

You can contribute any number of in-depth posts on all things data. [Write for Hevo](#) →



How to Install Kafka on Windows? 4 Easy Steps [2022 Guide]

Ishwarya M on Data Streaming, Event Stream Processing, Event Streams, Kafka • January 7th, 2022 • [WRITE FOR HEVO](#)



In the modern world, businesses have to rely on **Real-time Data** for quickly making data-driven decisions and serving customers better. However, often companies use data from the Static Database Server that contains information about users' Historical Data. While Batch Processing operations can be advantageous for building several data-driven solutions, making use of generated data in real-time can provide you with an **edge in the competitive market**.

Currently, many companies and businesses are building and upgrading applications based on real-time user preferences. Real-time customers' data can be used to **enhance** the **ML Models** for seamlessly building Recommendation Systems according to each customer engagement. Today, there are several Data Streaming platforms available for handling and processing real-time infinite or continuous data. One such **Data Streaming Platform** is **Kafka**, which allows you to access or consume real-time data to build event-driven applications.

In this article, you will learn about Kafka, features of Kafka, and how to Install Kafka on Windows Systems!

Table of Contents

- [Prerequisites](#)
- [What is Kafka?](#)
 - [Key Features of Kafka](#)
- [4 Steps to Install Kafka on Windows](#)
 - [Prerequisites](#)
 - [Install Kafka on Windows: Download Kafka](#)
 - [Install Kafka on Windows: Install and Configure Kafka](#)
 - [Install Kafka on Windows: Starting Zookeeper and Kafka](#)

- [Install Kafka on Windows: Testing Kafka by creating a Topic](#)
- [Conclusion](#)

Prerequisites

- Basic understanding of Streaming Data.

What is Kafka?



[Image Source](#)

Kafka is a Distributed Streaming platform that allows you to develop **Real-time Event-driven** applications. In other words, Kafka is a **High-Performance Message Processing** system that enables you to process and analyze a Continuous Stream of information for building real-time Data Pipelines or Applications. Kafka was originally developed by **LinkedIn's Engineering Team** in 2010 for tracking various activity events generated on a LinkedIn webpage or app, such as message exchanges, page views, ads presented, etc. However, in 2011, it was made **Open-source** via Apache Software Foundation, allowing organizations and users to access data that are streaming in real-time for free.

Kafka is also called a **Publish-subscribe Messaging System** because it involves the action of publishing as well as subscribing messages to and from the Kafka server by producers and consumers, respectively. Such efficient capabilities allow Kafka to be used by the most prominent companies worldwide. Based on a [report](#), Kafka is being used by more than 20,500 organizations worldwide, including **80%** of the **Fortune 500 Companies** like **Netflix, Airbnb, Uber, and Walmart**. For instance, based on real-time user engagements, Netflix uses Kafka to provide customers with instant recommendations that allow them to watch similar genres or content.

Key Features of Kafka

- **Real-time Analytics:** With Kafka, you can seamlessly perform analytics operations on data that is streaming in real-time. As a consumer, you can effectively filter and access the real-time or continuous flow of data stored in a Kafka Server or Broker to perform any data-related operations based on the use cases.
- **Fast:** As Kafka Decouples Data Streams, it has very low latency and a very high speed.
- **Consistency:** Kafka is highly capable of handling and processing trillions of data records per day, including petabytes of data. Even though the data is

vast and large, Kafka always maintains and organizes the occurrence order of each collected data. Such a feature allows users to effectively access and consume specific data from a Kafka server or broker based on the use cases.

- **High-Accuracy:** Kafka maintains a high level of accuracy in managing and processing real-time data records. With Kafka, you not only achieve high accuracy in organizing the streaming data but can also perform analytics and prediction operations on the real-time data.
- **Integrations:** Kafka can integrate with other data-processing frameworks or services like Apache Spark, Apache Storm, Hadoop, and AWS. By integrating Kafka with such applications, you can seamlessly incorporate the advantages of Kafka into your Real-time Data Pipelines.
- **Fault tolerance:** Since Kafka replicates and spreads your data frequently to other Servers or Brokers, it is highly fault-tolerant and reliable. If one of the Kafka Servers fails, the data will be available on other servers from which you can easily access the data.

Simplify Kafka ETL with Hevo's No-code Data Pipeline

Hevo Data, a No-code Data Pipeline, helps load data from any data source such as Databases, SaaS applications, Cloud Storage, SDK,s, and Streaming Services and simplifies the ETL process. It supports **100+ Data Sources** including **Apache Kafka, Kafka Confluent Cloud**, and other **40+ Free Sources**. You can use Hevo Pipelines to replicate the data from your **Apache Kafka Source or Kafka Confluent Cloud** to the Destination system. It loads the data onto the desired Data Warehouse/destination and transforms it into an analysis-ready form without having to write a single line of code.

