

## Course Outline: Table of Contents / Agenda

- 1 What is Kafka & Why We Need It?
- 2 Kafka Components & Internal Architecture
- 3 Different Ways to Install Kafka
  - Binary Distribution
  - Run using Docker Compose

## Course Outline: Table of Contents / Agenda

- 4 Play With Kafka using CLI - Hands-on Example
- 5 Create Producer Application using Kafka & Spring Boot
- 6 Create Consumer Application using Kafka & Spring Boot
- 7 Understand Pub-Sub Mechanism & Internal Workflow using Producer & Consumer Application

## Course Outline: Table of Contents / Agenda

- 8 Apache Kafka Object Serialize & Deserialize Example
- 9 Understand Message Routing with Specific Partitions in Kafka
- 10 Kafka End-to-End Integration Testing in Spring Boot with TestContainers
- 11 Kafka Error Handling with Spring Boot: Retry Strategies & Dead Letter Topics Detailed Explanation

## Course Outline: Table of Contents / Agenda

- 12 Kafka Schema Registry & Avro with Practical Example and Implementation

## Agenda :

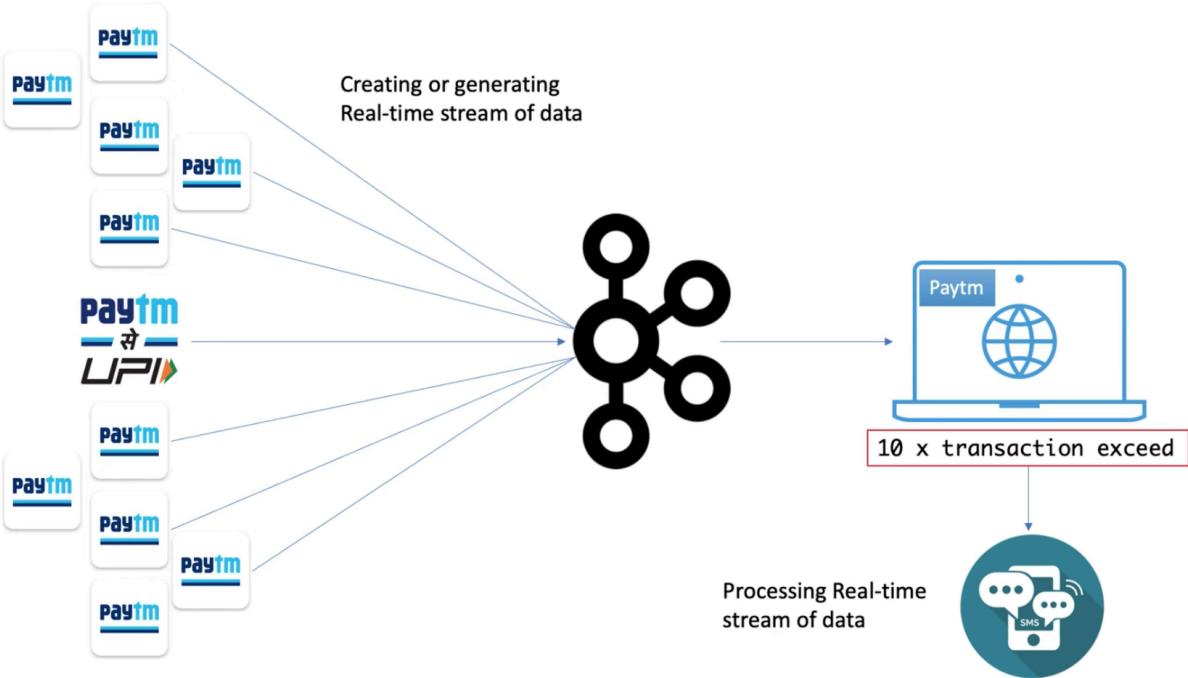
- What is Kafka ?
- Where does Kafka come from ?
- Why do we need Kafka ?
- How does it work (High-level overview)

## What is Kafka ?

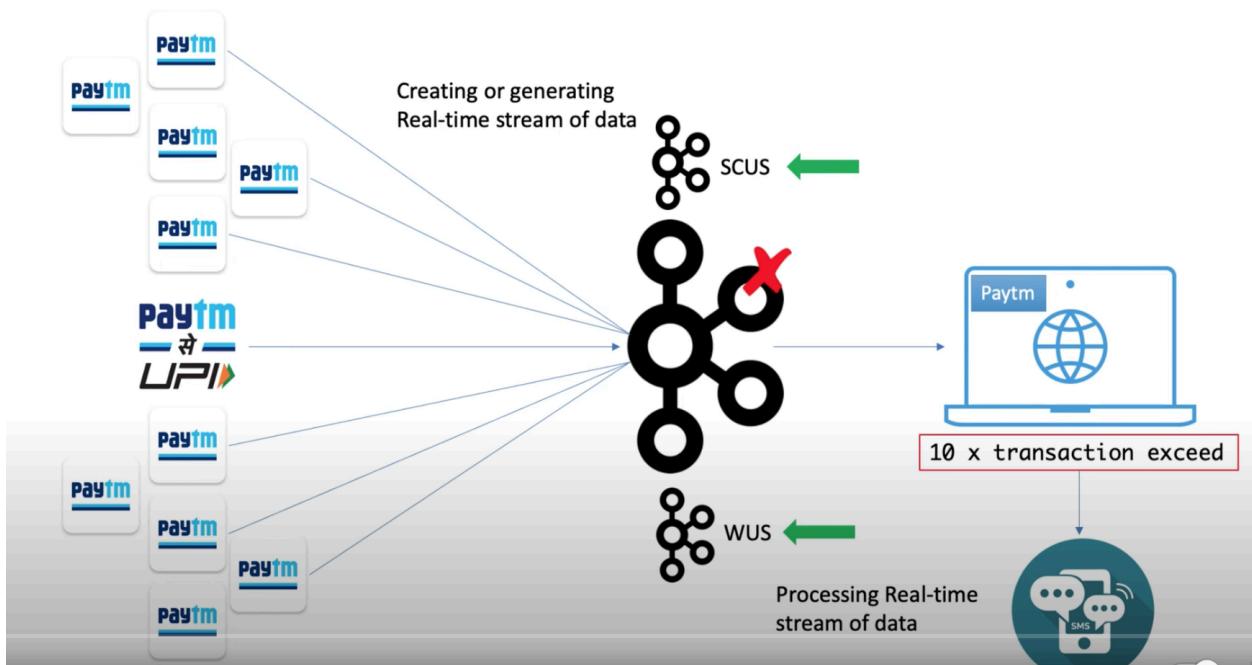
- Apache Kafka is an open-source distributed event streaming platform

Creating Real-time Stream

Processing Real-time Stream



Doing payment is create real-time stream of data  
Application processing them is called processing Real-time Stream of data.



