

Table of Contents

[1](#_Toc520463040)

[Table of Contents 2](#_Toc520463041)

[1 Lab Details: 3](#_Toc520463042)

[2 DevOps Tools: Guided Exercises 3](#_Toc520463043)

[2.1 Jenkins 3](#_Toc520463044)

[Guided Exercise 1 4](#_Toc520463045)

[Guided Exercise 2 12](#_Toc520463046)

[Guided Exercise 3 19](#_Toc520463047)

# 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

Overview

A leading Company is a share trading firm and an Indian financial service company that offers very low brokerage equity investments, retail and institutional broking, currencies and commodities trading, and mutual funds. The company is operating for years and have customer base all over India. There are many customer records which are still on papers and the company needs to migrate into a new website. The project needs to be developed and deployed in DevOps environment.

The web application is designed and developed in such a way that it should support continuous integration using Jenkins.

**Scope**:

You have been assigned the task to setup, configure, test, build, create job and deploy the provided application using Jenkins for continuous integration on a DevOps environment.

**Tasks:**

* Trigger build
* Execute Test cases
* Generate Report for test cases
* Send feedback to the developer
* Deploy the application

**Steps*:***

1. Install, test and Configure Jenkins
2. Install plug-in in Jenkins
3. Jenkins and Tomcat Setup
4. Setting Jenkins Home Directory
5. Configuring build jobs
6. Configure maven project
7. Jenkins configuration with Maven and GitHub
8. Configure Junit Reports in Jenkins
9. Configuring automated deployment in Jenkins
10. Configuring delivery pipeline in Jenkins

**Resources / References:**

* Maven3.5.0
* Tomcat 7.0.90
* Jenkins - 2.121.2
* Git-2.16.1.1.4
* https://tomcat.apache.org/download-70.cgi
* <https://jenkins.io/download/>
* <https://git-scm.com/downloads>
* <https://maven.apache.org/download.cgi>
* <https://github.com/>
* GitHub Repository

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

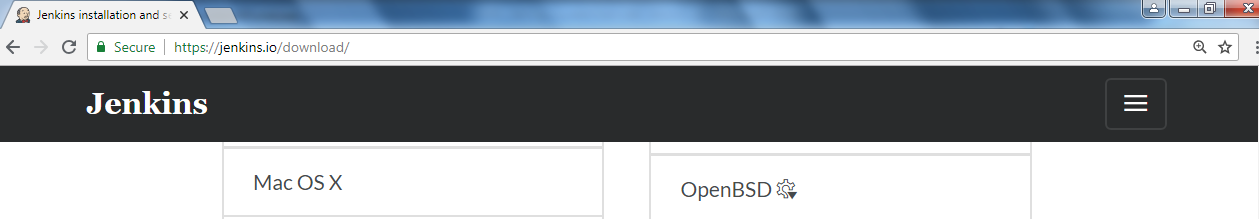
**Estimated Completion Time:** 15 Minutes

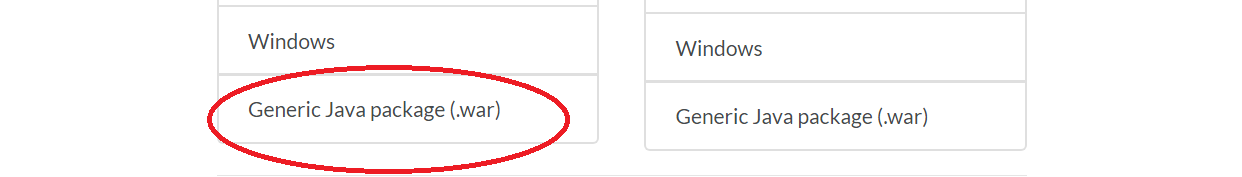
**Objective**: To Install Jenkins on Windows System

Steps to follow:

**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) [Fig 1.1]





[Fig 1.1]

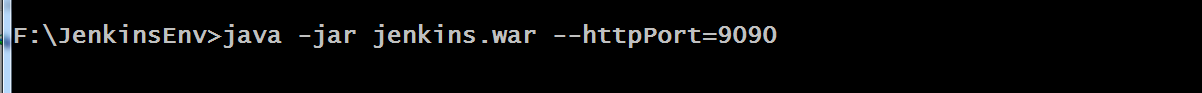
* + Create new folder and copy the Jenkins.war.

Eg F:\JenkinsEnv\jenkins.war

* + Start Jenkins on 9090 port (default port is 8080)

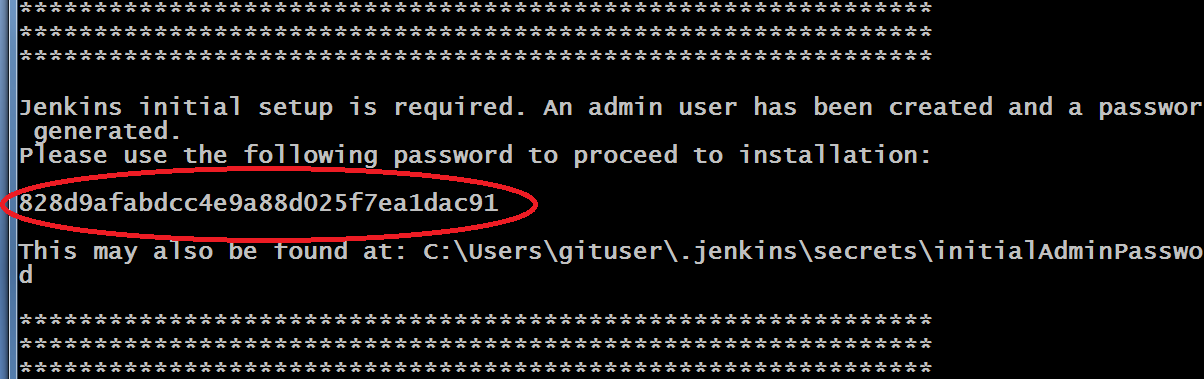
Result is shown in [Fig 1.2-1.4 ]

* + Open command prompt
  + Change directory to f:\JenkinsEnv and add following command
  + Java –jar Jenkins.war –httpPort-9090



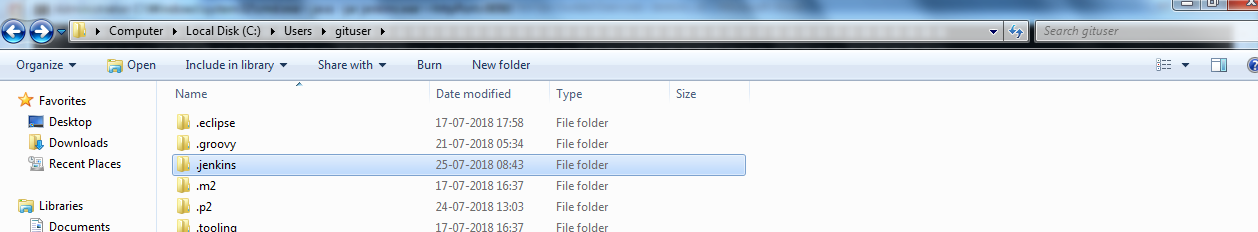
[Fig 1.2]

**Jenkins installation creates admin password:**



[Fig 1.3]

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



[Fig 1.4]

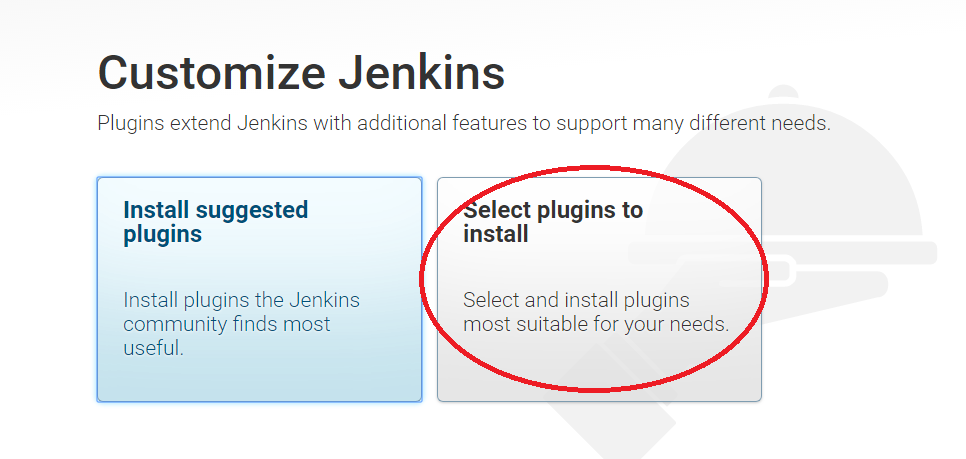
* Admin password is stored in initialAdminPassword file in users profile

