Microsoft

Data Factory in a Day

with Microsoft Fabric

Microsoft Fabric overview

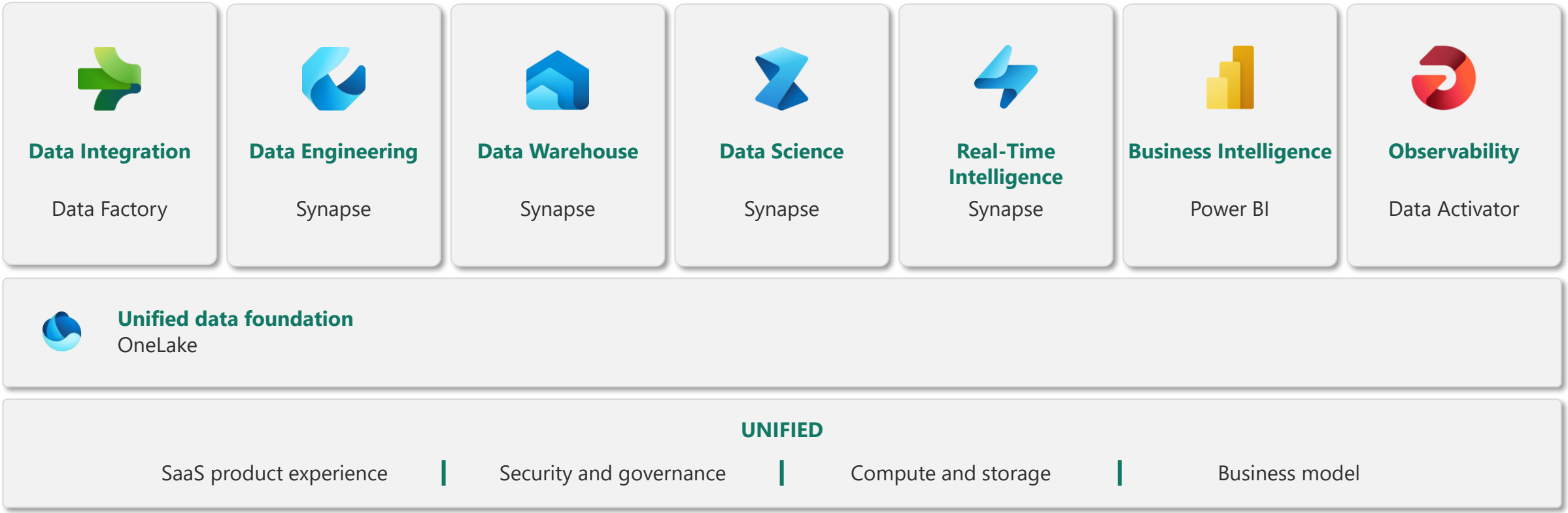
Microsoft Fabric overview

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

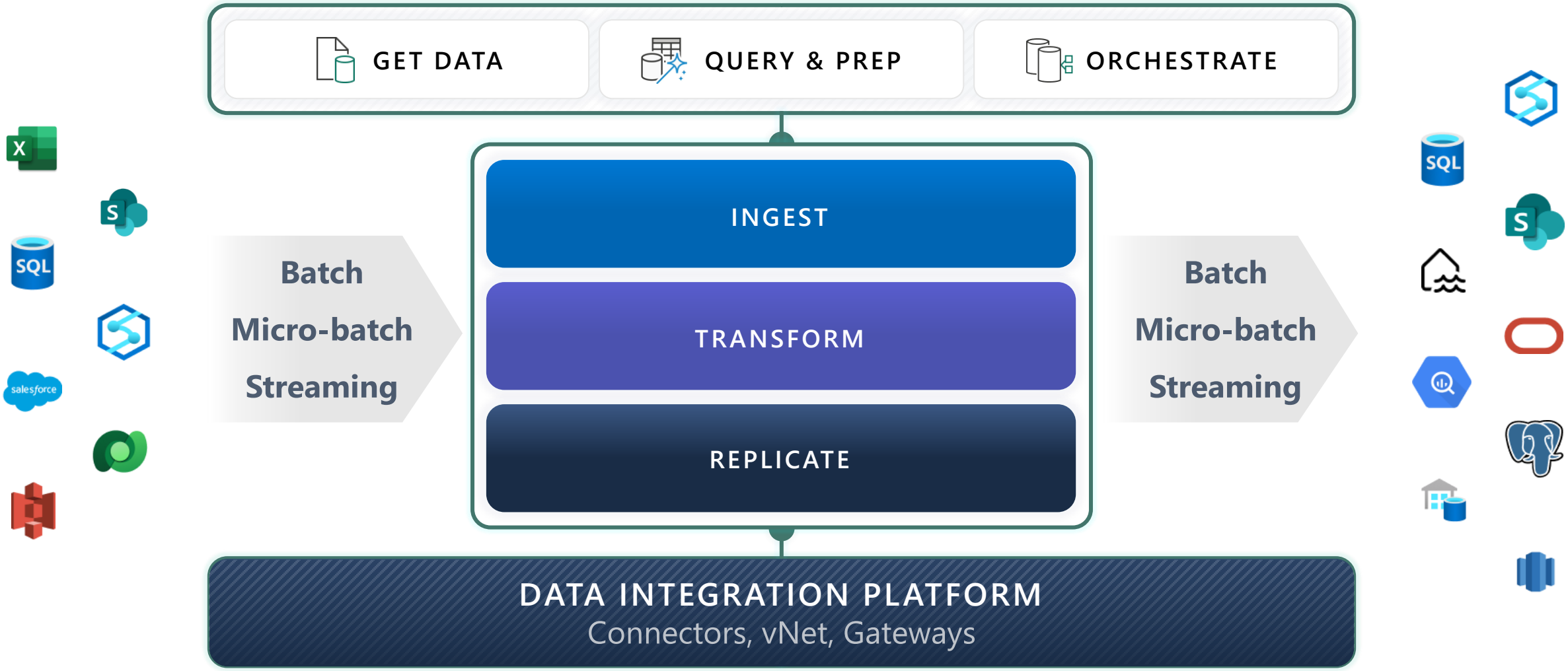
Microsoft Fabric does it all—in a unified solution

