

AWS

Spark: /hadoop--> 128 Mb, 64 Mb

batch data

finance, banking

ola, uber

driver customers

Linkedin

-- data needs to be transfer in real time(near real time)

real time: kafka -architecture

AWS: kinesis (handles)

distributed system

Gb---> few kb

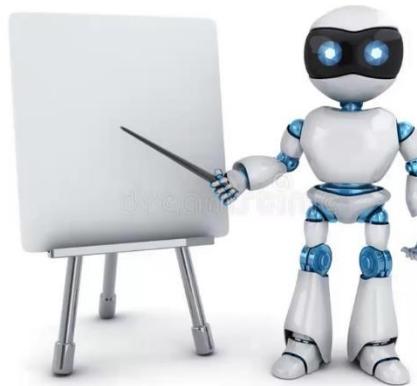
Ln 1, Col 31 258 characters 100% Windows (CRLF) UTF-8

Cloudy 21°C Search ENG IN 07:20 17-07-2024 PRE



# Kafka for Big Data

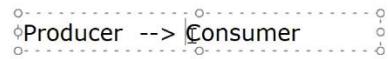
## Overview



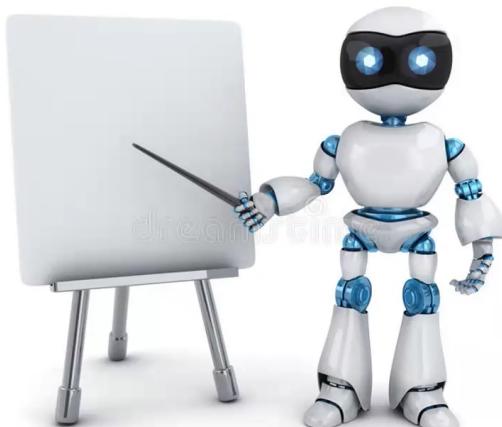
block/chunk size- few kb

# Kafka for Big Data

## Overview



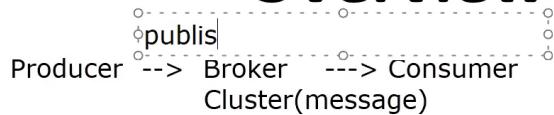
To exit full screen, press Esc



block/chunk size- few kb

# Kafka for Big Data

## Overview



Topic 1  
Topic 2  
Topic 3

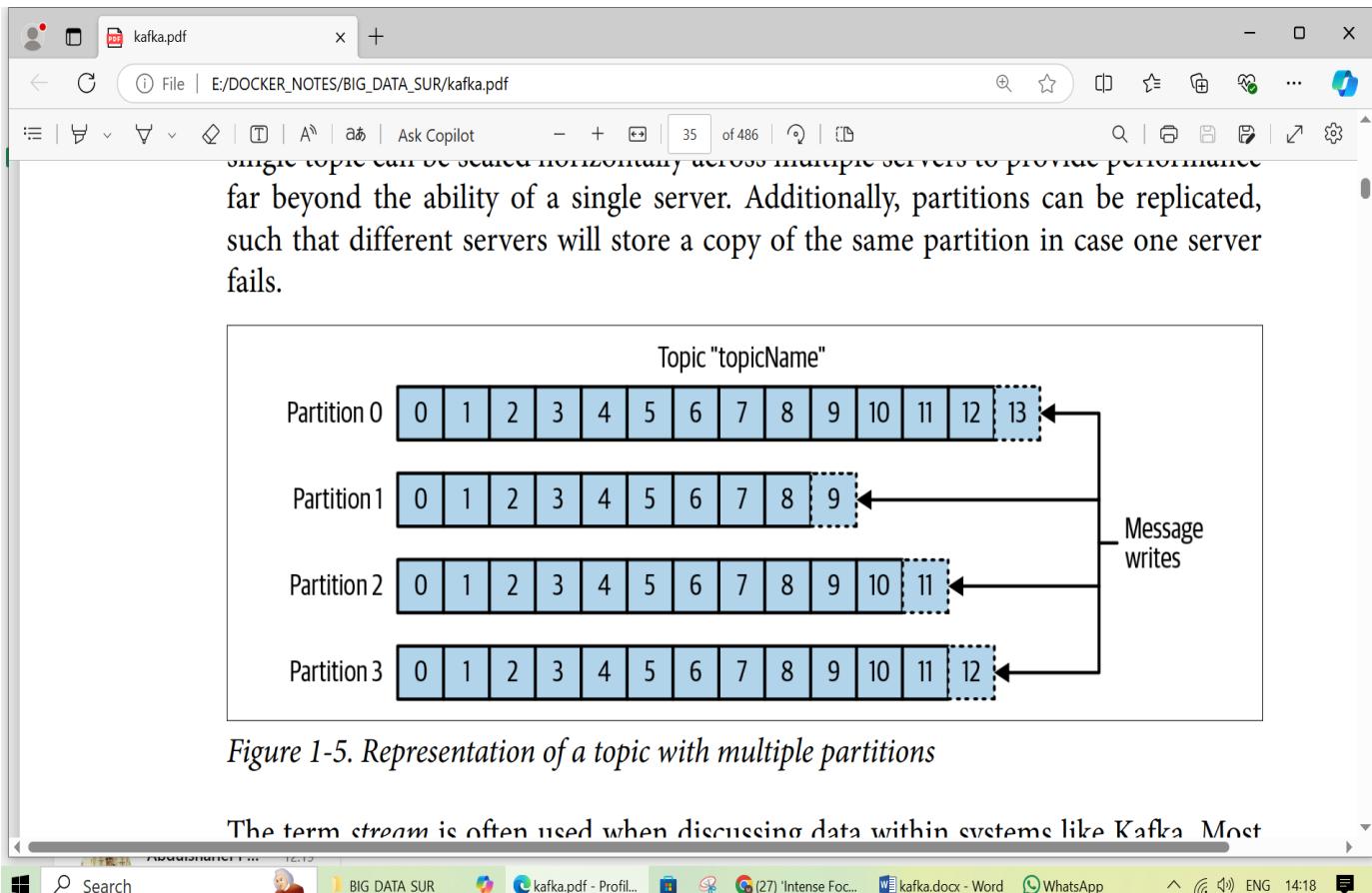


Figure 1-5. Representation of a topic with multiple partitions

The term *stream* is often used when discussing data within systems like Kafka. Most

