Ventagium
Data Consulting

Fabric Workshop

2025-02-28 Part 1

The Team

Roberto

Diego

Horner

Jesus

Fernando

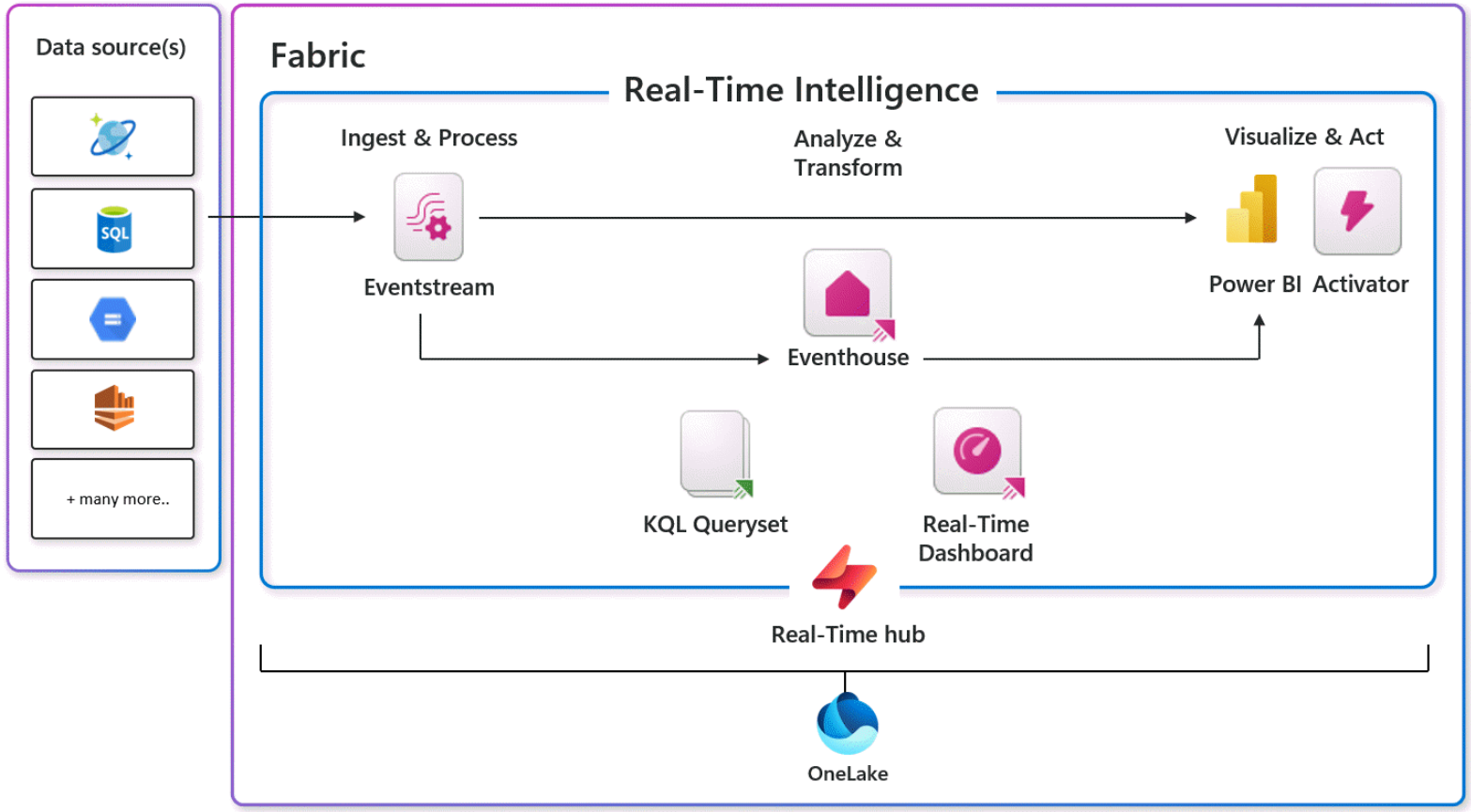
Arturo

Michelle

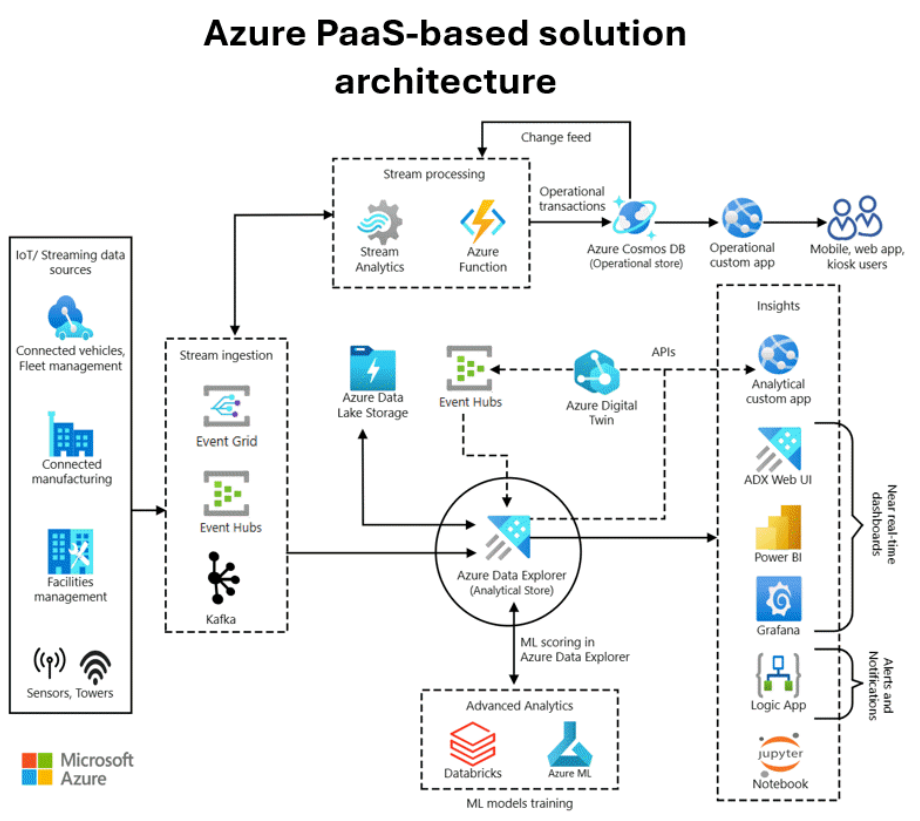
Real-Time Intelligence

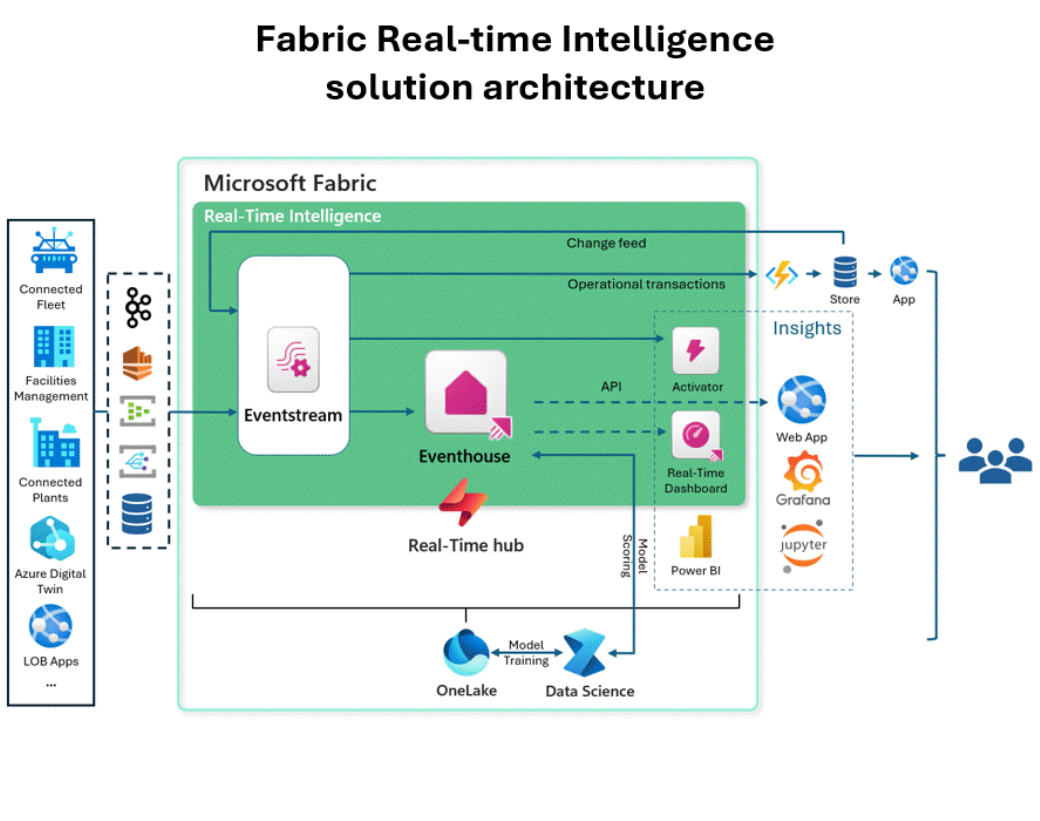
Real-Time Intelligence

- Comprehensive SaaS Offering
- Centralized Hub
- Rapid solution deployment
