MIRACLE
SOFTWARE SYSTEMS

Deploying your first Java Application to AWS-3 (Installations)

Hands-On Workshop | Digital Summit '18

Miracle Innovation Labs

Miracle Software Systems, Inc.

Deploying your first Java Application to AWS-3 (Installations)

Introduction

The goal of this document is to install and configure Java, Maven and deploying a sample Maven Application on Tomcat.

This guide was prepared by **Miracle's Innovation Labs.**

Pre-Requisites

All attendees must have their workstation (with Internet) to participate in the workshop (Both PC and MAC are compatible). The following pre-requisites will help you to make the workshop experience easier.

- AWS account
- Download and install Java
- Download and install Maven
- Text Editor such as Notepad++ (or) Sublime Text

Technology Involved

- AWS
- Java
- Maven
- Apache Tomcat
- Git

Lab Steps

So, let us get started with the application!

In this document, we will show you how to install Java, Maven and setting their permanent paths.

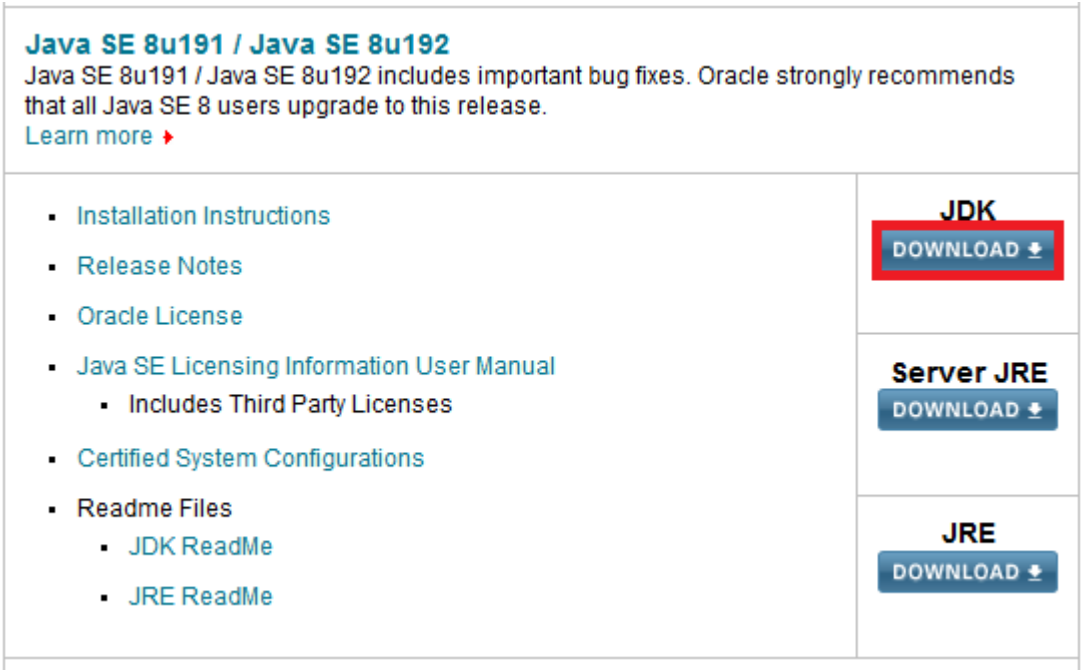
Also, we will be showing how to get the sample **Maven_Application** from Github, creating **.war** file and finally deploying the application on **Tomcat** server.

Step #1 | Installation of Java on Windows

To download the **Java Development Kit (JDK)**, launch your web browser (e.g. Internet Explorer) and go to the below link,
http://java.sun.com/javase/downloads/index.jsp

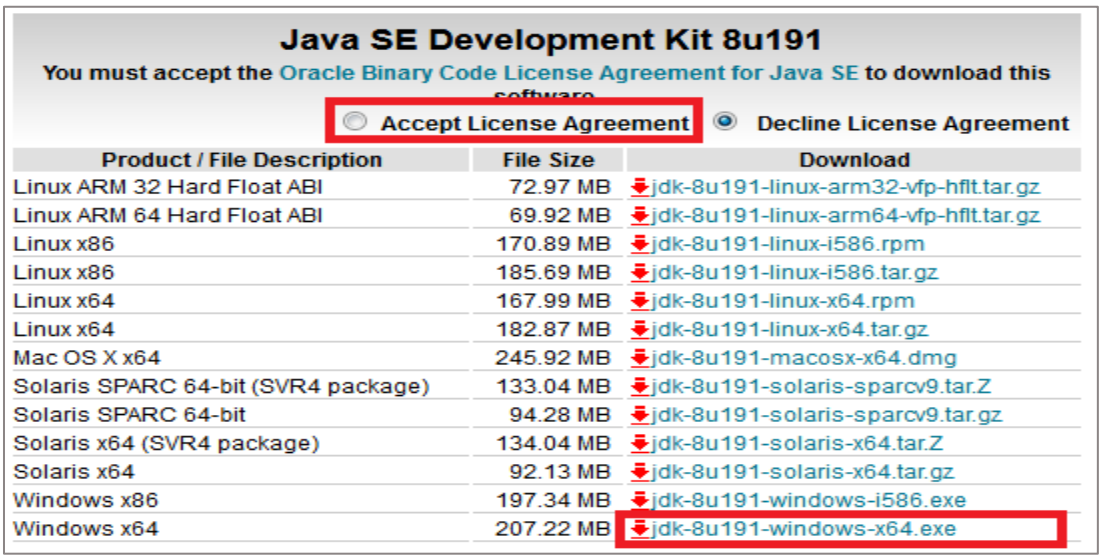
Once you open the link, it displays multiple download options. The page displays the most frequent JDK download options (e.g. JDK1.8).

Click on the **DOWNLOAD** button that corresponds to the Oracle Java JDK.



After clicking on the DOWNLOAD button, you will be redirected to the Downloads page.

