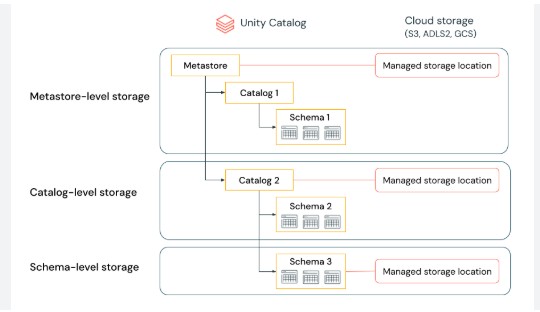
**CODING ASSESSMENT – AZURE DATABRICKS**

**LOKANYA G**

**Unity Catalog in Databricks**

Databricks announced Unity Catalog at the Data and AI Summit in 2021 to address the complexities of data governance by providing fine-grained access control within the Databricks ecosystem. Before Databricks Unity Catalog, data governance in Databricks was typically handled by various third party and open source tools, which, while effective, often lacked seamless integration with the Databricks ecosystem. These tools often lacked the integration and granular security controls specifically tailored for data lakes and were sometimes limited to certain cloud platforms. This limitation highlighted the need for a data governance solution that could provide more fine-grained access control and work across different cloud platforms while integrating seamlessly with the Databricks ecosystem. Thus, Databricks Unity Catalog was born to exactly address these challenges.

Databricks Unity Catalog serves as a centralized governance layer within the Databricks Data Intelligence Platform, streamlining the management and security of various data and AI assets. It supports a wide range of assets including files, tables, machine learning models, notebooks, and dashboards. Unity Catalog uniquely identifies each asset type, simplifying access control and ensuring that only authorized users can interact with specific data elements.

