Ffffcccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc cc

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**

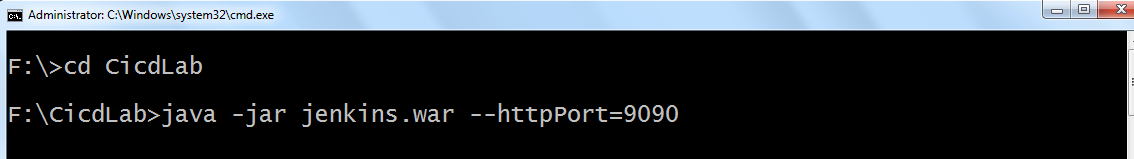
* Maven 3.5
* Tomcat 7
* Jenkins
* Git

**Guided Exercise 1:** **Setup Jenkins**

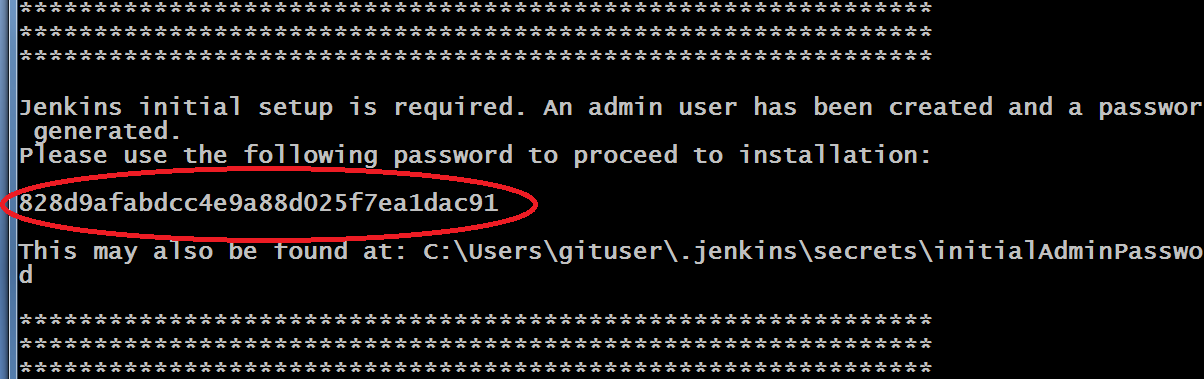
* + Verify the jenkins.war file is present in the path C:\Program Files (x86)\Jenkins
  + Open command prompt and navigate to the Jenkins folder mentioned above.
  + Start Jenkins on port 9090 by executing the below java command in the command prompt (default port is 8080) Result is shown in [Fig 1.9-2.1]

java –jar jenkins.war --httpPort=9090

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

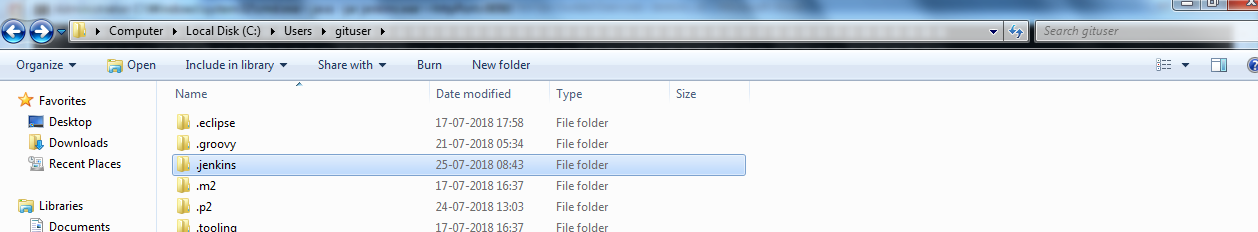


[Fig 1.9]



[Fig 2.1]

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



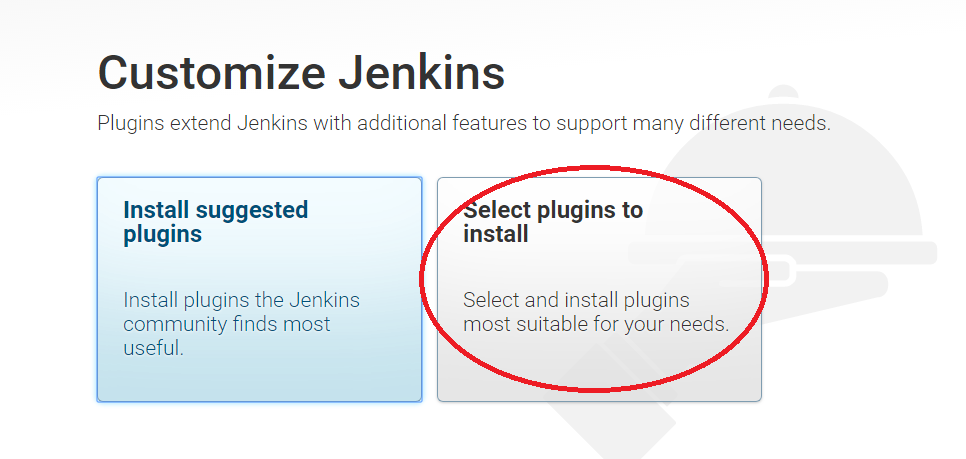
[Fig 2.2]

* Admin password is stored in initialAdminPassword file in users profile

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

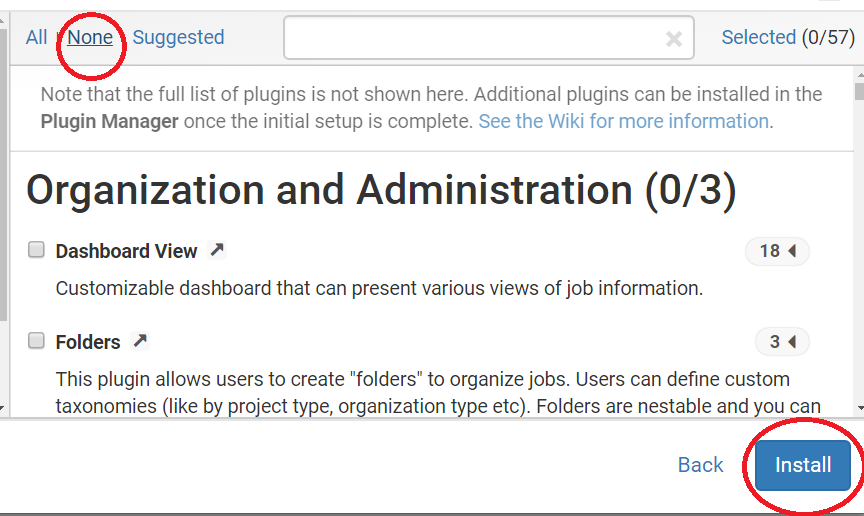
**Step 9: Set up the Jenkins Environment**