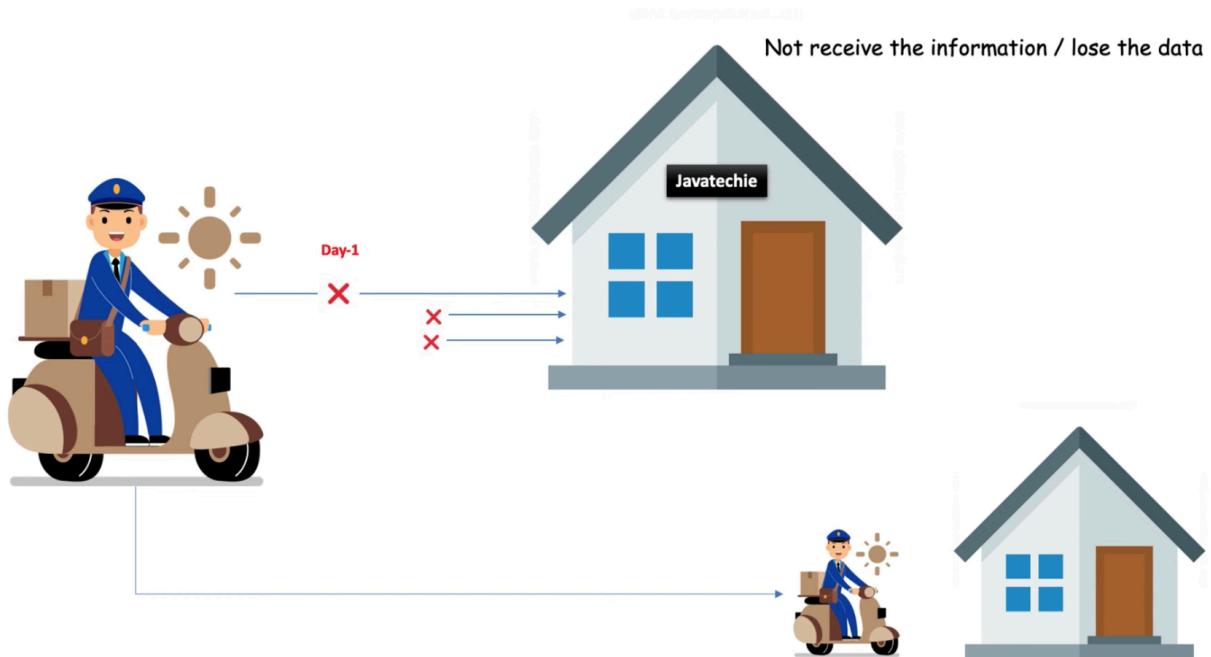
Avoids downtime of any server.

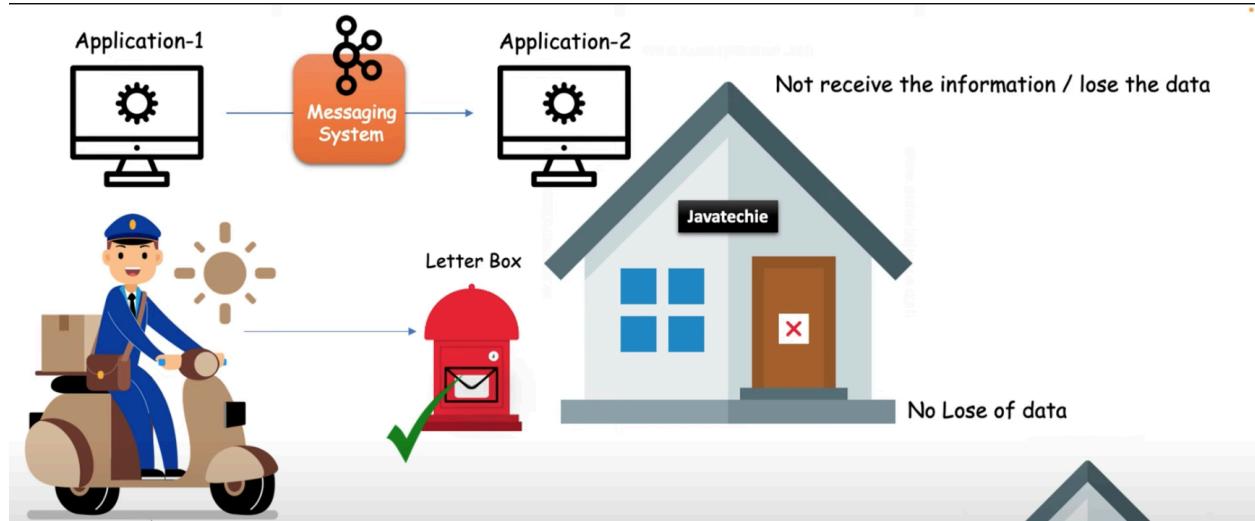
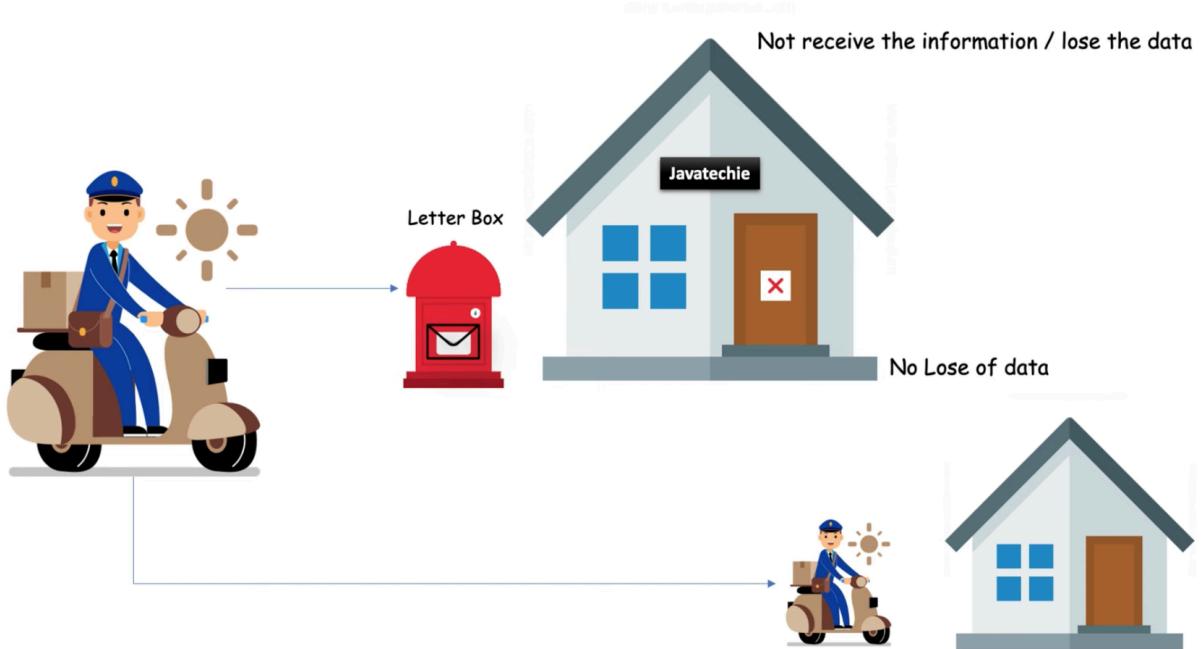
## Where does Kafka come from ?

Kafka was originally developed at [in](#) , and was subsequently open sourced in early **2011**



- Why do we need Kafka ?

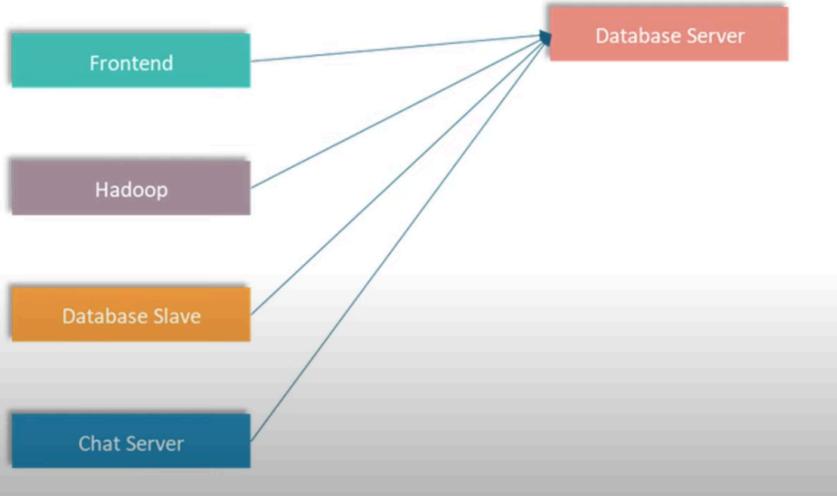




Kafka behaves like a letterbox between application 1 and application 2.

