



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Academic Year: 2023-24

Semester: V

Class / Branch: TEIT

Subject: DevOps Lab

Name of Instructor: Prof. Sujata Oak

Experiment No. 5

Aim: To create and build a CI/CD pipeline in Jenkins to test and deploy an application over the tomcat server.

Theory:

Understanding Pipeline Configuration in Jenkins

Jenkins is an open-source automation server used to automate parts of the software development process, including building, testing, and deploying code. A Jenkins pipeline is a suite of plugins that supports implementing and integrating continuous delivery pipelines.

- 1. Pipeline as Code:** Jenkins pipelines are typically defined using a domain-specific language (DSL) known as Pipeline DSL, which can be written in a Jenkinsfile.

```
pipeline {
    agent any
    stages {
        stage('Build') {
            steps {
                echo 'Building...'
                // Build steps go here
            }
        }
        stage('Test') {
            steps {
                echo 'Testing...'
                // Test steps go here
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying...'
                // Deploy steps go here
            }
        }
    }
}
```

Compiled By: Prof. Sujata Oak



```
}
```

Declarative vs. Scripted Pipeline: Jenkins supports two types of syntax for defining pipelines:

- **Declarative Pipeline:** A more simplified and opinionated syntax (as shown above) that is easier to read and write.
- **Scripted Pipeline:** A more flexible and expressive syntax that uses Groovy code.

Stages and Steps: Pipelines consist of stages that define major phases (e.g., Build, Test, Deploy) and steps that define individual tasks within those stages.

Agents: Define where the pipeline or a specific stage runs. It can be any Jenkins agent or a specific node.

Triggers and Parameters: Pipelines can be triggered by various events (e.g., SCM commits, schedule) and can accept parameters to customize the execution.

Pipeline Plugins: Jenkins has numerous plugins that enhance the functionality of pipelines, such as plugins for integrating with SCM, notifications, and deployment tools.

IMPLEMENTATION: CONFIGURING JENKINS PIPELINE

Once Jenkins is installed follow the below steps:

Step1: Now, we need to specify the Java location to Jenkins. Go back to your server command prompt and use the code below to fetch the directory of Java. Multiple directories will be listed using the below code. In our case, the directory is: '/usr/lib/jvm/java-11-openjdk-amd64/bin/java'.

```
find / -type f -name java
```

```
root@Ubuntu:/home/sujata/Desktop/JENKINS_LAB# find / -type f -name java
/var/lib/dpkg/alternatives/java
/snap/snapd/21759/usr/lib/snapd/apparmor.d/abstractions/ubuntu-browsers.d/java
/snap/core20/1828/etc/apparmor.d/abstractions/ubuntu-browsers.d/java
/snap/core20/1828/usr/share/bash-completion/completions/java
/snap/core20/2318/etc/apparmor.d/abstractions/ubuntu-browsers.d/java
/snap/core20/2318/usr/share/bash-completion/completions/java
/snap/core22/1380/etc/apparmor.d/abstractions/ubuntu-browsers.d/java
/snap/core22/1380/usr/share/bash-completion/completions/java
/snap/core22/1439/etc/apparmor.d/abstractions/ubuntu-browsers.d/java
/snap/core22/1439/usr/share/bash-completion/completions/java
find: '/run/user/1000/doc': Permission denied
find: '/run/user/1000/gvfs': Permission denied
/etc/apparmor.d/abstractions/ubuntu-browsers.d/java
/usr/share/bash-completion/completions/java
/usr/lib/jvm/java-11-openjdk-amd64/bin/java
```



Step 2) Copy the above location and go back to your Jenkins Dashboard. Look for Global Tool Configuration under the Manage Jenkins menu, as shown in the image below.

New version of Jenkins (2.462.1) is available for download ([changelog](#)).

Warnings have been published for the following currently installed components:

Jenkins 2.452.3 core and libraries:
[Multiple security vulnerabilities in Jenkins 2.470 and earlier, LTS 2.452.3 and earlier](#)
A fix for this issue is available. Update Jenkins now.

System Configuration

System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

Step 3) Unselect the Install automatically button from the JDK window and fill the fields. Paste the java location path and trim it as shown in the image below. In the below Git window, select the install automatically checkbox.



