# Kafka Fundamentals: Complete Developer Guide

Apache Kafka is a distributed event streaming platform that allows real-time data pipelines and streaming applications.   
It helps in transmitting data between systems efficiently and reliably.

## 1️⃣ What is Kafka?

Kafka is a distributed event streaming platform designed to handle real-time data feeds.   
It enables producers to send data and consumers to receive it asynchronously.  
Example: When a parcel is scanned at FedEx, Kafka sends that event to tracking microservices for real-time updates.

## 2️⃣ Main Components of Kafka

a) Producer – Sends data to Kafka (e.g., Scanner Service).  
b) Consumer – Reads data from Kafka (e.g., Tracking Service).  
c) Topic – Named stream or category of messages (similar to a table in a database).

## 3️⃣ Kafka Partitions

Each Kafka topic is divided into partitions. Each partition stores data sequentially.  
This helps with scalability and parallel processing.   
More partitions = more consumers can process data in parallel.

## 4️⃣ Role of Zookeeper

Zookeeper acts as the manager of the Kafka cluster.   
It handles broker coordination, cluster metadata, leader election, and configuration synchronization.  
Without Zookeeper, Kafka wouldn't know which broker or partition leader is active.  
Note: Modern Kafka (KRaft mode) can work without Zookeeper.

## 5️⃣ Kafka End-to-End Flow Example

Example: Order Processing System  
  
1. User places an order.  
2. Order Service (Producer) sends an event to 'order-events' topic.  
3. Kafka stores the event in the appropriate partition.  
4. Inventory Service and Billing Service (Consumers) read from the same topic.  
5. Inventory updates and billing is processed in real-time.

## 6️⃣ Kafka Key Terms Summary

|  |  |  |
| --- | --- | --- |
| Term | Description | Example |
| Broker | Kafka server that stores data | Broker-1, Broker-2 |
| Topic | Category of messages | order-events |
| Partition | Division inside topic | Partition-0,1,2 |
| Offset | Message position number | 0,1,2,3… |
| Producer | Sends data | Order Service |
| Consumer | Reads data | Billing Service |
| Zookeeper | Manages cluster | Tracks brokers & topics |

## 7️⃣ Why Developers Use Kafka?

Kafka provides several key advantages:  
- High throughput – can process millions of messages per second.  
- Fault tolerance – data loss is minimal.  
- Scalability – easily handles increased traffic.  
- Real-time streaming – supports instant data updates.  
- Decoupling – enables microservices independence.

Prepared by: Amit Kumar  
Date: October 2025