Table of Contents

[1](#_Toc518666382)

[Table of Contents 2](#_Toc518666383)

[1 Lab Details: 3](#_Toc518666384)

[2 DevOps Tools: Guided Exercises 3](#_Toc518666385)

[2.2 CI/CD pipeline 3](#_Toc518666386)

[Guided Exercise 1 4](#_Toc518666387)

[Guided Exercise 2 7](#_Toc518666388)

[Guided Exercise 3 10](#_Toc518666389)

# 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 –Problem Statement

A Leading Sport Promoters Company in India is planning to develop a web based fantasy sports platform that allows users to play fantasy cricket, football, kabaddi and basketball games with their statistical skills and game knowledge. They have introduced this platform as a mobile application initially and now planning to create a web application on the same.

There are lakhs of fans and players for this game and the company is planning to develop and deploy the project in DevOps environment. There are many developers working on this project and their daily builds shall be updated to the SCM repository. The Jenkins Server is present in the DevOps environment for automated build process.

**Scope:**

You have been assigned the task of managing the Jenkins to enable automated build on the projects uploaded to SCM repository. You need to perform the following tasks.

**Tasks:**

* Get the project builds from the GIT repository
* Automate project Build process
* Automate Test cases execution
* Deploy the application on Tomcat Server

**Steps*:***

1. Install and configure Jenkins
2. Install required plugins in Jenkins
3. Jenkins configuration with Maven and Git
4. Configure Junit Reports in Jenkins
5. Configuring automated deployment in Jenkins

**Pre- Requisites**

* Maven 3.5
* Tomcat 7
* Jenkins
* Git

Guided Exercise 1**:** **Install Jenkins**

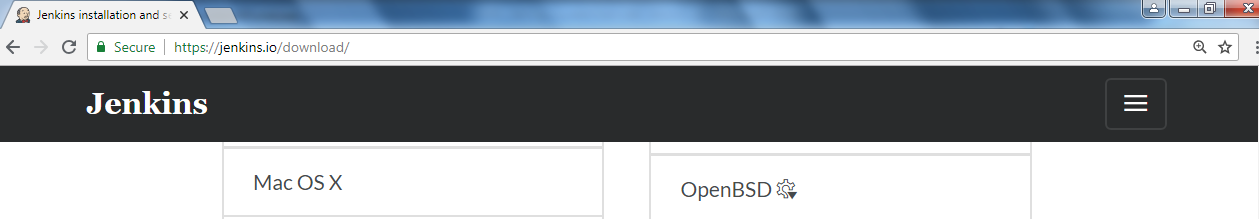
**Estimated Completion Time:** 10 Minutes

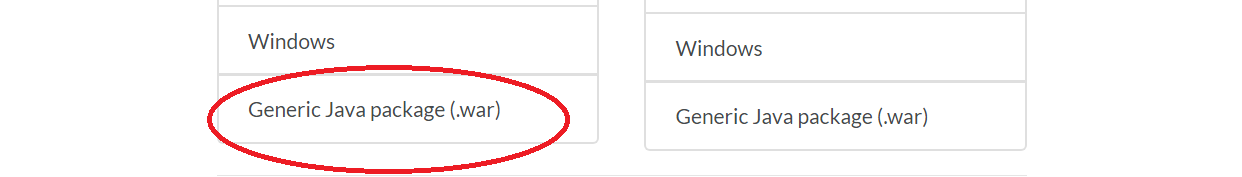
**Objective**: To Install and configure Jenkins on Windows Environment.

**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://maven.apache.org/download.cgi) as shown in Figure 1.1

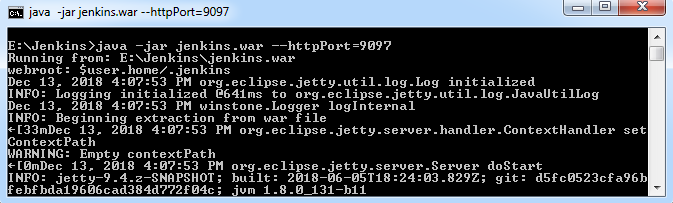




[Fig 1.1]

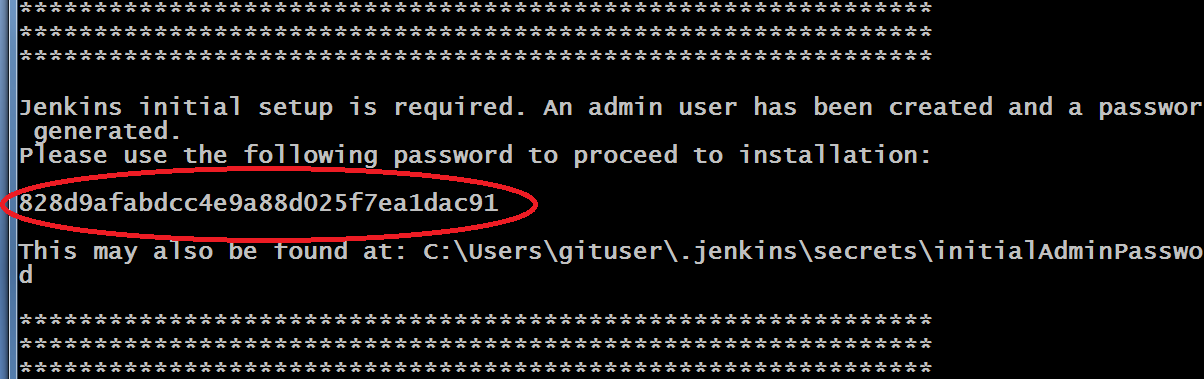
* + Copy the Jenkins.war to any folder. **E:\Jenkins** is the path used in this example.
  + Start Jenkins (default port is 8080). To start Jenkins in a port other than the default port, use –httpPort option.
  + Open command prompt
  + Navigate to the directory where jenkins.war is copied.
  + Use **java –jar jenkins.war –httpPort=9097** command to start Jenkins.

Refer Figure 1.2 and 1.3.



[Fig 1.2]

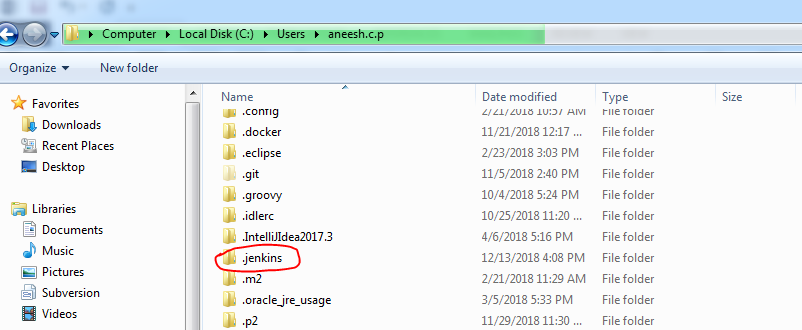
**Jenkins installation creates admin password:**



[Fig 1.3]

