

OneLake in Microsoft Fabric documentation

OneLake is a single, unified, logical data lake for the whole organization. OneLake comes automatically with every Microsoft Fabric tenant with no infrastructure to manage.

About OneLake

OVERVIEW

[What is OneLake?](#)

[OneLake security](#)

[OneLake access and APIs](#)

GET STARTED

[Create a lakehouse with OneLake](#)

[OneLake file explorer](#)

OneLake shortcuts

CONCEPT

[What are shortcuts?](#)

GET STARTED

[Create a shortcut](#)

HOW-TO GUIDE

[Access shortcuts](#)

OneLake and Azure integration

HOW-TO GUIDE

[Integrate OneLake with Azure Databricks](#)

[Integrate OneLake with Azure HDInsight](#)

[Integrate OneLake with Azure Storage Explorer](#)

[Integrate OneLake with Azure Synapse Analytics](#)

OneLake, the OneDrive for data

Article • 05/23/2023

OneLake is a single, unified, logical data lake for the whole organization. Like OneDrive, OneLake comes automatically with every Microsoft Fabric tenant and is designed to be the single place for all your analytics data. OneLake brings customers:

- **One data lake for the entire organization**
- **One copy of data** for use with multiple analytical engines

Important

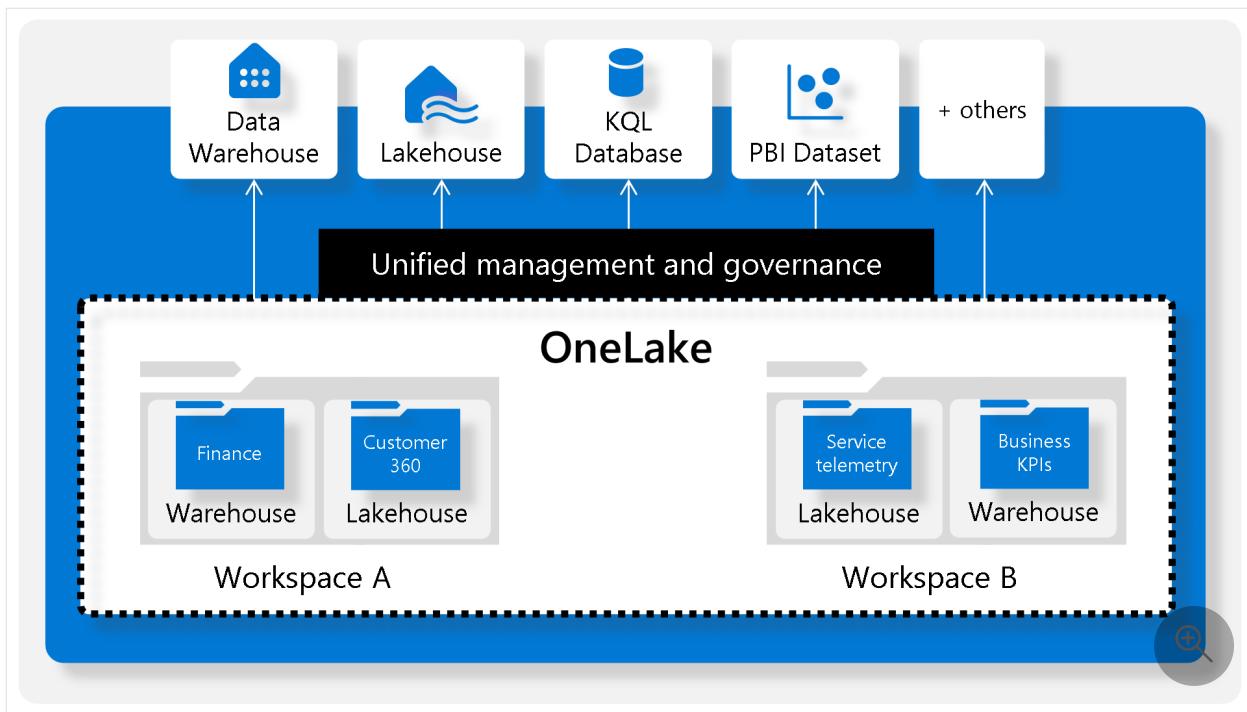
Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

One data lake for the entire organization

Prior to OneLake, it was easier for customers to create multiple lakes for different business groups rather than collaborating on a single lake, even with the extra overhead of managing multiple resources. OneLake focuses on removing these challenges by improving collaboration. Every customer tenant has exactly one OneLake. There can never be more than one and if you have Fabric, there can never be zero. OneLake is provisioned automatically with every Fabric tenant with no extra resources to set up or manage.

Governed by default with distributed ownership for collaboration

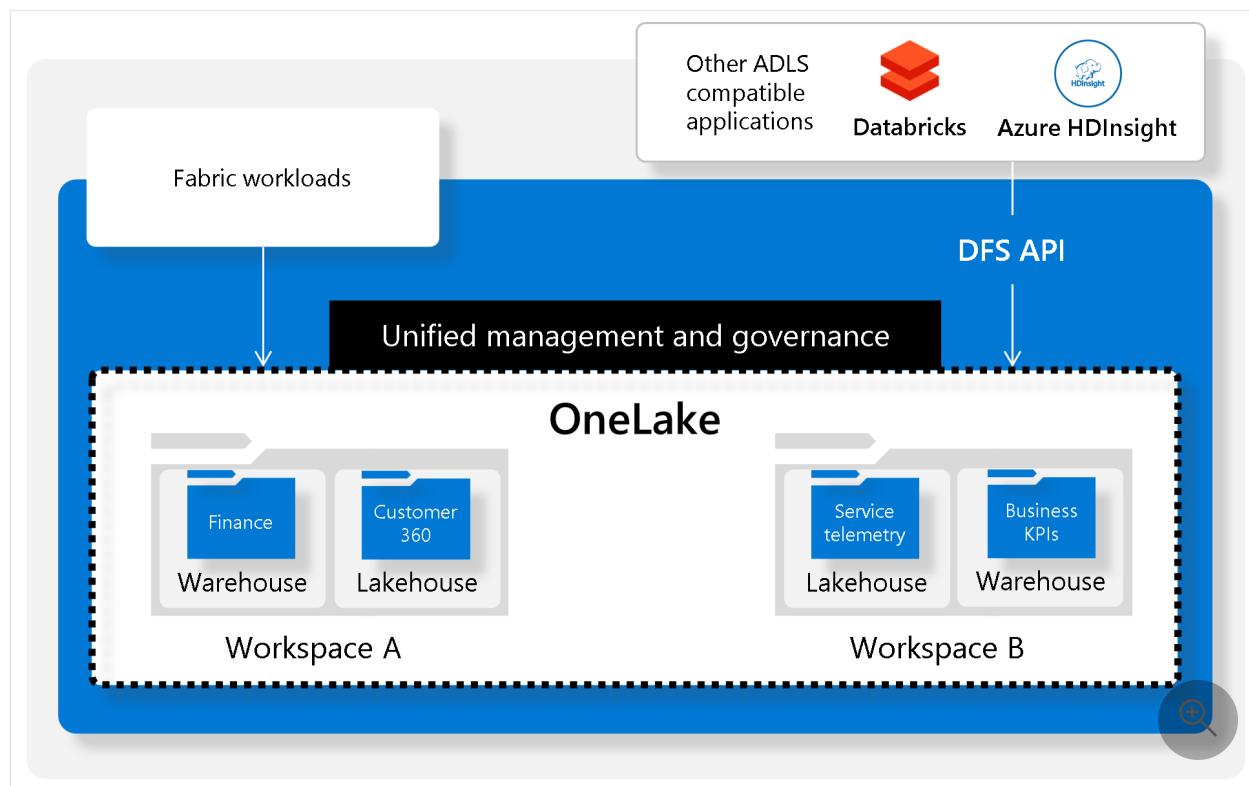
The concept of a tenant is a unique benefit of a SaaS service. Knowing where a customer's organization begins and ends, provides a natural governance and compliance boundary, which is ultimately under the control of a tenant admin. Any data that lands in OneLake is governed by default. While all data is within the boundaries set by the tenant admin, it's important that this admin doesn't become a central gatekeeper preventing other parts of the organization from contributing to OneLake. Within a tenant, you can create any number of workspaces. Workspaces enable different parts of the organization to distribute ownership and access policies. Each workspace is part of a capacity that is tied to a specific region and is billed separately.



Within a workspace, you can create data items and all data in OneLake is accessed through data items. Similar to how Office stores Word, Excel, and PowerPoint files in OneDrive, Fabric stores lakehouses, warehouses, and other items in OneLake. Items can give tailored experiences for each persona such the Spark developer experience in a lakehouse. For more information on how to get started using OneLake, see [Creating a lakehouse with OneLake](#).

Open at every level

OneLake is open at every level. Built on top of Azure Data Lake Storage Gen2, OneLake can support any type of file, structured or unstructured. All Fabric data items like data warehouses and lakehouses store their data automatically in OneLake in delta parquet format. This means when a data engineer loads data into a lakehouse using Spark and a SQL developer in a fully transactional data warehouse uses T-SQL to load data, everyone is still contributing to building the same data lake. All tabular data is stored in OneLake in delta parquet format. OneLake supports the same ADLS Gen2 APIs and SDKs to be compatible with existing ADLS Gen2 applications including Azure Databricks. Data in OneLake can be addressed as if it were one big ADLS storage account for the entire organization. Every workspace appears as a container within that storage account. Different data items appear as folders under those containers.



For more information on APIs and endpoints, see [OneLake access and APIs](#). For examples of OneLake integrations with Azure, see [Azure Synapse Analytics](#), [Azure storage explorer](#), [Azure Databricks](#), and [Azure HDInsight](#) articles.

OneLake file explorer for Windows

OneLake is the OneDrive for data. Just like OneDrive, OneLake data can be easily explored from Windows using the OneLake file explorer for Windows. Directly in Windows, you can navigate all your workspaces, data items, easily uploading, downloading or modifying files just like you can do in office. The OneLake file explorer simplifies data lakes putting them into the hands of even nontechnical business users. For more information, see [OneLake file explorer](#).

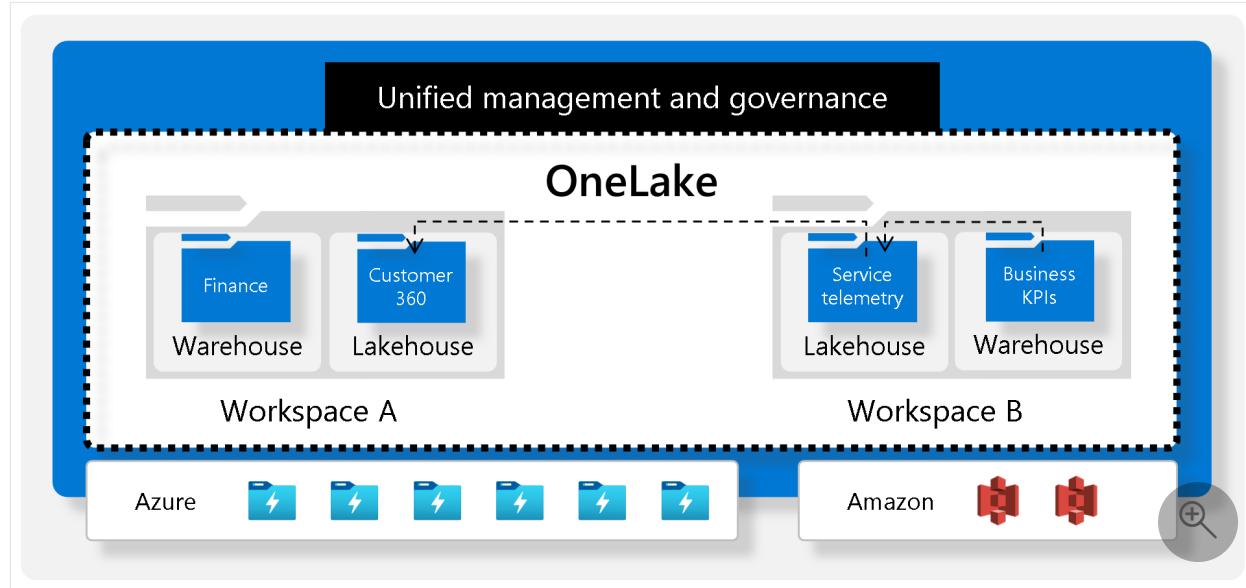
One copy of data

OneLake aims to give you the most value possible out of a single copy of data without data movement or duplication. You'll no longer need to copy data just to use it with another engine or to break down silos so that data can be analyzed with other data.

Shortcuts let you connect data across business domains without data movement

Shortcuts allow your organization to easily share data between users and applications without having to move and duplicate information unnecessarily. When teams work

independently in separate workspaces, shortcuts enable you to combine data across different business groups and domains into a virtual data product to fit a user's specific needs. A shortcut is a reference to data stored in other file locations. These file locations can be within the same workspace or across different workspaces, within OneLake or external to OneLake in ADLS or S3. No matter the location, the reference makes it appear as though the files and folders are stored locally.



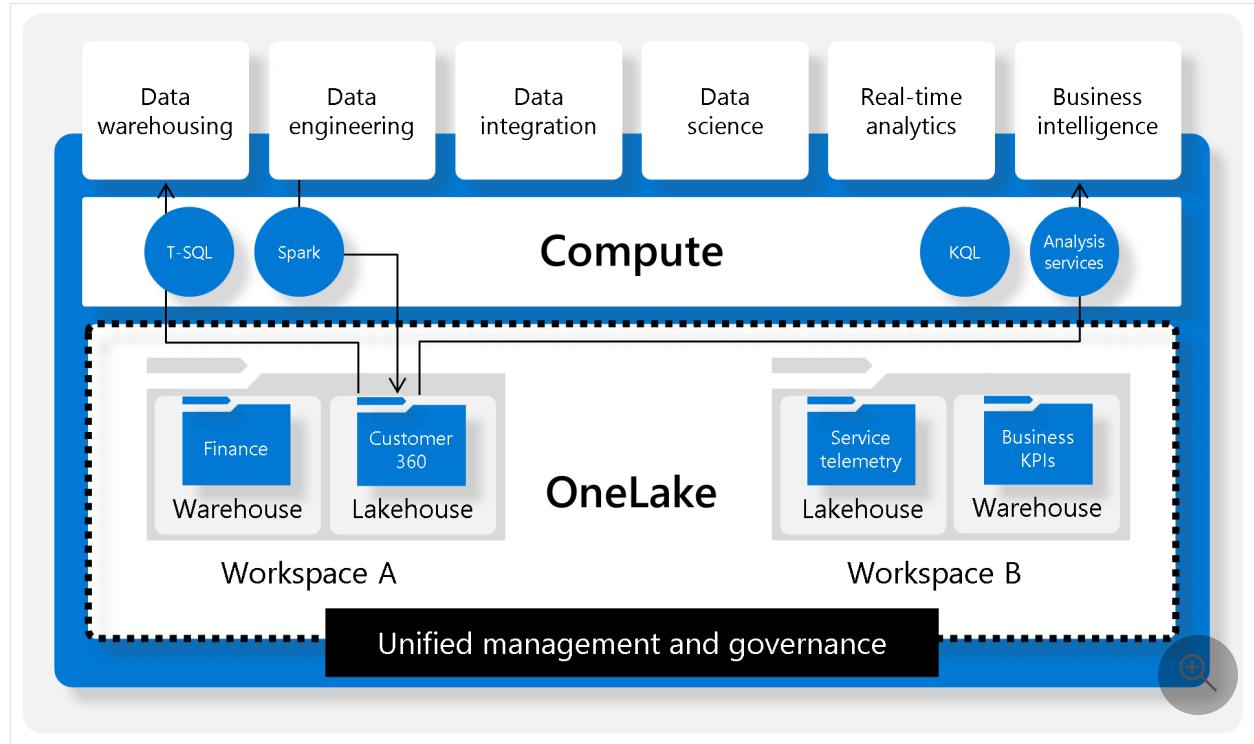
For more information on how to use shortcuts, see [OneLake shortcuts](#).

One copy of data with multiple analytical engines

While applications may have separation of storage and computing, the data is often optimized for a single engine, which makes it difficult to reuse the same data for multiple applications. With Fabric, the different analytical engines (T-SQL, Spark, Analysis Services, etc.) store data in the open delta parquet format to allow you to use the same data across multiple engines. There's no longer a need to copy data just to use it with another engine. You're always able to choose the best engine for the job that you're trying to do. For example, imagine you have a team of SQL engineers building a fully transaction data warehouse. They can use the T-SQL engine and all the power of T-SQL to create tables, transform, and load data to tables. If a data scientist wants to make use of this data, they no longer need to go through a special Spark/SQL driver. All data is stored in OneLake in delta parquet format. Data scientists can use the full power of the Spark engine and its open-source libraries directly over the data.

Business users can build Power BI reports directly on top of OneLake using the new direct lake mode in the Analysis Services engine. The Analysis Services engine is what powers Power BI Datasets and has always offered two modes of accessing data, import and direct query. Direct lake mode gives users all the speed of import without needing

to copy the data, combining the best of import and direct query. Learn more about direct lake: <https://aka.ms/DirectLake>.



Example diagram showing loading data using Spark, querying using T-SQL and viewing the data in a Power BI report.

Next steps

- Creating a lakehouse with OneLake

Create a lakehouse with OneLake

Article • 05/23/2023

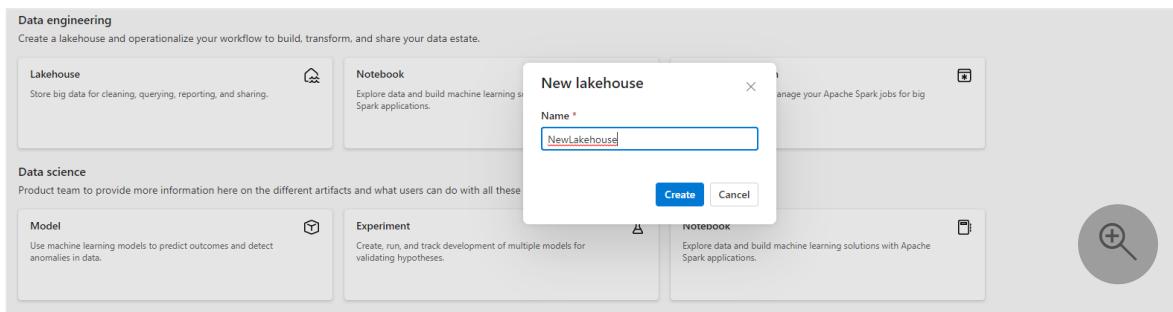
This tutorial is a quick guide to creating a lakehouse and getting started with the basic methods of interacting with it. After completing this tutorial, you'll have a lakehouse provisioned inside of Microsoft Fabric working on top of OneLake.

ⓘ Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Create a lakehouse

1. In the upper left corner of the workspace home page, select **New** and then **Show all**.
2. Scroll down and choose **Lakehouse** under the **Data engineering** header.
3. Give your lakehouse a name and select **Create**.

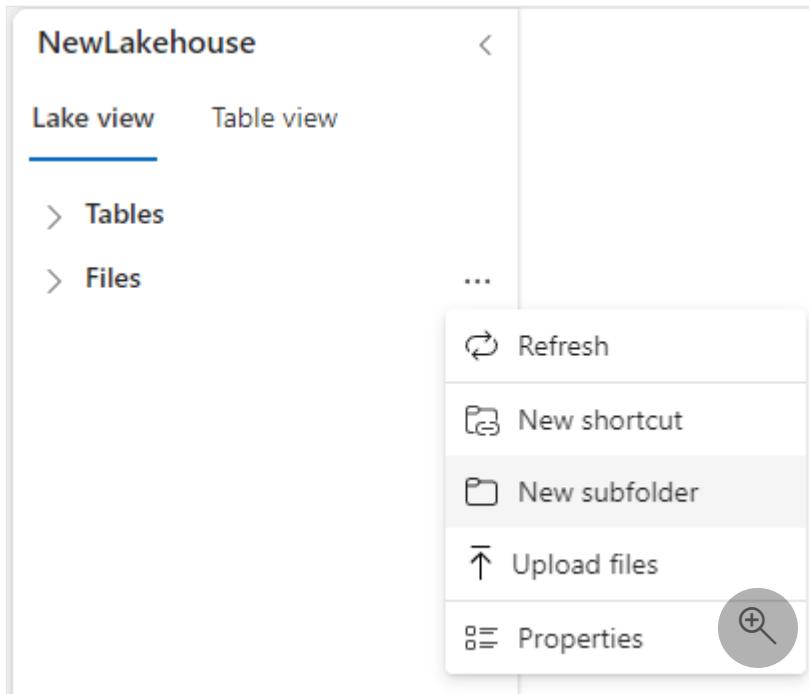


4. A new lakehouse is created and if this is your first OneLake item, it's provisioned behind the scenes.

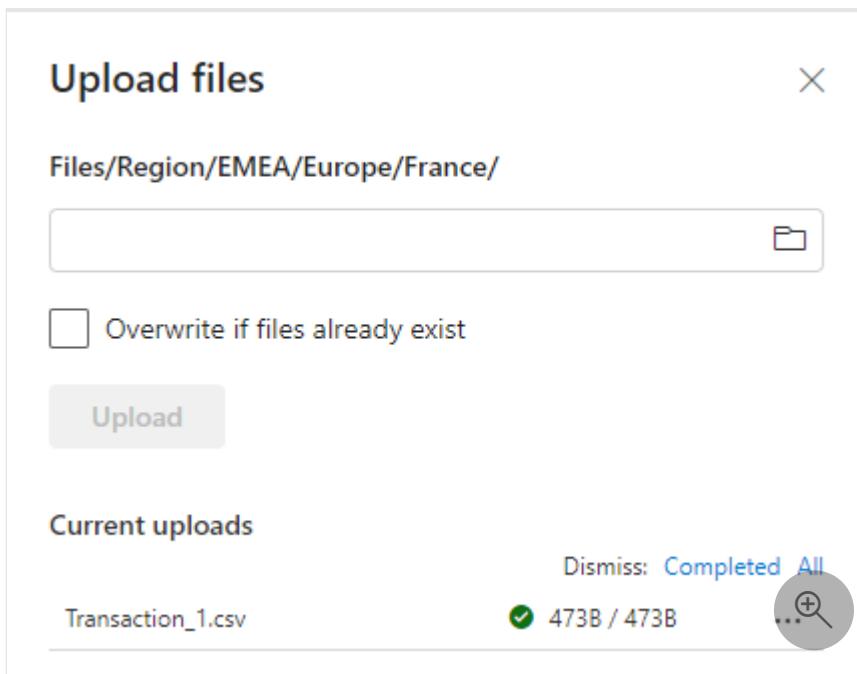
At this point, you have a lakehouse running on top of OneLake. Next, add some data and start organizing your lake.

Load data to a lakehouse

1. In the file browser on the left, select **Files** and then select **New subfolder**. Name your subfolder and select **Create**.



2. You can repeat this step to add more subfolders as needed.
3. Select a folder and the select **Upload files** from the list.
4. Choose the file you want from your local machine and then select **Upload**.



5. You've now added data to OneLake. To add data in bulk or schedule data loads into OneLake, use the **Get data** button to create pipelines. See more details about **Get data** features here: Data Integration Consolidated Documentation.
6. Select the More icon (...) for the file you uploaded and select **Properties** from the menu.

The **Properties** screen shows the various details for the file, including the URL and Azure Blob File System (ABFS) path for use with Notebooks. You can copy the ABFS into a Fabric Notebook to query the data using Spark. To learn more about notebooks in Fabric, see [Explore the data in your Lakehouse with a notebook](#).

Congratulations, you've created your first lakehouse with data stored in OneLake!

