

# How To Setup Kafka on Docker?

Last Updated : 29 Jul, 2024

---

The Apache Kafka is an open source stream processing platform developed by the Apache Software Foundation written in Scala and Java Programming. And It is designed to handle real time data feeds with high throughput, low latency, and durability. The Kafka is used to build real time data pipelines and streaming applications that can process data stream at scale. Running Kafka on Docker allows us to quickly set up a scalable Kafka environment, isolating it from our local machines environment and ensuring consistency across different development environments.

## Table of Content

- [Terminologies](#)
- [Step by Step Process to Set up Kafka on Docker](#)
  - [Step 1: Install Docker](#)
  - [Step 2: Create a Docker Network](#)
    - [docker-compose.yml](#)
  - [Step 3: Start Zookeeper](#)
  - [Step 4: Start Kafka](#)
  - [Step 5: Verify Kafka Installation](#)
- [Example](#)
  - [Create a Topic](#)
  - [Send a Message](#)
  - [Consume the Message](#)
  - [Output](#)

- [Frequently Asked Questions](#)
  - [1. Why do we need Zookeeper for Kafka?](#)
  - [2. Can I run Kafka without Docker?](#)
  - [3. What are the default ports used by Kafka and Zookeeper?](#)
  - [4. How can I scale Kafka using Docker?](#)
  - [5. What is the purpose of the kafka-network in the setup process?](#)
- [Conclusion](#)

## Terminologies

Here we explain some Terminologies for understanding the Apache Kafka.

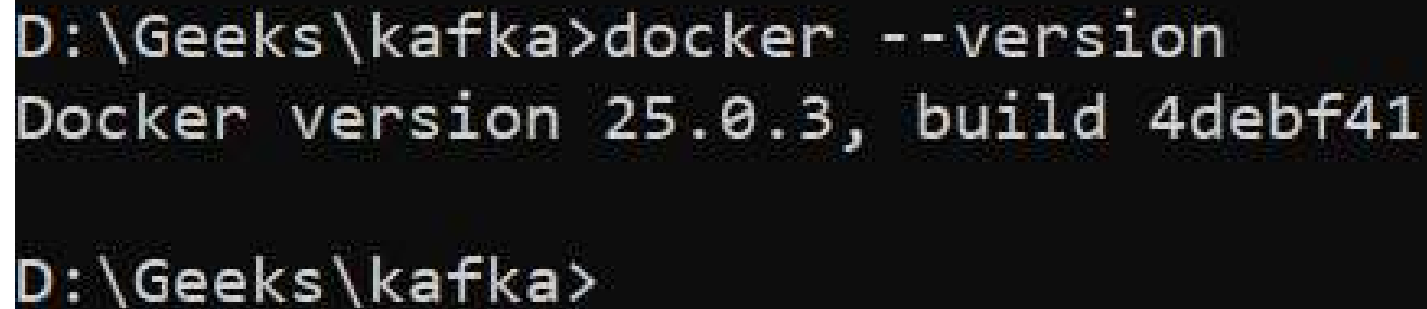
1. **Kafka Broker:** A Kafka server that stores and serves data.
2. **Topic:** A category to which records are published.
3. **Partition:** A division of topics data. The Kafka splits each topic into one or more partitions.
4. **Producer:** AN application that sends records to a Kafka topic.
5. **Consumer:** An application that reads records from a Kafka topic.
6. **Zookeeper:** A centralized service for maintain configuration information naming providing distributed synchronization and providing group services.

## Step by Step Process to Set up Kafka on Docker

Here we explain a step by step process to set up Kafka on Docker with examples and outputs for your reference.

### Step 1: Install Docker

First ensure the Docker is installed on your machine. Otherwise you can download Docker from official website and follow the installation instructions for your operating system.