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 Factory overview

Microsoft Data Integration

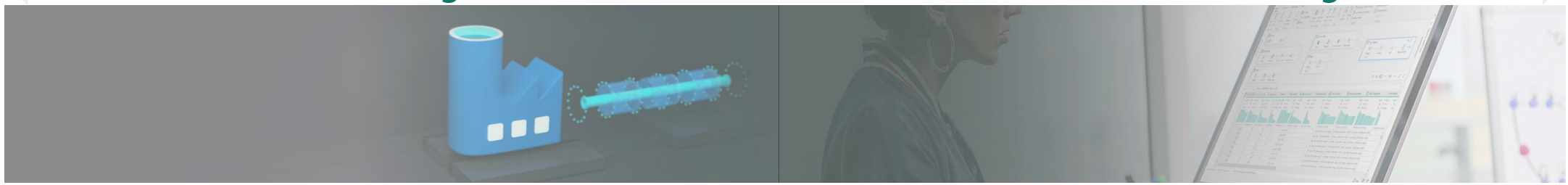


Microsoft Data Integration

Products

Professional Data Integration

Citizen Data Integration



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

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

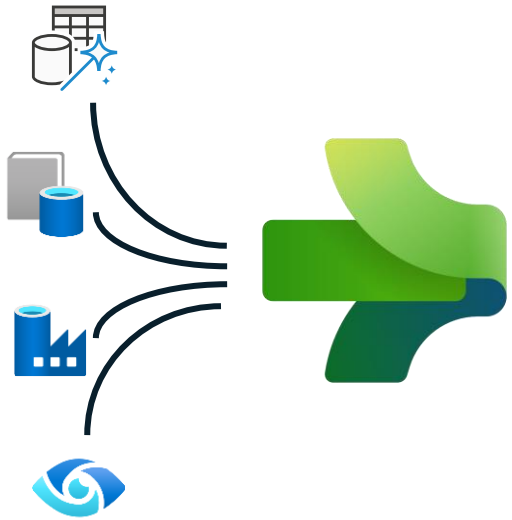
Professional & Citizen Data Integration

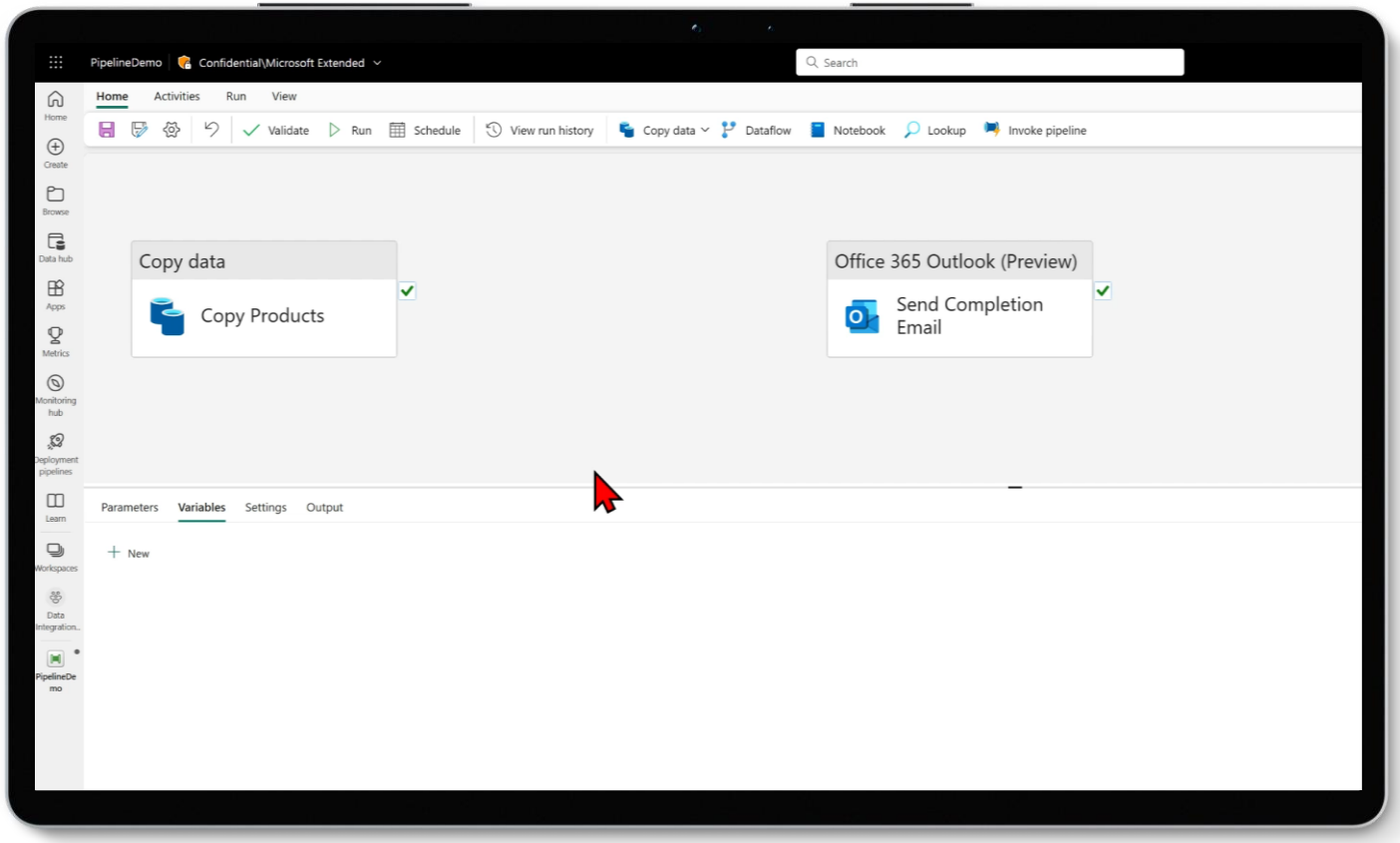
Data Factory in Microsoft Fabric

- 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.**



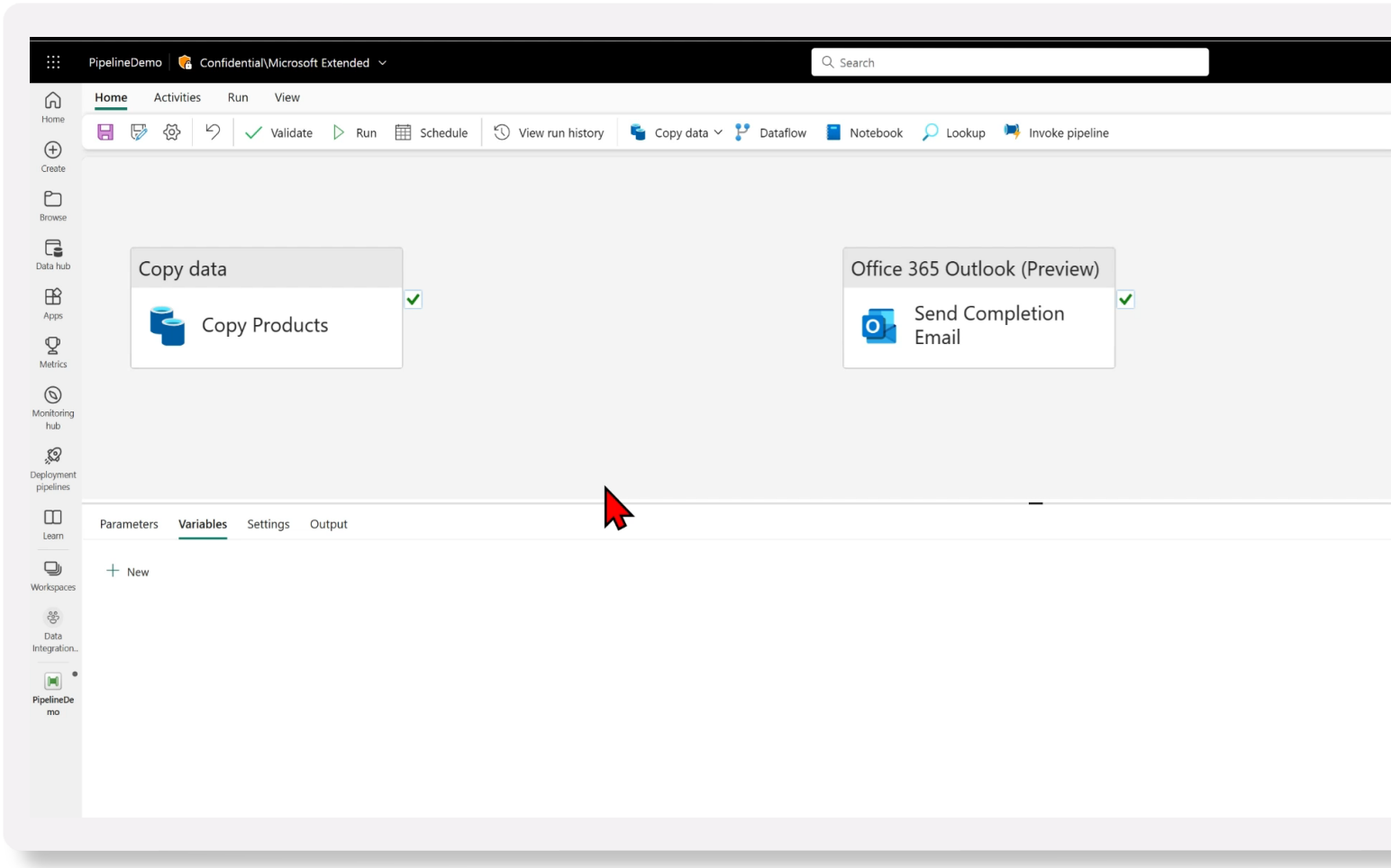


Data Integration for everyone

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..**



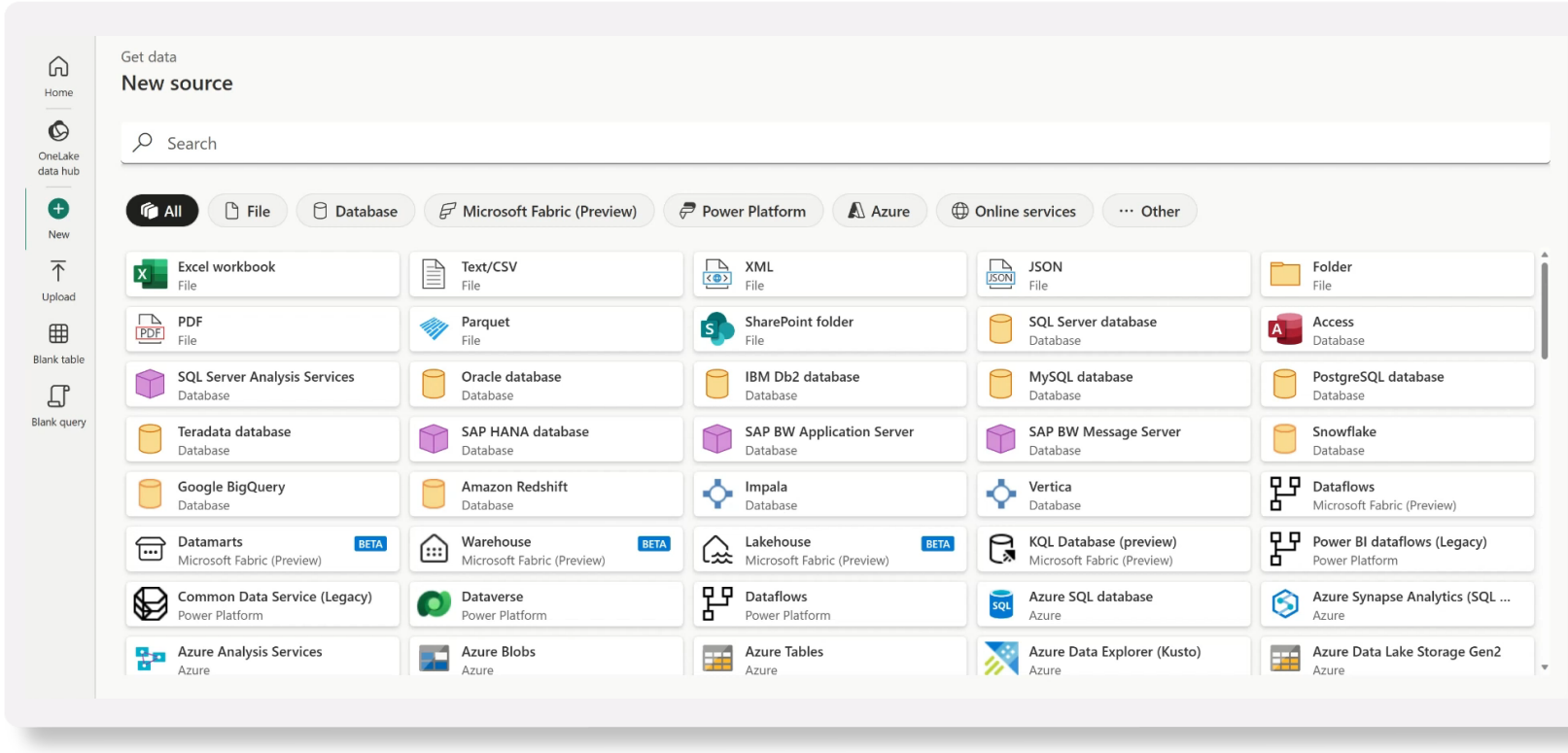
Rich connectivity

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



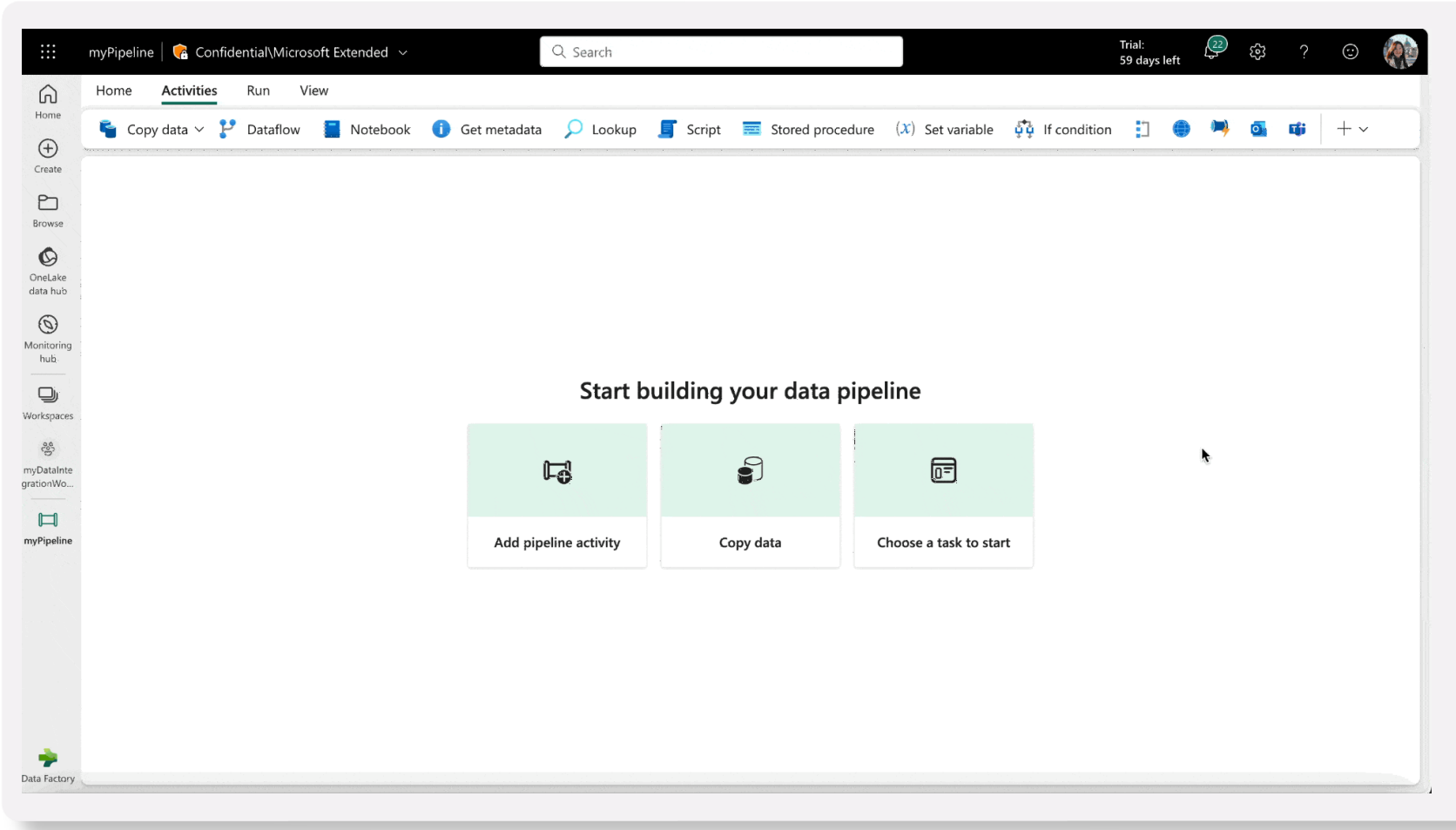
Data pipelines

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

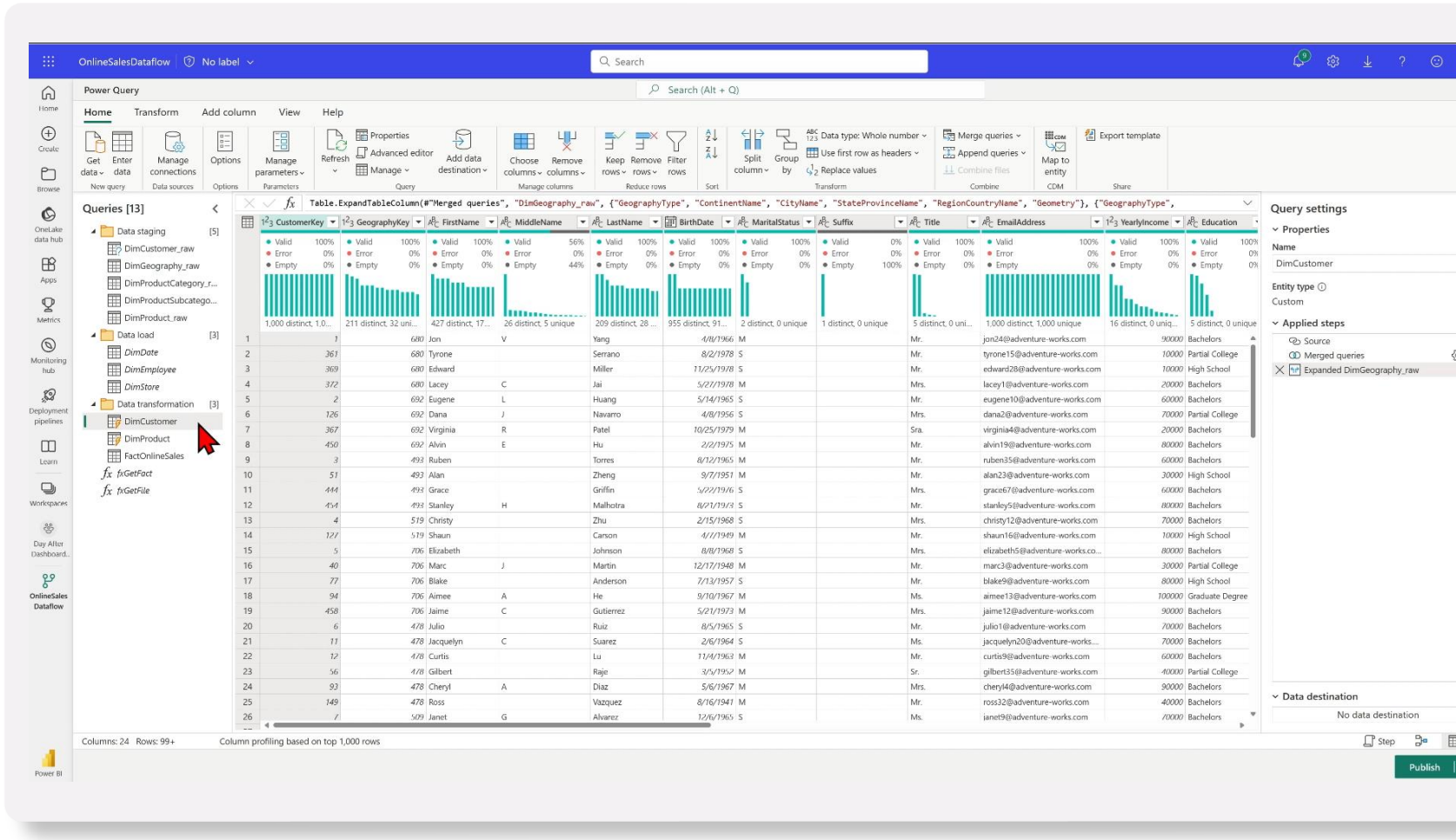
Next generation of **Data preparation**

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

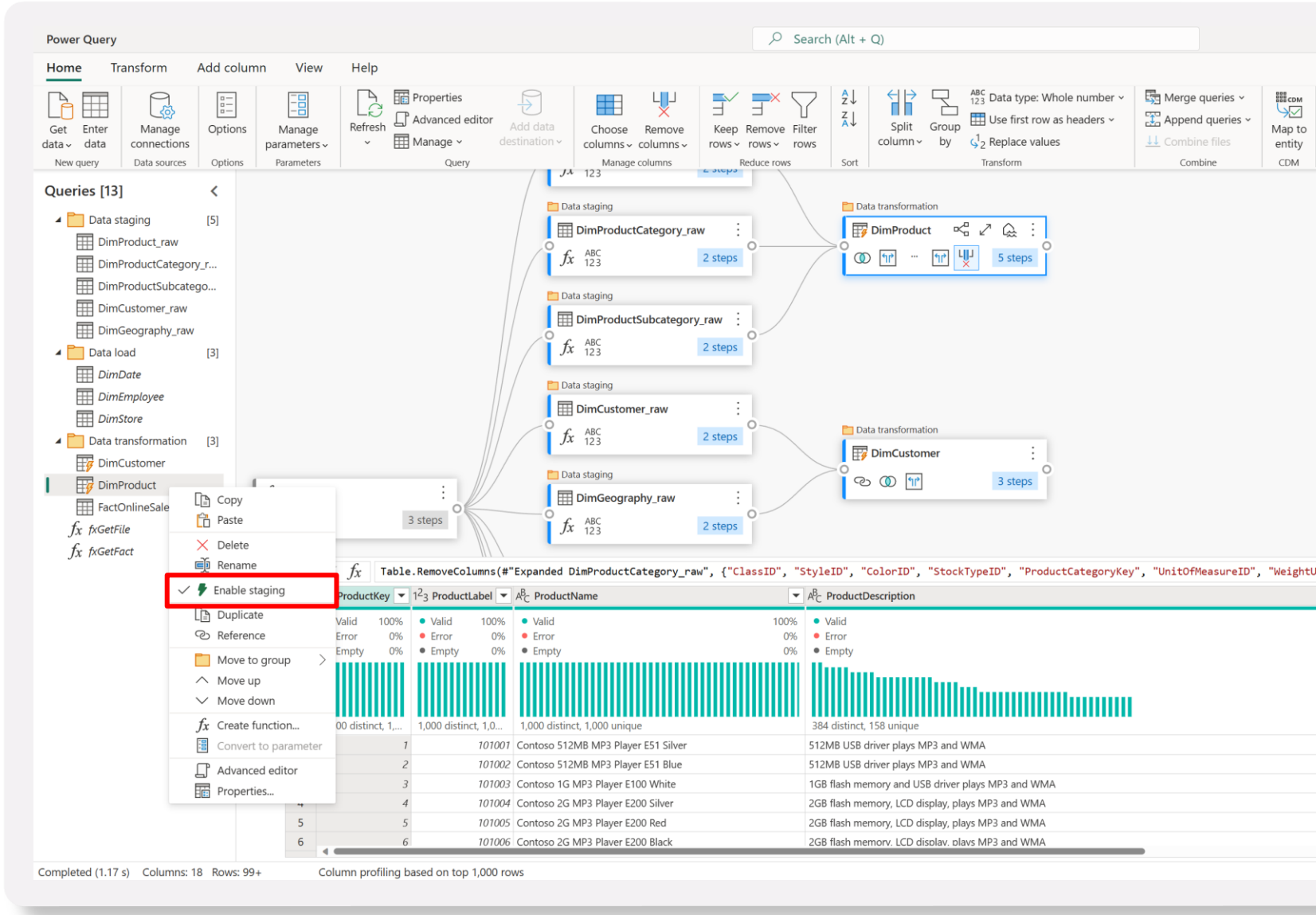


Dataflow Gen2

Highly scalable using **Fabric compute**

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

Abstracts away the complexities of traditional ETL and ELT

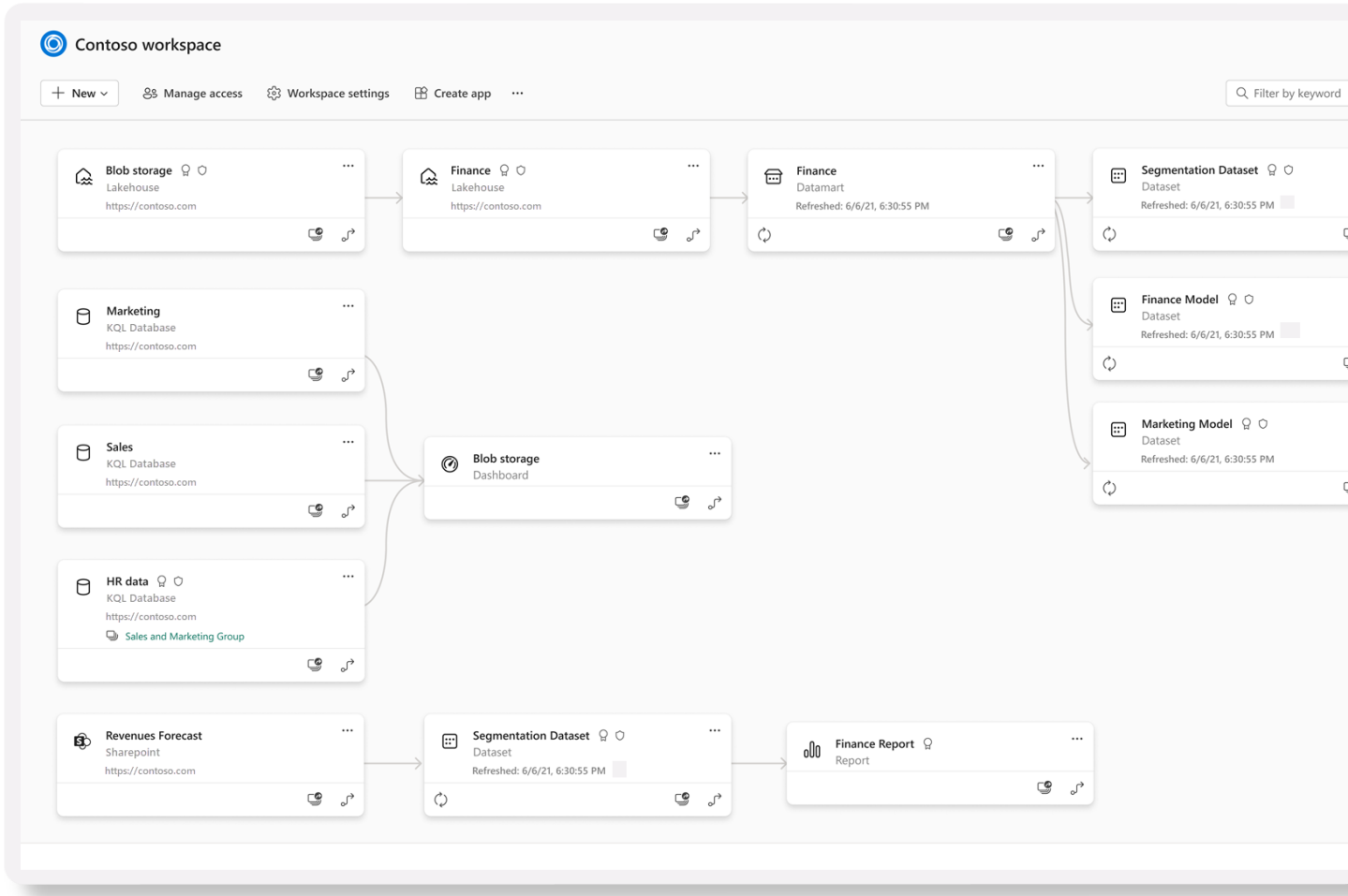


Lineage

Understand the **relationships** between analytical solutions

Available automatically with every workspace

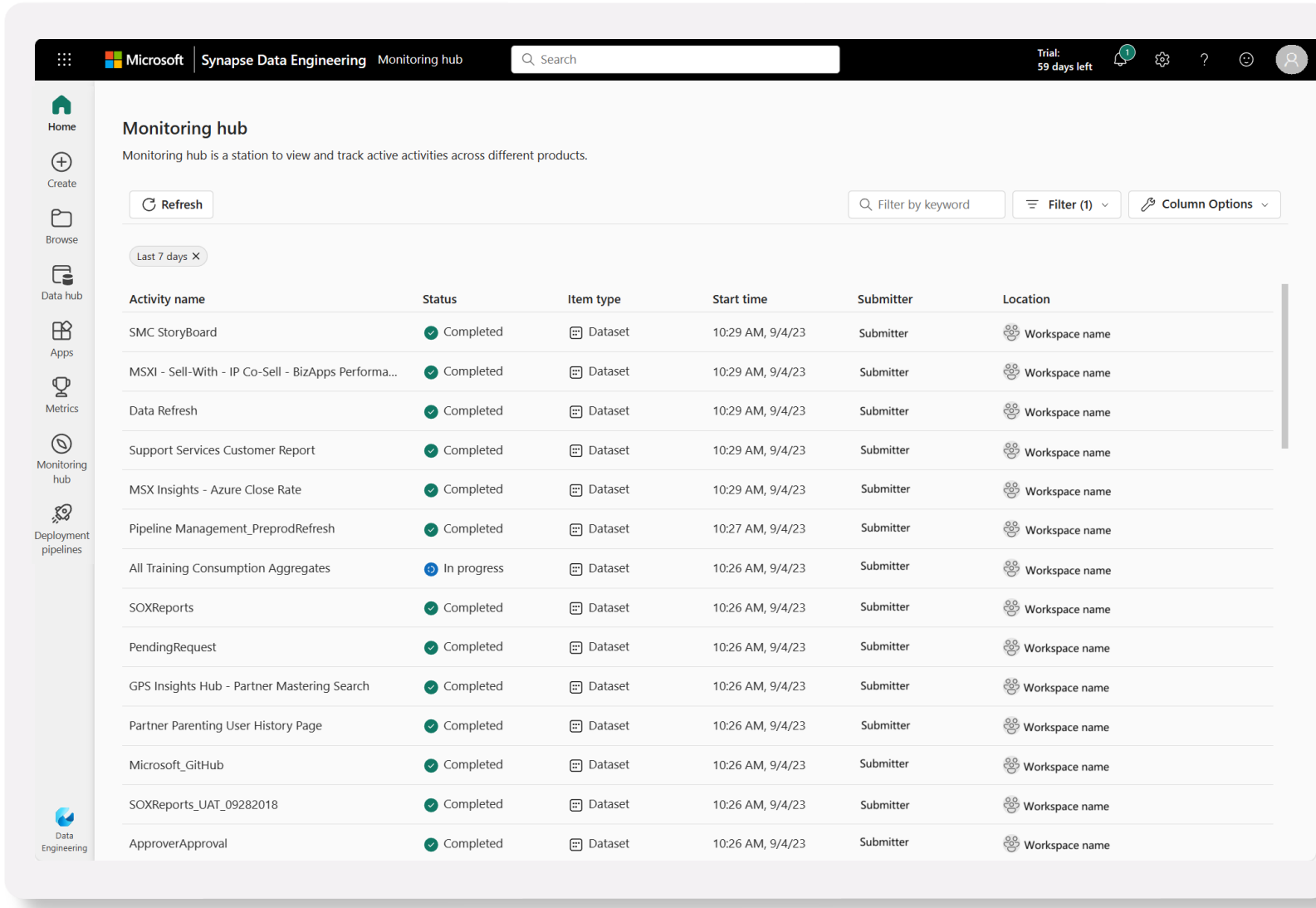
Assess the potential impact of change on downstream items using **impact analysis**



