

Data Factory documentation in Microsoft Fabric

Data factory in Microsoft Fabric provides cloud-scale data movement and data transformation services that allow you to solve the most complex data factory and ETL scenarios. It's intended to make your data factory experience easy to use, powerful, and truly enterprise-grade.

About Data Factory

OVERVIEW

[What is Data Factory?](#)

[Connector overview](#)

Get started

GET STARTED

[Create your first dataflow](#)

[Create your first data pipeline](#)

[Move and transform data](#)

TUTORIAL

[Create an end-to-end data integration scenario](#)

TRAINING

[Data Factory training](#)

Working with data pipelines

HOW-TO GUIDE

[Data pipeline runs](#)

[Copy data - copy activity](#)

[Monitor data pipeline runs](#)

[Activity Overview](#)

[Move data with copy assistant](#)

Working with dataflows

HOW-TO GUIDE

[Data source management](#)

[Monitor your dataflows](#)

[Save a draft of your dataflow](#)

What is Data Factory in Microsoft Fabric?

Article • 05/23/2023

Data Factory empowers you with a modern data integration experience to ingest, prepare and transform data from a rich set of data sources (for example, databases, data warehouse, Lakehouse, real-time data, and more). Whether you are a citizen or professional developer, you will be able to transform the data with intelligent transformations and leverage a rich set of activities. With Data Factory in Microsoft Fabric, we are bringing fast copy (data movement) capabilities to both dataflows and data pipelines. With Fast Copy, you can move data between your favorite data stores blazing fast. Most importantly, Fast Copy enables you to bring data to your Lakehouse and Data Warehouse in Microsoft Fabric for 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

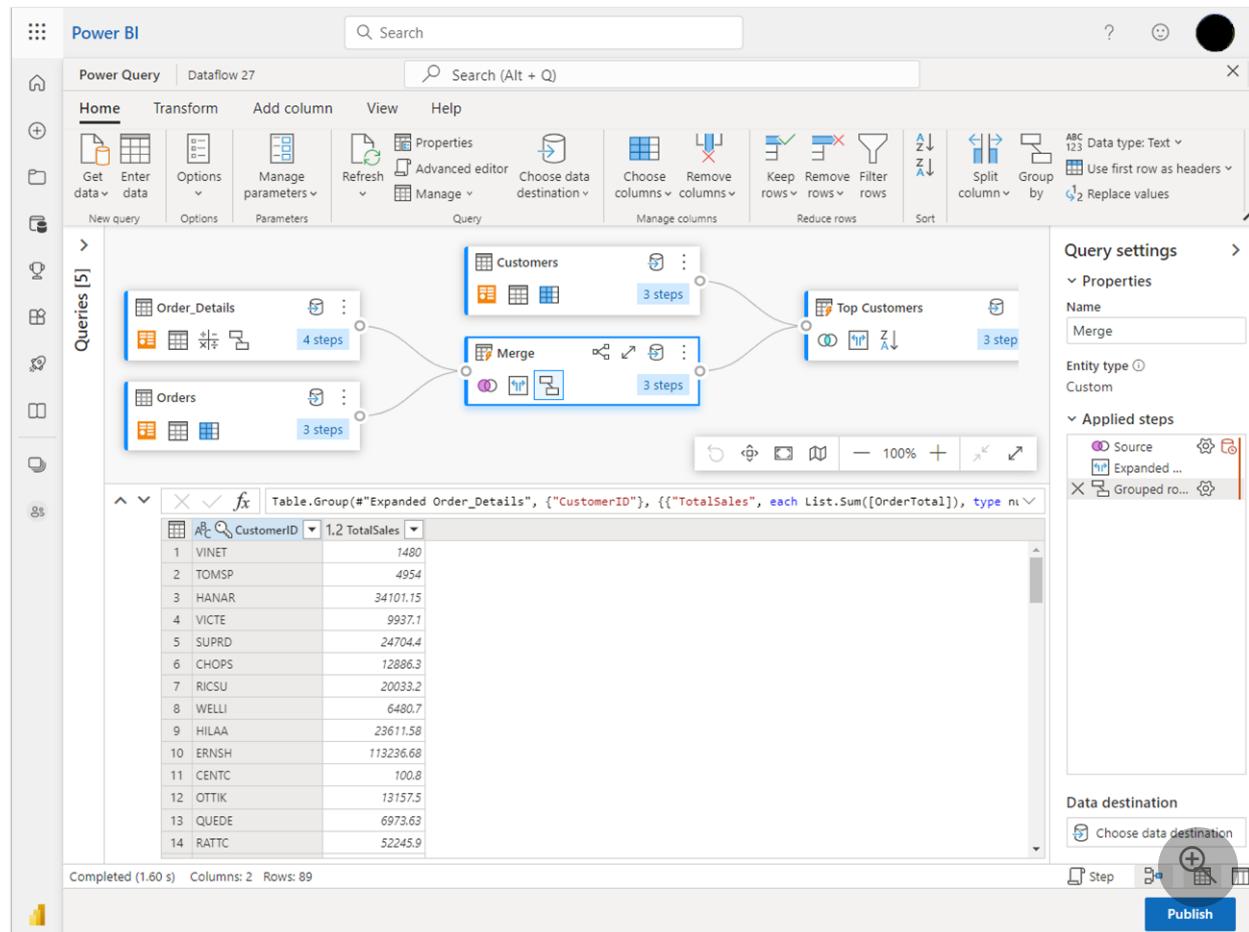
There are two primary high-level features Data Factory implements: dataflows and pipelines.

- Dataflows enable you to leverage more than 300 transformations in the dataflows designer, letting you transform data easier and with more flexibility than any other tool - including smart AI-based data transformations.
- Data pipelines enable you to leverage the out-of-the-box rich data orchestration capabilities to compose flexible data workflows that meet your enterprise needs.

Dataflows

Dataflows provide a low-code interface for ingesting data from hundreds of data sources, transforming your data using 300+ data transformations. You can then load the resulting data into multiple destinations, such as Azure SQL databases and more. Dataflows can be run repeatedly using manual or scheduled refresh, or as part of a data pipeline orchestration.

Dataflows are built using the familiar [Power Query](#) experience that's available today across several Microsoft products and services such as Excel, Power BI, Power Platform, Dynamics 365 Insights applications, and more. Power Query empowers all users, ranging from citizen to pro, to perform data ingestion and data transformations across their data estate. Perform joins, aggregations, data cleansing, custom transformations, and much more all from an easy-to-use, highly visual, low-code UI.

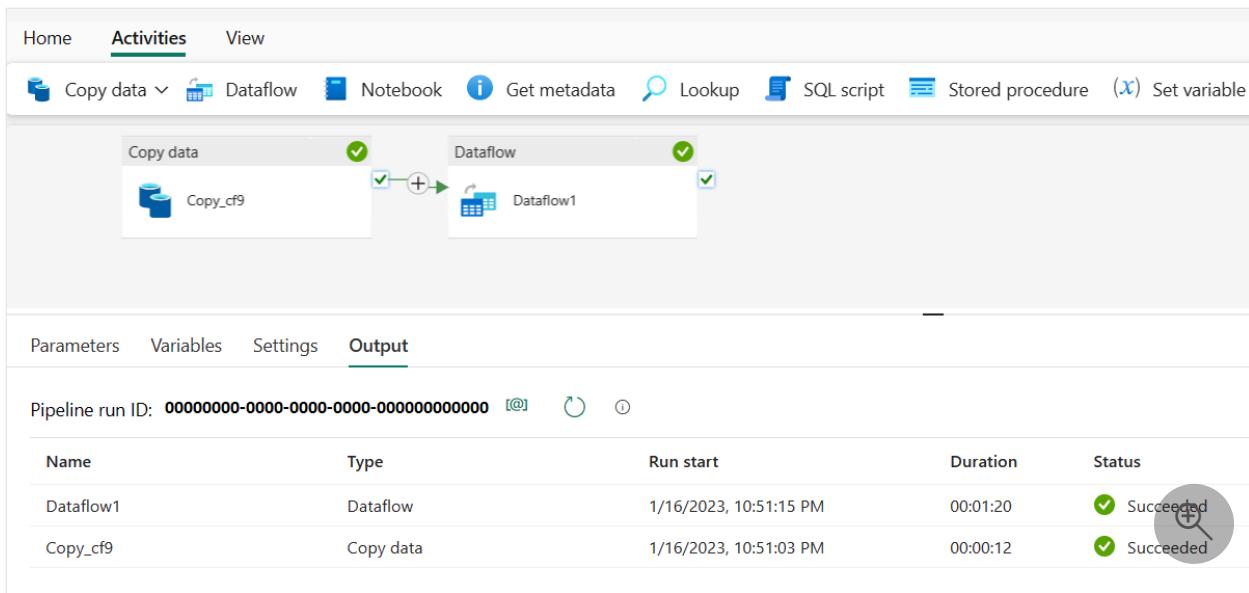


Data pipelines

Data pipelines enable powerful workflow capabilities at cloud-scale. With data pipelines, you can build complex workflows that can refresh your dataflow, move PB-size data, and define sophisticated control flow pipelines.

Use data pipelines to build complex ETL and data factory workflows that can perform many different tasks at scale. Control flow capabilities are built into data pipelines that allow you to build workflow logic, which provides loops and conditionals.

Add a configuration-driven copy activity together with your low-code dataflow refresh in a single pipeline for an end-to-end ETL data pipeline. You can even add code-first activities for Spark Notebooks, SQL scripts, stored procs, and more.



Next steps

To get started with Microsoft Fabric, go to [Quickstart: Create your first Dataflow Gen2 to get and transform data.](#)

Quickstart: Create your first dataflow to get and transform data

Article • 05/23/2023

Dataflows are a self-service, cloud-based, data preparation technology. In this article, you'll create your first dataflow, get data for your dataflow, then transform the data and publish the dataflow.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Prerequisites

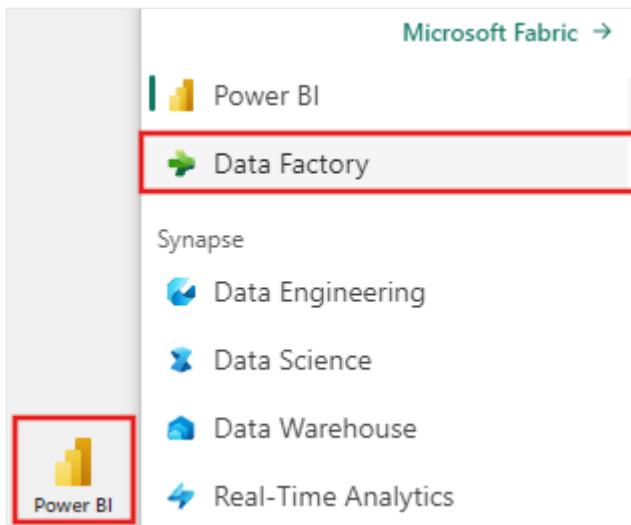
The following prerequisites are required before you start:

- A Microsoft Fabric tenant account with an active subscription. Create an account for free.
- A Microsoft Fabric enabled Workspace: [Create a workspace](#).

Create a dataflow

In this section, you'll be creating your first dataflow.

1. Switch to the **Data factory** or **Power BI** experience.



2. Navigate to your Microsoft Fabric workspace.

The Microsoft Fabric workspace 'Browse' screen is shown. The left sidebar includes Home, Create, Browse, Data hub, Monitoring hub, Workspaces, and DF Getting Started. The 'Recent' section shows three items: 'DF Getting Started' (Workspace), 'Release Plan Publish Schedule' (Report), and 'pipeline4' (Data pipeline). A search icon is at the bottom right.

Name	Type	Op
DF Getting Started	Workspace	33
Release Plan Publish Schedule	Report	3 h
pipeline4	Data pipeline	23

3. Select New, and then select Dataflow Gen2.

The 'DF Getting Started' workspace interface is shown. At the top, there's a 'New' button with a plus sign and a dropdown, an 'Upload' button with an upward arrow and dropdown, and a grid icon. Below this, there are three preview options: 'Data pipeline (Preview)', 'Dataflow Gen2 (Preview)' (which is highlighted with a red box), and 'Eventstream (Preview)'. The 'Dataflow Gen2 (Preview)' option is the target for selection.

Get data

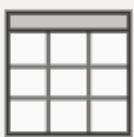
Let's get some data! In this example, you'll be getting data from an OData service. Use the following steps to get data in your dataflow.

1. In the dataflow editor, select **Get data** and then select **More**.

Home

Transform

Add column

Get
data ▾Enter
data

Options

Manage
parameters ▾

Parameters



Excel workbook



Dataflows



SQL Server database



Text/CSV



Web page



OData



Blank query

More...

2. Select OData as the data source.

Choose data source

Select a connector or directly drag a file from your computer.

All categories	File	Database	Power Platform	Azure	Online services	Other	Search
Excel workbook	Text/CSV	XML	JSON			Folder	
PDF	Parquet	SharePoint folder	SQL Server database			Access Database	
SQL Server Analysis Services	Oracle database	IBM Db2 database	MySQL database			PostgreSQL database	
Teradata database	SAP HANA database	SAP BW Application Server	SAP BW Message Server			Snowflake	
Google BigQuery	Amazon Redshift	Impala	Vertica			Dataflows Power Platform	
Power BI dataflows	Dataverse	Common Data Service (Legacy)	Azure SQL database			Azure Synapse Analytics (SQL DW)	
Azure Analysis Services	Azure Blobs	Azure Tables	Azure Data Explorer (Kusto)			Azure Data Lake Storage Gen2	
Azure HDInsight Spark	SharePoint Online list	Microsoft Exchange Online	Salesforce objects			Salesforce reports	
Google Analytics	Adobe Analytics	Web API	Web page			SharePoint list	
OData	Spark	Odbc	Actian			Amazon Athena	
AtScale cubes	BI Connector	Data Virtuality LDW	Dendodo			Dremio Cloud	
Dremio Software	Exasol	Indexima	InterSystems IRIS			Jethro	
Kyligence	Linkar PICK Style / MultiValue Data...	MariaDB	MarkLogic			TIBCO(R) Data Virtualization	
Azure Cosmos DB v2	Azure Cost Management	Azure Databricks	Azure Synapse Analytics workspace...			Azure Time Series Insights	
Asana	Assemble Views	Automation Anywhere	Automy Data Analytics			Databricks	
Digital Construction Works Insight...	Dynamics 365 Business Central	Dynamics 365 Customer Insights...	Emigo Data Source			Entersoft Business Suite	
FactSet Analytics	Funnel	Hexagon PPM Smart® API	Industrial App Store			Intune Data Warehouse	

OData Other

Cancel 

3. Enter the URL `https://services.odata.org/v4/northwind/northwind.svc/`, and then select **Next**.

Connect to data source

OData Other [Learn more](#)

Connection settings

URL *
`https://services.odata.org/v4/northwind/northwind.svc/`

Connection credentials

Connection
 

Connection name

Data gateway
 

Authentication kind

Back **Cancel** **Next**

4. Select the **Orders** and **Customers** tables, and then select **Create**.

Choose data

The screenshot shows the 'Choose data' interface in Power BI Data Flow. On the left, there's a sidebar with a search bar and a 'Display options' dropdown. Below that is a tree view of OData sources, with 'ODATA' expanded to show tables like 'Alphabetical_list_of_pro...', 'Categories', 'Category_Sales_for_1997', 'Current_Product_Lists', 'Customer_and_Suppliers...', 'CustomerDemographics', 'Customers' (which is selected, indicated by a checked checkbox), 'Employees', 'Invoices', 'Order_Details', 'Order_Details_Extendeds', 'Order_Subtotals', 'Orders' (which is also selected), 'Orders_Qries', 'Product_Sales_for_1997', and 'Products'. There's also a 'Select related tables' button. On the right, the 'Orders' table is shown as a data grid with 26 rows. The columns are OrderID, CustomerID, EmployeeID, OrderDate, RequiredDate, and ShippedDate. The data spans from July 1996 to August 1996. At the bottom right of the grid are 'Cancel' and 'Create' buttons, along with a magnifying glass icon.

You can learn more about the get data experience and functionality at [Getting data overview](#).

Apply transformations and publish

You have now loaded your data into your first dataflow, congratulations! Now it's time to apply a couple of transformations in order to bring this data into the desired shape.

You'll be doing this task from the Power Query editor. You can find a detailed overview of the Power Query editor at [The Power Query user interface](#).

Follow these steps to apply transformations and publish:

1. Make sure you've enabled the [diagram view](#) using the options under the [View tab](#) in the Power Query editor ribbon. Also ensure that the [Data Profiling tools](#) are enabled by navigating to [Home > Options > Global Options](#).

Options ?



Global

General

Data load

Diagnostics

Dataflow

Security

Data load

Regional settings

Default editor view

Data view

Schema view

Steps

Enable query folding indicators

Show script in step callout

Column profile

Enable column profile

Show column quality details in data preview

Show column value distribution in data preview

Show column profile in details pane

Column profile evaluation

Based on top 1,000 rows

Based on entire data set

Data view

Enable details pane

Display preview contents using a monospaced font

Show whitespace and newline characters

Parameters

Always allow parameterization in data source and transformation dialogs

OK

Cancel

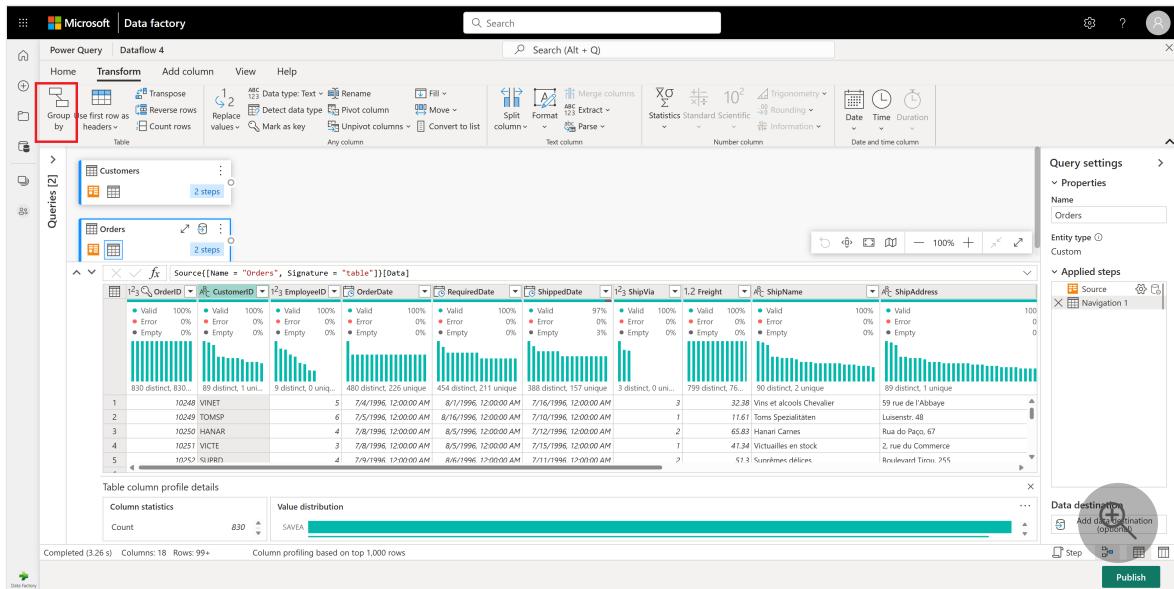
The screenshot shows the Microsoft Power Query Editor interface. At the top, there's a ribbon with Home, Transform, Add column, View, Help, and a search bar. Below the ribbon, there are various toolbar icons for operations like Get data, Enter data, Options, Manage parameters, Refresh, Advanced editor, Add data destination, Choose columns, Remove columns, Keep rows, Remove rows, Filter rows, Reduce rows, Split column, Group by, Use first row as headers, Sort, Transform, Merge queries, Append queries, Combine files, Map to entity, and CDM. On the left, there's a sidebar with 'Queries [2]' showing 'Customers' and 'Orders'. The main area displays the 'Customers' query results:

	CustomerID	CompanyName	ContactName	ContactTitle	Address	City	Region	PostalCode	Country	Phone
1	AUFI	Alfreds Futterkiste	Maria Anders	Sales Representative	Oberbe Str. 57	Berlin	Germany	12209	Germany	(0) 555-0
2	ANATR	Ara Trujillo Emparedados y helados	Ana Trujillo	Owner	Avenida de la Constitución 2222	México D.F.	México	05023	México	(5) 555-3
3	ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2312	México D.F.	México	05003	México	(5) 555-3
4	AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Sq.	London	UK	WA1 1DP	UK	(171) 555-
5	BERGS	Berglunds snabbköp	Christina Berglund	Order Administrator	Berglunds b&B	London	UK	SE-121 01	Sweden	(0) 555-12

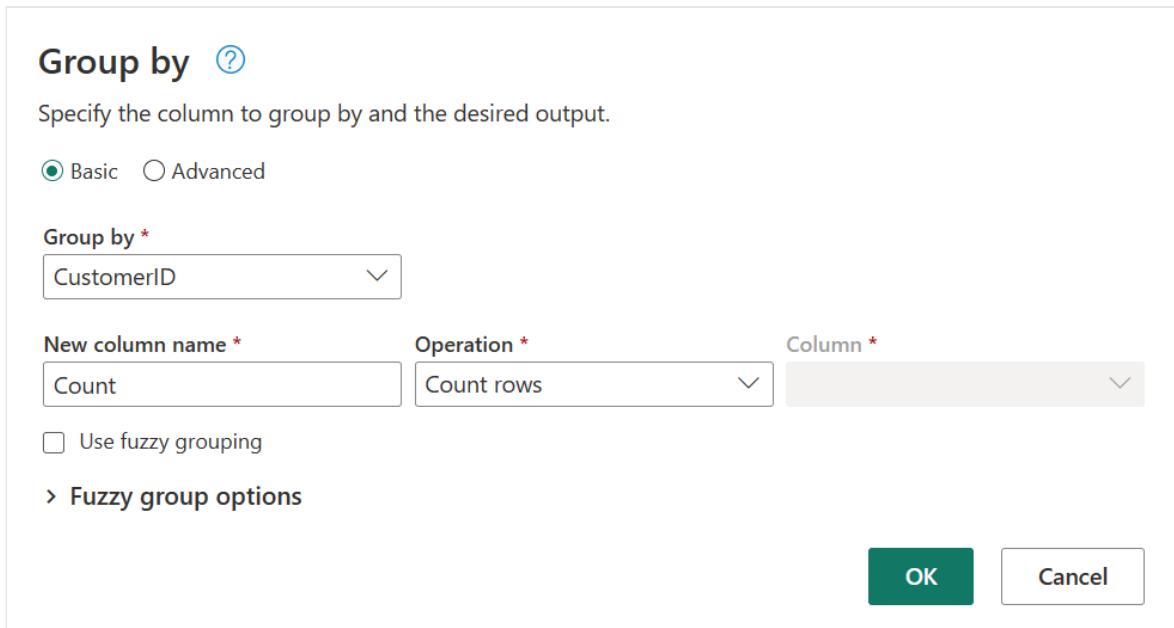
Below the table, it says 'Completed (4.02 s) Columns: 13 Rows: 91'. A note at the bottom states 'Column profiling based on top 1,000 rows'. On the right side of the interface, there are 'Query settings' and 'Data destination' sections.

- Within the Orders table, you'll calculate the total number of orders per customer. To achieve this goal, select the CustomerID column in the data preview and then

select Group By under the Transform tab in the ribbon.



3. You'll perform a count of rows as the aggregation within Group By. You can learn more about Group By capabilities at [Grouping or summarizing rows](#).



Group by [?](#)

Specify the column to group by and the desired output.

Basic Advanced

Group by *

CustomerID

New column name * Column *

Count Count rows

Use fuzzy grouping

[Fuzzy group options](#)

OK Cancel

4. After grouping data in the Orders table, we'll obtain a two-column table with CustomerID and Count as the columns.

Table.Group(#"Navigation 1", {"CustomerID"}, {"Count", each Table.RowCount(_), Int64.Type})

	CustomerID	Count
● Valid	100%	100%
● Error	0%	0%
● Empty	0%	0%
89 distinct, 89 unique		
1 VINET		5
2 TOMSP		6
3 HANAR		14
4 VICTE		10
5 SUPRD		12
6 CHOPS		8
7 RICSU		10
8 WELLI		9
9 HILAA		18
10 ERNSH		30

5. Next, you want to combine data from the Customers table with the Count of Orders per customer. To combine data, select the Customers query in the Diagram View and use the ":" menu to access the **Merge queries as new transformation**.

The screenshot shows the Microsoft Data Factory Power Query Editor interface. On the left, the 'Queries' pane lists 'Customers' and 'Orders'. The 'Customers' query is selected, and its context menu is open, with the 'Merge queries as new' option highlighted with a red box. The main workspace displays the 'Customers' table with various columns and their data distribution. On the right, the 'Query settings' pane shows the 'Source' tab and the 'Applied steps' pane, which includes a step named 'Navigation 1'.

6. Configure the **Merge operation** as shown in the following screenshot by selecting **CustomerID** as the matching column in both tables. Then select **Ok**.

Merge [?](#)

Select tables and matching columns to create a merged table.

Left table for merge *

Customers

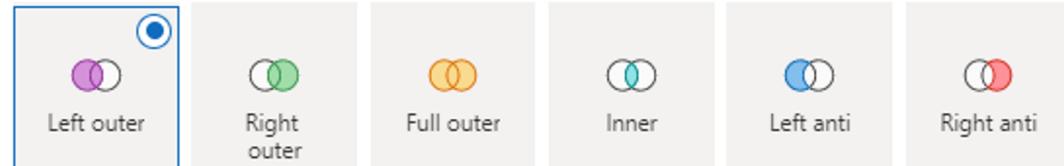
CustomerID	CompanyName	ContactName	ContactTitle	Address
ALFKI	Alfreds Futterkiste	Maria Anders	Sales Representative	Obere Str. 57
ANATR	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	Avda. de la Constitución 2
ANTON	Antonio Moreno Taquería	Antonio Moreno	Owner	Mataderos 2
AROUT	Around the Horn	Thomas Hardy	Sales Representative	120 Hanover Street

Right table for merge *

Orders

CustomerID	Count
VINET	5
TOMSP	6
HANAR	14
VICTE	10
BLAEN	12

Join kind *



Use fuzzy matching to perform the merge

› Fuzzy matching options

The selection matches 89 of 91 rows from the first table

OK

Cancel

- Upon performing the Merge queries as new operation, you'll obtain a new query with all columns from the Customers table and one column with nested data from the Orders table.

The screenshot shows the Microsoft Data Factory Power Query Editor interface. A 'Merge' step is selected, connecting the 'Customers' and 'Orders' tables. The 'Data destination' button is highlighted in red at the bottom right of the ribbon. The main area displays a preview of the merged data with various columns and their statistics.

8. In this example, you're only interested in a subset of columns in the Customers table. You'll select those columns by using the schema view. Enable the schema view within the toggle button on the bottom-right corner of the dataflows editor.

The screenshot shows the Microsoft Data Factory Power Query Editor interface with the schema view enabled. The 'Data destination' button is highlighted in red at the bottom right of the ribbon. The main area displays a preview of the merged data with various columns and their statistics.

9. The schema view provides a focused view into a table's schema information, including column names and data types. Schema view has a set of schema tools available through a contextual ribbon tab. In this scenario, you'll select the **CustomerID**, **CompanyName**, and **Orders (2)** columns, then select the **Remove columns** button, and then select **Remove other columns** in the Schema tools tab.

The screenshot shows the Microsoft Data Factory Power Query interface. A 'Merge' step is selected, which has joined the 'Customers' table with the 'Orders' table. The resulting schema is displayed in the table below. The 'Orders (2)' column is currently selected. In the top ribbon, the 'Schema tools' tab is active. On the far right, the 'Query settings' pane shows the step is named 'Merge' and is of type 'Custom'. The 'Applied steps' section shows the 'Source' step. The bottom right corner of the interface has a 'Remove columns' button, which is highlighted with a red box.

This screenshot shows the same Microsoft Data Factory Power Query interface as the previous one, but with a different focus. The 'Remove other columns' button in the 'Schema tools' ribbon is highlighted with a red box. This button is used to remove all columns except the ones explicitly listed in the schema.

10. The **Orders (2)** column contains nested information resulting from the merge operation you performed a few steps ago. Now, switch back to the data view by selecting the **Show data view** button next to the **Show schema view** button in the bottom-right corner of the UI. Then use the **Expand Column** transformation in the **Orders (2)** column header to select the **Count** column.

The screenshot shows the Microsoft Data Factory Power Query Editor interface. In the center, there's a 'Merge' step connecting two tables: 'Customers' and 'Orders'. The 'Customers' table has 91 rows, and the 'Orders' table has 91 distinct rows. A tooltip 'N/A for this column type' is displayed over the 'Count' column in the preview pane. On the right side, the 'Query settings' pane shows the 'Name' as 'Merge' and the 'Entity type' as 'Custom'. The 'Applied steps' section lists 'Source' and 'Removed other c...'. A tooltip 'Data destination No data destination' is shown near the bottom right.

- As the final operation, you want to rank your customers based on their number of orders. Select the **Count** column and then select the **Rank column** button under the **Add Column** tab in the ribbon.

The screenshot shows the Microsoft Data Factory Power Query Editor interface. The 'Add column' tab is selected in the ribbon. A tooltip 'Rank column' is highlighted in the ribbon bar. The main area shows the same 'Merge' step as before, with the preview pane showing the merged data. The 'Query settings' pane and 'Applied steps' section are also visible on the right.

- Keep the default settings in **Rank Column**. Then select **OK** to apply this transformation.

Rank ?

Specify the column to rank by and the rank criteria.

Basic Advanced

Rank by

Count

Rank criteria

Rank values from high to low

OK

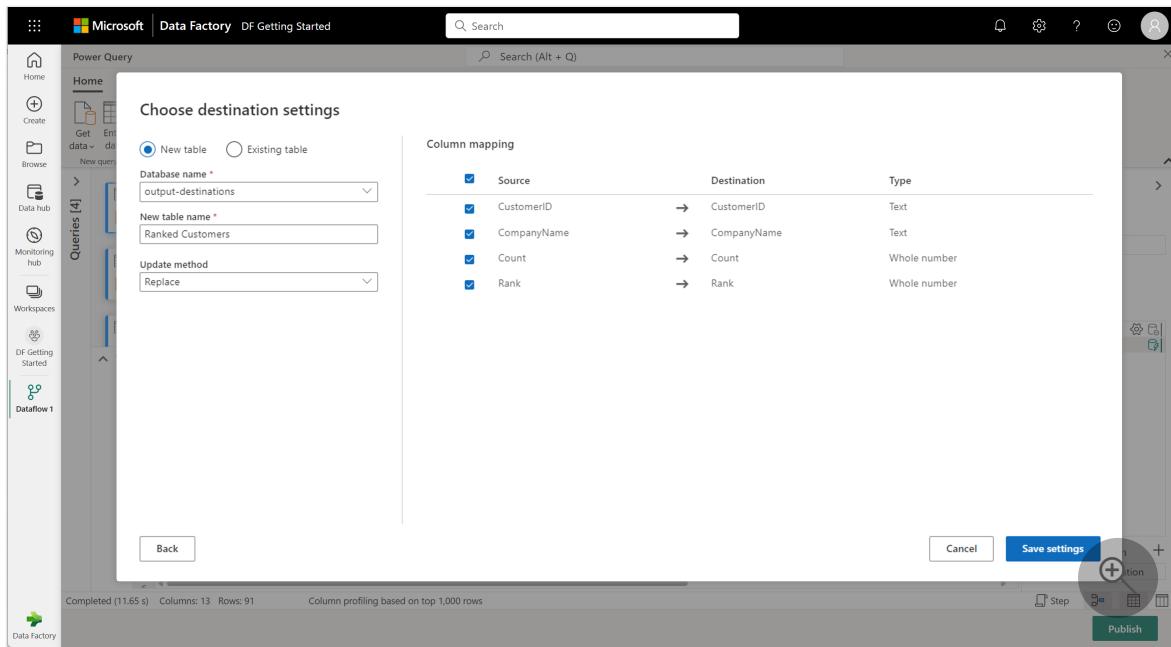
Cancel

13. Now rename the resulting query as **Ranked Customers** using the **Query settings** pane on the right side of the screen.

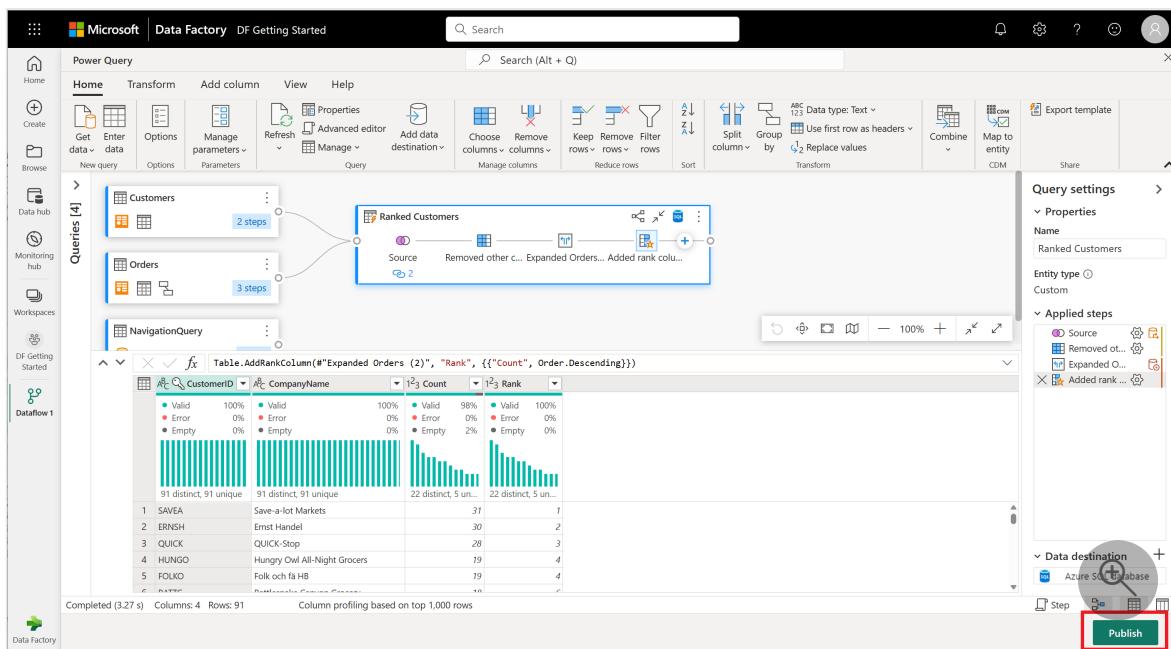
The screenshot shows the Microsoft Data Factory Power Query interface. On the left, there's a sidebar with icons for Home, Create, Browse, Data hub, Monitoring hub, Workspaces, and DF Getting Started. The main area shows two queries: 'Customers' (2 steps) and 'Orders' (3 steps). A third query, 'Ranked Customers', is highlighted with a blue border. It has a 'Source' step pointing to 'Customers' and another step pointing to 'Orders'. The 'Ranked Customers' query has a tooltip: 'Removed other c... Expanded Orders... Added rank col...'. Below the queries is a preview grid with columns: CustomerID, CompanyName, Count, and Rank. The 'Count' and 'Rank' columns are highlighted with green headers. The preview shows data for five rows. The bottom status bar says 'Completed (2.22 s) Columns: 4 Rows: 91'. To the right, the 'Query settings' pane is open, showing 'Properties' with 'Name' set to 'Ranked Customers' (which is highlighted with a red box). Other sections in the pane include 'Entity type' (Custom), 'Applied steps' (listing 'Source', 'Removed ot...', 'Expanded O...', and 'Added rank...'), and 'Data destination' (with a note 'No data destination'). At the bottom right of the pane is a 'Publish' button.

14. You've finished transforming and combining your data. So, you'll now configure its output destination settings. Select **Choose data destination** at the bottom of the **Query settings** pane.

15. For this step, you can configure an output to your Azure SQL database if you have one available, or skip this step if you don't. Within this experience, you're able to configure the destination server, database, and table for your query results, in addition to the update method (Append or Replace).



16. Your dataflow is now ready to be published. Review the queries in the diagram view, and then select **Publish**.



You'll now be returned to the workspace. A spinner icon next to your dataflow name indicates publishing is in progress. Once the publishing completes, your dataflow is ready to refresh!

17. In your workspace, select the **Schedule Refresh** icon.

Name	Type
Northwind sales Dataflow	Dataflow Gen2

18. Turn on the scheduled refresh, select "Add another time", and configure the refresh as shown in the following screenshot.

The screenshot shows the 'Scheduled refresh' configuration interface. Key elements include:

- A toggle switch labeled 'On' is highlighted with a red box.
- A dropdown menu for 'Refresh frequency' set to 'Daily'.
- A dropdown menu for 'Time zone' set to '(UTC) Coordinated Universal Time'.
- A time selector showing '4 00 AM'.
- A button labeled 'Add another time' highlighted with a red box.
- A section for 'Send refresh failure notifications to':
 - A checked checkbox for 'Dataflow owner' highlighted with a red box.
 - An unchecked checkbox for 'These contacts:'.
 - A text input field for 'Enter email addresses'.
- Buttons at the bottom: 'Apply' (highlighted with a red box) and 'Discard'.

Clean up resources

If you're not going to continue to use this dataflow, delete the dataflow using the following steps:

1. Navigate to your Microsoft Fabric workspace.

The screenshot shows the Microsoft Data Factory Browse interface. On the left, there's a sidebar with icons for Home, Create, Browse, Data hub, Monitoring hub, Workspaces, and DF Getting Started. The 'Recent' section is highlighted with a red box. It lists three items: 'DF Getting Started' (Workspace), 'Release Plan Publish Schedule' (Report), and 'pipeline4' (Data pipeline). The 'DF Getting Started' item has a red box around its name.

2. Select the vertical ellipsis next to the name of your dataflow and then select **Delete**.

The screenshot shows a list of dataflows. One dataflow, 'Northwind sales Demo', is selected. A context menu is open next to it, with the 'Delete' option highlighted by a red box. The menu also includes 'Edit', 'Properties', 'Refresh history', and 'Settings'.

3. Select **Delete** to confirm the deletion of your dataflow.

Delete dataflow

×

Are you sure you want to permanently delete Northwind sales Demo? Content from this dataflow that's included in any datasets, reports, dashboards, or tiles will be deleted.

Delete

Cancel

Next steps

The dataflow in this sample shows you how to load and transform data in Dataflow Gen2. You learned how to:

- ✓ Create a Dataflow Gen2.
- ✓ Transform data.
- ✓ Configure destination settings for transformed data.
- ✓ Run and schedule your data pipeline.

Advance to the next article to learn how to create your first data pipeline.

[Quickstart: Create your first data pipeline to copy data](#)

Quickstart: Create your first pipeline to copy data

Article • 05/23/2023

In this quickstart, you build a data pipeline to move a Sample dataset to the Lakehouse. This experience shows you a quick demo about how to use pipeline copy activity and how to load data into Lakehouse.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

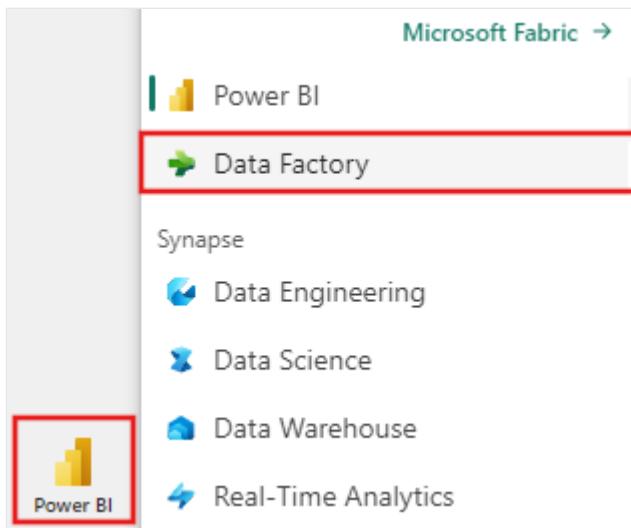
Prerequisites

To get started, you must complete the following prerequisites:

- A Microsoft Fabric tenant account with an active subscription. Create an account for free.
- Make sure you have a Microsoft Fabric enabled Workspace: [Create a workspace](#).

Create a data pipeline

1. Navigate to [Power BI](#).
2. Select the Power BI icon in the bottom left of the screen, then select **Data factory** to open homepage of Data Factory.



3. Navigate to your Microsoft Fabric workspace. If you created a new workspace in the prior Prerequisites section, use this one.

The Microsoft Fabric workspace browser is shown. The left sidebar includes Home, Create, Browse, Data hub, Monitoring hub, Workspaces, and DF Getting Started. The main area shows a 'Recent' section with three items: 'DF Getting Started' (Workspace), 'Release Plan Publish Schedule' (Report), and 'pipeline4' (Data pipeline). The 'DF Getting Started' item is highlighted with a red box.

Name	Type	Ops
DF Getting Started	Workspace	33
Release Plan Publish Schedule	Report	3 h
pipeline4	Data pipeline	23

4. Select **Data pipeline** and then input a pipeline name to create a new pipeline.



Microsoft

Data Factory SampleDFWorkspace

Search



Home



Create



Browse



Workspaces

SampleDFW
workspaceSampleDFPi
plinesample-
dataflow

More...



Data Factory



SampleDFWorkspace



+ New

Upload

...

Filter by keyword

Data pipeline (Preview)

Eventstream (Preview)

Experiment (Preview)

KQL Database (Preview)

KQL Queryset (Preview)

Lakehouse (Preview)

Model (Preview)

Notebook (Preview)

Reflex (Preview)

Report

Spark Job Definition (Preview)

Warehouse (Preview)

Show all

Quickstart dataflow

Ingest data at scale and schedule
data workflows.

Type

Dataset (default)

SQL endpoint

Lakehouse

Data pipeline

Notebook

Data pipeline

Data pipeline

Dataflow

New pipeline



Name

SamplePipeline

Create

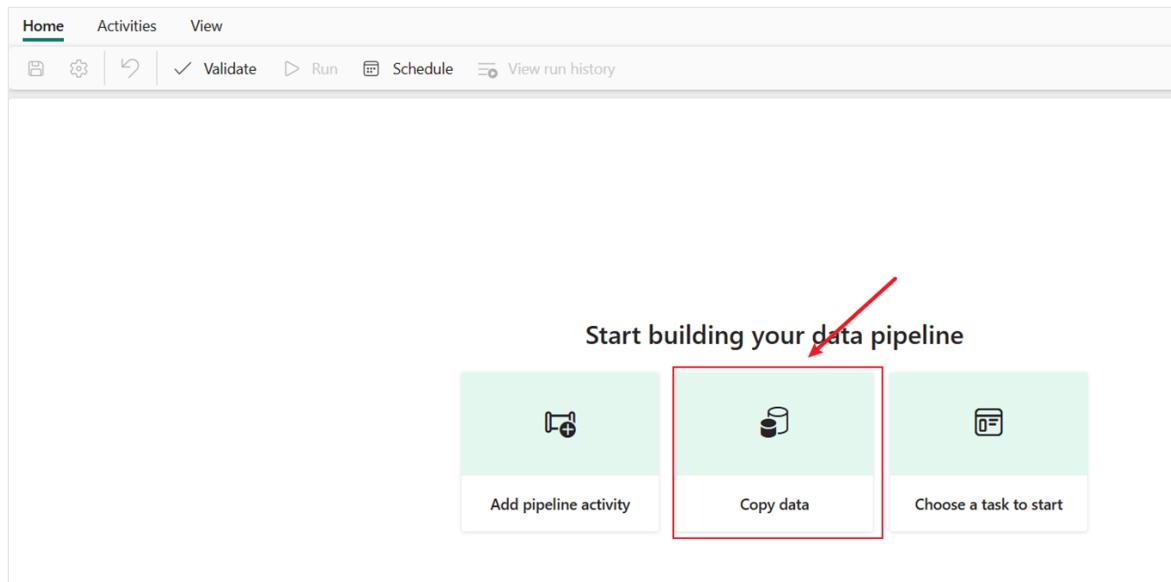
Cancel

Copy data using pipeline

In this session, you start to build your first pipeline by following below steps about copying from a sample dataset provided by pipeline into Lakehouse.

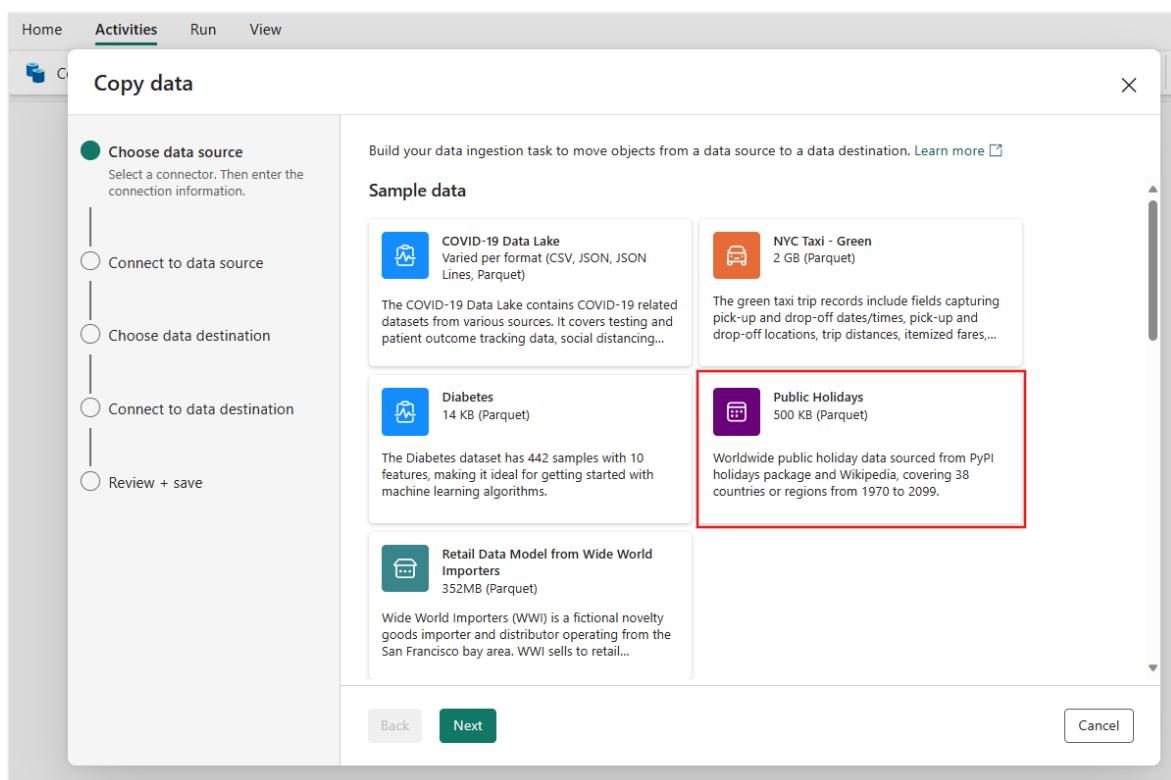
Step 1: Start with the Copy assistant

1. After selecting **Copy data** on the canvas, the **Copy assistant** tool will be opened to get started.



Step 2: Configure your source

1. Choose the **Public Holidays** sample data, and then select **Next**.



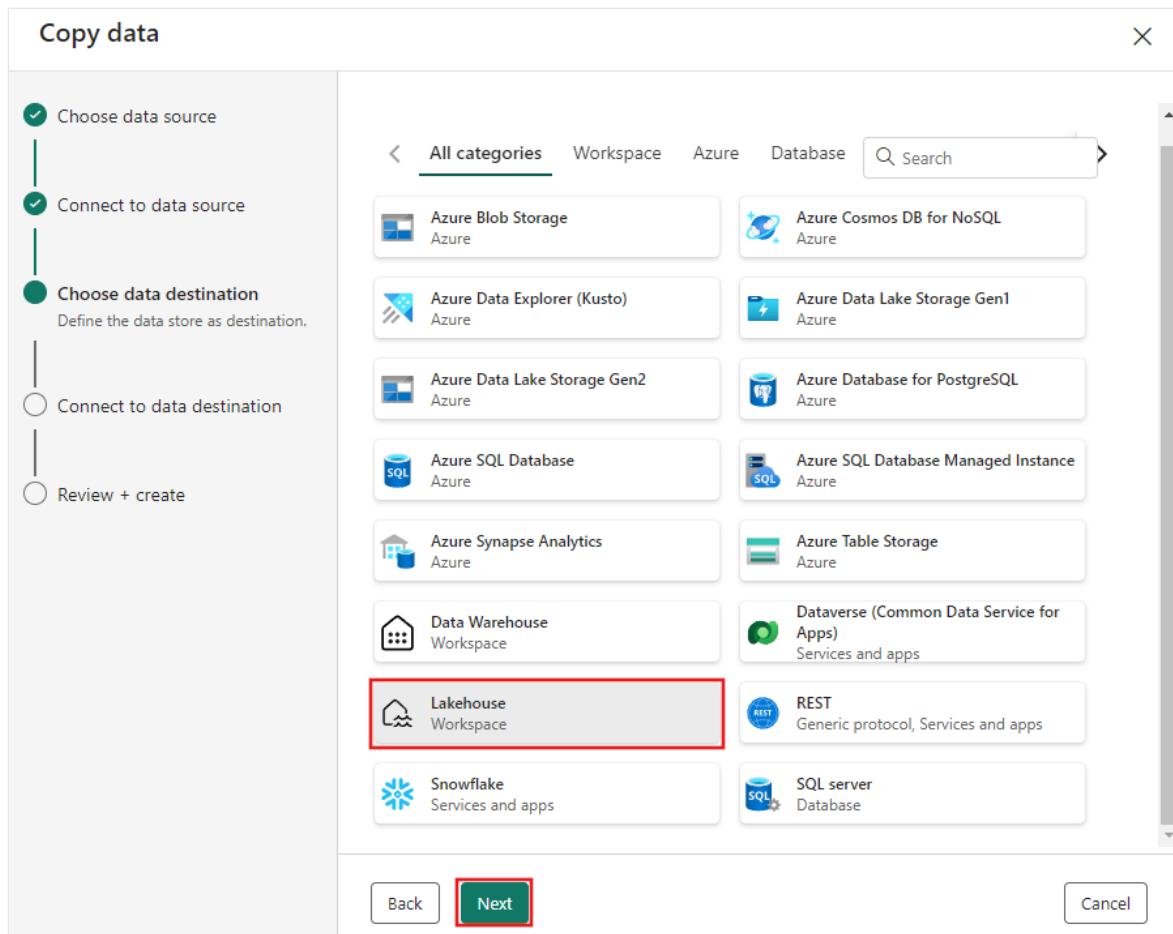
2. On the **Connect to data source** page of the assistant, the preview for the **Public Holidays** sample data is displayed, and then click **Next**.

The screenshot shows the 'Copy data' wizard interface. The top navigation bar includes 'Home', 'Activities' (which is underlined), 'Run', and 'View'. Below this is a toolbar with icons for 'New', 'Import', 'Export', 'Share', and 'Help'. The main area is titled 'Copy data' with a close button 'X'. On the left, a vertical list of steps is shown: 'Choose data source' (with a green checkmark), 'Connect to data source' (selected, highlighted in red), 'Choose data destination', 'Connect to data destination', and 'Review + save'. The 'Connect to data source' step has a sub-instruction: 'Select, preview, and choose the data.' To the right of this list is a 'Select a dataset' section with a dropdown menu showing 'Public Holidays' (also highlighted in red). Further right is a 'Preview data: Public Holidays' section containing a table with 10 rows of data. At the bottom are 'Back', 'Next' (in a green box), and 'Cancel' buttons.

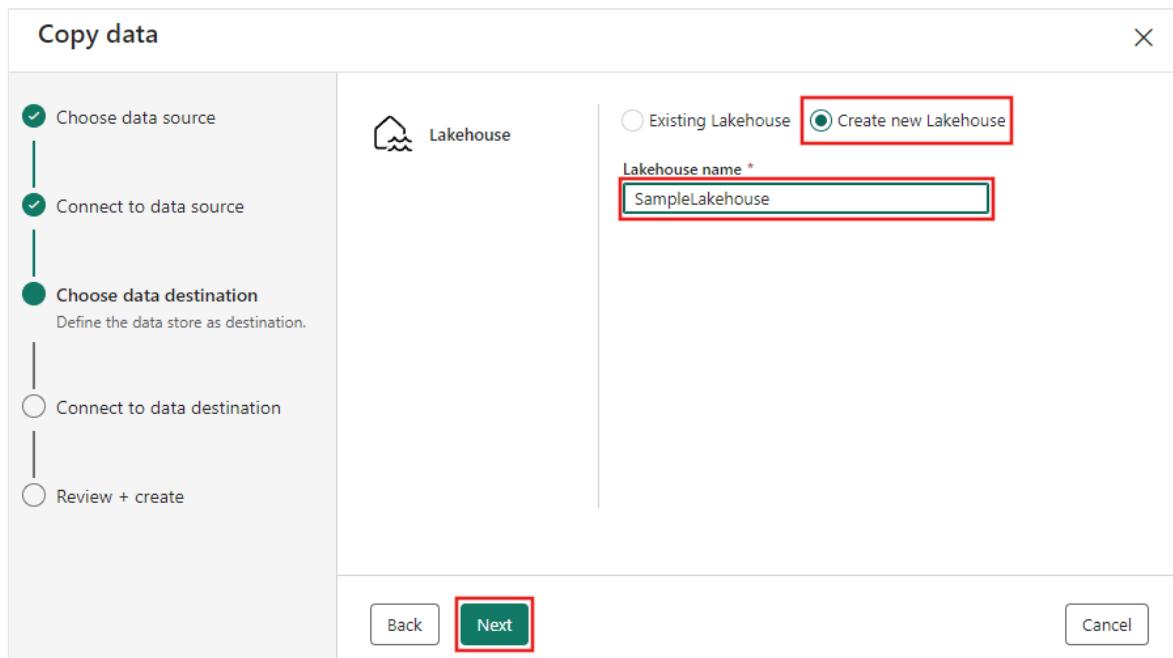
	abc countryOrRegion	abc holidayName	abc normalizeHolidayName	X✓ isPaidTimeOff
1	Argentina	Año Nuevo [New Year's Day]	Año Nuevo [New Year's Day]	
2	Australia	New Year's Day	New Year's Day	
3	Austria	Neujahr	Neujahr	
4	Belgium	Nieuwjaarsdag	Nieuwjaarsdag	
5	Brazil	Ano novo	Ano novo	
6	Canada	New Year's Day	New Year's Day	
7	Colombia	Año Nuevo [New Year's Day]	Año Nuevo [New Year's Day]	
8	Croatia	Nova Godina	Nova Godina	
9	Czech	Nový rok	Nový rok	
10	Denmark	Nytårsdag	Nytårsdag	

Step 3: Configure your destination

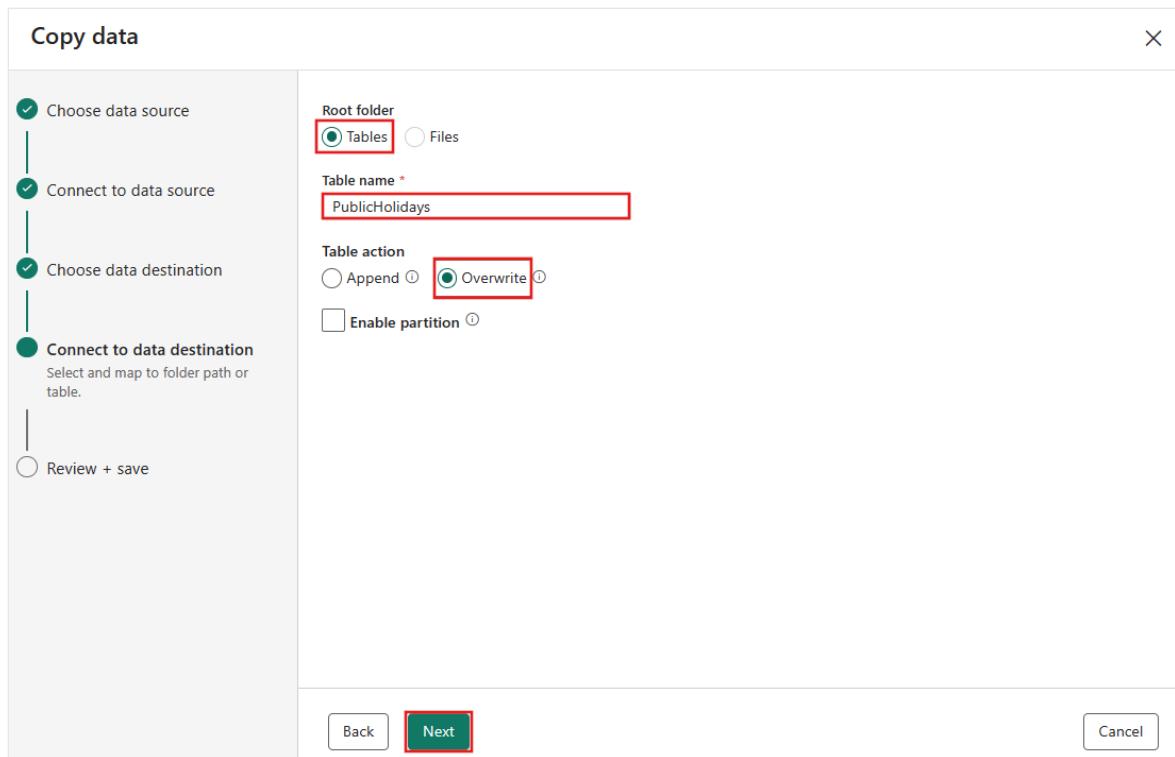
1. Select **Lakehouse** and then **Next**.



2. Select **Create new Lakehouse**, and enter a **Lakehouse name**, then select **Next**.

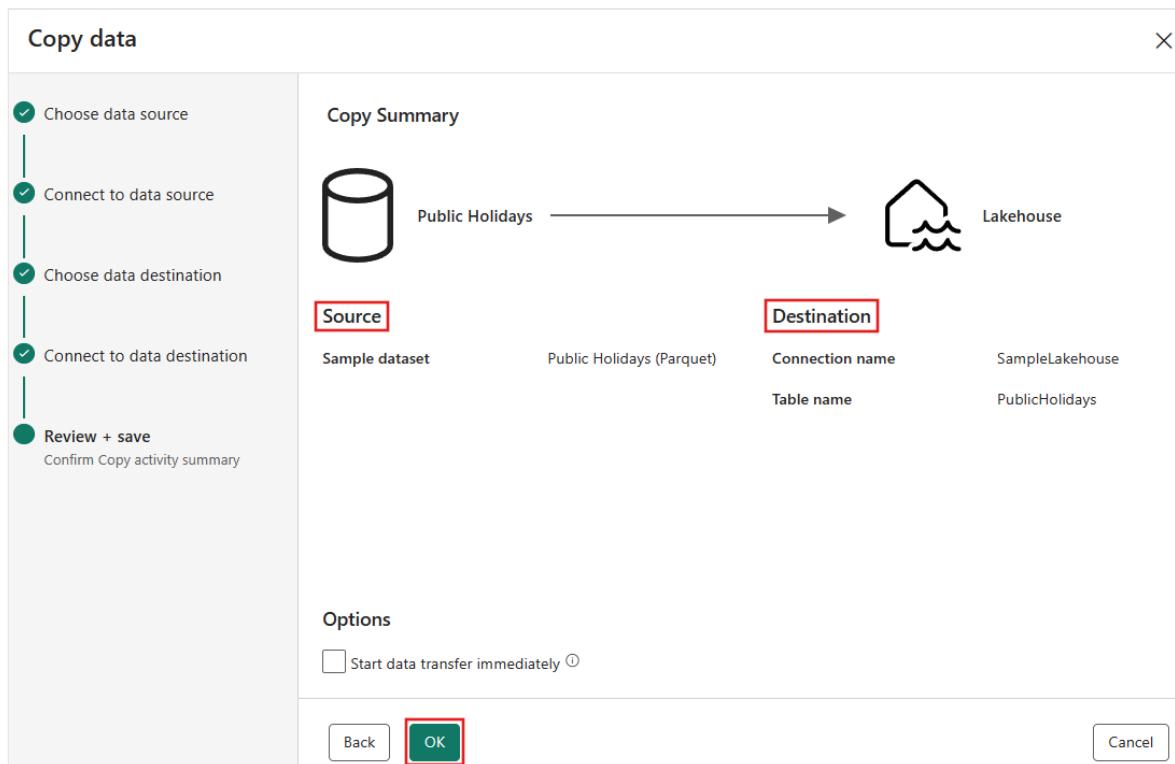


3. Configure and map your source data to the destination Lakehouse table. Select **tables** and provide a **Table name** under **Root folder**, then choose the **Overwrite** option for **Table action**, and select **Next**.



Step 4: Review and create your copy activity

1. Review your copy activity settings in the previous steps and select OK to finish. Or you can revisit the previous steps in the tool to edit your settings, if needed.



2. The Copy activity is added to your new data pipeline canvas. All settings including advanced settings for the activity are available in the tabs below the pipeline canvas when the created **Copy data** activity is selected.

The screenshot shows the Azure Data Factory Home tab. At the top, there are tabs for Home, Activities, Run, and View. Below the tabs is a toolbar with icons for Save, Edit, Settings, Refresh, Validate, Run, Schedule, View run history, and Copy data. A red box highlights the 'Copy data' icon. In the main area, a pipeline named 'Copy_glk' is displayed within a green header bar labeled 'Copy data'. The pipeline details are shown in a card with a red border:

General	
Name *	Copy_glk
Description	(empty)
Timeout ⓘ	0.12:00:00
Retry ⓘ	0
> Advanced	

Run and schedule your data pipeline

1. Switch to the **Home** tab and select **Run**. A confirmation dialog is displayed. Then select **Save and run** to start the activity.

The screenshot shows the Azure Data Factory Home tab. The 'Run' button in the toolbar is highlighted with a red box. A pipeline named 'Copy_r5e' is visible in the list. A 'Save and run?' dialog box is open in the foreground, containing the following text:

You have unsaved changes. If you continue, the pipeline will be saved and run.

3

Save and run Cancel

2. You can monitor the running process and check the results on the **Output** tab below the pipeline canvas. Select the run details button (with the glasses icon highlighted) to view the run details.

The screenshot shows the Azure Data Factory interface. At the top, there's a navigation bar with 'Home' (underlined), 'Activities', and 'View'. Below the navigation bar are several buttons: 'Validate' (with a checkmark icon), 'Cancel' (with a circular icon), 'Schedule' (with a calendar icon), and 'View run history' (with a history icon). To the right of these buttons is a search icon and a '+' icon for creating new pipelines. On the left side of the main area, there's a pipeline canvas with a single activity named 'Copy data' (represented by a blue cylinder icon) with the name 'Copy_gjk' next to it. A status indicator shows a green checkmark. To the right of the canvas is a vertical toolbar with icons for search, add, minus, copy, and up/down arrows. Below the canvas, a tab bar has 'Parameters', 'Variables', 'Settings', and 'Output' (which is underlined and highlighted with a red box). Underneath the tab bar, the text 'Pipeline run ID: 46410ce6-eae3-469b-95a6-3c4ba545d7a9' is displayed, followed by a refresh icon and a help icon. A table titled 'Run details' follows, showing the following data:

Name	Type	Run start	Duration	Status
Copy_gjk	Copy data	3/2/2023, 3:54:27 PM	00:03:20	In progress

3. The run details show how much data was read and written and various other details about the run.

Copy data details

Copy_2wo

Source	Destination
 Azure Blob Storage	 Lakehouse
Data read: ⓘ 394.593 KB	Data written: ⓘ 298.647 KB
Files read: ⓘ 1	Files written: ⓘ 1
Rows read: 69,557	Rows written: ⓘ 69,557

Status	 Succeeded
Start time	5/22/2023, 2:27:40 PM

Pipeline run activity ID	06091bae-a3f6-42ae-a668-bc3d47d98a17
Throughput	65.765 KB/s
Total duration	00:00:14

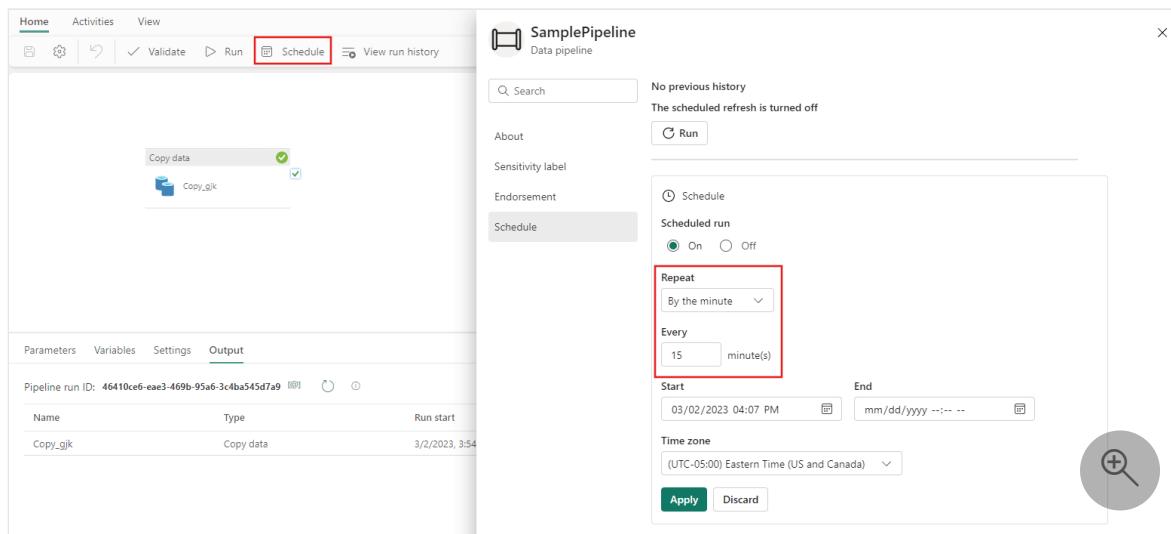
[Duration breakdown](#)

[Advanced](#)

[Close](#)

4. You can also schedule the pipeline to run with a specific frequency as required.

Below is an example scheduling the pipeline to run every 15 minutes.



The screenshot shows the Azure Data Factory pipeline editor interface. The top navigation bar includes Home, Activities, View, Validate, Run, Schedule, and View run history. The main area displays a pipeline named "SamplePipeline" with a single activity: "Copy data" (Copy_Blk). The "Schedule" tab is selected. On the right, the "Schedule" section is open, showing the "Scheduled run" status is "On". The "Repeat" dropdown is set to "By the minute" and "Every 15 minute(s)". The "Start" field is set to "03/02/2023 04:07 PM". The "End" field is set to "mm/dd/yyyy ---- --". The "Time zone" is "(UTC-05:00) Eastern Time (US and Canada)". At the bottom right is a circular "Add" button with a plus sign.

Next steps

The pipeline in this sample shows you how to copy sample data to Lakehouse. You learned how to:

- ✓ Create a data pipeline.
- ✓ Copy data with the Copy Assistant.
- ✓ Run and schedule your data pipeline.

Next, advance to learn more about monitoring your pipeline runs.

[How to monitor pipeline runs in Microsoft Fabric](#)

Quickstart: Move and transform data with dataflows and data pipelines

Article • 05/23/2023

In this tutorial, you'll see how the dataflow and data pipeline experience can create a powerful and comprehensive Data factory solution.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Prerequisites

To get started, you must complete the following prerequisites:

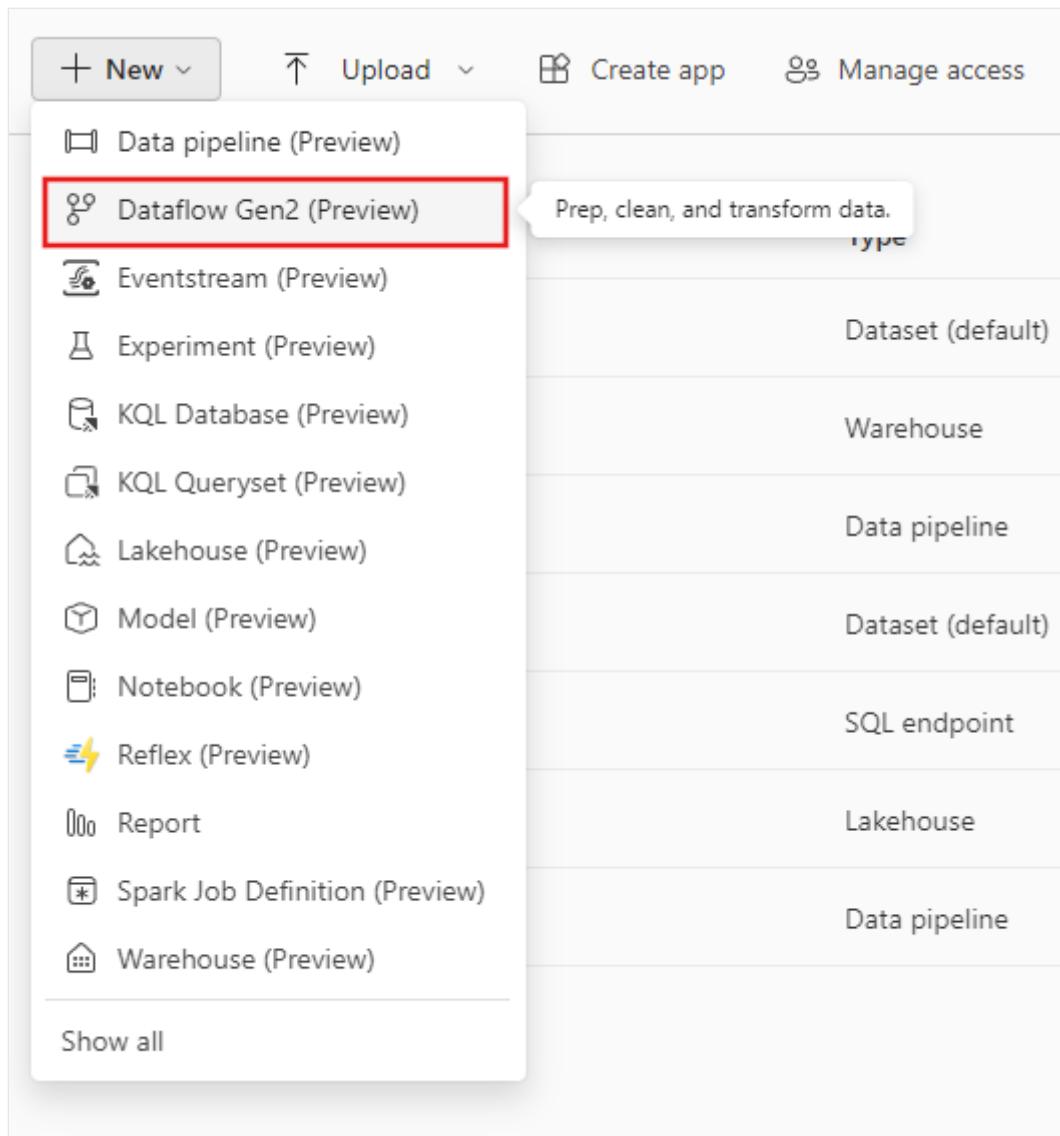
- A tenant account with an active subscription. Create a [free account](#).
- Make sure you have a Microsoft Fabric enabled Workspace: [Create a workspace](#) that isn't the default My Workspace.
- [An Azure SQL database with table data](#).
- [A Blob Storage account](#).

Transform data with dataflows

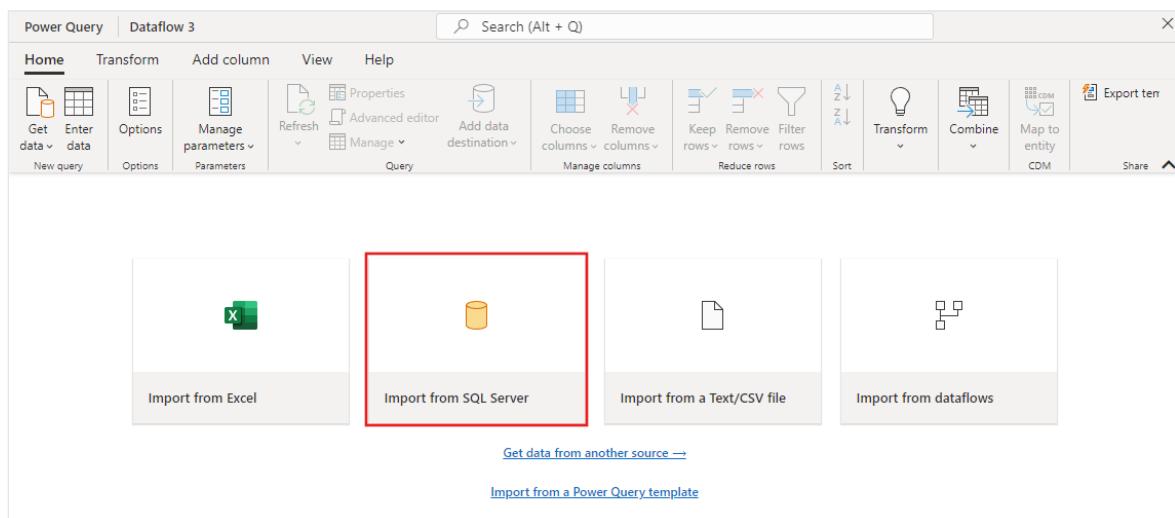
Follow these steps to set up your dataflow.

Step 1: Create a dataflow

1. Choose your Fabric enabled workspace, and then select **New**. Then select **Dataflow Gen2**.



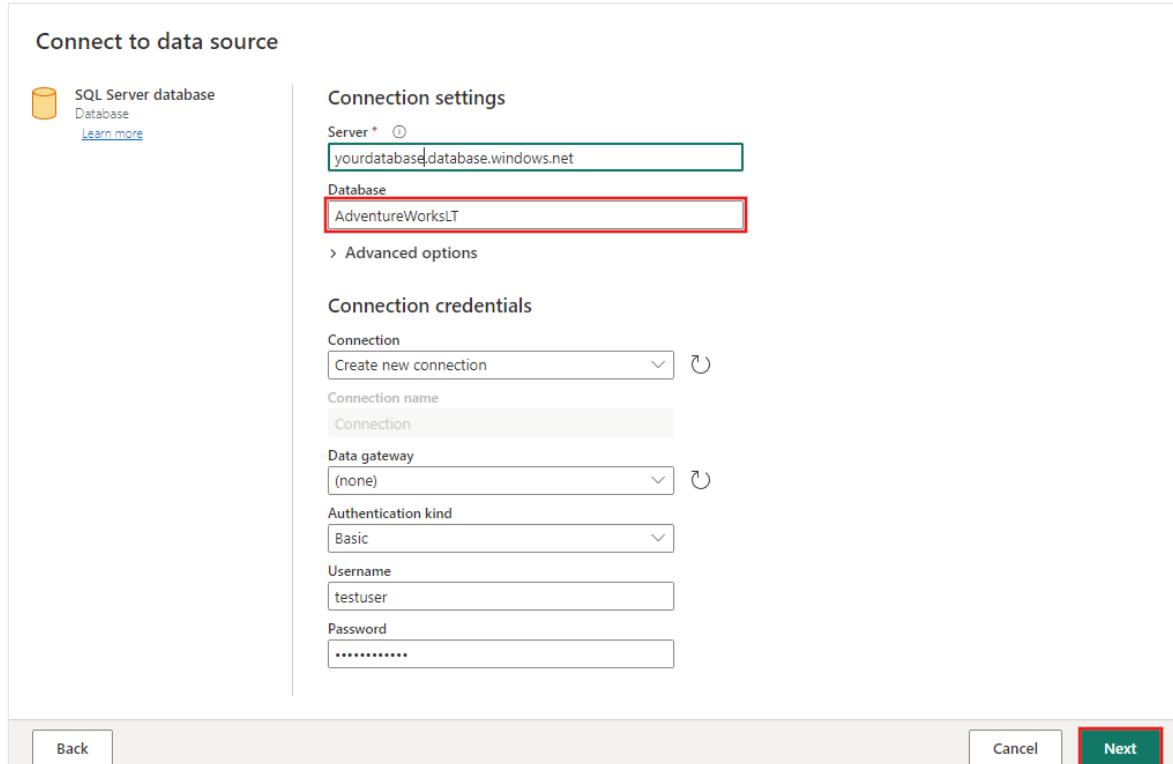
2. The dataflow editor window appears. Select the **Import from SQL Server** card.



Step 2: Get data

1. On the **Connect to data source** dialog presented next, enter the details to connect to your Azure SQL database, then select **Next**. For this example, you use the

AdventureWorksLT sample database configured when you set up the Azure SQL database in the prerequisites.

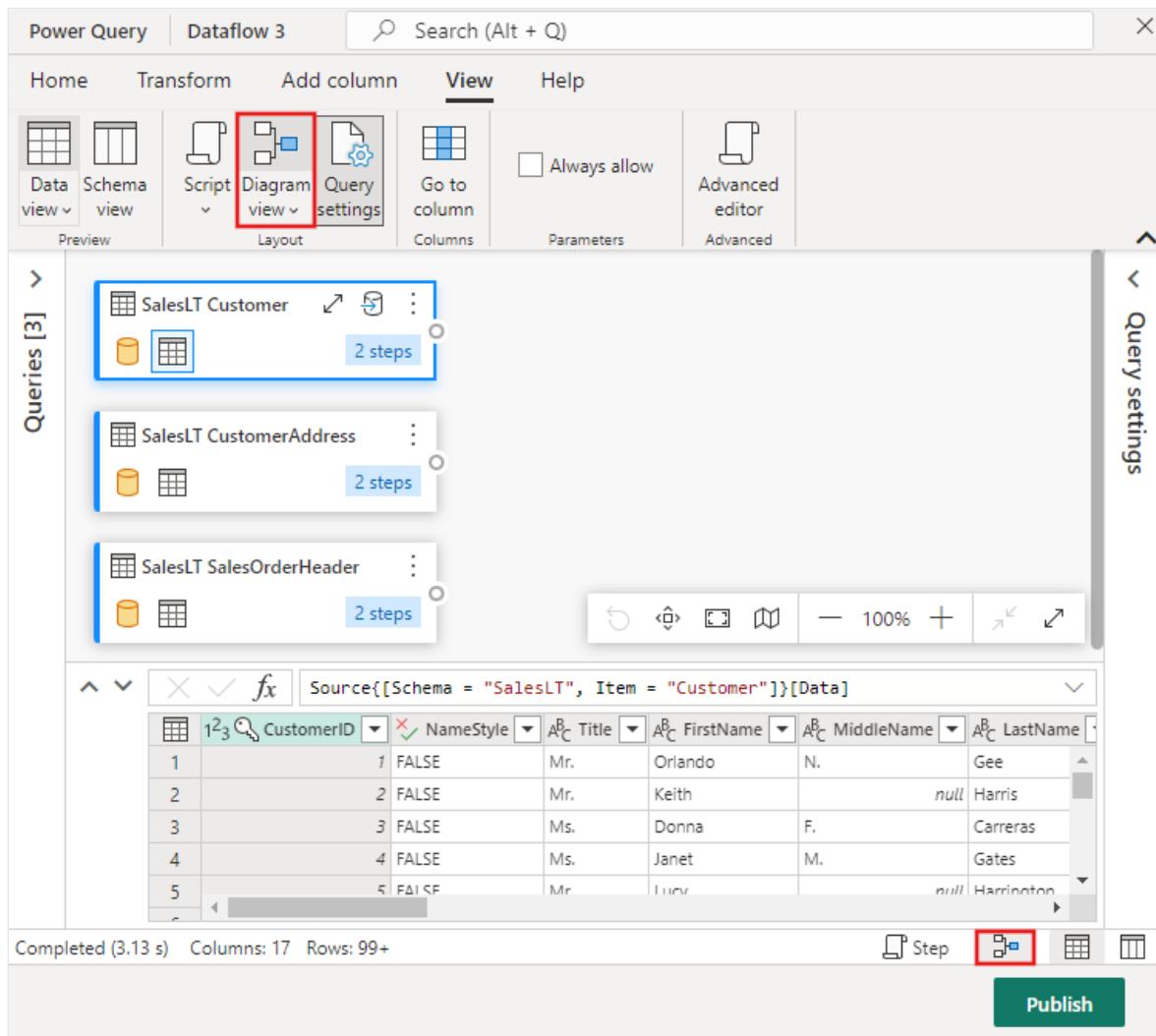


2. Select the data you'd like to transform and then select **Create**. For this quickstart, select **SalesLT.Customer** from the **AdventureWorksLT** sample data provided for Azure SQL DB, and then the **Select related tables** button to automatically include two other related tables.

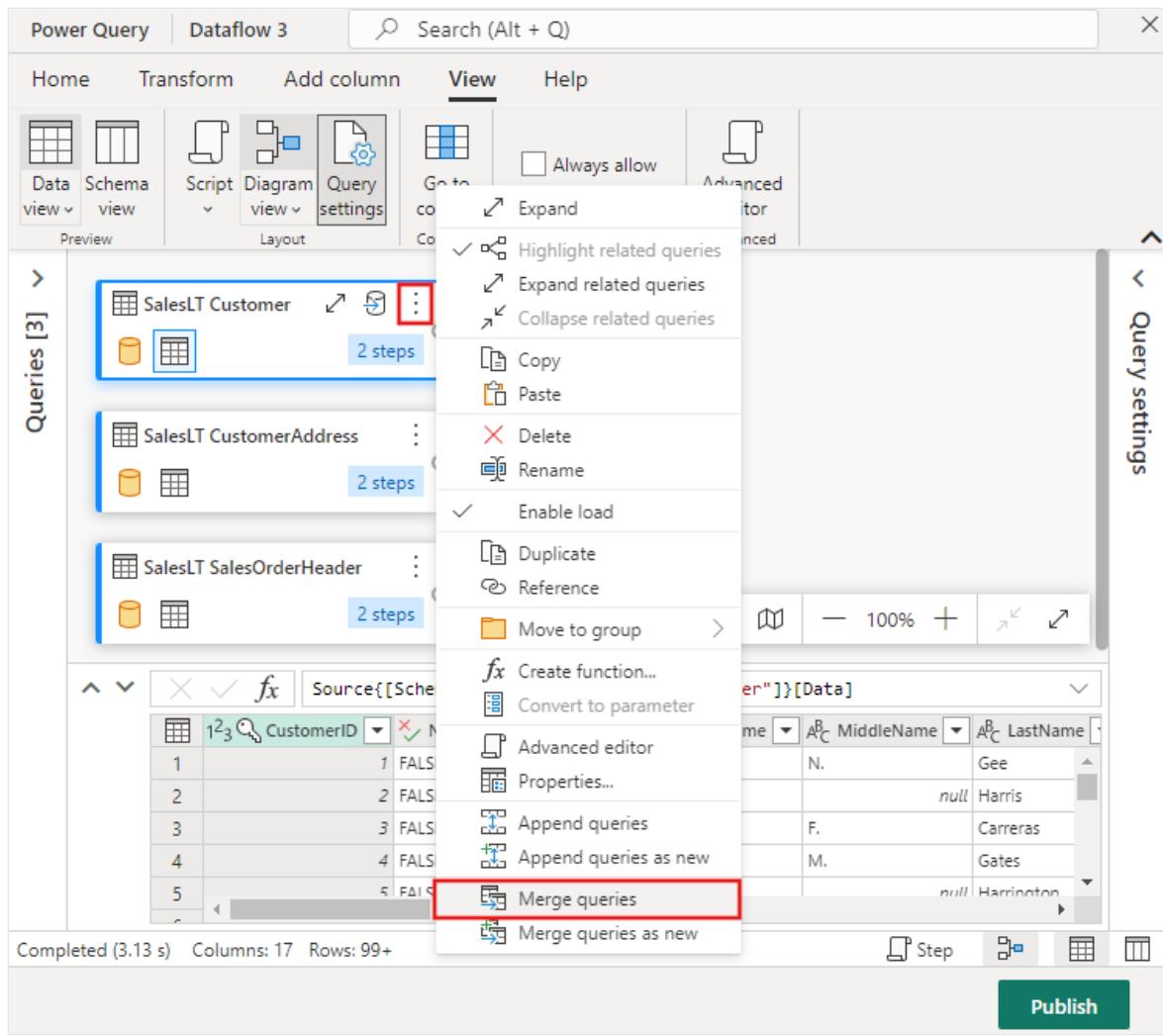
CustomerID	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	CompanyName
1	FALSE	Mr.	Orlando	N.	Gee	null	A Bike Store
2	FALSE	Mr.	Keith		Harris	null	Progressive Sports
3	FALSE	Ms.	Donna	F.	Carreras	null	Advanced Bike Components
4	FALSE	Ms.	Janet	M.	Gates	null	Modular Cycle Systems
5	FALSE	Mr.	Lucy		Harrington	null	Metropolitan Sports Supply
6	FALSE	Ms.	Rosmarie	J.	Carroll	null	Aerobic Exercise Company
7	FALSE	Mr.	Dominic	P.	Gash	null	Associated Bikes
10	FALSE	Ms.	Kathleen	M.	Garza	null	Rural Cycle Emporium
11	FALSE	Ms.	Katherine		Harding	null	Sharp Bikes
12	FALSE	Mr.	Johnny	A.	Caprio	Jr.	Bikes and Motorbikes
16	FALSE	Mr.	Christopher	R.	Beck	Jr.	Bulk Discount Store
18	FALSE	Mr.	David	J.	Liu	null	Catalog Store
19	FALSE	Mr.	John	A.	Beaver	null	Center Cycle Shop
20	FALSE	Ms.	Jean	P.	Handley	null	Central Discount Store
21	FALSE		Jinghao		Liu	null	Chic Department Stores
22	FALSE	Ms.	Linda	E.	Burnett	null	Travel Systems
23	FALSE	Mr.	Kerim		Hanif	null	Bike World
24	FALSE	Mr.	Kevin		Liu	null	Eastside Department Store
25	FALSE	Mr.	Donald	L.	Blanton	null	Coalition Bike Company
28	FALSE	Ms.	Jackie	E.	Blackwell	null	Commuter Bicycle Store
29	FALSE	Mr.	Bryan		Hamilton	null	Cross-Country Riding Supplies
30	FALSE	Mr.	Todd	R.	Logan	null	Cycle Merchants

Step 3: Transform your data

1. If it isn't selected, select the **Diagram view** button along the status bar at the bottom of the page, or select **Diagram view** under the **View** menu at the top of the Power Query editor. Either of these options can toggle the diagram view.



2. Right-click your **SalesLT Customer** query, or select the vertical ellipsis on the right of the query, then select **Merge queries**.



3. Configure the merge by selecting the **SalesOrderHeader** table as the right table for the merge, the **CustomerID** column from each table as the join column, and **Left outer** as the join kind. Then select **Ok** when to add the merge query.

Merge



Select a table and matching columns to create a merged table.

SalesLT Customer

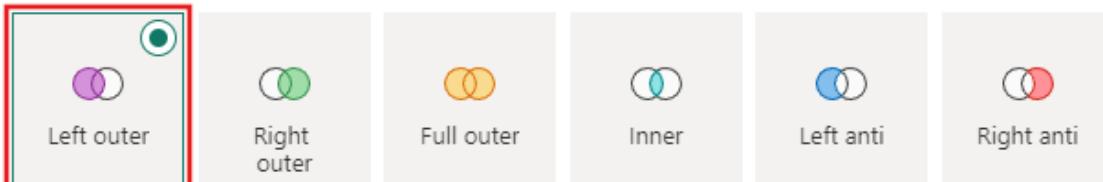
1 ^B _C CustomerID	✓ NameStyle	A ^B _C Title	A ^B _C FirstName	A ^B _C MiddleName	A ^B _C LastName	A ^B _C
1	FALSE	Mr.	Orlando	N.	Gee	
2	FALSE	Mr.	Keith		null	Harris
3	FALSE	Ms.	Donna	F.		Carreras
4	FALSE	Ms.	Janet	M.		Gates

Right table for merge *

SalesLT SalesOrderHeader

A ^B _C AccountNumber	1 ^B _C CustomerID	1 ^B _C ShipToAddressID	1 ^B _C BillToAddressID	A ^B _C ShipMethod	A ^B _C
10-4020-000609	29847		1092	1092	CARGO TRANSP..
10-4020-000106	30072		640	640	CARGO TRANSP..
10-4020-000340	30113		653	653	CARGO TRANSP..
10-4020-000582	29485		1086	1086	CARGO TRANSP..

Join kind *



Use fuzzy matching to perform the merge

› Fuzzy matching options

 The selection matches 32 of 847 rows from the first table

OK

Cancel

4. Select the **Add data destination** button, which looks like a database symbol with an arrow above it, from the new merge query you just created. Then select **Azure SQL database** as the destination type.

The screenshot shows the Microsoft Power Query Editor interface. At the top, the ribbon has tabs: Home, Transform, Add column, View, and Help. Under the View tab, there are buttons for Data view, Schema view, Script, Diagram view, Query settings, Go to column, Columns, Always allow, Parameters, Advanced editor, and Advanced. The main area displays a query flow with three steps:

- Step 1: SalesLT CustomerAddress (2 steps)
- Step 2: SalesLT SalesOrderHeader (2 steps)
- Step 3: SalesLT Cust... (2 steps)

Step 3 is currently selected. A context menu is open next to it, listing connection options: Azure SQL database (selected and highlighted with a red box), Azure Data Explorer (Kusto), and Azure Synapse Analytics (SQL DW). Below the queries, a preview pane shows the results of the nested join. The preview table has columns: ModifiedDate, SalesLT.CustomerAddress, SalesLT.SalesOrderHeader, and SalesLT SalesOrderHeader. The data shows five rows of sample data. At the bottom of the editor, status information says "Completed (2.13 s) Columns: 18 Rows: 99+", and there are buttons for Step, Publish, and other actions.

5. Provide the details for your Azure SQL database connection where the merge query will be published. In this example you can use the **AdventureWorksLT** database we used as the data source for the destination too.

Connect to data destination

Azure SQL database
Azure [Learn more](#)

Connection settings

Server * ⓘ
yourserver.database.windows.net

Database
AdventureWorksLT

Connection credentials

Connection
Create new connection ⚡

Connection name
Connection ⚡

Data gateway
(none) ⚡

Authentication kind
Basic ⚡

Username
YourUser

Password

Cancel **Next**

6. Choose a database to store the data, and provide a table name, then select **Next**.

Choose destination target

New table Existing table

Search

Display options ⚡

Azure SQL database [1]
AdventureWorksLT [12]

A new table will be created in database AdventureWorksLT

Table name *
CustomerSalesOrders

Back Cancel **Next**

7. You can leave the default settings on the **Choose destination settings** dialog, and just select **Save settings** without making any changes here.

Choose destination settings

Update method



Column mapping

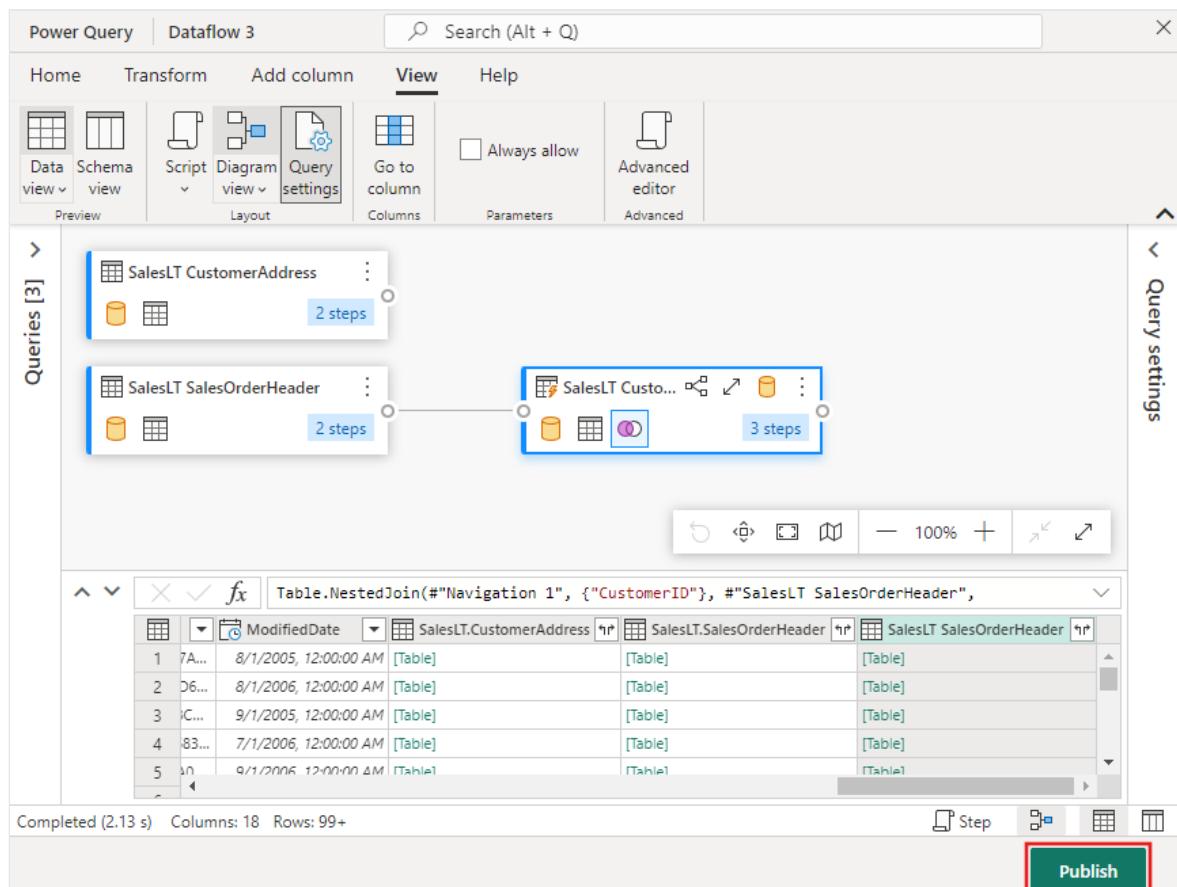
Source	Destination	Type
CustomerID	CustomerID	Whole number
NameStyle	NameStyle	True/False
Title	Title	Text
FirstName	FirstName	Text
MiddleName	MiddleName	Text
LastName	LastName	Text
Suffix	Suffix	Text
CompanyName	CompanyName	Text

Back

Cancel

Save settings

8. Select Publish back on the dataflow editor page, to publish the dataflow.

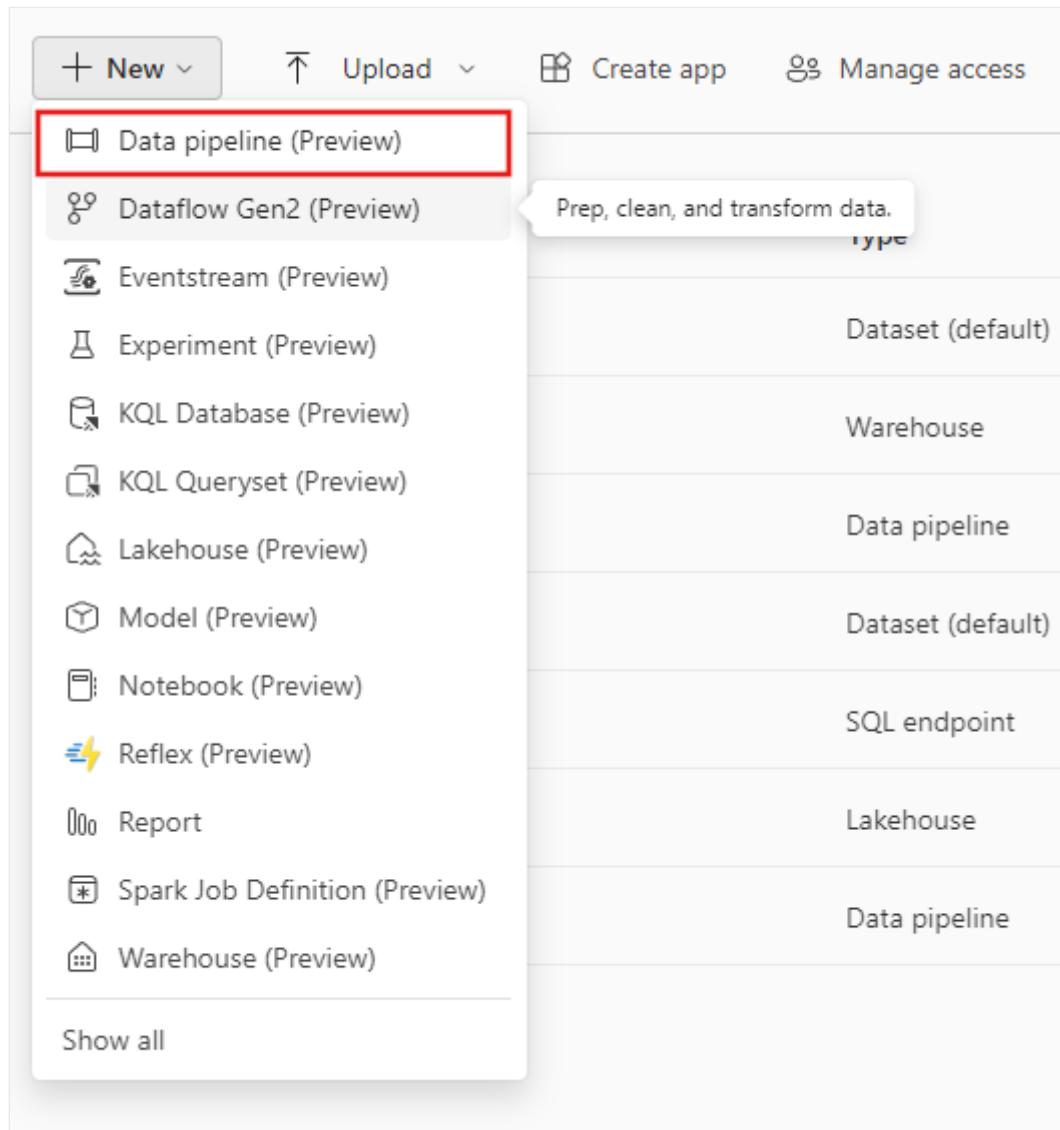


Move data with data pipelines

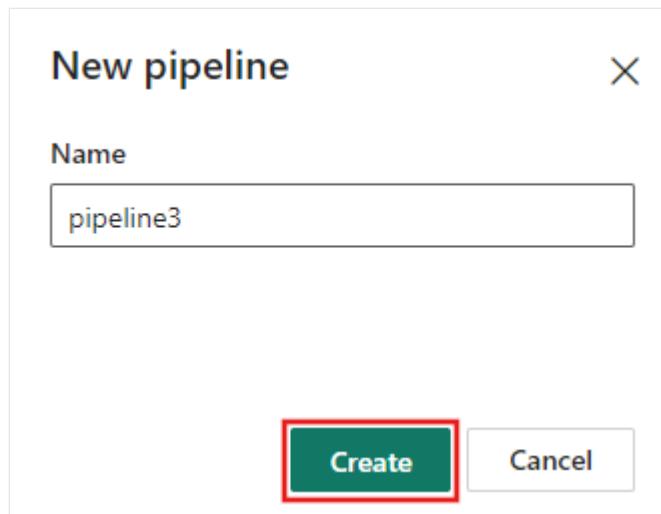
Now that you created a dataflow gen2, you can operate on it in a pipeline. In this example, we will copy the data generated from the dataflow into text format in an Azure Blob Storage account.

Step 1: Create a new data pipeline

1. From your workspace, select **New**, and then select **Data pipeline**.

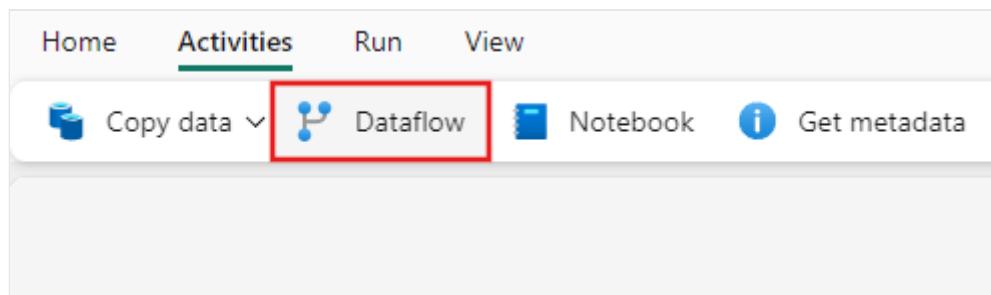


2. Name your pipeline then select **Create**.

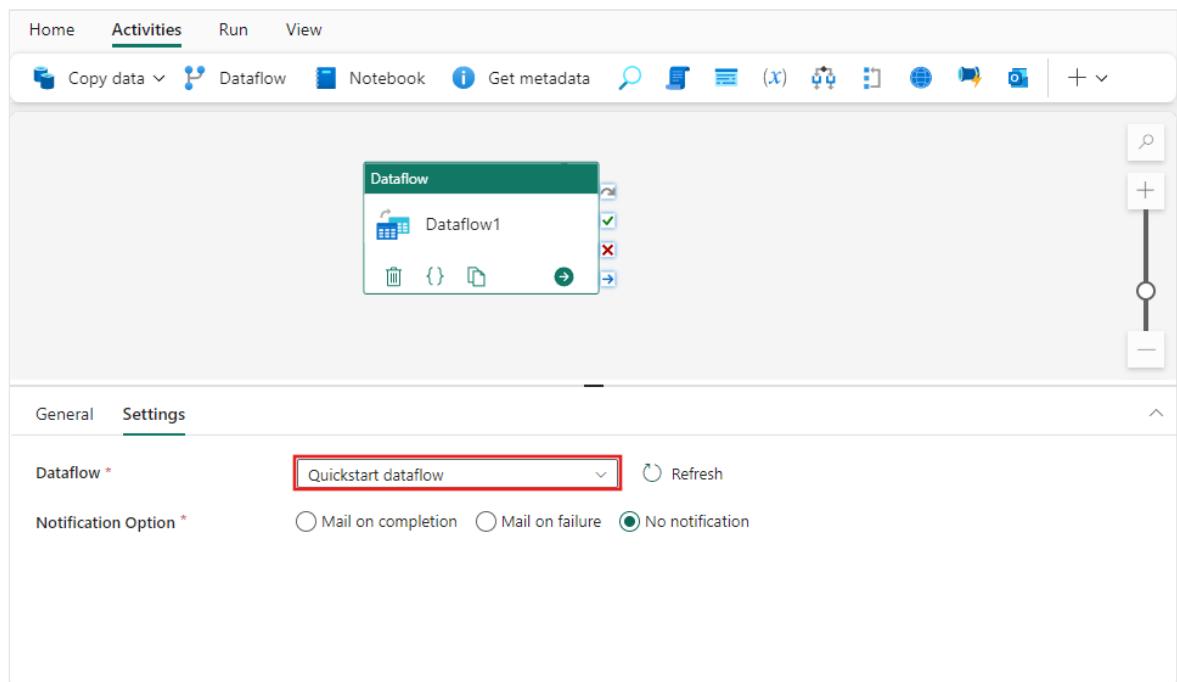


Step 2: Configure your dataflow

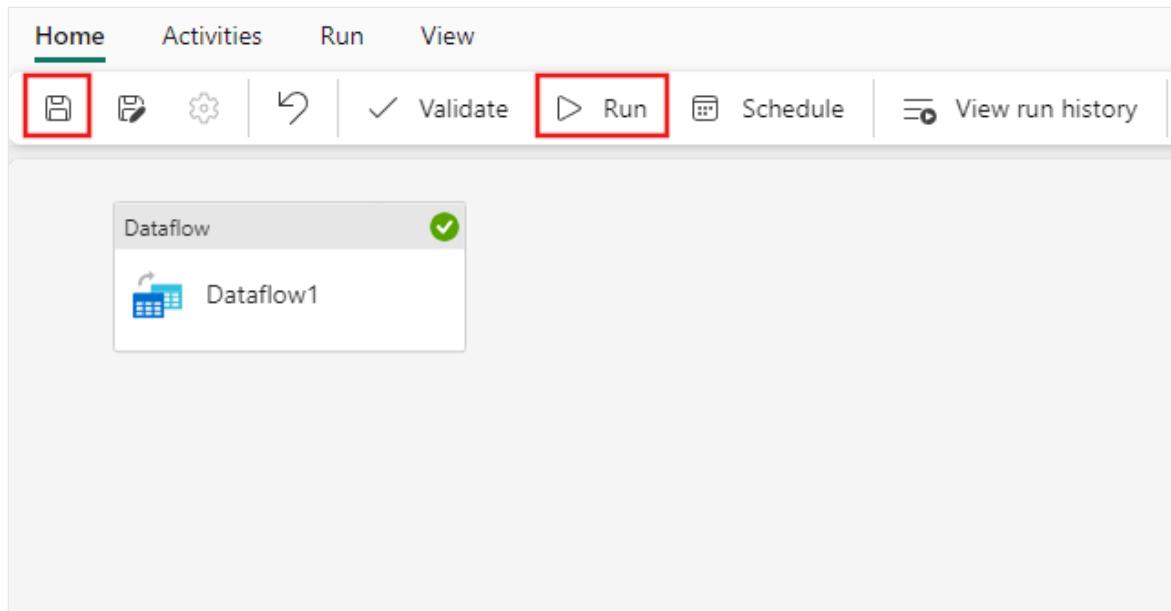
1. Add a new dataflow activity to your data pipeline by selecting **Dataflow** in the **Activities** tab.



2. Select the dataflow on the pipeline canvas, and then the **Settings** tab. Choose the dataflow you created previously from the drop-down list.

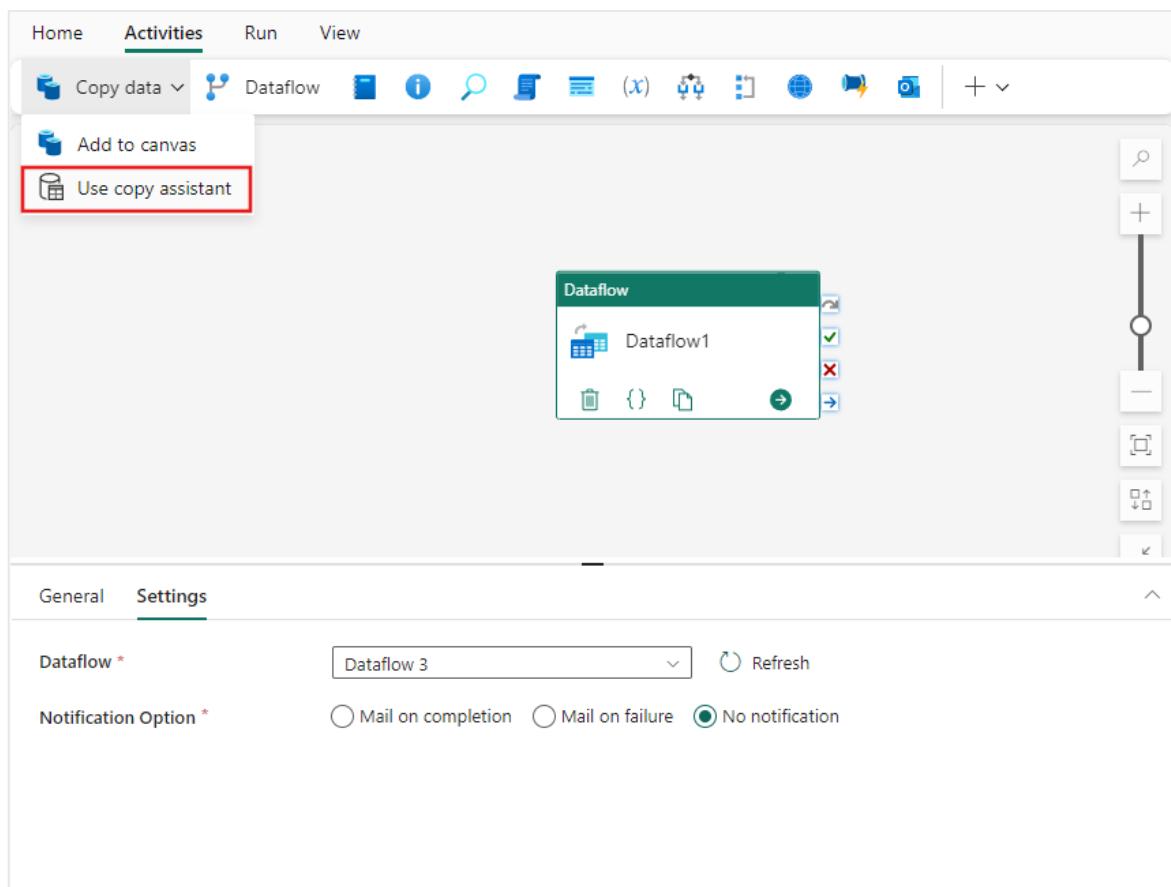


3. Select **Save**, and then **Run** to run the dataflow to initially populate its merged query table you designed in the prior step.

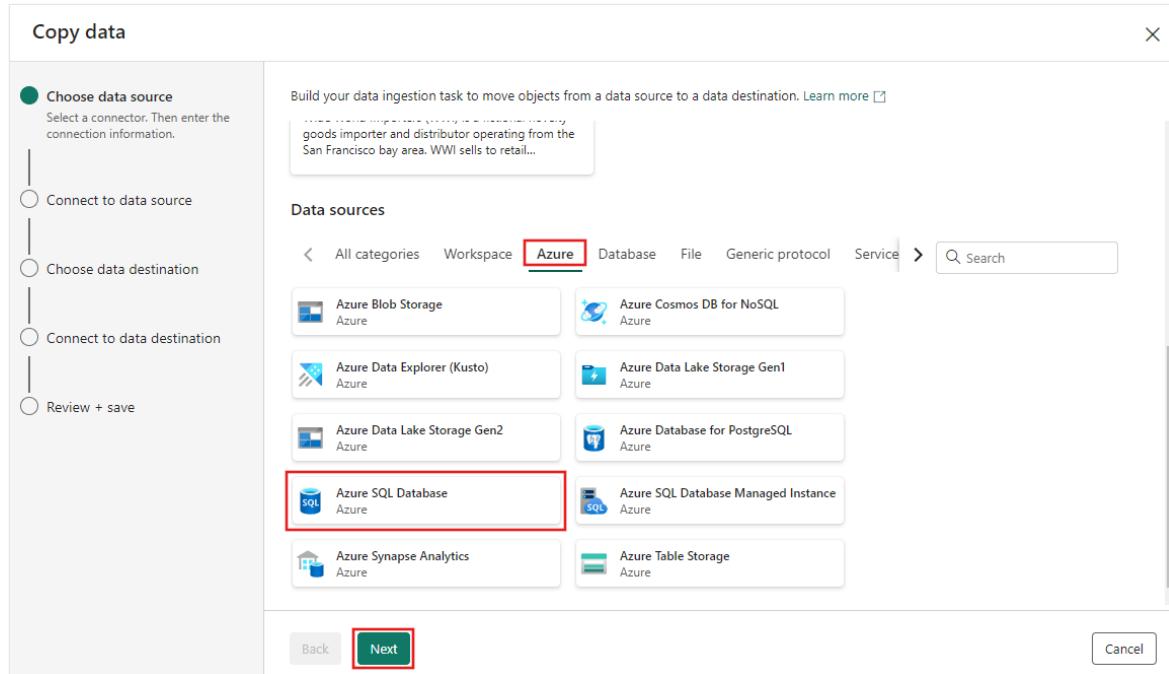


Step 3: Use the copy assistant to add a copy activity

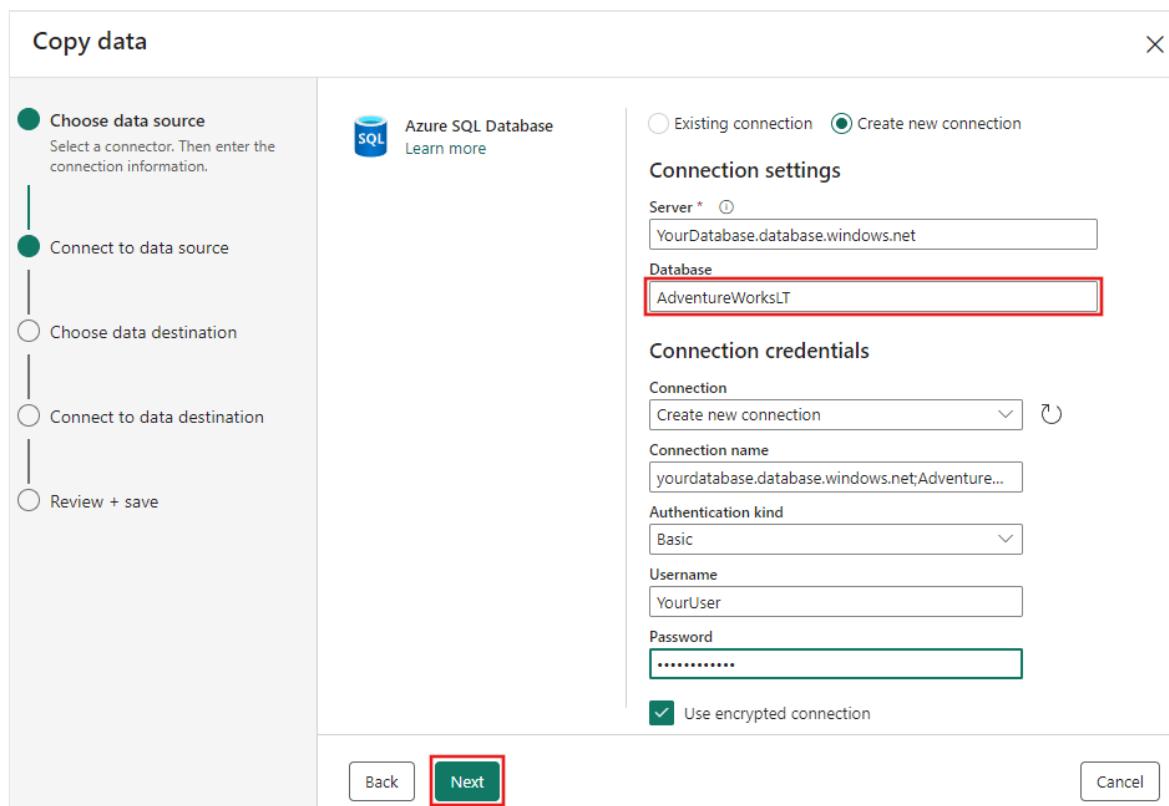
1. Select **Copy data** on the canvas to open the **Copy Assistant** tool to get started. Or select **Use copy assistant** from the **Copy data** drop down list under the **Activities** tab on the ribbon.



2. Choose your data source by selecting a data source type. In this tutorial, you'll use the Azure SQL Database used previously when you created the dataflow to generate a new merge query. Scroll down below the sample data offerings and select the **Azure** tab under **Data sources**, then **Azure SQL Database**. Then select **Next** to continue.



3. Create a connection to your data source by selecting **Create new connection**. Fill in the required connection information on the panel, and enter the AdventureWorksLT for the database, where we generated the merge query in the dataflow. Then select **Next**.



4. Select the table you generated in the dataflow step earlier, and then select **Next**.

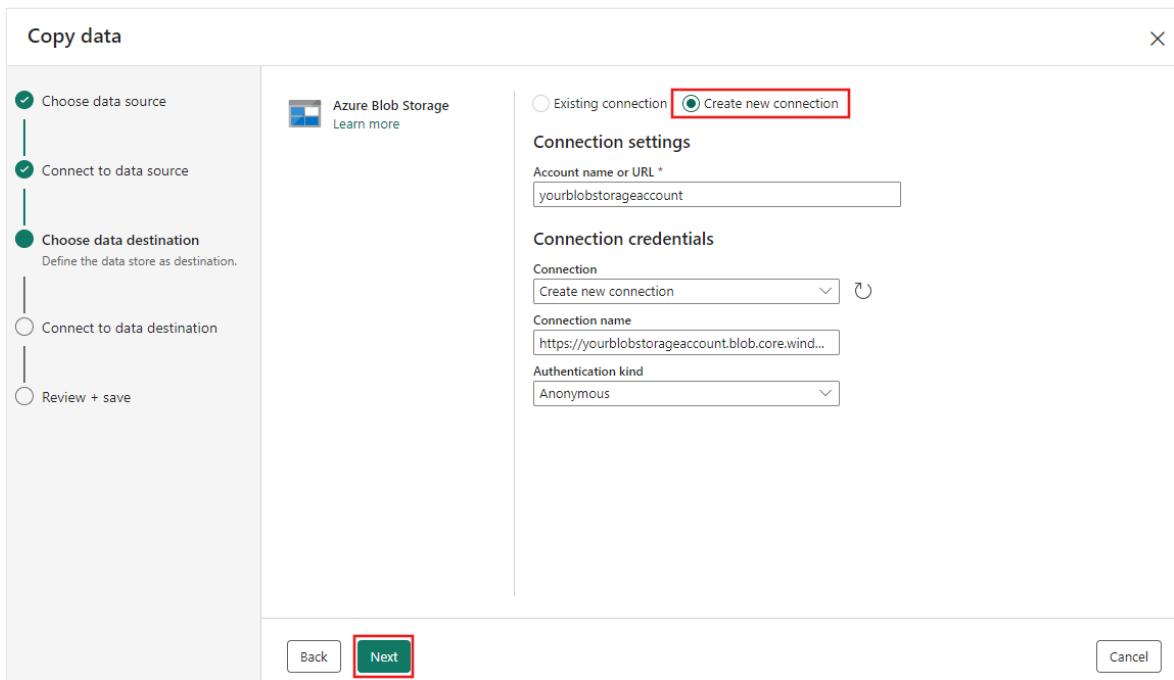
The screenshot shows the 'Copy data' wizard in progress. On the left, a vertical navigation pane lists steps: 'Choose data source' (selected), 'Connect to data source', 'Choose data destination', 'Connect to data destination', and 'Review + save'. The main area is titled 'Select a table' and shows 'Existing tables' selected. A preview window titled 'Preview data: dbo.CustomerSalesOrders' displays a table with columns: CustomerID, NameStyle, Title, FirstName, MiddleName, LastName, Suffix, CompanyName, SalesPerson, and several columns starting with 'ab'. Below the preview are buttons for 'Back', 'Next' (highlighted with a red box), and 'Cancel'.

5. For your destination, choose **Azure Blob Storage** and then select **Next**.

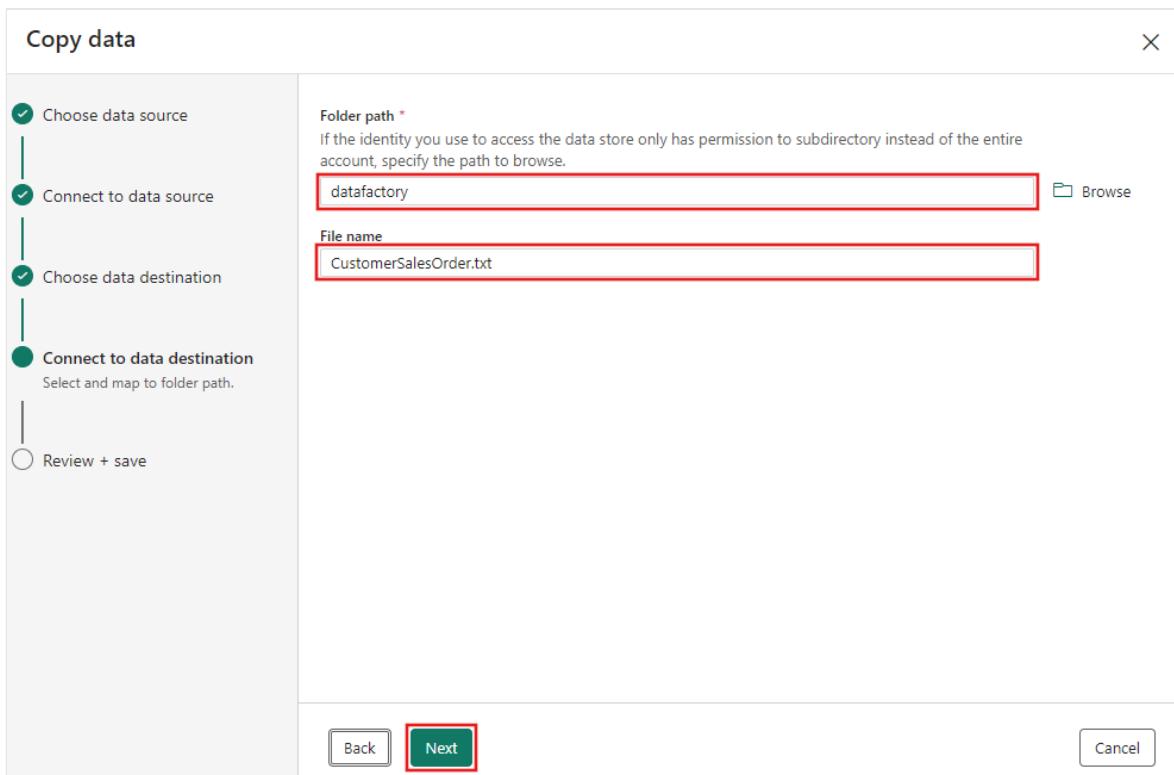
The screenshot shows the 'Copy data' wizard in progress. The left navigation pane shows 'Choose data source' (selected), 'Connect to data source' (selected), 'Choose data destination' (selected), 'Connect to data destination', and 'Review + save'. The main area is titled 'Data destinations' and shows tabs for 'All categories', 'Workspace', 'Azure' (selected), 'Database', 'Generic protocol', and 'Services and apps'. Under the Azure tab, 'Azure Blob Storage' is highlighted with a red box. Other options include 'Azure Cosmos DB for NoSQL', 'Azure Data Explorer (Kusto)', 'Azure Data Lake Storage Gen1', 'Azure Data Lake Storage Gen2', 'Azure Database for PostgreSQL', 'Azure SQL Database', 'Azure SQL Database Managed Instance', and 'Azure Table Storage'. Below the destination selection are buttons for 'Back', 'Next' (highlighted with a red box), and 'Cancel'.

6. Create a connection to your destination by selecting **Create new connection**.

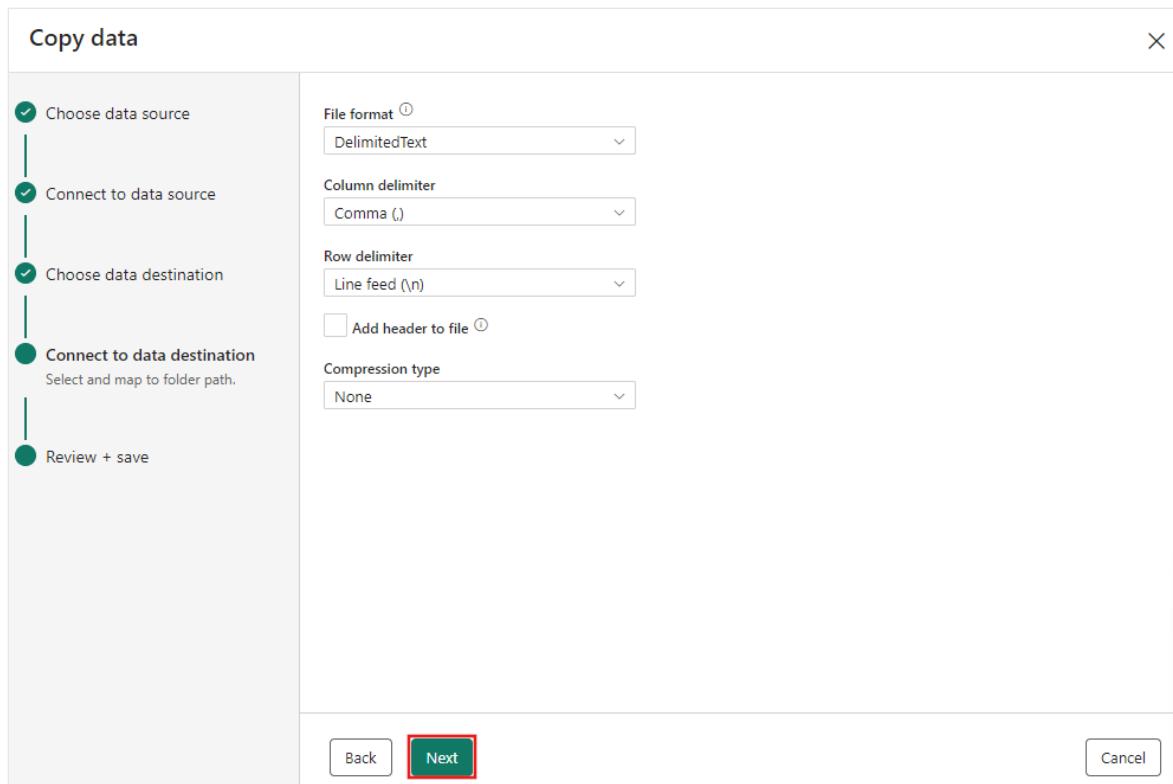
Provide the details for your connection, then select **Next**.



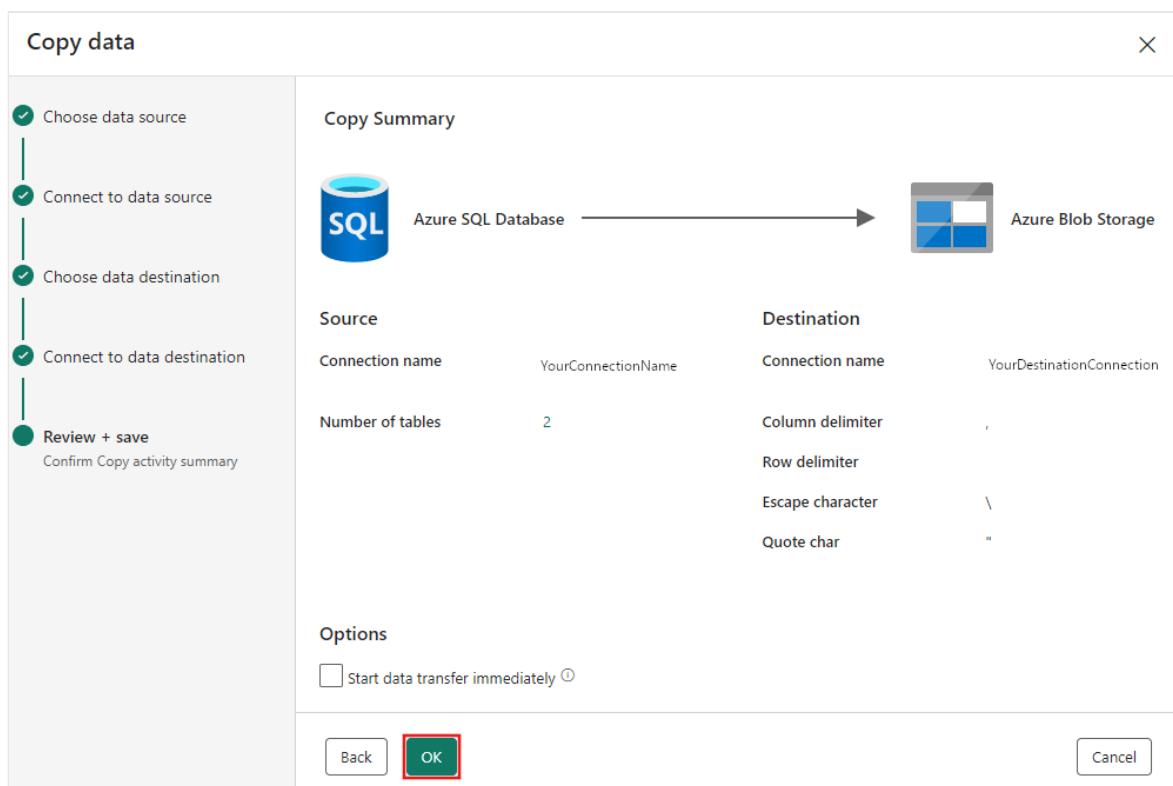
7. Select your **Folder path** and provide a **File name**, then select **Next**.



8. Select **Next** again to accept the default file format, column delimiter, row delimiter and compression type, optionally including a header.



9. Finalize your settings. Then, review and select **OK** to finish the process.

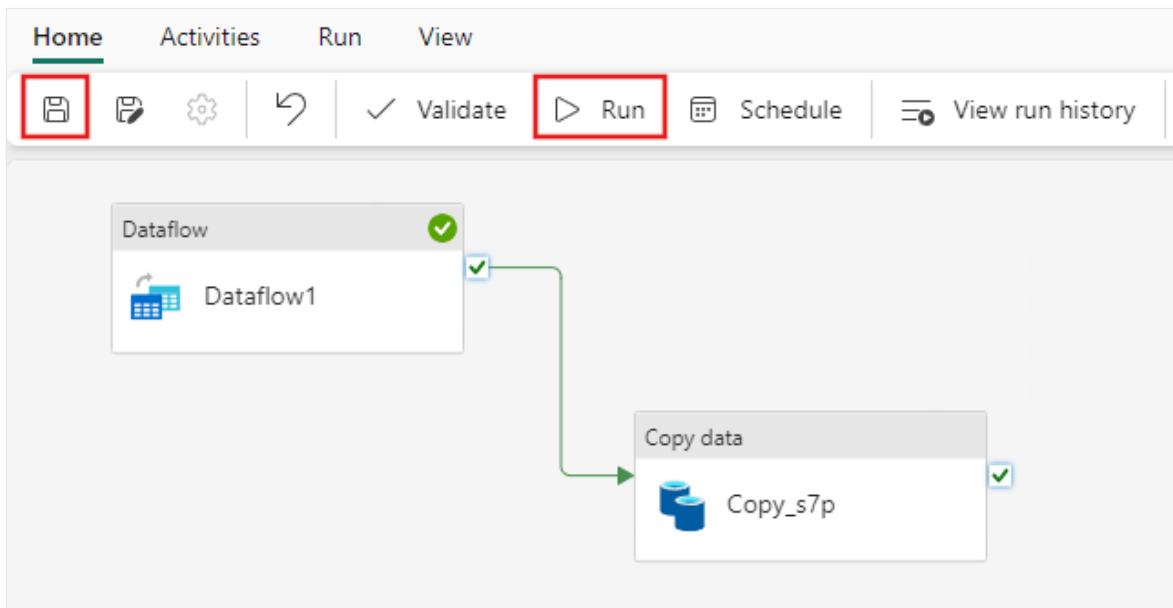


Step 5: Design your data pipeline and save to run and load data

1. To run the **Copy** activity after the **Dataflow** activity, drag from **Succeeded** on the **Dataflow** activity to the **Copy** activity. The **Copy** activity will only run after the **Dataflow** activity has succeeded.



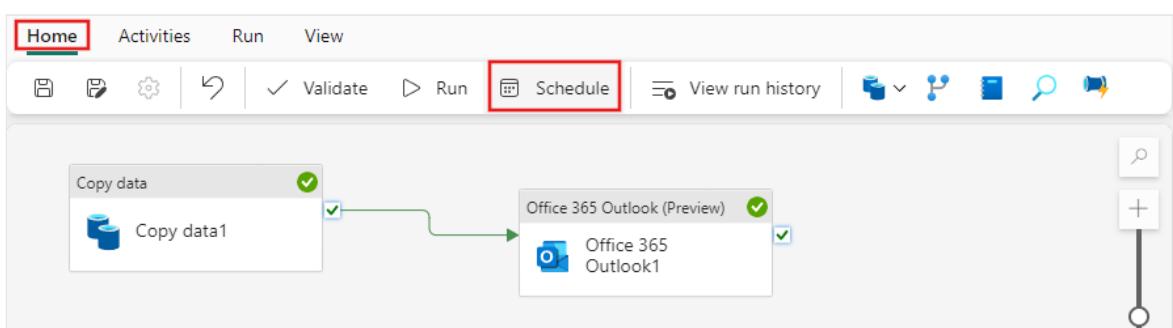
2. Select **Save** to save your data pipeline. Then select **Run** to run your data pipeline and load your data.



Schedule pipeline execution

Once you finish developing and testing your pipeline, you can schedule it to execute automatically.

1. On the **Home** tab of the pipeline editor window, select **Schedule**.



2. Configure the schedule as required. The example here schedules the pipeline to execute daily at 8:00 PM until the end of the year.

The screenshot shows the Microsoft Fabric Data Pipeline interface for a pipeline named "First_Pipeline". The "Schedule" tab is selected. Key configuration options visible include:

- Schedule**: A section with a "Scheduled run" toggle (set to "On") and a "Repeat" dropdown set to "Daily".
- Time**: A section showing the scheduled time as "08:00 PM" with a clock icon and a trash bin icon.
- Add a time**: A button to add additional scheduled times.
- Start**: A date and time input set to "04/21/2023 05:38 PM".
- End**: A date and time input set to "12/21/2023 03:38 PM".
- Time zone**: A dropdown menu showing "(UTC+08:00) Beijing, Chongqing, Hong Kong".

At the bottom are "Apply" and "Discard" buttons.

Next steps

This sample shows you how to create and configure a dataflow gen2 to create a merge query and store it in an Azure SQL database, then copy data from the database into a text file in Azure Blob Storage. You learned how to:

- ✓ Create a dataflow.
- ✓ Transform data with the dataflow.
- ✓ Create a data pipeline using the dataflow.
- ✓ Order the execution of steps in the pipeline.
- ✓ Copy data with the Copy Assistant.
- ✓ Run and schedule your data pipeline.

Next, advance to learn more about monitoring your pipeline runs.

[How to monitor pipeline runs in Microsoft Fabric](#)

Microsoft Fabric decision guide: copy activity, dataflow, or Spark

Article • 05/23/2023

Use this reference guide and the example scenarios to help you in deciding whether you need a copy activity, a dataflow, or Spark for your workloads using Microsoft Fabric.

ⓘ 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.

Copy activity, dataflow, and Spark properties

	Pipeline copy activity	Dataflow Gen 2	Spark
Use case	Data lake and data warehouse migration, data ingestion, lightweight transformation	Data ingestion, data transformation, data wrangling, data profiling	Data ingestion, data transformation, data processing, data profiling
Primary developer persona	Data engineer, data integrator	Data engineer, data integrator, business analyst	Data engineer, data scientist, data developer
Primary developer skill set	ETL, SQL, JSON	ETL, M, SQL	Spark (Scala, Python, Spark SQL, R)
Code written	No code, low code	No code, low code	Code
Data volume	Low to high	Low to high	Low to high
Development interface	Wizard, canvas	Power query	Notebook, Spark job definition

	Pipeline copy activity	Dataflow Gen 2	Spark
Sources	30+ connectors	150+ connectors	Hundreds of Spark libraries
Destinations	18+ connectors	Lakehouse, Azure SQL database, Azure Data explorer, Azure Synapse analytics	Hundreds of Spark libraries
Transformation complexity	Low: lightweight - type conversion, column mapping, merge/split files, flatten hierarchy	Low to high: 300+ transformation functions	Low to high: support for native Spark and open-source libraries

Review the following three scenarios for help with choosing how to work with your data in Fabric.

Scenario1

Leo, a data engineer, needs to ingest a large volume of data from external systems, both on-premises and cloud. These external systems include databases, file systems, and APIs. Leo doesn't want to write and maintain code for each connector or data movement operation. He wants to follow the medallion layers best practices, with bronze, silver, and gold. Leo doesn't have any experience with Spark, so he prefers the drag and drop UI as much as possible, with minimal coding. And he also wants to process the data on a schedule.

The first step is to get the raw data into the bronze layer lakehouse from Azure data resources and various third party sources (like Snowflake Web, REST, AWS S3, GCS, etc.). He wants a consolidated lakehouse, so that all the data from various LOB, on-premises, and cloud sources reside in a single place. Leo reviews the options and selects **pipeline copy activity** as the appropriate choice for his raw binary copy. This pattern applies to both historical and incremental data refresh. With copy activity, Leo can load Gold data to a data warehouse with no code if the need arises and pipelines provide high scale data ingestion that can move petabyte-scale data. Copy activity is the best low-code and no-code choice to move petabytes of data to lakehouses and warehouses from varieties of sources, either ad-hoc or via a schedule.

Scenario2

Mary is a data engineer with a deep knowledge of the multiple LOB analytic reporting requirements. An upstream team has successfully implemented a solution to migrate multiple LOB's historical and incremental data into a common lakehouse. Mary has been tasked with cleaning the data, applying business logics, and loading it into multiple destinations (such as Azure SQL DB, ADX, and a lakehouse) in preparation for their respective reporting teams.

Mary is an experienced Power Query user, and the data volume is in the low to medium range to achieve desired performance. Dataflows provide no-code or low-code interfaces for ingesting data from hundreds of data sources. With dataflows, you can transform data using 300+ data transformation options, and write the results into multiple destinations with an easy to use, highly visual user interface. Mary reviews the options and decides that it makes sense to use **Dataflow Gen 2** as her preferred transformation option.

Scenario3

Adam is a data engineer working for a large retail company that uses a lakehouse to store and analyze its customer data. As part of his job, Adam is responsible for building and maintaining the data pipelines that extract, transform, and load data into the lakehouse. One of the company's business requirements is to perform customer review analytics to gain insights into their customers' experiences and improve their services.

Adam decides the best option is to use **Spark** to build the extract and transformation logic. Spark provides a distributed computing platform that can process large amounts of data in parallel. He writes a Spark application using Python or Scala, which reads structured, semi-structured, and unstructured data from OneLake for customer reviews and feedback. The application cleanses, transforms, and writes data to Delta tables in the lakehouse. The data is then ready to be used for downstream analytics.

Next steps

- [How to copy data using copy activity](#)
- [Quickstart: Create your first dataflow to get and transform data](#)
- [How to create an Apache Spark job definition in Fabric](#)

Data Factory end-to-end scenario: introduction and architecture

Article • 05/23/2023

This tutorial helps you accelerate the evaluation process for Data Factory in Microsoft Fabric by providing a step-by-step guidance for a full data integration scenario within one hour. By the end of this tutorial, you understand the value and key capabilities of Data Factory and know how to complete a common end-to-end data integration scenario.

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Overview: Why Data Factory in Microsoft Fabric?

This section helps you understand the role of Fabric generally, and the role Data Factory plays within it.

Understand the value of Microsoft Fabric

Microsoft Fabric provides a one-stop shop for all the analytical needs for every enterprise. It covers a complete spectrum of services including data movement, data lake, data engineering, data integration and data science, real time analytics, and business intelligence. With Fabric, there's no need to stitch together different services from multiple vendors. Instead, your users enjoy an end-to-end, highly integrated, single, and comprehensive product that is easy to understand, onboard, create, and operate.

Understand the value of Data Factory in Microsoft Fabric

Data Factory in Fabric combines the ease-of-use of [Power Query](#) with the scale and power of [Azure Data Factory](#). It brings the best of both products together into a unified

experience. The goal is to make sure Data Integration in Factory works well for both citizen and professional data developers. It provides low-code, AI-enabled data preparation and transformation experiences, petabyte-scale transformation, hundreds of connectors with hybrid, multicloud connectivity. Purview provides governance, and the service features enterprise scale Data/Op commitments, CI/CD, application lifecycle management, and monitoring.

Introduction - Understand three key components of Data Factory

- Data ingestion: The Copy activity in pipelines lets you move petabyte-scale data from hundreds of data sources into your data Lakehouse for further processing.
- Data transformation and preparation: Dataflows gen2 provide a low-code interface for transforming your data using 300+ data transformations, with the ability to load the transformed results into multiple destinations such as Azure SQL databases, Lakehouse, and more.
- End-to-end integration flow automation: Pipelines provide orchestration of activities that include [Copy, Dataflow, and Notebook activities, and more](#). This lets you manage activities all in one place. Activities in a pipeline can be chained together to operate sequentially, or they can operate independently in parallel.

In this end-to-end data integration use case, you learn:

- How to ingest data using a Copy Activity in a pipeline
- How to transform the data using a dataflow either with a no-code experience, or by writing your own code to process the data with a Script or Notebook activity
- How to automate the entire end-to-end data integration flow using a pipeline with triggers and flexible control flow activities.

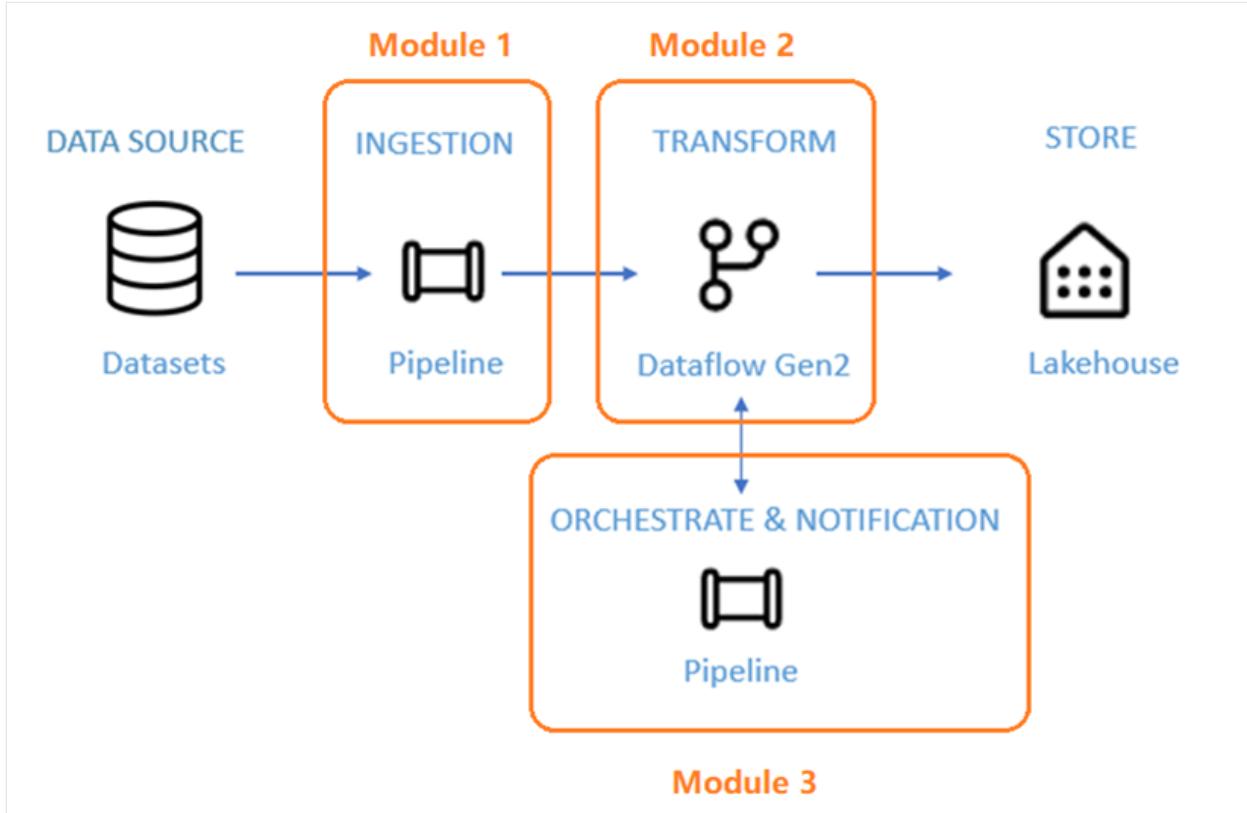
Architecture

In the next 50 minutes, you're tasked with completing an end-to-end data integration scenario. This includes ingesting raw data from a source store into the Bronze table of a Lakehouse, processing all the data, moving it to the Gold table of the data Lakehouse, sending an email to notify you once all the jobs are complete, and finally, setting up the entire flow to run on a scheduled basis.

The scenario is divided into three modules:

- [Module 1: Create a pipeline with Data Factory](#) to ingest raw data from a Blob storage to a Bronze table in a data Lakehouse.

- [Module 2: Transform data with a dataflow in Data Factory](#) to process the raw data from your Bronze table and move it to a Gold table in the data Lakehouse.
- [Module 3: Complete your first data integration journey](#) to send an email to notify you once all the jobs are complete, and finally, setup the entire flow to run on a scheduled basis.



You use the sample dataset **NYC-Taxi** as the data source for the tutorial. After you finish, you'll be able to gain insight into daily discounts on taxi fares for a specific period of time using Data Factory in Microsoft Fabric.

Next steps

In this introduction to our end-to-end tutorial for your first data integration using Data Factory in Microsoft Fabric, you learned:

- ✓ The value and role of Microsoft Fabric
- ✓ The value and role of Data Factory in Fabric
- ✓ Key components of Data Factory
- ✓ What you will learn in this tutorial

Continue to the next section now to create your data pipeline.

[Module 1: Create a pipeline with Data Factory](#)

Module 1: Create a pipeline with Data Factory

Article • 05/23/2023

This module takes 10 minutes, ingesting raw data from the source store into the Bronze table of a data Lakehouse using the Copy activity in a pipeline.

The high-level steps in module 1 are as follows:

1. Create a data pipeline.
2. Use a Copy Activity in the pipeline to load sample data into a data Lakehouse.

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Create a data pipeline

1. Sign into [Power BI](#) and open a workspace with Fabric Capacity enabled.

Create a data pipeline

1. Sign into [Power BI](#) using your administrator account credentials.
2. Choose your existing workspace with premium capacity enabled, or [create a new workspace](#) enabling premium capacity, leaving other options as their defaults.

Create a workspace

Name *

Available

Description

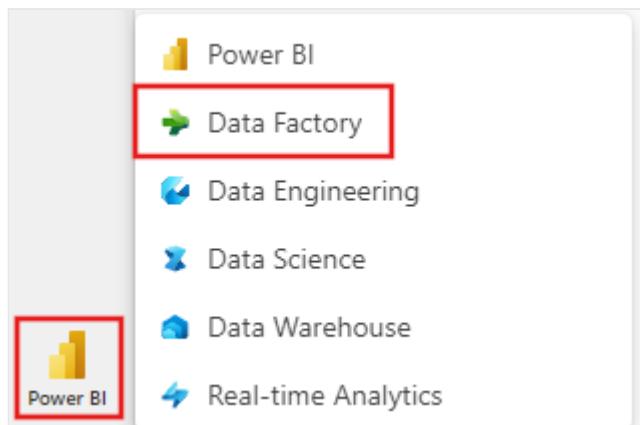
Domain (preview) ⓘ

Learn more about workspace settings ⓘ

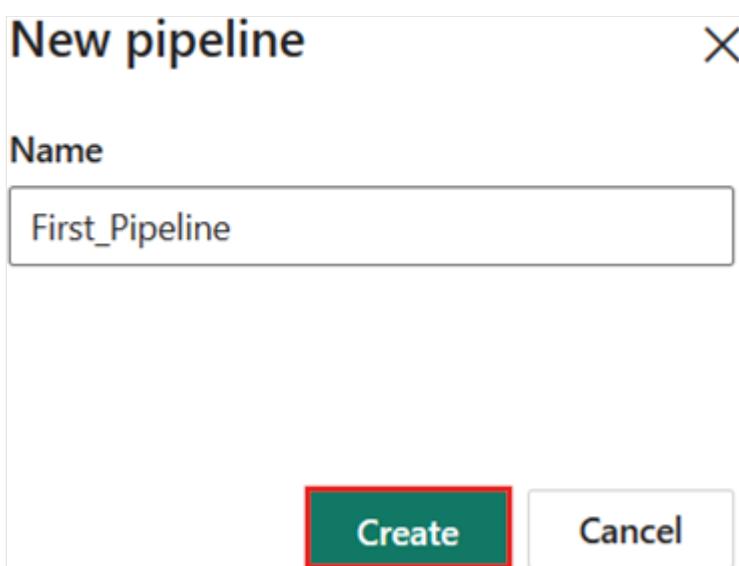
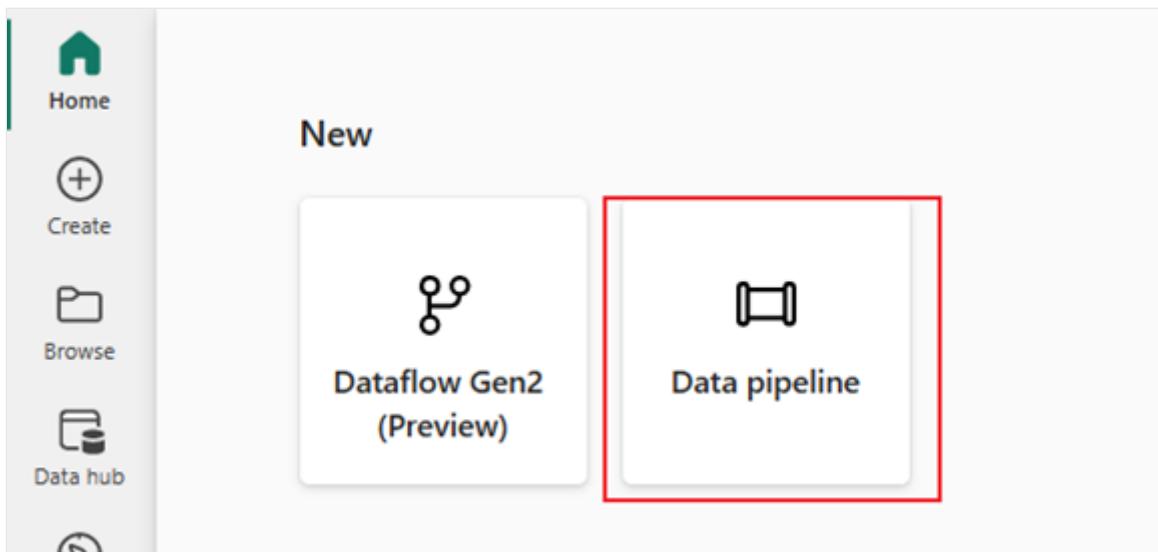
Workspace image



2. Select the default Power BI icon at the bottom left of the screen, and switch to the **Data Factory** experience.



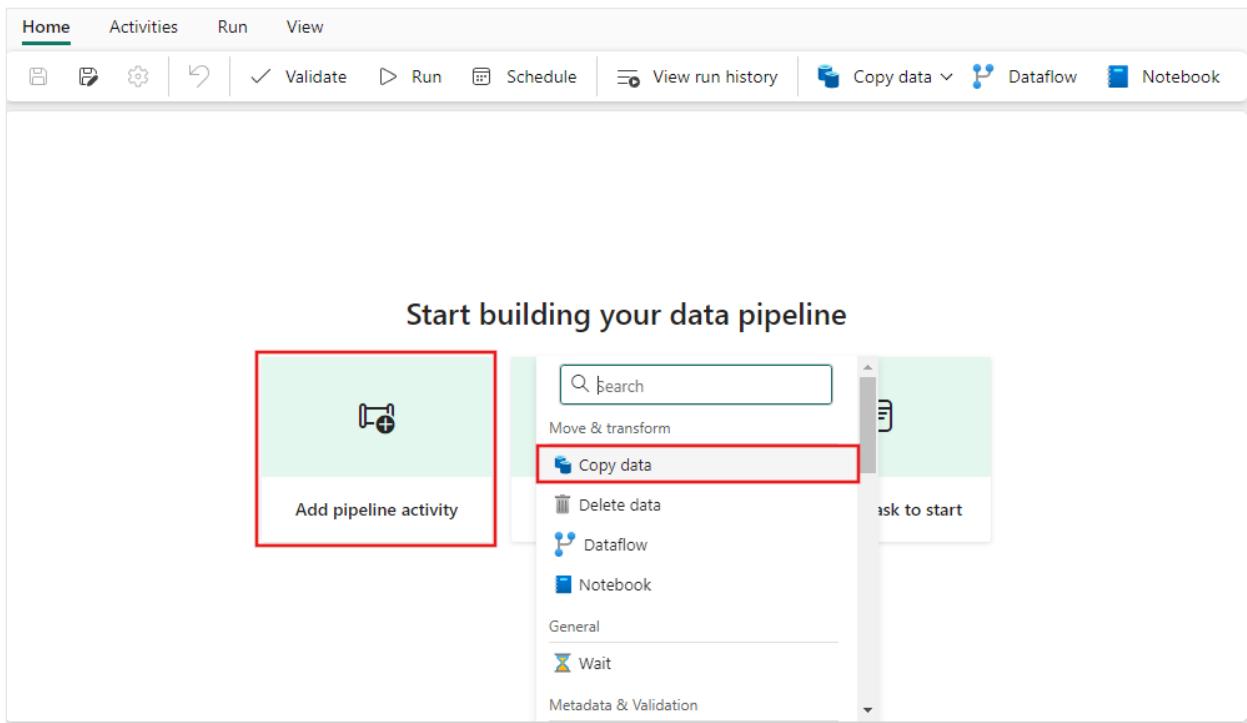
3. Select **Data pipeline** and provide a pipeline name. Then select **Create**.



