

Data Factory in a Day with Microsoft Fabric



Microsoft Fabric overview

Microsoft Fabric overview

Microsoft Fabric enables you to manage your data in one place with a suite of analytics experiences that work together seamlessly, including:

- Data Factory
- Synapse Data Engineering
- Synapse Data Warehouse
- Synapse Data Science
- Synapse Real-Time Intelligence
- Power BI

Microsoft Fabric does it all—in a unified solution

An end-to-end analytics platform that brings together all the data and analytics tools that organizations need to go from the data lake to the business user



Data Integration

Data Factory



Data Engineering

Synapse



Data Warehouse

Synapse



Data Science

Synapse



Real-Time Intelligence

Synapse



Business Intelligence

Power BI



Observability

Data Activator



Unified data foundation

OneLake

UNIFIED

SaaS product experience

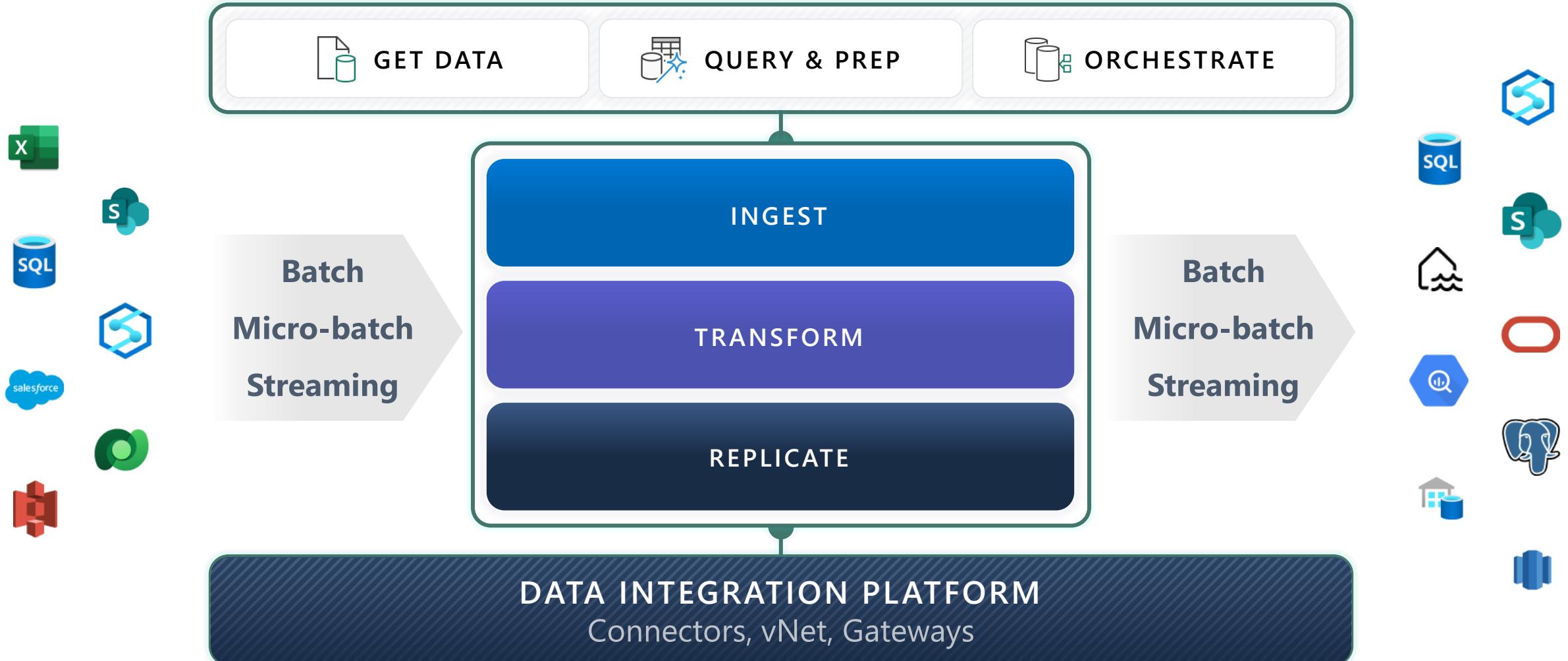
Security and governance

Compute and storage

Business model

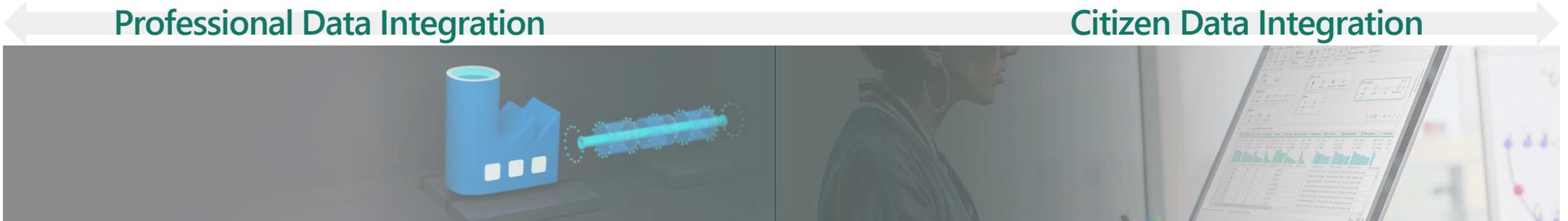
Data Factory overview

Microsoft Data Integration



Microsoft Data Integration

Products



Azure Data Factory, Azure Synapse Analytics, SQL Server Integration Services

- Fully managed, with serverless data integration services
- Visually integrate data sources with more than 100 built-in connectors
- Easily construct ETL and ELT processes code-free

Citizen Data Integration

Power Query

- Seamlessly integrated into many popular Microsoft products
- An easy to use, engaging, no-code experience
- Includes powerful and smart AI-based data preparation

Microsoft Data Integration

Unified Product Portfolio

Professional & Citizen Data Integration

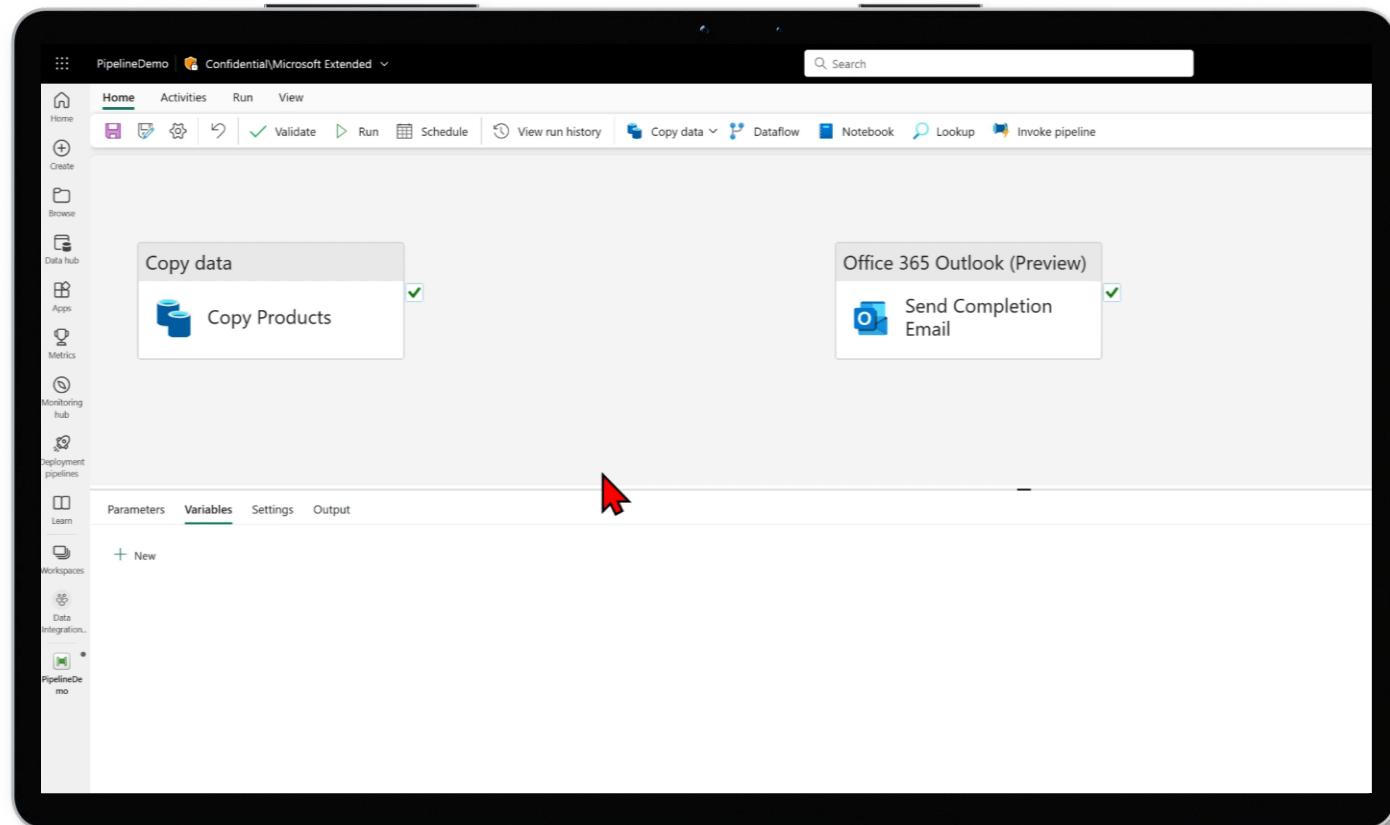
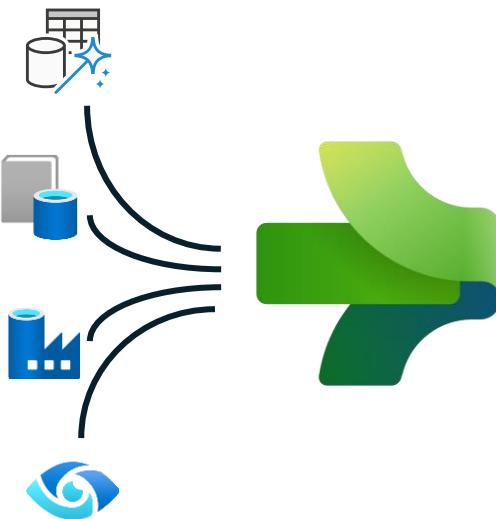


Data Factory in Microsoft Fabric

- Brings together the best of Power Query and Azure Data Factory, into a modern data integration experience
- Empowers both professional and citizen developers
- Ingest and transform data as well as orchestrate data workflows
- Data Factory enables everyone to connect to diverse data sources and to bring that data to where it can best help derive insights for better business decisions.

Data Factory in Microsoft Fabric

Data Factory converges our data capabilities into a single SaaS interface to provide the world's most complete data integration experience.

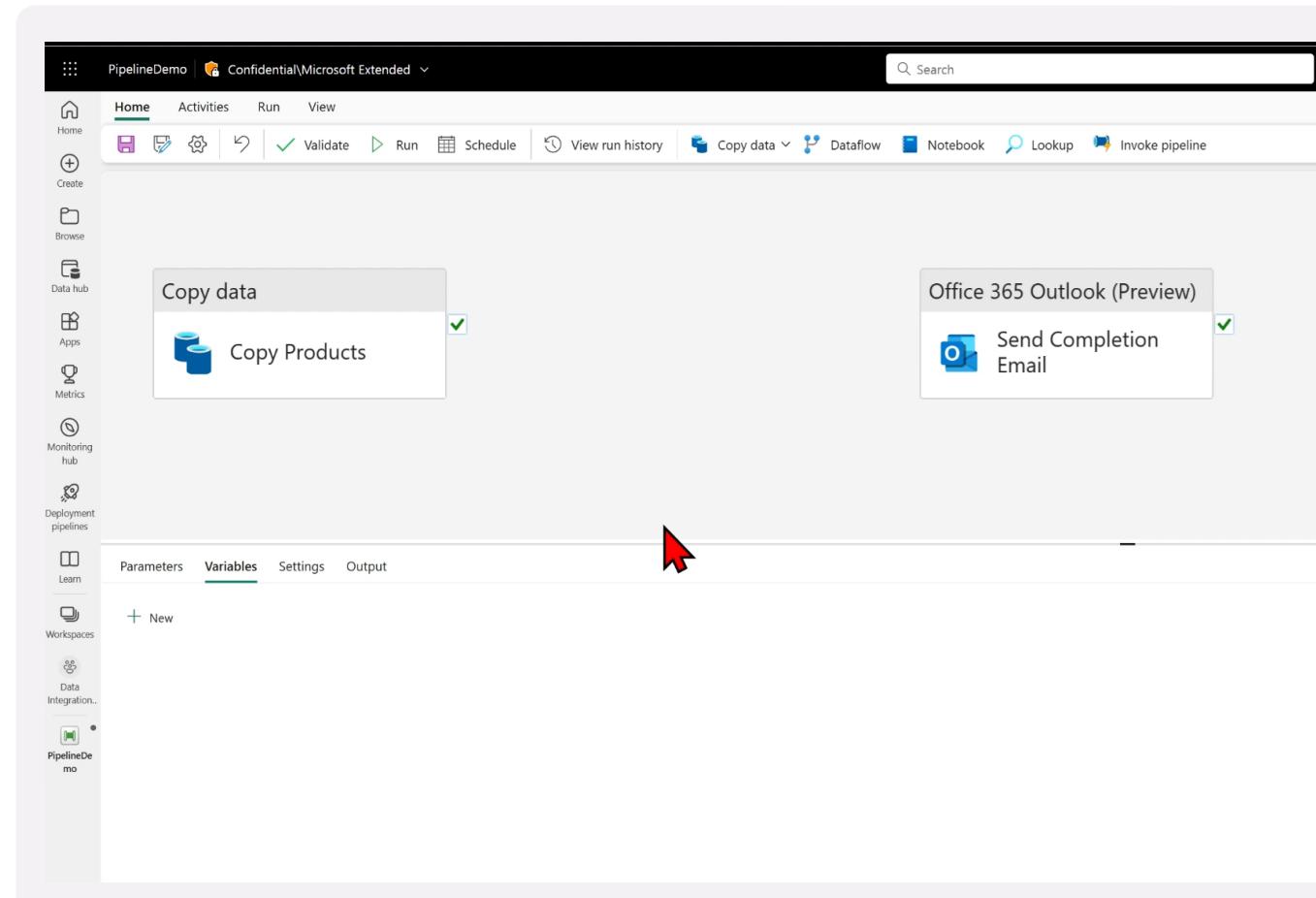


Data Integration for everyone

Empowering **every person**
to integrate data

From **citizen** to **professional** developers

Enterprise-scale data ingestion,
transformation, orchestration with
Dataflows Gen2 and **Data pipelines,**
Mirroring, and more to come..



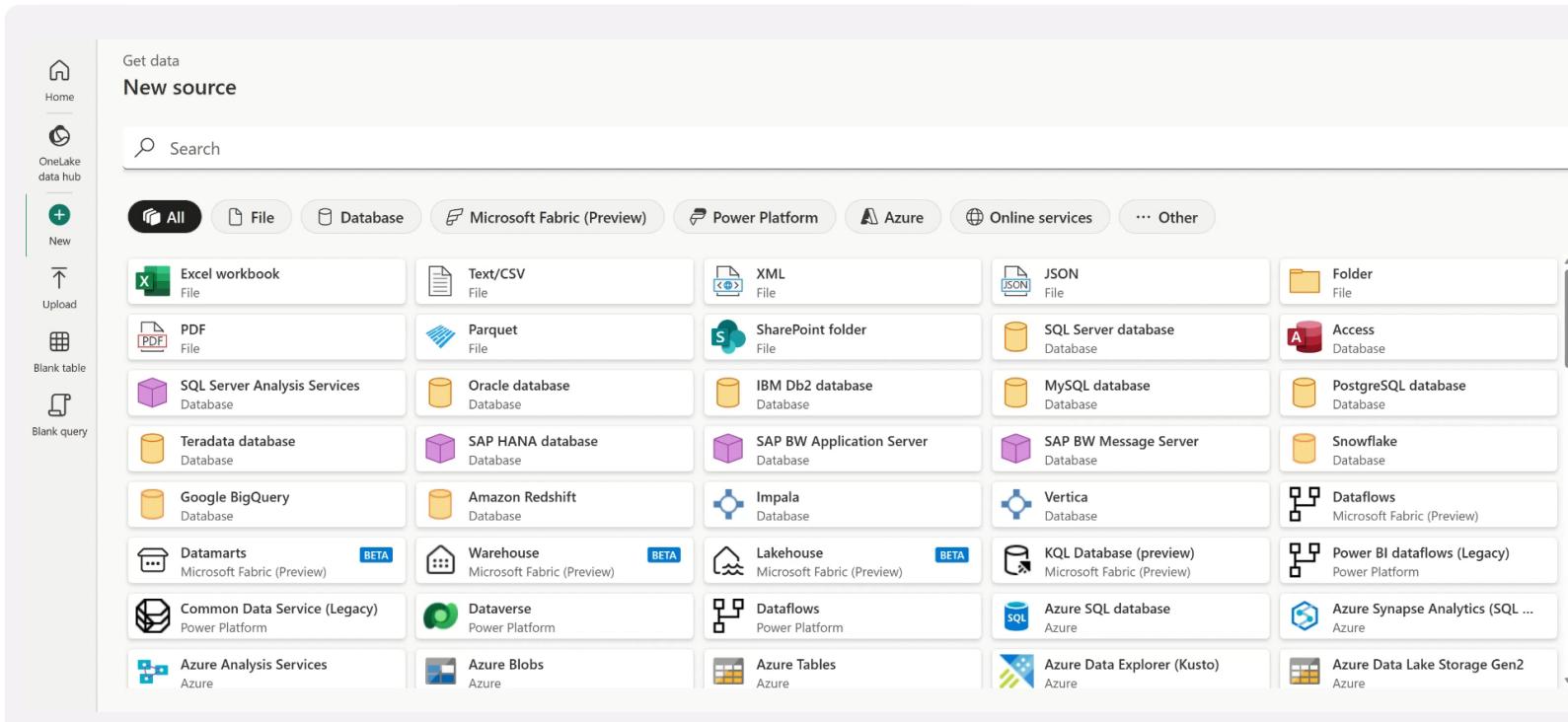
Rich connectivity

Seamlessly connect to more than **170+ data sources**

Connectivity to major data platforms
(from Azure, Google, Amazon, Snowflake, Databricks, Oracle, and more)

