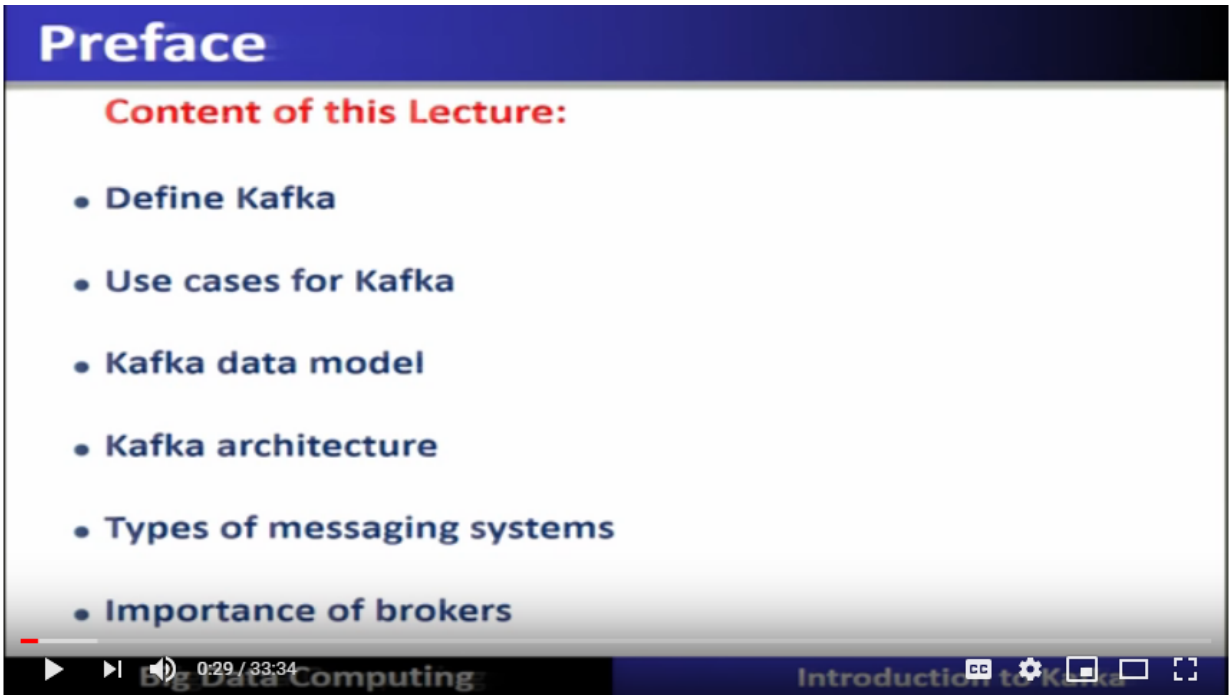
Lecture 22 Introduction to Kafka

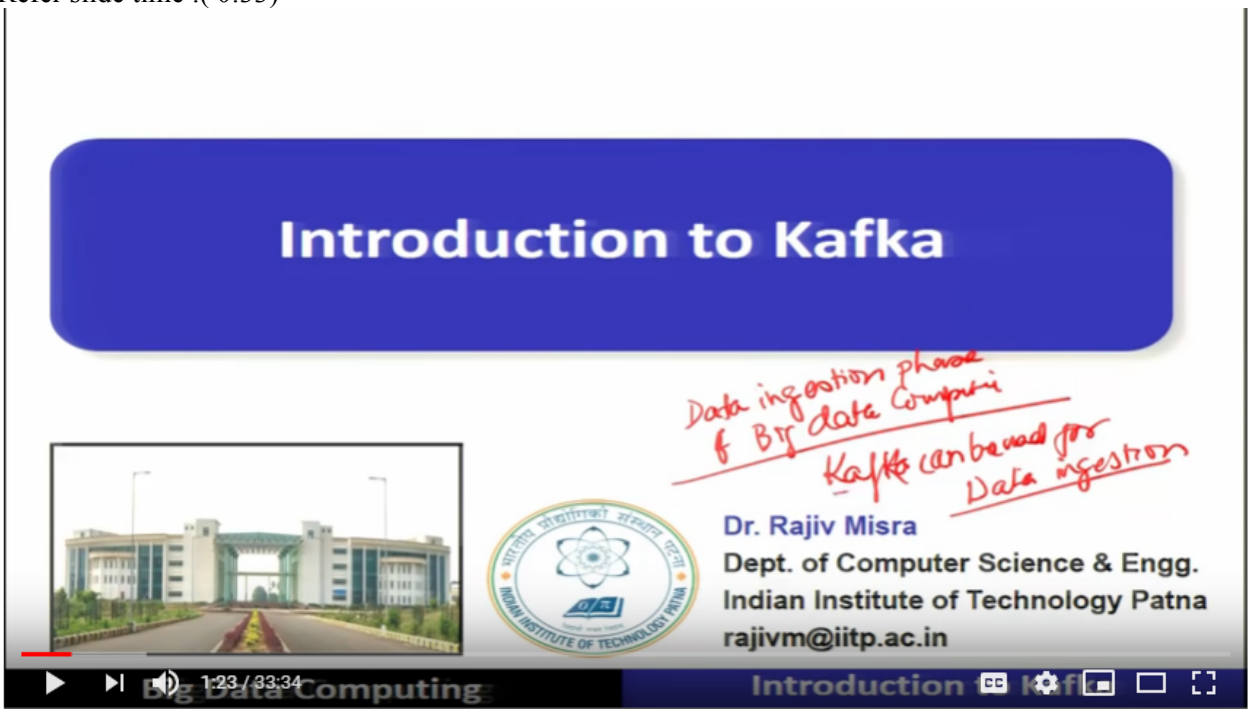
Introduction to Kafka

Refer slide time :(0:17)



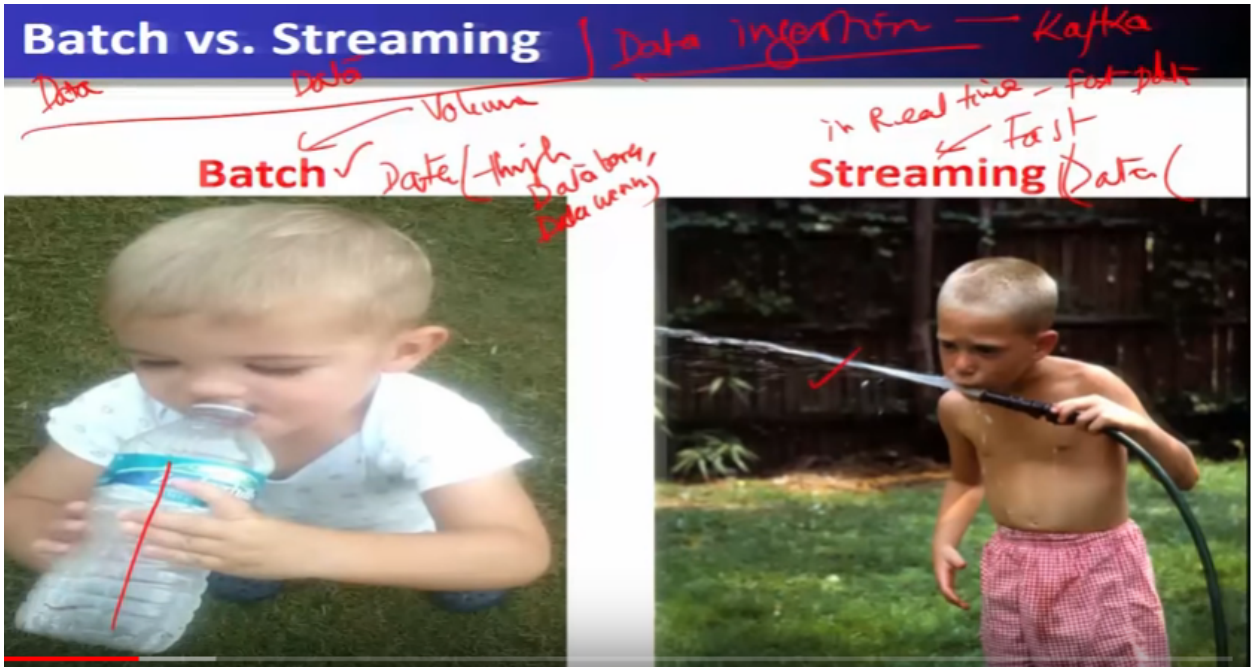
Preface content of this lecture; we will define, what is a Kafka, we will see the use case of Kafka and Kafka is a data model, Kafka is a architecture, types of messaging system, importance of Kafka, as the broker.

Refer slide time :(0:33)



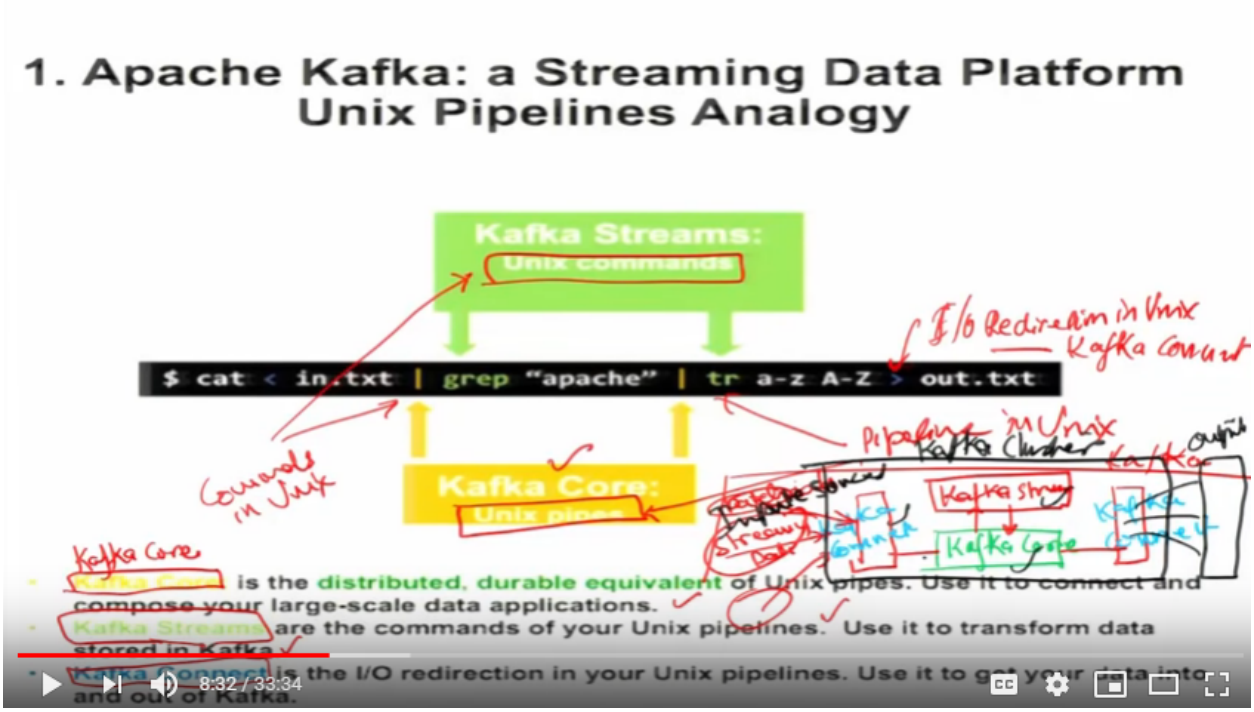
So, before we go ahead, let us see where what is the use of Kafka in big data. So, the Kafka is used for the data injection, data ingestion, of big data computing. So, in this particular discussion, we will see the details of the Kafka and how the Kafka, can be used, for data injection.