You need to check mark the radio button of **Accept License Agreement** and select on the link for your particular Operating System. Choose **Windows** Operating System to download the **.exe** installation file (do not download**.zip/tar.z/tar.gz/.rpm** extension files).



Java SE Development Kit 8u191

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

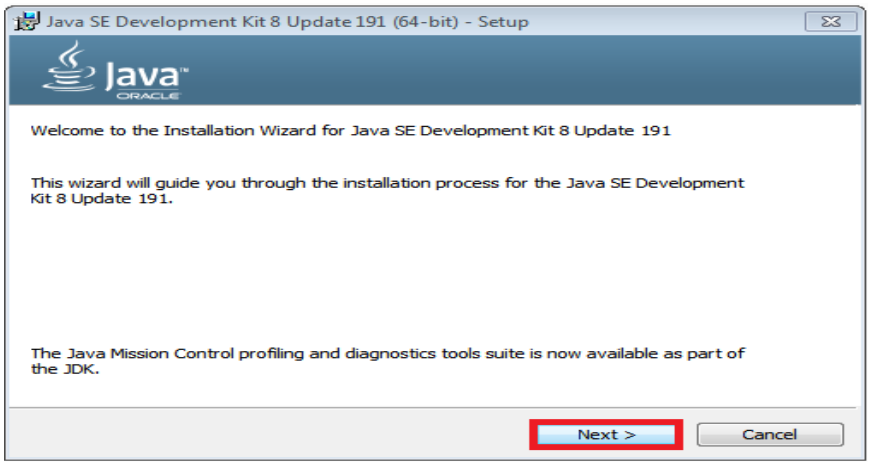
Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux ARM 32 Hard Float ABI	72.97 MB	jdk-8u191-linux-arm32-vfp-hflt.tar.gz
Linux ARM 64 Hard Float ABI	69.92 MB	jdk-8u191-linux-arm64-vfp-hflt.tar.gz
Linux x86	170.89 MB	jdk-8u191-linux-i586.rpm
Linux x86	185.69 MB	jdk-8u191-linux-i586.tar.gz
Linux x64	167.99 MB	jdk-8u191-linux-x64.rpm
Linux x64	182.87 MB	jdk-8u191-linux-x64.tar.gz
Mac OS X x64	245.92 MB	jdk-8u191-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	133.04 MB	jdk-8u191-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	94.28 MB	jdk-8u191-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	134.04 MB	jdk-8u191-solaris-x64.tar.Z
Solaris x64	92.13 MB	jdk-8u191-solaris-x64.tar.gz
Windows x86	197.34 MB	jdk-8u191-windows-i586.exe
Windows x64	207.22 MB	jdk-8u191-windows-x64.exe

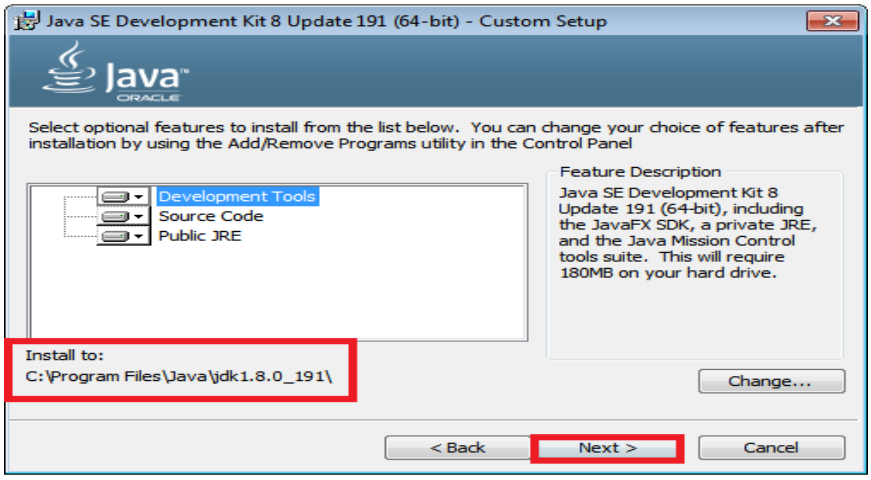
Once the file is saved, use your Windows Explorer to find and run the program by double-clicking on.exe file. Depending on your version of Windows and security settings you may get a security popup. Click on **Run** or **OK** to continue.

When setup is launched you should see the following screen as shown below,

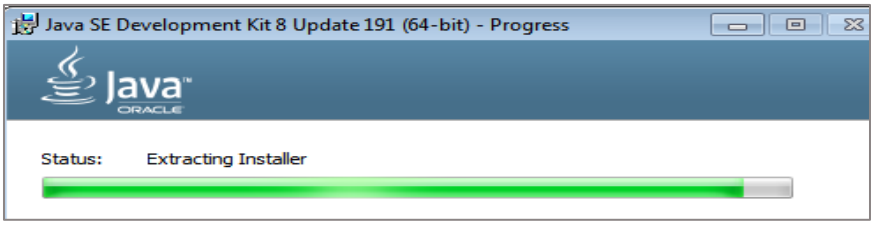
Click on **Next** to continue.



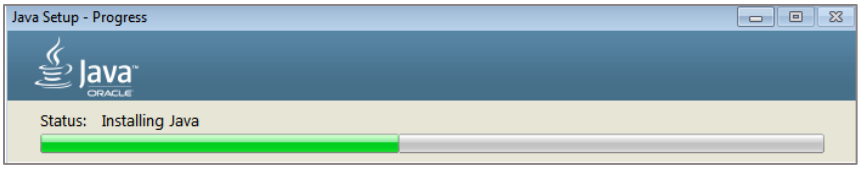
You can accept the default values and simply click on **Next** button to continue. There is no need to make any changes on this screen.



