

1. Click on New Item



+ New Item

👤 People

📄 Build History


⚙️ Manage Jenkins

📌 My Views


2. Type an Item Name and Create a Pipeline

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.

**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.

3. Enter the required details


General


Enabled 

Description

Testing Pipeline


[Plain text] [Preview](#)

- ☐ Discard old builds 
- ☐ Do not allow concurrent builds
- ☐ Do not allow the pipeline to resume if the controller restarts
- ☒ GitHub project

Project url <https://github.com/subrato3/netflix-clone.git>

4. Write the following script to run while building


Definition

Pipeline script 

Script

```
1 pipeline {  
2   agent any  
3  
4   stages {  
5     stage('Hello') {  
6       steps {  
7         echo 'Hello World'  
8       }  
9     }  
10  }  
11 }  
12
```

Hello World 

- ☒ Use Groovy Sandbox 

[Pipeline Syntax](#)

Save

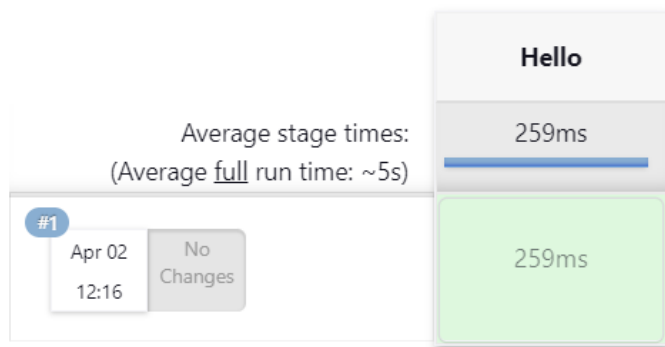
Apply

5. Apply then Save

- Status
- Changes
- Build Now
- Configure
- Delete Pipeline
- Full Stage View

6. Select Stage View to view the following

Stage View



Permalinks

7. Select console output to view the following:
Pipeline was set up and ran successfully.

- Status
- Changes
- Console Output
- View as plain text
- Edit Build Information
- Delete build '#1'
- Restart from Stage
- Replay
- Pipeline Steps
- Workspaces

Console Output

```
Started by user Subrato
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/First Pipeline
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Hello)
[Pipeline] echo
Hello World
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

8. Download tomcat on local machine and run the service before starting build job. Also configure the path on your local machine

```
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\Program Files\Java\jre1.8.0_351\bin
C:\Program Files\PowerShell\7\
C:\Program Files\Docker\Docker\resources\bin
C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin
```

```
C:\>cd Program Files\Apache Software Foundation\Tomcat 9.0\bin


C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin>startup
Using CATALINA_BASE:   "C:\Program Files\Apache Software Foundation\Tomcat 9.0"
Using CATALINA_HOME:   "C:\Program Files\Apache Software Foundation\Tomcat 9.0"
Using CATALINA_TMPDIR: "C:\Program Files\Apache Software Foundation\Tomcat 9.0\temp"
Using JRE_HOME:        "C:\Program Files\Java\jdk1.8.0_311"
Using CLASSPATH:       "C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin\bootstrap.jar;C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin\tomcat-juli.jar"
Using CATALINA_OPTS:   ""
C:\Program Files\Apache Software Foundation\Tomcat 9.0\bin>
```

9. On Jenkins on AWS, set up a new freestyle project

Enter an item name

sepm_5






» 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.

10. Write the necessary details and the following Build Steps. Make sure you write these on the Shell and not windows batch command. As you're operating on Ubuntu in AWS.

Configure

-  General
-  Source Code Management
-  Build Triggers
-  Build Environment
-  Build Steps

Build Steps

≡ **Execute Windows batch command** ?


Command


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


```
mvn package
mvn test
```


11. Add “Deploy to container” plugin


[Dashboard](#) > [Manage Jenkins](#) > [Plugins](#)

 Updates

 Available plugins

 Installed plugins

 Advanced settings

 Download progress


Download progress


Preparation

Ionicons API

Folders

- Checking internet connectivity
- Checking update center connectivity
- Success

 Success

 Success

12. Select Deploy war/ear to a container in Add post-build action

Aggregate downstream test results

Archive the artifacts

Build other projects

Publish JUnit test result report

Record fingerprints of files to track usage

Git Publisher

Deploy war/ear to a container

E-mail Notification

Editable Email Notification

Set GitHub commit status (universal)

Set build status on GitHub commit [deprecated]

Delete workspace when build is done

Add post-build action ▲

13. Write the following info/configurations in it. Select Tomcat 9.x remote

Deploy war/ear to a container

WAR/EAR files ?

**/*.war

Context path ?

sepm_5

Containers

Add Container ^

Filter

GlassFish 4.x

JBoss AS 3.x

JBoss AS 4.x

JBoss AS 5.x

JBoss AS 6.x

JBoss AS 7.x

Tomcat 4.x Remote

Tomcat 5.x Remote

Tomcat 6.x Remote

Tomcat 7.x Remote

Tomcat 8.x Remote

Tomcat 9.x Remote

REST API

14. Select Apply and then Save

Project sepm_5

SEPM 5- Build pipeline of jobs using maven in jenkins, create a pipeline to test and deploy an application over a server

Edit description

Disable Project

Permalinks

15. Build the project

Console Output

```
Started by user Subrato Tapaswi
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/MyFirstMavenJob
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/MyFirstMavenJob/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/subrato3/StudentDatabase.git # timeout=10
Fetching upstream changes from https://github.com/subrato3/StudentDatabase.git
> git --version # timeout=10
> git --version # 'git version 2.25.1'
```