Extensibility with Connector SDK

Hybrid, on-prem connectivity through gateways and VNet data gateways



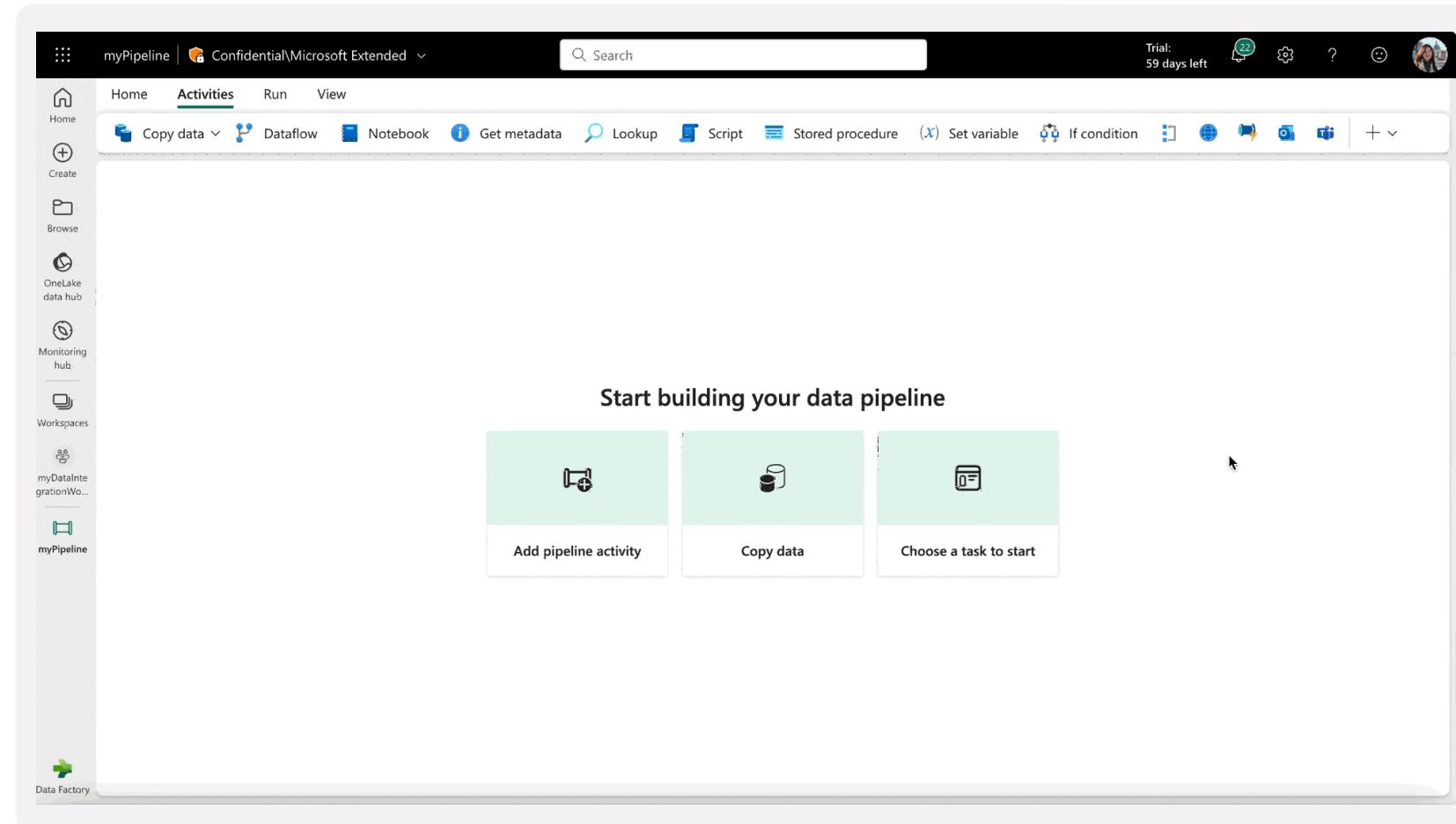
Data pipelines

Low-code Data orchestration

Copy Assistant to jumpstart any copy task from data source to destination

Rich set of orchestration **activities** to create robust and powerful solutions

Seamless connectivity to cloud databases, analytical platforms, business applications on-premises data sources, and more



Dataflow Gen2

Next generation of Data preparation

Easy to use, no-code ETL & ELT

Includes smart AI-based data prep

More than 300+ transformations

Output data destinations

Write output of dataflows to Azure SQL database, Data warehouse, Lakehouse and more

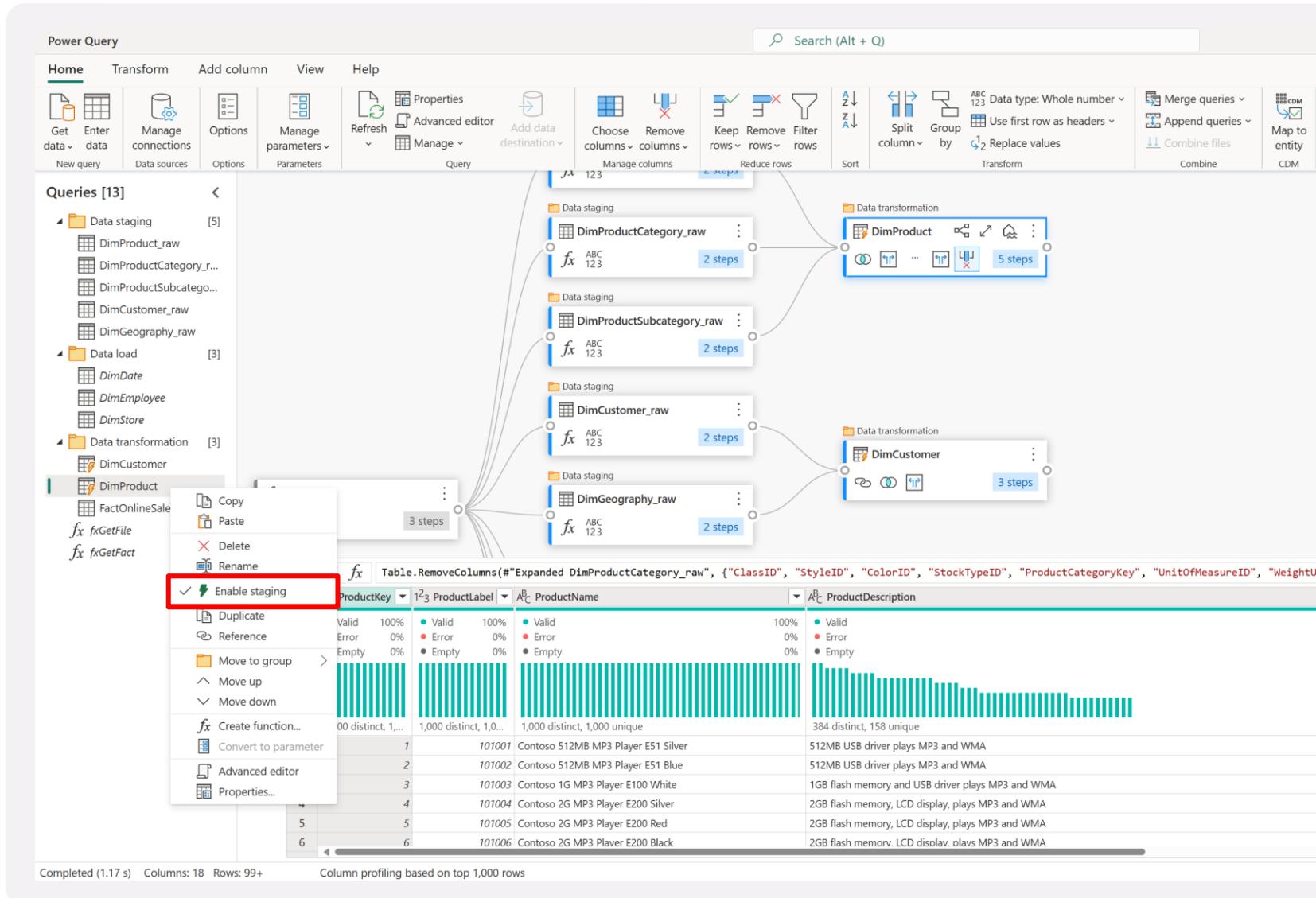
The screenshot shows the Microsoft Power BI Dataflow Gen2 interface. The main area is a "Power Query" editor titled "OnlineSalesDataflow". It displays a large data grid with 24 columns and 99+ rows. The columns represent various dimensions and facts from multiple raw data sources like DimCustomer, DimGeography, and FactOnlineSales. The interface includes a ribbon with tabs like Home, Transform, Add column, View, and Help. On the left, there's a navigation pane with sections for Home, Create, Browse, OneLake data hub, Apps, Metrics, Monitoring hub, Deployment pipelines, Learn, Workspaces, Day After Dashboard, and OnlineSales Dataflow. The right side features a "Query settings" pane with sections for Properties (Name: DimCustomer, Entity type: Custom), Applied steps (Source: Merged queries, Expanded DimGeography_raw), and Data destination (No data destination). The bottom right has a "Step" button and a "Publish" button.

Dataflow Gen2

Highly scalable using
Fabric compute

A **seamless experience** - yielding
fast, easy and powerful results

Abstracts away the complexities of
traditional ETL and ELT

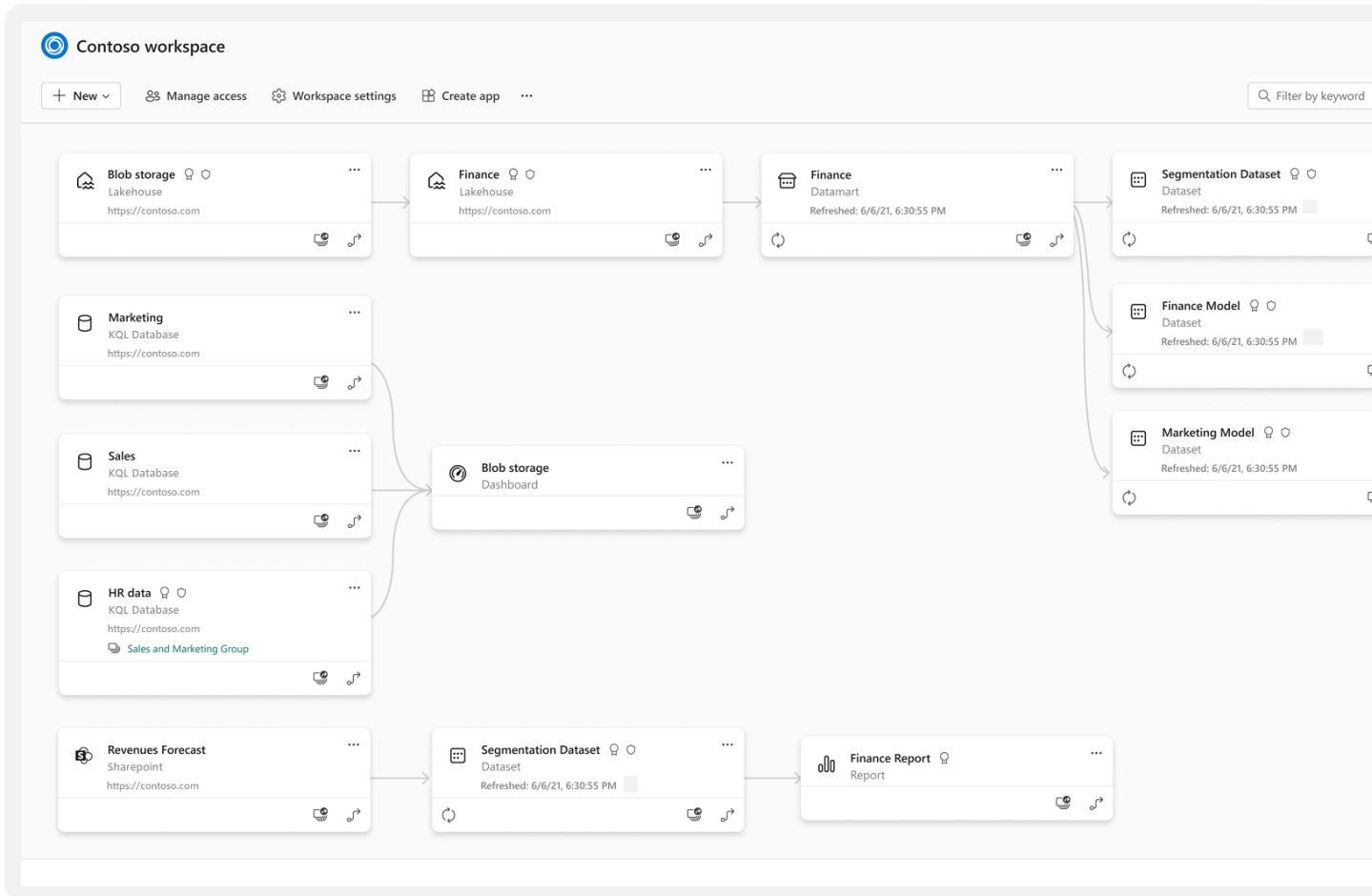


Lineage

Understand the **relationships** between analytical solutions

Available automatically with every workspace

Assess the potential impact of change on downstream items using **impact analysis**



Monitoring hub

Monitor activities from a central location

Gain **deep insights** into how your Fabric items are performing – including dataflows and pipelines

The screenshot shows the Microsoft Synapse Data Engineering Monitoring hub. The left sidebar includes links for Home, Create, Browse, Data hub, Apps, Metrics, Monitoring hub (which is selected), and Deployment pipelines. The main area displays a table of monitoring data with the following columns: Activity name, Status, Item type, Start time, Submitter, and Location. A filter bar at the top allows searching by keyword, applying filters, and adjusting column options. A message at the top states: "Monitoring hub is a station to view and track active activities across different products."

Activity name	Status	Item type	Start time	Submitter	Location
SMC StoryBoard	Completed	Dataset	10:29 AM, 9/4/23	Submitter	Workspace name
MSXI - Sell-With - IP Co-Sell - BizApps Performa...	Completed	Dataset	10:29 AM, 9/4/23	Submitter	Workspace name
Data Refresh	Completed	Dataset	10:29 AM, 9/4/23	Submitter	Workspace name
Support Services Customer Report	Completed	Dataset	10:29 AM, 9/4/23	Submitter	Workspace name
MSX Insights - Azure Close Rate	Completed	Dataset	10:29 AM, 9/4/23	Submitter	Workspace name
Pipeline Management_ProprodRefresh	Completed	Dataset	10:27 AM, 9/4/23	Submitter	Workspace name
All Training Consumption Aggregates	In progress	Dataset	10:26 AM, 9/4/23	Submitter	Workspace name
SOXReports	Completed	Dataset	10:26 AM, 9/4/23	Submitter	Workspace name
PendingRequest	Completed	Dataset	10:26 AM, 9/4/23	Submitter	Workspace name
GPS Insights Hub - Partner Mastering Search	Completed	Dataset	10:26 AM, 9/4/23	Submitter	Workspace name
Partner Parenting User History Page	Completed	Dataset	10:26 AM, 9/4/23	Submitter	Workspace name
Microsoft_GitHub	Completed	Dataset	10:26 AM, 9/4/23	Submitter	Workspace name
SOXReports_UAT_09282018	Completed	Dataset	10:26 AM, 9/4/23	Submitter	Workspace name
ApproverApproval	Completed	Dataset	10:26 AM, 9/4/23	Submitter	Workspace name



Data integration + AI

Copilot in Data Factory

Easily integrate generative AI into your dataflows and *pipelines using Copilot

- Chat with **Copilot** to describe data transformations in natural language
- Tap into generative AI capabilities from **Azure Open AI** as data transformation steps
- *Use **Copilot** to schedule and run and manage dataflows

Orders | Data updated 1/12/23

Home Transform Add Column View Help

Get data Enter data Refresh preview Add data destination Choose columns

Queries [2]

Orders - Staging

Orders

Source Navigation Add step

Source Split column Open AI Add step

Query settings

Name Orders

Applied steps

Reference Split column by... Open AI - Dissatisfacti...

Create dataflow transforms with Copilot

Bring orders and split location column by comma

Your dataflow has been updated with two queries: Orders - Shipping and Orders.

Identify the dissatisfaction reason from CustomerReview, where the reasons include "Product arrived late", "Product was damaged", "Product was defective", "Delivered to incorrect address"

Done - dataflow updated.

A new DissatisfactionReason column was created with the dissatisfaction categories extracted from the CustomerReview column.

