**Maven and Jenkins Integration**

* Maven is a build management tool and software project management for Java projects.
* This is to achieve standard way to build the projects, a clear definition of what the project consisted of, an easy way to publish project information and a way to share JARs across several projects.
* Without importing JARs, all dependencies can be added in POM.xml. It’s a central repository to get dependencies
* It maintains common structure across the organization
* It had flexibility in integrating with CI tools
* Plugins can be used for Test framework execution
* Profile plugin can be used to run different testing.xml files
* Surefire plugin is used to run the test cases in Maven.

**Maven Terminology:**

*GroupID:* This defines the project uniquely across all projects and when a maven project is created, GroupID will be created as a package

ArtifactID: This s the actual maven project we work on and it is the JAR that gets deployed to the Maven repository

Archetype-generate: This is used to create a maven project

**How to create a Maven Project from cmd prompt:**

Mvn archetype-generate -Dgroupid=packageName DartifactID=projectName DarchetypeArtifactID=maven-archetype-quickstart DinteractiveMode=false

**DarchetypeArtifactID=maven-archetype-quickstart** is used to create the standard folder structure across the organization.

**Importing Maven project to Eclipse:**

To import the project in Eclipse, .classpath and .project files needs to be created. Below is the cmd for the same

Mvn eclipse.eclipse

Now import the project to eclipse by selecting Existing Maven project

Check the maven repository in local by navigating to windows>Preferences>Maven>Usersettings(Path on local can be seen) on eclipse

Maven Clean install can be done to clean the project by giving **mvn clean** command

To compile the code(syntax validations), the command is **mvn compile**

**Mvn test** isto check the jars in the local which are imported from maven repository

For the first time, the jars are imported from repository and once all are imported, everytime the maven test runs by picking the jars from local

Each java class file needs to have the word test as per maven algorithm to run the test

**Integrate TestNg with Maven:**

* Add TestNG dependency in POM.xml
* Allows flexible configuration to run tests through tetsNG xml.
* When a configuration of testing xml is given in surefire plugin, the tests run from testing xml.

To run the single test from maven, command is

-Dtest=testName test

Different testing xmls can be created to differentiate smoke/regression/test in current build

<profiles>

<profile>

<id>Regression</id>

<Build>

…..

<Configuration>

…testing.xml

</Configuration>

…

</Build>

<profile>

<profile>

<id>Smoke</id>

<Build>

…..

<Configuration>

…testing.xml

</Configuration>

…

</Build>

<profile>

</profiles>

To run the profile from command prompt use

Mvn test –Pregression

**Jenkins**:

* It is a continuous integration tool which triggers the job as per the scheduled time.
* Inside job, we have the tests to run and on clicking Build Now, all the tests can be executed at once.
* Settings needs to be done so that the version of jars in local and Jenkins should be same.
* Install Jenkins and run the war file to open the jenkins

Java –jar Jenkins.war httpPort=8080

* When the Maven project is in local, then we have to add the project in Jenkins home directory or if it is from GIT/SVN, then the repository path should be given so that Jenkins can trigger the code from the repository
* While creating a Job in Jenkins, user can go to source code management and select GIt, give the repo URL to trigger the code.
* Job is the Project name in Jenkins
* Under Build Management>Build Periodically and give the time:
  + - \*7\*\*\* which is at 7 Am is the meaning

|  |  |
| --- | --- |
| MINUTE | Minutes within the hour (0–59) |
| HOUR | The hour of the day (0–23) |
| DOM | The day of the month (1–31) |
| MONTH | The month (1–12) |
| DOW | The day of the week (0–7) where 0 and 7 are Sunday. |

To specify multiple values for one field, the following operators are available. In the order of precedence,

* \* specifies all valid values
* M-N specifies a range of values
* M-N/X or \*/X steps by intervals of X through the specified range or whole valid range
* A,B,...,Z enumerates multiple values
* Use custom workspace is to give the project name as $(JENKINS\_HOME)Projectname
* Manage Jenkings>install TestNgresults plugin>download and restart Jenkins
* Post-Build-Action>Publish testNGReports>Save, By this TestNG reports will be generated when a build is executed
* In eclipse, we can see surefire reports with testing XML that can be connected to Jenkins

CI CD pipeline process to be studied