Introduction to Fabric ~~Data~~ Pipelines

Koen Verbeeck – He/him

Contact

Koen Verbeeck

@Ko_Ver

@koenv.bsky.social

SQLkover.com

Outline

01 What are Fabric Data Pipelines?

02 What are the building blocks?

03 Fabric Pipelines vs ADF

04 Tips & Tricks

Quick Poll

What are Fabric Data Pipelines?

Microsoft Fabric

Data Factory

Real-Time Intelligence

Databases

Analytics

Industry Solutions

Power BI

Partner solutions

Copilot in Fabric

OneLake

Microsoft Purview

A fully managed, ***serverless data integration*** *service.* ***Visually*** *integrate data sources with more than 170* ***built-in connectors*** *at no added cost. Easily construct* ***ETL and ELT*** *processes code-free in an intuitive environment or write your own code.*

Data Factory

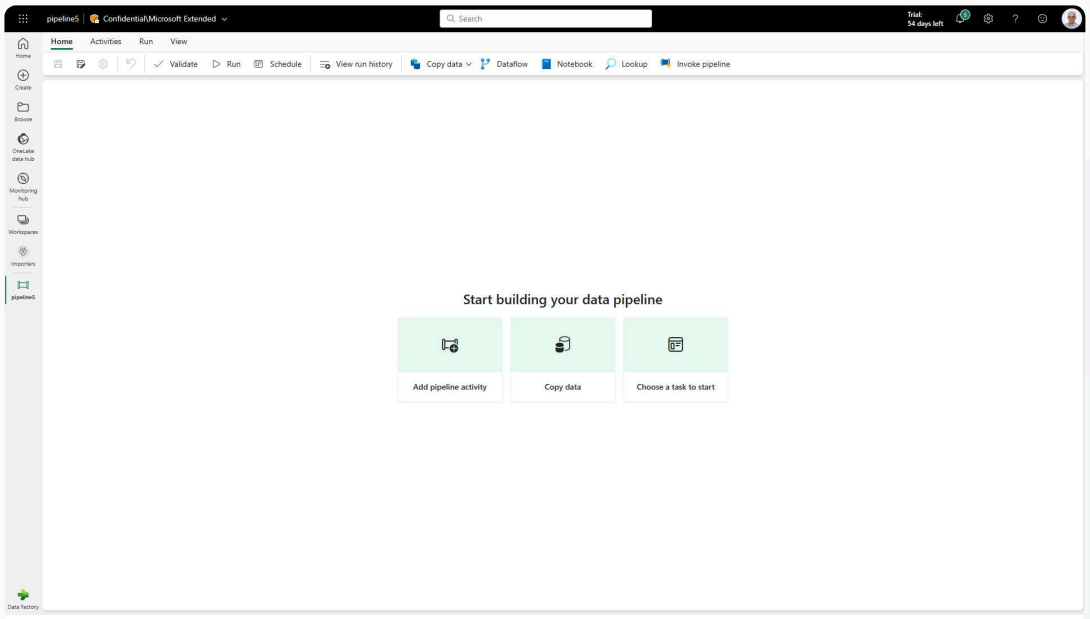
Data Factory in Microsoft Fabric provides cloud-scale data movement and data transformation services that allows you to solve the most complex ETL scenarios

Core to Data Factory are Data Pipelines and Dataflows to give users the option to a low-code, collaborative and enterprise scale approach for their ETL process

---

Key Capabilities:

- Latest capabilities:
- Output destination to Lakehouse
- 170+ connectors available in Data Factory
- Pipeline Lakehouse copy assist
- Create data pipeline in Lakehouse portal
- Pipeline templates
- Pipeline support for Spark notebooks
- Service principle auth support

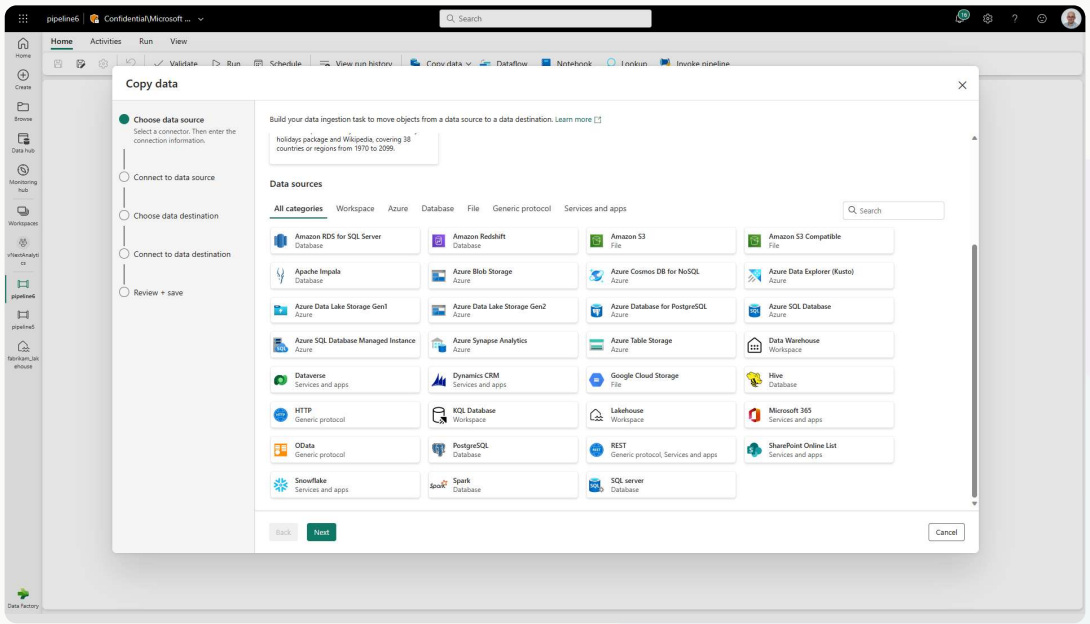


Autonomous ETL can unlock operational efficiencies and help orchestrate, monitor and manage pipeline performance.

Data Pipelines

Data Pipelines enable powerful workflow capabilities at cloud-scale like building complex workflows, moving PB-size data, and defining sophisticated control flow pipelines

Data pipelines can be used to build complex ETL and data factory workflows that can perform a number of different tasks at scale. Additionally, control flow capabilities are built into pipelines so you can build workflow logic which provide loops and conditional

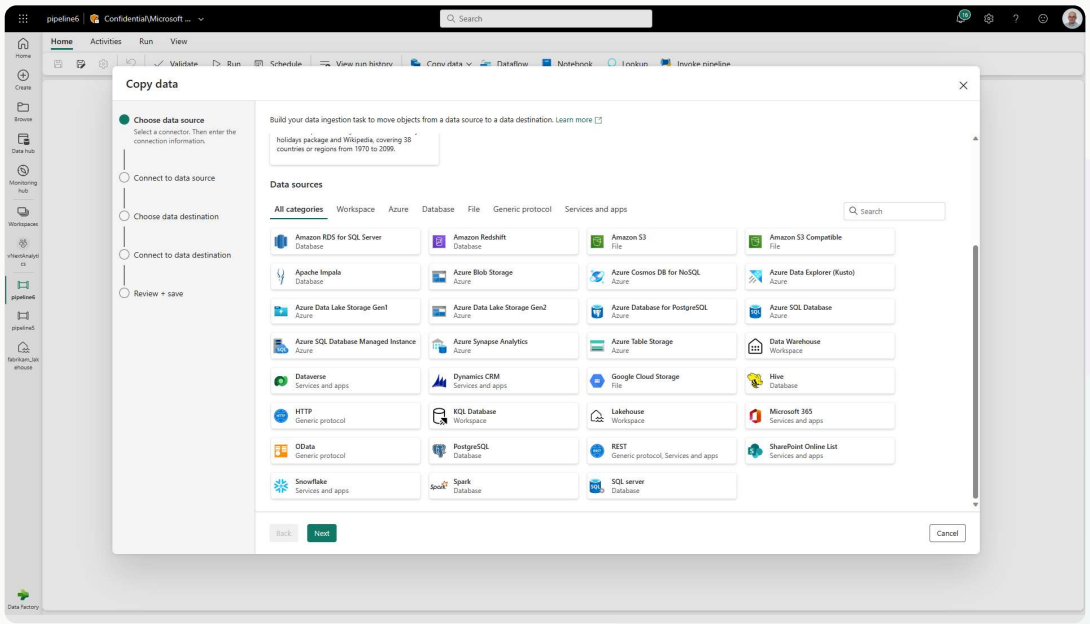


Data Pipelines | Connectors

New Connectors provide a low-code interface for ingesting data from a variety of data sources

Connectors:

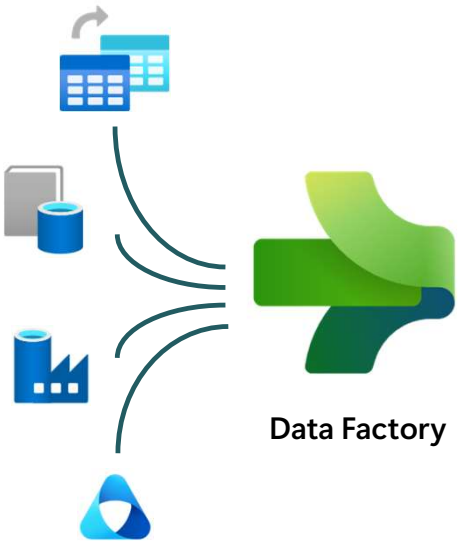
- Warehouse Connector; connect to existing Azure
- Lakehouse connector
- 100+ connectors in the copy activity
- Access to on-premises data
- Access protected data inside of a VNET



