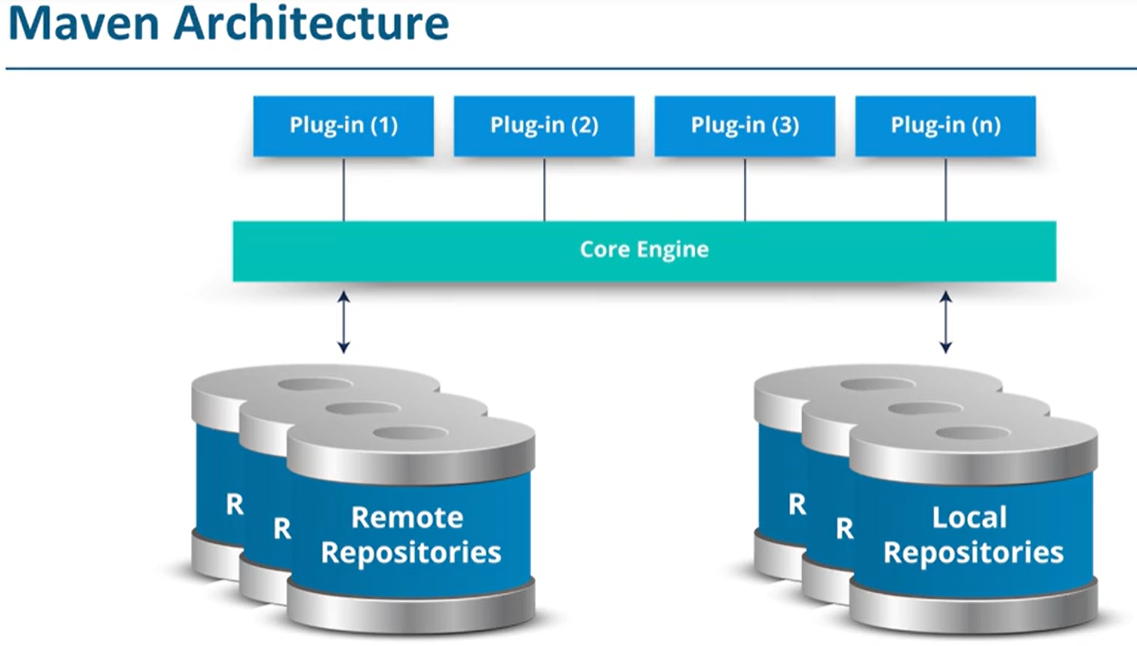
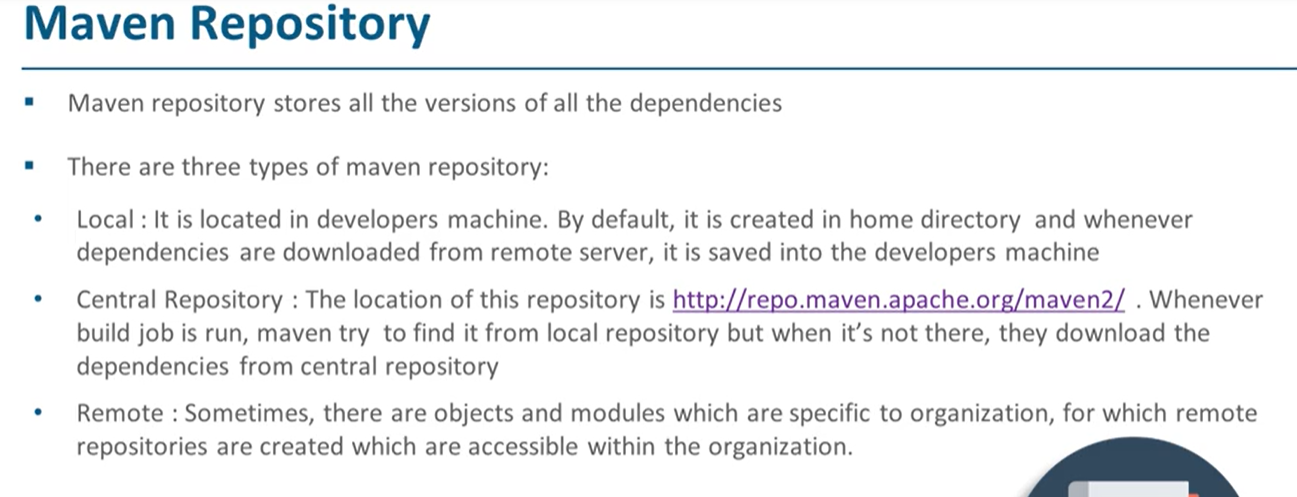
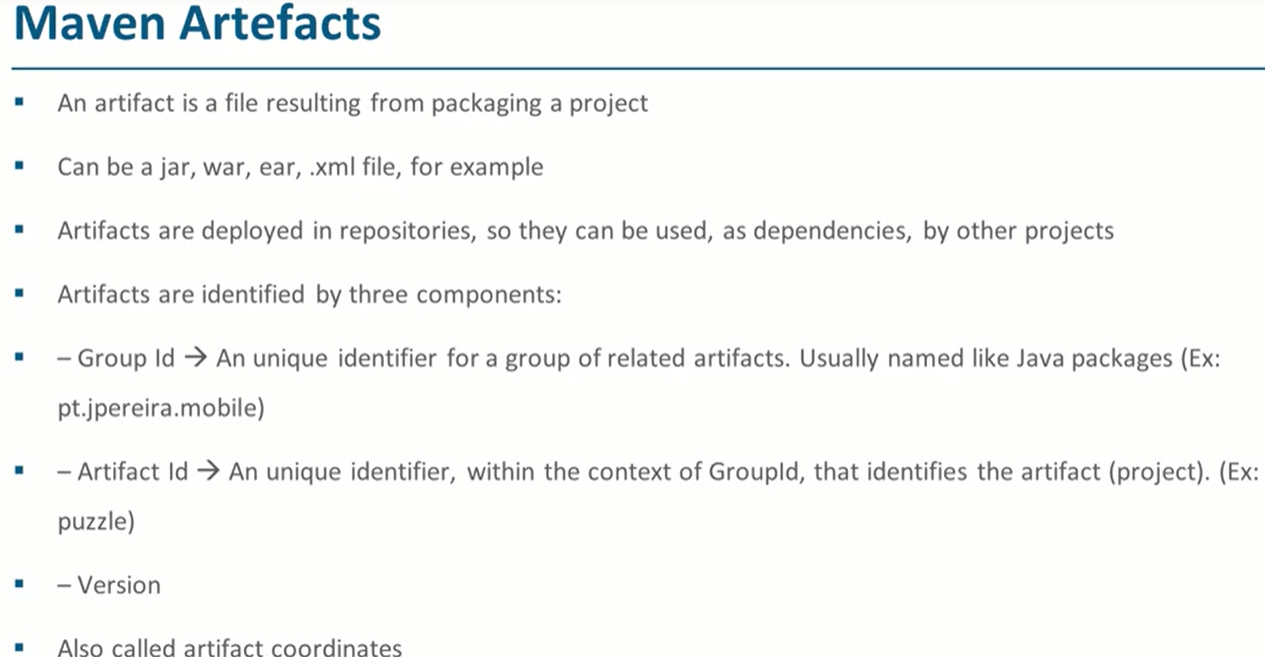
