Java

All docs related to Java

- Java Installation
  - [How-To] Install Java on Debian 12
  - [How-To] Install OpenJDK 21 on Ubuntu 24 VM

Java Installation

All docs related to Java installation.

[How-To] Install Java on Debian 12

Purpose

This document will explain how to install Java on Debian 12 via Shell commands.

Prerequisites

List of prerequisites:

- Root user or sudo user
- Debian 12 LXC or VM

Java Installation

Step 1: Determine Java Type Needed

OpenJDK and OpenJRE are related but, at the same time, distinct software components used in the Java development and execution environment. Here are the key differences between the two:

1. OpenJDK is an open-source Java SE (Standard Edition) platform implementation. It includes a Java Development Kit (JDK), which provides tools for developing and compiling Java applications, and a Java Runtime Environment (JRE), which is necessary for running Java applications on a user's computer.
2. OpenJRE is a runtime environment for Java applications. It includes the Java Virtual Machine (JVM), the engine that runs Java code, and the Java class libraries, which provide the core functionality of the Java platform. OpenJRE does not include the development tools and compilers that are part of the JDK.

In summary, OpenJDK is a complete Java development and runtime environment, while OpenJRE is a runtime environment only. The JDK component of OpenJDK includes the JRE component but not vice versa.

Therefore, if you are developing Java applications, you will need the OpenJDK, while if you want only to run Java-based ones, the OpenJRE is sufficient.

Step 2: Check for Java

Although Java 11 LTS is now the most widely used version, followed closely by Java 8 LTS, looking ahead and following Oracle's guidance, Java 17 LTS is the way to go. Fortunately, this is precisely the version available in Debian 12's repositories. First, ensure Java is not already installed on your Debian system. Type the following:

```
java
```

Step 3: Install Open JDK

For clarity, we shall state right away that Debian provides the "*default-jdk*" meta-package, a regularly updated one to ship the latest version of the current OpenJDK LTS release for the convenience of its users.

So, using the APT command below will install OpenJDK 17 LTS on your Debian 12 (Bookworm) system.

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

By installing the "*default-jdk*" package, you get both OpenJDK 17 and OpenJRE 17 installed on your system simultaneously.

However, if you only want to install only OpenJDK, you can achieve this by running:

```
sudo apt install openjdk-17-jdk
```

Similarly, if you want to install only the JRE engine, the command you need is:

```
sudo apt install openjdk-17-jre
```

Step 4: Verify the Installation

With the command provided below, verify that the installation was successful. It should output similar to the following:

```
java -version
```

As you can see from the command's output, we now have OpenJDK 17 successfully installed on our Debian 12 system.

Step 5: Set JAVA_HOME Environment Variable

The `JAVA_HOME` variable is used by various applications, such as development environments and build tools, to locate the Java binaries and libraries.

When `JAVA_HOME` is set, the system knows where to look for the JDK and can use it to execute Java applications or compile Java code. So, open the "*/etc/environment*" file with the terminal text editor you are using:

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

Then, add in it the line:

```
JAVA_HOME="/usr/lib/jvm/java-17-openjdk-amd64"
```

Finally, apply the changes and verify everything is correct by running the two commands below.

```
source /etc/environment
echo $JAVA_HOME
```

As we can see from the `echo` command output, the `JAVA_HOME` environment variable is set correctly. Therefore, executing the command below should produce an output like the one shown.

```
ls -l $JAVA_HOME
```

Uninstall Java on Debian 12

Step 1: List Java Packages

Run the following to list java packages to gain visibility of what is installed:

```
sudo dpkg -l | grep 'jdk\|jre'
```

Step 2: Remove Java packages

Then pass them as names after the "*sudo apt purge*" command. In our case, it would look like this:

```
sudo apt purge default-jdk default-jdk-headless default-jre default-jre-headless openjdk-17-jdk openjdk-17-jdk-headless openjdk-17-jre openjdk-17-jre-headless
```

Step 3: Remove Left Behind Java Files

Finally, clean your Debian system of any remaining dependencies.

```
sudo apt autoremove --purge
```

Don't forget to remove the `JAVA_HOME` variable from the "*/etc/environment*" file by simply opening it and deleting the line that contains it.

[How-To] Install OpenJDK 21 on Ubuntu 24 VM

Purpose

This article will focus on the installation of OpenJDK 21 on a headless Ubuntu 24 VM.

Prerequisites

List of prerequisites:

- Sudo user
- Ubuntu 24 LXC or VM

Instructions

Step 1: Install OpenJDK 21

First, update the system:

```
sudo apt update && sudo apt upgrade -y
```

Then, install OpenJDK 21:

```
sudo apt install openjdk-21-jdk -y
```

Step 2: Verify Installation

Finally, verify you have OpenJDK 21 installed by checking the version:

```
java -version
```

You should see output like this:

```
openjdk version "21" 2023-09-19

OpenJDK Runtime Environment (build 21+35-Ubuntu)

OpenJDK 64-Bit Server VM (build 21+35-Ubuntu, mixed mode)
```

Step 3: Set Default Version of Java (If Multiple Exist)

If you have installed more than one version of Java, you will want to select OpenJDK 21 as the default with this command and selecting the correct option for OpenJDK 21:

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

Step 4: Set Java home (Optional)

To set Java_HOME, run:

```
echo 'export JAVA_HOME=$(dirname $(dirname $(readlink -f $(which java))))' >> ~/.bashrc

source ~/.bashrc
```

Verify with:

```
echo $JAVA_HOME
```

Now, OpenJDK 21 is installed and ready to use!