Refer slide time :(01:23)



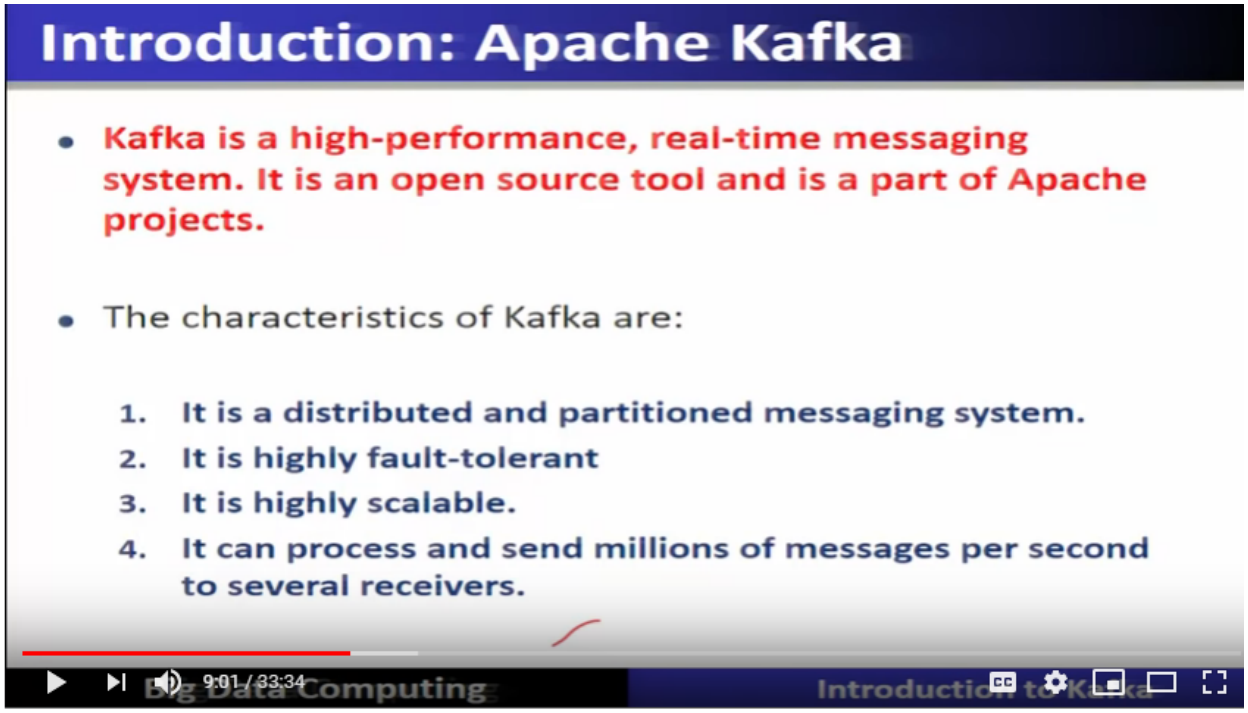
Before we understand, before we go in more detail, let us understand two important things, using this particular picture. One is the batch, batch jobs, means die stored this is the stored water this is if it is consumed, then it is called a,' Batch Processing,' whereas the, the water which is in the form of stream, if you are consuming then it is called,' Stream'. So, it is the data if we are consuming, in our scenario it is called a batch,' Data,' if we are using the, the stored data, for its consumption, that is through the through the databases data, warehouses and so on. And this is the streaming data, if we are consuming the data for any application such as, the Twitter stream data or the socket API socket connection, which brings into the network data and so on. So, those data is called the,' Streaming Data'. So, as the data is flowing we have to compute, this particular data then it is called the,' Streaming Data Processing'. So, this kind of data here, we have the big volume of data which is we have to store, that is the volume, we have to deal in the big data similarly, the streaming data, that means data is moving very fast and we have to compute, this processing which cannot be done, by a single computer system. So, that means how we have to compute in the real time, this fast data. So, we will see here, what do you mean by the how batch data and what is the streaming data and how we are going to handle, these cases in the data injection system. So, what is the technology? So, we will be discussing the Kafka, as the technology which deals with the both type of input sources, of the data that is the best data as the input source and the streaming data, as the input source.

Refer slide time :(03:54)



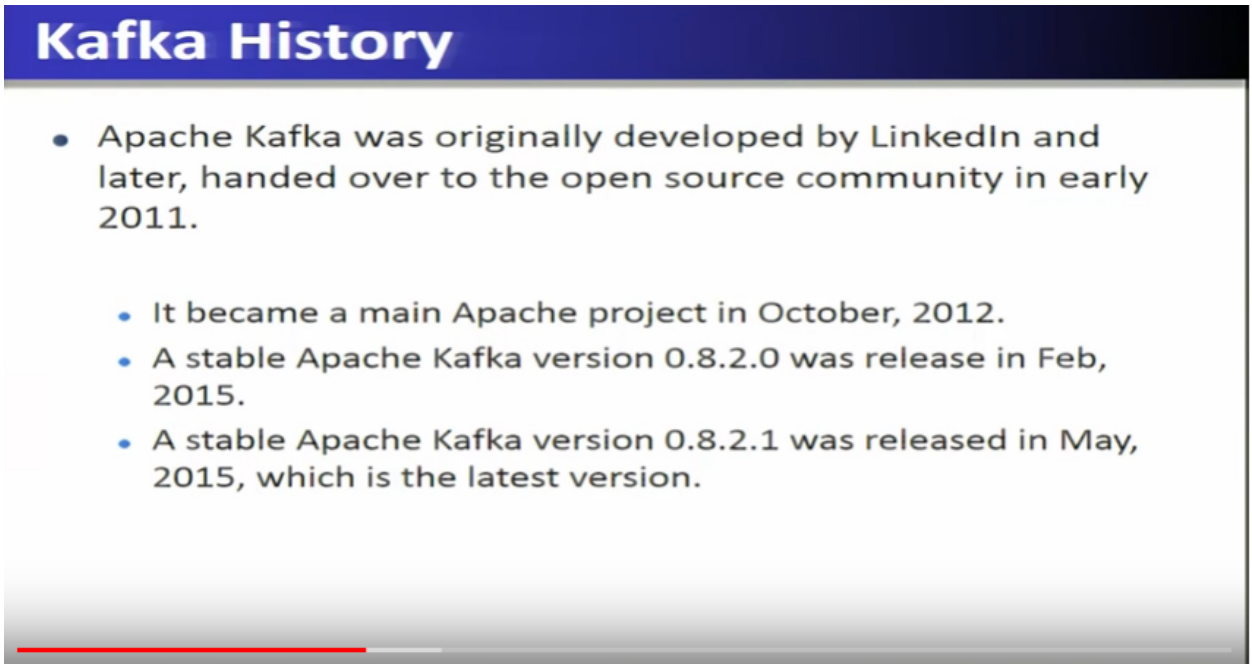
So, before we go ahead let us see, the Apache Kafka, as the streaming data platform, having the UNIX pipeline analogy. So, just if you have familiar with the UNIX pipeline analogy, now we will understand, using that analogy about the Kafka Apache a Kafka. So, here in the UNIX domain you might have noticed that, there are certain Unix commands and we can now form a pipeline, of these set of unit command, Unix command together and this is called the,' Unix Pipeline'. So, similar to this analogy, of eunuchs we will see different components of Kafka. So, there is a Kafka core, which is like UNIX pipes this, is called,' Kafka Core'. So, Kafka core is a distributed, durable equivalent of UNIX pipes. So, it used to connect and compose your large-scale data application, the second aspect here is called,' Kafka Streams', which is having the similar analogy as the commands you use in the UNIX pipelines. So, this particular commands are used to transform, the data which is stored in the Kafka third component is called,' Kafka Connect'. So, Kafka Connect is an IO redirection, just like here and this indicates Kafka connector. So, Kafka connect is the IO it is having analogy, with the IO II direction in the UNIX pipelines. So, it is used to get your data into and out of the Kafka system. So, if we draw a diagram, of a Kafka system, we will how, this is called' Kafka Connect', the next thing is about the Kafka streams, Kafka core and then we have the Kafka streams. So, this is the complete picture and here there will be the input sources, input sources may be generating the batch data, streaming data. And this is called the,' Input Sources'. And now here there will bean output or a consumers or output sources, this is called the,' Kafka cluster. Now let us go and understand, the design of each and every component, of these Kafka that is what do you mean by the Kafka connects. And Kafka stream and Kafka core and how together all of these components can be used to build the data, ingestion and data computation, pipeline in a big data scenario.

