Unity Catalog

Introduction

Unity catalog is a **unified governance** solution for data and AI assets on **lake house**. It provides **centralized metadata management**, **standard SQL syntax for permissions**, user-level audit logs, and more.

📌 Unity catalog is a **premium offering from Databricks** available on major Cloud platforms like AWS, Azure, and Google

📌 Unity catalog allows for a **centralized metadata layer**, **enabling sharing of databases, tables, and views across multiple Databricks workspaces**.

📌 Standard SQL syntax can be used to **grant permissions on databases, tables, and views, which work across all workspaces.**

📌 Unity catalog captures user-level audit logs and offers features like **data discovery and lineage**.

📌 The architecture of Unity catalog involves the management of identity and metastore, where Databricks identities and metadata reside.

📌 Without Unity catalog, access control, user management, and metadata are decentralized to each Databricks workspace.

📌 Unity catalog follows a hierarchical naming standard, with metastore at the root level, catalogs containing schemas, and databases containing tables and views.

📌 Enabling Unity catalog involves creating a premium edition Databricks workspace, obtaining initial admin access from Azure AD Global admin, and creating metastore.

Catalog configure prerequisites

It covers creating a user, assigning roles, and creating necessary resources like a data bricks workspace and storage account.

📝 The steps to create a user for the demo are explained, including navigating to Azure Active Directory, creating a new user, assigning the Global Administrator role, and noting down the user ID for future use.

⚙️ The process of assigning certain roles to the user at the subscription level is described, including searching for the subscription, accessing Identity and Access Management (IAM) control, and adding the role assignment.

🌐 The video instructs the user to log in with the newly created user account, change the password, and optionally set up multi-factor authentication. It also discusses creating a resource group, data bricks workspace, storage account, and access connector.

⏭️ The viewer is directed to launch the data bricks workspace and access the Unity catalog page for further configuration.

📺 The video concludes by mentioning that the prerequisites have been covered and the next video will focus on Unity catalog configuration.

📝 Creating a user and assigning roles in Azure Active Directory is crucial for managing access and permissions within Unity catalog.

⚙️ Assigning roles at the subscription level ensures that the user has the necessary permissions to deploy and manage resources in Unity.

🌐 Logging in with the newly created user account and configuring multi-factor authentication enhances security for accessing Unity catalog.

⏭️ Launching the data bricks workspace provides the platform for Unity catalog configuration and resource creation.

📺 The video script provides a comprehensive overview of the prerequisites, setting up the foundation for a smooth Unity catalog configuration process.

how to create a Unity catalog meta store in this video by logging into the Azure portal and following the steps provided.
🌐 Use accounts.ashodatabricks.net for Unity catalog access

🔑 Use Azure AD credential to login

⚙️ Explore workspace, data, user management, and settings tabs

📁 Create a meta store using ADLs storage

🔄 Integrate meta store with workspaces

✅ Validate meta store attachment from workspaceKey Insights

💡 Unity catalog meta store is used to store databases, table schema, etc., and is backed by ADLs storage.

💡 The URL “accounts.ashodatabricks.net” is used to access the Unity catalog.

💡 Azure AD credentials are required to login to the Unity catalog account page.

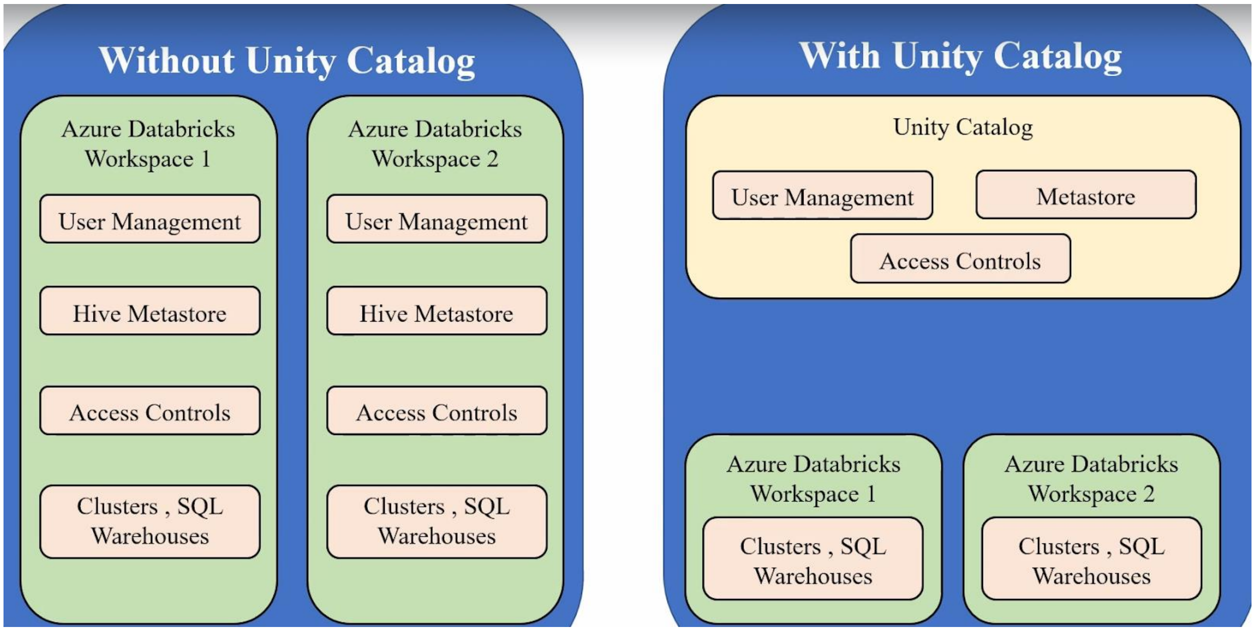
💡 The workspace menu lists available workspaces, while the data menu displays the created meta store.

💡 User management allows onboarding of users, service principles, and groups.

💡 Settings enable user provisioning and integration with Azure AD for automatic user loading.

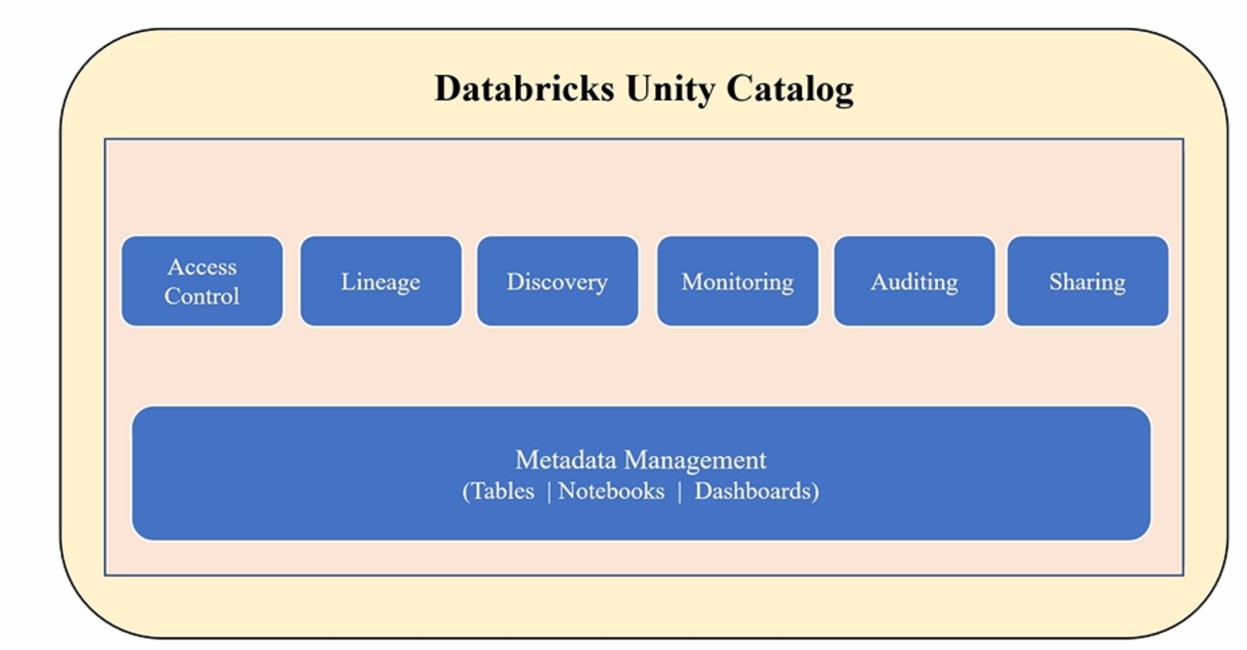
💡 The meta store can be validated from the workspace by checking the attached meta store.

