How to Install Java on Ubuntu 18.04

Java is one of the most popular programming languages in the world, used for building different types of cross-platform applications.

This tutorial describes how to install various versions of OpenJDK as well as Oracle Java on Ubuntu 18.04. The same instructions apply for Ubuntu 16.04 and any Ubuntu-based distribution, including Kubuntu, Linux Mint and Elementary OS.

Prerequisites

Before continuing with this tutorial, make sure you are logged in as a user with sudo privileges.

Java variations

Java is distributed in three different editions, Standard Edition (SE), Enterprise Edition (EE), and Micro Edition (ME). This tutorial covers the installation of the Java SE (Standard Edition) edition.

OpenJDK and Oracle Java are the two main implementations of Java, with almost no differences between them except that Oracle Java has a few additional commercial features.

There are two different Java packages in Ubuntu repositories, Java Runtime Environment (JRE), and the Java Development Kit (JDK).

If you only want to run Java programs, then you need JRE, which contains only the Java Runtime Environment. Java developer should install JDK, which also includes the development/debugging tools and libraries.

We will show you how to install various Java packages. If you don’t know which Java implementation or version to use, the general recommendation is to stick with the default OpenJDK version available on Ubuntu 18.04.

Installing the Default OpenJDK (Java 11)

At the time of writing, the latest LTS version of Java is version 11.

Follow the steps below to install Java OpenJDK 11 on your Ubuntu system:

1. First, update the `apt` package index with:

   ```
   sudo apt update
   ```

2. Once the package index is updated install the default Java OpenJDK package with:

   ```
   sudo apt install default-jdk
   ```

3. Verify the installation, by running the following command which will print the Java version:

   ```
   java -version
   ```

   The output will look something like this:

   ```
   openjdk version "11.0.2" 2019-01-15
   OpenJDK Runtime Environment (build 11.0.2+9-Ubuntu-3ubuntu118.04.3)
   OpenJDK 64-Bit Server VM (build 11.0.2+9-Ubuntu-3ubuntu118.04.3, mixed mode, sharing)
   ```

That's it! At this point, you should have successfully installed Java on your Ubuntu system.

JRE is included in the JDK package. If you need only JRE, install the `default-jre` package:

Installing OpenJDK 8

Java 8 is still the most widely-used version of Java. If your application requires Java 8, you can install it by typing the following commands:

```
sudo apt updatesudo apt install openjdk-8-jdk
```

Installing Oracle Java

Before installing Oracle Java, make sure you read the Oracle JDK License. The license permits only non-commercial use of the software, such as personal use and development use.

Oracle Java 11 can be installed from the Linux Uprising PPA.

The following steps describe how to install Oracle Java 11 on Ubuntu 18.04:

1. Install the dependencies necessary to add a new repository:

   ```
   sudo apt install software-properties-common
   ```

2. Enable the Linux Uprising PPA by running the following commands:

   ```
   sudo add-apt-repository ppa:linuxuprising/java
   ```

3. Once the repository is added, update the packages list and install the `oracle-java11-installer` package by typing:

   ```
   sudo apt updatesudo apt install oracle-java11-installer
   ```

You will be prompted to accept the Oracle license.

4. Verify the installation by running the following command which will print the R version:

   ```
   java -version
   ```

   ```
   java version "11.0.2" 2019-01-15 LTS
   Java(TM) SE Runtime Environment 18.9 (build 11.0.2+9-LTS)
   Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.2+9-LTS, mixed mode)
   ```

Set the Default Java Version

To check the default Java version you would use the following command:

```
java -version
```

```
openjdk version "11.0.2" 2019-01-15
OpenJDK Runtime Environment (build 11.0.2+9-Ubuntu-3ubuntu118.04.3)
OpenJDK 64-Bit Server VM (build 11.0.2+9-Ubuntu-3ubuntu118.04.3, mixed mode, sharing)
```

If you have multiple Java installations to change the default version, use the `update-alternatives` tool as shown below:

```
sudo update-alternatives --config java
```

```
There are 3 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-8-openjdk-amd64/jre/bin/java   1081      manual mode

Press <enter> to keep the current choice[*], or type selection number:
```

To change the default Java version just enter the version number (the number in the Selection column) and press `Enter`.

Set the JAVA_HOME Environment Variable

Some applications written in Java are using the `JAVA_HOME` environment variable to determine the Java installation location.

To set the `JAVA_HOME` environment variable, first, you need to find out the Java installation paths using the `update-alternatives` command

```
sudo update-alternatives --config java
```

In our case, the installation paths are as follows:

- OpenJDK 11 is located at `/usr/lib/jvm/java-11-openjdk-amd64/bin/java`
- OpenJDK 8 is located at `/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java`

Copy the installation path of your preferred installation. Next, open the `/etc/environment` file:

```
sudo nano /etc/environment
```

Add the following line, at the end of the file:

/etc/environment

```
JAVA_HOME="/usr/lib/jvm/java-11-openjdk-amd64"
```

Copy

Make sure you replace the path with the path to your preferred Java version.

You can either log out and log in or run the following source command to apply the changes to your current session:

```
source /etc/environment
```

To verify that the `JAVA_HOME` environment variable is correctly set, run the following echo command:

```
echo $JAVA_HOME
```

```
/usr/lib/jvm/java-11-openjdk-amd64
```

`/etc/environment` is a system-wide configuration file, which is used by all users. If you want to set the `JAVA_HOME` variable on a per-user basis, add the line to the `.bashrc` or any other configuration file which is loaded when the user logs in.

Uninstall Java

If for any reason you want to uninstall the Java package, you can uninstall it like any other package installed with `apt`.

For example, if you want to uninstall the `openjdk-8-jdk` package run:

```
sudo apt remove openjdk-8-jdk
```

Conclusion

In this tutorial, you learned how to install and manage multiple Java versions on your Ubuntu server.

You can now install applications that run on Java, such as Tomcat, JBoss/WildFly, Apache Maven, Glassfish, Elasticsearch, Cassandra, Jenkins, Gradle ...etc

If you have any questions or feedback, feel free to leave a comment.