Next steps

- [OneLake shortcuts](#)

OneCopy: Transform data with Spark and query with SQL

Article • 05/23/2023

In this guide, you will:

- Upload data to OneLake using OneLake file explorer.
- Use a Fabric notebook to read data on OneLake and write back as a delta table.
- Analyze and transform data with Spark using a Fabric notebook.
- Query one copy of data on OneLake with SQL.

Important

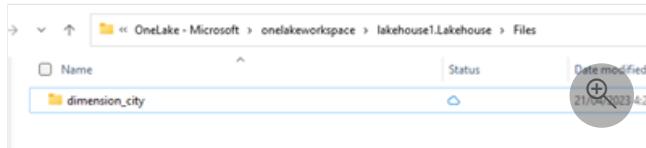
Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Prerequisites

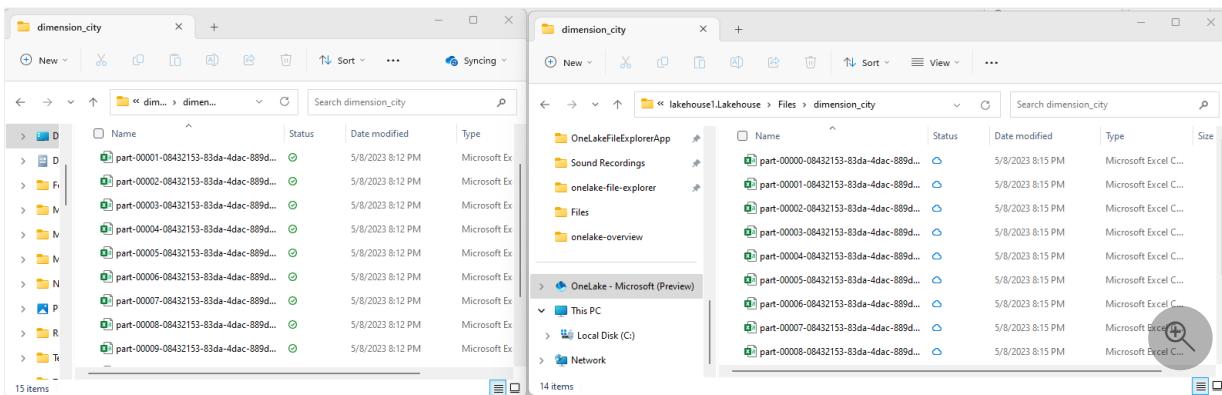
- Download and install [OneLake file explorer](#).
- A workspace with a lakehouse item
- Download the WideWorldImportersDW dataset to your computer to follow along with the instructions in this guide. You can use [Azure Storage Explorer](#) to connect to "https://azuresynapsestorage.blob.core.windows.net/sampledata/WideWorldImportersDW/csv/full/dimension_city" and download the set of csv files. You can also use your own csv data and update the details as required.

Steps

1. In OneLake file explorer, navigate to your lakehouse and under the /Files directory, create a subdirectory named dimension_city.



2. Copy your sample csv files to the OneLake directory /Files/dimension_city using OneLake file explorer.



3. Navigate to your lakehouse in the Power BI service and view your files.

Name
part-00001-08432153-83da-4dac-889d-ca08c4a341e5-c000.csv
part-00002-08432153-83da-4dac-889d-ca08c4a341e5-c000.csv
part-00003-08432153-83da-4dac-889d-ca08c4a341e5-c000.csv
part-00004-08432153-83da-4dac-889d-ca08c4a341e5-c000.csv
part-00005-08432153-83da-4dac-889d-ca08c4a341e5-c000.csv
part-00006-08432153-83da-4dac-889d-ca08c4a341e5-c000.csv

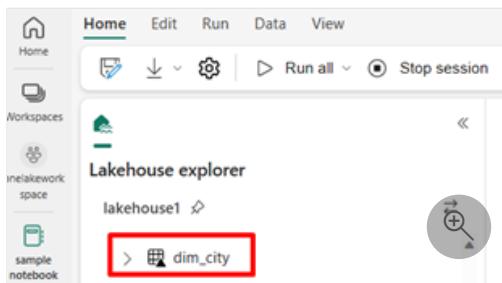
4. Select Open notebook, then New notebook to create a notebook.

5. Using the Fabric notebook, convert the CSV files to delta format. The following code snippet reads data from user created directory /Files/dimension_city and converts it to a delta table dim_city.

```
Python

import os
from pyspark.sql.types import *
for filename in os.listdir("/lakehouse/default/Files/<replace with your folder path>"):
    df=spark.read.format('csv').options(header="true",inferSchema="true").load("abfss://<replace with workspace name>@onelake.dfs.fabric.microsoft.com<replace with item name>.Lakehouse/Files/<folder name>/"+filename,on_bad_lines="skip")
    df.write.mode("overwrite").format("delta").save("Tables/<name of delta table>")
```

6. Refresh your view of the /Tables directory to see your new table.



7. Query your table with SparkSQL in the same Fabric notebook.

Python

```
%%sql  
SELECT * from <replace with item name>.dim_city LIMIT 10;
```

8. Modify the delta table by adding a new column named newColumn with data type integer. Set the value of 9 for all of the records for this newly added column.

Python

```
%%sql  
  
ALTER TABLE <replace with item name>.dim_city ADD COLUMN newColumn int;  
  
UPDATE <replace with item name>.dim_city SET newColumn = 9;  
  
SELECT City,newColumn FROM <replace with item name>.dim_city LIMIT 10;
```

9. Any delta table on OneLake can also be accessed via a SQL Endpoint. This SQL endpoint references the same physical copy of delta table on OneLake and offers T-SQL experience. Select the SQL Endpoint for lakehouse1 and then select "New SQL Query" to query the table using T-SQL

SQL

```
SELECT TOP (100) * FROM [<replace with item name>].[dbo].[dim_city];
```

Summary

In this quickstart guide, you used OneLake File explorer to copy external datasets to OneLake. The datasets were then transformed to delta table and analyzed using lakehouse and T-SQL experiences.

Shortcut ADLS data transformed with Azure Databricks into OneLake and build Power BI Report.

Article • 05/23/2023

ⓘ Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

In this guide, you will:

- Create a delta table in your ADLS Gen2 account using Azure Databricks.
- Create a OneLake shortcut to a delta table in ADLS.
- Use Power BI to analyze data in ADLS Shortcut.

Prerequisites

- A workspace with a Lakehouse item.
- An Azure Databricks workspace.
- An ADLS Gen2 account to store delta tables.

Steps

1. Using Azure Databricks notebook, create a delta table in ADLS Gen2 account.

Python

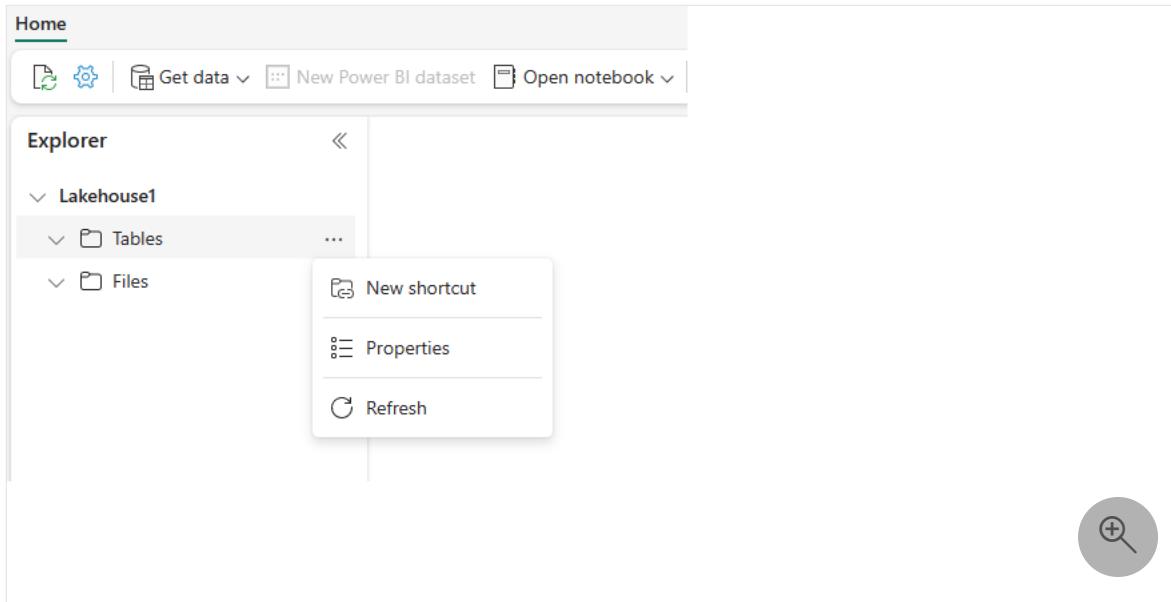
```
# Replace the path below to refer to your sample parquet data with
# this syntax "abfss://<storage name>@<container
# name>.dfs.core.windows.net/<filepath>"

# Read Parquet files from an ADLS account
df =
spark.read.format('Parquet').load("abfss://datasetsv1@olsdemo.dfs.core.
windows.net/demo/full/dimension_city/")

# Write Delta tables to ADLS account
```

```
df.write.mode("overwrite").format("delta").save("abfss://datasetsv1@ol  
sdemo.dfs.core.windows.net/demo/adb_dim_city_delta/")
```

2. In your lakehouse, click on ellipses (...) next to the Tables and select New Shortcut.



3. In the New shortcut screen, select Azure Data Lake Storage Gen2 tile



4. Specify the connection details this shortcut will use and select Next

New shortcut

(i) Geography_Lakehouse is located in the region **West Central US**. Any data sourced through this shortcut will be processed in the same region.

 Azure Data Lake Storage
Gen2
Azure

Connection settings

URL * ⓘ
Example: https://contosoadlscdm.dfs.core.windows.net/file...

Connection credentials

Connection
Create new connection ⚡

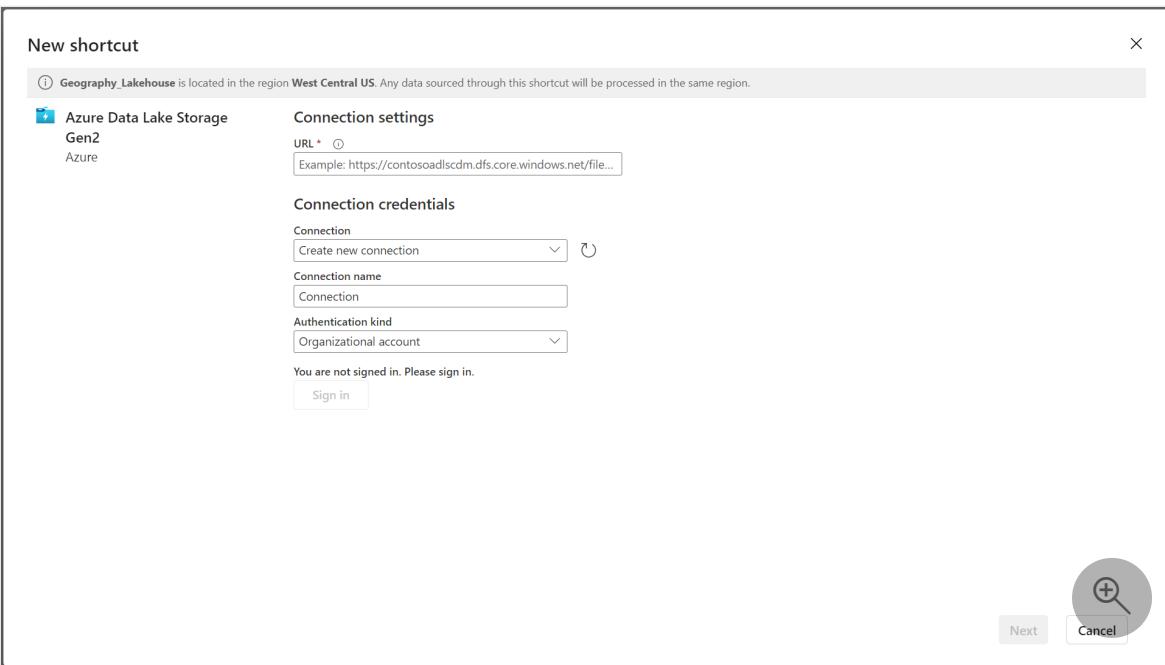
Connection name
Connection

Authentication kind
Organizational account

You are not signed in. Please sign in.
[Sign in](#)



Next Cancel



5. Specify the shortcut details. Provide a **Shortcut Name** and **Sub path** details and then click **Create**. The sub path should point to the directory where the delta table resides.

New shortcut

(i) Geography_Lakehouse is located in the region **West Central US**. Any data sourced through this shortcut will be processed in the same region.

 Azure Data Lake Storage
Gen2
Azure

Shortcut settings

Shortcut Name *

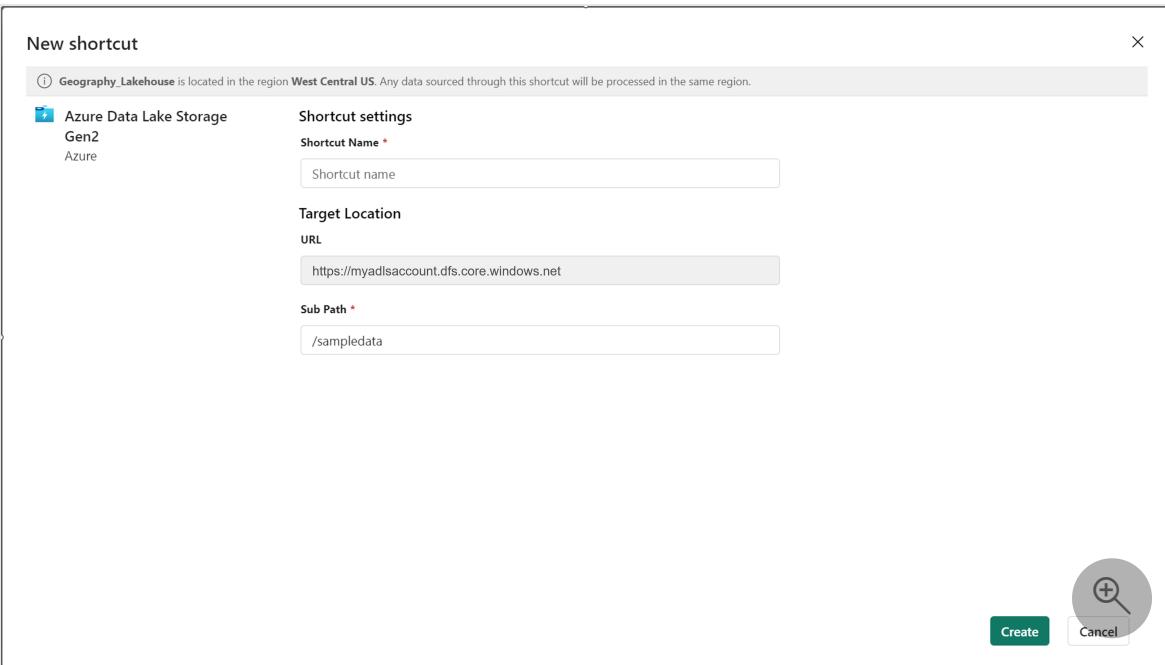
Target Location

URL

Sub Path *



Create Cancel



6. The shortcut pointing to a delta table created by Azure Databricks on ADLS now appears as a delta table under Tables.

The screenshot shows the Azure Data Lake Storage Explorer interface. On the left, there's a tree view under 'lakehouse1/Tables'. A specific table, 'adls_shortcut_adb_dim_city_delta', is selected and highlighted with a red box. Below it, other tables are listed: 'ABC CityKey', 'ABC WWICityID', 'ABC City', 'ABC StateProvince', 'ABC Country', and 'ABC Continent'. On the right side of the interface, there's a vertical bar with a search icon at the bottom.

7. This data can now be queried directly from notebook.

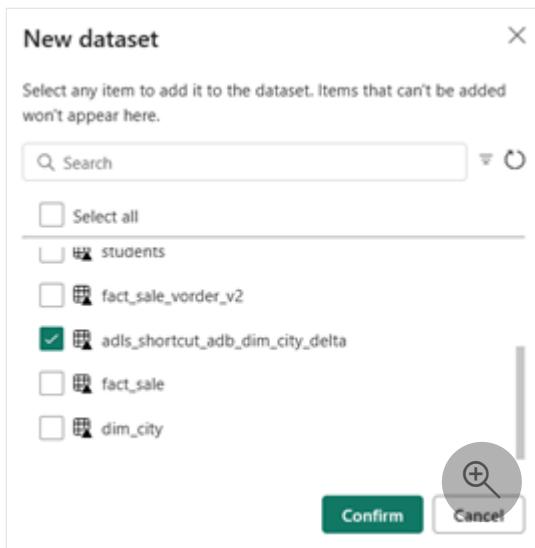
Python

```
df = spark.sql("SELECT * FROM  
lakehouse1.adls_shortcut_adb_dim_city_delta LIMIT 1000")  
display(df)
```

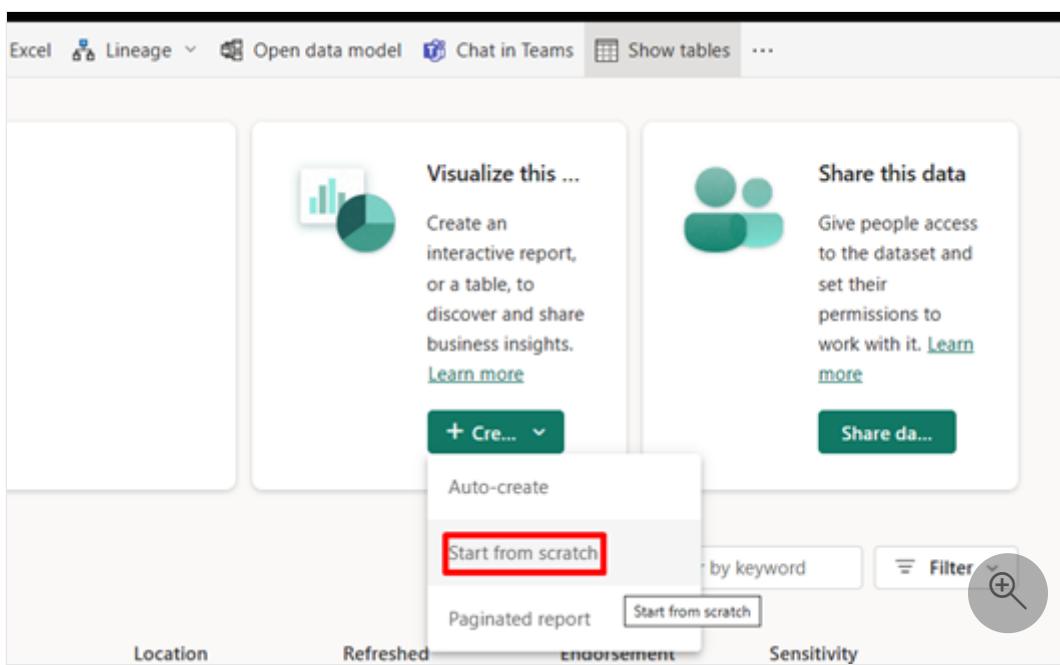
8. To access and analyze this delta table via Power BI, click on **New Power BI dataset**.

The screenshot shows the Azure Data Lake Storage Explorer interface again. At the top, there are several buttons: 'Get data', 'New Power BI dataset' (which is highlighted with a red box), and 'Open notebook'. Below the buttons, a message says: 'A SQL endpoint for SQL querying and a default dataset for reporting were created and will be updated automatically'. The main area shows the 'Explorer' tree view with 'lakehouse1/Tables' expanded, showing 'adls_shortcut_adb_dim_city_delta' and 'dim_city'.

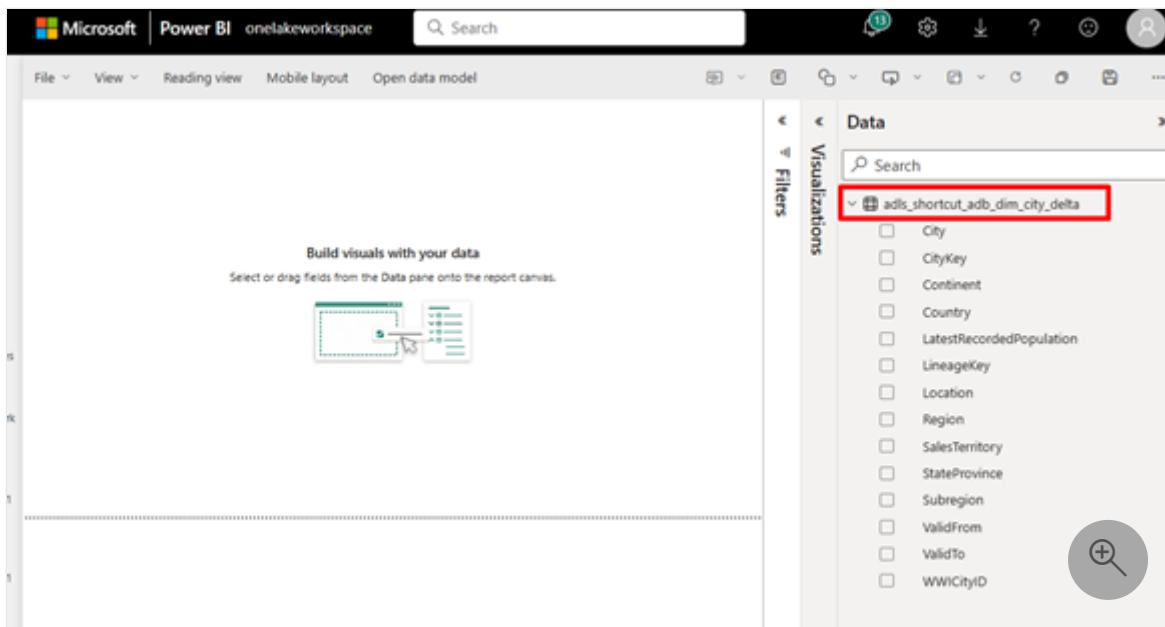
9. Select the shortcut and click on **Confirm**.



10. Once data has been published, click on Start from scratch.



11. In the report authoring experience, the shortcut data appears as a table along with all of its attributes.



12. Drag the attributes to the pane on the left-hand side to build a Power BI report.

A screenshot of the Microsoft Power BI application interface. The left side features a table visualization showing data from the 'adls_shortcut_adb_dim_city_delta' table. The columns are: City, Country, Location, Region, SalesTerritory, StateProvince, and Subregion. The data includes rows for cities like New York, Los Angeles, Chicago, etc., with their respective country, location, region, sales territory, state province, and subregion. To the right of the table is the 'Data' pane, which is identical to the one shown in the previous screenshot, displaying the same list of fields with 'Country' checked. A red box highlights the 'Country' field in the Data pane.

Summary

In this quickstart guide, you created a OneLake shortcut to read an Azure Databricks delta table on ADLS account. This shortcut is then used to analyze data using a notebook and Power BI report.

Open Access: Use pipelines to ingest data into OneLake and analyze with Azure Databricks

Article • 05/23/2023

ⓘ Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

In this guide, you will:

- Create a pipeline in workspace and ingest data into OneLake in delta format
- Analyze delta table in OneLake using Azure Databricks

Prerequisites

- A workspace with a lakehouse item.
- A premium Azure Databricks workspace. Only premium Azure Databricks workspaces support Microsoft Azure Active Directory credential passthrough. When creating your cluster, enable Azure Data Lake Storage credential passthrough in the Advanced Options.
- A sample dataset.

