Top 50 Devops Maven Interview Questions and Answers Pdf

1. Explain what is Maven ? How does it work ?

Answer: Maven is a project management tool. It provides the developer a complete build lifecycle framework. On executing Maven commands, it will look for POM file in Maven; it will run the command on the resources described in the POM.

2. List out what are the aspects does Maven Manages ?

Answer: Maven handles following activities of a developer

- Build
- Documentation
- Reporting
- Dependencies
- SCMs
- Releases
- Distribution
- Mailing list

3. Mention the three build lifecycle of Maven ?

Answer:

- Clean: Cleans up artifacts that are created by prior builds
- Default (build): Used to create the application
- Site: For the project generates site documentation

4. Explain what is POM ?

Answer: In Maven, POM (Project Object Model) is the fundamental unit of work. It is an XML file which holds the information about the project and configuration details used to build a project by Maven.

5. Explain what is Maven artifact ?

Answer: Usually an artifact is a JAR file which gets arrayed to a Maven repository. One or more artifacts a maven build produces such as compiled JAR and a sources JAR.

Each artifact includes a group ID, an artifact ID and a version string.

6. Explain what is Maven Repository ? What are their types ?

Answer: A Maven repository is a location where all the project jars, library jars, plugins or any other particular project related artifacts are stored and can be easily used by Maven.

Their types are local, central and remote.

7. Why Maven Plugins are used ?

Answer: Maven plugins are used to

- Create a jar file
- Create war file
- Compile code files
- Unit testing of code
- Documenting projects
- Reporting

8. List out the dependency scope in Maven ?

Answer: The various dependency scope used in Maven are:

- Compile: It is the default scope, and it indicates what dependency is available in the classpath of the project
- Provided: It indicates that the dependency is provided by JDK or web server or container at runtime
- Runtime: This tells that the dependency is not needed for compilation but is required during execution
- Test: It says dependency is available only for the test compilation and execution phases
- System: It indicates you have to provide the system path
- Import: This indicates that the identified or specified POM should be replaced with the dependencies in that POM’s section

9. Mention how profiles are specified in Maven ?

Answer: Profiles are specified in Maven by using a subset of the elements existing in the POM itself.

10. Explain how you can exclude dependency ?

Answer: By using the exclusion element, dependency can be excluded.

11. Mention the difference between Apache Ant and Maven ?

Answer: Apache Ant Maven

- Ant is a toolbox – Maven is a framework
- Ant does not have formal conventions like project directory structure – Maven has conventions
- Ant is procedural; you have to tell to compile, copy and compress – Maven is declarative (information on what to make & how to build)
- Ant does not have lifecycle; you have to add sequence of tasks manually – Maven has a lifecycle
- Ant scripts are not reusable – Maven plugins are reusable

12. In Maven what are the two setting files called and what are their location ?

Answer: In Maven, the setting files are called settings.xml, and the two setting files are located at

- Maven installation directory: $M2_Home/conf/settings.xml
- User’s home directory: ${ user.home }/ .m2 / settings.xml

13. List out what are the build phases in Maven ?

Answer: Build phases in Maven are

- Validate
- Compile
- Test
- Package
- Install
- Deploy

14. List out the build, source and test source directory for POM in Maven ?

Answer:

- Build = Target
- Source = src/main/java
- Test = src/main/test

15. Where do you find the class files when you compile a Maven project ?

Answer: You will find the class files ${basedir}/target/classes/.

16. Explain what would the "jar: jar" goal do ?

Answer: jar: jar will not recompile sources; it will imply just create a JAR from the target/classes directory considering that everything else has been done.

17. List out what are the Maven's order of inheritance ?

Answer: The maven's order of inheritance is

- Parent Pom
- Project Pom
- Settings
- CLI parameters

18. For POM what are the minimum required elements ?

Answer: The minimum required elements for POM are project root, modelVersion, groupID, artifactID and version.

19. Explain how you can produce execution debug output or error messages ?

Answer: To produce execution debug output you could call Maven with X parameter or e parameter.

20. Explain how to run test classes in Maven ?

Answer: To run test classes in Maven, you need surefire plugin, check and configure your settings in setting.xml and pom.xml for a property named "test."

21. What is a MOJO ?

Answer: A MOJO stands for Maven plain Old Java Object. Each MOJO is an executable goal in Maven, and a plugin is a distribution of one or more related MOJOs.

