

Home	Kafka	Scala	Java	C++	C#	SQL	Android	HTML	CSS	XML	JavaScript
------	-------	-------	------	-----	----	-----	---------	------	-----	-----	------------

Sending data to Kafka Topics

Kafka Console Producer

In order to send data to the Kafka topic, a producer is required. The role of the producer is to send or write data/messages to the Kafka topics.

In this section, we will learn how a producer sends messages to the Kafka topics.

There are following steps used to launch a producer:

Step1: Start the zookeeper as well as the kafka server.

C PROGRAM TO IMPLEMENT STACK

```

#define max 3
void push();
void pop();
void pri();
int st[max];
int top=-1;
void main()
{
    int ch;
    clrscr();
    while(1)
    {
        printf("\n 1. Push:");
        printf("\n 2. Pop:");
        printf("\n 3. Pri:");
        printf("\n 4. Exit:");
      
```

```

printf("\n Enter the choice:");
scanf("%d",&ch);
switch(ch)
{
    case 1:push();
    break;
    case 2: pop();
    break;
    case 3: pri();
    break;
    case 4: exit();
    default:
    printf("\n wrong choice :");
  
```

java point

Step2: Type the command: '**kafka-console-producer**' on the command line. This will help the user to read the data from the standard inputs and write it to the Kafka topic.



Note: Choose '.bat' or '.sh' as per the operating system.

```

Select C:\Windows\system32\cmd.exe

C:\>kafka-console-producer.bat
This tool helps to read data from standard input and publish it to Kafka.
Option                                     Description
-----
--batch-size <Integer: size>             Number of messages to send in a single
                                          batch if they are not being sent
                                          synchronously. (default: 200)
--broker-list <String: broker-list>      REQUIRED: The broker list string in
                                          the form HOST1:PORT1,HOST2:PORT2.
                                          The compression codec: either 'none',
                                          'gzip', 'snappy', 'lz4', or 'zstd'.
                                          If specified without value, then it
                                          defaults to 'gzip'
--compression-codec [String:
compression-codec]                       Print usage information.
--help
--line-reader <String: reader_class>      The class name of the class to use for
                                          reading lines from standard in. By
                                          default each line is read as a
                                          separate message. (default: kafka.
  
```

```

--socket-buffer-size <Integer: size>     the producer waits before refreshing
                                          the metadata. (default: 100)
                                          The size of the tcp RECV size.
                                          (default: 102400)
--sync                                     If set message send requests to the
                                          brokers are synchronously, one at a
                                          time as they arrive.
--timeout <Integer: timeout_ms>          If set and the producer is running in
                                          asynchronous mode, this gives the
                                          maximum amount of time a message
                                          will queue awaiting sufficient batch
                                          size. The value is given in ms.
                                          (default: 1000)
--topic <String: topic>                 REQUIRED: The topic id to produce
                                          messages to.
--version                                 Display Kafka version.

C:\>_
  
```

The highlighted text represents that a '**broker-list**' and a '**topic id**' is required to produce a message. It is because a producer must know the id of the topic to which the data is to be written.

Step3: After knowing all the requirements, try to produce a message to a topic using the command:

'**kafka-console-producer -broker-list localhost:9092 -topic <topic_name>**'. Press enter.

```

C:\Windows\system32\cmd.exe  kafka-console-producer.bat  broker list 127.0.0.1:9092  topic myfirst

C:\Users>kafka-console-producer.bat --broker-list 127.0.0.1:9092 --topic myfirst
>
  
```



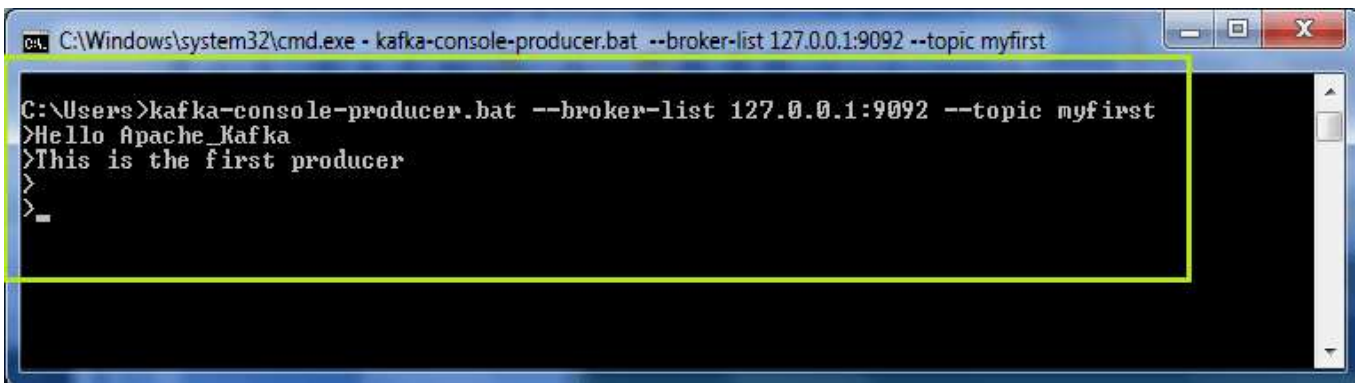
Note: Here, 9092 is the port number of the Kafka server.

