How To Install Java with Apt on Ubuntu 20.04

1. Introduction
-Java and the JVM (Java's virtual machine) are required for many kinds of software, including Tomcat, Jetty, Glassfish, Cassandra and Jenkins.
-In this guide, you will install various versions of the Java Runtime Environment (JRE) and the Java Developer Kit (JDK) using apt.
-You'll install OpenJDK as well as the official JDK from Oracle.
-You'll then select the version you wish to use for your projects.
-When you're finished, you'll be able to use the JDK to develop software or use the Java Runtime to run software.

2. Prerequisites
-To follow this tutorial, you will need:
One Ubuntu 20.04 server set up by following the the Ubuntu 20.04 initial server setup guide tutorial, including a sudo non-root user and a firewall.

3. Installing the Default JRE/JDK
-The easiest option for installing Java is to use the version packaged with Ubuntu.
-By default, Ubuntu 20.04 includes Open JDK 11, which is an open-source variant of the JRE and JDK.

-To install this version, first update the package index:

```
$ sudo apt update
```
Next, check if Java is already installed:

```
$ java -version
```
If Java is not currently installed, you'll see the following output:
Command 'java' not found, but can be installed with:

```
$ sudo apt install openjdk-11-jre-headless  # version 11.0.11+9-0ubuntu2~20.04, or
$ sudo apt install default-jre              # version 2:1.11-72
$ sudo apt install openjdk-13-jre-headless  # version 13.0.7+5-0ubuntu1~20.04
$ sudo apt install openjdk-16-jre-headless  # version 16.0.1+9-1~20.04
$ sudo apt install openjdk-8-jre-headless   # version 8u292-b10-0ubuntu1~20.04
```

-Execute the following command to install the default Java Runtime Environment (JRE), which will install the JRE from OpenJDK 11:
```
$ sud apt install default-jre
```

-The JRE will allow you to run almost all Java software.
-Verify the installation with:
```
$ java -version
```

-You'll see output similar to the following:
```
openjdk version "11.0.15" 2022-04-19
OpenJDK Runtime Environment (build 11.0.15+10-Ubuntu-0ubuntu0.20.04.1)
OpenJDK 64-Bit Server VM (build 11.0.15+10-Ubuntu-0ubuntu0.20.01.1, mixed mode)
```

-You may need the Java Development Kit (JDK) in addition to the JRE in order to compile and run some specific Java-based software.
-To install the JDK, execute the following command, which will also install the JRE:
```
$ sudo apt install default-jdk
```

-Verify that the JDK is installed by checking the version of javac, the Java compiler:
```
$ javac -version
javac 11.0.15
```

4. Managing Java
-You can have multiple Java installations on one server.
-You can configure which version is the default for use on the command line by using the update-alternatives command.
```
$ sudo update-alternatives --config java
```

-This is what the output would look like if you've installed both versions of Java in this tutorial:
-There are 2 choices for the alternative java (providing /usr/bin/java).

```
  Selection    Path                                        Priority   Status
------------------------------------------------------------
  0            /usr/lib/jvm/java-11-openjdk-amd64/bin/java   1111      auto mode
  1            /usr/lib/jvm/java-11-openjdk-amd64/bin/java   1111      manual mode
* 2            /usr/lib/jvm/java-11-oracle/bin/java          1091      manual mode

-Press <enter> to keep the current choice[*], or type selection number:
-Choose the number associated with the Java version to use it as the default, or press ENTER to leave the current settings in place.
-You can do this for other Java commands, such as the compiler (javac):
    $ sudo update-alternatives --config javac

-Other commands for which this command can be run include, but are not limited to: keytool, javadoc and jarsigner.


5. Setting the JAVA_HOME Environment Variable
-Many programs written using Java use the JAVA_HOME environment variable to determine the Java installation location.
-To set this environment variable, first determine where Java is installed. Use the update-alternatives command:
    $ sudo update-alternatives --config java
-This command shows each installation of Java along with its installation path:
-There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                        Priority   Status
------------------------------------------------------------
  0            /usr/lib/jvm/java-11-openjdk-amd64/bin/java   1111      auto mode
  1            /usr/lib/jvm/java-11-openjdk-amd64/bin/java   1111      manual mode
* 2            /usr/lib/jvm/java-11-oracle/bin/java          1091      manual mode

-Press <enter> to keep the current choice[*], or type selection number:
-In this case the installation paths are as follows:

    --OpenJDK 11 is located at /usr/lib/jvm/java-11-openjdk-amd64/bin/java.
    --Oracle Java is located at /usr/lib/jvm/java-11-oracle/jre/bin/java.

-Copy the path from your preferred installation. Then open /etc/environment using nano or your favorite text editor:
    $ sudo nano /etc/environment

-At the end of this file, add the following line, making sure to replace the highlighted path with your own copied path, but do not include the bin/ portion of the path:

    </etc/environment>
    JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"

-Modifying this file will set the JAVA_HOME path for all users on your system.
-Save the file and exit the editor.
-Now reload this file to apply the changes to your current session:
    $ source /etc/environment

-Verify that the environment variable is set:
    $ echo $JAVA_HOME
    /usr/lib/jvm/java-11-openjdk-amd64


6. Conclusion
In this tutorial you installed multiple versions of Java and learned how to manage them. You can now install software which runs on Java, such as Tomcat, Jetty, Glassfish, Cassandra or Jenkins.
```