**How set Global Home for Java/Maven:**

login with: sudo su -

[ec2-user@ip-172-31-31-132 ~]$ sudo su –

Add hostname: e.g. if this is Tomcat server and instance tag name also updated accordingly

[root@ip-172-31-31-132 ~]# hostname Tomcat

Login back as su to make hostname effective:

[root@ip-172-31-31-132 ~]# sudo su –

[root@Tomcat ~]# yum update -y

Install Java8:

[root@Tomcat ~]# yum install java-1.8\* -y

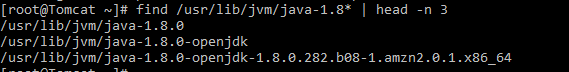
Set Java Home:

Search Java location:

1st Method:

# find /usr/lib/jvm/java-1.8\* | head -n 3

(Java path is shown with Instance Type e.g. Amazon

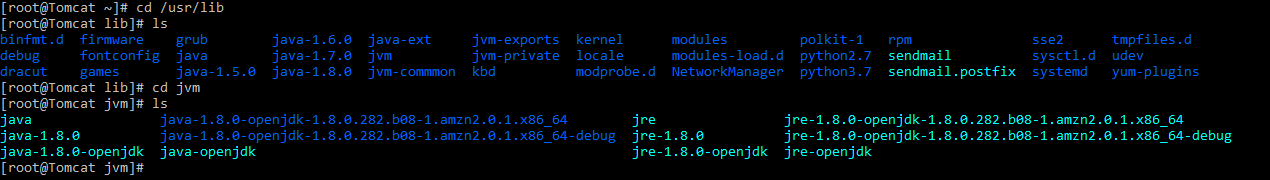


2nd Method

[root@Tomcat ~]# whereis java

java: /usr/bin/java /usr/lib/java /etc/java /usr/share/java /usr/share/man/man1/java.1.gz

Java is inside JVM container and this is way we can identify java location,



# java -version

openjdk version "1.8.0\_282"

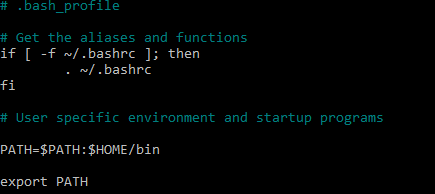
OpenJDK Runtime Environment (build 1.8.0\_282-b08)

OpenJDK 64-Bit Server VM (build 25.282-b08, mixed mode)

**Set Java/Maven Home so that it can be located globally:**

We edit the bash profile

This is the orginal Bash Profile:



Set the Java Home Path:

Java location is /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86\_64 so we add the line below uner User specific environment.

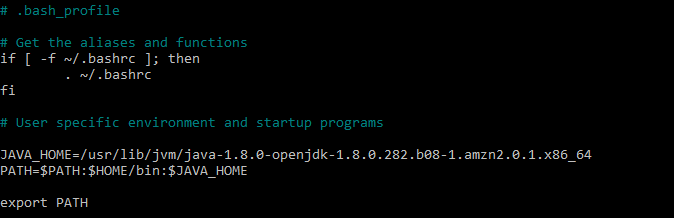
JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.282.b08-1.amzn2.0.1.x86\_64

Now change the PATH environment by adding the Java Home definition:

PATH=$PATH:$HOME/bin:JAVA\_HOME

The updated Java Global profile will is below:

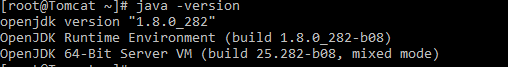
# nano ~/.bash\_profile



Save the environment:

# source ~/.bash\_profile

Now the Java is set globally:



**Set the Maven Global Home variable:**

Locate the Maven Installation: e.g. we have installed Maven in /opt/maven folder then path is-

/opt/maven/apache-maven-3.6.3/bin

Edit the bash profile and update it by using Maven folder M2.