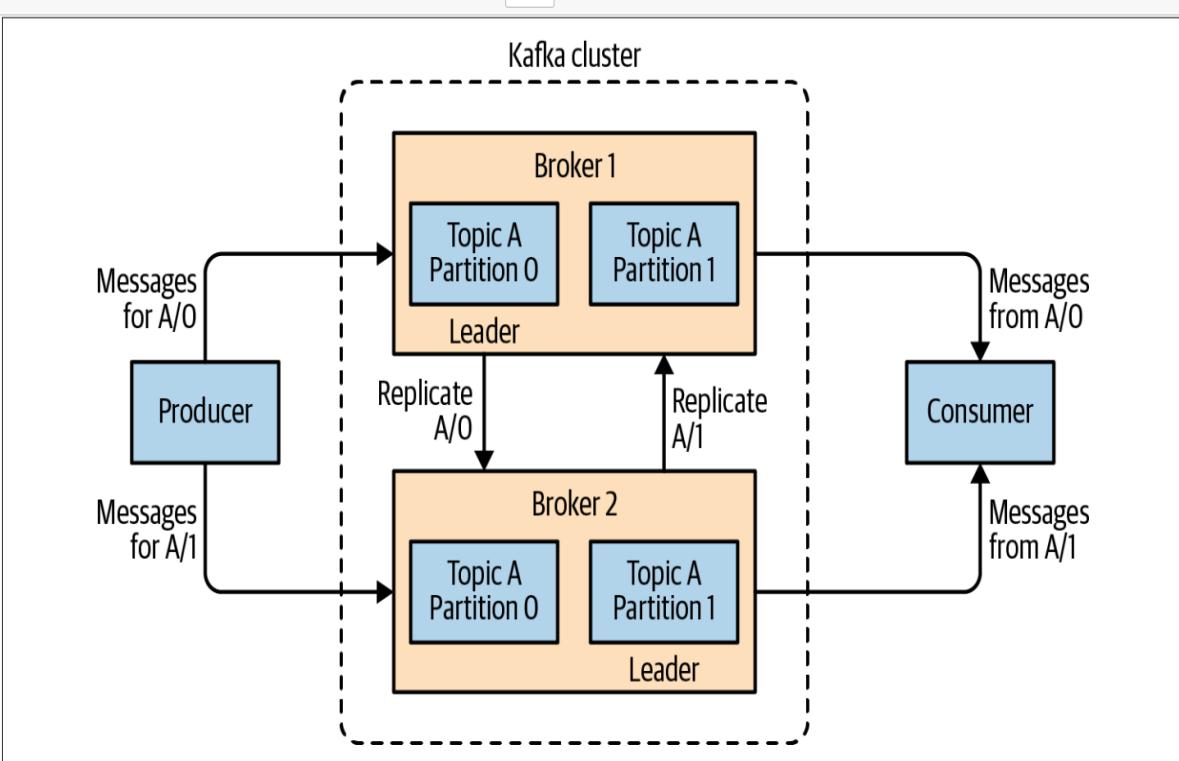


Figure 1-7 Replication of partitions in a cluster

defines a minimum amount of data available at any time. Individual topics can also be configured with their own retention settings so that messages are stored for only as long as they are useful. For example, a tracking topic might be retained for several days, whereas application metrics might be retained for only a few hours. Topics can also be configured as *log compacted*, which means that Kafka will retain only the last message produced with a specific key. This can be useful for changelog-type data, where only the last update is interesting.

## Multiple Clusters

As Kafka deployments grow, it is often advantageous to have multiple clusters. There are several reasons why this can be useful:

- Segregation of types of data
- Isolation for security requirements
- Multiple datacenters (disaster recovery)

When working with multiple datacenters in particular, it is often required that messages be copied between them. In this way, online applications can have access to user activity at both sites. For example, if a user changes public information in their profile, that change will need to be visible regardless of the datacenter in which search

