REPORT

Mar 11, 2025 P. Kurella

MAVEN:

Maven is an open-source build automation tool that is primarily used for Java-based applications and project management.

- It simplifies project setup, dependency management and builds lifecycle management.

Why use Maven?

- **Dependency Management:** It handles external libraries automatically. No need to manually download JAR files.
- **Build Automation:** Compiles, tests, packages, and deploys code in a single command.
- **Standardized Project Structure:** Uses a common directory layout for easy collaboration.
- **Plugin Support:** It is Extensible via plugins for additional functionality (e.g., Docker, Spring Boot).
- **Integration:** Works well with IDEs like IntelliJ, Eclipse and CI/CD tools like Jenkins and GitHub Actions.

What is a build tool in DevOps?

A build tool is a software program that automates the process of transforming the source code into a deployable and executable format.

- Maven was developed by Apache company.
- It is used for Java-based projects/applications.
- As a build tool, it automates code compilation& dependency management, assembles binary codes into packages, and executes test scripts.

Build Tools:

1. **Java:** Apache-Maven, Apache ANT (older version).
2. **Python:** Pybuilder.
3. **.Net:** MSbuild (Microsoft Build Engine)**.**
4. **Node.js:** Gulp, Grunt, Gradle, Web Pack.

Process/Workflow:

Developers → Code → Push → GitHub(Remote Repo) → Build (Maven) →Packages → war/jar/ear → Copy packages to the web server (Tomcat) → Eu (Access the application).

How Does Maven Work?

Pom.xml → Build → war/jar/ear → Packages (deployable format) → Copy Tomcat (Web server)

- **EAR:** Enterprise Application Archive.
- **WAR:** Web Applications Archive.
- **JAR:** Java Applications Archive.

→ POM.xml:

- POM.xml → Project Object Model .xml
- Maven works for POM.xml/POM.xml2.
- Developers will write the Pom.xml file.
- Pom.xml is used to manage the java applications.
- Pom.xml → Super POM / Parent POM.
- It contains all the dependency libraries.
- It should be unique.
- Each project contains one Pom.xml file.

→ Plugins:

Dependencies (external features) required for a project are known as plugins.

- There are two types of plugins:
  - Inbuilt plugins: one-time use.
  - Added plugins: reusable.
- Added plugins are reusable.
- It has a life cycle.

→ Repositories:

1. **Central-Repo:** Online repo → Github
2. **Remote-Repo:** Organization repo → Realtime(eg:IBM)
3. **Local-Repo:** Repositories in our own laptops and local machines.

→ Maven in DevOps should be utilized in 3 scenarios:

1. If the initiative has several significant dependencies.
2. If the dependencies version needs to be upgraded frequently.
3. If the task involves rapid documentation, compilation & building of the source code as a jar or zip files.

→ Apache ANT:

- An older version of Maven.
- Developed by Apache.

- Has no life-cycle.
- Build.xml → by developer.
- J-unit (Java unit) test cases are not there in ANT.
- Scripts are not reusable.

Maven Life-cycle:

The 3-step process of Maven includes the following:

1. **Default:** It takes the code from developers and performs a few functions.
    a. **Compile:** Compiles the entire source code.
    b. **Validate:** Validates the complied source code.
    c. **Test:** Run the J-unit test cases.
    d. **Package:** Create a software package.
    e. **Install:** Install all the generated packages.
    f. **Verify:** Verify the generated package.
2. **Clean:** Performed before compilation.
    a. **Pre-Clean:** Check for JAR/WAR/EAR.
    b. **Clean:** Delete the older JAR/WAR/EAR files.
    c. **Post-Clean:** The newly generated WAR/EAR/JAR files will be saved.
3. **Site:** It is like a folder where we will deploy our applications.
    a. **Pre-Site:** It receives the post-clean files.
    b. **Site:** It receives the pre-site files.
    c. **Post-Site:** It receives the files from the site.
    d. **Site:** Deploy to where we need to copy files.
4. Generates a comprehensive project documentation to copy the files and metrics(exact values) for better understanding & collaboration between team members.