INVISO

OneLake in Microsoft Fabric: An introduction

May, 2023

Contents

Preface 3

OneLake 4

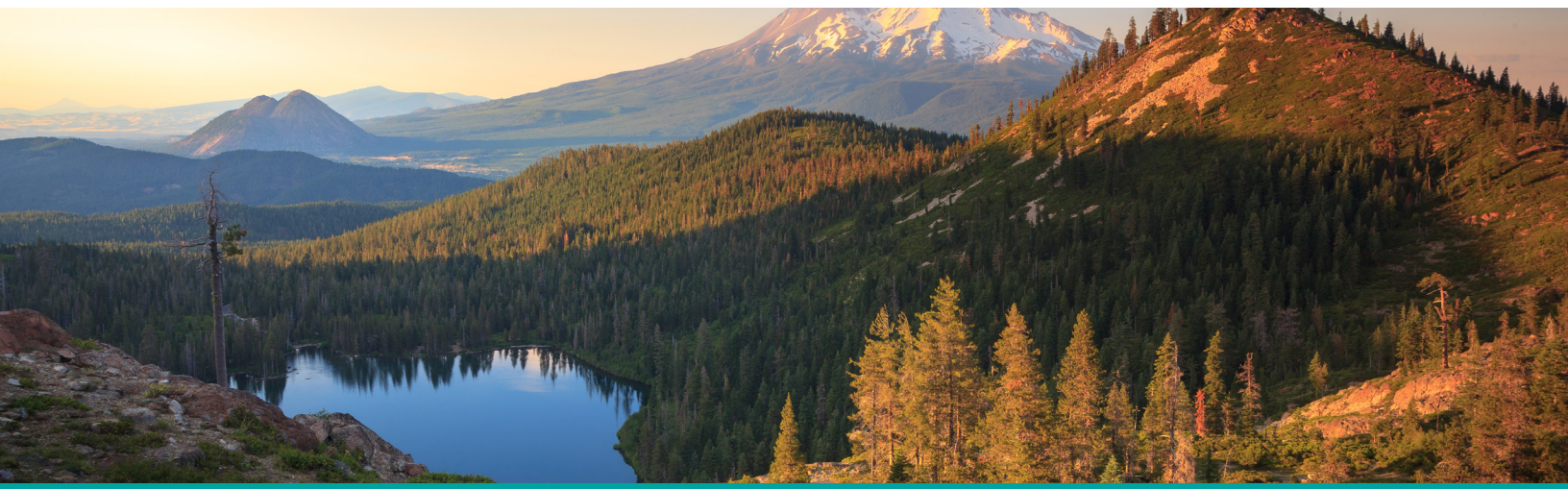
What is OneLake? 4

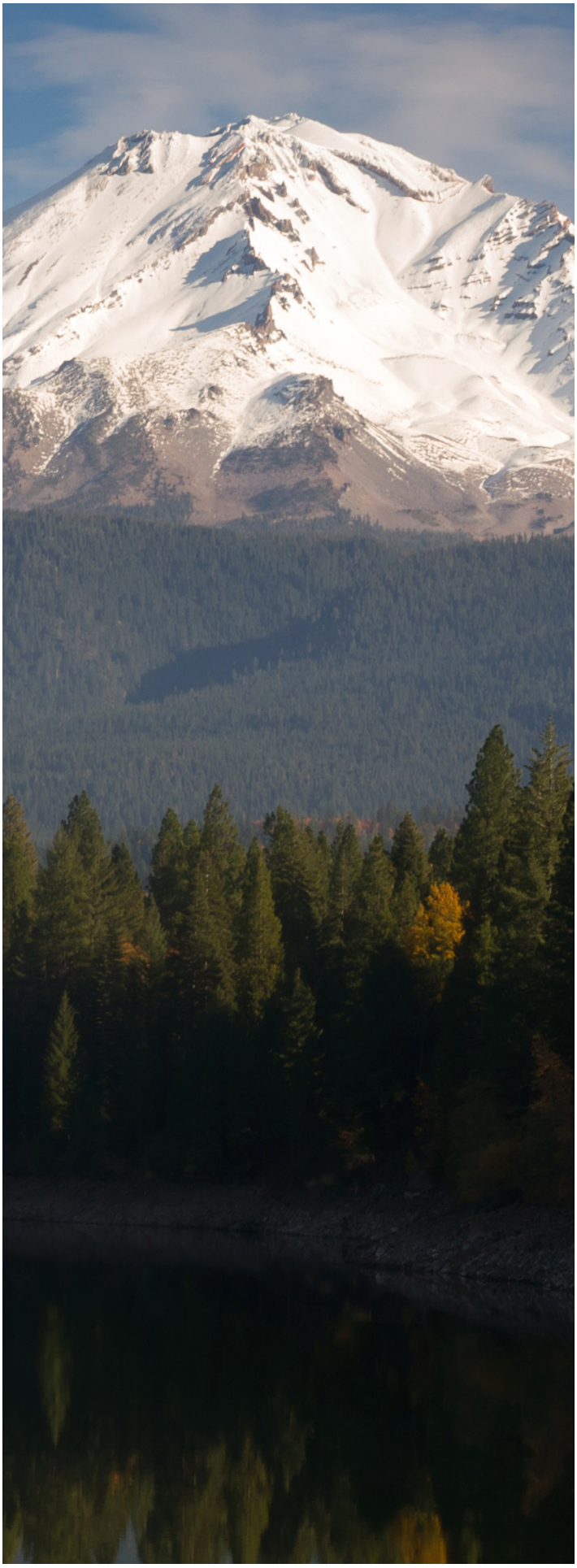
- Features of OneLake 5
- How is OneLake intended to be used? 6
- Security 6
- Item security 7
- Compute-specific security 7
- Limitations 7

Shortcuts 8

- What is Shortcuts? 8
- How is it intended to be used? 8
- Limitations 9

Coming features 9