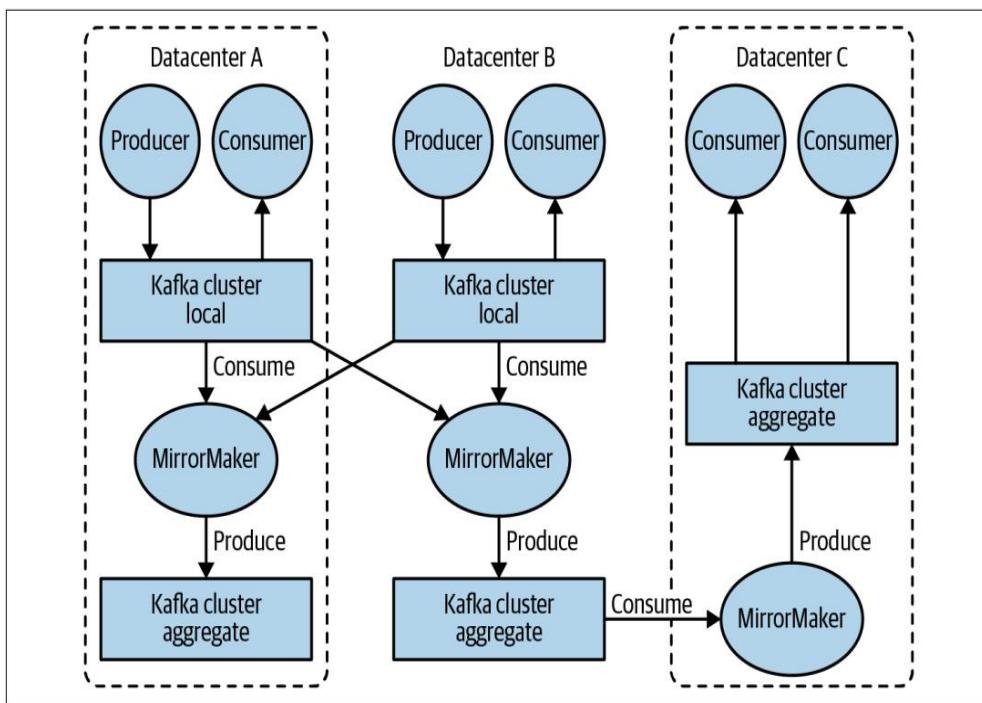


Figure 1-8. Multiple datacenters architecture

---

direct connections of any sort. Components can be added and removed as business cases are created and dissolved, and producers do not need to be concerned about who is using the data or the number of consuming applications.

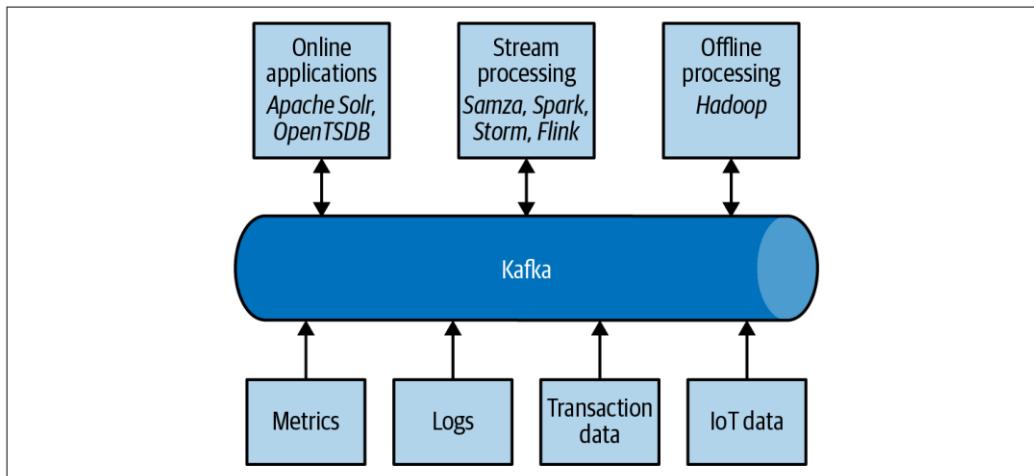


Figure 1-9. A big data ecosystem

## Activity tracking

The original use case for Kafka, as it was designed at LinkedIn, is that of user activity tracking. A website's users interact with frontend applications, which generate messages regarding actions the user is taking. This can be passive information, such as

page views and click tracking, or it can be more complex actions, such as information that a user adds to their profile. The messages are published to one or more topics, which are then consumed by applications on the backend. These applications may be generating reports, feeding machine learning systems, updating search results, or performing other operations that are necessary to provide a rich user experience.

## Messaging

Kafka is also used for messaging, where applications need to send notifications (such as emails) to users. Those applications can produce messages without needing to be concerned about formatting or how the messages will actually be sent. A single application can then read all the messages to be sent and handle them consistently, including:

- Formatting the messages (also known as *decorating*) using a common look and feel
- Collecting multiple messages into a single notification to be sent
- Applying a user's preferences for how they want to receive messages

Using a single application for this avoids the need to duplicate functionality in multiple applications, as well as allows operations like aggregation that would not otherwise be possible.

## Metrics and logging

Kafka is also ideal for collecting application and system metrics and logs. This is a use

While almost all usage of Kafka can be thought of as stream processing, the term is typically used to refer to applications that provide similar functionality to map/reduce processing in Hadoop. Hadoop usually relies on aggregation of data over a long time frame, either hours or days. Stream processing operates on data in real time, as quickly as messages are produced. Stream frameworks allow users to write small applications to operate on Kafka messages, performing tasks such as counting metrics, partitioning messages for efficient processing by other applications, or transforming messages using data from multiple sources. Stream processing is covered in [Chapter 14](#).

## Kafka's Origin

Kafka was created to address the data pipeline problem at LinkedIn. It was designed to provide a high-performance messaging system that can handle many types of data and provide clean, structured data about user activity and system metrics in real time.

Data really powers everything that we do.

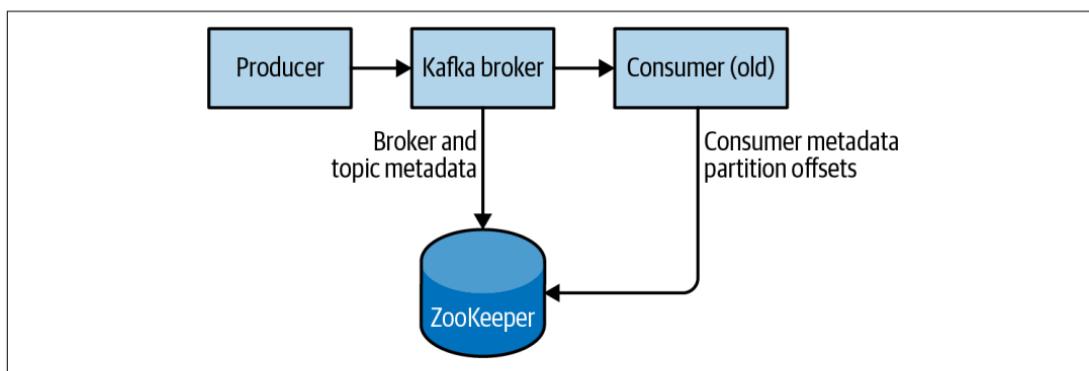
—Jeff Weiner, former CEO of LinkedIn

## LinkedIn's Problem

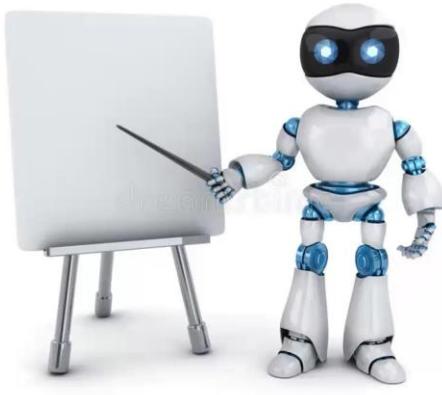
Similar to the example described at the beginning of this chapter, LinkedIn had a system for collecting system and application metrics that used custom collectors and

