BDT CCA 3

APACHE KAFKA

GROUP NO –

RONAK PATIDAR

HARSH KATIYAR

AMAN KANTHALIA

KRISHNA HITNALIKAR

INDEX

INTRODUCTION

Apache Kafka is a **distributed event store** and **stream-processing** platform developed by the Apache Software Foundation. Built for **real-time data management,** it allows efficient **publishing and subscribing** to data streams. Kafka Connect enables smooth integration with other applications, while Kafka Streams supports **real-time data processing**. Originally created at **LinkedIn** and open-sourced in 2011, Kafka is now widely used for its scalability, reliability, and ability to handle large volumes of data in real time.

APACHE KAFKA CORE CONCEPTS

Topics - A topic in Kafka is a category or feed name to which records are sent. Data in Kafka is stored in topics, which are partitioned and replicated across the Kafka cluster.

Producers - Producers are client applications that send records to Kafka topics. Producers can decide which partition a record should go to based on the key of the record.

Consumers - Consumers read data from Kafka topics. A consumer subscribes to one or more topics and processes the records.

Brokers - A Kafka broker is a server that stores and serves the data in Kafka. Each broker is responsible for maintaining the data for its partitions and handling data requests from producers and consumers.

APACHE KAFKA CORE CONCEPTS

Partitions - A topic is split into partitions to allow Kafka to scale horizontally. Each partition is an ordered sequence of records, and Kafka guarantees order within a partition.

Offsets - Each record in a partition has an offset, a unique ID that identifies the position of the record within the partition.

Replication - Kafka provides replication of partitions for fault tolerance. Each partition has a "leader" and multiple "replicas." If the leader fails, one of the replicas becomes the new leader.

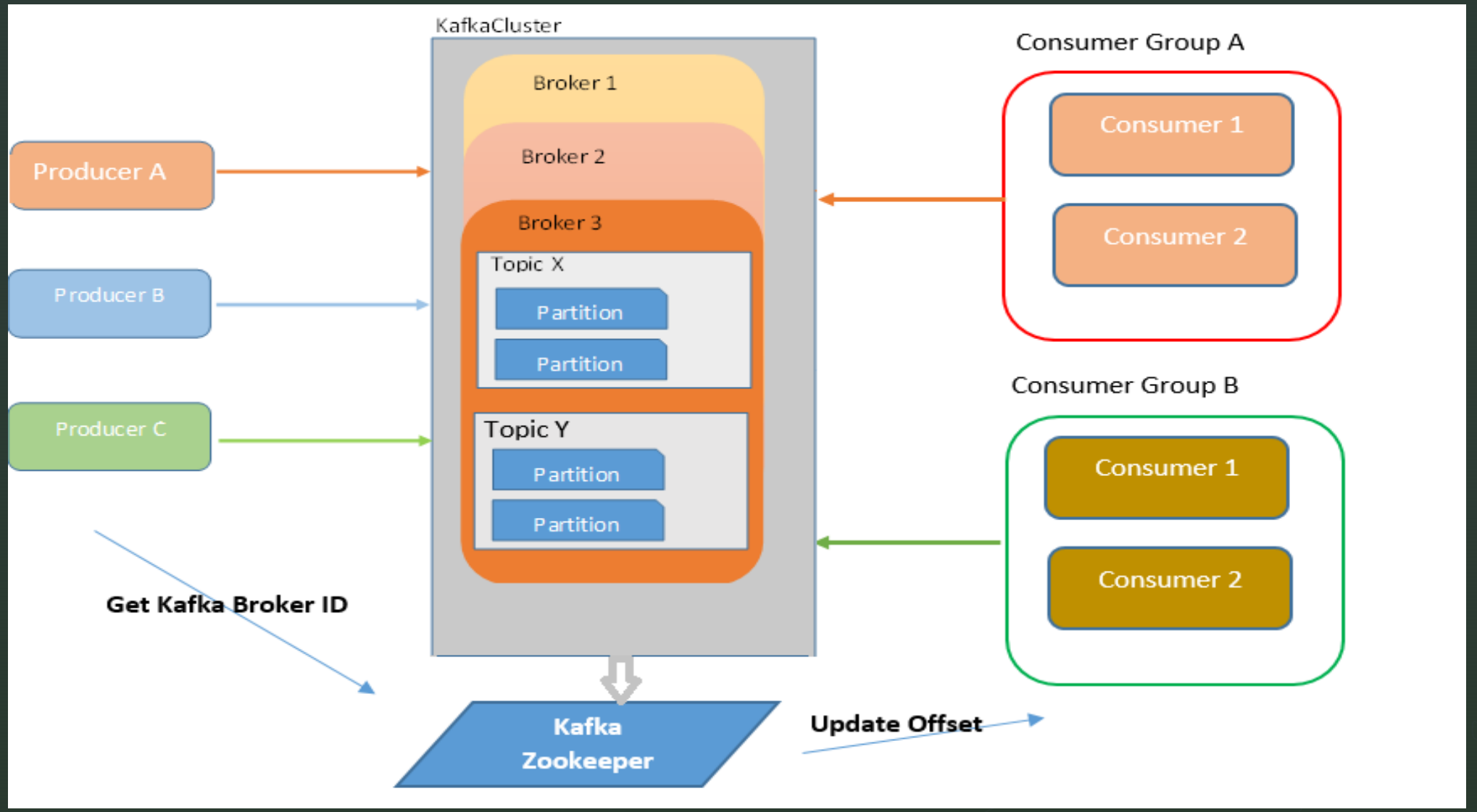
Consumer Groups - Kafka consumers can be grouped into consumer groups, where each consumer in the group reads from different partitions. This allows for scaling the consumption of data.