Steps

1. Navigate to your lakehouse in the Power BI service and select **Get Data** and then select **New data pipeline**.

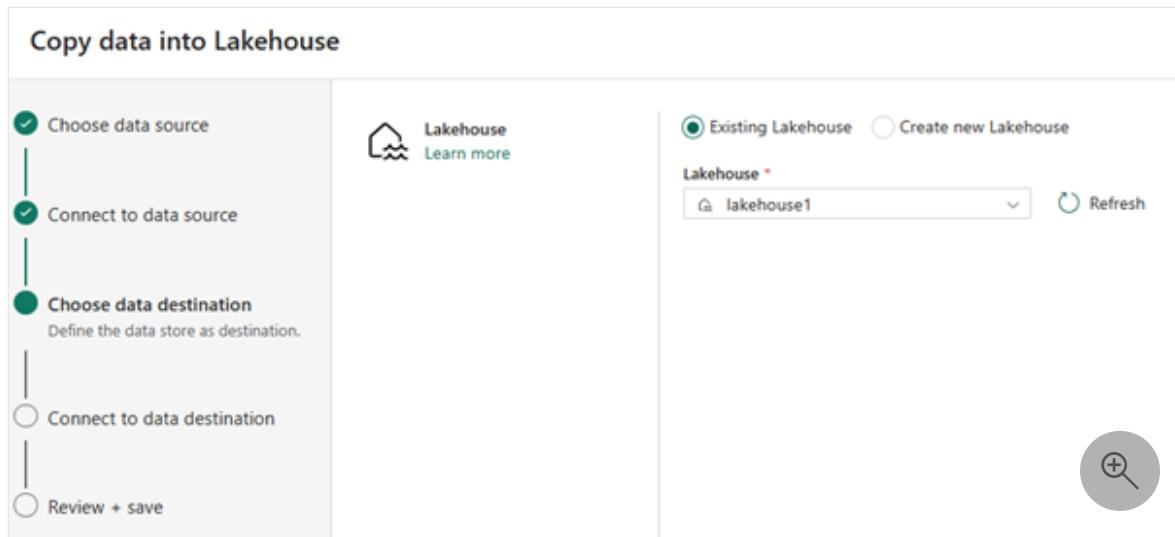
The screenshot shows the Power BI Home interface. In the top navigation bar, there are icons for 'Get data', 'New Power BI dataset', and 'Open notebook'. Below the navigation bar is the 'Explorer' pane, which lists a 'lakehouse' folder containing 'dim_city', 'nycsample', and 'dimension_city'. A red box highlights the 'New data pipeline' option under the 'Upload files' section. To the right of the Explorer pane is a preview of the 'dim_city' table, showing columns 'CityKey' and 'WWICityID' with five rows of data. A magnifying glass icon is located in the bottom right corner of the preview area.

2. In the New Pipeline prompt, enter a name for the new pipeline and then select **Create**.
3. For this exercise, choose NYC Taxi - Green sample data as the data source. Select **Next** after the sample dataset has been selected.

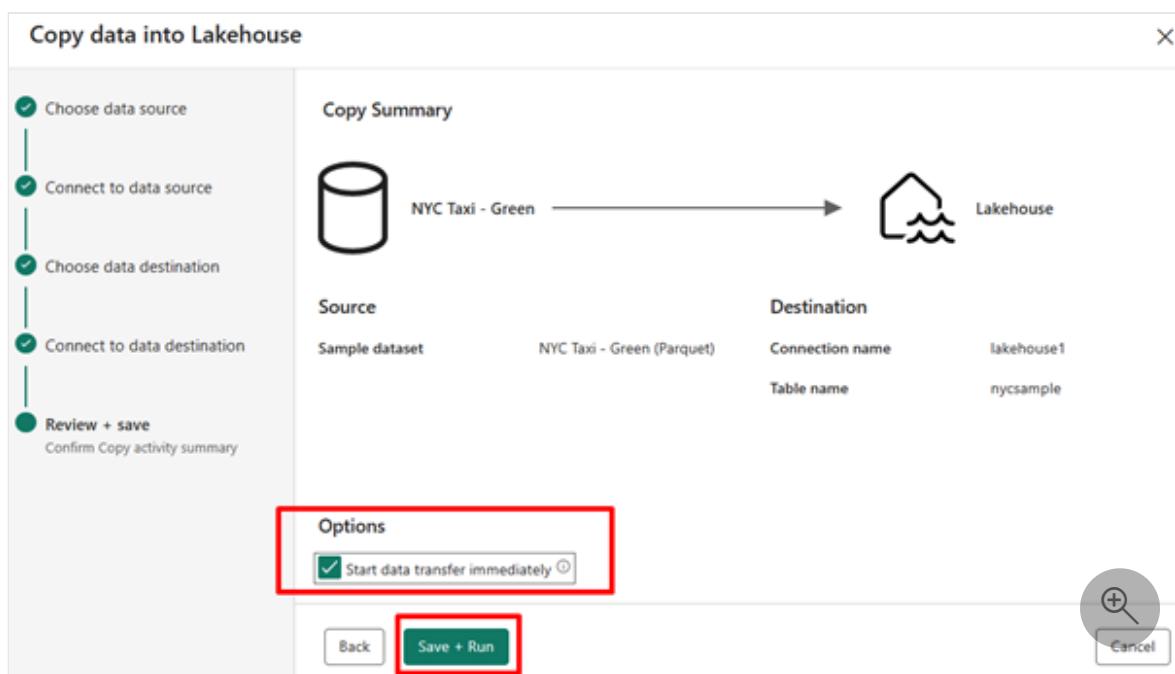
The screenshot shows the 'Sample data' selection screen. It displays four datasets: 'COVID-19 Data Lake', 'NYC Taxi - Green', 'Diabetes', and 'Public Holidays'. The 'NYC Taxi - Green' dataset is highlighted with a red box. Below the datasets are two buttons: 'Back' and 'Next'. A red box also highlights the 'Next' button. To the right of the 'Next' button is a magnifying glass icon.

4. Select **Next** again at the preview screen.
5. For data destination, select the name of the lakehouse where you want to store the data on OneLake as a delta table. You can choose an existing lakehouse or create a

new lakehouse.



6. Select where you want to store the output. Choose Tables as the Root folder and enter 'nycsample' as the table name.
7. At the Review + Save screen, select **Start data transfer immediately** and then select **Save + Run**.



8. Once the job has completed, navigate to your lakehouse and view the delta table listed under /Tables.
9. Copy the abfs path to your delta table to by right-clicking the table name in the Explorer view and selecting **Properties**.
10. Open your Azure Databricks notebook. Read the delta table on OneLake.

Python

```
olsPath = "abfss://<replace with workspace  
name>@onelake.dfs.fabric.microsoft.com/<replace with item  
name>.Lakehouse/Tables/nycsample"  
df=spark.read.format('delta').option("inferSchema", "true").load(olsPath  
)  
df.show(5)
```

11. Update data in the delta table on OneLake by updating a value of a field in the delta table.

Python

```
%sql  
update delta.`abfss://<replace with workspace  
name>@onelake.dfs.fabric.microsoft.com/<replace with item  
name>.Lakehouse/Tables/nycsample` set vendorID = 99999 where vendorID =  
1;
```

Summary

In this guide, you ingested data into OneLake using the pipeline experience and created a delta table. The delta table on OneLake is then read and modified via Azure Databricks.

OneLake shortcuts

Article • 05/23/2023

Shortcuts in Microsoft OneLake allow you to unify your data across domains, clouds and accounts by creating a single virtualized data lake for your entire enterprise. All Fabric experiences and analytical engines can directly connect to your existing data sources such as Azure, AWS and OneLake through a unified namespace. Permissions and credentials are all managed by OneLake, so each Fabric experience doesn't need to be separately configured to connect to each data source. Additionally, you can use shortcuts to eliminate edge copies of data and reduce process latency associated with data copies and staging.

ⓘ Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

What are shortcuts?

Shortcuts are objects in OneLake that point to other storage locations. The location can be internal or external to OneLake. The location that a shortcut points to is known as the "Target" path of the shortcut. The location that the shortcut appears is known as the "Shortcut" path. Shortcuts appear as folders in OneLake and can be used transparently by any experience or service that has access to OneLake. Shortcuts behave similar to symbolic links. They're an independent object from the target. If a shortcut is deleted, the target remains unaffected. If the target path is moved, renamed, or deleted the shortcut can break.

The screenshot displays two views of Microsoft Fabric:

- Lake View:** Shows a hierarchical file system structure. The "Managed" section contains a "Tables" folder which includes a "DeltaExample" item. The "Unmanaged" section contains a "Files" folder with "FolderA", "FolderB", "Shortcut2", and "Shortcut1". A dashed blue line connects the "DeltaExample" item in the Lake View to its corresponding table in the Table View.
- Table View:** Shows a tabular dataset named "DeltaExample". The table has columns "ID" and "Name". The data rows are:

ID	Name
1	Person 1
2	Person 2
3	Person 3
4	Person 4
5	Person 5

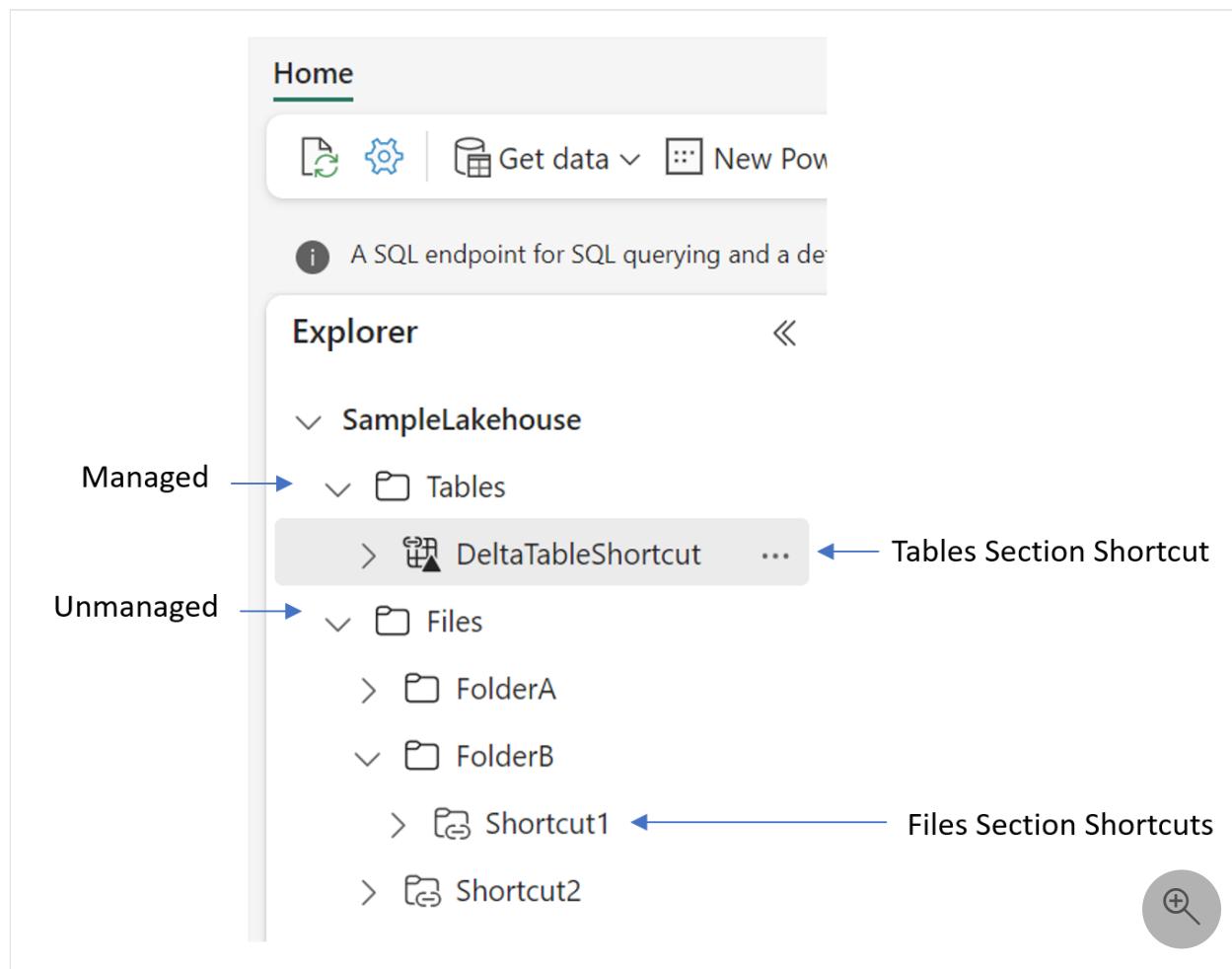
A magnifying glass icon is positioned over the "Person 4" row.

Where can I create shortcuts?

Shortcuts can be created both in Lakehouses and KQL Databases. Furthermore, the shortcuts created within these items can point to other OneLake locations, ADLS Gen2 or Amazon S3 storage accounts.

Lakehouse

When creating shortcuts in a Lakehouse, it's important to understand the folder structure of the item. Lakehouses are composed of two top level folders: The "Tables" folder and the "Files" folder. The "Tables" folder represents the managed portion of the Lakehouse while the "Files" folder is the unmanaged portion of the Lakehouse. In the "Tables" folder, you can only create shortcuts at the top level. Shortcuts aren't supported in other subdirectories of the "Tables" folder. If the target of the shortcut contains data in the delta\parquet format, the Lakehouse will automatically synchronize the metadata and recognize the folder as a Table. In the "Files" folder, there are no restrictions on where you can create shortcuts. They can be created at any level of the folder hierarchy. Table discovery doesn't happen in the "Files" folder.



KQL Database

When you create a shortcut in a KQL database, it appears in the "Shortcuts" folder of the database. The KQL database treats shortcuts like external tables. To query the shortcut, use the "external_table" function of the Kusto Query Language.

The screenshot shows the OneLake interface. On the left, the 'Data tree' sidebar is open, showing a 'Database' section with a search bar. Below it are sections for 'Tables', 'Shortcuts' (which is expanded), 'Materialized views', 'Functions', and 'Data streams'. Under 'Shortcuts', there are two entries: 'S3_Campaign' and 'ADLS_Campaign'. On the right, the main area is titled 'Database / Shortcuts' and lists the same two shortcut names. A circular button with a plus sign is visible in the bottom right corner of this area.

Where can I access shortcuts?

Any Fabric or non-Fabric service that can access data in OneLake can utilize shortcuts. Shortcuts are transparent to any service accessing data through the OneLake API. Shortcuts just appear as another folder in the lake. This allows Spark, SQL, Real-Time Analytics and Analysis Services to all utilize shortcuts when querying data.

Spark

Spark notebooks and Spark jobs can utilize shortcuts that are created in OneLake. Relative file paths can be used to directly read data from shortcuts. Additionally, if a shortcut is created in the "Tables" section of the Lakehouse and is in the delta format, it can also be read as a managed tables using Spark SQL syntax.

Python

```
df = spark.read.format("delta").load("Tables/MyShortcut")
display(df)
```

Python

```
df = spark.sql("SELECT * FROM MyLakehouse.MyShortcut LIMIT 1000")
display(df)
```

ⓘ Note

The delta format doesn't support tables with space characters in the name. Any shortcut containing a space in the name won't be discovered as a delta table in the lakehouse.

SQL

Shortcuts in the tables section of the Lakehouse can also be read through the SQL endpoint for the Lakehouse. This can be accessed through mode selector of the Lakehouse or through SQL Server Management Studio (SSMS).

SQL

```
SELECT TOP (100) *
FROM [MyLakehouse].[dbo].[MyShortcut]
```

Real-Time Analytics

Shortcuts in KQL DBs are recognized as external tables. To query the shortcut, use the "external_table" function of the Kusto Query Language.

Kusto

```
external_table('MyShortcut')
| take 100
```

ⓘ Note

KQL databases don't currently support data in the delta format. Tables in a KQL database are only exported to OneLake as parquet files. Shortcuts in KQL databases that contain delta formatted data in the target aren't recognized as tables.

Analysis Services

Power BI datasets can be created for Lakehouses containing shortcuts in the tables section of the Lakehouse. When the dataset runs in direct-lake mode, Analysis Services can read data directly from the shortcut.

Non-Fabric

Applications and services outside of Fabric can also access shortcuts through the OneLake API. OneLake supports a subset of the ADLS Gen2 and Blob storage APIs. To learn more about the OneLake API, see [OneLake access with APIs](#).

HTTP

```
https://onelake.dfs.fabric.microsoft.com/MyWorkspace/MyLakehouse/Tables/MyShortcut/MyFile.csv
```

Types of shortcuts

OneLake shortcuts support multiple filesystem data sources. These include internal OneLake locations, Azure Data Lake Storage Gen2 and Amazon S3.

Internal OneLake shortcuts

Internal OneLake shortcuts allow you to reference data within existing Fabric items. These items include Lakehouses, KQL Databases and Data Warehouses. The shortcut can point to a folder location within the same item, across items within the same workspace or even across items in different workspaces. When you create a shortcut across items, the item types don't need to match. For instance, you can create a shortcut in a Lakehouse that points to data in a Data Warehouse.

When accessing data through a shortcut to another OneLake location, the identity of the calling user will be utilized to authorize access to the data in the target path of the shortcut*. This user must have permissions in the target location to read the data.

Important

When accessing shortcuts through Power BI Datasets or T-SQL, **the calling user's identity is not passed through to the shortcut target**. The calling item owner's

identity is passed instead, delegating access to the calling user.

ADLS shortcuts

Shortcuts can also be created to ADLS Gen2 storage accounts. When you create shortcuts to ADLS, the target path can point to any folder within the hierarchical namespace. At a minimum, the target path must include a container name.

Access:

ADLS shortcuts must point to the DFS endpoint for the storage account.

Example: <https://accountname.dfs.core.windows.net/>

ⓘ Note

ADLS shortcuts don't support private endpoints.

Authorization:

ADLS shortcuts utilize a delegated authorization model. In this model, the shortcut creator specifies a credential for the ADLS shortcut and all access to that shortcut will be authorized using that credential. The supported delegated types are Account Key, SAS Token, OAuth and Service Principal.

- **SAS Token** - must include at least the following permissions: Read, List, Execute
- **OAuth identity** - must have Storage Blob Data Reader, Storage Blob Data Contributor or Storage Blob Data Owner role on storage account.
- **Service Principal** - must have Storage Blob Data Reader, Storage Blob Data Contributor or Storage Blob Data Owner role on storage account.

S3 shortcuts

Shortcuts can also be created to Amazon S3 accounts. When you create shortcuts to Amazon S3, the target path must contain a bucket name at a minimum. S3 doesn't natively support hierarchical namespaces but you can utilize prefixes to mimic a directory structure. You can include prefixes in the shortcut path to further narrow the scope of data accessible through the shortcut. When accessing data through an S3 shortcut prefixes will be represented as folders.

Access:

S3 shortcuts must point to the https endpoint for the S3 bucket. Example:

`https://bucketname.s3.region.amazonaws.com/`

 **Note**

S3 shortcuts don't support private endpoints.

Authorization:

S3 shortcuts utilize a delegated authorization model. In this model, the shortcut creator specifies a credential for the S3 shortcut and all access to that shortcut will be authorized using that credential. The supported delegated credential is a Key and Secret for an IAM user.

The IAM must have at least read only (Get, List) permissions on the bucket that the shortcut is pointing to.

 **Note**

S3 shortcuts are read-only. They don't support write operations regardless of the permissions for the IAM user.

How shortcuts utilize cloud connections

ADLS and S3 shortcut authorization is delegated through the use of cloud connections. When creating a new ADLS or S3 shortcut, a user either creates a new connection or selects an existing connection for the data source. When a connection is set for a shortcut, this is considered a "bind" operation. Only users with permission on the connection can perform the bind operation. If a user doesn't have permissions on the connection, they can't create new shortcuts using that connection.

Permissions

Permissions for shortcuts are governed by a combination of the permissions in the shortcut path and the target path. When a user accesses a shortcut, the most restrictive permission of the two locations is applied. Therefore, a user that has read/write permissions in the Lakehouse but only read permissions in the shortcut target won't be allowed to write to the shortcut target path. Likewise, a user that only has read permissions in the Lakehouse but read/write in the shortcut target will also not be allowed to write to the shortcut target path.

Workspace roles

The following table shows the shortcut-related permissions for each workspace role. For more information, see [Workspace roles](#).

Capability	Admin	Member	Contributor	Viewer
Create a shortcut	Yes ¹	Yes ¹	Yes ¹	-
Read file/folder content of shortcut	Yes ²	Yes ²	Yes ²	-
Write to shortcut target location	Yes ³	Yes ³	Yes ³	-
Read data from shortcuts in table section of the Lakehouse via TDS endpoint	Yes	Yes	Yes	Yes

¹ User must have a role that provides write permission the shortcut location and at least read permission target location

² User must have a role that provides read permission both in the shortcut location and target location

³ User must have a role that provides write permission both in the shortcut location and the target location

How do shortcuts handle deletions?

Shortcuts don't perform cascading deletes. When you perform a delete operation on a shortcut, you only delete the shortcut object. The data in the shortcut target remains unchanged. However, if you perform a delete operation on a file or folder within a shortcut, and you have permissions in the shortcut target to perform the delete operation, the files and/or folders are deleted in the target. The following example illustrates this point.

Delete example

User A has a lakehouse with the following path in it:

MyLakehouse\Files\MyShortcut\Foo\Bar

MyShortcut is a shortcut that points to an ADLS Gen2 account that contains the Foo\Bar directories.

Deleting a shortcut object

User A performs a delete operation on the following path:

MyLakehouse\Files\MyShortcut

In this case, **MyShortcut** is deleted from the Lakehouse. Shortcuts don't perform cascading deletes, therefore the files and directories in the ADLS Gen2 account *Foo\Bar* remain unaffected.

Deleting content referenced by a shortcut

User A performs a delete operation on the following path:

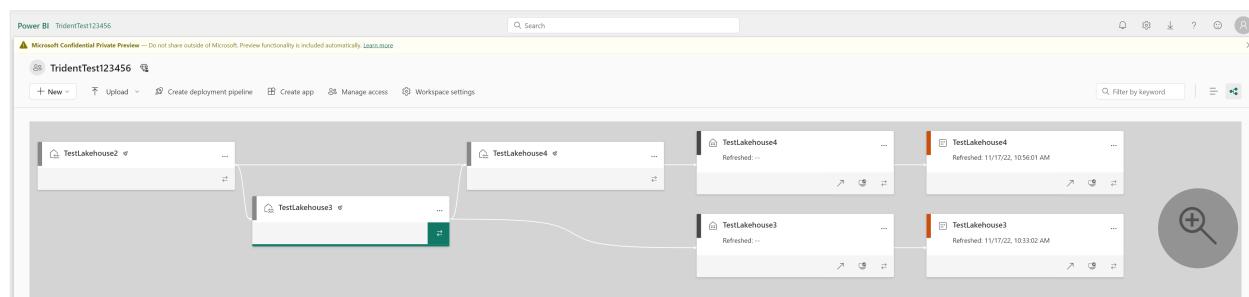
MyLakehouse\Files\MyShortcut\Foo\Bar

In this case, if User A has write permissions in the ADLS Gen2 account, the **Bar** directory is deleted from the ADLS Gen2 account.

Workspace lineage view

When creating shortcuts between multiple Fabric items within a workspace, you can visualize the shortcut relationships through the workspace lineage view. Select the

Lineage view button () in the upper right corner of the Workspace explorer.



ⓘ Note

The lineage view is scoped to a single workspace. Shortcuts to locations outside the selected workspace won't appear.