- Insights powered by real-time AI



Real-Time Intelligence





Real-time Intelligence: Components

- Real-time Hub
- Eventstreams
- Eventhouse
- Activator

Real-time Intelligence: Use Cases

- Manufacturing PLCs
- IoT Systems
- Fraud Detection
- Anomaly Detection

Real-time Intelligence: Use Cases

- Manufacturing PLCs
- IoT Systems
- Fraud Detection
- Anomaly Detection

Best Practices

- **Be use-case driven:** not everything needs to be real-time.
- **Automate decisions:** don’t expect someone to be reacting all of the time.
- Don’t use Fabric as an Operational Platform

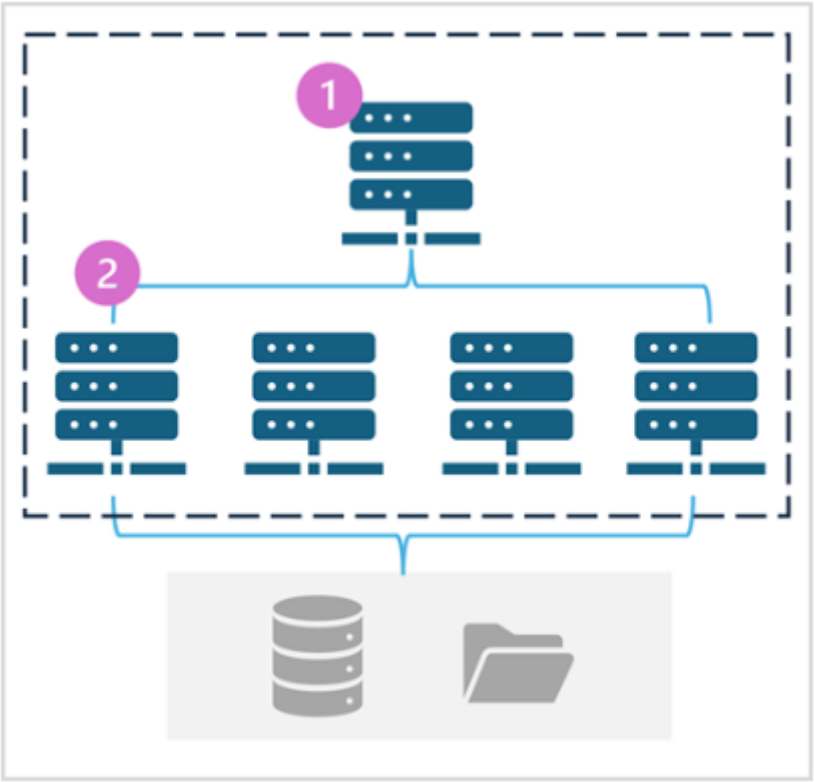
End-to-end Demo

Spark and Fabric Notebooks

1. What is Spark?
2. What are Fabric Notebooks?
3. Getting started…
4. Exercise: Demo and Tutorial

What is Spark?

