第七組

KAFKA+ZOOKEEPER
購物訂單及時後台統計

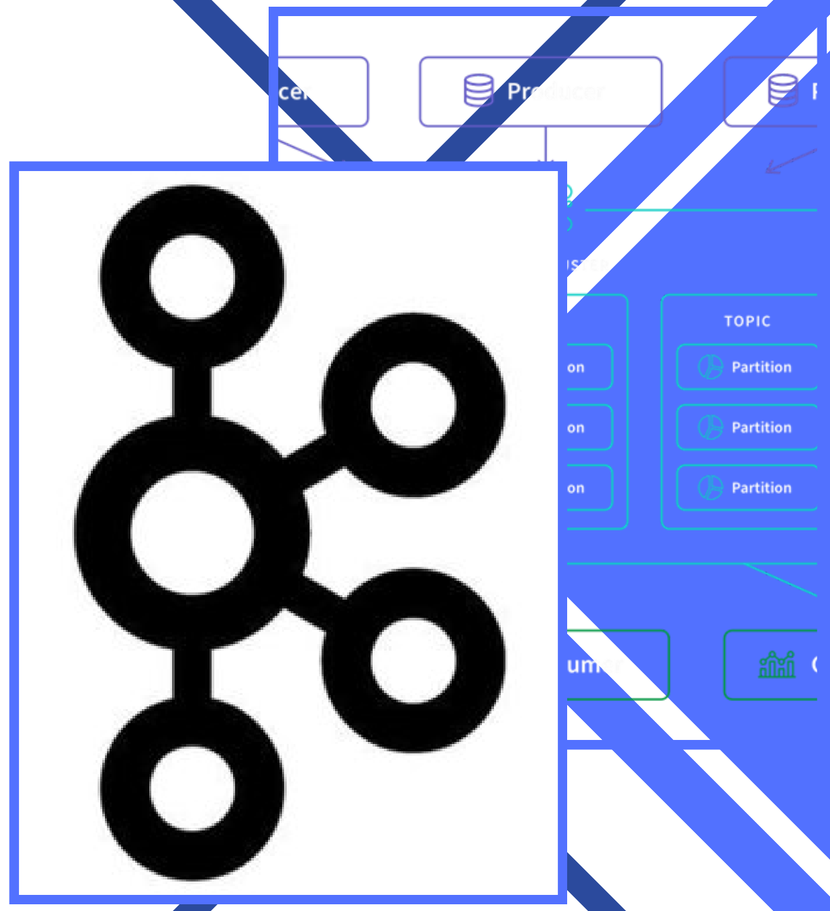
組員:楊凱旭、程翔、黃念祺、 陳韋溢、黃競、吳呈勳

OUTLINE

1 INTRODUCTION+功能介紹

2 架構解析+DEMO

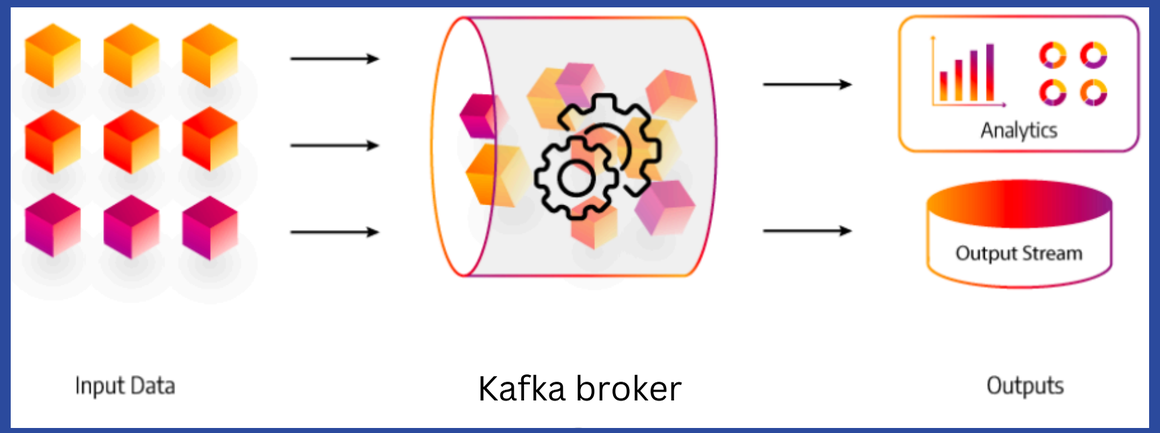
3 CONCLUSION



INTRODUCTION

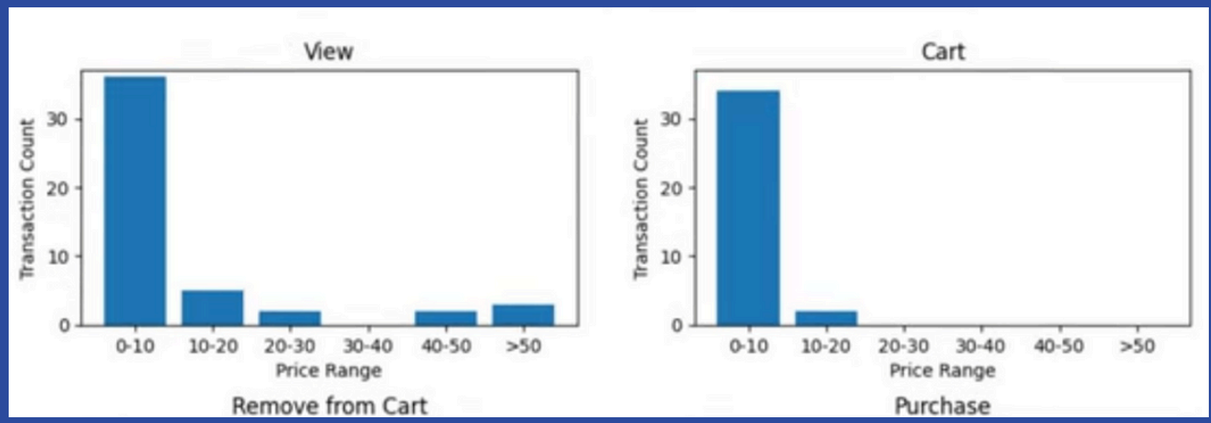
ZOOKEEPER KAFKA DOCKER SERVER AND CLIENT

Kafka and ZooKeeper work in conjunction to form a complete Kafka Cluster — with ZooKeeper providing the distributed clustering services, and Kafka handling the actual data streams and connectivity to clients.

