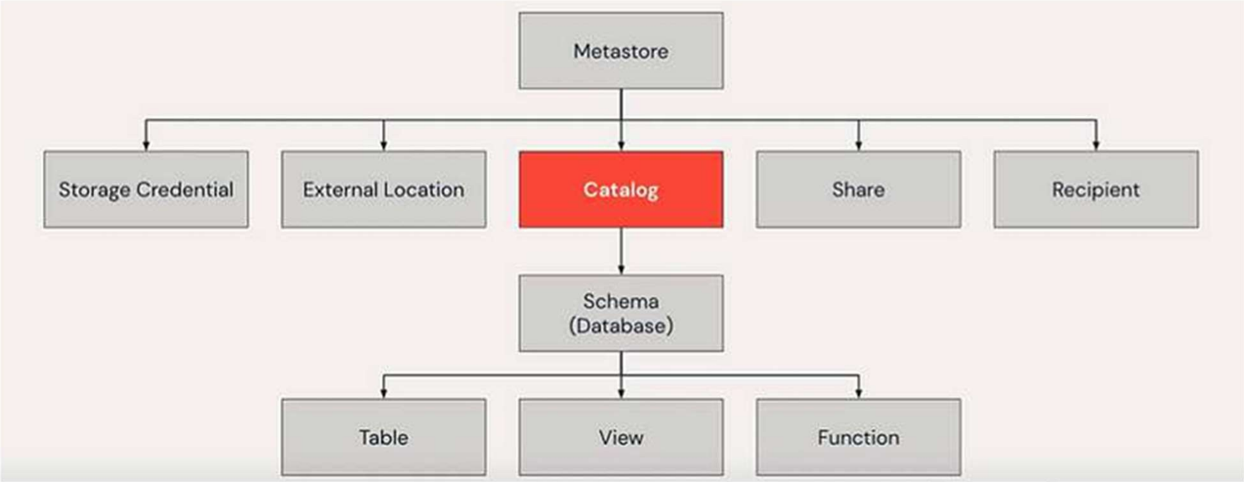
Mini Project: Implementation of Key Data Governance Capabilities Using Unity Catalog

Objective

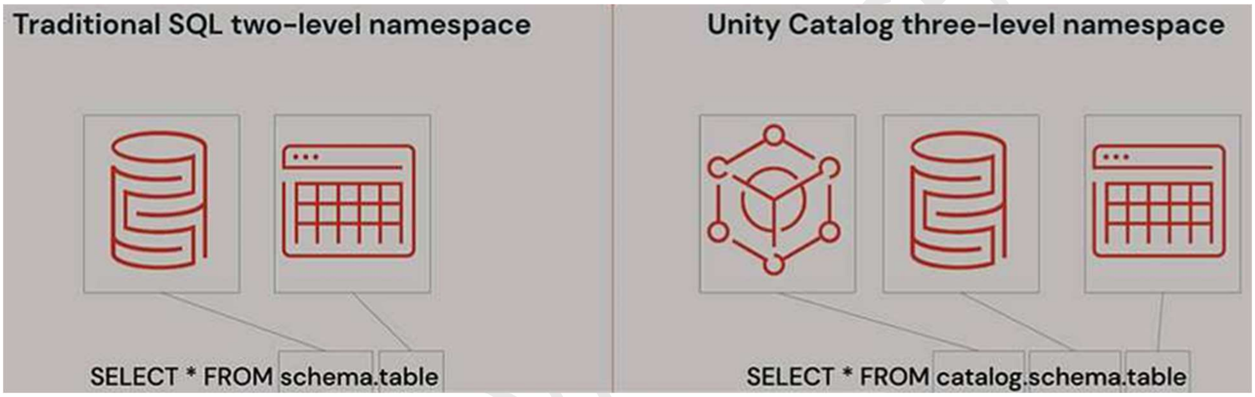
To explore and implement key data governance capabilities offered by Unity Catalog, such as **Data Discovery**, **Data Audit**, **Data Lineage**, and **Data Access Control**, by leveraging Azure Databricks. The project aims to enhance understanding and practical experience with Unity Catalog for managing and governing data effectively.

