

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



Course Content

Presentation

Case Study I - Publis...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

Personal Library

Presentation

3 / 118



COURSE OUTLINE //



MODULE 09

Introduction to Big Data Hadoop and Spark

Introduction to Python for Apache Spark

Functions, OOPs, Modules in Python

Deep Dive into Apache Spark Framework

Playing with Spark RDDs

Data Frames and Spark SQL

Machine Learning using Spark MLlib

Deep Dive into Spark MLlib

Understanding Apache Kafka and Apache Flume

Apache Spark Streaming - Processing Multiple Batches

Apache Spark Streaming - Data Sources

Implementing an End-to-End Project

My Classroom > PySpark Certification Training Course

X PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 4 / 118 ← →

Recap

What is Regression?

Regression is a process used to find the relationship between the independent variable and the dependent variable. It is used to predict the outcome based on the input variables.

Simple Linear Regression

Y = a + bx

The Logistic Regression Curve

Decision Tree

What is Random Forest?

What is Clustering?

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:36 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 5 / 118

Topics

- Why is Kafka needed ?
- What is Kafka ?
- Kafka features and components
- Kafka Architecture
- Kafka Scaling and Cluster
- Flume
- Flume Architecture and Features
- Flume Dataflows and Configuration



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:36 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 6 / 118

Objectives

After completing this module, you should be able to:

- Understand the need for Kafka
- Describe what is Kafka
- Explain Core Concepts of Kafka
- Understand Kafka Architecture
- Understand the components of Kafka Cluster
- Configuring Kafka Cluster
- Describe what is Apache Flume?
- Describe Basic Flume Architecture
- Explain Flume Configuration and Integrate Apache Flume and Apache Kafka



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Personal Library

Presentation

7 / 118

Why is Kafka Needed?

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

1 1 1 L X E M D 12:36 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

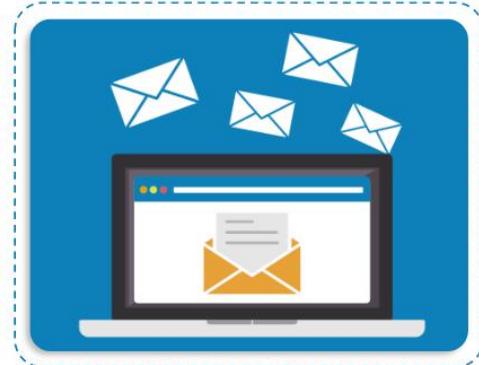
Presentation 8 / 118

Challenges with Real-Time Data

Problem:

There are two challenges in the present real-time data:

- First is to collect the data as data is huge in amount and,
- Second is to analyse real-time data. This analysis typically includes the following types of data and much more:
 - User behaviour data
 - Application performance tracing
 - Activity data in the form of logs
 - Event messages



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:36 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 9 / 118

Solution - Messaging Systems

Problem:

There are two challenges in the present real-time data:

- First is to collect the data as data is huge in amount and,
- Second is to analyse real-time data. This analysis typically includes the following types of data and much more:
 - User behaviour data
 - Application performance tracing
 - Activity data in the form of logs
 - Event messages

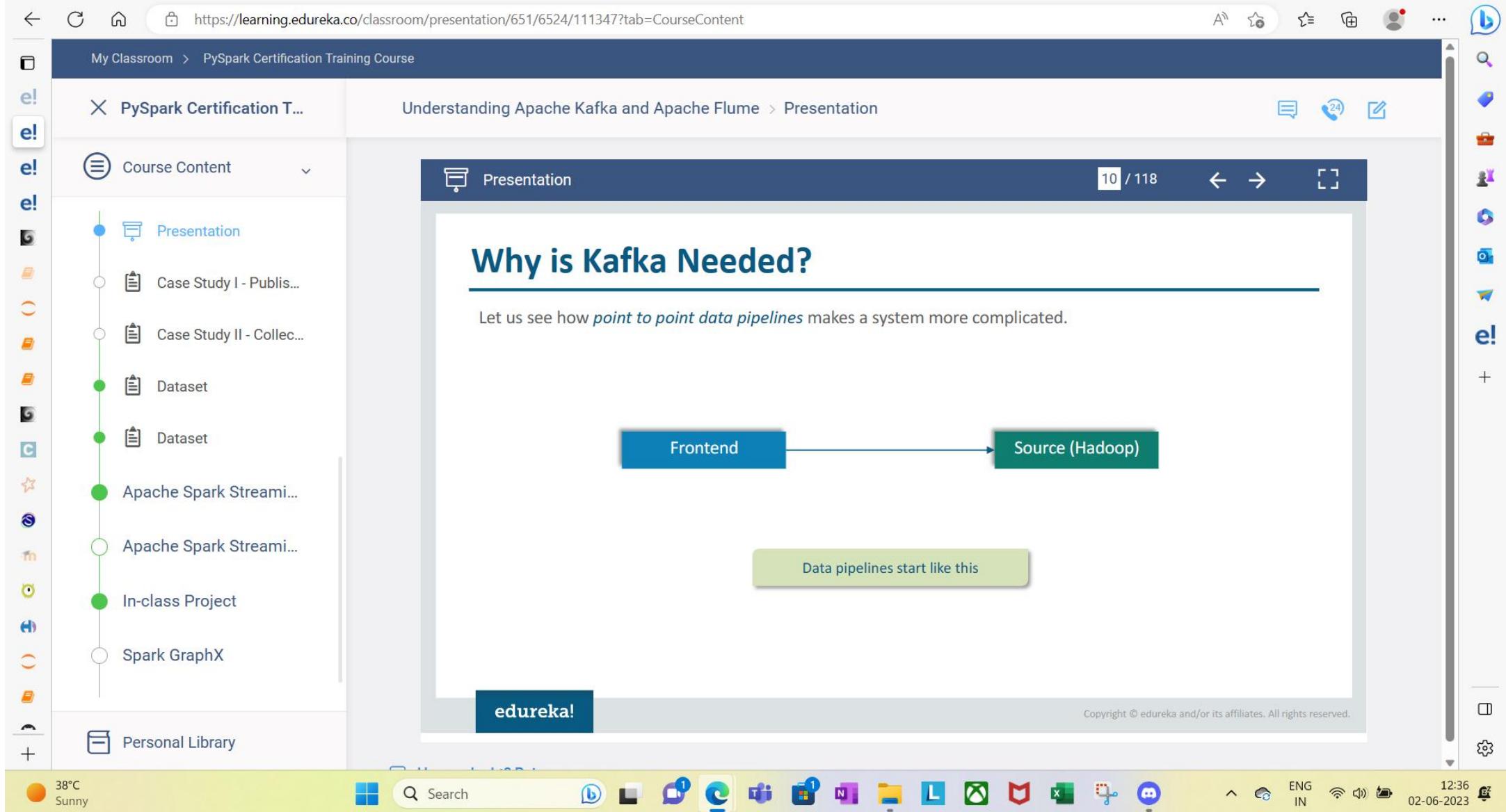
Solution:

i One way to solve this problem is by **using messaging systems**, messaging systems provide seamless integration among distributed applications with the help of messages, that are shared between them.



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.



My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 11 / 118

Point to Point Data Pipelines

Then we reuse these data pipelines

```
graph LR; Frontend1[Frontend] --> Hadoop[Hadoop]; Frontend2[Frontend] --> Hadoop; Frontend3[Frontend] --> Hadoop; Service[Service] --> Hadoop;
```

edureka!

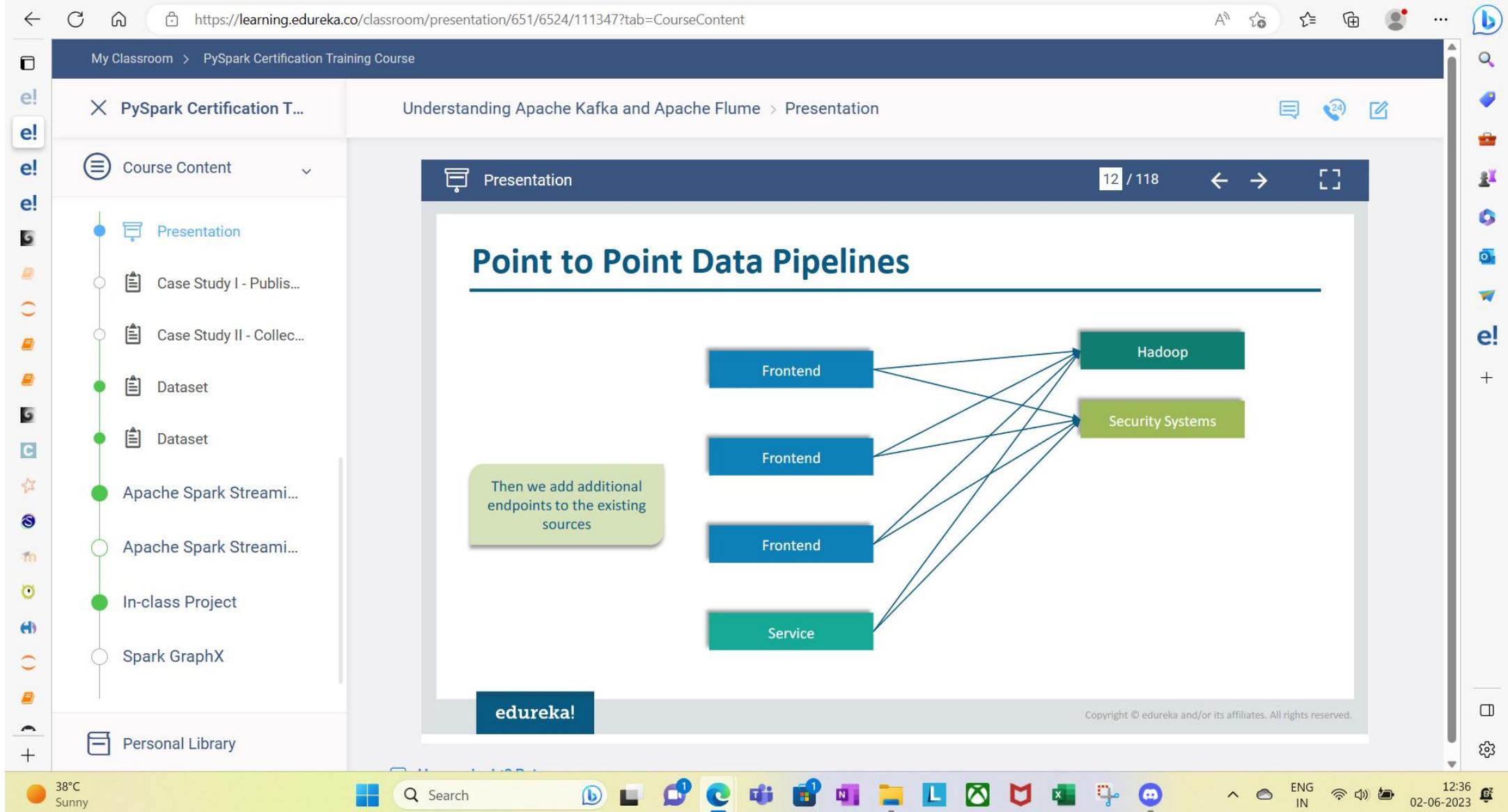
Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

L

12:36 02-06-2023



My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 13 / 118 ← →

Point to Point Data Pipelines

Then it starts to look like as it is shown in the picture and system becomes very complicated.

```
graph TD; DW[Data Warehouse] --> F1[Frontend]; DW --> F2[Frontend]; DW --> F3[Frontend]; DW --> F4[Frontend]; F1 --> H[Hadoop]; F1 --> SS[Security Systems]; F1 --> RT[Real-time monitoring]; F1 --> OS[Other Services]; F2 --> H; F2 --> SS; F2 --> RT; F2 --> OS; F3 --> H; F3 --> SS; F3 --> RT; F3 --> OS; F4 --> H; F4 --> SS; F4 --> RT; F4 --> OS;
```

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:36 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

L

12:36 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 15 / 118 ← →

Kafka Decouples Data Pipelines

```
graph TD; F1[Frontend] --> K[kafka]; F2[Frontend] --> K; F3[Frontend] --> K; S[Service] --> K; K --> HC[Hadoop Clusters]; K --> SS[Security Systems]; K --> RT[Real-time monitoring]; K --> OS[Other services]; K --> DW[Data Warehouse]
```

Kafka decouples Data Pipelines

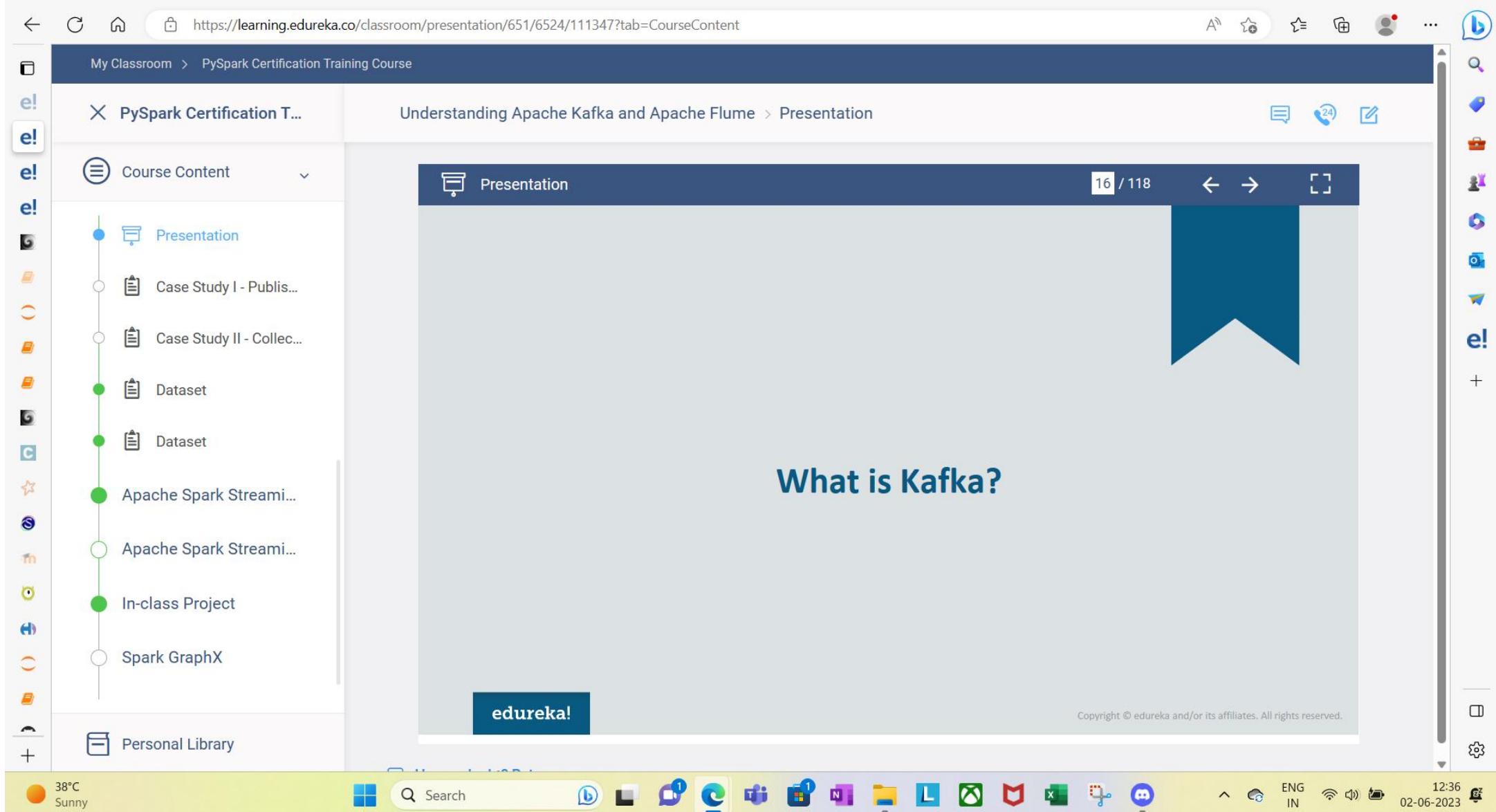
edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:36 02-06-2023



My Classroom > PySpark Certification Training Course

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

e! Presentation

e! Case Study I - Publis...

e! Case Study II - Collect...

e! Dataset

e! Dataset

e! Apache Spark Streami...

e! Apache Spark Streami...

e! In-class Project

e! Spark GraphX

e! Personal Library

What is Kafka?