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

Refer Figure 1.4

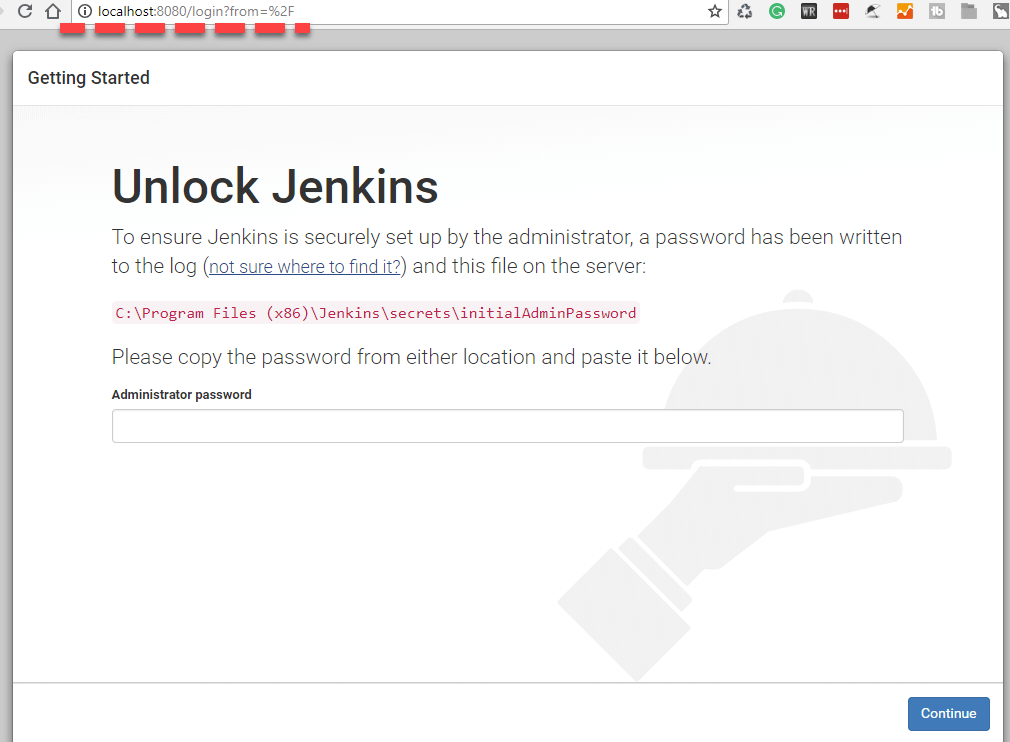


[Fig 1.4]

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

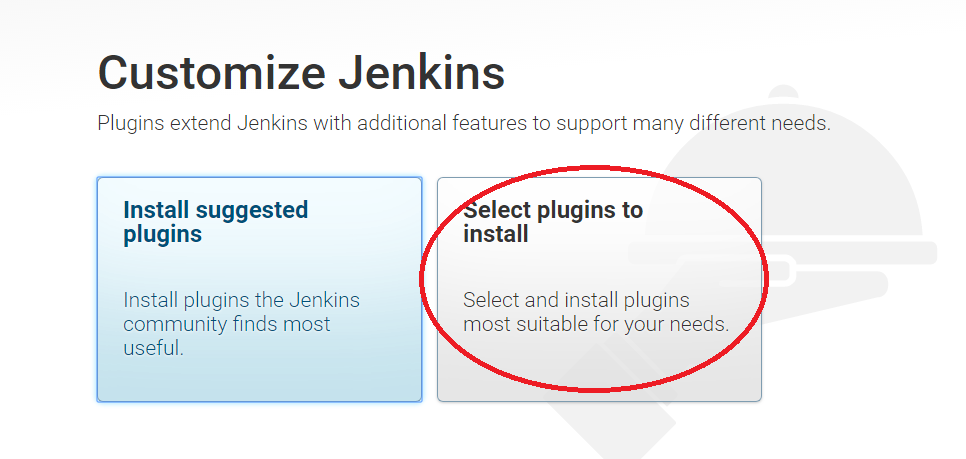
**Step 2: Set up the Jenkins Environment**

* Open the web browser and use the URL: <http://localhost:9097> to open the Jenkins web page. First time the page ask for admin password to unlock it as shown in Figure 1.5



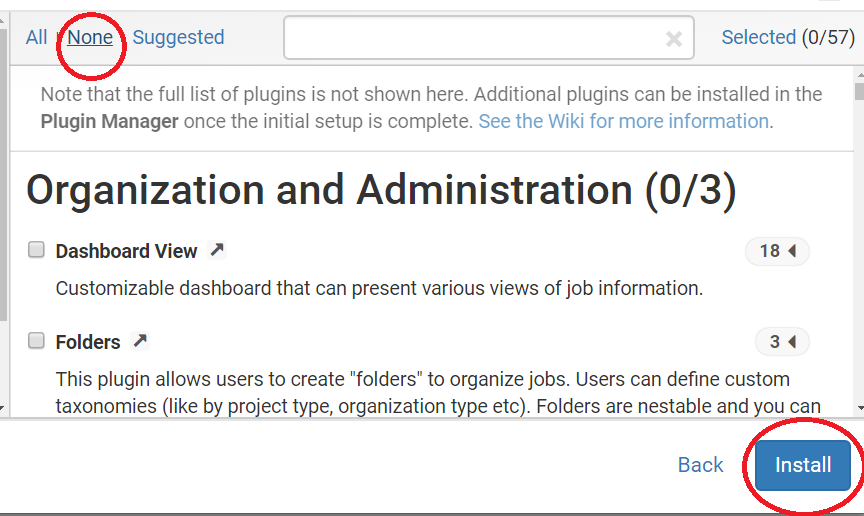
[Fig 1.5]

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



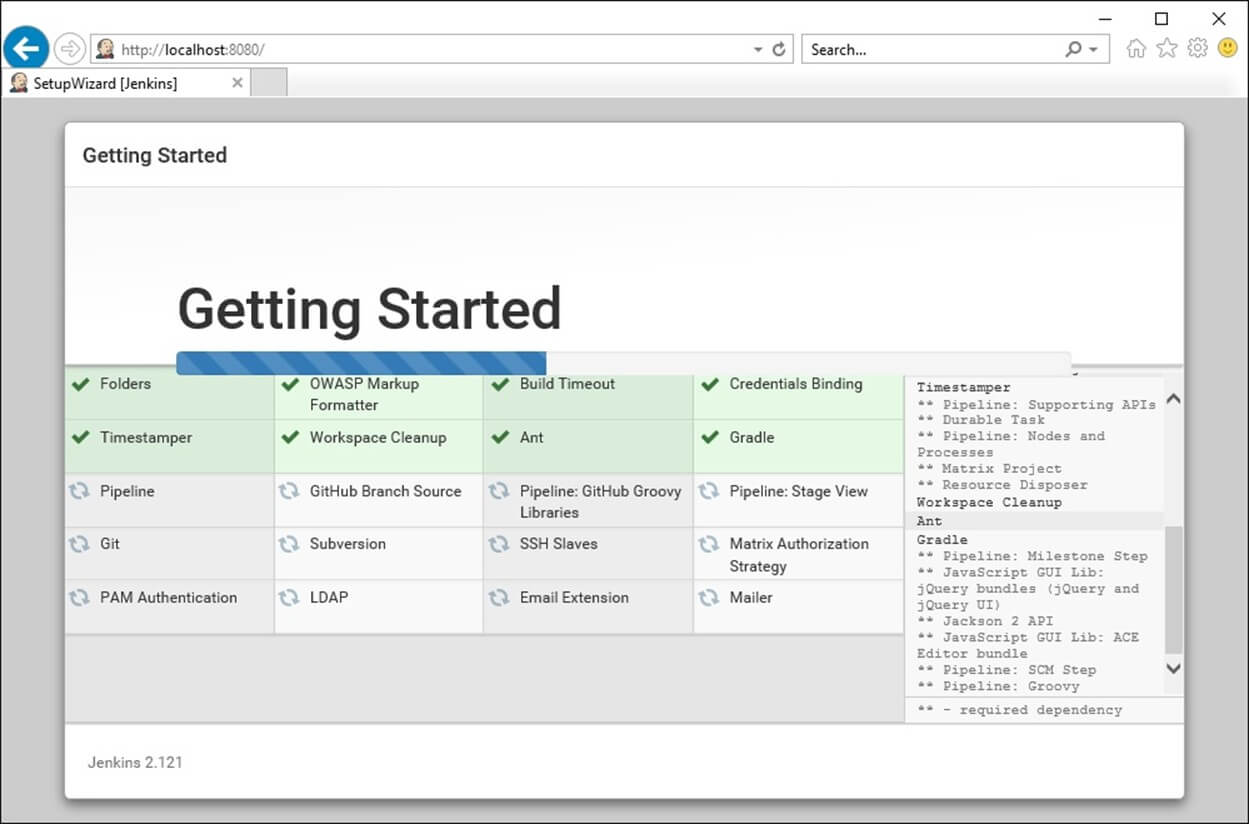
[Fig 1.6]