* 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 ]



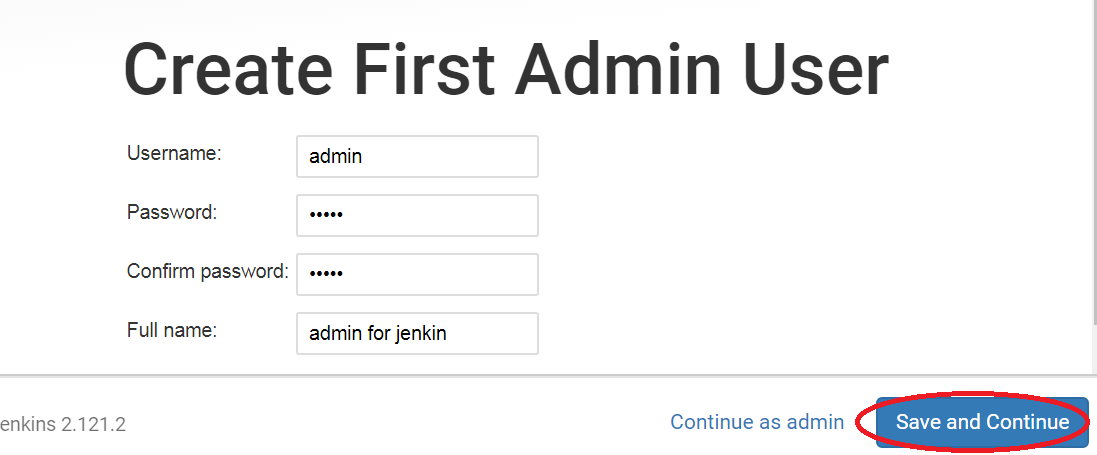
[Fig 2.3]

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

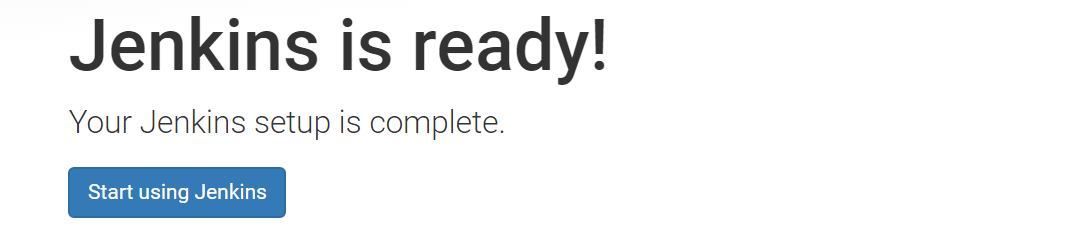


[Fig 2.4]

* 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] ]

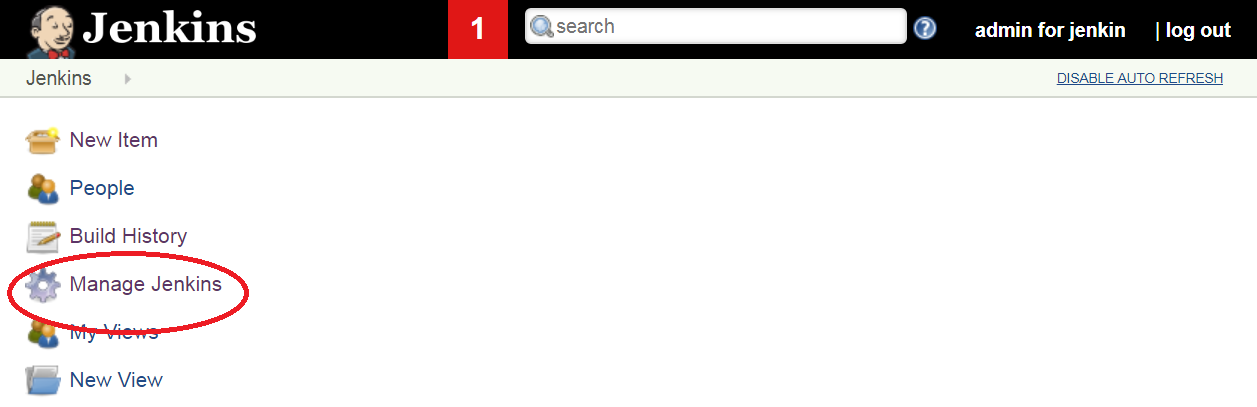


[Fig 2.5]



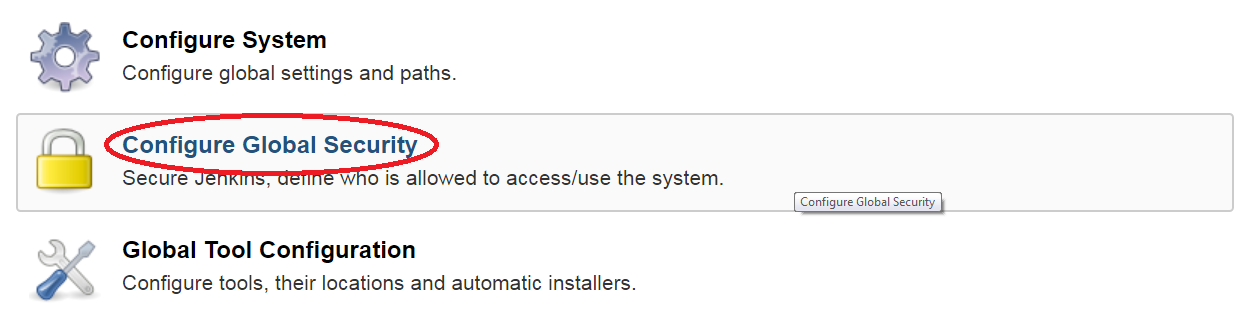
[Fig 2.6]