Unifying data in OneLake

Data Factory

Seamlessly connect to more than 170+ data stores



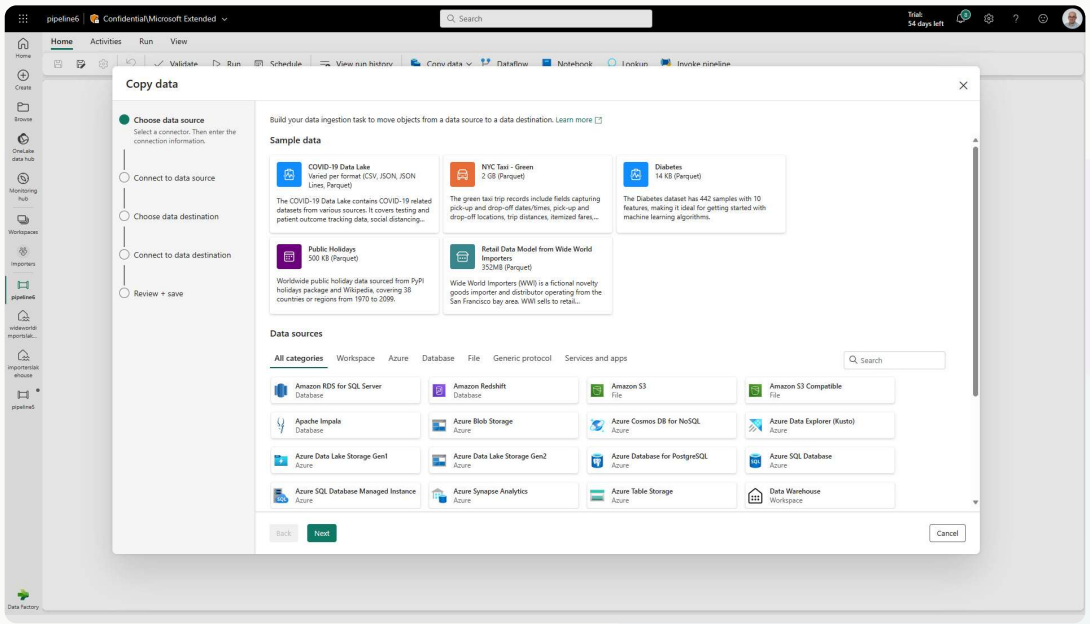
Azure Database for PostgreSQL	Azure Databricks Delta Lake	Amazon RDS for Oracle	Amazon RDS for SQL Server	Amazon Redshift	Phoenix	PostgreSQL	Presto	Magento (Preview)
Azure SQL Database	Azure SQL Database Managed Instance	Apache Impala	Azure SQL Database Managed Instance	DB2	SAP BW Open Hub	SAP BW via MDX	SAP HANA	Oracle Eloqua (Preview)
Azure Table Storage	MongoDB Atlas	Drill	Google AdWords	Google BigQuery	SAP Table	SQL server	Spark	PayPal (Preview)
Azure Cosmos DB (MongoDB API)	Azure Cosmos DB (SQLAPI)	Greenplum	HBase	Hive	Amazon S3	Amazon S3 Compatible	FTP	SAP Cloud For Customer
Azure Data Lake Storage Gen1	Azure Data Lake Storage Gen1 for Cosmos Structured Stream	Informix	MariaDB	Microsoft Access	File system	Google Cloud Storage (S3APD)	HDFS	Salesforce Marketing Cloud
Azure Data Lake Storage Gen1 for Cosmos Structured Stream	Azure Database for MariaDB	MySQL	Netezza	Oracle	HTTP	Oracle Cloud Storage (S3AP)	SFTP	Shopify (Preview)
Teradata	Vertica	ODBC	OData	REST	Amazon Marketplace Web Service	Concur (Preview)	Dataverse (Common Data Service for App)	Web Table
Jira	Kusto	SharePoint Online List	Dynamics 365	Dynamics AX	Dynamics CRM	Cassandra	Couchbase (Preview)	MongoDB

Data Pipelines | Sample data

Sample Datasets helps new users get started quickly, building out their ELT processes using Data Pipelines

Sample datasets:

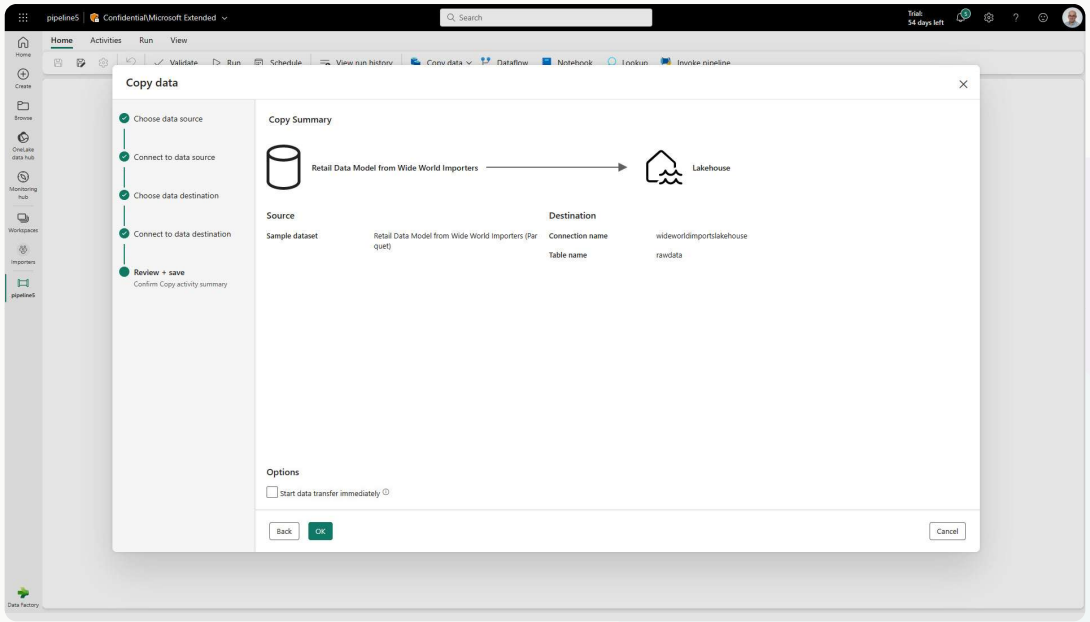
- COVID-19 Data Lake (CSV, JSON, JSON Lines, Parquet)
- NYC Tax – Green (2GB Parquet)
- Diabetes (14K Parquet)
- Public Holidays (500KB Parquet)
- Retail Data Model from Wide World Importers (352MB Parquet)



Data Pipelines | Lakehouse copy assist

Simply copying data to a Lakehouse with copy assist capabilities within the Data Pipeline

Additionally, users can create a Data Pipeline without having to leave the Lakehouse portal



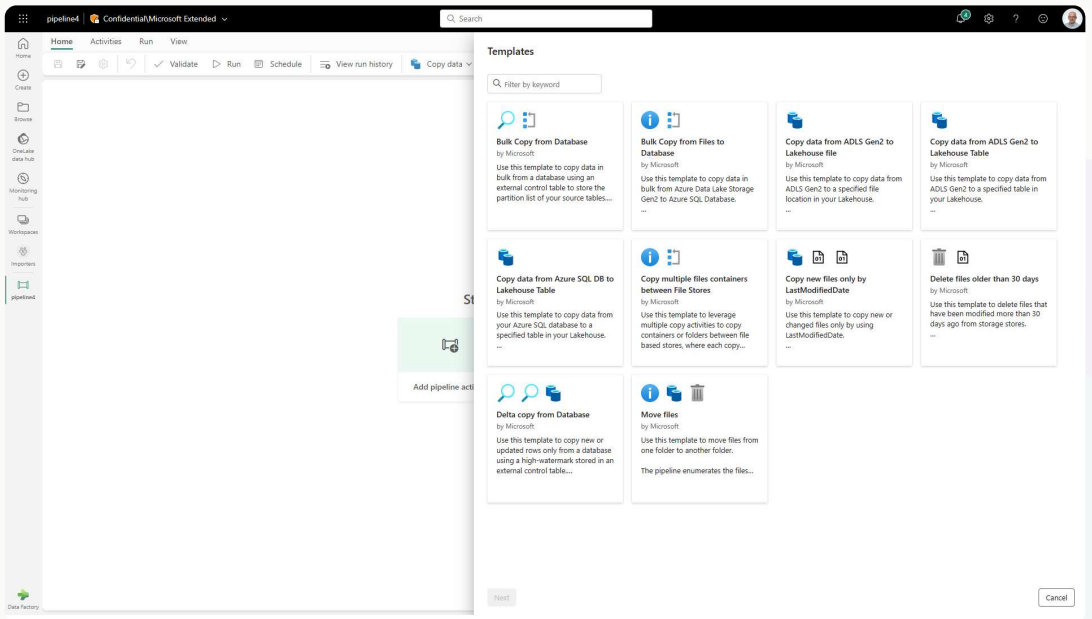
Data Pipelines | Templates

Quickly get started with data integration

Template help reduce development time by providing an easy way to create pipeline for common data integration scenarios

Available Data Pipeline Templates:

- Bulk copy from Database
- Bulk copy from File to Database
- Copy data from ADLS Gen2 to Lakehouse file
- Copy from ADLS Gen2 to Lakehouse Table
- Copy data from Azure AQL DB to Lakehouse Table
- Copy multiple files containers between File Stores
- Copy new files only by Last Modified Date
- Delete files older than 30 days
- Delta copy from Database
- Move files



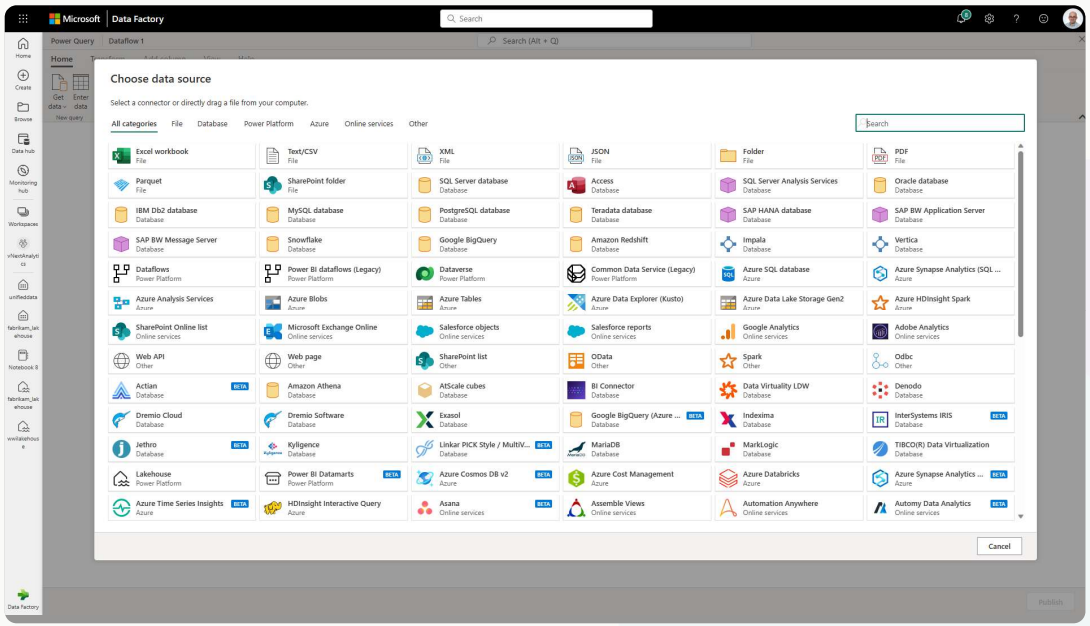
Dataflow

Dataflow provides a low-code interface for ingesting data from hundreds of data sources