Here, 'myfirst' topic is chosen to write messages to.

A '>' will appear in the new line. Start producing some messages, as shown below:

ADVERTISEMENT

ADVERTISEMENT



```
C:\Windows\system32\cmd.exe - kafka-console-producer.bat --broker-list 127.0.0.1:9092 --topic myfirst

C:\Users>kafka-console-producer.bat --broker-list 127.0.0.1:9092 --topic myfirst
>Hello Apache_Kafka
>This is the first producer
>
>
```

Step4: Press 'Ctrl+c' and exist by pressing the 'Y' key.

So, in this way, a producer can produce/send several messages to the Kafka topics.

Producer with Keys

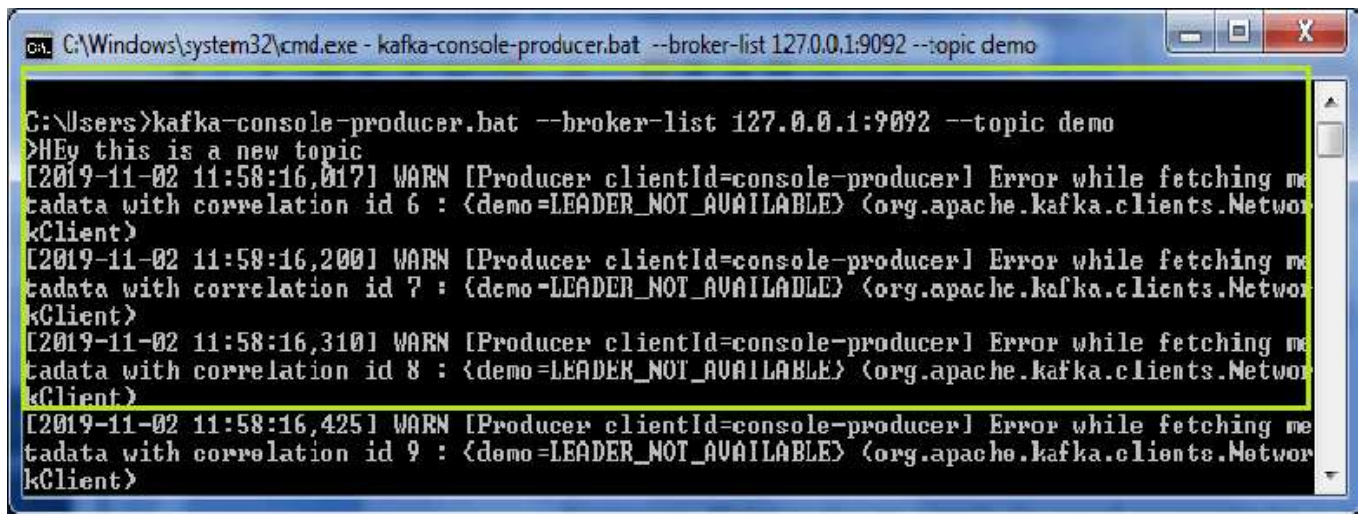
A Kafka producer can write data to the topic either with or without a key. If a producer does not specify a key, the data will be stored to any of the partitions with key=null, else the data will be stored to the specified partition only. A '**parse.key**' and a '**key.seperator**' is required to specify a key for the topic. The command used is:

```
'kafka-console-producer      --broker-list    localhost:9092      --topic    <topic_name>    --
property parse.key=true --property key.seperator=,
> key,value
> another key,another value'
```

Here, key is the specific partition, and value is the message to be written by the producer to the topic.

When a topic does not exist?

