

Experiment No: - 05

Aim: 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:

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


Create a job





Naming the job and setting it as freestyle:


Enter an item name


» Required field


**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 a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.


**Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

OK

Selecting build type as “Execute shell”:

Build Steps

Add build step ^

 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

Entering a simple command for the shell execution:

Build Steps

≡ Execute shell ?

Command

See [the list of available environment variables](#)

```
echo "Hello TSEC"
```

Advanced ▾

Applying and saving the project configuration:

✓ Saved

Save

Apply

Building the project:

▶ Build Now

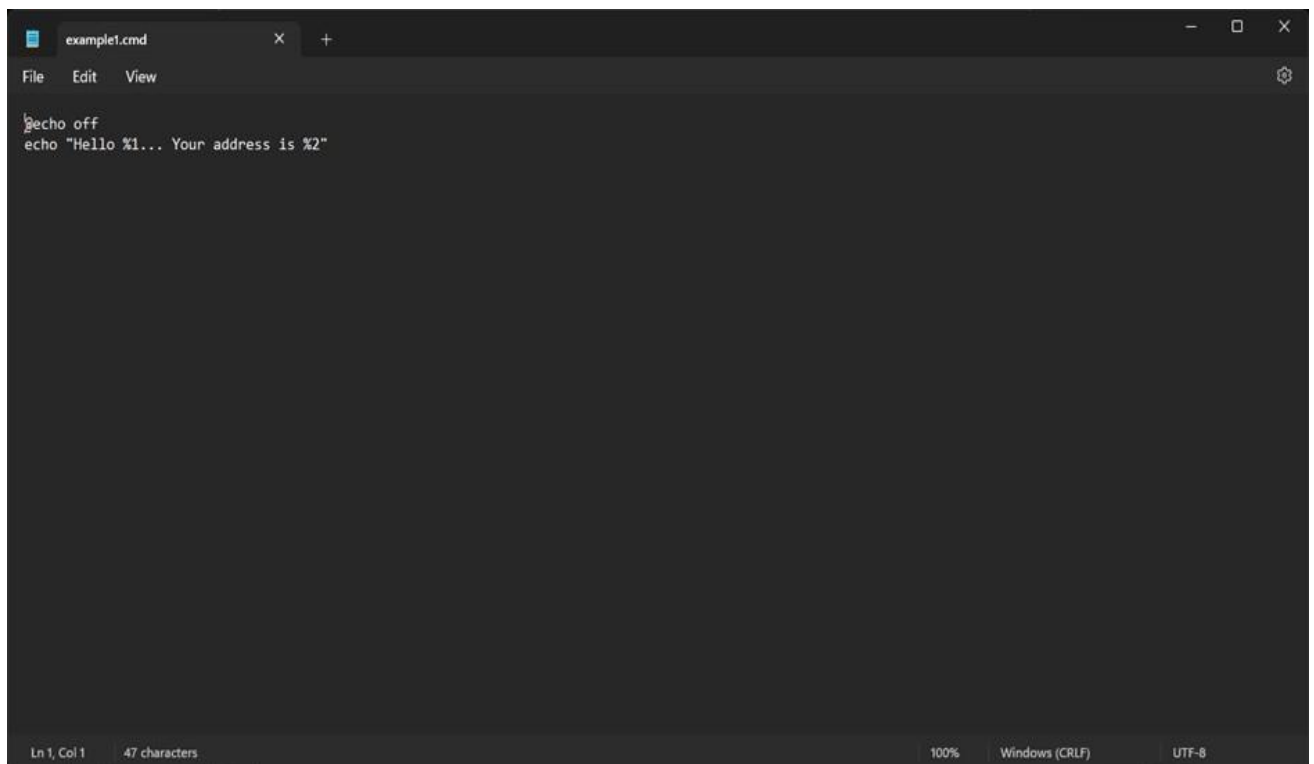
Vedanshi Shethia
2311137

Console output (after building):



Example 1.2: Taking parameters through files

Contents of script example1.cmd:

A screenshot of a Windows command prompt window titled 'example1.cmd'. The window has a menu bar with 'File', 'Edit', and 'View'. The command prompt shows the following text: 'set echo off' and 'echo "Hello %1... Your address is %2"'. The status bar at the bottom indicates 'Ln 1, Col 1', '47 characters', '100%', 'Windows (CRLF)', and 'UTF-8'.

