Databricks PySpark Course Outline – (40 hours)

# Databricks PySpark Course Overview

* The PySpark Certification Program is specially curated
* to provide you with the skills and technical know-how to become a Big Data and Spark developer.
* Starting from the basics of Big Data and Hadoop, this course will boil down
* to cover the key concepts of the PySpark ecosystem,
* Spark APIs, associated tools.

# Software Prerequisites

* Online Databricks Cloud (With Internet Community Edition)
* <https://docs.databricks.com/en/getting-started/community-edition.html>
* [**https://community.cloud.databricks.com/login.html**](https://community.cloud.databricks.com/login.html)

**Participants Can Create PySpark Local Environment**

* Recommended Operating System Ubuntu20/22 or CentOS (Latest Version)
* Hadoop Single Node Cluster
* Apache Spark
* Anaconda with Python 3.10
* PySpark on Hadoop Cloud Environment

# Day 1: Introduction to Big Data and Hadoop.

* 5 Vs of Big Data
* Benefits / Feature / Applications of Big Data
* Hadoop Architecture
* Hadoop Ecosystem
* Basic Hadoop Commands
* Map Reduce Concept with Demo
* Top Level view of YARN
* Node Manager, Resource Manager, Map Reduce processing paradigm

# Introduction to Databricks and Apache Spark:

* Overview of Databricks and its features
* Introduction to Apache Spark and its ecosystem
* Understanding the advantages of using Databricks for big data processing
* Session on creating & using Databrick Spark Cluster.
* Spark architecture – driver program, cluster manager, executors –
* spark operations – number of executors, executor memory
* Hands On Session Spark Cluster

# Conclusion and Summary

# Day2:Databricks Architecture and Components:

* Databricks workspace and notebooks
* Clusters and cluster management
* Introduction to Databricks Runtime and Spark versions
* Understanding the Databricks File System (DBFS)
* Session on creating & using Spark standalone system
* Setting up spark standalone cluster (Documents will be shared)
* understanding driver-memory, driver-cores, executor-cores,
* executor-memory options

**Spark Fundamentals:**

* RDDs (Resilient Distributed Datasets) and transformations
* Actions and lazy evaluation

# Conclusion and Summary

**Day3: Spark RDD Transformations and Actions**

**Transformations -**

* More focused session on
* filter, groupBy, sortBy, joins – inner, outer, cross, partitionBy,
* union, distinct, coalesce, repartition
* Brief overview on-map, flatmap, mapPartitions,
* mapPartitionsWithIndex, flatmapValues, groupByKey,
* reduceByKey, combineByKey, cogroup,
* sample, intersection, subtract, cartesian,
* zip, keyBy, zipWithInde

**Actions –**

* More focused session on
* count, min, max, sum, mean, variance, stdev, saveAsTextFile, saveAsSequenceFile, saveAsObjectFile, saveAsParequetFile
* Brief overview on
* Reduce, Collect, Keys, Values, Aggregate, First, take, foreach, top, collectAsMap, countByValue, countByKey

**HandOn Session: Basic Word Count Application**

* correlating with spark map reduce functioning
* Sparkf RDD application to problems
* basic word count, log file manipulation and statistics, entity resolution
* Spark Configuration - spark-defaults.conf, command line, application
* Spark-submit job packaging and submission
* Exercise – log processing and spark rdd transformations
* **Conclusion and Summary**

**Day 4: Spark SQL and DataFrames**

# Data Manipulation and Processing:

* Data ingestion from various sources (CSV, JSON, Parquet, etc.)
* Store and load the data using various formats – csv, avro, json, orc, parquet
* Data cleaning, filtering, and transformation
* Joins, aggregations, and window functions (Self Join, Recursive Join )
* Aggregate window functions – avg, count, max, min , sum
* Ranking window functions – cume\_dist, dense\_rank, ntile, percent\_rank, rank, row\_number
* Value window functions – lead, lag, first\_value, last\_value
* Handling missing data and outliers

**Data Visualization with Databricks:**

* Using Databricks to create insightful visualizations using various libraries like
* Matplotlib and Seaborn
* Building interactive dashboards using Databricks visualization capabilities.
* Real Time Case Study
* **Conclusion and Summary**

**Day 5. Spark SQL**

* SQL basics for those who are new to SQL.
* Spark SQL data frames and table creation
* Spark SQL querying data using Spark Session available/ created as spark
* Data manipulation and querying NSE Data sets.
* Some example operations and queries
* Creating udfs to transformation
* Joining the tables
* Correlate price, cash market volumes and future volumes
* **Conclusion and Summary**

**Day 6: Spark Streaming and real-time data processing**

* Introduction to Streaming
* Architecture
* Benefits
* How does it work?
* Handling streaming data with Databricks,   
  Introduction to Structured Streaming
* Architecture
* How does it work?
* using DStreams and structured streaming.
* Input Sources
* Sinks
* Structured Streaming Operations
* Windowing on the Streams
* Spark Streaming Versus Structured Streaming
* **Conclusion and Summary**

**Day 7: Memory Optimization /Performance Optimization**

* Performance tuning and optimization techniques
* Caching and persistence
* Broadcast variables and accumulators
* Working with large datasets and out-of-memory data processing
* Spark SQL performance understanding logical plan
* physical plan interpretation and finding improvement opportunities,
* tuning parameters to be aware of
* data frame partitioning basis backing rdd, shuffle partitions,
* max partition bytes,
* file open cost bytes, auto broadcast join threshold
* Understand spark map reduce processing, local block manager service,
* block transport service, role of local disk,
* exchange = shuffle and ways and means to manage it
* Very focused session on
* Spark logs web UI exploration and understanding – rdd creation, transformations, partitioning, task creation, stages and job chain
* Analytical queries using datawarehousing and window functions
* DataWarehousing – cube, rollup, grouping sets
* **Conclusion and Summary**

**Day 8 Introduction to Delta Lake**

* What is Delta Lake?
* Delta Lake operations on Databricks
* Create a table.
* Upsert to a table.
* Read from a table.
* Display table history.
* Query an earlier version of a table.
* Optimize a table
* Add a Z-order index.
* Liquid Clustering
* Vacuum unreferenced files.
* Hands On Session
* **Conclusion and Summary**

**Day 9 Advance Delta Lake Queries**

* Retrieve Delta table history
* History schema
* Operation metrics keys
* Retrieve Delta table details
* Detail schema
* Generate a manifest file
* Convert a Parquet table to a Delta table
* Convert an Iceberg table to a Delta table
* Convert a Delta table to a Parquet table
* Restore a Delta table to an earlier state
* Restore metrics
* Shallow clone a Delta table
* Clone metrics
* Clone use cases
* Clone Parquet or Iceberg table to Delta
* Remove files no longer referenced by a Delta table
* **Conclusion and Summary**

**Day10: Constraints & Certifications Guideline**

* Not Null
* Check
* How does Delta Lake manage feature compatibility?
* Delta column mapping
* What are deletion vectors?
* Enable deletion vectors
* Apply changes to Parquet data files
* Apply changes with REORG Table
* Storage configuration
* Concurrency control
* Optimistic concurrency control
* Write conflicts
* Avoid conflicts using partitioning and disjoint command conditions
* Conflict exceptions
* Certifications Guideline:
* Databricks Certified Associate Developer for Apache Spark
* **Conclusion and Summary**