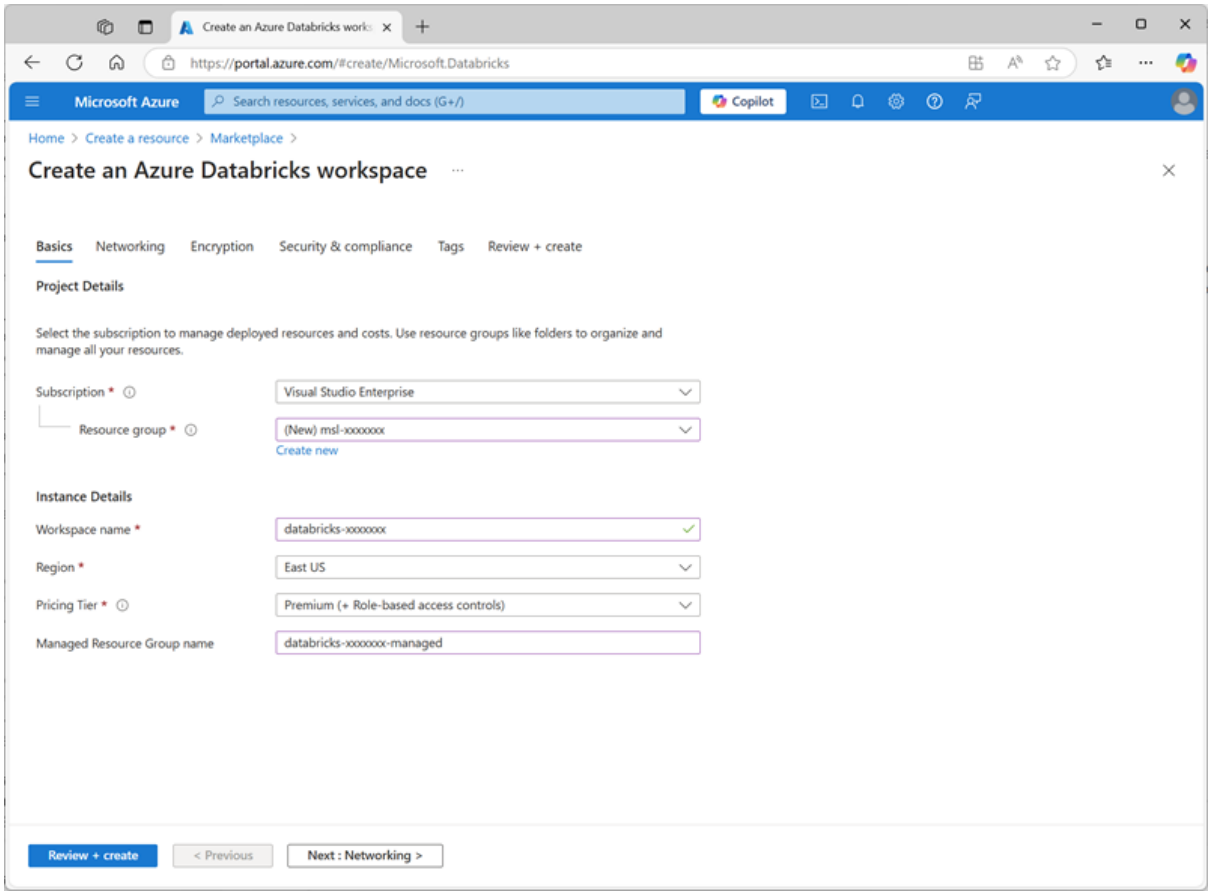
Explore Unity Catalog in Azure Databricks

Unity Catalog offers a centralized governance solution for data and AI, simplifying security by providing a single place to administer and audit data access. In this exercise, you'll configure Unity Catalog for an Azure Databricks workspace and use it to manage data.

Create an Azure Databricks workspace

Tip: If you already have a premium or Trial tier Azure Databricks workspace, you can skip this procedure and use your existing workspace.

1. Sign into the **Azure portal** at https://portal.azure.com.
2. Create an **Azure Databricks** resource with the following settings:
    - **Subscription**: *Select your Azure subscription*
    - **Resource group**: *Create a new resource group named e.g. databricksRG*
    - **Workspace name**: databricks-yourname
    - **Region**: *Central US or EastUS*
    - **Pricing tier**: *Trial*
    - **Managed Resource Group name**: databricks-yourname-managed



3. Select **Review + create** and wait for deployment to complete.

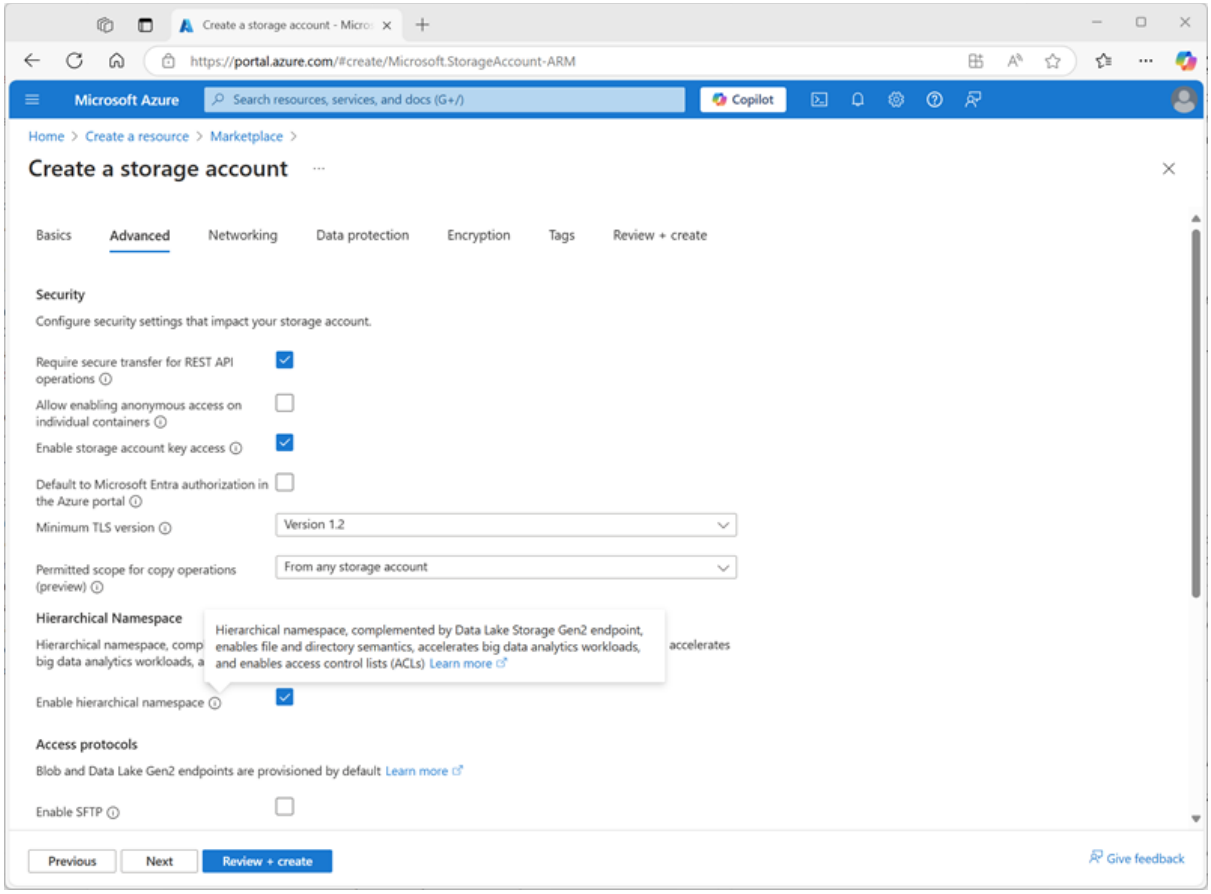
Prepare storage for the catalog

When using Unity Catalog in Azure Databricks, data is stored in an external store; which can be shared across multiple workspaces. In Azure, it's common to use an Azure Storage account with support for a Azure Data Lake Storage Gen2 hierarchical namespace for this purpose.

