

## 1. What is Maven? (Real-World Perspective)

**Definition:** Maven is a build automation and dependency management tool primarily used for Java projects.

### Real-World Use Case:

You're working on a **Banking Application** where multiple developers are contributing. Each module (like payments, loans, customer portal) needs specific Java libraries (e.g., Spring, Hibernate). Manually managing these libraries and building the application is error-prone.

### Maven helps:

- Automatically downloads required libraries (**dependencies**) from a central location.
- Builds project consistently across environments.
- Packages the final application into **.jar**, **.war**, or **.ear**.

## 2. Maven Outputs (Artifacts)

### Real-World Artifacts:

- **.jar** → For libraries/utilities (e.g., currency converter module)
- **.war** → For deploying web apps to a server (e.g., customer web portal)
- **.ear** → For large-scale enterprise apps containing multiple modules (e.g., ERP)

## 3. Repository Types in Maven (Explained with Diagram)

From your slide:

User's Machine → Local Repository (1st preference)  
→ Remote Repository (Organization network)  
→ Central Repository (Internet - Maven Central)

### Real-World Scenario:

Imagine you're at **TCS** working on a secured internal HR system. Due to firewall restrictions:

- Maven first checks if the dependency is in the **local repo** (`~/ .m2` folder).
- If not found, it checks the **internal remote repo** (e.g., Nexus/Artifactory).
- If not found even there, and if internet is allowed, it fetches from **Maven Central**.

## 4. Key Maven Commands and Usage

### Compiling & Packaging Java Project

```
mvn clean install
```

#### Result:

- Cleans previous builds.
- Compiles Java files.
- Packages the `.war` or `.jar`.
- Installs it into the **local repo** for reuse.

#### Real-World Use:

As a DevOps Engineer, you'd automate the build in Jenkins/GitHub Actions:

```
mvn test  
mvn package
```

This ensures the code is built and tested before deployment.

## 5. Automation Using Chocolatey + Maven (From Notes)

#### Real-World Scenario:

You're onboarding a new Windows developer. Instead of manual setup:

```
choco install maven  
choco install java8  
choco install git
```

Now they're ready to clone the repo and build with Maven.

## 6. Git + Maven Workflow (From Notes)

Steps:

1. `git clone <project-url>`
2. `cd <project>`
3. `mvn clean package`
4. Deploy the `.war` or `.jar` to test/production server.

Real-World Workflow in a CI/CD Pipeline:

1. **Pull** code from GitHub.
2. **Build** using Maven.
3. **Run Tests**.
4. **Generate Artifact**.
5. **Deploy** to a server using tools like Ansible, Docker, or Helm.

## 7. Dependency Resolution in Maven

Command: `mvn package`

Real-World Flow:

1. Checks for dependencies in `pom.xml`.
2. Pulls from:
  - o **Local** (if already downloaded)
  - o **Remote** (e.g., internal Nexus)

- **Central** (if allowed)

## 8. Maven Install Command

**Command:** `mvn install`

**Purpose:** Installs the compiled `.jar` into the **local repository** (`~/.m2`).

**Real-World Use:**

You created a custom Java library used by other internal projects. After `mvn install`, teammates can use it as a dependency in their `pom.xml`.

## Summary Table

Concept	Real-World Example
<code>mvn package</code>	Build a deployable <code>.war</code> file for web app
Local Repo	Developer's <code>.m2</code> folder
Remote Repo	Nexus/Artifactory inside organization
Central Repo	Maven Central (Internet)
Chocolatey	Simplifies setup on Windows machines
Artifact	Final build ( <code>.jar</code> , <code>.war</code> , <code>.ear</code> ) deployed to test/prod
Automation	CI/CD pipelines: Git → Build → Test → Deploy

Let me know if you want this as a formatted DOCX file or need CI/CD flow with Maven + Jenkins or GitHub Actions.

```
PS C:\Users\Administrator> choco --version
2.4.3
PS C:\Users\Administrator> java -version
openjdk version "1.8.0_442"
OpenJDK Runtime Environment (Temurin)(build 1.8.0_442-b06)
OpenJDK 64-Bit Server VM (Temurin)(build 25.442-b06, mixed mode)
PS C:\Users\Administrator> git --version
git : The term 'git' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or if a path was included, verify
that the path is correct and try again.
At line:1 char:1
+ git --version
+ ~~~
+ CategoryInfo          : ObjectNotFound: (git:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\Administrator> choco install git
Chocolatey v2.4.3
Installing the following packages:
git
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading chocolatey-compatibility.extension 1.0.0... 100%

chocolatey-compatibility.extension v1.0.0 [Approved]
chocolatey-compatibility.extension package files install completed. Performing other installation steps.
Installed/updated chocolatey-compatibility extensions.
The install of chocolatey-compatibility.extension was successful.
Deployed to 'C:\ProgramData\chocolatey\extensions\chocolatey-compatibility'
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading chocolatey-core.extension 1.4.0... 100%

chocolatey-core.extension v1.4.0 [Approved]
chocolatey-core.extension package files install completed. Performing other installation steps.
Installed/updated chocolatey-core extensions.
The install of chocolatey-core.extension was successful.
Deployed to 'C:\ProgramData\chocolatey\extensions\chocolatey-core'
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading git.install 2.49.0... 100%

git.install v2.49.0 [Approved]
git.install package files install completed. Performing other installation steps.
The package git.install wants to run 'chocolateyInstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N]o/[P]rint): Y

Using Git LFS
Installing 64-bit git.install...
Chocolatey v2.4.3
```

```
PS C:\Users\Administrator> choco install maven
Chocolatey v2.4.3
Installing the following packages:
maven
By installing, you accept licenses for the packages.
Downloading package from source 'https://community.chocolatey.org/api/v2/'
Progress: Downloading maven 3.9.9... 100%

maven v3.9.9 [Approved]
maven package files install completed. Performing other installation steps.
The package maven wants to run 'chocolateyinstall.ps1'.
Note: If you don't run this script, the installation will fail.
Note: To confirm automatically next time, use '-y' or consider:
choco feature enable -n allowGlobalConfirmation
Do you want to run the script?([Y]es/[A]ll - yes to all/[N)o/[P]rint): Y

C:\Users\Administrator\,m2
PATH environment variable does not have C:\ProgramData\chocolatey\lib\maven\apache-maven-3.9.9\bin in it. Adding...
Environment Vars (like PATH) have changed. Close/reopen your shell to
see the changes (or in powershell/cmd.exe just type `refreshenv`).
The install of maven was successful.
Deployed to 'C:\ProgramData\chocolatey\lib\maven\apache-maven-3.9.9'

Chocolatey installed 1/1 packages.
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).
PS C:\Users\Administrator> choco --version
2.4.3
```

Repo Link:

<https://github.com/maha4dsvc/mahalogin>

<https://github.com/AbdulRehaman082493/mahalogin>

[mahalogin](#)

Public

forked from [maha4dsvc/mahalogin](https://github.com/maha4dsvc/mahalogin)

```
Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project
$ pwd
/c/Users/Administrator/Desktop/Project

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project
$ git --version
git version 2.49.0.windows.1

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project
$ java --version
openjdk version "1.8.0_442"
OpenJDK Runtime Environment (Temurin)(build 1.8.0_442-b06)
OpenJDK 64-Bit Server VM (Temurin)(build 25.442-b06, mixed mode)

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project
$ mvn --version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcd97d260186937)
Maven home: C:\ProgramData\chocolatey\lib\maven\apache-maven-3.9.9
Java version: 1.8.0_442, vendor: Temurin, runtime: C:\Program Files\Eclipse Adoptium\jdk-8.0.442.6-hotspot\jre
Default locale: en_US, platform encoding: Cp1252
OS name: "windows server 2022", version: "10.0", arch: "amd64", family: "windows"

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project
$ git clone https://github.com/AbdulRehaman082493/mahalogin.git
Cloning into 'mahalogin'...
remote: Enumerating objects: 599, done.
remote: Counting objects: 100% (209/209), done.
remote: Compressing objects: 100% (138/138), done.
remote: Total 599 (delta 144), reused 67 (delta 66), pack-reused 390 (from 2)
Receiving objects: 100% (599/599), 33.42 MiB | 23.26 MiB/s, done.
Resolving deltas: 100% (199/199), done.

