

Lab: Integrate Databricks Unity Catalog with OneLake

Objective:

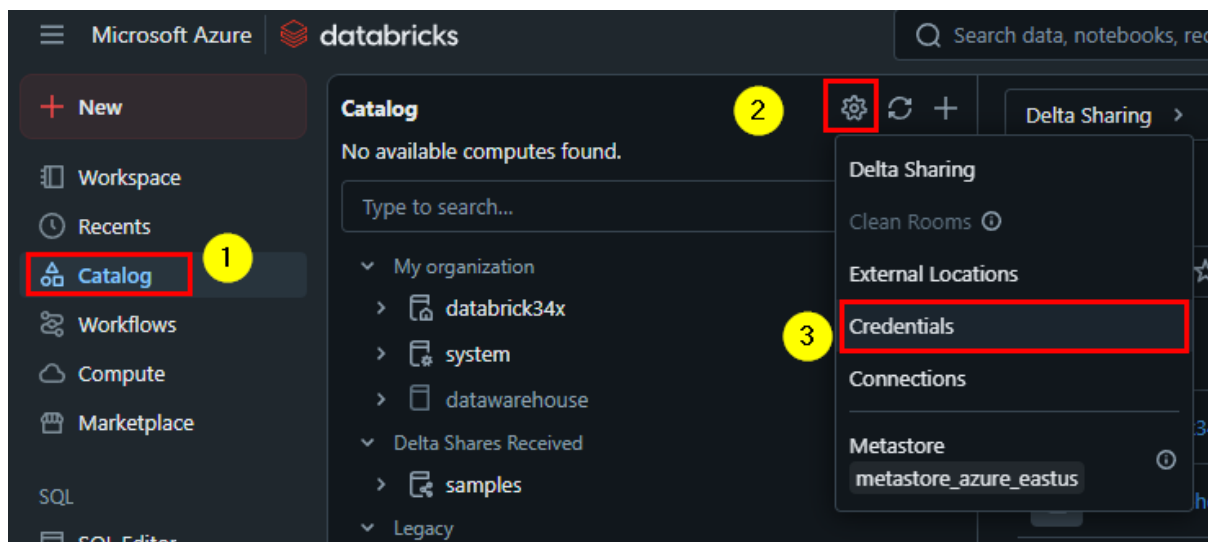
Understand the concept of integrating OneLake with the Unity Catalog.

Tasks:

1. Create a unity catalog in Databricks Workspace
2. Create external credentials and location
3. Set up your Cloud storage connection
4. Create a OneLake Shortcut