22. What does it mean when you say Maven uses Convention over Configuration ?

Answer: Maven uses Convention over Configuration which means developers are not required to create build process themselves. Developers do not have to mention each and every configuration details.

23. What are the main features of Maven ?

Answer: Some of the main features of Maven are:

Simple: Maven provides simple project setup that is based on best practices.

Fast: You can get a new project or module started in a few seconds in Maven.

Easy to learn: Maven usage and commands are easy to learn across all projects. Therefore ramp up time for new developers coming onto a project is very less.

Dependency management: Maven provides superior dependency management including automatic updates and transitive dependencies.

Multiple Projects: You can easily work with multiple projects at the same time by using Maven.

Large Library: Maven has a large and growing repository of libraries and metadata to use out of the box.

Extensible: Maven supports the ability to easily write plugins in Java or scripting languages for extending its core functionality.

Instant: Maven is online and it provides instant access to new features with very less configuration.

24. What are the main advantages of Maven ?

Answer: Maven has a long list of advantages for Software development. Some of the main advantages are:

Common Project Structure: By using Maven, every developer has a common project structure that helps in understanding the code as well as developing new features in a new project.

Modular Design: Maven promotes modular design that divides a complex project into multiple modules that are easier to manage. By using Maven, it is easier to manage multiple modules for build, test, release etc.

Centralized Dependency Management: With Maven, each developer does not have to include the jars separately in each project or module. Maven provides a centralized dependency management that can help improve efficiency of software development.

Fewer Decisions: With Maven a developer has to make fewer decisions about things unrelated to software development work. The project structure comes ready with Maven, dependency management is a uniform approach and build/release are handled by Maven. So a developer can focus on core work of developing software.

25. Why do we say "Maven uses convention over configuration" ?

Answer: Convention over configuration is a Software Design Paradigm that decreases the number of decisions made by a software developer, without losing flexibility.

In Maven, there are many conventions for setting up the project, building the artifacts, running unit tests and releasing the code. These conventions lead to common process for Software development.

In case of other tools, there are a lot of configuration options are present. But most of the time, a developer uses same set of configuration options. So it is better to make these as a default options. Maven uses default options from best practices and provides right conventions for Software development.

26. What are the different types of profile in Maven ? Where will you define these profiles ?

Answer: In Maven, we can have following types of Profile:

Per Project

It is defined in the POM itself (pom.xml).

Per User

We can define it in the Maven-settings (%USER_HOME%/.m2/settings.xml).

Global

It is defined in the global Maven-settings (${maven.home}/conf/settings.xml).

Profile descriptor

Descriptor is located in project basedir (profiles.xml) (It is not supported in Maven 3.0)

27. What are the differences between Ant and Maven ?

Answer: Key differences between Ant and Maven are:

- Ant is a Java library and command line toolbox for build process. Maven is a framework for many aspects of software development like- project setup, compile, build, documentation etc.
- Ant does not have any conventions for project structure or build processes. Maven has conventions for setting up project structure as well as for build processes.
- Ant is based on procedural programming. We have to write code for compilation build, copy etc. Maven is based on declarative programming. We have to just configure it for our project setup and programming.
- Ant does not impose any lifecycle. We need to create the sequence of tasks manually. Maven has a lifecycle for software build processes. There are well-defined phases that we can use in Maven.
- Ant scripts are not reusable in multiple projects. Maven has plugins that are reusable across multiple projects.

28. What is the difference between a Release version and SNAPSHOT version in Maven ?

Answer: A SNAPSHOT version in Maven is the one that has not been released.

Before every release version there is a SNAPSHOT version. Before 1.0 release there will be 1.0-SNAPSHOT. If we download 1.0-SNAPSHOT today then we may get different set of files than the one we get on downloading it yesterday. SNAPSHOT version can keep getting changes in it since it is under development.

But release version always gives exactly same set files with each download.

29. How will you verify if Maven is installed on Windows ?

Answer: To check this, type mvn –version in cmd prompt of Windows. This will give you the version of Maven installed on Windows.