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



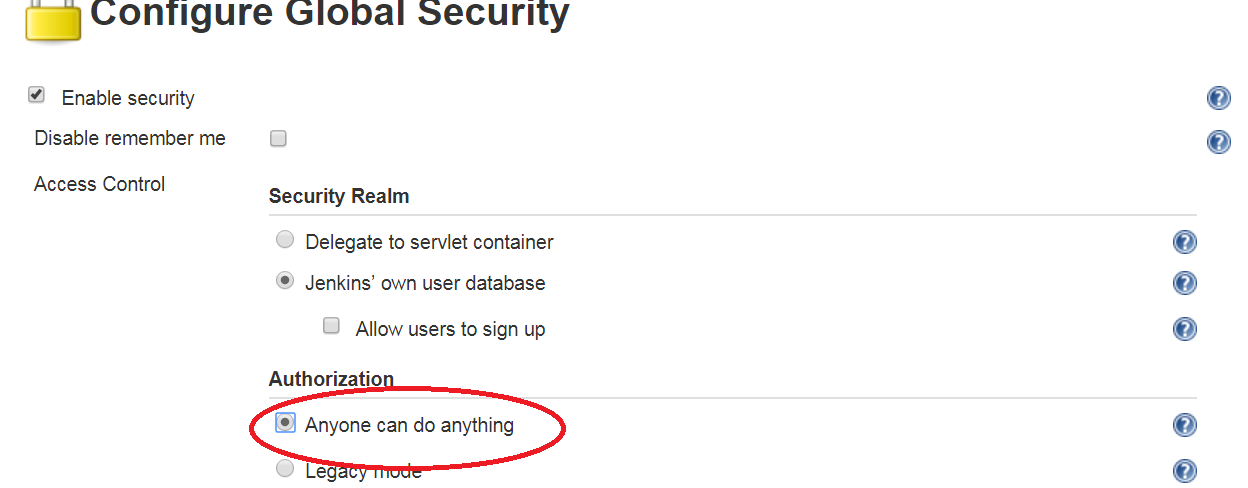
[Fig 1.11]

* 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: Add the required plugins in Jenkins and configure the global tools and system.**

**Estimated Completion Time:** 10 Minutes

**Objective**: Install and configure the required plugins in Jenkins.

**Steps to follow:**

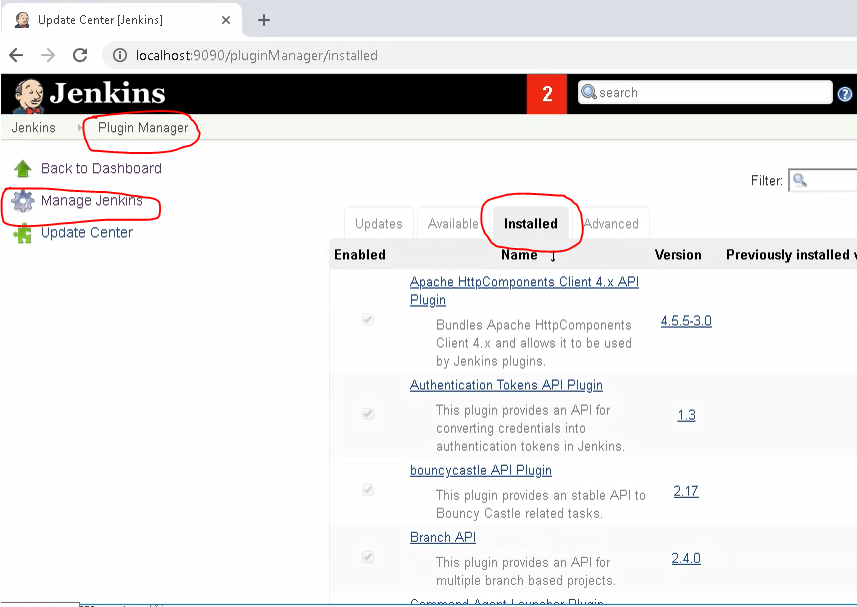
**Step 1: Install the required plugins.**