Task 1: Create credentials and external location

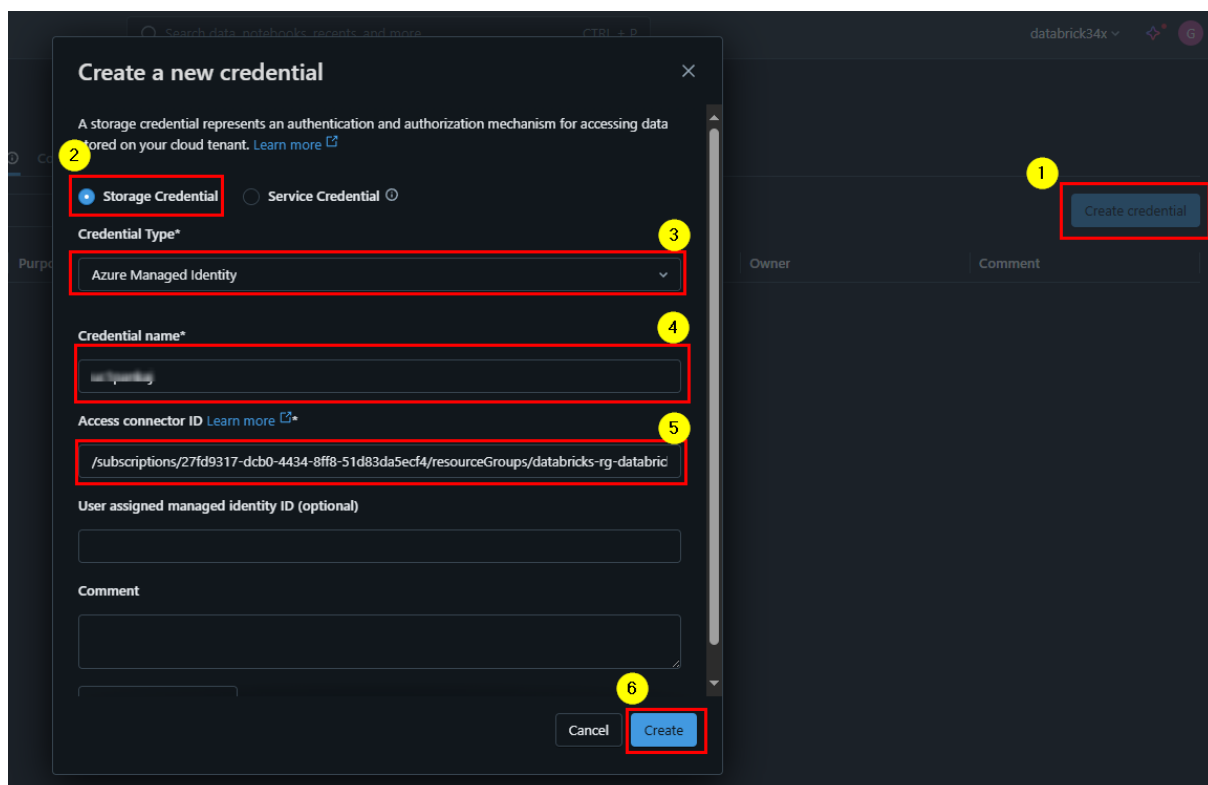
1. Open the Databricks workspace (<https://adb-3426738885164031.11.azuredatabricks.net/>) in a browser window.
2. Navigate to catalog click on the gear icon and select Credentials



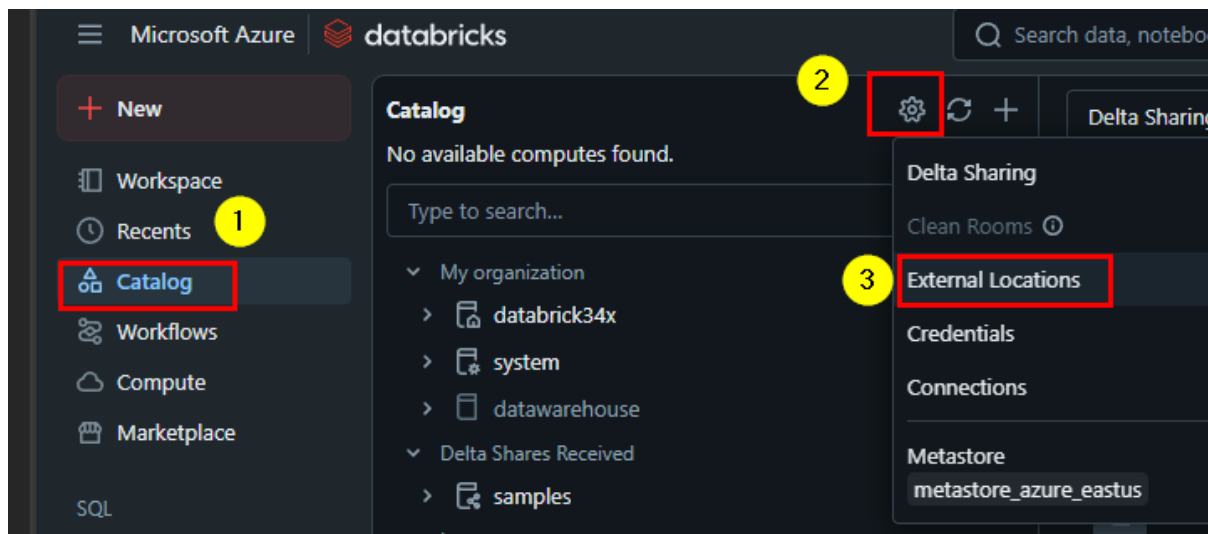
3. Click **Create credential** and provide the following details as shown:

Access connector ID:

</subscriptions/27fd9317-dcb0-4434-8ff8-51d83da5ecf4/resourceGroups/RG-Genpack/providers/Microsoft.Databricks/accessConnectors/databricks-access-connector>

