UNITY CATALOG

WHAT IS UNITY CATALOG:

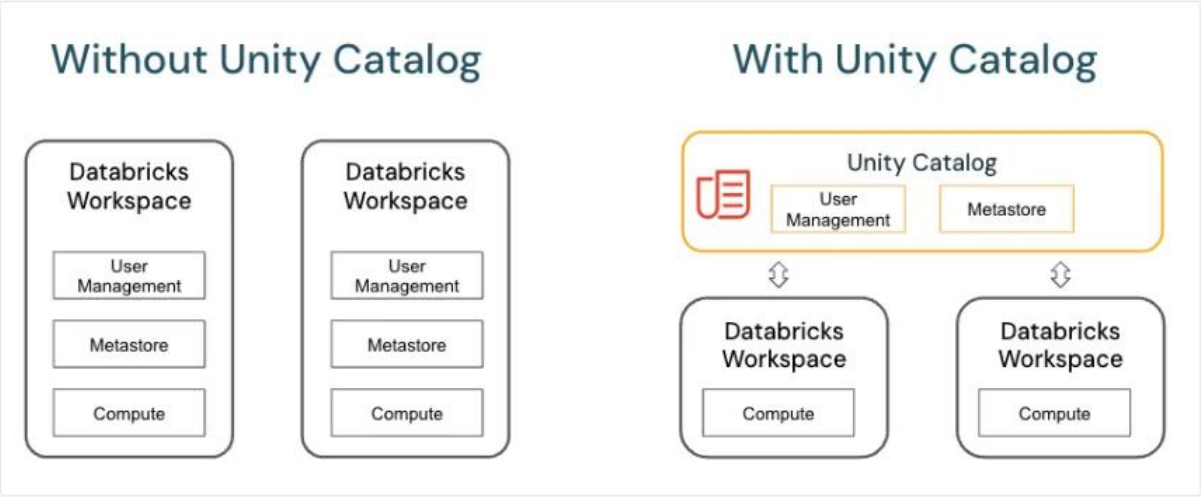
Unity Catalog is Databricks' unified data governance solution that provides a centralized platform for managing, securing, and auditing data across all Databricks workspaces within a cloud account.

It ensures that organizations have a **single source of truth** for:

- **Data Access Control** – Assign and manage permissions in a consistent, fine-grained manner.
- **Data Lineage Tracking** – See where data originates and how it's transformed across the platform.
- **Centralized Auditing** – Track who accessed which datasets, and when.
- **Data Discovery** – Organize and catalog datasets for easier search and use.

KEY BENEFITS

- Works across multiple workspaces in the same region.
- Supports structured (tables, views), unstructured (files, images), and machine learning assets (models).
- Uses **ANSI SQL** standard commands for access control (GRANT, REVOKE).
- Reduces duplication and increases security with a **central metastore**.
- Unity Catalog allows secure and controlled sharing of data across multiple Databricks workspaces in the same region, reducing data duplication and enabling collaborative analytics.
- Works seamlessly with structured (Parquet, Delta), semi-structured (JSON, Avro), and unstructured (images, videos, documents) data, making it a single governance layer for all data types.



OVERVIEW OF THE THREE-LEVEL NAMESPACE

Level 1: Catalog

- **Definition:** The top-level container for organizing schemas and data objects.
- **Purpose**: Typically used to group data by **department**, **project**, or **environment** (e.g., sales, finance, dev, prod).
- **Governance**: Permissions set at the catalog level apply to all schemas and objects inside it unless overridden.

Level 2: Schema

- **Definition:** A logical grouping of related data objects inside a catalog**.**
- **Purpose**: Organize datasets for a specific application, subject area, or team (e.g., customer_data, transactions).
- **Governance**: Permissions can be set at schema level to manage access to all objects inside it.

Level 3: Object

- **Definition:** The actual data or data-related asset.

Types of Objects:

- Tables (managed or external)
- Views (logical representations of data)
- Volumes (unstructured data like images or text files)
- Functions (user-defined or system functions)
- Models (machine learning assets tracked via MLflow)