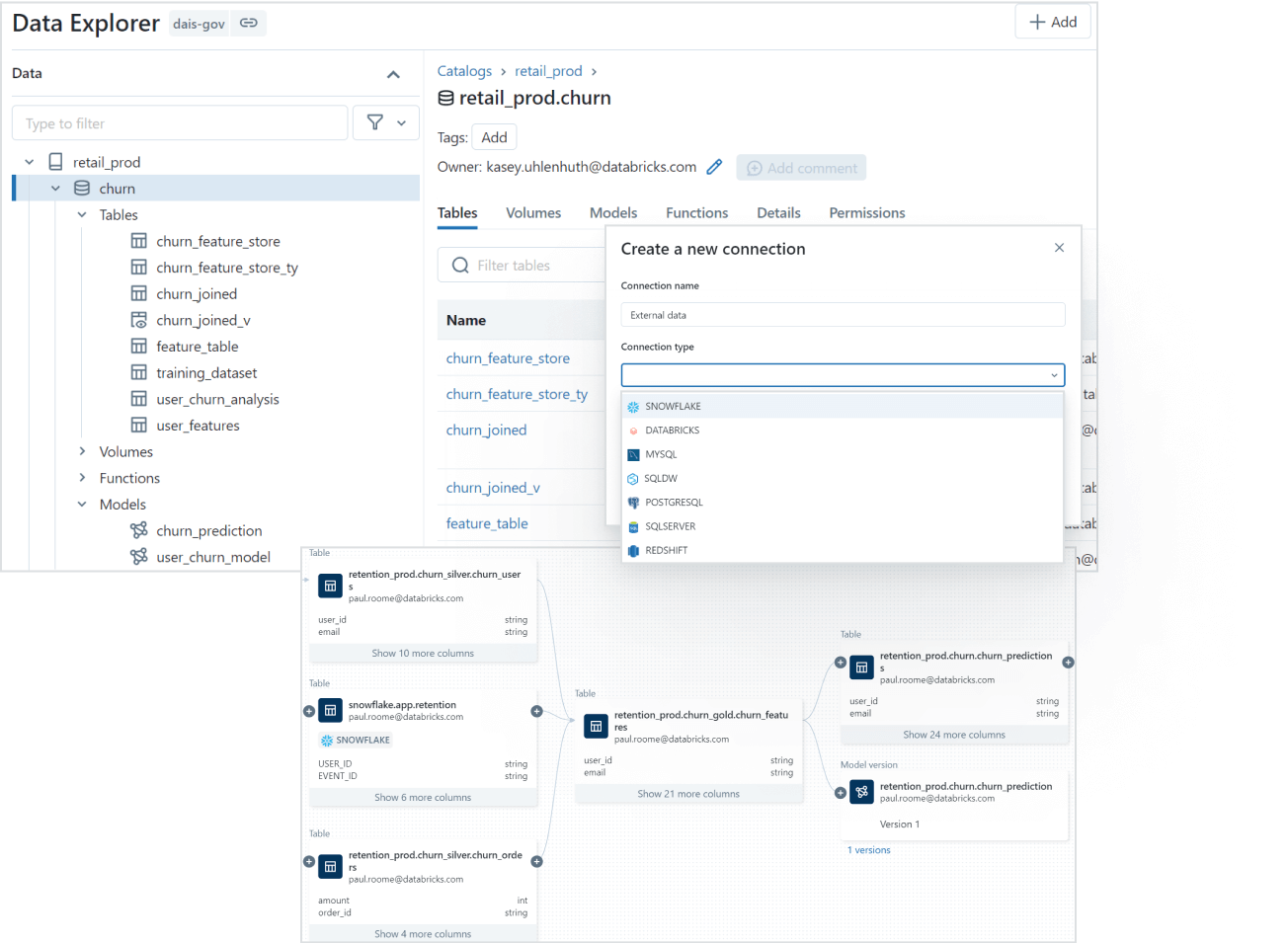


**Databricks Unity Catalog Architecture Breakdown**

Here’s an architecture breakdown of Databricks Unity Catalog:

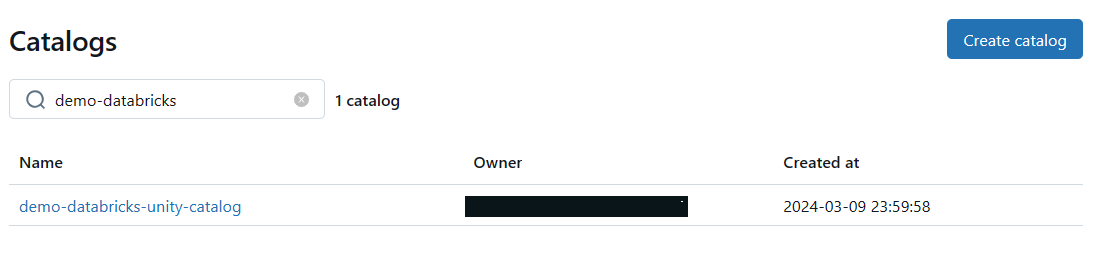
**1) Unified Governance Layer**

Databricks Unity Catalog offers a unified governance layer for both structured and unstructured data, tables, machine learning models, notebooks, dashboards and files  across any cloud or platform.