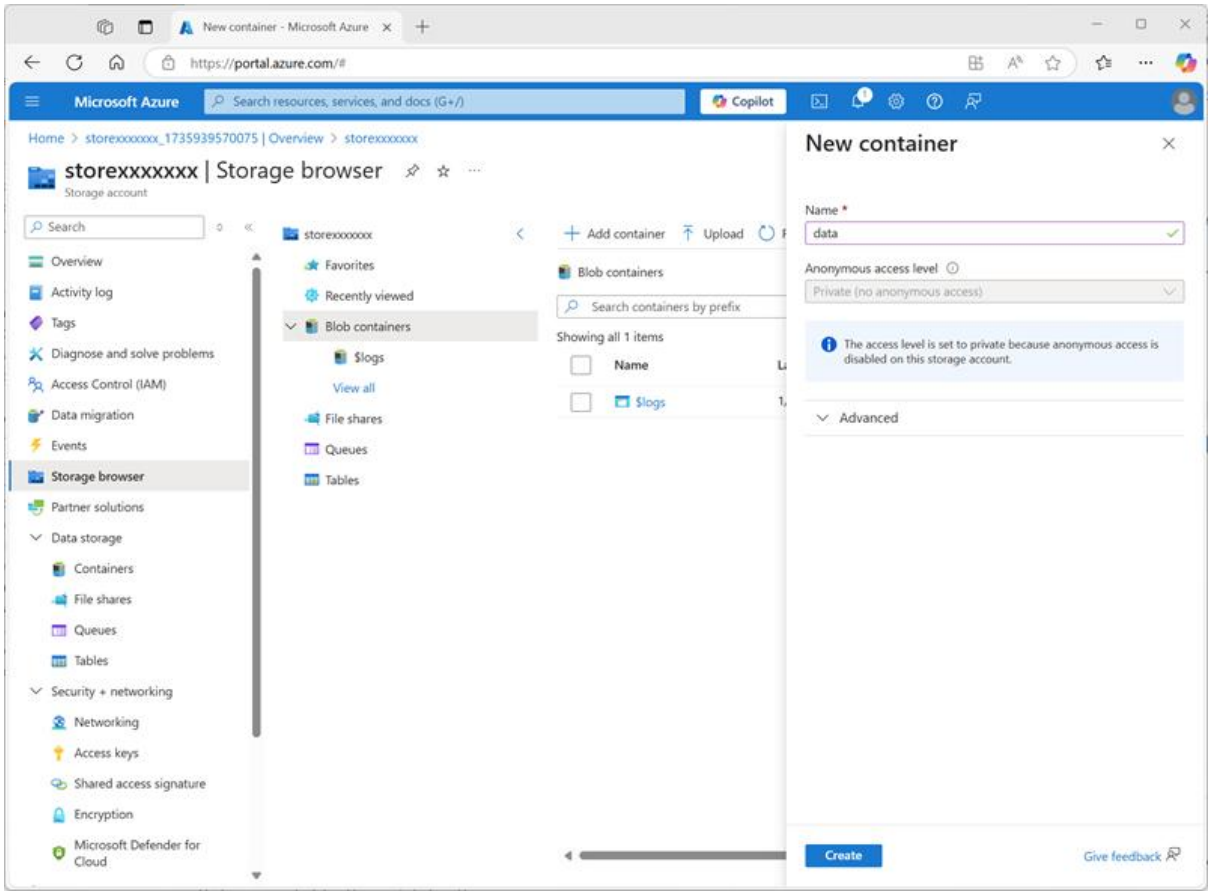
1. In the Azure portal, create a new **Storage account** resource with the following settings:
    - **Basics**:
        - **Subscription**: *Select your Azure subscription*
        - **Resource group**: *Select the existing **databricksRG** resource group where you created the Azure Databricks workspace.*
        - **Storage account name**: storeyourname
        - **Region**: *Select the region where you created the Azure Databricks workspace*

- **Primary service**: Azure Blob Storage or Azure Data Lake Storage Gen2
- **Performance**: Standard
- **Redundancy**: Locally-redundant storage (LRS) *(For a non-production solution like this exercise, this option has lower cost and capacity consumption benefits)*

- **Advanced**:
  - **Enable hierarchical namespace**: *Selected*



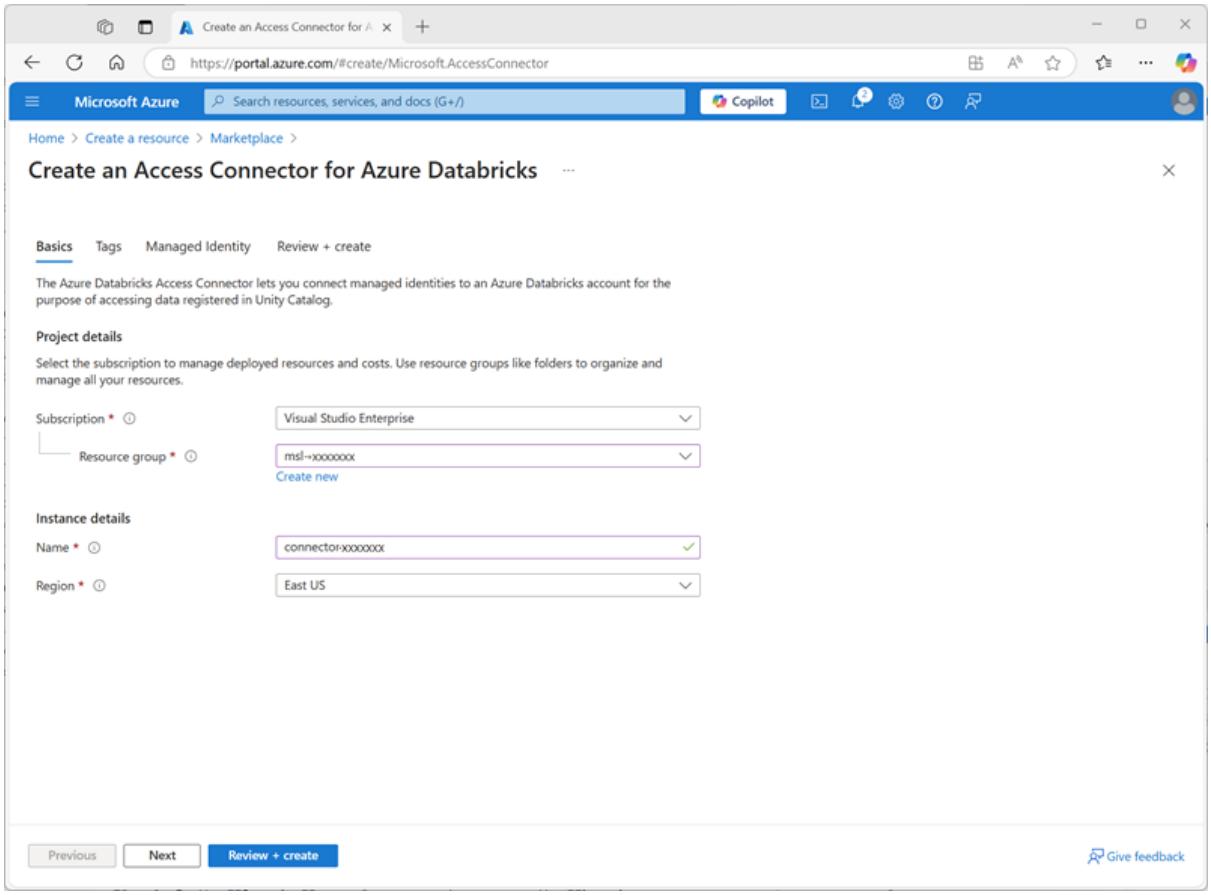
2. Select **Review + create** and wait for deployment to complete.
3. When deployment has completed, go to the deployed *storeyourname* storage account resource and use its **Storage browser** page to add a new blob container named data. This is where the data for your Unity Catalog objects will be stored.