Vedanshi Shethia
2311137

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 Gi
rgaon"
The system cannot find the path specified.
```

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

Build Steps



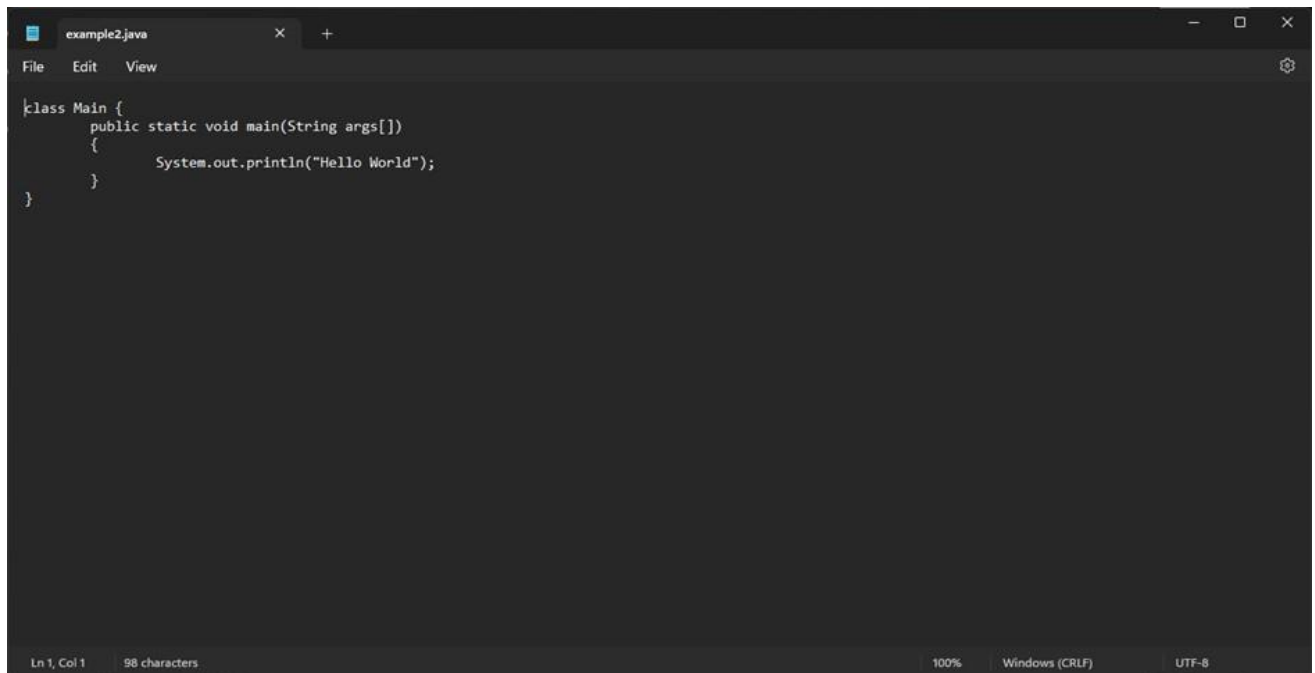
Console output after building the modified project:



Example 2

Example 2.1: Running a Java program under Jenkins

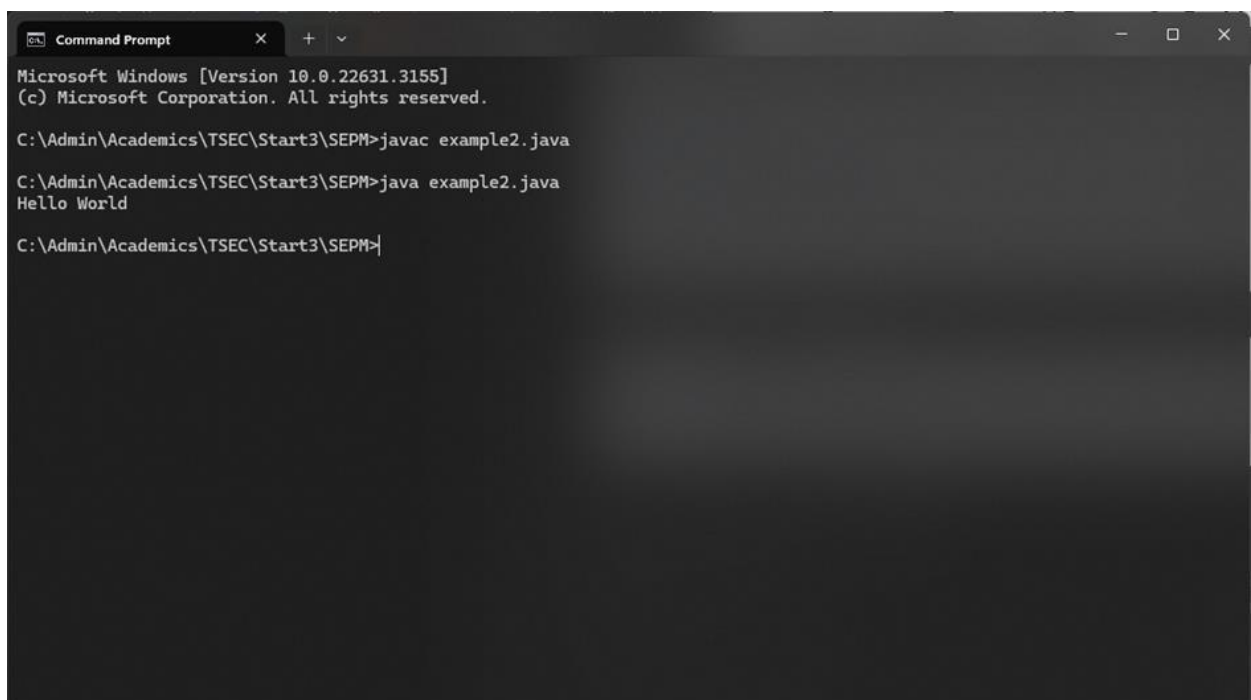
Creating a simple Java program:

A screenshot of an IDE window titled 'example2.java'. The window contains the following Java code:

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

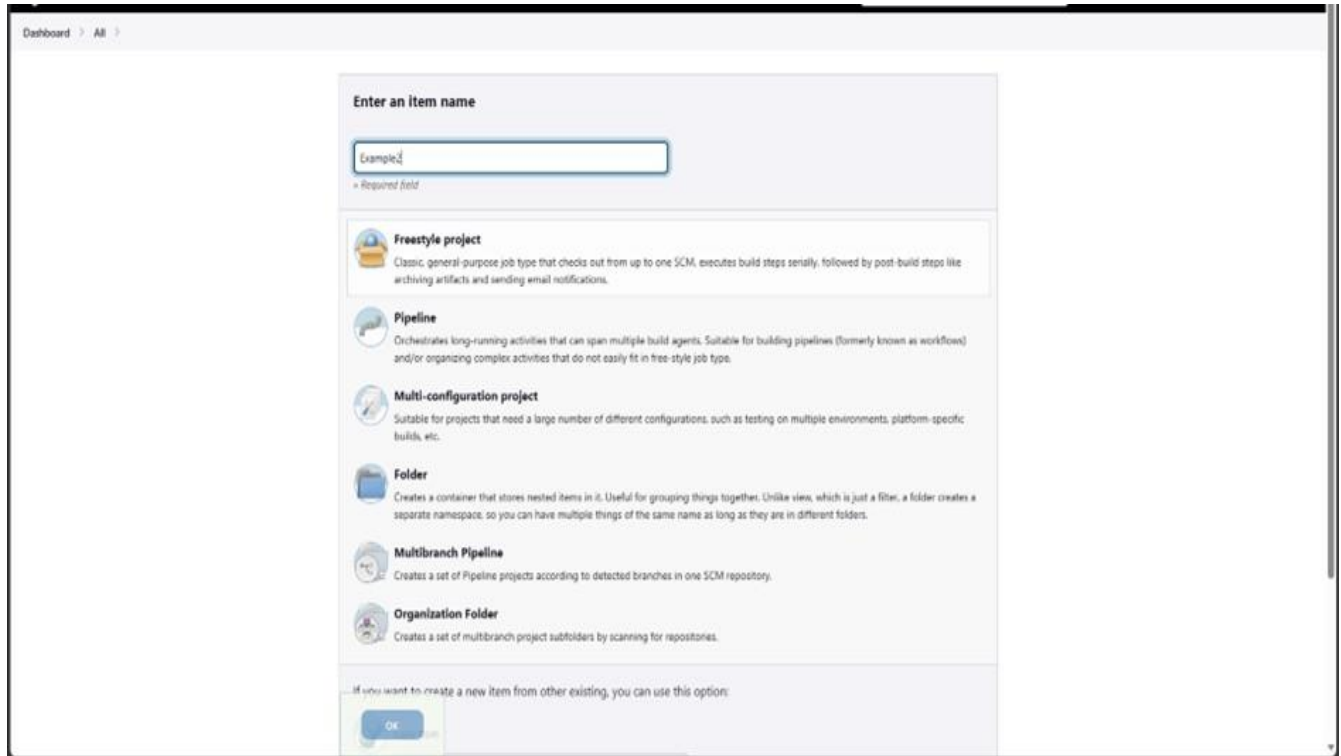
The status bar at the bottom indicates 'Ln 1, Col 1', '98 characters', '100%', 'Windows (CRLF)', and 'UTF-8'.

Compiling and running the program on the terminal:

A screenshot of a Windows Command Prompt window. The text displayed is as follows:

```
Microsoft Windows [Version 10.0.22631.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Admin\Academics\TSEC\Start3\SEPM>javac example2.java  
  
C:\Admin\Academics\TSEC\Start3\SEPM>java example2.java  
Hello World  
  
C:\Admin\Academics\TSEC\Start3\SEPM>|
```

Creating a new freestyle project:



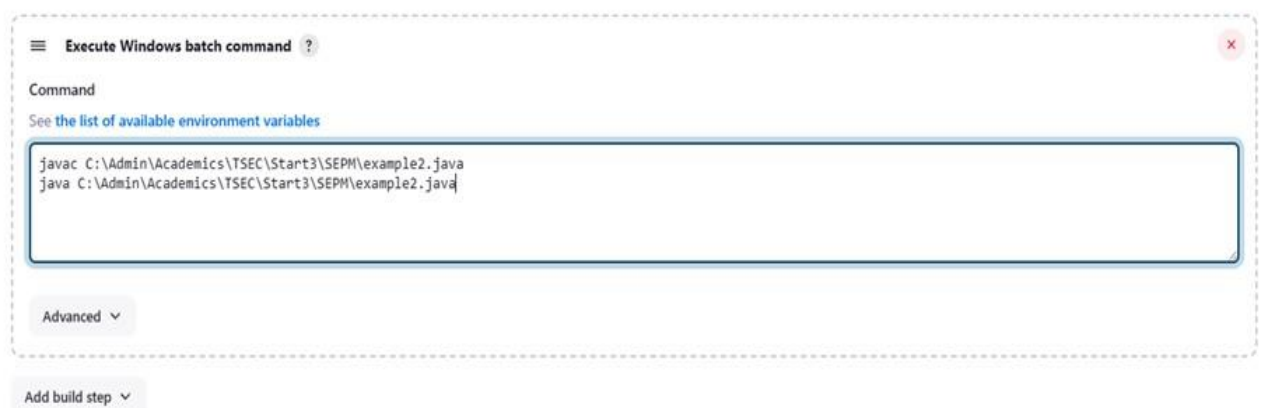
The screenshot shows the Jenkins 'Create new item' dialog box. At the top, there is a breadcrumb 'Dashboard > All' and a section titled 'Enter an item name' with a text input field containing 'Example2' and a red asterisk indicating it is a required field. Below this, there is a list of project types, each with an icon and a description:

- 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 a separate namespace, so you can have multiple things of the same name as long as they are in different folders.
- Multibranch Pipeline**: Creates a set of Pipeline projects according to detected branches in one SCM repository.
- Organization Folder**: Creates a set of multibranch project subfolders by scanning for repositories.

At the bottom, there is a note: 'If you want to create a new item from other existing, you can use this option:' followed by an 'OK' button.

Configure new project:

Build Steps



The screenshot shows the configuration for a 'Execute Windows batch command' build step. The title bar includes a hamburger menu, the text 'Execute Windows batch command', and a help icon. A red close button is in the top right corner. Below the title, the label 'Command' is followed by a link 'See the list of available environment variables'. A large text area contains the following commands:

```
javac C:\Admin\Academics\TSEC\Start3\SEPM\example2.java
java C:\Admin\Academics\TSEC\Start3\SEPM\example2.java
```

Below the text area is an 'Advanced' dropdown menu. At the bottom of the configuration box is an 'Add build step' button.

Vedanshi Shethia
2311137

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\Example2
[Example2] $ cmd /c call C:\WINDOWS\TEMP\jenkins15296462484398614135.bat

C:\ProgramData\Jenkins\jenkins\workspace\Example2>javac C:\Admin\Academics\TSEC\Start3\SEPH\example2.java

C:\ProgramData\Jenkins\jenkins\workspace\Example2>java C:\Admin\Academics\TSEC\Start3\SEPH\example2.java
Hello World

C:\ProgramData\Jenkins\jenkins\workspace\Example2>exit 0
Finished: SUCCESS
```


Example 3


Example 3.1: Parameterise build


Creating a new freestyle project:


Enter an item name


» Required field


**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.

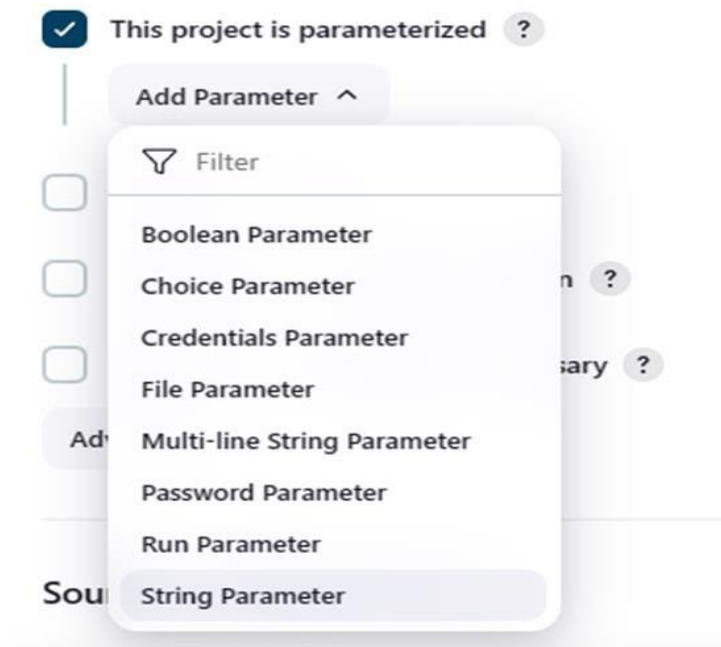
**Folder**
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

Enabling parameterisation and adding a String parameter:



Configuring the string parameter as Fname:

The image shows the configuration dialog for a 'String Parameter'. The dialog has a title bar with a hamburger menu icon, the text 'String Parameter', and a close button (X). The main area contains the following fields and options:

- Name:** A text input field containing the value 'Fname'.
- Default Value:** An empty text input field.
- Description:** A large, empty text area.
- Plain text:** A label followed by a 'Preview' link.
- Trim the string:** A checkbox that is currently unchecked, followed by a help icon (?).

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

The screenshot shows a 'Choice Parameter' configuration window. It has a title bar with a menu icon, the text 'Choice Parameter', and a help icon. Below the title bar, there are three sections: 'Name' with a text input field containing 'City'; 'Choices' with a list box containing 'Bandra', 'Kalyan', 'Dombivali', 'Churchgate', 'Thane', and 'Dadar'; and 'Description' with a large text area. At the bottom left, there are links for 'Plain text' and 'Preview'. A red close button is in the top right corner.

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.

C:\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.

C:\Users\AI&DS 202>C:\Admin\Academics\TSEC\Start3\SEPH example3.cmd Tanishq
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\SEPH>example3.cmd Tansishq Bandra
The system cannot find the path specified.

C:\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.

C:\Users\AI&DS 202>C:\Admin\Academics\TSEC\Start3\SEPH
```

Vedanshi Shethia
2311137

Configuring build steps:

Build Steps

Execute Windows batch command ?

Command

See [the list of available environment variables](#)