Below is complex scenario why we need kafak

## Why Kafka ?

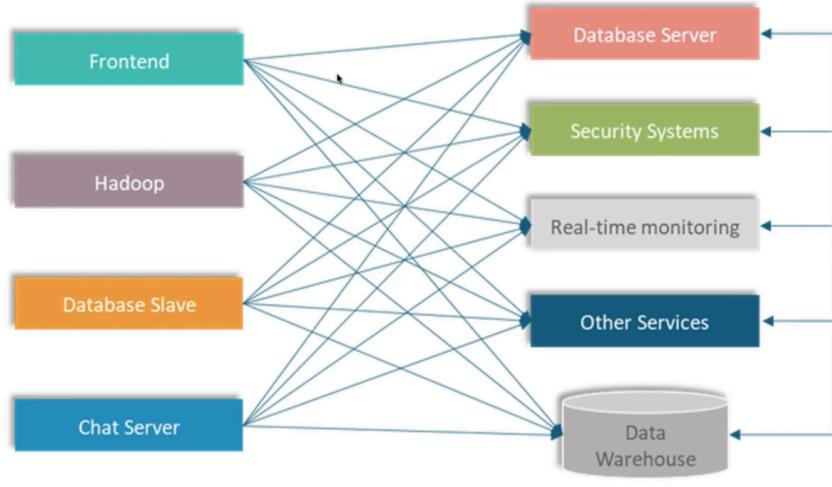


Data Format

Connection Type

Number of connection

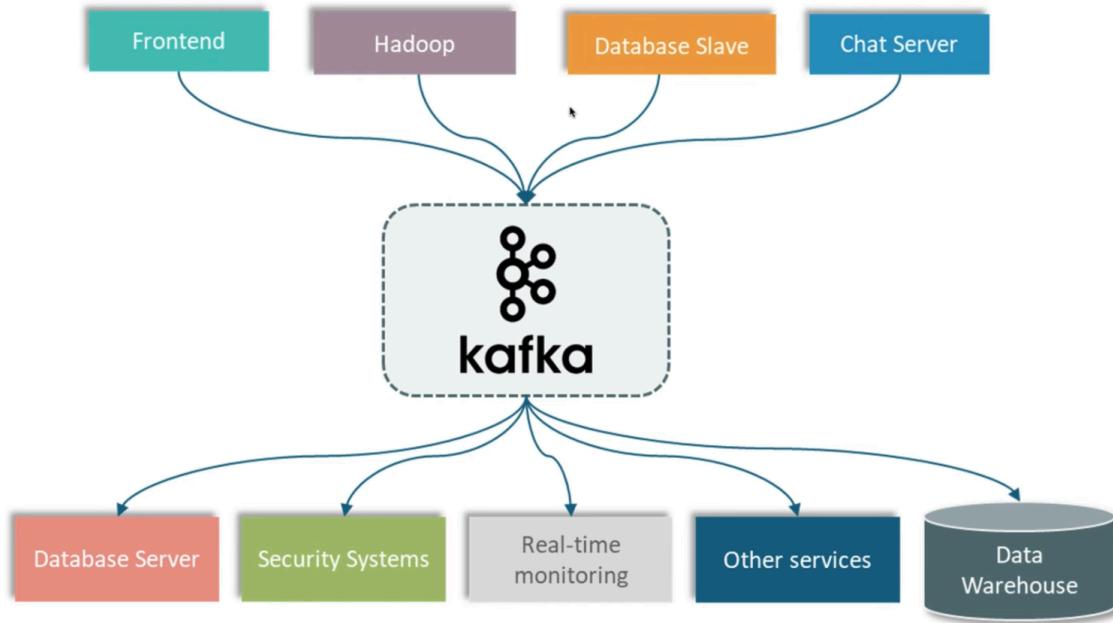
## Why Kafka ?



Every application has to maintain lot of connections to communicate with other servers / applications / systems

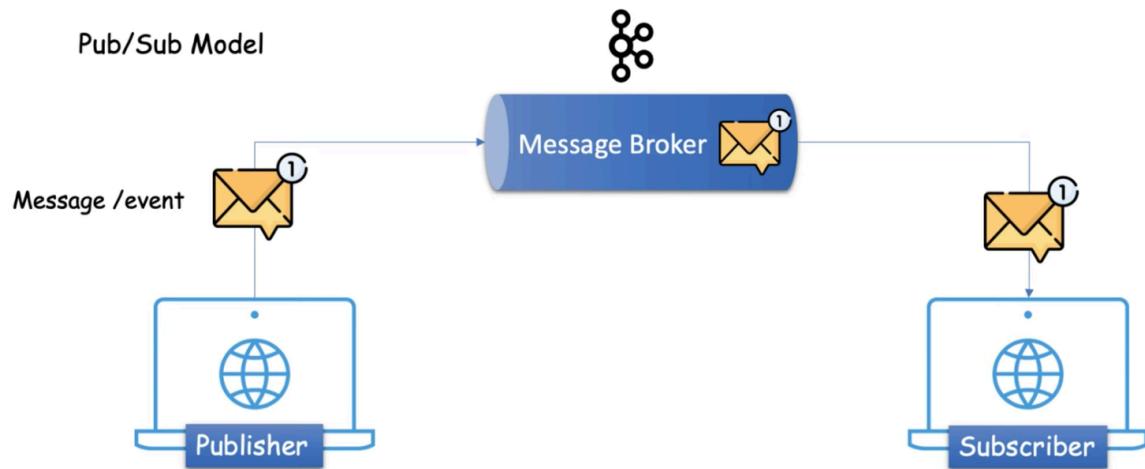
- Data format can be json / xml => each target system may expect different format of data
- Connection types can be http, tcp, JDBC => source systems have to maintain all the connections based on target system expectation.
- No. Of connections. => Source has to maintain all connections. Example in the above case every source system maintaining 5 connections.

For communication of 9 application we are maintaining  $4 \times 5 = 20$  connections.

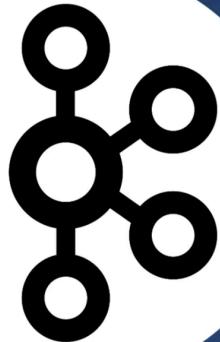


Here Kafka behaves like a letterbox. Here the number of connections are  $5 + 4 = 9$ .