Configure access to catalog storage

To access the blob container you have created for Unity Catalog, your Azure Databricks workspace must use a managed account to connect to the storage account through an *access connector*.

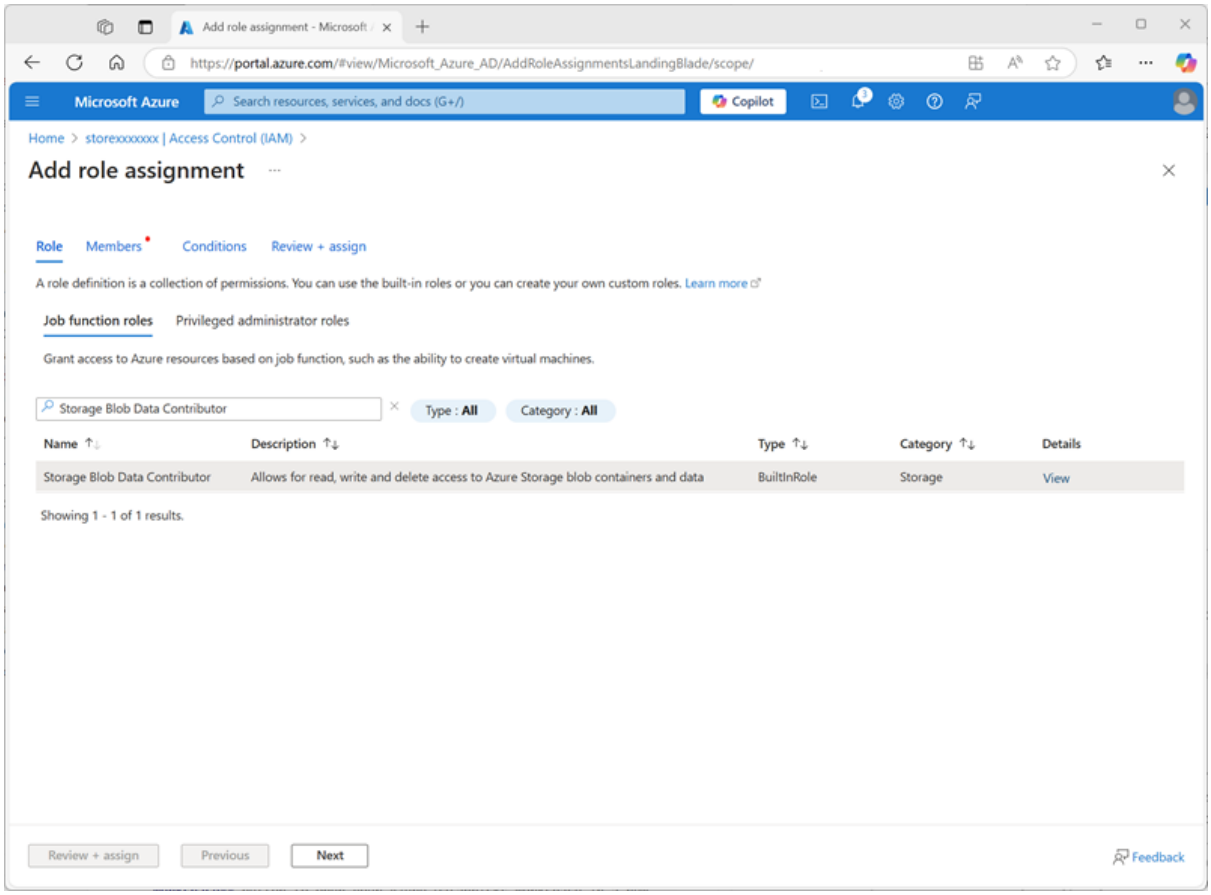
1. In the Azure portal, create a new **Access connector for Azure Databricks** resource with the following settings:
    - **Subscription**: *Select your Azure subscription*
    - **Resource group**: *Select the existing **databricksRG** resource group where you created the Azure Databricks workspace.*
    - **Name**: connector-yourname
    - **Region**: *Select the region where you created the Azure Databricks workspace*



