

120A3051

Shreya Idate

Batch: E3

## Experiment No: 5

### AIM:

- i. To set up and build a Java, Maven /Ant and Python jobs in Jenkins.
- ii. To build the pipeline of jobs using Maven / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server.

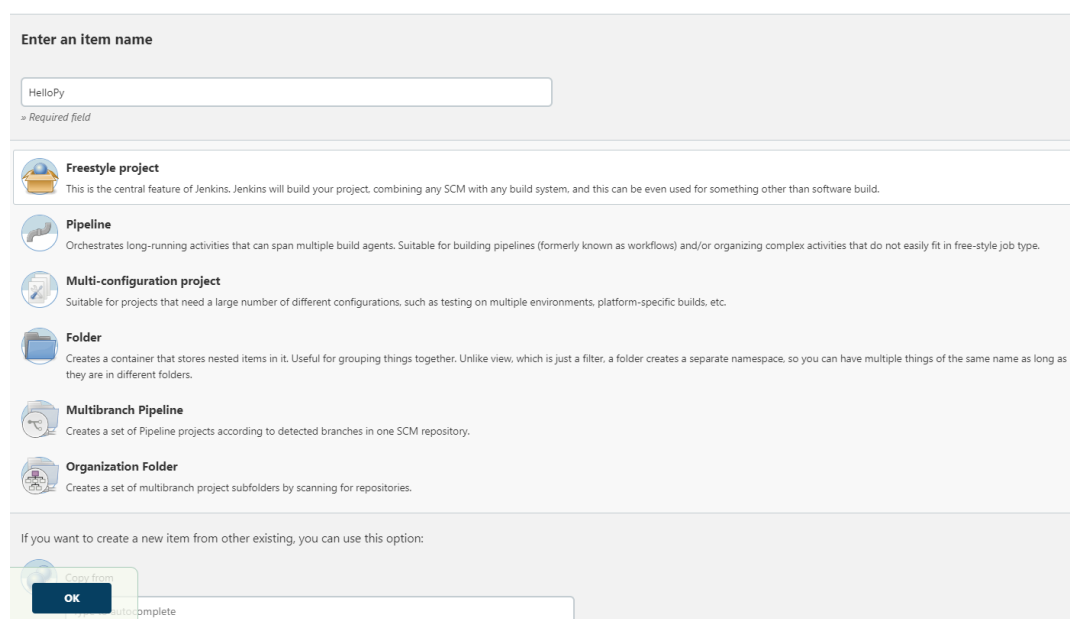
### THEORY:

Jenkins is a free and open-source automation server. It helps automate the parts of software development related to building, testing, and deploying, facilitating continuous integration and continuous delivery. It is a server-based system that runs in servlet containers such as Apache Tomcat.

- A. To set up and build a Java, Maven /Ant and Python jobs in Jenkins.

### Steps for Creating and Testing Python job in Jenkins:

1. Install python and set environment settings on our machine.
2. Install Python plugins in Jenkins i.e. Manage Jenkins-> Manage Plugins -> Available->Python Plugin -> select the plugin and click on install without restart.
3. Create a new job as freestyle project in Jenkins. New Item - > Freestyle Project. Add some optional description.



Enter an item name

HelloPy

» Required field

**Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

**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.

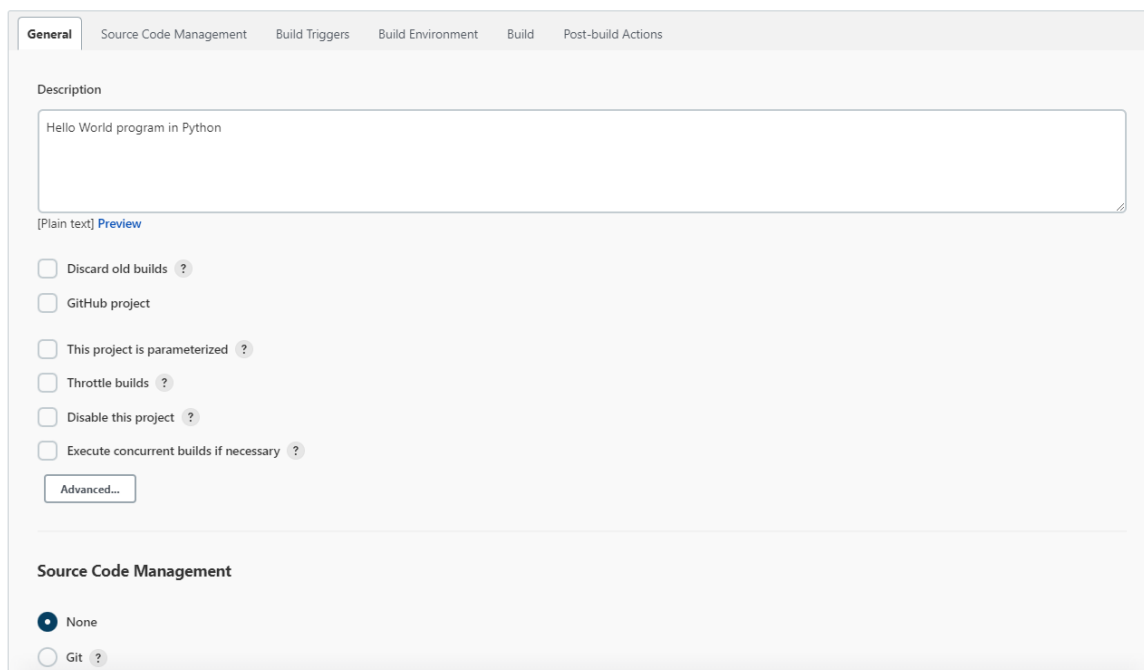
**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**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:

Copy from  
OK  
Autocomplete



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

Description

Hello World program in Python

[Plain text] [Preview](#)

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Disable this project ?

☐ Execute concurrent builds if necessary ?