## How does it work (High-level overview)



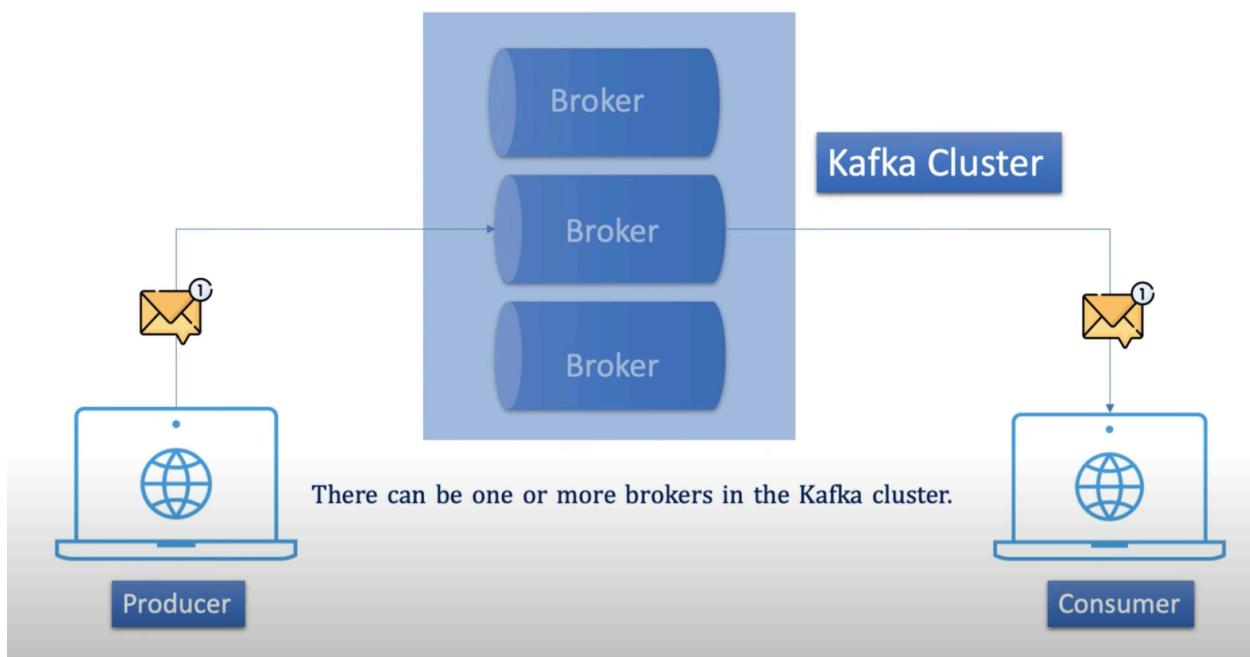
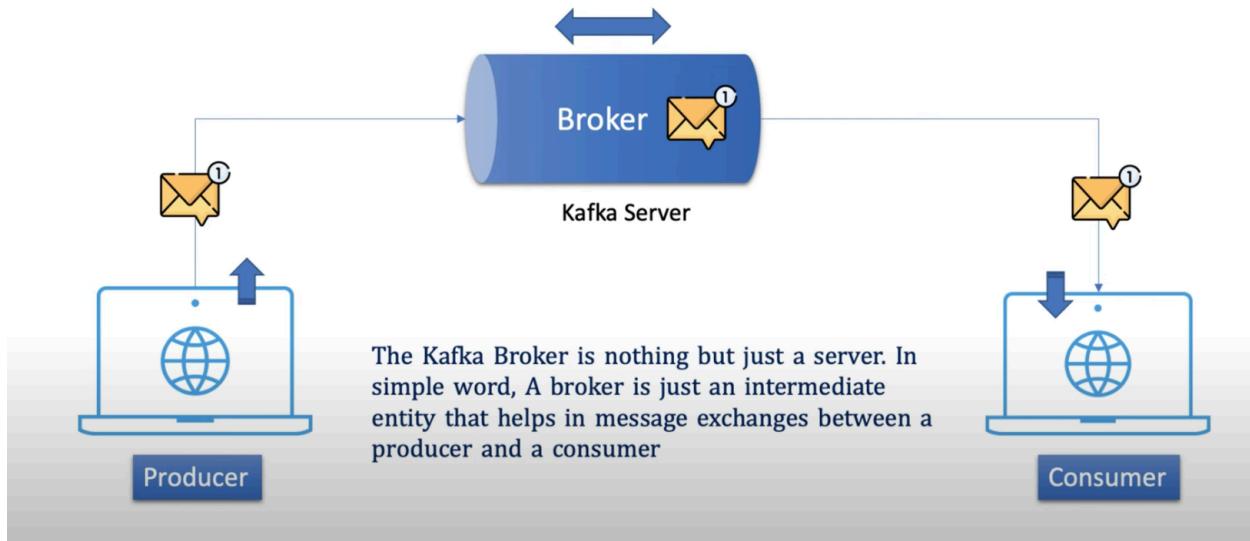
## Kafka Architecture & Components



## Kafka Components

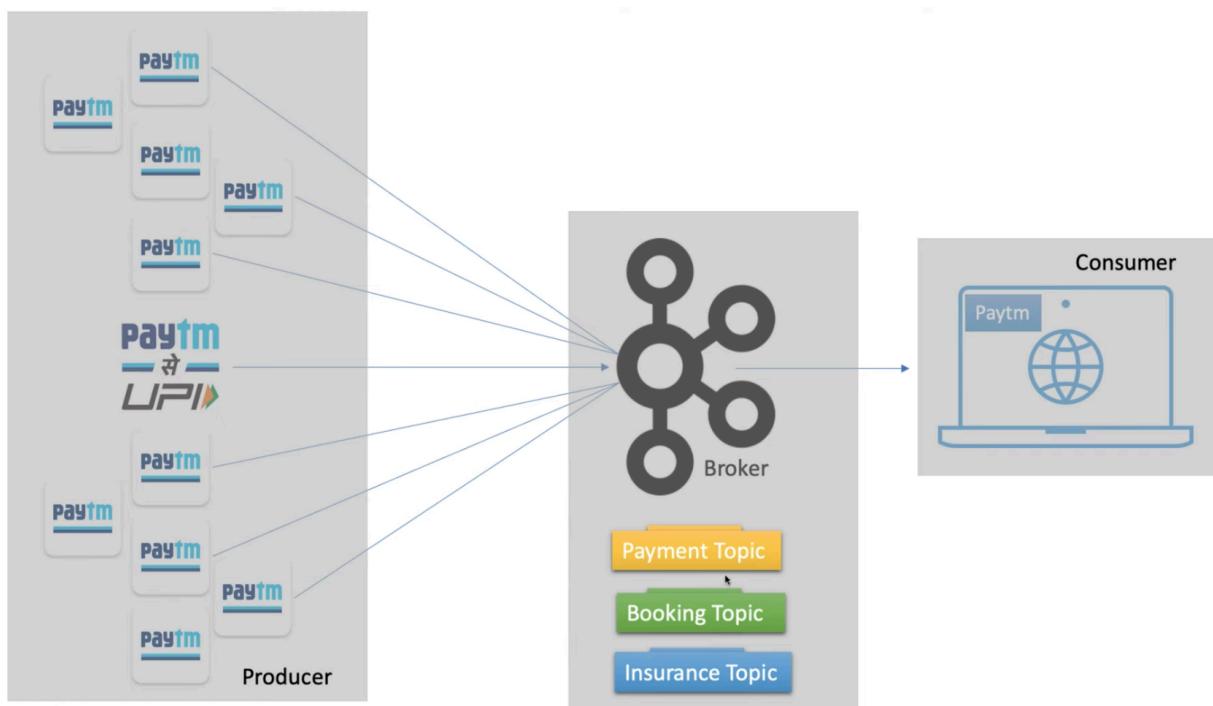
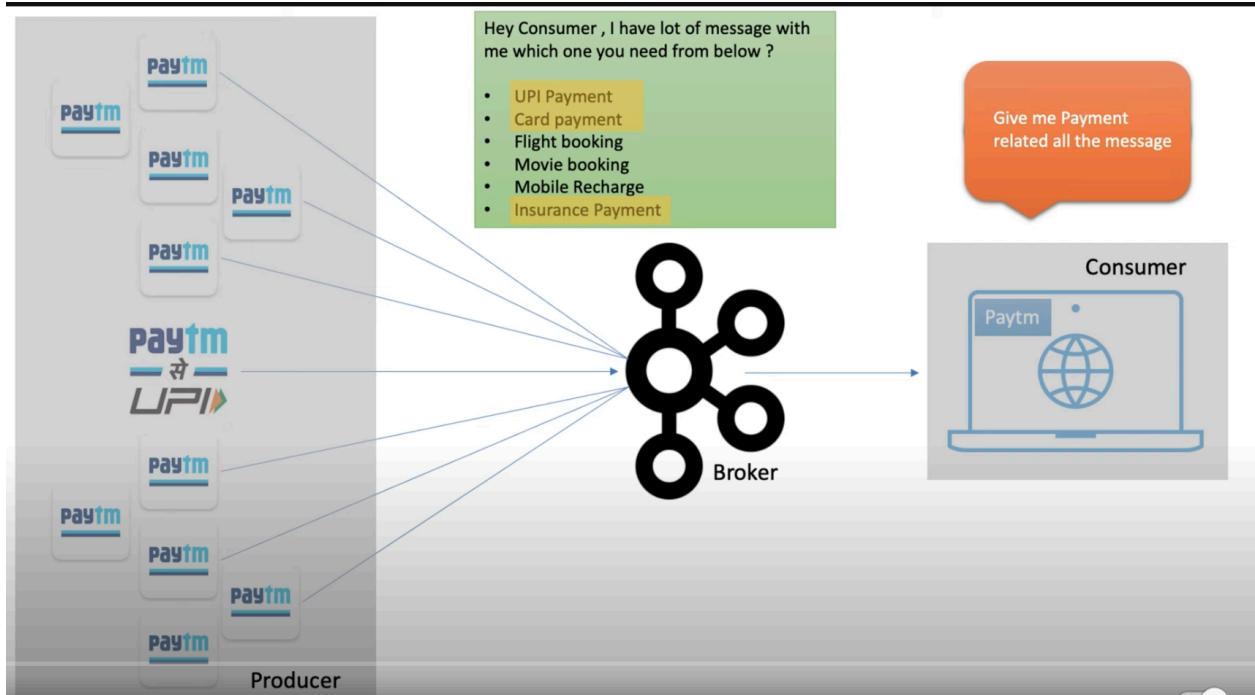
- Producer
- Consumer
- Broker
- Cluster
- Topic
- Partitions
- Offset
- Consumer Groups
- Zookeeper





Cluster means a group of computers in a distributed system work for a common goal.

