EXPERIMENT NO. 5

TO BUILD THE PIPELINE OF JOBS USING MAVEN / GRADLE / ANT IN JENKINS, CREATE A PIPELINE SCRIPT TO TEST AND DEPLOY AN APPLICATION OVER THE TOMCAT SERVER

Theory:

1. Introduction to Jenkins and CI/CD

Jenkins is an open-source automation server used for continuous integration and continuous deployment (CI/CD). It helps automate the software development lifecycle by building, testing, and deploying applications.

1.1 CI/CD Concepts

- Continuous Integration (CI): Automatically integrates code changes into a shared repository and runs tests.
- Continuous Deployment (CD): Automates the process of deploying applications to production or staging environments.

1.2 Tools Used

- Jenkins Automation server.
- Maven/Gradle/Ant Build automation tools.
- Tomcat A web server for deploying Java applications.

2. Installing and Configuring Jenkins

- Download and Install Jenkins:
 - o Download from Jenkins official site. o Install and start the Jenkins service.
 - o Access Jenkins at http://localhost:8080/.
- ➤ Install Required Plugins:
 - o Go to Manage Jenkins > Plugin Manager.
 - o Install Pipeline, Maven Integration, and Deploy to Container plugins.

3. Creating a Jenkins Pipeline

Jenkins pipelines define a series of automated steps for building, testing, and deploying applications.

- o Steps to Create a Pipeline
- 1. Open Jenkins Dashboard and click on New Item.
- 2. Select Pipeline and provide a project name.
- 3. Click OK and navigate to the Pipeline section.
- 4. Write a Pipeline script (Declarative or Scripted) to define the build and deployment process.

T23 - 134

4. Writing a Jenkins Pipeline Script

```
The following script builds a Java application using Maven and deploys it to Tomcat:
groovy CopyEdit pipeline {
                              agent any
       stage('Checkout Code') {
                                        steps {
git 'https://github.com/your-repository.git'
     }
     stage('Build with Maven') {
steps {
         sh 'mvn clean package'
       }
     }
    stage('Deploy to Tomcat') {
steps {
                    adapters:
                                [tomcat8(credentialsId:
                                                          'tomcat-cred',
          deploy
                                                                           path:
                                                                                        url:
'http://yourtomcat-server:8080')],
          war: '**/*.war'
       }
     }
  }
}
```

- 5. Configuring Jenkins for Deployment
 - 1. Configure Tomcat Server:
 - o Install Tomcat and start the server.
 - o Set up a user with deployment privileges in conf/tomcat-users.xml: xml CopyEdit

```
<role rolename="manager-gui"/>
<role rolename="manager-script"/>
<user username="admin" password="admin" roles="manager-gui,manager-script"/>
```

- o Restart Tomcat.
- 2. Set Up Jenkins Credentials:
 - o Go to Manage Jenkins > Credentials. o Add a username/password credential for Tomcat deployment.
- 3. Run the Pipeline in Jenkins:
 - o Click Build Now to execute the pipeline. o Verify the deployment at http://your-tomcat-server:8080/your-app.

Sanya Ramchandani T23 – 134

Example 1:

Creating a job:

Start building your software project

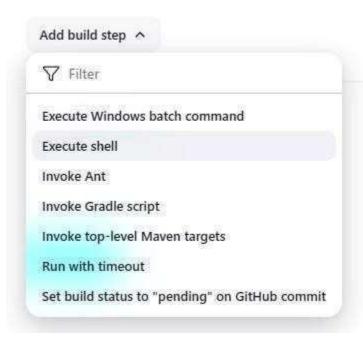


Naming the job and setting it as freestyle:

| xamp | le1 |
|--------|---|
| Requir | ed field |
| 0 | Freestyle project |
| | Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications. |
| | Pipeline |
| | Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type. |
| | Multi-configuration project |
| | Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc. |
| | Folder |
| | Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates separate namespace, so you can have multiple things of the same name as long as they are in different folders. |
| 2 | Multibranch Pipeline |
| To | Creates a set of Pipeline projects according to detected branches in one SCM repository. |
| | Organization Folder |
| 需是 | Creates a set of multibranch project subfolders by scanning for repositories. |

Selecting build type as "Execute shell":

Build Steps



Entering a simple command for the shell execution: Build Steps

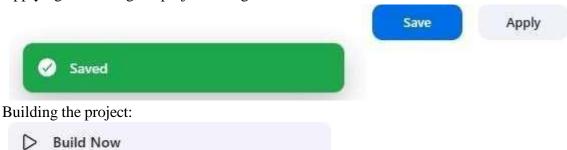
Execute shell ?