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

**Step 2:**

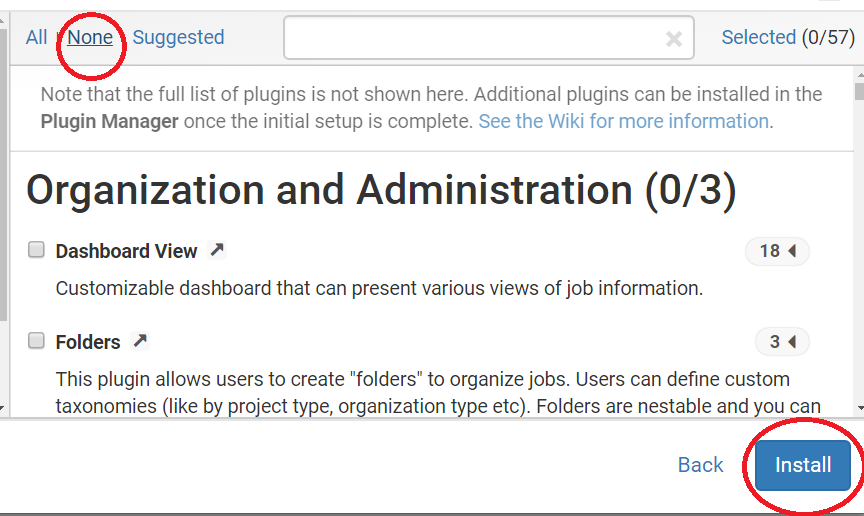
**Set up the Jenkins Environment**

* Open web browser add url : <http://localhost:9090>
* Give the admin password to Unlock Jenkins
* Select plugins to install Reference [Fig 1.5 ]



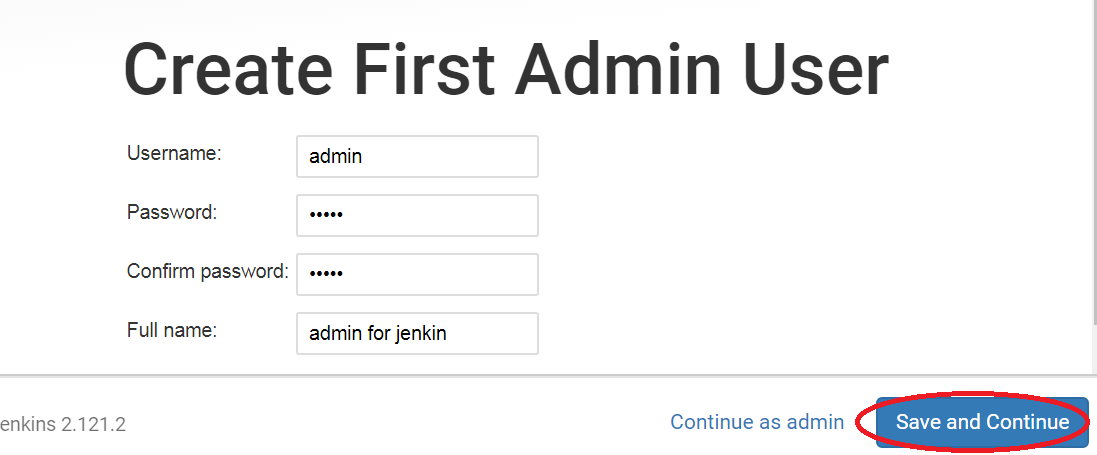
[Fig 1.4]

* Select None (click install ) Reference [Fig 1.5 ]

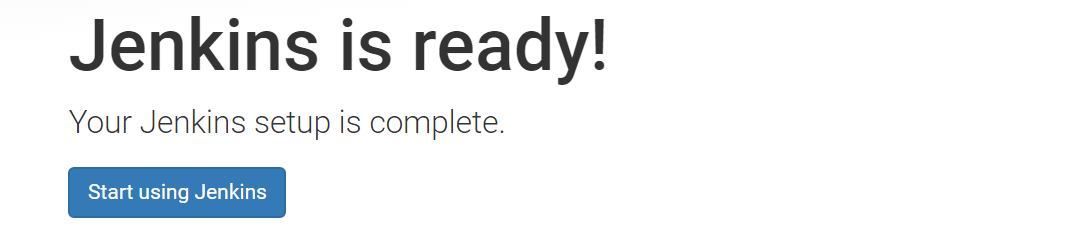


[Fig 1.5]

* Add user name and password (eg user admin password admin )
* Select save and continue Reference [Fig 1.6 -1.7]

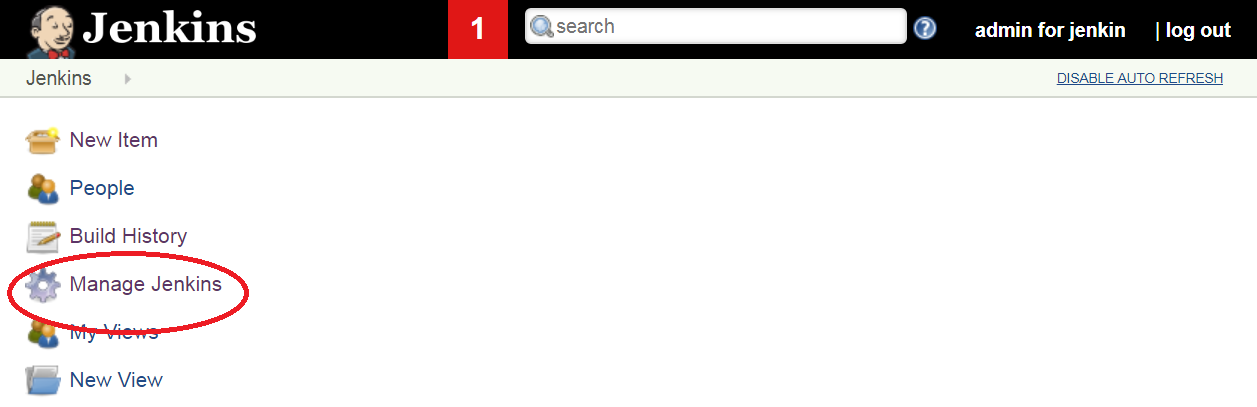


[Fig 1.6]



[Fig 1.7]

**Start Jenkins Dashboard**

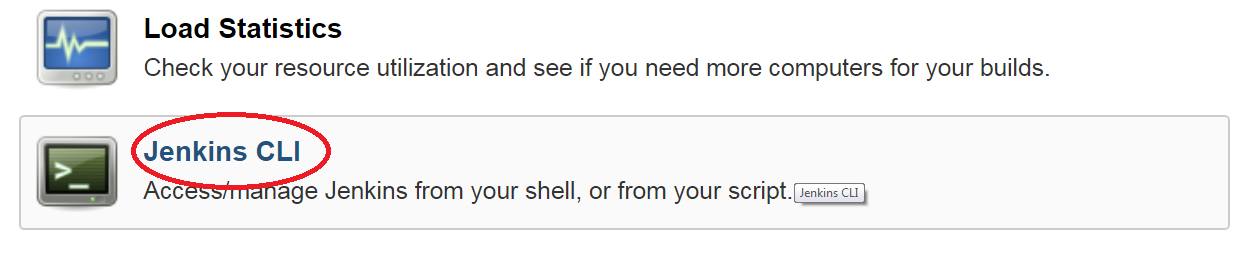


[Fig 1.8]