Dashboard > Manage Jenkins > Tools

JDK installations ^ Edit Edited

Add JDK

≡ JDK

Name

java_home



JAVA_HOME

/usr/lib/jvm/java-11-openjdk-amd64/



Install automatically ?



≡ **Git**

Name

Default

Path to Git executable ?

git ✓

Install automatically ?

Add Installer ▾

Step 4) Similarly, check the box for Maven and fill the name field. Save all the settings, and now the configuration of Jenkins is completed.



Dashboard > Manage Jenkins > Tools

Maven installations

Maven installations ^ Edited

Add Maven

Maven

Name

maven_home ✓



Install automatically ?

Install from Apache

Version

3.8.3 ✓

Save

Apply

Create CI/CD Pipeline Jenkins

We can now start creating pipelines using Jenkins after all the configuration and setup.
Continuous Integration



Step 1) Create New Item, select Freestyle Project and provide a name to your item.

Dashboard > All >

Enter an item name

CICD_Pipeline_Demo ✓

» Required field

Freestyle project ✓

Classic, general-purpose job type that archiving artifacts and sending email n

Pipeline

Orchestrates long-running activities th and/or organizing complex activities th

Multi-configuration project

Suitable for projects that need a large etc.

OK

Step 2) Switch to the Source Code Management window and paste your Github repository link. Specify your branch name of the repository below and Save it.



[Note: The above-linked Github repository 'github.com/subu123321/hello-world' contains a 'pom.xml' file used for Java compilation and generates a web app. It will be deployed to the server]

The screenshot shows the Jenkins 'Configure' screen for a pipeline job named 'CICD_Pipeline_Demo'. The left sidebar lists configuration sections: General, Source Code Management (highlighted with a red box), Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main panel shows the 'Git' configuration for the repository. The 'Repository URL' is set to 'https://github.com/subu123321/hello-world.git'. Under 'Credentials', it says '- none -'. There is an 'Advanced' dropdown and a 'Add Repository' button. In the 'Branches to build' section, the 'Branch Specifier' is set to '/master'. At the bottom, there are 'Save' and 'Apply' buttons, with 'Save' highlighted with a red box.

Step 3) Now click on Build Now button from the menu. With this step, all the repository files will be fetched by Jenkins. Click on Configure to go back to the same settings page.



Dashboard > CICD_Pipeline_Demo >

Status

Changes

Workspace

Build Now

Configure

Delete Project

Rename



Build History

trend



Filter...

/



#1

Aug 11, 2024, 3:48 PM



Dashboard > CICD_Pipeline_Demo > #1 > Console Output

</> Changes

Console Output

View as plain text

Edit Build Information

Delete build '#1'

Timings

Git Build Data

```
Started by user SUJATA OAK
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/CICD_Pipeline_Demo
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
  Cloning repository https://github.com/subu123321/hello-world.git
    > git init /var/lib/jenkins/workspace/CICD_Pipeline_Demo # timeout=10
  Fetching upstream changes from https://github.com/subu123321/hello-world.git
    > git --version # timeout=10
    > git --version # 'git version 2.25.1'
    > git fetch --tags --force --progress -- https://github.com/subu123321/hello-world.git +refs/heads/*:refs/remotes/origin/* # timeout=10
    > git config remote.origin.url https://github.com/subu123321/hello-world.git # timeout=10
    > git config -add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
  Avoid second fetch
    > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
  Checking out Revision f1bab369cca5c7de2edf86f676c3c3ac51613b3c (refs/remotes/origin/master)
    > git config core.sparsecheckout # timeout=10
    > git checkout -f f1bab369cca5c7de2edf86f676c3c3ac51613b3c # timeout=10
Commit message: "Add files via upload"
First time build. Skipping changelog.
Finished: SUCCESS
```

Step 4) Click on Build Tab and select build step as ‘Invoke top-level Maven targets’.



Poll SCM ?

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Environment

Delete workspace before build starts

Use secret text(s) or file(s) ?

Filter

Execute Windows batch command

Execute shell

Invoke Ant

Invoke Gradle script

Invoke top-level Maven targets

Run with timeout

Set build status to "pending" on GitHub commit

Add build step ^

Step 5) Select your maven name from the drop-down menu. Fill the goals with the multiple jobs you need to perform and separate them with one space. These goals are available in your repository, and you need to invoke them using Maven. Save it and again click on the '*Build Now*' button from the menu as we did in the previous steps. Now the maven commands will be executed that will generate a war file.