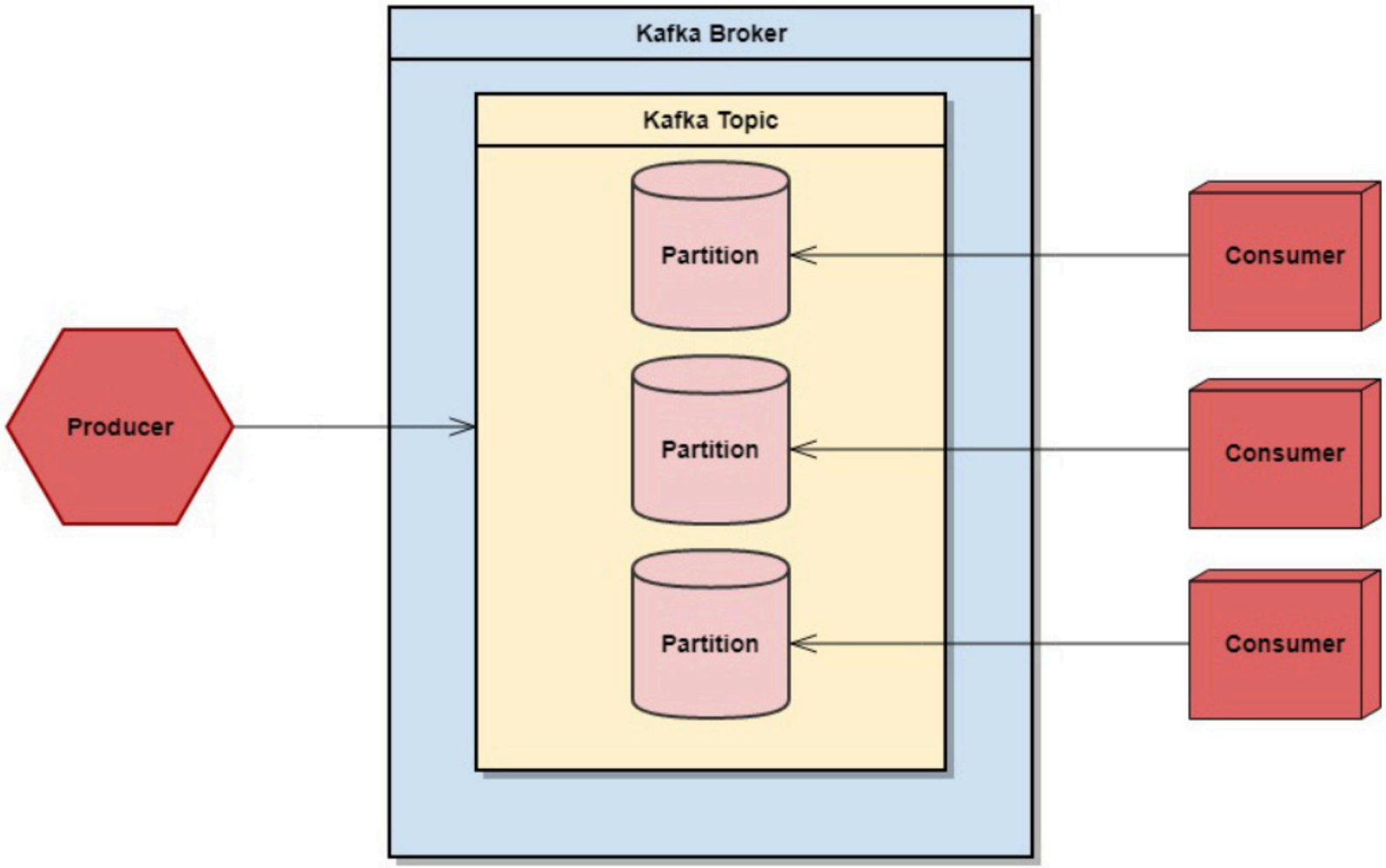


DATA ANALYTICS

File contains behavior data for a one month (December 2019) from a large multi-category online store, collected by Open CDP project. The data is classified into 4 event types View, Cart, Remove from cart and Purchase. We then show current result by graph.



