Index

Install Oracle Java 8 / 9 in Ubuntu 16.04.

Quick help on how to easily install Oracle Java JDK8 or JDK9 in Ubuntu 16.04 or Linux Mint 18 via PPA.

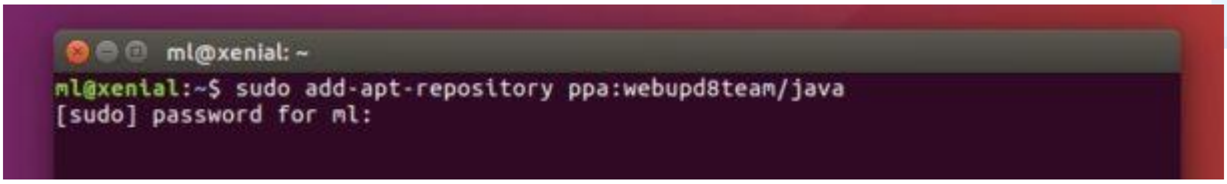
Webupd8 Team is maintaining a PPA repository with installer scripts for the latest Java 8 and 9, that automatically downloads the Java archive from Oracle website and sets up everything for you.

1. Add the PPA.

Open terminal (Ctrl+Alt+T) and run the command:

```
a) sudo add-apt-repository ppa:webupd8team/java
```

Type in your password when it asks and hit Enter.



```
ml@xenial:~$ sudo add-apt-repository ppa:webupd8team/java
[sudo] password for ml:
```

2. Update and install the installer script:

Run commands to update system package index and install Java installer script:

```
b) $ sudo apt-get update
```

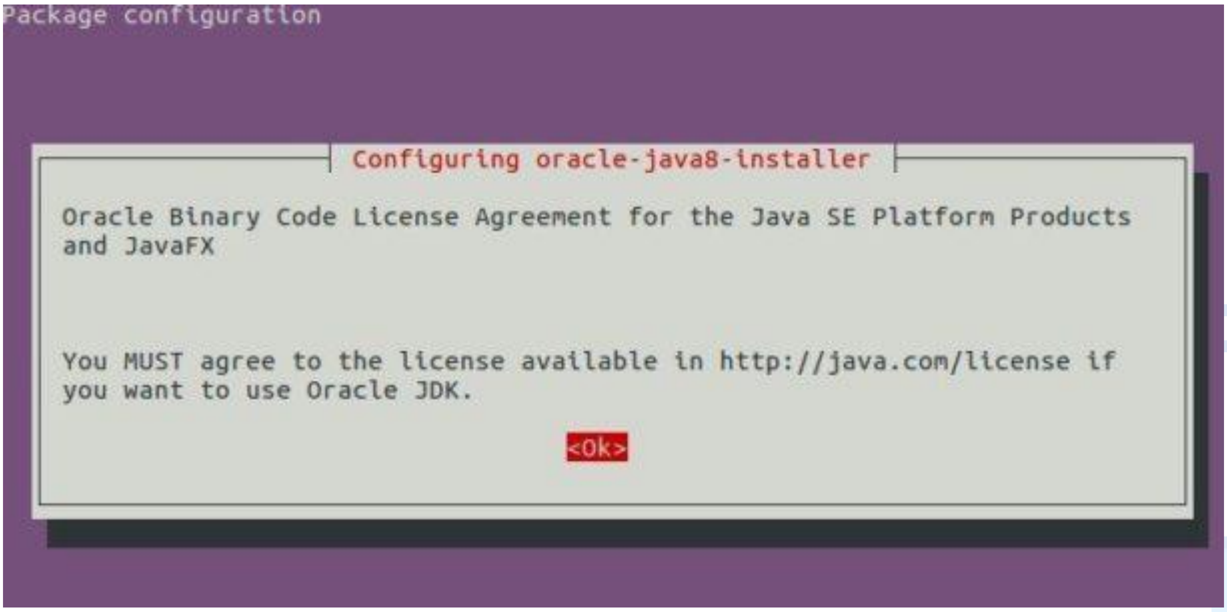
```
c) $ sudo apt-get install oracle-java8-installer
```

Optionally you can run below command to set Oracle Java as default. In most cases this would be executed automatically.

```
d) $ sudo apt-get install oracle-java8-set-default
```

*You may replace oracle-java8-installer with **oracle-java9-installer** to install Java 9.*

While the install process, you have to accept Java license to continue downloading & installing Java binaries.

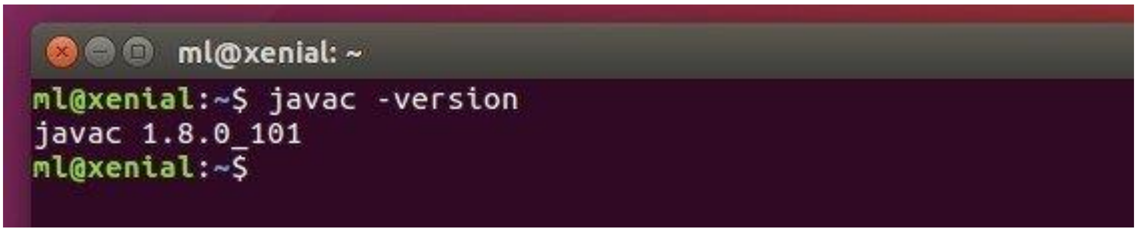


```
Package configuration

Configuring oracle-java8-installer

Oracle Binary Code License Agreement for the Java SE Platform Products
and JavaFX

You MUST agree to the license available in http://java.com/license if
you want to use Oracle JDK.
```