Hevo's fault-tolerant and scalable architecture ensures that the data is handled in a secure, consistent manner with zero data loss and supports different forms of data. Hevo supports two variations of Kafka as a Source. Both these variants offer the same functionality, with Confluent Cloud being the fully-managed version of Apache Kafka.

GET STARTED WITH HEVO FOR FREE

Check out why Hevo is the Best:

- **Secure:** Hevo has a fault-tolerant architecture that ensures that the data is handled securely and consistently with zero data loss.
- **Schema Management:** Hevo takes away the tedious task of schema management & automatically detects the schema of incoming data and maps it to the destination schema.
- **Minimal Learning:** Hevo, with its simple and interactive UI, is extremely simple for new customers to work on and perform operations.
- **Hevo Is Built to Scale:** As the number of sources and the volume of your data grows, Hevo scales horizontally, handling millions of records per minute with very little latency.
- **Incremental Data Load:** Hevo allows the transfer of data that has been modified in real-time. This ensures efficient utilization of bandwidth on both ends.
- **Live Support:** The Hevo team is available round the clock to extend exceptional support to its customers through chat, email, and support

calls.

- **Live Monitoring:** Hevo allows you to monitor the data flow and check where your data is at a particular point in time.

SIGN UP HERE FOR A 14-DAY FREE TRIAL!

4 Steps to Install Kafka on Windows

Here are the simple steps to Install Kafka on Windows:

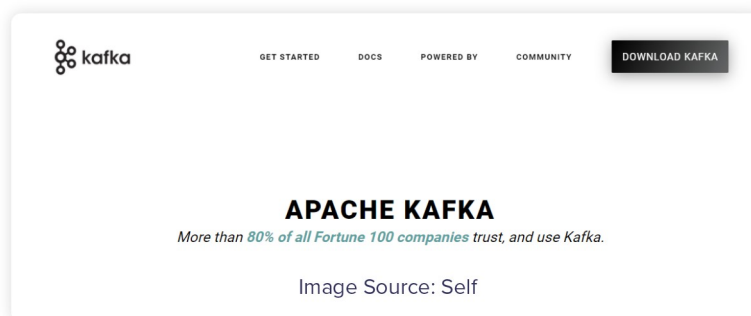
- [Prerequisites](#)
- [Download Kafka](#)
- [Install and Configure Kafka](#)
- [Starting Zookeeper and Kafka](#)
- [Testing Kafka by Creating a Topic](#)

Prerequisites

Before installing Kafka, you should have two applications pre-installed in your local machine.

- **Java Development Kit (JDK):** Java is the most needed prerequisite before installing Kafka on your computer. You can install the [JDK](#) by downloading it from the official Oracle website. Select and download the appropriate installer according to your system type, i.e, **32 bit or 64 bit**. After downloading, you can run the installer by following the on-screen instructions. You should also configure the file path and Java_Home environment variables to enable your operating system to locate the Java utilities. Finally, test your JDK installation by running the command ***java -version*** in your command prompt.
- **7-Zip or WinRAR:** [7-Zip](#) or [WinRAR](#) applications allow you to unzip or extract the downloaded Kafka files.

1) Install Kafka on Windows: Download Kafka



- To install Kafka on Windows, visit the official page of [Apache Kafka](#) and click on the **"Download Kafka"** button.

3.0.0

- Released September 21, 2021
- [Release Notes](#)
- Source download: [kafka-3.0.0-src.tgz](#) (asc, sha512)
- Binary downloads:
 - Scala 2.12 - [kafka-2.12-3.0.0.tgz](#) (asc, sha512)
 - Scala 2.13 - [kafka-2.13-3.0.0.tgz](#) (asc, sha512)

We build for multiple versions of Scala. This only matters if you are using Scala and you want a version built for the same Scala version you use. Otherwise any version should work (2.13 is recommended).

Image Source: Self

- Now, you will be redirected to the downloading page. You can then see the Binary Downloads option. Under that, select the latest Kafka version that is Scala 2.13.



Image Source: Self

- Then, you will be taken to another webpage where you will have the direct download link for downloading your Kafka file.
- Click on the **respective link**. Now, Kafka is successfully downloaded.

2) Install Kafka on Windows: Install and Configure Kafka

- After downloading, **extract or unzip** the Kafka files. Move the extracted folder to any of your preferred directories for quickly accessing it from the command prompt.
- Now, you have to perform some configurations in the extracted Kafka files to properly install Kafka on Windows.
- Usually, extracted Kafka files have Zookeeper files that run simultaneously with Kafka for managing all the Clusters and Configurations of Kafka servers.
- Instead of storing them in default temp folders, you can configure both the Kafka and ZooKeeper files to store Kafka and ZooKeeper data in separate folders.