Monitoring hub

Monitor activities from a **central location**

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

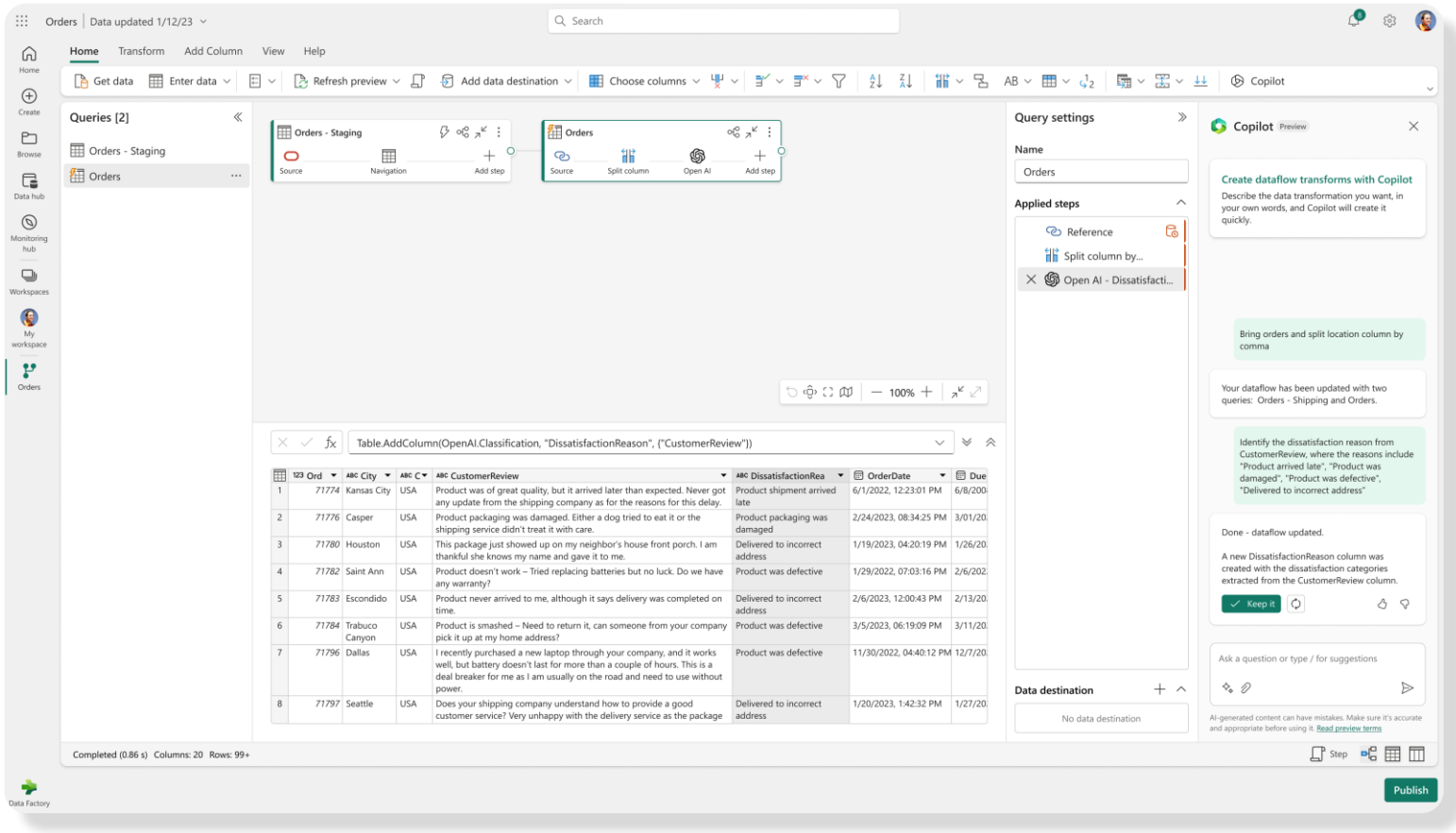


Data integration + AI

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

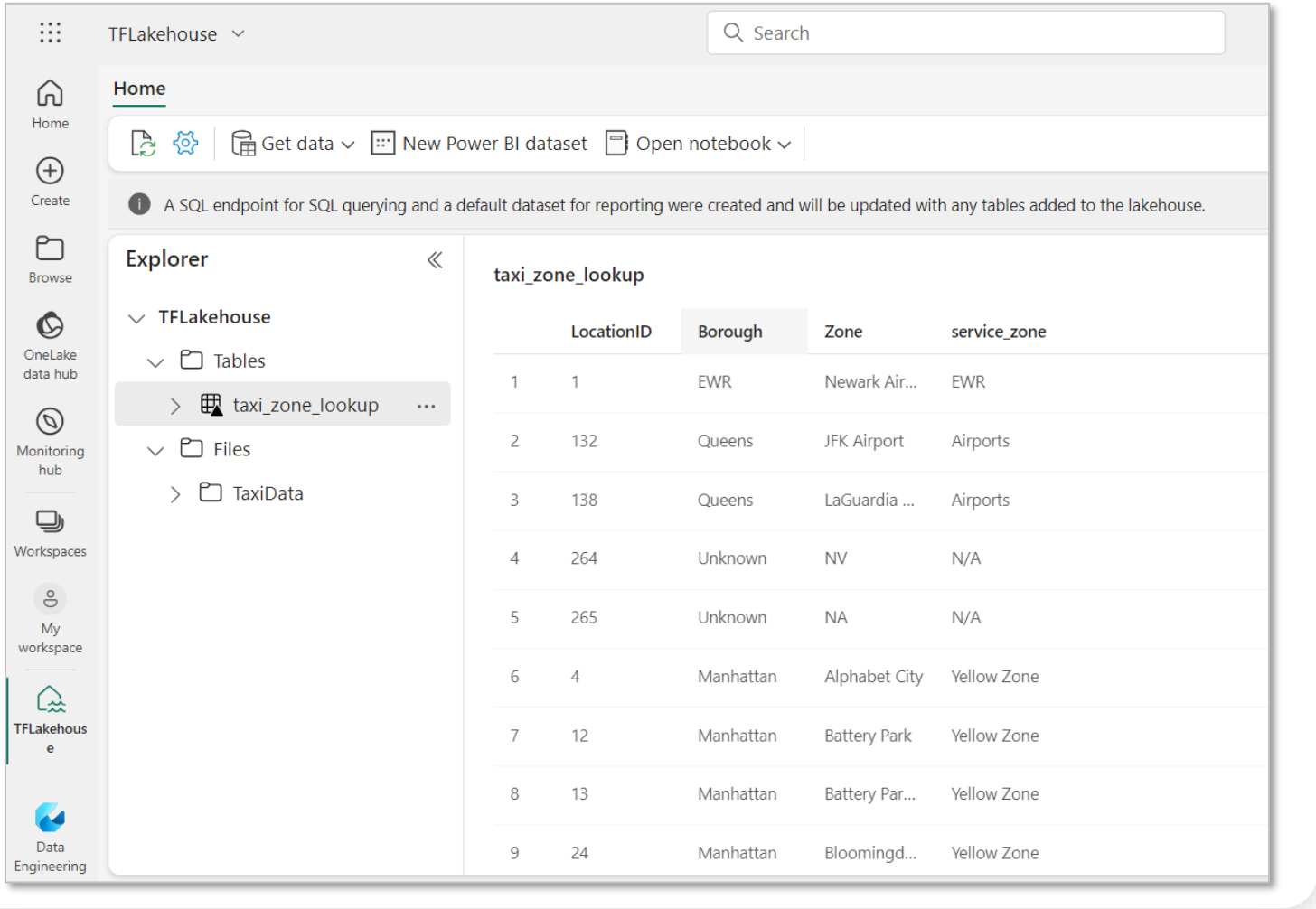


Lakehouse overview

Lakehouse overview

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

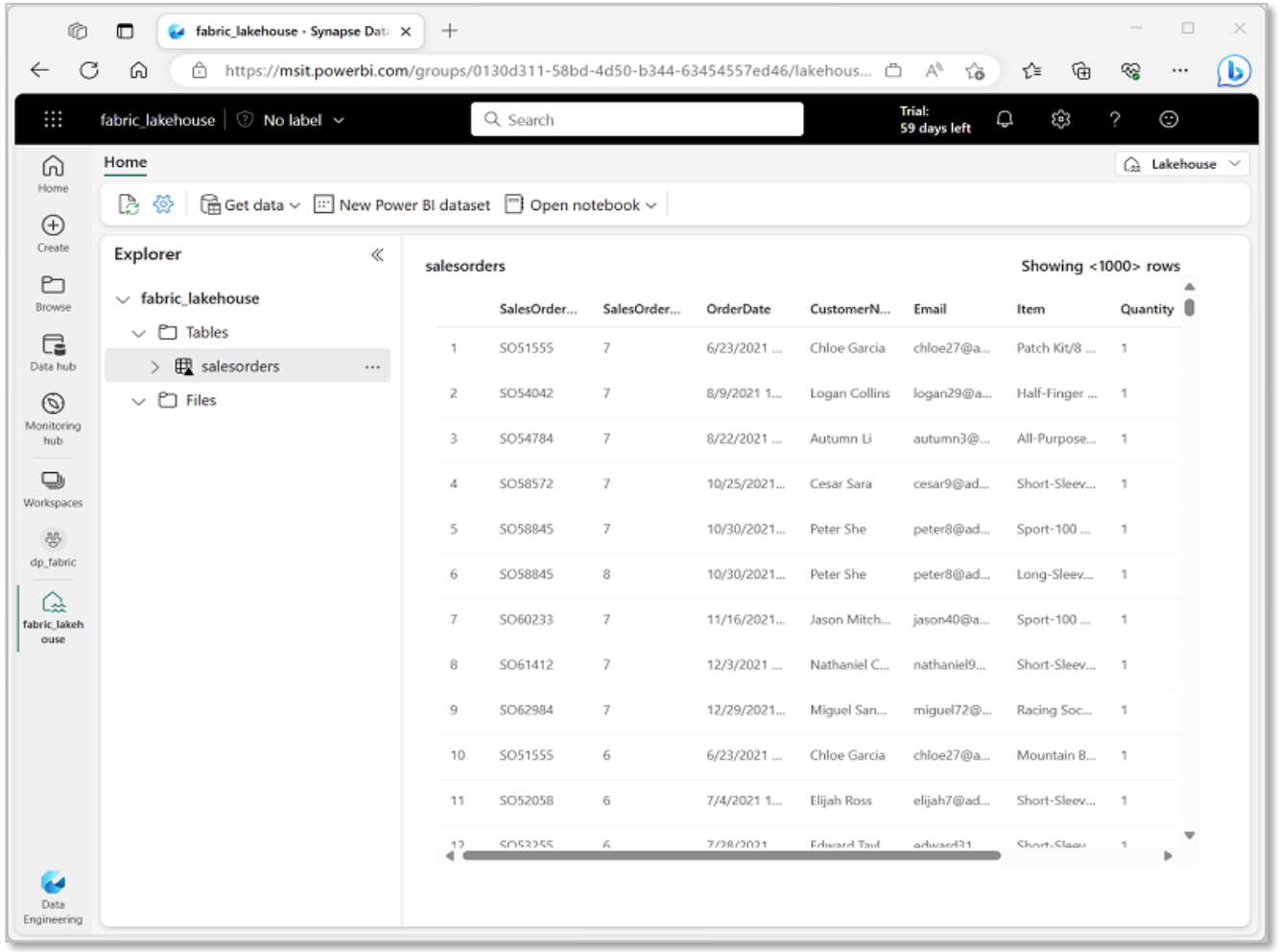
- Flexible and scalable solution that enables organizations to handle large data volumes of all types and sizes
- Built-in SQL endpoint unlocks Data Warehouse capabilities on top of your Lakehouse with no data movement
- Address the challenges of traditional Data Lakes by adding a **Delta Lake storage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



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



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

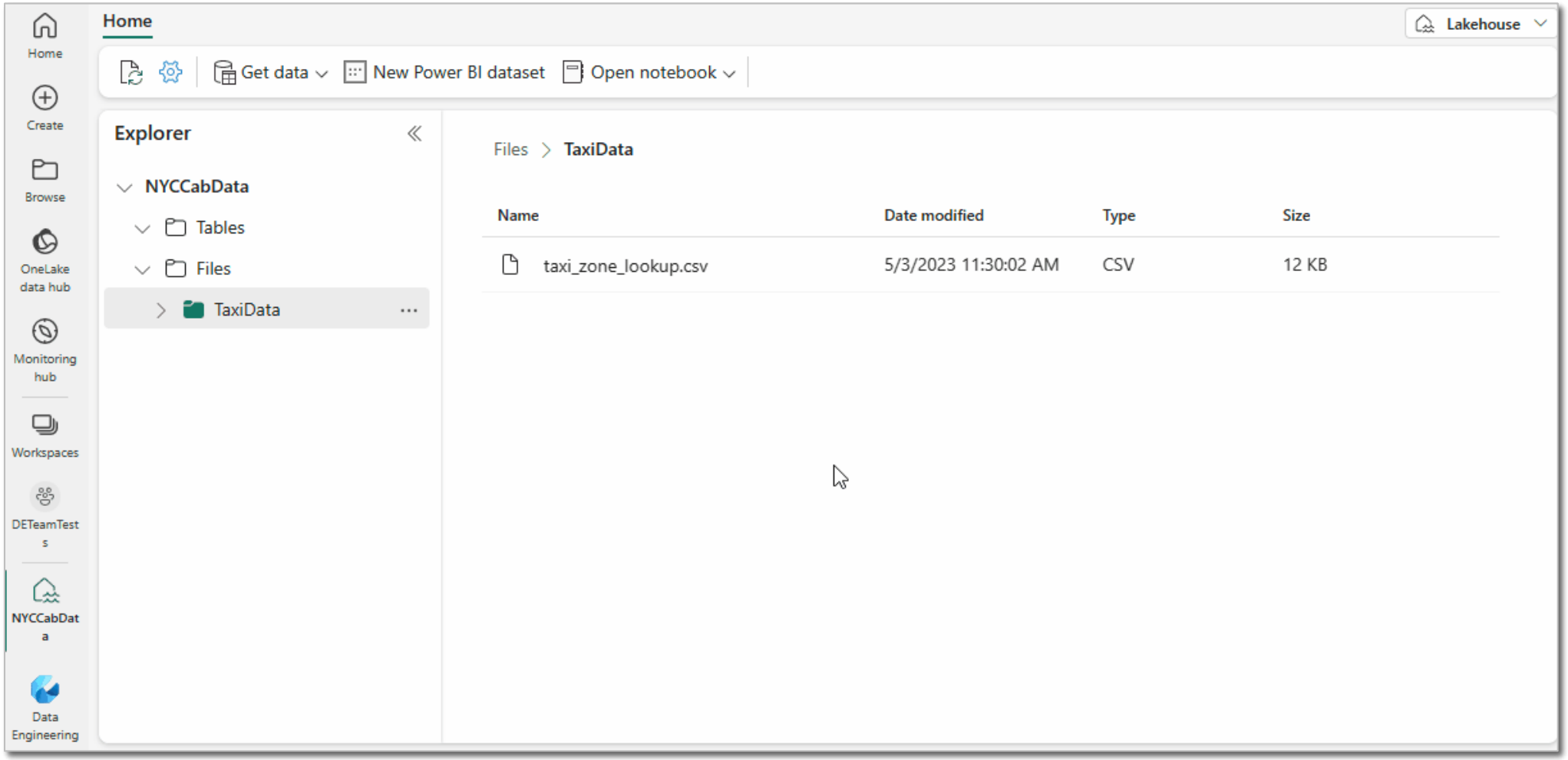
Apache Spark job definitions

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

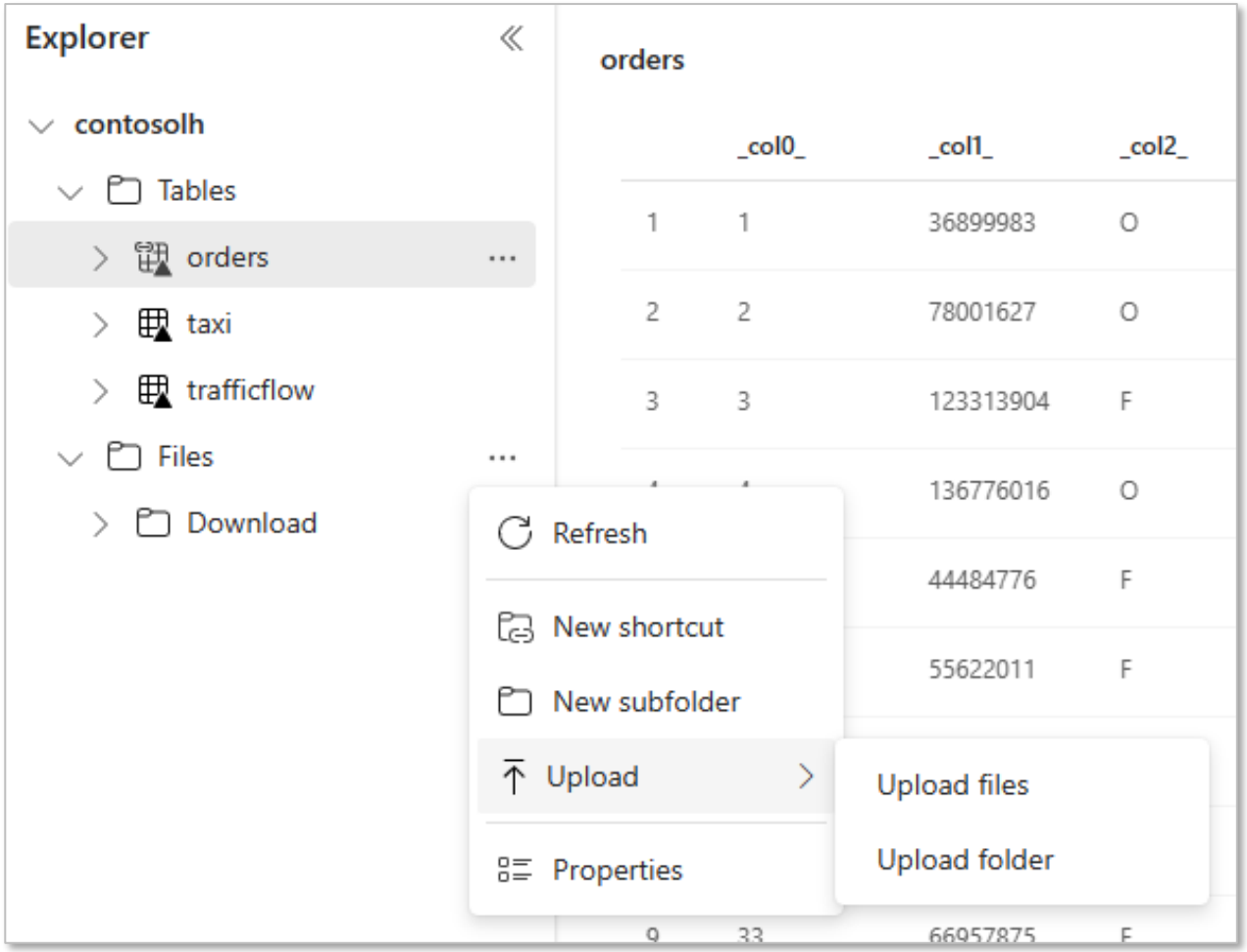
Automatic table discovery and registration



Data ingestion into Lakehouse

Easily ingest data into the Lakehouse through a variety of methods

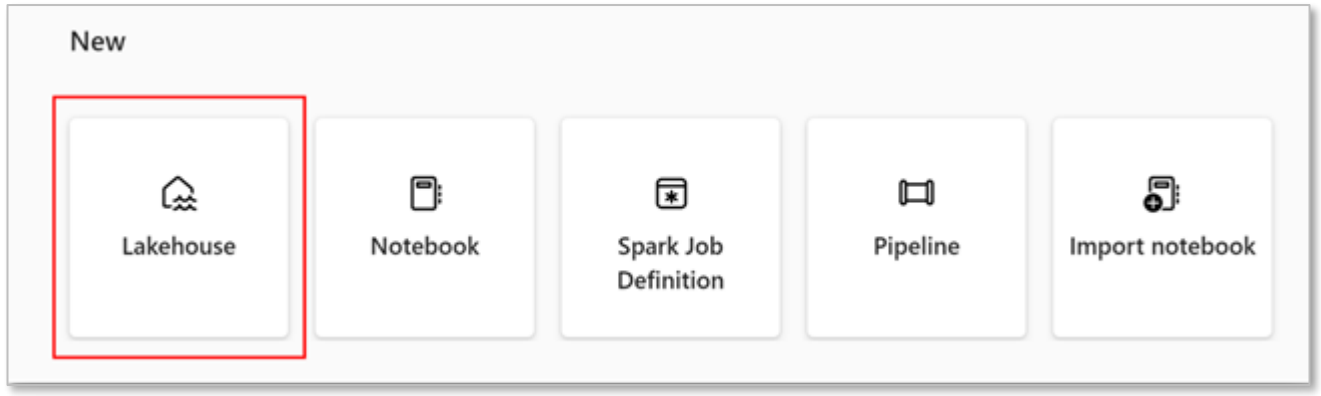
- Get files from your computer using direct upload
- Connect to 120+ data sources and apply multiple transformations using Dataflows or copy petabyte-sized lakes using the copy activity in Data pipelines
- Use Spark code to connect to data sources using available Spark libraries
- Leverage shortcuts 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



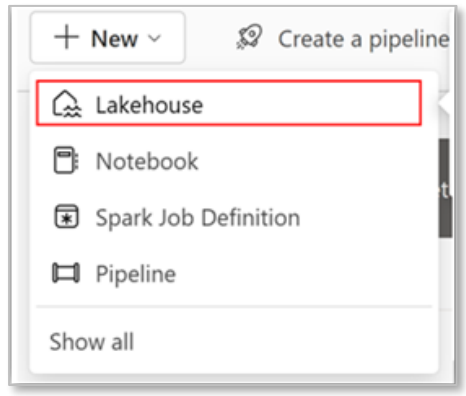
Ways to create a Lakehouse in Microsoft Fabric

