**Data Engineering with Databricks**

**Duration: 40 Hours**

**Prerequisites:**

* Familiarity with Python programming.
* Basic understanding of SQL.
* General knowledge of data processing and analytics.

**Scope of the Training:**

* Provide participants with foundational skills for building data engineering solutions.
* Enable participants to work effectively with Databricks and Spark for ETL workflows.
* Build confidence in handling real-world data engineering challenges.

**Course Content**

**Module 1: Foundations of Data Engineering and Databricks**

**Estimated Time**: 4 hours

**Concepts:**

* Role of a data engineer in the modern data stack
* Introduction to the Lakehouse architecture
* Overview of the Databricks platform: workspace, clusters, runtimes
* Types of clusters (all-purpose, job clusters), DBUs and pricing
* Workspace navigation and tools (notebooks, Repos, jobs, DBUtils)

**Hands-on:**

* Create and configure a cluster
* Navigate notebooks and execute PySpark cells
* Use widgets, %run, and secrets
* Create your first notebook to explore data

**Module 2: ETL Development with PySpark and Delta Lake**

**Estimated Time**: 6 hours

**Concepts:**

* Spark architecture: DAG, executors, partitions
* PySpark DataFrame API basics (transformations and actions)
* File formats: CSV, JSON, Parquet, Delta
* Delta Lake architecture and its benefits

**Hands-on:**

* Load raw datasets using DataFrames
* Clean, filter, and join data
* Write results to Delta and Parquet formats
* Convert Parquet to Delta format
* Use partitioning in writes

**Module 3: Data Quality and Schema Management**

**Estimated Time**: 4 hours

**Concepts:**

* Schema inference vs. enforcement
* Schema evolution in Delta tables
* Time travel and version control

**Hands-on:**

* Create Delta tables with schemas
* Use Merge, Update, and Delete operations
* Simulate schema evolution (add/remove fields)
* Query Delta history and use time travel

**Module 4: Batch and Stream Processing**

**Estimated Time**: 6 hours

**Concepts:**

* Batch vs. streaming: architecture and when to use each
* Structured Streaming: micro-batches, triggers
* Bronze, Silver, Gold architecture using Delta Lake

**Hands-on:**

* Ingest streaming data using file source
* Apply transformations and windowed aggregations
* Write to Delta Lake

**Module 5: Workflow Orchestration**

**Estimated Time**: 4 hours

**Concepts:**

* Introduction to Databricks Workflows
* Task dependencies and parameterization
* Scheduling pipelines (daily, hourly, event-driven)

**Hands-on:**

* Create a multi-step Databricks Workflow
* Add notebook tasks and set dependencies
* Pass parameters and handle task retries
* Schedule jobs and monitor execution

**Module 6: Security, Access Control, and Compliance**

**Estimated Time**: 4 hours

**Concepts:**

* Unity Catalog overview (catalogs, schemas, tables)
* Access control models (user, group, table, column)
* Logging, auditing, and data governance

**Hands-on:**

* Set up Unity Catalog and secure access
* Apply table-level and column-level permissions

**Module 7: Monitoring, Debugging, and Cost Optimization**

**Estimated Time**: 4 hours

**Concepts:**

* Spark UI for performance monitoring
* Common issues: data skew, OOM, task failures
* Autoscaling, cluster pools, and job clusters
* Cost analysis with DBU tracking and best practices

**Hands-on:**

* Debug slow jobs using the Spark UI
* Identify expensive stages or wide transformations
* Apply caching, partitioning, broadcast joins
* Use cluster pools and autoscaling
* Monitor cost via usage logs

**Module 8: Version Control, CI/CD, and Production Pipelines**

**Estimated Time**: 4 hours

**Concepts:**

* Introduction to Databricks Repos
* Git-based development: branch strategy, pull requests
* Dev → staging → production lifecycle

**Hands-on:**

* Connect notebooks to a Git repo
* Use Git for commits, branches, and conflict resolution
* Deploy notebooks via CLI

**Module 9: Capstone Project – End-to-End Pipeline**

**Estimated Time**: 4 hours

**Capstone Objectives:**

* Ingest both batch and streaming data
* Transform data with PySpark
* Use Delta Lake for reliability
* Orchestrate with Databricks Workflows

**Hands-on:**

* Implement an industry-style pipeline (e.g., retail sales, IoT, banking transactions)
* Present or export a reproducible final solution