Keep it

Ask a question or type / for suggestions

Completed (0.86 s) Columns: 20 Rows: 99+

No data destination

Step Publish

Lakehouse overview

Lakehouse overview

Store, manage and analyze all your data in a single location and easily share across the entire enterprise

- Flexible and scalable solution that enables organizations to handle large data volumes of all types and sizes
- Built-in SQL endpoint unlocks Data Warehouse capabilities on top of your Lakehouse with no data movement
- Address the challenges of traditional Data Lakes by adding a **Delta Lake storage layer** directly on top of the cloud Data Lake

The screenshot shows the TFLakehouse interface. On the left, there's a sidebar with various icons: Home, Create, Browse, OneLake data hub, Monitoring hub, Workspaces, My workspace, TFLakehouse (which is selected), and Data Engineering. The main area has a header with 'TFLakehouse' and a search bar. Below the header are buttons for 'Get data', 'New Power BI dataset', and 'Open notebook'. A message indicates that a SQL endpoint and default dataset were created. The 'Home' tab is selected. In the center, there's an 'Explorer' section showing a tree view of 'TFLakehouse' with 'Tables' and 'Files' nodes, and a 'taxi_zone_lookup' table highlighted. To the right is a detailed view of the 'taxi_zone_lookup' table with the following data:

LocationID	Borough	Zone	service_zone
1	EWR	Newark Air...	EWR
2	Queens	JFK Airport	Airports
3	Queens	LaGuardia ...	Airports
4	Unknown	NV	N/A
5	Unknown	NA	N/A
6	Manhattan	Alphabet City	Yellow Zone
7	Manhattan	Battery Park	Yellow Zone
8	Manhattan	Battery Par...	Yellow Zone
9	Manhattan	Bloomingd...	Yellow Zone

Lakehouse and Delta Lake

Microsoft Fabric Lakehouse -

Store, manage, and analyze structured and unstructured data in a single location

Delta Lake - Achieve seamless data access across all compute engines in Microsoft Fabric

The screenshot shows the Microsoft Fabric Data Engineering interface. On the left, the Explorer sidebar displays a folder structure under 'fabric_lakehouse' named 'Tables' containing 'salesorders' and 'Files'. The main area is titled 'salesorders' and shows a table with columns: SalesOrderID, SalesOrderNumber, OrderDate, CustomerName, Email, Item, and Quantity. The table contains 12 rows of sample data. The interface includes a navigation bar at the top with tabs for Home, Create, Browse, Data hub, Monitoring hub, Workspaces, and 'fabric_lakehouse'. A 'Lakehouse' button is also visible in the top right.

	SalesOrderID	SalesOrderNumber	OrderDate	CustomerName	Email	Item	Quantity
1	SO51555	7	6/23/2021 ...	Chloe Garcia	chloe27@ad...	Patch Kit/8 ...	1
2	SO54042	7	8/9/2021 1...	Logan Collins	logan29@ad...	Half-Finger ...	1
3	SO54784	7	8/22/2021 ...	Autumn Li	autumn3@ad...	All-Purpose...	1
4	SO58572	7	10/25/2021...	Cesar Sara	cesar9@ad...	Short-Sleev...	1
5	SO58845	7	10/30/2021...	Peter She	peter8@ad...	Sport-100 ...	1
6	SO58845	8	10/30/2021...	Peter She	peter8@ad...	Long-Sleev...	1
7	SO60233	7	11/16/2021...	Jason Mitch...	jason40@ad...	Sport-100 ...	1
8	SO61412	7	12/3/2021 ...	Nathaniel C...	nathaniel9...	Short-Sleev...	1
9	SO62984	7	12/29/2021...	Miguel San...	miguel72@ad...	Racing Soc...	1
10	SO51555	6	6/23/2021 ...	Chloe Garcia	chloe27@ad...	Mountain B...	1
11	SO52058	6	7/4/2021 1...	Elijah Ross	elijah7@ad...	Short-Sleev...	1
12	SO53255	6	7/28/2021 ...	Edward Tait	edward31@ad...	Short-Sleev...	1

Interacting with the Lakehouse

The Lakehouse explorer

Explore data in the Lakehouse using the object explorer

Notebooks

Use the Notebook to write code to read, transform and write directly to Lakehouse as tables and/or folders

Data pipelines

Use pipeline copy tool to pull data from other sources and land into the Lakehouse

Apache Spark job definitions

Develop robust applications and orchestrate the execution of compiled Spark jobs in Java, Scala, and Python

Dataflows Gen 2

Use Dataflows Gen 2 to ingest and prepare the data

Automatic table discovery and registration

The screenshot shows the Power BI Data Explorer interface. On the left, the sidebar includes icons for Home, Create, Browse, OneLake data hub, Monitoring hub, Workspaces, and Data Engineering. The main area has a breadcrumb navigation path: Files > TaxiData. The central view displays a table of files in the 'TaxiData' folder:

Name	Date modified	Type	Size
taxi_zone_lookup.csv	5/3/2023 11:30:02 AM	CSV	12 KB

The 'taxi_zone_lookup.csv' file is listed with its name, last modified date, type (CSV), and size (12 KB). The 'Lakehouse' dropdown in the top right corner indicates the data source type.

Data ingestion into Lakehouse

Easily ingest data into the Lakehouse through a variety of methods

- Get files from your computer using direct upload
- Connect to 120+ data sources and apply multiple transformations using Dataflows or copy petabyte-sized lakes using the copy activity in Data pipelines
- Use Spark code to connect to data sources using available Spark libraries
- Leverage shortcuts to create pointers to existing data in OneLake and external storage accounts with no data movement at all
- Shortcuts behave in the same way as hosted storage

The screenshot shows the Azure Data Explorer interface. On the left, the Explorer sidebar displays a hierarchical view of storage accounts and tables. Under 'contosolh', there are 'Tables' containing 'orders', 'taxi', and 'trafficflow', and a 'Files' folder containing 'Download'. The 'orders' table is selected. To the right, the 'orders' table is shown as a data grid with columns '_col0_', '_col1_', and '_col2_'. The data consists of five rows with values: (1, 1, 36899983, O), (2, 2, 78001627, O), (3, 3, 123313904, F), (136776016, O), and (44484776, F). A context menu is open over the first row, showing options: Refresh, New shortcut, New subfolder, Upload (selected), Properties, Upload files, and Upload folder.

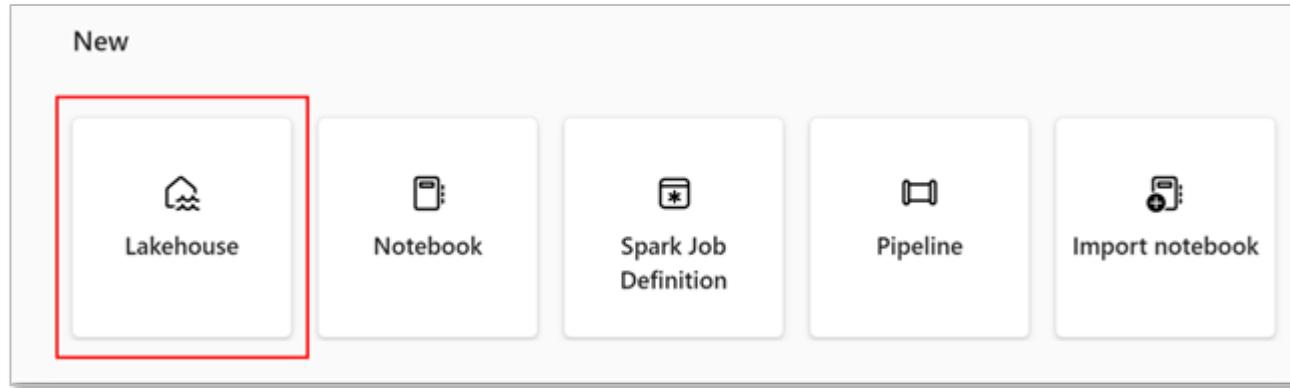
	col0	_col1_	_col2_
1	1	36899983	O
2	2	78001627	O
3	3	123313904	F
136776016		O	
44484776		F	
55622011		F	
66957875			F

Ways to create a Lakehouse in Microsoft Fabric

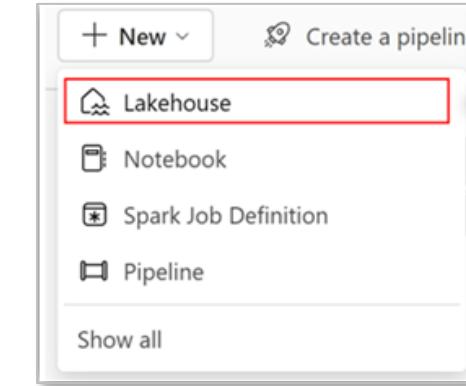


Ways to create a Lakehouse

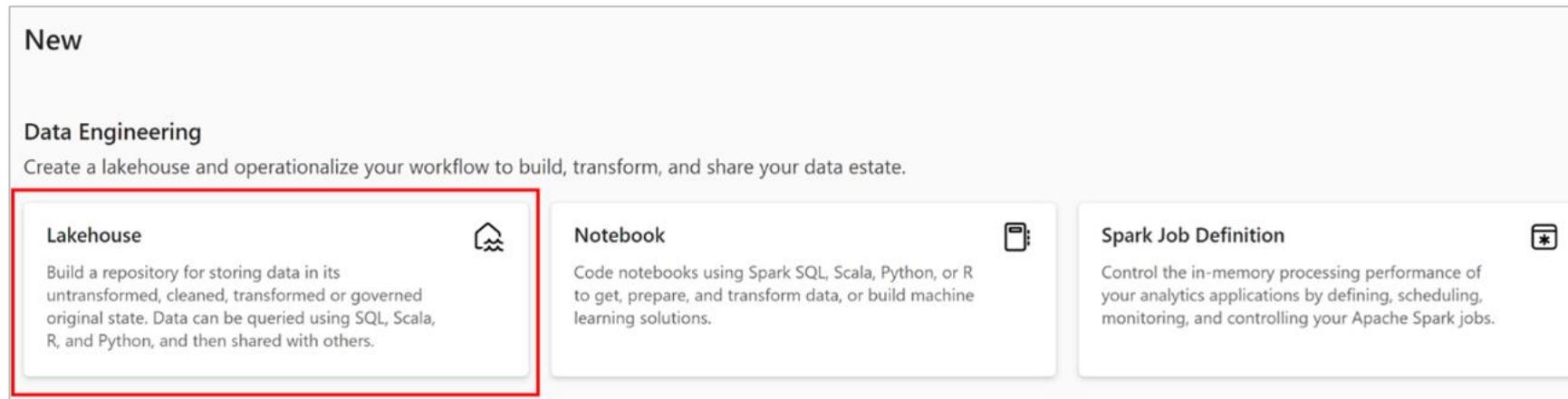
Using Data Engineering homepage



Using Workspace view



Creating Hub



Navigating the Fabric Lakehouse explorer

The screenshot shows the Microsoft Fabric Lakehouse explorer interface. On the left, there's a navigation sidebar with icons for Home, Create, Browse, Data hub, Monitoring hub, Workspaces, LHRedesign_AviWS, ContosoDailySales, and Power BI. The main area has a header with 'Home' and 'Lakehouse' dropdowns, and buttons for 'Get data', 'New Power BI dataset', and 'Open notebook'. A tooltip message indicates that a SQL endpoint for SQL querying and a default dataset for reporting were created and will be updated with any tables added to the lakehouse. The 'Lakehouse explorer' section shows a tree view of datasets: 'ContosoDailySales' (Tables: Customer, inventory, product, sales, Transactions; Unidentified: CustomerFeedbackAudio), 'Files' (Employee, Product, Sales, Transaction), and 'More...'. The right side displays the 'Customer' table data with columns: Index, UserId, FirstName, LastName, Sex, Email, Phone, DateOfBirth, and JobTitle. The table shows 11 rows of sample data.

Index	UserId	FirstName	LastName	Sex	Email	Phone	DateOfBirth	JobTitle	
1	2	3d5AD30A...	Jo	Rivers	Female	fergusonkat...	-10395	7/26/1931	Dancer
2	3	810Ce0F27...	Sheryl	Lowery	Female	fhoward@e...	(599)782-0...	11/25/2013	Copy
3	5	9afFEafAe1...	Lindsey	Rice	Female	elin@exam...	(390)417-1...	4/15/1923	Biomedical ...
4	13	CDA21B6e8...	Eddie	Barnes	Female	brandy23@...	801.809.91...	2/27/1975	Dramathera...
5	14	1CC30c5F2...	Ralph	Lowe	Female	dleon@exa...	+1-511-127...	4/10/1938	Presenter, b...
6	16	bFCFDdE54...	Carly	Abbott	Female	stricklando...	(416)979-0...	10/27/2007	Therapeutic...
7	18	aCeff56E59...	Natasha	Macias	Female	dorothyme...	(929)366-8...	10/31/1971	Recruitmen...
8	19	CF091D6b9...	Courtney	Jenkins	Female	estesana@...	(973)243-9...	1/20/1948	Accounting...
9	20	462EF46dca...	Perry	Mcmahon	Female	allison66@...	060-611-93...	11/24/2006	Education o...
10	24	3Cb9Fe3aB...	Norman	Walton	Female	samanthas...	(590)187-8...	6/19/1973	Personnel o...
11	25	be6BBa9EB...	Roaer	Sweeney	Female	leblanciohn...	-8153	9/9/2008	Race relatio...

Auto discovery of tables

The Lakehouse explorer provides a tree-like view of the objects in the Microsoft Fabric Lakehouse item

It has a key capability of discovering and displaying tables that are described in the metadata repository and in OneLake storage

The screenshot shows the Microsoft Fabric Lakehouse Explorer interface. On the left, there's a sidebar with icons for Home, Create, Browse, OneLake data hub, Monitoring hub, Workspaces, My workspace, TFLakehouse (which is selected), and Data Engineering. The main area has a header with 'TFLakehouse' and a search bar. Below the header, there's a message: 'A SQL endpoint for SQL querying and a default dataset for reporting were created and will be updated with any tables added to the lakehouse.' The central part is divided into two sections: 'Explorer' on the left and a table preview on the right. The 'Explorer' section shows a tree view under 'TFLakehouse' with 'Tables' expanded, showing 'taxi_zone_lookup' (selected), '123_LocationID', 'ABC_Borough', 'ABC_Zone', and 'ABC_service_zone'. Under 'Files', there's a 'TaxiData' folder. To the right of the tree view is a preview of the 'taxi_zone_lookup' table with the following data:

	LocationID	Borough	Zone	service_zone
1	1	EWR	Newark Air...	EWR
2	132	Queens	JFK Airport	Airports
3	138	Queens	LaGuardia ...	Airports
4	264	Unknown	NV	N/A
5	265	Unknown	NA	N/A
6	4	Manhattan	Alphabet City	Yellow Zone
7	12	Manhattan	Battery Park	Yellow Zone
8	13	Manhattan	Battery Par...	Yellow Zone
9	24	Manhattan	Bloomingd...	Yellow Zone
10	43	Manhattan	Central Park	Yellow Zone

Load to Delta Lake tables

Load single file into a new or existing table

Tables are loaded using the Delta Lake table format with V-Order optimization

The screenshot shows the Databricks UI interface. On the left, the sidebar includes icons for Home, Create, Browse, OneLake data hub, Monitoring hub, Workspaces, My workspace, and TFLakehouse. The main area has a header "TFLakehouse" with tabs for Home, Get data, New Power BI dataset, and Open notebook. A message at the top states: "A SQL endpoint for SQL querying and a default dataset for reporting were created and will be updated with any tables added to the lakehouse." Below this is the "Explorer" section, which shows the structure of the TFLakehouse. Under "Tables", there is a folder "taxi_zone_lookup" containing columns "123 LocationID", "ABC Borough", "ABC Zone", and "ABC service_zone". Under "Files", there is a folder "TaxiData" containing a file "taxi_zone_lookup.csv". A context menu is open over the "taxi_zone_lookup.csv" file, with the "Load to Tables" option highlighted by a red box. Other options in the menu include Rename, Delete, and Properties.