## Installing ZooKeeper

Apache Kafka uses Apache ZooKeeper to store metadata about the Kafka cluster, as well as consumer client details, as shown in [Figure 2-1](#). ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. This book won't go into extensive detail about ZooKeeper but will limit explanations to only what is needed to operate Kafka. While it is possible to run a ZooKeeper server using scripts contained in the Kafka distribution, it is trivial to install a full version of ZooKeeper from the distribution.



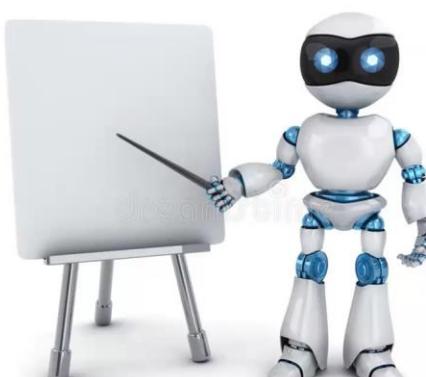
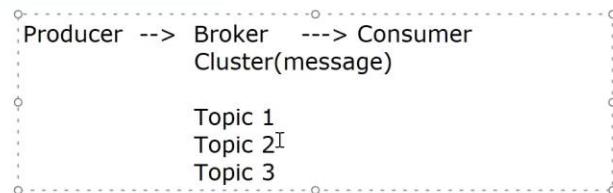
*Figure 2-1. Kafka and ZooKeeper*



block/chunk size- few kb

## Kafka for Big Data

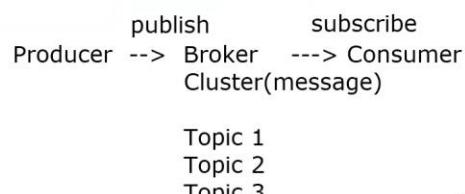
### Overview



block/chunk size- few kb

## Kafka for Big Data

### Overview



## What is Streaming?

---

Streaming means continuous generation of data by multiple data sources.

The strategy used here is to send data in small chunks.

The size of chunk can vary upto few kilobytes.



## What is Kafka?

---

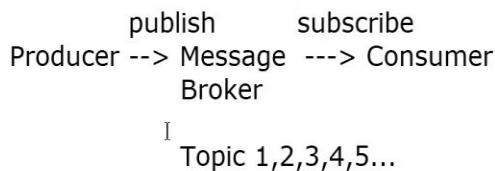
Apache Kafka is a distributed streaming platform that is used for building real-time data pipelines and streaming applications.

Kafka is a publish-subscribe messaging system, meaning that it allows multiple consumers to read from a topic (or "topic partition") at the same time. It also allows for storing streams of records in a fault-tolerant way.

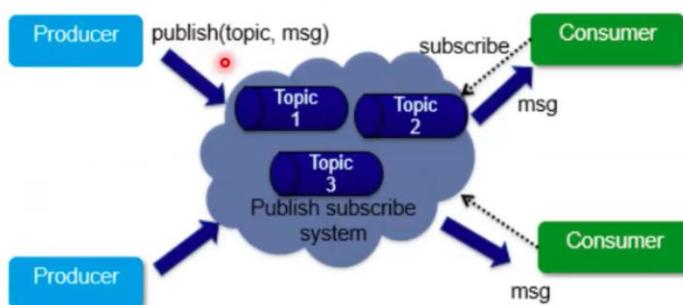
# What is Kafka?

Apache Kafka is a distributed streaming platform that is used for building real-time data pipelines and streaming applications.

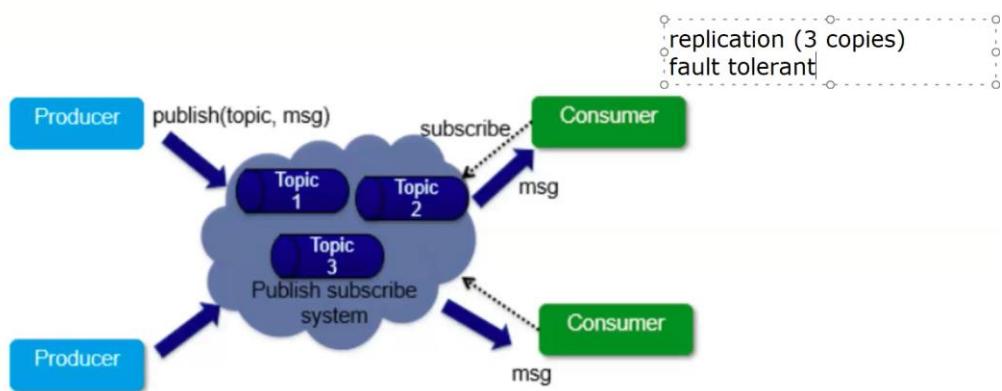
Kafka is a publish-subscribe messaging system, meaning that it allows multiple consumers to read from a topic (or "topic partition") at the same time. It also allows for storing streams of records in a fault-tolerant way.



