

Week 9: Pipeline Creation using script

1. Evaluation of Jenkins pipeline.
 2. **WORKING ON BUILD TRIGGERS FOR LAST JENKINS PIPILINE**
 3. Hands-on practice on creation of scripted Jenkins pipeline.
 4. Take the screenshots for above task
-

 Jenkins / All / New Item

New Item

Enter an item name
Harshitha_Jenkins_Java_using_Script

Select an item type

 Freestyle project
Classic, general-purpose job type that checks out from up to one or more sources, runs a build, and sends notifications.

 Maven project
Build a maven project. Jenkins takes advantage of your POM file to determine what to build.

 Pipeline
Orchestrates long-running activities that can span multiple builds (e.g., parallel builds, workflows) and/or organizing complex activities that do not easily fit into a single build step.

 Multi-configuration project
Suitable for projects that need a large number of different configurations, such as platform-specific builds, etc.

 Folder
Creates a container that stores nested items in it. Useful for grouping related items together. A folder creates a separate namespace, so you can have multiple items with the same name in different folders.

OK

onfigure

General

General

Triggers

Pipeline

Advanced

Description

Creating java project pipeline using Script

Definition

Pipeline script

Script ?

```
1< pipeline {
2     agent any
3     tools{
4         maven 'MAVEN-HOME'
5     }
6     stages {
7         stage('git repo & clean') {
8             steps {
9                 //bat "rmdir /s /q mavenjava"
10                bat "git clone https://github.com/Harshitha-Macha/Harshitha_Jenkins_Maven.git"
11                bat "mvn clean -f Harshitha_Jenkins_Maven"
12            }
13        }
14        stage('install') {
15            steps {
```

 Use Groovy Sandbox ?

Pipeline Syntax

Advanced

Advanced ▾

Save

Apply

 Saved

Procedure

General :

Description :- provide the description of the project

Build triggers: here we can provide with build triggers of your choice

Advance project option:

Definition:

Choose pipeline script

CODE:

```
pipeline {  
    agent any  
    tools{  
        jdk 'JAVA_HOME'  
        maven 'MAVEN_HOME'  
    }  
    stages {  
        stage('git repo & clean') {  
            steps {  
                bat "rmdir /s /q Harshitha_Jenkins_Maven || exit 0"  
                bat           "git           clone           https://github.com/Harshitha-  
Macha/Harshitha_Jenkins_Maven.git"  
                bat "mvn clean -f Harshitha_Jenkins_Maven/pom.xml"  
            }  
        }  
        stage('install') {  
            steps {  
                bat "mvn install -f Harshitha_Jenkins_Maven/pom.xml"  
            }  
        }  
        stage('test') {  
            steps {  
                bat "mvn test -f Harshitha_Jenkins_Maven/pom.xml"  
            }  
        }  
        stage('package') {  
            steps {  
                bat "mvn package -f Harshitha_Jenkins_Maven/pom.xml"  
            }  
        }  
    }  
}
```

Apply and save

figure

Pipeline

General Define your Pipeline using Groovy directly or pull it from source control.

Triggers Definition

Pipeline Pipeline script

Advanced

Script ?

```
1 v pipeline {
2     agent any
3 v     tools{
4         jdk 'JAVA_HOME'
5         maven 'MAVEN_HOME'
6     }
7 v     stages {
8         stage('Clean Workspace') {
9             steps {
10                 deleteDir()
11             }
12         }
13         stage('git repo & clean') {
14             steps {
15                 bat "rmdir /s /q Harshitha_Jenkins_Maven || exit 0"
16             }
17         }
18     }
19 }
20 v     stage('install') {
21     steps {
22         bat "mvn install -f Harshitha_Jenkins_Maven/pom.xml"
23     }
24 }
25 v     stage('test') {
26     steps {
27         bat "mvn test -f Harshitha_Jenkins_Maven/pom.xml"
28     }
29 }
```