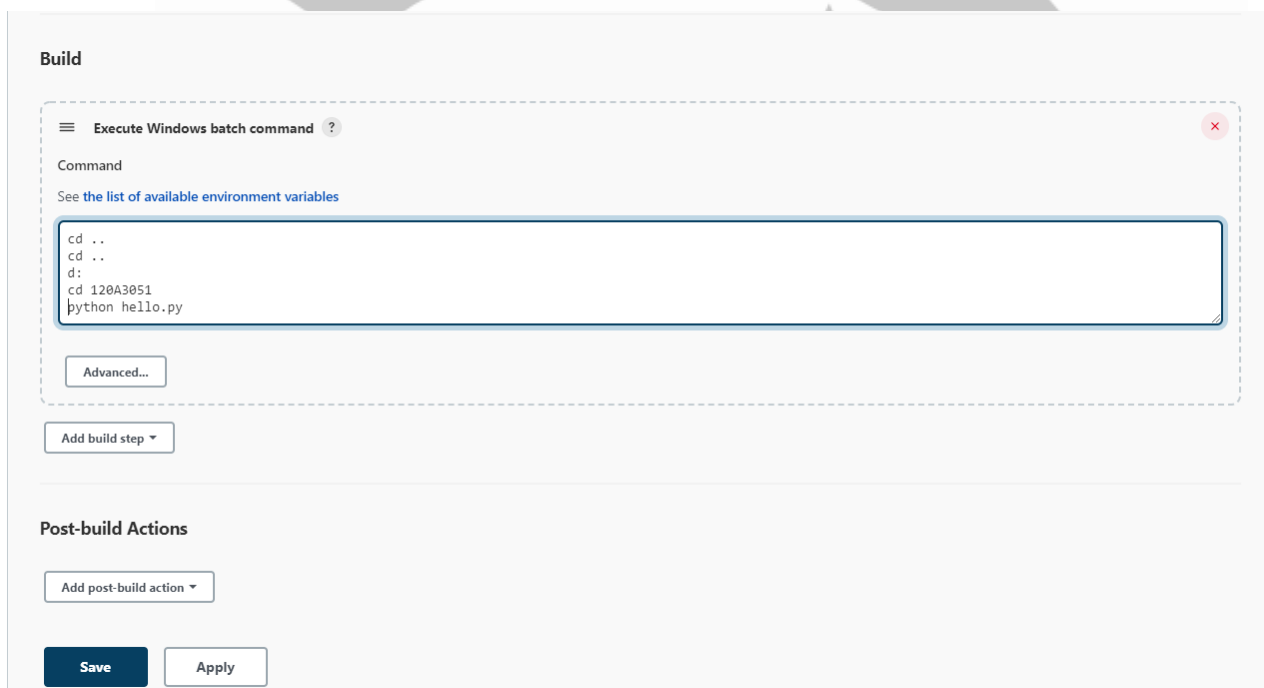
[Advanced...](#)

**Source Code Management**

☒ None

☐ Git ?

4. Configure -> Build section, add build steps-> commands to run the py script. Implicit = >select Execute python script => write some implicit python script here -> save and apply. Go to step 5. Explicit => write the following Build steps, assuming your python script is stored in this location D:\python scripts



**Build**

Execute Windows batch command ?

Command

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

```
cd ..
cd ..
d:
cd 120A3051
python hello.py
```

[Advanced...](#)

[Add build step](#)

**Post-build Actions**

[Add post-build action](#)

[Save](#) [Apply](#)

## 5. Build Now - > Console Output

### Console Output

```
Started by user Shreya Idate
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\.jenkins\workspace\HelloPy
[HelloPy] $ cmd /c call C:\Windows\TEMP\jenkins14028921767187002157.bat

C:\ProgramData\Jenkins\.jenkins\workspace\HelloPy>d:

D:\>cd 120A3051

D:\120A3051>python hello.py
Hello World!

D:\120A3051>exit 0
Finished: SUCCESS
```

### Steps for Testing Java job in Jenkins:

1. Install Java and set environment settings.
2. Create a new job as freestyle project.



3. Configure -> Build section, commands to run the Java program. Add build steps-> select Windows batch commands. Go to Configure -> Build section, write the following Build steps, assuming your java program is stored in this location D:\java program

E:

```
javac sum.java
```

```
java sum %a% %b%
```

-> save and apply

The screenshot shows the 'General' tab of a Jenkins job configuration. The 'Description' field contains the text 'Java program to print sum of 2 integers.' Below this, there are several unchecked checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterised', 'Throttle builds', 'Disable this project', and 'Execute concurrent builds if necessary'. An 'Advanced...' button is located below these options. The 'Source Code Management' section shows 'None' selected with a radio button, and 'Git' is also available but unchecked. At the bottom, there are 'Save' and 'Apply' buttons.

The screenshot shows the 'Build Triggers' and 'Build Environment' tabs of a Jenkins job configuration. In the 'Build Triggers' section, 'GitHub hook trigger for GITScm polling' and 'Poll SCM' are unchecked. The 'Build Environment' section has several unchecked checkboxes: 'Delete workspace before build starts', 'Use secret text(s) or file(s)', 'Add timestamps to the Console Output', 'Inspect build log for published Gradle build scans', 'Terminate a build if it's stuck', and 'With Ant'. The 'Build' section at the bottom has an 'Add build step +' button. A dropdown menu is open below it, showing a list of build steps: 'Execute Python script', 'Execute Windows batch command' (which is highlighted), 'Execute shell', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', and 'Set build status to "pending" on GitHub commit'.

General

Source Code Management

Build Triggers

Build Environment

Build

Post-build Actions

☒ This project is parameterised ?

String Parameter ?

Name ?

a

Default Value ?

10

Description ?

[Plain text] [Preview](#)

☐ Trim the string ?

String Parameter ?

Name ?

b

Default Value ?

20

Description ?

[Plain text] [Preview](#)

☐ Trim the string ?

Add Parameter ▾

Build

Execute Windows batch command ?

Command

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

d:  
cd 120A3051  
javac Sum.java  
java Sum %a% %b%

Advanced...

Add build step ▾

Dashboard > Sum >

↑ Back to Dashboard

☰ Status

</> Changes

📁 Workspace

▶ Build with Parameters

⚙️ Configure

🗑️ Delete Project

✎ Rename

### Project Sum

This build requires parameters:

a

b

Build

Build History trend

Filter builds...

