# TASK 8: Run a Simple Java Maven Build Job in Jenkins

**Objective:**

➢ Build and package a basic Java app using Maven in Jenkins running inside Docker.

**Tools Required:**

➢ Ec2 instance (ubuntu)
➢ Docker
➢ Pull the Jenkins from docker hub

**Install docker**

- sudo apt update

- sudo apt install -y docker.io -y

- sudo systemctl enable docker

- sudo systemctl start docker

- docker –version

- docker pull jenkins/jenkins

## Start Jenkins using Docker

docker run -d --name jenkins \

 -p 8080:8080 -p 50000:50000 \

 -v jenkins_home:/var/jenkins_home \

 jenkins/jenkins:lts

   o -d = detached
   o Jenkins UI will be at http://localhost:8080

```
root@ip-172-31-0-70:~# docker ps
CONTAINER ID   IMAGE               COMMAND              CREATED          STATUS          PORTS
                                   NAMES
39a577ca94a9   jenkins/jenkins:lts  "/usr/bin/tini -- /u…"  About an hour ago  Up About an hour  0.0.0.0:8080->8080/tcp, :::8080->8080/tcp, 0.0.0.0
:50000->50000/tcp, :::50000->50000/tcp   jenkins
```

**Unlock Jenkins**

➢ Check the admin password:
   o docker exec -it --user root <container_id_or_name> /bin/bash
   o cat /var/jenkins_home/secrets/initialAdminPassword
   o Copy that password → Open browser → http://localhost:8080 → paste it
   o Click: **Install Suggested Plugins**
      ▪ Stage view
      ▪ Maven integration

- Github integration

## Create Java Project in Docker

- o docker exec -it --user root <container_id_or_name> /bin/bash
- o apt update -y
- o git clone <url> ---- this is your wish I used a github for mvn build
- o mkdir -p /var/jenkins_home/hello-java-maven/src/main/java
- o cd /var/jenkins_home/hello-java-maven/src/main/java


- o Create HelloWorld.java:

```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, Jenkins + Maven!");
    }
}
```

- o Create pom.xml in /var/jenkins_home/hello-java-maven/:

```xml
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>hello</artifactId>
  <version>1.0</version>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <source>1.8</source>
          <target>1.8</target>
```

```
        </configuration>
    </plugin>
  </plugins>
 </build>
</project>
```

➢ I send code into git hub repo

Create Freestyle Jenkins Job

| S | W | Name ↓ | Last Success | Last Failure | Last Duration | |
|---|---|---|---|---|---|---|
| ✓ | ☁ | hello-java-maven | 36 min  #4 | 39 min  #3 | 9.8 sec | ▷ |

1. Go manage Jenkins----> tools----> search maven installations---> add maven--->version 3.8.6 ----> save and apply
2. Jenkins UI → **New Item** → Name: hello-java-maven → Freestyle Project
3. Go to configuration----> source code management -----> code is present in locally choice **none or** for **git** means give <url>-----> branches to build **main**

**Source Code Management**

Connect and manage your code repository to automatically pull the latest code for your builds.

○ None

● Git  ?
    Repositories  ?

    Repository URL  ?
    ┌─────────────────────────────────────────────────────────────── ✕ ┐
    │ https://github.com/Ramakrishnaragi/TASK-8-Run-a-Simple-Java-Maven-Build-Job-in-Jer │
    └──────────────────────────────────────────────────────────────────┘

    Credentials  ?
    ┌─────────────────────────────────────────────────────────────── ∨ ┐
    │ - none -                                                          │
    └──────────────────────────────────────────────────────────────────┘

    + Add

    Advanced ∨

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

*/main

Add Branch

Repository browser ?

(Auto)

Additional Behaviours

Add ⌄

4.  Search the Build steps----> give the maven version---> **goals** (clean package) ------>add advanced--------> In **POM** give the path of pom.xml file other show an error for build time ---> save and apply

**Build Steps**

Automate your build process with ordered tasks like code compilation, testing, and deployment.