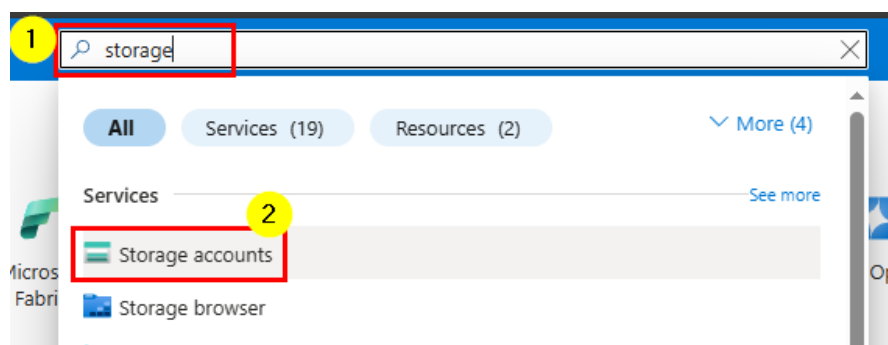


4. Create an external location

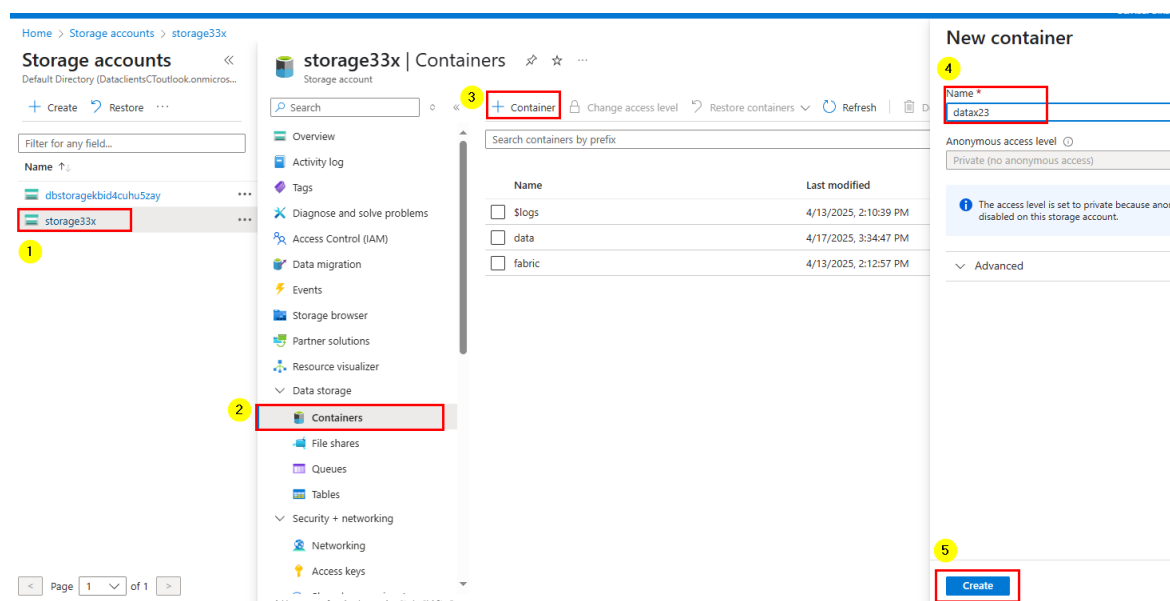


5. Create a container in Azure Data Lake storage

a. Search for Azure Storage:



b. Select **storage33x**, create a new container, and give a unique name.



- c. Construct your storage URL by replacing your container name

URL: `abfss://<replace-with-your-containerName>@storage34x.dfs.core.windows.net/`

Create a new external location

An external location allows you to access your data stored in cloud storage (e.g. Azure Data Lake Storage). You will need the cloud storage path and a paired credential (e.g. managed identity) which gives access to that path [Learn more](#)

External location name*

URL* ⓘ

Enter the bucket path that you want to use as the external location. Note: This must be an ADLS Gen2 storage account with a hierarchical namespace

Copy from DBFS ▾

Storage credential* [Learn more](#) ⓘ

Provide a storage credential capable of accessing the URL

Connector Id: /subscriptions/27fd9317-dcb0-4434-8ff8-51d83da5ecf4/resourceGroups/RG-Genpact/providers/Microsoft....