Table and column name validation and rules

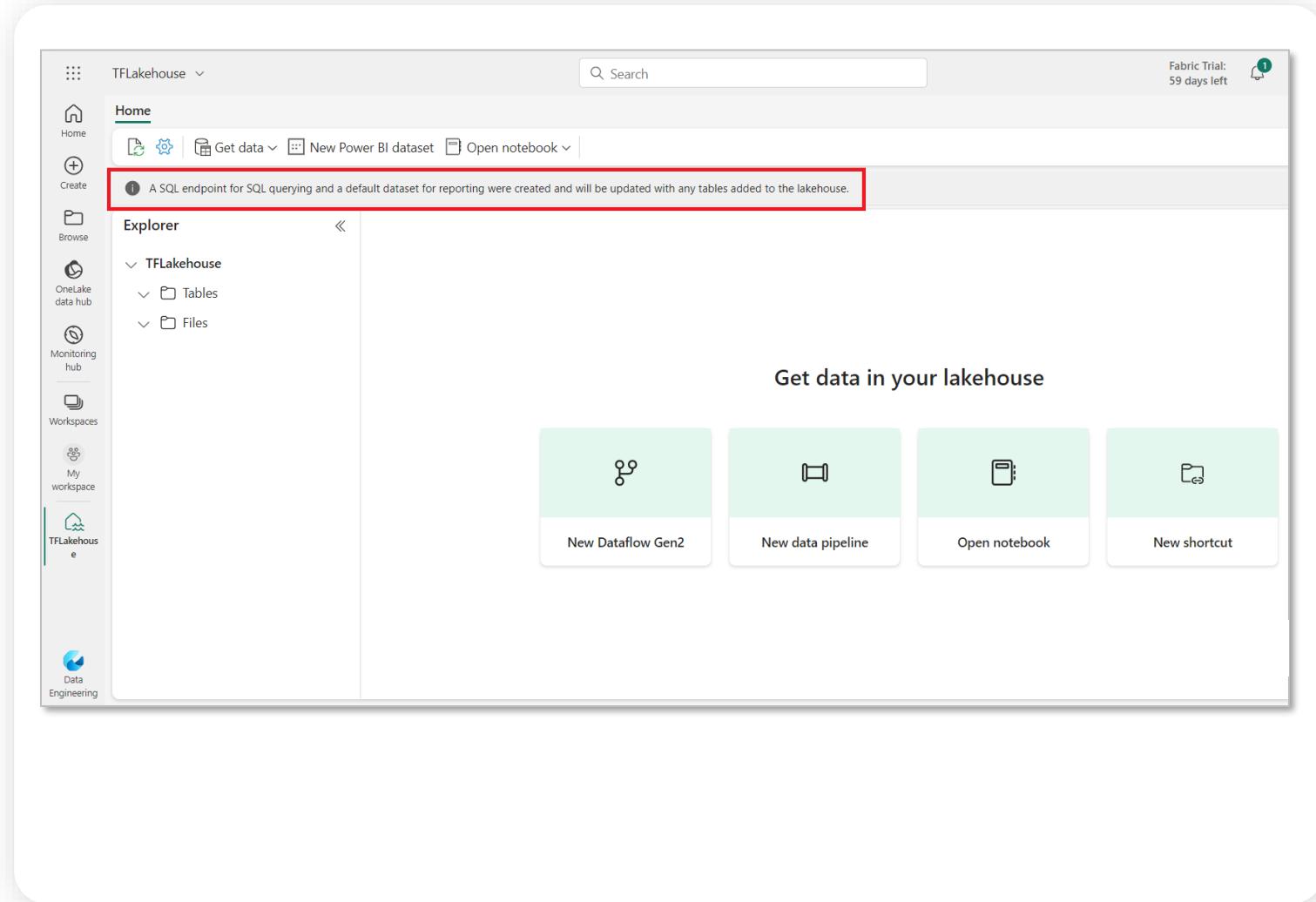
Table names can only contain alphanumeric characters and underscores

Text files without column headers are replaced with standard col# notation as the table column names

Column names are validated during the load action
If no proper column name is achieved during validation, the load action fails

Lakehouse SQL endpoint

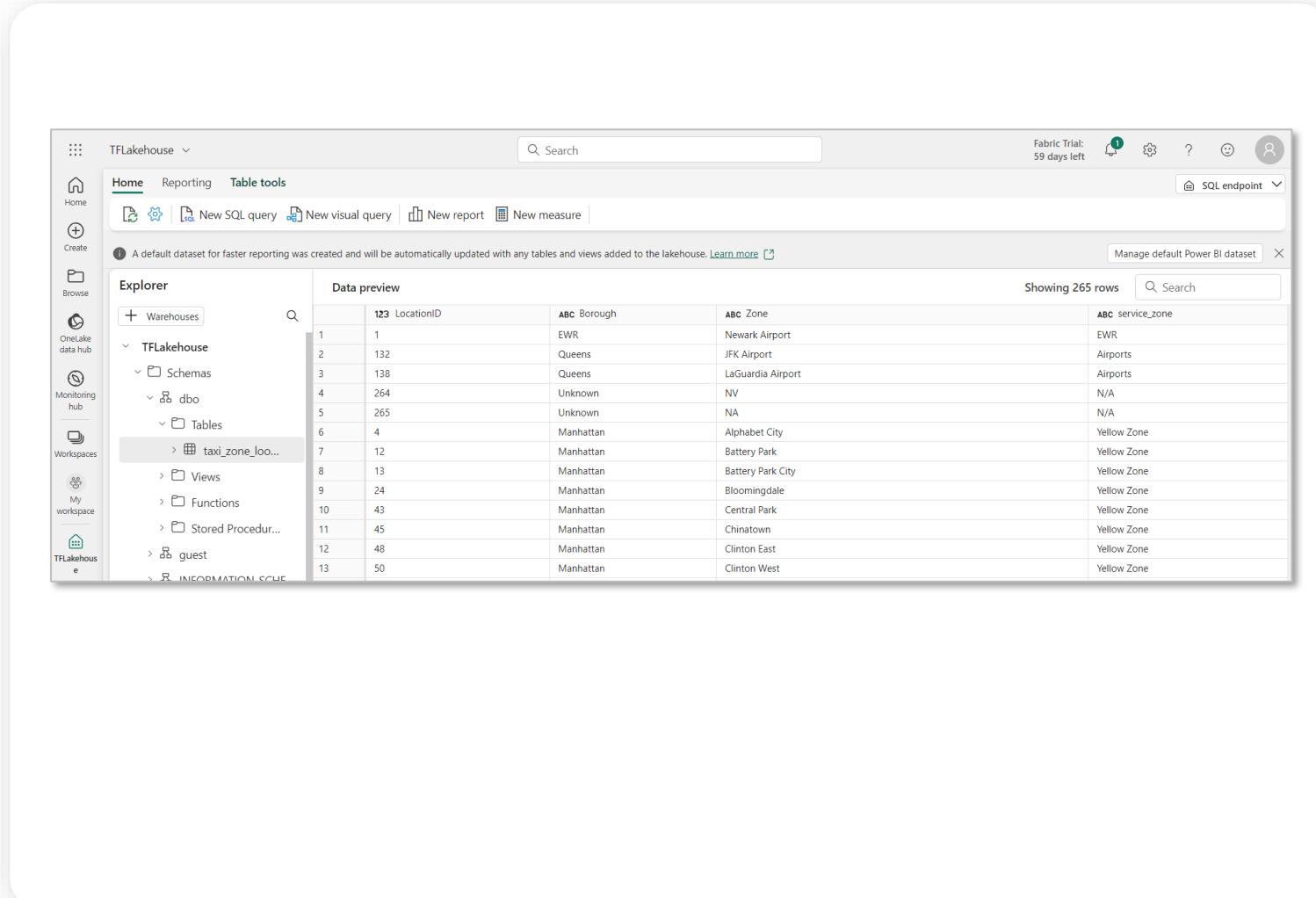
The Lakehouse creates a serving layer by automatically generating a SQL endpoint and a default dataset during creation



SQL endpoint read-only mode

You can only read data from delta tables using SQL endpoint

Save functions, views, and set SQL object-level security



The screenshot shows the Power BI desktop application interface. The left sidebar displays the 'TFLakehouse' workspace with various navigation options like Home, Create, Browse, OneLake data hub, Monitoring hub, Workspaces, My workspace, and TFLakehouse. The main area is titled 'Data preview' and shows a table with 13 rows of data from the 'taxi_zone_lookup' table. The columns are LocationID, ABC Borough, ABC Zone, and ABC service_zone. The data includes entries for Newark Airport, JFK Airport, LaGuardia Airport, and various Manhattan locations across different boroughs and zones. A message at the top indicates a default dataset was created for faster reporting.

	LocationID	ABC Borough	ABC Zone	ABC service_zone
1	1	EWR	Newark Airport	EWR
2	132	Queens	JFK Airport	Airports
3	138	Queens	LaGuardia Airport	Airports
4	264	Unknown	NV	N/A
5	265	Unknown	NA	N/A
6	4	Manhattan	Alphabet City	Yellow Zone
7	12	Manhattan	Battery Park	Yellow Zone
8	13	Manhattan	Battery Park City	Yellow Zone
9	24	Manhattan	Bloomingdale	Yellow Zone
10	43	Manhattan	Central Park	Yellow Zone
11	45	Manhattan	Chinatown	Yellow Zone
12	48	Manhattan	Clinton East	Yellow Zone
13	50	Manhattan	Clinton West	Yellow Zone

DEMO

Creating a Lakehouse



Getting data into Fabric Lakehouse



Getting data into the Fabric Lakehouse

File upload from local computer

Run a copy tool in pipelines

Set up a dataflow

Apache Spark libraries in Notebook code

The screenshot shows the Fabric Lakehouse interface. On the left, there's a sidebar with icons for Home, Create, Browse, OneLake data hub, Monitoring hub, and Workspaces. The main area has a title bar 'TFLakehouse' and a search bar. Below the title bar, there are several buttons: 'Get data' (with a red box around it), 'New Power BI dataset', and 'Open notebook'. The 'Get data' button has a dropdown menu open, showing options: 'Upload files', 'New data pipeline', 'New Dataflow Gen2', and 'New shortcut'. To the right of this, there's a note: 'Default dataset for reporting were created and will be updated with any tables added to the lakehouse.' Further down, there's a table titled 'taxi_zone_lookup' with columns: LocationID, Borough, Zone, and service_zone. The table contains four rows of data:

	LocationID	Borough	Zone	service_zone
1	1	EWR	Newark Air...	EWR
2	132	Queens	JFK Airport	Airports
3	138	Queens	LaGuardia ...	Airports
4	264	Unknown	NV	N/A

Considerations when loading data

Use case	Recommendation
Small file upload from local machine	Use OneLake file explorer / Upload file
Small data or specific connector	Use Dataflows Gen2
Large data source or specific connector	Use Copy activity in Data pipelines
Complex data transformations	Use Notebook code

Data ingestion scenarios

Connecting to existing SQL Server and copying data into Delta table on the Lakehouse

Uploading files from your computer

Copying and merging multiple tables from other Lakehouses into a new Delta table

Connecting to a streaming source to land data in a Lakehouse

Referencing data without copying it from other internal Lakehouses or external sources

Shortcuts in Lakehouse

Shortcuts in a Lakehouse allow users to reference data without copying it

It unifies data from different Lakehouses, workspaces, or external storage, such as

- ADLS Gen2
- AWS S3

The screenshot shows the TFLakehouse interface in Power BI. On the left, there's a sidebar with icons for Home, Create, Browse, OneLake data hub, Monitoring hub, Workspaces, My workspace, and TFLakehouse. The main area has a 'Home' tab selected, with options to 'Get data', 'New Power BI dataset', and 'Open notebook'. A message at the top says: 'A SQL endpoint for SQL querying and a default dataset for reporting were created and will be updated with any tables added to the lakehouse.' Below this is an 'Explorer' section. Under 'TFLakehouse > Tables > taxi_zone_lookup', there are columns for 'LocationID', 'Borough', 'Zone', and 'service_zone'. A context menu is open over the 'taxi_zone_lookup' table, with options: 'Load to Tables', 'New shortcut' (which is highlighted with a red box), 'New subfolder', 'Upload', 'Rename', 'Delete', 'Properties', and 'Refresh'. To the right of the table is a preview of the data in a table format:

ID	Borough	Zone	service_zone
EWR	Newark Air...	EWR	
Queens	JFK Airport	Airports	
Queens	LaGuardia ...	Airports	
Unknown	NV	N/A	
Unknown	NA	N/A	
Manhattan	Alphabet City	Yellow Zone	
Manhattan	Battery Park	Yellow Zone	

Lakehouse sharing

Share your Lakehouse as a data product with consumers

- Provide users access to a Lakehouse without adding them to your workspace
- Grant access to Lakehouse data through Spark, SQL Endpoint, and default dataset for Power BI reporting
- Use SQL Security or customer permission to grant access through SQL Endpoint
- Discover Lakehouses you have access to in the OneLake Data Hub

The screenshot shows the Microsoft Power BI OneLake data hub interface. On the left, there is a navigation sidebar with various icons for Home, Create, Browse, Data Hub, Monitoring hub, Metrics, Apps, Deployment pipelines, Learn, Workspaces, Unredesign AviWS, and ContosoDailySales. The main area displays a grid of cards for recommended datasets. Below the grid is a table titled 'Explorer' listing datasets and湖houses.

Name	Type	Owner	Location	Refreshed	Endorsement	Sensitivity
ContosoDailySales	Lakehouse	Avinanda Chattapad...	LHRedesign_AviWS	-	-	Confidential\Microsoft Ext...
Test	Lakehouse	Avinanda Chattapad...	Customer360WS	-	-	Confidential\Microsoft Ext...
Test2	Lakehouse	Avinanda Chattapad...	Customer360WS	-	-	Confidential\Microsoft Ext...
Customer360	Lakehouse	Avinanda Chattapad...	Customer360WS	-	-	Confidential\Microsoft Ext...
Test2	Warehouse (default)	Avinanda Chattapad...	Customer360WS	12/31/52, 4:07:02 PM	-	Confidential\Microsoft Ext...
Test2	Dataset (default)	Avinanda Chattapad...	Customer360WS	4/10/23, 1:52:44 PM	-	Confidential\Microsoft Ext...

Monitoring

Spark 3.1 and higher versions have a built-in structured streaming UI containing the following streaming metrics

- Input Rate
- Process Rate
- Input Rows
- Batch Duration
- Operation Duration

The screenshot shows a Databricks workspace titled "MonitoringBugBash". On the left, there's a sidebar with options like "+ New", "Create app", "Manage access", and "Workspace settings". Below that is a section titled "Introducing datamarts (preview)" with a sub-section "Get business-focused insights faster with new self-service datamarts". The main area displays a list of notebooks and job definitions:

Name	Type
Notebook 9	Notebook
NotebookSample	Notebook
oneplusone	Spark Job Definition
oneplusone2	Spark Job Definition
rongbilkh	Spark Job Definition
SJDSample	Spark Job Definition
Test	Spark Job Definition
wekoSJD	Spark Job Definition

A context menu is open over the "NotebookSample" entry, showing options: "Open", "Delete", "Settings", and "Recent runs". The "Recent runs" option is highlighted with a red box and has a red arrow pointing to it from the bottom right. To the right of the list is a modal window titled "NotebookSample" with the heading "Recent runs". This modal contains a table of recent runs:

Application name	Submitted	Submitter	Status	Total dura...	Run kind	Livy Id
sparksession	7/22/22 1:29:00 PM	Submitter	Queued	-	-	833545fc-b0e2-4...
sparksession	7/22/22 12:05:08 PM	Submitter	Stopped (sessior	13m 20s	-	3156834c-ddc2-4...
sparksession	7/22/22 11:31:36 AM	Submitter	Stopped (sessior	13m 43s	-	16ea5191-1fb6-4...
sparksession	7/22/22 10:53:51 AM	Submitter	Stopped (sessior	11m 49s	-	8175f0ec-b857-4...
sparksession	7/21/22 7:35:08 PM	Submitter	Stopped (sessior	14m 39s	-	a29726d2-dfe3-4...
sparksession	7/21/22 1:03:50 PM	Submitter	Stopped (sessior	11m 33s	-	05ae3edc-d4e1-4...
sparksession	7/21/22 12:59:03 PM	Submitter	Stopped (sessior	11m 24s	-	c5bc8149-ec1c-4...
sparksession	7/20/22 9:40:02 AM	Submitter	Stopped (sessior	10m 44s	-	0638e3cc-1778-4...
sparksession	7/15/22 8:35:42 AM	Submitter	Cancelled	3m 59s	-	90253fcf-d129-4...
sparksession	7/15/22 7:56:06 AM	Submitter	Queued	-	-	6ebea23a-51a4-4...
-Artifacts-s032ab742-de0da-d49fb...	7/14/22 7:40:35 AM	Submitter	Failed (...	15m 39s	-	d7161441-73d1-4...
-Artifacts-s032ab742-de0da-d49fb...	7/14/22 7:08:36 AM	Submitter	Failed (...	15m 39s	-	e95196f8-0aa9-4...
-Artifacts-s032ab742-de0da-d49fb...	7/13/22 2:41:12 PM	Submitter	Failed (...	15m 40s	-	80e0f91a-58fd-4...

Copy activity, dataflow, and Spark decision guide

	Data 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	Assistant, canvas	Power Query	Notebook, Spark job definition
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

Overview of Data Engineering in Microsoft Fabric



Data engineering in Fabric

Empower data engineers to transform data at scale and build a Lakehouse architecture



Build a Lakehouse
for all your
organizational data



Spark runtime with great
out of the box
performance and robust
admin controls



Delightful authoring
experience in your
tools of choice



Completely
integrated into the
Fabric foundation

Overview of Data Engineering



Lakehouse

Store and manage structured and unstructured data in a single location



Apache Spark job definition

Submit batch/streaming job to Spark cluster, apply different transformation logic to the data



Notebook

Create and share documents that contain live code, equations, visualizations, and narrative text



Data pipeline

Collect, process, and transform data from its raw form to a format that you can use for analysis and decision-making

Working with Data Pipelines in Microsoft Fabric



Data pipelines

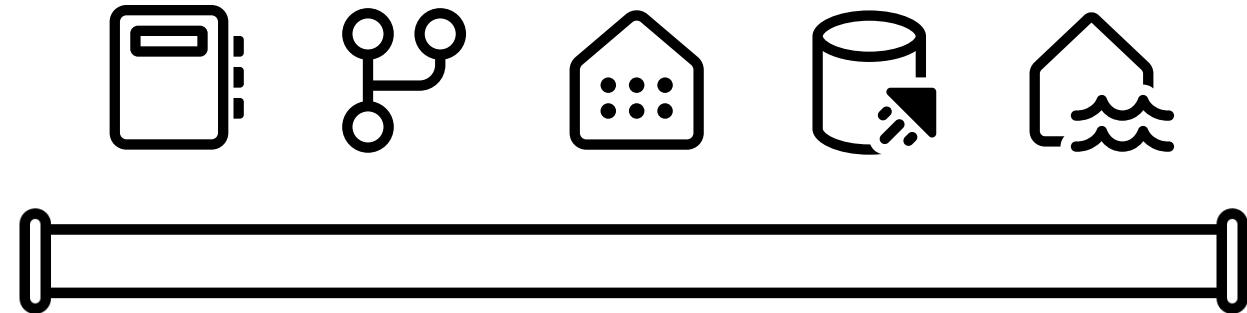
Seamlessly connect and ingest data into Fabric using a **no-code** interface.