≡   **Invoke top-level Maven targets** ?                                    ✕

Maven Version

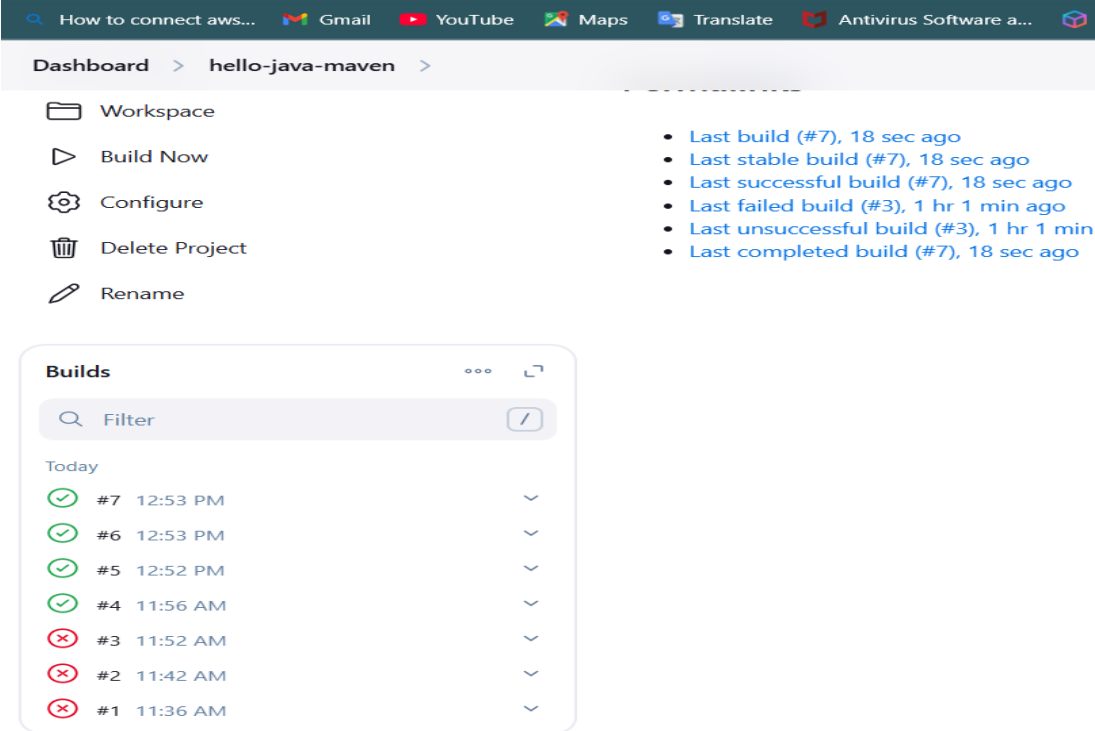Maven                                                                    ⌄

Goals

clean package                                                      ⌄

Advanced ⌃      ✎ Edited

POM ?

hello-java-maven/pom.xml

Properties ?

