Mastering Kafka Topics: Producing and Consuming Messages with Confluent CLI

A Step-by-Step Guide to Creating Kafka Topics, Producing Messages, and Consuming Data Using Confluent CLI

3 min read · Draft

Durga Gadiraju Follow

Introduction

Apache Kafka is the backbone of modern event streaming platforms, enabling real-time data processing for various applications. In this article, we'll dive into the practical aspects of working with Kafka topics — creating topics, producing messages, and consuming them in real time using the Confluent CLI. Whether you're exploring Kafka for the first time or looking to solidify your understanding, this hands-on guide will walk you through every step.

Creating Kafka Topics

Kafka topics are the core building blocks for managing data streams. To test Kafka efficiently, we'll start by creating a topic.

Command to Create a Topic

```
confluent kafka topic create <topic-name>
```

Example Command:

```
confluent kafka topic create my-topic
```

Output:

```
Created topic "my-topic".
```

This creates a topic named `my-topic`, which will store the messages you produce and allow you to consume them later.

Producing Messages to Kafka Topics

A producer sends data to a Kafka topic. Confluent CLI provides a built-in console producer for testing purposes.

Command to Start the Producer

```
confluent kafka topic produce <topic-name>
```

Example Command:

```
confluent kafka topic produce my-topic
```

Once the producer starts, you'll see:

```
Starting Kafka producer. Use Ctrl-C or Ctrl-D to exit.
```

Now, type messages directly into the terminal to send them to Kafka:

```
Hello Kafka
12345
```

Each line is treated as a message and sent to the Kafka topic `my-topic`.

To stop the producer, press `Ctrl+C`.

Consuming Messages from Kafka Topics

A consumer retrieves data from a Kafka topic. By default, consumers process data in real time, but you can also configure them to read from the beginning of the topic.

Command to Start the Consumer

```
confluent kafka topic consume <topic-name> --from-beginning
```

Example Command:

```
confluent kafka topic consume my-topic --from-beginning
```

Output:

```
Starting Kafka consumer. Use Ctrl-C to exit.
Hello Kafka
12345
```

- The `--from-beginning` flag ensures all previously sent messages are consumed.
- The consumer will continue to listen for new messages in real time.

Testing Real-Time Data Streaming

To test Kafka's real-time capabilities:

1.Open two terminal tabs:

- In one tab, start the consumer:

```
confluent kafka topic consume my-topic --from-beginning
```

- In the other tab, start the producer:

```
confluent kafka topic produce my-topic
```

2.Send messages from the producer tab:

```
Real-time message 1
Real-time message 2
```

3.Check the consumer tab. Messages should appear instantly:

```
Real-time message 1
Real-time message 2
```

Viewing Topics and Messages in Confluent Cloud

If you're using Confluent Cloud, you can also view the topic and messages via the web interface:

1.Log in to Confluent Cloud:
Navigate to your environment in the Confluent Cloud dashboard.

2.Access the Kafka Cluster:

- Go to **Environments** and select your environment.

- Choose your Kafka cluster (e.g., “Demo Kafka Cluster”).

3.View Topics:

- Go to the **Topics** section.
- Select your topic (e.g., `my-topic`).

4.View Messages:

- Navigate to the **Messages** tab.
- Select “From Beginning” to see all messages or “Latest” for new messages.

Deleting Environments and Clusters

To clean up resources after testing, delete the environment and its associated cluster.

Command to Delete an Environment

```
confluent environment delete <environment-name>
```

Steps:

1.List all environments:

```
confluent environment list
```

2.Delete the desired environment:

```
confluent environment delete KafkaCloudEnvironment
```

Confirm the deletion by typing the environment name.

Tips for Success

- **Use Descriptive Names:** Name your topics and clusters meaningfully to keep track of them easily.
- **Monitor Kafka:** Leverage the Confluent Cloud dashboard for a visual overview of your clusters and topics.
- **Practice Real-Time Scenarios:** Simulate real-world use cases, such as IoT sensors or log aggregators, to gain practical experience.
- **Secure Your Resources:** Clean up environments and clusters when they're no longer needed to avoid unnecessary charges.

Next Steps

In the next article, we'll explore more advanced Kafka operations, including configuring partitions, setting retention policies, and integrating Kafka with external systems using Kafka Connect. Stay tuned for actionable insights and hands-on examples!

Conclusion

Kafka is a powerful tool for managing data streams in real time. By using Confluent CLI, you can easily create topics, produce messages, and consume data for testing and development. Whether you're working locally or in the cloud, mastering these basics sets the foundation for advanced Kafka use cases.