Known issues and limitations

- The maximum number of shortcuts per Fabric item is 10,000.
- The maximum number of shortcuts in a single OneLake path is 10.
- The maximum number of direct shortcut to shortcut links is 5.
- ADLS and S3 shortcut target paths can't contain any reserved characters from RCF 3986 section 2.2.
- OneLake shortcut target paths can't contain "%" characters.
- Shortcuts don't support nonlatin characters.
- Copy Blob api not supported for ADLS or S3 shortcuts.
- Copy function doesn't work on shortcuts that directly point to ADLS containers. It's recommended to create ADLS shortcuts to a directory that is at least one level below a container.
- OneLake shortcuts pointing to ADLS or S3 shortcuts isn't supported.
- Additional shortcuts can't be created inside ADLS or S3 shortcuts.

Next steps

- [Creating shortcuts](#)

How-to: Create a OneLake shortcut

Article • 05/23/2023

In this how-to guide, you'll learn how to create a OneLake shortcut inside a Fabric Lakehouse. For an overview of shortcuts, see [OneLake shortcuts](#).

Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

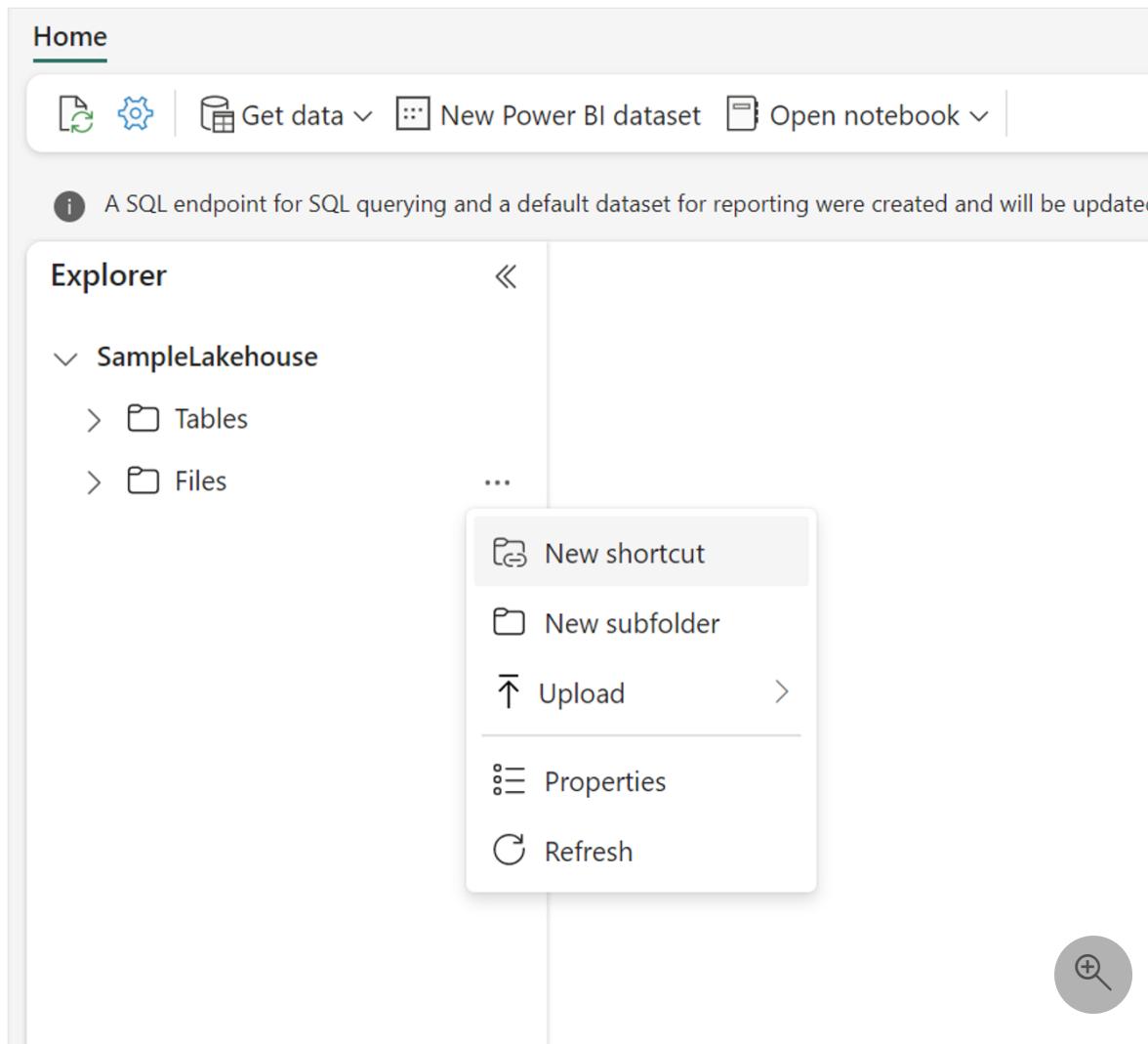
Prerequisite

Create a lakehouse by following these steps: [Creating a lakehouse with OneLake](#).

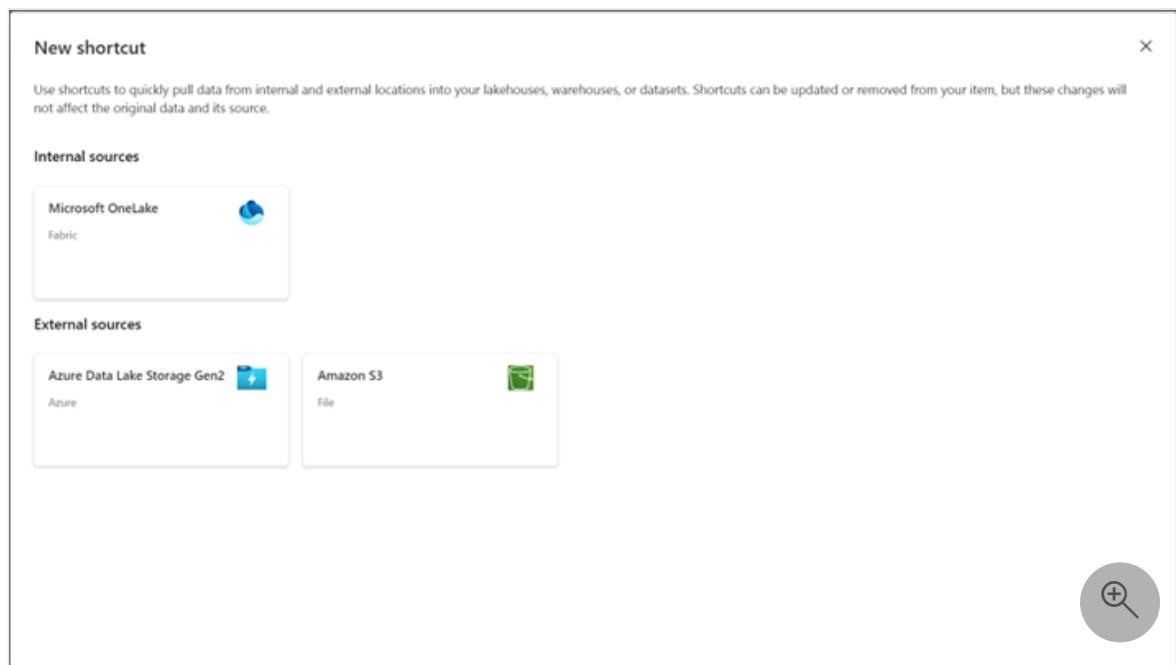
How to create a OneLake shortcut

To create a OneLake shortcut:

1. Open a lakehouse.
2. Right click on a directory within the **Lake** view of the lakehouse.
3. Select **New shortcut**.



4. Select the Microsoft OneLake tile.



5. Choose a Fabric item for the shortcut to point to. This can be a Lakehouse, Data Warehouse or KQL Database. Then select Next.

Select a data source type

⚠️ SQL-based services using this shortcut are authorized to access the data using the item owner's identity.

ⓘ Geography_Lakehouse is located in the region West Central US. Any data sourced through this shortcut will be processed in the same region.

Find and connect to the data you want to use with your shortcut.

	Name	Type	Capacity region	Owner	Location	Endorsement	Sensitivity
Workspaces	Geography_Lakehouse	Lakehouse	West Central US	Sam Centrell	NorthwindDemo	-	General ⓘ
	S3_Example	Lakehouse	West Central US	Elizabeth Moore	NorthwindDemo	-	General ⓘ
	NorthwindDemo	Lakehouse	West Central US	Eugenia Lopez	NorthwindDemo	-	General ⓘ
	TestLakehouse	Lakehouse	West Central US	Casey Jensen	NorthwindDemo	-	General ⓘ
	HR_Lakehouse	Lakehouse	West Central US	Sam Centrell	NorthwindDemo	-	General ⓘ

Previous Next +🔍 Cancel

6. Select a folder, then select **Create**.

New shortcut

⚠️ SQL-based services using this shortcut are authorized to access the data using the item owner's identity.

ⓘ Geography_Lakehouse is located in the region West Central US. Any data sourced through this shortcut will be processed in the same region.

Find and connect to the data you want to use with your shortcut.

OneLake	HR_Lakehouse > Tables > employees
HR_Lakehouse	Name Field type Last modified
Tables	_delta_log Folder Thu, 19 Jan 2023 18:37:46 GMT
employees	part-00000-dd521fa5-5... parquet Thu, 19 Jan 2023 18:37:47 GMT
Files	_delta_log
csv	

Previous Create +🔍 Cancel

7. See the folder icon with shortcut symbol in the explorer.

The screenshot shows the Power BI Home interface. At the top, there are navigation icons for Refresh, Settings, Get data, New Power BI dataset, and Open notebook. A status message indicates that a SQL endpoint and default dataset were created. The left pane, titled 'Explorer', shows a hierarchical view of 'SampleLakehouse' with 'Tables' and 'Files' expanded. Under 'Files', four items are listed: 'ADLS_Shortcut', 'LocalData', 'OneLake_Shortcut', and 'S3_Shortcut'. The right pane, titled 'Files', lists the same four items under the 'Name' column. A magnifying glass icon is located at the bottom right of the file list.

Name
ADLS_Shortcut
LocalData
OneLake_Shortcut
S3_Shortcut

Next steps

- [How-to: Create an ADLS Gen2 shortcut](#)
- [How-to: Create an Amazon S3 shortcut](#)

How-to: Create an ADLS Gen2 shortcut

Article • 05/23/2023

In this how-to guide, you'll learn how to create an ADLS shortcut inside a Fabric Lakehouse. For an overview of shortcuts, see [OneLake shortcuts](#).

Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

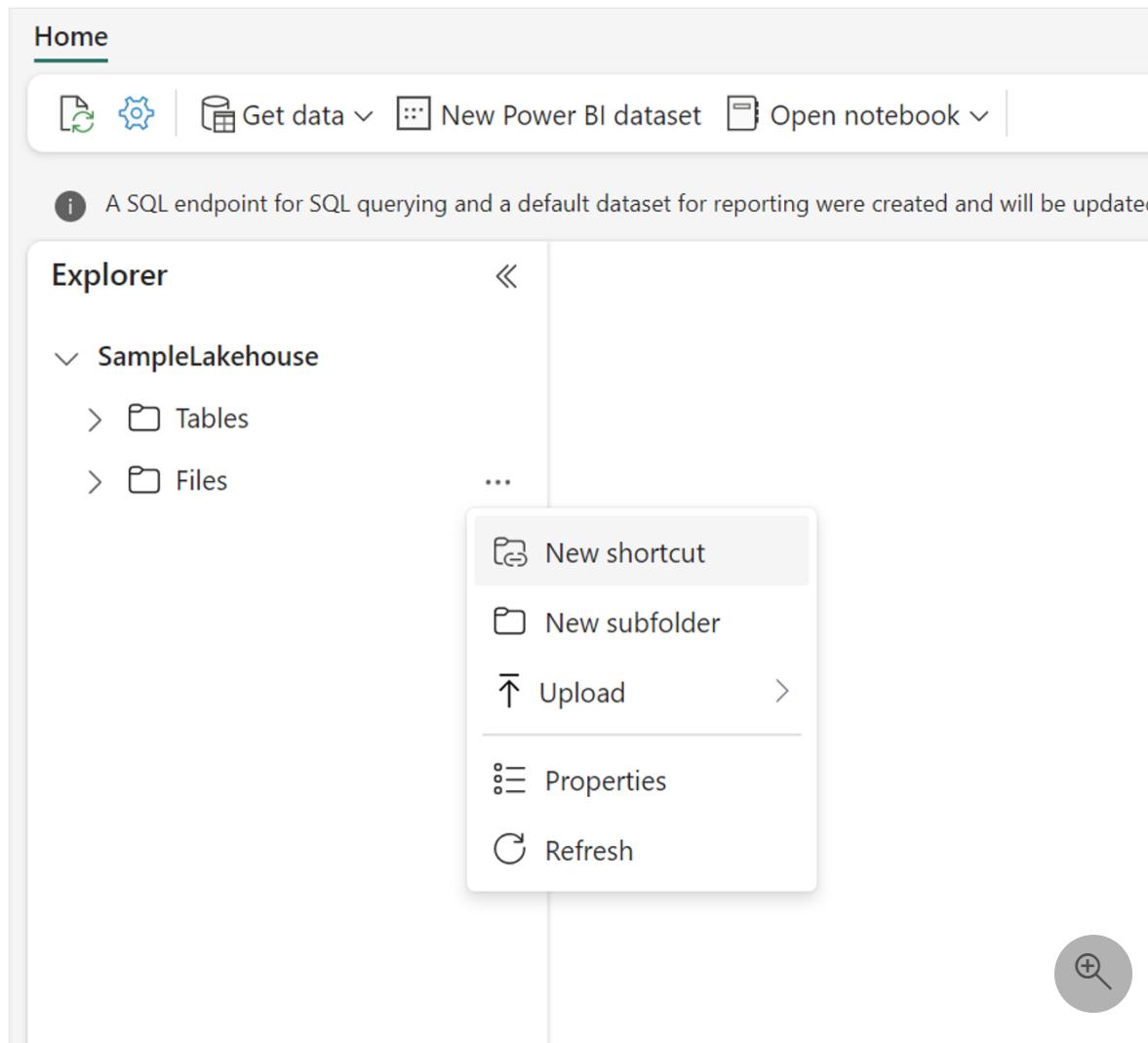
Prerequisite

Create a lakehouse by following these steps: [Creating a lakehouse with OneLake](#).

Create an ADLS shortcut

To create an Azure Data Lake Storage (ADLS) shortcut:

1. Open a lakehouse.
2. Right click on a directory within the **Lake** view of the lakehouse.
3. Select **New shortcut**.



4. Select the ADLS Gen 2 tile.



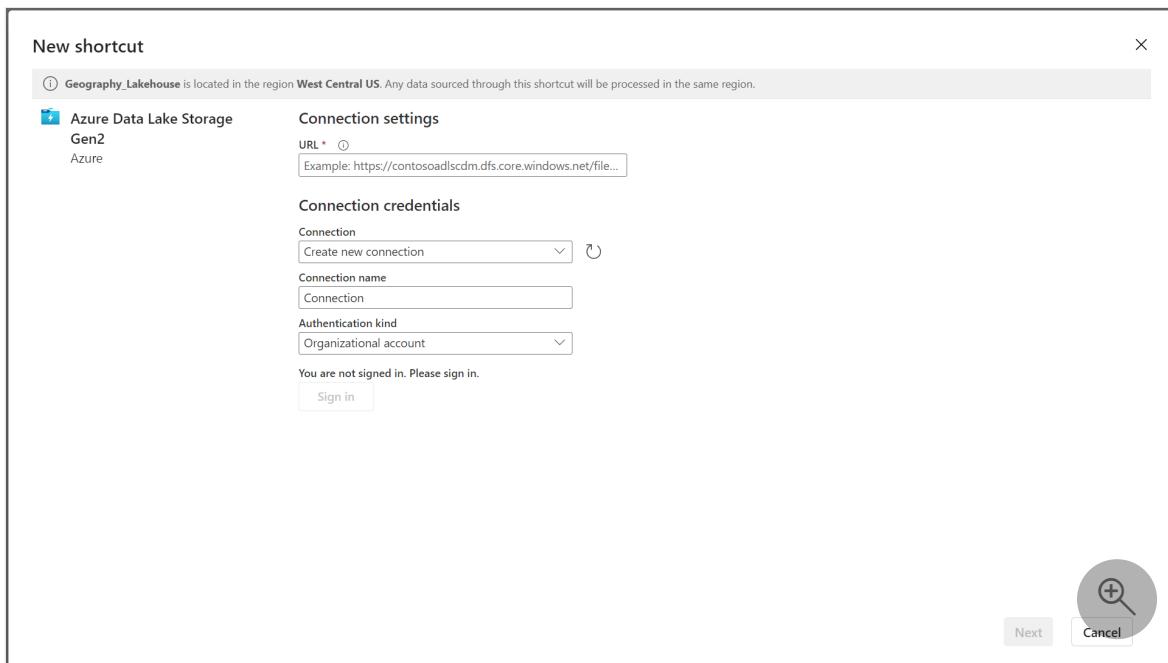
5. Specify the connection details this shortcut will use.

- a. Provide the endpoint for your ADLS account (URL).

⚠ Note

This must be the DFS endpoint for the storage account.

- b. If you have already defined a connection for this storage location, it automatically appears in the connection drop-down. Otherwise, you can choose **Create new connection**.
- c. Enter a **Connection name**.
- d. Select the **Authentication kind** you want to use for the connection.
- e. If you chose **Organizational account**, select the **Sign in** button.
- f. Select **Next**.



6. Specify the shortcut details.

- a. Provide a name for the shortcut.
- b. Provide a path for the shortcut (**Sub Path**). Enter a relative path that starts with a container for the storage account.

⚠ Note

Shortcut paths are case sensitive.

New shortcut

(i) Geography_Lakehouse is located in the region West Central US. Any data sourced through this shortcut will be processed in the same region.

Azure Data Lake Storage Gen2 Azure	Shortcut settings
	Shortcut Name *
<input type="text" value="Shortcut name"/>	
Target Location	
URL	
<input type="text" value="https://myadlsaccount.dfs.core.windows.net"/>	
Sub Path *	
<input type="text" value="/sampledata"/>	

Create
Cancel

7. Select **Create**.

8. See the folder icon with shortcut symbol in the explorer.

Home

Get data New Power BI dataset Open notebook

(i) A SQL endpoint for SQL querying and a default dataset for reporting were created and will be available in the Power BI service.

Explorer

- SampleLakehouse
 - Tables
 - Files
 - ADLS_Shortcut
 - LocalData
 - OneLake_Shortcut
 - S3_Shortcut

Files

Name
ADLS_Shortcut
LocalData
OneLake_Shortcut
S3_Shortcut

Next steps

- [How-to: Create a OneLake shortcut](#)
- [How-to: Create an Amazon S3 shortcut](#)

How-to: Create an Amazon S3 shortcut

Article • 05/23/2023

In this how-to guide, you'll learn how to create an S3 shortcut inside a Fabric Lakehouse. For an overview of shortcuts, see [OneLake shortcuts](#).

Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

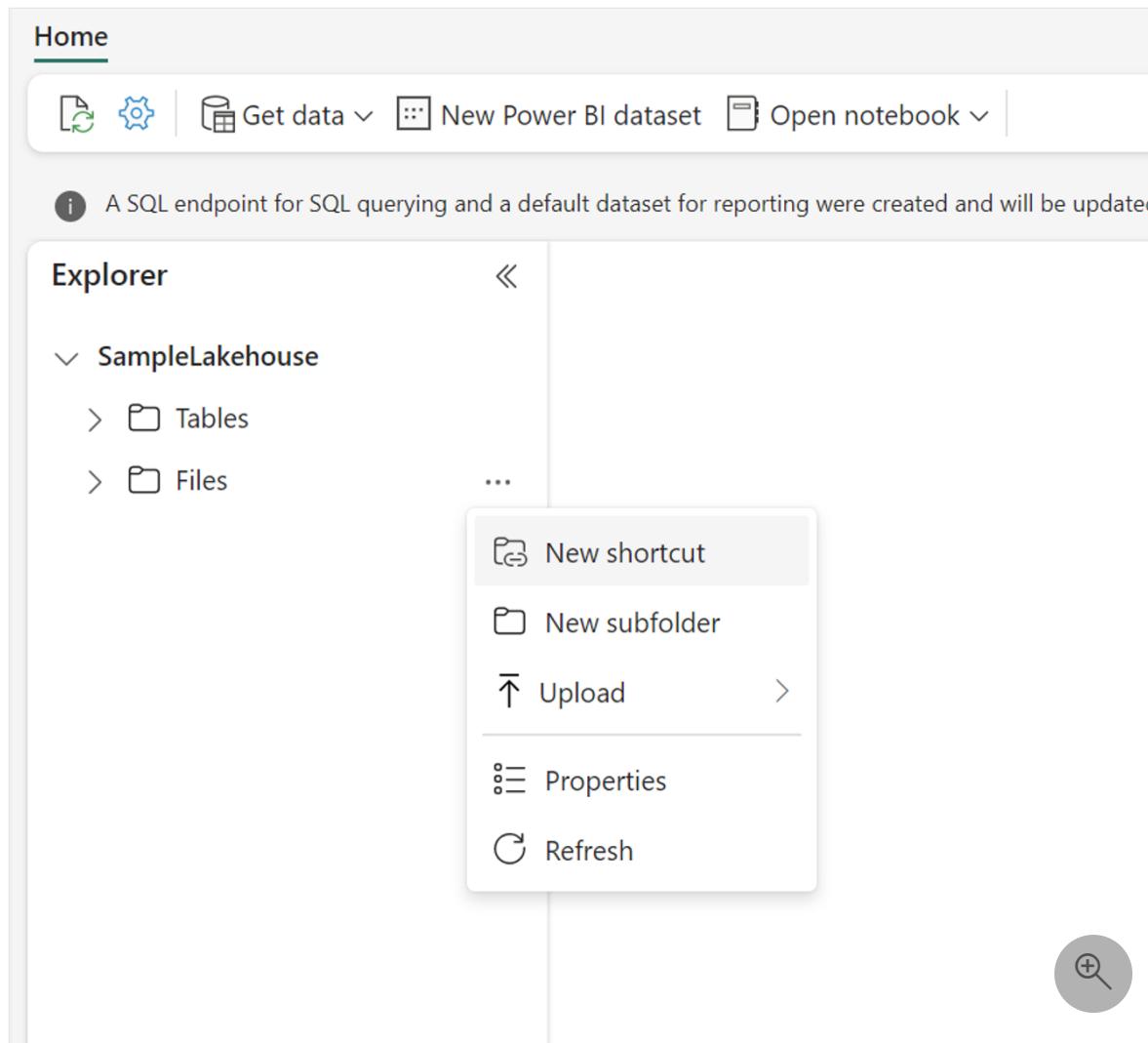
Prerequisite

Create a lakehouse by following these steps: [Creating a lakehouse with OneLake](#).