Key Features of Unity Catalog

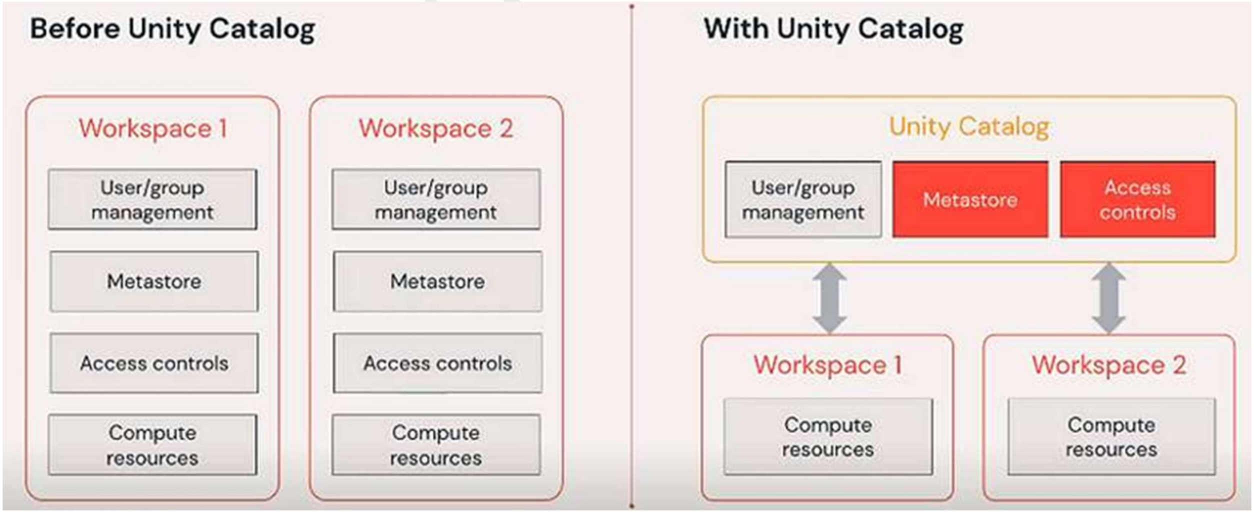
1. **Data Discovery**:
   - Enables users to search and explore datasets easily within an organization.
   - Importance: Helps analysts, engineers, and data scientists locate datasets quickly, reducing time and effort.
2. **Data Audit**:
   - Maintains an audit trail for all data access and modifications.
   - Importance: Ensures compliance with regulations and helps identify unauthorized activities.
3. **Data Lineage**:
   - Tracks the origin and transformation of data throughout its lifecycle.
   - Importance: Provides transparency, improves troubleshooting, and ensures trust in the data.
4. **Data Access Control**:
   - Manages fine-grained access to data based on roles and responsibilities.
   - Importance: Protects sensitive data and ensures only authorized users access specific datasets.



Metastore
Storage Credential
External Location
Catalog
Share
Recipient
Schema (Database)
Table
View
Function



Traditional SQL two-level namespace
SELECT * FROM schema.table
Unity Catalog three-level namespace
SELECT * FROM catalog.schema.table



Before Unity Catalog
Workspace 1
User/group management
Metastore
Access controls
Compute resources
Workspace 2
User/group management
Metastore
Access controls
Compute resources
With Unity Catalog
Unity Catalog
User/group management
Metastore
Access controls
Workspace 1
Compute resources
Workspace 2
Compute resources

Project Implementation Steps

Step 1: Set Up Azure Databricks

1. **Create an Azure Databricks Workspace**:
   - Go to the Azure Portal.
   - Create a resource and select **Azure Databricks**.
   - Configure the workspace (e.g., pricing tier, region, etc.).
2. **Set Up Unity Catalog**:

Step 2: Enable Unity Catalog

1. **Configure Storage for Unity Catalog**:
   - Set up an **Azure Data Lake Storage Gen2** for storing Unity Catalog metadata.
   - Create a storage account and configure it with managed identities.
2. **Attach Unity Catalog to Databricks**:
   - In Databricks Admin Console, configure Unity Catalog and associate it with the created workspace.

Step 3: Implement Data Governance Capabilities

1. Data Discovery

- **Procedure**:
  1. Organize datasets into catalogs and schemas.
  2. Use Databricks SQL to query and explore datasets.
  3. Implement tags and metadata for each dataset to improve searchability.
- **Verification**:
  - Use the Databricks search bar to locate specific datasets based on tags or names.

2. Data Audit

- **Procedure**:
  1. Enable audit logging in the workspace.

2. Configure logs to be stored in Azure Log Analytics or Azure Storage.
3. Monitor access and activity logs for datasets.