There will be a business/ project requirement to create multiple databricks workspace for each environment.
Where every workspace needs to be managed individually for user Management, Access control.
There is no centralized location for auditing regarding the governance, access.



Without Unity Catalog
Azure Databricks Workspace 1
User Management
Hive Metastore
Access Controls
Clusters , SQL Warehouses
Azure Databricks Workspace 2
User Management
Hive Metastore
Access Controls
Clusters , SQL Warehouses
With Unity Catalog
Unity Catalog
User Management
Metastore
Access Controls
Azure Databricks Workspace 1
Clusters , SQL Warehouses
Azure Databricks Workspace 2
Clusters , SQL Warehouses

With unity catalog, we can have a centralized location for access to database objects, User Management.



with unity catalog, we can have all the above features at one place.

Databricks Access Connector

We cannot grant permissions to unity catalog to ADLS Gen 2 storage account, because it is not a separate service, we need to have a managed identity

Creating Databricks Access Connector

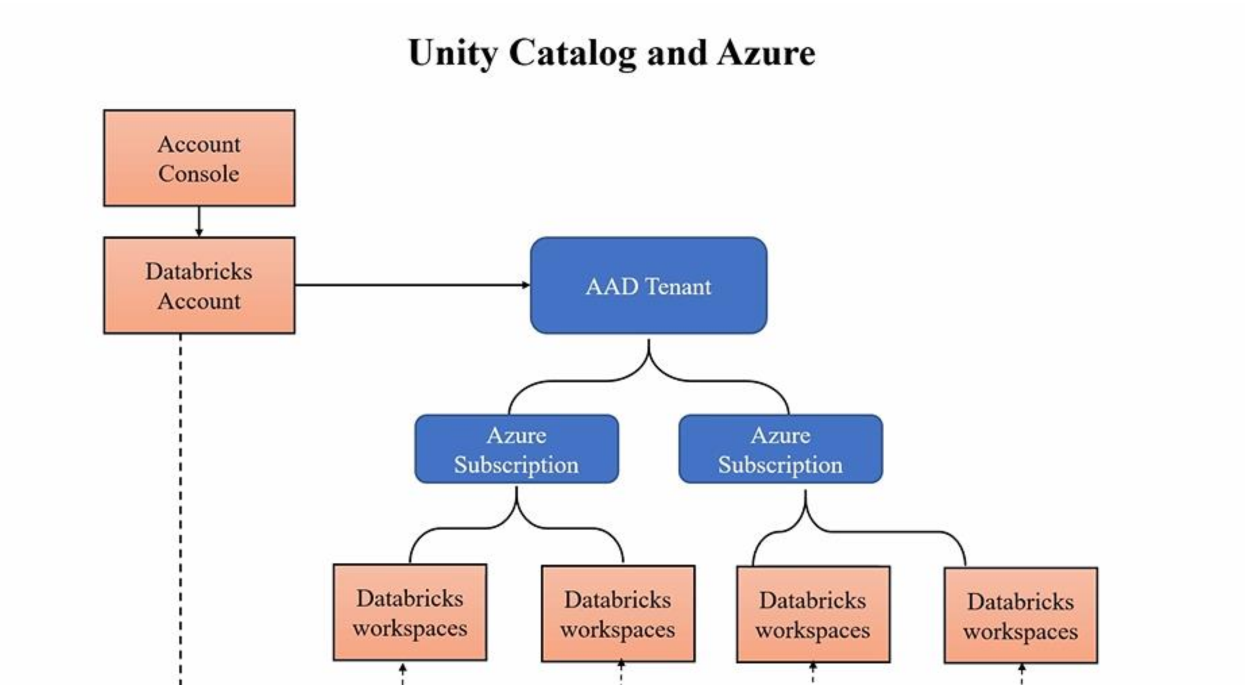
1. Create access connector through portal using create resource option.
2. Go to ADLS Gen 2 storage account, navigate to the Access Control(IAM) permissions.
    a. Add Role assignment–**(Storage Blob Data contributor)**
    b. Managed Identity–(Select the access connector created previously)

Prerequisites for creating a unity catalog

3. A premium databricks workspace is required.
4. Metastore should be created.

Creating a meta store

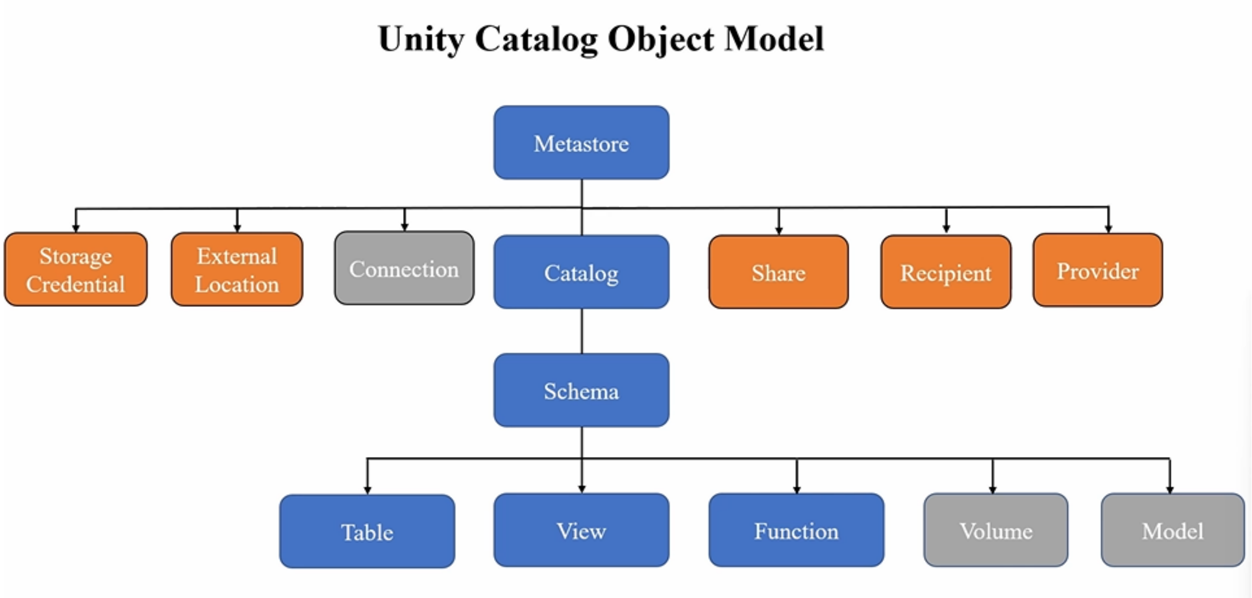
5. Admin access is required for the creating a metastore.
6. Go to accounts.azuredatabricks.net→data section → create a metastore
7. It is good to have storage accounts, metastore in a same region
8. *Only one metastore can be created per region, multiple metastores cannot be created in a region.*
9. For storing the managed tables in the metasore ADLS path is required.(abfss://<container_name>@<storageaccount>.dfs.core.windows.net/<folder_path>)
10. Provide the databricks access connector created in the previous step
11. Attach the premium workspace to enable the unity catalog



Account is associated at the Azure Active Directory tenant level, with this all the workspaces that are present in the different subscriptions are visible in the Account console.