Use a Copy activity in the pipeline to load sample data to a data Lakehouse

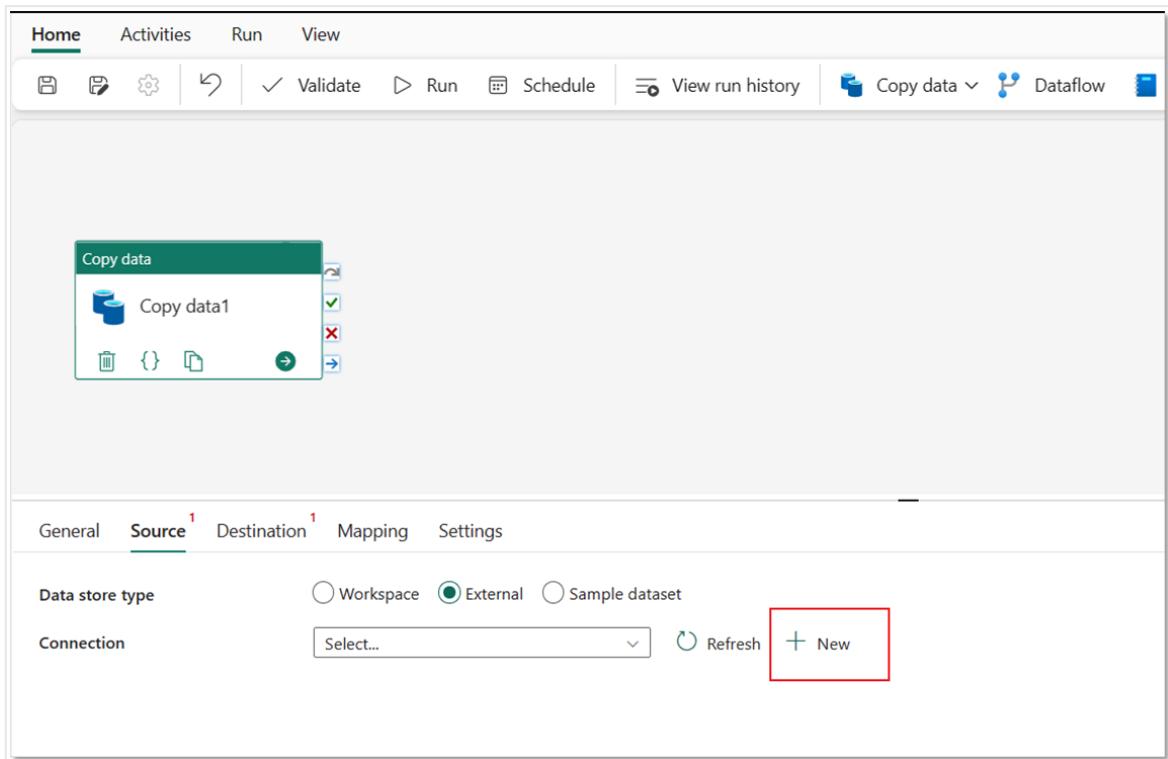
Step 1: Create a Copy activity in your new pipeline.

Select Add pipeline activity, and then choose **Copy data** from the displayed list of activities.

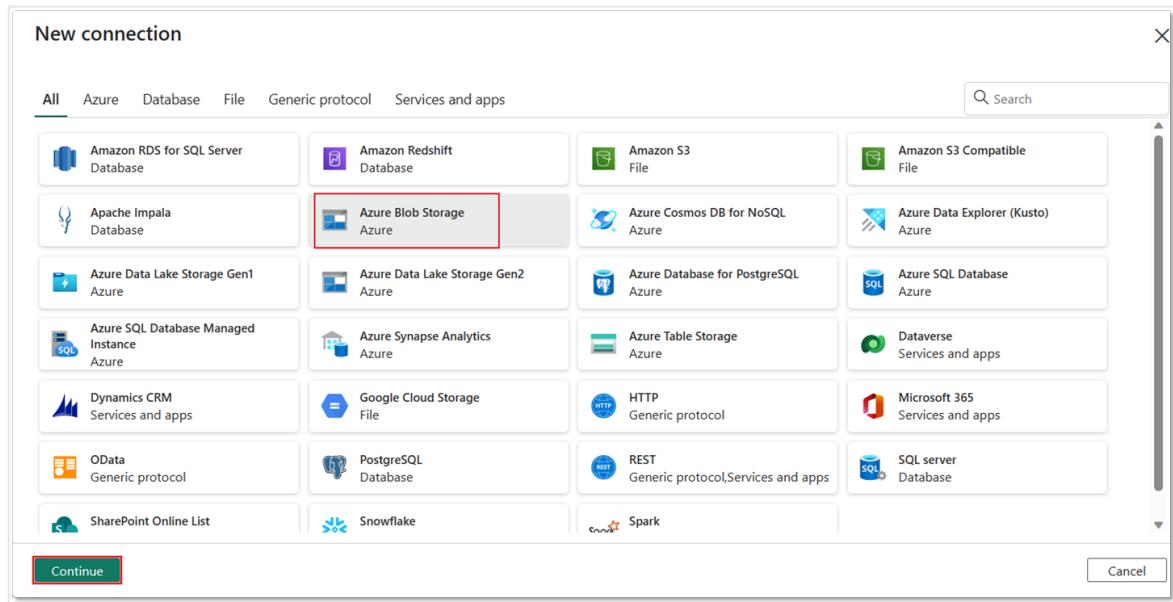


Step 2: Configure your source settings in your new Copy activity.

1. Select the **Source** tab in the properties area below the pipeline canvas, and then select **+ New** to create a new data source. (If you don't see the **Source** tab, you might need to first select the **Copy data** activity on the pipeline canvas area.)



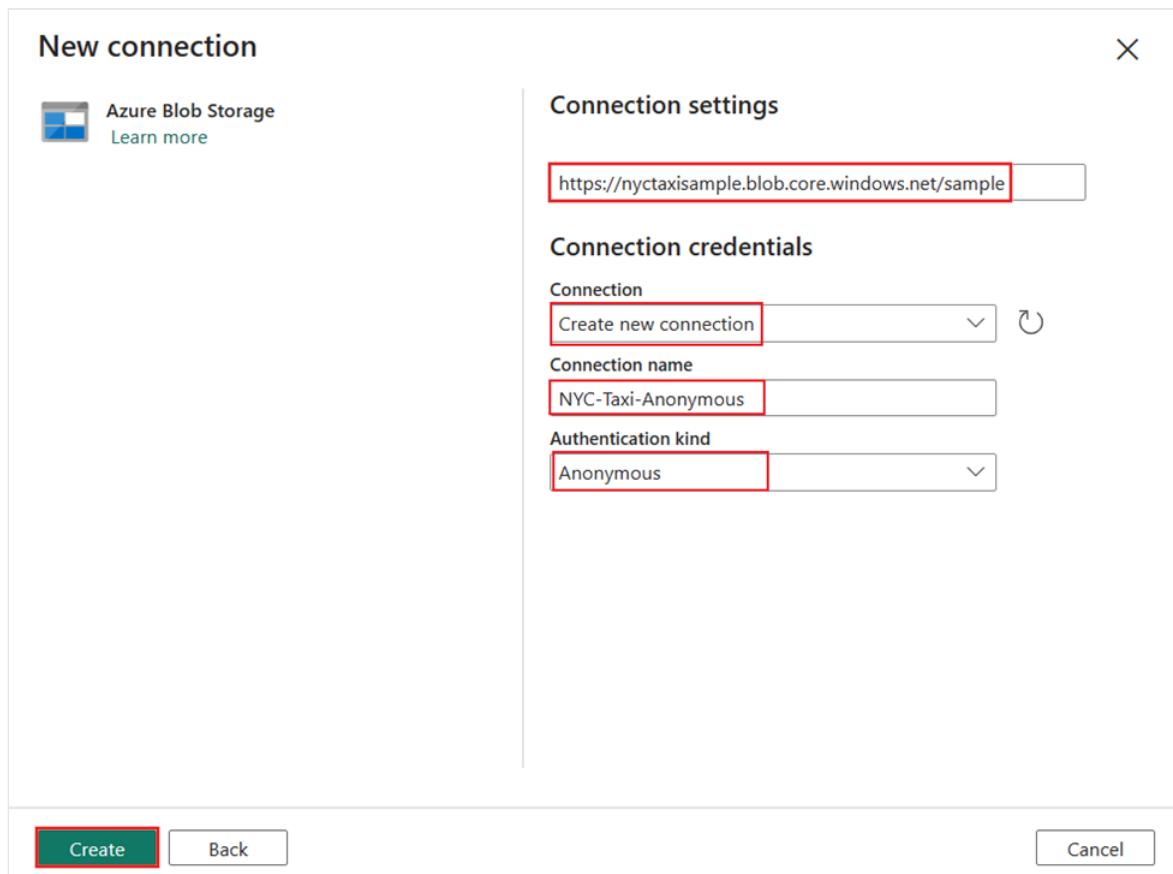
2. Select **Azure Blob Storage**, and then **Continue**.



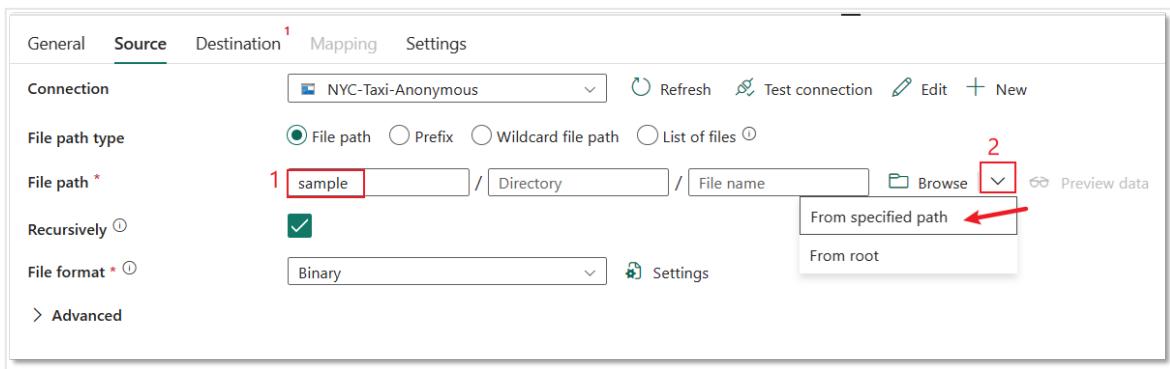
3. In the **New connection** dialog, provide the relevant details for the connection. For the tutorial, we use the following settings for the **NYC Taxi** sample data:

- **Account name or URL** -
<https://nyctaxisample.blob.core.windows.net/sample>
- **Connection** - unchanged, leaving **Create new connection** selected.
- **Connection name** - NYC-Taxi-Anonymous
- **Authentication kind** - Anonymous

Then select **Create**.



4. On the **Source** tab for the new data source you created, select the **File path** option for **File path type**, entering **sample** for the top level path, and then selecting the **Browse** dropdown to select **From specified path**.



5. On the **Browse** dialog presented, choose **NYC-Taxi-Green-2015-01.parquet** and select **OK**.

Browse

Select a file or folder.

Root folder > sample

 NYC-Taxi-Green-2015-01.parquet

Showing 1 item

OK

Cancel

6. Select Parquet for the File format dropdown, and then select Preview data.

General	Source	Destination	Mapping	Settings
Data store type <input type="radio"/> Workspace <input checked="" type="radio"/> External <input type="radio"/> Sample dataset				
Connection <input type="button" value="NYC-Taxi-Anonymous"/> Refresh <input type="button" value="Test connection"/> Edit <input type="button" value="New"/>				
File path type <input checked="" type="radio"/> File path <input type="radio"/> Prefix <input type="radio"/> Wildcard file path <input type="radio"/> List of files <input type="button" value="..."/>				
File path * <input type="text" value="sample"/> / <input type="text" value="Directory"/> / <input type="text" value="NYC-Taxi-Green-201..."/> <input type="button" value="Browse"/> <input type="button" value="Preview data"/> 2				
Recursively <input checked="" type="checkbox"/> 1				
File format * <input type="text" value="Parquet"/> <input type="button" value="Settings"/> 2				

Preview data

	vendorID	lpepPickupDatetime	lpepDropoffDatetime	passengerCount	tripDistance	puLocationId	doLocationId	pickupLong
1	2	2015-01-01T21:41:00	2015-01-01T21:47:44	1	0.94			-73.9173660
2	2	2015-01-01T21:43:03	2015-01-01T21:47:08	5	1.03			-73.8771591
3	2	2015-01-01T21:46:01	2015-01-01T21:48:08	1	0.8			-73.9368362
4	2	2015-01-01T21:36:57	2015-01-01T21:47:37	4	3.24			-73.8626785
5	2	2015-01-01T21:33:22	2015-01-01T21:47:47	1	3.17			-73.9197692
6	2	2015-01-01T21:40:47	2015-01-01T21:48:04	1	1.68			-73.8440933
7	2	2015-01-01T21:29:13	2015-01-01T21:30:59	1	0			-73.9307785
8	2	2015-01-01T21:22:21	2015-01-01T21:31:05	1	1.36			-73.9574890
9	2	2015-01-01T21:18:17	2015-01-01T21:31:26	1	3.79			-73.9165954
10	2	2015-01-01T21:19:25	2015-01-01T21:31:35	1	1.86			-73.9776840

Step 3: Configure the destination settings for your Copy activity.

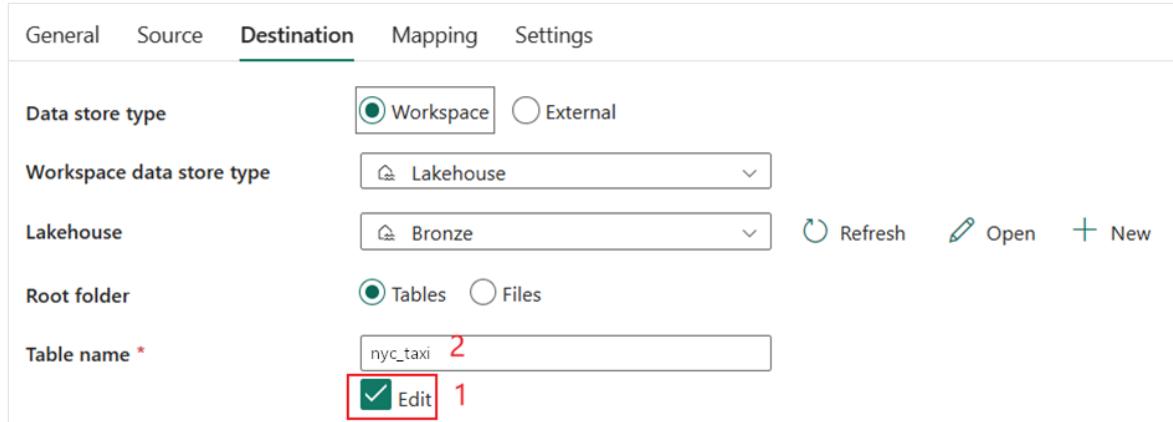
1. Select the **Destination** tab for your Copy activity, then select **+ New** to create a new Lakehouse destination, and give it a name. For this tutorial we name our destination Bronze. After providing the name, select **Create**.

New lakehouse X

Name *

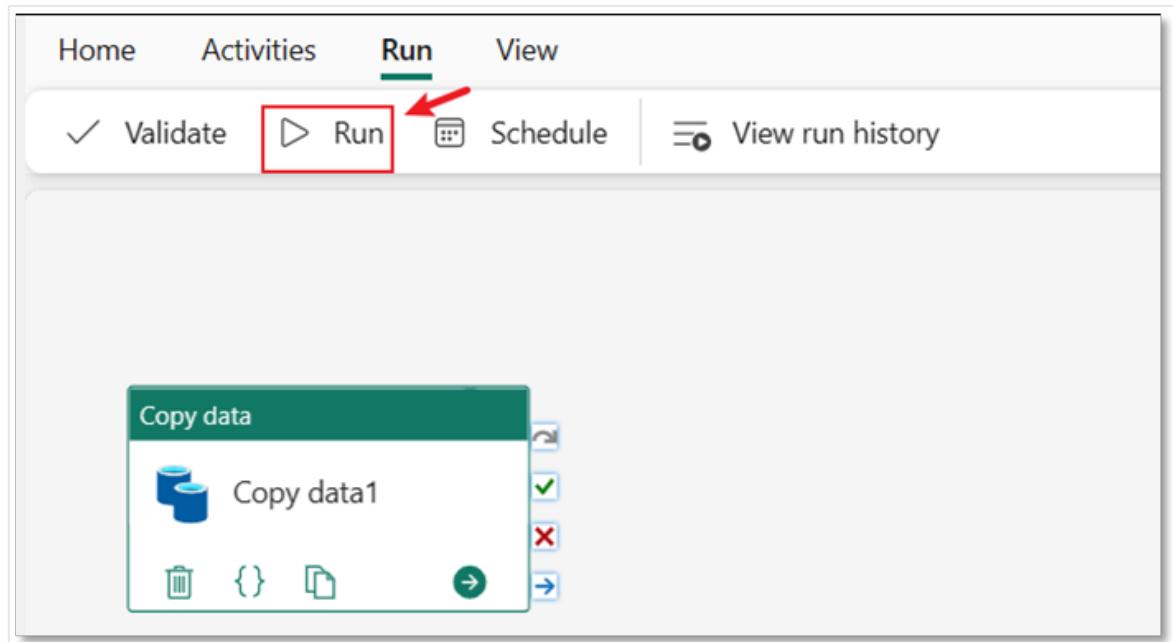
Create
Cancel

2. For the **Table name** property, select the **Edit** checkbox to create a new Lakehouse table where the data is loaded, and provide the name **nyc_taxi**.



Step 4: Run and view the results of your Copy activity.

1. Select the **Run** tab in the pipeline editor. Then select the **Run** button, and then **Save and run** at the prompt, to run the Copy activity.



Save and run?

X

You have unsaved changes. If you continue, the pipeline will be saved and run.

Save and run

Cancel

2. You can monitor the run and check the results on the **Output** tab below the pipeline canvas. Select the run details button (the "glasses" icon that appears when you hover over the running pipeline run) to view the run details.

The screenshot shows the Azure Data Factory Pipeline Output tab. At the top, there's a list of pipeline runs with one named 'Copy data1' selected. Below this, the 'Output' tab is active, showing a table of pipeline run details. The table has columns: Name, Type, Run start, Duration, and Status. One row is shown: 'Copy data1' (Type: Copy data, Run start: 4/26/2023, Duration: 00:00:59, Status: Succeeded). A 'Details' button is located at the bottom of this table row.

Name	Type	Run start	Duration	Status
Copy data1	Copy data	4/26/2023, 4:22:58 PM	00:00:59	Succeeded

3. The run details show 1,508,501 rows read and written.

Copy data details

Copy data1

Source	Destination
 Azure Blob Storage	 Lakehouse
Data read: ⓘ 1.185 GB	Data written: ⓘ 37.977 MB
Files read: ⓘ 1	Files written: ⓘ 1
Rows read: 1,508,501	Rows written: 1,508,501

Status  Succeeded

Start time 4/26/2023, 4:22:58 PM

Pipeline run activity ID 17506209-fd5b-4938-97d3-59ec40f4cf52

Throughput 23.698 MB/s

Total duration 00:00:56

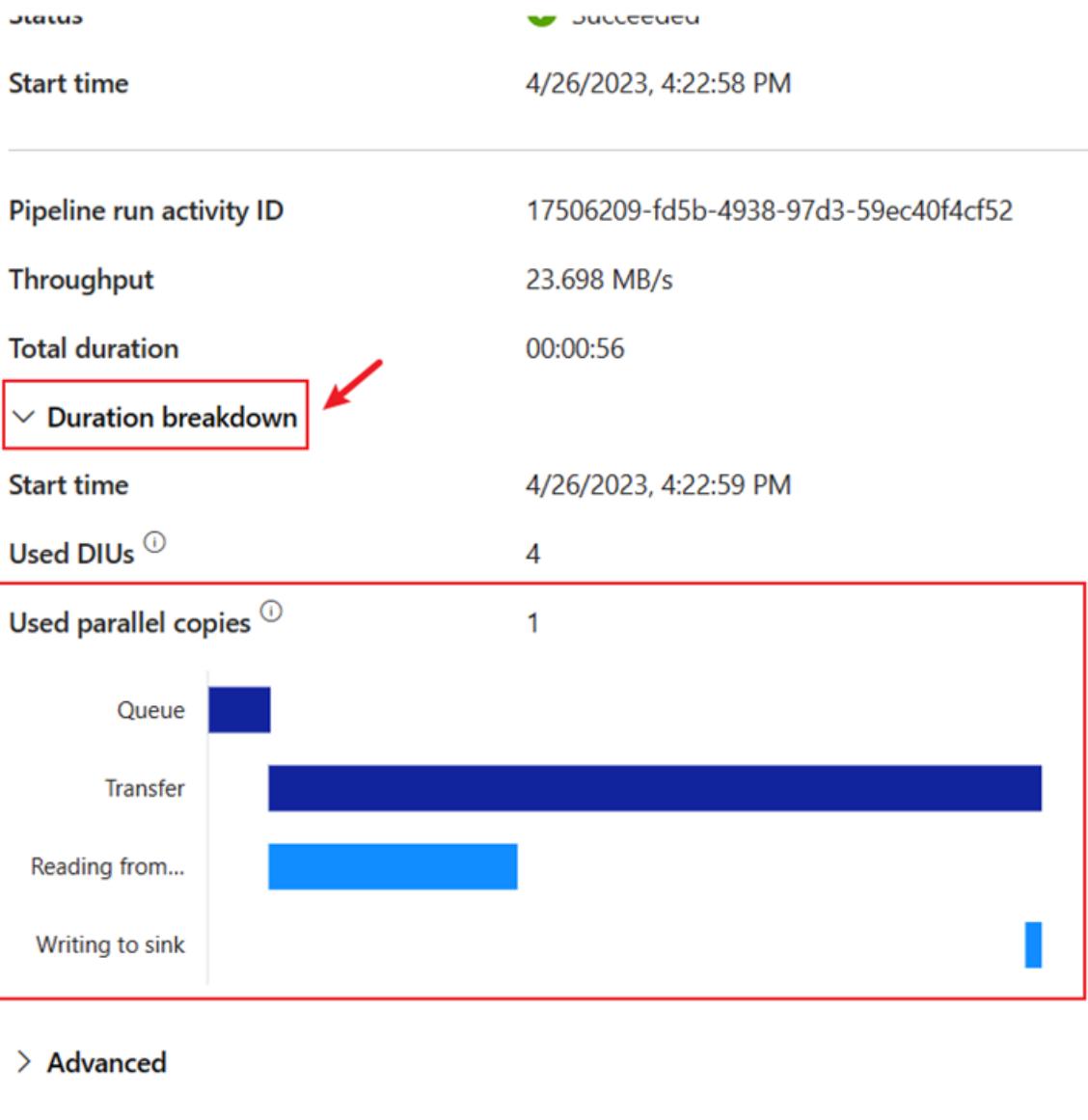
› Duration breakdown

› Advanced

Close

4. Expand the **Duration breakdown** section to see the duration of each stage of the Copy activity. After reviewing the copy details, select **Close**.

Copy data details



Next steps

In this first module to our end-to-end tutorial for your first data integration using Data Factory in Microsoft Fabric, you learned how to:

- ✓ Create a data pipeline.
- ✓ Add a Copy activity to your pipeline.
- ✓ Use sample data and create a data Lakehouse to store the data to a new table.
- ✓ Run the pipeline and view its details and duration breakdown.

Continue to the next section now to create your dataflow.

[Module 2: Transform data with a dataflow in Data Factory](#)

Module 2: Transform data with a dataflow in Data Factory

Article • 05/23/2023

This module takes about 25 minutes to create a dataflow, apply transformations, and move the raw data from the Bronze table into a Gold Lakehouse table.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

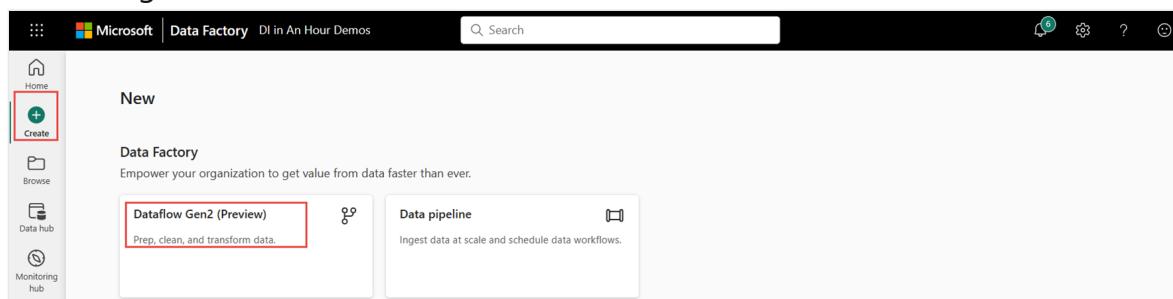
With the raw data loaded into your Bronze Lakehouse table from the last module, you can now prepare that data and enrich it by combining it with another table that contains discounts for each vendor and their trips during a particular day. This final Gold Lakehouse table is loaded and ready for consumption.

The high-level steps in the dataflow are as follows:

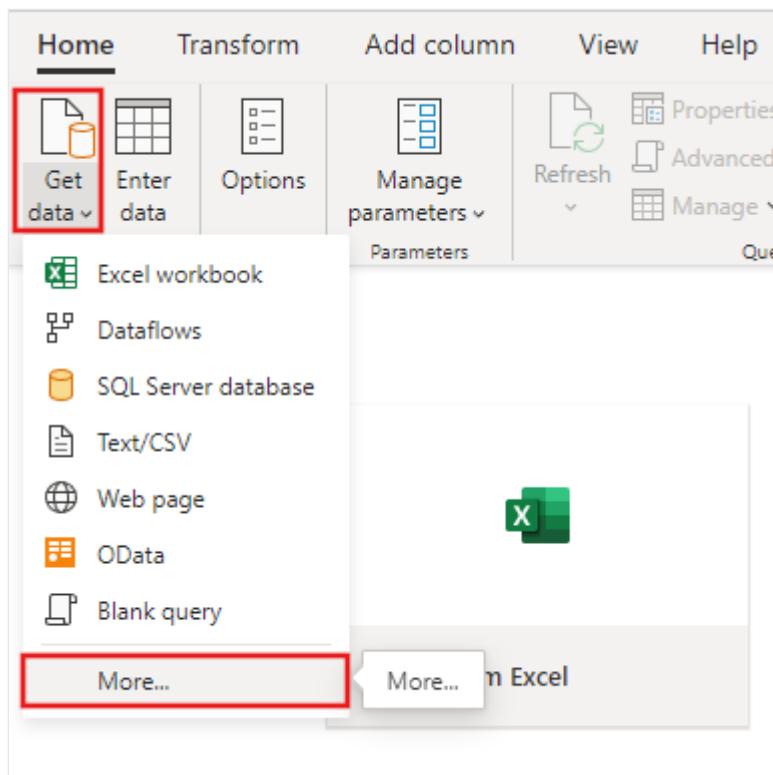
- Get raw data from the Lakehouse table created by the Copy activity in [Module 1: Create a pipeline with Data Factory](#).
- Transform the data imported from the Lakehouse table.
- Connect to a CSV file containing discounts data.
- Transform the discounts data.
- Combine trips and discounts data.
- Load the output query into the Gold Lakehouse table.

Get data from a Lakehouse table

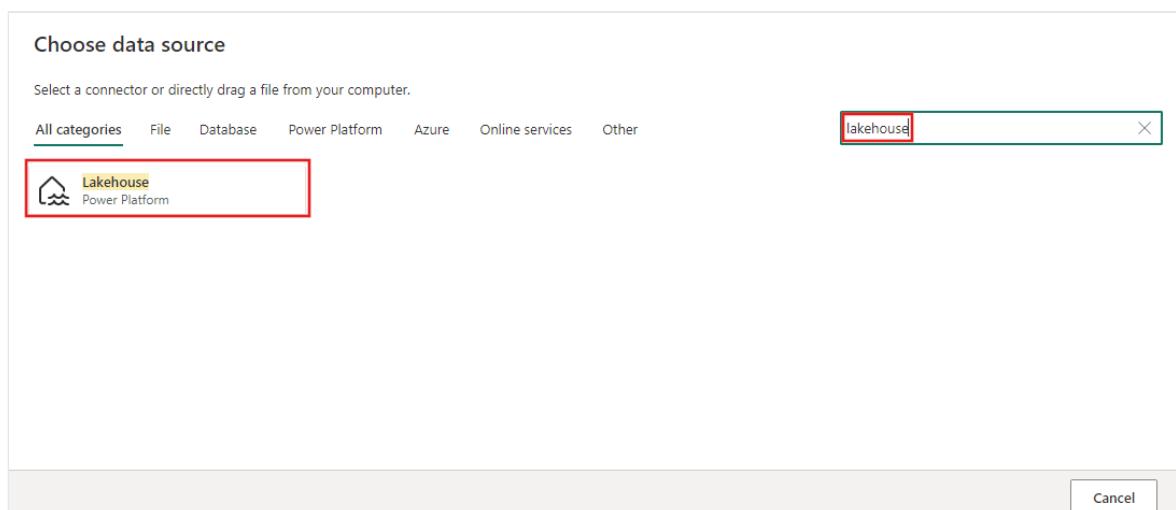
1. From the sidebar, select **Create**, and then **Dataflow Gen2 (Preview)** to create a new dataflow gen2.



2. From the new dataflow menu, select **Get data**, and then **More....**

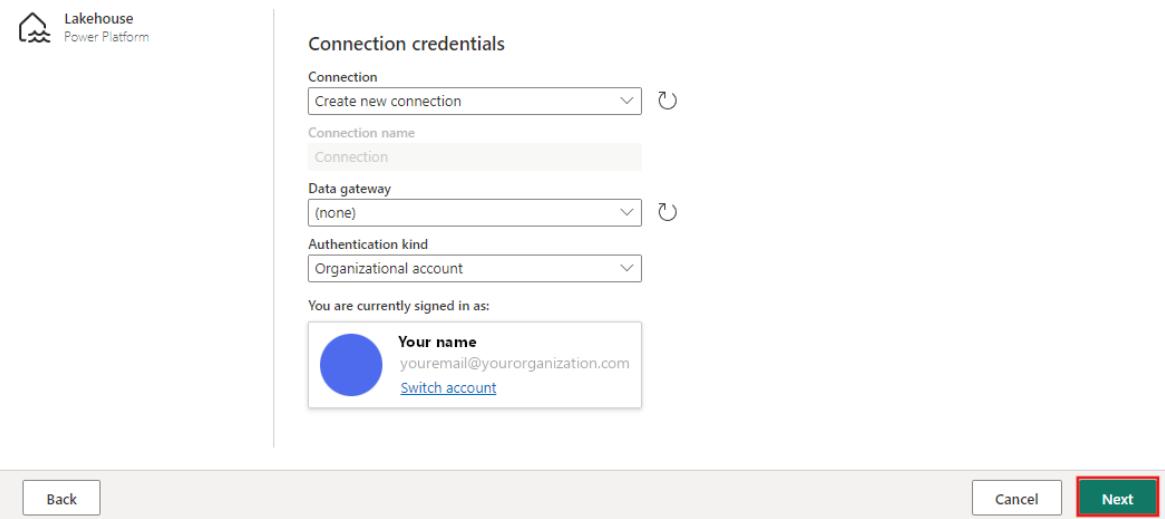


3. Search for and select the **Lakehouse** connector.



4. The **Connect to data source** dialog appears, and a new connection is automatically created for you based on the currently signed in user. Select **Next**.

Connect to data source

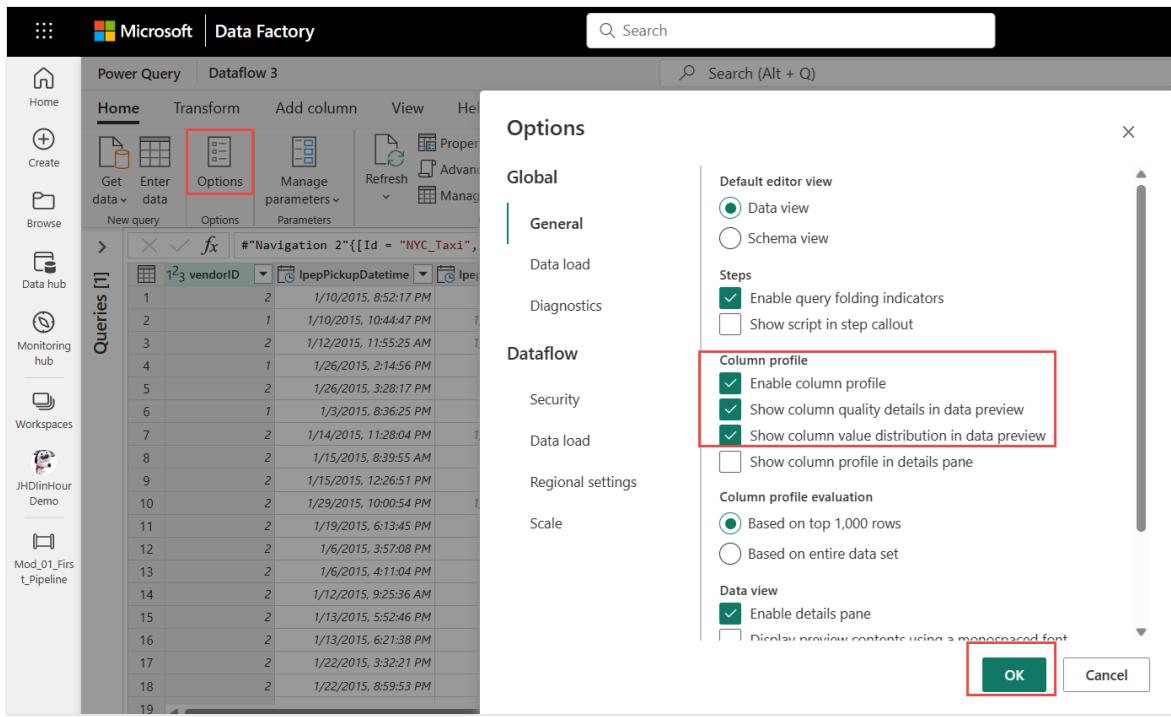


5. The **Choose data** dialog is displayed. Use the navigation pane to find the Lakehouse you created for the destination in the prior module, and select the nyc_taxi data table.

	vendorID	lpepPickupDatetime	lpepDropoffDatetime	passengerCount	tripDistance	puLocationId	doLocationId	pickUpTime
1	8/5/2016, 7:49:49 PM	8/5/2016, 7:55:12 PM		1	0.9	33	33	
1	8/5/2016, 7:15:30 PM	8/5/2016, 7:27:19 PM		1	1.9	244	243	
1	8/5/2016, 7:02:34 PM	8/5/2016, 7:05:43 PM		1	0.6	166	41	
1	8/12/2016, 3:59:17 PM	8/12/2016, 4:39:49 PM		1	3.6	97	79	
1	8/12/2016, 4:58:23 PM	8/12/2016, 5:09:43 PM		1	1.5	74	236	
1	8/12/2016, 4:20:26 PM	8/12/2016, 4:30:52 PM		1	2.5	29	22	
1	8/12/2016, 5:37:07 PM	8/12/2016, 5:43:04 PM		1	0.9	255	112	
1	8/12/2016, 5:33:34 PM	8/12/2016, 5:37:18 PM		1	0.6	63	260	
1	8/12/2016, 5:22:56 PM	8/12/2016, 5:36:27 PM		1	5.3	244	143	
1	8/15/2016, 5:04:14 PM	8/15/2016, 5:20:08 PM		1	3.9	129	138	
1	8/15/2016, 7:00:31 PM	8/15/2016, 7:51:47 PM		1	5.4	243	244	
1	8/5/2016, 6:59:21 PM	8/5/2016, 7:12:01 PM		1	2.2	33	49	
1	8/5/2016, 6:59:12 PM	8/5/2016, 7:06:51 PM		1	1.1	74	74	
1	8/5/2016, 6:55:51 PM	8/5/2016, 7:02:28 PM		1	0.9	41	42	
1	8/5/2016, 6:53:37 PM	8/5/2016, 6:56:13 PM		1	0.7	255	112	
1	8/5/2016, 6:52:43 PM	8/5/2016, 7:03:32 PM		1	3.5	243	166	

6. (*Optional*) Once your canvas is populated with the data, you can set **column profile** information, as this is useful for data profiling. You can apply the right transformation and target the right data values based on it.

To do this, select **Options** from the ribbon pane, then select the first three options under **Column profile**, and then select **OK**.

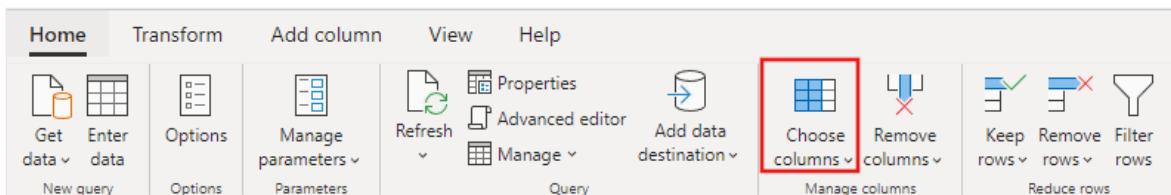


Transform the data imported from the Lakehouse

1. Select the data type icon in the column header of the second column, **IpepPickupDatetime**, to display a dropdown menu and select the data type from the menu to convert the column from the **Date/Time** to **Date** type.

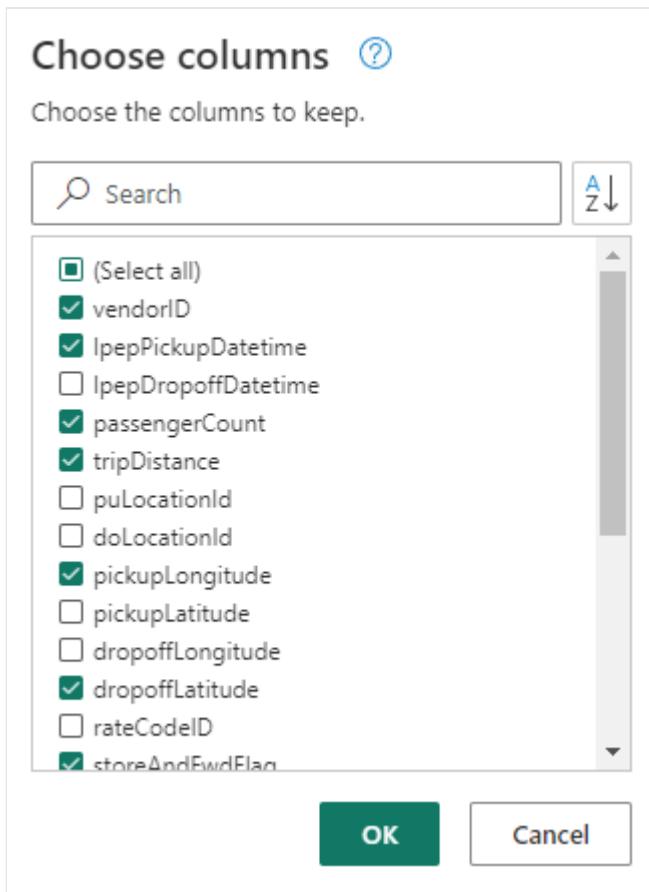
The screenshot shows the Microsoft Data Factory Power Query interface. The main area displays a table with columns: vendorID, IpepPickupDatetime, IpepDropoffDatetime, and passengerCount. The 'IpepPickupDatetime' column header has a dropdown arrow, and a red box highlights the 'Date' option in the resulting dropdown menu. The table data includes various pickup dates and times. The ribbon at the top has tabs for Home, Transform, Add column, View, and Help. The 'Home' tab is selected. The 'Options' dialog box is partially visible on the right.

2. (Optional) On the **Home** tab of the ribbon, select the **Choose columns** option from the **Manage columns** group.



3. (Optional) On the **Choose columns** dialog, deselect some columns listed here, then select OK.

- lpepDropoffDatetime
- puLocationId
- doLocationId
- pickupLongitude
- pickupLatitude
- dropoffLongitude
- dropoffLatitude
- rateCodeID



4. Select the **storeAndFwdFlag** column's filter and sort dropdown menu. (If you see a warning **List may be incomplete**, select **Load more** to see all the data.)

5. Select 'Y' to show only rows where a discount was applied, and then select **OK**.

A dropdown menu for filtering the 'storeAndFwdFlag' column. The menu includes:

- Sort ascending
- Sort descending
- Remove empty
- Text filters

The 'Text filters' section shows a search bar and a list of filter items:

- (Select all)
- N
- Y

Buttons at the bottom right are 'OK' (highlighted with a red border) and 'Cancel'.

6. Select the **Ipep_Pickup_Datetime** column sort and filter dropdown menu, then select **Date filters**, and choose the **Between...** filter provided for Date and Date/Time types.

A dropdown menu for filtering the 'IpepPickupDatetime' column. The menu includes:

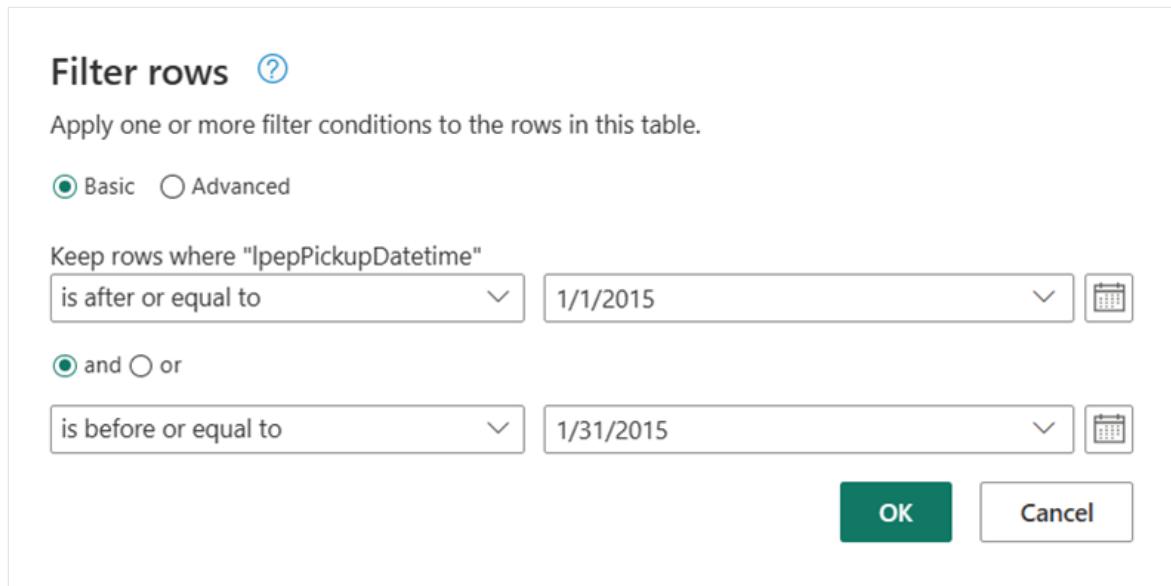
- Sort ascending
- Sort descending
- Remove empty
- Date filters

The 'Date filters' section shows a search bar and a list of filter items:

- (Select all)
- 8/5/2016
- 8/6/2016
- 8/12/2016
- 8/15/2016

A note says "List may be incomplete." with a "Load more" link. Buttons at the bottom right are 'OK' (highlighted with a red border) and 'Cancel'.

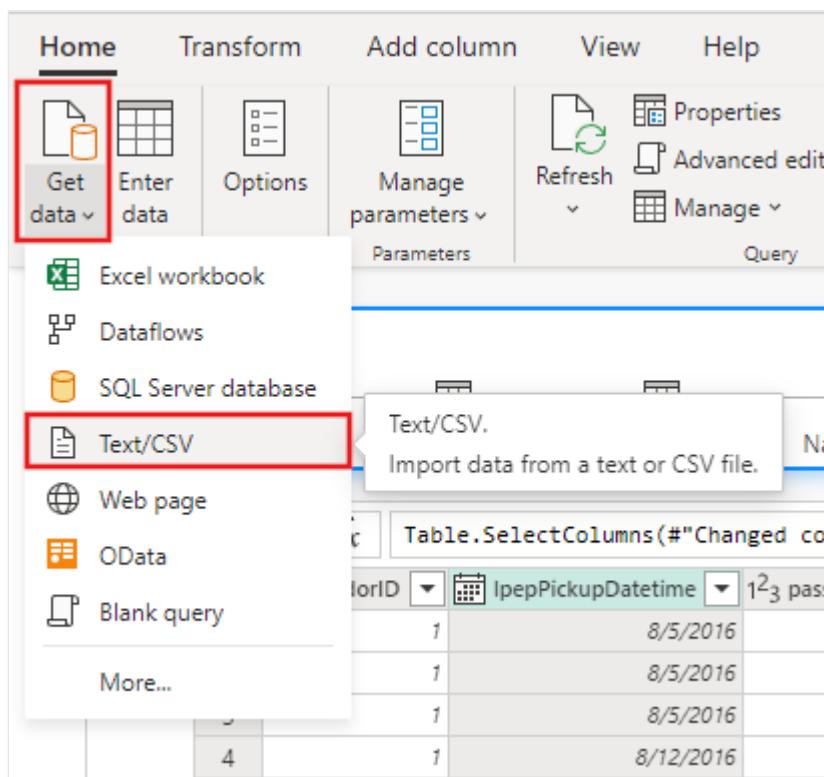
7. In the **Filter rows** dialog, select dates between January 1, 2015, and January 31, 2015, then select **OK**.



Connect to a CSV file containing discount data

Now, with the data from the trips in place, we want to load the data that contains the respective discounts for each day and VendorID, and prepare the data before combining it with the trips data.

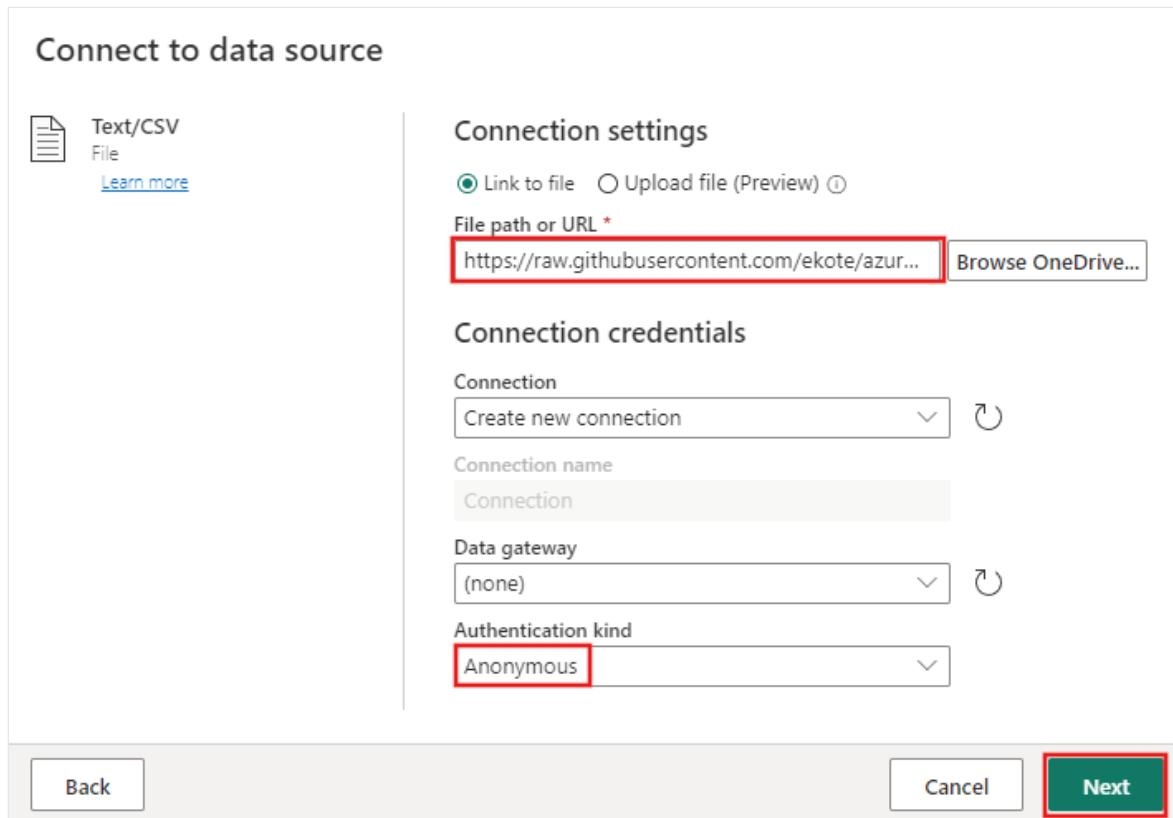
1. From the **Home** tab in the dataflow editor menu, select the **Get data** option, and then choose **Text/CSV**.



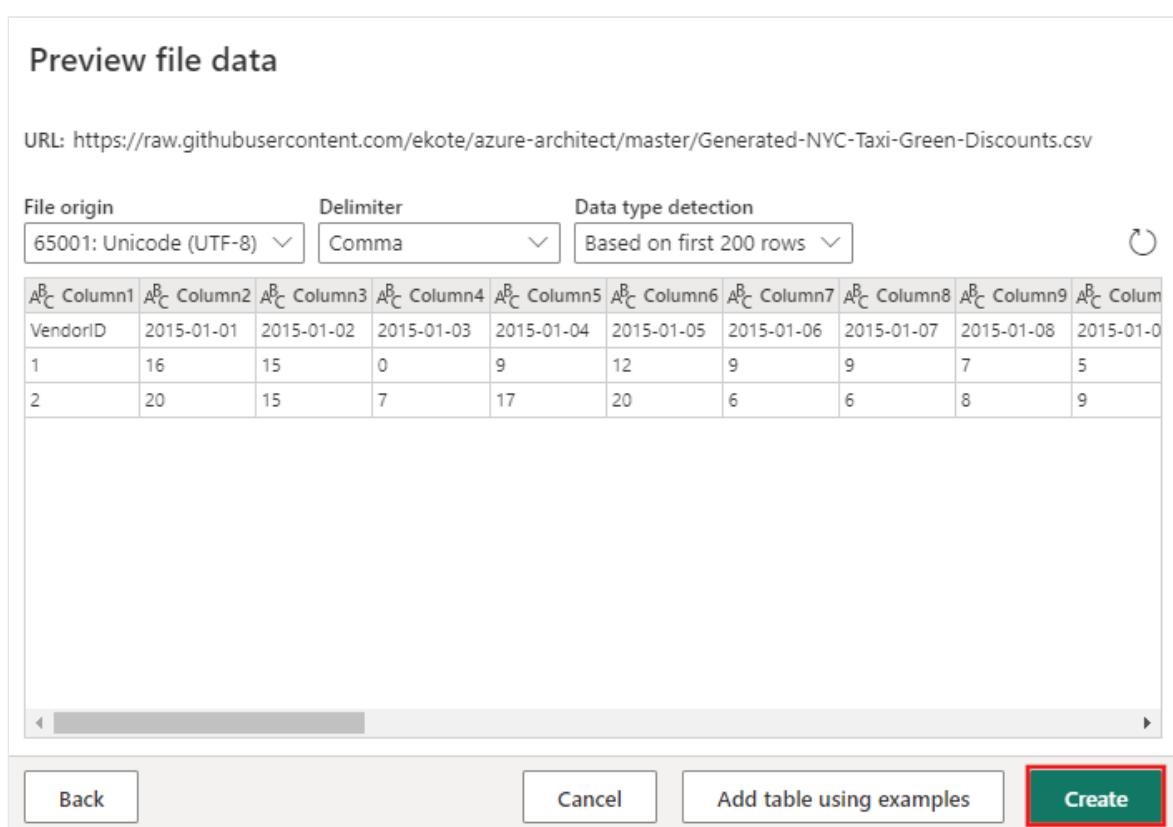
2. On the **Connect to data source** dialog, provide the following details:

- **File path or URL** - <https://raw.githubusercontent.com/ekote/azure-architect/master/Generated-NYC-Taxi-Green-Discounts.csv>
- **Authentication kind** - Anonymous

Then select **Next**.

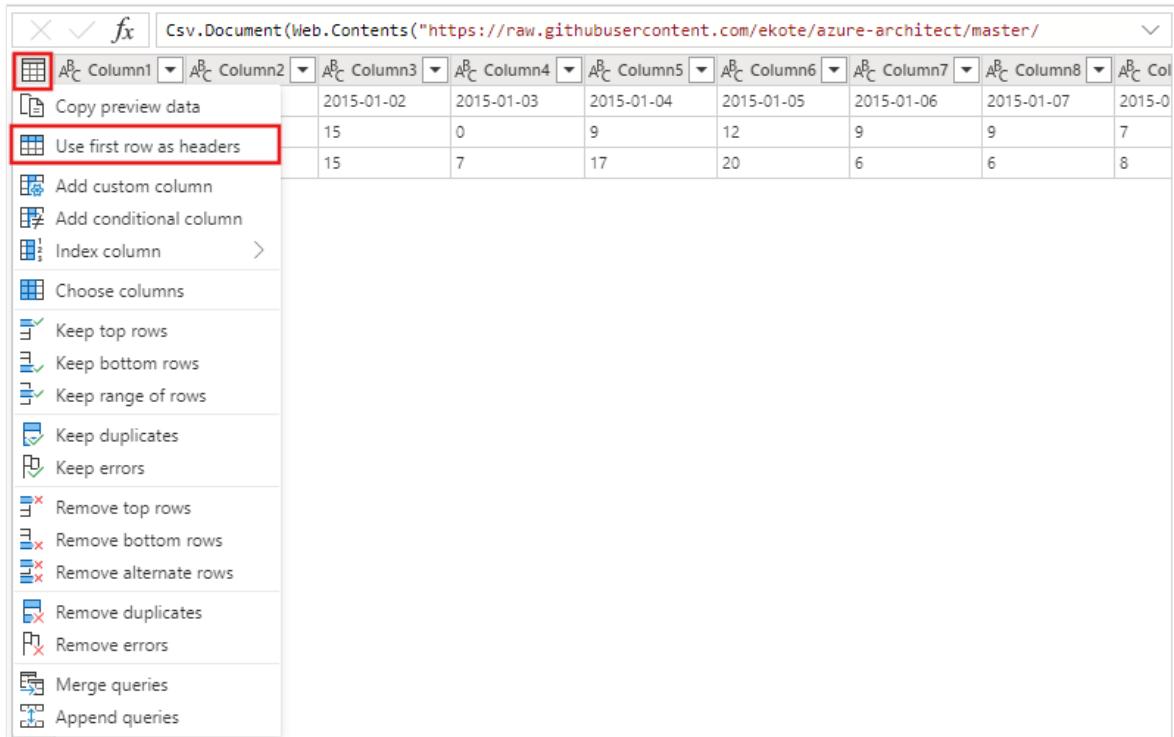


3. On the **Preview file data** dialog, select **Create**.



Transform the discount data

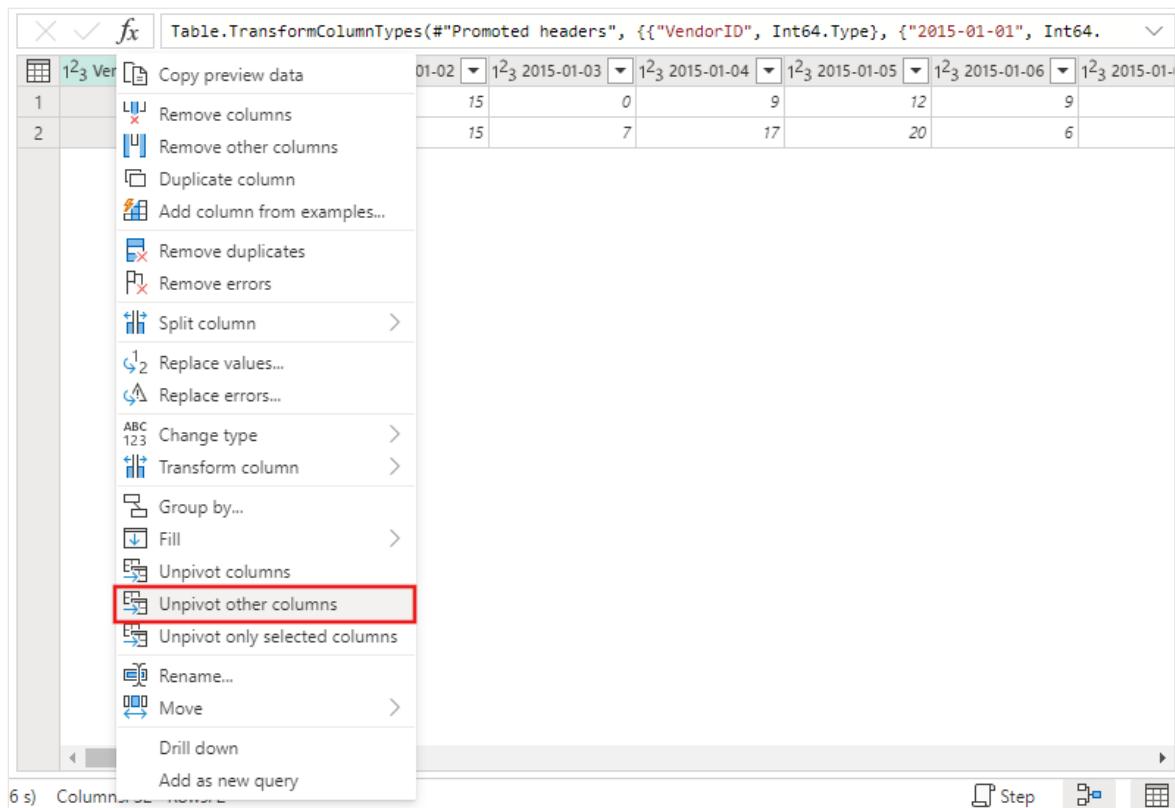
1. Reviewing the data, we see the headers appear to be in the first row. Promote them to headers by selecting the table's context menu at the top left of the preview grid area to select **Use first row as headers**.



⚠ Note

After promoting the headers, you can see a new step added to the **Applied steps** pane at the top of the dataflow editor to the data types of your columns.

2. Right-click the **VendorID** column, and from the context menu displayed, select the option **Unpivot other columns**. This allows you to transform columns into attribute-value pairs, where columns become rows.



- With the table unpivoted, rename the **Attribute** and **Value** columns by double-clicking them and changing **Attribute** to **Date** and **Value** to **Discount**.

	VendorID	Date	Discount
1		2015-01-01	16
2		2015-01-02	15
3		2015-01-03	0
4		2015-01-04	9
5		2015-01-05	12
6		2015-01-06	9
7		2015-01-07	9
8		2015-01-08	7
9		2015-01-09	5

- Change the data type of the Date column by selecting the data type menu to the left of the column name and choosing Date.

The screenshot shows the Power BI Data Editor interface. A context menu is open over the 'Date' column, listing various data type options. The 'Date' option is highlighted with a red box.

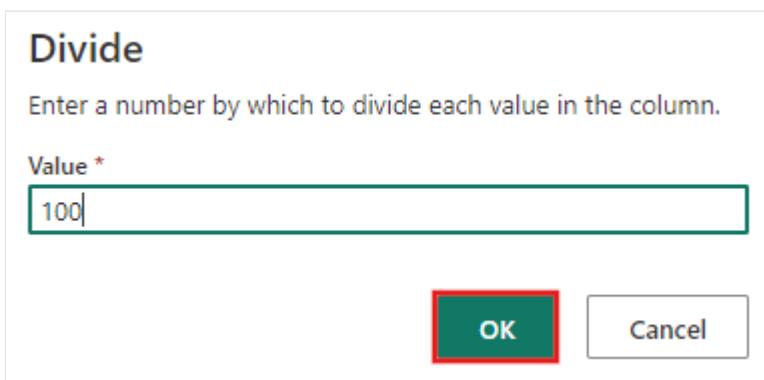
1	1	1.2	Decimal number
2	1	\$	Currency
3	1	123	Whole number
4	1	%	Percentage
5	1		Date/Time
6	1		Date
7	1		Time
8	1		Date/Time/Zone
9	1		Duration
10	1		Text
11	1		True/False
12	1	010	Binary
13	1	101	
14	1	ABC	Using locale...
15	1	123	
16	1	2015-01-16	3
17	1	2015-01-17	2

5. Select the **Discount** column and then select the **Transform** tab on the menu. Select **Number column**, and then select **Standard** numeric transformations from the submenu, and choose **Divide**.

The screenshot shows the Power BI Data Editor interface with the 'Transform' tab selected in the ribbon. A context menu is open over the 'Discount' column, showing the 'Divide' option under the 'Standard' submenu, which is highlighted with a red box.

1	1	2015-01-01	15
2	1	2015-01-02	0
3	1	2015-01-03	9
4	1	2015-01-04	12
5	1	2015-01-05	9
6	1	2015-01-06	9
7	1	2015-01-07	7

6. On the **Divide** dialog, enter the value 100.



Combine trips and discounts data

The next step is to combine both tables into a single table that has the discount that should be applied to the trip, and the adjusted total.

1. First, toggle the **Diagram view** button so you can see both of your queries.

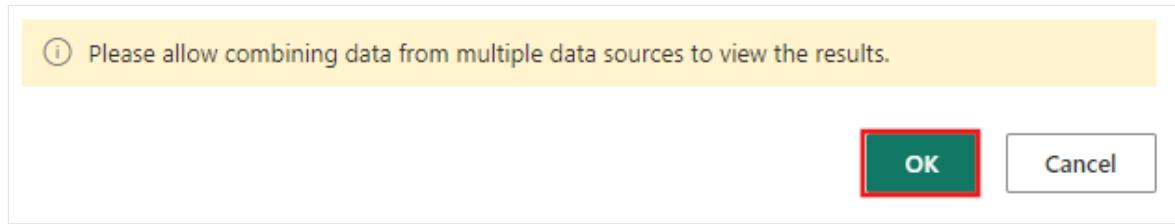
The screenshot shows the Power BI Data Editor interface. The ribbon at the top has the 'Transform' tab selected. Below the ribbon, there are several icons for data manipulation: Group by, Use first row as headers, Transpose, Reverse rows, Count rows, Any column, Text column, Number column, and Date and time column. On the left, a sidebar labeled 'Queries [2]' lists 'nyc_taxi' and 'Generated-NYC-T...'. The 'Generated-NYC-T...' query is currently selected, showing its 6 steps. The main area displays a table with columns: VendorID, Date, and Discount. The data shows various dates from January 1st to 10th, and corresponding discounts like 0.16, 0.15, 0.09, etc. At the bottom right of the table, there is a 'Diagram view' button, which is highlighted with a red box. The status bar at the bottom indicates 'Completed (1.98 s) Columns: 3 Rows: 62'.

2. Select the **nyc_taxi** query, and on the **Home** tab, Select the **Combine** menu and choose **Merge queries**, then **Merge queries as new**.

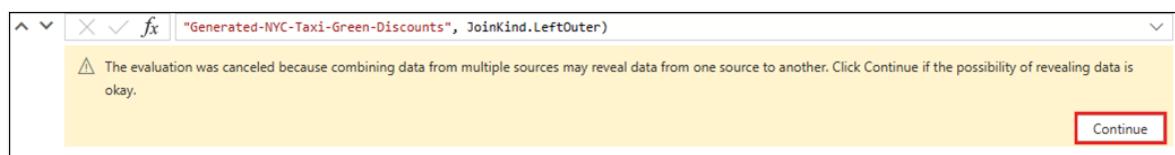
3. On the **Merge** dialog, select **Generated-NYC-Taxi-Green-Discounts** from the **Right table for merge** drop down, and then select the "light bulb" icon on the top right of the dialog to see the suggested mapping of columns between the two tables.

Choose each of the two suggested column mappings, one at a time, mapping the VendorID and date columns from both tables. When both mappings are added, the matched column headers are highlighted in each table.

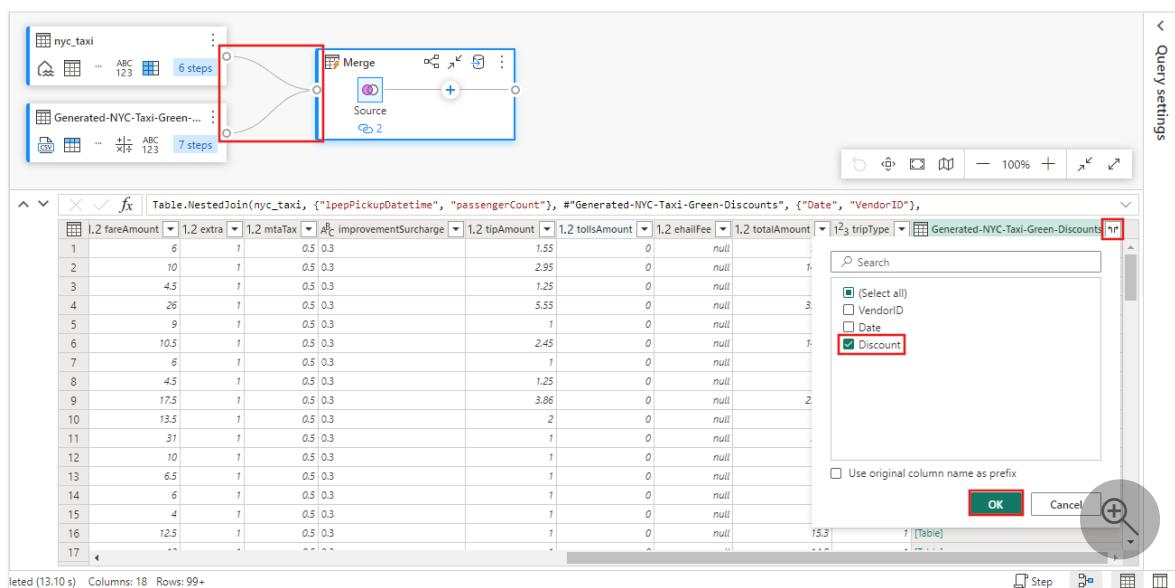
4. A message is shown asking you to allow combining data from multiple data sources to view the results. Select **OK** on the Merge dialog.



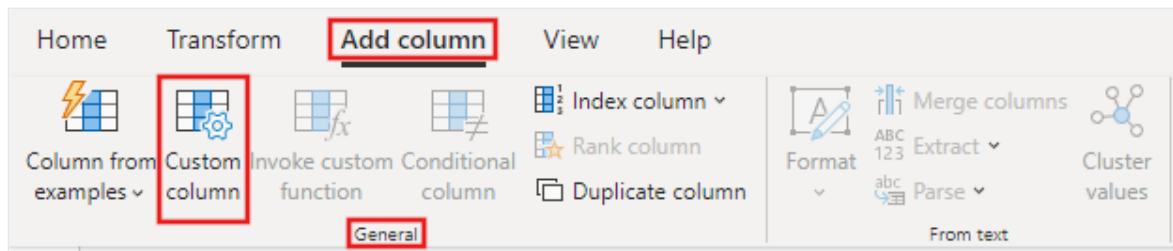
5. In the table area, you'll initially see a warning that "The evaluation was canceled because combining data from multiple sources may reveal data from one source to another. Select continue if the possibility of revealing data is okay." Select **Continue** to display the combined data.



6. Notice how a new query was created in Diagram view showing the relationship of the new Merge query with the two queries you previously created. Looking at the table pane of the editor, scroll to the right of the Merge query column list to see a new column with table values is present. This is the "Generated NYC Taxi-Green-Discounts" column, and its type is **[Table]**. In the column header there's an icon with two arrows going in opposite directions, allowing you to select columns from the table. Deselect all of the columns except **Discount**, and then select **OK**.



7. With the discount value now at the row level, we can create a new column to calculate the total amount after discount. To do so, select the **Add column** tab at the top of the editor, and choose **Custom column** from the **General** group.



8. On the **Custom column** dialog, you can use the [Power Query formula language](#) (also known as M) to define how your new column should be calculated. Enter **TotalAfterDiscount** for the **New column name**, select **Currency** for the **Data type**, and provide the following M expression for the **Custom column formula**:

`if [totalAmount] > 0 then [totalAmount] * (1 -[Discount]) else [totalAmount]`

Then select **OK**.

Custom column ?

Add a column that is computed from other columns or values.

New column name *

Data type

Currency

Custom column formula *

```
= if [totalAmount] > 0 then [totalAmount] * ( 1 - [Discount] ) else [totalAmount]
```

Available column(s)

- vendorid
- lpepPickupDatetime
- passengerCount
- tripDistance
- pickupLongitude
- dropoffLatitude
- storeAndEventFlag

Insert column OK Cancel

[Learn more about Power Query formulas](#)

9. Select the newly created **TotalAfterDiscount** column and then select the **Transform** tab at the top of the editor window. On the **Number column** group, select the **Rounding** drop down and then choose **Round....**

Home Transform Add column View Help

Group Use first row as headers by ... ABC 123 Data type: Currency v Rename Fill v Merge columns Trigonometry v
Count rows Reverse rows Pivot column Move v Extract v Statistics Standard Scientific Rounding v
Replace values v Detect data type Mark as key Unpivot columns v Convert to list Split column v Date and time column v
Any column

Queries [3]

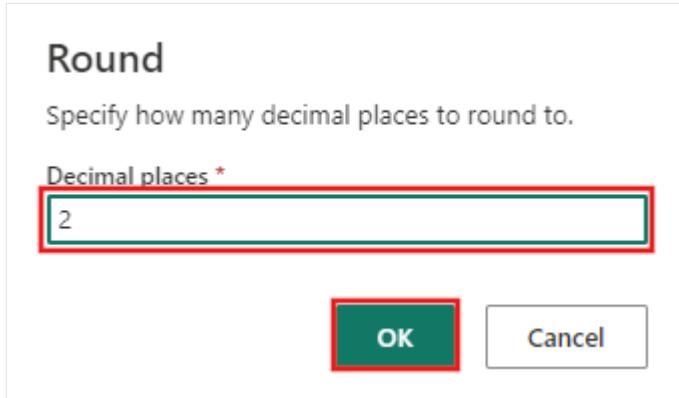
nyc_taxi ... ABC 123 6 steps

Merge ... ABC 123 7 steps

Generated-NYC-Taxi-Green-... ... ABC 123 7 steps

Round...

10. On the Round dialog, enter 2 for the number of decimal places and then select OK.



11. Change the data type of the lpepPickupDatetime from Date to Date/Time.

The screenshot shows the Power Query Editor interface. A table is displayed with various columns. The 'lpepPickupDatetime' column has its data type dropdown menu open, showing options like 'Decimal number', 'Currency', 'Whole number', 'Percentage', 'Date/Time' (which is highlighted with a red box), 'Date', 'Time', 'Date/Time/Zone', 'Duration', 'Text', 'True/False', 'Binary', and 'Using locale...'. The table data includes columns for vendorID, lpepPickupDatetime, passengerCount, tripDistance, pickupLongitude, dropoffLatitude, storeAndFwdFlag, paymentType, and a timestamp.

12. Finally, expand the **Query settings** pane from the right side of the editor if it isn't already expanded, and rename the query from **Merge to Output**.

The screenshot shows the Microsoft Data Factory Power Query Editor interface. On the left, there's a navigation bar with options like Home, Create, Browse, Data hub, Monitoring hub, Workspaces, and DF Getting Started. The main area is titled 'Power Query' and shows a 'Dataflow 4' tab. The 'Add column' tab is selected. A query named 'Ranked Customers' is displayed, which merges two tables: 'Customers' and 'Orders'. The 'Customers' table has 2 steps, and the 'Orders' table has 3 steps. The 'Ranked Customers' table has 2 steps and includes a 'Source' step, followed by 'Removed other c...', 'Expanded Orders...', and 'Added rank col...'. Below the table, a preview pane shows a table with columns: CustomerID, CompanyName, Count, and Rank. The preview pane also indicates 'Completed (2.22 s)' and 'Columns: 4 Rows: 91'. On the right, there are 'Query settings' (highlighted with a red box) and 'Applied steps' sections. The 'Properties' section shows the entity type as 'Custom' and the name as 'Ranked Customers'. The 'Applied steps' section lists 'Source', 'Removed o...', 'Expanded O...', and 'Added rank...'. At the bottom right, there's a green 'Publish' button.

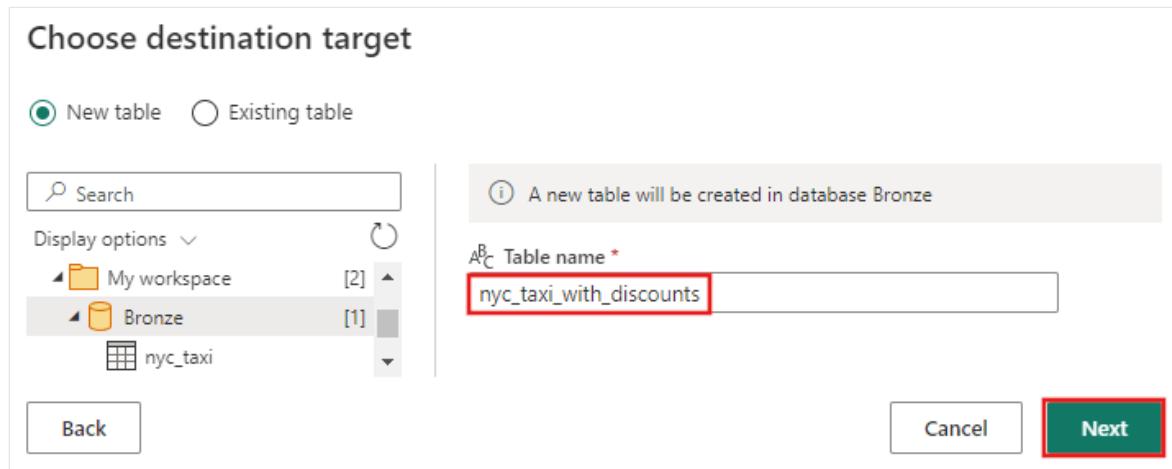
Load the output query to a table in the Lakehouse

With the output query now fully prepared and with data ready to output, we can define the output destination for the query.

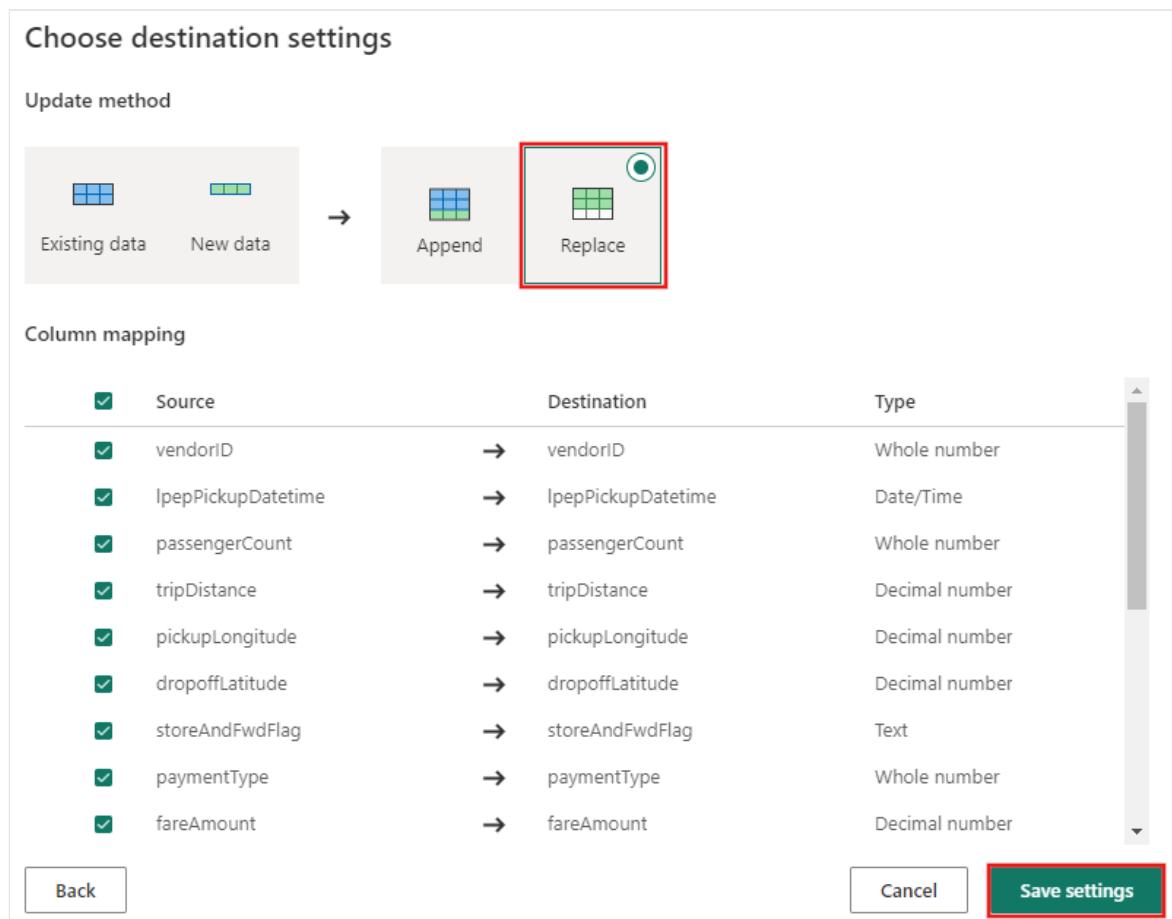
1. Select the **Output** merge query created previously. Then select the **Home** tab in the editor, and **Add data destination** from the **Query** grouping, to select a **Lakehouse** destination.

The screenshot shows the Microsoft Data Factory Power Query Editor interface with the 'Home' tab selected. In the top navigation bar, the 'Home' tab is highlighted. Below it, there are several icons: 'Get data', 'Enter data', 'Options', 'Manage parameters', 'Refresh', 'Advanced editor', 'Properties', 'Query' (which is highlighted with a red box), 'Choose columns', 'Remove columns', and 'Keep rows'. The main area shows two queries: 'nyc_taxi' (6 steps) and 'Generated-NYC-Taxi-Green...' (7 steps). The 'Generated-NYC-Taxi-Green...' query has a 'Source' step, followed by 'Expanded Gener...' and 'Added custom...'. On the right, there's a 'Query settings' panel with a 'Properties' section showing the name 'Lakehouse' (highlighted with a red box) and an 'Applied steps' section. The 'Applied steps' section lists 'Source', 'Removed o...', 'Expanded O...', and 'Added rank...'. At the bottom right, there's a green 'Publish' button.

2. On the **Connect to data destination** dialog, your connection should already be selected. Select **Next** to continue.
3. On the **Choose destination target** dialog, browse to the Lakehouse where you wish to load the data and name the new table *nyc_taxi_with_discounts*, then select **Next** again.



4. On the **Choose destination settings** dialog, leave the default **Replace** update method, double check that your columns are mapped correctly, and select **Save settings**.



5. Back in the main editor window, confirm that you see your output destination on the **Query settings** pane for the **Output** table, and then select **Publish**.

6. (Optional) On the workspace page, you can rename your dataflow by selecting the ellipsis to the right of the dataflow name that appears after you select the row, and choosing **Properties**.

The screenshot shows the Microsoft Fabric workspace interface. At the top, there's a header with a user icon, the title 'My workspace', and navigation links for '+ New', 'Upload', and '...'. A search bar on the right says 'Filter by keyword'. Below the header is a table with columns: 'Name', 'Type', and 'Owner'. The table contains four rows: 'pipeline1' (Data pipeline, Your User), 'Dataflow 1' (Dataflow Gen2, Your User), 'Dataflow 2' (Dataflow Gen2, Your User), and 'Dataflow 3' (Dataflow Gen2, Your User). For the 'Dataflow 3' row, a context menu is open, listing options: Delete, Edit, Export .json, Properties (which is highlighted with a red box), Refresh history, Settings, and View lineage.

Name	Type	Owner
pipeline1	Data pipeline	Your User
Dataflow 1	Dataflow Gen2	Your User
Dataflow 2	Dataflow Gen2	Your User
Dataflow 3	Dataflow Gen2	Your User

7. Select the refresh icon for the dataflow after selecting its row, and when complete, you should see your new Lakehouse table created as configured in the **Data destination** settings.

The screenshot shows the Microsoft Fabric workspace interface. The table from the previous step is shown again, but now the row for 'nyc_taxi_data_with_discounts' is selected. The refresh icon (a circular arrow) next to this row is highlighted with a red box.

Name	Type	Owner	Refreshed
pipeline1	Data pipeline	Your User	—
Dataflow 1	Dataflow Gen2	Your User	—
Dataflow 2	Dataflow Gen2	Your User	—
nyc_taxi_data_with_discounts	Dataflow Gen2	Your User	—

8. Check your Lakehouse to view the new table loaded there.

Next steps

In this second module to our end-to-end tutorial for your first data integration using Data Factory in Microsoft Fabric, you learned how to:

- ✓ Create a new Dataflow Gen2.
- ✓ Import and transform sample data.
- ✓ Import and transform text/CSV data.
- ✓ Merge data from both data sources into a new query.
- ✓ Transform data and generate new columns in a query.
- ✓ Configure an output destination source for a query.
- ✓ Rename and refresh your new dataflow.

Continue to the next section now to integrate your data pipeline.

Module 3: Automate and send notifications with Data Factory

Module 3: Automate and send notifications with Data Factory

Article • 05/23/2023

You'll complete this module in 10 minutes to send an email notifying you when all the jobs in a pipeline are complete, and configure it to run on a scheduled basis.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

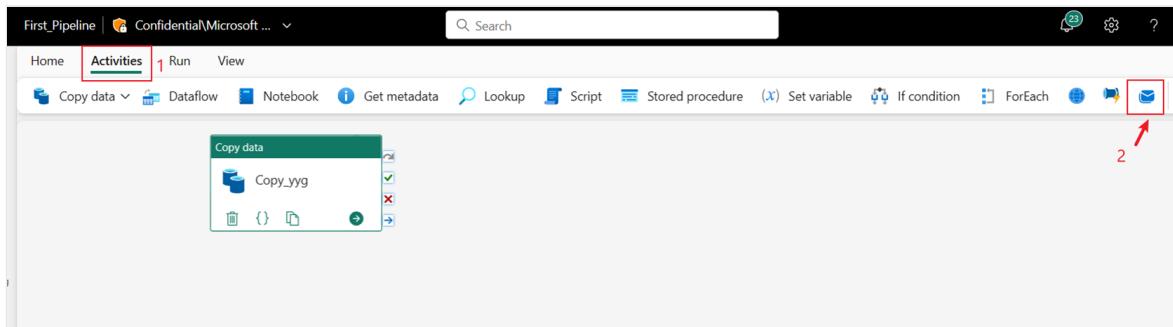
In this module you learn how to:

- Add an Office 365 Outlook activity to send the output of a Copy activity by email.
- Add schedule to run the pipeline.
- *(Optional)* Add a dataflow activity into the same pipeline.

Add an Office 365 Outlook activity to your pipeline

We use the pipeline you created in [Module 1: Create a pipeline in Data Factory](#).

1. Select the **Activities** tab in the pipeline editor and find the Office Outlook activity.



2. Select OK to grant consent to use your email address.

Grant consent



To use an Office365Outlook activity, you must grant consent to use your email address.

Ok

Cancel

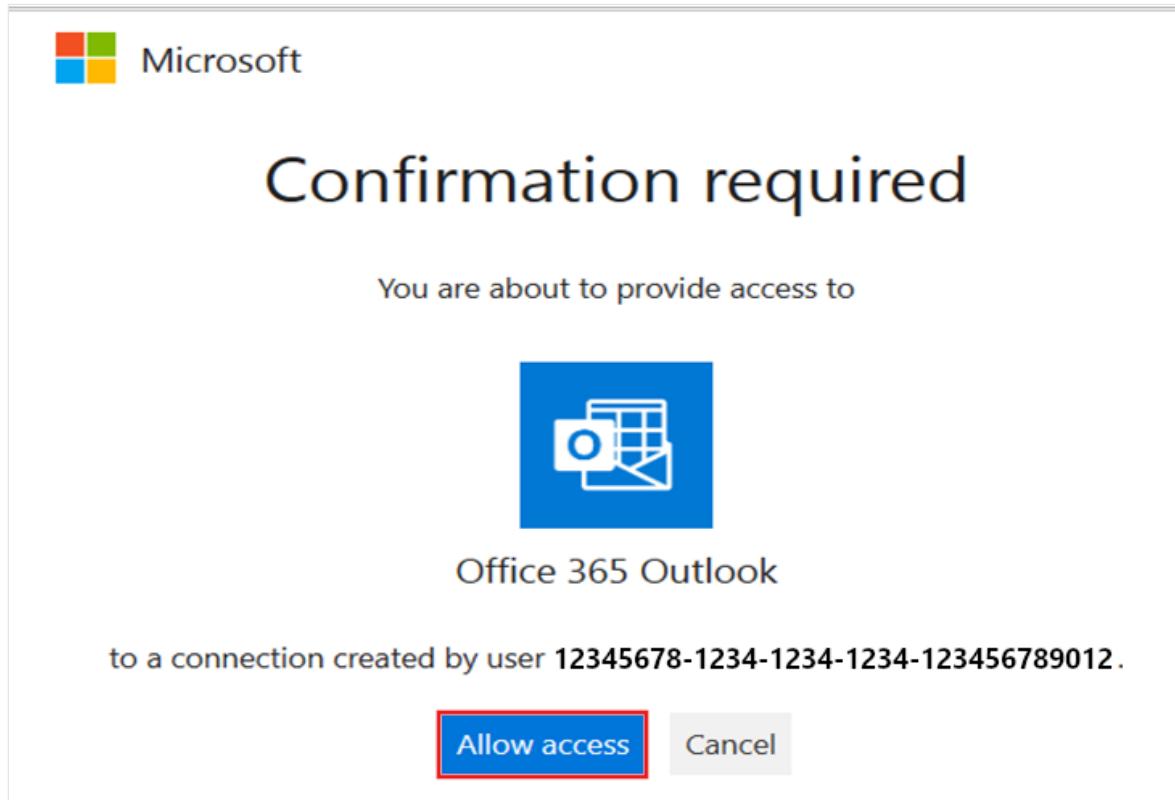
3. Select the email address you want to use.

The screenshot shows a Microsoft Edge browser window with the title "Sign in to your account - Work - Microsoft Edge". The URL in the address bar is "https://login.microsoftonline.com/common/oauth2/authorize...". The main content area displays the "Pick an account" screen from Microsoft's identity platform. It features the Microsoft logo and the text "Pick an account". Below this, there is a card for a user account: "youremailaccount@contoso.com" and "Connected to Windows". At the bottom, there is a button labeled "Use another account" with a plus sign icon.

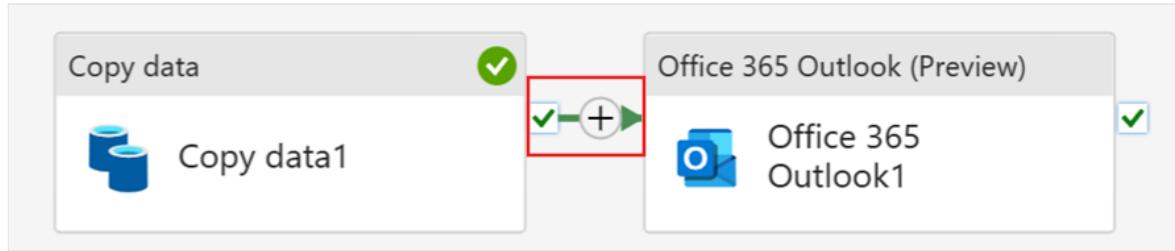
ⓘ Note

The service doesn't currently support personal email. You must use an enterprise email address.

4. Select **Allow access** to confirm.

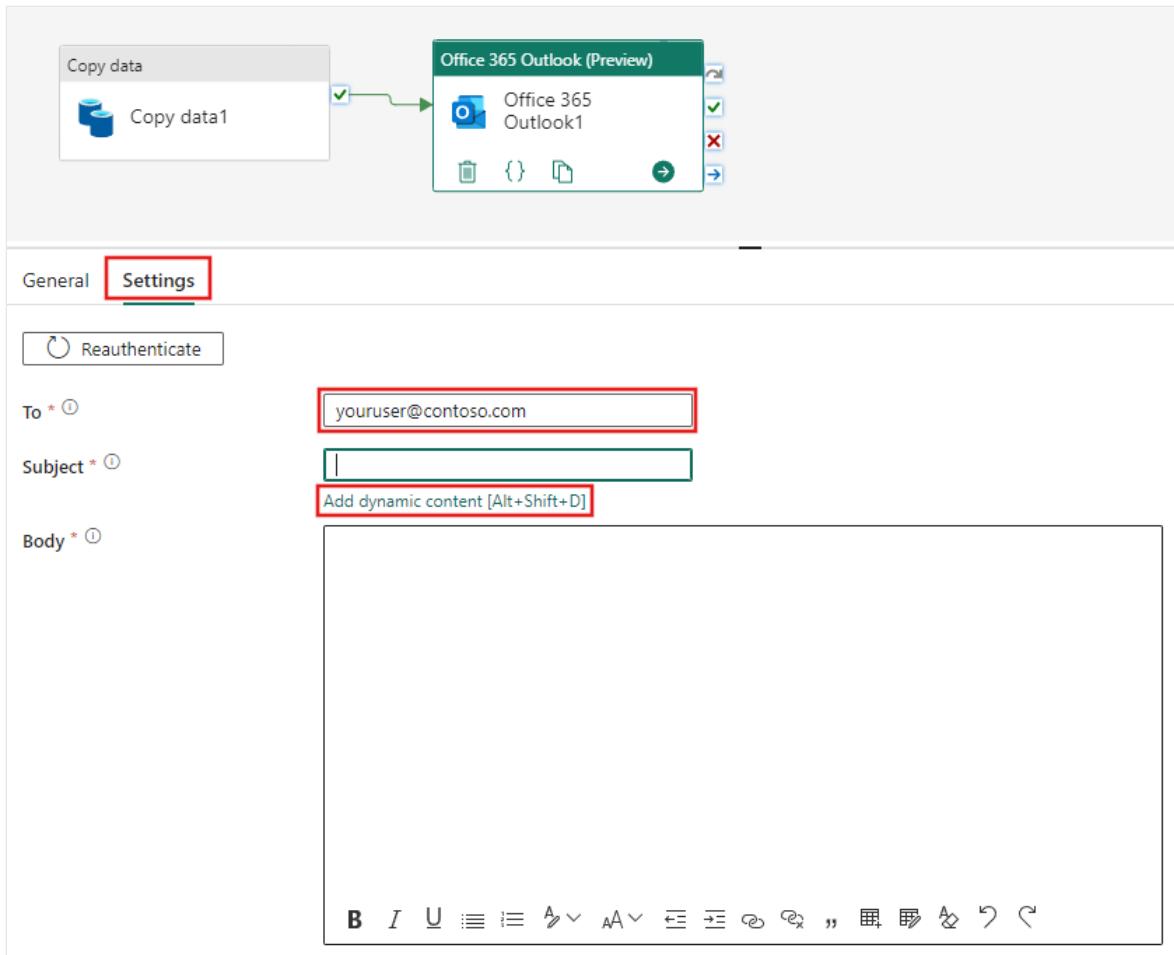


5. Select and drag the **On success** path (a green checkbox on the top right side of the activity in the pipeline canvas) from your Copy activity to your new Office 365 Outlook activity.



6. Select the Office 365 Outlook activity from the pipeline canvas, then select the **Settings** tab of the property area below the canvas to configure the email.

- Enter your email address in the **To** section. If you want to use several addresses, use ; to separate them.
- For the **Subject**, select the field so that the **Add dynamic content** option appears, and then select it to display the pipeline expression builder canvas.



7. The Pipeline expression builder dialog appears. Enter the following expression, then select OK:

```
@concat('DI in an Hour Pipeline Succeeded with Pipeline Run Id', pipeline().RunId)
```

Pipeline expression builder



Add dynamic content below using any combination of expressions, functions and system variables.

```
@concat('DI in an Hour Pipeline Succeeded with Pipeline Run Id',  
        pipeline().RunId)
```

[Clear contents](#)

[Activity outputs](#)

[Parameters](#)

[System variables](#)

[Functions](#)

[Variables](#)

[Search](#)

[Copy data1](#)

[Copy data1 activity output](#)

[Copy data1](#)

[Copy data1 pipeline return value \(preview\)](#)

[OK](#)

[Cancel](#)

8. For the **Body**, select the field again and choose the **Add dynamic content** option when it appears below the text area. Add the following expression again in the **Pipeline expression builder** dialog that appears, then select **OK**:

```
@concat('RunID = ', pipeline().RunId, ' ; ', 'Copied rows ', activity('Copy  
data1').output.rowsCopied, ' ; ''Throughput ', activity('Copy  
data1').output.throughput)
```

Note

Replace **Copy data1** with the name of your own pipeline copy activity.

9. Finally select the **Home** tab at the top of the pipeline editor, and choose **Run**. Then select **Save and run** again on the confirmation dialog to execute these activities.

The screenshot shows the Azure Data Factory pipeline editor interface. At the top, there's a ribbon with tabs: Home (highlighted with a red box), Activities, Run, View, and a toolbar with various icons. Below the ribbon, the pipeline is visualized as a sequence of activities connected by arrows. The first activity is 'Copy data' (Copy data1), which has a green checkmark indicating success. An arrow points from it to the second activity, 'Office 365 Outlook (Preview)'. This second activity also has a green checkmark and is associated with 'Office 365 Outlook1'. The pipeline editor also displays settings for the second activity, including fields for 'To', 'Subject', and 'Body'.

- After the pipeline runs successfully, check your email to find the confirmation email sent from the pipeline.

This screenshot shows the 'Output' tab of the pipeline run details page. It displays the run ID: 3275d11d-f672-4742-b568-431f90273a06. Below the run ID, a table lists the activities and their status:

Name	Type	Run start	Duration	Status	Run ID
Office 365 Outlook1	Office 365 Ou...	5/9/2023, 3:47:4	00:00:01	Succeeded	e279a5b5-51f0-42b0-b98d-744e5d
Copy data1	Copy data	5/9/2023, 3:42:4	00:05:04	Succeeded	a45a2344-fa7f-48be-854c-b8edae4

This screenshot shows an email message in an Outlook inbox. The subject of the email is 'DI in an Hour Pipeline Succeeded with Pipeline Run Id3275d11d-f672-474...'. The message body contains the following text:

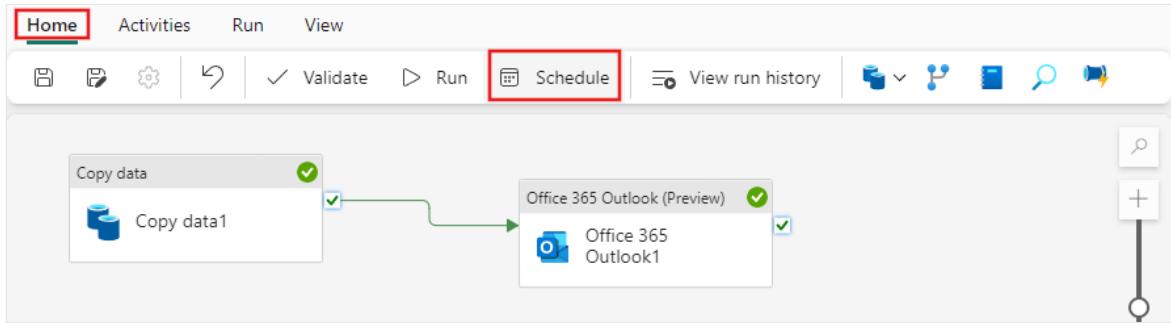
DI in an Hour Pipeline Succeeded with Pipeline Run Id3275d11d-f672-4742-b568-431f90273a06

RunID = 3275d11d-f672-4742-b568-431f90273a06 ; Copied rows 76513115 ; Throughput 143896.359

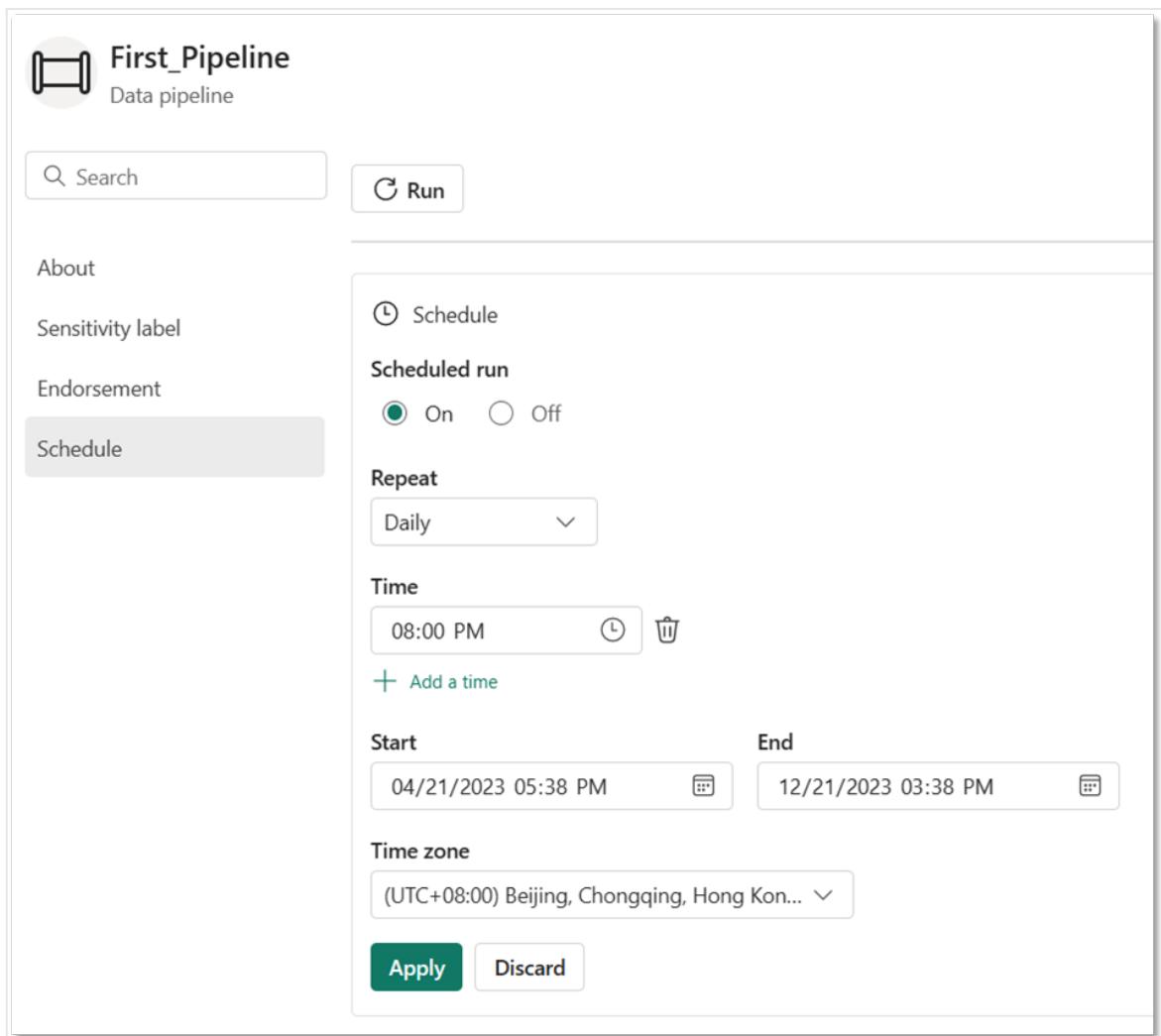
Schedule pipeline execution

Once you finish developing and testing your pipeline, you can schedule it to execute automatically.

1. On the **Home** tab of the pipeline editor window, select **Schedule**.



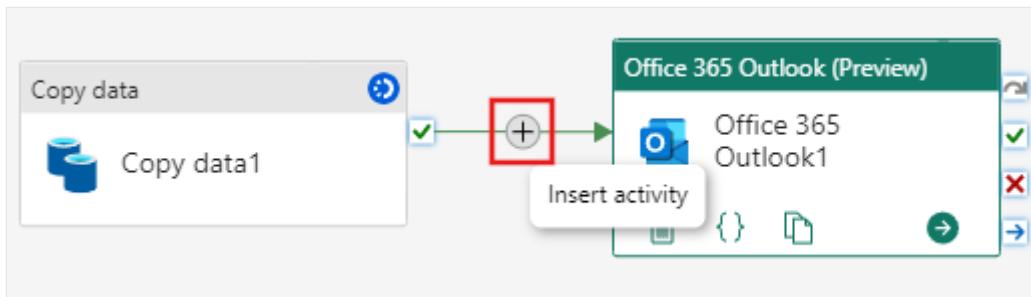
2. Configure the schedule as required. The example here schedules the pipeline to execute daily at 8:00 PM until the end of the year.



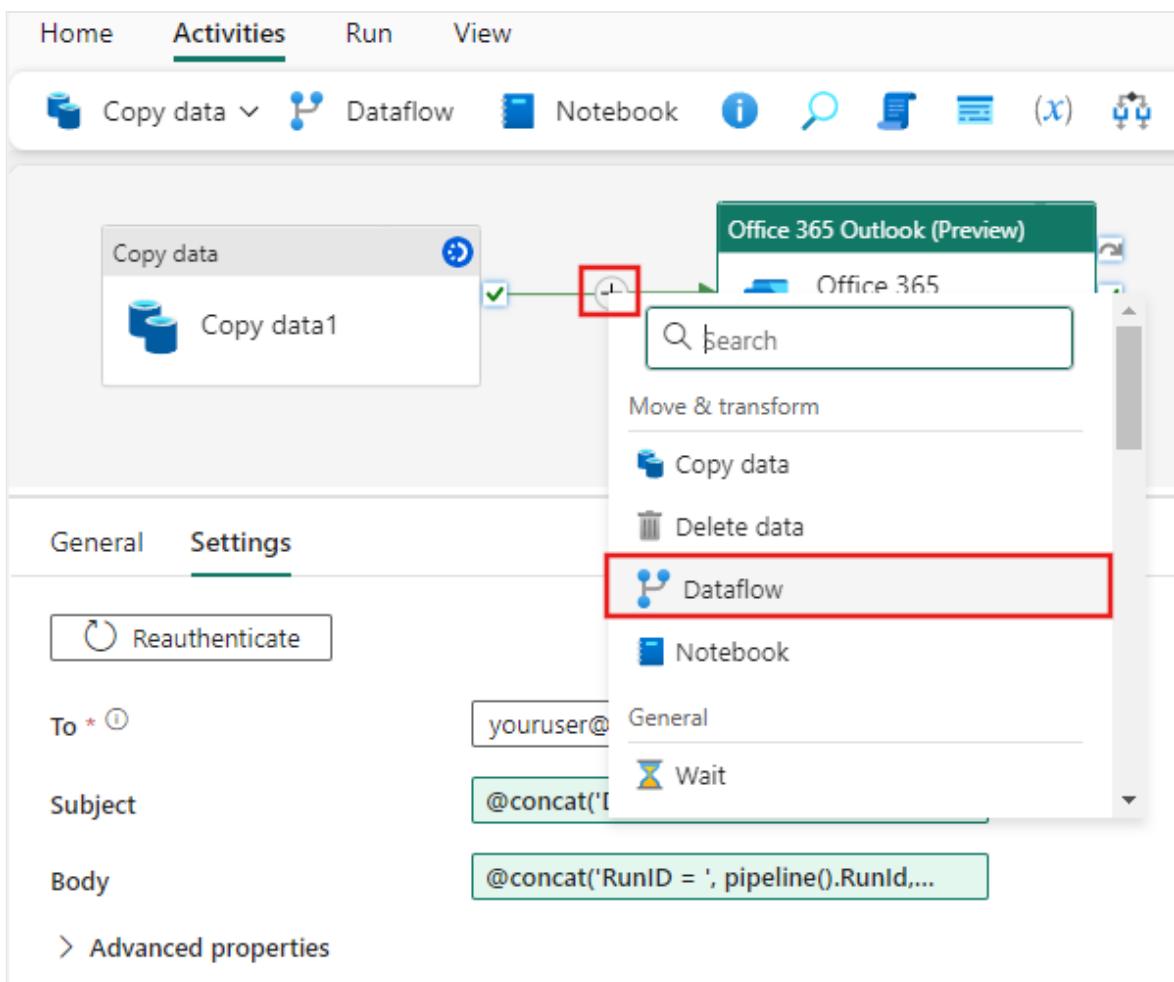
(Optional) Add a Dataflow activity to the pipeline

You can also add the dataflow you created in [Module 2: Create a dataflow in Data Factory](#) into the pipeline.

1. Hover over the green line connecting the Copy activity and the Office 365 Outlook activity on your pipeline canvas, and select the + button to insert a new activity.

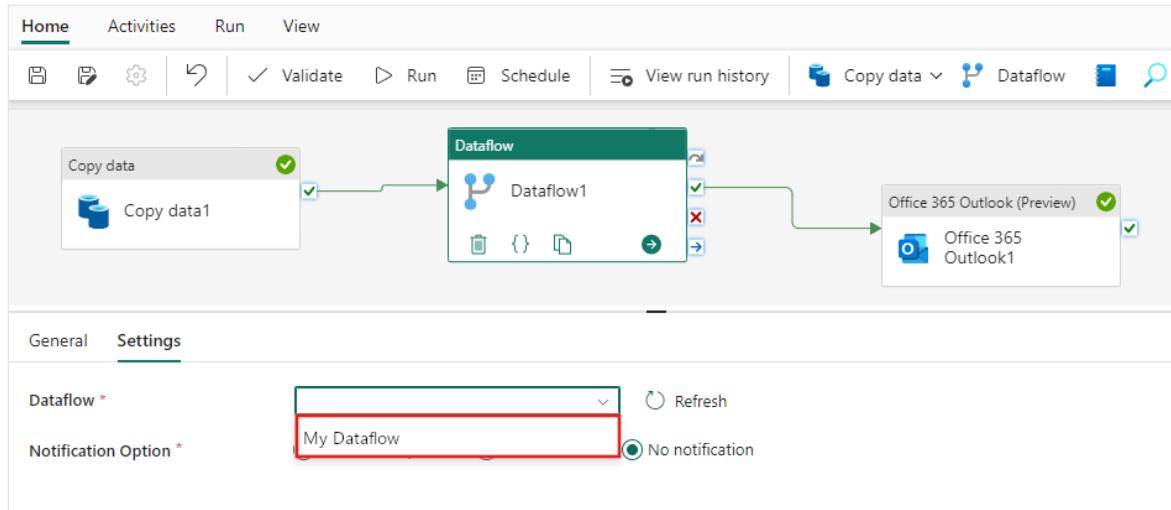


2. Choose Dataflow from the menu that appears.



3. The newly created Dataflow activity is inserted between the Copy activity and the Office 365 Outlook activity, and selected automatically, showing its properties in

the area below the canvas. Select the **Settings** tab on the properties area, and then select your dataflow created in [Module 2: Create a dataflow in Data Factory](#).



Next steps

In this third module to our end-to-end tutorial for your first data integration using Data Factory in Microsoft Fabric, you learned how to:

- ✓ Use a Copy activity to ingest raw data from a source store into a table in a data Lakehouse.
- ✓ Use a Dataflow activity to process the data and move it into a new table in the Lakehouse.
- ✓ Use an Office 365 Outlook activity to send an email notifying you once all the jobs are complete.
- ✓ Configure the pipeline to run on a scheduled basis.
- ✓ *(Optional)* Insert a Dataflow activity in an existing pipeline flow.

Now that you completed the tutorial, learn more about how to monitor pipeline runs:

[Monitor pipeline runs](#)

Connector overview

Article • 05/23/2023

Data Factory in Microsoft Fabric offers a rich set of connectors that allow you to connect to different types of data stores. You can take advantage of those connectors to transform data in dataflows or move a PB-level of dataset with high-scale in a data pipeline.

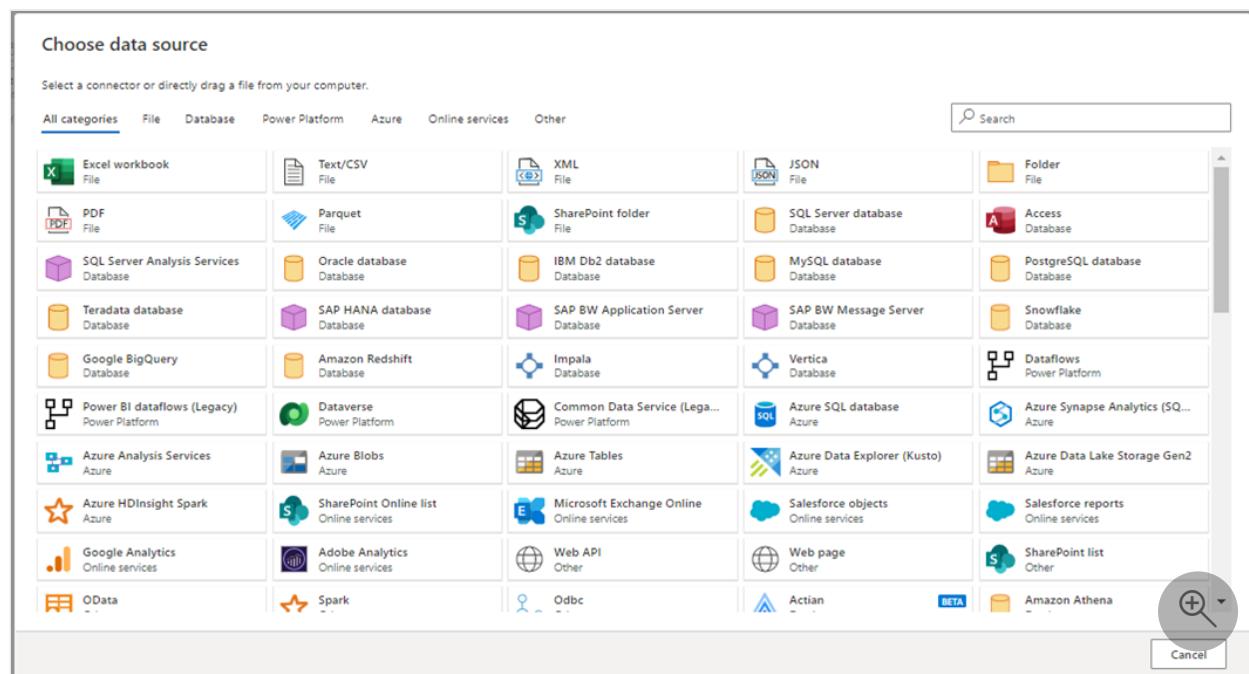
ⓘ Important

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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported data connectors in dataflows

Dataflows provide data ingestion and transformation capabilities over a wide range of data sources. These data sources include various types of files, databases, online, cloud, and on-premises data sources. There are greater than 135 different data connectors, which are accessible from the dataflows authoring experience within the get data experience.



You can find a comprehensive list of all connectors supported through our [public Power Query connectors reference](#). Supported connectors match the ones marked as

supported in the **Power BI (Dataflows)** column in the Power Query reference table.

The following connectors are currently available for output destinations in Dataflow Gen2:

- Azure Data Explorer
- Azure SQL
- Data Warehouse
- Lakehouse

Supported data stores in data pipeline

Data Factory in Microsoft Fabric supports the following data stores in a data pipeline via Copy, Lookup, Get Metadata, and Delete Data activities. Go to each data store to learn the supported capabilities and the corresponding configurations in detail.

Category	Data store	Copy activity (source/destination)	Lookup activity	Get Metadata activity	Delete activity	Script activity	Stored Procedure activity
Workspace	Data Warehouse	✓/✓	✓	✓	-	✓	✓
	KQL Database	✓/✓	✓	-	-	-	-
	Lakehouse	✓/✓	-	-	✓	-	-
Azure	Azure Blob Storage	✓/✓	✓	✓	✓	-	-
	Azure Cosmos DB for NoSQL	✓/✓	✓	✓	✓	-	-
	Azure Data Lake Storage Gen1	✓/✓	✓	✓	✓	-	-
	Azure Data Lake Storage Gen2	✓/✓	✓	✓	✓	-	-

Category	Data store	Copy activity (source/destination)	Lookup activity	Get Metadata activity	Delete activity	Script activity	Stored Procedure activity
	Azure Database for PostgreSQL	✓/✓	✓	-	-	-	-
	Azure SQL Database	✓/✓	✓	✓	✓	✓	✓
	Azure SQL Database Managed Instance	✓/✓	✓	✓	-	✓	✓
	Azure SQL Explorer	✓/✓	✓	-	-	-	-
	Azure Synapse Analytics	✓/✓	✓	✓	-	✓	✓
	Azure Table Storage	✓/✓	✓	-	-	-	-
Database	Amazon Redshift	✓/-	✓	-	-	-	-
	Amazon RDS for SQL Server	✓/-	✓	✓	✓	-	-
	Apache Impala	✓/-	✓	-	-	-	-
	Hive	✓/-	✓	-	-	-	-
	PostgreSQL	✓/-	✓	-	-	-	-
	Spark	✓/-	✓	-	-	-	-
	SQL Server	✓/✓	✓	✓	-	✓	✓
File	Amazon S3	✓/-	✓	✓	✓	-	-
	Amazon S3 Compatible	✓/-	✓	✓	✓	-	-

Category	Data store	Copy activity (source/destination)	Lookup activity	Get Metadata activity	Delete activity	Script activity	Stored Procedure activity
	Google Cloud Storage	✓/-	✓	✓	✓	-	-
Generic	HTTP	✓/-	✓	-	-	-	-
	OData	✓/-	✓	-	-	-	-
	REST	✓/✓	-	-	-	-	-
Services and apps	Dataverse	✓/✓	✓	-	-	-	-
	Dynamics CRM	✓/✓	✓	-	-	-	-
	Microsoft 365	✓/-	-	-	-	-	-
	SharePoint Online List	✓/-	✓	-	-	-	-
	Snowflake	✓/✓	✓	-	-	✓	-

Next steps

- [How to copy data using copy activity](#)
- [Data source management](#)

Amazon S3 connector overview

Article • 05/23/2023

This Amazon S3 connector is supported in Data Factory for Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/-)	None	Basic
Lookup activity	None	Basic
GetMetadata activity	None	Basic
Delete activity	None	Basic

Next steps

- [How to create an Amazon S3 connection](#)
- [How to configure Amazon S3 in a copy activity](#)

How to create an Amazon S3 connection

Article • 05/23/2023

This article outlines the steps to create an Amazon S3 connection.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported authentication types

The Amazon S3 connector supports the following authentication types for copy activity.

Authentication type	Copy	Dataflow Gen2
Basic	✓	n/a

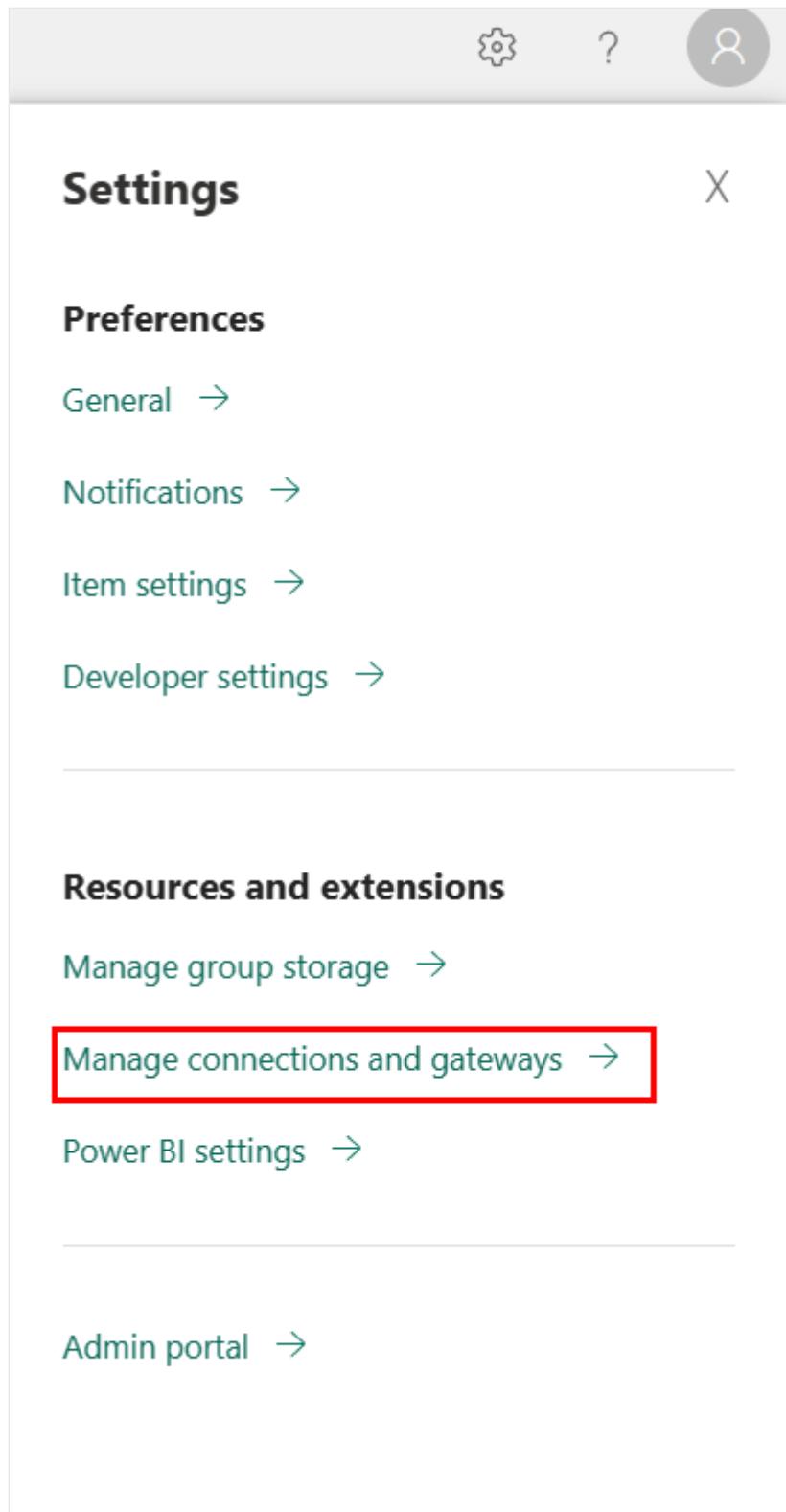
Prerequisites

The following prerequisites are required before you start:

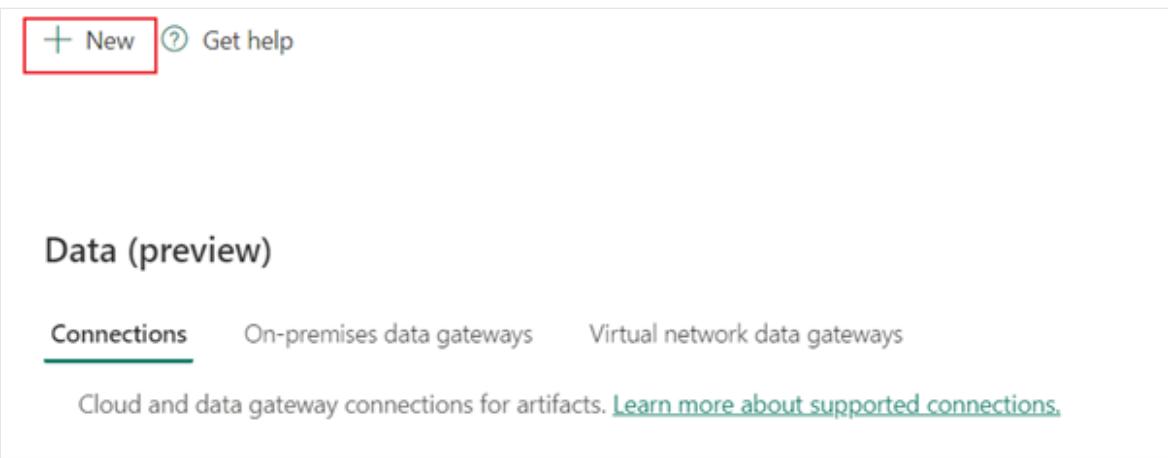
- A Microsoft Fabric tenant account with an active subscription. [Create an account for free](#).
- A Microsoft Fabric enabled Workspace. [Create a workspace](#).

Create a connection

1. From the page header in the Data Factory service, select **Settings**  > **Manage connections and gateways**.



2. Select **New** at the top of the ribbon to add a new data source.



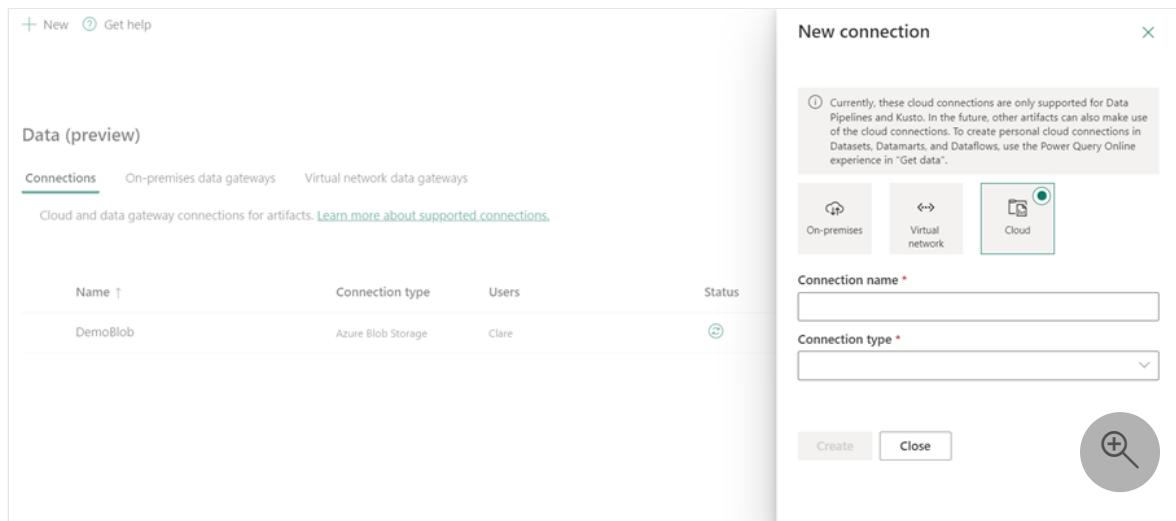
+ New Get help

Data (preview)

Connections On-premises data gateways Virtual network data gateways

Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)

The New connection pane is then displayed on the left side of the page.



+ New Get help

New connection

Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".

On-premises Virtual network **Cloud**

Connection name *

Connection type *

Create Close

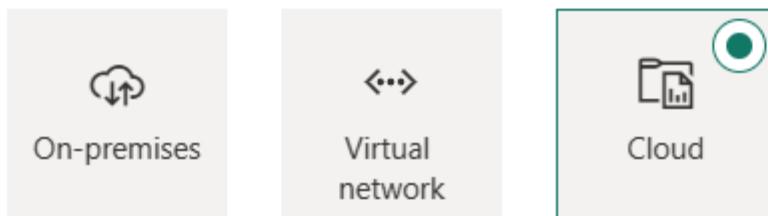
Set up the connection

Step 1: Specify the new connection name, type, data source path

New connection

X

- ⓘ Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



Connection name *

amazons3

Connection type *

Amazon S3

Url

< Your URL >

In the **New connection** pane, choose **Cloud**, and then specify the following fields:

- **Connection name:** Specify a name for your connection.
- **Connection type:** Select **Amazon S3** for your connection type.
- **Data source path:** Enter your account endpoint URL of your Amazon S3 data.

Step 2: Select and set your authentication

Under **Authentication method**, select your authentication method from the drop-down list and complete the related configuration. The Amazon S3 connector supports the following authentication types:

- Basic

Authentication

Authentication method *

Basic

Basic

Basic authentication

- **Username:** The user name to use to access the Amazon S3 endpoint.
- **Password:** The password for the specified username.

Authentication

Authentication method *

Basic

Username *

< Your username >

Password *

< Your password >

Step 3: Specify the privacy level that you want to apply

In the **General** tab, under **Privacy level**, select the privacy level that you want apply. Three privacy levels are supported. For more information, go to [privacy levels](#).

Step 4: Create your connection

Select **Create**. Your creation is successfully tested and saved if all the credentials are correct. If the credentials aren't correct, the creation fails with errors.

Data (preview)

Connections On-premises data gateways Virtual network data gateways

Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)

Name ↑	Connection type	Users	Status	Gateway cluster name
amazons3 ...	Amazon S3	< user name >	🕒	

Table summary

The following table contains connector properties that are supported in pipeline copy.

Name	Description	Required	Property	Copy
Connection name	A name for your connection.	Yes		✓
Connection type	Select Amazon S3 for your connection type.	Yes		✓
Data source path	Enter your account endpoint URL of your Amazon S3.	Yes		✓
Authentication	Go to Authentication	Yes		Go to Authentication

Authentication

The following table contains properties for the supported authentication type.

Name	Description	Required	Property	Copy
Basic				✓
- Username	The user name to use to access the Amazon S3 endpoint.	Yes		
- Password	The password for specified username.	Yes		

Next steps

- [How to configure Amazon S3 in a copy activity](#)

How to configure Amazon S3 in a copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in a data pipeline to copy data from and to Amazon S3.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Required permissions

To copy data from Amazon S3, make sure you've been granted the following permissions for Amazon S3 object operations: `s3:GetObject` and `s3:GetObjectVersion`.

In addition, `s3>ListAllMyBuckets` and `s3>ListBucket/s3:GetBucketLocation` permissions are required for operations like testing connection and browsing from root.

For the full list of Amazon S3 permissions, go to [Specifying Permissions in a Policy on the AWS site](#).

Supported format

Amazon S3 supports the following file formats. Refer to each article for format-based settings.

- Avro format
- [Binary format](#)
- [Delimited text format](#)
- [Excel format](#)
- JSON format
- ORC format
- [Parquet format](#)
- XML format

Supported configuration

For the configuration of each tab under copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Mapping](#)
- [Settings](#)

General

For **General** tab configuration, go to [General](#).

Source

The following properties are supported for Amazon S3 under the **Source** tab of a copy activity.

The screenshot shows the 'Source' tab configuration for an Amazon S3 connection. The 'Data store type' is set to 'External'. The 'Connection' dropdown shows 'https://s3.amazonaws.com'. The 'File path' section includes fields for 'Bucket', 'Directory', and 'File name', along with 'Browse', 'Preview data', and 'File settings' buttons. Under 'Advanced', 'File path type' is set to 'File path', 'Recursively' is checked, and 'Enable partition discovery' is unchecked. There are fields for 'Max concurrent connections' and 'Skip line count', both currently empty. A 'New' button is available for 'Additional columns'. A magnifying glass icon is located in the bottom right corner of the form area.

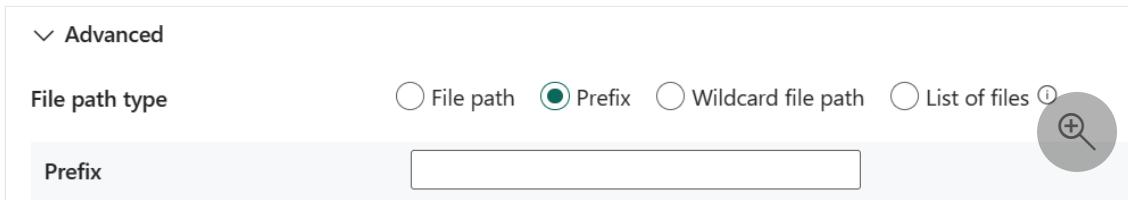
The following properties are **required**:

- **Data store type:** Select **External**.
- **Connection:** Select an **Amazon S3** connection from the connection list. If no connection exists, then create a new Amazon S3 connection by selecting **New**.
- **File path:** Select **Browse** to choose the file that you want to copy, or fill in the path manually.
- **File settings:** Select **File settings** to configure the file format. For settings of different file formats, refer to [Supported format](#) for detailed information.

Under **Advanced**, you can specify the following fields:

- **File path type:** You can choose **File path**, **Prefix**, **Wildcard file path**, or **List of files** as your file path type. The configuration of each of these settings is:

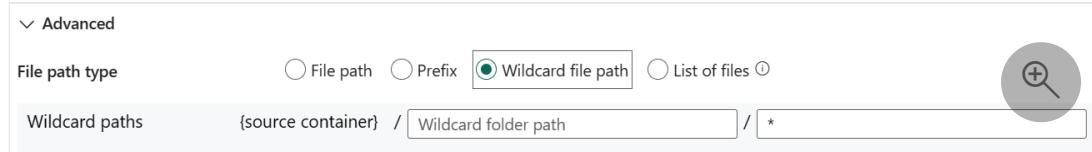
- **File path:** If you choose this type, the data can be copied from the given container or folder/file path specified previously.
- **Prefix:** Prefix for the S3 key name under the given bucket configured to filter source S3 files. S3 keys whose names start with `bucket/this_prefix` are selected. It utilizes S3's service-side filter, which provides better performance than a wildcard filter.



- **Wildcard file path:** Specify the folder or file path with wildcard characters under your given blob container to filter your source folders or files.

Allowed wildcards are: `*` (matches zero or more characters) and `?` (matches zero or single character). Use `^` to escape if your folder name has wildcard or this escape character inside.

- **Wildcard folder path:** The folder path with wildcard characters under the given bucket configured to filter source folders.



- **Wildcard file name:** The file name with wildcard characters under the given bucket and folder path (or wildcard folder path) to filter source files.
- **List of files:** Indicates to copy a given file set. Point to a text file that includes a list of files you want to copy, one file per line, which is the relative path to the path configured.



- **Recursively:** Indicates whether the data is read recursively from the subfolders or only from the specified folder. Note that when **recursive** is set to `true` and the

destination is a file-based store, an empty folder or subfolder isn't copied or created at the destination. Allowed values are **true** (default) and **false**. This property doesn't apply when you configure `fileListPath`.

- **Delete files after completion:** Indicates whether the binary files are deleted from the source store after successfully moving to the destination store. The file deletion is per file, so when a copy activity fails, you'll note some files have already been copied to the destination and deleted from the source, while others still remain on the source store. This property is only valid in the binary files copy scenario. The default value: `false`.
- **Max concurrent connection:** The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.

Mapping

For **Mapping** tab configuration, go to [Configure your mappings under mapping tab](#). If you choose Binary as your file format, mapping will not be supported.

Settings

For the **Settings** tab configuration, go to [Configure your other settings under settings tab](#).

Table summary

The following tables contain more information about the copy activity in Amazon S3.

Source information

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the source data store.	<your connection>	Yes	connection
File path	The file path of your source data.	<file path of your source >	Yes	container fileName

Name	Description	Value	Required	JSON script property
File path type	The file path type that you want to use.	<ul style="list-style-type: none"> • File path • Prefix • Wildcard folder path • List of files 	No	<ul style="list-style-type: none"> • prefix • wildcardFolderPath, wildcardFileName • path to file list
Recursively	Indicates whether the data is read recursively from the subfolders or only from the specified folder. Note that when recursive is set to true and the destination is a file-based store, an empty folder or subfolder isn't copied or created at the destination. Allowed values are true (default) and false . This property doesn't apply when you configure <code>fileListPath</code> .	Selected or unselect	No	recursive
Delete files after completion	Indicates whether the binary files will be deleted from the source store after successfully moving to the destination store. The file deletion is per file, so when a copy activity fails, you'll note some files have already been copied to the destination and deleted from the source, while others still remain on the source store. This property is only valid in the binary files copy scenario. The default value: false.	Selected or unselect	No	deleteFilesAfterCompletion

Name	Description	Value	Required	JSON script property
Max concurrent connection	The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.	<max concurrent connections>	No	maxConcurrentConnections

Next steps

[How to create an Amazon S3 connection](#)

Connect to Amazon S3 in dataflows

Article • 05/23/2023

The Amazon S3 connector isn't currently supported in Dataflow Gen2.

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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Next steps

- [How to create an Amazon S3 connection](#)
- [How to configure Amazon S3 in a copy activity](#)

Azure Blob Storage connector overview

Article • 05/23/2023

The Azure Blob Storage connector is supported in Data Factory in Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/destination)	None	Anonymous Key OAuth2 Service principal Shared Access Signature (SAS)
Lookup activity	None	Anonymous Key OAuth2 Service principal Shared Access Signature (SAS)
GetMetadata activity	None	Anonymous Key OAuth2 Service principal Shared Access Signature (SAS)

Next steps

- [How to create Azure Blob connection](#)
- [Copy data in Azure Blob Storage](#)
- [Connect to Azure Blob Storage in dataflows](#)

How to create an Azure Blob Storage connection

Article • 05/23/2023

Azure Blob Storage is Microsoft's object storage solution for the cloud. Blob storage is optimized for storing massive amounts of unstructured data. This article outlines the steps to create an Azure Blob Storage connection.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported authentication types

The Azure Blob Storage connector supports the following authentication types for copy and Dataflow Gen2 respectively.

Authentication type	Copy	Dataflow Gen2
Account key	✓	✓
Anonymous	✓	✓
Organizational account	✓	✓
Service Principal	✓	
Shared Access Signature (SAS)	✓	✓

ⓘ Note

For information about an Azure Blob Storage connection in Dataflow Gen2, go to [Connect to Azure Blob Storage in dataflows](#).

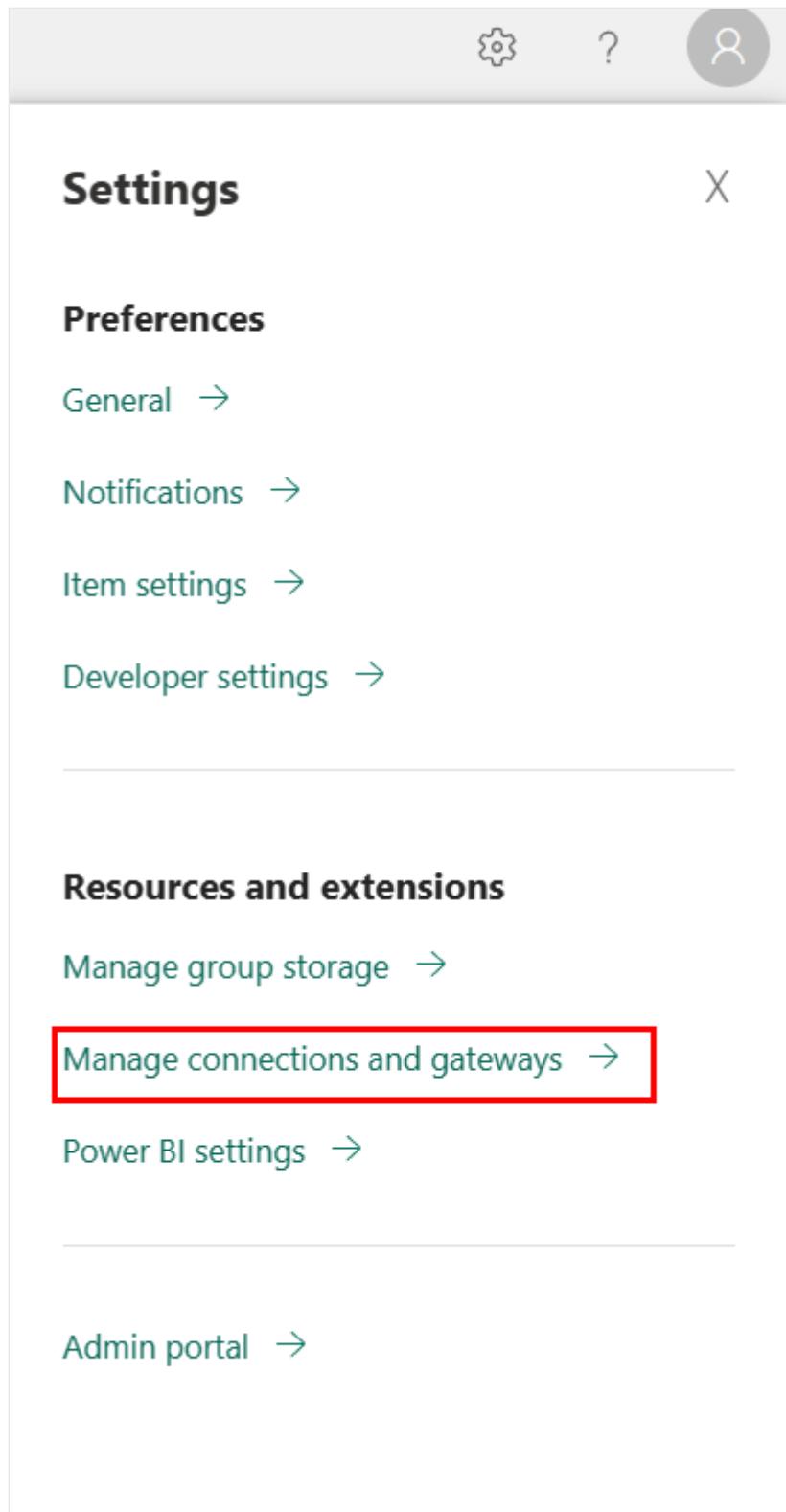
Prerequisites

The following prerequisites are required before you start:

- A Microsoft Fabric tenant account with an active subscription. [Create an account for free](#).
- A Microsoft Fabric enabled Workspace. [Create a workspace](#).

Go to Manage gateways to create a new connection

1. From the page header in Data Factory service, select **Settings**  > **Manage connections and gateways**.



2. Select **New** at the top of the ribbon to add a new data source.

[+ New](#) [Get help](#)

Data (preview)

Connections

On-premises data gateways

Virtual network data gateways

Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)



The New connection pane is then displayed on the left side of the page.

The screenshot shows the Power BI Data (preview) interface. On the left, there's a list of existing connections: "DemoBlob" (Azure Blob Storage, Clare). On the right, a "New connection" pane is open, titled "New connection". It includes a note about cloud connections being supported for Data Pipelines and Kusto. Below the note are three connection types: "On-premises" (selected), "Virtual network", and "Cloud" (highlighted with a green border). The "Connection name" field is empty, and the "Connection type" dropdown is also empty. At the bottom are "Create" and "Close" buttons, and a magnifying glass search icon.

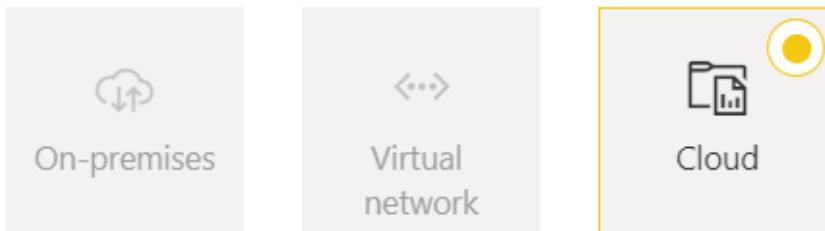
Set up your connection

Step 1: Specify the connection name, connection type, account, and domain

New connection

X

connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



Connection name *

azureblobstorage

Connection type *

Azure Blob Storage

Account *

testazurestorageaccount

Domain *

blob.core.windows.net

In the **New connection** pane, choose **Cloud**, and specify the following field:

- **Connection name:** Specify a name for your connection.
- **Connection type:** Select **Azure Blob Storage**.
- **Account:** Enter your Azure Blob Storage account name.
- **Domain:** Enter the domain of Azure Blob Storage: `blob.core.windows.net`.

Step 2: Select and set your authentication

Under **Authentication** method, select your authentication from the drop-down list and complete the related configuration. The Azure Blob Storage connector supports the following authentication types:

- Anonymous
- Key
- OAuth2
- Shared Access Signature (SAS)
- Service Principal

Authentication

Authentication method *

A screenshot of a dropdown menu titled "Authentication method *". The menu contains five options: "Anonymous", "Key", "OAuth2", "Shared Access Signature (SAS)", and "Service Principal". The "Anonymous" option is highlighted.

Anonymous authentication

Select **Anonymous** under **Authentication method**.

A screenshot of a dropdown menu titled "Authentication method *". The "Anonymous" option is highlighted and selected.

Key authentication

Specify the account key of your Azure Blob Storage. Go to your Azure Blob Storage account interface, browse to the **Access key** section, and get your account key.

Authentication

Authentication method *

Key

Account key *

.....

OAuth2 authentication

Open [Edit credentials](#). You'll see the sign in interface. Enter your account and password to sign in your account. After signing in, go back to the [New connection](#) page.

Authentication

Authentication method *

OAuth2

[Edit credentials](#)

Shared Access Signature (SAS) authentication

Specify the shared access signature token (SAS token) to the Storage resources, such as a blob or container.

Authentication

Authentication method *

Shared Access Signature (SAS)

SAS token *

.....

If you don't have a SAS token, switch to **Shared access signature** in your Azure Blob Storage account interface. Under **Allowed resource types**, select **Service**. Then select **Generate SAS and connection string**. You can get your SAS token from the **SAS token** that's displayed.

The shared access signature is a URI that encompasses in its query parameters all the information necessary for authenticated access to a storage resource. To access storage resources with the shared access signature, the client only needs to pass in the shared access signature to the appropriate constructor or method.

For more information about shared access signatures, go to [Shared access signatures: Understand the shared access signature model](#).

Service principal authentication

Authentication

Authentication method *

Service Principal

Tenant Id *

< your tenant ID >

Service principal ID *

< your application's client ID >

Service principal key *

.....

- **Tenant Id:** Specify the tenant information (domain name or tenant ID) under which your application resides. Retrieve it by hovering over the upper-right corner of the Azure portal.
- **Service principal ID:** Specify the application's client ID.
- **Service principal key:** Specify your application's key.

To use service principal authentication, follow these steps:

1. Register an application entity in Azure Active Directory (Azure AD) by following [Register your application with an Azure AD tenant](#). Make note of these values, which you use to define the connection:

- Tenant ID
- Application ID
- Application key

2. Grant the service principal proper permission in Azure Blob Storage. For more information on the roles, go to [Use the Azure portal to assign an Azure role for access to blob and queue data](#).

- As source, in **Access control (IAM)**, grant at least the **Storage Blob Data Reader** role.
- As destination, in **Access control (IAM)**, grant at least the **Storage Blob Data Contributor** role.

Step 3: Specify the privacy level that you want to apply

In the **General** tab, select the privacy level that you want apply in the **Privacy level** drop-down list. Three privacy levels are supported. For more information, go to [privacy levels](#).

Step 4: Create your connection

Select **Create**. Your creation will be successfully tested and saved if all the credentials are correct. If not correct, the creation will fail with errors.

Name ↑	Connection type	Users	Status	Gateway cluster name
azureblobstorage	Azure Blob Storage	<user name>		

Table summary

The following table contains connector properties that are supported in data pipeline creation.

Name	Description	Required	Property	Copy
Connection name	A name for your connection.	Yes		✓
Connection type	Select a type for your connection. Select Azure Blob Storage .	Yes		✓
Account	Azure Blob Storage account name.	Yes		✓
Domain	The domain of Azure Blob Storage: <code>blob.core.windows.net</code> .	Yes		✓
Authentication	Go to Authentication .	Yes		Go to Authentication .
Privacy Level	The privacy level that you want to apply. Allowed values are Organizational, Privacy, Public .	Yes		✓

Authentication

The properties in the following table are the supported authentication types.

Name	Description	Required	Property	Copy
Anonymous				✓
Key				✓
- Account key	The account key of the Azure Blob Storage.	Yes		
OAuth2				✓
Shared Access Signature (SAS)				✓
- SAS token	The shared access signature token to the Storage resources, such as a blob or container.	Yes		
Service Principal				✓
- Tenant ID	The tenant information (domain name or tenant ID).	Yes		

Name	Description	Required	Property	Copy
- Service Principal ID	The application's client ID.	Yes		
- Service Principal key	The application's key.	Yes		

Next steps

- [Copy data in Azure Blob Storage](#)
- [Connect to Azure Blob Storage in dataflows](#)

How to configure Azure Blob Storage in copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in data pipeline to copy data from and to Azure Blob Storage.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported format

Azure Blob Storage supports the following file formats. Refer to each article for format-based settings.

- Avro format
- [Binary format](#)
- [Delimited text format](#)
- [Excel format](#)
- JSON format
- ORC format
- [Parquet format](#)
- XML format

Supported configuration

For the configuration of each tab under copy activity, go to the following sections respectively.

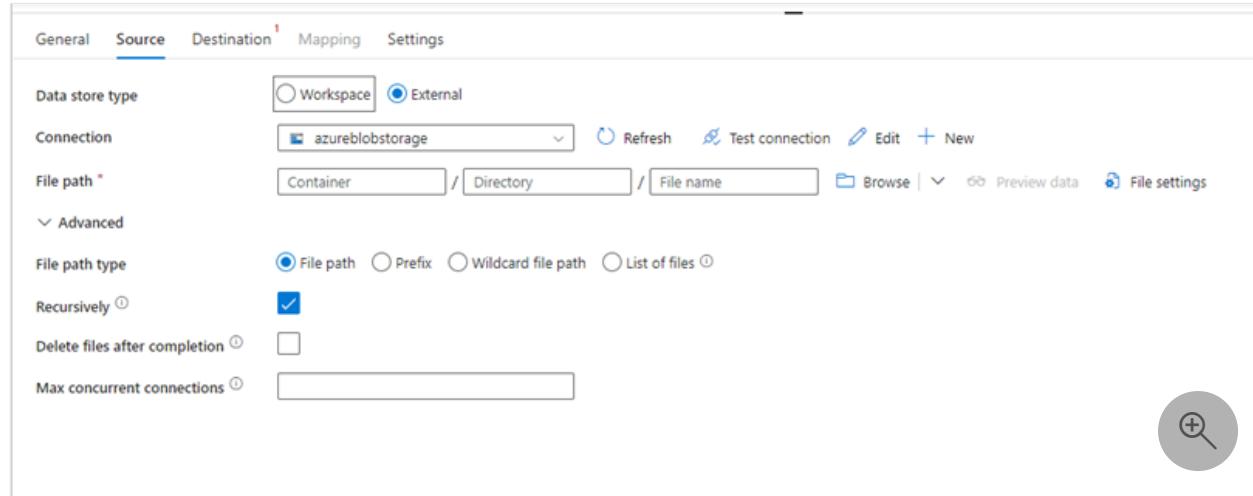
- [General](#)
- [Source](#)
- [Destination](#)
- [Mapping](#)
- [Settings](#)

General

Refer to the [General settings](#) guidance to configure the **General** settings tab.

Source

The following properties are supported for Azure Blob Storage under the **Source** tab of a copy activity.



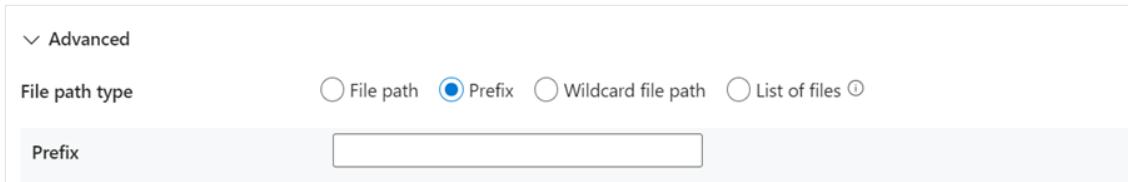
The following properties are **required**:

- **Data store type:** Select **External**.
- **Connection:** Select an Azure Blob Storage connection from the connection list. If no connection exists, then create a new Azure Blob Storage connection by selecting **New**.
- **File path:** Select **Browse** to choose the file that you want to copy, or fill in the path manually.
- **File settings:** Select **File settings** to configure the file format. For settings of different file formats, refer to articles in [Supported format](#) for detailed information.

Under **Advanced**, you can specify the following fields:

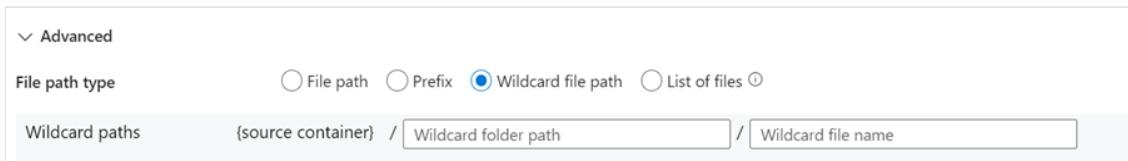
- **File path type:** You can choose **File path**, **Prefix**, **Wildcard file path**, **List of files** as your file path type. The configuration of each setting is :
 - **File path:** If you choose this type, the data can be copied from the given container or folder/file path specified previously.
 - **Prefix:** Prefix for the blob name under the given container configured to filter source blobs. Blobs whose names start with `container/this_prefix` are selected. It utilizes the service-side filter for blob storage.