User Assigned Managed Identity Id:

Comment

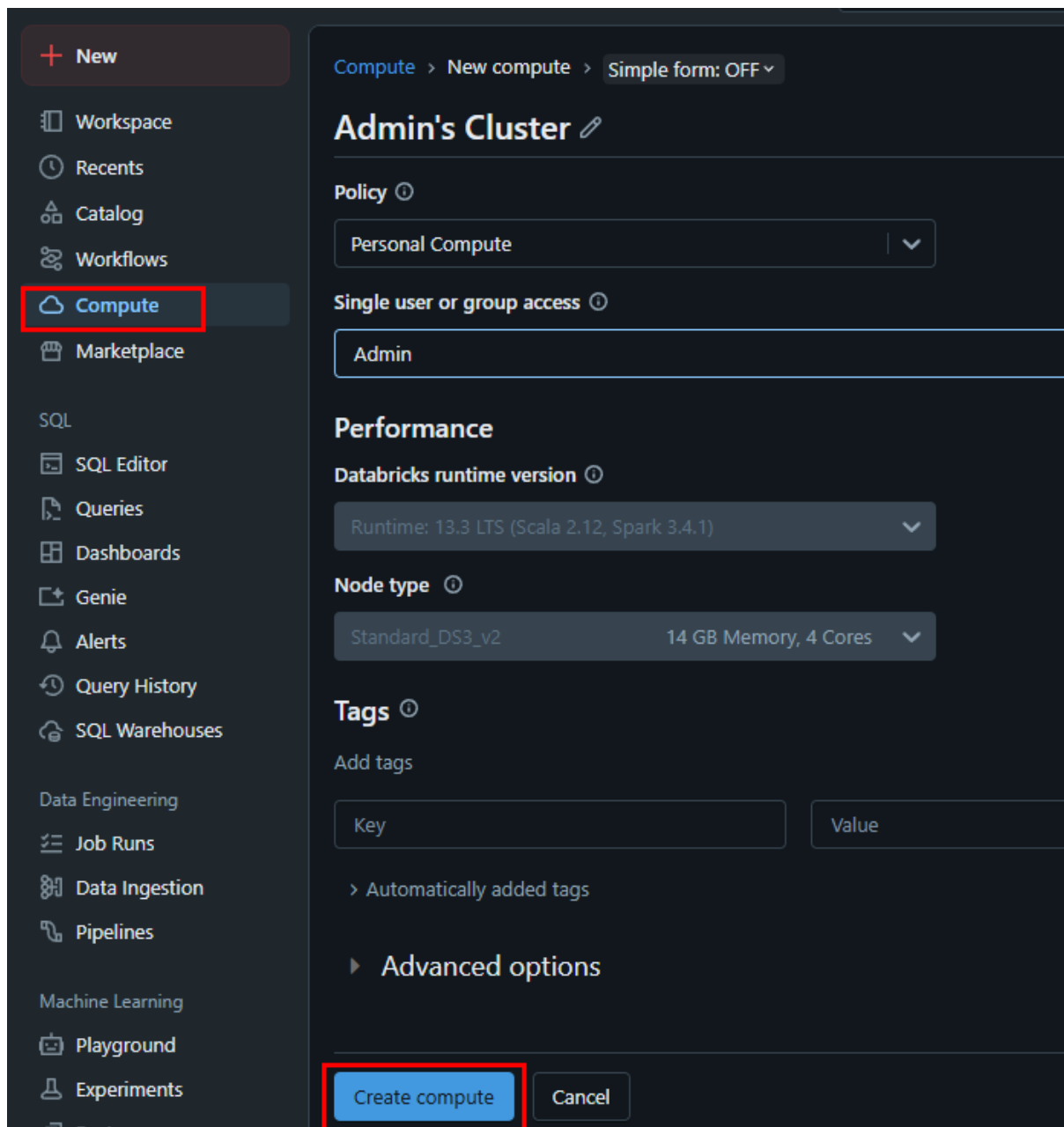
> Advanced Options

Cancel

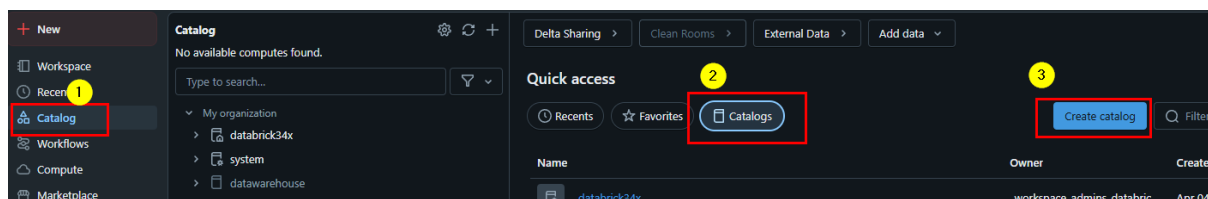
Create

Task 2: Create a unity catalog in Databricks Workspace

1. Create a new compute, navigate to compute and select **Create Compute**.



2. Select Catalog from the left vertical menu, select Catalogs, and click Create Catalog.



3. Enter the catalog name and select the storage location to external location.

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* 1

Type* 2 Standard

Storage location

[Create a new external location](#)

abfss://uc1pankaj@storage34x.dfs.core.windows.net/sales

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

3

Cancel Create

4. Select view catalog

Catalog created!

Catalog has been created with recommended permission settings. Click configure to set up additional permissions, workspace bindings, and to add metadata.

View catalog Configure catalog

5. Create a new notebook in Databricks inside the workspace

- In the first cell enter the following code and create an empty cell and run it, if no cluster is created then create a new one.

```
use catalog <your-catalog-name>;
create schema if not exists sales;
use schema sales;
create table if not exists department
(
  deptcode int,
  deptname string,
  location string
);
```

```
INSERT INTO department VALUES
(10, 'FINANCE', 'EDINBURGH'),
(20, 'SOFTWARE', 'PADDINGTON');
```

```
Select * from department;
```

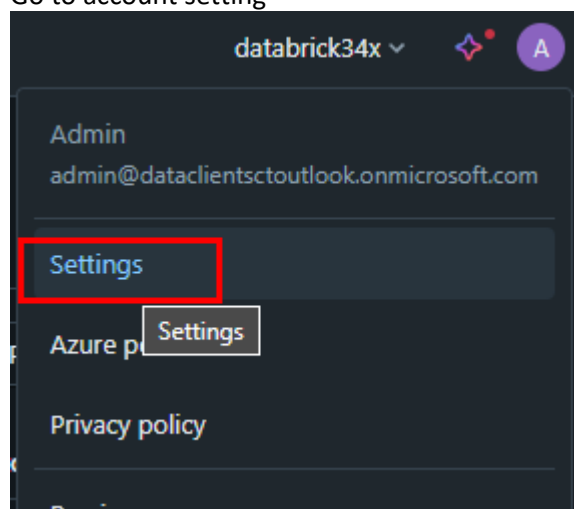
7. Create a new code cell and run the following code to create an external table. Replace your container name:

```
create external table trips_external
using delta
location 'abfss://<your-container-name>@storage34x.dfs.core.windows.net/trip'
as select * from samples.nyctaxi.trips;
```

