



Openminds Technologies



Azure Databricks Course

- Introduction to Databricks
- DBFS (Databricks File System)
- Notebook Utilities in Databricks
- Widget Utilities in Databricks
- Secrets Management in Databricks
- Read / Write Data from SQL DB Using JDBC Connector
- Access Notebook from ADF Pipeline
- Delta Lake
- Optimization and Performance
- RDD
- PySpark DataFrame
- PySpark Transformations
- PySpark SQL Functions
- Latest Announced Databricks CDC Features



openmindstechnologies.com

**Plot No:604/A, Nilgiri Block, Aditya Enclave, 6th Floor,
Ameerpet, Besides Metro Station**



+91 95421 73327, 98496 83033

1 Introduction to Databricks

- What is the Use of Databricks?
- Spark Architecture
- Workspace
- Types of Clusters and Runtimes
- Notebooks
- Jobs

2 DBFS(Databricks File System)

- Upload Files into DBFS
- `dbutils.fs`
- `dbutils.data`
- `dbutils.data.summarize`
- `dbutils.fs.cp`
- `dbutils.fs.head`
- `dbutils.fs.mkdirs`

3 Notebooks utilities in Databricks

- `dbutils.notebook.run()`
- `dbutils.notebook.exit()`
- `dbutils.data`
- `dbutils.data.summarize`
- `dbutils.fs.cp`
- `dbutils.fs.head`
- `dbutils.fs.mkdirs`

4 Widget utilities in Databricks

- `dbutils.widgets.combobox`
- `dbutils.dropdown.dropdown`
- `dbutils.dropdown.multiselect`
- `dbutils.dropdown.text`
- `dbutils.dropdown.get`
- `dbutils.dropdown.getArgument`
- `dbutils.widgets.remove`

5 Pass values to notebook parameters from another notebook using `run()` command

6 Create mount point in Azure Databricks using `dbutils.fs.mount()`

- Create mount point using Account Key
- Create mount point using SAS Token
- Connect ADLS Gen2 to Databricks
- Delete or Unmount Mount Points in Azure Databricks
- `mounts()` & `refreshMounts()` commands of File system Utilities in Azure Databricks
- Update Mount Point(`dbutils.fs.updateMount()`) in Azure Databricks

7 Secrets Management in Databricks

- Azure Key Vault-Backed Scopes
- Databricks-Backed Scopes

8 Read/Write Data from SQL Database using JDBC Connector

9 Access Notebook from ADF Pipeline

- Pass Parameter to Notebook from ADF Pipeline
- Send Exception or Error message from Pipeline to Notebook
- Send parameters from Notebook to ADF Pipelines

10 Delta Lake

- Time Travel / Versioning in Delta Lake
- RESTORE
- Get Delta Lake History
- VACUUM Command in Delta Lake
- MERGE Command in Delta Lake
- Schema Evolution in Delta Lake
- Change Data Feed in Delta Lake
- Z-Order Index
- Table Constraints
- Runbooks
 - Run SQL Commands from Runbook
 - Run Pipeline from Runbook

11

Optimization and Performance

- Optimize Performance with Caching on Azure Databricks
- Dynamic File Pruning
- Low Shuffle Merge
- Adaptive Query Execution
- Predictive I/O
- Cost-Based Optimizer
- Auto Optimize
- Higher-Order Functions
- Isolation Levels
- Column Mapping

12

RDD

- Parallelize
- collect()
- Repartition vs Coalesce
- Broadcast Variables
- Accumulators
- RDD Transformations
- Transformations & Actions

13

Pyspark Dataframe

- Create an Empty DataFrame
- Create an Empty DataFrame with Schema
- Convert an Empty RDD to DataFrame

14

Pyspark Transformations

- show()/Display()
- StructType() and StructField()
- Column class
- Select()
- WithColumn()
- WithColumnRenamed()
- Where() and Filter()
- Drop() and DropDuplicates()
- orderBy() and sort()
- groupBy()
- Join()
- union() and unionAll()
- unionByName()
- map()
- flatmap()
- fillna() and fill()
- pivot()
- partitionBy()
- mapType()
- foreach()
- User Defined Functions

15

Pyspark SQL Functions

- Aggregate Functions
- Window Functions
- Date and Timestamp Functions
- JSON Functions

16

Pyspark DataSources

- Read and Write CSV Files
- Read and Write Parquet Files
- Read and Write JSON Files
- Read Hive Tables
- Save to Hive Tables
- Read JDBC in Parallel
- Query Database Tables
- Read and Write SQL Server
- Read JDBC Tables

17

Pyspark Built-in functions

- when()
- expr()
- lit()
- split()
- concat_ws()
- substring()
- translate()
- regexp_replace()
- to_timestamp()
- to_date()
- date_format()

18

Pyspark Built-in functions

- | | |
|---------------------|-----------------------|
| • struct() | • map_keys() |
| • countDistinct() | • map_values() |
| • sum(), avg() | • months_between() |
| • row_number() | • explode() |
| • rank() | • array_constraints() |
| • dense_rank() | • datediff() |
| • from_json() | |
| • to_json() | |
| • json_tuple() | |
| • get_json_object() | |
| • schema_of_json() | |
| • array() | |
| • collect_list() | |
| • collect_set() | |
| • create_map() | |



- Auto Loader
 - Ingest data efficiently into Delta tables
- Unity Catalog
 - Govern and secure Delta tables across workspaces and clouds
- Mounting
 - Provide simple, persistent access to external storage locations

- CDC (Change Data Capture) in Databricks
- CDC Approaches in Databricks
- Implementing CDC in Databricks with Delta Lake
- Medallion Architecture in Databricks
 - What Is It?
 - Layers in Medallion Architecture
 - Key Benefits