✓ #3	05-Aug-2022 12:40
✓ #2	05-Aug-2022 12:20
✓ #1	05-Aug-2022 12:12

Atom feed for all Atom feed for failures

↑ Back to Project

☰ Status

</> Changes

📄 Console Output

📄 View as plain text

⚙️ Edit Build Information

🗑️ Delete build '#3'

🔍 Parameters

← Previous Build

## Console Output

```
Started by user Shreya Idade
Running as SYSTEM
Building in workspace C:\ProgramData\Jenkins\jenkins\workspace\Sum
[Sum] $ cmd /c call C:\Windows\TEMP\jenkins8327951732965565944.bat

C:\ProgramData\Jenkins\jenkins\workspace\Sum>d:

D:\>cd 120A3051

D:\120A3051>javac Sum.java

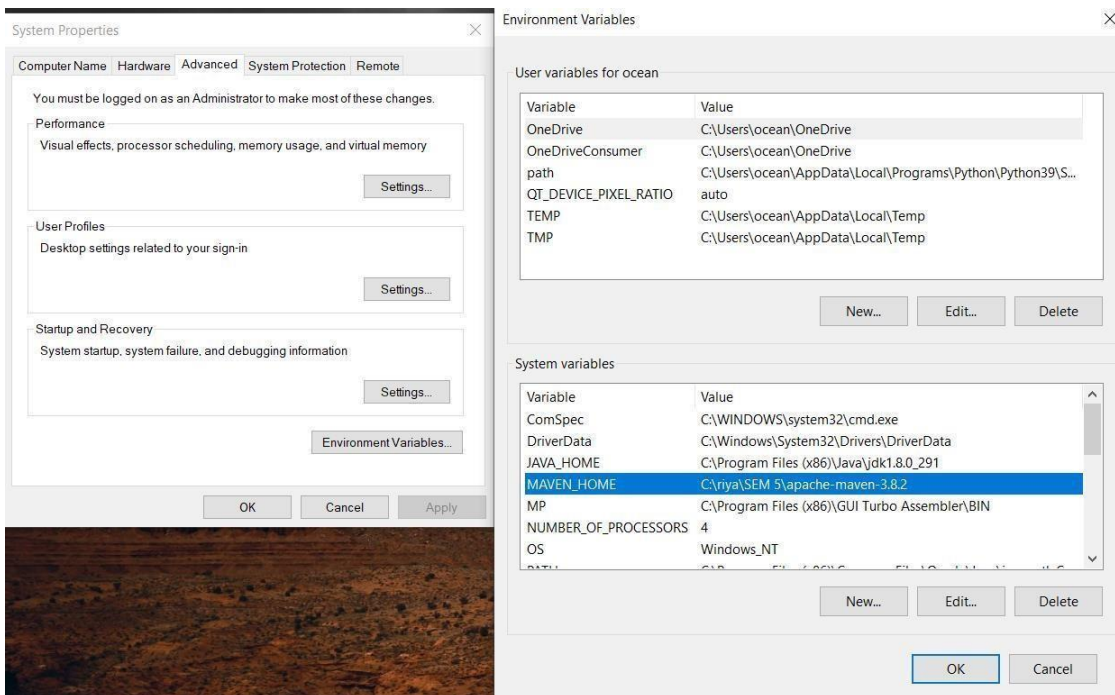
D:\120A3051>java Sum 10 20

Sum = 30

D:\120A3051>exit 0
Finished: SUCCESS
```

### Steps for Building Maven job from the GitHub remote repository into Jenkins:

1. Install Maven and set environment settings. Make sure you've previously set java settings properly.  
Install Maven Integration plugin if not installed earlier and got to Manage Jenkins -> Globaltool configurations -> Set Name => Maven <version> and Maven Home => path value of MAVEN\_HOME.



2. Build maven project in your machine using mvn commands to learn how to create and build any maven project as shown here: <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\exam>mvn -v
Apache Maven 3.8.6 (84538c9988a25aec085021c365c560670ad80f63)
Maven home: C:\apache-maven-3.8.6
Java version: 11.0.15.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-11.0.15.1
Default locale: en_IN, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

C:\Users\exam>

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1466]
(c) Microsoft Corporation. All rights reserved.

C:\ProgramData\Jenkins\jenkins\workspace>mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app
-DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false
[INFO] Scanning for projects...
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/2.5/maven-clean-plugin-2.5.pom (3.9 kB at 1.7 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/22/maven-pl
```

```
C:\Windows\System32\cmd.exe
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-bundles/1.4/maven-archetype-bundles-1.4.pom (4.5 kB at 15 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar (7.1 kB at 24 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archetypes/maven-archetype-quickstart/1.4/maven-archetype-quickstart-1.4.jar (7.1 kB at 24 kB/s)
[INFO]
[INFO] Using following parameters for creating project from Archetype: maven-archetype-quickstart:1.4
[INFO] -----
[INFO] Parameter: groupId, Value: com.mycompany.app
[INFO] Parameter: artifactId, Value: my-app
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Parameter: package, Value: com.mycompany.app
[INFO] Parameter: packageInPathFormat, Value: com/mycompany/app
[INFO] Parameter: package, Value: com.mycompany.app
[INFO] Parameter: groupId, Value: com.mycompany.app
[INFO] Parameter: artifactId, Value: my-app
[INFO] Parameter: version, Value: 1.0-SNAPSHOT
[INFO] Project created from Archetype in dir: C:\ProgramData\Jenkins\jenkins\workspace\my-app
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 08:28 min
[INFO] Finished at: 2022-09-12T10:34:18+05:30
[INFO] -----
C:\ProgramData\Jenkins\jenkins\workspace>

C:\ProgramData\Jenkins\jenkins\workspace>cd my-app
C:\ProgramData\Jenkins\jenkins\workspace\my-app>mvn package
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.mycompany.app:my-app >-----
[INFO] Building my-app 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resources-plugin/3.0.2/maven-resources-plugin-3.0.2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-resources-plugin/3.0.2/maven-resources-plugin-3.0.2.pom (7.1 kB at 7.4 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/30/maven-plugins-30.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-io/2.7.1/plexus-io-2.7.1.jar (86 kB at 144 kB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/tukaani/xz/1.5/xz-1.5.jar (100 kB at 134 kB/s)
[INFO] Building jar: C:\ProgramData\Jenkins\jenkins\workspace\my-app\target\my-app-1.0-SNAPSHOT.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 45.388 s
[INFO] Finished at: 2022-09-12T10:55:31+05:30
[INFO] -----
C:\ProgramData\Jenkins\jenkins\workspace\my-app>


C:\ProgramData\Jenkins\jenkins\workspace\my-app>java -cp target/my-app-1.0-SNAPSHOT.jar com.mycompany.app.App
Hello World!