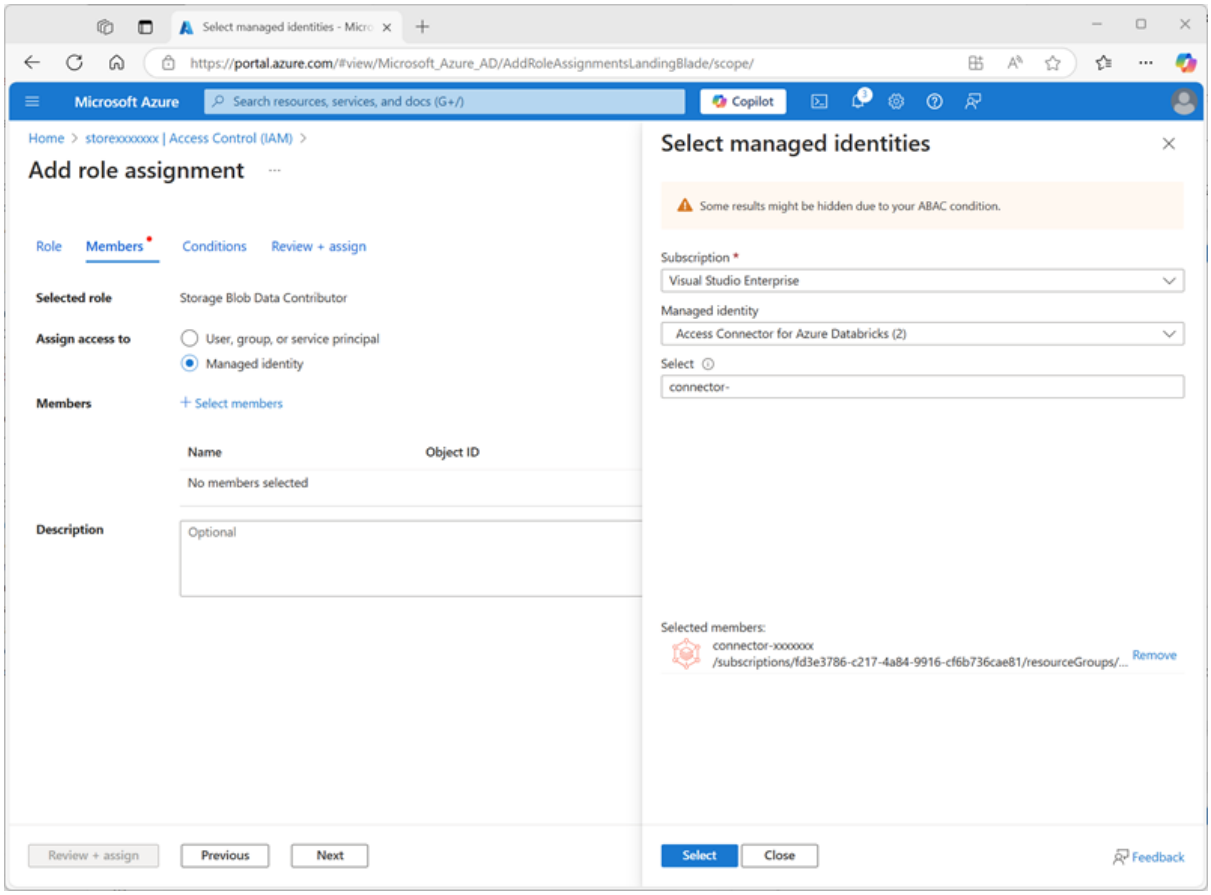
2. Select **Review + create** and wait for deployment to complete. Then go to the deployed resource and on its **Overview** page, note the **Resource ID**, which should be in the format */subscriptions/abc-123.../resourceGroups/databricksRG/providers/Microsoft.Databricks/accessConnectors/connector-yourname* - you'll need this later.

3. In the Azure portal, return to the *storeyourname* storage account resource and on its **Access Control (IAM)** page, add a new role assignment.

4. In the **Job function roles** list, search for and select the Storage blob data contributor role.



5. Select **Next**. Then on the **Members** page, select the option to assign access to a **Managed Identity** and then find and select the connector-yourname access connector for Azure Databricks you created previously (you can ignore any other access connectors that have been created in your subscription)



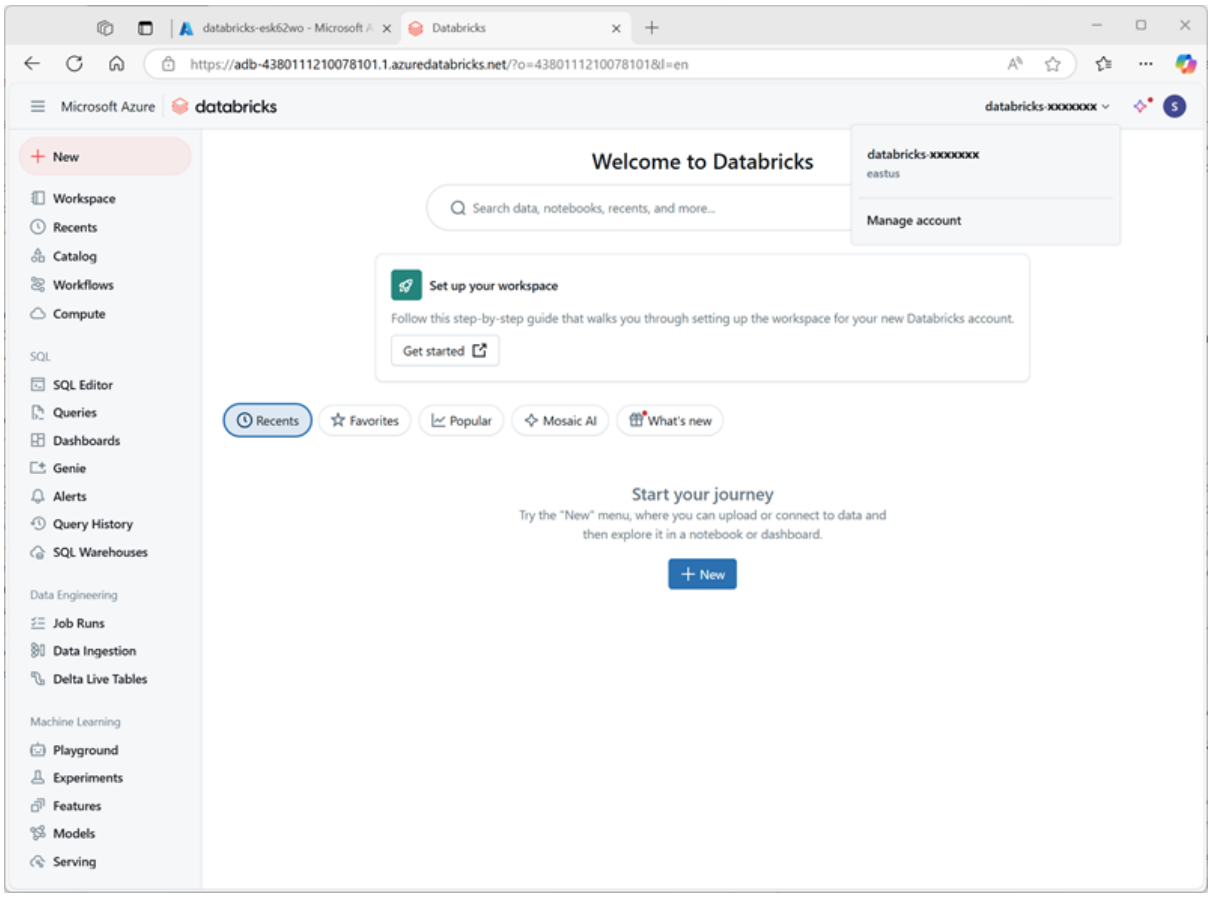
6. Review and assign the role membership to add the managed identity for your *connector-yourname* access connector for Azure Databricks to the Storage blob data contributor role for your *storeyourname* storage account - enabling it to access data in the storage account.

Configure Unity Catalog

Now that you have created a blob storage container for your catalog and provided a way for an Azure Databricks managed identity to access it, you can configure Unity Catalog to use a metastore based on your storage account.

1. In the Azure portal, view the **databricksRG** resource group, which should now contain three resources:
    - The **databricks-*yourname*** Azure Databricks workspace
    - The **storeyourname** storage account
    - The **connector-*yourname*** access connector for Azure Databricks
2. Open the **databricks-yourname** Azure Databricks workspace resource you created and earlier, and on its **Overview** page, use the **Launch Workspace** button to open your Azure Databricks workspace in a new browser tab; signing in if prompted.

3. In the **databricks-*yourname*** menu at the top right, select **Manage account** to open the Azure Databricks account console in another tab.

