23.02Jul.Apache Maven

Apache Maven

- Java Based Application.
- Create Structure of Enterprise. Project. (archetype)
- Build
  - Check Error. ⟶ src / test
  - Read pom.xml
    - Download dependencies.
    - Compile Test Cases.
    - Test (Integration, Unit, Load, Performance, Stress)
    - Package (Artifact)
    - Install (Compile ⟶ self test)
    - Deploy ⟶ EC2 / Tomcat
  - Verify.
  - Validate

POM.XML

$ mvn   Command1   Command2 ↵

or

$ mvn   Command1

Prerequisites:
1. Java Programming Language (8.0)
   a. JDK
   b. JRE
2. Apache Maven 3.9.3
3. VSCode

Check Versions:
$ javac -version   JDK
$ java -version   JRE
$ mvn --version
$ code --version
$ git --version

Group(Solution)
        Artifact(Project)

GENERATE PROJECT

Windows:
```
mvn archetype:generate ^
-DgroupId=com.nubeera.app ^
-DartifactId=my-app ^
-DarchetypeArtifactId=maven-archetype-quickstart ^
-DarchetypeVersion=1.4 ^
-DinteractiveMode=false
```

Linux/Mac OS:
```
mvn archetype:generate \
-DgroupId=com.nubeera.app \
-DartifactId=my-app \
-DarchetypeArtifactId=maven-archetype-quickstart \
-DarchetypeVersion=1.4 \
-DinteractiveMode=false
```

Check In to Repository(VCS- Github)

```
$ git add .
$ git commit -m "Message"
$ git push -u origin master
```

Developer Write Code:
- Java Files
- Configuration Files
- Push

1. Pull Code in Build Server(Maven)
2. Clean Previous Output(#   mvn clean   )
3. Compile Project (# mvn compile)
4. Run Project
   a. $ cd target\classes
   b. $ java com.nubeera.app.App
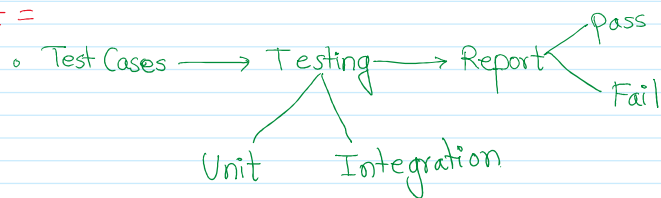
Apache Maven Build Lifecycle Phases

1] Validate =
- validate the project is correct
- Necessary information of Project.

2] Compile =
- Source Code ⟶ Translate(compile) ⟶ Output (in .class Format)

3] Test =
- Test Cases ⟶ Testing ⟶ Report ⟨ Pass / Fail

  Unit   Integration

4] Package =

  Package = Compile + Test ⟶ Output(JAR, WAR)

5] Verify =
- Run any checks on Result of Integration test
  to ensure project Quality   Criteria ⟨ Met / or Not
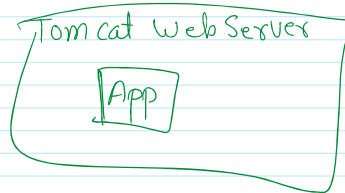 (Rules)

6] install =
       ◦ install Project output into an Local Server

7] Deploy =
       ◦ Build Environment output - - - → Deploy

Remote Environment
Tomcat Web Server

App

$ mvn surefire:test

Apache Maven

Plugins =   ◦ Extend Functionality of Apache Maven

       ◦ Deploy application to tomcat
             Maven ——→ Tomcat

Dependency =   ◦ Extend Functionality of Project

       ◦ Library of Project
Example

       Logging ——→ Log4J

VCS(Github)   Build Server   Web Server

dev1   src/main   Maven   deploy   (Tomcat)

Package

src/test   Compile   test

Code

tester1   Error   Rpt   Web container

Error/Test Report

Notification

Notification

Error/Test Report

Dependencies

Local

Central → Maven Repository.

Maven Repositories