At root folder

Install tomcat7

yum install tomcat7

yum install tomcat7-webapps

cd /usr/local

wget http://mirrors.estointernet.in/apache/maven/maven-3/3.6.1/binaries/apache-maven-3.6.1-bin.tar.gz

tar -xzvf apache-maven-3.6.1-bin.tar.gz

ln -s apache-maven-3.6.1 maven

The ln command is a standard Unix command utility used to create a hard link or a symbolic link (symlink) to an existing file.[1] The use of a hard link allows multiple filenames to be associated with the same file since a hard link points to the inode of a given file, the data of which is stored on disk. On the other hand, symbolic links are special files that refer to other files by name.[2]

The ln command by default creates hard links, and when called with the command line parameter ln -s creates symbolic links Links allow more than one filename to refer to the same file as in the case of a hard link or act as pointers to a filename as in the case of a soft link.

vi /etc/profile.d/maven.sh

export M2\_HOME=/usr/local/maven

export PATH=${M2\_HOME}/bin:${PATH}

source /etc/profile.d/maven.sh

mvn –version

cd /home/ec2-user/

mkdir project

cd project/

yum install -y git

git clone https://github.com/raghuopsdev/hello-world.git

cd hello-world/

vi pom.xml

mvn clean package

java –version

mvn –version

yum list | grep jdk

yum install java-1.8.0-openjdk-devel

alternatives --config java

select 2 option that is 1.8 version

java –version

mvn –version

mvn clean package

cp /home/ec2-user/project/hello-world/webapp/target/webapp.war /usr/share/tomcat7/webapps/

cd /usr/share/tomcat7/webapps/

service tomcat7 start

curl icanhazip.com

in website IP address :8080/webapp

**Maven Tutorial**

<https://www.tutorialspoint.com/maven/maven_repositories.htm>

<https://www.javatpoint.com/maven-tutorial>

**What is Maven?**

Maven is an automation and management tool developed by Apache Software Foundation. It was initially released on 13 July 2004. In Yiddish language the meaning of Maven is "accumulator of knowledge".

It is written in Java Language and used to build and manage projects written in C#, Ruby, Scala, and other languages. It allows the developer to create projects using Project Object Model and plugins.

It helps to build projects, dependency, and documentation. Its development process is very similar to ANT.However, it is much advanced than ANT.

Maven is also able to build any number of projects into desired output such as jar, war, metadata.

In this tutorial, you will learn

* [What is Maven?](https://www.guru99.com/maven-tutorial.html#1)
* [How can Maven benefit my development process?](https://www.guru99.com/maven-tutorial.html#2)
* [Processes which can manage using maven](https://www.guru99.com/maven-tutorial.html#3)
* [Maven Architecture](https://www.guru99.com/maven-tutorial.html#4)
* [How to use Maven](https://www.guru99.com/maven-tutorial.html#5)
* [Steps/process involved in building the project](https://www.guru99.com/maven-tutorial.html#6)

**How can Maven benefit my development process?**

Maven helps the developer to create a java-based project more easily. Accessibility of new feature created or added in Maven can be easily added to a project in Maven configuration. It increases the performance of project and building process.

The main feature of Maven is that it can download the project dependency libraries automatically.

Below are the examples of some popular IDEs supporting development with Maven:

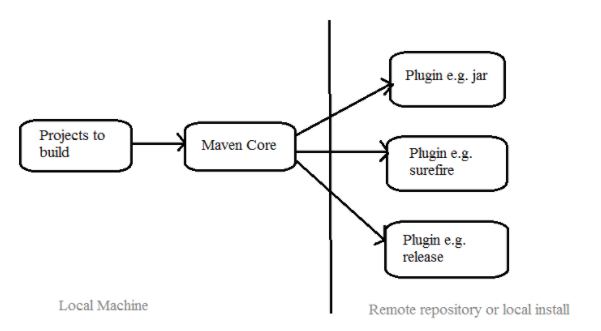
* Eclipse
* IntelliJ IDEA
* JBuilder
* NetBeans
* MyEclipse

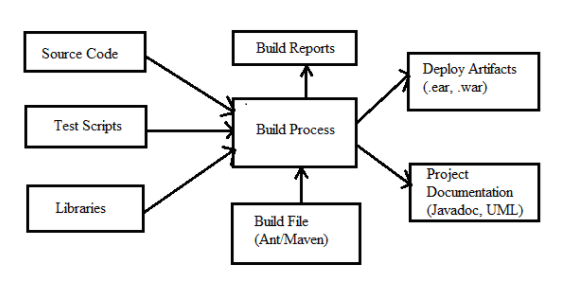
**Processes which can manage using maven:**

* Builds
* Documentation
* Reporting
* Dependencies
* SCMs
* Releases
* Distribution
* mailing list

**Maven Architecture**

Maven Architecture includes plugin jar, code file etc.

[](https://www.guru99.com/images/1/041318_1113_MavenTutori1.png)

[](https://www.guru99.com/images/1/041318_1113_MavenTutori2.png)

**How to use Maven**

* To configure the Maven, you need to use Project Object Model, which is stored in a pom.xml-file.
* POM includes all the configuration setting related to Maven. Plugins can be configured and edit in the <plugins> tag of a pom.xml file. And developer can use any plugin without much detail of each plugin.
* When user start working on Maven, it provides default setting of configuration, so the user does not need to add every configuration in pom.xml

**Steps/process involved in building the project:**

* Add / Write the code for application creation and process that into source code repository
* Edit configuration / pom.XML / plugin details
* Build the application
* Save the build process output as WAR or EAR file to a local location or server
* Get the file from local location or server and deploy the file to the production site or
* client site Updated the application document with date and updated version number of the application
* create and generate a report as per the application or requirement.

**Summary:**

* Maven is an automation and management tool.
* It is written in Java Language and used to build and manage projects written in C#, Ruby, Scala, and other languages.
* Maven helps the developer to create a java-based project more easily.
* To configure the Maven, you need to use Project Object Model, which is stored in a pom.xml-file.