Preface

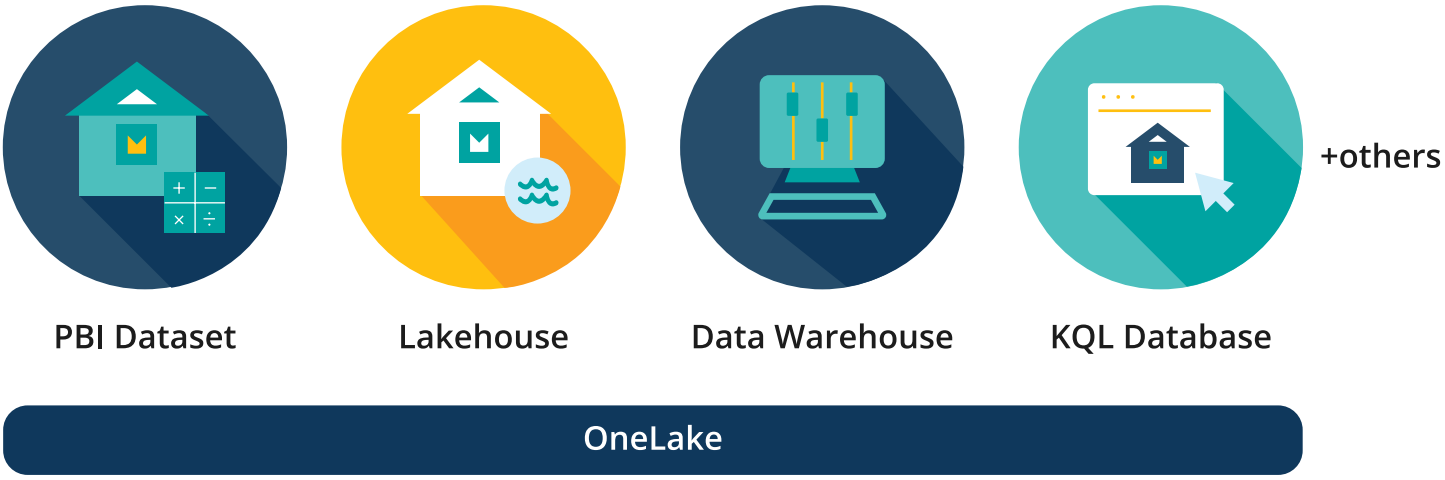
The services and features described below for Fabric are available at the time of the initial public preview release. Microsoft has a roadmap of many additional services and features to be added prior to General Availability release. This document provides an overview of the capabilities of the OneLake concept and the Shortcuts feature as they are at the time of the public preview launch.