# What is Kafka?



# What is Kafka?



## What is Kafka?

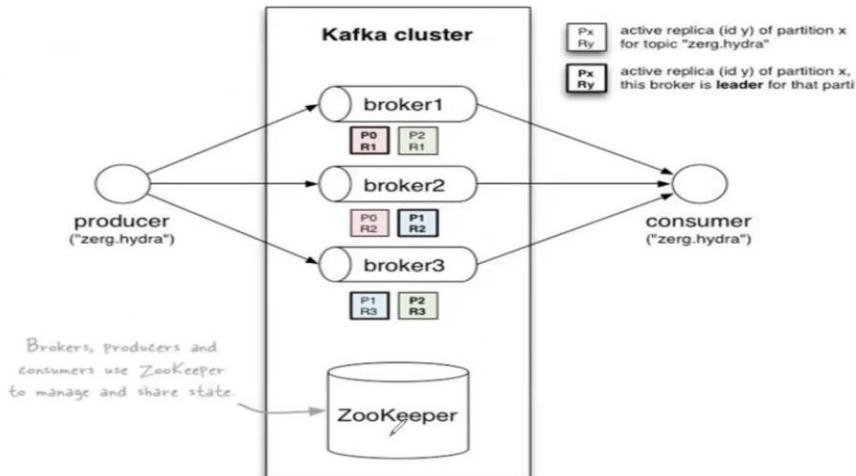
Producers: Applications that generate data and send it to topics in a Kafka cluster.

Topics: A category or feed name to which records are published.

Brokers: A cluster of servers that stores and distributes the published records

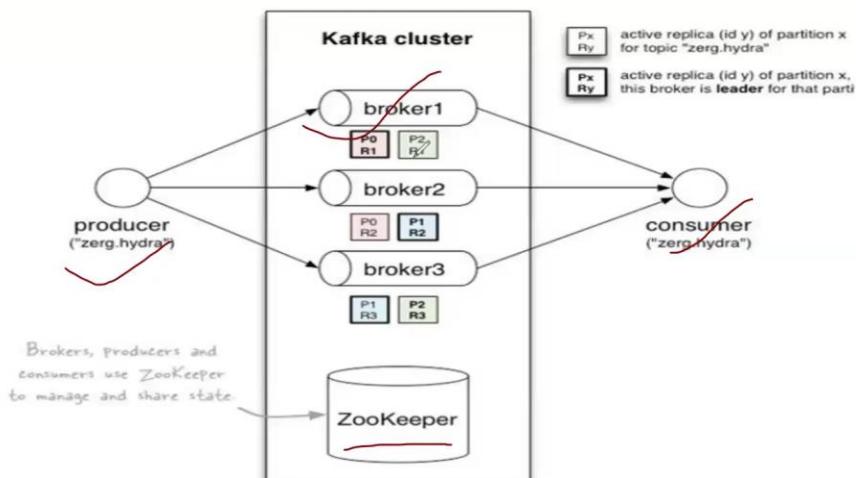
Consumers: Applications that read data from a topic.

# Kafka Architecture



DataMites

# Kafka Architecture



DataMites

Kafka For BigData [Protected View] - PowerPoint

To exit full screen, press Esc

File Home Insert Design Transitions Animations Slide Show Review View Help Share

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View.

Enable Editing

4 AWS Sparkhadoop-- 128 Mb, 64 Mb

5 Spark:/hadoop--> 128 Mb, 64 Mb  
batch data  
finance, banking

6 ola, uber  
driver customers

7 Linkedin  
-- data needs to be transfer in real time(near real time)  
real time: kafka -architecture

8 AWS: kinesis (handles)  
distributed system  
Gb--> few kb  
1) zookeeper start  
2) Start broker

9 Configuring Kafka cluster

Slide 8 of 11 English (U) Ln 23. Col 16 294 characters 100% Windows (CRLF) UTF-8 + 93%

<https://kafka.apache.org/downloads>

### 3.8.0

- Released July 29, 2024
- [Release Notes](#)
- Docker image: [apache/kafka:3.8.0](#).
- Docker Native image: [apache/kafka-native:3.8.0](#).
- Source download: [kafka-3.8.0-src.tgz](#) ([asc](#), [sha512](#))
- Binary downloads:

- Scala 2.12 - [kafka\\_2.12-3.8.0.tgz](#) ([asc](#), [sha512](#))
- Scala 2.13 - [kafka\\_2.13-3.8.0.tgz](#) ([asc](#), [sha512](#))

We build for multiple versions of Scala. This only matters if you are using Scala and you want a version built for the same Scala version you use. Otherwise any version should work (2.13 is recommended).

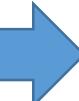
Kafka 3.8.0 includes a significant number of new features and fixes. For more information, please read our [blog post](#) and the detailed [Release Notes](#).

Extract>> bin >> windows

Run this command in CLI

```
bin\windows\zookeeper-server-start.bat config\zookeeper.properties
```