架構解析

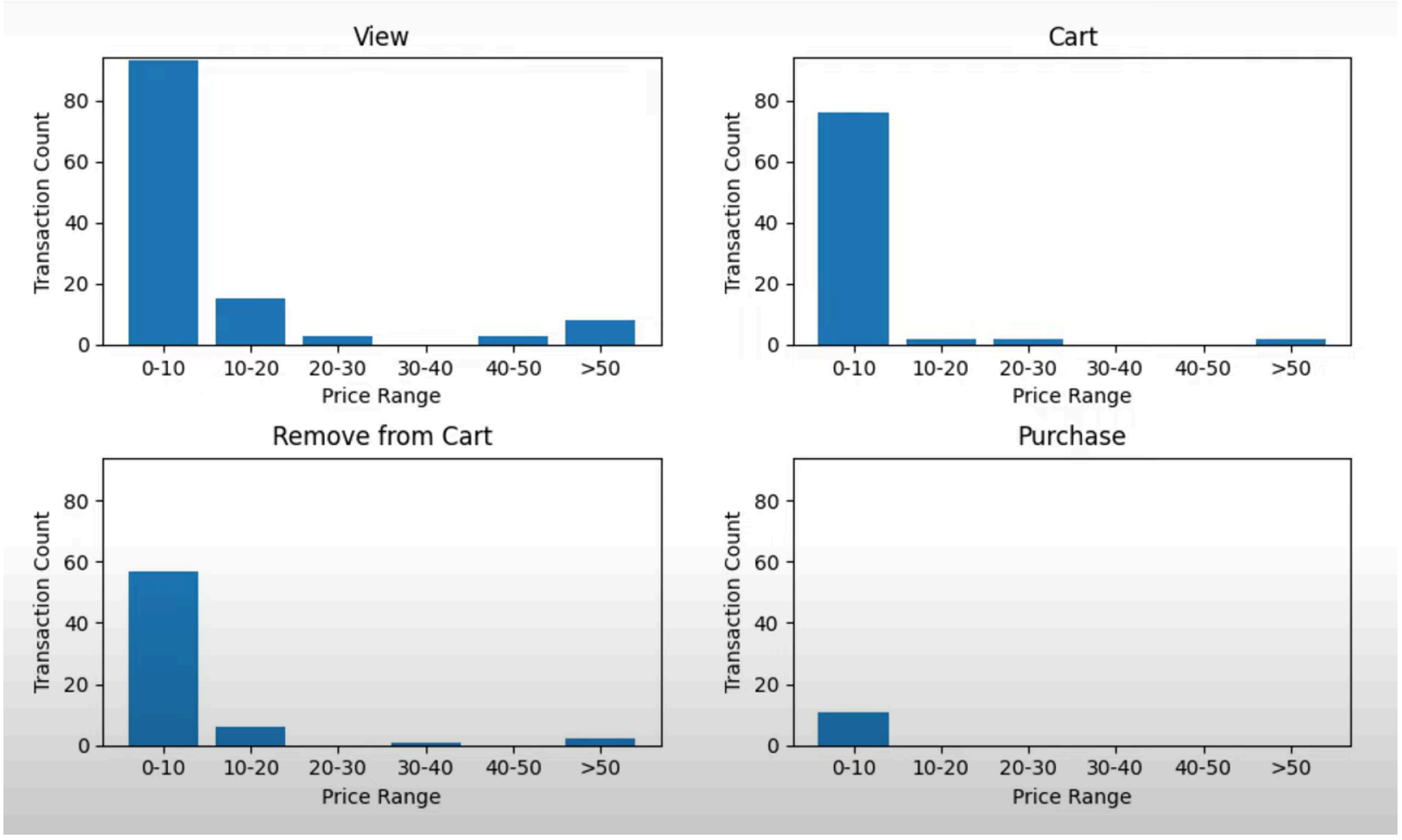

Kafka Broker
Kafka Topic
Partition
Partition
Partition
Producer
Consumer
Consumer
Consumer

DEMO

HTTPS://WWW.YOUTUBE.COM/WATCH?V=VOJ7W0CFKUM

github原始碼：

https://github.com/yuzu0230/ZookeeperClient


View
Transaction Count
0
20
40
60
80
0-10
10-20
20-30
30-40
40-50
>50
Price Range
Cart
Transaction Count
0
20
40
60
80
0-10
10-20
20-30
30-40
40-50
>50
Price Range
Remove from Cart
Transaction Count
0
20
40
60
80
0-10
10-20
20-30
30-40
40-50
>50
Price Range
Purchase
Transaction Count
0
20
40
60
80
0-10
10-20
20-30
30-40
40-50
>50
Price Range

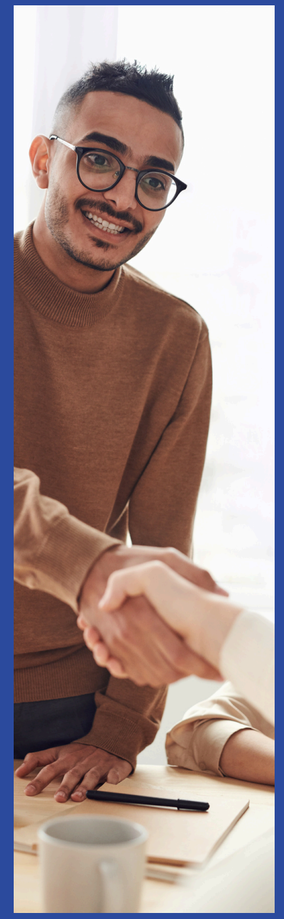
CONCLUSION

COMMENT

AS A RESULT, IN THIS ROLE OF ZOOKEEPER IN KAFKA, WE'VE SEEN THAT KAFKA RELIES HEAVILY ON ZOOKEEPER TO FUNCTION EFFECTIVELY IN A KAFKA CLUSTER.

