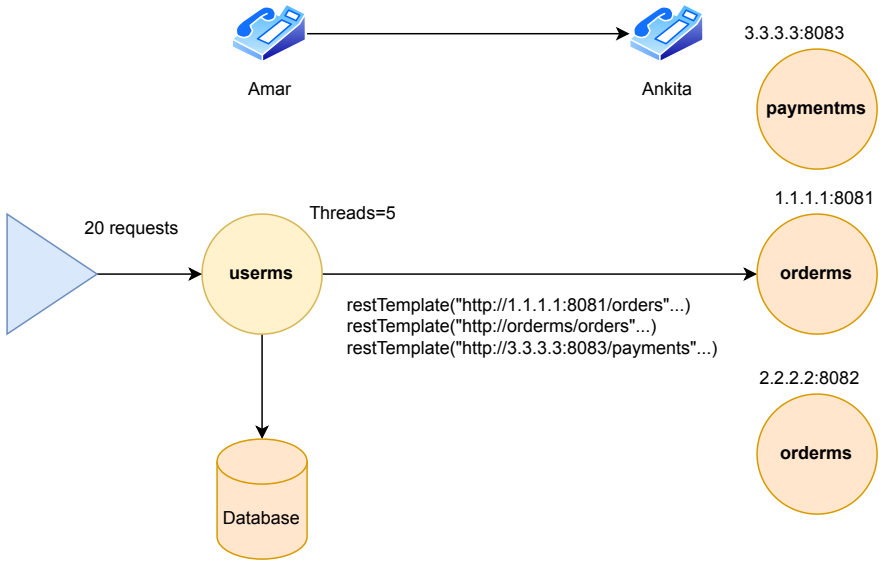
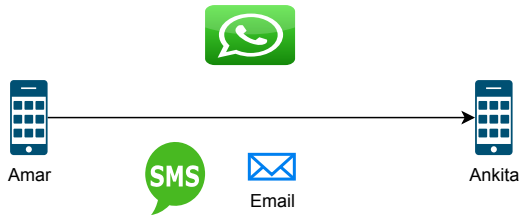
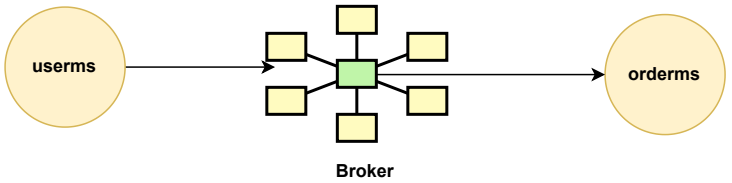
REST vs Messaging