Unity catalog Object Model

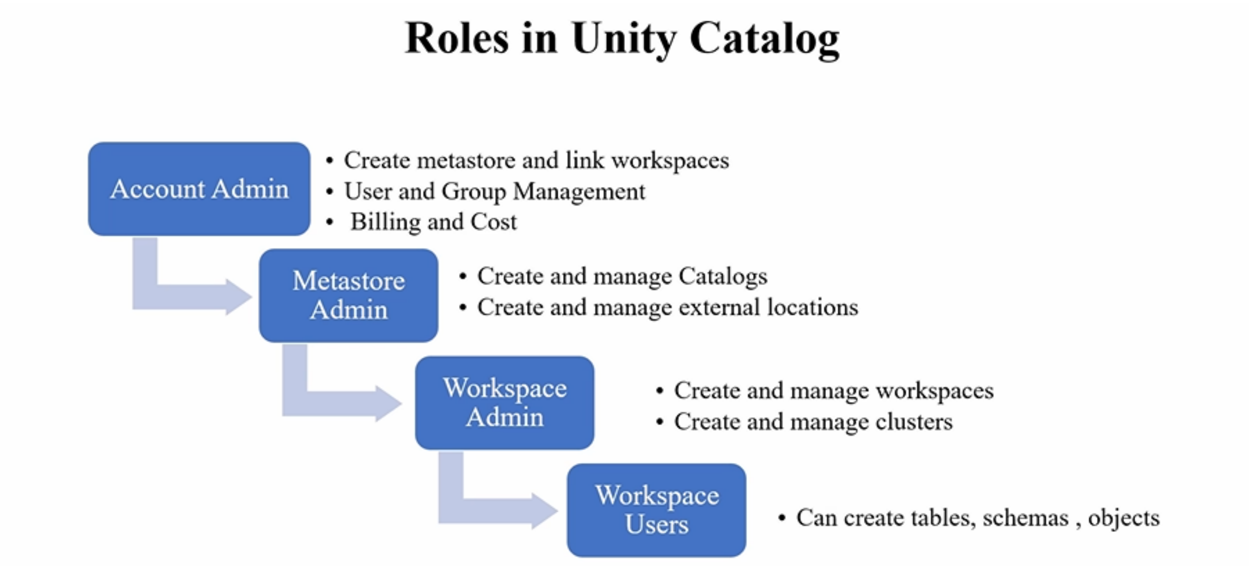
hive metastore and unity catalog enabled metastore are both different, by default a workspace comes with hive metastore, we cannot share the database objects through the hive metastore.



Unity catalog follows a 3 level namespace, a metastore can have multiple catalogs for each environment(Development, Staging and Production).

A catalog can have multiple schemas, which inturn have low level objects, Tables, Views, Function, Volumes, ML Models.

Roles in Unity Catalog


Roles in Unity Catalog
Account Admin
• Create metastore and link workspaces
• User and Group Management
• Billing and Cost
Metastore Admin
• Create and manage Catalogs
• Create and manage external locations
Workspace Admin
• Create and manage workspaces
• Create and manage clusters
Workspace Users
• Can create tables, schemas , objects

Creating users in the Microsoft Entra ID and user management

12. Create users through the Microsoft Entra ID
13. log in to the Databricks Admin account→Manage Account
14. Add users and create groups(add the users to the groups) through → User Management
15. Go to the workspace and select the permissions, add permission→select the group

Cluster policies

16. Users can be restricted from creating clusters through policies.
17. To limit the users to create cluster with specific attributes
18. To control the user ability to provision the cluster with a set of rules.
19. Cluster policies have ACLs limiting the use to specific users and groups.

Sample policy for creating a single node cluster with DBR version 13.3 LTS and Auto Termination time of   20 min.

```
{
  "node_type_id": {
    "type": "allowlist",
    "values": [
      "Standard_DS3_v2"
    ]
  },
  "spark_version": {
    "type": "fixed",
    "value": "13.3.x-scala2.12"
  },
  "runtime_engine": {
    "type": "fixed",
    "value": "STANDARD",
    "hidden": true
  },
  "num_workers": {
    "data_security_mode": {
    "type": "fixed",
    "value": "SINGLE_USER"
  },
  "cluster_type": {
    "type": "fixed",
    "value": "all-purpose"
  },
  "instance_pool_id": {
    "type": "forbidden",
"hidden": true
  },
  "azure_attributes.availability": {
    "type": "fixed",
    "value": "ON_DEMAND_AZURE",
    "hidden": true
  },
```

"type": "fixed",
"value": 0,
"hidden": true
},
	"spark_conf.spark.databricks.cluster.profile":
	{
	"type": "fixed",
	"value": "singleNode",
	"hidden": true
	},
	"autotermination_minutes": {
	"type": "fixed",
	"value": 20
	}
	}

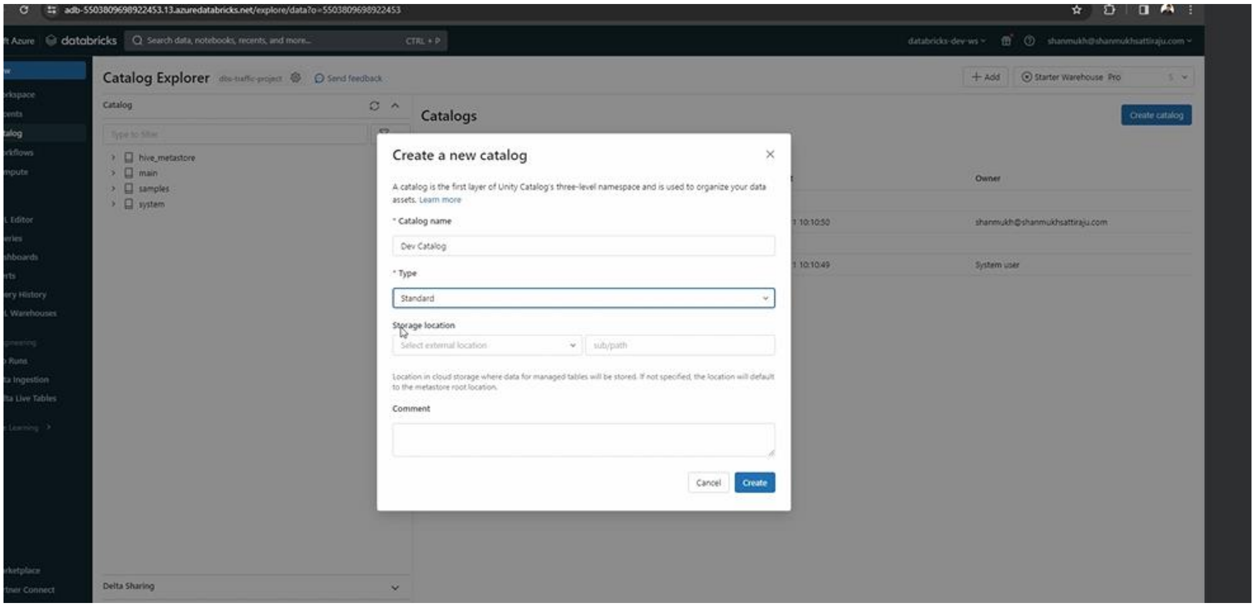
Creating a cluster pool

While executing a job if a cluster is in terminated state, to initiate the cluster, it will take 4-5 minutes, in the real time scenario it will have impact on job SLAs.

To mitigate the issue, a job pool with a Max capacity and IDLE resources can be provisioned, with this always a specific number of virtual machines will be readily available for executing the job, which reduces the lead times considerably.

In addition to it, cluster policy can be modified to select the cluster pools.

Creating a catalog



Create a new catalog

A catalog is the first layer of Unity Catalog's three-level namespace and is used to organize your data assets. Learn more

* Catalog name: Dev Catalog

* Type: Standard

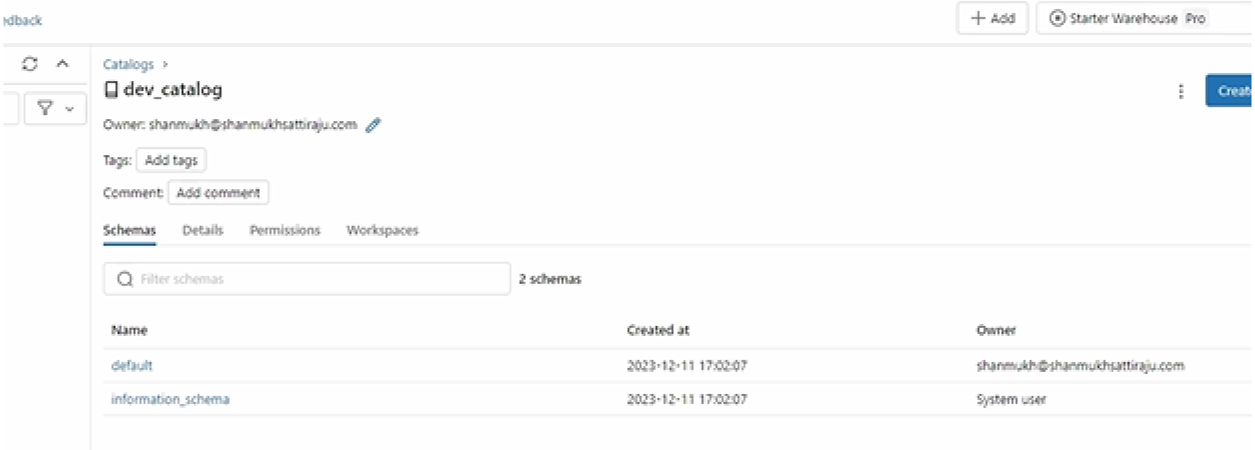
Storage location: Select external location | sub/path

Location in cloud storage where data for managed tables will be stored. If not specified, the location will default to the metastore root location.