APACHE KAFKA CORE CONCEPTS

Zookeeper - Kafka uses Zookeeper to manage metadata, track cluster health, leader election, and configuration changes. It ensures fault tolerance in the Kafka cluster.

Log Compaction - Kafka provides log compaction, which ensures that only the most recent record for a key is retained in a topic, useful for keeping latest state changes.

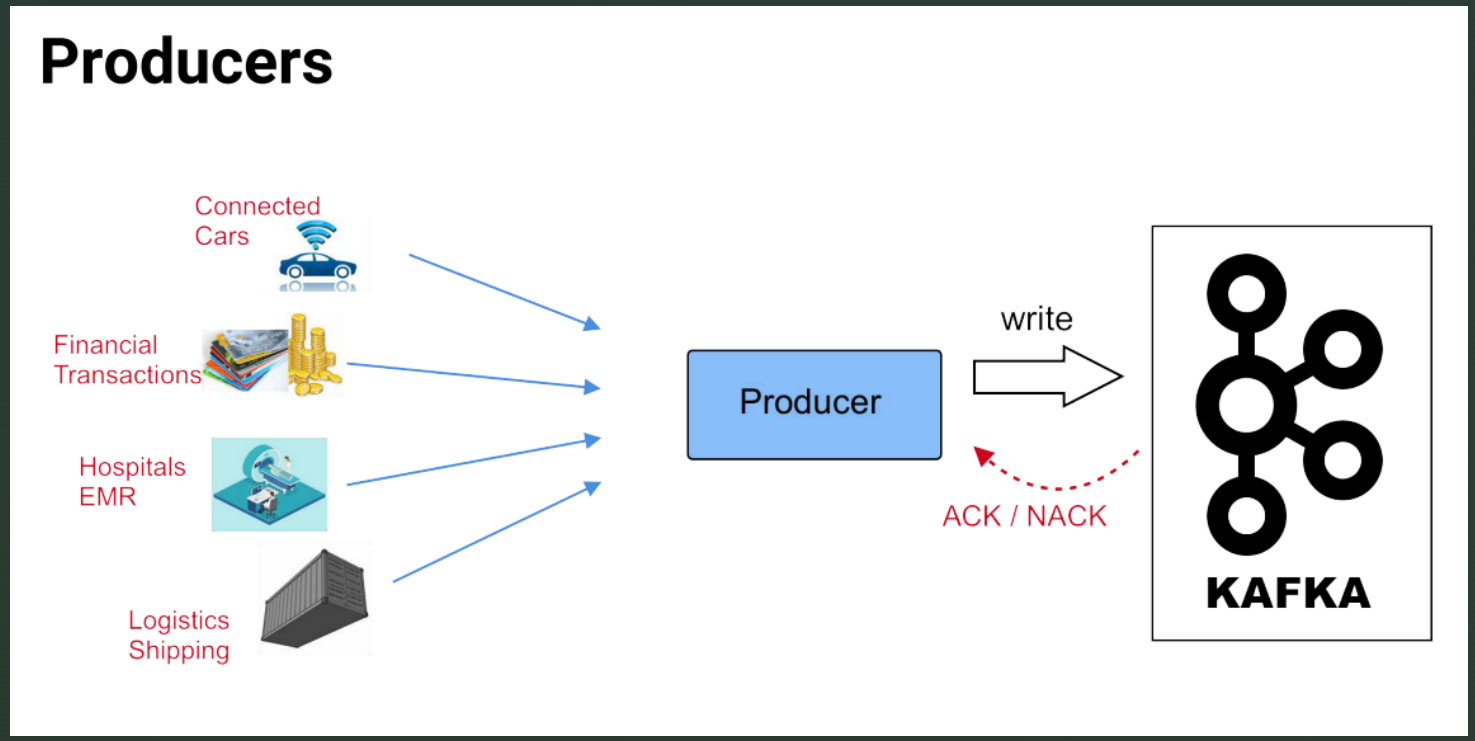
APACHE KAFKA ARCHITECTURE



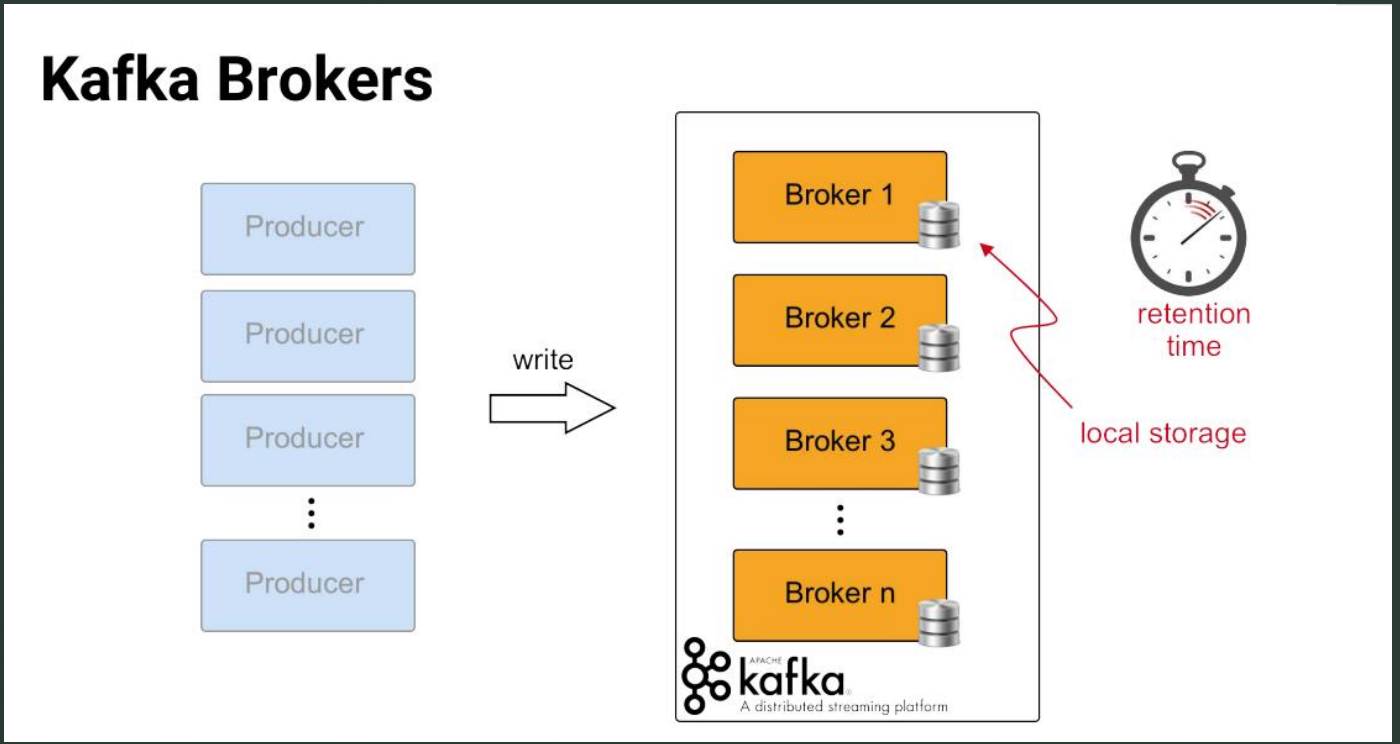
APACHE KAFKA ARCHITECTURE

Apache Kafka's architecture is a distributed system with key components: producers publish messages to topics (divided into partitions), brokers manage and replicate data across servers for fault tolerance, and consumers process messages in real time. Kafka also offers Kafka Connect for external data integration and Kafka Streams for stream processing. This design ensures high throughput, low latency, scalability, and reliable data handling.