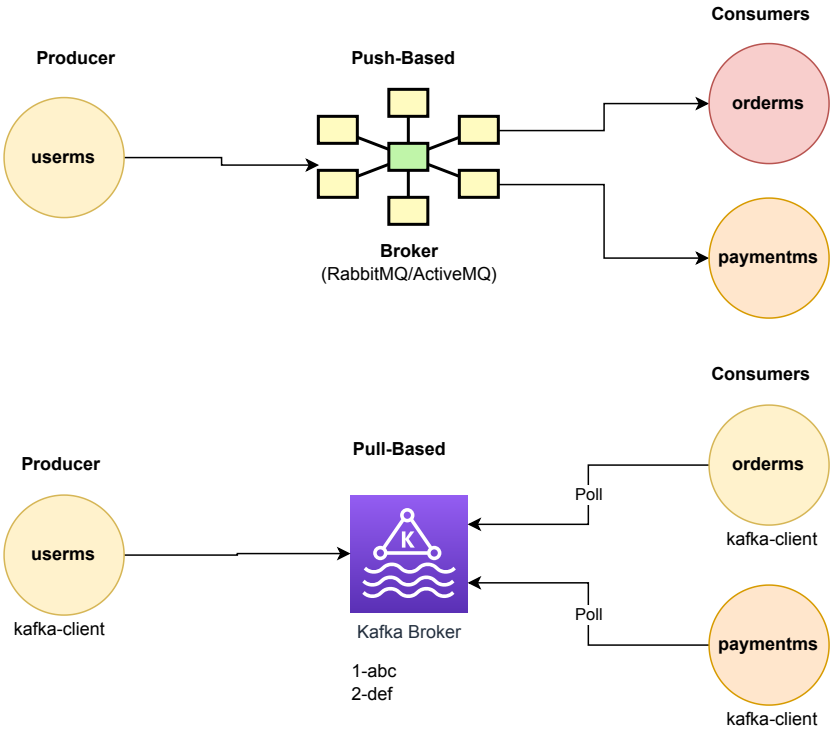


Messaging

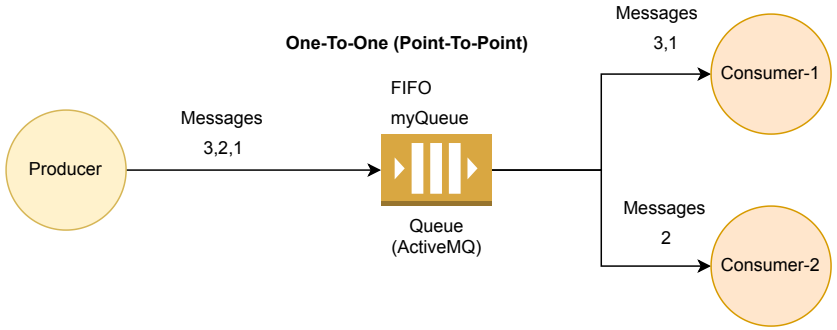


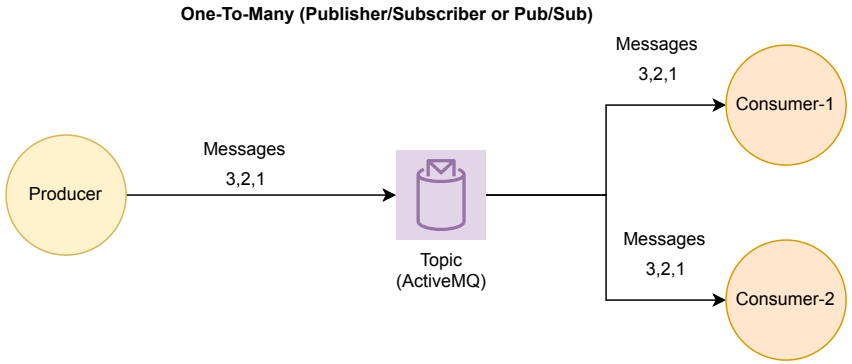


(ActiveMQ, RabbitMQ, Kafka)


Consumers
Producer
Push-Based
userms
orderms
paymentms
Broker
(RabbitMQ/ActiveMQ)
Consumers
Producer
Pull-Based
orderms
Poll
userms
kafka-client
Kafka Broker
1-abc
2-def
Poll
paymentms
kafka-client

Messaging


One-To-One (Point-To-Point)
Producer
Messages
3,2,1
FIFO
myQueue
Queue
(ActiveMQ)
Messages
3,1
Consumer-1
Messages
2
Consumer-2


One-To-Many (Publisher/Subscriber or Pub/Sub)
Producer
Messages
3,2,1
Topic
(ActiveMQ)
Messages
3,2,1
Consumer-1
Messages
3,2,1
Consumer-2

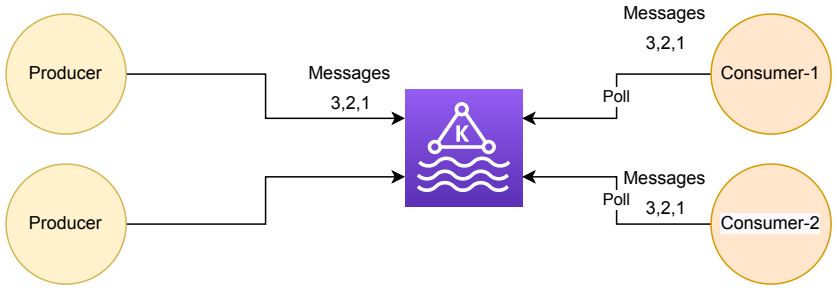
Kafka

Initially developed at LinkedIn
Later came into Apache Open source Licence

1. Apache Kafka
2. Confluent Kafka

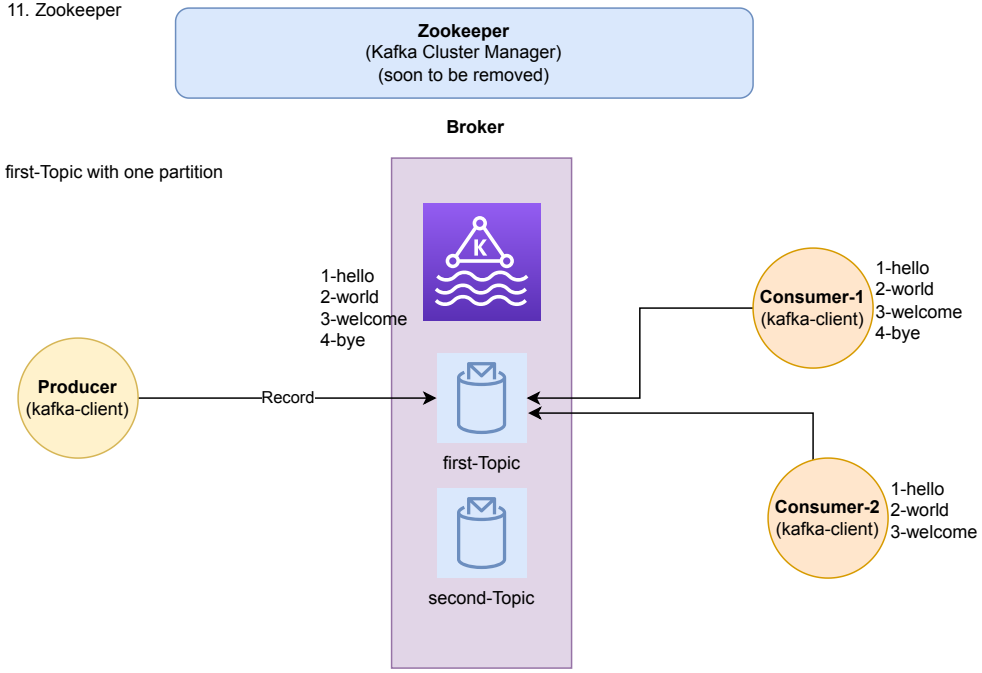
Kafka:

1. Highly Available
2. Scalable
3. Resilient
4. Fault Tolerant
5. DIstributed
6. Low Latency
7. High Throughput

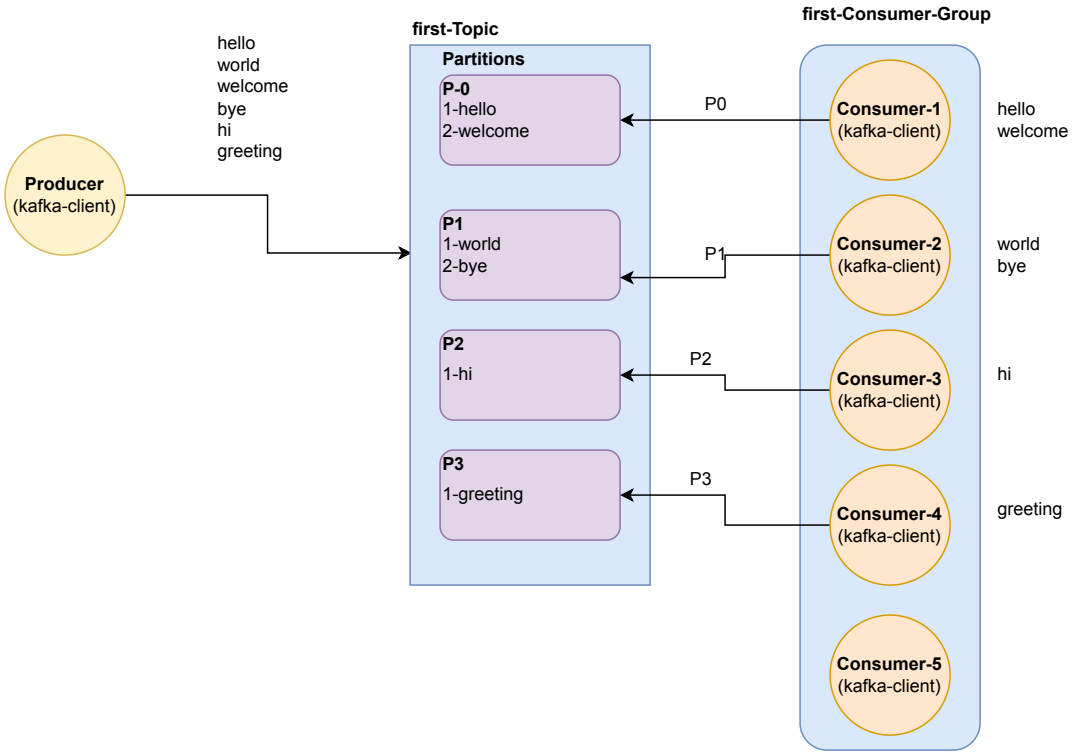


Kafka Terminologies

1. Broker
2. Topic
3. Record (Message)
4. Producer
5. Consumer
6. Consumer Group
7. Kafka Client
8. Partition (Scalable)
9. Replication Factor (Highly Available)
10. Offset
11. Zookeeper

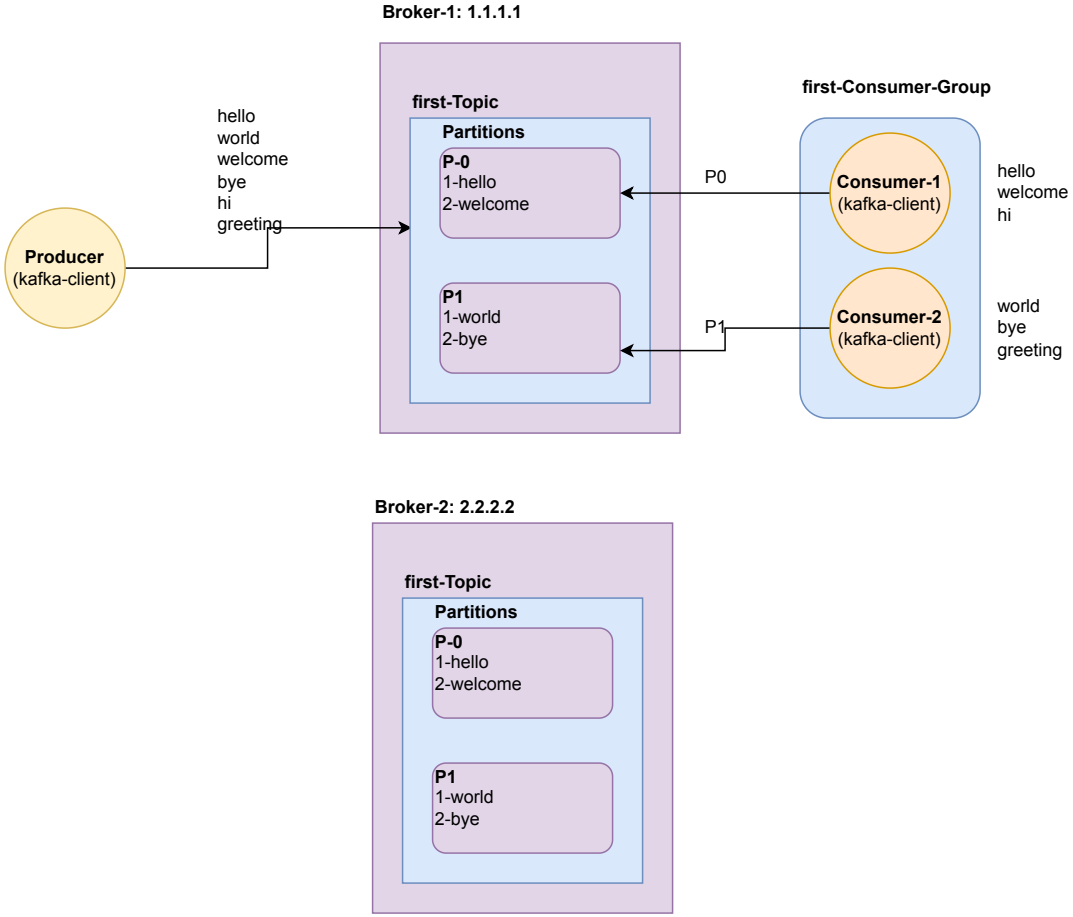


Kafka Partitions
(Scalable)