C:\ProgramData\Jenkins\jenkins\workspace\my-app>
```


### 3. Create a maven job as Maven Project.


Enter an item name


Required field


 **Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.


 **Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

 **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.

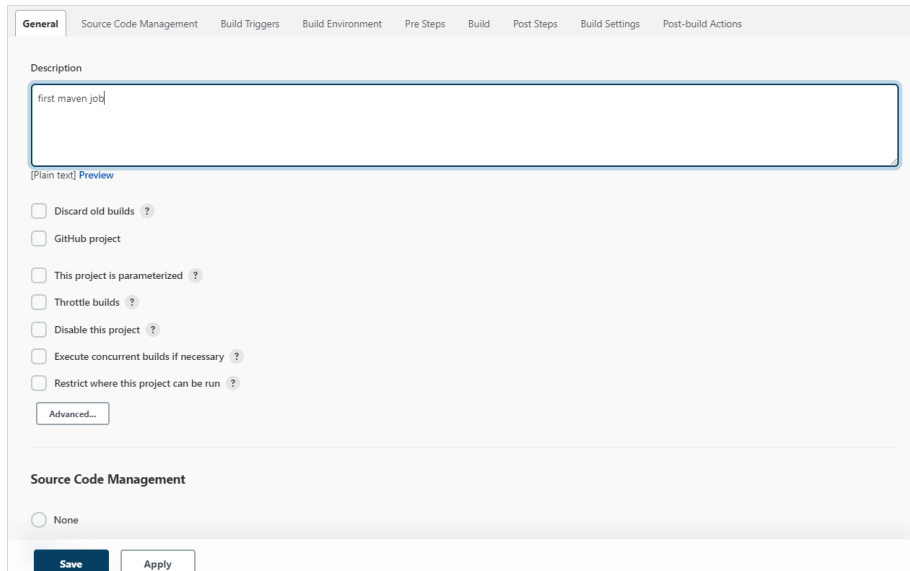
 **Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.

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

Use an existing item from other existing, you can use this option:

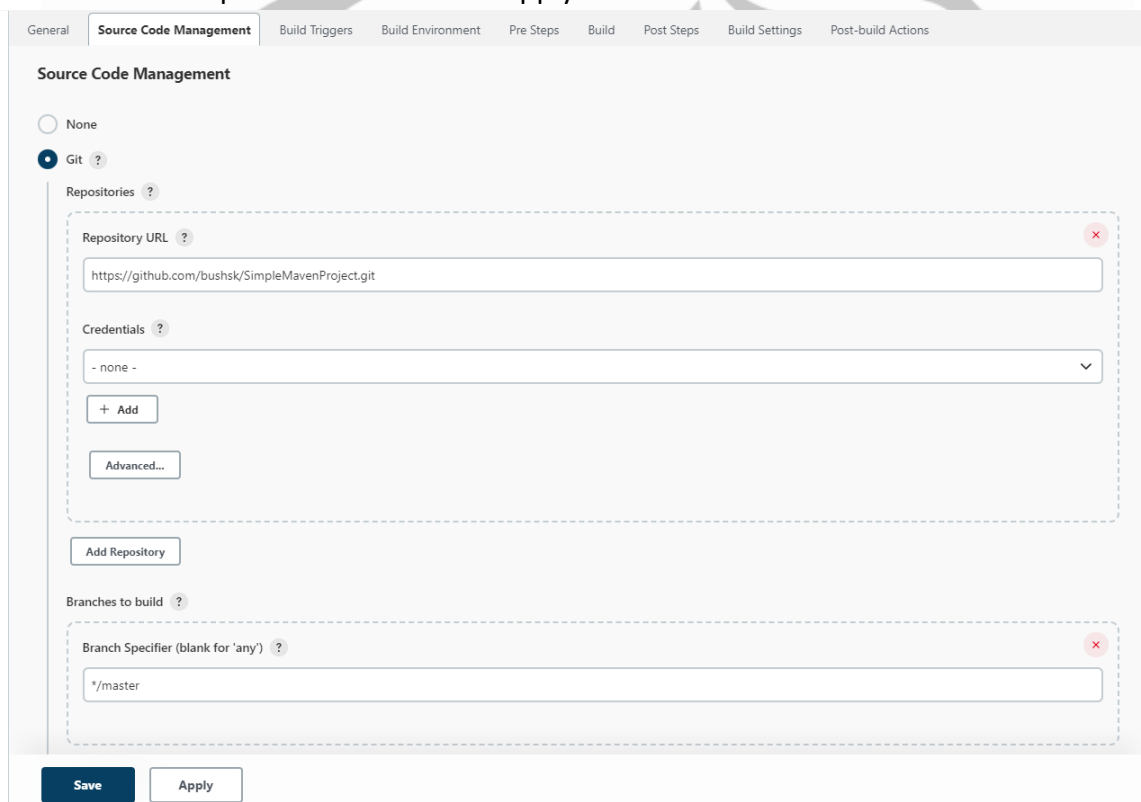


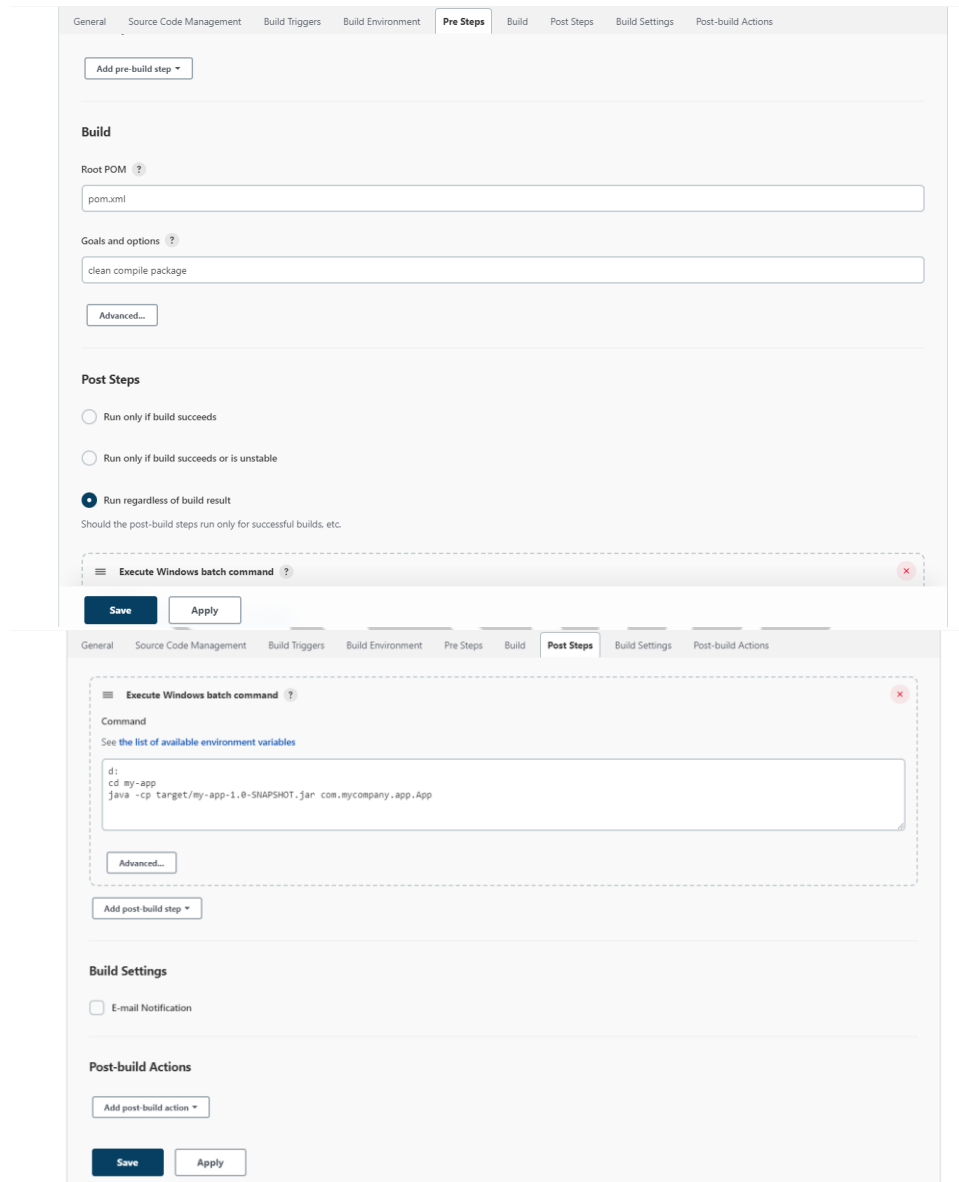
4. In Configure section -> go to Source Code Management -> Git -> paste this repository URL => <https://github.com/bushsk/SimpleMavenProject.git> (Assuming our maven project is on GitHub repository).



Note: Ensure that the GitHub Plugin is installed at this point. If it has not been installed, then do install it from Manage Jenkins-> Manage Plugins

5. In Build section -> Write the goals and options as => clean compile package. Make sure Root POM is set to pom.xml. -> save and apply





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

Add pre-build step ▾

**Build**

Root POM ?  
pom.xml

Goals and options ?  
clean compile package

Advanced...

**Post Steps**

☐ Run only if build succeeds

☐ Run only if build succeeds or is unstable

☒ Run regardless of build result  
Should the post-build steps run only for successful builds, etc.

Execute Windows batch command ?

Save Apply

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

Execute Windows batch command ?

Command  
See the list of available environment variables