Stay Connected

💡 ***Follow Durga Gadiraju*** *for more hands-on Kafka tutorials and data engineering insights!*
🔄 ***Share this article*** *with colleagues interested in real-time data streaming.*
💬 ***Comments and Questions:*** *Let us know your thoughts below — your feedback drives our learning journey!*

Confluent Kafka | Confluent Cloud | Kafka | Data Engineering | Event Streaming

Follow

Written by Durga Gadiraju

1.4K followers · 181 following

Founder of ITVersity and Technology Evangelist

Recommended from Medium

In Long. Sweet. Valuable. by Ossai Chinedum

I'll Instantly Know You Used Chat Gpt If I See This

Trust me you're not as slick as you think

May 16 8.5K 467

Prateek Jain

7 Open Source Diagram-as-Code Tools You Should Try

A hands-on guide to 7 open-source tools that let you draw cloud or application architecture from code

6d ago 1.1K 21

In JavaScript in Plain English by GeekSociety

I Stopped Building Frontends. Now I Use MCP Servers to Let AI Run My Apps

It's 2025, and the way we build applications has fundamentally changed.

Jun 2 4.1K 158

In Realworld AI Use Cases by Chris Dunlop

Why clients pay me 10x more than developers who are better at coding than me

Last week I charged $15,000 for work a better coder would do for $1,500 and I think you should learn these skills now that we have AI

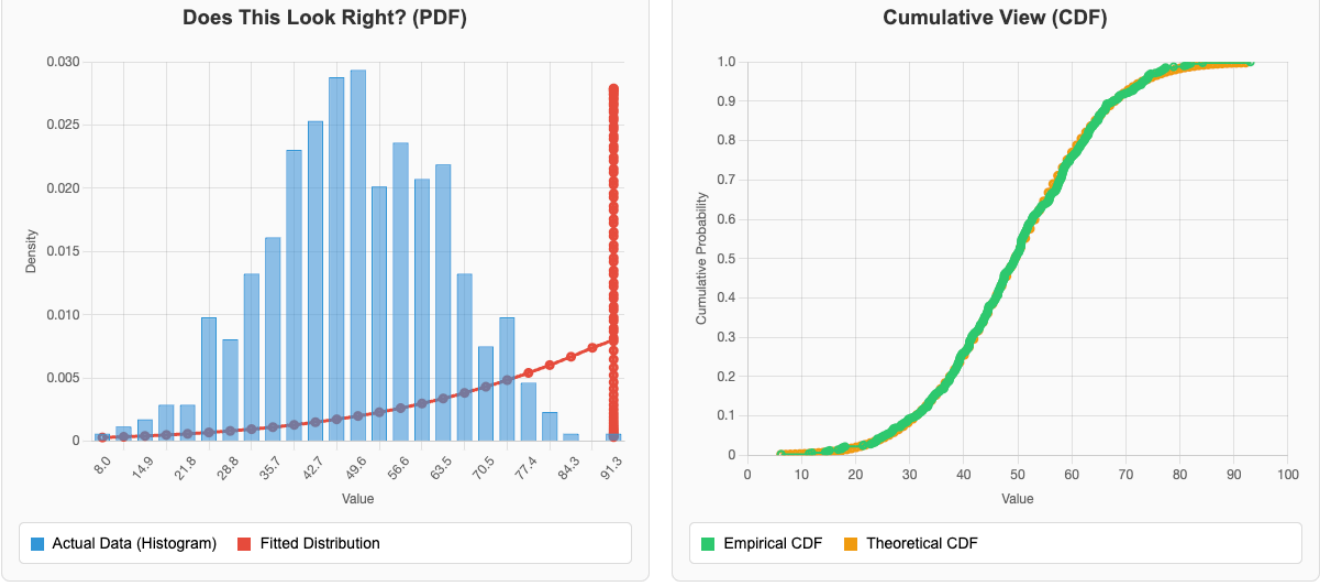
Jun 2 3.9K 60

In AI Advances by Kris Ograbek

If I started learning AI Agents & no-code Automation in 2025, here's what I'd do to move 10x faster

The ultimate, no-fluff learning guide for non-tech beginners.

6d ago 998 23



In Data Science Collective by Paolo Perrone

How to Find the Right Distribution for Your Data: A Practical Guide for Non-Statistician

With Two Interactive Tools to Get it Right

Jun 5 637 8

See more recommendations