- **Apache Kafka** is a distributed publish-subscribe messaging system.
- Originally developed at LinkedIn and later on became a part of Apache Project.
- Kafka is fast, scalable, durable, fault-tolerant and distributed by design.



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

edureka!

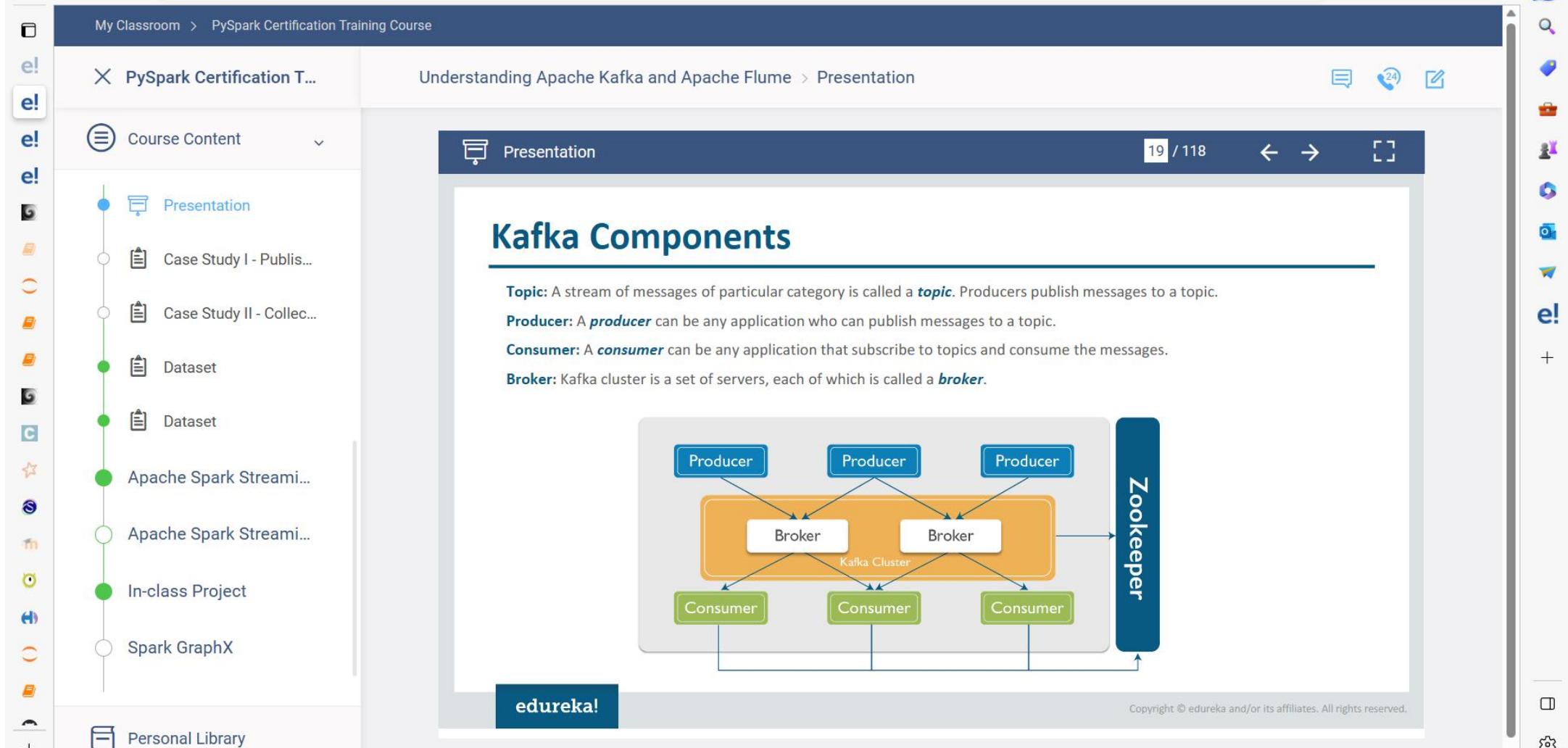
Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

L

12:36 02-06-2023



https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 20 / 118 ← →

Apache Kafka Overview

The diagram illustrates the Apache Kafka architecture and its integration with various systems:

- Kafka Cluster**: The central component.
- Producers**: Publish records to topics in Kafka.
- Consumers**: Pull records stored in Kafka topics.
- Connectors**: For in/out movement of data between Kafka and other systems (DB).
- Stream Processors**: Client library for processing data stored in Kafka.
- DB**: Database components.
- Apps**: Represented by green, teal, and orange boxes, indicating different types of applications interacting with the Kafka Cluster.

Annotations provide additional context:

- "Publishes records to the topics in Kafka" (Producers)
- "For in/out movement of data between Kafka and other systems" (Connectors)
- "Client library for processing data stored in Kafka" (Stream Processors)
- "Pulls records stored in Kafka topics" (Consumers)

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:36 02-06-2023

My Classroom > PySpark Certification Training Course

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

Presentation

Case Study I - Publis...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

Personal Library

Presentation

21 / 118



Let's Take a Look at Kafka Features

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 22 / 118

Kafka Features

- High Throughput**
Provides support for hundreds of thousands of messages with modest hardware.
- Scalability**
Highly scalable distributed systems with no downtime.
- Data Loss**
Kafka with the proper configurations can ensure zero data loss.
- Stream Processing**
Kafka can be used along with real time streaming applications like spark and storm.
- Replication**
Messages can be replicated across cluster, which provides support for multiple subscribers.
- Durability**
Provides support for persistence of messages to disk which can be further used for batch consumption.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

1 1 1 L X E X M F 12:36 02-06-2023 ENG IN

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 23 / 118 ← →

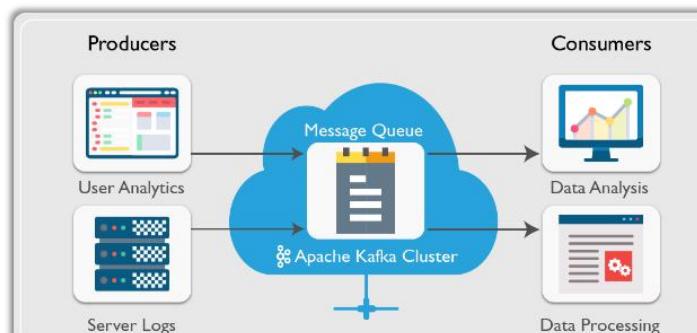
Common Use Cases for Kafka

Website activity tracking: The web application sends events such as page views where they become available for real-time processing, offline analytics in Hadoop.

Operational metrics: Alerting and reporting on operational metrics.

Log aggregation: Kafka can be used to collect logs from multiple services.

Stream processing: A framework such as Spark Streaming reads data from a topic, processes it and writes processed data to a new topic where it becomes available for users and applications.



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:36 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 24 / 118 ← →

Kafka Growth Exploding

- More than **1/3** of all Fortune 500 companies use **Kafka**.
- These companies includes the top ten travel companies, **7** of top ten banks, **8** of top ten insurance companies, **9** of top ten telecom companies.
- LinkedIn, Microsoft and Netflix** process billions of messages a day with Kafka (1,000,000,000,000).
- Kafka** is used for **real-time streams** of data & used to collect big data for **real time analysis**.

86% of respondents reported that the number of their systems that use Kafka is increasing

20% reported that the number is "growing a lot"

52% of organizations have at least 6 systems running Kafka.

Interest over time

Source: Google Trends

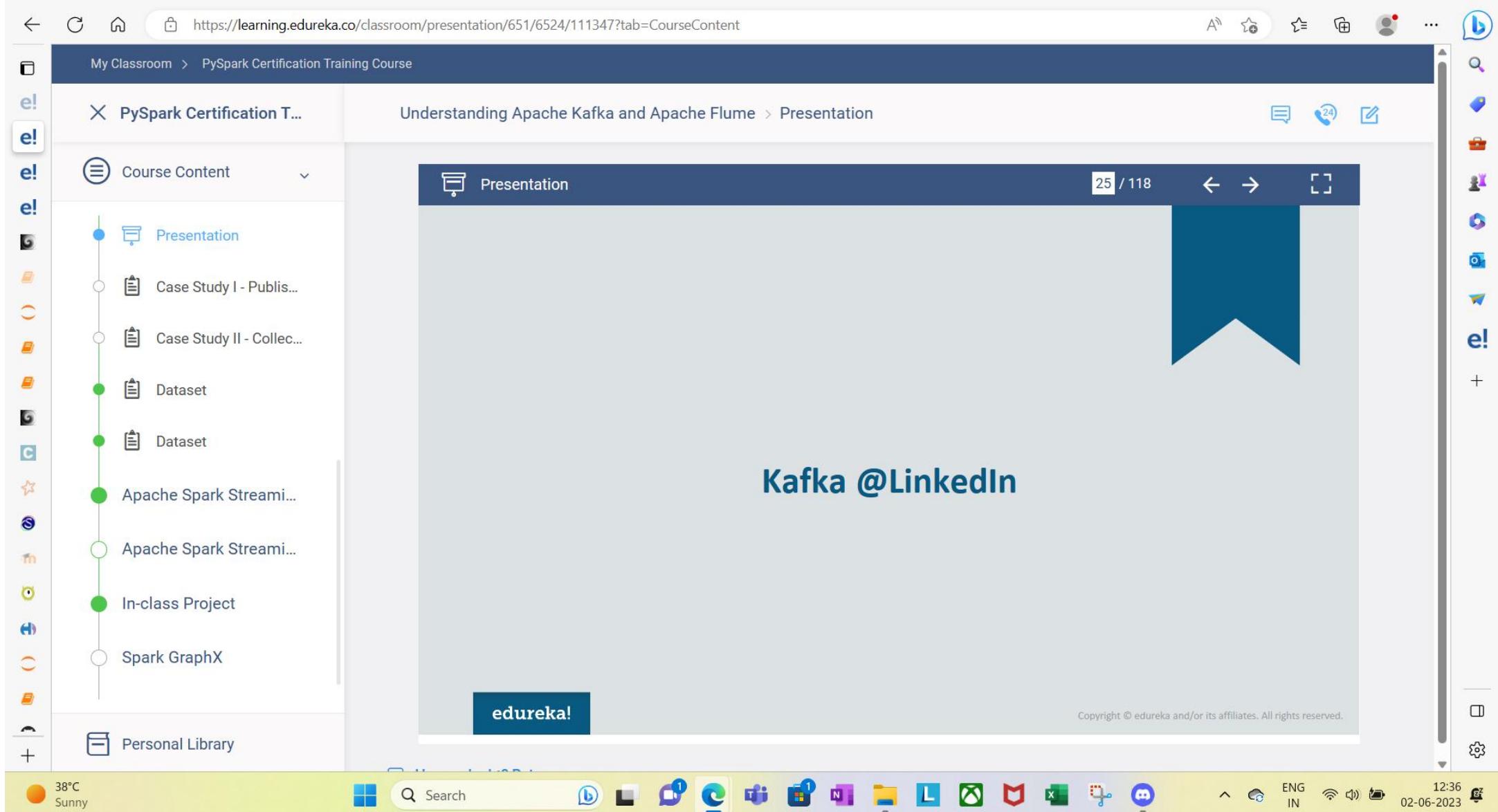
Copyright © edureka and/or its affiliates. All rights reserved.

edureka!

38°C Sunny

Search

12:36 02-06-2023



My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 26 / 118

Kafka @LinkedIn

1100+ commodity machines
31,000+ topics
350,000+ partitions

675 billion messages/day
150 TB/day in
580 TB/day out

Peak Load
10.5 million messages/sec
18.5 GB/sec Inbound
70.5 GB/sec Outbound

A modern stream-centric data architecture built around Kafka

Copyright © edureka and/or its affiliates. All rights reserved.

edureka!

38°C Sunny

Search

12:36 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 27 / 118

Kafka @LinkedIn

People in your network have new connections

Michael Hunger is now connected with: Mirko Fichtner Temalead Frontend Development at Movago GmbH 3h Connect

Luanne Misquitta is now connected with: Chandra Sekhara Rao Aluru ISB Biocon Certificate Programme in Business Analytics (Student) 7h Connect

Show more updates ▾

LinkedIn Newsfeed is powered by Kafka

edureka!

People Also Viewed

- Emil Eifrem Founder and CEO of Neo Technology
- Kenny Bastani Spring Developer Advocate at Pivotal
- Ryan Boyd Geek and Developer Relations at Neo4j
- Oliver Gierke Spring Data Project Lead at Pivotal
- Tobias Lindaaker Software Engineer and Technical Mentor at Neo Technology
- Mark Needham Field Engineer at Neo Technology

LinkedIn recommendations are powered by Kafka

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

edureka!

12:36 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 28 / 118

Kafka @LinkedIn

i Apart from this LinkedIn uses Kafka for many other purposes like log monitoring, performance metrics, search improvement etc.

edureka!

Notifications

- Srinivas Prasad K T likes your photo 1d
- Nicole Haywood Data Scientist at Feed Inc. is now a connection 1mth
- Srinivas Prasad K T endorsed you for 2 skills: Oracle Certified Java Programmer, Java 1mth
- Chetan Maheshwari likes your photo 2mth
- Samir Solanki published a new post: "SceneKit vs Threejs" 3mth

LinkedIn notifications are powered by Kafka

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

edureka!