Command

See the list of available environment variables

echo "Hello TSEC"

Applying and saving the project configuration:

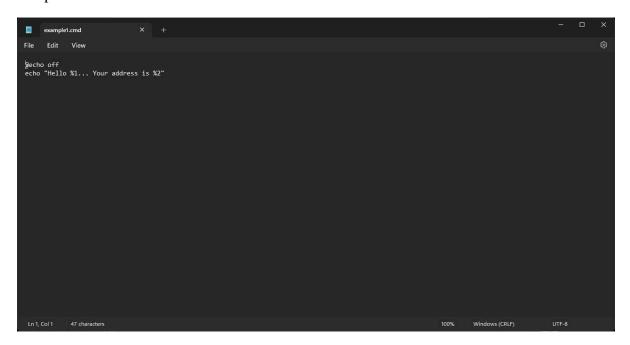


Console output (after building):



Example 1.2: Taking parameters through files Contents of script

example1.cmd:



Executing script example 1.cmd on the terminal:

```
Microsoft Windows [Version 10.0.22621.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\AIGDS 202>Microsoft Windows [Version 10.0.22631.3155) (c) Microsoft Corporation. All rights reserved.
'Microsoft' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\AIGDS 202>C:\Admin\Academics\TSEC\Start3\SEPM>example1.cmd
The system cannot find the path specified.

C:\Users\AIGDS 202>"Hello... Your address is "
'"Hello... Your address is "' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\AIGDS 202>C:\Admin\Academics\TSEC\Start3\SEPM>example1.cad Tanishq
The system cannot find the path specified.

C:\Users\AIGDS 202>"Hello Tanihsq... Your address is "
'"Hello Tanihsq... Your address is "' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\AIGDS 202>C:\Admin\Academics\TSEC\Start3\SEPM>example1.cmd Tanishq Girgaon "Helle Tanishq... Your address is Gi
rgaon"
The system cannot find the path specified.
```

Modifying the Jenkins project to execute the script while supplying required parameters:

| ■ Execute Windows batch command ? | × |
|--|---|
| Command | |
| See the list of available environment variables | |
| C:\Admin\Academics\TSEC\Start3\SEPM\example1.cmd Siddhant Goregaon | |
| | |
| Advanced ✓ | |
| Add build step ∨ | |

Console output after building the modified project:

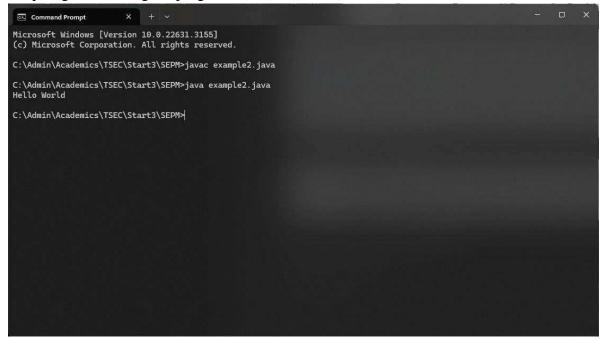
| F Status | ○ Console Output |
|--------------------------------|--|
| Changes | |
| Console Output | Started by user Siddhant Chellur Running as SYSTEH |
| View as plain text | Bullding in workspace C:IProgramBata\Penkins\-jenkins\workspace\tample\! [Example] \$ cmd /c call C:\ull00x857699\jenkins\075995818165161158.bet |
| Edit Build Information | C:\ProgramDuta\Jenkinsjenkins\morkspace\Examplel\C:\Idmin\Academic\YSEC\Start3\SEPM\examplel.cnd Siddhant Goregaon "Nello Siddhant Your address is Goregaon" |
| Delete build '#4' | "Hello stadmant Your address is Goregaon" Finished: SUCCESS |
| A COLUMN TO THE REAL PROPERTY. | |