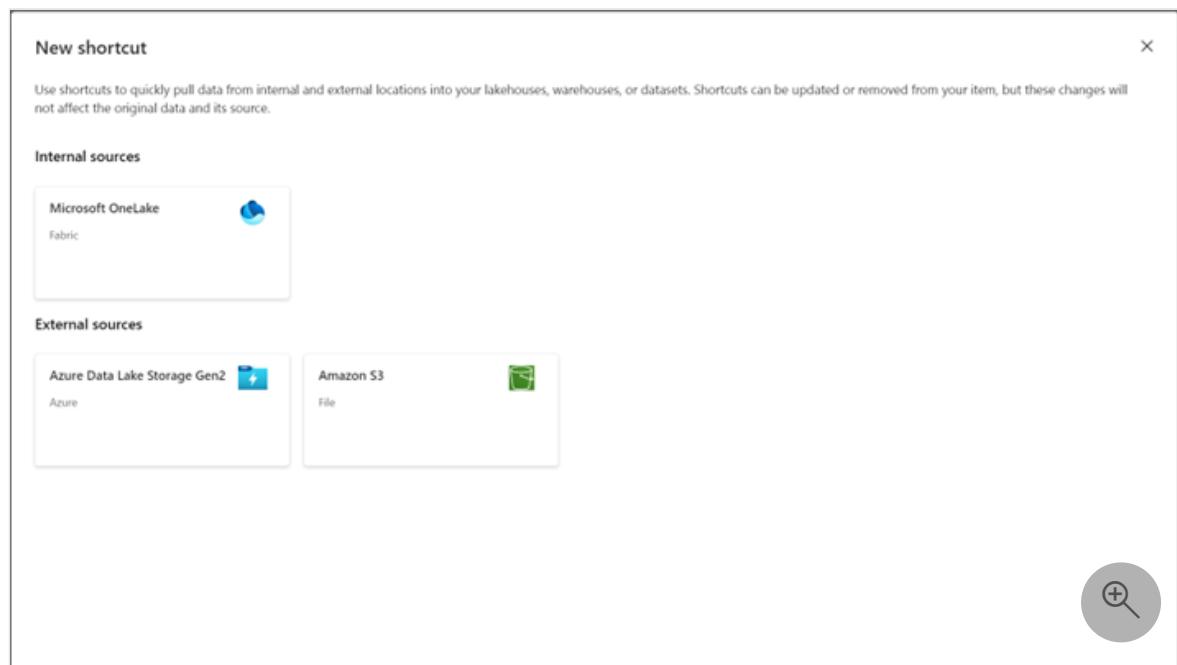
Create an S3 Shortcut

To create an Amazon S3 Shortcut:

1. Open a lakehouse.
2. Right click on a directory within the **Lake** view of the lakehouse.
3. Select **New shortcut**.



4. Select Amazon S3 tile.



5. Specify the connection details this shortcut will use.

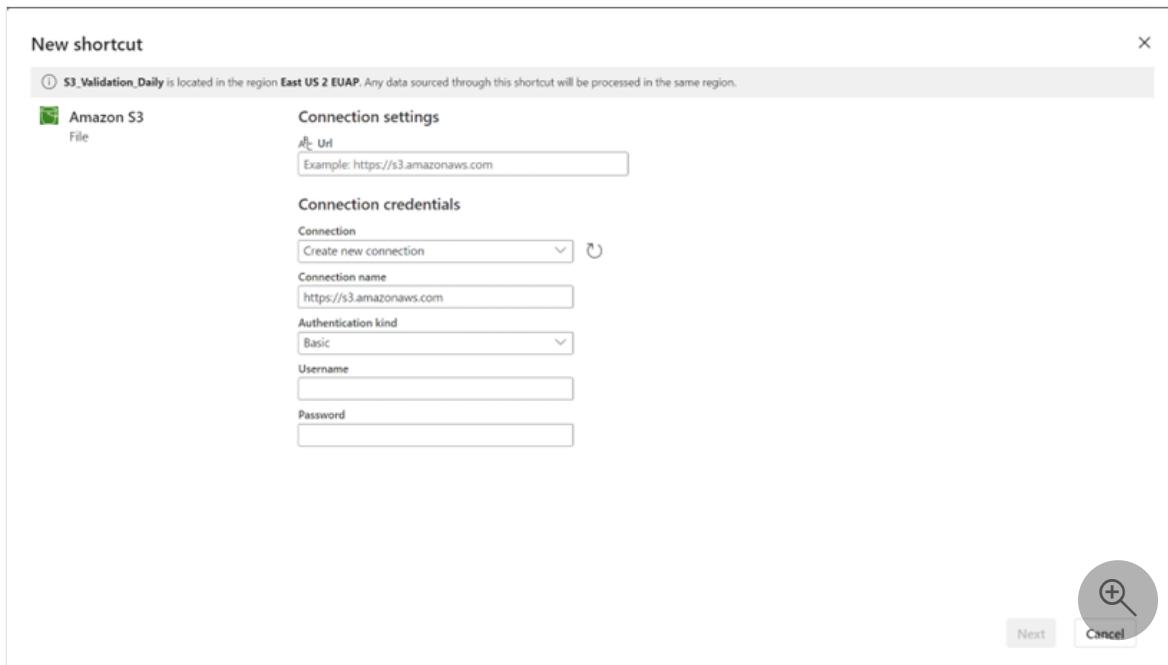
- Proved the endpoint for your S3 account (URL).

⚠ Note

URL must be in the following format

<https://bucketname.s3.region.amazonaws.com/>

- b. If you have already defined a connection for this storage location, it automatically appears in the connection drop-down. Otherwise, you can choose **Create new connection**.
- c. Enter a **Connection name**.
- d. Provide the **Username(Key)** **Password(Secret)** for your IAM user.
- e. Select **Next**.



6. Specify the shortcut details.

- a. Provide a name for the shortcut.
- b. Provide a path for the shortcut (**Sub Path**). Enter a relative path that starts after the bucket name.

⚠ Note

Shortcut paths are case sensitive.

New shortcut

SampleLakehouse is located in the region **East US 2 EUAP**. Any data sourced through this shortcut will be processed in the same region.

 Amazon S3 File Learn more	Shortcut settings Shortcut Name * <input type="text"/> Target Location URL <input type="text" value="s3://myS3bucket"/> Sub Path <input type="text" value="/subfolder"/>
---	--

Create
Cancel

7. Select **Create**.

8. See the folder icon with shortcut symbol in the explorer.

Home

  Get data  New Power BI dataset  Open notebook 

A SQL endpoint for SQL querying and a default dataset for reporting were created and will be available in a few minutes.

Explorer

- ▽ **SampleLakehouse**
 - >  Tables
 - ▽  Files ...
 - >  ADLS_Shortcut
 - >  LocalData
 - >  OneLake_Shortcut
 - >  S3_Shortcut

Files

Name
 ADLS_Shortcut
 LocalData
 OneLake_Shortcut
 S3_Shortcut

Next steps

- [How-to: Create a OneLake shortcut](#)
- [How-to: Create an ADLS Gen2 shortcut](#)

OneLake access and APIs

Article • 05/23/2023

ⓘ Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Overview

Microsoft OneLake provides open access to all of your Fabric items through existing ADLS Gen2 APIs and SDKs. You can access your data in OneLake through any tool compatible with ADLS Gen2 just by using a OneLake URI instead. You can upload data to a lakehouse through Azure Storage Explorer, or read a delta table through a shortcut from Azure Databricks.

As OneLake is software as a service (SaaS), some operations, such as managing permissions or updating items, must be done through Fabric experiences, and can't be done via ADLS Gen2 APIs. A full list of changes to these APIs can be found in the 'Supported API operations' section.

URI Syntax

Because OneLake exists across your entire Microsoft Fabric tenant, you can refer to anything in your tenant by its workspace, item, and path:

HTTP

```
https://onelake.dfs.fabric.microsoft.com/<workspace>/<item>.  
<itemtype>/<path>/<fileName>
```

ⓘ Note

Since item names can be reused across multiple item types, you must specify the item type in the extension. For example, ".lakehouse" for a lakehouse and ".datawarehouse" for a warehouse.

OneLake also supports referencing workspaces and items with GUIDs. OneLake assigns GUIDs and GUIDs don't change, even if the workspace or item name changes. You can find the associated GUID for your workspace or item in the URL on the Fabric portal. You must use GUIDs for both the workspace and the item, and don't need the item type.

HTTP

```
https://onelake.dfs.fabric.microsoft.com/<workspaceGUID>/<itemGUID>/Files/test.csv
```

When adopting a tool for use over OneLake instead of ADLS Gen2, use the following mapping:

- The account name is always 'onelake'.
- The container name is your workspace name.
- The data path starts at the item. For example: '/mylakehouse.lakehouse/Files/'.

OneLake and ADLS Gen2 Parity

While OneLake matches all ADLS Gen2 behavior wherever possible, not every concept in ADLS Gen2 has a direct correlation to OneLake. The following sections describe how OneLake differs from ADLS Gen2, from unsupported request headers to changes in response headers. For more information on ADLS Gen2 APIs, see [Azure Data Lake Storage Gen2 REST APIs](#).

Protected OneLake folders

OneLake doesn't support creating, updating, or deleting workspaces or items through the ADLS Gen2 APIs. Only HEAD calls are supported at the workspace level and account level, as you must make changes to the Fabric tenant and Fabric workspaces in the Fabric administration portal.

OneLake does enforce the Fabric item structure, meaning you can't create, read, update, or delete (CRUD) certain folders, even if you're the item or workspace owner. You must perform these operations via Fabric experiences, such as the Fabric portal or Fabric management APIs. Fabric-managed folders include the top-level folder in an item (for example, `/MyLakehouse.lakehouse`) and the first level of folders within it (for example, `/MyLakehouse.lakehouse/Files` and `/MyLakehouse.lakehouse/Tables`).

You can perform CRUD operations on any folder or file created within these managed folders.

Unsupported request headers and parameters

Even in user-created and owned files and folders, OneLake restricts some management operations through ADLS Gen2 APIs. You cannot update permissions, edit items or workspaces, or set access tiers, as these operations must be managed through Fabric.

OneLake will reject or ignore an API call if it contains a disallowed header or parameter value. OneLake ignores headers if the header doesn't change the behavior of the call, and returns the rejected header in a new 'x-ms-rejected-headers' response header.

OneLake rejects requests containing unallowed query parameters. OneLake doesn't allow the following behaviors and their associated request headers and URI parameters:

- Set access control
 - URI Parameter:
 - action: setAccessControl (Request rejected)
 - action: setAccessControlRecursive (Request rejected)
 - Request headers:
 - x-ms-owner (Header ignored)
 - x-ms-group (Header ignored)
 - x-ms-permissions (Header ignored)
 - x-ms-group (Header ignored)
 - x-ms-acls (Header ignored)
- Set encryption scope
 - Request headers:
 - x-ms-encryption-key (Header ignored)
 - x-ms-encryption-key (Header ignored)
 - x-ms-encryption-algorithm:AES256 (Header ignored)
- Set access tier
 - Request headers:
 - x-ms-access-tier (Header ignored)

Response header differences

Since OneLake uses a different permission model than ADLS Gen2, there are changes to the response headers related to permissions:

- 'x-ms-owner' and 'x-ms-group' always returns '\$superuser', as OneLake doesn't have owning users or groups.
- 'x-ms-permissions' always returns '-----], as OneLake doesn't have owning users, groups, or public access permissions.
- 'x-ms-acl' returns the Fabric permissions for the calling user converted to a POSIX access control list (ACL), in the form 'rwx'

Authorization

You can authenticate OneLake APIs using Microsoft Azure Active Directory (Azure AD) by passing through an authorization header. If a tool supports logging into your Azure account to enable AAD passthrough, you can select any subscription - OneLake only requires your AAD token and doesn't care about your Azure subscription.

Data residency

OneLake doesn't currently guarantee data residency in a particular region when using the global endpoint ('<https://onelake.dfs.fabric.microsoft.com>'). When you query data in a region different than your workspace's region, there's a possibility that data could leave your region during the endpoint resolution process. If you're concerned about data residency, using the correct regional endpoint for your workspace ensures your data stays within its current region and doesn't cross any regional boundaries. You can discover the correct regional endpoint by checking the region of the capacity that the workspace is attached to.

OneLake regional endpoints all follow the same format: '<https://<region>-onelake.dfs.fabric.microsoft.com>'. For example, a workspace attached to a capacity in the West US 2 region would be accessible through the regional endpoint '<https://westus-onelake.dfs.fabric.microsoft.com>'.

Samples

Create file

Request	<code>PUT https://onelake.dfs.fabric.microsoft.com/{workspace}/{item}.</code> <code>{itemtype}/Files/sample?resource=file</code>
Headers	<code>Authorization: Bearer <userAADToken></code>
Response	<code>ResponseCode: 201 Created</code> <code>Headers:</code> <code>x-ms-version : 2021-06-08</code> <code>x-ms-request-id : 272526c7-0995-4cc4-b04a-8ea3477bc67b</code> <code>x-ms-content-crc64 : 0AJ6r0dQWP0=</code> <code>x-ms-request-server-encrypted : true</code> <code>ETag : 0x8DA58EE365</code> <code>Body:</code>

Next steps

- OneLake integration with Azure Synapse Analytics

Integrate OneLake with Azure Synapse Analytics

Article • 05/23/2023

Azure Synapse is a limitless analytics service that brings together enterprise data warehousing and Big Data analytics. This tutorial shows how to connect to OneLake using [Azure Synapse Analytics](#).

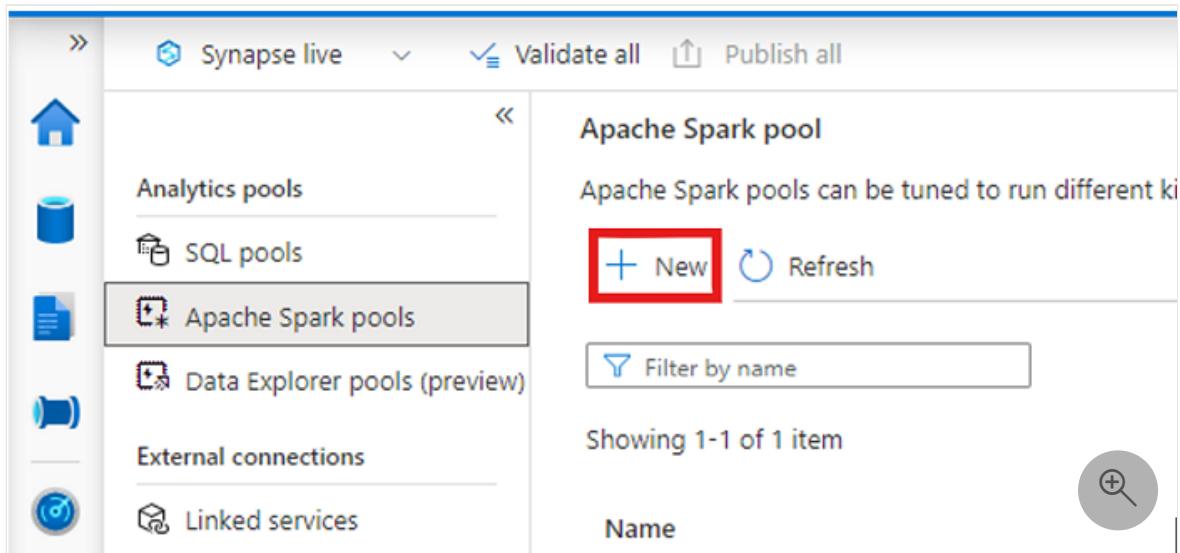
ⓘ Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Write data from Synapse using Apache Spark

Follow these steps to use Apache Spark to write sample data to OneLake from Azure Synapse Analytics.

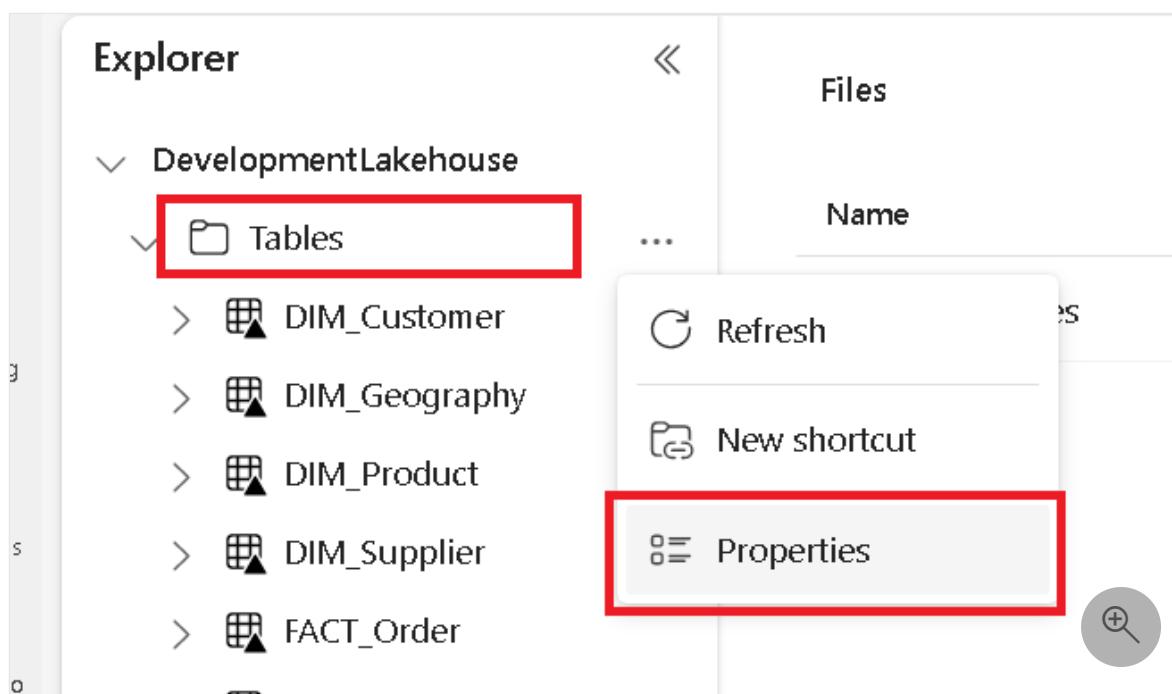
1. Open your Synapse workspace and [create an Apache Spark pool](#) with your preferred parameters.



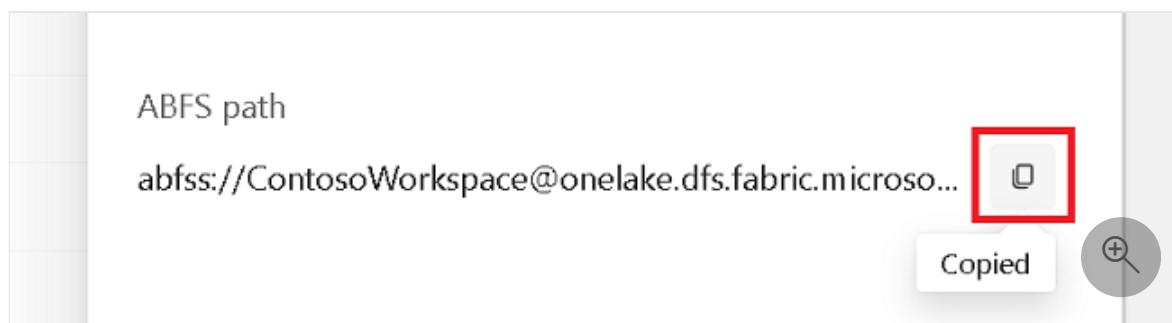
2. Create a new Apache Spark notebook.
3. Open the notebook, set the language to **PySpark (Python)**, and connect it to your newly created Spark pool.

4. In a separate tab, navigate to your Microsoft Fabric lakehouse and find the top-level **Tables** folder.

5. Right click on the **Tables** folder and click **Properties**.



6. Copy the **ABFS path** from the properties pane.



7. Back in the Azure Synapse notebook, in the first new code cell, provide the lakehouse path. This is where your data will be written later. Run the cell.

```
Python

# Replace the path below with the ABFS path to your lakehouse Tables
# folder.
oneLakePath =
'abfss://WorkspaceName@onelake.dfs.fabric.microsoft.com/LakehouseName.1
akehouse/Tables'
```

A screenshot of an Azure Synapse notebook cell. The language is set to 'Python'. The code cell contains the following Python code:

```
# Replace the path below with the ABFS path to your lakehouse Tables
# folder.
oneLakePath =
'abfss://WorkspaceName@onelake.dfs.fabric.microsoft.com/LakehouseName.1
akehouse/Tables'
```

8. In a new code cell, load data from an Azure open dataset into a dataframe. This is the dataset you will load into your lakehouse. Run the cell.

```
Python
```

A screenshot of an Azure Synapse notebook cell. The language is set to 'Python'. The code cell contains the word 'Python'.

```
yellowTaxiDf =  
spark.read.parquet('wasbs://nyctlc@azureopendatastorage.blob.core.windo  
ws.net/yellow/puYear=2018/puMonth=2/*.parquet')  
display(yellowTaxiDf.limit(10))
```

9. In a new code cell, filter, transform, or prep your data. For this scenario, you can trim down your dataset for faster loading, join with other datasets, or filter down to specific results. Run the cell.

Python

```
filteredTaxiDf =  
yellowTaxiDf.where(yellowTaxiDf.tripDistance>2).where(yellowTaxiDf.pass  
engerCount==1)  
display(filteredTaxiDf.limit(10))
```

10. In a new code cell, using your OneLake path, write your filtered dataframe to a new Delta-Parquet table in your Fabric lakehouse. Run the cell.

Python

```
filteredTaxiDf.write.format("delta").mode("overwrite").save(oneLakePath  
+ '/Taxi/')
```

11. Finally, in a new code cell, test that your data was successfully written by reading your newly loaded file from OneLake. Run the cell.

Python

```
lakehouseRead = spark.read.format('delta').load(oneLakePath + '/Taxi/')display(lakehouseRead.limit(10))
```

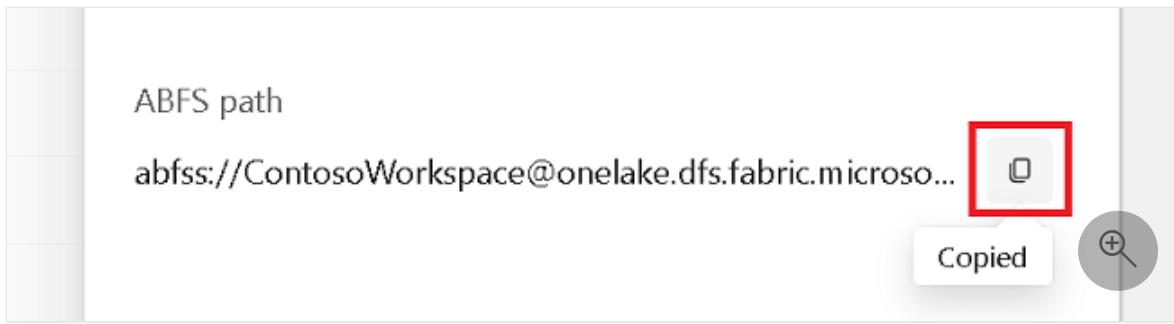
Congratulations! You can now read and write data in OneLake using Apache Spark in Azure Synapse Analytics.

Write data from Synapse using SQL

Follow these steps to use SQL serverless to read data from OneLake from Azure Synapse Analytics.

1. Open a Fabric lakehouse and identify a table that you'd like to query from Synapse.
2. Right-click on the table and click **Properties**.

3. Copy the **ABFS path** for the table.



4. Open your Synapse workspace in [Synapse Studio](#).

5. Create a new SQL script.

6. In the SQL query editor, enter the following query, replacing `ABFS_PATH_HERE` with the path you copied earlier.

```
SQL
SELECT TOP 10 *
FROM OPENROWSET(
    BULK 'ABFS_PATH_HERE',
    FORMAT = 'delta') as rows;
```

7. Run the query to view the top 10 rows of your table.

Congratulations! You can now read data from OneLake using SQL serverless in Azure Synapse Analytics.