What is OneLake?

OneLake is a single, unified, logical data lake for analytics data across the whole organization. It is a brand-new service and comes automatically with every Fabric tenant. It is automatically provisioned behind the scenes when the first OneLake artifact (such as a lakehouse) is created within Fabric. There is only one OneLake per tenant.

Built on top of ADLS (Azure Data Lake Storage) Gen 2, OneLake provides a single, SaaS-ified, tenant-wide store for data that serves both professional and citizen developers. As with the rest of Fabric, the SaaS-ification of the lake simplifies experiences without the user ever needing to understand any infrastructure concepts such as resource groups, RBAC (Role-Based Access Control), ARM, redundancy, or regions. In fact, users aren't even required to have an Azure account. Policy and security settings are enforced centrally and uniformly.

Comparatively, a lakehouse in Fabric is a collection of files/folders/tables that represent a database over a data lake used by the Spark engine and SQL engine for big data processing. This includes enhanced capabilities for ACID transactions when using open-source Delta Lake format tables.



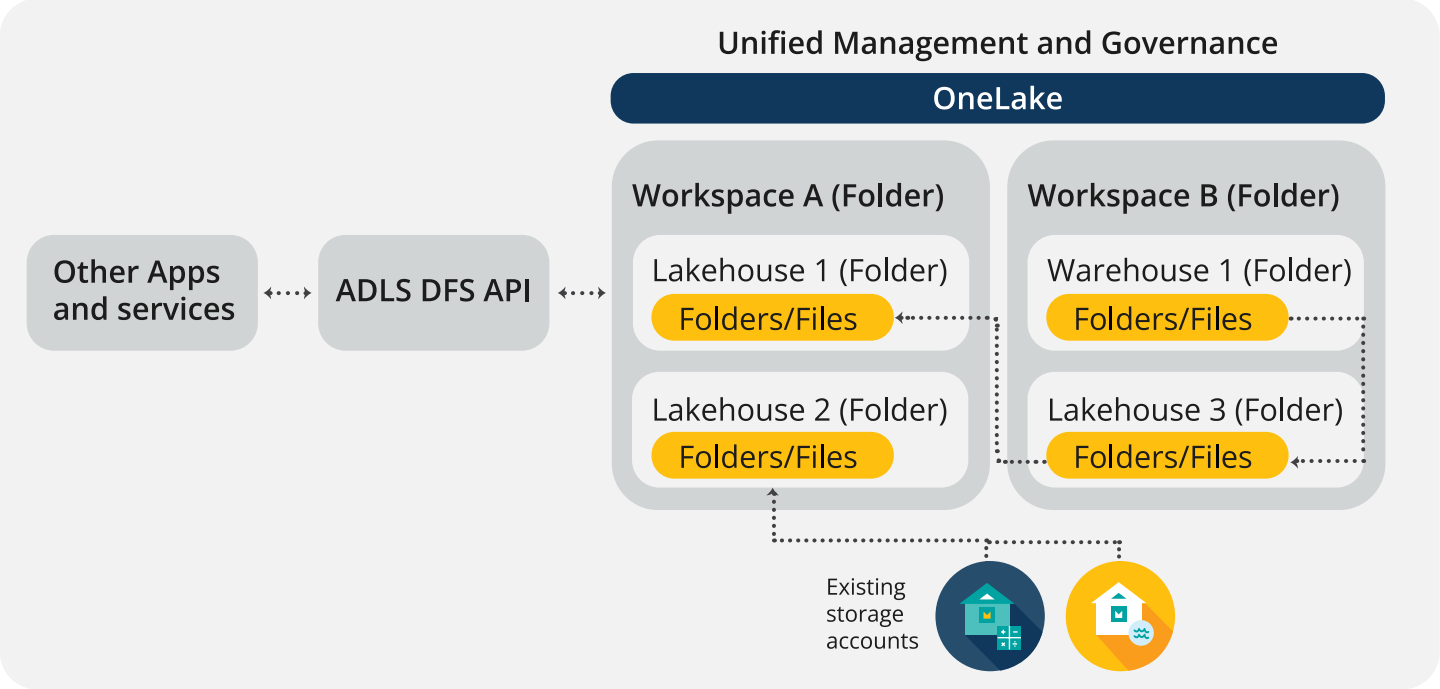
Features of OneLake

Open Access