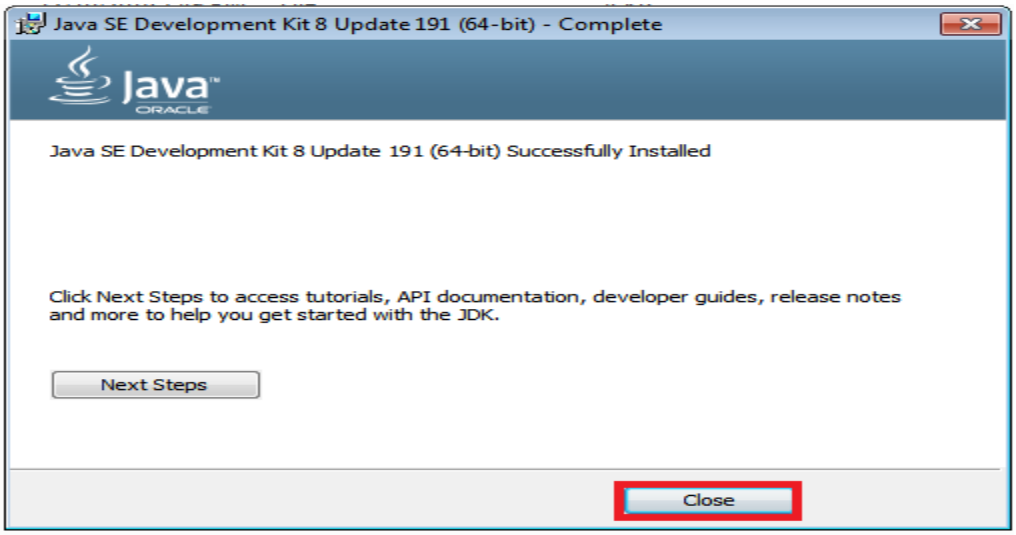
The JRE is automatically installed as a part of the JDK, and you will use the **Install to:** path to access both JRE and JDK components.



The next screen will display a simple progress bar while the JDK files are being installed.



When the JDK is finished installing, you should see a confirmation page similar to the one below. Simply click the **Close** button to finish.



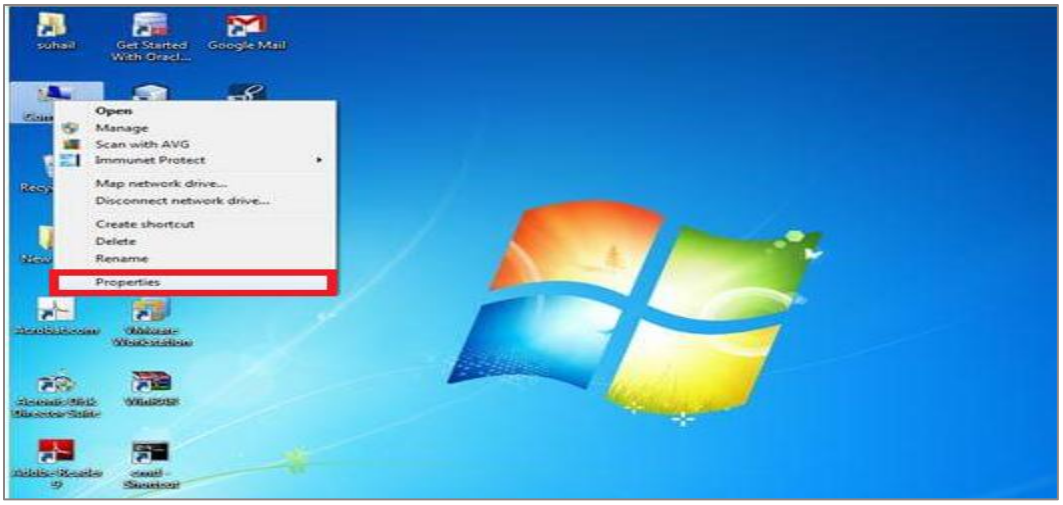
You have to set the path for using tools such as javac, java, etc.

There are two ways to set the path in Java,

- Temporary
- Permanent

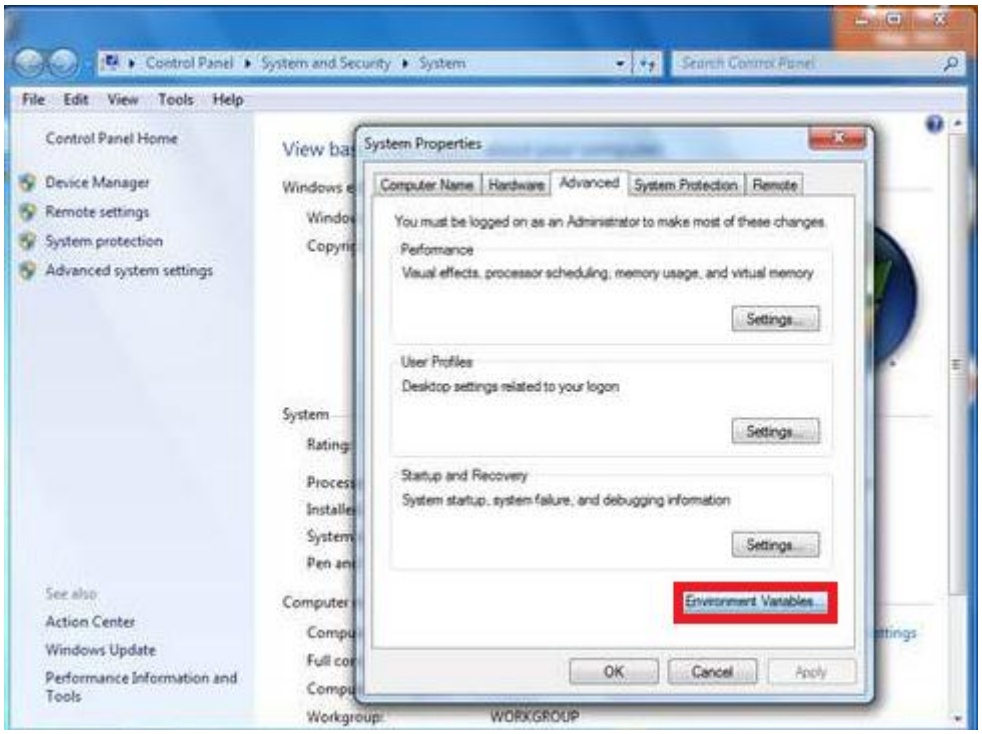
Step #2 | Set Permanent Path of JDK in Windows

Go to My Computer **Properties** on your desktop.