Next steps

- [Integrate OneLake with Azure Storage Explorer](#)

Integrate OneLake with Azure Storage Explorer

Article • 05/23/2023

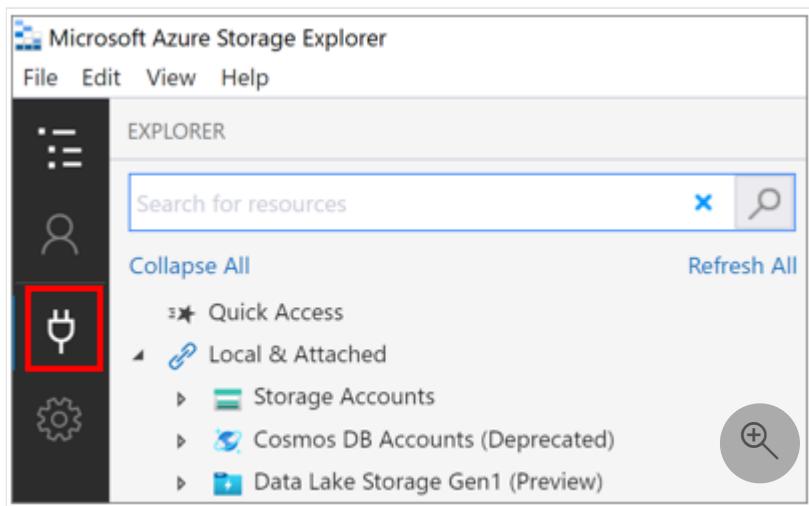
Azure Storage Explorer allows you to view and manage your cloud storage account's contents - upload, download or move files from one location to another. This tutorial shows OneLake integration with Azure Storage Explorer.

ⓘ Important

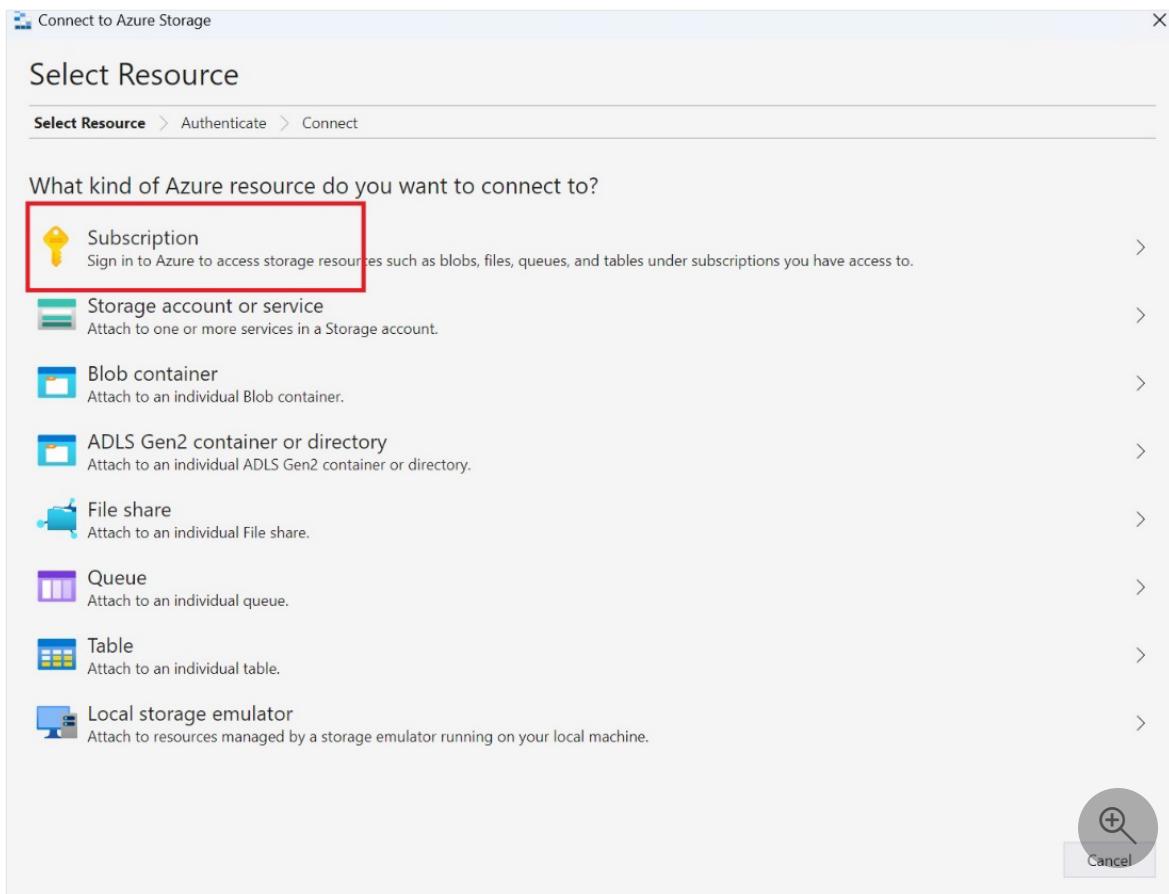
Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Using Azure Storage Explorer

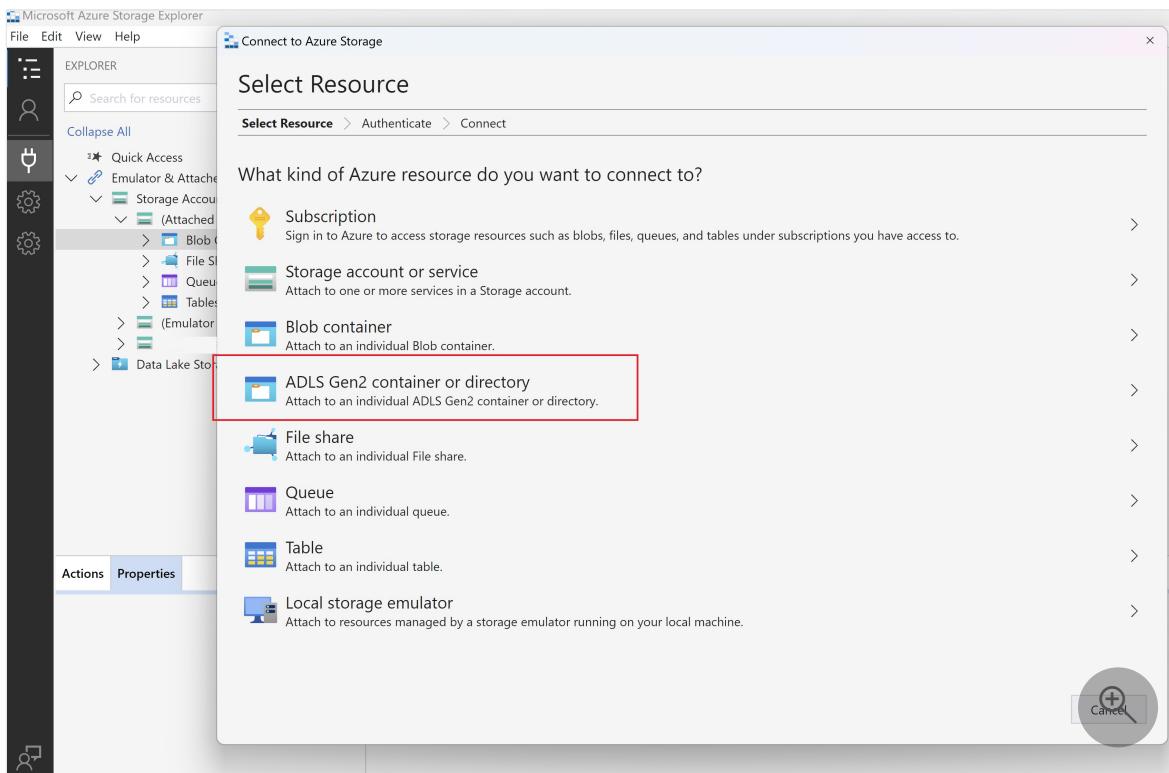
1. Install the latest Azure Storage Explorer bits from the [product webpage](#).
2. Check to ensure the version installed is 1.29.0 or higher. (You can check the version by selecting **Help > About**.)
3. Select the **Open connect dialog** icon.



4. Azure Storage Explorer requires you to sign in to connect to Azure resources. Select **Subscription** and follow the instructions to sign in.

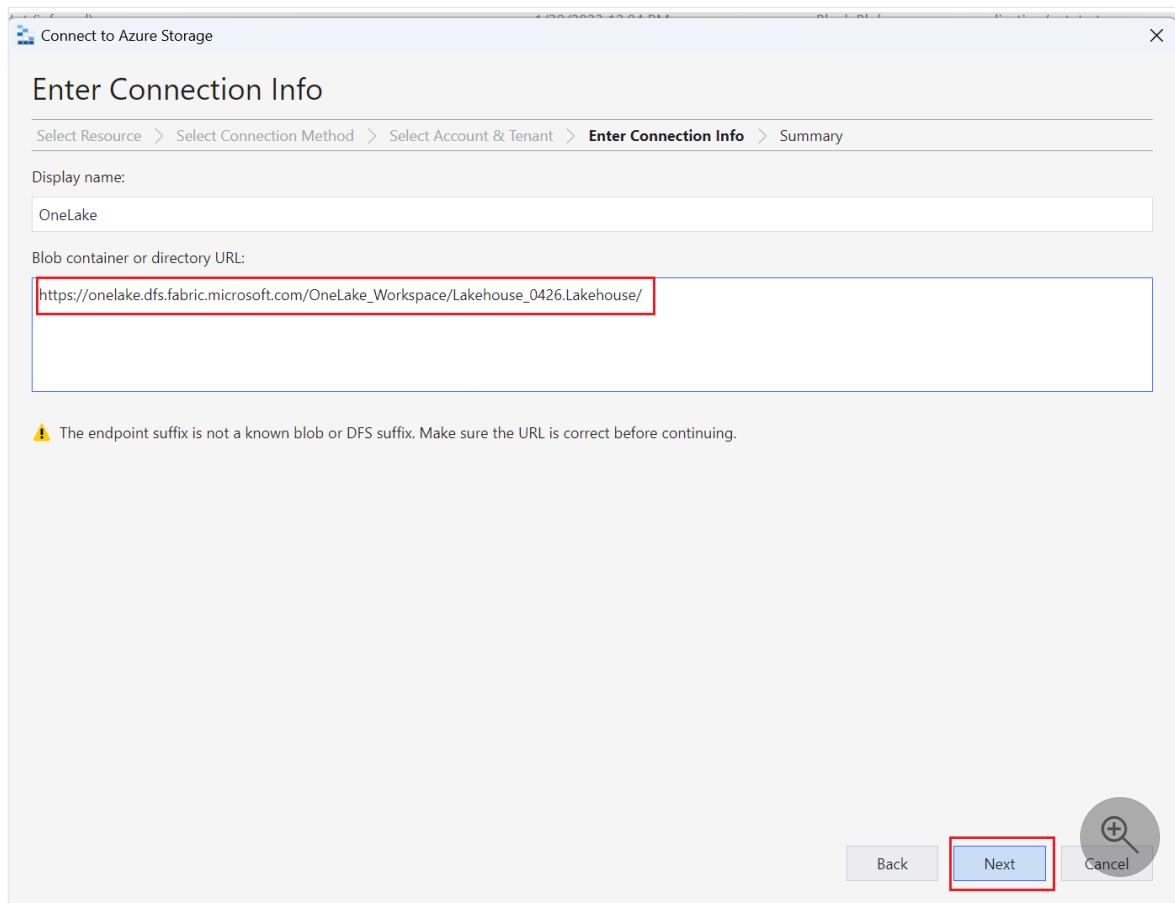


5. Connect to OneLake by selecting the **Open connect dialog** icon again and select **ADLS Gen2 container or directory**.

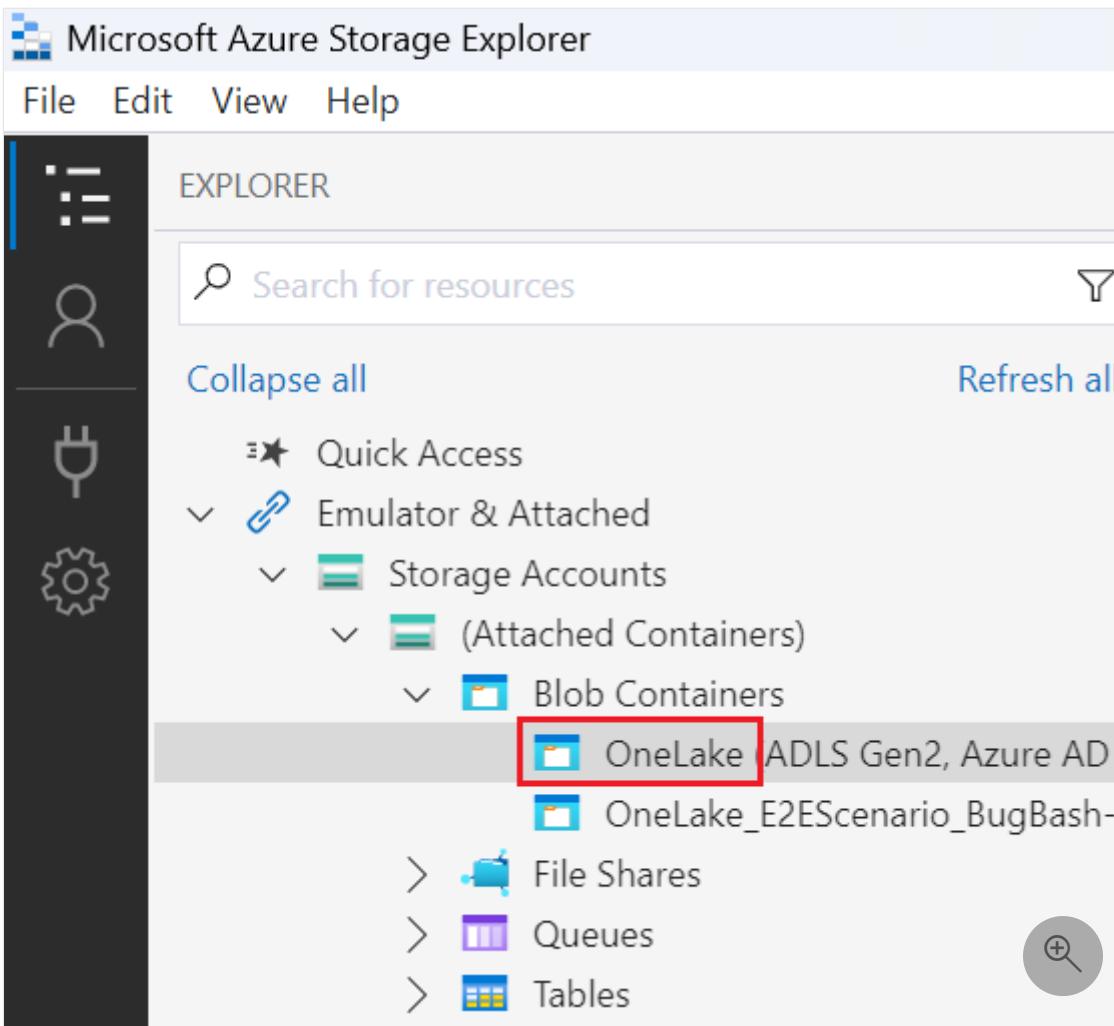


6. Enter URL details of the workspace or item you would like to connect to, in this format: `https://onelake.dfs.fabric.microsoft.com/{workspaceName}/{itemName.itemType}/`. You can find the workspace name and item name in the **Properties** pane of a file in the Microsoft Fabric portal.

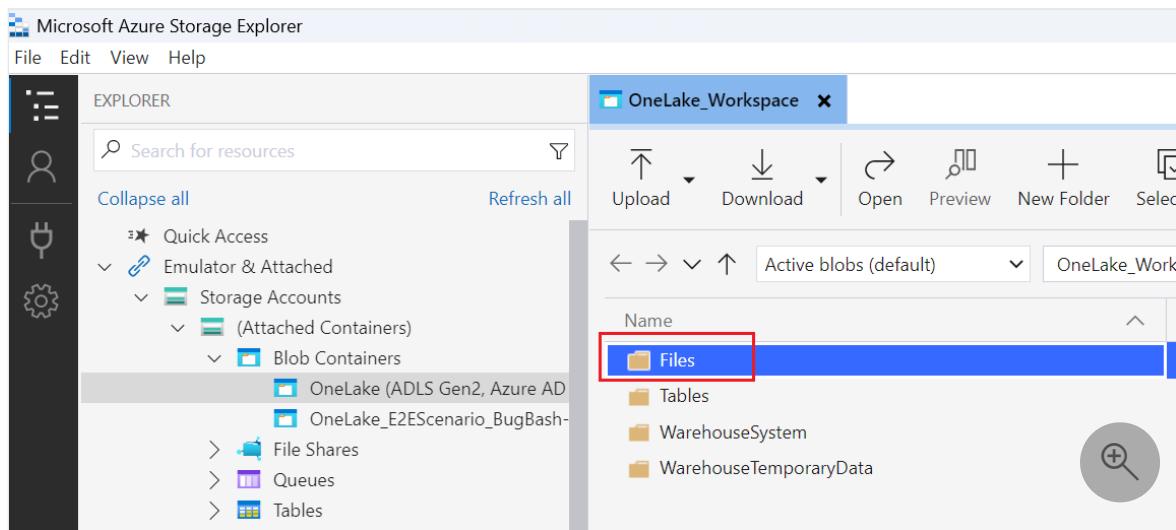
You can choose a **Display name** for convenience, then select **Next**.



7. Storage Explorer browses to the location of the OneLake you entered.



8. To view the contents, select the OneLake folder you connected.



9. Select **Upload**. In the **Select files to upload** dialog box, select the files that you want to upload.

The screenshot shows the OneLake_Workspace interface. At the top is a toolbar with various icons: Upload, Download, Open, Preview, New Folder, Select All, Copy, Paste, and Clone. The 'Upload' and 'Download' buttons are highlighted with a red box. Below the toolbar is a breadcrumb navigation bar showing 'OneLake_Workspace > Lakehouse_0426.Lakehouse'. The main area is a file browser table with columns for Name, Access Tier, and Access. The table contains several entries: 'Fold1' (Folder), 'TestFold1' (Folder), 'Campaign.csv' (File), 'PhotoFact1.parquet' (File), 'PhotoFact11.parquet' (File), and 'test11.csv' (File). All files are listed under the 'Hot (inferred)' access tier. A magnifying glass icon is located in the bottom right corner of the table area.

Name	Access Tier	Access
Fold1		
TestFold1		
Campaign.csv		Hot (inferred)
PhotoFact1.parquet		Hot (inferred)
PhotoFact11.parquet		Hot (inferred)
test11.csv		Hot (inferred)

10. To download, select the folders or files that you want to download and then select **Download**.

11. To copy data across locations, select the folders you want to copy and select **Copy**, then navigate to the destination location and select **Paste**.

The screenshot shows the OneLake_Workspace interface. The top toolbar includes 'Upload', 'Download', 'Open', 'Preview', 'New Folder', 'Select All', 'Copy', 'Paste', 'Clone', and 'Rename'. The 'Copy' and 'Paste' buttons are highlighted with a red box. The breadcrumb navigation bar shows 'OneLake_Workspace > Lakehouse_0426.Lakehouse > Files'. The file browser table lists the same set of files as the previous screenshot: 'Fold1', 'TestFold1', 'Campaign.csv', 'PhotoFact1.parquet', 'PhotoFact11.parquet', and 'test11.csv', all under 'Hot (inferred)' access tier. A magnifying glass icon is in the bottom right.

Name	Access Tier	Access
Fold1		
TestFold1		
Campaign.csv		Hot (inferred)
PhotoFact1.parquet		Hot (inferred)
PhotoFact11.parquet		Hot (inferred)
test11.csv		Hot (inferred)

Limitations

- If a workspace name has capital letters, deletion of files or folders fails due to restriction from storage service. We recommend using your workspace name in lowercase letters.

Next steps

- Integrate OneLake with Azure Databricks

Integrate OneLake with Azure Databricks

Article • 05/23/2023

ⓘ Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

This scenario shows how to connect to OneLake via Azure Databricks. After completing this tutorial, you'll be able to read and write to a Microsoft Fabric Lakehouse from your Azure Databricks workspace.

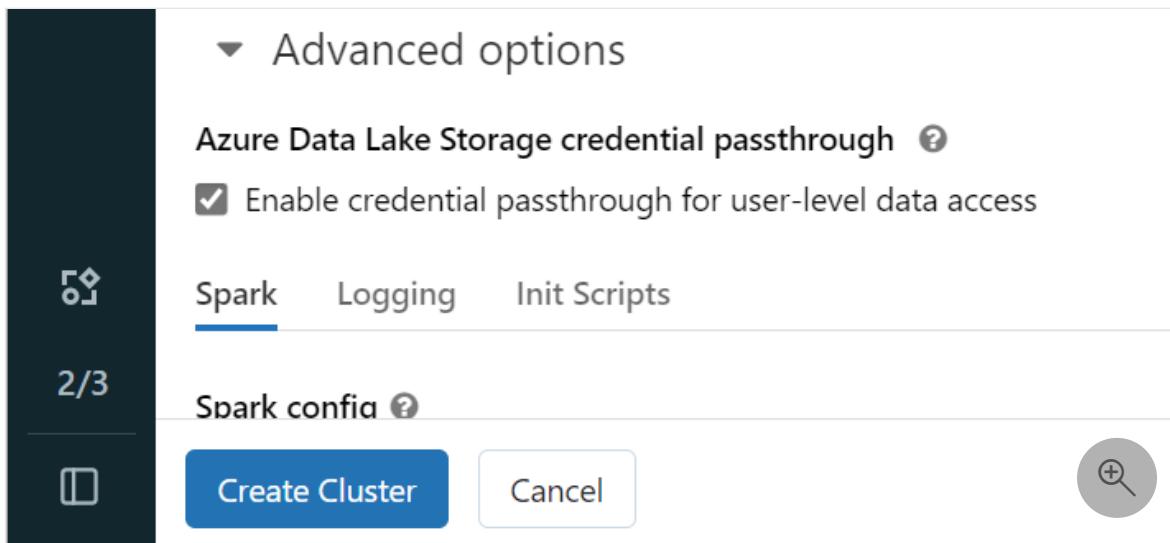
Prerequisites

Before you connect, complete these steps:

- A Fabric workspace and lakehouse.
- A premium Azure Databricks workspace. Only premium Azure Databricks workspaces support Microsoft Azure Active Directory credential passthrough, which is required for this scenario.

Set up your Databricks workspace

1. Open your Azure Databricks workspace and select **Create > Cluster**.
2. To authenticate to OneLake with your Azure AD identity, you must enable Azure Data Lake Storage credential passthrough on your cluster in the Advanced Options.



ⓘ Note

You can also connect Databricks to OneLake using a service principal. For more information about authenticating Azure Databricks using a service principal, see [Service principals for Azure Databricks automation](#).

3. Create the cluster with your preferred parameters. For more information on creating a Databricks cluster, see [Configure clusters - Azure Databricks](#).
4. Open a notebook and connect it to your newly created cluster.

Author your notebook

1. Navigate to your Fabric lakehouse and copy the ABFS path to your lakehouse. You can find it in the **Properties** pane.

ⓘ Note

Azure Databricks only supports the Azure Blob Filesystem (ABFS) driver when reading and writing to Azure Data Lake Storage (ADLS) Gen2 and OneLake:
abfss://myWorkspace@onelake.dfs.fabric.microsoft.com/

2. Save the path to your lakehouse in your Databricks notebook. This lakehouse is where you'll write your processed data later:

Python

```
oneLakePath =  
'abfss://myWorkspace@onelake.dfs.fabric.microsoft.com/myLakehouse.lakeh
```

```
ouse/Files/ '
```

3. Load data from a Databricks public dataset into a dataframe. You can also read a file from elsewhere in Fabric or choose a file from another ADLS Gen2 account you already own.

Python

```
yellowTaxiDF = (spark.read.format("csv").option("header",  
"true").option("inferSchema", "true").load("/databricks-  
datasets/nyctaxi/tripdata/yellow/yellow_tripdata_2019-12.csv.gz")
```

4. Filter, transform, or prep your data. For this scenario, you can trim down your dataset for faster loading, join with other datasets, or filter down to specific results.