Dashboard > CICD_Pipeline_Demo > Configuration

- Inspect build log for published build scans
- Terminate a build if it's stuck
- With Ant ?

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Build Steps

Invoke top-level Maven targets ?

Maven Version

maven_home

Goals

clean compile package

Advanced ▾

Add build step ▾

Post-build Actions

Save

Apply

Click-On Build Now:



A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Dashboard > CICD_Pipeline_Demo > #2 > Console Output

```
WARNING: Illegal reflective access by com.thoughtworks.xstream.core.util.Fields (file:/var/lib/jenkins/.m2/repository/com/thoughtworks/xstream/xstream/1.3.1/xstream-1.3.1.jar) to field java.util.Properties.defaults
WARNING: Please consider reporting this to the maintainers of com.thoughtworks.xstream.core.util.Fields
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Packaging webapp
[INFO] Assembling webapp [webapp] in [/var/lib/jenkins/workspace/CICD_Pipeline_Demo/webapp/target/webapp]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/CICD_Pipeline_Demo/webapp/src/main/webapp]
[INFO] Webapp assembled in [20 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/CICD_Pipeline_Demo/webapp/target/webapp.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO] -----
[INFO] Reactor Summary for Maven Project 1.0-SNAPSHOT:
[INFO]
[INFO] Maven Project ..... SUCCESS [ 1.927 s]
[INFO] Server ..... SUCCESS [ 8.677 s]
[INFO] Webapp ..... SUCCESS [ 1.673 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 12.544 s
[INFO] Finished at: 2024-08-11T15:58:53+05:30
[INFO] -----
[INFO] Finished: SUCCESS
```

Step 6) If you want to check the war file created in the previous steps, visit the workspace on your Jenkins dashboard or just run the directory commands in your server. Your directories and project name can vary, so you can use the ‘ls’ command to see the list inside that directory and also keep in mind the directory name is case sensitive.

```
root@Ubuntu:/home/sujata/Desktop/JENKINS_LAB# cd /var/lib/jenkins/workspace/
root@Ubuntu:/var/lib/jenkins/workspace# ls
CICD_Pipeline_Demo  demo_pipeline@tmp  RightTrianglePattern
demo_pipeline      Factorial_Program  RightTrianglePattern-1
demo_pipeline@2    FirstJavaApp       RightTrianglePattern-I
demo_pipeline@2@tmp HelloWorld
```

```
root@Ubuntu:/var/lib/jenkins/workspace# cd CICD_Pipeline_Demo/
root@Ubuntu:/var/lib/jenkins/workspace/CICD_Pipeline_Demo# ls
```

```
azure-pipelines.yml  pom.xml   server          webapp
Dockerfile           README.md  sonar-project.properties  webapp.war
```



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
root@Ubuntu:/var/lib/jenkins/workspace/CICD_Pipeline_Demo# cd webapp
root@Ubuntu:/var/lib/jenkins/workspace/CICD_Pipeline_Demo/webapp# ls
pom.xml  src  target
```

```
root@Ubuntu:/var/lib/jenkins/workspace/CICD_Pipeline_Demo/webapp# cd target/
root@Ubuntu:/var/lib/jenkins/workspace/CICD_Pipeline_Demo/webapp/target# ls
maven-archiver  surefire  webapp  webapp.war
```

Step 7) Now go back to Configure and visit the ‘*Post Build Actions*‘ tab. Click the drop-down and select ‘*Archive the Artifacts*‘ from the options. In the field, write down ‘***/*.war*‘ as shown in the image below. It will fetch all the directories and get the war file wherever it is present. Click again on Build Now button, and you will now see the Artifacts in the Jenkins dashboard.



Dashboard > CICD_Pipeline_Demo > Configuration

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Invoke top-level Maven targets [?](#)

Maven Version

maven_home

Goals

clean compile package

Advanced ▾

Add build step ▾

Post-build Actions

Add post-build action ▾ ✓



Configure

General

Advanced ▾

Source Code Management

Add build step ▾

Build Triggers

Post-build Actions

Build Environment

Build Steps

Post-build Actions

Archive the artifacts

Files to archive ?

**/*.war

Advanced ▾

Add post-build action ▾

Save

Apply



Dashboard > CICD_Pipeline_Demo > #3 > Console Output