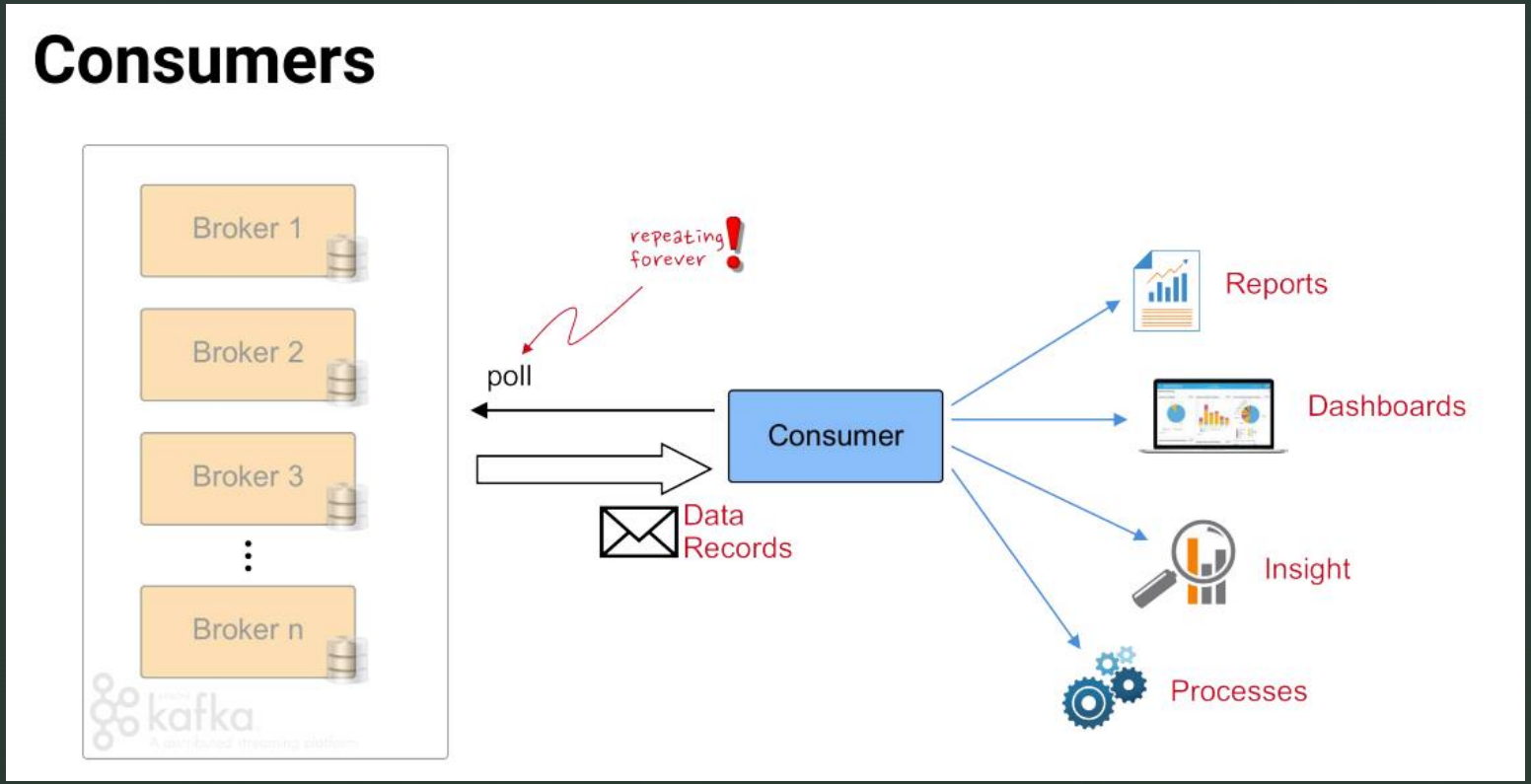
KAFKA PRODUCER AND CONSUMER



Producer:
Producers are responsible for publishing messages to Kafka topics. They send data to specific topics, which are divided into partitions. The producer controls which partition a message goes to, often based on keys or custom logic, ensuring efficient data distribution.



Broker:
Brokers are Kafka servers that manage topics and partitions. They store incoming messages and handle the replication of data across multiple brokers, ensuring fault tolerance and high availability. Brokers play a critical role in distributing and managing the load in a Kafka cluster.



Consumer:
Consumers read messages from Kafka topics, processing them in real-time. They subscribe to one or more topics and can process messages individually or in batches. Consumers are designed to be fault-tolerant and can reprocess data if needed, ensuring that no messages are lost.

