Azure Databricks Coding Challenge

Setting up Unity Catalog in Databricks

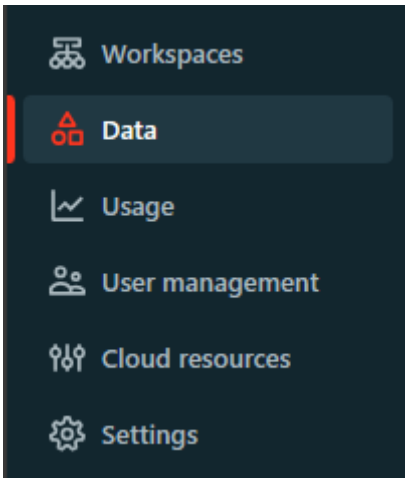
Step 1: Prerequisites

- You must be a Databricks account admin (not just workspace admin).
- You should have Azure Active Directory (AAD) permissions to register apps & assign roles.
- You need a Premium or Enterprise Databricks plan (Unity Catalog isn't on Standard).

Step 2: Create Unity Catalog Metastore

The metastore is like the central brain of Unity Catalog (stores all schemas, tables, permissions).

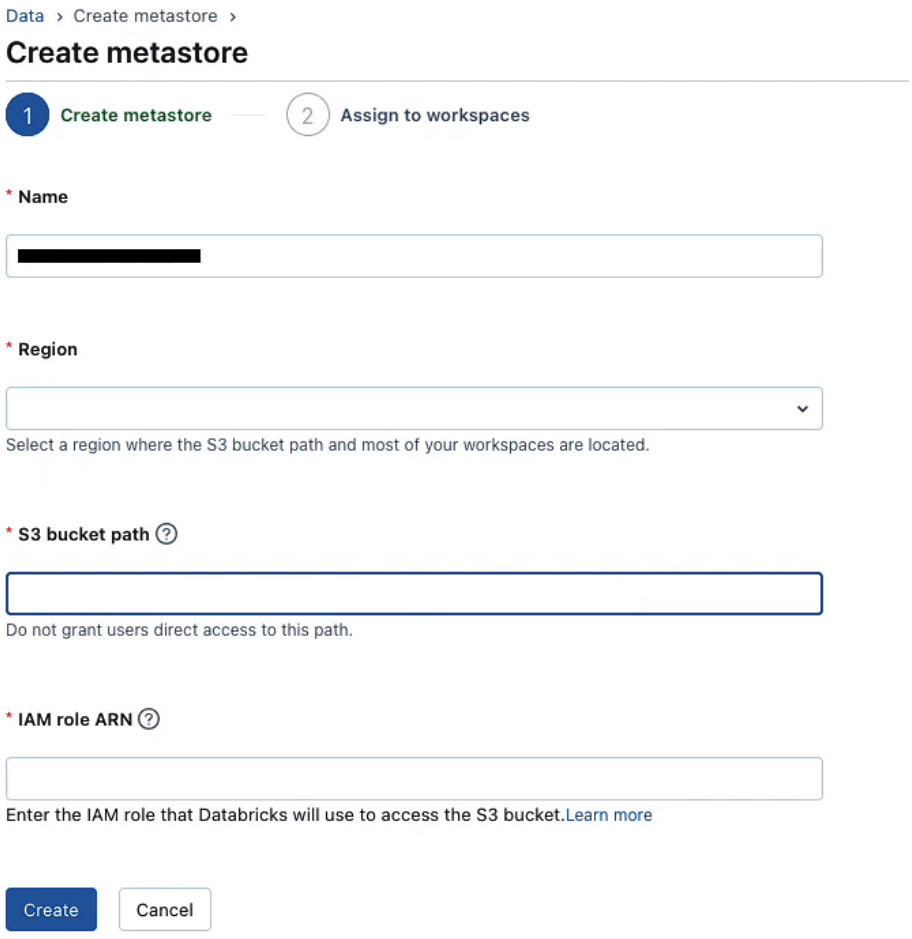
1. Go to the Databricks Account Console → Data → Metastores.
2. Log in to the Databricks account console as an account admin.
3. Click on the "**Data**" option in the left-hand navigation panel.



4. Navigating to "Data" Option - Databricks Unity Catalog
5. Click Create Metastore.

Creating Metastore - Databricks Unity Catalog

Step 3: Provide a name and choose the region for the metastore (same region as your workspaces).



Providing a name and choosing the region for the metastore - Databricks Unity Catalog

Step 4: Fill details:

a. Name: e.g., uc_metastore

b. Region: Must match your Azure region.

c. Storage Root: An ADLS Gen2 storage account container (e.g., abfss://uc-root@yourstorage.dfs.core.windows.net/).

Step 5: Assign a Metastore Admin.

