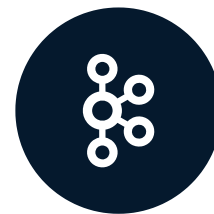


Introduction to Kafka

INTRODUCTION TO KAFKA



Mike Metzger
Data Engineering Consultant

About me

- 25+ year IT career, 18+ in data
- Tools:
 - Apache Spark
 - Apache Airflow
 - Apache Kafka



What is Apache Kafka?

- Open-source
- Distributed
- Event streaming
- Designed to handle large quantities of data
- Scalable



Event streaming in Kafka

- Obtain source information
- Stored reliably
- Distribute data to clients

Common uses

- Ecommerce
- Order tracking
- Ride-share / Food delivery
- Sensor data
- Cybersecurity



¹ Photo by Carlos Muza on Unsplash

Kafka components

- User facing components



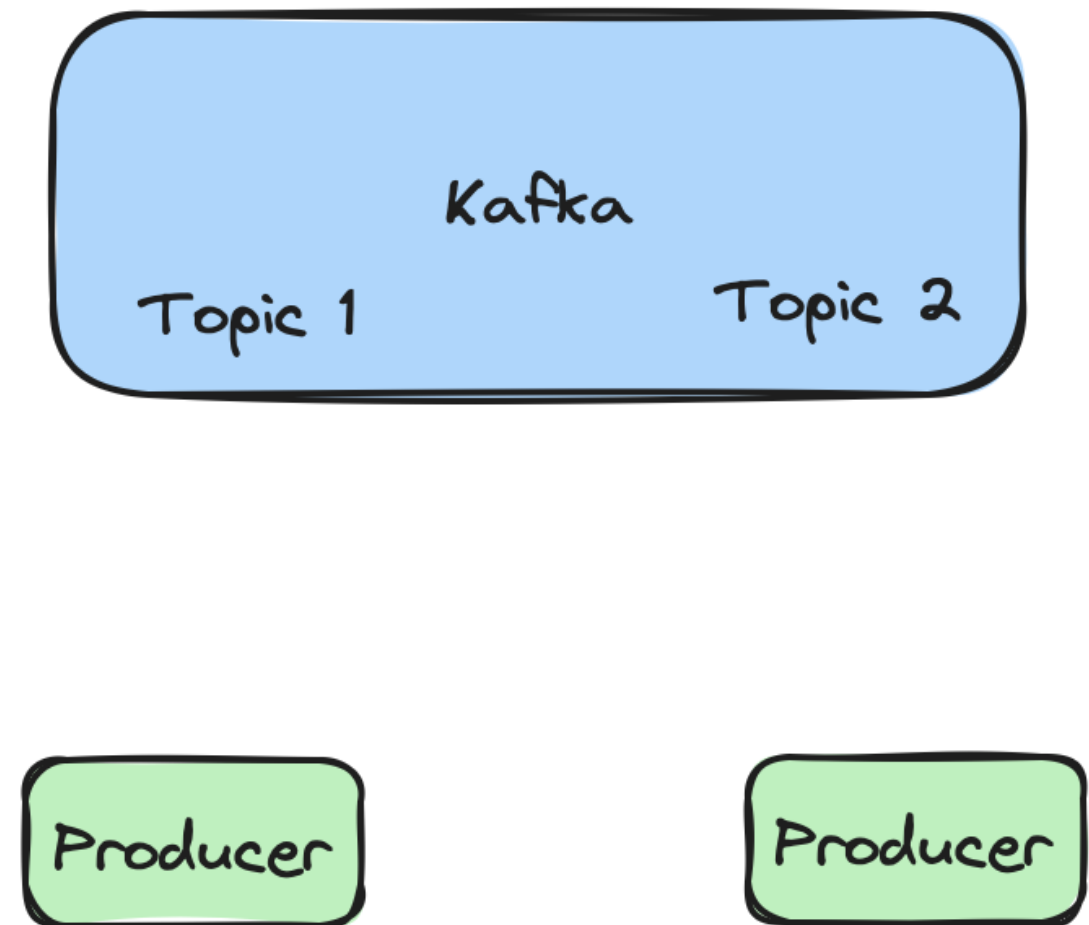
Kafka components

- User facing components
 - Topics



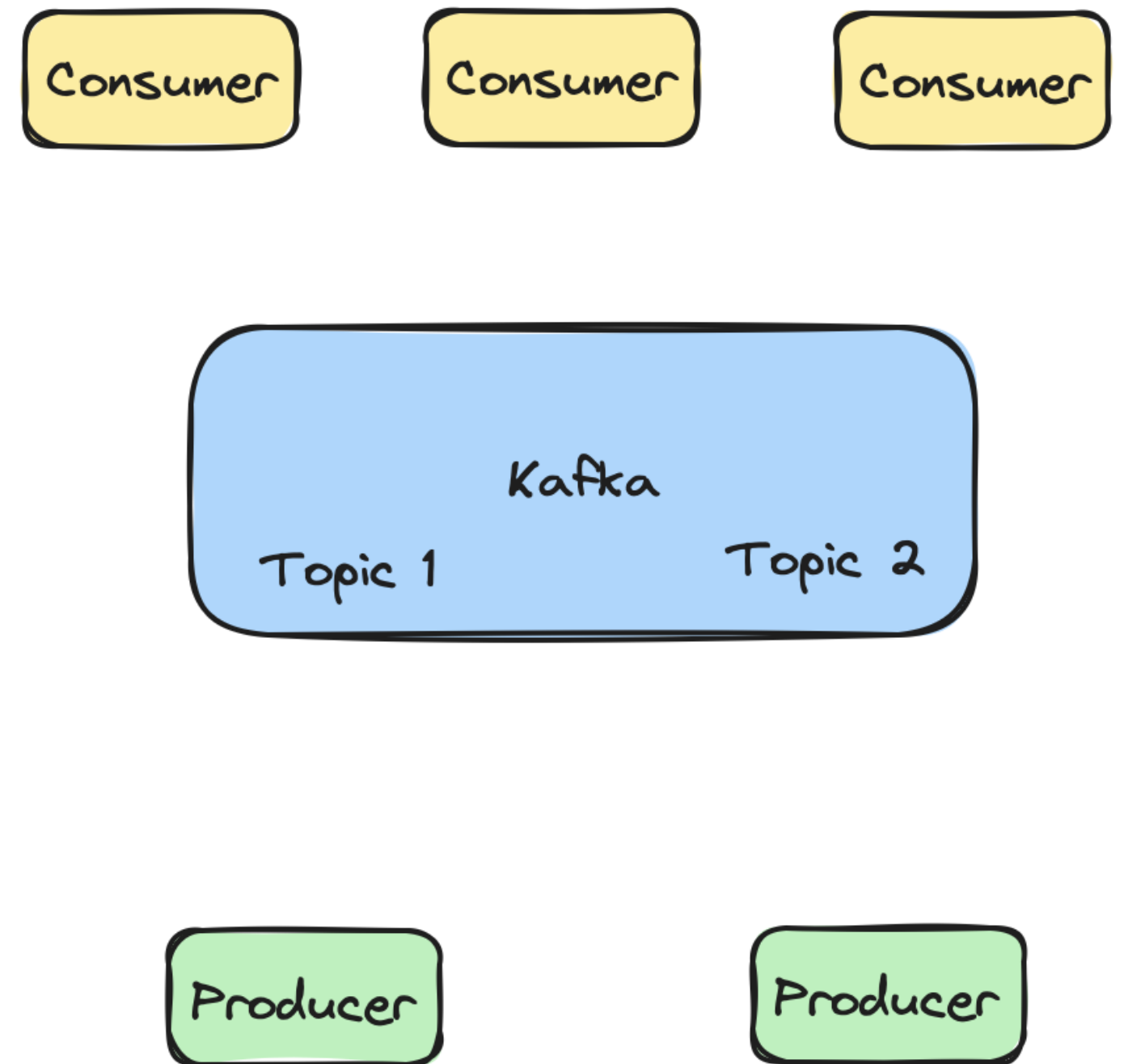
Kafka components

- User facing components
 - Topics
 - Producers (writers)