A terminal window with a black background and white text. The prompt is 'D:\Geeks\kafka>'. The command entered is 'docker --version'. The output is 'Docker version 25.0.3, build 4deb41'. The prompt is then shown again as 'D:\Geeks\kafka>'.

```
D:\Geeks\kafka>docker --version
Docker version 25.0.3, build 4deb41
D:\Geeks\kafka>
```

## Step 2: Create a Docker Network

Create a Docker network to allow the Kafka and Zookeeper container to communicate with each other. Below we provide the docker command for this.

**docker-compose.yml**

```
version: '3'
services:
  zookeeper:
    image: confluentinc/cp-zookeeper:latest
    environment:
      ZOOKEEPER_CLIENT_PORT: 2181
      ZOOKEEPER_TICK_TIME: 2000
```


- "2181:2181"

```
... kafka:
  ... image: confluentinc/cp-kafka:latest
  ... depends_on:
  ...   - zookeeper
  ... ports:
  ...   - "9092:9092"
  ... environment:
  ...   KAFKA_BROKER_ID: 1
  ...   KAFKA_ZOOKEEPER_CONNECT: zookeeper:2181
  ...   KAFKA_ADVERTISED_LISTENERS: PLAINTEXT://localhost:9092
  ...   KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR: 1
```

Run below command

```
docker compose -f docker-compose.yml up
```

```
docker network create kafka-network
```

 C:\Windows\System32\cmd.exe



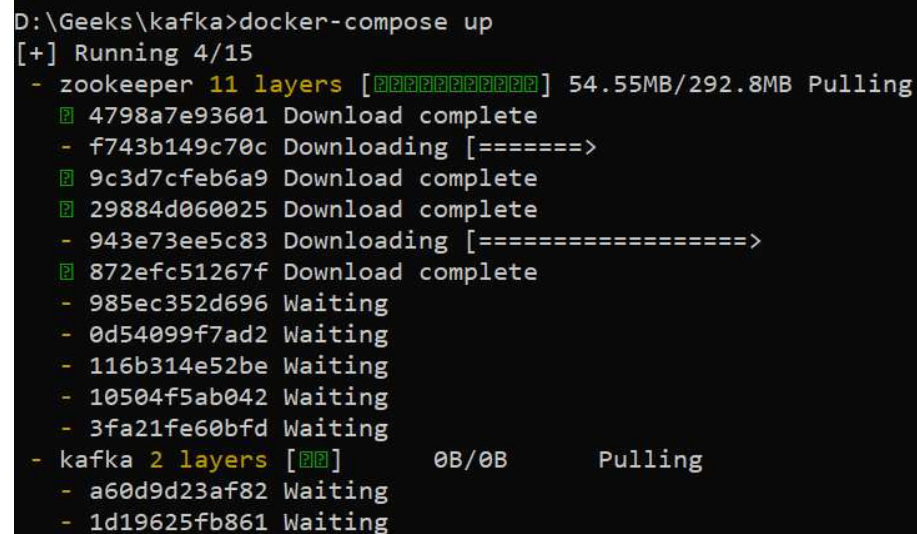
```
D:\Geeks>docker network create kafka-network
c3214a7158254d4d5f983bc945c6acc98a6956c174fd296ad3c80fbd4d9c7a81
```

*docker*

## Step 3: Start Zookeeper

Run a Zookeeper container using Docker and The Zookeeper is a prerequisite for running Kafka.

```
docker run -d --name zookeeper --network kafka-network -e ZOOKEEPER_CLIENT_PORT=2181  
confluentinc/cp-zookeeper:latest
```



```
D:\Geeks\kafka>docker-compose up  
[+] Running 4/15  
- zookeeper 11 layers [██████████████] 54.55MB/292.8MB Pulling  
  4798a7e93601 Download complete  
- f743b149c70c Downloading [=====>  
  9c3d7cfeb6a9 Download complete  
  29884d060025 Download complete  
- 943e73ee5c83 Downloading [=====>  
  872efc51267f Download complete  
- 985ec352d696 Waiting  
- 0d54099f7ad2 Waiting  
- 116b314e52be Waiting  
- 10504f5ab042 Waiting  
- 3fa21fe60bfd Waiting  
- kafka 2 layers [██] 0B/0B Pulling  
  a60d9d23af82 Waiting  
  1d19625fb861 Waiting
```

