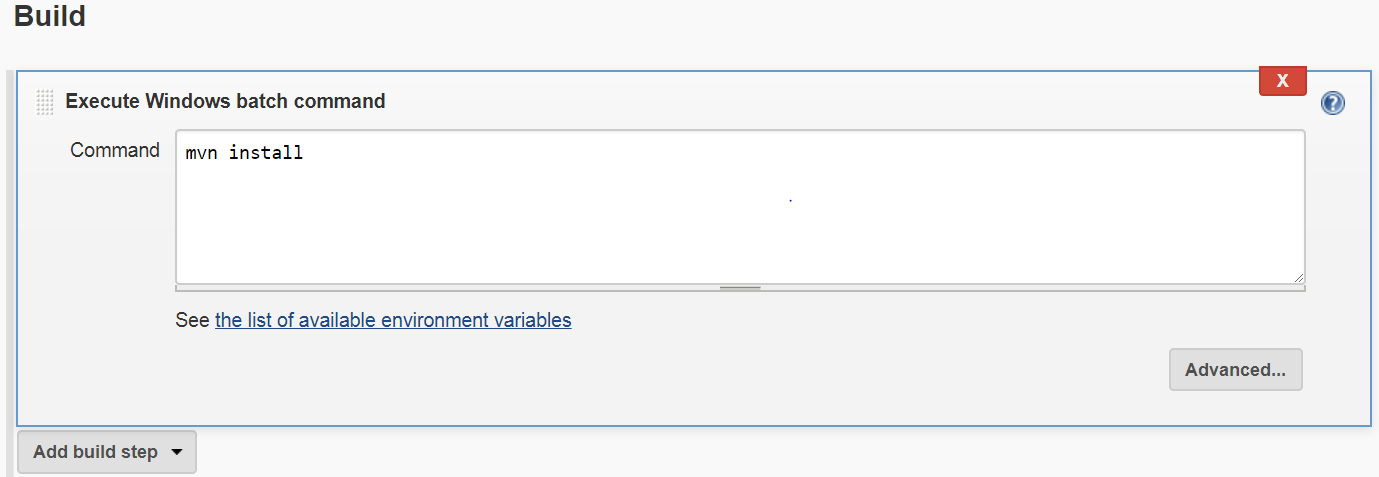


[Fig 2.5]



[Fig 2.6]

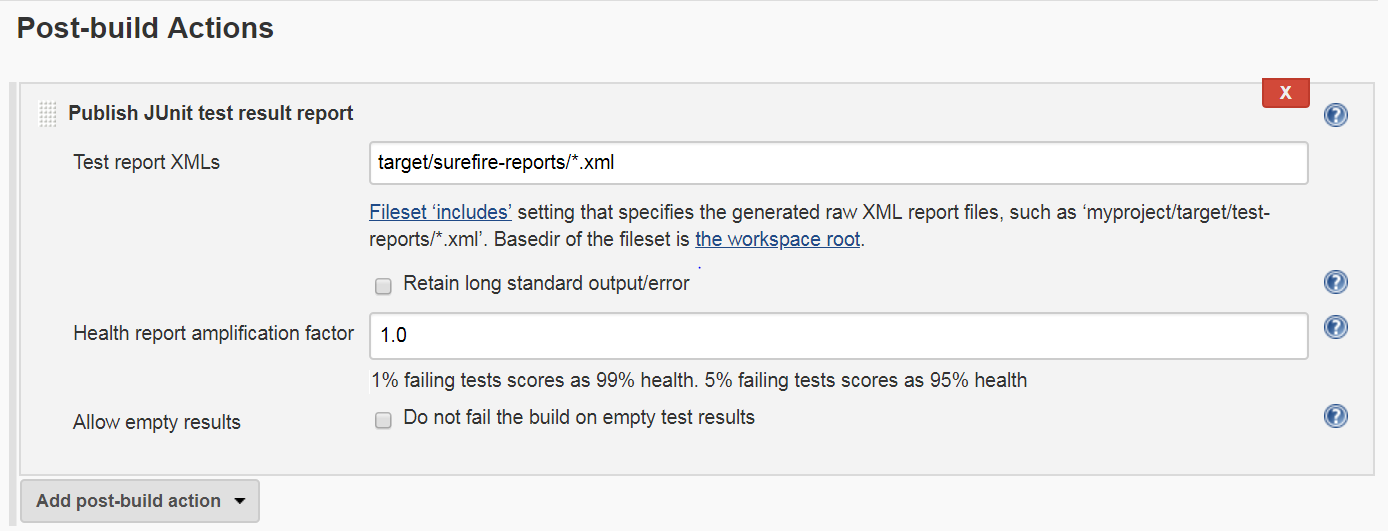
* Specify the command to be executed on build section. In this example mvn install command is given. Refer Figure 2.7