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project
$ l
bash: l: command not found

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project
$ ls
mahalogin/

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project
$ cd maha*
Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ ls
Dockerfile11 LICENSE README.md appgatewayurl.ps1 azure-pipelines.yml mahalLogin.war pom.xml src/ target/ testjava

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] _____< com.maha:mahalogin >_____
[INFO] Building Account Registration Webapp 2.0
[INFO]   From pom.xml
[INFO] _____[ war ]_____
[INFO]
[INFO] BUILD FAILURE
[INFO]
```

## What It Does

This Maven command performs two major actions:

1. **clean**: Deletes the `target/` directory to remove any previous build artifacts.
2. **package**: Compiles the code, runs tests, and packages it into a deployable format (`.jar`, `.war`, or `.ear`) as defined in the `pom.xml`.

## Real Folder Structure Before & After

**Before:**

```
/myproject
  └── src/
  └── pom.xml
  └── target/ ← might contain old builds
```

After Running:

```
mvn clean package
```

After:

```
/myproject
  └── src/
  └── pom.xml
  └── target/
    └── classes/
    └── myproject-1.0.0.jar / .war / .ear
    └── test-results/
```

## Real-World Example

You're building a **Java web application** for a college management system.

- Source code is in `/src/main/java`
- `pom.xml` defines dependencies and build type (`war`)

You run:

```
mvn clean package
```

## Result:

- Deletes any previously compiled classes or build outputs.
- Recompiles the code.
- Packages it into `target/college-portal-1.0.war`.

You can now deploy this `.war` to **Apache Tomcat**, **WildFly**, or use it in a CI/CD pipeline.

## When to Use

**Use Case**      Run `mvn clean package?`

You want to generate a fresh  
`.jar/.war`       Yes

You changed source  
code       Yes

You want to deploy to staging/production       Yes

You just want to run  
tests       Use `mvn test`

You want to skip tests       Add  
`-DskipTests`

`mvn clean package -DskipTests`

```
Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ ls -al
total 17073
drwxr-xr-x 1 Administrator 197121      0 Apr 11 18:59 .
drwxr-xr-x 1 Administrator 197121      0 Apr 11 18:51 ..
drwxr-xr-x 1 Administrator 197121      0 Apr 11 18:51 .git/
-rw-r--r-- 1 Administrator 197121      2 Apr 11 18:51 Dockerfile11
-rw-r--r-- 1 Administrator 197121     1100 Apr 11 18:51 LICENSE
-rw-r--r-- 1 Administrator 197121      65 Apr 11 18:51 README.md
-rw-r--r-- 1 Administrator 197121     484 Apr 11 18:51 appgatewayurl.ps1
-rw-r--r-- 1 Administrator 197121     428 Apr 11 18:51 azure-pipelines.yml
-rw-r--r-- 1 Administrator 197121 17452640 Apr 11 18:51 mahaLogin.war
-rw-r--r-- 1 Administrator 197121    4456 Apr 11 18:51 pom.xml
drwxr-xr-x 1 Administrator 197121      0 Apr 11 18:51 src/
drwxr-xr-x 1 Administrator 197121      0 Apr 11 19:00 target/
-rw-r--r-- 1 Administrator 197121     21 Apr 11 18:51 test.java

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ mvn clean package
[INFO] Scanning for projects...
[INFO]
[INFO] --- < com.maha:mahaLogin > ---
[INFO] Building Account Registration Webapp 2.0
[INFO]   from pom.xml
[INFO]   [ war ]
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ mahaLogin ---
[INFO] Deleting C:\Users\Administrator\Desktop\Project\mahalogin\target
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ mahaLogin ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 4 resources from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ mahaLogin ---
[INFO] Recompiling the module because of changed source code.
[WARNING] File encoding has not been set, using platform encoding windows-1252, i.e. build is platform dependent!
[INFO] Compiling 11 source files with javac [debug target 1.7] to target\classes
[WARNING] bootstrap class path not set in conjunction with -source 1.7
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ mahaLogin ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\Administrator\Desktop\Project\mahalogin\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ mahaLogin ---
[INFO] No sources to compile
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ mahaLogin ---
[INFO] No tests to run.
[INFO]
[INFO] --- war:3.3.1:war (default-war) @ mahaLogin ---
[INFO] Packaging webapp
[INFO] Assembling webapp [mahaLogin] in [C:\Users\Administrator\Desktop\Project\mahalogin\target\mahaLogin-2.0]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\Users\Administrator\Desktop\Project\mahalogin\src\main\webapp]
[INFO] Building war: C:\Users\Administrator\Desktop\Project\mahalogin\target\mahaLogin-2.0.war
[INFO]
[INFO] BUILD SUCCESS
[INFO]
```

```
Administrator@EC2AMAZ-6T5NOVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ java -version
openjdk version "1.8.0_442"
OpenJDK Runtime Environment (Temurin)(build 1.8.0_442-b06)
OpenJDK 64-Bit Server VM (Temurin)(build 25.442-b06, mixed mode)

Administrator@EC2AMAZ-6T5NOVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ git --version
git version 2.49.0.windows.1

Administrator@EC2AMAZ-6T5NOVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ mvn --version
Apache Maven 3.9.9 (8e857a9e76f7d015ee5ec7bfcd97d260186937)
Maven home: C:\ProgramData\chocolatey\lib\maven\apache-maven-3.9.9
Java version: 1.8.0_442, vendor: Temurin, runtime: C:\Program Files\Eclipse Adoptium\jdk-8.0.442.6-hotspot\jre
Default locale: en_US, platform encoding: Cp1252
OS name: "windows server 2022", version: "10.0", arch: "amd64", family: "windows"
```