## Step 4: Start Kafka

Run a Kafka container and link it to the Zookeeper container.

```
docker run -d --name kafka --network kafka-network -e  
KAFKA_ZOOKEEPER_CONNECT=zookeeper:2181 -e
```

```
KAFKA_ADVERTISED_LISTENERS=PLAINTEXT://localhost:9092 -e KAFKA_BROKER_ID=1 -e  
KAFKA_OFFSETS_TOPIC_REPLICATION_FACTOR=1 confluentinc/cp-kafka:latest
```

```
kafka-1 | [2024-07-21 07:54:15,236] DEBUG [PartitionStateMachine contro  
.controller.ZkPartitionStateMachine)  
kafka-1 | [2024-07-21 07:54:15,236] INFO [Controller id=1] Ready to ser  
kafka-1 | [2024-07-21 07:54:15,239] INFO [SocketServer listenerType=ZK_  
kafka-1 | [2024-07-21 07:54:15,240] WARN [Controller id=1, targetBroker  
Broker may not be available. (org.apache.kafka.clients.NetworkClient)  
kafka-1 | [2024-07-21 07:54:15,243] WARN [RequestSendThread controllerI  
nsuccessful (kafka.controller.RequestSendThread)  
kafka-1 | java.io.IOException: Connection to localhost:9092 (id: 1 rack  
kafka-1 | at org.apache.kafka.clients.NetworkClientUtils.awaitReady(Ne  
kafka-1 | at kafka.controller.RequestSendThread.brokerReady(Controller  
kafka-1 | at kafka.controller.RequestSendThread.doWork(ControllerChann  
kafka-1 | at org.apache.kafka.server.util.ShutdownableThread.run(Shutd  
kafka-1 | [2024-07-21 07:54:15,247] INFO [Controller id=1, targetBroker
```

## Step 5: Verify Kafka Installation

To ensure Kafka is running correctly and list all topics.

```
docker exec -it kafka kafka-topics --list --zookeeper zookeeper:2181
```

<input type="checkbox"/>	▼	 <a href="#">kafka</a>	Runnir 0.78%	2 minutes ago
<input type="checkbox"/>		 <a href="#">kafka</a> 58374de8a026 	<a href="#">wurstmeister</a> Runnir 0.67% <a href="#">9092:9092</a> 	2 minutes ago
<input type="checkbox"/>		 <a href="#">zookeeper</a> 697ded94abe6 	<a href="#">wurstmeister</a> Runnir 0.11% <a href="#">2181:2181</a> 	2 minutes ago

## Example

Here we provide an example for your understanding purpose. Now creating a topic and sending a message.

### Create a Topic

```
docker exec -it kafka kafka-topics --create --topic test-topic --partitions 1 --replication-factor 1 --if-not-exists --zookeeper zookeeper:2181
```