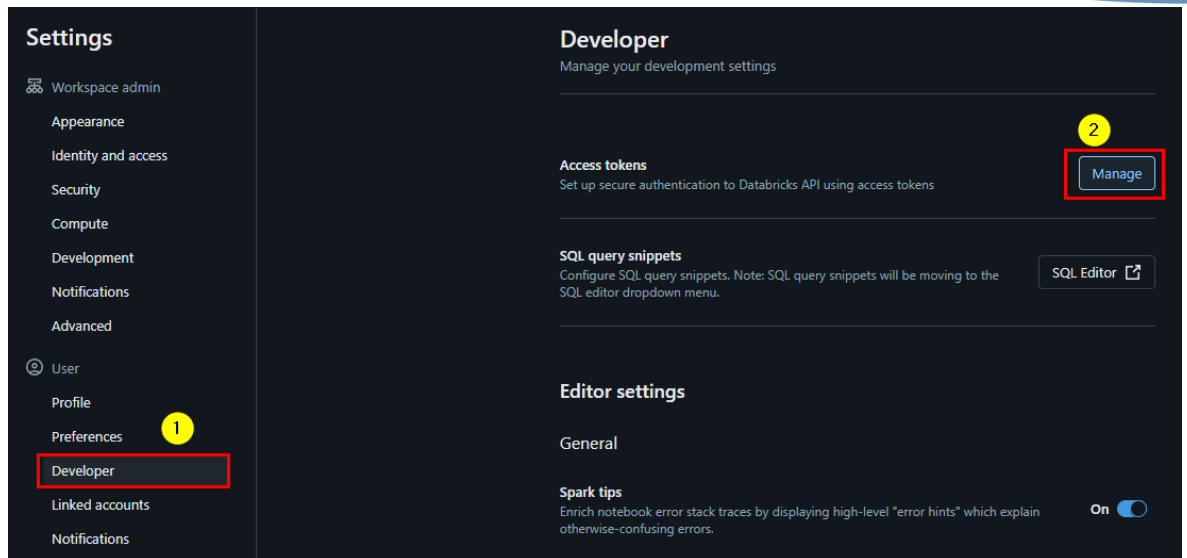
8. Terminate the compute resource.