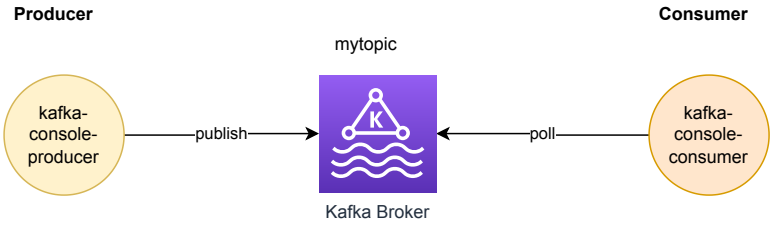


Kafka Replication Factor
(Highly Available)

Replication Factor: 2



Kafka Console Producer/Consumer



Sequential Processing-1

1. order_initiated
2. order_created
3. inventory_initated
4. payment_initiated
5. payment_completed
6. inventory_locked
7. shipment_initiated
8. shipment_completed
9. order_completed

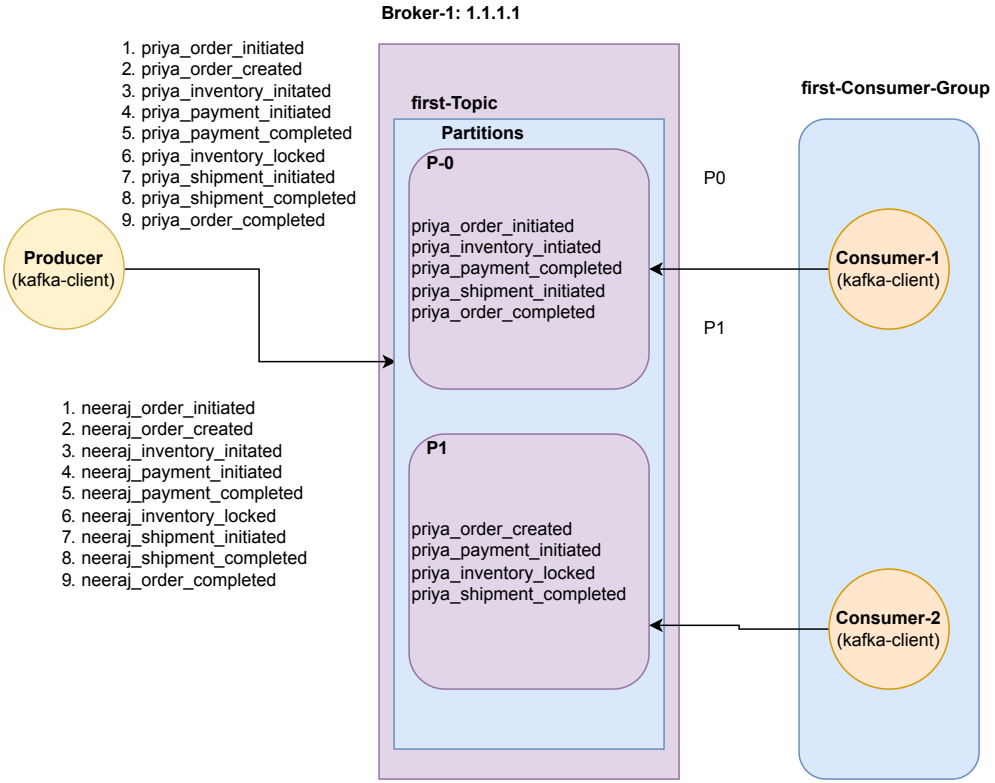
1. order_initiated
2. order_created
3. inventory_initated
4. payment_initiated
5. payment_failed
6. inventory_released
7. order_failed

Priya

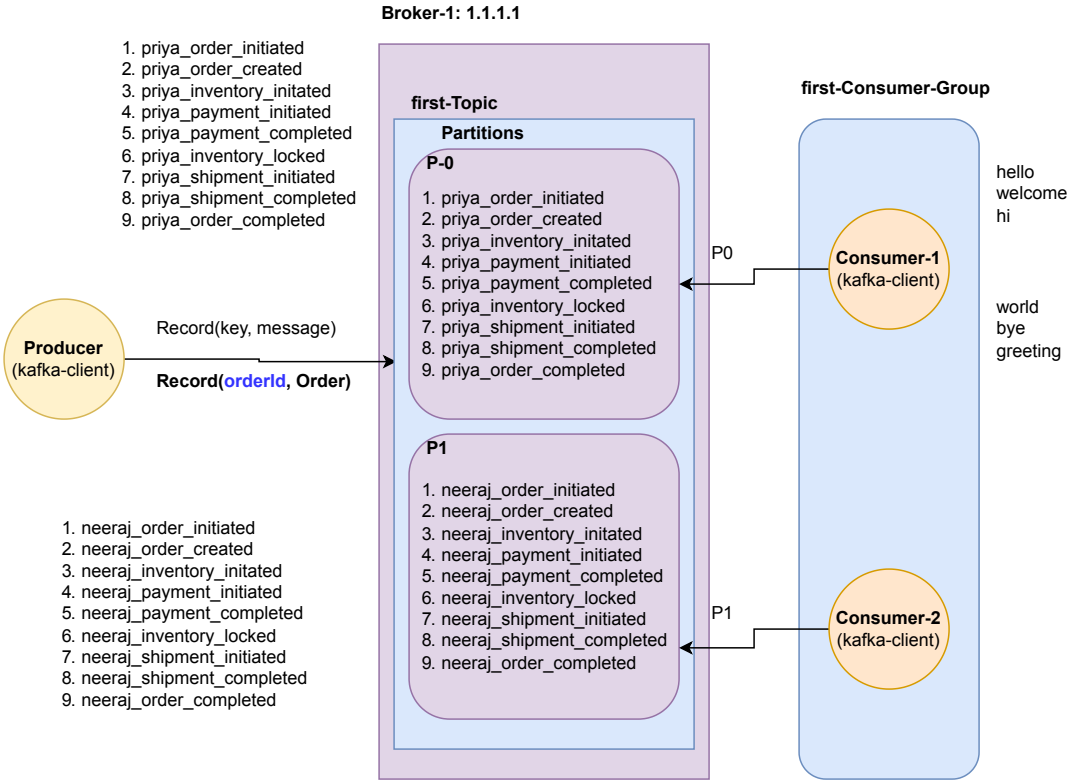
1. priya_order_initiated
2. priya_order_created
3. priya_inventory_initated
4. priya_payment_initiated
5. priya_payment_completed
6. priya_inventory_locked
7. priya_shipment_initiated
8. priya_shipment_completed
9. priya_order_completed

