BE IT/ Semester 8/ Rev 2016/ ITL 803 DevOps lab/ FH2022
Name of the student:
Roll number:
Experiment number:
Aim: To deploy a Java Web application using Jenkins.

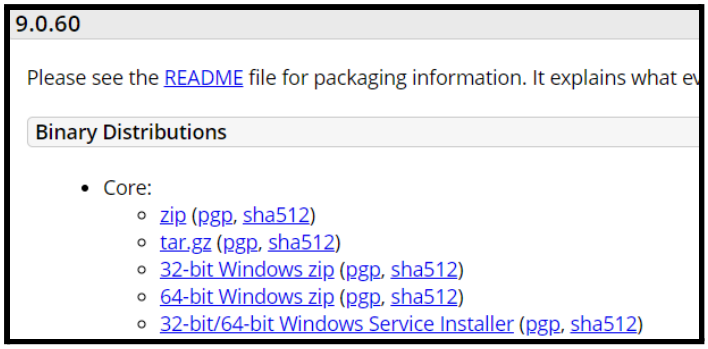
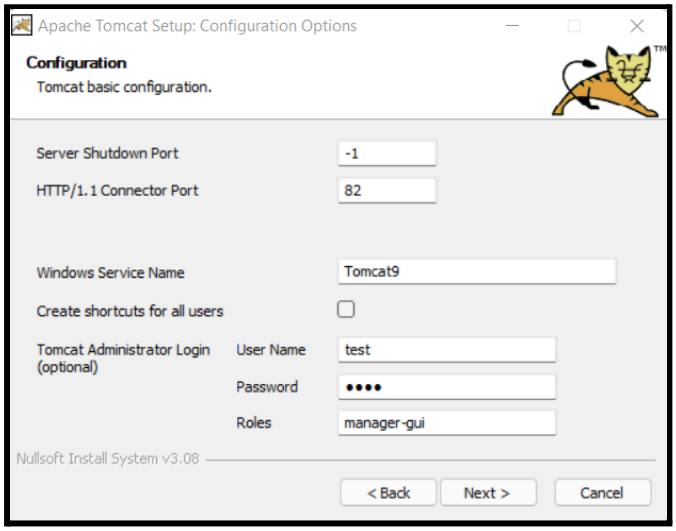
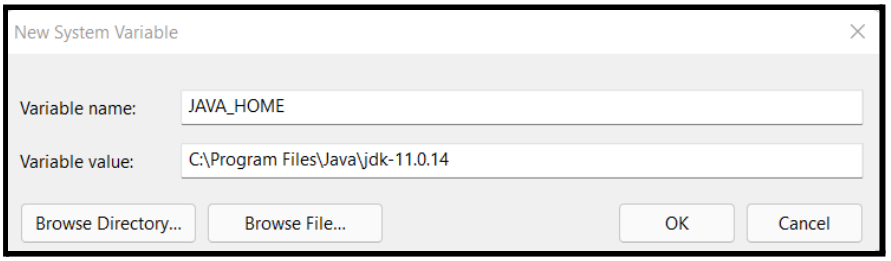
- **Software requirements**
  1. Java: see the Java Requirements page
  2. Maven
  3. Apache Tomcat
  4. Git
  5. Web browser: see the Web Browser Compatibility page
  6. For Windows operating system: Windows Support Policy
  7. For Linux operating system: Linux Support Policy
- **Hardware requirements (if any, the minimum requirements)**
  1. 256 MB of RAM
  2. 1 GB of drive space (although 10 GB is a recommended minimum if running Jenkins as a Docker container)

