DevOps

++++++++++

DevOps is a culture

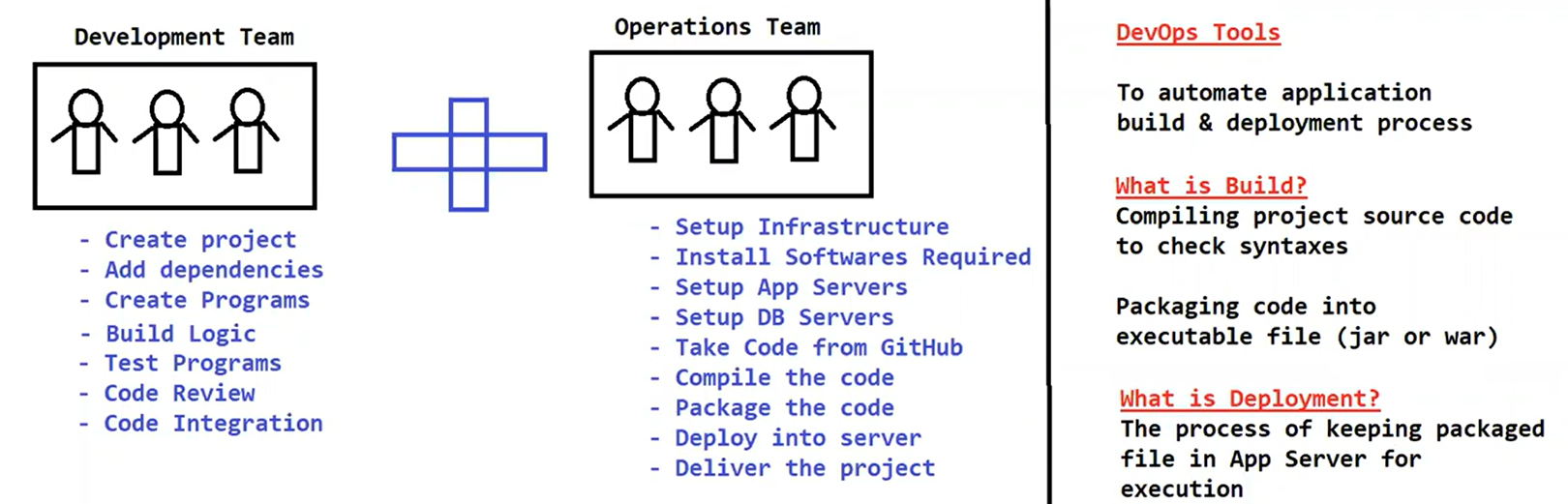
DevOps is a process

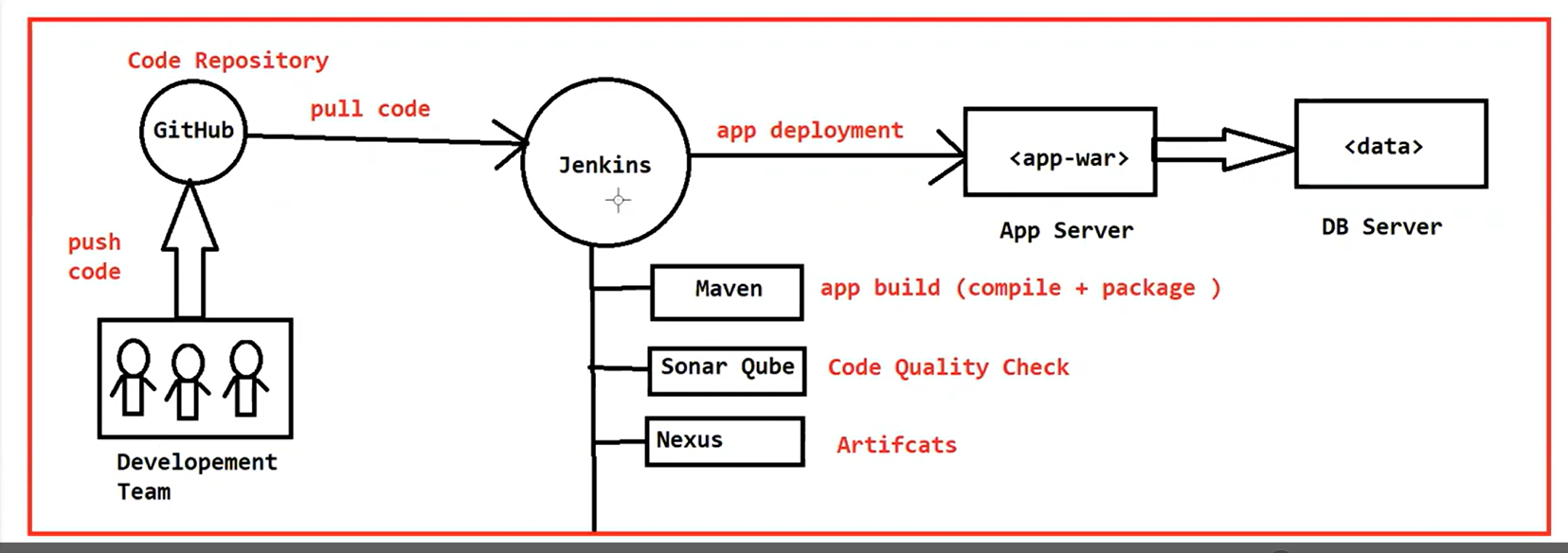
DevOps is set of practises

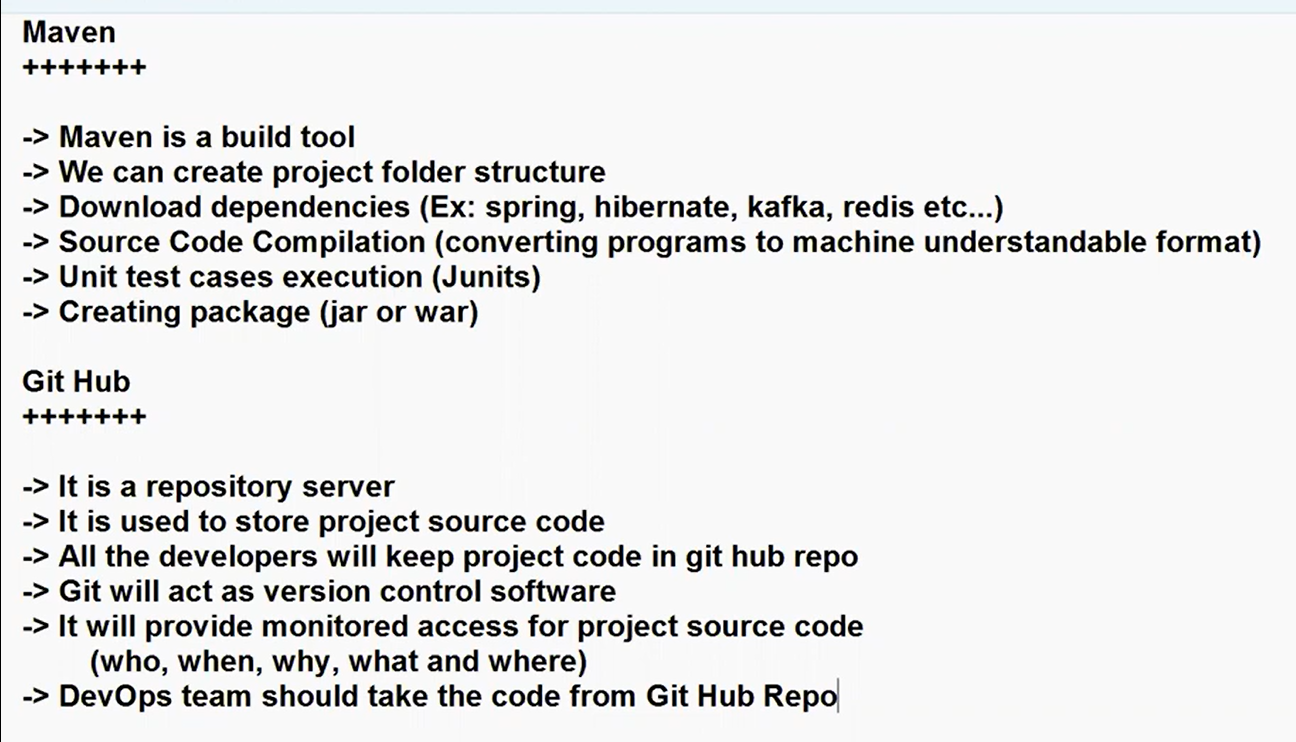
-> DevOps culture is used to colloborate Development and Operations in Software Project

