# MAVEN

- Maven is an automation project management tool.
- It is used to build the code
- once we build the code we will get JAR/WAR/EAR
- Maven is used to add the dependencies to our application.
- Maven is based on POM.xml

POM: Project Object Model

XML: Extensible Markup Language.

- POM.xml contains project related data (metadata, kind of project, kind of output, description, dependencies).
- Maven was developed by Apache software foundations.
- Maven was released in 2004.
- Maven can build any number of projects into desired Output such as .jar, .war and .ear

.jar = java archive file

.war = web archive file

.EAR = enterprise archive

- It is mostly used for java-based projects.
- It was initially released on 13 July 2004.
- Maven is written in java.

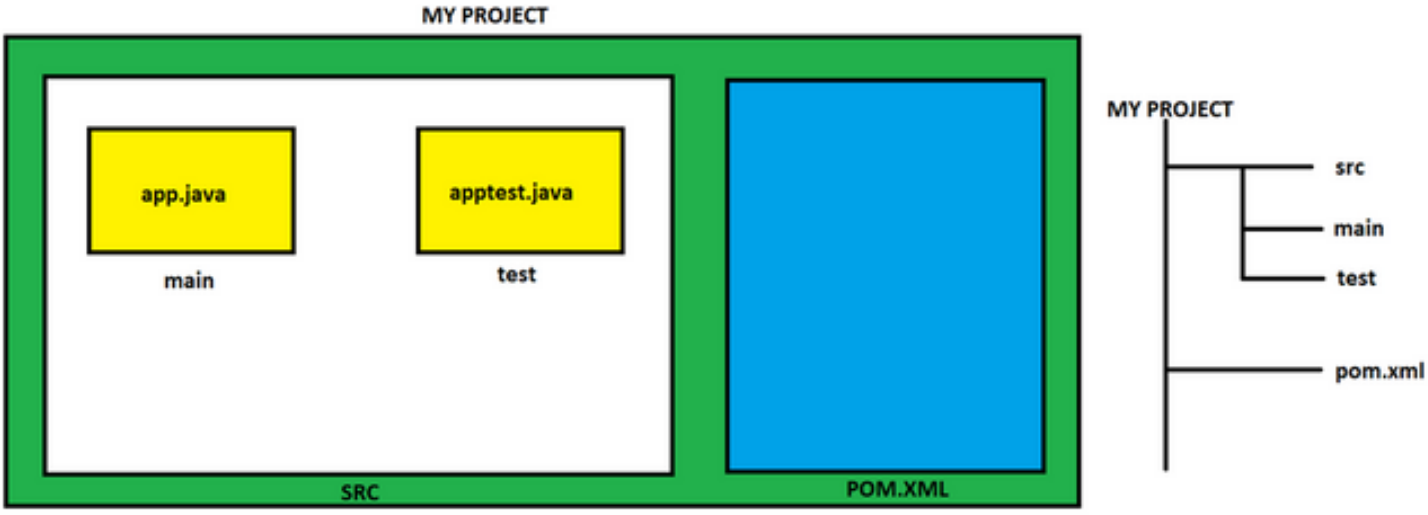
## MAVEN BUILD LIFE CYCLE:

1. 1 Generate Resource
2. 2. Compile Code
3. 3. Unit Test
4. 4. Package (build)
5. 5. install (into Local repo or Artifactory)
6. 6. Deploy (to servers)
7. 7. Clean (to delete all the runtime files)

## BUILD TOOL:

- it is used to set up everything which is required to run your java code This can be applied to your entire java project.
- It generates source code, compiling code, packaging code to a jar etc.
- POM refers the XML file that have all information regarding project and configuration details
- Main configuration file is in pom.xml.
- It has description of the Project details regarding version and configuration management.
- The XML file is in the Project home directory.

## MAVEN DIRECTORY STRUCTURE:




## WE HAVE 7 PHASES IN BUILD LIFE CYCLE

### STEP-1: LAUNCH EC2 INSTANCE

**INSTALL JAVA** : yum install java-1.8.0-openjdk -y

**TO CHECK THE VERSION**: java -version

**INSTALL MAVEN** : yum install maven -y

**TO CHECK VERSION**: mvn -v

**CLONE THE REPO**: git clone https://github.com/devops0014/one.git

**CHECKOUT TO BRANCH :** git checkout master

## STEP-2: COMPILE THE CODE

**GOAL :** mvn compile

## STEP-3: TEST THE CODE

**GOAL:** mvn test

## STEP-4: BUILDN THE CODE

**GOAL:** mvn package

## STEP-5: INSTALL THE CODE

**GOAL:** mvn install

## STEP-6: CLEAN THE CODE (opt)

**GOAL:** mvn clean

**NOTE:** TO DO AL THESE STEPS IN SINGLE GOAL : **mvn clean package**

## ANT VS MAVEN:

| Ant | Maven |
| --- | --- |
| Ant **doesn't has formal conventions**, so we need to provide information of the project structure in build.xml file. | Maven **has a convention** to place source code, compiled code etc. So we don't need to provide information about the project structure in pom.xml file. |
| Ant is **procedural**, you need to provide information about what to do and when to do through code. You need to provide order. | Maven is **declarative**, everything you define in the pom.xml file. |
| There is **no life cycle** in Ant. | There is **life cycle** in Maven. |
| It is **a tool** box. | It is **a framework**. |
| It is **mainly a build tool**. | It is **mainly a project management tool**. |
| The ant scripts are **not reusable**. | The maven plugins are **reusable**. |
| It is **less preferred** than Maven. | It is **more preferred** than Ant. |