**DevOps- MAVEN**

MAVEN is a build tool used to convert the human readable code to machine readable code. For ex: the applications which we use like notepad, zoom and etc written in some coding language. But when we download the app for our use we will download only the .exe file.

In java we have many tools to comprise the code. MAVEN is one of them. Maven is used to comprise the java code with all the information. The comprised file will be in .jar(java archive file) or .war file.

Grader is an another build tool used to comprise the code.

Java- MAVEN

Python- pybuilder

Golang- gogradle

Nodejs- webpack

Rust- cargo

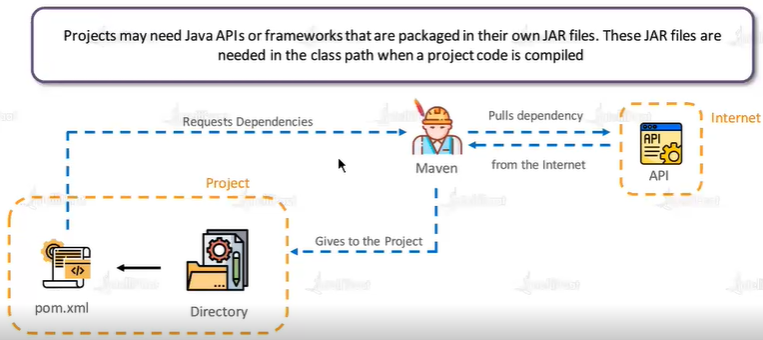
C# - cakebuild

**Why we use MAVEN**: the code which is written by the developers should be packaged and converted into an executable format. Converts the source code to executable format.This can be done using maven.

**Maven will perform 3 major tasks –** 1. Downloading dependencies, 2. Enforcing a directory structure 3. Building an executable with all dependencies.

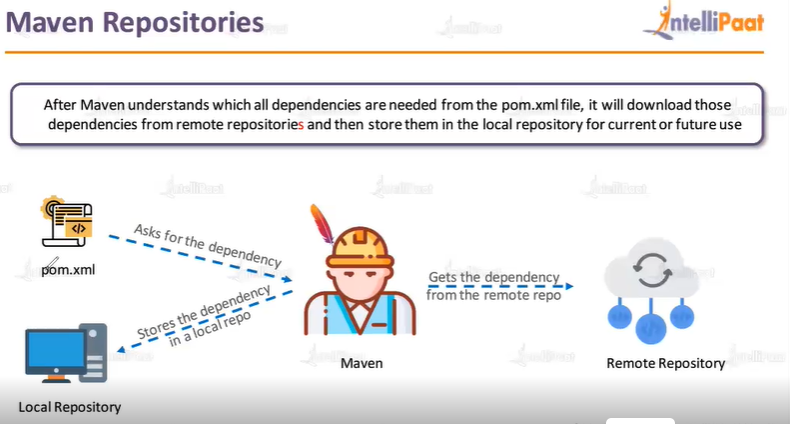
**What is maven?** Maven is a build tool, used to manage the entire life cycle of a project, generate the reports, and store the documents with its pom repository.

**Project Dependencies:**



Whenever we are working on Maven we will be getting pom.xml along with source. This POM is referred as project object model. For maven pom.xml is the input. Without pom.xml maven will not work. This POM is the heart of the MAVEN.

POM is the input file for maven. This pom file will be in xml format. We can create the POM file by following below steps.



1. Go to [Maven Repository: Search/Browse/Explore (mvnrepository.com)](https://mvnrepository.com/)
2. Search for the required dependency- select the version which is latest or compatible, and then copy and paste the file inside of POM.xml.

Once we have done the above steps, our pom.xml file will search the respective dependency in [Index of / (apache.org)](https://repo.maven.apache.org/). Then this will be given as a input to the project code.

For ex, below is our pom.xml

<!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

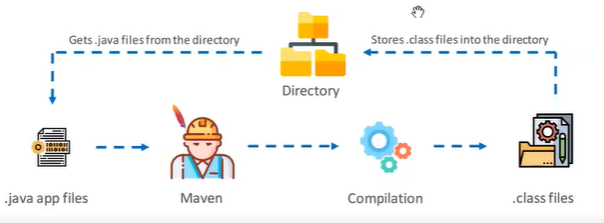
<version>8.0.33</version>

</dependency>

First it will search for groupid (mysql) dependency, then it will select the artifactid mentioned, then it will take the mentioned version.