- **Verification**:
  - Check audit logs for records of data access, modification, and lineage.

3. Data Lineage

- **Procedure**:
  1. Enable automatic data lineage tracking in Unity Catalog.
  2. Run queries or workflows to generate lineage records.
  3. Visualize lineage using the lineage graph feature in Databricks.
- **Verification**:
  - Inspect lineage graphs to ensure all transformations and data sources are correctly linked.

4. Data Access Control

- **Procedure**:
  1. Define roles and permissions in Unity Catalog.
  2. Assign users and groups to roles with specific permissions (e.g., read, write).
  3. Enforce row- or column-level security based on user roles.
- **Verification**:
  - Test access control by attempting data operations with different user roles.

Step 4: Validate and Document Results

1. **Validation**:
   - Conduct a series of tests for each feature (e.g., searchability, audit log completeness, lineage accuracy, access restrictions).
2. **Documentation**:
   - Record observations, challenges, and solutions encountered during implementation.

Sample Use Case

Scenario: A retail organization wants to manage customer transaction data. They need to ensure that only authorized personnel can access sensitive data, audit who accessed the data, and understand the lineage of data transformations applied during analytics.

1. **Data Discovery**:
    - Catalog data into "Customer Data" and "Transaction Data" schemas.
    - Add metadata tags: `sensitive`, `customer`, `transactions`.
2. **Data Audit**:
    - Track access to sensitive datasets and identify unauthorized access.
3. **Data Lineage**:
    - Understand the transformations applied to transaction data to generate monthly sales reports.
4. **Data Access Control**:
    - Allow analysts to view transaction summaries but restrict access to customer details.

Conclusion

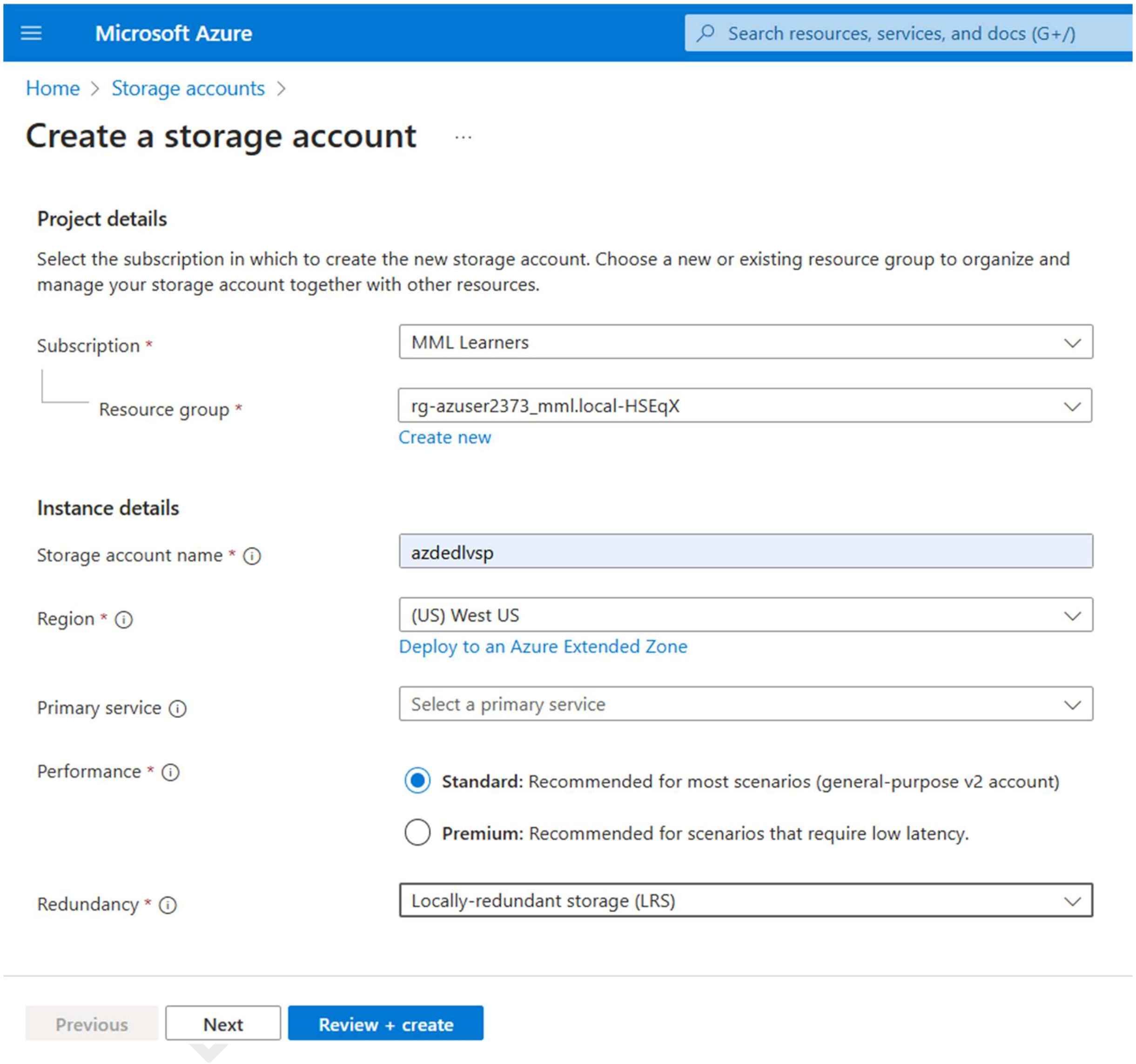
Unity Catalog in Azure Databricks offers a robust solution for implementing enterprise-grade data governance capabilities. By following this documentation, organizations can ensure their data is discoverable, auditable, traceable, and securely managed.