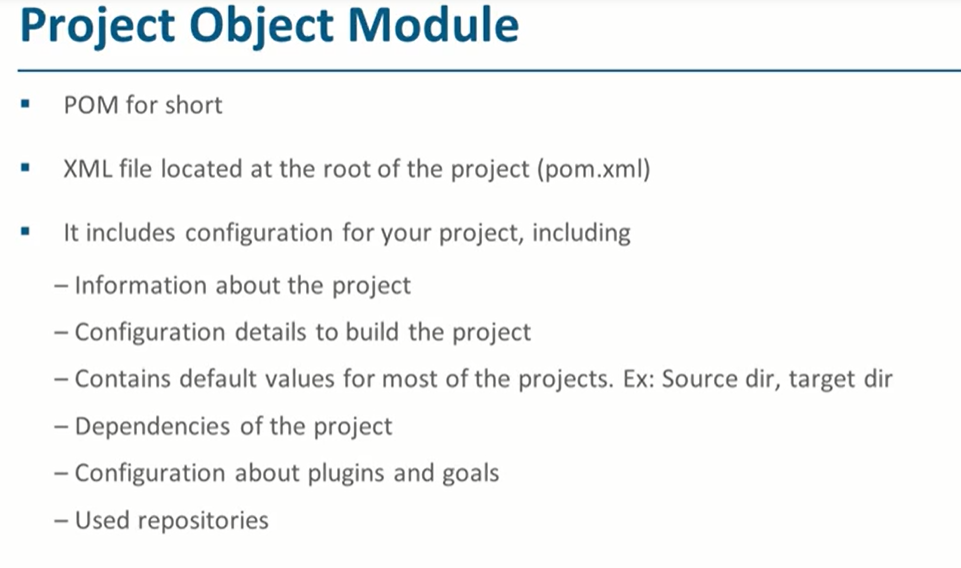
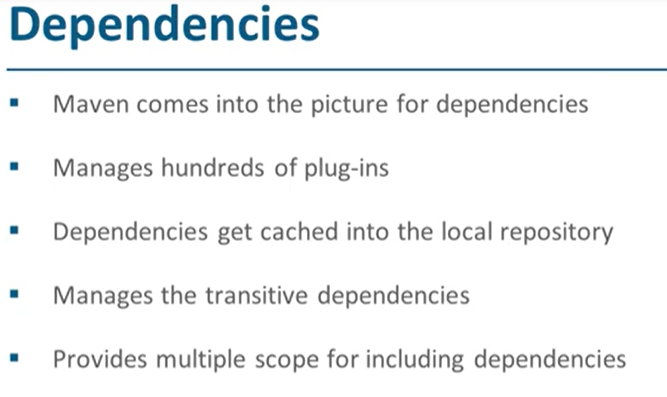
MAVEN

