**Build tool:**

Build tool basically does following automation

* Build tools creates deployment folder structure (development tool might have diff folder types for developing. build tool might have diff folder types for building………………..
* Our code might depend on external dependencies, build tool must make sure those dependencies are in class part.
* Compile source code
* It also can run test cases on your code
* Package

By executing single command all the above tasks are performed by build tool.

**Pom.xml**: it is a configuration file used by maven to perform all build related activities without pom.xml maven will not work.

**21-10-19**

**Creating maven based** java **project :** ( developers will do it)

Installing maven on windows

Maven is written on java so JVM is a dependency

Maven also compile java code it’s also require JDK

Install Java development kit on windows (JDK)

Install Maven (<http://apachemirror.wuchna.com/maven/maven-3/3.6.2/binaries/apache-maven-3.6.2-bin.zip>)

Unzip it

We need to add maven to the path and we should tell maven where java is installed

Adding maven to the path

Go to start menu – search for edit environment for system variables – one dialog box appears – click on environment variables

Path – there paste the path of bin folder of extracted maven folder -ok-ok

Go to environment variables – click on Java-Home where java is available , there paste the path of java folder (c disk – program files – java we can see jdk1.8.0\_231 – open and paste that path in java\_home variable.

Maven dependency manager:

My java guy wants to read data from excel sheet they will find the right dependency

POI is a dependency ( maven dependency is an external JAR file which we wanna use in our project)

Paste the depencency in pom.xml

Who manages dependencies – Maven

We have to declare what dependency we want in Pom.xml such that maven automatically downloads dependencies.

**From where depencencies are downloaded :**

Maven uses diff repositories

* Central repository
* Local repository
* Remote repository

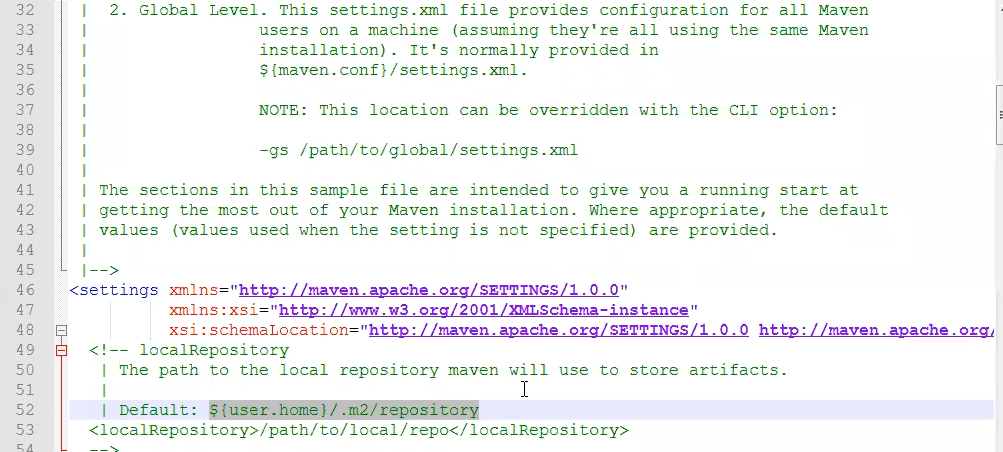
**Central repository** : it’s a repository maintained by maven itself over internet (server is maintained by maven, who upload dependencies in to that - organisation maintains dependencies, ex :apache maintains it.)

**Local repository** : it is a repository on local machine i.e the machine where we are running maven build commands.

IQ): What is maven local repo path ?

C:/users/beerelokesh/.m2/repository (C:\Users\Beere Lokesh\.m2\repository)

If we want to change path change path there (( C:\apache-maven-3.6.2\conf -seeting.xml page)



**Remote repository**:

Remoterepository is a server which is maintained within company N/W.