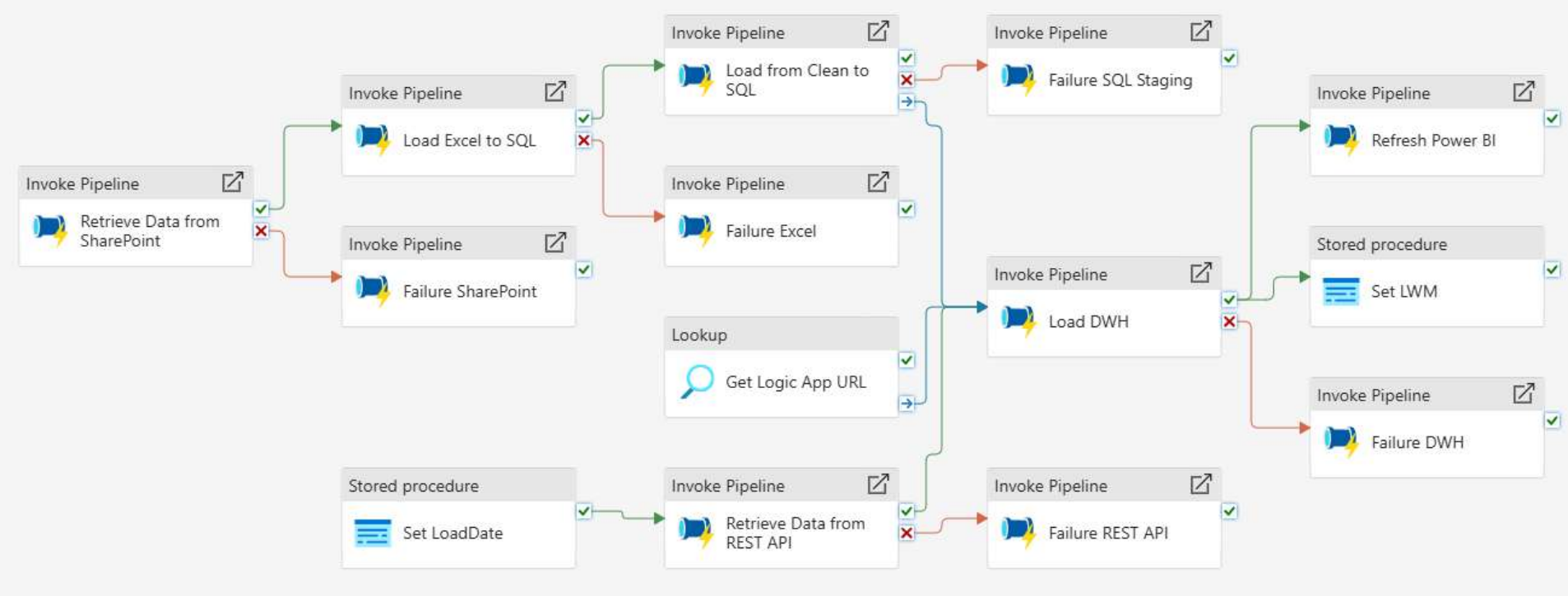
Dataflow quickly and easily unify disparate data sources, establish a more collaborative analytics approach, and promote more informed, agile decision making.

Key Capabilities:

- Accelerate data transformation with code-free data flows
- Scale out using Fabric compute and Data Factory fast copy
- Load results of data transformations into multiple destinations (Azure SQL Databases, Lakehouse, etc.)



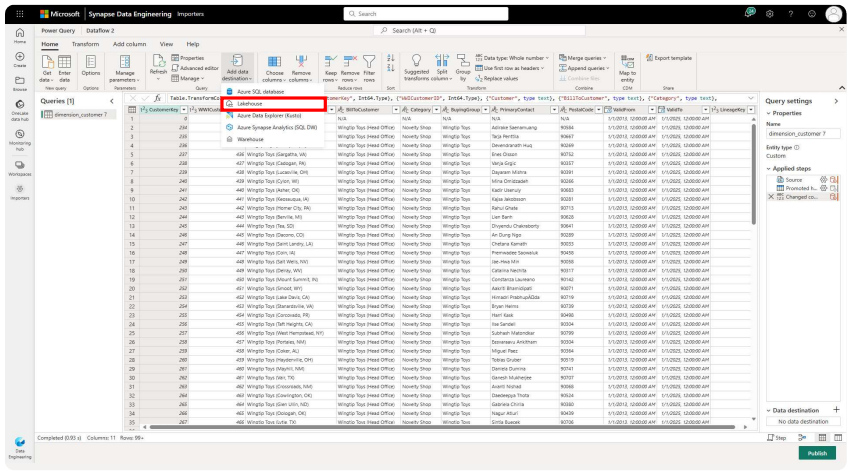
Basically a cloud ELT orchestrator



What about ETL?

- Transformations are done in other Fabric Compute
  - SQL in Warehouse
  - Spark/SQL in Notebook (or Spark Job)
  - KQL in Eventhouse
  - AI stuff in shortcuts?

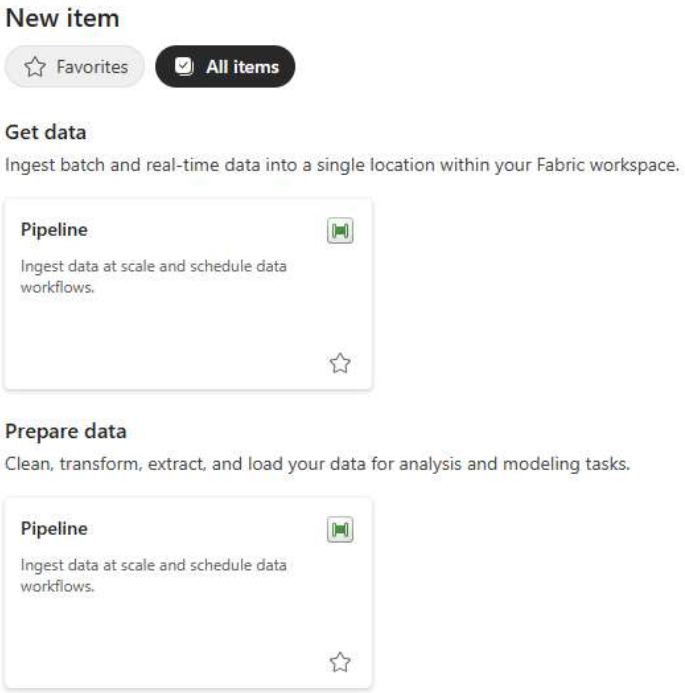
- Dataflows Gen 2
  - Aka Power Query on steroids
  - Many improvements w.r.t. Power BI Dataflows
  - But not really suited for scale?

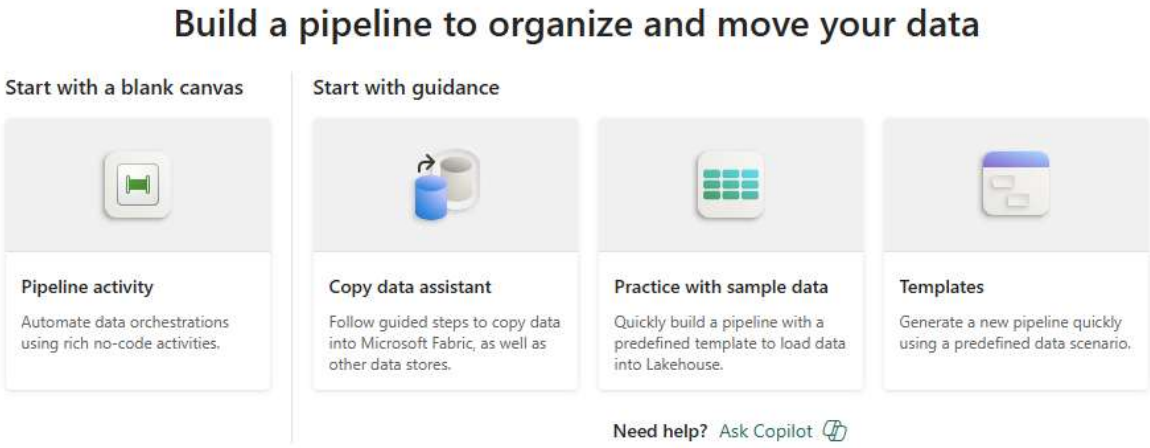


demo

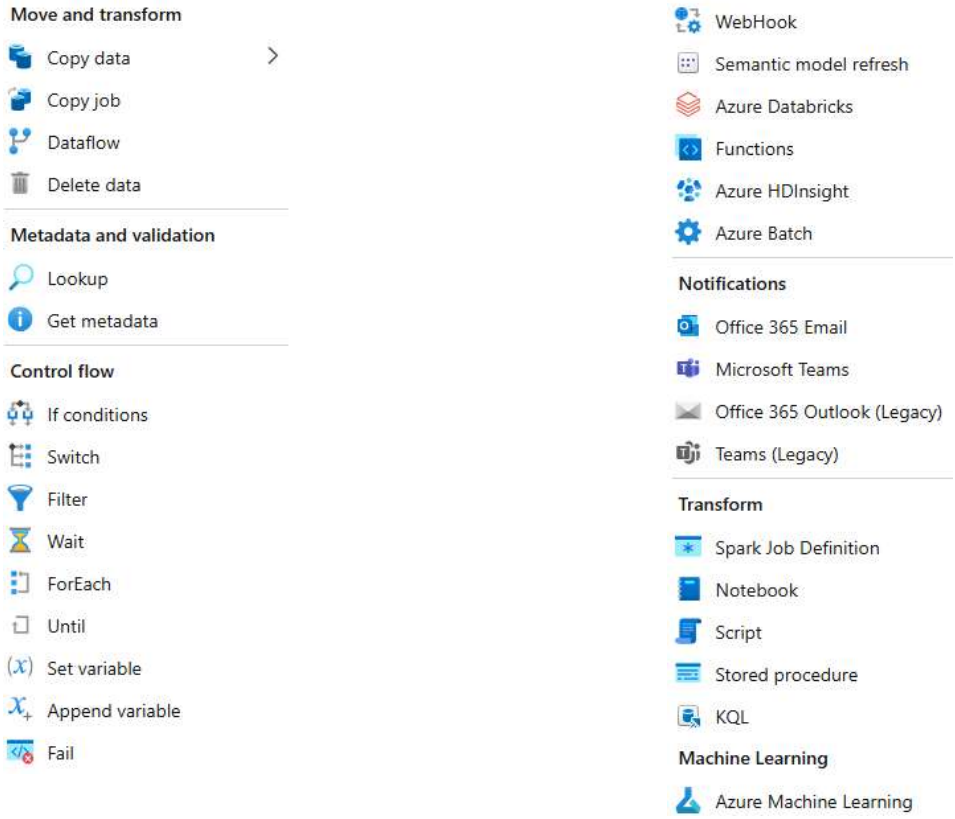
What are the building blocks?

Pipeline


New item
Favorites
All items
Get data
Ingest batch and real-time data into a single location within your Fabric workspace.
Pipeline
Ingest data at scale and schedule data workflows.
Prepare data
Clean, transform, extract, and load your data for analysis and modeling tasks.
Pipeline
Ingest data at scale and schedule data workflows.