9. Create a personal access token:

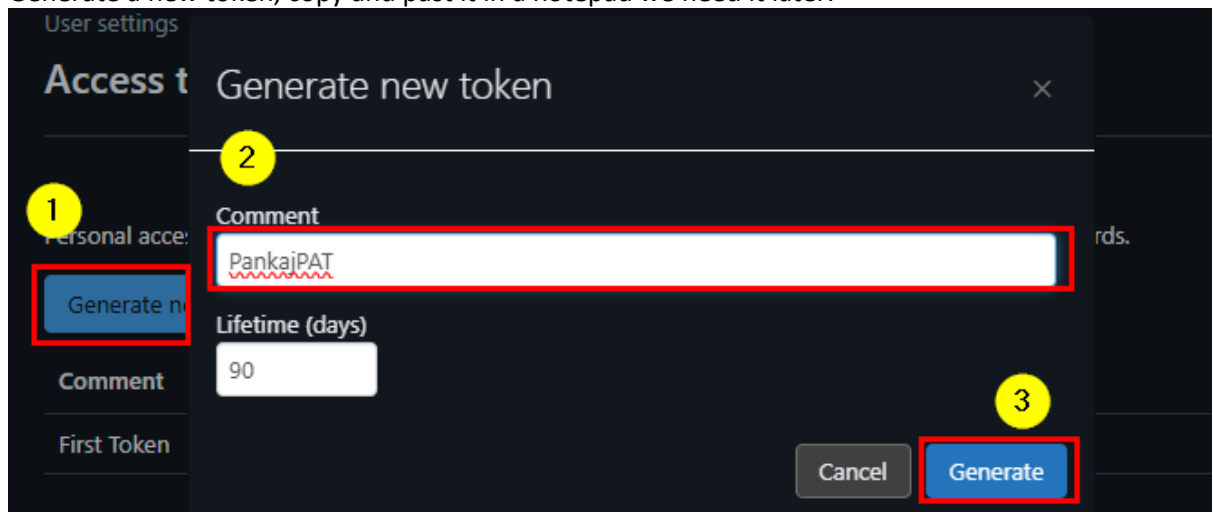
- a. Go to account setting



- b. Select Developer under User, and then Manage for Access token

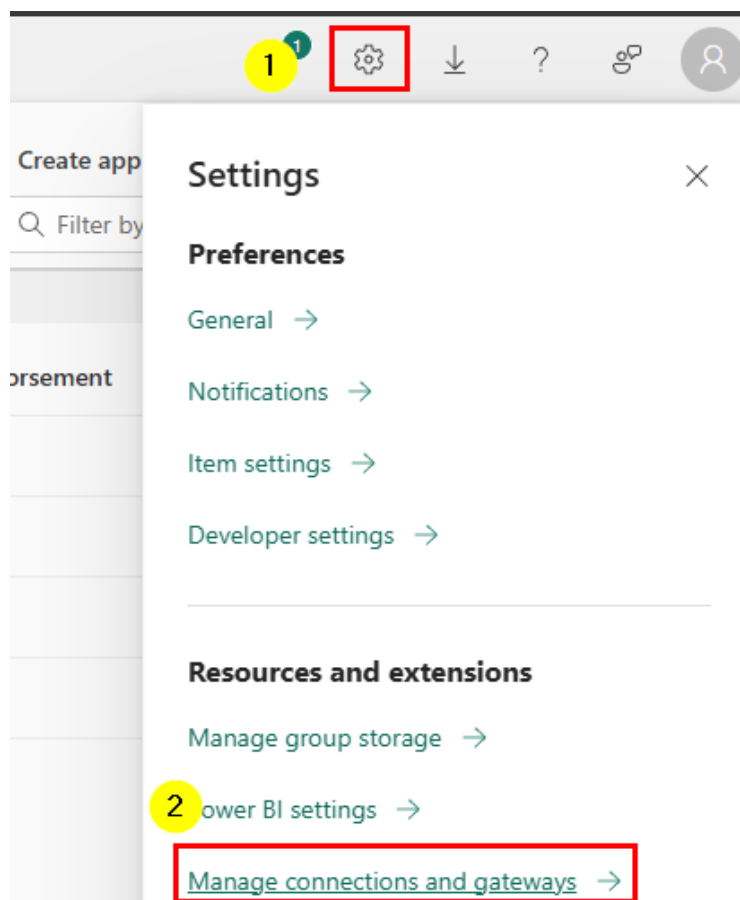


- c. Generate a new token, copy and paste it in a notepad we need it later.



Task 3: Set up your Cloud storage connection in Fabric

1. Switch back to the Microsoft Fabric portal and from setting open connections.



2. Enter the following details:

- a. Connection name: a unique name
- b. Connection type: ADLS Gen 2
- c. Storage URL: <https://storage33x.dfs.core.windows.net/>
- d. Full path: **Your Container Name**
- e. Authentication type: Key
- f. Account key:

Us86NWDEUvmsB4UsUdwBi+BRTu1YSL1HA9IuQNgcNkoShZmab45Tf7Uiz/y9MbdJ5E
RIG7iEx2Jf+ASOf+zMQ==

New connection ×

① Currently, these cloud connections are not supported by Dataflows, Dataflows Gen2, and Datamarts. To create personal cloud connections for these experiences, please use the Dataflows or Datamarts editor in "Get Data".

On-premises

Virtual network

Cloud

1

2

Connection name *

uc2pankaj

Connection type *

Azure Data Lake Storage Gen2 **3**

Server * ① **4**

Ex: https://contosoadlscdm.dfs.core.windows.net

Full path * ① **5**

uc2pankaj

Authentication **6**

Authentication method *

Key

Account key *

.....

- When the connection is created, Open the newly created connection and copy the connection ID:

Settings ×