- Unified analytics engine for large-scale data processing.
- Used for batch and stream processing.
- Able to be run on single machine or multiple nodes.
  - Coordinates work across multiple processing nodes in a cluster, known in Microsoft Fabric as a **Spark pool**.



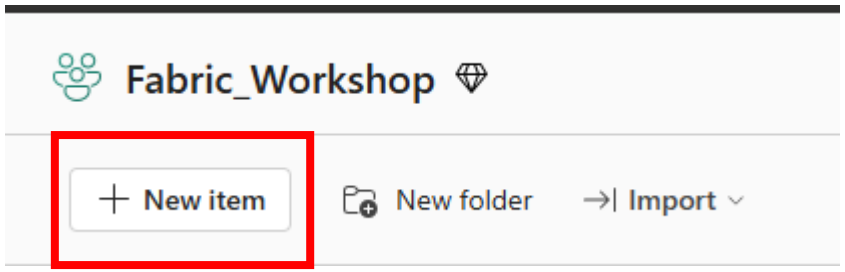
1. **Head node** coordinates via **driver**.
2. **Worker nodes** complete tasks via **executors**.

What are Fabric Notebooks?

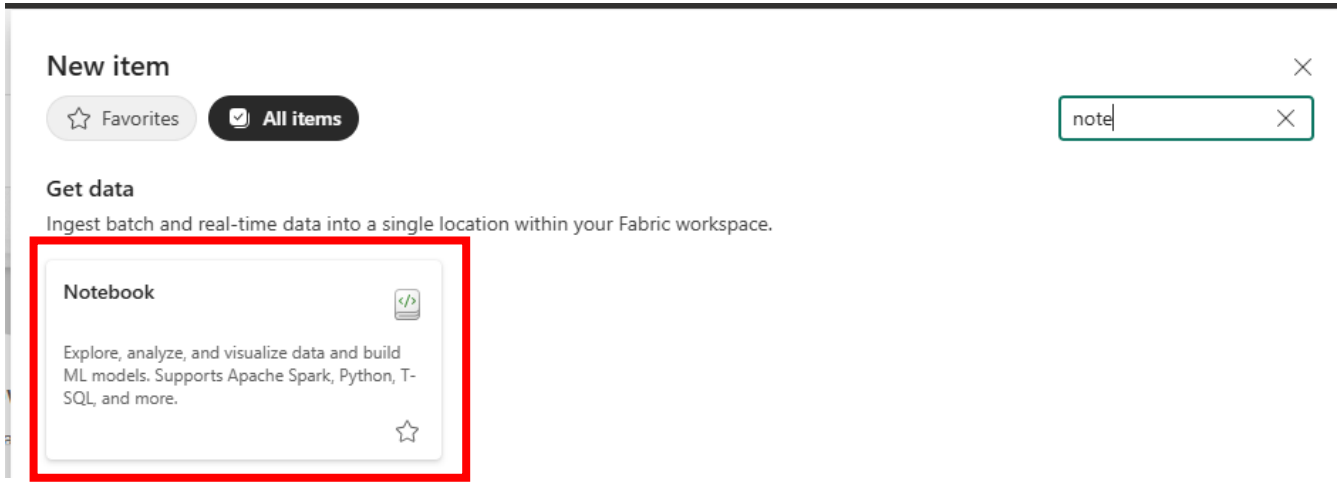
- Primary programming experience in Microsoft Fabric.
- Multiple programming languages supported.
  - Spark – Scala, PySpark, SparkR, Spark SQL
  - Python (Preview)
  - T-SQL
- Applications
  - ETL
  - EDA
  - Machine Learning

