edureka!
Discover Learning

APACHE KAFKA
CERTIFICATION TRAINING

Course Curriculum : Your 11 module Learning Plan

https://www.edureka.co/apache-kafka

About Edureka

Edureka is a leading e-learning platform providing live instructor-led interactive online training. We cater to professionals and students across the globe in categories like Big Data & Hadoop, Business Analytics, NoSQL Databases, Java & Mobile Technologies, System Engineering, Project Management and Programming. We have an easy and affordable learning solution that is accessible to millions of learners. With our students spread across countries like the US, India, UK, Canada, Singapore, Australia, Middle East, Brazil and many others, we have built a community of over 1 million learners across the globe.

About Course

Kafka is an open-source stream processing platform. Kafka can be integrated with Spark, Storm and Hadoop. Learn about Kafka Architecture, setup Kafka Cluster, understand Kafka Stream APIs, implement Twitter Streaming with Kafka, Flume, Hadoop and Storm.

Curriculum

Introduction to Big Data and Apache Kafka

Goal: In this module, you will understand where Kafka fits in the Big Data space, and Kafka Architecture. In addition, you will learn about Kafka Cluster, its Components, and how to Configure a Cluster

Skills:

- Kafka Concepts
- Configuring Kafka Cluster
- Kafka Installation

Objectives: At the end of this module, you should be able to:

- Explain what is Big Data
- Describe the need of Kafka
- Understand the role of ZooKeeper
- Classify different type of Kafka Clusters
- Understand why Big Data Analytics is important
- Know the role of each Kafka Components
- Install ZooKeeper and Kafka
- Work with Single Node-Single Broker Cluster

Topics:

- Introduction to Big Data
- Need for Kafka
- Kafka Features
- Kafka Architecture
- ZooKeeper
- Kafka Installation
- Types of Kafka Clusters
- Big Data Analytics
- What is Kafka?
- Kafka Concepts
- Kafka Components
- Where is Kafka Used?
- Kafka Cluster
- Configuring Single Node Single Broker Cluster

Hands on:

- Kafka Installation
- Implementing Single Node-Single Broker Cluster

Kafka Producer

Goal: Kafka Producers send records to topics. The records are sometimes referred to as Messages. In this Module, you will work with different Kafka Producer APIs.

Skills:

- Configure Kafka Producer
- Kafka Producer APIs
- Constructing Kafka Producer
- Handling Partitions

Objectives:

At the end of this module, you should be able to:

- Construct a Kafka Producer
- Send messages Synchronously & Asynchronously
- Serialize Using Apache Avro
- Send messages to Kafka
- Configure Producers
- Create & handle Partitions

Topics:

- Configuring Single Node Multi Broker Cluster
- Sending a Message to Kafka
- Sending a Message Synchronously & Asynchronously
- Serializers
- Partitions
- Constructing a Kafka Producer
- Producing Keyed and Non-Keyed Messages
- Configuring Producers
- Serializing Using Apache Avro

Hands On:

- Working with Single Node Multi Broker Cluster
- Configuring a Kafka Producer
- Creating a Kafka Producer
- Sending a Message Synchronously & Asynchronously

Kafka Consumer

Goal: Applications that need to read data from Kafka use a Kafka Consumer to subscribe to Kafka topics and receive messages from these topics. In this module, you will learn to construct Kafka Consumer, process messages from Kafka with Consumer, run Kafka Consumer and subscribe to Topics

Skills:

- Configure Kafka Consumer
- Constructing Kafka Consumer
- Kafka Consumer API

Objectives: At the end of this module, you should be able to:

- Perform Operations on Kafka
- Explain how Partition Rebalance occurs
- Configure Kafka Consumer
- Describe & implement different Types of Commit
- Define Kafka Consumer and Consumer Groups
- Describe how Partitions are assigned to Kafka Broker
- Create a Kafka consumer and subscribe to Topics
- Deserialize the received messages

Topics:

- Consumers and Consumer Groups
- Consumer Groups and Partition Rebalance
- Subscribing to Topics
- Configuring Consumers
- Standalone Consumer
- Creating a Kafka Consumer
- The Poll Loop
- Commits and Offsets

- Rebalance Listeners
- Deserializers
- Consuming Records with Specific Offsets

Hands-On:

- Creating a Kafka Consumer
- Working with Offsets
- Configuring a Kafka Consumer

Kafka Internals

Goal: Apache Kafka provides a unified, high-throughput, low-latency platform for handling real-time data feeds. Learn more about tuning Kafka to meet your high-performance needs.