```
Administrator@EC2AMAZ-6T5NOVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ choco -version
Parsing -version resulted in error (converted to warning):
Cannot bundle unregistered option '-e'.
This is a listing of all of the different things you can pass to choco.
```

## Options and Switches

-v, --version  
Version - Prints out the Chocolatey version.

Command	Description	Real-World Use
<code>mvn clean</code>	Deletes the <code>target/</code> directory (old builds)	Clean slate before new build
<code>mvn compile</code>	Compiles the Java source code	Checks for compilation errors
<code>mvn test</code>	Runs unit tests using JUnit/TestNG	Validates code logic
<code>mvn package</code>	Creates <code>.jar</code> , <code>.war</code> , or <code>.ear</code> in <code>target/</code>	Generates deployable artifact
<code>mvn install</code>	Installs the artifact into the <b>local repo</b> ( <code>~/.m2</code> )	Share artifact with other local projects
<code>mvn deploy</code>	Pushes the built artifact to a <b>remote repo</b> (e.g., Nexus, Artifactory)	Share artifact with team/org
<code>mvn site</code>	Generates documentation like reports, test coverage	For project analysis

## Run a Specific Phase

Maven uses a **build lifecycle**. These phases run in order:

```
mvn validate
mvn compile
mvn test
mvn package
mvn verify
mvn install
```

```
mvn deploy
```

You can run any one, and it executes all the ones before it.

Run with Flags/Options

Command	Description
<code>mvn package -DskipTests</code>	Builds artifact but skips unit tests
<code>mvn clean install -U</code>	Forces update of snapshots from repo
<code>mvn dependency:tree</code>	Shows all project dependencies (like a tree)
<code>mvn help:effective-pom</code>	Displays final merged <code>pom.xml</code> with inherited values

## Archetype (Project Creation)

Command	Description
<code>mvn archetype:generate</code>	Starts interactive project generator
<code>mvn archetype:generate -DgroupId=com.myapp -DartifactId=webapp -DarchetypeArtifactId=maven-archetype-webapp -DinteractiveMode=false</code>	Creates a Maven web app project non-interactively

Examples for DevOps/CI

```
# Build and test without tests  
mvn clean install -DskipTests
```

```
# Generate WAR and push to remote repo  
mvn clean deploy
```

```
# Display dependency hierarchy  
mvn dependency:tree
```

```
# Run a specific test class  
mvn -Dtest=MyTestClass test
```

## Common Maven Lifecycle

```
mvn clean  
mvn compile  
mvn test  
mvn package  
mvn install  
mvn deploy
```

```
Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)  
$ mvn install  
bash.exe: warning: could not find /tmp, please create!  
[INFO] Scanning for projects...  
[INFO] [ com.maha:mahaLogin > ]  
[INFO] Building Account Registration Webapp 2.0  
[INFO]   from pom.xml  
[INFO] [ war ]  
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.pom  
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.pom (6.5 kB at 7.4 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/42/maven-plugins-42.pom  
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/42/maven-plugins-42.pom (7.7 kB at 240 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven-parent/42/maven-parent-42.pom  
Downloaded From central: https://repo.maven.apache.org/maven-parent/42/maven-parent-42.pom (50 kB at 459 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/32/apache-32.pom  
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/apache/32/apache-32.pom (24 kB at 504 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.jar  
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.jar (32 kB at 540 kB/s)  
[INFO] [resources:3.3.1:resources (default-resources) @ mahaLogin ]  
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!  
[INFO] Copying 4 resources from src\main\resources to target\classes  
[INFO] [compiler:3.13.0:compile (default-compile) @ mahaLogin ]  
[INFO] Nothing to compile - all classes are up to date.  
[INFO] [resources:3.3.1:testResources (default-testResources) @ mahaLogin ]  
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!  
[INFO] skip non existing resourceDirectory C:\Users\Administrator\Desktop\Project\mahaLogin\src\test\resources  
[INFO] [compiler:3.13.0:testCompile (default-testCompile) @ mahaLogin ]  
[INFO] No sources to compile  
[INFO] [surefire:3.2.5:test (default-test) @ mahaLogin ]  
[INFO] No tests to run.  
[INFO] [war:3.3.1:war (default-war) @ mahaLogin ]  
[INFO] Packaging webapp  
[INFO] Assembling webapp [mahaLogin] in [C:\Users\Administrator\Desktop\Project\mahaLogin\target\mahaLogin-2.0]  
[INFO] Processing war project  
[INFO] Copying webapp resources [C:\Users\Administrator\Desktop\Project\mahaLogin\src\main\webapp]  
[INFO] Building war: C:\Users\Administrator\Desktop\Project\mahaLogin\target\mahaLogin-2.0.war  
[INFO] [install:3.1.2:install (default-install) @ mahaLogin ]  
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-util/1.9.18/maven-resolver-util-1.9.18.pom  
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-util/1.9.18/maven-resolver-util-1.9.18.pom (2.9 kB at 89 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver/1.9.18/maven-resolver-1.9.18.pom  
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver/1.9.18/maven-resolver-1.9.18.pom (22 kB at 1.4 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-api/1.9.18/maven-resolver-api-1.9.18.pom  
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-api/1.9.18/maven-resolver-api-1.9.18.pom (2.7 kB at 86 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-4.0.1.pom  
Downloaded From central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-4.0.1.pom (7.6 kB at 522 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-util/1.9.18/maven-resolver-util-1.9.18.jar  
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-util/1.9.18/maven-resolver-util-1.9.18.jar (196 kB at 3.1 kB/s)  
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-4.0.1.jar  
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-api/1.9.18/maven-resolver-api-1.9.18.jar  
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-api/1.9.18/maven-resolver-api-1.9.18.jar (157 kB at 2.5 kB/s)  
Downloaded From central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-4.0.1.jar (193 kB at 1.4 kB/s)  
[INFO] Installing C:\Users\Administrator\Desktop\Project\mahaLogin\pom.xml to C:\Users\Administrator\.m2\repository\com\maha\mahaLogin\2.0\mahaLogin-2.0.pom  
[INFO] Installing C:\Users\Administrator\Desktop\Project\mahaLogin\target\mahaLogin-2.0.war to C:\Users\Administrator\.m2\repository\com\maha\mahaLogin\2.0\mahaLogin-2.0.war  
[INFO] [BUILD SUCCESS]  
[INFO] [INFO] Total time: 10.982 s  
[INFO] Finished at: 2025-04-13T03:42:24Z  
[INFO]
```

## What Happened Step-by-Step

### 1. Project Scanning

Scanning for projects...

📌 Maven reads `pom.xml` to understand the project structure and dependencies.

## 2. Build Phases Triggered (Lifecycle)

Your command `mvn install` automatically ran all these phases:

Phase	Status
e	s

`validate` ✓ Check project structure

`compile` ✓ Compile source code

`test` ✓ Run unit tests (`0 tests` in your case)

`package` ✓ Created `mahaLogin-2.0.war`