Ways to create a Lakehouse

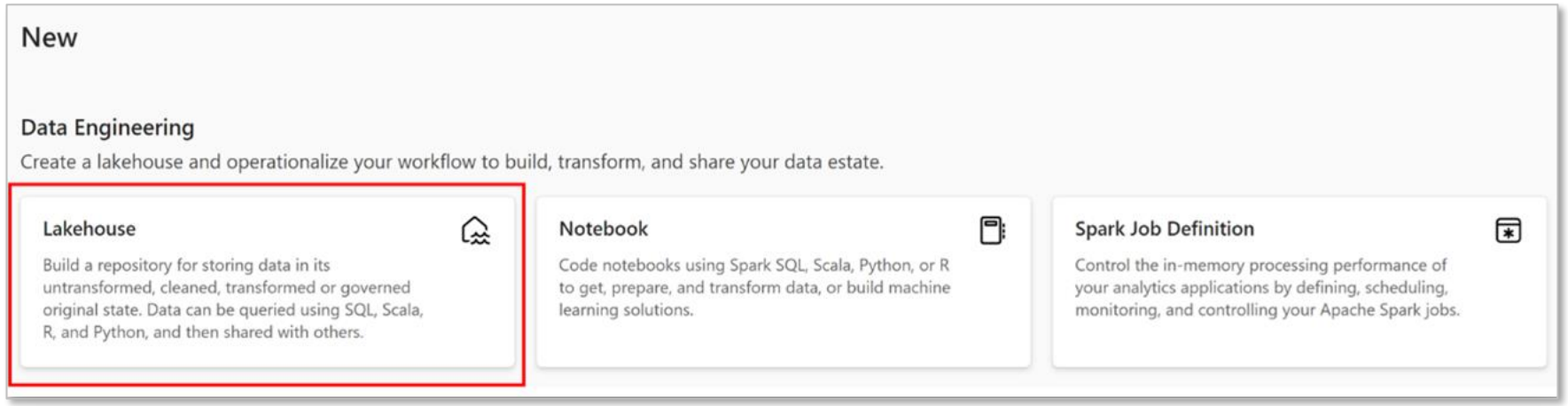
Using Data Engineering homepage



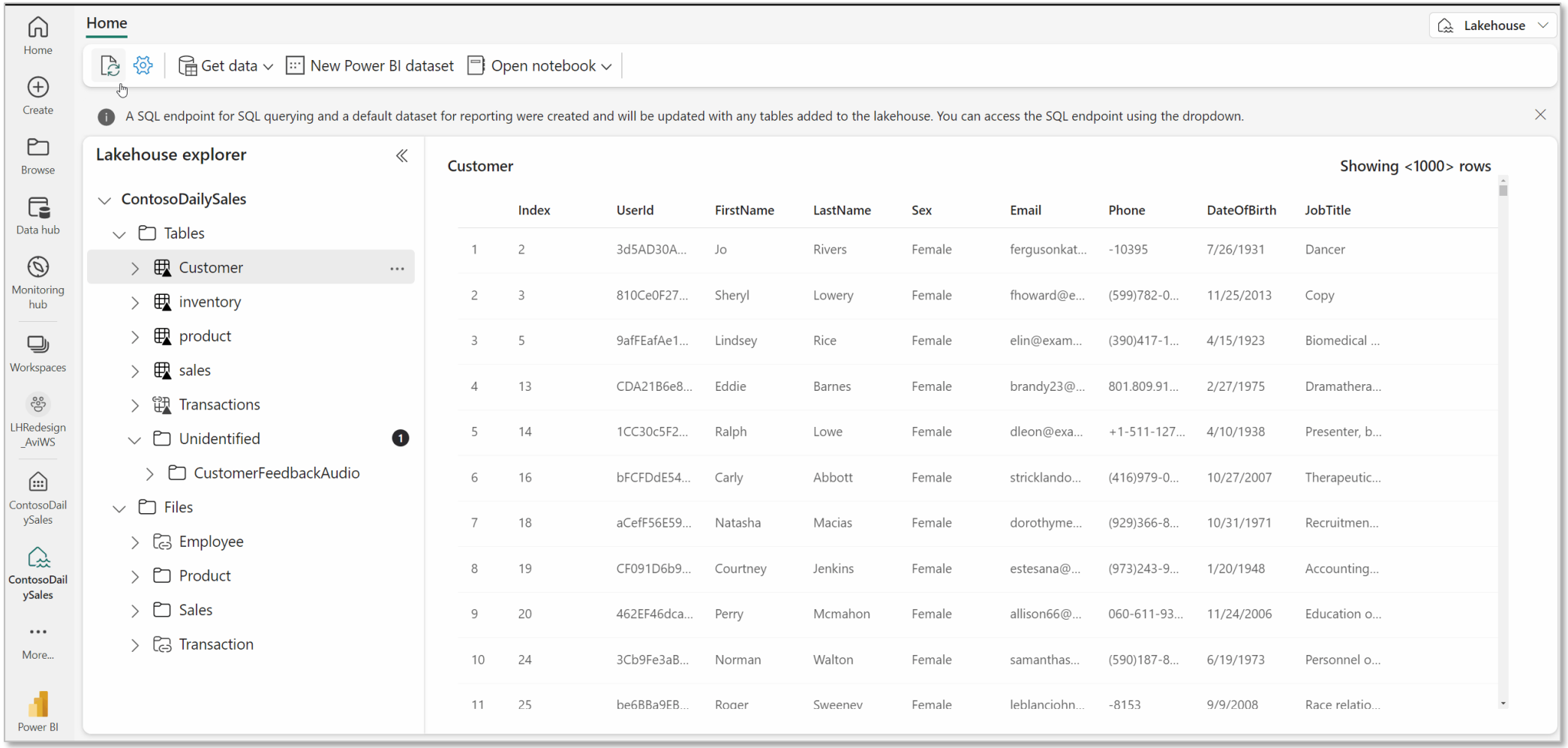
Using Workspace view



Creating Hub



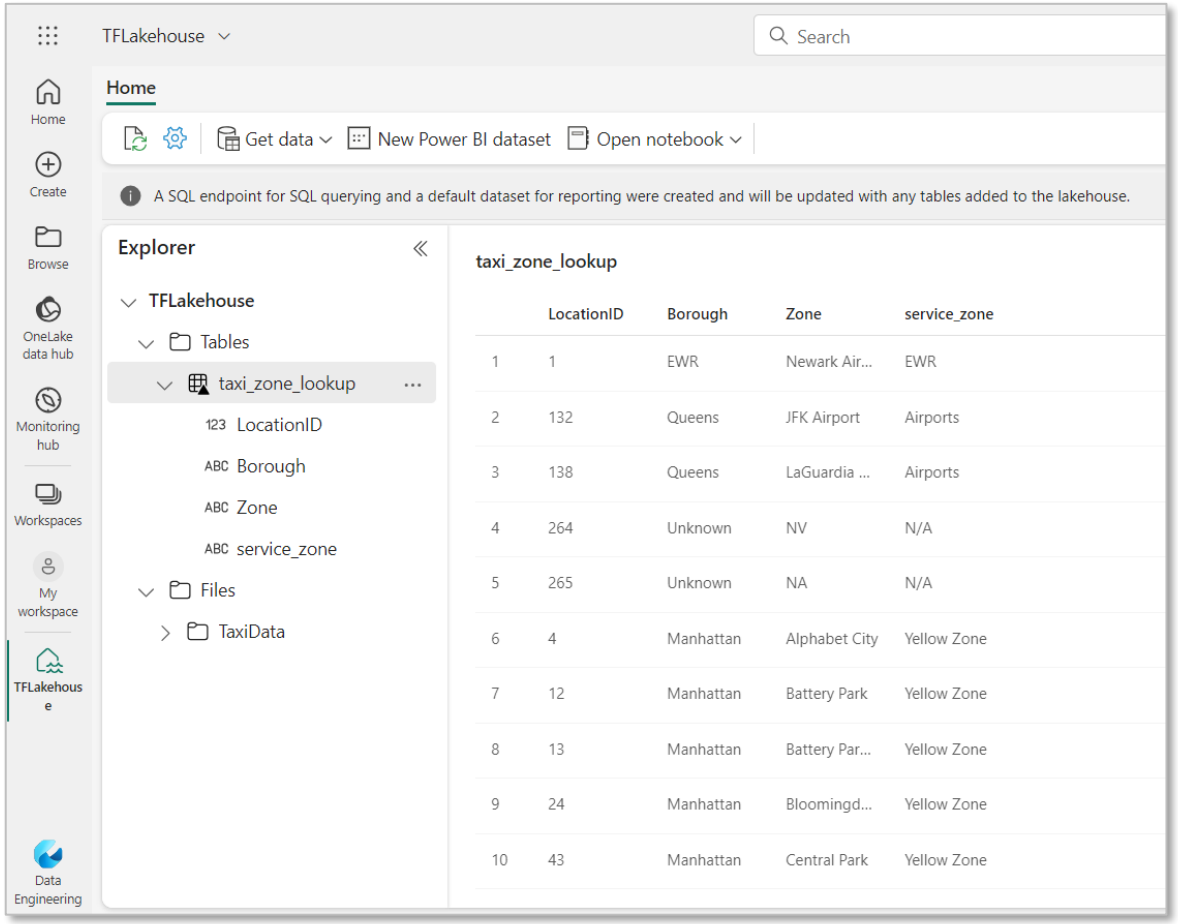
Navigating the Fabric Lakehouse explorer



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



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

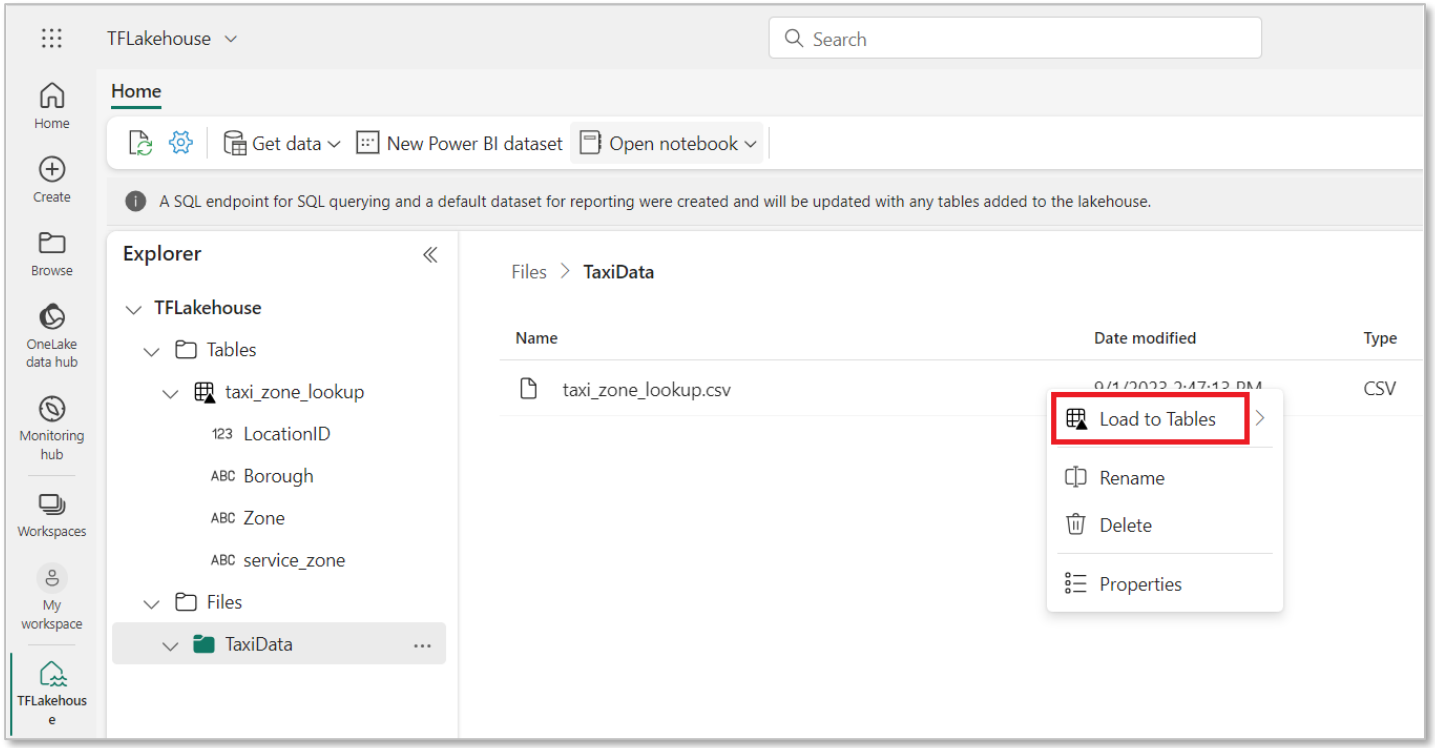


Table and column name validation and rules

Table names can only contain alphanumeric characters and underscores

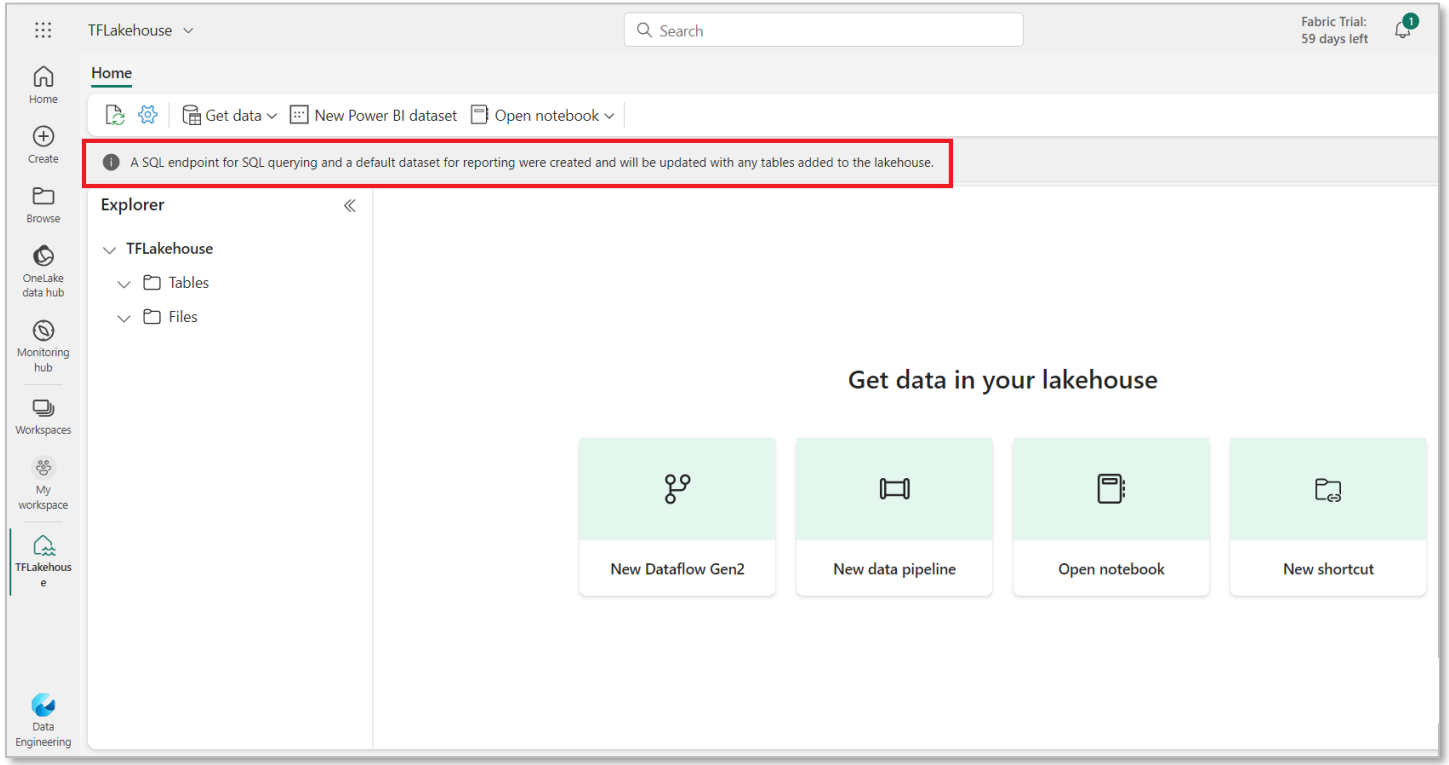
Text files without column headers are replaced with standard col# notation as the table column names

Column names are validated during the load action

If no proper column name is achieved during validation, the load action fails

Lakehouse SQL endpoint

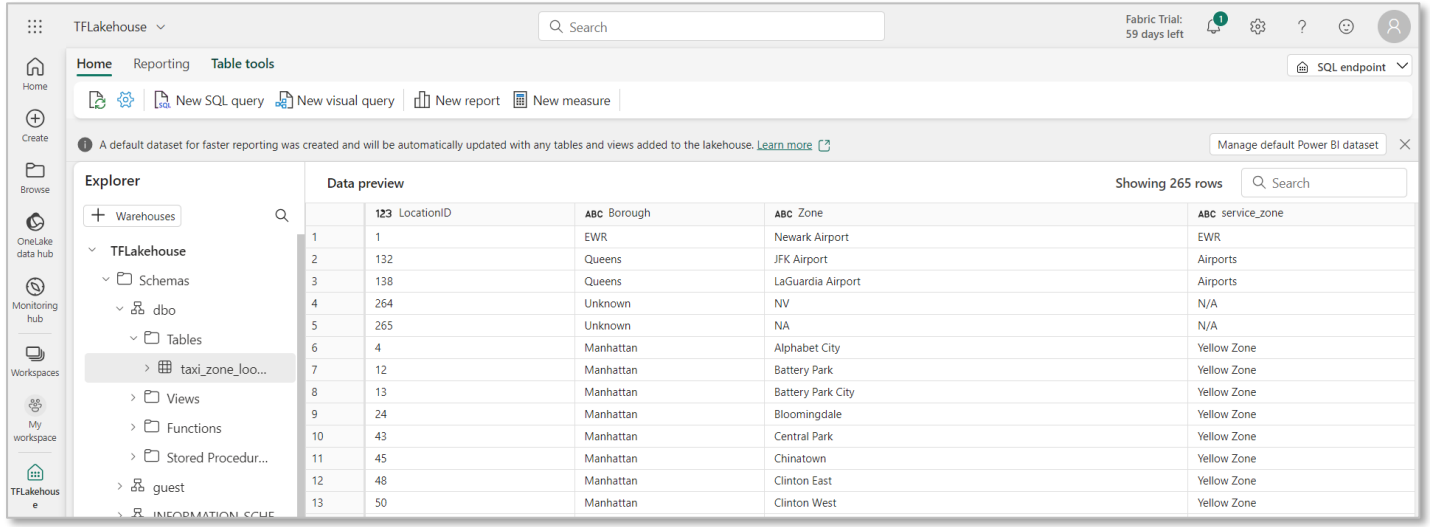
The Lakehouse creates a serving layer by automatically generating a SQL endpoint and a default dataset during creation



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



	LocationID	Borough	Zone	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

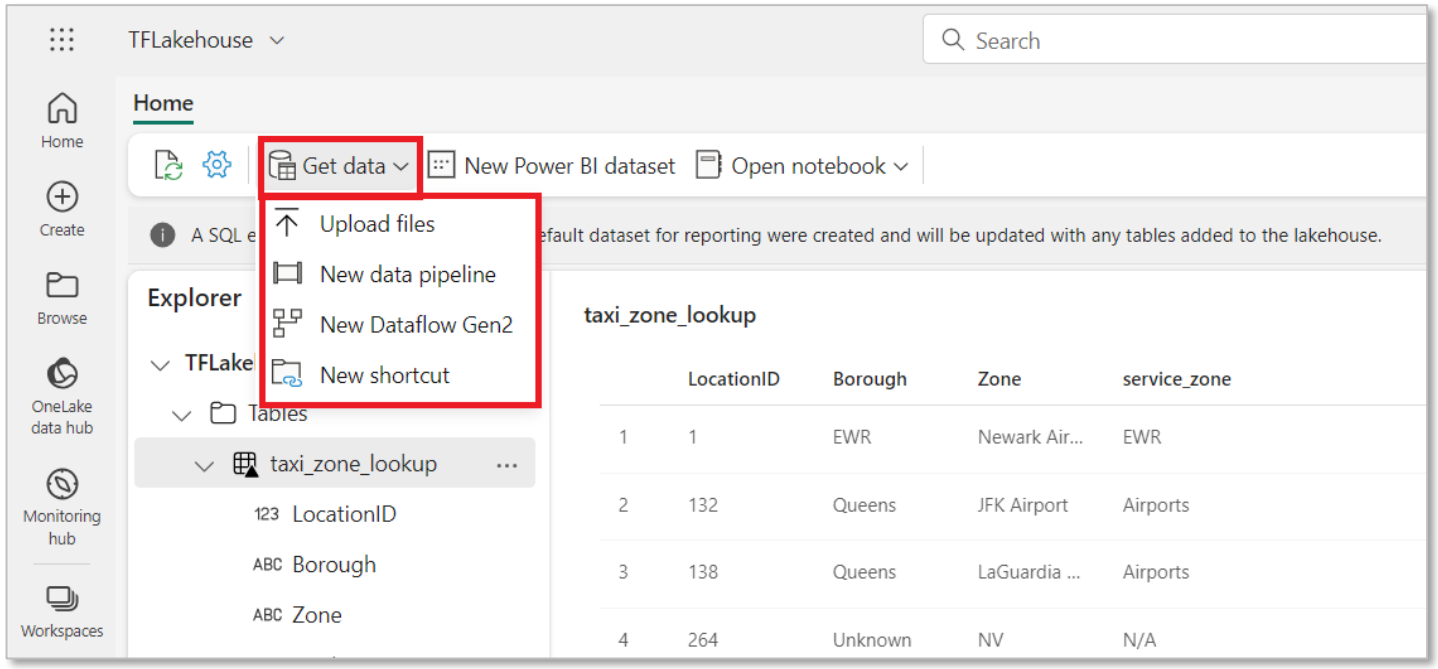
DEMO

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



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

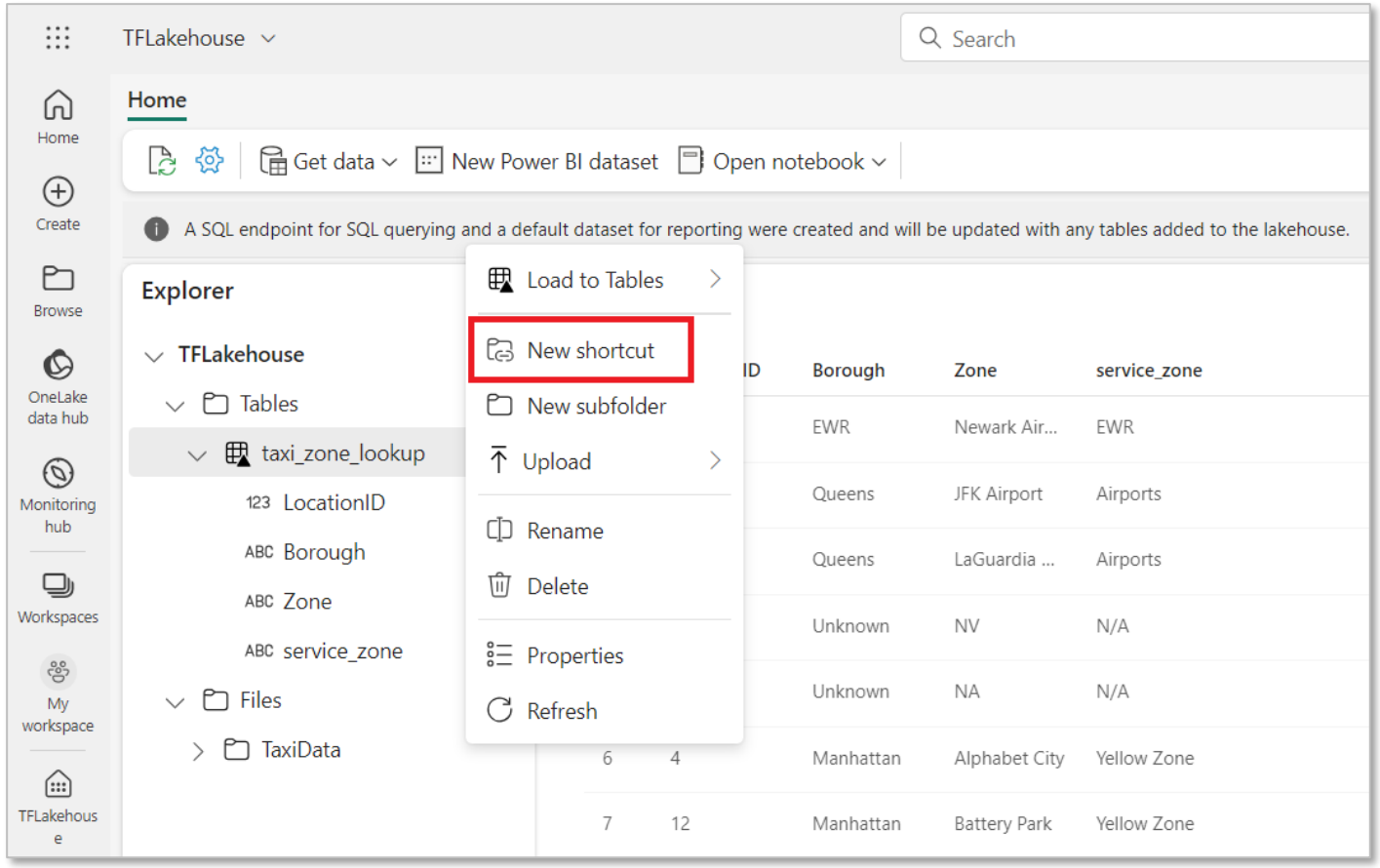
Referencing data without copying it from other internal Lakehouses or external sources