Skills:

- Kafka APIs
- Configure Broker
- Kafka Storage

Objectives:

At the end of this module, you should be able to:

- Understand Kafka Internals
- Differentiate between In-sync and Out-off-sync Replicas
- Classify and Describe Requests in Kafka
- Validate System Reliabilities
- Explain how Replication works in Kafka
- Understand the Partition Allocation
- Configure Broker, Producer, and Consumer for a Reliable System
- Configure Kafka for Performance Tuning

Topics:

- Cluster Membership
- Replication
- Physical Storage
- Broker Configuration
- The Controller
- Request Processing
- Reliability
- Using Producers in a Reliable System

- Using Consumers in a Reliable System
- Validating System Reliability
- Performance Tuning in Kafka

Hands On:

- Create topic with partition & replication factor 3 and execute it on multi-broker cluster
- Show fault tolerance by shutting down 1 Broker and serving its partition from another broker

Kafka Cluster Architectures & Administering Kafka

Goal: Kafka Cluster typically consists of multiple brokers to maintain load balance. ZooKeeper is used for managing and coordinating Kafka broker. Learn about Kafka Multi-Cluster Architectures, Kafka Brokers, Topic, Partitions, Consumer Group, Mirroring, and ZooKeeper Coordination in this module.

Skills:

- Administer Kafka

Objectives:

At the end of this module, you should be able to

- Understand Use Cases of Cross-Cluster Mirroring
- Learn Multi-cluster Architectures
- Explain Apache Kafka’s MirrorMaker
- Perform Topic Operations
- Understand Consumer Groups
- Describe Dynamic Configuration Changes
- Learn Partition Management
- Understand Consuming and Producing
- Explain Unsafe Operations

Topics:

- Use Cases - Cross-Cluster Mirroring
- Multi-Cluster Architectures
- Apache Kafka’s MirrorMaker
- Other Cross-Cluster Mirroring Solutions
- Topic Operations
- Consumer Groups

- Dynamic Configuration Changes
- Consuming and Producing
- Partition Management
- Unsafe Operations

Hands on:

- Topic Operations
- Partition Operations
- Consumer Group Operations
- Consumer and Producer Operations

Kafka Monitoring and Kafka Connect

Goal: Learn about the Kafka Connect API and Kafka Monitoring. Kafka Connect is a scalable tool for reliably streaming data between Apache Kafka and other systems.

Skills:

- Kafka Connect
- Monitoring Kafka
- Metrics Concepts

Objectives: At the end of this module, you should be able to:

- Explain the Metrics of Kafka Monitoring
- Build Data pipelines using Kafka Connect
- Perform File source and sink using Kafka Connect
- Understand Kafka Connect
- Understand when to use Kafka Connect vs Producer/Consumer API

- **Topics:**
- Metric Basics
- Client Monitoring
- End-to-End Monitoring
- When to Use Kafka Connect?
- Considerations When Building Data Pipelines
- Kafka Broker Metrics
- Lag Monitoring
- Kafka Connect
- Kafka Connect Properties

Hands on:

- Kafka Connect

Kafka Stream Processing

Goal: Learn about the Kafka Streams API in this module. Kafka Streams is a client library for building mission-critical real-time applications and microservices, where the input and/or output data is stored in Kafka Clusters.

Skills:

- Stream Processing using Kafka

Objectives:

- At the end of this module, you should be able to,
- Learn Different types of Programming Paradigm
- Explain Kafka Streams & Kafka Streams API
- Describe What is Stream Processing
- Describe Stream Processing Design Patterns

Topics:

- Stream Processing
- Stream-Processing Design Patterns
- Kafka Streams: Architecture Overview
- Stream-Processing Concepts
- Kafka Streams by Example

Hands on:

- Kafka Streams
- Word Count Stream Processing

Integration of Kafka With Hadoop, Storm and Spark

Goal: In this module, you will learn about Apache Hadoop, Hadoop Architecture, Apache Storm, Storm Configuration, and Spark Ecosystem. In addition, you will configure Spark Cluster, Integrate Kafka with Hadoop, Storm, and Spark.

Skills:

- Kafka Integration with Hadoop
- Kafka Integration with Storm

- Kafka Integration with Spark

Objectives:

At the end of this module, you will be able to:

- Understand What is Hadoop
- Integrate Kafka with Hadoop
- Explain Storm Components
- Understand What is Spark
- Explain Spark Components
- Explain Hadoop 2.x Core Components
- Understand What is Apache Storm
- Integrate Kafka with Storm
- Describe RDDs
- Integrate Kafka with Spark

Topics:

- Apache Hadoop Basics
- Kafka Integration with Hadoop
- Configuration of Storm
- Apache Spark Basics
- Kafka Integration with Spark
- Hadoop Configuration
- Apache Storm Basics
- Integration of Kafka with Storm
- Spark Configuration

Hands On:

- Kafka integration with Hadoop
- Kafka integration with Spark
- Kafka integration with Storm

Integration of Kafka With Talend and Cassandra

Goal: Learn how to integrate Kafka with Flume, Cassandra and Talend.