Neeraj

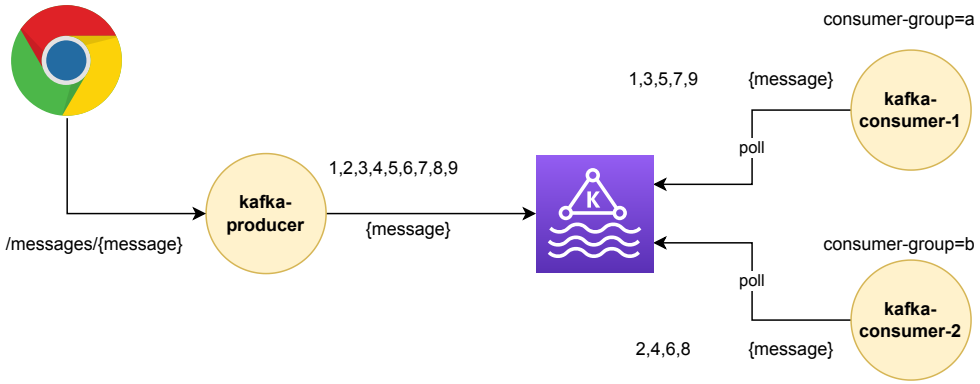
1. neeraj_order_initiated
2. neeraj_order_created
3. neeraj_inventory_initated
4. neeraj_payment_initiated
5. neeraj_payment_completed
6. neeraj_inventory_locked
7. neeraj_shipment_initiated
8. neeraj_shipment_completed
9. neeraj_order_completed



Sequential Processing-2



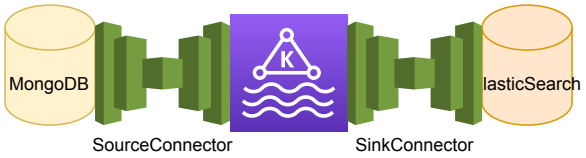
Day-2

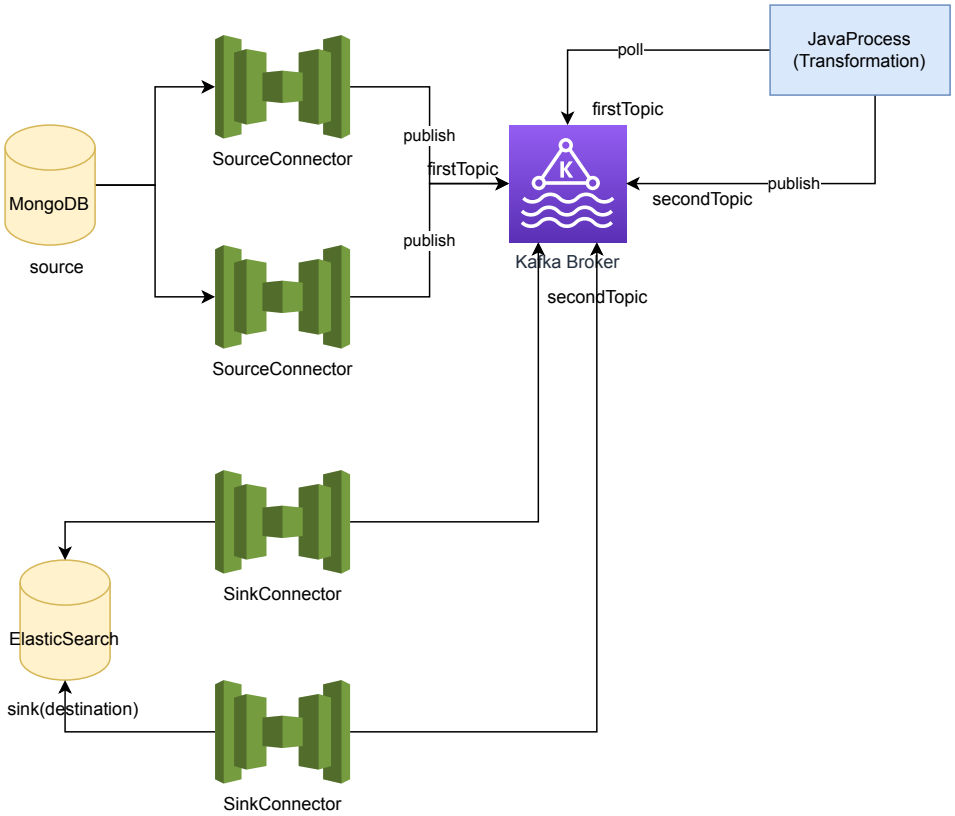

consumer-group=a
1,3,5,7,9
{message}
kafka-consumer-1
poll
1,2,3,4,5,6,7,8,9
kafka-producer
{message}
/messages/{message}
consumer-group=b
poll
kafka-consumer-2
2,4,6,8
{message}