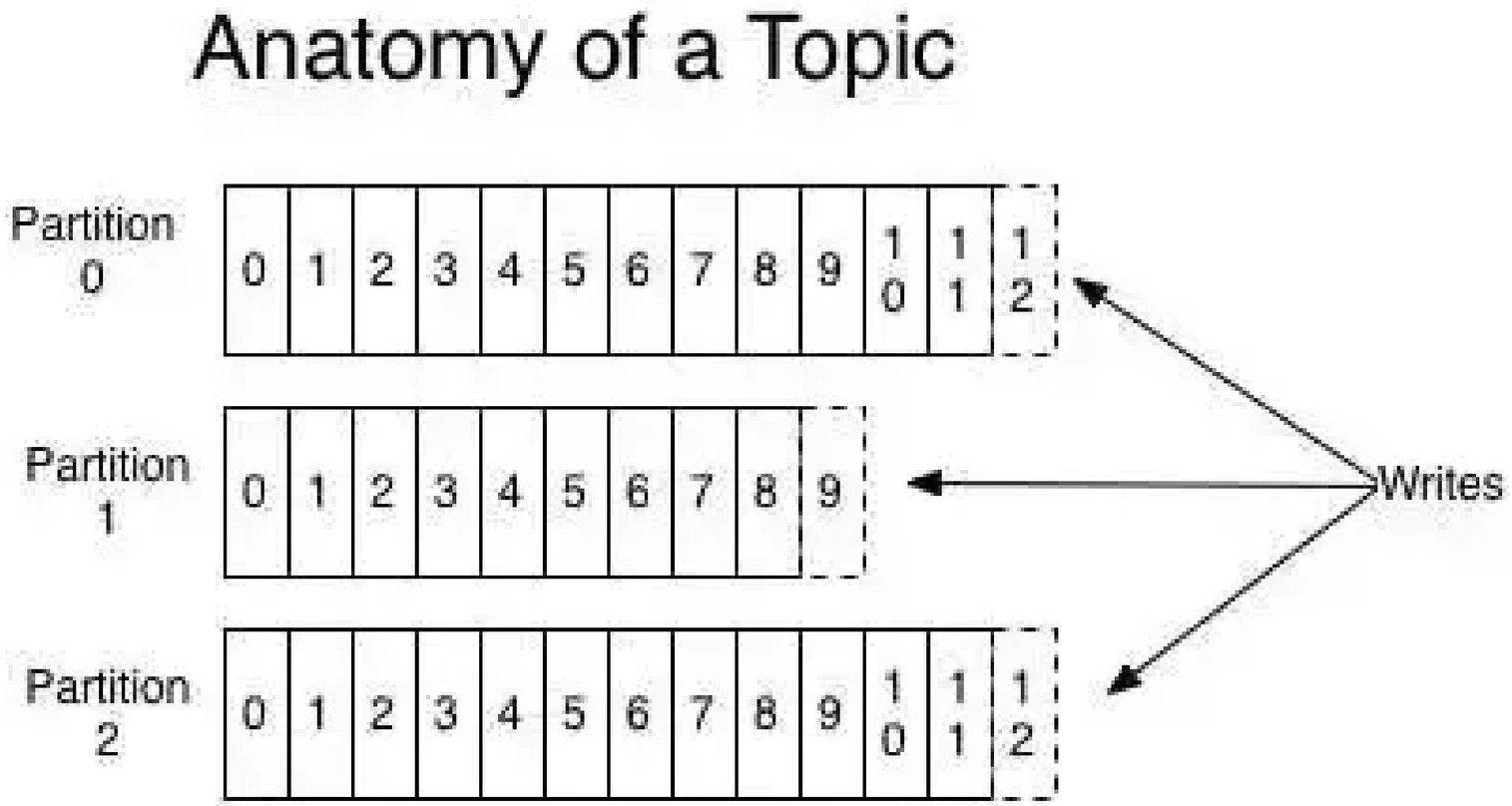
Mission

SOLVE THE LOW-LATENCY INGESTION OF LARGE AMOUNTS OF EVENT DATA FROM REAL-TIME EVENT PROCESSING SYSTEMS. KAFKA WAS DESIGNED TO PROVIDE THE INGESTION BACKBONE FOR THIS TYPE OF USE CASE