**Step 3:**

**Configure CLI client for Jenkins**

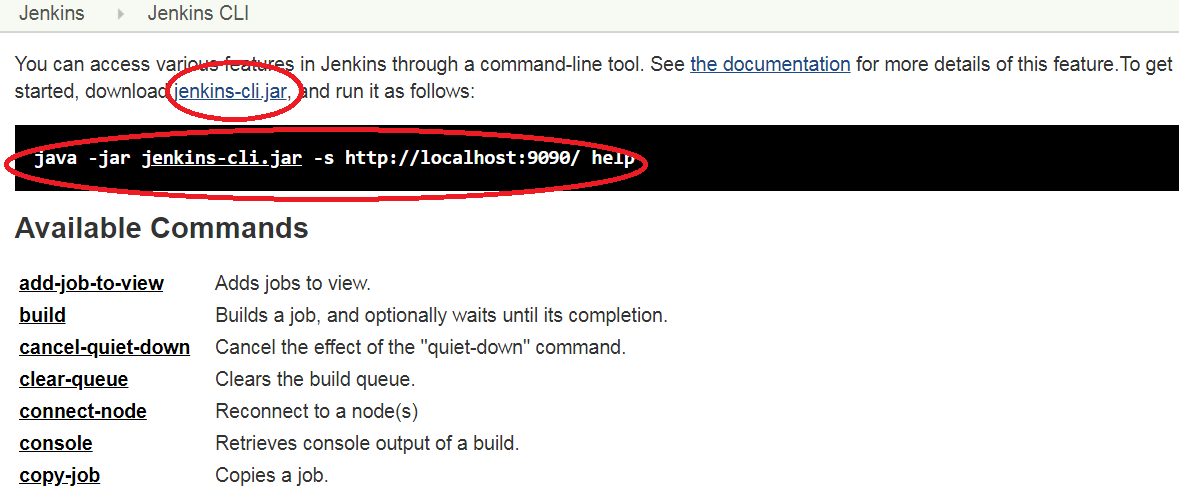
* Select Manage Jenkins Reference [Fig 1.8 ]
* Select Jenkins CLI Reference [Fig 1.9 ]



[Fig 1.9]

* Download Jenkins-cli.jar Reference [Fig 2.0 ]
* Copy the jar to working folder (F:\JenkinsEnv\jenkins-cli.jar)
* Open the command prompt
* Change directory to F:\JenkinsEnv
* Use command
* Java –jar Jenkins-cli.jar –s <http://localhost:9090/> help

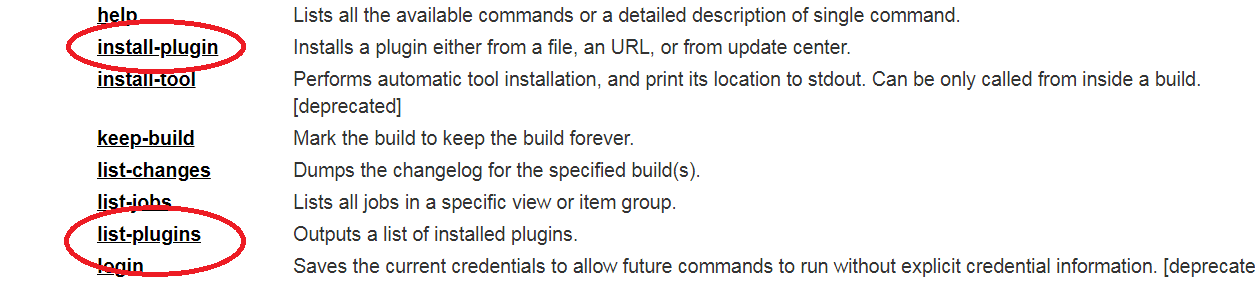
(shows list of all the command to configure Jenkins)

****

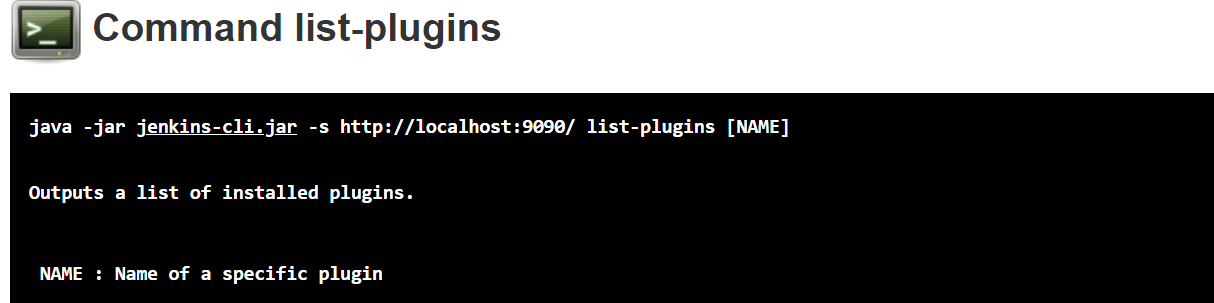
[Fig 2.0]

**Step 4:**

**Install required plug-ins using Jenkins CLI**



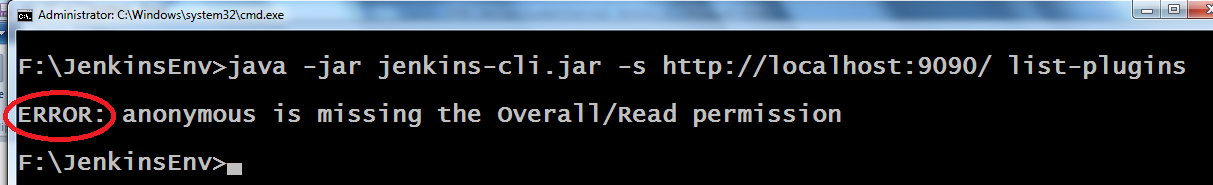
* Check list of installed plugins (select list-plugins ) Reference [Fig 2.1-2.2 ]



[Fig 2.1]

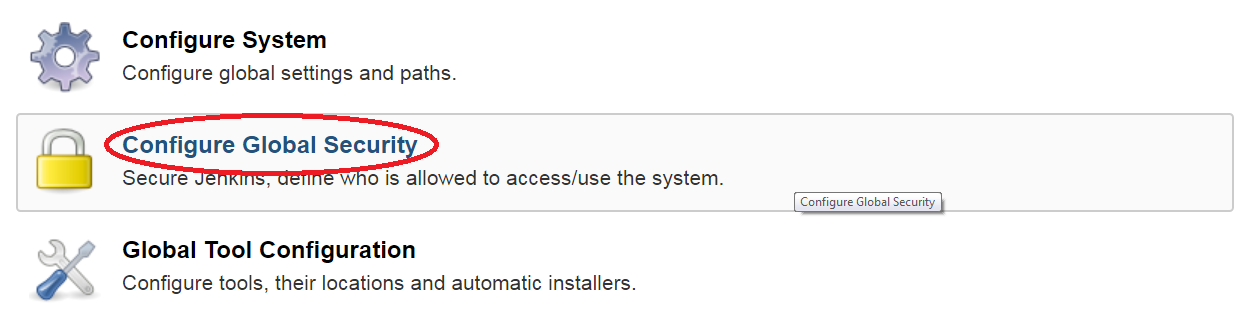
* Open the command prompt

Use command java –jar Jenkins-cli.jar –s <http://localhost:9090/> list-plugins



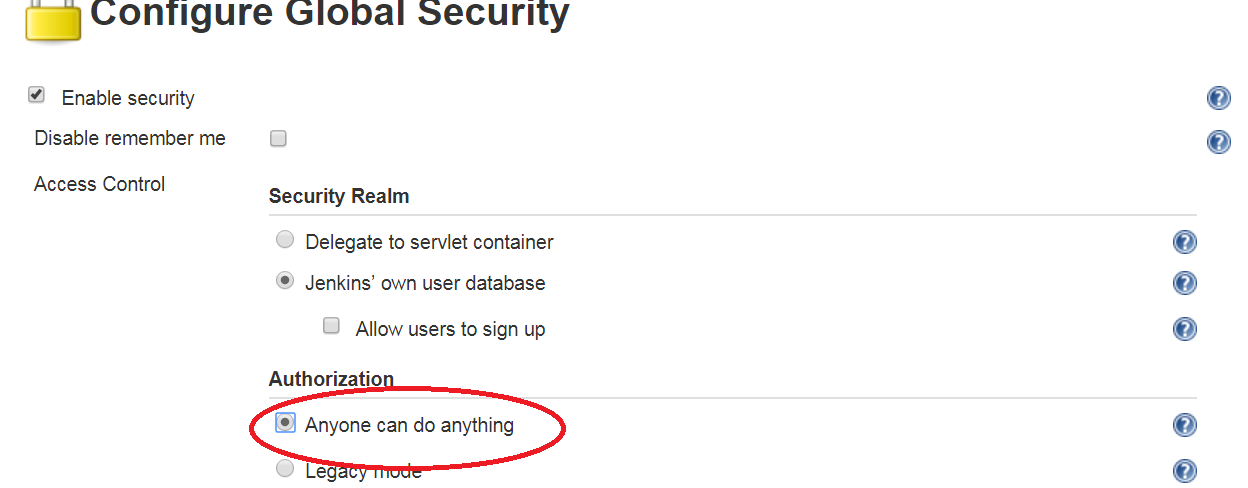
[Fig 2.2]