Build a pipeline to organize and move your data
Start with a blank canvas
Start with guidance
Pipeline activity
Automate data orchestrations using rich no-code activities.
Copy data assistant
Follow guided steps to copy data into Microsoft Fabric, as well as other data stores.
Practice with sample data
Quickly build a pipeline with a predefined template to load data into Lakehouse.
Templates
Generate a new pipeline quickly using a predefined data scenario.
Need help? Ask Copilot

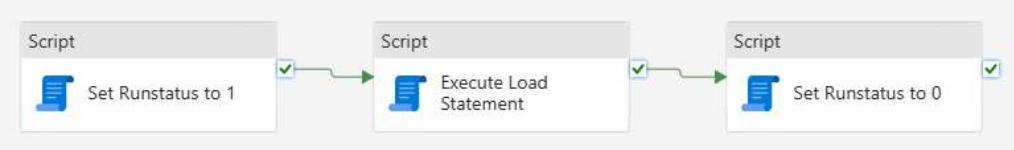
Activities


Move and transform
Copy data
Copy job
Dataflow
Delete data
Metadata and validation
Lookup
Get metadata
Control flow
If conditions
Switch
Filter
Wait
ForEach
Until
Set variable
Append variable
Fail
Orchestrate
Invoke Pipeline
Invoke Pipeline (Legacy)
Web
WebHook
Semantic model refresh
Azure Databricks
Functions
Azure HDInsight
Azure Batch
Notifications
Office 365 Email
Microsoft Teams
Office 365 Outlook (Legacy)
Teams (Legacy)
Transform
Spark Job Definition
Notebook
Script
Stored procedure
KQL
Machine Learning
Azure Machine Learning

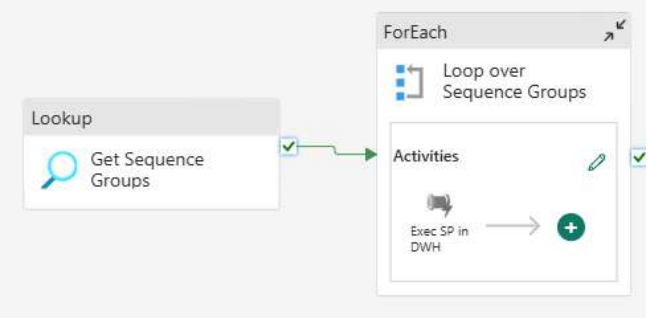
Three types of activity

- **Data Movement** activities
  - Copy Data activity and alternatives
  - Data is copied using Fabric compute (CU)

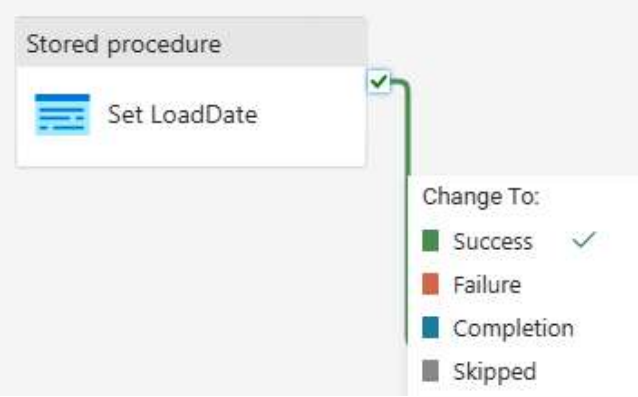
- **External** (transformation) activities
  - Compute is not in the Pipeline
  - E.g. stored procedure, Notebook, Azure/Fabric Function …

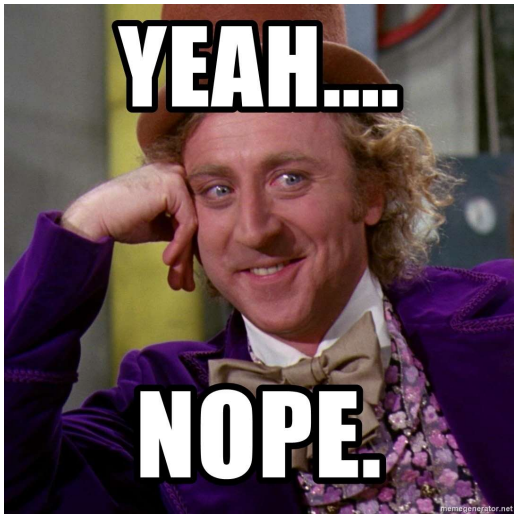


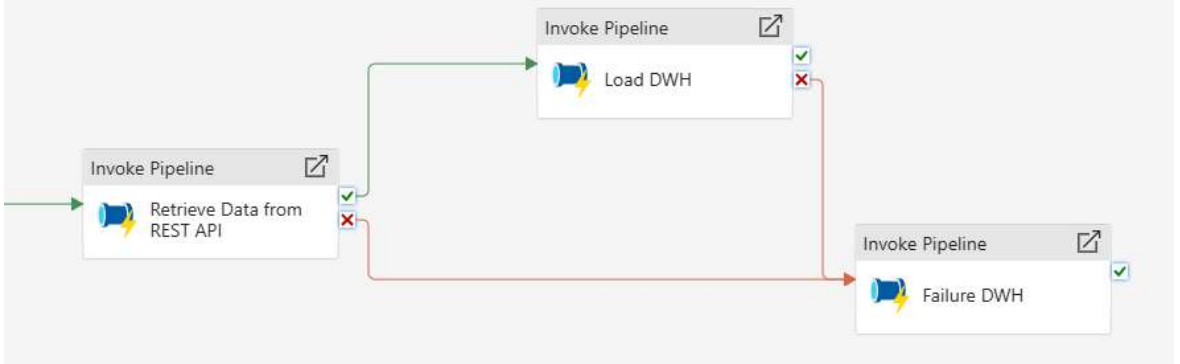
- **Pipeline** (control flow) activities
  - Lightweight "internal" activities
  - E.g. Lookup, Set Variable, For Each, If condition …