Confluent Kafka Products

1. Kafka Connector
2. Kafka Stream
3. KSQL
4. Kafka GUI
5. Confluent Kafka Cloud (Managed Service)
6. AWS MSK (Managed Streaming for Apache Kafka)

ETL
(Extract Transform Load)

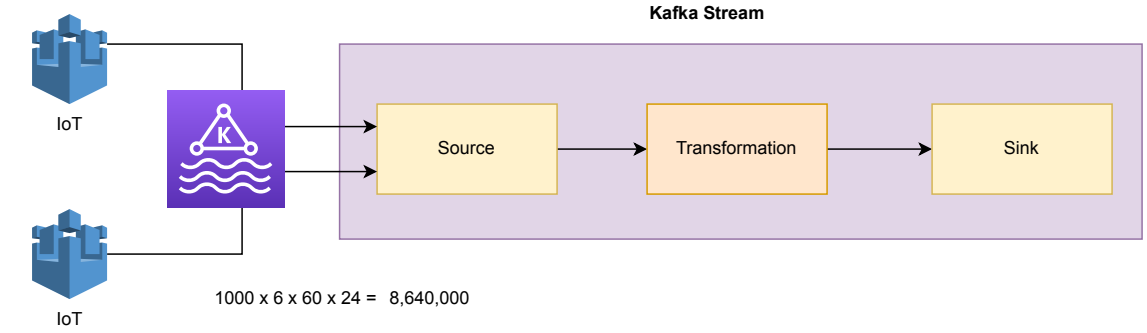




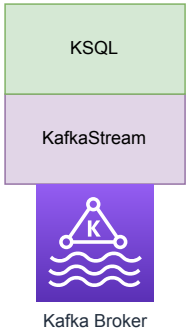
Confluent Kafka Products

Kafka Stream:

DAG(Directed Acyclic Graph)



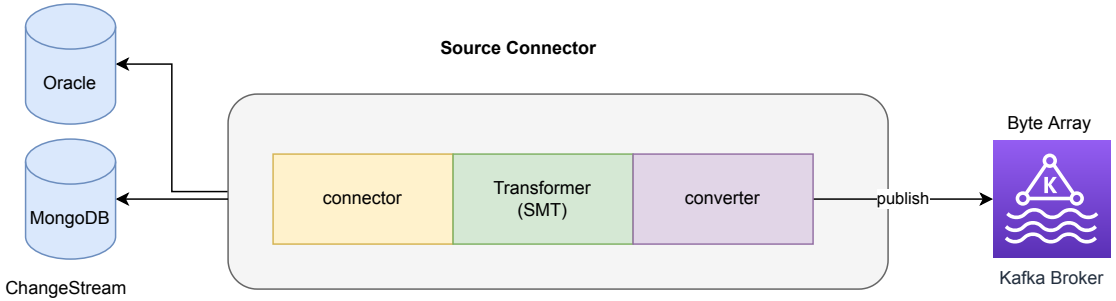
KSQL:



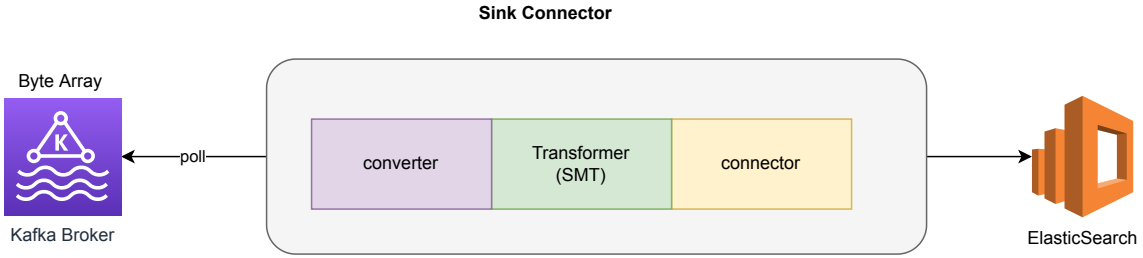
Day-3

Kafka Connect

source

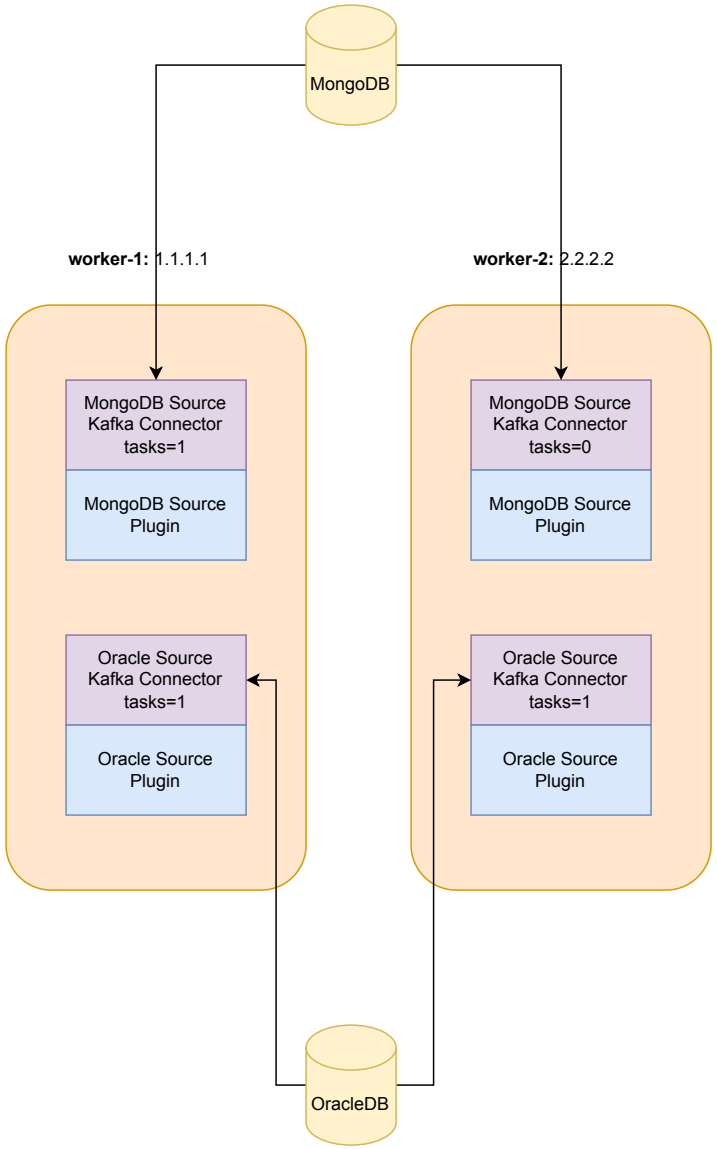


SMT: Single Message Transform



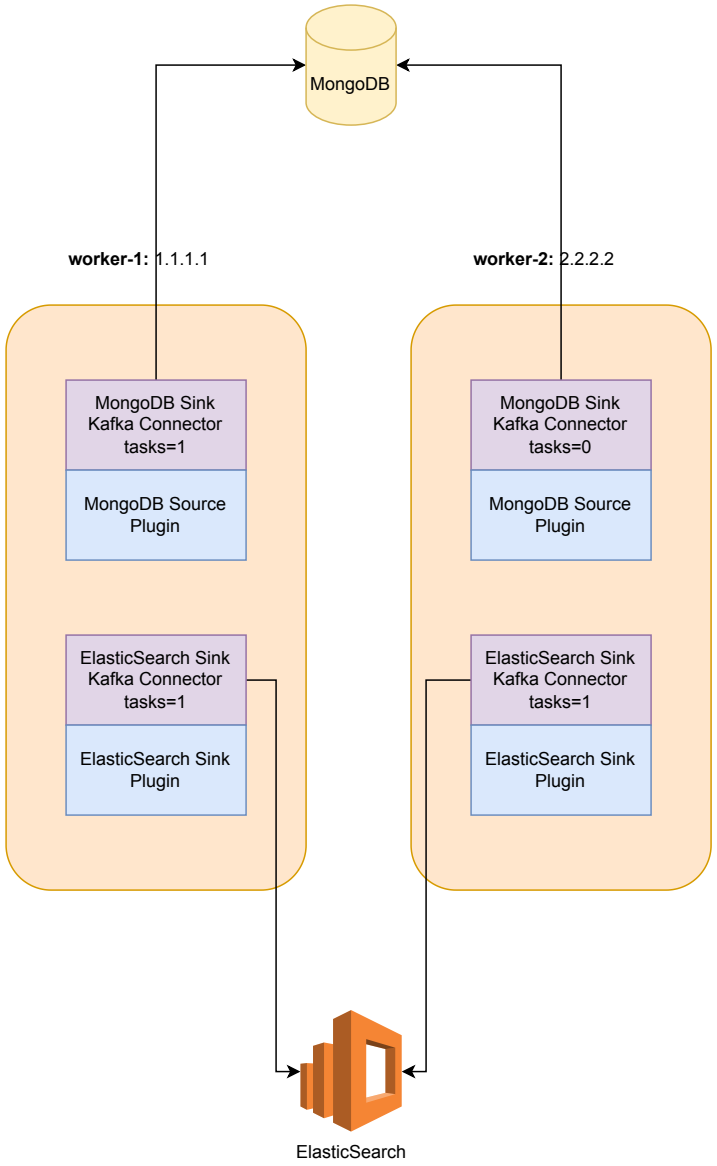
Kafka Connect

Worker and Tasks for Source



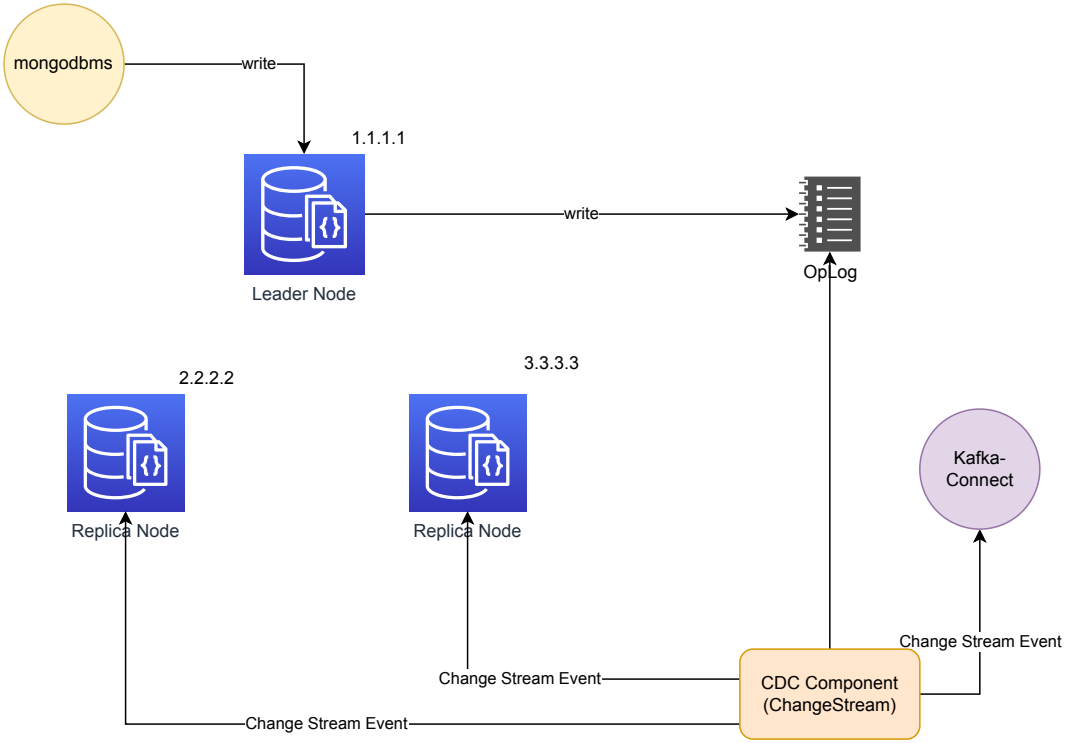
Kafka Connect

Worker and Tasks for Sink

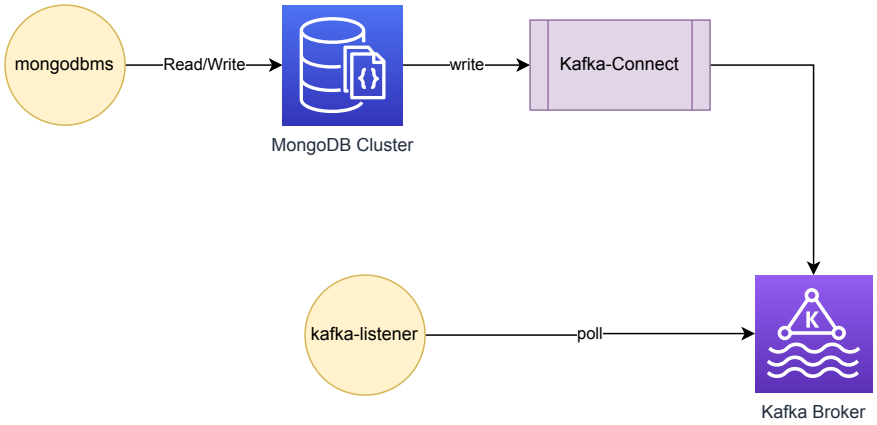


Change Data Capture

(MongoDB Eventual Consistent)



ChangeStreamEvent Listener



Relational DB vs MongoDB

1. Database
2. Table
3. Columns
4. Rows

1. Database
2. Collections
3. Fields