When you use **Prefix** and choose to copy to a file-based destination with preserving hierarchy, the sub-path after the last "/" in the prefix is preserved. For example, you have a source `container/folder/subfolder/file.txt`, and configure the prefix as `folder/sub`, then the preserved file path is `subfolder/file.txt`.



- **Wildcard file path:** Specify the folder or file path with wildcard characters under your given blob container to filter your source folders or files.

Allowed wildcards are `*` (matches zero or more characters) and `?` (matches zero or single character). Use `\` to escape if your folder name has a wildcard or this escape character inside. For more examples, go to [Folder and file filter examples](#).



Wildcard folder path: Specify the folder path with wildcard characters under your given container to filter source folders.

Wildcard file name: Specify the file name with wildcard characters under your given container and folder path (or wildcard folder path) to filter source files.

- **List of files:** Indicates a given file set to copy to. In **Path to file list**, enter or browse to a text file that includes a list of files you want to copy, one file per line, which is the relative path to each file.

When you're using this option, don't specify a file name. For more examples, go to [File list examples](#).



- **Recursively:** If this checkbox is selected, all files in the input folder and its subfolders are processed recursively. If you unselect the checkbox, just the ones in

the selected folder are processed. This setting is disabled when a single file is selected.

- **Delete files after completion:** If this checkbox is selected, the binary files are deleted from source store after successfully moving to the destination store. The file deletion is per file, so when copy activity fails, you'll notice that some files have already been copied to the destination and deleted from the source, while others are still remaining in the source store.

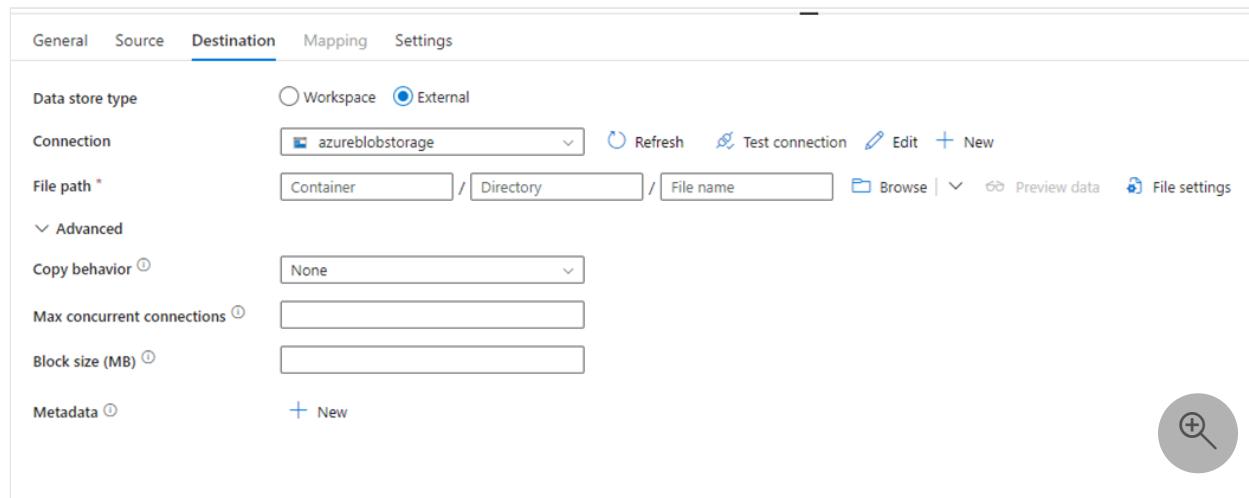
 **Note**

This property is only valid in a binary files copy scenario.

- **Max concurrent connections:** This property indicates the upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.

Destination

The following properties are supported for Azure Blob Storage under the **Destination** tab of a copy activity.



The screenshot shows the 'Destination' tab settings for a copy activity. The tabs at the top are General, Source, Destination (which is selected), Mapping, and Settings. Under 'Data store type', 'External' is selected. The 'Connection' dropdown shows 'azureblobstorage'. Below that, 'File path' is set to 'Container / Directory / File name'. There are sections for 'Advanced' (Copy behavior: 'None'), 'Max concurrent connections' (empty input field), 'Block size (MB)' (empty input field), and 'Metadata' (with a '+ New' button). A search icon is in the bottom right corner.

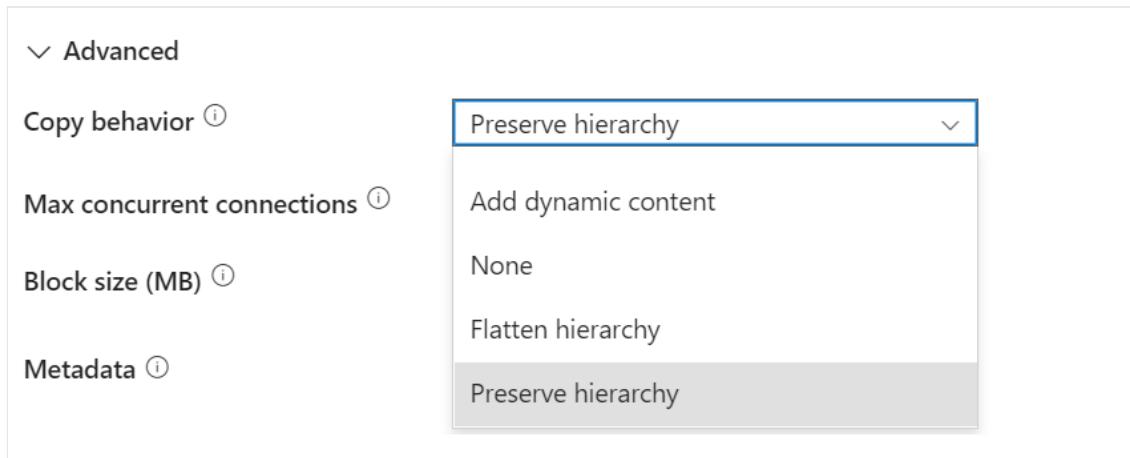
The following properties are **required**:

- **Data store type:** Select **External**.
- **Connection:** Select an Azure Blob Storage connection from the connection list. If the connection doesn't exist, then create a new Azure Blob Storage connection by selecting **New**.
- **File path:** Select **Browse** to choose the file that you want to copy or fill in the path manually.

- **File settings:** Select **File settings** to configure the file format. For settings of different file formats, refer to articles in [Supported format](#) for detailed information.

Under **Advanced**, you can specify the following fields:

- **Copy behavior:** Defines the copy behavior when the source is files from a file-based data store. You can choose **Add dynamic content**, **None**, **FlattenHierarchy**, or **Preserve hierarchy** from the drop-down list.
 - **Add dynamic content:** To specify an expression for a property value, select **Add dynamic content**. This selection opens the expression builder where you can build expressions from supported system variables, activity output, functions, and user-specified variables or parameters. For information about the expression language, go to [Expressions and functions](#).
 - **None:** Choose this selection to not use any copy behavior.
 - **Flatten hierarchy:** All files from the source folder are in the first level of the destination folder. The destination files have autogenerated names.
 - **Preserve hierarchy:** Preserves the file hierarchy in the target folder. The relative path of source file to source folder is identical to the relative path of target file to target folder.



- **Max concurrent connections:** The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.
- **Block size (MB):** Specify the block size, in megabytes, used to write data to block blobs. For more information, go to [About block blobs](#).
- **Metadata:** Set custom metadata when copying to a destination. Each object under the `metadata` array represents an extra column. The `name` defines the metadata key name, and the `value` indicates the data value of that key. If the `preserve attributes`

feature is used, the specified metadata will union/overwrite with the source file metadata.

Allowed data values are:

- **\$\$LASTMODIFIED**: a reserved variable indicates to store the source files' last modified time. Apply to file-based source with binary format only.
- **Expression**
- **Static value**



Mapping

For **Mapping** tab configuration, go to [Configure your mappings under mapping tab](#). If you choose Binary as your file format, mapping will not be supported.

Settings

For **Settings** tab configuration, see [Configure your other settings under settings tab](#).

Table summary

The following tables contain more information about the copy activity in Azure Blob Storage.

Source information

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the source data store.	<your connection>	Yes	connection
File path	The file path of your source data.	<file path of your source>	Yes	container fileName

Name	Description	Value	Required	JSON script property
File path type	The file path type that you want to use.	<ul style="list-style-type: none"> • File path • Prefix • Wildcard folder path, Wildcard file name • List of files 	No	<ul style="list-style-type: none"> • prefix • wildcardFolderPath, wildcardFileName • fileListPath
Recursively	Process all files in the input folder and its subfolders recursively or just the ones in the selected folder. This setting is disabled when a single file is selected.	Selected or unselect	No	recursive
Delete files after completion	The files in the source data store will be deleted right after being moved to the destination store. The file deletion is per file, so when a copy activity fails, you'll notice that some files have already been copied to the destination and deleted from source, while others are still in the source store.	Selected or unselect	No	deleteFilesAfterCompletion
Max concurrent connections	The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.	<max concurrent connections>	No	maxConcurrentConnections

Destination information

Name	Description	Value	Required	JSON script property
------	-------------	-------	----------	----------------------

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the destination data store.	<your connection>	Yes	connection
File path	The file path of your destination data.	File path of the source	Yes	container fileName
Copy behavior	Defines the behavior when copying files from one file system, like storage, to the other (for example, from one blob storage to another).	<ul style="list-style-type: none"> • None • Add dynamic content • Flatten hierarchy • Preserve hierarchy 	No	copyBehavior
Max concurrent connections	The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.	<max concurrent connections>	No	maxConcurrentConnections
Block size (MB)	Specify the block size in MB when writing data to Azure Blob Storage. Allowed value is between 4 MB and 100 MB.	<block size>	No	blockSizeInMB
Metadata	Set the custom metadata when copy to destination.	<ul style="list-style-type: none"> • <code>\$\$LASTMODIFIED</code> • Expression • Static value 	No	metadata

Next steps

- [How to create Azure Blob connection](#)
- [Connect to Azure Blob Storage in dataflows\]](#)

Connect to Azure Blob Storage in dataflows

Article • 05/23/2023

You can connect to Azure Blob Storage files in Dataflow Gen2 using the Azure Blob Storage connector provided by Data Factory in Microsoft Fabric.

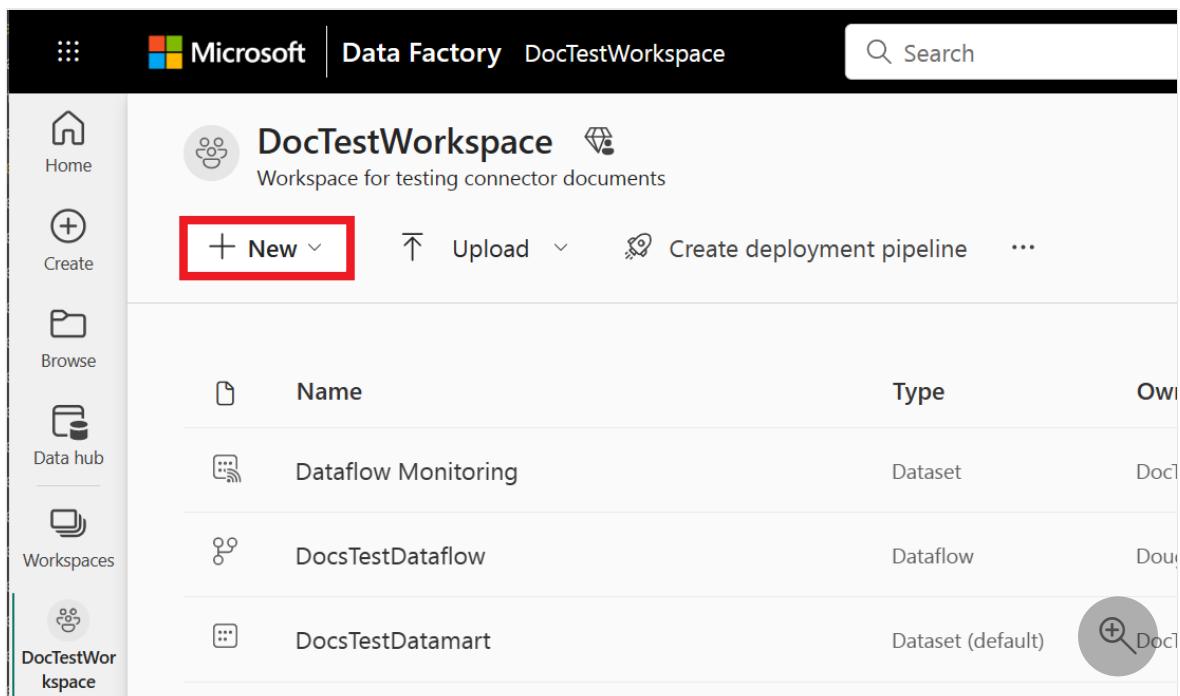
ⓘ Important

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Connect to Azure Blob Storage data

To connect to Azure Blob Storage from a dataflow:

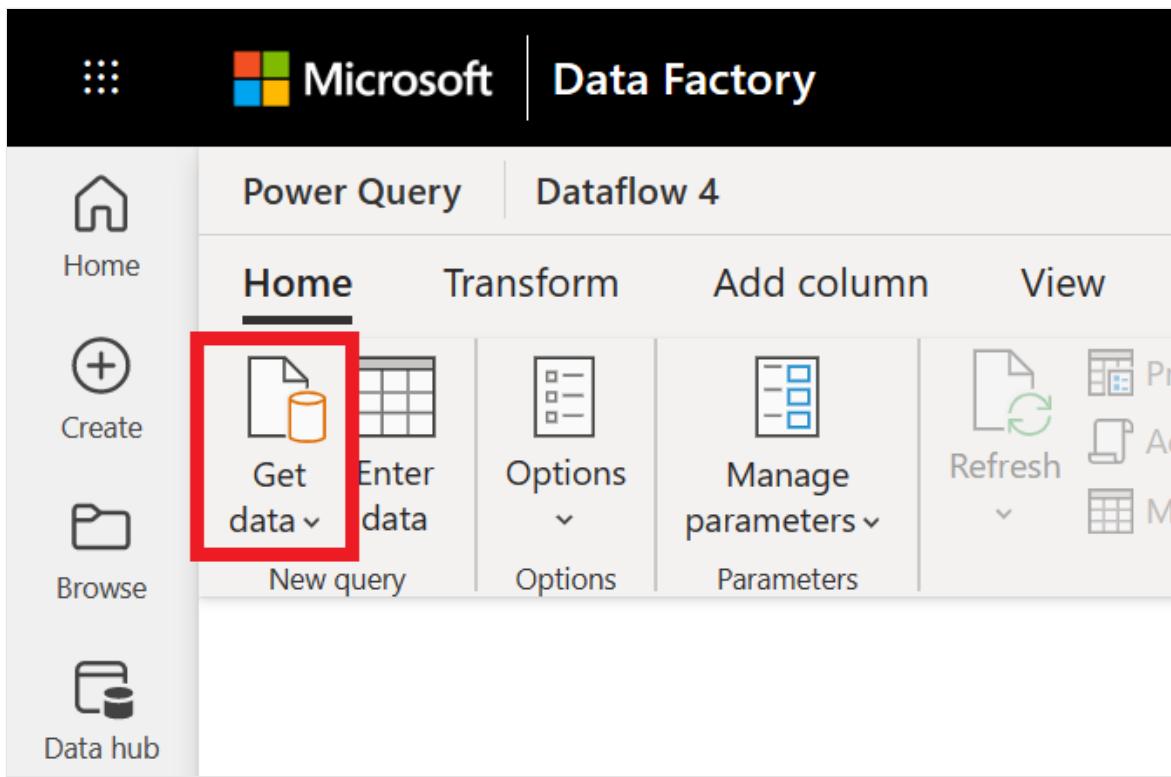
1. From your workspace, select **New > Dataflow Gen2 (Preview)** to create a new dataflow.



The screenshot shows the Microsoft Data Factory interface. At the top, there is a navigation bar with the Microsoft logo, the text "Data Factory DocTestWorkspace", and a search bar. On the left, a sidebar menu includes "Home", "Create" (which has a red box around its "New" button), "Browse", "Data hub", "Workspaces", and "DocTestWor kspace". The main area displays a workspace named "DocTestWorkspace" described as a "Workspace for testing connector documents". It features a "New" button with a dropdown arrow, an "Upload" button, a "Create deployment pipeline" button, and a three-dot menu. Below this, a table lists three items: "Dataflow Monitoring" (Dataset, Owner: DocT), "DocsTestDataflow" (Dataflow, Owner: DocT), and "DocsTestDatamart" (Dataset (default), Owner: DocT). A circular icon with a plus sign and a magnifying glass is visible in the bottom right corner of the main area.

	Name	Type	Owner
1	Dataflow Monitoring	Dataset	DocT
2	DocsTestDataflow	Dataflow	DocT
3	DocsTestDatamart	Dataset (default)	DocT

2. In Power Query, either select **Get data** in the ribbon or select **Get data from another source** in the current view.



3. From **Choose data source**, select the Azure category, and then select **Azure Blobs**.

A screenshot of the "Choose data source" dialog. At the top, it says "Choose data source" and "Select a connector or directly drag a file from your computer.". Below this, there is a navigation bar with tabs: "All categories", "File", "Database", "Power Platform", "Azure" (which is highlighted with a red box), "Online services", and "...". The main area displays a grid of data sources. The "Azure" category is shown in the second row. The "Azure Blobs" option is highlighted with a red box. Other options in the grid include "Azure SQL database", "Azure Synapse Analytics (SQL DW)", "Azure Tables", "Azure Data Lake Storage Gen2", "Azure HDInsight Spark", "Azure Cost Management", "Azure Databricks", "Azure Time Series Insights", and "HDInsight Interactive Query".

4. In **Connect to data source**, under **Connection settings**, enter your account name or the URL of your account.
5. If you're connecting to this data source for the first time, select the authentication type to use in **Authentication kind**, and then enter your credentials. The supported authentication types for this data source are:
 - Anonymous

- Account key
- Shared Access Signature (SAS)
- Organizational account

6. More information: [Connections and authentication in Power Query Online](#)

The screenshot shows the 'Connect to data source' dialog for 'Azure Blobs'. On the left, there's a section for 'Azure Blobs' with a 'Learn more' link. On the right, under 'Connection settings', there's a field for 'Account name or URL *' with a placeholder 'Example: testazurestorageacct'. Below it, under 'Connection credentials', there are dropdowns for 'Connection' (set to 'Create new connection'), 'Data gateway' (set to '(none)'), and 'Authentication kind' (set to 'Anonymous'). There are also refresh and search icons.

7. Select Next.

8. In **Choose data**, select the data item that you want to transform, and then select **Transform data**.

The screenshot shows the 'Choose data' dialog in Power Query. On the left, there's a tree view showing 'Azure Blobs' with a selected folder 'parquet-test-examples'. On the right, a preview grid shows a single file: 'Contoso Financial Sample.parquet'. The grid has columns for Content, Name, Extension, Date accessed, Date modified, Date created, and Attributes. At the bottom, there are 'Back', 'Cancel', and 'Transform data' buttons, along with a magnifying glass icon.

Content	Name	Extension	Date accessed	Date modified	Date created	Attributes
[Binary]	Contoso Financial Sample.parquet	.parquet	null	12/5/2022, 8:44:39 PM	null	[Record]

Advanced connector information

For more advanced information about connecting to your data source using the Azure Blob Storage connector, go to [Azure Blob Storage](#).

Next steps

- [How to create an Azure Blob connection](#)
- [How to configure Azure Blob Storage in copy activity](#)

Azure Cosmos DB for NoSQL connector overview

Article • 05/23/2023

The Azure Cosmos DB for NoSQL connector is supported in Data Factory in Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/destination)	None	Key
Lookup activity	None	Key

Next steps

- [How to create an Azure Cosmos DB for NoSQL connection](#)
- [How to configure Azure Cosmos DB for NoSQL in a copy activity](#)

How to create an Azure Cosmos DB for NoSQL connection

Article • 05/23/2023

This article outlines the steps to create an Azure Cosmos DB for NoSQL connection.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported authentication types

The Azure Cosmos DB for NoSQL connector supports the following authentication types for copy and Dataflow Gen2 respectively.

Authentication type	Copy	Dataflow Gen2
Account key	✓	n/a

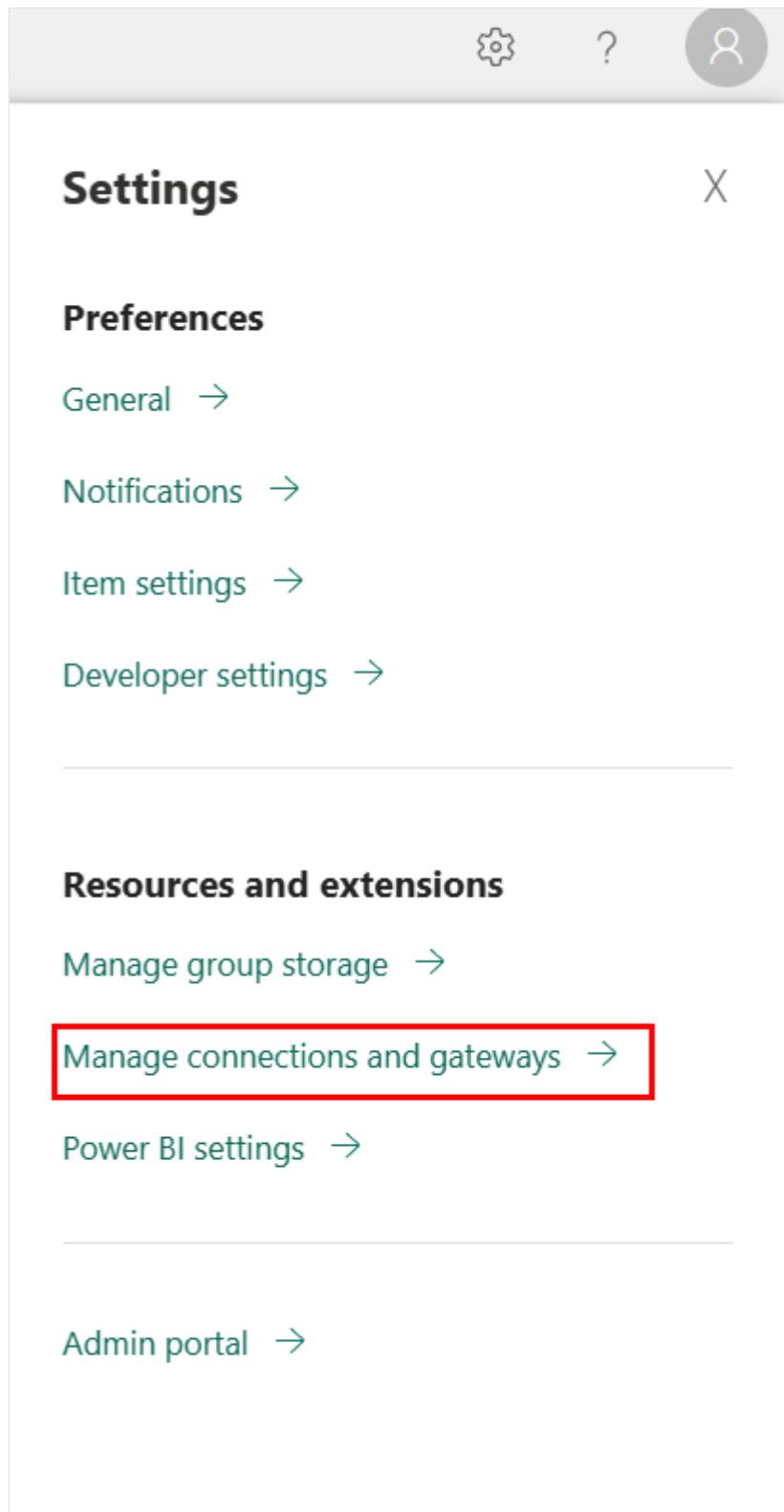
Prerequisites

The following prerequisites are required before you start:

- A Microsoft Fabric tenant account with an active subscription. [Create an account for free](#).
- A Microsoft Fabric enabled Workspace. [Create a workspace](#).

Go to manage gateways to create connection

1. From the page header in the Microsoft Fabric service, select **Settings**  > **Manage connections and gateways**.



2. Select **New** at the top of the ribbon to add a new data source.

The screenshot shows the 'Data (preview)' page. At the top left, there is a red box highlighting the '+ New' button. To its right is a 'Get help' link with a question mark icon. Below the header, there are three tabs: 'Connections' (underlined), 'On-premises data gateways', and 'Virtual network data gateways'. A search bar with a magnifying glass icon is located on the right. The main content area displays a message: 'Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)'

The New connection pane opens on the left side of the page.

The screenshot shows the 'New connection' pane open on the right side of the screen. The title is 'New connection'. It includes a note: 'Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".' There are three connection types: 'On-premises' (disabled), 'Virtual network' (disabled), and 'Cloud' (selected). The 'Cloud' section contains fields for 'Connection name *' and 'Connection type *'. At the bottom are 'Create' and 'Close' buttons, and a magnifying glass search icon.

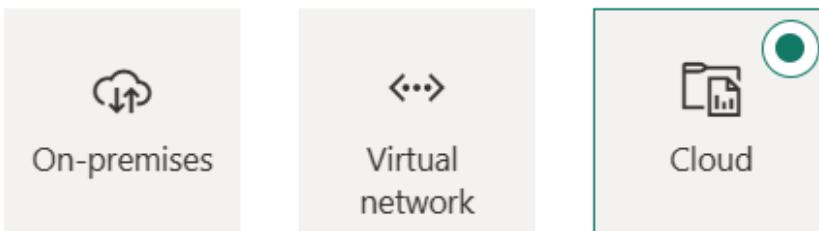
Setup connection

Step 1: Specify the new connection name, type, account endpoint and database

New connection



ⓘ Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



Connection name *

Connection type *



Account Endpoint *

Database *

In the **New connection** pane, choose **Cloud**, and specify the following field:

- **Connection name:** Specify a name for your connection.
- **Connection type:** Select **Azure CosmosDB (Data pipeline)** for your connection type.
- **Account Endpoint:** Enter your account endpoint URL of your Azure Cosmos DB for NoSQL.
- **Database:** Enter the database ID of your Azure Cosmos DB for NoSQL.

Step 2: Select and set your authentication

Under **Authentication method**, select your authentication method from the drop-down list and complete the related configuration. The Azure Cosmos DB for NoSQL connector supports the following authentication types:

- [Key](#)

Authentication

Authentication method *



The screenshot shows a dropdown menu with the word "Key" selected. A small downward arrow icon is located to the right of the selected item.

Key

Key authentication

Account key: Specify the account key of your Azure Cosmos DB for NoSQL connection. Go to your Azure Cosmos DB for NoSQL account interface, browse to the **Keys** section, and get your account key.

Authentication

Authentication method *



The screenshot shows a dropdown menu with the word "Key" selected. A small downward arrow icon is located to the right of the selected item.

Key

Account key *



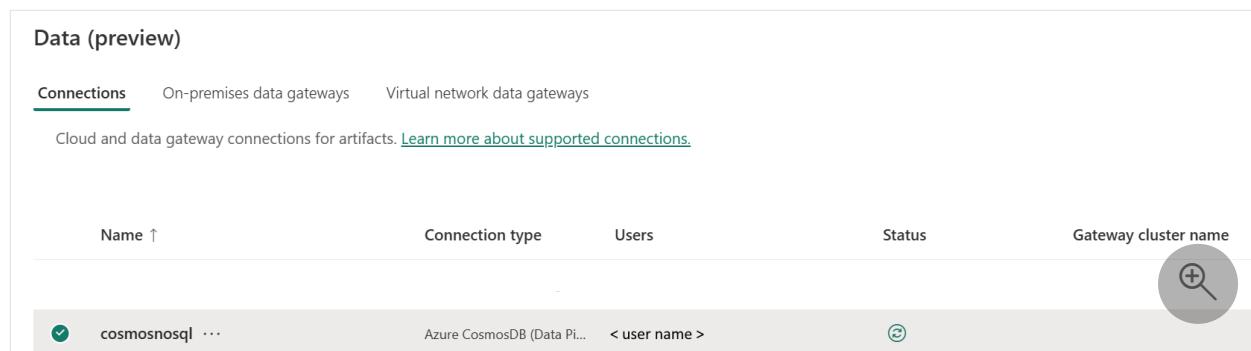
The screenshot shows a text input field containing a long sequence of redacted characters, represented by dots (...).

Step 3: Specify the privacy level that you want to apply

In the **General** tab, select the privacy level that you want apply in the **Privacy level** drop-down list. Three privacy levels are supported. For more information, see [privacy levels](#).

Step 4: Create your connection

Select **Create**. Your creation will be successfully tested and saved if all the credentials are correct. If not correct, the creation will fail with errors.



The screenshot shows the 'Data (preview)' section of the Azure portal. At the top, there are tabs for 'Connections', 'On-premises data gateways', and 'Virtual network data gateways'. Below the tabs, it says 'Cloud and data gateway connections for artifacts. [Learn more about supported connections](#)'. A table follows, with columns: 'Name ↑', 'Connection type', 'Users', 'Status', and 'Gateway cluster name'. There is one row visible with the name 'cosmosnosql ...', connection type 'Azure CosmosDB (Data Pi...', user '`< user name >`', status 'Active', and gateway cluster name 'cosmosnosql'. A search icon is in the top right of the table area.

Table summary

The connector properties in the following table are supported in pipeline copy:

Name	Description	Required	Property	Copy
Connection name	A name for your connection.	Yes		✓
Connection type	Select Azure CosmosDB (Data pipeline) for your connection type.	Yes		✓
Account Endpoint	Enter your Azure Cosmos DB for NoSQL account endpoint URL.	Yes		✓
Database	Enter the Azure Cosmos DB for NoSQL database ID.	Yes		✓
Authentication	Go to Authentication .	Yes	Go to Authentication .	Go to Authentication .
Privacy Level	The privacy level that you want to apply. Allowed values are Organizational , Privacy , and Public .	Yes		✓

Authentication

The properties in the following table are the supported authentication types.

Name	Description	Required	Property	Copy
Key				✓

Name	Description	Required	Property	Copy
- Account key	The Azure Cosmos DB for NoSQL account key.	Yes		

Next steps

- [How to configure Azure Cosmos DB for NoSQL in a copy activity](#)

How to configure Azure Cosmos DB for NoSQL in a copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in a data pipeline to copy data from and to Azure Cosmos DB for NoSQL.

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported configuration

For the configuration of each tab under the copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Destination](#)
- [Mapping](#)
- [Settings](#)

General

Refer to the [General settings](#) guidance to configure the **General** settings tab.

Source

The following properties are supported for Azure Cosmos DB for NoSQL under the **Source** tab of a copy activity.

Screenshot of the 'Source' tab configuration for an External data store type. The 'Connection' dropdown is set to 'cosmosnosql'. The 'Container' dropdown shows 'None' with an 'Edit' button. Under 'Advanced', 'Use query' is set to 'Table' (selected), and 'Query' is also available. A 'Page size' input field is empty. The 'Detect datetime' checkbox is checked. 'Preferred regions' and 'Additional columns' sections are present with 'New' buttons. A magnifying glass icon is in the bottom right corner.

The following three properties are required:

- **Data store type:** Select **External**.
- **Connection:** Select an Azure Cosmos DB for NoSQL connection from the connection list. If no connection exists, then create a new Azure Cosmos DB for NoSQL connection by selecting **New**.
- **Container:** Select the container that you want to use. Select **Edit** to enter the container name manually.

Under **Advanced**, you can specify the following fields:

- **Use query:** You can choose either **Table** or **Query** as your use query. The following list describes the configuration of each setting.
 - **Table:** Reads data from the table you specified in **Table**.
 - **Query:** Specifies the Azure Cosmos DB query to read data.

Screenshot showing the 'Advanced' configuration under 'Use query'. The 'Query' radio button is selected. The 'Query' input field contains 'select * from c' and has a pencil icon for editing.

- **Page size:** The number of documents per page of the query result. Default is "-1", which means to use the service side dynamic page size up to 1000.
- **Detect datetime:** Whether to detect datetime from the string values in the documents. Allowed values are: true (default), false.

- **Preferred regions:** The preferred list of regions to connect to when retrieving data from Azure Cosmos DB. Select one preferred region from the drop-down list after selecting **New**.

Preferred regions

+ New | Delete

<input type="checkbox"/>	

Add dynamic content [Alt+Shift+D]

- **Additional columns:** Add additional data columns to store source files' relative path or static value. Expression is supported for the latter. For more information, go to [Add additional columns during copy](#).

Additional columns ⓘ

+ New | Delete | Refresh

Name	Value
<input type="checkbox"/>	<input type="text"/> \$COLUMN

✖ Failed More

Destination

The following properties are supported for Azure Cosmos DB for NoSQL under the **Destination** tab of a copy activity.

General Source **Destination** Mapping Settings

Data store type Workspace External

Connection cosmosnosql Refresh Test connection Edit New

Container None Refresh Preview data Edit

Advanced

Write behavior Insert

Write batch timeout

Write batch size

Max concurrent connections

Disable performance metrics analytics

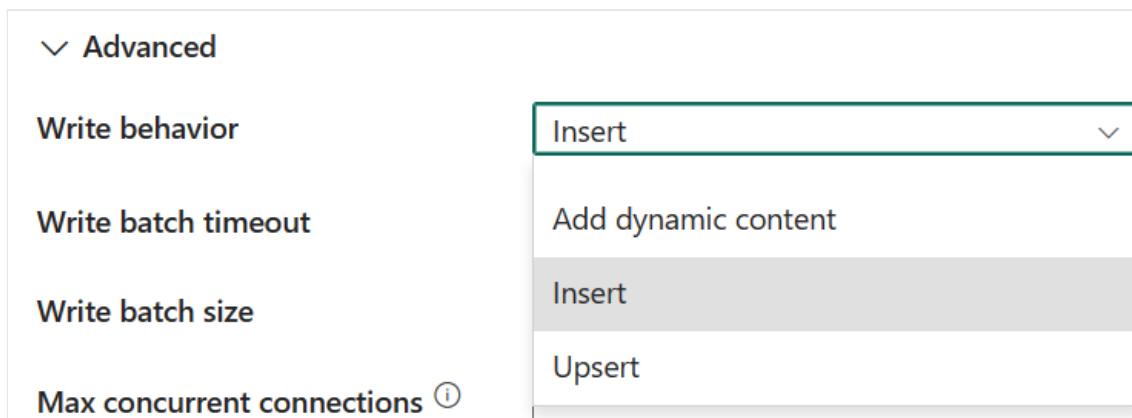
The following three properties are **required**:

- **Data store type:** Select **External**.
- **Connection:** Select an Azure Cosmos DB for NoSQL connection from the connection list.

- **Container:** Select **Browse** to choose the file that you want to copy, or fill in the path manually.

Under **Advanced**, you can specify the following fields:

- **Write behavior:** Defines the write behavior when the destination is files from a file-based data store. You can choose **Add dynamic content**, **Insert**, or **Upsert** from the drop-down list.
 - **Add dynamic content:** Open the **Add dynamic content** pane. This opens the expression builder where you can build expressions from supported system variables, activity output, functions, and user-specified variables or parameters. For information about the expression language, go to [Expressions and functions](#).
 - **Insert:** Choose this option if your source data has inserts.
 - **Upsert:** The behavior of upsert is to replace the document if a document with the same ID already exists; otherwise, insert the document.



- **Write batch timeout:** Wait time for the batch insert operation to complete before it times out. Allowed values are **Timespan**. An example is 00:30:00 (30 minutes).
- **Write batch size:** Specify the number of rows to insert into the SQL table per batch. The allowed value is an integer (number of rows). By default, the service dynamically determines the appropriate batch size based on the row size.
- **Max concurrent connections:** The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.
- **Disable performance metrics analytics:** This setting controls the collection of metrics, such as DTU, DWU, RU, and so on for copy performance optimization and recommendations. If you're concerned with this behavior, turn off this feature.

Mapping

For the **Mapping** tab configuration, go to [Configure your mappings under mapping tab.](#)

Settings

For the **Settings** tab configuration, go to [Configure your other settings under settings tab.](#)

Table summary

The following tables contain more information about the copy activity in Azure Cosmos DB for NoSQL.

Source table

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the source data store.	<your connection>	Yes	connection
Container	The container of your source data.	<container of your source>	Yes	container fileName
Use query	You can choose Table or Query as your use query.	<ul style="list-style-type: none">• Table• Query	No	type
Page size	The number of documents per page of the query result. Default is "-1", which means to use the service side dynamic page size up to 1000.	<your Page size>	No	pageSize
Delete datetime	The files on source data store will be deleted right after being moved to the destination store. The file deletion is per file, so when copy activity fails, you'll note that some files have already been copied to the destination and deleted from source while others are still on source store.	Selected or unselect	No	detectDatetime

Name	Description	Value	Required	JSON script property
Preferred regions	The preferred list of regions to connect to when retrieving data from Azure Cosmos DB. Select one preferred region from the drop-down list after selecting New.	<your preferred regions>	No	preferredRegions
Additional columns	Add additional data columns to store source files' relative path or static value. Expression is supported for the latter. For more information, go to Add additional columns during copy .	<max concurrent connections>	No	additionalColumns

Destination table

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the destination data store.	<your connection>	Yes	connection
Container	The container of your destination data.	<container of source>	Yes	container fileName
Write behavior	Describes how to write data to Azure Cosmos DB. Allowed values: insert and upsert. The behavior of upsert is to replace the document if a document with the same ID already exists; otherwise, insert the document.	<ul style="list-style-type: none"> • Add dynamic content • Insert • Upsert 	No	writeBehavior
Write batch timeout	Wait time for the batch insert operation to complete before it times out. Allowed values are Timespan. An example is 00:30:00 (30 minutes).	timespan	No	writeBatchTimeout

Name	Description	Value	Required	JSON script property
Write batch size	The number of rows to insert into the SQL table per batch. The allowed value is integer (number of rows). By default, the service dynamically determines the appropriate batch size based on the row size.	<number of rows > (integer)	No	writeBatchSize
Max concurrent connections	The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.	<max concurrent connections>	No	maxConcurrentConnections
Disable performance metrics analytics	This setting controls collection of metrics such as DTU, DWU, RU, and so on for copy performance optimization and recommendations. If you're concerned with this behavior, turn off this feature.	Selected or unselect	No	disableMetricsCollection

Next steps

- [How to create an Azure Cosmos DB for NoSQL connection](#)

Connect to Azure Cosmos DB for NoSQL in dataflows

Article • 05/23/2023

The Azure Cosmos DB for NoSQL connector isn't currently supported in Dataflow Gen2.

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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Next steps

- [How to create an Azure Cosmos DB for NoSQL connection](#)
- [How to configure Azure Cosmos DB for NoSQL in a copy activity](#)

Connect to Azure Data Explorer (Kusto) in dataflows

Article • 05/23/2023

You can connect to data that's either in an Azure Data Explorer (Kusto) cluster or in a Kusto Query Language (KQL) database in Dataflow Gen2 using the Azure Data Explorer (Kusto) connector provided by Data Factory in Microsoft Fabric.

ⓘ Important

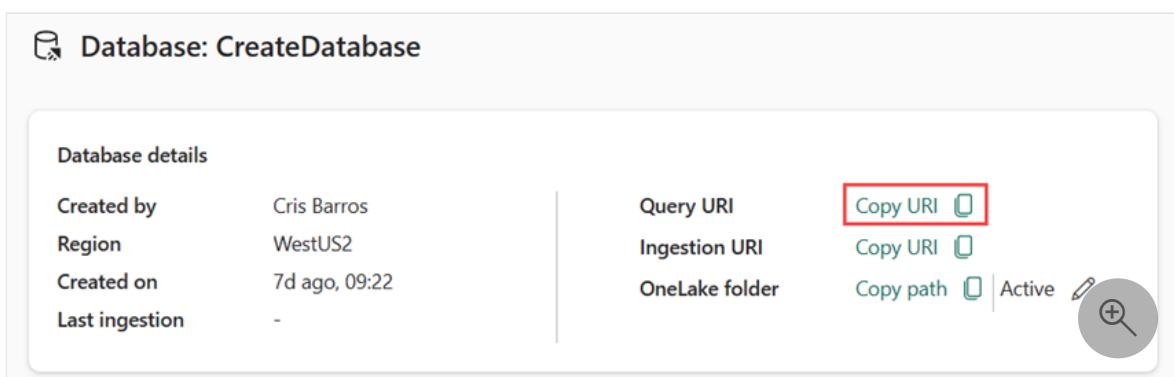
Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Copy the query URI

If you are using the Azure Data Explorer (Kusto) to connect to a KQL database, first copy your query URI using the following steps:

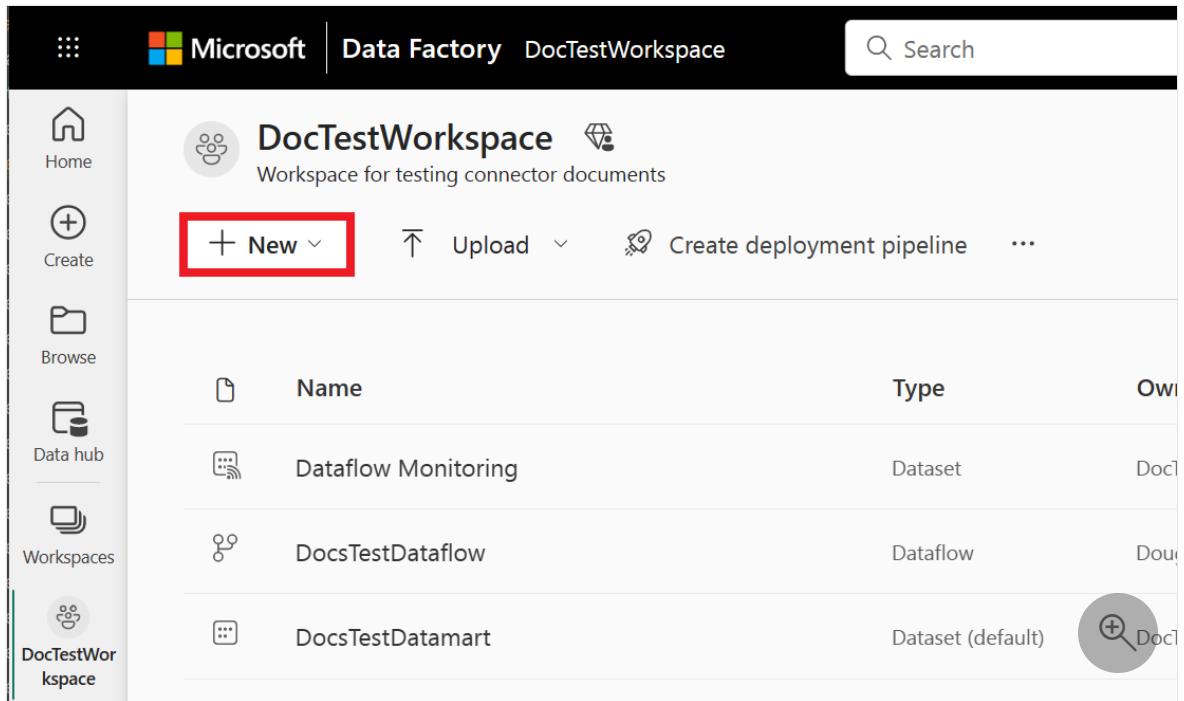
1. Navigate to your KQL database.
2. Copy the **Query URI** from the database details card in the database dashboard and paste it somewhere to use in a later step.



Connect to Azure Data Explorer (Kusto)

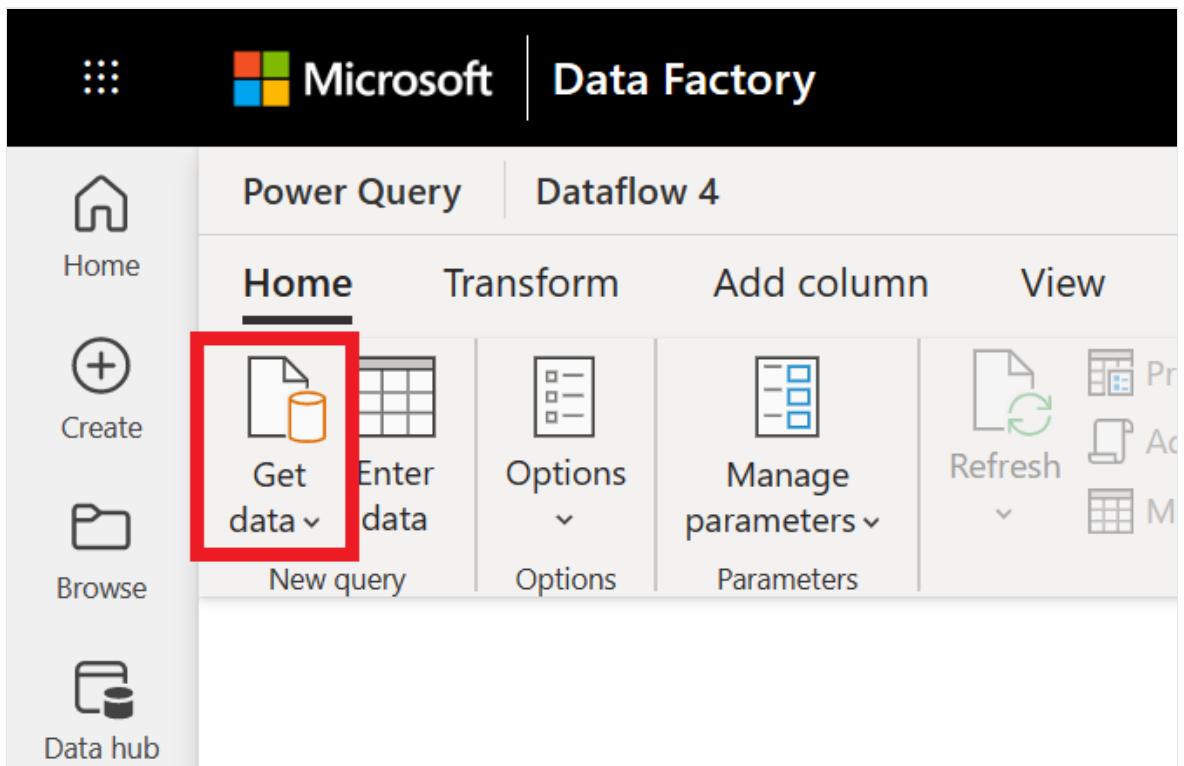
To connect to Azure Data Explorer from a dataflow:

1. From your workspace, select **New > Dataflow Gen2 (Preview)** to create a new dataflow.



The screenshot shows the Microsoft Data Factory interface within a browser window. The title bar reads "Microsoft Data Factory DocTestWorkspace". On the left, a sidebar menu includes "Home", "Create", "Browse", "Data hub", and "Workspaces" (with "DocTestWorkspace" selected). The main area displays a list of resources under "DocTestWorkspace": "Dataflow Monitoring" (Dataset), "DocsTestDataflow" (Dataflow), and "DocsTestDatamart" (Dataset (default)). A red box highlights the "+ New" button in the top navigation bar.

2. In Power Query, either select **Get data** in the ribbon or select **Get data from another source** in the current view.



The screenshot shows the Microsoft Power Query ribbon. The tabs include "Power Query" (selected), "Dataflow 4", "Home" (highlighted with a red box), "Transform", "Add column", and "View". The "Home" tab has several icons: "Get data" (highlighted with a red box), "Enter data", "Options", "Manage parameters", "Refresh", and "Parameters". Below the ribbon, there are buttons for "New query" and "Options". The left sidebar includes "Home", "Create", "Browse", and "Data hub".

3. From **Choose data source**, select the **Azure** category, and then select **Azure Data Explorer (Kusto)**.

Choose data source

Select a connector or directly drag a file from your computer.

All categories File Database Power Platform **Azure** Online services Other

Search

 Azure SQL database Azure	 Azure Synapse Analytics (S... Azure	 Azure Analysis Services Azure	 Azure Blobs Azure
 Azure Tables Azure	 Azure Data Explorer (Kusto) Azure	 Azure Data Lake Storage Ge... Azure	 Azure HDInsight Spark Azure
 Azure Cosmos DB v2 Azure	 Azure Cost Management Azure	 Azure Databricks Azure	 Azure Synapse Analyti... Azure
 Azure Time Series Insin... Azure	 HDInsight Interactive Query Azure		

Cancel

4. In **Connect to data source**, provide the Query URI from Microsoft Fabric's KQL Database dashboard that you copied in [Copy the query URI](#). For other clusters, the URL is in the form `https://_<ClusterName>_._<Region>_.kusto.windows.net`.

You can also select a database that's hosted on the cluster you're connecting to, and one of the tables in the database, or add a query like `StormEvents | take 1000`.

Connect to data source



Azure Data Explorer (Kusto)
Azure

3rd Party

Connection settings

Cluster *

Example: `https://mycluster.kusto.windows.net`

Database

Example: `MyDatabase`

Table name or Azure Data Explorer query

Example: `Logs | where [Timestamp] > ago(1h)`

> Advanced options

Connection credentials

Connection

Create new connection



Connection name

Connection



Data gateway

(none)



Authentication kind

Organizational account



5. If you want to use any advance options, select the option and enter the data to use with that option. More information: [Connect using advanced options](#)

6. If needed, select the on-premises data gateway in **Data gateway**.

7. Select the authentication type to use in **Authentication kind**, and then enter your credentials. More information: [Connections and authentication in Power Query Online](#)

8. Select **Next** to continue to the Power Query editor, where you can then begin to transform your data.

Choose data

Display options ▾

Azure Data Explorer (Kusto) [7]

ContosoSales

FindMyPartner

SampleIoTData

SampleLogs

SampleMetrics

Samples [63]

DailyCovid19

external_table(TaxiRid...)

ConferenceSessions

Covid19

Covid19_Bing

Covid19_map2

demo_clustering1

demo_make_series1

demo_make_series2

demo_many_series1

demo_prometheus

demo_series2

demo_series3

FHV_Trips

GeoRegions

Iris

irregular_ts

nyc_taxi

OccupancyDetection

PlotlyTemplate

PopulationData

SamplePowerRequire...

StormEvents

Back

Cancel

Create

StartTime	EndTime	EpisodeId	EventId	State	EventType	InjuriesDirect	InjuriesIndirect	DeathsDirect	DeathsIndirect	Damage
9/29/2007, 8:11:00 AM +00:00	9/29/2007, 8:11:00 AM +00:00	11091	61032	ATLANTIC SO...	Waterspout	0	0	0	0	0
9/18/2007, 8:00:00 PM +00:00	9/19/2007, 6:00:00 PM +00:00	11074	60904	FLORIDA	Heavy Rain	0	0	0	0	0
9/20/2007, 9:57:00 PM +00:00	9/20/2007, 10:05:00 PM +00:00	11078	60913	FLORIDA	Tornado	0	0	0	0	0
12/30/2007, 4:00:00 PM +00:00	12/30/2007, 4:05:00 PM +00:00	11749	64588	GEORGIA	Thunderstorm Wi...	0	0	0	0	0
12/20/2007, 7:50:00 AM +00:00	12/20/2007, 7:53:00 AM +00:00	12554	68796	MISSISSIPPI	Thunderstorm Wi...	0	0	0	0	0
12/20/2007, 10:32:00 AM +00:00	12/20/2007, 10:36:00 AM +00:00	12554	68814	MISSISSIPPI	Tornado	2	0	0	0	0
12/20/2007, 8:47:00 AM +00:00	12/20/2007, 8:48:00 AM +00:00	12554	68834	MISSISSIPPI	Thunderstorm Wi...	0	0	0	0	0
12/28/2007, 2:03:00 AM +00:00	12/28/2007, 2:11:00 AM +00:00	12561	68846	MISSISSIPPI	Hail	0	0	0	0	0
12/7/2007, 2:06:00 PM +00:00	12/8/2007, 4:00:00 AM +00:00	13183	73241	AMERICAN SA...	Flash Flood	0	0	0	0	0
12/13/2007, 9:02:00 AM +00:00	12/13/2007, 10:30:00 AM +00:00	11780	64725	KENTUCKY	Flood	0	0	0	0	0
12/23/2007, 6:02:00 AM +00:00	12/23/2007, 6:07:00 AM +00:00	11781	64726	OHIO	Thunderstorm Wi...	0	0	0	0	0
12/23/2007, 6:38:00 AM +00:00	12/23/2007, 6:43:00 AM +00:00	11781	64727	OHIO	Thunderstorm Wi...	0	0	0	0	0
12/23/2007, 6:36:00 AM +00:00	12/23/2007, 6:41:00 AM +00:00	11781	64728	OHIO	Thunderstorm Wi...	0	0	0	0	0
12/23/2007, 7:14:00 AM +00:00	12/23/2007, 7:19:00 AM +00:00	11781	64729	OHIO	Thunderstorm Wi...	0	0	0	0	0
12/11/2007, 9:45:00 PM +00:00	12/12/2007, 4:45:00 PM +00:00	12826	70787	KANSAS	Flood	0	0	0	0	0
12/28/2007, 2:47:00 AM +00:00	12/28/2007, 3:05:00 AM +00:00	12561	68867	MISSISSIPPI	Hail	0	0	0	0	0
12/28/2007, 3:05:00 AM +00:00	12/28/2007, 3:16:00 AM +00:00	12561	68868	MISSISSIPPI	Hail	0	0	0	0	0
12/10/2007, 1:10:00 PM +00:00	12/13/2007, 2:30:00 AM +00:00	12068	65995	KENTUCKY	Flood	0	0	0	0	0
12/15/2007, 1:00:00 PM +00:00	12/15/2007, 3:00:00 PM +00:00	11895	65282	KENTUCKY	Flood	0	0	0	0	0
12/15/2007, 12:00:00 PM +00:00	12/15/2007, 6:00:00 PM +00:00	11895	65283	KENTUCKY	Flood	0	0	0	0	0
12/28/2007, 3:30:00 AM +00:00	12/28/2007, 3:31:00 AM +00:00	12561	68871	MISSISSIPPI	Hail	0	0	0	0	0
12/16/2007, 2:30:00 AM +00:00	12/16/2007, 2:35:00 AM +00:00	11747	64586	FLORIDA	Thunderstorm Wi...	0	0	0	0	0
12/16/2007, 4:45:00 AM +00:00	12/16/2007, 4:50:00 AM +00:00	11747	64587	FLORIDA	Thunderstorm Wi...	0	0	0	0	0
12/23/2007, 5:00:00 AM +00:00	12/23/2007, 5:00:00 AM +00:00	12947	71552	MICHIGAN	Thunderstorm Wi...	0	0	0	0	0
12/15/2007, 6:00:00 PM +00:00	12/15/2007, 7:00:00 PM +00:00	12580	68889	ALABAMA	Heavy Rain	0	0	0	0	0
12/1/2007, 12:00:00 AM +00:00	12/1/2007, 1:00:00 AM +00:00	12558	68826	CALIFORNIA	Flash Flood	0	0	0	0	0
12/20/2007, 10:23:00 PM +00:00	12/20/2007, 10:23:00 PM +00:00	13011	72026	CALIFORNIA	Thunderstorm Wi...	0	0	0	0	0
12/1/2007, 5:30:00 AM +00:00	12/1/2007, 5:30:00 AM +00:00	13031	72052	NEW MEXICO	Thunderstorm Wi...	0	0	0	0	0
12/28/2007, 1:53:00 AM +00:00	12/28/2007, 2:00:00 AM +00:00	12561	68855	MISSISSIPPI	Hail	0	0	0	0	0
12/28/2007, 2:38:00 AM +00:00	12/28/2007, 2:49:00 AM +00:00	12561	68860	MISSISSIPPI	Hail	0	0	0	0	0
12/15/2007, 1:00:00 PM +00:00	12/15/2007, 4:00:00 PM +00:00	12692	69738	KENTUCKY	Flood	0	0	0	0	0
12/15/2007, 4:30:00 PM +00:00	12/15/2007, 7:30:00 PM +00:00	12692	69773	KENTUCKY	Flood	0	0	0	0	0
12/15/2007, 5:30:00 PM +00:00	12/15/2007, 8:30:00 PM +00:00	12692	69788	KENTUCKY	Flood	0	0	0	0	0
12/15/2007, 4:30:00 PM +00:00	12/15/2007, 8:30:00 PM +00:00	12692	69816	KENTUCKY	Flood	0	0	0	0	0
12/15/2007, 5:30:00 PM +00:00	12/15/2007, 7:30:00 PM +00:00	12692	69818	KENTUCKY	Flood	0	0	0	0	0

Advanced connector information

For more advanced information about connecting to your data using the Azure Data Explorer (Kusto) connector, go to [Azure Data Explorer \(Kusto\)](#).

Next steps

- How to configure KQL Database in a copy activity

Azure Data Lake Storage Gen2 connector overview

Article • 05/23/2023

The Azure Data Lake Storage Gen2 connector is supported in Data Factory for Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (Source/Destination)	None	Key OAuth2 Service principal Shared Access Signature (SAS)
Lookup activity	None	Key OAuth2 Service principal Shared Access Signature (SAS)
GetMetadata activity	None	Key OAuth2 Service principal Shared Access Signature (SAS)

Next steps

- [How to create a Azure Data Lake Storage Gen2 connection](#)
- [How to configure Azure Data Lake Storage Gen2 in copy activity](#)
- [Connect to Azure Data Lake Storage Gen2 in dataflows](#)

How to create an Azure Data Lake Storage Gen2 connection

Article • 05/23/2023

This article outlines the steps to create an Azure Date Lake Storage Gen2 connection.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported authentication types

The Azure Date Lake Storage Gen2 connector supports the following authentication types for copy and Dataflow Gen2 respectively.

Authentication type	Copy	Dataflow Gen2
Account key	✓	✓
Organizational account	✓	✓
Service Principal	✓	
Shared Access Signature (SAS)	✓	✓

ⓘ Note

For information about an Azure Data Lake Storage Gen2 connection in Dataflow Gen2, go to [Connect to Azure Data Lake Storage Gen2 in dataflows](#).

Prerequisites

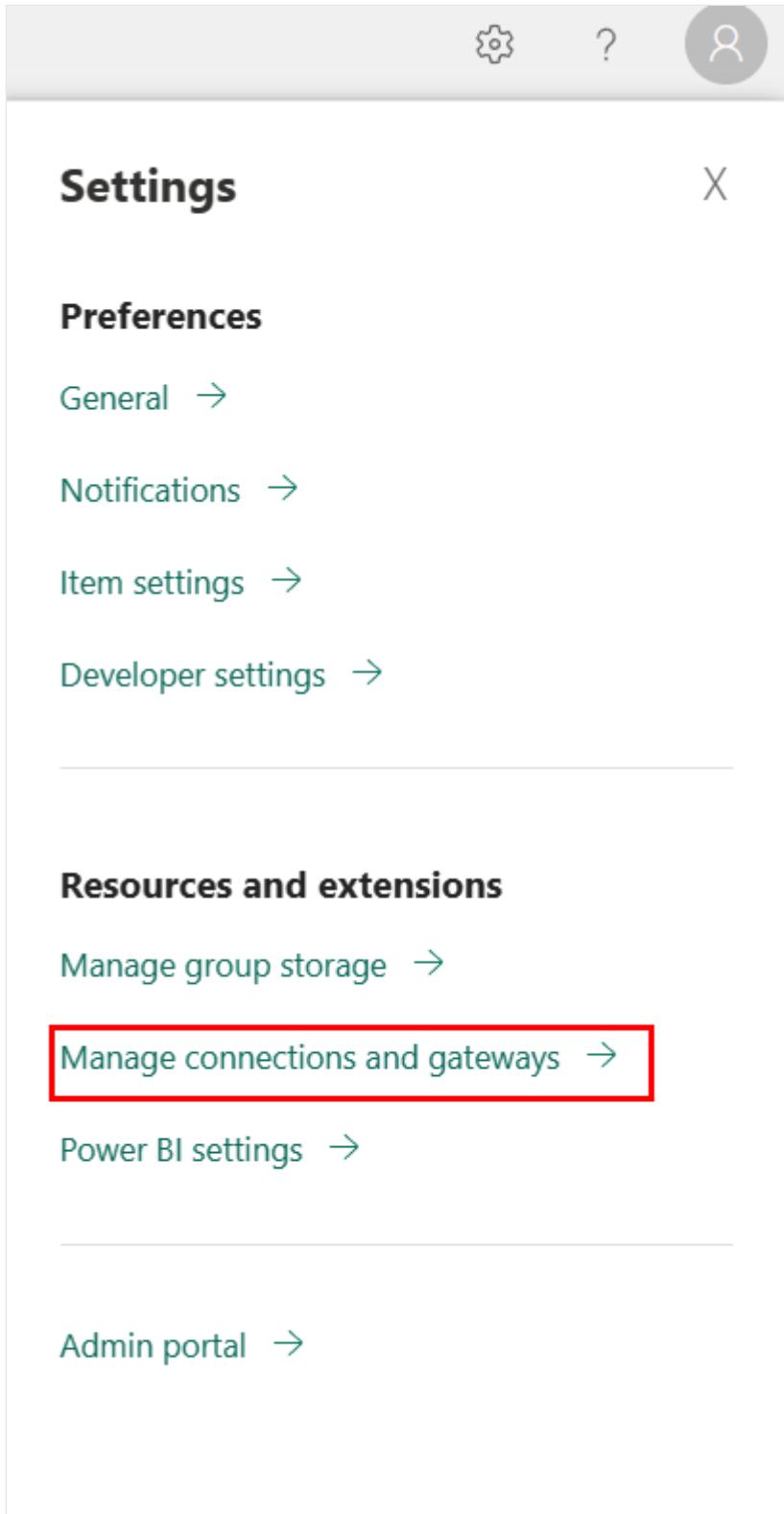
The following prerequisites are required before you start:

- A Microsoft Fabric tenant account with an active subscription. [Create an account for free](#).

- A Microsoft Fabric enabled Workspace. [Create a workspace.](#)

Go to Manage gateways to create connection

1. From the page header in Data Integration service, select **Settings** > **Manage connections and gateways**



2. Select **New** at the top of the ribbon to add a new data source.

[+ New](#)[Get help](#)

Data (preview)

Connections

[On-premises data gateways](#)[Virtual network data gateways](#)Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)

The New connection pane will show up on the left side of the page.

The screenshot shows the Power BI Data (preview) interface. On the left, there's a list of existing connections: "DemoBlob" (Azure Blob Storage, Clare). On the right, a "New connection" pane is open, titled "New connection". It includes a note about cloud connections being supported for Data Pipelines and Kusto. Below the note are three connection types: "On-premises" (selected), "Virtual network", and "Cloud" (highlighted with a green border). The "Connection name" field is empty, and the "Connection type" dropdown is also empty. At the bottom are "Create" and "Close" buttons, and a magnifying glass search icon.

Set up your connection

Step 1: Specify the new connection name, type, server and full path

New connection



ⓘ Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



Connection name *

azuredatalakestoragegen2

Connection type *

Azure Data Lake Storage Gen2

Server *

< Your Server name >

Full path *

< Your Container name >

In the **New connection** pane, choose **Cloud**, and specify the following fields:

- **Connection name:** Specify a name for your connection.
- **Connection type:** Select a type for your connection.
- **Server:** Enter your Azure Data Lake Storage Gen2 server name. For example, <https://contosoalscdm.dfs.core.windows.net>. Specify your Azure Data Lake Storage Gen2 server name. Go to your Azure Data Lake Storage Gen2 account

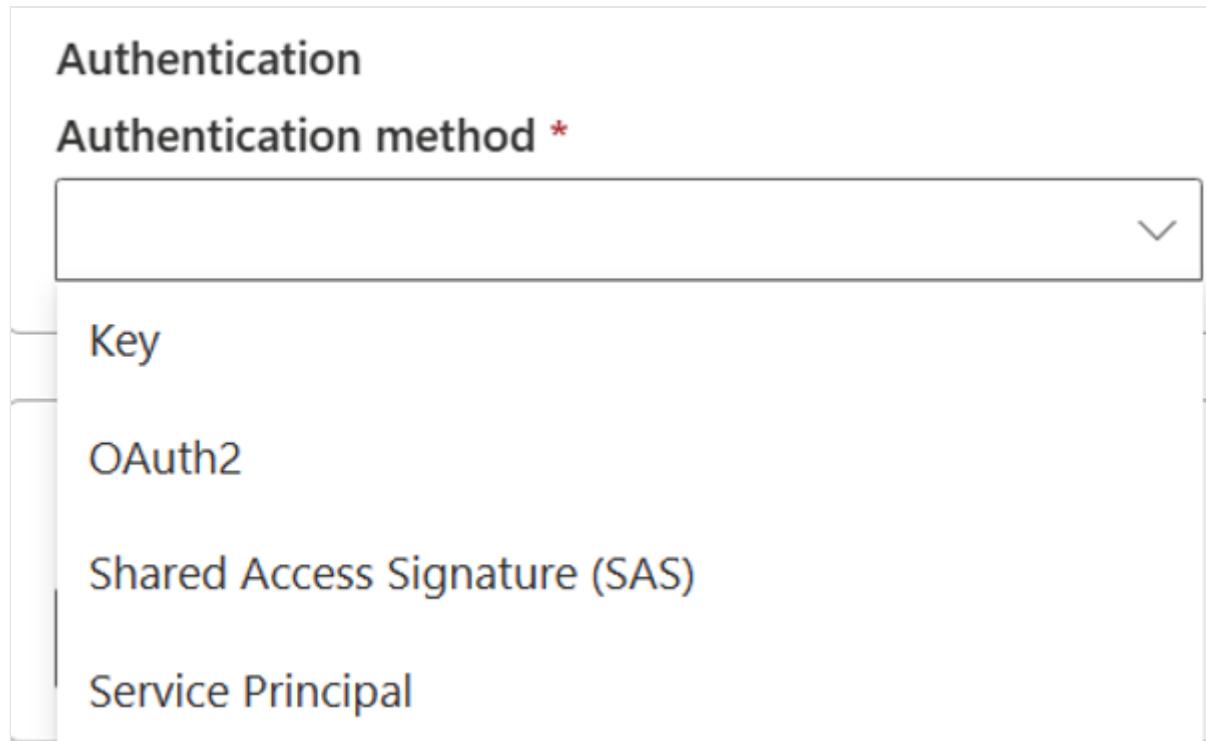
interface, browse to the **Endpoints** section, and get your Azure Data Lake Storage Gen2.

- **Full path:** Enter the full path to your Azure Data Lake Storage Gen2 container name.

Step 2: Select and set your authentication

Under **Authentication method**, select your authentication from the drop-down list and complete the related configuration. The Azure Data Lake Storage Gen2 connector supports the following authentication types:

- [Key](#)
- [OAuth2](#)
- [Shared Access Signature](#)
- [Service Principal](#)



Key authentication

Account key: Specify your Azure Data Lake Storage Gen2 account key. Go to your Azure Data Lake Storage Gen2 account interface, browse to the **Access key** section, and get your account key.

Authentication

Authentication method *

Key

Account key *

.....

OAuth2 authentication

Authentication

Authentication method *

OAuth2

[Edit credentials](#)

Open [Edit credentials](#). The sign-in interface opens. Enter your account and password to sign in to your account. After signing in, you'll come back to the **New connection** page.

Shared access signature authentication

Authentication

Authentication method *

Shared Access Signature (SAS)

SAS token *

.....

SAS token: Specify the shared access signature token for your Azure Data Lake Storage Gen2 container.

If you don't have a SAS token, switch to **Shared access signature** in your Azure Data Lake Storage Gen2 account interface. Under **Allowed resource types**, select **Container**, and then select **Generate SAS and connection string**. You can get your SAS token from the generated content that appears. The shared access signature is a URI that encompasses in its query parameters all the information necessary for authenticated access to a storage resource. To access storage resources with the shared access signature, the client only needs to pass in the shared access signature to the appropriate constructor or method. For more information about shared access signatures, go to [Shared access signatures: Understand the shared access signature model](#).

Service principal authentication

The screenshot shows the 'Authentication' section of a configuration page. It includes fields for 'Authentication method', 'Tenant Id', 'Service principal ID', and 'Service principal key'. The 'Authentication method' field is set to 'Service Principal'. The 'Tenant Id' field contains '< your tenant ID >'. The 'Service principal ID' field contains '< your application's client ID >'. The 'Service principal key' field contains a redacted password.

Authentication	
Authentication method *	Service Principal
Tenant Id *	< your tenant ID >
Service principal ID *	< your application's client ID >
Service principal key *	*****

- **Tenant Id:** Specify the tenant information (domain name or tenant ID) under which your application resides. Retrieve it by hovering over the upper-right corner of the Azure portal.
- **Service principal ID:** Specify the application (client) ID.
- **Service principal key:** Specify your application's key.

To use service principal authentication, follow these steps:

1. Register an application entity in Azure Active Directory (Azure AD) by following [Register your application with an Azure AD tenant](#). Make note of these values, which you use to define the connection:

- Tenant ID
- Application ID
- Application key

2. Grant the service principal proper permission. For examples of how permission works in Azure Data Lake Storage Gen2, go to [Access control lists on files and directories](#).

- **As source**, in Storage Explorer, grant at least **Execute** permission for all upstream folders and the file system, along with **Read** permission for the files to copy. Alternatively, in Access control (IAM), grant at least the **Storage Blob Data Reader** role.
- **As destination**, in Storage Explorer, grant at least **Execute** permission for all upstream folders and the file system, along with **Write** permission for the destination folder. Alternatively, in Access control (IAM), grant at least the **Storage Blob Data Contributor** role.

 **Note**

If you use a UI to author and the service principal isn't set with the "Storage Blob Data Reader/Contributor" role in IAM, when doing a test connection or browsing/navigating folders, choose **Test connection to file path** or **Browse from specified path**, and then specify a path with **Read + Execute** permission to continue.

Step 3: Specify the privacy level you want to apply

In the **General** tab, select the privacy level that you want apply in the **Privacy level** drop-down list. Three privacy levels are supported. For more information, go to General.

General

Privacy level *

Organizational



None

Private

Organizational

Public

Organizational



Step 4: Create your connection

Select **Create**. Your creation will be successfully tested and saved if all the credentials are correct. If not correct, the creation will fail with errors.

Data (preview)

Connections

On-premises data gateways

Virtual network data gateways

Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)

Name ↓

Connection type

Users

Status

Gateway cluster name

azuredatalakestoragegen2

Azure Data Lake Storage G... < user name >



Table summary

The connector properties in the following table are supported in pipeline copy.

Name	Description	Required	Property	Copy
Connection name	A name for your connection.	Yes		✓
Connection type	Select a type for your connection.	Yes		✓
Server	Enter the name of Azure Data Lake Storage Gen2 server, for example, <code>https://contosoадlscdm.dfs.core.windows.net</code> .	Yes		✓
Full path	Enter the full path of your Azure Data Lake Storage Gen2 container name.	Yes		✓
Authentication	Go to Authentication .	Yes	Go to Authentication .	
Privacy Level	The privacy level that you want to apply. Allowed values are Organizational, Privacy, and Public.	Yes		✓

Authentication

The properties in the following table are the supported authentication types.

Name	Description	Required	Property	Copy
Key				✓
- Account key	The Azure Data Lake Storage Gen2 account key.	Yes		
Shared Access Signature (SAS)				✓
- SAS token	Specify the shared access signature token for your Azure Data Lake Storage Gen2 container.	Yes		
Service Principal				✓
- Tenant ID	The tenant information (domain name or tenant ID).	Yes		
- Service Principal ID	The application's client ID.	Yes		
- Service Principal key	The application's key.	Yes		

Next steps

- How to configure Azure Data Lake Storage Gen2 in copy activity
- Connect to Azure Data Lake Storage Gen2 in dataflows

How to configure Azure Data Lake Storage Gen2 in copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in data pipeline to copy data from and to Azure Data Lake Storage Gen2.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported format

Azure Data Lake Storage Gen2 supports the following file formats. Refer to each article for format-based settings.

- Avro format
- [Binary format](#)
- [Delimited text format](#)
- [Excel format](#)
- JSON format
- ORC format
- [Parquet format](#)
- XML format

Supported configuration

For the configuration of each tab under copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Destination](#)
- [Mapping](#)
- [Settings](#)

General

Refer to the [General settings](#) guidance to configure the General settings tab.

Source

The following properties are supported for Azure Data Lake Storage Gen2 under the Source tab of a copy activity.

The screenshot shows the 'Source' tab selected in the top navigation bar. Under 'Data store type', 'External' is chosen. The 'Connection' dropdown is set to 'azuredatalakestoragegen2'. The 'File path' section includes fields for 'File system', 'Directory', and 'File name', along with 'Browse', 'Preview data', and 'File settings' buttons. An 'Advanced' section contains 'File path type' (set to 'File path'), 'Recursively' (checked), 'Delete files after completion' (unchecked), and 'Max concurrent connections' (empty input field). A magnifying glass icon is in the bottom right corner.

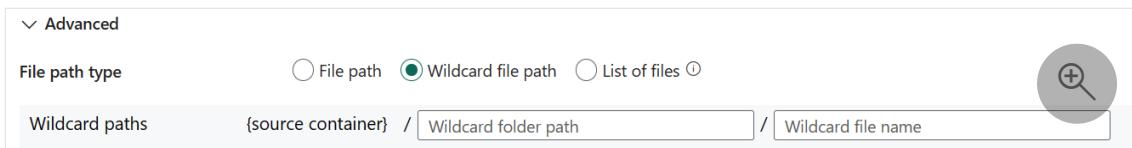
The following properties are required:

- **Data store type:** Select External.
- **Connection:** Select an Azure Data Lake Storage Gen2 connection from the connection list.
- **File path:** Select Browse to choose the file that you want to copy, or fill in the path manually.
- **File settings:** Select File settings to configure the file format. For settings of different file formats, refer to [Supported format](#) for detailed information.

Under Advanced, you can specify the following fields:

- **File path type:** You can choose File path, Wildcard file path, or List of files as your file path type. The configuration of each of these settings is :
 - **File path:** If you choose this type, the data can be copied from the given container or folder/file path specified previously.
 - **Wildcard file path:** Specify the folder or file path with wildcard characters under your given blob container to filter your source folders or files.

Allowed wildcards are: * (matches zero or more characters) and ? (matches zero or single character). Use ^ to escape if your folder name has wildcard or this escape character inside. For more examples, got to [Folder and file filter examples](#).

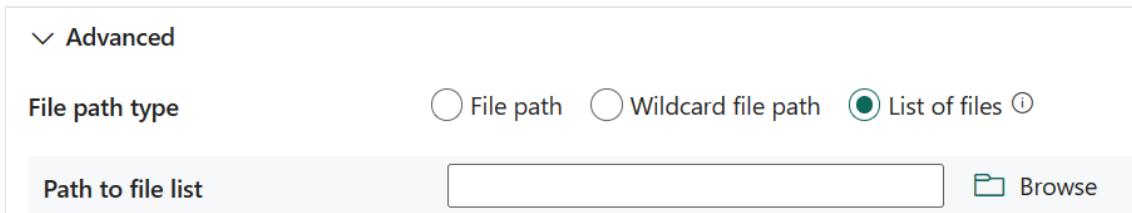


Wildcard folder path: Specify the folder path with wildcard characters under your given container to filter source folders.

Wildcard file name: Specify the file name with wildcard characters under your given container and folder path (or wildcard folder path) to filter source files.

- **List of files:** Indicates you want to copy a given file set. In **Path to file list**, point to a text file that includes a list of files you want to copy, one file per line, which is the relative path to the path.

When you're using this option, don't specify a file name. For more examples, go to [File list examples](#).



- **Recursively:** If this checkbox is selected, all files in the input folder and its subfolders will be processed recursively. If you unselect the checkbox, just the ones in the selected folder will be processed. This setting is disabled when a single file is selected.
- **Delete files after completion:** If this checkbox is selected, binary files will be deleted from the source store after successfully moving to the destination store. The file deletion is per file, so when copy activity fails, you'll see some files have already been copied to the destination and deleted from source, while others are still remaining in the source store.

Note

This property is only valid in a binary files copy scenario.

- **Max concurrent connections:** This property indicates the upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.

Destination

The following properties are supported for Azure Data Lake Storage Gen2 under the **Destination** tab of a copy activity.

General Source Destination Mapping Settings

Data store type Workspace External

Connection azuredatalakestoragegen2 Refresh Test connection Edit New

File path * File system / Directory / File name Browse Preview data File settings

Advanced

Copy behavior None

Max concurrent connections

Block size (MB)

Metadata New

The following properties are **required**:

- **Data store type:** Select **External**.
- **Connection:** Select an Azure Data Lake Storage Gen2 connection from the connection list.
- **File path:** Select **Browse** to choose the file that you want to copy or fill in the path manually.
- **File settings:** Select **File settings** to configure the file format. For settings of different file formats, refer to [Supported format](#) for detailed information.

Under **Advanced**, you can specify the following fields:

- **Copy behavior:** Defines the copy behavior when the source is files from a file-based data store. You can choose **Add dynamic content**, **None**, **Flatten hierarchy**, or **Preserve hierarchy** from the drop-down list.

Advanced

Copy behavior None

Max concurrent connections Add dynamic content

Block size (MB) None

Flatten hierarchy

Metadata Preserve hierarchy

- **Add dynamic content:** Open the **Add dynamic content** pane. This opens the expression builder where you can build expressions from supported system variables, activity output, functions, and user-specified variables or parameters.

For information about the expression language, go to [Expressions and functions](#).

- **None:** Choose this option to not use any copy behavior.
 - **Flatten hierarchy:** All files from the source folder are in the first level of the destination folder. The destination files have autogenerated names.
 - **Preserve hierarchy:** Preserves the file hierarchy in the target folder. The relative path of source file to source folder is identical to the relative path of target file to target folder.
- **Max concurrent connections:** The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.
 - **Block size (MB):** Specify the block size, in megabytes, used to write data to block blobs. More information: [Block Blobs](#)
 - **Metadata:** Set custom metadata when copying to a destination. Each object under the `metadata` array represents an extra column. The `name` defines the metadata key name, and the `value` indicates the data value of that key. If the [preserve attributes feature](#) is used, the specified metadata will union/overwrite with the source file metadata.

Allowed data values are:

- `$$LASTMODIFIED`: A reserved variable indicates to store the source files' last modified time. Apply to a file-based source with a binary format only.
- **Expression**
- **Static value**

Metadata ①		 New	 Delete
	Name	Value	
	<input type="text"/>	<input type="text"/> <code>\$\$LASTMODIFIED</code>	

Mapping

For **Mapping** tab configuration, go to [Configure your mappings under mapping tab](#). If you choose Binary as your file format, mapping will not be supported.

Settings

For the **Settings** tab configuration, go to [Configure your other settings under settings tab](#).

Table summary

The following tables contain more information about the copy activity in Azure Data Lake Storage Gen2.

Source

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	type
Connection	Your connection to the source data store.	<your connection>	Yes	connection
File path	The file path of your source data.	<file path of your source >	Yes	container fileName
File path type	The file path type that you want to use.	<ul style="list-style-type: none">• File path• Prefix• Wildcard folder path, Wildcard file name• List of files	No	<ul style="list-style-type: none">• prefix• wildcardFolderPath, wildcardFileName• fileListPath
Recursively	Process all files in the input folder and its subfolders recursively or just the ones in the selected folder. This setting is disabled when a single file is selected.	Selected or unselect	No	recursive

Name	Description	Value	Required	JSON script property
Delete files after completion	The files on source data store will be deleted right after being moved to the destination store. The file deletion is per file, so when copy activity fails, you'll see some files have already been copied to the destination and deleted from the source, while others are still in the source store.	Selected or unselect	No	deleteFilesAfterCompletion
Max concurrent connections	The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.	<max concurrent connections>	No	maxConcurrentConnections

Destination

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	type
Connection	Your connection to the destination data store.	<your connection>	Yes	connection
File path	The file path of your destination data.	File path of source	Yes	container fileName
Copy behavior	Defines the copy behavior when the source is files from a file-based data store.	<ul style="list-style-type: none"> • None • Add dynamic content • Flatten hierarchy • Preserve hierarchy 	No	copyBehavior

Name	Description	Value	Required	JSON script property
Max concurrent connections	The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.	<max concurrent connections>	No	maxConcurrentConnections
Block size (MB)	Specify the block size in MB when writing data to Azure Data Lake Storage Gen2. Allowed value is between 4 MB and 100 MB.	<block size>	No	blockSizeInMB
Metadata	Set custom metadata when copy to destination.	<ul style="list-style-type: none"> • <code>\$\$LASTMODIFIED</code> • Expression • Static value 	No	metadata

Next steps

- [How to create Azure Data Lake Storage Gen2 connection](#)
- [Connect to Azure Data Lake Storage Gen2 in dataflows](#)

Connect to Azure Data Lake Storage Gen2 in dataflows

Article • 05/23/2023

You can connect to Azure Data Lake Storage files in Dataflow Gen2 using the Azure Data Lake Storage Gen2 connector provided by Data Factory in Microsoft Fabric.

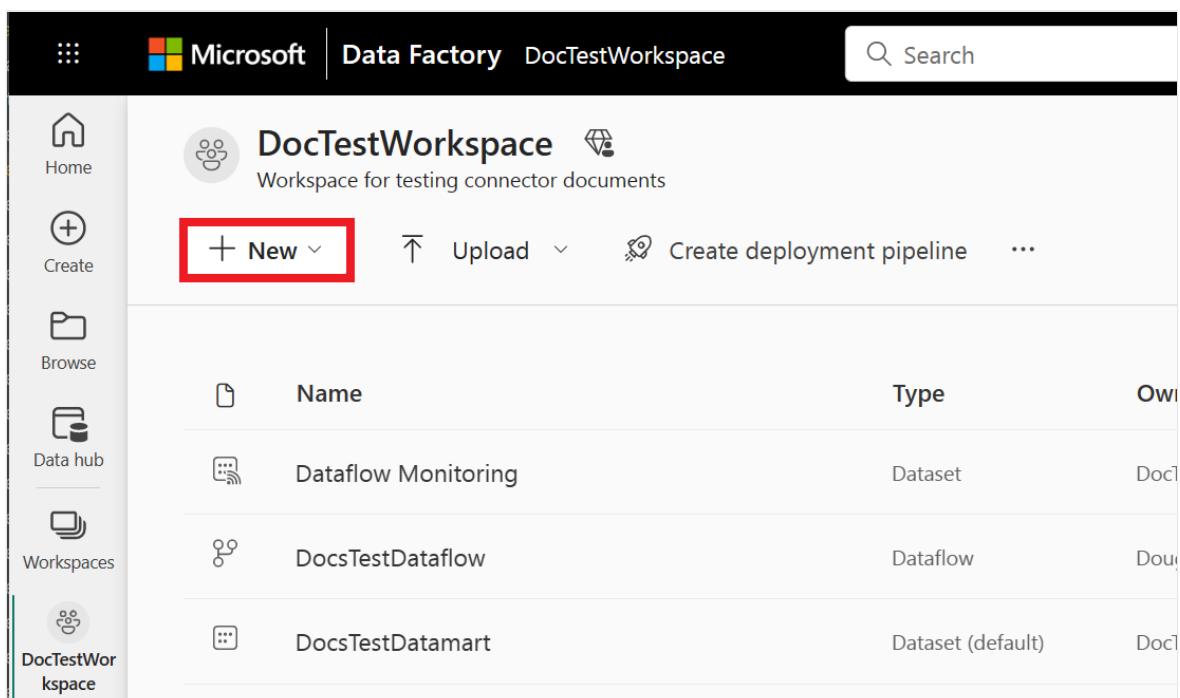
ⓘ Important

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Connect to Azure Data Lake Storage data

To connect to Azure Data Lake Storage Gen2 data from a dataflow:

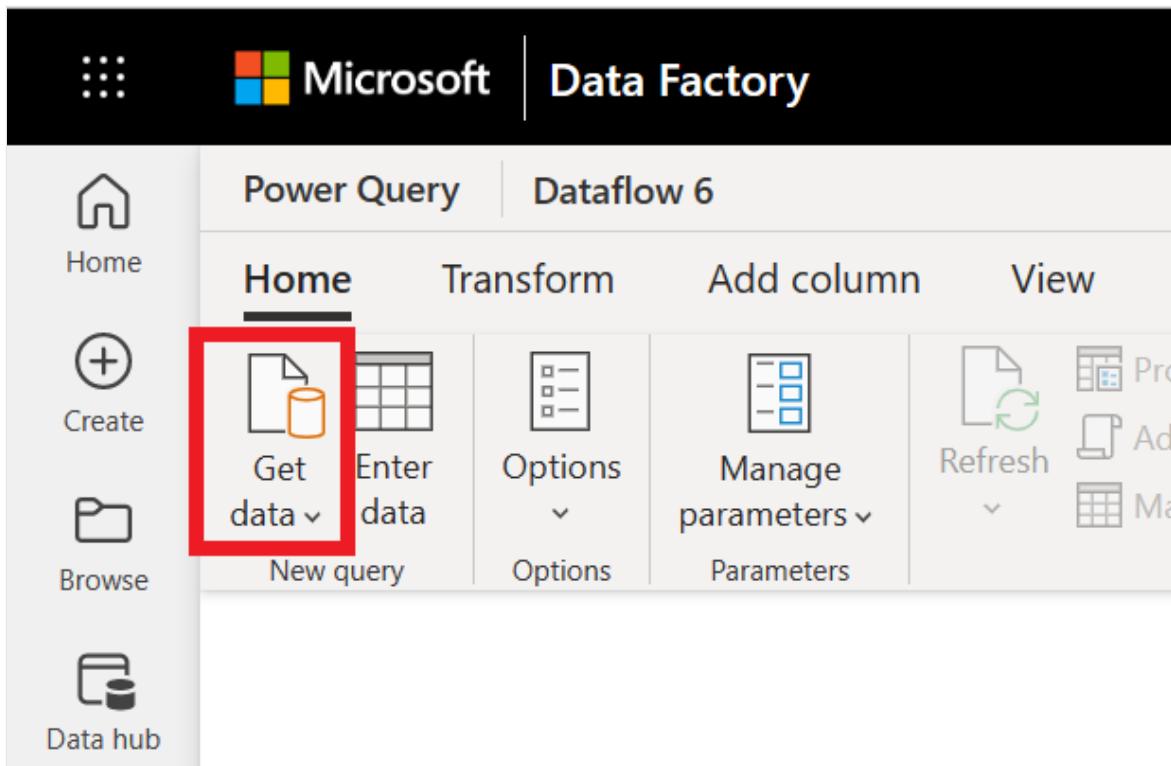
1. From your workspace, select **New > Dataflow Gen2 (Preview)** to create a new dataflow.



The screenshot shows the Microsoft Data Factory interface. At the top, there is a navigation bar with the Microsoft logo, the text "Data Factory DocTestWorkspace", and a search bar. On the left, a sidebar lists "Home", "Create" (with a plus icon), "Browse", "Data hub", "Workspaces", and "DocTestWor kspace" (which is selected and highlighted with a teal bar). The main area displays a workspace named "DocTestWorkspace" described as a "Workspace for testing connector documents". It features a "New" button with a red box around it, an "Upload" button, a "Create deployment pipeline" button, and a three-dot menu. Below this, a table lists three items: "Dataflow Monitoring" (Dataset, Owner: DocT), "DocsTestDataflow" (Dataflow, Owner: DocT), and "DocsTestDatamart" (Dataset (default), Owner: DocT).

	Name	Type	Owner
	Dataflow Monitoring	Dataset	DocT
	DocsTestDataflow	Dataflow	DocT
	DocsTestDatamart	Dataset (default)	DocT

2. In Power Query, either select **Get data** in the ribbon or select **Get data from another source** in the current view.



3. From **Choose data source**, select the **Azure** category, and then select **Azure Data Lake Storage Gen2**.

Choose data source

Select a connector or directly drag a file from your computer.

All categories File Database Power Platform **Azure** Online services ...

Azure SQL database Azure	Azure Synapse Analytics (SQL DW) Azure	Azure Data Lake Storage Gen2 Azure
Azure Blobs Azure	Azure Tables Azure	Azure HDInsight Spark Azure
Azure Cost Management Azure	Azure Databricks Azure	Unknown

4. In **Connect to data source**, under **Connection settings**, enter the URL of your account.
5. Select whether you want to use the file system view or the Common Data Model folder view.
6. If needed, select the on-premises data gateway in **Data gateway**.
7. Select the authentication type to use in **Authentication kind**, and then enter your credentials. More information: [Connections and authentication in Power Query Online](#)

Connect to data source

The screenshot shows the 'Connect to data source' dialog for the Azure Data Lake Storage Gen2 connector. On the left, there's a preview area showing a small grid icon and the text 'Azure Data Lake Storage Gen2' and 'Azure'. Below it is a 'Learn more' link. On the right, under 'Connection settings', there's a 'URL *' field with a placeholder 'Example: https://contosoadlscdm.dfs.core.windows.net/file...'. Under 'Data view', 'File system view' is selected. In the 'Connection credentials' section, there's a 'Connection' dropdown set to 'Create new connection', a 'Connection name' field containing 'Connection', a 'Data gateway' dropdown set to '(none)', and an 'Authentication kind' dropdown set to 'Organizational account'.

8. Select **Next**.

9. In **Choose data**, select the data item that you want to transform, and then select **Transform data**.

The screenshot shows the 'Power Query' dialog with the title 'Choose data'. On the left, there's a search bar and a 'Display options' dropdown. Below that is a tree view showing 'Azure Data Lake Storage Gen2 [1]' and a folder named 'testexamples'. On the right, there's a preview pane titled 'testexamples' showing a single file 'Drivers.txt'. The preview includes columns for Content, Name, Extension, Date accessed, Date modified, Date created, Attributes, and Folder Path. The file details are: Content [Binary], Name Drivers.txt, Extension .txt, Date accessed null, Date modified 4/29/2022, 8:58:19 PM, Date created null, Attributes [Record], and Folder Path https://contosoadlscdm.dfs.core.windows.net/testexamples/. At the bottom right are buttons for 'Cancel' and 'Transform data'.

Advanced connector information

For more advanced information about connecting to your data using the Azure Data Lake Storage Gen2 connector, go to [Azure Data Lake Storage Gen2](#).

Next steps

- How to create an Azure Data Lake Storage Gen2 connection
- How to configure Azure Data Lake Storage Gen2 in a copy activity

Azure SQL Database Connector Overview

Article • 05/23/2023

This Azure SQL Database connector is supported in Data Factory for Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (Source/Destination)	None	Basic OAuth2 Service principal
Lookup activity	None	Basic OAuth2 Service principal
GetMetadata activity	None	Basic OAuth2 Service principal
Script activity	None	Basic OAuth2 Service principal
Stored procedure activity	None	Basic OAuth2 Service principal

Next steps

- [How to create Azure SQL Database connection](#)

- How to configure Azure SQL Database in copy activity
- Connect to an Azure SQL database in dataflows

How to create an Azure SQL Database connection

Article • 05/23/2023

This article outlines how to set up connection to [Azure SQL Database](#).

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported authentication types

The Azure SQL Database connector supports the following authentication types for copy and Dataflow Gen2 respectively.

Authentication type	Copy	Dataflow Gen2
Basic	✓	✓
Organizational account	✓	✓
Service Principal	✓	

ⓘ Note

For information about an Azure SQL database connection in Dataflow Gen2, go to [Connect to an Azure SQL database in dataflows](#).

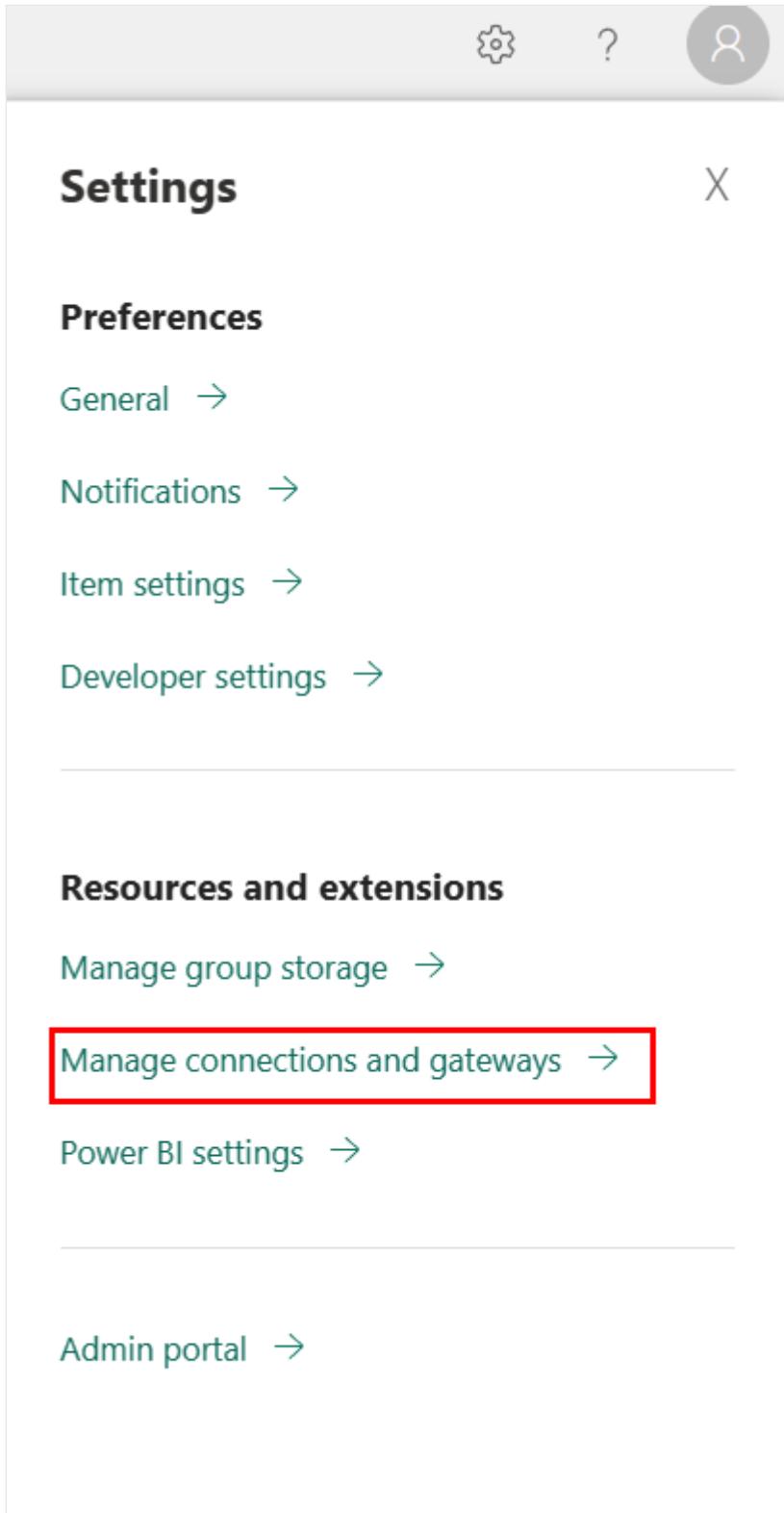
Prerequisites

The following prerequisites are required before you start:

- A Microsoft Fabric tenant account with an active subscription. [Create an account for free](#).
- A Microsoft Fabric enabled Workspace. [Create a workspace](#).

Go to Manage gateways to create a new connection

1. From the page header in Data Integration service, select **Settings**  > **Manage connections and gateways**.



2. Select **New** at the top of the ribbon to add a new data source.

[+ New](#)[Get help](#)

Data (preview)

Connections

[On-premises data gateways](#)[Virtual network data gateways](#)Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)

The New connection pane opens on the left side of the page.

The screenshot shows the 'Data (preview)' page with a 'New connection' pane open. The main area displays a table of existing connections:

Name ↑	Connection type	Users	Status
DemoBlob	Azure Blob Storage	Clare	Edit

The 'New connection' pane contains fields for 'Connection name *' and 'Connection type *'. It also includes tabs for 'On-premises', 'Virtual network', and 'Cloud' (which is selected). At the bottom are 'Create' and 'Close' buttons, and a magnifying glass search icon.

Set up your connection

Step 1: Specify the connection name, connection type, account, and domain

New connection

On-premises

Virtual network

Cloud

Connection name *

Connection type *

Server *

Database *

In the **New connection** pane, choose **Cloud**, and specify the following fields:

- **Connection name:** Specify a name for your connection.
- **Connection type:** Select **SQL Server**.
- **Server:** Enter your Azure SQL server name. You can find it in the **Overview** page of your Azure SQL server.
- **Database:** Enter your Azure SQL Database name.

Step 2: Select and set your authentication

Under **Authentication method**, select your authentication from the drop-down list and complete the related configuration. The Azure SQL Database connector supports the following authentication types.

- [Basic](#)
- [OAuth2](#)
- [Service Principal](#)

Authentication

Authentication method *

The screenshot shows a dropdown menu with three items: "Basic", "OAuth2", and "Service Principal". The "Basic" option is highlighted, indicating it is selected. The dropdown has a standard Windows-style arrow icon at the top right.

- Basic
- OAuth2
- Service Principal

Basic authentication

Select **Basic** under **Authentication method**.

Authentication

Authentication method *

The screenshot shows a form for basic authentication. It includes a dropdown menu set to "Basic", a field for "Username" containing "< your Azure SQL Database username >", and a field for "Password" containing a series of dots representing the password.

Basic

Username *

< your Azure SQL Database username >

Password *

.....

- **Username:** Specify the user name of your Azure SQL Database.
- **Password:** Specify the password of your Azure SQL Database.

OAuth2 authentication

Open **Edit credentials**. You'll see the sign in interface. Enter your account and password to sign in your account. After signing in, go back to the **New connection** page.

Authentication

Authentication method *

OAuth2

[Edit credentials](#)

Service Principal authentication

Authentication

Authentication method *

Service Principal

Tenant Id *

< your tenant ID >

Service principal ID *

< your application's client ID >

Service principal key *

.....

- **Tenant Id:** Specify the tenant information (domain name or tenant ID) under which your application resides. Retrieve it by hovering over the upper-right corner of the Azure portal.
- **Service principal ID:** Specify the application's client ID.
- **Service principal key:** Specify your application's key.

To use service principal authentication, follow these steps:

1. [Create an Azure Active Directory application](#) from the Azure portal. Make note of the application name and the following values that define the connection:
 - Tenant ID
 - Application ID
 - Application key

2. [Provision an Azure Active Directory administrator](#) for your server on the Azure portal if you haven't already done so. The Azure AD administrator must be an Azure AD user or Azure AD group, but it can't be a service principal. This step is done so that, in the next step, you can use an Azure AD identity to create a contained database user for the service principal.
3. [Create contained database users](#) for the service principal. Connect to the database from or to which you want to copy data by using tools like SQL Server Management Studio, with an Azure AD identity that has at least ALTER ANY USER permission. Sign in to your Azure SQL Database through Active Directory authentication and run the following T-SQL:

SQL

```
CREATE USER [your application name] FROM EXTERNAL PROVIDER;
```

4. Grant the service principal needed permissions as you normally do for SQL users or others. Run the following code. For more options, go to [ALTER ROLE \(Transact-SQL\)](#).

SQL

```
ALTER ROLE [role name] ADD MEMBER [your application name];
```

5. Configure an Azure SQL Database connection.

Step 3: Specify the privacy level that you want to apply

In the **General** tab, select the privacy level that you want apply in **Privacy level** drop-down list. Three privacy levels are supported. For more information, go to [privacy levels](#).

Step 4: Create your connection

Select **Create**. Your creation is successfully tested and saved if all the credentials are correct. If not correct, the creation fails with errors.

Data (preview)

Connections	On-premises data gateways	Virtual network data gateways		
Cloud and data gateway connections for artifacts. Learn more about supported connections.				
Name ↓	Connection type	Users	Status	Gateway cluster name
azuresqldatabase	SQL Server	< user name >		

Table summary

The following connector properties in the table are supported in pipeline copy.

Name	Description	Required	Property	Copy
Connection name	A name for your connection.	Yes		✓
Connection type	Select a type for your connection. Select SQL Server .	Yes		✓
Server	Azure SQL server name.	Yes		✓
Database	Azure SQL Database name.	Yes		✓
Authentication	Go to Authentication	Yes		Go to Authentication
Privacy Level	The privacy level that you want to apply. Allowed values are Organizational, Privacy, Public	Yes		✓

Authentication

The following properties in the table are the supported authentication types.

Name	Description	Required	Property	Copy
Basic				✓
- Username	The user name of your Azure SQL Database.	Yes		
- Password	The password of your Azure SQL Database.	Yes		
OAuth2				✓
Service Principal				✓

Name	Description	Required	Property	Copy
- Tenant ID	The tenant information (domain name or tenant ID).	Yes		
- Service Principal ID	The application's client ID.	Yes		
- Service Principal key	The application's key.	Yes		

Next steps

- [How to configure Azure SQL Database in copy activity](#)
- [Connect to an Azure SQL database in dataflows](#)

How to configure Azure SQL Database in copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in data pipeline to copy data from and to Azure SQL Database.

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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported configuration

For the configuration of each tab under copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Destination](#)
- [Mapping](#)
- [Settings](#)

General

Refer to the [General settings](#) guidance to configure the **General** settings tab.

Source

The following properties are supported for Azure SQL Database under the **Source** tab of a copy activity.

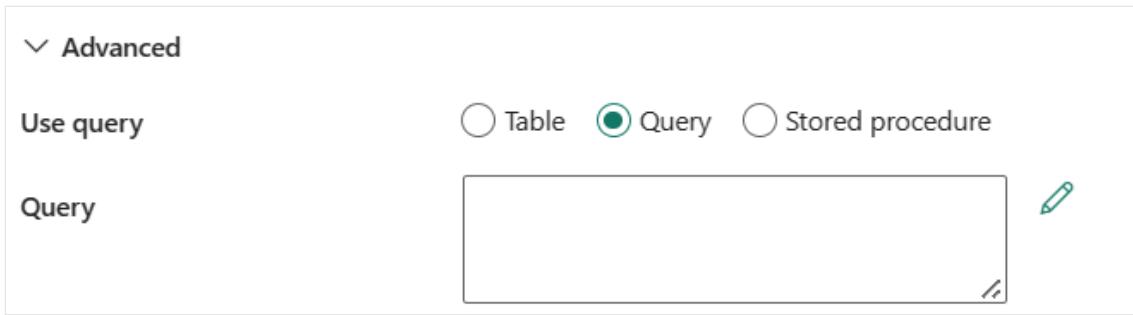
Screenshot of the 'Source' tab configuration in Azure Data Factory. The 'Data store type' is set to 'External'. The 'Connection' dropdown shows 'azuresqldatabase'. The 'Connection type' is 'Azure SQL Database'. The 'Table' dropdown shows 'SalesLT.Customer'. Below the table dropdown are 'Edit' and 'Advanced' buttons. Under 'Advanced', the 'Use query' option is selected ('Table'). The 'Query timeout (minutes)' is set to 120. The 'Isolation level' is 'Read committed'. The 'Partition option' is 'None'. A note says: 'Please preview data to validate the partition settings are correct before you trigger a run or publish the pipeline.' There is also a 'New' button and a magnifying glass icon.

The following properties are required:

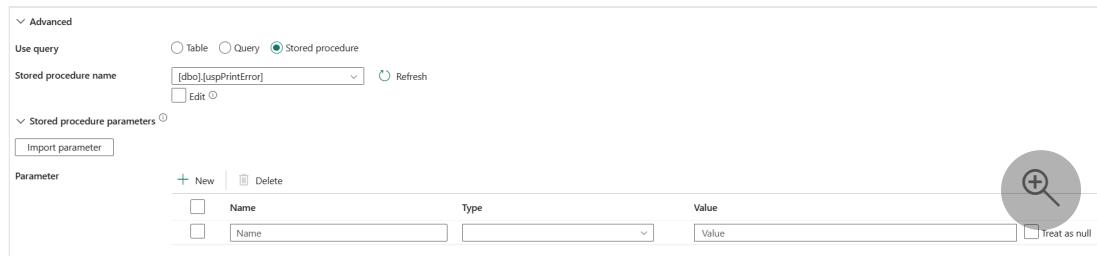
- **Data store type:** Select **External**.
- **Connection:** Select an Azure SQL Database connection from the connection list. If the connection doesn't exist, then create a new Azure SQL Database connection by selecting **New**.
- **Connection type:** Select **Azure SQL Database**.
- **Table:** Select the table in your database from the drop-down list. Or check **Edit** to enter your table name manually.
- **Preview data:** Select **Preview data** to preview the data in your table.

Under **Advanced**, you can specify the following fields:

- **Use query:** You can choose **Table**, **Query**, or **Stored procedure**. The following list describes the configuration of each setting :
 - **Table:** Read data from the table you specified in **Table** above if you select this button.
 - **Query:** Specify the custom SQL query to read data. An example is `select * from MyTable`. Or select the pencil icon to edit in code editor.



- **Stored procedure:** Use the stored procedure that reads data from the source table. The last SQL statement must be a SELECT statement in the stored procedure.
- **Stored procedure name:** Select the stored procedure or specify the stored procedure name manually when checking the **Edit** box to read data from the source table.
- **Stored procedure parameters:** Specify values for stored procedure parameters. Allowed values are name or value pairs. The names and casing of parameters must match the names and casing of the stored procedure parameters.



- **Query timeout (minutes):** Specify the timeout for query command execution, default is 120 minutes. If a parameter is set for this property, allowed values are timespan, such as "02:00:00" (120 minutes).

Query timeout (minutes) ⓘ	120
---------------------------	-----

- **Isolation level:** Specifies the transaction locking behavior for the SQL source. The allowed values are: **None**, **ReadCommitted**, **ReadUncommitted**, **RepeatableRead**, **Serializable**, or **Snapshot**. If not specified, **None** isolation level is used. Refer to [IsolationLevel Enum](#) for more details.

The screenshot shows the 'Isolation level' configuration section. A dropdown menu is open, listing several isolation levels: None, Read committed, Read uncommitted, Repeatable read, Serializable, and Snapshot. The 'None' option is currently selected. To the right of the dropdown, there is a note: 'Please preview data to validate the results' and a warning: 'Dynamic range partitioning is not supported for this operation'.

- **Partition option:** Specify the data partitioning options used to load data from Azure SQL Database. Allowed values are: **None** (default), **Physical partitions of table**, and **Dynamic range**. When a partition option is enabled (that is, not **None**), the degree of parallelism to concurrently load data from an Azure SQL Database is controlled by the [parallel copy](#) setting on the copy activity.

The screenshot shows the 'Partition option' configuration section. Three radio button options are available: 'None' (selected), 'Physical partitions of table', and 'Dynamic range'. The 'None' option is highlighted with a green circle.

- **None:** Choose this setting to not use a partition.
- **Physical partitions of table:** When using a physical partition, the partition column and mechanism are automatically determined based on your physical table definition.
- **Dynamic range:** When using query with parallel enabled, the range partition parameter(`?AdfDynamicRangePartitionCondition`) is needed. Sample query:
`SELECT * FROM <TableName> WHERE ?AdfDynamicRangePartitionCondition.`
- **Partition column name:** Specify the name of the source column in **integer or date/datetime** type (`int`, `smallint`, `bigint`, `date`, `smalldatetime`, `datetime`, `datetime2`, or `datetimeoffset`) that's used by range partitioning for parallel copy. If not specified, the index or the primary key of the table is autodetected and used as the partition column.
- **Partition upper bound:** Specify the maximum value of the partition column for partition range splitting. This value is used to decide the partition stride, not for filtering the rows in table. All rows in the table or query result are partitioned and copied.
- **Partition lower bound:** Specify the minimum value of the partition column for partition range splitting. This value is used to decide the partition stride,

not for filtering the rows in table. All rows in the table or query result are partitioned and copied.

- **Additional columns:** Add additional data columns to store source files' relative path or static value. Expression is supported for the latter. For more information, go to [Add additional columns during copy](#).

Destination

The following properties are supported for Azure SQL Database under the **Destination** tab of a copy activity.

General	Source	Destination	Mapping	Settings
		<input type="radio"/> Workspace <input checked="" type="radio"/> External		
Data store type				
Connection		<input type="button" value="azuresqldatabase"/>	<input type="button" value="Refresh"/> <input type="button" value="Edit"/> <input type="button" value="New"/>	
Connection type		<input type="button" value="Azure SQL Database"/>	<input type="button" value="Test connection"/>	
Table		<input type="button" value="SalesLT.Customer"/>	<input type="button" value="Refresh"/> <input type="button" value="Preview data"/>	<input type="checkbox"/> Edit
Advanced				
Write behavior		<input checked="" type="radio"/> Insert <input type="radio"/> Upsert <input type="radio"/> Stored procedure		
Bulk insert table lock <small>①</small>		<input checked="" type="radio"/> Yes <input type="radio"/> No		
Table option		<input checked="" type="radio"/> None <input type="radio"/> Auto create table <small>①</small>		
Pre-copy script <small>①</small>		<input type="text"/>		
Write batch timeout <small>①</small>		<input type="text" value="e.g. 00:30:00"/>		
Write batch size <small>①</small>		<input type="text"/>		
Max concurrent connections <small>①</small>		<input type="text"/>		
Disable performance metrics analytics <small>①</small>		<input type="checkbox"/>		

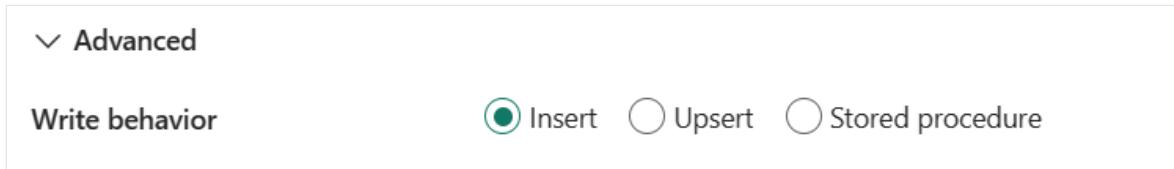
The following properties are **required**:

- **Data store type:** Select **External**.
- **Connection:** Select an Azure SQL Database connection from the connection list. If the connection doesn't exist, then create a new Azure SQL Database connection by selecting **New**.
- **Connection type:** Select **Azure SQL Database**.
- **Table:** Select the table in your database from the drop-down list. Or check **Edit** to enter your table name manually.

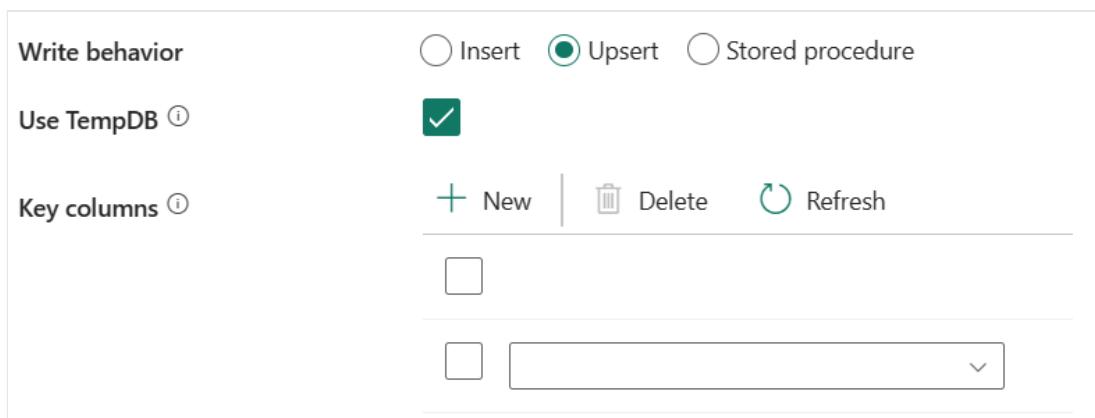
- **Preview data:** Select **Preview data** to preview the data in your table.

Under **Advanced**, you can specify the following fields:

- **Write behavior:** Defines the write behavior when the source is files from a file-based data store. You can choose **Insert**, **Upsert** or **Stored procedure**.



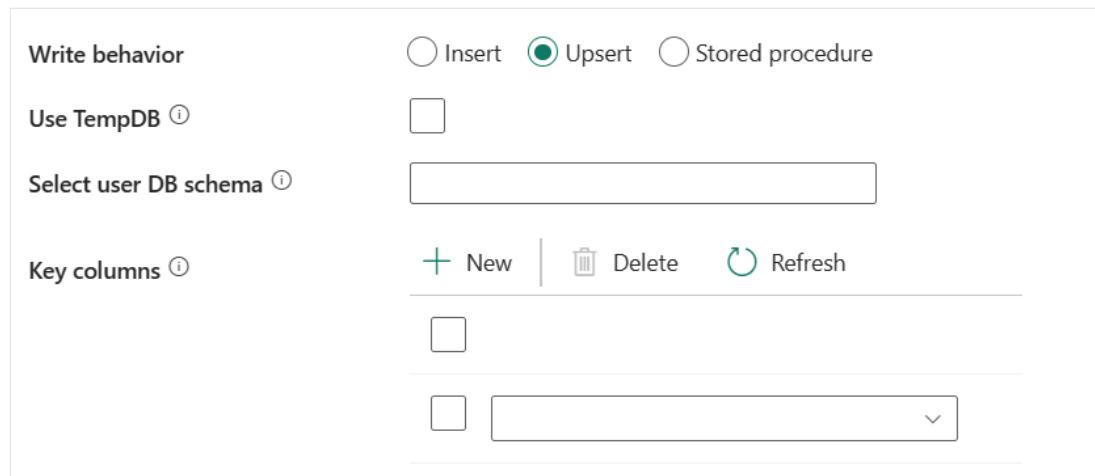
- **Insert:** Choose this option if your source data has inserts.
- **Upsert:** Choose this option if your source data has both inserts and updates.
 - **Use TempDB:** Specify whether to use a global temporary table or physical table as the interim table for upsert. By default, the service uses global temporary table as the interim table and this checkbox is selected.



- **Select user DB schema:** When the **Use TempDB** checkbox isn't selected, specify the interim schema for creating an interim table if a physical table is used.

Note

You must have the permission for creating and deleting tables. By default, an interim table will share the same schema as a destination table.



- **Key columns:** Specify the column names for unique row identification. Either a single key or a series of keys can be used. If not specified, the primary key is used.
- **Stored procedure:** Use the stored procedure that defines how to apply source data into a target table. This stored procedure is *invoked per batch*.
- **Stored procedure name:** Select the stored procedure or specify the stored procedure name manually when checking the **Edit** box to read data from the source table.
- **Stored procedure parameters:** Specify values for stored procedure parameters. Allowed values are name or value pairs. The names and casing of parameters must match the names and casing of the stored procedure parameters.



- **Bulk insert table lock:** Choose **Yes** or **No**. Use this setting to improve copy performance during a bulk insert operation on a table with no index from multiple clients. For more information, go to [BULK INSERT \(Transact-SQL\)](#)
- **Table option:** Specifies whether to [automatically create the destination table](#) if the table doesn't exist based on the source schema. Choose **None** or **Auto create table**. Auto table creation isn't supported when destination specifies a stored procedure.
- **Pre-copy script:** Specify a script for Copy Activity to execute before writing data into a destination table in each run. You can use this property to clean up the pre-

loaded data.

- **Write batch timeout:** Specify the wait time for the batch insert operation to finish before it times out. The allowed value is timespan. The default value is "00:30:00" (30 minutes).
- **Write batch size:** Specify the number of rows to insert into the SQL table per batch. The allowed value is integer (number of rows). By default, the service dynamically determines the appropriate batch size based on the row size.
- **Max concurrent connections:** Specify the upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.
- **Disable performance metrics analytics:** This setting is used to collect metrics, such as DTU, DWU, RU, and so on, for copy performance optimization and recommendations. If you're concerned with this behavior, select this checkbox.

Mapping

For **Mapping** tab configuration, go to [Configure your mappings under mapping tab](#).

Settings

For **Settings** tab configuration, go to [Configure your other settings under settings tab](#).

Table summary

The following tables contain more information about the copy activity in Azure SQL Database.

Source

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the source data store.	< your connection >	Yes	connection

Name	Description	Value	Required	JSON script property
Connection type	Your connection type. Select Azure SQL Database .	Azure SQL Database	Yes	/
Table	Your source data table.	< name of your destination table>	Yes	schema table
Use query	The custom SQL query to read data.	<ul style="list-style-type: none"> • None • Query • Stored procedure 	No	<ul style="list-style-type: none"> • sqlReaderQuery • sqlReaderStoredProcedureName, storedProcedureParameters
Query timeout	The timeout for query command execution, default is 120 minutes.	timespan	No	queryTimeout
Isolation level	Specifies the transaction locking behavior for the SQL source.	<ul style="list-style-type: none"> • None • ReadCommitted • ReadUncommitted • RepeatableRead • Serializable • Snapshot 	No	isolationLevel
Partition option	The data partitioning options used to load data from Azure SQL Database.	<ul style="list-style-type: none"> • None • Physical partitions of table • Dynamic range 	No	partitionOption: <ul style="list-style-type: none"> • PhysicalPartitionsOfTable • DynamicRange
Additional columns	Add additional data columns to store source files' relative path or static value. Expression is supported for the latter.	<ul style="list-style-type: none"> • Name • Value 	No	additionalColumns: <ul style="list-style-type: none"> • name • value

Destination

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the destination data store.	< your connection >	Yes	connection
Connection type	Your connection type. Select Azure SQL Database.	Azure SQL Database	Yes	/
Table	Your destination data table.	<name of your destination table>	Yes	schema table
Write behavior	Defines the write behavior when the source is files from a file-based data store.	<ul style="list-style-type: none"> • Insert • Upsert • Stored procedure 	No	writeBehavior: <ul style="list-style-type: none"> • insert • upsert • sqlWriterStoredProcName, sqlWriterTableType, storedProcedureParameters
Bulk insert table lock	Use this setting to improve copy performance during a bulk insert operation on a table with no index from multiple clients.	Yes or No	No	sqlWriterUseTableLock: true or false
Table option	Specifies whether to automatically create the destination table if it doesn't exist based on the source schema.	<ul style="list-style-type: none"> • None • Auto create table	No	tableOption: <ul style="list-style-type: none"> • autoCreate

Name	Description	Value	Required	JSON script property
Pre-copy script	A script for Copy Activity to execute before writing data into a destination table in each run. You can use this property to clean up the pre-loaded data.	< pre-copy script > (string)	No	preCopyScript
Write batch timeout	The wait time for the batch insert operation to finish before it times out. The allowed value is timespan. The default value is "00:30:00" (30 minutes).	timespan	No	writeBatchTimeout
Write batch size	The number of rows to insert into the SQL table per batch. By default, the service dynamically determines the appropriate batch size based on the row size.	< number of rows > (integer)	No	writeBatchSize
Max concurrent connections	The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.	< upper limit of concurrent connections > (integer)	No	maxConcurrentConnections

Name	Description	Value	Required	JSON script property
Disable performance metrics analytics	This setting is used to collect metrics, such as DTU, DWU, RU, and so on, for copy performance optimization and recommendations. If you're concerned with this behavior, select this checkbox.	select or unselect	No	disableMetricsCollection : true or false

Next steps

- [How to create Azure SQL Database connection](#)
- [Connect to an Azure SQL database in dataflows](#)

Connect to an Azure SQL database in dataflows

Article • 05/23/2023

You can connect to Azure SQL databases in Dataflow Gen2 using the Azure SQL database connector provided by Data Factory in Microsoft Fabric.

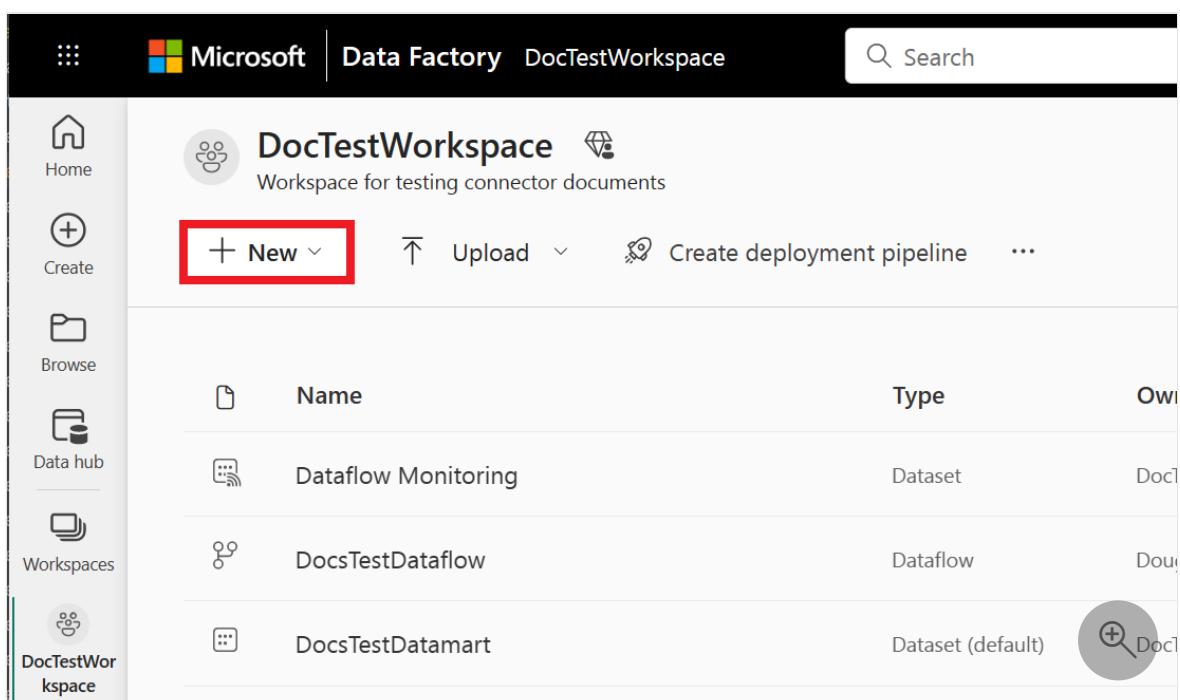
ⓘ Important

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Connect to an Azure SQL database

To connect to an Azure SQL database from a dataflow:

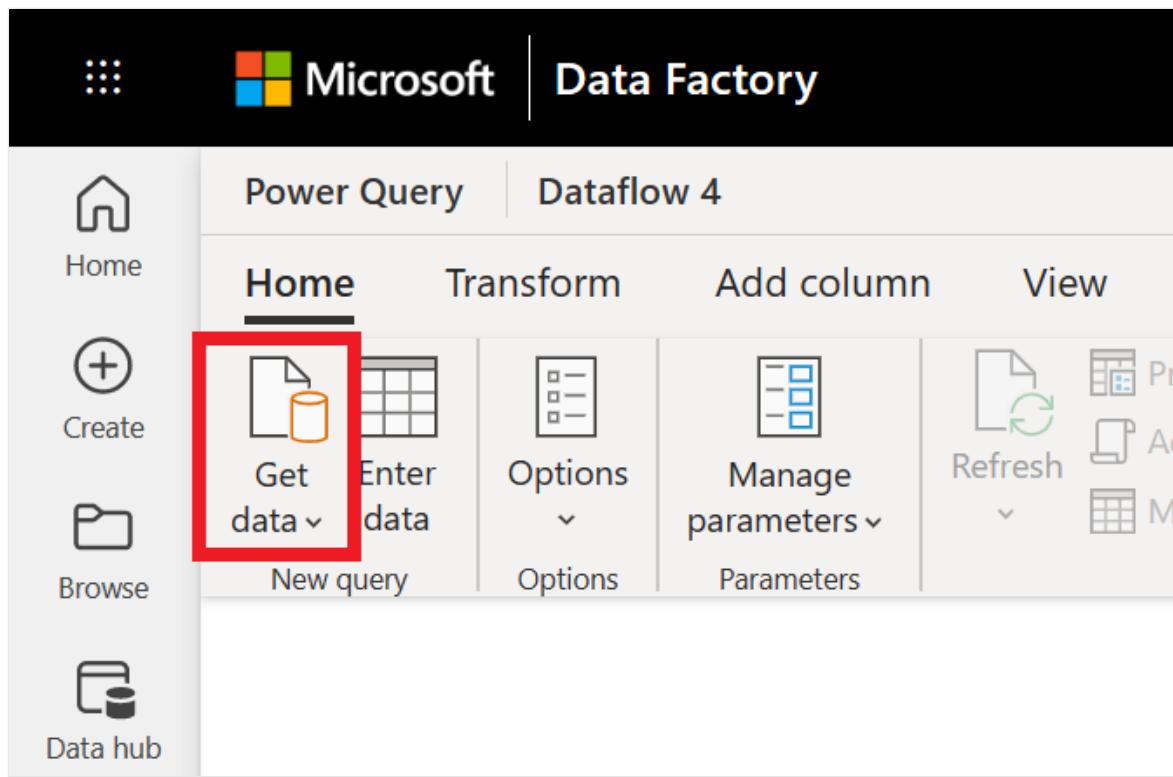
1. From your workspace, select **New > Dataflow Gen2 (Preview)** to create a new dataflow.



The screenshot shows the Microsoft Data Factory interface. At the top, there is a navigation bar with the Microsoft logo, the text "Data Factory DocTestWorkspace", and a search bar. On the left, a sidebar lists "Home", "Create" (with a red box around the "+ New" button), "Browse", "Data hub", "Workspaces", and "DocTestWor kspace". The main area displays a workspace named "DocTestWorkspace" described as a "Workspace for testing connector documents". It shows three items: "Dataflow Monitoring" (Dataset, Owner: DocT), "DocsTestDataflow" (Dataflow, Owner: DocT), and "DocsTestDatamart" (Dataset (default), Owner: DocT). A circular icon with a plus sign and a magnifying glass is visible in the bottom right corner.

Name	Type	Owner
Dataflow Monitoring	Dataset	DocT
DocsTestDataflow	Dataflow	DocT
DocsTestDatamart	Dataset (default)	DocT

2. In Power Query, either select **Get data** in the ribbon or select **Get data from another source** in the current view.



3. From **Choose data source**, select the Azure category, and then select **Azure SQL database**.

A screenshot of the 'Choose data source' dialog in Microsoft Data Factory. The title is 'Choose data source' and the sub-instruction is 'Select a connector or directly drag a file from your computer.' Below is a tab bar with 'All categories', 'File', 'Database', 'Power Platform', 'Azure' (which is underlined), 'Online services', and 'Other'. There are eight data source options listed in a grid:

Azure SQL database Azure	Azure Synapse Analytics (SQL ...) Azure	A
Azure Tables Azure	Azure Data Explorer (Kusto) Azure	A
Azure Cosmos DB v2 Azure	Azure Cost Management Azure	A
Azure Time Series Insights Azure	HDInsight Interactive Query Azure	A

The 'Azure SQL database' option is highlighted with a red box.

4. In **Connect to data source**, under **Connection settings**, provide the name of the server and database.

You can also select and enter advanced options that modify the connection query, such as a command timeout or a native query (SQL statement). More information: [Connect using advanced options](#)

5. If necessary, select the name of your on-premises data gateway.

6. If you're connecting to this database for the first time, select the authentication type to use in **Authentication kind**, and then enter your credentials. More information: [Connections and authentication in Power Query Online](#)

Connect to data source

Azure SQL database
Azure
[Learn more](#)

Connection settings

Server * ⓘ
Example: testazuresqlserver.database.windows.net

Database
Example: Contoso_DB

> Advanced options

Connection credentials

Connection
Create new connection ⚡

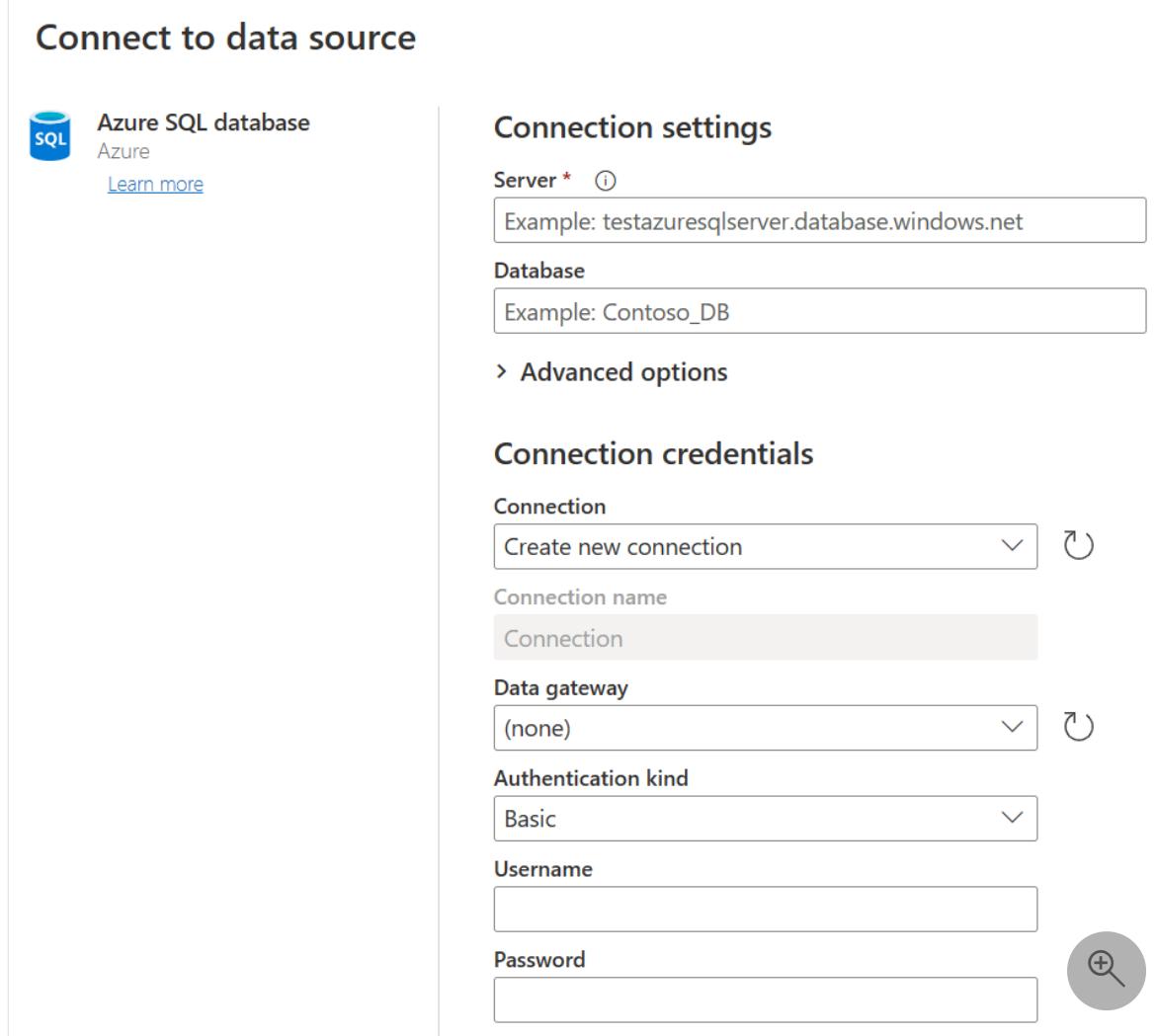
Connection name
Connection

Data gateway
(none) ⚡

Authentication kind
Basic ⚡

Username
[empty input field]

Password
[empty input field] ⚡



7. Select **Next**.

8. In **Choose data**, select the data item that you want to transform, and then select **Create**.

Choose data

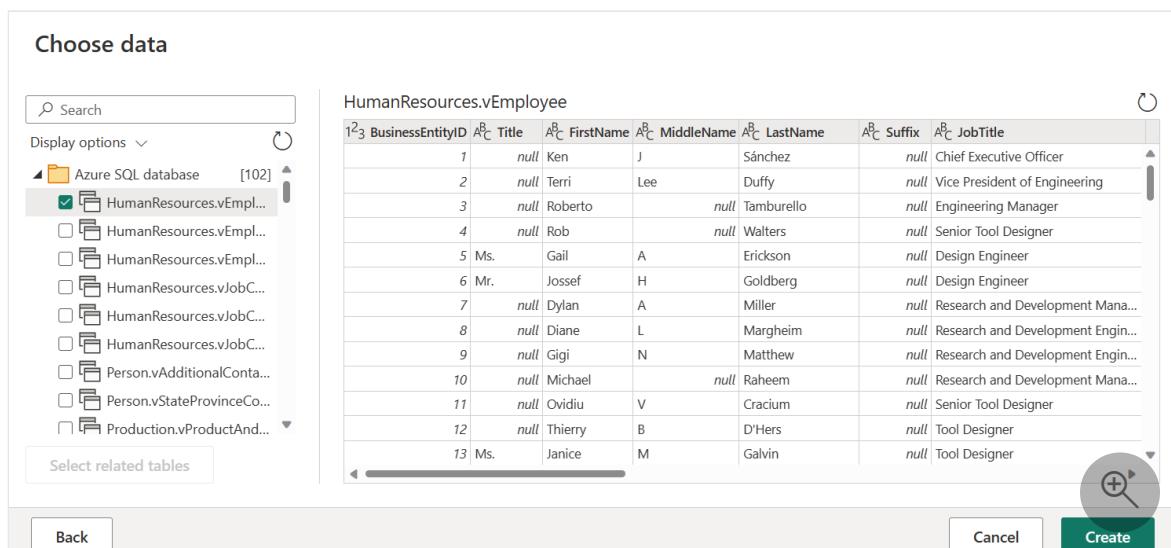
Search
Display options ⚡

Azure SQL database [102]
 HumanResources.vEmployee...
 HumanResources.vEmpl...
 HumanResources.vEmpl...
 HumanResources.vJobC...
 HumanResources.vJobC...
 Person.vAdditionalConta...
 Person.vStateProvinceCo...
 Production.vProductAnd...
[Select related tables](#)

HumanResources.vEmployee

BusinessEntityID	Title	FirstName	MiddleName	LastName	Suffix	JobTitle
1	null	Ken	J	Sánchez	null	Chief Executive Officer
2	null	Terri	Lee	Duffy	null	Vice President of Engineering
3	null	Roberto		Tamburello	null	Engineering Manager
4	null	Rob		Walters	null	Senior Tool Designer
5	Ms.	Gail	A	Erickson	null	Design Engineer
6	Mr.	Jossef	H	Goldberg	null	Design Engineer
7	null	Dylan	A	Miller	null	Research and Development Mana...
8	null	Diane	L	Margheim	null	Research and Development Engin...
9	null	Gigi	N	Matthew	null	Research and Development Engin...
10	null	Michael		Raheem	null	Research and Development Mana...
11	null	Ovidiu	V	Craciun	null	Senior Tool Designer
12	null	Thierry	B	D'Hers	null	Tool Designer
13	Ms.	Janice	M	Galvin	null	Tool Designer

Back **Cancel** **Create** ⚡



Advanced connector information

For more advanced information about connecting to your data using the Azure SQL server connector, go to [Azure SQL database](#).

Next steps

- [How to create an Azure SQL database connection](#)
- [How to configure Azure SQL database in a copy activity](#)

Data Warehouse connector overview

Article • 05/23/2023

The Data Warehouse connector is supported in Data Factory for Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/destination)	None	User Auth
Lookup activity	None	User Auth
GetMetadata activity	None	User Auth
Script activity	None	User Auth
Stored Procedure Activity	None	User Auth

Next steps

- [How to configure Data Warehouse in a copy activity](#)

How to configure the Data Warehouse connector for the copy activity in Data Factory in Microsoft Fabric

Article • 05/23/2023

This article outlines how to use the copy activity in data pipeline to copy data from and to a Data Warehouse.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported configuration

For the configuration of each tab under copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Destination](#)
- [Mapping](#)
- [Settings](#)

General

For the General tab configuration, go to [General](#).

Source

The following properties are supported for Data Warehouse as Source in a copy activity.

General Source Destination ¹ Mapping Settings

Data store type	<input checked="" type="radio"/> Workspace <input type="radio"/> External <input type="radio"/> Sample dataset
Workspace data store type	<input type="button" value="Data Warehouse"/>
Data Warehouse	<input type="button" value="DataWarehouse"/> <input type="button" value="Refresh"/>
Use query	<input checked="" type="radio"/> Table <input type="radio"/> Query <input type="radio"/> Stored procedure
Table	<input type="button" value="Edit"/> <input type="button" value="Refresh"/> <input type="button" value="Preview data"/>
▼ Advanced	
Query timeout (minutes) ^①	<input type="text" value="120"/>
Isolation level ^①	<input type="button" value="None"/>
Partition option ^①	<input checked="" type="radio"/> None <input type="radio"/> Dynamic range ^①
ⓘ Please preview data to validate the partition settings are correct before you trigger a run or publish the pipeline.	
Additional columns ^①	<input type="button" value="New"/>

The following properties are required:

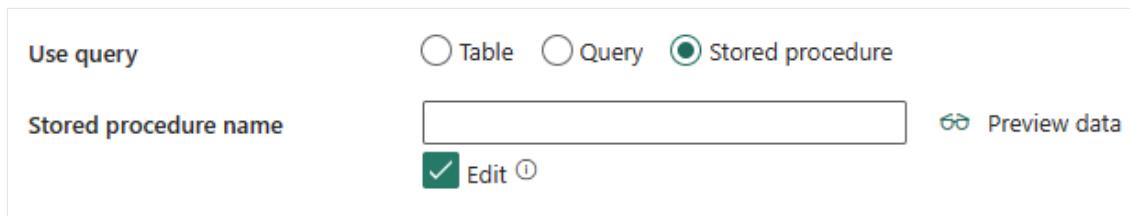
- **Data store type:** Select **Workspace**.
- **Workspace data store type:** Select **Data Warehouse** from the data store type list.
- **Data Warehouse:** Select an existing **Data Warehouse** from the workspace.
- **Use query:** Select **Table**, **Query**, or **Stored procedure**.
 - If you select **Table**, choose an existing table from the table list, or specify a table name manually by selecting the **Edit** box.

Use query	<input checked="" type="radio"/> Table <input type="radio"/> Query <input type="radio"/> Stored procedure
Table	<input type="button" value="Edit"/> <input type="button" value="Preview data"/>

- If you select **Query**, use the custom SQL query editor to write a SQL query that retrieves the source data.

Use query	<input type="radio"/> Table <input checked="" type="radio"/> Query <input type="radio"/> Stored procedure
Query *	<input type="button" value="Edit"/> <input type="button" value="Preview data"/>

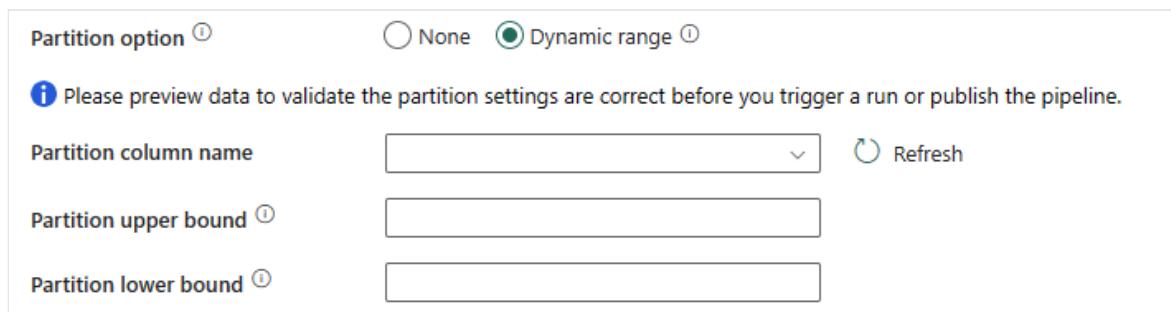
- If you select **Stored procedure**, choose an existing stored procedure from the drop-down list, or specify a stored procedure name as the source by selecting the **Edit** box.



Under **Advanced**, you can specify the following fields:

- **Query timeout (minutes)**: Timeout for query command execution, with a default of 120 minutes. If this property is set, the allowed values are in the format of a timespan, such as "02:00:00" (120 minutes).
- **Isolation level**: Specify the transaction locking behavior for the SQL source.
- **Partition option**: Specify the data partitioning options used to load data from Data Warehouse. You can select **None** or **Dynamic range**.

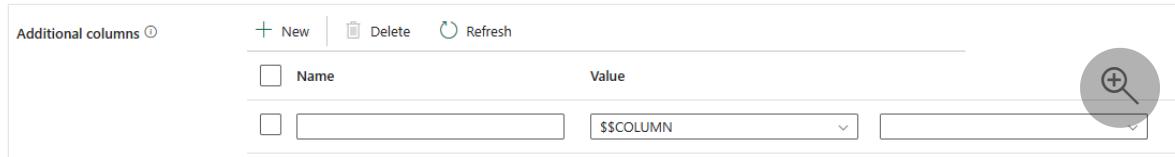
If you select **Dynamic range**, the range partition parameter([?](#) `AdfDynamicRangePartitionCondition`) is needed when using query with parallel enabled. Sample query: `SELECT * FROM <TableName> WHERE ? AdfDynamicRangePartitionCondition`.



- **Partition column name**: Specify the name of the source column in `integer` or `date/datetime type` (`int`, `smallint`, `bigint`, `date`, `smalldatetime`, `datetime`, `datetime2`, or `datetimeoffset`) that is used by range partitioning for parallel copy. If not specified, the index or the primary key of the table is detected automatically and used as the partition column.
- **Partition upper bound**: The maximum value of the partition column for partition range splitting. This value is used to decide the partition stride, not for filtering the rows in table. All rows in the table or query result are partitioned and copied.
- **Partition lower bound**: The minimum value of the partition column for partition range splitting. This value is used to decide the partition stride, not for filtering

the rows in table. All rows in the table or query result are partitioned and copied.

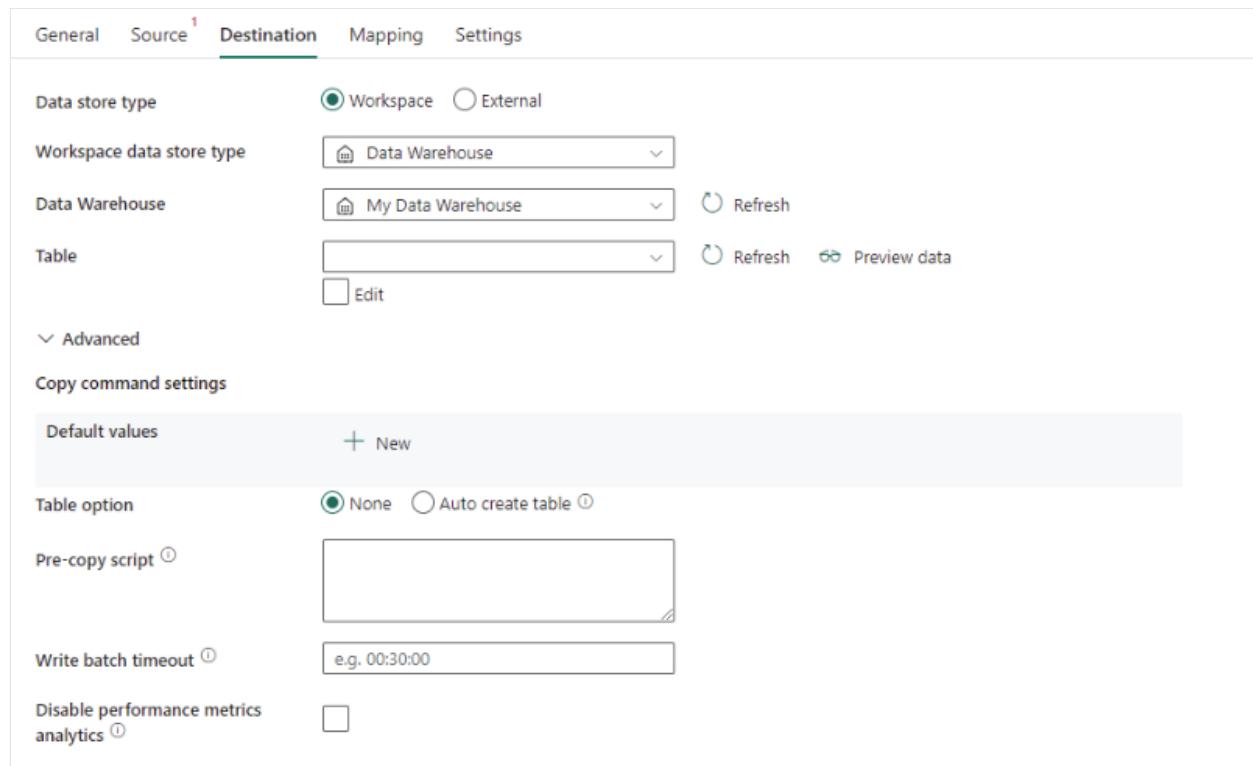
- **Additional columns:** Add additional data columns to store source files' relative path or static value. Expression is supported for the latter.



The screenshot shows a user interface for managing additional columns. At the top, there are buttons for 'New', 'Delete', and 'Refresh'. Below this is a table with two columns: 'Name' and 'Value'. A search icon is located in the top right corner of the table area.

Destination

The following properties are supported for Data Warehouse as **Destination** in a copy activity.



The screenshot shows the 'Destination' tab settings for a copy activity. The tabs at the top are General, Source (with a red exclamation mark), Destination (selected), Mapping, and Settings. Under 'Destination', the 'Data store type' is set to 'Workspace' (radio button selected). The 'Workspace data store type' dropdown is set to 'Data Warehouse'. The 'Data Warehouse' dropdown shows 'My Data Warehouse'. The 'Table' dropdown is empty. There are 'Refresh' and 'Preview data' buttons. A 'Copy command settings' section contains fields for 'Default values', 'Table option' (set to 'None'), 'Pre-copy script', 'Write batch timeout' (set to 'e.g. 00:30:00'), and 'Disable performance metrics analytics' (checkbox). A 'Advanced' section is collapsed.

The following properties are **required**:

- **Data store type:** Select **Workspace**.
- **Workspace data store type:** Select **Data Warehouse** from the data store type list.
- **Data Warehouse:** Select an existing **Data Warehouse** from the workspace.
- **Table:** Choose an existing table from the table list or specify a table name as destination.

Under **Advanced**, you can specify the following fields:

- **Copy command settings:** Specify copy command properties.

Copy command settings

Default values		New	Delete	More	Refresh
<input type="checkbox"/>	Column	Value			
<input type="checkbox"/>	<input type="text"/>	<input type="text"/>			
Add dynamic content [Alt+Shift+D]					

- **Table options:** Specify whether to automatically create the destination table if none exists based on the source schema. You can select **None** or **Auto create table**.
- **Pre-copy script:** Specify a SQL query to run before writing data into Data Warehouse in each run. Use this property to clean up the preloaded data.
- **Write batch timeout:** The wait time for the batch insert operation to finish before it times out. The allowed values are in the format of a timespan. The default value is "00:30:00" (30 minutes).
- **Disable performance metrics analytics:** The service collects metrics for copy performance optimization and recommendations. If you're concerned with this behavior, turn off this feature.

If your source data is in **Azure Blob Storage** or **Azure Data Lake Storage Gen2**, and the format is COPY statement compatible, copy activity directly invokes the COPY command to let Data Warehouse pull the data from the source.

1. The source data and format contain the following types and authentication methods:

Supported source data store type	Supported format	Supported source authentication type
Azure Blob Storage	Delimited text Parquet	Anonymous authentication Shared access signature authentication
Azure Data Lake Storage Gen2	Delimited text Parquet	Shared access signature authentication

2. The following Format settings can be set:

- a. For **Parquet**: compression can be no compression, Snappy, or GZip.
- b. For **Delimited text**:
 - i. `rowDelimiter` is explicitly set as **single character** or "`\r\n`", the default value isn't supported.
 - ii. `nullValue` is left as default or set to **empty string ("")**.

iii. `encodingName` is left as default or set to **utf-8** or **utf-16**.

iv. `skipLineCount` is left as default or set to 0.

v. compression can be **no compression** or **GZip**.

3. If your source is a folder, `recursive` in a copy activity must be set to true.

4. `modifiedDateTimeStart`, `modifiedDateTimeEnd`, `prefix`, `enablePartitionDiscovery`, and `additionalColumns` aren't specified.

If your source data store and format isn't originally supported by a COPY statement, use the Staged copy by using the COPY statement feature instead. The staged copy feature also provides you with better throughput. It automatically converts the data into a COPY statement compatible format, then calls a COPY statement to load data into Data Warehouse.

Mapping

For the **Mapping** tab configuration, go to [Mapping](#).

Settings

For the **Settings** tab configuration, go to [Settings](#).

Table summary

The following tables contain more information about a copy activity in Data Warehouse.

Source information

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	Workspace	Yes	/
Workspace data store type	The section to select your workspace data store type.	Data Warehouse	Yes	type
Data Warehouse	The Data Warehouse that you want to use.	<your data warehouse>	Yes	endpoint artifactId

Name	Description	Value	Required	JSON script property
Use query	The way to read data from Data Warehouse.	<ul style="list-style-type: none"> Tables Query Stored procedure 	No	<p>(under <code>typeProperties -> source</code>)</p> <ul style="list-style-type: none"> • <code>typeProperties:</code> schema table • <code>sqlReaderQuery</code> • <code>sqlReaderStoredProcedureName</code>
Query timeout (minutes)	Timeout for query command execution, with a default of 120 minutes. If this property is set, the allowed values are in the format of a timespan, such as "02:00:00" (120 minutes).	timespan	No	<code>queryTimeout</code>
Isolation level	The transaction locking behavior for source.	<ul style="list-style-type: none"> • None • Snapshot 	No	<code>isolationLevel</code>
Partition option	The data partitioning options used to load data from Data Warehouse.	<ul style="list-style-type: none"> • None • Dynamic range 	No	<code>partitionOption</code>
Partition column name	<p>The name of the source column in integer or date/datetime type (<code>int, smallint, bigint, date, smalldatetime, datetime, datetime2, or datetimeoffset</code>) that is used by range partitioning for parallel copy. If not specified, the index or the primary key of the table is detected automatically and used as the partition column.</p>	<partition column name>	No	<code>partitionColumnName</code>

Name	Description	Value	Required	JSON script property
Partition upper bound	The maximum value of the partition column for partition range splitting. This value is used to decide the partition stride, not for filtering the rows in table. All rows in the table or query result are partitioned and copied.	<partition upper bound>	No	partitionUpperBound
Partition lower bound	The minimum value of the partition column for partition range splitting. This value is used to decide the partition stride, not for filtering the rows in table. All rows in the table or query result are partitioned and copied.	<partition lower bound>	No	partitionLowerBound
Additional columns	Add additional data columns to store source files' relative path or static value.	<ul style="list-style-type: none"> • Name • Value 	No	additionalColumns: <ul style="list-style-type: none"> • name • value

Destination information

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	Workspace	Yes	/
Workspace data store type	The section to select your workspace data store type.	Data Warehouse	Yes	type
Data Warehouse	The Data Warehouse that you want to use.	<your data warehouse>	Yes	endpoint artifactId

Name	Description	Value	Required	JSON script property
Table	The destination table to write data.	<name of your destination table>	Yes	schema table
Copy command settings	The copy command property settings. Contains the default value settings.	Default value: • Column • Value	No	copyCommandSettings: defaultValues: • columnName • defaultValue
Table option	Whether to automatically create the destination table if none exists based on the source schema.	• None • Auto create table	No	tableOption: • autoCreate
Pre-copy script	A SQL query to run before writing data into Data Warehouse in each run. Use this property to clean up the preloaded data.	<pre-copy script>	No	preCopyScript
Write batch timeout	The wait time for the batch insert operation to finish before it times out. The allowed values are in the format of a timespan. The default value is "00:30:00" (30 minutes).	timespan	No	writeBatchTimeout
Disable performance metrics analytics	The service collects metrics for copy performance optimization and recommendations, which introduce additional master DB access.	select or unselect	No	disableMetricsCollection: true or false

Next steps

- [Data Warehouse connector overview](#)

Binary format for Data Factory in Microsoft Fabric

Article • 05/23/2023

This article outlines how to configure Binary format in Data Factory.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Binary format is supported for the following activities and connectors as source and destination.