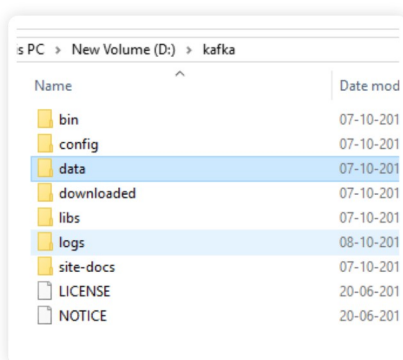


Image Source

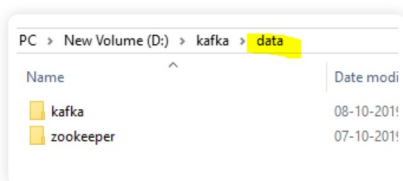


Image Source

- Create a new folder named **"Data"** inside the Kafka folder. Inside the Data folder, create two separate folders named **"Kafka"** and **"Zookeeper."**
- After creating separate folders for Kafka and Zookeeper, you have to make

some changes in the configuration files for pointing to the newly created folders.

- For that, initially copy the file path of the Zookeeper folder created inside the data folder.

```
# the directory where the snapshot is stored.
dataDir=D:/kafka/data/zookeeper
# the port at which the clients will connect
clientPort=2181
```

[Image Source](#)

- From the config folder present inside the extracted Kafka files, open the “**Zookeeper.properties**” file with any text editor applications like Notepad or Notepad++. In the opened file, replace your “**dataDir**” location with the copied Zookeeper folder path, as shown in the above image. Make sure you change the path with forwarding slashes instead of backward slashes. Finally, save the file to update the changes made to the file configurations.
- After configuring the Zookeeper properties, you must configure the Kafka Server properties. For that, copy the file path of the Kafka Folder created inside the data folder.
- Open the server.properties file from the “**Config**” folder present inside the extracted Kafka files.

```
57 ***** log_DIRS *****
58
59 # A comma separated list of directories under which to store log files
60 log.dirs=D:/kafka/data/kafka
61
62 # The default number of log partitions per topic. More partitions allow gr
```

[Image Source](#)

- In the server.properties file, replace the “**logs.dirs**” location with the copied Kafka folder path as shown in the above image. Make sure you change the path with **forward slashes instead of backward slashes**. Replace the backward slashes with forward slashes in the file path and save the file.
- Now, you have made the necessary changes and configurations to the Kafka files and are ready to set up and start Kafka on your computer.

These are the simple steps to install Kafka on Windows.

3) Install Kafka on Windows: Starting Zookeeper and Kafka

After configuring Zookeeper and Kafka, you have to start and run Zookeeper and Kafka separately from the command prompt window.

A) Starting Zookeeper

Open the command prompt and navigate to the **D:Kafka** path. Now, type the below command.

```
zookeeper-server-start.bat ....configzookeeper.properties
```

```
[2019-10-08 15:04:35,816] INFO Server environment:user.home=C:\Users\WaheshAiyil (org.apache.zookeeper.server.ZooKeeperServer)
[2019-10-08 15:04:35,816] INFO Server environment:user.dir=D:\kafka\bin\windows (org.apache.zookeeper.server.ZooKeeperServer)
[2019-10-08 15:04:35,868] INFO tickTime set to 3000 (org.apache.zookeeper.server.ZooKeeperServer)
[2019-10-08 15:04:35,869] INFO minSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2019-10-08 15:04:35,871] INFO maxSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2019-10-08 15:04:35,917] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection factory (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2019-10-08 15:04:35,936] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
```

Image Source

You can see from the output that Zookeeper was initiated and bound to port **2181**. By this, you can confirm that the Zookeeper Server is started successfully. Do not close the command prompt to keep the Zookeeper running.

B) Starting Kafka

Open another command prompt window and type the below command.

```
kafka-server-start.bat ....configserver.properties
```

The Kafka Server has started successfully and is ready for Streaming Data.

Now, both Zookeeper and Kafka have started and are running successfully. To confirm that, navigate to the newly created Kafka and Zookeeper folders. When you open the respective Zookeeper and Kafka folders, you can notice that certain new files have been created inside the folders.

4) Install Kafka on Windows: Testing Kafka by Creating a Topic

As you have successfully started Kafka and Zookeeper, you can test them by creating new Topics and then Publishing and Consuming messages using the topic name. Topics are the virtual containers that store and organize a stream of messages under several categories called Partitions. Each Kafka topic is always identified by an arbitrary and unique name across the entire Kafka cluster.