Topic: what topic consumer is interested in.

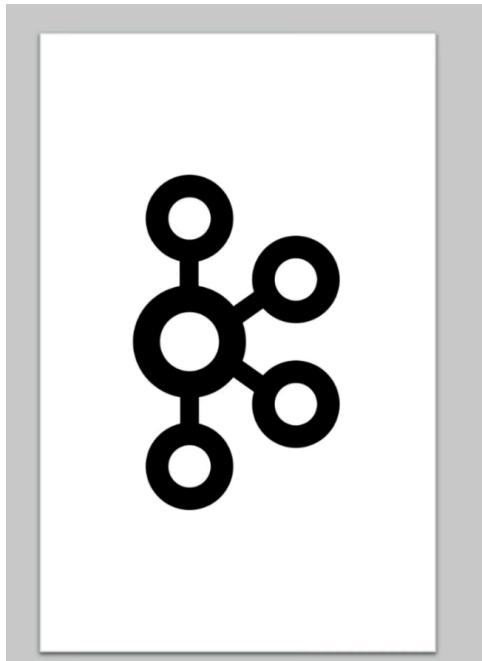
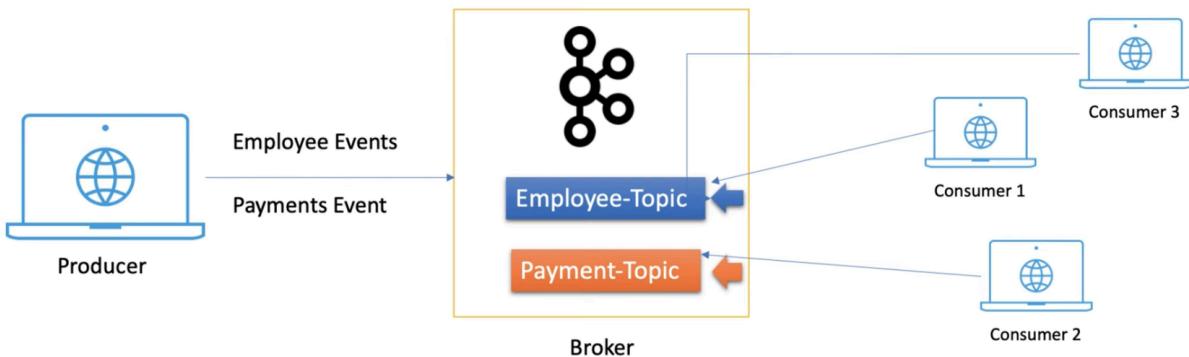


EMPLOYEE\_TABLE

ID	NAME	DEPT	SALARY
1	john	IT	50000
2	sam	Bank	80000
3	joe	Admin	20000

PAYMENT\_TABLE

PAYMENT_ID	CNAME	PAY MODE	SRC AC	DEST AC	AMOUNT
df364bdjfb	Basant	CREDIT	HDFC12	SBI239837	100000
bcsr72386d	Santosh	DEBIT	ICICI893	HDFC3626	890012
dfv834tjsd8	Krishna	CHEQUE	SBI2367	IDBI23726	23273582

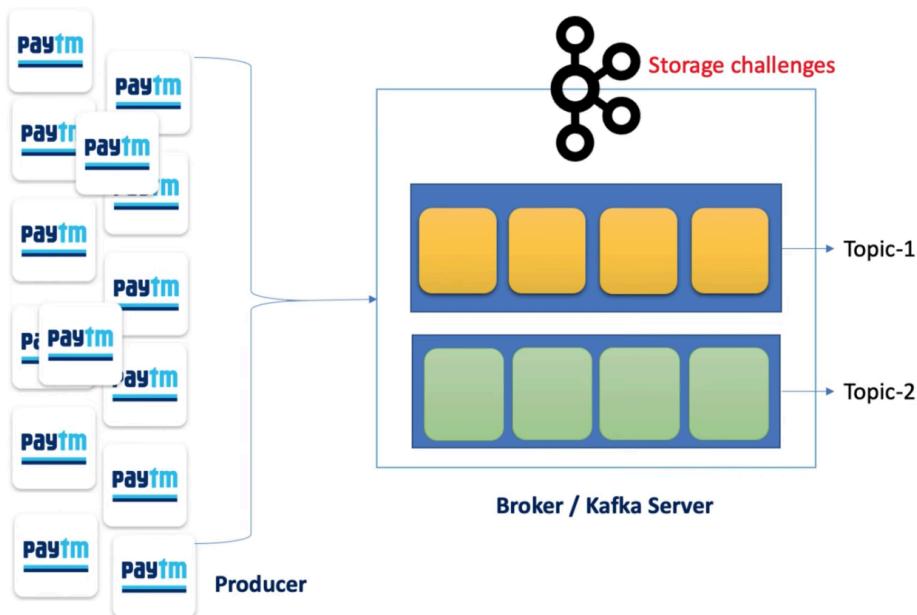


TOPIC

It specifies the category of the message or the classification of the message. Listeners can then just respond to the messages that belong to the topics they are listening on.