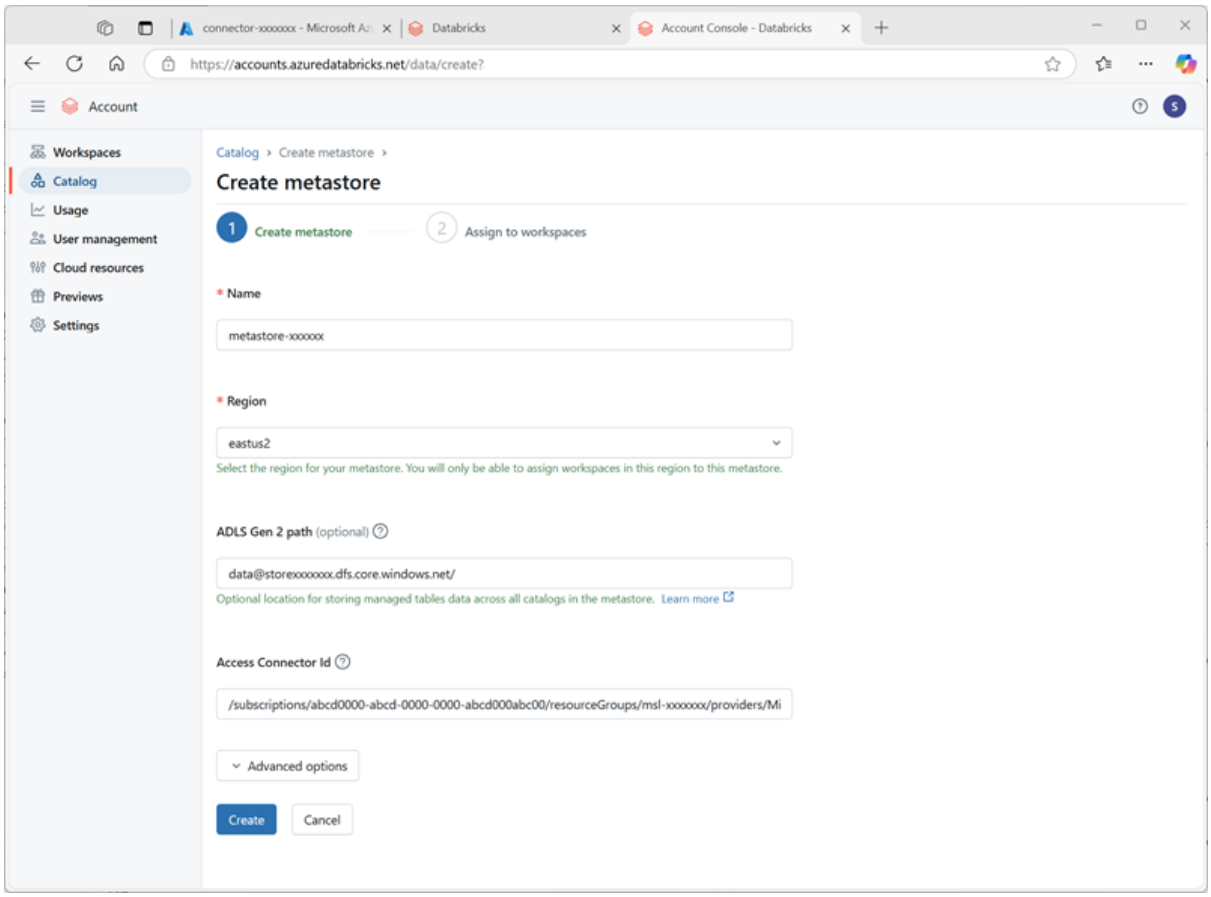


Note: If ***Manage account*** is not listed or doesn't successfully open, you may need to have a global administrator add your account to the ***Account Admin*** role in your Azure Databricks workspace.

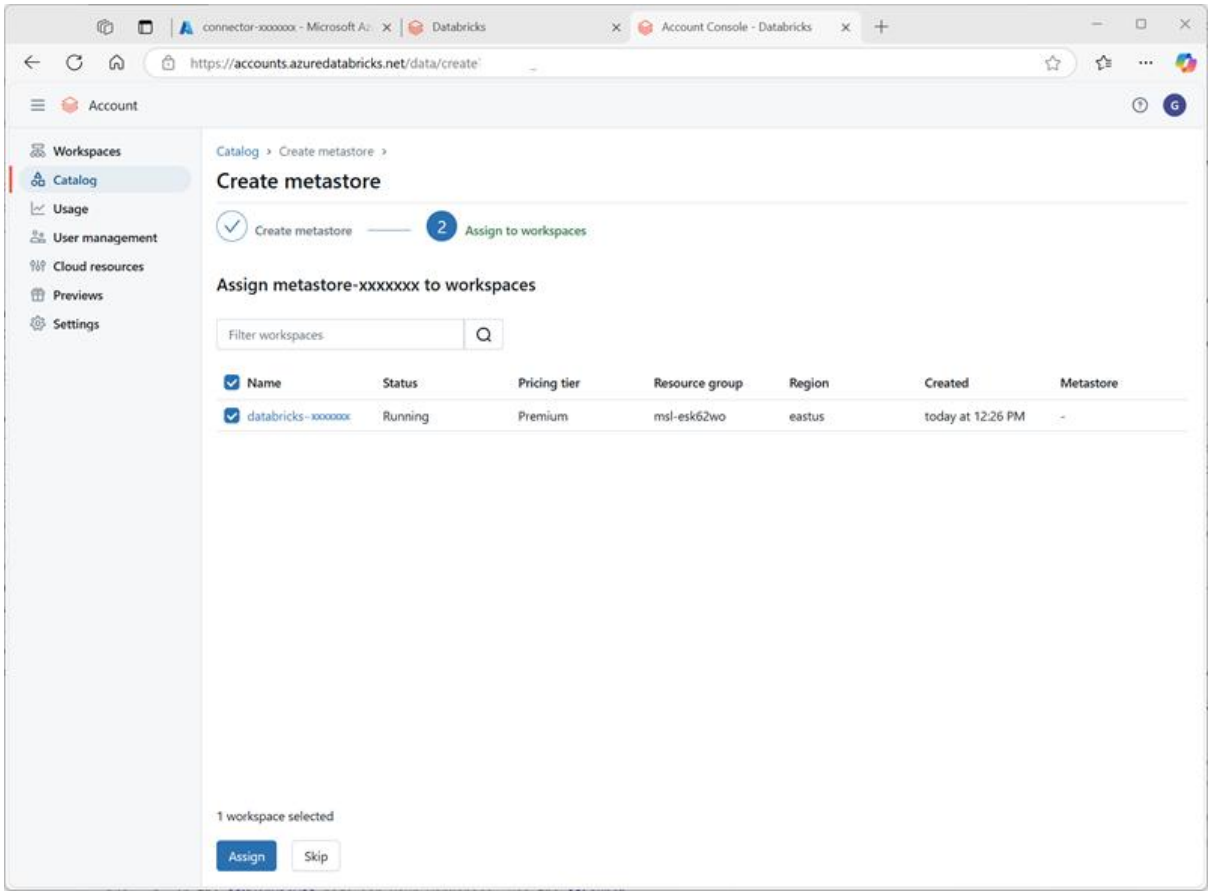
If you're using a personal Azure subscription that you created using a personal Microsoft account (such as an oultook.com account), an "external" Entra ID account may have been automatically created in your Azure directory, and you may need to sign in using that account name.