Pipeline

Definition

Pipeline Pipeline script

Script ?

```
15     bat "rmdir /s /q Harshitha_Jenkins_Maven || exit 0"
16     bat "git clone https://github.com/Harshitha-Macha/Harshitha_Jenkins_Maven.git"
17     bat "mvn clean -f Harshitha_Jenkins_Maven/pom.xml"
18 }
19 }
20 v     stage('install') {
21     steps {
22         bat "mvn install -f Harshitha_Jenkins_Maven/pom.xml"
23     }
24 }
25 v     stage('test') {
26     steps {
27         bat "mvn test -f Harshitha_Jenkins_Maven/pom.xml"
28     }
29 }
```

Harshitha_Jenkins_Java_using_Script / #6

✓ #6 (Oct 7, 2025, 10:47:07 AM)



Started by user Macha Harshitha

ion



This run spent:

- 11 ms waiting;
- 31 sec build duration;
- 31 sec total from scheduled to completion.



No changes.

Jenkins / Harshitha_Jenkins_Java_using_Script / #6

```
[INFO] Running KMIT.Harshitha_Jenkins_Maven.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed:
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.2:jar (default-jar) @ Harshitha_Jenkins_Maven ---
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time:  3.485 s
[INFO] Finished at: 2025-10-07T10:47:38+05:30
[INFO] -----
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

Build after other projects are built ?

Build periodically ?

Schedule ?

⚠ Do you really mean "every minute" when you say "***"? Perhaps you meant "H * * * *" to poll once per hour**

Would last have run at Tuesday, 7 October, 2025 at 12:11:00 pm India Standard Time; would next run at Tuesday, 7 October, 2025 at 12:12:00 pm India Standard

GitHub hook trigger for GITScm polling ?

Poll SCM ?

Save

Apply

<input checked="" type="checkbox"/>	#21 11:30 AM	▼
<input checked="" type="checkbox"/>	#20 11:29 AM	▼
<input checked="" type="checkbox"/>	#19 11:28 AM	▼
<input checked="" type="checkbox"/>	#18 11:27 AM	▼
<input checked="" type="checkbox"/>	#17 11:13 AM	▼
<input checked="" type="checkbox"/>	#16 11:12 AM	▼
<input checked="" type="checkbox"/>	#15 11:11 AM	▼
<input checked="" type="checkbox"/>	#14 11:10 AM	▼
<input checked="" type="checkbox"/>	#13 11:09 AM	▼
<input checked="" type="checkbox"/>	#12 11:08 AM	▼
<input checked="" type="checkbox"/>	#11 11:07 AM	▼
<input checked="" type="checkbox"/>	#10 11:06 AM	▼
<input checked="" type="checkbox"/>	#9 11:05 AM	▼
<input checked="" type="checkbox"/>	#8 11:04 AM	▼
<input checked="" type="checkbox"/>	#7 11:03 AM	▼

SBQ's:

1. Your manager asks you to clean the workspace before building — which stage in this pipeline takes care of it?

A) “Git Repo & Clean” stage handles workspace cleaning

bat "rmdir /s /q Harshitha_Jenkins_Maven || exit 0"

2. The GitHub repository link is missing in the script — where exactly do you provide it?

A) Inside the “Git Repo & Clean” stage, in the line:

bat "git clone https://github.com/Harshitha-Macha/Harshitha_Jenkins_Maven.git".

3. If Maven is not configured globally in Jenkins, which section of the pipeline will fail first?

A) Tools section will fail during initialization, tools { maven 'MAVEN_HOME' } depends on a globally configured Maven installation.

4. A teammate complains the pipeline is not creating .war files — which stage is responsible?

A) The Package stage is responsible for creating .jar or .war files using mvn package.

5. Your test cases failed, but the pipeline still continued — how will Jenkins behave in the test stage?

A) By default, Jenkins marks the Test stage as failed but continues to later stages unless error or failFast conditions are used.

6. Instead of running nightly builds, you want this pipeline to trigger only when GitHub changes occur — where will you configure it?

A) In the Build Triggers section of the project configuration, select

GitHub hook trigger for GITScm polling.

7. If you replace mvn clean with mvn compile, what difference will it make to the project build?

A) mvn clean deletes previous build artifacts, ensuring a fresh build.

mvn compile only compiles source code and doesn't clean or remove old files.

8. The project folder name is not mavenjava but studentapp — which parts of the script must you edit?

A) Every Maven command path and folder reference:

mvn clean -f studentapp/pom.xml, mvn install -f studentapp/pom.xml, etc.

9. If the install stage fails, will the test and package stages still run in this pipeline?

A) No. Jenkins Declarative Pipelines stop executing subsequent stages if any previous stage fails by default.

10. A student asks where to add deployment steps to Tomcat after packaging — which is the best place in this script?

A) Add a new “Deploy” stage after the Package stage

11. Why is tools { maven 'MAVEN-HOME' } used in this pipeline?

A) It tells Jenkins to automatically install and use the configured Maven tool from

Manage Jenkins -> Global Tool Configuration, ensuring Maven is available for build commands.

12. If GitHub credentials are private, how can you secure them in the git repo & clean stage?

A) Store credentials in Jenkins using *Credentials Manager*, then use:

```
withCredentials([usernamePassword(credentialsId: 'git-creds', usernameVariable: 'USER',  
passwordVariable: 'PASS')]) {
```

```
bat "git clone https://\$USER:\$PASS@github.com/username/repo.git"  
}
```

13. Which Jenkins plugin is needed to recognize the pipeline { ... } structure in this script?

A) Pipeline Plugin must be installed

14. In Windows, the script uses bat commands — what change would you make if Jenkins runs on Linux?

A) Replace all bat commands with sh commands.

15. How will you modify the pipeline to stop execution if the GitHub clone command fails?

Your project lead asks you to print a welcome message in Jenkins to confirm the pipeline is working — how would you script it?

A) Add a condition or use returnStatus

```
script {
```

```
    def status = bat(script: "git clone https://github.com/...git", returnStatus: true)  
  
    if (status != 0) {  
  
        error("Git clone failed — stopping pipeline.")  
  
    }  
  
}
```

16. A teammate forgot to pull the latest GitHub code before building; how can you fix this in your pipeline?

A) Add a simple echo step at the beginning

```
stage('Welcome') {  
  
    steps {  
  
        echo "Welcome to Jenkins Pipeline — Build Started Successfully!"  
  
    }  
  
}
```

17. Your Java file Hello.java must be compiled every time code is committed — how will you add this step?

A) Add a pull step before building:

```
bat "git -C Harshitha-Maven-JavaProject pull"
```

18. Tests should stop the pipeline if they fail — where will you put the mvn test command?

A) Add a Compile stage after cloning:

```
stage('Compile') {  
    steps {  
        bat "mvn compile -f project/pom.xml"  
    }  
}
```

19. Every evening at 6 PM your pipeline should run automatically — how can you set this in Jenkins?

A) Inside the Test stage, a separate stage so Jenkins can halt execution automatically if it fails.

20. You only want to package the project if compilation succeeds — how would you connect the stages?

A) Under *Build Triggers → Build periodically*, enter this H 18 * * *

21. Your professor wants a .jar file generated for submission — what pipeline stage will you add?

A) The Package stage handles .jar creation

22. A Git clone step is failing due to wrong credentials — how would you secure and use them in your script?

A) Store credentials in Jenkins Credentials Manager, then access them securely:

```
withCredentials([usernamePassword(credentialsId: 'git-creds', usernameVariable: 'USER',  
passwordVariable: 'PASS')]) {
```

```
    bat "git clone https://\$${USER}\$\${PASS}@github.com/username/repo.git"  
}
```

23. A teammate wants to see the build number inside console logs — how do you print it in the pipeline?

A) Use the Jenkins environment variable BUILD_NUMBER:

```
echo "Running Build Number: \${env.BUILD_NUMBER}"
```