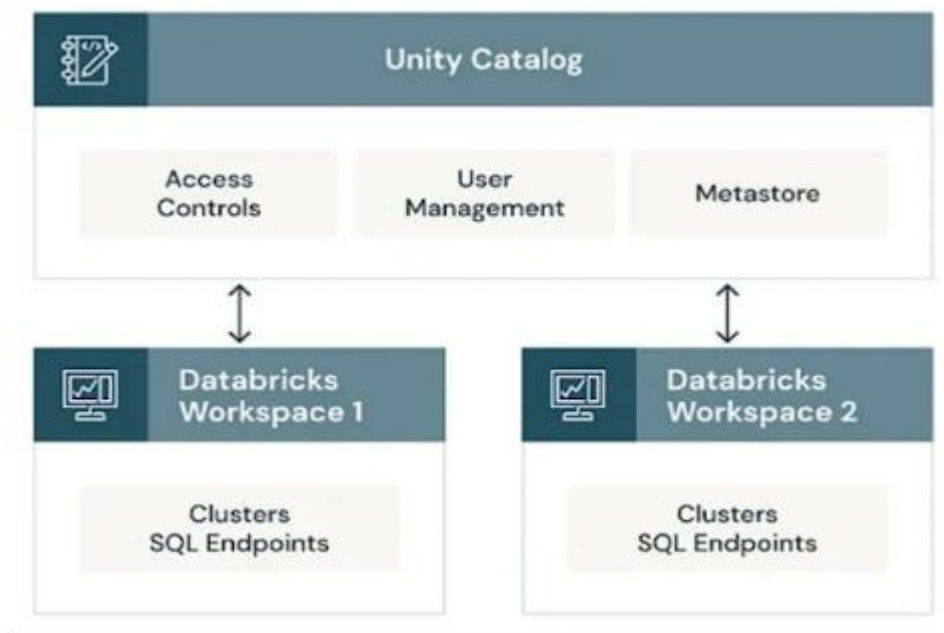


UNITY CATALOG

1. INTRODUCTION TO UNITY CATALOG

Unity Catalog is Databricks' centralized data governance solution that provides:

- Fine-grained access control for data and AI assets
- Unified security model for all data types (tables, files, ML models)
- Auditing and lineage tracking
- Centralized policy enforcement across multiple workspaces



Unity Catalog standardizes data organization using a three-level namespace:

<catalog>. <schema>. <table>.

This provides a clear separation between data governance levels.

2. CREATING A UNITY CATALOG METASTORE

Step 1 – Prerequisites

- You must be a Databricks account admin.
- Your Databricks deployment must be on AWS, Azure, or GCP with Unity Catalog support.
- Your workspace must be on the Premium or Enterprise tier.

Step 2 – Create the Metastore

1. Log in to Databricks Account Console (<https://accounts.cloud.databricks.com>).
2. In the left navigation, click Data → Metastores.
3. Click Create Metastore.
4. Fill in:
 - Name – descriptive name (e.g., org_metastore)
 - Region – must match your cloud storage region
 - Storage Root – cloud storage path for managed tables (Example AWS: s3://my-company-unity-catalog/)
5. Click Create.

Step 3 – Assign the Metastore to a Workspace

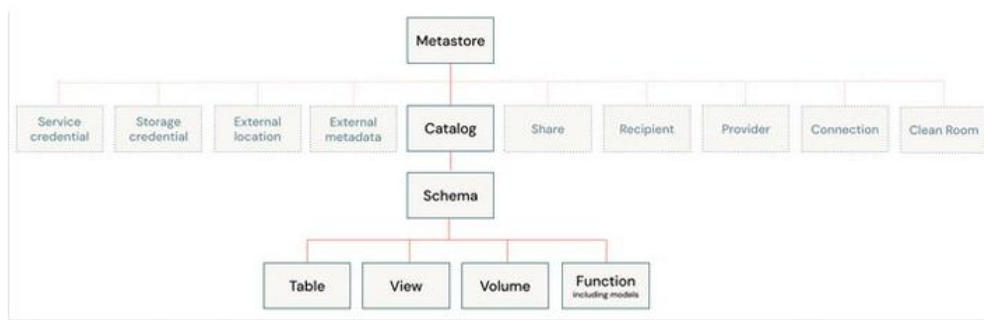
1. In the Metastore list, select the one you just created.
2. Click Assign to Workspace.
3. Select the workspace(s) to enable.
4. Click Assign.

3. ENABLING UNITY CATALOG IN A DATABRICKS WORKSPACE

After assigning:

1. Go to your Databricks workspace.
2. In Admin Settings, ensure Unity Catalog is enabled.
3. Verify by running in a notebook:

SHOW CATALOGS;



4. UNDERSTANDING THE 3-LEVEL NAMESPACE

Unity Catalog organizes data as:

1. Catalog – Top-level container, typically representing an organization-wide grouping of data assets.
2. Schema – Logical grouping of tables/views inside a catalog.
3. Table – The actual data object.

Example:

```
SELECT * FROM sales_data.europe.customers;
```

```
-- catalog = sales_data
```

```
-- schema = europe
```

```
-- table = customers
```

5. CREATING UNITY CATALOG OBJECTS

- **Create a Catalog:**

```
CREATE CATALOG sales_data COMMENT 'Catalog for sales analytics';
```

- **Create a Schema:**

```
CREATE SCHEMA sales_data.europe COMMENT 'Schema for European sales data';
```

- **Create a Managed Table:**

```
CREATE TABLE sales_data.europe.customers (  
    customer_id STRING,  
    name STRING,  
    country STRING  
)  
  
COMMENT 'Customer details for Europe'  
TBLPROPERTIES ('quality'='gold');
```

- **Create an External Table:**

```
CREATE TABLE sales_data.europe.orders  
USING PARQUET  
LOCATION 's3://external-bucket/orders_data/';
```

6. KEY BENEFITS OF UNITY CATALOG

- Fine-grained access control (row/column-level)
- Audit & lineage tracking
- Multi-cloud compatibility
- Consistent namespace across the organization