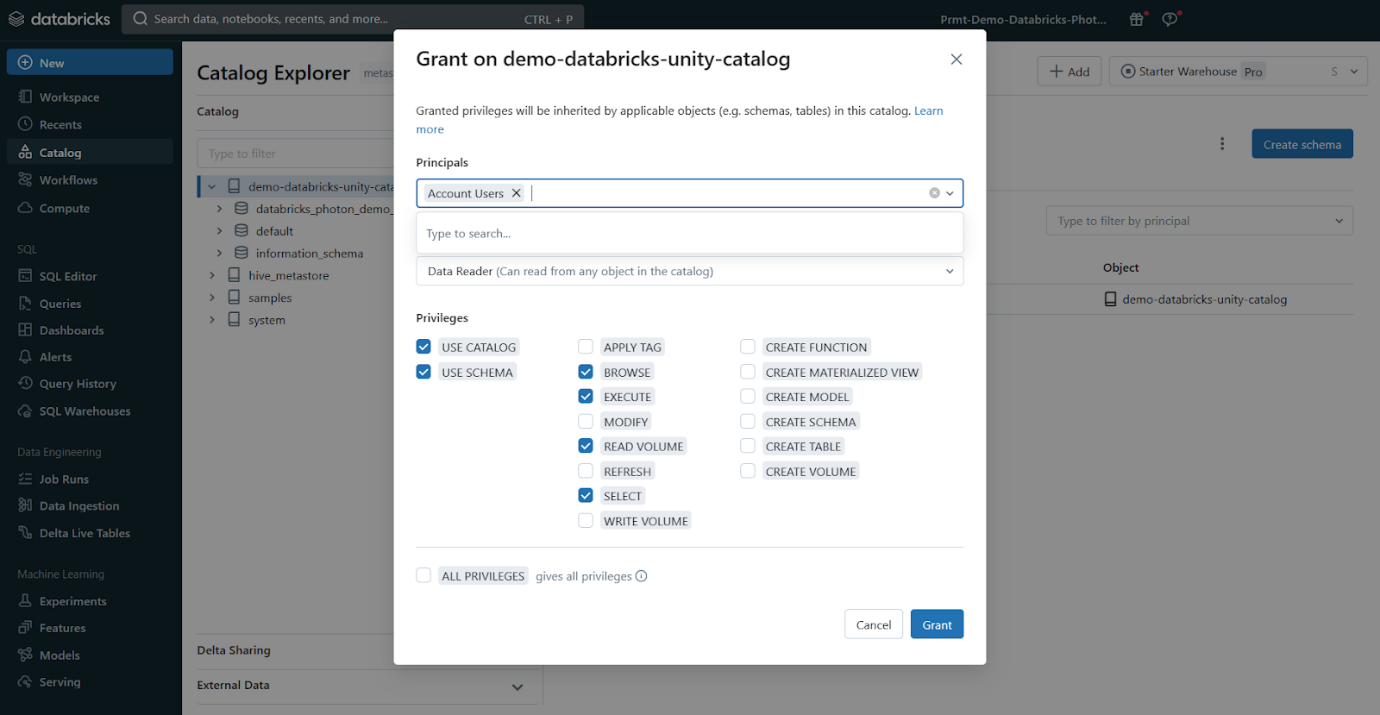
**2) Data Discovery**

You can tag and document your data assets, and then use the search interface to locate the specific data you need based on keywords, tags, or other metadata.