Evolution of **Azure Data Factory** pipelines

Rich library of **activities**

Fast copy

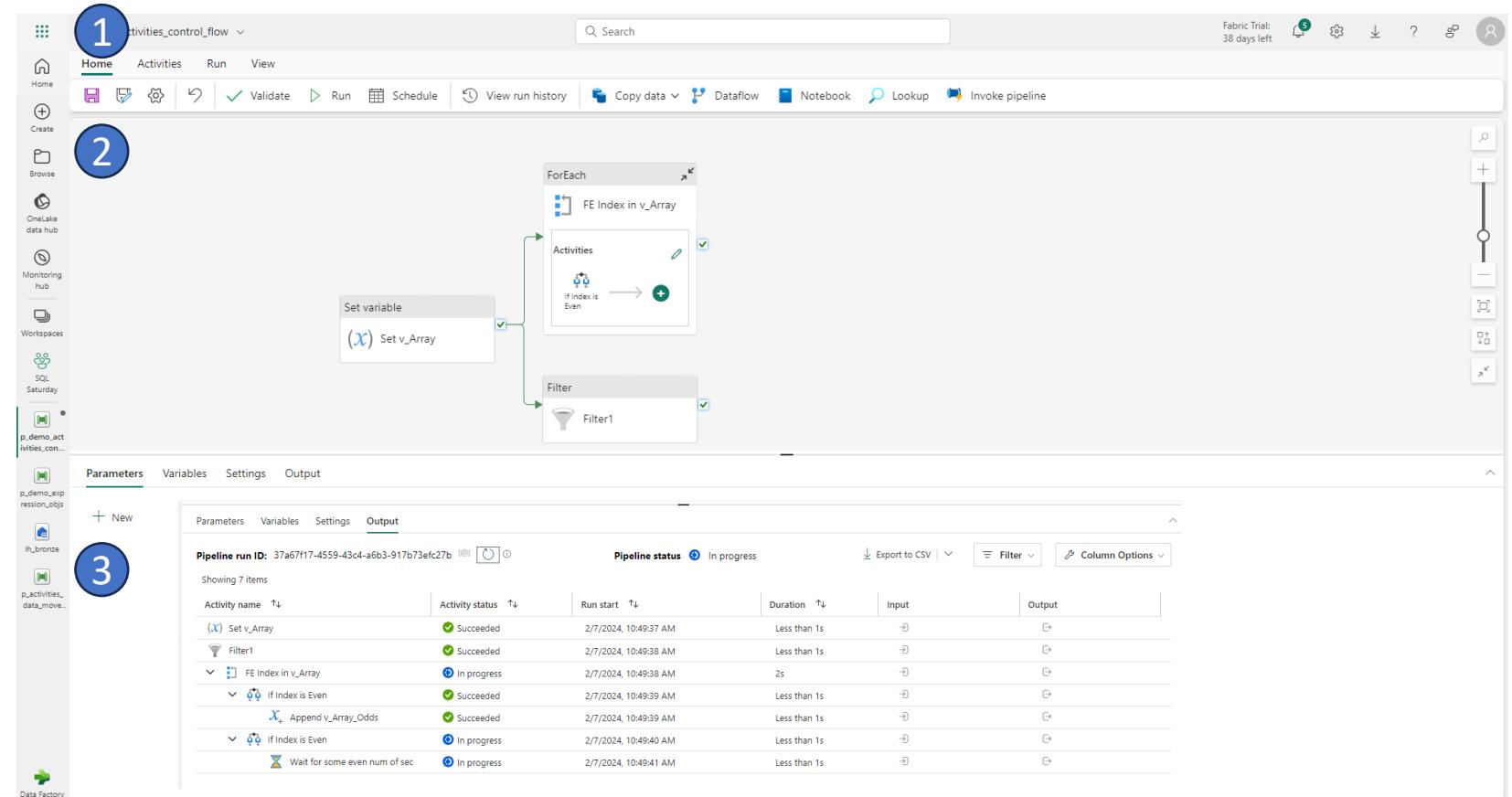
Jumpstart with **Copy Assistant**



Exploring Data pipelines

Data Pipeline

1. Command Ribbon
2. Authoring canvas
3. Properties/output



Copy activity

The Copy Activity is the **core** of Data pipelines.

Jumpstart with **Copy Assistant**

Fine grain control of file type conversion, column mapping, merging/splitting of files, and more..

Workspace Source and Destinations include Lakehouse (Tables & Files), Warehouse, and KQL Database

Supports external sources and destinations

Copy data



Orders to Bronze



* Copy Activity can only interact with Fabric Items within the same workspace as the Data Pipeline; you must create a Shortcut to interact with data in other workspaces.

And so much more...

Control Flow	Nesting	Parameterization	Expression Language	Repeatable
<ul style="list-style-type: none">• Conditional paths• Loops	<ul style="list-style-type: none">• Invoking data pipelines from other data pipelines• Bi-directional communication via output variables	<ul style="list-style-type: none">• Parameters of varying data types	<ul style="list-style-type: none">• Almost every field can be made dynamic!• Extensive built-in functions allow for massive customizations (data driven designs)	<ul style="list-style-type: none">• Scheduled

Data Pipeline: Schedule

Seamlessly schedule events and set a desired time value.

Built in monitoring

Repeat: By the minute, Hourly, Daily, Weekly

The screenshot shows the SalesPipeline Data pipeline interface. At the top, there is a logo and the text "SalesPipeline Data pipeline". Below the logo is a search bar with the placeholder "Search". To the right of the search bar are several status indicators: "Last success is in" (May 11, 2023 at 12:00:04 AM, UTC-06:00 Central Time (US and Canada)), "Next refresh in" (10 hour(s) 38 minute(s)), and a "Run" button. On the left side of the main area, there are navigation links: "About", "Sensitivity label", "Endorsement", and "Schedule" (which is highlighted). In the center, under the "Schedule" section, there is a "Scheduled run" toggle switch (set to "On") and a "Repeat" dropdown menu. The repeat options include "Daily", "Hourly", and "Weekly", with "Daily" currently selected. Below the repeat options are "start" and "End" date/time fields, and a "Time zone" dropdown set to "(UTC-06:00) Central Time (US and Canada)". At the bottom of the schedule section is an "Apply" button.

Data Pipeline: Event Trigger (Preview)

Invoke a Data Pipeline upon a file event using event streams and Reflex triggers

Extensive event types and filtering capabilities

Ability to link multiple pipelines to a single event



DEMO

Data pipeline



Migrate from ADF/Synapse Pipelines to Fabric Data pipelines

Feature	Fabric Data Pipelines	Azure Data Factory	Synapse Pipelines
Office 365 Outlook Activity (Email activity)	Yes	*No	*No
MS Teams Activity	Yes	*No	*No
Refresh a Dataflow Gen2	Yes	*No	*No
Refresh a Power BI semantic model	Yes	*No	*No
Data Store Type: Workspace	Yes	No	No
Change Data Capture	Yes	Yes	No
Disable Activity	Yes	Yes	Yes
Managed Airflow	Yes	Yes	

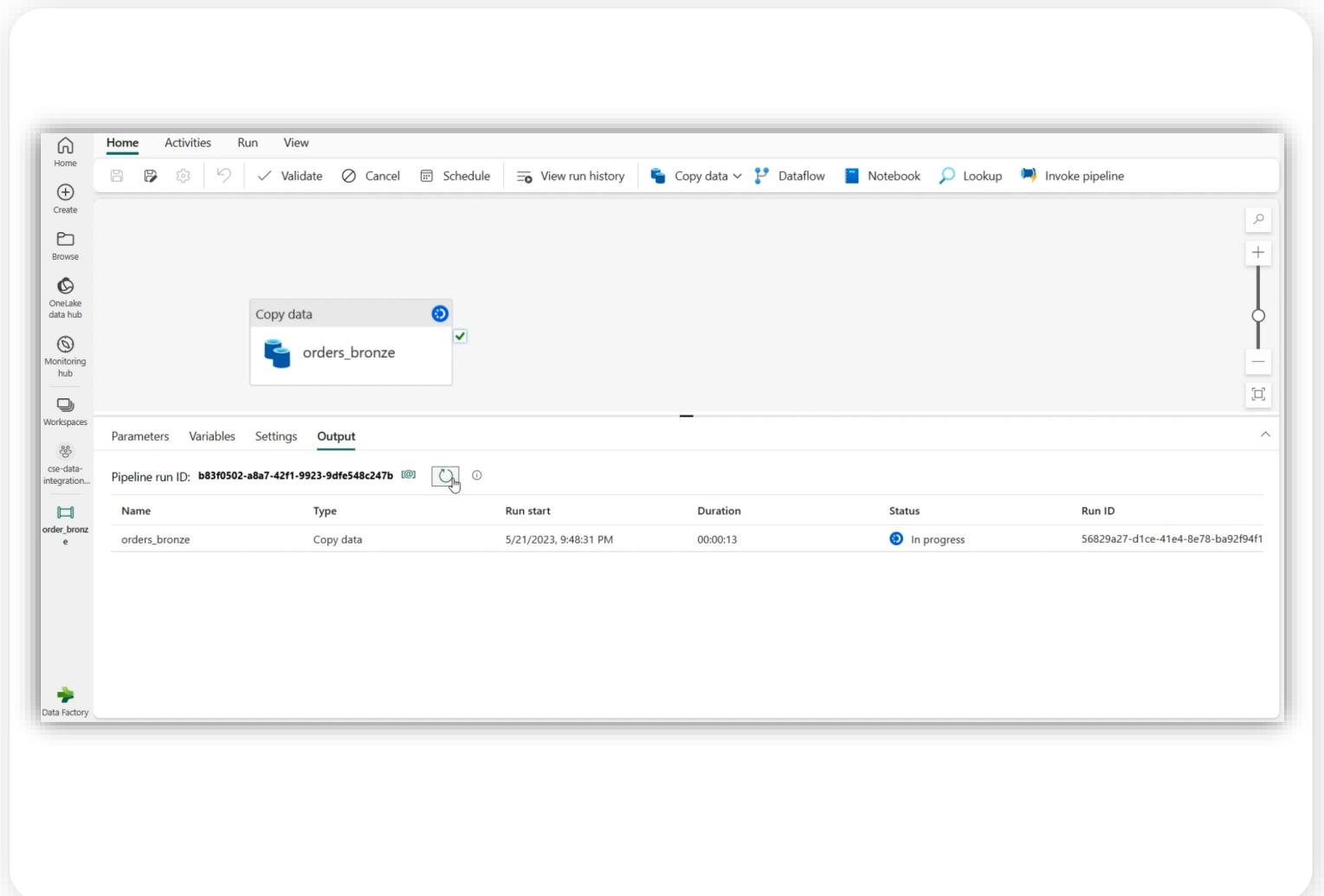
* Supported by external services / REST APIs

