**SoftLogic**
Systems  *We Build Your Future*

# HADOOP / BIG DATA TRAINING SYLLABUS

**WWW.SOFTLOGICSYS.COM** | **+91 86816 84318**

## Hadoop / Big Data Training Syllabus

SLA is one of its kind Institute which not only equips you in technology skills but will train you for free in Aptitude skills, Soft Skills, Mock Interviews, Interview Skills, Work ethics and Corporate Values that you need to know to get into IT Industry. Our training is given by expert real time development experienced trainers and we enable each and every student of ours to do their own real time projects by the end of the program.

We do not buy you a job by bribing companies to secure a place in IT, instead we equip you with the skills needed to get employed in IT and will support you with unlimited number of relevant interview opportunities so that your career in IT becomes assured. Our support will be there until you get placed in an IT Company as that's our mission too.

If you want to just have an IT certification, you can do your course anywhere. If you aspire to get into an IT Job, then you should choose SLA. With SLA, your IT dream will definitely come into reality.

Please go through the long list of our student reviews / offer letters @ www.joinsla.com  to get to know more about us.

### All-in-One Javascript Development Suite

### Big Data Introduction:
- ❖ What is Big Data
- ❖ Evolution of Big Data
- ❖ Benefits of Big Data
- ❖ Operational vs Analytical Big Data
- ❖ Need for Big Data Analytics
- ❖ Big Data Challenges

### Hadoop cluster:
- ❖ Master Nodes
- ❖ Name Node
- ❖ Secondary Name Node
- ❖ Job Tracker
- ❖ Client Nodes
- ❖ Slaves
- ❖ Hadoop configuration
- ❖ Setting up a Hadoop cluster

**HDFS:**

- ❖ Introduction to HDFS
- ❖ HDFS Features
- ❖ HDFS Architecture
- ❖ Blocks
- ❖ Goals of HDFS
- ❖ The Name node & Data Node
- ❖ Secondary Name node
- ❖ The Job Tracker
- ❖ The Process of a File Read
- ❖ How does a File Write work
- ❖ Data Replication
- ❖ Rack Awareness
- ❖ HDFS Federation
- ❖ Configuring HDFS
- ❖ HDFS Web Interface
- ❖ Fault tolerance
- ❖ Name node failure management
- ❖ Access HDFS from Java

**Yarn**

- ❖ Introduction to Yarn
- ❖ Why Yarn
- ❖ Classic MapReduce v/s Yarn
- ❖ Advantages of Yarn
- ❖ Yarn Architecture
- ❖ Resource Manager
- ❖ Node Manager
- ❖ Application Master
- ❖ Application submission in YARN
- ❖ Node Manager containers
- ❖ Resource Manager components
- ❖ Yarn applications
- ❖ Scheduling in Yarn
- ❖ Fair Scheduler
- ❖ Capacity Scheduler
- ❖ Fault tolerance

**MapReduce:**

- ❖ What is MapReduce
- ❖ Why MapReduce
- ❖ How MapReduce works
- ❖ Difference between Hadoop 1 & Hadoop 2
- ❖ Identity mapper & reducer
- ❖ Data flow in MapReduce
- ❖ Input Splits
- ❖ Relation Between Input Splits and HDFS Blocks
- ❖ Flow of Job Submission in MapReduce
- ❖ Job submission & Monitoring
- ❖ MapReduce algorithms
- ❖ Sorting
- ❖ Searching
- ❖ Indexing
- ❖ TF-IDF

**Hadoop Fundamentals:**

- ❖ What is Hadoop
- ❖ History of Hadoop
- ❖ Hadoop Architecture
- ❖ Hadoop Ecosystem Components
- ❖ How does Hadoop work
- ❖ Why Hadoop & Big Data
- ❖ Hadoop Cluster introduction
- ❖ Cluster Modes
- ❖ Standalone
- ❖ Pseudo-distributed
- ❖ Fully – distributed
- ❖ HDFS Overview
- ❖ Introduction to MapReduce
- ❖ Hadoop in demand