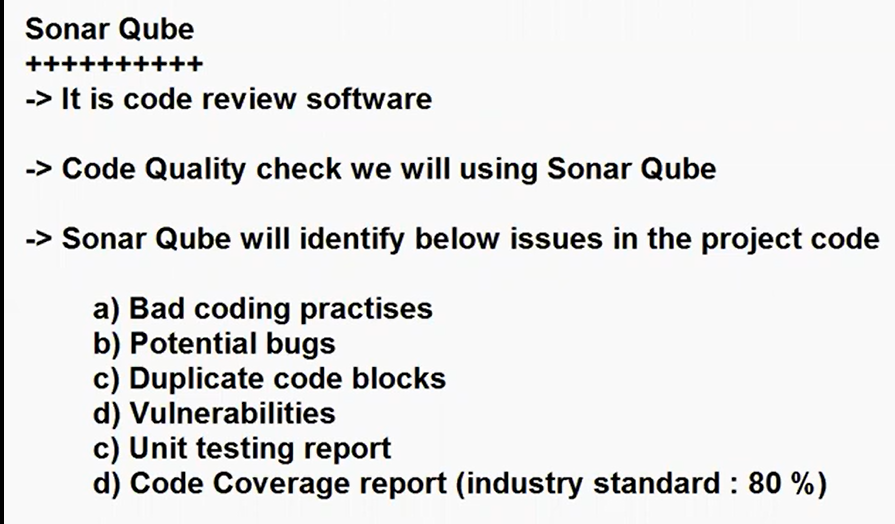
-> Using DevOps culture we can simplify software project delivery process to clients

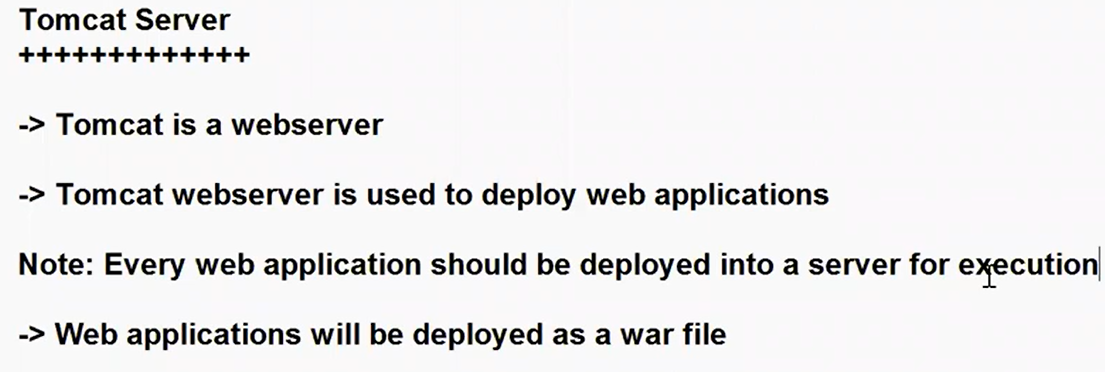
-> DevOps is used throughout software development life cycle process



