What is m2 folder in Maven?  
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.

===================================

==========================  
What is m2 repository?  
Artifact **Repositories**. A **repository** in **Maven** is used to hold build artifacts and dependencies of varying types. ... The local **repository** refers to a copy on your own installation that is a cache of the remote downloads, and also contains the temporary build artifacts that you have not yet released.  
===================================================  
What are the repositories in Maven?  
In **Maven** terminology, a **repository** is a directory where all the project jars, library jar, plugins or any other project specific artifacts are stored and can be used by**Maven** easily. **Maven repository** are of three types.  
========================================================  
What is a snapshot Maven?  
A **snapshot** version in **Maven** is one that has not been released. The idea is that before a 1.0 release (or any other release) is done, there exists a 1.0-**SNAPSHOT** . That version is what might become 1.0 . It's basically " 1.0 under development".  
==========================================================  
  
What is use of Maven surefire plugin?     
**Maven Surefire Plugin**. ... The **Surefire Plugin** is used during the test phase of the build lifecycle to execute the unit tests of an application. It generates reports in two different file formats: Plain text files (\*.txt)  
====================================================  
Why do we use Maven?

**Maven** is a powerful project management tool that is based on POM (project object model). It is **used** for projects build, dependency and documentation. ... - **maven**repository is a directory of packaged JAR file with pom.xml file. **Maven** searches for dependencies in the repositories.  
===================================================  
What is the use of Maven compiler plugin?  
Apache **Maven Compiler Plugin**. The **Compiler Plugin** is used to **compile** the sources of your project. Since 3.0, the default **compiler** is javax.tools.JavaCompiler (if you are using java 1.6) and is used to **compile** Java sources. ... Other compilers than javac can be used and work has already started on AspectJ, .NET, and C#.  
===================================================  
What is the Junit?  
**JUnit** is a unit testing framework for the Java programming language. **JUnit** has been important in the development of test-driven development, and is one of a family of unit testing frameworks which is collectively known as xUnit that originated with SUnit.  
==================================================  
Why do we need to use Junit?  
**JUnit** is a unit testing framework for the Java Programming Language. It is important in the test driven development, and is one of a family of unit testing frameworks collectively known as xUnit. ... **JUnit** tests can be run automatically and they check their own results and provide immediate feedback.  
===================================================

What is meant by goal in Maven?  
  
Phases executes sequentially. Executing a phase means executes all previous phases. Plugin is a collection of **goals**. Analogy : Plugin is a class and **goals** are methods within the class. **Maven** is based around the central concept of a build lifecycle.  
=====================================================  
What is the use of Maven clean?  
The **Maven Clean** Plugin, as the name implies, attempts to **clean** the files and directories generated by **Maven** during its build. While there are plugins that generate additional files, the **Clean** Plugin assumes that these files are generated inside the target directory.  
===================================================  
What is an artifact in Maven?  
An **artifact** is a file, usually a JAR, that gets deployed to a **Maven** repository. A**Maven** build produces one or more **artifacts**, such as a compiled JAR and a "sources" JAR. Each **artifact** has a group ID (usually a reversed domain name, like com.example.foo), an **artifact** ID (just a name), and a version string.  
=========================================================  
  
What is a maven build?  
**Maven** is a **build** automation tool used primarily for Java projects. In Yiddish, the word **maven** means "accumulator of knowledge". **Maven** addresses two aspects of building software: first, it describes how software is built, and second, it describes its dependencies.     
======================================================  
What is the profile in Maven?  
A **Build profile** is a set of configuration values, which can be used to set or override default values of **Maven build**. Using a **build profile**, you can customize **build** for different environments such as Production v/s Development environments.  
======================================================  
What does Maven package do?

**package** will compile your code and also **package** it. For example, if your pom says the project is a jar, it will create a jar for you when you **package** it and put it somewhere in the target directory (by default). install will compile and **package**, but it will also put the **package** in your local repository.  
=======================================================

What is the use of Maven in Jenkins?  
**Maven** is a build tool, in short a successor of ant. It helps in build and version control. However **Jenkins** is continuous integration system, where in **maven** is used for build. **Jenkins** can be used to automate the deployment process  
====================================================  
8 mins  
What is Gradle and Maven?  
**Gradle** combines good parts of both tools and builds on top of them with DSL and other improvements. ... **Gradle** does not use XML. Instead, it had its own DSL based on Groovy (one of JVM languages). As a result, **Gradle** build scripts tend to be much shorter and clearer than those written for Ant or **Maven**.  
========================================================  
What is the use of plugins in Maven?  
What is a **Plugin**? "**Maven**" is really just a core framework for a collection of **Maven Plugins**. In other words, **plugins** are where much of the real action is performed,**plugins** are used to: create jar files, create war files, compile code, unit test code, create project documentation, and on and on.  
=========================================================  
What is a Maven POM file?  
A Project Object Model or **POM** is the fundamental unit of work in **Maven**. It is an XML file that contains information about the project and configuration details used by**Maven** to build the project. It contains default values for most project  
====================================================  
What is an archetype in Maven?  
In short, **Archetype** is a **Maven** project templating toolkit. An **archetype** is defined as an original pattern or model from which all other things of the same kind are made. The name fits as we are trying to provide a system that provides a consistent means of generating **Maven** projects.  
=========================================================

What are the types of Maven repository?  
A **repository** in **Maven** is used to hold build artifacts and dependencies of varying**types**. There are strictly only two **types** of **repositories**: local and remote. ... Remote **repositories** refer to any other **type** of **repository**, accessed by a variety of protocols such as file:// and http://.  
====================================================  
Where is my maven local repository?  
The **maven local repository** is a **local** folder that is used to store all your project's dependencies (plugin jars and other files which are downloaded by **Maven**). In simple, when you build a **Maven** project, all dependency files will be stored in your**Maven local repository**.  
=========================================================  
What is the use of Maven compiler plugin?  
Apache **Maven Compiler Plugin**. The **Compiler Plugin** is used to **compile** the sources of your project. Since 3.0, the default **compiler** is javax.tools.JavaCompiler (if you are using java 1.6) and is used to **compile** Java sources. ... Other compilers than javac can be used and work has already started on AspectJ, .NET, and C#  
=============================================================  
What is a Maven dependency?  
When **maven** is run on project B version 1.0 of artifacts a, b, c, and d will be used regardless of the version specified in their pom. a and c both are declared as**dependencies** of the project so version 1.0 is used due to **dependency** mediation. Both will also have runtime scope since it is directly specified.  
=======================================================  
Where are Maven dependencies downloaded to?  
The jars , **dependency** files and other files which are **downloaded** by **Maven** reside in the **Maven** local repository. By default the **Maven** local repository is the .m2 folder. You can copy the jar directly into where it is meant to go. **Maven** will find this file next time it is runs.

=====================================================  
What is Maven settings XML?  
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**.