```
instal ✓ Installed .war in local  
1 repo
```

Artifact Created

Building war: C:\Users\Administrator\Desktop\Project\mahaLogin\target\mahaLogin-2.0.war

A .war file is created from your `src/main/webapp` folder — this is deployable on servers like **Tomcat**.

### Installed to Local Repository

Installed: C:\Users\Administrator\.m2\repository\com\maha\mahaLogin\2.0\mahaLogin-2.0.war

This means you can now reuse this war file as a dependency in other local Maven projects.

### Minor Log Warnings

[WARNING] cmd: not found /tmp/\_setup\_pre.js create!

This is not a Maven issue — seems like a leftover from environment setup (ignore unless you use a custom shell script).

### Final Status

[INFO] BUILD SUCCESS

Total time: 10.902 s

### Next Steps You Can Try

#### Action   Command

Deploy .war to Tomcat	Copy target/mahaLogin-2.0.war to webapps/ folder in Tomcat
-----------------------	--

Run on Jenkins/GitHub Actions    Use `mvn clean install` in pipeline

Share `.war` internally    Upload to Nexus/Artifactory using `mvn deploy`

Use in another Maven project    Add dependency in `pom.xml` with same `groupId`, `artifactId`, `version`

Let me know if you'd like help with:

- **Creating a CI pipeline**
- **Deploying this `.war` to Tomcat or Azure**
- **Generating a Maven `site` report**
- **Understanding `pom.xml` content**

**Command:** `mvn clean`

**Purpose:**

Deletes the `target/` directory created by a previous build.

**Why it's used:**

- To remove compiled `.class` files, jars, and other build artifacts.
- Ensures the **next build starts fresh** without any old or cached data.
- Prevents issues due to **residual files** from previous builds.

**Typical Use Case (Real World):**

Let's say you're developing a Java web app using Maven. You made code changes and built the project using `mvn package`. But something's not working right in the deployed `.war` file.

You suspect some old compiled files are lingering. Running:

```
mvn clean package
```

will:

1. Remove everything inside the `target/` folder.
2. Rebuild your application from scratch.
3. Generate a clean, new `.war` or `.jar` file without stale artifacts.

```
Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ ls
Dockerfile11 LICENSE README.md appgatewayurl.ps1 azure-pipelines.yml mahalogin.war pom.xml src/ testjava
Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ mvn clean package
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Building Account Registration Webapp 2.0
[INFO]   from pom.xml
[INFO]   [ war ]
[INFO]   [INFO] --- clean:3.2.0:clean (default-clean) @ mahalogin ---
[INFO]   [INFO] --- resources:3.3.1:resources (default-resources) @ mahalogin ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO]   Copying 4 resources from src/main/resources to target\classes
[INFO]   [INFO] --- compiler:3.13.0:compile (default-compile) @ mahalogin ---
[INFO]   Recompiling the module because of changed source code.
[WARNING] File encoding has not been set, using platform encoding windows-1252, i.e. build is platform dependent!
[INFO]   Compiling 11 source files with javac [debug target 1.7] to target\classes
[WARNING] bootstrap class path not set in conjunction with -source 1.7
[INFO]   [INFO] --- resources:3.3.1:testResources (default-testResources) @ mahalogin ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO]   skip non existing resourceDirectory C:\Users\Administrator\Desktop\Project\mahalogin\src\test\resources
[INFO]   [INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ mahalogin ---
[INFO]   No sources to compile
[INFO]   [INFO] --- surefire:3.2.5:test (default-test) @ mahalogin ---
[INFO]   No tests to run.
[INFO]   [INFO] --- war:3.3.1:war (default-war) @ mahalogin ---
[INFO]   Packaging webapp
[INFO]   Assembling webapp [mahalogin] in [C:\Users\Administrator\Desktop\Project\mahalogin\target\mahalogin-2.0]
[INFO]   Processing war project
[INFO]   Copying webapp Resources [C:\Users\Administrator\Desktop\Project\mahalogin\src\main\webapp]
[INFO]   Building war: C:\Users\Administrator\Desktop\Project\mahalogin\target\mahalogin-2.0.war
[INFO]   [INFO] BUILD SUCCESS
[INFO]   [INFO] Total time: 13.525 s
[INFO]   Finished at: 2025-04-13T03:20:47Z
[INFO]   [INFO]
```

## Maven Build Summary

### Command Run:

```
mvn clean package
```

# Project Info

- **Project Name:** mahaLogin
- **WAR File:** mahaLogin-2.0.war
- **Final Output:** Located in  
target/mahaLogin-2.0/mahaLogin-2.0.war

## Phases Breakdown

Phase	Plugin & Goal	Action Taken
clean	maven-clean-plugin:3.2.0:clean	Deleted previous target/directory.
resource	maven-resources-plugin:3.1.0:resources	Copied filtered resources to target/classes. Warning: using platform encoding Cp1252.
compile	maven-compiler-plugin:3.13.0:compile	Compiled 11 Java source files targeting Java 1.7.
testResources	maven-resources-plugin:3.1.0:testResources	Skipped: test resources directory does not exist.
testCompile	maven-compiler-plugin:3.13.0:testCompile	Skipped: no test sources found.
test	maven-surefire-plugin:3.2.5:test	Skipped: no tests to run.

```
package maven-war-plugin:3.3  Created .war file and copied resources from
e      .1:war                      src/main/webapp.
```

## Result

BUILD SUCCESS  
Total time: 13.525 s

## Next Steps (if needed):

- Deploy the WAR to Tomcat, JBoss, or another servlet container.
- Add test cases under `src/test/java` to enable test phases.
- Update `pom.xml` to set correct Java version if needed (currently using 1.7).

`mvn compile`

## What It Does:

- Compiles the project's **main source code** (`src/main/java`) into `.class` files.
- Outputs the compiled classes into the `target/classes/` directory.

## Maven Phase Triggered:

- Invokes the `compile` phase of the Maven **build lifecycle**.
- Uses the `maven-compiler-plugin` internally.

## Real Project Folder Example:

Let's say your project has:

```
src/
  main/
    java/
      com/
        example/
          App.java
```

After running:

```
bash
CopyEdit
mvn compile
```

You'll get:

```
swift
CopyEdit
target/classes/com/example/App.class
```

## Real-World Scenario:

You're working on a Java backend service and just updated some business logic in `UserService.java`.

To **test locally**, you want to compile the updated classes without creating a full `.jar` or `.war`.

Run:

```
mvn compile
```

This quickly checks for:

- **Syntax errors**
- **Missing imports**
- **Type mismatches**

...before you move on to packaging or deploying.

## Bonus Tips:

- To compile **and package**, use:

```
mvn clean package
```

To **recompile everything** from scratch:

```
mvn clean compile
```

Want to see what the `maven-compiler-plugin` settings look like in `pom.xml`?

```
Administrator@EC2AMAZ-6T5NOVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] < com.maha:mahaLogin > -----
[INFO] Building Account Registration Webapp 2.0
[INFO]   from pom.xml
[INFO] -----
[INFO] [ war ] -----
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ mahaLogin ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 4 resources from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ mahaLogin ---
[INFO] Nothing to compile - all classes are up to date.
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 4.195 s
[INFO] Finished at: 2025-04-13T03:25:14Z
[INFO] -----
```

  

```
Administrator@EC2AMAZ-6T5NOVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ ls
Dockerfile11 LICENSE README.md appgatewayur1.ps1 azure-pipelines.yml mahaLogin.war pom.xml src/ target/ testjava
```

Perfect – based on your latest screenshot, here's what the output of `mvn compile` indicates:

**`mvn compile` – Successful Build (No Recompilation Needed)**

## What Happened:

- Maven scanned your project and found it was **already compiled**.
- It skipped Java file compilation because **no changes were made** since the last successful build.

## Detailed Log Explanation:

Step	Plugin	Action
resources:resources	maven-resource-plugin	Copied 4 resource files from <code>src/main/resources</code> to <code>target/classes</code> .
compiler:compile	maven-compiler-plugin	 <b>"Nothing to compile - all classes are up to date."</b>

This means your `.java` files in `src/main/java` haven't changed since your last `mvn clean package`.

If you want to **force recompilation**, run:

```
mvn clean compile
```

That will delete the `target/` directory and freshly compile everything.

Let me know if you want to try a code change and recompile to see it in action.