Troubleshooting tips

Understand how to find errors

Determine if it is a Pipeline error or an error from the Source/Destination

Set a Retry attempt



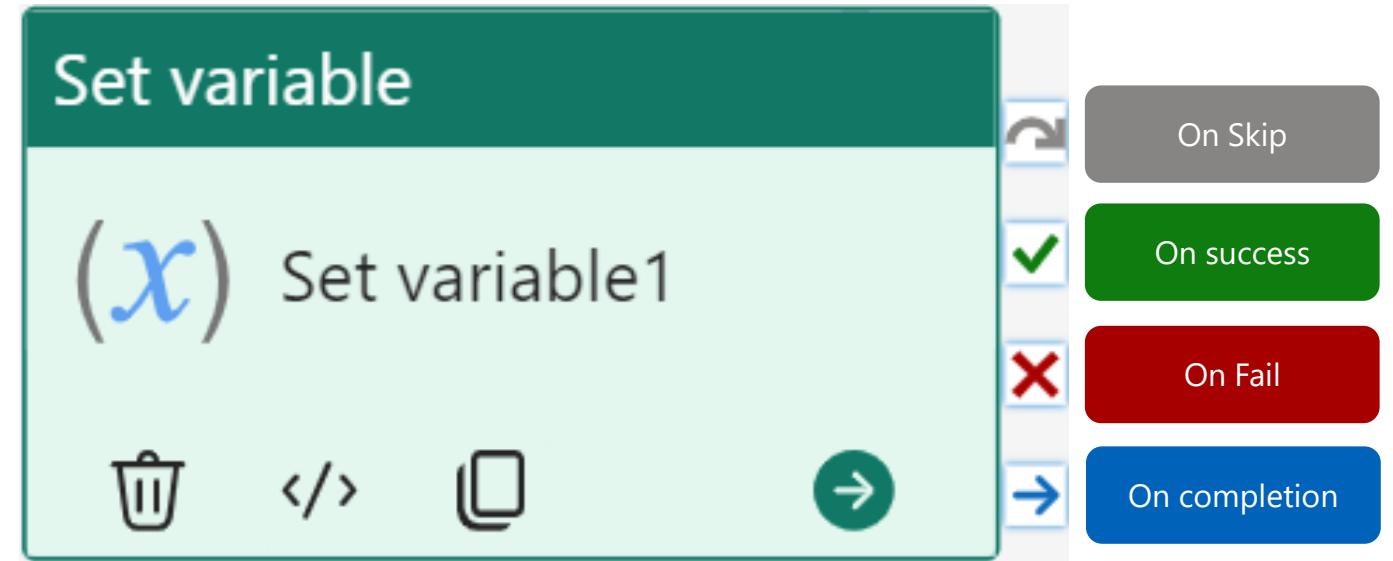
Error handling

Create conditional paths
for orchestration

Blocking dependencies

Non-blocking dependencies

Error handling



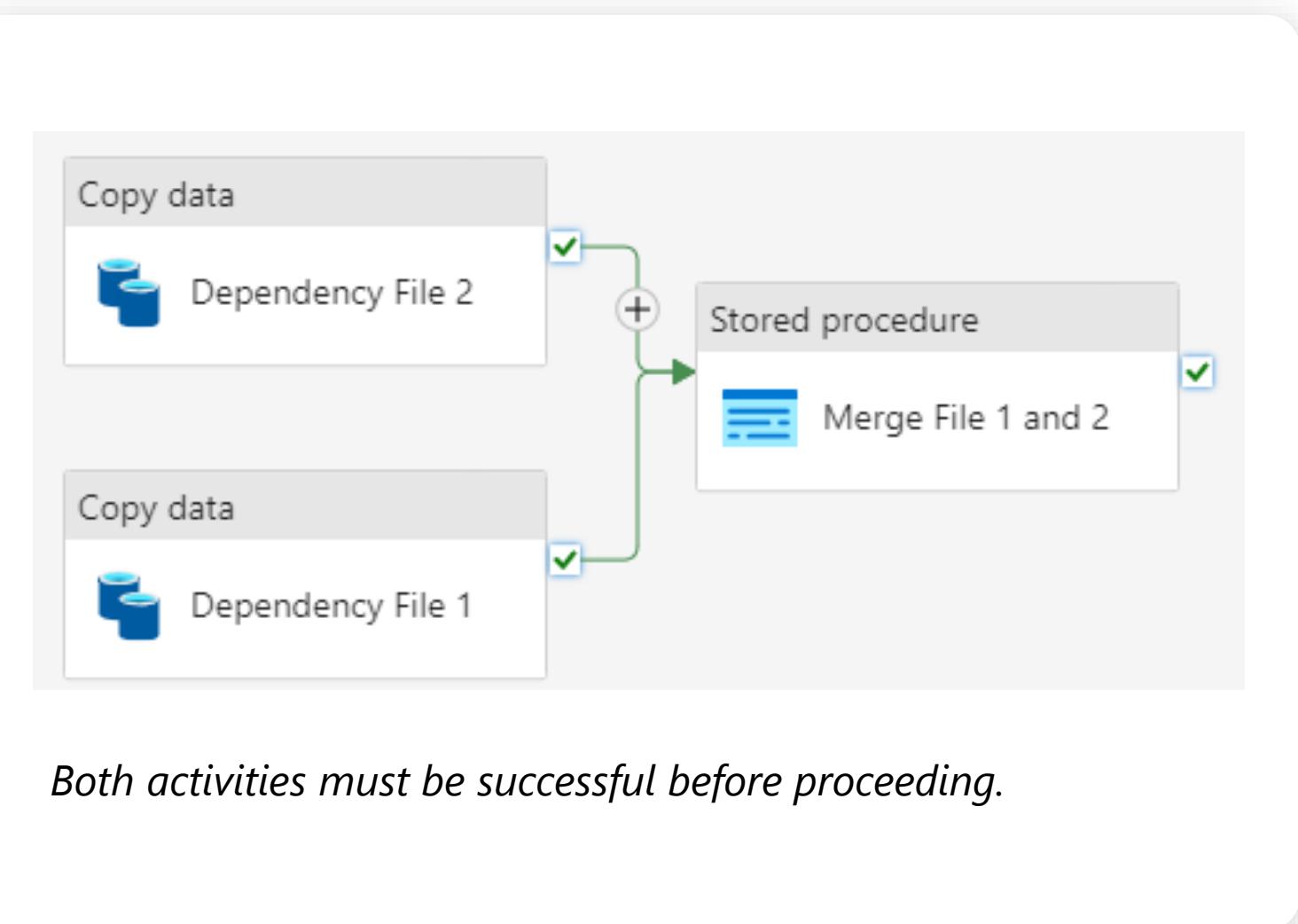
Error handling

Create conditional paths
for orchestration

Blocking dependencies

Non-blocking dependencies

Error handling



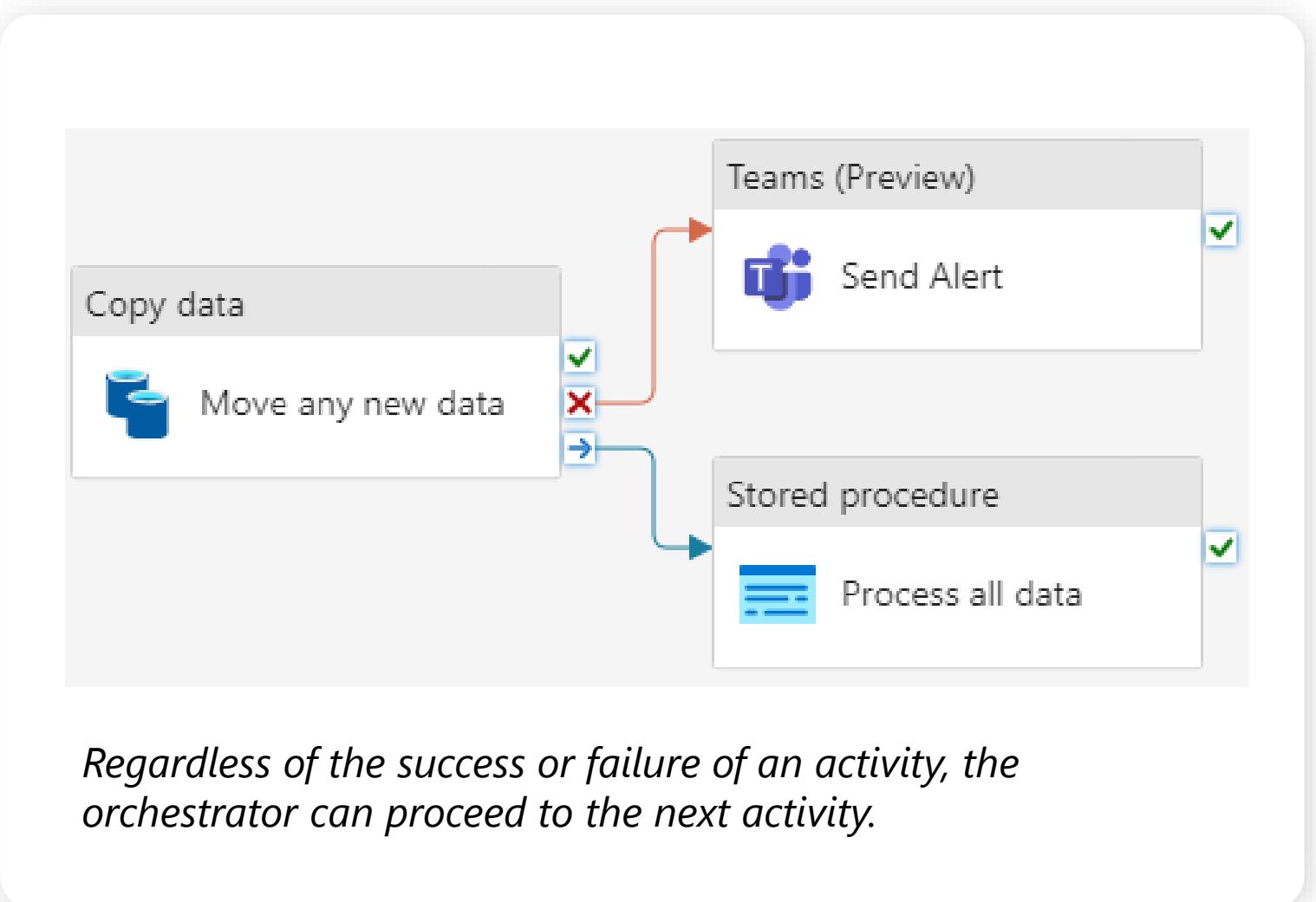
Error handling

Create conditional paths
for orchestration

Blocking dependencies

Non-blocking dependencies

Error handling



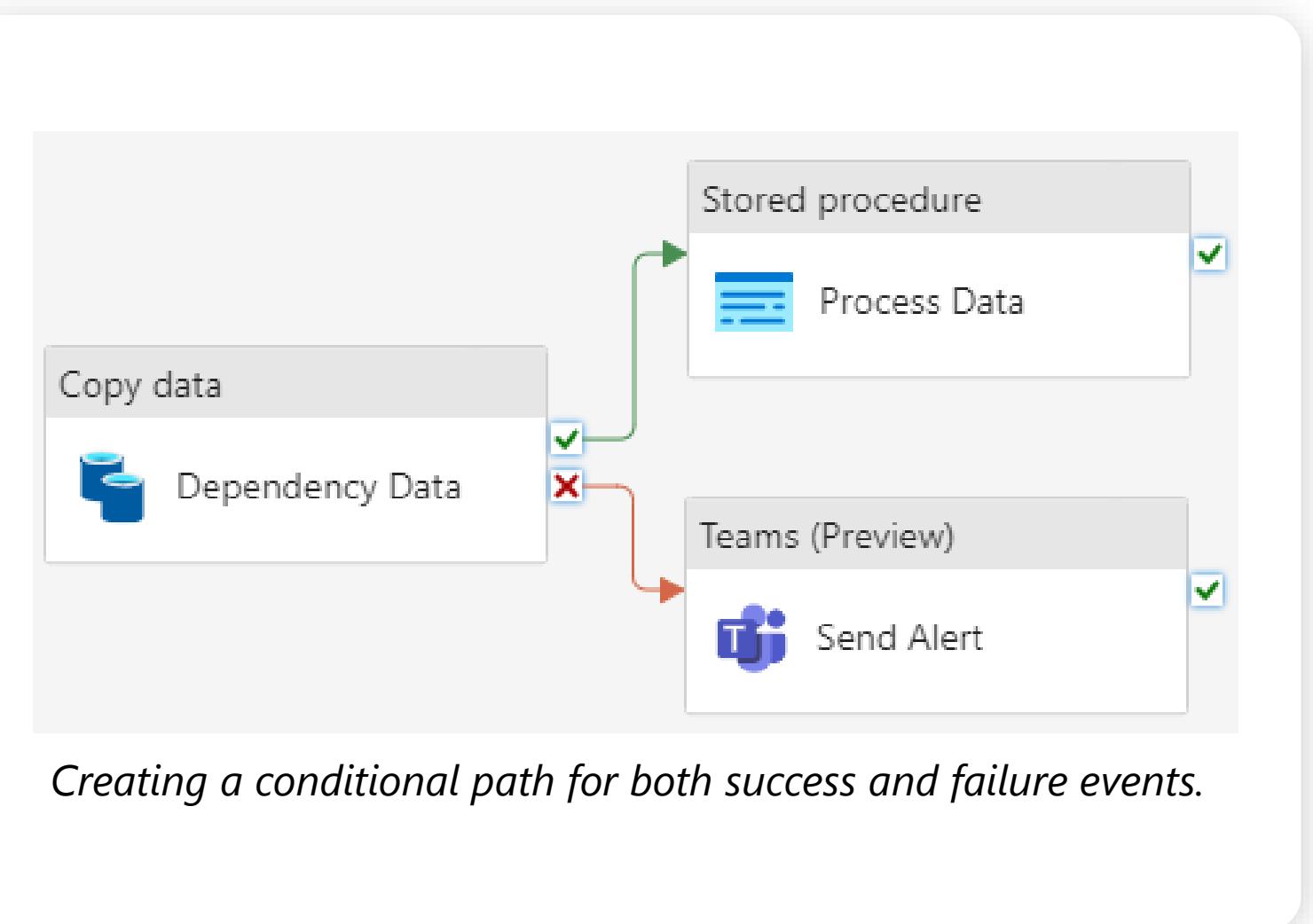
Error handling

Create conditional paths
for orchestration

Blocking dependencies

Non-blocking dependencies

Error handling



Expression builder

A powerful tool that allows you to create and manage dynamic data-driven content

Wide range of functions and operators that can be used to manipulate and transform data

Greater flexibility and reusability, and to perform advanced data transformations without the need for custom code

The screenshot shows the 'Pipeline expression builder' interface. At the top, there is a large input field with a green border containing the placeholder text 'Add dynamic content [Alt+Shift+D]'. Below this is a large gray arrow pointing upwards. To the right of the input field is a code editor window titled 'Pipeline expression builder' with the following code:

```
@string(add(mul(int(if(
    startswith(
        split(split(pipeline().parameters.interval, ':')[0],
        '.')[0]
        , '0')
    , substring(
        split(split(pipeline().parameters.interval, ':')[0],
        '.')[0]
        , 1
```

Below the code editor is a 'Clear contents' button. Underneath the code editor, there is a navigation bar with tabs: 'Parameters', 'System variables', 'Functions' (which is underlined in green), and 'Variables'. To the right of the navigation bar is a search bar with the placeholder 'Search' and a 'Search' button. Below the search bar is a list of function categories with checkboxes:

- Expand all
- Collection Functions
- Conversion Functions
- Date Functions
- Logical Functions
- Math Functions
- String Functions

Implementing a Medallion Lakehouse Architecture

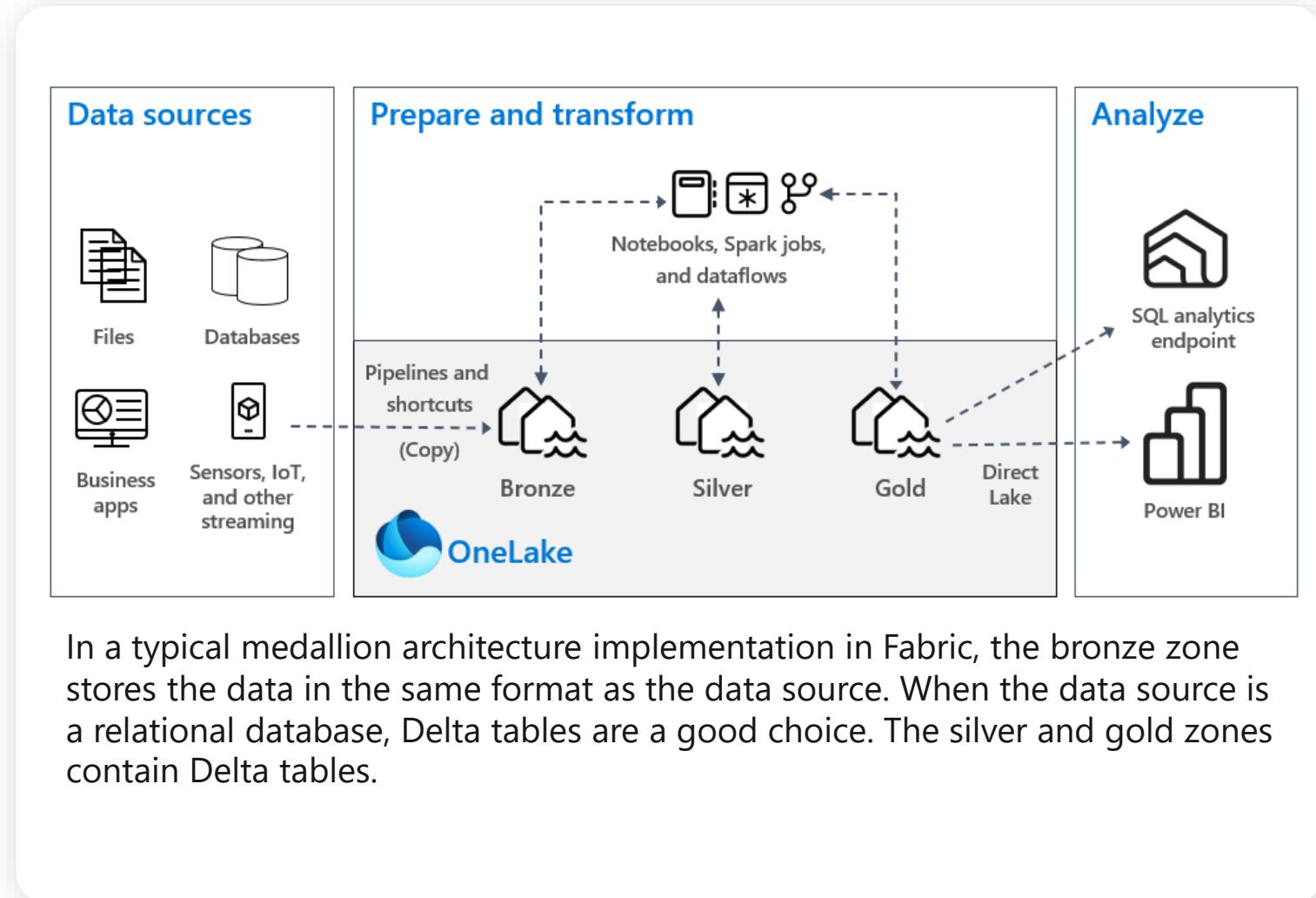


Medallion Lakehouse architecture

Build a single source of truth for enterprise data products with a multi-layered approach.

Medallion architecture comprises three distinct layers—or zones. Each layer indicates the quality of data stored in the lakehouse, with higher levels representing higher quality.

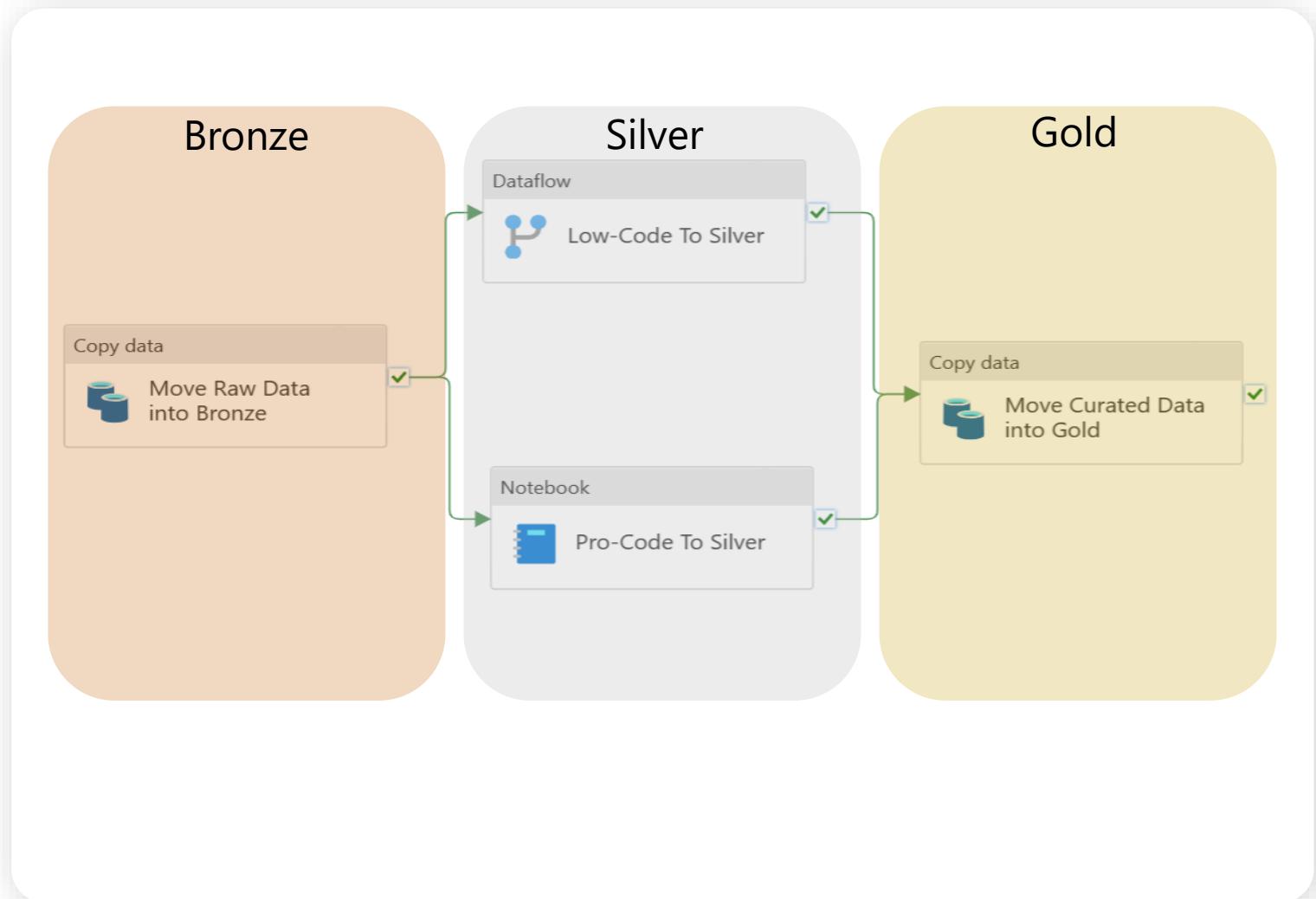
Importantly, medallion architecture guarantees the ACID set of properties (Atomicity, Consistency, Isolation, and Durability) as data progresses through the layers. Starting with raw data, a series of validations and transformations prepares data that's optimized for efficient analytics. There are three medallion stages: bronze (raw), silver (validated), and gold (enriched).