References

- Unity Catalog Documentation (Databricks)
- Azure Databricks Documentation

SCREENSHOTS

Creating storage account



Microsoft Azure
Search resources, services, and docs (G+/)
Home > Storage accounts >

Create a storage account ...

Project details

Select the subscription in which to create the new storage account. Choose a new or existing resource group to organize and manage your storage account together with other resources.

Subscription *: MML Learners

Resource group *: rg-azuser2373_mml.local-HSEqX

Instance details

Storage account name * ⓘ: azdedlvsp

Region * ⓘ: (US) West US

Primary service ⓘ: Select a primary service

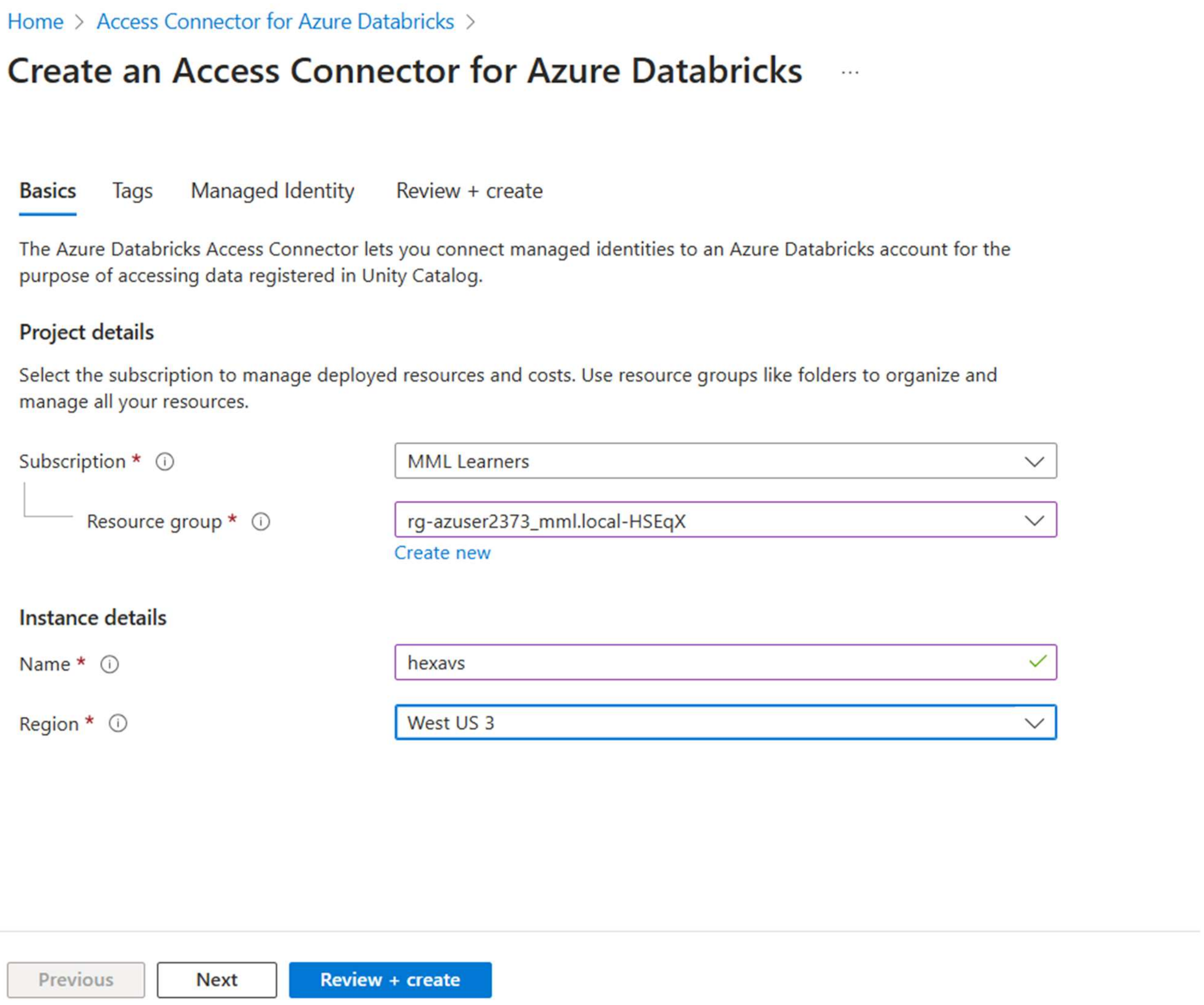
Performance * ⓘ:

● **Standard:** Recommended for most scenarios (general-purpose v2 account)

○ **Premium:** Recommended for scenarios that require low latency.

Redundancy * ⓘ: Locally-redundant storage (LRS)

Creating access connector



Home > Access Connector for Azure Databricks >

Create an Access Connector for Azure Databricks ...

The Azure Databricks Access Connector lets you connect managed identities to an Azure Databricks account for the purpose of accessing data registered in Unity Catalog.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

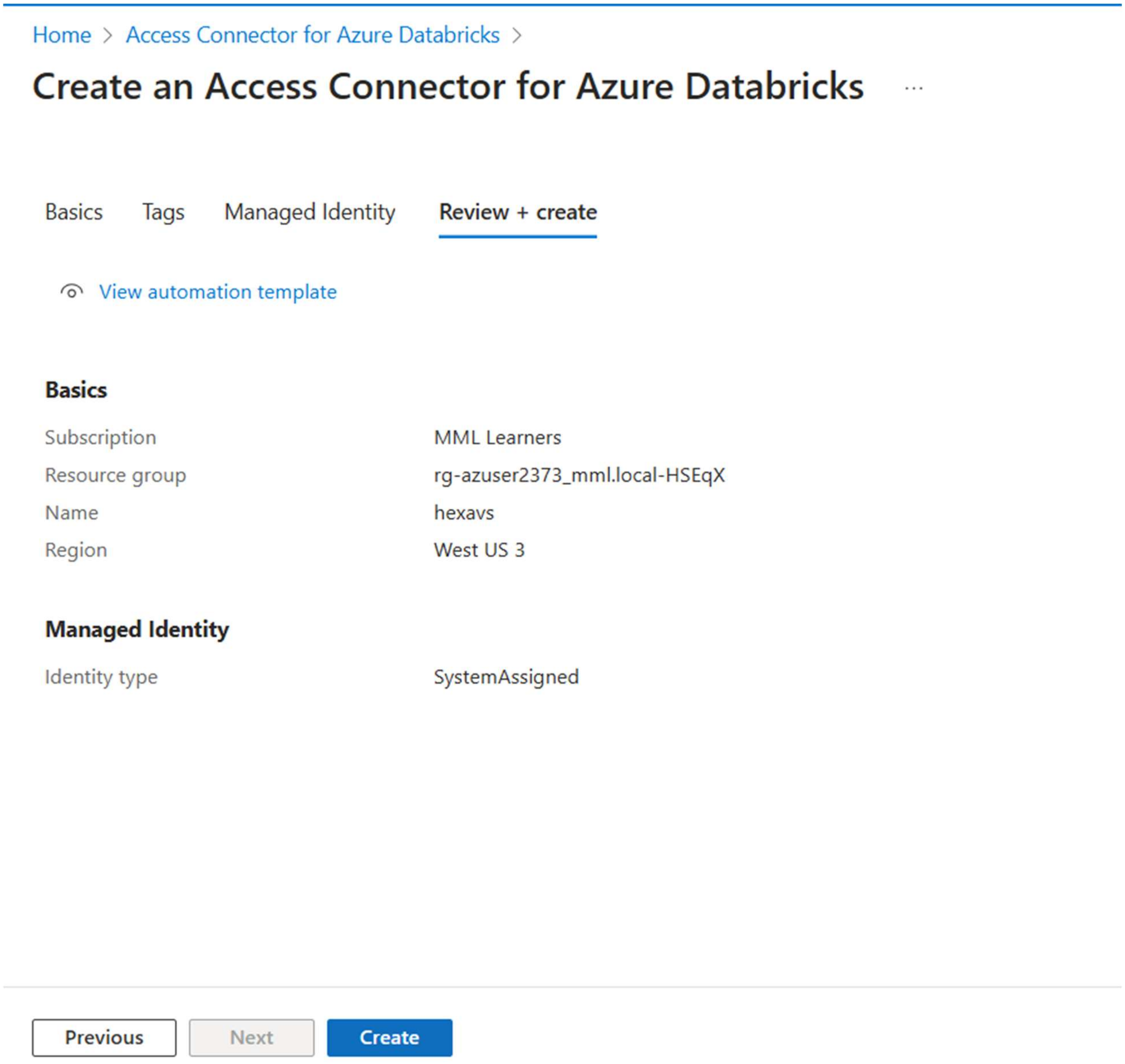
Subscription * : MML Learners