Refer slide time :(08:40)



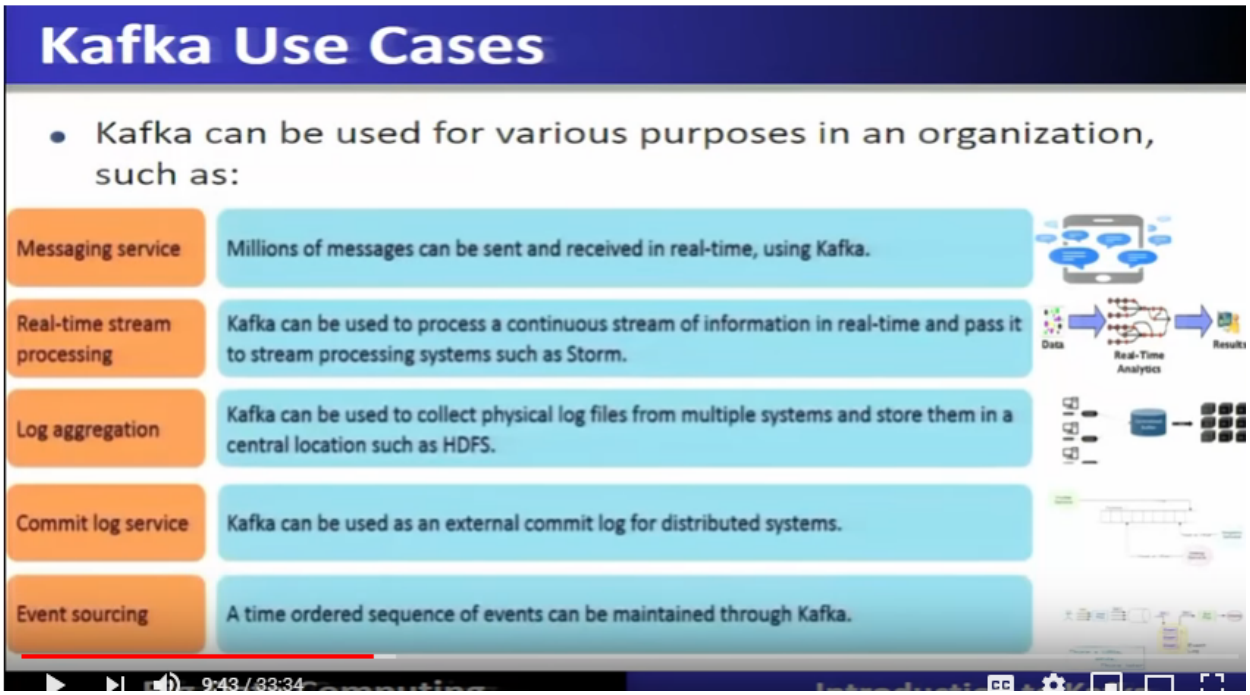
So, Kafka is the high performance real time messaging system and it is an open source tool and is the part of Apache project. So, the characteristics of Kafka are it is the distributed and partition messaging system, it is highly called tolerant, it is highly scalable, it can process and send millions of messages, per second to the several receivers.

Refer slide time :(09:01)



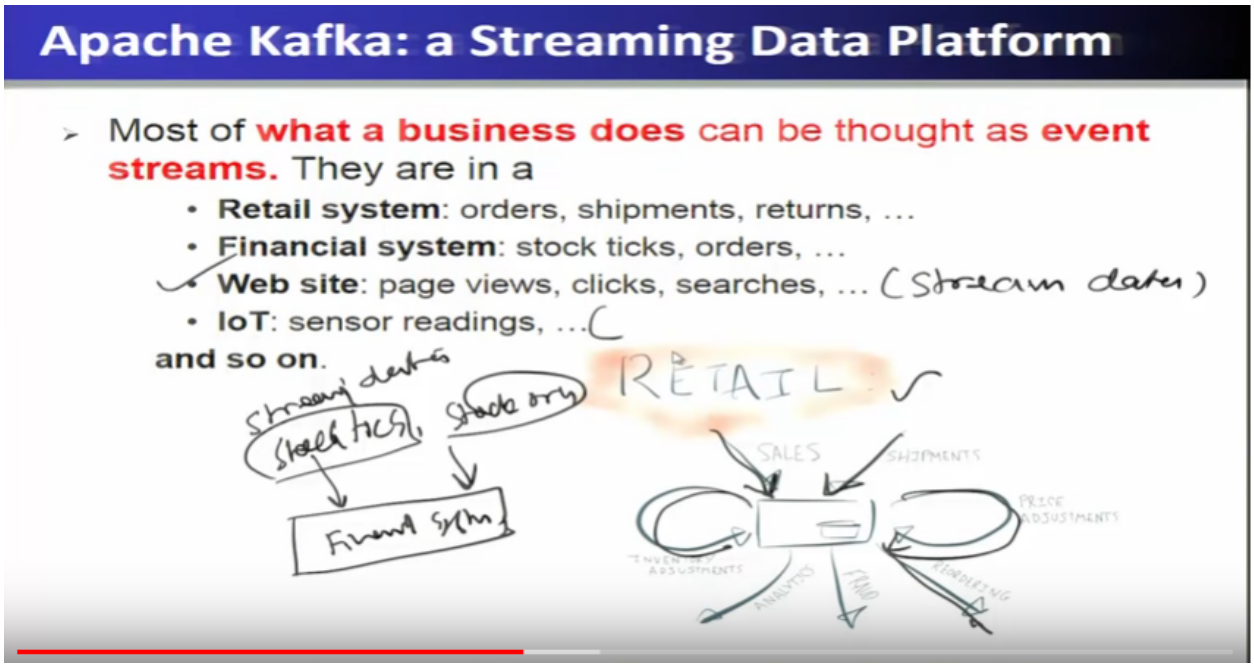
So, the history of apache Kafka, was originally developed by Lyndon and later on handed over to the open source, community in the 2011 it became and main Apache project in October 2012, I stable Apache Kafka version, 0.8.1 released in February 2015 a stable Kafka, version eight point two point one, was released in May 2015 which is the latest version.

Refer slide time :(09:43)



So, let us see some, of the use cases of Kafka so Kafka can be used for various purposes, for ended in the industries, such as Kafka can be used as a messaging server, messaging service that is millions of messages, can be sent and received in the real time using Kafka. So, real time stream processing that means Kafka can be used to process, a continuous stream of information in real time and pass it to the stream processing system, such as storm. So, log aggregation that is Kafka can be used, to collect the physical log files with from multiple systems and store them in the central location, such as HDFS, commit log service Kafka can be used as an external commit log for distributed systems, even sourcing ,time ordered sequence of events, can be maintained through the Kafka. So, these are some of the important, use cases of Kafka, which is used indifferent organizations and is used in applications. So, let us summarize it that Kafka can be used as a messaging server, for handling with the multiple numbers, of messages and in the real time its connotations. Real-time stream processing cover can be used to process continuous stream of information and the, the log aggregation that is Kafka can be used, to collect the physical log files from multiple systems. And store them to the central agencies so, commit log service Kappa can be used, as an external commit log for distributed systems, event sourcing that is a time ordered sequence, of events can be maintained through the Kafka, Apache Kafka a streaming data platform.

Refer slide time :(11:49)

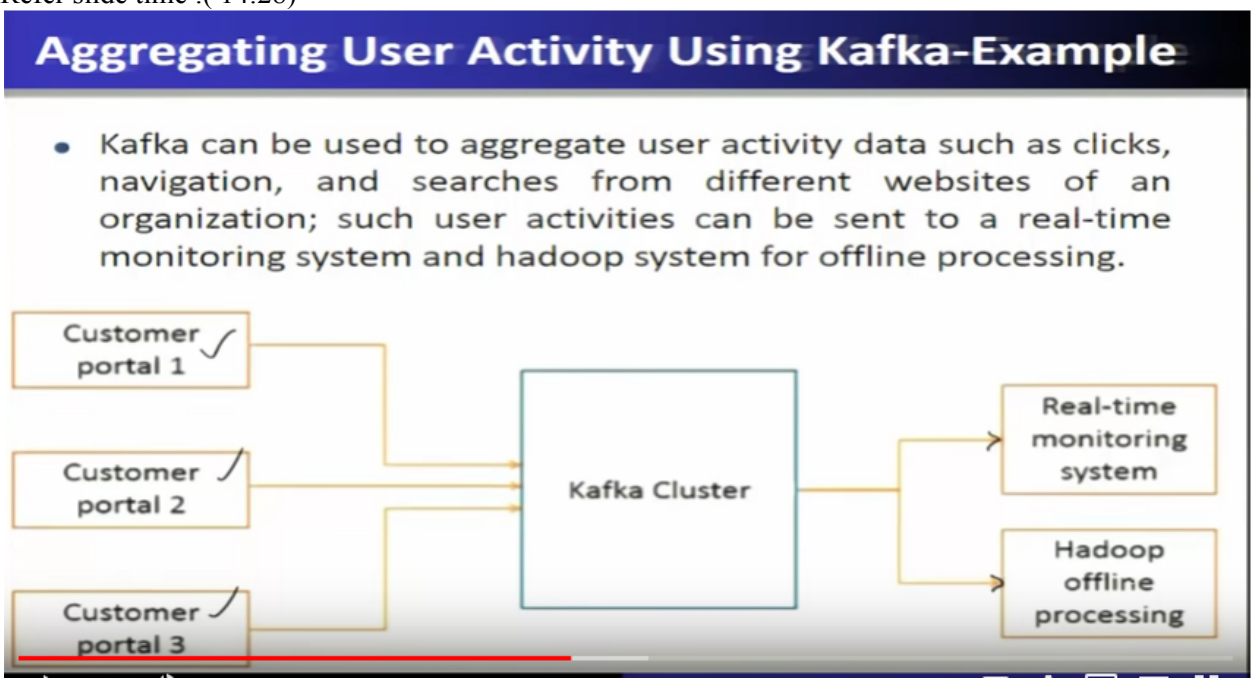


So, most of what a business does can be thought of as an event streams. So, they are in used in the retail system, in the form of orders, shipments and returns in the financial system such as stock ticks orders etcetera, the web site such as the page views clicks searches and IOT that means sensor readings and so on. So, all these are the data, which is called the streaming data which is required to be input into the systems, which is shown here in this particular diagram that the, the input streams, are stream data which basically, is in the form of the sales and the shipment details also, is a streaming form of data, than various price adjustments reordering and, and inventory adjustments. So, all these are some of the data system which is out of the retail system similarly, in the financial systems the stop ticks, different stock orders that will be the stream data, which is required to be computed, as an event, for the different business application, similarly as far as the website, is concerned page views page clicks searches on the website also will generate the stream, of the data as an event, IOT devices, wherein the sensors are reading taking the readings and now generating, the events streams, out of this IOT sensor, data is also one of the streaming data platform and is supported in the Kafka.