Medallion Lakehouse architecture

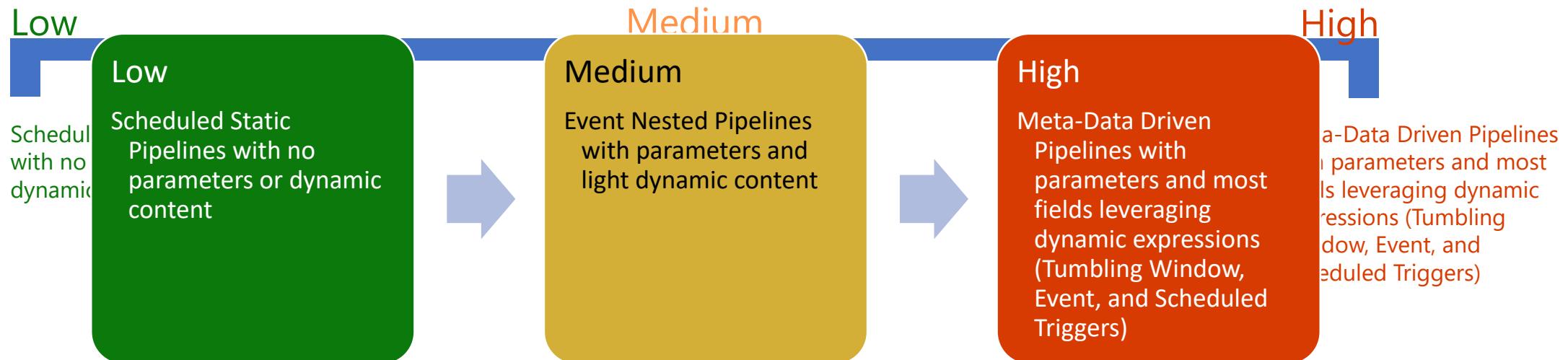
Medallion architecture consists of three distinct layers (or zones).

- **Bronze:** Also known as the *raw zone*, this first layer stores source data in its original format. The data in this layer is typically append-only and immutable.
- **Silver:** Also known as the *enriched zone*, this layer stores data sourced from the bronze layer. The raw data has been cleansed and standardized, and it's now structured as tables (rows and columns). It might also be integrated with other data to provide an enterprise view of all business entities, like customer, product, and others.
- **Gold:** Also known as the *curated zone*, this final layer stores data sourced from the silver layer. The data is refined to meet specific downstream business and analytics requirements. Tables typically conform to [star schema design](#), which supports the development of data models that are optimized for performance and usability.



Bringing it all together

Given the vastness of features, architecture designs of Data Pipelines often vary across a wide spectrum of complexity



Ingesting and Transforming Data with Dataflows Gen2

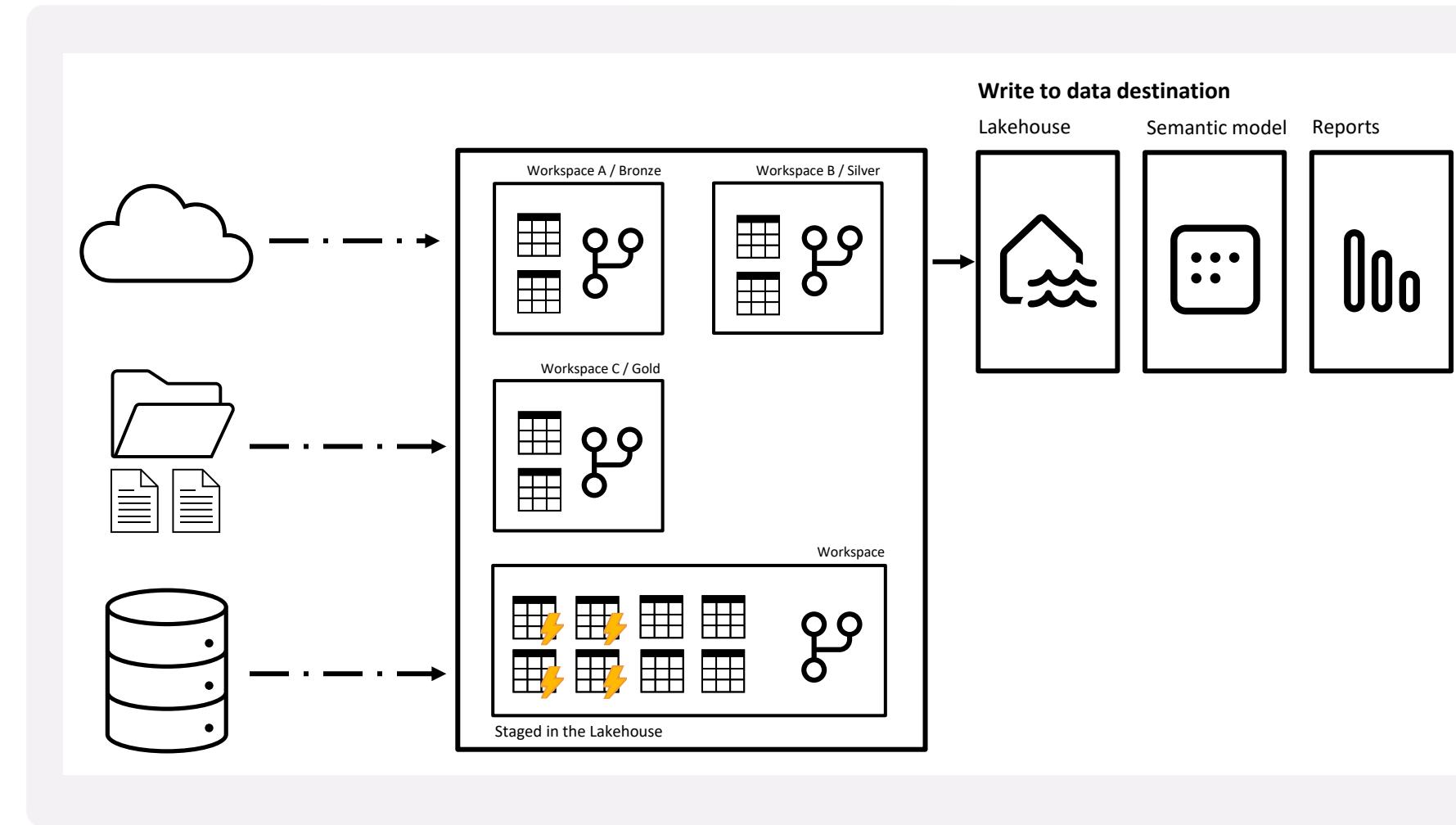


Dataflow Gen2

Next generation of
Data preparation

Dataflows are a self-service, cloud-based, data preparation technology

Dataflow Gen 2 is available in Microsoft Fabric



Dataflow Gen2

Next generation of
Data preparation

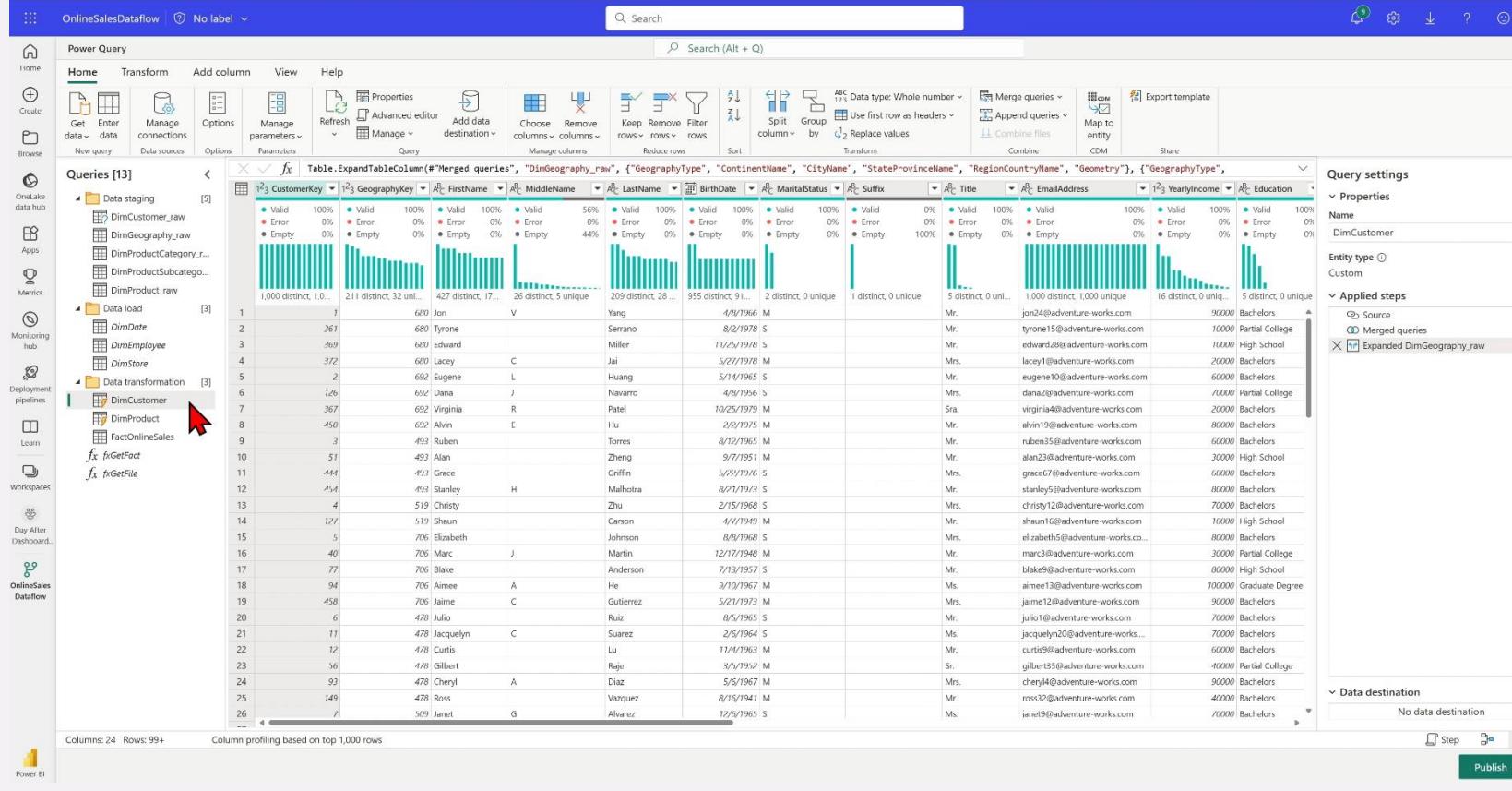
Easy to use, no-code ETL & ELT

Includes **smart AI-based** data prep

More than **300+** transformations

Output data destinations

Write output of dataflows to Azure
SQL database, Data warehouse,
Lakehouse and more



The screenshot shows the Microsoft Power Query interface with the following details:

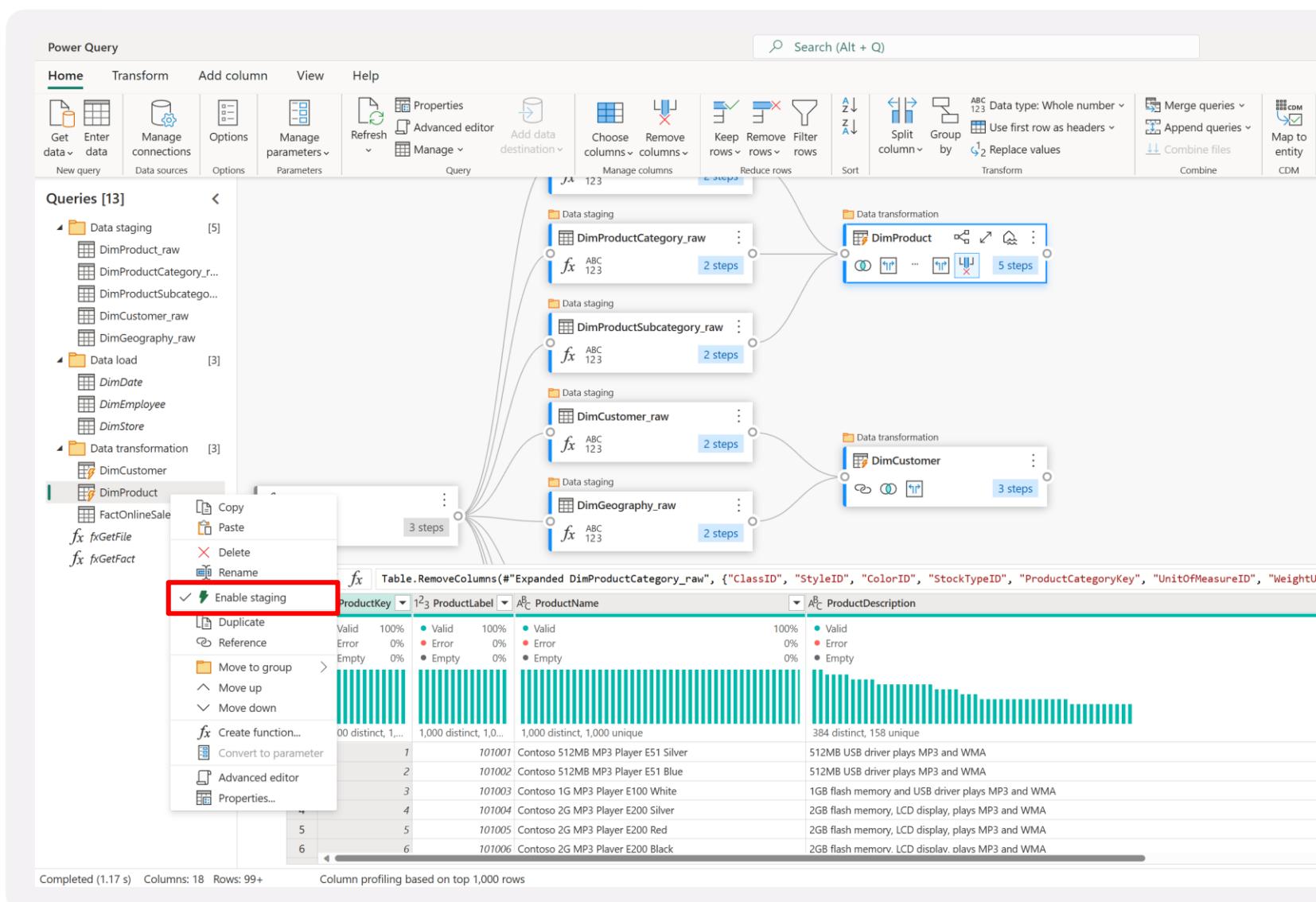
- Queries [13]:** A list of queries including Data staging, Data load, and Data transformation. The 'DimCustomer' query is highlighted with a cursor.
- Table View:** A preview of the 'DimCustomer' data with columns: CustomerKey, GeographyKey, FirstName, MiddleName, LastName, BirthDate, MaritalStatus, Suffix, Title, EmailAddress, YearlyIncome, and Education.
- Transform ribbon:** Shows various transformation tools like Get data, Transform, Add column, View, and Help.
- Power BI ribbon:** Shows Home, Transform, Add column, View, Help, and other Power BI specific options.
- Query settings pane:** Shows properties for the 'DimCustomer' query, including Name (DimCustomer), Entity type (Custom), and Applied steps (Source, Merged queries, Expanded DimGeography_raw).
- Bottom pane:** Shows the Data destination section with 'No data destination' selected.

Dataflow Gen2

Highly scalable using
Fabric compute

A **seamless experience** - yielding fast,
easy and powerful results

Abstracts away the complexities of
traditional ETL and ELT



*Previously titled "Enable load"

Dataflow Gen2

Computed tables in Staging

Queries are staged in Lakehouse/Warehouse for additional compute and scale

DataflowsStagingLakehouse	Dataset (default)
DataflowsStagingLakehouse	SQL endpoint
DataflowsStagingLakehouse	Lakehouse
DataflowsStagingWarehouse	Dataset (default)
DataflowsStagingWarehouse	Warehouse

DataflowStaging* items are an implementation detail of Dataflow Gen2.

The items aren't visible in the workspace, but might be accessible in other experiences such as the Notebook, SQL-endpoint, Lakehouse, and Warehouse experiences.

The screenshot shows the Power BI Dataflow Gen2 interface. The top navigation bar includes Home, Reporting, Table tools, Get data, New SQL query, New visual query, New report, and New measure. A message indicates a default dataset was created for reporting. The Explorer pane on the left lists Warehouses, Functions, StoredProcedures, and Tables. The Tables section contains a list of generated parquet files, each with a preview icon. The Data preview pane on the right shows a grid of two columns for 22 rows, with the first row highlighted. The bottom navigation bar includes Data, Query, and Model.