* Setting permission Reference [Fig 2.3-2.4 ]
* Select manage Jenkins
* Select Configure Global Security

****

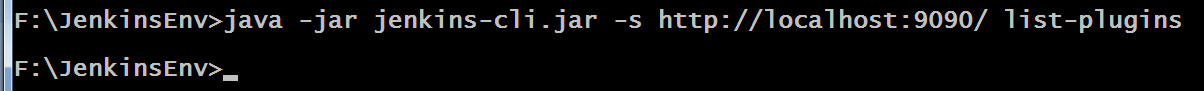
[Fig 2.3]

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

****

[2.4]

* Run the command again shown in [Fig 2.5 ]

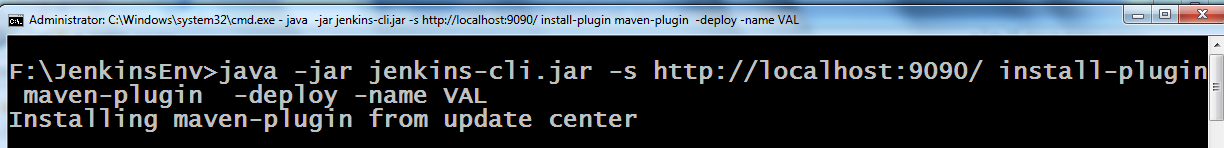
****

[2.5]

**Step 5:**

**Install plugins**

* Select Jenkins CLI install plugin option
* Show the command to install the plugin
* Open the command prompt
* Use command to install the plugins
* java -jar jenkins-cli.jar -s http://localhost:9090/ install-plugin maven-plugin -deploy -name VAL
* Maven plugin name is : maven-plugin Reference [Fig 2.6 ]

****

[Fig 2.6]

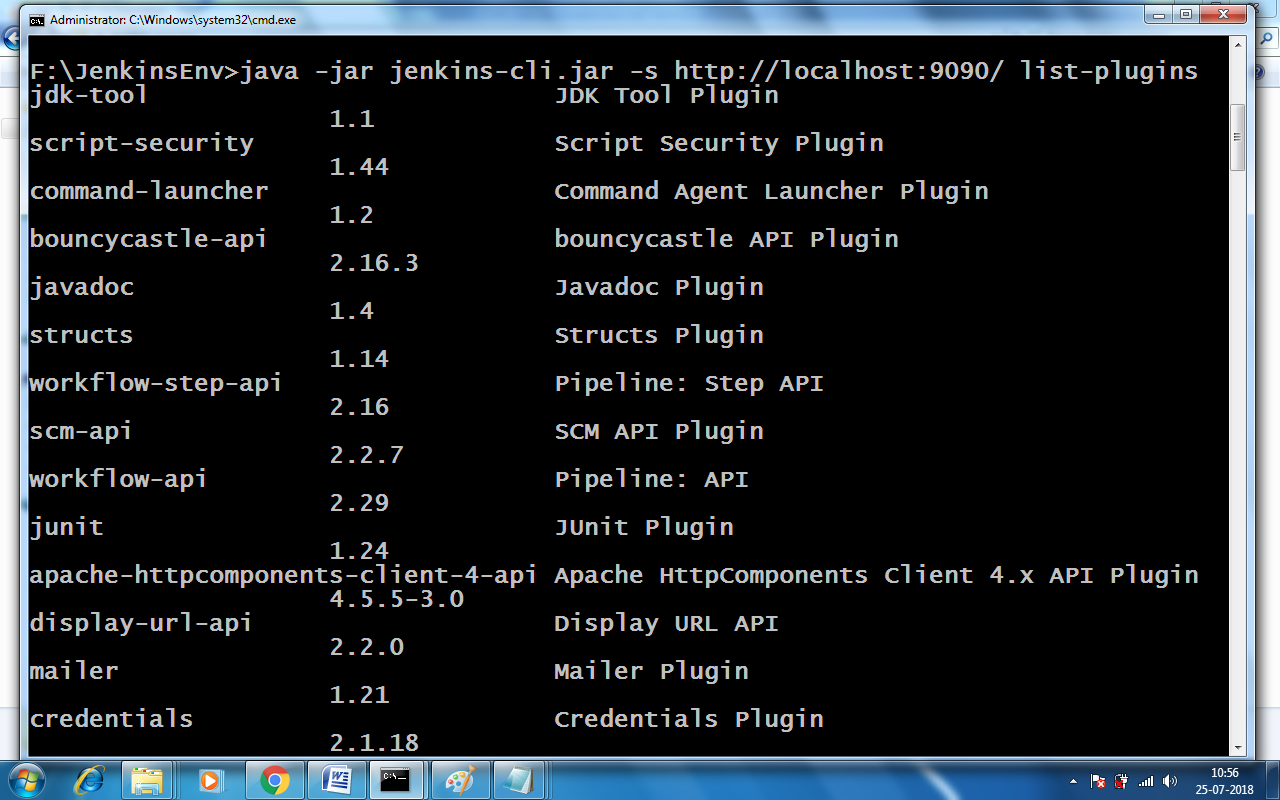
* Use same command to install the following required plugins

**Plugin name list:**

* + workflow-aggregator
  + git
  + deploy

* Open the command prompt
* Verify the installed plugins add following command Reference [Fig 2.7 ]

**Java –jar Jenkins-cli.jar –s** [**http://localhost:9090/**](http://localhost:9090/) **list-plugins**

****

[Fig 2.7]

Guided Exercise 2**:** **Create Maven Job in Jenkins**

**Estimated Completion Time:**20 Minutes

**Objective**: To create and configure Maven job in Jenkins.

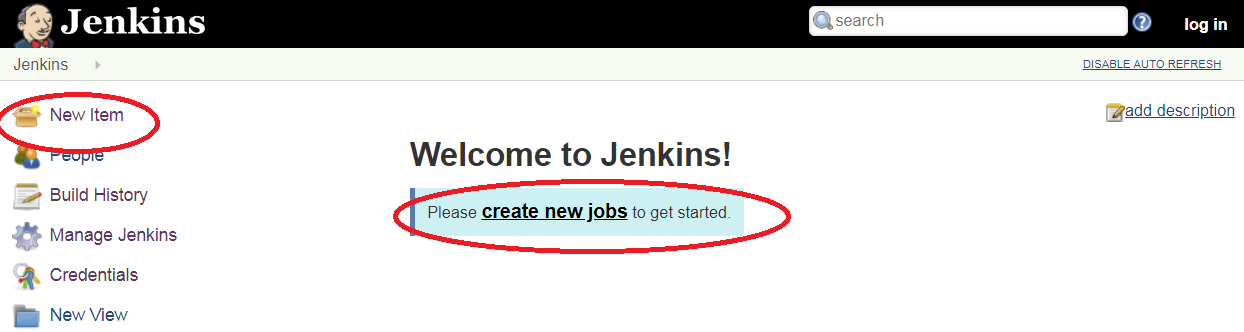
Steps to follow:

**Download jenkinsTemplateProject**



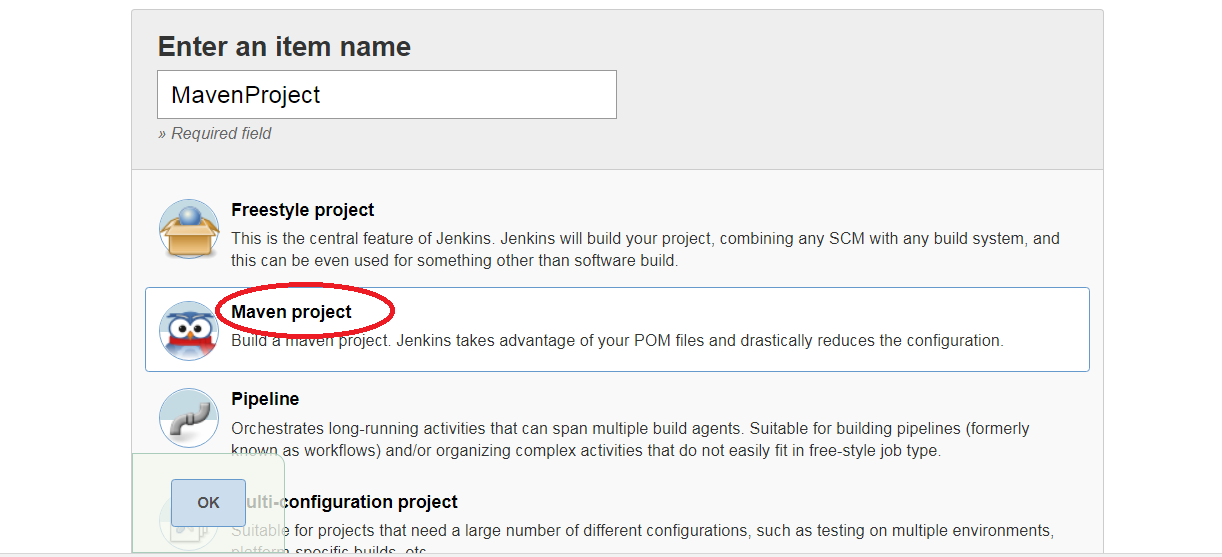
**Step 1:**

**Select create new jobs option (or New Item)** Reference [Fig 2.8 ]

****

[Fig 2.8]