Refer slide time :(14:03)

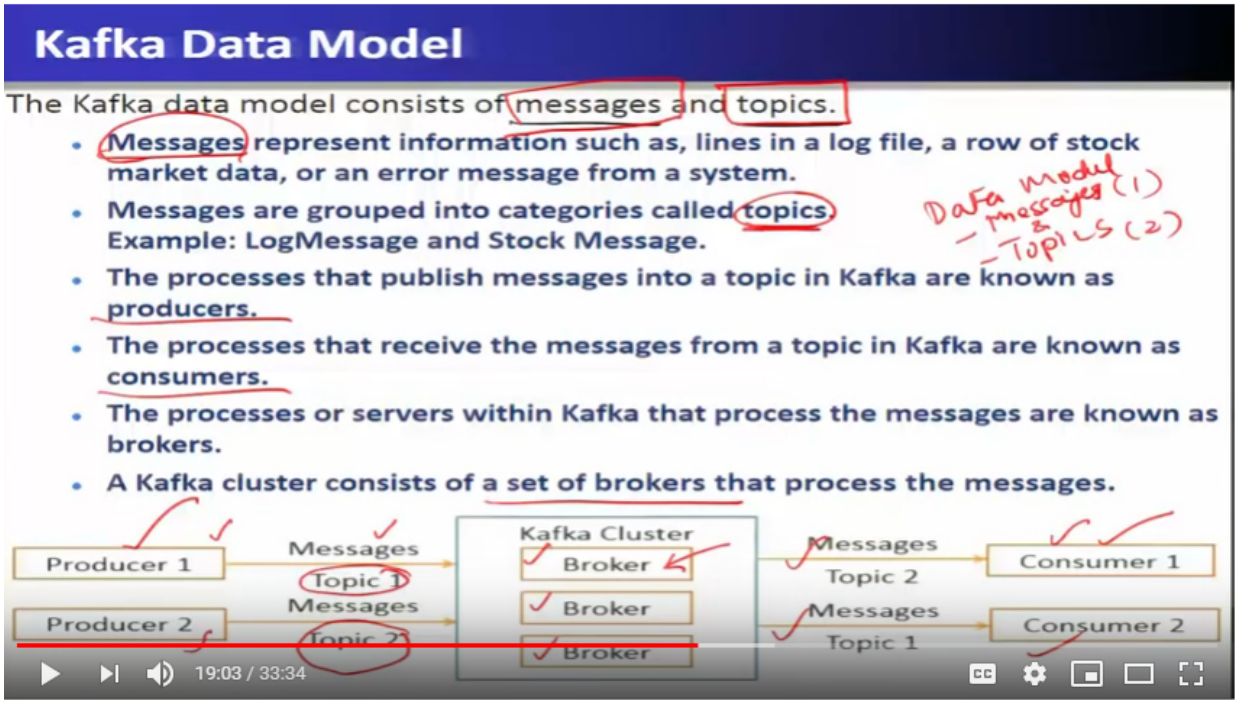
So, the Kafka, we see here more than 1000different companies worldwide, uses Kafka in some form or the others, such as Netflix, eBay, Adobe, Cisco and so on, all these big data companies they uses the Kafka, in some form of analysis and computations.

Refer slide time :(14:28)



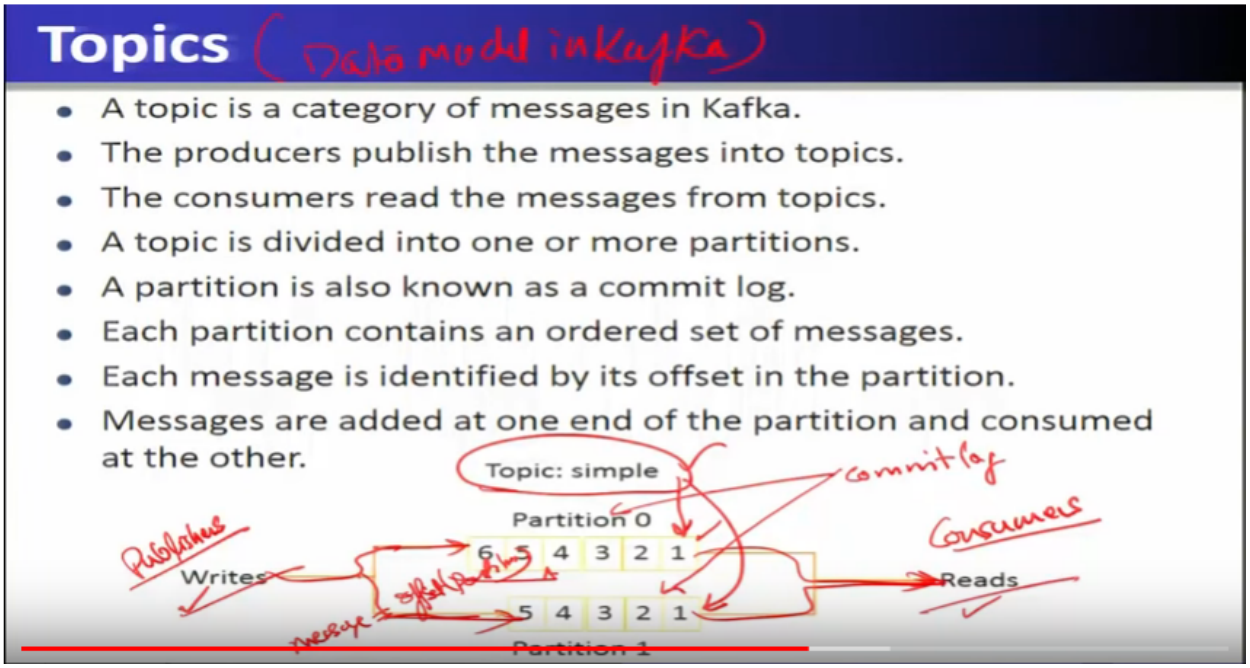
So, let us see some more Kafka example so, aggregating user's activity using the Kafka as an example. So, Kafka can be used to aggregate the user activity data such as clicks navigation searches from different website of an organization, such user activities can be sent to a real-time monitoring system and Hadoop for. So, we can see here as an example, of user activities aggregating user activities that means different users who are accessing through different portals, will generate the stream of data and that will enter into the Kafka cluster and Kafka will after computation, now it will give it to the consumers either, for the real-time monitoring or it will store in the Hadoop offline file process.

Refer slide time :(15:21)



So, Kafka data model consists, of the messages and the topics. So, messages represent the information, such as lines in a log file rows of a stock market data or an error message from a system they are called,' Messages'. So, messages are sometimes groups, together into the, into the topics. So, for example log message or a stock, message together, are grouped together and called as the,' Topics'. So, the processes that publish messages, into the, into the topic, in Kafka is known as the producers. And the processes that receive the message from the topic in the Kafka is known as the consumer, the processes or the servers within the Kafka that processes, the messages are known as, the Kafka brokers, in the Kafka cluster consists, of set of brokers that processes the messages, let us understand all these different components, of Kafka through this particular picture. So, the so, the group of messages, is called a,' Topic'. And so, here in this particular example, we have shown two different groups, of messages that is topic 1 and topic 2 these topics, are or these messages are generated and those by the processes, which we call it as producers. So, these particular topics, are now processed using things which is called the,' Brokers'. So,

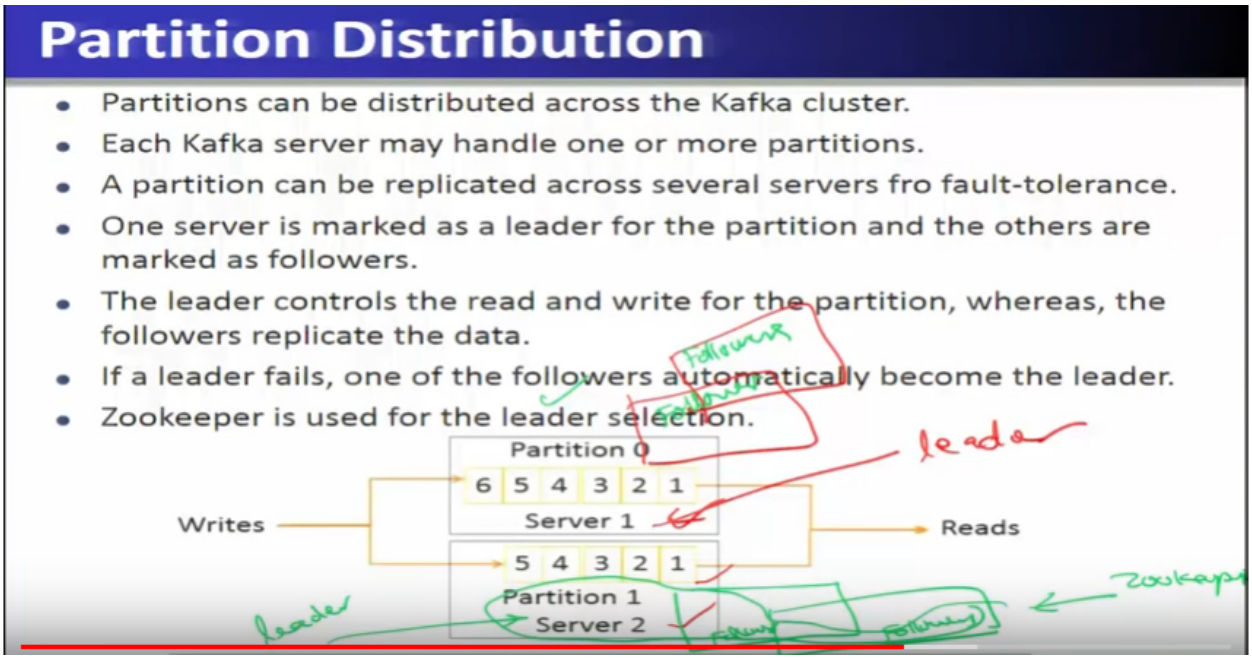
brokers process them and again these particular topics, after the transformation, will be passed on, to the consumers. So, again let us understand, that the data model, in Kafka is either messages and the collection or the group of messages, which is called the,' Topics'. So, data model in Kafka comprises, of messages and topics, now these messages represents, the information such as lines, in a log file row of a stock market data or error messages. So, these are called messages, these messages are grouped together and into the different categories and called the,' Topics'. So, related messages are grouped together and they called as,' Topics', furthermore the processes that publishes the messages, into the, into a topic in the Kafka is known as the producers. So, producers publishes, the that the messages into the, into the topic. So, the processes those receive the message that receives the messages, from the topics in the Kafka is known as the consumers. Now the processes and the servers within the Kafka that processes the messages are known as the brokers. So, a cosmic cluster consists of set of brokers that processes the messages. So, let us see what you mean by the, the topic, which is nothing but the data model,
Refer slide time :(19:13)



in Kafka so the topic is the category of the messages in Kafka. So, the producers published the messages into the topics and consumers, who read the messages from the topics. So, the topic is divided into one or more partitions and partition, is also known as the commit log. So, each partition contains an ordered set of messages. So, each message is identified by its offset in the partition and the messages are added at, one end of the partition and consumed at the other end of the partition. Let us understand the concept of topics and partition, by this particularly simple example. So, here the, the topics are nothing but the, the

publishers, publishes the messages into the topics and once they do this then they have to write down. So, the publishers, they will write they will do the write operations on the topic and the consumers will read the messages, out of from the topic. So, these operations, write is related to the publisher read operation is related to the consumer and the topic is divided into one or more partition, for example this is the topic a simple, it is divided into two partitions partition 1 and partition 2 .So, each partition is known as the commit log. So, partition is nothing but a commit log. So, each partition contains an ordered set0 of messages that we see the order in which the messages are received so; each message is identified, by its offset in two. So, each message is identified by, its offset in the in the partition, message is nothing but, an offset into the partition. So, messages are added at, one end of the partition like here it will be added and consumed at the other end of the partition, that is added with the help of using the write command and consumed, with the help of read command having understood, the topics the partition, the offset the write and read the publisher and consumer let us understand the partition in more details.