[Fig 2.7]

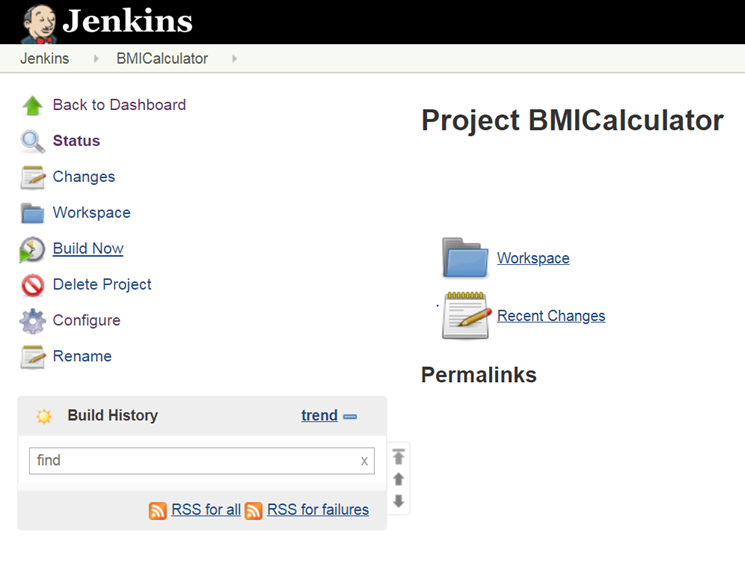
**Step 5:** Set the Post Build action for generating test reports. Click **Add post build actions** and select **Publish Junit test result report.**

In Test report xmls specify the path of surefire report folder. Refer Fig 2.8



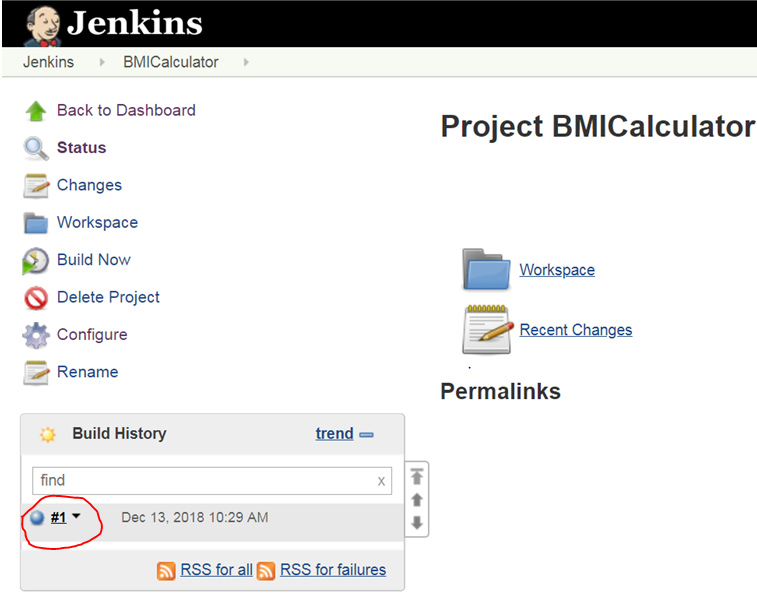
[Fig 2.8]

* 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 2.9



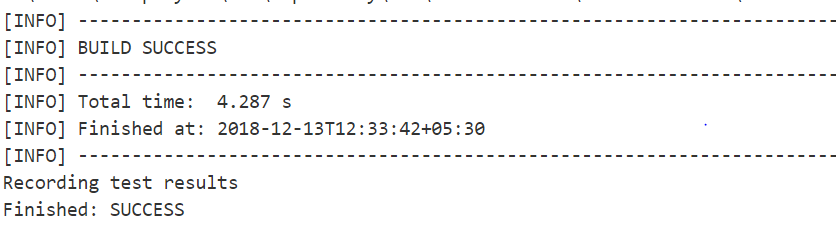
[Fig 2.9]

* The build progress and build status will get displayed on the Build History panel as shown in Figure 2.10.



