One of the devOps tools for better project management. Maven is written in Java and is used to build projects written in C#, Scala, Ruby, etc. Based on the Project Object Model (POM).

**Handles**:

Builds, Documentation, Dependencies, Reports, SCM, Distribution, Releases, Mailing list, Download dependencies.

**POM**

Contains the project information and configuration details for Maven to build the project. It contains information such as dependencies, source directory, a build directory, plugins, goals, etc. Maven reads the pom.xml file and executes the desired goal.

When maven is executing goals and tasks maven searches for POM.XML in the current directory. It reads configuration from pom file and executes the desired goal.

This build tool compiles and packages the code into an executable form (Jar/War).

A Maven plugin refers to the group of goals that may or may not be in the same phase.

**Phases of the build life cycle:**

Compile, Test-Compile, Test, Package, Integration-Test, Verify, Install, Deploy

Pre-site, Site, Post-site, Site-deploy.

Two types of Maven Plugins:

**Build plugins** – These plugins are executed during the build and are configured in the <build/> element of pom.xml.

**Reporting plugins** – These plugins are executed during the stage generation and are configured in the <reporting/> element of the pom.xml.

The order of inheritance in Maven is:

Settings (settings.xml?), CLI parameters, Parent POM, Project POM.

The different types of dependency scopes are:

Compile, Provided, Runtime, Test, System, Import.

**Primary Artifacts**:

These are like the primary dependencies which will have their own pom and further dependencies.

**Secondary Artifacts**

secondary artifact is not expected to have a remote pom and has thus never has any dependencies. With an extra tag </classifier>

To verify the downloaded artifacts Maven computes the md5 and sha1 checksum for that artifact and compares it to the values found in the checksum files located at $ARTIFACT\_URL.md5, or $ARTIFACT\_URL.sha1, respectively.

**Snapshot**:

The idea is that you can continuously push your latest changes to 3.0.0-SNAPSHOT and anyone depending on it will get the latest changes every time they build their project. Then, after a few iterations, and everyone is happy with the latest state of 3.0.0-SNAPSHOT, it can be permanently released as 3.0.0, and rapid development can continue 3.0.1-SNAPSHOT.