① Currently, these cloud connections are not supported by Dataflows, Dataflows Gen2, and Datamarts. To create personal cloud connections for these experiences, please use the Dataflows or Datamarts editor in "Get Data".

On-premises

Virtual network

Cloud

Connection name *

uc1pankaj

Connection ID

a8f3bc91-591e-473e-97b3-1d11dedd74f7

Connection type

Azure Data Lake Storage Gen2

Task 4: Create a OneLake Shortcut

1. **Import the sync notebook** to your Fabric workspace. Download [This notebook](#)
2. **Configure the parameters** in the first cell of the notebook to integrate Unity Catalog tables. The Databricks API, authenticated through the PAT token, is utilized for exporting Unity Catalog tables. The following snippet is used to configure the source (Unity Catalog) and destination (OneLake) parameters. Ensure to replace them with your own values.

Databricks workspace

dbx_workspace = "<databricks_workspace_url>"

dbx_token = "<pat_token>"

Unity Catalog

dbx_uc_catalog = "catalog1"

dbx_uc_schemas = ["sales"]

Fabric

fab_workspace_id = "<workspace_id>"

fab_lakehouse_id = "<lakehouse_id>"

fab_shortcut_connection_id = "<connection_id>"

If True, UC table renames and deletes will be considered

fab_consider_dbx_uc_table_changes = True

3. Run all cells of the notebook to start synchronizing Unity Catalog Delta tables to OneLake using shortcuts. Once notebook is completed, shortcuts to Unity Catalog Delta tables are available in the lakehouse, SQL analytics endpoint, and semantic model.