Maven

Apache Maven is a software project management tool which can manage project's build, reporting, and documentation and is used primarily for java projects. There are several other build tools in the market but maven is on top of that.

Lets install maven,

```
# yum install maven -y
```

Generally, developers decide which build tool will be used in the project and this happens before they start coding. Basically, they first create a directory structure of the project including pom.xml file.

Create a project directory,

```
# mvn archetype:generate -DgroupId=com.mycompany.app \ -
DartifactId=my-app \ -DarchetypeArtifactId=maven-archetype-
quickstart \ -DarchetypeVersion=1.4 -DinteractiveMode=false
```

After executing the above command, you will see a directory having the same name as that of artifactid. And also a hierarchy of directories are created along with the pom file. 'pom.xml' file is a project object model file which contains the majority of information required to build the project such as dependencies and other configurations.

Above command contains a goal '`archetype:generate`' and passes some parameters to that goal. Basically we work with Maven using Maven phases and Maven goals. Let's discuss some terminologies used in Maven:

Maven Build Lifecycle - The Maven build follows some lifecycle to deploy the project. There are three built-in lifecycles,

- default - the main lifecycle as it is responsible for project deployment
- clean - it cleans the project and its files generated by the previous build
- site - it generates the project site's documentation

Each Build lifecycle contain several phases, we will look into some important phases here,

Maven Phases - A Maven phase represents a step in the build lifecycle. Each phase is responsible for a specific task. Whenever you define a phase, maven will execute every phase in the sequence up to and including the one defined. Following are some phases in Maven default lifecycle,

- validate - validate the code
- compile - compile the source code
- test - test the compiled source code using suitable testing framework
- package - wrap the compiled code and create an artifact
- integration-test - process and deploy for integration testing
- verify - run quality analysis
- install - install package into local repository
- deploy - put final package to the remote repository

Maven Goals - Each phase is a sequence of goals, and each goal is responsible for a specific task. When you execute a phase, all the goals that are bound to the phase execute in sequence.
To list all goals bound to the specific phase,

```
# mvn help:describe -Dcmd=<PHASENAME>
```

Maven Plugins - Maven plugin is a group of goals, you can imagine it as a library of the goals.
To list all goals from specific plugin,

```
# mvn <PLUGINNAME>:help
```

Now lets build the project,

```
# mvn clean dependency:copy-dependencies package
```

You can execute phases and goals in sequence as used above. It will clean the project, copy dependencies, and creates the artifact.

Practical 1: Pull project from git repository, make build using maven, and deploy on tomcat server using Jenkins CICD pipeline.