OneLake supports ADLS Gen2 APIs and SDKs, so existing applications will simply work. There is no lock-in to proprietary technologies or formats. Existing tools can access data in OneLake simply by pointing to the OneLake endpoints.

One Copy

OneLake artifacts store data natively in open-source formats, enabling it to be accessed across multiple analytical engines without data duplication.



Flexible security

Easily secure data in OneLake by granting access either directly to the data or by limiting access to specific query engines.

Shortcuts

Virtualize data across domains and clouds into a single logical data lake. Data can be shared between users and applications without having to copy data across multiple lakes or other data stores. Data can also be shared across workspaces while physically stored in a single file location.

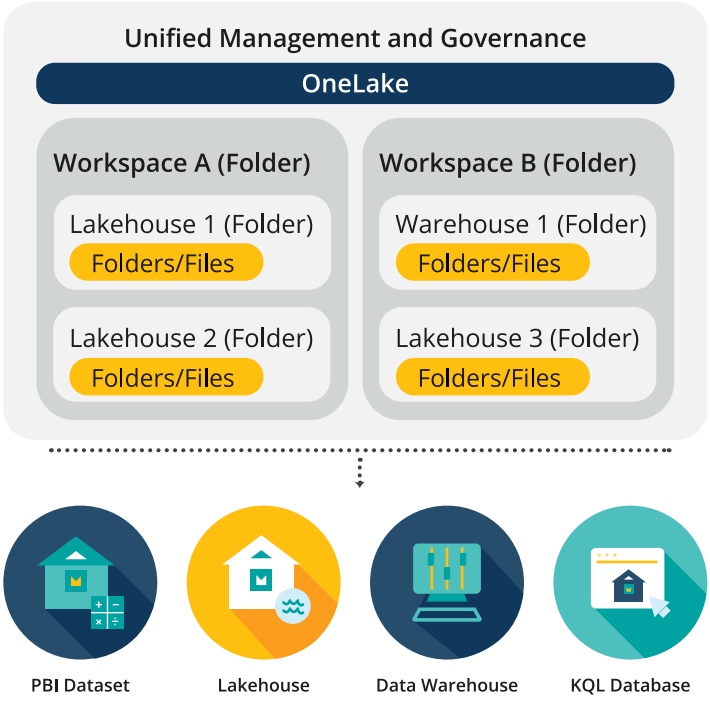
How is OneLake intended to be used?

With everyone contributing to the same underlying data lake, the tenant admin can unify management and governance policies across all teams and domains in an organization. Within a tenant, you can create any number of workspaces. Within a workspace you can create artifacts and all data in OneLake is accessed through data artifacts. Similar to how Office stores Word, Excel and PowerPoint files in OneDrive, Fabric stores lakehouses, warehouses, and other artifacts in OneLake.

