**Maven introduction:**

**Meaning of maven:**

* It is an open-source build automation tool.
* Project management for java applications
* Primarily used for java applications.
* Automates the source code compilation, depecdency management, assembling of binary codes into packages and executes the test scripts.
* It follows a conversion over configuration approach.
* It provides default project structures and lifecycle phases and reduces manual configurations.
* Firstly developers, develops the code, then pushes to the GitHub (repo), then devops engineer will turn the code into packages by using build tools, deploys into the server, the end product will be accessed by clients/end users.
* Maven developed by Apache tomcat.
* Packages of maven:
* War: web applications
* Jar: Java applications
* Ear: enterprise applications.
* Build tools:
* Java – maven/ANT
* Python- pybuilder
* .net -MS build(Microsoft build engine)
* NodeJS-Gulp, Grunt, and Webpack.
* Difference between Apache Maven and Apache ANT

|  |  |
| --- | --- |
| **Apache maven** | **Apache ANT** |
| Convention over configuration (standardized project structure) | Configuration over convention  (manual setup of tasks) |
| Built-in repositories management with repositories. | No built-in repository management requires management tools. |
| Uses pom.xml, which defines project dependencies and life cycle. | Uses build.xml where tasks and dependencies must be explicitly defined. |
| Defines a standardized build life cycle. | No pre-defined build life cycle, developers must define the tasks manually. |
| Plug-in based architecture. | Task based architecture. |
| Works well with CI/CD pipelines, GitHub actions. | Works well but requires additional setup for dependencies. |
| Limited flexibility due to its opinionated nature.  Best for large projects. | Highly flexible, allows custom build process.  Best for building customized projects. |

* **Plugins:**
* Maven plugins are used to extend and customize the build lifecycle of a maven project.
* Installing the dependencies from externally/ additionally is known as ‘plugins’.
* **Types of plugins:**
* Build plugins: automate the build process.
* Testing plugins: facilitate testing of the code.
* Reporting plugins: used to generate reports on various aspects of the project.
* Dependency management plugins: manage project dependencies and their configurations.
* Deployment plugins: manage the deployment of artifacts to repositories.
* Code quality plugins: analyze code quality and enforce best practices.
* Custom plugins: allow developers to define custom tasks specific to their needs.
* Lifecycle plugins: integrate with the maven build lifecycle.
* **Types of maven repositories:**

There are three types of maven repositories:

1. **Local repository:**

* A local repository is a directory on your machine where maven stores all the artifacts that you download or build.(stored on your own machine).

1. **Central repository:**

* The central repository is a large, well-known repository maintained by the maven community, hosting a vast number of commonly used libraries and plugins.

1. **Remote repository**:

* Remote repositories are any repositories that are not local and can be accessed over a network. They can be public or private. (external repositories).