END

APPENDICES- KAFKA PARTITION



APPENDICES- DOCKER COMPOSE

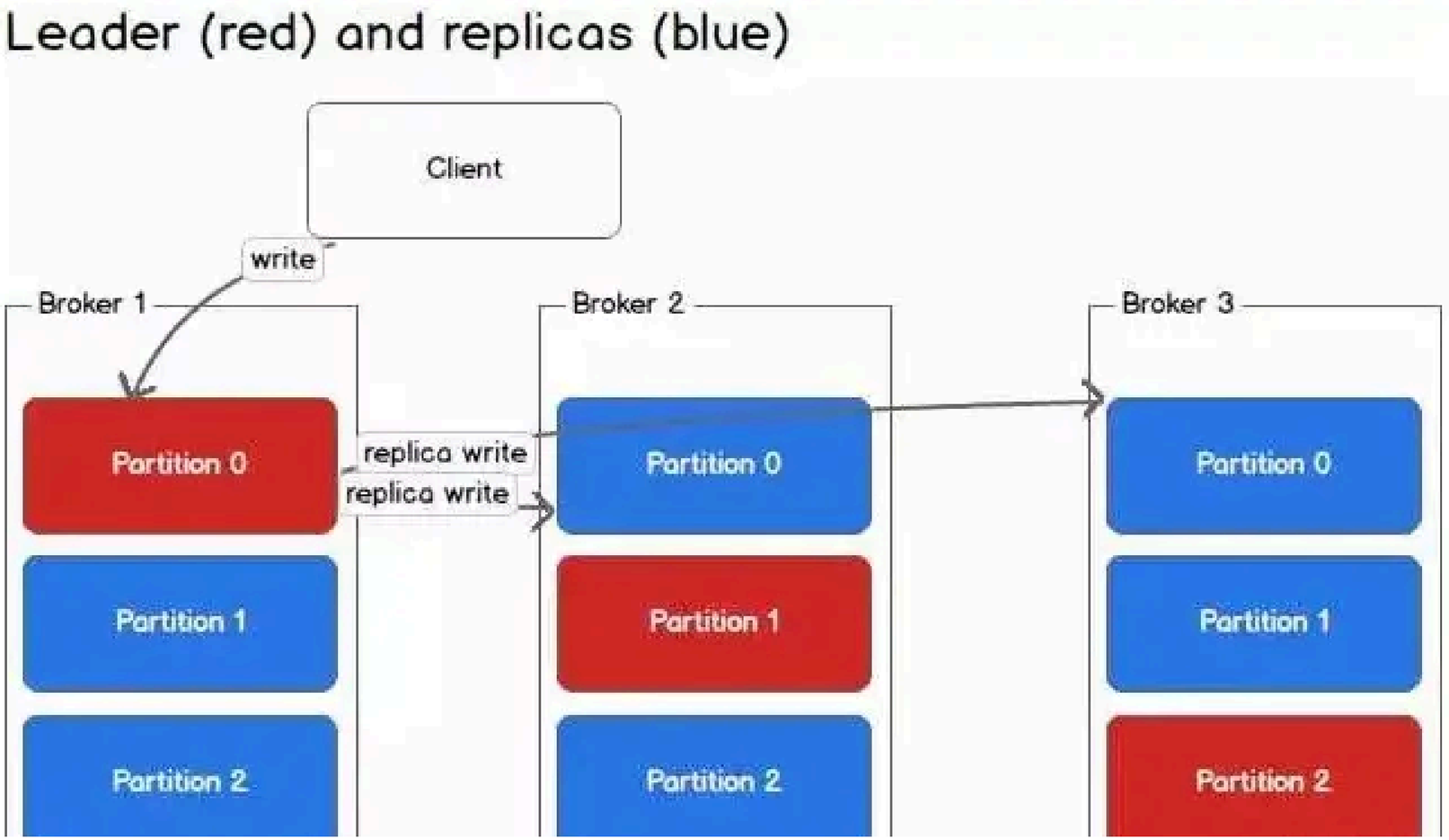
```
kafka-1:
  image: confluentinc/cp-kafka:latest
  hostname: kafka1
  depends_on:
    - zookeeper-1
    - zookeeper-2
    - zookeeper-3
  environment:
    KAFKA_BROKER_ID: 1
    KAFKA_ZOOKEEPER_CONNECT: zoo1:2181,zoo2:2181,zoo3:2181
    KAFKA_ADVERTISED_HOST_NAME: kafka1
    KAFKA_ADVERTISED_LISTENERS: INTERNAL://kafka1:9092,FROM_HOST://127.0.0.1:19092
    KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INTERNAL:PLAINTEXT,FROM_HOST:PLAINTEXT
    KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
    KAFKA_BOOTSTRAP.SERVERS: 'kafka1:9092,kafka2:9092,kafka3:9092'
  ports:
    - "19092:19092"

kafka-2:
  image: confluentinc/cp-kafka:latest
  hostname: kafka2
  depends_on:
    - zookeeper-1
    - zookeeper-2
    - zookeeper-3
  environment:
    KAFKA_BROKER_ID: 2
    KAFKA_ZOOKEEPER_CONNECT: zoo1:2181,zoo2:2181,zoo3:2181
    KAFKA_ADVERTISED_HOST_NAME: kafka2
    KAFKA_ADVERTISED_LISTENERS: INTERNAL://kafka2:9092,FROM_HOST://127.0.0.1:29092
    KAFKA_LISTENER_SECURITY_PROTOCOL_MAP: INTERNAL:PLAINTEXT,FROM_HOST:PLAINTEXT
    KAFKA_INTER_BROKER_LISTENER_NAME: INTERNAL
    KAFKA_BOOTSTRAP.SERVERS: 'kafka1:9092,kafka2:9092,kafka3:9092'
  ports:
    - "29092:29092"

kafka-3:
  image: confluentinc/cp-kafka:latest
  hostname: kafka3
  depends_on:
```