* Select None (click install ) Reference [Fig 1.7 ]



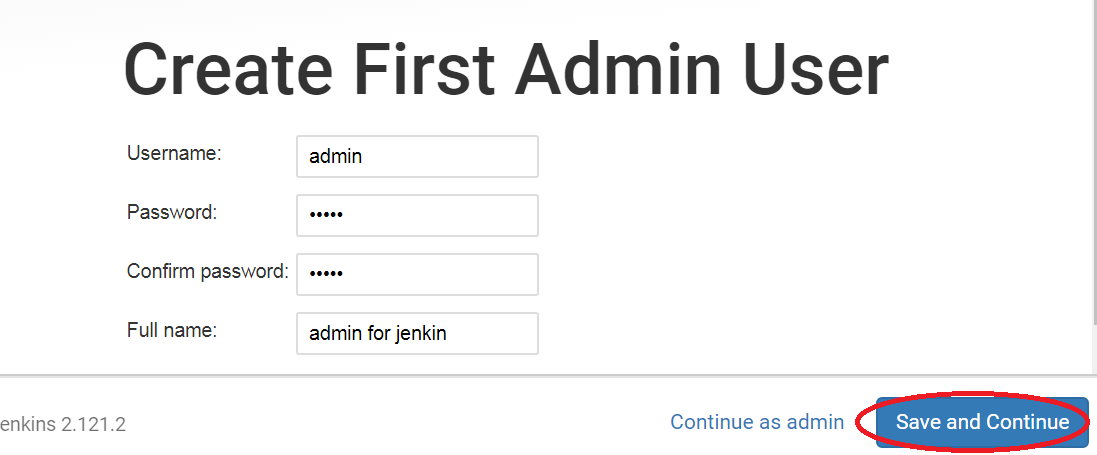
[Fig 1.7]

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



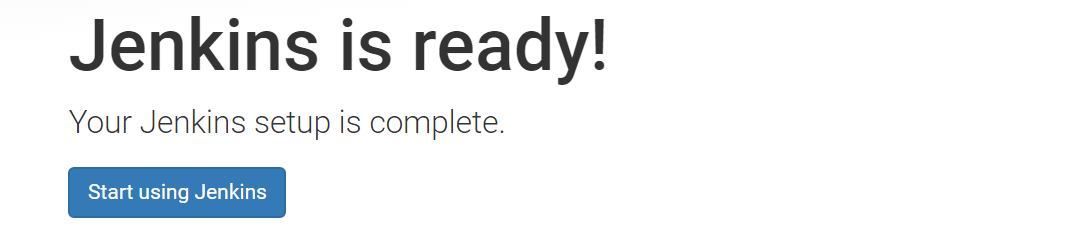
[Fig 1.8]

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



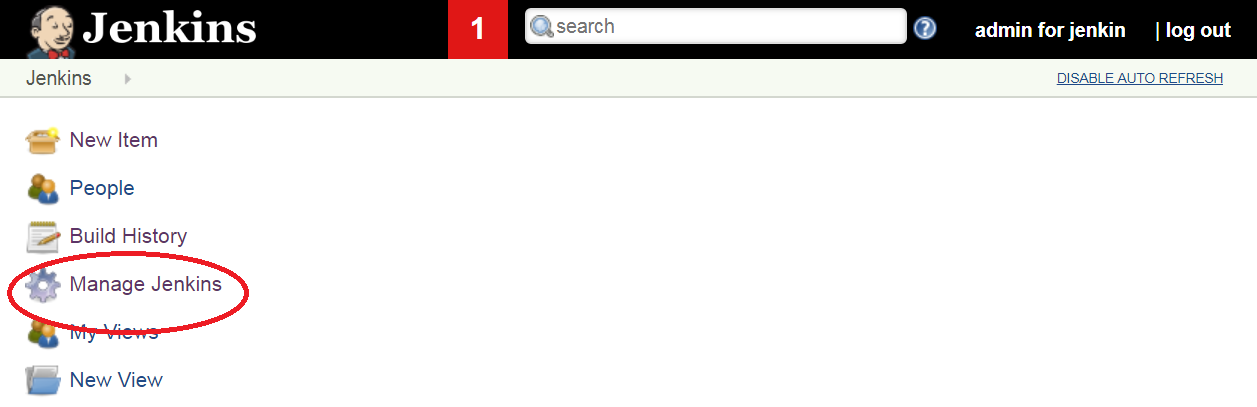
[Fig 1.9]

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



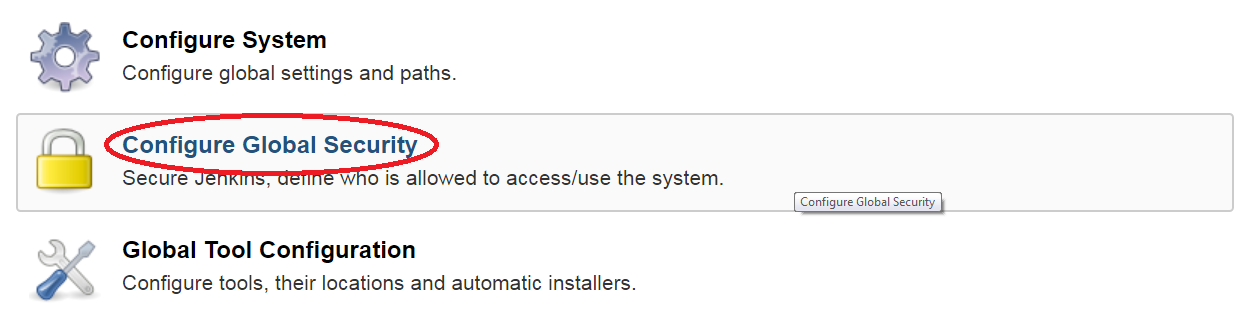
[Fig 1.10]

