

Introduction To Kafka – Lab:1

What is Kafka?

Imagine a school campus where students need to exchange messages without talking directly. Instead, there's a central bulletin board for each topic, like "Science," "Math," or "Clubs." Students post their messages on these boards, and others can read them based on their interests.

Apache Kafka is a bit like this campus system. It's a tool that allows different parts of a computer system to share information in real-time by "posting" it on virtual boards that other parts can "read" when they need to.

Kafka's Four Main Components

Let's look at the four main components in Kafka, using our bulletin board example

1. Broker – The Bulletin Board

The broker is like the physical bulletin board itself, where messages are posted. It stores all the messages posted by students and organizes them based on different topics.

In Kafka, the broker is a server that manages and stores data, making it available to other parts of the system.

2. Producer – Student Leaders Posting Messages

The producer is like a student leader who writes a message and posts it on a specific board. The student leaders don't need to know who will read it.

The producer creates and sends messages to the broker, by putting them on the appropriate topic for others to read.

3. Topic – The Bulletin Board's Subject

Each bulletin board topic, like "Science" or "Math," holds specific messages. If a student leader posts something about the school science fair, they'll pin it on the "Science" board.

Similarly, In Kafka, a topic is a channel that groups similar types of data. For example, there might be one topic for user activity data and another for system logs.

4. Consumer – The Student Reading Messages

A consumer is like a student who goes to the bulletin board to read messages. They look only at the topics that interest them and ignore the others.

Consumers read messages from specific topics. In Kafka, they can read at their own pace and keep track of which messages they've already read.

How to Install and Run Kafka on A Linux Virtual Machine

In this section, we'll go through the steps to install and start Kafka on a Linux virtual machine. Kafka runs on Java, so we'll start by installing Java.

Step 1: Install Java on Linux

Kafka requires Java to run, so the first step is to install Java.

1. Update the Package List

Open your terminal and type:

“sudo apt update”

2. Install Java Development Kit (JDK)

Type the following command and press Enter:

“sudo apt install default-jdk -y”

This installs Java on your system.

To confirm Java is installed, type:

“java -version”

You should see the Java version information displayed.