Resource group * : rg-azuser2373_mml.local-HSEqX

Instance details

Name * : hexavs

Region * : West US 3



Home > Access Connector for Azure Databricks >

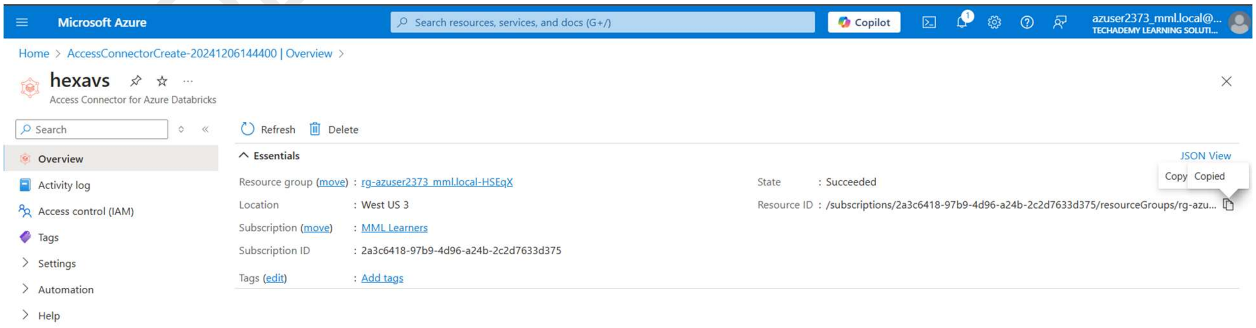
Create an Access Connector for Azure Databricks ...

Basics

Subscription	MML Learners
Resource group	rg-azuser2373_mml.local-HSEqX
Name	hexavs
Region	West US 3

Managed Identity

Identity type	SystemAssigned



Microsoft Azure
Search resources, services, and docs (G+/)
Copilot
azuser2373_mml.local@...
TECHADEMY LEARNING SOLUTI...

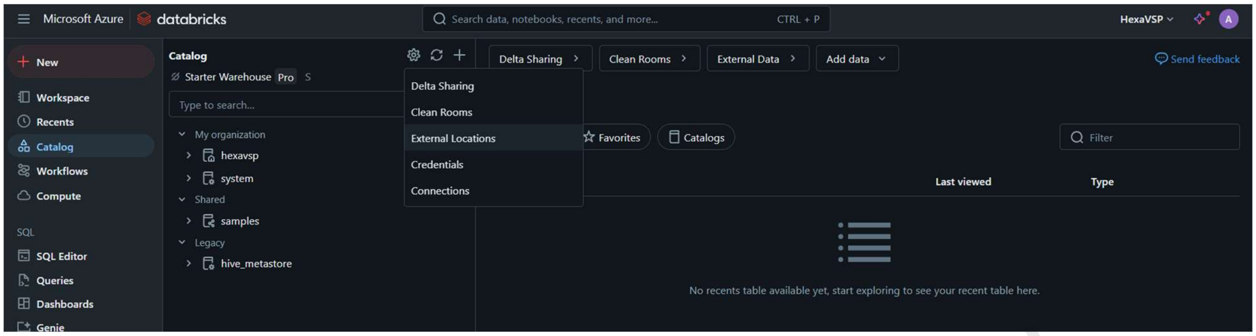
Home > AccessConnectorCreate-20241206144400 | Overview >