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 Figure 1.11 and 1.12



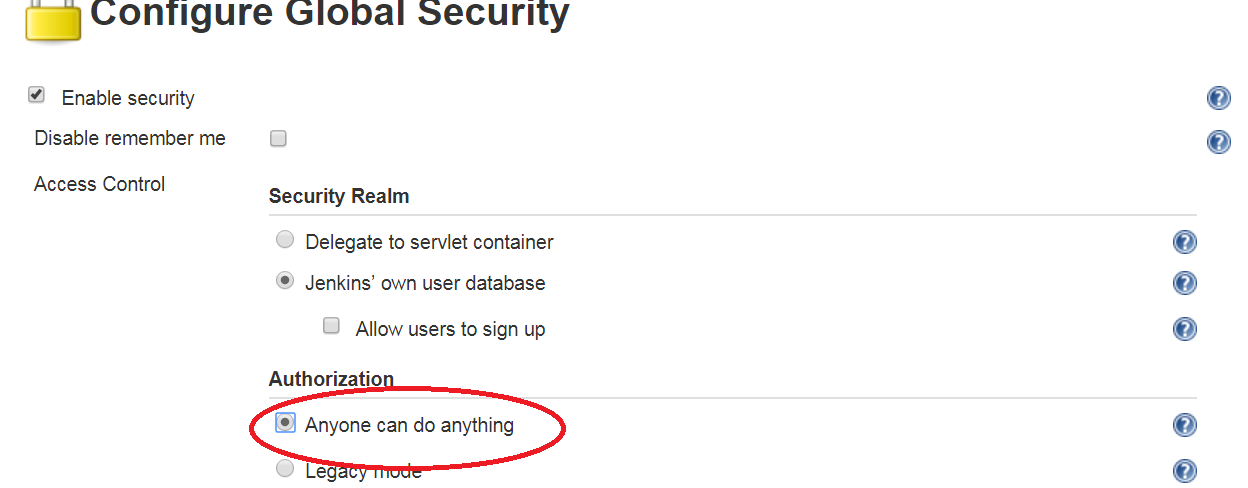
[Fig 1.11]

* Select Configure Global Security

****

[Fig 1.12]

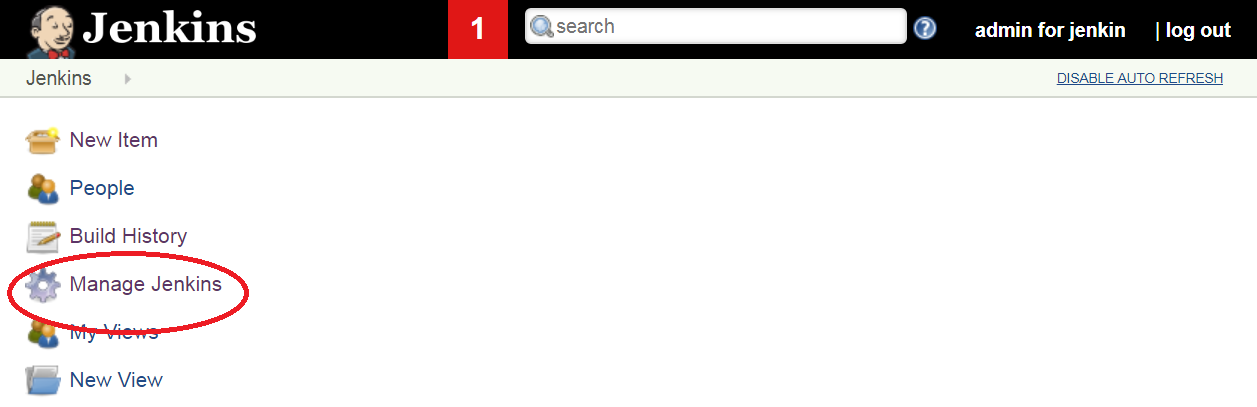
* Select option ‘Anyone can do anything’ (apply and save the setting). Refer Figure 1.13

****

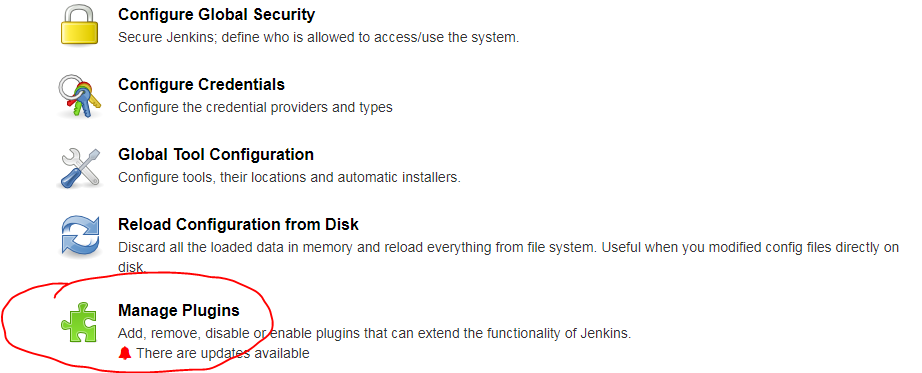
[Fig 1.13]

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

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

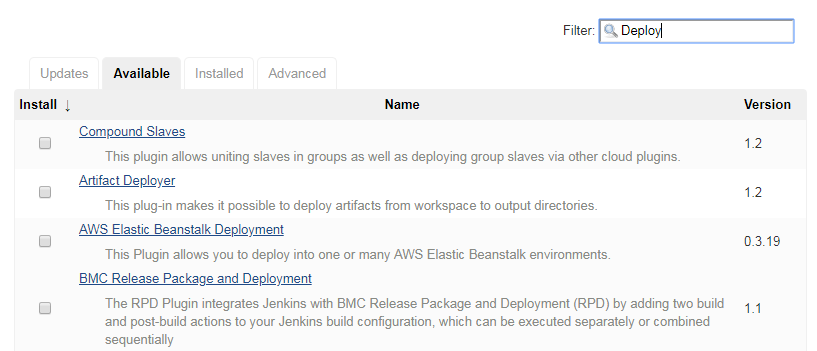


[Fig 1.14]



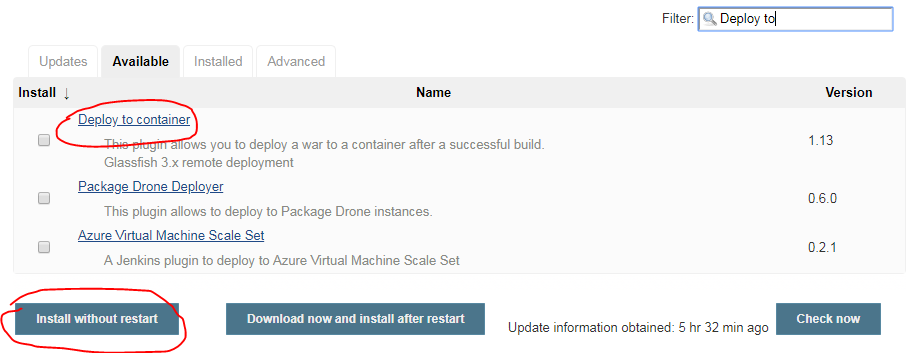
[Fig 1.15]

Navigate to available tab and search for the plugins to be installed as shown in Figure 1.16



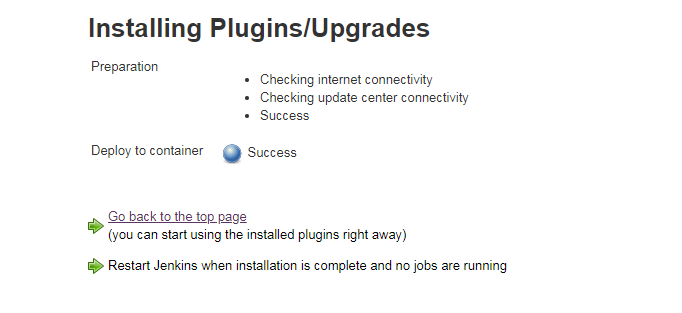
[Fig 1.16]

Select the plugin to be installed and use **Install without Restart** button to install the plugin without a Jenkins server restart. Refer Figure 1.17



[Fig 1.17]

The plugin will get installed as shown in Figure 1.18



[Fig 1.18]

Add the following plugins using the above mentioned step.