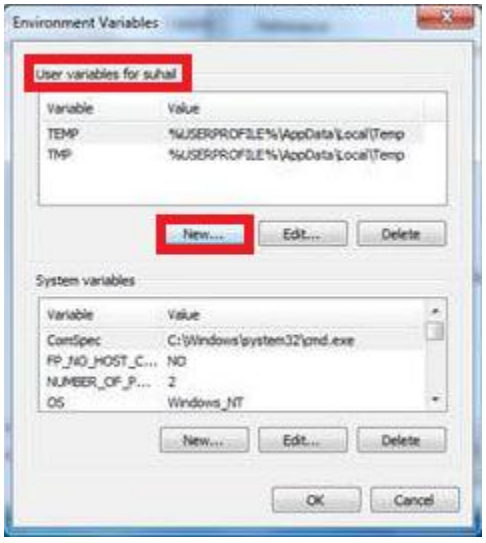


Click on the **Advanced systems settings** tab.

It displays a pop-up with all the system properties. Click on **Environment Variables.**



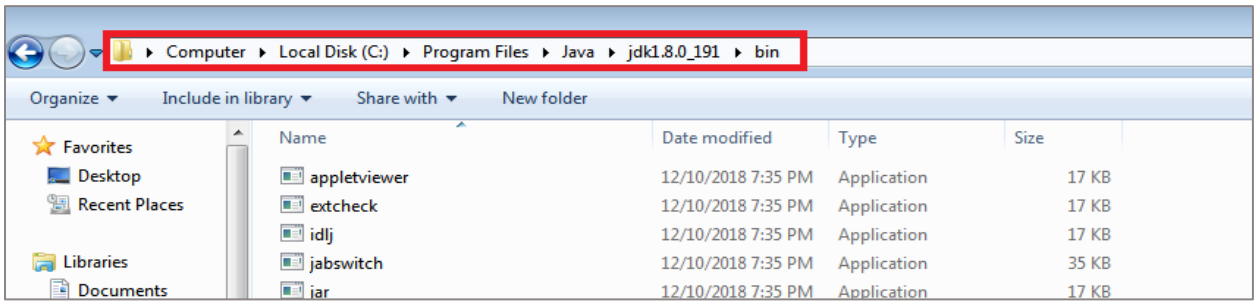
In the Environment Variables, click on the **New…** tab of User variables.



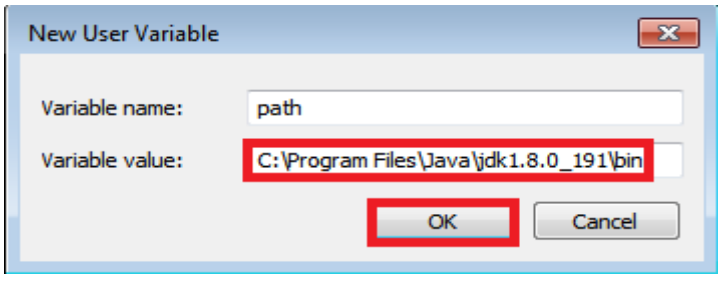
Provide the variable name as **path**.

In your system, Local Disk (C) --> Program Files --> Java --> jdk1.8.0_191--> bin.

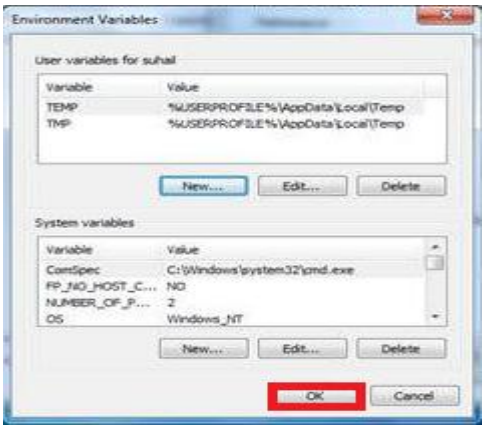
Now, copy the path from the address bar as shown below.



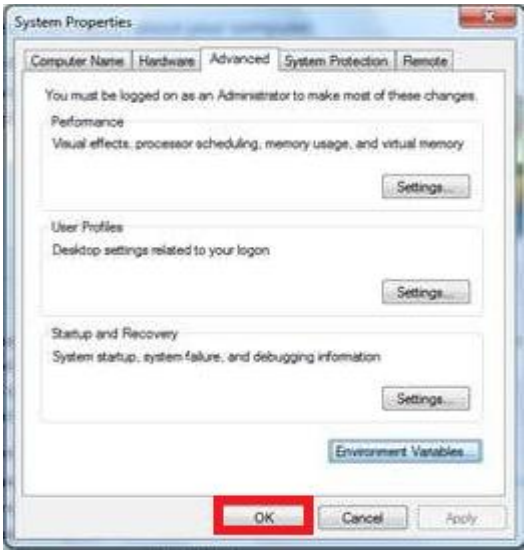
Go back to the User Variables, Paste the path of bin folder in the Variable value and click on **OK** as shown below.



You will be redirected back to Environment Variables, click on **OK**.



Now, you can see the pop-up for System Properties. Click on **OK**.