4. In the Azure Databricks account console, on the **catalog** page, select **Create metastore**.
5. Create a new metastore with the following settings:
    - **Name**: metastore-yourname
    - **Region**: *Select the region where you created your Azure resources*
    - **ADLS Gen 2 path**: data@storeyourname.dfs.core.windows.net/ *(where storeyourname is the your storage account name)*

- **Access Connector Id**: *The resource ID for your access connector (copied from its Overview page in the Azure portal)*



6. After creating the metastore, select the **databricks-*yourname*** workspace and assign the metastore to it.

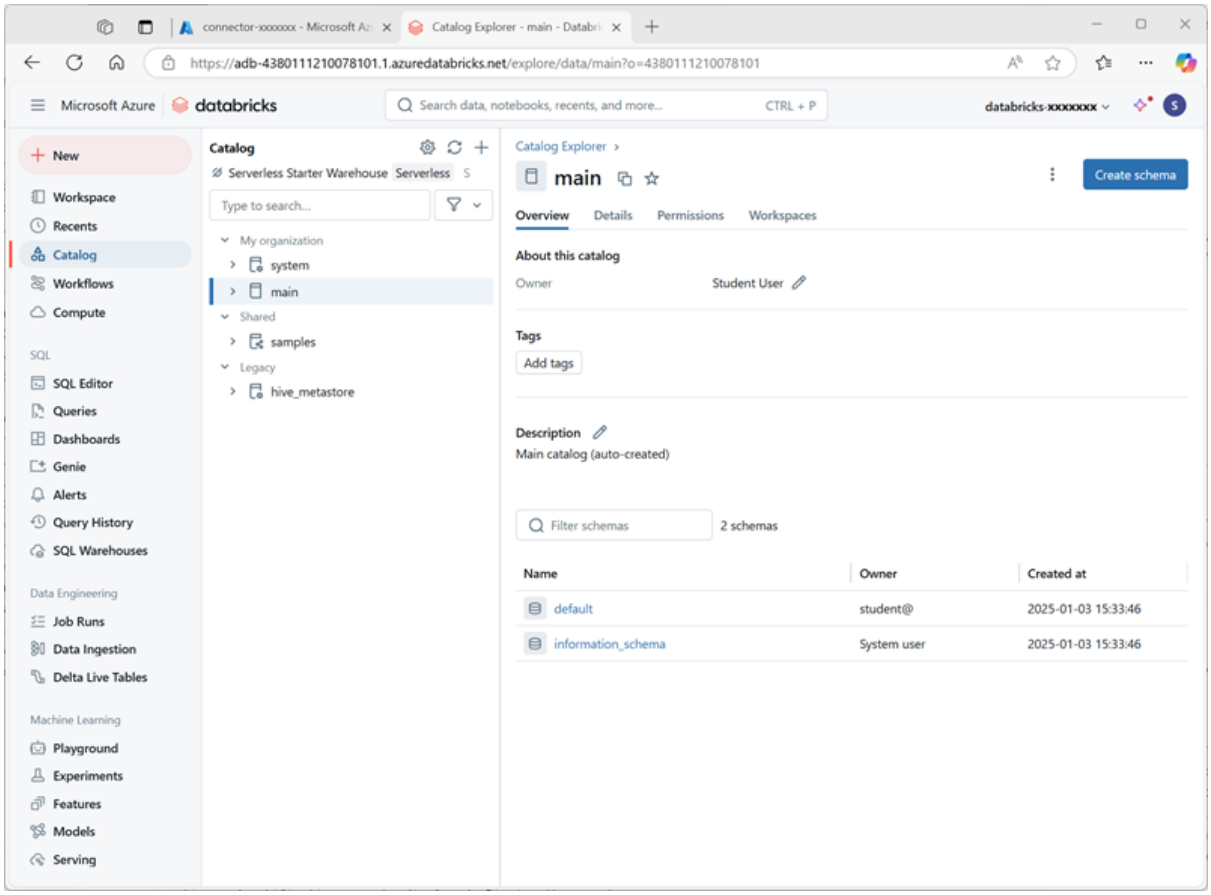


Work with data in Unity Catalog

Now that you've assigned an eternal metastore and enabled Unity Catalog, you can use it to work with data in Azure Databricks.

Create and load a table

1. Close the Azure Databricks account console browser tab and return to the tab for your Azure Databricks workapace. Then refresh the browser.
2. On the **Catalog** page, select the **Main** catalog for your organization and note that schemas named **default** and **Information_schema** have already been created in your catalog.