Steps of the experiment

Step#	Step details (with unique snapshot if produced any at the respective step)
1	Download maven from the official website https://maven.apache.org/ Go to download section and download the binary zip archive such as apache-maven-3.8.4-bin.zip Link Binary tar.gz archive — apache-maven-3.8.5-bin.tar.gz Binary zip archive — apache-maven-3.8.5-bin.zip Source tar.gz archive — apache-maven-3.8.5-src.tar.gz Source zip archive — apache-maven-3.8.5-src.zip
2	extract the folder at particular drive and set the environment variable. To set the environment variable go to extracted folder-> Apache maven -> bin Copy the path and go on system environment variable setting. Click on environmental variable, select path and click on edit. Add a new path and paste the maven path. C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\170... C:\Program Files (x86)\Common Files\Siemens\ C:\Program Files (x86)\Common Files\Siemens\Interfaces C:\Program Files (x86)\Common Files\Siemens\ACE\Interfaces E:\Sem 8\DevOps\Prac Material\apache-maven-3.8.5-bin\ap... OK Cancel

3	To check maven is configured or not on system, go on command prompt (open command prompt as administrator) and type mvn -v , If you find the maven Version then maven system is installed and configured successfully.
	`E:\> mvn -v`
	`Apache Maven 3.8.5 (3599d3414f046de2324203b78ddcf9b5e4388aa0)`
	`Maven home: E:\Sem 8\DevOps\Prac Material\apache-maven-3.8.5-bin\apache-maven-3.8.5`
	`Java version: 11.0.14, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-11.0.14`
	`Default locale: en_IN, platform encoding: Cp1252`
	`OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"`
	`E:\>`

4	Now, Download Apache Tomcat from the official website. Visit https://tomcat.apache.org/ . On the left side panel

	you can see various tomcat versions, we will refer to Tomcat 9.0 click on it. Scroll down the page till the binary distribution section. 32-bit/64-bit Windows Service Installer click on this installer. The .exe file will download. 
5	Once it is downloaded, install the Apache Tomcat. While installation it will ask you for username & password and also the port number. Remember the port number. Username: test Password: 12345 Click on next and finish the procedure. 
6	For Apache Tomcat configuration it will ask for the JAVA_HOME variable. To set JAVA_HOME variable, go on drive at which you installed Java mostly in c drive. Copy the path such as local disk c-> program files -> Java -> jdk 11.0.8 Copy the path and open system environment variable setting. Add new variable where variable name will be JAVA_HOME and paste the path at the path section. To check JAVA_HOME path is configured or not properly, open command prompt as administrator and type java -version it will show the Java version and Java_home version. 
7	Copy jenkins.war file at Apache Tomcat webapps folder.

BE IT/ Semester 8/ Rev 2016/ ITL 803 DevOps lab/ FH2022
Name of the student:
Roll number:
Experiment number:
Aim: To deploy a Java Web application using Jenkins.

<table>
<tr><td></td><td>

Name	Date modified	Type	Size
docs	16-03-2022 20:03	File folder	
manager	16-03-2022 20:03	File folder	
ROOT	16-03-2022 20:03	File folder	
jenkins.war	16-03-2022 13:58	WAR File	89,087 KB

</td></tr>
<tr><td>8</td><td>

Now, go on Apache Tomcat folder -> bin and copy the path, open the command prompt as administrator and change the directory as per copied path and type the command startup. The system will open Apache Tomcat server.