```
C:\Admin\Academics\TSEC\Start3\SEPM\example3.cmd %Fname% %City%
```

Advanced ▾

Add build step ▾

Entering parameters for build:

Project Example3

This build requires parameters:

Fname

Aditya

City

Bandra ▾

▶ Build

Cancel

Console output after building:

✓ Console Output

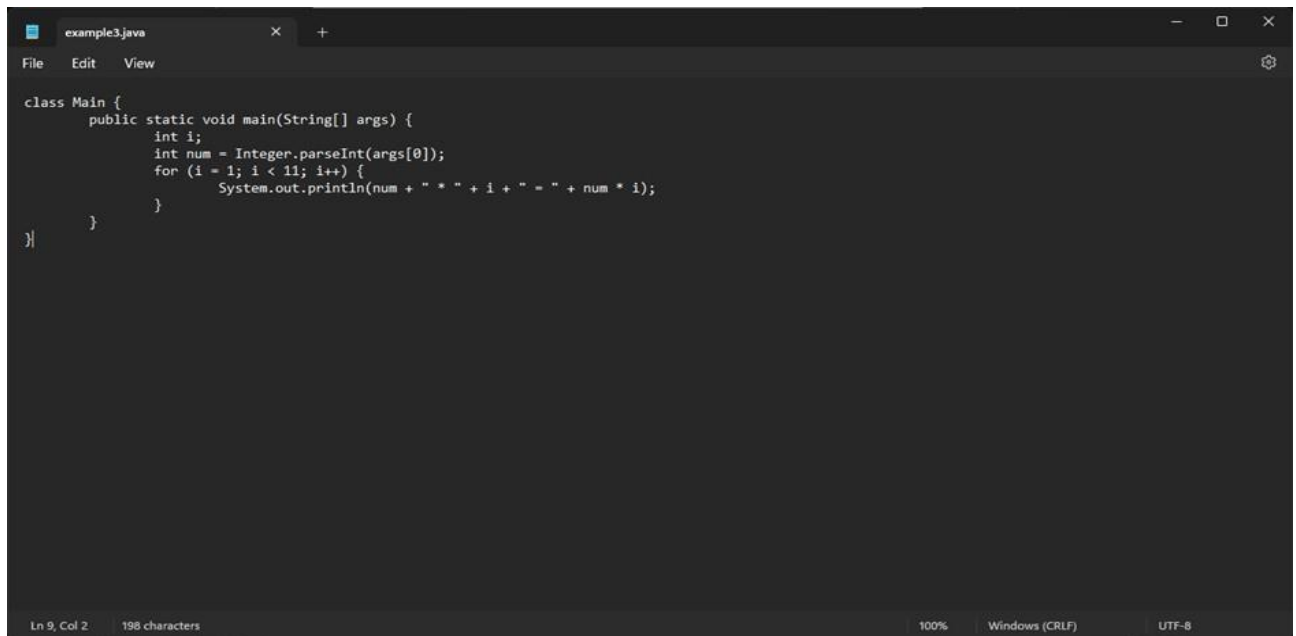
```
Started by user Siddhant Chetlur
Running as SYSTEM
[EnvInject] - Loading node environment variables.
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Example3
[Example3] $ cmd /c call C:\WINDOWS\TEMP\jenkins14094536165150986151.bat

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

Vedanshi Shethia
2311137

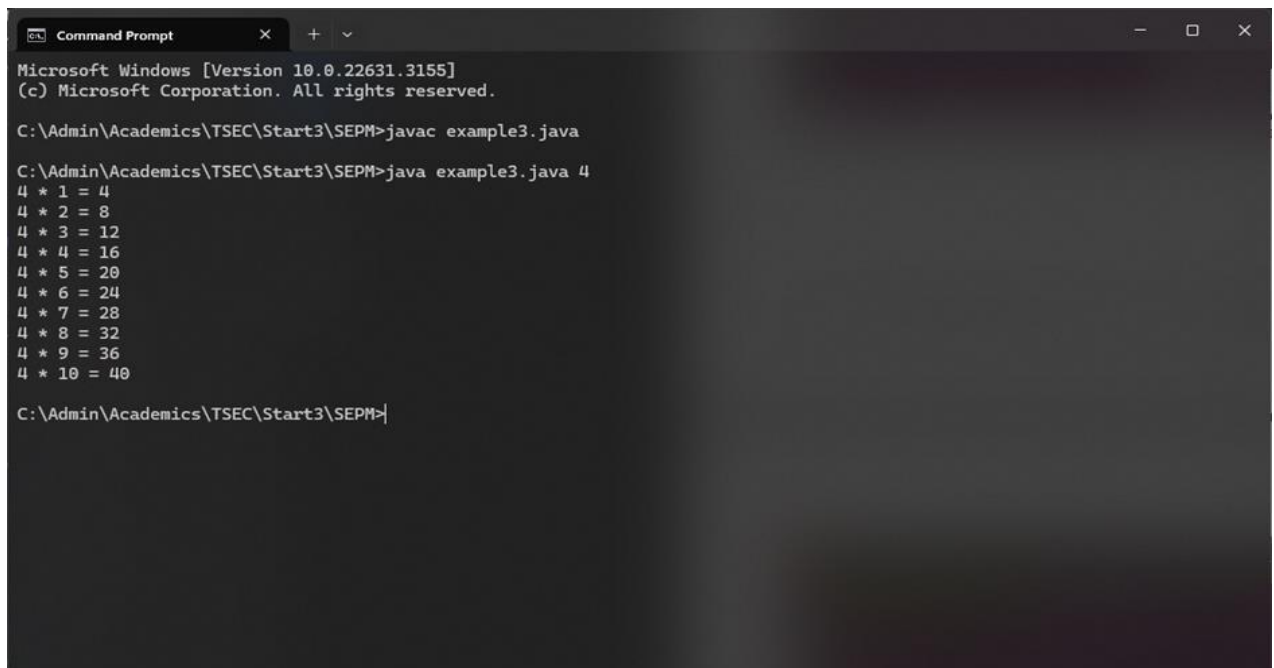
Example 3.2: Running a Java program with parameters