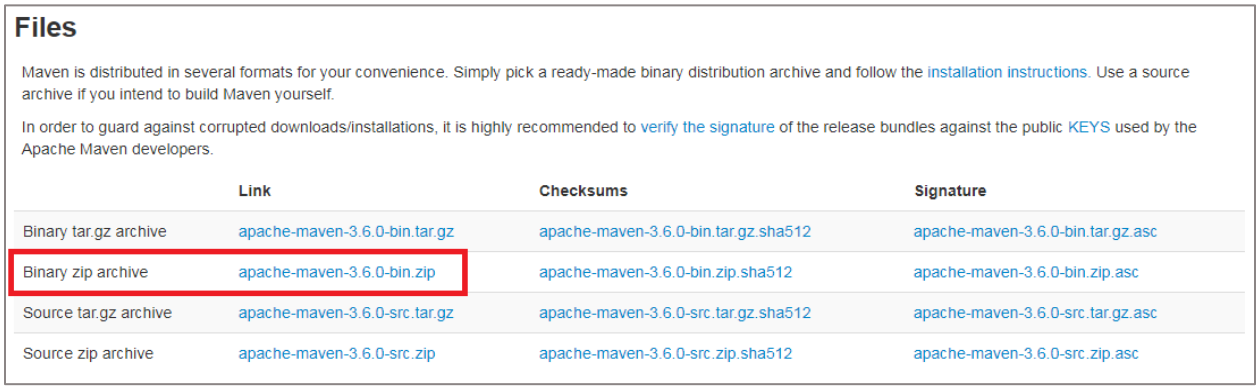
Finally, your permanent Java path is successfully set.

Note: To check if Java is successfully installed on your system or not. Go to command prompt and give command **java -version**

Step #3 | Maven Installation on Windows

Maven is written in Java (and primarily used to build Java programs). Thus, the major pre-requisite is the Java SDK and you should install it to a pathname.

Simply pick a ready-made binary distribution archive from the following link, **https://maven.apache.org/download.cgi**. Download **Binary zip archive** as shown below



Files

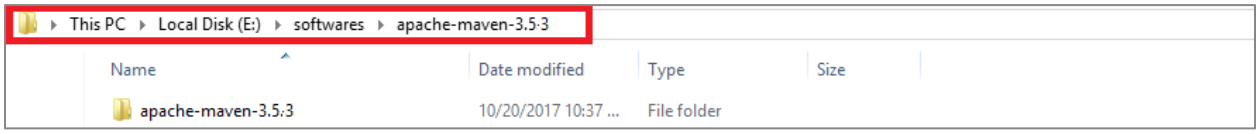
Maven is distributed in several formats for your convenience. Simply pick a ready-made binary distribution archive and follow the installation instructions. Use a source archive if you intend to build Maven yourself.

In order to guard against corrupted downloads/installations, it is highly recommended to verify the signature of the release bundles against the public KEYS used by the Apache Maven developers.

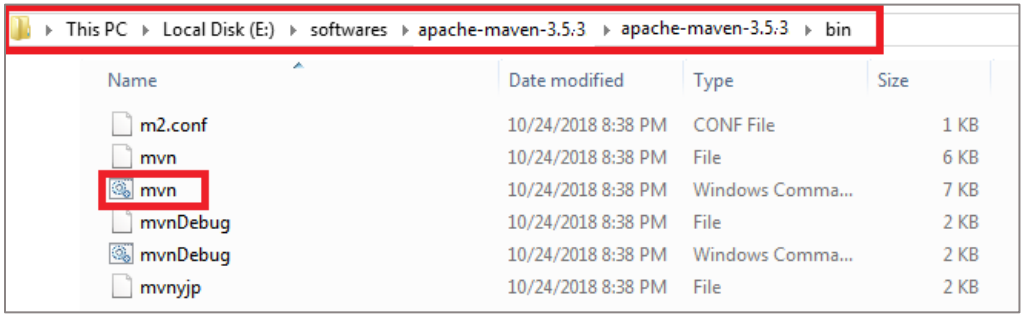
	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.6.0-bin.tar.gz	apache-maven-3.6.0-bin.tar.gz.sha512	apache-maven-3.6.0-bin.tar.gz.asc
Binary zip archive	apache-maven-3.6.0-bin.zip	apache-maven-3.6.0-bin.zip.sha512	apache-maven-3.6.0-bin.zip.asc
Source tar.gz archive	apache-maven-3.6.0-src.tar.gz	apache-maven-3.6.0-src.tar.gz.sha512	apache-maven-3.6.0-src.tar.gz.asc
Source zip archive	apache-maven-3.6.0-src.zip	apache-maven-3.6.0-src.zip.sha512	apache-maven-3.6.0-src.zip.asc

Step #4 | Setting Maven Path on Windows

Unpack the Maven distribution in any one of the drive on your local machine. Extract the .zip file into a specified folder with the same name.



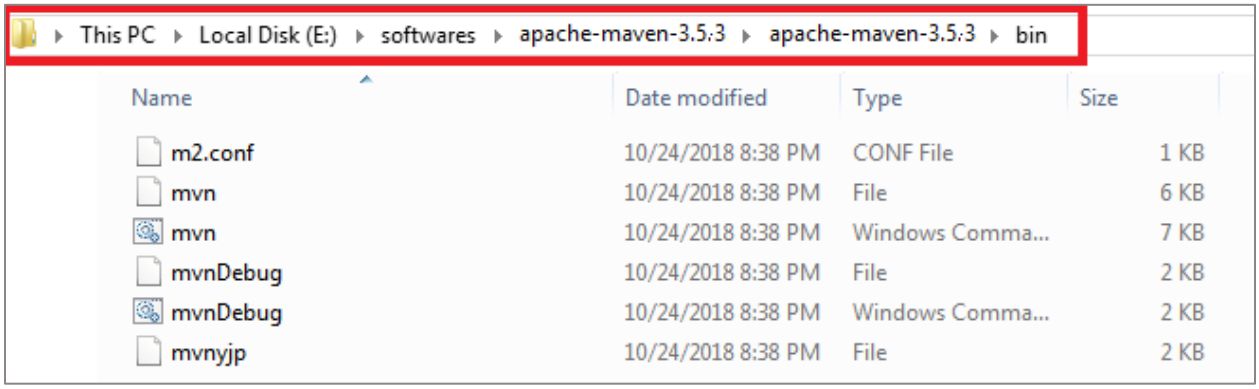
Run the Maven by invoking a command-line tool: **mvn.cmd** from the bin directory of the Maven.



