

# How to Produce Messages to Kafka Topics using Apache Kafka CLI

Apache Kafka is a distributed streaming platform that allows you to publish and subscribe to streams of data. In this tutorial, you will learn how to use the Apache Kafka command-line interface (CLI) to produce messages to Kafka topics.

A Kafka topic is a logical name for a stream of data. You can create multiple topics in Kafka, and each topic can have multiple partitions. Partitions are the units of parallelism in Kafka, and they allow you to scale up your data processing.

## How to produce messages to Kafka topics without a key

To produce messages to Kafka topics without a key, you will use the `kafka-console-producer.sh` script that comes with the Kafka installation. This script allows you to send messages to a Kafka topic from the standard input (stdin).

Before you run the script, you need to make sure that you have a running Kafka cluster.

In this tutorial, I will assume that you have a Kafka cluster running on `localhost:9092`. You can change this to match your cluster configuration.

To produce messages to a Kafka topic called `product-created-events-topic`, you can run the following command in a terminal:

```
./kafka-console-producer.sh --bootstrap-server localhost:9092
--topic product-created-events-topic
```

The `--bootstrap-server` option specifies the address of the Kafka broker that the producer will connect to. The `--topic` option specifies the name of the topic that the producer will send messages to.

After you run the command, you will see a `>` prompt, indicating that the producer is ready to accept messages from stdin. You can type any message you want, and press enter to send it to the topic. For example:

```
> Hello, world!
> This is a test message.
> I am learning how to use Kafka CLI.
```

Each line that you type will be sent as a separate message to the topic. You can verify that the messages are received by the topic by using another terminal to run the `kafka-console-consumer.sh` script, which allows you to consume messages from a Kafka topic. For example:

```
./kafka-console-consumer.sh --bootstrap-server localhost:9092
--topic product-created-events-topic --from-beginning
```

The `--from-beginning` option tells the consumer to read all the messages from the topic, starting from the earliest one. You should see the messages that you typed in the producer terminal, in the same order. For example:

```
Hello, world!
This is a test message.
I am learning how to use Kafka CLI.
```

To stop the producer or the consumer, you can press `Ctrl+C` in the terminal.

## How to produce messages to Kafka topics with a key

Messages in Kafka topics are stored as a key-value pair. If you send a message without a key, then the key is null. However, sometimes you may want to send a message with a key, for example, to ensure that messages with the same key are stored in the same partition, and thus preserve the order of the messages.

To produce messages to Kafka topics with a key, you will use the same `kafka-console-producer.sh` script, but with some additional properties. The first property is `parse.key=true`, which tells the producer to expect each message to be in the format of `key:value`. The second property is `key.separator=:` (or any other character you prefer), which tells the producer what character to use to separate the key and the value in each message.

To produce messages with a key to the same topic `product-created-events-topic`, you can run the following command in a terminal:

```
./kafka-console-producer.sh --bootstrap-server localhost:9092
--topic product-created-events-topic --property "parse.key=true"
--property "key.separator=:"
```

After you run the command, you will see a > prompt, indicating that the producer is ready to accept messages from stdin. You can type any message you want, as long as it follows the key:value format, and press enter to send it to the topic. For example:

```
> product_id:123
> product_name:Kafka mug
> product_price:9.99
```

Each line that you type will be sent as a separate message to the topic, with the key and the value separated by the : character. You can verify that the messages are received by the topic by using another terminal to run the kafka-console-consumer.sh script, with the same properties as the producer. For example:

```
./kafka-console-consumer.sh --bootstrap-server localhost:9092
--topic product-created-events-topic --from-beginning --property
"print.key=true" --property "key.separator=:"
```

The print.key=true property tells the consumer to print both the key and the value of each message, separated by the key.separator character. You should see the messages that you typed in the producer terminal, in the same order. For example:

```
product_id:123
product_name:Kafka mug
product_price:9.99
```

To stop the producer or the consumer, you can press Ctrl+C in the terminal.