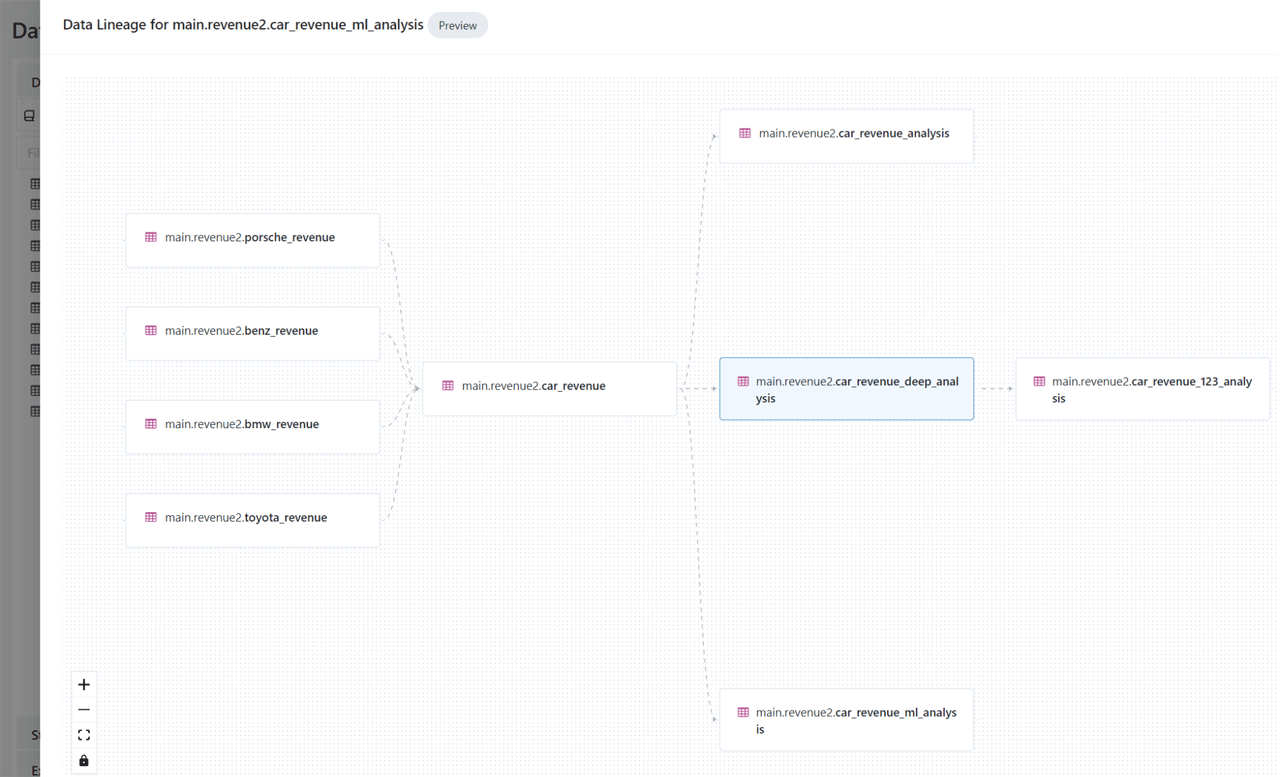
**3) Access Control and Security**

Databricks Unity Catalog simplifies access management by providing a single interface to define access policies on data and AI assets. It supports fine-grained control on rows and columns and manages access through low-code attribute-based access policies that scale seamlessly across different clouds and platforms.



**4) Auditing and Lineage**

Databricks Unity Catalog automatically captures audit logs that record who accessed which data assets .

Auditing and Lineage - Databricks Unity Catalog

**5) Open Data Sharing**

Databricks Unity Catalog integrates with open source Delta Sharing, which allows you to securely share data and AI assets across clouds, regions, and platforms without relying on proprietary formats or complex ETL processes.

**6) Object Model**

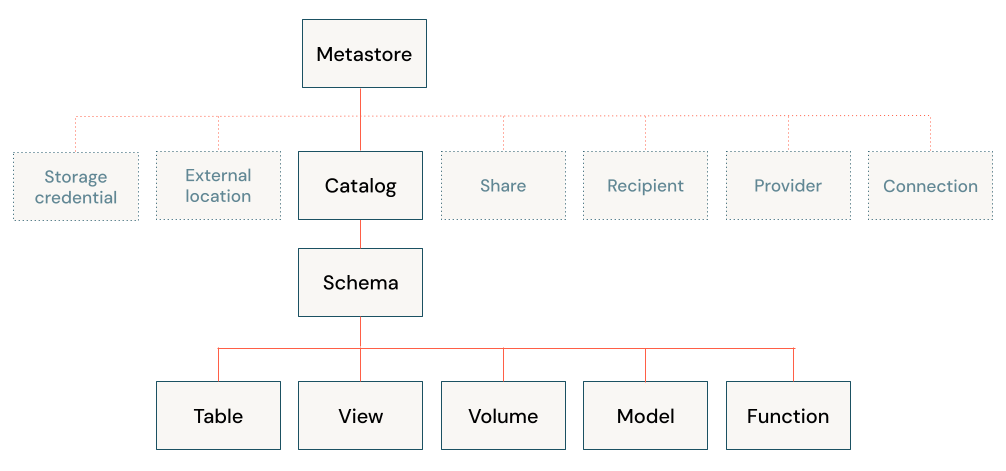
Databricks Unity Catalog organizes your data and AI assets into a hierarchical structure: Metastore ► Catalog ► Schema ► Tables, Views, Volumes, and Models. At the top level, you have the metastore, which contains your schemas. Within each schema, you can have tables, views, or volumes (for unstructured data). To reference any asset, you use a three-part naming convention: <catalog>.<schema>.<asset>. We will dive deeper into the object model in Unity Catalog in next section

**7) Operational Intelligence**

Databricks Unity Catalog provides AI-powered monitoring and observability capabilities that give you deep insights into your data and AI assets. You can set up active alerts, track data lineage at the column level, and gain comprehensive visibility into how your assets are being used and managed.