12:36 02-06-2023

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



Course Content

Presentation

Case Study I - Publis...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

Personal Library

Presentation

29 / 118



Who Else Uses Kafka?



DataSift uses Kafka as a collector of monitoring events and to track user's consumption of data streams in real time.



Wooga uses Kafka to aggregate and process tracking data from all their Facebook games (hosted at various providers) in a central location.



Twitter uses Kafka as a part of its Storm – a stream processing infrastructure.



Spongecell uses Kafka to run their entire analytics and monitoring pipeline driving both real-time and ETL applications.



Loggly is the world's most popular cloud-based log management. It uses Kafka for log collection.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C
Sunny

Search



https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

edureka!

But what about other messaging systems?
e.g. ActiveMQ, RabbitMQ etc.

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:36 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 31 / 118 ← →

Comparing Messaging Systems

Kafka has a more efficient storage format.

On average, each message had an overhead of 9 bytes in Kafka, versus 144 bytes in ActiveMQ.

Copyright © edureka and/or its affiliates. All rights reserved.

edureka!

38°C Sunny

Search

12:36 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Comparing Messaging Systems

In both ActiveMQ and RabbitMQ brokers maintain delivery state of every message by writing to disk but in case of Kafka there is no disk write which makes it fast.

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/course/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

L

12:36 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 34 / 118 ← →

Recap : Kafka Architecture

The diagram illustrates the Kafka architecture. At the center is a cluster of three orange boxes labeled 'Broker'. Three blue boxes labeled 'Producer' have arrows pointing to the brokers. Three green boxes labeled 'Consumer' also have arrows pointing to the brokers. To the right of the brokers is a tall blue box labeled 'Zookeeper'. Arrows point from the brokers to the Zookeeper.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:36 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 35 / 118 ← →

Kafka Architecture - Deep Dive

The diagram illustrates the Kafka architecture. A central blue box labeled "Topic" contains three orange cylinders representing "Partition 1", "Partition 2", and "Partition 3". Each cylinder has three slots labeled 0, 1, and 2, each with a small envelope icon. Dashed arrows point from "Producer A" and "Producer B" to the partitions. Solid arrows point from the partitions to "Consumer A", "Consumer B", and "Consumer C".

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:36 02-06-2023

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



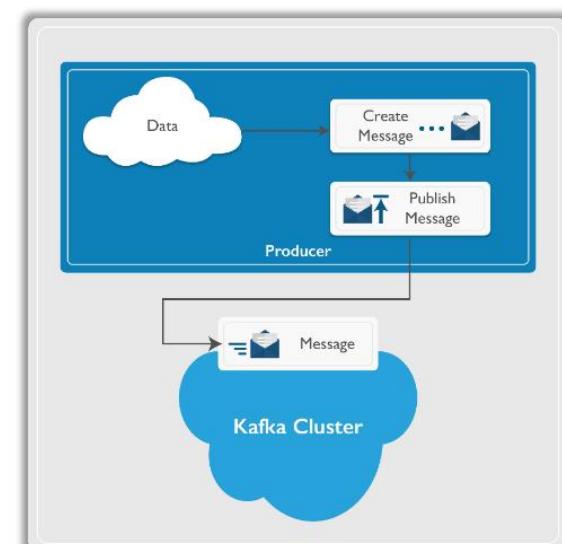
e! Course Content

- Presentation
- Case Study I - Publish...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

e! Personal Library

Kafka Producer

- Applications publishes messages to the topic in Kafka cluster.
- While writing messages, it is also possible to attach a key with the message.
- By attaching key the producers basically provide a guaranty that all messages with the same key will arrive in the same partition.
- Supports both *async* (less durable) and *sync* modes.
- Publishes as much messages as fast as the broker in a cluster can handle.



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Course Content Presentation Understanding Apache Kafka and Apache Flume > Presentation

Presentation 37 / 118

Kafka Broker

- Kafka cluster basically comprised of one or more servers.
- *Each of the servers in the cluster* is called a **broker**.
- *Handles hundreds of megabytes of writes from producers and reads from consumers.*
- *Retains all published messages irrespective of whether it is consumed or not.*
- If retention is configured for n days ,then messages once published it is available for consumptions for configured n days and thereafter it is discarded.
- *Works like a queue if consumer instances belong to same consumer group, else works like publish-subscribe.*

The diagram illustrates the Kafka Broker architecture. It shows three brokers (Broker 1, Broker 2, and Broker 3) each containing Topic-1 with three partitions (Partition-0, Partition-1, and Partition-2). Producers A and B are shown publishing messages to these partitions. Consumers are shown reading from the partitions. The diagram highlights that a single topic can have multiple partitions distributed across different brokers, and consumers can belong to the same consumer group to read from multiple partitions.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:36 02-06-2023

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

Presentation

Case Study I - Publis...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

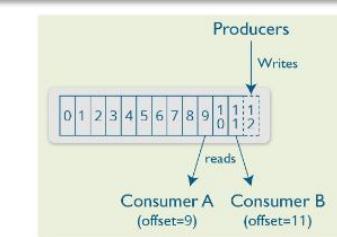
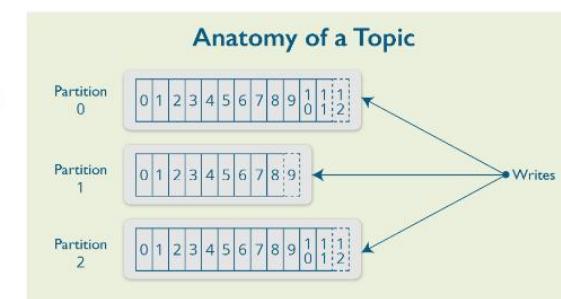
In-class Project

Spark GraphX

Personal Library

Topics and Partitions

- A *topic* is a *category* or *feed name* to which *records are published*.
- Topics are *broken up* into *ordered commit logs* called *partitions*.
- Each *message* in a *partition* is assigned a *sequential id* called an *offset*.
- Data is retained for a *configurable period of time*.
- Writes to a partition are generally *sequential* thereby *reducing* the *number of hard disk seeks*.
- Reading messages from partition can either be from *beginning* and also can *rewind* or *skip* to any *point* in partition by *supplying an offset value*.



edureka!

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



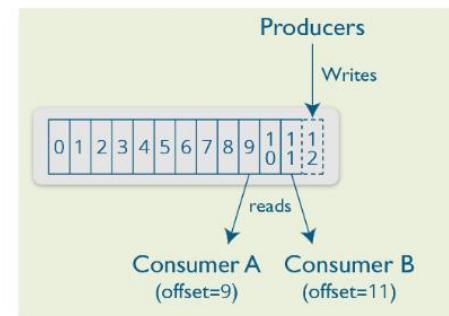
e! Course Content

- Presentation
- Case Study I - Publish...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

e! Personal Library

Message Ordering & Guarantees

- *Ordering is only guaranteed within a **partition for a topic**.*
- To ensure ordering:
 - *Group messages in a partition by key (producer).*
 - *Configure exactly one consumer instance per partition within a consumer group.*
- *Messages sent by a producer to a particular topic **partition** will be appended in the order they are sent.*
- A *consumer instance* sees messages in the *order they are stored in the log*.
- For a *topic with replication factor N*, Kafka can *tolerate up to N-1 server failures* without “losing” any messages committed to the log.



My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 40 / 118 ← →

Topics vs. Partitions vs. Replicas

A topic configured to use 4 partitions

Broker 1 Broker 2 Broker 3 Broker 4 Broker 5

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny Search

12:37 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 41 / 118 ← →

Topics vs. Partitions vs. Replicas

The diagram illustrates the relationship between Topics, Partitions, and Replicas. It shows five brokers labeled Broker 1 through Broker 5. Below them is a circular icon divided into four quadrants, each containing a number (0, 1, 2, 3). A callout box points to this icon with the text "A topic configured to use 4 partitions". Another callout box points to one of the quadrant numbers with the text "Each partition has an ID".

Broker 1 Broker 2 Broker 3 Broker 4 Broker 5

A topic configured to use 4 partitions

Each partition has an ID

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 42 / 118

Topics vs. Partitions vs. Replicas

A topic configured to use 4 partitions

Each partition has an ID

If say, replication factor of a topic is set to 3, then Kafka will create 3 identical replicas of each partition and place those replicas on available brokers in the cluster.

Copyright © edureka and/or its affiliates. All rights reserved.

edureka!

38°C Sunny

Search

12:37 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 43 / 118

Topics vs. Partitions vs. Replicas

A topic configured to use 4 partitions

Each partition has an ID

The ID of a replica is same as the ID of the broker that hosts it

If say, replication factor of a topic is set to 3, then Kafka will create 3 identical replicas of each partition and place those replicas on available brokers in the cluster.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 44 / 118 ← →

Topics vs. Partitions vs. Replicas

A topic configured to use 4 partitions

Each partition has an ID

The ID of a replica is same as the ID of the broker that hosts it

For each partition Kafka will elect one broker as the 'leader'

If say, replication factor of a topic is set to 3, then Kafka will create 3 identical replicas of each partition and place those replicas on available brokers in the cluster.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

My Classroom > PySpark Certification Training Course

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

e! Presentation

e! Case Study I - Publis...

e! Case Study II - Collect...

e! Dataset

e! Dataset

e! Apache Spark Streami...

e! Apache Spark Streami...

e! In-class Project

e! Spark GraphX

e! Personal Library

Kafka Broker Configurations

The essential configurations are the following:

- broker.id
- log.dirs.
- zookeeper.connect

Name	Description	Default Value
broker.id	Each broker is uniquely identified by a non-negative integer ID.	0
zookeeper.connect	This specifies the ZooKeeper's connection string in the hostname:port/chroot form. Here, chroot is a base directory that is prepended to all path operations.	localhost:2181
log.dirs	These are the directories in which the log data is stored.	/tmp/kafka-logs
default.replication.factor	default replication factors for automatically created topics	1
num.partitions	The default number of log partitions per topic	1
host.name	Only used when 'listeners' is not set. This is the hostname of broker.	NULL

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 46 / 118 ← →

Kafka Consumer

- Applications subscribes and consumes messages from brokers in Kafka cluster.
- During consumption of messages from a topic a consumer group can be configured with multiple consumers.
- Each consumer of consumer group reads messages from a unique subset of partitions in each topic they subscribe to.
- Messages with same key arrives at same consumer.
- Supports both Queuing and Publish-Subscribe.
- Consumers have to maintain the number of messages consumed.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



Course Content

Presentation

Case Study I - Publis...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

Personal Library

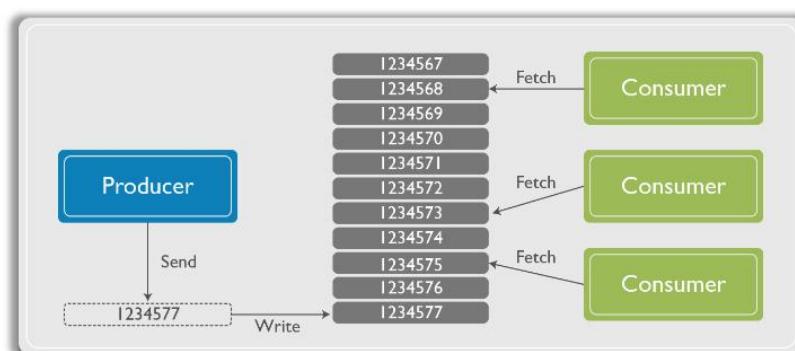
Presentation

47 / 118



Kafka Consumer

- *Multiple consumers can read from the same topic.*
- *Each consumer is responsible for managing its own offset.*
- *Messages stay on Kafka i.e. they are not removed after they are consumed.*



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 48 / 118

Consumer - Groups

Consumers can be *organised* into *consumer groups*.

Common Patterns:

- *All consumer instances in one group*
 - Acts like a *traditional queue* with load balancing.
- *All consumer instances in different groups*
 - *All messages* are *broadcast* to *all consumer instances*.
- *"Logical-Subscriber" – Many consumer instances in a group*
 - Consumers are *added* for *scalability* and *fault tolerance*.
 - *Each consumer instance* reads from one or more partitions for a topic.
 - There *cannot* be more *consumer instances* than *partitions*.

Kafka Cluster

Broker 1: P0, P3

Broker 2: P1, P2

Consumer Group A: C1, C2

Consumer Group B: C3, C4, C5, C6

Consumer groups provide isolation to topics and partition.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 49 / 118

Consumer - Groups

The diagram shows a Kafka Cluster with two brokers, Broker 1 and Broker 2, each containing partitions P0, P1, P3, and P2 respectively. Two consumer groups, Consumer Group A and Consumer Group B, are shown below. Consumer Group A has one active consumer (C1) and one inactive consumer (C2, marked with a red X). Consumer Group B has three active consumers (C3, C4, C5, and C6). Solid arrows indicate active assignments, while dashed arrows indicate inactive or rebalanced assignments.

Consumers can rebalance themselves for partitions

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

edureka!

Scaling Kafka

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

L

12:37 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Kafka Scale and Speed

How can Kafka scale if multiple producers and consumers read/write to same Kafka Topic log?

- Kafka writes fast: Sequential writes to filesystem are fast (700 MB or more a second)
- Kafka scales writes and reads by sharding:
 - Topic logs into Partitions (parts of a Topic log).
 - Topics logs can be split into multiple partitions which can be stored on multiple different servers, and those servers can use multiple disks.
 - Multiple Producers can write to different Partitions of the same Topic.
 - Multiple Consumers Groups can read from different partitions efficiently.



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

L

12:37 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 54 / 118 ← → []

ZooKeeper Service

The diagram illustrates the ZooKeeper service architecture. At the top, a blue box labeled "Zookeeper Service" contains a cartoon character of a person holding a shovel. Below it, a blue box labeled "Leader" is connected by a downward arrow to a central server node. This central node is labeled "Server". Four arrows point from four green trapezoid shapes labeled "Client" at the bottom to the "Server" node. To the right of the "Leader" box, there are two more "Server" nodes, each also connected by an upward arrow from a "Client" node. The entire diagram is set against a light gray background with the "edureka!" logo repeated in the background.