Category	Connector/Activity
Supported connector	Amazon S3
	Azure Blob Storage
	Azure Data Lake Storage Gen1
	Azure Data Lake Storage Gen2
	Google Cloud Storage
	HTTP
Supported activity	Copy activity
	GetMetadata activity
	Delete activity

Binary format in copy activity

To configure Binary format, choose your connection in the source or destination of the data pipeline copy activity, and then select **Binary** in the drop-down list of **File format**.

Select **Settings** for further configuration of this format.

The screenshot shows the 'File format' settings dialog box. The 'File format' dropdown is set to 'Avro'. A red box highlights the 'Settings' button. A dropdown menu lists several file formats: Avro, Binary, DelimitedText, Excel, JSON, ORC, Parquet, and XML. The 'Avro' option is selected.

(!) Note

When using Binary format in a copy activity, source and destination should both use Binary format.

Binary as source

After you select **Settings** in the **File format** section under the **Source** tab, the following properties are displayed in the pop-up **File format settings** dialog box.

The screenshot shows the 'File format settings' dialog box. The 'Compression type' dropdown is set to 'None'.

- **Compression type:** The compression codec used to read binary files. You can choose from the **None**, **bzip2**, **gzip**, **deflate**, **ZipDeflate**, **TarGzip**, or **tar** type in the drop-down list.

If you select **ZipDeflate** as the compression type, **Preserve zip file name as folder** is displayed under the **Advanced** settings in the **Source** tab.

- **Preserve zip file name as folder:** Indicates whether to preserve the source zip file name as a folder structure during copy.
 - If this box is checked (default), the service writes unzipped files to `<specified file path>/<folder named as source zip file>/`.
 - If this box is unchecked, the service writes unzipped files directly to `<specified file path>`. Make sure you don't have duplicated file names in different source zip files to avoid racing or unexpected behavior.

If you select **TarGzip/tar** as the compression type, **Preserve compression file name as folder** is displayed under the **Advanced** settings in the **Source** tab.

- **Preserve compression file name as folder:** Indicates whether to preserve the source compressed file name as a folder structure during copy.
 - If this box is checked (default), the service writes decompressed files to `<specified file path>/<folder named as source compressed file>/`.
 - If this box is unchecked, the service writes decompressed files directly to `<specified file path>`. Make sure you don't have duplicated file names in different source zip files to avoid racing or unexpected behavior.
- **Compression level:** The compression ratio. You can choose from **Optimal** or **Fastest**.
 - **Fastest:** The compression operation should complete as quickly as possible, even if the resulting file isn't optimally compressed.
 - **Optimal:** The compression operation should be optimally compressed, even if the operation takes a longer time to complete. For more information, go to the [Compression Level](#) article.

Under **Advanced** settings in the **Source** tab, further Binary format related property are displayed.

- **Delete files after completion:** Indicates whether the binary files are deleted from the source store after successfully moving to the destination store. The file deletion is per file. So when a copy activity fails, some files have already been copied to the destination and deleted from the source, while others still remain on the source store.

Binary as destination

After you select **Settings** in the **File format** section under the **Destination** tab, following properties are displayed in the pop-up **File format settings** dialog box.

File format settings

Compression type

None

- **Compression type:** The compression codec used to write binary files. You can choose from the **None**, **bzip2**, **gzip**, **deflate**, **ZipDeflate**, **TarGzip** or **tar** type in the drop-down list.
- **Compression level:** The compression ratio. You can choose from **Optimal** or **Fastest**.
 - **Fastest:** The compression operation should complete as quickly as possible, even if the resulting file isn't optimally compressed.
 - **Optimal:** The compression operation should be optimally compressed, even if the operation takes a longer time to complete. For more information, go to the [Compression Level](#) article.

Table summary

Binary as source

The following properties are supported in the copy activity **Source** section when using Binary format.

Name	Description	Value	Required	JSON script property
File format	The file format that you want to use.	Binary	Yes	type (<i>under datasetSettings</i>): Binary

Name	Description	Value	Required	JSON script property
Compression type	The compression codec used to read binary files.	Choose from: None bzip2 gzip deflate ZipDeflate TarGzip tar	No	type (<i>under compression</i>): bzip2 gzip deflate ZipDeflate TarGzip tar
Compression level	The compression ratio. Allowed values are Optimal or Fastest.	Optimal or Fastest	No	level (<i>under compression</i>): Fastest Optimal
Preserve zip file name as folder	Indicates whether to preserve the source zip file name as a folder structure during copy.	Selected or unselect	No	preserveZipFileNameAsFolder (<i>under compressionProperties -> type as ZipDeflateReadSettings</i>)
Preserve compression file name as folder	Indicates whether to preserve the source compressed file name as a folder structure during copy.	Selected or unselect	No	preserveCompressionFileNameAsFolder (<i>under compressionProperties -> type as TarGZipReadSettings or TarReadSettings</i>)
Delete files after completion	Indicates whether the binary files are deleted from the source store after successfully moving to the destination store.	Selected or unselect	No	deleteFilesAfterCompletion: true or false

Binary as destination

The following properties are supported in the copy activity **Destination** section when using Binary format.

Name	Description	Value	Required	JSON script property
File format	The file format that you want to use.	Binary	Yes	type (<i>under datasetSettings</i>): Binary
Compression type	The compression codec used to write binary files.	Choose from: None bzip2 gzip deflate ZipDeflate TarGzip tar	No	type (<i>under compression</i>): bzip2 gzip deflate ZipDeflate TarGzip tar
Compression level	The compression ratio. Allowed values are Optimal or Fastest.	Optimal or Fastest	No	level (<i>under compression</i>): Fastest Optimal

Next steps

- [Connectors overview](#)
- [Connect to Binary format in dataflows](#)

Connect to Binary format in dataflows

Article • 05/23/2023

A Binary format connector isn't currently supported in Dataflow Gen2.

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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Next steps

- [How to configure Binary format in Data Factory](#)

Delimited text format in Data Factory in Microsoft Fabric

Article • 05/23/2023

This article outlines how to configure delimited text format in the data pipeline of Data Factory in Microsoft Fabric.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

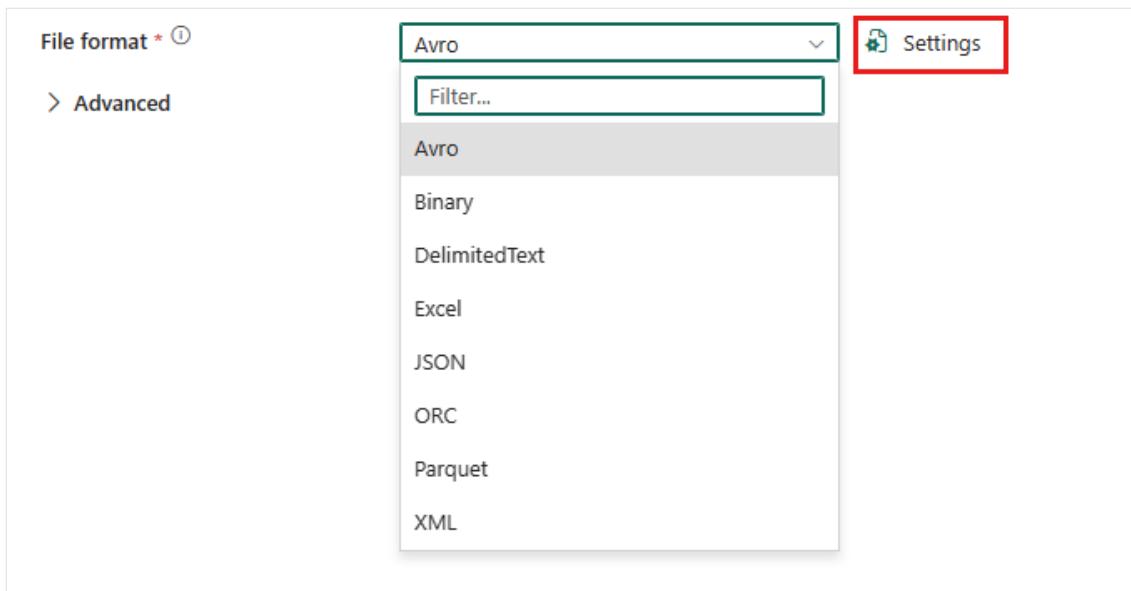
Supported capabilities

Delimited text format is supported for the following activities and connectors as source and destination.

Category	Connector/Activity
Supported connector	Amazon S3
	Azure Blob Storage
	Azure Data Lake Storage Gen1
	Azure Data Lake Storage Gen2
	Google Cloud Storage
	HTTP
Supported activity	Copy activity
	Lookup activity
	GetMetadata activity
	Delete activity

Delimited text format in copy activity

To configure delimited text format, choose your connection in the source or destination of data pipeline copy activity, and then select **DelimitedText** in the drop-down list of **File format**. Select **Settings** for further configuration of this format.



Delimited text format as source

After selecting **Settings** in **File format** section, following properties are shown up in the pop-up **File format settings** dialog box.

File format settings

Compression type	None
Column delimiter ⓘ	Comma (,)
Row delimiter ⓘ	Line feed (\n)
Encoding ⓘ	Default(UTF-8)
Escape character ⓘ	Backslash (\)
Quote character ⓘ	Double quote ("")
First row as header ⓘ	<input type="checkbox"/>
Null value ⓘ	<input type="text"/>

- **Compression type:** The compression codec used to read delimited text files. You can choose from **None, bzip2, gzip, deflate, ZipDeflate, TarGzip or tar** type in the drop-down list.

If you select **ZipDeflate** as compression type, **Preserve zip file name as folder** will show up under **Advanced** settings in **Source** tab.

- **Preserve zip file name as folder:** Indicates whether to preserve the source zip file name as folder structure during copy.
 - If this box is checked (default), the service writes unzipped files to `<specified file path>/<folder named as source zip file>/`.
 - If this box is unchecked, the service writes unzipped files directly to `<specified file path>`. Make sure you don't have duplicated file names in different source zip files to avoid racing or unexpected behavior.

If you select **TarGzip/tar** as compression type, **Preserve compression file name as folder** will show up under **Advanced** settings in **Source** tab.

- **Preserve compression file name as folder:** Indicates whether to preserve the source compressed file name as folder structure during copy.
 - If this box is checked (default), the service writes decompressed files to `<specified file path>/<folder named as source compressed file>/.`
 - If this box is unchecked, the service writes decompressed files directly to `<specified file path>`.
Make sure you don't have duplicated file names in different source zip files to avoid racing or unexpected behavior.
- **Compression level:** Specify the compression ratio when you select a compression type. You can choose from **Optimal** or **Fastest**.
 - **Fastest:** The compression operation should complete as quickly as possible, even if the resulting file is not optimally compressed.
 - **Optimal:** The compression operation should be optimally compressed, even if the operation takes a longer time to complete. For more information, see [Compression Level topic](#).
- **Column delimiter:** The character(s) used to separate columns in a file. The default value is **comma (,)**.
- **Row delimiter:** Specify the character used to separate rows in a file. Only one character is allowed. The default value is line feed `\n`.
- **Encoding:** The encoding type used to read/write test files. The default value is **UTF-8**.
- **Escape character:** The single character to escape quotes inside a quoted value. The default value is **backslash **. When escape character is defined as empty string, the **Quote character** must be set as empty string as well, in which case make sure all column values don't contain delimiters.
- **Quote character:** The single character to quote column values if it contains column delimiter. The default value is **double quotes "**. When **Quote character** is defined as empty string, it means there is no quote char and column value is not quoted, and escape character is used to escape the column delimiter and itself.
- **First row as header:** Specifies whether to treat/make the first row as a header line with names of columns. Allowed values are selected and unselected (default). When first row as header is unselected, note UI data preview and lookup activity output auto generate column names as Prop_{n} (starting from 0), copy activity requires [explicit mapping](#) from source to destination and locates columns by ordinal (starting from 1).
- **Null value:** Specifies the string representation of null value. The default value is empty string.

Under **Advanced** settings in **Source** tab, further delimited text format related property are shown up.

- **Skip line count:** Indicates the number of nonempty rows to skip when reading data from input files. If both **Skip line count** and **First row as header** are specified, the lines are skipped first, and then the header information is read from the input file.

Delimited text format as destination

After selecting **Settings in File format** section, following properties are shown up in the pop-up **File format settings** dialog box.

File format settings

Compression type	None
Column delimiter ⓘ	Comma (,)
Row delimiter ⓘ	Line feed (\n)
Encoding ⓘ	Default(UTF-8)
Escape character ⓘ	Backslash (\)
Quote character ⓘ	Double quote ("")
First row as header ⓘ	<input type="checkbox"/>
Null value ⓘ	

- **Compression type:** The compression codec used to write delimited text files. You can choose from **None**, **bzip2**, **gzip**, **deflate**, **ZipDeflate**, **TarGzip** or **tar** type in the drop-down list.
- **Compression level:** Specify the compression ratio when you select a compression type. You can choose from **Optimal** or **Fastest**.
 - **Fastest:** The compression operation should complete as quickly as possible, even if the resulting file is not optimally compressed.
 - **Optimal:** The compression operation should be optimally compressed, even if the operation takes a longer time to complete. For more information, see [Compression Level topic](#).
- **Column delimiter:** The character(s) used to separate columns in a file. The default value is comma (,).
- **Row delimiter:** The character used to separate rows in a file. Only one character is allowed. The default value is line feed (\n).
- **Encoding:** The encoding type used to write test files. The default value is **UTF-8**.
- **Escape character:** The single character to escape quotes inside a quoted value. The default value is **backslash** (\). When escape character is defined as empty string, the **Quote character** must be set as empty string as well, in which case make sure all column values don't contain delimiters.
- **Quote character:** The single character to quote column values if it contains column delimiter. The default value is **double quotes** (""). When **Quote character** is defined as empty string, it means there is no quote char and column value is not quoted, and escape character is used to escape the column delimiter and itself.
- **First row as header:** Specifies whether to treat/make the first row as a header line with names of columns. Allowed values are selected and unselected (default). When first row as header is unselected, note UI data preview and lookup activity output auto generate column names as Prop_{n} (starting from 0), copy activity requires [explicit mapping](#) from source to destination and locates columns by ordinal (starting from 1).
- **Null value:** Specifies the string representation of null value. The default value is empty string.

Under **Advanced** settings in **Destination** tab, further delimited text format related property are shown up.

- Quote all text:** Enclose all values in quotes.
- File extension:** The file extension used to name the output files, for example, `.csv`, `.txt`.
- Max rows per file:** When writing data into a folder, you can choose to write to multiple files and specify the max rows per file.
- File name prefix:** Applicable when **Max rows per file** is configured. Specify the file name prefix when writing data to multiple files, resulted in this pattern: `<fileNamePrefix>_00000.<fileExtension>`. If not specified, file name prefix will be auto generated. This property does not apply when source is file based store or partition option enabled data store.

Table summary

Delimited text as source

The following properties are supported in the copy activity **Source** section when using delimited text format.

Name	Description	Value	Required	JSON script property
File format	The file format that you want to use.	DelimitedText	Yes	type (under <code>datasetSettings</code>): DelimitedText
Compression type	The compression codec used to read delimited text files.	Choose from: None bzip2 gzip deflate ZipDeflate TarGzip tar	No	type (under <code>compression</code>): bzip2 gzip deflate ZipDeflate TarGzip tar
Preserve zip file name as folder	Indicates whether to preserve the source zip file name as folder structure during copy. Applies when you select ZipDeflate compression.	Selected or unselect	No	preserveZipFileNameAsFolder (under <code>compressionProperties->type as ZipDeflateReadSettings</code>)
Preserve compression file name as folder	Indicates whether to preserve the source compressed file name as folder structure during copy. Applies when you select TarGzip/tar compression.	Selected or unselect	No	preserveCompressionFileNameAsFolder (under <code>compressionProperties->type as TarGZipReadSettings or TarReadSettings</code>)
Compression level	The compression ratio. Allowed values are Optimal or Fastest.	Optimal or Fastest	No	level (under <code>compression</code>): Fastest Optimal

Name	Description	Value	Required	JSON script property
Column delimiter	The character(s) used to separate columns in a file.	< the selected column delimiter > <code>comma</code> , (by default)	No	columnDelimiter
Row delimiter	The character used to separate rows in a file.	< the selected row delimiter > <code>\r</code> , <code>\n</code> (by default), or <code>r\n</code>	No	rowDelimiter
Encoding	The encoding type used to read/write test files.	"UTF-8" (by default), "UTF-8 without BOM", "UTF-16", "UTF-16BE", "UTF-32", "UTF-32BE", "US-ASCII", "UTF-7", "BIG5", "EUC-JP", "EUC-KR", "GB2312", "GB18030", "JOHAB", "SHIFT-JIS", "CP875", "CP866", "IBM00858", "IBM037", "IBM273", "IBM437", "IBM500", "IBM737", "IBM775", "IBM850", "IBM852", "IBM855", "IBM857", "IBM860", "IBM861", "IBM863", "IBM864", "IBM865", "IBM869", "IBM870", "IBM01140", "IBM01141", "IBM01142", "IBM01143", "IBM01144", "IBM01145", "IBM01146", "IBM01147", "IBM01148", "IBM01149", "ISO-2022-JP", "ISO-2022-KR", "ISO-8859-1", "ISO-8859-2", "ISO-	No	encodingName

Name	Description	Value	Required	JSON script property
		8859-3", "ISO- 8859-4", "ISO- 8859-5", "ISO- 8859-6", "ISO- 8859-7", "ISO- 8859-8", "ISO- 8859-9", "ISO- 8859-13", "ISO- 8859-15", "WINDOWS-874", "WINDOWS- 1250", "WINDOWS- 1251", "WINDOWS- 1252", "WINDOWS- 1253", "WINDOWS- 1254", "WINDOWS- 1255", "WINDOWS- 1256", "WINDOWS- 1257", "WINDOWS-1258"		
Escape character	The single character to escape quotes inside a quoted value. When escape character is defined as empty string, the Quote character must be set as empty string as well, in which case make sure all column values don't contain delimiters.	< your selected escape character > backslash \ (by default)	No	escapeChar
Quote character	The single character to quote column values if it contains column delimiter. When Quote character is defined as empty string, it means there is no quote char and column value is not quoted, and escape character is used to escape the column delimiter and itself.	< your selected quote character > double quotes " (by default)	No	quoteChar

Name	Description	Value	Required	JSON script property
First row as header	Specifies whether to treat the first row in the given worksheet/range as a header line with names of columns.	Selected or unselected	No	firstRowAsHeader: true or false (default)
Null value	Specifies the string representation of null value. The default value is empty string.	< the string representation of null value > empty string (by default)	No	nullValue
Skip line count	Indicates the number of non-empty rows to skip when reading data from input files. If both Skip line count and First row as header are specified, the lines are skipped first and then the header information is read from the input file.	< your skip line count >	No	skipLineCount

Delimited text as destination

The following properties are supported in the copy activity **Destination** section when using delimited text format.

Name	Description	Value	Required	JSON script property
File format	The file format that you want to use.	DelimitedText	Yes	type (<i>under datasetSettings</i>): DelimitedText
Compression type	The compression codec used to write delimited text files.	Choose from: None bzip2 gzip deflate ZipDeflate TarGzip tar	No	type (<i>under compression</i>): bzip2 gzip deflate ZipDeflate TarGzip tar
Preserve zip file name as folder	Indicates whether to preserve the source zip file name as folder structure during copy.	Selected or unselect	No	preserveZipFileNameAsFolder (<i>under compressionProperties - > type as ZipDeflateReadSettings</i>)
Preserve compression file name as folder	Indicates whether to preserve the source compressed file name as folder structure during copy.	Selected or unselect	No	preserveCompressionFileNameAsFolder (<i>under compressionProperties - > type as TarGzipReadSettings or TarReadSettings</i>)

Name	Description	Value	Required	JSON script property
Compression level	The compression ratio. Allowed values are Optimal or Fastest.	Optimal or Fastest	No	level (<i>under compression</i>): Fastest Optimal
Column delimiter	The character(s) used to separate columns in a file.	< the selected column delimiter > comma <code>,</code> (by default)	No	columnDelimiter
Row delimiter	The character used to separate rows in a file.	< the selected row delimiter > <code>\r</code> , <code>\n</code> (by default), or <code>r\n</code>	No	rowDelimiter
Encoding	The encoding type used to read/write test files.	"UTF-8" (by default), "UTF-8 without BOM", "UTF-16", "UTF-16BE", "UTF-32", "UTF-32BE", "US-ASCII", "UTF-7", "BIG5", "EUC-JP", "EUC-KR", "GB2312", "GB18030", "JOHAB", "SHIFT-JIS", "CP875", "CP866", "IBM00858", "IBM037", "IBM273", "IBM437", "IBM500", "IBM737", "IBM775", "IBM850", "IBM852", "IBM855", "IBM857", "IBM860", "IBM861", "IBM863", "IBM864", "IBM865", "IBM869", "IBM870", "IBM01140", "IBM01141", "IBM01142", "IBM01143", "IBM01144", "IBM01145", "IBM01146", "IBM01147", "IBM01148", "IBM01149", "ISO-2022-JP", "ISO-	No	encodingName

Name	Description	Value	Required	JSON script property
		2022-KR", "ISO-8859-1", "ISO-8859-2", "ISO-8859-3", "ISO-8859-4", "ISO-8859-5", "ISO-8859-6", "ISO-8859-7", "ISO-8859-8", "ISO-8859-9", "ISO-8859-13", "ISO-8859-15", "WINDOWS-874", "WINDOWS-1250", "WINDOWS-1251", "WINDOWS-1252", "WINDOWS-1253", "WINDOWS-1254", "WINDOWS-1255", "WINDOWS-1256", "WINDOWS-1257", "WINDOWS-1258"		
Escape character	The single character to escape quotes inside a quoted value. When escape character is defined as empty string, the Quote character must be set as empty string as well, in which case make sure all column values don't contain delimiters.	< your selected escape character > backslash \ (by default)	No	escapeChar
Quote character	The single character to quote column values if it contains column delimiter. When Quote character is defined as empty string, it means there is no quote char and column value is not quoted, and escape character is used to escape the column delimiter and itself.	< your selected quote character > double quotes " (by default)	No	quoteChar

Name	Description	Value	Required	JSON script property
First row as header	Specifies whether to treat the first row in the given worksheet/range as a header line with names of columns.	Selected or unselected	No	firstRowAsHeader: true or false (default)
Quote all text	Enclose all values in quotes.	Selected (default) or unselected	No	quoteAllText: true (default) or false
File extension	The file extension used to name the output files.	< your file extension > .txt (by default)	No	fileExtension
Max rows per file	When writing data into a folder, you can choose to write to multiple files and specify the max rows per file.	< your max rows per file >	No	maxRowsPerFile
File name prefix	Applicable when Max rows per file is configured. Specify the file name prefix when writing data to multiple files, resulted in this pattern: <code><fileNamePrefix>_00000. <fileExtension>.</code> If not specified, file name prefix will be auto generated. This property does not apply when source is file based store or partition option enabled data store.	< your file name prefix >	No	fileNamePrefix

Next steps

[Connectors overview](#)

Excel format in Data Factory in Microsoft Fabric

Article • 05/23/2023

This article outlines how to configure Excel format in the data pipeline of Data Factory in Microsoft Fabric.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

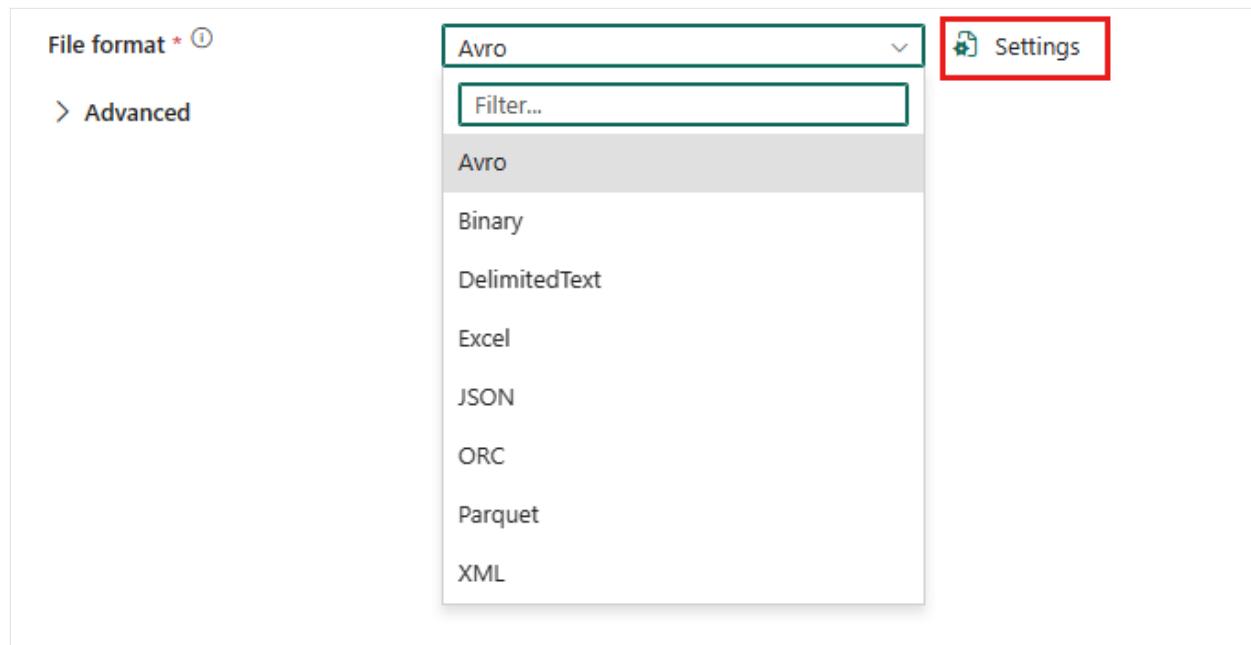
Supported capabilities

Excel format is supported for the following activities and connectors as source.

Category	Connector/Activity
Supported connector	Amazon S3
	Azure Blob Storage
	Azure Data Lake Storage Gen1
	Azure Data Lake Storage Gen2
	Google Cloud Storage
	HTTP
Supported activity	Copy activity
	Lookup activity
	GetMetadata activity
	Delete activity

Excel format in copy activity

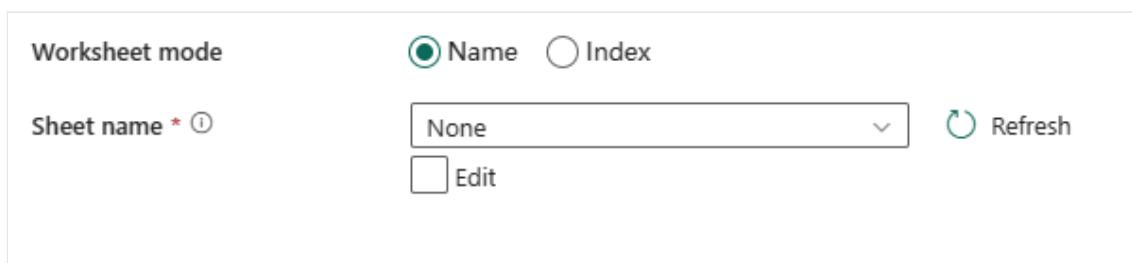
To configure Excel format, choose your connection in the source of data pipeline copy activity, and then select **Excel** in the drop-down list of **File format**. Select **Settings** for further configuration of this format.



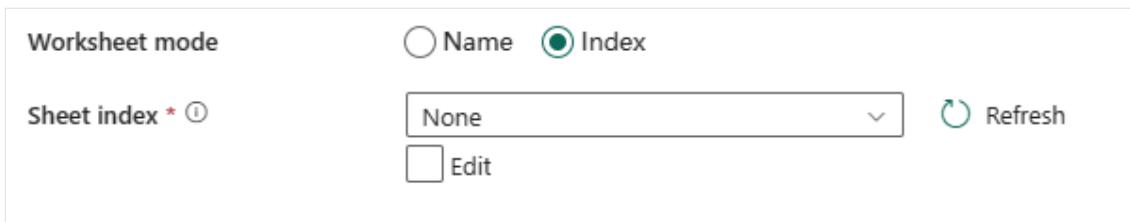
Excel as source

After choosing Excel format, following properties are shown up.

- **Worksheet mode:** Select the worksheet mode that you want to use to read Excel data. Choose **Name** or **Index**.
 - **Name:** When you choose **Name**, in **Sheet name** section, select the Excel worksheet name to read data, or select **Edit** to specify the worksheet name manually. If you point to a folder or multiple files, make sure this particular worksheet exists in all those files.



- **Index:** When you choose **Index**, in **Sheet index** section, select the Excel worksheet index to read data, or select **Edit** to specify the worksheet name manually. The data read start from 0. If there is worksheet added or deleted from excel file, the index of existed worksheets will change automatically.



After selecting **Settings** in **File format** section, following properties are shown up in the pop-up **File format settings** dialog box.



- **Compression type:** The compression codec used to read Excel files. You can choose from **None**, **bzip2**, **gzip**, **deflate**, **ZipDeflate**, **TarGzip** or **tar** type in the drop-down list.
- **Compression level:** Specify the compression ratio when you select a compression type. You can choose from **Optimal** or **Fastest**.
 - **Fastest:** The compression operation should complete as quickly as possible, even if the resulting file is not optimally compressed.
 - **Optimal:** The compression operation should be optimally compressed, even if the operation takes a longer time to complete. For more information, see [Compression Level topic](#).
- **Range:** The cell range in the given worksheet to locate the selective data, e.g.:
 - Not specified: reads the whole worksheet as a table from the first non-empty row and column.
 - **A3:** reads a table starting from the given cell, dynamically detects all the rows below and all the columns to the right.
 - **A3:H5:** reads this fixed range as a table.
 - **A3:A3:** reads this single cell.
- **Null value:** Specifies the string representation of null value. The default value is empty string.

- **First row as header:** Specifies whether to treat the first row in the given worksheet/range as a header line with names of columns. Unselected by default.

Table summary

Excel as source

The following properties are supported in the copy activity **Source** section when using Excel format.

Name	Description	Value	Required	JSON script property
File format	The file format that you want to use.	Excel	Yes	type (<i>under datasetSettings</i>): Excel
Worksheet mode	The worksheet mode that you want to use to read Excel data.	- Name - Index	Yes	- sheetName - sheetIndex
Compression type	The compression codec used to read Excel files.	Choose from: None bzip2 gzip deflate ZipDeflate TarGzip tar	No	type (<i>under compression</i>): bzip2 gzip deflate ZipDeflate TarGzip tar
Compression level	The compression ratio. Allowed values are Optimal or Fastest.	Optimal or Fastest	No	level (<i>under compression</i>): Fastest Optimal
Range	The cell range in the given worksheet to locate the selective data.	<your cell range>	No	range
Null value	The string representation of null value.	<your null value> empty string (by default)	No	nullValue

Name	Description	Value	Required	JSON script property
First row as header	Specifies whether to treat the first row in the given worksheet/range as a header line with names of columns.	Selected or unselected	No	firstRowAsHeader: true or false (default)

Next steps

[Connectors overview](#)

Connect to an Excel workbook in dataflows

Article • 05/23/2023

You can connect to Excel workbooks in Dataflow Gen2 using the Excel connector provided by Data Factory in Microsoft Fabric.

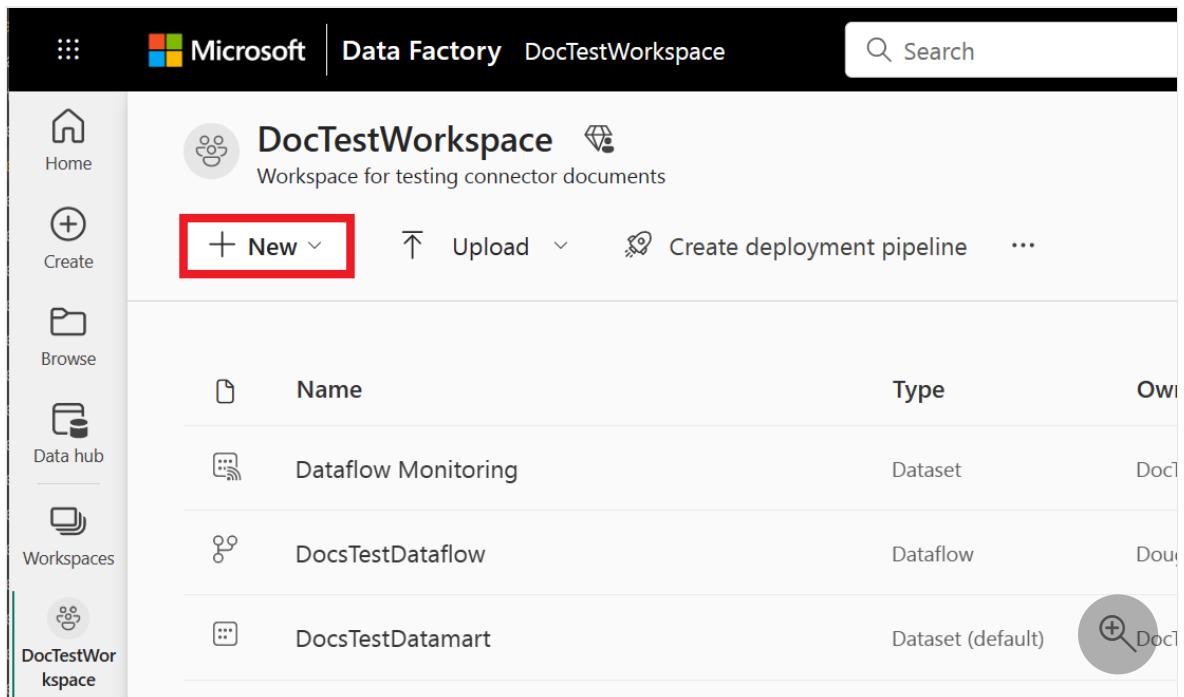
ⓘ Important

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Connect to an Excel workbook

To connect to an Excel workbook in a dataflow:

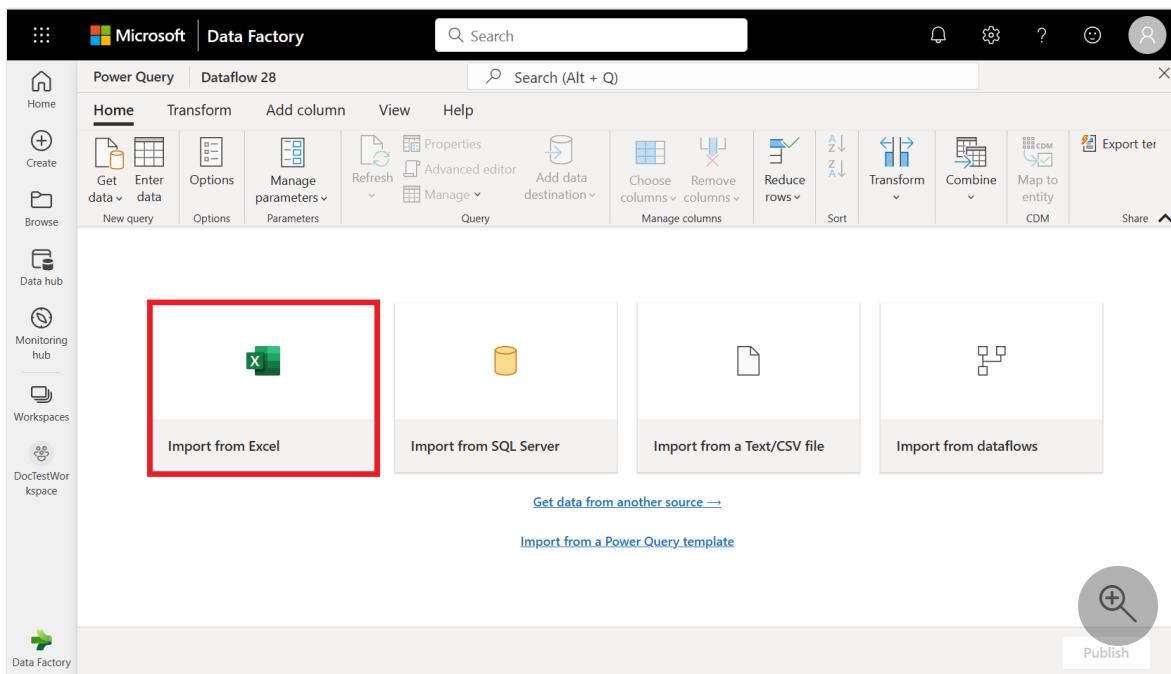
1. From your workspace, select **New > Dataflow Gen2 (Preview)** to create a new dataflow.



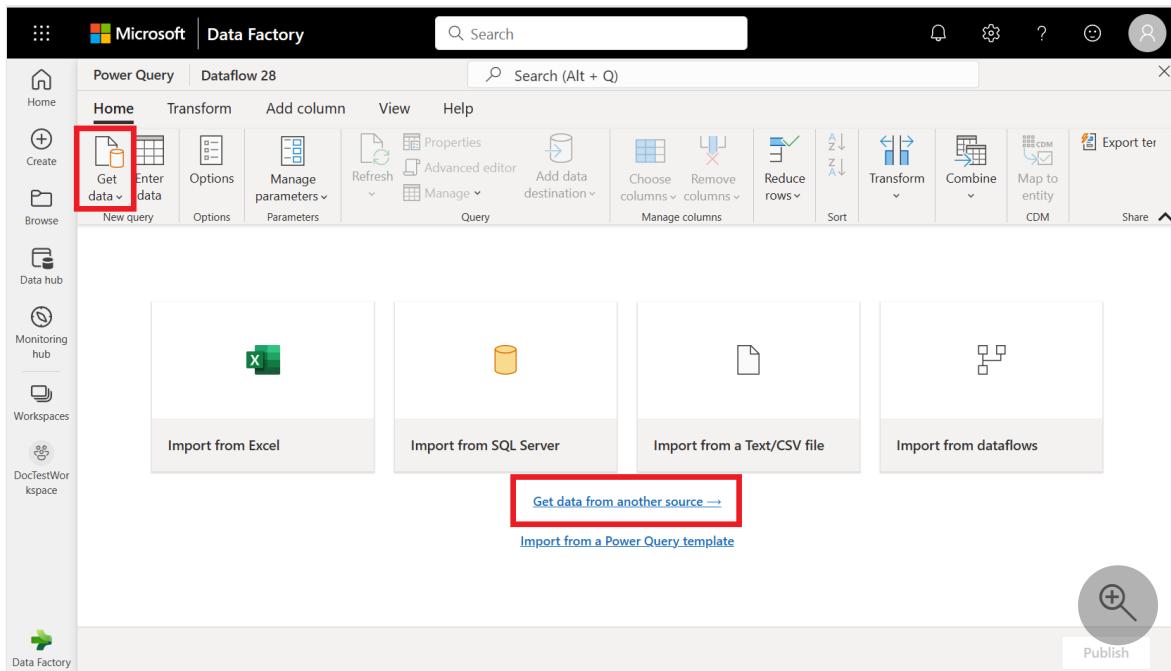
The screenshot shows the Microsoft Data Factory workspace interface. The top navigation bar includes the Microsoft logo, the 'Data Factory' title, and the workspace name 'DocTestWorkspace'. A search bar is also present. On the left, a sidebar menu lists 'Home', 'Create' (with a '+' icon), 'Browse', 'Data hub', 'Workspaces' (which is selected and highlighted with a green vertical bar), and 'DocTestWor kspace'. The main content area displays a table of existing resources. The first row of the table has a red box around the '+ New' button. The table columns are 'Name', 'Type', and 'Owner'. The data rows are:

Name	Type	Owner
Dataflow Monitoring	Dataset	DocT
DocsTestDataflow	Dataflow	DocT
DocsTestDatamart	Dataset (default)	DocT

2. In Power Query, select **Import from Excel**.



You can also select **Get data** in the Power Query ribbon, or select **Get data from another source**. If you choose either of these selections, select the **File** category, and then select **Excel workbook** from the **Choose data source** dialog.



Choose data source

Select a connector or directly drag a file from your computer.

All categories

File

Database

Power Platform

Azure

Online services

Other



Excel workbook
File



Text/CSV
File



XML
File



Folder
File



PDF
File



Parq
File

3. In **Connect to data source**, under **Connection settings**, enter the path or URL for your Excel workbook. If the workbook is stored online, select **Link to file**. Then either enter the URL to your online Excel workbook, or select the **Browse OneDrive** to navigate to a workbook on OneDrive. If the workbook is stored locally, either enter the path and filename to your local Excel workbook, or select **Upload file** and drag the file to the page or select **Browse** to browse to the location of the local Excel workbook.

4. If you're connecting to this data source for the first time, select the authentication type to use in **Authentication kind**, and then enter your credentials. The supported authentication types for this data source are:

- Anonymous
- Basic
- Organizational account

5. More information: [Connections and authentication in Power Query Online](#)

Connect to data source

The screenshot shows the 'Connect to data source' interface. On the left, there's a section for 'Excel workbook' with a file icon and a 'File' button. Below it is a link to 'Learn more'. On the right, under 'Connection settings', there are two radio buttons: 'Link to file' (selected) and 'Upload file (Preview)'. A field for 'File path or URL' is present with a placeholder 'Example: https://contoso-my.sharepoint.com/p...' and a 'Browse OneDrive...' button. Below this is the 'Connection credentials' section, which includes fields for 'Connection name' (set to 'Connection'), 'Data gateway' (set to '(none)'), and 'Authentication kind' (set to 'Anonymous'). There's also a magnifying glass icon with a plus sign.

6. Select **Next**.

7. In **Choose data**, select the data item that you want to transform, and then select **Create**.

The screenshot shows the 'Choose data' interface. On the left, there's a search bar and display options. Below that is a list of data sources: 'Excel workbook' (with 2 items) and 'Financial Sample 1' (which is checked). Under 'Financial Sample 1', there's a sub-item 'Sheet1'. To the right, the 'Financial Sample 1' table is displayed with columns: Segment, Country, Product, Discount Band, Units Sold, Manufacturing Price, Sale Price, and Gross Sale. The table contains 20 rows of data. At the bottom, there are 'Back', 'Cancel', and 'Create' buttons, along with a magnifying glass icon.

Advanced connector information

For more advanced information about connecting to your data source using the Excel workbook connector, go to [Excel](#).

Next steps

- [Excel format in Data Factory in Microsoft Fabric](#)

Parquet format in Data Factory in Microsoft Fabric

Article • 05/23/2023

This article outlines how to configure Parquet format in the data pipeline of Data Factory in Microsoft Fabric.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

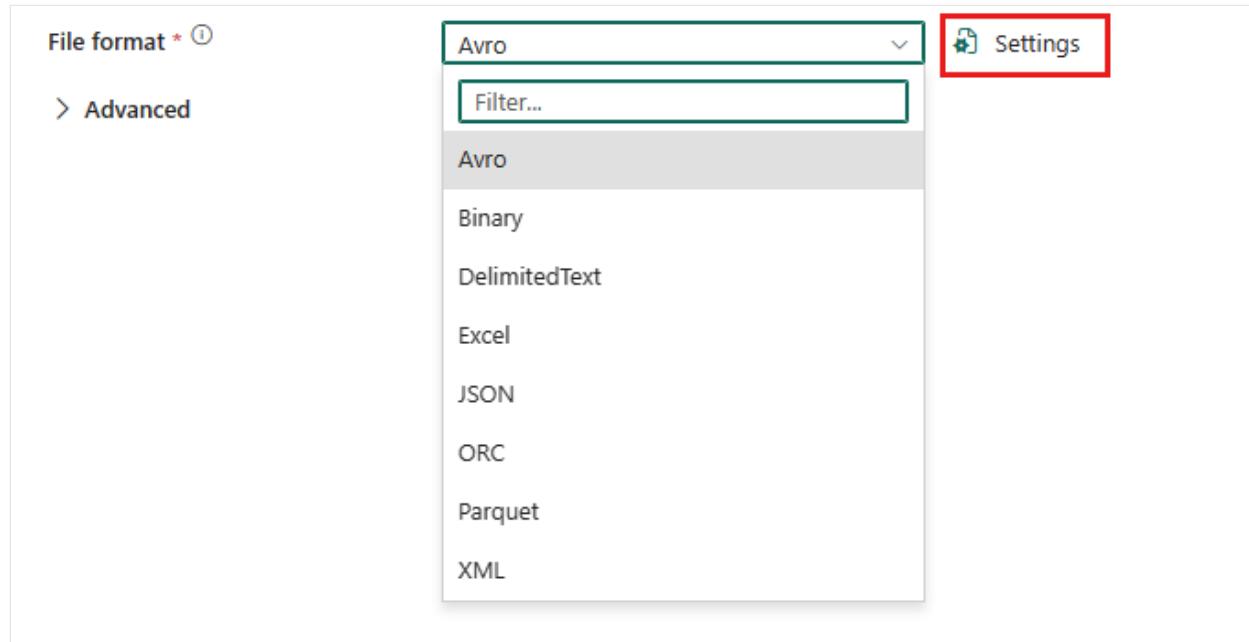
Supported capabilities

Parquet format is supported for the following activities and connectors as a source and destination.

Category	Connector/Activity
Supported connector	Amazon S3
	Azure Blob Storage
	Azure Data Lake Storage Gen1
	Azure Data Lake Storage Gen2
	Google Cloud Storage
	HTTP
Supported activity	Copy activity
	Lookup activity
	GetMetadata activity
	Delete activity

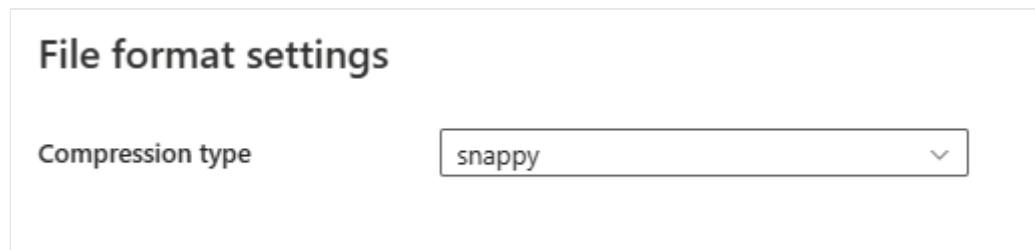
Parquet format in copy activity

To configure Parquet format, choose your connection in the source or destination of data pipeline copy activity, and then select **Parquet** in the drop-down list of **File format**. Select **Settings** for further configuration of this format.



Parquet format as source

After you select **Settings** in the **File format** section, the following properties are shown in the pop-up **File format settings** dialog box.



- **Compression type:** Choose the compression codec used to read Parquet files in the drop-down list. You can choose from **None**, **gzip (.gz)**, **snappy**, **Izo**, **Brotli (.br)**, **Zstandard**, **Iz4**, **Iz4frame**, **bzip2 (.bz2)**, or **Iz4hadoop**.

Parquet format as destination

After you select **Settings**, the following properties are shown in the pop-up **File format settings** dialog box.

File format settings

Enable Verti-Parquet ⓘ	<input checked="" type="checkbox"/>
Compression type	snappy

- **Enable Verti-Parquet:** Select this checkbox to optimize with Verti-Parquet technology.
- **Compression type:** Choose the compression codec used to write Parquet files in the drop-down list. You can choose from **None**, **gzip (.gz)**, **snappy**, **Izo**, **Brotli (.br)**, **Zstandard**, **Iz4**, **Iz4frame**, **bzip2 (.bz2)**, or **Iz4hadoop**.

Under **Advanced** settings in the **Destination** tab, the following Parquet format related properties are displayed.

- **Max rows per file:** When writing data into a folder, you can choose to write to multiple files and specify the maximum rows per file. Specify the maximum rows that you want to write per file.
- **File name prefix:** Applicable when **Max rows per file** is configured. Specify the file name prefix when writing data to multiple files, resulted in this pattern:
`<fileNamePrefix>_00000.<fileExtension>`. If not specified, the file name prefix is auto generated. This property doesn't apply when the source is a file based store or a partition option enabled data store.

Table summary

Parquet as source

The following properties are supported in the copy activity **Source** section when using the Parquet format.

Name	Description	Value	Required	JSON script property
File format	The file format that you want to use.	Parquet	Yes	type (<i>under datasetSettings</i>): Parquet

Name	Description	Value	Required	JSON script property
Compression type	The compression codec used to read Parquet files.	Choose from: None gzip (.gz) snappy lzo Brotli (.br) Zstandard lz4 lz4frame bzip2 (.bz2) lz4hadoop	No	compressionCodec:

Parquet as destination

The following properties are supported in the copy activity **Destination** section when using the Parquet format.

Name	Description	Value	Required	JSON script property
File format	The file format that you want to use.	Parquet	Yes	type (<i>under datasetSettings</i>): Parquet
Compression type	The compression codec used to write Parquet files.	Choose from: None gzip (.gz) snappy lzo Brotli (.br) Zstandard lz4 lz4frame bzip2 (.bz2) lz4hadoop	No	compressionCodec:

Name	Description	Value	Required	JSON script property
Max rows per file	When writing data into a folder, you can choose to write to multiple files and specify the maximum rows per file. Specify the maximum rows that you want to write per file.	<your max rows per file>	No	maxRowsPerFile
File name prefix	Applicable when Max rows per file is configured. Specify the file name prefix when writing data to multiple files, resulted in this pattern: <code><fileNamePrefix>_0000.</code> <code><fileExtension></code> . If not specified, the file name prefix is auto generated. This property doesn't apply when the source is a file based store or a partition option enabled data store.	<your file name prefix>	No	fileNamePrefix

Next steps

- [Connect to Parquet files in dataflows](#)
- [Connectors overview](#)

Connect to Parquet files in dataflows

Article • 05/23/2023

You can connect to Parquet files in Dataflow Gen2 using the Parquet connector provided by Data Factory in Microsoft Fabric.

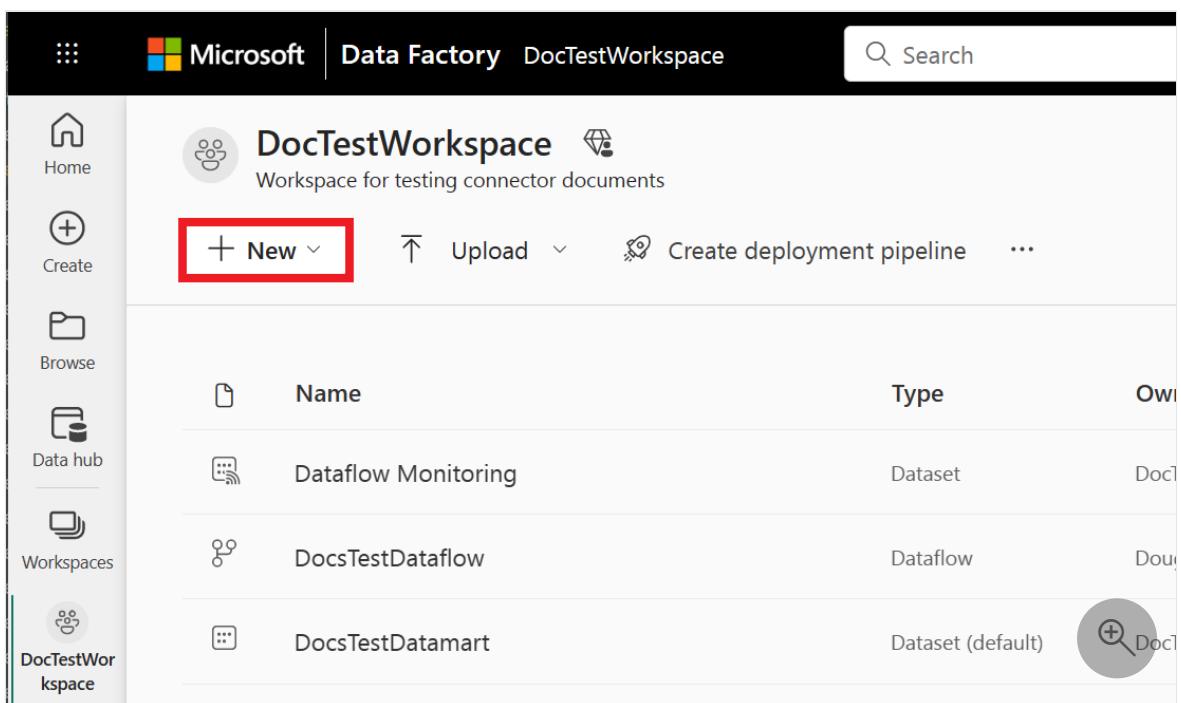
ⓘ Important

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Connect to Parquet files

To connect to Parquet files in a dataflow:

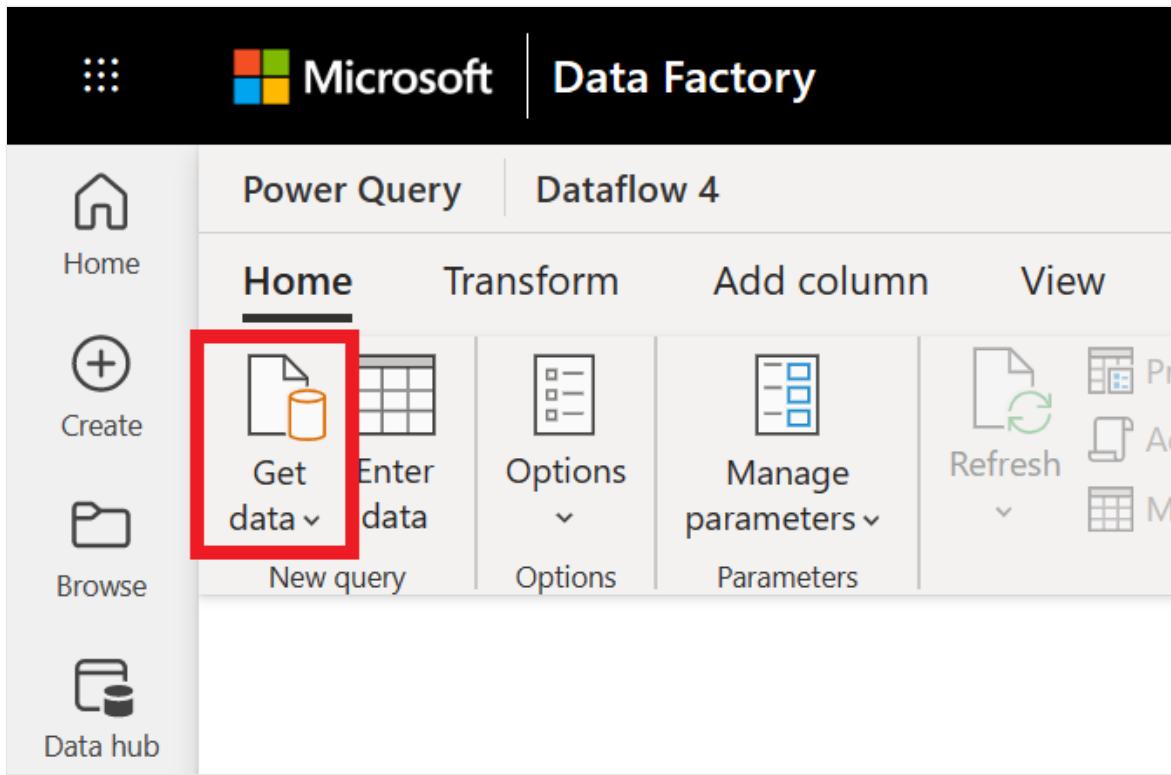
1. From your workspace, select **New > Dataflow Gen2 (Preview)** to create a new dataflow.



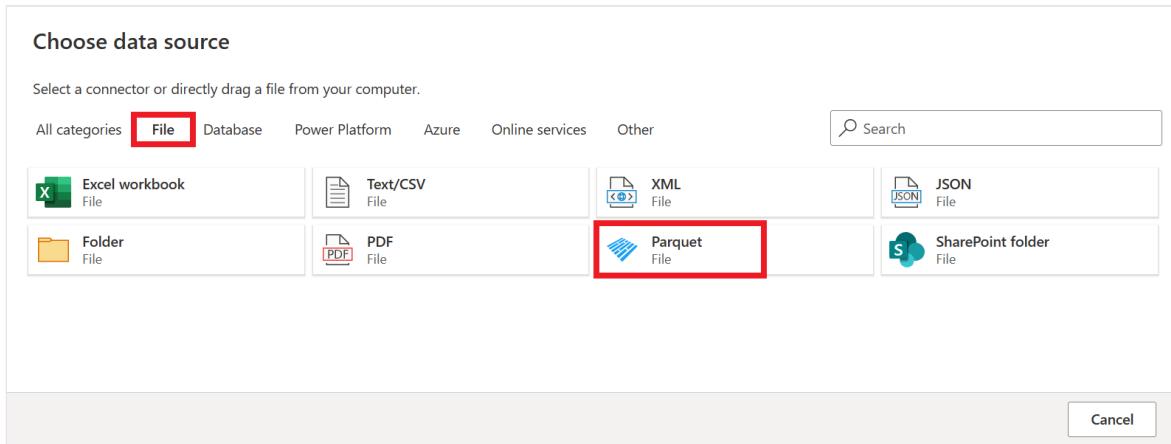
The screenshot shows the Microsoft Data Factory workspace interface. At the top, there is a navigation bar with icons for Home, Create, Browse, Data hub, Workspaces, and DocTestWorkspace. The 'DocTestWorkspace' section displays a workspace for testing connector documents. Below the navigation bar, there is a search bar and a 'New' button, which is highlighted with a red box. Other buttons include 'Upload', 'Create deployment pipeline', and a three-dot menu. A table below lists datasets and dataflows:

	Name	Type	Owning Workspace
Dataset	Dataflow Monitoring	Dataset	DocTestWorkspace
Dataflow	DocsTestDataflow	Dataflow	DocTestWorkspace
Dataset (default)	DocsTestDatamart	Dataset (default)	DocTestWorkspace

2. In Power Query, either select **Get data** in the ribbon or select **Get data from another source** in the current view.



3. From **Choose data source**, select the **File** category, and then select **Parquet**.



4. In **Connect to data source**, under **Connection settings**, enter the file path and filename or the URL of the online file location.
5. Select the authentication type to use in **Authentication kind**, and then enter your credentials. More information: [Connections and authentication in Power Query Online](#)

Connect to data source

The screenshot shows the 'Connect to data source' dialog. On the left, there's a 'Parquet File' icon. The main area has two sections: 'Connection settings' and 'Connection credentials'. In 'Connection settings', there's a 'File path or URL *' field with an example URL and a 'Create new connection' dropdown. In 'Connection credentials', there's a 'Connection name' dropdown set to 'Connection', a 'Data gateway *' dropdown set to '(none)', and an 'Authentication kind' dropdown set to '(none)'. There are also refresh and search icons.

6. Select **Next** to continue to the Power Query editor, where you can then begin to transform your data.

The screenshot shows the Microsoft Power Query Editor interface. The top navigation bar includes 'Power Query' and 'Dataflow 12'. The left sidebar has options like 'Home', 'Create', 'Browse', 'Data hub', 'Monitoring hub', 'Workspaces', and 'My workspace'. The main area shows a 'Queries [1]' list with one item: 'Contoso Financial ...' which is connected to a 'Source' (Parquet file). The preview pane shows a table with columns: Segment, Country, Product, Discount Band, Units Sold, Manufacturing Price, Sale Price, Gross Sales, Discounts. The bottom right corner has 'Query settings' and 'Applied steps' sections, and a 'Data destination' section indicating 'No data destination'.

Advanced connector information

For more advanced information about connecting to your data using the parquet connector, go to [Parquet](#).

Next steps

- [How to configure Parquet format](#)

Google Cloud Storage connector Overview

Article • 05/23/2023

This Google Cloud Storage connector is supported in Data Factory for Microsoft Fabric with the following capabilities.

i 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/-)	None	Basic
Lookup activity	None	Basic
GetMetadata activity	None	Basic
Delete activity	None	Basic

Next steps

[How to create Google Cloud Storage connection](#)

[How to configure Google Cloud Storage in copy activity](#)

How to create Google Cloud Storage connection

Article • 05/23/2023

This article outlines the steps to create Google Cloud Storage connection.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported authentication types

This Google Cloud Storage connector supports the following authentication types for copy activity.

Authentication type	Copy	Dataflow Gen2
Basic	✓	n/a

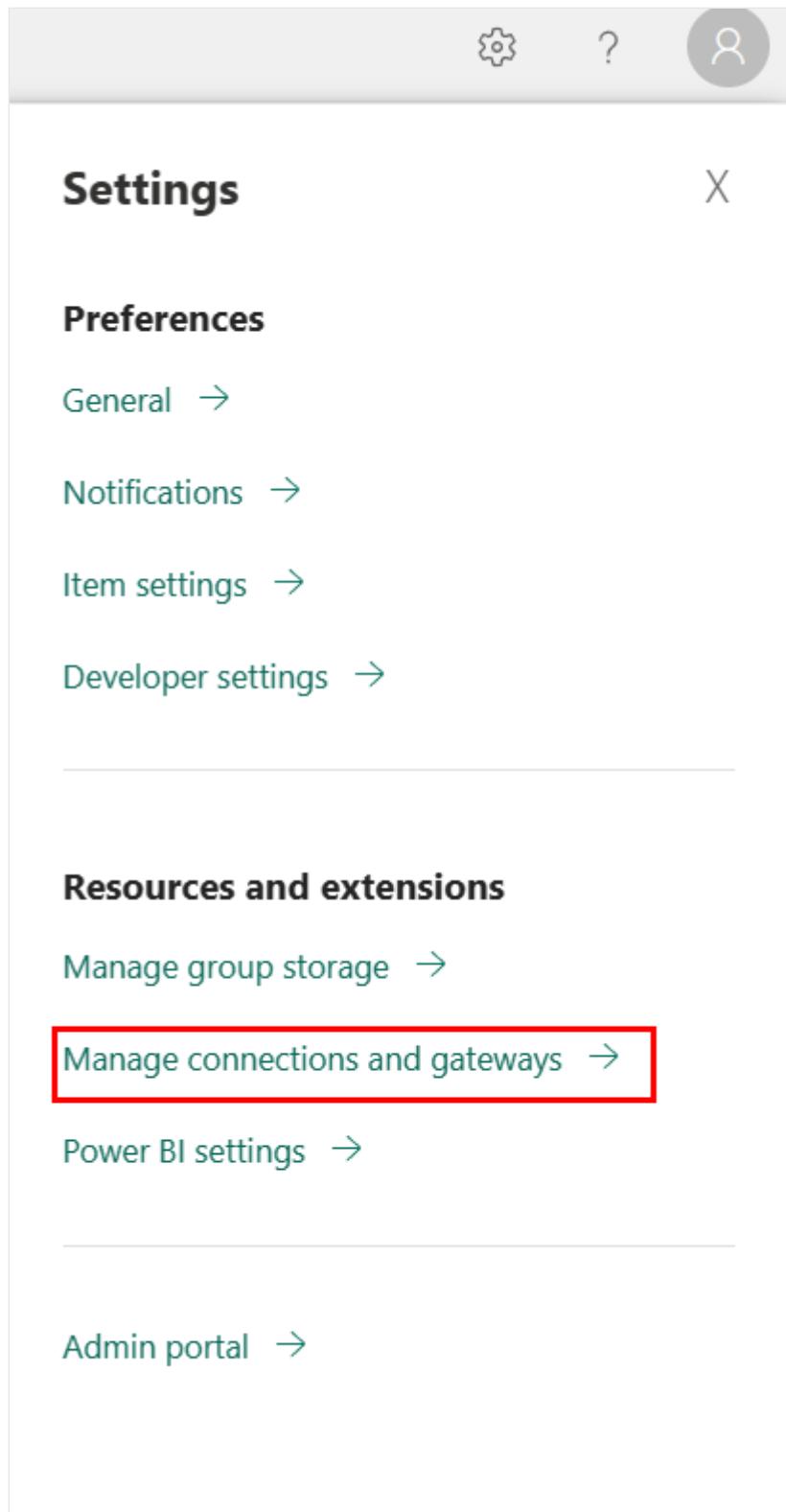
Prerequisites

The following prerequisites are required before you start:

- A Microsoft Fabric tenant account with an active subscription. [Create an account for free](#).
- A Microsoft Fabric enabled Workspace. [Create a workspace](#).

Go to manage gateways to create connection

1. From the page header in the Data Factory service, select **Settings**  > **Manage connections and gateways**



2. Select **New** at the top of the ribbon to add a new data source.

[+ New](#) [Get help](#)

Data (preview)

Connections

On-premises data gateways

Virtual network data gateways

Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)

The New connection pane now appears on the left side of the page.

The screenshot shows the Power BI Data (preview) interface. On the left, there's a list of existing connections: 'DemoBlob' (Azure Blob Storage, Clare). On the right, a 'New connection' pane is open, titled 'New connection'. It includes a note: 'Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".' Below this are three connection types: 'On-premises', 'Virtual network', and 'Cloud' (highlighted with a green border). The 'Connection name' field is empty, and the 'Connection type' dropdown is set to 'Cloud'. At the bottom are 'Create' and 'Close' buttons.

Setup connection

Step 1: Specify the new connection name, type, and URL

New connection

X

ⓘ Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



Connection name *

googlecloudstorage

Connection type *

Google Cloud Storage

Url

Ex: <https://storage.googleapis.com>

In the **New connection** pane, choose **Cloud**, and specify the following fields:

- **Connection name:** Specify a name for your connection.
- **Connection type:** Select **Google Cloud Storage** for your connection type.
- **Url:** Specify the custom GCS endpoint as <https://storage.googleapis.com>.

Step 2: Select and set your authentication

Under **Authentication method**, select your authentication from the drop-down list and complete the related configuration. This Google Cloud Storage connector supports the following authentication types.

- [Basic](#)

Authentication

Authentication method *

Basic

Basic

Basic authentication

- **Username:** ID of the secret access key. To find the access key and secret, go to [Prerequisites](#).
- **Password:** The secret access key itself.

Authentication

Authentication method *

Basic

Username *

< Your username >

Password *

< Your password >

Skip test connection

Step 3: Specify the privacy level that you want to apply

In the **General** tab, select the privacy level that you want apply in the **Privacy level** drop-down list. Three privacy levels are supported. For more information, go to [privacy levels](#).

Step 4: Create your connection

Select **Create**. Your creation is successfully tested and saved if all the credentials are correct. If not correct, the creation fails with errors.

Data (preview)				
Connections	On-premises data gateways	Virtual network data gateways	Status	Gateway cluster name
Cloud and data gateway connections for artifacts. Learn more about supported connections .				
Name ↓	Connection type	Users	Status	Gateway cluster name
Google Cloud Storage	Google Cloud Storage	< user name >		

Table summary

The following table contains connector properties that are supported in pipeline copy.

Name	Description	Required	Property	Copy
Connection name	A name for your connection.	Yes		✓
Connection type	Select a type for your connection. Select Google Cloud Storage .	Yes		✓
Url	The base Url to the Google Cloud Storage service.	Yes		✓
Authentication	Go to Authentication	Yes	Go to Authentication	

Authentication

The following properties in the table are the supported authentication types.

Name	Description	Required	Property	Copy
Basic				✓
- Username	ID of the secret access key. To find the access key and secret, go to Prerequisites .	Yes		
- Password	The secret access key itself.	Yes		

Next steps

- [How to configure Google Cloud Storage in copy activity](#)

How to configure Google Cloud Storage in copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in data pipeline to copy data from and to Google Cloud Storage.

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Prerequisites

The following setup is required on your Google Cloud Storage account:

1. Enable interoperability for your Google Cloud Storage account
2. Set the default project that contains the data you want to copy from the target GCS bucket.
3. Create a service account and define the right levels of permissions by using Cloud IAM on GCP.
4. Generate the access keys for this service account.

 Storage	Settings				
<ul style="list-style-type: none">  Browser  Transfer  Transfer Appliance  Settings 	<p>Project Access <u>Interoperability</u></p> <p>The Interoperability API lets you use HMAC authentication and lets Cloud Storage interoperate with tools written for other cloud storage systems. Turn on this API only if you require interoperable access for the current user. This API is enabled per project member, not per project. Each member can set a default project and maintain their own access keys.</p> <p>Request endpoint In the tools or libraries you use with other cloud storage systems like Amazon S3, change the request endpoint to use the Cloud Storage URI: https://storage.googleapis.com. Learn more</p> <p>Default project for interoperable access The default project is used with the Interoperability API for all create bucket and list bucket requests.</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px;">  <your project name> is your default project for interoperable access </div> <p>Interoperable storage access keys Use an access key to authenticate yourself when making requests to Cloud Storage. The key is linked to your Google user account. Learn more</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">Access Key</td> <td style="width: 50%;">Secret</td> </tr> <tr> <td><your account key></td> <td><your secret></td> </tr> </table> <div style="text-align: right; margin-top: 10px;"> Create a new key Delete </div>	Access Key	Secret	<your account key>	<your secret>
Access Key	Secret				
<your account key>	<your secret>				

Required permissions

To copy data from Google Cloud Storage, make sure you've been granted the following permissions for object operations: `storage.objects.get` and `storage.objects.list`.

In addition, `storage.buckets.list` permission is required for operations like testing connection and browsing from root.

For the full list of Google Cloud Storage roles and associated permissions, go to [IAM roles for Cloud Storage](#) on the Google Cloud site.

Supported format

Google Cloud Storage supports the following file formats. Refer to each article for format-based settings.

- Avro format
- [Binary format](#)
- [Delimited text format](#)
- [Excel format](#)
- JSON format
- ORC format

- Parquet format
- XML format

Supported configuration

For the configuration of each tab under copy activity, go to the following sections respectively.

- General
- Source
- Mapping
- Settings

General

For General tab configuration, go to [General](#).

Source

The following properties are supported for Google Cloud Storage under the **Source** tab of a copy activity.

The screenshot shows the 'Source' tab of the Azure Data Factory Copy Activity configuration pane. The tabs at the top are General, Source (selected), Destination, Mapping, and Settings. The 'Source' tab has a red notification badge with the number '1' in the top right corner. The configuration area includes:

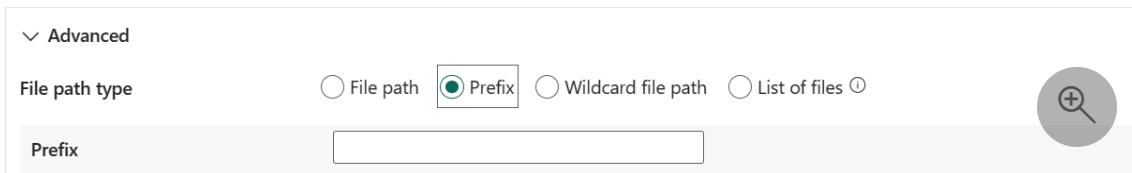
- Data store type:** Radio buttons for Workspace (unchecked), External (checked), and Sample dataset (unchecked).
- Connection:** A dropdown menu showing 'Google Cloud Storage' (selected) with a refresh icon, a 'Test connection' button, and 'Edit' and 'New' buttons.
- File path ***: Fields for 'Bucket' (text input: 'Bucket'), 'Directory' (text input: '/'), and 'File name' (text input: ''). Buttons include 'Browse' (with a dropdown arrow), 'Preview data', and 'File settings' (with a magnifying glass icon).
- Advanced** section:
 - File path type:** Radio buttons for File path (checked), Prefix, Wildcard file path, and List of files (with a help icon).
 - Recursively:** A checked checkbox.
 - Delete files after completion:** An unchecked checkbox.
 - Max concurrent connections:** A text input field with a placeholder '(0)' and a search icon.

The following properties are **required**:

- **Data store type:** Select **External**.
- **Connection:** Select a **Google Cloud Storage** connection from the connection list. If no connection exists, then create a new Google Cloud Storage connection by selecting **New**.
- **File path:** Select **Browse** to choose the file that you want to copy, or fill in the path manually.
- **File settings:** Select **File settings** to configure the file format. For settings of different file formats, refer to [Supported format](#) for detailed information.

Under **Advanced**, you can specify the following fields:

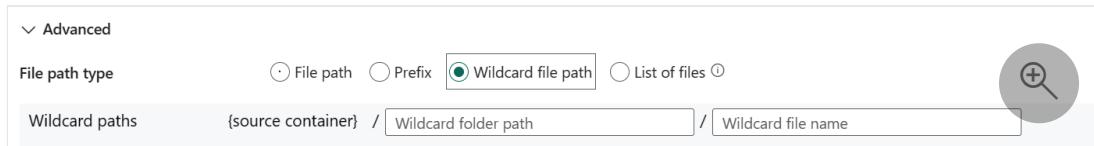
- **File path type:** You can choose **File path**, **Prefix**, **Wildcard file path**, or **List of files** as your file path type. The configuration of each of these settings is :
 - **File path:** If you choose this type, the data can be copied from the given bucket or folder/file path specified in **File path**.
 - **Prefix:** Prefix for the GCS key name under the given bucket configured to filter source GCS files. GCS keys whose names start with `given_bucket/this_prefix` are selected. It utilizes GCS's service-side filter, which provides better performance than a wildcard filter.



- **Wildcard file path:** Specify the folder or file path with wildcard characters under your given bucket to filter your source folders or files.

Allowed wildcards are: `*` (matches zero or more characters) and `?` (matches zero or single character). Use `^` to escape if your folder name has wildcard or this escape character inside. For more examples, go to [Folder and file filter examples](#).

- **Wildcard folder path:** Specify the folder path with wildcard characters under the given bucket to filter source folders.



- **Wildcard file name:** Specify the file name with wildcard characters under the given bucket and folder path (or wildcard folder path) to filter source files.
- **List of files:** Indicates to copy a given file set. Point to a text file that includes a list of files you want to copy, one file per line, which is the relative path to the path configured in **File path**.

When you're using this option, don't specify a file name. For more examples, go to [File list examples](#).

File path type

Path to file list

File path Prefix Wildcard file path List of files ⓘ

Browse

- **Recursively:** Indicates whether the data is read recursively from the subfolders or only from the specified folder. Note that when this checkbox is selected, and the destination is a file-based store, an empty folder or subfolder isn't copied or created at the destination.
- **Delete files after completion :** Indicates whether the binary files are deleted from the source store after successfully moving to the destination store. The file deletion is per file, so when a copy activity fails, you'll note that some files have already been copied to the destination and deleted from the source, while others are still remaining on source store. This property is only valid in the binary files copy scenario.
- **Max concurrent connection:** The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.

Mapping

For **Mapping** tab configuration, see [Configure your mappings under mapping tab](#). If you choose Binary as your file format, mapping will not be supported.

Settings

For the **Settings** tab configuration, go to [Configure your other settings under settings tab](#).

Table summary

The following tables contain more information about the copy activity in Google Cloud Storage.

Source information

Name	Description	Value	Required	JSON script property
------	-------------	-------	----------	----------------------

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the source data store.	<your connection>	Yes	connection
File path	If you choose this type, the data can be copied from the given bucket or folder/file path specified in File path .	Yes		container fileName
File path type	The file path type that you want to use.	<ul style="list-style-type: none"> • File path • Prefix • Wildcard folder path • List of files 	No	<ul style="list-style-type: none"> • prefix • wildcardFolderPath, wildcardFileName • path to file list
Recursively	Indicates whether the data is read recursively from the subfolders or only from the specified folder. Note that when this checkbox is selected, and the destination is a file-based store, an empty folder or subfolder isn't copied or created at the destination.	Selected or unselect	No	recursive

Name	Description	Value	Required	JSON script property
Delete files after completion	Indicates whether the binary files will be deleted from the source store after successfully moving to the destination store. The file deletion is per file, so when copy activity fails, you'll note some files have already been copied to the destination and deleted from the source, while others are still remaining on the source store. This property is only valid in binary files copy scenario.	Selected or unselect	No	deleteFilesAfterCompletion
Max concurrent connection	The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.	<max concurrent connections>	No	maxConcurrentConnections

Next steps

[How to create Google Cloud Storage connection](#)

Connect to Google Cloud Storage in dataflows

Article • 05/23/2023

The Google Cloud Storage connector isn't currently supported in Dataflow Gen2.

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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Next steps

- [How to create a Google Cloud Storage connection](#)
- [How to configure Google Cloud Storage in copy activity in Data Factory in Microsoft Fabric](#)

HTTP connector overview

Article • 05/23/2023

The HTTP connector is supported in Data Factory for Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/destination)	None	Basic
Lookup activity	None	Basic

Next steps

- [How to create HTTP connection](#)
- [How to configure HTTP in a copy activity](#)
- [Connect to HTTP data in dataflows](#)

How to create an HTTP connection

Article • 05/23/2023

This article outlines the steps to create HTTP connection.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported authentication types

The HTTP connector supports the following authentication types for copy and Dataflow Gen2 respectively.

Authentication type	Copy	Dataflow Gen2
Anonymous	✓	✓
Basic	✓	✓
Organizational account		✓
Windows		✓

ⓘ Note

For information about an HTTP connection in Dataflow Gen2, go to [Connect to HTTP data in dataflows](#).

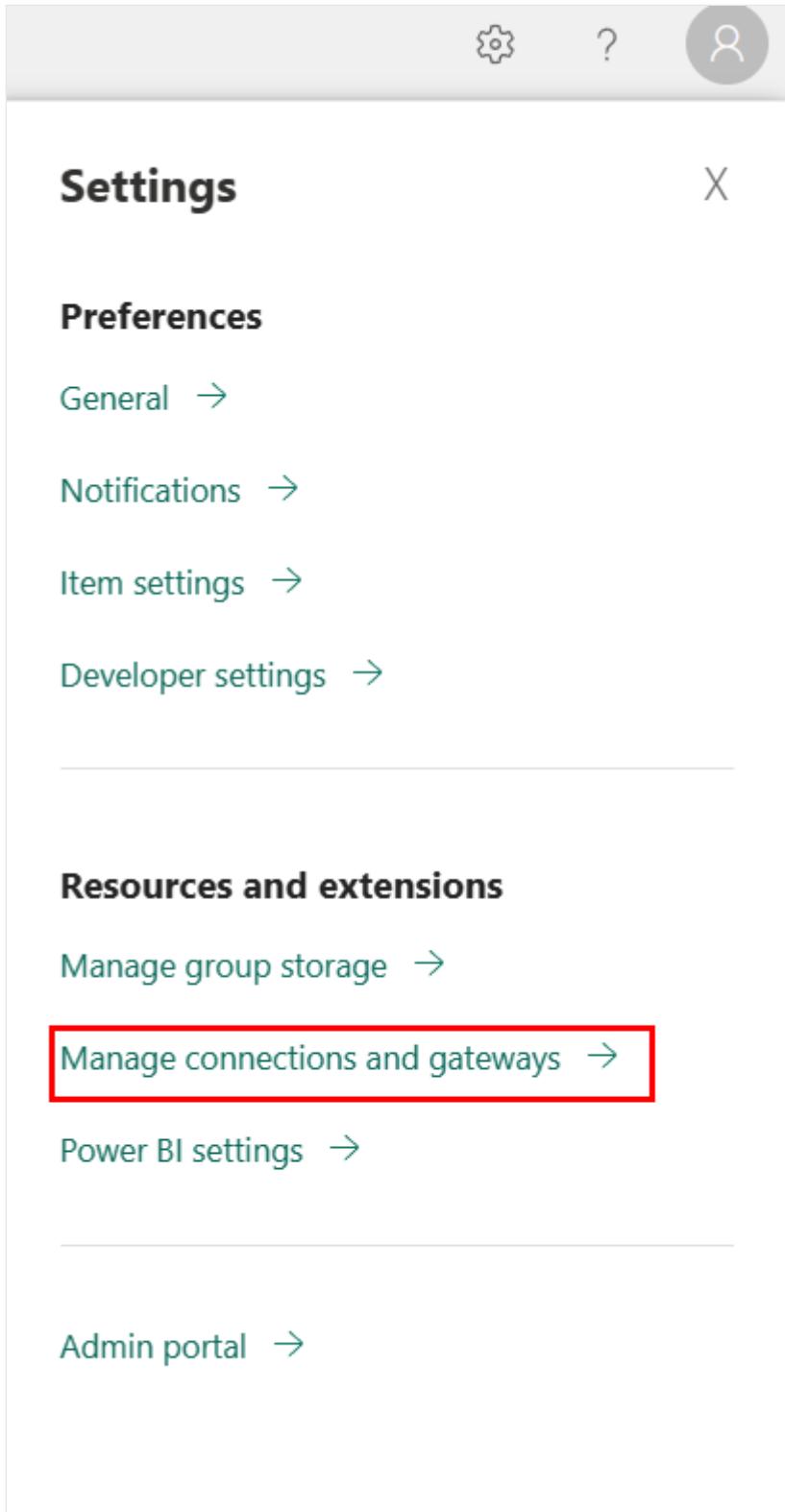
Prerequisites

The following prerequisites are required before you start:

- A Microsoft Fabric tenant account with an active subscription. [Create an account for free](#).
- A Microsoft Fabric enabled Workspace. [Create a workspace](#).

Go to manage gateways to create a connection

1. From the page header in the Data Factory service, select **Settings**  > **Manage connections and gateways**.



2. Select **New** at the top of the ribbon to add a new data source.

The screenshot shows the 'Data (preview)' page. At the top left, there is a red box highlighting the '+ New' button. To its right is a 'Get help' link with a question mark icon. Below the header, there are three tabs: 'Connections' (underlined), 'On-premises data gateways', and 'Virtual network data gateways'. A search bar with a magnifying glass icon is located on the right. The main content area displays a message: 'Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)'

The New connection pane opens on the left side of the page.

The screenshot shows the 'New connection' pane. On the left, there is a preview of the 'Data (preview)' page with a single connection listed: 'DemoBlob' (Azure Blob Storage, Clare). On the right, the 'New connection' pane has a title bar with a close button. It contains a note about cloud connections being supported for Data Pipelines and Kusto. Below the note are three connection type buttons: 'On-premises', 'Virtual network', and 'Cloud' (which is selected and highlighted with a green border). The main area has fields for 'Connection name *' and 'Connection type *'. At the bottom are 'Create' and 'Close' buttons, and a magnifying glass search icon.

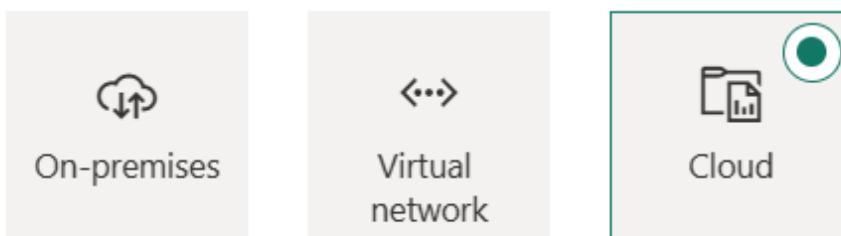
Setup connection

Step 1: Specify the new connection name, type, and URL

New connection



(i) Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



Connection name *

generichttp

Connection type *

Web



URL *

< The base URL to the web server >

In the **New connection** pane, choose **Cloud**, and then specify the following fields:

- **Connection name:** Specify a name for your connection.
- **Connection type:** Select Web for your connection type.
- **URL:** The base URL to the web server.

Step 2: Select and set your authentication

Under **Authentication method**, select your authentication from the drop-down list and complete the related configuration. This HTTP connector supports the following authentication types:

- Basic

Authentication

Authentication method *

Anonymous

Anonymous

Basic

OAuth2

Service Principal

Basic authentication

- **Username:** The user name to use to access the HTTP endpoint.
- **Password:** The password for specified username.

Authentication

Authentication method *

Basic

Username

< Your username >

Password

< Your password >



Skip test connection

Step 3: Specify the privacy level that you want to apply

In the **General** tab, select the privacy level that you want apply in the **Privacy level** drop-down list. Three privacy levels are supported. For more information, go to [privacy levels](#).

Step 4: Create your connection

Select **Create**. Your creation is successfully tested and saved if all the credentials are correct. If not correct, the creation fails with errors.

Data (preview)

Connections On-premises data gateways Virtual network data gateways

Cloud and data gateway connections for artifacts. [Learn more about supported connections](#).

Name ↑	Connection type	Users	Status	Gateway cluster name
generichttp ...	Web	< user name >	(status icon)	(status icon)

Table summary

The following connector properties in the table are supported in pipeline copy:

Name	Description	Required	Property	Copy
Connection name	A name for your connection.	Yes		✓/
Connection type	Select Web for your connection type.	Yes		✓/
URL	The base URL to the HTTP server.	Yes		✓/
Authentication	Go to Authentication	Yes		Go to Authentication
Privacy Level	The privacy level that you want to apply. Allowed values are Organizational , Privacy , Public	Yes		✓/

Authentication

The following properties in the table are the supported authentication types.

Name	Description	Required	Property	Copy
------	-------------	----------	----------	------

Name	Description	Required	Property	Copy
Basic				✓
- Username	The user name to use to access the HTTP endpoint.	Yes		
- Password	The password for specified username.		Yes	

Next steps

- [How to configure HTTP in copy activity](#)
- [Connect to HTTP data in dataflows](#)

How to configure HTTP in a copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in data pipeline to copy data from and to HTTP.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported format

HTTP supports the following file formats. Refer to each article for format-based settings.

- Avro format
- [Binary format](#)
- [Delimited text format](#)
- [Excel format](#)
- JSON format
- ORC format
- [Parquet format](#)
- XML format

Supported configuration

For the configuration of each tab under copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Mapping](#)
- [Settings](#)

ⓘ Note

Destination isn't supported in the HTTP connector.

General

Refer to the [General settings](#) guidance to configure the General settings tab.

Source

The following properties are supported for HTTP under the Source tab of a copy activity.

The screenshot shows the 'Source' tab configuration for an HTTP connection. The top navigation bar includes tabs for General, Source (which is selected and highlighted in green), Destination, Mapping, and Settings. Under the Source tab, there are several configuration sections:

- Data store type:** A radio button group where 'External' is selected (indicated by a green dot). Other options include 'Workspace' and 'Sample dataset'.
- Connection:** A dropdown menu showing 'generichttp'. To its right are three buttons: 'Refresh' (refresh icon), 'Edit' (pencil icon), and 'New' (plus icon).
- Connection type:** A dropdown menu showing 'HTTP'.
- Relative URL:** An input field containing an empty URL.
- Advanced:** A collapsed section containing:
 - Request method:** A dropdown menu showing 'GET'.
 - Additional headers:** An input field for defining headers.
 - Request body:** An input field for defining the request body.
 - Request timeout:** An input field for specifying the timeout.
 - Max concurrent connections:** An input field for specifying the maximum number of concurrent connections.
 - Skip line count:** An input field for specifying the number of lines to skip.
- Additional columns:** A section with a '+ New' button and a 'Add dynamic content [Alt+Shift+D]' link. To the right is a search icon.

The following three properties are required:

- **Data store type:** Select **External**.
- **Connection:** Select an HTTP connection from the connection list. If no connection exists, then create a new HTTP connection by selecting **New**.
- **Connection type:** Select **HTTP**.
- **Relative URL:** A relative URL to the resource that contains the data. When this property isn't specified, only the URL that's specified in the connection definition is

used. The HTTP connector copies data from the combined URL: `/[relative URL specified]`.

- **File settings:** Select **File settings** to configure the file format. For settings of different file formats, refer to articles in [Supported format](#) for detailed information.

Under **Advanced**, you can specify the following fields:

- **Request method:** The HTTP method. Allowed values are **Get** (default) and **Post**.
- **Additional headers:** Additional HTTP request headers.
- **Request body:** The request body for the HTTP request.
- **Request timeout:** The timeout (the timespan value) for the HTTP request to get a response. This value is the timeout to get a response, not the timeout to read response data. The default value is 00:01:40.
- **Max concurrent connections:** The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.
- **Skip line count:** The number of non-empty rows to skip when reading data from input files.
- **Additional columns:** Add additional data columns to store source files' relative path or static value. Expression is supported for the latter.

Name	Value
\$\$COLUMN	\$\$COLUMN

Failed More

Mapping

For **Mapping** tab configuration, see [Configure your mappings under mapping tab](#). If you choose Binary as your file format, mapping will not be supported.

Settings

For **Settings** tab configuration, go to [Configure your other settings under settings tab](#).

Table summary

The following table contains more information about the copy activity in HTTP.

Source

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the source data store.	<your connection>	Yes	connection
Connection type	The connection of your source data.	<connection of your source>	Yes	/
Relative URL	A relative URL to the resource that contains the data. When this property isn't specified, only the URL that's specified in the connection definition is used. The HTTP connector copies data from the combined URL: <code>/[relative URL specified]</code> .	<your relative url>	No	relativeUrl
Request method	The HTTP method. Allowed values are Get (default) and Post .	•GET •POST	No	requestMethod
Additional headers	Additional HTTP request headers.	<your additional headers>	No	additionalHeaders
Request body	The request body for the HTTP request.	<body for POST HTTP request>	No	requestBody

Name	Description	Value	Required	JSON script property
Request timeout	The timeout (the timespan value) for the HTTP request to get a response. This value is the timeout to get a response, not the timeout to read response data. The default value is 00:01:40.	timespan	No	requestTimeout
Max concurrent connections	The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.	<max concurrent connections>	No	maxConcurrentConnections
Skip line count	The number of non-empty rows to skip when reading data from input files.	<your skip line count>	No	skipLineCount
Additional columns	Add additional data columns to store source files' relative path or static value. Expression is supported for the latter.	<ul style="list-style-type: none"> • Name • Value 	No	additionalColumns: <ul style="list-style-type: none"> • name • value

Next steps

- [How to create an HTTP connection](#)

Connect to HTTP data in dataflows

Article • 05/23/2023

You can connect to HTTP in Dataflow Gen2 using the Web page connector provided by Data Factory in Microsoft Fabric.

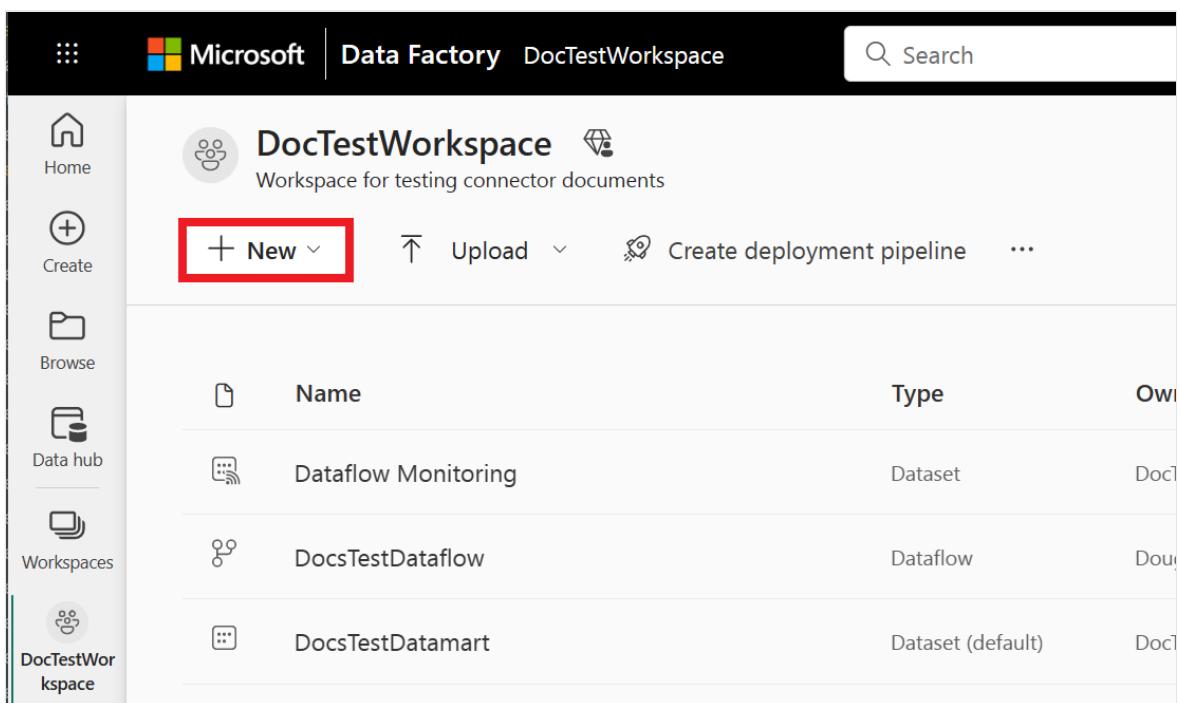
ⓘ Important

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Connect to HTTP data

To connect to HTTP data from a dataflow:

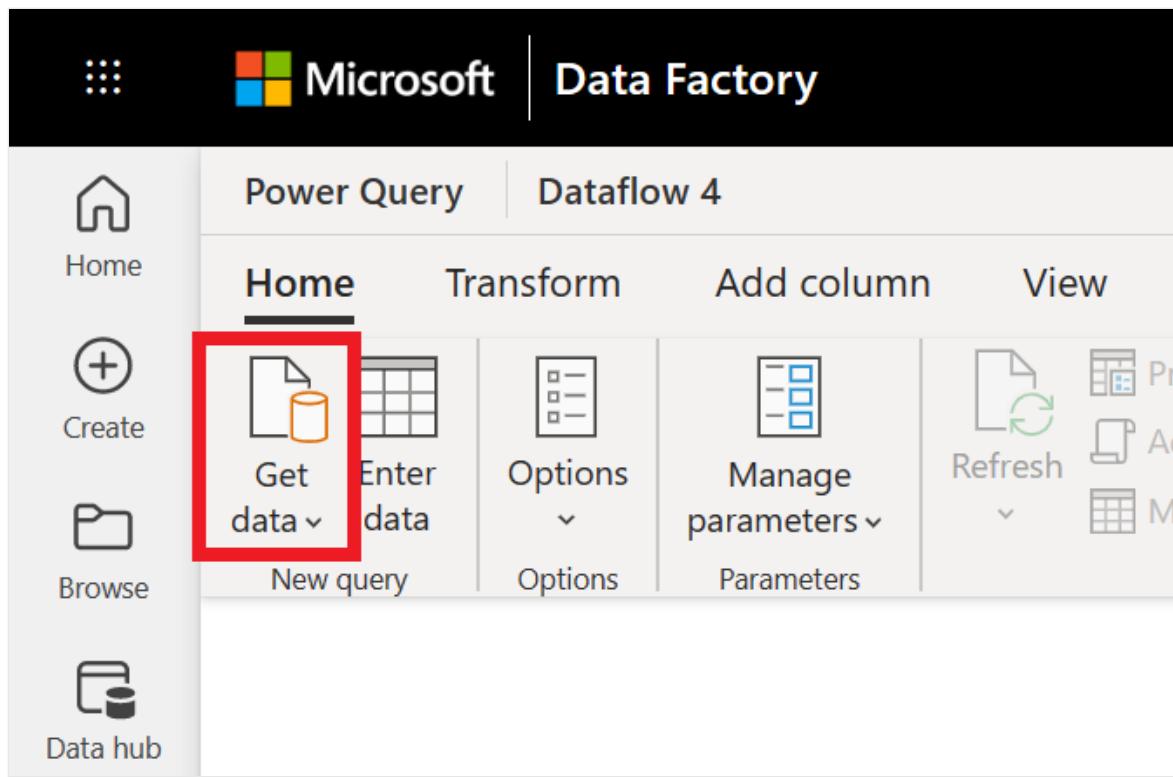
1. From your workspace, select **New > Dataflow Gen2 (Preview)** to create a new dataflow.



The screenshot shows the Microsoft Data Factory workspace interface. At the top, there is a navigation bar with icons for Home, Create, Browse, Data hub, Workspaces, and DocTestWorkspace. The 'DocTestWorkspace' section displays a workspace for testing connector documents. A red box highlights the '+ New' button in the top right corner of the workspace area. Below this, there is a table listing three items: 'Dataflow Monitoring' (Dataset, Owner: DocTest), 'DocsTestDataflow' (Dataflow, Owner: DocTest), and 'DocsTestDatamart' (Dataset (default), Owner: DocTest). The left sidebar also lists 'Home', 'Create', 'Browse', 'Data hub', 'Workspaces', and 'DocTestWorkspace'.

Name	Type	Owner
Dataflow Monitoring	Dataset	DocTest
DocsTestDataflow	Dataflow	DocTest
DocsTestDatamart	Dataset (default)	DocTest

2. In Power Query, either select **Get data** in the ribbon or select **Get data from another source** in the current view.



3. From **Choose data source**, select the **Other** category, and then select **Web page**.

A screenshot of the 'Choose data source' dialog. At the top, it says 'Choose data source' and 'Select a connector or directly drag a file from your computer.' Below is a navigation bar with tabs: All categories, File, Database, Power Platform, Azure, Online services, and Other (which is highlighted with a red box). The main area shows various connectors in a grid: Web API (Other), Web page (Other) (highlighted with a red box), Share (Other), Spark (Other), Odbc (Other), Acter (Other), Anaplan, BOF Core, and BitSic.

4. In **Connect to data source**, under **Connection settings**, enter the URL of the HTTP page.

5. Select the on-premises data gateway in **Data gateway**.

⚠ Note

The Web page connector requires a gateway. If you want to access a non-HTML resource without the gateway, use the Web API connector or one of the File connectors instead.

6. Select the authentication type to use in **Authentication kind**, and then enter your credentials. The authentication kinds supported by the Web page connector are:

- Anonymous
- Basic
- Organizational account
- Windows

More information: [Connections and authentication in Power Query Online](#)

Connect to data source

 **Web page**
Other
[Learn more](#)

Connection settings

URL *
Example: https://powerquery.microsoft.com/en-...

Connection credentials

Connection
Create new connection 

Connection name
Connection 

Data gateway *
(none) 
The Web Page connector requires a gateway. If you want to access a non-HTML resource without the gateway, please use the Web API connector or one of the File connectors instead.

Authentication kind
(none) 

7. Select **Next**.

8. In **Choose data**, select the data item that you want to transform, and then select **Next**.

Choose data

Search

Display options ▾

Web page [2]

Document

Table 0

Add table using examples

Back

Cancel

Create

Table 0

#	Label No.	Title	Performed by:	Duration
3	Vocal Sextette From Lucia	Kronold	Four minute	
126	Cello Solo, Nina	Anthony & Harrison	Four minute	
179	Are You Coming Home Tonight	A. Spalding	Four minute	
217	Garden Melody, Violin	Venetian Trio	Four minute	
232	Dream Of The Tyrolienne	Band	Four minute	
265	Superba Lancers - 1st And 2nd Figures	Band	Four minute	
266	Superba Lancers - 3rd And 4th Figures	Band	Four minute	
268	Petunia QuadRille - 1st And 2nd Figures	Band	Four minute	
274	Dublin Daisies Two-Step	Band	Four minute	
275	Lucky Moon Three-Step	Band	Four minute	
303	My Old Kentucky Home	Knickerbocker Quartet	Four minute	
321	He Was A Wonderful Man	Jones & Murray	Four minute	
363	Medley Of Emmett's Yodel Songs	Watson	Four minute	
...				

Advanced connector information

For more advanced information about connecting to your data using the Web connector, go to [Web](#).

Next steps

- [How to create an HTTP connection](#)
- [How to configure HTTP in a copy activity](#)

KQL Database connector overview

Article • 05/23/2023

This KQL Database connector is supported in Data Factory for Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/destination)	None	User Auth
Lookup activity	None	User Auth

Next steps

- [How to configure KQL Database in a copy activity](#)

How to configure KQL Database in a copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in a data pipeline to copy data from and to KQL Database.

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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported configuration

For the configuration of each tab under copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Destination](#)
- [Mapping](#)
- [Settings](#)

General

For General tab configuration, go to [General](#).

Source

The following properties are supported for KQL Database under the **Source** tab of a copy activity.

General Source ¹ Destination Mapping Settings

Data store type Workspace External Sample dataset

Workspace data store type

KQL Database Refresh

Use query Table Query

Table Refresh Preview data
 Edit

Advanced

Query timeout ⓘ

No truncation ⓘ

Additional columns + New
Add dynamic content [Alt+Shift+D]

The following properties are **required**:

- **Data store type:** Select **Workspace**.
- **Workspace data store type:** Select **KQL Database** from the data store type list.
- **KQL Database:** Select an existing KQL Database from the workspace.
- **Use query:** Select **Tables** or **Query**.
 - **Tables:** The name of the table that the connection refers to.

Use query Table Query

Table Refresh Preview data

- **Query:** Specify the read-only request given in a **KQL format**. Use the custom KQL query as a reference.

Use query Table Query

Query ⓘ Preview data

Under **Advanced**, you can specify the following fields:

- **Query timeout:** Specify the wait time before the query request times out. Default value is 10 minutes (00:10:00). Allowed max value is 1 hour (01:00:00).
- **No truncation:** Indicates whether to truncate the returned result set. By default result is truncated after 500,000 records or 64 MB. Truncation is strongly recommended for a proper behavior of the activity.
- **Additional columns:** Add additional data columns to store source files' relative path or static value. Expression is supported for the latter. To learn more, go to [Add additional columns during copy](#).

The screenshot shows a user interface for managing additional columns. At the top, there are buttons for '+ New', 'Delete', and 'Refresh'. Below this is a table with two columns: 'Name' and 'Value'. The 'Value' column contains a dropdown menu with the option '\$\$COLUMN'. A magnifying glass icon is located in the top right corner of the interface.

Destination

The following properties are supported for KQL Database under the **Destination** tab of a copy activity.

The screenshot shows the 'Destination' tab settings for a copy activity. The tabs at the top are General, Source (with a red notification badge), Destination (selected), Mapping, and Settings. Under 'Data store type', 'Workspace' is selected. 'Workspace data store type' is set to 'KQL Database'. 'KQL Database' is set to 'KQLDatabase'. 'Table' is set to 'None'. There are 'Edit' and 'Refresh' buttons next to the table selection. Under 'Advanced', 'Ingestion mapping name' is empty. Under 'Additional properties', there is a '+ New' button.

The following properties are **required**:

- **Data store type:** Select **Workspace**.
- **Workspace data store type:** Select **KQL Database** from the data store type list.
- **KQL Database:** Select an existing KQL Database from the workspace.
- **Table:** The name of the table that the connection refers to.

Under **Advanced**, you can specify the following fields:

- **Ingestion mapping name:** The name of a mapping that was pre-created and assigned to a KQL Database destination table in advance. To learn more, go to [Azure Data Explore data ingestion](#).
- **Additional properties:** A property bag that can be used for specifying any of the ingestion properties that aren't being set already by the KQL Database destination. Specifically, it can be useful for specifying ingestion tags.

Mapping

For **Mapping** tab configuration, go to [Configure your mappings under mapping tab](#).

Settings

For **Settings** tab configuration, go to [Configure your other settings under settings tab](#).

Table summary

The following tables contain more information about a copy activity in a KQL Database.

Source information

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	Workspace	Yes	/
Workspace data store type	Select KQL Database from the data store type list.	KQL Database	Yes	/
KQL Database	Select an existing KQL Database from the workspace.	<your KQL Database>	Yes	/

Name	Description	Value	Required	JSON script property
Use query	Select Tables or Query.	<ul style="list-style-type: none"> • Tables • Query 	No	table query
Query timeout	Specify the wait time before the query request times out. Default value is 10 minutes (00:10:00). Allowed maximum value is 1 hour (01:00:00).	timespan	No	queryTimeout
No truncation	Indicates whether to truncate the returned result set. By default, the result is truncated after 500,000 records or 64 MB. Truncation is strongly recommended for a proper behavior of the activity.	select or unselect	No	noTruncation: true or false
Additional columns	Add additional data columns to the store source files' relative path or static value. Expression is supported for the latter. Learn more from Add additional columns during copy .	<ul style="list-style-type: none"> • Name • Value 	No	additionalColumns: <ul style="list-style-type: none"> • name • value

Destination information

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	Workspace	Yes	/
Workspace data store type	Select KQL Database from the data store type list.	KQL Database	Yes	/
KQL Database	Select an existing KQL Database from the workspace.	<your KQL Database>	Yes	/
Table	The name of the table that the connection refers to.	<your table name>	Yes	table
Ingestion mapping name	The name of a mapping that was pre-created and assigned to KQL Database destination table in advance. Learn more from Azure Data Explore data ingestion .	<your ingestion mapping name>	Yes	ingestionMappingName

Name	Description	Value	Required	JSON script property
Additional properties	A property bag that can be used for specifying any of the ingestion properties that aren't being set already by the KQL Database destination. Specifically, it can be useful for specifying ingestion tags.	<ul style="list-style-type: none">• Name• Type• Value	Yes	additionalProperties

Next steps

- [KQL Database connector overview](#)

Lakehouse connector overview

Article • 05/23/2023

The Lakehouse connector is supported in Data Factory for Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/destination)	None	User Auth
Delete activity	None	User Auth

Next steps

- [How to configure Lakehouse in a copy activity](#)

How to configure Lakehouse in a copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in a data pipeline to copy data from and to the Fabric Lakehouse.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported format

Lakehouse supports the following file formats. Refer to each article for format-based settings.

- Avro format
- [Binary format](#)
- [Delimited text format](#)
- [Excel format](#)
- JSON format
- ORC format
- Parquet format
- XML format

Supported configuration

For the configuration of each tab under copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Destination](#)
- [Mapping](#)
- [Settings](#)

General

For the General tab configuration, go to [General](#).

Source

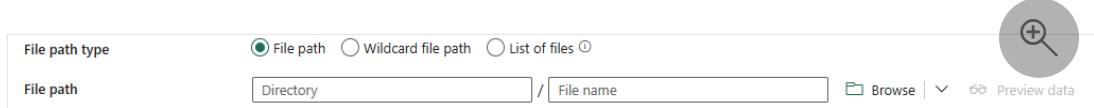
The following properties are supported for Lakehouse under the Source tab of a copy activity.

The screenshot shows the 'Source' tab of a copy activity configuration. The tabs at the top are General, Source (selected), Destination (with a red notification badge), Mapping, and Settings. The 'Data store type' is set to 'Workspace' (selected). Under 'Workspace data store type', 'Lakehouse' is selected from a dropdown. The 'Lakehouse' dropdown also includes 'Refresh', 'Open', and 'New' options. The 'Root folder' is set to 'Tables'. The 'Table name *' field is empty, with a 'Preview data' button and an 'Edit' checkbox. The 'Advanced' section is expanded, showing fields for 'Timestamp' and 'Version'. The 'Additional columns' section has a '+ New' button and a search icon. A red notification badge is visible on the 'Destination' tab.

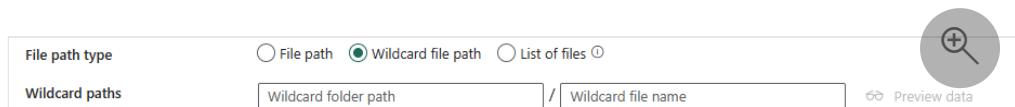
The following properties are required:

- **Data store type:** Select **Workspace**.
- **Workspace data store type:** Select **Lakehouse** from the data store type list.
- **Lakehouse:** Select an existing Lakehouse from the workspace. If none exists, then create a new Lakehouse by selecting **New**.
- **Root folder:** Select **Tables** or **Files**, which indicates the virtual view of the managed or unmanaged area in your lake. For more information, refer to [Lakehouse introduction](#).
 - If you select **Tables**:
 - **Table name:** Choose an existing table from the table list or specify a table name as the source.
 - Under **Advanced**, you can specify the following fields:
 - **Timestamp:** Specify to query an older snapshot by timestamp.
 - **Version:** Specify to query an older snapshot by version.
 - **Additional columns:** Add additional data columns to the store source files' relative path or static value. Expression is supported for the latter.

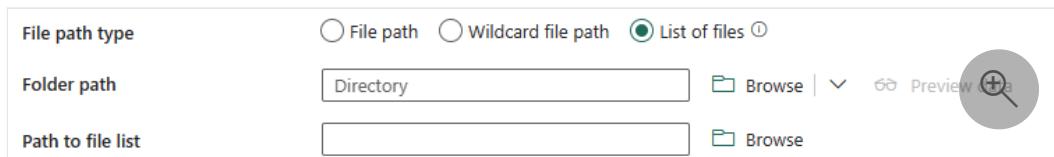
- If you select **Files**:
 - **File path type**: You can choose **File path**, **Wildcard file path**, or **List of files** as your file path type. The following list describes the configuration of each setting :



- **File path**: Select **Browse** to choose the file that you want to copy, or fill in the path manually.
- **Wildcard file path**: Specify the folder or file path with wildcard characters under your given Lakehouse unmanaged area (under Files) to filter your source folders or files. Allowed wildcards are: `*` (matches zero or more characters) and `?` (matches zero or single character). Use `\` to escape if your folder or file name has wildcard or this escape character inside.
- **Wildcard folder path**: The path to the folder under the given container. If you want to use a wildcard to filter the folder, skip this setting and specify that information in the activity source settings.
- **Wildcard file name**: The file name under the given Lakehouse unmanaged area (under Files) and folder path.



- **List of files**: Indicates to copy a given file set.
 - **Folder path**: Points to a folder that includes files you want to copy.
 - **Path to file list**: Points to a text file that includes a list of files you want to copy, one file per line, which is the relative path to the file path configured.



- **Recursively**: Indicates whether the data is read recursively from the subfolders or only from the specified folder. If enabled, all files in the input folder and its subfolders are processed recursively. This property doesn't apply when you configure your file path type as **List of files**.
- **File format**: Select your file format from the drop-down list. Select the **Settings** button to configure the file format. For settings of different file

formats, refer to articles in [Supported format](#) for detailed information.

- Under **Advanced**, you can specify the following fields:
 - **Filter by last modified**: Files are filtered based on the last modified dates. This property doesn't apply when you configure your file path type as **List of files**.
 - **Start time**: The files are selected if their last modified time is greater than or equal to the configured time.
 - **End time**: The files are selected if their last modified time is less than the configured time.
 - **Enable partition discovery**: For files that are partitioned, specify whether to parse the partitions from the file path and add them as extra source columns.
 - **Partition root path**: When partition discovery is enabled, specify the absolute root path in order to read partitioned folders as data columns.
 - **Max concurrent connections**: Indicates the upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.

Destination

The following properties are supported for Lakehouse under the **Destination** tab of a copy activity.

The screenshot shows the 'Destination' tab settings for a copy activity. The tabs at the top are General, Source, Destination (selected), Mapping, and Settings. The 'Destination' tab contains the following fields:

- Data store type:** Workspace (radio button selected)
- Workspace data store type:** Lakehouse (dropdown menu)
- Lakehouse:** Lakehouse (dropdown menu) with Refresh, Open, and New buttons
- Root folder:** Tables (radio button selected)
- Table name ***: (empty text input field) with Edit button (checkbox checked)
- Advanced** section (button expanded):
 - Max rows per file**: (empty text input field)
 - Table action:** Append (radio button selected) Overwrite (radio button)
 - Max concurrent connections**: (empty text input field) with a search icon

The following properties are **required**:

- **Data store type**: Select **Workspace**.

- **Workspace data store type:** Select **Lakehouse** from the data store type list.
- **Lakehouse:** Select an existing Lakehouse from the workspace. If none exists, then create a new Lakehouse by selecting **New**.
- **Root folder:** Select **Tables** or **Files**, which indicates the virtual view of the managed or unmanaged area in your lake. For more information, refer to [Lakehouse introduction](#).

- If you select **Tables**:

- **Table name:** Choose an existing table from the table list or specify a table name as the destination.

The screenshot shows a user interface for configuring a root folder. At the top, there are two radio buttons: 'Tables' (which is selected) and 'Files'. Below this, there is a text input field labeled 'Table name *' with a red asterisk indicating it is required. At the bottom right of the input field is a green 'Edit' button with a checkmark icon.

- Under **Advanced**, you can specify the following fields:
 - **Max rows per file:** Specify the maximum rows per file when writing data into Lakehouse.
 - **Table actions:** Specify the operation against the selected table.
 - **Append:** Append new values to existing table.
 - **Overwrite:** Overwrite the existing data and schema in the table using the new values. If this operation is selected, you can enable partition on your target table:
 - **Enable Partition:** This selection allows you to create partitions in a folder structure based on one or multiple columns. Each distinct column value (pair) is a new partition. For example, "year=2000/month=01/file". This selection supports insert-only mode and requires an empty directory in the destination.
 - **Partition column name:** Select from the destination columns in schemas mapping. Supported data types are string, integer, boolean, and datetime. Format respects type conversion settings under the **Mapping** tab.
 - **Max concurrent connections:** The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.

- If you select **Files**:

- **File path:** Select **Browse** to choose the file that you want to copy, or fill in the path manually.



- **File format:** Select your file format from the drop-down list. Select **Settings** to configure the file format. For settings of different file formats, refer to articles in [Supported format](#) for detailed information.
- Under **Advanced**, you can specify the following fields:
 - **Copy behavior:** Defines the copy behavior when the source is files from a file-based data store. You can choose **Add Dynamic content**, **None**, **Flatten hierarchy**, or **Preserve hierarchy** as your copy behavior. The configuration of each setting is :
 - **Add dynamic content:** To specify an expression for a property value, select **Add dynamic content**. This field opens the expression builder where you can build expressions from supported system variables, activity output, functions, and user-specified variables or parameters. For more information about the expression language, go to [Expressions and functions](#).
 - **None:** Choose this selection to not use any copy behavior.
 - **Flatten hierarchy:** All files from the source folder are in the first level of the destination folder. The destination files have autogenerated names.
 - **Preserve hierarchy:** Preserves the file hierarchy in the target folder. The relative path of a source file to the source folder is identical to the relative path of a target file to the target folder.

Advanced

Copy behavior ⓘ

Max concurrent connections ⓘ

Block size (MB) ⓘ

Metadata ⓘ

Preserve hierarchy

Add dynamic content

None

Flatten hierarchy

Preserve hierarchy

- **Max concurrent connections:** The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.

- **Block size (MB):** Specify the block size in MB when writing data to Lakehouse. Allowed value is between 4 MB and 100 MB.
- **Metadata:** Set custom metadata when copying to the destination data store. Each object under the `metadata` array represents an extra column. The `name` defines the metadata key name, and the `value` indicates the data value of that key. If [preserve attributes feature](#) is used, the specified metadata will union/overwrite with the source file metadata. The allowed data values are:
 - `$$LASTMODIFIED`: a reserved variable indicates to store the source files' last modified time. Apply to a file-based source with binary format only.
 - Expression
 - Static value

Metadata ⓘ		New	Delete
Name	Value		
<input type="text"/>	<input type="text"/> <code>\$\$LASTMODIFIED</code>	+	

Mapping

For the **Mapping** tab configuration, go to [Mapping](#). If you choose Binary as your file format, mapping isn't supported.

Settings

For the **Settings** tab configuration, go to [Settings](#).

Table summary

The following tables contain more information about a copy activity in Lakehouse.

Source information

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	Workspace	Yes	/

Name	Description	Value	Required	JSON script property
Workspace data store type	The section to select your workspace data store type.	Lakehouse	Yes	type
Lakehouse	The Lakehouse that you use as source.	<your Lakehouse>	Yes	workspaceId artifactId
Root folder	The type of the root folder.	* Tables * Files	No	rootFolder: Table or Files
Table name	The name of the table to read data.	<table name>	Yes when you select Tables in Root folder	table (under typeProperties -> source -> typeProperties)
Timestamp	The timestamp to query an older snapshot.	<timestamp>	No	timestampAsOf
Version	The version to query an older snapshot.	<version>	No	versionAsOf
Additional columns	Additional data columns to store source files' relative path or static value. Expression is supported for the latter.	* Name * Value	No	additionalColumns: * name * value
File path type	The type of the file path that you use.	* File path * Wildcard file path * List of files	Yes	/
File path	Copy from the path to a folder/file under source data store. Apply when choosing File path in File path type.	<file path>	Yes when choosing File path	* filePath * fileName

Name	Description	Value	Required	JSON script property
Wildcard paths	The folder path with wildcard characters under the source data store configured to filter source folders. Apply when choosing Wildcard file path in File path type .	<wildcard paths>	Yes when choosing Wildcard file path	* wildcardFolderPath * wildcardFileName
Folder path	Points to a folder that includes files you want to copy. Apply when choosing List of files in File path type .	<folder path>	No	folderPath
Path to file list	Indicates to copy a given file set. Point to a text file that includes a list of files you want to copy, one file per line, which is the relative path to the path configured. Apply when choosing List of files in File path type .	<path to file list>	No	fileListPath
Recursively	Process all files in the input folder and its subfolders recursively or just the ones in the selected folder. This setting is disabled when a single file is selected.	select or unselect	No	recursive: true or false
File format	The format of the file that you use.	<file format>	Yes	type (under formatSettings): DelimitedTextReadSettings

Name	Description	Value	Required	JSON script property
Filter by last modified	The files with last modified time in the range [Start time, End time) will be filtered for further processing. The time will be applied to UTC time zone in the format of 'yyyy-mm-ddThh:mm:ss.fffZ'.	* Start time * End time	No	modifiedDatetimeStart modifiedDatetimeEnd
Enable partition discovery	This property can be skipped which means no file attribute filter will be applied. This property doesn't apply when you configure your file path type as List of files .	Selected or unselected	No	enablePartitionDiscovery: true or false (default)
Partition root path	The absolute partition root path to read partitioned folders as data columns.	<your partition root path>	No	partitionRootPath
Max concurrent connections	The upper limit of concurrent connections established to the data store during the activity run. A value is needed only when you want to limit concurrent connections.	<max concurrent connections>	No	maxConcurrentConnections

Destination information

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	Workspace	Yes	/

Name	Description	Value	Required	JSON script property
Workspace data store type	The section to select your workspace data store type.	Lakehouse	Yes	type
Lakehouse	The Lakehouse that you use as destination.	<your Lakehouse>	Yes	workspaceId artifactId
Root folder	The type of the root folder.	* Tables * Files	Yes	rootFolder: Table or Files
Table name	The name of the table to which you want to write data.	<your table name>	Yes when you select Tables in Root folder	table (under <code>typeProperties</code> -> <code>sink</code> -> <code>typeProperties</code>)
Max rows per file	When writing data into a folder, you can choose to write to multiple files and specify the max rows per file.	<max rows per file>	No	maxRowsPerFile
Table action	Append new values to an existing table or overwrite the existing data and schema in the table using the new values.	* Append * Overwrite	No	tableActionOption: Append or Overwrite
Max concurrent connections	The upper limit of concurrent connections established to the data store during the activity run. Specify a value only when you want to limit concurrent connections.	<max concurrent connections>	No	maxConcurrentConnections
File path	Write data to the path to a folder/file under destination data store.	<file path>	No	* filePath * fileName

Name	Description	Value	Required	JSON script property
File format	The format of the file that you use.	<file format>	Yes	type (under <code>formatSettings</code>): <code>DelimitedTextWriteSettings</code>
Copy behavior	The copy behavior defined when the source is files from a file-based data store.	<ul style="list-style-type: none"> * Add dynamic content * None * Flatten hierarchy * Preserve hierarchy 	No	<code>copyBehavior:</code> <ul style="list-style-type: none"> * <code>FlattenHierarchy</code> * <code>PreserveHierarchy</code>
Block size (MB)	The block size in MB used to write data to Lakehouse. Allowed value is between 4 MB and 100 MB.	<block size>	No	<code>blockSizeInMB</code>
Metadata	The custom metadata set when copying to a destination.	<ul style="list-style-type: none"> * <code>\$\$LASTMODIFIED</code> * Expression * Static value 	No	<code>metadata</code>

Next steps

- [Lakehouse connector overview](#)

Microsoft 365 connector overview

Article • 05/23/2023

The Microsoft 365 connector is supported in Data Factory for Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/-)	None	Service principal

Next steps

- [How to create Microsoft 365 connection](#)
- [How to configure Microsoft 365 in copy activity](#)

How to create a Microsoft 365 connection

Article • 05/23/2023

This article outlines the steps to create a Microsoft 365 connection.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported authentication types

This Microsoft 365 connector supports the following authentication types for copy and Dataflow Gen2 respectively.

Authentication type	Copy	Dataflow Gen2
Service principal	✓	n/a

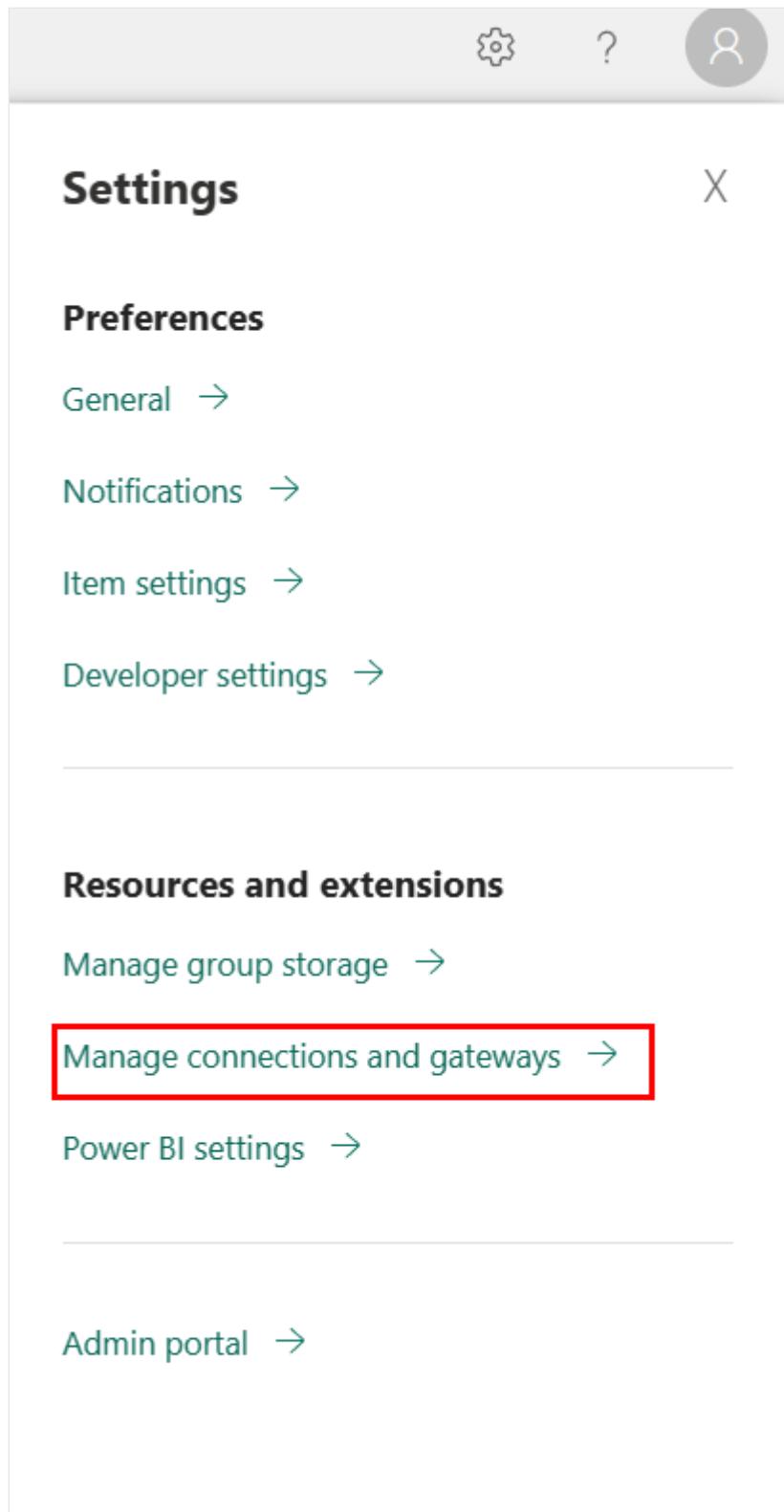
Prerequisites

The following prerequisites are required before you start:

- A Microsoft Fabric tenant account with an active subscription. [Create an account for free](#).
- A Microsoft Fabric enabled Workspace. [Create a workspace](#).

Go to Manage gateways to create connection

1. From the page header in Data Integration service, select **Settings**  > **Manage connections and gateways**.



2. Select **New** at the top of the ribbon to add a new data source.

The screenshot shows the 'Data (preview)' page. At the top left, there is a red box highlighting the '+ New' button. To its right is a 'Get help' link with a question mark icon. Below the header, there are three tabs: 'Connections' (underlined), 'On-premises data gateways', and 'Virtual network data gateways'. A search bar with a magnifying glass icon is located on the right. The main content area displays a message: 'Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)'

The **New connection** pane is displayed on the left side of the page.

The screenshot shows the 'New connection' pane. On the left, there is a preview of the 'Data (preview)' page with a table listing a single connection named 'DemoBlob'. On the right, the 'New connection' pane has a title bar with a close button. It contains a note about cloud connections being supported for Data Pipelines and Kusto. Below the note are three connection type buttons: 'On-premises', 'Virtual network', and 'Cloud' (which is highlighted with a green border). The main area has fields for 'Connection name *' and 'Connection type *'. At the bottom are 'Create' and 'Close' buttons, and a magnifying glass search icon.

Setup connection

Step 1: Specify the new connection name, type

New connection



i Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



On-premises



Virtual
network



Cloud

Connection name *

microsoft365

Connection type *

Microsoft365



In the **New connection** pane, choose **Cloud**, and then specify the following fields:

- **Connection name:** Specify a name for your connection.
- **Connection type:** Select **Microsoft 365** for your connection type.

Step 2: Select and set your authentication

Under **Authentication method**, select your authentication from the drop-down list and complete the related configuration. This Microsoft 365 connector supports the following authentication types.

[Service Principal](#)

Authentication

Authentication method *

OAuth2

OAuth2

Service Principal

Service Principal authentication

Authentication

Authentication method *

Service Principal

Tenant Id *

< your tenant ID >

Service principal ID *

< your application's client ID >

Service principal key *

.....

- **Tenant Id:** Your service principal tenant ID. Specify the tenant information under which your Azure AD web application resides.
- **Service principal ID:** Specify the application's client ID.
- **Service principal key:** Specify the application's key.

Step 3: Specify the privacy level that you want to apply

In the **General** tab, select the privacy level that you want apply in the **Privacy level** drop-down list. Three privacy levels are supported. For more information, go to [privacy levels](#).

Step 4: Create your connection

Select **Create**. Your creation is successfully tested and saved if all the credentials are correct. If not correct, the creation fails with errors.

Data (preview)

Connections On-premises data gateways Virtual network data gateways

Cloud and data gateway connections for artifacts. [Learn more about supported connections](#).

Name ↑	Connection type	Users	Status	Gateway cluster name
Microsoft365 ...	Microsoft365	<user name>	Connected	Microsoft365

Microsoft365 ... Microsoft365 <user name> Connected

Table summary

The connector properties in the following table are supported in pipeline copy:

Name	Description	Required	Property	Copy
Connection name	A name for your connection.	Yes		✓
Connection type	Select Microsoft-365 for your connection type.	Yes		✓
Authentication	Go to Authentication	Yes		Go to Authentication

Authentication

The following table contains the supported authentication type properties.

Name	Description	Required	Property	Copy

Name	Description	Required	Property	Copy
Service Principal				✓
- Tenant ID	Your service principal tenant ID. Specify the tenant information under which your Azure AD web application resides.	Yes		
- Service Principal ID	Specify the application's client ID.	Yes		
- Service Principal key	Specify the application's key.	Yes		

Next steps

- [How to configure Microsoft 365 in a copy activity](#)

How to configure Microsoft 365 in a copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in a data pipeline to copy data from Microsoft 365. For now, within a single copy activity, you can only ingest data from Microsoft 365 into Azure Blob Storage, Azure Data Lake Storage Gen1, and Azure Data Lake Storage Gen2.

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Prerequisites

To copy and transform data from Microsoft 365 into Azure, you need to complete the following prerequisite steps:

- Your Microsoft 365 tenant admin must complete on-boarding actions as described [here](#).
- Create and configure an Azure AD web application in Azure Active Directory. For instructions, go to [Create an Azure AD application](#).
- Make note of the following values, which you use to define the connection for Microsoft 365: Tenant ID. For instructions, go to [Get tenant ID](#).
- Application ID and Application key. For instructions, go to [Get application ID and authentication key](#). Add the user identity who will be making the data access request as the owner of the Azure AD web application (from the Azure AD web application > Settings > Owners > Add owner).
- The user identity must be in the Microsoft 365 organization you're getting data from and must not be a Guest user.

Approving new data access requests

If you're requesting data for this context for the first time (a combination of which data table is being accessed, which destination account is the data being loaded into, and which user identity is making the data access request), the copy activity status is displayed as **In Progress**. Only when you select the [Details link under Actions](#) will the status be displayed as **RequestingConsent**. A member of the data access approver group needs to approve the request in the Privileged Access Management before the data extraction can proceed.

Refer to the [frequently asked questions](#) on how the approver can approve the data access request. Refer to the [data connect integration with PAM](#) article for an explanation of the overall integration with Privileged Access Management, including how to set up the data access approver group.

Supported format

Microsoft 365 supports the following file formats. Refer to each article for format-based settings.

- [Binary format](#)

Supported configuration

For the configuration of each tab under copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Mapping](#)
- [Settings](#)

General

For the **General** tab configuration, go to [General](#).

Source

The following properties are supported for Microsoft 365 under the **Source** tab of a copy activity.

The screenshot shows the 'Source' tab configuration for a copy activity. The tabs at the top are General, Source (selected), Destination (with a red exclamation mark), Mapping, and Settings. Under the Source tab, the 'Data store type' is set to 'External'. The 'Connection' dropdown shows '111Microsoft365'. Below it, the 'Table' field is empty, and there is a 'Scope' dropdown set to 'All users or groups in the Microsoft ...'. The 'Output columns' section includes an 'Import schema' button and a table for defining output columns with columns for 'Column name', 'Type', and 'Description'. A search icon is also present in the output column table.

The following properties are **required**:

- **Data store type:** Select External.

- **Connection:** Select a **Microsoft 365** connection from the connection list. If no connection exists, then create a new Microsoft 365 connection by selecting **New**.
- **Table:** Name of the table to extract from **Microsoft 365**.

Under **Advanced**, you can specify the following fields:

- **Scope:** You can select **All users or groups in the Microsoft 365 tenant** or **Select groups from the Microsoft 365 tenant**

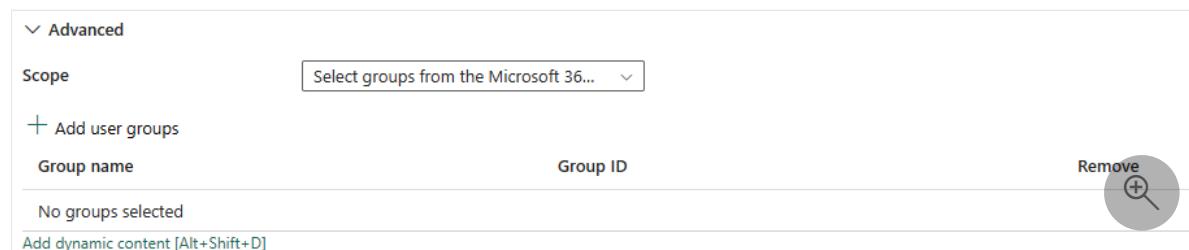
If you select **All users or groups in the Microsoft 365 tenant**, the scope filter is displayed.



- **Scope filter:** You can use a predicate expression that's applied on the entire tenant to filter the specific rows to extract from Microsoft 365. The predicate format should match the query format of Microsoft Graph APIs, for example

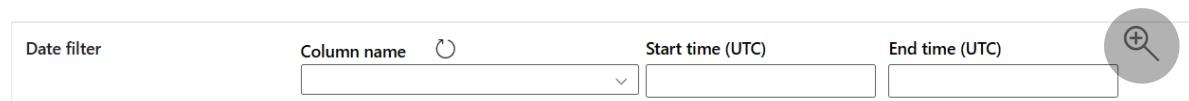
```
https://graph.microsoft.com/v1.0/users?$filter=Department eq 'Finance'.
```

If you select **Select groups from the Microsoft 365 tenant**, you can select **Add user groups** to select groups from the Microsoft 365 tenant. Use this property to select up to 10 user groups for whom the data is retrieved. If no groups are specified, then data is returned for the entire organization.



- **Date filter:** Specify the name of the DateTime filter column. Use this property to limit the time range for which Microsoft 365 data is extracted. If your dataset has one or more DateTime columns, you need to specify a column here. Refer to [Filtering](#) for a list of datasets that require this DateTime filter.

Specify the **Start time (UTC)** and **End time (UTC)** to filter on when you select a DateTime filter column.



- **Output columns:** Array of the columns to copy to the destination.

Mapping

For the **Mapping** tab configuration, go to [Configure your mappings under mapping tab.](#)

Settings

For the **Settings** tab configuration, go to [Configure your other settings under settings tab.](#)

Table summary

The following tables contain more information about the copy activity in Microsoft 365.

Source information

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the source data store.	<your connection>	Yes	connection
Table	Name of the table to extract from Microsoft 365.	<table>	Yes	table
Scope	When the user group isn't specified, you can use a predicate expression that's applied on the entire tenant to filter the specific rows to extract from Microsoft 365. The predicate format should match the query format of Microsoft Graph APIs, for example <code>https://graph.microsoft.com/v1.0/users?</code> <code>\$filter=Department eq 'Finance'.</code>	<your scope>	Yes	scope

Name	Description	Value	Required	JSON script property
Scope filter	When the <code>allowedGroups</code> property isn't specified, you can use a predicate expression that's applied on the entire tenant to filter the specific rows to extract from Microsoft 365. The predicate format should match the query format of Microsoft Graph APIs, for example <code>https://graph.microsoft.com/v1.0/users?</code> <code>\$filter=Department eq 'Finance'.</code>	<scope filter>	No	<code>userScopeFilterUri</code>
Group ID	Group selection predicate. Use this property to select up to 10 user groups for whom the data is retrieved. If no groups are specified, then data is returned for the entire organization.	<group id>	No	<code>allowedGroups</code>
Date filter (Column name)	Name of the DateTime filter column. Use this property to limit the time range for which Microsoft 365 data is extracted.	<your DateTime filter column>	Yes if data has one or more DateTime columns.	<code>dateFilterColumn</code>
Start time (UTC)	Start DateTime value to filter on.	<start time>	Yes if <code>dateFilterColumn</code> is specified	<code>startTime</code>
End time (UTC)	End DateTime value to filter on.	<end time>	Yes if <code>dateFilterColumn</code> is specified	<code>endTime</code>
Output columns	Array of the columns to copy to destination.	<output columns>	No	<code>outputColumns</code>

Next steps

- [How to create a Microsoft 365 connection](#)

OData connector overview

Article • 05/23/2023

This OData connector is supported in Data Factory for Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/-)	None	Anonymous Basic
Lookup activity	None	Anonymous Basic

Next steps

- [How to create an OData connection](#)
- [How to configure OData in a copy activity](#)
- [Connect to OData in dataflows](#)

How to create an OData connection

Article • 05/23/2023

This article outlines the steps to create an OData connection in a data pipeline.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported authentication types

This OData connector supports the following authentication types for copy and Dataflow Gen2 respectively.

Authentication type	Copy	Dataflow Gen2
Anonymous	✓	✓
Basic	✓	✓
Organizational account		✓

ⓘ Note

For information about the OData connection in Dataflow Gen2, go to [Connect to OData in dataflows](#).

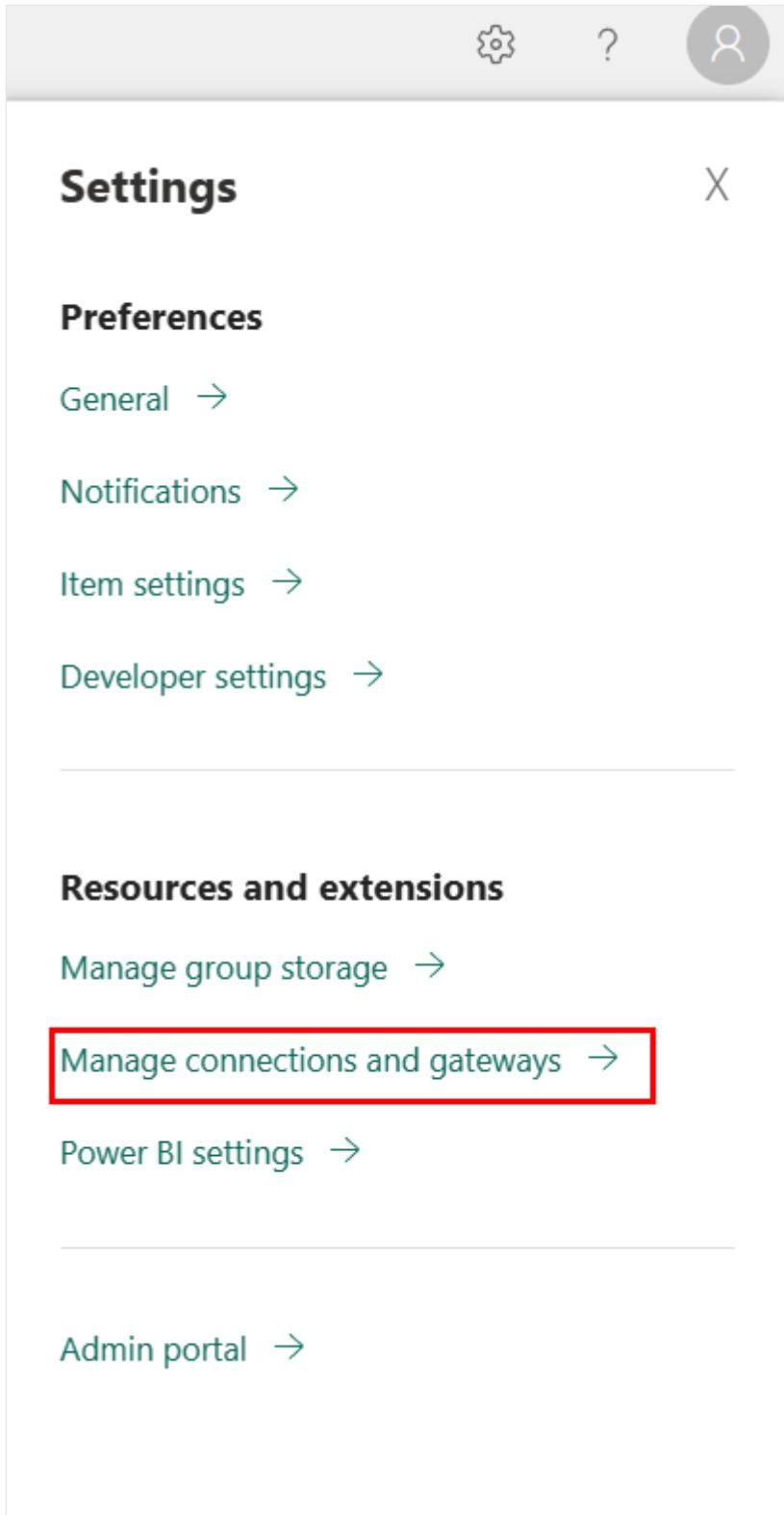
Prerequisites

The following prerequisites are required before you start:

- A Microsoft Fabric tenant account with an active subscription. [Create an account for free](#).
- A Microsoft Fabric enabled Workspace. [Create a workspace](#).

Go to manage gateways to create connection

1. From the page header in the Data Factory service, select **Settings**  > **Manage connections and gateways**.



2. Select **New** at the top of the ribbon to add a new data source.

[+ New](#) [Get help](#)

Data (preview)

Connections

On-premises data gateways

Virtual network data gateways

Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)

The New connection pane shows up on the left side of the page.

The screenshot shows the Power BI Data (preview) interface. On the left, there's a navigation bar with 'Data (preview)', 'Connections', 'On-premises data gateways', and 'Virtual network data gateways'. Below this, it says 'Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)'. On the right, a 'New connection' pane is open, titled 'New connection'. It includes a note: 'Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".' There are three connection types: 'On-premises' (disabled), 'Virtual network' (disabled), and 'Cloud' (selected). The 'Cloud' section has fields for 'Connection name *' and 'Connection type *'. At the bottom are 'Create' and 'Close' buttons, and a magnifying glass icon.

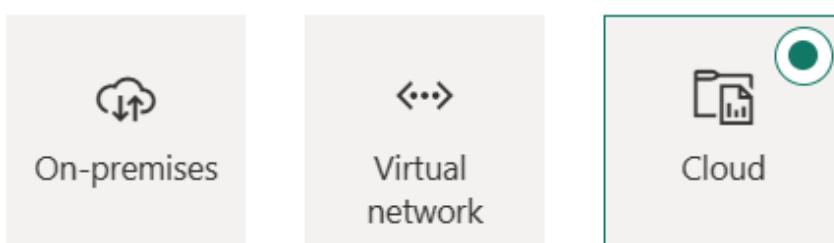
Setup connection

Step 1: Specify the new connection name, type, and URL

New connection



(i) Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



Connection name *

Connection type *

URL *

In the New connection pane, choose **Cloud**, and specify the following fields:

- **Connection name:** Specify a name for your connection.
- **Connection type:** Select **OData** for your connection type.
- **URL:** Enter the root URL of the **OData** service.

Step 2: Select and set your authentication

Under **Authentication method**, select your authentication from the drop-down list and complete the related configuration. This OData connector supports the following authentication types:

- Anonymous
- Basic

Authentication

Authentication method *

Anonymous



Anonymous

Basic

Key

OAuth2

Anonymous authentication

Under Authentication method, select **Anonymous**.

Authentication

Authentication method *

Anonymous



Skip test connection

Basic authentication

- **Username:** The user name to use to access the OData endpoint.
- **Password:** The password for the specified username.

Authentication

Authentication method *

Basic



Username *

< Your username >

Password *

< Your password >



Skip test connection

Step 3: Specify the privacy level that you want to apply

In the **General** tab, select the privacy level that you want apply in the **Privacy level** drop-down list. Three privacy levels are supported. For more information, go to [privacy levels](#).

Step 4: Create your connection

Select **Create**. Your creation is successfully tested and saved if all the credentials are correct. If not correct, the creation fails with errors.

Data (preview)

Connections On-premises data gateways Virtual network data gateways

Cloud and data gateway connections for artifacts. [Learn more about supported connections](#).

Name ↑	Connection type	Users	Status	Gateway cluster name
genericodata ...	OData	< user name >		

Table summary

The connector properties in the following table are supported in pipeline copy:

Name	Description	Required	Property	Copy
Connection name	A name for your connection.	Yes		✓
Connection type	Select OData for your connection type.	Yes		✓
URL	The base URL to the OData server.	Yes		✓
Authentication	Go to Authentication	Yes		Go to Authentication
Privacy Level	The privacy level that you want to apply. Allowed values are Organizational, Privacy, Public	Yes		✓

Authentication

The properties in the following table are the supported authentication type.

Name	Description	Required	Property	Copy
Anonymous				✓
Basic				✓
- Username	The user name to use to access the OData endpoint.	Yes		
- Password	The password for the specified username.	Yes		

Next steps

- [How to configure OData in copy activity](#)
- [Connect to OData in dataflows](#)

How to configure OData in a copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in a data pipeline to copy data from and to OData.

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported configuration

For the configuration of each tab under the copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Mapping](#)
- [Settings](#)

General

For General tab configuration, go to [General](#).

Source

The following properties are supported for OData under the **Source** tab of a copy activity.

The screenshot shows the 'Source' tab of the Azure Data Factory configuration interface. The 'Data store type' is set to 'External'. The 'Connection' dropdown is set to 'ODataConnection'. Under 'Use query', 'Path' is selected. The 'Path' input field is empty. The 'Edit' checkbox is checked. In the 'Advanced' section, 'Request timeout' is set to '00:05:00'. There is a 'New' button for 'Additional columns'. A magnifying glass icon is located in the bottom right corner.

The following three properties are required:

- **Data store type:** Select **External**.
- **Connection:** Select an OData connection from the connection list. If no connection exists, then create a new OData connection by selecting **New**.
- **Path:** Select the path to the OData resource. Or you can select **Edit** to enter the path manually.

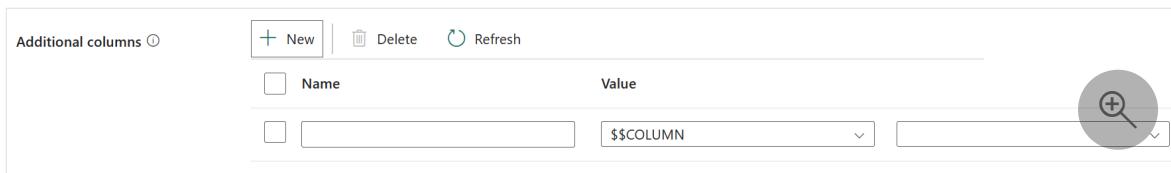
Under **Advanced**, you can specify the following fields:

- **Use query:** You can choose **Path** or **Query** as your use query. The following list describes the configuration of each setting.
 - **Path:** Read data from the specified path if you select this button.
 - **Query:** OData query options for filtering data. Example:
"\$select=Name,Description&\$top=5".

ⓘ Note

The OData connector copies data from the combined URL: [URL specified in the connection]/[path specified]?[query specified in copy activity source]. For more information, go to [OData URL components](#).

- **Request timeout:** Timeout for the HTTP request to get a response. Format is in timespan. This value is the timeout to get a response, not the timeout to read response data. The default value is 00:05:00.
- **Additional columns:** Add additional data columns to store source files' relative path or static value. Expression is supported for the latter.



Mapping

For the **Mapping** tab configuration, go to [Configure your mappings under mapping tab](#).

Settings

For the **Settings** tab configuration, go to [Configure your other settings under settings tab](#).

Table summary

The following table contains more information about the copy activity in OData.

Source

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the source data store.	<your connection>	Yes	connection
Path	The path to the OData resource.	<the path to the OData resource>	Yes	path
Use query	You can choose Path or Query as your use query.	<ul style="list-style-type: none">• Path• Query	No	query
Request timeout	Timeout for the HTTP request to get a response. Format is in timespan. This value is the timeout to get a response, not the timeout to read response data. The default value is 00:05:00.	timespan	No	requestTimeout

Name	Description	Value	Required	JSON script property
Additional columns	Add additional data columns to store source files' relative path or static value. Expression is supported for the latter.	- Name - Value	No	additionalColumns: - name - value

Next steps

- [How to create OData connection](#)
- [Connect to OData in dataflows](#)

Connect to OData in dataflows

Article • 05/23/2023

You can connect to an OData feed in Dataflow Gen2 using the OData connector provided by Data Factory in Microsoft Fabric.