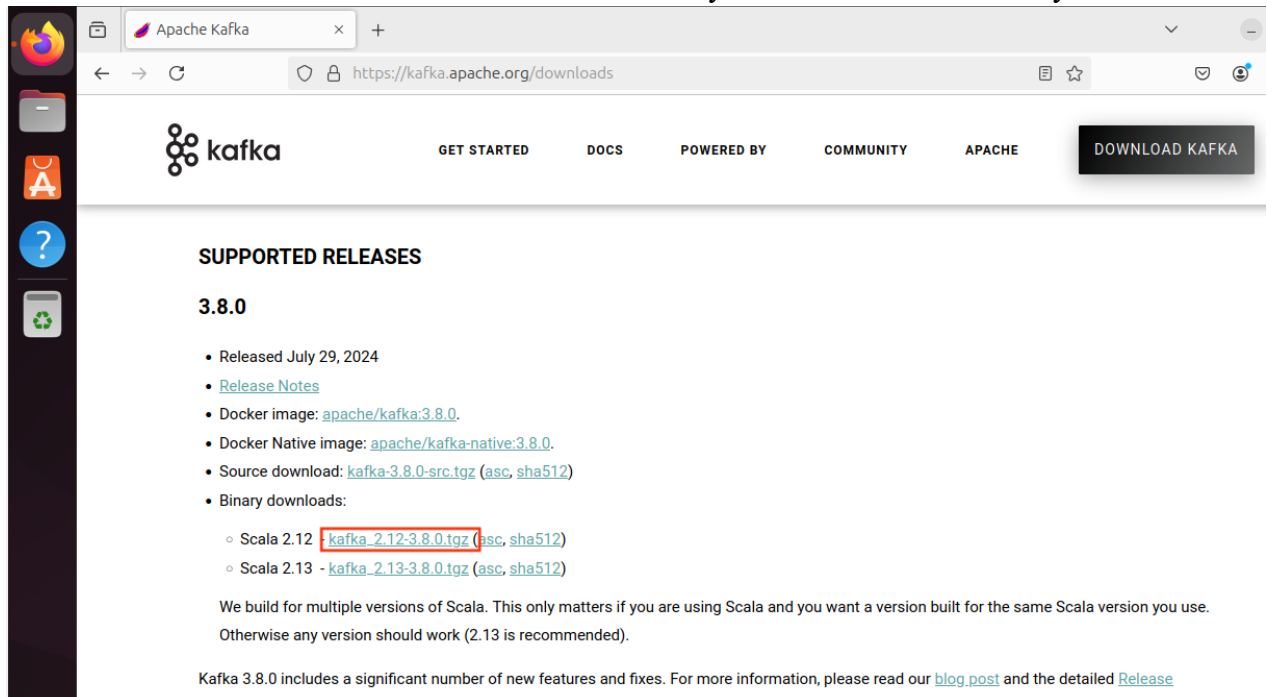
Step 2: Download Kafka

1. Go to the Kafka Download Page:

Visit [Kafka's Official Download Page](https://kafka.apache.org/downloads) in a web browser.

2. Choose the Version:

Select the latest Kafka release and download the Binary files. These files are ready to be used.



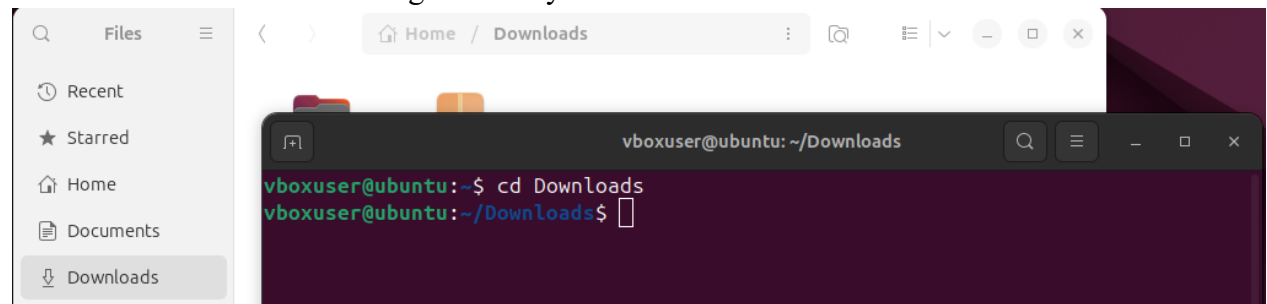
3. Transfer the File to Your VM:

If you downloaded Kafka on your local machine, transfer it your Linux virtual machine. Or you can download it directly on your VM.

4. Navigate to the Downloads Directory:

Once the file is on your VM, open your terminal and navigate to the directory which for most of us will be Downloads. To get to the Downloads directory follow the given commands

Use “cd Downloads” to change directory to Downloads.



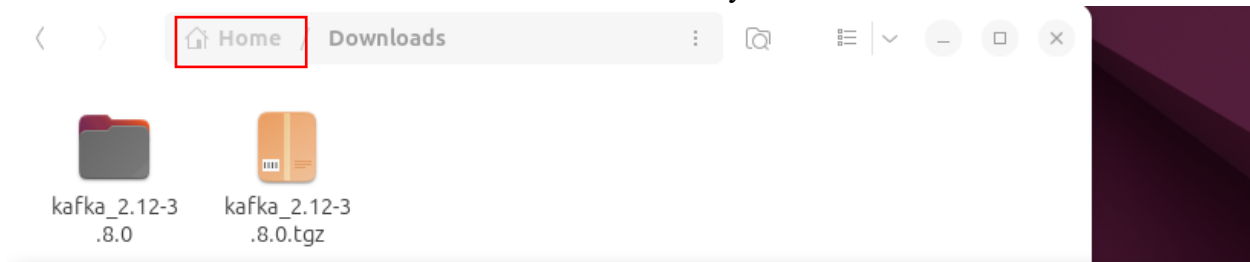
5. Extract the Downloaded File:

Once we get to the directory where the file has been downloaded and/or stored use the following command to extract the file:

“tar -xzf kafka_2.13-2.8.0.tgz”

```
vboxuser@ubuntu: ~/Downloads
vboxuser@ubuntu:~/Downloads$ ls
kafka_2.12-3.8.0.tgz
vboxuser@ubuntu:~/Downloads$
```

The “ls” command shows the list of items in the directory



```
vboxuser@ubuntu: ~/Downloads
vboxuser@ubuntu:~/Downloads$ ls
kafka_2.12-3.8.0.tgz
vboxuser@ubuntu:~/Downloads$ tar -xzf kafka_2.12-3.8.0.tgz
vboxuser@ubuntu:~/Downloads$
```

Once you extract the file you will see the folder named kafka_2.12-3.8.0
Rename the file as “kafka” and move to the home directory.

```
vboxuser@ubuntu: ~
vboxuser@ubuntu:~$ ls
Desktop  Downloads  Music      Public  Templates
Documents kafka      Pictures   snap    Videos
vboxuser@ubuntu:~$
```

Step 3: Start Zookeeper

Kafka requires a tool called Zookeeper to keep track of the broker and manage their configuration.