```
d:
cd my-app
java -cp target/my-app-1.0-SNAPSHOT.jar com.mycompany.app.App
```

Advanced...

Add post-build step ▾

**Build Settings**

☐ E-mail Notification

**Post-build Actions**

Add post-build action ▾

Save Apply

6. Go to build now -> console output and verify whether the .jar file with the given artefact has been created or not inside the given path shown in the console output.

The screenshot displays the Jenkins web interface for a build job named 'MavenJob'. The 'Console Output' tab is selected, showing the following log details:

- Started by user Shreya Idarte**
- Running as SVSTDR
- Building on the built-in node in workspace C:\ProgramData\Jenkins\jenkins\workspace\MavenJob
- The recommended git tool is: NONE
- No credentials specified
- git.exe rev-parse --resolve-git-dir C:\ProgramData\Jenkins\jenkins\workspace\MavenJob\git # timeout=10
- Fetching changes from the remote Git repository
- git.exe config remote.origin.url https://github.com/bushsk/SimpleMavenProject.git # timeout=10
- Fetching upstream changes from https://github.com/bushsk/SimpleMavenProject.git
- git.exe --version # timeout=10
- git --version # 'git version 2.35.1.windows.2'
- git.exe fetch --tags --force --progress -- https://github.com/bushsk/SimpleMavenProject.git refs/heads/\*:refs/remotes/origin/\* # timeout=10
- git.exe rev-parse "refs/remotes/origin/master" {commit} # timeout=10
- Checking out Revision db95abdf4e7c2c5585b66f3235e3a0ed09006d68 (refs/remotes/origin/master)
- git.exe config core.sparsecheckout # timeout=10
- git.exe checkout -f db95abdf4e7c2c5585b66f3235e3a0ed09006d68 # timeout=10
- Commit message: "Merge pull request #1 from medined/dependabot/maven/junit-junit-4.13.1"
- git.exe rev-list --no-walk db95abdf4e7c2c5585b66f3235e3a0ed09006d68 # timeout=10
- Parsing POMs
- Modules changed, recalculating dependency graph
- Established TCP socket on 54147
- [MavenJob] \$ "C:\Program Files\Java\jdk1.8.0\_321\bin/java" -cp C:\ProgramData\Jenkins\jenkins\plugins\maven-plugin\WEB-INF\lib\maven3-agent-1.13.jar;D:\apache-maven-3.8.6\boot\plexus-classworlds-2.6.0.jar;D:\apache-maven-3.8.6\conf\logging-jenkins.maven3.agent.Maven3Main D:\apache-maven-3.8.6 C:\ProgramData\Jenkins\workspace\MavenJob\pom.xml com.affy:SimpleMavenProject:1.0-SNAPSHOT
- [JENKINS REMOTING CAPACITY]-----channel started
- Executing Maven: -B -f C:\ProgramData\Jenkins\jenkins\workspace\MavenJob\pom.xml clean compile package
- [INFO] Scanning for projects...
- [INFO]
- [INFO] -----[ com.affy:SimpleMavenProject ]-----
- [INFO] Building SimpleMavenProject 1.0-SNAPSHOT
- [INFO] -----[ jar ]-----
- [INFO]
- [INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ SimpleMavenProject ---
- [INFO] Nothing to compile - all classes are up to date
- [INFO]
- [INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ SimpleMavenProject ---
- [INFO] Using 'UTF-8' encoding to copy filtered resources.
- [INFO] skip non existing resourceDirectory C:\ProgramData\Jenkins\jenkins\workspace\MavenJob\src\test\resources
- [INFO]
- [INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ SimpleMavenProject ---
- [INFO] No sources to compile
- [INFO]
- [INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ SimpleMavenProject ---
- [INFO] No tests to run.
- [JENKINS] Recording test results
- [INFO]
- [INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ SimpleMavenProject ---
- [INFO] Building jar: C:\ProgramData\Jenkins\jenkins\workspace\MavenJob\target\SimpleMavenProject-1.0-SNAPSHOT.jar
- [INFO] -----
- [INFO] BUILD SUCCESS
- [INFO] -----
- [INFO] Total time: 1.708 s
- [INFO] Finished at: 2022-09-14T15:12:29+05:30
- [INFO] -----
- Waiting for Jenkins to finish collecting data
- [JENKINS] Archiving C:\ProgramData\Jenkins\jenkins\workspace\MavenJob\pom.xml to com.affy:SimpleMavenProject:1.0-SNAPSHOT/SimpleMavenProject-1.0-SNAPSHOT.pom
- [JENKINS] Archiving C:\ProgramData\Jenkins\jenkins\workspace\MavenJob\target\SimpleMavenProject-1.0-SNAPSHOT.jar to com.affy:SimpleMavenProject:1.0-SNAPSHOT/SimpleMavenProject-1.0-SNAPSHOT.jar
- channel stopped
- [MavenJob] \$ cmd /c call C:\Windows\TEMP\jenkins13664948543124068562.bat
- C:\ProgramData\Jenkins\jenkins\workspace\MavenJob>d:
- D:\>cd my-app
- D:\my-app>java -cp target/my-app-1.0-SNAPSHOT.jar com.mycompany.app.App
- Hello World!
- D:\my-app>exit 0
- Finished: SUCCESS

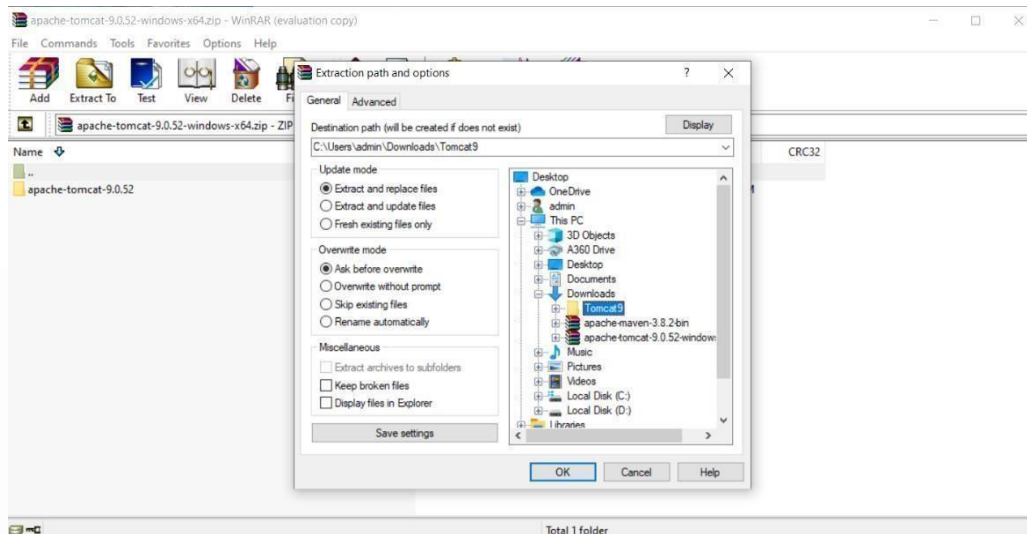
## B. To build the pipeline of jobs using Maven / Ant in Jenkins, create a pipeline script to Test and deploy an application over the tomcat server.

Steps for Building Maven job from the GitHub remote repository into Jenkins using Jenkins pipeline:  
Setup and run apache tomcat server:

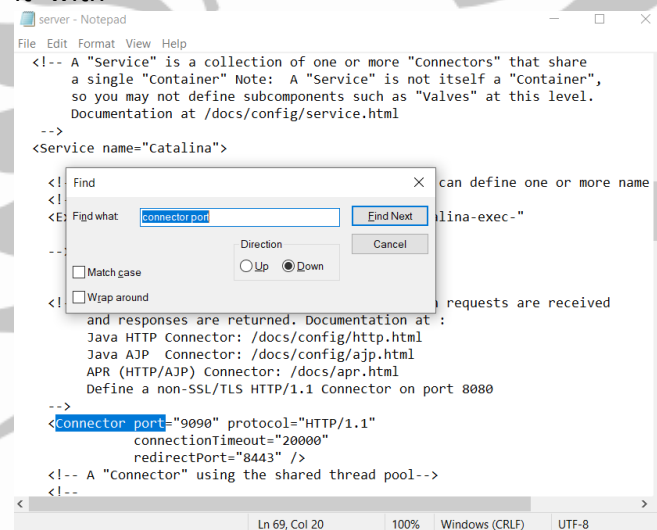
1.

a. Download apache tomcat server from official apache's tomcat website and

Install it by extracting and set it to run on port 9090 or any other port since Jenkins is already running on 8080 port and by default tomcat also runs on 8080.



- b. To do the above, go to the apache tomcat directory and find server.xml inside config folder. Find this line by Ctrl+f -<Connector port="8080"|| and replace it with



-<Connector port="9090" and save it.


- c. Start the server by running the startup.bat file found inside apache-tomcat-9.0.52\bin folder and go to browser and test if the home page of server is up or not onlocalhost:9090.