Comment

We can either create a Standard or Foreign catalog depending on the use case

If all the objects created under the catalog needs to be stored in an external location(an location can be specified)



Catalogs >

dev_catalog

Owner: shanmukh@shanmukhsattiraju.com

Tags: Add tags

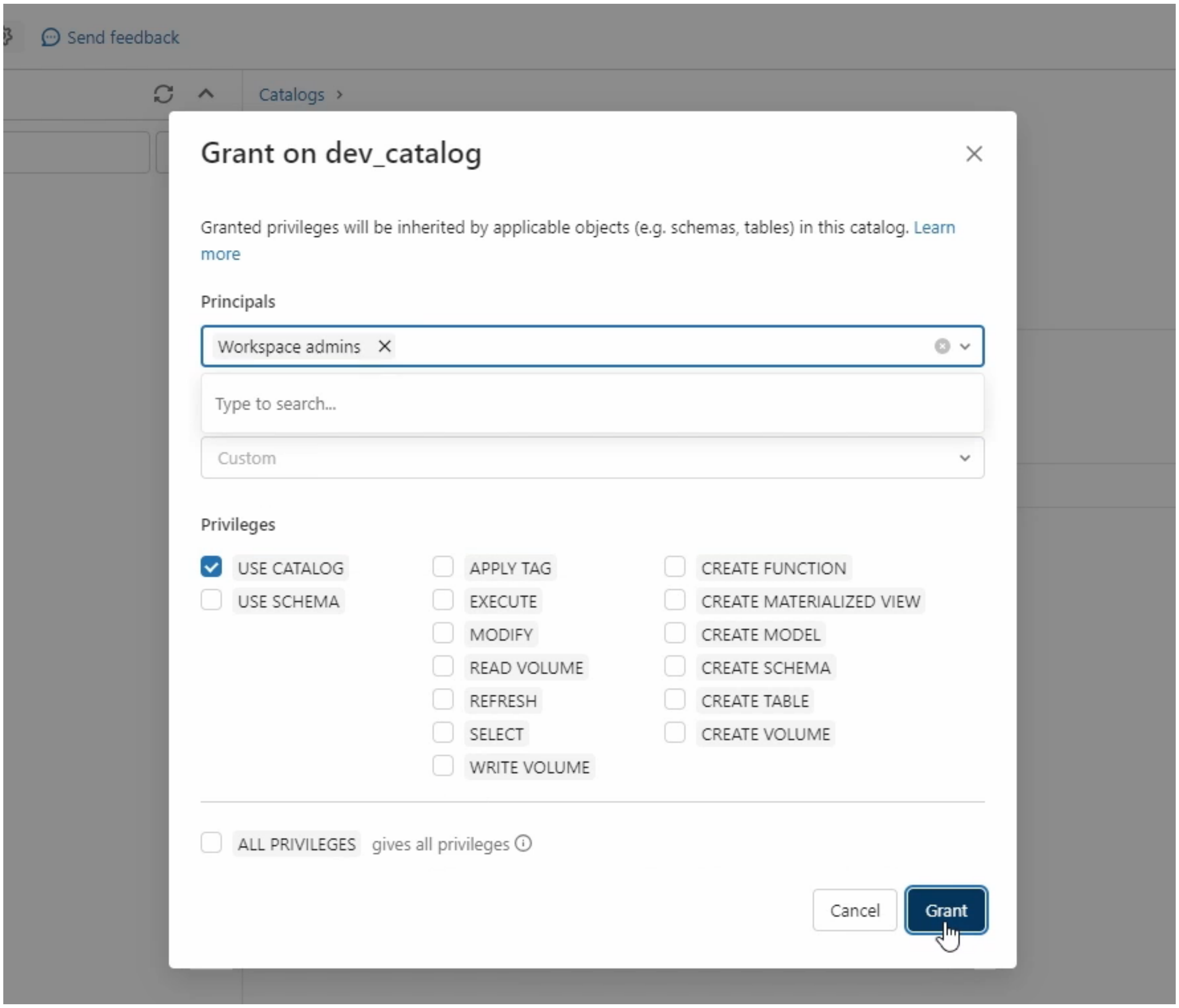
Comment: Add comment

2 schemas

Name	Created at	Owner
default	2023-12-11 17:02:07	shanmukh@shanmukhsattiraju.com
information_schema	2023-12-11 17:02:07	System user

Initially a catalog will be created with **default schema** and **information_schema,** if a table is created without specifying the schema, it will be created under default schema. Information_schema has the metadata about all the tables in the catalog.

Privileges on Unity catalog



Grant on dev_catalog

Granted privileges will be inherited by applicable objects (e.g. schemas, tables) in this catalog. Learn more

Principals

Workspace admins ✕

Type to search...

