Kafka - Setup and Installation

1. You must have Windows 10 or above
2. Install WSL2
3. Install Java JDK version 11
4. Download Apache Kafka from <https://kafka.apache.org/downloads> under Binary
5. Extract the contents on WSL2
6. Start Zookeeper using the binaries in WSL2
7. Start Kafka using the binaries in another process in WSL2
8. Setup the $PATH environment variables for easy access to the Kafka binaries

**Installing WSL2**

WSL2 is Windows Subsystem for Linux 2 and provides a Linux environment for your Windows computer that does not require a virtual machine

You can use most Linux commands on WSL2 which makes your Kafka installation experience closer to the guides provided for Linux & Mac.

Installing WSL2

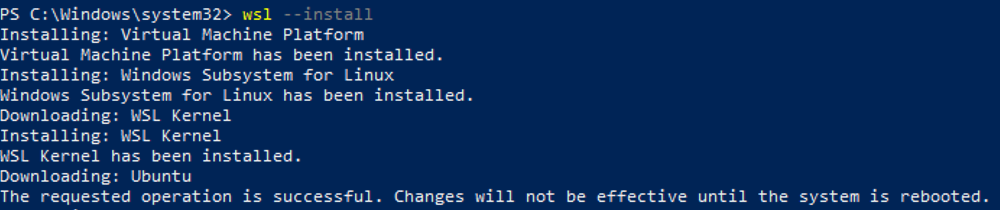
To install WSL2, make sure you're on Windows 10 version 2004 and higher (Build 19041 and higher) or Windows 11.

To check your Windows version do Windows logo key + R, type winver, select OK

You can now install everything you need to run Windows Subsystem for Linux (WSL) by entering this command in **an administrator PowerShell** or Windows Command Prompt and then restarting your machine.

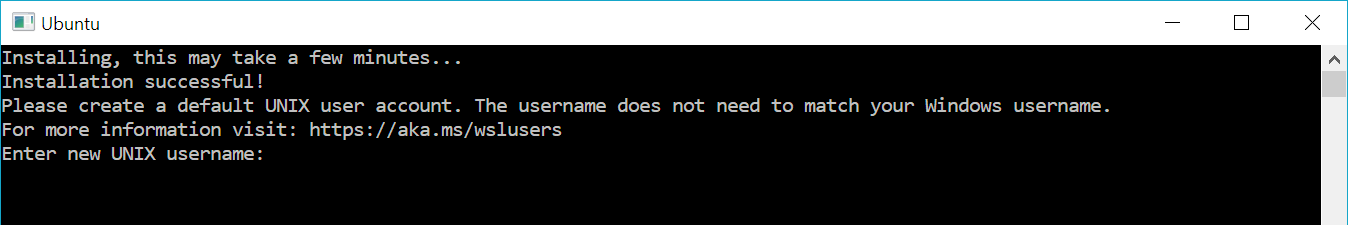
wsl --install

This command will enable the required optional components, download the latest Linux kernel, set WSL 2 as your default, and install a Linux distribution for you *(Ubuntu by default)*.

  
  
The first time you launch a newly installed Linux distribution, a console window will open and you'll be asked to wait for files to de-compress and be stored on your machine. All future launches should take less than a second.

### **Set up your Linux username and password**

Once the process of installing your Linux distribution with WSL is complete, open the distribution (Ubuntu by default) using the Start menu. You will be asked to create a **Username** and **Password** for your Linux distribution.

  
  
This **Username** and **Password** is specific to each separate Linux distribution that you install and has no bearing on your Windows user name.  
  
Once you create a **Username** and **Password**, the account will be your default user for the distribution and automatically sign-in on launch.  
  
This account will be considered the Linux administrator, with the ability to run sudo (Super User Do) administrative commands.

## **Disable IPv6 on WSL2**

WSL2 [currently has a networking issue](https://github.com/microsoft/WSL/issues/4851) that prevents outside programs to connect to Kafka running on WSL2 (for example your Java programs, Conduktor, etc...);

To fix this, we it is recommended to disabe IPv6 on WSL2. Your Windows password will be prompted on the first command:

sudo sysctl -w net.ipv6.conf.all.disable\_ipv6=1

sudo sysctl -w net.ipv6.conf.default.disable\_ipv6=1

### **Installing Java JDK 11**

To install Apache Kafka on WSL2 Ubuntu, Java 11 is the only prerequisite.