30. How does Maven looks for a dependency or resource ?

Answer: It refers to the settings.xml to look for the repositories to look for the resource. First It looks into the configured local repository, then it looks into the configured Remote repositories. If the resource is still not found , it looks it within maven repository central i.e repo1.maven.org. If its still not found, it throws the exception saying “Unable to find resource in repository central”.

31. What is maven repository central ?

Answer: Its the repository provided by Maven. In case your POM specify the dependencies and its not available in the configured local and the remote repository. It then looks for the resource in Maven Central. Maven provides most of the generic dependency resources at this remote location.

32. What would you do if you have to add a jar to the project using Maven ?

Answer: If its already there in Maven local repository, We can add that as a dependency in the project pom file with its Group Id, Artifact Id and version.

We can provide additional attribute SystemPath if its unable to locate the jar in the local repository.

If its not there in the local repository, we can install it first in the local repository and then can add it as dependency.

33. Have you ever had problem getting your projects in eclipse refreshed after you made changes in the Pom files ?

Answer: Yes, It happens many times but I would usually perform mvn eclipse:eclipse and this would resolve the project refresh problems.

34. What is the difference between compile and install ?

Answer: Compile compiles the source code of the project

whereas

Install installs the package into the local repository, for use as a dependency in other projects locally

35. What is a cyclic dependency ?

Answer: A has dependency of B, B has dependency of C and C has dependency of A.

With Maven 2 , came transitive dependency wherein in above scenario, C will acts as a dependency of A as if this dependency has been defined directly in A but the negative side is that if it leads to cyclic dependency , it creates problems.

36. What is a Test Dependency Scope in Maven ?

Answer: This scope indicates that the dependency is not required for normal use of the application, and is only available for the test compilation and execution phases. This scope is not transitive.

37. What is the POM packaging in Maven ?

Answer: Pom packaging is simply a specification that states the primary artifact is not a war or jar, but the pom.xml itself.

38. Difference between Maven Project and Module ?

Answer: Maven Module has a Parent whereas Project doesnt. when we add the parent section to the pom file, it adds the module section to the parent project pom file. When we execute mvn compile / install, it basically checks that module section of the parent to identify all the modules that needs to be compiled first.

39. How to specify the sequence in which sub modules needs to be built ?

Answer: By specifying the modules in the same sequence in the parent pom.

40. What is the best practice configuration usage for files – pom.xml or settings.xml ?

Answer: The best practice guideline between settings.xml and pom.xml is that configurations in settings.xml must be specific to the current user and that pom.xml configurations are specific to the project.

41. How can I change the default location of the generated jar when I command "mvn package" ?

Answer: By default, the location of the generated jar is in ${project.build.directory} or in your target directory. We can change this by configuring the outputDirectory of maven-jar-plugin.

42. How would you see the version of Maven ?

Answer: mvn –version

43. How can we see Dependencies for the project and where exactly they are defined ?

Answer:

Using

mvn dependency:tree

44. What are the benefits of transitive depency in Maven ?

Answer: Transitive dependencies allows to avoid specifying the libraries that are required by the project which are specified in other dependent projects – Remote or Local.

45. What is the difference between Maven, Ant and Jenkins ?

Answer: Maven and Ant are Build Technologies whereas Jenkins is a continuous integration tool.

46. Have you heard of Ban Duplicate Classes Maven enforcer plugin ? What is its use ?

Answer: Yes , we have been using this plugin with our projects and its purpose is to warn and stop the Build if there are duplicates of the same package and class are being carried either directly or through transitive dependencies. the duplicate could be coming through different types of dependencies or through different versions of the same dependency. Its purpose is to make sure that there is only one copy thats being used at compile time and runtime and hence shouldnt later result in runtime problems.

47. How to tackle duplicate classes in maven build ?

Answer: The simplest way is to ignore them if Maven enforcer plugin is complaining about it but it may lead to runtime problems later.

We can do the dependency:tree to see from where these duplicate ones are coming and hence can exclude the duplicate one.

48. What is Archetype ?

Answer: Archetype is a Maven plugin which has the task of creating a maven project structure.

49. What means SNAPSHOT in Maven ?

Answer: SNAPSHOT is a type of version that indicates a current deployment copy. Maven checks during each build for a new SNAPSHOT version in the remote repository.

50. What is the difference between version and SNAPSHOT ?

Answer: Maven will download always the specified version. In case of SNAPSHOT Maven wil download the latest SNAPSHOT.