Custom

Privileges

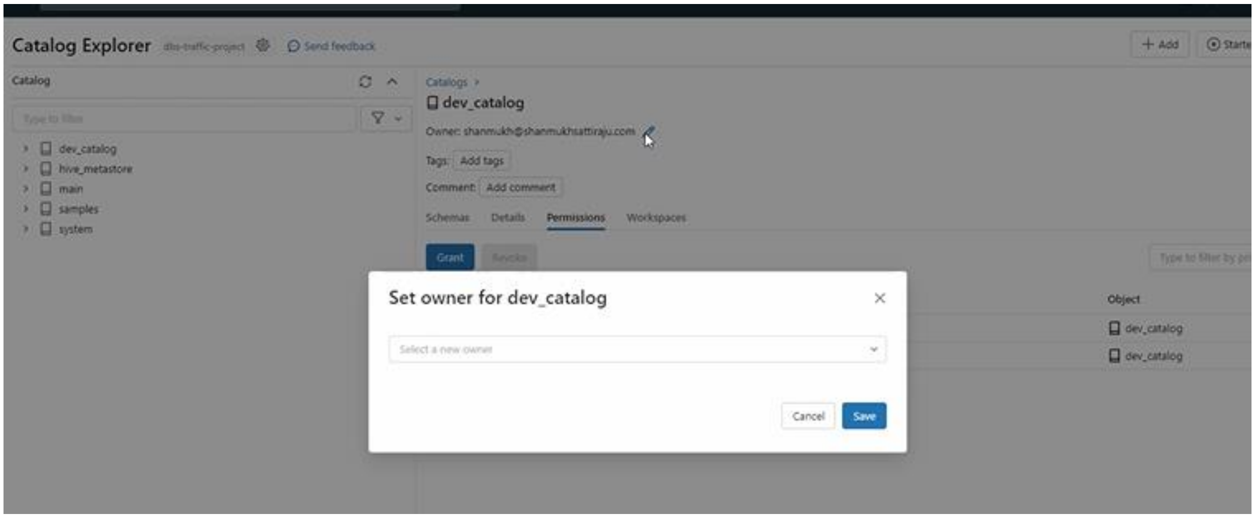
☑ USE CATALOG
☐ USE SCHEMA

☐ APPLY TAG
☐ EXECUTE
☐ MODIFY
☐ READ VOLUME
☐ REFRESH
☐ SELECT
☐ WRITE VOLUME

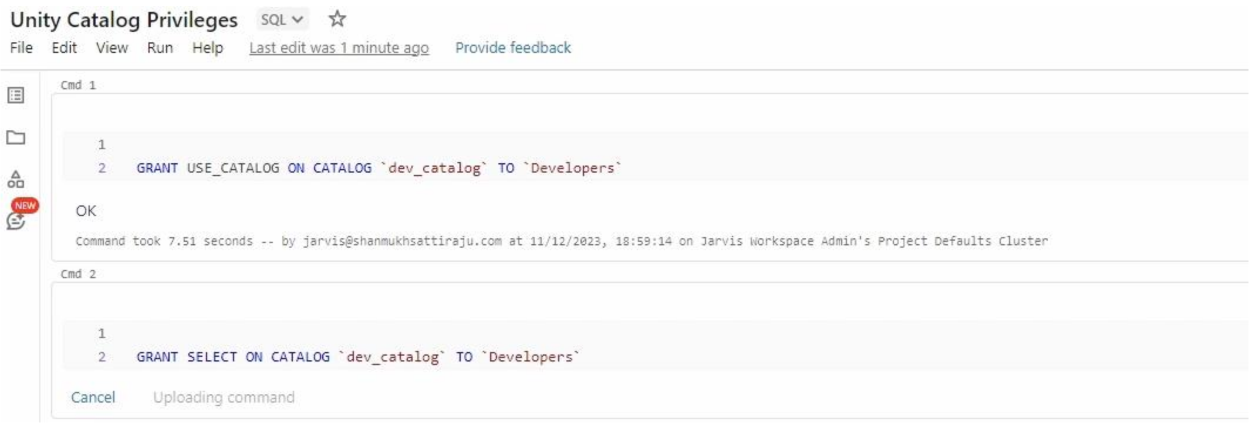
☐ CREATE FUNCTION
☐ CREATE MATERIALIZED VIEW
☐ CREATE MODEL
☐ CREATE SCHEMA
☐ CREATE TABLE
☐ CREATE VOLUME

☐ ALL PRIVILEGES gives all privileges ⓘ

From the workspace Admin account privileges can be granted against the unity catalog



It is a good practice to change the owner of the catalog from individual user to group.



Cmd 1

```
GRANT USE_CATALOG ON CATALOG `dev_catalog` TO `Developers`
```

OK

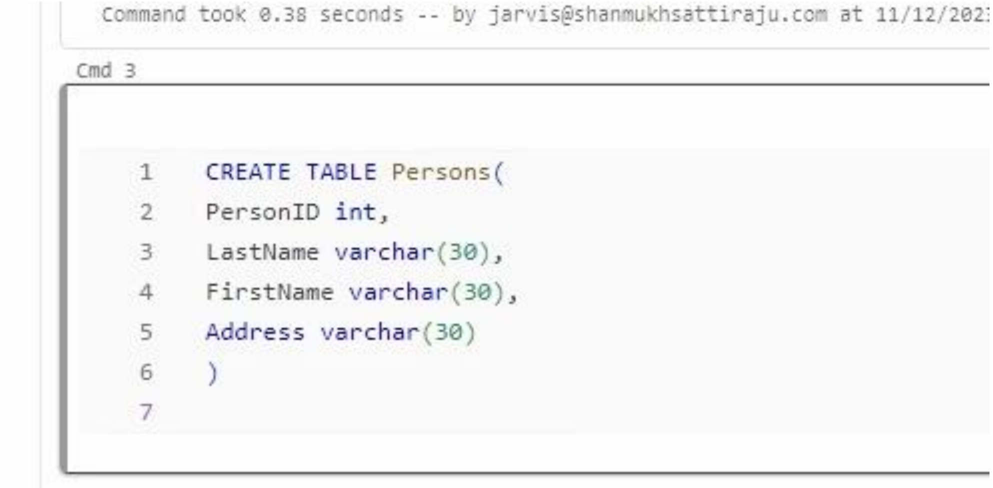
```
Command took 7.51 seconds -- by jarvis@shanmukhsattiraju.com at 11/12/2023, 18:59:14 on Jarvis Workspace Admin's Project Defaults Cluster
```

Cmd 2

```
GRANT SELECT ON CATALOG `dev_catalog` TO `Developers`
```

Cancel   Uploading command

Using SQL commands to grant permissions for an object to a user or group



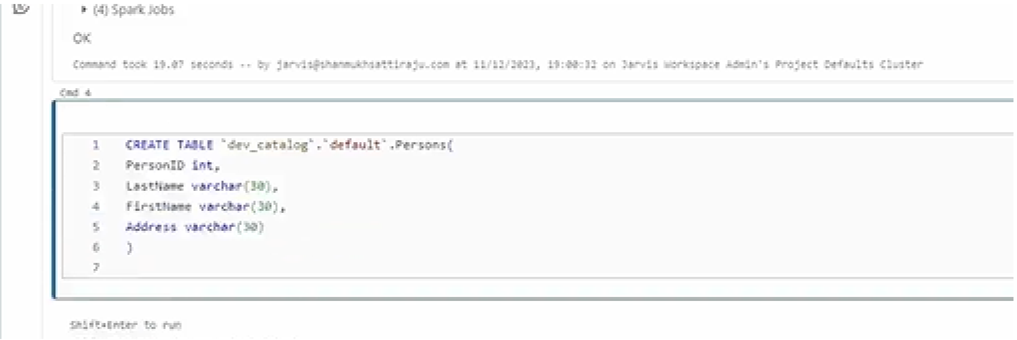
```
Command took 0.38 seconds -- by jarvis@shanmukhsattiraju.com at 11/12/202
```

Cmd 3

```sql
CREATE TABLE Persons(
PersonID int,
LastName varchar(30),
FirstName varchar(30),
Address varchar(30)
)

```

If we create a table in a unity catalog enabled workspace with specifying the 3 level namespace, table will be created in the legacy hive metastore default schema. (We cannot have any data lineage or delta sharing features in the tables created in the hive metastore)



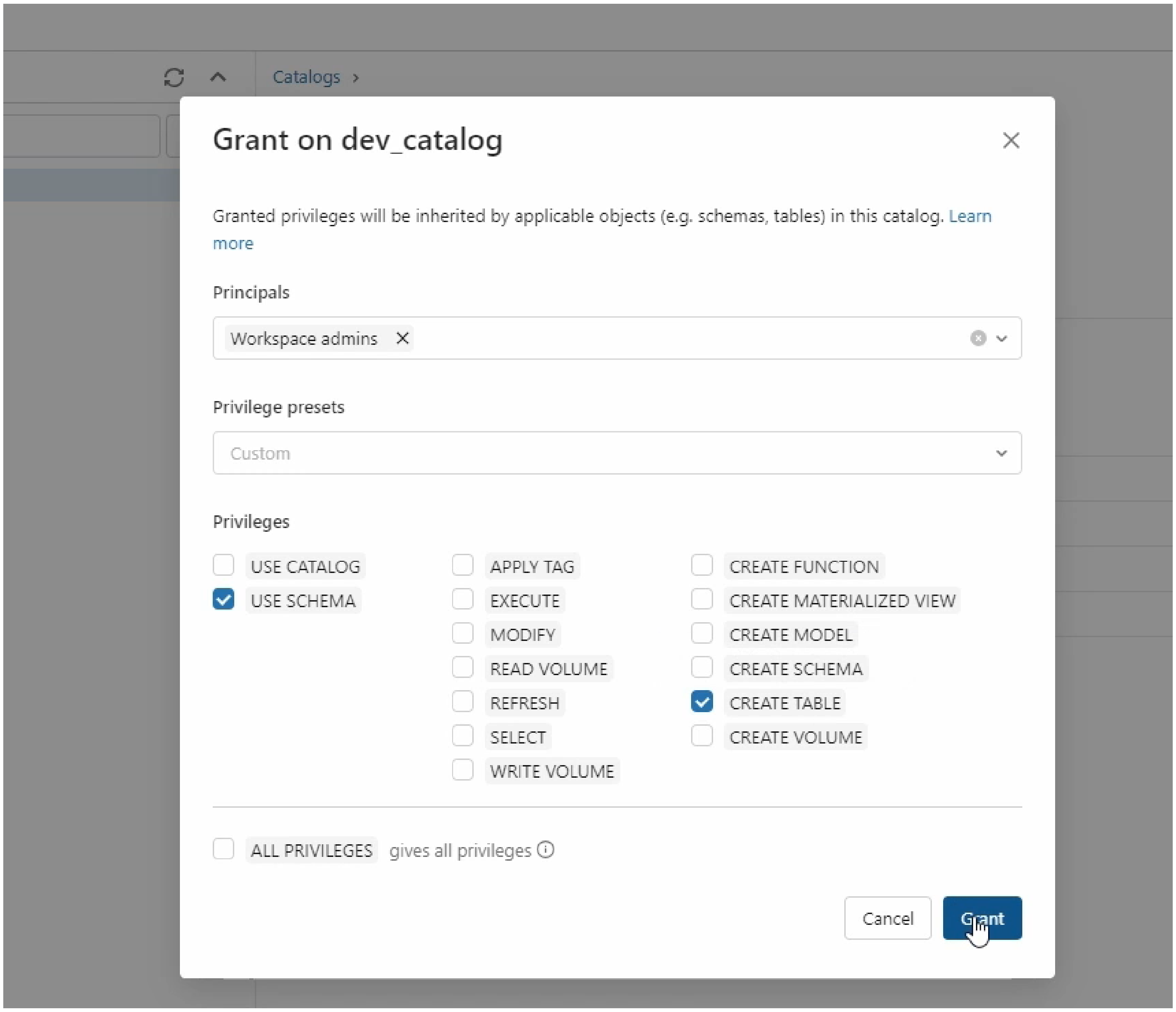
```
OK
Command took 19.07 seconds -- by jarvis@shanmukhsattiraju.com at 11/12/2023, 19:00:32 on Jarvis Workspace Admin's Project Defaults Cluster
```