While Fabric is deeply committed to supporting open data formats across all its workloads (compute engines), Fabric treats Delta Lake as a first-class format that is natively supported by all workloads (Data Engineering, Data Factory, Data Science, Data Warehousing, Kusto, and Power BI). This deep commitment to a common open data format means that data can be loaded into the lake only once and all the workloads can operate on the same data, without having to separately ingest it into each engine.

Security

The structure of the components within Fabric allows for a layered security model approach. Security begins with the user of Azure Active Directory (AAD) authentication that is compatible with user identities and service principles. Authorization begins at the workspace just as with the Power BI service. Within each workspace, different items have their own security that can be defined, and access through compute engines like SQL can be set separately from direct data lake access.



Item security

Individual items in Fabric can have additional security or settings applied. In general, the access a user has to an item is inherited from their Fabric RBAC role on the workspace. However, items can be shared with users directly, granting them access to just that item without adding them to a workspace role. The table below outlines the scenarios around sharing lakehouses and warehouses and how they apply to OneLake.

Action	View files in OneLake	Write files in OneLake
If a warehouse is shared with you...	No	No
If a lakehouse is shared with you...	Yes	No
If a lakehouse with write access is shared with you...	Yes	Yes

Compute-specific security

Some compute engines in Fabric have their own security models. For example, warehouse lets users define access using T-SQL statements. Compute-specific security is always enforced when accessing data using that engine, but users in certain Fabric roles may not be bound by those conditions when accessing OneLake directly.

As a general rule, users in the Viewer workspace role cannot access OneLake directly. Viewers are limited to accessing data through the compute engine and are restricted by any rules set there. All other roles contain direct OneLake access, allowing them to query data through Spark, APIs, or a OneLake file explorer. However, compute-specific security will still apply to those users when accessing data through that compute engine.

Example: Martha is an administrator for a Fabric workspace. Her co-administrator Pradeep set up rules in SQL to restrict access to the Sales table unless you have accounts assigned to you. When Martha runs a SELECT query from the Sales table, she only sees data for accounts assigned to her. However, because she has OneLake access she can use the lakehouse explorer to view and open the files backing the Sales table.

Limitations

Private link access to OneLake data via non-Fabric workloads and Fabric Spark is not currently supported.

What is Shortcuts?

Also known as symbolic links, shortcuts are embedded references within a lakehouse in OneLake that point to other files' stores.

How is it intended to be used?

OneLake Shortcuts provides a way to connect to existing data without having to directly copy it. This eliminates the need to set up and monitor data movement jobs and keep data in sync across sources.

Shortcuts can be created that point to external storage accounts such as Azure Data Lake Storage Gen 2. This allows you to quickly source your existing cloud data without having to copy it.

Shortcut locations

Lakehouses are comprised of two top level folders. The "Tables" folder and the "Files" folder.

In the "Tables" folder, you can only create shortcuts at the top level. If the source of the shortcut contains data in the Delta\Parquet format, the lakehouse will automatically synchronize the meta data 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.

Authorization

Shortcuts to other OneLake locations

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 source of the shortcut*. This user must have "readAll" permissions on the source data.

IMPORTANT*: When accessing shortcuts through Power BI Datasets or T-SQL, the calling user's identity is not passed through to the shortcut source. The calling artifact owner's identity is passed instead.

Shortcuts can be accessed through the ABFS driver or REST endpoint directly.

Limitations

Currently supported Shortcuts sources are Fabric lakehouse, ADLS Gen2 and Amazon S3.

