Setting up Unity Catalog in Databricks

1. Prerequisites

Before beginning the setup process, the following requirements must be met:

- A Databricks account with Admin privileges on the Premium plan is required.
- The individual setting up Unity Catalog must hold Databricks account admin privileges (workspace admin rights alone are not sufficient).
- Azure Active Directory (AAD) permissions are required to register applications and assign roles.

2. Configure Storage

Create ADLS Gen2 Storage Account & Container

- In the Azure portal, create a Storage Account (with hierarchical namespace enabled for ADLS Gen2).
- Inside the storage account, create a container that will be used as the root storage location for the Unity Catalog metastore.

3. Access Connector

- Create an Access Connector for Databricks in the Azure portal (it comes with a managed identity).
- In the ADLS Gen2 storage account, go to IAM and assign the connector identity the role **Storage Blob Data Contributor**.

4. Create the Unity Catalog Metastore

Step 1 – Log in to Databricks Account Console

- Sign in as a Databricks account admin.

Step 2 – Navigate to “Data”

- In the left-hand navigation panel, click Data.

Step 3 – Create the Metastore

- Click Create Metastore.
- Enter a name for the metastore.

- Select the Azure region (must match the workspaces' region).
- Provide the ADLS Gen2 storage path created earlier (add storage path in this format: abfss://<container>@<storageaccount>.dfs.core.windows.net/)
- Enter the Resource ID of the Access Connector.
- Finally, attach the metastore to the workspace.

5. Enable Workspace for Unity Catalog

Step 1 – Access the Metastore

- From the Databricks account console, click on the metastore name.

Step 2 – Assign to Workspaces

- Go to the Workspaces tab.
- Click Assign to workspaces.

Step 3 – Select Workspaces

- Choose the workspace(s) that will use this metastore.

Step 4 – Confirm Assignment

- Click Assign → Enable.
- Verify that the workspaces now appear under the Workspaces tab.

6. Setting Up and Managing Unity Catalog

Prerequisites:

- The workspace must already have a metastore.
- Proper admin role (account, metastore, or workspace).

Steps:

Check if Workspace is Enabled
Run this SQL in a notebook**:**

```
SELECT CURRENT_METASTORE();
```

1. If it shows an ID, it means the workspace is enabled.
2. **Users and Role:**
    - Add users/groups.

- Assign admin roles as required.

3. **Compute Resources**
   - Create SQL warehouses or clusters with Shared Access or Single User Access.

User Privileges
Example SQL:
GRANT CREATE SCHEMA ON CATALOG <my-catalog> TO `data-consumers`;

4. **Catalogs and Schemas**
   - Create a catalog for organizing data and AI assets.

7. Working with Unity Catalog Objects

Prerequisites:

- Unity Catalog enabled workspace.
- A working compute resource.
- Permissions: USE CATALOG, USE SCHEMA, CREATE TABLE.

Steps:

- **New Catalog:** Creates a catalog in Unity Catalog and links it to a storage location in ADLS Gen2.

```
CREATE CATALOG IF NOT EXISTS <catalog_name>
MANAGED LOCATION
'abfss://<container>@<storage>.dfs.core.windows.net/catalogs/<catalog_name>';
```

- **Grant Permissions:** Gives account users rights to use the catalog, create schemas, and create tables.

```
USE CATALOG <catalog_name>;
GRANT CREATE SCHEMA, CREATE TABLE, USE CATALOG
ON CATALOG <catalog_name>
TO `account users`;
```

- **Schema:** Creates a schema inside the catalog to organize tables and views.

```
CREATE SCHEMA IF NOT EXISTS <schema_name>
COMMENT "A new Unity Catalog schema";
```

- **Table:** Creates a Delta table in the schema and inserts sample records.

```
USE <schema_name>;
CREATE TABLE IF NOT EXISTS <table_name> (columnA INT, columnB STRING);
INSERT INTO TABLE <table_name> VALUES (1, "one"), (2, "two");
```

- **Query:** Retrieves data from the created table using the catalog → schema → table structure.

```
SELECT * FROM <catalog_name>.<schema_name>.<table_name>;
```

- **Permissions**: Grants specific users/groups the right to read data (e.g., SELECT on a table).

```
GRANT SELECT ON TABLE <schema_name>.<table_name> TO `account users`;
```