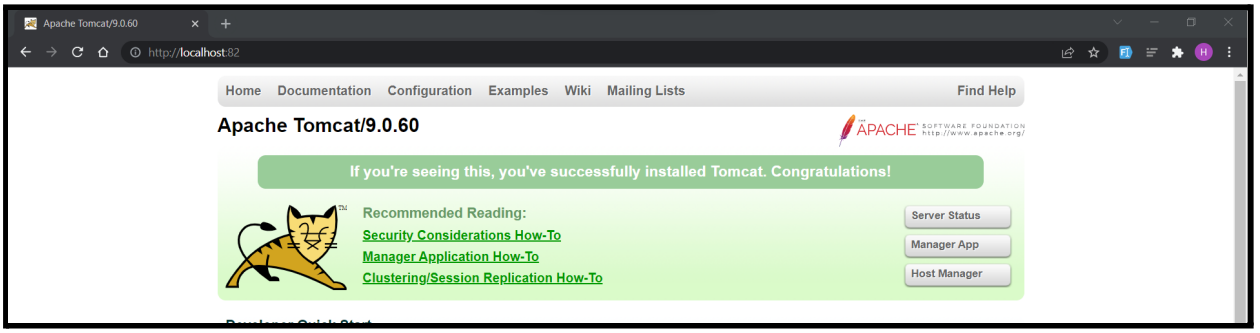
```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.22000.556]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>cd C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin
```

```
C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin>
```

```
metadata
16-Mar-2022 20:20:26.664 INFO [Download metadata thread] hudson.model.AsyncPeriodicWork.lambda$doRun$1 Finished Download
 metadata. 5 ms
16-Mar-2022 20:20:26.737 INFO [pool-6-thread-10] jenkins.InitReactorRunner$1.onAttained Completed initialization
16-Mar-2022 20:20:26.769 INFO [Jenkins initialization thread] hudson.lifecycle.Lifecycle.onReady Jenkins is fully up and
 running
```

</td></tr>
<tr><td>9</td><td>

command prompt will show the port number for Apache Tomcat. To check Apache Tomcat installed successfully or not then type http://localhost:portnumber/Jenkins
Jenkins will ask you administrator password, paste the password and you can see Tomcat server successfully installed as a message.
If you can't find the port then go on Apache tomcat folder and open conf folder and edit server.xml and change the port as per your wish.



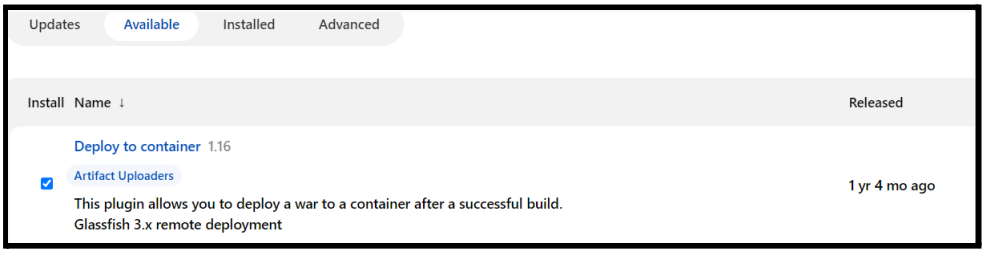
</td></tr>
<tr><td>10</td><td>

Now close the Apache tomcat and open Jenkins dashboard. To open Jenkins dashboard
First we will run the war file through the command prompt and then open the browser.
Type jenkins port such as http://localhost:portnumber and it will open the login page. Type the username and password.