KEY FEATURES

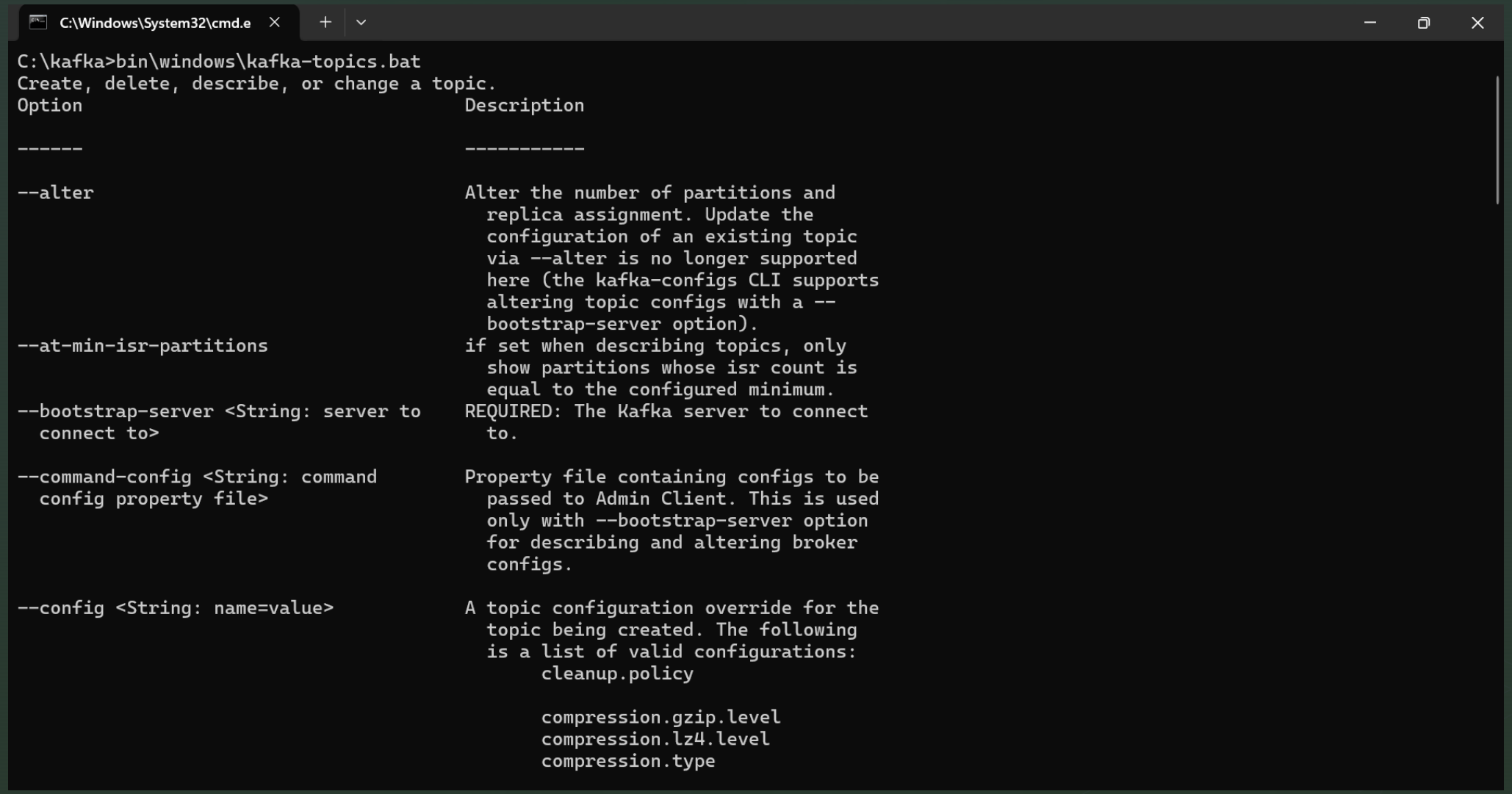
•**High Throughput**:
•Kafka is designed to handle large volumes of data at high speed. It can process millions of messages per second with minimal delay.
•This makes it ideal for applications requiring fast, real-time data processing, such as streaming platforms, monitoring systems, or financial transactions.

•**Scalability and Fault Tolerant:**
•Kafka scales by adding brokers and partitioning data to handle growing loads, while its fault tolerance ensures high availability through data replication across brokers.

•**Real-Time Stream Processing**:
•Kafka supports real-time stream processing, allowing you to process data as it arrives.
•Tools like **Kafka Streams** and **Apache Flink** integrate with Kafka, enabling complex event processing, aggregations, and transformations in real time, which is crucial for applications like fraud detection, monitoring systems, and recommendation engines.