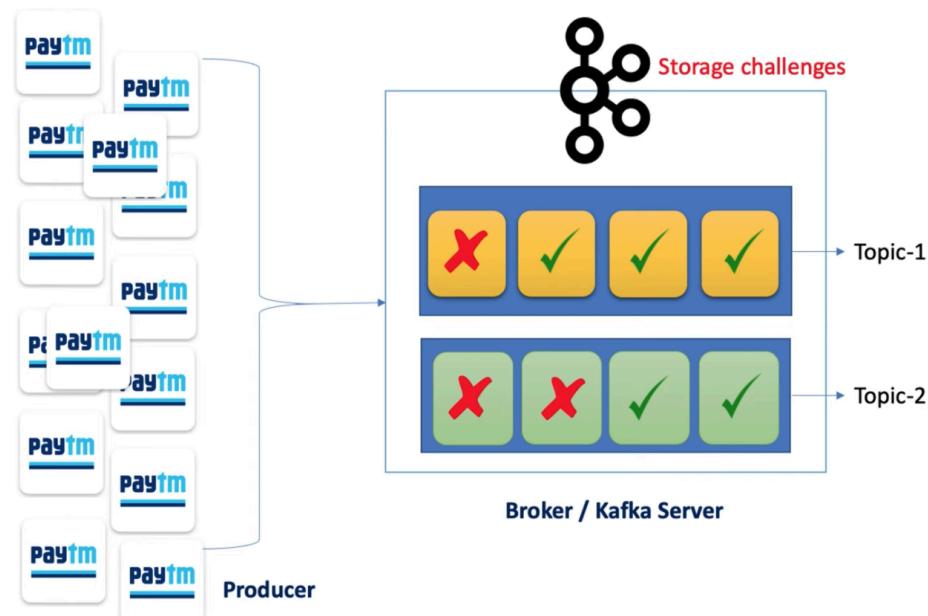
Autoplay is on

## Partitions



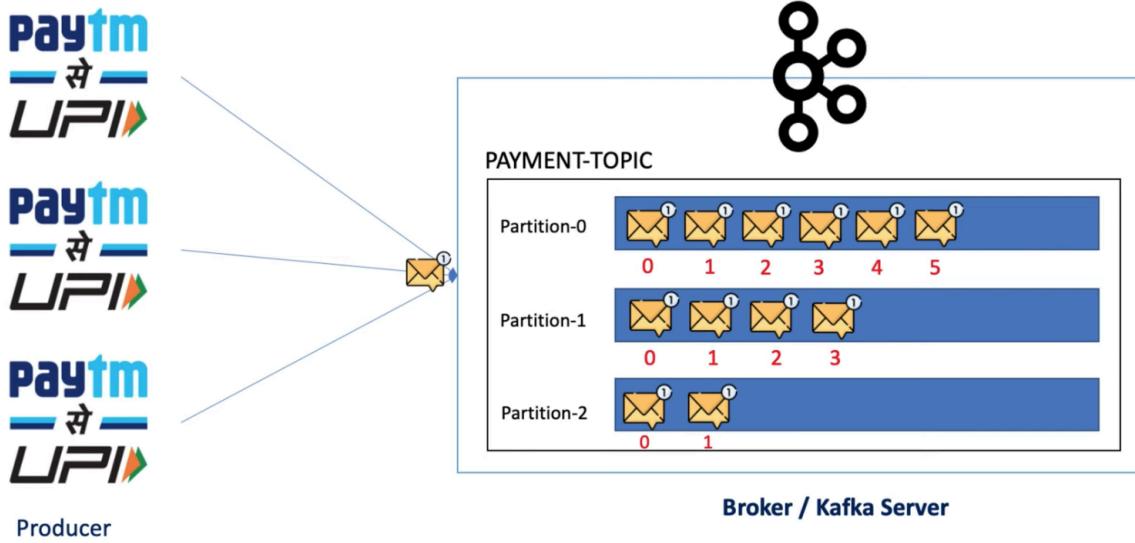
Each partition concurrently accept the message. Even any partition goes off then other partitions are available to handle load without application downtime.

## Partitions

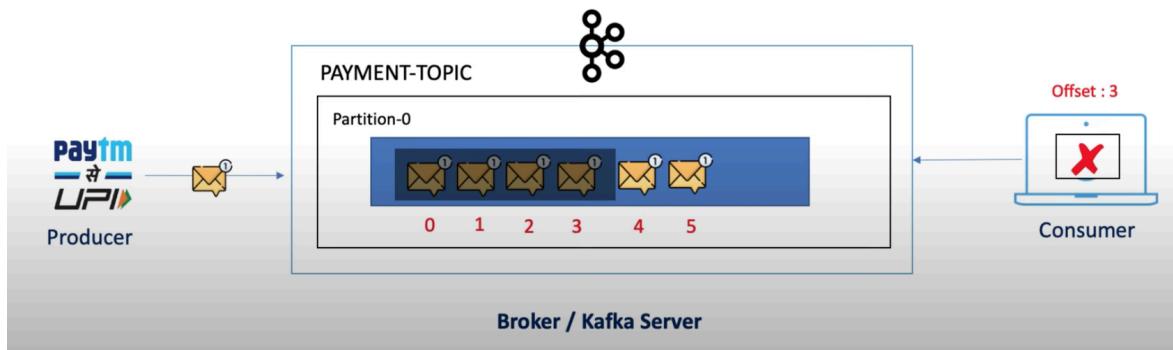


What is offset.:

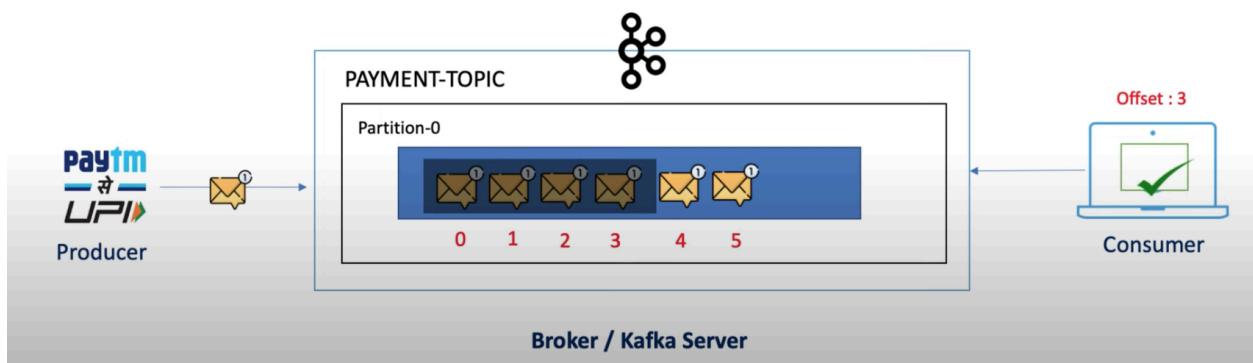
In Kafka, a sequence number is assigned to each message in each partition of a Kafka topic. This sequence number is called Offset.



If a consumer went down and came back after some time as shown below.

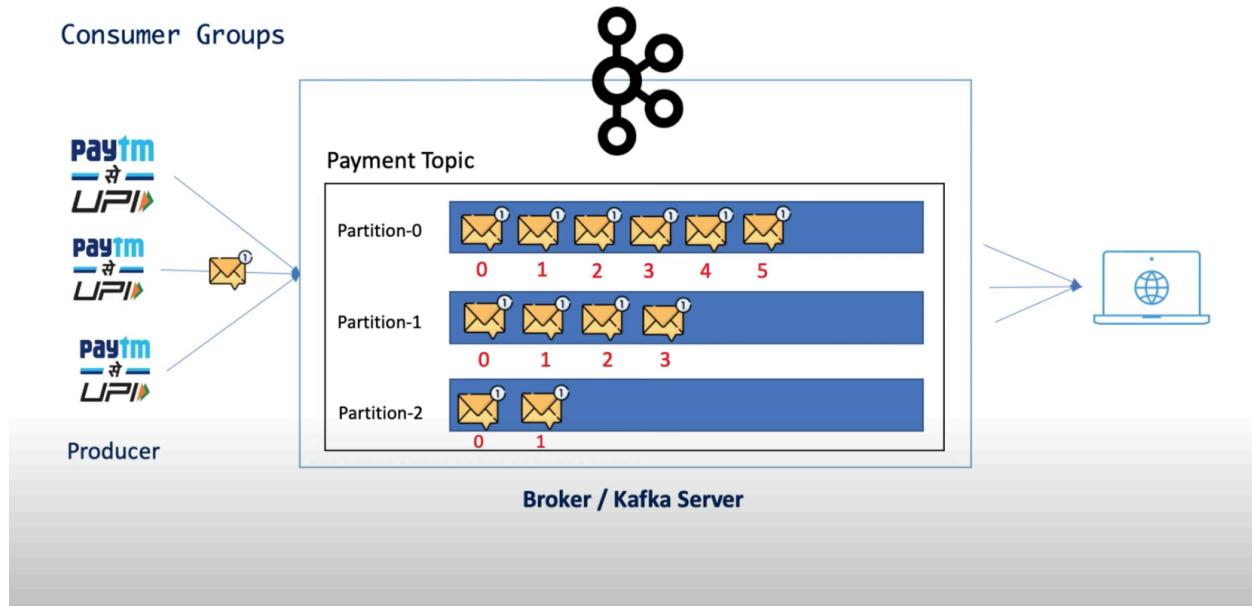


When consumer recovered, offset helps consumer from which offset it should start to read.