**Maven lifecycle or the phases of Maven or commands of Maven:**

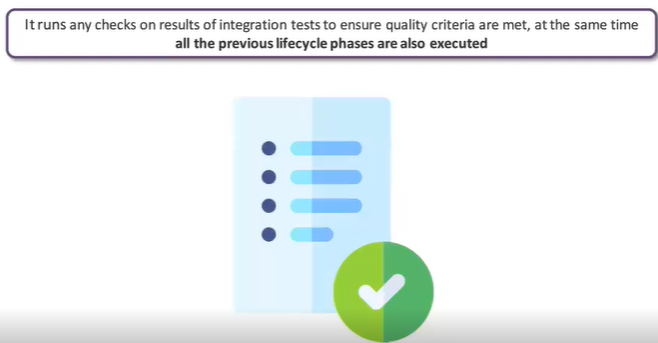
1. **Validate phase- mvn validate (checks whether the structure is proper or not)** 🡪 Maven will check whether the project is correct and all the necessary information is available – with the use of command
2. **Compile phase - mvn compile (converts .java file to .class**)🡪 in this phase maven will convert human readable code to machine language code and put back in the same location.



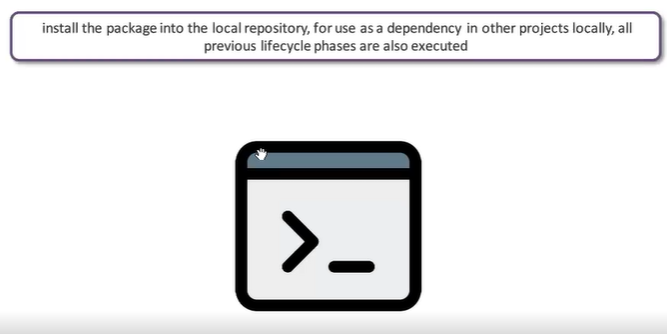
1. **Test phase - mvn test (test cases will be executed)🡪** maven will execute the specific test cases and provides summary log as a test report.
2. **Package phase - mvn package(converts all the .class files into either .jar or .war or .ear)🡪**

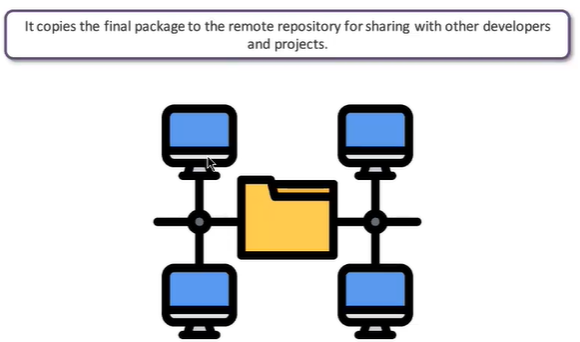


1. **Verify phase- mvn verify**



1. **Install phase- mvn install**- install the app in our local.



1. **Deploy phase- mvn deploy-** publish & make the file available to other developers and projects.

**Practice session.**

Login to EC2- install the jdk before installing the maven.

To install java- **sudo yum install java-11-amazon-corretto-headless**

To install maven**- sudo yom install maven**

Check whether its installed or not**- mvn –version**

To work in maven we need one java project. We can use the below link to create the basic sample java program. [Maven – Maven in 5 Minutes (apache.org)](https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html)

By using the below code, we can create simple maven project.

mvn archetype:generate -DgroupId=com.mycompany.app -DartifactId=my-app -DarchetypeArtifactId=maven-archetype-quickstart -DarchetypeVersion=1.4 -DinteractiveMode=false

Once we run the above code, one new directory will get created. Inside that we can find all the POM.xml, source code, test code and other few files. We can see the files using **tree** command. Tree is the basic linux command used to show the all files with their flow.

We can execute mvn package command one we have all the required files. When we execute mvn package the .jar file will get created. When we are giving back the code to developer we no need to give the target files, that time we can use mvn clean command to remove the target folder.

*As a devops person, we are just responsible to check whether we are able to create a .jar file or not.*

*We use apache tomcat tool to deploy our java code.*

*We can change the packaging format inside the POM.xml*

Once the .jar or .war file is created we need to deploy the application into respective server. Here we are using tomcat tool to deploy.

**Below are the steps to deploy web application manually:**

To download apache tomcat, go to copy the link. Run **wget** link which we copied command

Then we need to untar the zip folder using **tar –xvzf.**

After unzipping the folder, we need to start the server using **./startup.sh**