* Select Maven Project (click ok) Reference [Fig 2.9 ]

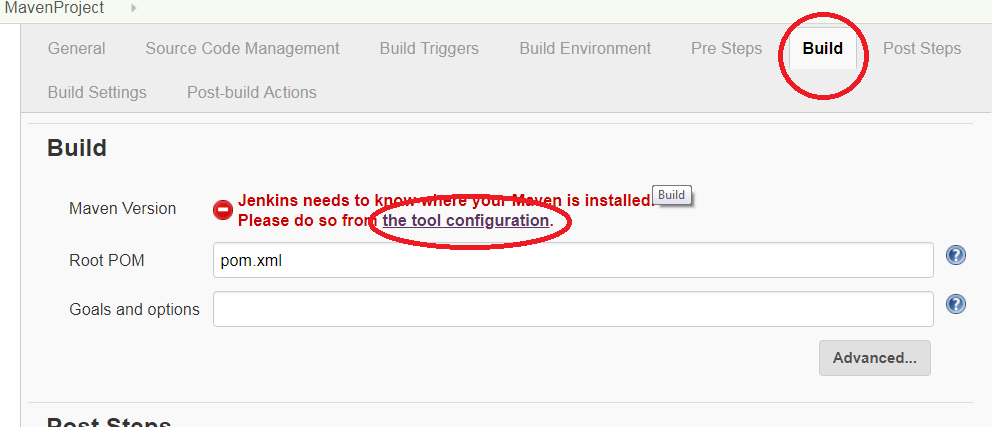


[Fig 2.9]

**Step 2:**

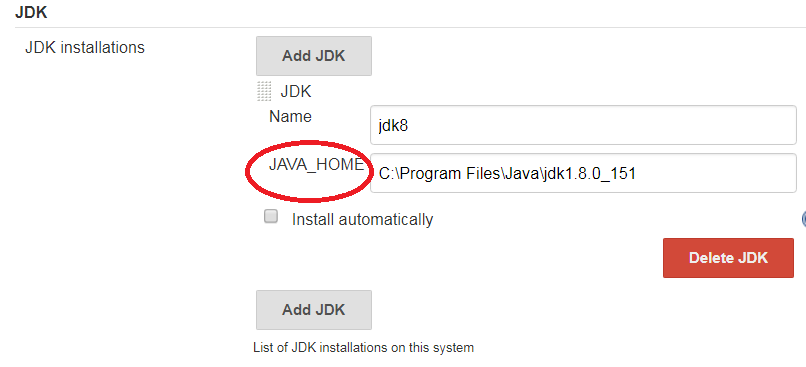
**Configure jdk 8 and Maven 3** Reference [Fig 3.0-3.2 ]

* Select the tool configuration
* Uncheck the install automatically

****

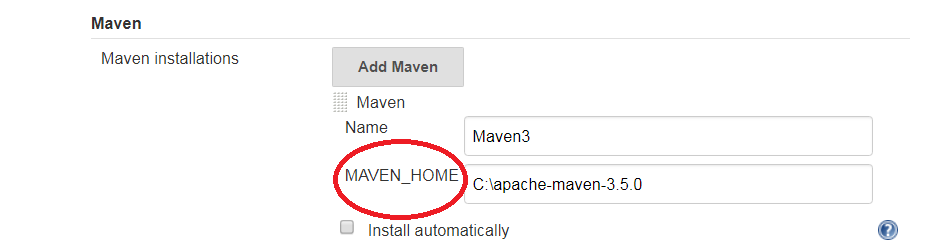
[Fig 3.0]

* Set JAVA\_HOME to jdk8 home path

****

[Fig 3.1]

* Set the Maven Home

****

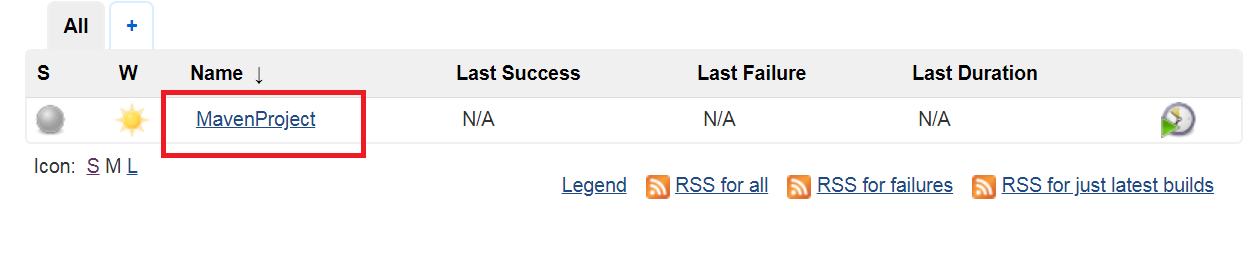
[Fig 3.2]

* Set the path for the git.exe (Eg C:\Program Files\Git\bin\git.exe)
* Apply and save the settings

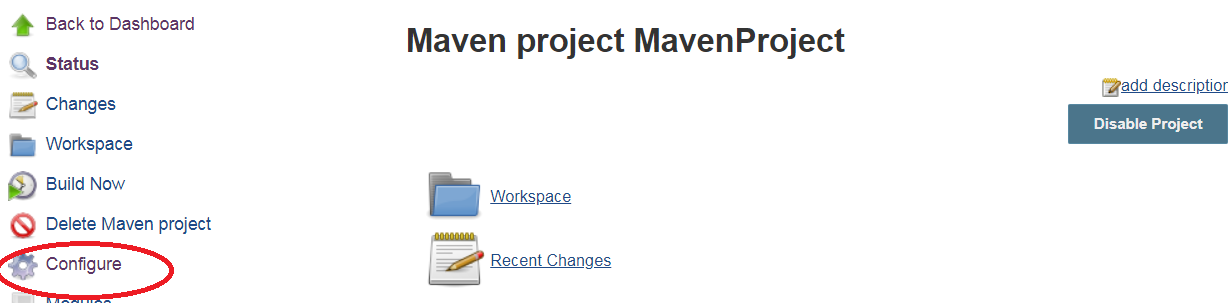
**Step 3:**

**Configure Maven Job**

* Select MavenProject from the Dashboard to configure Reference [Fig 3.3 ]

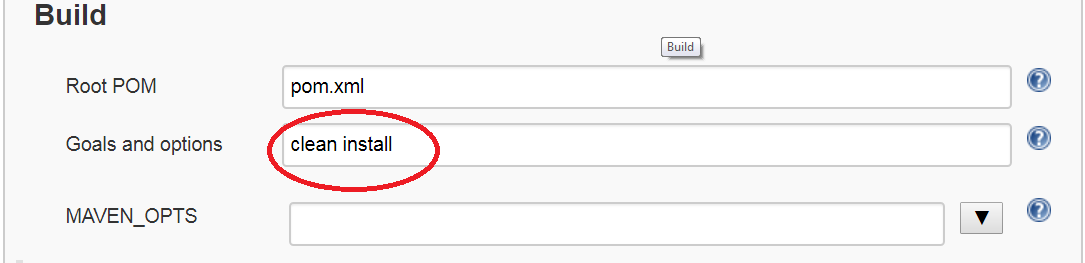


[Fig 3.3]