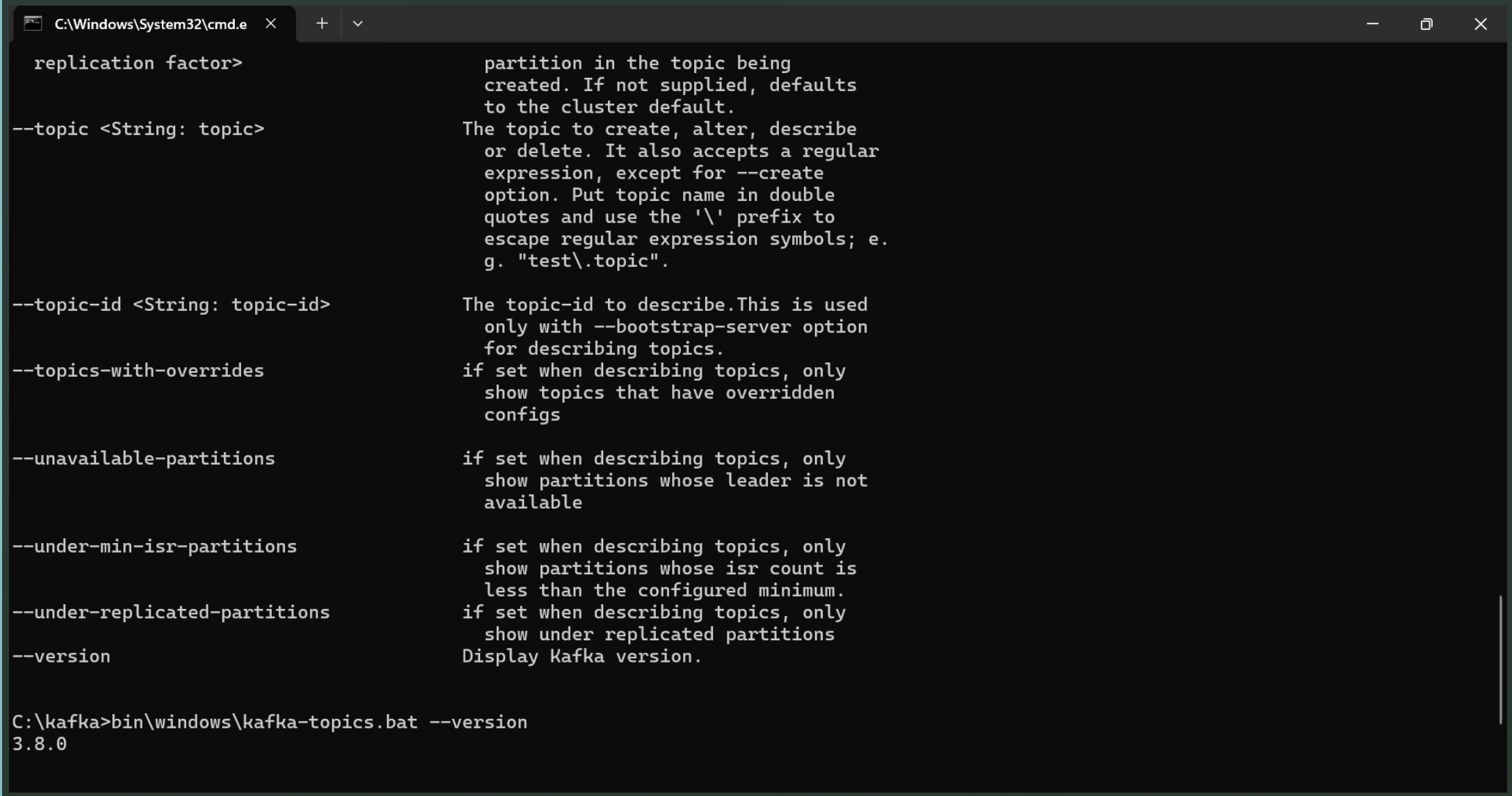
COMPANIES THAT USE KAFKA -

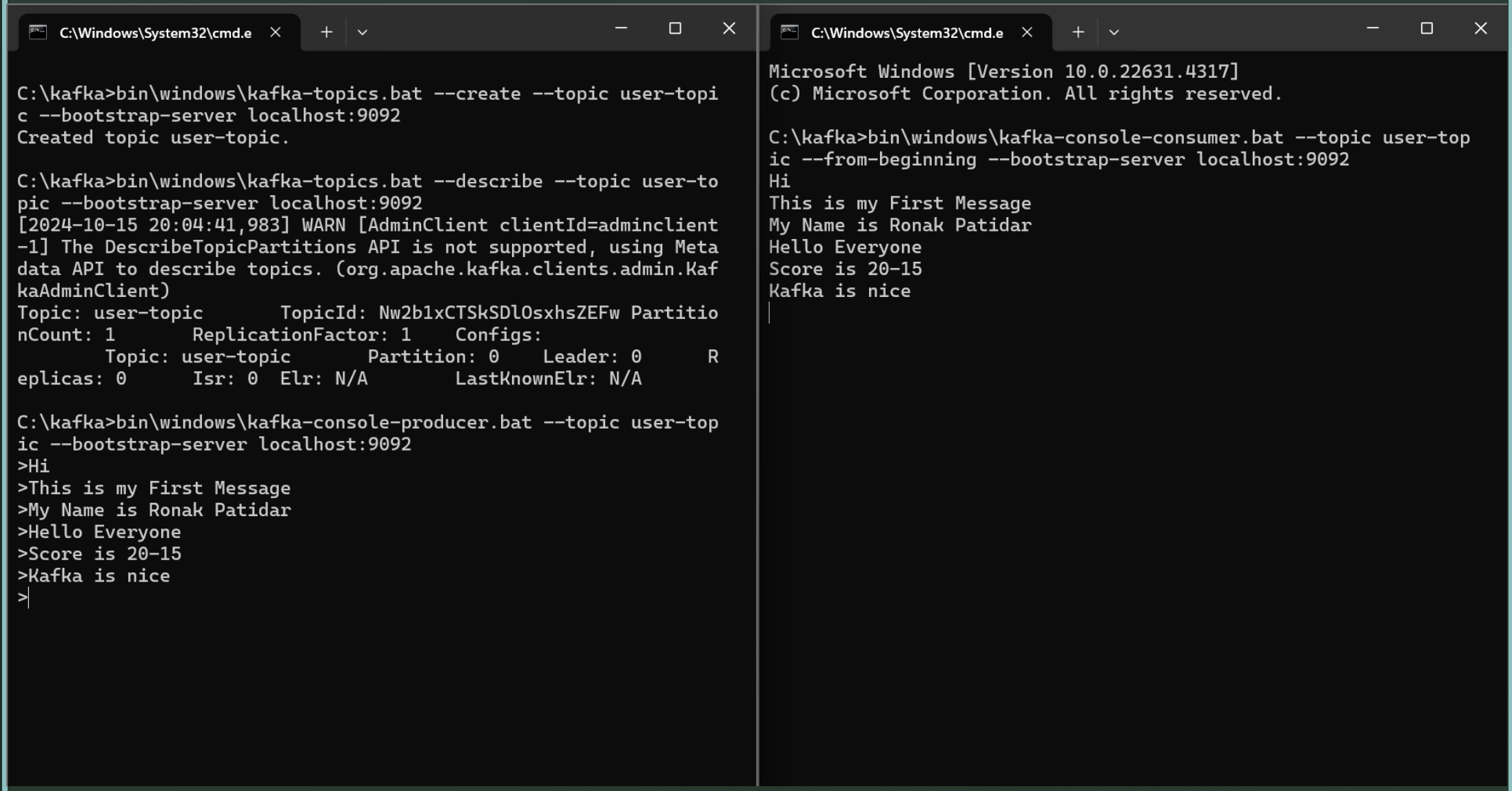
IMPLEMENTATION

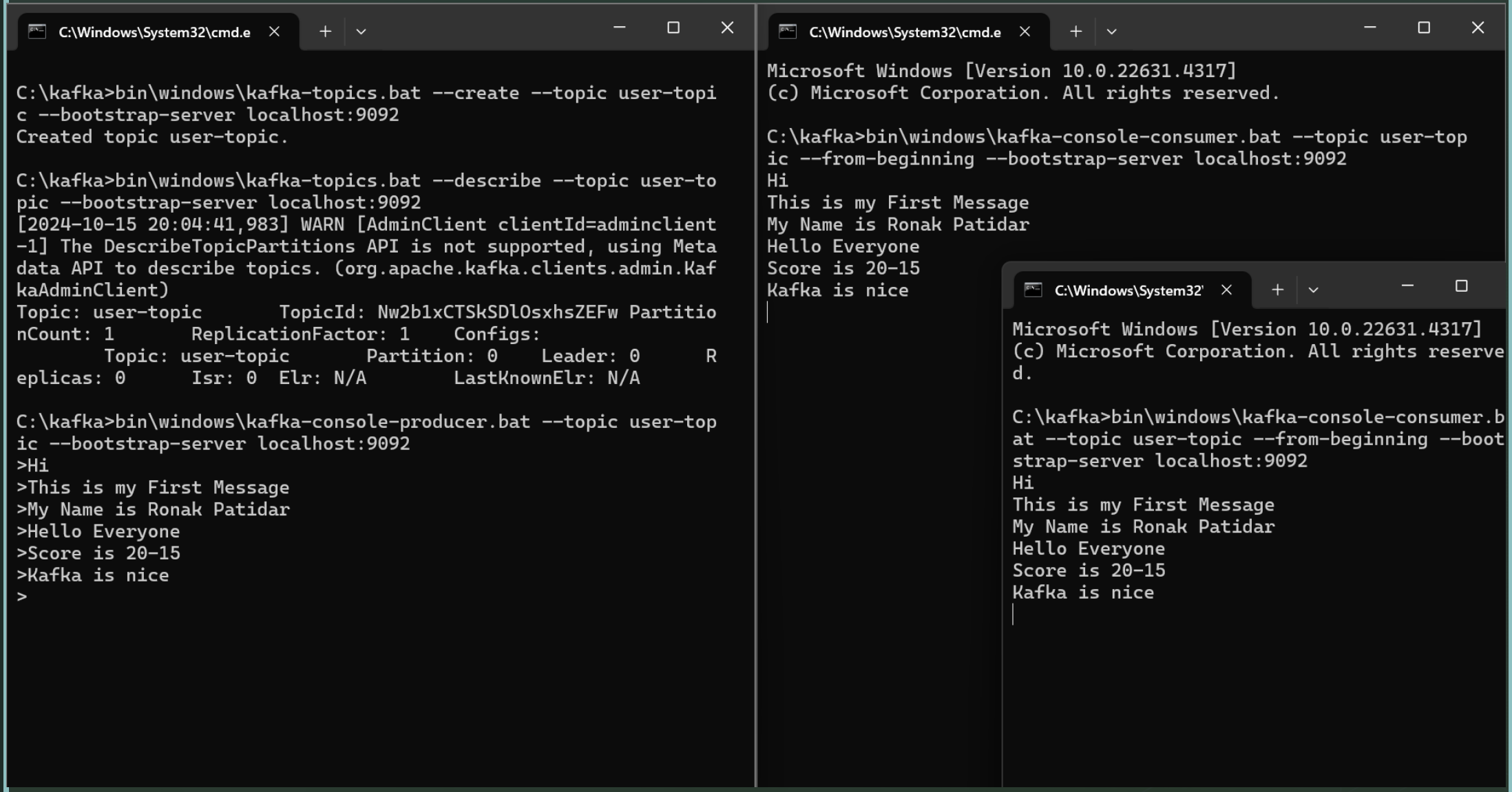


C:\Windows\System32\cmd.e

```
                                          supports altering topic configs with
                                          a --bootstrap-server option).
--create                                  Create a new topic.

--delete                                  Delete a topic

--delete-config <String: name>            A topic configuration override to be
                                          removed for an existing topic (see
                                          the list of configurations under the
                                          --config option). Not supported with
                                          the --bootstrap-server option.
--describe                                List details for the given topics.
--exclude-internal                        exclude internal topics when running
                                          list or describe command. The
                                          internal topics will be listed by
                                          default

--help                                    Print usage information.
--if-exists                               if set when altering or deleting or
                                          describing topics, the action will
                                          only execute if the topic exists.
--if-not-exists                           if set when creating topics, the
                                          action will only execute if the
                                          topic does not already exist.
--list                                    List all available topics.
--partition-size-limit-per-response       the maximum partition size to be
  <Integer: maximum number of             included in one

  partitions in one response.>            DescribeTopicPartitions response.
--partitions <Integer: # of partitions>   The number of partitions for the topic
                                          being created or altered (WARNING:
                                          If partitions are increased for a
                                          topic that has a key, the partition
```