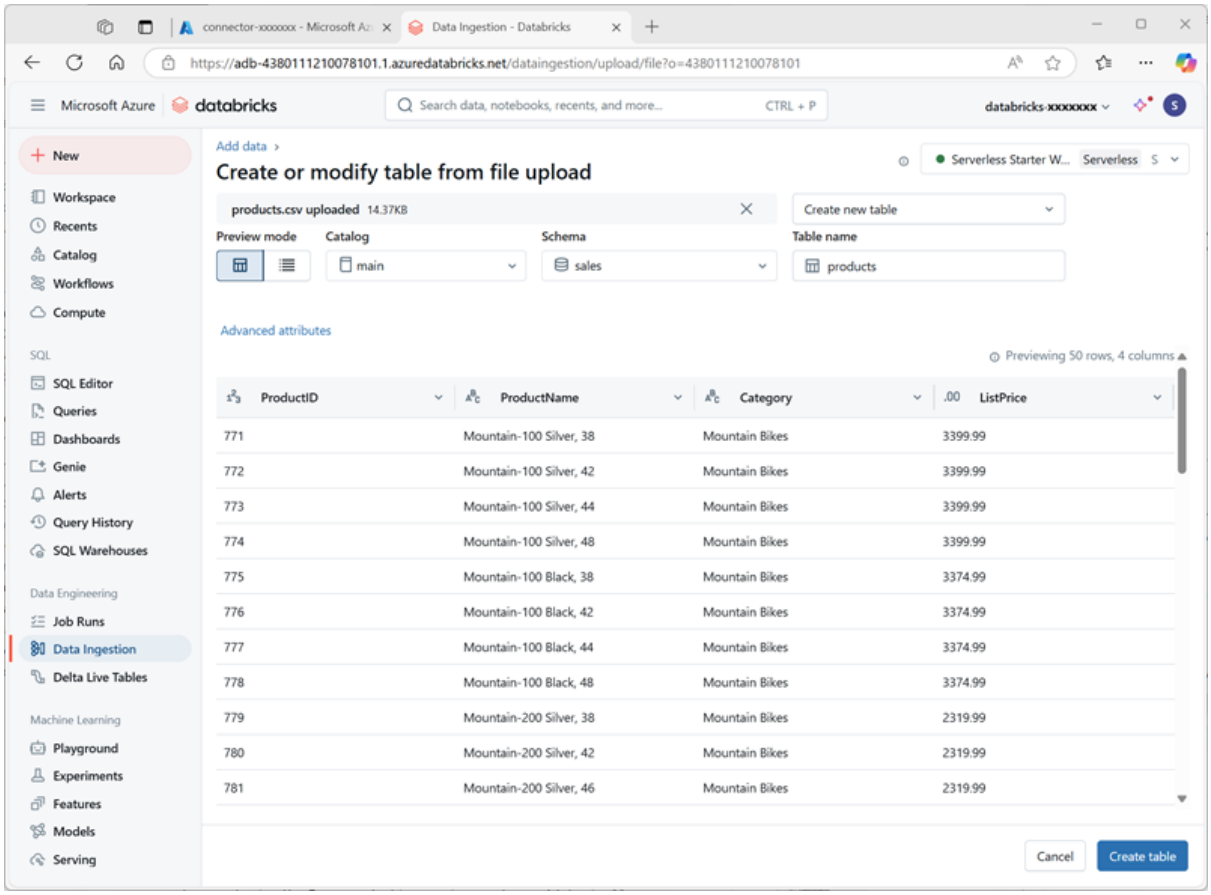


3. Select **Create Schema** and create a new schema named sales (leave the storage location unspecified so the default metastore for the catalog will be used).

4. In a new browser tab, download the **products.csv** file from https://raw.githubusercontent.com/parveenkrraina/essilor-batch02/refs/heads/main/data/products.csv to your local computer, saving it as **products.csv**.

5. In the Catalog explorer in Azure Databricks workspace, with the **sales** schema selected, select **Create** > **Create table**. Then upload the **products.csv** file you downloaded to create a new table named **products** in the **sales** schema.

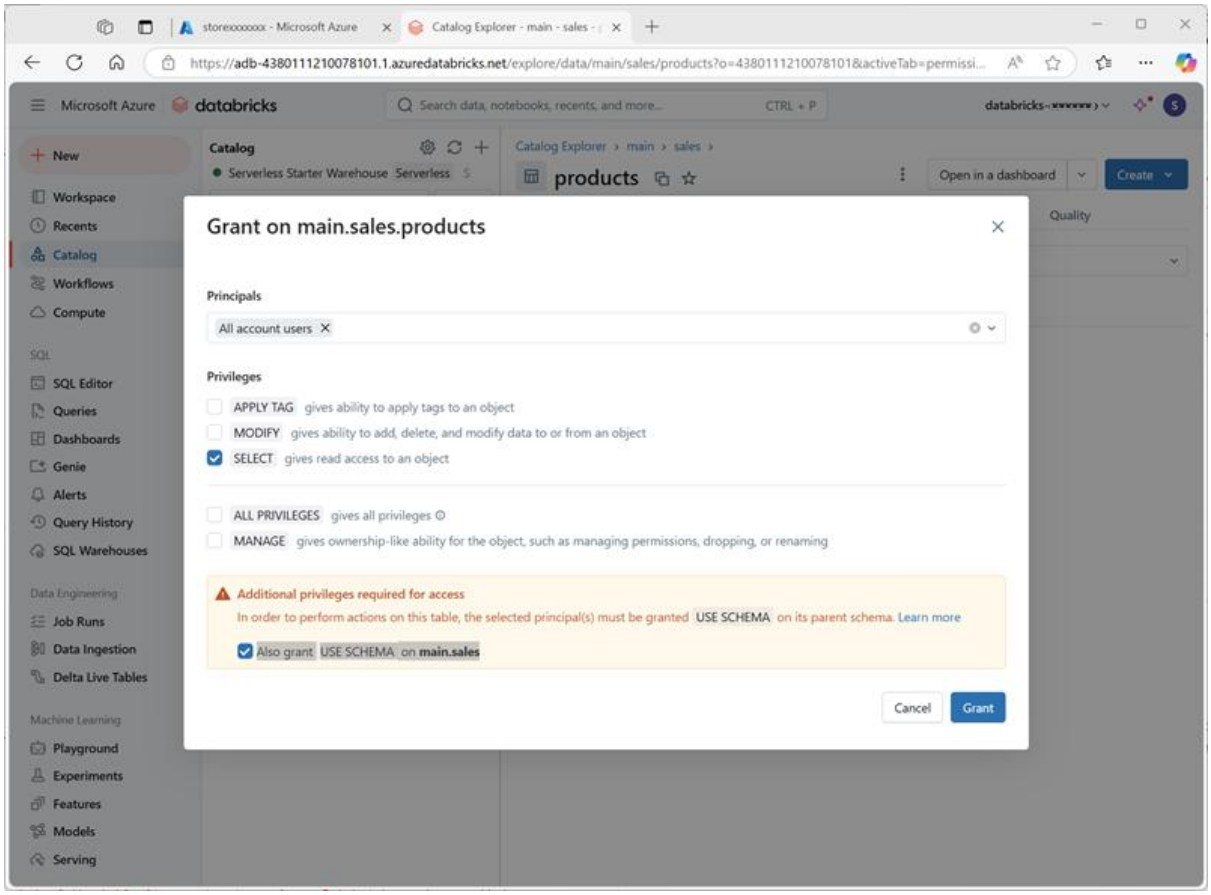
Note: You may need to wait a few minutes for serverless compute to start.



6. Create the table. If an AI-generated description is suggested, accept it.

Manage permissions

1. With the **products** table selected, on the **Permissions** tab, verify that by default there are no permissions assigned for the new table (you can access it because you have full administrative rights, but no other users can query the table).
2. Select **Grant**, and configure access to the table as follows:
    - **Principals**: All account users
    - **Privileges**: SELECT
    - **Additional privileges required for access**: Also grant USE SCHEMA on main.sales