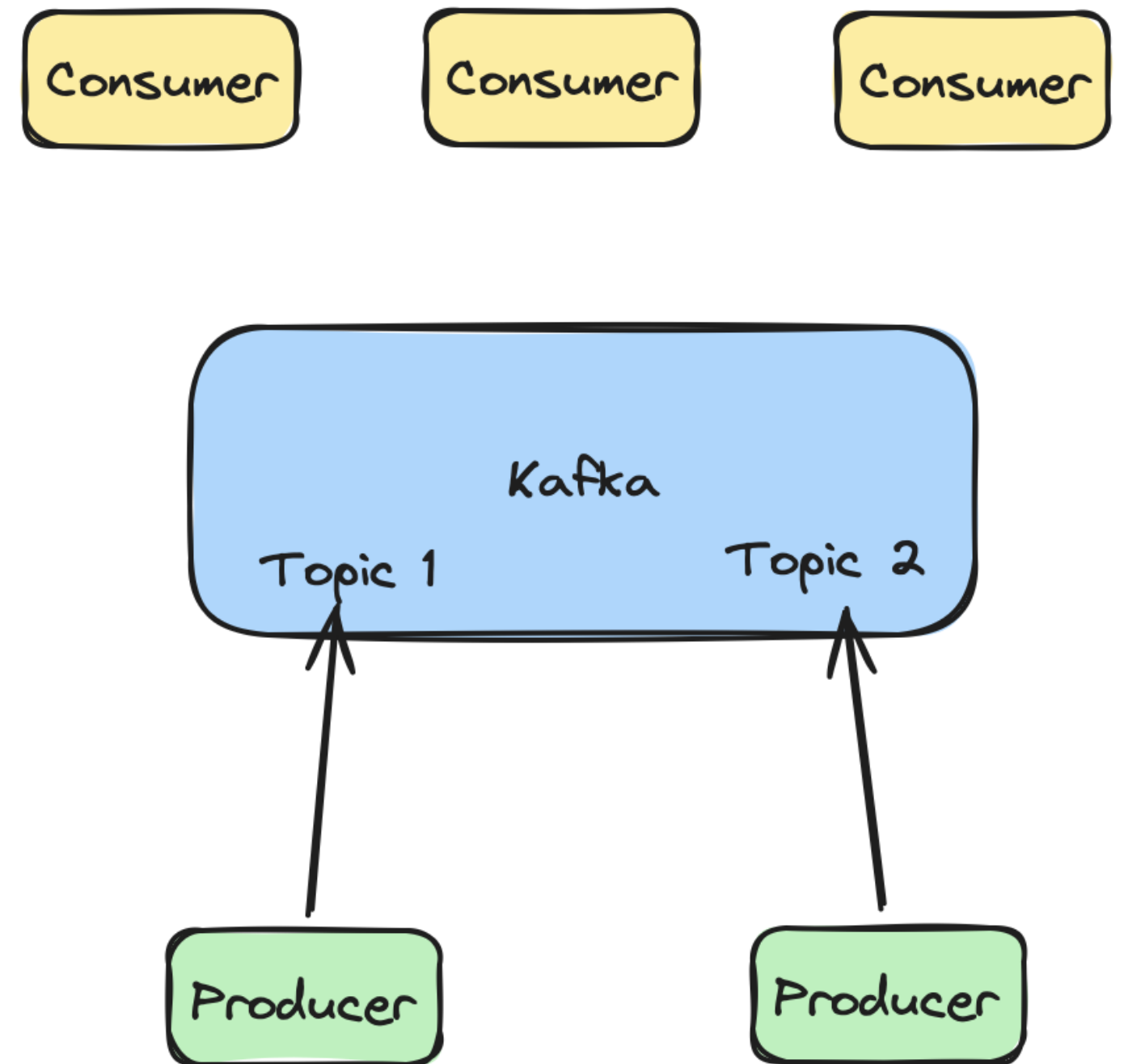
Kafka components

- User facing components
 - Topics
 - Producers (writers)
 - Consumers (readers)