Cmd 4

```sql
CREATE TABLE `dev_catalog`.`default`.Persons(
PersonID int,
LastName varchar(30),
FirstName varchar(30),
Address varchar(30)
)

```

A three level namespace should be used to create a tables in the unity catalog enabled hive metastore.



Grant on dev_catalog

Granted privileges will be inherited by applicable objects (e.g. schemas, tables) in this catalog. Learn more

Principals

Workspace admins ✕

Privilege presets

Custom

Privileges

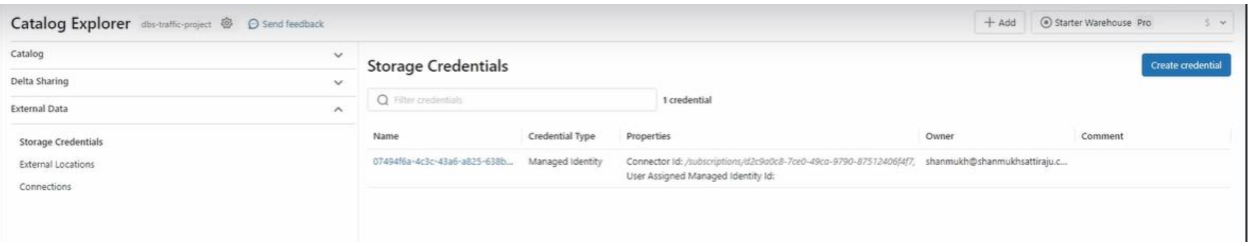
- ☐ USE CATALOG
- ☑ USE SCHEMA
- ☐ APPLY TAG
- ☐ EXECUTE
- ☐ MODIFY
- ☐ READ VOLUME
- ☐ REFRESH
- ☐ SELECT
- ☐ WRITE VOLUME
- ☐ CREATE FUNCTION
- ☐ CREATE MATERIALIZED VIEW
- ☐ CREATE MODEL
- ☐ CREATE SCHEMA
- ☑ CREATE TABLE
- ☐ CREATE VOLUME

☐ ALL PRIVILEGES gives all privileges ⓘ

Create table and Use schema permissions are required to create a table in the metastore

Storage credentials and external location

Storage credentials and external location are managed at the catalog level



Catalog Explorer dbs-traffic-project

Storage Credentials

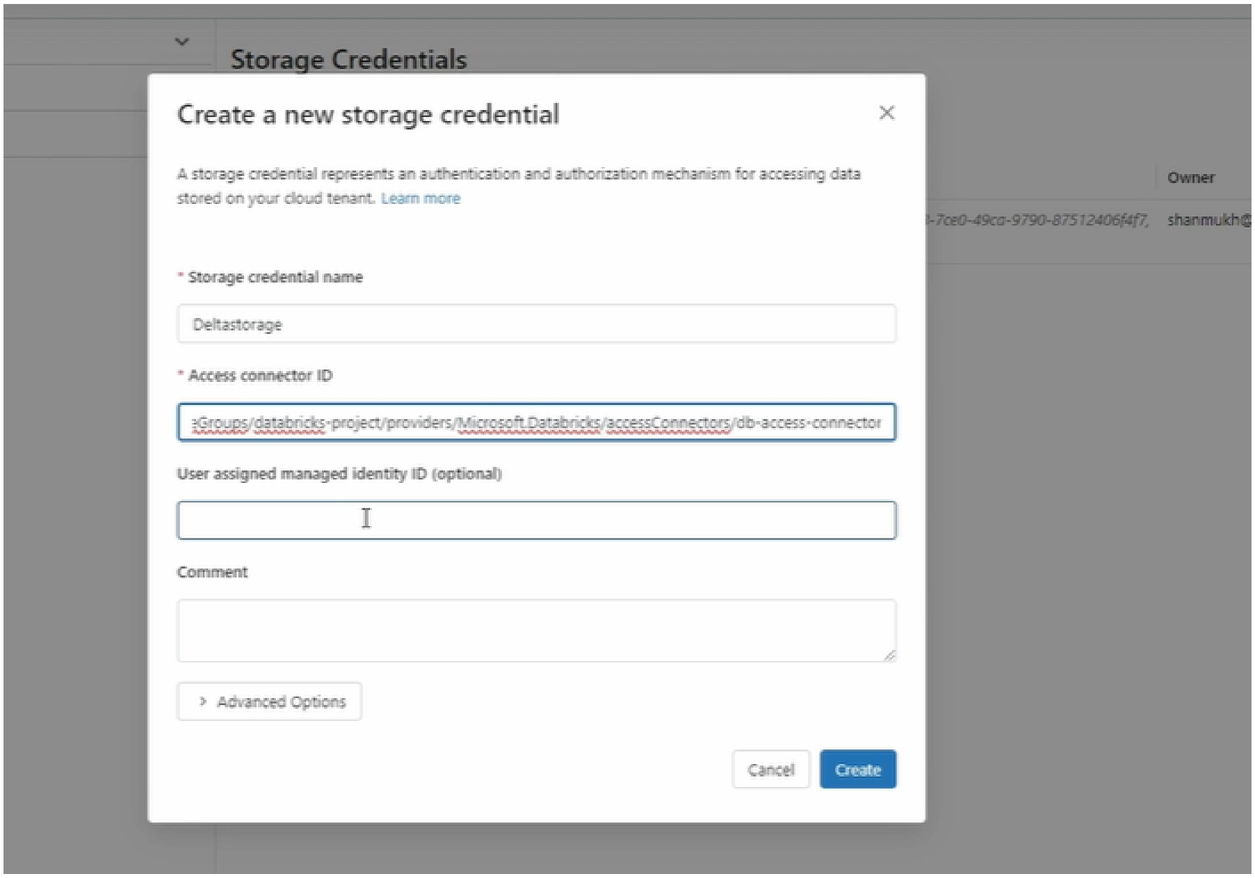
1 credential

Name	Credential Type	Properties	Owner	Comment
07494f6a-4c3c-43a6-a825-638b...	Managed Identity	Connector Id: /subscriptions/d2c9a0c8-7ce0-49ca-9790-87512406f4f7, User Assigned Managed Identity Id:	shanmukh@shanmukhsattiraju.c...	

Firstly, in the storage credentials only one Managed Identity will exist which is while granting access to the storage account for creating managed tables.

To access any external account and its container, to the external storage account, Storage Blob contributor access needs to be added.

A new storage credential can be created using the UI, Access connector can be fetched from the databricks access connector resource.



