<u>Aim</u>: 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:

Continuous Integration (CI) is a software development practice where developers frequently integrate their code changes into a shared repository—often multiple times per day. Each integration is verified through an automated build and test process to detect errors as early as possible.

Jenkins is an open-source automation server used to implement CI/CD pipelines. It supports building, testing, and deploying applications through customizable jobs. Jenkins can execute scripts, compile code, run tests, and more, using plugins and freestyle or pipeline projects.

Key Features of Jenkins:

- Open-source and extensible
- Support for various plugins (Maven, Git, Docker, etc.)
- Freestyle and pipeline job support
- ❖ Easy integration with Git, GitHub, Bitbucket
- Parameterised builds and scheduling
- ❖ Build monitoring through Console Output and Blue Ocean UI

Demonstration of Jenkins Job Execution (Theoretical Steps):

Example 1.1: Deploying a Freestyle App in Jenkins

- 1. Login to Jenkins Dashboard.
- 2. Create a new item \rightarrow Name it \rightarrow Select Freestyle project.
- 3. Go to **Build Section** → Select **Execute Shell**.
- 4. Enter shell command, e.g.:

echo "Hello, Jenkins!"

- 5. Click **Apply** and **Save**.
- 6. Click Build Now.
- 7. View **Console Output** to verify execution.

Example 2.1: Running a Script with Parameters

1. Create a shell script (e.g., example1.cmd):

echo "Welcome, \$1!"

- 2. Test on terminal:
 - sh example1.cmd John
- 3. Modify Jenkins job to run the script with parameters:

sh example1.cmd John

4. Build and check console output for the message.

Running a Java Program under Jenkins

1. Write a simple Java program:

```
public class Hello {
   public static void main(String[] args) {
      System.out.println("Hello from Java!");
   }
}
```

2. Compile and run in terminal:

```
javac Hello.java
java Hello
```

- 3. Create a Jenkins freestyle project → Execute Shell:
 - javac Hello.java java Hello
- 4. Build and check the output.

Example 3.1: Parameterised Build

- 1. Create a new freestyle project.
- 2. Enable "This project is parameterized".
- 3. Add a String parameter (e.g., Fname).
- 4. Add a Choice parameter (e.g., City) with values: Mumbai, Pune, Delhi.
- 5. In Build → Execute Shell:

```
echo "Hello $Fname from $City!"
```

- 6. Click Build with Parameters and enter values.
- 7. Output displays personalized greeting.

Example 4.1: Running a Python Program

1. Write a Python script (greet.py):

```
import sys
print(f"Hello, {sys.argv[1]}!")
```

2. Run in terminal:

python3 greet.py Alice

- 3. Create Jenkins freestyle project.
- 4. Enable parameterization → Add String parameter: Name
- 5. Configure build step:

python3 greet.py \$Name

- 6. Build with parameter Name = Alice.
- 7. Output: Hello, Alice!

Use Case Examples:

- Automating builds and deployments
- Testing code on every commit
- Running scripts or data processing pipelines
- Hosting Java or Python apps using Tomcat
- Scheduled jobs for cleanup, backup, etc.

Implementation:

Programming in Jenkins:

Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible." In simple way, Continuous integration (CI) is the practice of frequently building and testing each change done to your code automatically.

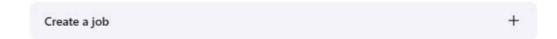
Jenkins is a self-contained, open-source automation server which can be used to automate all sorts of tasks related to building, testing, and delivering or deploying software.

Our first job will execute the shell commands. The freestyle project provides enough options and features to build the complex jobs that you will need in your projects.

Example 1

Example 1.1: Deploying a freestyle app in Jenkins Creating a job:

Start building your software project

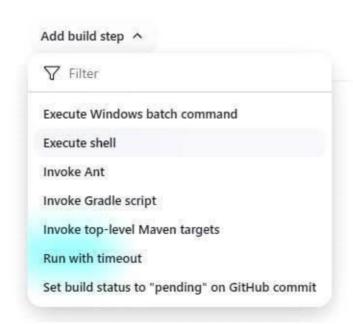


Naming the job and setting it as freestyle:



Selecting build type as "Execute shell":

Build Steps



Entering a simple command for the shell execution:



Applying and saving the project configuration:



Building the project:

Build Now

Console output (after building):



Save

Apply



Example 1.2: Taking parameters through files Contents of script

example1.cmd:

Executing script example1.cmd on the terminal:

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

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\SEPM>example1.cmd
The system cannot find the path specified.