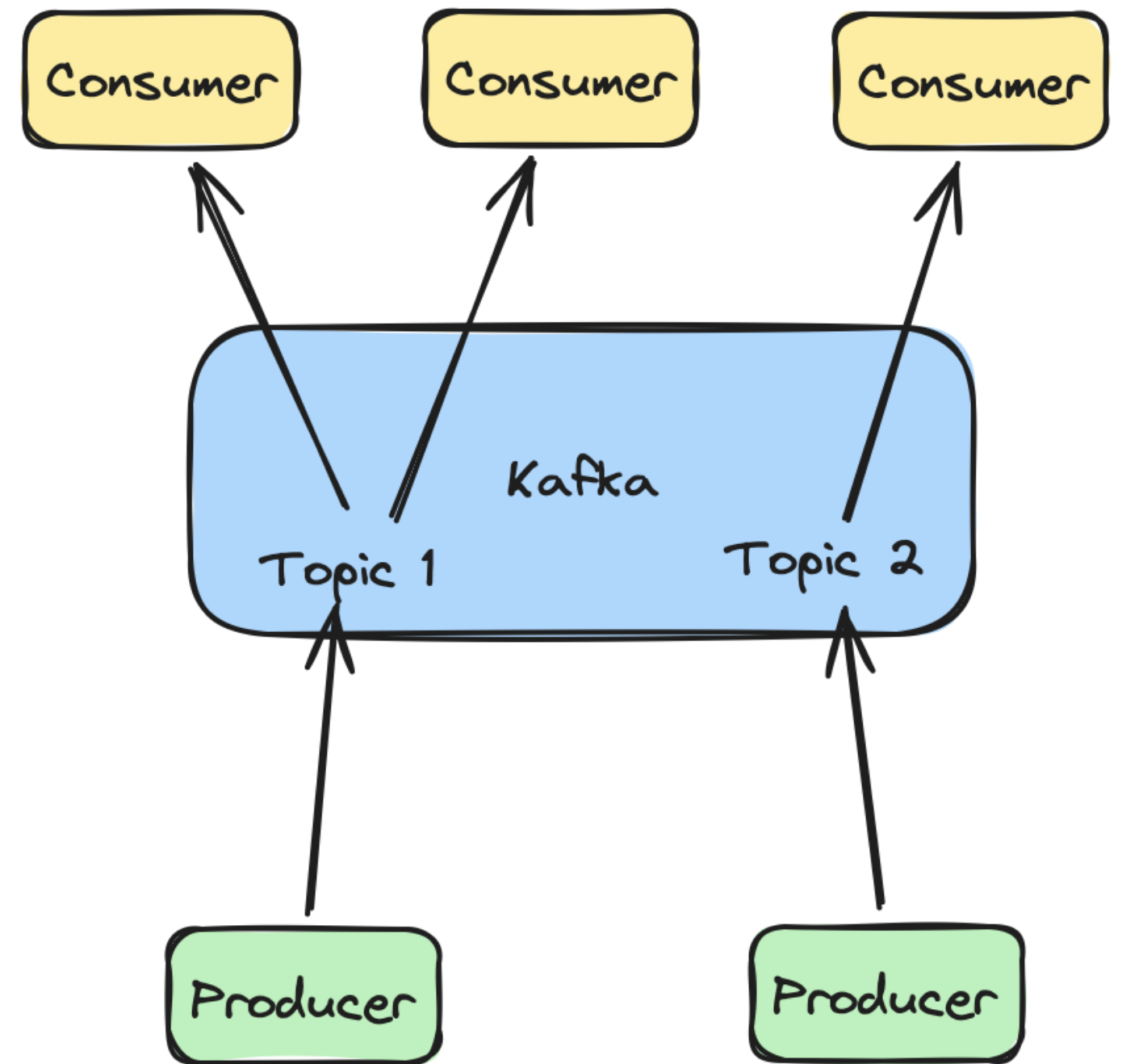
Kafka components

- User facing components
 - Topics
 - Producers (writers)
 - Consumers (readers)



Kafka components

- User facing components
 - Topics
 - Producers (writers)
 - Consumers (readers)



Let's practice!

INTRODUCTION TO KAFKA

Kafka producers

INTRODUCTION TO KAFKA



Mike Metzger

Data Engineering Consultant

What is a producer?

- Writes messages to Kafka topics
 - Sometimes called publisher
- Messages sent via producers are stored on Kafka for later use
- Can have many producers
- Producers can write to one or multiple topics

Types of producers

- Many different kinds of producers
 - Command-line
 - `kafka-console-producer.sh`
 - Python
 - Java
 - Many other languages
 - Kafka Connect

kafka-console-producer.sh

- Found in `bin/` folder, with other Kafka tools
 - `bin/kafka-console-producer.sh`
- Several command-line options, some required, others not
- Required options:
 - `--bootstrap-server` - Specifies which server to use
 - `--bootstrap-server localhost:9092`
 - `--topic`
 - `--topic phishing-sites`
- By default, opens a connection where messages can be typed
 - Use Ctrl-C to exit
- Can also use `|` symbol to pass messages

Example

- Interactive example

```
bin/kafka-console-producer.sh --bootstrap-server localhost:9092 --topic testing
```

```
>This is the first message  
>This is the second message  
>This is the third message  
<ctrl+c to exit>
```

- Non-interactive

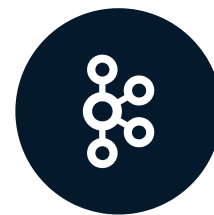
```
echo "This is the first message" |  
bin/kafka-console-producer.sh --bootstrap-server localhost:9092 --topic testing
```

Let's practice!