**M2**: A settings. xml file that contains global settings for all **maven** executions.

Add below in bash profile:

M2\_HOME=/opt/maven/apache-maven-3.6.3

M2=/opt/maven/apache-maven-3.6.3/bin

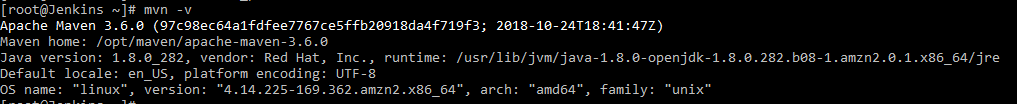
PATH=$PATH:$HOME/bin:$JAVA\_HOME:$M2\_HOME:$M2

# nano ~/.bash\_profile and # source ~/.bash\_profile

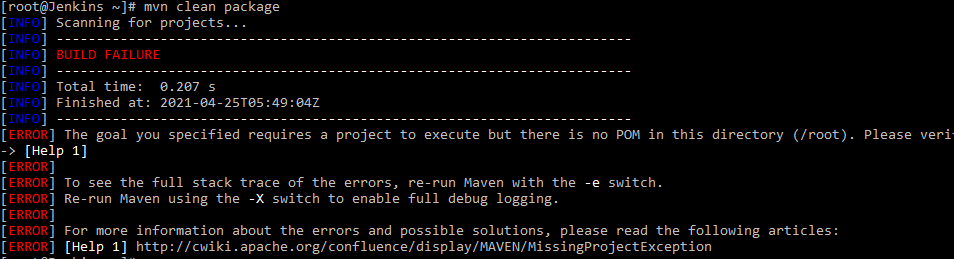


Verify Maven Installation:

# mvn -v



# mvn clean package

(This is the best way to ensure Maven Path is correct as we actually build Maven this way in the Jenkins Build job)

**Tomcat Installation:**

We have installed Apache Tomcat in /opt directory.

Customization for Tomcat working by editing files tomcat-users.xml, server.xml, context.xml.

File tomcat-users.xml is used to add users and group inside Tomcat Server to login as well as access folders like Server Status, Manager App (Manger-gui) and Host Manager(admin-gui)

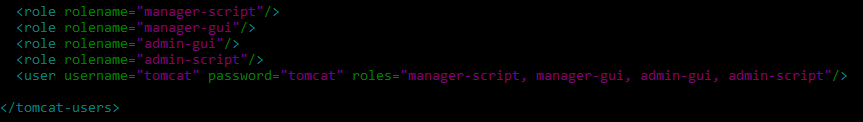
# nano /opt/apache-tomcat-8.5.65/conf/tomcat-users.xml

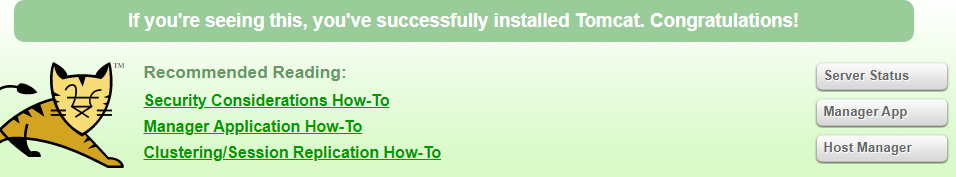
<role rolename="manage-script"/>

<role rolename="manage-gui"/>

<role rolename="admin-gui"/>

<user username="tomcat" password="tomcat" roles="manage-script, manage-gui, admin-gui"/>





Optional in case Jenkins and Tomcat are sharing the same server then port 8080 needs to be changed to 8081 for the Tomcat because Jenkins default port is 8080.

# nano /opt/apache-tomcat-8.5.65/conf/server.xml



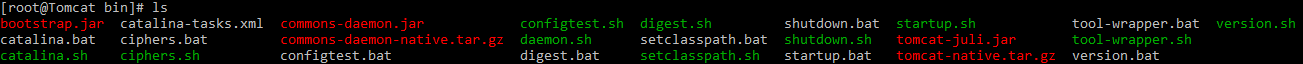
# nano /opt/apache-tomcat-8.5.65/webapps/manager/META-INF/context.xml

(To comment in the below line to make Tomcat accessible remotely because this line makes Tomcat only locally accessible 127.0.0.1)



Stop and start Tomcat service inside bin directory:

# cd /opt/apache-tomcat-8.5.65/bin/



# ./shutdown.sh

# ./startup.sh