Certified Big Data Hadoop and Spark Scala Course – Curriculum

The Certified Big Data Hadoop and Spark Scala course by DataFlair is a perfect blend of in-depth theoretical knowledge and strong practical skills via implementation of real life projects to give you a headstart and enable you to bag top Big Data jobs in the industry.

Course duration: 70 Hours

Part 1 - Big Data and Hadoop:

Module 1: The big picture of Big Data

This module introduces you to the rudiments of Big Data (data sets too large/ complex for traditional data processing application software), why we choose to adopt it, and its different dimensions and implementations. It also discusses its future in the industry.

- What is Big Data
- Necessity of Big Data in the industry
- Paradigm shift - why the industry is shifting to Big Data tools
- Different dimensions of Big Data
- Data explosion in the industry
- Various implementations of Big Data
- Different technologies to handle Big Data
- Traditional systems and associated problems
- Future of Big Data in the IT industry

Module 2: Demystifying Hadoop

We then introduce you to the Hadoop framework, its architecture and design principles, and its ingredients. This familiarizes you with the Hadoop ecosystem and its components. Finally, we talk of various flavors of Hadoop.

- Why Hadoop is at the heart of every Big Data solution
- Introduction to the Hadoop framework
- Hadoop architecture and design principles
- Ingredients of Hadoop
- Hadoop characteristics and data-flow
- Components of the Hadoop ecosystem
- Hadoop Flavors – Apache, Cloudera, Hortonworks, and more

Module 3: Setup and Installation of Hadoop

This module deals with setting up and installing both single- and multi-node clusters. It teaches you to configure Hadoop, run it in various modes, and troubleshoot problems observed. You will also learn how to configure masters and slaves on the cluster.

Setup and Installation of single-node Hadoop cluster

- Hadoop environment setup and pre-requisites
- Installation and configuration of Hadoop
- Working with Hadoop in pseudo-distributed mode
- Troubleshooting encountered problems

Setup and Installation of Hadoop multi-node cluster

- Hadoop environment setup on the cloud (Amazon cloud)
- Installation of Hadoop pre-requisites on all nodes
- Configuration of masters and slaves on the cluster
- Playing with Hadoop in distributed mode

Module 4: HDFS – The Storage Layer

Next, we discuss HDFS(Hadoop Distributed File System), its architecture and mechanisms, and its characteristics and design principles. We also take a good look at HDFS masters and slaves. Finally, we discuss terminologies and some best practices.

- What is HDFS (Hadoop Distributed File System)
- HDFS daemons and architecture
- HDFS data flow and storage mechanism
- Hadoop HDFS characteristics and design principles
- Responsibility of HDFS Master – NameNode
- Storage mechanism of Hadoop meta-data
- Work of HDFS Slaves – DataNodes
- Data Blocks and distributed storage
- Replication of blocks, reliability, and high availability
- Rack-awareness, scalability, and other features
- Different HDFS APIs and terminologies
- Commissioning of nodes and addition of more nodes
- Expanding clusters in real-time
- Hadoop HDFS Web UI and HDFS explorer

HDFS best practices and hardware discussion

Module 5: A Deep Dive into MapReduce

After finishing this module, you will be comfortable with MapReduce, the processing layer of Hadoop, and will be aware of its need, components, and terminologies. MapReduce lets you process and generate big data sets with a parallel, distributed algorithm on a cluster with map and reduce methods. We will demonstrate using examples as we move on to optimization of MapReduce jobs and will introduce you to combiners as we move on to the next module.

- What is MapReduce, the processing layer of Hadoop
- The need for a distributed processing framework

- Issues before MapReduce and its evolution
- List processing concepts
- Components of MapReduce – Mapper and Reducer
- MapReduce terminologies- keys, values, lists, and more
- Hadoop MapReduce execution flow
- Mapping and reducing data based on keys
- MapReduce word-count example to understand the flow
- Execution of Map and Reduce together
- Controlling the flow of mappers and reducers
- Optimization of MapReduce Jobs
- Fault-tolerance and data locality
- Working with map-only jobs
- Introduction to Combiners in MapReduce
- How MR jobs can be optimized using combiners

Module 6: MapReduce – Advanced Concepts

Time to dig deeper into MapReduce! This module takes you to more advanced concepts of MapReduce- those like its data types and constructs like InputFormat and RecordReader.

- Anatomy of MapReduce
- Hadoop MapReduce data types
- Developing custom data types using Writable & WritableComparable
- InputFormat in MapReduce
- InputSplit as a unit of work
- How Partitioners partition data
- Customization of RecordReader
- Moving data from mapper to reducer – shuffling & sorting
- Distributed cache and job chaining
- Different Hadoop case-studies to customize each component
- Job scheduling in MapReduce

Module 7: Hive – Data Analysis Tool

Halfway through the course now, we begin to explore Hive, a data warehouse software project. We take a look at its architecture, various DDL and DML operations, and meta-stores. Then, we talk of where this would be useful. Finishing this module, you will be able to perform data query and analysis.