Double click on **mvn** and run the Maven software.

Open the Environment Variables as shown in **Step #2 |Set permanent path of JDK in Windows.**

Goto the folder where you had extracted Maven and copy the Maven path upto **\bin**



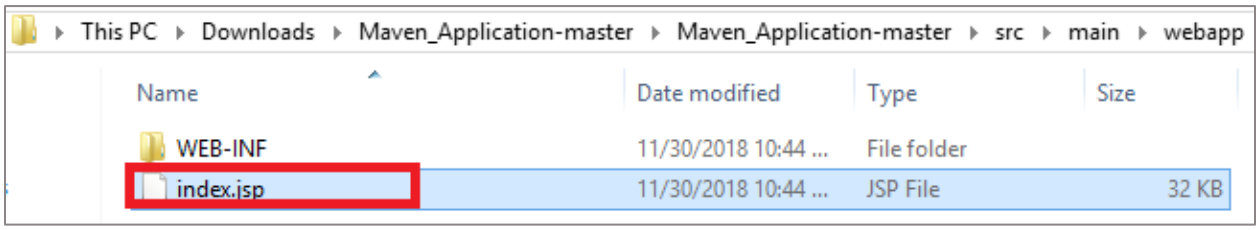
Now go back to Environmental variables and set User Variables with the above path and click on **OK**.

Open your command prompt (cmd) and give the command **mvn -version**.

In the above screenshot, Apache Maven version is 3.5.3

Step #5 | Create .war File

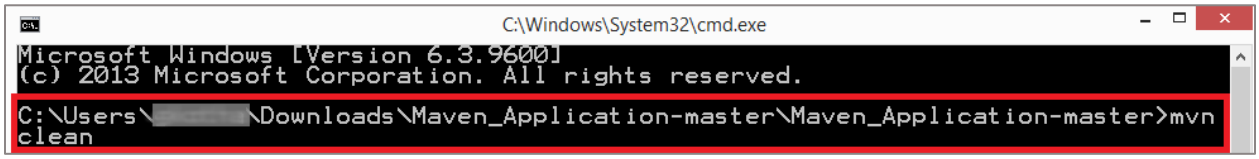
After installing Maven on your system, open source code which you have downloaded from GitHub and open **index.jsp** file through notepad, which is in the path: **src--> main--> webapp**



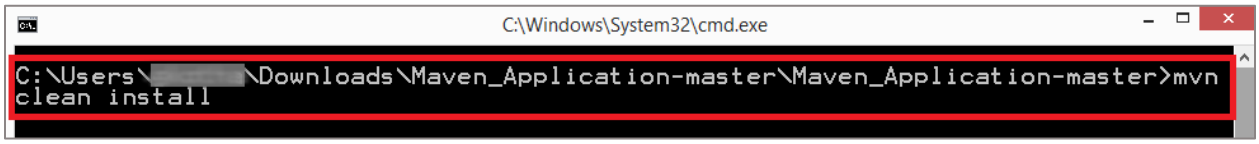
Provide the details of your Database Server i.e, URL, Port, Database Name Username, and Password and save the file.

```
<%
try
{

 Class.forName("com.mysql.jdbc.Driver");
    String url="jdbc:mysql://digitalsummit.cwyafhmsarvs.us-east-1.rds.amazonaws.com:3306/miracle?useSSL=false";
    String username="miracle";
    String password="Miracle123";
    String query="select * from tblDS18Colleges";

    Connection conn=DriverManager.getConnection(url, username, password);
    Statement stmt=conn.createStatement();
    ResultSet rs=stmt.executeQuery(query);
    while(rs.next())
    {
%>
```

Open command line interface and set the path to your application. Enter the command **mvn clean** which executes Maven, and its build life cycle named clean.



Enter the **mvn clean package** or **mvn clean install** to execute your Maven application.

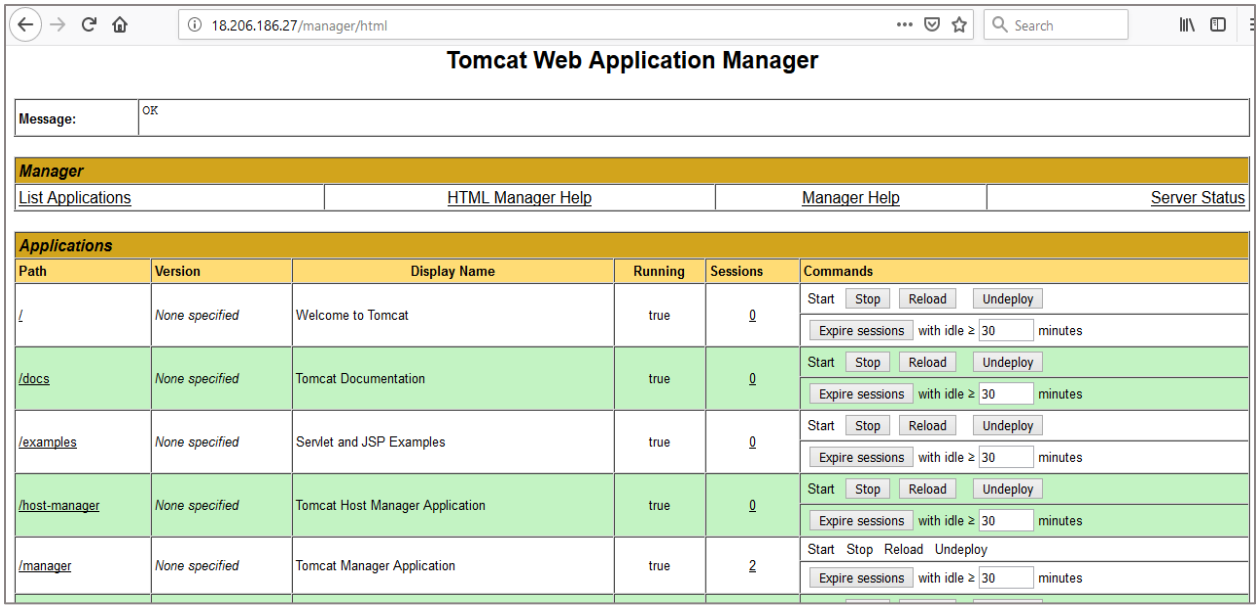


The target folder will be created and in that, you can find your application's **.war** file.