5. Execute the **build now**

Workspace

Build Now

Configure

Delete Project

Rename

- Last build (#7), 18 sec ago
- Last stable build (#7), 18 sec ago
- Last successful build (#7), 18 sec ago
- Last failed build (#3), 1 hr 1 min ago
- Last unsuccessful build (#3), 1 hr 1 min
- Last completed build (#7), 18 sec ago
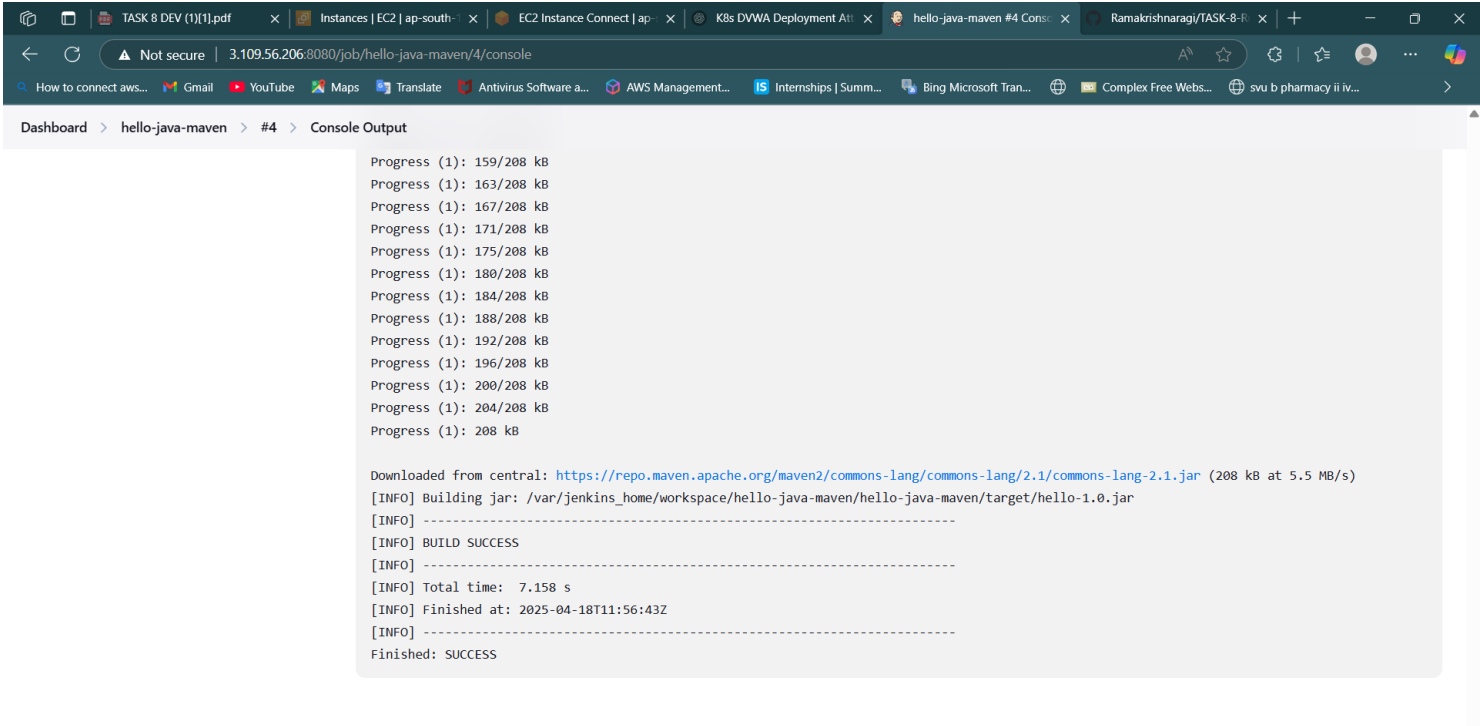
**Builds**                                    ○○○