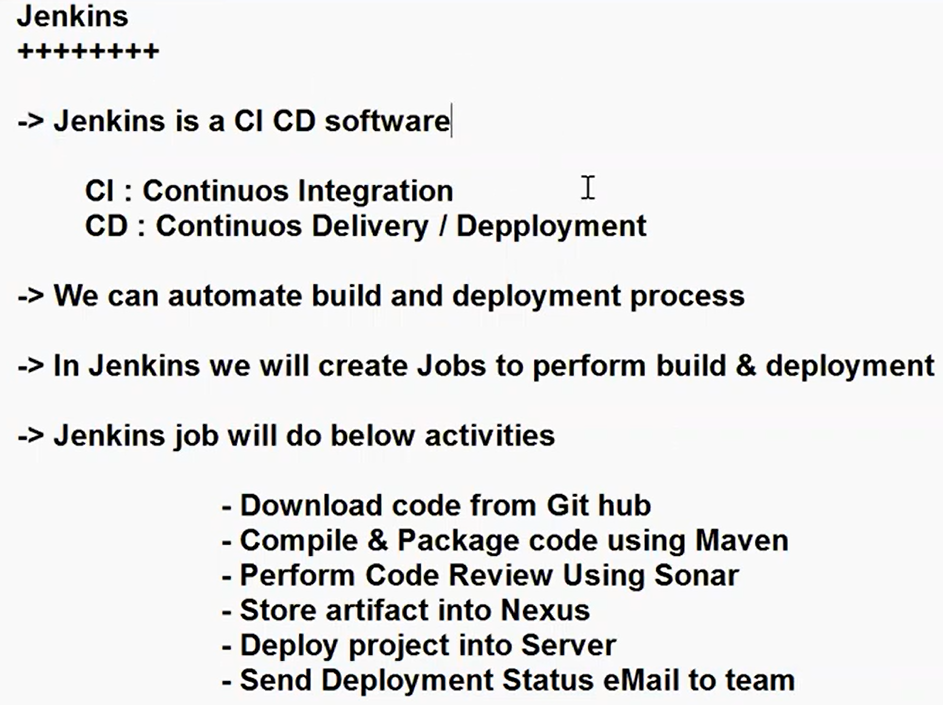






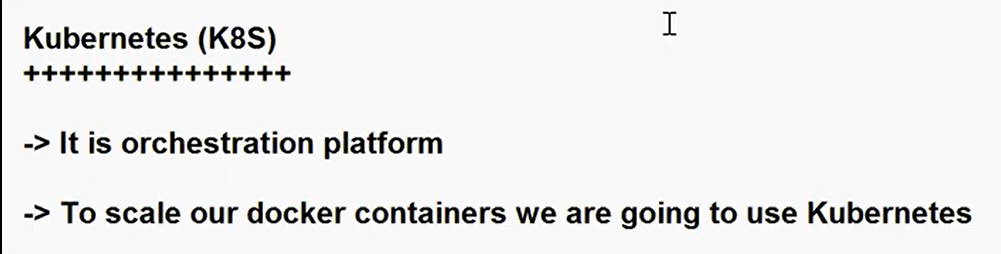


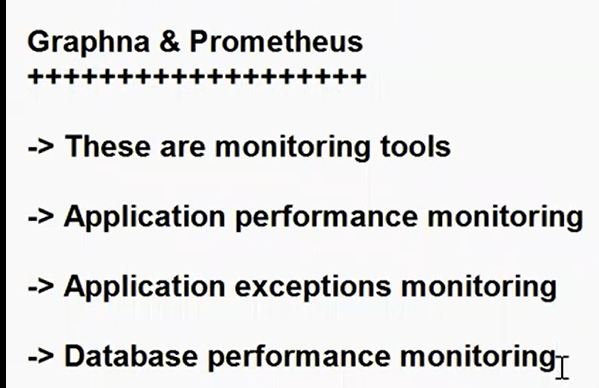


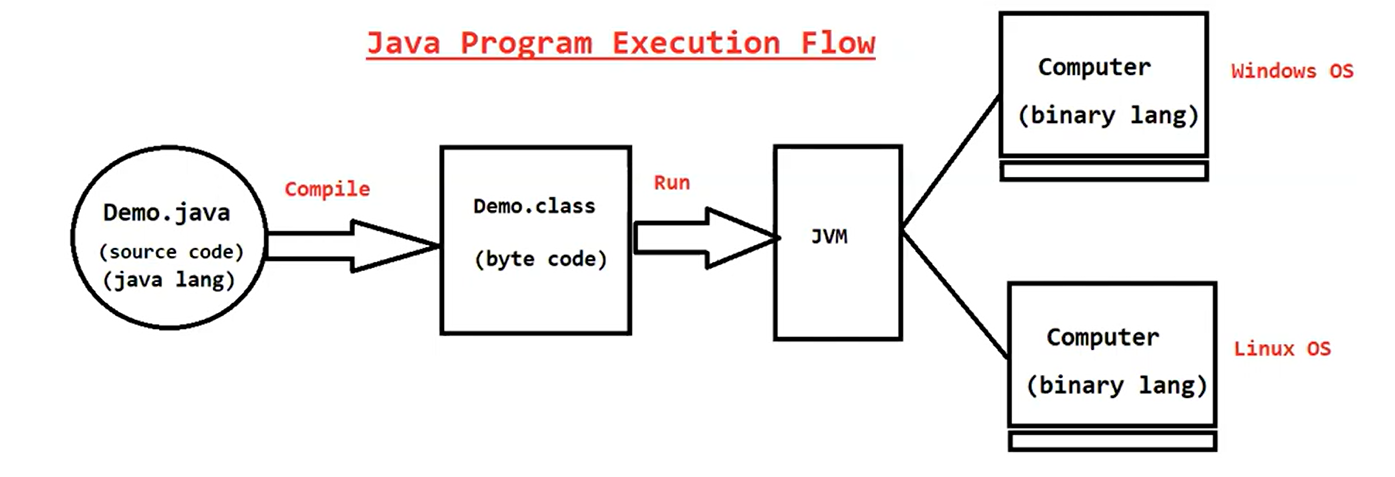


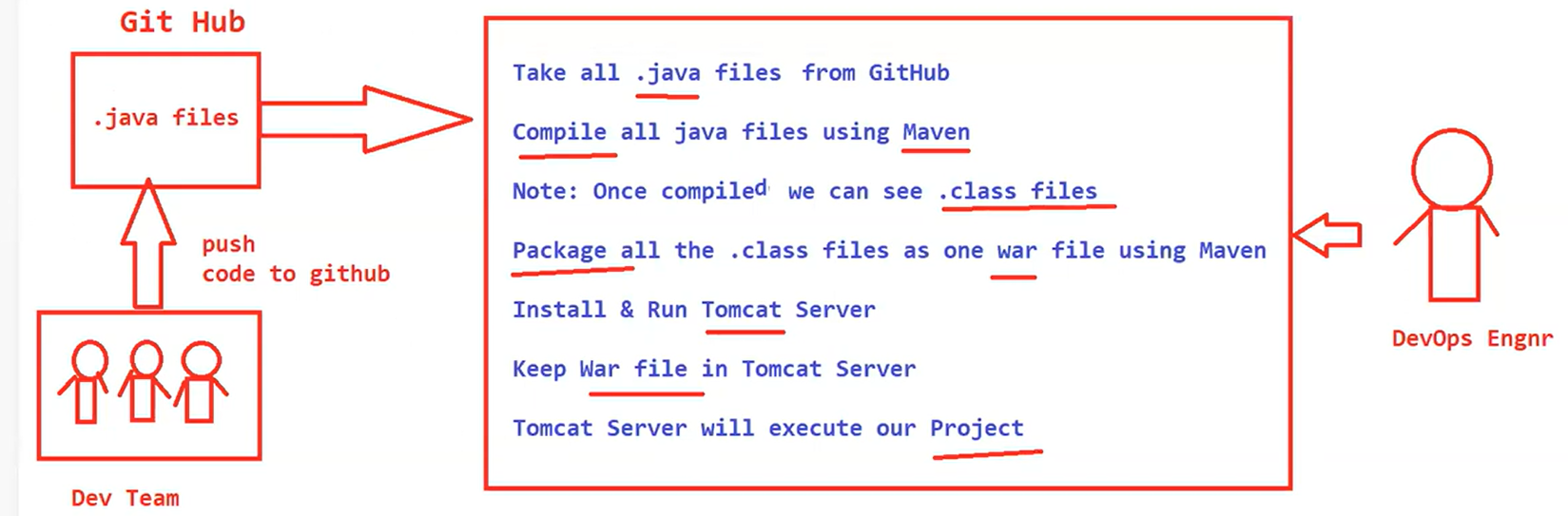












-> Java is a programming language

-> Java language developed by sun microsystem company

-> Oracle company acquired Sun Microsystem

-> Java is under license of Oracle company

-> Java is a high level programming language

-> Java is simple programming language

-> Java program files will have .java as extension

Ex: Demo.java, Hello.java, Driver.java, Calculator.java etc..

-> We can't execute .java files directley

-> Java Programs should be converted into Machine understandable format to execute

-> Java Programs (.java file) contains source code

-> We need to compile java source code into byte code using java compiler (javac)

Ex: javac Demo.java

-> When we compile java code it will create .class file

-> We need to execute .class file to run the java program

Ex: java Demo

-> When we run java program using java command, JVM will start and it will execute java program