4. Documents

Relational DB:
Database: userdb
Table: user_data

userId, userName, age
1 John 21
2 Mary 24

NoSQL: MongoDB
Database: userdb
Collection: user_data

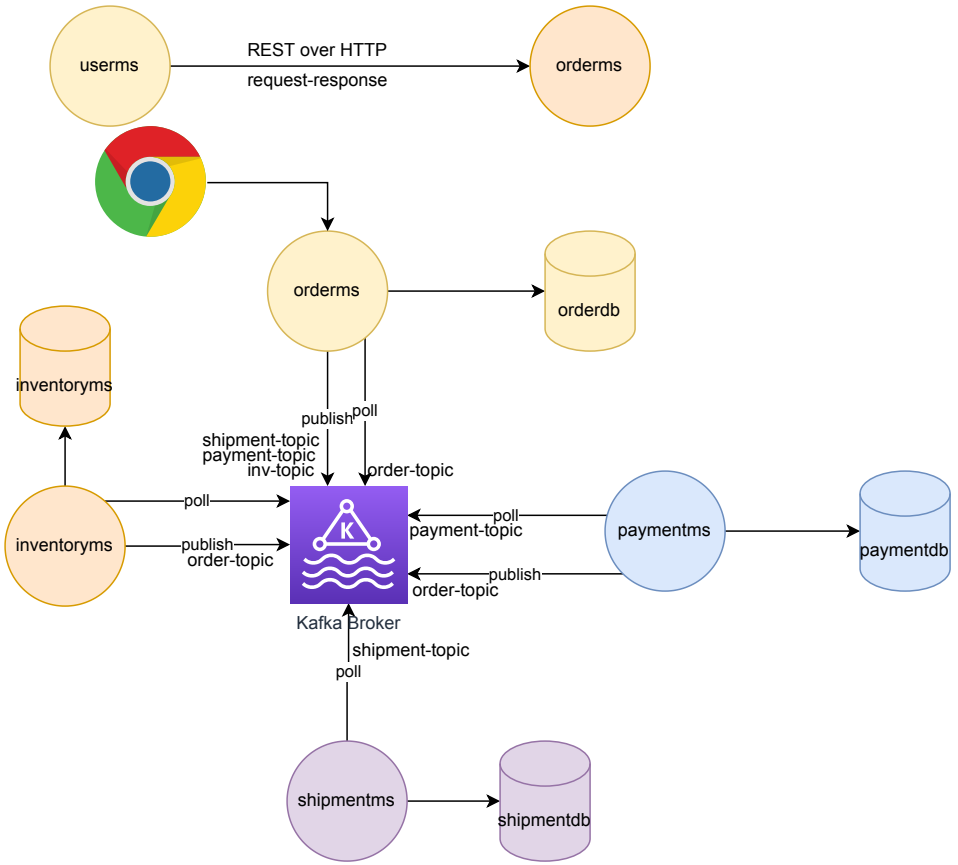
```
{
   "_id": "unique-1",
   "userName": "John",
    "age": 21
}

{
   "_id": "unique-2",
   "userName": "Mary",
    "age": 24
}
```

MongoDB Tools

MongoDB Atlas: Cloud based MongoDB cluster
MongoDB Client: Compass

EDA/MDA

Event Driven Archtecture / Message Driven Architecture



EDA/MDA

Event Driven Archtecture / Message Driven Architecture