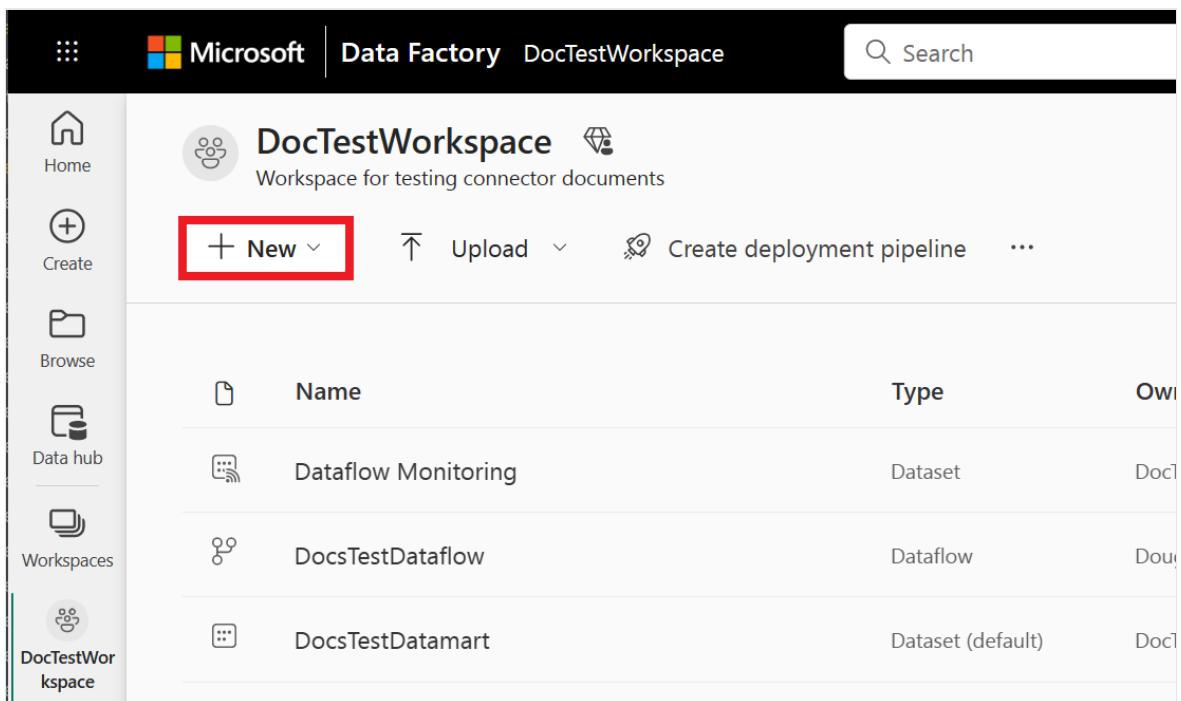
ⓘ Important

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Connect to an OData feed

To connect to an OData feed from a dataflow:

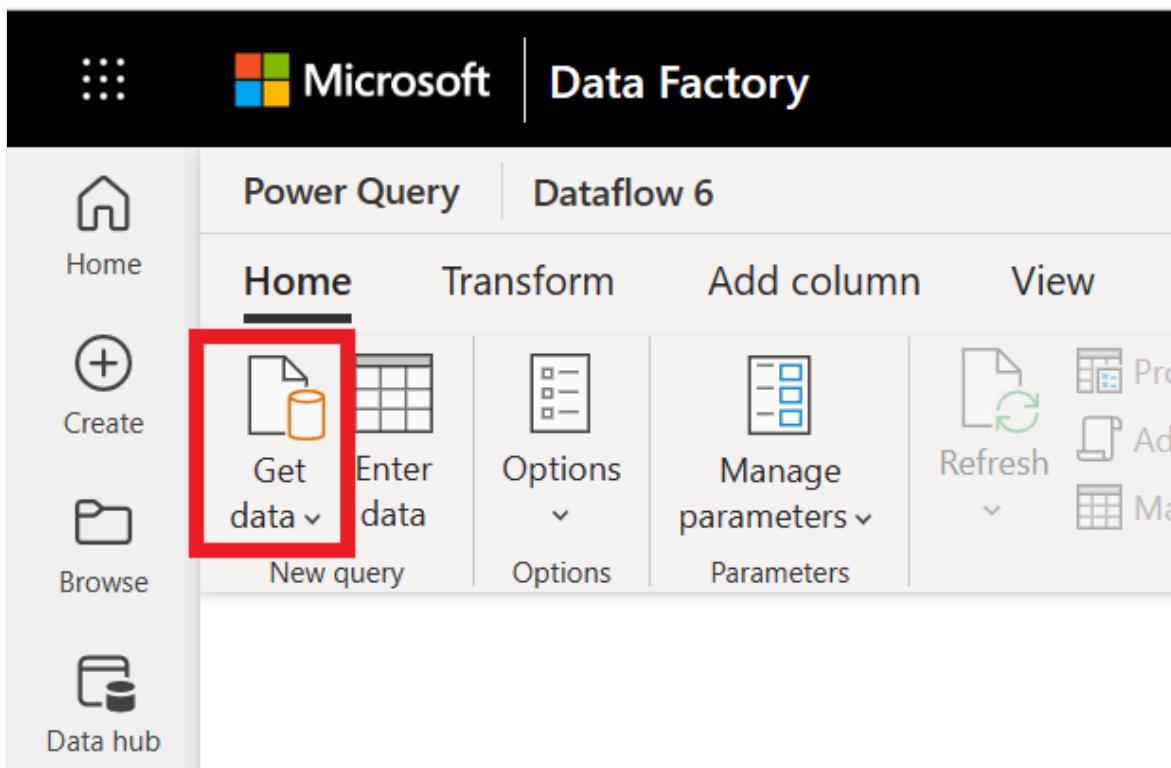
1. From your workspace, select **New > Dataflow Gen2 (Preview)** to create a new dataflow.



The screenshot shows the Microsoft Data Factory workspace interface. At the top, there is a navigation bar with icons for Home, Create, Browse, Data hub, Workspaces, and DocTestWorkspace. The 'DocTestWorkspace' section displays a workspace for testing connector documents. Below the navigation bar, there is a search bar and a 'New' button, which is highlighted with a red box. Other buttons include 'Upload', 'Create deployment pipeline', and a three-dot menu. On the left, a sidebar lists 'Home', 'Create', 'Browse', 'Data hub', 'Workspaces', and 'DocTestWorkspace'. The main area shows a table with three rows of data:

	Name	Type	Owner
Dataset	Dataflow Monitoring	Dataset	DocTest
Dataflow	DocsTestDataflow	Dataflow	DocT
Dataset (default)	DocsTestDatamart	Dataset (default)	DocT

2. In Power Query, either select **Get data** in the ribbon or select **Get data from another source** in the current view.



3. From **Choose data source**, select the **Other** category, and then select **OData**.

The screenshot shows the 'Choose data source' dialog. At the top, there's a search bar and a 'Search' button. Below it, a list of categories is shown: All categories, File, Database, Power Platform, Azure, Online services, and Other (which is highlighted with a red box). The main area displays various data sources in a grid:

Web API Other	Web page Other	SharePoint list Other
OData Other	Spark Other	Odbc Other
Acterys : Model Automation & ... Other	Amazon OpenSearch Service Other	Anaplan Other
Autodesk Construction Cloud Other	BQE Core Other	BitSight Security Ratings Other
Bloomberg Data and Analytics	Cherwell	Cognite Data Fusion

4. In **Connect to data source**, under **Connection settings**, enter the URL of your data.

5. If needed, select the on-premises data gateway in **Data gateway**.

6. Select the authentication type to use in **Authentication kind**, and then enter your credentials. More information: [Connections and authentication in Power Query Online](#)

Connect to data source

The screenshot shows the 'Connect to data source' page. On the left, there's a sidebar with an 'OData' icon and a 'Learn more' link. The main area has two sections: 'Connection settings' and 'Connection credentials'. In 'Connection settings', there's a 'URL *' input field with the placeholder 'Example: https://services.odata.org/Contoso/Contoso.svc'. In 'Connection credentials', there's a dropdown for 'Connection name' set to 'Connection', a dropdown for 'Data gateway' set to '(none)', and a dropdown for 'Authentication kind' set to 'Anonymous'. There are also refresh and search icons.

7. Select **Next**.

8. In **Choose data**, select the data items you want to transform, and then select **Transform data**.

The screenshot shows the 'Choose data' page. On the left, there's a tree view of data sources under 'OData' (26 items), with 'Employees' selected. A 'Select related tables' button is at the bottom. On the right, there's a table titled 'Employees' with columns: EmployeeID, LastName, FirstName, Title, TitleOfCourtesy, BirthDate, and HireDate. The table contains 9 rows of employee data. At the bottom right are 'Cancel' and 'Create' buttons.

	EmployeeID	LastName	FirstName	Title	TitleOfCourtesy	BirthDate	HireDate
1	Davolio	Nancy	Sales Representative	Ms.	12/8/1948, 12:00:00 AM +00:00	5/1/1992, 12:00:00 AM	
2	Fuller	Andrew	Vice President, Sales	Dr.	2/19/1952, 12:00:00 AM +00:00	8/14/1992, 12:00:00 AM	
3	Leverling	Janet	Sales Representative	Ms.	8/30/1963, 12:00:00 AM +00:00	4/1/1992, 12:00:00 AM	
4	Peacock	Margaret	Sales Representative	Mrs.	9/19/1937, 12:00:00 AM +00:00	5/3/1993, 12:00:00 AM	
5	Buchanan	Steven	Sales Manager	Mr.	3/4/1955, 12:00:00 AM +00:00	10/17/1993, 12:00:00 AM	
6	Suyama	Michael	Sales Representative	Mr.	7/2/1963, 12:00:00 AM +00:00	10/17/1993, 12:00:00 AM	
7	King	Robert	Sales Representative	Mr.	5/29/1960, 12:00:00 AM +00:00	1/2/1994, 12:00:00 AM	
8	Callahan	Laura	Inside Sales Coordinator	Ms.	1/9/1958, 12:00:00 AM +00:00	3/5/1994, 12:00:00 AM	
9	Dodsworth	Anne	Sales Representative	Ms.	1/27/1966, 12:00:00 AM +00:00	11/15/1994, 12:00:00 AM	

Advanced connector information

For more advanced information about connecting to your data using the OData connector, go to [OData feed](#).

Next steps

- [How to create an OData connection](#)
- [How to configure OData in a copy activity](#)

REST connector overview

Article • 05/23/2023

The REST connector is supported in Data Factory in Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/destination)	None	Anonymous Basic

Next steps

- [How to create a REST connection](#)
- [How to configure REST in a copy activity](#)

How to create REST connection

Article • 05/23/2023

This article outlines the steps to create REST connection.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported authentication types

This REST connector supports the following authentication types for copy and Dataflow Gen2 respectively.

Authentication type	Copy	Dataflow Gen2 (Web API)
Anonymous	✓	✓
Basic	✓	✓
Organizational account		✓
Windows		✓

ⓘ Note

For information about a REST connection in Dataflow Gen2, go to [Connect to REST APIs in dataflows](#).

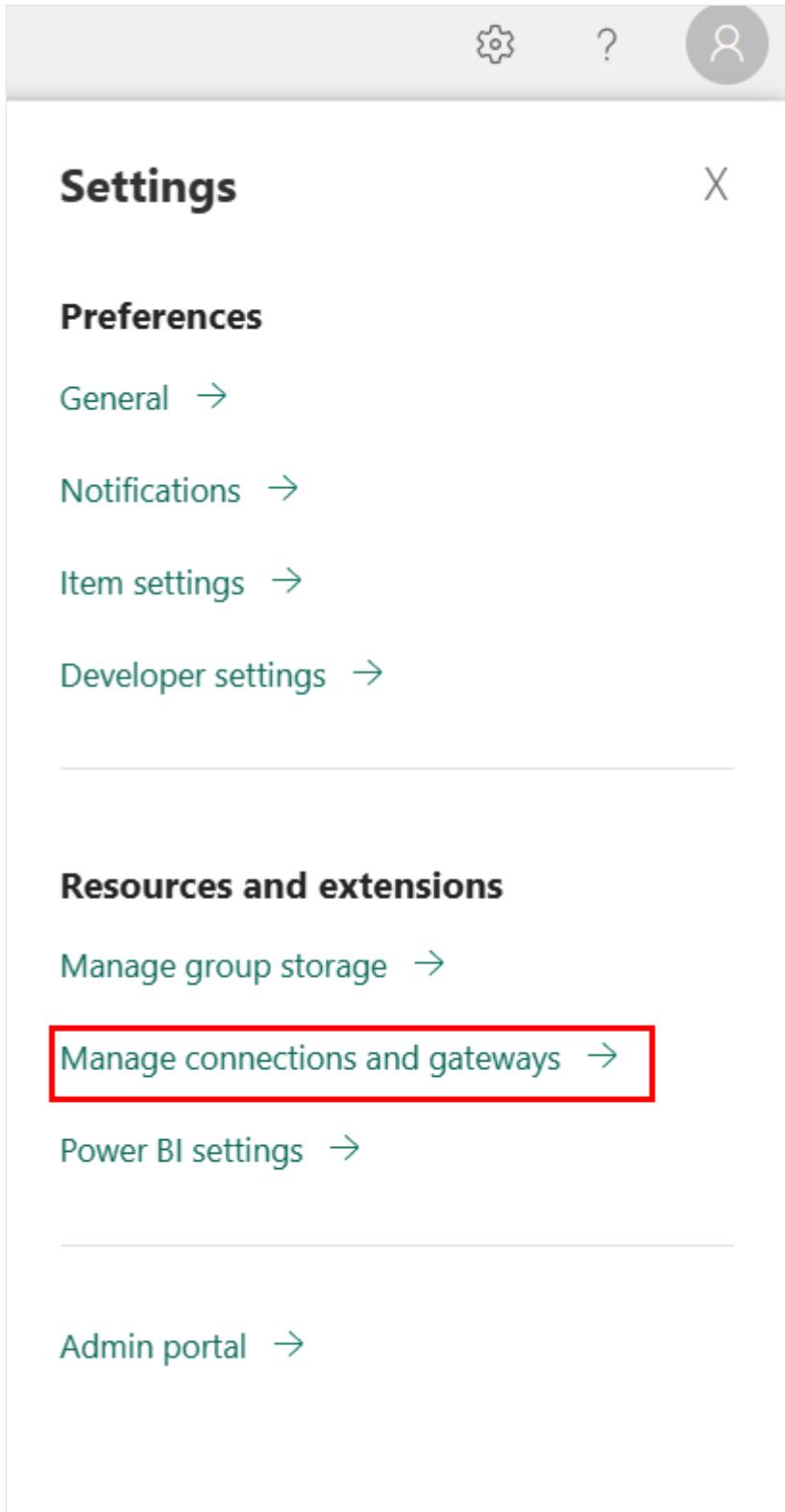
Prerequisites

The following prerequisites are required before you start:

- A Microsoft Fabric tenant account with an active subscription. [Create an account for free](#).
- A Microsoft Fabric enabled Workspace. [Create a workspace](#).

Go to Manage gateways to create connection

1. From the page header in Data Factory service, select **Settings**  > **Manage connections and gateways**.



2. Select **New** at the top of the ribbon to add a new data source.

[+ New](#) [Get help](#)

Data (preview)

Connections

On-premises data gateways

Virtual network data gateways

Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)

The New connection pane appears on the left side of the page.

The screenshot shows the Power BI Data (preview) interface. On the left, there's a list of existing connections: 'DemoBlob' (Azure Blob Storage, Clare). On the right, a 'New connection' dialog box is open, titled 'New connection'. It includes a note about cloud connections being supported for Data Pipelines and Kusto. Below the note are three connection types: 'On-premises' (selected), 'Virtual network', and 'Cloud' (highlighted with a green border). The 'Connection name' field is empty, and the 'Connection type' dropdown is also empty. At the bottom are 'Create' and 'Close' buttons.

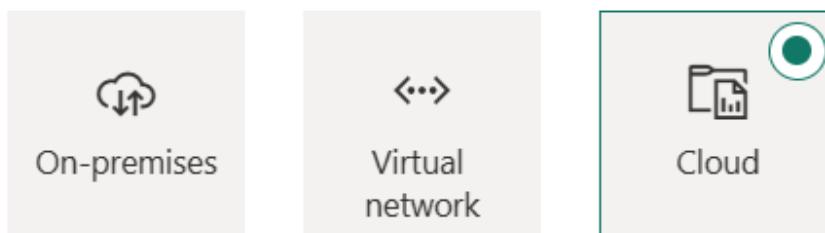
Setup connection

Step 1: Specify the new connection name, type, and URL

New connection



ⓘ Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



Connection name *

Connection type *



URL *

In the **New connection** pane, choose **Cloud**, and then specify the following fields:

- **Connection name:** Specify a name for your connection.
- **Connection type:** Select **Web** for your connection type.
- **URL:** The base URL to the web server.

Step 2: Select and set your authentication

Under **Authentication method**, select your authentication from the drop-down list and complete the related configuration. The REST connector supports the following authentication types:

- Anonymous
- Basic

Authentication

Authentication method *

Anonymous

Basic

OAuth2

Service Principal

Anonymous authentication

Select Anonymous under Authentication method.

Authentication

Authentication method *

Anonymous

Skip test connection

Basic authentication

- **Username:** The user name to use to access the REST endpoint.
- **Password:** The password for the specified username.

Authentication

Authentication method *

Basic



Username *

< Your username >

Password *

< Your password >



Skip test connection

Step 3: Specify the privacy level that you want to apply

In the **General** tab, select the privacy level that you want apply in the **Privacy level** drop-down list. Three privacy levels are supported. For more information, go to [privacy levels](#).

Step 4: Create your connection

Select **Create**. Your creation is successfully tested and saved if all the credentials are correct. If not correct, the creation fails with errors.

Data (preview)				
Connections	On-premises data gateways	Virtual network data gateways		
Cloud and data gateway connections for artifacts. Learn more about supported connections .				
Name ↑	Connection type	Users	Status	Gateway cluster name
genericrest ...	Web	< user name >		

Table summary

The following table contains connector properties that are supported in pipeline copy.

Name	Description	Required	Property	Copy
Connection name	A name for your connection.	Yes		✓
Connection type	Select Web for your connection type.	Yes		✓
URL	The base URL to the REST server.	Yes		✓
Authentication	Go to Authentication	Yes	See Authentication	
Privacy Level	The privacy level that you want to apply. Allowed values are Organizational, Privacy, Public	Yes		✓

Authentication

The following the following table contains the properties for supported authentication types.

Name	Description	Required	Property	Copy
Anonymous				✓
Basic				✓
- Username	The user name to use to access the REST endpoint.	Yes		
- Password	The password for the specified username.	Yes		

Next steps

- [How to configure REST in copy activity](#)

How to configure REST in a copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in data pipeline to copy data from and to a REST endpoint.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Specifically, this generic REST connector supports:

- Copying data from a REST endpoint by using the **GET** or **POST** methods and copying data to a REST endpoint by using the **POST**, **PUT** or **PATCH** methods.
- **Pagination** in the REST APIs.
- For REST as a source, copy the REST JSON response as-is or parse it by using schema mapping. Only the response payload in **JSON** is supported.

Supported configuration

For the configuration of each tab under a copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Destination](#)
- [Mapping](#)
- [Settings](#)

General

For General tab configuration, go to [General](#).

Source

The following properties are supported for REST under the **Source** tab of a copy activity.

The screenshot shows the 'Source' tab configuration for an external data store. Key settings include:

- Data store type:** External
- Connection:** genericrest
- Connection type:** REST
- Relative URL:** (empty)
- Advanced:**
 - Request method:** GET
 - Request timeout:** 00:01:40
 - Request interval (ms):** 10
 - Additional headers:** New
 - Pagination rules:** New
 - Additional columns:** New

The following first three properties are **required**:

- **Data store type:** Select **External**.
- **Connection:** Select an **REST** connection from the connection list. If no connection exists, then create a new REST connection by selecting **New**.
- **Connection type:** Select **REST**.
- **Relative URL:** A relative URL to the resource that contains the data. When this property isn't specified, only the URL that's specified in the connection definition is used. The HTTP connector copies data from the combined URL: `[URL specified in connection]/[relative URL specified]`.

Under **Advanced**, you can specify the following fields:

- **Request method:** The HTTP method. Allowed values are **GET** (default) and **POST**.

The screenshot shows the 'Advanced' section with the 'Request method' dropdown open. The selected value is **GET**, and the other options are **POST** and **PUT**.

- **Request timeout:** The timeout (the timespan value) for the HTTP request to get a response. This value is the timeout to get a response, not the timeout to read response data. The default value is 00:01:40.

- **Request interval (ms):** The interval time between different requests for multiple pages in milliseconds. Request interval value should be a number between [10, 60000].
- **Additional headers:** Additional HTTP request headers.

Name	Value
<input type="text"/>	<input type="text"/>

- **Pagination rules:** The pagination rules to compose next page requests. Refer to the [pagination support section](#) for details.

Name	Value
<input type="text"/> None	<input type="text"/> <input type="button" value="None"/>

- **Additional columns:** Add more data columns to store source files' relative path or static value. Expression is supported for the latter.

Name	Value	
<input type="text"/>	<input type="text"/> <input type="button" value="\$\$COLUMN"/>	<input type="button"/>

Destination

The following properties are supported for REST under the **Destination** tab of a copy activity.

General Source **Destination** Mapping Settings

Data store type Workspace External

Connection

Connection type

Relative URL

▼ Advanced

Request method *

Request timeout

Request interval (ms) ⓘ

Write batch size ⓘ

Http Compression type ⓘ

Additional headers

The following first three properties are **required**:

- **Data store type:** Select **External**.
- **Connection:** Select a **REST** connection from the connection list. If no connection exists, then create a new REST connection by selecting **New**.
- **Connection type:** Select **REST**.
- **Relative URL:** A relative URL to the resource that contains the data. When this property isn't specified, only the URL that's specified in the connection definition is used. The HTTP connector copies data from the combined URL: `[URL specified in connection]/[relative URL specified]`.

Under **Advanced**, you can specify the following fields:

- **Request method:** The HTTP method. Allowed values are **POST** (default), **PUT**, and **PATCH**.
- **Request timeout:** The timeout (the timespan value) for the HTTP request to get a response. This value is the timeout to get a response, not the timeout to write the data. The default value is **00:01:40**.
- **Request interval (ms):** The interval time between different requests for multiple pages in milliseconds. Request interval value should be a number between [10, 60000].
- **Write batch size:** Number of records to write to the REST destination per batch. The default value is **10000**.

- **Http Compression type:** HTTP compression type to use while sending data with Optimal Compression Level. Allowed values are **None** and **GZip**.
- **Additional headers:** Additional HTTP request headers.

<input type="checkbox"/>	Name	Value
<input type="checkbox"/>	Name	Value

Mapping

For the **Mapping** tab configuration, go to [Configure your mappings under mapping tab](#). Mapping is not supported when both source and destination are hierarchical data.

Settings

For the **Settings** tab configuration, go to [Configure your other settings under settings tab](#).

Table summary

The following tables contain more information about the copy activity in REST.

Source information

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	<ul style="list-style-type: none"> • Workspace • External • Sample dataset 	Yes	type
Connection	Your connection to the source data store.	<your connection>	Yes	connection
Connection type	Your connection type. Select REST.	REST	Yes	type: RestResource

Name	Description	Value	Required	JSON script property
Relative URL	A relative URL to the resource that contains the data. When this property isn't specified, only the URL that's specified in the connection definition is used. The HTTP connector copies data from the combined URL: [URL specified in connection]/[relative URL specified].	<your relative url>	No	relativeUrl
Request method	The HTTP method. Allowed values are GET (default) and POST .	• GET • POST	No	requestMethod
Request timeout	The timeout (the timespan value) for the HTTP request to get a response. This value is the timeout to get a response, not the timeout to read response data. The default value is 00:01:40.	timespan	No	httpRequestTimeout
Request interval (ms)	The interval time between different requests for multiple pages in milliseconds. Request interval value should be a number between [10, 60000].	[10, 60000]	No	requestInterval
Additional headers	Additional HTTP request headers.	<your additional headers>	No	additionalHeaders
Pagination rules	The pagination rules to compose next page requests. Refer to the pagination support section for details.	Go to pagination	No	paginationRules
Additional columns	Add more data columns to the store source files' relative path or static value. Expression is supported for the latter.	• Name • Value	No	additionalColumns: • name • value

Destination information

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	<ul style="list-style-type: none"> • Workspace • External 	Yes	type
Connection	Your connection to the source data store.	<your connection>	Yes	connection
Connection type	Your connection type. Select REST.	REST	Yes	type: RestResource
Relative URL	A relative URL to the resource that contains the data. When this property isn't specified, only the URL that's specified in the connection definition is used. The HTTP connector copies data from the combined URL: [URL specified in connection]/[relative URL specified].	<your relative url>	No	relativeUrl
Request method	The HTTP method. Allowed values are POST (default), PUT, and PATCH.	<ul style="list-style-type: none"> • POST • GET • PATCH 	No	requestMethod
Request timeout	The timeout (the timespan value) for the HTTP request to get a response. This value is the timeout to get a response, not the timeout to read response data. The default value is 00:01:40.	timespan	No	httpRequestTimeout
Request interval (ms)	The interval time between different requests for multiple pages in milliseconds. Request interval value should be a number between [10, 60000].	[10, 60000]	No	requestInterval
Write batch size	Number of records to write to the REST destination per batch. The default value is 10000.	<number of rows> (integer)	No	writeBatchSize

Name	Description	Value	Required	JSON script property
Http Compression type	HTTP compression type to use while sending data with Optimal Compression Level. Allowed values are None and GZip .	<ul style="list-style-type: none"> • None • GZip 	No	httpCompressionType
Additional headers	Additional HTTP request headers.	<your additional headers>	No	additionalHeaders

Next steps

- [How to create a REST connection](#)

Connect to REST APIs in dataflows

Article • 05/23/2023

You can connect to REST APIs in Dataflow Gen2 using the Web API connector provided by Data Factory in Microsoft Fabric.

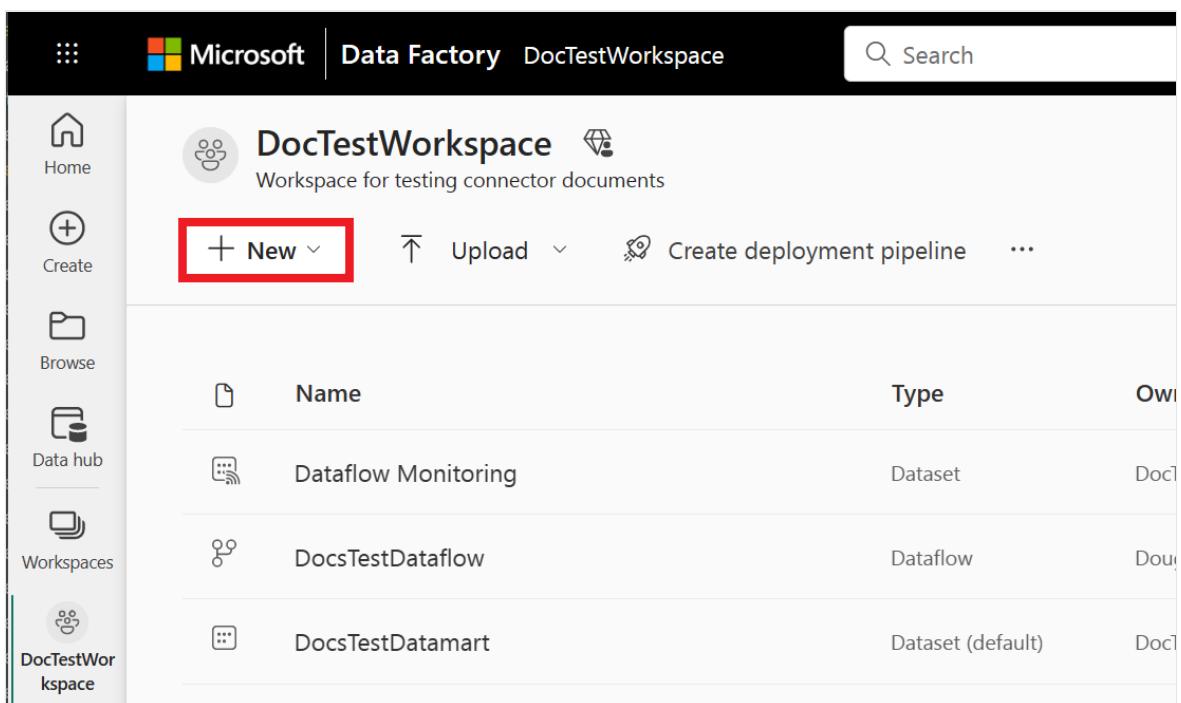
ⓘ Important

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Connect to REST APIs

To connect to REST APIs in a dataflow:

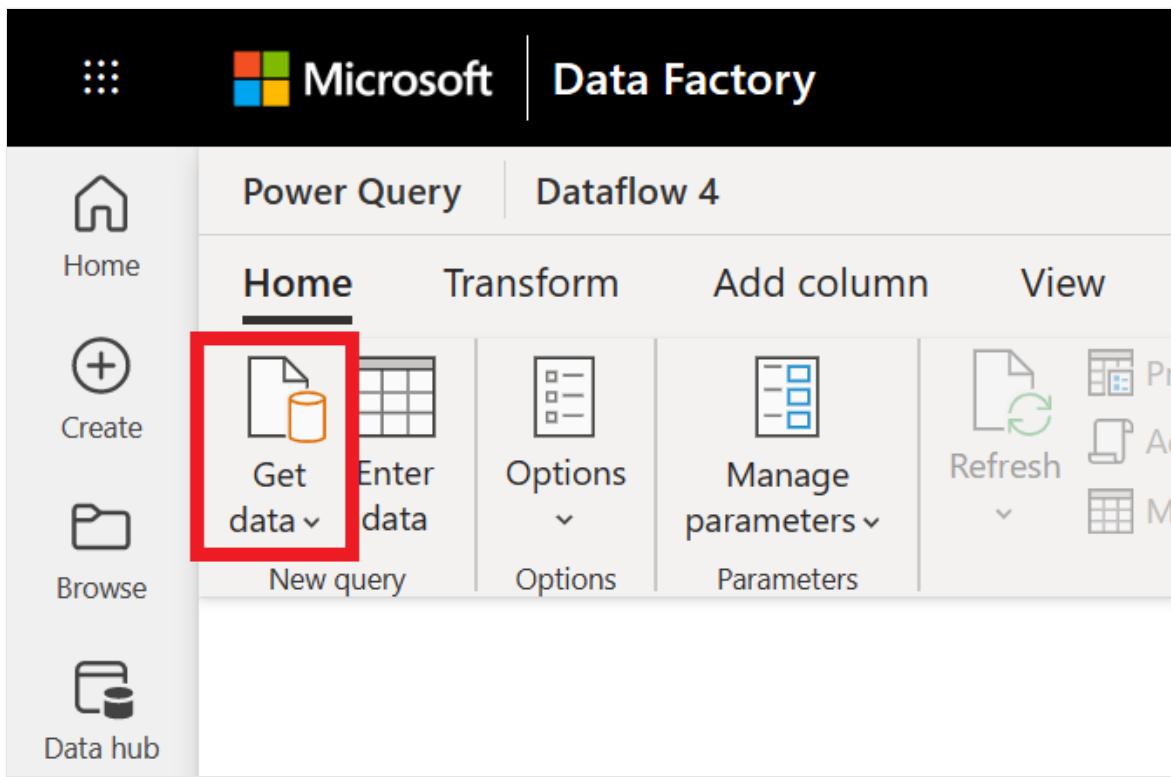
1. From your workspace, select **New > Dataflow Gen2 (Preview)** to create a new dataflow.



The screenshot shows the Microsoft Data Factory workspace interface. At the top, there is a navigation bar with icons for Home, Create, Browse, Data hub, Workspaces, and DocTestWorkspace. The 'DocTestWorkspace' tab is selected. On the right side of the header, there is a 'Search' bar. Below the header, there is a main area titled 'DocTestWorkspace' with a subtitle 'Workspace for testing connector documents'. In the center, there is a 'New' button with a dropdown arrow, which is highlighted with a red box. To the right of the 'New' button are 'Upload', 'Create deployment pipeline', and a three-dot menu icon. Below this, there is a table listing three items: 'Dataflow Monitoring' (Dataset, Owner: DocTest), 'DocsTestDataflow' (Dataflow, Owner: DocTest), and 'DocsTestDatamart' (Dataset (default), Owner: DocTest). The left sidebar also lists 'Home', 'Create', 'Browse', 'Data hub', 'Workspaces', and 'DocTestWorkspace'.

Name	Type	Owner
Dataflow Monitoring	Dataset	DocTest
DocsTestDataflow	Dataflow	DocTest
DocsTestDatamart	Dataset (default)	DocTest

2. In Power Query, either select **Get data** in the ribbon or select **Get data from another source** in the current view.



3. From **Choose data source**, select the **Other** category, and then select **Web API**.

A screenshot of the 'Choose data source' dialog. At the top, it says 'Select a connector or directly drag a file from your computer.' Below are tabs for All categories, File, Database, Power Platform, Azure, Online services, and Other (which is highlighted with a red box). The main area shows a grid of connectors: Web API (highlighted with a red box), Web page, Anaplan, BQE Core, Delta Sharing, EQuIS, and others. The 'Web API' connector has a 'BETA' badge next to it.

4. In **Connect to data source**, under **Connection settings**, enter the URL of the REST API. Generally, this entry is a URI with some parameters, for example `https://api.\<some domain>.com?format=json`. However, there are no rules around how this URI is configured. To compile the URI, consult the documentation for the REST APIs you're going to be using.

5. If necessary, select the on-premises data gateway in **Data gateway**.

6. Select the authentication type to use in **Authentication kind**, and then enter your credentials. The authentication kinds supported by the Web page connector are:

- Anonymous
- Basic

- Organizational account
- Windows

More information: [Connections and authentication in Power Query Online](#)

7. Select **Next**.

8. In **Choose data**, select the data item that you want to transform, and then select **Next**.

Kind	Name	Children	Text
Element	HTML	[Table]	null

The data that's returned is generally in JSON format, but might also be XML.

Advanced connector information

For more advanced information about connecting to your data using the Web connector, go to [Web](#).

Next steps

- [How to create a REST connection](#)
- [How to configure REST in a copy activity](#)

SharePoint Online List connector overview

Article • 05/23/2023

The SharePoint Online List connector is supported in Data Factory for Microsoft Fabric with the following capabilities.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported capabilities

Supported capabilities	Gateway	Authentication
Copy activity (source/-)	None	Anonymous OAuth2 Service Principal
Lookup activity	None	Anonymous OAuth2 Service Principal

Next steps

- [How to create SharePoint Online List connection](#)
- [How to configure SharePoint Online List in copy activity](#)
- [Connect to a SharePoint Online list in dataflows](#)

How to create SharePoint Online List connection

Article • 05/23/2023

This article outlines the steps to create SharePoint Online List connection.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Supported authentication types

This SharePoint Online List connector supports the following authentication types for copy and Dataflow Gen2 respectively.

Authentication type	Copy	Dataflow Gen2
Anonymous	✓	
OAuth2	✓	
Service Principal	✓	
Organizational account		✓

ⓘ Note

For information about a SharePoint Online list connection in Dataflow Gen2, go to [Connect to a SharePoint Online list in dataflows](#).

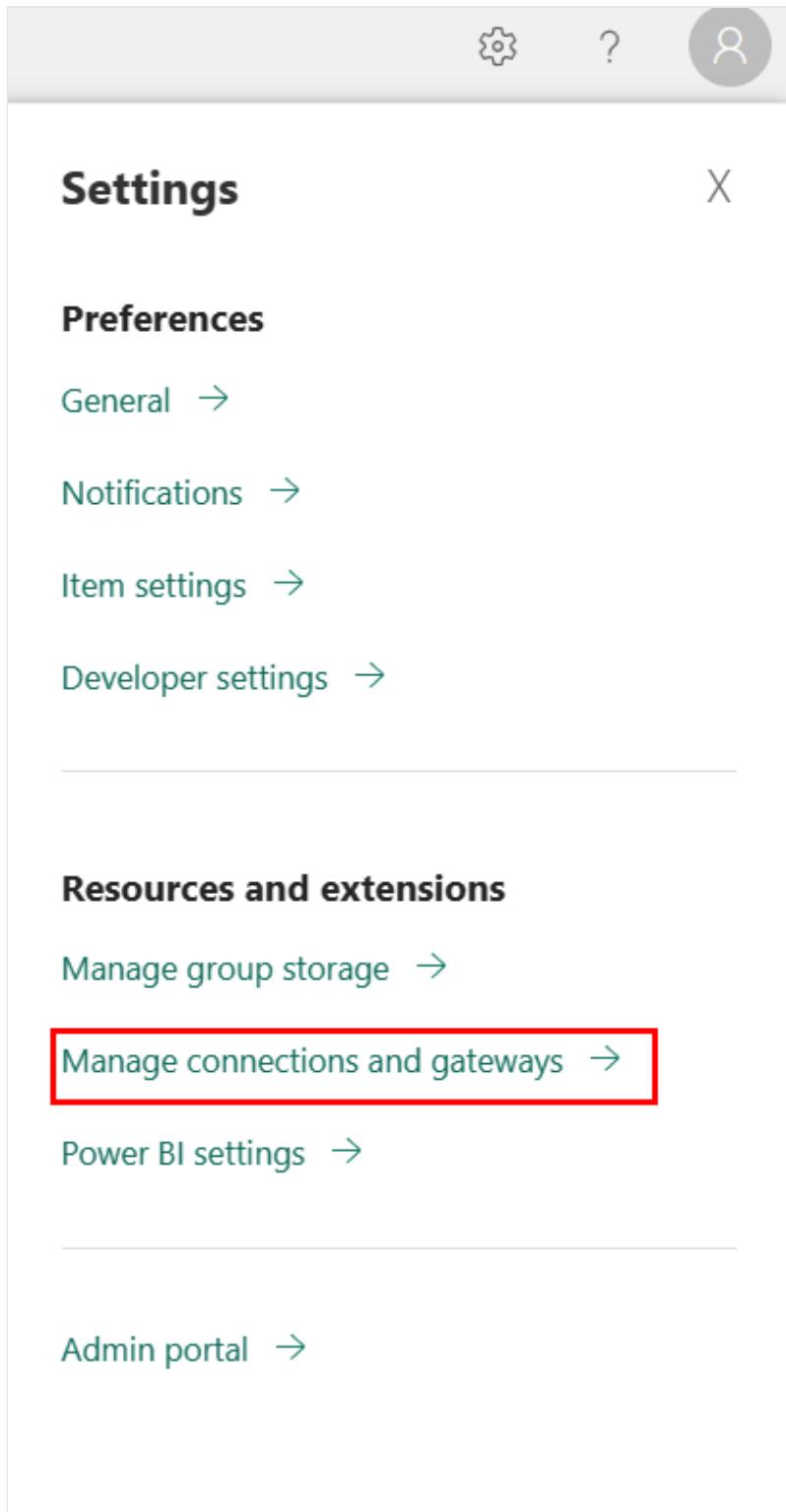
Prerequisites

The following prerequisites are required before you start:

- A Microsoft Fabric tenant account with an active subscription. [Create an account for free](#).
- A Microsoft Fabric enabled Workspace. [Create a workspace](#).

Go to manage gateways to create connection

1. From the page header in Data Factory service, select **Settings**  > **Manage connections and gateways**.



2. Select **New** at the top of the ribbon to add a new data source.

[+ New](#)[Get help](#)

Data (preview)

[Connections](#)[On-premises data gateways](#)[Virtual network data gateways](#)Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)

The **New connection** pane appears on the left side of the page.

The screenshot shows the Power BI Data (preview) interface. On the left, there's a list of existing connections: "Data (preview)", "Connections", "On-premises data gateways", and "Virtual network data gateways". A tooltip indicates that cloud connections are currently only supported for Data Pipelines and Kusto. On the right, the "New connection" pane is open, showing options for "On-premises", "Virtual network", and "Cloud". The "Cloud" option is selected. Fields for "Connection name" and "Connection type" are present, along with "Create" and "Close" buttons and a search icon.

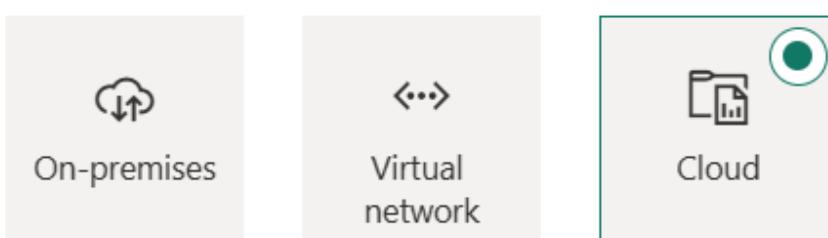
Setup connection

Step 1: Specify the new connection name, type, and URL

New connection



(i) Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



Connection name *

SharePointOnlineList

Connection type *

SharePoint

SharePoint site URL *

< Your URL >

In the New connection pane, choose **Cloud**, and specify the following fields:

- **Connection name:** Specify a name for your connection.
- **Connection type:** Select **SharePoint** for your connection type.
- **URL:** The SharePoint Online site URL, for example
`https://contoso.sharepoint.com/sites/siteName.`

Step 2: Select and set your authentication

Under **Authentication method**, select your authentication from the drop-down list and complete the related configuration. The SharePoint Online List connector supports the following authentication types.

- Anonymous
- OAuth2
- Service Principal

Authentication

Authentication method *

Anonymous

Anonymous

OAuth2

Service Principal

Anonymous authentication

Select **Anonymous** under **Authentication method**.

Authentication

Authentication method *

Anonymous

OAuth2 authentication

Select **OAuth2** under **Authentication method**.

Authentication

Authentication method *

OAuth2

[Edit credentials](#)

Service Principal authentication

Select Service Principal under Authentication method, and fill in the required properties.

Authentication

Authentication method *

Service Principal ▼

Tenant ID *

< Your Tenant ID >

Service principal ID *

< Your Service Principal ID >

Service principal key *

< Your Service Principal key >

- **Tenant Id:** The tenant ID under which your application resides.
- **Service principal ID:** The Application (client) ID of the application registered in Azure Active Directory. Refer to [Prerequisites](#) for more details, including the permission settings.
- **Service principal key:** The application's key.

Step 3: Specify the privacy level that you want to apply

In the General tab, select the privacy level that you want apply in the Privacy level drop-down list. Three privacy levels are supported. For more information, go to [privacy levels](#).

Step 4: Create your connection

Select **Create**. Your creation is successfully tested and saved if all the credentials are correct. If not correct, the creation fails with errors.

Data (preview)

Connections On-premises data gateways Virtual network data gateways

Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)

Name ↑	Connection type	Users	Status	Gateway cluster name
SharePointOnlineList ...	SharePoint	<user name>		(+)

Table summary

The following table contains the connector properties supported in a pipeline copy.

Name	Description	Required	Property	Copy
Connection name	A name for your connection.	Yes		✓
Connection type	Select SharePoint for your connection type. If no connection exists, then create a new connection by selecting New .	Yes		✓
URL	The SharePoint Online site URL, for example <code>https://contoso.sharepoint.com/sites/siteName</code> .	Yes		✓
Authentication	Go to Authentication	Yes		Go to Authentication
Privacy Level	The privacy level that you want to apply. Allowed values are Organizational , Privacy , Public	Yes		✓

Authentication

The following table contains the supported authentication type properties.

Name	Description	Required	Property	Copy
Anonymous				✓
OAuth2				✓
Service Principal				✓
- Tenant ID	The tenant ID under which your application resides.	Yes		
- Service Principal ID	The Application (client) ID of the application registered in Azure Active Directory. Refer to Prerequisites for more details including the permission settings.	Yes		

Name	Description	Required	Property	Copy
- Service Principal key	The application's key.	Yes		

Next steps

- [How to configure SharePoint Online List in a copy activity](#)
- [Connect to a SharePoint Online list in dataflows](#)

How to configure SharePoint Online List in a copy activity

Article • 05/23/2023

This article outlines how to use the copy activity in a data pipeline to copy data from SharePoint Online List.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Prerequisites

1. The SharePoint List Online connector uses service principal authentication to connect to SharePoint. Follow these steps to set it up:
 2. Register an application with the Microsoft Identity platform. To learn how, go to [Quickstart: Register an application with the Microsoft identity platform](#). Make note of these values, which you use to define the connection:
 - Application ID
 - Application key
 - Tenant ID
 3. Use the following steps to grant SharePoint Online site permission to your registered application. To grant permission, you need a site admin role.
 - a. Open a SharePoint Online site link, for example `https://[your_site_url]/_layouts/15/appinv.aspx` (replace the site URL).
 - b. Search the application ID you registered, fill the empty fields, and then select **Create**.
 - App Domain: `contoso.com`
 - Redirect URL: `https://www.contoso.com`
 - Permission Request XML:

XML

```
<AppPermissionRequests AllowAppOnlyPolicy="true">
  <AppPermissionRequest
    Scope="http://sharepoint/content/sitecollection/web"
    Right="Read"/>
</AppPermissionRequests>
```

App Id:

<application GUID>

Lookup

Title:

spodemo

App Domain:

localhost.com

Example: "www.contoso.com"

Redirect URL:

https://www.localhost.com

Example: "https://www.contoso.com/default.aspx"

Permission Request XML:

```
<AppPermissionRequests
  AllowAppOnlyPolicy="true">
  <AppPermissionRequest
    Scope="http://sharepoint/content/sitecollection/we
    b" Right="Read"/>
</AppPermissionRequests>
```

① Note

In the context of configuring the SharePoint connector, the **App Domain** and **Redirect URL** refer to the SharePoint app that you've registered in Azure Active Directory (Azure AD) to allow access to your SharePoint data. The **App Domain** is the domain where your SharePoint site is hosted. For example, if your SharePoint site is located at <https://contoso.sharepoint.com>, then the **App Domain** would be `contoso.sharepoint.com`. The **Redirect URL** is the URL that the SharePoint app redirects to after the user has authenticated and granted permissions to the app. This URL should be a page on your SharePoint site that the app has permission to access. For example, you could use the URL of a page that displays a list of files in a library, or a page that displays the contents of a document.

- c. Select **Trust It** for this app.

Supported configuration

For the configuration of each tab under a copy activity, go to the following sections respectively.

- [General](#)
- [Source](#)
- [Mapping](#)
- [Settings](#)

General

For General tab configuration, go to [General](#).

Source

The following properties are supported for SharePoint Online List under the **Source** tab of a copy activity.

The screenshot shows the 'Source' tab of a data store configuration. The 'Data store type' is set to 'External'. The 'Connection' dropdown is set to 'SharePointOnlineList'. Under 'Use query', 'List name' is selected. The 'List name' dropdown is set to 'None'. There is an 'Edit' link next to it. Below this, there is an 'Advanced' section with a 'Request timeout' input set to '00:05:00'. At the bottom right is a circular button with a plus sign and a magnifying glass icon.

The following properties are **required**:

- **Data store type:** Select **External**.
- **Connection:** Select a SharePoint Online List connection from the connection list. If no connection exists, then create a new HTTP connection by selecting **New**.
- **Use query:** Choose either **List name** or **Query** as your use query. The configuration of each setting is:
 - **List name:** The name of the SharePoint Online list.

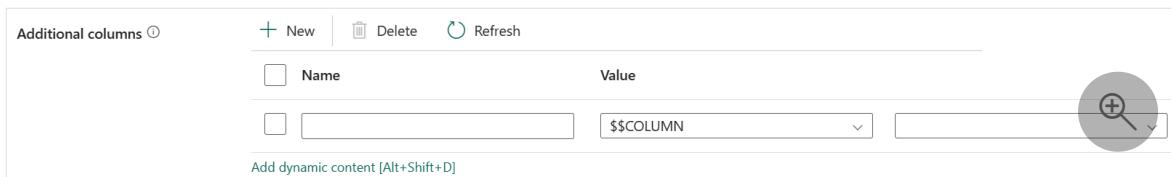
This screenshot shows the 'Source' tab configuration for an external data store. The 'Use query' section has 'List name' selected. The 'List name' dropdown is set to 'None'. There is an 'Edit' link next to it. To the right are 'Refresh' and 'Preview data' buttons, and a circular button with a plus sign and a magnifying glass icon.

- **Query:** The OData query to filter the data in SharePoint Online list. For example, `"$top=1"`.

This screenshot shows the 'Source' tab configuration for an external data store. The 'Use query' section has 'Query' selected. Below it is a large text input field for the query, which is currently empty. To the right are 'Edit' and 'Preview data' buttons, and a circular button with a plus sign and a magnifying glass icon.

Under **Advanced**, you can specify the following fields:

- **Request timeout:** The wait time to get a response from SharePoint Online. Default value is 5 minutes (00:05:00).
- **Additional columns:** Add additional data columns to store source files' relative path or static value. Expression is supported for the latter.



Mapping

For **Mapping** tab configuration, go to [Configure your mappings under mapping tab](#).

Settings

For **Settings** tab configuration, go to [Configure your other settings under settings tab](#).

Table summary

The following table contains more information about a copy activity in SharePoint Online List.

Source

Name	Description	Value	Required	JSON script property
Data store type	Your data store type.	External	Yes	/
Connection	Your connection to the source data store.	<your connection>	Yes	connection
Use query	You can choose List name or Query as your use query.	-List name -Query	No	type
List name	The name of the SharePoint Online list.	<your connection>	Yes	listName
Request timeout	The wait time to get a response from SharePoint Online. Default value is 5 minutes (00:05:00).	timespan	No	requestTimeout
Additional columns	Add additional data columns to store source files' relative path or static value. Expression is supported for the latter.	• Name • Value	No	additionalColumns: • name • value

Next steps

- [How to create a SharePoint Online List connection](#)
- [Connect to a SharePoint Online list in dataflows](#)

Connect to a SharePoint Online list in dataflows

Article • 05/23/2023

You can connect to a SharePoint Online list in Dataflow Gen2 using the SharePoint Online list connector provided by Data Factory in Microsoft Fabric.

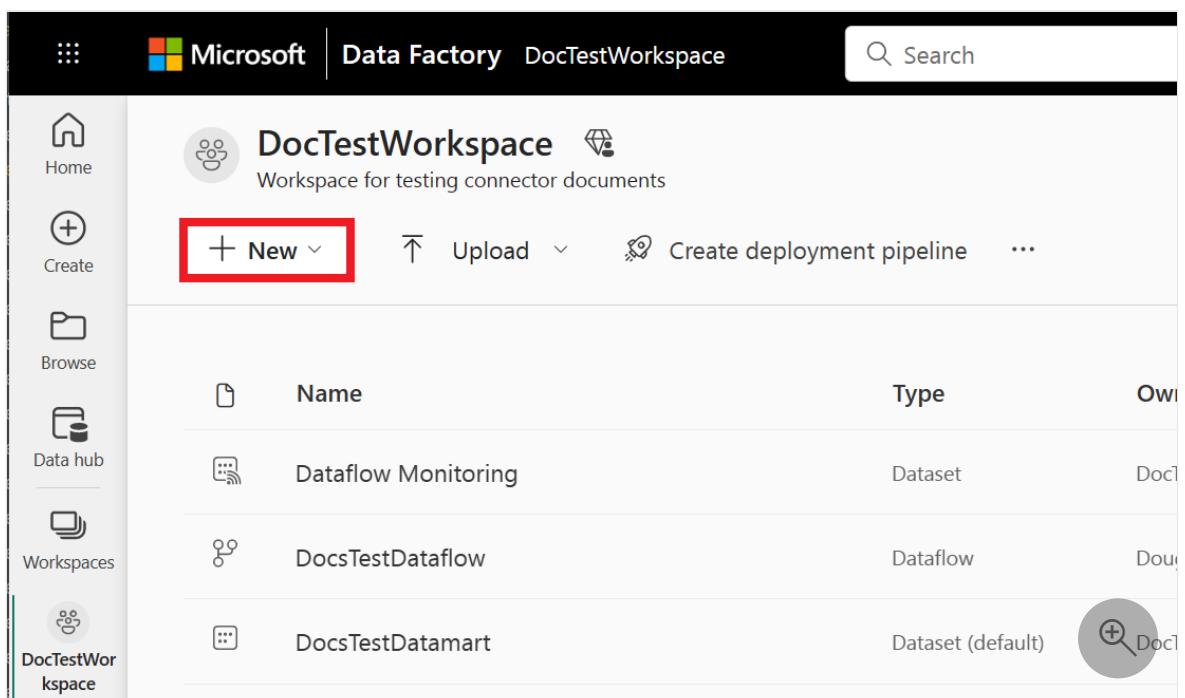
ⓘ Important

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Connect to a SharePoint Online list

To connect to a SharePoint Online list from a dataflow:

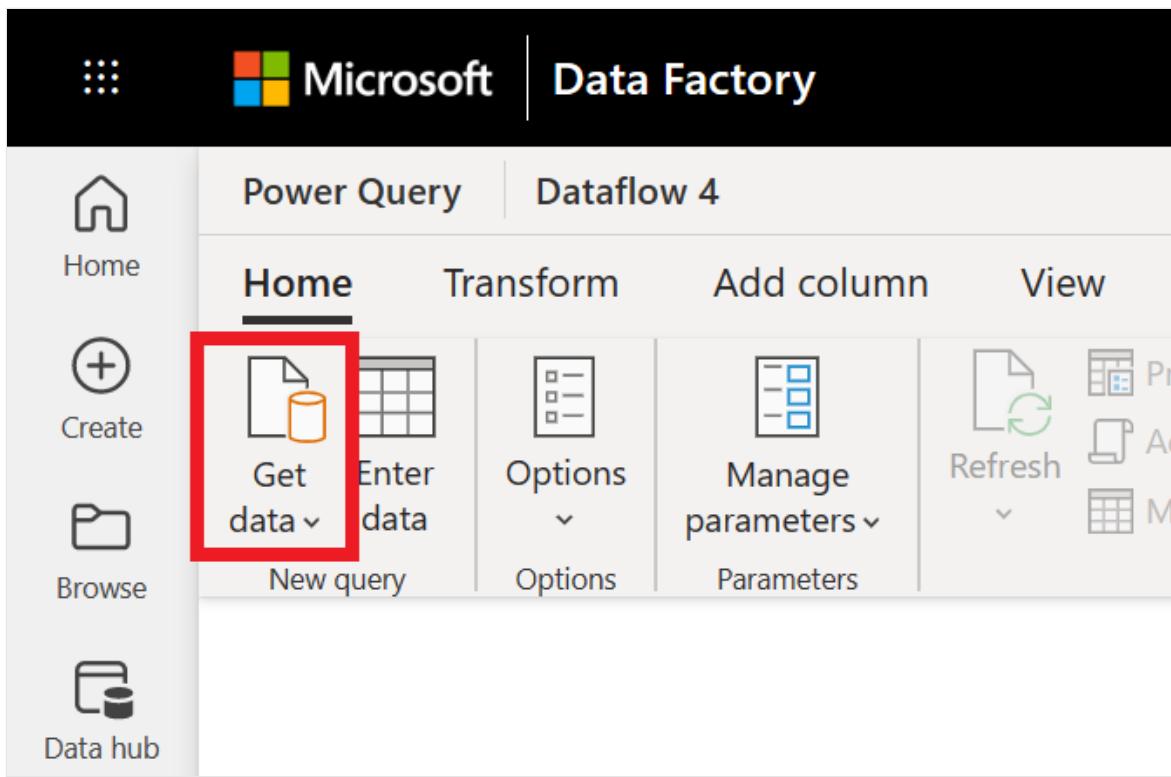
1. From your workspace, select **New > Dataflow Gen2 (Preview)** to create a new dataflow.



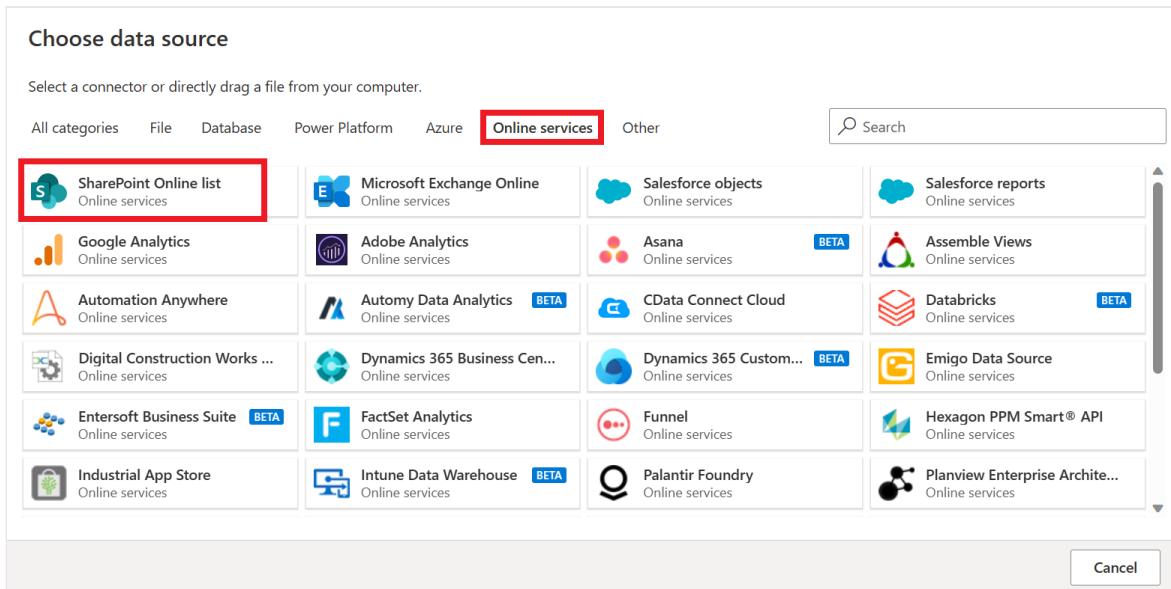
The screenshot shows the Microsoft Data Factory interface. At the top, there is a navigation bar with the Microsoft logo, the text "Data Factory DocTestWorkspace", and a search bar. On the left, a sidebar lists "Home", "Create" (with a red box around the "+ New" button), "Browse", "Data hub", "Workspaces", and "DocTestWor kspace" (which is currently selected). The main area displays a table with three rows of data:

	Name	Type	Owner
File icon	Dataflow Monitoring	Dataset	DocT
User icon	DocsTestDataflow	Dataflow	DocT
Table icon	DocsTestDatamart	Dataset (default)	DocT

2. In Power Query, either select **Get data** in the ribbon or select **Get data from another source** in the current view.



3. From **Choose data source**, select the **Online services** category, and then select **SharePoint Online list**.



4. In **Connect to data source**, under **Connection settings**, enter the site URL. For information on determining the site URL, go to [Determine the site URL](#).
5. Select the Implementation you want to use for the connection. More information: [Connect to SharePoint Online list v2.0](#)
6. Enter the name of an on-premises data gateway if needed.
7. If you're connecting to this data source for the first time, select the authentication type to use in **Authentication kind**, and then enter your credentials. More information: [Connections and authentication in Power Query Online](#)

Connect to data source

SharePoint Online list
Online services
[Learn more](#)

Connection settings

Site URL * ⓘ
Example: <https://contoso.sharepoint.com/>

Implementation
1.0

> Advanced options

Connection credentials

Connection
Create new connection ⚡

Connection name
Connection

Data gateway
(none) ⚡

Authentication kind
Organizational account

+ ⚡

8. Select **Next** to continue to the Power Query editor, where you can then begin to transform your data.

Choose data

Search

Display options ⚡

SharePoint Online list [22]

- appdata
- appfiles
- Composed Looks
- Content type publishing ...
- Converted Forms
- Documents
- Events
- Form Templates
- List Template Gallery
- Maintenance Log Library
- marketing
- MarketingData** (checked)
- Master Page Gallery

MarketingData

FileSystemObjectType	Id	ServerRedirectedEmbedUri	ServerRedirectedEmbedUrl	ID	Content
0	1	null		1	0x0100C
0	2	null		2	0x0100C
0	3	null		3	0x0100C
0	4	null		4	0x0100C
0	5	null		5	0x0100C
0	6	null		6	0x0100C
0	7	null		7	0x0100C
0	8	null		8	0x0100C
0	9	null		9	0x0100C
0	10	null		10	0x0100C
0	11	null		11	0x0100C
0	12	null		12	0x0100C
0	13	null		13	0x0100C
0	14	null		14	0x0100C
0	15	null		15	0x0100C
0	16	null		16	0x0100C
	..				

Back Cancel Create

Advanced connector information

For more advanced information about connecting to your data using the SharePoint Online list connector, go to [SharePoint Online list](#).

Next steps

- [How to create a SharePoint Online list connection](#)
- [How to configure SharePoint Online List in a copy activity](#)

Activity overview

Article • 05/23/2023

This article helps you understand activities in Microsoft Fabric and use them to construct end-to-end data-driven workflows for your data movement and data processing scenarios.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Overview

A Microsoft Fabric Workspace can have one or more pipelines. A pipeline is a logical grouping of activities that together perform a task. For example, a pipeline could contain a set of activities that ingest and clean log data, and then kick off a mapping data flow to analyze the log data. The pipeline allows you to manage the activities as a set instead of each one individually. You deploy and schedule the pipeline instead of the activities independently.

The activities in a pipeline define actions to perform on your data. For example, you may use a copy activity to copy data from SQL Server to an Azure Blob Storage. Then, use a Dataflow activity or a Notebook activity to process and transform data from the blob storage to an Azure Synapse Analytics pool on top of which business intelligence reporting solutions are built.

Microsoft Fabric has three types of activities: data movement activities, data transformation activities, and control activities.

Data movement activities

Copy activity in Microsoft Fabric copies data from a source data store to a sink data store. Fabric supports the data stores listed in the [Connector overview](#) article. Data from any source can be written to any sink.

For more information, see [How to copy data using the copy activity](#).

Data transformation activities

Microsoft Fabric supports the following transformation activities that can be added either individually or chained with another activity.

For more information, see the [data transformation activities](#) article.

Data transformation activity	Compute environment
Copy data	Compute manager by Microsoft Fabric
Dataflow Gen2	Compute manager by Microsoft Fabric
Delete data	Compute manager by Microsoft Fabric
Fabric Notebook	Apache Spark clusters managed by Microsoft Fabric
Fabric Spark job definition	Apache Spark clusters managed by Microsoft Fabric
Stored Procedure	Azure SQL, Azure Synapse Analytics, or SQL Server
SQL script	Azure SQL, Azure Synapse Analytics, or SQL Server

Control flow activities

The following control flow activities are supported:

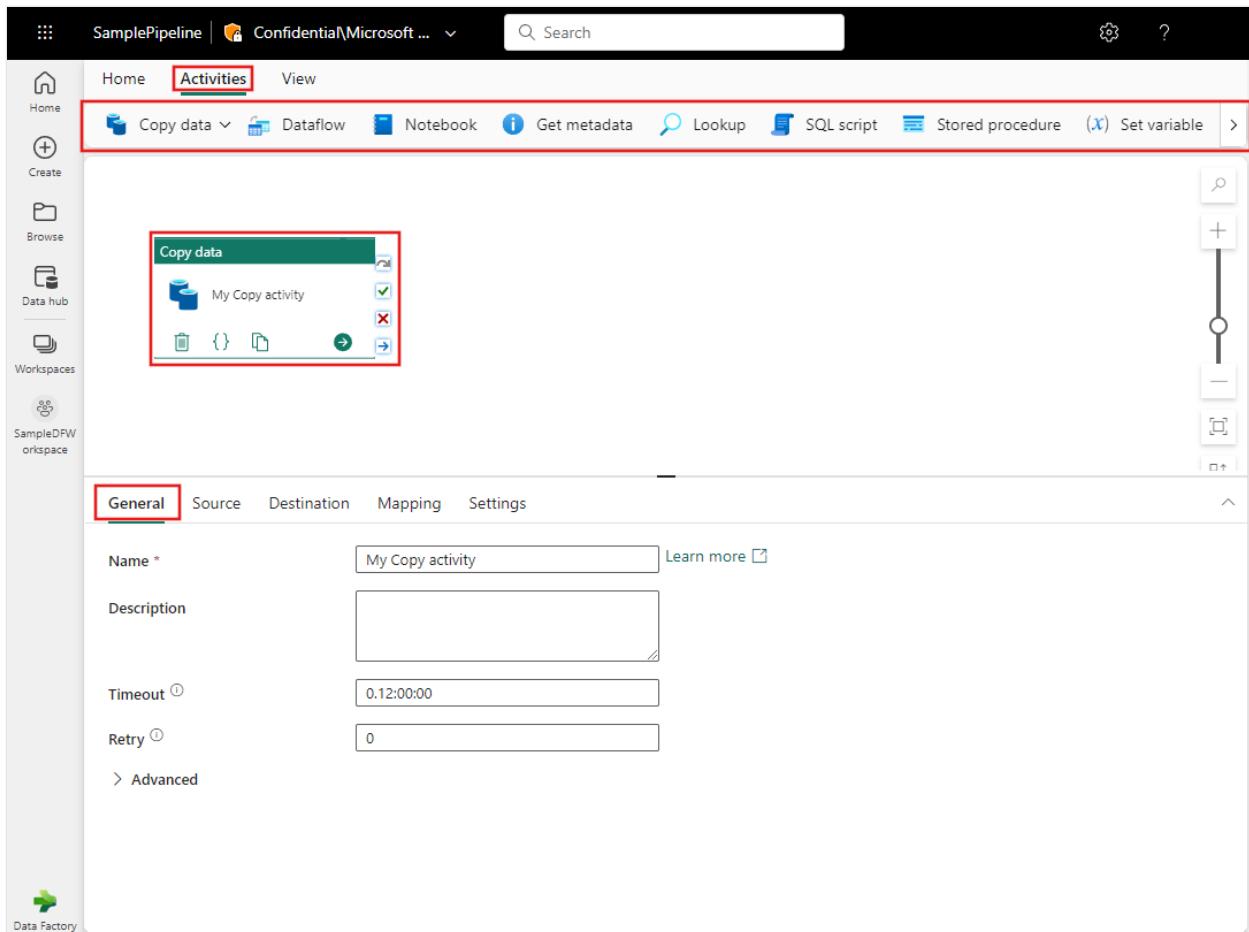
Control activity	Description
Append variable	Add a value to an existing array variable.
Invoke pipeline	Execute Pipeline activity allows a Data Factory or Synapse pipeline to invoke another pipeline.
Filter	Apply a filter expression to an input array.
ForEach	ForEach Activity defines a repeating control flow in your pipeline. This activity is used to iterate over a collection and executes specified activities in a loop. The loop implementation of this activity is similar to the Foreach looping structure in programming languages.
Get metadata	GetMetadata activity can be used to retrieve metadata of any data in a Data Factory or Synapse pipeline.

Control activity	Description
If condition	The If Condition can be used to branch based on condition that evaluates to true or false. The If Condition activity provides the same functionality that an if statement provides in programming languages. It evaluates a set of activities when the condition evaluates to <code>true</code> and another set of activities when the condition evaluates to <code>false</code> .
Lookup Activity	Lookup Activity can be used to read or look up a record/ table name/ value from any external source. This output can further be referenced by succeeding activities.
Set Variable	Set the value of an existing variable.
Until Activity	Implements Do-Until loop that is similar to Do-Until looping structure in programming languages. It executes a set of activities in a loop until the condition associated with the activity evaluates to true. You can specify a timeout value for the until activity.
Wait Activity	When you use a Wait activity in a pipeline, the pipeline waits for the specified time before continuing with execution of subsequent activities.
Web Activity	Web Activity can be used to call a custom REST endpoint from a pipeline. You can pass datasets and linked services to be consumed and accessed by the activity.
Webhook Activity	Using the webhook activity, call an endpoint, and pass a callback URL. The pipeline run waits for the callback to be invoked before proceeding to the next activity.

Adding activities to a pipeline with the Microsoft Fabric UI

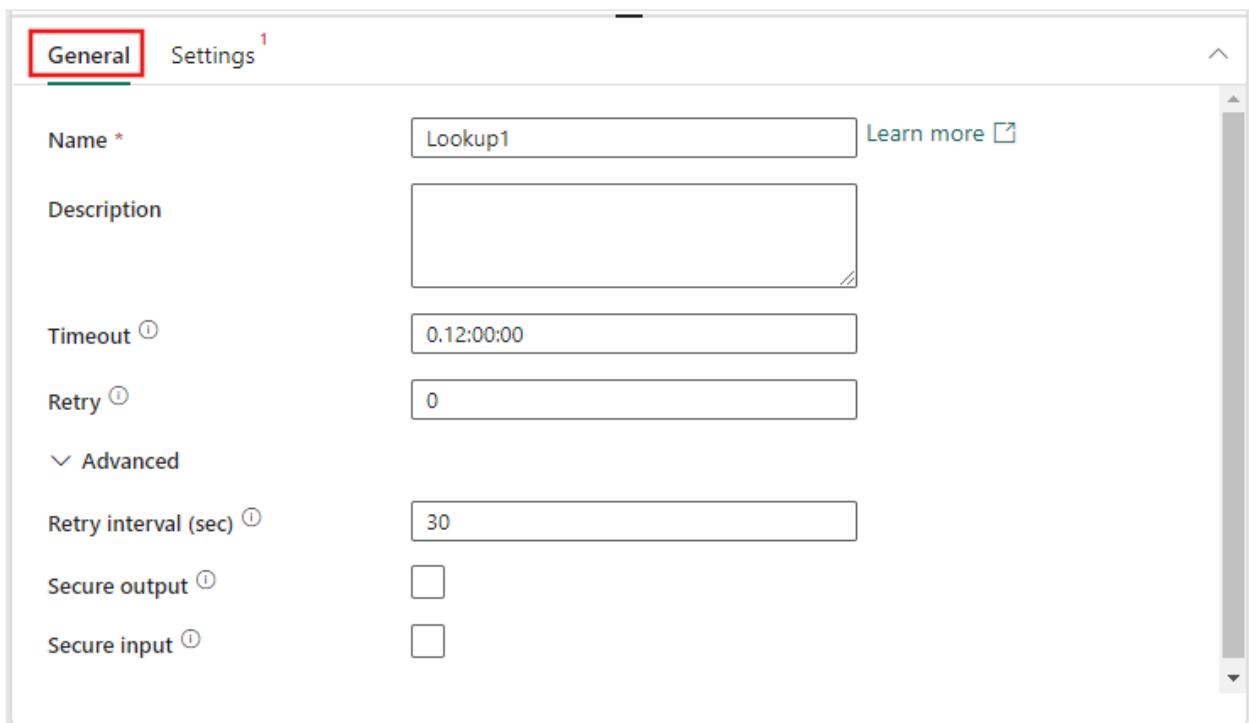
Use these steps to add and configure activities in a Microsoft Fabric pipeline:

1. Create a new pipeline in your workspace.
2. On the Activities tab for the pipeline, browse the activities displayed, scrolling to the right if necessary to see all activities. Select an activity to add it to the pipeline editor.
3. When you add an activity and select it in the pipeline editor canvas, its **General** settings will appear in the properties pane below the canvas.
4. Each activity also contains custom properties specific to its configuration on other tabs in the properties pane.



General settings

When you add a new activity to a pipeline and select it, you'll see its properties panes in the area at the bottom of the screen. These properties panes include **General**, **Settings**, and sometimes other panes as well.



The general settings will always include **Name** and **Description** fields for every activity. Some activities also include the following:

Setting	Description
Timeout	The maximum amount of time an activity can run. The default is 12 hours, and the maximum amount of time allowed is seven days. The format for the timeout is in D.HH:MM:SS.
Retry	Maximum number of retry attempts.
(Advanced properties) Retry interval (sec)	The number of seconds between each retry attempt.
(Advanced properties) Secure output	When checked, output from the activity isn't captured in logging.
(Advanced properties) Secure input	The number of seconds between each retry attempt.

Next steps

[Create your first pipeline](#)

How to copy data using copy activity

Article • 05/23/2023

In Data Pipeline, you can use the Copy activity to copy data among data stores located in the cloud.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

After you copy the data, you can use other activities to further transform and analyze it. You can also use the Copy activity to publish transformation and analysis results for business intelligence (BI) and application consumption.

To copy data from a source to a destination, the service that runs the Copy activity performs these steps:

1. Reads data from a source data store.
2. Performs serialization/deserialization, compression/decompression, column mapping, and so on. It performs these operations based on the configuration.
3. Writes data to the destination data store.

Prerequisites

To get started, you must complete the following prerequisites:

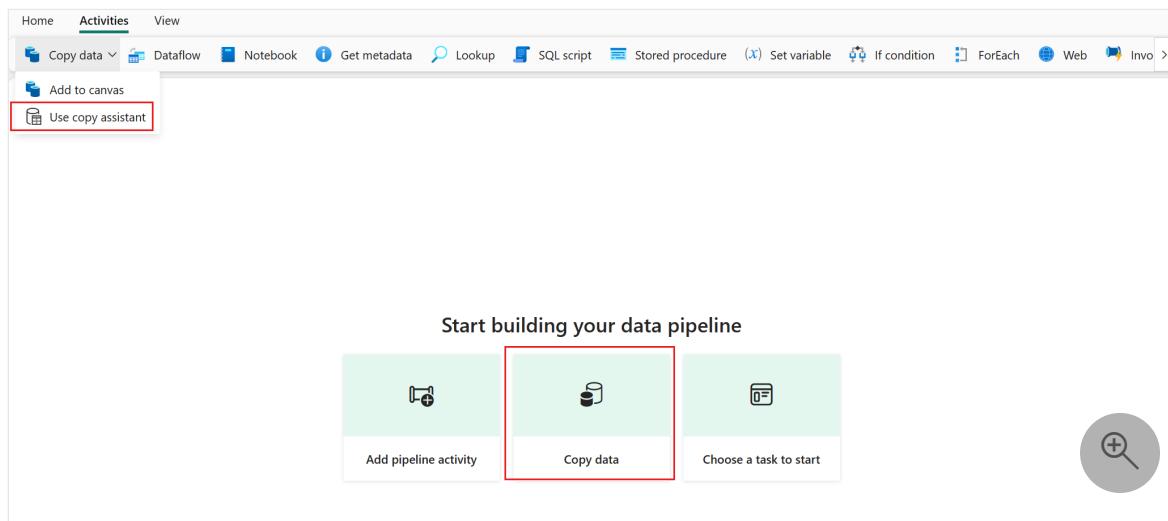
- A Microsoft Fabric tenant account with an active subscription. Create an account for free.
- Make sure you have a Microsoft Fabric enabled Workspace.

Add a copy activity using copy assistant

Follow these steps to set up your copy activity using copy assistant.

Start with copy assistant

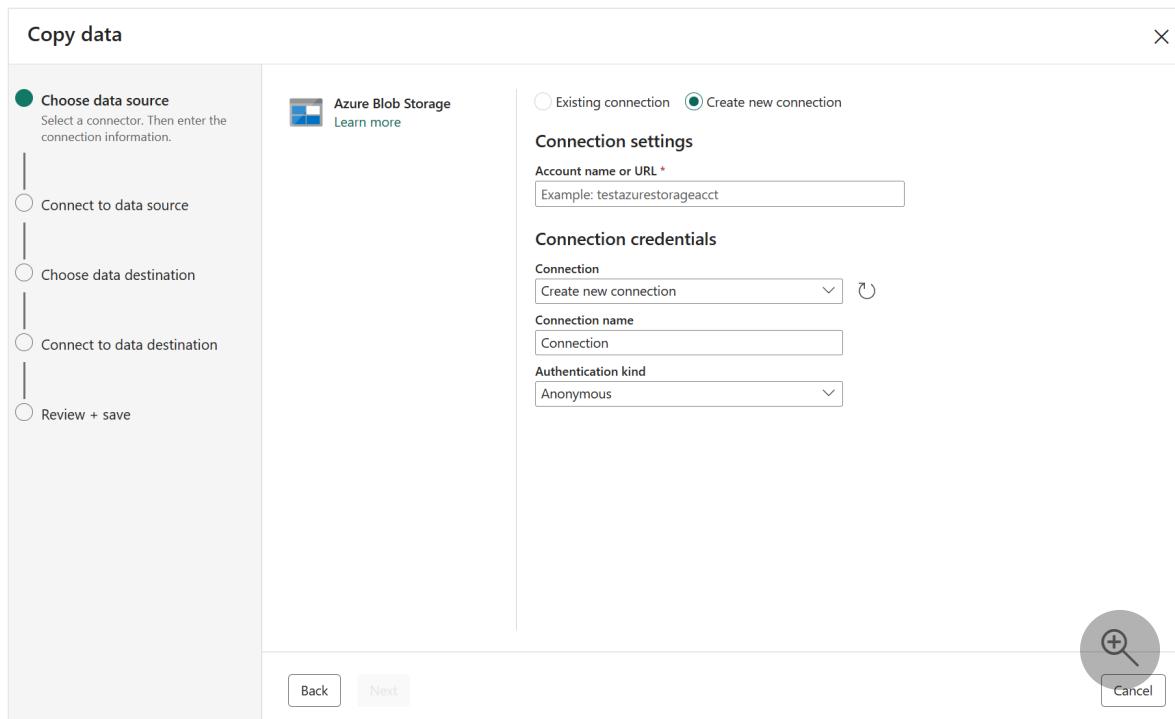
1. Open an existing data pipeline or create a new data pipeline.
2. Select **Copy data** on the canvas to open the **Copy Assistant** tool to get started. Or select **Use copy assistant** from the **Copy data** drop down list under the **Activities** tab on the ribbon.



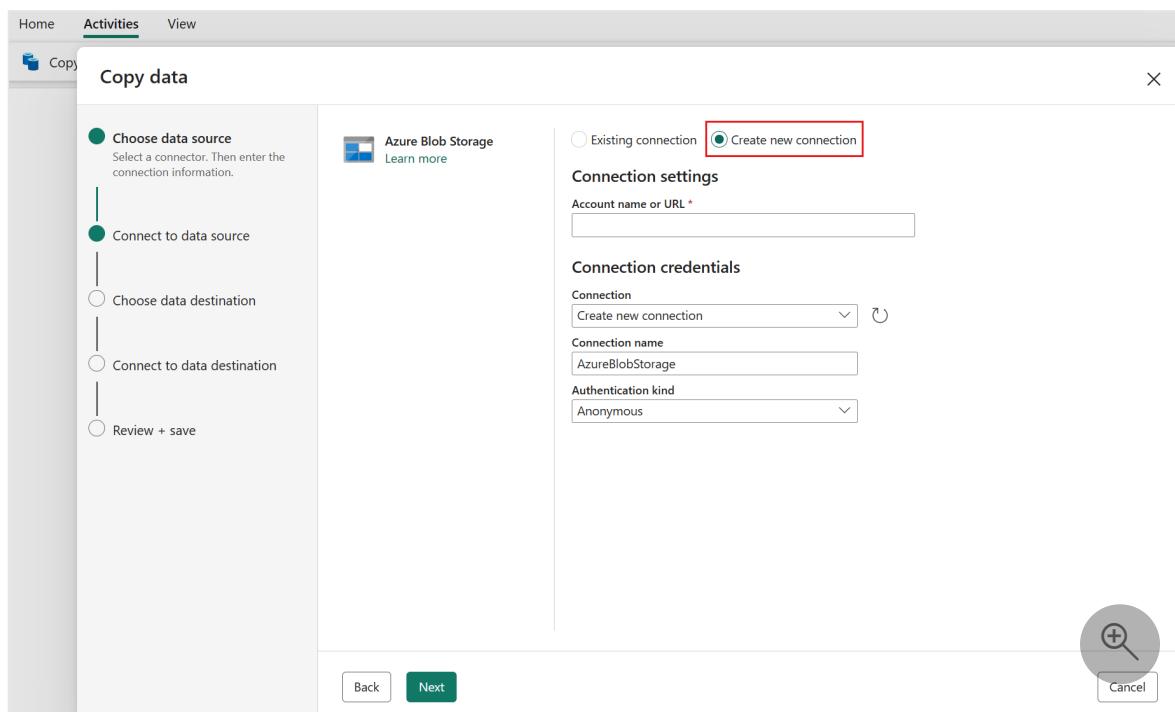
Configure your source

1. Select a data source type from the category. You'll use Azure Blob Storage as an example. Select **Azure Blob Storage** and then select **Next**.

Category	Source	Destination
All categories	Amazon Redshift Database	Amazon S3 File
	Apache Impala Database	Azure Blob Storage Azure
	Azure Cosmos DB for NoSQL Azure	Azure Data Explorer (Kusto) Azure
	Azure Data Lake Storage Gen1 Azure	Azure Data Lake Storage Gen2 Azure
	Azure Database for PostgreSQL Azure	Azure SQL Database Azure
	Azure SQL Database Managed Instance Azure	Azure Synapse Analytics Azure
	Azure Table Storage Azure	Data Warehouse Workspace
	Dataverse (Common Data Service for Annel)	Google Cloud Storage (S3 API)

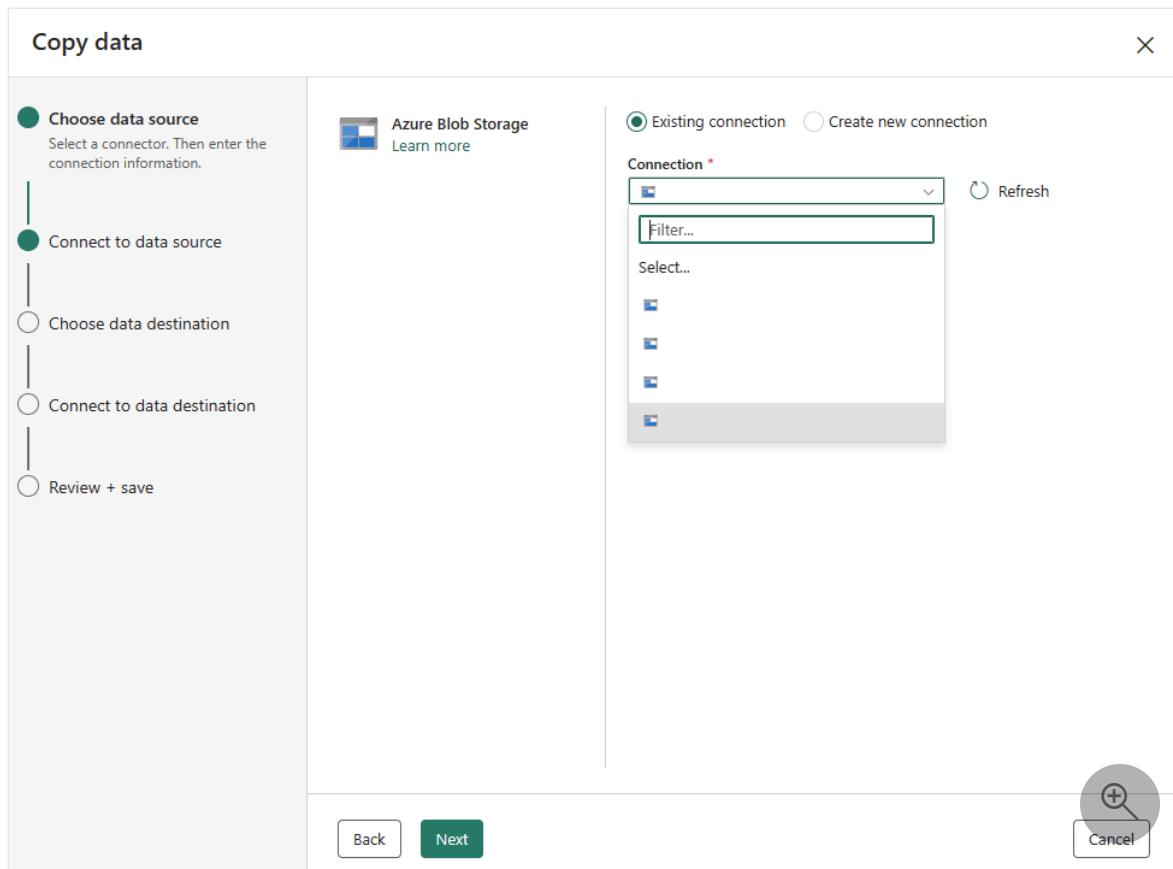


2. Create a connection to your data source by selecting **Create new connection**.



After you select **Create new connection**, fill in the required connection information and then select **Next**. For the details of connection creation for each type of data source, you can refer to each [connector article](#).

If you have existing connections, you can select **Existing connection** and select your connection from the drop-down list.



3. Choose the file or folder to be copied in this source configuration step, and then select **Next**.

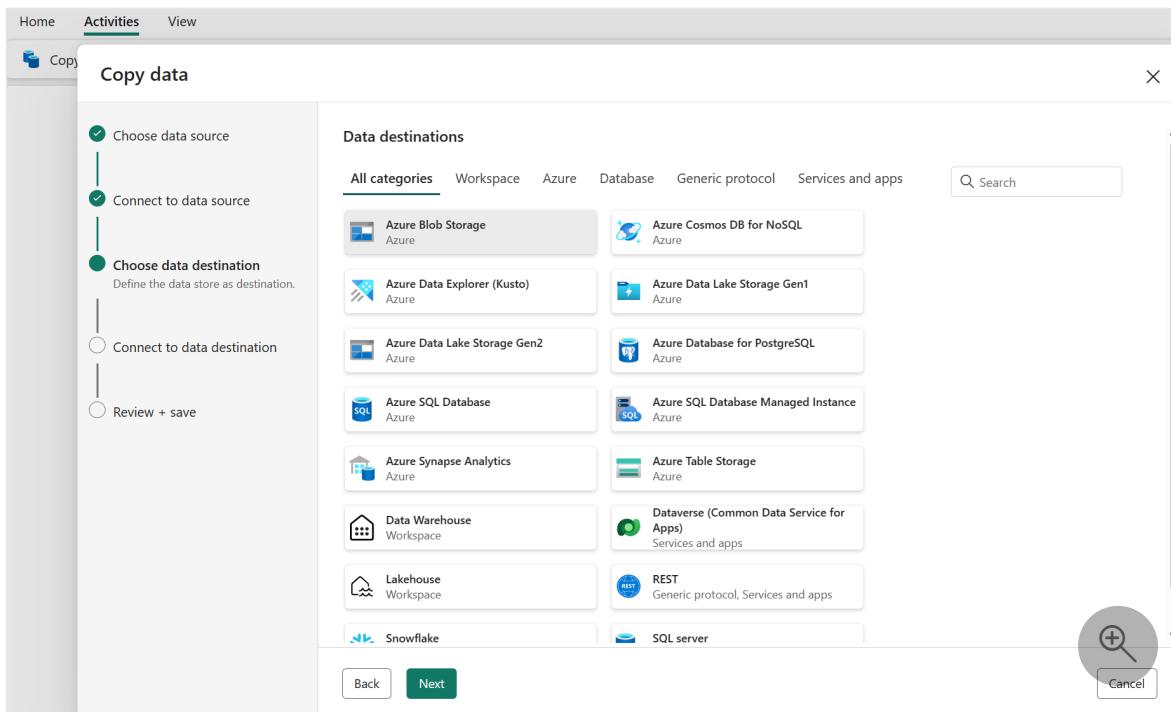
This screenshot shows the 'Copy data' wizard in step 2. The left sidebar shows the previous steps. The main area displays a file selection interface. On the left is a tree view of containers: '0129' (selected) containing '0101.csv', '0313.csv', '1.csv', '2.json', '21.json', '22.csv', '221.csv', '222.csv', 'Documents.txt', 'dbo.StudentsTest.txt', and 'new_adfdataflowtest.txt'. To the right is a preview of '0101.csv' data:

	abc Prop_0	abc Prop_1	abc Prop_2	abc Prop_3	abc Prop_4	abc Prop_5	abc Prop_6	abc Prop_7
1	4936965	170030	31	1	31.00	34100.93	0.02	0.03
2	1198310	197200	4758	1	23.00	29835.60	0.05	0.06
3	5582849	47184	7185	1	6.00	6787.08	0.00	0.05
4	2956227	132295	2296	1	41.00	54418.89	0.05	0.07
5	639744	171755	1756	1	4.00	7307.00	0.10	

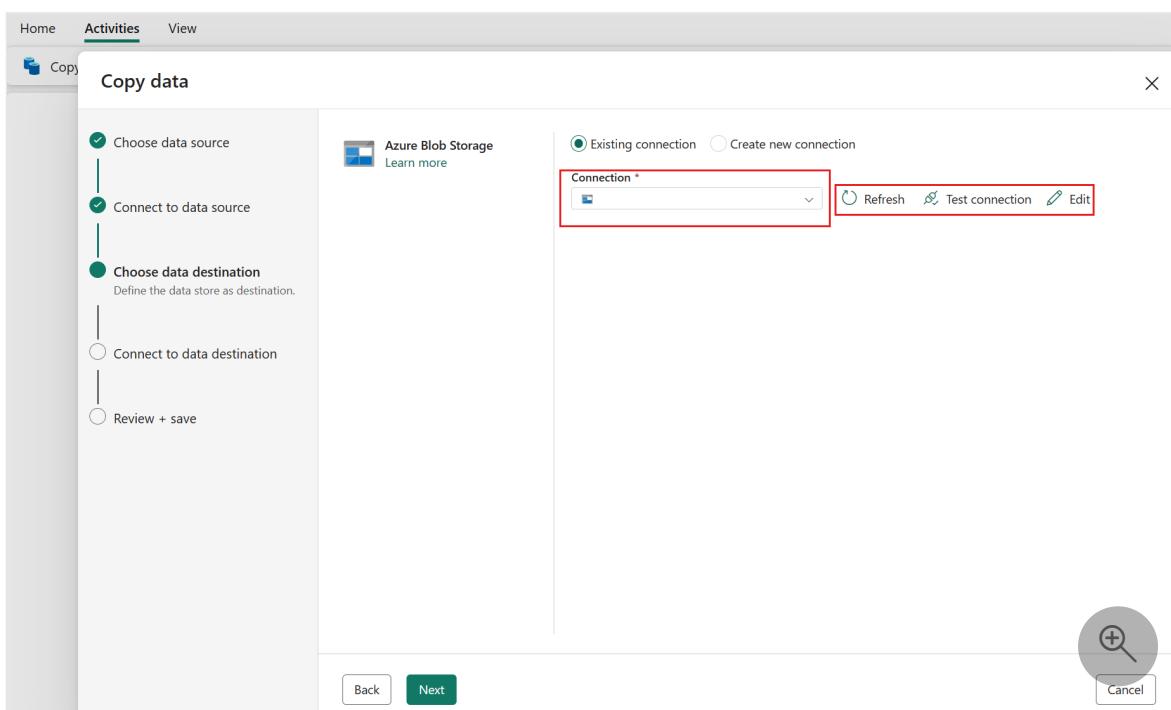
Buttons at the bottom are 'Back' and 'Next'.

Configure your destination

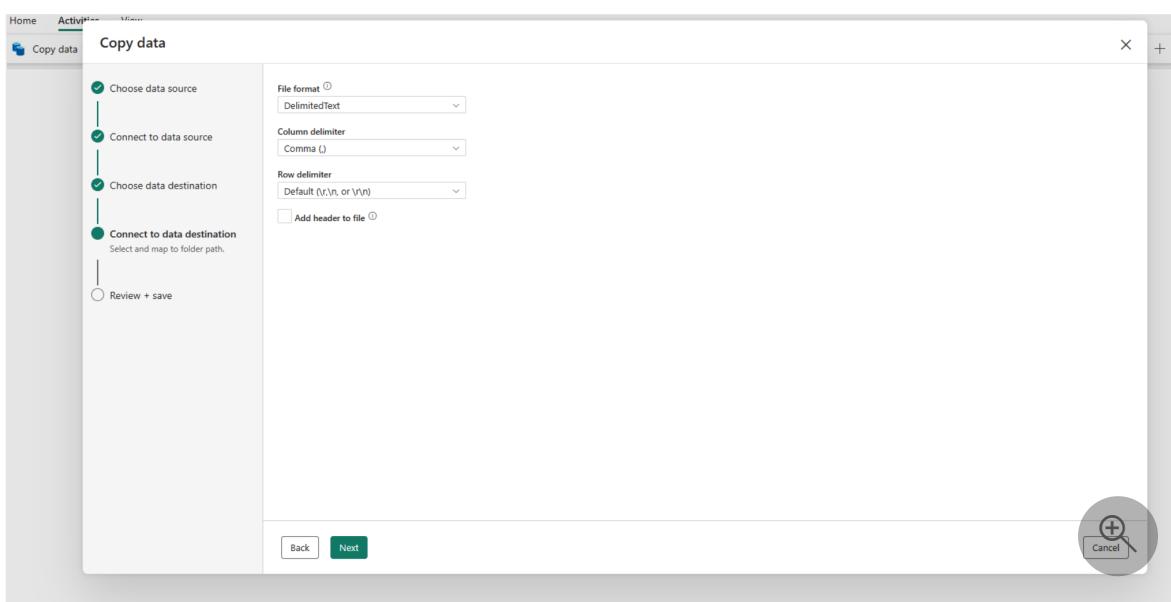
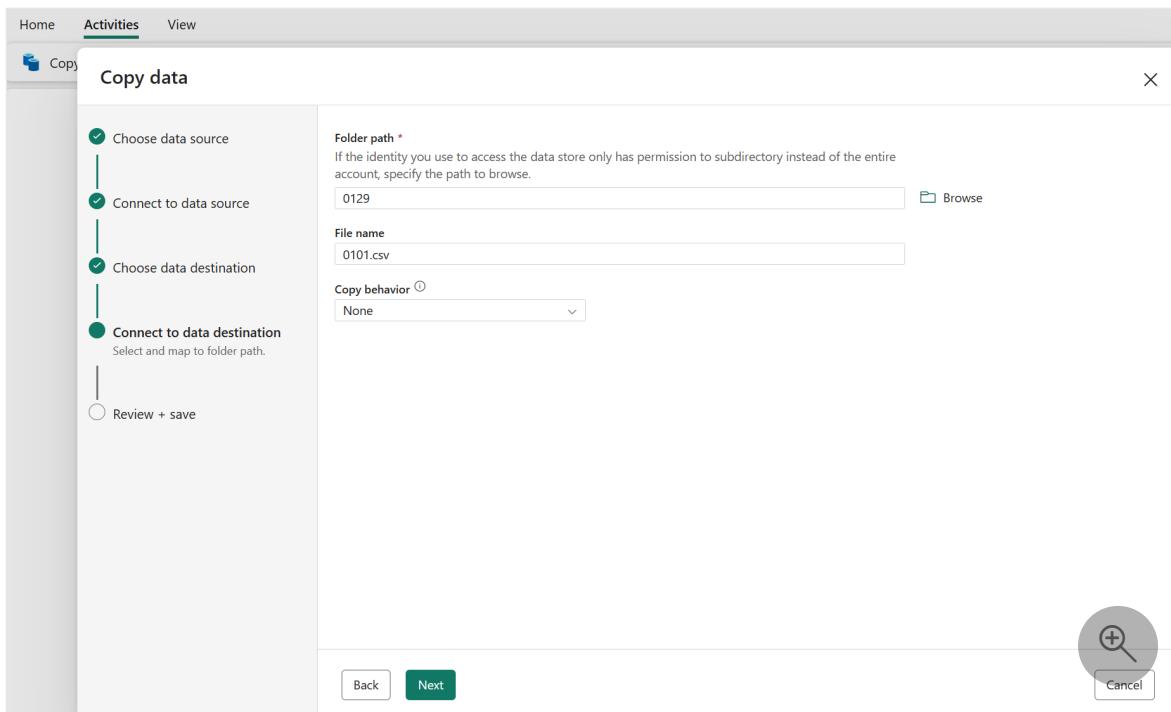
1. Select a data source type from the category. You'll use Azure Blob Storage as an example. Select **Azure Blob Storage**, and then select **Next**.



2. You can either create a new connection that links to a new Azure Blob Storage account by following the steps in the previous section or use an existing connection from the connection drop-down list. The capabilities of **Test connection** and **Edit** are available to each selected connection.

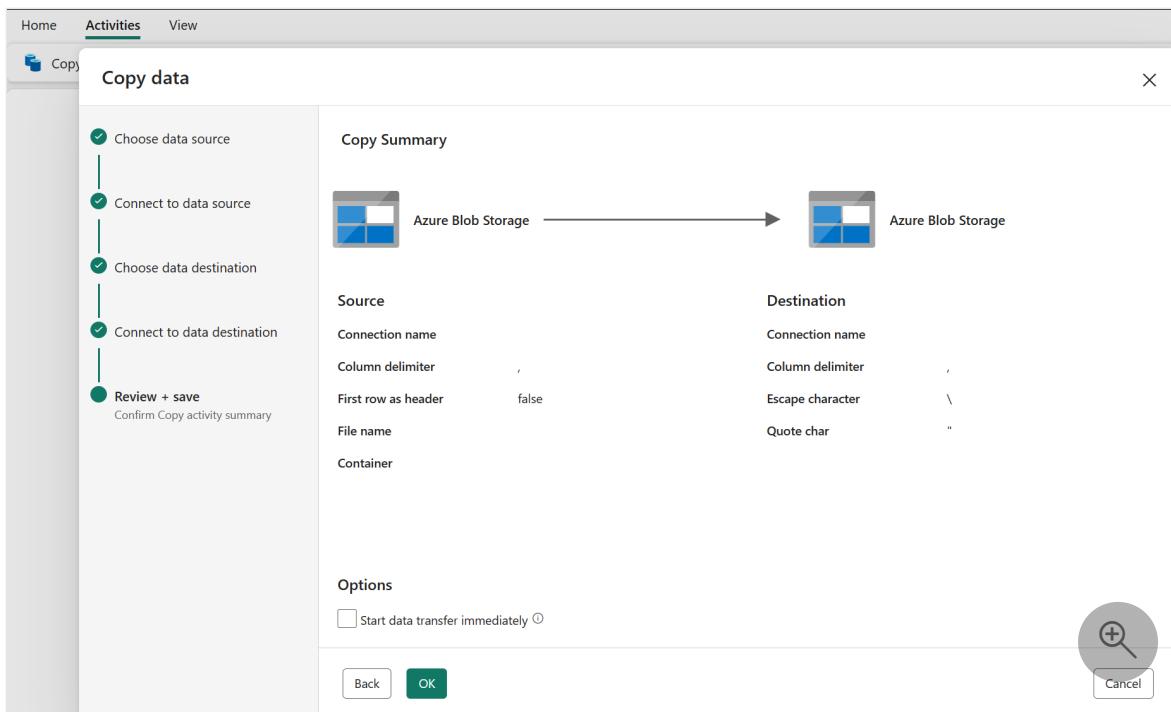


3. Configure and map your source data to your destination. Then select **Next** to finish your destination configurations.



Review and create your copy activity

1. Review your copy activity settings in the previous steps and select **OK** to finish. Or you can go back to the previous steps to edit your settings if needed in the tool.



Once finished, the copy activity will then be added to your data pipeline canvas. All settings, including advanced settings to this copy activity, are available under the tabs when it's selected.

Home Activities View

Copy data Dataflow Notebook Get metadata Lookup

Copy data

Copy_uk3

trash, copy, paste, refresh, save, run

General Source Destination Mapping Settings

Name * Copy_uk3 Learn more ↗

Description

Timeout ⓘ 0.12:00:00

Retry ⓘ 0

Advanced

Retry interval (sec) ⓘ 30

Secure output ⓘ

Secure input ⓘ +

Now you can either save your data pipeline with this single copy activity or continue to design your data pipeline.

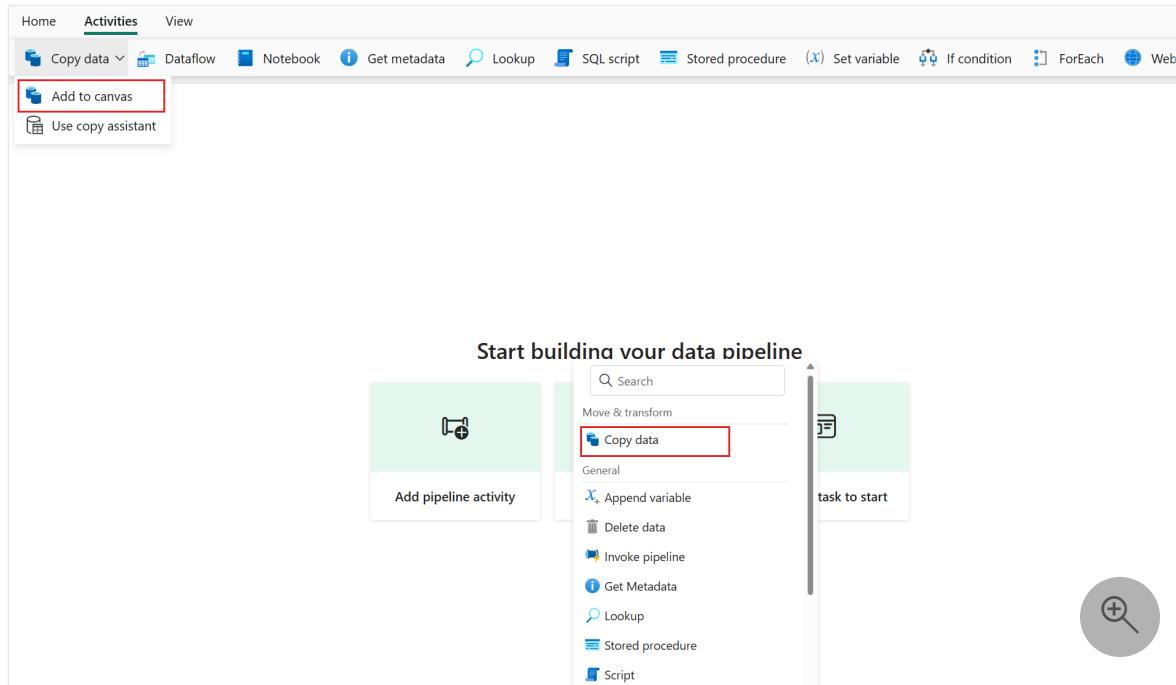
Add a copy activity directly

Follow these steps to add a copy activity directly.

Add a copy activity

1. Open an existing data pipeline or create a new data pipeline.

2. Add a copy activity either by selecting **Add pipeline activity** > **Copy activity** or by selecting **Copy data** > **Add to canvas** under the **Activities** tab.

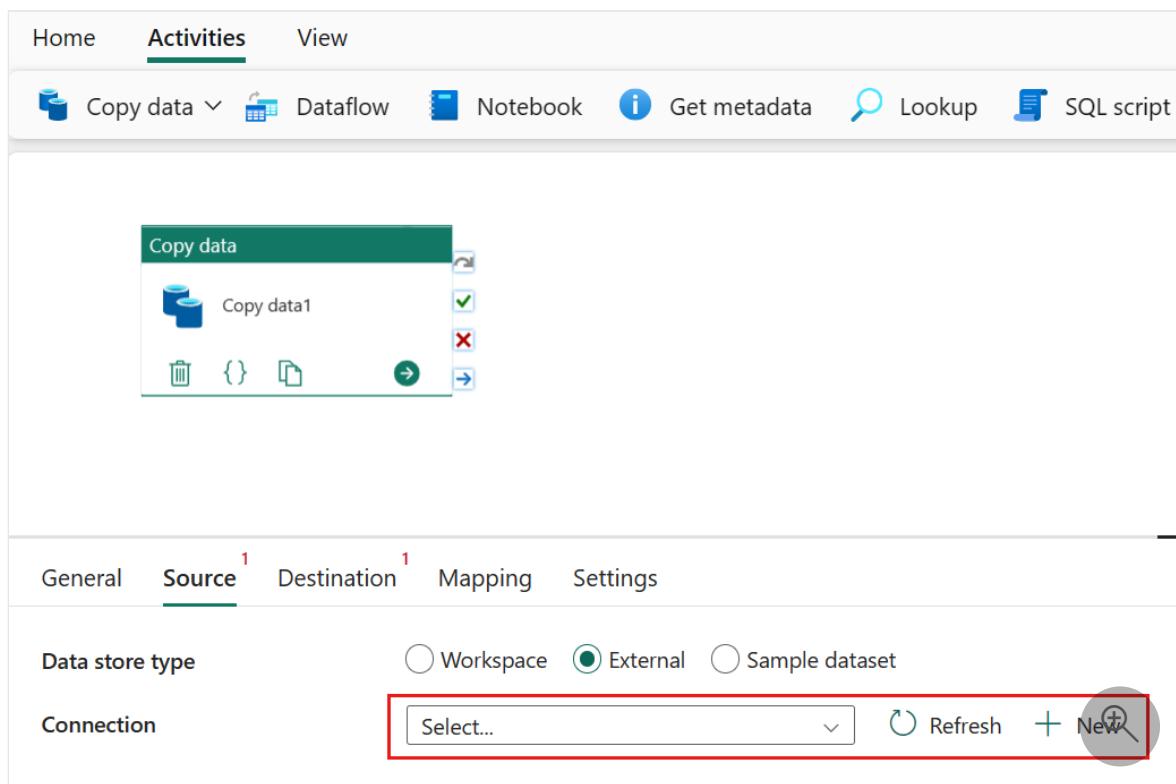


Configure your general settings under general tab

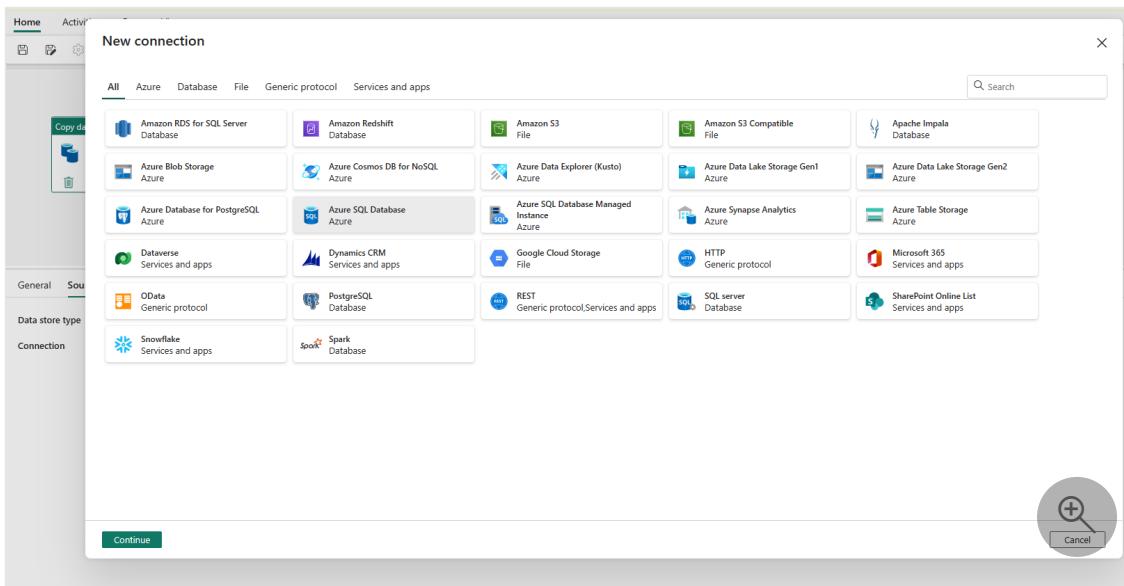
To learn how to configure your general settings, see [General](#).

Configure your source under the source tab

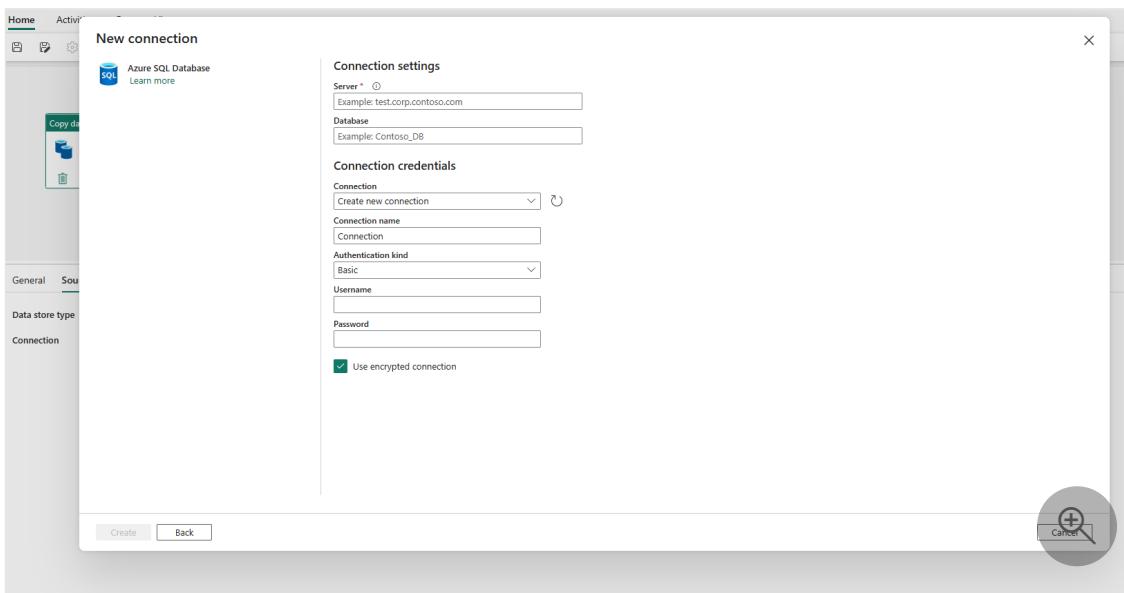
1. Select + New beside the **Connection** to create a connection to your data source.



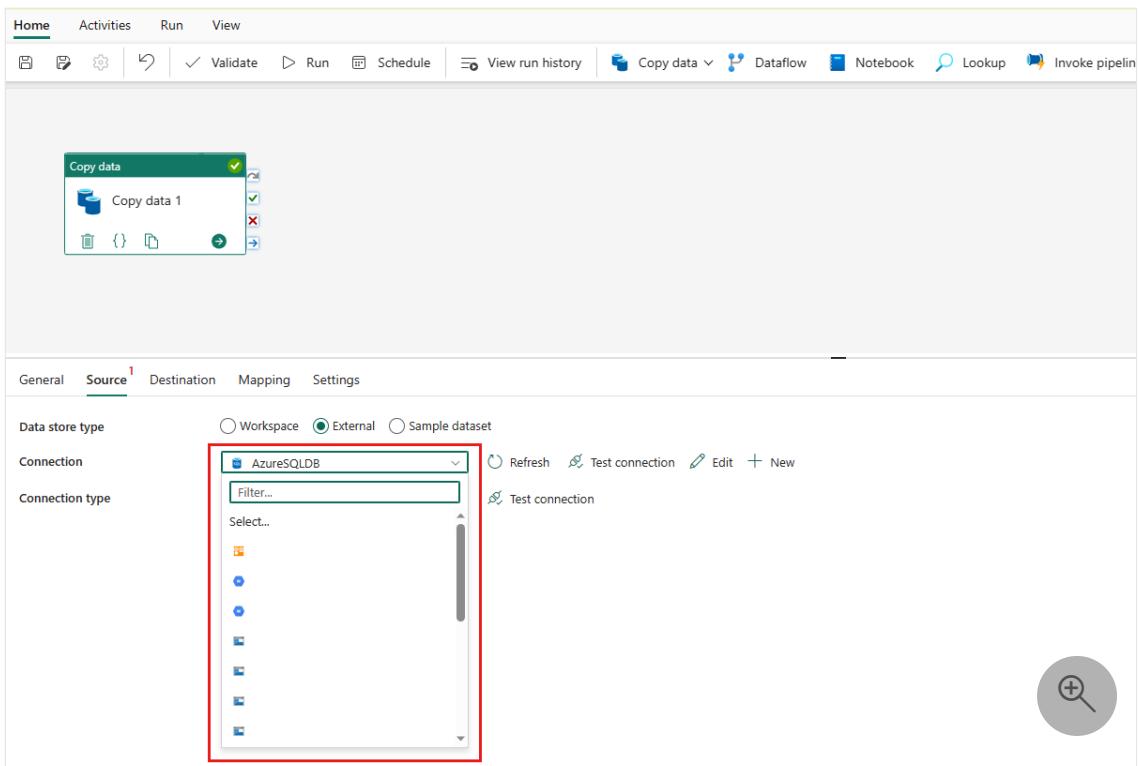
- a. Choose the data source type from the pop-up window. You'll use Azure SQL Database as an example. Select **Azure SQL Database**, and then select **Continue**.



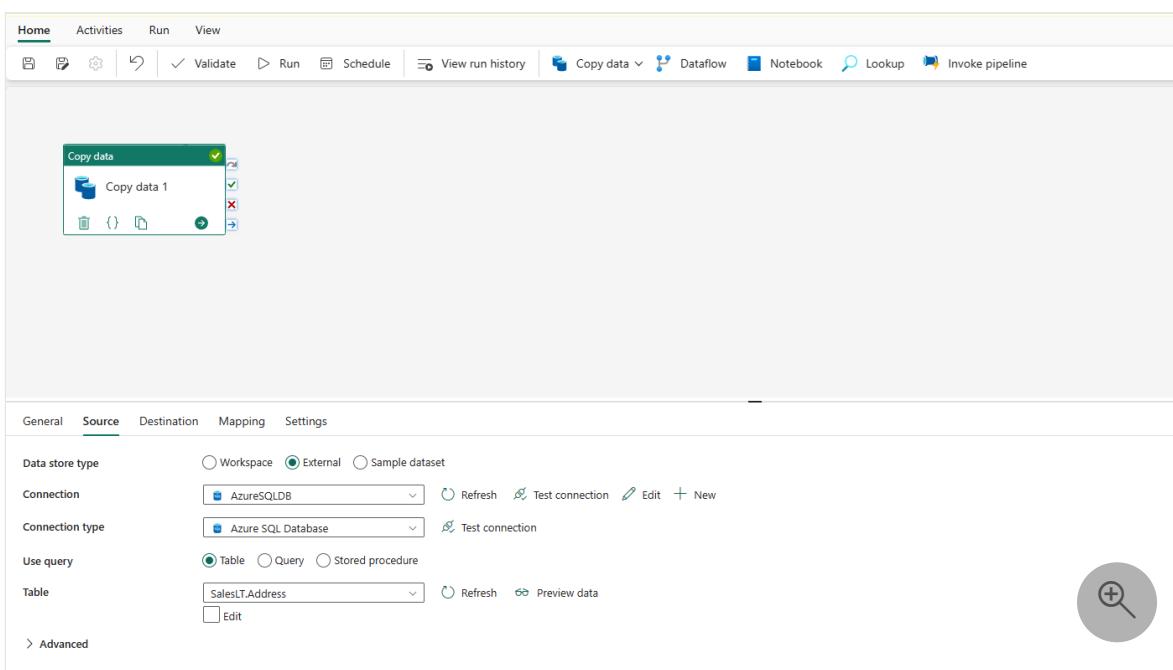
- b. It navigates to the connection creation page. Fill in the required connection information on the panel, and then select **Create**. For the details of connection creation for each type of data source, you can refer to each [connector article](#).



- c. Once your connection is created successfully, it takes you back to the data pipeline page. Then select **Refresh** to fetch the connection that you created from the drop-down list. You could also choose an existing Azure SQL Database connection from the drop-down directly if you already created it before. The capabilities of **Test connection** and **Edit** are available to each selected connection. Then select **Azure SQL Database** in **Connection type**.



2. Specify a table to be copied. Select **Preview data** to preview your source table. You can also use **Query** and **Stored procedure** to read data from your source.



3. Expand **Advanced** for more advanced settings.

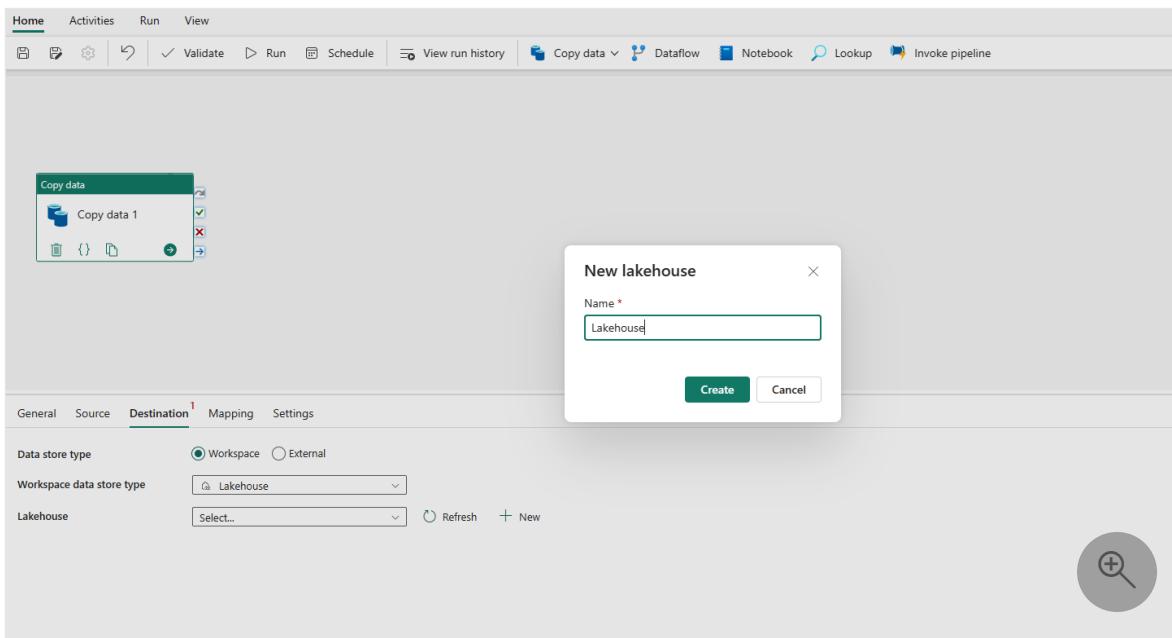
The screenshot shows the Azure Data Factory interface for configuring a 'Copy data' activity. The 'Source' tab is active. In the 'Connection' section, 'AzureSQLDB' is selected. Under 'Table', 'SalesLT.Address' is chosen. The 'Advanced' section is expanded, showing 'Query timeout (minutes)' set to 120, 'Isolation level' set to 'None', and 'Partition option' set to 'None'. A note at the bottom of this section advises previewing data to validate partition settings. A red box highlights this 'Advanced' section. On the right, there's a magnifying glass search icon.

Configure your destination under the destination tab

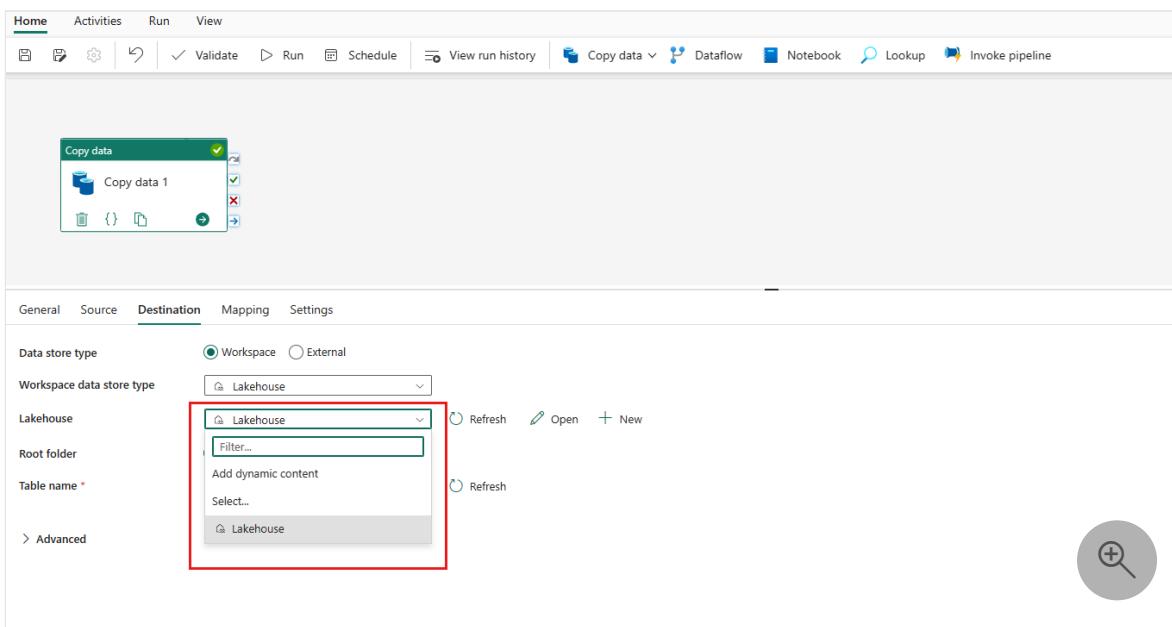
1. Choose your destination type. It could be either your internal first class data store from your workspace, such as Lakehouse, or your external data stores. You'll use Lakehouse as an example.

The screenshot shows the Azure Data Factory 'Activities' page. The 'Destination' tab is selected. In the 'Data store type' section, 'Workspace' is selected, and 'Lakehouse' is chosen from the dropdown. A red number '1' is shown above the 'Destination' tab. A search icon is located on the right side of the page.

2. Choose to use **Lakehouse** in **Workspace** data store type. Select **+ New**, and it navigates you to the Lakehouse creation page. Specify your Lakehouse name and then select **Create**.



3. Once your connection is created successfully, it takes you back to the data pipeline page. Then select **Refresh** to fetch the connection that you created from the drop-down list. You could also choose an existing Lakehouse connection from the drop-down directly if you already created it before.



4. Specify a table or set up the file path to define the file or folder as the destination. Here select **Tables** and specify a table to write data.

General Source **Destination** Mapping Settings

Data store type Workspace External

Workspace data store type

Lakehouse Refresh Open New

Root folder Tables Files

Table name * SalesAddress Edit

> Advanced

5. Expand Advanced for more advanced settings.

General Source **Destination** Mapping Settings

Data store type Workspace External

Workspace data store type

Lakehouse Refresh Open New

Root folder Tables Files

Table name * SalesAddress Edit

Advanced

Max rows per file

Table action Append Overwrite

Max concurrent connections

Now you can either save your data pipeline with this single copy activity or continue to design your data pipeline.

Configure your mappings under mapping tab

If the connector that you apply supports mapping, you can go to **Mapping** tab to configure your mapping.

1. Select **Import schemas** to import your data schema.

The screenshot shows the Azure Data Factory interface with the 'Copy data' activity selected. The top navigation bar includes Home, Activities, Run, View, and several icons for file operations like Open, Save, and Refresh. Below the navigation is a toolbar with Validate, Run, Schedule, View run history, and Copy data. The main area displays a 'Copy data' dialog with a green checkmark icon. Inside the dialog, there's a section for 'Copy data 1' with a trash can icon and a copy/paste icon. Below the dialog are tabs for General, Source, Destination, Mapping (which is selected), and Settings. Under the Mapping tab, there's a section for 'Type conversion settings'. A red box highlights the 'Import schemas' button. To its right are buttons for Preview source, New mapping (with a plus sign), Clear (with a minus sign), Reset (with a circular arrow), and Delete (with a trash can). A magnifying glass icon is also present.

2. You can see the auto mapping is shown up. Specify your **Source** column and **Destination** column. If you create a new table in the destination, you can customize your **Destination** column name here. If you want to write data into the existing destination table, you can't modify the existing **Destination** column name. You can also view the **Type** of source and destination columns.

This screenshot shows the 'Mapping' tab with detailed column mappings between a source and destination table. The table has four columns: Source, Type, Destination, and Type. The source table columns are: AddressID (int), AddressLine1 (nvarchar), AddressLine2 (nvarchar), City (nvarchar), StateProvince (nvarchar), CountryRegion (nvarchar), PostalCode (nvarchar), rowguid (uniqueidentifier), and ModifiedDate (datetime). The destination table columns are: AddressID (Int32), AddressLine1 (String), AddressLine2 (String), City (String), StateProvince (String), CountryRegion (String), PostalCode (String), rowguid (Guid), and ModifiedDate (DateTime). Each row shows a mapping from a source column to a destination column, with a plus sign (+) and a minus sign (-) icon for modifying the mapping. A magnifying glass icon is located in the bottom right corner of the mapping grid.

Source	Type	Destination	Type
AddressID	int	AddressID	Int32
AddressLine1	nvarchar	AddressLine1	String
AddressLine2	nvarchar	AddressLine2	String
City	nvarchar	City	String
StateProvince	nvarchar	StateProvince	String
CountryRegion	nvarchar	CountryRegion	String
PostalCode	nvarchar	PostalCode	String
rowguid	uniqueidentifier	rowguid	Guid
ModifiedDate	datetime	ModifiedDate	DateTime

Besides, you can select **+ New mapping** to add new mapping, select **Clear** to clear all mapping settings, and select **Reset** to reset all mapping **Source** column.

Configure your type conversion

Expand Type conversion settings to configure your type conversion if needed.

The screenshot shows the 'Mapping' tab selected in a top navigation bar. Below it, a section titled 'Type conversion settings' is expanded, indicated by a red border. This section contains five input fields: 'Allow data truncation' (checked), 'Treat boolean as number' (unchecked), and three empty boxes for 'DateTime format', 'DateTimeOffset format', and 'TimeSpan format'. At the bottom of this section are buttons for 'Import schemas', 'Preview source', 'New mapping', 'Clear', 'Reset', 'Delete', and a magnifying glass icon.

See the following table for the setting details.

Setting	Description
Allow data truncation	Allow data truncation when converting source data to destination with different type during copy. For example, from decimal to integer, from DateTimeOffset to Datetime.
Treat boolean as number	Treat boolean as number. For example, treat true as 1.
DateTime format	Format string when converting between dates without time zone offset and strings. For example, "yyyy-MM-dd HH:mm:ss.fff".
DateTimeOffset format	Format string when converting between dates with time zone offset and strings. For example, "yyyy-MM-dd HH:mm:ss.fff zzz".
TimeSpan format	Format string when converting between time periods and strings. For example, "dd.hh:mm:ss".
Culture	Culture information to be used when convert types. For example, "en-us", "fr-fr".

Configure your other settings under settings tab

The **Settings** tab contains the settings of performance, staging, and so on.

The screenshot shows the Azure Data Factory interface. At the top, there's a navigation bar with links like Home, Activities, Run, View, Validate, Run, Schedule, View run history, Copy data, Dataflow, Notebook, Lookup, and Invoke pipeline. Below the navigation bar, a 'Copy data' activity is selected, showing a preview of 'Copy data1'. The main area is titled 'Settings' and contains several configuration options:

- Intelligent throughput optimization**: A dropdown menu set to 'Auto' with an 'Edit' checkbox.
- Degree of copy parallelism**: An input field with a dropdown menu containing a checked 'Edit' option.
- Fault tolerance**: A dropdown menu.
- Enable logging**: An unchecked checkbox.
- Enable staging**: A checked checkbox.
- Staging settings**: A section with a 'Select...' dropdown, a 'Refresh' button, and a 'New' button. To the right is a search icon.

See the following table for the description of each setting.

Setting	Description
Intelligent throughput optimization	Specify to optimize the throughput. You can choose from: <ul style="list-style-type: none">• Auto• Standard• Balanced• Maximum When you choose Auto , the optimal setting is dynamically applied based on your source-destination pair and data pattern. You can also customize your throughput, and custom value can be 2-256 while higher value implies more gains.
Degree of copy parallelism	Specify the degree of parallelism that data loading would use.
Fault tolerance	When selecting this option, you can ignore some errors occurred in the middle of copy process. For example, incompatible rows between source and destination store, file being deleted during data movement, etc.
Enable logging	When selecting this option, you can log copied files, skipped files and rows
Enable staging	Specify whether to copy data via an interim staging store. Enable staging only for the beneficial scenarios.

Setting	Description
Staging account connection	When selecting Enable staging , specify the connection of an Azure storage data source as an interim staging store. Select + New to create a staging connection if you don't have it.

Next steps

- [Connector overview](#)
- [How to monitor pipeline runs](#)

Transform data with a ForEach activity

Article • 05/23/2023

The ForEach Activity defines a repeating control flow in a Microsoft Fabric pipeline. This activity is used to iterate over a collection and executes specified activities in a loop. The loop implementation of this activity is similar to a ForEach looping structure in programming languages.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Add a ForEach activity to a pipeline

This section describes how to use a ForEach activity in a pipeline.

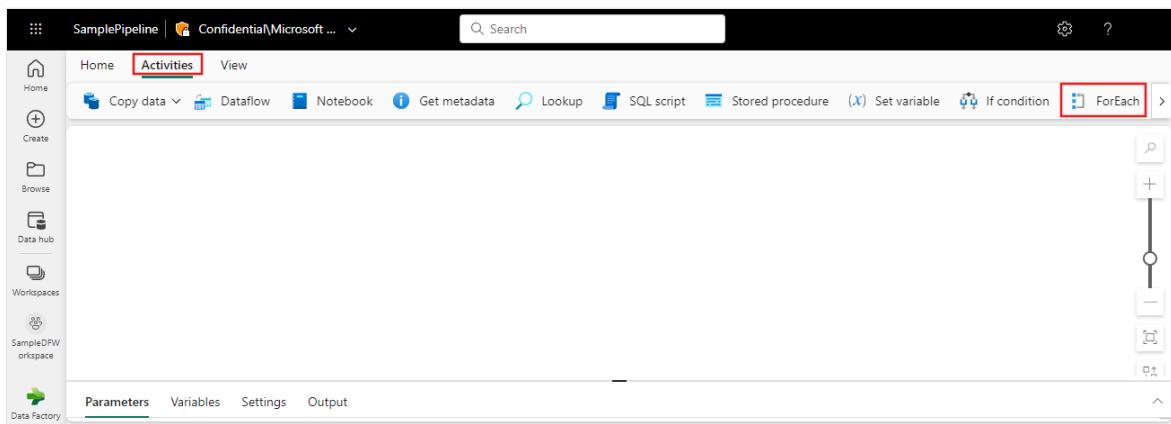
Prerequisites

To get started, you must complete the following prerequisites:

- A tenant account with an active subscription. [Create an account for free](#).
- A workspace is created.

Creating the activity

1. Create a new pipeline in your workspace.
2. Search for **ForEach** in the pipeline **Activities** pane, and select it to add it to the pipeline canvas. If you can't see it initially, use the arrow on the right side of the activities toolbar to scroll to the right to find it.



3. Select the new ForEach activity on the canvas if it isn't already selected.

A screenshot of the Azure Data Factory pipeline editor interface. The 'Activities' tab is selected. On the canvas, a 'ForEach' activity is displayed with a red border around its preview window. Inside the preview, it shows 'ForEach1' and 'Activities No activities'. Below the canvas, the 'General' tab is selected in the activity properties pane. Other tabs include 'Settings' and 'Activities (0)'. The 'Name' field is set to 'ForEach1'. There is also a 'Description' field. A red box highlights the 'General' tab. The left sidebar and bottom tabs are similar to the first screenshot.

4. In the General tab at the bottom of the canvas, enter a name for the activity in the Name property.

5. (Optional) You can also enter a description.

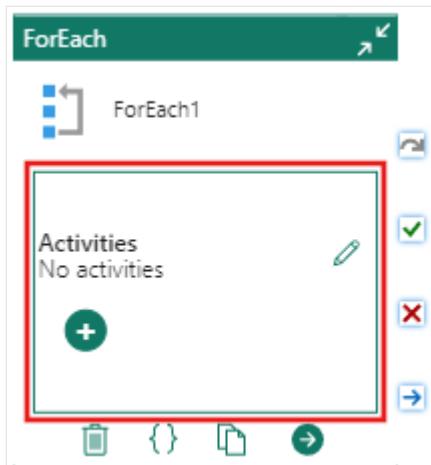
ForEach settings

Select the **Settings** tab, where you can specify whether processing of the items in the batch should be **Sequential** (or otherwise in parallel). You can also specify a maximum number of items to process at the same time with **Batch count**. Finally, you must specify a list of comma delimited **Items**, which can be parameterized or include dynamic content. Add a few items to the **Items** list as shown in the example.

The screenshot shows the Azure Data Factory Studio interface. On the left is a navigation sidebar with icons for Home, Create, Browse, Data hub, Workspaces, and SampleDFW workspace. The main area has a top navigation bar with Home, Activities (which is selected), View, Copy data, Dataflow, Notebook, Get metadata, Lookup, and a search bar. Below this is a workspace canvas with a green 'ForEach' activity node. Inside the 'ForEach' node is a sub-node labeled 'ForEach1'. To the right of the canvas are various control buttons like search, add, remove, and edit. At the bottom of the canvas are icons for save, new, delete, and run. Below the canvas is a settings pane with tabs for General, Settings (which is selected), and Activities (0). The 'Settings' tab contains three fields: 'Sequential' with an unchecked checkbox, 'Batch count' with an empty input field, and 'Items *' with a text input field containing 'Apple, Orange, Kiwi'. This last input field is highlighted with a red border. At the bottom of the settings pane is a note: 'Add dynamic content [Alt+Shift+D]'. The bottom right corner of the studio window has a 'Data Factory' icon.

ForEach activities

You'll also define an activity or activities to be performed on each of the items in the list, in the **ForEach Activities** pane.



Select the + button to add a new activity to the pane. You'll see a list of activities to choose. You can add multiple activities to the ForEach activity, and each is run on each of the items in the **Items** list. Whether the **Sequential** option is selected in the ForEach settings or not, each of the child activities in the ForEach activities pane are processed sequentially to one another for each item. However, if **Sequential** isn't selected, multiple items are processed in parallel, each of them running sequentially through the list of child activities specified.

The screenshot shows the 'General' tab of the 'Settings' pane for the 'ForEach' activity. The 'Sequential' checkbox is unchecked. The 'Items' dropdown contains the values 'Apple, Orange, Kiwi'. The context menu for the 'Script1' activity is open, showing options like 'Copy data', 'General', 'Append variable', 'Delete data', 'Invoke pipeline', 'Get Metadata', 'Lookup', 'Stored procedure', 'Script', 'Set variable', 'Web', 'WebHook', and 'Office365.Outlook'.

Referencing an item within an activity

Select one of the child activities in the **ForEach Activities** pane, and switch to its **Settings** tab. In this example, a **Stored Procedure** activity was selected. Populate the settings for the activity as you normally would select a connection and stored procedure. You can use the **@item()** iterator to refer to the current item being processed anywhere within an activity that supports dynamic content. Here the **@item()** was used as the value for the **FruitName** parameter that is passed to a stored procedure.

The screenshot shows the Azure Data Factory designer interface. At the top, there's a green header bar with the title "ForEach". Below it, the main workspace shows a "ForEach1" activity with a "Stored procedure..." child activity. A red box highlights the "Stored procedure..." icon. To the right, there's a "Activities" pane with icons for "Copy data1", "Script1", and a plus sign. Below the workspace are standard UI controls: a trash bin, a brace for grouping, a copy/paste icon, and a refresh arrow.

Below the workspace is a detailed configuration pane:

- General** tab (disabled)
- Settings** tab (selected, highlighted with a red box)

The "Settings" tab contains the following fields:

- Connection ***: A dropdown menu labeled "Select..." with a "Refresh" and "New" button.
- Stored procedure name ***: A dropdown menu labeled "Select..." with a "Refresh" button and a "Edit" link.

Under "Stored procedure parameters":

- A toolbar with "Import", "New", and "Delete" buttons.
- A table with columns: **Name**, **Type**, and **Value**.
- The first row shows a checkbox, the name "FruitName", the type "String", and the value "@item()". This last field is also highlighted with a red box.

Next steps

[How to monitor pipeline runs](#)

Use the Lookup activity to look up data from a data source

Article • 05/23/2023

The Fabric Lookup activity can retrieve a dataset from any of the data sources supported by Microsoft Fabric. You can use it to dynamically determine which objects to operate on in a subsequent activity, instead of hard coding the object name. Some object examples are files and tables.

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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Lookup activity reads and returns the content of a configuration file or table. It also returns the result of executing a query or stored procedure. The output can be a singleton value or an array of attributes, which can be consumed in a subsequent copy, transformation, or control flow activities like ForEach activity.

Prerequisites

To get started, you must complete the following prerequisites:

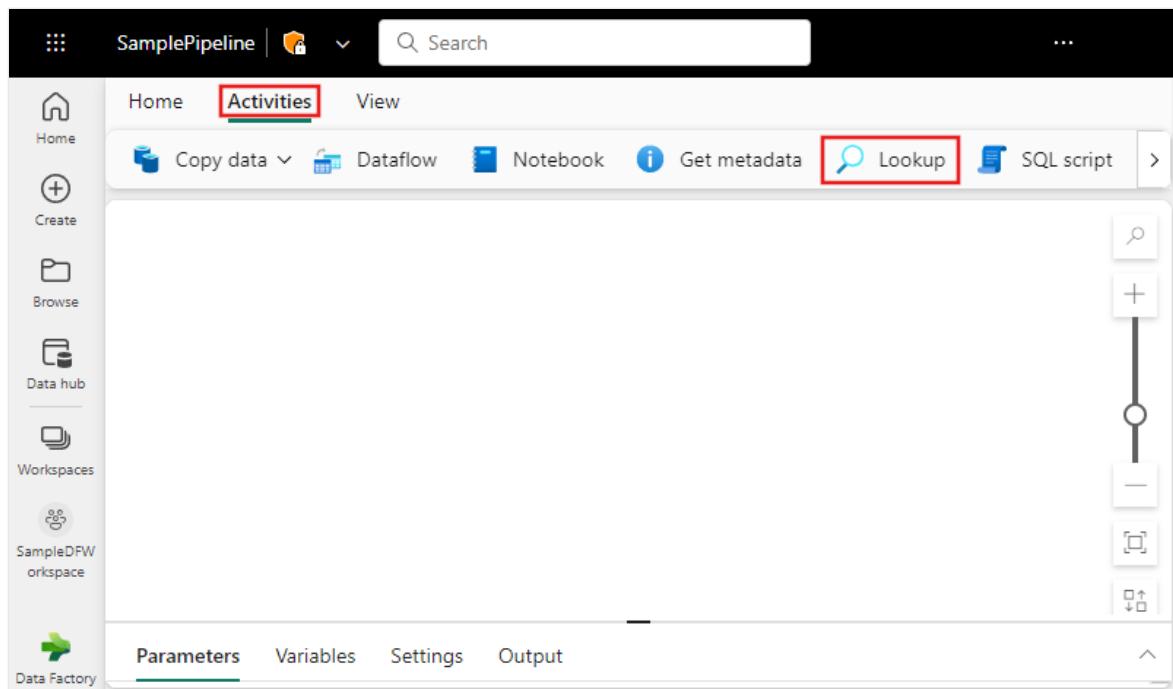
- A tenant account with an active subscription. [Create an account for free](#).
- A workspace is created.

Add a lookup activity to a pipeline with UI

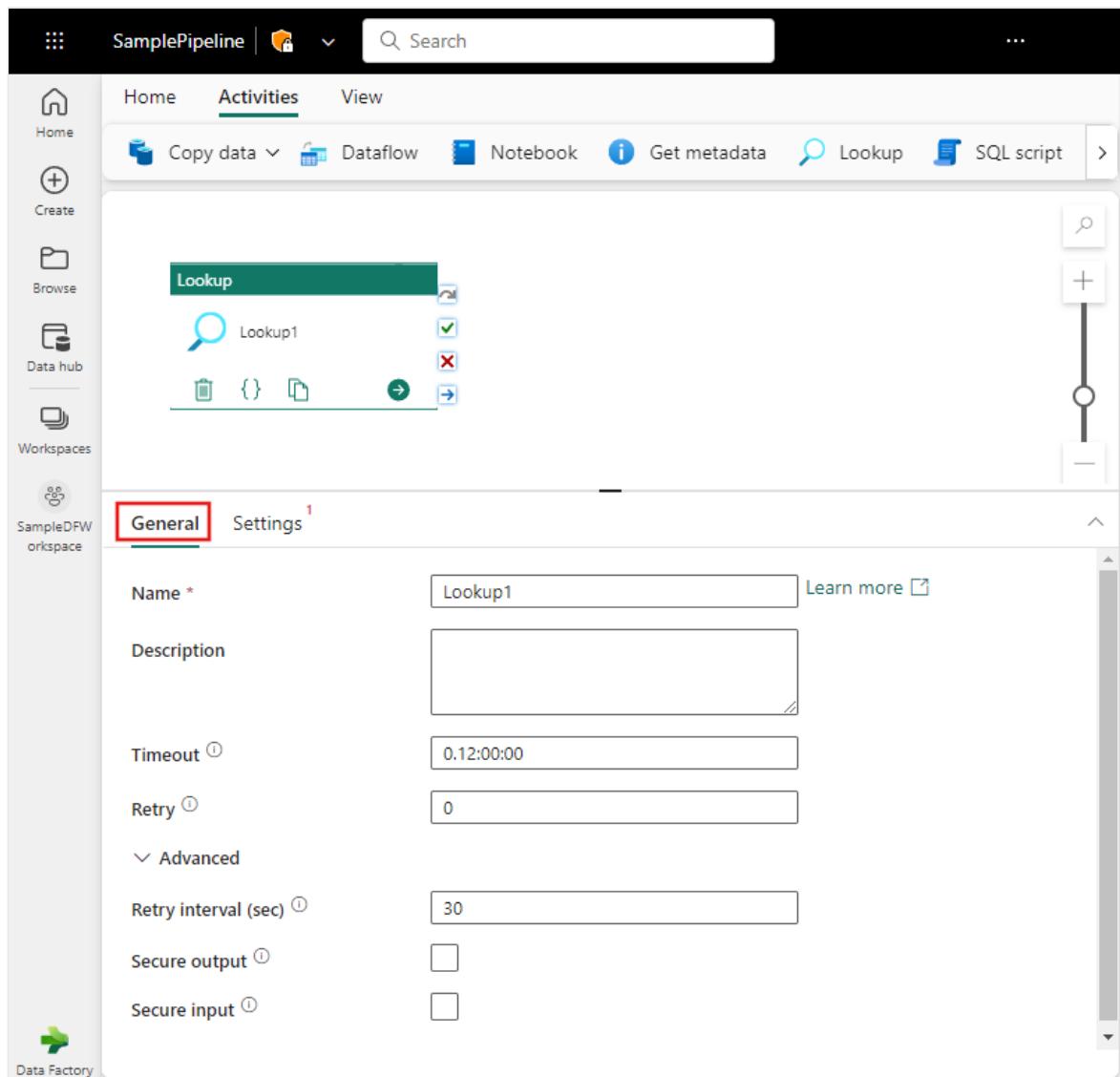
To use a Lookup activity in a pipeline, complete the following steps:

Creating the activity

1. Create a new pipeline in your workspace.
2. Search for Lookup in the pipeline **Activities** pane, and select it to add it to the pipeline canvas.



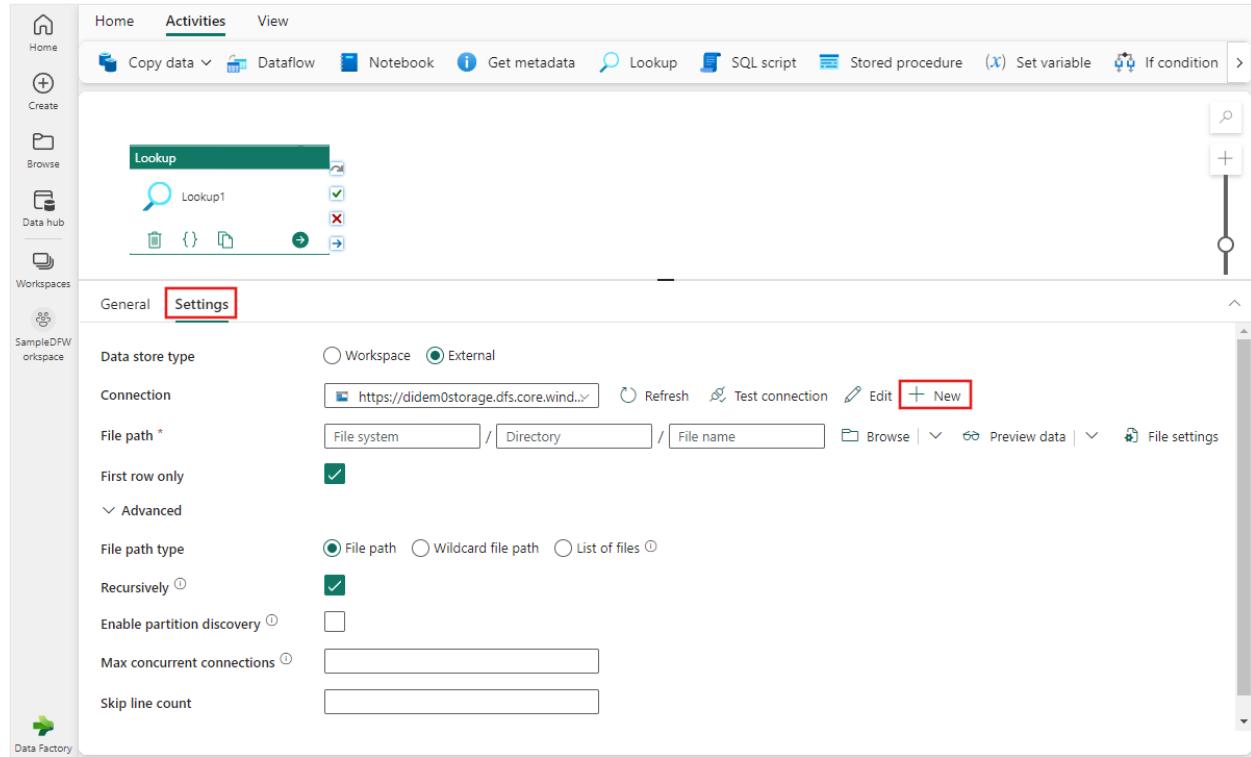
3. Select the new Lookup activity on the canvas if it isn't already selected.



Refer to the [General settings](#) guidance to configure the General settings tab.

Lookup settings

Select the **Settings** tab, select an existing connection from the **Connection** dropdown, or use the **+ New** button to create a new connection, and specify its configuration details.



The example in the previous image shows a blob storage connection, but each connection type has its own configuration details specific to the data source selected.

Supported capabilities

- The Lookup activity can return up to 5000 rows; if the result set contains more records, the first 5000 rows are returned.
- The Lookup activity output supports up to 4 MB in size; activity fails if the size exceeds the limit.
- The longest duration for Lookup activity before timeout is 24 hours.

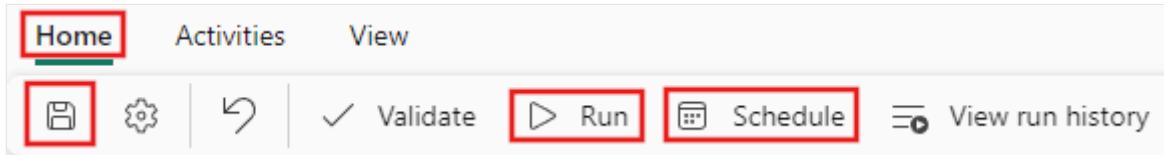
Note

When you use query or stored procedure to look up data, make sure to return one and exact one result set. Otherwise, Lookup activity fails.

Fabric supports the data stores listed in the [Connector overview](#) article. Data from any source can be used.

Save and run or schedule the pipeline

Switch to the **Home** tab at the top of the pipeline editor, and select the save button to save your pipeline. Select **Run** to run it directly, or **Schedule** to schedule it. You can also view the run history here or configure other settings.



Next steps

[How to monitor pipeline runs](#)

Transform data by running a notebook

Article • 05/23/2023

The Notebook activity in pipeline allows you to run Notebook created in Microsoft Fabric. You can create a Notebook activity directly through the Fabric user interface. This article provides a step-by-step walkthrough that describes how to create a Notebook activity using the Data Factory user interface.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Add a Notebook activity to a pipeline

This section describes how to use a Notebook activity in a pipeline.

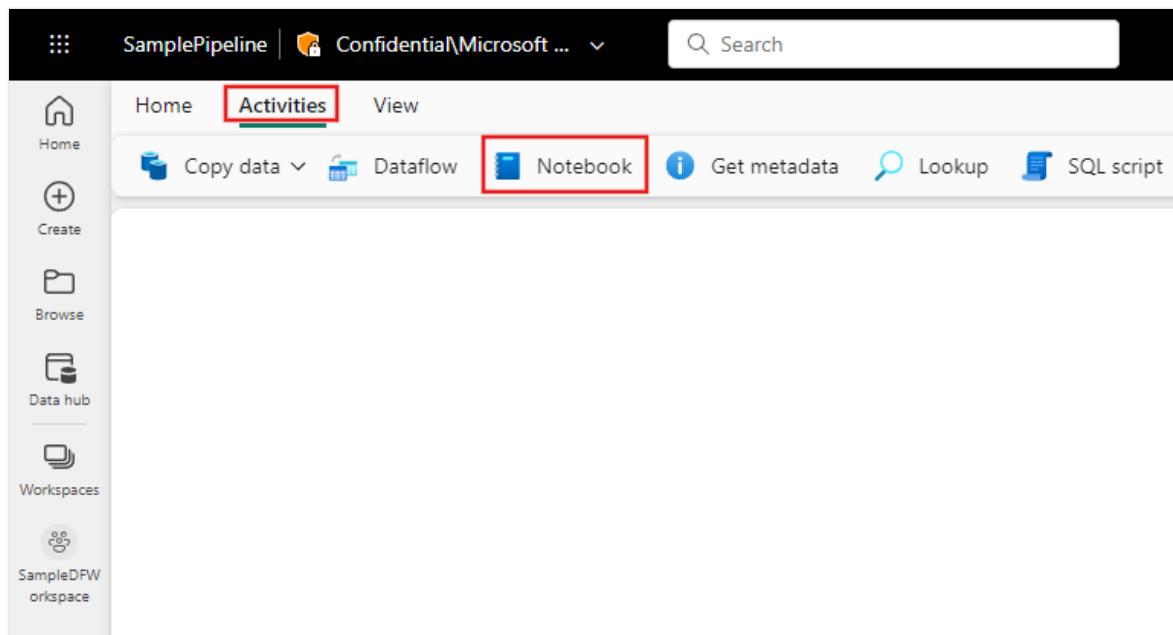
Prerequisites

To get started, you must complete the following prerequisites:

- A tenant account with an active subscription. [Create an account for free](#).
- A workspace is created.
- A notebook is created in your workspace. To create a new notebook, refer to [How to create Microsoft Fabric notebooks](#).

Creating the activity

1. Create a new pipeline in your workspace.
2. Search for Notebook in the pipeline **Activities** pane, and select it to add it to the pipeline canvas.



3. Select the new Notebook activity on the canvas if it isn't already selected.

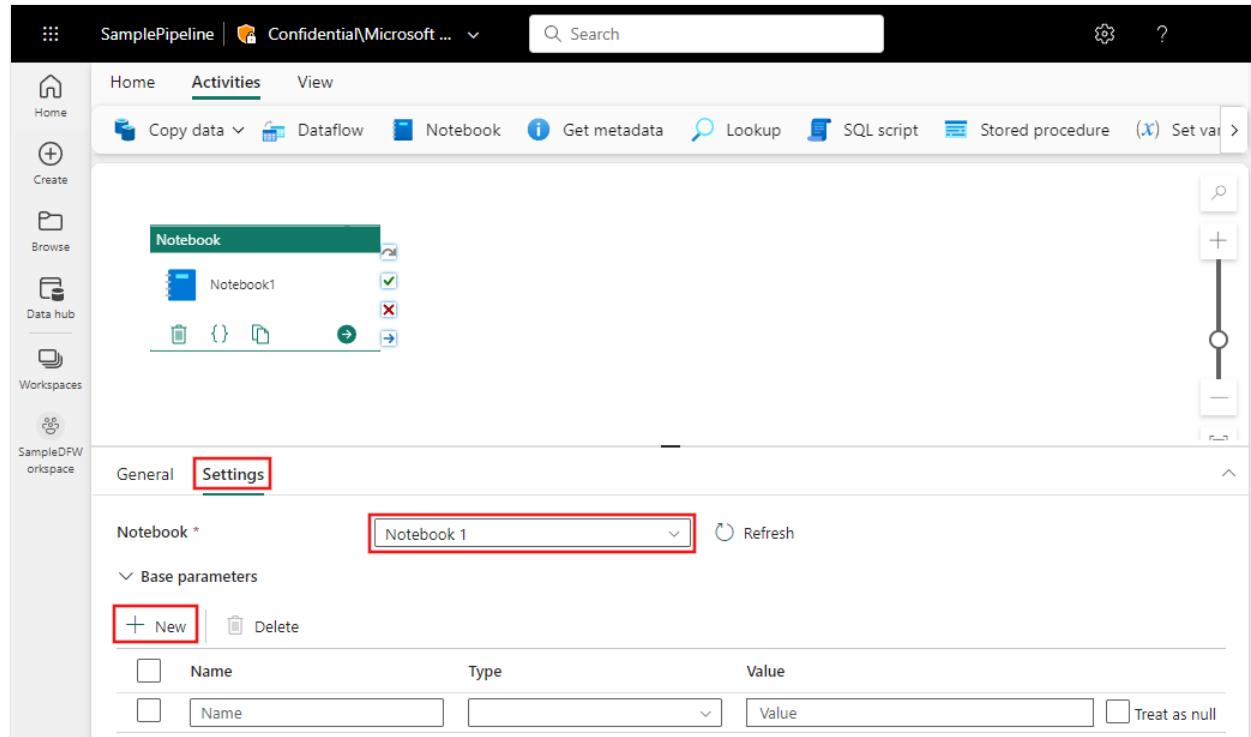
A screenshot of the Azure Data Factory 'Notebook' configuration page. The 'Activities' tab is selected. In the center, there's a preview area for a 'Notebook' named 'Notebook1'. Below this, there are two tabs: 'General' (highlighted with a red box) and 'Settings'. The 'General' tab contains the following fields:

Name *	Notebook1	Learn more 🔗
Description		
Timeout ⓘ	0:12:00:00	
Retry ⓘ	0	
Advanced		
Retry interval (sec) ⓘ	30	
Secure output ⓘ	<input type="checkbox"/>	
Secure input ⓘ	<input type="checkbox"/>	

Refer to the [General settings](#) guidance to configure the General settings tab.

