DATA SCIENCE | ADVANCED | 5 DAYS

Apache Spark with Scala

Build a strong foundation in big data processing using Apache Spark and Scala by understanding Spark architecture and core APIs. Learn how to process large datasets, apply advanced analytics and streaming, optimize performance, and integrate Spark with diverse data sources for scalable data driven solutions.

Why Spark with Scala?

High Performance

By leveraging Scala's concise syntax and functional programming features, your business can achieve superior performance and speed in processing and analysing large datasets.

Advanced Analytics

Apache Spark with Scala provides a rich set of libraries and APIs for advanced analytics. These capabilities empower your business to gain valuable insights, make data-driven decisions, and uncover hidden patterns and trends in your data.

Scalability

Spark's ability to distribute data and computations across multiple nodes allows your business to scale seamlessly as data volumes grow.

What you'll be able to do

- Gain a comprehensive understanding of the Apache Spark framework, its architecture, and its components.
- Acquire proficiency in the Scala programming language, including its syntax, features, and functional programming concepts.
- Learn how to process and manipulate large datasets using Spark's core APIs and RDDs.
- Explore Spark's advanced analytics capabilities, including machine learning, graph processing, and stream processing.
- Understand techniques and best practices for optimising Spark performance.
- Discover how to integrate Spark with various data sources, including Hadoop Distributed File System (HDFS), Apache Hive, and other popular data storage systems.
- Explore Spark's capabilities for real-time data processing and stream processing.
- Gain familiarity with the broader Spark ecosystem and related technologies.

Who this is for

- Data Engineers
- Data Scientists
- Data Analysts
- Software Engineers

Curriculum

01 Spark and Scala Foundations

- Position Spark for large-scale data processing
- Use essential Scala syntax for Spark jobs
- Set up and navigate a Spark development workflow
- Contrast Spark with single-node data tools

02 Spark Architecture and RDDs

- Explain driver, executors, and cluster roles
- Work with RDDs as the core distributed abstraction
- Understand lineage, partitions, and fault tolerance
- Choose when RDDs vs higher-level APIs fit

03 Transformations and Actions

- Apply common transformations for ETL-style processing
- Trigger actions and understand lazy evaluation
- Persist intermediate results intentionally
- **Hands-on:** Build a multi-step RDD/DataFrame job

04 Spark SQL, DataFrames, and Datasets

- Query structured data with Spark SQL
- Use DataFrames/Datasets for typed analytics
- Read and write common file formats
- Optimize tabular workflows with Catalyst basics

05 Spark Streaming and Real-Time Analytics

- Ingest streaming sources for near-real-time pipelines
- Apply windowed aggregations and stateful patterns
- Design for late data and exactly-once concerns
- Build a simple streaming analytics flow

06 MLlib and ML Pipelines

- Use MLlib for scalable machine learning
- Assemble featurization and model stages as pipelines
- Evaluate models in distributed settings
- Export pipeline stages for reuse

07 GraphX and Advanced Analytics

- Model graph structures for relationship analytics
- Run graph algorithms for influence and connectivity
- Combine graph and tabular analysis patterns
- Identify practical GraphX use cases and limits

08 Data Integration and the Big Data Ecosystem

- Integrate Spark with common storage and messaging systems
- Position Spark alongside Hadoop/cloud lake tooling
- Design batch and streaming ingestion boundaries
- Standardize schemas across pipeline stages

09 Performance Optimization

- Diagnose shuffles, skew, and partition problems
- Tune memory, caching, and join strategies
- Use Spark UI to find bottlenecks
- Apply a performance checklist before production

10 Deployment, Monitoring, Testing, and Debugging

- Deploy Spark jobs to cluster environments
- Monitor job health and resource usage
- Test and debug distributed applications systematically
- Establish operational runbooks for failures

11 Distributed Machine Learning

- Scale ML workflows across a Spark cluster
- Compare distributed training patterns and trade-offs
- Productionize feature and model pipelines
- Review end-to-end Spark ML architecture choices