## HDFS Operations:

- ❖ Starting HDFS
- ❖ Listing files in HDFS
- ❖ Writing a file into HDFS
- ❖ Reading data from HDFS
- ❖ Shutting down HDFS

## HDFS Command Reference:

- ❖ Listing contents of directory
- ❖ Displaying and printing disk usage
- ❖ Moving files & directories
- ❖ Copying files and directories
- ❖ Displaying file contents

## Java Overview For Hadoop:

- ❖ Object oriented concepts
- ❖ Variables and Data types
- ❖ Static data type
- ❖ Primitive data types
- ❖ Objects & Classes
- ❖ Java Operators
- ❖ Method and its types
- ❖ Constructors
- ❖ Conditional statements
- ❖ Looping in Java
- ❖ Access Modifiers
- ❖ Inheritance
- ❖ Polymorphism
- ❖ Method overloading & overriding
- ❖ Interfaces

## MapReduce Programming:

- ❖ Hadoop data types
- ❖ The Mapper Class
- ❖ Map method
- ❖ The Reducer Class
- ❖ Shuffle Phase
- ❖ Sort Phase
- ❖ Secondary Sort

- ❖ Reduce Phase
- ❖ The Job class
- ❖ Job class constructor
- ❖ JobContext interface
- ❖ Combiner Class
- ❖ How Combiner works
- ❖ Record Reader
- ❖ Map Phase
- ❖ Combiner Phase
- ❖ Reducer Phase
- ❖ Record Writer
- ❖ Partitioners
- ❖ Input Data
- ❖ Map Tasks
- ❖ Partitioner Task
- ❖ Reduce Task
- ❖ Compilation & Execution

**Hadoop Ecosystems**
**Pig:**

- ❖ What is Apache Pig?
- ❖ Why Apache Pig?
- ❖ Pig features
- ❖ Where should Pig be used
- ❖ Where not to use Pig
- ❖ The Pig Architecture
- ❖ Pig components
- ❖ Pig v/s MapReduce
- ❖ Pig v/s SQL
- ❖ Pig v/s Hive
- ❖ Pig Installation
- ❖ Pig Execution Modes & Mechanisms
- ❖ Grunt Shell Commands
- ❖ Pig Latin – Data Model
- ❖ Pig Latin Statements
- ❖ Pig data types
- ❖ Pig Latin operators
- ❖ CaseSensitivity

- ❖ Grouping & Co Grouping in Pig Latin
- ❖ Sorting & Filtering
- ❖ Joins in Pig latin
- ❖ Built-in Function
- ❖ Writing UDFs
- ❖ Macros in Pig

**HBase:**
- ❖ What is HBase
- ❖ History Of HBase
- ❖ The NoSQL Scenario
- ❖ HBase & HDFS
- ❖ Physical Storage
- ❖ HBase v/s RDBMS
- ❖ Features of HBase
- ❖ HBase Data model
- ❖ Master server
- ❖ Region servers & Regions
- ❖ HBase Shell
- ❖ Create table and column family
- ❖ The HBase Client API

**Spark:**
- ❖ Introduction to Apache Spark
- ❖ Features of Spark
- ❖ Spark built on Hadoop
- ❖ Components of Spark
- ❖ Resilient Distributed Datasets
- ❖ Data Sharing using Spark RDD
- ❖ Iterative Operations on Spark RDD
- ❖ Interactive Operations on Spark RDD
- ❖ Spark shell
- ❖ RDD transformations
- ❖ Actions
- ❖ Programming with RDD
- ❖ Start Shell
- ❖ Create RDD
- ❖ Execute Transformations

- ❖ Caching Transformations
- ❖ Applying Action
- ❖ Checking output
- ❖ GraphX overview

## Impala:

- ❖ Introducing Cloudera Impala
- ❖ Impala Benefits
- ❖ Features of Impala
- ❖ Relational databases vs Impala
- ❖ How Impala works
- ❖ Architecture of Impala
- ❖ Components of the Impala
- ❖ The Impala Daemon
- ❖ The Impala Statestore
- ❖ The Impala Catalog Service
- ❖ Query Processing Interfaces
- ❖ Impala Shell Command Reference
- ❖ Impala Data Types
- ❖ Creating & deleting databases and tables
- ❖ Inserting & overwriting table data
- ❖ Record Fetching and ordering
- ❖ Grouping records
- ❖ Using the Union clause
- ❖ Working of Impala with Hive
- ❖ Impala v/s Hive v/s HBase