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

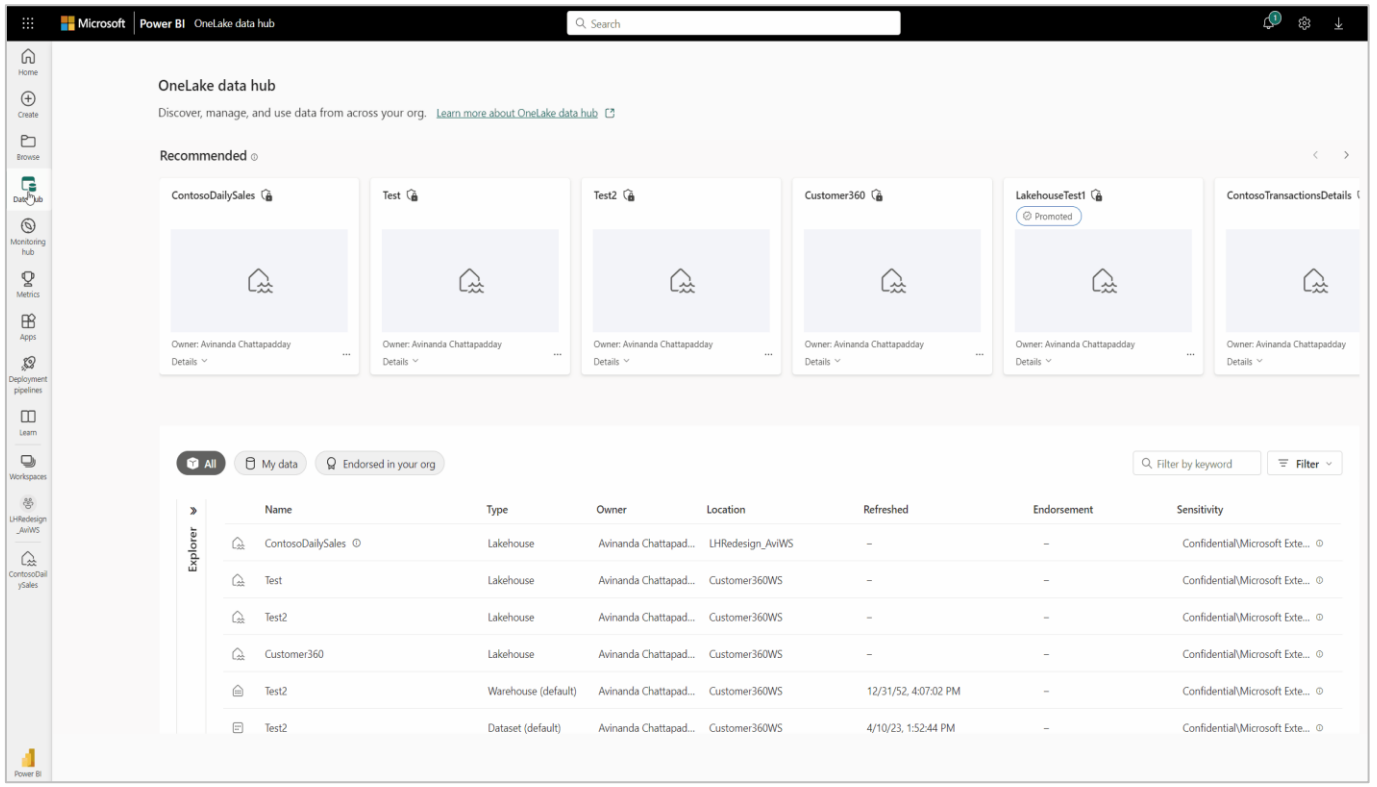
- ADLS Gen2
- AWS S3



Lakehouse sharing

Share your Lakehouse as a data product with consumers

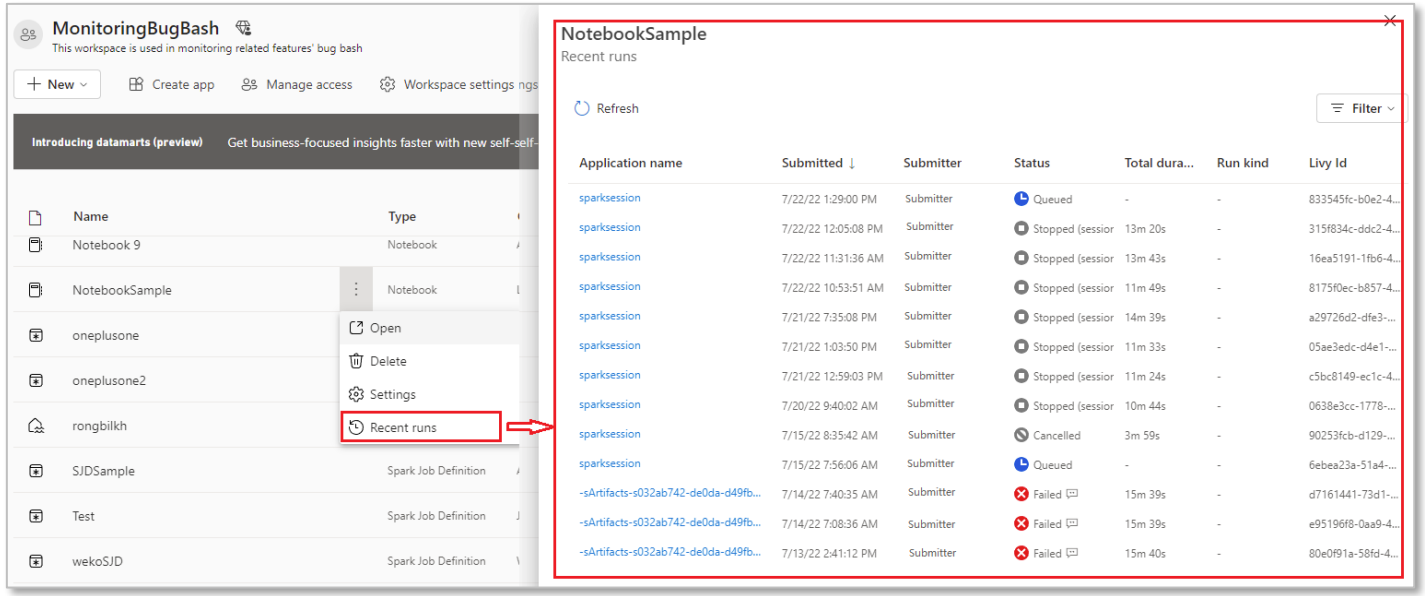
- Provide users access to a Lakehouse without adding them to your workspace
- Grant access to Lakehouse data through Spark, SQL Endpoint, and default dataset for Power BI reporting
- Use SQL Security or customer permission to grant access through SQL Endpoint
- Discover Lakehouses you have access to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



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



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

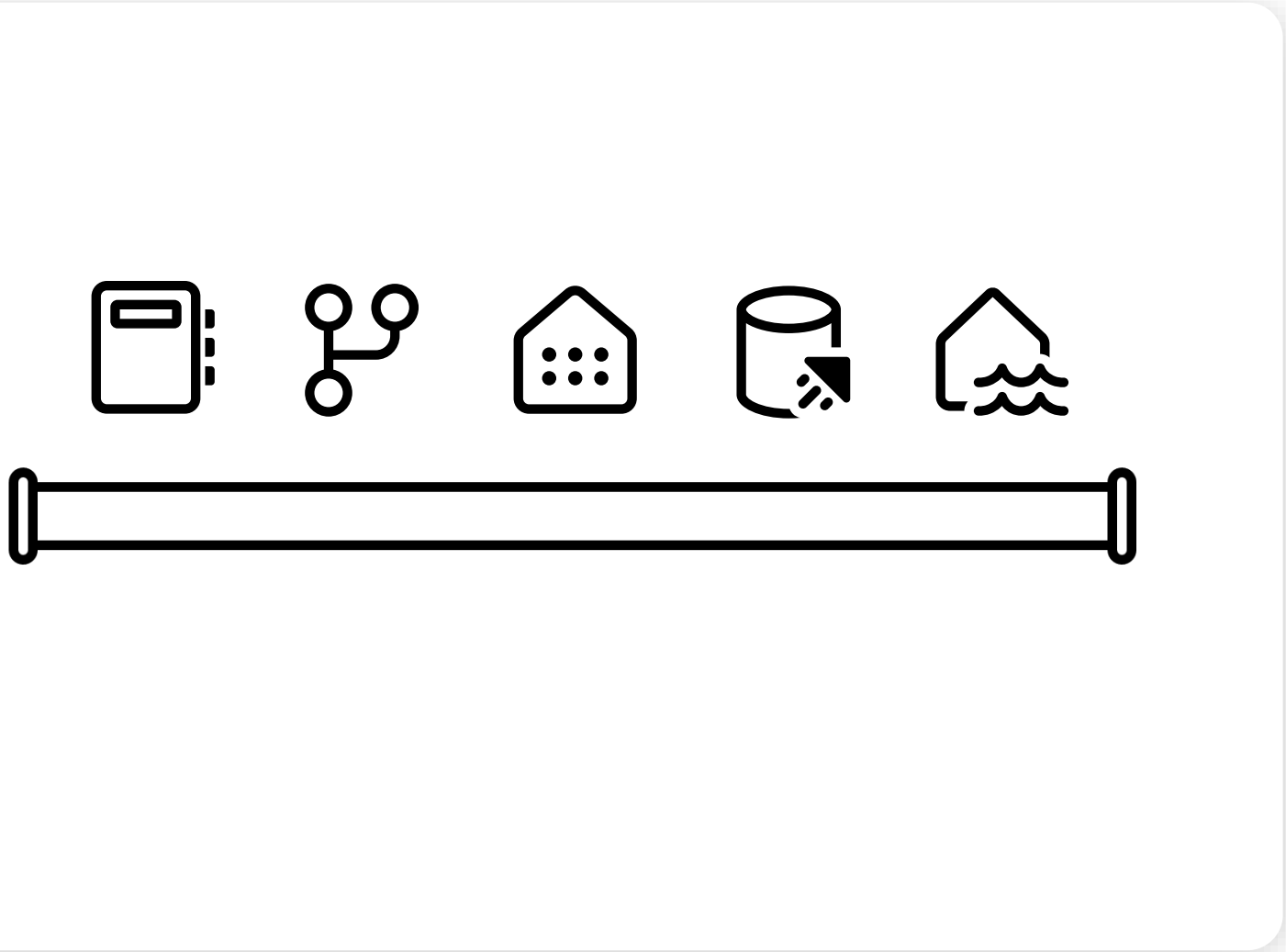
Collect, process, and transform data from its raw form to a format that you can use for analysis and decision-making

Working with Data Pipelines in Microsoft Fabric

Data pipelines

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

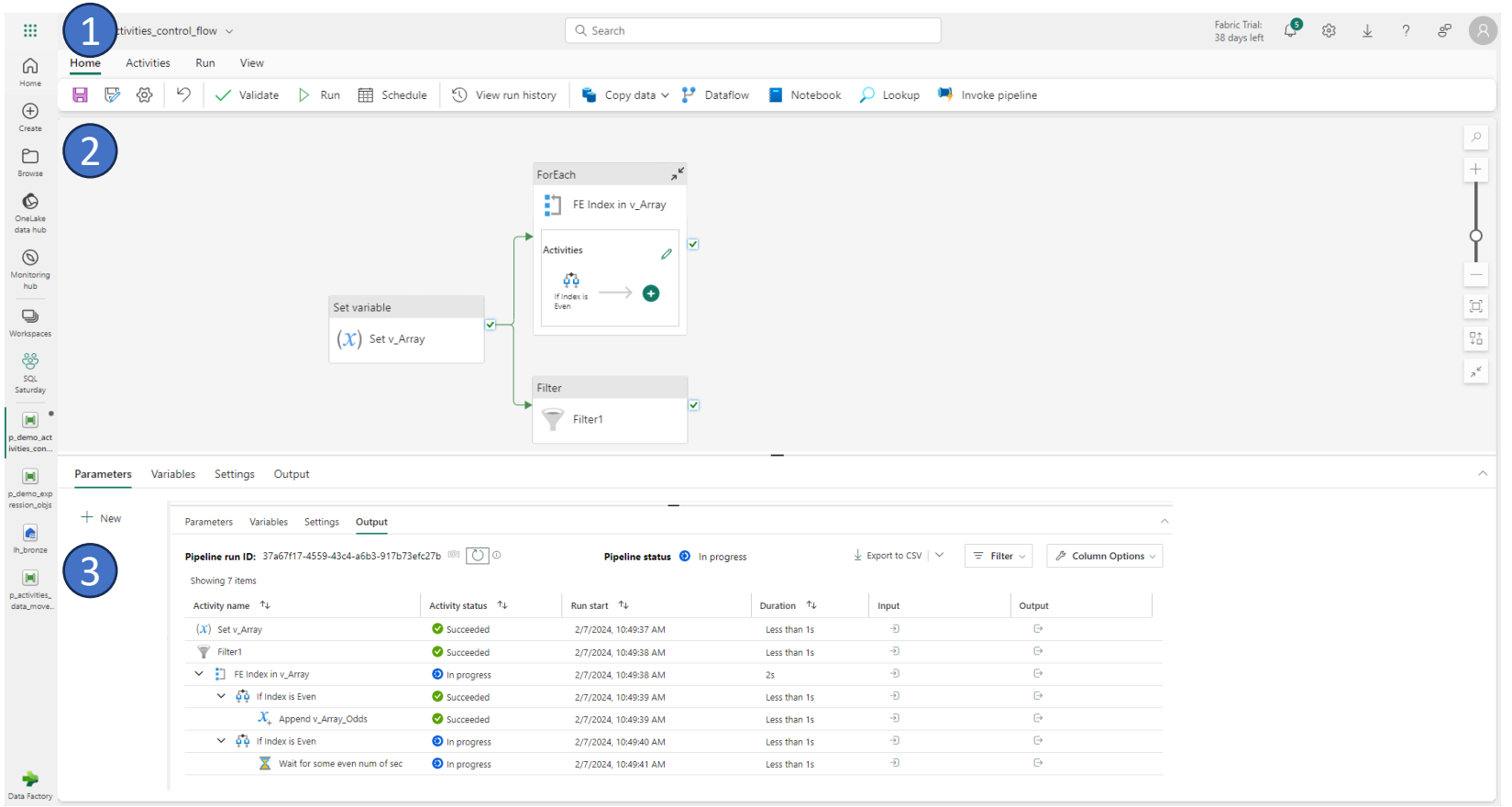
- Evolution of **Azure Data Factory** pipelines
- Rich library of **activities**
- Fast copy
- Jumpstart with **Copy Assistant**



Exploring Data pipelines

Data Pipeline

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



Copy activity

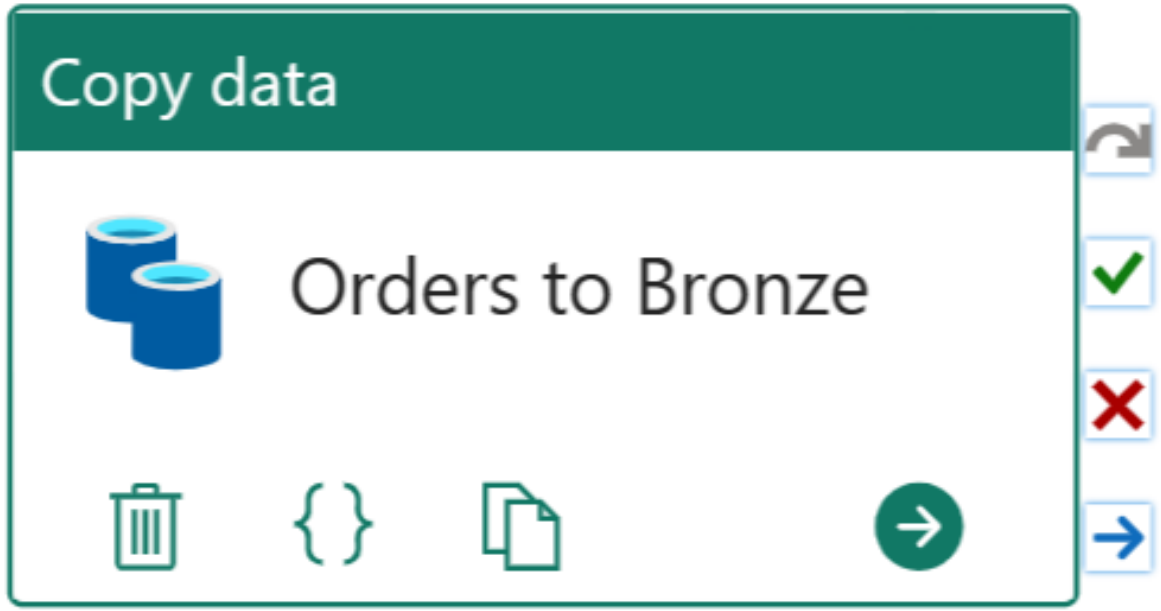
The Copy Activity is the **core** of Data pipelines.

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 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.

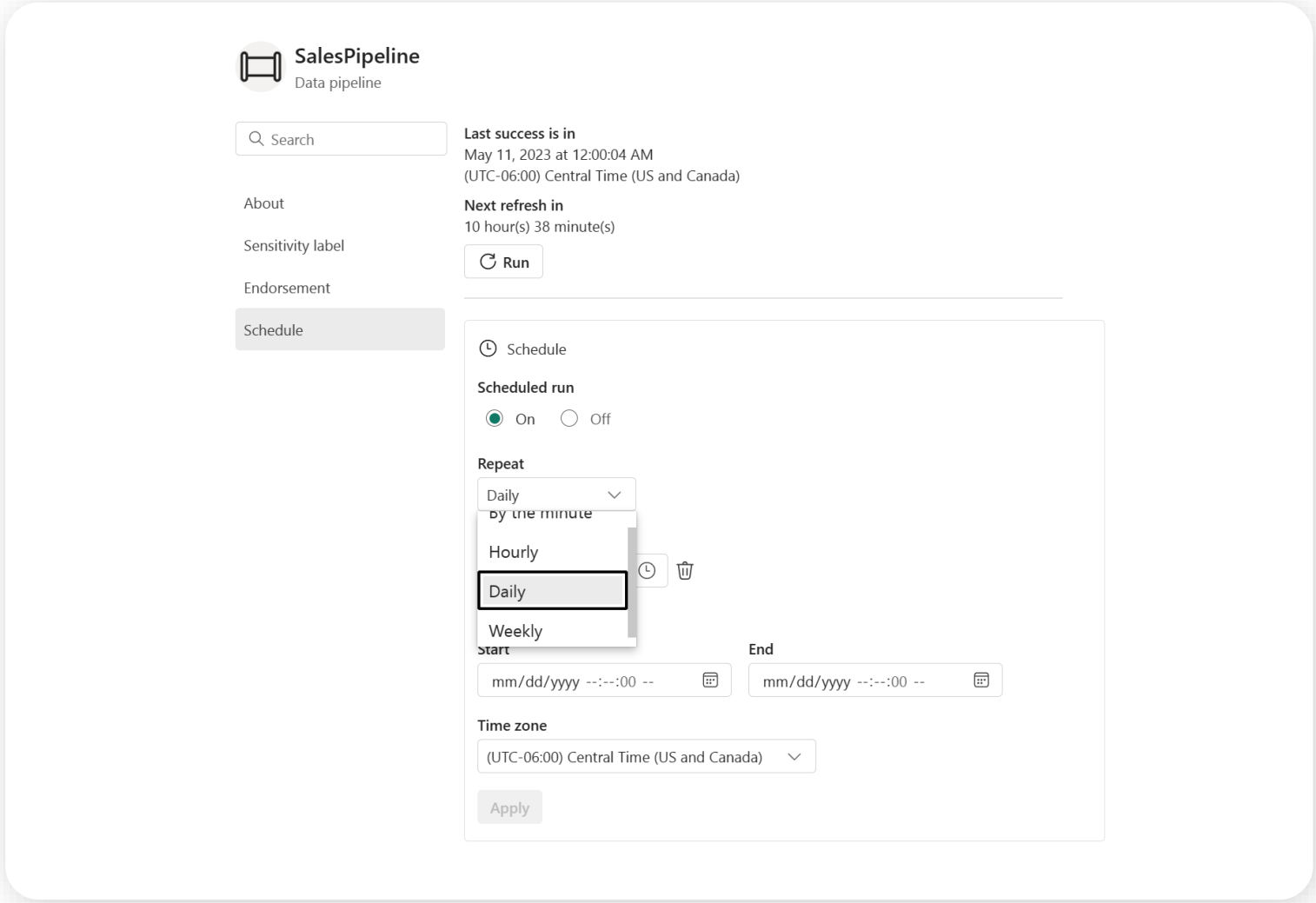
And so much more…

Control Flow	Nesting	Parameterization	Expression Language	Repeatable
• Conditional paths • Loops	• Invoking data pipelines from other data pipelines • Bi-directional communication via output variables	• Parameters of varying data types	• Almost every field can be made dynamic! • Extensive built-in functions allow for massive customizations (data driven designs)	• Schedulable

Data Pipeline: Schedule

Seamlessly schedule events and set a desired time value.