```
Administrator@EC2AMAZ-6T5NOVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ mvn clean
bash.exe: warning: could not find /tmp, please create!
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.maha:mahaLogin >-----
[INFO] Building Account Registration Webapp 2.0
[INFO]   from pom.xml
[INFO] ----- [ war ] -----
[INFO]
[INFO] --- clean:3.2.0:clean (default-clean) @ mahaLogin ---
[INFO] Deleting C:\Users\Administrator\Desktop\Project\mahalogin\target
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 1.844 s
[INFO] Finished at: 2025-04-13T03:28:37Z
[INFO]
```

```
Administrator@EC2AMAZ-6T5NOVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ ls
Dockerfile LICENSE README.md appgatewayurl.ps1 azure-pipelines.yml mahaLogin.war pom.xml src/ testjava
```

```
Administrator@EC2AMAZ-6T5NOVT MINGW64 ~/Desktop/Project/mahalogin (master)
```

Awesome – in this last screenshot, the command `mvn clean` worked as expected. Here's a quick analysis:

**mvn clean – Executed Successfully**

 **What It Did:**

**Deleted** the `target/` directory:

mathematica

Deleting C:\Users\Administrator\Desktop\Project\mahalogin\target

- 
- Resulted in a **clean slate** for your next build.

Displayed:

nginx

BUILD SUCCESS

**Notable Warning:**

lua

bash.exe: warning: could not find /tmp, please create!

This warning is harmless for Maven but can be fixed by:

- Creating a `/tmp` directory for Git Bash to avoid future warnings:

bash

`mkdir /tmp`

## Why This Matters:

Now, when you run:

bash

```
mvn compile
```

or

bash

```
mvn package
```

it will **rebuild everything from scratch**, ensuring no old artifacts mess up the process.

Let me know if you'd like to do a `mvn package` next to verify everything rebuilds properly.

```
Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ mvn compile
bash.exe: warning: could not find /tmp, please create!
[INFO] Scanning for projects...
[INFO]
[INFO]   <!-- com.maha:mahaLogin -->
[INFO] Building Account Registration Webapp 2.0
[INFO]   from pom.xml
[INFO]   [ war ]
[INFO]
[INFO]   -- resources:3.3.1:resources (default-resources) @ mahaLogin --
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 4 resources from src\main\resources to target\classes
[INFO]
[INFO]   -- compiler:1.13.0:compile (default-compile) @ mahaLogin --
[INFO] Recompiling the module because of changed source code.
[WARNING] File encoding has not been set, using platform encoding windows-1252, i.e. build is platform dependent!
[INFO] Compiling 11 source files with javac [debug target 1.7] to target\classes
[WARNING] bootstrap class path not set in conjunction with -source 1.7
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 4.833 s
[INFO] Finished at: 2025-04-13T03:32:43Z
[INFO]

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ ls
Dockerfile LICENSE README.md appgateawayurl.ps1 azure-pipelines.yml mahalogin.war pom.xml src/ target/ testjava
Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ ls -al target/*
target/classes:
total 11
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 .
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 ..
-rw-r--r-- 1 Administrator 197121 255 Apr 11 18:51 application.properties
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 cov/
-rw-r--r-- 1 Administrator 197121 1295 Apr 11 18:51 db.sql
-rw-r--r-- 1 Administrator 197121 603 Apr 11 18:51 logback.xml
-rw-r--r-- 1 Administrator 197121 280 Apr 11 18:51 validation.properties

target/generated-sources:
total 0
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 .
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 ..
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 annotations/

target/maven-status:
total 0
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 .
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 ..
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 maven-compiler-plugin/

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$
```

Perfect – this confirms that the `mvn compile` step ran successfully after `mvn clean`, and everything was freshly rebuilt.

---

## ✓ mvn compile Summary (Based on Screenshot)

### 🔧 Key Actions:

Step Description  
on

Resource Copy      Copied 4 resources from `src/main/resources` →  
`target/classes`.

Compilation on      Compiled 11 Java source files using `javac`  
targeting version 1.7.

Output      Compiled classes and resources are now in  
`target/classes`.

## **target/classes/ Contents:**

Type	Name	Notes
*.class	In com/	Your compiled Java classes.
.properties	application.properties, validation.properties	Copied from resources.
.sql	db.sql	Likely a schema/data file used at runtime.
.xml	logback.xml	Logging config (likely Logback for SLF4J).

## **Other Target Subdirectories:**

- `target/generated-sources/annotations/`: Auto-generated files (e.g., for JPA, Lombok, etc.).
- `target/maven-status/maven-compiler-plugin/`: Status metadata for build tracking.

### **status:**

nginx

CopyEdit

BUILD SUCCESS

Everything looks good! You're now ready to:

- Run tests with `mvn test` (if any).
- Build deployable artifact using `mvn package`.

Would you like to:

1. Run `mvn package` now, or
2. Explore the output WAR file structure after packaging?

```

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ mvn test
bash.exe: warning: could not find /tmp, please create!
[INFO] Scanning for projects...
[INFO]
[INFO] < com.maha:mahaLogin >
[INFO] Building Account Registration Webapp 2.0
[INFO]   from pom.xml
[INFO]           [ war ]
[INFO]
[INFO] --- resources:3.3.1:resources (default-resources) @ mahaLogin ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 4 resources from src\main\resources to target\classes
[INFO]
[INFO] --- compiler:3.13.0:compile (default-compile) @ mahaLogin ---
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] --- resources:3.3.1:testResources (default-testResources) @ mahaLogin ---
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\Administrator\Desktop\Project\mahalogin\src\test\resources
[INFO]
[INFO] --- compiler:3.13.0:testCompile (default-testCompile) @ mahaLogin ---
[INFO] No sources to compile
[INFO]
[INFO] --- surefire:3.2.5:test (default-test) @ mahaLogin ---
[INFO] No tests to run.
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 4.346 s
[INFO] Finished at: 2025-04-13T03:35:15Z
[INFO]

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ ls
Dockerfile11 LICENSE README.md appgatewayurl.ps1 azure-pipelines.yml mahaLogin.war pom.xml src/ target/ testjava

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ ls -al target/*
ls: cannot access 'target/*': No such file or directory

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ ls -al target/*
target/classes:
total 11
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 ../
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 ../
-rw-r--r-- 1 Administrator 197121 255 Apr 11 18:51 application.properties
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 com/
-rw-r--r-- 1 Administrator 197121 1295 Apr 11 18:51 db.sql
-rw-r--r-- 1 Administrator 197121 603 Apr 11 18:51 logback.xml
-rw-r--r-- 1 Administrator 197121 280 Apr 11 18:51 validation.properties

target/generated-sources:
total 0
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 ../
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 ../
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 annotations/

target/maven-status:
total 0
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 ../
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 ../
drwxr-xr-x 1 Administrator 197121 0 Apr 13 03:32 maven-compiler-plugin/

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$
```

Great! You've successfully run:

```
bash
```

```
mvn test
```

Here's the breakdown of what happened:

## **mvn test – Test Phase Summary**

### **Key Observations:**

Phase	Action
resources:testResources	Looked for test resource files in <code>src/test/resources</code> , but none existed.
compiler:testCompile	No test source files to compile.
surefire:test	No unit tests found or executed.
<b>Result</b>	 <b>BUILD SUCCESS</b> – clean test run with no errors or failures.

## **Current target/ Directory Contents:**

You've listed the directory with:

```
bash
```

```
ls -al target/
```

## Existing Folders:

- `classes/` – Contains compiled main classes and resource files.
- `generated-sources/annotations/` – Placeholder for any annotation processors (like Lombok).
- `maven-status/maven-compiler-plugin/` – Metadata for Maven's internal state tracking.

 All folders were generated correctly during the build.

## Why No Tests Were Run?

Because:

- No `.java` test files in `src/test/java`.
- No test classes with JUnit/TestNG annotations.

## Next Steps You Can Try:

Create a test class in `src/test/java/com/maha/`:

```
java  
  
package com.maha;  
  
  
import org.junit.jupiter.api.Test;  
  
import static org.junit.jupiter.api.Assertions.*;
```

```
public class SampleTest {  
    @Test  
    void testAddition() {  
        assertEquals(2, 1 + 1);  
    }  
}
```

- 1.
2. Then run:

```
bash  
  
mvn test
```

```

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ mvn package
bash.exe: warning: could not find /tmp, please create!
[INFO] Scanning for projects...
[INFO] [ com.maha:mahaLogin >]
[INFO] Building Account Registration Webapp 2.0
[INFO]   from pom.xml
[INFO] [ war ]
[INFO]
[INFO] — resources:3.3.1:resources (default-resources) @ mahaLogin —
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 4 resources from src\main\resources to target\classes
[INFO]
[INFO] — compiler:3.13.0:compile (default-compile) @ mahaLogin —
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] — resources:3.3.1:testResources (default-testResources) @ mahaLogin —
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\Administrator\Desktop\Project\mahalogin\src\test\resources
[INFO]
[INFO] — compiler:3.13.0:testCompile (default-testCompile) @ mahaLogin —
[INFO] No sources to compile
[INFO]
[INFO] — surefire:3.2.5:test (default-test) @ mahaLogin —
[INFO] No tests to run.
[INFO]
[INFO] — war:3.3.1:war (default-war) @ mahaLogin —
[INFO] Packaging webapp
[INFO] Assembling webapp [mahaLogin] in [C:\Users\Administrator\Desktop\Project\mahalogin\target\mahaLogin-2.0]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\Users\Administrator\Desktop\Project\mahalogin\src\main\webapp]
[INFO] Building war: C:\Users\Administrator\Desktop\Project\mahalogin\target\mahaLogin-2.0.war
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time:  9.197 s
[INFO] Finished at: 2025-04-13T03:38:43Z
[INFO]

```

```

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$
```

## mvn package – Build Success & WAR Creation



### What Happened:

Phase	Status	Details
	s	
resources:resources	✓	4 resources copied to target/classes.

compiler:compile	✓	Skipped, no changes since last compile.
resources:testResources	✓	Skipped, test resources folder doesn't exist.
compiler:testCompile	✓	No test source files to compile.
surefire:test	✓	No tests to run.
war:war	✓	<b>WAR file created:</b> target/mahaLogin-2.0.war

## WAR File Location:

bash

CopyEdit

C:\Users\Administrator\Desktop\Project\mahalogin\target\mahaLogin-2.0.war

This .war file is ready to be deployed to a servlet container like **Apache Tomcat, JBoss, or Jetty**.

## Build Outcome:

css

CopyEdit

BUILD SUCCESS

Total time: 9.197 s

## Next Steps (Optional):

- **Deploy to Tomcat:** Copy `mahaLogin-2.0.war` to the `webapps/` folder of your Tomcat server and restart it.

**Explore WAR Contents (optional):** Run:

bash

```
jar tf target/mahaLogin-2.0.war
```

- to view its structure.

```

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~\Desktop\Project\mahaLogin (master)
$ mvn install
bash.exe: warning: could not find /tmp, please create!
[INFO] Scanning for projects...
[INFO]
[INFO] [ com.maha:mahaLogin >]
[INFO] Building Account Registration Webapp 2.0
[INFO] from pom.xml
[INFO] [ war ]
[INFO]
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.pom (8.5 kB at 7.4 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/42/maven-plugins-42.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/42/maven-plugins-42.pom (7.7 kB at 240 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/42/maven-parent-42.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/42/maven-parent-42.pom (50 kB at 459 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/32/apache-32.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/32/apache-32.pom (24 kB at 504 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-install-plugin/3.1.2/maven-install-plugin-3.1.2.jar (32 kB at 540 kB/s)
[INFO]
[INFO] — resources:3.3.1:resources (default-resources) @ mahaLogin —
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 4 resources from src\main\resources to target\classes
[INFO]
[INFO] — compiler:3.13.0:compile (default-compile) @ mahaLogin —
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] — resources:3.3.1:testResources (default-testResources) @ mahaLogin —
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory C:\Users\Administrator\Desktop\Project\mahaLogin\src\test\resources
[INFO]
[INFO] — compiler:3.13.0:testCompile (default-testCompile) @ mahaLogin —
[INFO] No sources to compile
[INFO]
[INFO] — surefire:3.2.5:test (default-test) @ mahaLogin —
[INFO] No tests to run.
[INFO]
[INFO] — war:3.3.1:war (default-war) @ mahaLogin —
[INFO] Packaging webapp
[INFO] Assembling webapp [mahaLogin] in [C:\Users\Administrator\Desktop\Project\mahaLogin\target\mahaLogin-2.0]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\Users\Administrator\Desktop\Project\mahaLogin\src\main\webapp]
[INFO] Building war: C:\Users\Administrator\Desktop\Project\mahaLogin\target\mahaLogin-2.0.war
[INFO]
[INFO] — install:3.1.2:install (default-install) @ mahaLogin —
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-util/1.9.18/maven-resolver-util-1.9.18.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-util/1.9.18/maven-resolver-util-1.9.18.pom (2.9 kB at 89 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver/1.9.18/maven-resolver-1.9.18.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver/1.9.18/maven-resolver-1.9.18.pom (22 kB at 1.4 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-api/1.9.18/maven-resolver-api-1.9.18.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-api/1.9.18/maven-resolver-api-1.9.18.pom (2.7 kB at 86 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-4.0.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-4.0.1.pom (7.8 kB at 522 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-util/1.9.18/maven-resolver-util-1.9.18.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-util/1.9.18/maven-resolver-util-1.9.18.jar (196 kB at 3.1 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-api/1.9.18/maven-resolver-api-1.9.18.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/resolver/maven-resolver-api/1.9.18/maven-resolver-api-1.9.18.jar (157 kB at 2.5 MB/s)
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-utils/4.0.1/plexus-utils-4.0.1.jar (193 kB at 1.4 MB/s)
[INFO] Installing C:\Users\Administrator\Desktop\Project\mahaLogin\pom.xml to C:\Users\Administrator\.m2\repository\com\maha\mahaLogin\2.0\mahaLogin-2.0.pom
[INFO] Installing C:\Users\Administrator\Desktop\Project\mahaLogin\target\mahaLogin-2.0.war to C:\Users\Administrator\.m2\repository\com\maha\mahaLogin\2.0\mahaLogin-2.0.war
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 10.982 s
[INFO] Finished at: 2025-04-13T03:42:24Z
[INFO]

```