Suppose the producer wants to send messages to a new topic that does not exist yet. In such a situation, a warning will appear, as shown in the below snapshot, after producing a message. It is just a warning.



```
C:\Windows\system32\cmd.exe - kafka-console-producer.bat --broker-list 127.0.0.1:9092 --topic demo

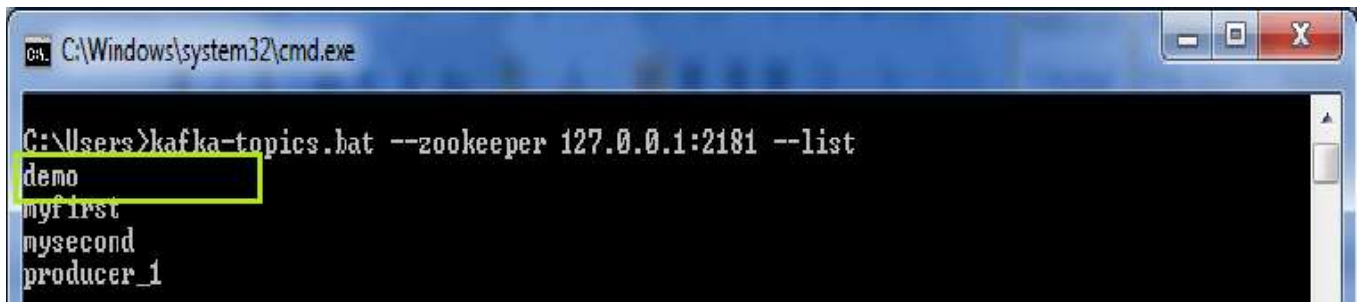
C:\Users>kafka-console-producer.bat --broker-list 127.0.0.1:9092 --topic demo
>HEY this is a new topic
[2019-11-02 11:58:16,017] WARN [Producer clientId=console-producer] Error while fetching me
tadata with correlation id 6 : {demo=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.Networ
kClient)
[2019-11-02 11:58:16,200] WARN [Producer clientId=console-producer] Error while fetching me
tadata with correlation id 7 : {demo=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.Networ
kClient)
[2019-11-02 11:58:16,310] WARN [Producer clientId=console-producer] Error while fetching me
tadata with correlation id 8 : {demo=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.Networ
kClient)
[2019-11-02 11:58:16,425] WARN [Producer clientId=console-producer] Error while fetching me
tadata with correlation id 9 : {demo=LEADER_NOT_AVAILABLE} (org.apache.kafka.clients.Networ
kClient)
```

ADVERTISEMENT

Why this warning?

The warning occurred because earlier the topic 'demo' didn't exist. But, as soon the producer wrote a message, Kafka somehow created that topic. Although, no leader election held for this unexpected topic, '**LEADER_NOT_AVAILABLE**' error could be seen. But, for the next time, the producer can continue to write more messages as no warning will appear again. It is because the topic comes in the existing list now.

The users can check using the '**-list**' command, as shown below:



```
C:\Windows\system32\cmd.exe

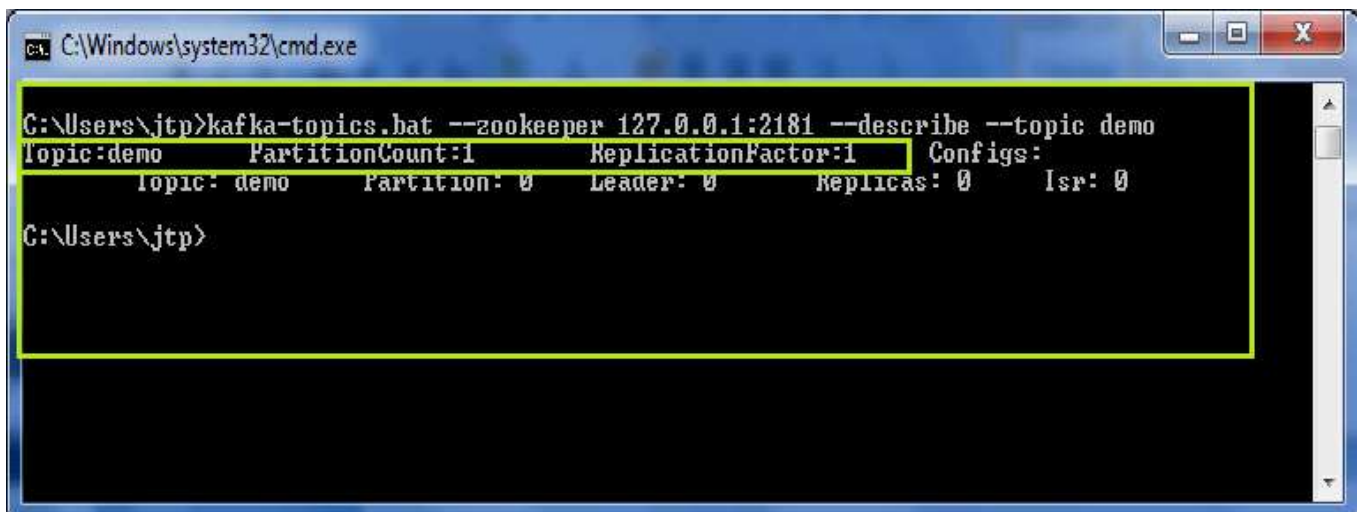
C:\Users>kafka-topics.bat --zookeeper 127.0.0.1:2181 --list
demo
myfirst
mysecond
producer_1
```

The topic 'demo' can be seen on the list.