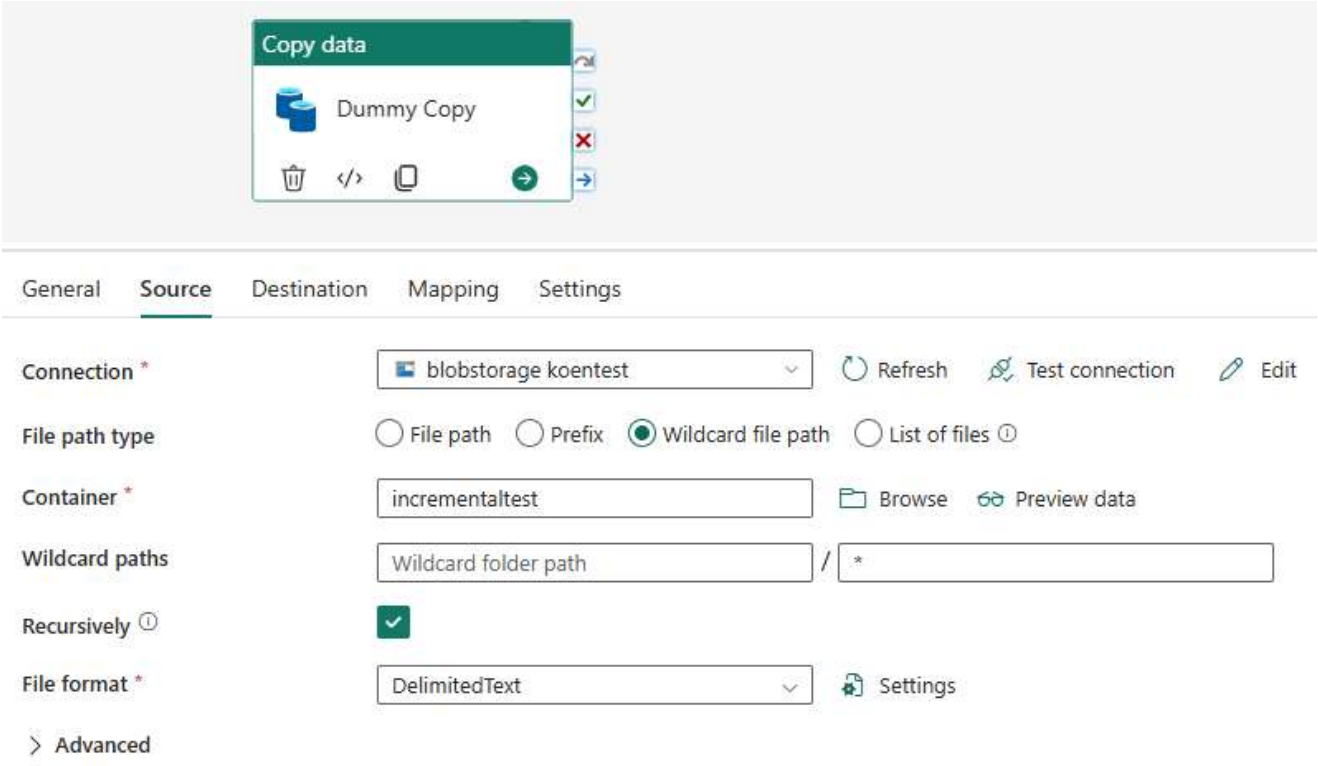
Activity dependency



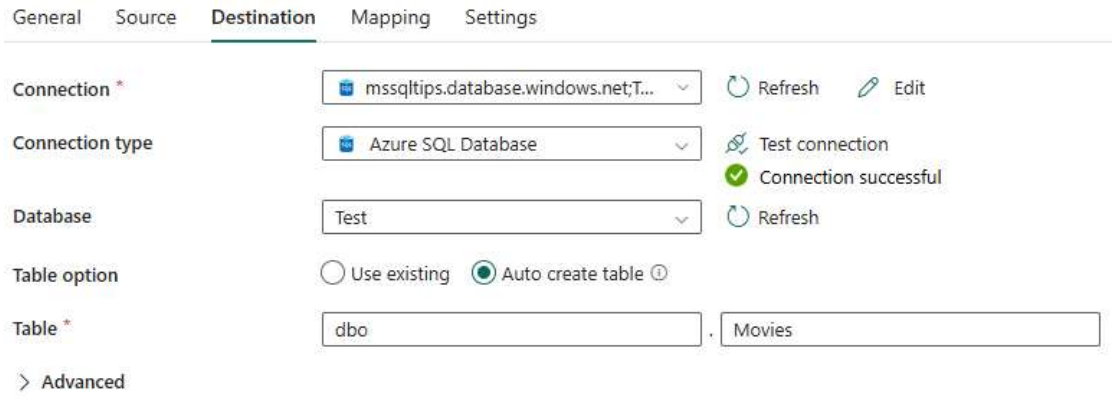


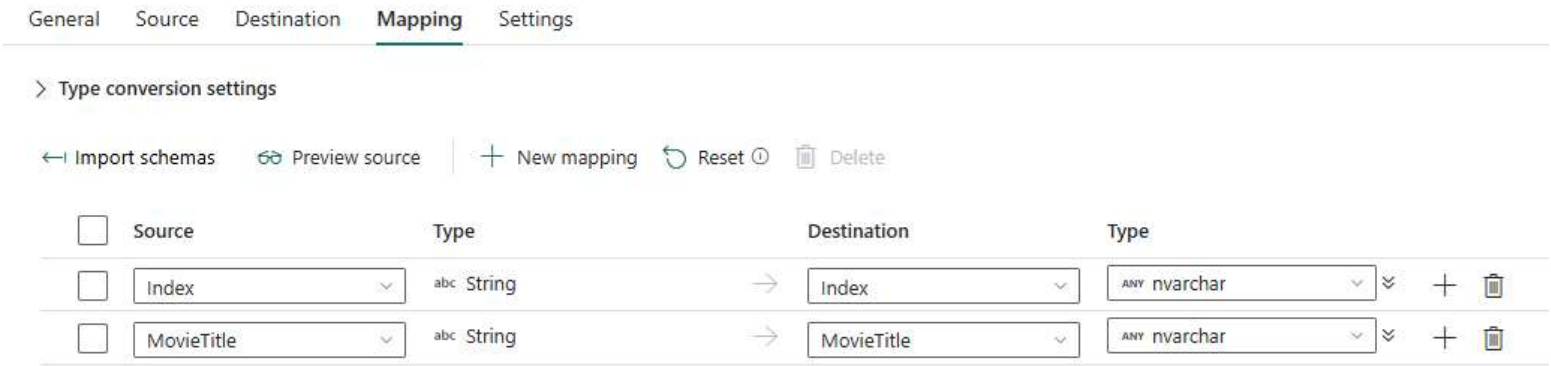


Copy data activity



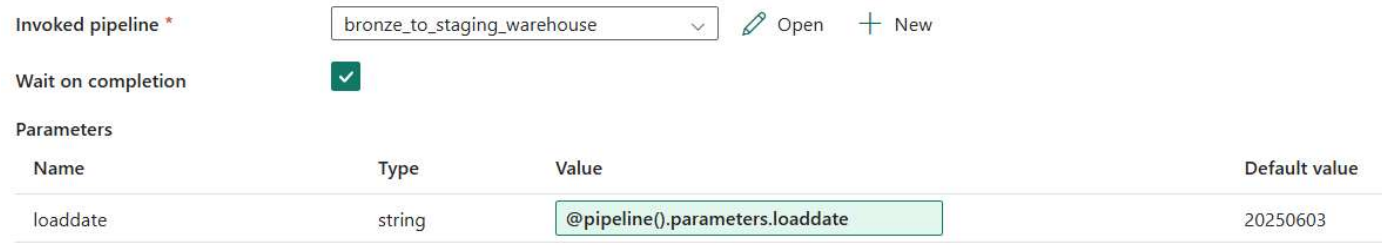
Copy data activity

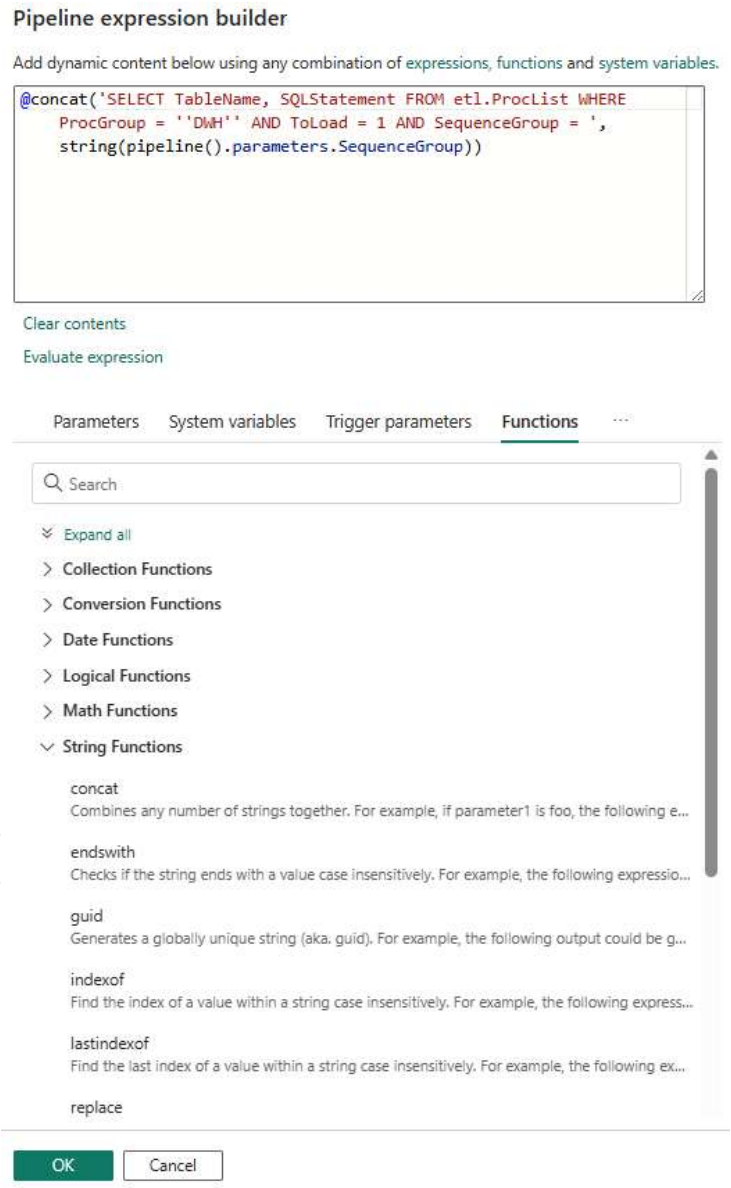




Dynamic Content

- Pipelines support **parameters** & **variables**
- On many fields you can define **dynamic content**
- You can use these to make your pipelines more flexible and dynamic





Copy Job

- New item type
- Simplified data movement, like Copy Data activity
- Supports
  - Full copy / Incremental copy / CDC replication
  - Truncate destination
  - Append / overwrite / merge
- Copy Job activity in Fabric Pipeline for orchestration

Copy Job

	Pipeline copy activity	Copy job
Use case	Data lake and data warehouse migration, data ingestion,	Data Ingestion, Incremental copy, plication, d Data Warehouse igration, t transformation
Primary developer persona		ess Analyst, Integrator, a Engineer
Primary developer skill set	ETL	ETL, SQL, JSON
Code written		o code, w code
Data volume		w to high
Development interface	canvas	Wizard, canvas
Sources	50+ connectors	50+ connectors

https://learn.microsoft.com/en-us/fabric/fundamentals/decision-guide-pipeline-dataflow-spark

demo

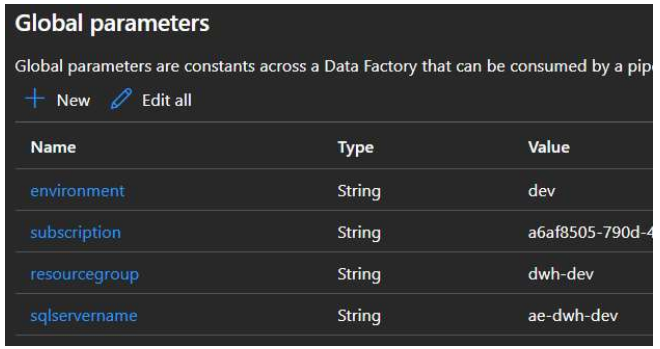
Fabric Pipelines vs ADF

ADF vs Fabric Pipelines

- They are basically the same
- … aside from some differences

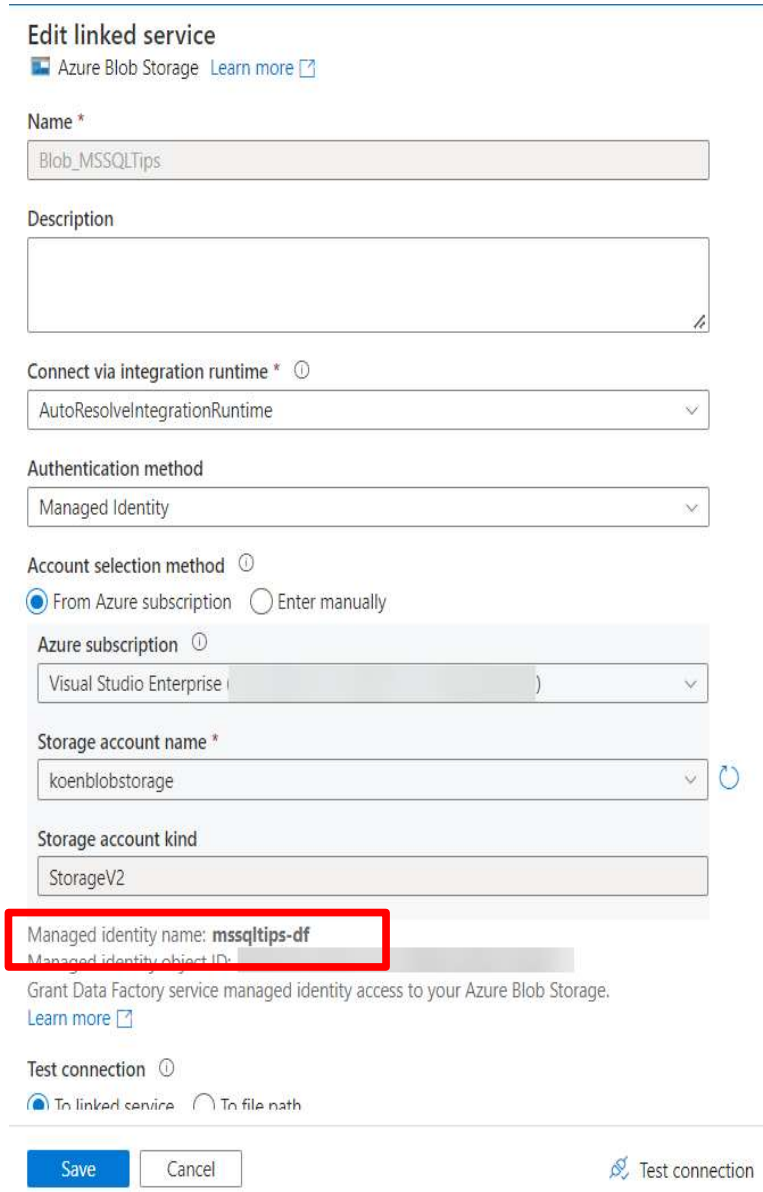
- ADF and Synapse Pipelines are almost equal
  - Different Devops task for deploying
  - No global parameters in Synapse
  - UI is slightly different