3. Check the Java version

To check the Java version after installing the package, run command:

```
javac -version
```



```
ml@xenial:~$ javac -version
javac 1.8.0_101
ml@xenial:~$
```

4. Set Java environment variables

The PPA also contains a package to automatically set Java environment variables, just run command:

```
sudo apt-get install oracle-java8-set-default
```

For Java 9, install the package ***oracle-java9-set-default*** *instead.*

Oracle JAVA installation on CentOS 6.x / 7.x

Download Oracle JDK 8 latest version from location,

https://www.dropbox.com/s/54vt88vze5aloln/jdk-8u221-linux-x64.rpm?dl=0

Once you have downloaded the rpm package install it using below command.

```
rpm -ivh jdk-8u171-linux-x64.rpm
```

Sample output:

```
Preparing...                          ###########################
###### [100%]

Updating / installing...

   1:jdk1.8.0_25-2000:1.8.0_25-
fcs       ################################# [100%]

Unpacking JAR files...

    rt.jar...

    jsse.jar...

    charsets.jar...

    tools.jar...

    localedata.jar...

    jfxrt.jar...
```

Check Java version

Now, check for the installed JDK version in your system using command:

```
java -version
```

Sample output:

```
java version "1.8.0_25"
```

```
Java(TM) SE Runtime Environment (build 1.8.0_25-b17)

Java HotSpot(TM) 64-Bit Server VM (build 25.25-b02, mixed mode)
```

As you see above, latest java 1.8 has been installed.

Setup Global Environment Variables

We can easily set the environment variables using the export command as shown below.

```
export JAVA_HOME=/usr/java/jdk1.8.0_171-amd64

export PATH=$PATH:$JAVA_HOME
```

Now, let us check for the environment variables using commands:

```
echo $JAVA_HOME
```

Sample output:

```
/usr/java/jdk1.8.0_171-amd64/
```

Or

```
echo $PATH
```

Sample output:

```
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin:/usr/
java/jdk1.8.0_171-amd64/
```

However, the above method is not recommended. Because, the path will be disappeared when the system reboots. To make it permanent, you have to add the paths in the system wide profile.

To do that, create a file called **java.sh** under **/etc/profile.d/** directory.

```
vi /etc/profile.d/java.sh
```

Add the following lines:

```
#!/bin/bash

JAVA_HOME=/usr/java/jdk1.8.0_171-amd64/
```

```
PATH=$JAVA_HOME/bin:$PATH

export PATH JAVA_HOME

export CLASSPATH=.
```

Save and close the file. Make it executable using command:

```
chmod +x /etc/profile.d/java.sh
```

Then, set the environment variables permanently by running the following command:

```
source /etc/profile.d/java.sh
```

That’s it.

What if I didn’t remove the old JDK versions from my system?

As I mentioned before, make sure you have removed all old JDK versions from your system. If you didn’t remove the older versions from your server before installing latest JDK version, you should tell your system from where java should be executed.

By default, the JDK 1.8.x will be installed in /usr/java/jdk1.8.0_171-amd64/ location. In order to tell our system, from where java should be executed, we need to run the following commands one by one.

```
alternatives --install /usr/bin/java java /usr/java/jdk1.8.0_171-amd64/jre/bin/java 20000

alternatives --install /usr/bin/jar jar /usr/java/jdk1.8.0_171-amd64bin/jar 20000

alternatives --install /usr/bin/javac javac /usr/java/jdk1.8.0_171-amd64/bin/javac 20000

alternatives --install /usr/bin/javaws javaws /usr/java/jdk1.8.0_171-amd64/jre/bin/javaws 20000

alternatives --set java /usr/java/jdk1.8.0_171-amd64/jre/bin/java
alternatives --set jar /usr/java/jdk1.8.0_171-amd64/bin/jar

alternatives --set javac /usr/java/jdk1.8.0_171-amd64/bin/javac

alternatives --set javaws /usr/java/jdk1.8.0_171-
amd64/jre/bin/javaws
```

All done. Let us check the alternatives.

```
ls -lA /etc/alternatives/
```

Sample output:

```
lrwxrwxrwx. 1 root root 29 Dec  2 16:24 jar ->
/usr/java/jdk1.8.0_171-amd64/bin/jar

lrwxrwxrwx. 1 root root 34 Dec  2 16:24 java ->
/usr/java/jdk1.8.0_171-amd64/jre/bin/java

lrwxrwxrwx. 1 root root 31 Dec  2 16:24 javac ->
/usr/java/jdk1.8.0_171-amd64/bin/javac

lrwxrwxrwx. 1 root root 36 Dec  2 16:24 javaws ->
/usr/java/jdk1.8.0_171-amd64/jre/bin/javaws

[...]
```

That’s it. Now check for the java version using command:

```
java -version
```

Sample output:

```
java version "1.8.0_171"

Java(TM) SE Runtime Environment (build 1.8.0_171-b11)

Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
```