- The need for an adhoc SQL based solution – Apache Hive
- Introduction to and architecture of Hadoop Hive
- Playing with the Hive shell and running HQL queries
- Hive DDL and DML operations
- Hive execution flow
- Schema design and other Hive operations

- Schema-on-Read vs Schema-on-Write in Hive
- Meta-store management and the need for RDBMS
- Limitations of the default meta-store
- Using SerDe to handle different types of data
- Optimization of performance using partitioning
- Different Hive applications and use cases

Module 8: Pig – Data Analysis Tool

This module teaches you all about Pig, a high-level platform for developing programs for Hadoop. We will take a look at its execution flow and various operations, and will then compare it to MapReduce. Pig can execute its jobs in MapReduce.

- The need for a high level query language - Apache Pig
- How Pig complements Hadoop with a scripting language
- What is Pig
- Pig execution flow
- Different Pig operations like filter and join
- Compilation of Pig code into MapReduce
- Comparison - Pig vs MapReduce

Module 9: NoSQL Database – HBase

We move on to HBase, an open-source, non-relational, distributed NoSQL database. In this module, we talk of its rudiments, architecture, datastores, and the Master and Slave model. We also compare it to both HDFS and RDBMS. Finally, we discuss data access mechanisms.

- NoSQL databases and their need in the industry
- Introduction to Apache HBase
- Internals of the HBase architecture
- The HBase Master and Slave Model
- Column-oriented, 3-dimensional, schema-less datastores
- Data modeling in Hadoop HBase
- Storing multiple versions of data
- Data high-availability and reliability
- Comparison - HBase vs HDFS
- Comparison - HBase vs RDBMS
- Data access mechanisms
- Working with HBase using the shell

Module 10: Data Collection using Sqoop

With Apache Sqoop, you can always go about another helping of data from a relational database into Hadoop or the other way around. This is a command-line interface application.

- The need for Apache Sqoop

- Introduction and working of Sqoop
- Importing data from RDBMS to HDFS
- Exporting data to RDBMS from HDFS
- Conversion of data import/export queries into MapReduce jobs

Module 11: Data Collection using Flume

Apache Flume is a reliable distributed software that lets us efficiently collect, aggregate, and move large amounts of log data. Here, we talk about its architecture and various tools it has to offer.

- What is Apache Flume
- Flume architecture and aggregation flow
- Understanding Flume components like data Sources and Sinks
- Flume channels to buffer events
- Reliable & scalable data collection tools
- Aggregating streams using Fan-in
- Separating streams using Fan-out
- Internals of the agent architecture
- Production architecture of Flume
- Collecting data from different sources to Hadoop HDFS
- Multi-tier Flume flow for collection of volumes of data using AVRO

Module 12: Apache YARN & advanced concepts in the latest version

Version 2 of Hadoop brought with it Yet Another Resource Negotiator (YARN). It will allow you to efficiently allocate resources.

- The need for and the evolution of YARN
- YARN and its eco-system
- YARN daemon architecture
- Master of YARN – Resource Manager
- Slave of YARN – Node Manager
- Requesting resources from the application master
- Dynamic slots (containers)
- Application execution flow
- MapReduce version 2 application over Yarn
- Hadoop Federation and Namenode HA

Part 2 – Spark and Scala:

Module 1: Exploring Scala

This module introduces you to the rudiments of Scala, how to get it up and running, its functions and procedures, and different operations with APIs for those.

- What is Scala
- Setup and configuration of Scala
- Developing and running basic Scala Programs
- Scala operations
- Functions and procedures in Scala
- Different Scala APIs for common operations
- Loops and collections- Array, Map, Lists, Tuples
- Pattern matching for advanced operations
- Eclipse with Scala

Module 2: Object-Oriented and Functional Programming

We then introduce you to the concepts of object-oriented programming and their constructs- nested classes, constructors, and more. Finally, we will take a good look at call-by-name and call-by-value.

- Introduction to object-oriented programming
- Different OOPS concepts
- Constructor, getter, setter, singleton, overloading, and overriding
- Nested Classes and visibility rules
- Functional structures
- Functional programming constructs
- Call by Name, Call by Value

Module 3: Big Data and the need for Spark

This module deals with the challenges to older Big Data solutions and introduces you to the alternatives. We discuss the limitations of each of those.

- Introduction to Big Data
- Challenges to old Big Data solutions
- Batch vs Real-time vs in-Memory processing
- MapReduce and its limitations
- Apache Storm and its limitations
- Need for a general purpose solution - Apache Spark

Module 4: Diving deep into Apache Spark

Next, we discuss Spark and its components. This is much about its features and design principles.

- What is Apache Spark?
- Components of Spark architecture
- Apache Spark design principles
- Spark features and characteristics
- Apache Spark ecosystem components and their insights

Module 5: Deploying Spark in local mode

After finishing this module, you will be comfortable with Spark and its structures. This module deals with setting it up on your machine in different modes. This will also guide you with issues you will likely encounter.