Notebook settings

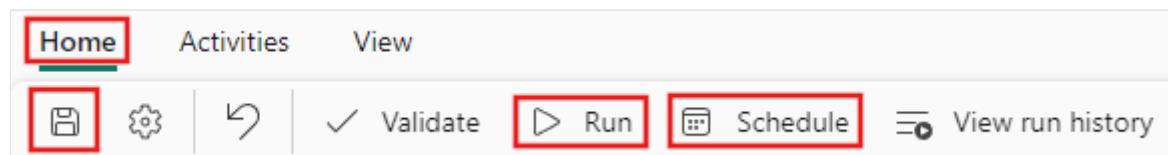
Select the **Settings** tab, select an existing notebook from the **Notebook** dropdown, and optionally specify any parameters to pass to the notebook.



The screenshot shows the Azure Data Factory pipeline editor interface. On the left, there's a sidebar with icons for Home, Create, Browse, Data hub, Workspaces, and SampleDFW workspace. The main area has tabs for Home, Activities, and View. Below these are buttons for Copy data, Dataflow, Notebook, Get metadata, Lookup, SQL script, Stored procedure, and Set variable. A search bar is at the top right. The main content area shows a list of Notebooks: 'Notebook' (selected) and 'Notebook1'. Below this is a 'General' tab and a 'Settings' tab, which is highlighted with a red box. Under the Settings tab, there's a dropdown labeled 'Notebook *' with 'Notebook 1' selected, also highlighted with a red box. A 'Refresh' button is next to it. A section for 'Base parameters' follows, with a '+ New' button highlighted with a red box. A table below shows columns for Name, Type, and Value, with a 'Treat as null' checkbox. The entire screenshot is framed by a red border.

Save and run or schedule the pipeline

Switch to the **Home** tab at the top of the pipeline editor, and select the save button to save your pipeline. Select **Run** to run it directly, or **Schedule** to schedule it. You can also view the run history here or configure other settings.



The screenshot shows the Azure Data Factory pipeline editor interface with the Home tab selected. At the top, there are buttons for Home, Activities, and View. Below these are four main buttons: a save icon (highlighted with a red box), a gear icon, a validate icon, and a refresh icon. To the right of these are 'Validate', 'Run' (highlighted with a red box), 'Schedule' (highlighted with a red box), and 'View run history'.

Next steps

[How to monitor pipeline runs](#)

How to use Script activity

Article • 05/23/2023

In this article, you learn how to add a new Script activity, add a new connection, and configure script content.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

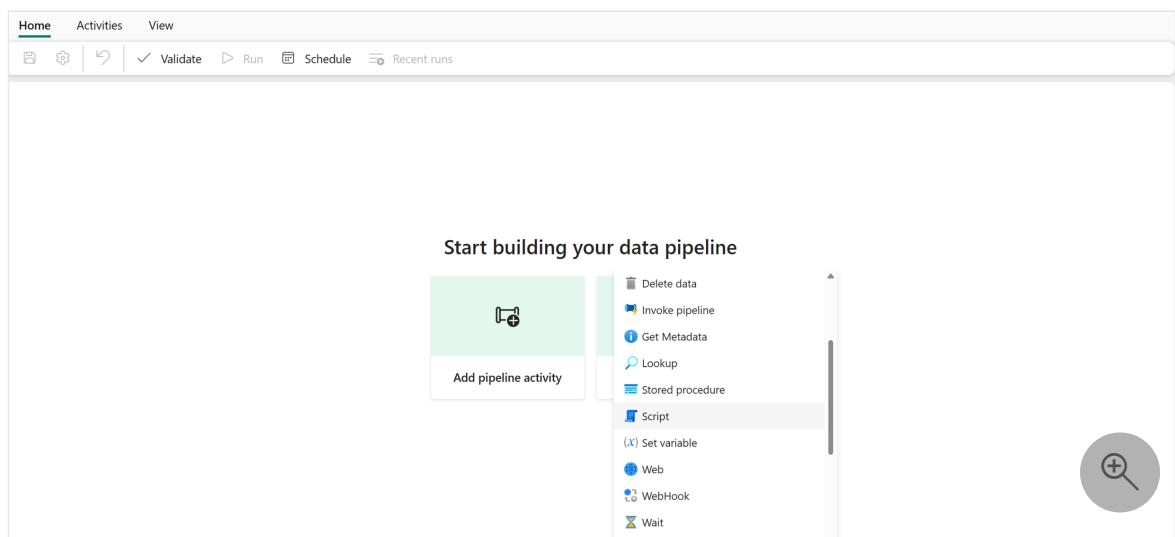
Prerequisites

To get started, you must complete the following prerequisites:

- A tenant account with an active subscription. Create an account for free.
- A workspace is created.

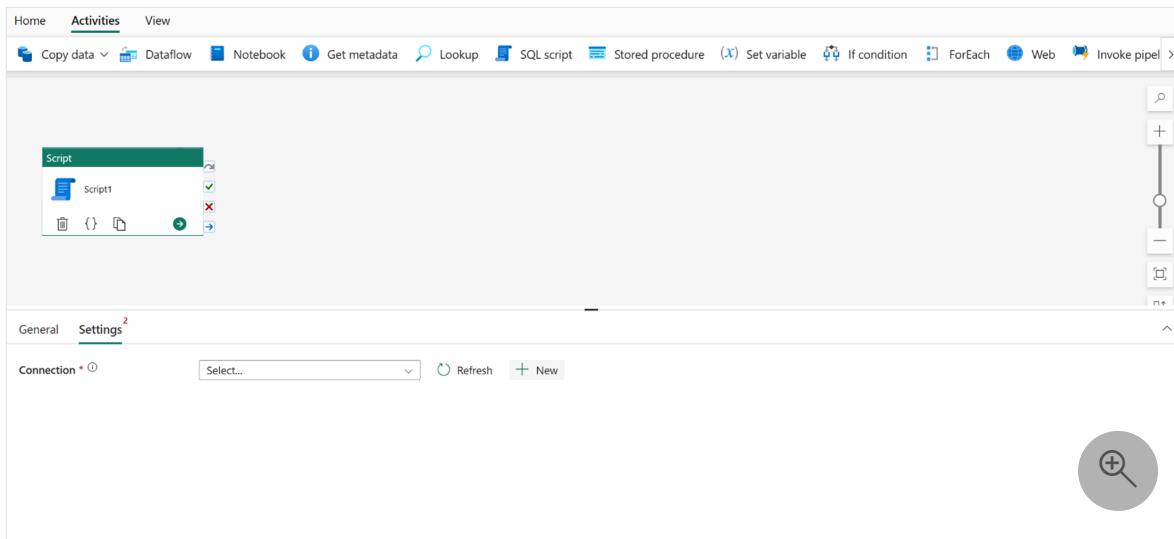
Step 1: Add a new Script activity in Pipeline canvas

1. Open an existing data pipeline or create a new data pipeline.
2. Select the **Script** activity.

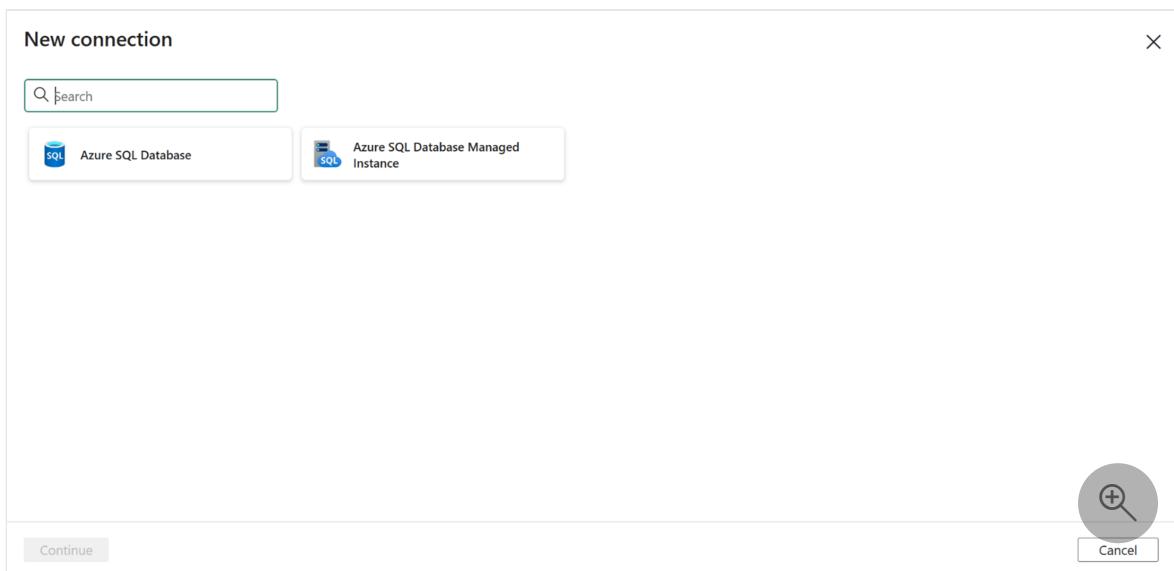


Step 2: Add a new connection for SQL

1. Select the **Settings** tab. Select **New** to create a new connection.



2. In the new popup window, choose the target SQL source.

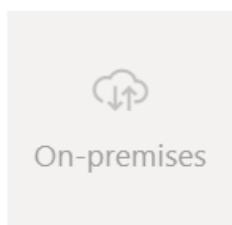


3. Create a new connection for the SQL source.

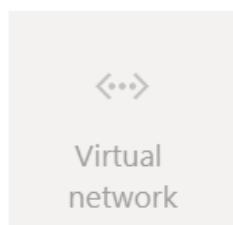
New connection



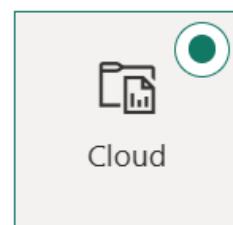
(i) Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".



On-premises



Virtual
network



Cloud

Connection name *

Connection type *

SQL Server

Server *

Ex: testazuresqlserver.database.windows.net

Database *

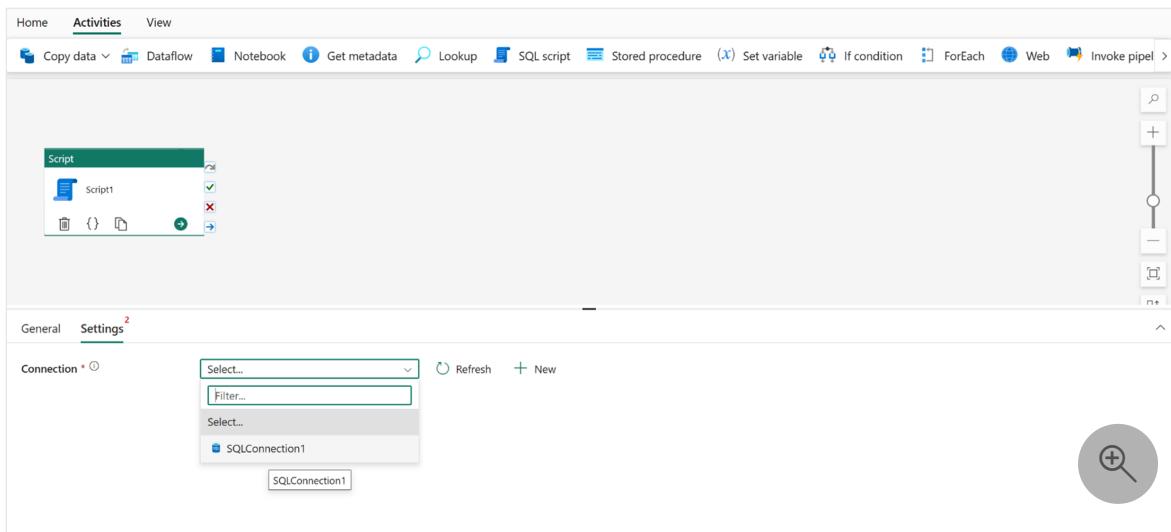
Ex: ContosoDB

Create

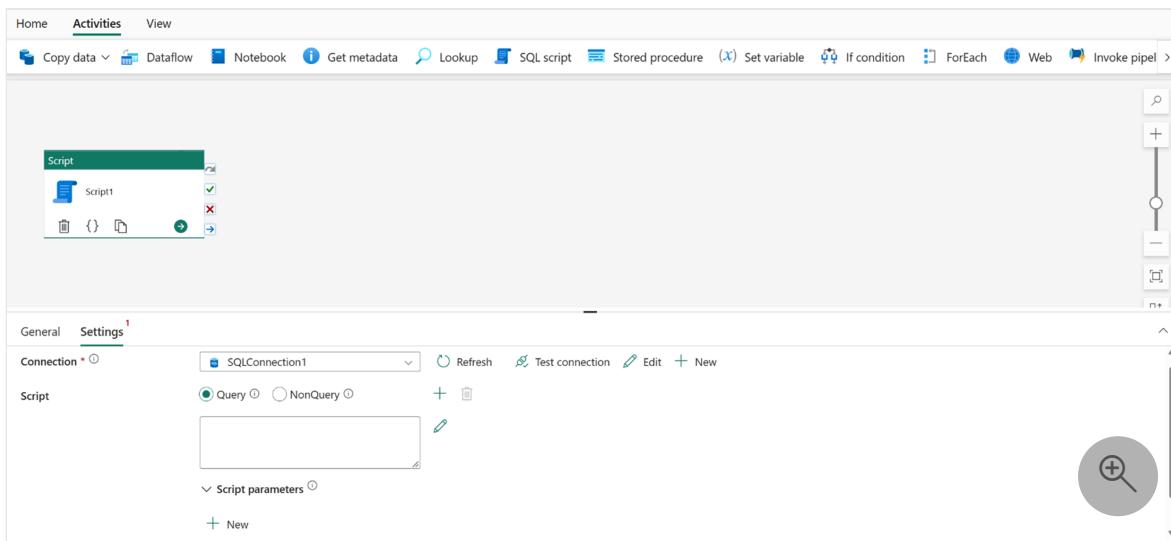
Close

Step 3: Configure script content

1. Select the connection you created in the previous step.



2. You can choose either **Query** to get a data result or **NonQuery** for any catalog operations.
3. Input the script content in the input box.
4. You can also define parameters for your script.



5. The Script activity is successfully created and you can run it directly.

Next steps

[Monitor pipeline runs](#)

How to monitor data pipeline runs in Microsoft Fabric

Article • 05/23/2023

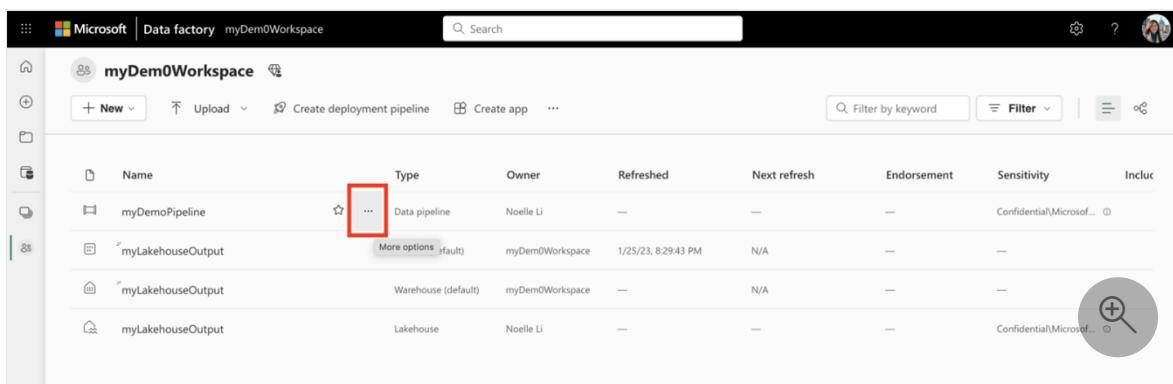
In this how-to guide, you'll learn the different ways to review and monitor your pipeline runs.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Monitor data pipeline runs

1. To monitor your data pipeline runs, hover over your pipeline in your workspace. Doing so will bring up three dots to the right of your pipeline name.



The screenshot shows the Microsoft Fabric Data factory workspace. On the left is a sidebar with icons for Home, New, Upload, Create deployment pipeline, Create app, and three dots. The main area shows a list of pipelines under 'myDem0Workspace'. The first item, 'myDemoPipeline', has a red box around its three-dot menu icon. A tooltip 'More options (1 fault)' is visible above the menu. The table columns are Name, Type, Owner, Refreshed, Next refresh, Endorsement, Sensitivity, and Includ. Other items listed are 'myLakehouseOutput' (Warehouse, default) and 'myLakehouseOutput' (Lakehouse).

2. Select the three dots to find a list of options. Then select **View run history**. This action opens a fly-out on the right side of your screen with all your recent runs and run statuses.

Microsoft | Data factory myDem0Workspace

+ New Upload Create deployment pipeline Create app ...

Search Filter

Name	Type	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Inclu...
myDemoPipeline	Data pipeline	Noelle Li	—	—	—	Confidential\Microsoft...	...
myLakehouseOutput	Dataset (default)	myDem0Workspace	Open	1/25/23, 8:29:43 PM	N/A	—	—
myLakehouseOutput	Warehouse (default)	myDem0Workspace	Delete	—	N/A	—	—
myLakehouseOutput	Lakehouse	Noelle Li	Settings	—	—	—	...
			Add to Favorites	—	—	Confidential\Microsoft...	...
			View lineage	—	—	—	...
			View run history	—	—	—	...

Microsoft | Data factory myDem0Workspace

+ New Upload Create deployment pipeline Create app ...

Search Filter

Name	Type	Owner
myDemoPipeline	Data pipeline	Noelle Li
myLakehouseOutput	Dataset (default)	myDem0Workspace
myLakehouseOutput	Warehouse (default)	myDem0Workspace
myLakehouseOutput	Lakehouse	Noelle Li

Recent runs

Refresh Export as csv

myDemoPipeline

1 of 1

Run start	Run end	Duration	Triggered by	Status
1/25/2023, 8:50:01 PM	1/25/2023, 8:50:23 PM	00:00:22	ScheduleTrigger	✓ Succeeded
1/25/2023, 8:45:00 PM	1/25/2023, 8:45:22 PM	00:00:22	ScheduleTrigger	✓ Succeeded
1/25/2023, 8:40:00 PM	1/25/2023, 8:40:21 PM	00:00:21	ScheduleTrigger	✓ Succeeded
1/25/2023, 8:35:00 PM	1/25/2023, 8:35:19 PM	00:00:19	ScheduleTrigger	✓ Succeeded
1/25/2023, 8:32:15 PM	1/25/2023, 8:32:37 PM	00:00:22	Manual	✓ Succeeded
1/25/2023, 8:29:54 PM	1/25/2023, 8:30:19 PM	00:00:25	Manual	✓ Succeeded

3. Use the Filter to find specific data pipeline runs. You can filter on **Status** or on **End time**.

myDemoPipeline

Recent runs

⌚ Refresh ⚡ Export as csv

Run start	Run end	Duration	Triggered by	Status
1/25/2023, 8:50:01 PM	1/25/2023, 8:50:23 PM	00:00:22	ScheduleTrigger	Success
1/25/2023, 8:45:00 PM	1/25/2023, 8:45:22 PM	00:00:22	ScheduleTrigger	Success
1/25/2023, 8:40:00 PM	1/25/2023, 8:40:21 PM	00:00:21	ScheduleTrigger	Success
1/25/2023, 8:35:00 PM	1/25/2023, 8:35:19 PM	00:00:19	ScheduleTrigger	Success
1/25/2023, 8:32:15 PM	1/25/2023, 8:32:37 PM	00:00:22	Manual	Success
1/25/2023, 8:29:54 PM	1/25/2023, 8:30:19 PM	00:00:25	Manual	Success

Filter Clear applied filters

Status

- All
- Succeeded
- In Progress
- Queued
- Failed
- Canceled

End time

- Last 24 hours
- Last 7 days
- Last 30 days

4. Select one of your pipeline runs to view detailed information. You'll be able to view what your pipeline looks like and view more properties like Run ID or errors if your pipeline run failed.

Microsoft | Data factory myDem0Workspace

+ New Upload Create deployment pipeline Create app

myDemoPipeline

Recent runs

⌚ Refresh ⚡ Export as csv

Filter

Name	Type	Owner	Run start	Run end	Duration	Triggered by	Status
myDemoPipeline	Data pipeline	Noelle L	1/25/2023, 8:50:01 PM	1/25/2023, 8:50:23 PM	00:00:22	ScheduleTrigger	Success
myLakehouseOutput	Dataset (default)	myDem0	1/25/2023, 8:45:00 PM	1/25/2023, 8:45:22 PM	00:00:22	ScheduleTrigger	Success
myLakehouseOutput	Warehouse (default)	myDem0	1/25/2023, 8:40:00 PM	1/25/2023, 8:40:21 PM	00:00:21	ScheduleTrigger	Success
myLakehouseOutput	Lakehouse	Noelle L	1/25/2023, 8:35:00 PM	1/25/2023, 8:35:19 PM	00:00:19	ScheduleTrigger	Success
			1/25/2023, 8:32:15 PM	1/25/2023, 8:32:37 PM	00:00:22	Manual	Success
			1/25/2023, 8:29:54 PM	1/25/2023, 8:30:19 PM	00:00:25	Manual	Success



The screenshot shows the Microsoft Data Factory interface for managing pipeline runs. At the top, it displays the pipeline name "myDemoPipeline". Below the pipeline name, there are tabs for "List" and "Gantt". The main area shows a hierarchical tree of activities: a "ForEach" activity containing a "Copy" activity. To the right, a detailed view of the "Copy" activity is shown in a pop-up window. The details include:

- Owner**
- Run by**
- Start time**: 1/25/2023, 8:50:01 PM
- End time**: 1/25/2023, 8:50:23 PM
- Status**: Succeeded
- Pipeline run ID**: Pipeline run ID

At the bottom of the main screen, there is a search bar and a magnifying glass icon.

5. To find additional information on your pipeline runs **Input** and **Output**, hover over an activity row and select either the **Input** or **Output** icon. Details will be shown in a pop-up.

The screenshot shows the "Activity runs" section of the Data Factory interface. It lists six activity runs, including the "Copy" activity from the previous screenshot. The "Copy" activity row is highlighted with a red box. Next to the activity name, there are two small icons: a blue square with a white arrow pointing right (Input) and a blue square with a white arrow pointing left (Output). The table columns are:

Activity name	Run start	Duration	Status	Run ID
ForEach_4i1	1/25/2023, 8:50:01 PM	00:00:22	Succeeded	d3c80328-6a8a-47b7-9a1e-d32f4ca17b3d
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:13	Succeeded	b2b1b978-a1b5-487f-b581-361d281ebaba
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:16	Succeeded	a72f1663-d3c7-4191-a8bc-7ba2ff4414f0b
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:14	Succeeded	1f3b6546-4076-4d7a-b980-a129a05f2966
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:13	Succeeded	b2b1b978-a1b5-487f-b581-361d281ebaba
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:16	Succeeded	a72f1663-d3c7-4191-a8bc-7ba2ff4414f0b

Input

Copy to clipboard

```
{
  "source": {
    "type": "AzureSqlSource",
    "partitionOption": "None",
    "datasetSettings": {
      "type": "AzureSqlTable",
      "typeProperties": {
        "schema": "SalesLT",
        "table": "ProductDescription"
      }
    }
  }
}
```

Activity runs

Activity name	Run start	Duration	Status	Run ID
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:13	Succeeded	b2b1b978-a1b5-487f-b581-361d281ef6aba
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:16	Succeeded	a72f1663-d3c7-4191-a8bc-7ba244140b0
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:14	Succeeded	1f3b6546-4076-4d7a-b980-a129a05f2966

6. To view performance details, hover over an activity row and select the glasses icon. Performance details will pop up.

Activity runs

Activity name	Run start	Duration	Status	Run ID
ForEach_4i1	1/25/2023, 8:50:01 PM	00:00:22	Succeeded	d3c80328-6a8a-47b7-9a1e-d32f4ca17b3d
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:13	Succeeded	b2b1b978-a1b5-487f-b581-361d281ef6aba
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:16	Succeeded	a72f1663-d3c7-4191-a8bc-7ba244140b0
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:14	Succeeded	1f3b6546-4076-4d7a-b980-a129a05f2966

All pipeline runs > **myDemoPipeline**

Copy data details

Copy_4i1

Source: Azure SQL Database → **Destination**: Lakehouse

Data read: 125.828 KB
Rows read: 762

Data written: 78.212 KB
Files written: 1
Rows written: 762

Status: Succeeded
Start time: 1/25/2023, 8:50:02 PM

Pipeline run activity ID: b2b1b978-a1b5-487f-b581-361d281ef6aba

Throughput: 31.457 KB/s
Total duration: 00:00:10

> Duration breakdown
> Advanced

Activity runs

Activity name	Run start	Duration	Status
ForEach_4i1	1/25/2023, 8:50:01 PM	00:00:22	Succeeded
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:13	Succeeded
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:16	Succeeded
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:14	Succeeded

More details can be found under Duration breakdown and Advanced.

Copy data details

Data read:		Data written:	
Rows read:	762	Files written:	1
		Rows written:	762

Status	✓ Succeeded
Start time	1/25/2023, 8:50:02 PM

Pipeline run activity ID

Throughput	31.457 KB/s
Total duration	00:00:10

✗ Duration breakdown

Start time	1/25/2023, 8:50:04 PM
Used parallel copies	1

Queue 

Transfer 
00:00:05
Period: Queue
Description: The elapsed time until the copy activity actually starts on the service.

✗ Advanced

Triggered by	ScheduleTrigger
--------------	-----------------

Close

7. If your data pipeline failed, view the error message by hovering over the activity row and select the message icon under Status. This selection will bring up error details, such as the error code and message.

Activity runs					View run detail	▲
Pipeline run ID					Export to CSV	▼
All status	Container	Run start	Duration	Status	Run ID	1
Showing 1 - 7 items						
Activity name		Run start	Duration	Status	Run ID	
ForEach_4i1		1/25/2023, 9:30:30 PM	00:00:17	✓ Succeeded	acd4d558-5014-4af0c-b97a-c71e399424cb	
Copy_4i1		1/25/2023, 9:30:31 PM	00:00:14	✓ Succeeded	e2ed3b19-b3fd-47fb-b3c1-6269a9b049a1	
Copy_4i1		1/25/2023, 9:30:31 PM	00:00:13	✓ Succeeded	58324b60-1604-43c2-ac12-e2156c0c11a3	
Copy_4i1		1/25/2023, 9:30:31 PM	00:00:13	✓ Succeeded	14a82738-2978-468a-b2b6-bf37b67f37e6	
Copy_4i1		1/25/2023, 9:30:31 PM	00:00:13	✓ Succeeded	ee8b520d-8cba-41b4-a88a-75a843a9e396	
Copy_4i1		1/25/2023, 9:30:31 PM	00:00:12	✓ Succeeded	db7141fa-71c3-47fd-b892-697a9932ca22	
Script1		1/25/2023, 9:30:46 PM	00:00:09	✗ Failed 	2b9d6ab9-c922-4598-bd68-02ed7c232dac	

The screenshot shows the 'Activity runs' section of the Azure Data Factory pipeline run history. A red box highlights the error details for the 'Script1' activity, which failed at 9:30:46 PM on 1/25/2023. The error message is as follows:

```
{
  "errorCode": "2011",
  "message": "Incorrect syntax near the keyword 'is'.",
  "failureType": "UserError",
  "target": "Script1",
  "details": []
}
```

Below the error details, the pipeline run status is shown as 'Failed'.

8. Select **Update pipeline** to make changes to your pipeline. This selection will land you back in the pipeline canvas.

The screenshot shows the pipeline canvas for the 'myDemoPipeline'. A red box highlights the 'Update pipeline' button. The 'Script1' activity is highlighted with a red circle, indicating it is the failed activity.

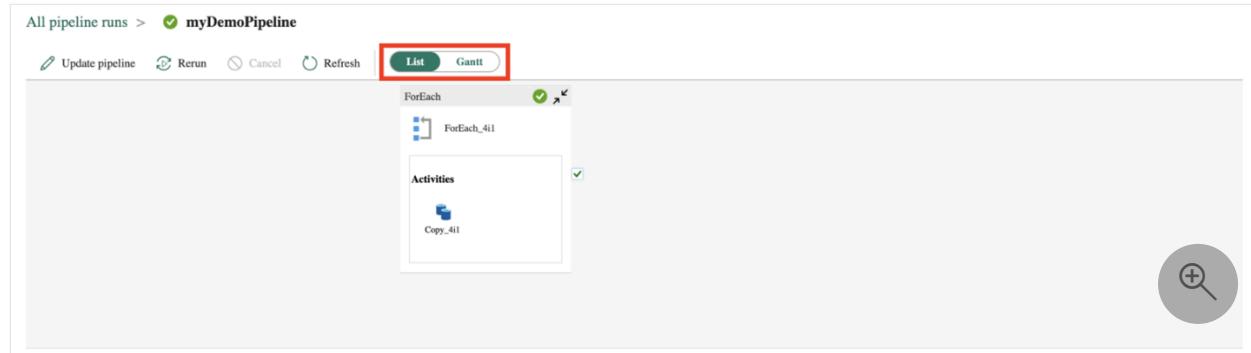
9. You can also **Rerun** your data pipeline. You can choose to rerun the entire pipeline or only rerun the pipeline from the failed activity.

The screenshot shows the 'All pipeline runs' page for 'myDemoPipeline'. A red box highlights the 'Rerun' dropdown menu. The 'Script1' activity is highlighted with a red circle.

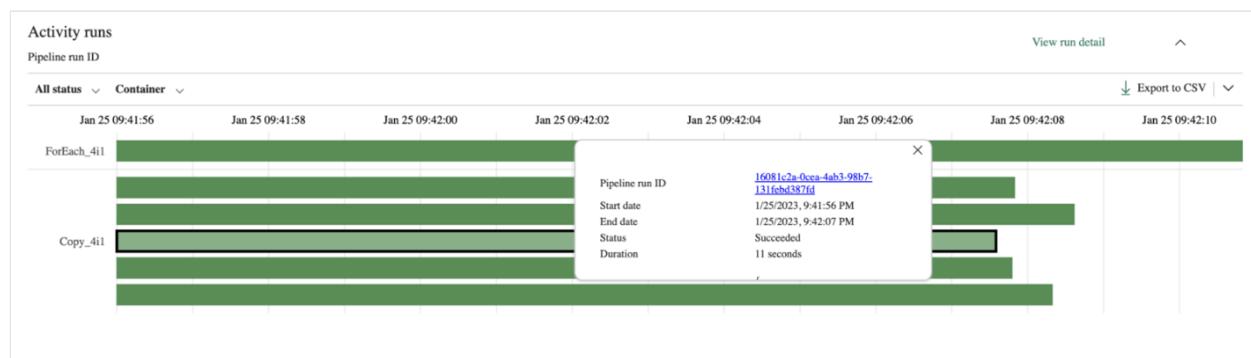
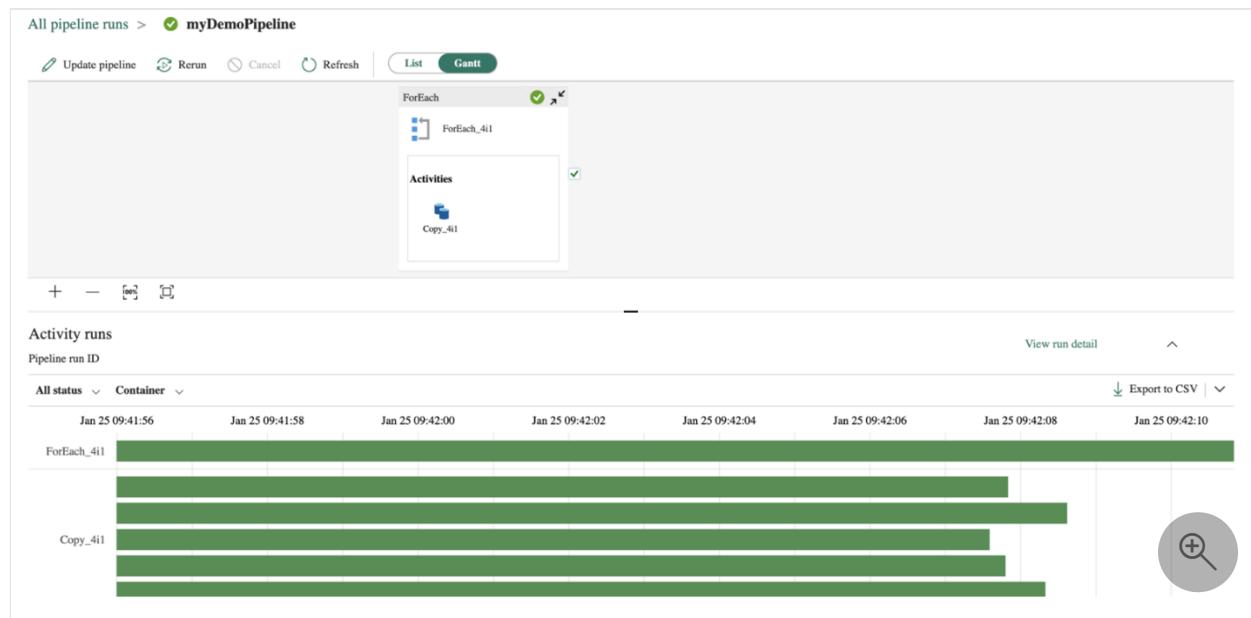
The screenshot shows the 'All pipeline runs' page for 'myDemoPipeline'. A red box highlights the 'Rerun' options: 'Rerun' and 'Rerun from failed activity'. The pipeline canvas shows the 'Script1' activity highlighted with a red circle.

Gantt view

A Gantt chart is a view that lets you see the run history over a time range. If you switch to a Gantt view, all pipeline runs will be grouped by name, displayed as bars relative to how long the run took.



The length of the bar relates to the duration of the pipeline. You can select the bar to view more details.



Next steps

- Quickstart: Create your first data pipeline to copy data

- Quickstart: Create your first Dataflow Gen2 to get and transform data

Parameters for Data Factory in Microsoft Fabric

Article • 05/23/2023

This document describes how to use parameters in your pipelines for Data Factory in Fabric.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

How to use parameters, expressions and functions in pipelines for Data Factory in Fabric

In this document, we focus on learning fundamental concepts with various examples to explore the ability to create parameterized data pipelines within Data Factory in Fabric. Parameterization and dynamic expressions can save a tremendous amount of time and allow for a much more flexible Extract, Transform, Load (ETL) or Extract, Load, Transform (ELT) solution, which will dramatically reduce the cost of solution maintenance and speed up the implementation of new features into existing pipelines. These gains are because parameterization minimizes the amount of hard coding and increases the number of reusable objects and processes in a solution.

Parameter and expression concepts

You can use parameters to pass external values into pipelines. Once the parameter has been passed into the resource, it cannot be changed. By parameterizing resources, you can reuse them with different values each time. Parameters can be used individually or as a part of expressions. Parameter values in the definition can be literal or expressions that are evaluated at runtime.

Expressions can appear anywhere in a string value and always generate another string value. Here, password is a pipeline parameter in the expression. If a parameter value is an expression, the body of the expression is extracted by removing the at-sign (@). If a

literal string is needed that starts with @, it must be escaped by using @@. The following examples show how expressions are evaluated.

Parameter value	Result
"parameters"	The characters 'parameters' are returned.
"parameters[1]"	The characters 'parameters[1]' are returned.
"@@"	A 1 character string that contains '@' is returned.
" @"	A 2 character string that contains ' @' is returned.

Expressions can also appear inside strings, using a feature called *string interpolation* where expressions are wrapped in `@{ ... }`. For example, the following string includes parameter values and literal string values:

```
"First Name: @{pipeline().parameters.firstName} Last Name:  
@{pipeline().parameters.lastName}"
```

Using string interpolation, the result is always a string. For example, if you defined `myNumber` as `42` and `myString` as `foo`:

Parameter value	Result
"@pipeline().parameters.myString"	Returns <code>foo</code> as a string.
"@{pipeline().parameters.myString}"	Returns <code>foo</code> as a string.
"@pipeline().parameters.myNumber"	Returns <code>42</code> as a number.
"@{pipeline().parameters.myNumber}"	Returns <code>42</code> as a string.
"Answer is: @{pipeline().parameters.myNumber}"	Returns the string <code>Answer is: 42.</code>
"@concat('Answer is: ', string(pipeline().parameters.myNumber))"	Returns the string <code>Answer is: 42</code>
"Answer is: @{@{pipeline().parameters.myNumber}"	Returns the string <code>Answer is: @{@{pipeline().parameters.myNumber}}.</code>

Examples of using parameters in expressions

Creating and using parameters

To create parameters, select the background of the pipeline editor canvas, and then the **Parameters** tab of the properties window at the bottom. Select the **+ New** button to add a new parameter to the pipeline, give it a name, a data type, and a default value:

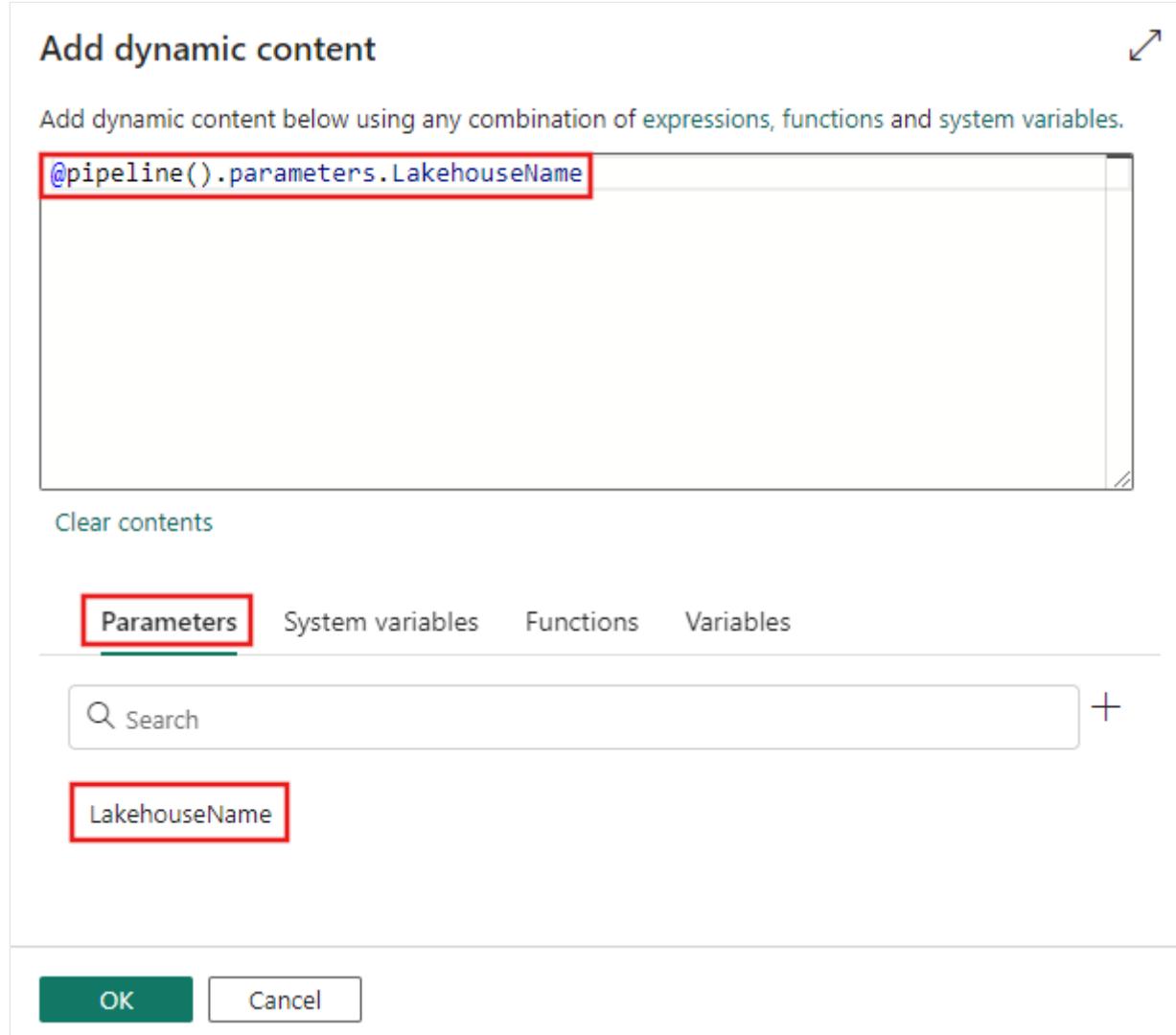
The screenshot shows the 'Parameters' tab of the properties window. At the top, there are buttons for '+ New' and 'Delete'. Below is a table with columns 'Name', 'Type', and 'Default value'. A single row is present with 'LakehouseName' as the name, 'String' as the type, and 'MyLakehouse' as the default value.

Name	Type	Default value
LakehouseName	String	MyLakehouse

You can then use the parameter anywhere in your pipeline where dynamic content is supported. In this example, the parameter is used to dynamically provide the name of a Lakehouse data store on the **Source** tab of a copy activity's property pages.

The screenshot shows the pipeline editor with a 'Copy data' activity named 'My Copy activity'. The 'Source' tab is selected. Under 'Data store type', 'Workspace' is selected. In the 'Lakehouse' dropdown, a 'Select...' button is highlighted with a red box. A context menu is open over this button, with 'Add dynamic content' highlighted with a red box. Other options in the menu include 'Select...', 'Add dynamic content', and 'SampleLakehouse'.

The Add dynamic content window is displayed, allowing you to specify any kind of dynamic content, including parameters, [system variables](#), [functions](#), or pipeline variables. In this example, the previously defined parameter is selected, and the dynamic content window is automatically populated with the correct expression to reference the parameter.



Complex expression example

The below example shows a complex example that references a deep sub-field of activity output. To reference a pipeline parameter that evaluates to a sub-field, use [] syntax instead of dot(.) operator (as in case of subfield1 and subfield2)

```
@activity('*activityName*').output.*subfield1*.*subfield2*  
[pipeline().parameters.*subfield3*].*subfield4*
```

Dynamic content editor

The dynamic content editor automatically escapes characters in your content when you finish editing. For example, the following content in the content editor is a string interpolation with an expression function:

```
@{toUpperCase('myData')}
```

The dynamic content editor converts the above content to the following expression:

```
MYDATA
```

Using functions and variables in expressions

You can call functions and use variables within expressions. The following sections provide information about the functions that can be used in an expression.

Pipeline scope variables

These system variables can be referenced anywhere in the pipeline JSON.

Variable Name	Description
@pipeline().DataFactory	Name of the data or Synapse workspace the pipeline run is running in
@pipeline().Pipeline	Name of the pipeline
@pipeline().RunId	ID of the specific pipeline run
@pipeline().TriggerId	ID of the trigger that invoked the pipeline
@pipeline().TriggerName	Name of the trigger that invoked the pipeline
@pipeline().TriggerTime	Time of the trigger run that invoked the pipeline. This is the time at which the trigger actually fired to invoke the pipeline run, and it may differ slightly from the trigger's scheduled time.
@pipeline().GroupId	ID of the group to which pipeline run belongs.
@pipeline()?.TriggeredByPipelineName	Name of the pipeline that triggers the pipeline run. Applicable when the pipeline run is triggered by an ExecutePipeline activity. Evaluate to <i>Null</i> when used in other circumstances. Note the question mark after @pipeline()

Variable Name	Description
@pipeline()?.TriggeredByPipelineRunId	Run ID of the pipeline that triggers the pipeline run. Applicable when the pipeline run is triggered by an ExecutePipeline activity. Evaluate to <i>Null</i> when used in other circumstances. Note the question mark after @pipeline()

ⓘ Note

Trigger-related date/time system variables (in both pipeline and trigger scopes) return UTC dates in ISO 8601 format, for example, 2017-06-01T22:20:00.4061448Z.

String functions

To work with strings, you can use these string functions and also some [collection functions](#). String functions work only on strings.

String function	Task
concat	Combine two or more strings, and return the combined string.
endsWith	Check whether a string ends with the specified substring.
guid	Generate a globally unique identifier (GUID) as a string.
indexOf	Return the starting position for a substring.
lastIndexOf	Return the starting position for the last occurrence of a substring.
replace	Replace a substring with the specified string, and return the updated string.
split	Return an array that contains substrings, separated by commas, from a larger string based on a specified delimiter character in the original string.
startsWith	Check whether a string starts with a specific substring.
substring	Return characters from a string, starting from the specified position.
toLower	Return a string in lowercase format.
toUpper	Return a string in uppercase format.
trim	Remove leading and trailing whitespace from a string, and return the updated string.

Collection functions

To work with collections, generally arrays, strings, and sometimes, dictionaries, you can use these collection functions.

Collection function	Task
contains	Check whether a collection has a specific item.
empty	Check whether a collection is empty.
first	Return the first item from a collection.
intersection	Return a collection that has <i>only</i> the common items across the specified collections.
join	Return a string that has <i>all</i> the items from an array, separated by the specified character.
last	Return the last item from a collection.
length	Return the number of items in a string or array.
skip	Remove items from the front of a collection, and return <i>all the other</i> items.
take	Return items from the front of a collection.
union	Return a collection that has <i>all</i> the items from the specified collections.

Logical functions

These functions are useful inside conditions, they can be used to evaluate any type of logic.

Logical comparison function	Task
and	Check whether all expressions are true.
equals	Check whether both values are equivalent.
greater	Check whether the first value is greater than the second value.
greaterOrEquals	Check whether the first value is greater than or equal to the second value.
if	Check whether an expression is true or false. Based on the result, return a specified value.

Logical comparison function	Task
less	Check whether the first value is less than the second value.
lessOrEquals	Check whether the first value is less than or equal to the second value.
not	Check whether an expression is false.
or	Check whether at least one expression is true.

Conversion functions

These functions are used to convert between each of the native types in the language:

- string
- integer
- float
- boolean
- arrays
- dictionaries

Conversion function	Task
array	Return an array from a single specified input. For multiple inputs, see createArray .
base64	Return the base64-encoded version for a string.
base64ToBinary	Return the binary version for a base64-encoded string.
base64ToString	Return the string version for a base64-encoded string.
binary	Return the binary version for an input value.
bool	Return the Boolean version for an input value.
coalesce	Return the first non-null value from one or more parameters.
createArray	Return an array from multiple inputs.
dataUri	Return the data URI for an input value.
dataUriToBinary	Return the binary version for a data URI.
dataUriToString	Return the string version for a data URI.
decodeBase64	Return the string version for a base64-encoded string.

Conversion function	Task
decodeDataUri	Return the binary version for a data URI.
decodeURIComponent	Return a string that replaces escape characters with decoded versions.
encodeURIComponent	Return a string that replaces URL-unsafe characters with escape characters.
float	Return a floating point number for an input value.
int	Return the integer version for a string.
json	Return the JavaScript Object Notation (JSON) type value or object for a string or XML.
string	Return the string version for an input value.
uriComponent	Return the URI-encoded version for an input value by replacing URL-unsafe characters with escape characters.
uriComponentToBinary	Return the binary version for a URI-encoded string.
uriComponentToString	Return the string version for a URI-encoded string.
xml	Return the XML version for a string.
xpath	Check XML for nodes or values that match an XPath (XML Path Language) expression, and return the matching nodes or values.

Math functions

These functions can be used for either types of numbers: **integers** and **floats**.

Math function	Task
add	Return the result from adding two numbers.
div	Return the result from dividing two numbers.
max	Return the highest value from a set of numbers or an array.
min	Return the lowest value from a set of numbers or an array.
mod	Return the remainder from dividing two numbers.
mul	Return the product from multiplying two numbers.
rand	Return a random integer from a specified range.
range	Return an integer array that starts from a specified integer.

Math function	Task
sub	Return the result from subtracting the second number from the first number.

Date functions

Date or time function	Task
addDays	Add a number of days to a timestamp.
addHours	Add a number of hours to a timestamp.
addMinutes	Add a number of minutes to a timestamp.
addSeconds	Add a number of seconds to a timestamp.
addToTime	Add a number of time units to a timestamp. See also getFutureTime .
convertFromUtc	Convert a timestamp from Universal Time Coordinated (UTC) to the target time zone.
convertTimeZone	Convert a timestamp from the source time zone to the target time zone.
convertToUtc	Convert a timestamp from the source time zone to Universal Time Coordinated (UTC).
dayOfMonth	Return the day of the month component from a timestamp.
dayOfWeek	Return the day of the week component from a timestamp.
dayOfYear	Return the day of the year component from a timestamp.
formatDateTime	Return the timestamp as a string in optional format.
getFutureTime	Return the current timestamp plus the specified time units. See also addToTime .
getPastTime	Return the current timestamp minus the specified time units. See also subtractFromTime .
startOfDay	Return the start of the day for a timestamp.
startOfHour	Return the start of the hour for a timestamp.
startOfMonth	Return the start of the month for a timestamp.
subtractFromTime	Subtract a number of time units from a timestamp. See also getPastTime .
ticks	Return the <code>ticks</code> property value for a specified timestamp.

Date or time function	Task
utcNow	Return the current timestamp as a string.

Next steps

[Expression language](#)

Expressions and functions for Data Factory in Microsoft Fabric

Article • 05/23/2023

This article provides details about expressions and functions supported by Data Factory in Microsoft Fabric.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Expressions

Expression values in the definition can be literal or expressions that are evaluated at runtime. For example:

`"value"`

or

`"@pipeline().parameters.password"`

Expressions can appear anywhere in a string value and always result in another string value. If a value is an expression, the body of the expression is extracted by removing the at-sign (@). If a literal string is needed that starts with @, it must be escaped by using @@. The following examples show how expressions are evaluated.

Expression value	Result
<code>"parameters"</code>	The characters 'parameters' are returned.
<code>"parameters[1]"</code>	The characters 'parameters[1]' are returned.
<code>"@@"</code>	A 1 character string that contains '@' is returned.
<code>" @"</code>	A 2 character string that contains ' @' is returned.

Expressions can also appear inside strings, using a feature called *string interpolation* where expressions are wrapped in `@{ ... }`. For example: "First Name:

```
@{pipeline().parameters.firstName} Last Name: @{pipeline().parameters.lastName}"
```

Using string interpolation, the result is always a string. Say I have defined `myNumber` as `42` and `myString` as `foo`:

Expression value	Result
"@pipeline().parameters.myString"	Returns <code>foo</code> as a string.
"@{pipeline().parameters.myString}"	Returns <code>foo</code> as a string.
"@pipeline().parameters.myNumber"	Returns <code>42</code> as a <i>number</i> .
"@{pipeline().parameters.myNumber}"	Returns <code>42</code> as a <i>string</i> .
"Answer is: @{pipeline().parameters.myNumber}"	Returns the string <code>Answer is: 42</code> .
"@concat('Answer is: ', string(pipeline().parameters.myNumber))"	Returns the string <code>Answer is: 42</code>
"Answer is: @@{pipeline().parameters.myNumber}"	Returns the string <code>Answer is: @{pipeline().parameters.myNumber}</code> .

In the control flow activities like ForEach activity, you can provide an array to be iterated over for the property items and use `@item()` to iterate over a single enumeration in ForEach activity. For example, if items is an array: [1, 2, 3], `@item()` returns 1 in the first iteration, 2 in the second iteration, and 3 in the third iteration. You can also use `@range(0,10)` like expression to iterate 10 times starting with 0 ending with 9.

You can use `@activity('activity name')` to capture output of activity and make decisions. Consider a web activity called Web1. For placing the output of the first activity in the body of the second, the expression generally looks like: `@activity('Web1').output` or `@activity('Web1').output.data` or something similar depending upon what the output of the first activity looks like.

Examples

Complex expression example

The below example shows a complex example that references a deep sub-field of activity output. To reference a pipeline parameter that evaluates to a sub-field, use `[]`

syntax instead of dot(.) operator (as in case of subfield1 and subfield2), as part of an activity output.

```
@activity('*activityName*').output.*subfield1*.*subfield2*
[pipeline().parameters.*subfield3*].*subfield4*
```

Creating files dynamically and naming them is common pattern. Let us explore few dynamic file naming examples.

- Append Date to a filename: `@concat('Test_', formatDateTime.UtcNow(), 'yyyy-dd-MM'))`
- Append DateTime in customer timezone: `@concat('Test_', convertFromUtc.UtcNow(), 'Pacific Standard Time'))`
- Append Trigger Time: `@concat('Test_', pipeline().TriggerTime)`
- Output a custom filename in a Mapping Data Flow when outputting to a single file with date: `'Test_' + toString(currentDate()) + '.csv'`

In above cases, four dynamic filenames are created starting with Test_.

Dynamic content editor

The dynamic content editor automatically escapes characters in your content when you finish editing. For example, the following content in content editor is a string interpolation with an expression function.

```
"@{toUpperCase('myData')}"
```

The dynamic content editor converts above content to expression `"@{toUpperCase('myData')}"`. The result of this expression is the formatted string showed below.

```
"MYDATA"
```

Replacing special characters

The dynamic content editor automatically escapes characters like double quote, backslash in your content when you finish editing. This causes trouble if you want to replace line feed or tab by using \n, \t in replace() function. You can edit your dynamic content in code view to remove the extra \ in the expression, or you can follow below steps to replace special characters using expression language:

1. URL encoding against the original string value

2. Replace URL encoded string, for example, line feed (%0A), carriage return(%0D), horizontal tab(%09).
3. URL decoding

For example, variable `companyName` with a newline character in its value, expression `@uriComponentToString(replace(uriComponent(variables('companyName')), '%0A', ''))` can remove the newline character.

Contoso- Corporation

Escaping single quote character

Expression functions use single quote for string value parameters. Use two single quotes to escape a ' character in string functions. For example, expression `@concat('Baba', '''s ', 'book store')` will return below result.

Baba's book store

Pipeline scope variables

These system variables can be referenced anywhere in the pipeline.

Variable Name	Description
<code>@pipeline().DataFactory</code>	Name of the data or Synapse workspace the pipeline run is running in
<code>@pipeline().Pipeline</code>	Name of the pipeline
<code>@pipeline().RunId</code>	ID of the specific pipeline run
<code>@pipeline().TriggerId</code>	ID of the trigger that invoked the pipeline
<code>@pipeline().TriggerName</code>	Name of the trigger that invoked the pipeline
<code>@pipeline().TriggerTime</code>	Time of the trigger run that invoked the pipeline. This is the time at which the trigger actually fired to invoke the pipeline run, and it may differ slightly from the trigger's scheduled time.
<code>@pipeline().GroupId</code>	ID of the group to which pipeline run belongs.

Variable Name	Description
@pipeline()?.TriggeredByPipelineName	Name of the pipeline that triggers the pipeline run. Applicable when the pipeline run is triggered by an ExecutePipeline activity. Evaluate to <i>Null</i> when used in other circumstances. Note the question mark after @pipeline()
@pipeline()?.TriggeredByPipelineRunId	Run ID of the pipeline that triggers the pipeline run. Applicable when the pipeline run is triggered by an ExecutePipeline activity. Evaluate to <i>Null</i> when used in other circumstances. Note the question mark after @pipeline()

① Note

Trigger-related date/time system variables (in both pipeline and trigger scopes) return UTC dates in ISO 8601 format, for example, `2017-06-01T22:20:00.4061448Z`.

Functions

You can call functions within expressions. The following sections provide information about the functions that can be used in an expression.

Date functions

Date or time function	Task
addDays	Add a number of days to a timestamp.
addHours	Add a number of hours to a timestamp.
addMinutes	Add a number of minutes to a timestamp.
addSeconds	Add a number of seconds to a timestamp.
addToTime	Add a number of time units to a timestamp. See also getFutureTime .
convertFromUtc	Convert a timestamp from Universal Time Coordinated (UTC) to the target time zone.
convertTimeZone	Convert a timestamp from the source time zone to the target time zone.
convertToUtc	Convert a timestamp from the source time zone to Universal Time Coordinated (UTC).

Date or time function	Task
dayOfMonth	Return the day of the month component from a timestamp.
dayOfWeek	Return the day of the week component from a timestamp.
dayOfYear	Return the day of the year component from a timestamp.
formatDateTime	Return the timestamp as a string in optional format.
getFutureTime	Return the current timestamp plus the specified time units. See also addToTime .
getPastTime	Return the current timestamp minus the specified time units. See also subtractFromTime .
startOfDay	Return the start of the day for a timestamp.
startOfHour	Return the start of the hour for a timestamp.
startOfMonth	Return the start of the month for a timestamp.
subtractFromTime	Subtract a number of time units from a timestamp. See also getPastTime .
ticks	Return the <code>ticks</code> property value for a specified timestamp.
utcNow	Return the current timestamp as a string.

String functions

To work with strings, you can use these string functions and also some [collection functions](#). String functions work only on strings.

String function	Task
concat	Combine two or more strings, and return the combined string.
endsWith	Check whether a string ends with the specified substring.
guid	Generate a globally unique identifier (GUID) as a string.
indexOf	Return the starting position for a substring.
lastIndexOf	Return the starting position for the last occurrence of a substring.
replace	Replace a substring with the specified string, and return the updated string.

String function	Task
split	Return an array that contains substrings, separated by commas, from a larger string based on a specified delimiter character in the original string.
startsWith	Check whether a string starts with a specific substring.
substring	Return characters from a string, starting from the specified position.
toLowerCase	Return a string in lowercase format.
toUpperCase	Return a string in uppercase format.
trim	Remove leading and trailing whitespace from a string, and return the updated string.

Collection functions

To work with collections, generally arrays, strings, and sometimes, dictionaries, you can use these collection functions.

Collection function	Task
contains	Check whether a collection has a specific item.
empty	Check whether a collection is empty.
first	Return the first item from a collection.
intersection	Return a collection that has <i>only</i> the common items across the specified collections.
join	Return a string that has <i>all</i> the items from an array, separated by the specified character.
last	Return the last item from a collection.
length	Return the number of items in a string or array.
skip	Remove items from the front of a collection, and return <i>all the other</i> items.
take	Return items from the front of a collection.
union	Return a collection that has <i>all</i> the items from the specified collections.

Logical functions

These functions are useful inside conditions, they can be used to evaluate any type of logic.

Logical comparison function	Task
and	Check whether all expressions are true.
equals	Check whether both values are equivalent.
greater	Check whether the first value is greater than the second value.
greaterOrEquals	Check whether the first value is greater than or equal to the second value.
if	Check whether an expression is true or false. Based on the result, return a specified value.
less	Check whether the first value is less than the second value.
lessOrEquals	Check whether the first value is less than or equal to the second value.
not	Check whether an expression is false.
or	Check whether at least one expression is true.

Conversion functions

These functions are used to convert between each of the native types in the language:

- string
- integer
- float
- boolean
- arrays
- dictionaries

Conversion function	Task
array	Return an array from a single specified input. For multiple inputs, see createArray .
base64	Return the base64-encoded version for a string.
base64ToBinary	Return the binary version for a base64-encoded string.
base64ToString	Return the string version for a base64-encoded string.

Conversion function	Task
binary	Return the binary version for an input value.
bool	Return the Boolean version for an input value.
coalesce	Return the first non-null value from one or more parameters.
createArray	Return an array from multiple inputs.
dataUri	Return the data URI for an input value.
dataUriToBinary	Return the binary version for a data URI.
dataUriToString	Return the string version for a data URI.
decodeBase64	Return the string version for a base64-encoded string.
decodeDataUri	Return the binary version for a data URI.
decodeUriComponent	Return a string that replaces escape characters with decoded versions.
encodeUriComponent	Return a string that replaces URL-unsafe characters with escape characters.
float	Return a floating point number for an input value.
int	Return the integer version for a string.
string	Return the string version for an input value.
uriComponent	Return the URI-encoded version for an input value by replacing URL-unsafe characters with escape characters.
uriComponentToBinary	Return the binary version for a URI-encoded string.
uriComponentToString	Return the string version for a URI-encoded string.
xml	Return the XML version for a string.
xpath	Check XML for nodes or values that match an XPath (XML Path Language) expression, and return the matching nodes or values.

Math functions

These functions can be used for either types of numbers: **integers** and **floats**.

Math function	Task
add	Return the result from adding two numbers.

Math function	Task
div	Return the result from dividing two numbers.
max	Return the highest value from a set of numbers or an array.
min	Return the lowest value from a set of numbers or an array.
mod	Return the remainder from dividing two numbers.
mul	Return the product from multiplying two numbers.
rand	Return a random integer from a specified range.
range	Return an integer array that starts from a specified integer.
sub	Return the result from subtracting the second number from the first number.

Function reference

This section lists all the available functions in alphabetical order.

add

Return the result from adding two numbers.

```
add(<summand_1>, <summand_2>)
```

Parameter	Required	Type	Description
<summand_1>, <summand_2>	Yes	Integer, Float, or mixed	The numbers to add

Return value	Type	Description
<result-sum>	Integer or Float	The result from adding the specified numbers

Example

This example adds the specified numbers:

```
add(1, 1.5)
```

And returns this result: 2.5

addDays

Add a number of days to a timestamp.

```
addDays('<timestamp>', <days>, '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<days>	Yes	Integer	The positive or negative number of days to add
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is "o" (yyyy-MM-ddTHH:mm:ss.ffffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<updated-timestamp>	String	The timestamp plus the specified number of days

Example 1

This example adds 10 days to the specified timestamp:

```
addDays('2018-03-15T13:00:00Z', 10)
```

And returns this result: "2018-03-25T00:00:000000Z"

Example 2

This example subtracts five days from the specified timestamp:

```
addDays('2018-03-15T00:00:00Z', -5)
```

And returns this result: "2018-03-10T00:00:000000Z"

addHours

Add a number of hours to a timestamp.

```
addHours('<timestamp>', <hours>, '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<hours>	Yes	Integer	The positive or negative number of hours to add
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is "o" (yyyy-MM-ddTHH:mm:ss.fffffffK), which complies with ISO 8601 ↗ and preserves time zone information.

Return value	Type	Description
<updated-timestamp>	String	The timestamp plus the specified number of hours

Example 1

This example adds 10 hours to the specified timestamp:

```
addHours('2018-03-15T00:00:00Z', 10)
```

And returns this result: "2018-03-15T10:00:000000Z"

Example 2

This example subtracts five hours from the specified timestamp:

```
addHours('2018-03-15T15:00:00Z', -5)
```

And returns this result: "2018-03-15T10:00:000000Z"

addMinutes

Add a number of minutes to a timestamp.

```
addMinutes('<timestamp>', <minutes>, '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<minutes>	Yes	Integer	The positive or negative number of minutes to add
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is "o" (yyyy-MM-ddTHH:mm:ss.fffffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<updated-timestamp>	String	The timestamp plus the specified number of minutes

Example 1

This example adds 10 minutes to the specified timestamp:

```
addMinutes('2018-03-15T00:10:00Z', 10)
```

And returns this result: "2018-03-15T00:20:00.0000000Z"

Example 2

This example subtracts five minutes from the specified timestamp:

```
addMinutes('2018-03-15T00:20:00Z', -5)
```

And returns this result: "2018-03-15T00:15:00.0000000Z"

addSeconds

Add a number of seconds to a timestamp.

```
addSeconds('<timestamp>', <seconds>, '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<seconds>	Yes	Integer	The positive or negative number of seconds to add
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is "o" (yyyy-MM-ddTHH:mm:ss.ffffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<updated-timestamp>	String	The timestamp plus the specified number of seconds

Example 1

This example adds 10 seconds to the specified timestamp:

```
addSeconds('2018-03-15T00:00:00Z', 10)
```

And returns this result: "2018-03-15T00:00:10.0000000Z"

Example 2

This example subtracts five seconds to the specified timestamp:

```
addSeconds('2018-03-15T00:00:30Z', -5)
```

And returns this result: "2018-03-15T00:00:25.0000000Z"

addToTime

Add a number of time units to a timestamp. See also [getFutureTime\(\)](#).

```
addToTime('<timestamp>', <interval>, '<timeUnit>', '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<interval>	Yes	Integer	The number of specified time units to add
<timeUnit>	Yes	String	The unit of time to use with <i>interval</i> : "Second", "Minute", "Hour", "Day", "Week", "Month", "Year"
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is " <code>o</code> " (yyyy-MM-ddTHH:mm:ss.fffffffK), which complies with ISO 8601 ↗ and preserves time zone information.

Return value	Type	Description
<updated-timestamp>	String	The timestamp plus the specified number of time units

Example 1

This example adds one day to the specified timestamp:

```
addTime('2018-01-01T00:00:00Z', 1, 'Day')
```

And returns this result: "2018-01-02T00:00:00.0000000Z"

Example 2

This example adds one day to the specified timestamp:

```
addTime('2018-01-01T00:00:00Z', 1, 'Day', 'D')
```

And returns the result using the optional "D" format: "Tuesday, January 2, 2018"

and

Check whether both expressions are true. Return true when both expressions are true, or return false when at least one expression is false.

```
and(<expression1>, <expression2>)
```

Parameter	Required	Type	Description
<expression1>, <expression2>	Yes	Boolean	The expressions to check
Return value	Type	Description	
true or false	Boolean	Return true when both expressions are true. Return false when at least one expression is false.	

Example 1

These examples check whether the specified Boolean values are both true:

```
and(true, true)
and(false, true)
and(false, false)
```

And returns these results:

- First example: Both expressions are true, so returns `true`.
- Second example: One expression is false, so returns `false`.
- Third example: Both expressions are false, so returns `false`.

Example 2

These examples check whether the specified expressions are both true:

```
and>equals(1, 1), equals(2, 2))
and>equals(1, 1), equals(1, 2))
and>equals(1, 2), equals(1, 3))
```

And returns these results:

- First example: Both expressions are true, so returns `true`.
- Second example: One expression is false, so returns `false`.
- Third example: Both expressions are false, so returns `false`.

array

Return an array from a single specified input. For multiple inputs, see [createArray\(\)](#).

```
array('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The string for creating an array

Return value	Type	Description
[<value>]	Array	An array that contains the single specified input

Example

This example creates an array from the "hello" string:

```
array('hello')
```

And returns this result: ["hello"]

base64

Return the base64-encoded version for a string.

```
base64('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The input string

Return value	Type	Description
<base64-string>	String	The base64-encoded version for the input string

Example

This example converts the "hello" string to a base64-encoded string:

```
base64('hello')
```

And returns this result: "aGVsbG8="

base64ToBinary

Return the binary version for a base64-encoded string.

```
base64ToBinary('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The base64-encoded string to convert

Return value	Type	Description
<binary-for-base64-string>	String	The binary version for the base64-encoded string

Example

This example converts the "aGVsbG8=" base64-encoded string to a binary string:

```
base64ToBinary('aGVsbG8=')
```

And returns this result:

```
"0110000101000111010101100111001101100010010001110011100000111101"
```

base64ToString

Return the string version for a base64-encoded string, effectively decoding the base64 string. Use this function rather than [decodeBase64\(\)](#). Although both functions work the same way, `base64ToString()` is preferred.

```
base64ToString('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The base64-encoded string to decode

Return value	Type	Description
<decoded-base64-string>	String	The string version for a base64-encoded string

Example

This example converts the "aGVsbG8=" base64-encoded string to just a string:

```
base64ToString('aGVsbG8=')
```

And returns this result: "hello"

binary

Return the binary version for a string.

```
binary('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The string to convert

Return value	Type	Description
<binary-for-input-value>	String	The binary version for the specified string

Example

This example converts the "hello" string to a binary string:

```
binary('hello')
```

And returns this result:

```
"011010001100101011011000110110001101111"
```

bool

Return the Boolean version for a value.

```
bool(<value>)
```

Parameter	Required	Type	Description
<value>	Yes	Any	The value to convert

Return value	Type	Description
true or false	Boolean	The Boolean version for the specified value

Example

These examples convert the specified values to Boolean values:

```
bool(1)  
bool(0)
```

And returns these results:

- First example: `true`
- Second example: `false`

coalesce

Return the first non-null value from one or more parameters. Empty strings, empty arrays, and empty objects aren't null.

```
coalesce(<object_1>, <object_2>, ...)
```

Parameter	Required	Type	Description
<code><object_1>, <object_2>, ...</code>	Yes	Any, can mix types	One or more items to check for null

Return value	Type	Description
<code><first-non-null-item></code>	Any	The first item or value that isn't null. If all parameters are null, this function returns null.

Example

These examples return the first non-null value from the specified values, or null when all the values are null:

```
coalesce(null, true, false)
coalesce(null, 'hello', 'world')
coalesce(null, null, null)
```

And returns these results:

- First example: `true`
- Second example: `"hello"`
- Third example: `null`

concat

Combine two or more strings, and return the combined string.

```
concat('<text1>', '<text2>', ...)
```

Parameter	Required	Type	Description
<code><text1>, <text2>, ...</code>	Yes	String	At least two strings to combine

Return value	Type	Description
<code><text1text2...></code>	String	The string created from the combined input strings

Example

This example combines the strings "Hello" and "World":

```
concat('Hello', 'World')
```

And returns this result: "HelloWorld"

contains

Check whether a collection has a specific item. Return true when the item is found, or return false when not found. This function is case-sensitive.

```
contains('<collection>', '<value>')
contains([<collection>], '<value>')
```

Specifically, this function works on these collection types:

- A *string* to find a *substring*
- An *array* to find a *value*
- A *dictionary* to find a *key*

Parameter	Required	Type	Description
<collection>	Yes	String, Array, or Dictionary	The collection to check
<value>	Yes	String, Array, or Dictionary, respectively	The item to find

Return value	Type	Description
true or false	Boolean	Return true when the item is found. Return false when not found.

Example 1

This example checks the string "hello world" for the substring "world" and returns true:

```
contains('hello world', 'world')
```

Example 2

This example checks the string "hello world" for the substring "universe" and returns false:

```
contains('hello world', 'universe')
```

convertFromUtc

Convert a timestamp from Universal Time Coordinated (UTC) to the target time zone.

```
convertFromUtc('<timestamp>', '<destinationTimeZone>', '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<destinationTimeZone>	Yes	String	The name for the target time zone. For time zone names, see Microsoft Time Zone Values , but you might have to remove any punctuation from the time zone name.
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is "o" (yyyy-MM-ddTHH:mm:ss.ffffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<converted-timestamp>	String	The timestamp converted to the target time zone

Example 1

This example converts a timestamp to the specified time zone:

```
convertFromUtc('2018-01-01T08:00:00.000000Z', 'Pacific Standard Time')
```

And returns this result: "2018-01-01T00:00:00Z"

Example 2

This example converts a timestamp to the specified time zone and format:

```
convertFromUtc('2018-01-01T08:00:00.000000Z', 'Pacific Standard Time', 'D')
```

And returns this result: "Monday, January 1, 2018"

convertTimeZone

Convert a timestamp from the source time zone to the target time zone.

```
convertTimeZone('<timestamp>', '<sourceTimeZone>', '<destinationTimeZone>', '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<sourceTimeZone>	Yes	String	The name for the source time zone. For time zone names, see Microsoft Time Zone Values , but you might have to remove any punctuation from the time zone name.
<destinationTimeZone>	Yes	String	The name for the target time zone. For time zone names, see Microsoft Time Zone Values , but you might have to remove any punctuation from the time zone name.
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is " <code>o</code> " (yyyy-MM-ddTHH:mm:ss.fffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<converted-timestamp>	String	The timestamp converted to the target time zone

Example 1

This example converts the source time zone to the target time zone:

```
convertTimeZone('2018-01-01T08:00:00.000000Z', 'UTC', 'Pacific Standard Time')
```

And returns this result: "2018-01-01T00:00:00.000000"

Example 2

This example converts a time zone to the specified time zone and format:

```
convertTimeZone('2018-01-01T08:00:00.000000Z', 'UTC', 'Pacific Standard Time', 'D')
```

And returns this result: "Monday, January 1, 2018"

convertToUtc

Convert a timestamp from the source time zone to Universal Time Coordinated (UTC).

```
convertToUtc('<timestamp>', '<sourceTimeZone>', '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<sourceTimeZone>	Yes	String	The name for the source time zone. For time zone names, see Microsoft Time Zone Values , but you might have to remove any punctuation from the time zone name.
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is "o" (yyyy-MM-ddTHH:mm:ss.fffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<converted-timestamp>	String	The timestamp converted to UTC

Example 1

This example converts a timestamp to UTC:

```
convertToUtc('01/01/2018 00:00:00', 'Pacific Standard Time')
```

And returns this result: "2018-01-01T08:00:00.000000Z"

Example 2

This example converts a timestamp to UTC:

```
convertToUtc('01/01/2018 00:00:00', 'Pacific Standard Time', 'D')
```

And returns this result: "Monday, January 1, 2018"

createArray

Return an array from multiple inputs. For single input arrays, see [array\(\)](#).

```
createArray('<object1>', '<object2>', ...)
```

Parameter	Required	Type	Description
<object1>, <object2>, ...	Yes	Any, but not mixed	At least two items to create the array

Return value	Type	Description
[<object1>, <object2>, ...]	Array	The array created from all the input items

Example

This example creates an array from these inputs:

```
createArray('h', 'e', 'l', 'l', 'o')
```

And returns this result: ["h", "e", "l", "l", "o"]

dataUri

Return a data uniform resource identifier (URI) for a string.

```
dataUri('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The string to convert

Return value	Type	Description
<data-uri>	String	The data URI for the input string

Example

This example creates a data URI for the "hello" string:

```
dataUri('hello')
```

And returns this result: "data:text/plain;charset=utf-8;base64,aGVsbG8="

dataUriToBinary

Return the binary version for a data uniform resource identifier (URI). Use this function rather than [decodeDataUri\(\)](#). Although both functions work the same way, `dataUriBinary()` is preferred.

```
dataUriToBinary('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The data URI to convert

Return value	Type	Description
<binary-for-data-uri>	String	The binary version for the data URI

Example

This example creates a binary version for this data URI:

```
dataUriToBinary('data:text/plain;charset=utf-8;base64,aGVsbG8=')
```

And returns this result:

```
"011001000110000101110100011000010011101001110100011001010111000011101000010111101  
1100000  
110110001100001011010010110110001110110100011011010000101110010011100110110100010111101  
01010111  
010000111010111010101110100011001100010110100111000011101101100010011000010111001  
10110010  
1001101100011010000101100011000010100011101010110011100110110001001000111001110000  
111101"
```

dataUriToString

Return the string version for a data uniform resource identifier (URI).

```
dataUriToString('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The data URI to convert

Return value	Type	Description
<string-for-data-uri>	String	The string version for the data URI

Example

This example creates a string for this data URI:

```
dataUriToString('data:text/plain;charset=utf-8;base64,aGVsbG8=')
```

And returns this result: "hello"

dayOfMonth

Return the day of the month from a timestamp.

```
dayOfMonth('<timestamp>')
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp

Return value	Type	Description
<day-of-month>	Integer	The day of the month from the specified timestamp

Example

This example returns the number for the day of the month from this timestamp:

```
dayOfMonth('2018-03-15T13:27:36Z')
```

And returns this result: 15

dayOfWeek

Return the day of the week from a timestamp.

```
dayOfWeek('<timestamp>')
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp

Return value	Type	Description
<day-of-week>	Integer	The day of the week from the specified timestamp where Sunday is 0, Monday is 1, and so on

Example

This example returns the number for the day of the week from this timestamp:

```
dayOfWeek('2018-03-15T13:27:36Z')
```

And returns this result: 3

dayOfYear

Return the day of the year from a timestamp.

```
dayOfYear('<timestamp>')
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp

Return value	Type	Description
<day-of-year>	Integer	The day of the year from the specified timestamp

Example

This example returns the number of the day of the year from this timestamp:

```
dayOfYear('2018-03-15T13:27:36Z')
```

And returns this result: 74

decodeBase64

Return the string version for a base64-encoded string, effectively decoding the base64 string. Consider using [base64ToString\(\)](#) rather than `decodeBase64()`. Although both functions work the same way, `base64ToString()` is preferred.

```
decodeBase64('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The base64-encoded string to decode

Return value	Type	Description
<decoded-base64-string>	String	The string version for a base64-encoded string

Example

This example creates a string for a base64-encoded string:

```
decodeBase64('aGVsbG8=')
```

And returns this result: "hello"

decodeDataUri

Return the binary version for a data uniform resource identifier (URI). Consider using [dataUriToBinary\(\)](#), rather than `decodeDataUri()`. Although both functions work the same way, `dataUriToBinary()` is preferred.

```
decodeDataUri('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The data URI string to decode

Return value	Type	Description
<binary-for-data-uri>	String	The binary version for a data URI string

Example

This example returns the binary version for this data URI:

```
decodeDataUri('data:text/plain;charset=utf-8;base64,aGVsbG8=')
```

And returns this result:

```
"011001000110000101110100011000010011101001110100011001010111000011101000010111101  
1100000  
11011000110000101101001011011000111011000110110100010111001001011100100111001101101  
01010111  
01000011101011101010111010001100110001011010011100001110110110001001100010111001  
10110010  
100110110001101000010110001100001010001110101011001110001001000111001110000  
111101"
```

decodeUriComponent

Return a string that replaces escape characters with decoded versions.

```
decodeUriComponent('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The string with the escape characters to decode

Return value	Type	Description
<decoded-uri>	String	The updated string with the decoded escape characters

Example

This example replaces the escape characters in this string with decoded versions:

```
decodeUriComponent('http%3A%2F%2Fcontoso.com')
```

And returns this result: "https://contoso.com"

div

Return the integer result from dividing two numbers. To get the remainder result, see [mod\(\)](#).

```
div(<dividend>, <divisor>)
```

Parameter	Required	Type	Description
<dividend>	Yes	Integer or Float	The number to divide by the <i>divisor</i>
<divisor>	Yes	Integer or Float	The number that divides the <i>dividend</i> , but can't be 0

Return value	Type	Description
<quotient-result>	Integer	The integer result from dividing the first number by the second number

Example

Both examples divide the first number by the second number:

```
div(10, 5)
div(11, 5)
```

And return this result: 2

encodeUriComponent

Return a uniform resource identifier (URI) encoded version for a string by replacing URL-unsafe characters with escape characters. Consider using [uriComponent\(\)](#), rather than [encodeUriComponent\(\)](#). Although both functions work the same way, [uriComponent\(\)](#) is preferred.

```
encodeUriComponent('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The string to convert to URI-encoded format

Return value	Type	Description
<encoded-uri>	String	The URI-encoded string with escape characters

Example

This example creates a URI-encoded version for this string:

```
encodeUriComponent('https://contoso.com')
```

And returns this result: "http%3A%2F%2Fcontoso.com"

empty

Check whether a collection is empty. Return true when the collection is empty, or return false when not empty.

```
empty('<collection>')
empty([<collection>])
```

Parameter	Required	Type	Description
<collection>	Yes	String, Array, or Object	The collection to check

Return value	Type	Description
true or false	Boolean	Return true when the collection is empty. Return false when not empty.

Example

These examples check whether the specified collections are empty:

```
empty('')
empty('abc')
```

And returns these results:

- First example: Passes an empty string, so the function returns true.

- Second example: Passes the string "abc", so the function returns `false`.

endsWith

Check whether a string ends with a specific substring. Return true when the substring is found, or return false when not found. This function isn't case-sensitive.

```
endsWith('<text>', '<searchText>')
```

Parameter	Required	Type	Description
<code><text></code>	Yes	String	The string to check
<code><searchText></code>	Yes	String	The ending substring to find

Return value	Type	Description
true or false	Boolean	Return true when the ending substring is found. Return false when not found.

Example 1

This example checks whether the "hello world" string ends with the "world" string:

```
endsWith('hello world', 'world')
```

And returns this result: `true`

Example 2

This example checks whether the "hello world" string ends with the "universe" string:

```
endsWith('hello world', 'universe')
```

And returns this result: `false`

equals

Check whether both values, expressions, or objects are equivalent. Return true when both are equivalent, or return false when they're not equivalent.

```
equals('<object1>', '<object2>')
```

Parameter	Required	Type	Description
<object1>, <object2>	Yes	Various	The values, expressions, or objects to compare

Return value	Type	Description
true or false	Boolean	Return true when both are equivalent. Return false when not equivalent.

Example

These examples check whether the specified inputs are equivalent.

```
equals(true, 1)
equals('abc', 'abcd')
```

And returns these results:

- First example: Both values are equivalent, so the function returns `true`.
- Second example: Both values aren't equivalent, so the function returns `false`.

first

Return the first item from a string or array.

```
first('<collection>')
first([<collection>])
```

Parameter	Required	Type	Description
<collection>	Yes	String or Array	The collection where to find the first item

Return value	Type	Description
<first-collection-item>	Any	The first item in the collection

Example

These examples find the first item in these collections:

```
first('hello')
first(createArray(0, 1, 2))
```

And return these results:

- First example: "h"
- Second example: 0

float

Convert a string version for a floating-point number to an actual floating point number.

```
float('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The string that has a valid floating-point number to convert

Return value	Type	Description
<float-value>	Float	The floating-point number for the specified string

Example

This example creates a string version for this floating-point number:

```
float('10.333')
```

And returns this result: 10.333

formatDateTime

Return a timestamp in the specified format.

```
formatDateTime('<timestamp>', '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is " <code>o</code> " (yyyy-MM-ddTHH:mm:ss.fffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<reformatted-timestamp>	String	The updated timestamp in the specified format

Example

This example converts a timestamp to the specified format:

```
formatDateTime('03/15/2018 12:00:00', 'yyyy-MM-ddTHH:mm:ss')
```

And returns this result: "2018-03-15T12:00:00"

getFutureTime

Return the current timestamp plus the specified time units.

```
getFutureTime(<interval>, <timeUnit>, <format>?)
```

Parameter	Required	Type	Description
<interval>	Yes	Integer	The number of specified time units to add

Parameter	Required	Type	Description
<timeUnit>	Yes	String	The unit of time to use with <i>interval</i> : "Second", "Minute", "Hour", "Day", "Week", "Month", "Year"
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is " <code>o</code> " (yyyy-MM-ddTHH:mm:ss.fffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<updated-timestamp>	String	The current timestamp plus the specified number of time units

Example 1

Suppose the current timestamp is "2018-03-01T00:00:00.0000000Z". This example adds five days to that timestamp:

```
getFutureTime(5, 'Day')
```

And returns this result: "2018-03-06T00:00:00.0000000Z"

Example 2

Suppose the current timestamp is "2018-03-01T00:00:00.0000000Z". This example adds five days and converts the result to "D" format:

```
getFutureTime(5, 'Day', 'D')
```

And returns this result: "Tuesday, March 6, 2018"

getPastTime

Return the current timestamp minus the specified time units.

```
getPastTime(<interval>, <timeUnit>, <format>?)
```

Parameter	Required	Type	Description
<code><interval></code>	Yes	Integer	The number of specified time units to subtract
<code><timeUnit></code>	Yes	String	The unit of time to use with <code>interval</code> : "Second", "Minute", "Hour", "Day", "Week", "Month", "Year"
<code><format></code>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is " <code>o</code> " (yyyy-MM-ddTHH:mm:ss.fffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<code><updated-timestamp></code>	String	The current timestamp minus the specified number of time units

Example 1

Suppose the current timestamp is "2018-02-01T00:00:00.0000000Z". This example subtracts five days from that timestamp:

```
getPastTime(5, 'Day')
```

And returns this result: "2018-01-27T00:00:00.0000000Z"

Example 2

Suppose the current timestamp is "2018-02-01T00:00:00.0000000Z". This example subtracts five days and converts the result to "D" format:

```
getPastTime(5, 'Day', 'D')
```

And returns this result: "Saturday, January 27, 2018"

greater

Check whether the first value is greater than the second value. Return true when the first value is more, or return false when less.

```
greater(<value>, <compareTo>)
greater('<value>', '<compareTo>')
```

Parameter	Required	Type	Description
<value>	Yes	Integer, Float, or String	The first value to check whether greater than the second value
<compareTo>	Yes	Integer, Float, or String, respectively	The comparison value

Return Type	Description
value	true or Boolean Return true when the first value is greater than the second value. Return false when the first value is equal to or less than the second value.

Example

These examples check whether the first value is greater than the second value:

```
greater(10, 5)
greater('apple', 'banana')
```

And return these results:

- First example: true
- Second example: false

greaterOrEquals

Check whether the first value is greater than or equal to the second value. Return true when the first value is greater or equal, or return false when the first value is less.

```
greaterOrEquals(<value>, <compareTo>)
greaterOrEquals('<value>', '<compareTo>')
```

Parameter	Required	Type	Description

Parameter	Required	Type	Description
<value>	Yes	Integer, Float, or String	The first value to check whether greater than or equal to the second value
<compareTo>	Yes	Integer, Float, or String, respectively	The comparison value

Return Type	Description
value	true or Boolean false

Return true when the first value is greater than or equal to the second value.
Return false when the first value is less than the second value.

Example

These examples check whether the first value is greater or equal than the second value:

```
greaterOrEquals(5, 5)
greaterOrEquals('apple', 'banana')
```

And return these results:

- First example: `true`
- Second example: `false`

guid

Generate a globally unique identifier (GUID) as a string, for example, "c2ecc88d-88c8-4096-912c-d6f2e2b138ce":

```
guid()
```

Also, you can specify a different format for the GUID other than the default format, "D", which is 32 digits separated by hyphens.

```
guid('<format>')
```

Parameter	Required	Type	Description
<format>	No	String	A single format specifier for the returned GUID. By default, the format is "D", but you can use "N", "D", "B", "P", or "X".

Return value	Type	Description
<GUID-value>	String	A randomly generated GUID

Example

This example generates the same GUID, but as 32 digits, separated by hyphens, and enclosed in parentheses:

```
guid('P')
```

And returns this result: `"(c2ecc88d-88c8-4096-912c-d6f2e2b138ce)"`

if

Check whether an expression is true or false. Based on the result, return a specified value.

```
if(<expression>, <valueIfTrue>, <valueIfFalse>)
```

Parameter	Required	Type	Description
<expression>	Yes	Boolean	The expression to check
<valueIfTrue>	Yes	Any	The value to return when the expression is true
<valueIfFalse>	Yes	Any	The value to return when the expression is false

Return value	Type	Description
<specified-return-value>	Any	The specified value that returns based on whether the expression is true or false

Example

This example returns "yes" because the specified expression returns true. Otherwise, the example returns "no":

```
if>equals(1, 1), 'yes', 'no')
```

indexOf

Return the starting position or index value for a substring. This function isn't case-sensitive, and indexes start with the number 0.

```
indexOf('<text>', '<searchText>')
```

Parameter	Required	Type	Description
<text>	Yes	String	The string that has the substring to find
<searchText>	Yes	String	The substring to find

Return value	Type	Description
<index-value>	Integer	The starting position or index value for the specified substring. If the string isn't found, return the number -1.

Example

This example finds the starting index value for the "world" substring in the "hello world" string:

```
indexOf('hello world', 'world')
```

And returns this result: 6

int

Return the integer version for a string.

```
int('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The string to convert

Return value	Type	Description
<integer-result>	Integer	The integer version for the specified string

Example

This example creates an integer version for the string "10":

```
int('10')
```

And returns this result: `10`

json

Return the JavaScript Object Notation (JSON) type value or object for a string or XML.

```
json('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String or XML	The string or XML to convert

Return Type	Description
<code>value</code>	The JSON native type value or object for the specified string or XML. If the string is null, the function returns an empty object.

Example 1

This example converts this string to the JSON value:

```
json('[1, 2, 3]')
```

And returns this result: [1, 2, 3]

Example 2

This example converts this string to JSON:

```
json('{"fullName": "Sophia Owen"}')
```

And returns this result:

```
{
  "fullName": "Sophia Owen"
}
```

Example 3

This example converts this XML to JSON:

```
json(xml('<?xml version="1.0"?> <root> <person id="1"> <name>Sophia
Owen</name> <occupation>Engineer</occupation> </person> </root>'))
```

And returns this result:

JSON

```
{
  "?xml": { "@version": "1.0" },
  "root": {
    "person": [ {
      "@id": "1",
      "name": "Sophia Owen",
      "occupation": "Engineer"
    } ]
  }
}
```

Return a collection that has *only* the common items across the specified collections. To appear in the result, an item must appear in all the collections passed to this function. If one or more items have the same name, the last item with that name appears in the result.

```
intersection([<collection1>], [<collection2>], ...)  
intersection('<collection1>', '<collection2>', ...)
```

Parameter	Required	Type	Description
<collection1>, <collection2>, ...	Yes	Array or Object, but not both	The collections from where you want <i>only</i> the common items

Return value	Type	Description
<common-items>	Array or Object, respectively	A collection that has only the common items across the specified collections

Example

This example finds the common items across these arrays:

```
intersection(createArray(1, 2, 3), createArray(101, 2, 1, 10),  
createArray(6, 8, 1, 2))
```

And returns an array with *only* these items: [1, 2]

join

Return a string that has all the items from an array and has each character separated by a *delimiter*.

```
join([<collection>], '<delimiter>')
```

Parameter	Required	Type	Description
<collection>	Yes	Array	The array that has the items to join

Parameter	Required	Type	Description
<delimiter>	Yes	String	The separator that appears between each character in the resulting string

Return value	Type	Description
<char1><delimiter><char2> <delimiter>...	String	The resulting string created from all the items in the specified array

Example

This example creates a string from all the items in this array with the specified character as the delimiter:

```
join(createArray('a', 'b', 'c'), '.')
```

And returns this result: "a.b.c"

last

Return the last item from a collection.

```
last('<collection>')
last([<collection>])
```

Parameter	Required	Type	Description
<collection>	Yes	String or Array	The collection where to find the last item

Return value	Type	Description
<last-collection-item>	String or Array, respectively	The last item in the collection

Example

These examples find the last item in these collections:

```
last('abcd')
last(createArray(0, 1, 2, 3))
```

And returns these results:

- First example: "d"
- Second example: 3

lastIndexOf

Return the starting position or index value for the last occurrence of a substring. This function isn't case-sensitive, and indexes start with the number 0.

```
lastIndexOf('<text>', '<searchText>')
```

Parameter	Required	Type	Description
<text>	Yes	String	The string that has the substring to find
<searchText>	Yes	String	The substring to find

Return value	Type	Description
<ending-index-value>	Integer	The starting position or index value for the last occurrence of the specified substring. If the string isn't found, return the number -1.

Example

This example finds the starting index value for the last occurrence of the "world" substring in the "hello world" string:

```
lastIndexOf('hello world', 'world')
```

And returns this result: 6

length

Return the number of items in a collection.

```
length('<collection>')
length([<collection>])
```

Parameter	Required	Type	Description
<collection>	Yes	String or Array	The collection with the items to count

Return value	Type	Description
<length-or-count>	Integer	The number of items in the collection

Example

These examples count the number of items in these collections:

```
length('abcd')
length(createArray(0, 1, 2, 3))
```

And return this result: 4

less

Check whether the first value is less than the second value. Return true when the first value is less, or return false when the first value is more.

```
less(<value>, <compareTo>)
less('<value>', '<compareTo>')
```

Parameter	Required	Type	Description
<value>	Yes	Integer, Float, or String	The first value to check whether less than the second value
<compareTo>	Yes	Integer, Float, or String, respectively	The comparison item

Return	Type	Description
value		

Return Type	Description
true or false	Boolean Return true when the first value is less than the second value. Return false when the first value is equal to or greater than the second value.

Example

These examples check whether the first value is less than the second value.

```
less(5, 10)
less('banana', 'apple')
```

And return these results:

- First example: `true`
- Second example: `false`

lessOrEquals

Check whether the first value is less than or equal to the second value. Return true when the first value is less than or equal, or return false when the first value is more.

```
lessOrEquals(<value>, <compareTo>
lessOrEquals('<value>', '<compareTo>')
```

Parameter	Required	Type	Description
<code><value></code>	Yes	Integer, Float, or String	The first value to check whether less than or equal to the second value
<code><compareTo></code>	Yes	Integer, Float, or String, respectively	The comparison item

Return Type	Description
true or false	Boolean Return true when the first value is less than or equal to the second value. Return false when the first value is greater than the second value.

Example

These examples check whether the first value is less or equal than the second value.

```
lessOrEquals(10, 10)
lessOrEquals('apply', 'apple')
```

And return these results:

- First example: `true`
- Second example: `false`

max

Return the highest value from a list or array with numbers that is inclusive at both ends.

```
max(<number1>, <number2>, ...)
max([<number1>, <number2>, ...])
```

Parameter	Required	Type	Description
<code><number1></code> , <code><number2></code> , ...	Yes	Integer, Float, or both	The set of numbers from which you want the highest value
<code>[<number1>,</code> <code><number2>, ...]</code>	Yes	Array - Integer, Float, or both	The array of numbers from which you want the highest value

Return value	Type	Description
<code><max-value></code>	Integer or Float	The highest value in the specified array or set of numbers

Example

These examples get the highest value from the set of numbers and the array:

```
max(1, 2, 3)
max(createArray(1, 2, 3))
```

And return this result: `3`

min

Return the lowest value from a set of numbers or an array.

```
min(<number1>, <number2>, ...)  
min([<number1>, <number2>, ...])
```

Parameter	Required	Type	Description
<number1>, <number2>, ...	Yes	Integer, Float, or both	The set of numbers from which you want the lowest value
[<number1>, <number2>, ...]	Yes	Array - Integer, Float, or both	The array of numbers from which you want the lowest value

Return value	Type	Description
<min-value>	Integer or Float	The lowest value in the specified set of numbers or specified array

Example

These examples get the lowest value in the set of numbers and the array:

```
min(1, 2, 3)  
min(createArray(1, 2, 3))
```

And return this result: 1

mod

Return the remainder from dividing two numbers. To get the integer result, see [div\(\)](#).

```
mod(<dividend>, <divisor>)
```

Parameter	Required	Type	Description

Parameter	Required	Type	Description
<dividend>	Yes	Integer or Float	The number to divide by the <i>divisor</i>
<divisor>	Yes	Integer or Float	The number that divides the <i>dividend</i> , but can't be 0.

Return value	Type	Description
<modulo-result>	Integer or Float	The remainder from dividing the first number by the second number

Example

This example divides the first number by the second number:

```
mod(3, 2)
```

And return this result: 1

mul

Return the product from multiplying two numbers.

```
mul(<multiplicand1>, <multiplicand2>)
```

Parameter	Required	Type	Description
<multiplicand1>	Yes	Integer or Float	The number to multiply by <i>multiplicand2</i>
<multiplicand2>	Yes	Integer or Float	The number that multiples <i>multiplicand1</i>

Return value	Type	Description
<product-result>	Integer or Float	The product from multiplying the first number by the second number

Example

These examples multiple the first number by the second number:

```
mul(1, 2)  
mul(1.5, 2)
```

And return these results:

- First example: 2
- Second example 3

not

Check whether an expression is false. Return true when the expression is false, or return false when true.

JSON

```
not(<expression>)
```

Parameter	Required	Type	Description
<expression>	Yes	Boolean	The expression to check

Return value	Type	Description
true or false	Boolean	Return true when the expression is false. Return false when the expression is true.

Example 1

These examples check whether the specified expressions are false:

JSON

```
not(false)  
not(true)
```

And return these results:

- First example: The expression is false, so the function returns true.
- Second example: The expression is true, so the function returns false.

Example 2

These examples check whether the specified expressions are false:

JSON

```
not>equals(1, 2))  
not>equals(1, 1))
```

And return these results:

- First example: The expression is false, so the function returns `true`.
- Second example: The expression is true, so the function returns `false`.

Or

Check whether at least one expression is true. Return true when at least one expression is true, or return false when both are false.

```
or(<expression1>, <expression2>)
```

Parameter	Required	Type	Description
<code><expression1>, <expression2></code>	Yes	Boolean	The expressions to check

Return value	Type	Description
<code>true</code> or <code>false</code>	Boolean	Return true when at least one expression is true. Return false when both expressions are false.

Example 1

These examples check whether at least one expression is true:

JSON

```
or(true, false)  
or(false, false)
```

And return these results:

- First example: At least one expression is true, so the function returns `true`.
- Second example: Both expressions are false, so the function returns `false`.

Example 2

These examples check whether at least one expression is true:

JSON

```
or>equals(1, 1), equals(1, 2))  
or>equals(1, 2), equals(1, 3))
```

And return these results:

- First example: At least one expression is true, so the function returns `true`.
- Second example: Both expressions are false, so the function returns `false`.

rand

Return a random integer from a specified range, which is inclusive only at the starting end.

```
rand(<minValue>, <maxValue>)
```

Parameter	Required	Type	Description
<code><minValue></code>	Yes	Integer	The lowest integer in the range
<code><maxValue></code>	Yes	Integer	The integer that follows the highest integer in the range that the function can return

Return value	Type	Description
<code><random-result></code>	Integer	The random integer returned from the specified range

Example

This example gets a random integer from the specified range, excluding the maximum value:

```
rand(1, 5)
```

And returns one of these numbers as the result: 1, 2, 3, or 4

range

Return an integer array that starts from a specified integer.

```
range(<startIndex>, <count>)
```

Parameter	Required	Type	Description
<startIndex>	Yes	Integer	An integer value that starts the array as the first item
<count>	Yes	Integer	The number of integers in the array

Return value	Type	Description
[<range-result>]	Array	The array with integers starting from the specified index

Example

This example creates an integer array that starts from the specified index and has the specified number of integers:

```
range(1, 4)
```

And returns this result: [1, 2, 3, 4]

replace

Replace a substring with the specified string, and return the result string. This function is case-sensitive.

```
replace('<text>', '<oldText>', '<newText>')
```

Parameter	Required	Type	Description
<text>	Yes	String	The string that has the substring to replace
<oldText>	Yes	String	The substring to replace

Parameter	Required	Type	Description
<newText>	Yes	String	The replacement string

Return value	Type	Description
<updated-text>	String	The updated string after replacing the substring If the substring isn't found, return the original string.

Example

This example finds the "old" substring in "the old string" and replaces "old" with "new":

```
replace('the old string', 'old', 'new')
```

And returns this result: "the new string"

skip

Remove items from the front of a collection, and return *all the other* items.

```
skip([<collection>], <count>)
```

Parameter	Required	Type	Description
<collection>	Yes	Array	The collection whose items you want to remove
<count>	Yes	Integer	A positive integer for the number of items to remove at the front

Return value	Type	Description
[<updated-collection>]	Array	The updated collection after removing the specified items

Example

This example removes one item, the number 0, from the front of the specified array:

```
skip(createArray(0, 1, 2, 3), 1)
```

And returns this array with the remaining items: [1,2,3]

split

Return an array that contains substrings, separated by commas, based on the specified delimiter character in the original string.

```
split('<text>', '<delimiter>')
```

Parameter	Required	Type	Description
<text>	Yes	String	The string to separate into substrings based on the specified delimiter in the original string
<delimiter>	Yes	String	The character in the original string to use as the delimiter

Return value	Type	Description
[<substring1>, <substring2>,...]	Array	An array that contains substrings from the original string, separated by commas

Example

This example creates an array with substrings from the specified string based on the specified character as the delimiter:

```
split('a_b_c', '_')
```

And returns this array as the result: ["a", "b", "c"]

startOfDay

Return the start of the day for a timestamp.

```
startOfDay('<timestamp>', '<format>?')
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is "o" (yyyy-MM-ddTHH:mm:ss.fffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<updated- timestamp>	String	The specified timestamp but starting at the zero-hour mark for the day

Example

This example finds the start of the day for this timestamp:

```
startOfDay('2018-03-15T13:30:30Z')
```

And returns this result: "2018-03-15T00:00:00.000000Z"

startOfDay

Return the start of the day for a timestamp.

```
startOfDay('<timestamp>', '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is "o" (yyyy-MM-ddTHH:mm:ss.fffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<updated- timestamp>	String	The specified timestamp but starting at the zero-minute mark for the hour

Example

This example finds the start of the hour for this timestamp:

```
startOfHour('2018-03-15T13:30:30Z')
```

And returns this result: "2018-03-15T13:00:00.0000000Z"

startOfMonth

Return the start of the month for a timestamp.

```
startOfMonth('<timestamp>', '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is "o" (yyyy-MM-ddTHH:mm:ss.fffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<updated-timestamp>	String	The specified timestamp but starting on the first day of the month at the zero-hour mark

Example

This example returns the start of the month for this timestamp:

```
startOfMonth('2018-03-15T13:30:30Z')
```

And returns this result: "2018-03-01T00:00:00.0000000Z"

startsWith

Check whether a string starts with a specific substring. Return true when the substring is found, or return false when not found. This function isn't case-sensitive.

```
startsWith('<text>', '<searchText>')
```

Parameter	Required	Type	Description
<text>	Yes	String	The string to check
<searchText>	Yes	String	The starting string to find

Return value	Type	Description
true or false	Boolean	Return true when the starting substring is found. Return false when not found.

Example 1

This example checks whether the "hello world" string starts with the "hello" substring:

```
startsWith('hello world', 'hello')
```

And returns this result: `true`

Example 2

This example checks whether the "hello world" string starts with the "greetings" substring:

```
startsWith('hello world', 'greetings')
```

And returns this result: `false`

string

Return the string version for a value.

```
string(<value>)
```

Parameter	Required	Type	Description
<value>	Yes	Any	The value to convert

Return value	Type	Description
<string-value>	String	The string version for the specified value

Example 1

This example creates the string version for this number:

```
string(10)
```

And returns this result: "10"

Example 2

This example creates a string for the specified JSON object and uses the backslash character (\) as an escape character for the double-quotation mark (").

```
string( { "name": "Sophie Owen" } )
```

And returns this result: "{ \\\"name\\\": \\"Sophie Owen\\" }"

sub

Return the result from subtracting the second number from the first number.

```
sub(<minuend>, <subtrahend>)
```

Parameter	Required	Type	Description
<minuend>	Yes	Integer or Float	The number from which to subtract the <i>subtrahend</i>

Parameter	Required	Type	Description
<subtrahend>	Yes	Integer or Float	The number to subtract from the <i>minuend</i>

Return value	Type	Description
<result>	Integer or Float	The result from subtracting the second number from the first number

Example

This example subtracts the second number from the first number:

```
sub(10.3, .3)
```

And returns this result: 10

substring

Return characters from a string, starting from the specified position, or index. Index values start with the number 0.

```
substring('<text>', <startIndex>, <length>)
```

Parameter	Required	Type	Description
<text>	Yes	String	The string whose characters you want
<startIndex>	Yes	Integer	A positive number equal to or greater than 0 that you want to use as the starting position or index value
<length>	Yes	Integer	A positive number of characters that you want in the substring

Return value	Type	Description
<substring>	String	A substring with the specified number of characters, starting at the specified index position in the source string

Example

This example creates a five-character substring from the specified string, starting from the index value 6:

```
substring('hello world', 6, 5)
```

And returns this result: "world"

subtractFromTime

Subtract a number of time units from a timestamp. See also [getPastTime](#).

```
subtractFromTime('<timestamp>', <interval>, '<timeUnit>', '<format>'?)
```

Parameter	Required	Type	Description
<timestamp>	Yes	String	The string that contains the timestamp
<interval>	Yes	Integer	The number of specified time units to subtract
<timeUnit>	Yes	String	The unit of time to use with <i>interval</i> : "Second", "Minute", "Hour", "Day", "Week", "Month", "Year"
<format>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is "o" (yyyy-MM-ddTHH:mm:ss.ffffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<updated-timestamp>	String	The timestamp minus the specified number of time units

Example 1

This example subtracts one day from this timestamp:

```
subtractFromTime('2018-01-02T00:00:00Z', 1, 'Day')
```

And returns this result: "2018-01-01T00:00:00:000000Z"

Example 2

This example subtracts one day from this timestamp:

```
subtractFromTime('2018-01-02T00:00:00Z', 1, 'Day', 'D')
```

And returns this result using the optional "D" format: "Monday, January, 1, 2018"

take

Return items from the front of a collection.

```
take('<collection>', <count>)
take([<collection>], <count>)
```

Parameter	Required	Type	Description
<collection>	Yes	String or Array	The collection whose items you want
<count>	Yes	Integer	A positive integer for the number of items that you want from the front

Return value	Type	Description
<subset> or [<subset>]	String or Array, respectively	A string or array that has the specified number of items taken from the front of the original collection

Example

These examples get the specified number of items from the front of these collections:

```
take('abcde', 3)
take(createArray(0, 1, 2, 3, 4), 3)
```

And return these results:

- First example: "abc"
- Second example: [0, 1, 2]

ticks

Return the `ticks` property value for a specified timestamp. A *tick* is a 100-nanosecond interval.

```
ticks('<timestamp>')
```

Parameter	Required	Type	Description
<code><timestamp></code>	Yes	String	The string for a timestamp

Return value	Type	Description
<code><ticks-number></code>	Integer	The number of ticks that have elapsed since 12:00:00 midnight, January 1, 0001 in the Gregorian calendar since the input timestamp

toLower

Return a string in lowercase format. If a character in the string doesn't have a lowercase version, that character stays unchanged in the returned string.

```
toLower('<text>')
```

Parameter	Required	Type	Description
<code><text></code>	Yes	String	The string to return in lowercase format

Return value	Type	Description
<code><lowercase-text></code>	String	The original string in lowercase format

Example

This example converts this string to lowercase:

```
toLowerCase('Hello World')
```

And returns this result: "hello world"

toUpperCase

Return a string in uppercase format. If a character in the string doesn't have an uppercase version, that character stays unchanged in the returned string.

```
toUpperCase('<text>')
```

Parameter	Required	Type	Description
<text>	Yes	String	The string to return in uppercase format

Return value	Type	Description
<uppercase-text>	String	The original string in uppercase format

Example

This example converts this string to uppercase:

```
toUpperCase('Hello World')
```

And returns this result: "HELLO WORLD"

trim

Remove leading and trailing whitespace from a string, and return the updated string.

```
trim('<text>')
```

Parameter	Required	Type	Description
-----------	----------	------	-------------

Parameter	Required	Type	Description
<text>	Yes	String	The string that has the leading and trailing whitespace to remove

Return value	Type	Description
<updatedText>	String	An updated version for the original string without leading or trailing whitespace

Example

This example removes the leading and trailing whitespace from the string " Hello World ":

```
trim(' Hello World  ')
```

And returns this result: "Hello World"

union

Return a collection that has *all* the items from the specified collections. To appear in the result, an item can appear in any collection passed to this function. If one or more items have the same name, the last item with that name appears in the result.

```
union('<collection1>', '<collection2>', ...)
union([<collection1>], [<collection2>], ...)
```

Parameter	Required	Type	Description
<collection1>, <collection2>, ...	Yes	Array or Object, but not both	The collections from where you want <i>all</i> the items

Return value	Type	Description
<updatedCollection>	Array or Object, respectively	A collection with all the items from the specified collections - no duplicates

Example

This example gets *all* the items from these collections:

```
union(createArray(1, 2, 3), createArray(1, 2, 10, 101))
```

And returns this result: [1, 2, 3, 10, 101]

uriComponent

Return a uniform resource identifier (URI) encoded version for a string by replacing URL-unsafe characters with escape characters. Use this function rather than [encodeUriComponent\(\)](#). Although both functions work the same way, `uriComponent()` is preferred.

```
uriComponent('<value>')
```

Parameter	Required	Type	Description
<code><value></code>	Yes	String	The string to convert to URI-encoded format

Return value	Type	Description
<code><encoded-uri></code>	String	The URI-encoded string with escape characters

Example

This example creates a URI-encoded version for this string:

```
uriComponent('https://contoso.com')
```

And returns this result: "http%3A%2F%2Fcontoso.com"

uriComponentToBinary

Return the binary version for a uniform resource identifier (URI) component.

```
uriComponentToBinary('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The URI-encoded string to convert

Return value	Type	Description
<binary-for-encoded-uri>	String	The binary version for the URI-encoded string. The binary content is base64-encoded and represented by <code>\$content</code> .

Example

This example creates the binary version for this URI-encoded string:

```
uriComponentToBinary('http%3A%2F%2Fcontoso.com')
```

And returns this result:

```
"001000100110100001110100011101000111000000100101001100  
11010000010010010100110010010001100010010010100110010010001  
10011000110110111101101110011101000110111101110011011011  
110010111001100011011011110110110100100010"
```

uriComponentToString

Return the string version for a uniform resource identifier (URI) encoded string, effectively decoding the URI-encoded string.

```
uriComponentToString('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The URI-encoded string to decode

Return value	Type	Description
<decoded-uri>	String	The decoded version for the URI-encoded string

Example

This example creates the decoded string version for this URL-encoded string:

```
uriComponentToString('http%3A%2F%2Fcontoso.com')
```

And returns this result: "https://contoso.com"

utcNow

Return the current timestamp.

```
utcNow('<format>')
```

Optionally, you can specify a different format with the *<format>* parameter.

Parameter	Required	Type	Description
<i><format></i>	No	String	Either a single format specifier or a custom format pattern . The default format for the timestamp is "o" (yyyy-MM-ddTHH:mm:ss.fffffffK), which complies with ISO 8601 and preserves time zone information.

Return value	Type	Description
<i><current-timestamp></i>	String	The current date and time

Example 1

Suppose today is April 15, 2018 at 1:00:00 PM. This example gets the current timestamp:

```
utcNow()
```

And returns this result: "2018-04-15T13:00:00.000000Z"

Example 2

Suppose today is April 15, 2018 at 1:00:00 PM. This example gets the current timestamp using the optional "D" format:

```
utcNow('D')
```

And returns this result: "Sunday, April 15, 2018"

xml

Return the XML version for a string that contains a JSON object.

```
xml('<value>')
```

Parameter	Required	Type	Description
<value>	Yes	String	The string with the JSON object to convert The JSON object must have only one root property, which can't be an array. Use the backslash character (\) as an escape character for the double quotation mark (").

Return value	Type	Description
<xml-version>	Object	The encoded XML for the specified string or JSON object

Example 1

This example creates the XML version for this string, which contains a JSON object:

```
xml(json('{ \"name\": \"Sophia Owen\" }'))
```

And returns this result XML:

XML

```
<name>Sophia Owen</name>
```

Example 2

Suppose you have this JSON object:

JSON

```
{
  "person": {
    "name": "Sophia Owen",
    "city": "Seattle"
  }
}
```

This example creates XML for a string that contains this JSON object:

```
xml(json('{"person": {"name": "Sophia Owen", "city": "Seattle"}'))
```

And returns this result XML:

XML

```
<person>
  <name>Sophia Owen</name>
  <city>Seattle</city>
<person>
```

xpath

Check XML for nodes or values that match an XPath (XML Path Language) expression, and return the matching nodes or values. An XPath expression, or just "XPath", helps you navigate an XML document structure so that you can select nodes or compute values in the XML content.

```
xpath('<xml>', '<xpath>')
```

Parameter	Required	Type	Description
<xml>	Yes	Any	The XML string to search for nodes or values that match an XPath expression value
<xpath>	Yes	Any	The XPath expression used to find matching XML nodes or values

Return value	Type	Description
<xml-node>	XML	An XML node when only a single node matches the specified XPath expression

Return value	Type	Description
<value>	Any	The value from an XML node when only a single value matches the specified XPath expression
[<xml-node1>, <xml-node2>, ...] -or- [<value1>, <value2>, ...]	Array	An array with XML nodes or values that match the specified XPath expression

Example 1

Following on Example 1, this example finds nodes that match the `<count></count>` node and adds those node values with the `sum()` function:

```
xpath(xml(parameters('items')), 'sum(/produce/item/count)')
```

And returns this result: 30

Example 2

For this example, both expressions find nodes that match the `<location></location>` node, in the specified arguments, which include XML with a namespace. The expressions use the backslash character (\) as an escape character for the double quotation mark (").

- *Expression 1*

```
xpath(xml(body('Http')), '/*[name()=\"file\"]/*[name()=\"location\"]')
```

- *Expression 2*

```
xpath(xml(body('Http')), '/*[local-name()=\"file\" and namespace-uri()=\"http://contoso.com\"]/*[local-name()=\"location\"]')
```

Here are the arguments:

- This XML, which includes the XML document namespace,

```
xmlns="http://contoso.com":
```

XML

```
<?xml version="1.0"?> <file xmlns="http://contoso.com">
<location>Paris</location> </file>
```

- Either XPath expression here:

- `/*[name()=\"file\"]/*[name()=\"location\"]]`
- `/*[local-name()=\"file\" and namespace-uri()=\"http://contoso.com\"]/*[local-name()=\"location\"]]`

Here's the result node that matches the `<location></location>` node:

XML

```
<location xmlns="https://contoso.com">Paris</location>
```

Example 3

Following on Example 3, this example finds the value in the `<location></location>` node:

```
xpath(xml(body('Http')), 'string(/*[name()=\"file\"]/*[name()=\"location\"])' )
```

And returns this result: "Paris"

ⓘ Note

One can add comments to data flow expressions, but not in pipeline expressions.

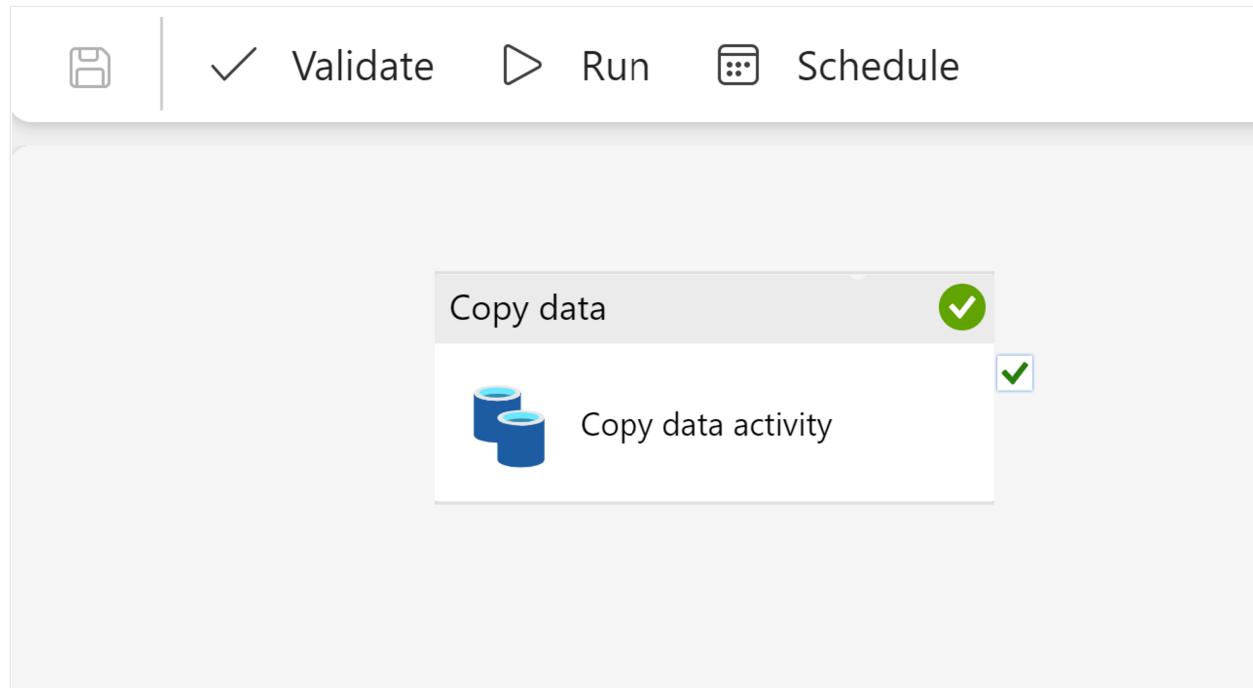
Next steps

For instructions on general parameter usage, refer to [Parameters for Data Factory in Fabric](#)

Concept: Data pipeline Runs

Article • 05/23/2023

A data pipeline run occurs when a data pipeline is executed. This means that the activities in your data pipeline will run and be executed to completion. For example, running a data pipeline with a **Copy data** activity will perform that action and copy your data. Each data pipeline run will have its own unique pipeline run ID.



A data pipeline run can be triggered one of two ways, either on-demand or by setting up a schedule. A scheduled pipeline will be able to run based on the time and frequency that you set.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

On-demand data pipeline run

To manually trigger a data pipeline run, select **Run** found in the top banner of the **Home** tab.

The screenshot shows the Azure Data Factory pipeline editor interface. At the top, there is a navigation bar with tabs: Home (selected), Activities, and View. Below the navigation bar is a toolbar with icons for Save, Validate, Run, and Schedule. The 'Run' icon is highlighted with a red box. The main area displays a pipeline named 'Copy data' which contains a single activity named 'Copy data activity'. This activity is marked with a green checkmark. Below the pipeline, there are tabs for Parameters (selected), Variables, Settings, and Output. A 'New' button is located at the bottom left of the pipeline area.

You'll be prompted to save your changes before triggering the pipeline run. Select **Save and run** to continue.

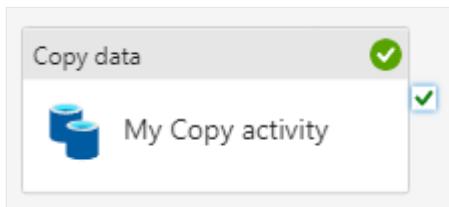
The screenshot shows the same Azure Data Factory pipeline editor interface as the previous one, but with a modal dialog box overlaid. The dialog is titled 'Save and run?' and contains the message: 'You have unsaved changes. If you continue, the pipeline will be saved and run.' At the bottom of the dialog, there are two buttons: 'Save and run' (highlighted with a red box) and 'Cancel'. The background of the pipeline editor is dimmed to indicate it is inactive while the dialog is open.

After your changes are saved, your pipeline will run. You can view the progress of the run in the **Output** tab found at the bottom of the canvas.

The screenshot shows the Azure Data Factory interface. At the top, there are tabs for Home, Activities, View, Validate, Run, and Schedule. Below this is a search bar and a plus sign icon. The main area displays a pipeline run titled "Copy data". Underneath it, a "Copy data activity" is listed. The "Output" tab is selected, indicated by a red border around the tab name. Below the tabs, the pipeline run ID is shown as "b42c1334-c33e-4220-b47c-305525d385e8". The table below contains the following data:

Name	Type	Run start	Duration	Status	Run ID
Copy data activity	Copy data	Jul 21, 2022, 1:51:13 pm	00:00:19	In progress	b5b91277-c438-4d24-bcd1-12fc

Once an activity has completed in a run, a green check mark appears in the corner of the activity.



Once the entire pipeline has been executed and the output status updates to **Succeeded**, you've had a successful pipeline run!

The screenshot shows the same pipeline run details as the previous image, but the status has now updated to "Succeeded". The "Status" column is highlighted with a red box. The table data remains the same as in the previous screenshot.

Name	Type	Run start	Duration	Status	Run ID
Copy data activity	Copy data	Jul 21, 2022, 1:51:13 pm	00:00:19	Succeeded	b5b91277-c438-4d24-bcd1-12fc

Scheduled data pipeline runs

When you schedule a data pipeline run, you can choose the frequency that your pipeline runs. Select **Schedule**, found in the top banner of the **Home** tab, to view your options. By default, your data pipeline won't be set on a schedule.

Screenshot of the Azure Data Factory Home page showing the Schedule button highlighted.

The screenshot shows the Azure Data Factory interface. At the top, there are tabs: Home (selected), Activities, Run, View. Below the tabs is a toolbar with icons for Save, Delete, Settings, Refresh, Validate, Run, and Schedule. The Schedule icon is highlighted with a red box. In the main area, there is a card titled "Copy data" with a green checkmark and the sub-tile "My Copy activity". Below the card, there are tabs for Parameters (selected), Variables, Settings, and Output. A "New" button is also visible.

On the Schedule configuration page, you can specify a schedule frequency, start and end dates and times, and time zone.

Screenshot of the Schedule configuration page for a Data pipeline named "SamplePipeline".

The page displays the following information:

- Search bar: Last success is in April 26, 2023 at 2:07:01 PM (UTC-05:00) Eastern Time (US and Canada)
- About: The scheduled refresh is turned off. There is a "Run" button.
- Sensitivity label: Endorsement
- Schedule tab (selected):
 - Scheduled run: On (radio button selected)
 - Repeat: By the minute
 - Every: 15 minute(s)
 - Start: 05/23/2023 08:07 AM
 - End: 05/26/2023 08:46 AM
 - Time zone: (UTC-08:00) Pacific Time (US and Canada)

At the bottom are "Apply" and "Discard" buttons, and a magnifying glass search icon.

Once configured, select **Apply** to set your schedule. You can view or edit the schedule again anytime by selecting the **Schedule** button again.

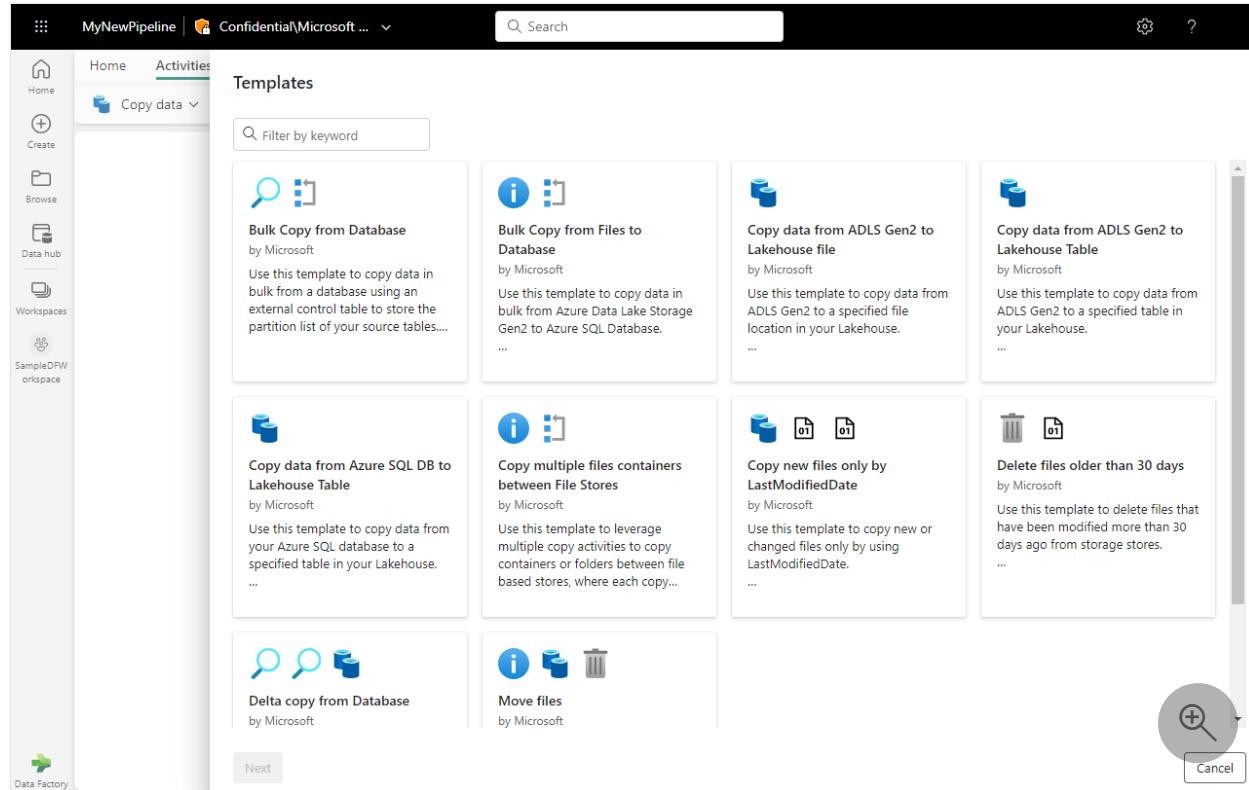
Next steps

- [How to monitor data pipeline runs in Microsoft Fabric](#)
- [Quickstart: Create your first data pipeline to copy data](#)

Templates for Data Factory in Microsoft Fabric

Article • 05/23/2023

Templates are pre-defined pipelines that allow you to get started quickly with Data Factory. These templates help to reduce development time by providing an easy way to create pipelines for common data integration scenarios.



ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

How to build a pipeline from a template

1. To get started with a template, start by selecting **New** and then **Data pipeline**.

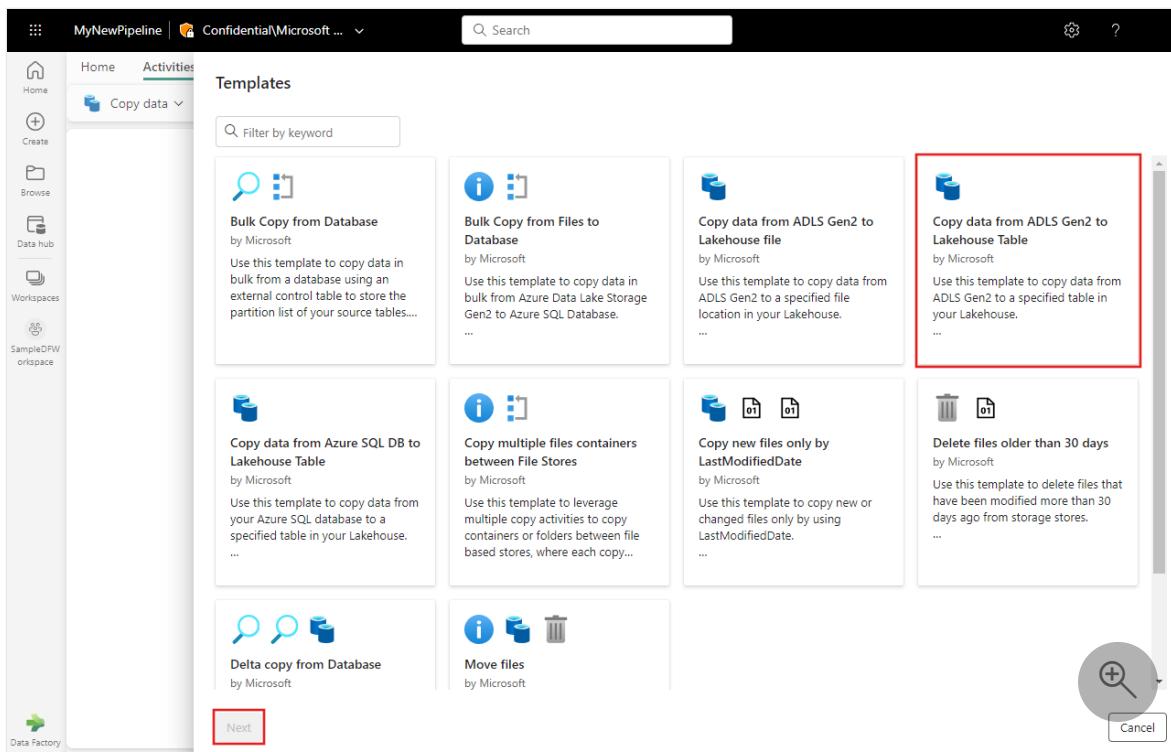
The screenshot shows the Microsoft Data Factory workspace interface. On the left, there's a sidebar with icons for Home, Create, Browse, Data hub, Workspaces, and SampleDFWorkspace. The main area has a search bar at the top. Below it, there's a button for 'New' with a dropdown menu open, showing options like 'Dataflow Gen2 (Preview)' and 'Data pipeline'. The 'Data pipeline' option is highlighted with a red box. A tooltip for 'Data pipeline' says 'Ingest data at scale and schedule data workflows.' To the right is a table listing pipelines, showing columns for Name, Type, Owner, Refreshed, and Next refresh. Pipelines listed include 'Notebook', 'pipeline1', 'SampleLakehouse', 'SampleLakehouse', 'SampleLakehouse', and 'SamplePipeline'.

2. Set a name for your new pipeline and select **Create**. The pipeline editor window opens.

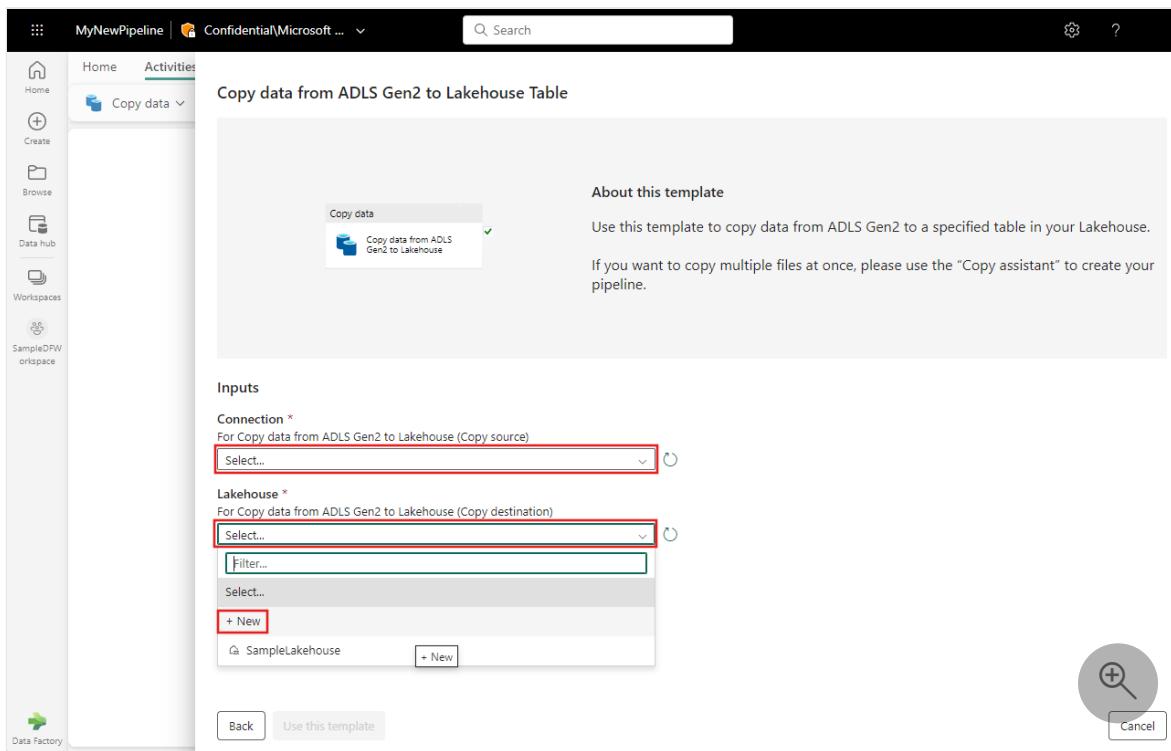
3. Select **Choose a task to start**.

The screenshot shows the Microsoft Data Factory pipeline editor for 'MyNewPipeline'. The left sidebar includes Home, Activities, View, and tabs for Validate, Run, Schedule, and View run history. The main area has a heading 'Start building your data pipeline' and three cards: 'Add pipeline activity', 'Copy data', and 'Choose a task to start'. The 'Choose a task to start' card is highlighted with a red box.

4. The template browser appears. Select the **Copy data from ADLS Gen2 to Lakehouse Table** template, and then select **Next**.

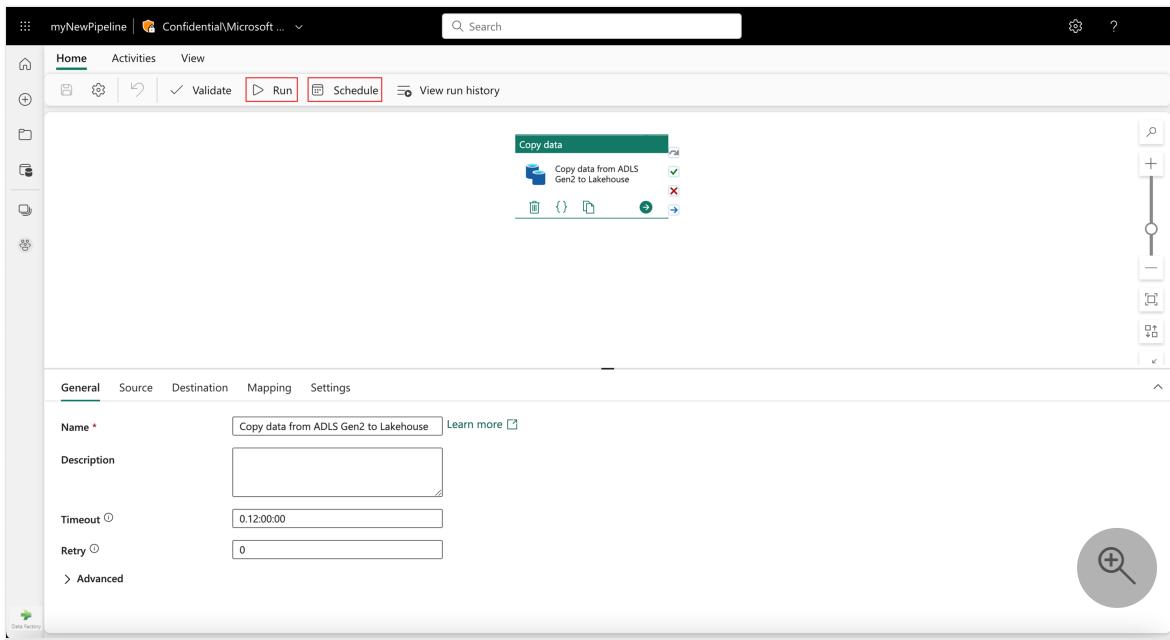


5. Select from the drop-down list your source and destination connections or use + New to create a new connection.



6. After making your selections or creating new connections, select the **Use this template** button to generate a new pipeline directly.

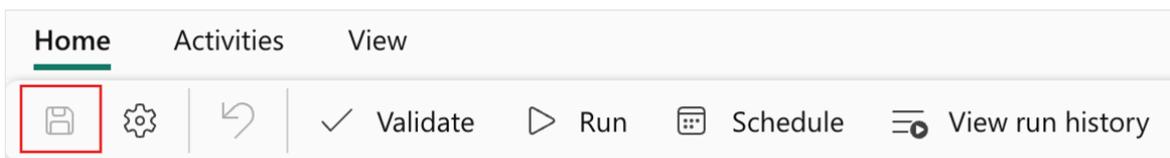
7. The pipeline is created using the connections you set. You can view the newly created pipeline in the pipeline editor, where you can use the **Run** and **Schedule** buttons to control its execution.



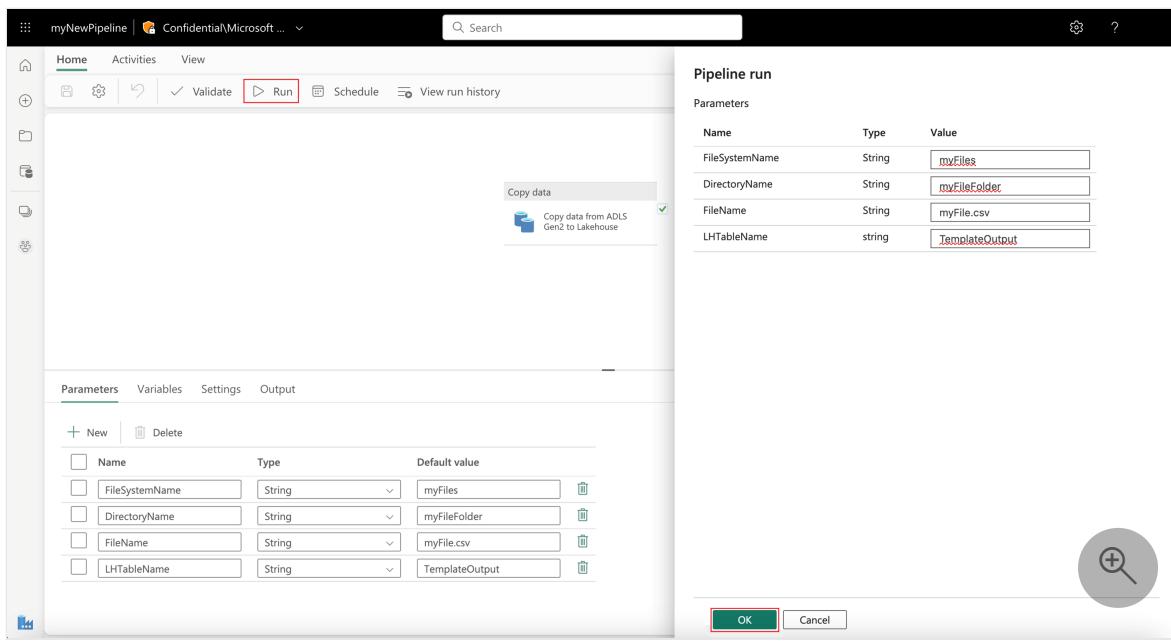
8. Initially the new Copy activity is highlighted on the editor canvas, and its properties shown in the properties pane at the bottom of the editor.
9. When you select the background of the pipeline canvas, you can see the general pipeline properties in the properties pane, where you can add or edit existing parameters.

A screenshot of the pipeline properties pane. The top navigation bar has tabs for Parameters, Variables, Settings, and Output. The Parameters tab is selected and highlighted with a red border. Below the tabs is a table for managing parameters. The table has columns for Name, Type, and Default value. It contains four rows with parameter names: FileSystemName,DirectoryName,FileName, and LHTableName. Each row includes a checkbox for Name, a dropdown for Type (set to String), and a text input field for Default value. There are also 'New' and 'Delete' buttons at the top of the table.

10. When you're done, save your edits by selecting the save button on the toolbar of the **Home** tab.



11. Now you can run your pipeline, providing values for any parameters as required.



Next steps

[How to monitor pipeline runs in Microsoft Fabric](#)

How to monitor data pipeline runs in Microsoft Fabric

Article • 05/23/2023

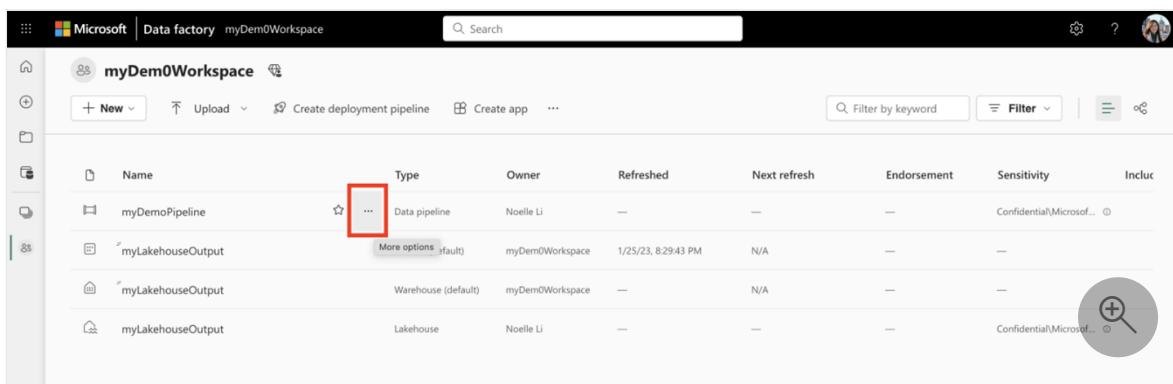
In this how-to guide, you'll learn the different ways to review and monitor your pipeline runs.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Monitor data pipeline runs

1. To monitor your data pipeline runs, hover over your pipeline in your workspace. Doing so will bring up three dots to the right of your pipeline name.



The screenshot shows the Microsoft Fabric Data factory workspace. On the left is a sidebar with icons for Home, New, Upload, Create deployment pipeline, Create app, and three dots. The main area shows a list of pipelines under 'myDem0Workspace'. The first item, 'myDemoPipeline', has a red box around its three-dot menu icon. A tooltip 'More options (1 fault)' is visible above the menu. The table columns are Name, Type, Owner, Refreshed, Next refresh, Endorsement, Sensitivity, and Includ. There are three other pipeline entries below it: 'myLakehouseOutput' (Warehouse), 'myLakehouseOutput' (Lakehouse), and another 'myLakehouseOutput' (Lakehouse). On the far right of the table is a circular search icon with a magnifying glass and a plus sign.

2. Select the three dots to find a list of options. Then select **View run history**. This action opens a fly-out on the right side of your screen with all your recent runs and run statuses.

Microsoft | Data factory myDem0Workspace

+ New Upload Create deployment pipeline Create app ...

Search Filter

Name	Type	Owner	Refreshed	Next refresh	Endorsement	Sensitivity	Inclu...
myDemoPipeline	Data pipeline	Noelle Li	—	—	—	Confidential\Microsoft...	...
myLakehouseOutput	Dataset (default)	myDem0Workspace	Open	1/25/23, 8:29:43 PM	N/A	—	—
myLakehouseOutput	Warehouse (default)	myDem0Workspace	Delete	—	N/A	—	—
myLakehouseOutput	Lakehouse	Noelle Li	Settings	—	—	—	...
			Add to Favorites	—	—	Confidential\Microsoft...	...
			View lineage	—	—	—	...
			View run history	—	—	—	...

Microsoft | Data factory myDem0Workspace

+ New Upload Create deployment pipeline Create app ...

Search Filter

Name	Type	Owner
myDemoPipeline	Data pipeline	Noelle Li
myLakehouseOutput	Dataset (default)	myDem0Workspace
myLakehouseOutput	Warehouse (default)	myDem0Workspace
myLakehouseOutput	Lakehouse	Noelle Li

Recent runs

Refresh Export as csv

myDemoPipeline

1 of 1

Run start	Run end	Duration	Triggered by	Status
1/25/2023, 8:50:01 PM	1/25/2023, 8:50:23 PM	00:00:22	ScheduleTrigger	✓ Succeeded
1/25/2023, 8:45:00 PM	1/25/2023, 8:45:22 PM	00:00:22	ScheduleTrigger	✓ Succeeded
1/25/2023, 8:40:00 PM	1/25/2023, 8:40:21 PM	00:00:21	ScheduleTrigger	✓ Succeeded
1/25/2023, 8:35:00 PM	1/25/2023, 8:35:19 PM	00:00:19	ScheduleTrigger	✓ Succeeded
1/25/2023, 8:32:15 PM	1/25/2023, 8:32:37 PM	00:00:22	Manual	✓ Succeeded
1/25/2023, 8:29:54 PM	1/25/2023, 8:30:19 PM	00:00:25	Manual	✓ Succeeded

3. Use the Filter to find specific data pipeline runs. You can filter on **Status** or on **End time**.

myDemoPipeline

Recent runs

⟳ Refresh ⚡ Export as csv

1 of 1

Run start	Run end	Duration	Triggered by
1/25/2023, 8:50:01 PM	1/25/2023, 8:50:23 PM	00:00:22	ScheduleTrigger
1/25/2023, 8:45:00 PM	1/25/2023, 8:45:22 PM	00:00:22	ScheduleTrigger
1/25/2023, 8:40:00 PM	1/25/2023, 8:40:21 PM	00:00:21	ScheduleTrigger
1/25/2023, 8:35:00 PM	1/25/2023, 8:35:19 PM	00:00:19	ScheduleTrigger
1/25/2023, 8:32:15 PM	1/25/2023, 8:32:37 PM	00:00:22	Manual
1/25/2023, 8:29:54 PM	1/25/2023, 8:30:19 PM	00:00:25	Manual

Filter

Clear applied filters

Status

- All
- Succeeded
- In Progress
- Queued
- Failed
- Canceled

End time

- Last 24 hours
- Last 7 days
- Last 30 days

4. Select one of your pipeline runs to view detailed information. You'll be able to view what your pipeline looks like and view more properties like Run ID or errors if your pipeline run failed.

The screenshot shows the Microsoft Data Factory interface. On the left, there's a sidebar with icons for New, Upload, Create deployment pipeline, and Create app. The main area shows the 'myDem0Workspace' workspace. A search bar at the top right has the placeholder 'Search'. Below it, a 'Recent runs' section with a 'Filter' button. The 'myDemoPipeline' run details are shown, including its name, type (Data pipeline), owner (Noelle L.), and a list of 7 runs. The first run is highlighted with a red box. The 'Status' column for all runs shows green checkmarks indicating success. A magnifying glass icon is overlaid on the bottom right of the run table.

Run start	Run end	Duration	Triggered by	Status
1/25/2023, 8:50:01 PM	1/25/2023, 8:50:23 PM	00:00:22	ScheduleTrigger	<input checked="" type="checkbox"/> Succeeded
1/25/2023, 8:45:00 PM	1/25/2023, 8:45:22 PM	00:00:22	ScheduleTrigger	<input checked="" type="checkbox"/> Succeeded
1/25/2023, 8:40:00 PM	1/25/2023, 8:40:21 PM	00:00:21	ScheduleTrigger	<input checked="" type="checkbox"/> Succeeded
1/25/2023, 8:35:00 PM	1/25/2023, 8:35:19 PM	00:00:19	ScheduleTrigger	<input checked="" type="checkbox"/> Succeeded
1/25/2023, 8:32:15 PM	1/25/2023, 8:32:37 PM	00:00:22	Manual	<input checked="" type="checkbox"/> Succeeded
1/25/2023, 8:29:54 PM	1/25/2023, 8:30:19 PM	00:00:25	Manual	<input checked="" type="checkbox"/> Succeeded

The screenshot shows the Microsoft Data Factory interface for pipeline runs. At the top, it says "All pipeline runs > myDemoPipeline". Below this, there are tabs for "List" and "Gantt". The main area displays a hierarchical tree of activities: "ForEach" which contains "ForEach_4i1", which in turn contains "Activities" and "Copy_4i1". Below this, the "Activity runs" section shows a table of six items. One row for "Copy_4i1" is selected and highlighted with a red border. To the right of the table, a "Details" pop-up window is open, providing specific information about the selected activity.

Activity name	Run start	Duration	Status	Run ID
ForEach_4i1	1/25/2023, 8:50:01 PM	00:00:22	Succeeded	d3c80328-6a8a-47b7-9a1e-d32f4ca17b3d
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:13	Succeeded	b2b1b978-a1b5-487f-b581-361d281ebaba
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:16	Succeeded	a72f1663-d3c7-4191-a8bc-7ba2ff4414f0b
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:14	Succeeded	1f3b6546-4076-4d7a-b980-a129a05f2966
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:13	Succeeded	b2b1b978-a1b5-487f-b581-361d281ebaba
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:16	Succeeded	a72f1663-d3c7-4191-a8bc-7ba2ff4414f0b

5. To find additional information on your pipeline runs **Input** and **Output**, hover over an activity row and select either the **Input** or **Output** icon. Details will be shown in a pop-up.

This screenshot shows the same "Activity runs" table from the previous image, but with a red box highlighting the second row for "Copy_4i1". To the right of the table, there is a large magnifying glass icon.

Input

Copy to clipboard

```
{
  "source": {
    "type": "AzureSqlSource",
    "partitionOption": "None",
    "datasetSettings": {
      "type": "AzureSqlTable",
      "typeProperties": {
        "schema": "SalesLT",
        "table": "ProductDescription"
      }
    }
  }
}
```

Activity runs

Activity name	Run start	Duration	Status	Run ID
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:13	Succeeded	b2b1b978-a1b5-487f-b581-361d281ef6aba
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:16	Succeeded	a72f1663-d3c7-4191-a8bc-7ba244140b0
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:14	Succeeded	1f3b6546-4076-4d7a-b980-a129a05f2966

6. To view performance details, hover over an activity row and select the glasses icon. Performance details will pop up.

Activity runs

Activity name	Run start	Duration	Status	Run ID
ForEach_4i1	1/25/2023, 8:50:01 PM	00:00:22	Succeeded	d3c80328-6a8a-47b7-9a1e-d32f4ca17b3d
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:13	Succeeded	b2b1b978-a1b5-487f-b581-361d281ef6aba
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:16	Succeeded	a72f1663-d3c7-4191-a8bc-7ba244140b0
Copy_4i1	1/25/2023, 8:50:02 PM	00:00:14	Succeeded	1f3b6546-4076-4d7a-b980-a129a05f2966

All pipeline runs > **myDemoPipeline**

Copy data details

Copy_4i1

Source: Azure SQL Database → **Destination**: Lakehouse

Data read: 125.828 KB
Rows read: 762

Data written: 78.212 KB
Files written: 1
Rows written: 762

Status: Succeeded
Start time: 1/25/2023, 8:50:02 PM

Pipeline run activity ID: b2b1b978-a1b5-487f-b581-361d281ef6aba

Throughput: 31.457 KB/s
Total duration: 00:00:10

> Duration breakdown
> Advanced

Close

More details can be found under Duration breakdown and Advanced.

Copy data details

Data read:		Data written:	
Rows read:	762	Files written:	1
		Rows written:	762

Status	✓ Succeeded
Start time	1/25/2023, 8:50:02 PM

Pipeline run activity ID

Throughput	31.457 KB/s
Total duration	00:00:10

✗ Duration breakdown

Start time	1/25/2023, 8:50:04 PM
Used parallel copies	1

Queue 

Transfer 
00:00:05
Period: Queue
Description: The elapsed time until the copy activity actually starts on the service.

✗ Advanced

Triggered by	ScheduleTrigger
--------------	-----------------

Close

7. If your data pipeline failed, view the error message by hovering over the activity row and select the message icon under Status. This selection will bring up error details, such as the error code and message.

Activity runs					View run detail	▲
Pipeline run ID					Export to CSV	▼
All status	Container	Run start	Duration	Status	Run ID	1
Showing 1 - 7 items						
Activity name		Run start	Duration	Status	Run ID	
ForEach_4i1		1/25/2023, 9:30:30 PM	00:00:17	✓ Succeeded	acd4d558-5014-4af0c-b97a-c71e399424cb	
Copy_4i1		1/25/2023, 9:30:31 PM	00:00:14	✓ Succeeded	e2ed3b19-b3fd-47fb-b3c1-6269a9b049a1	
Copy_4i1		1/25/2023, 9:30:31 PM	00:00:13	✓ Succeeded	58324b60-1604-43c2-ac12-e2156c0c11a3	
Copy_4i1		1/25/2023, 9:30:31 PM	00:00:13	✓ Succeeded	14a82738-2978-468a-b2b6-bf37b67f37e6	
Copy_4i1		1/25/2023, 9:30:31 PM	00:00:13	✓ Succeeded	ee8b520d-8cba-41b4-a88a-75a843a9e396	
Copy_4i1		1/25/2023, 9:30:31 PM	00:00:12	✓ Succeeded	db7141fa-71c3-47fd-b892-697a9932ca22	
Script1		1/25/2023, 9:30:46 PM	00:00:09	✗ Failed 	2b9d6ab9-c922-4598-bd68-02ed7c232dac	

The screenshot shows the 'Activity runs' section of the Azure Data Factory pipeline run history. A red box highlights the error details for the 'Script1' activity, which failed at 9:30:46 PM on 1/25/2023. The error message is as follows:

```
{
  "errorCode": "2011",
  "message": "Incorrect syntax near the keyword 'is'.",
  "failureType": "UserError",
  "target": "Script1",
  "details": []
}
```

Below the error details, the pipeline run status is shown as 'Failed'.