- Built in monitoring
- Repeat: By the minute, Hourly, Daily, Weekly



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

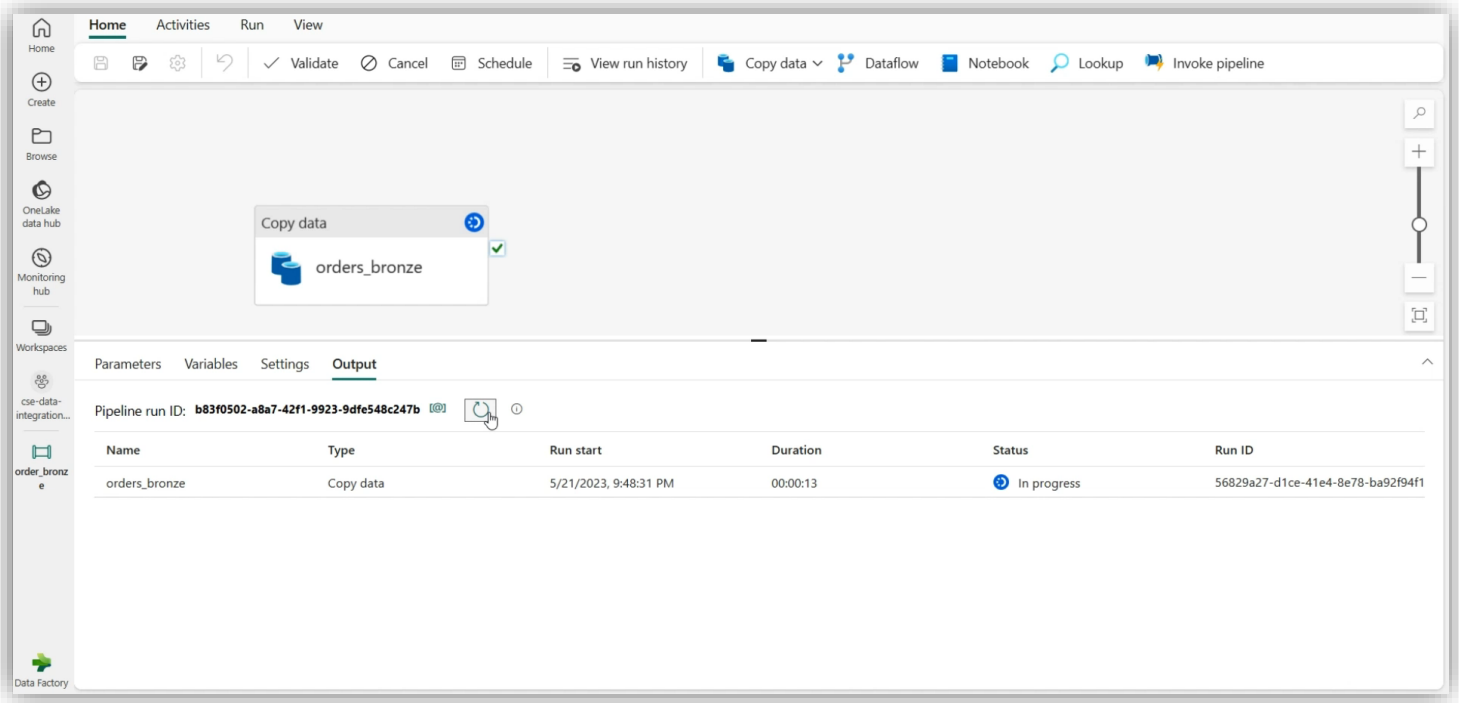
Migrate from ADF/Synapse Pipelines to Fabric Data pipelines

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

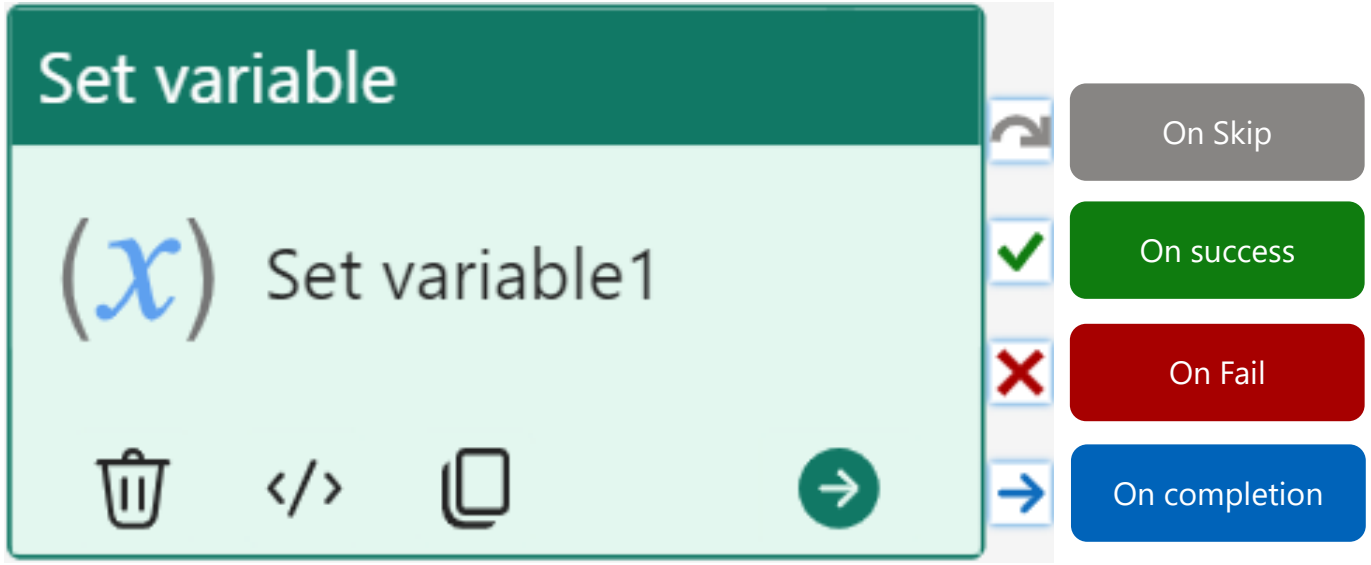
- Understand how to find errors
- Determine if it is a Pipeline error or an error from the Source/Destination
- Set a Retry attempt



Error handling

Create conditional paths for orchestration

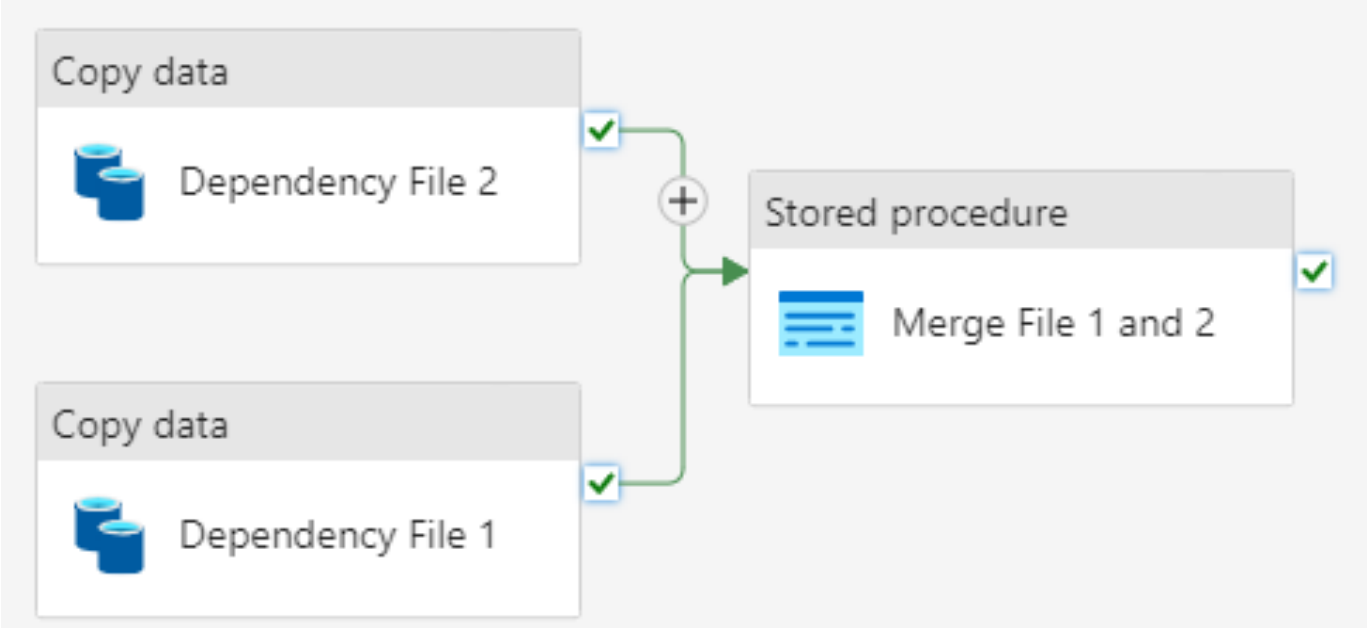
- Blocking dependencies
- Non-blocking dependencies
- Error handling



Error handling

Create conditional paths for orchestration

- **Blocking dependencies**
- Non-blocking dependencies
- Error handling

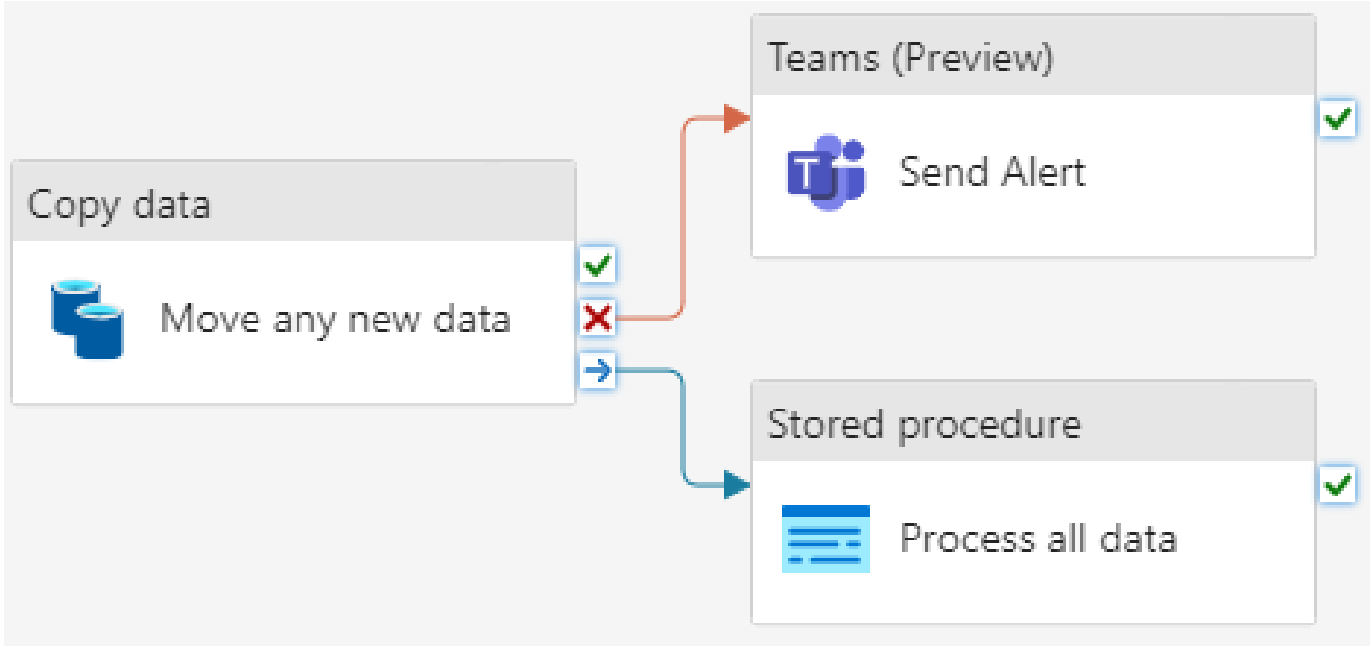


Both activities must be successful before proceeding.

Error handling

Create conditional paths for orchestration

- Blocking dependencies
- **Non-blocking dependencies**
- Error handling

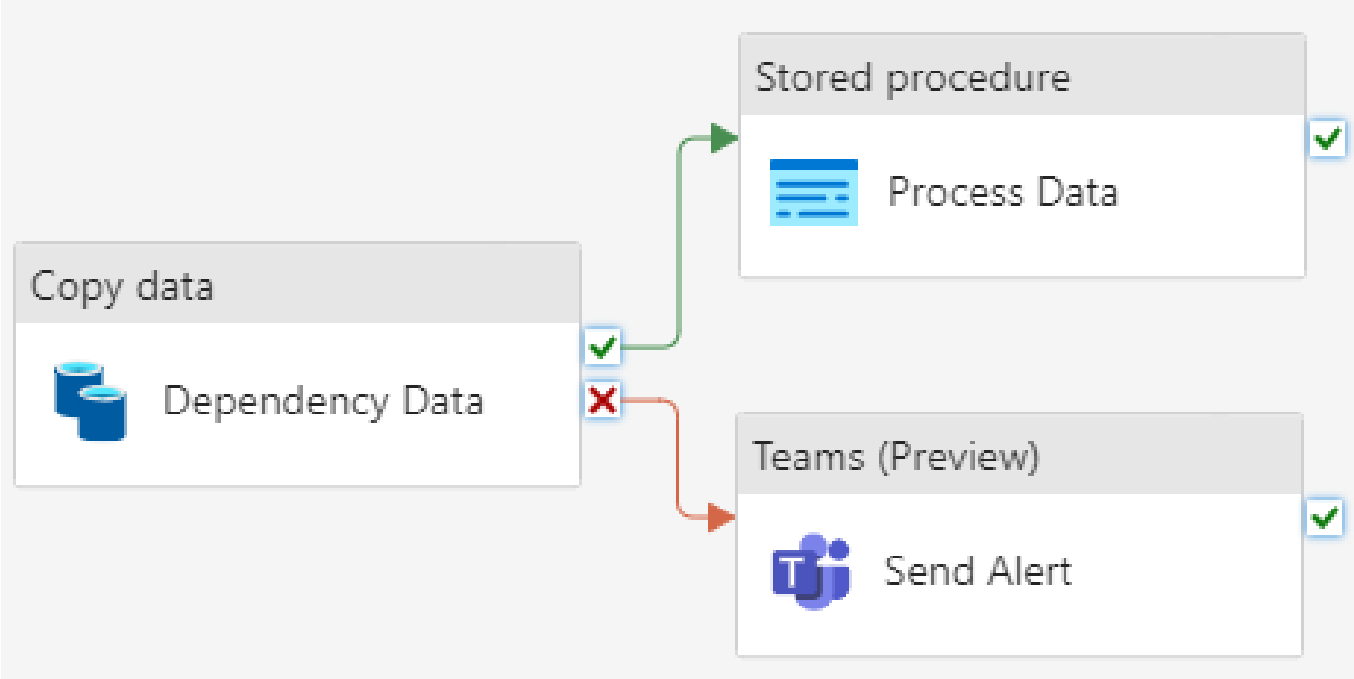


Regardless of the success or failure of an activity, the orchestrator can proceed to the next activity.

Error handling

Create conditional paths for orchestration

- Blocking dependencies
- Non-blocking dependencies
- **Error handling**



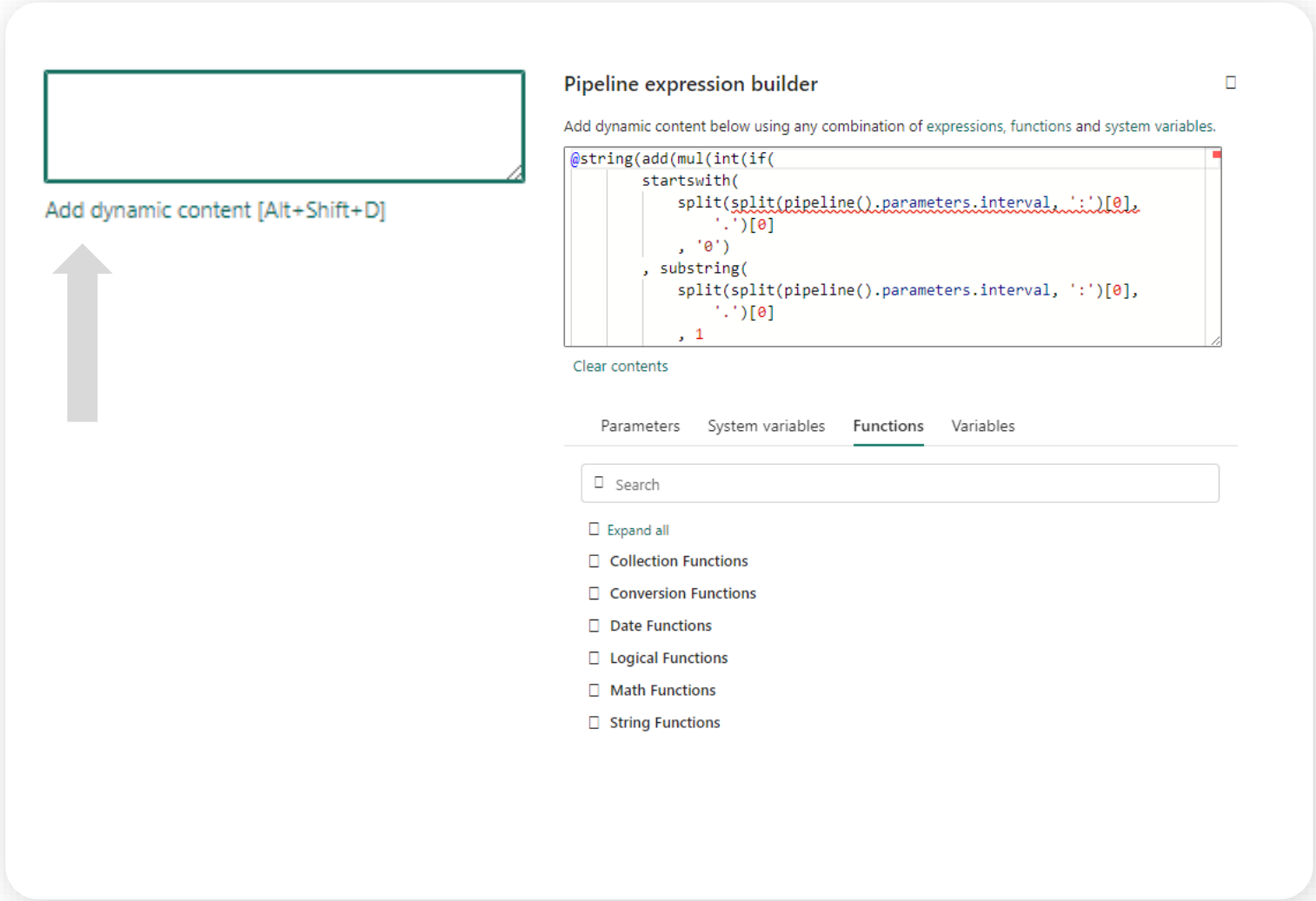
Creating a conditional path for both success and failure events.