Step 6: Configure Storage (External Location)

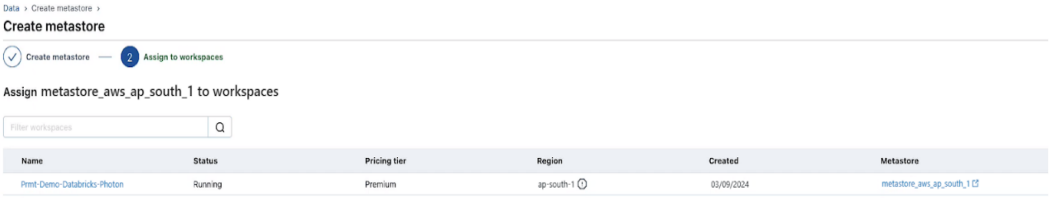
Unity Catalog needs a root storage for managed tables + optional external locations.

1. In Azure Portal, create a Storage Account (ADLS Gen2).
2. Create a Container (e.g., uc-root).
3. Register a Service Principal (SPN) in Azure AD.
    - Copy client ID, secret, tenant ID.
4. Give SPN Storage Blob Data Contributor role on your ADLS.
5. Back in Databricks:
    - Go to External Locations → Create Location.
    - Point to ADLS path.

Step 7: Attach Metastore to Workspace

Now, link the metastore with your Databricks workspace.

1. Go to Account Console → Workspaces.
2. Select your workspace.
3. Click Assign Metastore → choose uc_metastore.



Assign metastore to workspace - Databricks Unity Catalog

Now the workspace is governed by Unity Catalog.

Step 8: Create 3-Level Namespace

Unity Catalog introduces a 3-level namespace:

catalog.schema.table

1. Open a Databricks Notebook (SQL mode).
2. Create a catalog:

```
CREATE CATALOG sales_catalog;

CREATE SCHEMA sales_catalog.retail;

CREATE TABLE sales_catalog.retail.orders (

   order_id INT,

   product STRING,

   amount DOUBLE

);
```

Step 9: Set Permissions (Data Governance)

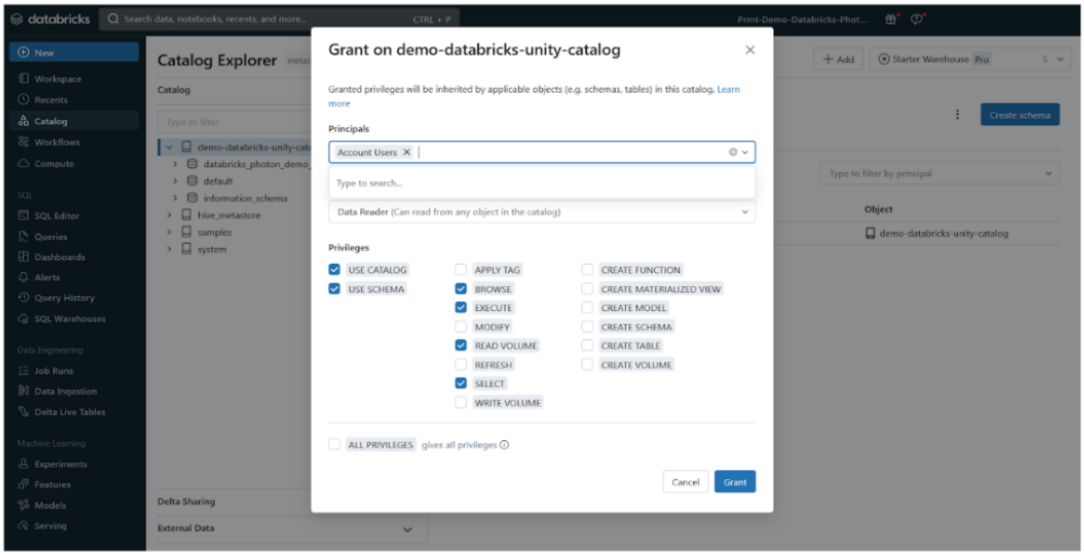
Unity Catalog uses GRANT/REVOKE commands.

```
-- Give analysts access

GRANT SELECT ON TABLE sales_catalog.retail.orders TO `analyst_group`;
```

-- Give data engineers write access

GRANT MODIFY ON TABLE sales_catalog.retail.orders TO `data_engineers`;



Access Control and Security - Databricks Unity Catalog

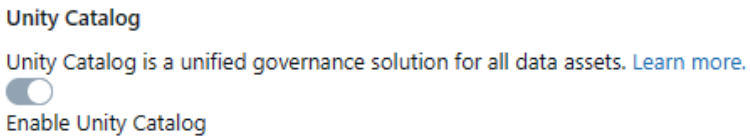
Step 10: Assign and Confirm

Click "Assign", and then confirm by clicking "Enable" on the dialog that appears.

Optional—Enable Databricks Unity Catalog When Creating a Workspace

If you are creating a new workspace, you can enable Databricks Unity Catalog during the workspace creation process:

1) Toggle the "Enable Unity Catalog" option



Toggling the "Enable Unity Catalog" option - Databricks Unity Catalog

2) Select the metastore you want to associate with the new workspace

3) Confirm by clicking "Enable"

4) Complete the process by providing the necessary configuration settings and clicking "Save"

Step 11: Checking workspace is enabled for Databricks Unity Catalog

Run a quick SQL query in the SQL query editor or a notebook connected to a cluster:

```
SELECT CURRENT_METASTORE();
```

If the query result shows a metastore ID, your workspace is attached to a Unity Catalog metastore.