*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**

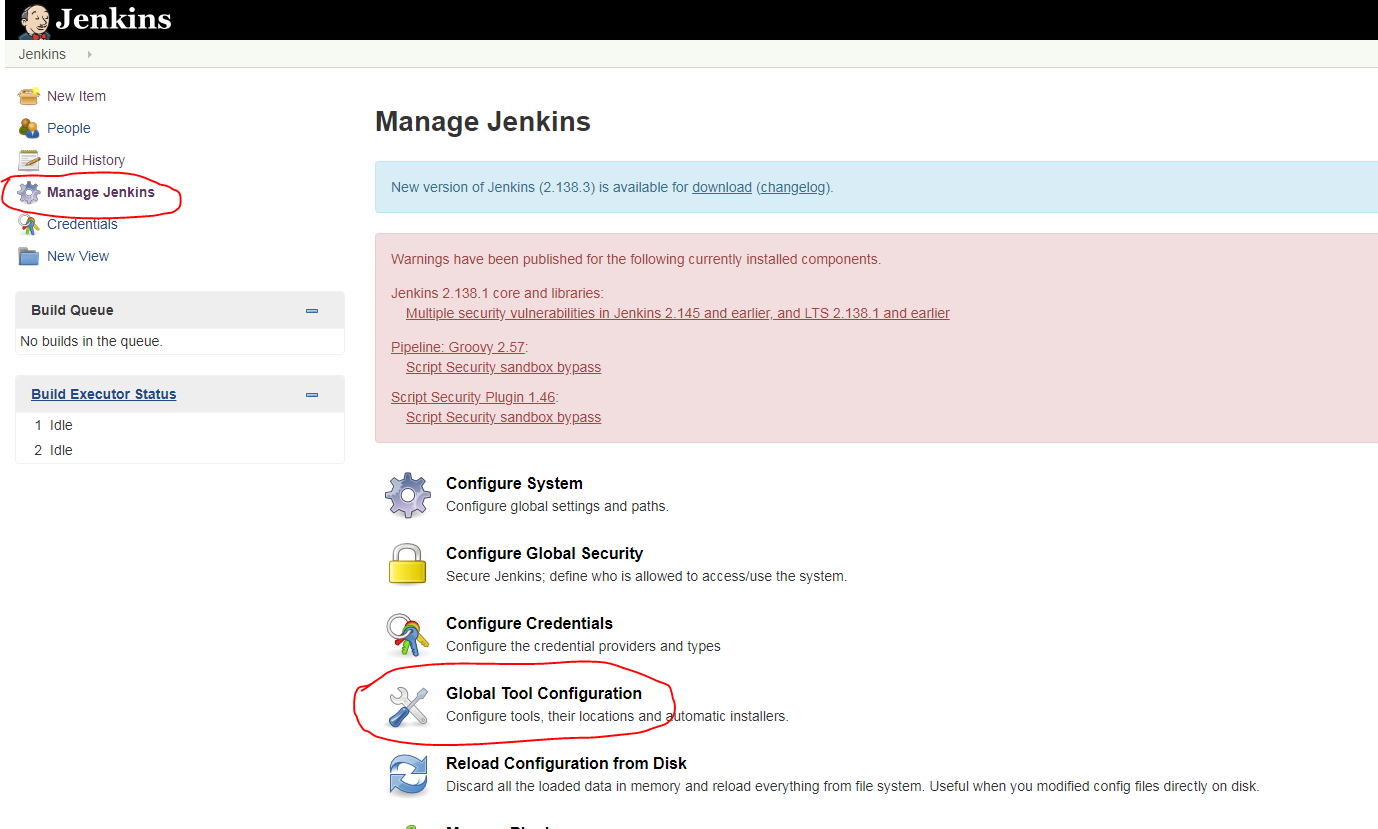
1. In a browser, navigate to the URL, <http://localhost:9090>
2. Click on Manage Jenkins -> Manage Plugins -> Installed
3. Here you will see a list of all the installed plugins.



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

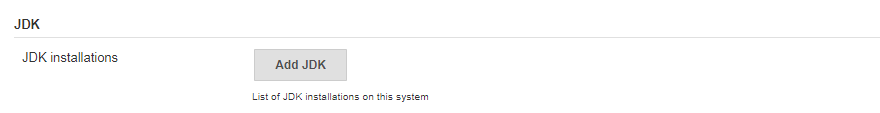
The installed plugins and build environment configurations are as follows.

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



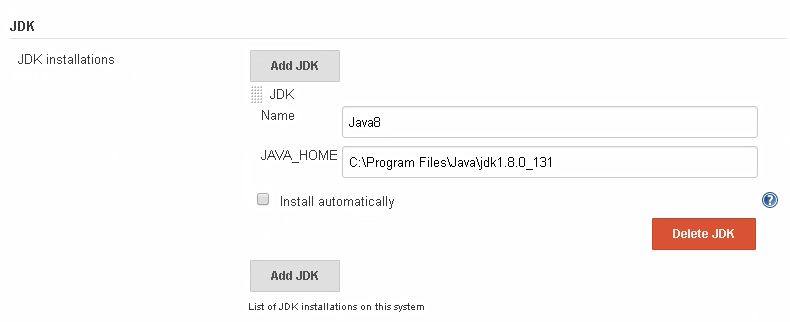
[Fig 3.9]

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



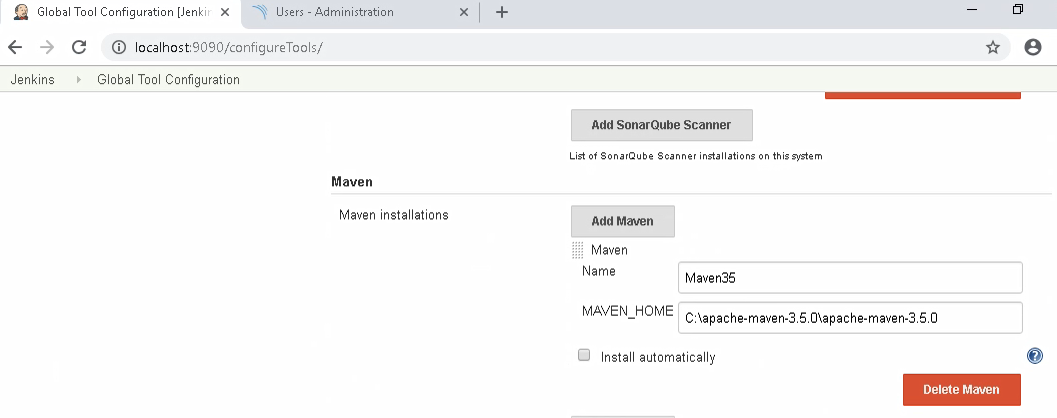
[Fig 3.10]

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



[Fig 4.0]