hexavs

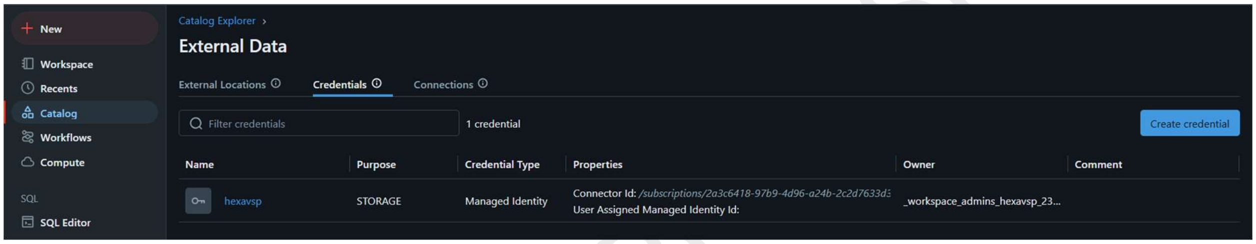
Access Connector for Azure Databricks

Resource group (move)	: rg-azuser2373_mml.local-HSEqX	State	: Succeeded
Location	: West US 3	Resource ID	: /subscriptions/2a3c6418-97b9-4d96-a24b-2c2d7633d375/resourceGroups/rg-azu...
Subscription (move)	: MML Learners		
Subscription ID	: 2a3c6418-97b9-4d96-a24b-2c2d7633d375		
Tags (edit)	: Add tags		

In azure databricks, go to catalog click on + and give external location.

