MAVEN COMPLETE

Maven does three things rather well
1.Dependency Management:
2. Compilation through convention:
3. Everything Java: Maven can also run code quality checks, execute test cases and even deploy applications to remote servers, through plugins.

Java source code is to be meant to live in the ***"/src/main/java"*** folder
Maven will put compiled Java classes into the ***"target/classes"*** folder
Maven will also build a .jar or .war file, depending on your project, that lives in the "target" folder.

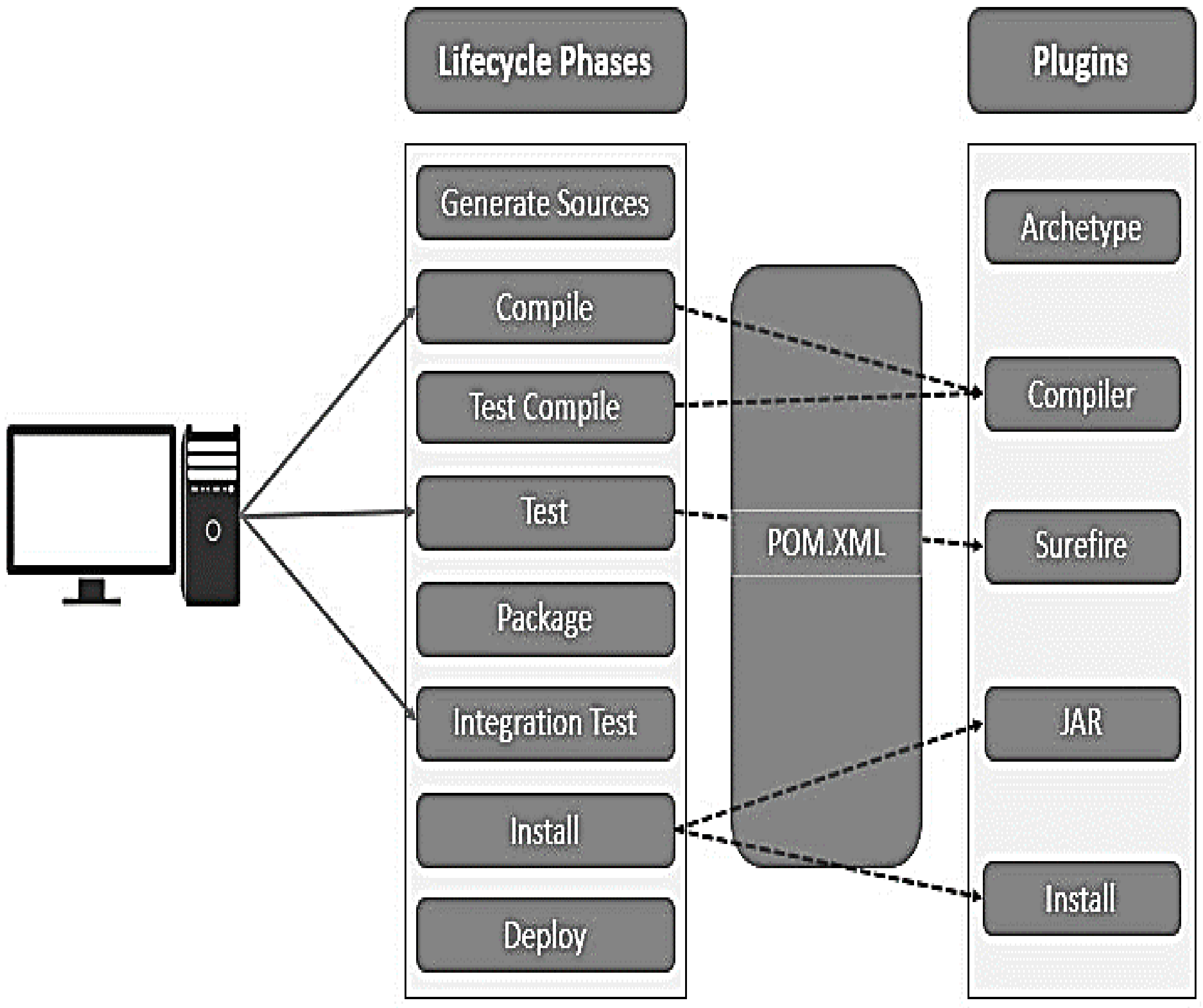
SUPPORT 3 TYPES OF REPOS:
Local (named .m2)– Folder location on the local Dev machine. When maven build is executed, Maven automatically downloads all the dependency jars into the local repository.
Linux: /home/<User_Name>/.m2 : Windows: C:\Users\<User_Name>\.m2
Central – Repository provided by Maven community
Remote – Organization owned custom repository