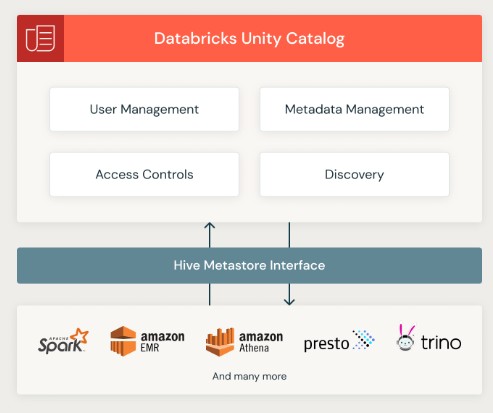
**Databricks Unity Catalog Object Model**

* **Metastore**: The top-level container for metadata, exposing a three-level namespace (catalog.schema.table) to organize data assets.
* **Catalog**: The first layer of the object hierarchy, used to organize data assets logically, often aligned with organizational units or data domains.
* **Schema**: Also known as databases, schemas are the second layer of the object hierarchy, containing tables, views, and volumes.
* **Tables, Views, and Volumes**: The lowest level in the data object hierarchy, where:
  + **Table**: A structured data asset that represents a collection of rows with a defined schema.
  + **View**: A virtual table that is defined by a query.
  + **Volume**: A container for unstructured (non-tabular) data files.
* **Models**: Machine learning models registered in the MLflow Model Registry can also be managed within Databricks Unity Catalog.



**What Is the Difference Between Unity Catalog and Hive Metastore?**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Hive Metastore** | **Unity Catalog** |
| **Scope** | Workspace/Cluster-level | Account-level (multi-workspace, multi-cloud) |
| **Governance** | Basic, often requires external tools | Unified, built-in, fine-grained |
| **Security** | Basic access control | Advanced access control, auditing, lineage |
| **Managed Tables** | Immediate data deletion on drop | Data retention for recovery after drop |
| **Data Assets** | Primarily tables and partitions | All data and AI assets (tables, models, notebooks) |

****

**How to Create Unity Catalog Metastore (AWS)**

To create a Unity Catalog metastore in the AWS cloud environment, follow these prerequisites and steps:

**Step 1—Configure Storage**

Before creating a Databricks Unity Catalog metastore, you may want to create an S3 bucket to store data that is managed at the metastore level.

**Step 2—Create an IAM Role to Access the Storage Location**

Next, create an AWS IAM role to allow Databricks to access the S3 bucket you created in Step 1. This role should have the necessary permissions to read and write data to the bucket.

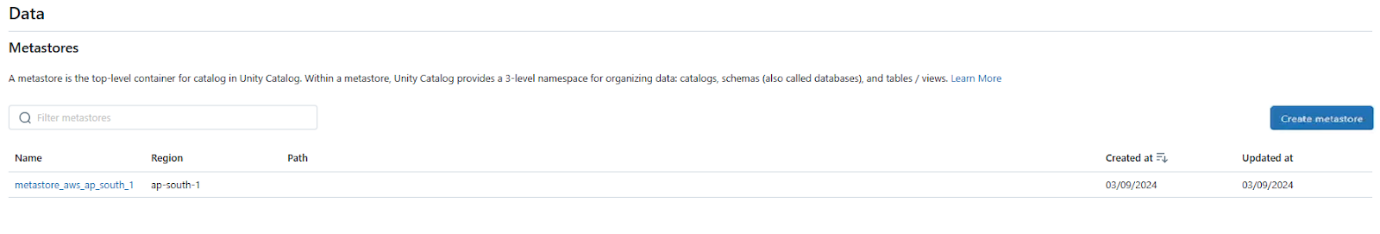
Follow the Creating IAM roles guide if needed.

**Step 3—Create the Metastore and Attach a Workspace**

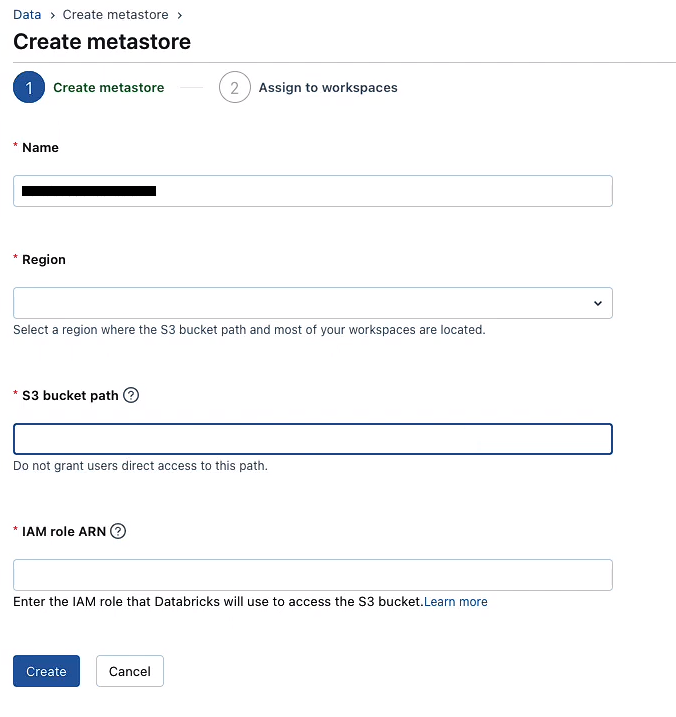
Finally, create the Unity Catalog metastore in the Databricks account console. To do so:

**1)**Log in to the Databricks console and navigate to the Data option.

**2)**Click "**Create Metastore**".

Creating Metastore - Databricks Unity Catalog

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

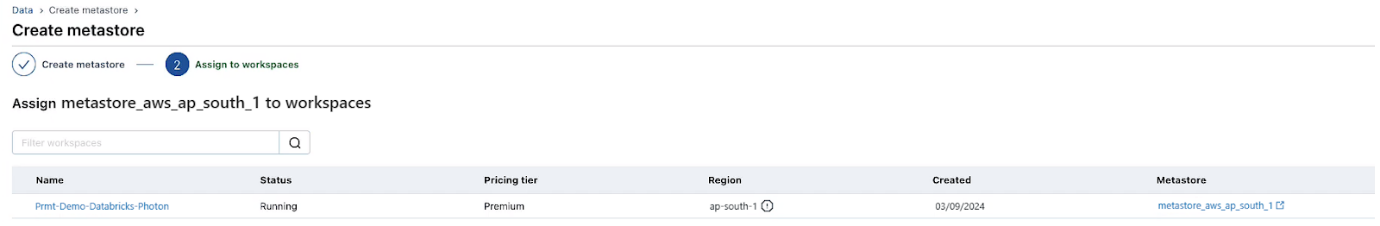




Providing a name and choosing the region for the metastore - Databricks Unity Catalog

**4)**Optionally, attach the storage location you created in step 1.

**5)**Assign the metastore to your workspace.

Assign metastore to workspace - Databricks Unity Catalog

Check out this documentation for a full guide on creating a Databricks Unity Catalog metastore in AWS.

**Step-By-Step Guide to Enable Your Workspace for Databricks Unity Catalog**

Enabling Databricks Unity Catalog on your workspace is a straightforward process that can be done through the Databricks account console or during workspace creation. Follow these step-by-step instructions to enable Databricks Unity Catalog:

**Step 1—Log in to the Databricks Account**

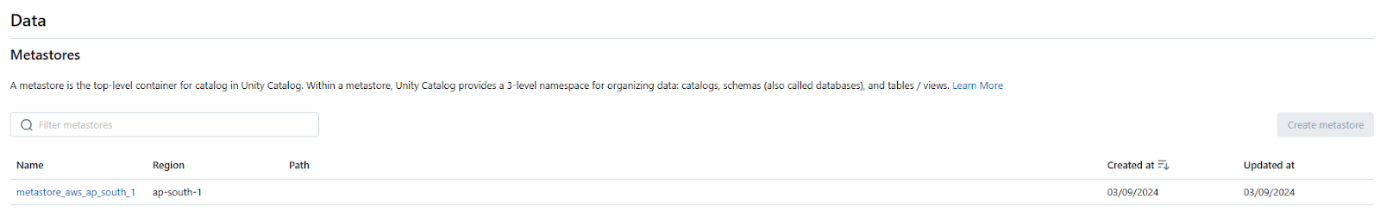
Log in to the Databricks account console as an account admin.

**Step 2—Click on “Data” Option**

Click on the "**Data**" option in the left-hand navigation panel.