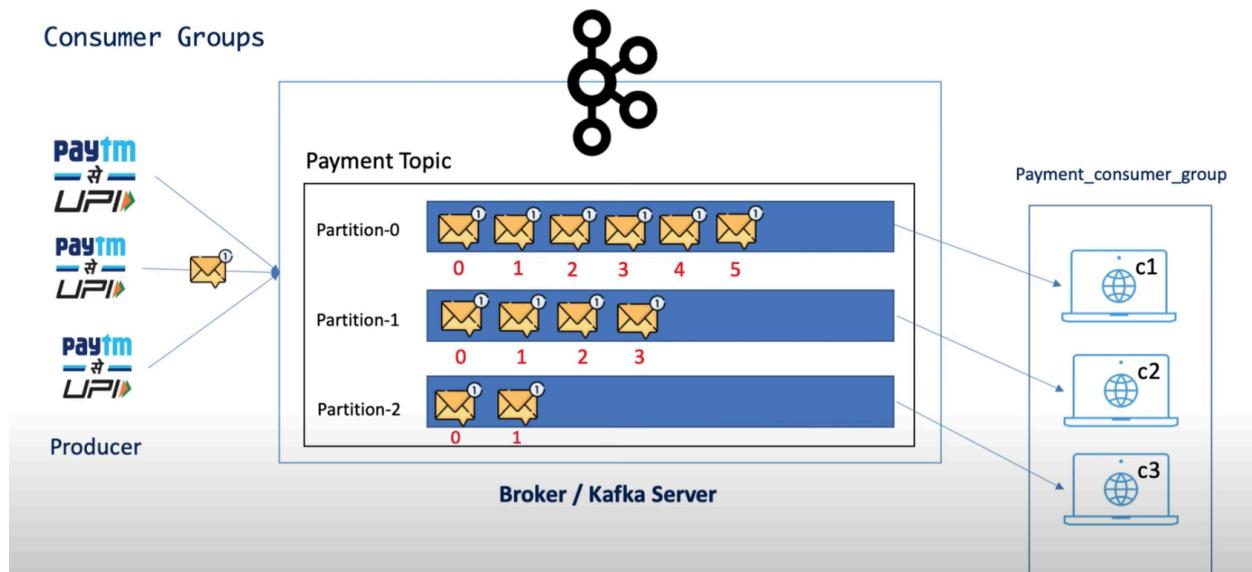


## Need of consumer group

In the below we have only one consumer instance so there is chance of performance issue.

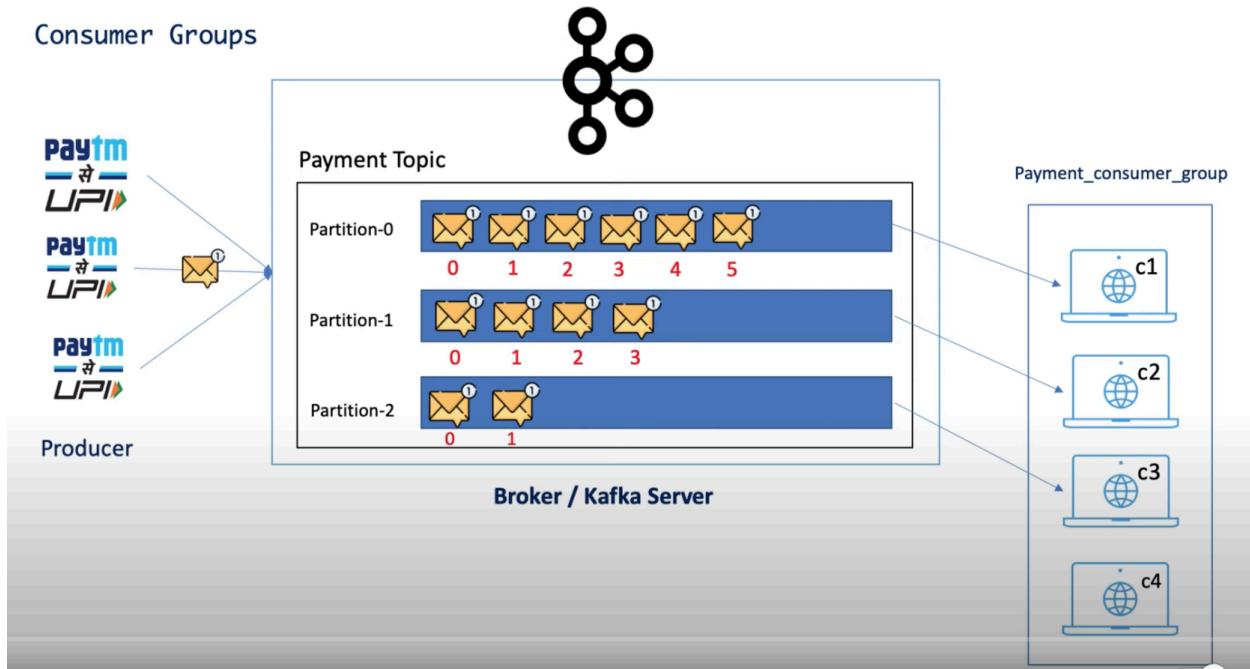


Let us create 3 instances and so each instance read messages from each topic as shown below and group those instances using consumer group. Payment consumer group is example.



Here for simplity we draw order in sequence. But in real time this order my differ (partition 0 will be listen by c3 , etc.

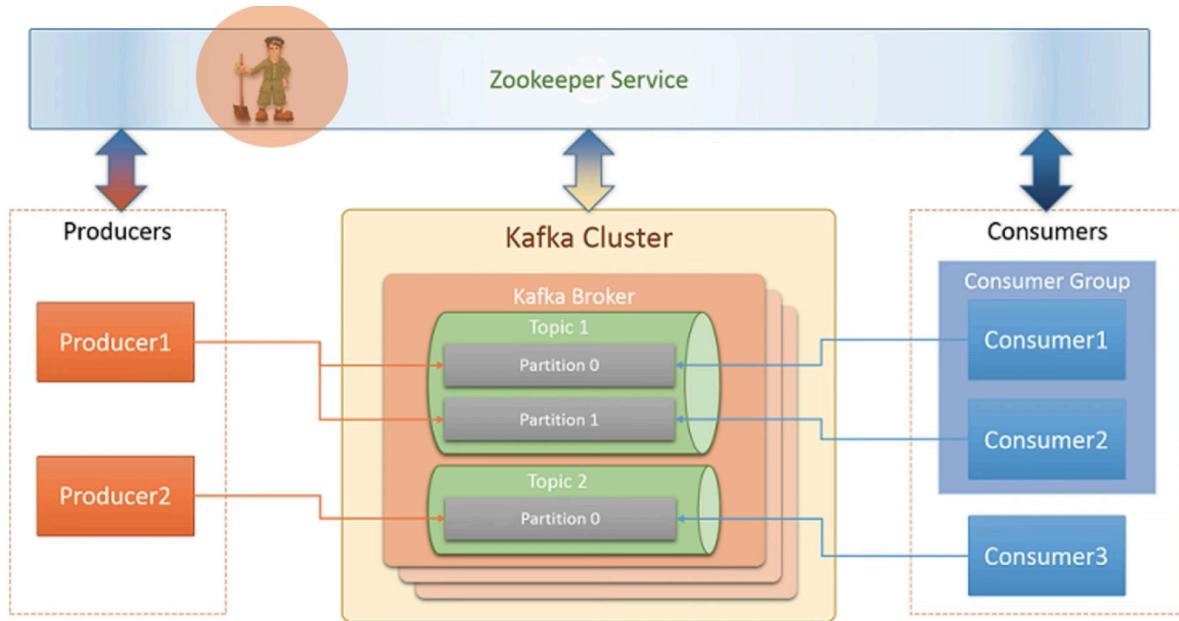
Assume that we have consumer 4 as shown below. It will sit idle as there is no partition available for it to read. If any one consumer goes down then c4 will start consuming those messages. It is called - consumer rebalancing.



What is zoo keeper ? - manager for Kafka

- Partitions
- Offset
- Consumer Groups
- Zookeeper

Zookeeper is a prerequisite for Kafka. Kafka is a distributed system, and it uses Zookeeper for coordination and to track the status of Kafka cluster nodes. It also keeps track of Kafka topics, partitions, offsets, etc.



## Kafka Installation

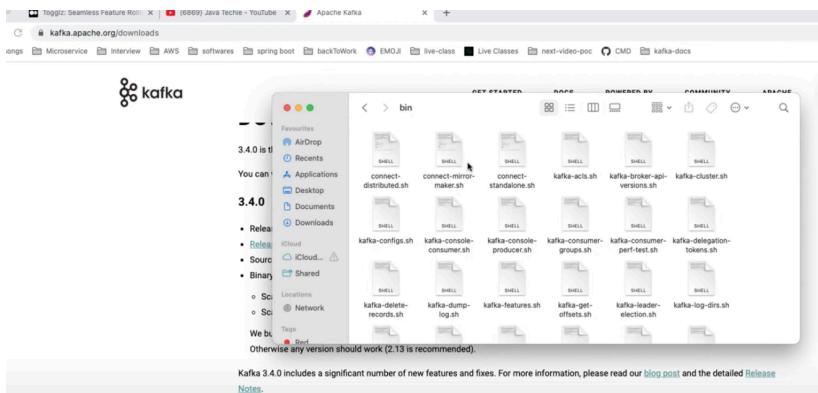
Open Source : Apache Kafka

Commercial distribution : Confluent Kafka

Managed Kafka service : confluent & AWS

The screenshot shows the Apache Kafka download page. At the top, it says "3.4.0 is the latest release. The current stable version is 3.4.0". Below that, it says "You can verify your download by following these [procedures](#) and using these [KEYS](#)". Under the heading "3.4.0", there is a bulleted list of details:

- Released Feb 7, 2023
- [Release Notes](#)
- Source download: [kafka-3.4.0-src.tgz](#) ([asc](#), [sha512](#))
- Binary downloads:
  - Scala 2.12 - [kafka\\_2.12-3.4.0.tgz](#) ([asc](#), [sha512](#))
  - Scala 2.13 - [kafka\\_2.13-3.4.0.tgz](#) ([asc](#), [sha512](#))



All sh files will be available under bin folder.