**zookeeper is running on client port address 2181**



```
C:\Windows\System32\cmd.exe - bin\windows\zookeeper-server-start.bat config\zookeeper.properties
Microsoft Windows [Version 10.0.19045.4894]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LENOVO\kafka>bin\windows\zookeeper-server-start.bat config\zookeeper.properties
[2024-09-20 07:18:46,677] INFO Reading configuration from: config\zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,681] WARN config\zookeeper.properties is relative. Prepend .\ to indicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,692] WARN \tmp\zookeeper is relative. Prepend .\ to indicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,696] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,696] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,696] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,697] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,702] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-09-20 07:18:46,703] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-09-20 07:18:46,703] INFO Peer task is not scheduled. (org.apache.zookeeper.server.DatadirCleanupManager)
[2024-09-20 07:18:46,703] WARN Either no config or no quorum defined in config, running in standalone mode (org.apache.zookeeper.server.quorum.QuorumPeerMain)
[2024-09-20 07:18:46,706] INFO Log4j 1.2 jmx support not found; jmx disabled. (org.apache.zookeeper.jmx.ManagedUtil)
[2024-09-20 07:18:46,708] INFO Reading configuration from: config\zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,708] WARN config\zookeeper.properties is relative. Prepend .\ to indicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,709] WARN \tmp\zookeeper is relative. Prepend .\ to indicate that you're sure! (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,710] INFO clientPortAddress is 0.0.0.0:2181 (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,710] INFO secureClientPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,711] INFO observerMasterPort is not set (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,711] INFO metricsProvider.className is org.apache.zookeeper.metrics.impl.DefaultMetricsProvider (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2024-09-20 07:18:46,714] INFO Starting server (org.apache.zookeeper.server.ZooKeeperServerMain)
[2024-09-20 07:18:46,769] INFO ServerMetrics initialized with provider org.apache.zookeeper.metrics.impl.DefaultMetricsProvider@48e4374 (org.apache.zookeeper.server.ServerMetrics)
[2024-09-20 07:18:46,773] INFO ACL digest algorithm is: SHA1 (org.apache.zookeeper.server.auth.DigestAuthenticationProvider)
[2024-09-20 07:18:46,773] INFO zookeeper.DigestAuthenticationProvider.enabled = true (org.apache.zookeeper.server.auth.DigestAuthenticationProvider)
[2024-09-20 07:18:46,780] INFO zookeeper.snapshot.trust.empty : false (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2024-09-20 07:18:46,803] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-20 07:18:46,804] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-20 07:18:46,804] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-20 07:18:46,805] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-20 07:18:46,806] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-20 07:18:46,806] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-20 07:18:46,807] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-20 07:18:46,810] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-20 07:18:46,812] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-20 07:18:46,820] INFO (org.apache.zookeeper.server.ZooKeeperServer)
[2024-09-20 07:18:46,838] INFO Server environment:zookeeper.version=3.8.4-9316c2a7a97e1666d8f4593f34dd6fc36ecc436c, built on 2024-02-12 22:16 UTC (org.apache.zookeeper.server.ZooKeeperServer)
```

# Now we need to start the kafka

```
bin\windows\kafka-server-start.bat config\server.properties
```

```
bin\windows\kafka-topics.bat --create --topic demo-topic
```

```
bin\windows\kafka-topics.bat --create --topic demo-topic --bootstrap-server:9092 --partitions 3
```

```
bin\windows\kafka-topics.bat --create --topic demo-topic --partitions 3 --replication-factor 1 --bootstrap-server localhost:9092
```

```
bin\windows\kafka-topics.bat --bootstrap-server localhost:9092 --list
```

create two CLI , in the producer cli paste this

```
bin\windows\kafka-console-producer.bat --broker-list localhost:9092 --topic demo-trail
```

in consumer cli

```
bin\windows\kafka-console-consumer.bat --bootstrap-server localhost:9092 --topic demo-trail --from-beginning
```

[offsetexplorer.com/index.html](http://offsetexplorer.com/index.html)

```
C:\Users\LENOVO\kafka>bin\windows\kafka-topics.bat --bootstrap-server localhost:9092 --list  
__consumer_offsets  
demo-topic  
demo-topic11  
demo-trail  
kareem
```

```
C:\Users\LENOVO\kafka>
```

```
C:\Users\LENOVO\kafka>bin\windows\kafka-topics.bat --bootstrap-server localhost:9092 --describe --topic kareem  
[2024-09-20 18:30:36,301] WARN [AdminClient clientId=adminclient-1] The DescribeTopicPartitions API is not supported, using Metadata.fetchTopicsMetadata(KafkaClientsMetadataSupplierSupplier).  
Topic: kareem TopicId: Lub1ihy_T1C2epPQtkr9cw PartitionCount: 3 ReplicationFactor: 1 Configs:  
  Topic: kareem Partition: 0 Leader: 0 Replicas: 0 Isr: 0 Elr: N/A LastKnownElr: N/A  
  Topic: kareem Partition: 1 Leader: 0 Replicas: 0 Isr: 0 Elr: N/A LastKnownElr: N/A  
  Topic: kareem Partition: 2 Leader: 0 Replicas: 0 Isr: 0 Elr: N/A LastKnownElr: N/A
```

```
C:\Users\LENOVO\kafka>
```

My Webinars - Zoom Post Attendee - Zoom Notebook - Editor - AWS Glue Roles | IAM | Global

1) Big data Gb, Tb, Pb

```
import sys
from awsglue.transforms import *
from awsglue.utils import *
from pyspark.context import *
from awsglue.context import *
from awsglue.job import *

sc = SparkContext()
glueContext = GlueContext(sc)
spark = glueContext
job = Job(glueContext)

Welcome to the Glue Notebook!
For more information, please visit our documentation at:
Please view our Getting Started guide at:
https://docs.aws.amazon.com/glue/latest/dg/quick-start.html
```

Initialized (additional)

CloudShell Feedback

21°C Mostly cloudy

Q Search

ENG IN 07:30 22-07-2024

My Webinars - Zoom Post Attendee - Zoom Manage AWS Resources - AWS Glue

1) Big data Gb, Tb, Pb

```
Drag and Drop: 60% of the time
```

2) Traditional data: kb, Mb

Lambda: Pandas, Numpy

2) Azure databricks

Real-time streaming:

1) Kafka

d) Producer- sending  
e) Consumer- receiving  
b) Broker/kafka server -- write the message/ topic  
c) Topic-- start the topic  
a) Zookeeper- manager

2) Kinesis:

a) data stream  
b) data firehose  
c) data analytics

To exit full screen, press Esc

AWS Management Console

Products Services

Everywhere in one web interface

Ln 31, Col 1 436 characters 100% Windows (CRLF) UTF-8

1) Big data Gb, Tb, Pb

File Edit View

Glue: Notebook (can't use in Free account)

Drag and Drop: 60% of the time

2) Traditional data: kb, Mb

Lambda: Pandas, [NumPy](#)

