

5. Dockerizing Jenkins Pipeline.

○ Pull an image from Registry

- Ensure that docker is installed in your pc
- Pull the maven image (run following command)
- `$docker pull maven`

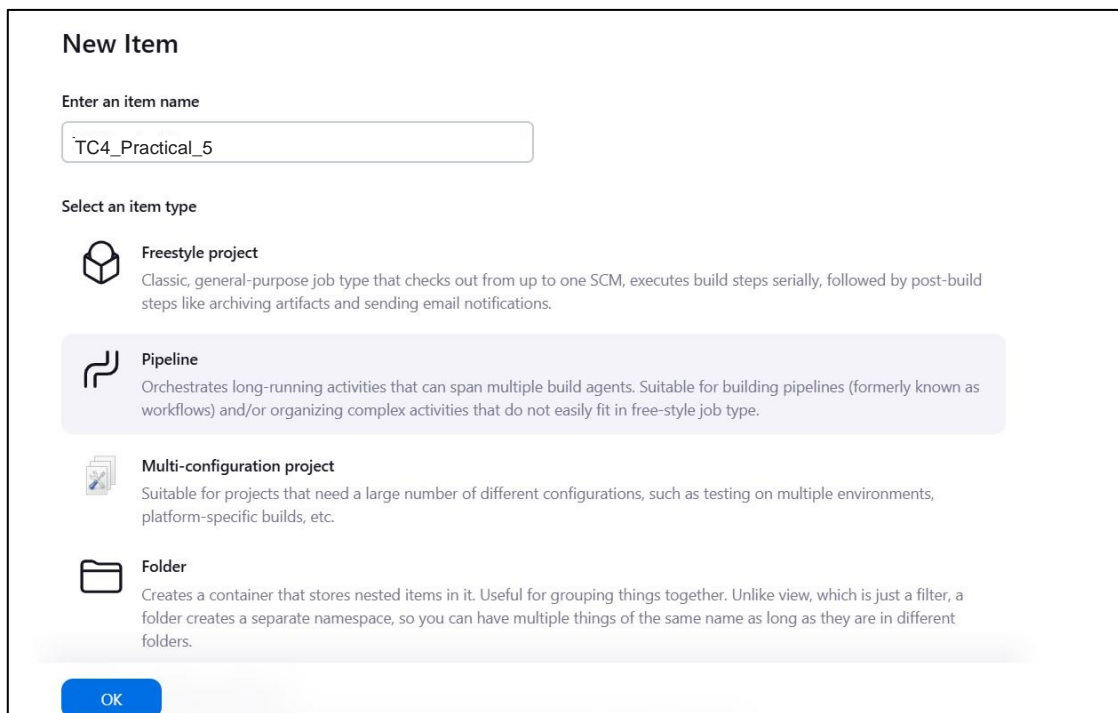
```
C:\Users\Administrator>docker pull maven
Using default tag: latest
latest: Pulling from library/maven
32b824d45c61: Pull complete
fe18bb7e114f: Pull complete
581ebfe08d3f: Pull complete
7c7bdd063feb: Pull complete
28f1e2918031: Pull complete
5d747676dacd: Pull complete
5f80e8f4bfde: Pull complete
e626418b2bad: Pull complete
4f4fb700ef54: Pull complete
2a669a7f0633: Pull complete
Digest: sha256:9e63e6c1ab3535c4afe34fd86ec5716719b43c3ba5bdccb03b9afad579110bde
Status: Downloaded newer image for maven:latest
docker.io/library/maven:latest
```

What's next:

View a summary of image vulnerabilities and recommendations → [docker scout quickview maven](#)

```
C:\Users\Administrator>|
```

- Open Jenkins Dashboard
- Click on "Dashboard"
- Click on "New Item"
- Give Project Name i.e. "TC4_Practical_5"
- Select "Pipeline" job type
- Click on "OK"







New Item

Enter an item name

TC4_Practical_5

Select an item type

-  **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.

OK

- Give few words description "Practical - 5"

General

Enabled 

Description

"Practical - 5"

Plain text [Preview](#)

- ☐ 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 ?

- Go to "Pipeline"

- There are two option (i) Pipeline script (ii) Pipeline script from SCM
- Select option one (i) Pipeline script
- Write the following script:

```
pipeline { agent any
  stages {
    stage('Build') {
      steps {
        bat 'docker run --rm --name my-maven-project -v
          "D:/Marwadi University/SEMESTER - 7/DEVOPS ESSENTIALS (
          01CE0717)/LabPractical/Practicals/Prac_2/firstmavenapp:/usr/src/mymaven" -w /usr/src/mymaven maven mvn clean install'
      }
    } stage('Test')
    { steps {
      bat 'mvn -version'
    }
  }
}
```

- Please change the path with your maven project folder.