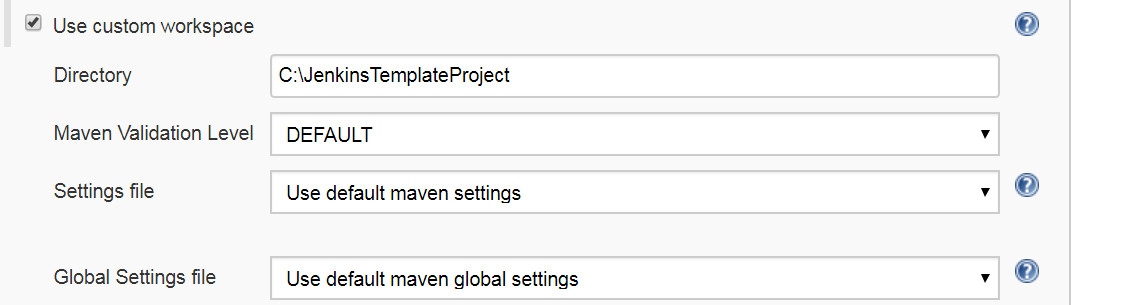
[Fig 3.4]

* Configure the build for maven project Reference [Fig 3.5 ]



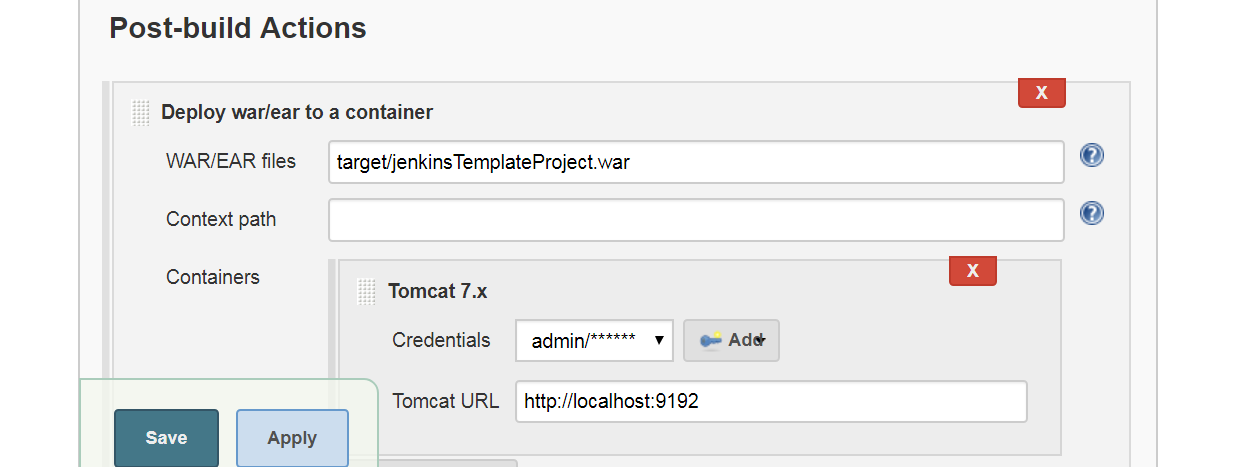
[Fig 3.5]

* Select advance tab to configure custom workspace Reference [Fig 3.6 ]



[Fig 3.6]

* Set the post build action for maven project. Reference [Fig 3.7 ]



[Fig 3.7]

**Step 4:**

**Configure Tomcat container**

* Set tomcat user for Jenkins auto container deployment
* Add user to tomcat-users.xml

F:\JenkinsEnv\apache-tomcat-7.0.90\apache-tomcat-7.0.90\conf\tomcat-users.xml

<role rolename="manager-gui"/>

<role rolename="manager-script"/>

<user username="admin" password="admin" roles="manager-gui,manager-

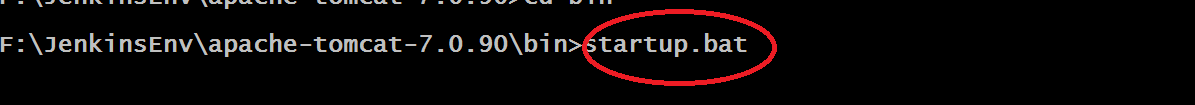
script"/>

* Add tomcat http port to server.xml (port =9192)

<Connector connectionTimeout="20000" port="9192" protocol="HTTP/1.1"

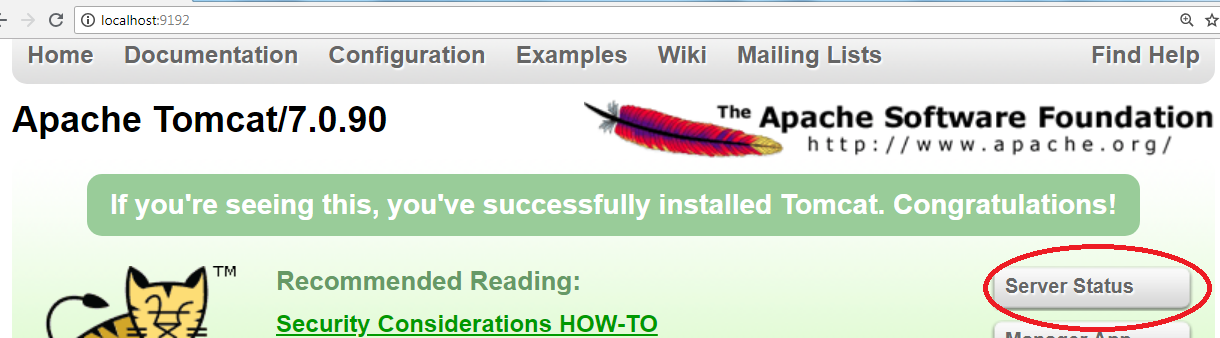
redirectPort="8443"/>

* Open command prompt change directory to tomcat bin directory
* Start tomcat server (Startup.bat) Reference [Fig 3.8 ]

****

[Fig 3.8]

* Select the server status to view the project. Reference [Fig 3.9 ]

****

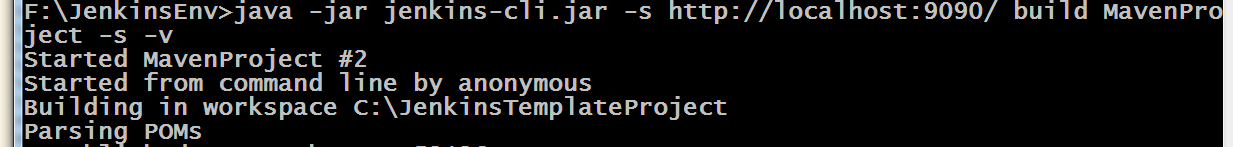
[Fig 3.9]

**Step 5:**

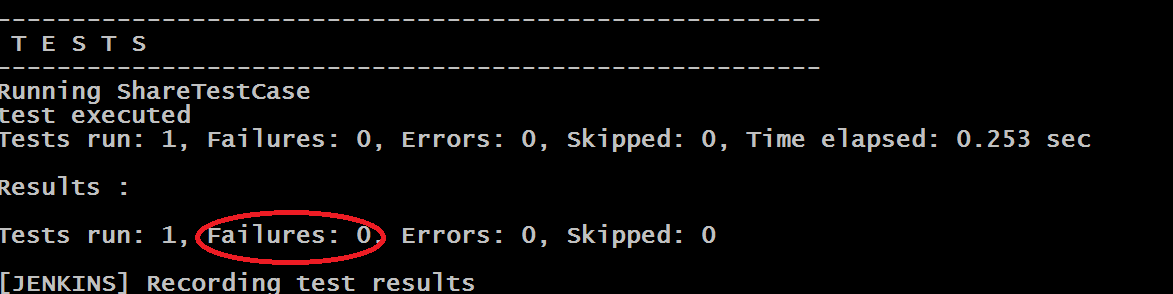
**Build maven project using CLI command** Reference [Fig 4.0-4.2 ]

* Open command prompt
* Change directory to F;\jenkinsEnv
* Use command: java –jar jenkines-cli.jar –s <http://localhost:9090> list-jobs
* Use command: java –jar Jenkins-cli.jar –s <http://localhost:9090/> build MavenProject –s -v