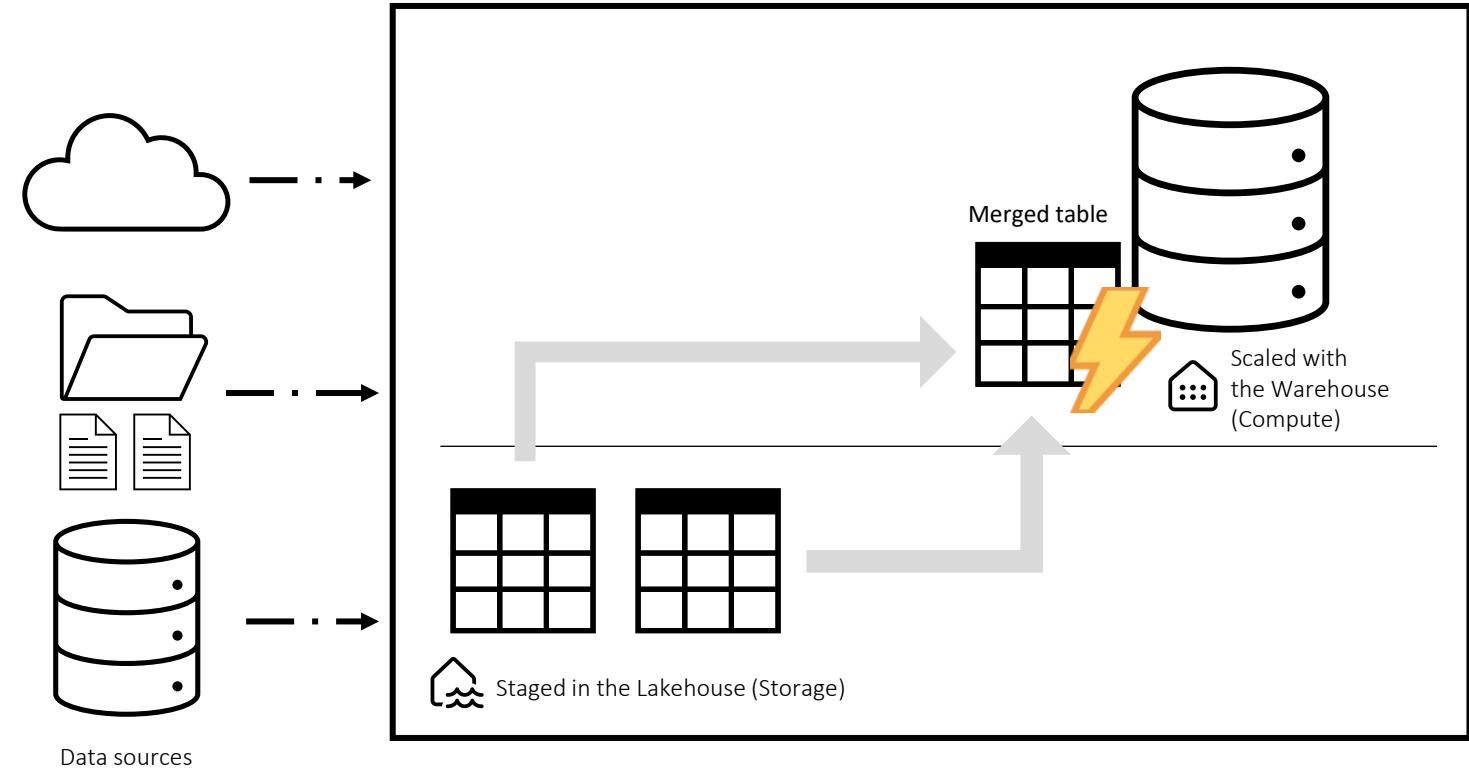
	Column1	Column2
1	445cf2d1-8abf-4bf2-b7dd-89ffc17ef03	a732c5e6-d282-47e8-83cd-c0d89e7e72ec
2	8ecc620c-363f-462c-85b7-660d4542a704	bd38e5a1-7534-405a-9a72-b34f550125ad
3	909027c4-b7c7-4d94-9313-0386cc0121f9	e00b28b0-acb6-4a09-acba-17aa20810bd9
4	50e14fc2-9583-46af-9bfb-ba88ca380125	a84b61ae-b672-4485-8d54-7531e4a6de72
5	53aa4325-dacc-40da-9770-bffa8823f2b	085c2a84-8758-493d-8f77-56cb96109263
6	ecd61159-9397-497c-8a62-f2ec7578d6c1	a44d5081-7033-4cd7-ab63-9749bb6b0554
7	4df5648d-267c-439a-abc8-3f5a006f0e0c	81e63627-90c5-44d1-9555-eb8516ea5d7b
8	766611f6-2400-48a2-b1f4-271a579f6778	21da7f00-b227-46ee-b5b5-9ef03abebd8
9	a2a4c713-16f8-49ce-9ea9-51ab1675ff	331c9917-299e-4e82-91a9-da5e138fc998
10	67788c40-57c6-4b1b-a37e-2c9d042452cb	0a959627-9757-42d5-9265-99448b132d0c
11	86183731-50f3-4692-9516-fb641d492137	e26036a1-8f9c-479c-8318-3ef3ffe9b997
12	601d300a-196c-49ef-b032-041d597e399e	940764bd-b24b-47d6-a7ae-cf28ee836bf2
13	d43c4b9d-958d-4239-9507-644dad2f8dd7	0aa9d40c-814d-4a36-a6cb-37baa04ca952
14	347f6d07-fe9a-4a3c-9592-848b169b79a2	8731f589-2680-42a9-a2fa-b0e4c9a6140a
15	1b6df66b-f874-47bb-a5b9-fe8d33e50967	886dd381-ef54-446d-82e3-10b6640ba461
16	e797be81-7826-4b50-afb7-bff131d85f8e	33498135-110c-449e-9d8b-ad714bedc442
17	64e5c4d4-0405-490a-80d8-d92ff298de78	4f249590-1ded-4c72-ad70-1bb62ceaa035
18	b7945ab8-ebf8-42e6-beef-c019826760b1	871fa65b-774e-49f2-a779-c98b4192ba88
19	973bfae8-8815-495a-901d-6a1fdafc67c0	63cac59e-bbe1-4475-87e0-603537cb8093
20	a7d82782-3d5e-412e-a414-3c24f7f16aaa	8cf0e451-ed53-4dab-aac7-6f75fd81fc5
21	182e4a9f-ce41-423d-ae63-855f94170ba9	6a520e96-d8db-4eec-aab8-a917643c71e6
22	d558a3a4-a3a7-4b68-87b8-50e3643c32ff	011f5f0-a7e8-4ca1-89e6-26cc58c4f25

Succeeded (2 sec 713 ms)

Dataflow Gen2

Optimize the use of dataflows with Fabric compute

1. Connect to your data and copy it into the Lakehouse using *“Enable staging” (**on by default**)
2. Create a reference query in a new query.
3. Apply transformation steps to the computed table for complex ETL operations such as join, distinct, filter and group by – leveraging the Warehouse for compute.



*Previously titled “Enable load”

Dataflow Gen2 | Check your knowledge

A single dataflow has a limit of how many queries/tables?

25

50

100

Dataflow Gen2 | Check your knowledge

A dataflow can write to a Lakehouse's Delta Table and Files section.

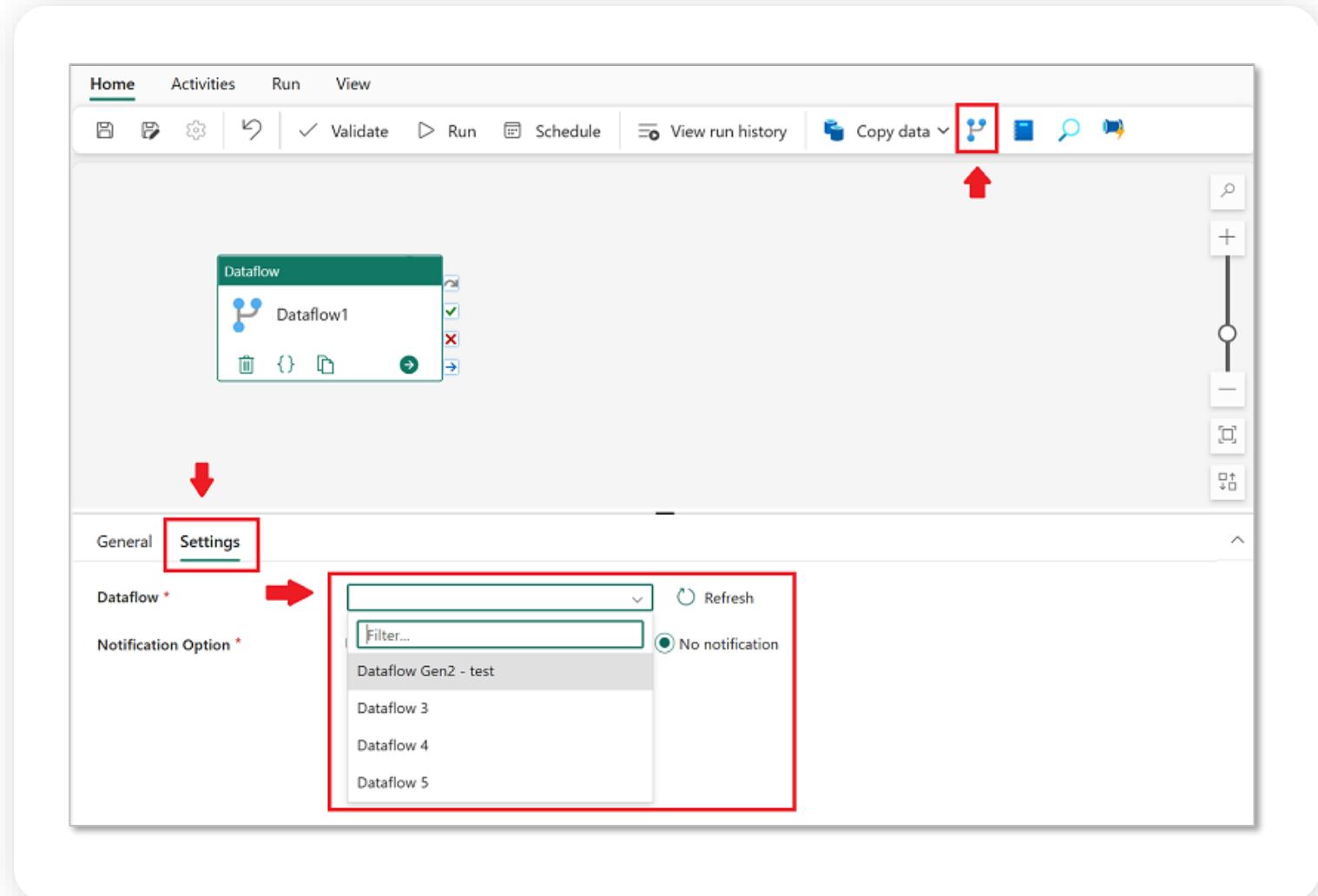
TRUE

FALSE

Integrating Dataflows (Gen2) and Pipelines

A dataflow for data ingestion and transformation, and landing into a Lakehouse using dataflows

Then incorporate the dataflow into a pipeline to orchestrate additional activities



Dataflow Gen2 scenarios

Separate dimension tables and fact tables into separate dataflows. Orchestrate the refresh in a data pipeline.

Separate tables with and without staging into separate dataflows. Orchestrate the refresh in a data pipeline.

Ingest and transform all your tables in a single dataflow. Use dataflow refresh scheduler.

Separate tables based on the data source and if they do or do not support query folding.

Use copy activity in data pipelines to ingest data. Dataflows to transform and write data.

Additional considerations...

- Orchestrate the refresh in a data pipeline.
- Use the dataflow refresh scheduler.

Fast Copy

Enables the richness of the Data pipelines Copy activity within the Get Data experience using Power Query in Dataflows Gen2

Abstracts away complexities of traditional ETL and ELT

The screenshot shows the Microsoft Power BI Dataflows Gen2 interface. At the top, the title bar reads "Untitled 1 | Data updated 10/20/22". The ribbon menu includes Home, Transform, Add Column, View, Help, and Editing. A search bar is at the top right. The main area is titled "Queries [2]" and shows two data flows:

- Customers - Staged**: The source is a flat file icon, and the navigation step is a cloud icon.
- Customers**: The source is a flat file icon, and the navigation step is a cloud icon. It includes options for Promoted headers, Changed columns, Filter rows, and Add step.

Below the queries, a preview pane displays a table with 100 rows of customer data. The columns are labeled: CustomerID, NameStyle, Title, FirstName, MiddleName, LastName, Suffix, and Company. The first few rows of data are:

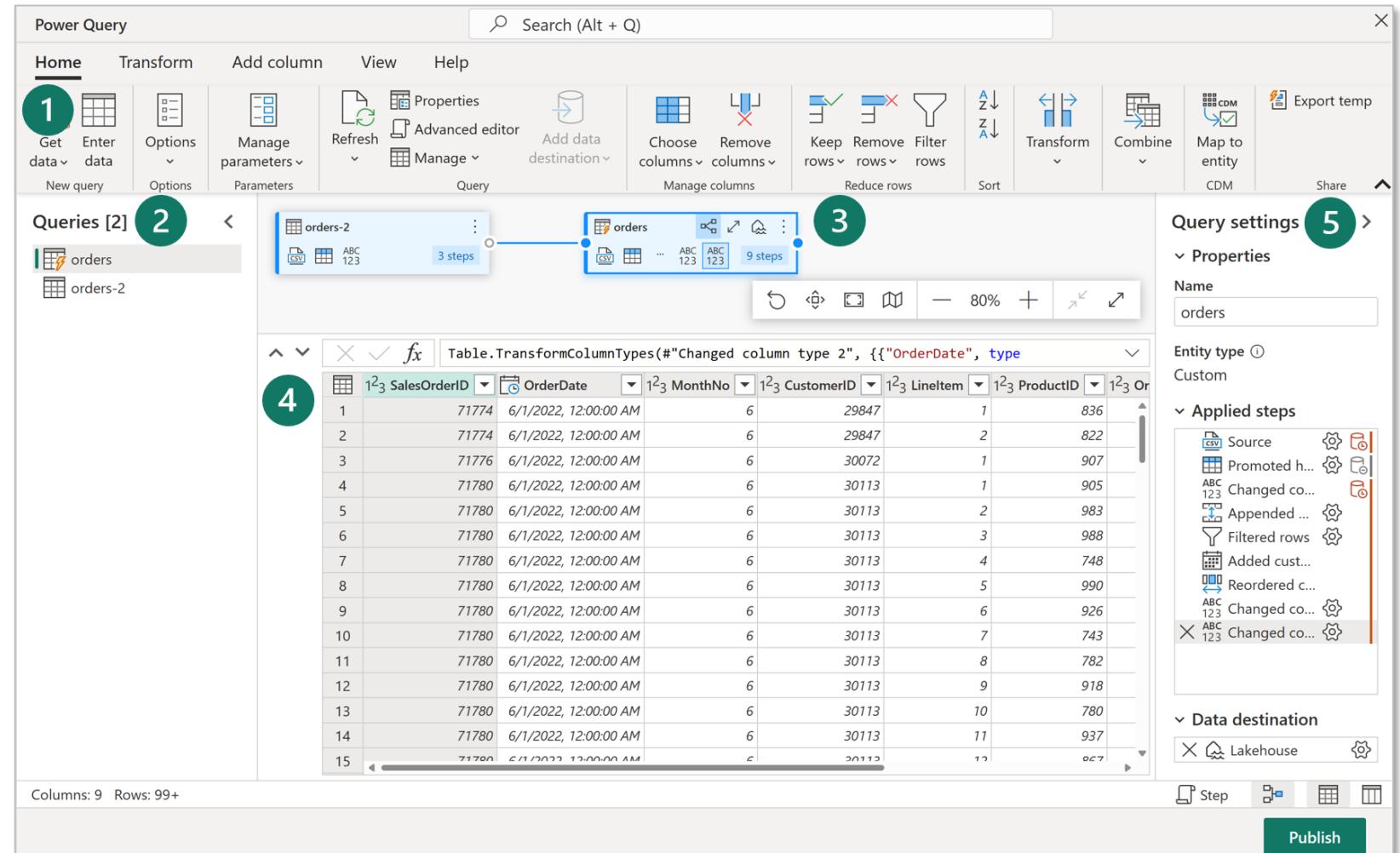
CustomerID	NameStyle	Title	FirstName	MiddleName	LastName	Suffix	Company
1	29524	FALSE	Mr.	Jay			Adams
2	29560	FALSE	Mr.	François			Ferrier
3	29565	FALSE	Mr.	John			Arthur
4	29574	FALSE	Mr.	Chris			Ashton
5	29580	FALSE	Mr.	Matthias			Berndt
6	29582	FALSE	Mr.	Jimmy			Bischoff
7	29583	FALSE	Mr.	Randall			Boseman
8	29590	FALSE	Mr.	Richard			Bready
9	29591	FALSE	Mr.	Ted			Bremer
10	29598	FALSE	Mr.	Alan			Brewer
11	29623	FALSE	Mr.	Jo			Brown
12	29635	FALSE	Mr.	Robert			Brown
13	29636	FALSE	Mr.	Michael			Brundage
14	29645	FALSE	Mr.	Chris			Cannon
15	29653	FALSE	Mr.	Rob			Caron
16	29654	FALSE	Mr.	Andy			Carothers
17	29660	FALSE	Mr.	Andrew			Cencini
18	29662	FALSE	Mr.	Paul			Sports Merch

At the bottom, a status bar indicates "Completed (0.86 s) Columns: 7 Rows: 9+".

Exploring Dataflows (Gen2)

Power Query editor

1. Ribbon
2. Queries pane
3. Visual diagram
4. Data preview grid
5. Query settings



DEMO

Dataflow (Gen2)



Migrate from Power BI dataflow to Fabric dataflow

Feature	Power BI Dataflow	Dataflow Gen2
Output file format	CSV	Parquet (V-Ordered)
Fast copy	No	Yes
Data destination output	No	Yes
Premium capacity required	No	Yes
AI Insights	Yes	No
AutoML	Deprecated	No
Attach Common Data Model (CDM) folder	Yes	No
Linked Tables	Yes	No (use Shortcuts)

Metadata-Driven design benefits

- Configuration Driven
- Scalable
- Adaptable
- Independent Components



Metadata-Driven design concepts

Metadata Management – Empower users to configure new data models

Logging – Leveraging native and custom logging for detailed insights and performance metrics

Alerting – Point in Time notifications directly to MS Teams or Outlook

Lineage - End to end visibility of processes and operations

Extendibility – Easily integrate additional features



Metadata-Driven design components

- **Metadata Storage**
 - History, Audit
- **Custom Logging Framework**
 - Status, Rollback, Performance
- **Custom Alerting Framework**
 - MS Teams & Outlook
- **Isolation and compartmentalization of tasks**
 - Independent components that can operate together
- **Medallion Architecture**



Thank you