8. Select **Update pipeline** to make changes to your pipeline. This selection will land you back in the pipeline canvas.

The screenshot shows the pipeline canvas for the 'myDemoPipeline'. A red box highlights the 'Update pipeline' button. The 'Script1' activity is highlighted with a red circle, indicating it is the failed activity.

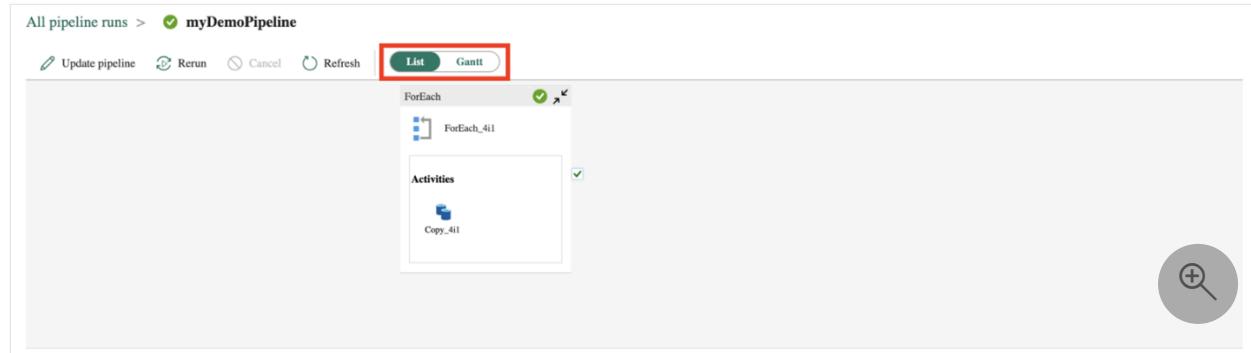
9. You can also **Rerun** your data pipeline. You can choose to rerun the entire pipeline or only rerun the pipeline from the failed activity.

The screenshot shows the 'All pipeline runs' page for 'myDemoPipeline'. A red box highlights the 'Rerun' dropdown menu. The 'Script1' activity is highlighted with a red circle.

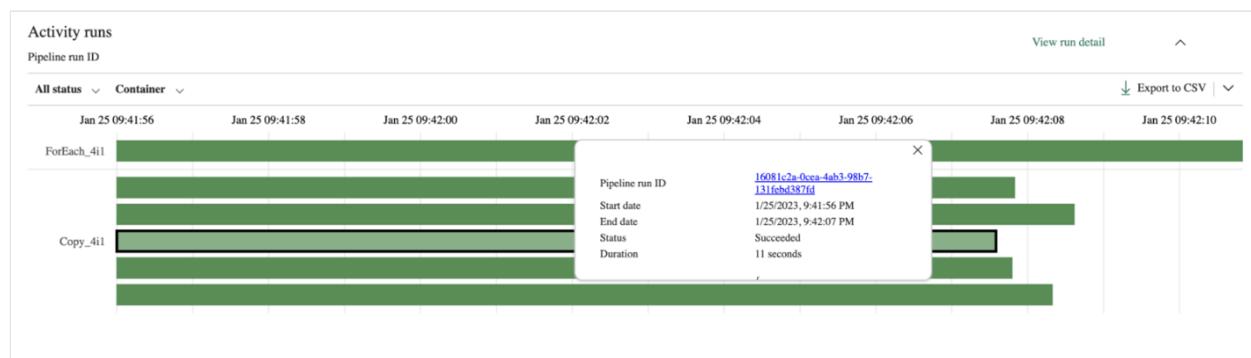
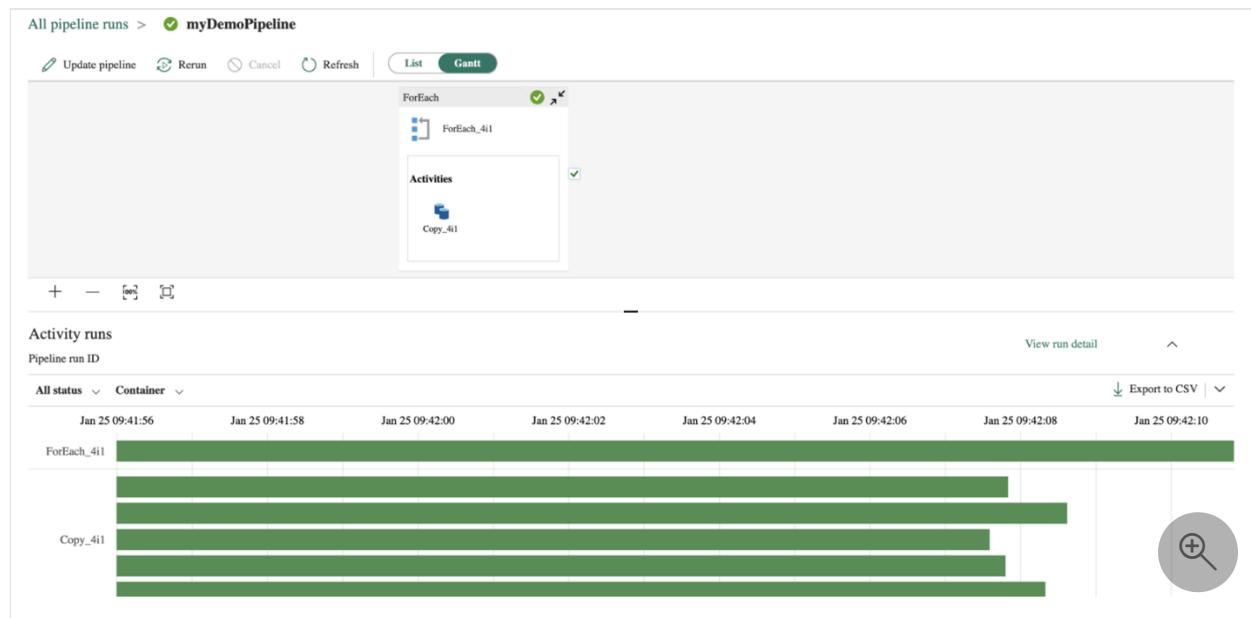
The screenshot shows the 'All pipeline runs' page for 'myDemoPipeline'. A red box highlights the 'Rerun' options: 'Rerun' and 'Rerun from failed activity'. The pipeline canvas shows the 'Script1' activity highlighted with a red circle.

Gantt view

A Gantt chart is a view that lets you see the run history over a time range. If you switch to a Gantt view, all pipeline runs will be grouped by name, displayed as bars relative to how long the run took.



The length of the bar relates to the duration of the pipeline. You can select the bar to view more details.



Next steps

- Quickstart: Create your first data pipeline to copy data

- Quickstart: Create your first Dataflow Gen2 to get and transform data

Browse data pipeline runs in the Monitoring hub

Article • 05/23/2023

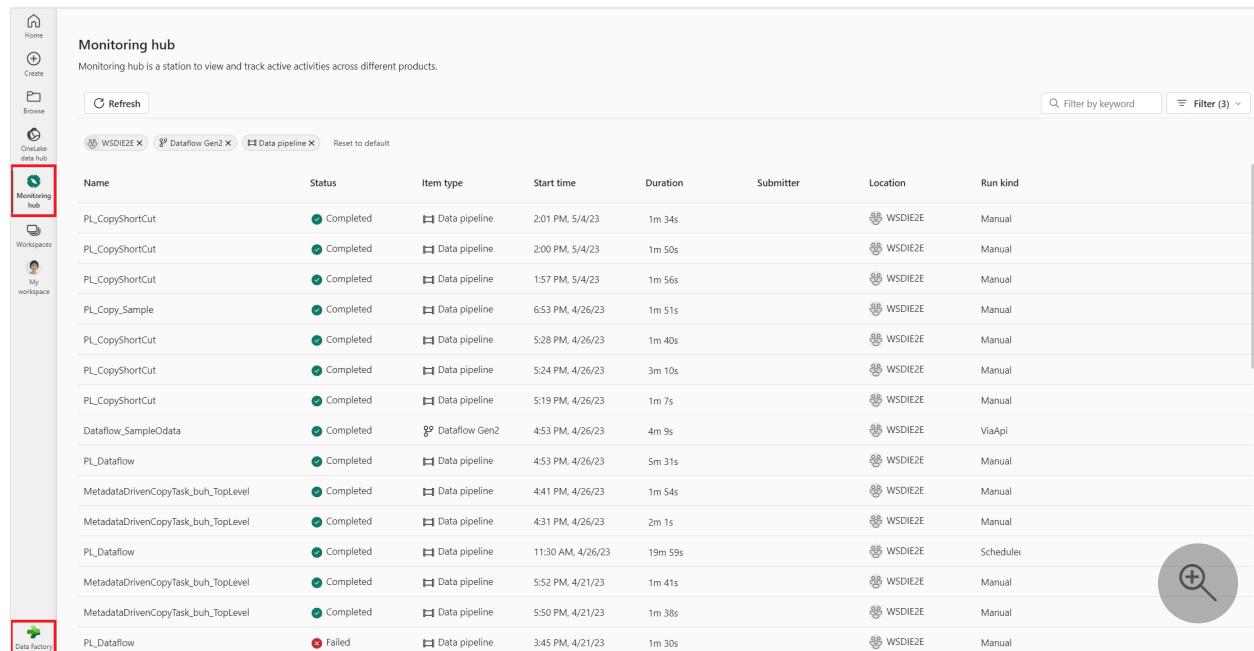
The Monitoring hub serves as a centralized portal for browsing data pipeline runs across items, when you are in the **Data Factory** or **Data Engineering** experience.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Access the monitoring hub

In the **Data Factory** or **Data Engineering** experience, you can access the Monitoring hub to view various data pipeline runs by selecting **Monitoring hub** in the left-side navigation links.



The screenshot shows the Microsoft Fabric Monitoring hub. On the left, there is a vertical navigation bar with icons for Home, Create, Browse, OneLake data hub, Monitoring hub (which is highlighted with a red box), Workspaces, My workspace, and Data Factory (which is also highlighted with a red box). The main area is titled "Monitoring hub" and contains a table of data pipeline runs. The table has columns for Name, Status, Item type, Start time, Duration, Submitter, Location, and Run kind. The data includes various pipeline names like PL_CopyShortCut, PL_Copy_Sample, and PL_Dataflow, all in completed status. A search bar and a filter button are at the top of the table. A magnifying glass icon is in the bottom right corner of the table area.

Name	Status	Item type	Start time	Duration	Submitter	Location	Run kind
PL_CopyShortCut	Completed	Data pipeline	2:01 PM, 5/4/23	1m 34s	WSDIE2E	Manual	
PL_CopyShortCut	Completed	Data pipeline	2:00 PM, 5/4/23	1m 50s	WSDIE2E	Manual	
PL_CopyShortCut	Completed	Data pipeline	1:57 PM, 5/4/23	1m 56s	WSDIE2E	Manual	
PL_Copy_Sample	Completed	Data pipeline	6:53 PM, 4/26/23	1m 51s	WSDIE2E	Manual	
PL_CopyShortCut	Completed	Data pipeline	5:28 PM, 4/26/23	1m 40s	WSDIE2E	Manual	
PL_CopyShortCut	Completed	Data pipeline	5:24 PM, 4/26/23	3m 10s	WSDIE2E	Manual	
PL_CopyShortCut	Completed	Data pipeline	5:19 PM, 4/26/23	1m 7s	WSDIE2E	Manual	
Dataflow_Sampleodata	Completed	Dataflow Gen2	4:53 PM, 4/26/23	4m 9s	WSDIE2E	ViaApi	
PL_Dataflow	Completed	Data pipeline	4:53 PM, 4/26/23	5m 31s	WSDIE2E	Manual	
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	4:41 PM, 4/26/23	1m 54s	WSDIE2E	Manual	
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	4:31 PM, 4/26/23	2m 1s	WSDIE2E	Manual	
PL_Dataflow	Completed	Data pipeline	11:30 AM, 4/26/23	19m 59s	WSDIE2E	Schedule	
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	5:52 PM, 4/21/23	1m 41s	WSDIE2E	Manual	
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	5:50 PM, 4/21/23	1m 38s	WSDIE2E	Manual	
PL_Dataflow	Failed	Data pipeline	3:45 PM, 4/21/23	1m 30s	WSDIE2E	Manual	

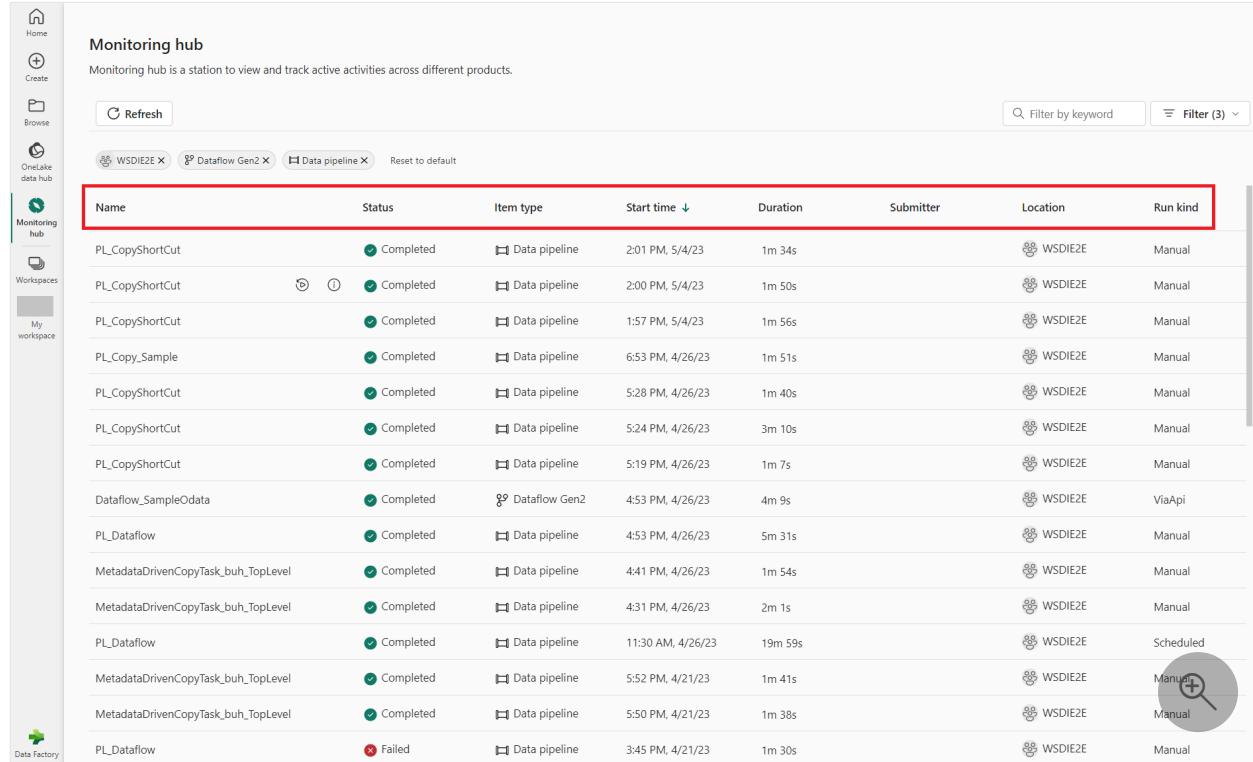
Sort, search and filter data pipeline runs

For better usability and discoverability, you can sort the data pipeline runs by selecting different columns in the UI. You can also filter the pipeline runs based on different

columns and search for specific pipeline runs.

Sort data pipeline runs

To sort data pipeline runs, you can select on each column header, such as **Name**, **Status**, **Item type**, **Start time**, **Location**, **Run kind**, and so on.



The screenshot shows the Azure Data Factory Monitoring hub interface. On the left, there's a sidebar with icons for Home, Create, Browse, OneLake data hub, Workspaces, My workspace, and Data Factory. The main area is titled "Monitoring hub" and contains a table of pipeline runs. The table has columns: Name, Status, Item type, Start time, Duration, Submitter, Location, and Run kind. The "Start time" column is currently sorted in descending order, as indicated by the downward arrow icon. The table lists various pipeline runs, mostly completed, with some failing or pending. A red box highlights the "Start time" column header. At the top right of the table, there are "Filter by keyword" and "Filter (3)" buttons. The bottom right corner of the screenshot has a circular watermark with a magnifying glass icon and the word "Manual".

Name	Status	Item type	Start time	Duration	Submitter	Location	Run kind
PL_CopyShortCut	Completed	Data pipeline	2:01 PM, 5/4/23	1m 34s	WSDIE2E	WSDIE2E	Manual
PL_CopyShortCut	Completed	Data pipeline	2:00 PM, 5/4/23	1m 50s	WSDIE2E	WSDIE2E	Manual
PL_CopyShortCut	Completed	Data pipeline	1:57 PM, 5/4/23	1m 56s	WSDIE2E	WSDIE2E	Manual
PL_Copy_Sample	Completed	Data pipeline	6:53 PM, 4/26/23	1m 51s	WSDIE2E	WSDIE2E	Manual
PL_CopyShortCut	Completed	Data pipeline	5:28 PM, 4/26/23	1m 40s	WSDIE2E	WSDIE2E	Manual
PL_CopyShortCut	Completed	Data pipeline	5:24 PM, 4/26/23	3m 10s	WSDIE2E	WSDIE2E	Manual
PL_CopyShortCut	Completed	Data pipeline	5:19 PM, 4/26/23	1m 7s	WSDIE2E	WSDIE2E	Manual
Dataflow_SampleOdata	Completed	Dataflow Gen2	4:53 PM, 4/26/23	4m 9s	WSDIE2E	WSDIE2E	ViaApi
PL_Dataflow	Completed	Data pipeline	4:53 PM, 4/26/23	5m 31s	WSDIE2E	WSDIE2E	Manual
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	4:41 PM, 4/26/23	1m 54s	WSDIE2E	WSDIE2E	Manual
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	4:31 PM, 4/26/23	2m 1s	WSDIE2E	WSDIE2E	Manual
PL_Dataflow	Completed	Data pipeline	11:30 AM, 4/26/23	19m 59s	WSDIE2E	WSDIE2E	Scheduled
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	5:52 PM, 4/21/23	1m 41s	WSDIE2E	WSDIE2E	Manual
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	5:50 PM, 4/21/23	1m 38s	WSDIE2E	WSDIE2E	Manual
PL_Dataflow	Failed	Data pipeline	3:45 PM, 4/21/23	1m 30s	WSDIE2E	WSDIE2E	Manual

Filter data pipeline runs

You can filter data pipeline runs by **Status**, **Item Type**, **Start Time**, **Submitter**, and **Location** using the Filter pane in the upper-right corner.

Monitoring hub
Monitoring hub is a station to view and track active activities across different products.

Refresh Filter (3) Filter by keyword

Name	Status	Item type	Start time	Duration	Submitter
PL_CopyShortCut	Completed	Data pipeline	2:01 PM, 5/4/23	1m 34s	
PL_CopyShortCut	Completed	Data pipeline	2:00 PM, 5/4/23	1m 50s	
PL_CopyShortCut	Completed	Data pipeline	1:57 PM, 5/4/23	1m 56s	WSDIE2E
PL_Copy_Sample	Completed	Data pipeline	6:53 PM, 4/26/23	1m 51s	WSDIE2E
PL_CopyShortCut	Completed	Data pipeline	5:28 PM, 4/26/23	1m 40s	WSDIE2E
PL_CopyShortCut	Completed	Data pipeline	5:24 PM, 4/26/23	3m 10s	WSDIE2E
PL_CopyShortCut	Completed	Data pipeline	5:19 PM, 4/26/23	1m 7s	WSDIE2E
Dataflow_SampleOdata	Completed	Dataflow Gen2	4:53 PM, 4/26/23	4m 9s	WSDIE2E
PL_Dataflow	Completed	Data pipeline	4:53 PM, 4/26/23	5m 31s	WSDIE2E
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	4:41 PM, 4/26/23	1m 54s	WSDIE2E
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	4:31 PM, 4/26/23	2m 1s	WSDIE2E
PL_Dataflow	Completed	Data pipeline	11:30 AM, 4/26/23	19m 59s	WSDIE2E
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	5:52 PM, 4/21/23	1m 41s	WSDIE2E
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	5:50 PM, 4/21/23	1m 38s	WSDIE2E
PL_Dataflow	Failed	Data pipeline	3:45 PM, 4/21/23	1m 30s	WSDIE2E

Search data pipeline runs

To search for specific data pipelines, you can enter certain keywords in the search box located in the upper-right corner.

Monitoring hub
Monitoring hub is a station to view and track active activities across different products.

Refresh Note Filter (3) Filter by keyword

Name	Status	Item type	Start time	Duration	Submitter	Location	Run kind
PL_NotebookSampleHoliday	Failed	Data pipeline	10:09 AM, 4/21/23	3m 7s		WSDIE2E	Manual
PL_NotebookSampleHoliday	Completed	Data pipeline	9:35 AM, 4/21/23	1m 59s		WSDIE2E	Scheduled
PL_NotebookSampleHoliday	Completed	Data pipeline	9:25 AM, 4/21/23	1m 59s		WSDIE2E	Scheduled
PL_NotebookSampleHoliday	Completed	Data pipeline	8:53 AM, 4/21/23	1m 43s		WSDIE2E	Manual
PL_NotebookSampleHoliday	Failed	Data pipeline	11:20 AM, 4/10/23	1m 48s		WSDIE2E	Manual
PL_NotebookSampleHoliday	Failed	Data pipeline	11:18 AM, 4/10/23	1m 30s		WSDIE2E	Manual
PL_NotebookSampleHoliday	Completed	Data pipeline	2:28 PM, 3/13/23	2m 0s		WSDIE2E	Scheduled
PL_NotebookSampleHoliday	Completed	Data pipeline	2:27 PM, 3/13/23	1m 33s		WSDIE2E	Manual

Manage a data pipeline run

When you hover over a data pipeline run row, you can see various row-level actions that enable you to manage a particular data pipeline run.

View data pipeline run detail pane

You can hover over a data pipeline run row and click the **View detail** icon to open the **Detail** pane and view more details about a data pipeline run.

Monitoring hub

Monitoring hub is a station to view and track active activities across different products.

Name	Status	Item type	Start time	Duration	Submitter	
PL_CopyShortCut	Completed	Data pipeline	2:01 PM, 5/4/23	1m 34s		
PL_CopyShortCut	Completed	Data pipeline	2:00 PM, 5/4/23	1m 50s		
PL_CopyShortCut	Completed	Data pipeline	1:57 PM, 5/4/23	1m 56s		
PL_Copy_Sample	Completed	Data pipeline	6:53 PM, 4/26/23	1m 51s		
PL_CopyShortCut	Completed	Data pipeline	5:28 PM, 4/26/23	1m 40s		
PL_CopyShortCut	Completed	Data pipeline	5:24 PM, 4/26/23	3m 10s		
PL_CopyShortCut	Completed	Data pipeline	5:19 PM, 4/26/23	1m 7s		
Dataflow_SampleOdata	Completed	Dataflow Gen2	4:53 PM, 4/26/23	4m 9s	WSDIE2E	ViaApi
PL_Dataflow	Completed	Data pipeline	4:53 PM, 4/26/23	5m 31s	WSDIE2E	Manual
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	4:41 PM, 4/26/23	1m 54s	WSDIE2E	Manual
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	4:31 PM, 4/26/23	2m 1s	WSDIE2E	Manual
PL_Dataflow	Completed	Data pipeline	11:30 AM, 4/26/23	19m 59s	WSDIE2E	Scheduled
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	5:52 PM, 4/21/23	1m 41s	WSDIE2E	
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	5:50 PM, 4/21/23	1m 38s	WSDIE2E	
PL_Dataflow	Failed	Data pipeline	3:45 PM, 4/21/23	1m 30s	WSDIE2E	Manual

Details

General

Name: PL_Copy_Sample

Owner:

Start time: 4/26/2023, 6:53 PM

End time: 4/26/2023, 6:53 PM

Status: Succeeded

Pipeline run ID: c46425be-8ea1-4166-a752-4c727fe08bcb

Schedule: Every day at 11:30

Related artifacts: Lakehouse DILake

Location: WSDIE2E

Run kind: ViaApi

Submitter: WSDIE2E

Last modified: 4/26/2023, 6:53 PM

Last modified by: WSDIE2E

Last modified location: WSDIE2E

Last modified run kind: ViaApi

Last modified submitter: WSDIE2E

Retry a data pipeline run

If you need to retry a completed data pipeline run, hover over its row and click the **Retry** icon.

Monitoring hub

Monitoring hub is a station to view and track active activities across different products.

Name	Status	Item type	Start time	Duration	Submitter	Location	Run kind
PL_CopyShortCut	Completed	Data pipeline	2:01 PM, 5/4/23	1m 34s	WSDIE2E	Manual	
PL_CopyShortCut	Completed	Data pipeline	2:00 PM, 5/4/23	1m 50s	WSDIE2E	Manual	
PL_CopyShortCut	Completed	Data pipeline	1:57 PM, 5/4/23	1m 56s	WSDIE2E	Manual	
PL_Copy_Sample	Completed	Data pipeline	6:53 PM, 4/26/23	1m 51s	WSDIE2E	Manual	
PL_CopyShortCut	Completed	Data pipeline	5:28 PM, 4/26/23	1m 40s	WSDIE2E	Manual	
PL_CopyShortCut	Completed	Data pipeline	5:24 PM, 4/26/23	3m 10s	WSDIE2E	Manual	
PL_CopyShortCut	Completed	Data pipeline	5:19 PM, 4/26/23	1m 7s	WSDIE2E	Manual	
Dataflow_SampleOdata	Completed	Dataflow Gen2	4:53 PM, 4/26/23	4m 9s	WSDIE2E	ViaApi	
PL_Dataflow	Completed	Data pipeline	4:53 PM, 4/26/23	5m 31s	WSDIE2E	Manual	
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	4:41 PM, 4/26/23	1m 54s	WSDIE2E	Manual	
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	4:31 PM, 4/26/23	2m 1s	WSDIE2E	Manual	
PL_Dataflow	Completed	Data pipeline	11:30 AM, 4/26/23	19m 59s	WSDIE2E	Scheduled	
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	5:52 PM, 4/21/23	1m 41s	WSDIE2E		
MetadataDrivenCopyTask_buh_TopLevel	Completed	Data pipeline	5:50 PM, 4/21/23	1m 38s	WSDIE2E		
PL_Dataflow	Failed	Data pipeline	3:45 PM, 4/21/23	1m 30s	WSDIE2E	Manual	

Details

General

Name: PL_Copy_Sample

Owner:

Start time: 4/26/2023, 6:53 PM

End time: 4/26/2023, 6:53 PM

Status: Succeeded

Pipeline run ID: c46425be-8ea1-4166-a752-4c727fe08bcb

Schedule: Every day at 11:30

Related artifacts: Lakehouse DILake

Location: WSDIE2E

Run kind: ViaApi

Submitter: WSDIE2E

Last modified: 4/26/2023, 6:53 PM

Last modified by: WSDIE2E

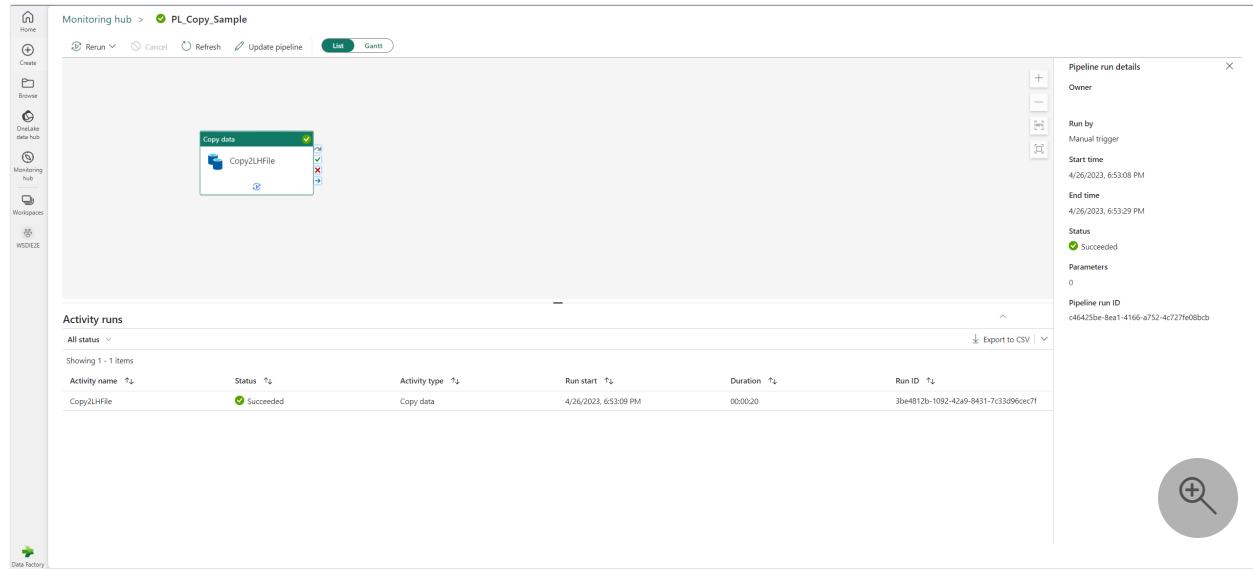
Last modified location: WSDIE2E

Last modified run kind: ViaApi

Last modified submitter: WSDIE2E

Navigate to data pipeline run detail view

If you need more information about detail activity runs of the data pipeline run, you can click on the name of a data pipeline run to navigate to its corresponding data pipeline run detail page.



The screenshot shows the Azure Data Factory Monitoring hub interface. In the top navigation bar, the pipeline name 'PL_Copy_Sample' is selected. Below the navigation bar, there are buttons for 'Rerun', 'Cancel', 'Refresh', 'Update pipeline', and tabs for 'List' and 'Gantt'. On the left side, a sidebar includes links for Home, Create, Browse, Monitor data hub, Monitoring hub, Workspaces, and WSO2 IDE. The main area displays a 'Copy data' activity named 'Copy2LHFile' with a green checkmark indicating success. To the right, a detailed pane titled 'Pipeline run details' provides information such as Owner, Run by (Manual trigger), Start time (4/26/2023, 6:53:08 PM), End time (4/26/2023, 6:53:29 PM), Status (Succeeded), Parameters (0), Pipeline run ID (c46425be-8ea1-4166-a752-4c727e088cb), and an 'Export to CSV' button. A magnifying glass icon is located in the bottom right corner of this pane.

Next steps

- Quickstart: Create your first data pipeline to copy data
- Quickstart: Create your first Dataflows Gen2 to get and transform data

Copy from Azure Blob Storage to Lakehouse

Article • 05/23/2023

In this tutorial, you'll build a data pipeline to move a CSV file from an input folder of an Azure Blob Storage source to a Lakehouse destination.

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Prerequisites

To get started, you must complete the following prerequisites:

- Make sure you have a Project Microsoft Fabric enabled Workspace: [Create a workspace](#).
- Select the **Try it now!** button to prepare the Azure Blob Storage data source of the Copy. Create a new resource group for this Azure Blob Storage and select **Review + Create > Create**.

 **Try it now!** 

Project details

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

Subscription *	contososubscription
Resource group *	(New) datafactorygetstarted
	Create new

Instance details

Region *	East US
Location	[resourceGroup().location]

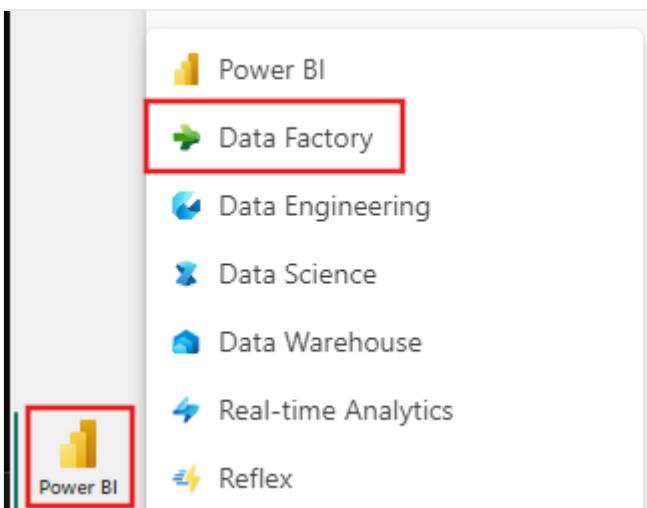
[Review + create](#) < Previous Next : Review + create >

Then an Azure Blob Storage will be created and [moviesDB2.csv](#) will be uploaded to the input folder of the created Azure Blob Storage.

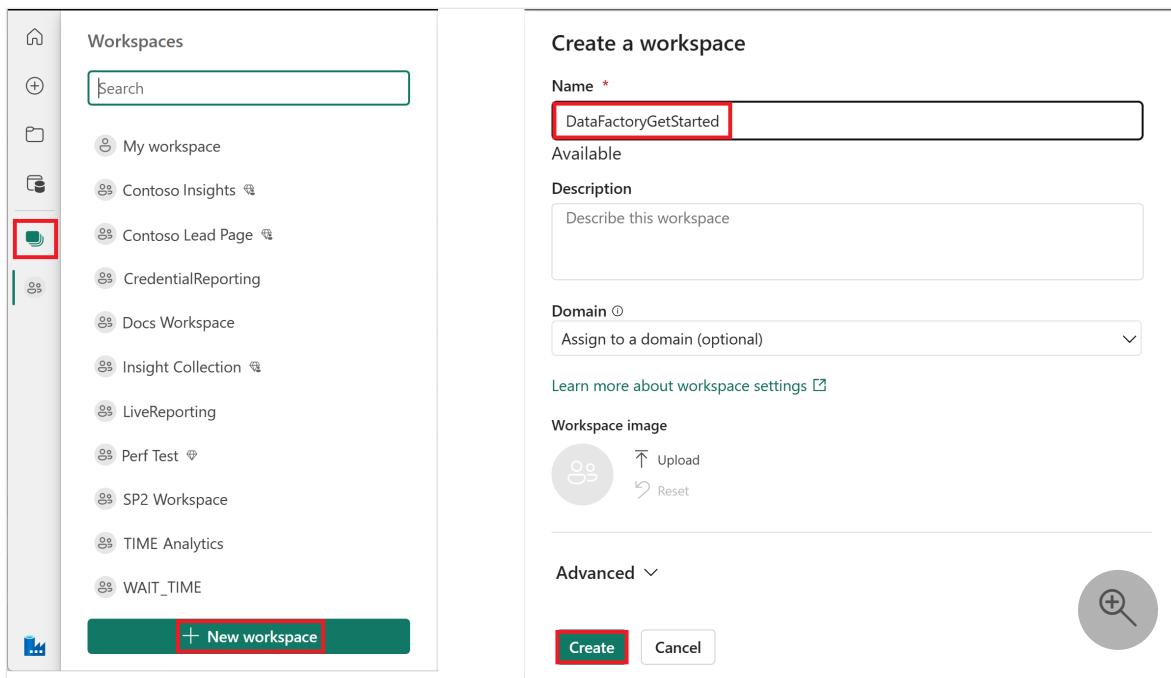
Name ↑↓	Type ↑↓
<input type="checkbox"/>  storagewyjf6p3l6x2lg	Storage account

Create a data pipeline

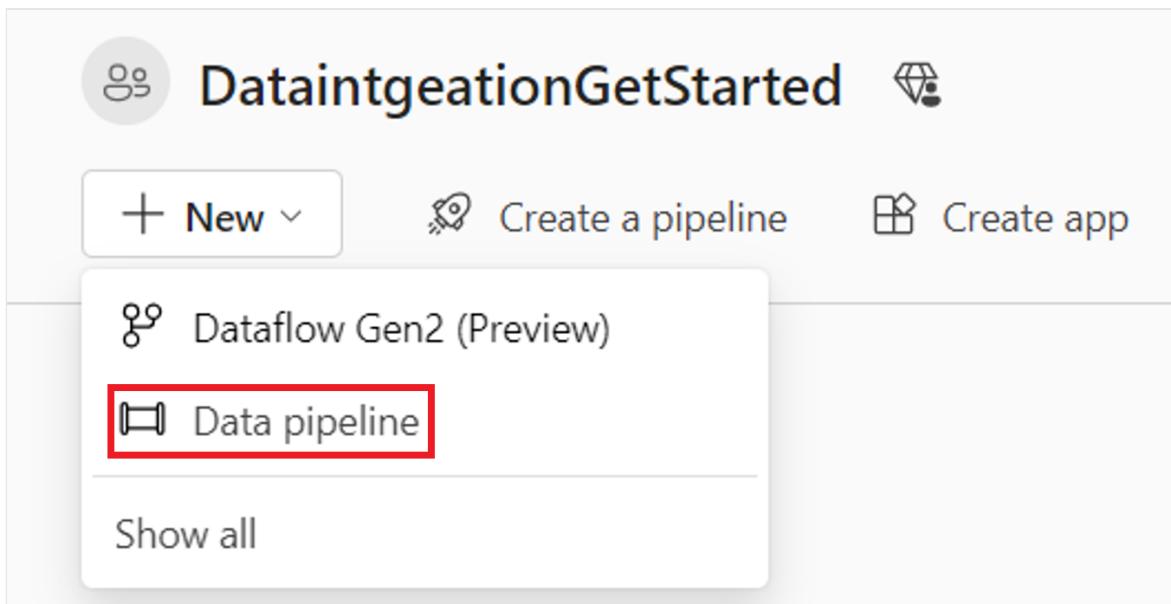
1. Switch to **Data factory** on the app.powerbi.com page.



2. Create a new workspace for this demo.



3. Select **New**, and then select **Data Pipeline**.

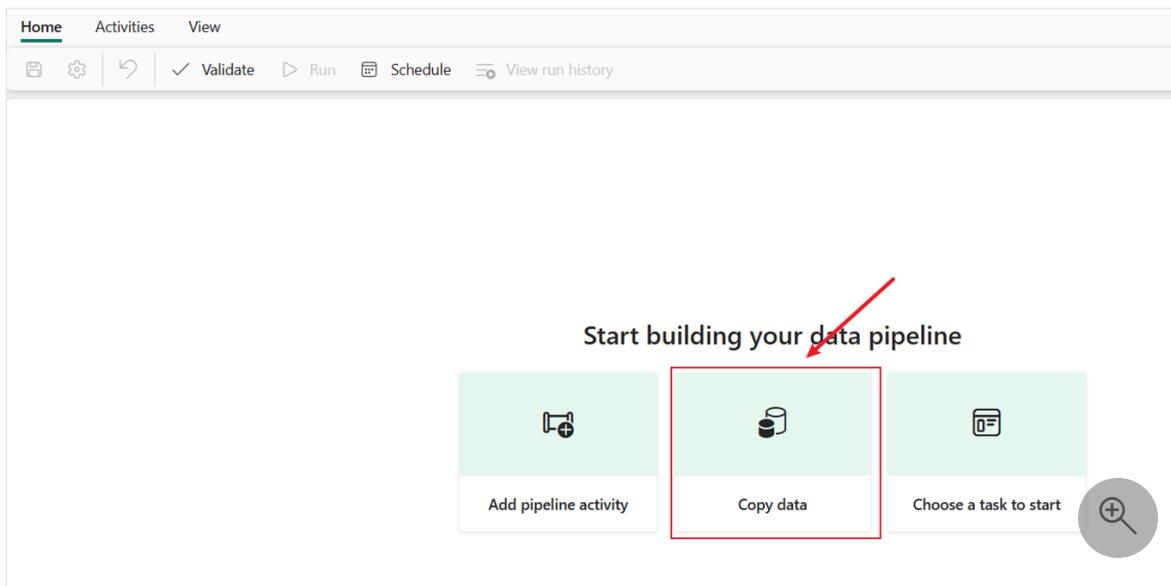


Copy data using the Copy Assistant

In this session, you'll start to build a data pipeline by using the following steps. These steps copy a CSV file from an input folder of an Azure Blob Storage to a Lakehouse destination using the copy assistant.

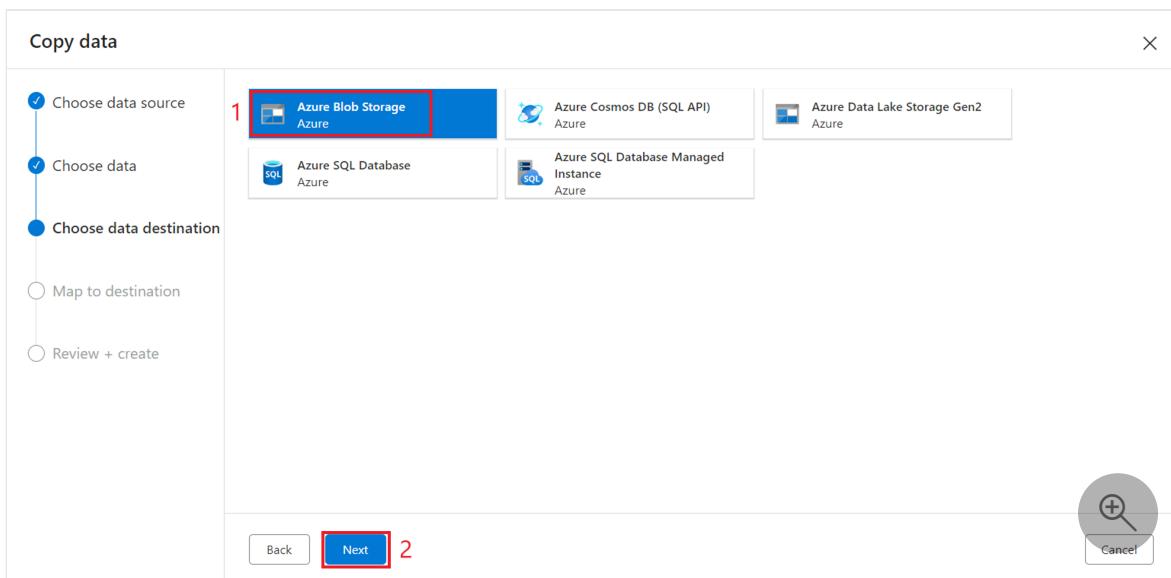
Step 1: Start with copy assistant

1. Select **Copy data** on the canvas to open the **copy assistant** tool to get started. Or Select **Use copy assistant** from the **Copy data** drop down list under the **Activities** tab on the ribbon.

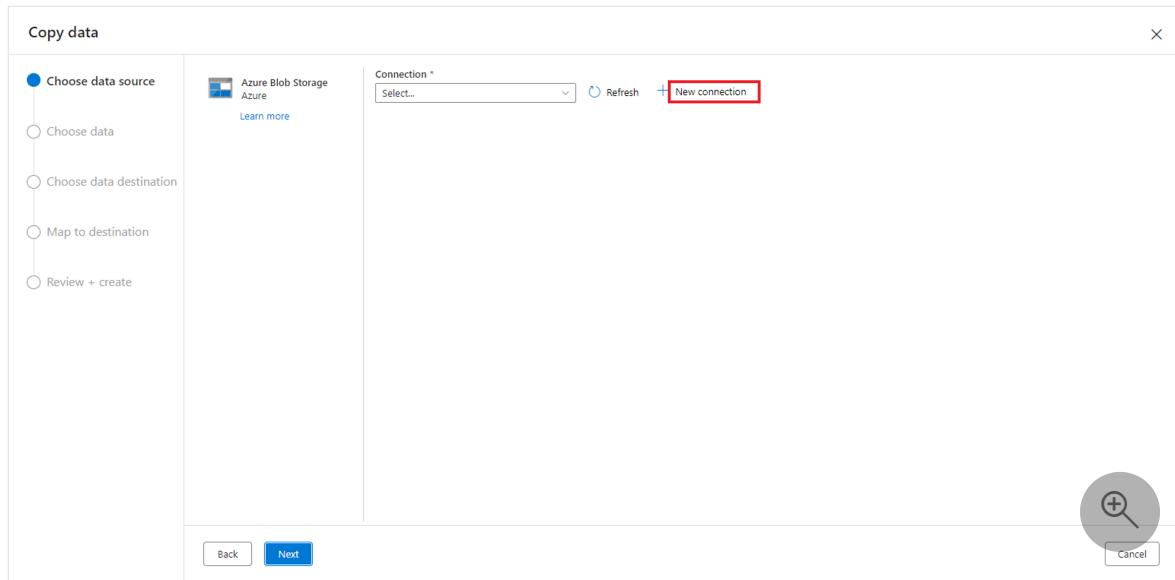


Step 2: Configure your source

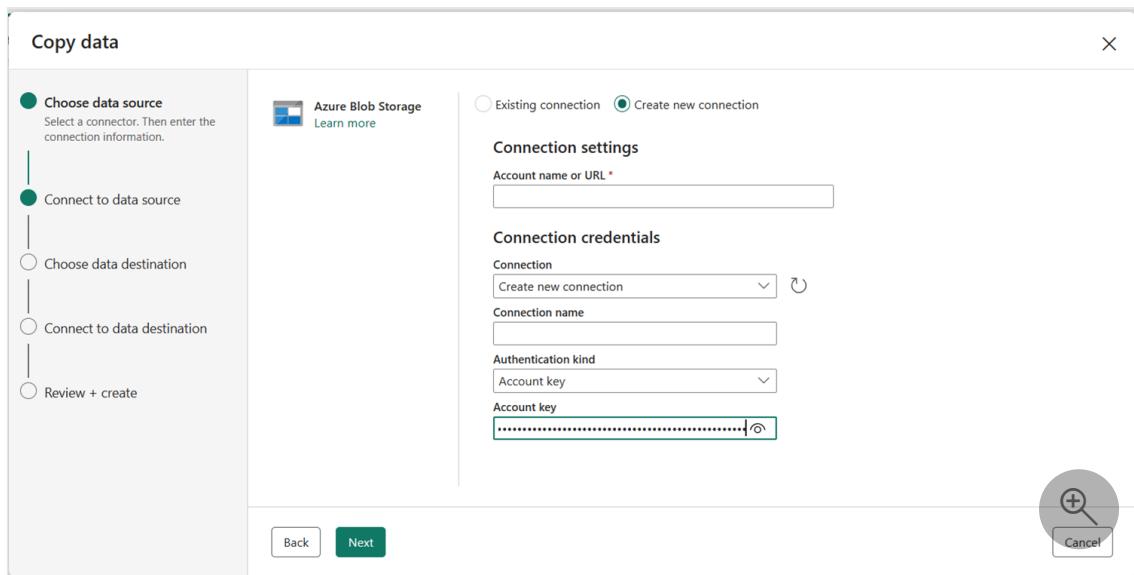
1. Select **Azure Blob Storage**, and then select **Next**.



2. Create a connection to your data source by selecting **New connection**.



- a. After selecting **Create new connection**, you only need to fill in **Account name or URL**, and **Authentication kind**. If you input **Account name or URL** using your Azure Blob Storage account name, the connection will be auto filled. In this demo, we will choose **Account key** but you can choose other Authentication kind regarding your preference. After selecting **Sign in**, you only need to log in to one account that having this blob storage permission.



- b. Once your connection is created successfully, you only need to select **Next** to **Connect to data source**.
3. Choose the file moviesDB2.csv in the source configuration to preview, and then select **Next**.

Copy data

Choose data source

Choose data

Choose data destination

Map to destination

Review + create

blobwyjfp3l6x2lg
input
moviesDB2.csv

Start time (UTC)
8/9/22 00:00:00

Filter by last modified

End time (UTC)

moviesDB2.csv

	abc Prop_0	abc Prop_1	abc Prop_2	abc Prop_3	abc Prop_4
1	movies	Title	genresgenre	YEAR	Rating
2	108583	Fawlty Towers (1975)	Comedy	-1980	1
3	32898	Trip to the Moon, A (Voyage dans la lune, Le)	Action Adventure Fantasy Sci-Fi	1902	7
4	7065	Birth of a Nation, The	Drama War	1915	6
5	7243	Intolerance: Love's Struggle Throughout the Ages	Drama	1915	4
6	62383	20,000 Leagues Under	Action Adventure Sci-Fi	1915	9

Back Next

Cancel

Step 3: Configure your destination

1. Select Lakehouse and then Next.

Copy data

Choose data source

Connect to data source

Choose data destination

Define the data store as destination.

Connect to data destination

Review + create

Azure Blob Storage
Azure

Azure Cosmos DB for NoSQL
Azure

Azure Data Explorer (Kusto)
Azure

Azure Data Lake Storage Gen2
Azure

Azure Database for PostgreSQL
Azure

Azure SQL Database
Azure

Azure Synapse Analytics
Azure

Azure Table Storage
Azure

Data Warehouse
Workspace

Lakehouse
Workspace

REST
Generic protocol, Services and apps

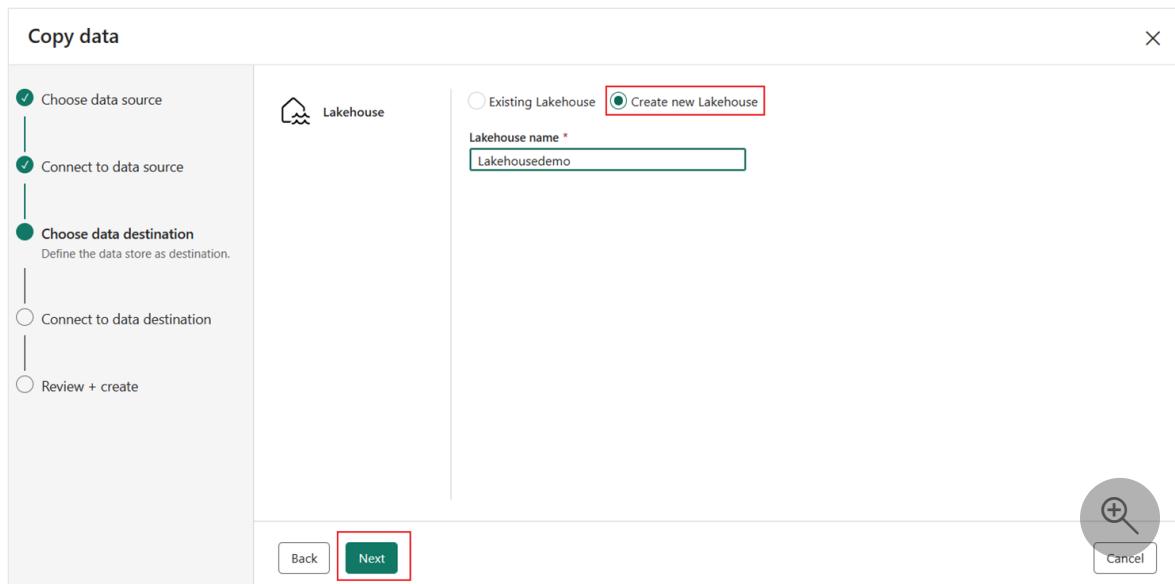
All categories Database Generic protocol Services and apps Workspace Azure

Search

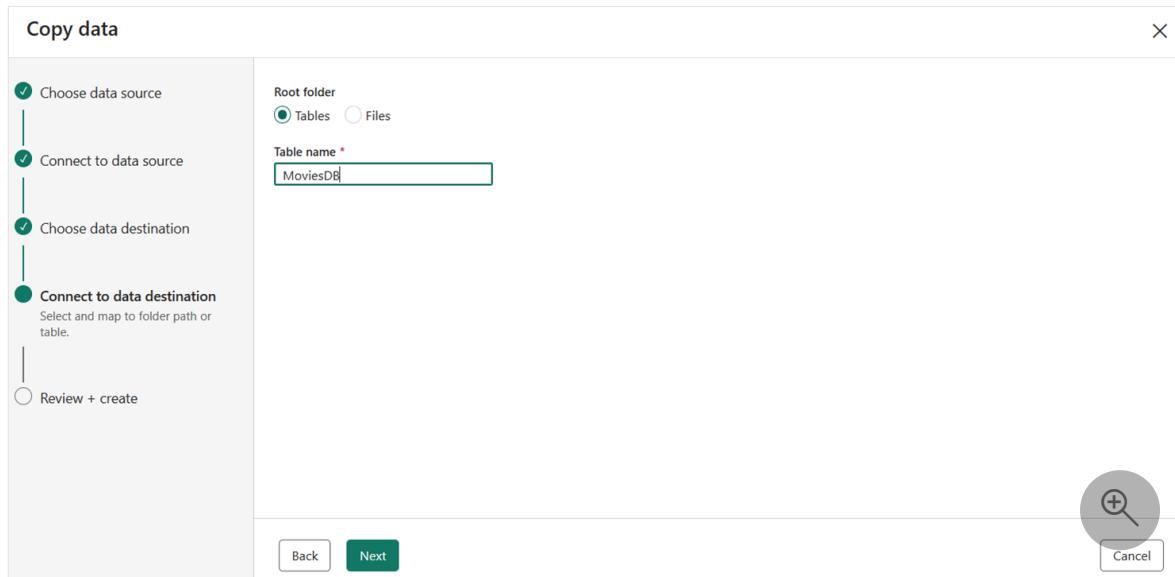
Back Next

Cancel

2. Create a new Lakehouse and input the Lakehouse name. Then select Next.

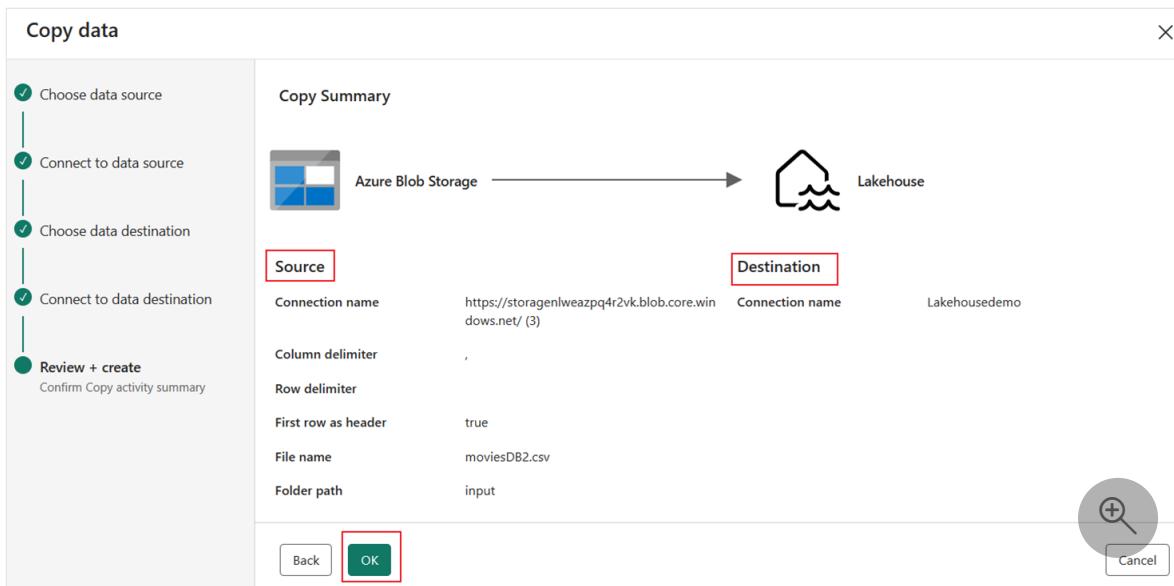


3. Configure and map your source data to your destination; then select **Next** to finish your destination configurations.



Step 4: Review and create your copy activity

1. Review your copy activity settings in the previous steps and select OK to finish. Or you can go back to the previous steps to edit your settings if needed in the tool.



- Once finished, the copy activity will then be added to your data pipeline canvas. All settings including advanced settings to this copy activity are available under the tabs below when it's selected.

Run and schedule your data pipeline

- Switch to **Home** tab and select **Run**. Then select **Save and Run**.

The screenshot shows the Azure Data Factory pipeline editor interface. At the top, there's a navigation bar with 'Home' (highlighted with a red box), 'Activities', 'View', and other buttons like 'Validate' and 'Run'. The 'Run' button is also highlighted with a red box and has the number '2' above it, indicating two pending runs. Below the navigation is a toolbar with icons for file operations. The main area displays a 'Copy data' activity named 'Copy_z31'. The 'Source' tab is selected. Under 'Data store type', 'External' is chosen. The 'Connection' field shows 'https://storagenlweazpq4r2vk.blob.c...'. The 'File path' field shows 'datafactory/input/moviesDB2.csv'. There are buttons for 'Browse', 'Preview data', and 'File settings'. A small circular icon with a magnifying glass is also present.

This screenshot is similar to the one above, showing the pipeline editor with the 'Source' tab selected for the 'Copy_z31' activity. However, a modal dialog box titled 'Save and run?' is overlaid on the interface. The dialog contains the message: 'You have unsaved changes. If you continue, the pipeline will be saved and run.' It features a green 'Save and run' button (highlighted with a red box) and a 'Cancel' button.

2. Select the **Details** button to monitor progress and check the results of the run.

The screenshot shows the Azure Data Factory pipeline history page. The top navigation bar includes 'Home', 'Activities', 'View', and 'Run history'. Below is a table under the 'Output' tab showing the results of a pipeline run. The table has columns for 'Name', 'Type', 'Run start', 'Duration', and 'Status'. One row shows 'Copy_z31' as a 'Copy data' task that started at 2023-09-11T10:00:14Z and took 00:00:14. The status is 'Succeeded' with a green checkmark icon. A circular icon with a magnifying glass is also present. At the bottom of the table, there's a 'Details' button for the 'Copy_z31' row, which is highlighted with a red box.

3. The **Copy data details** dialog displays results of the run including status, volume of data read and written, start and stop times, and duration.

Copy data details

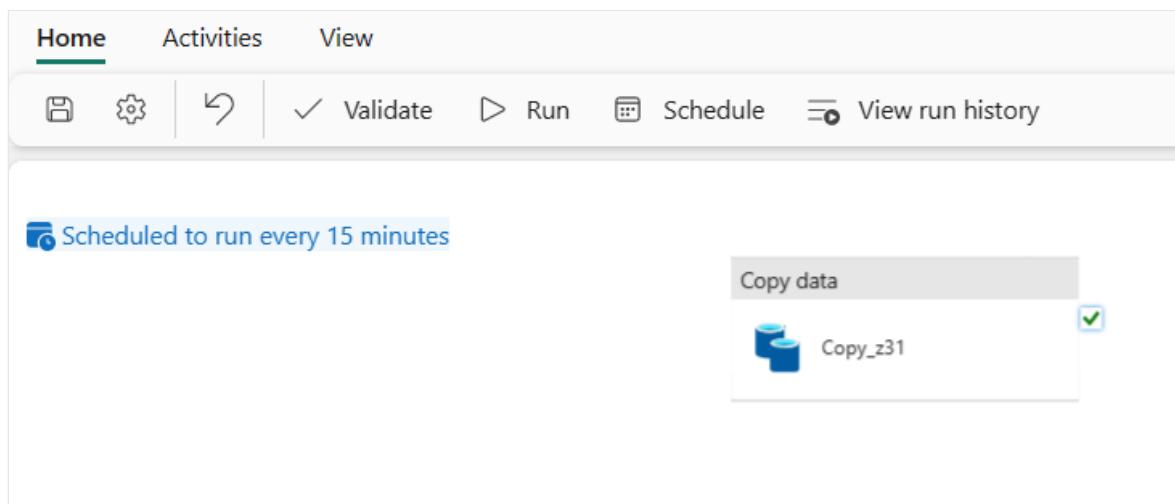
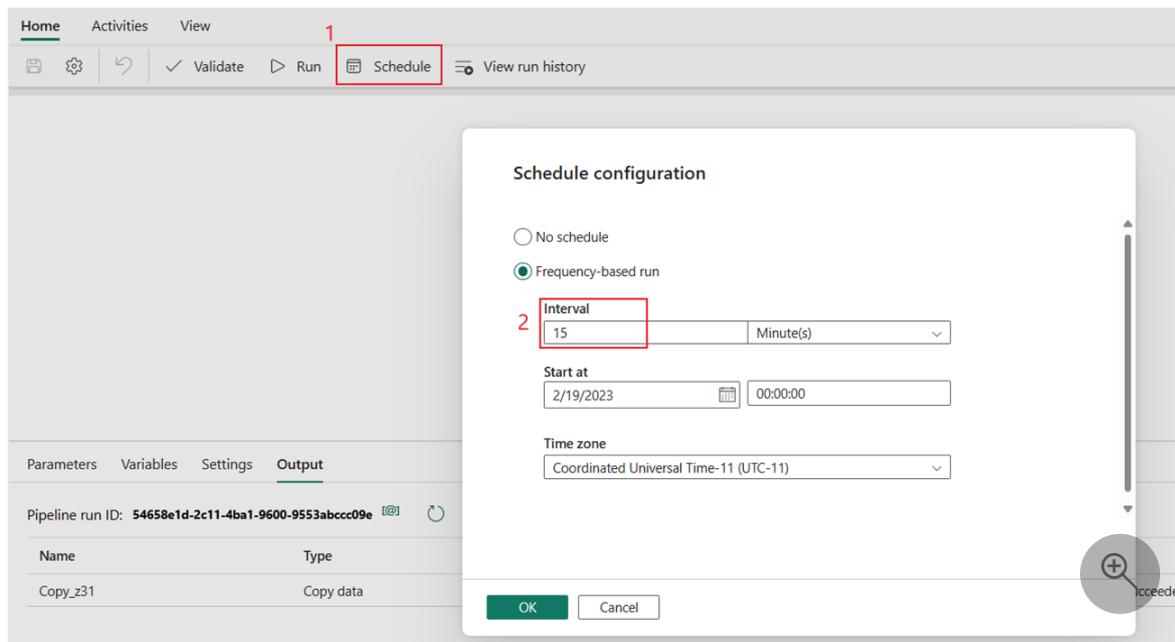
Source	Destination
 Azure Blob Storage	 Lakehouse
Data read: ⓘ 450.221 KB	Data written: ⓘ 154.466 KB
Files read: ⓘ 1	Files written: ⓘ 1
Rows read: 9,125	Rows written: ⓘ 9,125

Status  Succeeded

Start time

Pipeline run activity ID	e5d8fddb-433c-4878-96d8-38c7ffbb9ae0
Throughput	75.037 KB/s
Total duration	00:00:11
▼ Duration breakdown	
Start time	2/19/2023, 2:17:33 PM
Used parallel copies ⓘ	1
Queue	

4. You can also schedule the pipeline to run with a specific frequency as required.
Below is the sample to schedule the pipeline to run every 15 minutes.



Next steps

The pipeline in this sample shows you how to copy data from Azure Blob Storage to Lakehouse. You learned how to:

- ✓ Create a data pipeline.
- ✓ Copy data with the Copy Assistant.
- ✓ Run and schedule your data pipeline.

Next, advance to learn more about monitoring your pipeline runs.

[How to monitor pipeline runs in Microsoft Fabric](#)

Copy sample data into Lakehouse and transform with a dataflow with Data Factory in Microsoft Fabric

Article • 05/23/2023

In this tutorial, we provide end-to-end steps to a common scenario that uses the pipeline to load source data into Lakehouse at high performance copy and then transform the data by dataflow to make users can easily load and transform data.

ⓘ Important

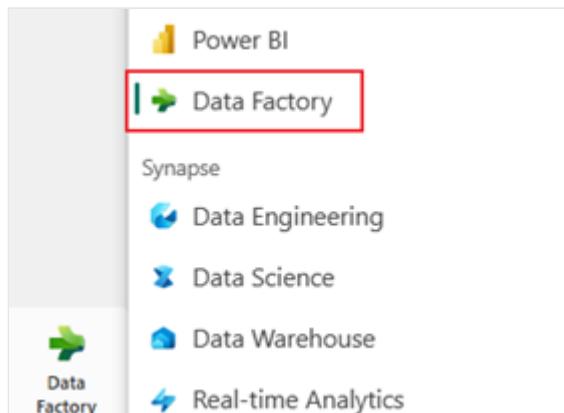
Microsoft Fabric is currently in PREVIEW. This information relates to a prerelease product that may be substantially modified before it's released. Microsoft makes no warranties, expressed or implied, with respect to the information provided here. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Prerequisites

A Microsoft Fabric enabled workspace. If you don't already have one, refer to the article [Create a workspace](#).

Create a data pipeline

1. Switch to the **Data Factory** experience.



2. Select **New** and then **Data pipeline**, and then input a name for your pipeline.

The screenshot shows the Microsoft Data Factory interface. On the left, there's a sidebar with icons for Home, Create, Browse, Workspaces, and Data Factory. Under Workspaces, 'SampleDFWorkspace' is selected, and under it, 'SampleDFPipeline' is also selected. The main area has a title 'SampleDFWorkspace' with a gear icon. Below it is a toolbar with '+ New', 'Upload', and a three-dot menu, followed by a search bar 'Filter by keyword'. A red box highlights the 'Data pipeline (Preview)' option in a list of items. To the right of the list is a descriptive text: 'Ingest data at scale and schedule data workflows.' and a 'Type' column. The list includes:

Type
Dataset (default)
SQL endpoint
Lakehouse
Data pipeline
Notebook
Data pipeline
Data pipeline
Dataflow

Below the list are buttons for 'Show all' and 'Quickstart dataflow'.

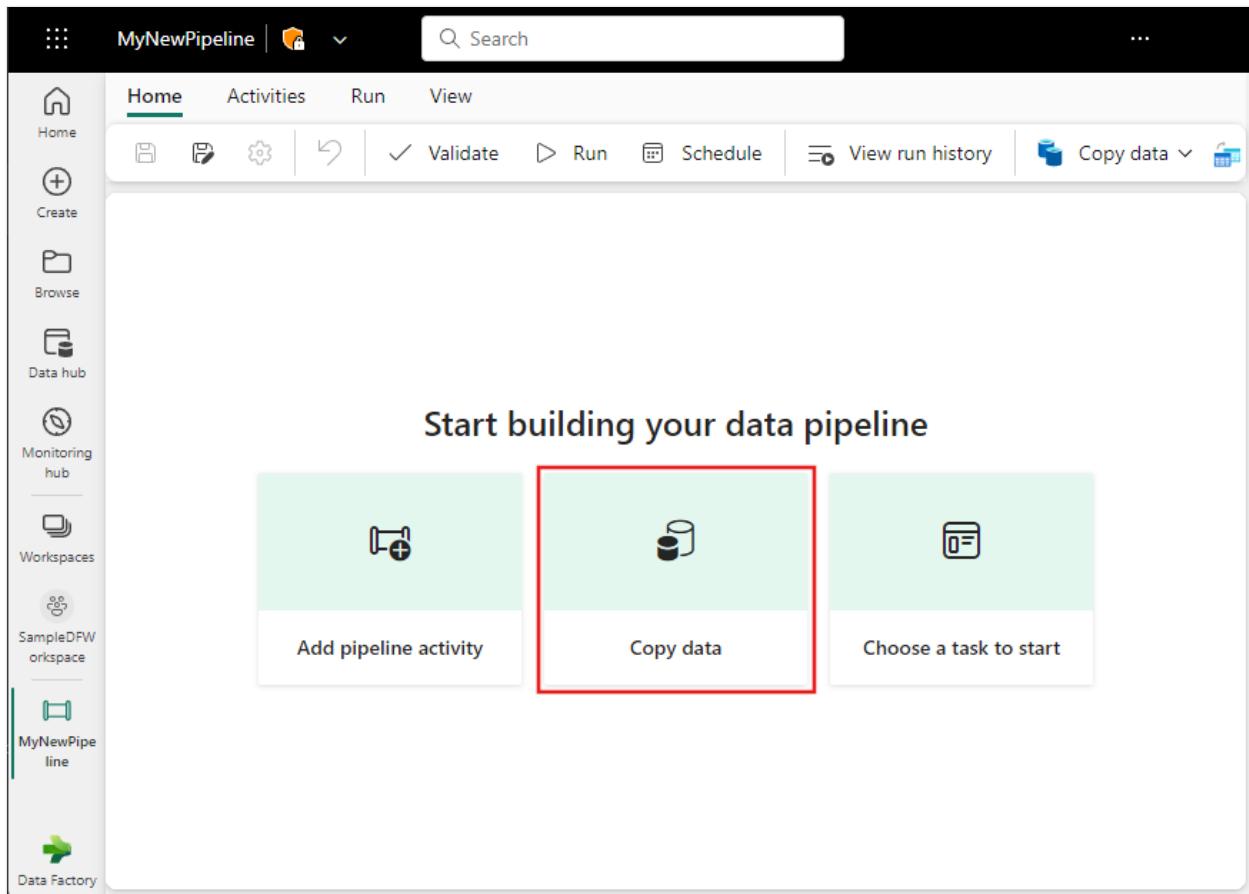
A modal dialog box titled 'New pipeline' is shown. It has a 'Name' field containing '03_Create_First_Pipeline', which is highlighted with a green border. At the bottom are two buttons: a green 'Create' button and a white 'Cancel' button.

Use a pipeline to load sample data into Lakehouse

Use the following steps to load sample data into Lakehouse.

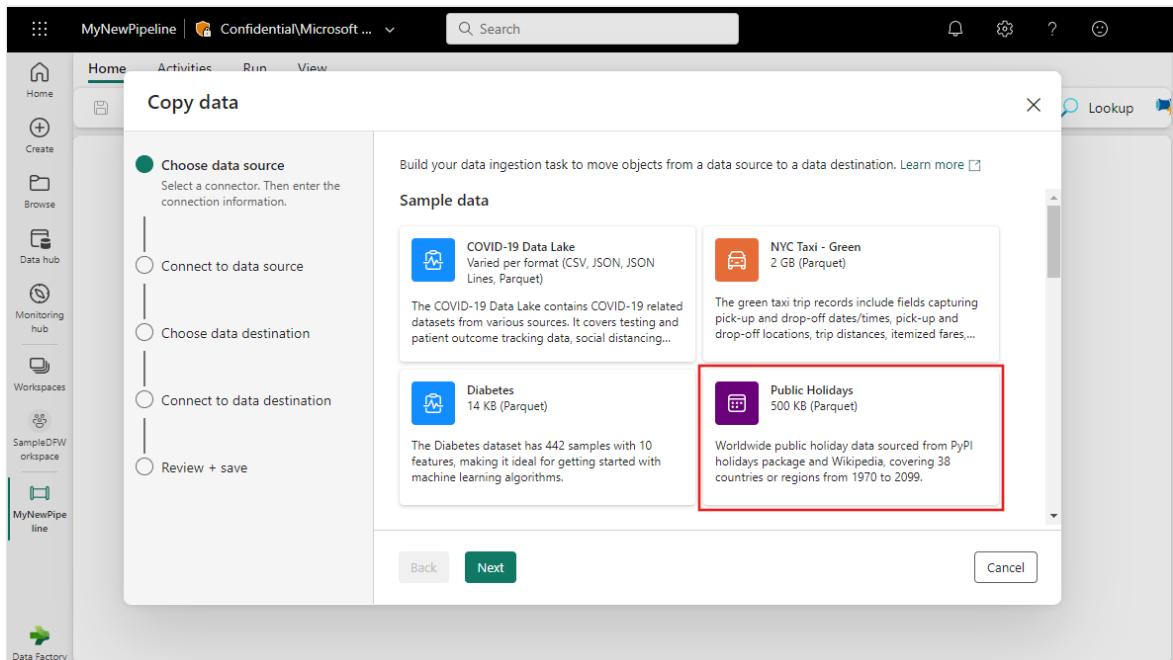
Step 1: Start with the Copy assistant

Select **Copy Data** on the canvas, to open the **Copy assistant** tool to get started.



Step 2: Configure your source

1. Choose the **Public Holidays** from the **Sample data** options for your data source, and then select **Next**.



2. In the Connect to data source section of the **Copy data** assistant, a preview of the sample data is displayed. Select **Next** to move on to the data destination.

The screenshot shows the 'Copy data' assistant in the Azure Data Factory portal. The left sidebar shows 'MyNewPipeline' selected. The main area has a title 'Copy data' and a progress bar with three steps: 'Choose data source' (completed), 'Connect to data source' (in progress), and 'Choose data destination' (not yet started). To the right, a preview window titled 'Preview data: Public Holidays' displays a table with 10 rows of data. At the bottom are 'Back', 'Next' (highlighted in green), and 'Cancel' buttons.

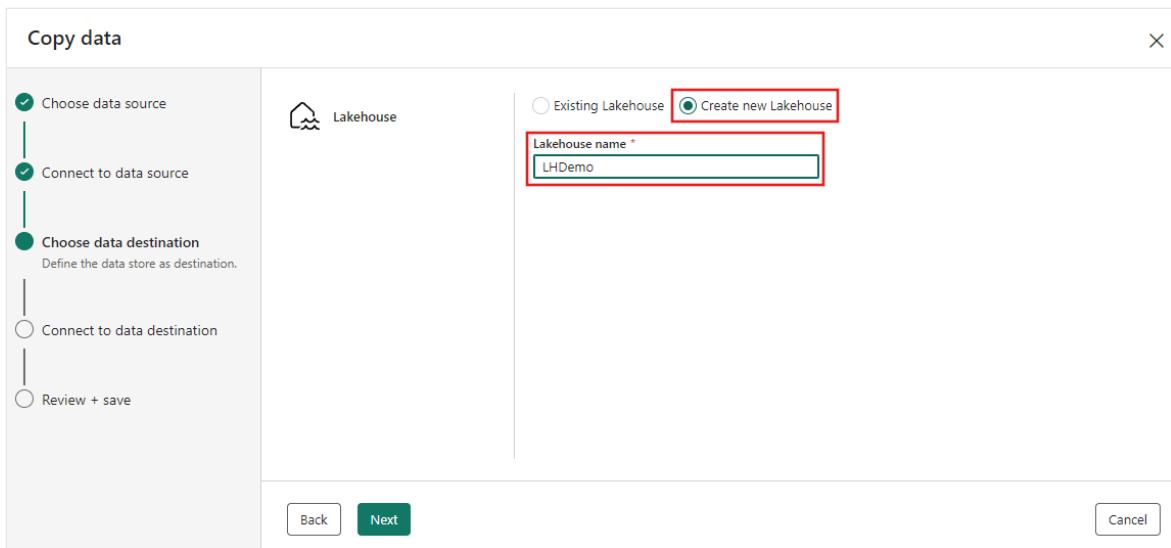
	abc	countryOrRegion	abc_holidayName	abc_normalizeHolidayName	is
1	Argentina		Año Nuevo [New Year's Day]	Año Nuevo [New Year's Day]	
2	Australia		New Year's Day	New Year's Day	
3	Austria		Neujahr	Neujahr	
4	Belgium		Nieuwjaarsdag	Nieuwjaarsdag	
5	Brazil		Ano novo	Ano novo	
6	Canada		New Year's Day	New Year's Day	
7	Colombia		Año Nuevo [New Year's Day]	Año Nuevo [New Year's Day]	
8	Croatia		Nova Godina	Nova Godina	
9	Czech		Nový rok	Nový rok	

Step 3: Configure your destination

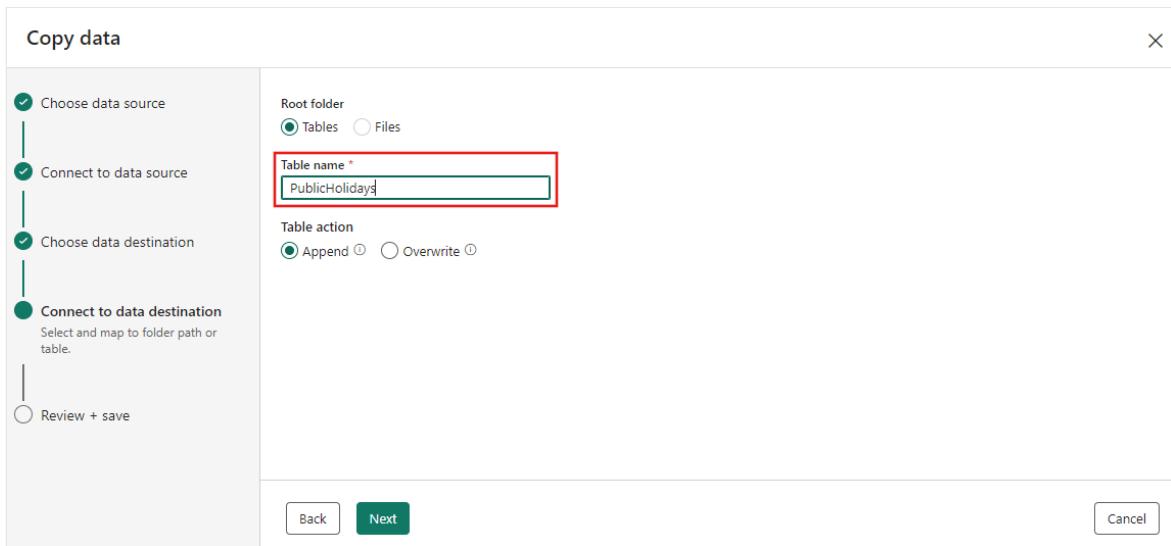
1. Select the **Workspace** tab and choose **Lakehouse**. Then select **Next**.

The screenshot shows the 'Copy data' assistant in the Azure Data Factory portal. The left sidebar shows 'MyNewPipeline' selected. The main area has a title 'Copy data' and a progress bar with three steps: 'Choose data source' (completed), 'Connect to data source' (in progress), and 'Choose data destination' (selected). The 'Choose data destination' step says 'Define the data store as destination.' Below it are 'Connect to data destination' and 'Review + create' options. To the right, a list of destination categories is shown under 'All categories'. The 'Lakehouse' option is highlighted with a red box. Other options include Azure Blob Storage, Azure Cosmos DB for NoSQL, Azure Data Explorer (Kusto), Azure Data Lake Storage Gen1, Azure Data Lake Storage Gen2, Azure Database for PostgreSQL, Azure SQL Database, Azure SQL Database Managed Instance, Azure Synapse Analytics, Azure Table Storage, Data Warehouse, Dataverse (Common Data Service for Apps), REST, Snowflake, and SQL server. At the bottom are 'Back', 'Next' (highlighted in green), and 'Cancel' buttons.

2. Select **Create new Lakehouse** and enter **LHDemo** for the name, then select **Next**.

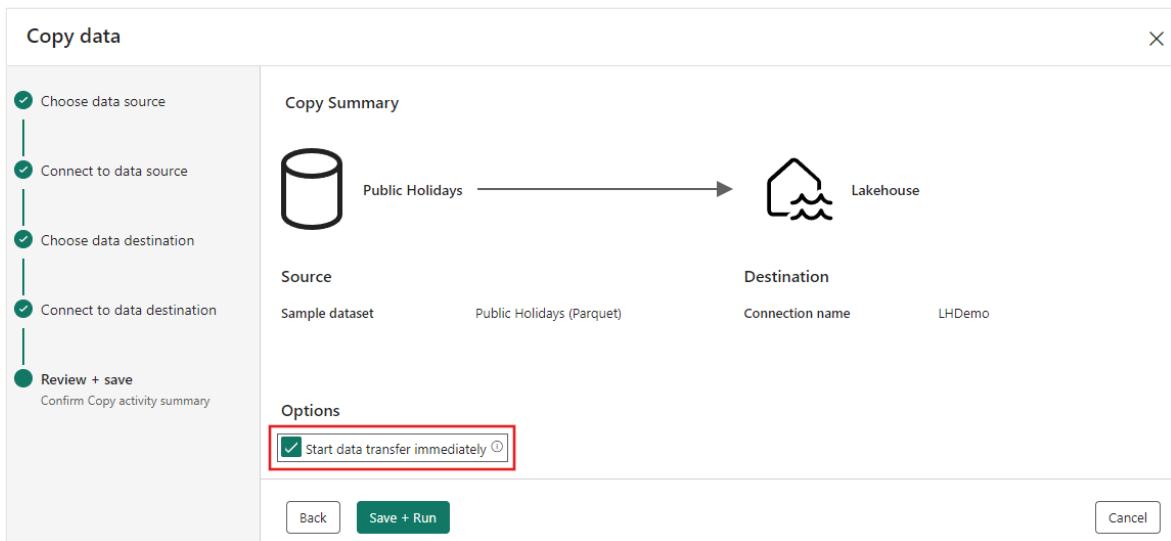


3. Configure and map your source data to the destination Lakehouse table by entering **Table name**, then select **Next** one more time.

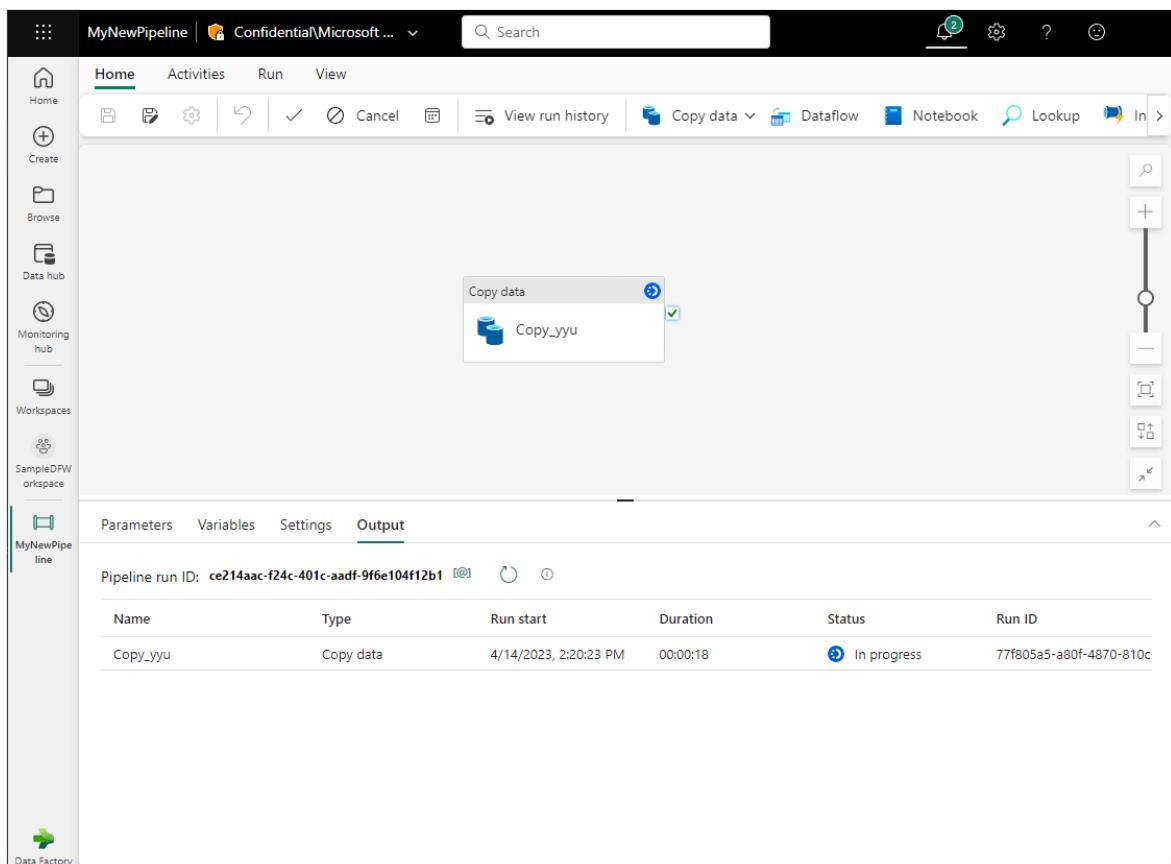


Step 4: Review and create your copy activity

1. Review your copy activity settings in the previous steps and select **Start data transfer immediately**. Then select **Save + Run** to run the new pipeline.



2. Once finished, the copy activity is added to your new data pipeline canvas, and the pipeline automatically runs to load data into Lakehouse.



3. You can monitor the running process and check the results on the **Output** tab below the pipeline canvas. Hover over the name in the output row to see the **Run details** button (an icon of a pair of glasses, highlighted) to view the run details.

Name	Type	Run start	Duration	Status	Run ID
Copy_yyu	Copy data	4/14/2023, 2:20:23 PM	00:00:17	Succeeded	77f805a5-a80f-4870-810c

4. The run details show 69,557 rows were read and written, and various other details about the run, including a breakdown of the duration.

Copy data details

Copy_yyu

Source **Destination**

Azure Blob Storage → Lakehouse

Data read:	663.358 KB	Data written:	298.638 KB
Files read:	1	Files written:	1
Rows read:	69,557	Rows written:	69,557

Status: Succeeded

Start time: 4/14/2023, 2:20:23 PM

Pipeline run activity ID: 77f805a5-a80f-4870-810c-fb3957291fea

Throughput: 60.305 KB/s

Total duration: 00:00:15

Duration breakdown:

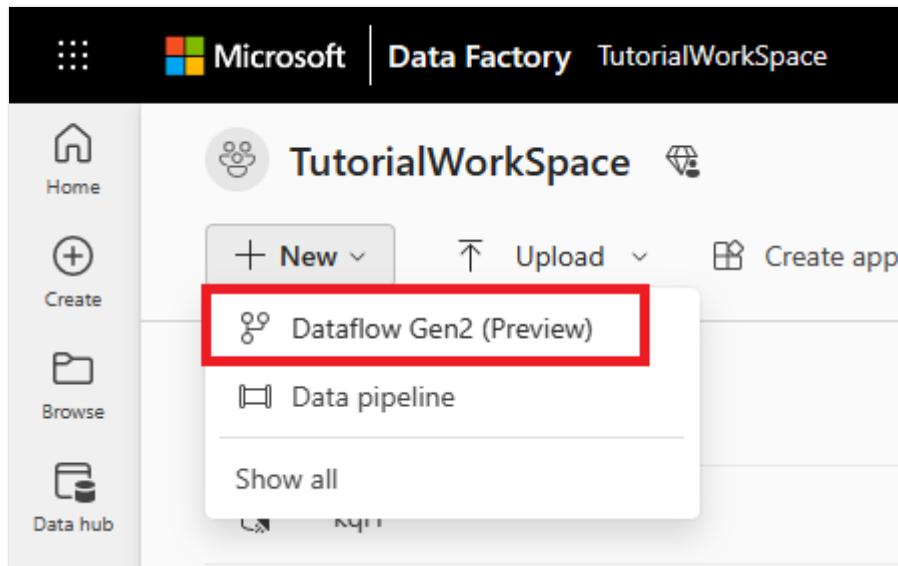
Start time	4/14/2023, 2:20:24 PM
Used parallel copies:	1
Queue	<div style="width: 30%;"></div>
Transfer	<div style="width: 70%;"></div>

Close

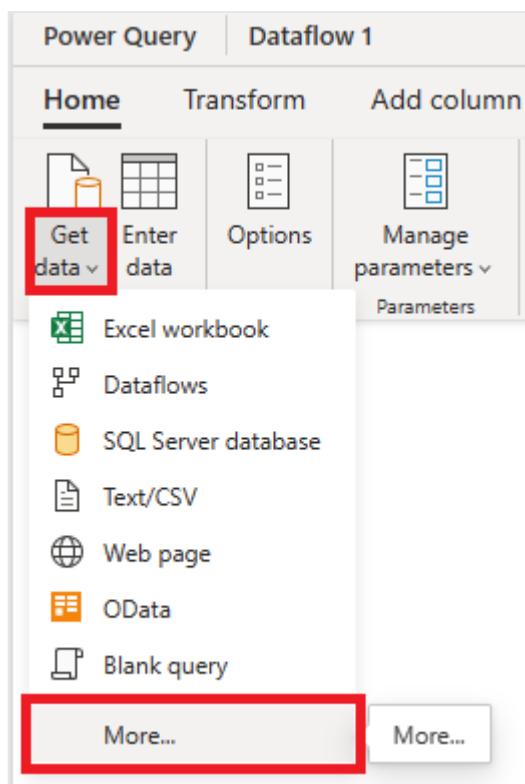
Use a dataflow gen2 to transform data in the Lakehouse

You now have a Lakehouse with sample data loaded. Next, you'll use a dataflow to transform the data. Dataflows are a code-free way to transform data at scale.

1. Select New and then Dataflow Gen2.



2. Click on get data dropdown and select More....



3. Search for Lakehouse and select Lakehouse in Microsoft Fabric.

Choose data source

Select a connector or directly drag a file from your computer.

All categories

File

Database

Power Platform

Azure

...

lakehouse X



Lakehouse in Microsoft Fabric
Other

4. Sign-in and click **Next** to continue.

Connect to data source

Lakehouse in Microsoft Fabric
Other

Connection credentials

Connection: Lakehouse (none)

Connection name: Connection

Data gateway: (none)

Authentication kind: Organizational account

You are currently signed in.
[Switch account](#)

Back Cancel Next

5. Select the table you created in the previous step and click **Create**.

Display options ▾

LHDemo [4]

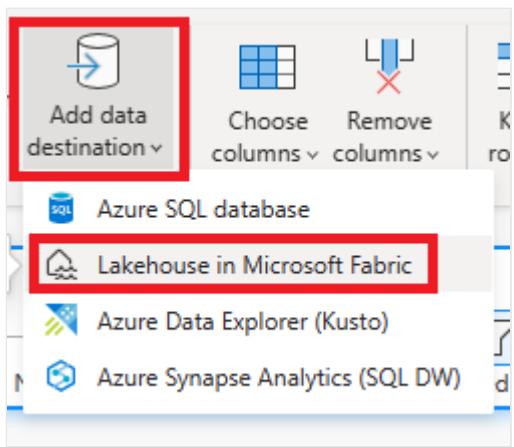
Files

PublicHoliday

6. Review the data preview in the editor.

7. Apply a filter to the dataflow to only include rows where the **Countryorregion** column is equal to **Belgium**.

8. Add a data destination to the query by selecting **Add data destination** and then **Lakehouse in Microsoft Fabric**.



9. Sign-in and click **Next** to continue.

Connect to data source

Lakehouse in Microsoft Fabric Other

Connection credentials

Connection: Lakehouse (none)

Connection name: Connection

Data gateway: (none)

Authentication kind: Organizational account

You are currently signed in.
Switch account

Back Cancel Next

10. Create a new table called **BelgiumPublicHolidays** and click **Next**.

Choose destination target

New table Existing table

Search

Display options

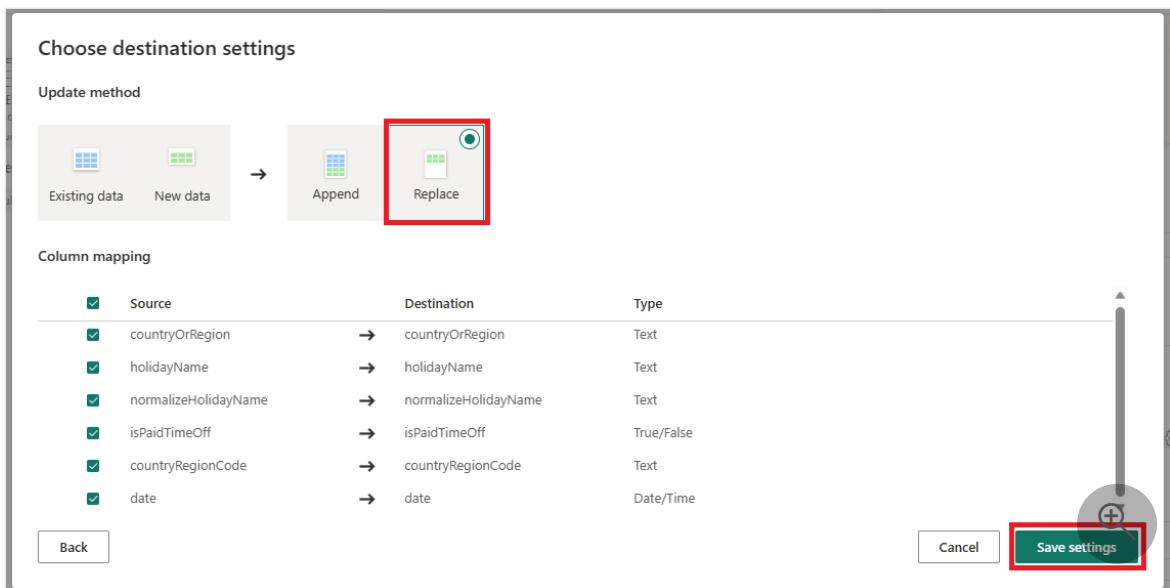
TutorialWorkSpace [3]
Lakehouse_For_Dataflows
LHDemo
LHDemo1 [1]
PublicHoliday

A new table will be created in database LHDemo1

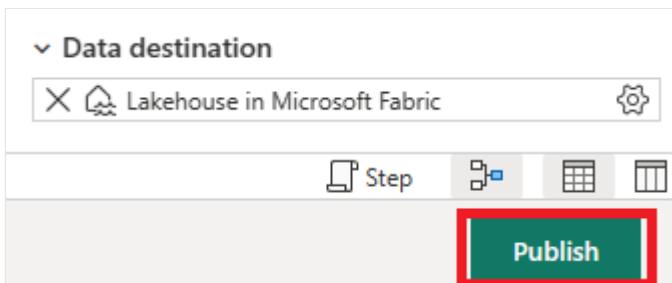
Table name * BelgiumPublicHolidays

Next

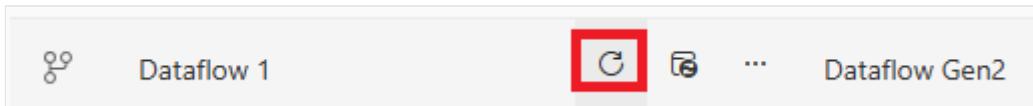
11. Review the settings and click **Save settings**.



12. Publish the dataflow by clicking **Publish**.



13. After the dataflow is published, click **Refresh now** to run the dataflow.



After the refresh is complete, you can view the data in the Lakehouse table. You can also use this data now to create reports, dashboards, and more.

Next steps

This sample shows you how to copy sample data to Lakehouse and transform the data with a dataflow using Data Factory in Microsoft Fabric. You learned how to:

- ✓ Create a data pipeline.
- ✓ Use the pipeline to load sample data into Lakehouse.
- ✓ Use dataflow to transform data in the Lakehouse.

Next, advance to learn more about monitoring your pipeline runs.

[How to monitor pipeline runs in Microsoft Fabric](#)

Preprocess data with a stored procedure before loading into Lakehouse

Article • 05/23/2023

In this tutorial, we show you how to use a pipeline Script activity to run a stored procedure to create a table and preprocess the data in a Synapse Data Warehouse. After that, we load the preprocessed table into Lakehouse.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Prerequisites

- A Microsoft Fabric enabled workspace. If you don't already have one, refer to the article [Create a workspace](#).
- Prepare a stored procedure in your Azure Synapse Data Warehouse. Create the following stored procedure in advance:

SQL

```
CREATE PROCEDURE spM_add_names
AS
    --Create initial table
    IF EXISTS (SELECT * FROM sys.objects
    WHERE object_id = OBJECT_ID(N'[dbo].[names]') AND TYPE IN (N'U'))
    BEGIN
        DROP TABLE names
    END;

    CREATE TABLE names
    (id INT,fullname VARCHAR(50));

    --Populate data
    INSERT INTO names VALUES (1,'John Smith');
    INSERT INTO names VALUES (2,'James Dean');

    --Alter table for new columns
    ALTER TABLE names
    ADD first_name VARCHAR(50) NULL;
```

```

ALTER TABLE names
ADD last_name VARCHAR(50) NULL;

--Update table
UPDATE names
SET first_name = SUBSTRING(fullname, 1, CHARINDEX(' ', fullname)-1);

UPDATE names
SET last_name = SUBSTRING(fullname, CHARINDEX(' ', fullname)+1,
LEN(fullname)-CHARINDEX(' ', fullname));

--View Result
SELECT * FROM names;

```

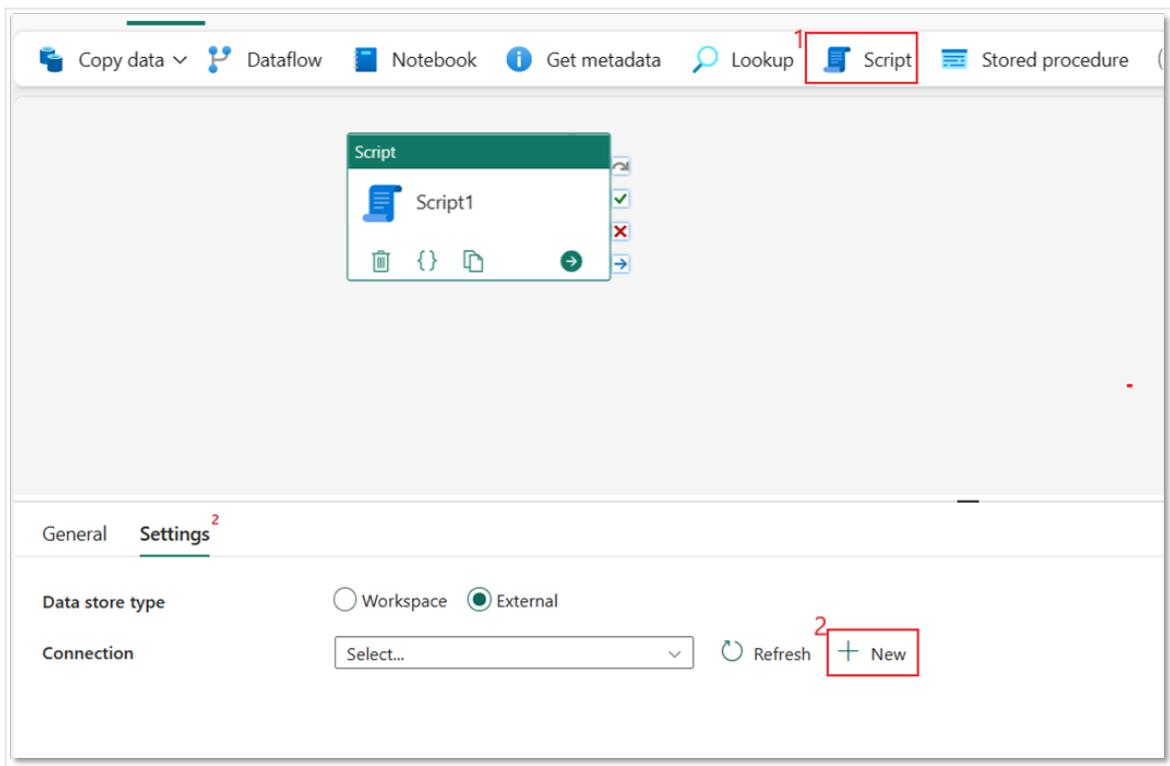
The screenshot shows the SSMS interface with the query window titled 'EXEC spM_add_names'. Below the title bar, there's a zoom level indicator at '100 %'. The main area displays two tabs: 'Results' and 'Messages'. The 'Results' tab is selected and contains a table with four columns: 'id', 'fullname', 'first_name', and 'last_name'. The data shows two rows: row 1 with id 1, fullname 'John Smith', first_name 'John', and last_name 'Smith'; and row 2 with id 2, fullname 'James Dean', first_name 'James', and last_name 'Dean'. The 'Messages' tab is also visible but contains no text.

	id	fullname	first_name	last_name
1	1	John Smith	John	Smith
2	2	James Dean	James	Dean

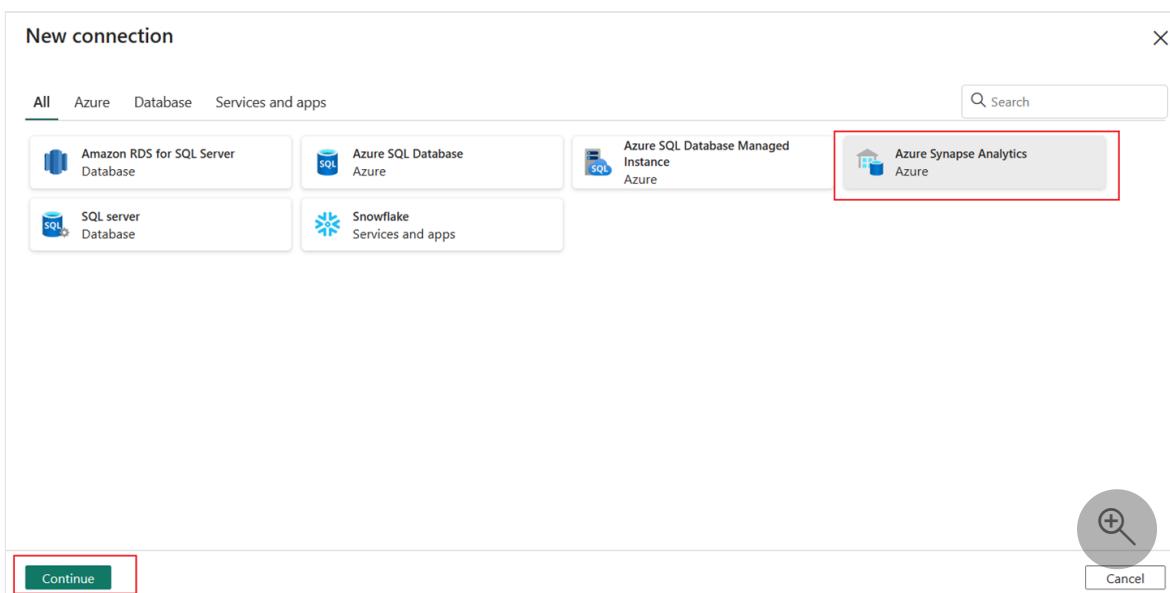
Create a pipeline Script activity to run the stored procedure

In this section, we use a Script activity to run the stored procedure created in the prerequisites.

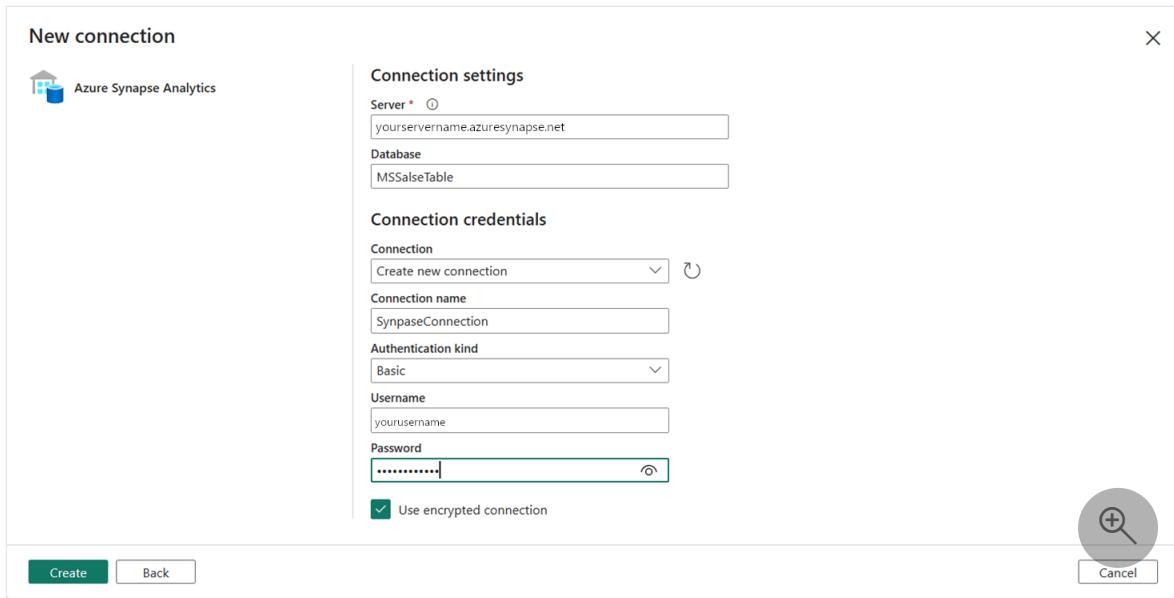
1. Choose Script activity and then select **New** to connect to your Azure Synapse Data Warehouse.



2. Select Azure Synapse Analytics and then **Continue**.



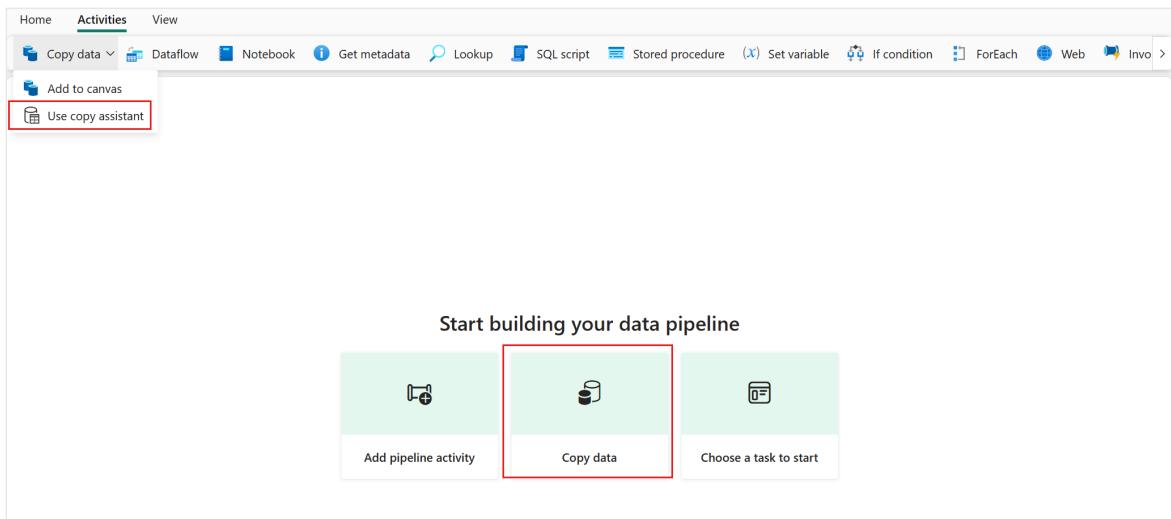
3. Provide your **Server**, **Database**, and **Username** and **Password** fields for **Basic authentication**, and enter **SynapseConnection** for the **Connection name**. Then select **Create** to create the new connection.



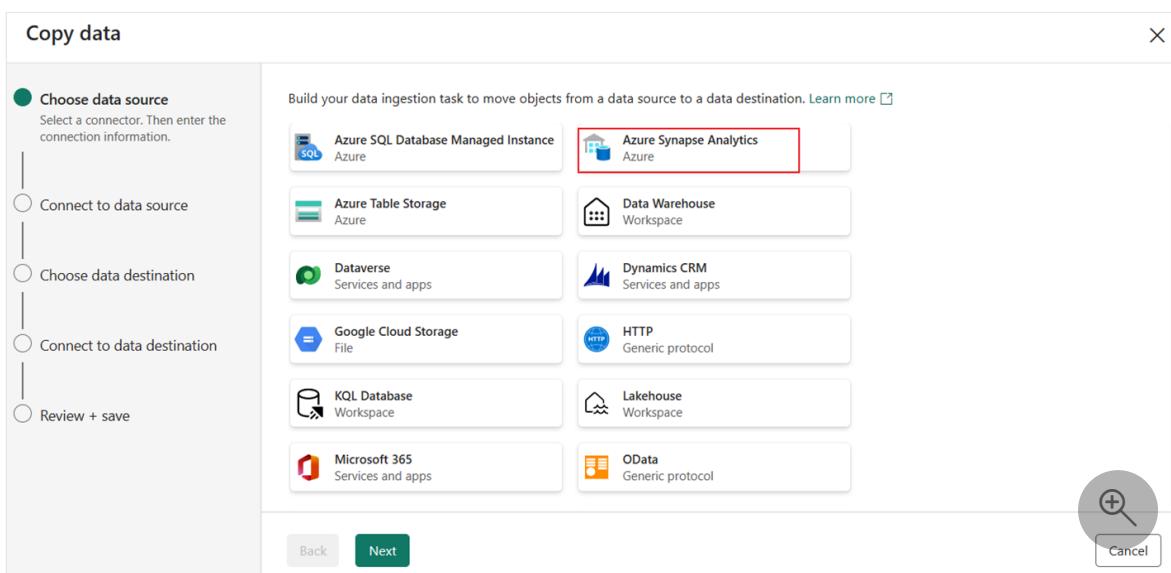
4. Input `EXEC spM_add_names` to run the stored procedure. It creates a new table `dbo.name` and preprocess the data with a simple transformation to change the `fullname` field into two fields, `first_name` and `last_name`.

Use a pipeline activity to load preprocessed table data into Lakehouse

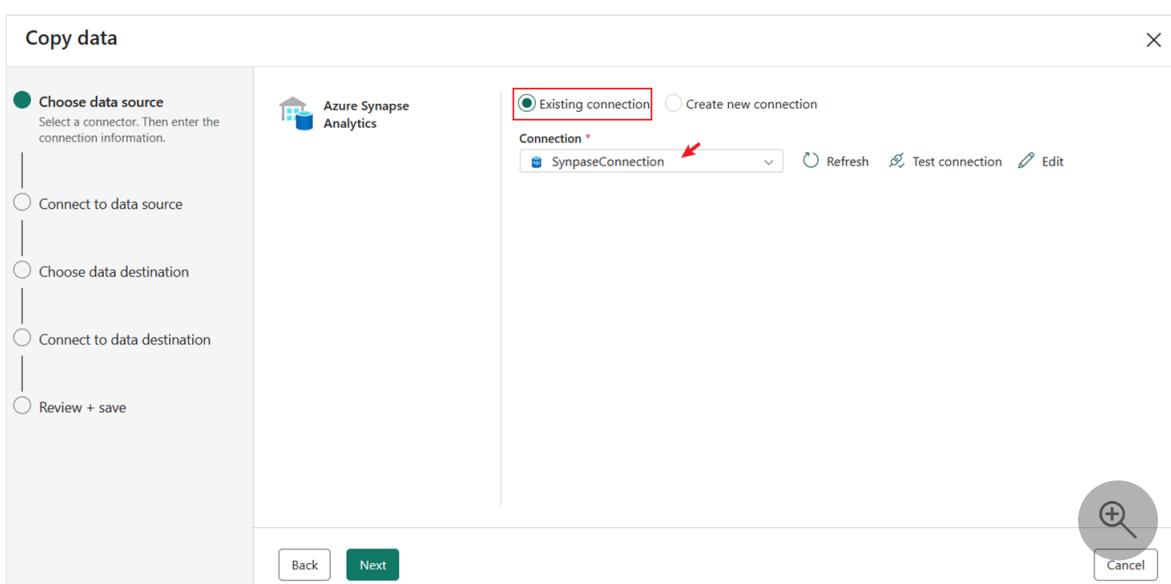
1. Select Copy data and then select Use copy assistant.



2. Select **Azure Synapse Analytics** for the data source, and then select **Next**.



3. Choose the existing connection **SynapseConnection** that you created previously.



4. Choose the table **dbo.names** that was created and preprocessed by the stored procedure. Then select **Next**.

Copy data

Choose data source

Connect to data source
Select, preview, and choose the data.

Choose data destination

Connect to data destination

Review + save

Select a table

Existing tables

Use query

Search Search

MSSalseTable

Select all

dbo.kafka...

dbo.names

dbo.PreTable

Preview data: dbo.names

123 id	abc fullname	abc first_name	abc last_name
1	John Smith	John	Smith
2	James Dean	James	Dean

Back Next

5. Select **Lakehouse** under the **Workspace** tab as the destination, and then select **Next** again.

Copy data

Choose data source

Connect to data source

Choose data destination
Define the data store as destination.

Connect to data destination

Review + save

Data destinations

All categories

Workspace

Azure

Database

Generic protocol

Services and apps

Search Search

Data Warehouse Workspace

KQL Database Workspace

Lakehouse Workspace

Back Next Cancel

6. Choose an existing or create a new Lakehouse, then select **Next**.

Copy data

Choose data source

Connect to data source

Choose data destination
Define the data store as destination.

Connect to data destination

Review + save

Lakehouse Learn more

Existing Lakehouse

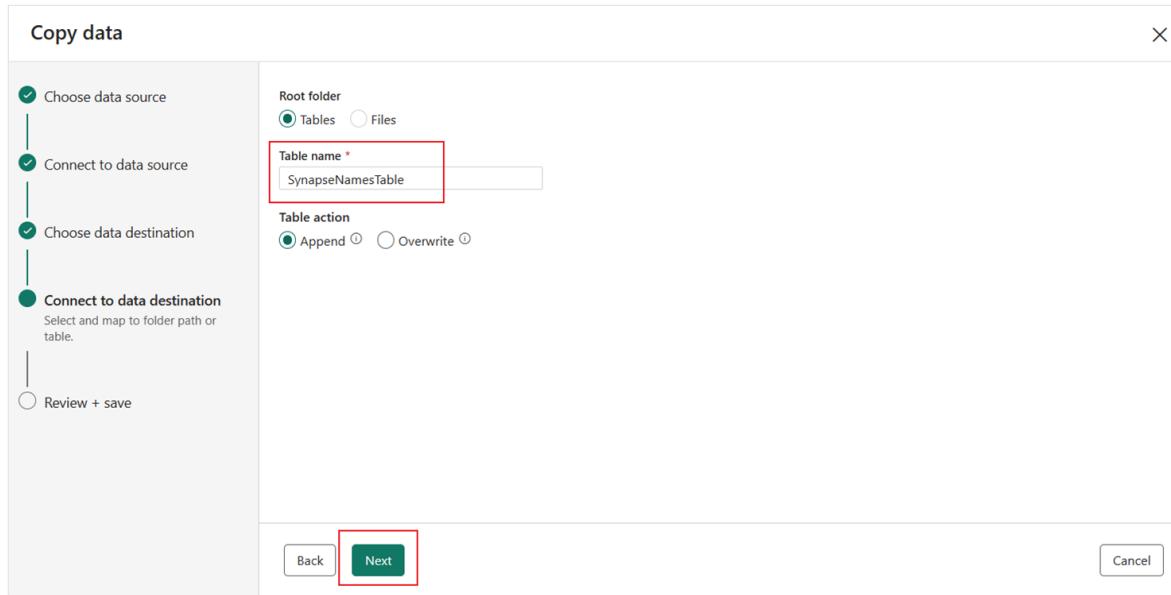
Create new Lakehouse

Lakehouse * LHDemo

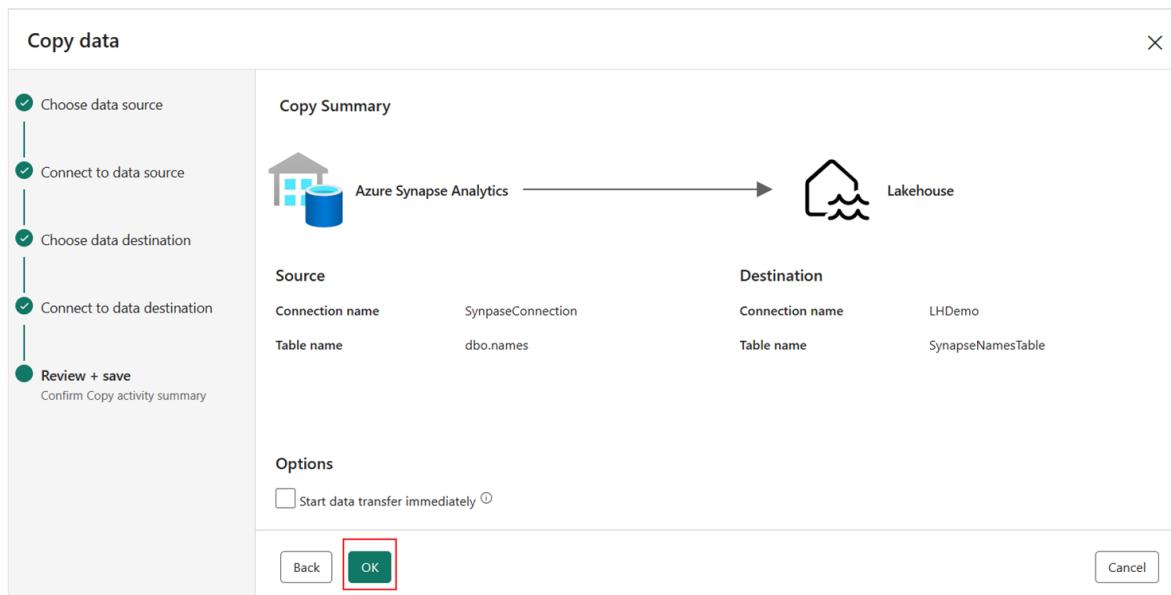
Refresh

Back Next Cancel

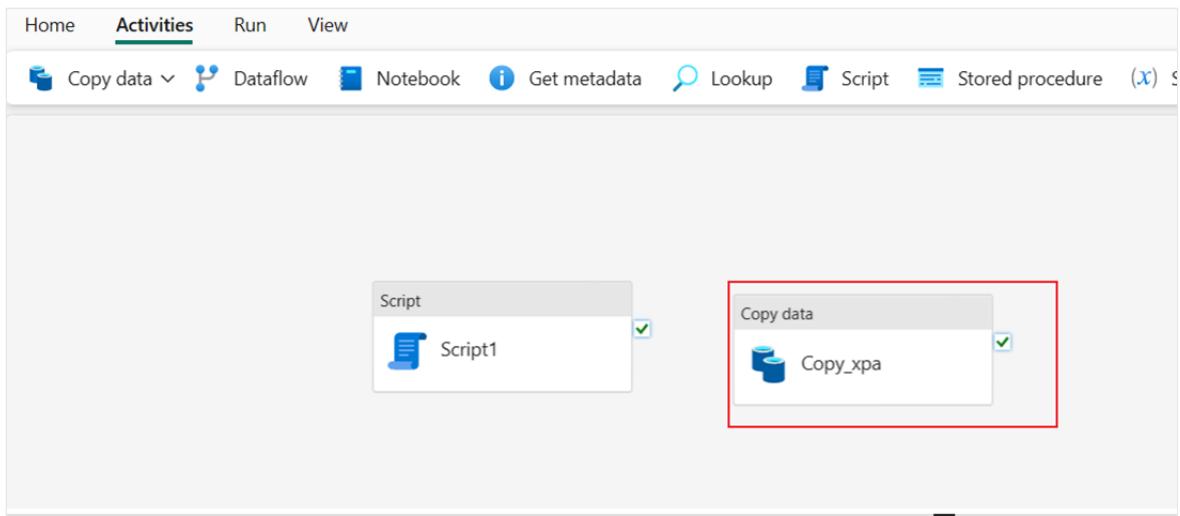
7. Input a destination table name for the data to be copied into for the Lakehouse destination and select **Next**.



8. Review the summary on the final page of the Copy assistant and then select **OK**.

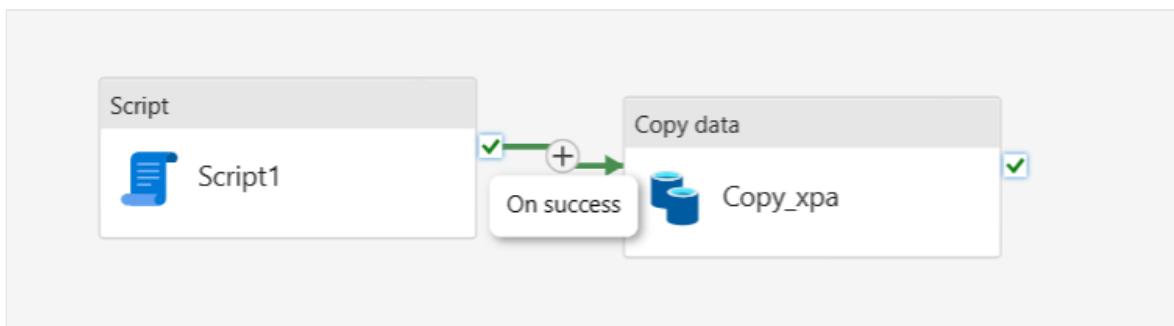


9. After you select **OK**, the new Copy activity will be added onto the pipeline canvas.

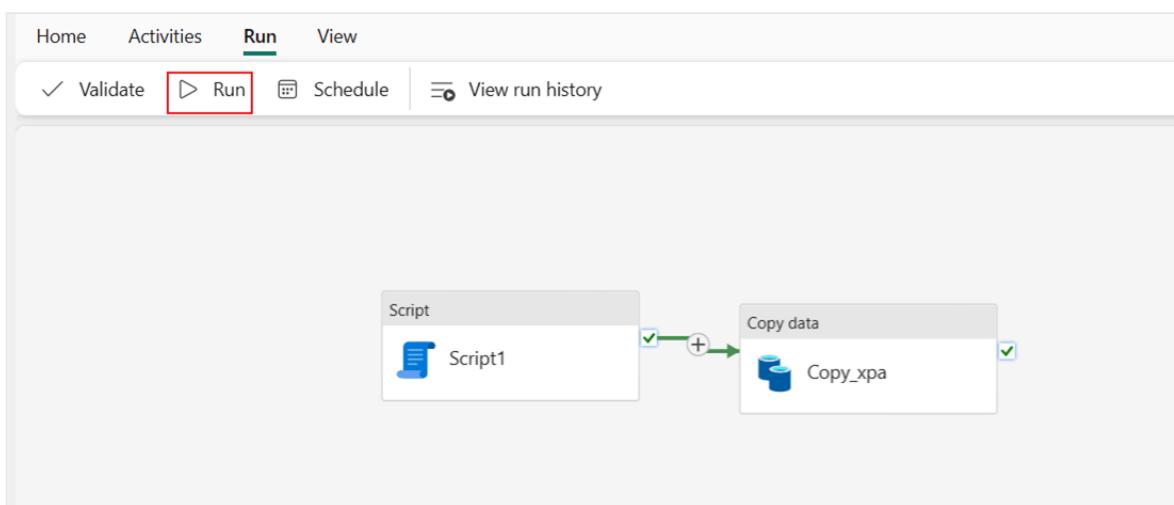


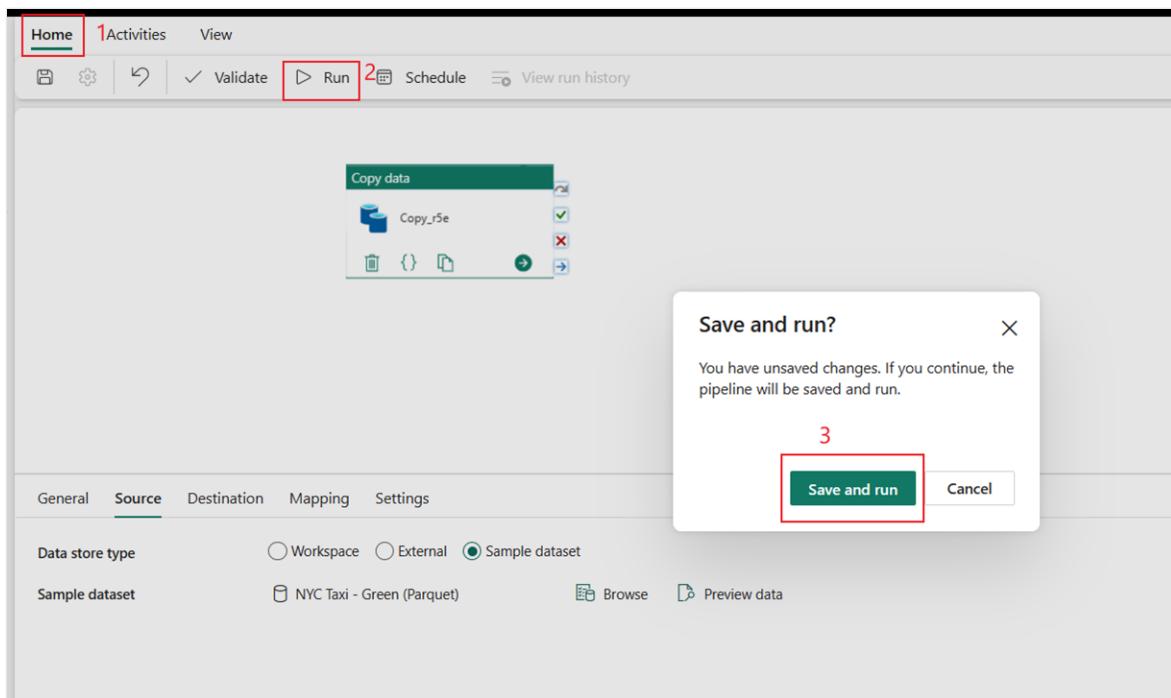
Execute the two pipeline activities to load the data

1. Connect the Script and Copy data activities by **On success** from the Script activity.

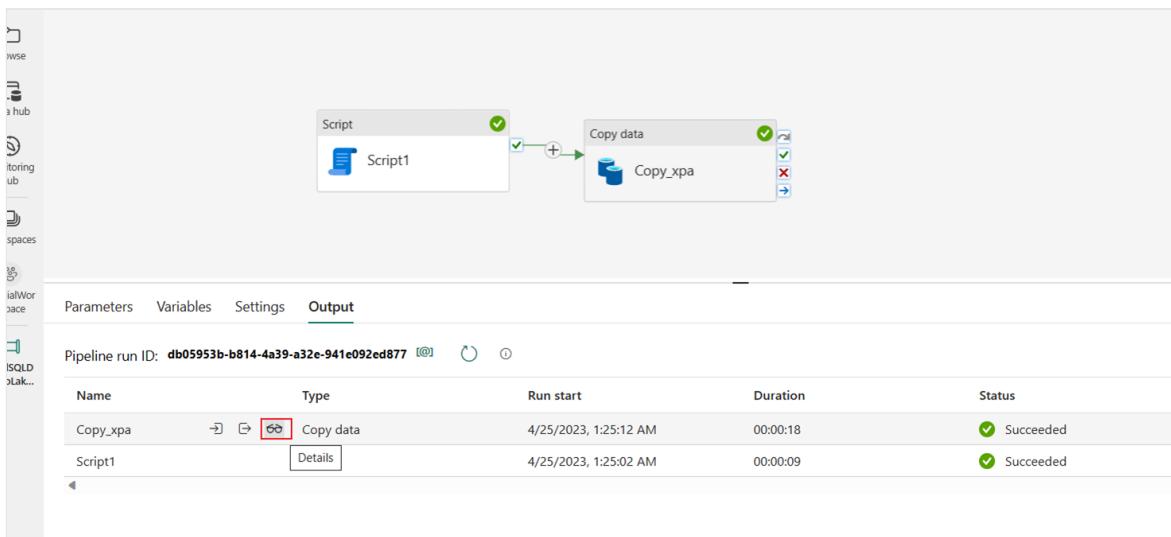


2. Select **Run** and then **Save and run** to run the two activities in the pipeline.





3. After the pipeline successfully runs, you can view the details for more information.



Copy data details

Copy_xpa

Source	Destination
 Azure Synapse Analytics	 Lakehouse
Data read: <small>①</small>	84 bytes
Rows read:	2
Data written: <small>①</small>	1.119 KB
Files written: <small>①</small>	1
Rows written: <small>①</small>	2

Status	 Succeeded
Start time	4/25/2023, 1:25:12 AM
Pipeline run activity ID	76f4d7ef-2a21-46c8-bb76-74ffd4e1c4d6
Throughput	10 bytes/s
Total duration	00:00:15
Duration breakdown	
Start time	4/25/2023, 1:25:13 AM

[Close](#)

4. Switch to the workspace and select the Lakehouse to check the results.

	Name	Type
Dataset	LHDemo	Dataset (default)
SQL endpoint	LHDemo	SQL endpoint
Lakehouse	LHDemo	Lakehouse

5. Select the table SynapseNamesTable to view the data loaded into Lakehouse.

SynapseNamesTable					
		id	fullname	first_name	last_name
		1	John Smith	John	Smith
		2	James Dean	James	Dean

Next steps

This sample shows you how to preprocess data with a stored procedure before loading the results into Lakehouse. You learned how to:

- ✓ Create a data pipeline with a Script activity to run a stored procedure.
- ✓ Use a pipeline activity to load the preprocessed table data into Lakehouse.
- ✓ Execute the pipeline activities to load the data.

Next, advance to learn more about monitoring your pipeline runs.

[How to monitor pipeline runs in Microsoft Fabric](#)

Move data from Azure SQL DB into Lakehouse via copy assistant

Article • 05/23/2023

This tutorial describes the steps to move data into Lakehouse.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

Two approaches are provided using the copy assistant:

1. The first approach moves source data into destination tables with delta format.
2. The second approach moves source data into destination files.

Prerequisites

To get started, you must complete the following prerequisites:

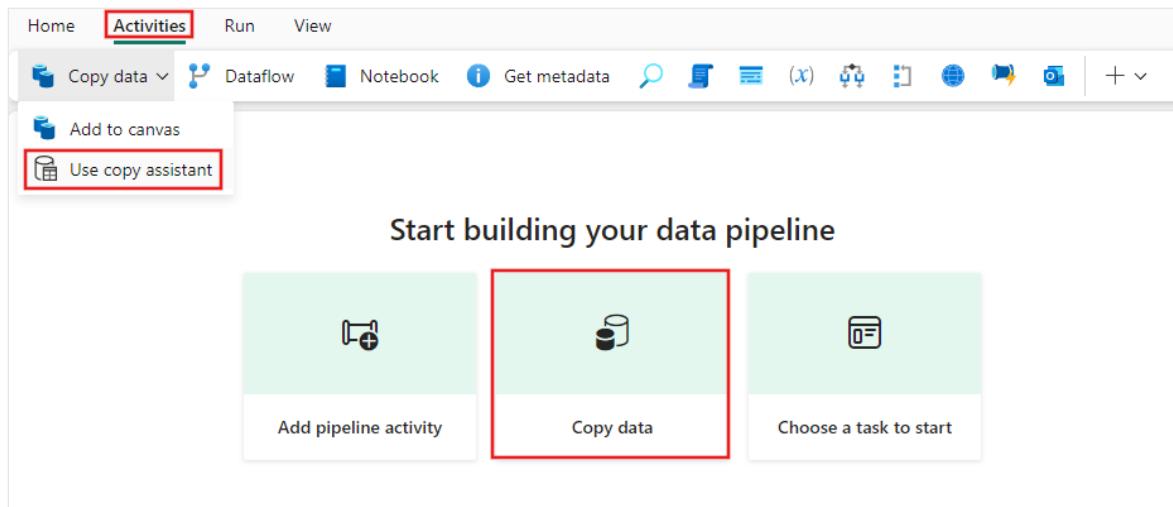
- A tenant account with an active subscription. Create an account for free.
- A workspace is created.
- A Lakehouse is created in your workspace.

Move files into Lakehouse as tables in delta format via copy assistant

Follow these steps to set up your copy activity.

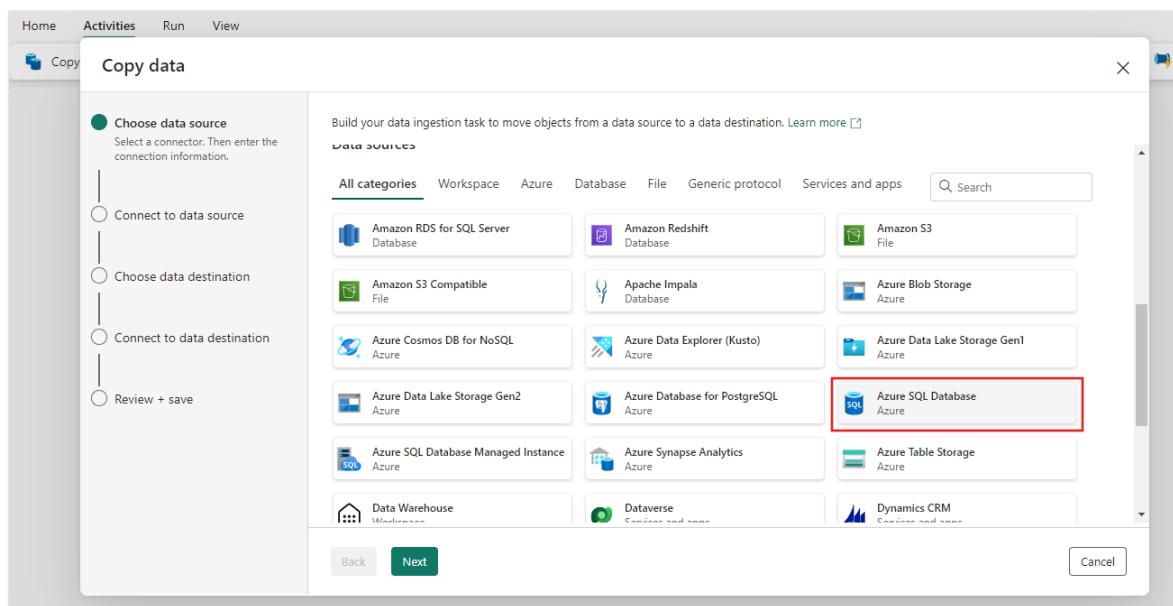
Step 1: Start with copy assistant

1. Open an existing data pipeline or create a new data pipeline.
2. Select **Copy data** on the canvas to open the **Copy Assistant** tool to get started. Or select **Use copy assistant** from the **Copy data** drop down list under **Activities** tab on the ribbon.

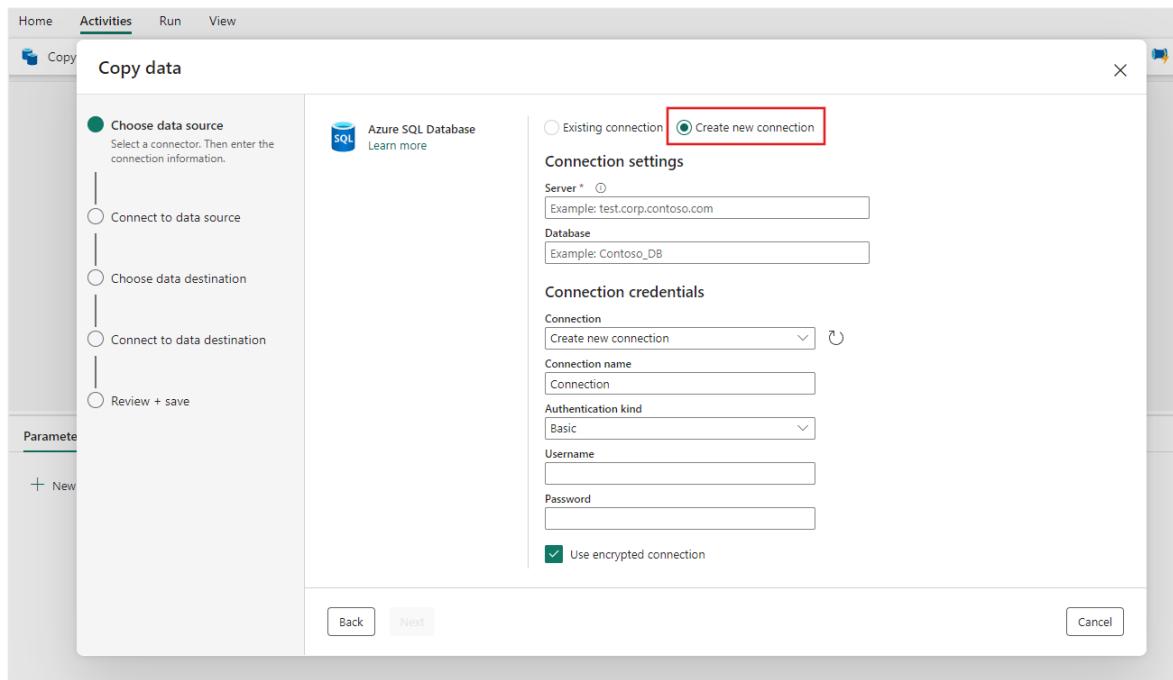


Step 2: Configure your source

1. Choose your data source by choosing a data source type. In this tutorial, we'll use Azure SQL Database as an example. Scroll down on the **Choose data source** screen to find and select **Azure SQL Database** and then select **Next**.

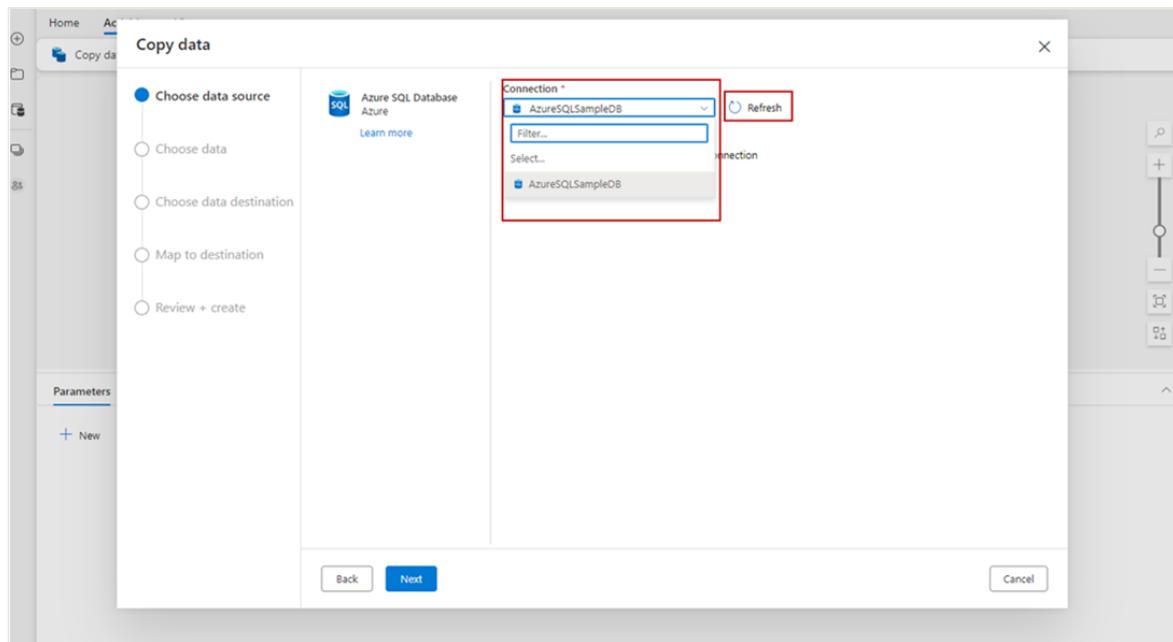


2. Create a connection to your data source by selecting **New Connection**, and filling in the required connection information on the panel.



After you fill in the required connection information on the panel, select **Next**.

Once your connection is created successfully, you will see a list of tables you can select.



3. Select the table(s) that is to be moved. Then, select **Next**.

The screenshot shows the 'Copy data' wizard in the Microsoft Data Integration service. The current step is 'Choose data'. On the left, there's a navigation pane with 'Copy data' selected. The main area has two tabs: 'Existing tables' (selected) and 'Use query'. Under 'Existing tables', a search bar and a list of tables are shown, with 'SalesLT.Product' selected. To the right is a preview grid for 'SalesLT.Product' containing the following data:

	ProductID	Name	ProductNumber	Color	StandardCost	ListPrice	Size
1	680	HL Road Frame - Black, 58	FR-R92B-58	Black	1059.3100	1431.5000	58
2	706	HL Road Frame - Red, 58	FR-R92R-58	Red	1059.3100	1431.5000	58
3	707	Sport-100 Helmet, Red	HL-U509-R	Red	13.0863	34.9900	
4	708	Sport-100 Helmet, Black	HL-U509	Black	13.0863	34.9900	
5	709	Mountain Bike Socks, M	SO-8909-M	White	3.3963	9.5000	M
6	710	Mountain Bike Gloves, L	SO-8909-L	White	3.3963	9.5000	L

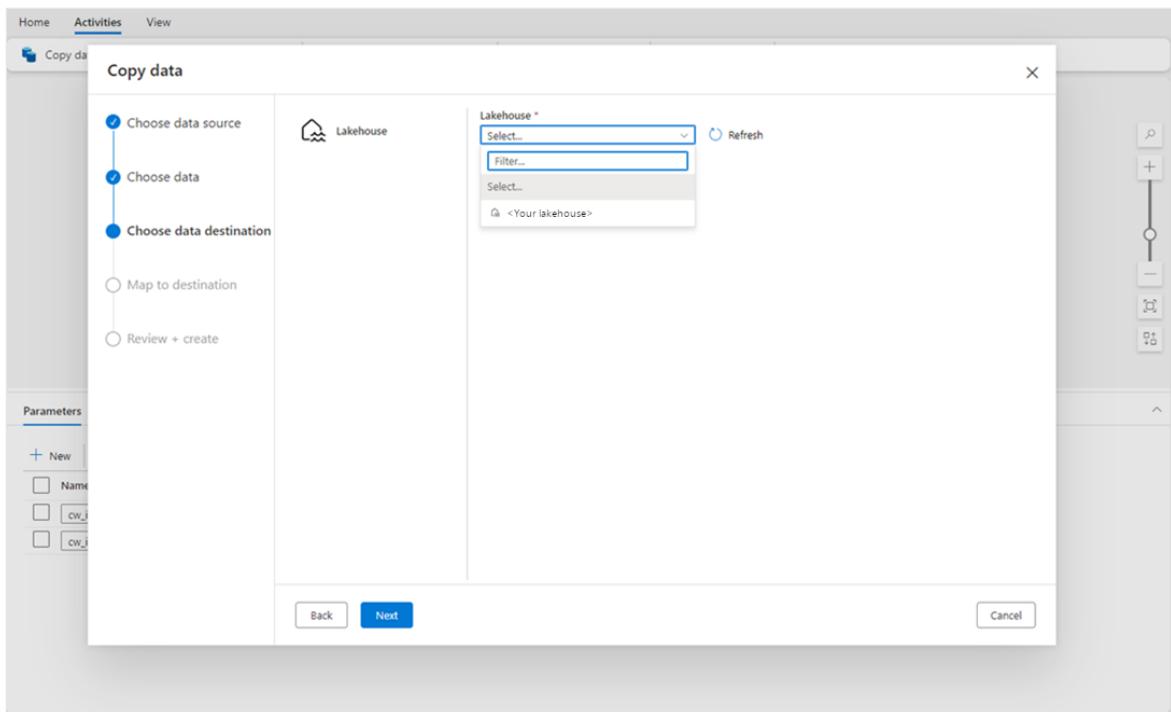
At the bottom are 'Back', 'Next', and 'Cancel' buttons.

Step 3: Configure your destination

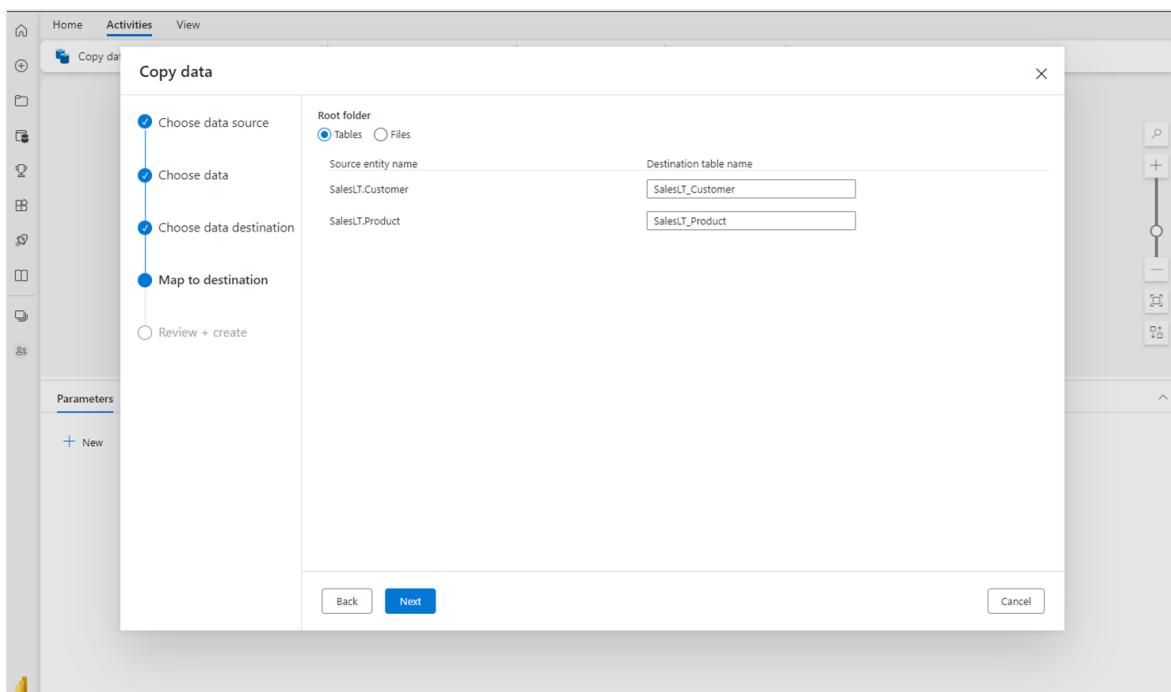
1. Choose Lakehouse as your destination and then go to next.

The screenshot shows the 'Copy data' wizard in the Microsoft Data Integration service. The current step is 'Configure your destination'. On the left, there's a navigation pane with 'Copy data' selected. The main area has a 'Data destinations' section with tabs: All categories, Workspace (selected), Azure, Database, Generic protocol, Services and apps. Under 'Workspace', there are three options: 'Data Warehouse Workspace', 'KQL Database Workspace', and 'Lakehouse Workspace', with 'Lakehouse Workspace' highlighted with a red box. At the bottom are 'Back', 'Next', and 'Cancel' buttons.

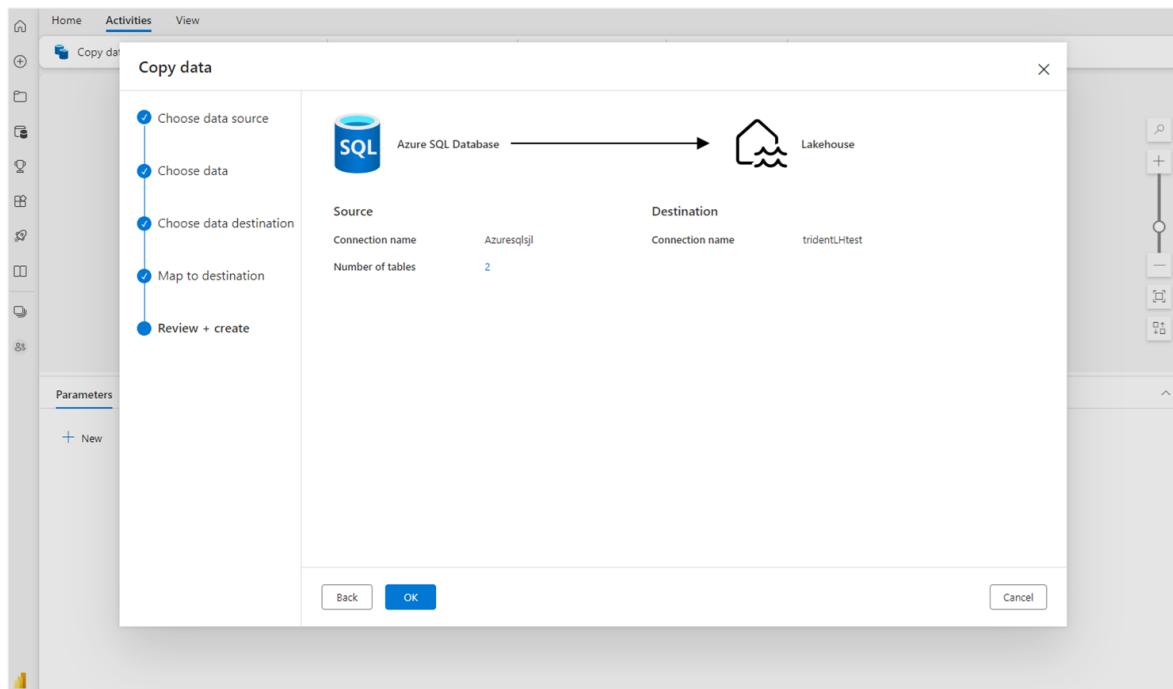
Select your existing Lakehouse from your current workspace directly and then go to next.



2. Configure your table settings in Lakehouse. Select **Tables** under Root folder and specify the **table name**. Select **Next** to continue.

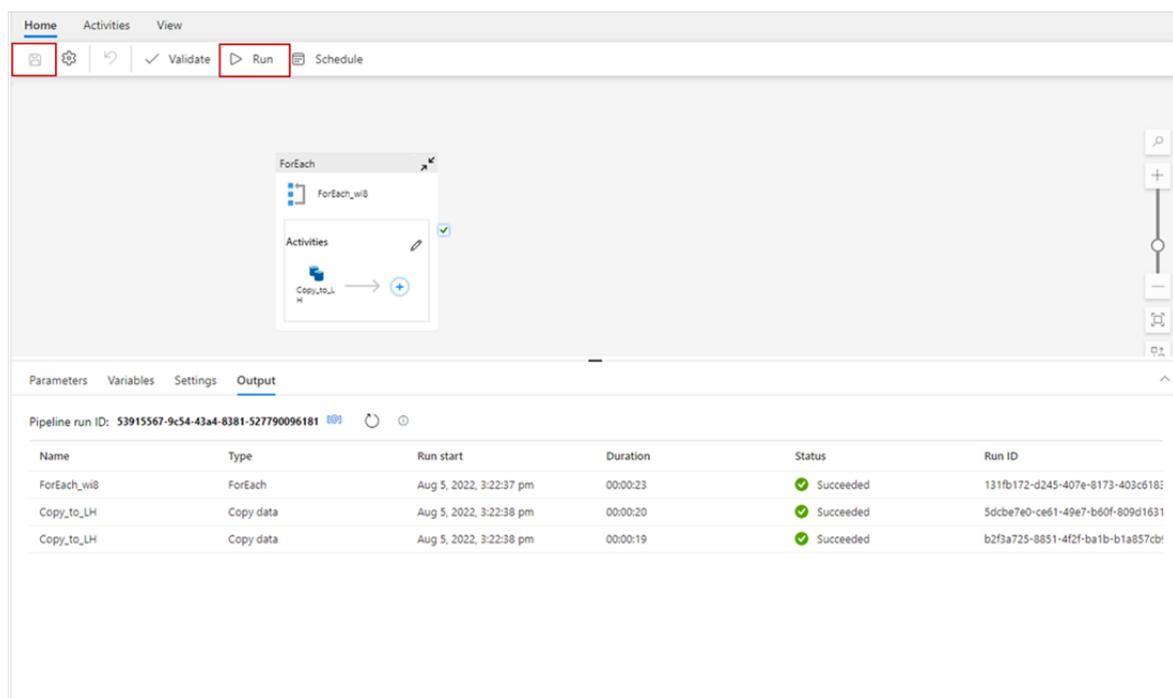


3. Select **OK** to finish the assistant experience.

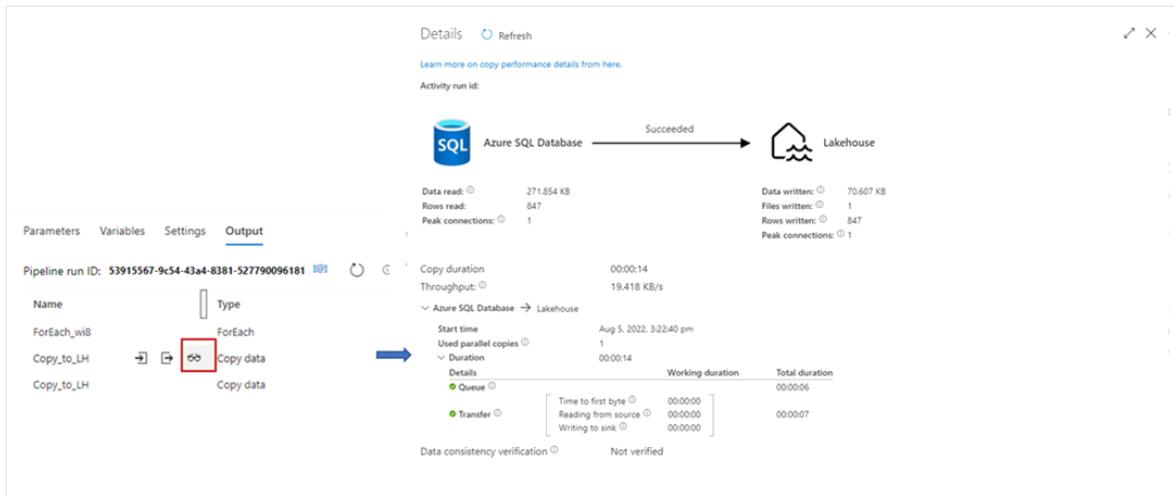


Step 4: Save your data pipeline and run it to load data

1. Select **Save**. Then, select **Run**



2. Select the **glasses** icon to view the details for each copy activity run:



Step 5: View your tables from Lakehouse

1. Go to your Lakehouse and refresh your **Lake view** to see the latest data ingested.
2. Switch to **Table view** to view the data in table.