- Navigate to the Kafka Directory and Start Zookeeper:

Use “cd” command to get to kafka

Then use “bin/zookeeper-server-start.sh config/zookeeper.properties” to start the zookeeper server.

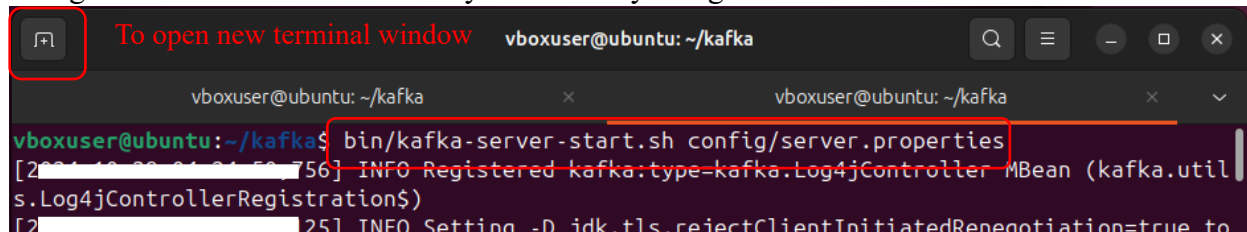
```
vboxuser@ubuntu: ~/kafka
vboxuser@ubuntu:~$ ls
Desktop  Downloads  Music      Public  Templates
Documents kafka      Pictures   snap    Videos
vboxuser@ubuntu:~$ cd kafka
vboxuser@ubuntu:~/kafka$ bin/zookeeper-server-start.sh config/zookeeper.properties
[2023-07-10 10:10:10,886] INFO Reading configuration from: config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2023-07-10 10:10:10,887] WARN config/zookeeper.properties is relative. Please use absolute paths
```

You will see logs in the terminal as Zookeeper starts up. Keep this terminal window open to let Zookeeper keep running.

Step 4: Start the Kafka Broker

Keep the Zookeeper terminal open and open a new terminal window.

- Navigate back to the Kafka directory if needed by using “cd” command.



When you open a new terminal, it mostly will keep you in the Kafka Directory.

In the new terminal start the Kafka broker using the following command:

“bin/kafka-server-start.sh config/server.properties”.

The terminal will display logs showing the Kafka is starting up.

Again DO NOT CLOSE THIS TWO WINDOWS

Step 5: Install Python on Linux VM

In this lab you will be using python code to see how producer side and consumer side works. For that we need to install Python using Terminal.

1. Install Python:

Open Another Terminal Window. When you open another window make sure that the there's no directory opened except your username. To go back to the root directory use “cd ..”

Install Python 3 using the following command:

“sudo apt install python3 -y”

2. Verify Installation:

Confirm the Python was installed by checking the version using following command:

“python3 --version”

3. Install Pip (Python Package Installer):

Pip is required to install additional Python packages, including the kafka-python library, which allows python to interact with Kafka. Use the following command:

“sudo apt install python3-pip -y”

4. Install the kafka-python Library:

With Pip installed, let's install kafka-python, a library for connecting Python code to Kafka.

Use the following command to install it by overriding the errors for an installation:

“pip3 install kafka-python --break-system-packages”

Step 6: Interact with Python Files – Producer.py and Consumer.py

Attached with the lab, there are two Python files named Producer and Consumer. We will also run those files in terminal window. Open another terminal window. When you open another window make sure that the there's no directory opened except your username

1. Run the Producer file

First, we need to run the Producer.py so that the message is on the server. In your Linux VM keep both python file in the home directory.

To run the Python file, use this command – “python3 Producer.py”

2. Run the Consumer file

Open a new terminal window, and run the file using this command – “python3 Consumer.py”

You should see any messages sent by the producer appear in the consumer output.

End of the Lab