In the below steps, you will learn how to create topics:

For creating a topic, open a new command prompt and write the below command:

```
.binwindowskafka-topics.bat --create --zookeeper localhost:2181 --repl
```

In the above command, **TestTopic** is the unique name given to the Topic, and **zookeeper localhost:2181** is the port that runs Zookeeper. After the execution of the command, a new topic is created successfully.

When you need to create a new Topic with a different name, you can replace the same code with another topic name. For example:

```
.binwindowskafka-topics.bat --create --zookeeper localhost:2181 --repl
```

In the command, you have only replaced the topic name while other command parts remain the same. To list all the available topics, you can execute the below command:

```
.binwindowskafka-topics.bat --list --zookeeper localhost:2181
```

By this simple Topic Creation method, you can confirm that Kafka is successfully installed on Windows and is working fine. Further, you can add and publish messages to the specific topic then consume all messages from

the same topic.

Conclusion

In this article, you have learned about Kafka and the distinct features of Kafka. You have also learned how to **Install Kafka on Windows**, create Topics in Kafka, and test whether your Kafka is working correctly. Since Kafka can perform more high-end operations, including Real-time Data Analytics, Stream Processing, building Data Pipelines, Activity Tracking, and more, it makes one of the go-to tools for working with streaming data.

Extracting complicated data from **Apache Kafka**, on the other hand, can be **difficult** and **time-consuming**. If you're having trouble with these issues and want to find a solution, **Hevo** is a good place to start!

[VISIT OUR WEBSITE TO EXPLORE HEVO](#)

Hevo Data is a **No-Code Data Pipeline** that offers a faster way to move data from **100+ Data Sources** including **Apache Kafka, Kafka Confluent Cloud**, and other **40+ Free Sources**, into your Data Warehouse to be visualized in a BI tool. You can use Hevo Pipelines to replicate the data from your Apache Kafka Source or Kafka Confluent Cloud to the Destination system. Hevo is fully automated and hence does not require you to code.

Getting Started with Hevo - An Overview



Want to take Hevo for a spin? [SIGN UP](#) for a **14-day Free Trial** and experience the feature-rich Hevo suite first hand. You can also have a look at the unbeatable **pricing** that will help you choose the right plan for your business needs.

Have you tried to **Install Kafka on Windows**? Share your experience with us in the comments section below!

No-code Data Pipeline For Apache Kafka

[TRY FOR FREE](#)

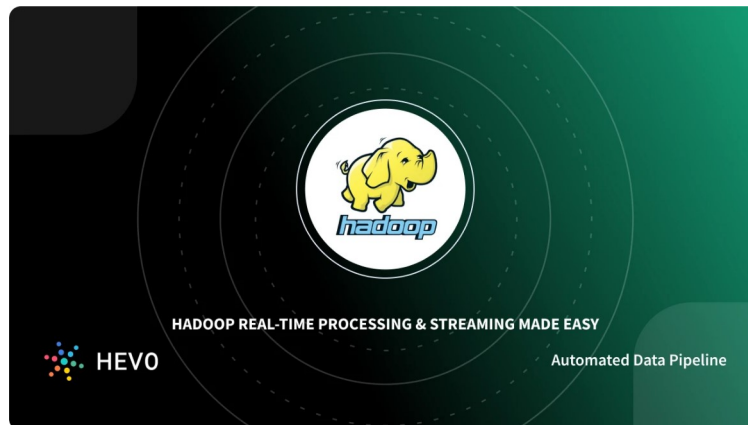
[Install Kafka on Windows](#)[Kafka](#)[Kafka on Windows](#)

Continue Reading



Divyansh Sharma on Apache Kafka, Data Integration, Data Streaming, Databricks

A Quick-Start Guide to Databricks Kafka Integration: 5 Comprehensive Aspects



Isola Saheed Ganiyu on Big Data, Data Processing, Data Streaming, Distributed System, Hadoop, Java, Python, Ubuntu

Hadoop Real-Time Processing & Streaming Simplified 101



Become a Contributor

You can contribute any number of in-depth posts on all things data.

[WRITE FOR HEVO](#)

Bring Real-Time Data from Any Source into your Warehouse

Your Work Email

GET STARTED FOR FREE

Talk to a Product Expert →

PLATFORM

Product
Integrations
Pricing
Free Trial
Changelog
Status

CONCEPTS

ETL
Amazon Redshift
Google BigQuery
Snowflake

COMPARISON GUIDES

ETL Tools
Data Pipeline Tools
Data Integration Tools
Redshift vs BigQuery
BigQuery vs Snowflake
Snowflake vs Redshift

TUTORIALS

Redshift ETL
BigQuery ETL
Snowflake ETL
Change Data Capture

WRITE FOR HEVO

You can contribute any number of in-depth posts on all things data.
[Know More](#)



HEVO



© Hevo Data Inc. 2022. All Rights Reserved.