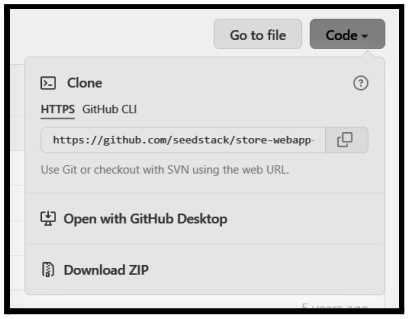
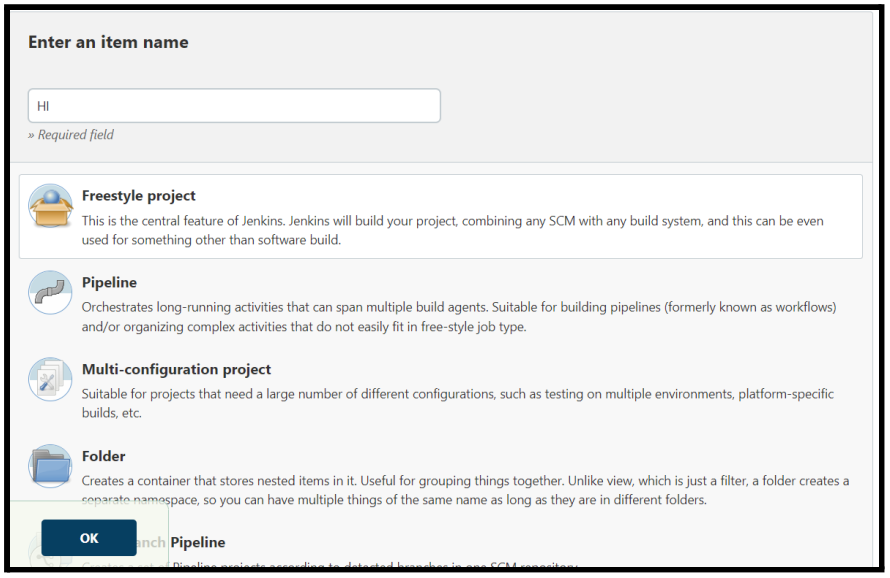
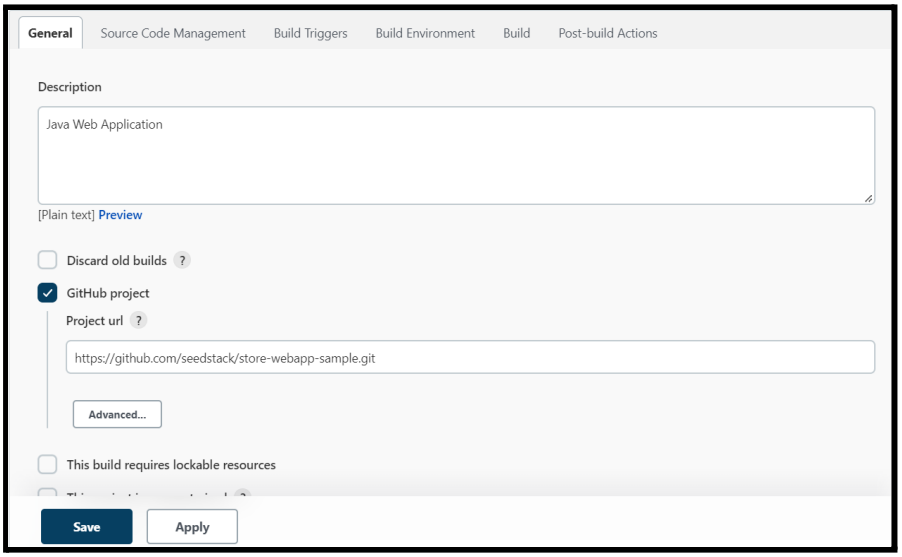
```
E:\Sem 8\DevOps\Prac Material>java -jar jenkins.war --httpPort=9191
Running from: E:\Sem 8\DevOps\Prac Material\jenkins.war
webroot: $user.home/.jenkins
```