```

Administrator@EC2AMAZ-6T5NDVT MINGW64 ~\Desktop\Project\mahaLogin (master)
$
```

Awesome! You've just run:

bash

```
mvn install
```

## Here's the Full Breakdown of What Happened:

---

### Phase: `install`

This command does everything that `package` does **plus**:

- 📦 Installs the built `.war` artifact into your local Maven repository (`~/.m2/repository`)  
So you (or other Maven modules) can reuse it **without re-downloading or rebuilding**.

Step	Plugin/Phase	Status	Description
p	se	s	on

validate, compile, test, package		Reused cached builds (no changes).
-------------------------------------	---	---------------------------------------

```
war:war ✓ Built the WAR:  
r           target/mahaLogin-2.0.war
```

```
install:install ✓ Copied WAR  
to:
```

```
C:\Users\Administrator\.m2\repository\com\maha\mahaLogin\2.0\mahaLog  
in-2.0.war
```

### Installed Artifact Location:

bash

```
~/.m2/repository/com/maha/mahaLogin/2.0/mahaLogin-2.0.war
```

This WAR can now be referenced from any Maven project using:

xml

CopyEdit

```
<dependency>
```

```
  <groupId>com.maha</groupId>
```

```
  <artifactId>mahaLogin</artifactId>
```

```
  <version>2.0</version>
```

```
<type>war</type>  
</dependency>
```

## Final Result:

css

BUILD SUCCESS

Total time: 10.982 s

## Next Steps (optional):

- ✓ Deploy WAR to server
- ↲ Create another Maven module and add this WAR as a dependency
- 🔎 Inspect local .m2 to verify it was installed correctly

```
Administrator@EC2AMAZ-6TSD9V: ~\Desktop\Project\nahaLogin (master)
$ mvn deploy
[WARNING] Could not find /tmp, please create!
[INFO] Scanning for projects...
[INFO]
[INFO] <!-- con:nahaLogin -->
[INFO] Building Account Registration Webapp 2.0
[INFO] from pom.xml
[INFO] [ war ]
[INFO]
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/nexus/plugins/nexus-deploy-plugin/3.1.2/nexus-deploy-plugin-3.1.2.pom
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/nexus/plugins/nexus-deploy-plugin/3.1.2/nexus-deploy-plugin-3.1.2.pom (9.6 kB at 14 kB/s)
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/nexus/plugins/nexus-deploy-plugin/3.1.2/nexus-deploy-plugin-3.1.2.jar
Downloaded From central: https://repo.maven.apache.org/maven2/org/apache/nexus/plugins/nexus-deploy-plugin/3.1.2/nexus-deploy-plugin-3.1.2.jar (40 kB at 363 kB/s)
[INFO]
[INFO] — resources:3.1.1:resources (default-resources) @ nahaLogin —
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] Copying 4 resources from src/main/resources to target\classes
[INFO]
[INFO] — compiler:3.1.3:compile (default-compile) @ nahaLogin —
[INFO] Nothing to compile - all classes are up to date.
[INFO]
[INFO] — resources:3.1.1:testResources (default-testResources) @ nahaLogin —
[WARNING] Using platform encoding (Cp1252 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resource directory C:\Users\Administrator\Desktop\Project\nahaLogin\src\test\resources
[INFO]
[INFO] — compiler:3.1.3:testCompile (default-testCompile) @ nahaLogin —
[INFO] No sources to compile
[INFO]
[INFO] — surefire:3.2.5:test (default-test) @ nahaLogin —
[INFO] No tests to run.
[INFO]
[INFO] — war:3.1.1:war (default-war) @ nahaLogin —
[INFO] Packaging webapp
[INFO] Assembling webapp [nahaLogin] in [C:\Users\Administrator\Desktop\Project\nahaLogin\target\nahaLogin-2.0]
[INFO] Processing war project
[INFO] Copying webapp resources [C:\Users\Administrator\Desktop\Project\nahaLogin\src\main\webapp]
[INFO] Building war: C:\Users\Administrator\Desktop\Project\nahaLogin\target\nahaLogin-2.0.war
[INFO]
[INFO] — install:3.1.2:install (default-install) @ nahaLogin —
[INFO] Installing C:\Users\Administrator\Desktop\Project\nahaLogin\pom.xml to C:\Users\Administrator\.m2\repository\com\aha\ahaLogin\2.0\ahaLogin-2.0.pom
[INFO] Installing C:\Users\Administrator\Desktop\Project\nahaLogin\target\ahaLogin-2.0.war to C:\Users\Administrator\.m2\repository\com\aha\ahaLogin\2.0\ahaLogin-2.0.war
[INFO]
[INFO] — deploy:3.1.2:deploy (default-deploy) @ nahaLogin —
[INFO]
[INFO] BUILD FAILURE
[INFO]
[INFO] Total time: 10.021 s
[INFO] Finished at: 2025-04-13T03:45:54Z
[INFO]
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-deploy-plugin:3.1.2:deploy (default-deploy) on project nahaLogin: deployment failed: repository element was not specified in the POM inside distributionManagement element or in -DaltDeploymentRepository<id>:url parameter
> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following articles:
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/NojoExecutionException

Administrator@EC2AMAZ-6TSD9V: ~\Desktop\Project\nahaLogin (master)
$
```

Thanks for sharing the screenshot — you ran:

bash

```
mvn deploy
```

And you got this error:

```
X Deploy failed: repository element was not specified in the  
POM inside distributionManagement element or in  
-DaltDeploymentRepository=id::url
```

## Problem Explained

The `mvn deploy` goal is used to upload the built `.war` or `.jar` to a remote Maven repository (like Nexus, Artifactory, etc.).

But your `pom.xml` is missing the required `distributionManagement` section, so Maven doesn't know where to deploy.

## Fix: Add `distributionManagement` in `pom.xml`

Here's how to fix it:

 In your `pom.xml`, add:

xml

```
<distributionManagement>  
    <repository>  
        <id>my-repo</id>  
  
        <url>http://your-nexus-or-artifactory-server/repository/releases</url>  
    </repository>  
</distributionManagement>
```

Replace the `url` with your actual remote repository URL (e.g., Nexus or Artifactory endpoint).

## Iso Add Authentication (if needed)

In your Maven `settings.xml` (usually in `~/.m2/settings.xml`), add:

xml

```
<servers>
  <server>
    <id>my-repo</id> <!-- This must match the <id> in your pom.xml -->
    <username>your-username</username>
    <password>your-password</password>
  </server>
</servers>
```

## Alternative (Temporary) Deploy:

If you just want to deploy without modifying `pom.xml`, use the command line:

bash

```
mvn deploy:deploy-file \
-DgroupId=com.maha \
-DartifactId=mahaLogin \
-Dversion=2.0 \
-Dpackaging=war \
-Dfile=target/mahaLogin-2.0.war \
-DrepositoryId=my-repo \
-Durl=http://your-repo-url/repository/releases
```

Let me know if you want help setting this up for a real Nexus or Artifactory deployment.