Describing the new topic

As such topics which are created directly by the producer grabs the default number of partitions and its replication factor as 1.

For example,



```
C:\Windows\system32\cmd.exe

C:\Users\jtp>kafka-topics.bat --zookeeper 127.0.0.1:2181 --describe --topic demo
Topic:demo      PartitionCount:1      ReplicationFactor:1      Configs:
topic: demo      Partition: 0      Leader: 0      Replicas: 0      Isr: 0

C:\Users\jtp>
```

The topic 'demo' when described using the '**-describe**' command, gives the value of 'PartitionCount' and 'ReplicationFactor' as 1(default value). Thus, it is always a better option to create a topic before producing messages to it.

ADVERTISEMENT

Changing the Default Values

Follow the below steps to change the default values for the new topic:

1. Open '**server.properties**' file using Notepad++, or any other text editor.
2. Edit the value of num.partitions=1 to a new value. Let it be 3. So, whenever such new topics are introduced, the number of PartitionCount and ReplicationFactor will be 3(whatever the user has set).
3. Save the file.

But, always create topics before.

← Prev

Next →

 [For Videos Join Our Youtube Channel: Join Now](#)


Feedback

- Send your Feedback to feedback@javatpoint.com

Help Others, Please Share





Learn Latest Tutorials


 [Splunk tutorial](#)
Splunk


 [SPSS tutorial](#)
SPSS

 [Swagger tutorial](#)
Swagger

 [T-SQL tutorial](#)
Transact-SQL

 [Tumblr tutorial](#)
Tumblr


 [React tutorial](#)
ReactJS

 [Regex tutorial](#)
Regex

 [Reinforcement learning tutorial](#)
Reinforcement Learning

 [R Programming tutorial](#)
R Programming

 [RxJS tutorial](#)
RxJS

 [React Native tutorial](#)
React Native

 [Python Design Patterns](#)
Python Design Patterns

 [Python Pillow tutorial](#)

 [Python Turtle tutorial](#)

 [Keras tutorial](#)
Keras

[Python Pillow](#)[Python Turtle](#)

Preparation

 [Aptitude](#)[Aptitude](#)[Logical Reasoning](#)[Reasoning](#)[Verbal Ability](#)[Verbal Ability](#)[Interview Questions](#)[Interview Questions](#)[Company Interview Questions](#)[Company Questions](#)

Trending Technologies

[Artificial Intelligence](#)[Artificial Intelligence](#)[AWS Tutorial](#)[AWS](#)[Selenium tutorial](#)[Selenium](#)[Cloud Computing](#)[Cloud Computing](#)[Hadoop tutorial](#)[Hadoop](#)[ReactJS Tutorial](#)[ReactJS](#)[Data Science Tutorial](#)[Data Science](#)[Angular 7 Tutorial](#)[Angular 7](#)[Blockchain Tutorial](#)[Blockchain](#)[Git Tutorial](#)[Git](#)[Machine Learning Tutorial](#)[Machine Learning](#)[DevOps Tutorial](#)[DevOps](#)

ADVERTISEMENT

B.Tech / MCA

 DBMS tutorial DBMS	 Data Structures tutorial Data Structures	 DAA tutorial DAA	 Operating System Operating System
 Computer Network tutorial Computer Network	 Compiler Design tutorial Compiler Design	 Computer Organization and Architecture Computer Organization	 Discrete Mathematics Tutorial Discrete Mathematics
 Ethical Hacking Ethical Hacking	 Computer Graphics Tutorial Computer Graphics	 Software Engineering Software Engineering	 html tutorial Web Technology
 Cyber Security tutorial Cyber Security	 Automata Tutorial Automata	 C Language tutorial C Programming	 C++ tutorial C++
 Java tutorial Java	 .Net Framework tutorial .Net	 Python tutorial Python	 List of Programs Programs
 Control Systems tutorial Control System	 Data Mining Tutorial Data Mining	 Data Warehouse Tutorial Data Warehouse	