- There are more discrepancies between Fabric and ADF/Synapse



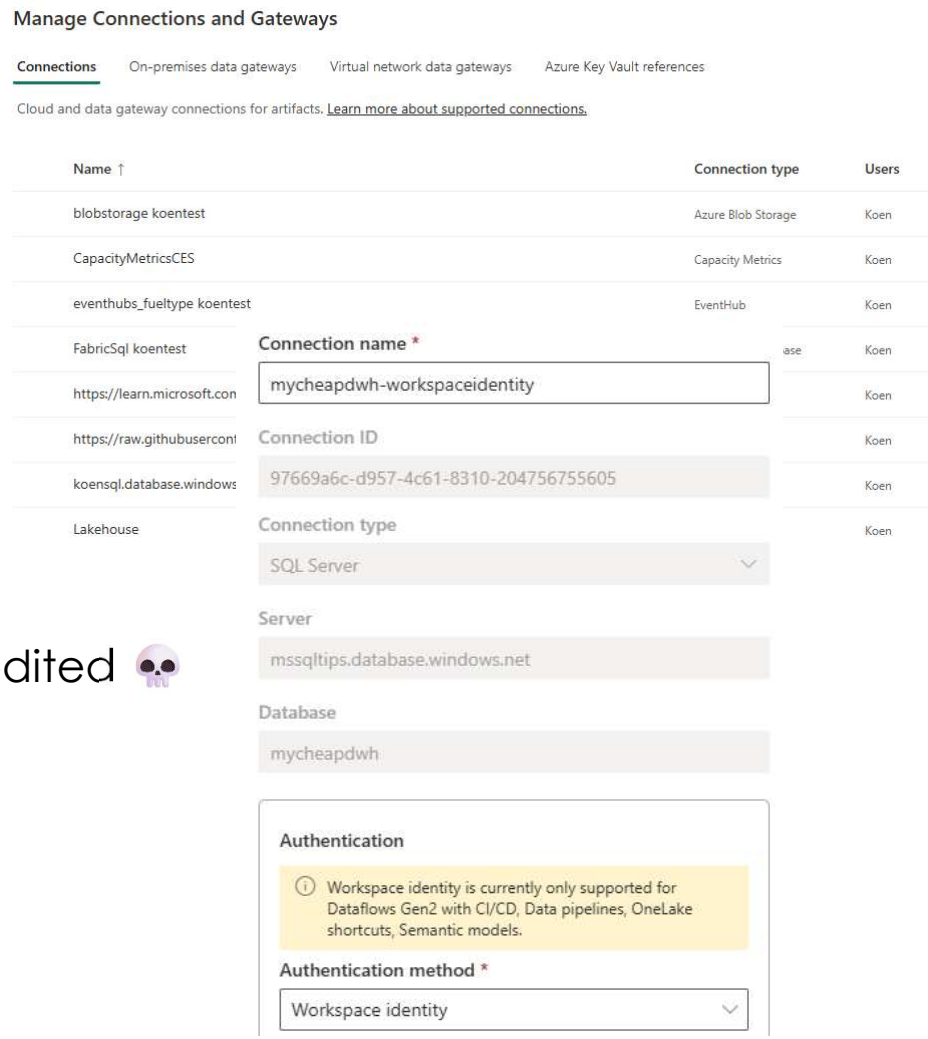
Linked service ≈ Fabric connection

- points to a certain data location
- contains info on how to authenticate
- can integrate with Azure Key Vault



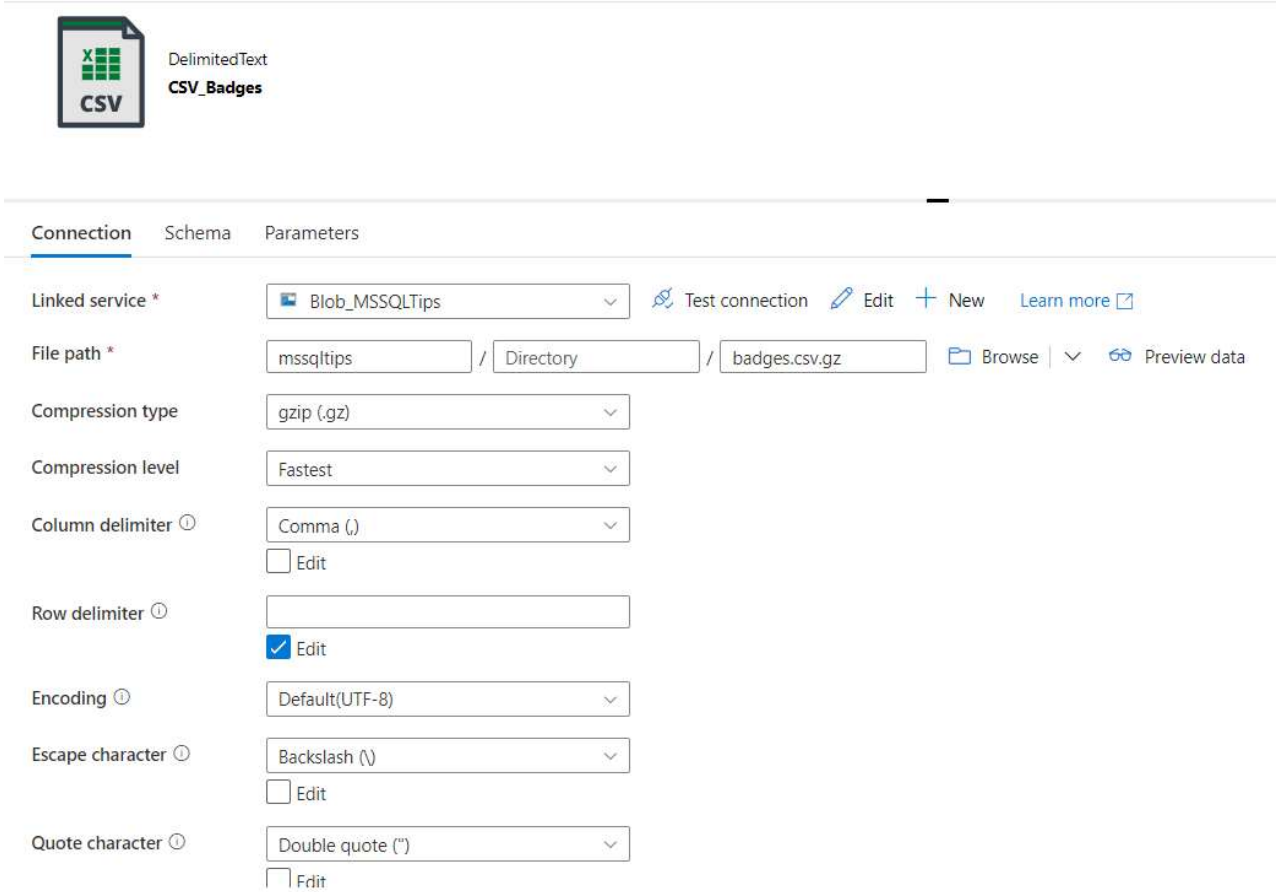
Linked service ≈ Fabric connection

- Fabric connections have an owner 🥹
- By default, Fabric adds your username 😖
- After creation, only authentication can be edited 💀



Datasets

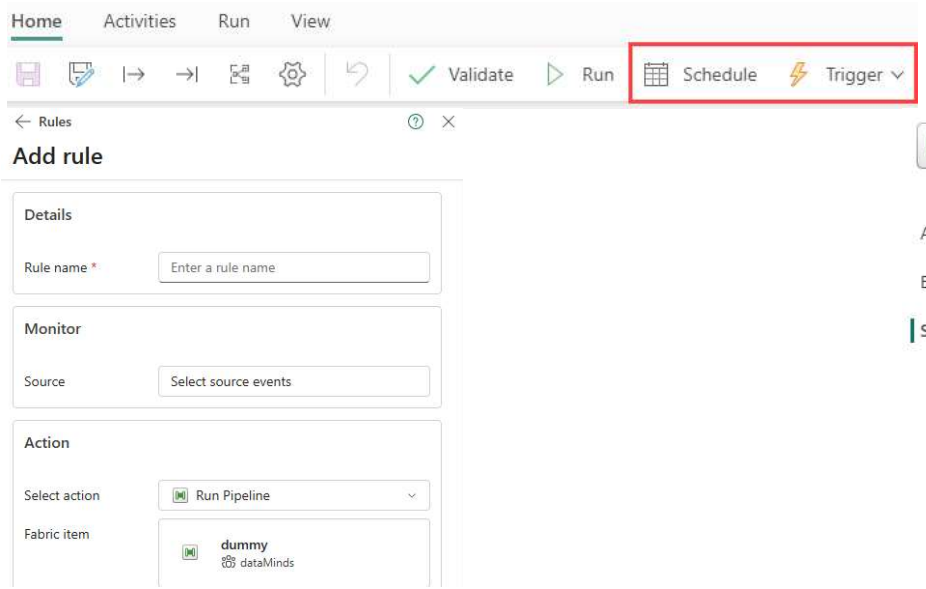
Don’t exist in Fabric 😱

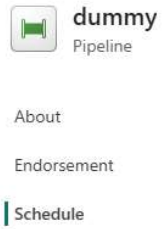


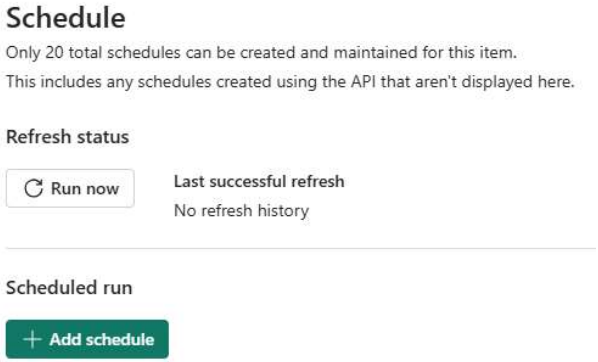
Triggers

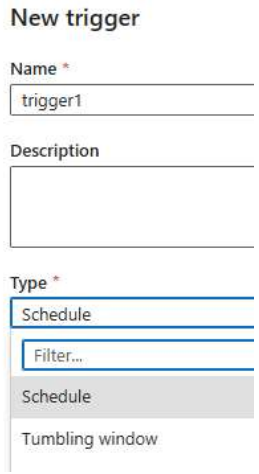
- In ADF/Synapse, everything that can kick off a pipeline is a trigger

- In Fabric, there's a distinction made between schedules and triggers









Integration runtimes

Azure

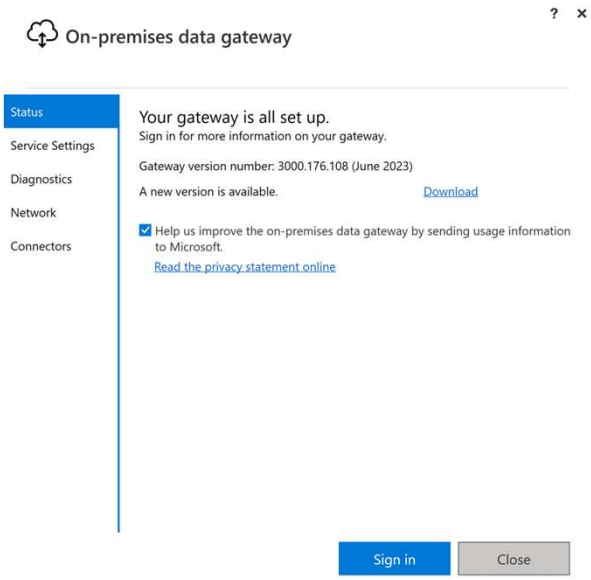
Use this for running data flows, data movement, external and pipeline activities in a fully managed, serverless compute in Azure.