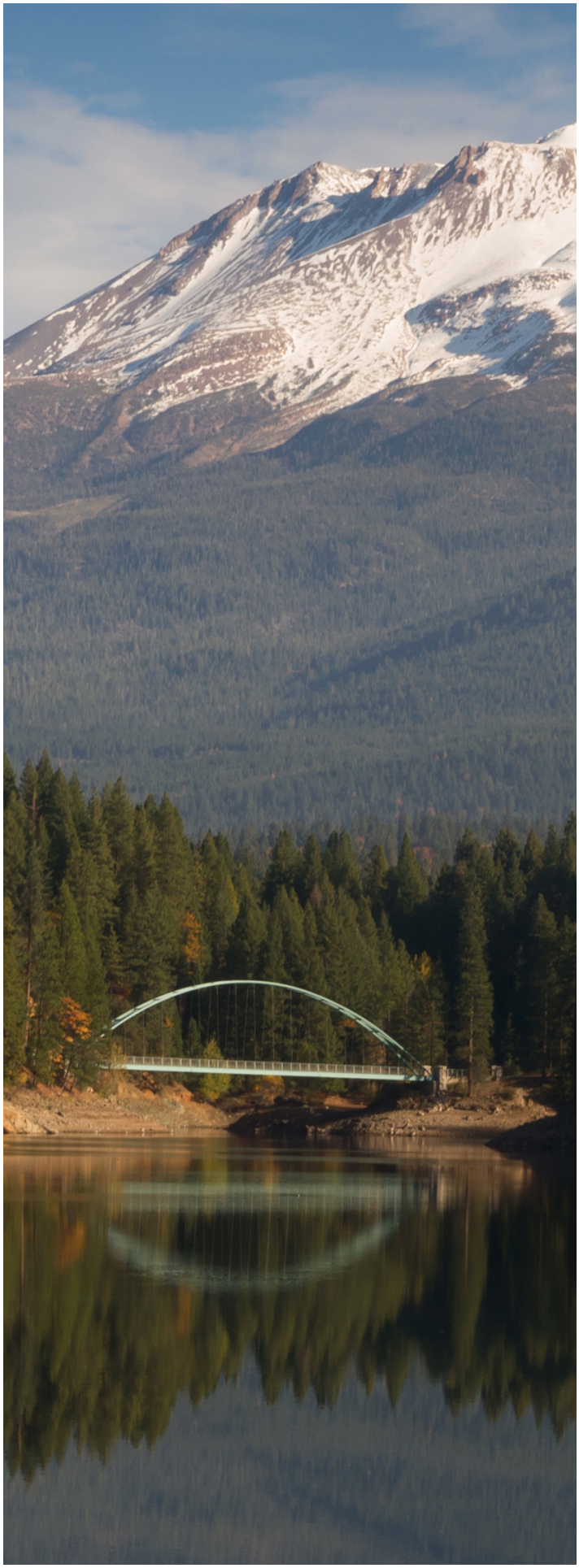
Note these current considerations for use:

- The ability to pass through a credential to an ADLS location will be removed later in public preview. At that point you will need to explicitly define a credential to use. This could be a user or an SPN.
- Shortcuts not currently supported by datasets.
- Shortcuts not currently supported by data warehouse and lakehouse TDS endpoint.
- When creating a shortcut to an ADLS Gen 2 account, the shortcut must point to a public endpoint. Private endpoints not currently supported for shortcuts to ADLS Gen 2.
- Accessing shortcuts directly through Python is not currently supported.
- Unable to create shortcuts to files in a lakehouse at this point in time. The target must be a folder.
- The maximum depth of shortcuts to other shortcuts is 10.

Coming features

In the future, the Shortcuts capability of OneLake will extend to additional non-Microsoft storage systems such as Google Storage, enabling users to compose and analyze data across multiple clouds with transparent, smart caching that reduces egress costs and brings data closer to compute.

If you have any questions or would like further information on the new functionality opened up with OneLake in Fabric, please reach out to us at **info@invisocorp.com.**



References

- Microsoft OneLake Consolidated Documentation v2.0.pdf
- Microsoft Fabric Overview and Concepts.pdf
- Microsoft OneLake Scenarios.pdf

NOTICE

The information contained in this document represents the current view of Inviso on the issues discussed as of the date of publication and is subject to change at any time without notice. This document and its contents are provided AS IS without warranty of any kind and should not be interpreted as an offer or commitment on the part of Inviso and Inviso cannot guarantee the accuracy of any information presented.

INVISO MAKES NO WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

All trademarks are the property of their respective companies.
All rights reserved.