

# Install & configure Maven build tool on Jenkins

In this page, we will describe the installation and configuration process of Maven in our Jenkins server

- Vagrant, Centos 7 OS & Jenkins Server.
- Also Works for : AWS EC2, Red Hat, Jenkins Server

## step 1 : connect to the Jenkins server and install Maven

as a reminder, we have our jenkins server installed with vagrant which contains the following characteristics :

- IP Address : **192.168.56.117**
- user : **vagrant**
- password : **vagrant**

the process is as follows:

- 1- Open the terminal,
- 2- connect to the server with **ssh**
- 3- use the **yum** package manager to **install git**

```
ssh vagrant@192.168.56.177
```

## step 2 : Copy the maven latest version of binary source

**Maven** is a **code build tool** which used to convert your code to an artifact. this is a widely used **plugin** to **build in continuous integration**.

### I- Process for copy the maven link

- 1- Open a new tab in the browser
- 2- Open the link: <https://maven.apache.org/download.cgi>

**Note :** the page that opens is the page of the **official site of Maven**. the last version of the tool is there. to obtain it go in the section File of this page, make a right click on the link **apache-maven-3.8.5-src.tar.gz** then copy the address of this link.

- 3- Scroll down in the **Files** section of this page. the line that is marked **Binary tar.gz archive** is the one we will use
- 4- Right-click on **apache-maven-3.8.5-bin.tar.gz** , in the popup that display click on **Copy link address**

### Files

Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the [link](#) to Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to [verify the signature](#) of the release bundles

	Link	Checksums
Binary tar.gz archive	<a href="#">apache-maven-3.8.5-bin.tar.gz</a>	<a href="#">apache-maven-3.8.5-bin.tar.gz.sha512</a>
Binary zip archive	<a href="#">apache-maven-3.8.5-bin.zip</a>	<a href="#">apache-maven-3.8.5-bin.zip.sha512</a>
Source tar.gz archive	<a href="#">apache-maven-3.8.5-src.tar.gz</a>	<a href="#">apache-maven-3.8.5-src.tar.gz.sha512</a>
Source zip archive	<a href="#">apache-maven-3.8.5-src.zip</a>	<a href="#">apache-maven-3.8.5-src.zip.sha512</a>

- [Release Notes](#)
- [Reference Documentation](#)
- [Apache Maven Website As Documentation Archive](#)
- All current release sources (plugins, shared libraries,...) available at <https://downloads.apache.org/maven/>

### II- Download the binary using wget

in the terminal, when you are connected to the Jenkins server, execute these commands :

# cd /opt then # ls

# wget paste the link and let it download (# ls to make sure it downloaded successfully)

# tar -xf apache-maven-3.8.5-bin.tar.gz then # ls

- Remove the tar file: # rm -rf apache-maven-3.8.5-bin.tar.gz
- Let's change the directory name to simply maven:

# mv apache-maven-3.8.5/ maven then # ls

```
sudo -i
cd /opt
ls
wget https://dlcdn.apache.org/maven/maven-3/3.8.5/binaries/apache-
maven-3.8.5-bin.tar.gz --no-check-certificate
tar -xf apache-maven-3.8.5-bin.tar.gz
ls
rm -rf apache-maven-3.8.5-bin.tar.gz
mv apache-maven-3.8.5/ maven
ls
```

```
▼ TERMINAL
[vagrant@jenkinshost-utrains ~]$ sudo -i
[root@jenkinshost-utrains ~]# cd /opt
[root@jenkinshost-utrains opt]# ls
[root@jenkinshost-utrains opt]# wget https://dlcdn.apache.org/maven/maven-3/3.8.5/binaries/apache-maven-3.8.5-bin.tar.gz --no-check-certificate
--2022-04-08 19:06:16-- https://dlcdn.apache.org/maven/maven-3/3.8.5/binaries/apache-maven-3.8.5-bin.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
WARNING: cannot verify dlcdn.apache.org's certificate, issued by '/C=US/O=Let's Encrypt/CN=R3':
  Issued certificate has expired.
HTTP request sent, awaiting response... 200 OK
Length: 8673123 (8.3M) [application/x-gzip]
Saving to: 'apache-maven-3.8.5-bin.tar.gz'

100%[=====] 8,673,123  4.90MB/s  in 1.7s

2022-04-08 19:06:18 (4.90 MB/s) - 'apache-maven-3.8.5-bin.tar.gz' saved [8673123/8673123]

[root@jenkinshost-utrains opt]# ls
apache-maven-3.8.5-bin.tar.gz
[root@jenkinshost-utrains opt]# tar -xf apache-maven-3.8.5-bin.tar.gz
[root@jenkinshost-utrains opt]# ls
apache-maven-3.8.5  apache-maven-3.8.5-bin.tar.gz
[root@jenkinshost-utrains opt]# rm -rf apache-maven-3.8.5-bin.tar.gz
[root@jenkinshost-utrains opt]# ls
apache-maven-3.8.5
[root@jenkinshost-utrains opt]# mv apache-maven-3.8.5/ maven
[root@jenkinshost-utrains opt]# ls
maven
[root@jenkinshost-utrains opt]#
```

command for download the maven Binary

command to unzip archive of maven

### III- Setup M2\_HOME and M2 paths in .bash\_profile of the user and add these to the path variable

# cd (to go back to the root directory)

# vim .bash\_profile then go to the Insert mode

M2\_HOME=/opt/maven

M2=/opt/maven/bin

- Modify the PATH variable by adding the new variable created : PATH=\$PATH:\$HOME/bin:\$JAVA\_HOME:\$M2\_HOME:\$M2
- Save and Quit

```
cd
vim ~/.bash_profile
M2_HOME=/opt/maven
M2=$M2_HOME/bin
PAHT=<Existing_PATH>:$M2_HOME:$M2
```

- **.bash\_profile** file content

#### ▼ TERMINAL

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
    . ~/.bashrc
fi

# User specific environment and startup programs

# Maven Path conf:
M2_HOME=/opt/maven
M2=/opt/maven/bin

# Java path conf :
JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.322.b06-2.el8_5.x86_64

PATH=$PATH:$HOME/bin:$JAVA_HOME:$M2_HOME:$M2

export PATH

~
```

- make the final command for apply path changes : **source .bash\_profile**
- **mvn** command : check if all is correctly configured