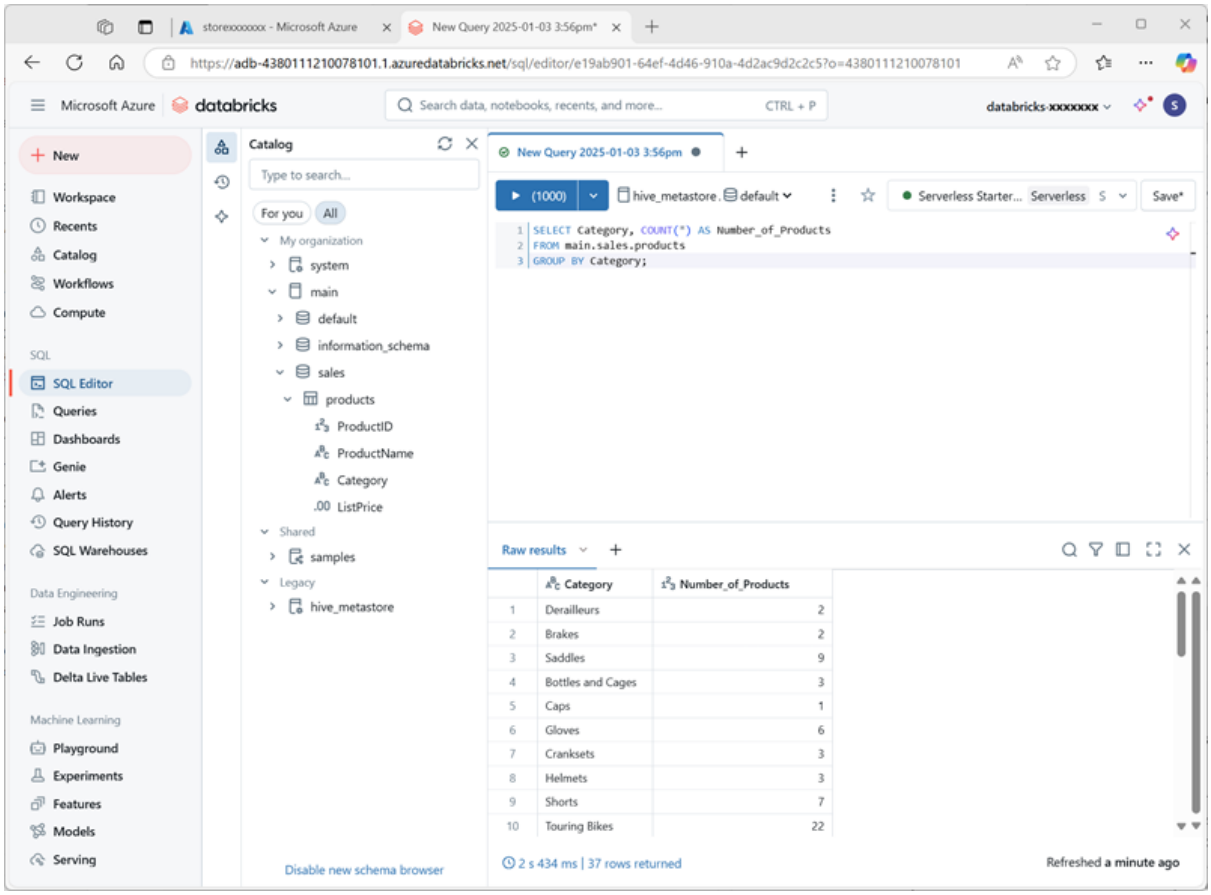
Track lineage

1. On the **+ New** menu, select **Query** and create a new query with the following SQL code:

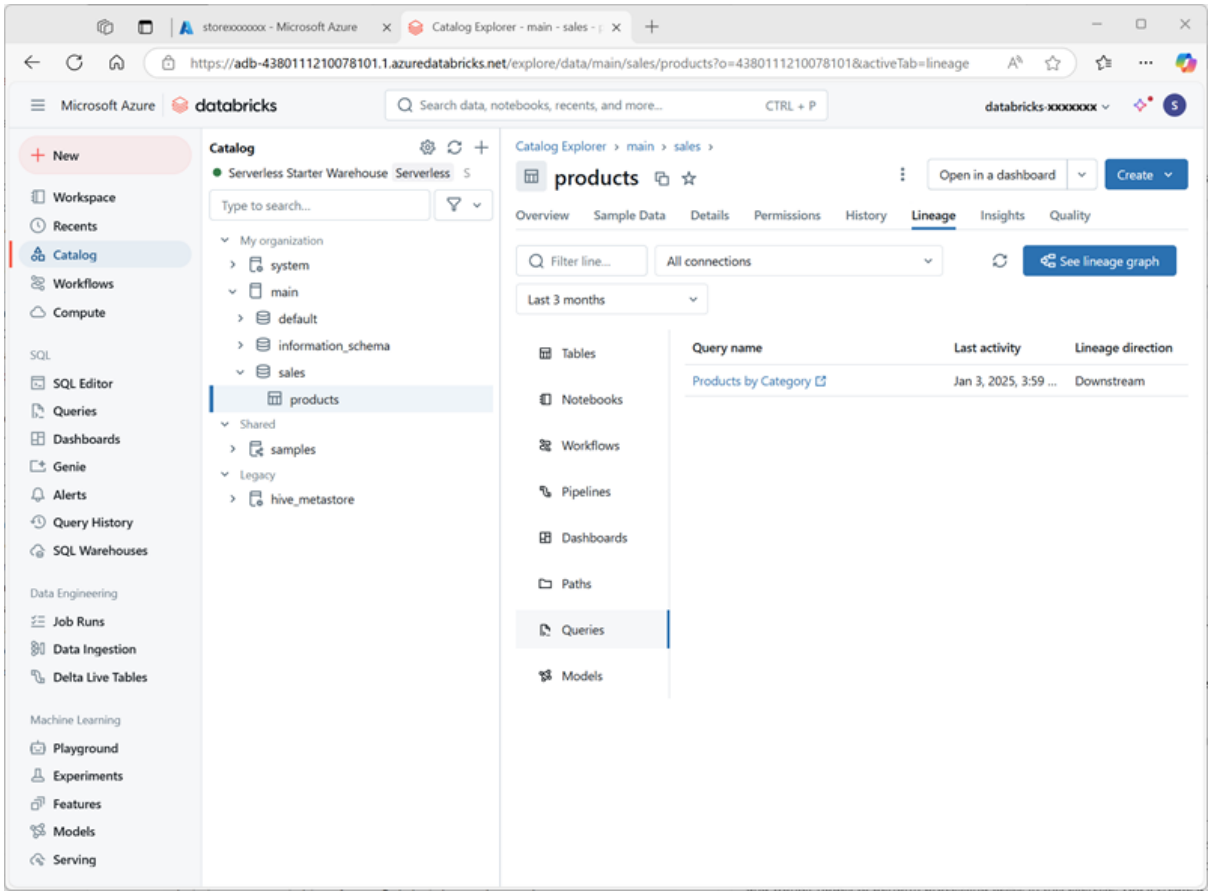
sql

```
SELECT Category, COUNT(*) AS Number_of_Products
FROM main.sales.products
GROUP BY Category;
```

2. Ensure serverless compute is connected, and run the query to see the results.



3. Save the query as Products by Category in the workspace folder for your Azure Databricks user account.
4. Return to the **Catalog** page. Then expand the **main** catalog and the **sales** schema, and select the **products** table.
5. On the **Lineage** tab, select **Queries** to verify that the lineage from the query you created to the source table has been tracked by Unity Catalog.


storexxxxxxx - Microsoft Azure
Catalog Explorer - main - sales -
https://adb-4380111210078101.1.azuredatabricks.net/explore/data/main/sales/products?o=4380111210078101&activeTab=lineage
Microsoft Azure
databricks
Search data, notebooks, recents, and more...
CTRL + P
databricks-xxxxxxx
New
Workspace
Recents
Catalog
Workflows
Compute
SQL
SQL Editor
Queries
Dashboards
Genie
Alerts
Query History
SQL Warehouses
Data Engineering
Job Runs
Data Ingestion
Delta Live Tables
Machine Learning
Playground
Experiments
Features
Models
Serving
Catalog
Serverless Starter Warehouse
Type to search...
My organization
system
main
default
information_schema
sales
products
Shared
samples
Legacy
hive_metastore
Catalog Explorer › main › sales ›
products
Open in a dashboard
Create
Overview
Sample Data
Details
Permissions
History
Lineage
Insights
Quality
Filter line...
All connections
See lineage graph
Last 3 months
Tables
Notebooks
Workflows
Pipelines
Dashboards
Paths
Queries
Models
Query name
Last activity
Lineage direction
Products by Category
Jan 3, 2025, 3:59 ...
Downstream