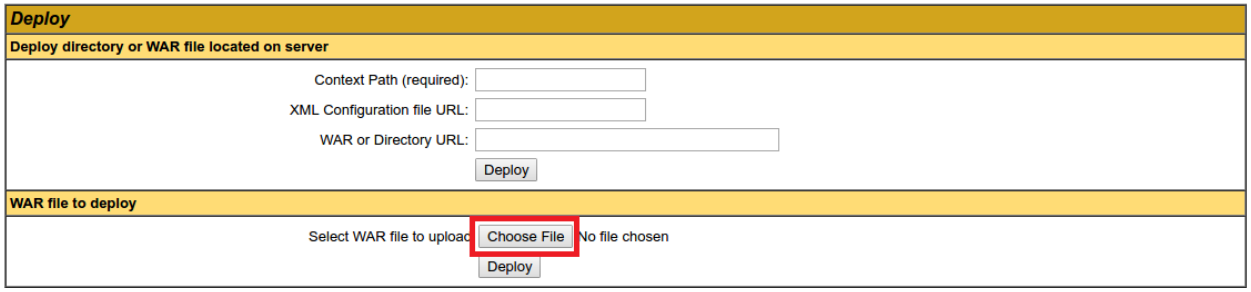
Now, it's time to access your application through **Tomcat**.

Give your EC2 instance IP address where your Tomcat is installed along with Port in your browser. Now, provide your Username and Password to open the **Manager App**.

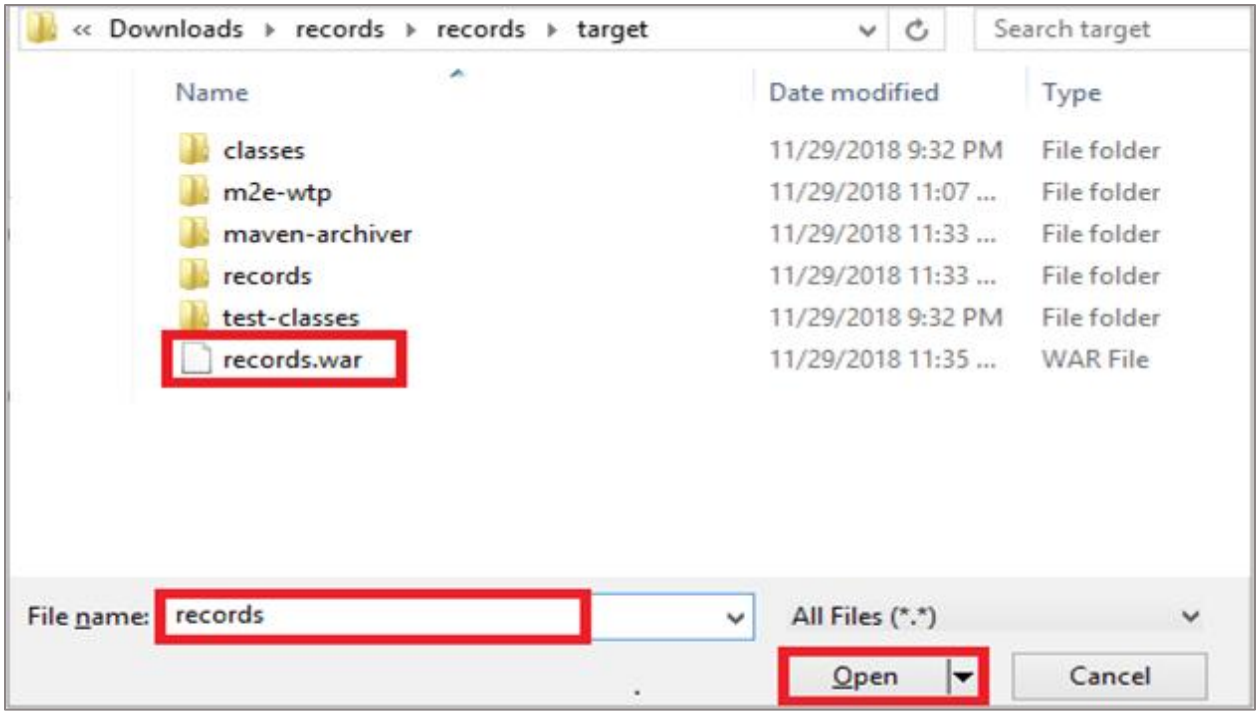
After opening **Manager App** page, you will get the list of all deployed contexts.



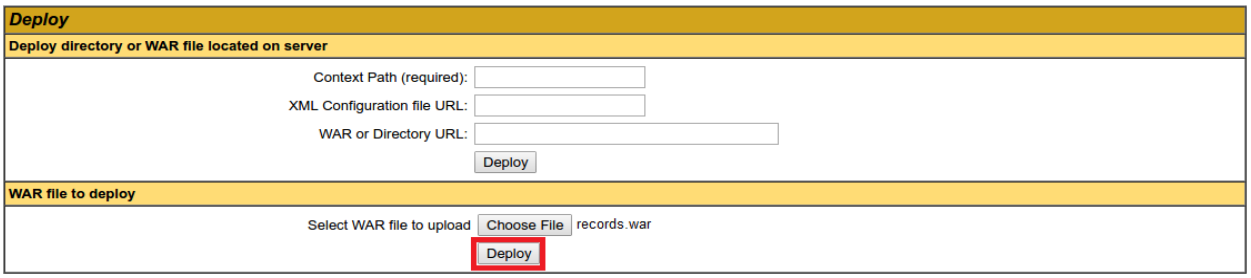
Scroll down to upload the **.war** file of your application, click on **Choose File** as shown below.