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



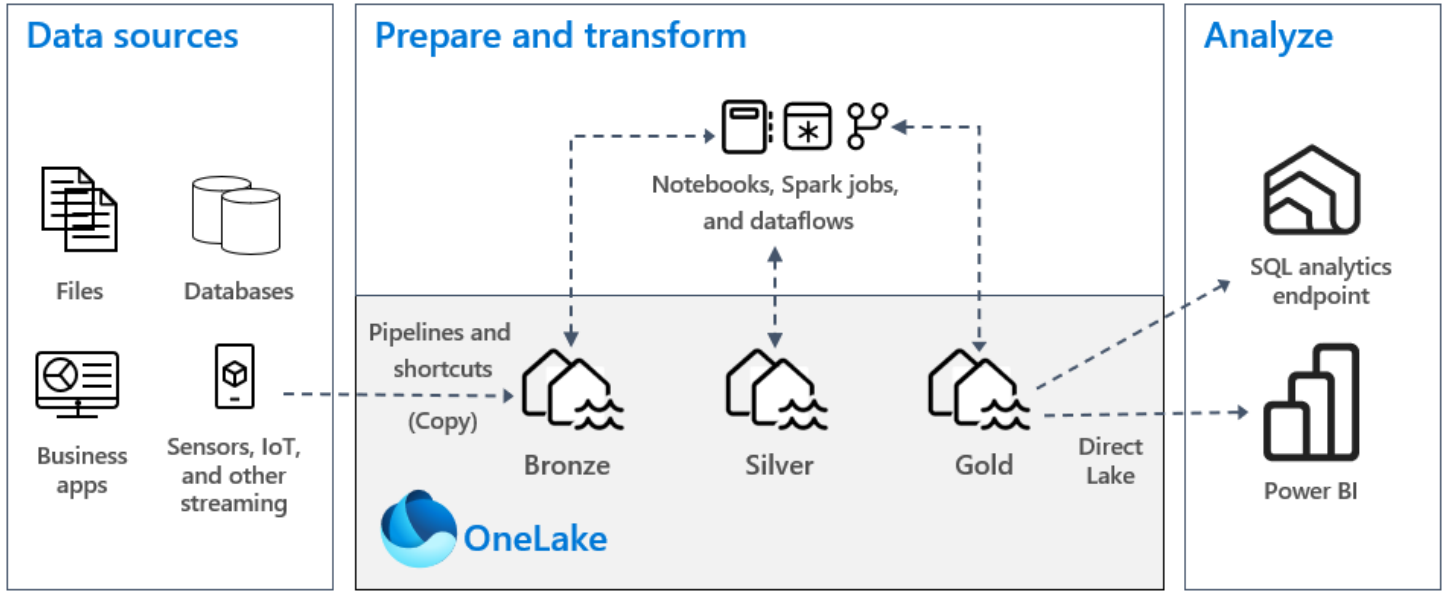
Implementing a Medallion Lakehouse Architecture

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).

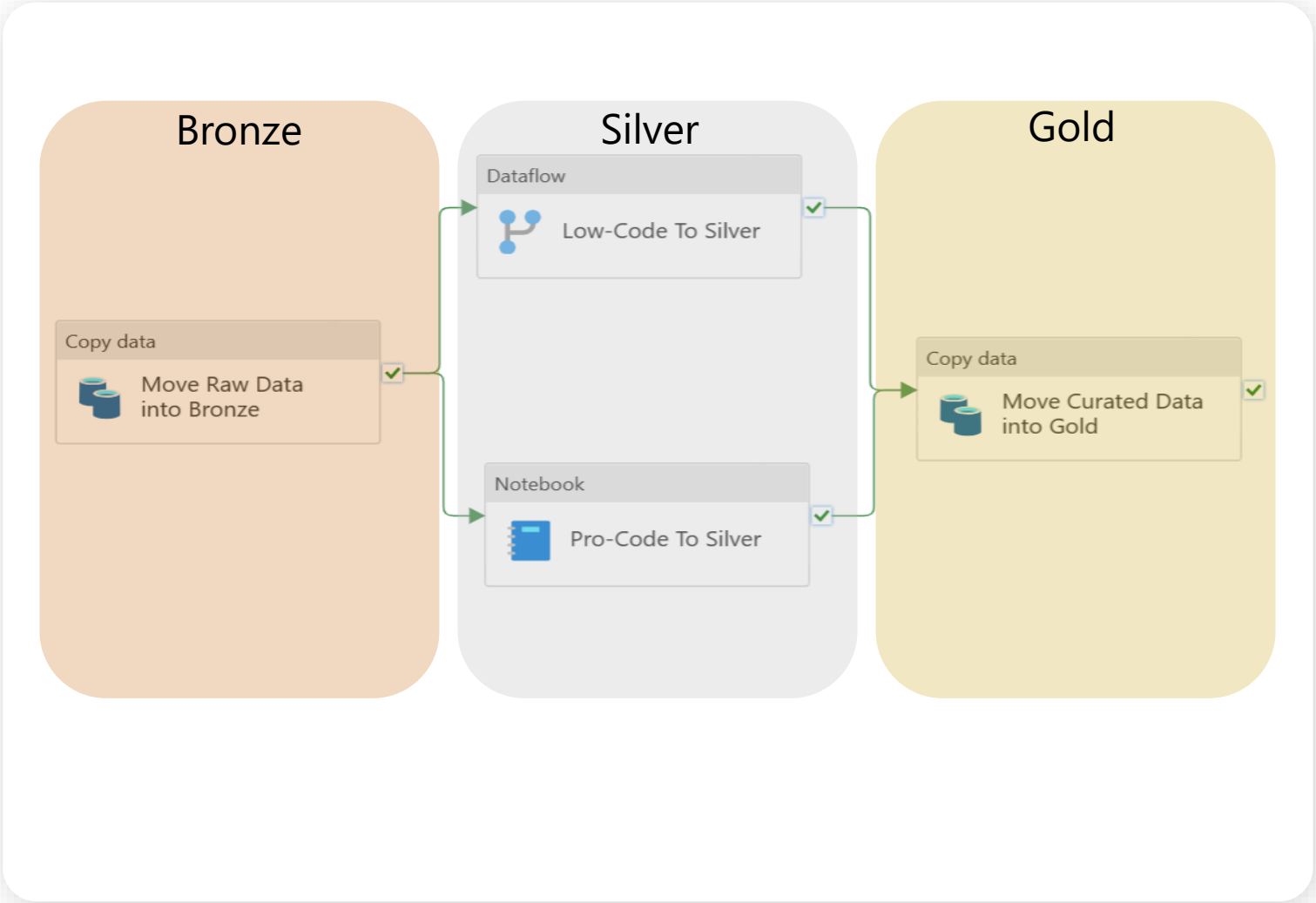


In a typical medallion architecture implementation in Fabric, the bronze zone stores the data in the same format as the data source. When the data source is a relational database, Delta tables are a good choice. The silver and gold zones contain Delta tables.

Medallion Lakehouse architecture

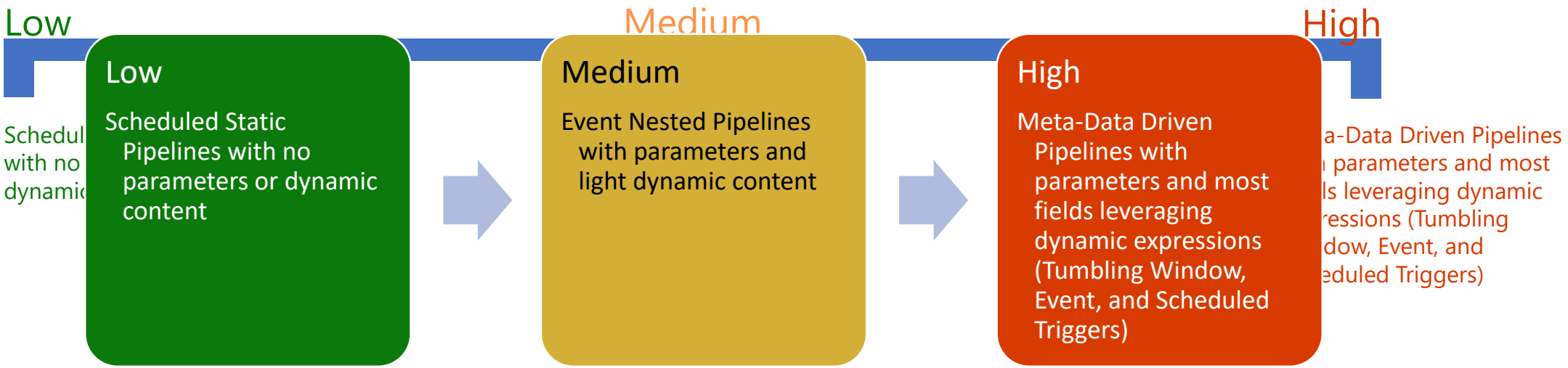
Medallion architecture consists of three distinct layers (or zones).

- **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



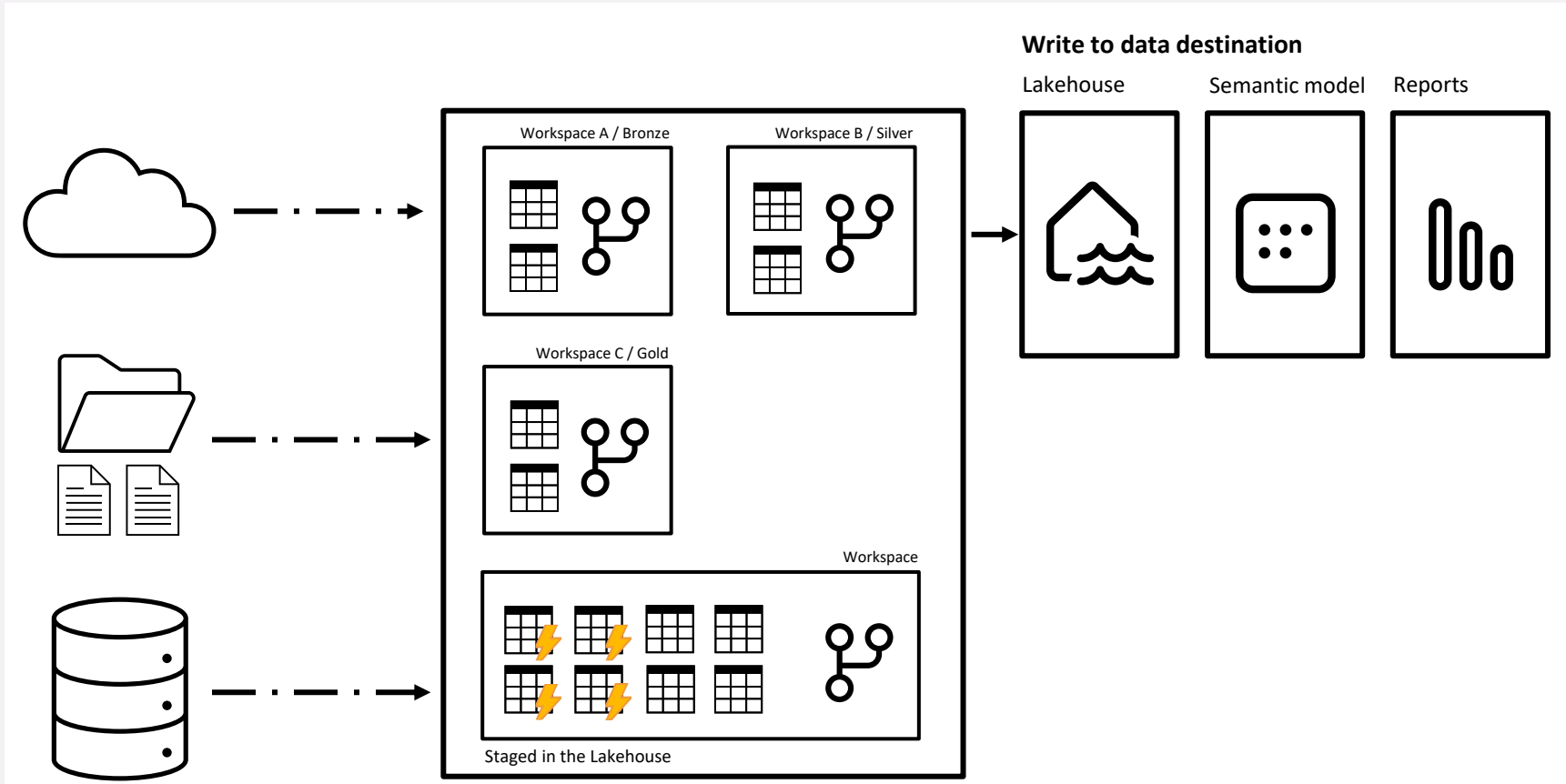
Ingesting and Transforming Data with Dataflows Gen2

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

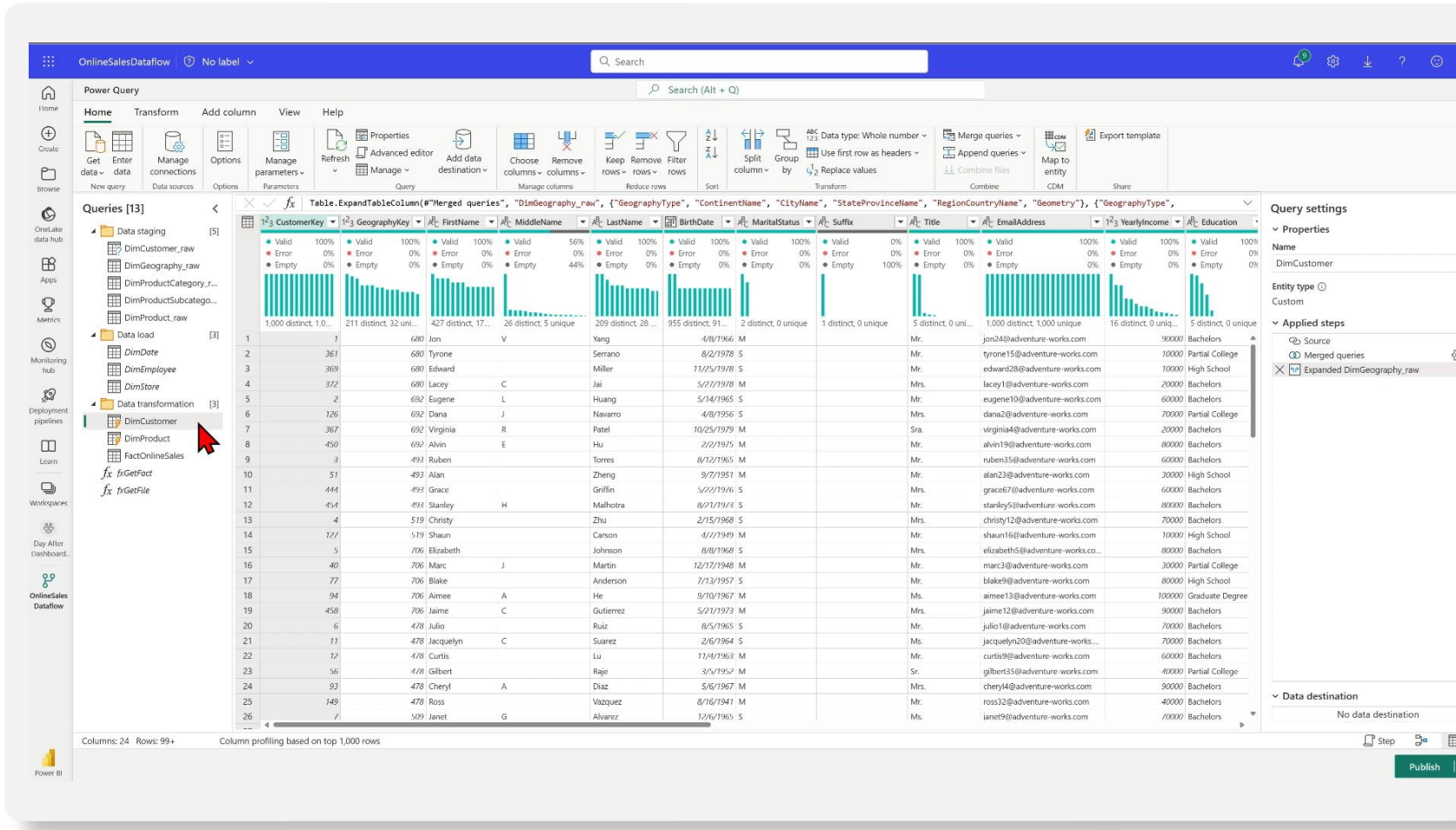
Next generation of **Data preparation**

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

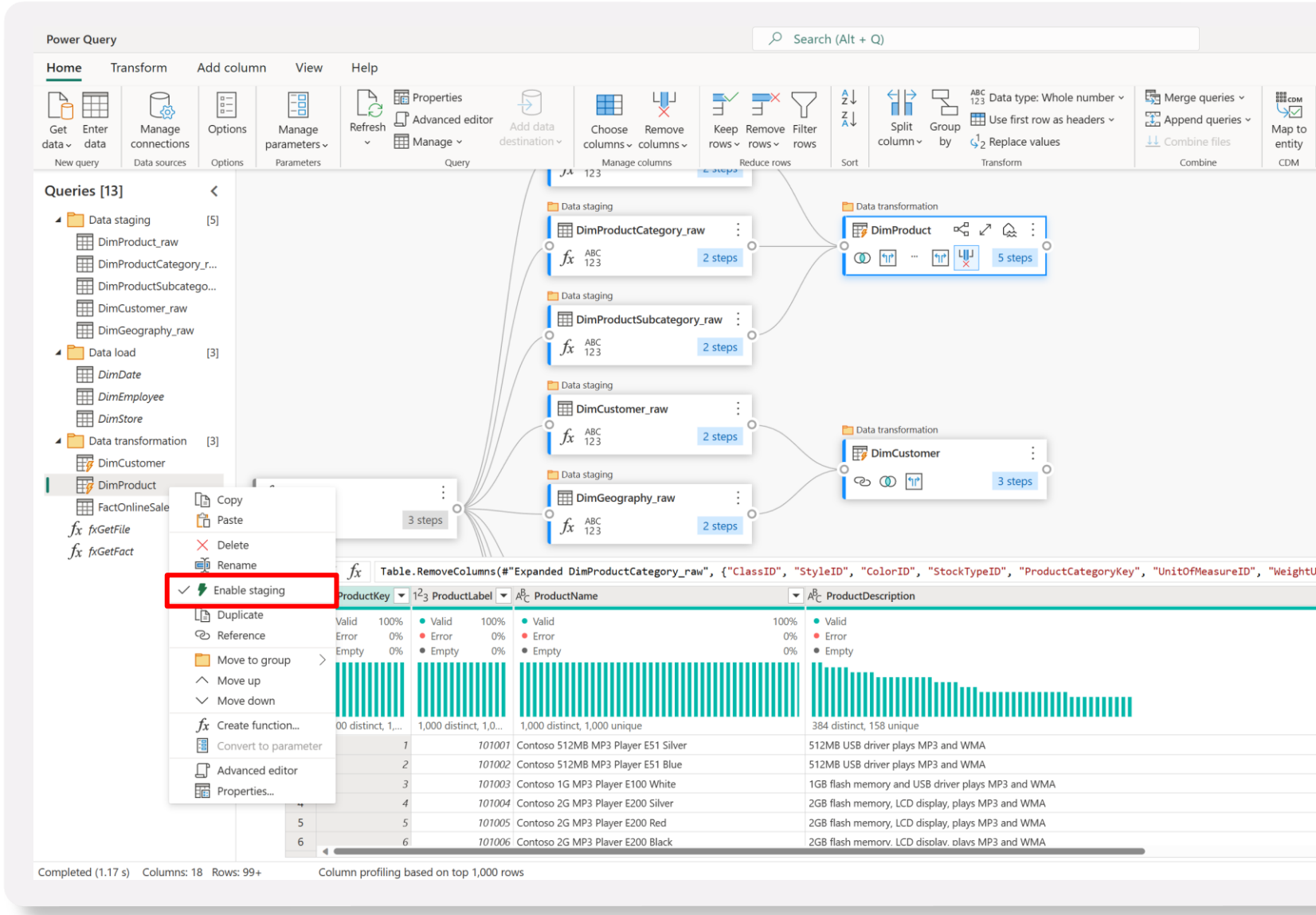


Dataflow Gen2

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

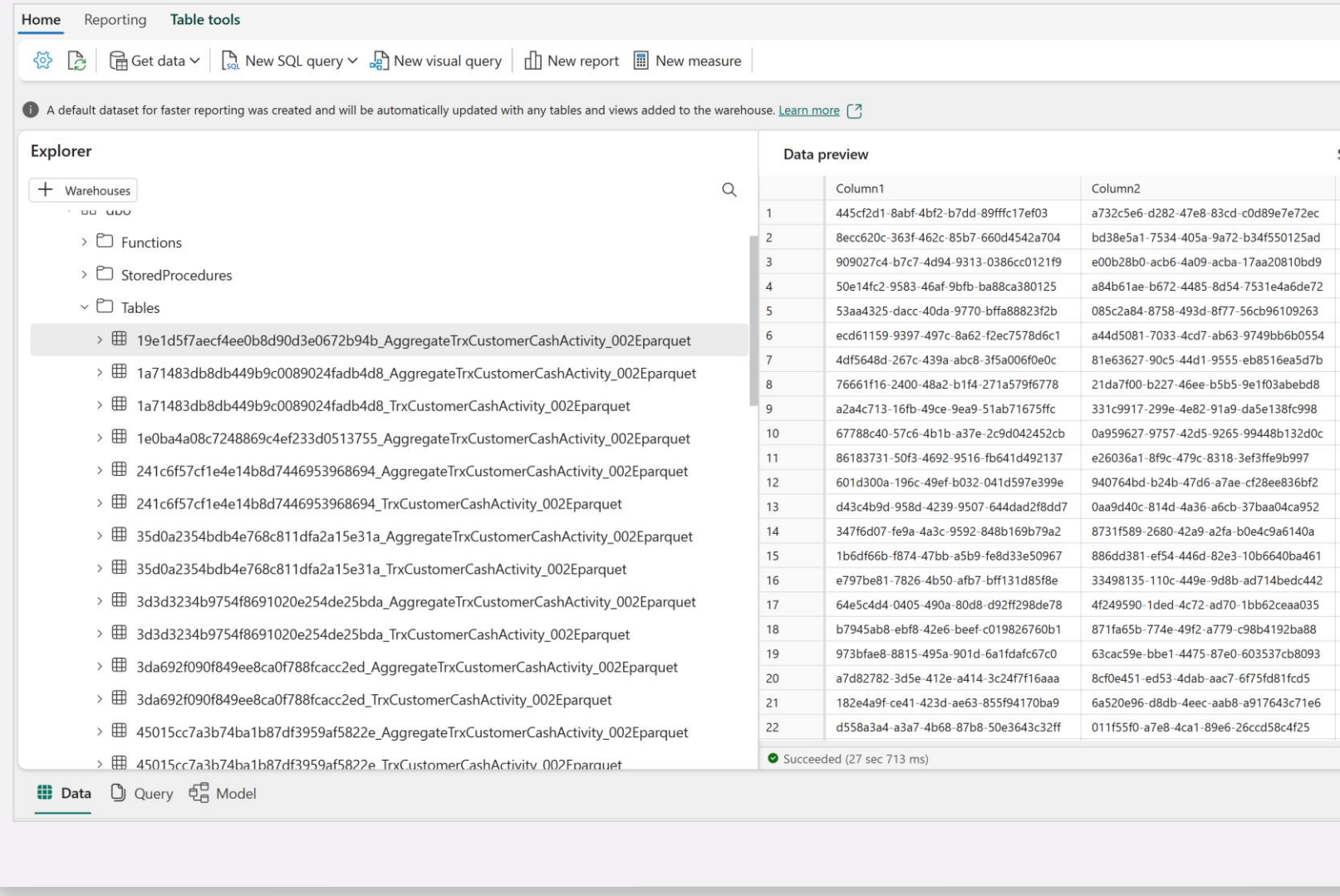
Computed tables in Staging

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.