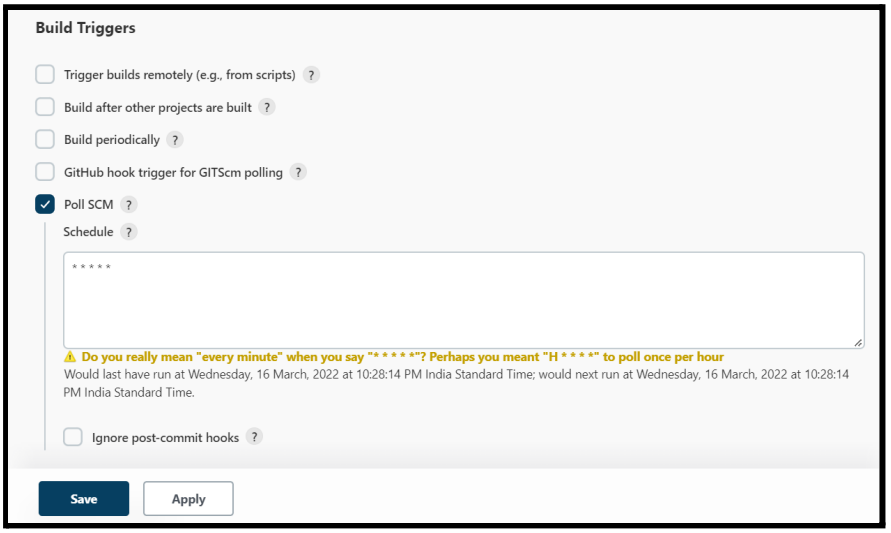
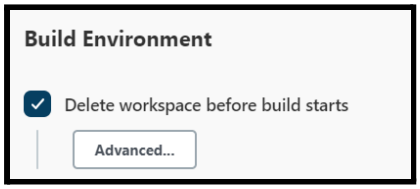
</td></tr>
</table>

BE IT/ Semester 8/ Rev 2016/ ITL 803 DevOps lab/ FH2022
Name of the student:
Roll number:
Experiment number:
Aim: To deploy a Java Web application using Jenkins.

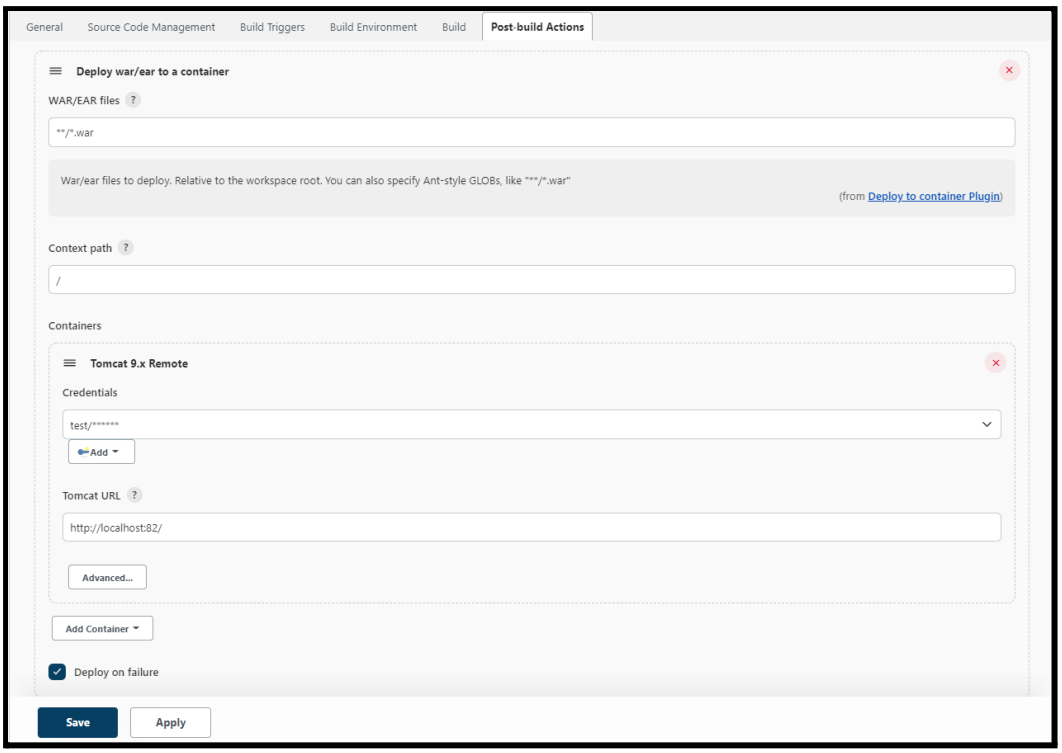
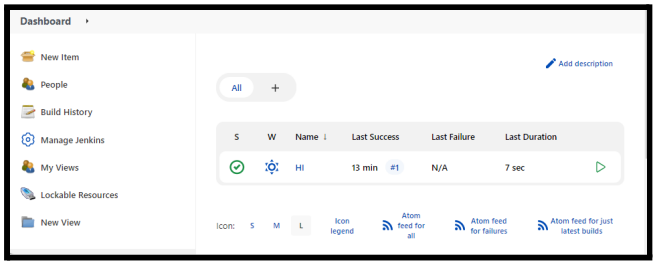
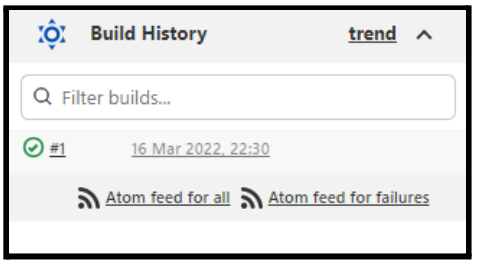
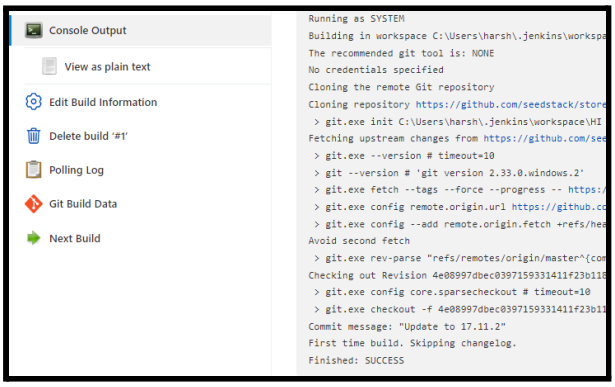
	