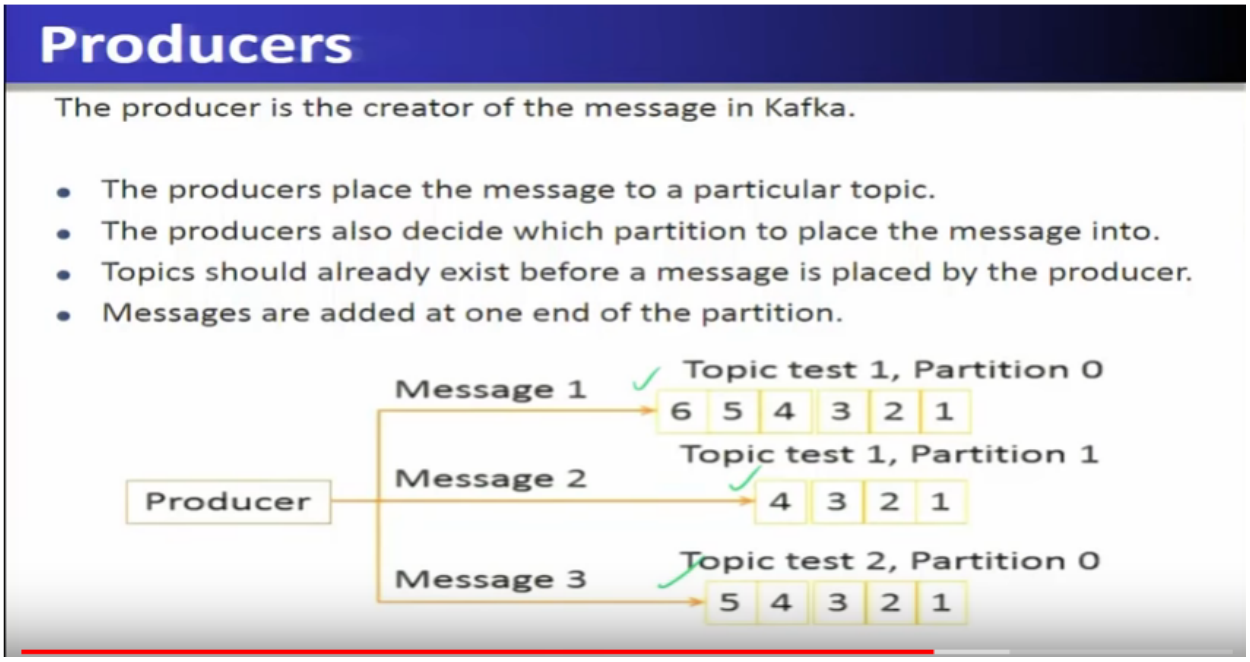
Refer slide time :(22:07)



So, topics are divided into, the partitions, which are the unit of parallelism, in Kafka. So, partitions allow the messages, in a topic to be distributed, allow the messages in atopic to be distributed to multiple servers. So, a topic can have any number of partitions and each partition should fit in a single Kafka cluster. And the number of partition decides the parallelism of that particular topic. So, partition distribution, partition can be distributed across Kafka cluster and each Kafka server, may handle one or more, partition and here in this example, which is shown that this server one, will handle partition 0 and any other server- we'll handle the partition 1 are both partition 0 and 1 can be handled by one server. So, each a partition, can be replicated across several servers for fault tolerance. So, one server is marked, as the leader for the partition others are marked as the followers. So, reader controls the read and write operation, for the partition whereas the followers which replicate the data. Now if the leader fails, one of

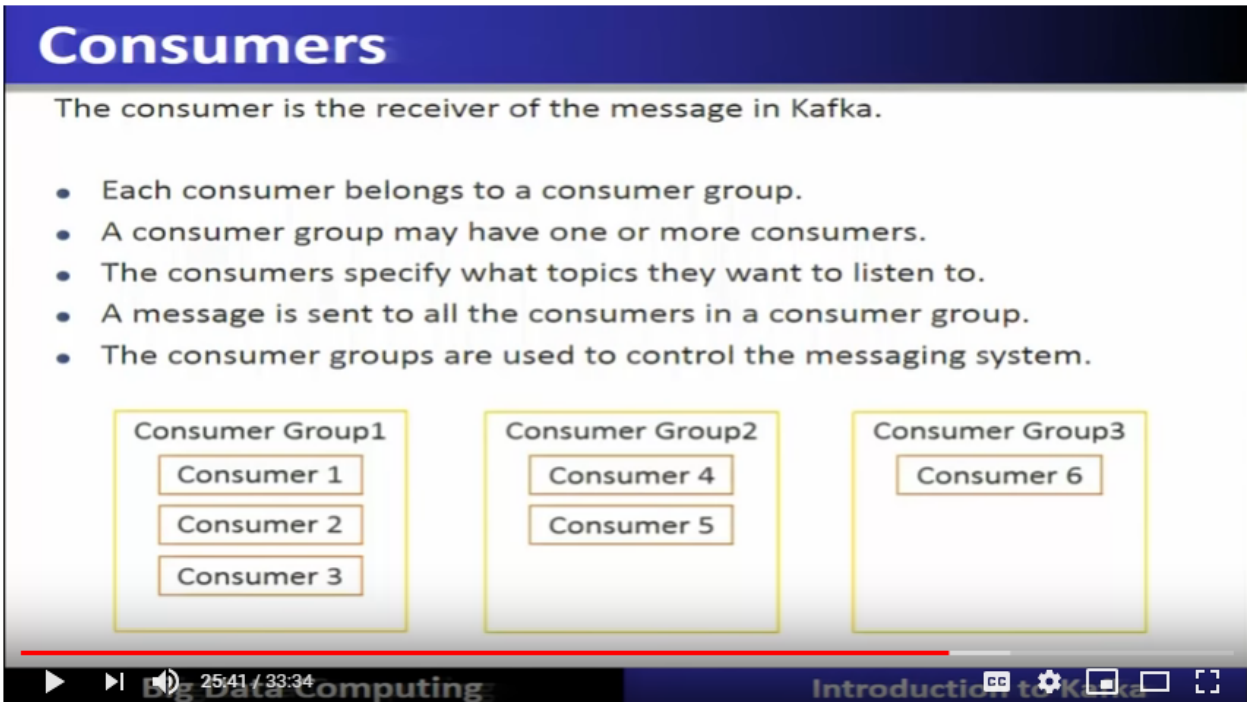
the followers automatically becomes, the leader zookeeper is used to, to do the leader selection, process here to maintain. So, here which is shown as the server this is, is nothing but a leader and the same partitions are replicated, in the other servers also and they are being replicated and called as the followers, similarly this particular, server is nothing but the leader and same partition is replicated, on the other servers which are called,' Followers'. And this particular leader election, across different followers, is done by the zookeeper, this is called,' Partition Distribution'.

Refer slide time :(24:30)



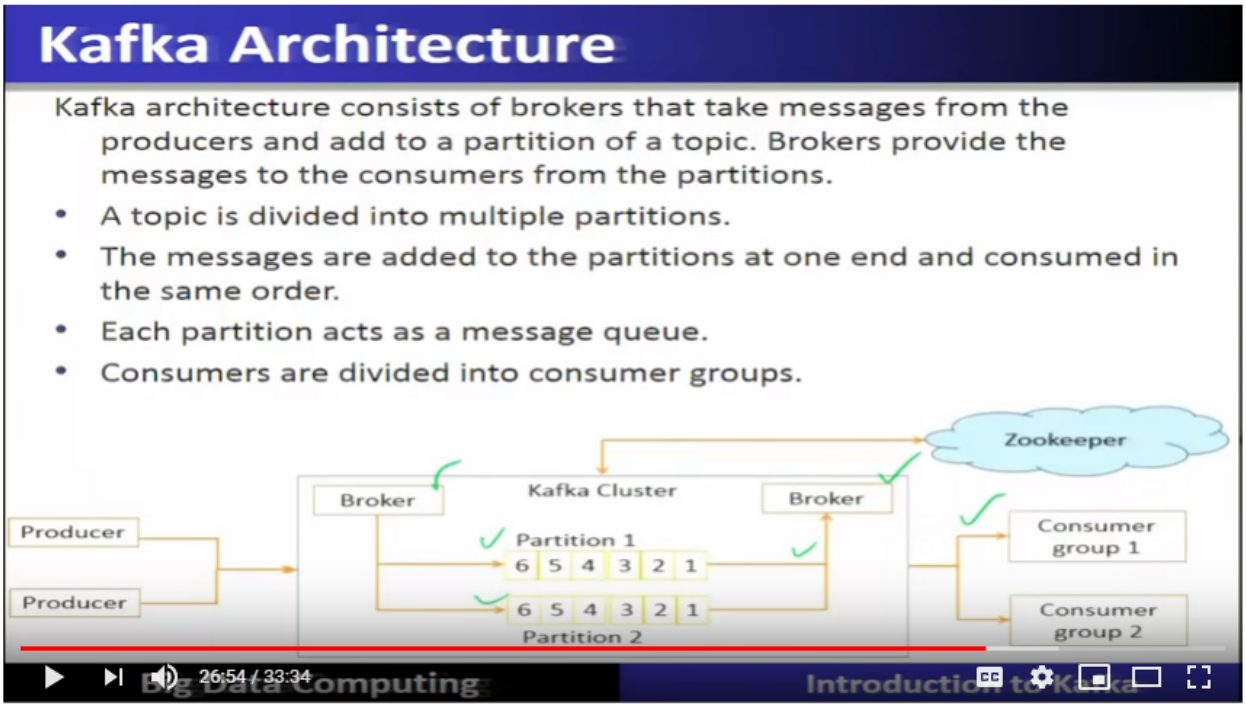
Now let us see the producers. So, producer is the creator of message in Kafka. So, the producer plays the message into, a particular topic so producers also decide which partition to place the message into it. And the topic should already exist, before a message is placed by the producer messages are added at, one end of the topic, one end of the partition. So, this is the example which is shown here that the producer, is producing the messages and different messages are now joining, the different topics, depending upon the topics, which are to be partitioned. So, here the producer will add the messages, to different topics.