Python

```
filteredTaxiDF =  
yellowTaxiDF.where(yellowTaxiDF.fare_amount<4).where(yellowTaxiDF.passe  
nger_count==4)  
display(filteredTaxiDF)
```

5. Write your filtered dataframe to your Fabric Lakehouse using your OneLake path.

Python

```
filteredTaxiDF.write.format("csv").mode("overwrite").csv(oneLakePath)
```

6. Test that your data was successfully written by reading your newly loaded file.

Python

```
lakehouseRead = spark.read.format('csv').load(oneLakePath)  
display(lakehouseRead.limit(10))
```

Congratulations! You can now read and write data in Fabric using Azure Databricks.

Next steps

- [Integrate OneLake with Azure HDInsight](#)

Integrate OneLake with Azure HDInsight

Article • 05/23/2023

Azure HDInsight is a managed cloud-based service for big data analytics that helps organizations process large amounts of data. This tutorial shows how to connect to OneLake with a Jupyter notebook from an Azure HDInsight cluster.

ⓘ Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Using Azure HDInsight

To connect to OneLake with a Jupyter notebook from an HDInsight cluster:

1. Create an HDI Spark cluster. Follow these instructions: [Set up clusters in HDInsight](#).
 - a. While providing cluster information, remember your Cluster login Username and Password, as you will need them later to access the cluster.
 - b. Create a user assigned managed identity (UAMI): [Create for Azure HDInsight - UAMI](#) and choose it as the identity in the **Storage** screen.

The screenshot shows the 'Create HDInsight cluster' wizard with the 'Storage' tab selected. The 'Primary storage type' is set to 'Azure Data Lake Storage Gen2'. Under 'Primary storage account', a dropdown menu shows 'Select an existing storage account' with the option 'test11111111-2023-02-07t20-15-10-425z' highlighted with a green checkmark. In the 'Identity' section, there is a dropdown for 'User-assigned managed identity' which is currently empty. A red box highlights this dropdown. A magnifying glass icon is located in the bottom right corner of the form area.

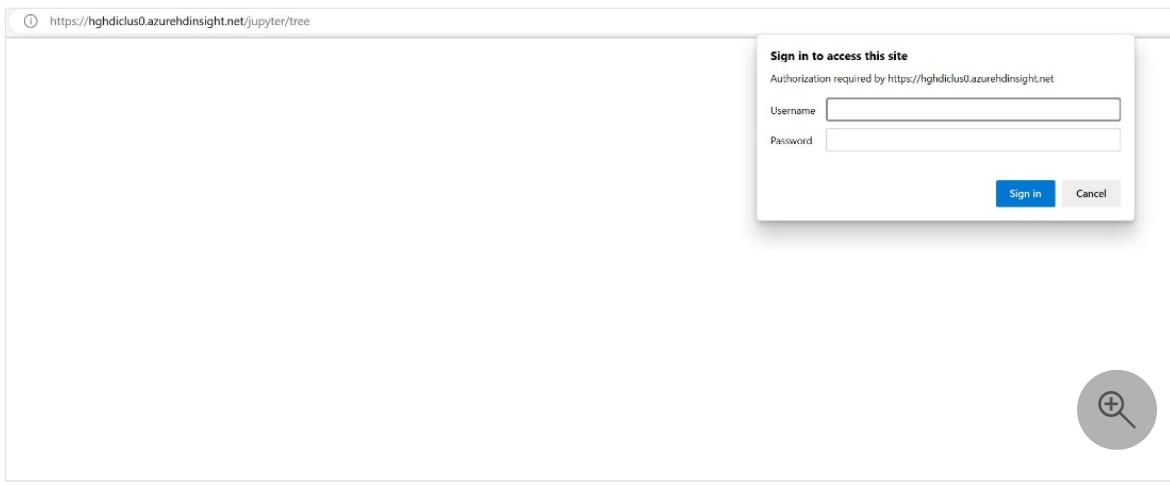
2. Give this UAMI access to the Fabric workspace that contains your items. Learn more about Fabric role-based access control (RBAC): [Workspace roles](#) to decide what role is suitable.

The screenshot shows the Microsoft Fabric Home page. On the left, there's a sidebar with options like 'Home', 'Create', 'Workspaces', and a list of Lakehouses and Notebooks. The main area has a heading 'Load data in your lakehouse' and three cards: 'New Dataflow Gen2', 'New data pipeline', and 'Open notebook'. On the right, a 'Manage access' modal is open for the workspace 'OneLake_Worksplace'. It includes a search bar, a list of users, and a button to 'Add people or groups'. A specific user entry, 'hghdiwest2MI Member', is highlighted with a red box.

3. Navigate to your Lakehouse and find the Name for your workspace and Lakehouse. You can find them in the URL of your Lakehouse or the **Properties** pane for a file.
4. In the Azure portal, look for your cluster and select the notebook.

The screenshot shows the Azure portal's 'HDInsight cluster' overview for 'hghdiclus0'. The left sidebar includes sections for Overview, Activity log, Tags, Diagnose and solve problems, Settings, Monitoring, Automation, and Support + troubleshooting. The main area displays cluster details like Resource group, Status, Location, and Subscription. It also shows 'Recommended features' such as Auto scale, Applications (with a red box around the 'Jupyter notebook' icon), Script actions, and Monitor integration. The 'Monitoring' section includes Insights (Preview), Alerts, Metrics, Diagnostic settings, Monitor integration, Log (Preview), and Workbooks (Preview). The 'Automation' section includes Tasks (preview) and Export template. The 'Support + troubleshooting' section includes Resource health.

5. Enter the credential information you provided while creating the cluster.



6. Create a new Spark Notebook.

7. Copy the workspace and Lakehouse names into your notebook and build your OneLake URL for your Lakehouse. Now you can read any file from this file path.

Python

```
fp = 'abfss://' + 'Workspace Name' +  
    '@onelake.dfs.fabric.microsoft.com/' + 'Lakehouse Name' + '/Files/'  
df = spark.read.format("csv").option("header", "true").load(fp +  
    "test1.csv")  
df.show()
```

8. Try writing some data into the Lakehouse.

Python

```
writecsvdf = df.write.format("csv").save(fp + "out.csv")
```

9. Test that your data was successfully written by checking in your Lakehouse or by reading your newly loaded file.

You can now read and write data in OneLake using your Jupyter notebook in an HDI Spark cluster.

Next steps

- [OneLake security](#)

Manage OneLake with PowerShell

Article • 05/23/2023

ⓘ Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Connecting to OneLake with Azure PowerShell

Connect to OneLake from PowerShell by following these steps:

1. Install the Azure Storage PowerShell module.

```
PowerShell  
Install-Module Az.Storage -Repository PSGallery -Force
```

2. Sign in to your Azure account.

```
PowerShell  
Connect-AzAccount
```

3. Create the storage account context.

- a. Storage account name is 'onelake'.
- b. Set '-UseConnectedAccount' to passthrough your Azure credentials.
- c. Set '-endpoint' as 'fabric.microsoft.com'.

4. Run the same commands used for ADLS Gen2. For more information about ADLS Gen 2 and the Azure Storage PowerShell module, see [Use PowerShell to manage ADLS Gen2](#).

Example: Get size of an item or directory

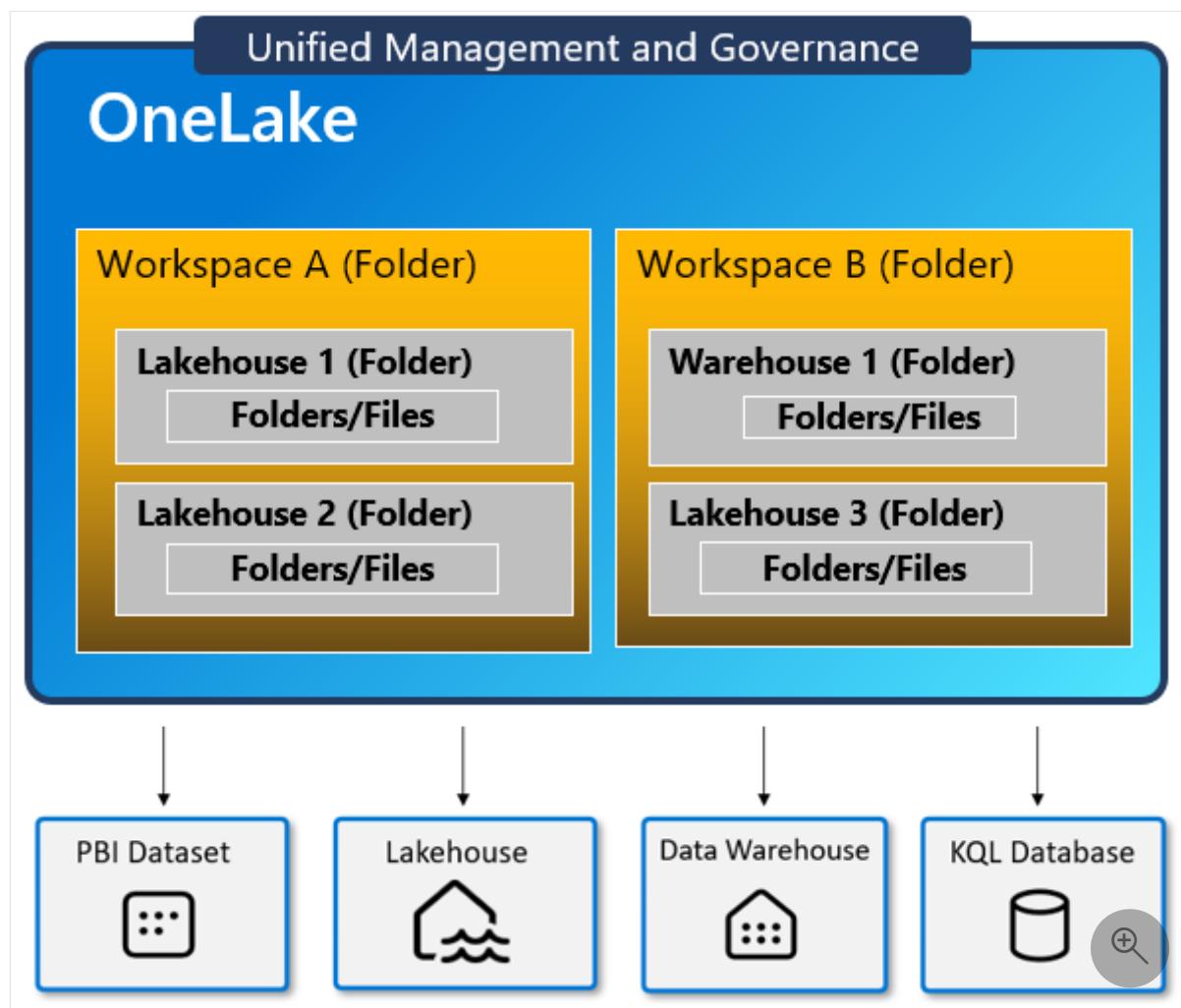
```
PowerShell  
Install-Module Az.Storage -Repository PSGallery -Force  
Connect-AzAccount
```

```
$ctx = New-AzStorageContext -StorageAccountName 'onelake' -  
UseConnectedAccount -endpoint 'fabric.microsoft.com'  
  
# This example uses the workspace and item name. If the workspace name does  
not meet Azure Storage naming criteria (no special characters), you can use  
GUIDs instead.  
$workspaceName = 'myworkspace'  
$itemPath = 'mylakehouse.lakehouse/Files'  
  
# Recursively get the length of all files within your lakehouse, sum, and  
convert to GB.  
$colitems = Get-AzDataLakeGen2ChildItem -Context $ctx -FileSystem  
$workspaceName -Path $itemPath -Recurse -FetchProperty | Measure-Object -  
property Length -sum  
"Total file size: " + ($colitems.sum / 1GB) + " GB"
```

OneLake security

Article • 05/23/2023

OneLake uses a layered security model built around the organizational structure of components within Microsoft Fabric. Security is derived from Azure Active Directory (Azure AD) authentication and is compatible with user identities, service principals, and managed identities. Using Azure AD and Fabric components, you can build out robust security mechanisms across OneLake, ensuring that you keep your data safe while also reducing copies and minimizing complexity.



ⓘ Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Workspace security

The workspace is the primary security boundary for data within OneLake. Each workspace represents a single domain or project area where teams can collaborate on data. Security in the workspace is managed through Fabric workspace roles. Learn more about Fabric role-based access control (RBAC): [Workspace roles](#)

Workspace roles in Fabric grant the following permissions in OneLake.

Capability	Admin	Member	Contributor	Viewer
View files in OneLake	Yes	Yes	Yes	No
Write files in OneLake	Yes	Yes	Yes	No

Compute-specific security

Some compute engines in Fabric have their own security models. For example, Fabric Warehouse lets users define access using T-SQL statements. Compute-specific security is always enforced when you access data using that engine, but those conditions may not apply to users in certain Fabric roles when they access OneLake directly. See the documentation for Warehouse, Real-time analytics, and Power BI datasets for more details on what types of compute security can be defined.

As a rule, users in the Viewer role can only access data through select compute engines and any security rules defined in those engines apply. All other roles have direct OneLake access, allowing them to query data through Spark, APIs, or a OneLake File Explorer. However, compute-specific security still applies to those users when accessing data through that compute engine.

Example: Martha is an administrator for a Fabric workspace and Pradeep is a Viewer. Martha wants to restrict access to certain tables in LakehouseA. She connects to SQL and defines object level security using GRANT and DENY statements. When Pradeep accesses the data through SQL he is only able to see the tables from LakehouseA that he was granted access to.

Shortcut security

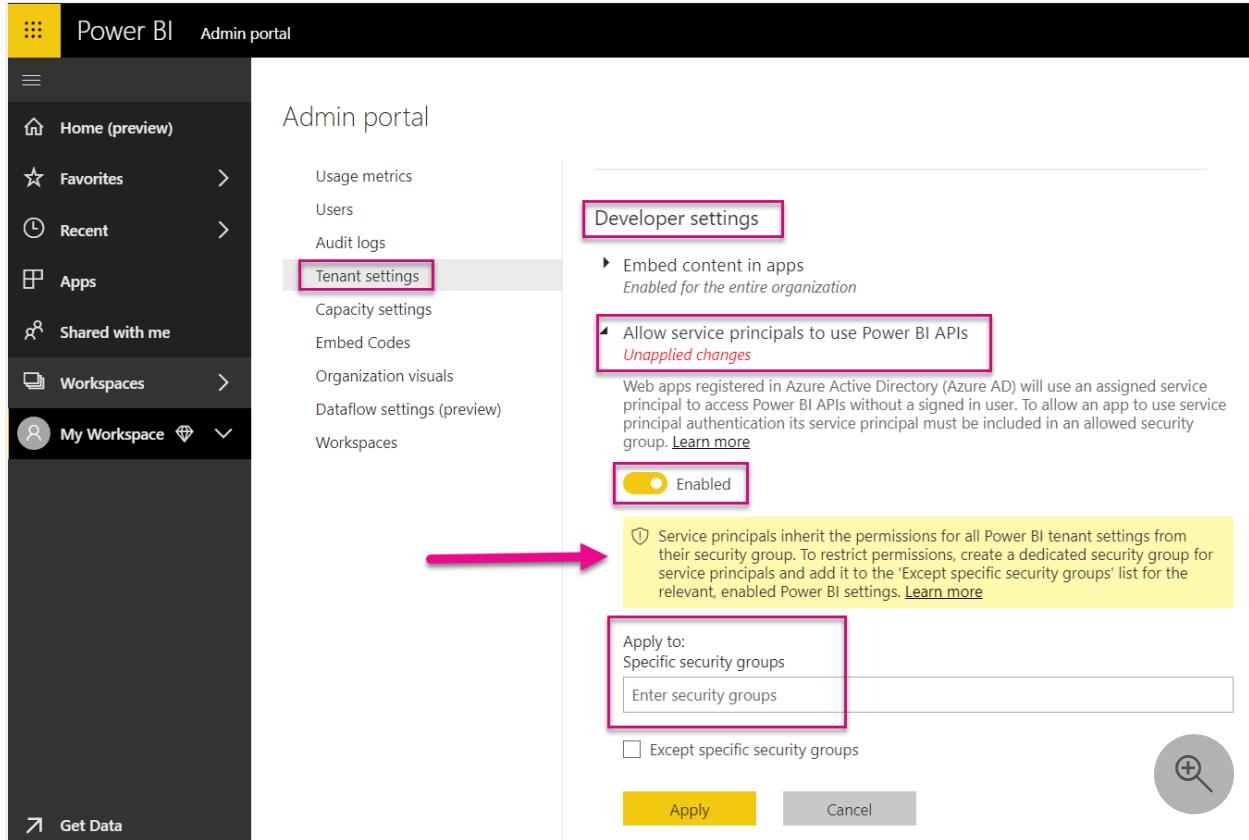
Shortcuts in Microsoft Fabric allow for greatly simplified data management, but have some security considerations to note. For information on managing shortcut security see this [document](#).

Authentication

OneLake uses Azure Active Directory (Azure AD) for authentication; you can use it to give permissions to user identities and service principals. OneLake automatically extracts the user identity from tools, which use Azure AD authentication and map it to the permissions you set in the Fabric portal.

ⓘ Note

To use service principals in a Fabric tenant, a tenant administrator must enable Service Principal Names (SPNs) for the entire tenant or specific security groups.



Private links

Fabric doesn't currently support private link access to OneLake data via non-Fabric products and Spark.

Allow apps running outside of Fabric to access data via OneLake

OneLake provides the ability to restrict access to data from applications running outside of Fabric environments. Admins can find the setting in the tenant admin portal. When this switch is turned ON, data can be accessed via all sources. When this switch is

turned OFF, data cannot be accessed via applications running outside of Fabric environments. For example, data can be access via applications like Azure Databricks, custom applications using ADLS APIs, or OneLake file explorer.

Next steps

- [OneLake file explorer](#)
- [Workspace roles](#)

Get started with securing data in OneLake

Article • 05/23/2023

As a single data lake for your entire organization, it's critical to implement a scalable and robust security model in OneLake to keep sensitive data compartmentalized. Microsoft OneLake and Microsoft Fabric provide several out of the box capabilities to keep data access restricted to only those users that need it. This article takes a look at how to best secure your data estate with the current capabilities in OneLake.

Important

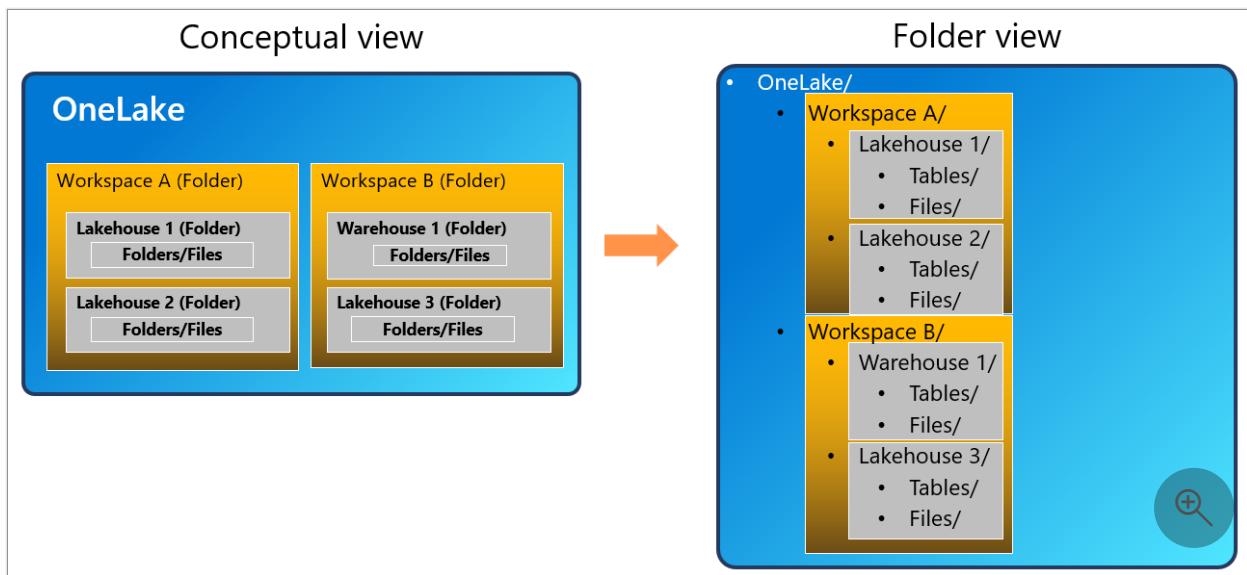
Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

OneLake structure

OneLake is a hierarchical data lake, similar to ADLS gen 2 or the Windows file system. This structure allows for security to be set at different levels in the hierarchy to govern access. OneLake offers more features and controls at certain levels in the folder hierarchy. These levels are:

Workspace: a collaborative environment that is used to create and manage items.

Item: a set of capabilities bundled together into single component. A data item is a subtype of item that allows for data to be stored within it using OneLake. Items always live within workspaces and workspaces always live directly under the OneLake namespace. This structure can be visualized as follows:



Workspace permissions

In the public preview release of Microsoft Fabric, permissions can only be configured at the workspace level.

To grant a user access to an item, users need permissions from one of the Fabric [workspace roles](#). At a minimum, users need Read access on an item to see and connect to that item, which is provided by the Viewer role. Users in the Viewer role can also connect to and read data using SQL endpoints for the Warehouse and Lakehouse items or read data through Power BI datasets.

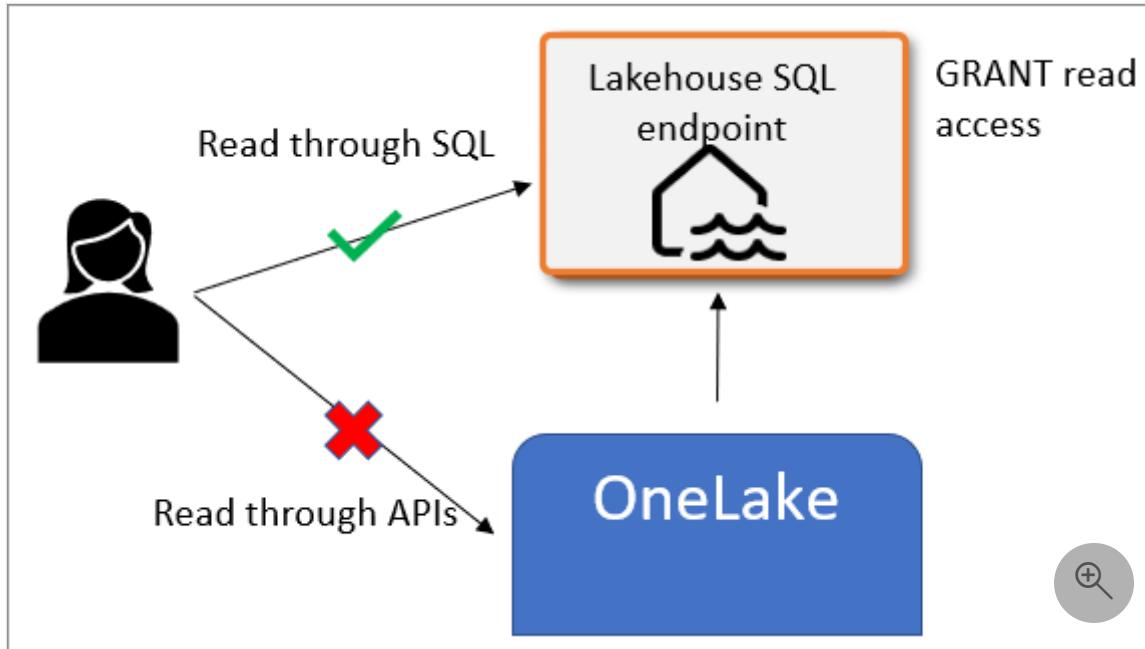
Access directly to OneLake or to write data is provided through the other roles. Admin, member, and contributor all provide access to read data directly in OneLake through Spark or APIs, and write data to those sources. The Warehouse item is read-only through the lake interface, so even Admins are not able to write data to a Warehouse through APIs, but they can write data through SQL.