Create a new storage credential

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

* Storage credential name

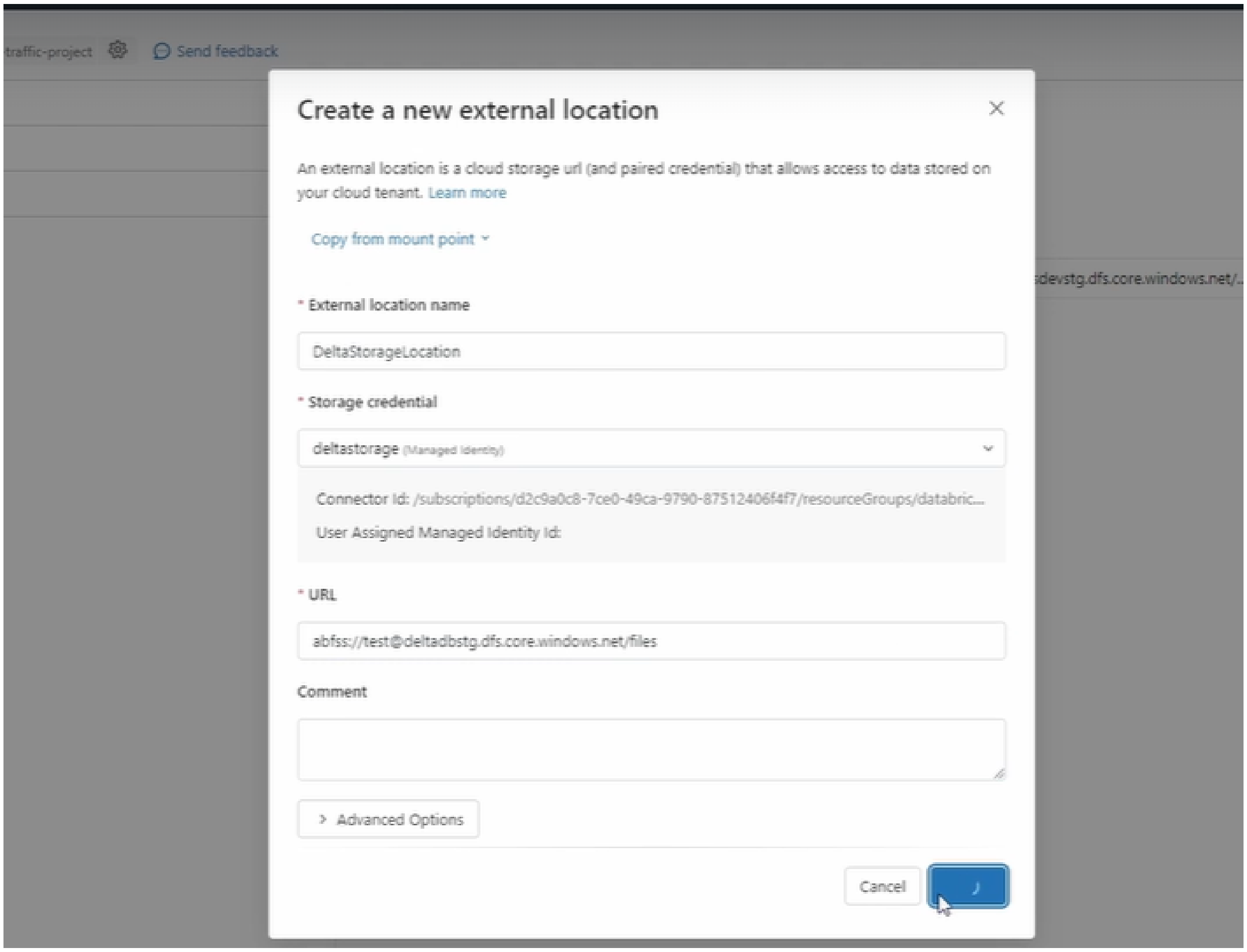
Deltastorage

* Access connector ID

:Groups/databricks-project/providers/Microsoft.Databricks/accessConnectors/db-access-connector

User assigned managed identity ID (optional)

Comment



Create a new external location

An external location is a cloud storage url (and paired credential) that allows access to data stored on your cloud tenant. Learn more

Copy from mount point

* External location name

DeltaStorageLocation

* Storage credential

deltastorage (Managed Identity)

Connector Id: /subscriptions/d2c9a0c8-7ce0-49ca-9790-87512406f4f7/resourceGroups/databric...

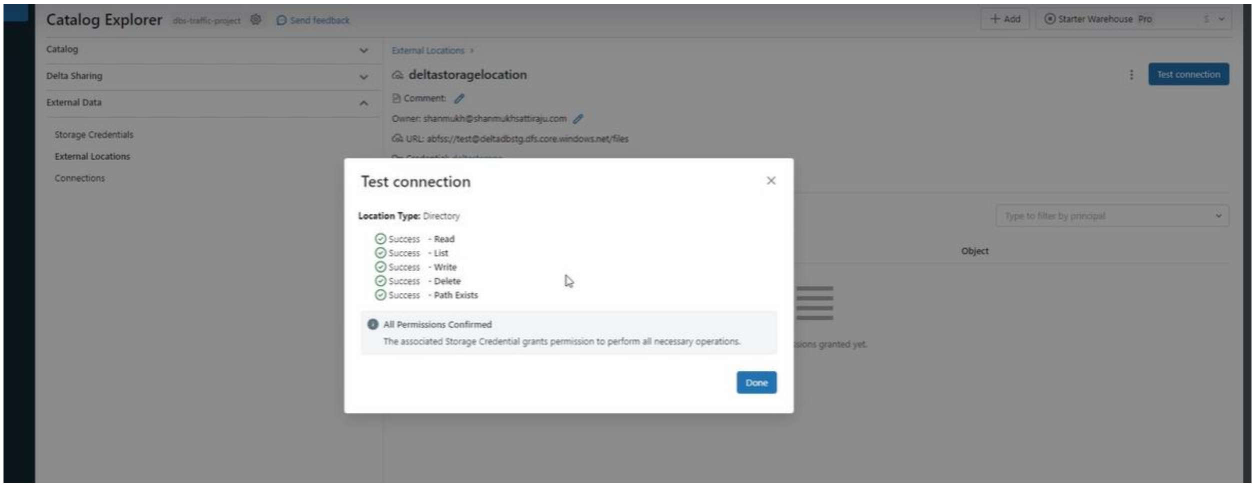
User Assigned Managed Identity Id:

* URL

abfss://test@deltadbstg.dfs.core.windows.net/files

Comment

an external location can be created, selecting the storage credential and providing the storage account URL.



deltastoragelocation

Comment:

Owner: shanmukh@shanmukhsattiraju.com

URL: abfss://test@deltadbstg.dfs.core.windows.net/files

Test connection

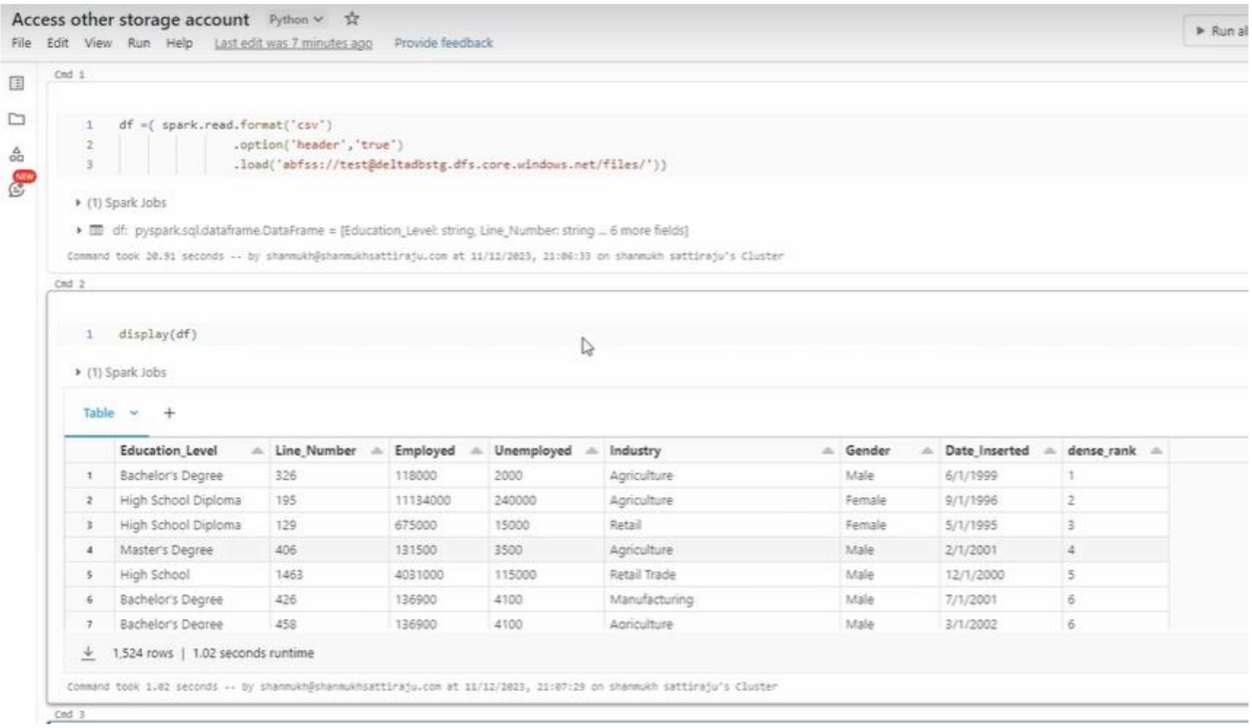
Location Type: Directory

- Success - Read
- Success - List
- Success - Write
- Success - Delete
- Success - Path Exists

All Permissions Confirmed

The associated Storage Credential grants permission to perform all necessary operations.

