

# How to Install Kafka on Windows Operating System?

## Step1: Prerequisites

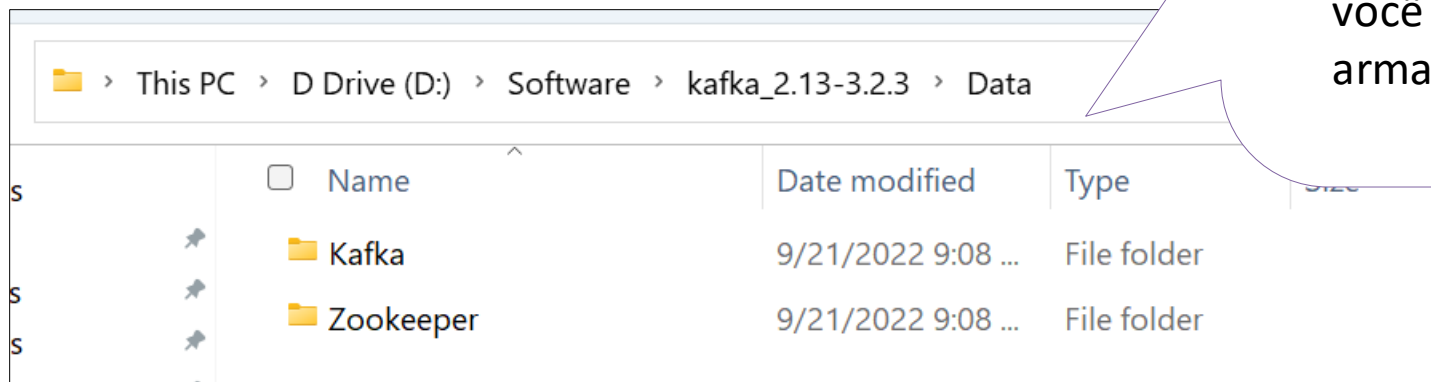
Espero que você já tenha instalado o java em sua máquina agora, então apenas verifique usando o seguinte comando.

```
C:\Users\Babu>java -version
java version "18.0.2.1" 2022-08-18
Java(TM) SE Runtime Environment (build 18.0.2.1+1-1)
Java HotSpot(TM) 64-Bit Server VM (build 18.0.2.1+1-1,
```

- Normalmente, os arquivos Kafka extraídos possuem arquivos Zookeeper que são executados simultaneamente com o Kafka para gerenciar todos os clusters e configurações dos servidores Kafka.

## Step 2: Install and Configure Kafka

- ✓ Em vez de armazená-los em pastas temporárias padrão, você pode configurar os arquivos Kafka e ZooKeeper para armazenar dados Kafka e ZooKeeper em pastas separadas.



Folder Path: This PC > D Drive (D:) > Software > kafka_2.13-3.2.3 > Data				
<input type="checkbox"/>	Name	Date modified	Type	Size
	Kafka	9/21/2022 9:08 ...	File folder	
	Zookeeper	9/21/2022 9:08 ...	File folder	

## Step 2: Install and Configure Kafka

```
zookeeper.properties x
13 # See the License for the specific language governing permissions and
14 # limitations under the License.
15 # the directory where the snapshot is stored.
16 dataDir=D:/Software/kafka_2.13-3.2.3/Data/Zookeeper
17 # the port at which the clients will connect
18 clientPort=2181
19 # disable the per-ip limit on the number of connections since this is a non-production config
20 maxClientCnxns=0
21 # Disable the adminserver by default to avoid port conflicts.
22 # Set the port to something non-conflicting if choosing to enable this
23 admin.enableServer=false
24 # admin.serverPort=8080
25
```

```
server.properties x
58
59 ##### Log Basics #####
60
61 # A comma separated list of directories under which to store log files
62 log.dirs=D:/Software/kafka_2.13-3.2.3/Data/Kafka
63
64 # The default number of log partitions per topic. More partitions allow greater
65 # parallelism for consumption, but this will also result in more files across
66 # the brokers.
67 num.partitions=1
68
```

### Step 3: Starting Zookeeper

```
D:\Software\kafka_2.13-3.2.3\bin\windows>zookeeper-server-start.bat D:/Software/kafka_2.13-3.2.3/config/zookeeper.properties
[2022-09-21 09:29:41,509] INFO Reading configuration from: D:/Software/kafka_2.13-3.2.3/config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-09-21 09:29:41,515] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-09-21 09:29:41,515] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-09-21 09:29:41,516] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2022-09-21 09:29:41,516] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
```