```
Tomcat
12-Sep-2022 10:10:38.324 WARNING [main] org.apache.catalina.util.SessionIdGeneratorBase.createSecureRandom Creation of SecureRandom instance for session ID generation using [SHA1PRNG] took [270] milliseconds.
12-Sep-2022 10:10:38.355 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application to directory [C:\apache-tomcat-9.0.65\webapps\docs] has finished in [802] ms
12-Sep-2022 10:10:38.355 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\apache-tomcat-9.0.65\webapps\examples]
12-Sep-2022 10:10:39.829 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application to directory [C:\apache-tomcat-9.0.65\webapps\examples] has finished in [1,474] ms
12-Sep-2022 10:10:39.829 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\apache-tomcat-9.0.65\webapps\host-manager]
12-Sep-2022 10:10:39.929 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application to directory [C:\apache-tomcat-9.0.65\webapps\host-manager] has finished in [100] ms
12-Sep-2022 10:10:39.929 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\apache-tomcat-9.0.65\webapps\manager]
12-Sep-2022 10:10:39.998 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application to directory [C:\apache-tomcat-9.0.65\webapps\manager] has finished in [69] ms
12-Sep-2022 10:10:39.998 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application directory [C:\apache-tomcat-9.0.65\webapps\ROOT]
12-Sep-2022 10:10:40.029 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application to directory [C:\apache-tomcat-9.0.65\webapps\ROOT] has finished in [31] ms
12-Sep-2022 10:10:40.045 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-4748"]
12-Sep-2022 10:10:40.145 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in [2773] milliseconds
```

Home Documentation Configuration Examples Wiki Mailing Lists Find Help

## Apache Tomcat/9.0.65

If you're seeing this, you've successfully installed Tomcat. Congratulations!

 Recommended Reading:

- [Security Considerations How-To](#)
- [Manager Application How-To](#)
- [Clustering/Session Replication How-To](#)

Server Status  
Manager App  
Host Manager

### Developer Quick Start

- [Tomcat Setup](#)
- [First Web Application](#)
- [Realms & AAA](#)
- [JDBC DataSources](#)
- [Examples](#)
- [Servlet Specifications](#)
- [Tomcat Versions](#)

#### Managing Tomcat

For security, access to the `manager` webapp is restricted. Users are defined in:

```
$CATALINA_HOME/conf/tomcat-users.xml
```

In Tomcat 9.0 access to the manager application is split between different users. [Read more...](#)

[Release Notes](#)  
[Changelog](#)  
[Migration Guide](#)  
[Security Notices](#)

#### Documentation

[Tomcat 9.0 Documentation](#)  
[Tomcat 9.0 Configuration](#)  
[Tomcat Wiki](#)

Find additional important configuration information in:

```
$CATALINA_HOME/RUNNING.txt
```

Developers may be interested in:

- [Tomcat 9.0 Bug Database](#)
- [Tomcat 9.0 JavaDocs](#)
- [Tomcat 9.0 Git Repository at GitHub](#)

#### Getting Help

[FAQ and Mailing Lists](#)

The following mailing lists are available:

- [tomcat-announce](#)  
Important announcements, releases, security vulnerability notifications. (Low volume).
- [tomcat-users](#)  
User support and discussion
- [tomcat-dev](#)  
User support and discussion for [Apache Tomcat](#)
- [tomcat-dev](#)  
Development mailing list, including commit messages

#### Other Downloads

- [Tomcat Connectors](#)
- [Tomcat Native](#)
- [Taglibs](#)
- [Deployer](#)

#### Other Documentation

- [Tomcat Connectors](#)
- [mod\\_jk Documentation](#)
- [Tomcat Native](#)
- [Deployer](#)

#### Get Involved

- [Overview](#)
- [Source Repositories](#)
- [Mailing Lists](#)
- [Wiki](#)

#### Miscellaneous

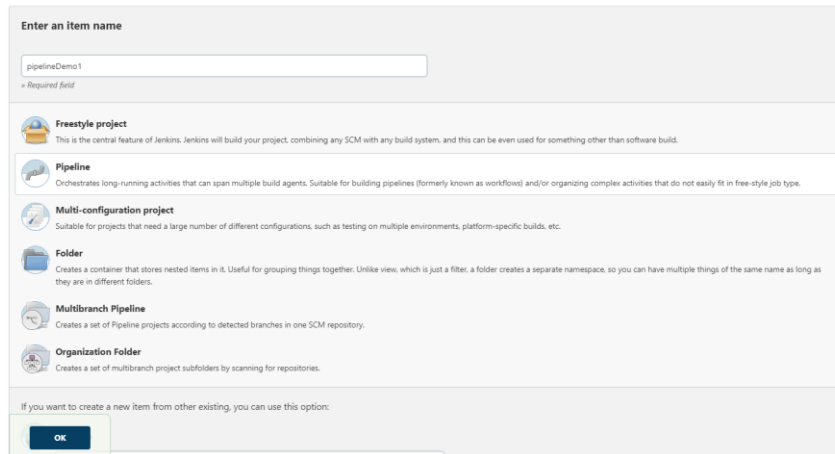
- [Contact](#)
- [Legal](#)
- [Sponsorship](#)
- [Thanks](#)

#### Apache Software Foundation

- [Who We Are](#)
- [Heritage](#)
- [Apache Home](#)
- [Resources](#)

Copyright ©1999-2022 Apache Software Foundation. All Rights Reserved

2. Go to Jenkins Dashboard and create a new pipeline project.



Enter an item name

pipelineDemo1

+ Required field

- Freestyle project**  
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.
- 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:

OK

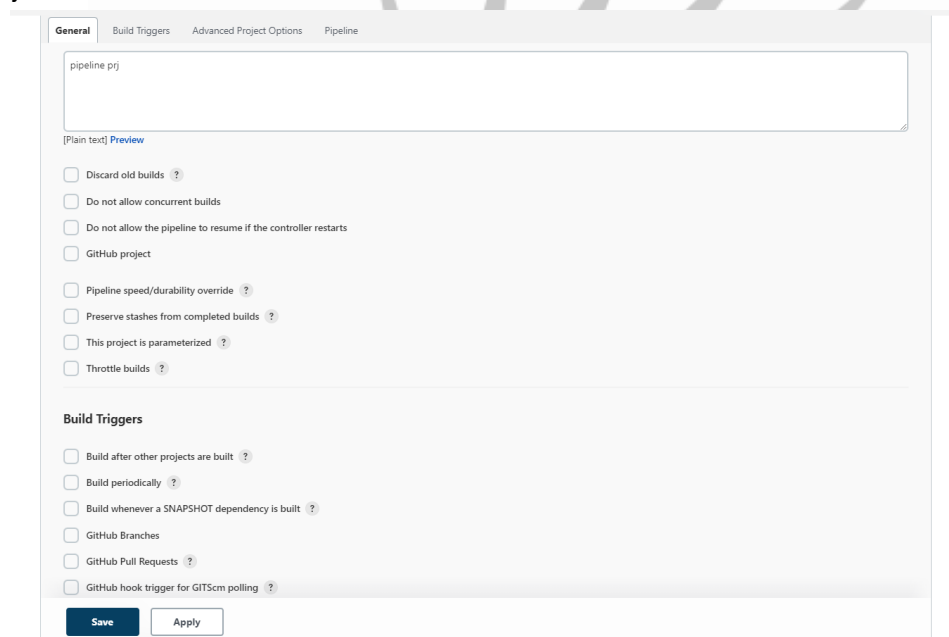
3. Go to Pipeline Section and choose Pipeline Script and write the following script here:

```
pipeline
{
  agent any
  tools
  {
    maven 'Maven 3.8.2'
  }
  stages
  {
    stage("Git Clone")
    {
      steps
      {
        git 'https://github.com/bushsk/hello-world.git'
      }
    }