Note: JVM stands for Java Virtual Machine

-> JVM will convert byte code into machine understandable code

-> Java project contains several java programs (.java files)

-> We need to compile project source code into byte code

-> When we compile project source code we will get .class files

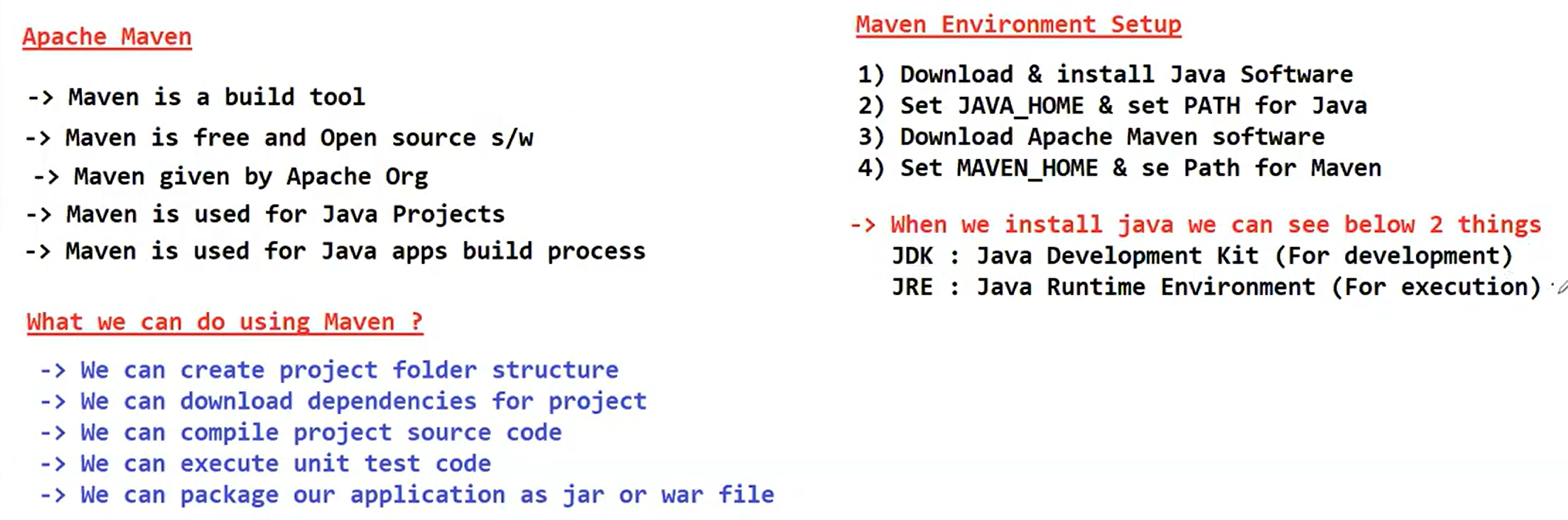
-> To deploy java project, we will package all .class files as JAR or WAR file

JAR : Java Archieve

WAR : Web Archieve

-> Standalone java projects will be packaged as JAR file

-> Web Applications will be packaged as WAR file



Maven

+++++

-> Maven is a free and open source software given by Apache Organization

-> Maven s/w is developed using Java programming language

-> Maven is used to perform Build Automation for java projects

-> Maven is called as Java Build Tool

What we can do using maven

++++++++++++++++++++++++++

1) We can create initial project folder structure using maven

2) We can download "project dependencies" using maven

(ex : springboot, hibernate, kafka, redis, email, log4j, junit, security...)