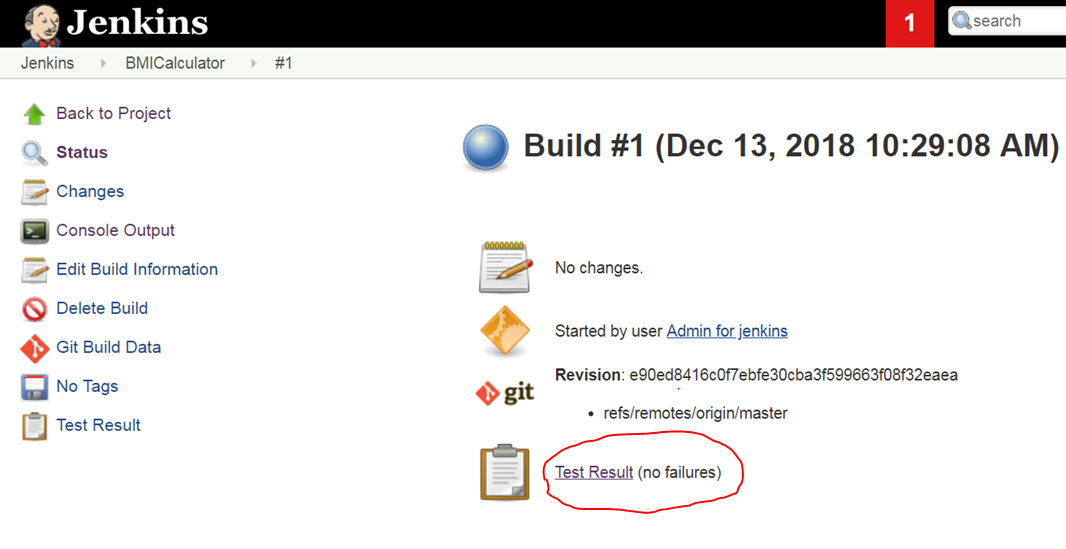
[Fig 2.10]

If the build is success, the bubble will get displayed in blue color else in red color. 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 2.11

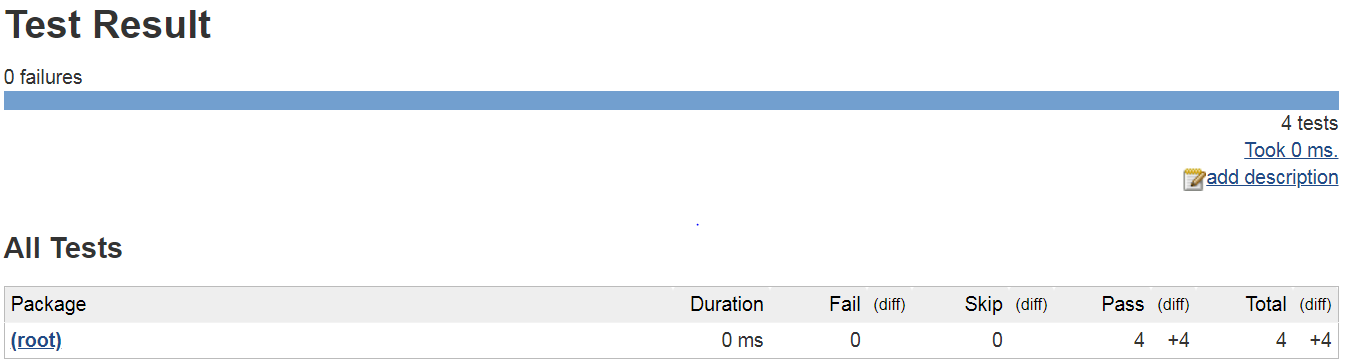


[Fig 2.11]

To view the test report of your project click on TestResult option as shown in Fig 2.12 & 2.13



[Fig 2.12]



[Fig 2.13]

**Summary:**

You have learnt to manage Jenkins in DevOps environment.