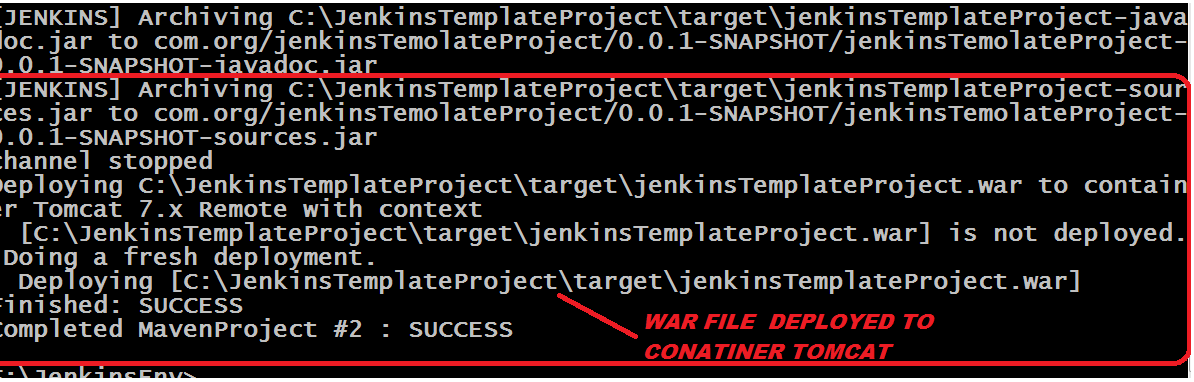
**Console Output:**

****

[Fig 4.0]

****

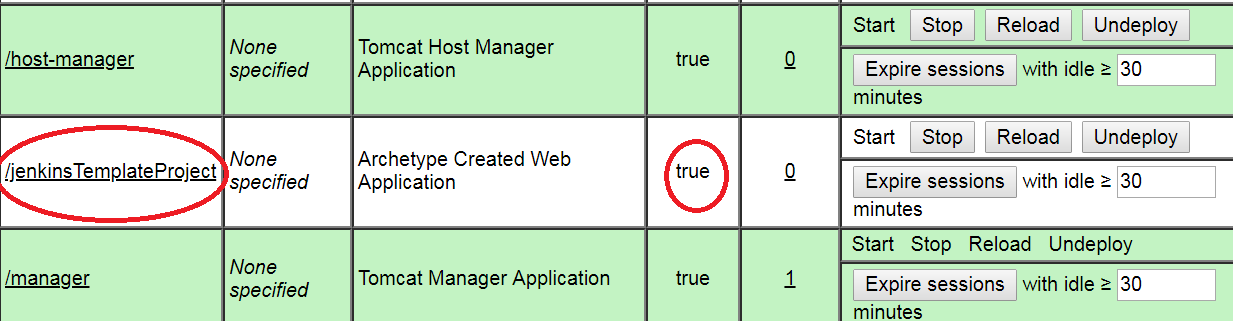
[Fig 4.1]

****

[Fig 4.2]

**Result: Maven job compiled – tested – auto deployed war file to tomcat using Jenkins**

Reference [Fig 4.3 ]

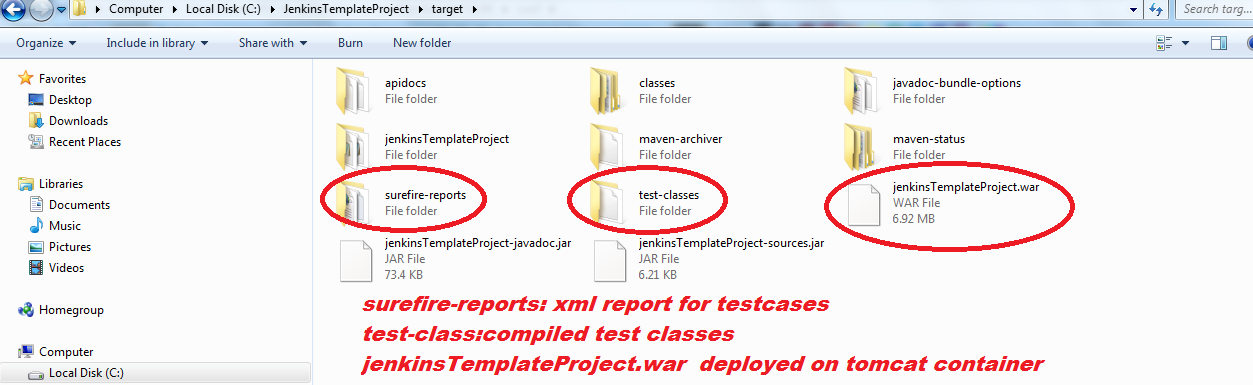
****

[Fig 4.3]

**Step 6:**

**Open the project folder (c:\jenkinsTemplateProject )** Reference [Fig 4.4 ]

Project target folder output

****

[Fig 4.4]

Guided Exercise 3**:** **Create Pipeline Job in Jenkins**

**Estimated Completion Time:** 10 Minutes

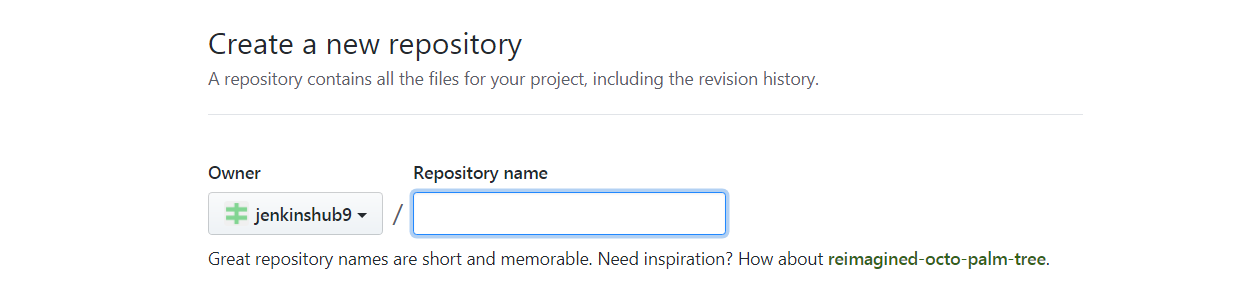
**Objective**: To create pipeline job in Jenkins using groovy declarative script to deploy maven

project form git-hub

Steps to follow:

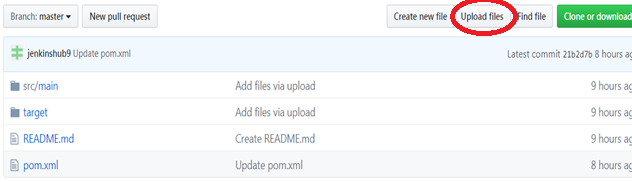
**Step 1:**

**GitHub Configuration**

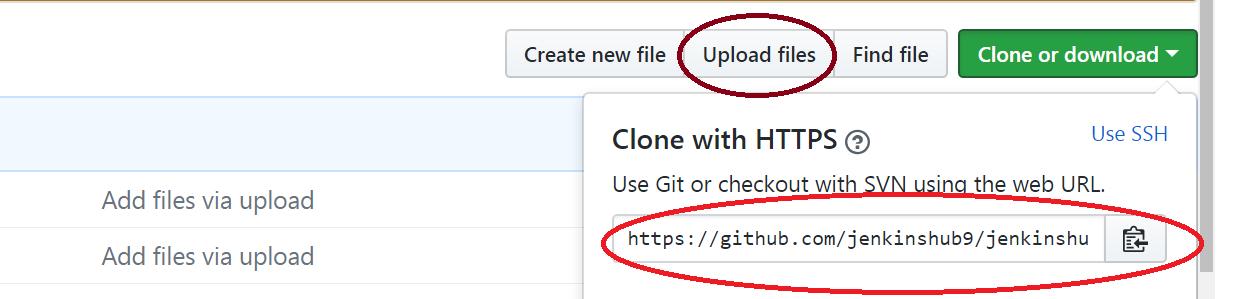
* Use the given git-hub repository url
* Git-hub repository is not provided then follow the steps
* Create account on git-hub (Using <https://github.com/>)
* Create public repository on git-hub Reference [Fig 4.5 ] ****

[Fig 4.5]

* Upload files to the repository
* C:\JenkinsTemplateProject (Select all folders and files and drag it to repository)
* Commit the changes Reference [Fig 4.6 ]

****

[Fig 4.6]

****

[Fig 4.7]