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

1 1 1 L X E M D 12:37 02-06-2023 ENG IN

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Course Content

Presentation Case Study I - Publis... Case Study II - Collect... Dataset Dataset Apache Spark Streami... Apache Spark Streami... In-class Project Spark GraphX Personal Library

Understanding Apache Kafka and Apache Flume > Presentation

Presentation 55 / 118

ZooKeeper and Kafka

- Each *Kafka broker* coordinates with *other Kafka brokers* using *ZooKeeper*.
- *Producers* and *consumers* are *notified* by *ZooKeeper service* about the presence of new broker in Kafka system or failure of the broker in Kafka system.
- Zookeeper is mainly used to *track status of nodes present in Kafka cluster* and also to *keep track of Kafka topics, messages, etc.*



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 56 / 118 ← →

ZooKeeper and Kafka

```
graph TD; Producer1[Producer] --> Kafka[Kafka Cluster]; Producer2[Producer] --> Kafka; Producer3[Producer] --> Kafka; Kafka --> Consumer1[Consumer]; Kafka --> Consumer2[Consumer]; Kafka --> Consumer3[Consumer]; Zookeeper[Zookeeper] --- Kafka; Zookeeper --- Consumer1;
```

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

L

12:37 02-06-2023

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

Presentation

Case Study I - Publish...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

e! Personal Library

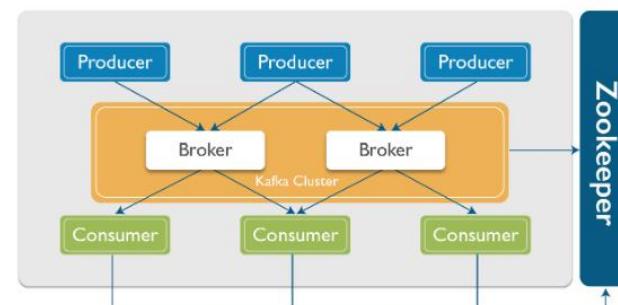
Presentation

58 / 118



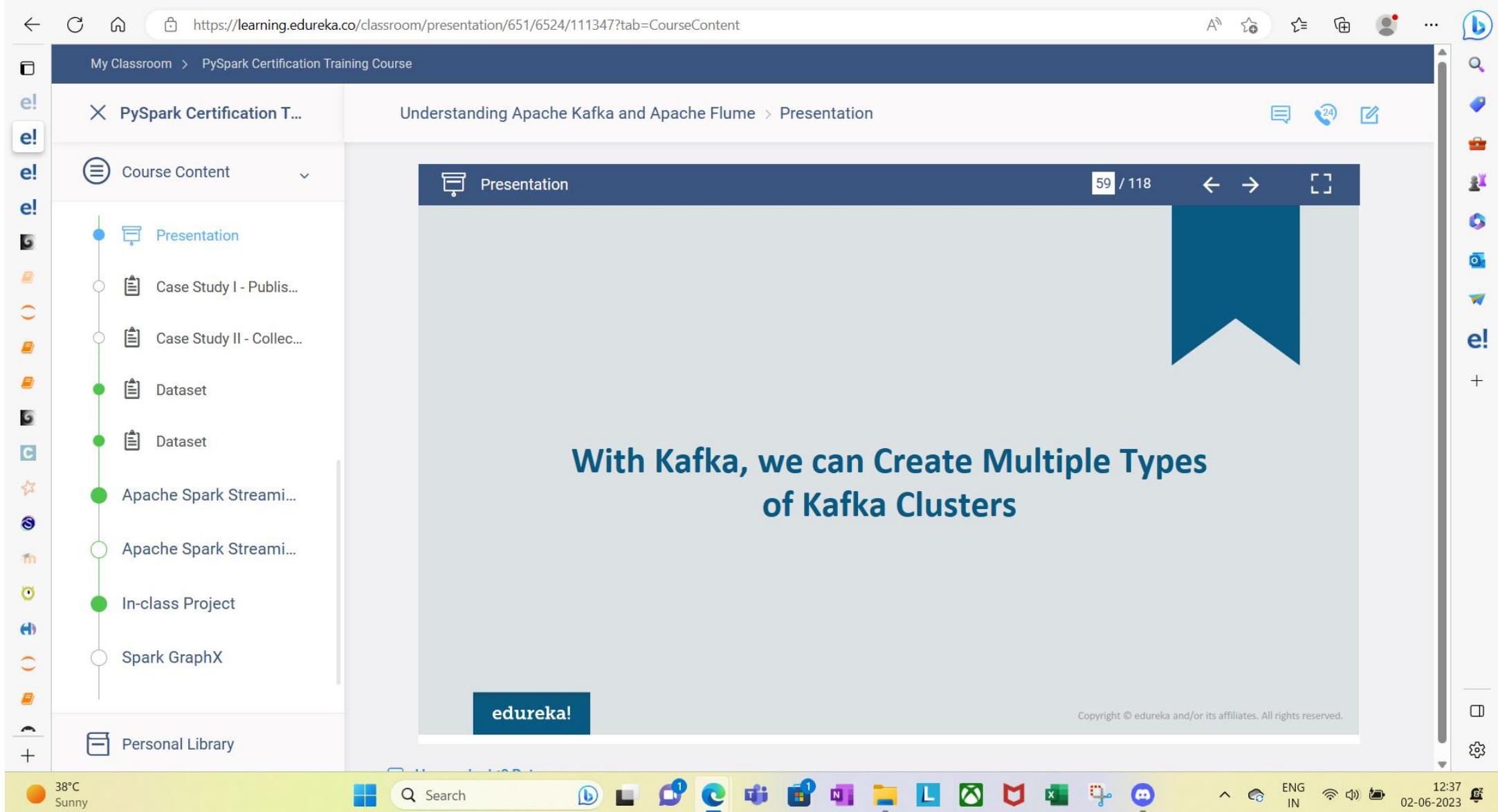
Kafka Cluster

- A Kafka Cluster is generally *fast, highly scalable messaging system.*
- A *publish & subscribe messaging system.*
- Can be used effectively in place of Java Messaging System (JMS) and Advanced Messaging Queuing Protocol(AMQP).
- Can be *integrated with Hadoop Ecosystem.*
- *Expanding* of the *cluster* can be done with *ease.*
- *Effective* for applications which involves *large scale message processing.*



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.



My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 60 / 118

Types of Kafka Clusters

- 1 Single Node-Single Broker Cluster
- 2 Single Node-Multiple Broker Cluster
- 3 Multiple Nodes-Multiple Broker Cluster

The diagram illustrates the three types of Kafka clusters:

- Single Node-Single Broker Cluster:** A single producer connects to a single Kafka Broker.
- Single Node-Multiple Broker Cluster:** Three producers connect to a single Kafka Broker.
- Multiple Nodes-Multiple Broker Cluster:** Three producers connect to three separate Kafka Brokers, which then serve three consumers.

A Zookeeper server is shown connected to the Kafka Brokers.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 61 / 118

Types of Kafka Clusters

- 1 Single Node-Single Broker Cluster
- 2 Single Node-Multiple Broker Cluster
- 3 Multiple Nodes-Multiple Broker Cluster

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

L

12:37 02-06-2023



Course Content

Presentation

Case Study I - Publish...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

Personal Library

Presentation

62 / 118

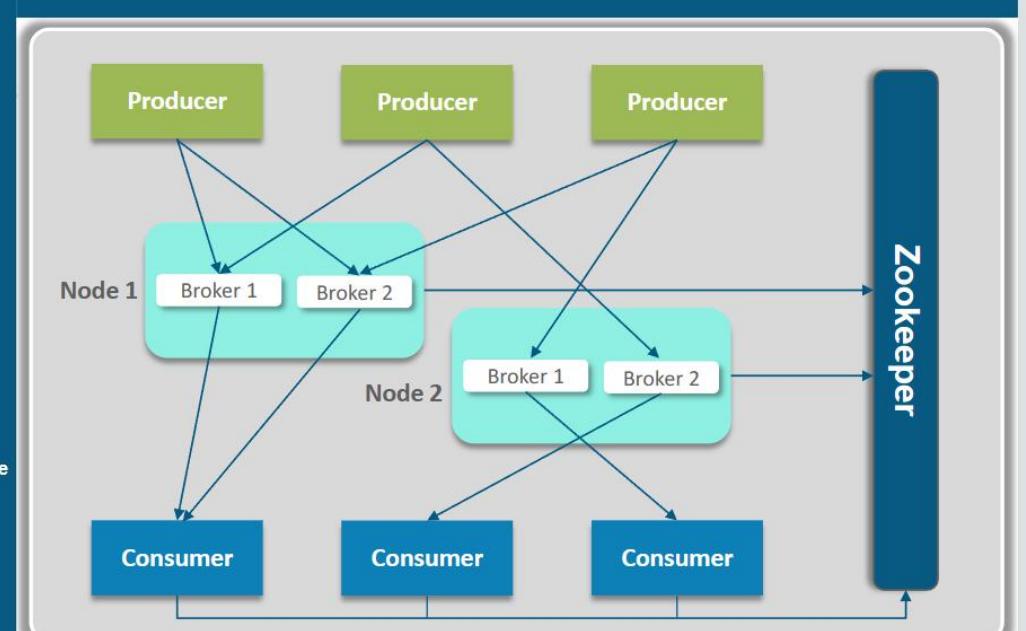


Types of Kafka Clusters

1 Single Node-Single Broker Cluster

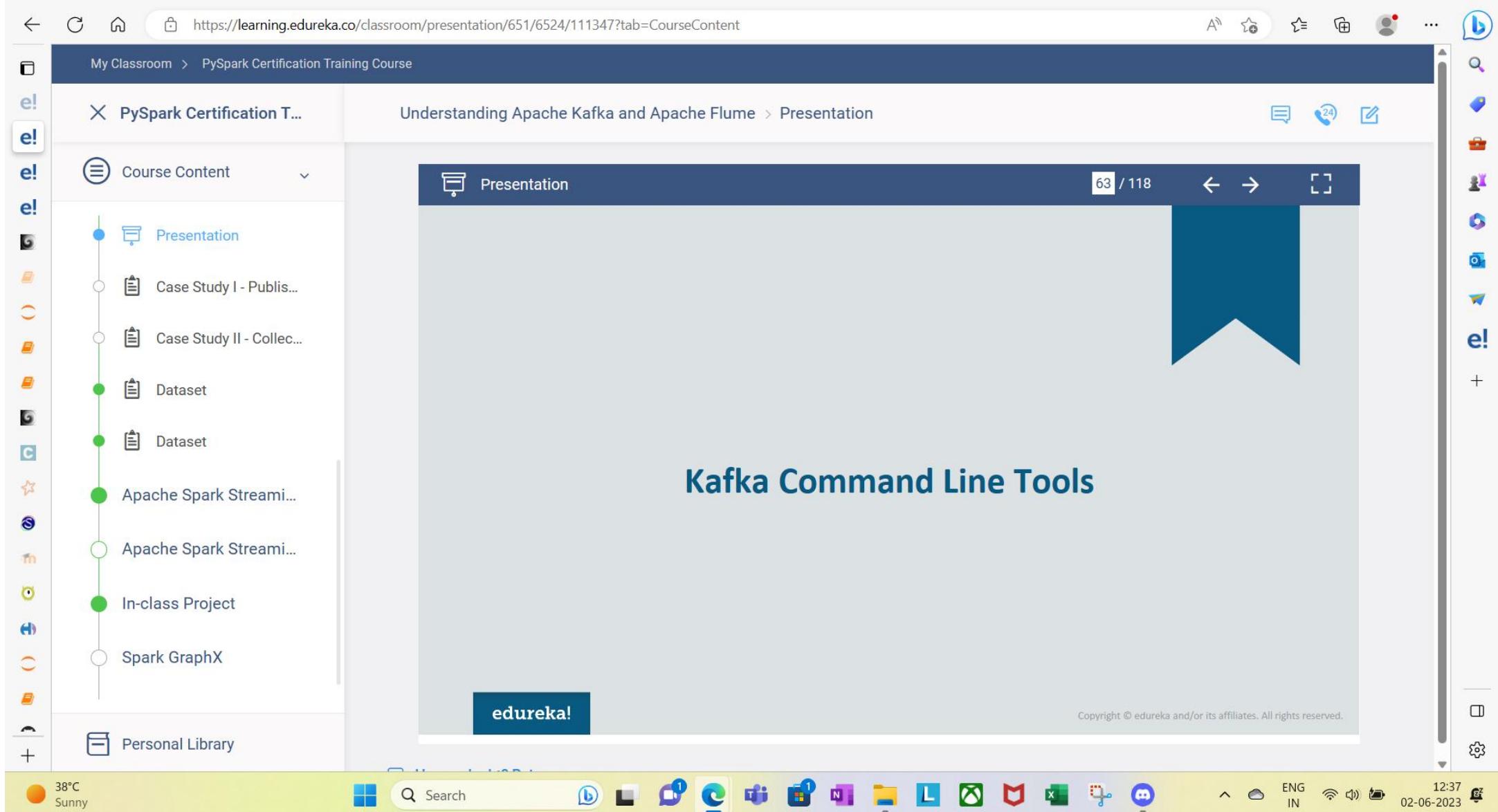
2 Single Node-Multiple Broker Cluster

3 Multiple Nodes-Multiple Broker Cluster



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.



X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

e! Presentation

e! Case Study I - Publis...

e! Case Study II - Collect...

e! Dataset

e! Dataset

e! Apache Spark Streami...

e! Apache Spark Streami...

e! In-class Project

e! Spark GraphX

e! Personal Library

Presentation

64 / 118



Testing Kafka Cluster

- Kafka Cluster can run against the following broker model:
 - Single Broker Cluster
 - Multi Broker Cluster
- *Single broker cluster* generally *runs only one instance* compared to *multi broker* which *runs multiple instances*.
- To test the Kafka Cluster the following shell scripts can be used:

Kafka Shell Scripts	Description
zookeeper-server-start.sh	Starts the zookeeper with the properties configured under config/zookeeper.properties
kafka-server-start.sh	Starts the Kafka server with the properties configured under config/server.properties
kafka-topics.sh	Used to create and list topics
kafka-console-producer.sh	Command line client to send messages to the Kafka Cluster
kafka-console-consumer.sh	Command line client to consume messages from the Kafka Cluster

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

edureka!