```
thoughtworks/xstream/xstream/1.3.1/xstream-1.3.1.jar) to field java.util.Properties.defaults
WARNING: Please consider reporting this to the maintainers of com.thoughtworks.xstream.core.util.Fields
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
[INFO] Packaging webapp
[INFO] Assembling webapp [webapp] in [/var/lib/jenkins/workspace/CICD_Pipeline_Demo/webapp/target/webapp]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/CICD_Pipeline_Demo/webapp/src/main/webapp]
[INFO] Webapp assembled in [43 msecs]
[INFO] Building war: /var/lib/jenkins/workspace/CICD_Pipeline_Demo/webapp/target/webapp.war
[INFO] WEB-INF/web.xml already added, skipping
[INFO] -----
[INFO] Reactor Summary for Maven Project 1.0-SNAPSHOT:
[INFO]
[INFO] Maven Project ..... SUCCESS [ 0.134 s]
[INFO] Server ..... SUCCESS [ 2.747 s]
[INFO] Webapp ..... SUCCESS [ 0.890 s]
[INFO] -----
[INFO] BUILD SUCCESS ✓
[INFO] -----
[INFO] Total time: 3.912 s
[INFO] Finished at: 2024-08-11T16:14:18+05:30
[INFO] -----
Archiving artifacts ✓
Finished: SUCCESS ✓
```

Continuous Deployment

Step 8) We need to install Apache Tomcat, and for this, you need to visit the [Tomcat Download](#) page. In the core section, hover over the ‘tar.gz’ link and copy it. Now, use the below commands in your server one by one.

- First, four commands will create one temporary directory and user group to access the file. Here, use the command curl -O ‘paste tomcat download link‘ as shown in the command below.
- Use further commands to create a tomcat directory and extract the gzip file. Just cross-check the version number of the Tomcat that you are downloading and extracting.
- Now the permission of the files needs to be configured with the below commands. In the last command, replace it with your username by which you are accessing the server.

sudo groupadd tomcat



A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ su root
Password:
root@Ubuntu:/home/sujata/Desktop/JENKINS_LAB# sudo groupadd tomcat
```

```
sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

```
root@Ubuntu:/home/sujata/Desktop/JENKINS_LAB# sudo useradd -s /bin/false -g tomcat -d /opt/tomcat tomcat
```

```
cd
```

```
root@Ubuntu:/home/sujata/Desktop/JENKINS_LAB# cd
```

```
cd /tmp
```

```
root@Ubuntu:~# cd /tmp
```

```
curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz
```

```
root@Ubuntu:/tmp# curl -O https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.93/bin/apache-tomcat-9.0.93.tar.gz
% Total    % Received % Xferd  Average Speed   Time     Time     Time  Current
          Dload  Upload Total   Spent    Left  Speed
100 11.5M  100 11.5M    0      0  16.0M      0 --:--:-- --:--:-- --:--:-- 16.0M
```

```
sudo mkdir /opt/tomcat
```

```
root@Ubuntu:/tmp# sudo mkdir /opt/tomcat
```

```
sudo tar xzvf apache-tomcat-9.0.54.tar.gz -C /opt/tomcat --strip-components=1
```

```
root@Ubuntu:/tmp# sudo tar xzvf apache-tomcat-9.0.93.tar.gz -C /opt/tomcat --strip-components=1
apache-tomcat-9.0.93/conf/
apache-tomcat-9.0.93/conf/catalina.policy
apache-tomcat-9.0.93/conf/catalina.properties
apache-tomcat-9.0.93/conf/context.xml
apache-tomcat-9.0.93/conf/jaspic-providers.xml
apache-tomcat-9.0.93/conf/jaspic-providers.xsd
apache-tomcat-9.0.93/conf/logging.properties
apache-tomcat-9.0.93/conf/server.xml
apache-tomcat-9.0.93/conf/tomcat-users.xml
apache-tomcat-9.0.93/conf/tomcat-users.xsd
```



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
apache-tomcat-9.0.93/bin/digest.sh
apache-tomcat-9.0.93/bin/makebase.sh
apache-tomcat-9.0.93/bin/setclasspath.sh
apache-tomcat-9.0.93/bin/shutdown.sh
apache-tomcat-9.0.93/bin/startup.sh
apache-tomcat-9.0.93/bin/tool-wrapper.sh
apache-tomcat-9.0.93/bin/version.sh
```

```
cd /opt/tomcat
```

```
sudo chgrp -R tomcat /opt/tomcat
```

```
sudo chmod -R g+r conf
```

```
sudo chmod g+x conf
```

```
root@Ubuntu:/tmp# cd /opt/tomcat
root@Ubuntu:/opt/tomcat# sudo chgrp -R tomcat /opt/tomcat
root@Ubuntu:/opt/tomcat# sudo chmod -R g+r conf
root@Ubuntu:/opt/tomcat# sudo chmod g+x conf
root@Ubuntu:/opt/tomcat#
```

```
cd ..
```

```
root@Ubuntu:/opt/tomcat# cd ..
```

```
sudo chown -R sujata:sujata tomcat/
```

```
root@Ubuntu:/opt# sudo chown -R sujata:sujata tomcat/
root@Ubuntu:/opt#
```

