**What is Databricks?**

1. Databricks is a cloud-based data and artificial intelligence (AI) platform designed to facilitate big data analytics, data engineering, and machine learning.
2. It was founded by the creators of Apache Spark and is widely used by organizations to process and analyze massive datasets efficiently.

**Key Features of Databricks:**

1. **Unified Analytics Platform**: Combines data engineering, data science, and business analytics in a single platform.
2. **Collaborative Workspace**: Allows teams to work together using notebooks that support multiple languages, including Python, R, Scala, and SQL.
3. **Apache Spark Integration**: Built on Apache Spark, enabling fast and distributed data processing.
4. **Machine Learning Capabilities**: Provides tools for developing, training, and deploying machine learning models.
5. **Data Lake Integration**: Works seamlessly with data lakes for efficient data storage and retrieval.
6. **Scalability and Performance**: Offers auto-scaling and optimized performance in the cloud.
7. **Data Governance and Security**: Ensures data security, compliance, and access control.

**Databricks Architecture Overview:**

Databricks architecture is designed as a unified platform that integrates big data processing, machine learning, and data analytics. It operates on a cloud-based infrastructure, offering a scalable and collaborative environment.

**Key Components of Databricks Architecture:**

**1) Control Plane (Managed by Databricks):**

1. **Workspace Management:** Manages user authentication, job scheduling, and cluster configuration.
2. **Notebook Interface:** Interactive workspace for development using multiple languages (Python, SQL, Scala, R).
3. **REST API & Web UI:** Allows access to Databricks services via the web interface or APIs.

**2) Data Plane (Managed by Cloud Provider):**

1. **Clusters:** Distributed computing environments for running workloads.
2. **Data Storage:** Supports cloud-native storage systems like Azure Data Lake, AWS S3, and Google Cloud Storage.
3. **Networking:** Manages secure communication and data transfer.
4. **Data Processing Engines:** Built on Apache Spark, enabling fast and distributed data processing.
5. **Delta Lake:**

A key component that ensures reliable data lakes with features like ACID transactions, schema enforcement, and data versioning.

1. **Security & Governance:**

Data encryption, secure networking, and access control ensure enterprise-grade security.

**What is Databricks File System (DBFS)?**

1. Databricks File System (DBFS) is an abstraction layer over cloud storage, providing a distributed file system interface in Databricks.
2. It allows users to store and access data files in a scalable and fault-tolerant manner while integrating seamlessly with cloud-based storage systems like AWS S3, Azure Data Lake Storage, and Google Cloud Storage.

**DBFS Storage Structure:**

DBFS has a file-like structure similar to traditional file systems with directories and files. It uses specific prefixes to distinguish between storage types:

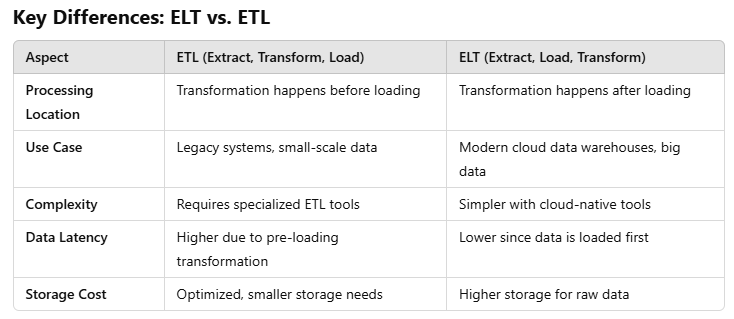
1. **/FileStore**: Temporary file storage, often used for sharing files and visualizations.
2. **/databricks-datasets**: Built-in datasets provided for learning and experimentation.
3. **/mnt**: Mount point for external cloud storage systems like AWS S3 or Azure Blob Storage.

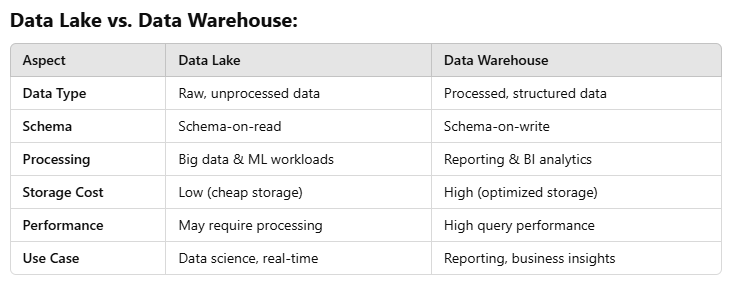
**What is a Data Lake?**

1. A **data lake** is a centralized repository that allows you to store structured, semi-structured, and unstructured data at any scale.
2. It enables organizations to collect, store, and process data from various sources without needing to structure it before storage.
3. Data lakes support big data analytics, machine learning, and real-time data processing.

**Key Characteristics of a Data Lake:**

1. **Storage of All Data:** Stores raw data of all types—structured (databases), semi-structured (JSON, CSV), and unstructured (images, videos, logs).
2. **Scalability:** Highly scalable, enabling storage of petabytes of data across distributed systems.
3. **Schema-on-Read:** Data is stored in its native format and only processed when accessed, enabling flexibility in how data is used.
4. **Real-Time and Batch Processing:** Supports both real-time streaming and batch processing workloads.
5. **Data Integration:** Centralized storage for data from various sources like IoT devices, social media, and enterprise applications.





**Hands-On:**

1. RDDs
2. DFs
3. Spark SQL