C:\Windows\System32\cmd.e
replication factor>
partition in the topic being
created. If not supplied, defaults
to the cluster default.
--topic <String: topic>
The topic to create, alter, describe
or delete. It also accepts a regular
expression, except for --create
option. Put topic name in double
quotes and use the '\' prefix to
escape regular expression symbols; e.
g. "test\.topic".
--topic-id <String: topic-id>
The topic-id to describe.This is used
only with --bootstrap-server option
for describing topics.
--topics-with-overrides
if set when describing topics, only
show topics that have overridden
configs
--unavailable-partitions
if set when describing topics, only
show partitions whose leader is not
available
--under-min-isr-partitions
if set when describing topics, only
show partitions whose isr count is
less than the configured minimum.
--under-replicated-partitions
if set when describing topics, only
show under replicated partitions
--version
Display Kafka version.
C:\kafka>bin\windows\kafka-topics.bat --version
3.8.0


C:\Windows\System32\cmd.e
C:\kafka>bin\windows\kafka-topics.bat --create --topic user-topi
c --bootstrap-server localhost:9092
Created topic user-topic.
C:\kafka>bin\windows\kafka-topics.bat --describe --topic user-to
pic --bootstrap-server localhost:9092
[2024-10-15 20:04:41,983] WARN [AdminClient clientId=adminclient
-1] The DescribeTopicPartitions API is not supported, using Meta
data API to describe topics. (org.apache.kafka.clients.admin.Kaf
kaAdminClient)
Topic: user-topic TopicId: Nw2b1xCTSkSDlOsxhsZEFw Partitio
nCount: 1 ReplicationFactor: 1 Configs:
Topic: user-topic Partition: 0 Leader: 0 R
eplicas: 0 Isr: 0 Elr: N/A LastKnownElr: N/A
C:\kafka>bin\windows\kafka-console-producer.bat --topic user-top
ic --bootstrap-server localhost:9092
>Hi
>This is my First Message
>My Name is Ronak Patidar
>Hello Everyone
>Score is 20-15
>Kafka is nice
>
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.
C:\kafka>bin\windows\kafka-console-consumer.bat --topic user-top
ic --from-beginning --bootstrap-server localhost:9092
Hi
This is my First Message
My Name is Ronak Patidar
Hello Everyone
Score is 20-15
Kafka is nice


C:\Windows\System32\cmd.e
C:\kafka>bin\windows\kafka-topics.bat --create --topic user-topi
c --bootstrap-server localhost:9092
Created topic user-topic.
C:\kafka>bin\windows\kafka-topics.bat --describe --topic user-to
pic --bootstrap-server localhost:9092
[2024-10-15 20:04:41,983] WARN [AdminClient clientId=adminclient
-1] The DescribeTopicPartitions API is not supported, using Meta
data API to describe topics. (org.apache.kafka.clients.admin.Kaf
kaAdminClient)
Topic: user-topic TopicId: Nw2b1xCTSkSDlOsxhsZEFw Partitio
nCount: 1 ReplicationFactor: 1 Configs:
Topic: user-topic Partition: 0 Leader: 0 R
eplicas: 0 Isr: 0 Elr: N/A LastKnownElr: N/A
C:\kafka>bin\windows\kafka-console-producer.bat --topic user-top
ic --bootstrap-server localhost:9092
>Hi
>This is my First Message
>My Name is Ronak Patidar
>Hello Everyone
>Score is 20-15
>Kafka is nice
>
C:\Windows\System32\cmd.e
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.
C:\kafka>bin\windows\kafka-console-consumer.bat --topic user-top
ic --from-beginning --bootstrap-server localhost:9092
Hi
This is my First Message
My Name is Ronak Patidar
Hello Everyone
Score is 20-15
Kafka is nice
C:\Windows\System32'
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserve
d.
C:\kafka>bin\windows\kafka-console-consumer.b
at --topic user-topic --from-beginning --boot
strap-server localhost:9092
Hi
This is my First Message
My Name is Ronak Patidar
Hello Everyone
Score is 20-15
Kafka is nice

CONCLUSION

Apache Kafka is a scalable, high-performance platform for real-time data processing, enabling organizations to handle vast data streams with low latency for quick decisions. Its event-driven architecture supports decoupled systems, enhancing resilience and ease of maintenance. Kafka integrates seamlessly with big data frameworks like Apache Spark and Hadoop, making it versatile across industries. Backed by a strong community, Kafka is vital for businesses aiming to fully leverage real-time data.

THANK YOU !!