Dataflow Gen2

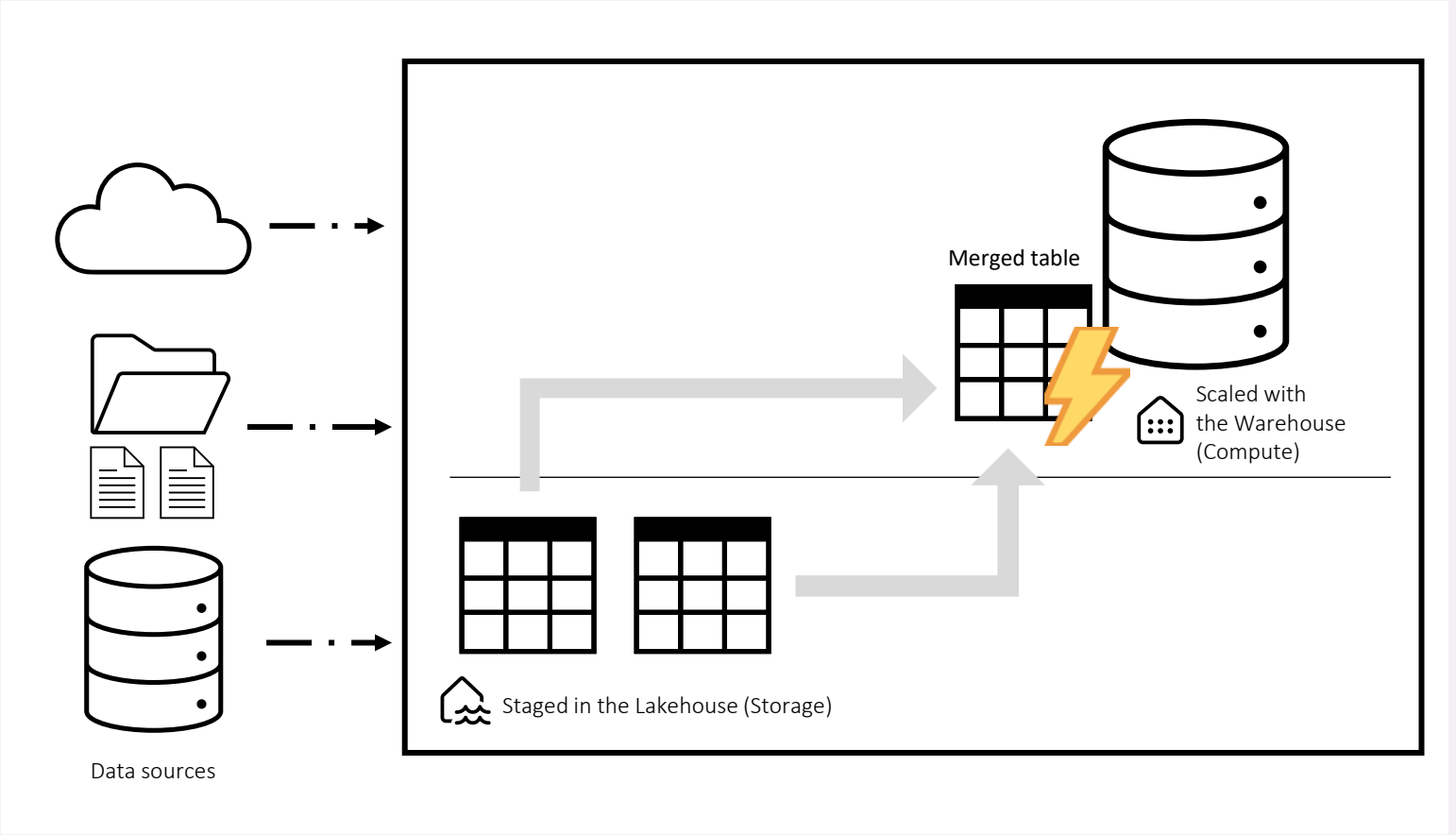
Optimize the use of dataflows with Fabric compute

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.

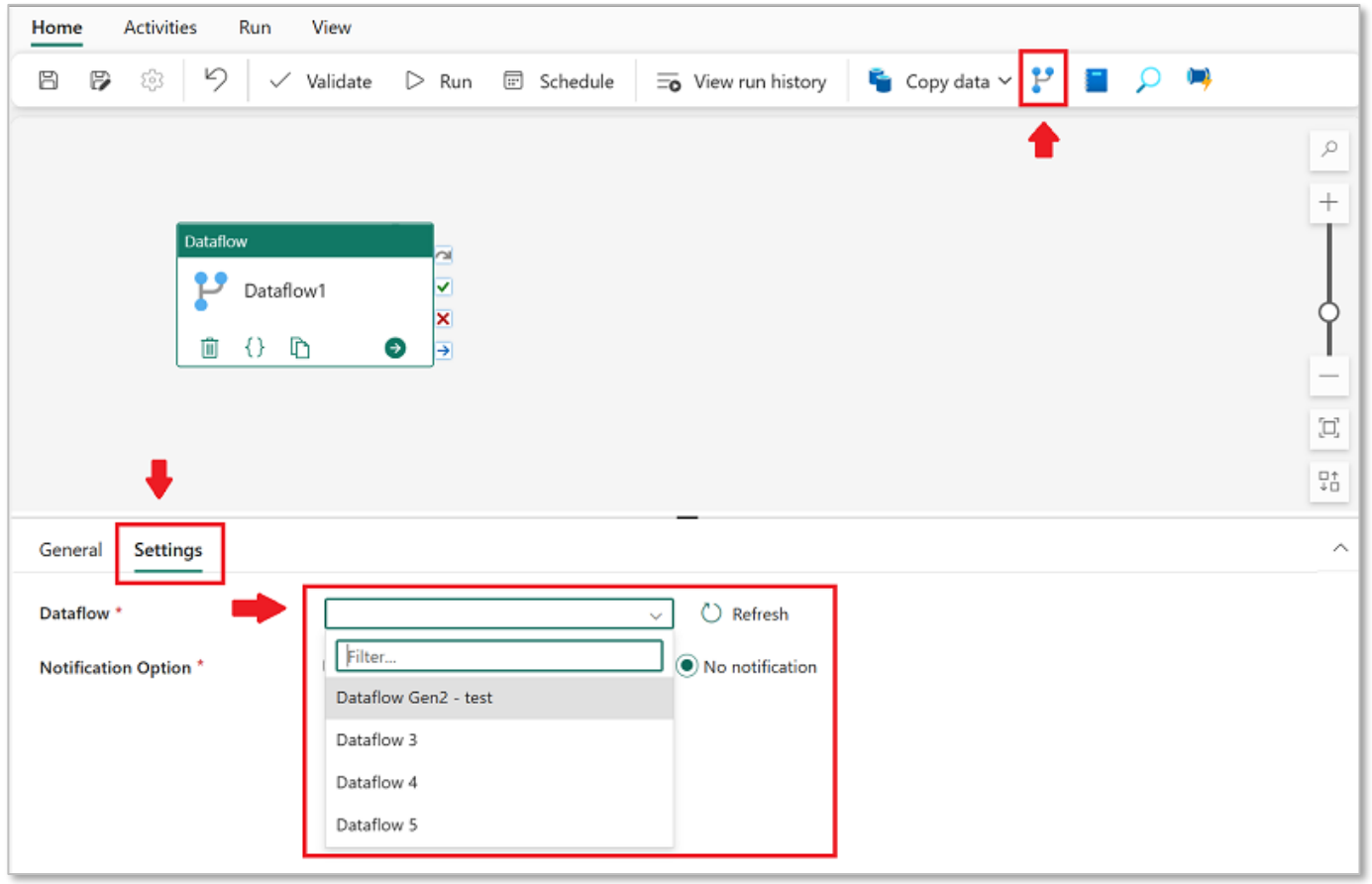
TRUE

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.

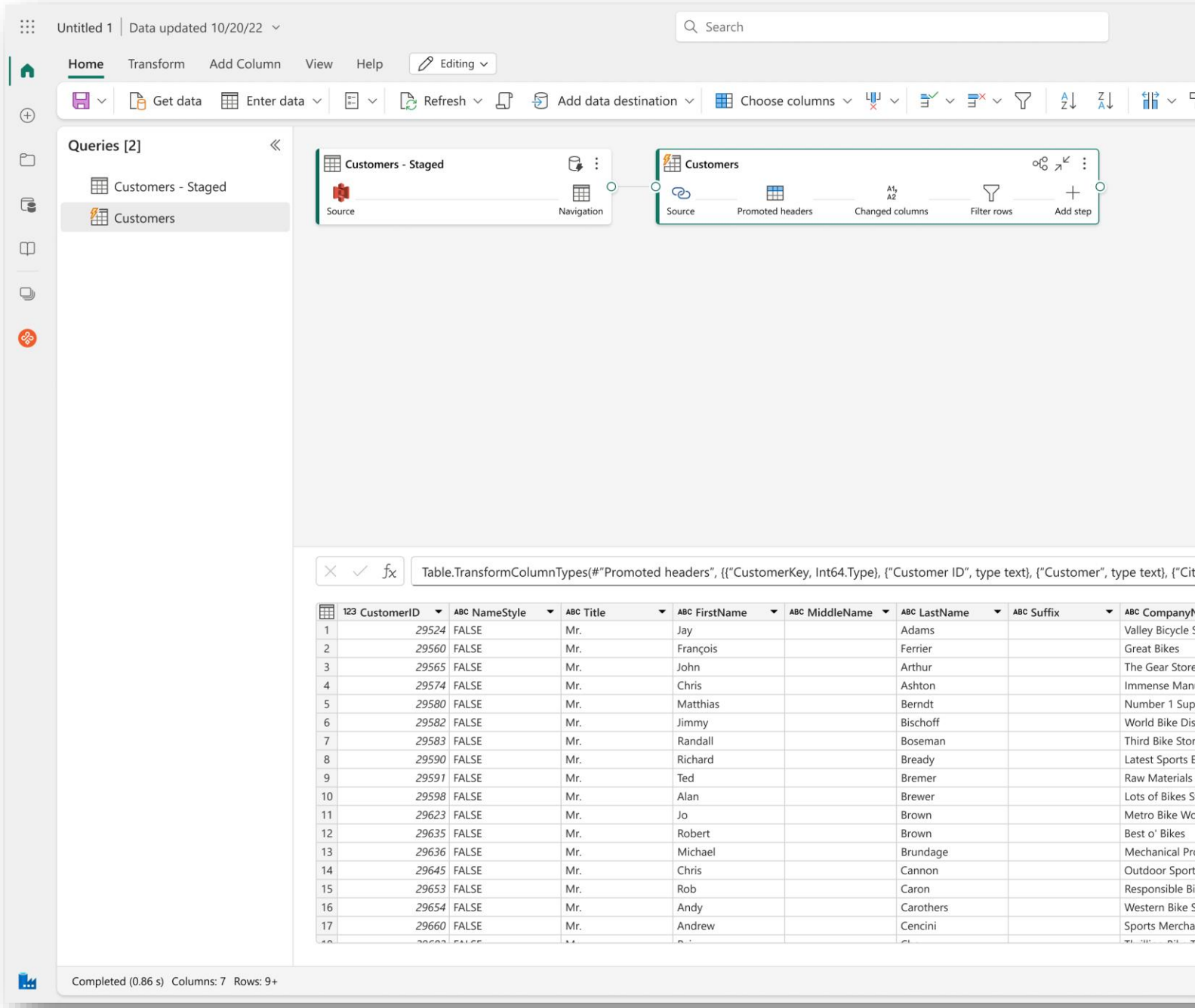
Additional considerations...

- 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

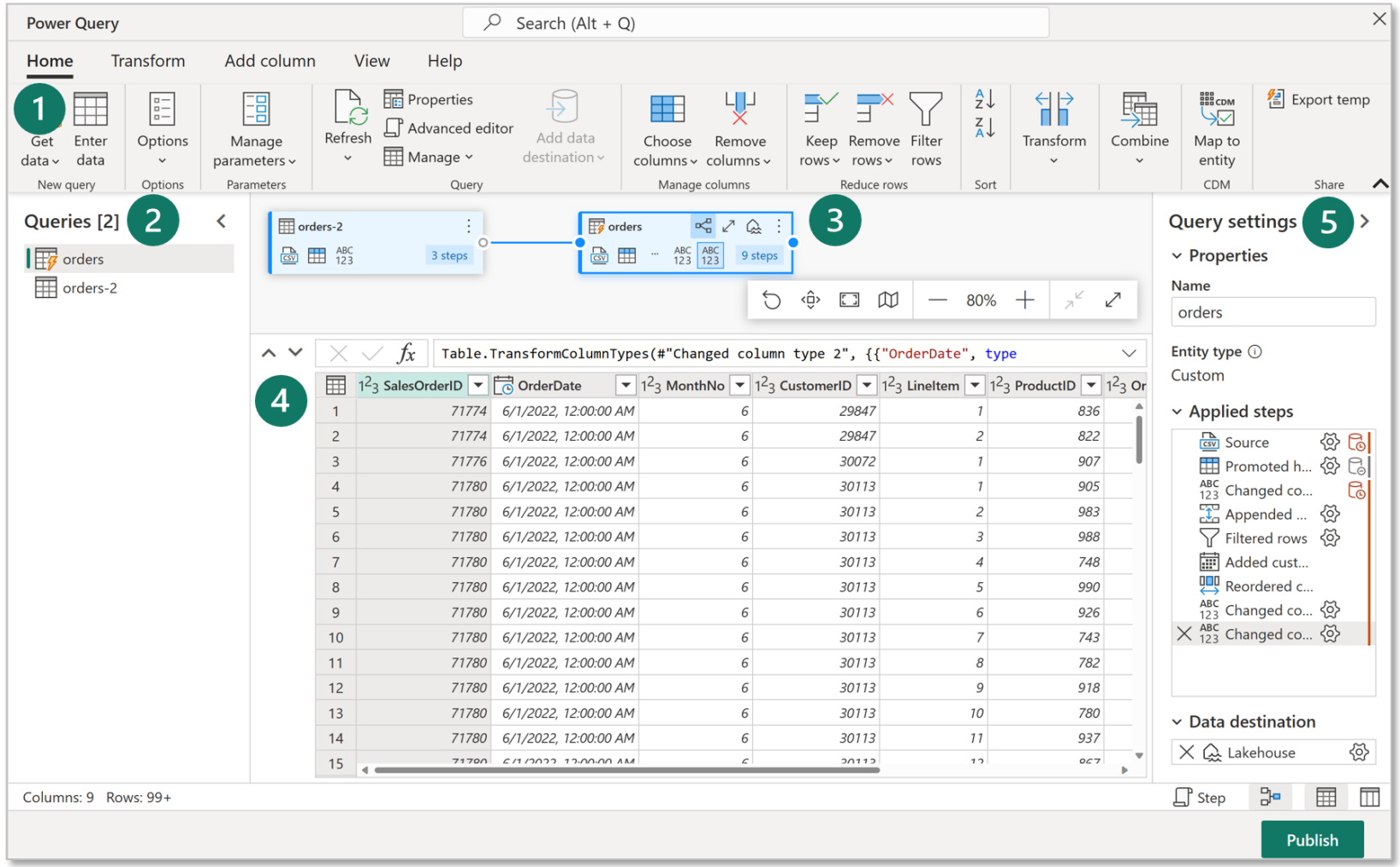
Abstracts away complexities of traditional ETL and ELT



Exploring Dataflows (Gen2)

Power Query editor

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



DEMO

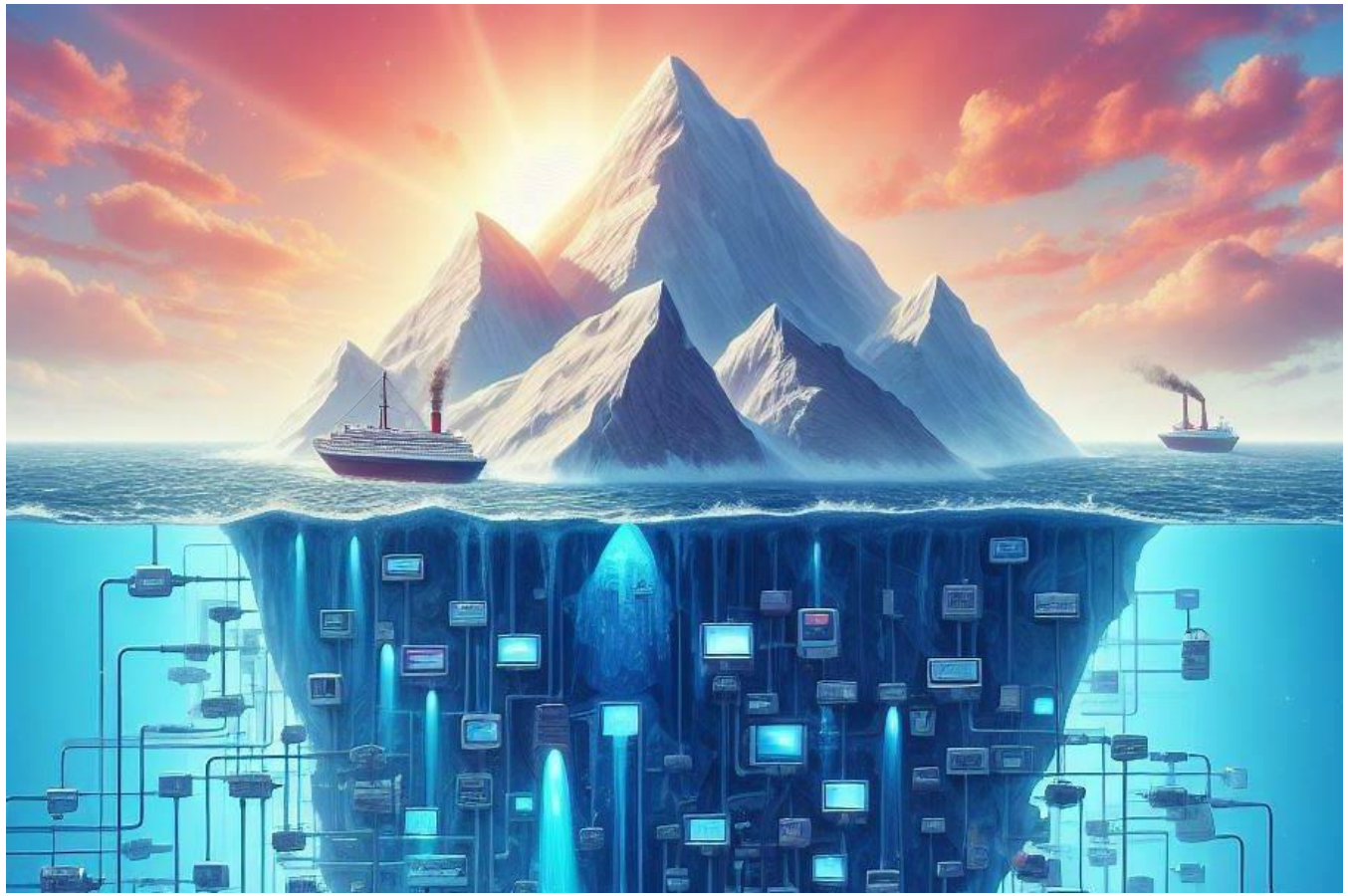
Dataflow (Gen2)

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