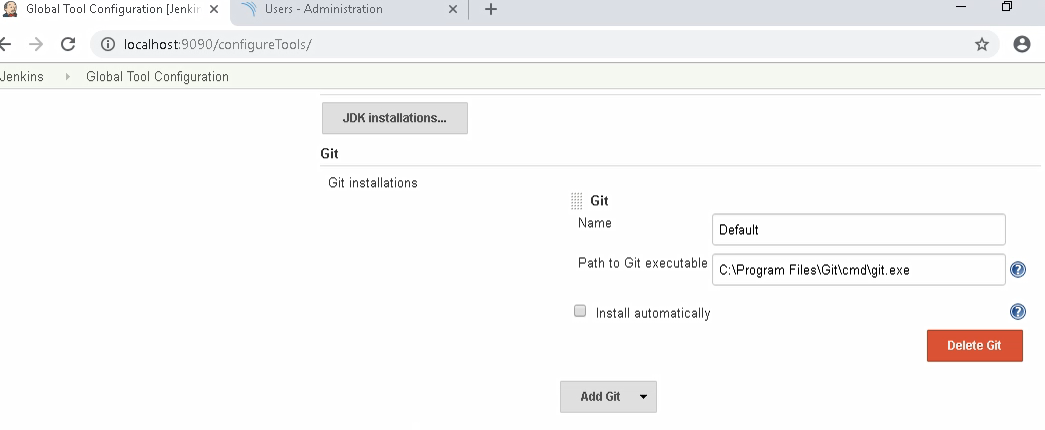
3. Set up Maven as same as done in the previous step. Refer Fig 4.1



[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

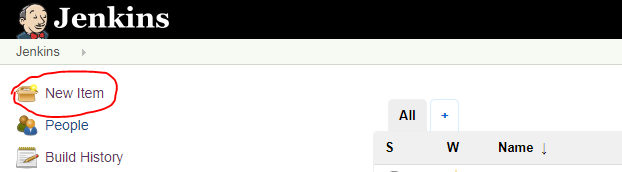
**Objective**: To create a freestyle project in Jenkins which pulls the project build from Git, builds and packages the project into deployable format.

**Solution: Follow the given steps**

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

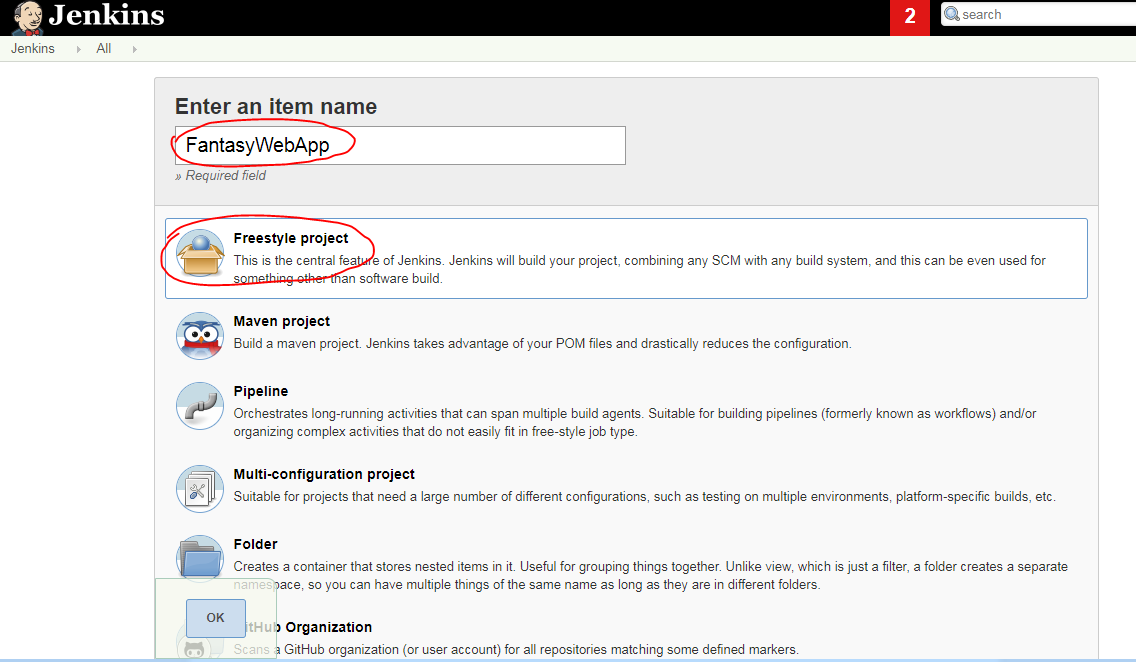
Follow the steps to create a new project.

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



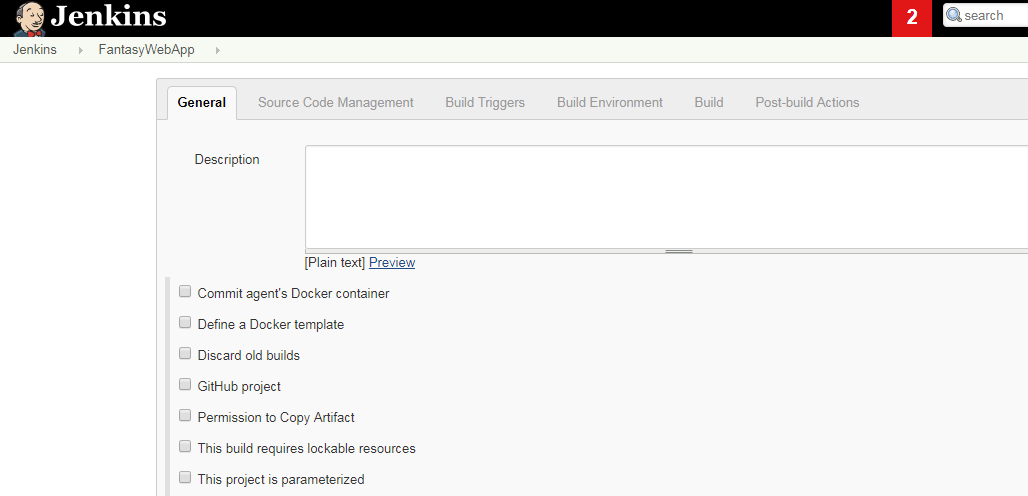
[Fig 2.1]

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



[Fig 2.2]