```
source .bash_profile
echo $M2
mvn -version
```

```
[root@jenkinshost-utrains ~]# vim .bash_profile
[root@jenkinshost-utrains ~]# source .bash_profile
[root@jenkinshost-utrains ~]# echo $M2
/opt/maven/bin
[root@jenkinshost-utrains ~]# mvn -version
Apache Maven 3.8.5 (3599d3414f046de2324283b786ddcf9b5e4388aa0)
Maven home: /opt/maven
Java version: 1.8.0_322, vendor: Red Hat, Inc., runtime: /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.322.b06-1.el7_9.x86_64/
jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "3.10.0-1127.el7.x86_64", arch: "amd64", family: "unix"
[root@jenkinshost-utrains ~]#
```

command for display all maven config

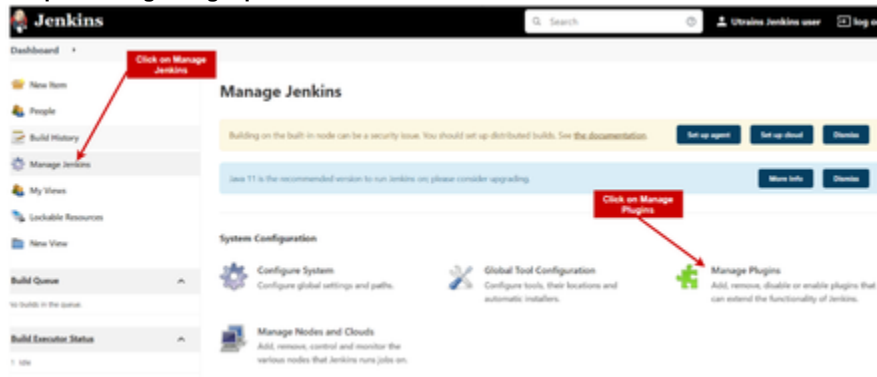
So far we have completed the installation of maven software to support maven plugin on the jenkins console. Let's jump onto Jenkins to complete the remaining steps.

#### IV- Setup maven on Jenkins console

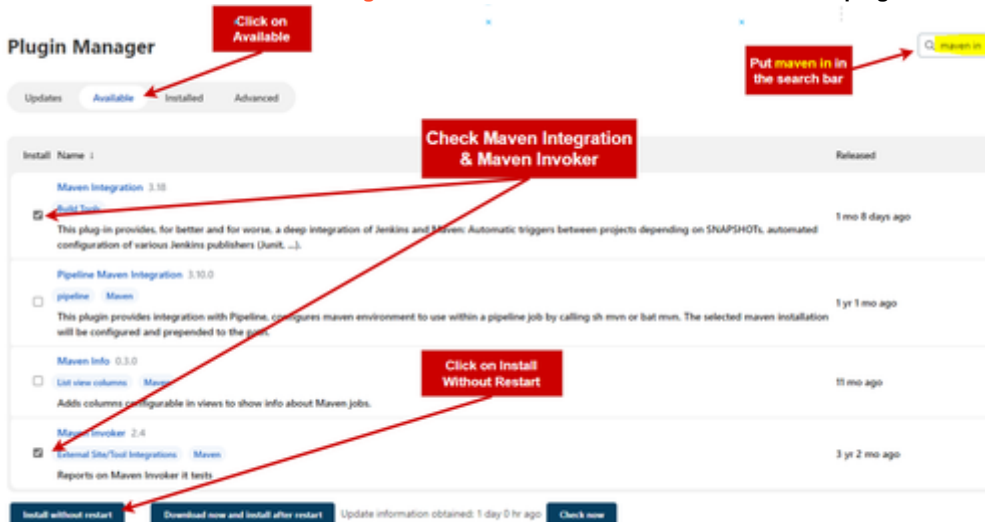
##### 1. Install maven plugin without restart

- Manage Jenkins > Jenkins Plugins > available > Maven Invoker
- Manage Jenkins > Jenkins Plugins > available > Maven Integration]

##### a- Open Manage Plugin process



##### a- Search Maven In check Maven Integration & Maven Invocation then install these 2 plugins



##### 2. Configure maven path

- Manage Jenkins > Global Tool Configuration > Maven

click on **Add Maven**, Uncheck **Install Automatically** then put these configuration :

**Name:** M2\_HOME

**MAVEN\_HOME:** /opt/maven

Click on **Apply** then **Save** to save the maven path in our Jenkins web interface

## Maven

### Maven installations

Add Maven

Add Maven



Maven

Name

M2\_HOME

MAVEN\_HOME

/opt/maven

☐

Install automatically ?

Add Maven

List of Maven installations on this system

Save

Apply

#### V- Maven First Project creation

Click on New Item to  
create new Project

- New Item
- People
- Build History
- Manage Jenkins
- My Views
- Lockable Resources

Enter an item name

Put the Project Name.  
Here mavenproject

mavenproject


» Required field


Click on Maven  
Project





Freestyle project


This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this


**Maven project**  
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.

**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known

**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, pla

**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a

**Multibranch Pipeline**  
Creates a set of Pipeline projects according to detected branches in one SCM repository.


**Organization Folder**  
Creates a set of multibranch project subfolders by scanning for repositories.

**Click on Ok button to validate project creation**

**OK**

#### Releases

- **1.0** : build for Utrains Student

 Date : **01/04/2022**

 Participants : **hermann.chefouetmeka@utrains.org**

 Link for similar document

-  [Create EC2 in AWS using Red Hat Enterprise image](#)