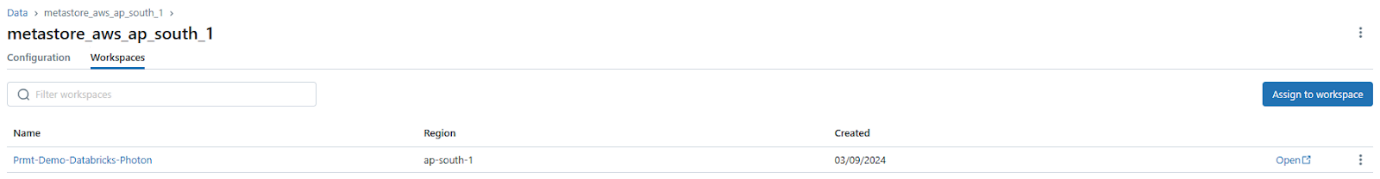
**Step 3—Access the Metastore**

Access the metastore by clicking on the metastore name.



**Step 4—Navigate to the Workspaces Tab**

Within the metastore, head over to the "**Workspaces**" tab.

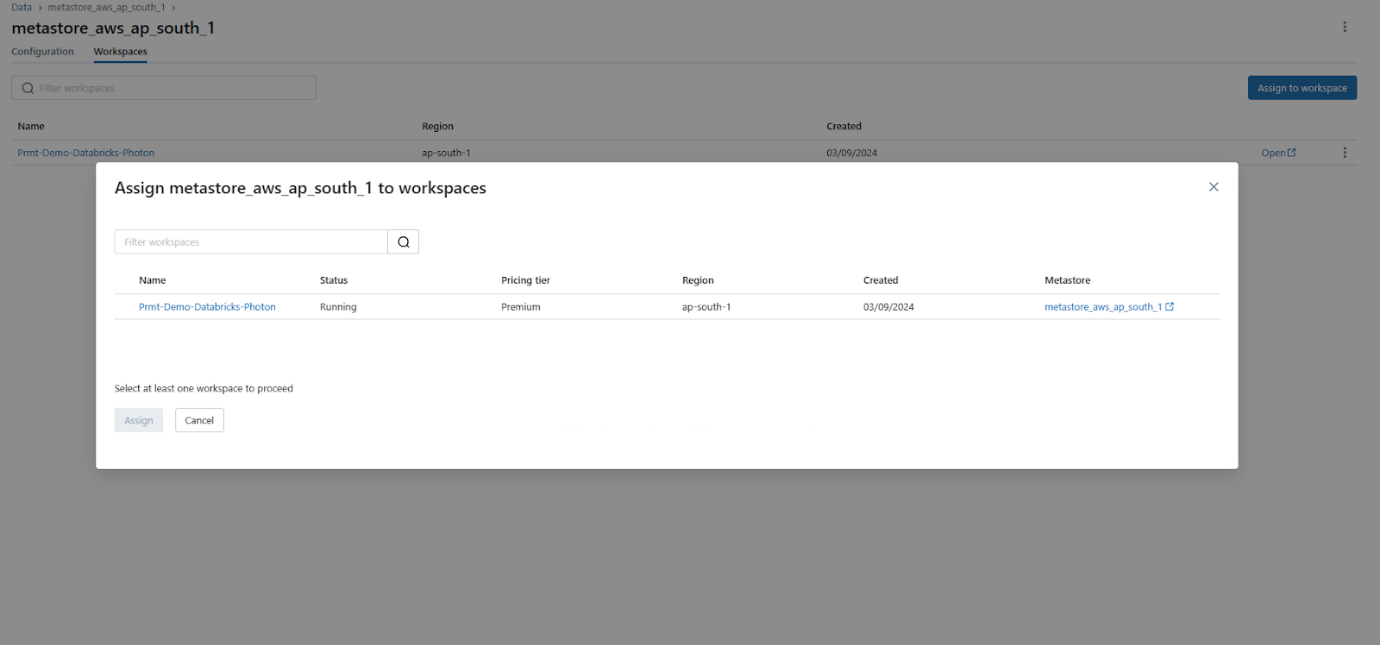
Navigate to the “Workspaces” Tab within metastore - Databricks Unity Catalog

**Step 5—Assign to Workspaces**

Click the "**Assign to workspaces**" button to enable Databricks Unity Catalog on one or more workspaces.

**Step 6—Choose One or More Workspaces to Enable**

In the "**Assign Workspaces**" dialog, select one or more workspaces you want to enable for Databricks Unity Catalog.



**Step 7—Assign and Confirm**

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