Pipeline

Definition

Pipeline script

Script ?

```

1 pipeline {
2   agent any
3
4   stages {
5     stage('Build') {
6       steps {
7         bat 'docker run --rm --name my-maven-project -v "D:/Marwadi University/SEMESTER - 7/DEVOPS ESSENTIALS ( 01CE0717 )/Lab 1'
8       }
9     }
10    stage('Test') {
11      steps {
12        bat 'mvn -version'
13      }
14    }
15  }
16 }

```

try sample Pipeline...

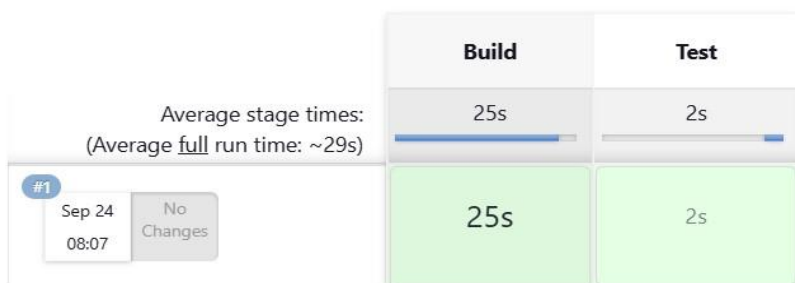
☒ Use Groovy Sandbox ?

- Click on **"Apply"** and then click on **"Save"**
- Click on **"Build Now"**

✓ TC4_Practical-5

"Practical - 5"

Stage View



Permalinks

- [Last build \(#1\), 1 min 37 sec ago](#)
- [Last stable build \(#1\), 1 min 37 sec ago](#)
- [Last successful build \(#1\), 1 min 37 sec ago](#)
- [Last completed build \(#1\), 1 min 37 sec ago](#)

✓ Console Output

[Download](#)[Copy](#)[View as plain text](#)

```
Started by user Jay dalsaniya
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in C:\ProgramData\Jenkins\.jenkins\workspace\TC5_Practical-5
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Build)
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\TC5_Practical-5>docker run --rm --name my-maven-project -v "D:/Marwadi University/SEMESTER - 7/DEVOPS ESSENTIALS ( 01CE0717 )/Lab Practical/Practicals/Prac_2/firstmavenapp:/usr/src/mymaven" -w /usr/src/mymaven maven mvn clean install
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.7TC5.app:firstmavenapp >-----
[INFO] Building firstmavenapp 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.1.0/maven-clean-plugin-3.1.0.pom
Progress (1): 1.4/5.2 kB
Progress (1): 2.8/5.2 kB
Progress (1): 4.1/5.2 kB
Progress (1): 5.2 kB

Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-clean-plugin/3.1.0/maven-clean-plugin-3.1.0.pom (5.2 kB
```

```
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/3.0.15/plexus-utils-3.0.15.jar (239 kB at 327 kB/s)
[INFO] Installing /usr/src/mymaven/target/firstmavenapp-1.0-SNAPSHOT.jar to /root/.m2/repository/com/7TC5/app/firstmavenapp/1.0-SNAPSHOT/firstmavenapp-1.0-SNAPSHOT.jar
[INFO] Installing /usr/src/mymaven/pom.xml to /root/.m2/repository/com/7TC5/app/firstmavenapp/1.0-SNAPSHOT/firstmavenapp-1.0-SNAPSHOT.pom
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 22.525 s
[INFO] Finished at: 2024-09-24T02:37:40Z
[INFO] -----
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { (Test)
[Pipeline] bat

C:\ProgramData\Jenkins\.jenkins\workspace\TC5_Practical-5>mvn -version
Apache Maven 3.9.8 (36645f6c9b5079805ea5009217e36f2cffd34256)
Maven home: D:\Marwadi University\SEMESTER - 7\DEVOPS ESSENTIALS ( 01CE0717 )\Lab Practical\Practicals\apache-maven-3.9.8
Java version: 21.0.1, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk-21
Default locale: en_IN, platform encoding: UTF-8
OS name: "windows 11", version: "10.0", arch: "amd64", family: "windows"
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```