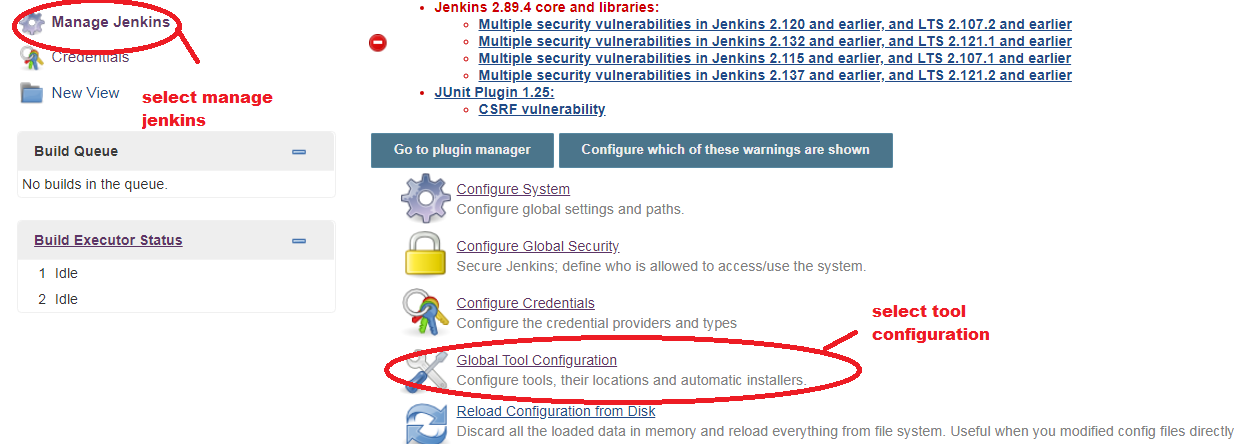
Plugins to be installed or verified for existence.

* Unleash maven plugin
* Pipeline
* GitHub plugin
* Deploy to Container

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

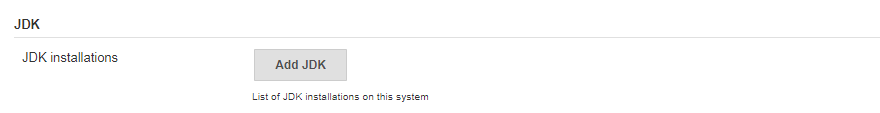
The installed plugins has to be configured to use it with Jenkins.

* Navigate to Manage Jenkins ->Global Tool Configuration. Refer Figure 1.19

****

[Fig 1.19]

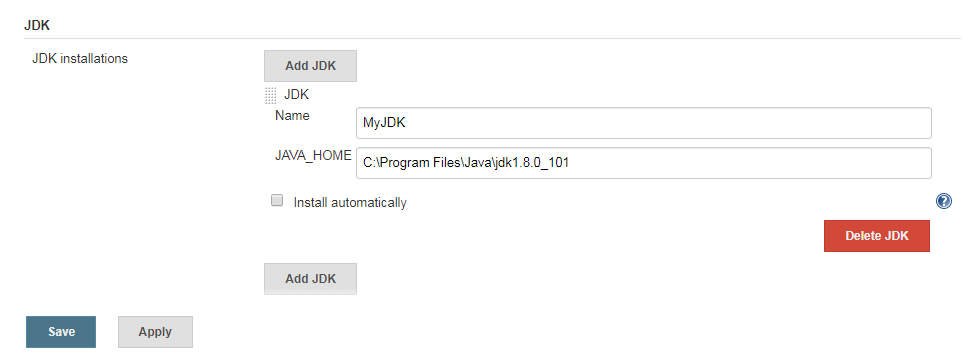
* Set up JDK. Click on JDK installations -> Add JDK. Refer Fig 1.20



[Fig 1.20]

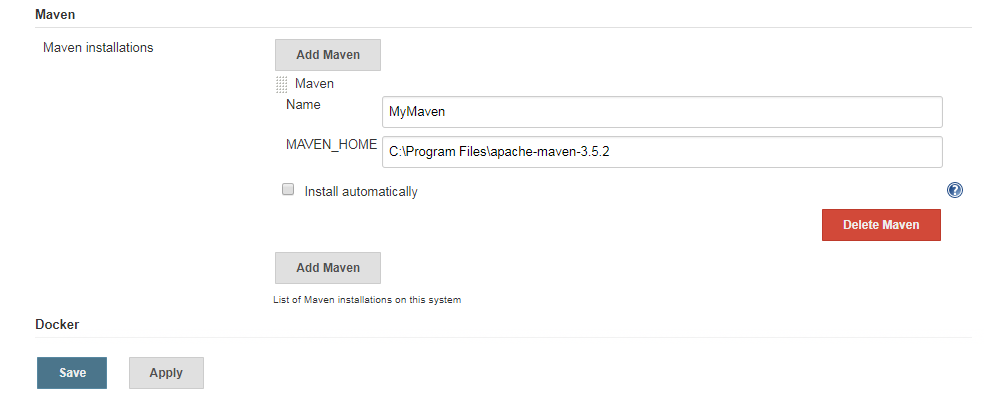
Specify the JDK Home path as per your JDK Installation.

Give a name and uncheck “**install automatically**” .Refer Fig 1.21



[Fig 1.21]

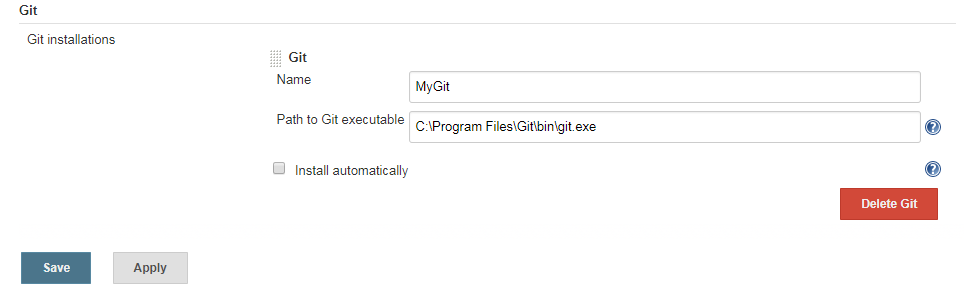
* Set up Maven as same as done in the previous step. Refer Fig 1.22



[Fig 1.22]

* Set up GIT configuration.

Configure the GIT installation path. Refer Fig 1.23



[Fig 1.23]

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

**Estimated Completion Time:** 20 Minutes

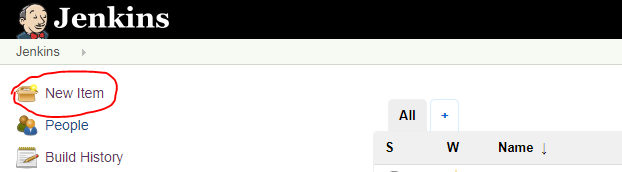
**Objective**: To create a freestyle project in Jenkins which pulls the project build from Git, builds and packages the project into deployable format.