Creating a Java program with an input argument:



```
class Main {  
    public static void main(String[] args) {  
        int i;  
        int num = Integer.parseInt(args[0]);  
        for (i = 1; i < 11; i++) {  
            System.out.println(num + " * " + i + " = " + num * i);  
        }  
    }  
}
```

Testing the program on the terminal:





```
Microsoft Windows [Version 10.0.22631.3155]  
(c) Microsoft Corporation. All rights reserved.  
  
C:\Admin\Academics\TSEC\Start3\SEPM>javac example3.java  
  
C:\Admin\Academics\TSEC\Start3\SEPM>java example3.java 4  
4 * 1 = 4  
4 * 2 = 8  
4 * 3 = 12  
4 * 4 = 16  
4 * 5 = 20  
4 * 6 = 24  
4 * 7 = 28  
4 * 8 = 32  
4 * 9 = 36  
4 * 10 = 40  
  
C:\Admin\Academics\TSEC\Start3\SEPM>
```


Creating a new freestyle project:


Enter an item name


» Required field


**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 a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**
Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

Parameterise the project by adding a string parameter as follows:

☒ This project is parameterized ?

String Parameter ?

Name ?

Default Value ?

Description ?

Plain text [Preview](#)

☐ Trim the string ?

Add Parameter ▾

Configure the build steps:

Build Steps

≡ Execute Windows batch command ?

Command

See the list of available environment variables

```
javac C:\Admin\Academics\TSEC\Start3\SEPM\example3.java
java C:\Admin\Academics\TSEC\Start3\SEPM\example3.java %num%
```

Advanced ▾

Add build step ▾

Entering the parameter for the build:

Project Example4

This build requires parameters:

num

25



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\Jenkins\jenkins\workspace\Example4
[Example4] $ cmd /c call C:\WINDOWS\TEMP\jenkins15119185770823247708.bat

C:\ProgramData\Jenkins\jenkins\workspace\Example4>javac C:\Admin\Academics\TSEC\Start3\SEPM\example3.java

C:\ProgramData\Jenkins\jenkins\workspace\Example4>java C:\Admin\Academics\TSEC\Start3\SEPM\example3.java 25
25 * 1 = 25
25 * 2 = 50
25 * 3 = 75
25 * 4 = 100
25 * 5 = 125
25 * 6 = 150
25 * 7 = 175
25 * 8 = 200
25 * 9 = 225
25 * 10 = 250