APPENDICES- DOCKER COMPOSE

```
zookeeper-1:
  image: confluentinc/cp-zookeeper:latest
  hostname: zoo1
  environment:
    ZOOKEEPER_SERVER_ID: 1
    ZOOKEEPER_CLIENT_PORT: 2181
    ZOOKEEPER_TICK_TIME: 2000
    ZOOKEEPER_INIT_LIMIT: 5
    ZOOKEEPER_SYNC_LIMIT: 2
    ZOOKEEPER_SERVERS: zoo1:2888:3888;zoo2:2888:3888;zoo3:2888:3888
  ports:
    - 12181:2181
zookeeper-2:
  image: confluentinc/cp-zookeeper:latest
  hostname: zoo2
  environment:
    ZOOKEEPER_SERVER_ID: 2
    ZOOKEEPER_CLIENT_PORT: 2181
    ZOOKEEPER_TICK_TIME: 2000
    ZOOKEEPER_INIT_LIMIT: 5
    ZOOKEEPER_SYNC_LIMIT: 2
    ZOOKEEPER_SERVERS: zoo1:2888:3888;zoo2:2888:3888;zoo3:2888:3888
  ports:
    - 22181:2181

zookeeper-3:
  image: confluentinc/cp-zookeeper:latest
  hostname: zoo3
  environment:
    ZOOKEEPER_SERVER_ID: 3
    ZOOKEEPER_CLIENT_PORT: 2181
    ZOOKEEPER_TICK_TIME: 2000
    ZOOKEEPER_INIT_LIMIT: 5
    ZOOKEEPER_SYNC_LIMIT: 2
    ZOOKEEPER_SERVERS: zoo1:2888:3888;zoo2:2888:3888;zoo3:2888:3888
  ports:
    - 32181:2181
```

APPENDICES- KAFKA CLIENT