-> To develop one java project we will use several frameworks like spring, hibernate etc along with Java

-> We need to download those frameworks and we should add to our java project

-> These frameworks we are using in our project are called as our project dependencies

-> Instead of we are downloading dependencies, we can tell to maven s/w to download dependencies

Note: Required dependencies we will add in "pom.xml" file then maven s/w will download them

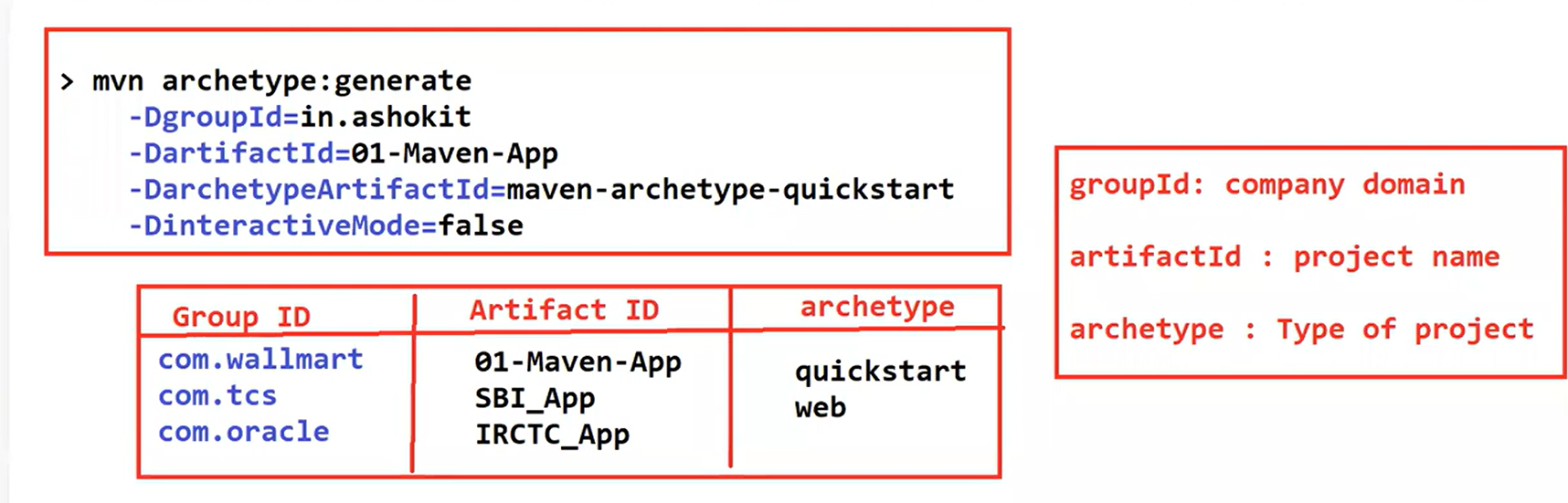
-> pom stands for project object model

-> When we create maven project then pom.xml file will be created automatically

-> pom.xml will act as input file for maven software

3) We can compile project source code using maven

4) We can package java project as jar or war file using maven



Maven Installation

++++++++++++++++++

1) Download and install Java software

-> When we install java we will below 2 things

a) JDK (Java Development Kit)

b) JRE (Java Runtime Environment)

-> JDK contains set of tools to develop java programs

-> JRE contains platform/environment which is used to run java programs

Link To Download Java : https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html

2) Set JAVA\_HOME in Environment Variables (System Env Variables)

User Environment Variables: Specific to particular account which logged in our PC

System Envrionment Variables : For All Accounts

Note: Environment Variables will be used by operating system

JAVA\_HOME = C:\Program Files\Java\jdk1.8.0\_202

3) Set Path for JAVA (Go to System Env Variables -> Env Variables -> System Variables -> Select Path and Click on Edit then add JDK path)

Path = C:\Program Files\Java\jdk1.8.0\_202\bin

4) Verify Java installation by executing below command in "Command Promot"

> java -version

Note: It should dipslay java version which we have installed

5) Download Maven software from Apache website

Link To download Maven : https://maven.apache.org/download.cgi

File Name : apache-maven-3.8.5-bin.zip (Binary Archive)

6) Extract Maven Zip file -> Copy extracted maven folder and paste it in "C" drive

7) Set MAVEN\_HOME in System Environment Variables

MAVEN\_HOME = C:\apache-maven-3.8.5

8) Set Path for Maven in System Environment Variables

Path : C:\apache-maven-3.8.5\bin

9) Open Command Prompt and verify Maven Installaton using below command

> mvn -version

--------------------------------------------------------------------------------------------------------------

Maven Terminology

++++++++++++++++++

archetype

groupId

artifactId

packaging

-> Archteype represents what type of project we want to create

=> maven-archetype-quickstart : It represents java standalone application

=> maven-archetype-webapp : It represents java web application

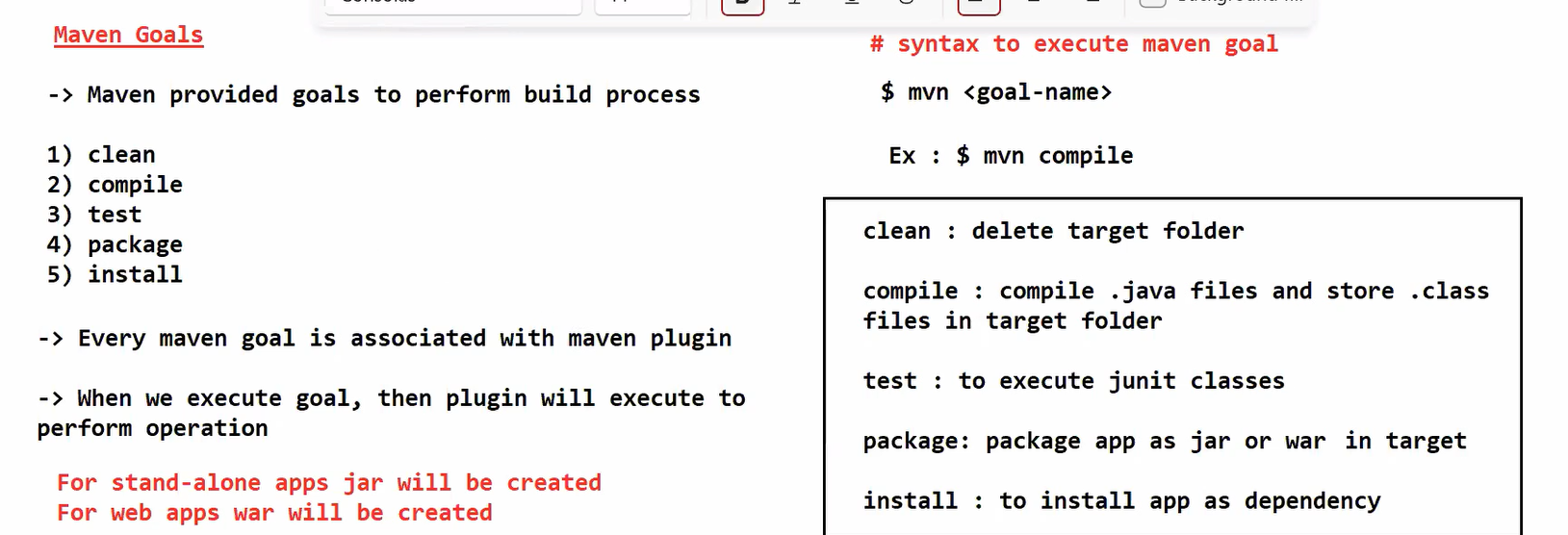
Note: Maven providing 1500+ archetypes

-> groupId represents company name or project name

-> artifactId represents project name or project module name

-> packaging represents how we want to package our java application (jar or war)