**Solution: Follow the given steps**

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

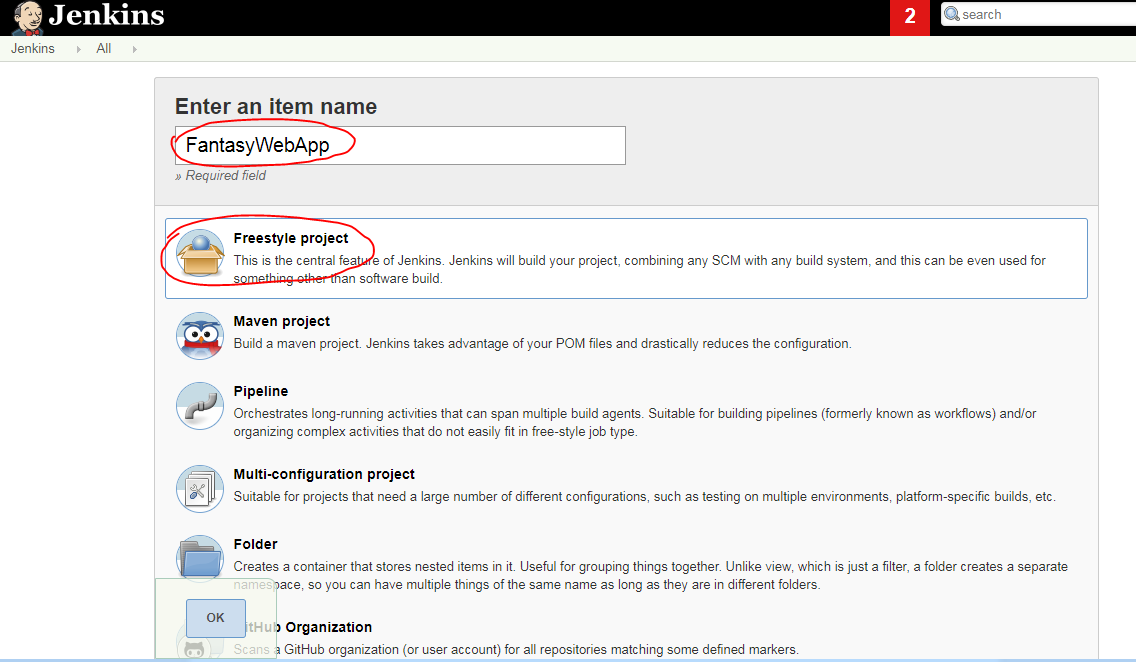
Follow the steps to create a new project.

* On Jenkins home page, select New Item. Refer Figure 2.1



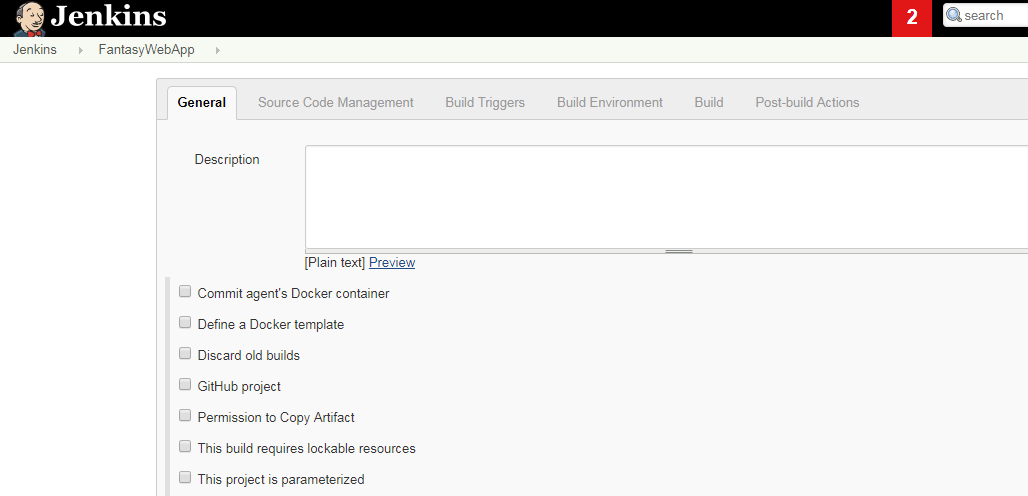
[Fig 2.1]

* Enter the project name as **FantasyWebApp** and select **FreeStyle Project** as the project template. Refer Figure 2.2



[Fig 2.2]

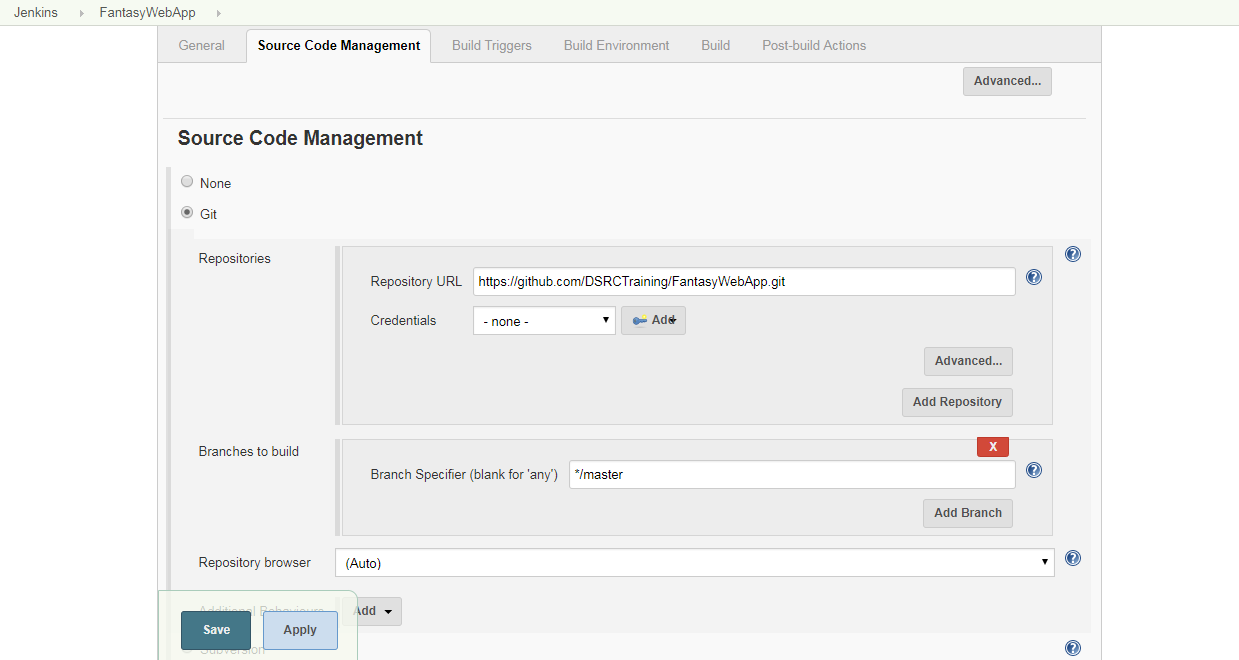
* On click of OK button, the project configuration page will get opened. Refer Figure 2.3



[Fig 2.3]

**Step 2:**  Configure source code management.

* Navigate to source code management and select Git radio button as show in Figure 2.4



[Fig 2.4]

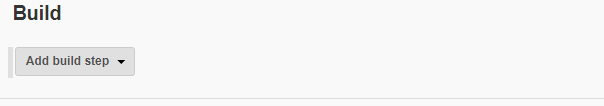
Specify the repository URL (Git URL where the project is pushed) and credentials if any.