Refer slide time :(25:21)



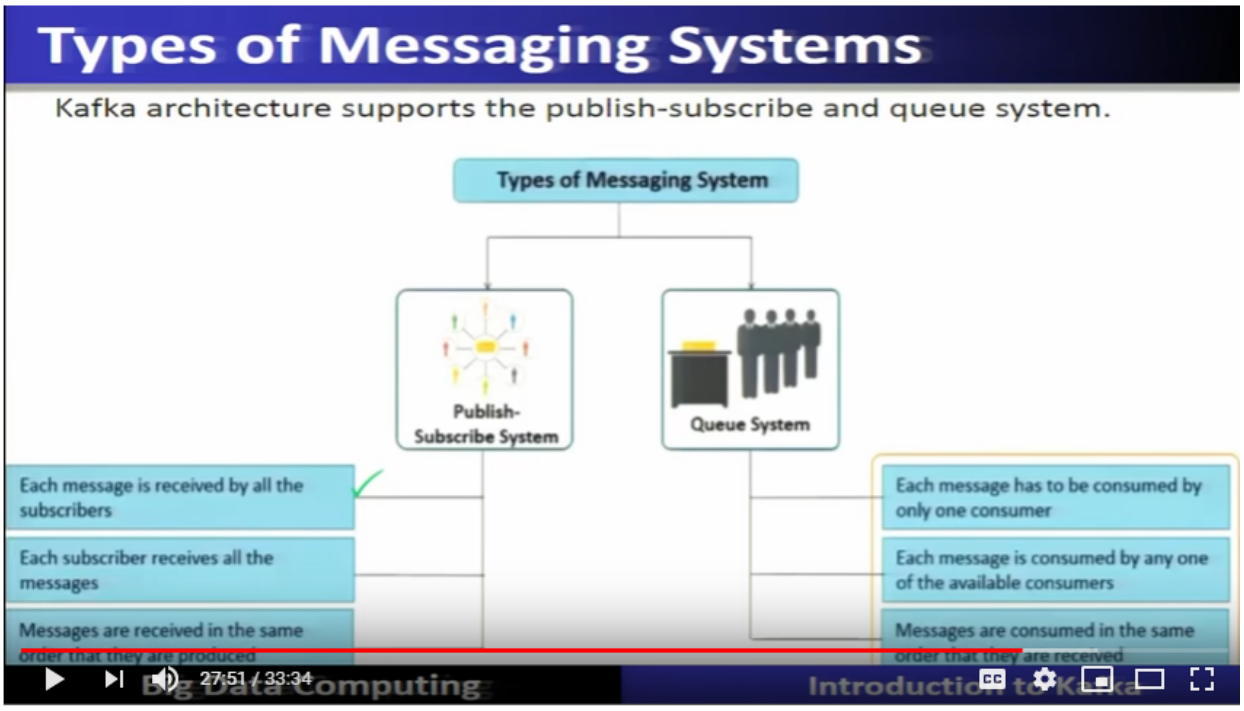
And consumer is the receiver of the message in the in the in the Kafka. So, each consumer belongs to, a consumer group, of consumer group a how one or more consumers. So, consumers specify, what topic they want to listen and the message is sent, to all the consumers in a particular, consumer group, consumer groups are used to control the messaging system.

Refer slide time :(25:41)



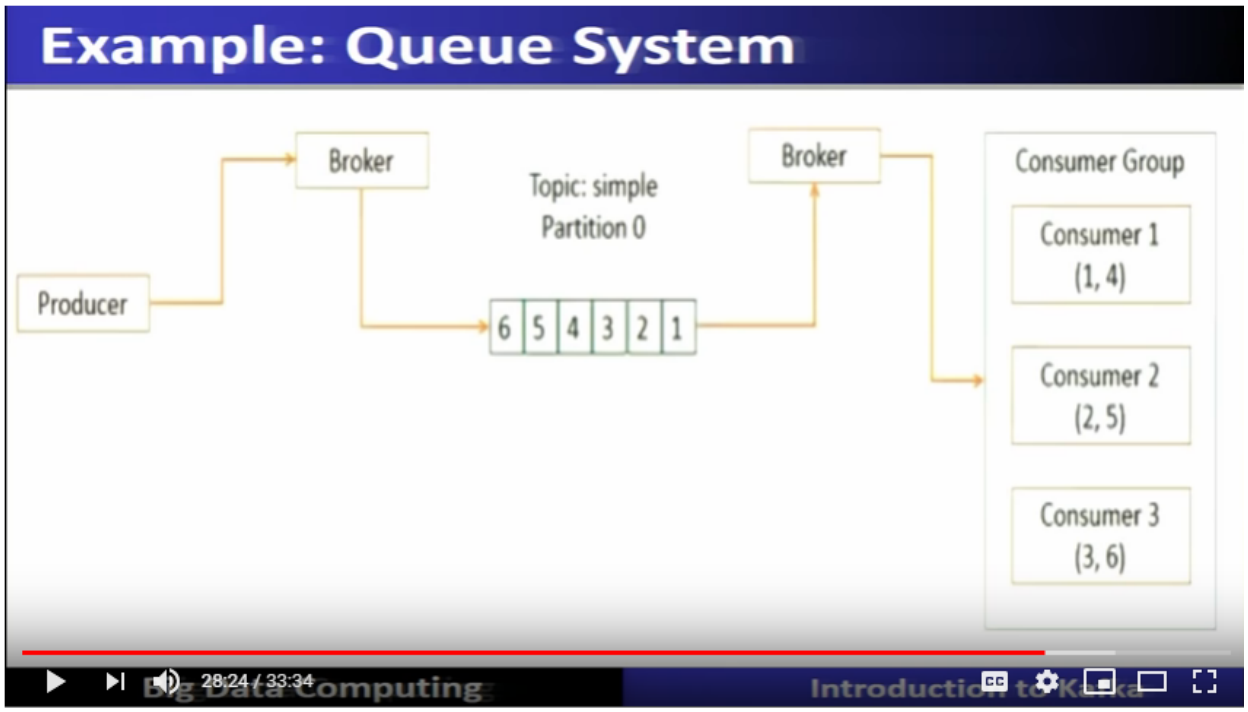
Now let us see all these, component together and see the overall Kafka architecture. So, Kafka architecture comprises, of the brokers that take the messages from the producer and add to the partition, of a topic broker provides the messages to the consumer from the partition. So, a topic is divided, into multiple partitions. So, the message is added to the partition at one end and consumed, at the other end that we have already seen and that is being coordinated by or that is handled by the brokers. So, each partition, willet as the message queue and the consumers are divided, into the consumer groups, here in this scenario, which is shown over here. And all the brokers, together are called the,' Kafka Clusters'. And the different server machines are used with the help of zookeeper in this scenario which is shown.

Refer slide time :(26:59)



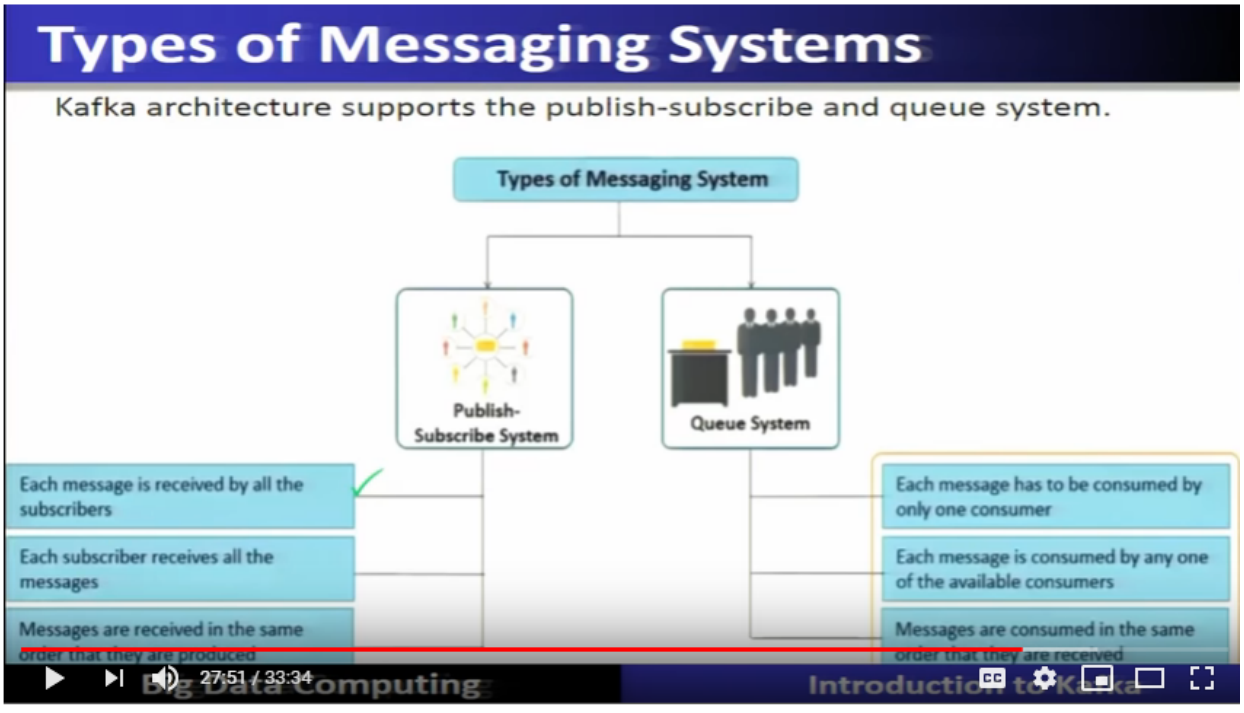
Now different type of messaging systems, are available as on date. So, Kafka architecture supports publish subscribe and queuing system. So, publish subscribe system on the left is shown, that each message is received by, all the subscribers that is are the consumers. So, each subscriber receives, all the messages and the messages are received in the same order that are in the same order that they are being produced by, in the publish subscribe system. Whereas in the queuing system, each message is consumed by only one consumer and each message is consumed by any one of the available consumers and messages are consumed, in the order in which they are removed. So, here there is a difference, between queuing and publish subscribe system, both the models of messaging system, is supported in the Kafka architecture.