Maven Build Lifecycle
The Maven build follows a specific life cycle to deploy and distribute the target project.
There are three built-in life cycles:
DEFAULT: the main life cycle as it's responsible for project deployment
CLEAN: to clean the project and remove all files generated by the previous build
SITE: to create the project's site documentation

Maven Phase: Some of the most important phases in the default build lifecycle:
Validate: check if all information necessary for the build is available
Compile: compile the source code
Test-compile: compile the test source code
Test: run unit tests
Package: package compiled source code into the distributable format (jar, war, …)
Integration-test: process and deploy the package if needed to run integration tests
Install: install the package to a local repository
Deploy: copy the package to the remote repository
EX: if we run the deploy phase – which is the last phase in the default build lifecycle – that will execute all phases before the deploy phase as well, which is the entire default lifecycle

Maven Goal :
Each phase is a sequence of goals, and each goal is responsible for a specific task.
When we run a phase – all goals bound to this phase are executed in order.
Here are some of the phases and default goals bound to them:
compiler:compile – the compile goal from the compiler plugin is bound to the compile phase
compiler:testCompile is bound to the test-compile phase
surefire:test is bound to test phase
install:install is bound to install phase
jar:jar and war:war is bound to package phase

Maven Plugins:
Maven is actually a plugin execution framework where every task is actually done by plugins. Maven Plugins are generally used to −
1. create jar file 2. create war file 3. compile code files 4. unit testing of code 5. create project documentation 6. create project reports
mvn [plugin-name]:[goal-name] EX: mvn compiler:compile
Two types of plugins
1. **Build plugins**: They execute during the build process and should be configured in the <build/>
2. **Reporting plugins**: They execute during the site generation process and they should be configured in the <reporting/>
some plugins
clean: Cleans up target after the build. Deletes the target directory.
compiler: Compiles Java source files.
surefire: Runs the JUnit unit tests. Creates test reports.
jar: Builds a JAR file from the current project.
javadoc: Generates Javadoc for the project.



Maven's Multi-module project

A multi-module project is built from an Parent POM and sub modules.
The Parent is located in the project's root directory and must have packaging of type pom.
Parent POM: Maven supports inheritance in a way that each pom.xml file has the implicit parent POM, it's called Super POM and can be located in the Maven binaries. By setting packaging to pom type, we're declaring that project will serve as a parent.
Submodules or subprojects are regular Maven projects that inherit from parent POM.
To build or release our project in one shot, we have to declare our submodules explicitly in parent POM. when running mvn package command in the parent project directory, Maven will build and test all three modules.

Settings.xml

The settings element in the settings.xml file contains elements used to define values which configure Maven execution in various ways, like the pom.xml, but should not be bundled to any specific project, or distributed to an audience. These include values such as the local repository location, alternate remote repository servers, and authentication information.
There are two locations where a settings.xml file may live:

- **The Maven install: ${maven.home}/conf/settings.xml**
- **A user's install: ${user.home}/.m2/settings.xml**

Settings.xml file contains:

- Server properties: id, username, pwd, file permissions,
- Proxy Properties: id, protocol, host ports
- Repositories: release snapshot, update policy
- Profiles: versions, propety, os
- Properties: evn variable, env path