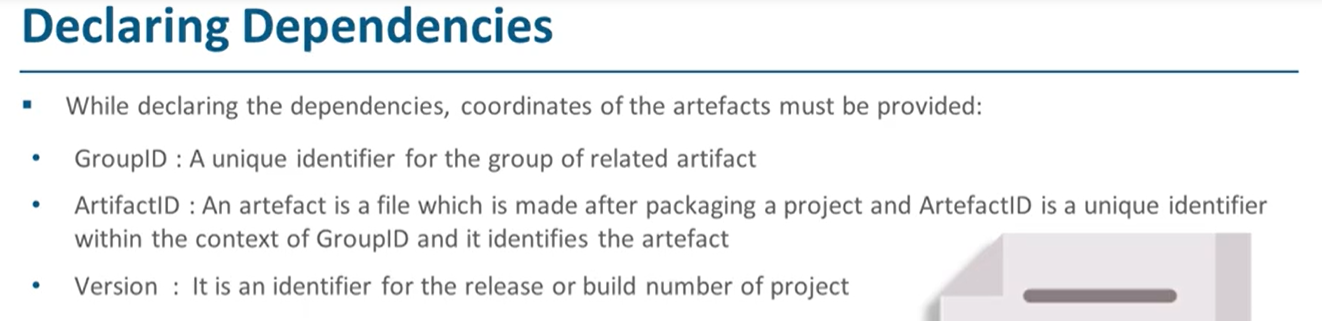


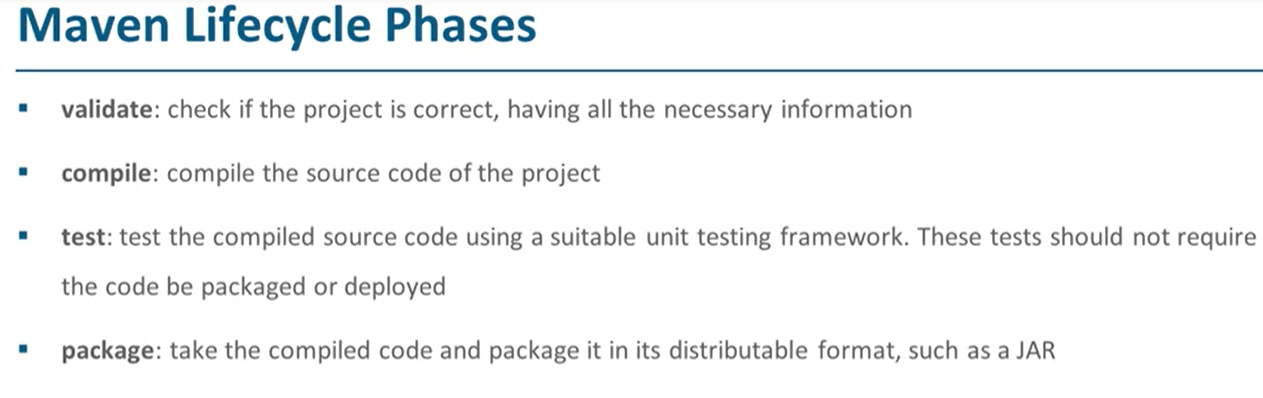












Global tool setting

Maven integration plugin

Build pipeline plugin

Deploy to container plugin

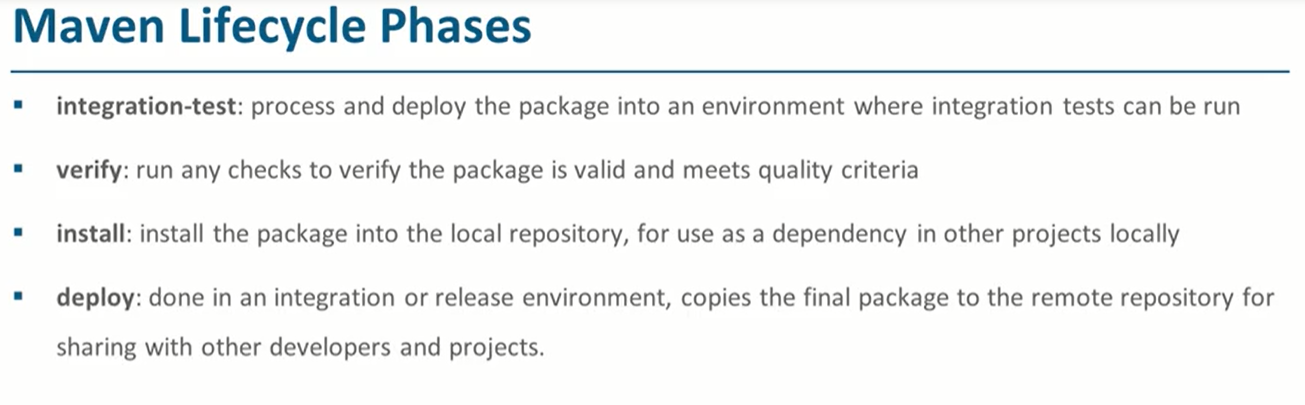
Email extension pulugin/template

Docker pipeline

Git

Junit

pipeline



Maven Installation:

Maven download <https://maven.apache.org/download.cgi>

Maven installation <https://maven.apache.org/install.html>

mvn --version

**Creating a Project**

You need somewhere for your project to reside. Create a directory somewhere and start a shell in that directory. On your command line, execute the following Maven goal:

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

*If you have just installed Maven, it may take a while on the first run. This is because Maven is downloading the most recent artifacts (plugin jars and other files) into your local repository. You may also need to execute the command a couple of times before it succeeds. This is because the remote server may time out before your downloads are complete. Don't worry, there are ways to fix that.*

You will notice that the *generate* goal created a directory with the same name given as the artifactId. Change into that directory.

cd my-app

Under this directory you will notice the following [standard project structure](https://maven.apache.org/guides/introduction/introduction-to-the-standard-directory-layout.html).

my-app

|-- pom.xml

`-- src

|-- main

| `-- java

| `-- com

| `-- mycompany

| `-- app

| `-- App.java

`-- test

`-- java

`-- com

`-- mycompany

`-- app

`-- AppTest.java

The src/main/java directory contains the project source code, the src/test/java directory contains the test source, and the pom.xml file is the project's Project Object Model, or POM.

### The POM

The pom.xml file is the core of a project's configuration in Maven. It is a single configuration file that contains the majority of information required to build a project in just the way you want. The POM is huge and can be daunting in its complexity, but it is not necessary to understand all of the intricacies just yet to use it effectively. This project's POM is:

1. <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
2. xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3. <modelVersion>4.0.0</modelVersion>