Creating standalone application using maven

+++++++++++++++++++++++++++++++++++++++++++++

1) Create one folder for maven practise

2) Open Command prompt from that folder

3) Execute below command to create maven project

>> mvn archetype:generate -DgroupId=in.ashokit -DartifactId=01-Maven-App -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

4) Once project created verify project folder structure

01-Maven-App

- src

- main

-java

- test

-java

- pom.xml

src/main/java : Application source code (.java files)

src/test/java : Application Unit Test code (.java files)

pom.xml : Project Object Model (Maven configuration file)

5) We can add dependencies in pom.xml file

6) We can find maven dependencies in www.mvnrepository.com website

7) Add below dependency in pom.xml file

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>5.2.22.RELEASE</version>

</dependency>

How maven will downoad dependencies

++++++++++++++++++++++++++++++++++++

-> Maven will download dependencies using repository

-> In Maven we have 3 types of repositories

1) Central Repository

2) Remote Repository

3) Local Repository

-> central repository is maintaining by apache organization

-> every company will maintain their own remote repository

-> Local repository will be created in our system (Location : C://users/<uname>/.m2)

-> When we add dependency in pom.xml, maven will search for that dependency in local repository. If it is available it will add to project build path.

-> If dependency not available in local repository then maven will connect to Central Repository or Remote Repository based on our configuration.

Note: By default maven will connect with central repository. If we want to use remote reposiotry then we need to configure remote repository details.

Note: Every software company will maintain their own remote repository (Ex: JFrog)

Configuring Remote Repository

+++++++++++++++++++++++++++++

<repositories>

<repository>

<id>id</id>

<url>jfrong-repo-url/</url>

</repository>

</repositories>

Maven Goals

++++++++++++

-> To perform project build activities maven prorvided several goals for us

clean

compile

test

package

install

-> clean goal is used to delete target folder

-> compile goal is used to compile project source code. Compiled code will be stored in target folder

compile

.java ------------> .class

-> test goal is used to execute unit test code of our application (junit code)

-> package goal is used to generate jar or war file for our application based on packaging type available in pom.xml file.

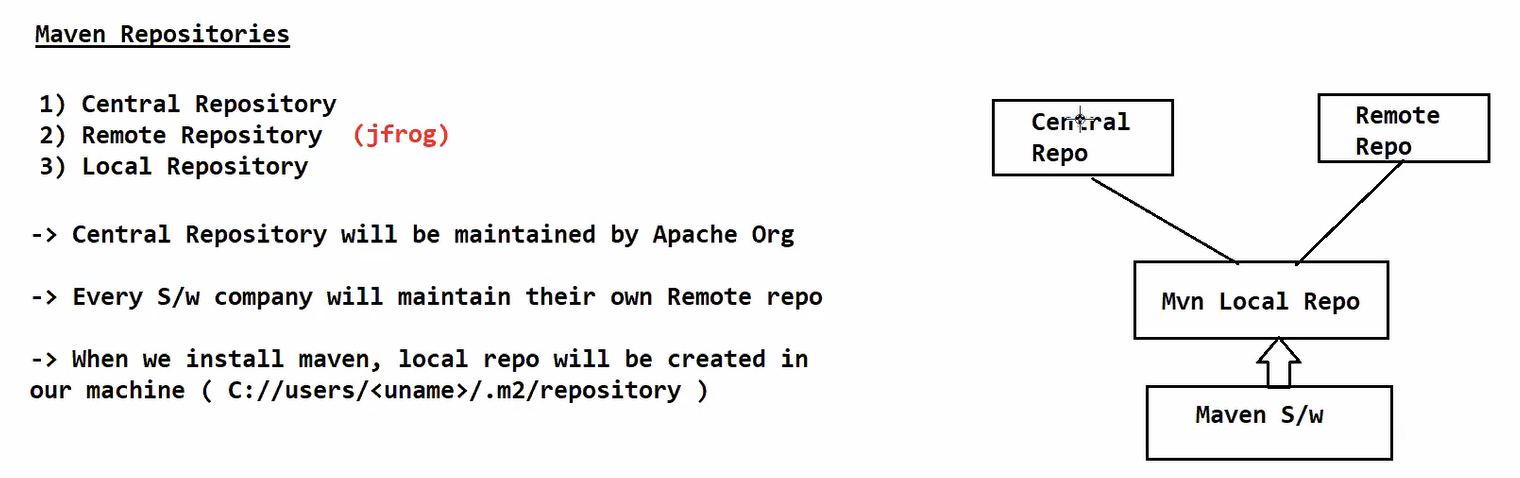
Note: jar or war file will be created in target folder.

-> install goal is used to install our project as a dependency in maven local repository.

Note: Every maven goal is associated with maven plugin. When we execute maven goal then respective maven plugin will execute to perform the operation.

Syntax : mvn <goal-name>

Note: We need to execute maven goals from project folder



creating web application using maven

+++++++++++++++++++++++++++++++++++++++

>> mvn archetype:generate -DarchetypeArtifactId=maven-archetype-webapp -DgroupId=in.ashokit -DartifactId=01-maven-web-app -DinteractiveMode=false

----------------------------------------------------------------------------------------------------

What is Java ?

What is source code ?

What is Byte code ?

What is machine understandable code ?