- Setting up the Spark Environment
- Installing and configuring prerequisites
- Installing Apache Spark in local mode
- Working with Spark in local mode
- Troubleshooting encountered problems in Spark

Module 6: Deploying Spark in different modes

Time to dig deeper into Spark! This module tells you about many more modes to install Spark in.

- Installing Spark in standalone mode
- Installing Spark in YARN mode
- Installing & configuring Spark on a real multi-node cluster
- Playing with Spark in cluster mode
- Best practices for Spark deployment

Module 7: Demystifying Apache Spark

More than halfway through the course now, we begin to demystify Spark. We take you right to the Spark shell so you can expect a full hands-on experience.

- Playing with the Spark shell
- Executing Scala and Java statements in the shell
- Understanding the Spark context and driver
- Reading data from the local filesystem
- Integrating Spark with HDFS
- Caching the data in memory for further use
- Distributed persistence
- Testing and troubleshooting

Module 8: Basic abstraction RDDs

This module teaches you all about RDDs in Spark. You will learn about the operations, transformations, and fault tolerance.

- What is an RDD in Spark
- How do RDDs make Spark a feature-rich framework
- Transformations in Apache Spark RDDs
- Spark RDD action and persistence
- Spark Lazy Operations - Transformation and Caching

- Fault tolerance in Spark
- Loading data and creating RDD in Spark
- Persist RDD in memory or disk
- Pair operations and key-value in Spark
- Spark integration with Hadoop
- Apache Spark practicals and workshops

Module 9: Spark Streaming

We move on to Spark Streaming. In this module, we talk of its need, operations, and execution flow. Finally, we discuss ways to optimize performance.

- The need for stream analytics
- Comparison with Storm and S4
- Real-time data processing using Spark streaming
- Fault tolerance and check-pointing
- Stateful stream processing
- DStream and window operations
- Spark Stream execution flow
- Connection to various source systems
- Performance optimizations in Spark

Module 10: Spark MLlib and Spark GraphX

Before we move on to the final project of this course, let's learn about machine learning and its libraries with Spark. Algorithms like clustering and classification form a perfect fit for this purpose.

- Why Machine Learning is needed
- What is Spark Machine Learning
- Various Spark ML libraries
- Algorithms for clustering, statistical analytics, classification etc.
- What is GraphX
- The need for different graph processing engines
- Graph handling using Apache Spark

Module 11: Spark SQL

This module familiarizes you with Spark SQL and explains its features, components, and techniques. We also talk about Data-frames and Hive queries.

- What is Spark SQL
- Apache Spark SQL features and data flow
- Spark SQL architecture and components
- Hive and Spark SQL together
- Play with Data-frames and data states

- Data loading techniques in Spark
- Hive queries through Spark
- Various Spark SQL DDL and DML operations
- Performance tuning in Spark

Module 12: Real Life Hadoop and Spark Project

We conclude this course with a live Hadoop and Spark project to prepare you for the industry. Here, we make use of various constructs of Hadoop and Spark to solve real-world problems in Big Data Analytics.

- *Web Analytics* - Weblogs are web server logs where web servers like Apache record all events along with a remote IP, timestamp, requested resource, referral, user agent, and other such data. The objective is to analyze weblogs to generate insights like user navigation patterns, top referral sites, and highest/lowest traffic-times.
- *Sentiment Analysis* - Sentiment analysis is the analysis of people's opinions, sentiments, evaluations, appraisals, attitudes, and emotions in relation to entities like individuals, products, events, services, organizations, and topics. It is achieved by classifying the observed expressions as opinions positive or negative.
- *Crime Analysis* - Learn to analyze US crime data and find the most crime-prone areas along with the time of crime and its type. The objective is to analyze crime data and generate patterns like time of crime, district, type of crime, latitude, and longitude. This is to ensure that additional security measures can be taken in crime-prone areas.
- *IVR Data Analysis* - Learn to analyze IVR(Interactive Voice Response) data and use it to generate multiple insights. IVR call records are meticulously analyzed to help with optimization of the IVR system in an effort to ensure that maximum calls complete at the IVR itself, leaving no room for the need for a call-center.
- *Titanic Data Analysis* - Titanic was one of the most colossal disasters in the history of mankind, and it happened because of both natural events and human mistakes. The objective of this project is to analyze multiple Titanic data sets to generate essential insights pertaining to age, gender, survived, class, and embarked.
- *Amazon Data Analysis* - Amazon data sets are made of users' reviews and ratings of products and services. Analyzing review data, companies attempt to process the sentiments of their users regarding their products to help improve the same.
- *Set Top Box Data Analysis* - Learn to analyze Set-Top-Box data and generate insights about smart tv usage patterns. Analyze set top box media data and generate patterns of channel navigation and VOD. This Spark Project includes details about users' activities tuning a channel or duration, browsing for videos, or purchasing videos using VOD.
- *YouTube Data Analysis* - Yearn to analyze YouTube Data and generate insights like the 10 topmost videos in various categories, user demographics, no. of views, ratings and such. The data holds fields like id, age, category, length, views, ratings, and comments.
- And so many more projects of retail, telecom, media, etc..