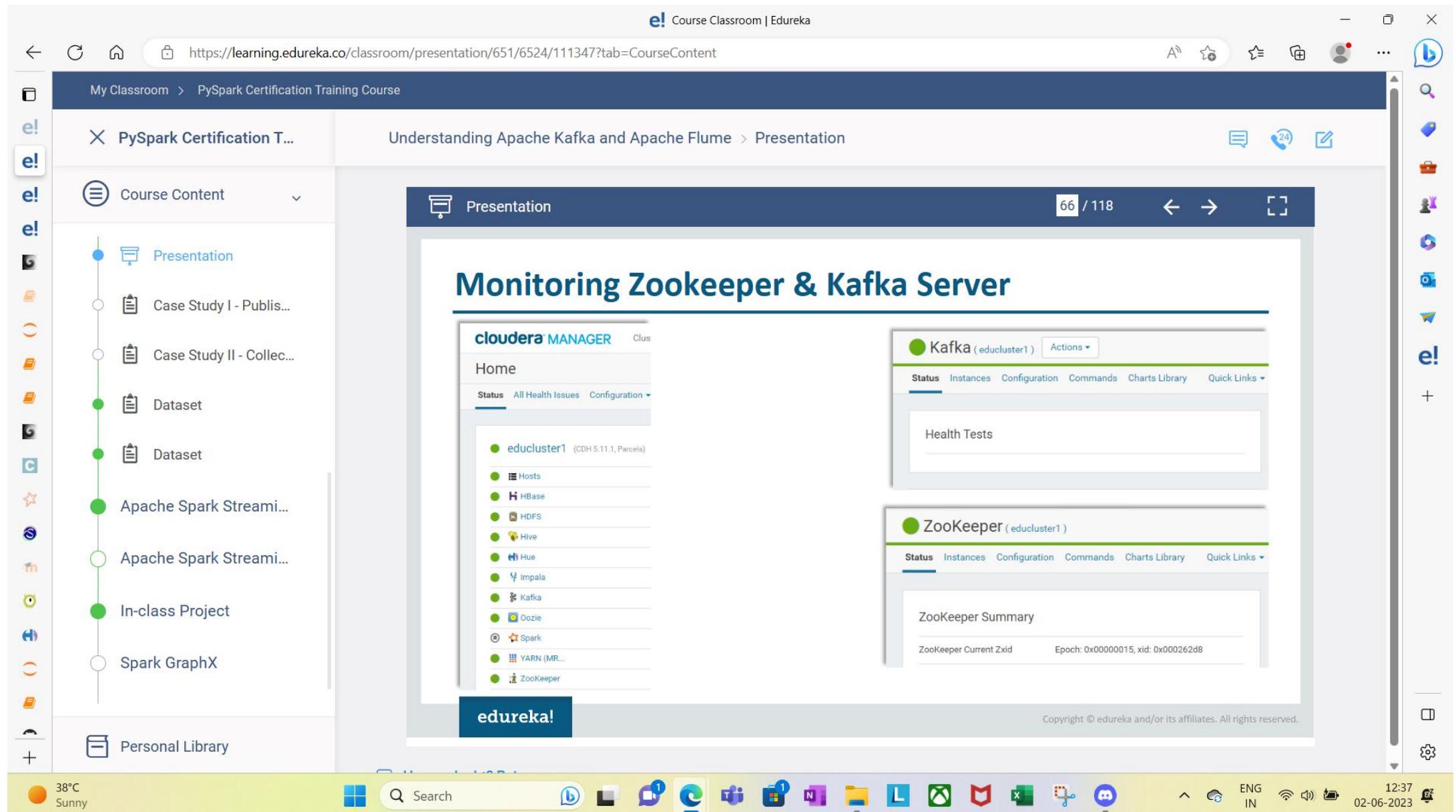
Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

L

12:37 02-06-2023



My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 67 / 118 ← →

Monitoring Zookeeper Server

ZooKeeper (educluster1)

Role Type	State	Host	Commission State	Role Group
Server	Started	ip-20-0-31-161.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-21-161.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-21-85.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-31-249.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-21-196.ec2.internal	Commissioned	Server Default Group

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

My Classroom > PySpark Certification Training Course

X PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publish...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Personal Library

Presentation

Monitoring Kafka Server

Kafka (educluster1) Actions ▾

Status Instances Configuration Commands Charts Library Quick Links ▾

Filters

Search

Role Type	State	Host	Commission State	Role Group
Gateway	N/A	ip-20-0-41-93.ec2.internal	Commissioned	Gateway Default Group
Kafka Broker	Started	ip-20-0-31-161.ec2.internal	Commissioned	Kafka Broker Default Group
Kafka Broker	Stopped	ip-20-0-31-78.ec2.internal	Commissioned	Kafka Broker Default Group
Kafka Broker (Active Controller)	Started	ip-20-0-31-249.ec2.internal	Commissioned	Kafka Broker Default Group
Kafka Broker	Stopped	ip-20-0-32-147.ec2.internal	Commissioned	Kafka Broker Default Group
Kafka Broker	Started	ip-20-0-31-221.ec2.internal	Commissioned	Kafka Broker Default Group
Kafka Broker	Stopped	ip-20-0-31-127.ec2.internal	Commissioned	Kafka Broker Default Group

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Presentation

69 / 118

Demo 1 – Kafka Commands

Refer to the file Module-9 Demo 1 provided in the LMS for all the steps in detail

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

My Classroom > PySpark Certification Training Course

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

Presentation

Case Study I - Publis...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

Personal Library

Presentation

70 / 118



Setting up Kafka Single Node Multi-Broker Cluster

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

My Classroom > PySpark Certification Training Course

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

Presentation

Case Study I - Publis...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

Personal Library

Monitor ZooKeeper Server

ZooKeeper (educluster1)

Status Instances Configuration Commands Charts Library Quick Links

Filters

▼ STATUS

Good Health

Role Type	State	Host	Commission State	Role Group
Server	Started	ip-20-0-31-161.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-21-161.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-21-85.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-31-249.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-21-196.ec2.internal	Commissioned	Server Default Group

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C
Sunny

Search



12:37 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Presentation

72 / 118

What do you think is the change in Single Node Multi-Broker Cluster Setup?

Refer to the file Module-9 Demo 2 provided in the LMS for all the steps in detail

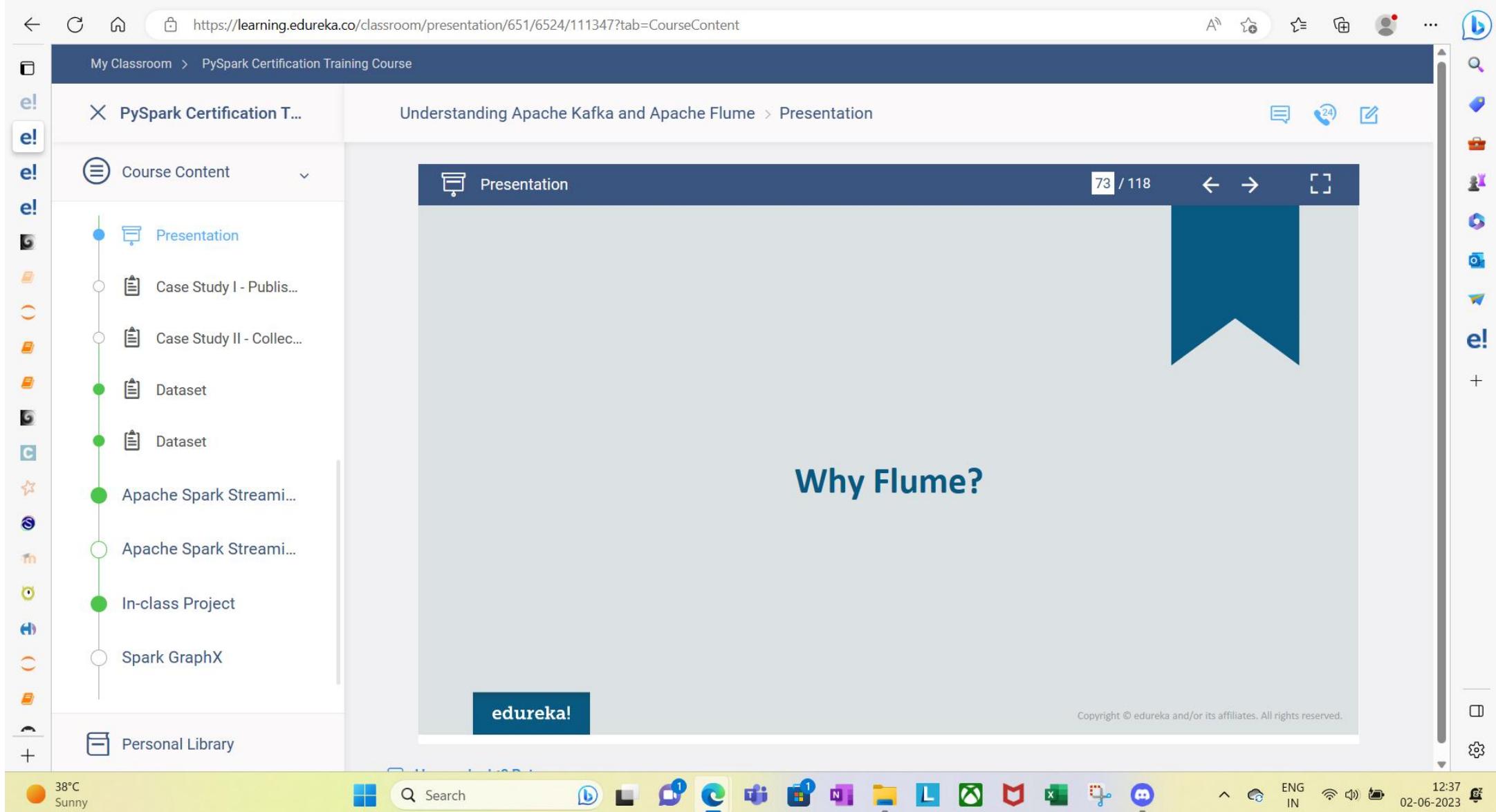
edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023



X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

Presentation

Case Study I - Publis...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

e! Personal Library

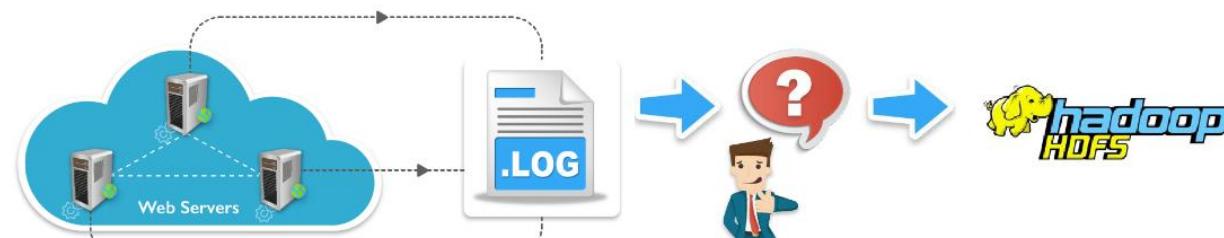
Why Flume?

Scenario

Suppose we have hundreds of services running in different servers that produce lots of large logs which should be analysed altogether. We have Hadoop to process them.

Problem

How do we send all the logs to a place that has Hadoop? We need a reliable, scalable, extensible and manageable way to do it!



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

Presentation

Case Study I - Publis...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

e! Personal Library

Why Flume?

Scenario

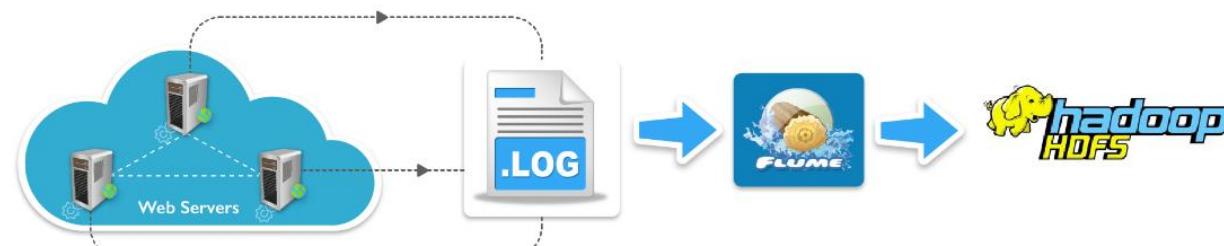
Suppose we have hundreds of services running in different servers that produce lots of large logs which should be analysed altogether. We have Hadoop to process them.

Problem

How do we send all the logs to a place that has Hadoop? We need a reliable, scalable, extensible and manageable way to do it!

Solution

Apache Flume is the most reliable, distributed, and available service for systematically collecting, aggregating, and moving large amounts of streaming data (logs) into the HDFS.



edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Personal Library

Presentation

76 / 118

What is Flume?

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Course Content

Presentation Case Study I - Publis... Case Study II - Collect... Dataset Dataset Apache Spark Streami... Apache Spark Streami... In-class Project Spark GraphX Personal Library

Understanding Apache Kafka and Apache Flume > Presentation

Presentation 77 / 118

What is Flume?

“Flume is a distributed, reliable and available service for efficiently collecting, aggregating, and moving large amounts of log data from various sources to a centralized data store.”

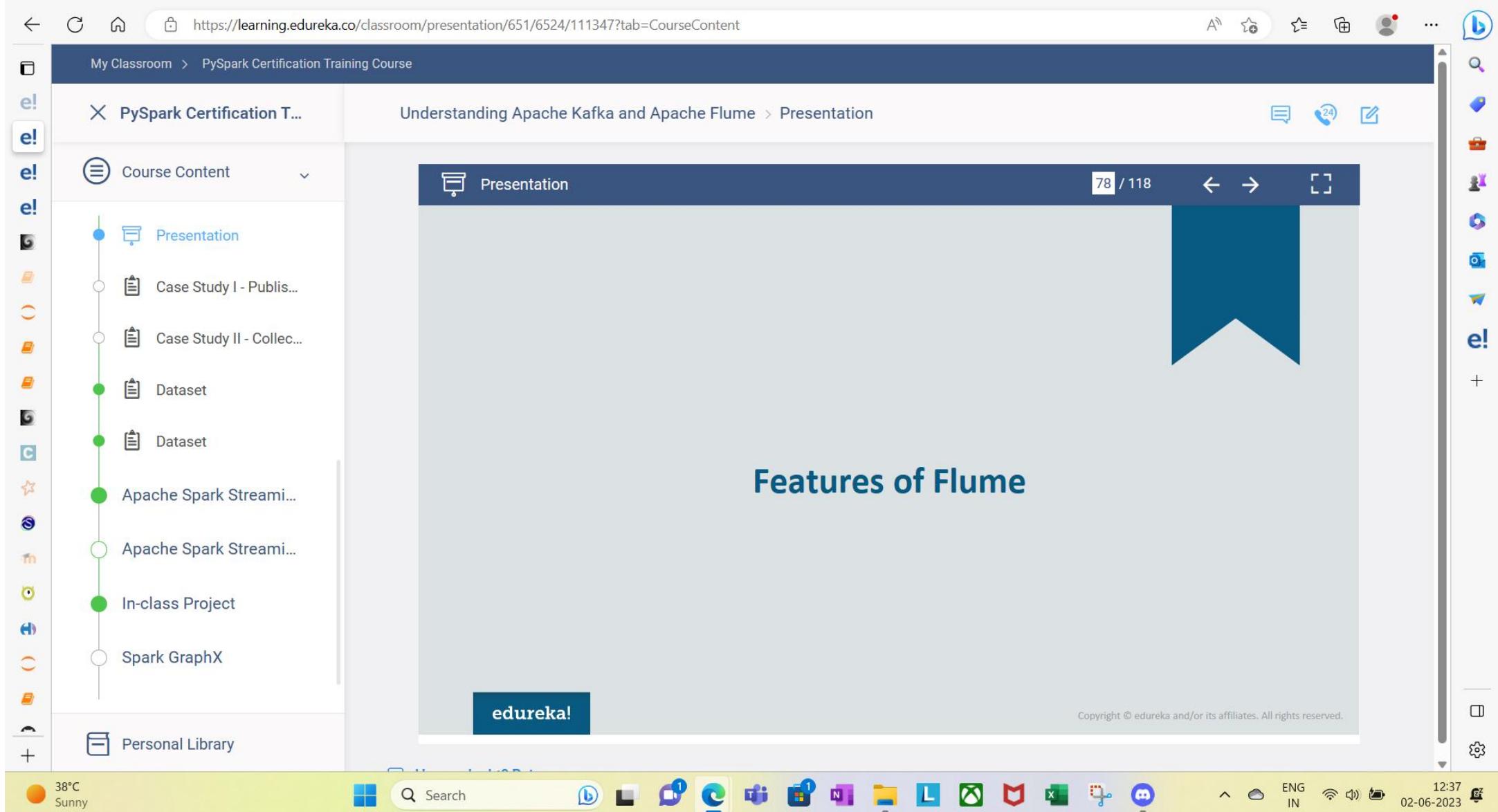
The diagram illustrates the Flume architecture. On the left, a box labeled "Log/Event Data Generators" contains icons for "BIG DATA Cloud", "Twitter", "Facebook", and "Web Servers". Arrows labeled "Log/Event Data" point from these generators to a central box labeled "FLUME". From the "FLUME" box, arrows labeled "Log/Event Data" point to a box on the right labeled "Centralized Storage", which contains icons for "APACHE HBASE" and "hadoop HDFS".