2) Azure [databricks](#)

Real-time streaming:

1) Kafka

- d) Producer- sending
- e) Consumer- receiving
- b) Broker/kafka server -- write the message/ topic
- c) Topic-- start the topic
- a) Zookeeper- manager

2) Kinesis:

- a) data stream
- b) data firehose
- c) data analytics

EC2(logs) ---> data stream --- data firehose ---> s3

```
graph LR; EC2[EC2] -- "data stream" --> DF[DF]; DF -- "data firehose" --> S3[S3]
```

Ln 33, Col 41 | 634 characters | 100% | Windows (CRLF) | UTF-8

1) Big data Gb, Tb, Pb

File Edit View

Glue: Notebook (can't use in Free account)

Drag and Drop: 60% of the time

2) Traditional data: kb, Mb

Lambda: Pandas, [NumPy](#)

2) Azure [databricks](#)

Real-time streaming:

1) Kafka

- d) Producer- sending
- e) Consumer- receiving
- b) Broker/kafka server -- write the message/ topic
- c) Topic-- start the topic
- a) Zookeeper- manager

2) Kinesis:

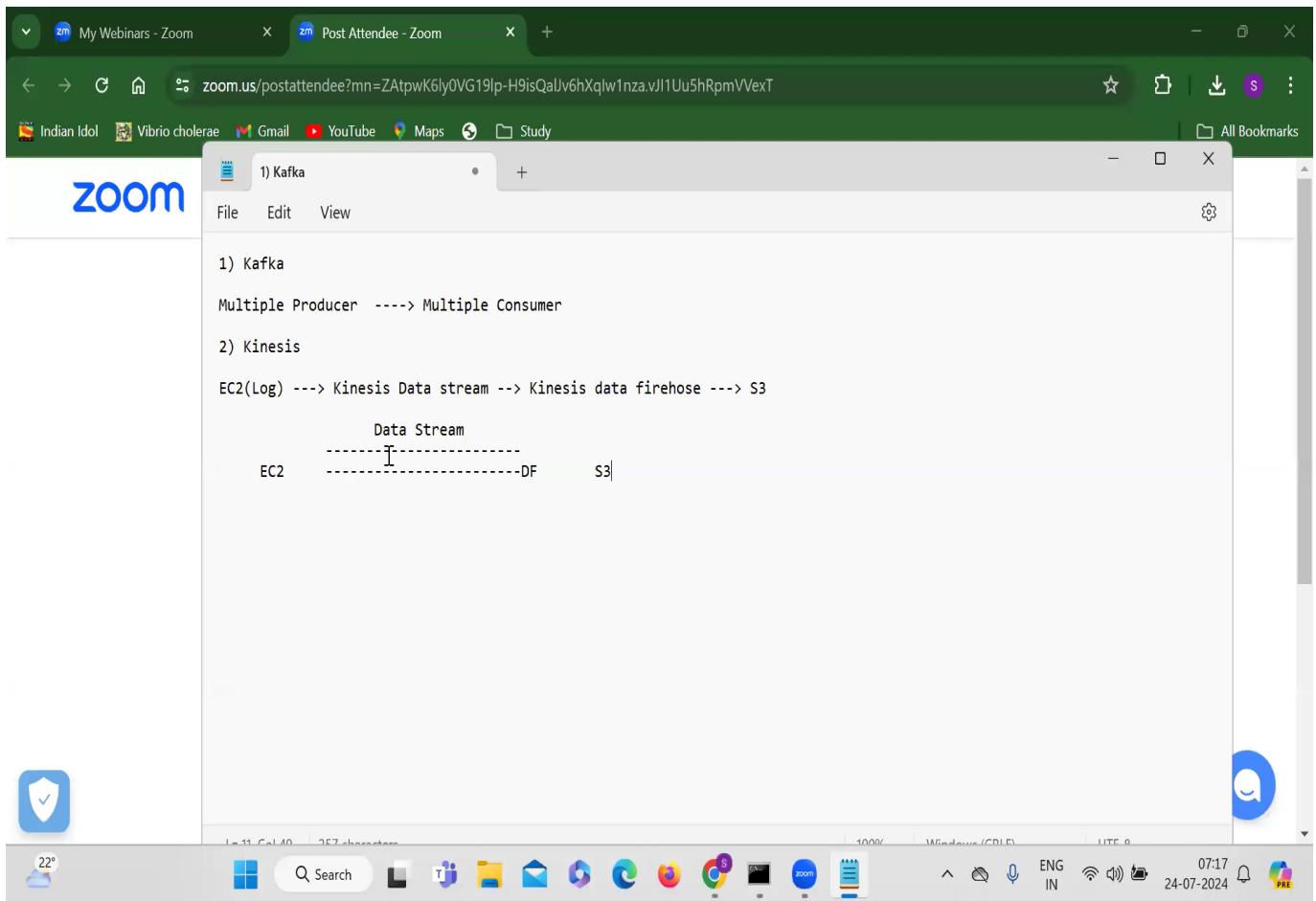
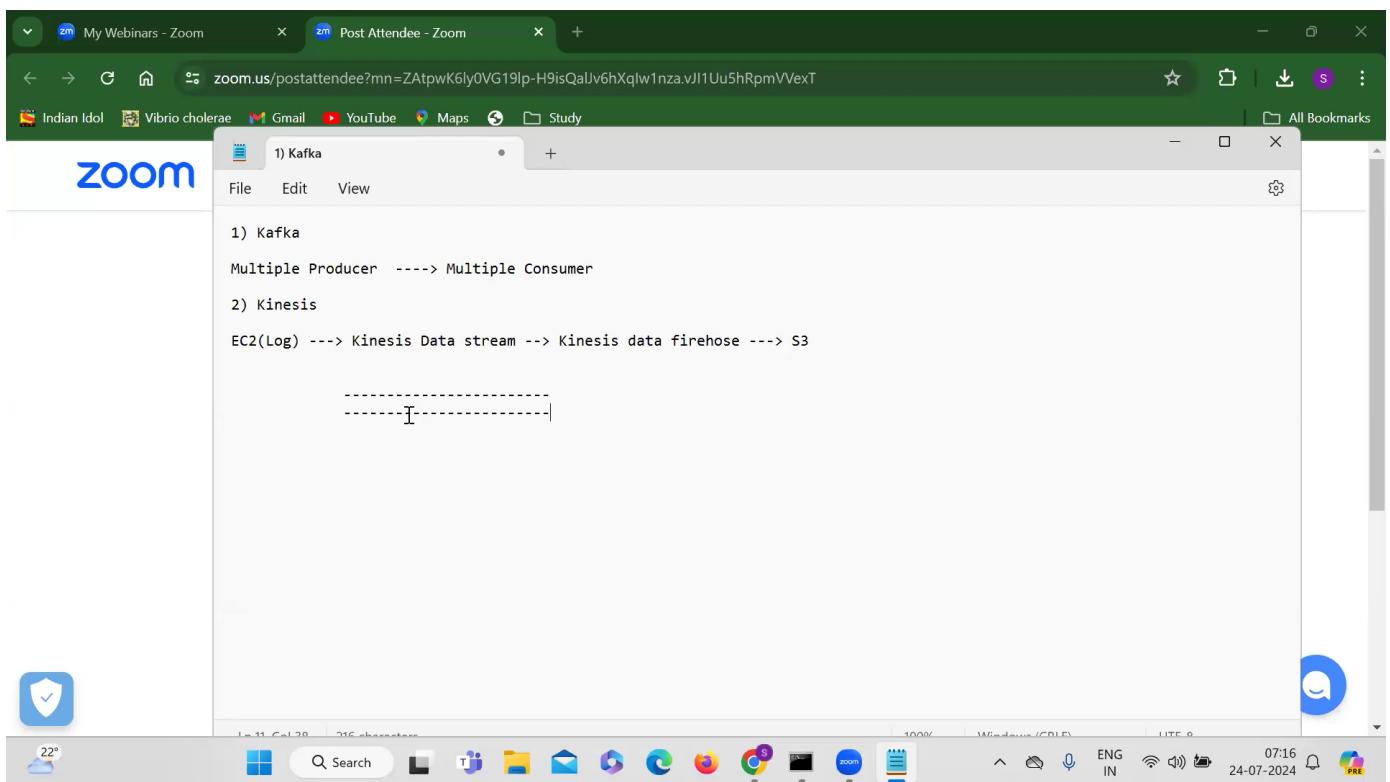
- a) data stream
- b) data firehose
- c) data analytics