All bat files will be available under bin/windows folder.

Using these files we will control all the below kafka components.

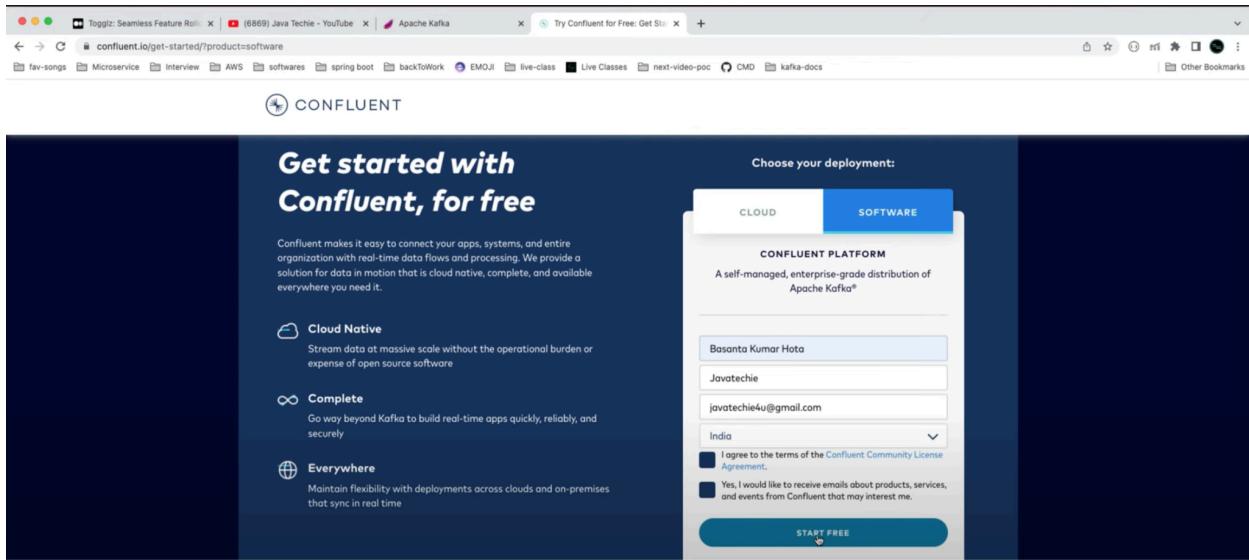
## Kafka Components

- Producer
- Consumer
- Broker
- Cluster
- Topic
- Partitions
- Offset
- Consumer Groups
- Zookeeper

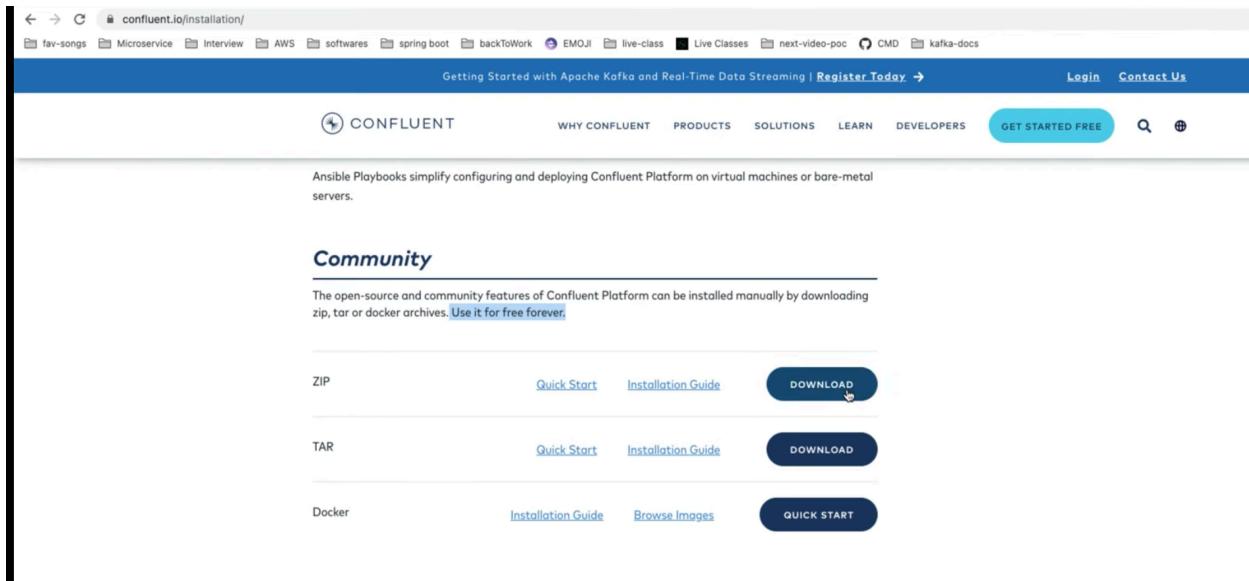


Inside the config folder server.properties and zookeeper.properties file can be found. They contain kafka and zookeeper specifications.

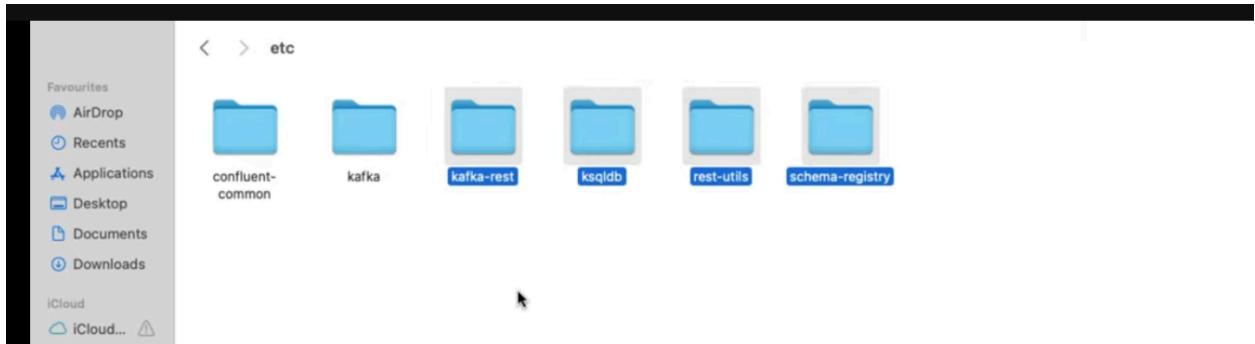
## Confluent Kafka



Download Confluent Kafka community edition to play around



We have lot of utility under confluent kafka



## Kafka offset explorer

← → ⌘ [kafkatool.com/download.html](http://kafkatool.com/download.html)

fav-songs Microservice Interview AWS softwares spring boot backToWork EMOJI live-class Live Classes next-video-poc CMD kafka-docs Other Bookmarks

### Offset Explorer 2.3.2

[For Kafka version 0.11 and later]

Platform	Size	Action
Windows 32-Bit	35 MB	<a href="#">Download</a>
Windows 64-Bit	34 MB	<a href="#">Download</a>
macOS	55 MB	<a href="#">Download</a>
Linux	16 MB	<a href="#">Download</a>