1. What is .pem file?

* It stands for privacy-Enhanced Mail.
* It is a base64 container format for encoding keys and certificates.
* Pem download from AWS when you created your key pair.
* This is only a one time download and you cannot download it again.
* Pem files are commonly used for connecting to EC2 instances.
* The .pem file will see in downloads folder.

1. How can we identify the build got successes?

* Mvn clean install or mvn clean package

1. Where will be the source code present?

* The source code for a maven project is located in the src/main/java directory for the main application code.

1. Who will write the source code?

* The source code is primarily written by software developers.

1. When we generate var/jar files in target file?

* In a maven project war/jar files are generated during the build process, specifically during the packaging phase.
* With the package phase being crucial for creating jar/war files.

1. Where will be the build files stored?

* Stored in target directory.

1. What is the home directory of Maven?

* /usr/share/apache-maven

1. What is meant by build tool?

* A build tool is a software application that automates the process of building and managing software projects.
* It converting source code into executable programs or libraries.
* It manages various tasks involved in the development lifecycle, including compiling code, linking libraries, running tests, packaging applications, and deploying them.
* Key functions of build tool:
* Compilation: converts source code written in programming language into machine code or bytecode that can be executed by a computer.
* Dependency management: manages external libraries and dependencies required by the project.
* Packaging: compiled code and resources into distributable formats(jar/var/ear).
* Testing: automates the execution of unit tests, integration tests, and other types of tests.
* Code quality checks: checks the code quality standards.

1. Explain the process of building?

* The building process in software development refers to the steps involved in converting source code into a runnable application or library.
* This process typically includes compiling, linking, testing, and packaging the code.
* Source code preparation: developers write code in a programming language using an integrated development environment.
* Dependency management: the build tool identifies external libraries and frameworks required by the project.
* Compilation: the source code is compiled into an intermediate form bytecode or machine code.
* Linking: the compiled code is linked with necessary libraries and resources to create a single executable or library.
* Testing: the build process often includes running automated tests.
* Code quality checks: it checks code standards.
* Packaging: the final output is packaged for distribution like war/jar/ear.
* Deployment: the built application is deployed to various environments (development, staging, production)
* Continuous integration and continuous deployment: the building process is often integrated into CI/CD pipelines, automating the steps from code commit to deployment.

1. Does maven support all types of files?

* Primarily it supports java files.
* Supported files (.java,. json, pom.xml,.xml)
* It can also handle other file types indirectly through plugins or extensions.
* Ex: html, css, js.
* With the right plugins, maven can support other languages (ex: scala, groovy)
* To manage other file types or languages specific plugins may be needed.

1. For python project what is the build tool to use?

* Pybuilder is a python-based build automation tool that is primarily used for managing and automating various tasks in python project. It provides a framework for building, testing, and packaging python applications.
* Key features:
* Build automation
* Dependency management
* Testing integration
* Code quality checks
* Packaging
* Custom task definitions
* Integration with CI/CD

1. Difference between compile and validate?

* Validate: validates the project is correct and all necessary information is available. It checks whether java installed or not, testing is done or not.
* Compile: compile the source code of the project.

|  |  |  |
| --- | --- | --- |
| Feature | validate | compile |
| purpose | Check project configuration and structure. | Compile the source code into bytecode. |
| Actions performed | Validates pom.xml and project setup. | Compiles java source files to .class files. |
| outcome | Stops the build if validation fails. | Stops the build if compilation fails. |
| Location | First phase in the maven life cycle. | Second phase in the maven life cycle. |

1. Can we create one jar file or multiple jar files?

* Depending on our project we can create single or multiple jar files.
* By default, maven create single jar file for our project.
* Creating multiple jar files:
* If multi module maven project where each module can produce its own JAR file.

1. What is maven and uses?

* Maven is a powerful build automation and project management tool primarily used for java projects.
* It was developed by Apache community.
* It was created in 2003.
* It has build life cycle and different types of plugins. built in plugins are not reusable but adding plugins are reusable.
* It follows “convention over configuration approach”.

Uses of maven:

* Dependency management: automatically downloads and manages project dependencies from a central repository.
* Build automation: automates the build process, including compiling source code, running tests, and packaging the application.
* Project structure: it gives brief project structure.
* Plugin system: add more functionality for tasks such as code analysis, documentation generation and deployment.
* Multi module projects
* Cross platform compatibility: works on any platform that supports java, making it a versatile tool for diverse development environments.

1. Explain the pom file in maven.

* Pom means project object model.
* In maven it is a fundamental part of a maven project.
* It is an xml file named pom.xml that contains information about the project and configuration details used by maven to build the project.
* Key points of pom.xml:
* Groupid: it specifies the id for the project group.
* Atrifactid: feature of groupid and project name
* Packaging: it defines the packaging type such as jar, war and ear.
* Version: the version of the project.

1. What are maven coordinates and what they represent?

* Maven coordinates are a set of identifiers used to uniquely a particular artifact (like a library, project, or module) in a maven repository.
* Maven coordinates consist of four main elements:
* Groupid
* Artifactid
* Version
* Packaging
* Scope.

1. How do you manage dependencies in maven?

* Managing dependencies in maven is a fundamental aspect of using the tool effectively. maven provides a structured way to specify, retrieve, and manage libraries and other resources that your project needs.
* Defining dependencies: dependencies are defined in the pom.xml file of your maven project.
* Dependency scopes
* Transitive dependencies
* Excluding transitive dependencies
* Dependency management section
* Using repositories
* Updating repositories

1. Explain the concept of maven life cycle?

* Different phases in maven life cycle:
* Validate: validates the project structure and configuration.
* Compile: compiles the source code of the project. The compiled code is typically placed in the target/classes directory.
* Test: runs unit tests against the compiled code.
* Package: packages the compiled code into a distributable format (ex: war/jar). The packaged artifact is placed in the target directory.
* Verify: to verify the quality of the package. Ensure that the package is valid and meets quality standards.
* Install: installs the packaged artifact into the local maven repository.
* Deploy: deploys the packaged artifact to a remote repository.

1. What are maven goals and how do you refer them phases?

* Maven goals and phases are fundamental concepts in the maven build lifecycle, serving different purposes in the build process.
* Goals are specific tasks that maven can execute.
* Each goal represents a single task that contributes to the overall build process.
* Goals can be executed independently or as part of a larger lifecycle phase.
* Maven goals:
* Compile: compiles the source code of the project.
* Test: runs the tests for the project.
* Package: packages the compiled code into a distributable format(war/jar/ear)
* Install: installs the package into the local maven repository.
* Deploy: copies the final package to a remote repository for sharing with other developers.
* Each phase in the maven lifecycle associated with one or more goals. when you execute a phase, maven runs all the goals associated with that phase and all preceding phases.

1. How do you create maven project?

* Select file> new> project> maven > maven project.
* Click next
* Select create a simple project
* Click next
* Enter or select values for the following fields: groupid, artifactid,version ,packaging, etc.
* Enter parent project values.
* Click finish.

1. How do you handle versions in maven project?

* Handling version in a maven project is crucial for managing dependencies, ensuring compatibility, and maintaining the stability of your application.
* Specifying versions in pom.xml
* Using property for version management <properties>
* Dependency management section <dependency management>
* Version ranges
* Using snapshot versions and update versions.