## MongoDB Overview:

- ❖ Introduction to MongoDB
- ❖ MongoDB v/s RDBMS
- ❖ Why & Where to use MongoDB
- ❖ Databases & Collections
- ❖ Inserting & querying documents
- ❖ Schema Design
- ❖ CRUD Operations

**Oozie & Hue Overview:**

- ❖ Introduction to Apache Oozie
- ❖ Oozie Workflow
- ❖ Oozie Coordinators
- ❖ Property File
- ❖ Oozie Bundle system
- ❖ CLI and extensions
- ❖ Overview of Hue

**Hive:**

- ❖ What is Hive?
- ❖ Features of Hive
- ❖ The Hive Architecture
- ❖ Components of Hive
- ❖ Installation & configuration
- ❖ Primitive types
- ❖ Complex types
- ❖ Built in functions
- ❖ Hive UDFs
- ❖ Views & Indexes
- ❖ Hive Data Models
- ❖ Hive vs Pig
- ❖ Co-groups
- ❖ Importing data
- ❖ Hive DDL statements
- ❖ Hive Query Language
- ❖ Data types & Operators
- ❖ Type conversions
- ❖ Joins
- ❖ Sorting & controlling data flow
- ❖ local vs mapreduce mode
- ❖ Partitions
- ❖ Buckets

**Sqoop:**
- ❖ Introducing Sqoop
- ❖ Scoop installation
- ❖ Working of Sqoop
- ❖ Understanding connectors
- ❖ Importing data from MySQL to Hadoop HDFS
- ❖ Selective imports
- ❖ Importing data to Hive
- ❖ Importing to Hbase
- ❖ Exporting data to MySQL from Hadoop
- ❖ Controlling import process

**Flume:**
- ❖ What is Flume?
- ❖ Applications of Flume
- ❖ Advantages of Flume
- ❖ Flume architecture
- ❖ Data flow in Flume
- ❖ Flume features
- ❖ Flume Event
- ❖ Flume Agent
- ❖ Sources
- ❖ Channels
- ❖ Sinks
- ❖ Log Data in Flume

**Zookeeper Overview:**
- ❖ Zookeeper Introduction
- ❖ Distributed Application
- ❖ Benefits of Distributed Applications
- ❖ Why use Zookeeper
- ❖ Zookeeper Architecture
- ❖ Hierarchial Namespace
- ❖ Znodes
- ❖ Stat structure of a Znode
- ❖ Electing a leader

**Kafka Basics:**

- ❖ Messaging Systems
- ❖ Point-to-Point
- ❖ Publish – Subscribe
- ❖ What is Kafka
- ❖ Kafka Benefits
- ❖ Kafka Topics & Logs
- ❖ Partitions in Kafka
- ❖ Brokers
- ❖ Producers & Consumers
- ❖ What are Followers
- ❖ Kafka Cluster Architecture
- ❖ Kafka as a Pub-Sub Messaging
- ❖ Kafka as a Queue Messaging
- ❖ Role of Zookeeper
- ❖ Basic Kafka Operations
- ❖ Creating a Kafka Topic
- ❖ Listing out topics
- ❖ Starting Producer
- ❖ Starting Consumer
- ❖ Modifying a Topic
- ❖ Deleting a Topic
- ❖ Integration With Spark

**Scala Basics:**

- ❖ Introduction to Scala
- ❖ Spark & Scala interdependence
- ❖ Objects & Classes
- ❖ Class definition in Scala
- ❖ Creating Objects
- ❖ Scala Traits
- ❖ Basic Data Types
- ❖ Operators in Scala
- ❖ Control structures
- ❖ Fields in Scala
- ❖ Functions in Scala
- ❖ Collections in Scala
- ❖ Mutable collection
- ❖ Immutable collection

Are you happy with our course curriculum? Then why you delay? Take your mobile phone and ring us quickly on +91 86816 84318