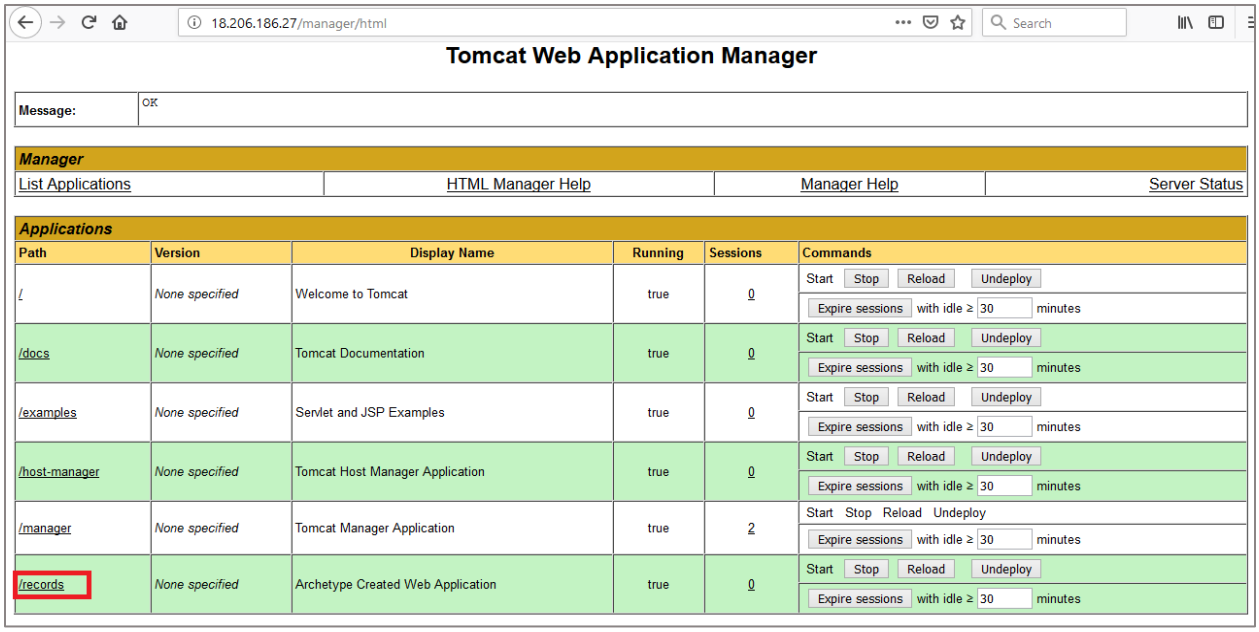
Select **.war** file from the application folder in your local machine.



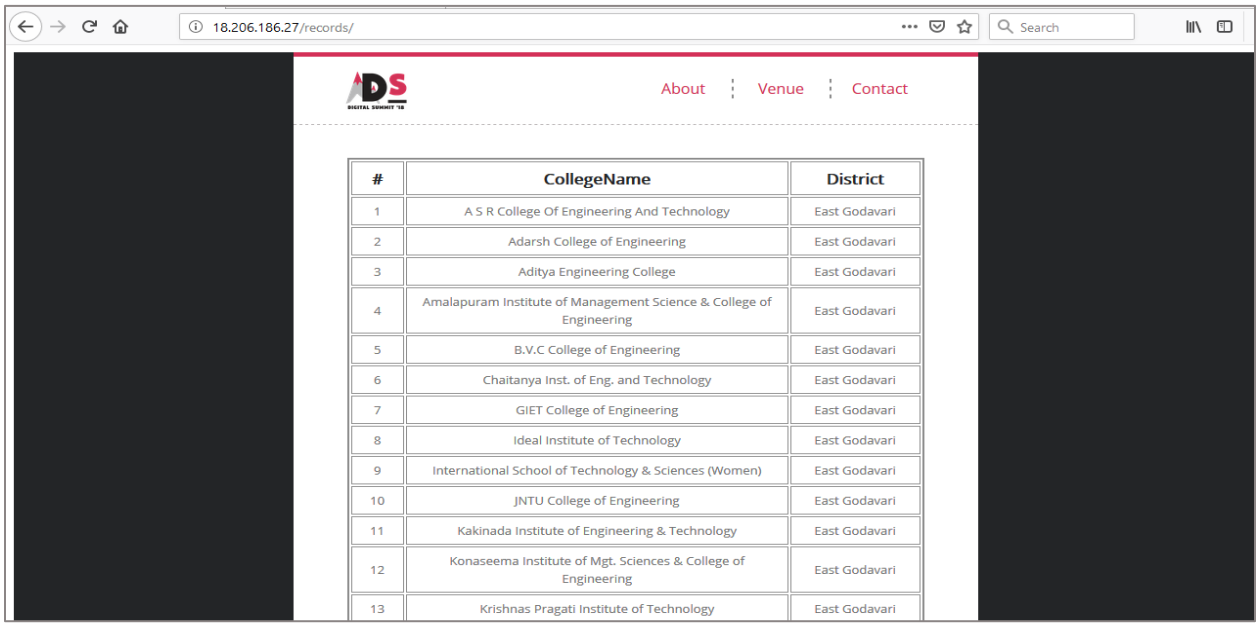
Click on **Deploy**.



After deployment of your **.war** file, you will get **/records** in the deployed contexts as shown below.



Click on **/records** to access your application in your browser as shown below.



For any questions regarding the lab please feel free to reach out to **innovation@miraclesoft.com**. **We hope you enjoyed this!**