Project -1

**Project Title:**

Real-Time Twitter Sentiment Analysis using Spark Structured Streaming, Kafka, and Scala

**Project Description:**

This project builds an end-to-end real-time streaming data pipeline to ingest live Twitter feeds via Kafka, process the tweets in Spark Structured Streaming using Scala, perform sentiment classification based on Natural Language Processing (NLP) techniques, and store the results in a persistent data store (e.g., Delta Lake or Elasticsearch) for analytics. The system is designed to handle continuous high-velocity data, provide near real-time insights into public sentiment on trending topics, and integrate with a visualization dashboard for monitoring sentiment trends over time.

**Objective:**

Create a scalable real-time data pipeline using Spark Structured Streaming and Scala.

Process and clean incoming unstructured tweet text data.

Apply sentiment analysis ML model to classify tweets as Positive, Negative, or Neutral in real time.

Persist processed results to a database or Delta Lake for analytics.

Enable sentiment trend visualization on a dashboard for actionable insights.

**Tools Used:**

Programming: Scala

Streaming Framework: Apache Spark Structured Streaming

Message Broker: Apache Kafka

ML & NLP: Spark MLlib, NLP libraries

Data Storage: HDFS/S3/Delta Lake / Elasticsearch / MongoDB

Data Visualization: Kibana / Grafana

Environment: Hadoop/Spark cluster or local setup with Docker

Weeks [during training]: week4 and week 5

**Project Type:**

Real-Time Big Data Streaming & Analytics Project

**Outcome:**

By the end of the project, a working real-time sentiment analysis system will be deployed that can ingest live tweets, classify them by sentiment within seconds, and display evolving sentiment trends on a visual dashboard.

Project - 2

**Project Title:**Real-Time Log Processing and Analytics for Application Monitoring Using Spark and Scala

**Project Description:**This project involves building a scalable, fault-tolerant real-time data processing pipeline that ingests live application and server log data, processes it using Apache Spark Structured Streaming with Scala, and produces near real-time analytics to monitor application performance and detect anomalies or errors. The system consumes log streams (from sources like Kafka or file streams), applies parsing, filtering, windowed aggregations, and stores processed data in a sink like Delta Lake, Elasticsearch, or a database for querying and visualization.

The pipeline enables IT teams and DevOps to monitor application health by tracking error rates, request volumes, response times, and user behavior continuously, with the ability to detect and react to issues quickly.

**Objective:**

* Implement a real-time log ingestion mechanism using Spark Structured Streaming with Scala.
* Parse raw log lines into structured format for easy querying and analysis.
* Apply filters to identify error and warning messages dynamically.
* Perform windowed aggregations (e.g., counts of error types over time windows).
* Store streaming results in a scalable data sink (Delta Lake, Elasticsearch) for dashboards.
* Enable alerting or notifications by integrating with monitoring tools (optional).
* Ensure fault tolerance and exactly-once data processing by leveraging Spark checkpointing.

**Tools Used:**

* Programming Language: Scala
* Streaming Engine: Apache Spark Structured Streaming
* Message Broker (optional): Apache Kafka for log ingestion
* Data Storage: Delta Lake, Elasticsearch, or HDFS
* Visualization: Kibana, Grafana, or custom dashboards
* Cluster Environment: Hadoop or Spark standalone cluster / cloud
* Logging: Spark's internal logging combined with application-level log parsing

Weeks [during training]:week 3 ,week4

**Project Type:**  
Real-Time Big Data Streaming and Analytics Project

**Outcome:**  
At project completion, a fully functional real-time log analysis pipeline will be deployed that converts raw logs into actionable analytics with low latency. The system will demonstrate the ability to detect anomalies, track application health metrics, and provide a foundation for IT monitoring and operational insights using Spark and Scala stream processing techniques.

Project -3

**Project Title:**

Real-Time Fraud Detection Using Spark Streaming and Kafka

**Project Description:**

This project builds a real-time data processing system that ingests transactional or event data from Kafka topics, processes them with Apache Spark Structured Streaming using Scala, applies fraud detection logic (rules or machine learning models), and outputs suspicious transactions for alerting or further investigation. It is designed to handle high-throughput streaming data with low latency, enabling rapidly evolving fraud patterns to be detected and mitigated as transactions occur. The pipeline supports fault tolerance, scalability, and exactly-once processing semantics.

**Objective:**

* Design a scalable streaming pipeline for ingesting transaction/event data from Kafka using Spark Structured Streaming in Scala.
* Implement data validation, cleansing, and feature extraction in streaming transformations.
* Apply fraud detection logic, which can range from rule-based filters to real-time machine learning model inference.
* Continuously flag suspicious transactions and push alerts or store results in a persistent sink (NoSQL store, Delta Lake, etc.).
* Ensure data consistency and fault tolerance with checkpointing and exactly-once guarantees.
* Provide metrics and dashboards for fraud monitoring and operational insight.

**Tools Used:**

* Programming Language: Scala
* Streaming Framework: Apache Spark Structured Streaming
* Message Broker: Apache Kafka
* Modeling & ML: Spark MLlib or external ML serving (optional)
* Storage: Delta Lake, Cassandra, HBase, or Elasticsearch for persistent storage and querying
* Dashboard/Alerting: Kibana, Grafana, or custom alerting services
* Cluster Environment: YARN, Kubernetes, or standalone Spark Cluster

Weeks [during training]: week 4 and week 5

**Project Type:**

Real-Time Big Data Streaming and Fraud Analytics Project

**Outcome:**

By the project end, you will have a fully functional streaming fraud detection system that takes live data from Kafka, processes and analyzes it in near real-time using Spark Structured Streaming with Scala, flags potential fraudulent events, and stores these for analysis and alerting. This will demonstrate capabilities in building robust streaming ETL pipelines, applying real-time analytics with machine learning, and integrating streaming data architectures.