Java Program Execution Process

What is JDK

What is JRE

What is JVM

What is Project Dependency? (Ex : hibernate, spring, kafka, redis etc)

What is Maven

What are the advantages of Maven

Java Installation

Setting JAVA\_HOME & JDK PATH

Maven Installation

setting MAVEN\_HOME & Maven Path

Maven Terminology

- archetype

- groupId

- artifactId

- version

- packaging

Pom.xml

Adding dependency in pom.xml

Java Standalone project creation using Maven

Maven Project Folder Structure

Java Web application creating using Maven

Maven Repositories

- Local Repository

- Central Repository

- Remote Repository

Maven Goals

- clean

- compile

- test

- package

- install

Maven Plugins

-------------------------------------------------------------------------------------------------------

Software Development Life Cycle (SDLC)

--------------------------------------

-> The process of developing and delivering software project is called as SDLC

-> In SDLC we have several phases

1) Requirements Gathering

2) Requirements Analysis

3) Design / Planning

4) Development / Coding / Implementation

5) Testing

6) Deployment

7) Delivery

8) Maintenence

Waterfall Methodology

+++++++++++++++++++++++++

-> Earlier people used to follow Waterfall Methodology to develop projects

-> Waterfall is a linear methodology to develop and deliver projects

-> Everything will happen step by step

-> If one step completed then only we will go to next step

-> We will move only in forward direction (No backward direction)

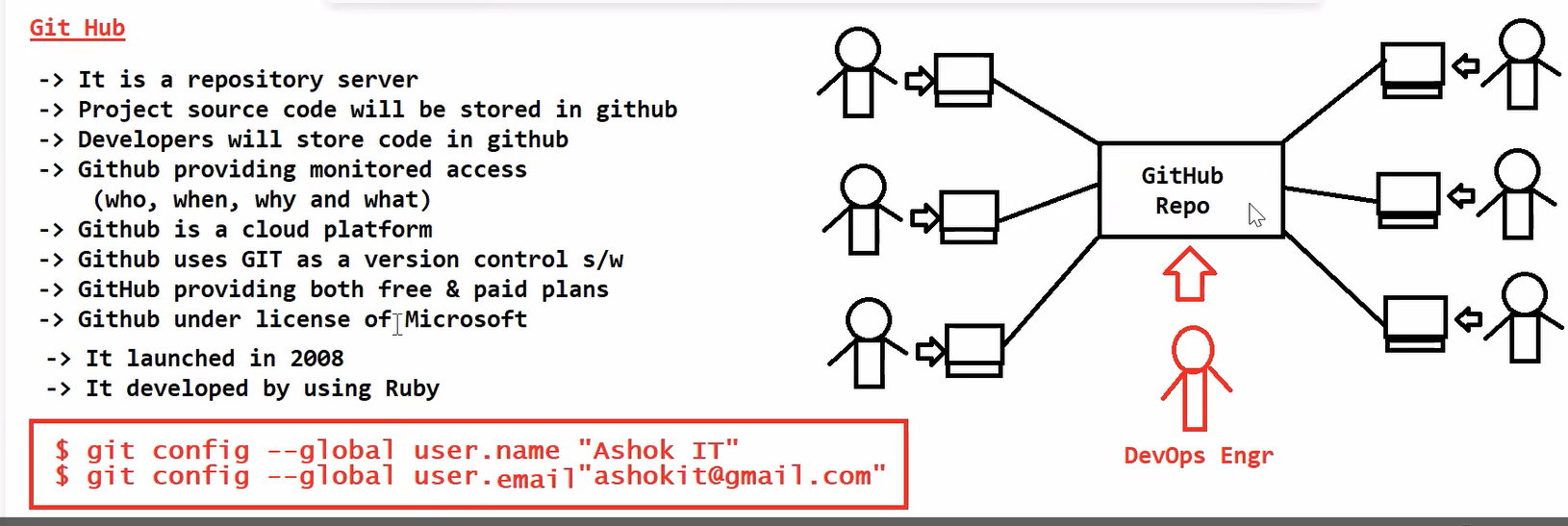
-> Requirements are fixed

-> Budget is fixed

-> Client involvement is very less

-> Client will see the project at the end

Note: Waterfall Methodology is not suitable for big projects



Agile Methodology

+++++++++++++++++

-> Agile is an iterative approach to develop and deliver the projects

-> Development and testing will happen parallelly

-> Client involvment will be very high

-> We will deliver project in multiple releases (Sprints)

-> For every release we will take client feedback

-> Requirements are not fixed

-> Budget is not fixed

-> Project Development, Testing & Delivery is very frequent is Agile

--------------------------------------------------------------------------------------------------------

-> Using DevOps culture we can adopt agile methodology very easily

-> DevOps is promoting Agile methodology

-> Using DevOps we can achive Continutous Integration (CI) & Continous Deployment / Delivery (CD)

CI CD

-------------------------------------------------------------------------------------------------------

DevOps Advantages

-------------------------------------------------------------------------------------------------------

1) Speed

2) Rapid Development

3) Quick Releases

4) Reliability

5) Security

6) Client Satisfaction

7) Teams Colloboration

Note: DevOps is not one person job, it is everyone's job in the project

-------------------------------------------------------------------------------------------------------

DevOps tools overview

--------------------------------------------------------------------------------------------

Build Tools ( Ant / Maven / Gradle )

Repository Tools ( SVN / Git Hub / BitBucket )

Code Review Tools ( PMD / Sonar Qube / Sonar lint )

Code Deployment Tools ( Jenkins / UDeploy )

Containerization Tools ( Docker )

Orchestration Tools ( Kubernetes )

Configuration Tools ( Chef / Ansible )

Infrastructure as a Code (IaaC) (Terraform)

Monitoring Tools (Nagios / Graphana )

Project Management Tools ( Jira )

-> Java is a programming language

-> Java language developed by sun microsystem company

-> Oracle company acquired Sun Microsystem

-> Java is under license of Oracle company

-> Java is a high level programming language

-> Java is simple programming language

-> Java program files will have .java as extension

Ex: Demo.java, Hello.java, Driver.java, Calculator.java etc..

-> We can't execute .java files directley

-> Java Programs should be converted into Machine understandable format to execute

-> Java Programs (.java file) contains source code

-> We need to compile java source code into byte code using java compiler (javac)

Ex: javac Demo.java

-> When we compile java code it will create .class file

-> We need to execute .class file to run the java program

Ex: java Demo

-> When we run java program using java command, JVM will start and it will execute java program

Note: JVM stands for Java Virtual Machine

-> JVM will convert byte code into machine understandable code

-> Java project contains several java programs (.java files)

-> We need to compile project source code into byte code

-> When we compile project source code we will get .class files

-> To deploy java project, we will package all .class files as JAR or WAR file

JAR : Java Archieve

WAR : Web Archieve

-> Standalone java projects will be packaged as JAR file

-> Web Applications will be packaged as WAR file

Maven

+++++

-> Maven is a free and open source software given by Apache Organization

-> Maven s/w is developed using Java programming language

-> Maven is used to perform Build Automation for java projects

-> Maven is called as Java Build Tool

What we can do using maven

++++++++++++++++++++++++++

1) We can create initial project folder structure using maven

2) We can download "project dependencies" using maven

(ex : springboot, hibernate, kafka, redis, email, log4j, junit, security...)