* Security
* Company might have its own framework which is internal to company those dependencies can be managed by remote repository

Maven has **setting.xml** it refers to central repository (C:\apache-maven-3.6.2\conf), if you want to point to remote. we have to customize, remove central and place remote

Group id represents company (if we are developing for HCL then HCL)

Project version followed by project name



One would be syntaxial issues

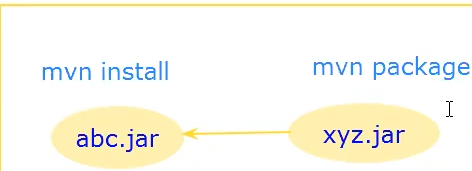
Other keep things n wrong position

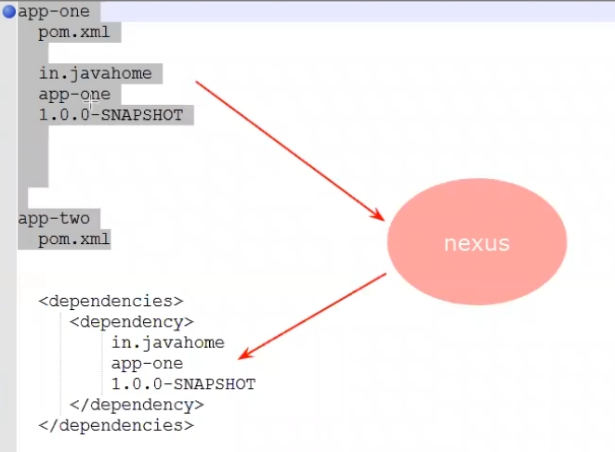
Tags which s not a/c to your schema

**IQ) what is maven build life cycle: imp ( VCTPVID)**

* Validate: it validates pom .xml
* Compile: it compiles source code (to compile source code it requires dependencies)
* Test: Executes **Junit** test cases. (
* Package: it is going to package your software by creating jar or war or ear
* Verify: To verify integration test results
* Install: copies package into local repository

(We have 2 projects xyz proj depends on abc proj (both are my projects) if we want to build xyz proj we need abc, I run mvn install here , it validates, compile, test ,package and copy that package to local repository then I will use mvn package. In my pom.xml of xyz I say group id, artifact id, dependency, version all that , it picks the jar file from local file so when you have dependency modules or proj install that such that your jar file dependency will be available in local repo that is used by other project then only we use install)





* Deploy**:**  it uploads artifacts to remote repository

If we have 2 projects, if 2 projects are inside same machine, you are building them together and depend on each other and we use install and we keep it on local

Let say I am working on machine 1 and you are working on machine 2 then you depend on my dependency, then requires we keep it in remote. I will deploy my packge to remote. Then maven downloads package from remote.

**22/10/19**

**IQ) How to skip Junit test cases in Maven ?**

We have two ways to do that

* mvn package -Dmaven.test.skip=true

( developpers write a code ,some times test cases will fail , we can skip that and testing team will test that )

**Maven target folder ?**

(The target folder is created when we run maven commands)

Target folder is o/p folder of Maven, it is created when we run maven build commands, it is safe to delete (because it can be regenerated).

mvn clean package ( delete the target package)

mvn test

mvn package

mvn install

mvn deploy ( upload artifacts to remote repository )[ we need to configure nexus \*D ………..

**-------------------------------------- Sonar type Nexus ----------------------------------**

* It is used for maven remote repositories

Iit stores artifacts (after build what we get called artifact - (jar,war,ear (enterprise application archive) ) , artifacts are nothing but package format of our applictions

**IQ)** which artifactory you are using in your project?

* Sonartype Nexus.

**INSTALL AND CONFIGURE SONARTYPE NEXUS**

* It is written in java , jvm is a dependency
* Install and configure nexus on linux machine
* Connecting to Ec2
* Mobaxterm – it is used for connecting to remote servers like putty.
* Install sonartype nexus
* For installing and
* Install package on linux machine (Sudo yum install java | grep java-1.8.0

**23-10-19**

**Agile metholodology**

Scrum :

Sprint

2 to 4 weeks

Standup metting

Scrum master : to make sure every scrum member is

We need to give status in the standing meeting regarding what we have done and we are going to do.

Responsible for delivering the product on time

We will get the requirement from scrum master , scrum master interacts with product owner – scrum master interacts with BA guy,

BA should have idea on functional and technical knowledge.