Step 9) We need to update the port number from 8080 to 8090 in the server.xml file.
We are updating it as this port number is already in use by Jenkins, and we have created 8090 in Azure VM for Tomcat. Use the below commands to edit the file. When you enter the file, click the INSERT button to edit. Now search for a similar code, as shown in the image below. Update the port



number to 8090. To save the file, press the *Esc* key, type :*wq* and click on *Enter* button.

cd

```
root@Ubuntu:/opt# cd
```

cd /opt/tomcat/conf

```
root@Ubuntu:~# cd /opt/tomcat/conf
```

vi server.xml

```
root@Ubuntu:/opt/tomcat/conf# vi server.xml
```

```
<Connector port="8090" protocol="HTTP/1.1"
connectionTimeout="20000"
redirectPort="8443"
maxParameterCount="1000"
/>
```

Step 10) Similarly, we need to edit the '*tomcat-users.xml*' file to update the roles that enable us to deploy files using Tomcat. In the file before *tomcat-users* ending code, paste the below roles code. To save the file press, the *Esc* key, type :*wq* and press *Enter*.

cd

```
root@Ubuntu:/opt/tomcat/conf# cd
```

cd /opt/tomcat/conf

```
root@Ubuntu:~# cd /opt/tomcat/conf
```

vi tomcat-users.xml

```
root@Ubuntu:/opt/tomcat/conf# vi tomcat-users.xml
```



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
<role rolename="manager-gui"/>  
  
<role rolename="manager-script"/>  
  
<role rolename="manager-jmx"/>  
  
<role rolename="manager-status"/>  
  
<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>  
  
<user username="deployer" password="deployer" roles="manager-script"/>  
  
<user username="tomcat" password="s3cret" roles="manager-gui"/>
```

```
<role rolename="manager-gui"/>  
<role rolename="manager-script"/>  
<role rolename="manager-jmx"/>  
<role rolename="manager-status"/>  
<user username="admin" password="admin" roles="manager-gui, manager-script, manager-jmx, manager-status"/>  
<user username="deployer" password="deployer" roles="manager-script"/>  
<user username="tomcat" password="s3cret" roles="manager-gui"/>  
  
/>
```

Step 11) We also need to update the context.xml file to remove the IP restriction. Use the same steps to edit the file with the below commands.

cd

```
root@Ubuntu:/opt/tomcat/conf# cd
```

```
cd /opt/tomcat/webapps/manager/META-INF
```

```
root@Ubuntu:~# cd /opt/tomcat/webapps/manager/META-INF
```

```
vi context.xml
```

```
root@Ubuntu:/opt/tomcat/webapps/manager/META-INF# vi context.xml
```



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



Remove all the content present inside ‘context‘ as shown in the image below and save it. :wq <enter key>

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
    Licensed to the Apache Software Foundation (ASF) under one or more
    contributor license agreements. See the NOTICE file distributed with
    this work for additional information regarding copyright ownership.
    The ASF licenses this file to You under the Apache License, Version 2.0
    (the "License"); you may not use this file except in compliance with
    the License. You may obtain a copy of the License at

        http://www.apache.org/licenses/LICENSE-2.0

    Unless required by applicable law or agreed to in writing, software
    distributed under the License is distributed on an "AS IS" BASIS,
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
    See the License for the specific language governing permissions and
    limitations under the License.
-->

<Context antiResourceLocking="false" privileged="true">
</Context>
```