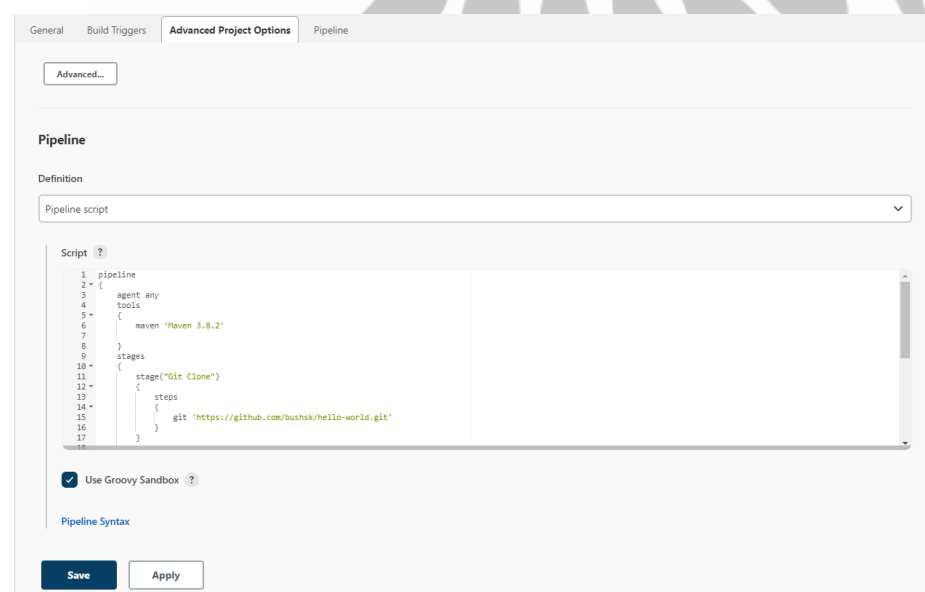
    stage("Build")
    {
      steps
      {
        bat 'mvn clean install'
      }
    }
    stage("Deploy")
    {
```

```
    steps
    {
        bat      'copy
C:\\ProgramData\\Jenkins\\\\.jenkins\\workspace\\Pipeline\\webapp\\target\\webapp.war
D:\\apache-tomcat-9.0.65\\webapps'
    }
}

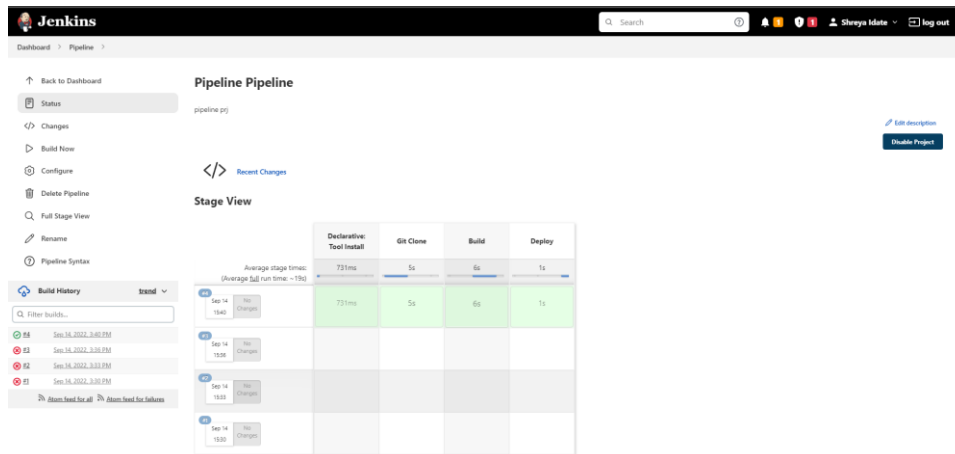
}
```



The screenshot shows the 'General' tab of the Jenkins Pipeline Configuration interface. At the top, there are tabs for 'General', 'Build Triggers', 'Advanced Project Options', and 'Pipeline'. The 'General' tab is active. It features a text area for the pipeline name, currently set to 'pipeline prj'. Below this is a 'Plain text' preview button. A list of checkboxes is present, including 'Discard old builds', 'Do not allow concurrent builds', 'Do not allow the pipeline to resume if the controller restarts', 'GitHub project', 'Pipeline speed/durability override', 'Preserve stashes from completed builds', 'This project is parameterized', and 'Throttle builds'. The 'Build Triggers' section includes checkboxes for 'Build after other projects are built', 'Build periodically', 'Build whenever a SNAPSHOT dependency is built', 'GitHub Branches', 'GitHub Pull Requests', and 'GitHub hook trigger for GITScm polling'. At the bottom, there are 'Save' and 'Apply' buttons.



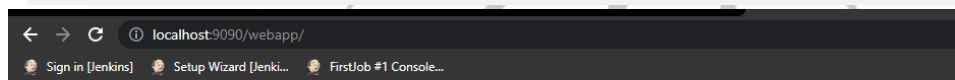
The screenshot shows the 'Advanced Project Options' tab of the Jenkins Pipeline Configuration interface. It features a 'Pipeline' section with a 'Definition' dropdown menu set to 'Pipeline script'. Below this is a 'Script' section with a text area containing a Groovy script. The script defines a pipeline with a stage named 'Git Clone' containing a step named 'git' that clones a repository. At the bottom, there is a checkbox for 'Use Groovy Sandbox' which is checked. Below the script area is a 'Pipeline Syntax' link. At the bottom, there are 'Save' and 'Apply' buttons.



4. Got to Build Now and verify if the pipeline build is successful. Go to <http://localhost:9090/webapp/> and verify if the home page of your application is up or not.

### Console Output

```
Started by user Shreya Idarte
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Git Clone)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] git
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/bushsk/hello-world.git
> git.exe init C:\ProgramData\Jenkins\jenkins\workspace\Pipeline # timeout=10
Fetching upstream changes from https://github.com/bushsk/hello-world.git
> git.exe --version # timeout=10
> git --version # "git version 2.35.1.windows.2"
> git.exe fetch --tags --force --progress -- https://github.com/bushsk/hello-world.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git.exe config remote.origin.url https://github.com/bushsk/hello-world.git # timeout=10
> git.exe config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
```



## Welcome to SIES Graduate School of Technology -DevOps Lab

Setup and deploy using jenkins for E1 TE IT batch

**CONCLUSION:** Successfully installed Jenkins and configured Jenkins with Maven/Ant/Gradle, GitHub and Python Plugins to set up a build Job.