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

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

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



Console output after building the modified project:

TSEC

Batch:-T12

Name & Roll No:- Sarthak Hinge - 35



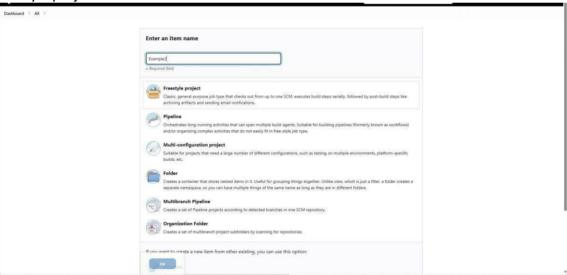
: Running a Java program under Jenkins

Creating a simple Java program:

Compiling and running the program on the terminal:

```
C:\Users\richminds\Desktop\sepm>javac 24.java
C:\Users\richminds\Desktop\sepm>java 24.java
This is T12
C:\Users\richminds\Desktop\sepm>
```

Creating a new freestyle project:



Configure new project:

Command

See the list of available environment variables

javac C:\Users\richminds\Desktop\sepm\24.java
java C:\Users\richminds\Desktop\sepm\24.java

Console output after building:

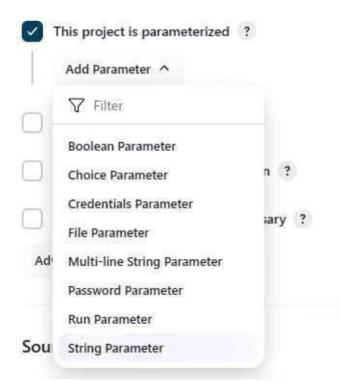


Example 3

Example 3.1: Parameterise build Creating a new

freestyle project:

Enabling parameterisation and adding a String parameter:



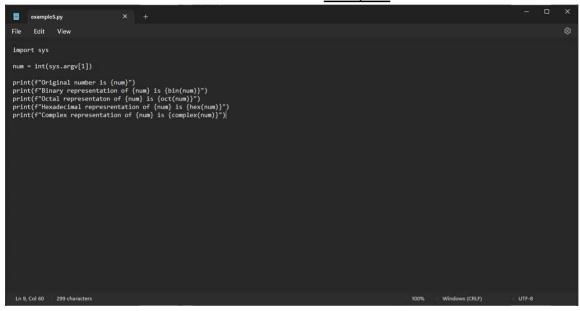
Configuring the string parameter as Fname:

Software and Project Management Lab Experiment No: - 05 Aim: To Build the pipeline of jobs using Maven / Gradle / Ant in Jenkins ■ String Parameter ? Name ? Fname Description ? Plain text Preview Trim the string ? Adding a choice parameter and configuring it as City with the following choices: Choice Parameter Name City Choices Ambernath Badlapur Kalyan Dombivli Requires Choices. Description Configuring build steps: Build Steps **≡** Execute Windows batch command ? See the list of available environment variables C:\Admin\Academics\TSEC\Start3\SEPM\example3.cmd %Fname% %City% Add build step Y Entering parameters for build: **Project Example3** This build requires parameters: Fname Siddhant Bandra

Console output after building:



Example 5



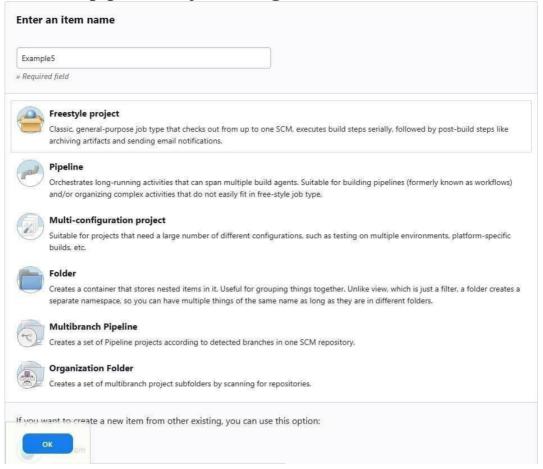
Example 5.1: Running a Python program Creating a simple

Python script:

Running the Python script on the terminal:

```
::\Users\richminds\Desktop\sepm>python 24.py 10
lumber is 10
::\Users\richminds\Desktop\sepm>_
```

Creating a new freestyle project:



Parameterising the project with a string parameter as follows:



Configuring the build steps:

Command

See the list of available environment variables

python C:\Users\richminds\Desktop\sepm\24.py

Setting the parameter for the build:



Conclusion: We have successfully built the pipeline of jobs using Maven / Gradle / Ant in Jenkins, created a pipeline script to Test and deploy an application over the tomcat server.

LO Mapping: LO is mapped