The screenshot shows the Lakehouse portal interface. On the left, there's a navigation sidebar with 'Tables' selected. Below it, there are two tabs: 'Lake view' (which is currently active) and 'Table view'. Under 'Tables', there's a folder named 'Customer_tables' containing three items: a file named '535d7e54-6f40-4598-ad6f-62c7cbf1...', a folder named '_delta_log', and a file named 'abbf64a8-e2b1-4bcb-874e-1ff7e524...'. On the right, the 'Table view' section shows a table with columns: Name, Date modified, Type, and Size. The data listed is:

Name	Date modified	Type	Size
535d7e54-6f40-4598-ad6f-62c7cbf1...	8/5/2022 3:10:45 PM	PARQUET	163k
_delta_log	8/5/2022 3:10:45 PM	Folder	1 items
abbf64a8-e2b1-4bcb-874e-1ff7e524...	8/5/2022 3:10:44 PM	PARQUET	70k

⚠ Note

Currently data lands into Lakehouse Tables folders (a managed area) in Delta format only. Those files will be automatically registered as a table and be visible under Table view from Lakehouse portal. Only the first layer folders under Tables will be registered as delta table. Browsing or Previewing from Lakehouse Table isn't supported yet. Data that gets loaded into the same table will be appended. Delete or Update to tables isn't supported yet.

Next steps

This sample shows you how to move data from Azure SQL DB into Lakehouse with the Copy Assistant in Data Factory for Microsoft Fabric. You learned how to:

- ✓ Move files into Lakehouse as tables in delta format with the Copy Assistant.

Next, advance to learn more about monitoring your pipeline runs.

[How to monitor pipeline runs in Microsoft Fabric](#)

Load Sample data to Data Warehouse

Article • 05/23/2023

In this tutorial, you build a data pipeline to move a Sample dataset to the Data Warehouse. This experience shows you a quick demo about how to use pipeline copy activity and how to load data into Data Warehouse.

ⓘ 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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

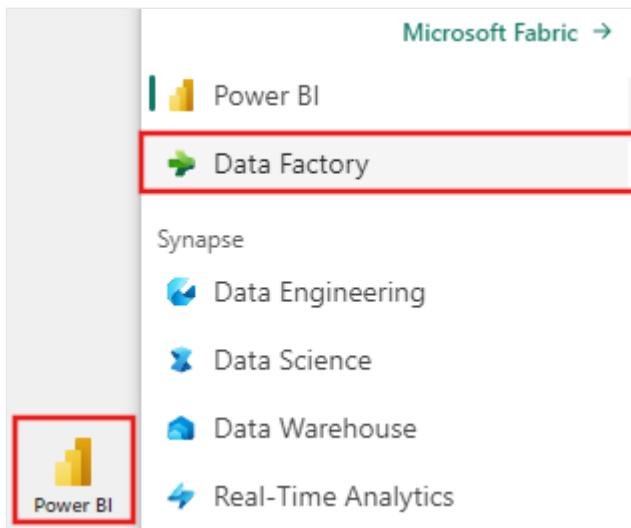
Prerequisites

To get started, you must complete the following prerequisites:

- A Microsoft Fabric tenant account with an active subscription. Create an account for free.
- Make sure you have a Microsoft Fabric enabled Workspace: [Create a workspace](#).
- Make sure you have already created a Data Warehouse. To create it, refer to [Create a Data Warehouse](#)

Create a data pipeline

1. Navigate to [Power BI](#).
2. Select the Power BI icon in the bottom left of the screen, then select **Data factory** to open homepage of Data Factory.



3. Navigate to your Microsoft Fabric workspace. If you created a new workspace in the prior Prerequisites section, use this one.

The screenshot shows the Microsoft Fabric Data Factory workspace. The left sidebar has icons for Home, Create, Browse, Data hub, Monitoring hub, Workspaces, and DF Getting Started (which is also highlighted with a red box). The main area is titled 'Recent' and shows a table of recent items:

	Name	Type	Op
	DF Getting Started	Workspace	33
	Release Plan Publish Schedule	Report	3 h
	pipeline4	Data pipeline	23

4. Select **Data pipeline** and then input a pipeline name to create a new pipeline.

The screenshot shows the Microsoft Data Factory interface. The top navigation bar includes the Microsoft logo, 'Data Factory', 'SampleDFWorkspace', and a search bar. On the left, a sidebar lists 'Home', 'Create', 'Browse', 'Workspaces' (with 'SampleDFWorkspace' selected), 'SampleDFPipeline', 'sample-dataflow', '...', 'More...', and 'Data Factory'. The main content area displays a list of items under 'Data pipeline (Preview)', which is highlighted with a red box. Other items include 'Eventstream (Preview)', 'Experiment (Preview)', 'KQL Database (Preview)', 'KQL Queryset (Preview)', 'Lakehouse (Preview)', 'Model (Preview)', 'Notebook (Preview)', 'Reflex (Preview)', 'Report', 'Spark Job Definition (Preview)', and 'Warehouse (Preview)'. To the right of the list, a tooltip provides a brief description: 'Ingest data at scale and schedule data workflows.' Below the list, a table maps item types to Data Factory components:

Type
Dataset (default)
SQL endpoint
Lakehouse
Data pipeline
Notebook
Data pipeline
Data pipeline
Dataflow

New pipeline X

Name

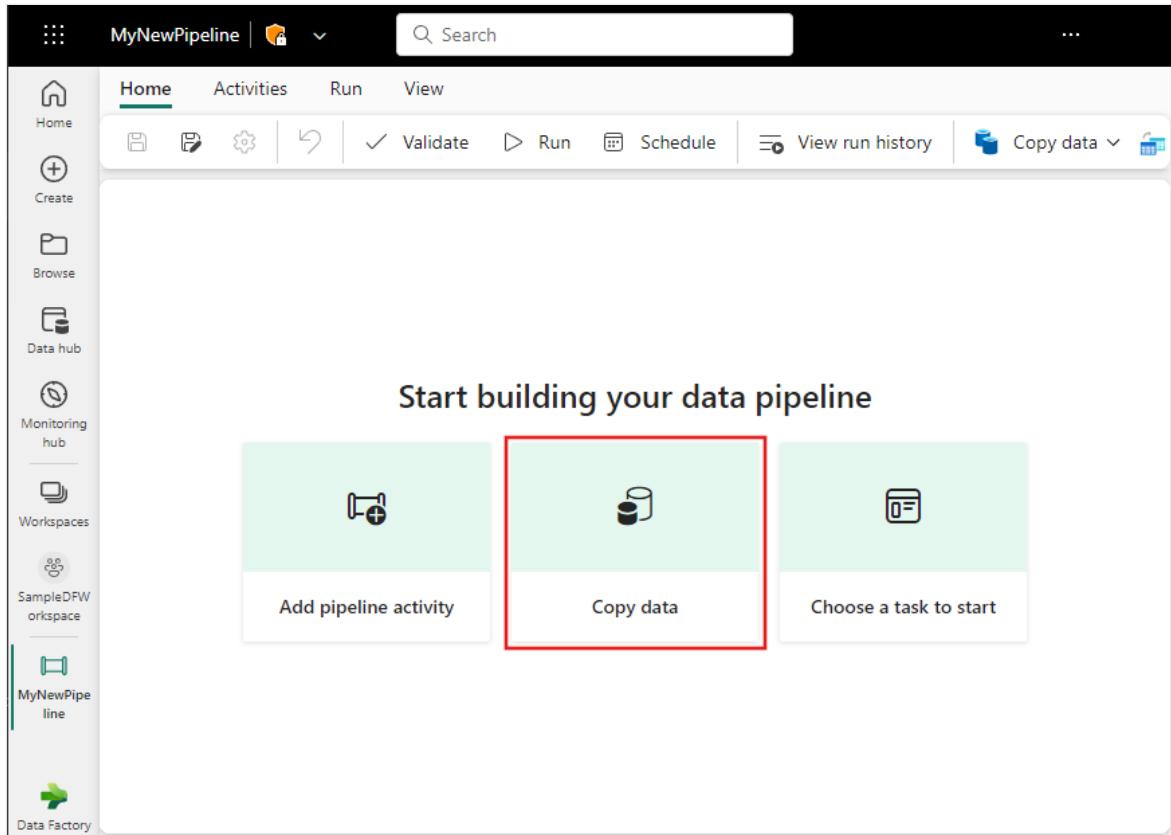
Create Cancel

Copy data using pipeline

In this session, you start to build your pipeline by following below steps about copying from a sample dataset provided by pipeline into Data Warehouse.

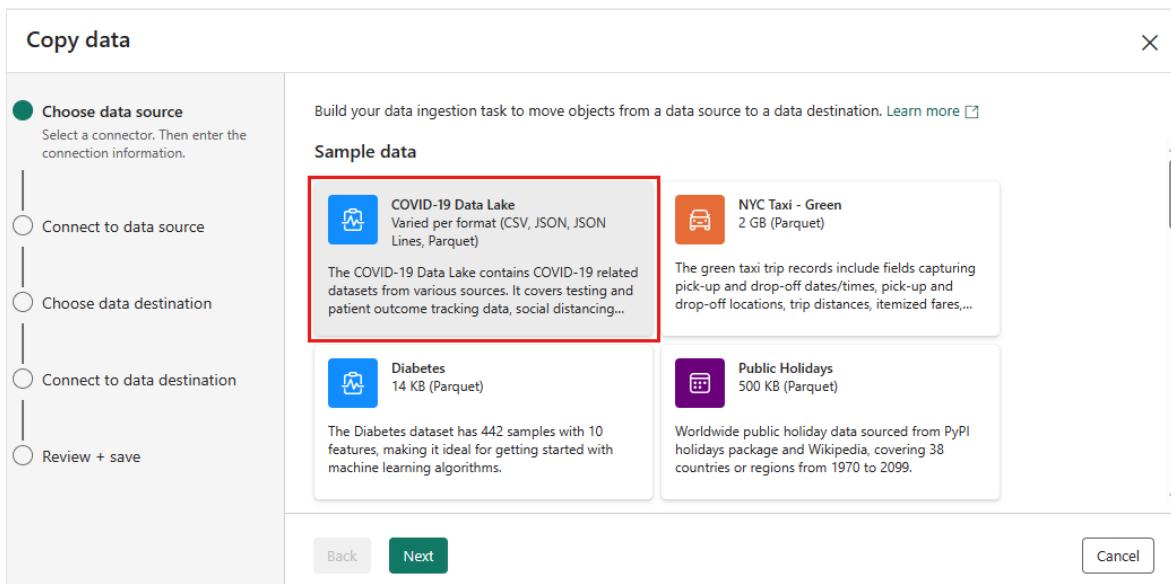
Step 1: Start with the Copy assistant

1. After selecting **Copy data** on the canvas, the **Copy assistant** tool will be opened to get started.



Step 2: Configure your source

1. Choose the **COVID-19 Data Lake** from the **Sample data** options for your data source, and then select **Next**.



2. In the **Connect to data source** section of the **Copy data** assistant, a preview of the sample data **Bing COVID-19** is displayed. Select **Next** to move on to the data destination.

Copy data

Choose data source

Connect to data source
Select, preview, and choose the data.

Choose data destination

Connect to data destination

Review + save

Select a dataset

- COVID-19 Data Lake
- Bing COVID-19**
- COVID Tracking project
- European Centre for Disease Prevention and Control (ECDC) COVID-19 Cases
- Oxford COVID-19 Government Response Tracker

Preview data: Bing COVID-19

Format: CSV (16.1 MB)

	abc_id	abc_updated	abc_confirmed	abc_confirmed_change	abc_deaths
1	338995	2020-01-21	262		0
2	338996	2020-01-22	313	51	0
3	338997	2020-01-23	578	265	0
4	338998	2020-01-24	841	263	0
5	338999	2020-01-25	1320	479	0
6	339000	2020-01-26	2014	694	0
7	339001	2020-01-27	2798	784	0
8	339002	2020-01-28	4593	1795	0
9	339003	2020-01-29	6065	1472	0
10	339004	2020-01-30	7818	1753	0

Back Next Cancel

Step 3: Configure your destination

1. Select the **Workspace** tab and choose **Data warehouse**. Then select **Next**.

Copy data

Choose data source

Connect to data source

Choose data destination
Define the data store as destination.

Connect to data destination

Review + save

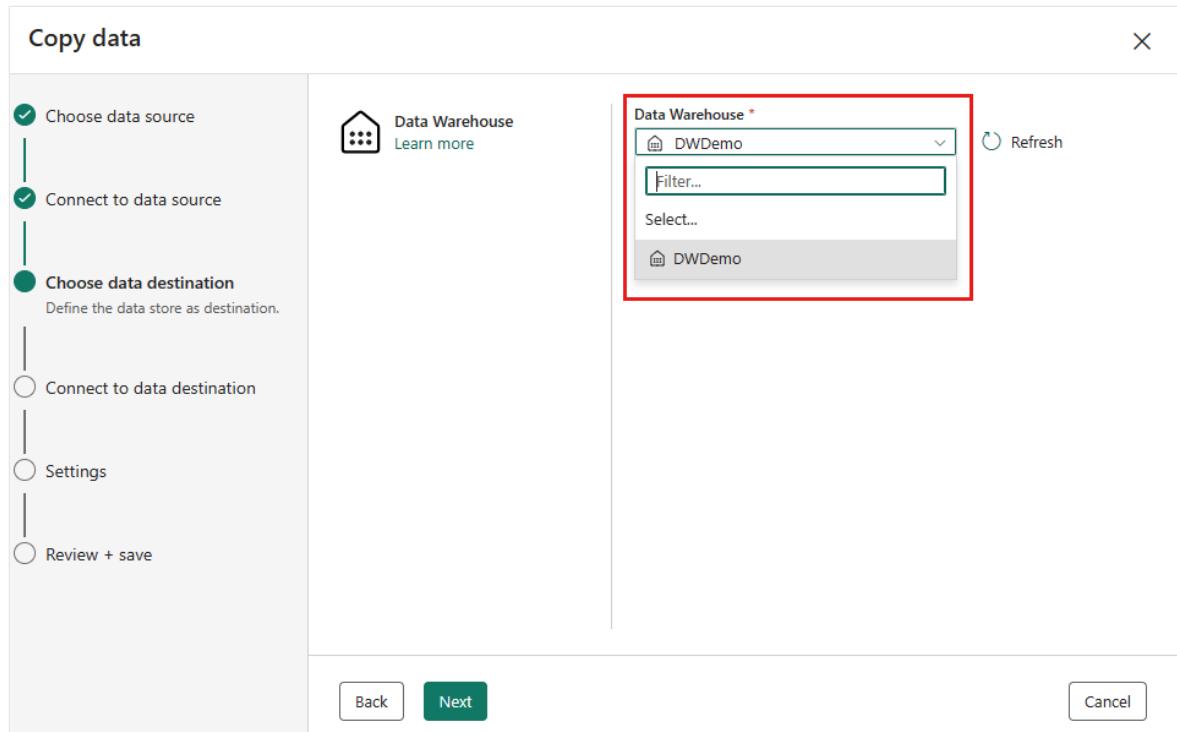
Data destinations

All categories **Workspace** Azure Database General Search

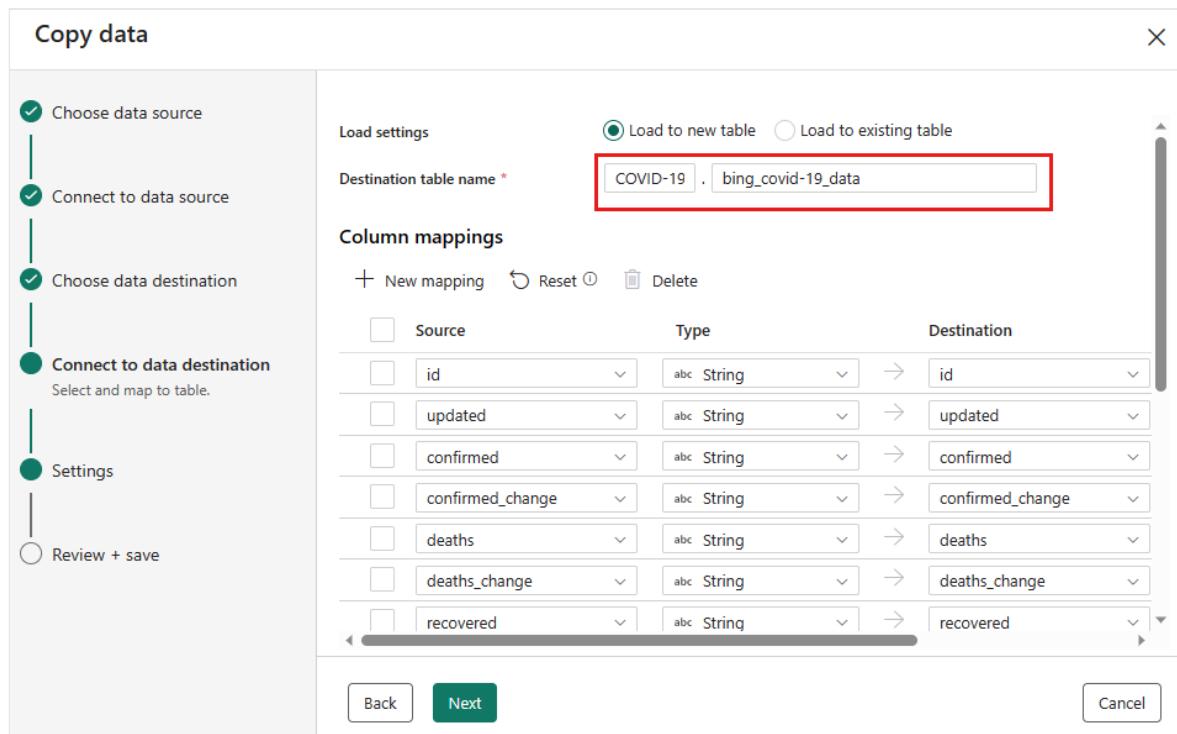
- Data Warehouse Workspace**
- Lakehouse Workspace
- KQL Database Workspace

Back Next Cancel

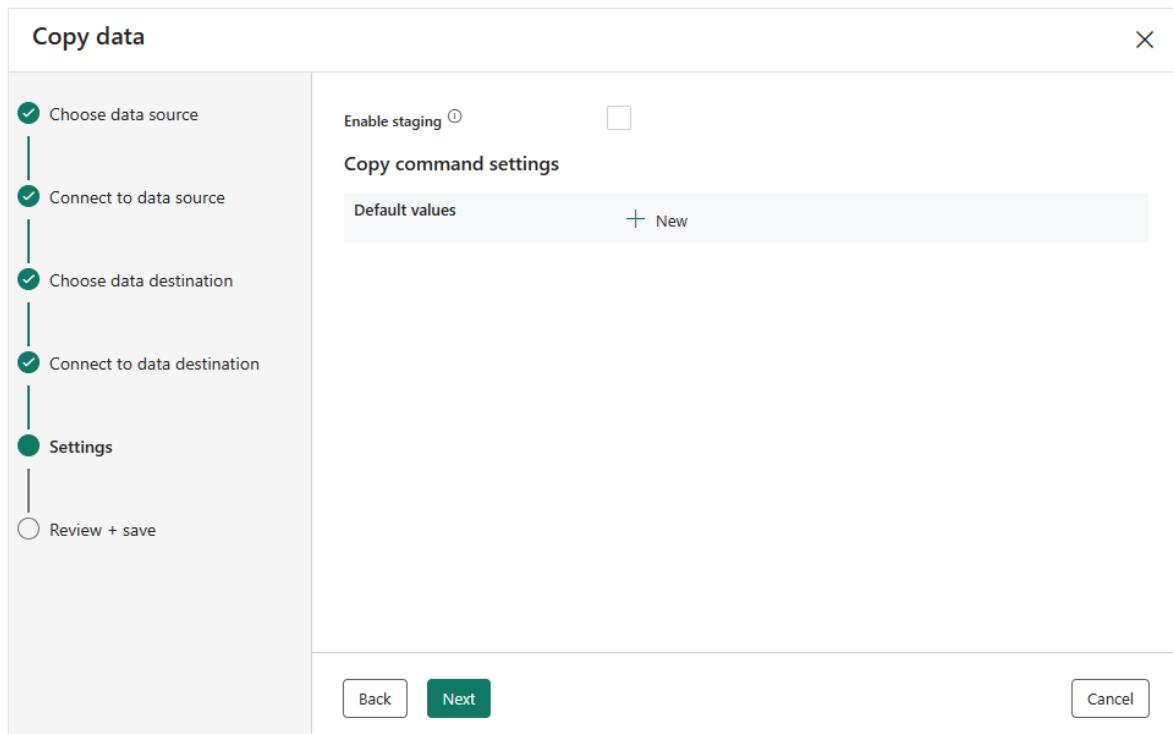
2. Select your Data Warehouse from the drop-down list, then select **Next**.



3. Configure and map your source data to the destination Data Warehouse table by entering **Destination table name**, then select **Next** one more time.

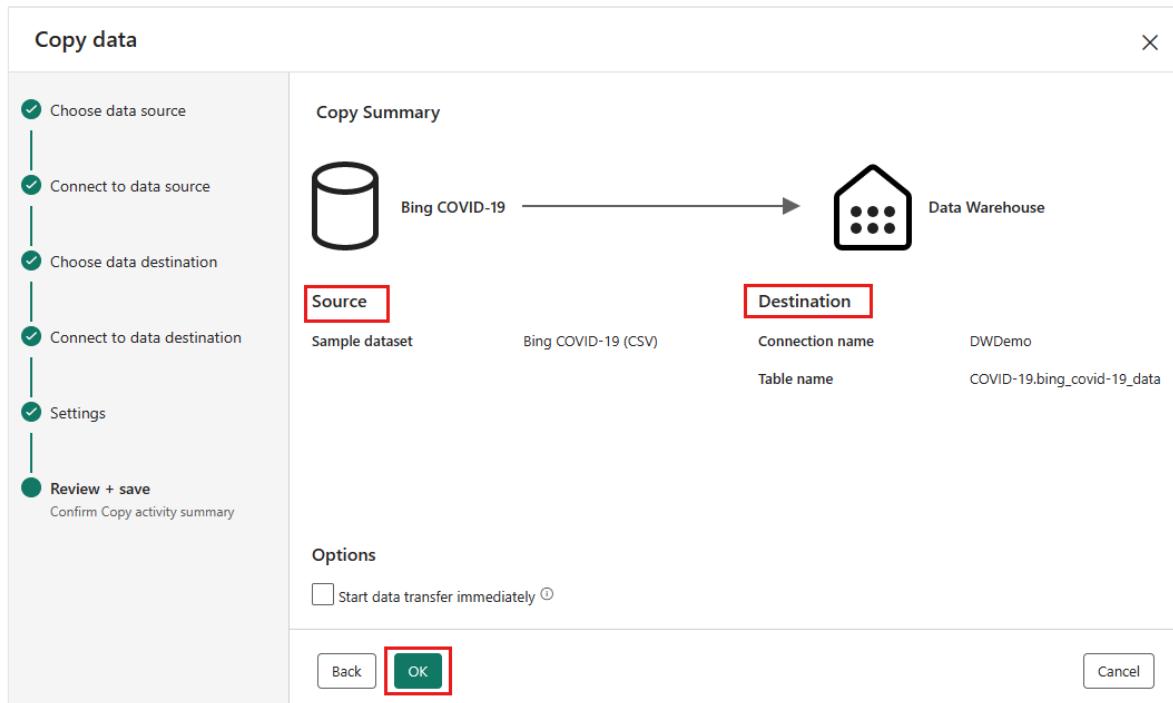


4. Configure other settings on **Settings** page. In this tutorial, select **Next** directly since you don't need to use staging and copy command.

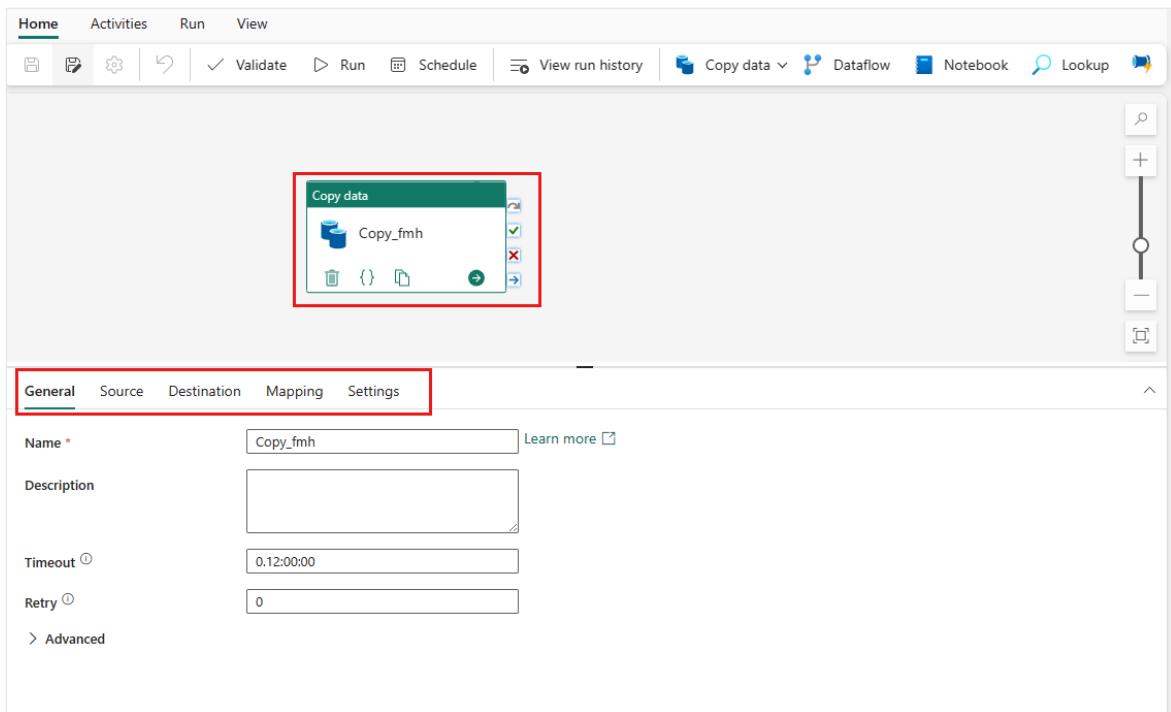


Step 4: Review and create your copy activity

1. Review your copy activity settings in the previous steps and select **OK** to finish. Or you can revisit the previous steps in the tool to edit your settings, if needed.

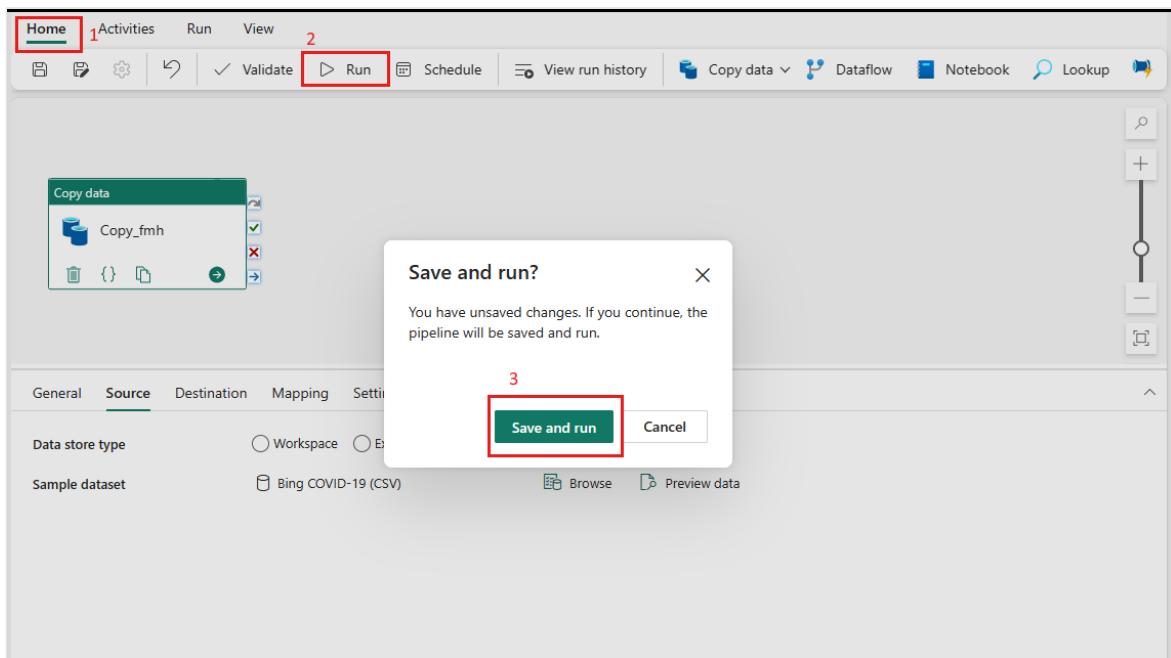


2. The Copy activity is added to your new data pipeline canvas. All settings including advanced settings for the activity are available in the tabs below the pipeline canvas when the created **Copy data** activity is selected.



Run and schedule your data pipeline

1. Switch to the **Home** tab and select **Run**. A confirmation dialog is displayed. Then select **Save and run** to start the activity.



2. You can monitor the running process and check the results on the **Output** tab below the pipeline canvas. Select the run details button (with the glasses icon highlighted) to view the run details.

The screenshot shows the Azure Data Factory interface. At the top, there's a navigation bar with 'Home' (underlined), 'Activities', 'Run', and 'View'. Below the navigation bar are several icons: a file icon, a pencil icon, a gear icon, a refresh icon, a checkmark icon labeled 'Validate', a play icon labeled 'Run', a calendar icon labeled 'Schedule', a history icon labeled 'View run history', a blue folder icon labeled 'Copy data', a dataflow icon, a notebook icon, and a search icon. The main area displays a 'Copy data' activity named 'Copy_fmh' with a green checkmark indicating success. Below this, there are tabs for 'Parameters', 'Variables', 'Settings', and 'Output' (which is highlighted with a red box). A pipeline run ID is shown as 'a74e2920-0c98-4fd2-80af-fc82a9c32096'. A table lists the run details:

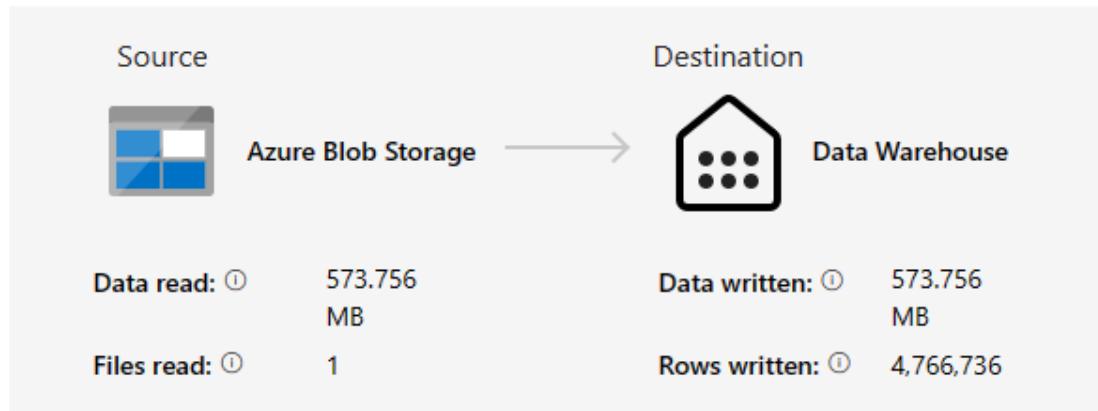
Name	Type	Run start	Duration	Status
Copy_fmh	Copy data	4/28/2023, 1:03:40 PM	00:00:53	Succeeded

A 'Details' button is located at the bottom of the table.

3. The run details show how much data was read and written and various other details about the run.

Copy data details

Copy_fmh

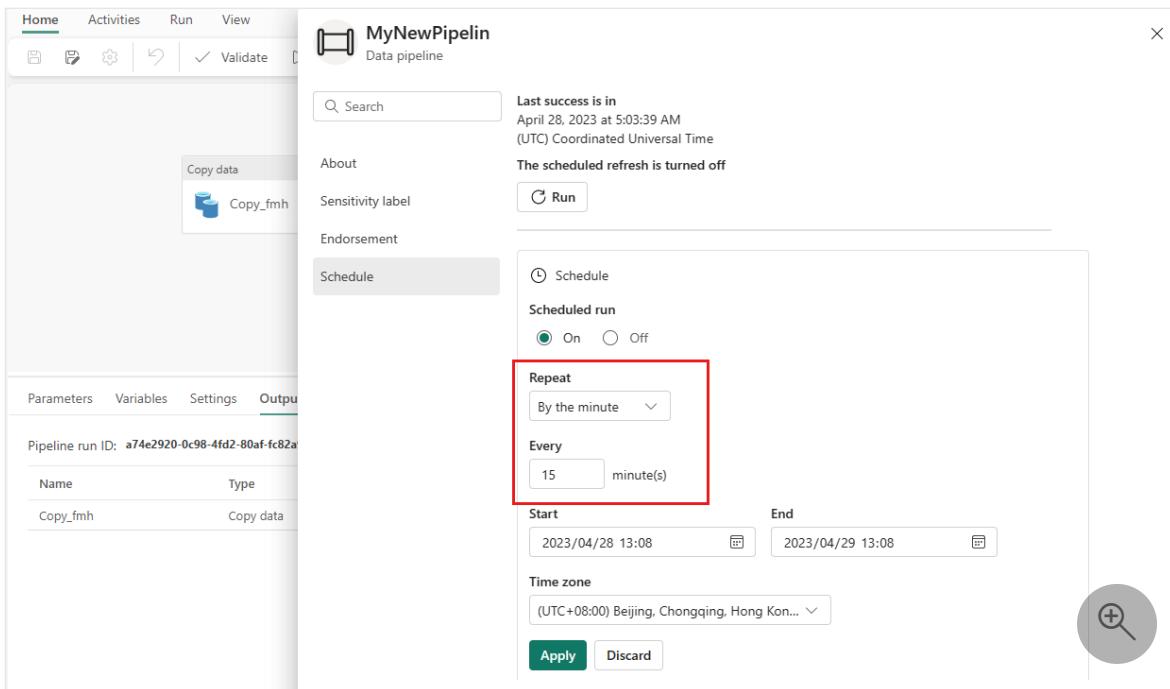


Status	 Succeeded
Start time	4/28/2023, 1:03:40 PM

Pipeline run activity ID	c80a9baf-8940-4b9a-8e68-131761c1ebf2
Throughput	12.75 MB/s
Total duration	00:00:51
Duration breakdown	
Advanced	

Close

4. You can also schedule the pipeline to run with a specific frequency as required. Below is an example scheduling the pipeline to run every 15 minutes. You can also specify the **Start** time and **End** time for your schedule. If you don't specify a start time, the start time is the time your schedule applies. If you don't specify an end time, your pipeline run will keep recurring every 15 minutes.



Next steps

This sample shows you how to load sample data into a Data Warehouse using Data Factory in Microsoft Fabric. You learned how to:

- ✓ Create a data pipeline.
- ✓ Copy data using your pipeline.
- ✓ Run and schedule your data pipeline.

Next, advance to learn more about monitoring your pipeline runs.

[How to monitor pipeline runs in Microsoft Fabric](#)

Data source management

Article • 05/23/2023

Microsoft Fabric supports many on-premises and cloud data sources, and each source has its own requirements. To learn how to add and manage an on-premises data source, go to [Add or remove a gateway data source](#). In this article, you'll learn how to add an Azure SQL Server as a cloud data source. The steps are similar for other data sources.

ⓘ 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.

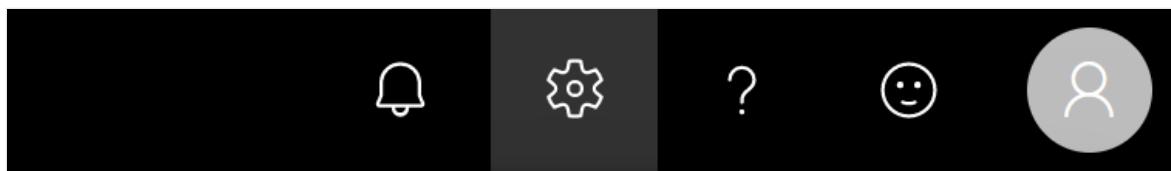
Refer to [Azure Data Factory documentation](#) for the service in Azure.

ⓘ Note

Currently, these cloud connections are only supported for data pipelines and Kusto. In the future, other items can also make use of the cloud connections. To create personal cloud connections in datasets, datamarts, and dataflows, use the Power Query Online experience in "get data".

Add a data source

1. From the page header in the Microsoft Fabric service, select the **Settings** icon, and then select **Manage connections and gateways**.



Settings

Settings

X

Preferences

General →

Notifications →

Item settings →

Developer settings →

Resources and extensions

Manage group storage →

Power BI settings →

Manage connections and gateways →

Manage embed codes →

2. Select the **Connections** tab, then select **New** at the top of the screen to add a new data source.

3. In the New connection screen, select **Cloud**, provide a **Connection name**, and select the **Connection Type**. For this example, choose **SQL server**.
4. Enter information about the data source. For SQL server, provide the **Server** and **Database**.

New connection

(i) Currently, these cloud connections are only supported for Data Pipelines and Kusto. In the future, other artifacts can also make use of the cloud connections. To create personal cloud connections in Datasets, Datamarts, and Dataflows, use the Power Query Online experience in "Get data".

 On-premises

 Virtual network

 Cloud

Connection name *
AdventureWorksProducts

Connection type *
SQL Server

Server *
adventureworks.database.windows.net

Database *
Products

5. Select an **Authentication method** to use when connecting to the data source, either **Basic**, **OAuth2**, or **Service Principal**. For example, choose **OAuth2** and sign in with your account.

Authentication

Authentication method *

OAuth2

[Edit credentials](#)

If you selected the **OAuth2** authentication method:

- Any query that runs longer than the OAuth token expiration policy might fail.
- Cross-tenant Azure Active Directory (Azure AD) accounts aren't supported.

6. Under **General > Privacy level**, optionally configure a [privacy level](#) for your data source. This setting doesn't apply to [DirectQuery](#).

General

Encrypted connection *

Encrypted

Privacy level *

Organizational

None

Private

Organizational

Public

7. Select **Create**. Under **Settings**, you see **Created new connection** if the process succeeds.

Settings



Created new connection.

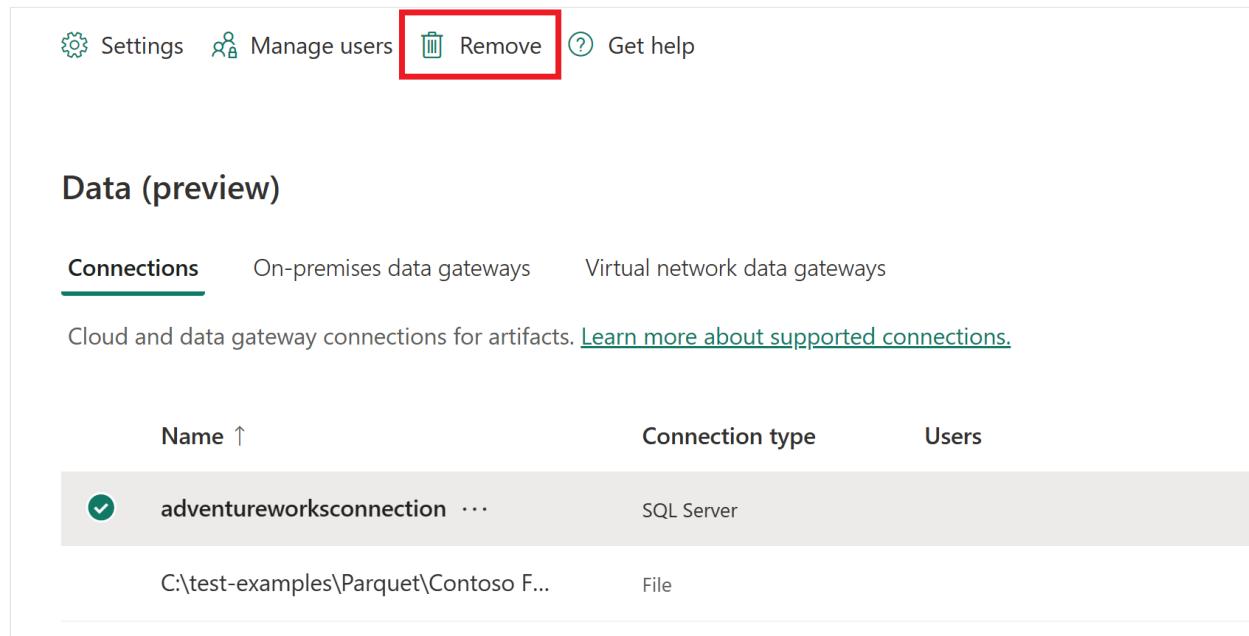


You can now use this data source to include data from Azure SQL in the supported Microsoft Fabric items.

Remove a data source

You can remove a data source if you no longer use it. If you remove a data source, any items that rely on that data source no longer work.

To remove a data source, select the data source from the **Data** screen in **Manage connections and gateways**, and then select **Remove** from the top ribbon.



The screenshot shows the 'Data' screen in the Power BI service. At the top, there are four buttons: 'Settings' (gear icon), 'Manage users' (two people icon), 'Remove' (trash bin icon, highlighted with a red box), and 'Get help' (question mark icon). Below the buttons, the title 'Data (preview)' is displayed. Underneath, there are three tabs: 'Connections' (underlined), 'On-premises data gateways', and 'Virtual network data gateways'. A note below the tabs says 'Cloud and data gateway connections for artifacts. [Learn more about supported connections.](#)' A table follows, listing data sources. The first row has columns 'Name ↑', 'Connection type', and 'Users'. It lists 'adventureworksconnection' (selected, indicated by a checkmark) with 'SQL Server' as the connection type and no users listed. The second row lists 'C:\test-examples\Parquet\Contoso F...' with 'File' as the connection type and no users listed.

Name ↑	Connection type	Users
adventureworksconnection ...	SQL Server	
C:\test-examples\Parquet\Contoso F...	File	

Manage users

After you add a cloud data source, you give users and security groups access to the specific data source. The access list for the data source controls only who is allowed to use the data source in items that include data from the data source.

Add users to a data source

1. From the page header in the Power BI service, select the **Settings** icon, and then select **Manage connections and gateways**.
2. Select the data source where you want to add users. To easily find all cloud connections, use the top ribbon to filter or search.

The screenshot shows the 'Data (preview)' screen. At the top, there are links for 'New' and 'Get help'. On the right, there are sections for 'Feature flags', 'Cloud connections' (which is highlighted with a dropdown menu showing 'On-premises connections', 'Virtual network connections', 'Cloud connections', and 'All'), and a search bar. Below the header, there are tabs for 'Connections' (selected), 'On-premises data gateways', and 'Virtual network data gateways'. A note says 'Cloud and data gateway connections for artifacts. [Learn more about supported connections](#)'. The main table has columns: 'Name ↑', 'Connection type', 'Users', 'Status', and 'Gateway cluster name'. A search icon is in the bottom right corner.

3. Select **Manage users** from the top ribbon.
4. In the **Manage users** screen, enter the users and/or security groups from your organization who can access the selected data source.
5. Select the new user name, and select the role to assign, either **User**, **User with resharing**, or **Owner**.
6. Select **Share**. The added member's name is added to the list of people who can publish reports that use this data source.

The screenshot shows the 'Manage users' screen. It includes a note about users who can use the connection in artifacts and a message about owner permissions. A search bar and a note about adding, removing, and modifying users are also present. The left panel shows 'New users' (Nia Wysocka, User with resharing) and 'Shared with' (Milan Blahos, Owner). The right panel shows three role options: 'User' (Allows the user to use the data source), 'User with resharing' (Allows the user to use the data source and reshare with others, which is selected), and 'Owner' (Allows the user to use the data source, manage data source configurations and credentials). At the bottom, there are 'Share' and 'Cancel' buttons.

Remember that you need to add users to each data source that you want to grant access to. Each data source has a separate list of users. Add users to each data source separately.

Remove users from a data source

On the **Manage Users** tab for the data source, you can remove users and security groups that use this data source.

Next steps

[Connectors overview](#)

View refresh history and monitor your dataflows

Article • 05/23/2023

Monitoring your dataflow refreshes is key in ensuring that your dataflows are running as expected. Refresh history and monitoring hub allows you to evaluate in detail what happened during the refresh of your dataflow. This article provides you with an overview of the features that are available in the refresh history and monitoring hub. We also provide you with some guidance on how to use these features.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Refresh history

Refresh history is available using the drop-down menu in your workspace. You can access it by selecting the **Refresh History** button.

Name	Type	Owner
Dataflow 1	Dataflow Gen2	Jean
LakeSunshine		
LakeSunshine		
LakeSunshine		
Notebook Sunbook		

A context menu is open over the "Dataflow 1" row, specifically over the three-dot ellipsis icon. The menu items are: Delete, Edit, Export json, Properties, Refresh history (which is highlighted with a red border), Settings, and View lineage.

When you open your dataflow refresh history, you first notice a list of all your data refreshes. This first screen provides you with information about:

- Start time
- Status
- Duration
- Type

The screenshot shows a modal window titled "Refresh history" for "Dataflow 1". At the top, there are three buttons: "Refresh now" (with a circular arrow icon), "Schedule refresh" (with a calendar icon), and "Download as CSV" (with a CSV icon). Below these buttons is a table with four columns: "Start time", "Status", "Duration", and "Type". There are two rows of data in the table. The first row shows a start time of "2/13/2023, 12:05:46 PM", a status of "Succeeded" (indicated by a green checkmark icon), a duration of "00:01:10", and a type of "On demand". The second row shows a start time of "2/9/2023, 8:26:42 PM", a status of "Succeeded" (green checkmark), a duration of "00:01:32", and a type of "On demand". At the bottom left of the modal, there is a link labeled "Edit dataflow" with a pencil icon.

Start time	Status	Duration	Type
2/13/2023, 12:05:46 PM	Succeeded	00:01:10	On demand
2/9/2023, 8:26:42 PM	Succeeded	00:01:32	On demand

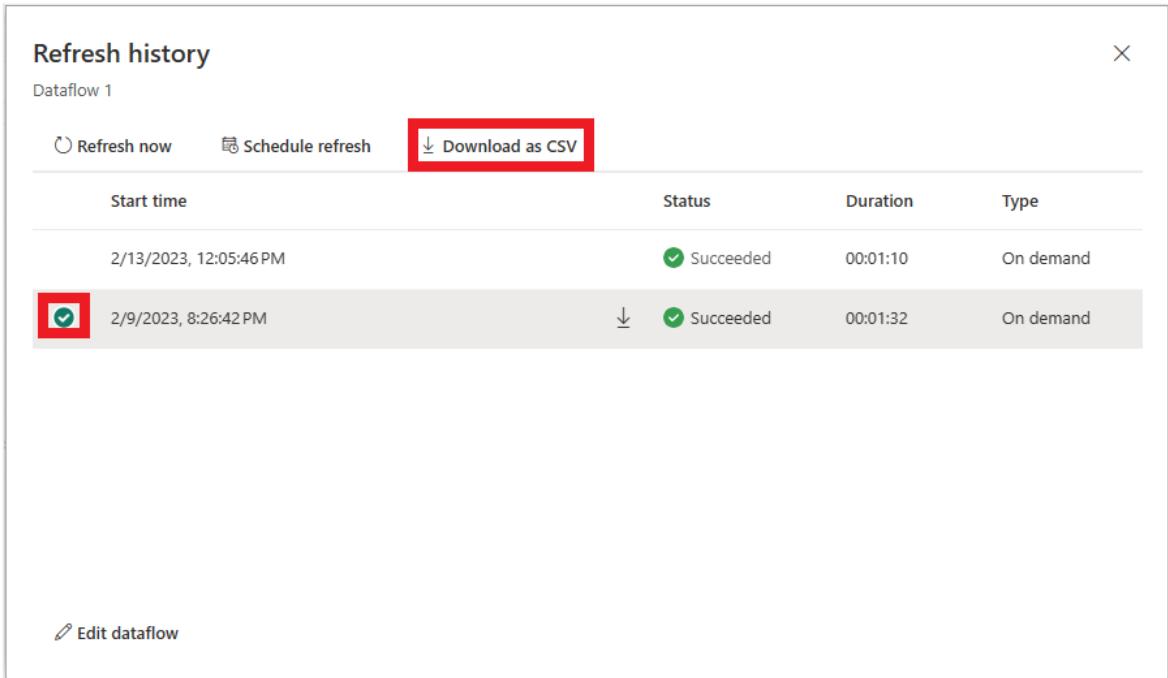
You can take some actions right away from this page like start a new refresh, schedule a refresh, or edit the dataflow.

Download a CSV file of the refresh

Some times you might need to get a CSV file of your refresh. To get this file, take the following steps:

1. Open the refresh history of the dataflow.
2. Select the run you want to get a CSV file from.

3. Download the CSV.



The screenshot shows a 'Refresh history' interface for a dataflow named 'Dataflow 1'. At the top, there are three buttons: 'Refresh now', 'Schedule refresh', and 'Download as CSV' (which is highlighted with a red box). Below this is a table with columns: Start time, Status, Duration, and Type. There are two rows of data:

Start time	Status	Duration	Type
2/13/2023, 12:05:46 PM	Succeeded	00:01:10	On demand
2/9/2023, 8:26:42 PM	Succeeded	00:01:32	On demand

At the bottom left is an 'Edit dataflow' link.

Reviewing your dataflow refresh from the UI

Once you've determined which dataflow you want to investigate, you can drill down into one of the refreshes by selecting the **Start time** field. This screen provides you with more information about the refresh that was performed. This includes general information about the refresh and a list of tables and activities.

The screenshot shows the 'Details' screen for a completed dataflow named 'Dataflow 1'. The top section displays basic information: Status (Succeeded), Type (On demand), Start time (2/9/2023, 8:26:42 PM), End time (2/9/2023, 8:28:14 PM), Duration (00:01:32), Request ID (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx), Session ID (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx), and Dataflow ID (xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx). Below this, the 'Tables' section lists 'Customers' and 'Orders' both in 'Succeeded' status. Under 'Activities', two tasks are listed: 'WriteToDatabaseTableFrom_TransformForOutputToDatabaseTableFrom_Customers' and 'WriteToDatabaseTableFrom_TransformForOutputToDatabaseTableFrom_Orders', both also in 'Succeeded' status. At the bottom left is a link to 'Edit dataflow'.

In short, this overview provides you:

- Status of the dataflow
- Type of refresh
- Start and End time
- Duration
- Request ID
- Session ID
- Dataflow ID

The **Tables** section reflects all the entities you've enabled load for in your dataflow. Meaning that those tables shown here are being loaded into the staging area of your dataflow. These tables are the entities you can access using the Power Query Dataflow connector in Power BI, Excel, or dataflows. You can select any of the listed table names to view the details of this specific table. Once you select the name, you arrive at the following **Details** screen:

[← Details](#)

Dataflow 1 > 2/9/2023, 8:26:42 PM > Customers

Name	Status
Customers	✓ Succeeded
Start time	End time
2/9/2023, 8:26:53 PM	2/9/2023, 8:27:35 PM
Duration	
00:00:41	

[Edit dataflow](#)

The **Activities** section reflects all the actions that have taken place during the refresh, for example loading data to your output destination. This table also allows you to dive deeper into the details of the specific activity. By selecting the name of the activity, you arrive at the following **Details** screen:

Details

Dataflow 1 > 2/9/2023, 8:26:42 PM >
WriteToDatabaseTableFrom_TransformForOutputToDatabaseTableFrom_Customers

Name	WriteToDatabaseTableFrom_TransformForOutputToDatabaseTableFrom_Customers
Status	✓ Succeeded
Start time	2/9/2023, 8:27:35 PM
End time	2/9/2023, 8:28:14 PM
Duration	00:00:39
Activity statistics	
Volume processed	Destination
50,208 bytes	Lakehouse
Volume processed	Destination
91 rows	SQL

[Edit dataflow](#)

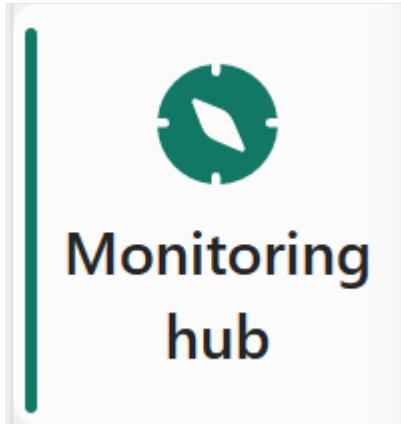
This screen gives you more clarity in what happened during the activity. For example, for output destinations the activity screen provides you with the:

- Status of the Activity
- Start and End time
- Duration
- Activity statistics:
 - Output destinations:
 - Endpoints contacted
 - Volume processed by the connector

To investigate what happened, you can drill down into an activity or table. The following screen provides you with general information about the refresh and errors. If you're drilling into an activity, you are presented with how much data got processed and sent to your output destination.

Monitoring hub

The monitoring hub is available using the side menu in your workspace. You can access it by selecting the **Monitoring hub** button.



The monitoring hub provides you with a dashboard that gives you an overview of the status of your dataflows.

Monitoring hub

Monitoring hub is a station to view and track active activities across different products.

Name	Status	Item type	Start time	Duration	Submitter	Location	Capacity	Average duration	Refreshes per day	Refresh type
Let the sun shine flow	Completed	Dataflow Gen2	5:53 PM, 3/8/23	1m 8s	Henry Cooper	Sunshine	Premium Per User - Re...	0s	0	OnDemand
New name for my dataflow	Unknown	Dataflow Gen2	10:06 AM, 3/6/23	1m 45s	Henry Cooper	Sunshine	Premium Per User - Re...	0s	0	OnDemand

Showing all available data

This dashboard provides you with the following information:

- Status of your dataflows
- Start time of the refresh
- Refresh duration
- Submitter of the refresh
- Workspace name
- Fabric capacity used for the refresh of your dataflow
- Average refresh duration
- Number of refreshes per day
- Type of refresh

Next steps

- Compare differences between Dataflow Gen1 and Gen2 in Data Factory
- Dataflows save as draft

Save a draft of your dataflow

Article • 05/23/2023

With Dataflow Gen2, we changed how saving a dataflow works. We wanted to improve the experience and resiliency of the Dataflows Gen1 by:

1. Automatically saving to the cloud any change made to a dataflow. This saved change is called the draft version of the dataflow.
2. Deferring long running validation required to guarantee a dataflow can refresh, to the background. The version of the dataflow that passed validation and is ready to refresh is called the published version.

This powerful feature allows you to make changes to your dataflow without immediately publishing them to your workspace. Instead, all your changes are automatically saved as a draft, which you can review at a later time, and then publish when you're ready. With this feature, you don't have to worry about losing your work if you want to resume it at a later time, if your dataflow fails validation, or if your editing session abruptly ends. In this article, you learn how to use the new Dataflow Gen2 auto-save and publish feature and how it can benefit your dataflow development and management.

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

How to save a draft version of your dataflow

Saving a draft of your dataflow is as easy as just closing your browser, closing the dataflows editors, or navigating to another workspace. Anytime you add a new Power Query step to your dataflow, a draft of your changes is saved to the workspace.

How to publish your draft dataflow

To publish the changes you made in your draft, you take the following steps:

1. Navigate to your workspace.

2. Open the draft dataflow that you recently saved changes to.
3. Review all the changes you made last time.
4. Publish the dataflow with the button on the bottom of the page

Your dataflow starts the publishing process in the background, which is indicated by a spinner next to the dataflow name. Once the spinner completes, the dataflow can be refreshed. If there are any publishing related errors, an indication is visible next to the dataflow name. Selecting the indication reveals the publishing errors and allows you to edit the dataflow from your last saved version.

Next steps

- Compare differences between Dataflow Gen1 and Gen2 in Data Factory

- Dataflows refresh history and monitoring

Data obfuscation in Data Factory with Delphix Compliance Services

Article • 05/23/2023

The following how-to outlines the use of Delphix Compliance Services (DCS) in Data Factory in Microsoft Fabric dataflows to mask sensitive data prior to delivery.

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.

Refer to [Azure Data Factory documentation](#) for the service in Azure.

DCS is a highly scalable masking API service that automatically masks personally identifiable information (PII), supplanting manual processes for delivering compliant data. Its out-of-the-box and configurable algorithms replace sensitive data values with fictitious yet realistic ones, so teams mitigate risk while ensuring end-users can easily consume the right data.

Masked data maintains multicloud referential integrity, is production-like in quality, and remains fully functional for accurate analysis or testing. Note that a DCS account needs to be created prior to use, and you can sign up for a [free trial](#).

What is the challenge?

The cloud is filled with personally identifiable information (PII), fueling privacy and security risk. PII from production apps needs to flow to downstream systems for analytics, exposing organizations to risks or creating data silos. Power Query and DCS automate data compliance and security to unblock data movement.

Breaking down data silos is difficult:

- Data must be manipulated to fit a common format. ETL pipelines must be adapted to each system of record and must scale to support the massive data sets of modern enterprises.
- Compliance with regulations regarding sensitive information must be maintained when data is moved from systems of record. Customer content and other sensitive elements must be obscured without impacting the business value of the data set.

How do DCS and Data Factory solve automating compliant data?

The movement of secure data is a challenge for all organizations. Delphix makes achieving consistent data compliance easy, while Data Factory enables connecting and moving data seamlessly. Together Delphix and Data Factory make the delivery of on-demand, compliant data easy.

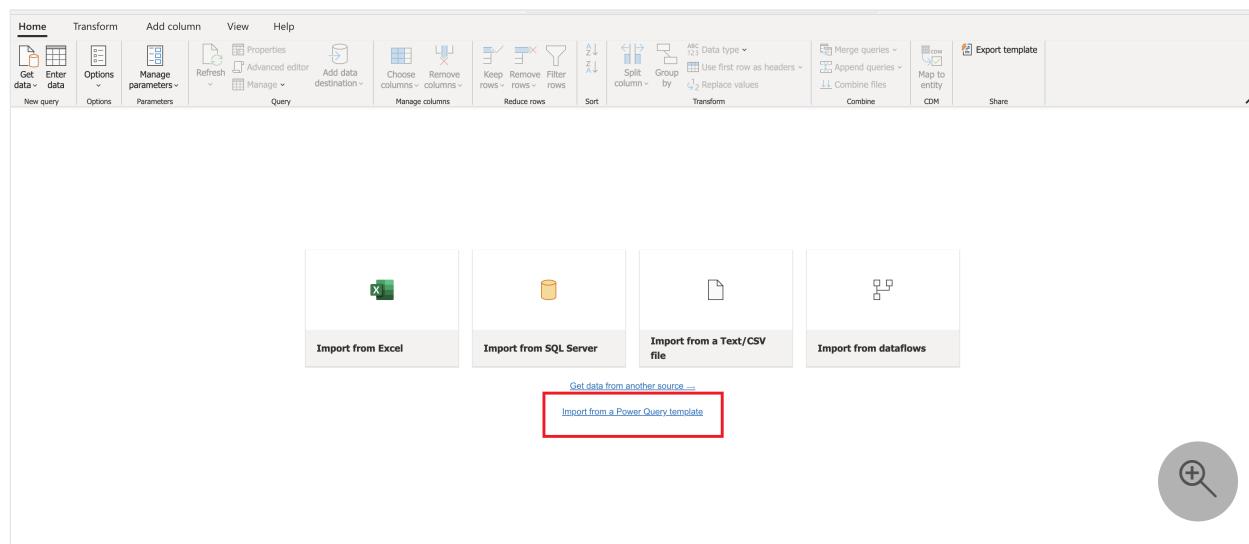
Using Data Factory data flows, you can create a workflow that automates the following steps:

- Read data from the desired source.
- Map sensitive fields to appropriate masking algorithms (and manage as a central configuration table).
- Call DCS masking APIs to replace sensitive data elements with similar but fictitious values.
- Load the compliant data to a desired target.

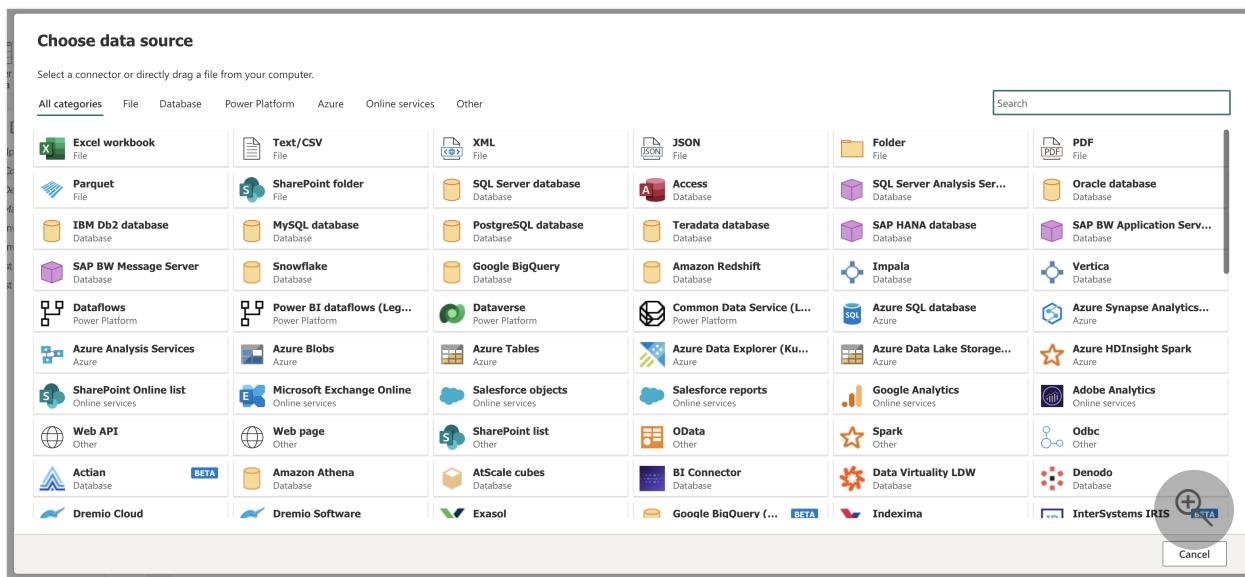
How to get started

Go to the [Delphix free preview page](#) to request a free trial of DCS. The Delphix team then contacts you for access and provides the template that is used in the example setup scenario described in this article.

In Power Query, upload the provided template by selecting **Import from a Power Query Template**, and then select the Power Query template file to import. This selection loads a set of queries.



Import the data source that contains sensitive data that you would like masked.

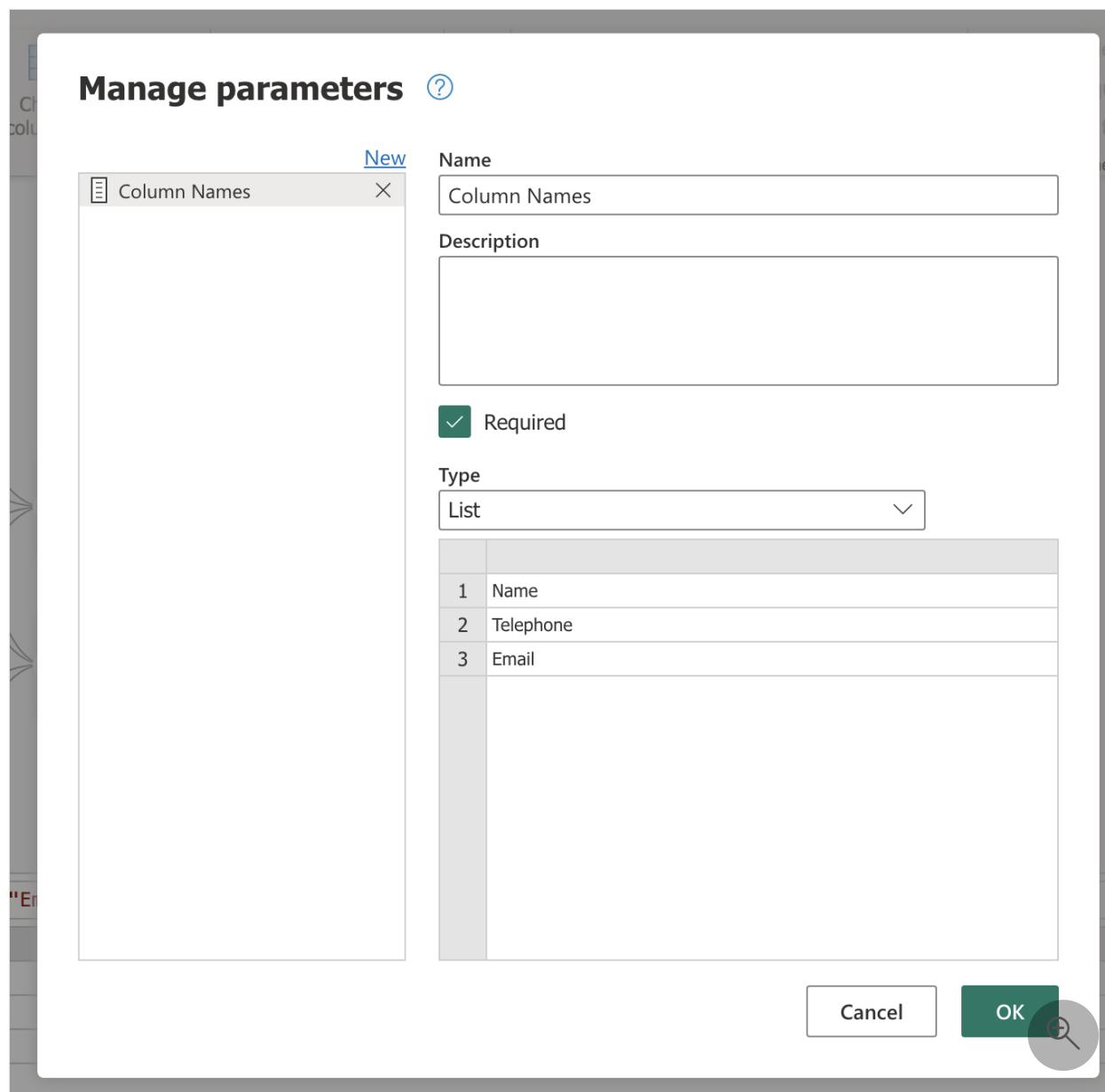


The mapping table is where you configure what fields to mask, and which Delphix masking algorithms to use. In the **Query Settings** pane, right-click on the gear icon. Enter the column names where sensitive data resides in **Original Column**. Enter the corresponding Delphix algorithm in **Algorithm**. Details on available algorithms can be found in the [Delphix documentation](#).

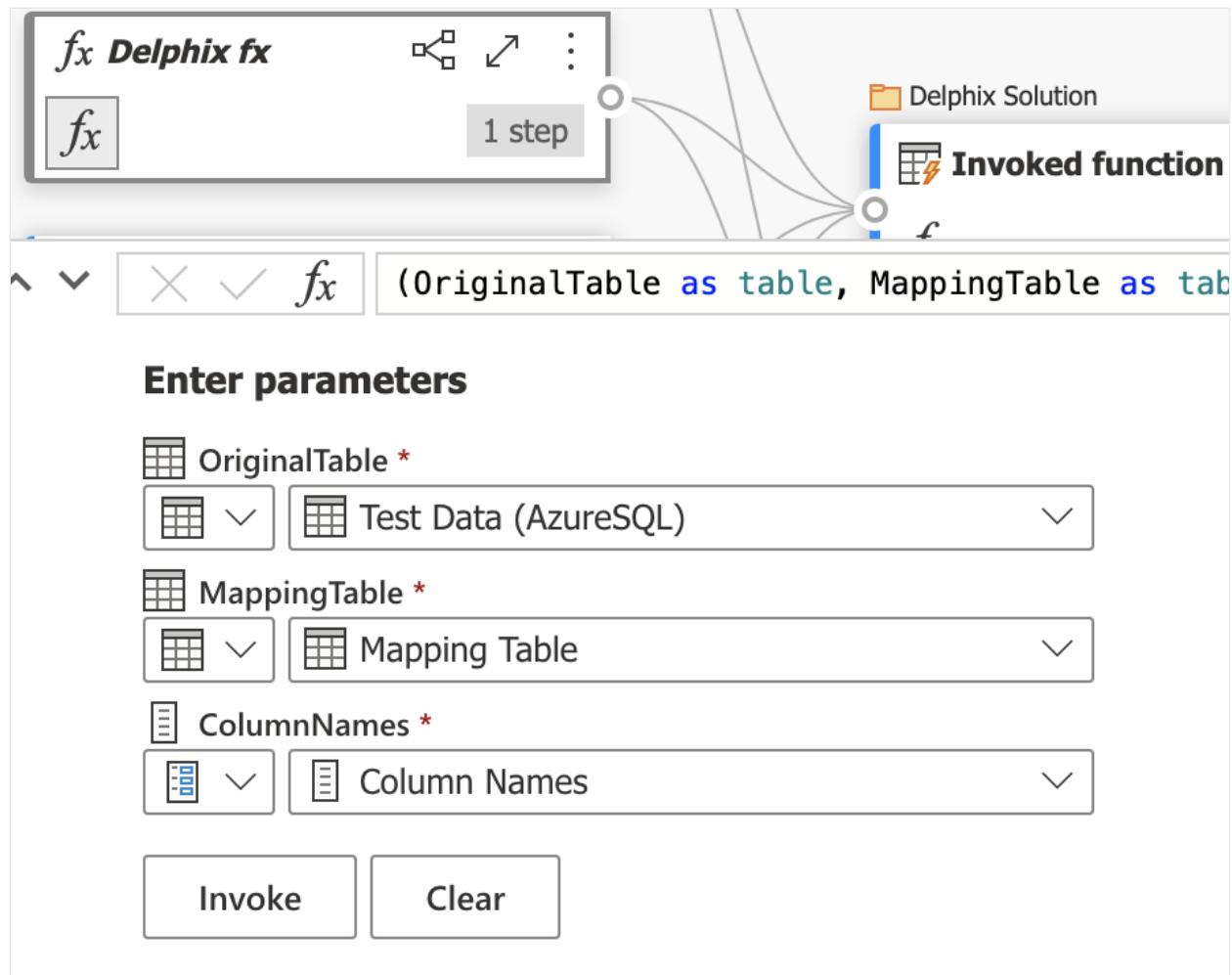
Delphix's out-of-the-box masking algorithms can be customized, or new algorithms can be defined if needed. All Delphix masking algorithms replace sensitive data with fictitious, yet realistic values, and do so consistently across data sets.

Original Column...	Algorithm
Name	dlpx-core:FullName
Telephone	dlpx-core:Phone US
Email	dlpx-core>Email SL

This mapping table can be a global configuration across any tables you would like to mask. Should you want to leave any columns unmasked for a given table, the **Column Names** parameter serves as a filter. Copy and paste the list of **Original Columns** (from the mapping table) into the **Column Names** parameter, and delete any column names that you would like to leave unmasked.



You're now ready to mask your data. Select **Delphix fx** and enter parameters as displayed in the following image (with the `OriginalTable` field as the data source that contains sensitive data).



Once this change is complete, select **Invoke** to run the data flow. This selection automatically calls the DCS masking API service to mask data prior to delivery to the destination of your choice.

Your data is now ready to be used safely by end users. The data is masked consistently, ensuring that references remain intact across data sets. As an example, George Smith becomes Edward Robinson regardless of data source or destination, ensuring it's still valuable for integrated analytics scenarios.

Next steps

- [Delphix free preview page ↗](#)
- [Delphix documentation ↗](#)

Move queries from Dataflow Gen1 to Dataflow Gen2

Article • 05/23/2023

Dataflow Gen2 is the new generation of dataflows. However, many existing dataflows have been created over the years using the original version of dataflows (now called *Dataflow Gen1* and listed as the **Dataflow** type in your workspace). This article describes how you can reuse your older Dataflow Gen1 dataflows by importing Dataflow Gen1 dataflows into Dataflow Gen2 using the export template feature. It also includes information on how to copy the Gen1 queries and paste them into a new Dataflow Gen2.

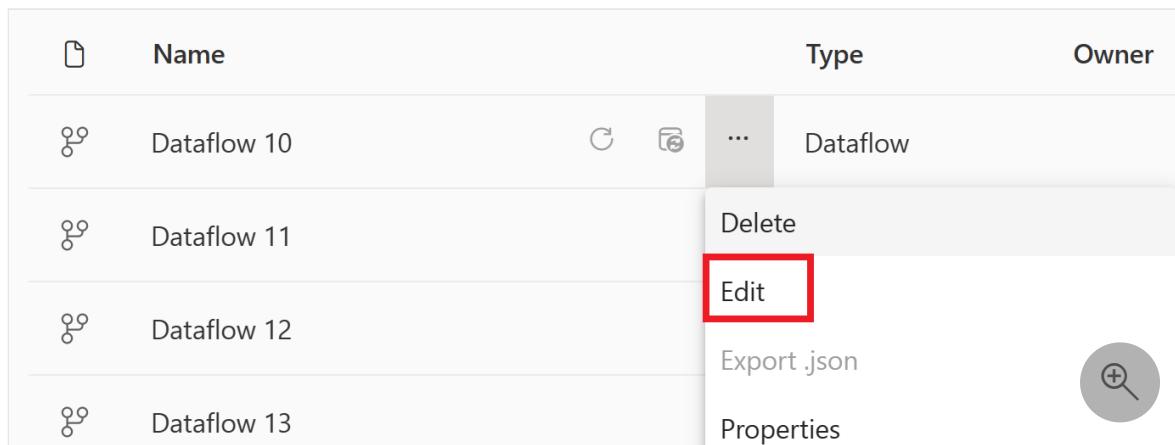
ⓘ Important

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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

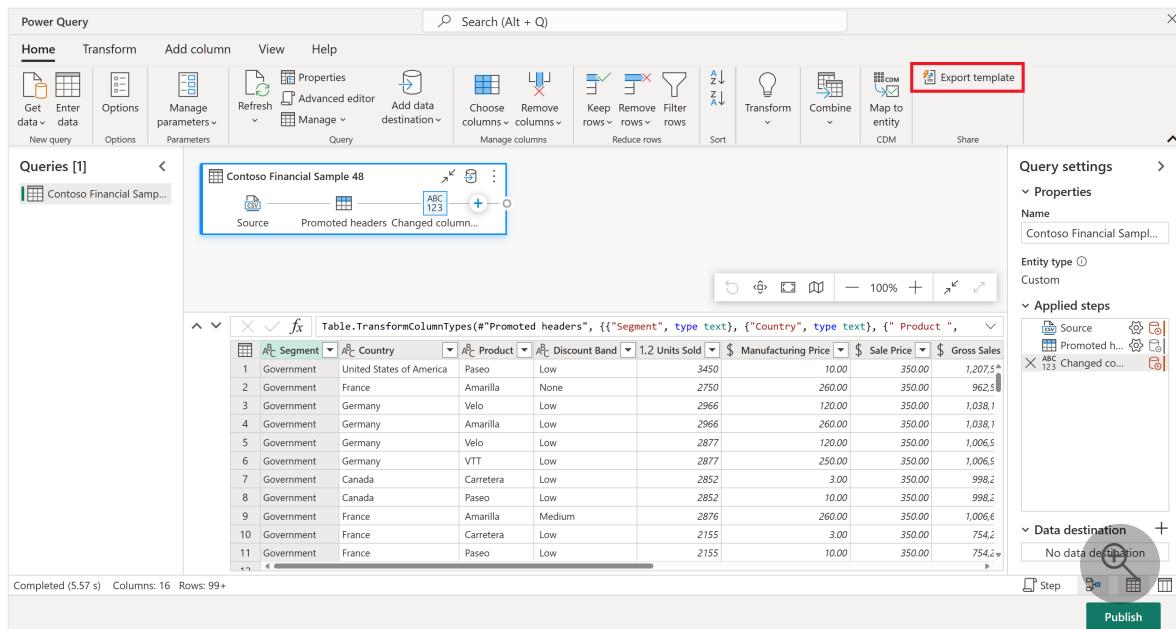
Use the export template feature

To use the export template feature:

1. From your Power BI or Data Factory workspace, select the ellipsis next to an existing Dataflow Gen1 and then select **Edit**.



2. In the **Home** tab of the Power Query editor, select **Export template**.



3. In **Export template**, enter the name you want to call this template in **Name**.
Optionally, you can add a description for this template in **Description**.

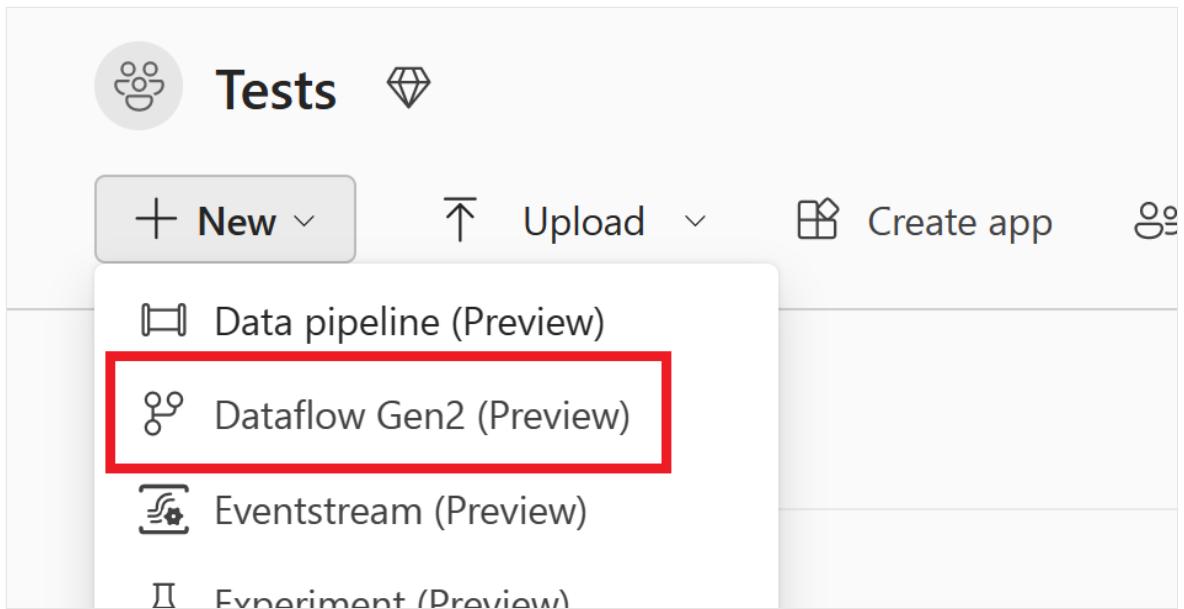
Export template

Name *

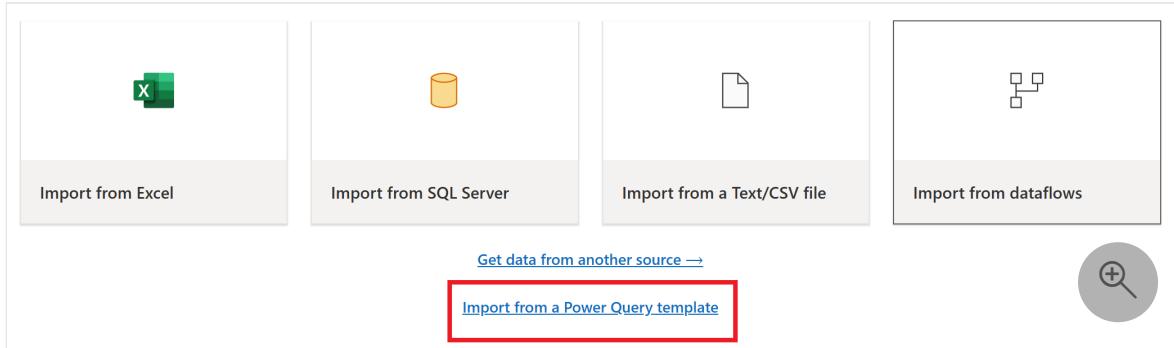
Description

OK
Cancel

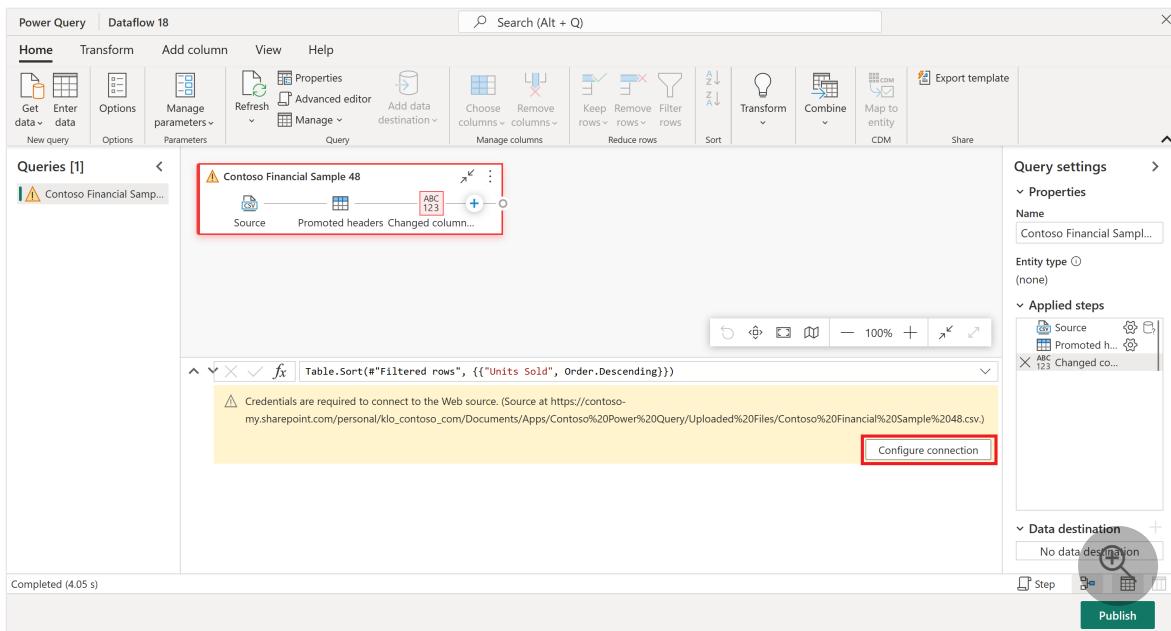
4. Select **OK** to save the template. The template is saved in your default Downloads folder.
5. From your Data Factory workspace, select **New**, and then select **Dataflow Gen2 (Preview)**.



6. From the current view pane of the Power Query editor, select **Import from a Power Query template**.



7. In the **Open** dialog box, browse to your default Downloads folder and select the .pqt file you saved in the previous steps. Then select **Open**.
8. The template is then imported to your Dataflow Gen2. You might be required to enter your credentials at this time. If so, select **Configure connection** and enter your credentials. Then select **Connect**.



Your Dataflow Gen1 has now been imported to Dataflow Gen2.

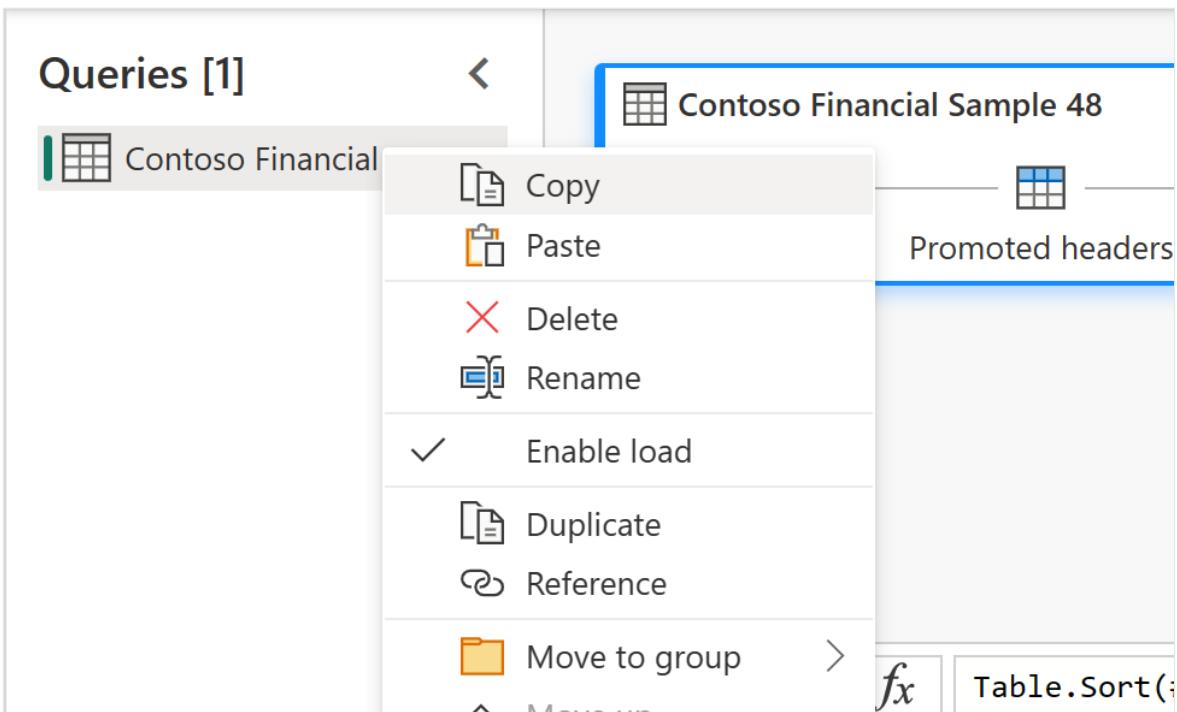
Copy and paste existing Dataflow Gen1 queries

To copy existing Dataflow Gen1 queries:

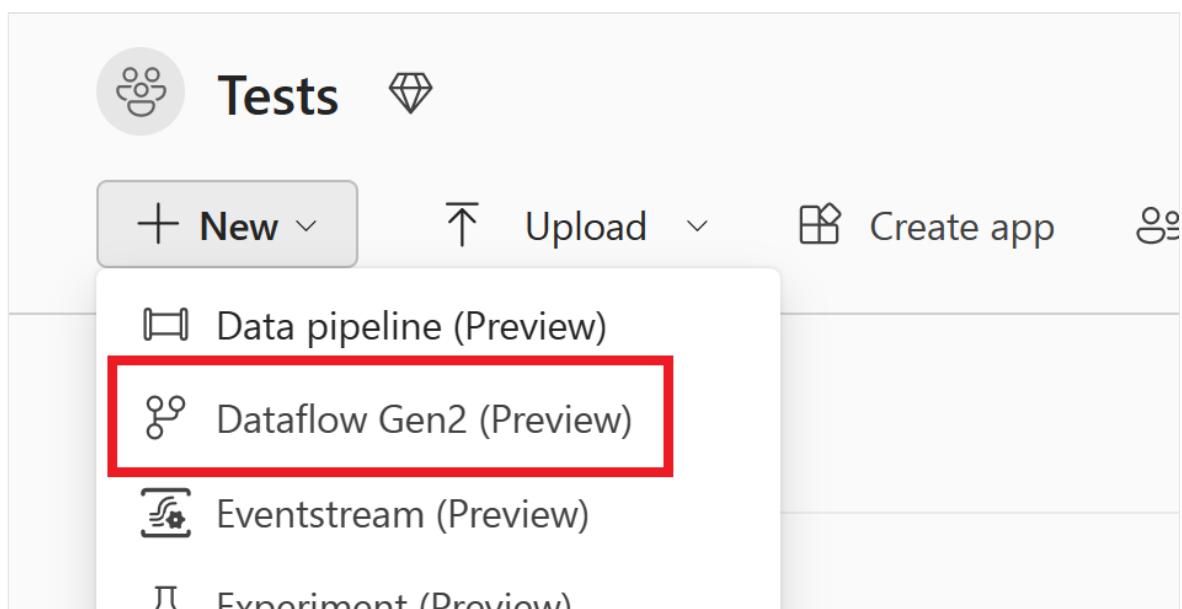
1. From your Power BI or Data Factory workspace, select the ellipsis next to an existing Dataflow Gen1 and then select **Edit**.

	Name	Type	Owner
⋮	Dataflow 10	...	Dataflow
⋮	Dataflow 11	Delete	
⋮	Dataflow 12	Edit	
⋮	Dataflow 13	Export .json	
		Properties	🔍

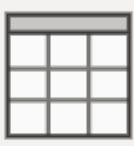
2. In Power Query, select the query or queries you want to copy. If there are multiple queries or folders (called *groups* in Power Query) you want to copy, select **Ctrl** as you select the queries or folders you want to copy. Then either select **Ctrl+C** or right-click in the selection and select **Copy**.



3. Open an existing Dataflow Gen2, or create a new Dataflow Gen2 in Data Factory.
To create a new dataflow, open an existing workflow and select **New > Dataflow Gen2 (Preview)**.



4. In the Power Query editor, select **Get data > Blank query**.

Home**Transform****Add column**Get
data ▾Enter
data

Options

Manage
parameters ▾

Parameters



Excel workbook



Dataflows



SQL Server database



Text/CSV



Web page



OData

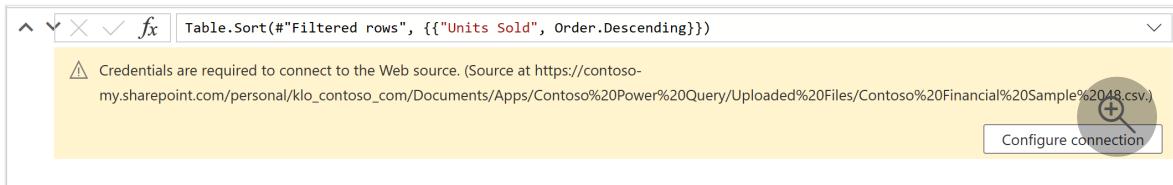


Blank query

More...

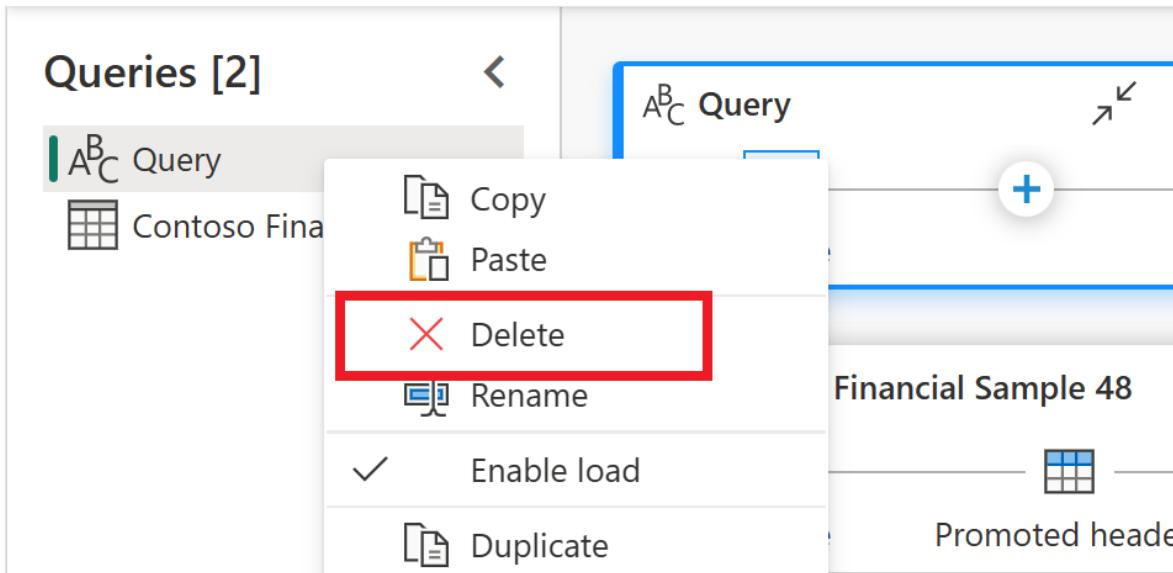
5. In the **Blank query** dialog box, select **Next**.
6. Select inside the **Queries** pane of the Power Query editor, and then select **Ctrl+V** to paste the query.
7. You might need to add your credentials before you can use the pasted queries. If a **Credentials are required to connect to the Web source.** message appears, select

Configure connection, and then enter your credentials.



8. Select **Connect** to connect to your data.

9. Once you've connected to your data, right-click the initial blank query, and then select **Delete** to remove the empty query.



Next steps

- Differences between Dataflow Gen1 and Gen2 in Microsoft Fabric

Use a dataflow in a pipeline

Article • 05/23/2023

In this tutorial, you build a data pipeline to move OData from a Northwind source to a lakehouse destination and send an email notification when the pipeline is completed.

ⓘ 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. Refer to [Azure Data Factory documentation](#) for the service in Azure.

Prerequisites

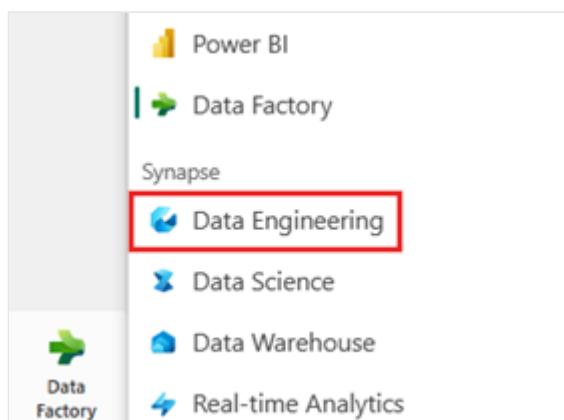
To get started, you must complete the following prerequisites:

- Make sure you have a [Microsoft Fabric enabled Workspace](#) that isn't the default My Workspace.

Create a Lakehouse

To start, you first need to create a lakehouse. A lakehouse is a data lake that is optimized for analytics. In this tutorial, you create a lakehouse that's used as a destination for the dataflow.

1. Switch to the **Data Engineering** experience.



2. Go to your Fabric enabled workspace.

The screenshot shows the Data Factory interface for the 'Sunshine' workspace. On the left, there's a sidebar with icons for Home, Create, Browse, Data hub, Monitoring hub, and a recent items section. The main area is titled 'Workspaces' and contains a search bar. Below it, 'My workspace' is listed. A red box highlights the 'Pinned' section, which contains a card for 'Sunshine' with a small icon and a gear icon. Other pinned items include 'All', 'Sales team', and 'Marketing workspace'. The background shows a light gray grid pattern.

3. Select **Lakehouse** in the create menu.

The screenshot shows the 'New' create menu. At the top are buttons for '+ New' and 'Upload'. Below them is a list of options: 'Lakehouse' (which is highlighted with a red box), 'Notebook', 'Spark Job Definition', and 'Data pipeline'. At the bottom of the list is a 'Show all' link.

4. Enter a **Name** for the lakehouse.

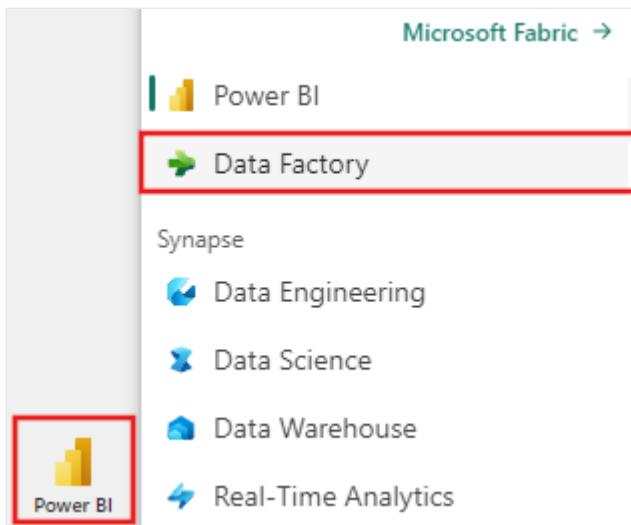
5. Select **Create**.

Now you've created a lakehouse and you can now set up the dataflow.

Create a dataflow

A dataflow is a reusable data transformation that can be used in a pipeline. In this tutorial, you create a dataflow that gets data from an OData source and writes the data to a lakehouse destination.

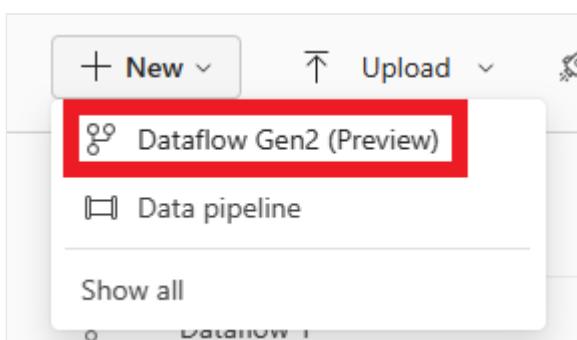
1. Switch to the **Data Factory** experience.



2. Go to your Fabric enabled workspace.

The screenshot shows the 'Data Factory Sunshine' workspace list. On the left is a sidebar with icons for Home, Create, Browse, Data hub, Monitoring hub, and another Data hub. The 'Create' icon is highlighted with a red border. The main area shows a search bar and a list of workspaces. The 'Sunshine' workspace is highlighted with a red border. Other listed workspaces include 'My workspace', 'Sales team', and 'Marketing workspace'.

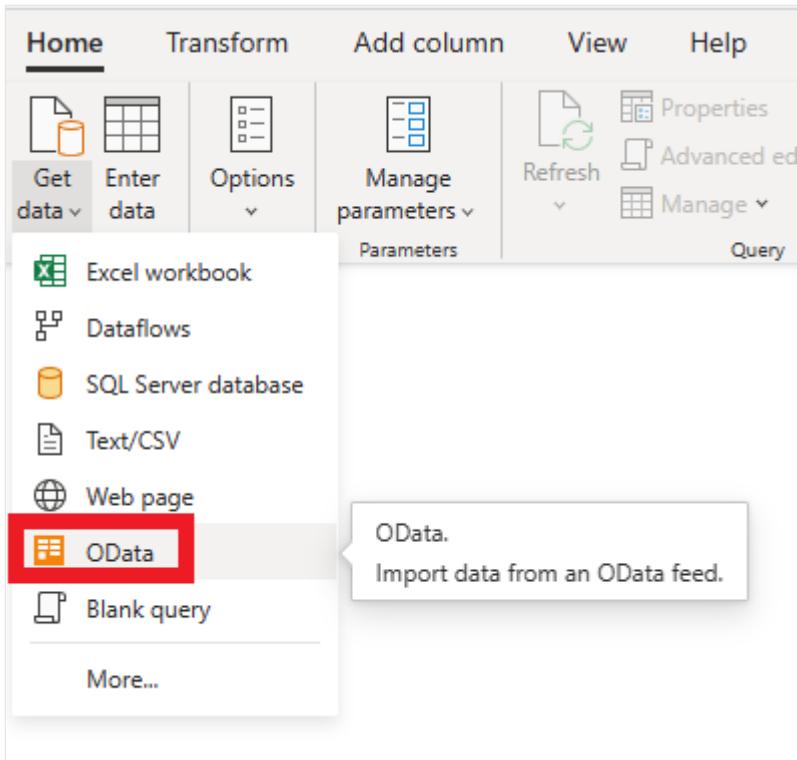
3. Select Dataflow Gen2 in the create menu.



4. Ingest the data from the OData source.

a. Select Get data.

b. Select OData.



c. Enter the URL of the OData source. For this tutorial, use the [OData sample service](#).

d. Select **Next**.

e. Select the **Entity** that you want to ingest. In this tutorial, use the **Orders** entity.

Choose data

Display options ▾

Search

Orders

OrderID	CustomerID	EmployeeID	OrderDate	RequiredDate	ShippedDate	ShipVia	Freight	ShipName
10248	VINET	5	7/4/1996, 12:00:00 AM +00:00	8/1/1996, 12:00:00 AM +00:00	7/16/1996, 12:00:00 AM +00:00	3	32.38	Vins et alcools
10249	TOMSP	6	7/5/1996, 12:00:00 AM +00:00	8/16/1996, 12:00:00 AM +00:00	7/10/1996, 12:00:00 AM +00:00	1	11.61	Toms Spezialitäten
10250	HANAR	4	7/8/1996, 12:00:00 AM +00:00	8/5/1996, 12:00:00 AM +00:00	7/12/1996, 12:00:00 AM +00:00	2	65.83	Hanari Carnes
10251	VICTE	3	7/8/1996, 12:00:00 AM +00:00	8/5/1996, 12:00:00 AM +00:00	7/15/1996, 12:00:00 AM +00:00	1	41.34	Victuailles en stock
10252	SUPRD	4	7/9/1996, 12:00:00 AM +00:00	8/6/1996, 12:00:00 AM +00:00	7/11/1996, 12:00:00 AM +00:00	2	51.3	Suprêmes délices
10253	HANAD	3	7/10/1996, 12:00:00 AM +00:00	7/24/1996, 12:00:00 AM +00:00	7/16/1996, 12:00:00 AM +00:00	2	58.17	Hanari Carnes
10254	CHOPS	5	7/11/1996, 12:00:00 AM +00:00	8/8/1996, 12:00:00 AM +00:00	7/23/1996, 12:00:00 AM +00:00	2	22.98	Chop-suey Chi
10255	RICSU	9	7/12/1996, 12:00:00 AM +00:00	8/9/1996, 12:00:00 AM +00:00	7/15/1996, 12:00:00 AM +00:00	3	148.33	Richter Supermarkt
10256	WELLI	3	7/15/1996, 12:00:00 AM +00:00	8/12/1996, 12:00:00 AM +00:00	7/17/1996, 12:00:00 AM +00:00	2	13.97	Wellington Importadora
10257	HILAA	4	7/16/1996, 12:00:00 AM +00:00	8/13/1996, 12:00:00 AM +00:00	7/22/1996, 12:00:00 AM +00:00	3	81.91	HILARION-Alimentos
10258	ERNSH	7	7/17/1996, 12:00:00 AM +00:00	8/14/1996, 12:00:00 AM +00:00	7/23/1996, 12:00:00 AM +00:00	1	140.51	Ernst Handel
10259	CENTC	4	7/18/1996, 12:00:00 AM +00:00	8/15/1996, 12:00:00 AM +00:00	7/25/1996, 12:00:00 AM +00:00	3	3.25	Centro comercial
10260	OTTIK	4	7/19/1996, 12:00:00 AM +00:00	8/16/1996, 12:00:00 AM +00:00	7/29/1996, 12:00:00 AM +00:00	1	55.09	Ottiles Kaselac
10261	QUEDE	4	7/19/1996, 12:00:00 AM +00:00	8/16/1996, 12:00:00 AM +00:00	7/30/1996, 12:00:00 AM +00:00	2	3.05	Que Delícia
10262	RATTG	8	7/22/1996, 12:00:00 AM +00:00	8/12/1996, 12:00:00 AM +00:00	7/23/1996, 12:00:00 AM +00:00	3	48.29	Rattlersnake
10263	ERNSH	9	7/23/1996, 12:00:00 AM +00:00	8/20/1996, 12:00:00 AM +00:00	7/31/1996, 12:00:00 AM +00:00	3	146.06	Ernst Handel
10264	FOLKO	6	7/24/1996, 12:00:00 AM +00:00	8/21/1996, 12:00:00 AM +00:00	8/23/1996, 12:00:00 AM +00:00	3	3.67	Folk och fä HB
10265	BLONP	2	7/25/1996, 12:00:00 AM +00:00	8/22/1996, 12:00:00 AM +00:00	7/21/1996, 12:00:00 AM +00:00	1	55.28	Biondel père et fils
10266	WARTH	3	7/26/1996, 12:00:00 AM +00:00	9/6/1996, 12:00:00 AM +00:00	7/31/1996, 12:00:00 AM +00:00	3	25.73	Wartian Herkis
10267	FRANK	4	7/29/1996, 12:00:00 AM +00:00	8/26/1996, 12:00:00 AM +00:00	8/6/1996, 12:00:00 AM +00:00	1	208.50	Frankenvansia
10268	GROSR	8	7/30/1996, 12:00:00 AM +00:00	8/27/1996, 12:00:00 AM +00:00	8/2/1996, 12:00:00 AM +00:00	3	66.29	GROSELLA-Reserva
10269	WHITC	5	7/31/1996, 12:00:00 AM +00:00	8/14/1996, 12:00:00 AM +00:00	8/9/1996, 12:00:00 AM +00:00	1	4.56	White Clover
10270	WARTH	1	8/1/1996, 12:00:00 AM +00:00	8/29/1996, 12:00:00 AM +00:00	8/2/1996, 12:00:00 AM +00:00	1	136.54	Wartian Herkis
10271	SPLIR	6	8/1/1996, 12:00:00 AM +00:00	8/29/1996, 12:00:00 AM +00:00	8/30/1996, 12:00:00 AM +00:00	2	4.54	Split Rail Beer
10272	RATTG	6	8/2/1996, 12:00:00 AM +00:00	8/30/1996, 12:00:00 AM +00:00	8/6/1996, 12:00:00 AM +00:00	2	98.03	Rattlersnake
10273	QUICK	3	8/5/1996, 12:00:00 AM +00:00	9/2/1996, 12:00:00 AM +00:00	8/1/1996, 12:00:00 AM +00:00	3	76.07	QUICK-Stop
10274	VINET	6	8/6/1996, 12:00:00 AM +00:00	9/3/1996, 12:00:00 AM +00:00	8/16/1996, 12:00:00 AM +00:00	1	6.01	Vins et alcools
10275	MAGAA	1	8/7/1996, 12:00:00 AM +00:00	9/4/1996, 12:00:00 AM +00:00	8/9/1996, 12:00:00 AM +00:00	1	26.93	Magazzini Alim.
10276	TORTU	8	8/8/1996, 12:00:00 AM +00:00	8/22/1996, 12:00:00 AM +00:00	8/14/1996, 12:00:00 AM +00:00	3	13.84	Tortuga Restau
10277	MORGK	2	8/9/1996, 12:00:00 AM +00:00	9/6/1996, 12:00:00 AM +00:00	8/13/1996, 12:00:00 AM +00:00	3	125.77	Morgenstern G
10278	BERGS	8	8/12/1996, 12:00:00 AM +00:00	9/9/1996, 12:00:00 AM +00:00	8/16/1996, 12:00:00 AM +00:00	2	92.69	Berglunds snickerier
10279	LEHMS	8	8/13/1996, 12:00:00 AM +00:00	9/10/1996, 12:00:00 AM +00:00	8/16/1996, 12:00:00 AM +00:00	2	25.83	Lehmans Mar
10280	BERGS	2	8/14/1996, 12:00:00 AM +00:00	9/11/1996, 12:00:00 AM +00:00	9/12/1996, 12:00:00 AM +00:00	1	6.98	Berglunds snickerier
10281	ROMEY	4	8/14/1996, 12:00:00 AM +00:00	8/28/1996, 12:00:00 AM +00:00	8/21/1996, 12:00:00 AM +00:00	1	2.94	Romeo y tomate
10282	ROMEY	4	8/15/1996, 12:00:00 AM +00:00	9/12/1996, 12:00:00 AM +00:00	8/21/1996, 12:00:00 AM +00:00	1	12.69	Romeo y tomate
10283	LILAS	3	8/16/1996, 12:00:00 AM +00:00	9/13/1996, 12:00:00 AM +00:00	8/23/1996, 12:00:00 AM +00:00	3	84.81	LILA-Supermercado
10284	LEHMS	4	8/19/1996, 12:00:00 AM +00:00	9/16/1996, 12:00:00 AM +00:00	8/27/1996, 12:00:00 AM +00:00	1	76.56	Lehmans Mar
10285	QUICK	1	8/20/1996, 12:00:00 AM +00:00	9/17/1996, 12:00:00 AM +00:00	8/26/1996, 12:00:00 AM +00:00	2	76.63	QUICK-Stop
10286	QUICK	8	8/21/1996, 12:00:00 AM +00:00	9/18/1996, 12:00:00 AM +00:00	8/30/1996, 12:00:00 AM +00:00	3	229.24	QUICK-Stop
10287	RICAR	8	8/22/1996, 12:00:00 AM +00:00	9/19/1996, 12:00:00 AM +00:00	8/28/1996, 12:00:00 AM +00:00	3	12.76	Ricardo Adocic
10288	REGGC	4	8/23/1996, 12:00:00 AM +00:00	9/20/1996, 12:00:00 AM +00:00	8/23/1996, 12:00:00 AM +00:00	1	7.45	Reggiani Casei
10289	BSEBV	7	8/26/1996, 12:00:00 AM +00:00	9/23/1996, 12:00:00 AM +00:00	8/28/1996, 12:00:00 AM +00:00	3	22.77	B's Beverages
10290	COLWI	4	8/27/1996, 12:00:00 AM +00:00	9/24/1996, 12:00:00 AM +00:00	8/29/1996, 12:00:00 AM +00:00	4	10.00	Colwi

Select related tables

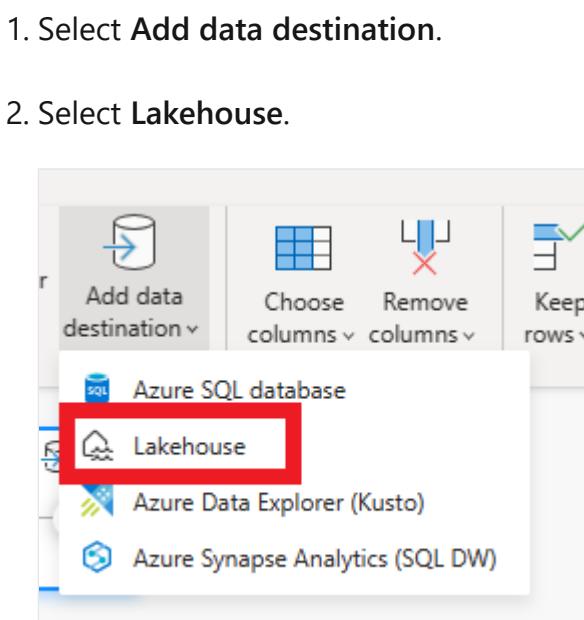
Back

Create

f. Select Create.

Now that you've ingested the data from the OData source, you can set up the lakehouse destination.

To ingest the data to the lakehouse destination:



3. Configure the connection you want to use to connect to the lakehouse. The default settings are fine.

4. Select **Next**.

5. Navigate to the workspace where you created the lakehouse.

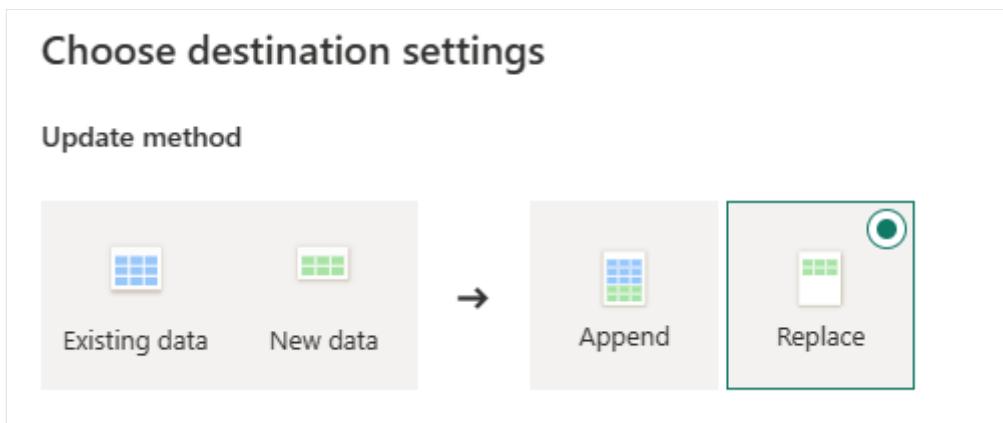
6. Select the lakehouse that you created in the previous steps.



7. Confirm the table name.

8. Select **Next**.

9. Confirm the update method and select **Save settings**.



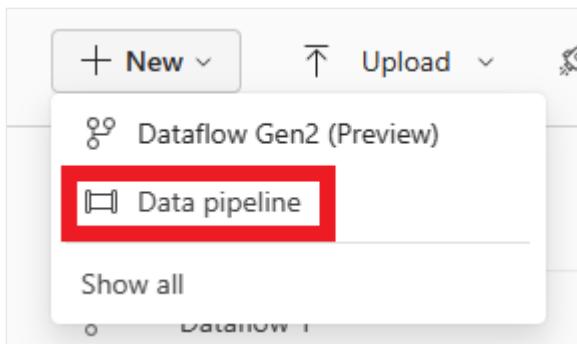
10. Publish the dataflow.

Now that you've ingested the data to the lakehouse destination, you can set up your data pipeline.

Create a data pipeline

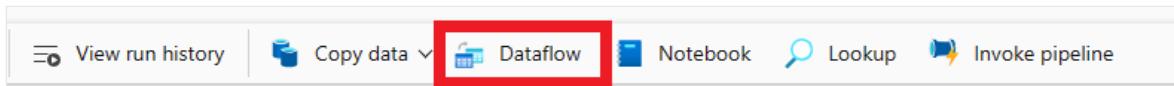
A data pipeline is a workflow that can be used to automate data processing. In this tutorial, you create a data pipeline that runs the Dataflow Gen2 that you created in the previous procedure.

1. Navigate back to the workspace overview page and select **Data Pipelines** in the create menu.



2. Provide a **Name** for the data pipeline.

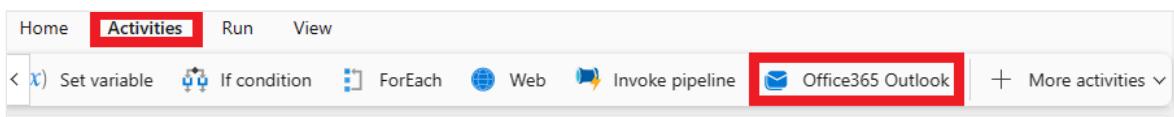
3. Select the **Dataflow** activity.



4. Select the **Dataflow** that you created in the previous procedure in the **Dataflow** dropdown list under **Settings**.

A screenshot of the Azure portal showing the 'Dataflow' settings page. At the top, there is a 'Dataflow' section containing a list with one item 'Dataflow1'. Below this is a toolbar with icons for 'General' and 'Settings' (which is underlined). In the 'General' section, there is a 'Dataflow *' field with a dropdown menu open. The dropdown menu shows three options: 'My new dataflow' (highlighted with a green box), 'Another dataflow', and 'Sunshine dataflow'. To the right of the dropdown is a 'Refresh' button and a radio button for 'No notification' which is selected.

5. Add an **Email notification** activity.



6. Configure the **Email notification** activity.

a. Authenticate with your Office 365 account.

b. Select the **Email address** that you want to send the notification to.

c. Enter a **Subject** for the email.

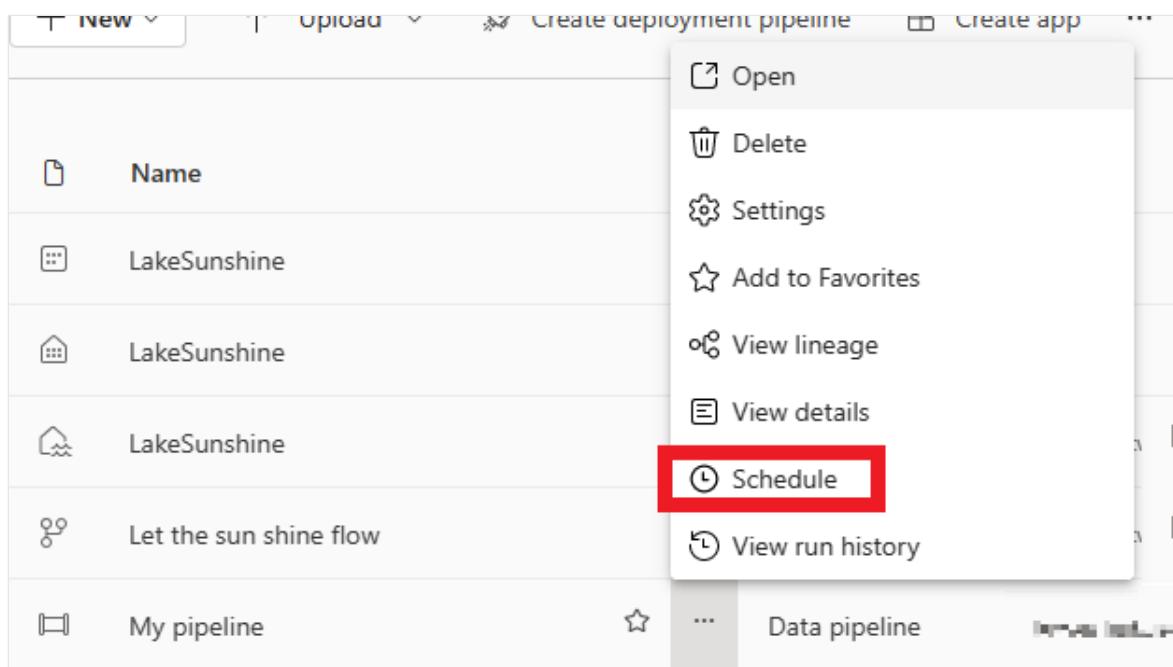
d. Enter a **Body** for the email.

The screenshot shows the 'Settings' tab selected in a mail compose interface. At the top, there is a 'Reauthenticate' button. Below it, there are fields for 'To' (with a red asterisk), 'Subject' (with a red asterisk), and 'Body' (with a red asterisk). The 'Body' field is a large text area with a toolbar containing bold, italic, underline, and other rich text options. Below the body area, there is a section titled 'Advanced properties' with dropdown menus for 'From (Send as)', 'CC', 'BCC', 'Sensitivity', 'Reply To', and 'Importance' (set to 'Normal').

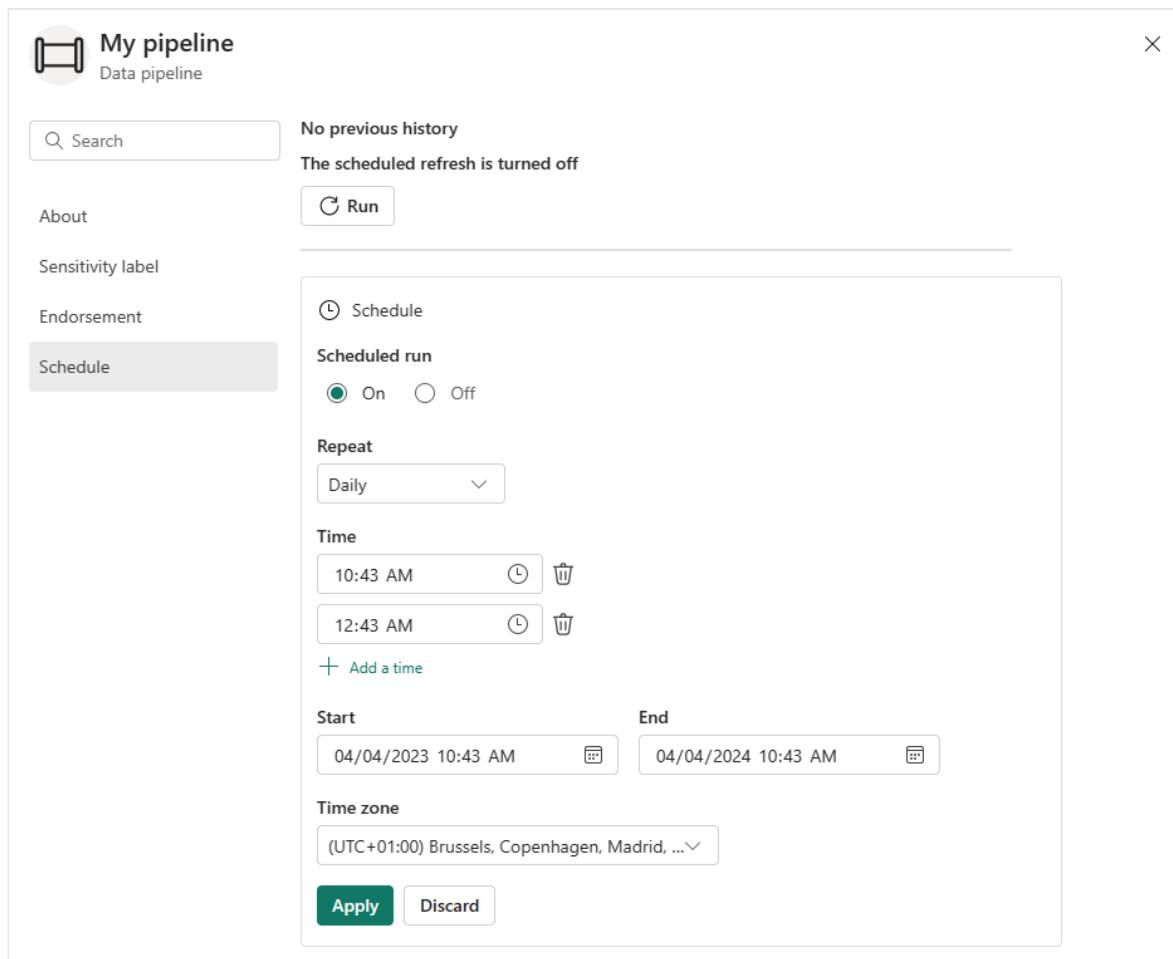
Run and schedule the data pipeline

In this section, you run and schedule the data pipeline. This schedule allows you to run the data pipeline on a schedule.

1. Go to your workspace.
2. Open the dropdown menu of the data pipeline that you created in the previous procedure, and then select **Schedule**.



3. In Scheduled run, select On.



4. Provide the schedule you want to use to run the data pipeline.

- Repeat, for example, every **Day** or every **Minute**.
- When selected **Daily**, you can also select the **Time**.
- Start **On** a specific **Date**.
- End **On** a specific **Date**.

e. Select the **Timezone**.

5. Select **Apply** to apply the changes.

You've now created a data pipeline that runs on a schedule, refreshes the data in the lakehouse, and sends you an email notification. You can check the status of the data pipeline by going to the **Monitor Hub**. You can also check the status of the data pipeline by going to **Data Pipeline** and selecting the **Run history** tab in the dropdown menu.

Next steps

This sample shows you how to use a dataflow in a pipeline with Data Factory in Microsoft Fabric. You learned how to:

- ✓ Create a dataflow.
- ✓ Create a pipeline invoking your dataflow.
- ✓ Run and schedule your data pipeline.

Next, advance to learn more about monitoring your pipeline runs.

[How to monitor pipeline runs in Microsoft Fabric](#)

Getting from Azure Data Factory to Data Factory in Microsoft Fabric

Article • 05/23/2023

Data Factory in Microsoft Fabric is the next generation of Azure Data Factory which provides cloud-scale data movement and data transformation services that allow you to solve the most complex ETL scenarios. It's intended to make your experience easy to use, powerful, and truly enterprise-grade. This article compares the differences between Azure Data Factory and Data Factory in Microsoft Fabric.

ⓘ 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.

Feature mapping

In the modern experience of Data Factory in Fabric, there are some different features concepts compared to Azure Data Factory. Detail features mapping is presented as the table below.

Azure Data Factory	Data Factory in Fabric	Description
Pipeline	Data pipeline	Data pipeline in Fabric is better integrated with the unified data platform including Lakehouse, Datawarehouse, and more.
Mapping dataflow	Dataflow Gen2	Dataflow Gen2 provides easier experience to build transformation. We are in progress of letting more functions of mapping dataflow supported in Dataflow Gen2
Activities	Activities	We are in progress to make more activities of ADF supported in Data Factory in Fabric. Data Factory in Fabric also has some newly attracted activities like Office 365 Outlook activity. Details are in Activity overview .
Dataset	Not Applicable	Data Factory in Fabric doesn't have dataset concepts. Connection will be used for connecting each data source and pull data.

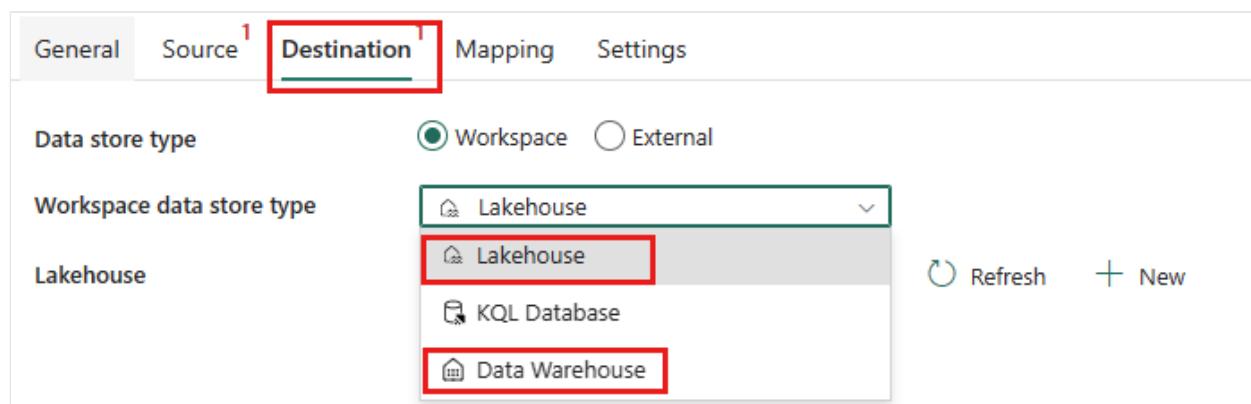
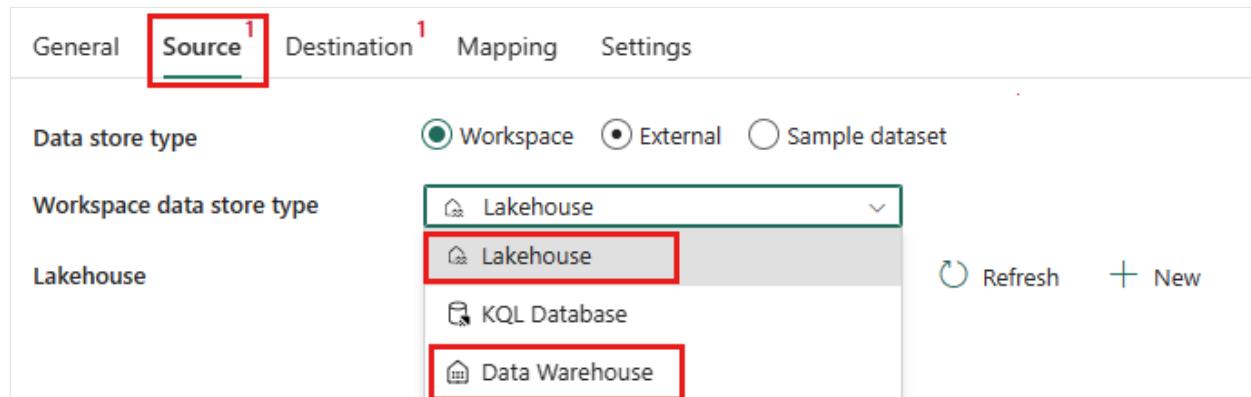
Azure Data Factory	Data Factory in Fabric	Description
Linked Service	Connections	Connections have similar functionality as linked service, but connections in Fabric have more intuitive way to create.
Triggers	Schedules (other triggers are in progress)	Fabric can use the schedule to automatically run pipeline. We are adding more triggers supported by ADF in Microsoft Fabric.
Publish	Save, Run	For pipeline in Fabric, you don't need to publish to save the content. Instead, you can use Save button to save the content directly. When you click Run button, it will save the content before running pipeline.
Autoresolve and Azure Integration runtime	Not Applicable	In Fabric, we don't have the concept of Integration runtime.
Self-hosted integration runtimes	On-premises Data Gateway(in design)	The capability in Fabric is still in progress of design.
Azure-SSIS integration runtimes	To be determined	The capability in Fabric hasn't confirmed the roadmap and design.
MVNet and Private End Point	To be determined	The capability in Fabric hasn't confirmed the roadmap and design.
Expression language	Expression language	Expression language is similar in ADF and Fabric.
Authentication type in linked service	Authentication kind in connection	Authentication kind in Fabric pipeline already supported popular authentication types in ADF, and more authentication kinds will be added.
CI/CD	CI/CD	CI/CD capability in Fabric Data Factory will be coming soon.
Export and Import ARM	Save as	Save as is available in Fabric pipeline to duplicate a pipeline.
Monitoring	Monitoring, Run history	The monitoring hub in Fabric has more advanced functions and modern experience like monitoring across different workspaces for better insights.

Data pipeline of data factory in Microsoft Fabric

There are many exciting features of data pipeline in Data Factory of Microsoft Fabric. Leveraging these features, you can feel the power of pipeline in Fabric.

Lakehouse/Datawarehouse integration

Lakehouse and Data Warehouse are available as source and destination in Pipeline of Fabric, so it's extremely convenient for you to build your own projects integrated with Lakehouse and Datawarehouse.



Office 365 outlook activity

Office 365 outlook activity provides an intuitive and simple way to send customized email notification about info of pipeline and activity, and output of pipeline by easy configuration.

The screenshot shows the Microsoft Fabric Data Factory interface. At the top, there's a navigation bar with 'Home' (underlined), 'Activities', 'Run', and 'View'. Below the navigation bar is a toolbar with icons for file operations, validation, scheduling, run history, copy data, and dataflow. A red box highlights the 'Run' button, and a red arrow points to it from the top right. To the right of the toolbar, there's a 'Copy data' activity card for 'Copy_yyg' and a 'Schedule' card for 'Office 365 Outlook (Preview)'. The 'Office 365 Outlook (Preview)' card shows a connection named 'Office 365 Outlook1' with a green checkmark. Below these cards, the 'Settings' tab is selected, showing fields for 'To', 'Subject', and 'Body', along with a rich text editor toolbar. At the bottom left, there's a link to 'Advanced properties'.

Get data experience

A modern and easy Get Data experience is provided in Data Factory in Fabric, so it's super-fast for you to set up your copy pipeline and create a new connection.

Copy data

Choose data source
Select a connector. Then enter the connection information.

Connect to data source
 Choose data destination
 Connect to data destination
 Review + save

Build your data ingestion task to move objects from a data source to a data destination. [Learn more](#)

Data sources

All categories Workspace Azure Database File Search

Amazon RDS for SQL Server Database	Amazon Redshift Database
Amazon S3 File	Amazon S3 Compatible File
Apache Impala Database	Azure Blob Storage Azure
Azure Cosmos DB for NoSQL Azure	Azure Data Explorer (Kusto) Azure
Azure Data Lake Storage Gen1 Azure	Azure Data Lake Storage Gen2 Azure
Azure Database for PostgreSQL Azure	Azure SQL Database Azure
Azure SQL Database Managed Instance Azure	Azure Synapse Analytics Azure

Back Next Cancel

Copy data

Choose data source
Select a connector. Then enter the connection information.

Connect to data source
 Choose data destination
 Connect to data destination
 Review + save

Azure Blob Storage [Learn more](#)

Existing connection Create new connection

Connection settings

Account name or URL *
Example: testazurestorageacct

Connection credentials

Connection: Create new connection ⟳
Connection name: Connection
Authentication kind: Anonymous
Anonymous
Shared Access Signature (SAS)
Organizational account
Service principal

Back Next Cancel

Modern monitoring experience

With the combined capabilities of the monitoring hub and the items of Data Factory, such as data flows and data pipelines, we can get a full view of all the workloads and drill into any activity within a data factory experience. It's also convenient for you to do the cross-workspace analysis through monitoring hub.

The screenshot shows the Azure Data Lake Storage Gen2 Monitoring hub. On the left, there's a sidebar with navigation links: Home, Create, Browse, OneLake data hub, Monitoring hub (which is selected and highlighted in green), Workspaces, Bug Bash Tests, and pipeline1. The main area is titled "Monitoring hub" and contains a brief description: "Monitoring hub is a station to view and track active activities across different products." Below this is a search bar with "Refresh" and "Filter by keyword" buttons, and a "Filter (4)" dropdown. A "Completed" filter is applied. The main content is a table with columns: Name, Status, Item type, Start time, Duration, Submitter, and Location. The table lists eight completed pipeline runs, each with a green circular status icon and a "Data pipeline" item type. The "Start time" column shows various times from 2:50 AM to 10:31 AM. The "Duration" column is collapsed. The "Submitter" column shows icons representing different users. The "Location" column is also collapsed.

The pipeline copy monitoring results provides breakdown detail of copy activity. By selecting the run details button (with the glasses icon highlighted) to view the run details. Expand the **Duration breakdown**, you can know the time duration of each stage in copy activity.

The screenshot shows the "Activity runs" section. At the top, there's a green header box labeled "Copy data" with a checkmark icon. Below it is a sub-box labeled "Copy_n25" with a blue cylinder icon. To the right of the sub-box are four small icons: a circular arrow, a checkmark, a red X, and a right-pointing arrow. The main table has columns: "Activity name" (sorted by name), "Status" (sorted by status), and "Activity type". There is one item listed: "Copy_n25" with a status of "Succeeded" (indicated by a green checkmark icon) and an activity type of "Copy data". The "Status" column has a sorting arrow pointing up. The "Activity name" and "Activity type" columns have sorting arrows pointing up. A "Details" button is located at the bottom of the row for "Copy_n25". A red box highlights the "Details" button. The table also includes a "Showing 1 - 1 items" message and a "All status" dropdown menu.

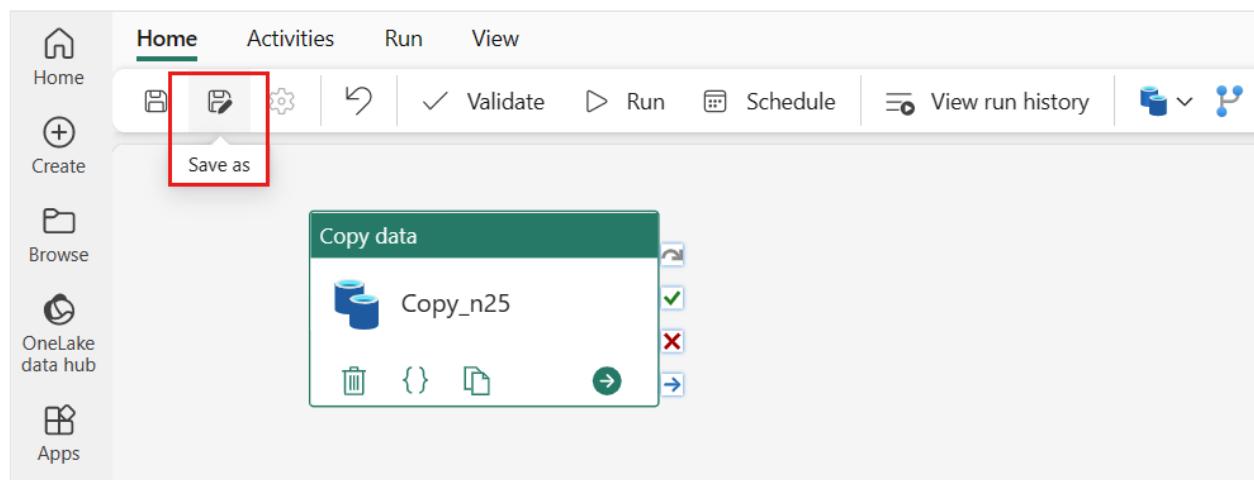
Copy data details

Status	JULLECUU
Start time	4/26/2023, 4:22:58 PM
Pipeline run activity ID	< Pipeline run activity ID >
Throughput	< Throughput >
Total duration	00:00:56
Duration breakdown	
Start time	4/26/2023, 4:22:59 PM
Used DIUs ⁽¹⁾	4
Used parallel copies ⁽¹⁾	1
Queue	
Transfer	
Reading from...	
Writing to sink	

> Advanced

Save as

Save as in Fabric pipeline provides a convenient way for you to duplicate an existing pipeline for other development purposes.



Next steps

- Differences between Dataflow Gen1 Gen2
- Build your first data integration

Getting from Dataflow Generation 1 to Dataflow Generation 2

Article • 05/23/2023

Dataflow Gen2 is the new generation of dataflows. The new generation of dataflows resides alongside the Power BI Dataflow (Gen1) and brings new features and improved experiences. The following section provides a comparison between Dataflow Gen1 and Dataflow Gen2.

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.

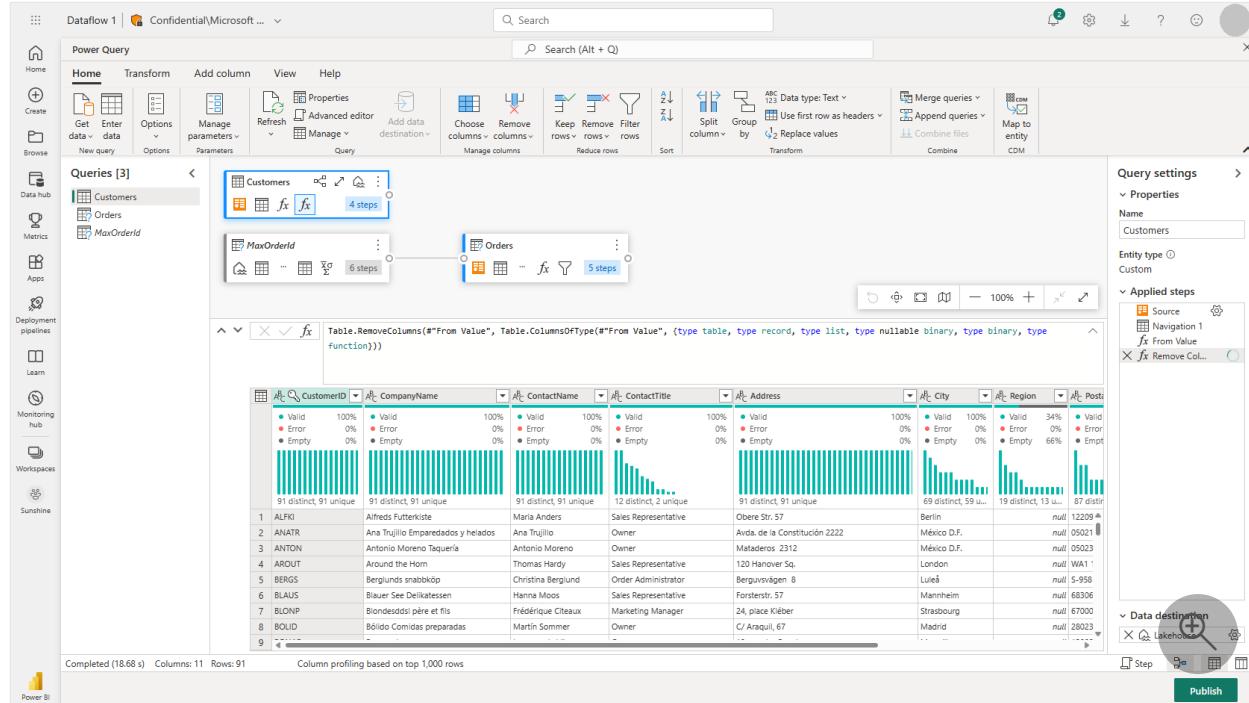
Refer to [Azure Data Factory documentation](#) for the service in Azure.

Feature overview

Feature	Dataflow Gen2	Dataflow Gen1
Author dataflows with Power Query	✓	✓
Shorter authoring flow	✓	
Auto-Save and background publishing	✓	
Output destinations	✓	
Improved monitoring and refresh history	✓	
Integration with data pipelines	✓	
High-scale compute	✓	
Get Data via Dataflows connector	✓	✓
Direct Query via Dataflows connector		✓
Incremental refresh		✓
AI Insights support		✓

Shorter authoring experience

Working with Dataflow Gen2 feels like coming home. We kept the full Power Query experience you're used to in Power BI dataflows. When you enter the experience, you're guided step-by-step for getting the data into your dataflow. We also shorten the authoring experience to reduce the number of steps required to create dataflows, and added a few new features to make your experience even better.



New dataflow save experience

With Dataflow Gen2, we changed how saving a dataflow works. Any changes made to a dataflow are autosaved to the cloud. So you can exit the authoring experience at any point and continue from where you left off at a later time. Once you're done authoring your dataflow, you publish your changes and those changes are used when the dataflow refreshes. In addition, publishing the dataflow saves your changes and runs validations that must be performed in the background. This feature lets you save your dataflow without having to wait for validation to finish.

To learn more about the new save experience, go to [Save a draft of your dataflow](#).

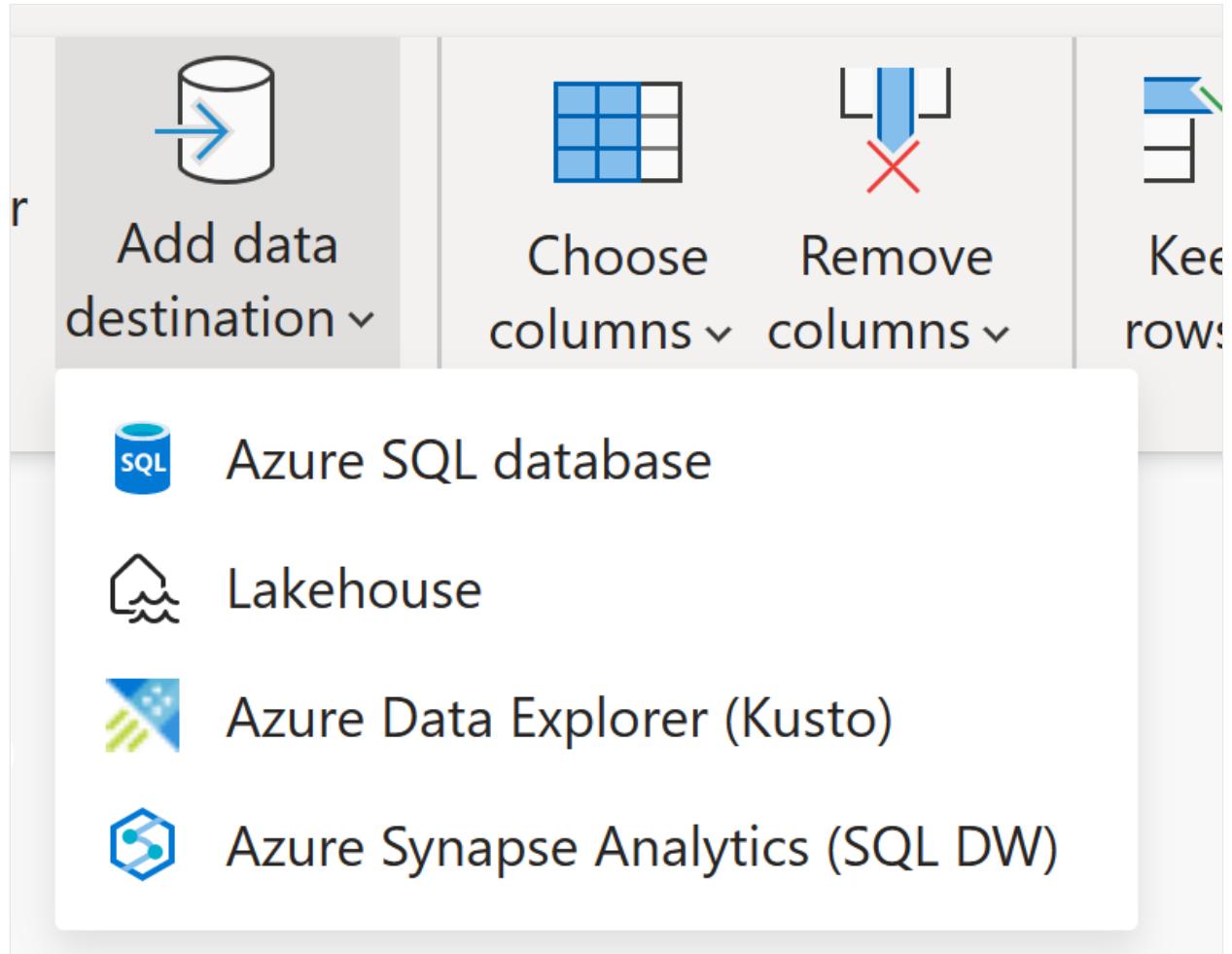
Output destinations

Similar to Dataflow Gen1, Dataflow Gen2 allows you to transform your data into dataflow's internal/staging storage where it can be accessed using the Dataflow connector. Dataflow Gen2 also allows you to specify an output destination for your data. Using this feature, you can now separate your ETL logic and destination storage. This feature benefits you in many ways. For example, you can now use a dataflow to load data into a lakehouse and then use a notebook to analyze the data. Or you can use a

dataflow to load data into an Azure SQL database and then use a data pipeline to load the data into a data warehouse.

In Dataflow Gen2, we added support for the following destinations and many more are coming soon:

- Fabric Lakehouse
- Azure Data Explorer (Kusto)
- Azure Synapse Analytics (SQL DW)
- Azure SQL Database



New refresh history and monitoring

With Dataflow Gen2, we introduce a new way for you to monitor your dataflow refreshes. We integrate support for [Monitoring Hub](#) and give our [Refresh History](#) experience a major upgrade.

[← Details](#)

Dataflow 1 > 2/9/2023, 8:26:42 PM

Status	Type
Succeeded	On demand
Start time	End time
2/9/2023, 8:26:42 PM	2/9/2023, 8:28:14 PM
Duration	Request ID
00:01:32	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
Session ID	Dataflow ID
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx	xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx

Tables

Name	Status
Customers	Succeeded
Orders	Succeeded

Activities

Name	Status
WriteToDatabaseTableFrom_TransformForOutputToDatabaseTableFrom_Customers	Succeeded
WriteToDatabaseTableFrom_TransformForOutputToDatabaseTableFrom_Orders	Succeeded

[Edit dataflow](#)

Integration with data pipelines

Data pipelines allow you to group activities that together perform a task. An activity is a unit of work that can be executed. For example, an activity can copy data from one location to another, run a SQL query, execute a stored procedure, or run a Python notebook.

A pipeline can contain one or more activities that are connected by dependencies. For example, you can use a pipeline to ingest and clean data from an Azure blob, and then kick off a Dataflow Gen2 to analyze the log data. You can also use a pipeline to copy data from an Azure blob to an Azure SQL database, and then run a stored procedure on the database.

The screenshot shows the Dataflow Gen2 interface. At the top, there's a green header bar with the word "Dataflow". Below it is a toolbar with icons for refresh, save, and other operations. The main area has two tabs: "General" (which is selected) and "Settings". Under "General", there are fields for "Name" (containing "Refresh my Dataflow" with a "Learn more" link), "Description" (an empty text area), "Timeout" (set to "0.12:00:00"), and "Retry" (set to "0"). There's also a "Advanced" section indicated by a small arrow icon.

Save as draft

With Dataflow Gen2, we introduce a worry free experience by removing the need for publishing to save your changes. With save as draft functionality, we store a draft version of your dataflow every time you make a change. Did you lose internet connectivity? Did you accidentally close your browser? No worries; we got your back. Once you return to your dataflow, your recent changes are still there and you can continue where you left off. This is a seamless experience and doesn't require any input from you. This allows you to work on your dataflow without having to worry about losing your changes or having to fix all the query errors before you can save your changes. To learn more about this feature, go to [Save a draft of your dataflow](#).

High scale compute

Similar to Dataflow Gen1, Dataflow Gen2 also features an enhanced compute engine to improve performance of both transformations of referenced queries and get data scenarios. To achieve this, Dataflow Gen2 creates both Lakehouse and Warehouse items in your workspace, and uses them to store and access data to improve performance for all your dataflows.

Try out Dataflow Gen2 by reusing your queries from Dataflow Gen1

You probably have many Dataflow Gen1 queries and you're wondering how you can try them out in Dataflow Gen2. We have a few options for you to recreate your Gen1 dataflows as Dataflow Gen2.

- Export your Dataflow Gen1 queries and import them into Dataflow Gen2

You can now export queries in both the Dataflow Gen1 and Gen2 authoring experiences and save them to a PQT file you can then import into Dataflow Gen2.

For more information, go to [Use the export template feature](#).

- Copy and paste in Power Query

If you have a dataflow in Power BI or Power Apps, you can copy your queries and paste them in the editor of your Dataflow Gen2. This functionality allows you to migrate your dataflow to Gen2 without having to rewrite your queries. For more information, go to [Copy and paste existing Dataflow Gen1 queries](#).

Next steps

- [Dataflows refresh history and monitoring](#)
- [Dataflows save as draft](#)
- [Move queries from Dataflow Gen1 to Dataflow Gen2](#)