Name	Type	Sub-type	Status	Related	Region
AutoResolveIntegrationRuntime	Azure	Public	Running	0	Auto Resolve

Self-Hosted

Use this for running activities in an on-premises / private network

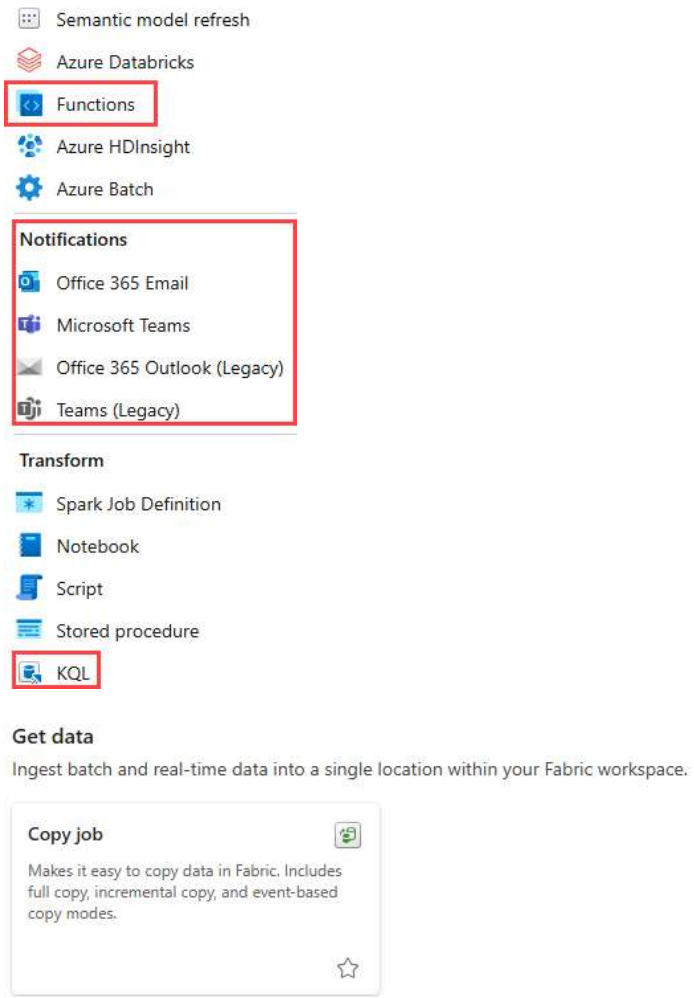
View more

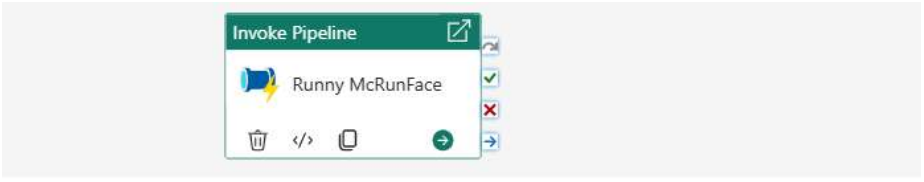


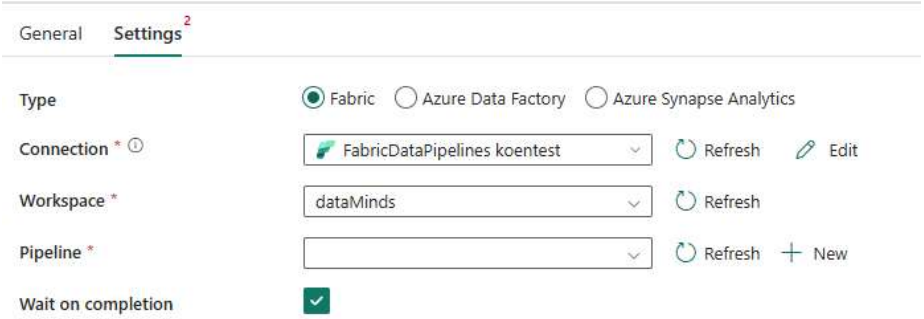
Azure-SSIS

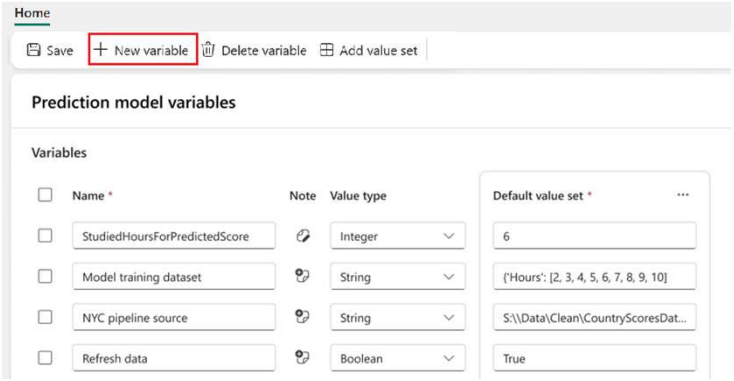
Lift-and-shift existing SSIS packages to execute in Azure.

Only in Fabric…









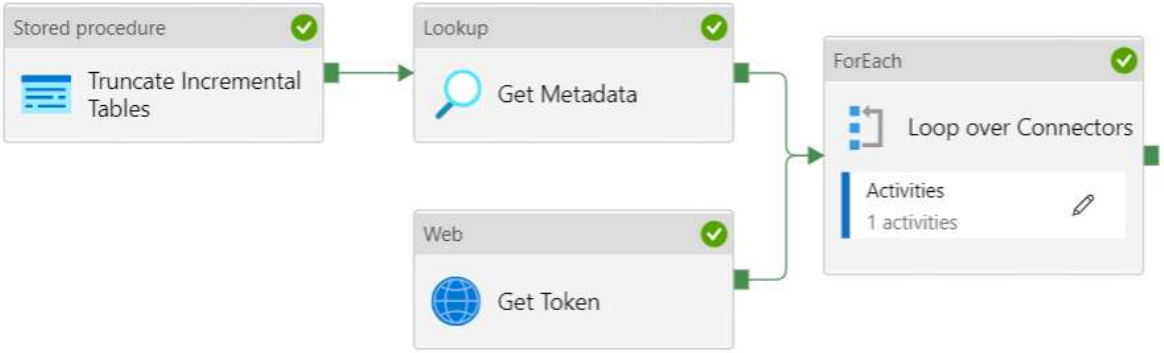
Costs in ADF

https://youtube.com/watch?v=Fct1dcZMyWs

Important to remember

ALL ACTIVITIES ARE PRO-RATED BY THE **MINUTE** AND ROUNDED **UP**

Example



2 minutes and 23 seconds runtime

Parameters | Variables | Settings | **Output**

Showing 101 - 104 of 104 items

Name	Type	Run start	Duration	Status	Integr
Loop over Connectors	ForEach	2022-03-27T09:08:02.63	00:02:14	Succeeded	
Get Metadata	Lookup	2022-03-27T09:07:58.71	00:00:03	Succeeded	AutoR
Truncate Incremental Tables	Stored proc...	2022-03-27T09:07:55.25	00:00:03	Succeeded	AutoR
Get Token	Web	2022-03-27T09:07:55.25	00:00:03	Succeeded	AutoR

Example

Debug run consumption

	Quantity	Unit
Pipeline orchestration		
Activity runs	104	Activity runs
Pipeline execution		
Azure integration runtime		
Data movement activities	1.6667	DIU-hour
Pipeline activities	0.0167	Execution hours
External activities	0.0333	Execution hours

1 minute
2 minutes

100 minutes!

Orchestration (Azure-IR): $0.001/run

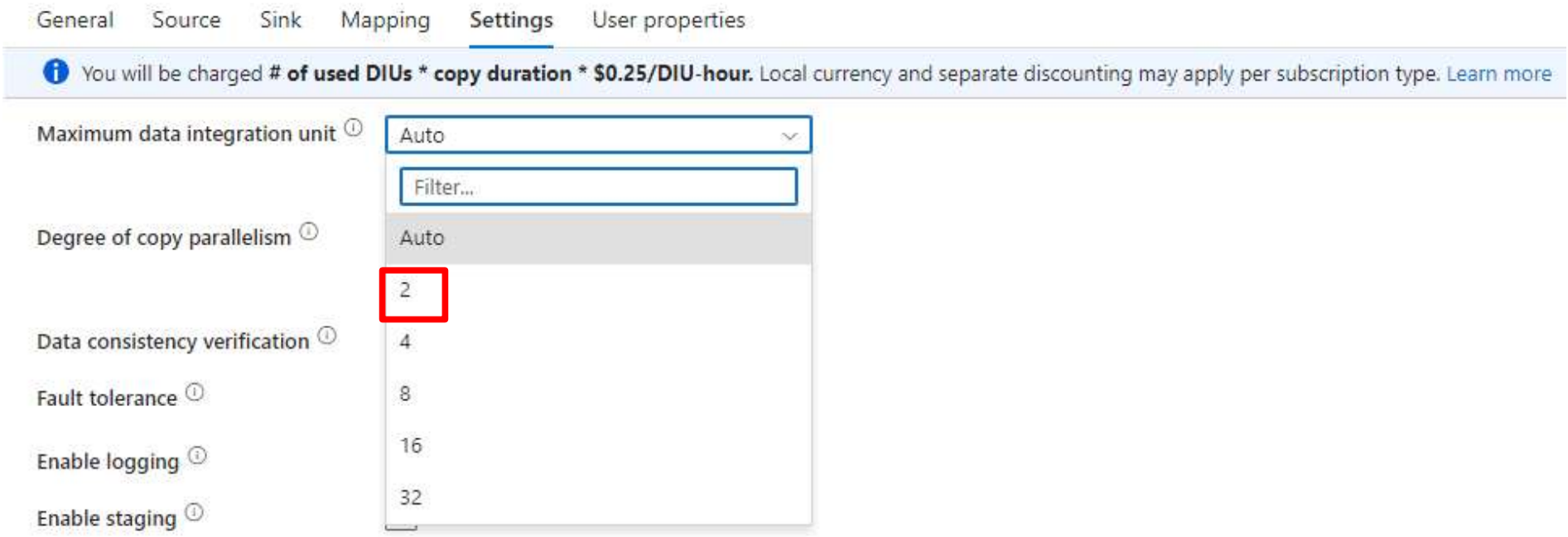
Pipeline (Azure-IR): $0.005/hour
→ $0,0000835/run

External (Azure-IR): $0.00025/hour
→ $0,000008325/run

Data Movement (Azure-IR): $0,25/hour
→ 2 * 1.6667 * $0.25 = $0,83335/run

$0,834441825 / run
once every day: $25.03
once every hour: $600.80
once every 5 min: $7209.58

50% cost saving with one single config



Fabric uses CU consumption

- Only two types of activities:
  - Data movement – **1,5** CU hours
  - Data orchestration – 0,0056 CU hours per activity run
  - In docs, 1 CU hour = **$0,18**