Password Encryption: in settings.xml

${user.home}/.m2/settings-security.xml -----> Master Password is stored

```
mvn --encrypt-master-password <password>
```

settings-security.xml

```
<settingsSecurity>
  <master>{jSMOWnoPFgsHVpMvz5VrIt5kRbzGpI8u+9EF1iFQyJQ=}</master>
</settingsSecurity>
```

settings.xml -----> encrypt server passwords

```
<settings>
<server>
     <id>my.server</id>
     <username>foo</username>
     <password>{COQLCE6DU6GtcS5P=}</password>
   </server>
</settings>
mvn --encrypt-password <password>
```

MAVEN Questions

1. Maven Goals

ANS: compiler:compile – the compile goal from the compiler plugin is bound to the compile phase
compiler:testCompile is bound to the test-compile phase
surefire:test is bound to test phase
install:install is bound to install phase
jar:jar and war:war is bound to package phase

2. mvn install what will happen

Install will then compile, test & package your Java project and even install/copy your built .jar/.war file into your local Maven repository. which lives in ~/.m2/repository.

3 .M2 folder what is this

Local (named .m2)– Folder location on the local Dev machine. When maven build is executed, Maven automatically downloads all the dependency jars into the local repository.

4 What does mvn -U do?

Maven always tries to download the latest snapshot dependency versions, invoke it with the -U switch. Sometimes, if your project depends on SNAPSHOT dependencies, Maven will not update your local Maven repository with the very latest snapshot version.

5 Settings required before mvn deploy

Ans: Add a server definition with an id that matches that of the deployment repository. Put your username, password in settings.xml

6. MVN execution 1st and 2nd time. time difference

Ans: First time every thing is stored in .m2 or local repo. Second time it uses. so less time

7. Maven Archtype

Archetype is a Maven project templating tool used for reference to create model files
mvn archetype:generate
Eclipse ----> new Java project
mvn eclipse:eclipse
maven-archetype-site-simple
Maven Webapp Archetype

8. Multi Module Project and Config need to do in parent and child project? What is dependency management?

A multi-module project is built from an Parent POM and sub modules.
The Parent is located in the project's root directory and must have packaging of type pom.

9. Transitive Dependency?

project A specifies a dependency on another project B, and project B specifies a dependency on project C. C is a transitive dependency for A.
C has scope compile within B, then declaring B as dependency of A suffices to build A with Maven.

10. .M2 is local repo. Can i change my local repo folder? How?

Ans:config file is located in the Maven installation directory named settings.xml

```
<settings>
   <localRepository>C:/maven_repository</localRepository>
```

11 Maven follows Convention over configuration that means code under src/main/java. I want to change it is it possible?
ANS: You can set the sourceDirectory in the build tag of your POM

```
<build>
   <sourceDirectory>src/Javasource</sourceDirectory>
   ...
  </build>
```

13. Maven Build Life cycle
Ans: default: the main life cycle as it's responsible for project deployment
clean: to clean the project and remove all files generated by the previous build
site: to create the project's site documentation

14. Should we have to mention output artefact type in maven? Under which tag it is mentioned?
Ans: For jar not required but for war it is required. Tag is packaging

15. What are things need to set if you want to download from dependency from private Repo?
Ans: Have to define the project location in pom.xml as id. Have to put credentials in other file.

16. Issues faced while working on Maven Projects
Ans: **Dependencies** are not built in the local repository: Module A and Module B. Module B has a dependency to Module A. The changes you made to Module B are not visible in Module A. make any changes, you need to place a copy of new jar by using **mvn install** in the changed project
Dependency version is not correct: because of the Nearest Definition First rule version is not correct. mvn dependency:tree it will output a tree will all the dependencies and versions for the project. This is very helpful to debug this kind of problems.

17. command to skip the test cases in maven
To skip running the tests for a particular project, set the skipTests property to true.
skip the tests via the command line
$ mvn install -DskipTests
or $ ***mvn install -Dmaven.test.skip=true***