## Course duration

- 4 days

## Course Benefits

- Understand the need for Spark in data processing
- Understand the Spark architecture and how it distributes computations to cluster nodes
- Be familiar with basic installation / setup / layout of Spark
- Use the Spark shell for interactive and ad-hoc operations
- Understand RDDs (Resilient Distributed Datasets), and data partitioning, pipelining, and computations
- Understand/use RDD ops such as map(), filter() and others.
- Understand and use Spark SQL and the DataFrame/DataSet API.
- Understand DataSet/DataFrame capabilities, including the Catalyst query optimizer and Tungsten memory/cpu optimizations.
- Be familiar with performance issues, and use the DataSet/DataFrame and Spark SQL for efficient computations
- Understand Spark's data caching and use it for efficient data transfer
- Write/run standalone Spark programs with the Spark API
- Use Spark Streaming / Structured Streaming to process streaming (real-time) data
- Ingest streaming data from Kafka, and process via Spark Structured Streaming
- Understand performance implications and optimizations when using Spark

## Course Outline

1. (Optional): Scala Ramp Up
    1. Scala Introduction, Variables, Data Types, Control Flow
    2. The Scala Interpreter
    3. Collections and their Standard Methods (e.g. map())
    4. Functions, Methods, Function Literals
    5. Class, Object, Trait, case Class
2. Introduction to Spark
    1. Overview, Motivations, Spark Systems
    2. Spark Ecosystem
    3. Spark vs. Hadoop
    4. Acquiring and Installing Spark
    5. The Spark Shell, SparkContext
3. Session 3: RDDs and Spark Architecture
    1. RDD Concepts, Lifecycle, Lazy Evaluation
    2. RDD Partitioning and Transformations
    3. Working with RDDs - Creating and Transforming (map, filter, etc.)

4. Spark SQL, DataFrames, and DataSets
    1. Overview
    2. SparkSession, Loading/Saving Data, Data Formats (JSON, CSV, Parquet, text ...)
    3. Introducing DataFrames and DataSets (Creation and Schema Inference)
    4. Supported Data Formats (JSON, Text, CSV, Parquet)
    5. Working with the DataFrame (untyped) Query DSL (Column, Filtering, Grouping, Aggregation)
    6. SQL-based Queries
    7. Working with the DataSet (typed) API
    8. Mapping and Splitting (flatMap(), explode(), and split())
    9. DataSets vs. DataFrames vs. RDDs
5. Shuffling Transformations and Performance
    1. Grouping, Reducing, Joining
    2. Shuffling, Narrow vs. Wide Dependencies, and Performance Implications
    3. Exploring the Catalyst Query Optimizer (explain(), Query Plans, Issues with lambdas)
    4. The Tungsten Optimizer (Binary Format, Cache Awareness, Whole-Stage Code Gen)
6. Performance Tuning
    1. Caching - Concepts, Storage Type, Guidelines
    2. Minimizing Shuffling for Increased Performance
    3. Using Broadcast Variables and Accumulators
    4. General Performance Guidelines
7. Creating Standalone Applications
    1. Core API, SparkSession.Builder
    2. Configuring and Creating a SparkSession
    3. Building and Running Applications - sbt/build.sbt and spark-submit
    4. Application Lifecycle (Driver, Executors, and Tasks)
    5. Cluster Managers (Standalone, YARN, Mesos)
    6. Logging and Debugging
8. Spark Streaming
    1. Introduction and Streaming Basics
    2. Spark Streaming (Spark 1.0+)
    3. DStreams, Receivers, Batching
    4. Stateless Transformation
    5. Windowed Transformation
    6. Stateful Transformation
    7. Structured Streaming (Spark 2+)
    8. Continuous Applications
    9. Table Paradigm, Result Table
    10. Steps for Structured Streaming
    11. Sources and Sinks
    12. Consuming Kafka Data
    13. Kafka Overview
    14. Structured Streaming - "kafka" format
    15. Processing the Stream

## Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this Spark class:

- Working knowledge of some programming language - no Java experience needed