Getting started in 2 steps…

1. Click on the "**+ New item**" button.



2. Click on **Notebook** item.



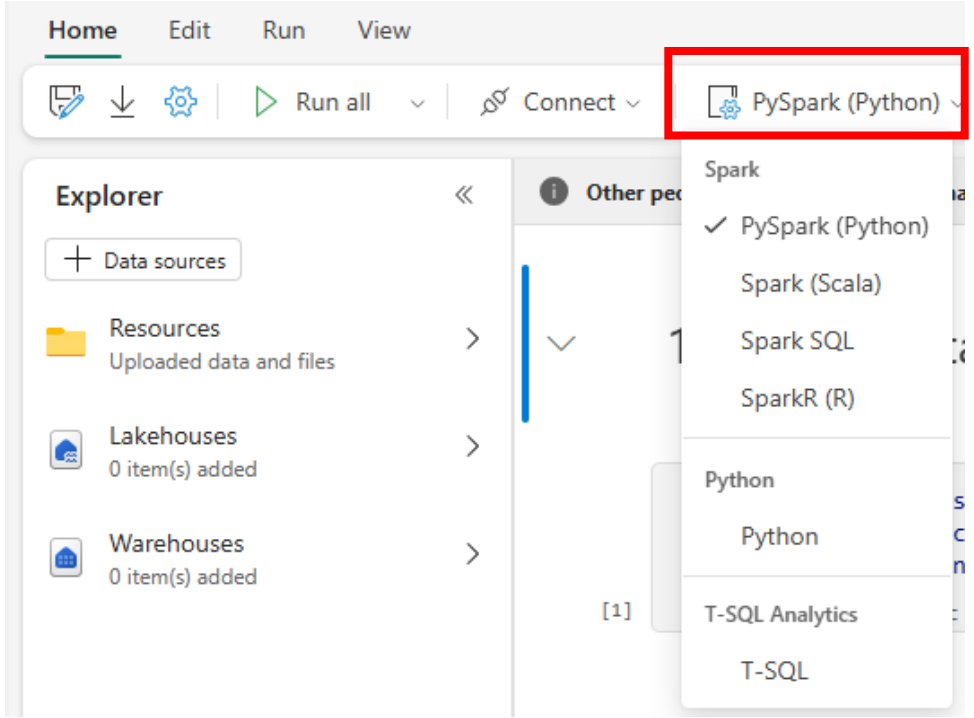
Exercise: Notebooks Demo and Tutorial

Fabric Notebooks Advanced Features (Speedrun)

1. Programming Languages
2. Parameter Cells
3. Frozen Cells
4. Notebook Resources
5. High Concurrency
6. Spark Starter Pools
7. Spark Custom Pools
8. Managing Libraries
9. Environments
10. Manage Notebooks with APIs
11. Code Snippets

Different Programming Languages

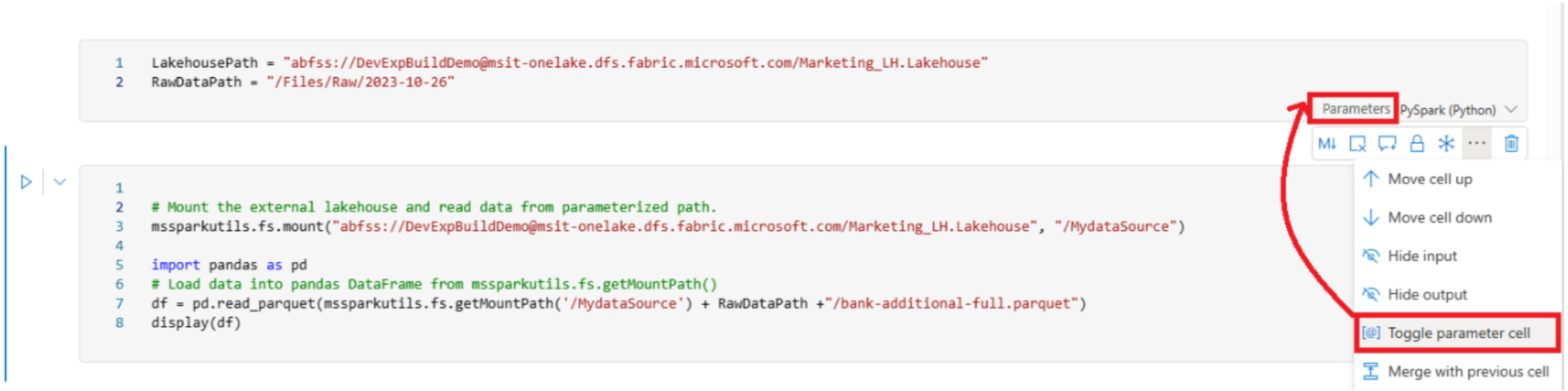
- Set for entire notebook for single cell.