- Copying 1TB of data to a warehouse
  - Takes about 11 minutes (662 seconds)
  - "intelligent throughput optimization" was 4
  - Utilized CU hours = 4 * 1,5 * (11/60) = 1,1 CU hours (or 3960 CU seconds)
  - 1,1 CU hours * ($0,18 / CU hours) = $0,20

- Check out the pricing examples (they are misleading though as they deal with only 1 copy activity)

ADF vs Fabric

- In ADF, duration matters for all activities
- In Fabric, duration only matters for copy data activity

- Cost of the same pipeline
  - ADF **= $0,83335 per run** (with default settings: **$1,66 per run**)
  - Fabric = 36000 CUs = 10 CU hour

Operation name	CU (s)	Duration (s)	Users	Billing type
DataMovement	36,000.0000	1,981.9620	1	Billable
ActivityRun	80.6400	178.2830	1	Billable
Total	**36,080.6400**	**2,160.2450**	**1**	**Billable**

=> 10 CU hour / (4 * 1,5) = 1,666667 hours (same as in ADF, thus 100 minutes)
=> 10 CU hour * ($0,18/CU hour) = **$1,8 per run**

- Fabric is about 8% more expensive (keep in mind you pay for the capacity running)

Tips & tricks

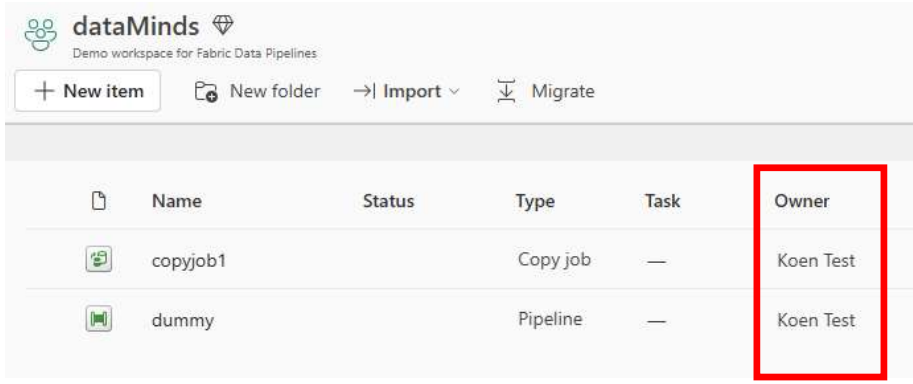
Ownership!!!

Manage Connections and Gateways

Connections | On-premises data gateways | Virtual network data gateways | Azure Key Vault references

Cloud and data gateway connections for artifacts. Learn more about supported connections.

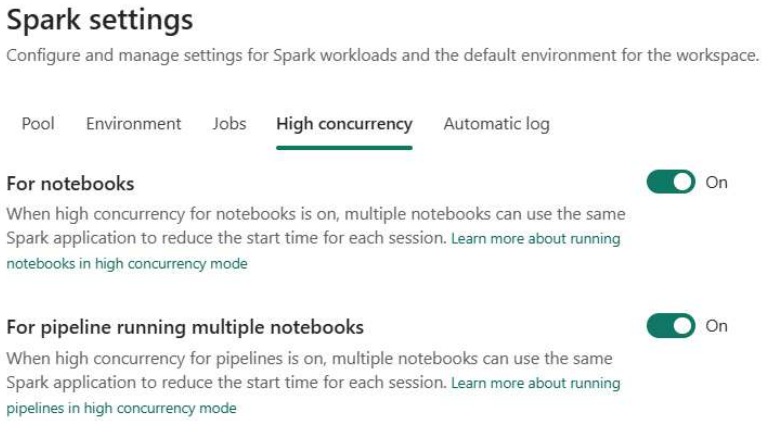
Name ↑	Connection type	Users
blobstorage koentest	Azure Blob Storage	Koen
CapacityMetricsCES	Capacity Metrics	Koen
eventhubs_fueltype koentest	EventHub	Koen
FabricSql koentest	Fabric SQL database	Koen
https://learn.microsoft.com/en-us/fabric/release-plan/	Web	Koen
https://raw.githubusercontent.com/MicrosoftLearning/dp-data/main/orders.csv	Web	Koen
koensql.database.windows.net;mycheapdwh	SQL Server	Koen
Lakehouse	Lakehouse	Koen



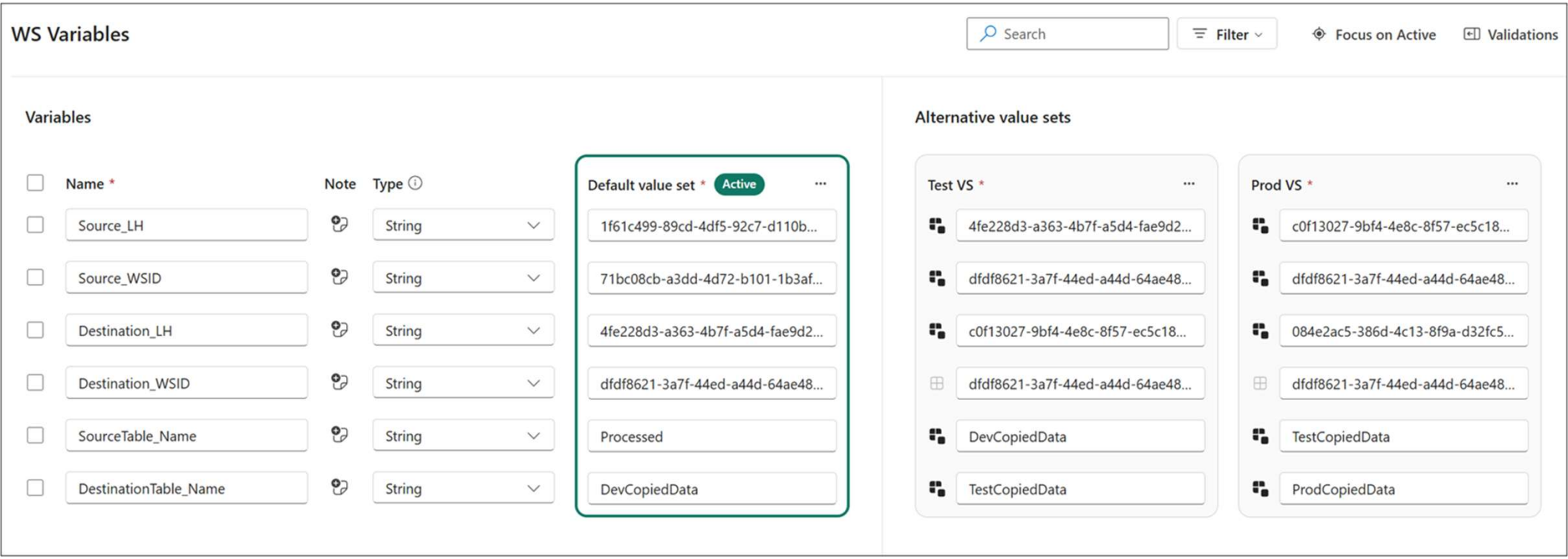
Capacity Consumption

- In general, the less code, the more CU usage ;)
- Follow the same guidelines as for ADF cost
  - Minimize (Copy Data) activities
  - Use Fabric Data Pipelines as an ELT orchestrator
  - Use shortcuts to minimize copying

- When you schedule notebooks, allow the session to be reused

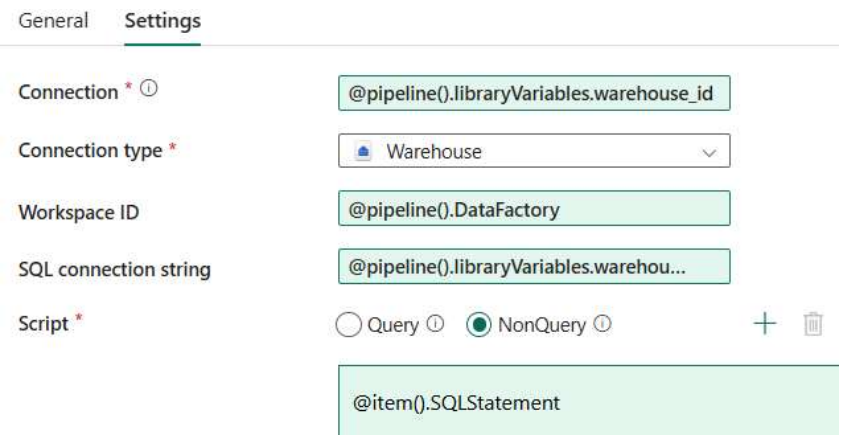


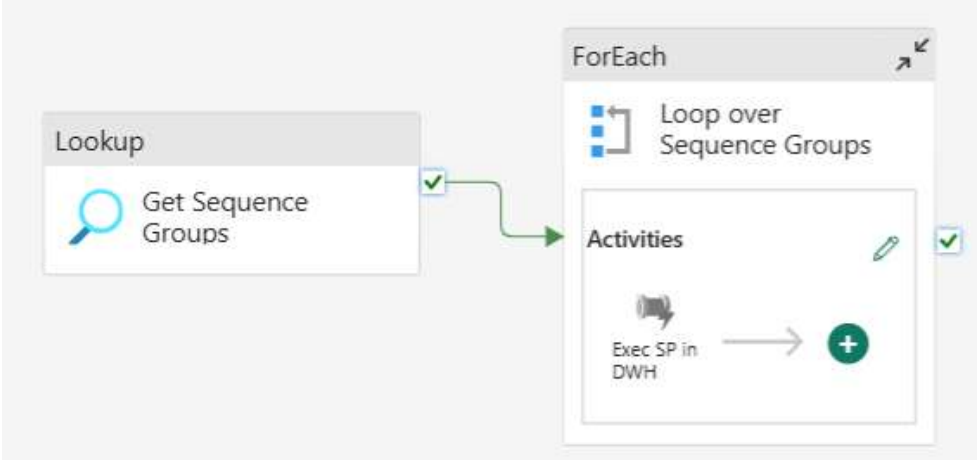
Library Variables



Metadata FTW

- Parameterize everything
- Build metadata-driven pipelines
  - Create a pipeline once, add metadata later





Resiliency

- Create resilient pipelines by implementing restartability & retries
  - Change the default time-out (no task should take 12 hours)
  - Prevent errors e.g. check for existence

- Build modular and **idempotent** pipelines
  - Don't build one monolithic pipeline that does it all
- Don't forget about monitoring and alerting

Various

- Use git and CI/CD, obviously (and multiple environments!)
- Use clear naming conventions and use folders to organize
- Use Key Vault if possible
- Use managed identities / workspace identity where possible

- Typically I don't parse JSON in a Copy Data activity

conclusion

Conclusion

- There's a big overlap between ADF and Fabric Pipelines
- Biggest differences:
  - how cost is calculated due to Fabric capacity
  - invoking child pipelines
  - CI/CD and parameterization
  - connection configuration

- Most ADF best practices & design patterns apply to Fabric Pipelines
- Use Fabric Pipelines as an ELT orchestrator
  - definitively not for streaming data

thank you