Project management tool /bug tracking tool:

Jira – not an open source , owner is Atlas

Environments :

* Dev -----------------🡪Developers – we have end-to -end access once developers give sign off then it will move to test
* Test------------------🡪 we won’t give access to developers for test e…nvironment db.(it is completely for testers)[tester sign off required]
* Preprod/Uat-----🡪 if your code is working in one environment and not working In another environ means there will be configuration issue or DB issue. ( can be used by testers or BA or client ) [ product sign off required]
* Prod/live --------🡪

If we want to move from one env to another sign off required

Code mgmt tools :

SVN

GIT

Jar/war/ear -------🡪 packaging

For stand alone app we can use Jar

War is used for web application

Ear is used for e-commerce appliocation

1. Build

2. Deployment

Build tools : responsibility is creating a jar /war/ear

* Maven
* Ant
* Gradle

Deployement:

* Stop server
* Take backup existing Jar
* Copy the jar file in to the server
* Before starting any configurations are required we can do it
* Start the server

**Day12 24-10-19**

<https://github.com/javahometech/nexus/blob/master/README.md>

**Setting up Nexus on AWS Linux**

**SSH into Linux server**

Using putty or MobaXterm or any ssh client

**Install java**

sudo yum install java-1.8.0-openjdk -y

**Install Nexus 3**

cd /opt/

sudo wget https://download.sonatype.com/nexus/3/latest-unix.tar.gz

Untar the file

sudo tar xvf latest-unix.tar.gz

Change the ownership

sudo chown -R ec2-user:ec2-user nexus-3.19.1-01 sonatype-work

**Run nexus as a service**

**1 Change user for nexus**

open bin/nexus.rc and make sure the following line is present

run\_as\_user="ec2-user"

**2 Execute following command**

sudo ln -s /opt/nexus-3.19.1-01/bin/nexus /etc/init.d/nexus

cd /etc/init.d

sudo chkconfig --add nexus

sudo chkconfig --levels 345 nexus on

sudo service nexus start

**Loging to Nexus3 using browser**

http:public-ip:8081

and follow instructions to get access to nexus

**Store artifacts into nexus**

We are going to use following repositories to store artifacts

Nexus settings --> repositories

* maven-release
* maven-snapshots

**Under Maven configure Nexus details**

**1. Configure nexus user/password.**

Open maven settings($MAVEN\_HOME/conf/settings.xml) file and add following snippet

<servers>

<server>

<id>nexusRepo</id>

<username>admin</username>

<password>javahome</password>

</server>

</servers>

**1. Configure pom.xml of your project**

Make suer the following snippet exists in pom.xml

<distributionManagement>

<snapshotRepository>

<id>nexusRepo</id>

<url>http://13.233.230.166:8081/repository/maven-snapshots/</url>

</snapshotRepository>

<repository>

<id>nexusRepo</id>

<url>http://13.233.230.166:8081/repository/maven-releases/</url>

</repository>

</distributionManagement>

[(LDAP:

Server to place users

If I want to have access git hub , or laptop or Jenkins those are integrated to LDAP then we get the acess)]

**Store artifacts into Nexus:**

We are using existing repositories

Nexus settings page – repositories- create repository – maven2(hosted)

-maven-release

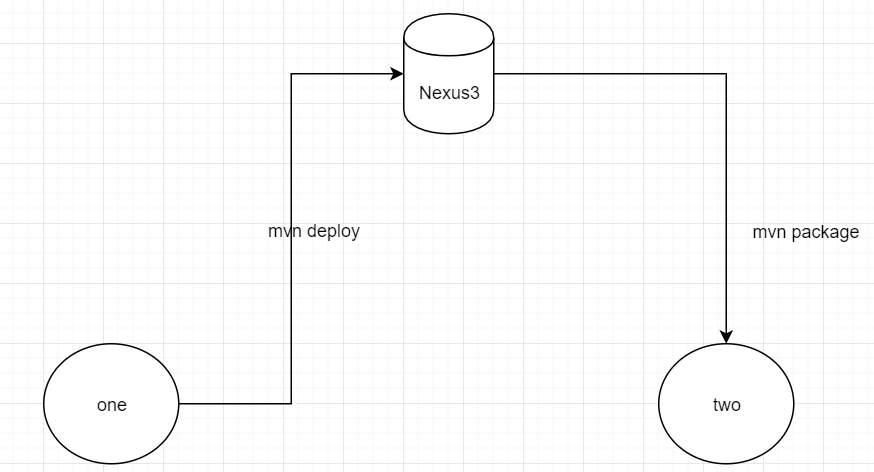
-maven-snapshots

### unser maven configure Nexus details

1. Configure Nexus user/password.

Open maven settings file and add following snippet

**25-10-19**



Git sources source code

Nexus is the code having ready to deploy

We can tag source code versions I can keep that in git

We can build, package by nexus all deployments happen through nexus.

Nexus will have all versions, if present version fails we can roll back to previous version, we no need to go to source code(git)

Versioning the binary and give it to the cu stomer but not source code

=========================================================================