Skills:

- Kafka Integration with Flume
- Kafka Integration with Talend
- Kafka Integration with Cassandra

Objectives:

At the end of this module, you should be able to,

- Understand Flume
- Setup a Flume Agent
- Understand Cassandra
- Create a Keyspace in Cassandra
- Understand Talend
- Integrate Kafka with Talend
- Explain Flume Architecture and its Components
- Integrate Kafka with Flume
- Learn Cassandra Database Elements
- Integrate Kafka with Cassandra
- Create Talend Jobs

Topics:

- Flume Basics
- Cassandra Basics such as and KeySpace and Table Creation
- Talend Basics
- Integration of Kafka with Flume
- Integration of Kafka with Cassandra
- Integration of Kafka with Talend

Hands On:

- Kafka demo with Flume
- Kafka demo with Talend
- Kafka demo with Cassandra

Kafka In-Class Project

Goal: In this module, you will work on a project, which will be gathering messages from multiple

sources.

Scenario:

In E-commerce industry, you must have seen how catalog changes frequently. Most deadly problem they face is "How to make their inventory and price

consistent?".

There are various places where price reflects on Amazon, Flipkart or Snapdeal. If you will visit Search page, Product Description page or any ads on Facebook/google. You will find there are some mismatch in price and availability. If we see user point of view that's very disappointing because he spends more time to find better products and at last if he doesn't purchase just because of consistency.

Here you have to build a system which should be consistent in nature. For example, if you are getting product feeds either through flat file or any event

stream you have to make sure you don't lose any events related to product specially inventory and price.

If we talk about price and availability it should always be consistent because there might be possibility that the product is sold or the seller doesn't want to sell it anymore or any other reason. However, attributes like Name, description doesn't make that much noise if not updated on time.

Problem Statement

You have given set of sample products. You have to consume and push products to Cassandra/MySQL once we get products in the consumer. You have to save below-mentioned fields in Cassandra.

1. PogId

2. Supc

3. Brand

4. Description

5. Size

6. Category

7. Sub Category

8. Country

9. Seller Code

In MySQL, you have to store

1. PogId

2. Supc

3. Price

4. Quantity

Certification Project

This Project enables you to gain Hands-On experience on the concepts that you have learned as part of this Course.

You can email the solution to our Support team within 2 weeks from the Course Completion Date. Edureka will evaluate the solution and award a Certificate with a Performance-based Grading.

Problem Statement:

You are working for a website techreview.com that provides reviews for different technologies. The company has decided to include a new feature in the website which will allow users to compare the popularity or trend of multiple technologies based on twitter feeds. They want this comparison to happen in real time. So, as a big data developer of the company, you have been task to implement following things:

- Near Real Time Streaming of the data from Twitter for displaying last minute's count of people tweeting about a particular technology.

- Store the twitter count data into Cassandra.

Project

What are the system requirements for this course?

- Minimum RAM required: 4GB (Suggested: 8GB)
- Minimum Processor i3 or above
- Participant’s machines must support a 64-bit VirtualBox guest image.
- Minimum Free Disk Space: 25GB
- Operating System of 64bit

How will I execute the practicals?

We will help you to setup Edureka's Virtual Machine in your System with local access. The detailed installation guides are provided in the LMS for setting up the environment. For any doubt, the 24*7 support team will promptly assist you. Edureka Virtual Machine can be installed on Mac or Windows machine.

Which case studies will be a part of the course?

Case Study 1:

Stock Profit Ltd, India’s first discount broker, offers zero brokerage & unlimited online share trading in Equity Cash. Design a system to capture real-time stocks data from source (i.e. Yahoo.com) and calculate the profit and loss for customers who are subscribed to the tool. Finally, store the result in HDFS.

Case Study 2:

You are a SEO specialist in a company. You get an email from management wherein the requirement is to get Top Trending Keywords. You have to write the topology which can consume keywords from Kafka. You have given a file containing various search keywords across multiple verticals.

Case Study 3:

You have to build a system which should be consistent in nature. For example, if you are getting product feeds either through flat file or any event stream you have to make sure you don’t lose any events related to product specially inventory and price.

If we talk about price and availability it should always be consistent because there might be possibility that product is sold or seller doesn’t want to sell it anymore or any other reason. However, attributes like Name, description doesn’t make that much noise if not updated on time.

Case Study 4:

John wants to build an e-commerce portal like Amazon, Flipkart or Paytm. He will ask sellers/local brands to upload all their products on the portal so that users/buyers can visit portal online and purchase. John doesn’t have much knowledge about the system and he hired you to build a reliable and scalable solution for him where buyers and sellers can easily update their products.