```
Administrator@EC2AMAZ-6T5NDVT MINGW64 ~/Desktop/Project/mahalogin (master)
$ mvn site
bash.exe: warning: could not find /tmp, please create!
[INFO] Scanning for projects...
[INFO]
[INFO] -----< com.maha:mahaLogin >-----
[INFO] Building Account Registration Webapp 2.0
[INFO]   from pom.xml
[INFO] ----- [ war ] -----
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-site-plugin/3.12.1/maven-site-plugin-3.12.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-site-plugin/3.12.1/maven-site-plugin-3.12.1.pom (20 kB at 30 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/36/maven-plugins-36.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-plugins/36/maven-plugins-36.pom (9.9 kB at 183 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/36/maven-parent-36.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-parent/36/maven-parent-36.pom (45 kB at 722 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/apache/26/apache-26.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/apache/26/apache-26.pom (21 kB at 411 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-site-plugin/3.12.1/maven-site-plugin-3.12.1.jar
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/plugins/maven-site-plugin/3.12.1/maven-site-plugin-3.12.1.jar (119 kB at 2.5 MB/s)
[INFO]
[INFO] --- site:3.12.1:site (default-site) @ mahaLogin ---
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting-api/3.1.1/maven-reporting-api-3.1.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting-api/3.1.1/maven-reporting-api-3.1.1.pom (3.8 kB at 221 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/doxia/doxia-sink-api/1.11.1/doxia-sink-api-1.11.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/doxia/doxia-sink-api/1.11.1/doxia-sink-api-1.11.1.pom (1.6 kB at 49 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/doxia/doxia/1.11.1/doxia-1.11.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/doxia/doxia/1.11.1/doxia-1.11.1.pom (18 kB at 564 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/doxia/doxia-logging-api/1.11.1/doxia-logging-api-1.11.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/doxia/doxia-logging-api/1.11.1/doxia-logging-api-1.11.1.pom (1.6 kB at 51 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting-exec/1.6.0/maven-reporting-exec-1.6.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting-exec/1.6.0/maven-reporting-exec-1.6.0.pom (14 kB at 434 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting-api/3.1.0/maven-reporting-api-3.1.0.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/reporting/maven-reporting-api/3.1.0/maven-reporting-api-3.1.0.pom (3.8 kB at 118 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-model/3.2.5/maven-model-3.2.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-model/3.2.5/maven-model-3.2.5.pom (4.2 kB at 88 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-component-annotations/2.1.1/plexus-component-annotations-2.1.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-component-annotations/2.1.1/plexus-component-annotations-2.1.1.pom (770 B at 25 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-containers/2.1.1/plexus-containers-2.1.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus-containers/2.1.1/plexus-containers-2.1.1.pom (6.0 kB at 189 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/6.5/plexus-6.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/codehaus/plexus/plexus/6.5/plexus-6.5.pom (26 kB at 1.5 MB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/eclipse/sisu/org.eclipse.sisu.plexus/0.3.5/org.eclipse.sisu.plexus-0.3.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/eclipse/sisu/org.eclipse.sisu.plexus/0.3.5/org.eclipse.sisu.plexus-0.3.5.pom (4.3 kB at 138 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/eclipse/sisu/sisu-plexus/0.3.5/sisu-plexus-0.3.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/eclipse/sisu/sisu-plexus/0.3.5/sisu-plexus-0.3.5.pom (14 kB at 858 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/eclipse/sisu/org.eclipse.sisu.inject/0.3.5/org.eclipse.sisu.inject-0.3.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/eclipse/sisu/org.eclipse.sisu.inject/0.3.5/org.eclipse.sisu.inject-0.3.5.pom (2.6 kB at 82 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/eclipse/sisu/sisu-inject/0.3.5/sisu-inject-0.3.5.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/eclipse/sisu/sisu-inject/0.3.5/sisu-inject-0.3.5.pom (14 kB at 465 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/archiver/3.5.2/maven-archiver-3.5.2.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/archiver/3.5.2/maven-archiver-3.5.2.pom (5.5 kB at 179 kB/s)
Downloading from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-artifact/3.1.1/maven-artifact-3.1.1.pom
Downloaded from central: https://repo.maven.apache.org/maven2/org/apache/maven/maven-artifact/3.1.1/maven-artifact-3.1.1.pom (2.0 kB at 12 kB/s)
```

xcellent — you've now run:

bash

`mvn site`

And Maven is in the process of downloading reporting plugins and generating project documentation.

## What `mvn site` Does:

The `site` phase generates a static website for your project, including:

Report	Description
<code>Project Info</code>	Basic metadata from <code>pom.xml</code> like name, description, license, developers.
<code>Dependencies</code>	All project dependencies and their scopes.
<code>Plugins</code>	List of plugins used.
<code>Javadoc</code>	Auto-generated JavaDocs for your source code.
<code>Surefire Reports</code>	Unit test results (if any).
<code>Project Summary</code>	Overview of the project.

## Where Is the Output?

Once finished, you'll find a full HTML site in:

bash

`target/site/index.html`

You can open it in your browser to explore the Maven reports.

## What's Next?

Once the download and generation finish:

Navigate to:

`bash`

`target/site/index.html`

- 1.
2. Open it in any browser:

`pgsql`

`start target/site/index.html`

## What is a Maven Repository?

A Maven repository is a storage location where project dependencies, plugins, and build artifacts are stored and retrieved.

It's used by Maven to:

- Download external libraries (like JARs, WARs).
- Store your own project outputs (when you use `mvn install` or `mvn deploy`).

## Types of Maven Repositories

Type	Description	Examples
Local Repository	A folder on your system where Maven caches downloaded	<code>~/.m2/repository</code>

**artifacts and stores  
builds you install.**

**Central Repository** Maven's default public repository. Used if dependency not found locally.

**Remote Repository** An external/internal server (e.g. Nexus, Artifactory) used to share artifacts in teams or enterprises.

**Private/Internal Repository** Customized repositories for company use – for private builds or restricted environments.

## Lifecycle Example: When You Run `mvn install`

1. Searches Local Repo (`~/.m2/repository`)
2. If not found, downloads from:
  - Maven Central
  - or Remote Repos (defined in `pom.xml` or `settings.xml`)
3. Installs output (`.jar/.war`) into local repo

4. If you run `mvn deploy`, it uploads to a configured Remote Repo (via `distributionManagement`)

## Configure Repositories in `pom.xml`

xml

CopyEdit

```
<repositories>

    <repository>
        <id>central</id>
        <url>https://repo.maven.apache.org/maven2</url>
    </repository>

    <repository>
        <id>my-internal-repo</id>
        <url>https://repo.company.com/maven2</url>
    </repository>

</repositories>
```

## `distributionManagement` for Deployment

xml

CopyEdit

```
<distributionManagement>

    <repository>
```

```
<id>releases</id>

<url>https://repo.company.com/releases</url>

</repository>

<snapshotRepository>

<id>snapshots</id>

<url>https://repo.company.com/snapshots</url>

</snapshotRepository>

</distributionManagement>
```

Used during `mvn deploy` to push your build artifacts.

Let me know if you want:

- Example `settings.xml` for authentication
- Setup for deploying to Nexus or Artifactory
- Explanation of `snapshot` vs `release` repositories