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny Search

12:37 02-06-2023



My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 79 / 118

Flume Features

Flume collects the log data from multiple web servers and ingests them into a centralized store (HDFS, HBase) efficiently.

Using Flume we can collect the data from multiple servers in real-time as well as in batch mode.

Multi-hop flows, fan-in fan-out flows, contextual routing, etc are supported by Flume.

Flume is also used to import huge volumes of event data produced by sites like Facebook, Twitter, Amazon and Flipkart.

Flume can be scaled horizontally.

Flume can collect data from a large set of sources and move them to multiple destinations.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Personal Library

Presentation

80 / 118

Basic Flume Architecture

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

My Classroom > PySpark Certification Training Course

X PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

edureka!

Flume Architecture

▪ Flume has a simple and flexible architecture based on streaming **data flow model**.
▪ It is **robust** and **fault tolerant** with tunable reliability mechanisms and many failover and recovery mechanisms.

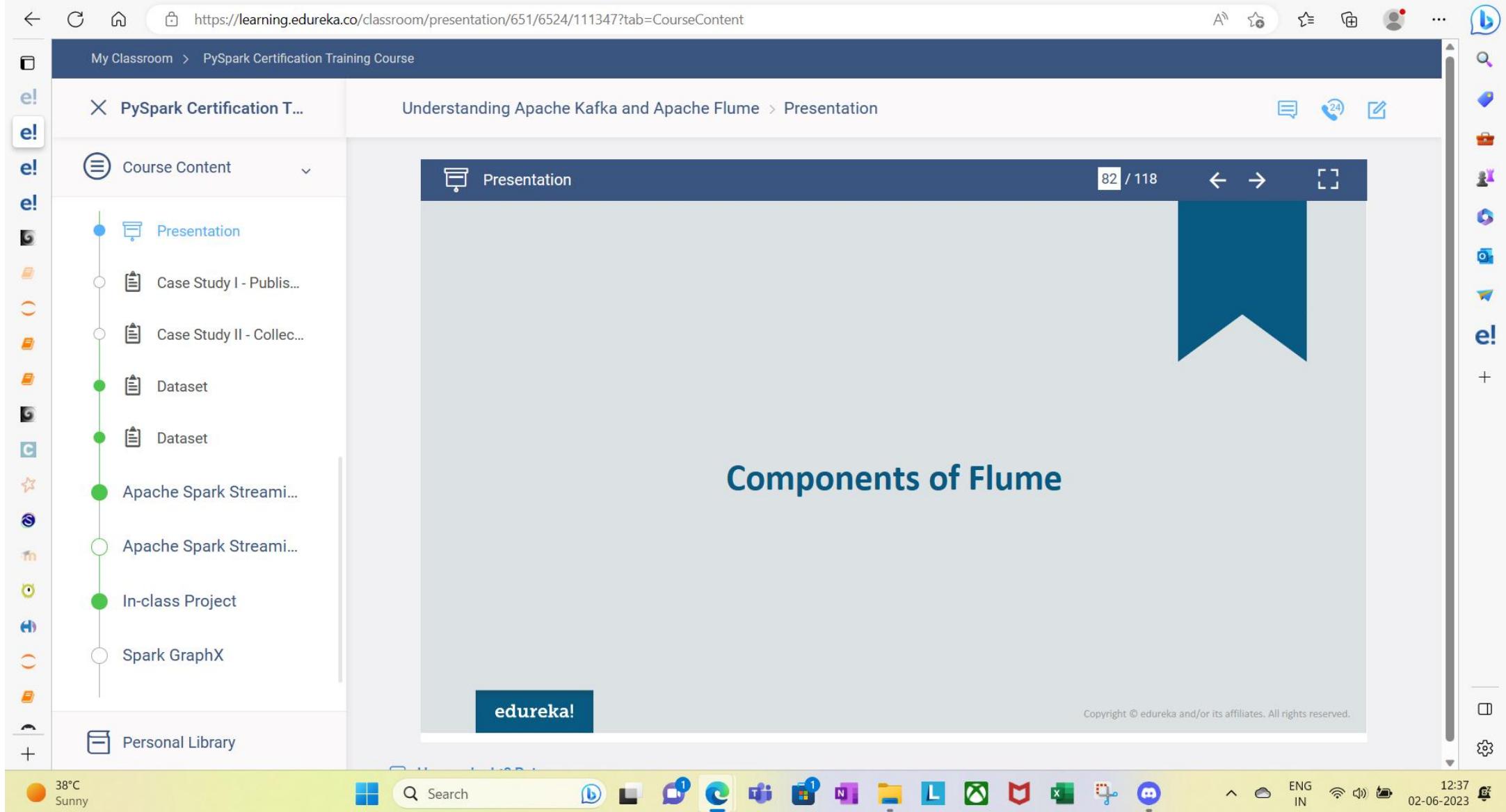
```
graph LR; WS([Web Servers]) -- "Log/Event Data" --> S[Source]; S --> MC[Memory Channel<br/>Agent]; MC --> SINK[Sink]; SINK -- "Log/Event Data" --> HDFS[HDFS]
```

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023



My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 83 / 118 ← →

Flume Components

```
graph LR; Client[Client] -- Data --> Source[Source]; Source -- Event --> MemoryChannel[Memory Channel]; MemoryChannel -- Event --> Sink[Sink]; subgraph Agent [Agent]; Client --- Source; Source --- MemoryChannel; MemoryChannel --- Sink; end;
```

The entity through which data enters into Flume.
Flume supports Avro, Netcat, Thrift, exec, syslog as source of data.

The conduit between the Source and the Sink.
Sources ingest events into the channel and the sinks drain the channel.

The entity that receive events from the channel.
A variety of sinks allow data to be streamed to a range of destinations. One example is the HDFS sink.

The entity that produces and transmits the event to the source operating within the Agent.

Any physical Java virtual machine running Flume.
It is a collection of sources, sinks and channels.

A singular unit of data that is transported by Flume (typically a single log entry).

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

L

12:37 02-06-2023

My Classroom > PySpark Certification Training Course

X PySpark Certification T...

e! Course Content

- Presentation
- Case Study I - Publish...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

+ Personal Library

Understanding Apache Kafka and Apache Flume > Presentation

Presentation 86 / 118 ← →

How does Flume Work?

Flume uses agents which have:

- A source**
 - Listen for events delivered to it by an external source like a web server.
 - Write events to channel
- A channel**
 - Channel keeps the event until it's consumed by a Flume sink
 - Queue event data as transactions
- A sink**
 - Write event data to target like HDFS
 - Remove event from queue

```
graph LR; Source[Source] -- "Log/Event Data" --> Channel[Channel]; Channel --> Sink[Sink]; Sink -- "Log/Event Data" --> Agent[Flume Agent];
```

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

https://learning.edureka.co/course/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Personal Library

Presentation

87 / 118

Types of Dataflow

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:37 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 88 / 118

Multi-Agent Flow

Within Flume, there can be **multiple agents** and before reaching the final destination, an event may travel through more than one agent. This is known as **multi-agent flow**.

```
graph LR; SA[Source A] --> CA[Channel A]; CA --> SA_Avro[Sink A (Avro)]; SA_Avro --> SB[Source B (Avro)]; SB --> CB[Channel B]; CB --> SB_Avro[Sink B (Avro)];
```

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

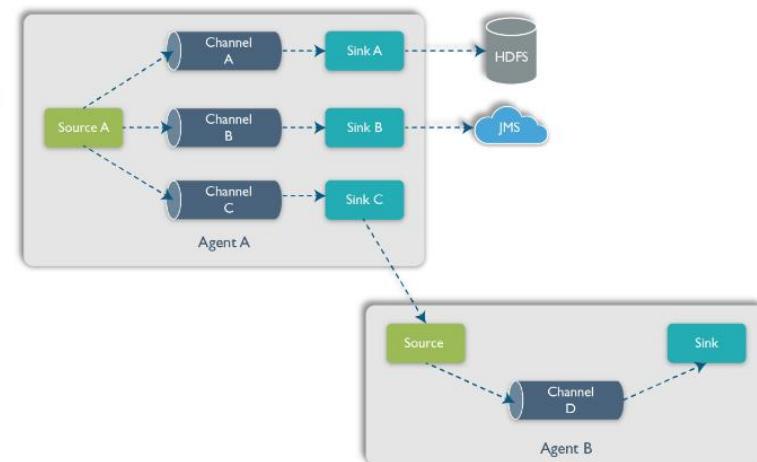
38°C Sunny

Search

12:37 02-06-2023

Fan-Out Flow : Multiplexing the Flow

- The dataflow from one source to multiple channels is known as **fan-out flow**.
- Flume supports multiplexing the event flow to one or more destinations.
- This is achieved by defining a flow multiplexer that can replicate or selectively route an event to one or more channels.

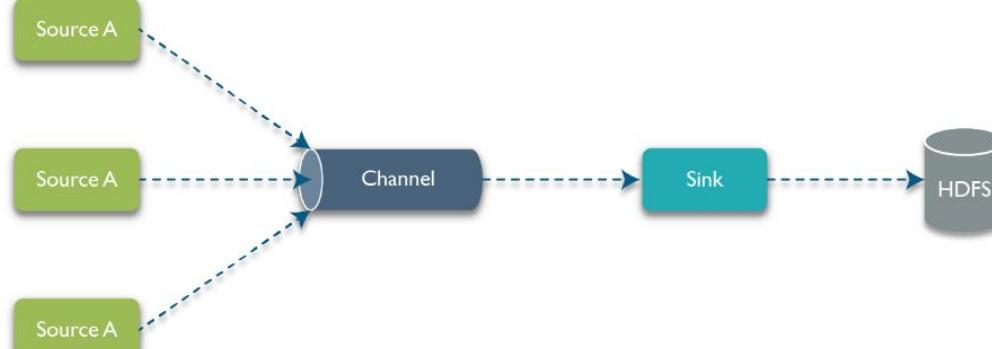


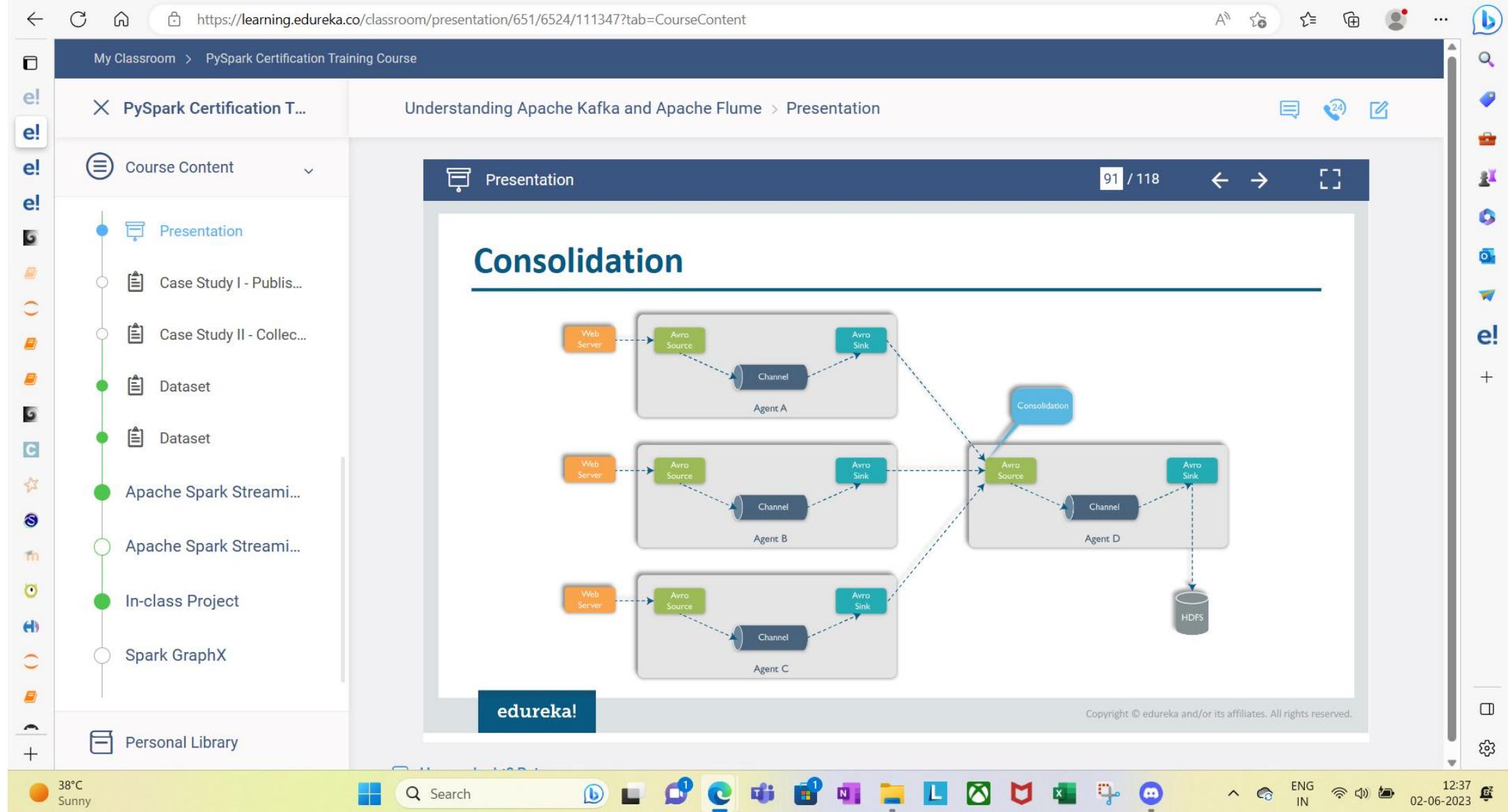
edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

Fan-In Flow

The data flow in which the data will be transferred from many sources to one channel is known as **fan-in flow**.





https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Presentation

92 / 118

Demo 3 – Flume Commands

Refer to the file Module-9 Demo 3 provided in the LMS for all the steps in detail