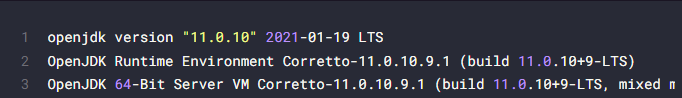
1. Navigate to Amazon Corretto 11 Linux install page and follow the steps.
2. For example on Ubuntu (Debian-based systems).

1 wget -O- https://apt.corretto.aws/corretto.key | sudo apt-key add -

2 sudo add-apt-repository 'deb https://apt.corretto.aws stable main'

3 sudo apt-get update; sudo apt-get install -y java-11-amazon-corretto-jdk

1. Upon completion, you should get a similar output when doing java -version:

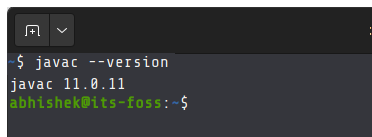


## **Step 1: Set JAVA\_HOME environment variable**

The simplest way to check if Java Development Kit (JDK) is installed on your Linux system is by running this command:

javac --version

The above command checks the version of Java compiler. If it is installed, it will show the Java version.



**Java Compiler is installed**

If the command shows an error like javac command not found, you’ll have to install JDK.

Once you have made sure that Java Compiler is present on your system, it’s time to find its location.

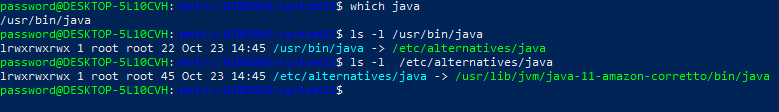
## **Step 2: Get the location of JDK executable (Java Compiler)**

The executable is usually located in the /usr/lib/jvm directory. I won’t left you on your own for a guessing game. Instead, let’s find out the path of the Java executable.

[Use the which command](https://linuxhandbook.com/which-command/" \t "https://itsfoss.com/set-java-home-ubuntu/_blank) to get the location of Java Compiler executable:

which javac

The problem here is that the location it gives is actually a [symbolic link](https://linuxhandbook.com/symbolic-link-linux/" \t "https://itsfoss.com/set-java-home-ubuntu/_blank). You’ll have to follow it a couple of times:



And when you find a path like /usr/lib/jvm/java-11-amazon-corretto/bin/java/bin/javac, you ****remove the /bin/javac**** from it to get something like ****/usr/lib/jvm/java-11-amazon-corretto.****

An easier method is to follow the symbolic link and get to the actual executable file directly using this command:

readlink -f `which javac` | sed "s:/bin/javac::"

The readlink command follows a symbolic link. I have used ` around *which java*. This is called command substitution and it replaces the command with its output. Sed is then used to replace /bin/javac by nothing and thus removing it altogether.

In my example, the location of the executable file is ****/usr/lib/jvm/java-11-amazon-corretto****. It could be different for you. Copy the correct path you got from the above command in your system. You know, you can copy paste in the Ubuntu terminal.

## **Step 3: Setting JAVA\_HOME variable**

Now that you have got the location, use it to set the JAVA\_HOME environment variable:

export JAVA\_HOME=/usr/lib/jvm/java-11-amazon-corretto

Check the value of JAVA\_HOME directory:

echo $JAVA\_HOME



Try to run your program or project in the SAME TERMINAL and see if it works.

This is not over yet. The JAVA\_HOME variable you just declared is temporary. If you close the terminal or start a new session, it will be empty again.

To set JAVA\_HOME variable ‘permanently’, you should add it to the bashrc file in your home directory.

You can [use the Nano editor for editing files in the Linux terminal](https://itsfoss.com/nano-editor-guide/" \t "https://itsfoss.com/set-java-home-ubuntu/_blank). If you do not want that and take a simple copy-paste approach, use the following commands:

Back up your bashrc file (in case you mess it, you can get it back):

cp ~/.bashrc ~/.bashrc.bak

Next, [use the echo command to append](https://linuxhandbook.com/echo-command/" \t "https://itsfoss.com/set-java-home-ubuntu/_blank) the export command you used at the beginning of this section. ***Change the command below to use the correct path as displayed by your system in***

echo "export JAVA\_HOME=/usr/lib/jvm/java-11-amazon-corretto

" >> ~/.bashrc

Verify that it has been correctly added to the end of the file:

tail -3 ~/.bashrc

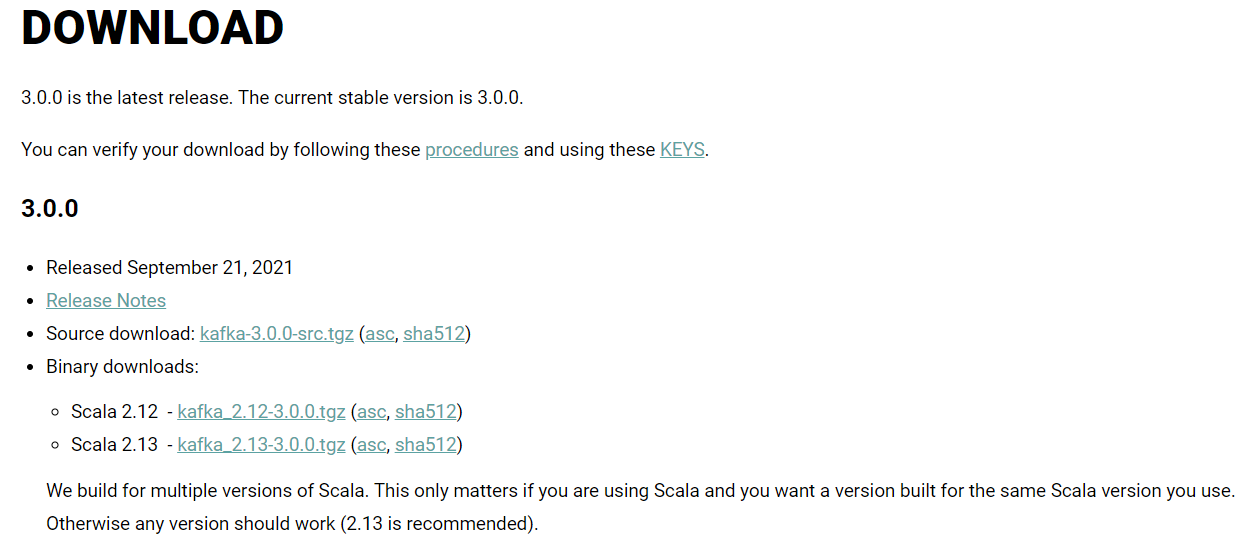
The above [tail command](https://linuxhandbook.com/tail-command/" \t "https://itsfoss.com/set-java-home-ubuntu/_blank) will show the last 3 lines of the specified file.



Now, even if you exit the session or restart the system, the JAVA\_HOME variable will still be set to the value you specified.

### **Install Apache Kafka**

1. Download the latest version of Apache Kafka from <https://kafka.apache.org/downloads> under Binary downloads.

  
  
2. Click on any of the binary downloads (it is preferred to choose the most recent Scala version - example 2.13). For this illustration, we will assume version 2.13-3.0.0.  
Alternatively you can run a wget command:  
  
1 wget <https://archive.apache.org/dist/kafka/3.0.0/kafka_2.13-3.0.0.tgz>  
3. Download and extract the contents to a directory of your choice, for example ~/kafka\_2.13-3.0.0 .  
1 tar xzf kafka\_2.13-3.0.0.tgz   
2 mv kafka\_2.13-3.0.0 ~  
  
4. Open a Shell and navigate to the root directory of Apache Kafka. For this example, we will assume that the Kafka download is expanded into the ~/kafka\_2.13-3.0.0 directory.

### **Opening WSL from a new shell**

1. Open a new Powershell window or Command Prompt as Administrator.
2. You can list all distributions with :

wsl.exe -l

To launch any distribution, you can also use this for commandline:

wsl.exe -d <name\_of\_the\_distribution>

**Like :**

C:\Users\steve>wsl --distribution Ubuntu

password@DESKTOP-5L10CVH:/mnt/c/Users/steve$ cd ~

password@DESKTOP-5L10CVH:~$

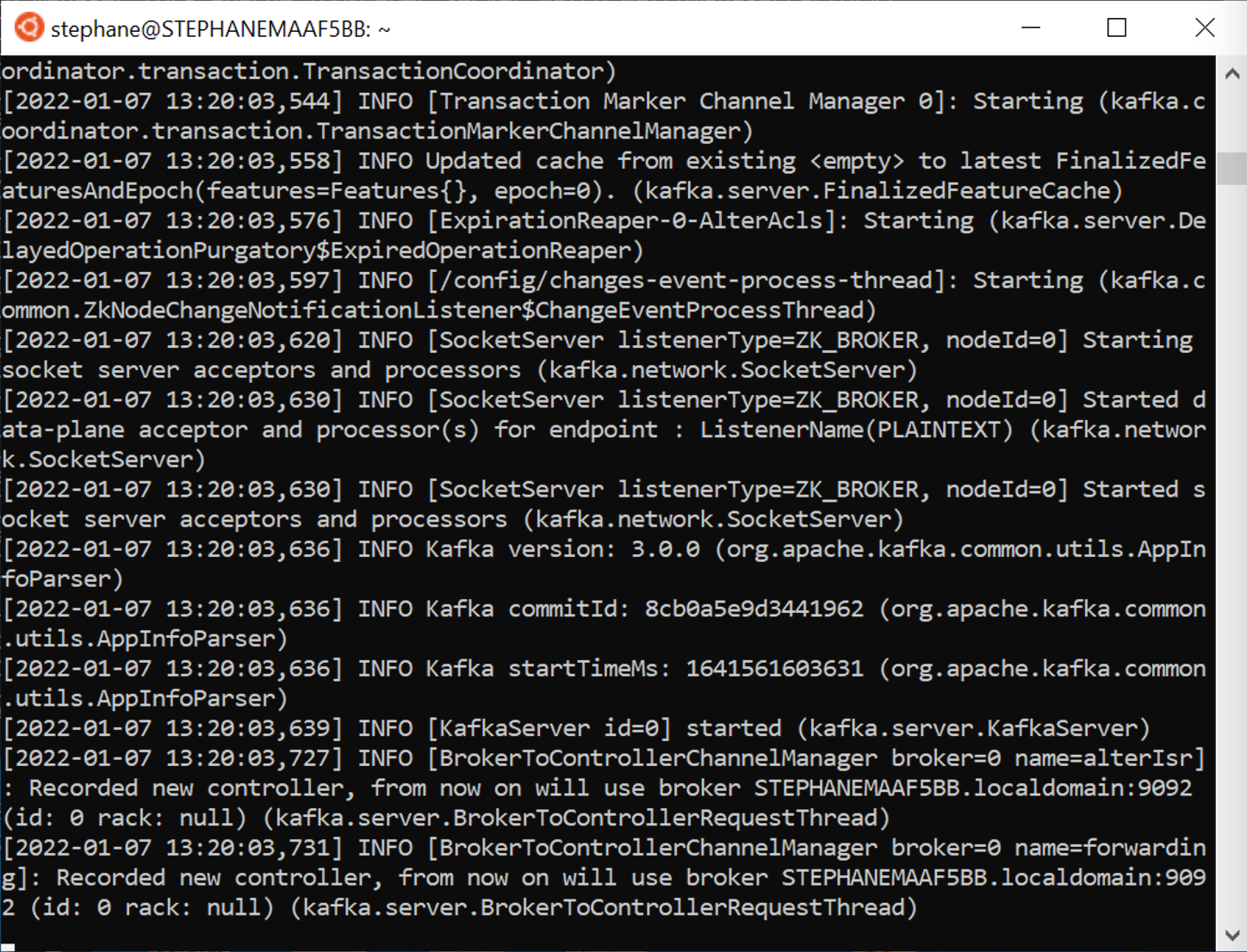
#### **Start Zookeeper**

Apache Kafka depends on Zookeeper for cluster management. Hence, prior to starting Kafka, Zookeeper has to be started. There is no need to explicitly install Zookeeper, as it comes included with Apache Kafka.

Make sure your JAVA\_HOME environment variable is set first, as instructed in the Install Java section, so that Java 11 is used when doing java -version

From the root of Apache Kafka, run the following command to start Zookeeper:

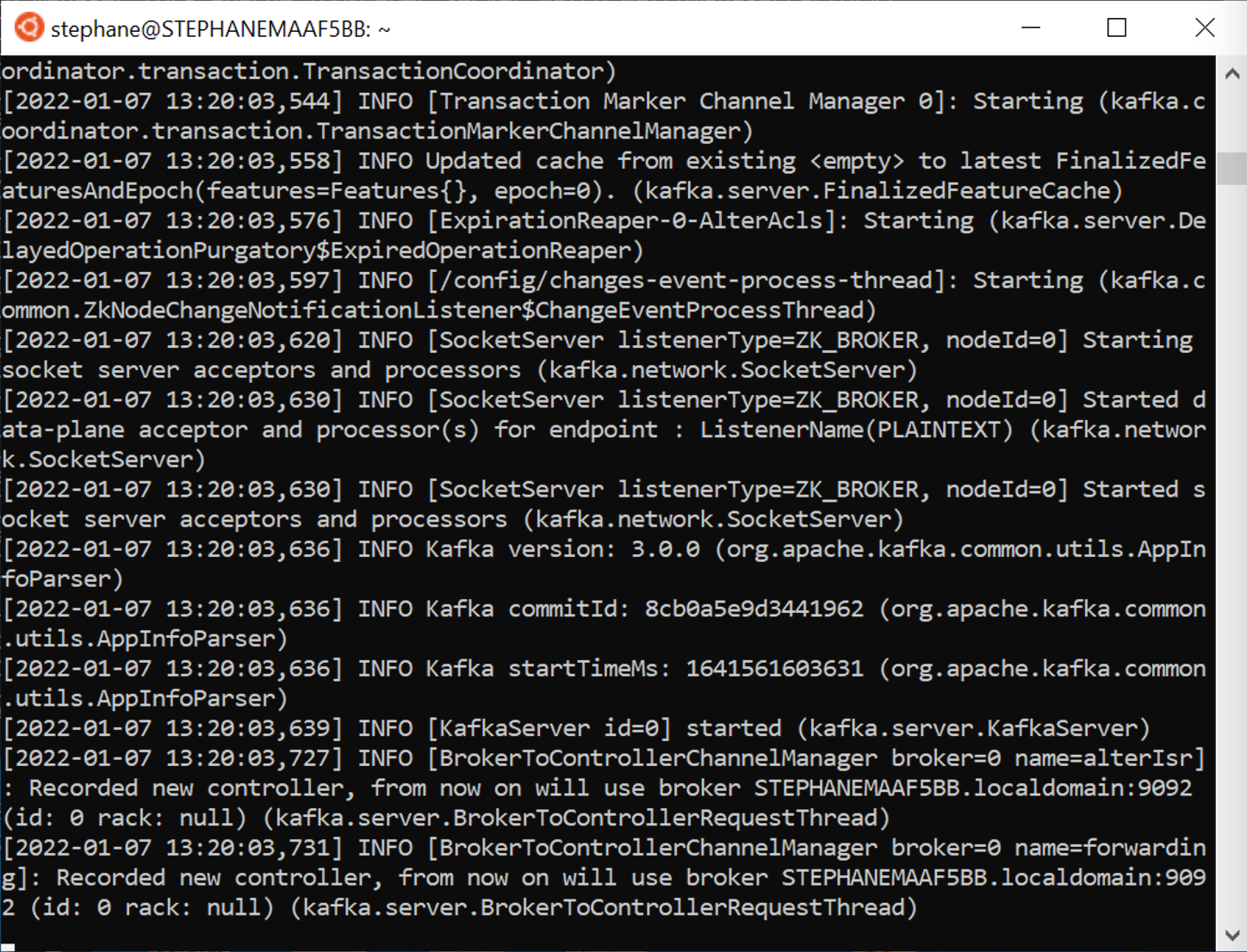
~/kafka\_2.13-3.0.0/bin/zookeeper-server-start.sh ~/kafka\_2.13-3.0.0/config/zookeeper.properties



#### **Start Apache Kafka**

**Open the first Shell window** and run the following command from the root of Apache Kafka to start Apache Kafka.

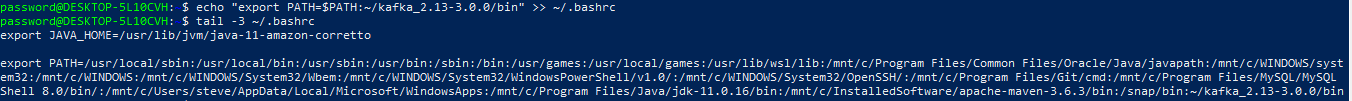
~/kafka\_2.13-3.0.0/bin/kafka-server-start.sh ~/kafka\_2.13-3.0.0/config/server.properties

  
  
Ensure to keep both shell windows opened, otherwise you will shut down Kafka or Zookeeper.  
  
**Kafka is now started, congratulations!**

### **Setup the $PATH environment variable**

In order to easily access the Kafka binaries, you can edit your PATH variable by adding the following line to your system run commands to~/.bashrc :

PATH="$PATH:~/kafka\_2.13-3.0.0/bin"



This ensures that you can now run the kafka commands without prefixing them.

After reloading your shell, the following should work from any directory

kafka-topics.sh

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*