Example 2 Example 2.1: Running a Java program under Jenkins

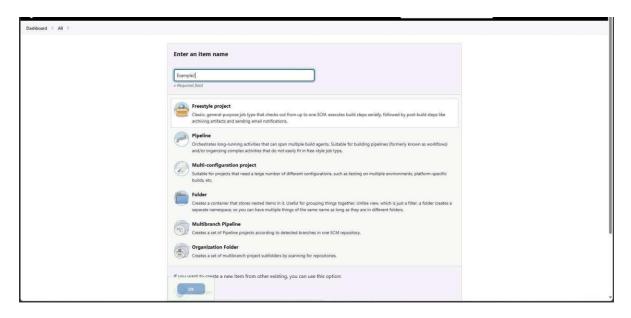
Creating a simple Java program:



Compiling and running the program on the terminal:



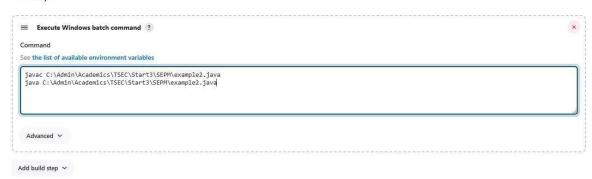
Creating a new freestyle project:



Configure new project:

T23 - 134

Build Steps



Console output after building:

⊘ Console Output

```
Started by user Siddhant Chetlur
Running as SYSTEM
[Envinject] - Loading node_environment variables.
Building in owckspace (:\ProgramOata\)ankins\,jenkins\workspace\Example2
[Example2] $ cmd /c call C:\WINDOWS\TEMP\jenkins15296462484398614135.bat

C:\ProgramData\]ankins\,jenkins\workspace\Example2>java C:\Admin\Academics\TSEC\Start3\SEPH\example2.java

C:\ProgramData\]ankins\,jenkins\workspace\Example2>java C:\Admin\Academics\TSEC\Start3\SEPH\example2.java

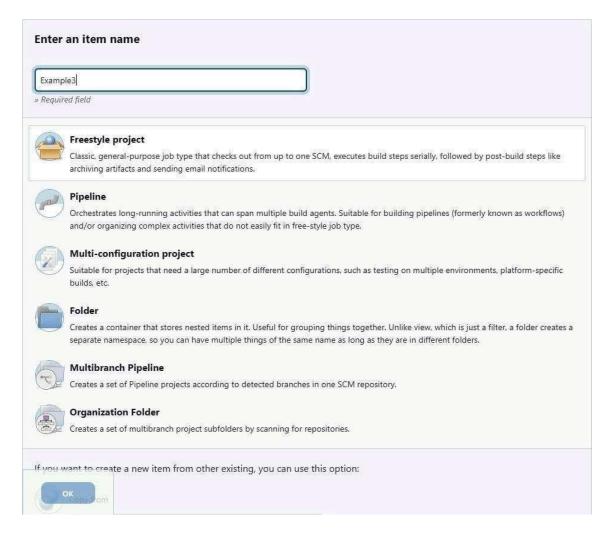
Hello World

C:\ProgramData\]ankins\,jenkins\workspace\Example2>exit 0

Finished: SUCCESS
```

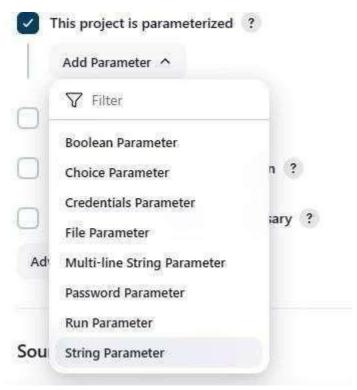
Example 3 Example 3.1: Parameterise build

Creating a new freestyle project:



Enabling parameterisation and adding a String parameter:

T23 - 134



Configuring the string parameter as Fname:



Adding a choice parameter and configuring it as City with the following choices:

T23 - 134

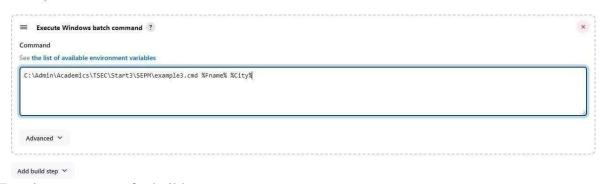


Creating a script which takes 2 arguments for name and city:

```
C:\Users\AI&DS 202>Microsoft Windows [Version 10.0.22631.3155] (c) Microsoft Corporation. All rights reserved.
'Microsoft' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\AI&DS 202>C:\Admin\Academics\TSEC\Start3\SEPH>example3.cnd
The system cannot find the path specified.
:\Users\AI&DS 202>Hello your name is and your city is
'Hello' is not recognized as an internal or external command,
operable program or batch file.
The system cannot find the path specified.
C:\Users\AI&DS 202>Hello your name is Tanishq and your city is
'Hello' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\AI&DS 202>C:\Admin\Academics\TSEC\Start3\SEPM>example3.cmd Tansishq Bandra
The system cannot find the path specified.
 :\Users\AI&DS 202>Hello your name is Tanishq and your city is Bandra
'Hello' is not recognized as an internal or external command, operable program or batch file.
```

Configuring build steps:

Build Steps



Entering parameters for build:

T23 - 134

Project Example3

| This build requires parameters: | |
|---------------------------------|---|
| Fname | |
| Siddhant | |
| City | |
| Bandra | v |
| ▷ Build Cancel | |

Console output after building:

✓ Console Output

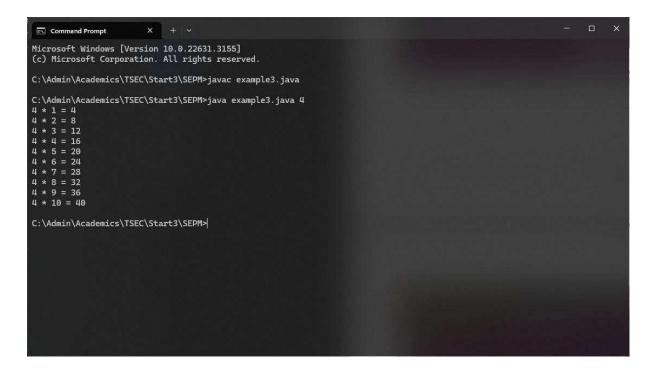
```
Started by user Siddhant Chetlur
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace C:\ProgramData\zenkins\.jenkins\workspace\Example3
[Example3] $ cmd /c call C:\MINDOWS\TEMP\jenkins\doo436165150986151.bat

C:\ProgramData\zenkins\.jenkins\workspace\Example3>C:\Admin\Academics\TSEC\Start3\SEPH\example3.cmd Siddhant Bandra
Hello your name is Siddhant and your city is Bandra
Finished: SUCCESS
```

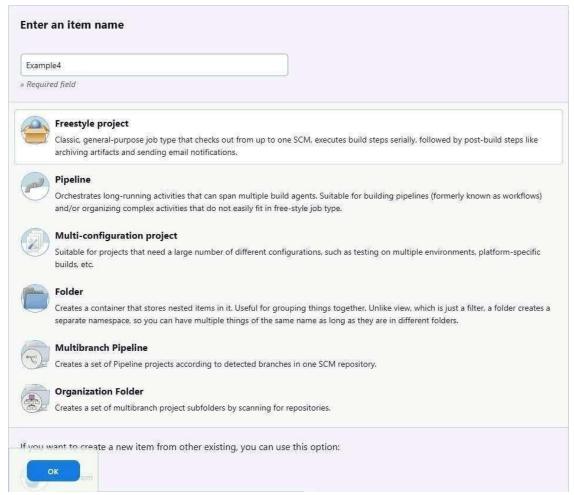
Example 3.2: Running a Java program with parameters Creating

a Java program with an input argument:

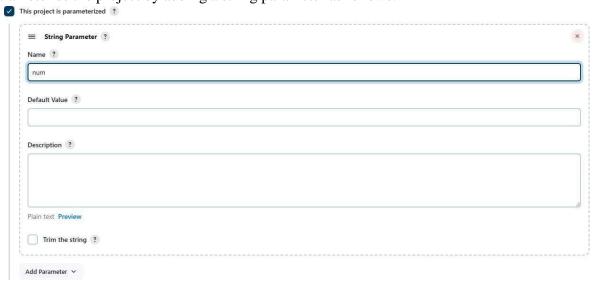
Testing the program on the terminal:



Creating a new freestyle project:



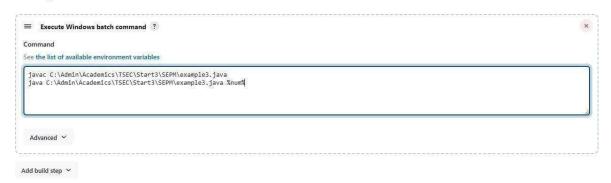
Parameterise the project by adding a string parameter as follows:



Configure the build steps:

T23 - 134

Build Steps



Entering the parameter for the build:

Project Example4



Console output after building:

Console Output

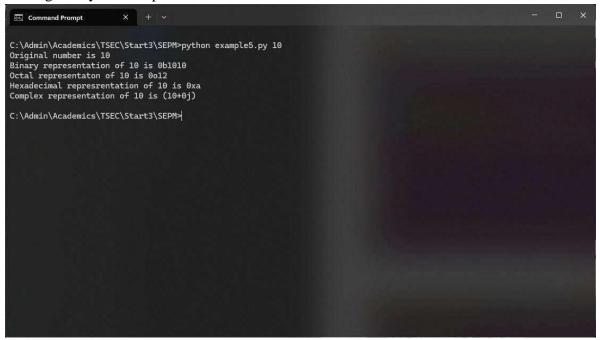


Example 5 Example

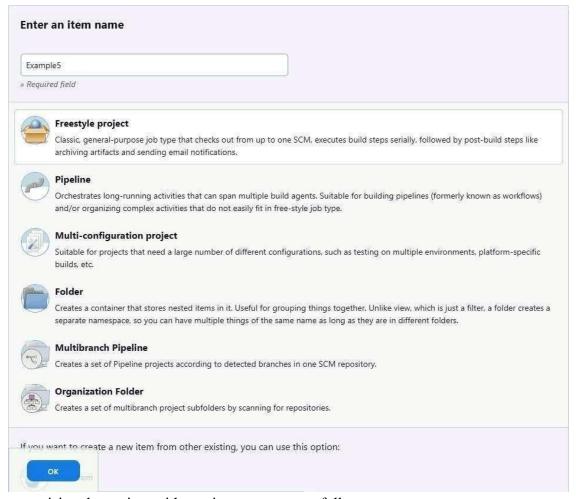
5.1: Running a Python program Creating a simple Python script:



Running the Python script on the terminal:



Creating a new freestyle project:



Parameterising the project with a string parameter as follows:



Configuring the build steps:



Setting the parameter for the build:

Project Example5



Console output after building:

✓ Console Output

```
Started by user Siddhant Chetlur

Running as SYSTEM

[EnvInject] - Loading node environment variables.

Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\Example5

[Example5] $ cmd /c call C:\WINDOWs\TEMP\jenkins11157306491994478222.bat

C:\ProgramData\Jenkins\.jenkins\workspace\Example5>python C:\Admin\Academics\TSEC\Start3\SEPM\example5.py 10

Original number is 10

Binary representation of 10 is 0b1010

Octal representation of 10 is 0b101

Octal representation of 10 is 0b10

Hexadecimal representation of 10 is (1040)

C:\ProgramData\Jenkins\workspace\Example5>exit 0

Finished: SUCCESS.
```

Conclusion

This experiment demonstrated how to automate a software build and deployment process using Jenkins. By integrating Maven, Gradle, or Ant, we streamlined the compilation and testing of applications, while Jenkins facilitated continuous deployment to a Tomcat server.