Azure Databricks

Submitted By: Aniroop Gupta
DE Batch 1

Azure Databricks Overview:

- Distributed computing platform built on Apache Spark.
- Unified engine for SQL, streaming, machine learning (ML), and graph processing.
- Optimized Spark (5x faster) with tools like Delta Lake and MLFlow.
- Integrated with Azure services (e.g., Power BI, Azure Data Lake, Azure ML).

Apache Spark Core Concepts:

- In-memory processing engine for fast data processing.
- Unified engine: handles SQL, streaming, ML, and graph workloads.
- Open source under Apache License.

Apache Spark Architecture:

1. Spark Core: Manages basic I/O, task scheduling, and fault tolerance.
2. RDD (Resilient Distributed Dataset): Immutable, distributed data structure.
3. DataFrame / Dataset APIs: Schema-based data processing (DataFrame for untyped, Dataset for strongly typed).
4. Spark SQL Engine:
    - Catalyst Optimizer: Query optimization.
    - Tungsten Execution Engine: Efficient memory and CPU usage.

Databricks Features:

- Optimized Spark (5x faster).
- Delta Lake: ACID transactions for data lakes.
- MLFlow: Manage ML lifecycle.

Azure Databricks Integration:

- Integrated with Azure Active Directory, Azure Storage (Blob, Data Lake), Azure SQL, Power BI, and Azure ML.
- Unified Azure Portal for management.

Azure Databricks Architecture:

- Control Plane: Manages Databricks workspace, jobs, and clusters.
- Data Plane: Customer's Azure resources (e.g., storage, VMs).
- Azure Resource Manager: Manages Azure resource deployment.

Databricks Workspace Components:

- Notebooks: Code, visualizations, and narrative text.
- Clusters: VMs for running Databricks jobs.
- Jobs: Automated workflows for code execution.
- Data/Models: Input/output data and ML models.

Databricks Clusters:

- Cluster Types:
    1. All-Purpose Cluster: Persistent, shared, expensive.
    2. Job Cluster: Created for specific jobs, cheaper, ephemeral.
- Cluster Configuration:
    1. Multi-Node: Distributed, scalable.
    2. Single-Node: Single VM, smaller workloads.
- Cluster Pools: Reuse clusters to reduce startup time.

Cost and Administration:

- Cluster Policies: Control cluster usage and cost.
- Cost Control: Auto-scaling and resource optimization.