INTRODUCTION TO KAFKA

Kafka consumers

INTRODUCTION TO KAFKA



Mike Metzger

Data Engineering Consultant

What is a consumer?

- Reads messages from Kafka topics
 - Sometimes called subscribers
- Kafka provides messages stored in topics to consumers
- Can have many consumers
- Consumers can read from one or multiple topics

Types of consumers

- Many different kinds of consumers
 - Command-line
 - `kafka-console-consumer.sh`
 - Python
 - Java
 - Many other languages
 - Kafka Connect

kafka-console-consumer.sh

- Found in `bin/` folder, with other Kafka tools
 - `bin/kafka-console-consumer.sh`
- Several options, some required
- Required options same as `kafka-console-producer` :
 - `--topic`
 - `--bootstrap-server`
- By default, opens a connection where messages appear (real-time) and maintain position.
- Use Ctrl-C to exit

Optional arguments

- `--from-beginning`
 - Tells the Kafka system to send all messages in topic, even ones that may have been already read.
- `--max-messages <number>`
 - Used to read up to a maximum number of messages before exiting

Interactive example

- Interactive example

```
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 \  
  --topic testing --from-beginning
```

```
This is the first message  
This is the second message  
This is the third message  
<ctrl+c to exit>
```


Non-interactive example

- Non-interactive

```
bin/kafka-console-consumer.sh --bootstrap-server localhost:9092 /  
  --topic testing --from-beginning --max-messages 2
```

```
This is the first message  
This is the second message  
<automatically exits>
```

Let's practice!

INTRODUCTION TO KAFKA