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. Você pode ver na saída que o Zookeeper foi iniciado e vinculado à porta 2181. Com isso, você pode confirmar que o Zookeeper Server foi iniciado com sucesso. Não feche o prompt de comando para manter o Zookeeper em execução opt to keep the Zookeeper running

### Step 4: Starting Kafka

```
D:\Software\kafka_2.13-3.2.3\bin\windows>kafka-server-start.bat D:/Software/kafka_2.13-3.2.3/config/server.properties
[2022-09-21 09:38:24,972] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
[2022-09-21 09:38:25,131] INFO Setting -D jdk.tls.rejectClientInitiatedRenegotiation=true to disable client-initiated TLS renegotiation
(org.apache.zookeeper.common.X509Util)
[2022-09-21 09:38:25,198] INFO starting (kafka.server.KafkaServer)
[2022-09-21 09:38:25,198] INFO Connecting to zookeeper on localhost:2181 (kafka.server.KafkaServer)
[2022-09-21 09:38:25,212] INFO [ZooKeeperClient Kafka server] Initializing a new session to localhost:2181. (kafka.zookeeper.ZooKeeperClient)
```

O servidor Kafka foi iniciado com sucesso e está pronto para streaming de dados.

Agora, o Zookeeper e o Kafka foram iniciados e executados com sucesso. Para confirmar isso, navegue até as pastas Kafka e Zookeeper recém-criadas. Ao abrir as respectivas pastas Zookeeper e Kafka, você pode notar que alguns novos arquivos foram criados dentro das pastas

### Step 5: Testing Kafka by Creating a Topic

```
D:\Software\kafka_2.13-3.2.3\bin\windows>kafka-topics.bat --create --topic AnimalTopic --bootstrap-server localhost:9092
Created topic AnimalTopic.
```

1. Como você iniciou o Kafka e o Zookeeper com êxito, pode testá-los criando novos tópicos e, em seguida, publicando e consumindo mensagens usando o nome do tópico.
2. Tópicos são os contêineres virtuais que armazenam e organizam um fluxo de mensagens em várias categorias chamadas de Partições.
3. Cada tópico Kafka é sempre identificado por um nome arbitrário e exclusivo em todo o cluster Kafka.
4. No comando acima, AnimalTopic é o nome exclusivo dado ao tópico. Após a execução do comando, um novo tópico é criado com sucesso.
5. Por este método simples de criação de tópico, você pode confirmar que o Kafka foi instalado com sucesso no Windows e está funcionando bem. Além disso, você pode adicionar e publicar mensagens no tópico específico e consumir todas as mensagens do mesmo tópico.

## How to Install Kafka on Windows Operating System?

### Producer

```
D:\Software\kafka_2.13-3.2.3\bin\windows>kafka-console-producer.bat --topic AnimalTopic --bootstrap-server localhost:9092
>Dog
>Cat
>Lion
>_
```

### Consumer

```
D:\Software\kafka_2.13-3.2.3\bin\windows>kafka-console-consumer.bat --topic AnimalTopic --from-beginning --bootstrap-server localhost:9092
Dog
Cat
Lion
```

To list all the available topics, you can execute the below command

```
D:\Software\kafka_2.13-3.2.3\bin\windows>kafka-topics.bat --list --bootstrap-server localhost:9092
AnimalTopic
OrderTopic
UserTopic
__consumer_offsets
```

### Create a Topic to Store your events and Write some events into the Topic

- ✓ Kafka é uma plataforma de streaming de eventos distribuídos que permite ler, gravar, armazenar e processar eventos (também chamados de registros ou mensagens na documentação) em várias máquinas.
- ✓ Exemplos de eventos são transações de pagamento, atualizações de geolocalização de telefones celulares, pedidos de remessa, medições de sensores de dispositivos IoT ou equipamentos médicos e muito mais.
- ✓ Esses eventos são organizados e armazenados em tópicos. Muito simplificado, um tópico é semelhante a uma pasta em um sistema de arquivos, e os eventos são os arquivos dessa pasta.
- ✓ Um cliente Kafka se comunica com os agentes Kafka por meio da rede para gravar (ou ler) eventos. Depois de recebidos, os corretores armazenarão os eventos de maneira durável e tolerante a falhas pelo tempo que você precisar - até para sempre.
- ✓ Os eventos são armazenados de forma durável no Kafka, eles podem ser lidos quantas vezes e por quantos consumidores você quiser. Você pode verificar isso facilmente abrindo outra sessão de terminal e executando novamente o comando anterior.