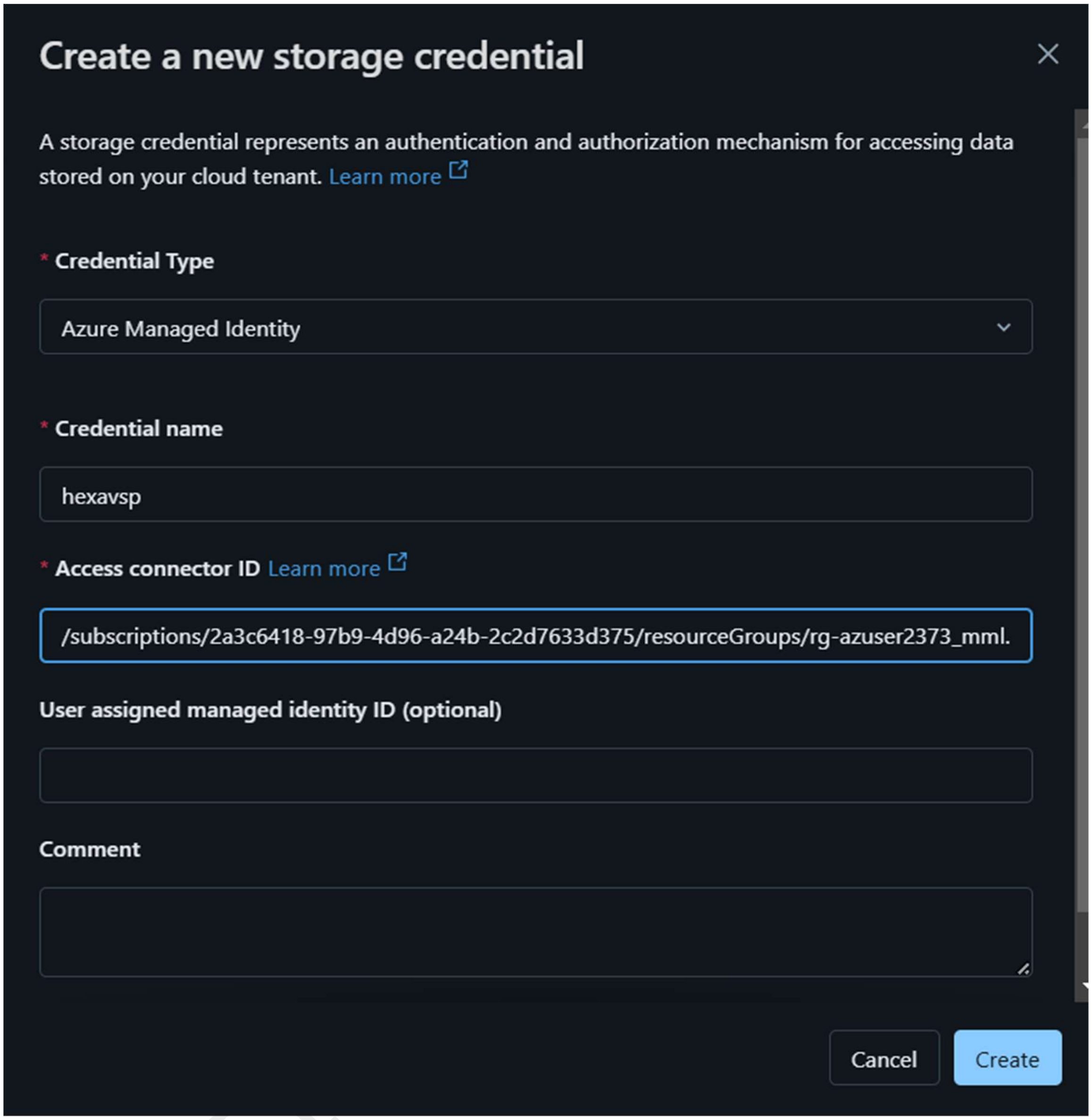


Click on create credential



Name	Purpose	Credential Type	Properties	Owner	Comment
hexavsp	STORAGE	Managed Identity	Connector Id: /subscriptions/2a3c6418-97b9-4d96-a24b-2c2d7633d3 User Assigned Managed Identity Id:	_workspace_admins_hexavsp_23...	

Provide necessary details.



Create a new storage credential

A storage credential represents an authentication and authorization mechanism for accessing data stored on your cloud tenant. Learn more

* Credential Type

Azure Managed Identity

* Credential name

hexavsp

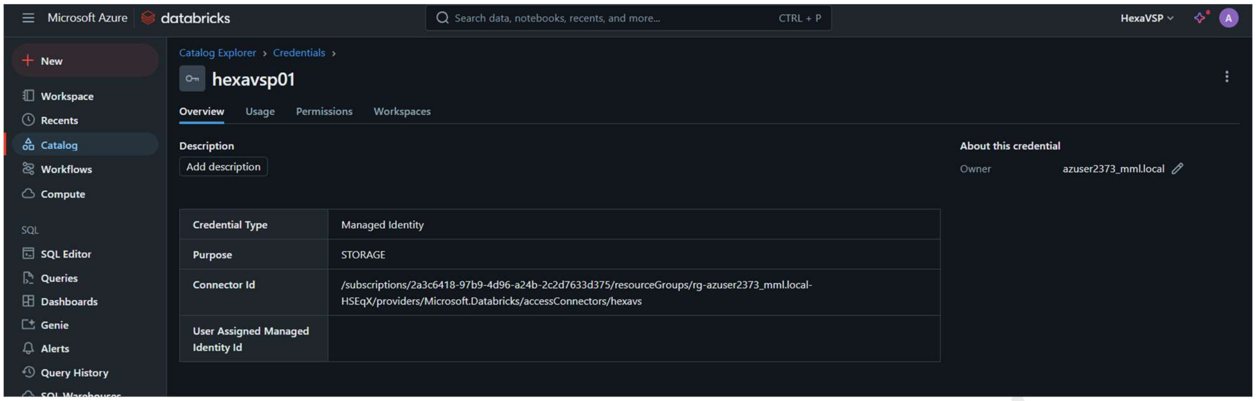
* Access connector ID Learn more

/subscriptions/2a3c6418-97b9-4d96-a24b-2c2d7633d375/resourceGroups/rg-azuser2373_mml.

User assigned managed identity ID (optional)

Comment

Catalog created



hexavsp01

Description

About this credential

Owner azuser2373_mml.local

Credential Type	Managed Identity
Purpose	STORAGE
Connector Id	/subscriptions/2a3c6418-97b9-4d96-a24b-2c2d7633d375/resourceGroups/rg-azuser2373_mml.local-HSEqX/providers/Microsoft.Databricks/accessConnectors/hexavs
User Assigned Managed Identity Id	

And to further process for accessing data, currently account doesn’t support for account actions.