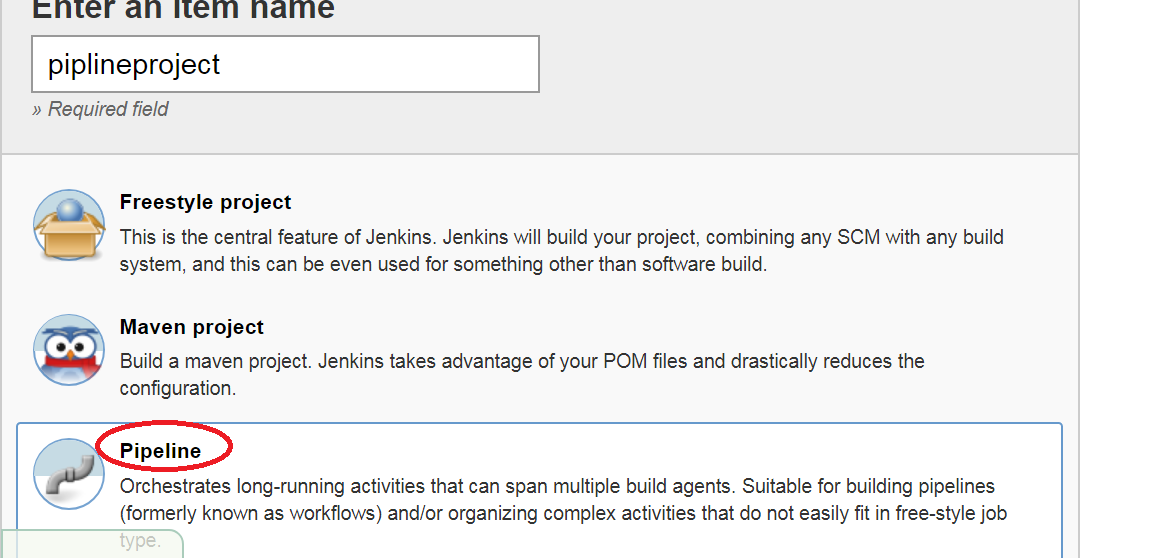
* Use the url to pull project data from git-hub Reference [Fig 4.7 ]

(E.g. https://github.com/jenkinshub9/jenkinshub.git)

**Step 2:**

**Create pipeline project**

* Select new items from Jenkins Dashboard Reference [Fig 4.8 ]

****

[Fig 4.8]

* Add pipeline script for the job Reference [Fig 4.9 ]

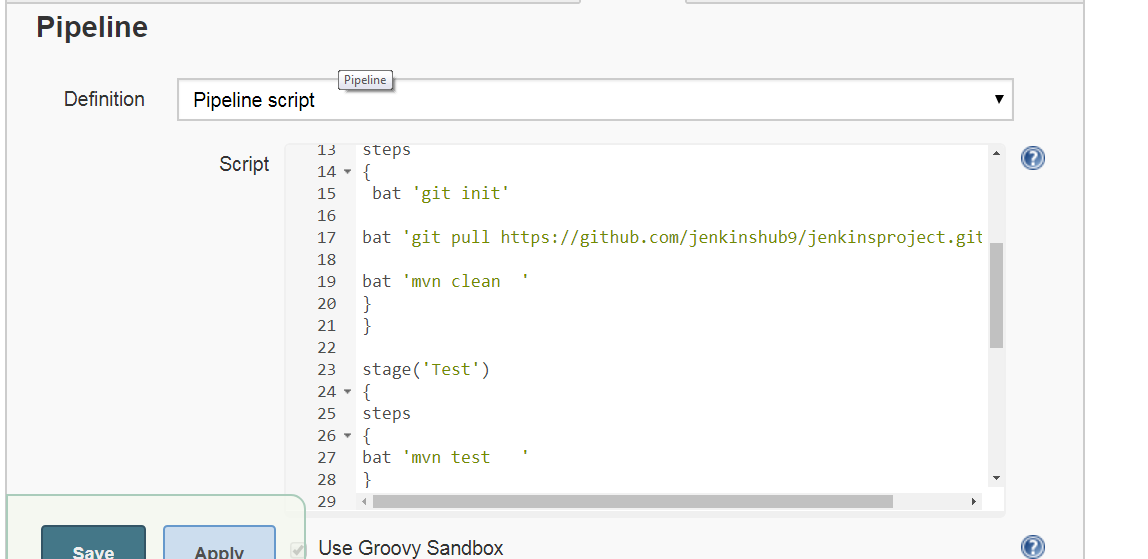
****

Fig 4.9]

* Copy the following Script for pipeline (apply and save the project)

pipeline{

agent any

tools {

maven 'Maven3'

jdk 'jdk8'

}

stages{

stage('clone repo'){

steps{

bat 'git init'

bat 'git pull https://github.com/jenkinshub9/jenkinsproject.git'

bat 'mvn clean'

} }

stage('Test'){

steps{

bat 'mvn test '

} }

stage('Deploy')

{

steps{

bat 'mvn package '

} } }

post {

always {

junit '\*\*/surefire-reports/\*.xml'

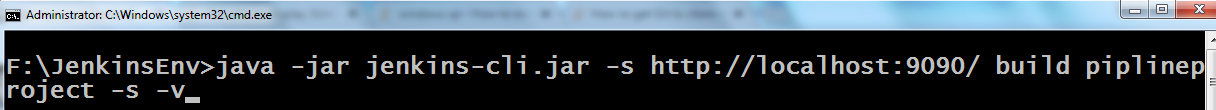
} } }

**Step 3:**

**Build the pipeline project using CLI command** Reference [Fig 5.0]

* Open command prompt
* Change directory to F:\jenkinsEnv
* Use command: java –jar jenkines-cli.jar –s <http://localhost:9090> list-jobs
* Use command: java –jar Jenkins-cli.jar –s <http://localhost:9090/> build piplineproject –s -v

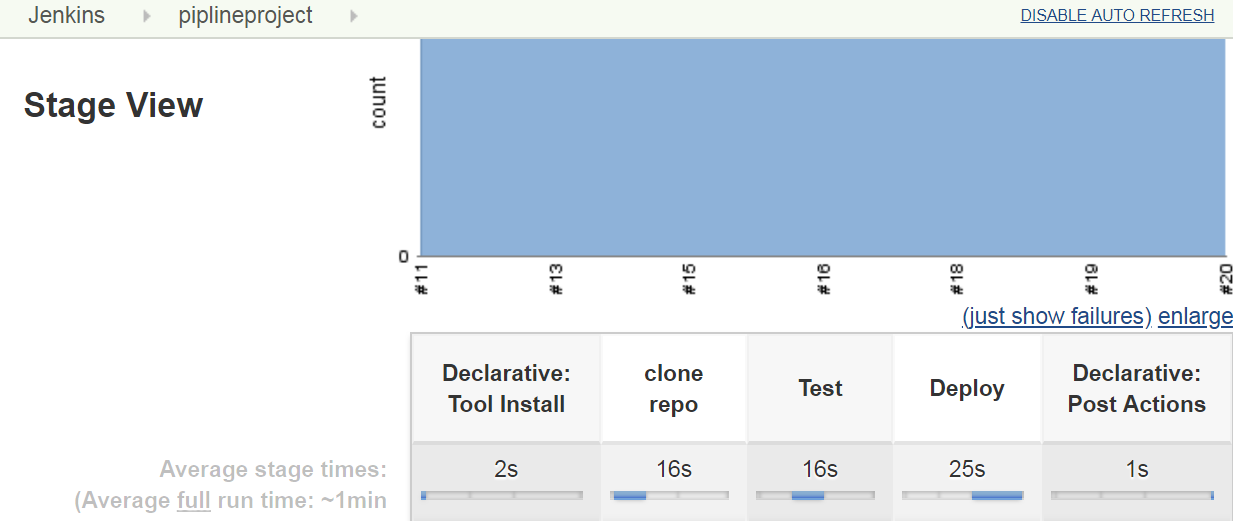
**Console Output:**

****

[Fig 5.0]

**Step 4:**

**RESULT:** Reference [Fig 5.1-5.2 ]

****

[Fig 5.1]

****

[Fig 5.2]

**Troubleshooting tips:**

1. Jenkins default port is 8080

Java –jar Jenkins.war (starts Jenkins at 8080 port)

To avoid port conflict start Jenkins with option --httpPort=<portnumber>

1. To run CLI command make sure Jenkins-cli.jar is present in working directory

Configure global security to get access to CLI ( if overall/read permission error is shown for CLI command)

1. Change tomcat server port to avoid conflict (server.xml)

Default port for tomcat is 8080

For maven exercise it is set to 9192

1. Jenkins fails to auto deploy war file to container

Check roles: manager-gui and manager-script in (tomcat-users.xml )

1. Uploading project to git-hub repository

Open the project folder c:\jenkinsTemplateProject

On git-hub repository select upload files

Drag and drop all the folders and file one by one ( src , pom.xml )

**Solution kit:**



**Summary:**

You have learnt to install and configure Jenkins job for DevOps Environment.