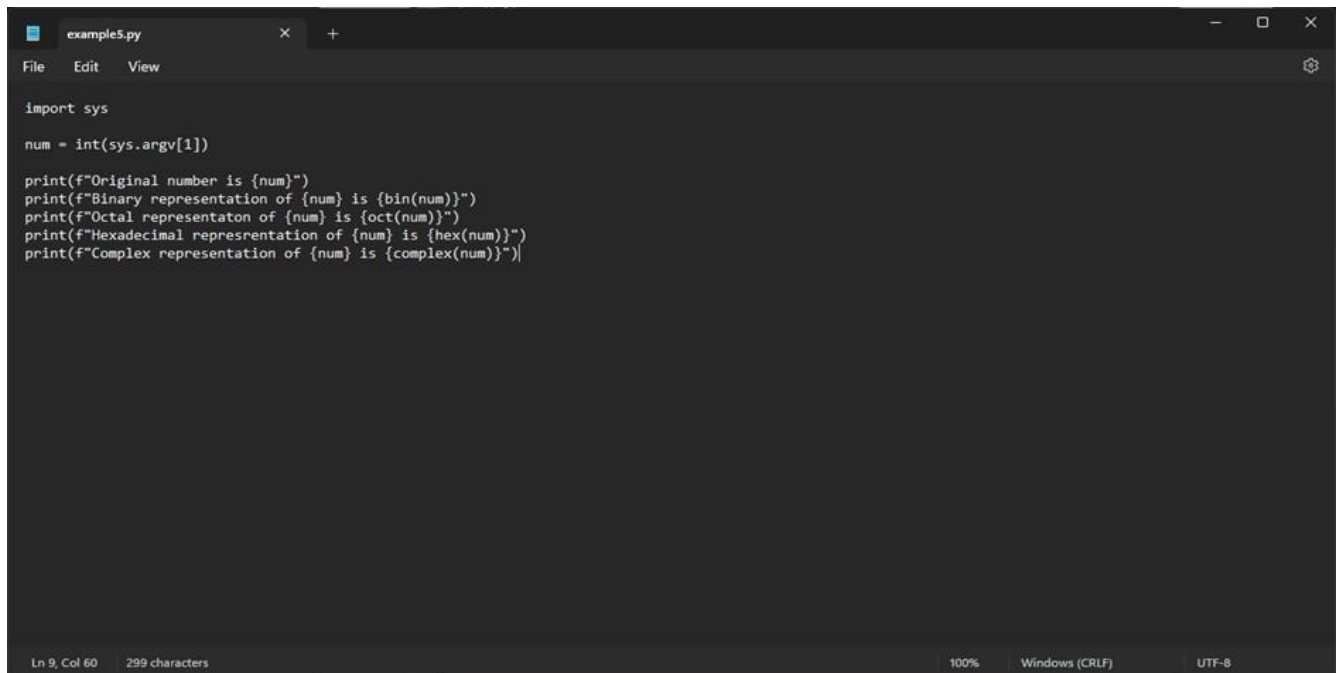
C:\ProgramData\Jenkins\jenkins\workspace\Example4>exit 0
Finished: SUCCESS
```

Vedanshi Shethia
2311137

Example 5

Example 5.1: Running a Python program

Creating a simple Python script:



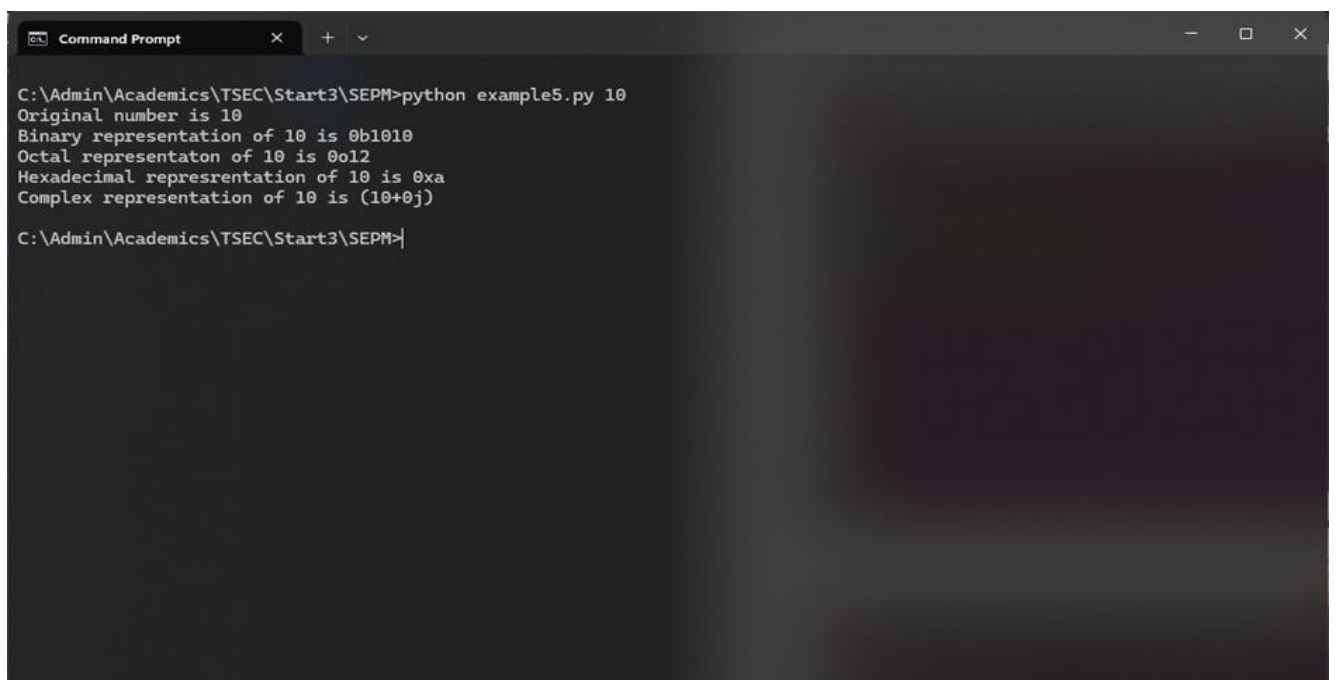
```
example5.py
File Edit View
import sys

num = int(sys.argv[1])

print(f"Original number is {num}")
print(f"Binary representation of {num} is {bin(num)}")
print(f"Octal representation of {num} is {oct(num)}")
print(f"Hexadecimal representation of {num} is {hex(num)}")
print(f"Complex representation of {num} is {complex(num)}")

Ln 9, Col 60 299 characters 100% Windows (CRLF) UTF-8
```

Running the Python script on the terminal:




```
Command Prompt
C:\Admin\Academics\TSEC\Start3\SEPM>python example5.py 10
Original number is 10
Binary representation of 10 is 0b1010
Octal representation of 10 is 0o12
Hexadecimal representation of 10 is 0xa
Complex representation of 10 is (10+0j)
C:\Admin\Academics\TSEC\Start3\SEPM>
```


Creating a new freestyle project:

Enter an item name


» Required field

**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 a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**Multibranch Pipeline**

Creates a set of Pipeline projects according to detected branches in one SCM repository.

**Organization Folder**

Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

Parameterising the project with a string parameter as follows:

☒ This project is parameterized ?

String Parameter ?

Name ?

Default Value ?

Description ?

Plain text [Preview](#)

☐ Trim the string ?

Add Parameter

Configuring the build steps:

Build Steps

≡ Execute Windows batch command ?

Command

See [the list of available environment variables](#)

python C:\Admin\Academics\TSEC\Start3\SEPM\example5.py %num%

Advanced ▾

Add build step ▾

Setting the parameter for the build:

Project Example5

This build requires parameters:

num

▶ 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\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 0o12
Hexadecimal representation of 10 is 0xa
Complex representation of 10 is (10+0j)

C:\ProgramData\Jenkins\.jenkins\workspace\Example5>exit 0
Finished: SUCCESS
```

Some Screenshots:

The image shows two screenshots of the Jenkins web interface. The top screenshot displays the 'Configure' page for a pipeline named 'AIDS_Pipeline'. The 'Script' tab is selected, showing a Groovy pipeline script with four stages: 'build', 'test', 'deploy', and 'postdeploy'. Each stage contains an 'echo' step. The 'Use Groovy Sandbox' checkbox is checked. The bottom screenshot shows the 'Status' view of the 'AIDS_Pipeline'. It includes a 'Stage View' table with columns for 'Build', 'Test', 'Deploy', and 'Postdeploy', showing average stage times and a bar chart. Below the table is a 'Permalinks' section with links to the last build, last stable build, last successful build, and last completed build. On the left, there is a sidebar with navigation options like 'Status', 'Changes', 'Build Now', 'Configure', etc., and a 'Builds' section showing the latest build (#1) from February 27, 2023.