Refer slide time :(27:56)



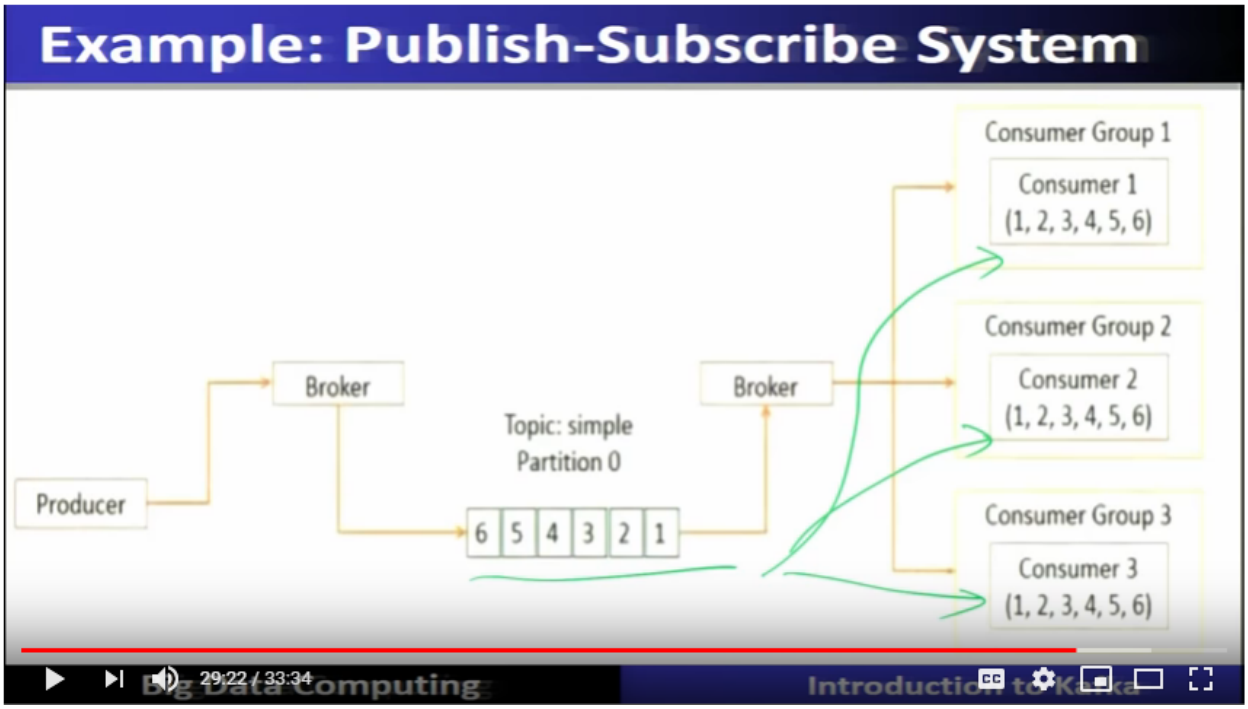
Now let us see the Queen queuing, queue system which is supported by the Kafka. So, here the producer, will produce, the message and this will be put into the partition or it will be divided into the topic, with the help of the broker and a particular consumer, will consume this particular message, in this particular system.

Refer slide time :(28:24)



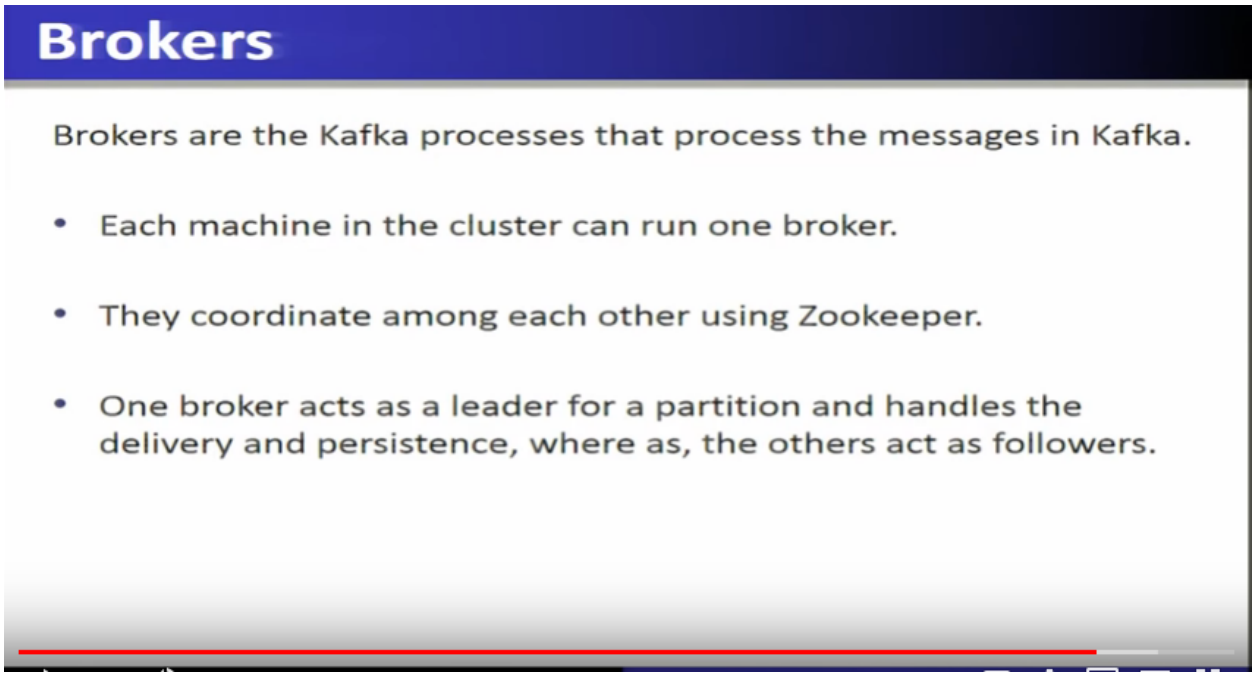
So, we have to see that each message has to be consumed by only one, consumer and each message is consumed by any one of the available consumers. So, and the messages are consumed in the same order that they are being received.

Refer slide time :(28:35)



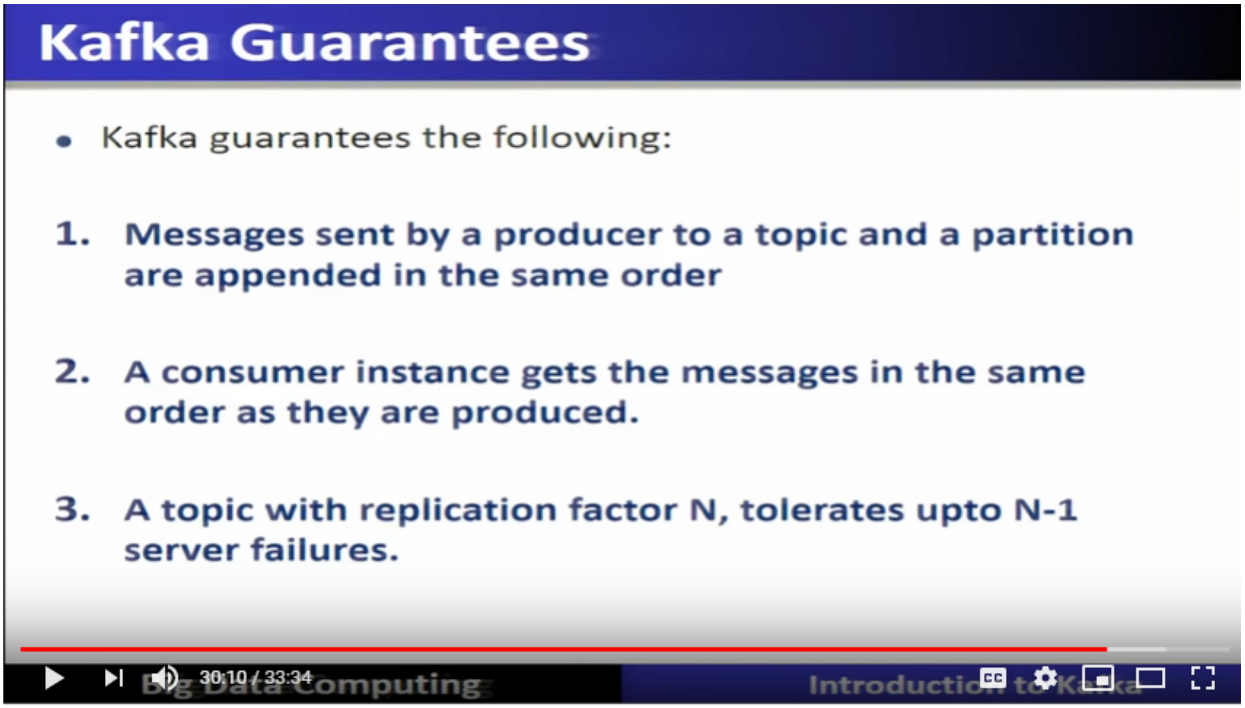
So, here you can see that consumer number one is available. So, it will consume my sage 1 and 4similarly consumer number 2 is now when it is available it will consume 2 and 5and when consumer 3, is available it will consume 3 and 6. So, any of these consumers will consume and every message is consumed by only one consumer. So, that is called the,' Queuing System'. And it is being supported over here, whereas in publish subscribe system, every message is received by all these consumers. So, you see here in this publish subscribe system so, this message 1 2 3 4 5 6 is being consumed in the same order, by all the consumer groups.

Refer slide time :(29:27)



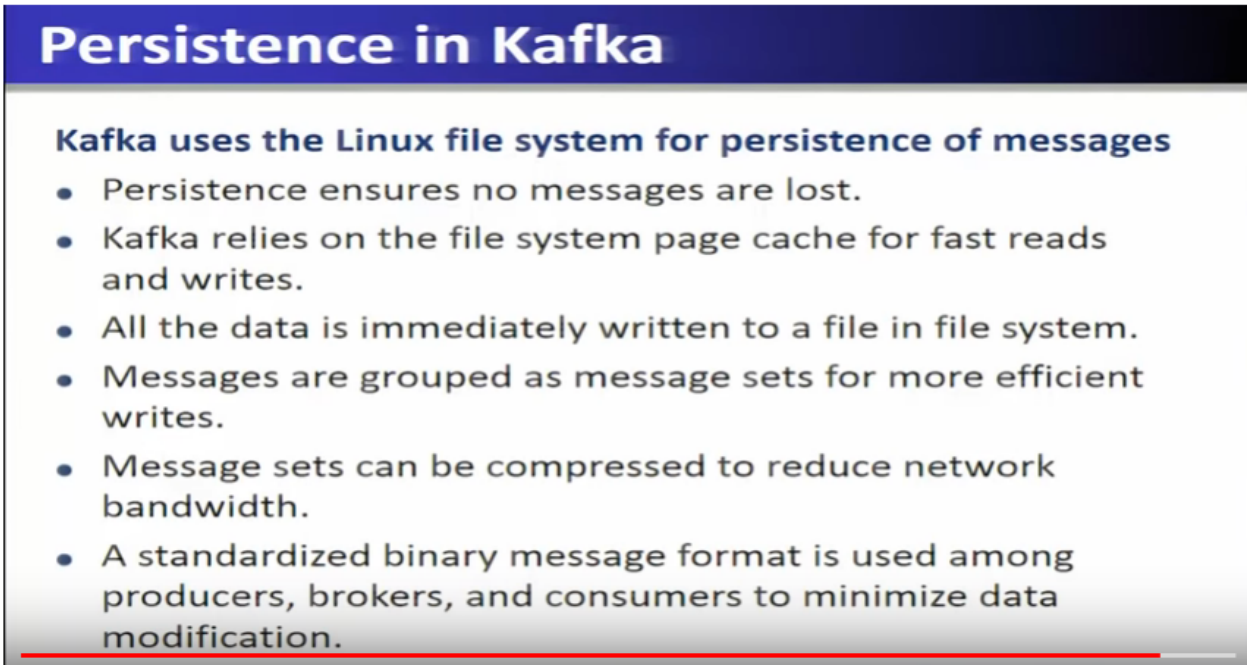
Now as far as the brokers are concerned, brokers are the Kafka processes that process the messages, in Kafka each message each machine, in the Kafka in the cluster can run one broker. So, they coordinate among each other using zookeeper, one broker is act as the leader, for the partition and handles the delivery and persistence whereas others are act as the followers.