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:38 02-06-2023

https://learning.edureka.co/classroom/presentation/651/6524/111347?tab=CourseContent

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Presentation

93 / 118

Demo 4 – Setting up an Agent

Refer to the file Module-9 Demo 4 provided in the LMS for all the steps in detail

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:38 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Setting up an Agent

- Flume agents are configured using plain text configuration file.
- Flume configuration uses the Java properties file format, which is simply a plain text file with new line separated key-value pairs.
- Configurations for one or more agents can be specified in the same configuration file.
- The configuration file includes properties of each source, sink and channel in an agent and how they are wired together to form data flows.

```
flume-conf.properties  
/usr/lib/apache-flume-1.7.0-bin/conf  
  
# The configuration file needs to define the sources,  
# the channels and the sinks.  
# Sources, channels and sinks are defined per agent,  
# in this case called 'agent'  
  
agent.sources = seqGenSrc  
agent.channels = memoryChannel  
agent.sinks = loggerSink  
  
# For each one of the sources, the type is defined  
agent.sources.seqGenSrc.type = seq  
  
# The channel can be defined as follows.  
agent.sources.seqGenSrc.channels = memoryChannel  
  
# Each sink's type must be defined  
agent.sinks.loggerSink.type = logger  
  
#Specify the channel the sink should use  
agent.sinks.loggerSink.channel = memoryChannel  
  
# Each channel's type is defined.  
agent.channels.memoryChannel.type = memory  
  
# Other config values specific to each type of channel(sink or source)  
# can be defined as well  
# In this case, it specifies the capacity of the memory channel  
agent.channels.memoryChannel.capacity = 100
```

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:38 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 95 / 118

Simple Configuration File

Name the components on the agent a1

```
a1.sources = r1
a1.sinks = k1
a1.channels = c1
```

Describe/Configure the source

```
a1.sources.r1.type = netcat
a1.sources.r1.bind = localhost
a1.sources.r1.port = 44444
```

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

L

12:38 02-06-2023

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 96 / 118

Simple Configuration File

Describe/Configure the sink

```
a1.sinks.k1.type = logger
```

Use a channel which buffers events in memory

```
a1.channels.c1.type = memory
```

```
a1.channels.c1.capacity = 1000
```

```
a1.channels.c1.transactionCapacity = 100
```

Bind the source and sink to the channel

```
a1.sources.r1.channels = c1
```

```
a1.sinks.k1.channels = c1
```

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

Personal Library

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 97 / 118 ← →

Starting a Flume Agent

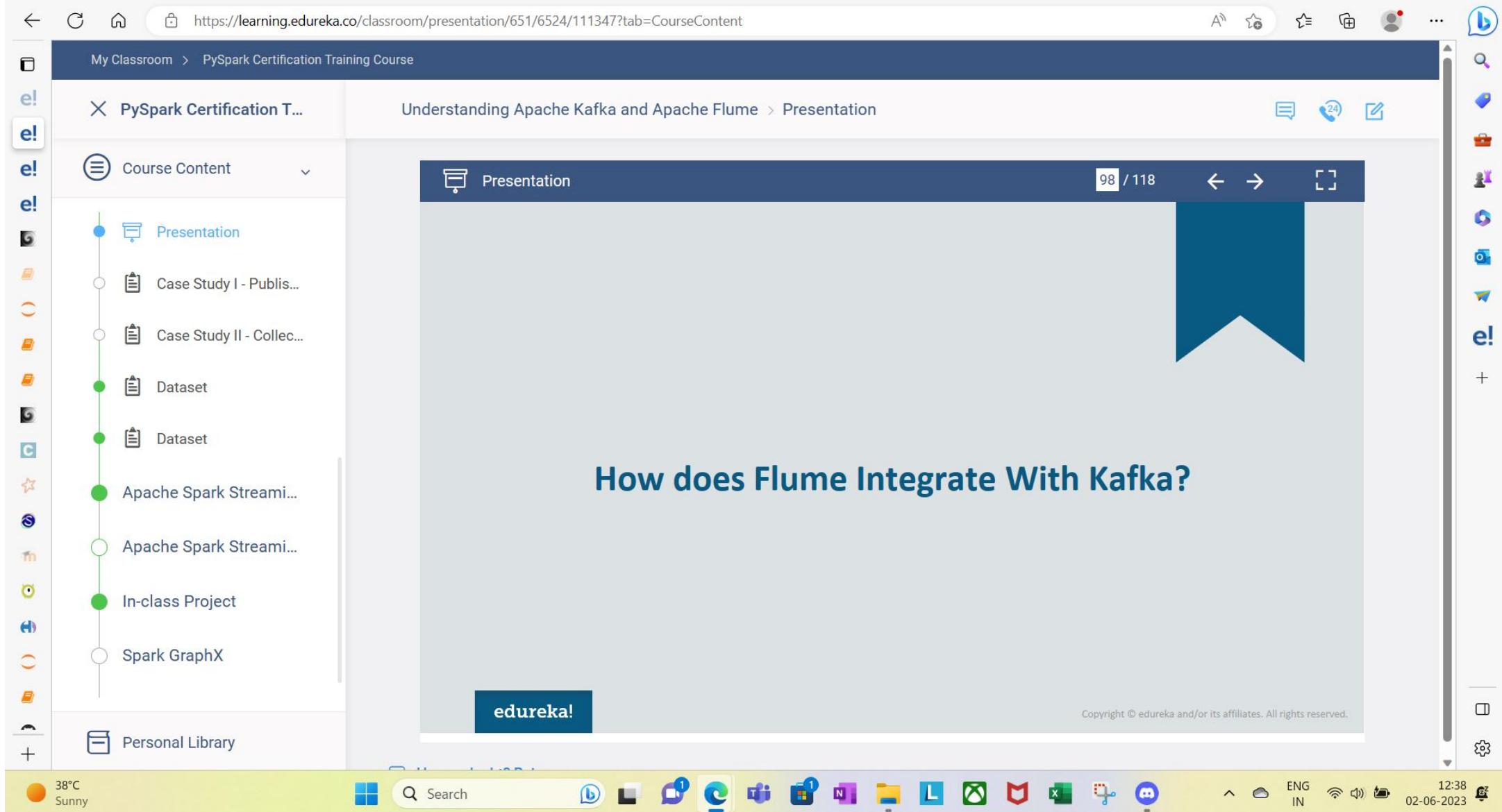
- An agent is started using a shell script called flume-ng which is located in the bin directory of the Flume distribution.
- We need to specify the agent name, the config directory, and the config file on the command line:

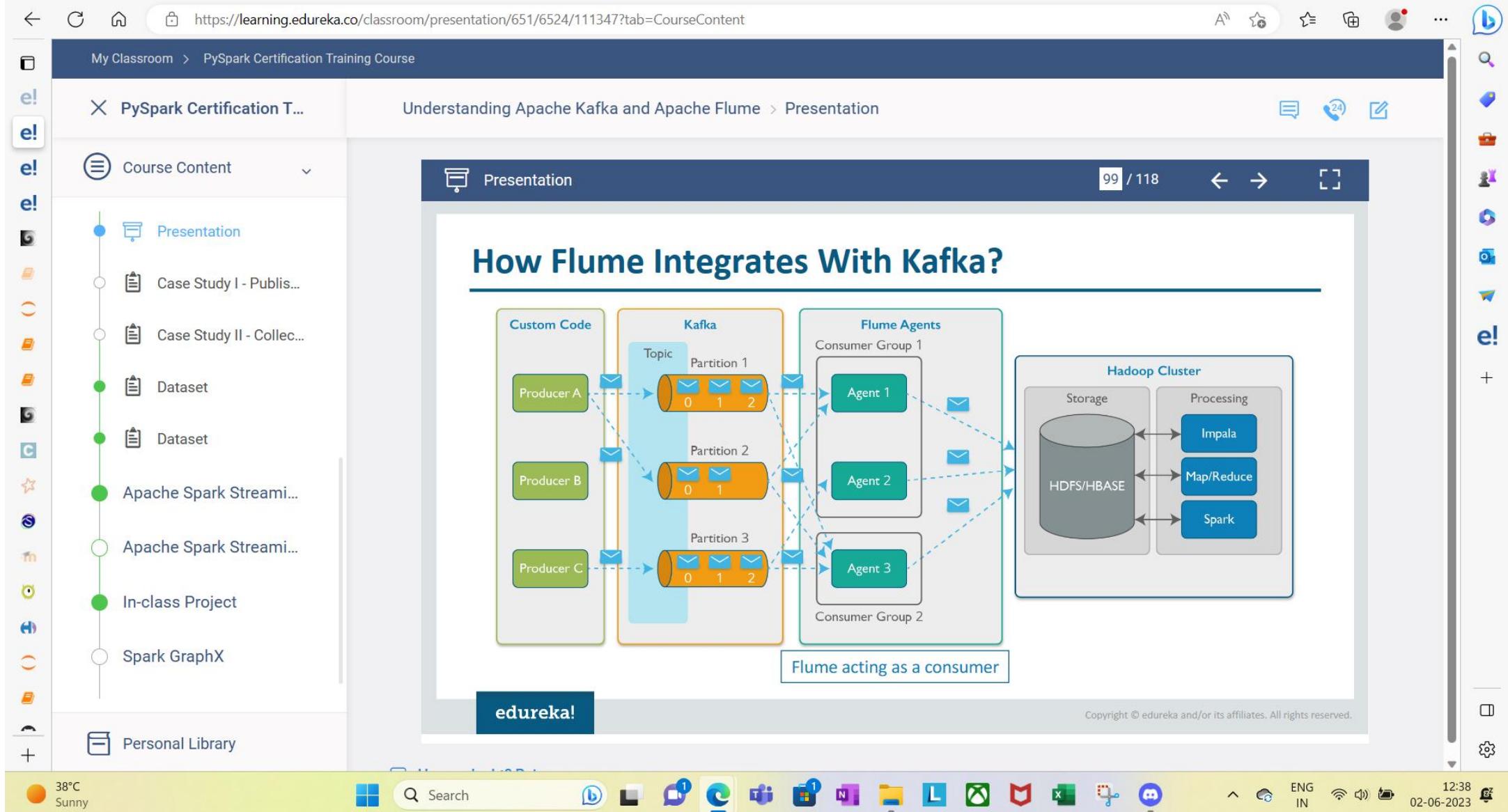
```
$ bin/flume-ng agent --conf conf --conf-file flume.conf --name al  
-Dflume.root.logger=INFO,console
```

Parameter	Description
agent	Command to start the Flume agent
--conf , -c<conf>	Use configuration file in the conf directory
-f<file>	Specifies a config file path
--name , -n<name>	Name of the agent
-D property=value	Sets a Java system property value

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.





My Classroom > PySpark Certification Training Course

X PySpark Certification T...

e! Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

+ Personal Library

Understanding Apache Kafka and Apache Flume > Presentation

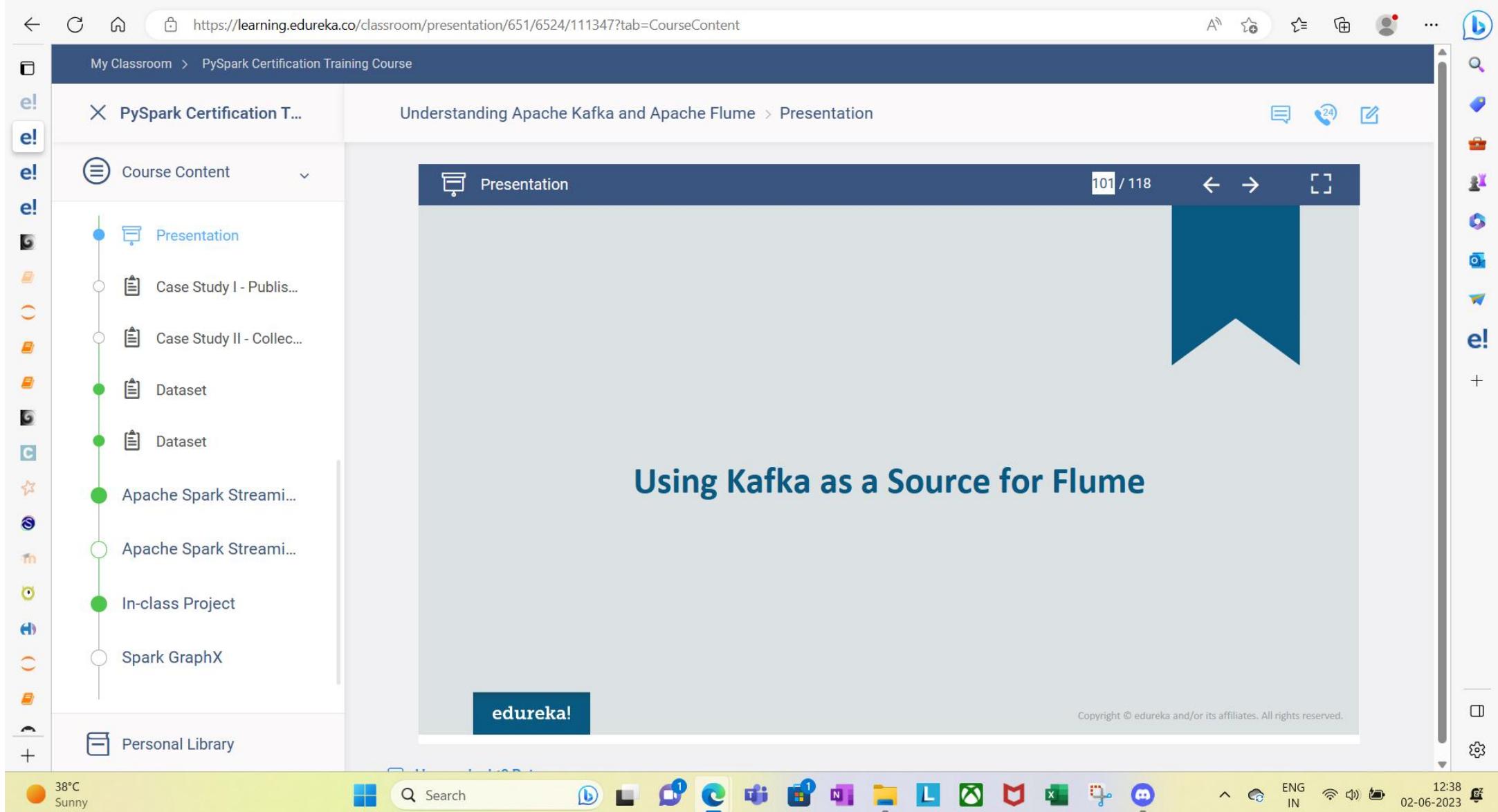
Presentation 100 / 118 ← →

How Flume Integrates With Kafka?