🔍 Filter                                        /

Today

✓  #7  12:53 PM                                ⌄
✓  #6  12:53 PM                                ⌄
✓  #5  12:52 PM                                ⌄
✓  #4  11:56 AM                                ⌄
✕  #3  11:52 AM                                ⌄
✕  #2  11:42 AM                                ⌄
✕  #1  11:36 AM                                ⌄



```
Progress (1): 159/208 kB
Progress (1): 163/208 kB
Progress (1): 167/208 kB
Progress (1): 171/208 kB
Progress (1): 175/208 kB
Progress (1): 180/208 kB
Progress (1): 184/208 kB
Progress (1): 188/208 kB
Progress (1): 192/208 kB
Progress (1): 196/208 kB
Progress (1): 200/208 kB
Progress (1): 204/208 kB
Progress (1): 208 kB

Downloaded from central: https://repo.maven.apache.org/maven2/commons-lang/commons-lang/2.1/commons-lang-2.1.jar (208 kB at 5.5 MB/s)
[INFO] Building jar: /var/jenkins_home/workspace/hello-java-maven/hello-java-maven/target/hello-1.0.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  7.158 s
[INFO] Finished at: 2025-04-18T11:56:43Z
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS
```
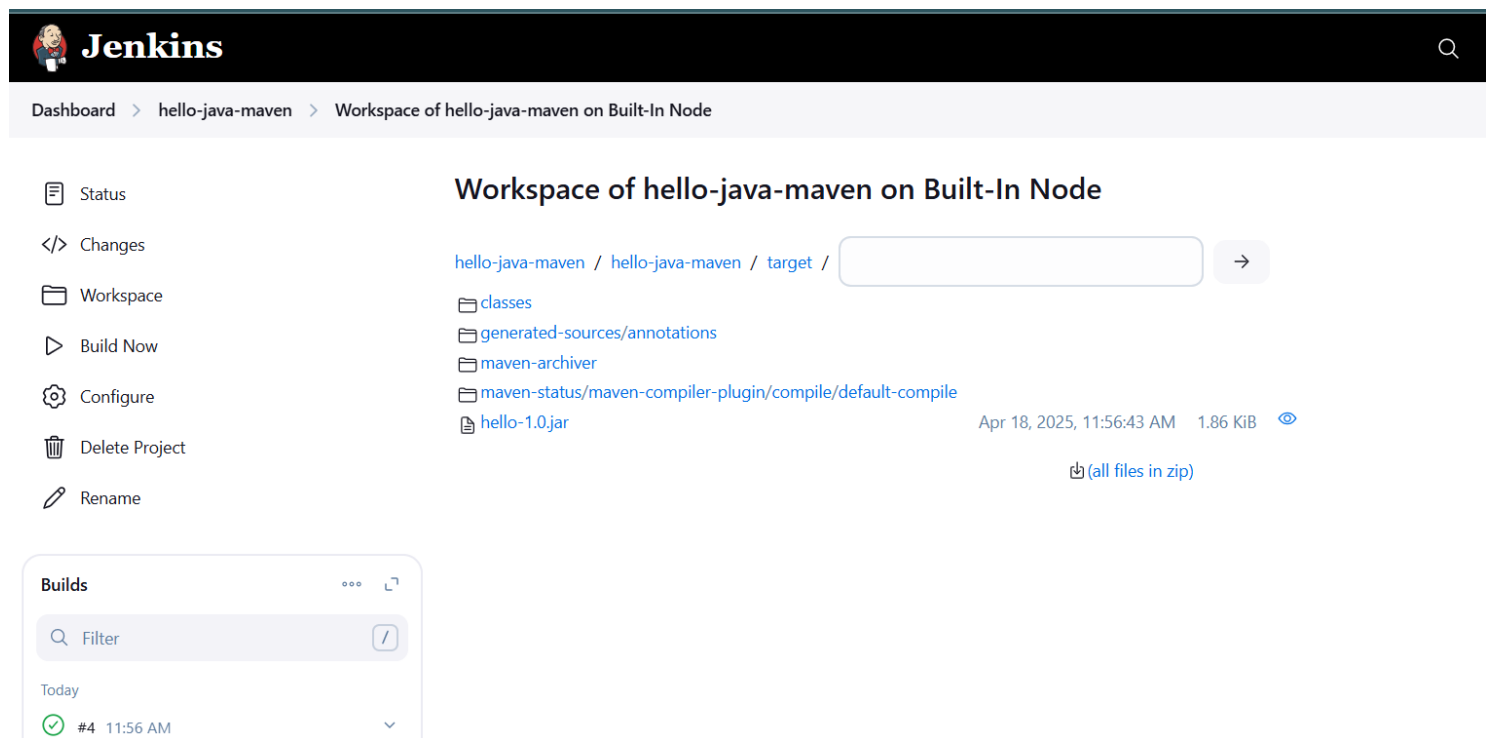
6. Finally check the **target file and .war or jar file**

Go to job----> workspace---> check hello-java-maven----> target file
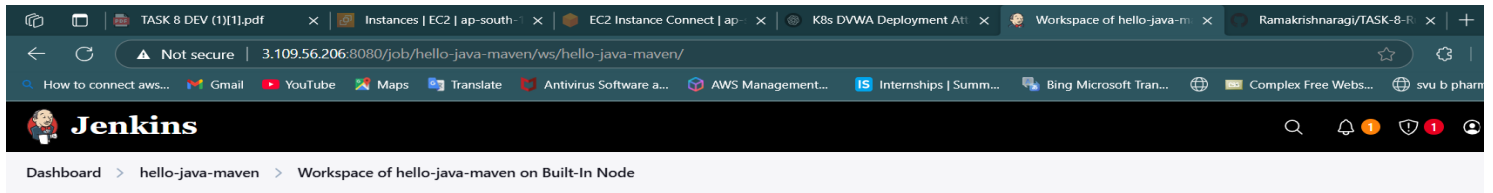


7. Finally check the .jar file inside the target dir



THANK YOU