-> To develop one java project we will use several frameworks like spring, hibernate etc along with Java

-> We need to download those frameworks and we should add to our java project

-> These frameworks we are using in our project are called as our project dependencies

-> Instead of we are downloading dependencies, we can tell to maven s/w to download dependencies

Note: Required dependencies we will add in "pom.xml" file then maven s/w will download them

-> pom stands for project object model

-> When we create maven project then pom.xml file will be created automatically

-> pom.xml will act as input file for maven software

3) We can compile project source code using maven

4) We can package java project as jar or war file using maven

Maven Installation

++++++++++++++++++

1) Download and install Java software

-> When we install java we will below 2 things

a) JDK (Java Development Kit)

b) JRE (Java Runtime Environment)

-> JDK contains set of tools to develop java programs

-> JRE contains platform/environment which is used to run java programs

Link To Download Java : https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html

2) Set JAVA\_HOME in Environment Variables (System Env Variables)

User Environment Variables: Specific to particular account which logged in our PC

System Envrionment Variables : For All Accounts

Note: Environment Variables will be used by operating system

JAVA\_HOME = C:\Program Files\Java\jdk1.8.0\_202

3) Set Path for JAVA (Go to System Env Variables -> Env Variables -> System Variables -> Select Path and Click on Edit then add JDK path)

Path = C:\Program Files\Java\jdk1.8.0\_202\bin

4) Verify Java installation by executing below command in "Command Promot"

> java -version

Note: It should dipslay java version which we have installed

5) Download Maven software from Apache website

Link To download Maven : https://maven.apache.org/download.cgi

File Name : apache-maven-3.8.5-bin.zip (Binary Archive)

6) Extract Maven Zip file -> Copy extracted maven folder and paste it in "C" drive

7) Set MAVEN\_HOME in System Environment Variables

MAVEN\_HOME = C:\apache-maven-3.8.5

8) Set Path for Maven in System Environment Variables

Path : C:\apache-maven-3.8.5\bin

9) Open Command Prompt and verify Maven Installaton using below command

> mvn -version

--------------------------------------------------------------------------------------------------------------

Maven Terminology

++++++++++++++++++

archetype

groupId

artifactId

packaging

-> Archteype represents what type of project we want to create

=> maven-archetype-quickstart : It represents java standalone application

=> maven-archetype-webapp : It represents java web application

Note: Maven providing 1500+ archetypes

-> groupId represents company name or project name

-> artifactId represents project name or project module name

-> packaging represents how we want to package our java application (jar or war)

Creating standalone application using maven

+++++++++++++++++++++++++++++++++++++++++++++

1) Create one folder for maven practise

2) Open Command prompt from that folder

3) Execute below command to create maven project

>> mvn archetype:generate -DgroupId=in.ashokit -DartifactId=01-Maven-App -DarchetypeArtifactId=maven-archetype-quickstart -DinteractiveMode=false

4) Once project created verify project folder structure

01-Maven-App

- src

- main

-java

- test

-java

- pom.xml

src/main/java : Application source code (.java files)

src/test/java : Application Unit Test code (.java files)

pom.xml : Project Object Model (Maven configuration file)

5) We can add dependencies in pom.xml file

6) We can find maven dependencies in www.mvnrepository.com website

7) Add below dependency in pom.xml file

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-core</artifactId>

<version>5.2.22.RELEASE</version>

</dependency>

How maven will downoad dependencies

++++++++++++++++++++++++++++++++++++

-> Maven will download dependencies using repository

-> In Maven we have 3 types of repositories

1) Central Repository

2) Remote Repository

3) Local Repository

-> central repository is maintaining by apache organization

-> every company will maintain their own remote repository

-> Local repository will be created in our system (Location : C://users/<uname>/.m2)

-> When we add dependency in pom.xml, maven will search for that dependency in local repository. If it is available it will add to project build path.

-> If dependency not available in local repository then maven will connect to Central Repository or Remote Repository based on our configuration.

Note: By default maven will connect with central repository. If we want to use remote reposiotry then we need to configure remote repository details.

Note: Every software company will maintain their own remote repository (Ex: JFrog)

Configuring Remote Repository

+++++++++++++++++++++++++++++

<repositories>

<repository>

<id>id</id>

<url>jfrong-repo-url/</url>

</repository>

</repositories>

Maven Goals

++++++++++++

-> To perform project build activities maven prorvided several goals for us

clean

compile

test

package

install

-> clean goal is used to delete target folder

-> compile goal is used to compile project source code. Compiled code will be stored in target folder

compile

.java ------------> .class

-> test goal is used to execute unit test code of our application (junit code)

-> package goal is used to generate jar or war file for our application based on packaging type available in pom.xml file.

Note: jar or war file will be created in target folder.

-> install goal is used to install our project as a dependency in maven local repository.

Note: Every maven goal is associated with maven plugin. When we execute maven goal then respective maven plugin will execute to perform the operation.

Syntax : mvn <goal-name>

Note: We need to execute maven goals from project folder

creating web application using maven

+++++++++++++++++++++++++++++++++++++++

>> mvn archetype:generate -DarchetypeArtifactId=maven-archetype-webapp -DgroupId=in.ashokit -DartifactId=01-maven-web-app -DinteractiveMode=false

----------------------------------------------------------------------------------------------------

What is Java ?

What is source code ?

What is Byte code ?

What is machine understandable code ?

Java Program Execution Process

What is JDK

What is JRE

What is JVM

What is Project Dependency? (Ex : hibernate, spring, kafka, redis etc)

What is Maven

What are the advantages of Maven

Java Installation

Setting JAVA\_HOME & JDK PATH

Maven Installation

setting MAVEN\_HOME & Maven Path

Maven Terminology

- archetype

- groupId

- artifactId

- version

- packaging

Pom.xml

Adding dependency in pom.xml

Java Standalone project creation using Maven

Maven Project Folder Structure

Java Web application creating using Maven

Maven Repositories

- Local Repository

- Central Repository

- Remote Repository

Maven Goals

- clean

- compile

- test

- package

- install

Maven Plugins