5. <groupId>com.mycompany.app</groupId>
6. <artifactId>my-app</artifactId>
7. <version>1.0-SNAPSHOT</version>
9. <properties>
10. <maven.compiler.source>1.7</maven.compiler.source>
11. <maven.compiler.target>1.7</maven.compiler.target>
12. </properties>
14. <dependencies>
15. <dependency>
16. <groupId>junit</groupId>
17. <artifactId>junit</artifactId>
18. <version>4.12</version>
19. <scope>test</scope>
20. </dependency>
21. </dependencies>
22. </project>

### What did I just do?

You executed the Maven goal *archetype:generate*, and passed in various parameters to that goal. The prefix *archetype* is the [plugin](https://maven.apache.org/plugins/index.html) that provides the goal. If you are familiar with [Ant](http://ant.apache.org/), you may conceive of this as similar to a task. This *archetype:generate* goal created a simple project based upon a [maven-archetype-quickstart](https://maven.apache.org/archetypes/maven-archetype-quickstart/) archetype. Suffice it to say for now that a *plugin* is a collection of *goals* with a general common purpose. For example the jboss-maven-plugin, whose purpose is "deal with various jboss items".

### Build the Project

mvn package

The command line will print out various actions, and end with the following:

...

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 2.953 s

[INFO] Finished at: 2019-11-24T13:05:10+01:00

[INFO] ------------------------------------------------------------------------

Unlike the first command executed (*archetype:generate*), the second is simply a single word - *package*. Rather than a *goal*, this is a *phase*. A phase is a step in the [build lifecycle](https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html), which is an ordered sequence of phases. When a phase is given, Maven executes every phase in the sequence up to and including the one defined. For example, if you execute the *compile* phase, the phases that actually get executed are:

1. validate
2. generate-sources
3. process-sources
4. generate-resources
5. process-resources
6. compile

You may test the newly compiled and packaged JAR with the following command:

java -cp target/my-app-1.0-SNAPSHOT.jar com.mycompany.app.App

Which will print the quintessential:

Hello World!