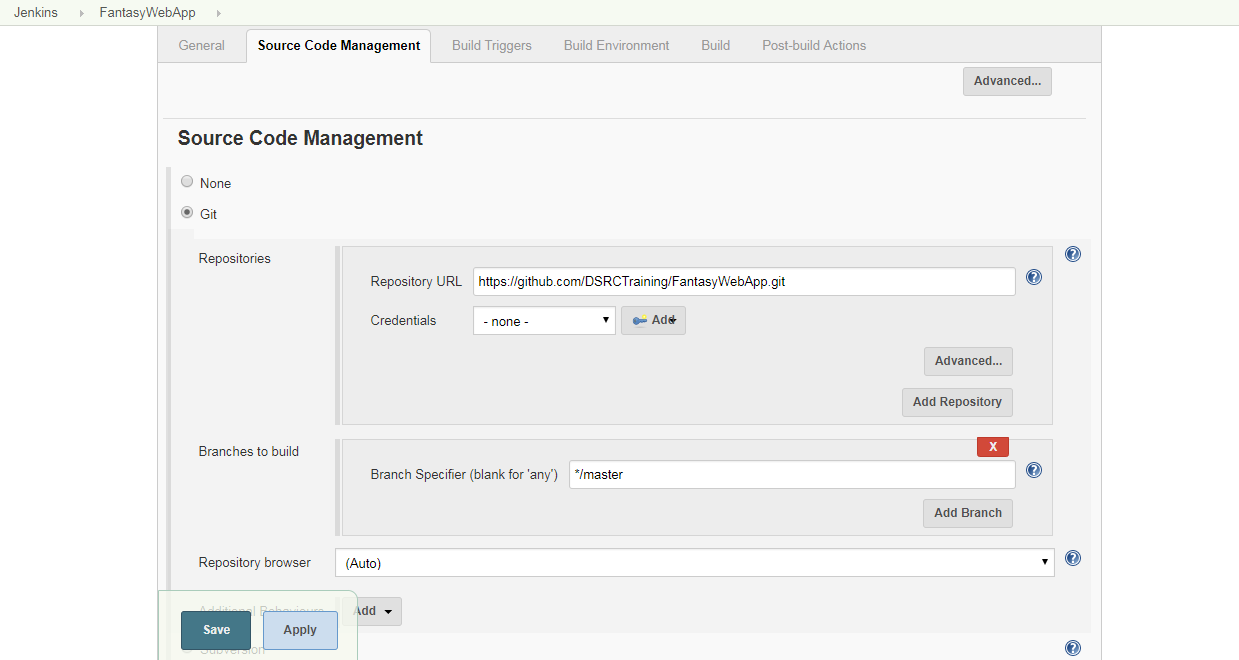
* On click of OK button, the project configuration page will get opened. Refer Figure 2.3



[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 your repository URL.

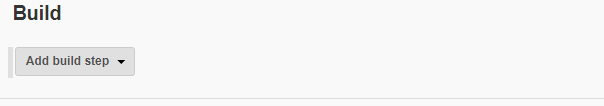
Example URL: https://code.cognizant.com/250034/cicdproject.git

Note: Please log into <https://code.cognizant.com> with your cognizant credentials. Copy the sample codebase from <https://github.com/DSRCTraining/FantasyWebApp.git> into your personal repository.

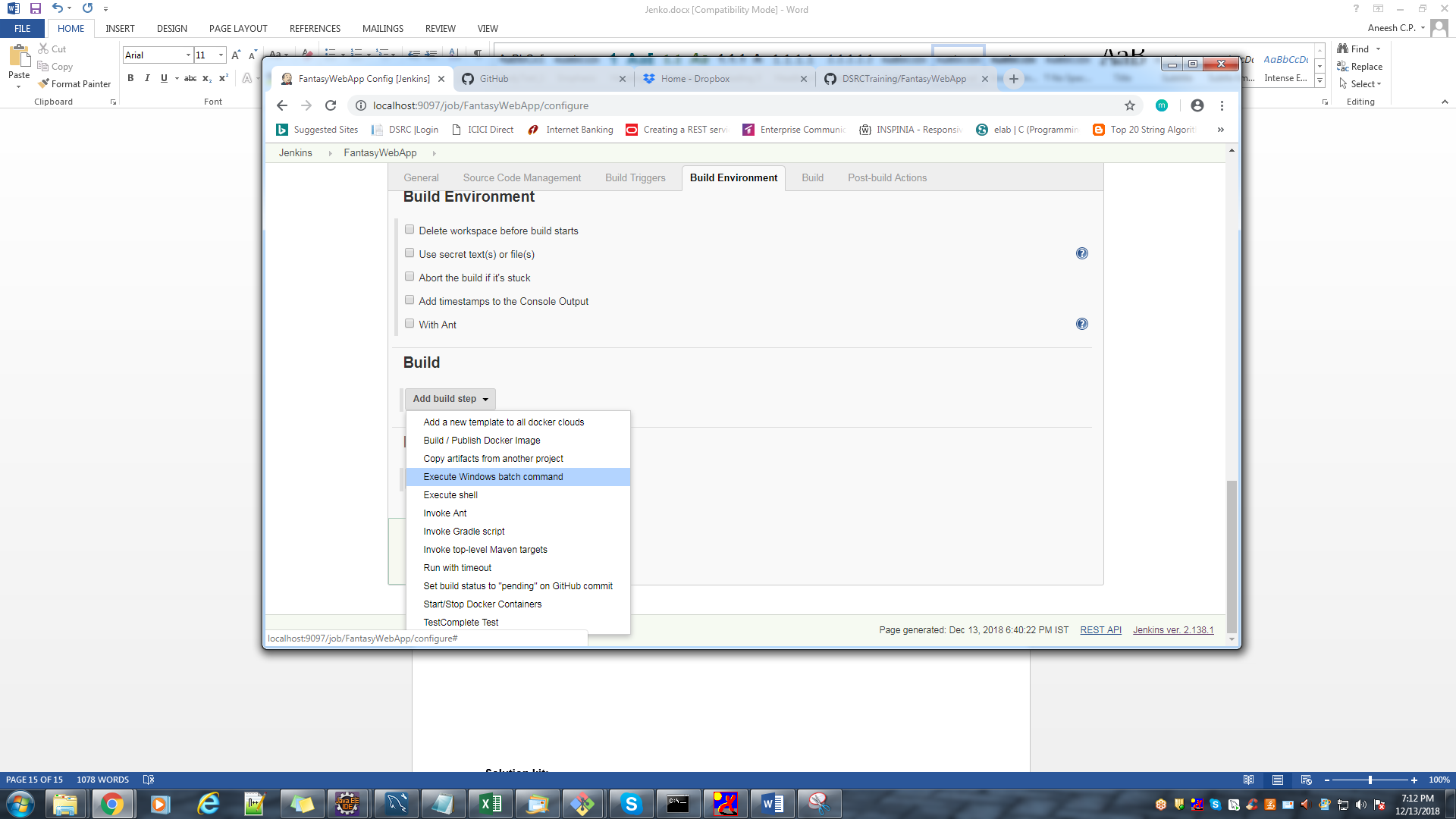
If you want to perform any branch operations, the branch name can be given inside **branches to build** section in Jenkins.