The diagram illustrates the integration between Flume and Kafka. On the left, a box labeled "Data Sources" contains icons for Logs, JMS, and WebServer, etc. Three dashed arrows point from these sources to three boxes labeled "Flume Agents": "Agent 1", "Agent 2", and "Agent 3". Below these agents is a box labeled "Flume acting as a producer". To the right, a box labeled "Kafka" contains a "Topic" with three "Partition 1", "Partition 2", and "Partition 3". Each partition is represented by a cylinder divided into three segments, each containing a blue envelope icon.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.



My Classroom > PySpark Certification Training Course

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

e! Presentation

e! Case Study I - Publis...

e! Case Study II - Collect...

e! Dataset

e! Dataset

e! Apache Spark Streami...

e! Apache Spark Streami...

e! In-class Project

e! Spark GraphX

e! Personal Library

Monitoring Zookeeper Server

ZooKeeper (educluster1)

Status Instances Configuration Commands Charts Library Quick Links

Filters

STATUS

Good Health

Role Type	State	Host	Commission State	Role Group
Server	Started	ip-20-0-31-161.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-21-161.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-21-85.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-31-249.ec2.internal	Commissioned	Server Default Group
Server	Started	ip-20-0-21-196.ec2.internal	Commissioned	Server Default Group

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

My Classroom > PySpark Certification Training Course

X PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Personal Library

Presentation

Monitoring Kafka Server

Kafka (educluster1) Actions ▾

Status Instances Configuration Commands Charts Library Quick Links ▾

Filters

Search

Role Type	State	Host	Commission State	Role Group
Gateway	N/A	ip-20-0-41-93.ec2.internal	Commissioned	Gateway Default Group
Kafka Broker	Started	ip-20-0-31-161.ec2.internal	Commissioned	Kafka Broker Default Group
Kafka Broker	Stopped	ip-20-0-31-78.ec2.internal	Commissioned	Kafka Broker Default Group
Kafka Broker (Active Controller)	Started	ip-20-0-31-249.ec2.internal	Commissioned	Kafka Broker Default Group
Kafka Broker	Stopped	ip-20-0-32-147.ec2.internal	Commissioned	Kafka Broker Default Group
Kafka Broker	Started	ip-20-0-31-221.ec2.internal	Commissioned	Kafka Broker Default Group
Kafka Broker	Stopped	ip-20-0-31-127.ec2.internal	Commissioned	Kafka Broker Default Group

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:38 02-06-2023

My Classroom > PySpark Certification Training Course

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

Presentation

Case Study I - Publis...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

Personal Library

Create Kafka Topic

```
Command: kafka-topics --create --zookeeper ip-20-0-21-161.ec2.internal:2181 --replication-factor 1 --partitions 1 --topic Flume_Topic
```

```
[edureka_249489@ip-20-0-41-190 ~]$ kafka-topics --create --zookeeper ip-20-0-21-161.ec2.internal:2181 --replication-factor 1 --partitions 1 --topic Flume_Topic
18/02/16 12:42:35 INFO zkclient.ZkClient: zookeeper state changed (SyncConnected)
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not both.
18/02/16 12:42:36 INFO admin.AdminUtils$: Topic creation {"version":1,"partitions":[{"id":1,"topic": "Flume_Topic"}]}
Created topic "Flume_Topic".
18/02/16 12:42:36 INFO zkclient.ZkEventThread: Terminate ZkClient event thread.
18/02/16 12:42:36 INFO zookeeper.ZooKeeper: Session: 0x76166221525602b closed
18/02/16 12:42:36 INFO zookeeper.ClientCnxn: EventThread shut down for session: 0x76166221525602b
```

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C
Sunny

Search



12:38

02-06-2023

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

Presentation

Case Study I - Publis...

Case Study II - Collect...

Dataset

Dataset

Apache Spark Streami...

Apache Spark Streami...

In-class Project

Spark GraphX

Personal Library

e! Presentation

106 / 118



Setup Flume Configuration File

```
flumexercise.conf - Notepad
File Edit Format View Help
agent1.sources = source1
agent1.channels = channel1
agent1.sinks = sink1

agent1.sources.source1.type = org.apache.flume.source.kafka.KafkaSource
agent1.sources.source1.kafka.bootstrap.servers = ip-20-0-31-210.ec2.internal:9092
agent1.sources.source1.kafka.topics = Flume_Topic
agent1.sources.source1.kafka.consumer.group.id = flume
agent1.sources.source1.channels = channel1
agent1.sources.source1.interceptors = i1
agent1.sources.source1.interceptors.i1.type = timestamp
agent1.sources.source1.kafka.consumer.timeout.ms = 100

agent1.channels.channel1.type = memory
agent1.channels.channel1.capacity = 10000
agent1.channels.channel1.transactionCapacity = 1000

agent1.sinks.sink1.type = hdfs
agent1.sinks.sink1.hdfs.path = hdfs://nameservice1/user/edureka_249489/Flume_Kafka
agent1.sinks.sink1.hdfs.rollInterval = 5
agent1.sinks.sink1.hdfs.rollSize = 0
agent1.sinks.sink1.hdfs.rollCount = 0
agent1.sinks.sink1.hdfs.fileType = DataStream
agent1.sinks.sink1.channel = channel1
```

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.



X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



Start Flume Agent

```
Command: flume-ng agent --conf conf --conf-file flumexercise.conf --name agent1 -Dflume.root.logger=INFO,console
```

```
[derekula_249489@ip-20-0-41-190 ~] flume-ng agent --conf conf --conf-file flumexercise.conf --name agent1 -Dflume.root.logger=INFO,console  
Info: Including Hadoop libraries found via (/bin/hadoop) for HDFS access  
Info: Including HBase libraries found via (/bin/hbase) for HBase access  
Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release  
Info: Including Hive libraries found via () for Hive access  
+ exec /usr/java/jdk1.8.0_144-cloudera/bin/java -Xmx20m -Dflume.root.logger=INFO,console -cp 'conf:/opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1-1.0.0.1-1000/lib/*:/etc/hadoop/conf:/opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1-1.p0.4/lib/hadoop/libexec/../../hadoop/lib/*:/opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1-1.p0.4/lib/hadoop/libexec/../../hadoop/libexec/*:/opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1-1.p0.4/lib/hadoop/libexec/../../hadoop-hdfs/lib/*:/opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1-1.p0.4/lib/hadoop/libexec/../../hadoop-hdfs/lib/*:/opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1-1.p0.4/lib/hadoop/libexec/../../hadoop-yarn/lib/*:/opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1-1.p0.4/lib/hadoop/libexec/../../hadoop-mapreduce/lib/*:/opt/cloudera/parcels/CDH/lib/hadoop-mapreduce/../../hadoop-mapreduce/*:/opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1-1.p0.4/lib/hbase/bin/*:/conf:/usr/java/jdk1.8.0_144-cloudera/lib/tools.jar:/opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1-1.p0.4/lib/hbase/bin/../../opt/cloudera/parcels/CDH-5.11.1-1.cdh5.11.1-1.p0.4/lib/hbase/bin/....
```

edureka

Copyright © edureka-and/or its affiliates. All rights reserved.



X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



Run Kafka Producer

```
Command: kafka-console-producer --broker-list ip-20-0-31-210.ec2.internal:9092 --topic Flume_Topic
```

```
[edureka_249489@ip-20-0-41-190 ~]$ kafka-console-producer --broker-list ip-20-0-31-210.ec2.internal:9092 --topic Flume_Topic
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/KAFKA-3.0.0-1.3.0.0.p0.40/lib/kafka/libs/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/Log4jLoggerFactory.class]
SLF4J: Found binding in [jar:file:/opt/cloudera/parcels/KAFKA-3.0.0-1.3.0.0.p0.40/lib/kafka/libs/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Log4jLoggerFactory.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple\_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
18/02/16 12:57:38 INFO producer.ProducerConfig: ProducerConfig values:
```

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publish...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Presentation

109 / 118

Send Message From Kafka Producer

```
18/02/16 12:57:38 INFO utils.AppInfoParser: Kafka version : 0.11.0-kafka-3.0.0
18/02/16 12:57:38 INFO utils.AppInfoParser: Kafka commitId : unknown
>This is a test for kafka and flume integration
>Welcome to Flume
>
```

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Presentation

Check Flume Agent window

```
18/02/16 12:55:32 WARN consumer.ConsumerConfig: The configuration timeout.ms = 100 was supplied but isn't a known config.
18/02/16 12:55:32 INFO utils.AppInfoParser: Kafka version : 0.9.0-kafka-2.0.2
18/02/16 12:55:32 INFO utils.AppInfoParser: Kafka commitId : unknown
18/02/16 12:55:32 INFO internals.AbstractCoordinator: Discovered coordinator ip-20-0-31-221.ec2.internal:9092 (id: 2147483350) for group flume.
18/02/16 12:55:32 INFO internals.ConsumerCoordinator: Revoking previously assigned partitions [] for group flume.
18/02/16 12:55:32 INFO internals.AbstractCoordinator: (Re-)joining group flume
18/02/16 12:55:33 INFO internals.ConsumerCoordinator: Successfully joined group flume with generation 5
18/02/16 12:55:33 INFO internals.ConsumerCoordinator: Setting newly assigned partitions [Flume_Topic-0] for group flume
18/02/16 12:55:35 INFO kafka.SourceRebalanceListener: topic Flume_Topic - partition 0 assigned.
18/02/16 12:55:35 INFO kafka.KafkaSource: Kafka source source1 started.
18/02/16 12:55:35 INFO Instrumentation.MonitoredCounterGroup: Monitored counter group for type: SOURCE, name: source1: Successfully registered new MBean.
18/02/16 12:55:35 INFO instrumentation.MonitoredCounterGroup: Component type: SOURCE, name: source1 started
18/02/16 12:58:50 INFO hdfs.HDFSDataStream: Serializer = TEXT, UseRawLocalFileSystem = false
18/02/16 12:58:51 INFO hdfs.BucketWriter: Creating hdfs://nameservice1/user/edureka_249489/Flume_Kafka/FlumeData.1518785930893.tmp
18/02/16 12:58:56 INFO hdfs.BucketWriter: Closing hdfs://nameservice1/user/edureka_249489/Flume_Kafka/FlumeData.1518785930893.tmp
18/02/16 12:58:56 INFO hdfs.BucketWriter: Renaming hdfs://nameservice1/user/edureka_249489/Flume_Kafka/FlumeData.1518785930893.tmp to hdfs://nameservice1/user/edureka_249489/Flume_Kafka/FlumeData.1518785930893
18/02/16 12:58:56 INFO hdfs.HDFSEventSink: Writer callback called.
```

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:38 02-06-2023

My Classroom > PySpark Certification Training Course

X PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Browse to the HDFS Directory

HUE File Browser

Name	User	Group	Permissions	Date
hdfs	supergroup	drwxr-xr-x		February 15, 2018 05:50 PM
.	edureka_249489	hadoop	drwx-----	February 16, 2018 04:58 AM
.Trash	edureka_249489	hadoop	drwxr-xr-x	January 31, 2018 11:00 AM
.sparkStaging	edureka_249489	hadoop	drwxr-xr-x	February 03, 2018 09:25 AM
.staging	edureka_249489	hadoop	drwx-----	January 31, 2018 10:05 AM
Flume_Kafka	edureka_249489	hadoop	drwx-----	February 16, 2018 04:58 AM
Flume_tweets	edureka_249489	hadoop	drwx-----	January 30, 2018 11:36 PM

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

38°C Sunny

Search

12:38 02-06-2023

My Classroom > PySpark Certification Training Course

X PySpark Certification T...

Understanding Apache Kafka and Apache Flume > Presentation



e! Course Content

e! Presentation

e! Case Study I - Publis...

e! Case Study II - Collect...

e! Dataset

e! Dataset

e! Apache Spark Streami...

e! Apache Spark Streami...

e! In-class Project

e! Spark GraphX

e! Personal Library

Open the File Created

HUE						
File Browser		edureka_249489				
Actions		Move to trash		History		
Search for file name					Upload	
Home	/ user / edureka_249489 / Flume_Kafka <th data-cs="5" data-kind="parent">Trash</th> <th data-kind="ghost"></th> <th data-kind="ghost"></th> <th data-kind="ghost"></th> <th data-kind="ghost"></th>	Trash				
Name	Size	User	Group	Permissions	Date	
..		edureka_249489	hadoop	drwx----	February 16, 2018 04:58 AM	
.		edureka_249489	hadoop	drwx----	February 16, 2018 04:58 AM	
FlumeData.1518785930893	64 bytes	edureka_249489	hadoop	-rw-r--r--	February 16, 2018 04:58 AM	

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

My Classroom > PySpark Certification Training Course

PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Presentation 113 / 118 ← →

Check Output

The screenshot shows the Hue File Browser interface. The left sidebar has a 'File Browser' section with options: 'View as binary', 'Edit file', 'Download', 'View file location', and 'Refresh'. The main area displays a file named 'FlumeData.1518785930893' with the content: 'This is a test for kafka and flume integration' and 'Welcome to Flume'. A red box highlights this content. At the bottom of the browser window, there's a footer with the 'edureka!' logo and the text 'Copyright © edureka and/or its affiliates. All rights reserved.'

38°C Sunny

Search

12:38 02-06-2023

My Classroom > PySpark Certification Training Course

X PySpark Certification T... Understanding Apache Kafka and Apache Flume > Presentation

Course Content

- Presentation
- Case Study I - Publis...
- Case Study II - Collect...
- Dataset
- Dataset
- Apache Spark Streami...
- Apache Spark Streami...
- In-class Project
- Spark GraphX

Personal Library

Presentation

114 / 118 ← → []

Summary

Challenges with Real-Time Data

Apache Kafka is designed to handle real-time data processing. It can handle large volumes of data in real-time, which makes it ideal for real-time data processing. However, there are some challenges associated with real-time data processing:

- Apache Kafka is designed to handle real-time data processing. It can handle large volumes of data in real-time, which makes it ideal for real-time data processing. However, there are some challenges associated with real-time data processing:
- Apache Kafka is designed to handle real-time data processing. It can handle large volumes of data in real-time, which makes it ideal for real-time data processing. However, there are some challenges associated with real-time data processing:
- Apache Kafka is designed to handle real-time data processing. It can handle large volumes of data in real-time, which makes it ideal for real-time data processing. However, there are some challenges associated with real-time data processing:

Why is Kafka Needed?

Let us see how using big data pipelines makes a system more complicated.

Kafka Cluster

Apache Kafka is a distributed, reliable, and fault-tolerant system for efficiently collecting, aggregating, and moving large amounts of log data from various sources to a distributed database.

Why Flume?

Apache Flume is a distributed, reliable, and fault-tolerant system for efficiently collecting, aggregating, and moving large amounts of log data from various sources to a distributed database.

What is Flume?

Apache Flume is a distributed, reliable, and fault-tolerant system for efficiently collecting, aggregating, and moving large amounts of log data from various sources to a distributed database.

Flume Architecture : Deep Dive

Apache Flume is a distributed, reliable, and fault-tolerant system for efficiently collecting, aggregating, and moving large amounts of log data from various sources to a distributed database.

edureka!

Copyright © edureka and/or its affiliates. All rights reserved.