A test connection can be performed after creating the storage and external credential.



```python
df =( spark.read.format('csv')
            .option('header','true')
            .load('abfss://test@deltadbstg.dfs.core.windows.net/files/'))
```

```
(1) Spark Jobs
df:  pyspark.sql.dataframe.DataFrame = [Education_Level: string, Line_Number: string ... 6 more fields]
Command took 20.91 seconds -- by shanmukh@shanmukhsattiraju.com at 11/12/2023, 21:06:33 on shanmukh sattiraju's Cluster
```

```python
display(df)
```

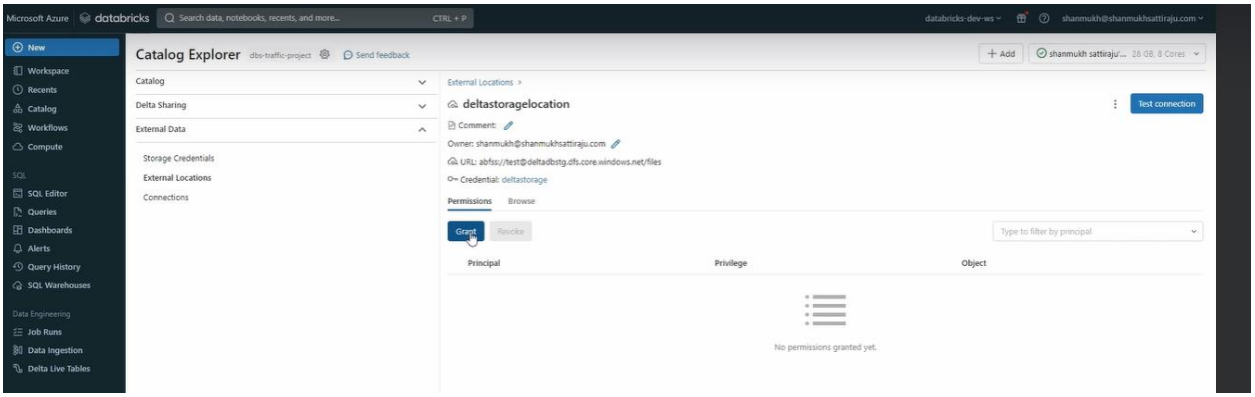
```
(1) Spark Jobs
```

	Education_Level	Line_Number	Employed	Unemployed	Industry	Gender	Date_Inserted	dense_rank
1	Bachelor's Degree	326	118000	2000	Agriculture	Male	6/1/1999	1
2	High School Diploma	195	11134000	240000	Agriculture	Female	9/1/1996	2
3	High School Diploma	129	675000	15000	Retail	Female	5/1/1995	3
4	Master's Degree	406	131500	3500	Agriculture	Male	2/1/2001	4
5	High School	1463	4031000	115000	Retail Trade	Male	12/1/2000	5
6	Bachelor's Degree	426	136900	4100	Manufacturing	Male	7/1/2001	6
7	Bachelor's Degree	458	136900	4100	Agriculture	Male	3/1/2002	6

```
1,524 rows | 1.02 seconds runtime
Command took 1.02 seconds -- by shanmukh@shanmukhsattiraju.com at 11/12/2023, 21:07:29 on shanmukh sattiraju's Cluster
```

Reading the data after creating the external storage access credentials

To read the data from different container in the same storage account a separate external storage credential is required.



deltastoragelocation

Comment:

Owner: shanmukh@shanmukhsattiraju.com

URL: abfss://test@deltadbstg.dfs.core.windows.net/files

Credential: deltastorage

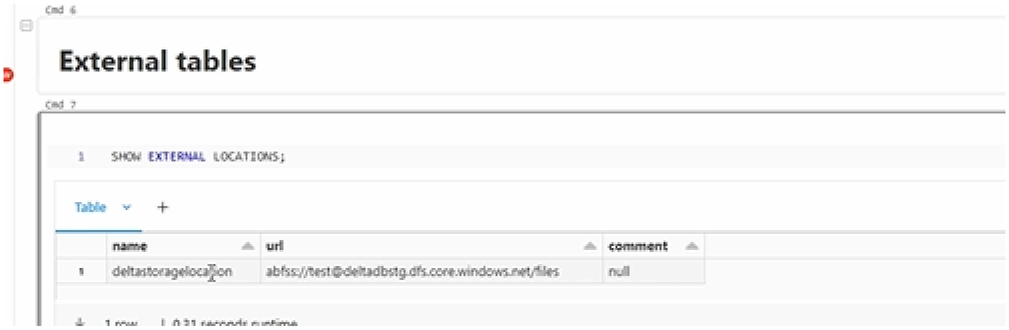
No permissions granted yet.

Access can be fine tuned for the external containers

Specific permissions can be granted to either the groups or users.

Managed and external tables in Unity catalog

Managed Tables	External Tables
Tables can be created **without** specifying the location.	Location **need to be specified**
Dropping the table will removes the metadata as well as actual data, however in unity catalog underlying data will be present for **30 days**	Dropping the table will deletes only the meta data, actual data will still exist.
	External location and managed storage credential needs to be place



Cmd 6

External tables

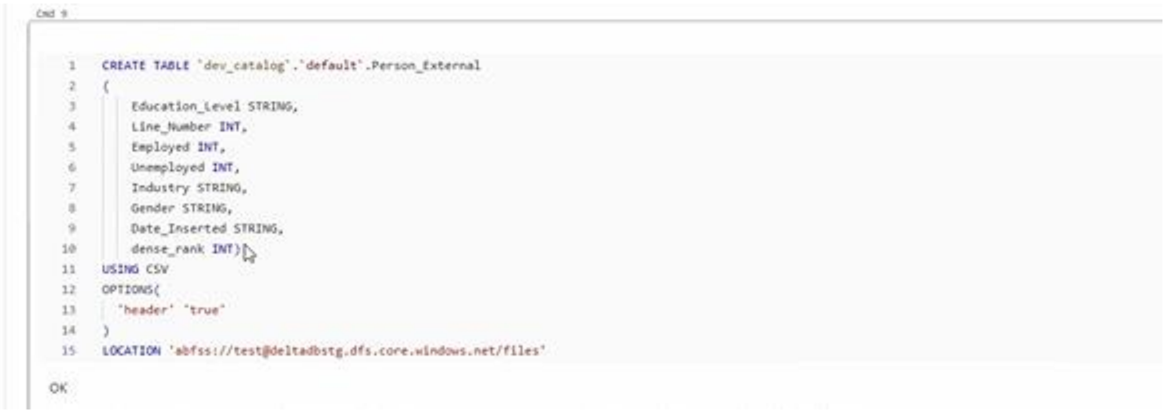
Cmd 7

```
SHOW EXTERNAL LOCATIONS;
```

	name	url	comment
1	deltastoragelocation	abfss://test@deltadbstg.dfs.core.windows.net/files	null

1 row | 0.21 seconds runtime

external locations can be listed



Cmd 9

```
CREATE TABLE `dev_catalog`.`default`.Person_External
(
    Education_Level STRING,
    Line_Number INT,
    Employed INT,
    Unemployed INT,
    Industry STRING,
    Gender STRING,
    Date_Inserted STRING,
    dense_rank INT)
USING CSV
OPTIONS(
  'header' 'true'
)
LOCATION 'abfss://test@deltadbstg.dfs.core.windows.net/files'
```

```
OK
```

Creating an external table, LOCATION must be specified