**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.

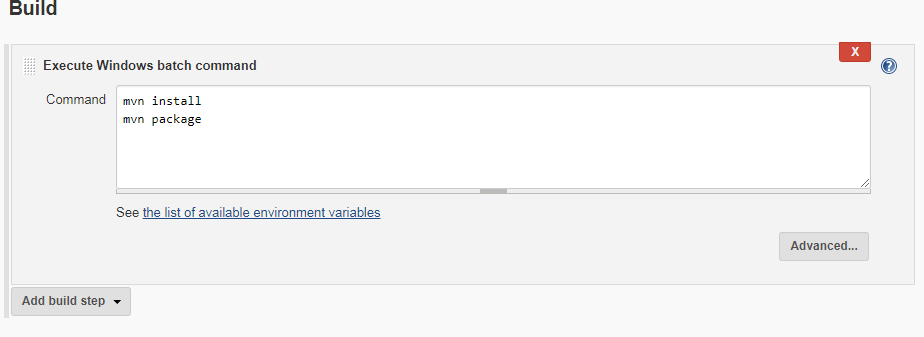


[Fig 2.5]



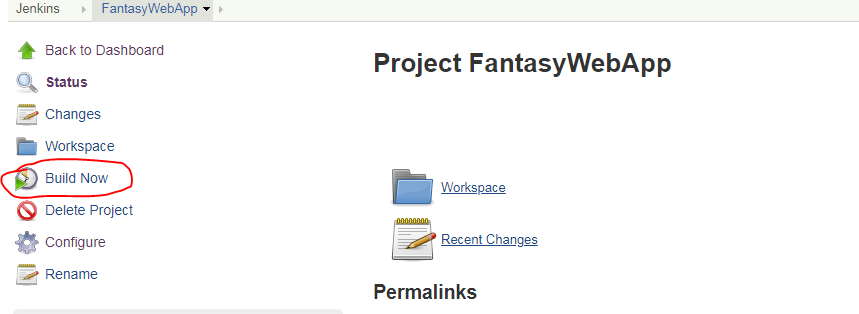
[Fig 2.6]

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



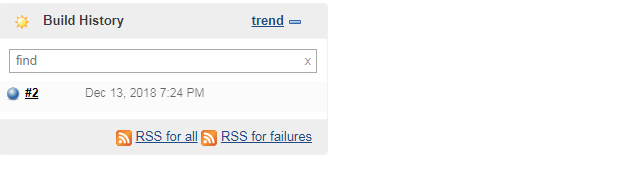
[Fig 2.7]

* 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.

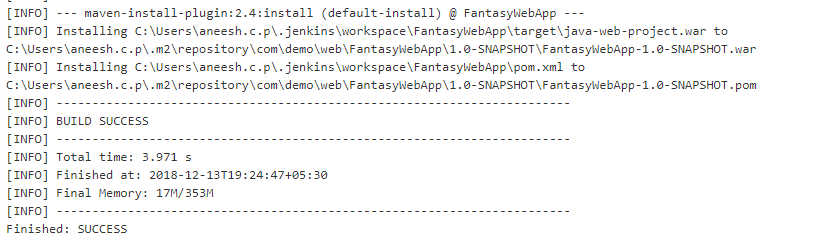


[Fig 2.9]

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.**

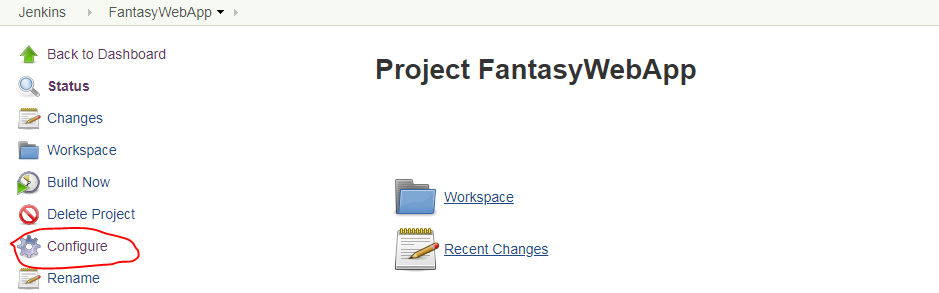
**Estimated Completion Time:** 20 Minutes

**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.

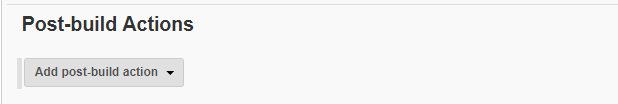
* Select the project in Jenkins Home page and select **configure** as shown in Figure 3.1



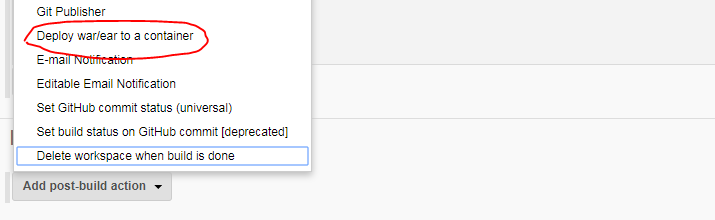
[Fig 3.1]