Refer slide time :(29:50)



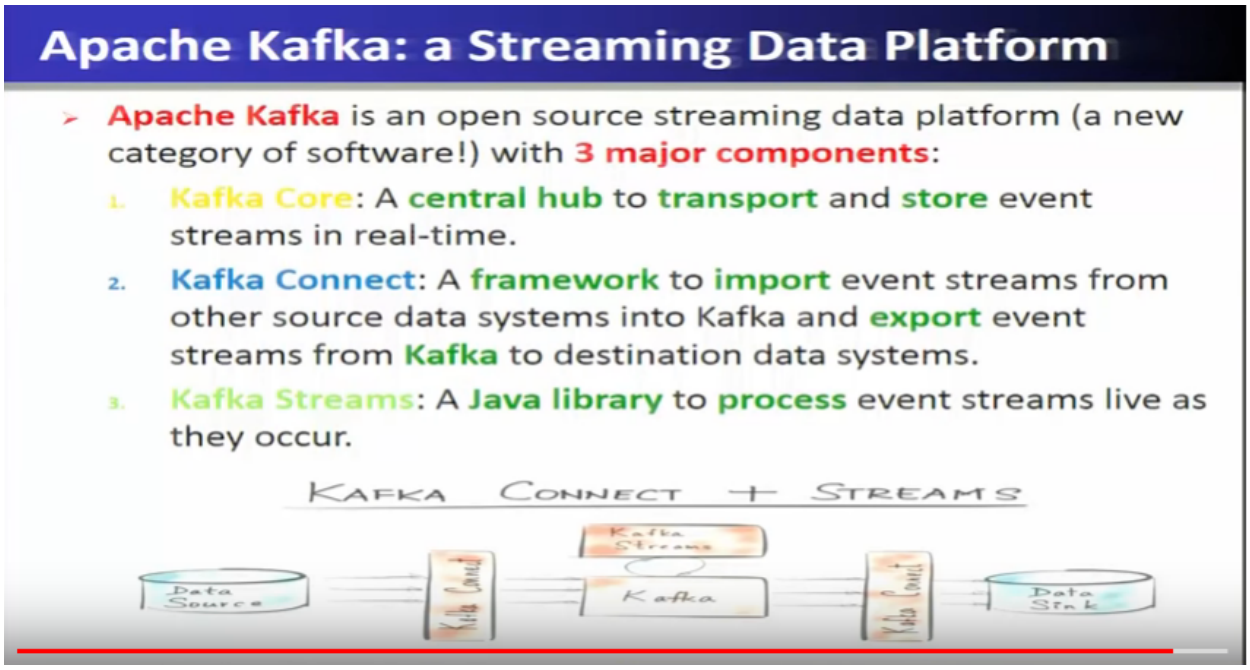
So, Kafka guarantees, the following first is the message is sent by the producer, to a particular topic and partitions are appended in the same order. Second guarantees is the consumer, instance gets the message in the same order that they are being produced. Third guarantees is that a topic with the replication, factor and all rates up to n minus one failures.

Refer slide time :(30:15)



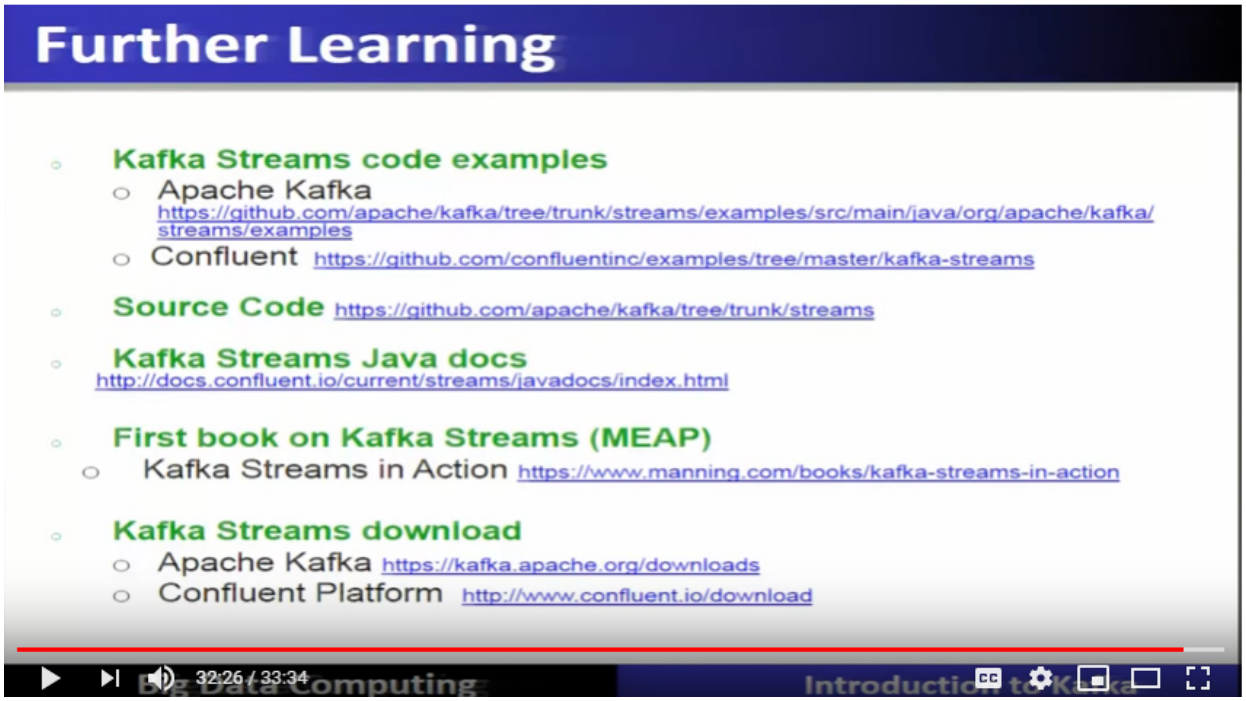
Now replication in Kafka .so, Kafka uses primary backup method for the replication that is one machine, which is one replica is called the leader and chosen as the primary and the remaining machine are the replicas, chosen as the followers and X as the backup. So, the leader propagates the right to the followers and the leader waits until all the rights are completed, on the on the replicas if the if the replica is down it is skipped, for the right until it comes back, if the leader fails, one of the followers will be chosen as the new leader and this mechanism can tolerate n minus one failures, if the replicas of their application factor is n. Now persistence in Kafka so, Kafka uses Linux file, system for persistence of the messages. So, persistence ensure no messages are lost Kafka relies on the file system page caching for fast reads and writes all the messages is immediately written, to a file in the file system and messages are grouped as the messages, set for the most efficient writes. So, messages sets can be compressed, to reduce the network, bandwidth a standard battery message format is used among the producers, brokers, consumers, to minimize the data modification.

Refer slide time :(31:32)



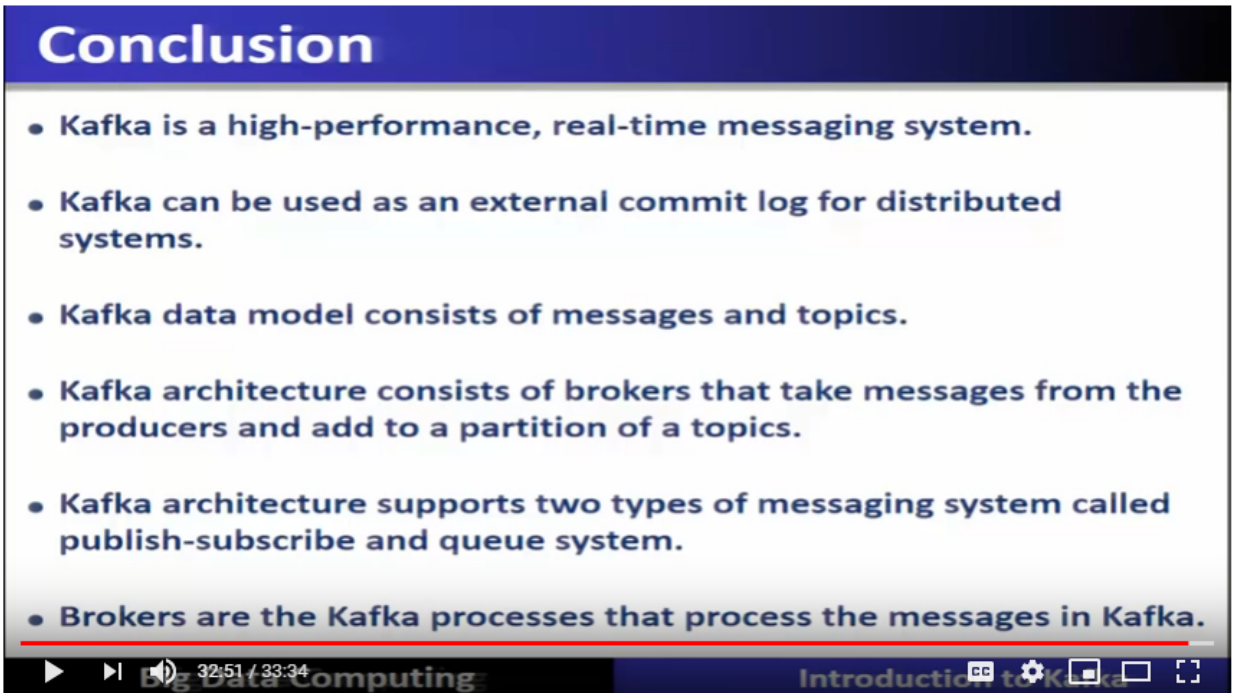
So, again in a nutshell let us see, how the streaming data platform is supported in Kafka. So, Apache capitals and open source streaming, data platform and its suppose three with the three major components, Kafka core is a central hub to transport and store, the event streams in a real-time Kafka connect, well now is the framework to import the event streams from, other sources into the Kafka and export the event streams from, Kafka to the destination data systems. Kafka stream is Java library to process, event streams,

Refer slide time :(32:07)



live as they occur iron for further reading, this is given here the references, for Kafka streams code, examples and Kafka stream java docsiron first book on Kafka streams. And Kafka streams download, is given at HTTPS Kafka dot, Apache dot, org downloads.

Refer slide time :(32:07)



Conclusion Kafka is high-performance, real-time messaging system, Kafka can be used as an external commit law, for distributed system Kafka data model consists of messages and topics Kafka architecture, consists of brokers, that take messages from producers. And to add to the partition of a topic. Kafka architecture supports two types of messaging system, called,' Publish Subscribe'. And queuing system brokers are the Kafka processes, that process the messages in the Kafka. Thank you.