EC2(logs) ---> data stream --- data firehose ---> s3

```
graph LR; EC2[EC2] -- "data stream" --> DF[DF]; DF -- "data firehose" --> S3[S3]
```

Ln 34, Col 54 | 645 characters | 100% | Windows (CRLF) | UTF-8

08:31 22-07-2024 ENG IN



My Webinars - Zoom Post Attendee - Zoom zoom.us/postattendee?mn=ZAtpwK6ly0VG19lp-H9isQalJv6hXqlw1nza.vJl1Uu5hRpmVVexT

Indian Idol Vibrio cholerae Gmail YouTube Maps Study All Bookmarks

**ZOOM**

1) Kafka

Multiple Producer ----> Multiple Consumer

2) Kinesis

EC2(Log) ---> Kinesis Data stream --> Kinesis data firehose ---> S3

Data Stream  
-----  
EC2 -----DF----- S3  
Lambda

Step 1: IAM role (kinesis full access, S3 full access)

22° 🔍 Search 📁 Trello 📩 Gmail 🎵 YouTube 🌎 Maps 🌐 Study 🔍 All Bookmarks

14:16 CALLED 350 characters 10000 14:16 07:19 ENG IN 🔍 LITE 🔍 24-07-2024 🔍 PRE

My Webinars - Zoom Post Attendee - Zoom zoom.us/postattendee?mn=ZAtpwK6ly0VG19lp-H9isQalJv6hXqlw1nza.vJl1Uu5hRpmVVexT

Indian Idol Vibrio cholerae Gmail YouTube Maps Study All Bookmarks

**ZOOM**

1) Kafka

Multiple Producer ----> Multiple Consumer

2) Kinesis

EC2(Log) ---> Kinesis Data stream --> Kinesis data firehose ---> S3

Data Stream  
-----  
EC2 -----DF----- S3  
Lambda

Step 1: IAM role on EC2 (kinesis full access, S3 full access)

██████████

My Webinars - Zoom Post Attendee - Zoom zoom.us/postattendee?mn=ZAtpwK6ly0VG19lp-H9isQaJv6hXqlw1nza.vJl1Uu5hRpmVVexT Indian Idol Vibrio cholerae Gmail YouTube Maps Study

1) Kafka

Multiple Producer ----> Multiple Consumer

2) Kinesis

EC2(Log) ---> Kinesis Data stream --> Kinesis data firehose ---> S3

EC2 -----DF----- S3  
Data Stream  
Lambda

Step 1: IAM role on EC2 (kinesis full access, S3 full access)  
Step 2: Create a user (access key, secret access key)

I

The screenshot shows a Zoom video call interface. The main content area displays a presentation slide with two sections: '1) Kafka' and '2) Kinesis'. Section 1, 'Kafka', describes a producer-consumer model where multiple producers send data to multiple consumers. Section 2, 'Kinesis', shows a flow from EC2 logs through a Kinesis Data Stream, Kinesis Data Firehose, and finally to S3, with a Lambda function integrated into the stream. Below the slide, a terminal window is open with commands for Step 1 (IAM role creation) and Step 2 (user creation). The bottom of the screen shows the Windows taskbar with various pinned icons and system status indicators.