Step 12) Now, all the files are edited successfully. To update the Tomcat, we need to restart the system to accept all our changes. Use the below commands in the server for shutdown and startup of Tomcat. With this step, Tomcat is ready to deploy our container.

cd

```
root@Ubuntu:/opt/tomcat/webapps/manager/META-INF# cd
```

cd /opt/tomcat/bin/

```
root@Ubuntu:~# cd /opt/tomcat/bin/
```



./shutdown.sh

```
root@Ubuntu:/opt/tomcat/bin# ./shutdown.sh
```

./startup.sh

```
root@Ubuntu:/opt/tomcat/bin# ./startup.sh
Using CATALINA_BASE:      /opt/tomcat
Using CATALINA_HOME:      /opt/tomcat
Using CATALINA_TMPDIR:    /opt/tomcat/temp
Using JRE_HOME:           /usr
Using CLASSPATH:          /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

Step 13) In the Jenkins Dashboard, click on Manage Jenkins and then visit Manage Plugins.

The screenshot shows the Jenkins dashboard with the 'Manage Jenkins' link highlighted by a red box. The page displays a message about a new Jenkins version available for download, a warning about security vulnerabilities in Jenkins 2.452.3, and sections for System Configuration and Plugins. The 'Plugins' section is also highlighted with a red box.

Step 14) Click on the Available tab and search for the ‘Deploy to Container’ plugin. Select the plugin and click on the ‘Install without restart’ button.

The screenshot shows the Jenkins 'Plugins' page under 'Available plugins'. A search bar at the top contains 'deploy to container'. Below it, a table lists the 'Deploy to container' plugin by 'Artifact Updaters'. A red box highlights the 'Install' button for this plugin. The 'Updates' tab is also highlighted with a red box.



Plugins

Updates 11

Available plugins

Installed plugins

Advanced settings

Download progress

Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

SSH server Success ✓

Deploy to container Success ✓

Loading plugin extensions Success ✓

→ Go back to the top page
(you can start using the installed plugins right away)

→ Restart Jenkins when installation is complete and no jobs are running

Step 15) Go back to your Configure window and select the Post-build Actions tab. Select the ‘Add post-build action’ drop-down button and select the ‘Deploy war/ear to a container’ plugin. Fill the same path of your war file here, as shown in the image below.



Add build step ▾

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Post-build Actions

Archive the artifacts ?

Files to archive ?

**/*.war



Advanced ▾

Deploy war/ear to a container

WAR/EAR files ?

**/*.war



Context path ?

Containers

Save

Apply

Step 16) Now click on the ‘Add Container‘ button and select the ‘Tomcat 9.x Remote‘ as we are using version 9 of the Tomcat. Fill in the URL of the same virtual machine with the new port number 8090.



Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Context path ?

Containers

Tomcat 9.x Remote ✓

Credentials

- none -

+ Add ▾

Tomcat URL ?

http://localhost:8090/ ✓

Advanced ▾

Add Container ▾

On the credentials drop-down button, select Jenkins.

Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Containers

Tomcat 9.x Remote

Credentials

- none -

+ Add ▾

Jenkins

http://localhost:8090/

Step 17) In this window, fill in the username and password that we have used in the 'tomcat-users.xml' file (in Step 10). Fill in the ID, description and click on the button



‘Add’.

Jenkins Credentials Provider: Jenkins

Global (Jenkins, nodes, items, all child items, etc) ▾

Username ?
admin ✓

Treat username as secret ?

Password ?
**** ✓

ID ?
tomcat ✓

Description ?
tomcat ✓

Cancel Add

Step 18) Click on the credentials drop-down button and select the recently created credential. Save all the settings and again click on the ‘Build Now’ button from the Jenkins dashboard. If the Build is successful, the war file will get deployed.



Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Containers

Tomcat 9.x Remote

Credentials

admin/******** (tomcat)

- none -

admin/******** (tomcat)

sujata/********

Tomcat 9.x

http://localhost:8090/

Advanced ▾

Add Container ▾

Deploy on failure

Add post-build action ▾

Save

Apply

BUILD GOT FAILED:

```
context null
ERROR: Build step failed with exception
org.codehaus.cargo.container.ContainerException: Failed to redeploy [/var/lib/jenkins/workspace/CICD_Pipeline_Demo/webapp.war]
    at
org.codehaus.cargo.container.tomcat.internal.AbstractTomcatManagerDeployer.redeploy(AbstractTomcatManagerDeployer.java:176)
    at hudson.plugins.deploy.CargoContainerAdapter.deploy(CargoContainerAdapter.java:81)
```

```
root@Ubuntu:/home/sujata# cd
root@Ubuntu:~# cd /opt/tomcat/webapps/manager/META-INF
root@Ubuntu:/opt/tomcat/webapps/manager/META-INF# vi context.xml
```



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
<?xml version="1.0" encoding="UTF-8"?>
<!--
Licensed to the Apache Software Foundation (ASF) under one or more
contributor license agreements. See the NOTICE file distributed with
this work for additional information regarding copyright ownership.
The ASF licenses this file to You under the Apache License, Version 2.0
(the "License"); you may not use this file except in compliance with
the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
-->

<CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
    sameSiteCookies="strict" />
<Valve className="org.apache.catalina.valves.RemoteAddrValve"
    allow="127\\.\\d+\\.\\d+\\.\\d+|::1|0:0:0:0:0:0:1" />
<Manager sessionAttributeValueClassNameFilter="java\\.lang\\.\\(?:Boolean|Integer|Long|Number|String)|org\\.a
pache\\.catalina\\.filters\\.CsrfPreventionFilter\\$LruCache\\(?:\\$1)?|java\\.util\\.\\(?:Linked)?HashMap"/>
</Context>
```

Replace with



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



```
<?xml version="1.0" encoding="UTF-8"?>
<!--
```

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to You under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

<http://www.apache.org/licenses/LICENSE-2.0>

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

```
-->
```

```
<Context antiResourceLocking="false" privileged="true">

</Context>
```

```
root@Ubuntu:/opt/tomcat/webapps/manager/META-INF# cd ../../bin/
root@Ubuntu:/opt/tomcat/bin# ./shutdown.sh
```

```
root@Ubuntu:/opt/tomcat/bin# ./shutdown.sh
Using CATALINA_BASE:      /opt/tomcat
Using CATALINA_HOME:      /opt/tomcat
Using CATALINA_TMPDIR:    /opt/tomcat/temp
Using JRE_HOME:           /usr
Using CLASSPATH:          /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli.jar
Using CATALINA_OPTS:
NOTE: Picked up JDK_JAVA_OPTIONS: --add-opens=java.base/java.lang=ALL-UNNAMED --add-opens=java.base/java.io=ALL-UNNAMED --add-opens=java.base/java.util=ALL-UNNAMED --add-opens=java.base/java.util.concurrent=ALL-UNNAMED --add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED
```



```
root@Ubuntu:/opt/tomcat/bin# ./startup.sh
Using CATALINA_BASE:      /opt/tomcat
Using CATALINA_HOME:       /opt/tomcat
Using CATALINA_TMPDIR:    /opt/tomcat/temp
Using JRE_HOME:           /usr
Using CLASSPATH:          /opt/tomcat/bin/bootstrap.jar:/opt/tomcat/bin/tomcat-juli
.jar
Using CATALINA_OPTS:
Tomcat started.
```

The screenshot shows the Apache Tomcat 9.0.93 welcome page. At the top, there's a navigation bar with links to Home, Documentation, Configuration, Examples, Wiki, and Mailing Lists. On the right, there are links for Find Help, Server Status, Manager App, and Host Manager. The main content area has a green header bar that says "If you're seeing this, you've successfully installed Tomcat. Congratulations!" Below this, there's a section titled "Developer Quick Start" with links to Tomcat Setup, First Web Application, Realms & AAA, JDBC DataSources, Examples, and Servlet Specifications/Tomcat Versions. The "Documentation" section includes links to Tomcat 9.0 Documentation, Configuration, and Wiki, along with information about the RUNNING.txt file. The "Getting Help" section lists mailing lists: tomcat-announce (announcements), tomcat-users (user support), taglibs-user (support for Taglibs), and tomcat-dev (development messages). There are also links to Release Notes, Changelog, Migration Guide, and Security Notices.

Conclusion: In this experiment we implement and configuring Jenkins pipelines to deploy tomcat server.