11	click on manage Jenkins and visit the configure tool and Add JDK Installer, name will be JDK name such as JDK 11.0.8 and path will be your system JDK path. Also Add maven installer, name will be maven name such as maven-3.0.8 and path will be your system maven path.  
12	visit the manage plugin section and click on the available tab and install Deploy to container plugin. And make sure the git plugin is also enabled. 

BE IT/ Semester 8/ Rev 2016/ ITL 803 DevOps lab/ FH2022
Name of the student:
Roll number:
Experiment number:
Aim: To deploy a Java Web application using Jenkins.

13	Open a new tab and open this git link https://github.com/seedstack/store-webapp-sample and copy the .git repository URL. 
14	Now back to Jenkins dashboard and click on create new item. Then Enter the project name and select freestyle project(which type of your project), then click on ok. 
15	In the description, the section adds about your project, Such as "sample web application deployment" and Tick mark on GitHub project checkbox. add the repository of your project URL. 

16	In Source Code Management click on git and add git repository url. In build triggers section tick mark on poll SCM (here you can schedule your job (like Cron jobs) here * * * * * means for every one minute clean and build and deploy the project into a tomcat server.  
17	In the Build Environment section tick mark on Delete workspace before build starts. 
18	In the build section select invoke top-level maven targets(here I am creating the maven project right that is why here I am selecting maven targets). In the maven version select maven Version which we configured to manage Jenkins. And type install at goal section 

BE IT/ Semester 8/ Rev 2016/ ITL 803 DevOps lab/ FH2022
Name of the student:
Roll number:
Experiment number:
Aim: To deploy a Java Web application using Jenkins.

19	In Post-build Actions add username and password which we used for Apache Tomcat server and select the deploy war/ear to a container, in WAR/EAR files section add the **/*.war(it means fetch the war file from the workspace). Click on save and apply buttons and then click on the build now button. your build process is running after the successful build (it will show the blue button) if the build fails it will show the red button 
20	Check the console output. After successful deployment, you can check your tomcat and test the application whether it's successfully deployed or not.   

Conclusion: Hence, we have successfully deployed a Java Web application using Jenkins.