Compute permissions

In addition to the workspace permissions, data access can be given through the SQL compute engine in Microsoft Fabric. The access granted through SQL only applies to users accessing data through SQL, but this security can be used to give more selective access to certain users. In its current state, SQL supports restricting access to specific tables and schemas with row level security planned in a future release.

In the below example, a user is shared a Lakehouse but with only Viewer access. They are then granted SELECT through the SQL endpoint. When that user tries to write data

through the OneLake APIs the access gets denied since they don't have sufficient permissions, but reads made through SQL SELECT statements would succeed.



Securing OneLake

Now that we understand the permissions available in Microsoft Fabric, let us look at an example of how to best structure data in OneLake. To start, we build a standard medallion architecture. In this approach, we typically want to have a limited set of users that have access to the Bronze and Silver layers, with broader access to the Gold layer. One way to structure that is as follows:



The people responsible for managing Bronze and Silver can be added to Member or Contributor roles so that they can update and manage all the data in those

environments. Since those users need write access this is currently the only method to accomplish this. Users that need access to specific data items within the Bronze and Silver layer can be given the Viewer role and access data through SQL endpoints.

For the Gold layer, access can be divided up across a number of smaller workspaces. Each workspace can be scoped to a business domain or set of users that would need to access that data. Within each workspace, end users can be given the Viewer role. Data engineers that build and manage the Gold layer can utilize the Contributor or Member role which gives them Write access. If a specific environment needs more stringent access controls, specific Warehouses or Lakehouses can define object level security through their SQL endpoints. This allows for only some tables to be shared with users while others are hidden.

The example above is only one of many ways that data can be structured in OneLake, however it provides recommendations for how to leverage the capabilities of Microsoft Fabric to secure data. In the next section we will look at some general guidance for applying security.

General guidance

The following general rules can be used to guide structuring data in OneLake to keep it secure.

Write access: Users that need write access must be part of a workspace role that grants write access. This applies to all data items, so scope workspaces to a single team of data engineers.

Lake access: To give users direct read access to data in OneLake they need to be part of the Admin, Member, or Contributor workspace roles.

General data access: Any user with Viewer permissions can access data through the SQL endpoint for warehouses, lakehouses, and datasets.

Object level security: To protect sensitive data, give users access to a Warehouse or Lakehouse SQL endpoint through the Viewer role and use SQL DENY statements to restrict access to certain tables.

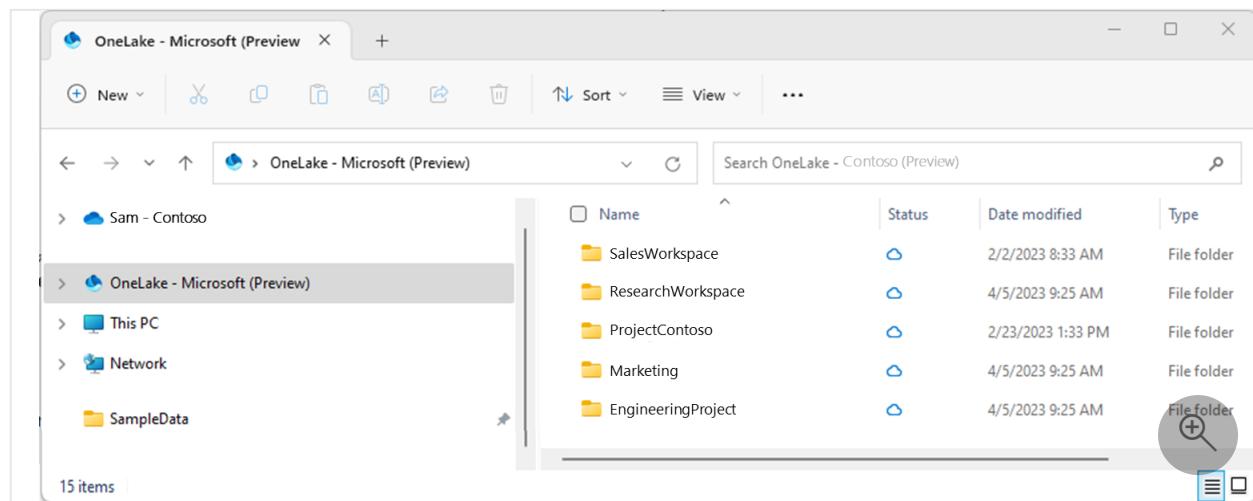
Next steps

- [Workspace roles](#)
- [OneLake security](#)
- [OneLake file explorer](#)

OneLake File Explorer

Article • 05/23/2023

The OneLake file explorer application seamlessly integrates OneLake with Windows File Explorer. This application automatically syncs all OneLake items that you have access to in Windows File Explorer. “Sync” refers to pulling up-to-date metadata on files and folders, and sending changes made locally to the OneLake service. Syncing doesn’t mean downloading the data. Instead placeholders are created. You must double click on a file to download the data locally.



ⓘ Important

Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

When you create, update, or delete a file via File Explorer, it automatically syncs the changes to OneLake service. Updates to your item made outside of your File Explorer aren't automatically synced. To pull these updates, you need to right click on the item or subfolder in Windows File Explorer and select **Sync from OneLake**.

Installation instructions

OneLake file explorer currently supports Windows and has been validated on Windows 10 and 11.

To install:

1. Download the [OneLake file explorer](#).

2. Double click the file to start installing.

The location on your PC where the placeholders and any downloaded content are stored is `\%USERPROFILE%\OneLake - Microsoft\`.

Once the application is installed and launched, you can now see your OneLake data in Windows File Explorer.

Known issues

- Workspace names with "/" character, encoded escape characters such as `%23` and names that look like GUIDs will fail to sync.
- Files or folders containing Windows reserved characters ([learn more](#)) fail to sync.
- Updating Office files (.xlsx, .pptx, .docx etc.) isn't currently supported.
- Moving a folder (cut and paste or drag and drop) from a location outside of OneLake to OneLake will fail to sync the contents in that folder. The contents move to the OneLake directory locally, but only the top-level folder syncs to OneLake. You must trigger a sync by either opening the files and saving them or moving them back out of OneLake and then copying and pasting (versus moving).
- Windows File Explorer is case insensitive, while OneLake is case sensitive. You can create files with the same name but different cases in the OneLake service using other tools, but Windows File Explorer only shows one of the files (the oldest one).

Scenarios

The following scenarios provide details for working with the OneLake file explorer.

Starting and exiting OneLake file explorer

OneLake file explorer starts automatically at startup of Windows. You can disable the application from starting automatically by selecting Startup apps in Windows Task Manager and then right clicking OneLake and select **Disable**.

- To manually start the application, search for "OneLake" using Windows search (Windows + S) and select the OneLake application. The views for any folders that were previously synced are then refreshed automatically.
- To exit, right-click on the OneLake icon in the Windows System Tray and select **Exit**. The sync is paused and placeholder files and folders cannot be accessed. You will

continue to see the blue cloud icon for placeholders that were previously synced but not downloaded.

Sync updates from OneLake

To optimize performance during the initial sync, OneLake file explorer syncs the placeholder files for the top-level workspaces and item names. When you open an item, OneLake file explorer syncs the files directly in that folder. Then, opening a folder within the item syncs the files directly in that folder. This allows you to navigate your OneLake content seamlessly, without having to wait for all files to sync before starting to work.

When you create, update, or delete a file via OneLake file explorer, it automatically syncs the changes to OneLake service. Updates to your item made outside of your OneLake file explorer aren't automatically synced. To pull these updates, you need to right click on the workspace name, item name, folder name or file in OneLake file explorer and select **Sync from OneLake**. This action refreshes the view for any folders that were previously synced. To pull updates for all workspaces, right click on the OneLake root folder and select **Sync from OneLake**.

Offline support

The OneLake file explorer only syncs updates when you're online and the application is running. When the application starts, the views for any folders that were previously synced are then refreshed automatically. Any files that were added or updated while offline are shown as sync pending until you save them again. Any files deleted while offline are recreated during the refresh if they still exist on the service.

Create files or folders in OneLake file explorer

1. Navigate to the OneLake section in Windows File Explorer.
2. Navigate to the appropriate folder in your item.
3. Right click and select **New folder or new file type**.

Note

If you write data to locations where you don't have write permission, such as the root of the item or workspace, the sync will fail. Clean up files or folders that failed to sync by moving them to the correct location or deleting them.

Delete files or folders in OneLake file explorer

1. Navigate to the OneLake section in Windows File Explorer.
2. Navigate to the **Files** or **Tables** folder in your item.
3. Select a file or folder and delete.

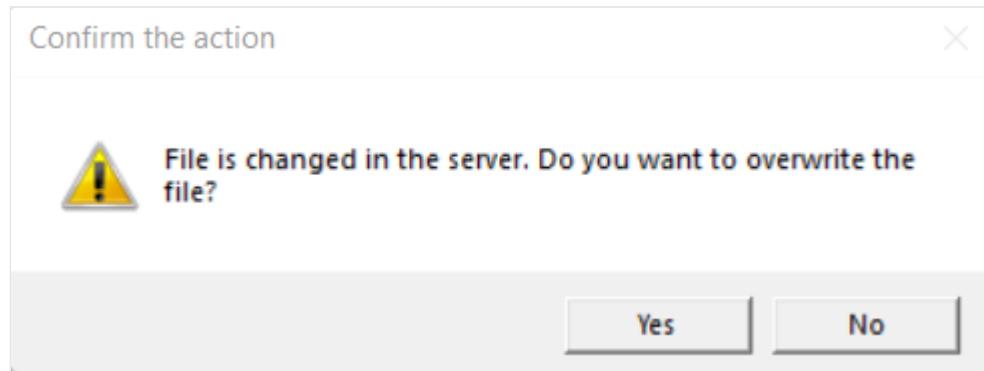
Edit files

You can open files using your favorite apps and make edits. Selecting **Save** syncs the file to OneLake.

Note

OneLake file explorer does not currently support updating Office files (excel, ppt, etc.).

If you edit a file locally and select **Save**, the OneLake file explorer app detects if that file has been updated elsewhere (by someone else) since you last selected **Sync from OneLake**.



If you select **Yes**, then your local changes overwrite any other changes made to the file since the last time you selected **Sync from OneLake**.

If you select **No**, then the local changes aren't sent to the OneLake service. You can then select **Sync from OneLake** to revert your local changes and pull the file from the service. Or you can copy the file with a new name to avoid conflicts.

Copying or moving files

You can copy files to, from, and within your items using standard keyboard shortcuts like **Ctrl+C** and **Ctrl+V**. You can also move files by dragging and dropping them.

Support for large files and a large number of files

When you upload or download files using the OneLake file explorer, the performance should be similar to using OneLake APIs. In general, the time it takes to sync changes from OneLake is proportional to the number of files.

OneLake shortcut support

All folders in your items including [OneLake shortcuts](#) are visible. You can view, update, delete the files and folders in those shortcuts.

Client-side logs

Client-side logs can be found on your local machine under
`%temp%\OneLake\Diagnostics\`.

Uninstall instructions

To uninstall the app, in Windows, search for “OneLake”. Select **Uninstall** in the list of options under OneLake.

OneLake file explorer icons

These OneLake file explorer icons are visible in Windows File Explorer and tell you the sync state of the file or folder.

Icon	Icon description	Meaning
	Blue cloud icon	The file is only available online. Online-only files don't take up space on your computer.
	Green tick	The file is downloaded to your local computer.
	Sync pending arrows	Sync is in progress. This icon may appear when you're uploading files. If the sync pending arrows are persistent, then your file or folder may have an error syncing. You can find more information in the client-side logs on your local machine under %temp%\OneLake\Diagnostics.

Next steps

Learn more about [OneLake security](#)

Discover data items in the OneLake data hub

Article • 05/23/2023

The OneLake data hub makes it easy to find, explore, and use the Fabric data items in your organization that you have access to. It provides information about the items and entry points for working with them.

The data hub provides:

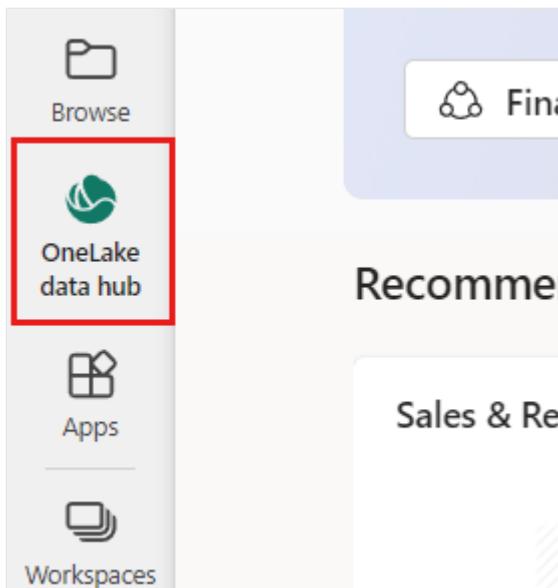
- A filterable list of all the data items you can access
- A gallery of recommended data items
- A way of finding data items by workspace
- A way to display only the data items of a selected domain
- An options menu of things you can do with the data item

This article explains what you see on the data hub and describes how to use it.

The screenshot shows the Power BI OneLake data hub interface. On the left, there's a navigation pane with icons for Home, Create, Browse, OneLake data hub (selected), Monitoring hub, Workspaces, and My workspace. The main area has a search bar at the top right. Below it, there's a section titled "OneLake data hub" with a sub-section "Recommended". This section displays six cards: "Product MAU, DAU, and NPS" (Certified), "Marketing insights" (Promoted), "Marketing" (Certified), "Top features" (Certified), "Customer feedback" (Certified), and "AppAccessSettings" (Promoted). Each card includes a small chart or icon and details like owner and last refresh. At the bottom, there's an "Explorer" section with a table of data items. The columns are: Name, Type, Refreshed, Owner, Location, Endorsement, and Sensitivity. Items listed include Sales FY21 (Dataset, 7m ago, Tim Deboar, Contoso workspace, Certified, Highly Confidential\Contoso...), Marketing DB (Datamart, 38m ago, Daichi Fukuda, New product insights, Certified, —), Client Logs Db (KQL Database, 2h ago, Emiliano Ceballos, Azure data, Promoted, Confidential\Contoso FTE), Top Campaigns (Dataset, 7h ago, Mikhail Kotov, Azure data, —, Public), Dataflow for triggers (Dataset, Yesterday at 11:12 AM, Marie Beaudouin, Contoso workspace, —, —), Daily Sales (Lakehouse, June 18 at 9:02 AM, Oscar Krog, Contoso workspace, Certified, Non-Business), Contoso DB (Warehouse, May 23 at 3:00 PM, Marie Beaudouin, Big data, Promoted, —), Test datamart (Datamart, May 15 at 5:13 AM, Tim Deboar, Events, Certified, Public), Primary dataflow (KQL Database, April 29 at 8:45 PM, Oscar Krog, Big data, —, —), and Contoso DB (Lakehouse, April 11 at 11:56 AM, Ruth Bengtsson, Contoso workspace, Certified, Confidential\Contoso FTE). There are also "All data", "My data", and "Endorsed in your org" buttons, a search bar, and a filter button.

Open the data hub

To open the data hub, select the OneLake data hub icon in the navigation pane.



Find items in the data items list

The data items list displays all the data items you have access to. To shorten the list, you can filter by keyword or data-item type using the filters at the top of the list. If you select the name of an item, you'll get to the item's details page. If you hover over an item, you'll see three dots that open the [options menu](#) when you select them.

Name	Type	Refreshed	Owner	Location	Endorsement	Sensitivity
Sales FY21	Dataset	7m ago	Tim Deboar	Contoso workspace	Certified	Highly Confidential\Contoso...
Marketing DB	Datamart	38m ago	Daichi Fukuda	New product insights	Certified	—
Client Logs Db	KQL Database	2h ago	Emiliano Ceballos	Azure data	Promoted	Confidential\Contoso PTE...
Top Campaigns	Dataset	7h ago	Mikhail Kotov	Azure data	—	Public
Dataflow for trincores	Dataset	Yesterday at 11:12 AM	Marie Beaudouin	Contoso workspace	—	—

The list has three tabs to narrow down the list of data items.

Tab	Description
All	Data items that you're allowed to find.
My data	Data items that you own.
Endorsed in your org	Endorsed data items in your organization that you're allowed to find. Certified data items are listed first, followed by promoted data items. For more information about endorsement, see the Endorsement overview

The columns of the list are described below.

Column	Description
Name	The data item name. Select the name to open the item's details page.
Endorsement	Endorsement status.

Column	Description
Owner	Data item owner (listed in the <i>All</i> and <i>Endorsed in your org</i> tabs only).
Workspace	The workspace the data item is located in.
Refreshed	Last refresh time (rounded to hour, day, month, and year. See the details section on the item's detail page for the exact time of the last refresh).
Next refresh	The time of the next scheduled refresh (<i>My data</i> tab only).
Sensitivity	Sensitivity, if set. Select the info icon to view the sensitivity label description.

Find items by workspace

Related data items are often grouped together in a workspace. To see the data items by workspace, expand the **Explorer** pane and select the workspace you're interested in. The data items you're allowed to see in that workspace will be displayed in the data items list.

The screenshot shows the Power BI interface with the following elements:

- Top Navigation:** Buttons for "All data", "My data", and "Endorsed in your org".
- Left Sidebar (Explorer):**
 - A button labeled "All workspaces" is highlighted with a red box.
 - Below it is a search bar with the placeholder "Search" and a filter icon.
 - A list of workspaces:
 - My Workspace
 - Contoso workspace
 - New product insights
 - RandomFolder
 - ShortcutLink Folder
 - Azure data design
- Right Panel:** A list of data items grouped by workspace:
 - Sales FY21
 - Marketing DB
 - Client Logs Db
 - Top Campaigns
 - Daily Sales
 - Contoso DB
 - Test datamart
 - Primary dataflow
 - Contoso DB
 - Primary dataflow

Note

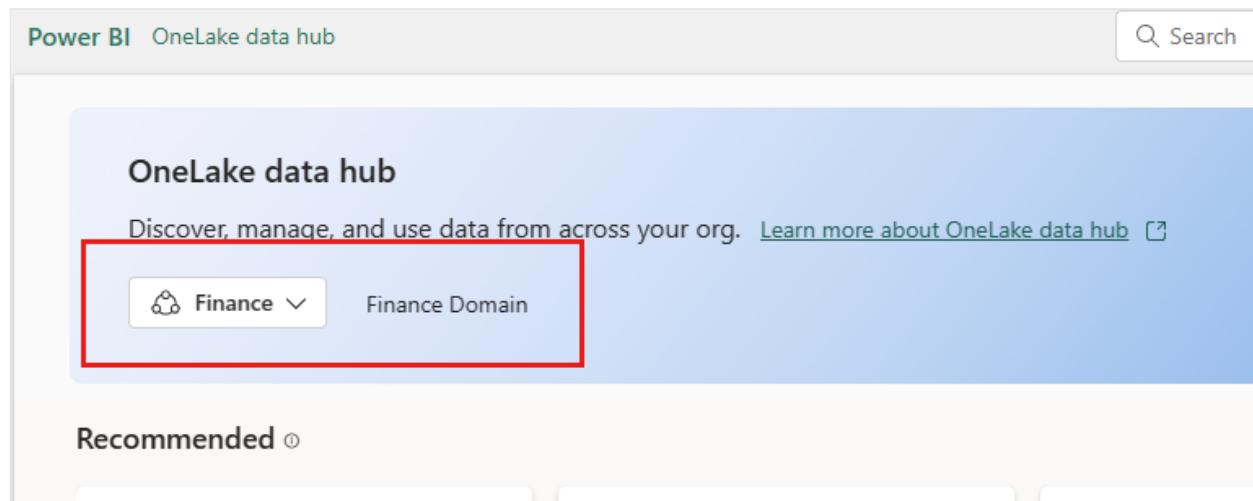
The Explorer pane may list workspaces that you don't have access to if the workspace contains items that you do have access to (through explicitly granted permissions, for example). If you select such a workspace, only the items you have access to will be displayed in the data items list.

Find recommended items

Use the tiles across the top of the data hub to find and explore recommended data items. Recommended data items are data items that have been [certified or promoted](#) by someone in your organization or have recently been refreshed or accessed. Each tile contains information about the item and an [options menu](#) for doing things with the item. When you select a recommended tile, you are taken to the item's details page.

Display only data items belonging to a particular domain

If [domains](#) have been defined in your organization, you can use the domain selector to select a domain so that only data items belonging to that domain will be displayed. If an image has been associated with the domain, you'll see that image on the data hub to remind you of the domain you're viewing.

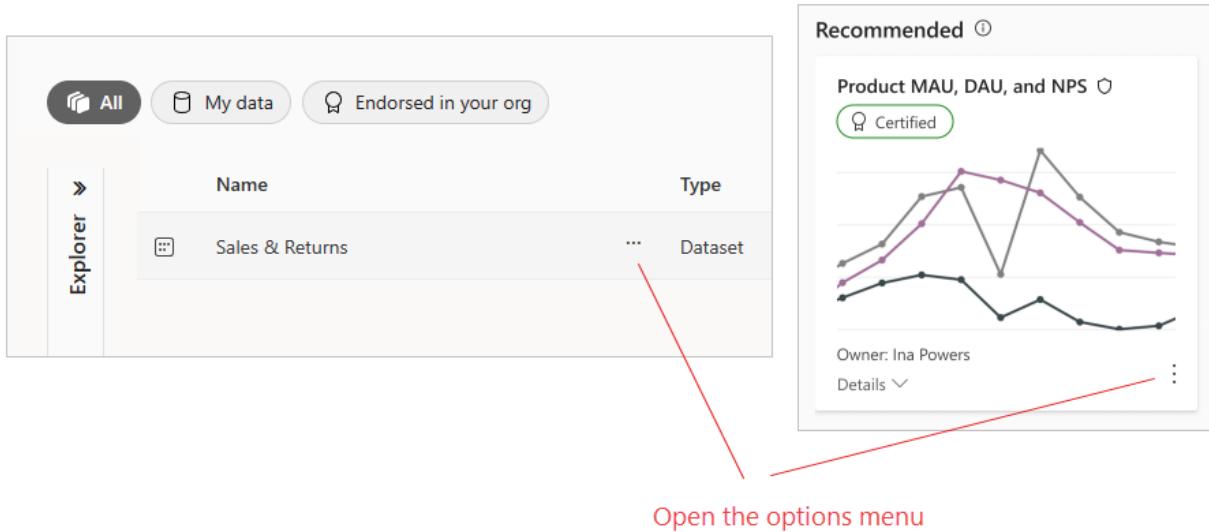


For more information about domains, see the [Domains overview](#)

Open an item's options menu

Each item shown in the data hub has an options menu that enables you to do things, such as open the item's settings, manage item permissions, etc. The options available depend on the item and your permissions on the item.

To display the options menu, select **More options (...)** on one of the items shown in the data items list or a recommended item. In the data items list, you need to hover over the item to reveal **More options**.



ⓘ Note

The Explorer pane may list workspaces that you don't have access to if the workspace contains items that you do have access to (through explicitly granted permissions, for example). If you select such a workspace, only the items you have access to will be displayed in the data items list.

Next steps

- [Navigate to your items from Microsoft Fabric Home](#)
- [Endorsement](#)