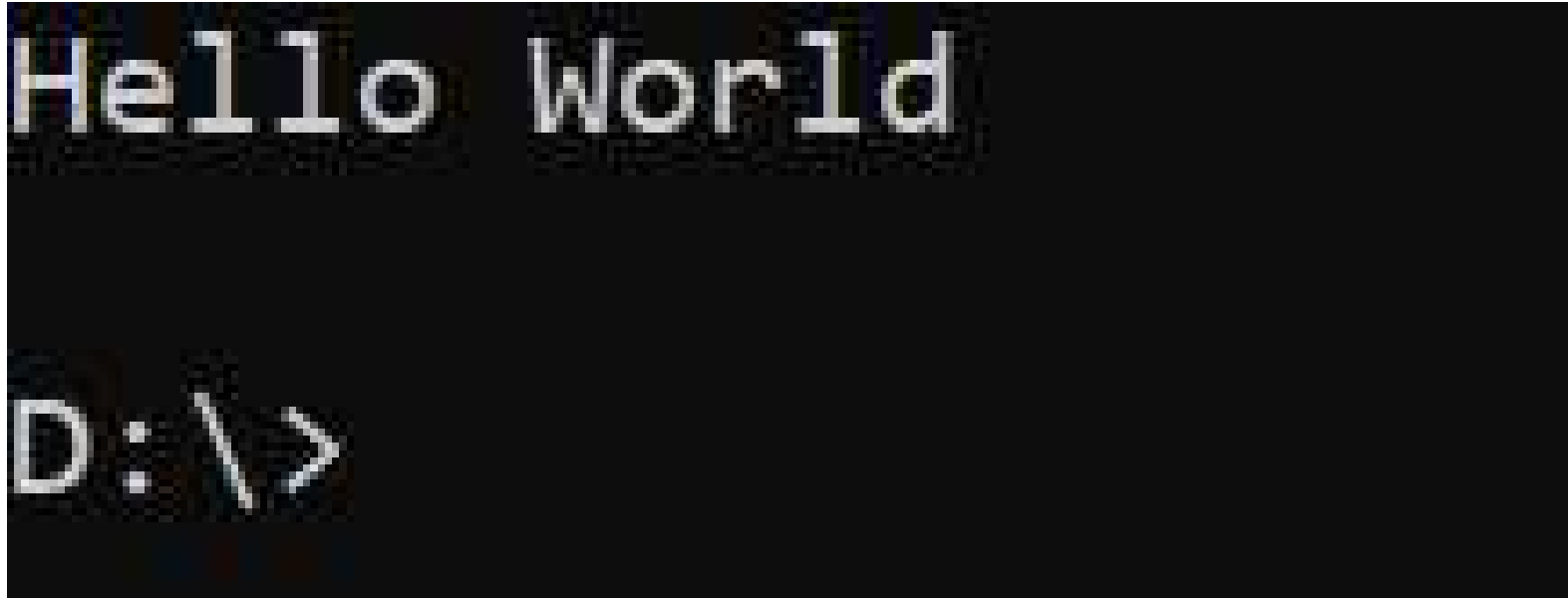
### Send a Message

```
docker exec -it kafka kafka-console-producer --broker-list localhost:9092 --topic test-topic
```

### Consume the Message

```
docker exec -it kafka kafka-console-consumer --bootstrap-server localhost:9092 --topic  
test-topic --from-beginning
```

## Output

A screenshot of a terminal window with a black background and white text. The text "Hello World" is displayed on the first line. On the second line, a command prompt "D:\>" is visible, indicating the user is at the root of the D drive in Windows.

## Frequently Asked Questions

### 1. Why do we need Zookeeper for Kafka?

*Answer: Zookeeper manages and coordinates the Kafka brokers. It helps in leader election, configuration management and synchronizing the state between brokers.*



## 2. Can I run Kafka without Docker?

***Answer:** Yes, you can run Kafka directly on your machine by downloading the Kafka binaries and setting it up manually. Docker simplifies the setup process and ensures consistency across environments.*

## 3. What are the default ports used by Kafka and Zookeeper?

***Answer:** Kafka uses port 9092 by default while Zookeeper uses port 2181.*

## 4. How can I scale Kafka using Docker?

***Answer:** You can scale Kafka by running multiple Kafka broker container. Each broker should have a unique broker ID and advertised listener.*

## 5. What is the purpose of the kafka-network in the setup process?

***Answer:** The kafka-network Docker network allows the Kafka and Zookeeper containers to communicate with each other ensuring they are on the same virtual network.*

## Conclusion

Setting up Kafka on Docker is a streamlined and efficient way to build a scalable and consistent Kafka environment by leveraging Docker's containerization capabilities and you can isolate your Kafka setup from the local machine. Ensuring consistency across different environments and simplifying the management of dependencies and configuration. Implementing Kafka on Docker not