URL: <https://github.com/DSRCTraining/FantasyWebApp.git>

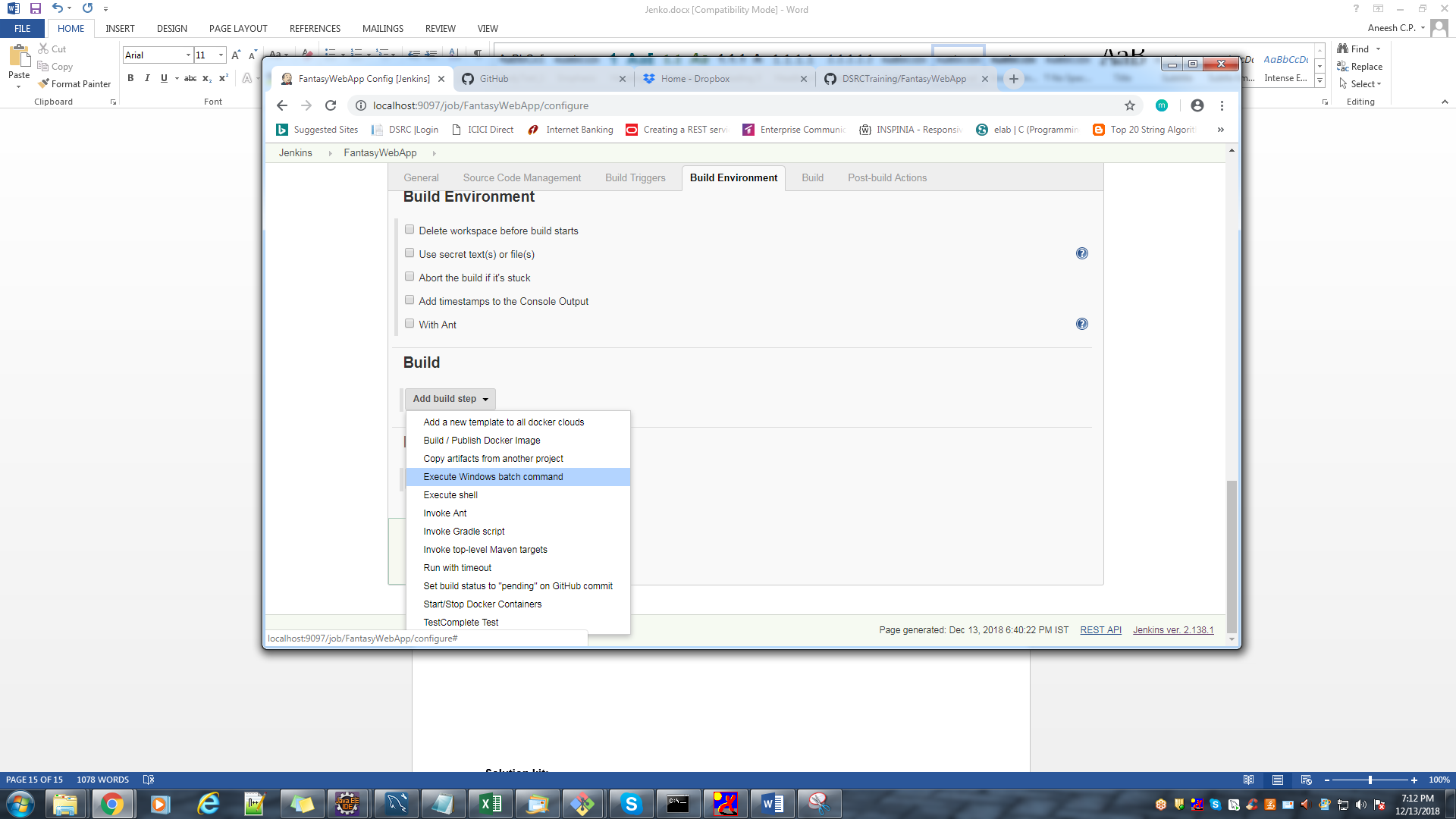
Note: The url given is a public repository and can be used for your testing. If you want to perform any branch operations, the branch name can be given inside **branches to build** section.

**Step 3:**  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.

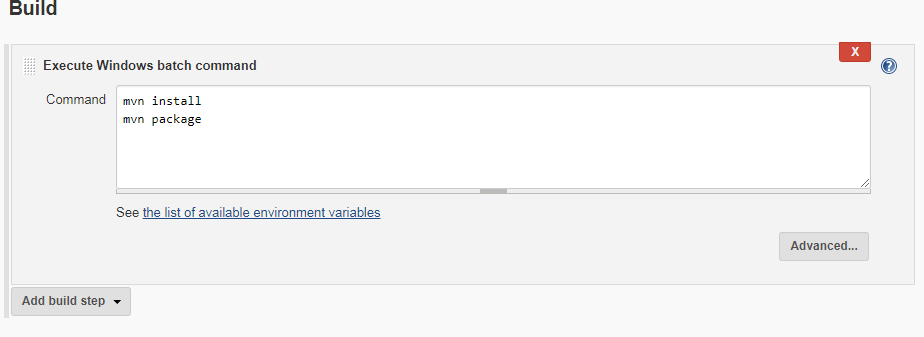


[Fig 2.5]



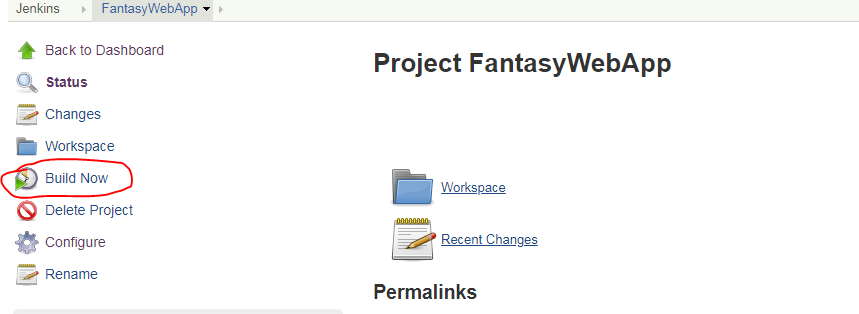
[Fig 2.6]

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



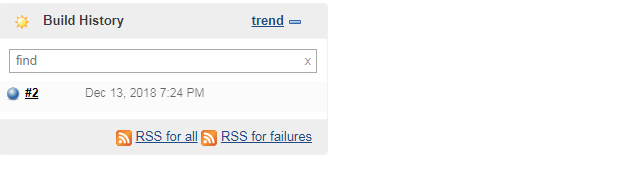
[Fig 2.7]

* 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 2.8



[Fig 2.8]

* The build progress and build status will get displayed on the Build History panel as shown in Figure 2.9.

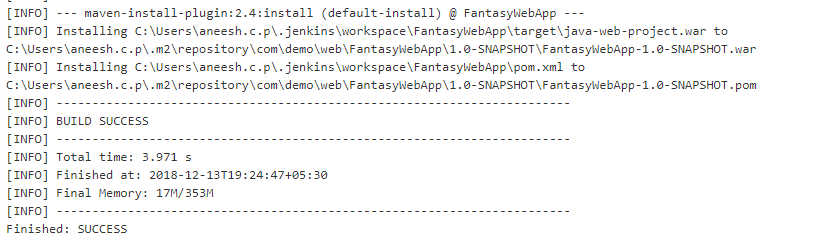


[Fig 2.9]

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 as shown in figure 2.10 and 2.11.