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

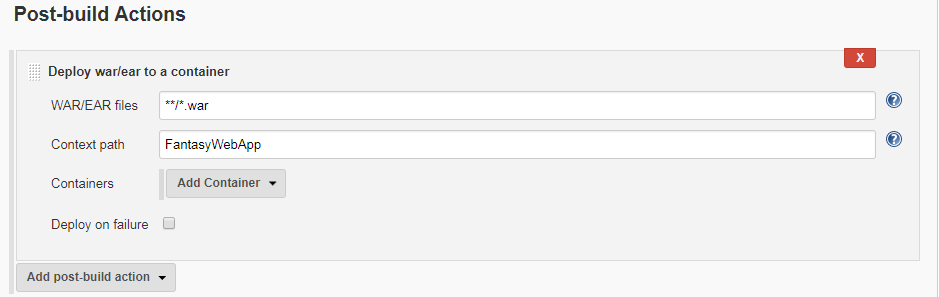


[Fig 3.2]



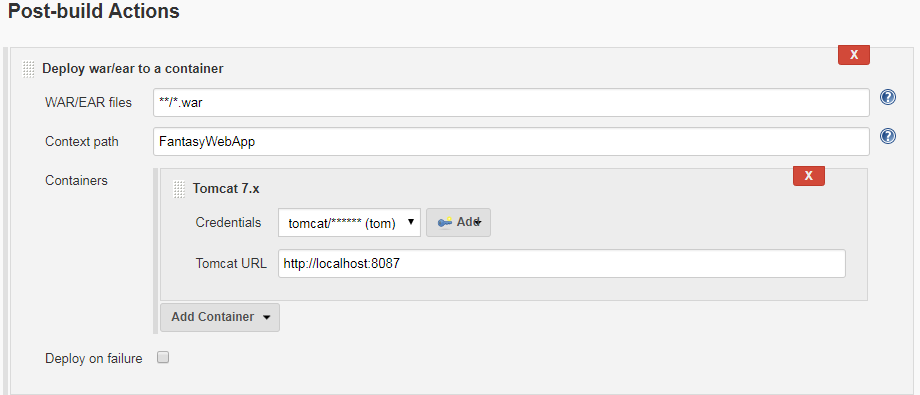
[Fig 3.3]

* 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



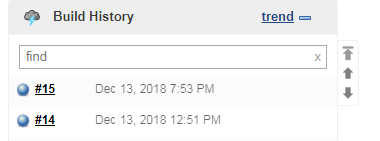
[Fig 3.4]

* 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



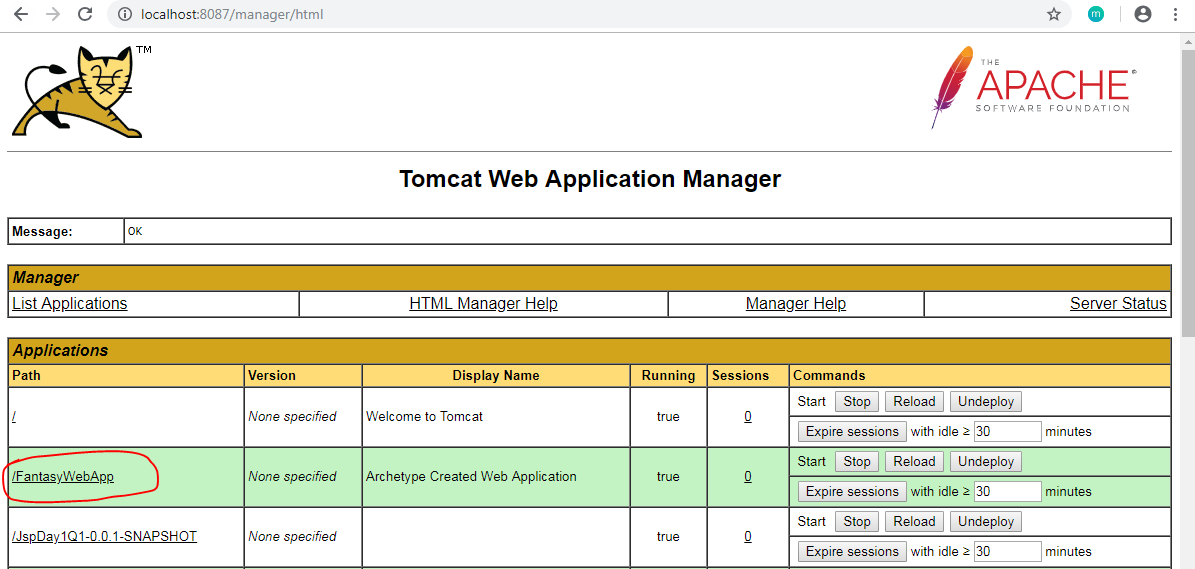
[Fig 3.5]

* 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>>

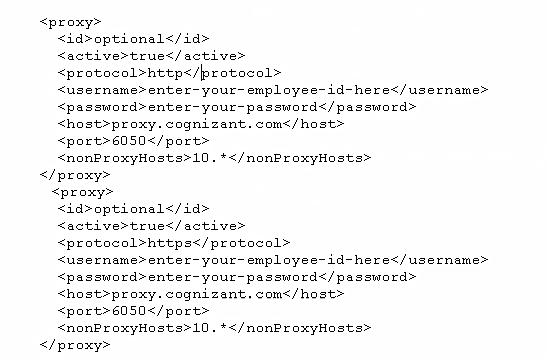
**Summary:**

You have learnt to manage Jenkins in DevOps Environment.

**Trobleshooting:**

**Configure proxy for Maven**

1. Open the settings.xml file in the maven\_installation\_dir\conf.
2. Under proxies, look for the section for the http protocol. Update this section with:  
   username – your CTS Employee ID  
   password – your CTS network password  
   host – proxy.cognizant.com  
   port – 6050  
   nonProxyHosts – 10.\*
3. Add another proxy section for the https protocol.

****

1. Ensure that the same settings.xml file is present under YOUR\_HOME\_DIR\**.**m2\conf and maven\_installation\_dir\conf. This will ensure that you don’t encounter any errors when running Maven commands from the command prompt as well as from within Jenkins builds.