Configure

Advanced

General

Build Triggers

Advanced Project Options

Pipeline

Pipeline

Definition

Pipeline script

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('build') {
6       steps {
7         echo 'Building.. This is the build phase'
8       }
9     }
10    stage('test') {
11      steps {
12        echo 'Testing.. This is the testing phase'
13      }
14    }
15    stage('deploy') {
16      steps {
17        echo 'Deploying.... This is the deployment phase'
18      }
19    }
20    stage('Postdeploy') {
21      steps {
22        echo 'Postdeployment phase....'
23      }
24    }
25  }
26 }
27
```

☒ Use Groovy Sandbox

[Pipeline Syntax](#)

Save Apply

Jenkins

Dashboard > AIDS_Pipeline

Status

Changes

Build Now

Configure

Delete Pipeline

Full Stage View

Rename

Pipeline Syntax

AIDS_Pipeline

Add description

Stage View

	Build	Test	Deploy	Postdeploy
Average stage times:	46ms	44ms	34ms	43ms
(Average full run time ~507ms)				
Feb 27 11:58	46ms	44ms	34ms	43ms

Permalinks

- Last build (#1), 2 yr 0 mo ago
- Last stable build (#1), 2 yr 0 mo ago
- Last successful build (#1), 2 yr 0 mo ago
- Last completed build (#1), 2 yr 0 mo ago

Builds

Filter

February 27, 2023

#1 1:58 PM

localhost:8080/job/AIDS_Pipeline/1/

REST API Jenkins 2.492.2

Jenkins

Search

🔔

🛡️

🔥

admin

log out

Dashboard > AIDS_Pipeline

Status

<> Changes

▶ Build Now

⚙️ Configure

🗑️ Delete Pipeline

🔍 Full Stage View

✎ Rename

🔗 Pipeline Syntax

Builds

Filter

Today

🟢 #2 1:45 PM

February 27, 2023

🟢 #1 1:58 PM

🟢 AIDS_Pipeline

Add description

Stage View

Average stage times:
(Average full run time: ~7s)

	Build	Test	Deploy	Postdeploy
Mar 11 13:46	160ms	53ms	66ms	66ms
Feb 27 13:58	46ms	44ms	34ms	43ms

Permalinks

- Last build (#1), 2 yr 0 mo ago
- Last stable build (#1), 2 yr 0 mo ago
- Last successful build (#1), 2 yr 0 mo ago
- Last completed build (#1), 2 yr 0 mo ago

Jenkins

Search

🔔

🛡️

🔥

admin

log out

Dashboard > AIDS_Pipeline > #2 > Pipeline Steps

Status

<> Changes

📄 Console Output

🔍 Edit Build Information

🗑️ Delete build #2

🔄 Restart from Stage

🎮 Replay

Pipeline Steps

📁 Workspaces

← Previous Build

Step

Arguments

Status

Start of Pipeline - (2.1 sec in block)

🟢

node - (0.78 sec in block)

🟢

node block - (0.51 sec in block)

🟢

stage - (0.19 sec in block)

Build

🟢

stage block (Build) - (0.13 sec in block)

🟢

echo - (13 ms in self)

Building... This is the build phase

🟢

stage - (68 ms in block)

Test

🟢

stage block (Test) - (39 ms in block)

🟢

echo - (13 ms in self)

Testing... This is the testing phase

🟢

stage - (80 ms in block)

Deploy

🟢

stage block (Deploy) - (40 ms in block)

🟢

echo - (1 ms in self)

Deploying... This is the deployment phase

🟢

stage - (67 ms in block)

Postdeploy

🟢

stage block (Postdeploy) - (40 ms in block)

🟢

echo - (2 ms in self)

Postdeployment phase...

🟢

Jenkins 2.492.2

Dashboard > AIDS_Pipeline > #2

Console Output

```
Started by user admin
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\AIDS_Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] echo
Building.. This is the build phase
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] echo
Testing.. This is the testing phase
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Deploy)
[Pipeline] echo
Deploying.... This is the deployment phase
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Postdeploy)
[Pipeline] echo
Postdeployment phase....
[Pipeline] }
[Pipeline] // stage
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

REST API Jenkins 2.492.2

Stage Logs (Build)

Print Message -- Building.. This is the build phase (self time 13ms)

Building.. This is the build phase

Stage View

	Build	Test	Deploy	Postdeploy
Average stage times: (Average full run time: ~1s)	91ms	50ms	51ms	58ms
#3 Mar 11 13:47 No Changes	69ms	53ms	54ms	67ms
#2 Mar 11 13:46 No Changes	160ms	53ms	66ms	66ms
#1 Feb 27 13:58 No Changes	45ms	44ms	34ms	43ms

Permalinks

- Last build (#3), 27 ms ago
- Last stable build (#2), 1 min 40 sec ago
- Last successful build (#2), 1 min 40 sec ago
- Last completed build (#2), 1 min 40 sec ago

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