[Fig 2.10]



[Fig 2.11]

Guided Exercise 3**:** **Deploy the application in Tomcat from Jenkins.**

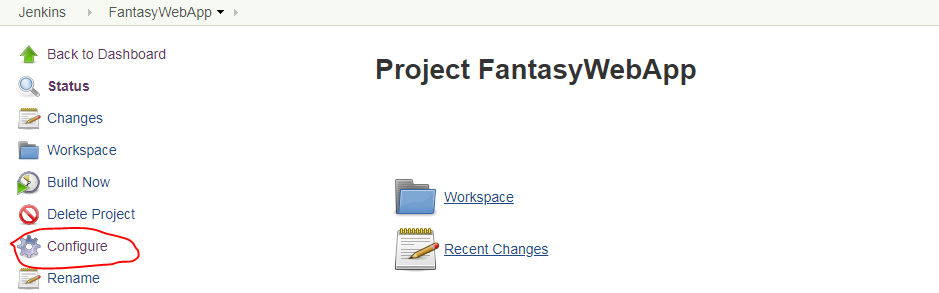
**Estimated Completion Time:** 20 Minutes

**Objective**: To deploy the project in tomcat from Jenkins server after successful build.

**Solution: Follow the given steps**

**Step 1:**  Configure the post build actions.

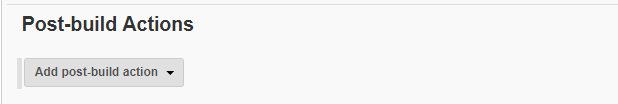
* Select the project in Jenkins Home page and select **configure** as shown in Figure 3.1



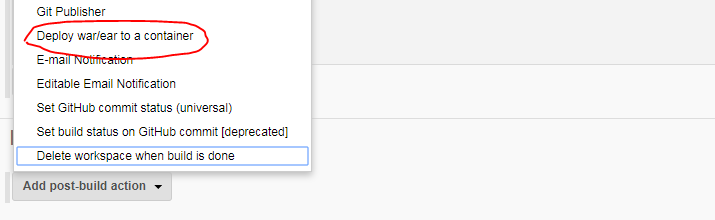
[Fig 3.1]

The project configuration page will get opened.

* Navigate to Post Build Actions and select “deploy war/ear to container”. Refer Figure 3.2 and 3.3

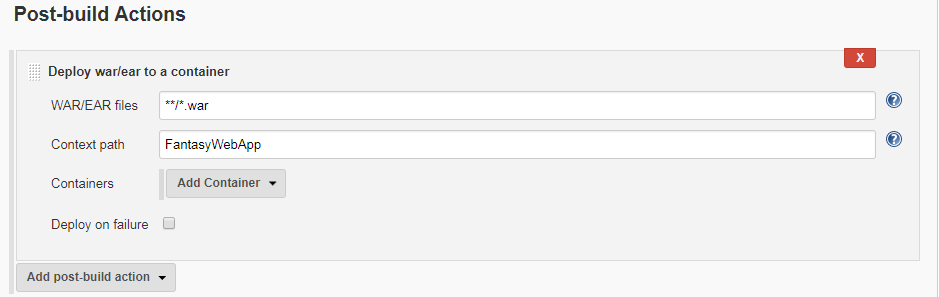


[Fig 3.2]



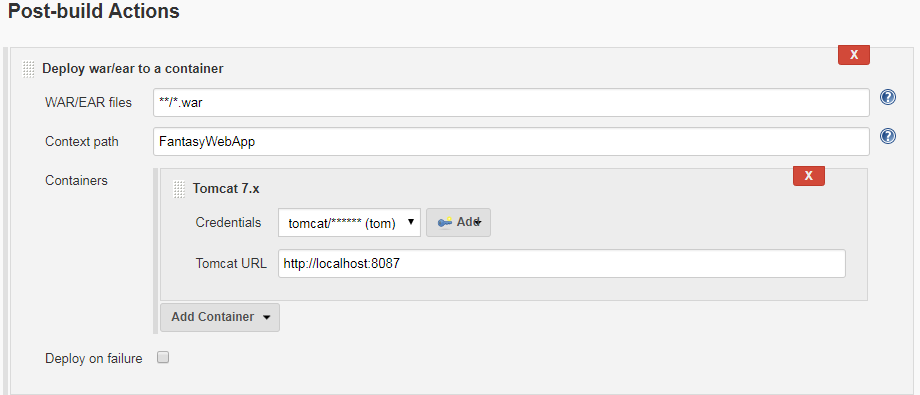
[Fig 3.3]

* Specify the location of the war/ear file and context path ( Name of the application when it is deployed ) in this section as shown in Figure 3.4



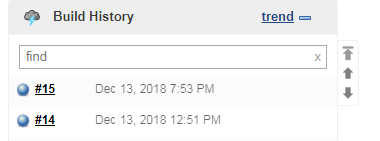
[Fig 3.4]

* Click on Add Container to specify the container/web server to which the war file needs to be deployed. Specify the tomcat credentials and server url as shown in Figure 3.5



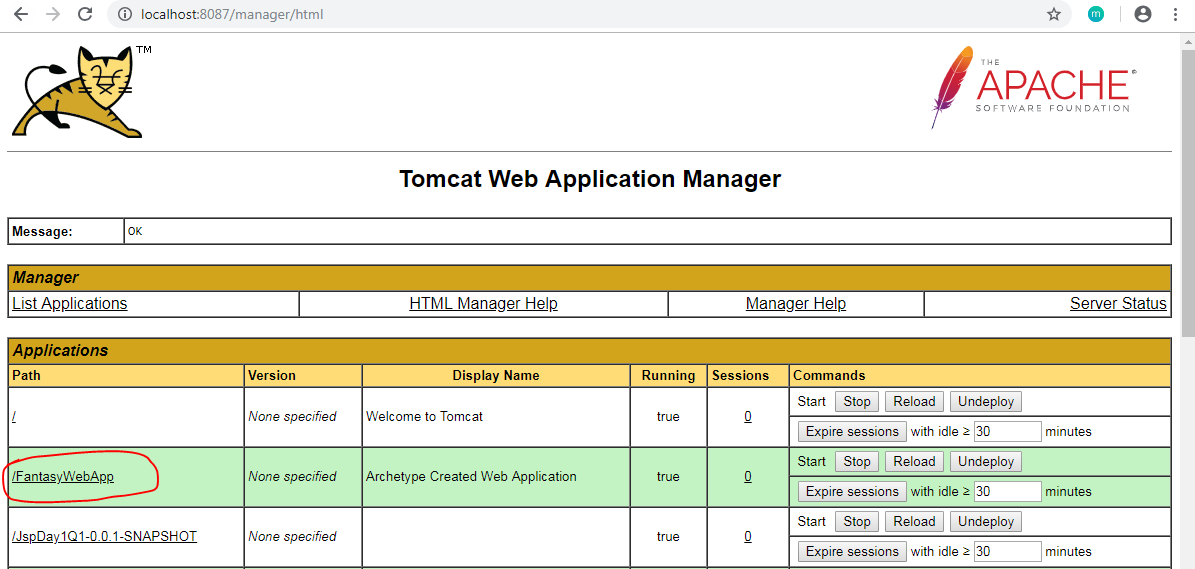
[Fig 3.5]

* Save and build the project again to deploy the project in tomcat server.
* Click on **build now** to build the project and check the status in the build history as shown in Figure 3.6.



[Fig 3.6]

* Open tomcat manager page and verify the project is deployed as shown in Figure 3.7



[Fig 3.7]

* To access the web application use the URL: <http://localhost:8087/FantasyWebApp> and verify the result.

**Solution kit:**

<<Embedded Solution file>>

**Summary:**

You have learnt to manage Jenkins in DevOps Environment.