Magic command	Language	Description
%%pyspark	Python	Execute a **Python** query against Apache Spark Context.
%%spark	Scala	Execute a **Scala** query against Apache Spark Context.
%%sql	SparkSQL	Execute a **SparkSQL** query against Apache Spark Context.
%%html	Html	Execute a **HTML** query against Apache Spark Context.
%%sparkr	R	Execute a **R** query against Apache Spark Context.

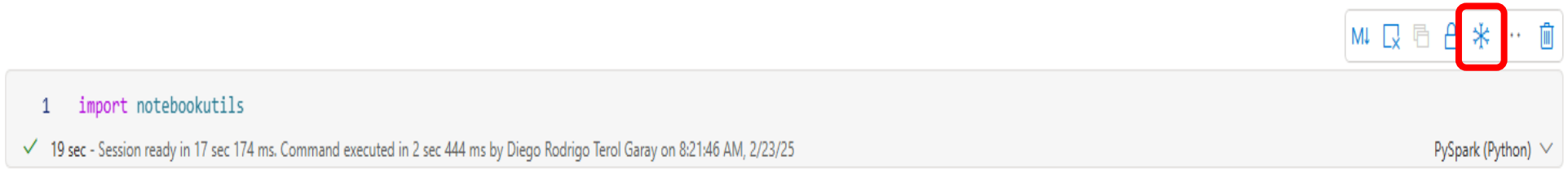
Parameter Cells

- Pass arguments via runMultiple or Pipeline.



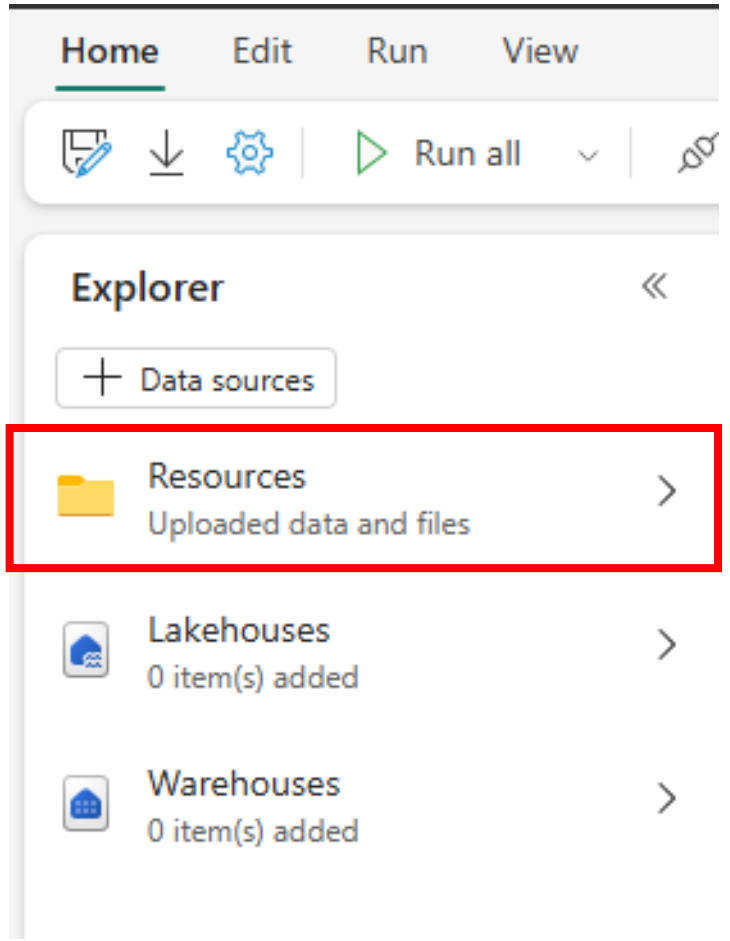
Frozen Cells

- Make cells read-only.



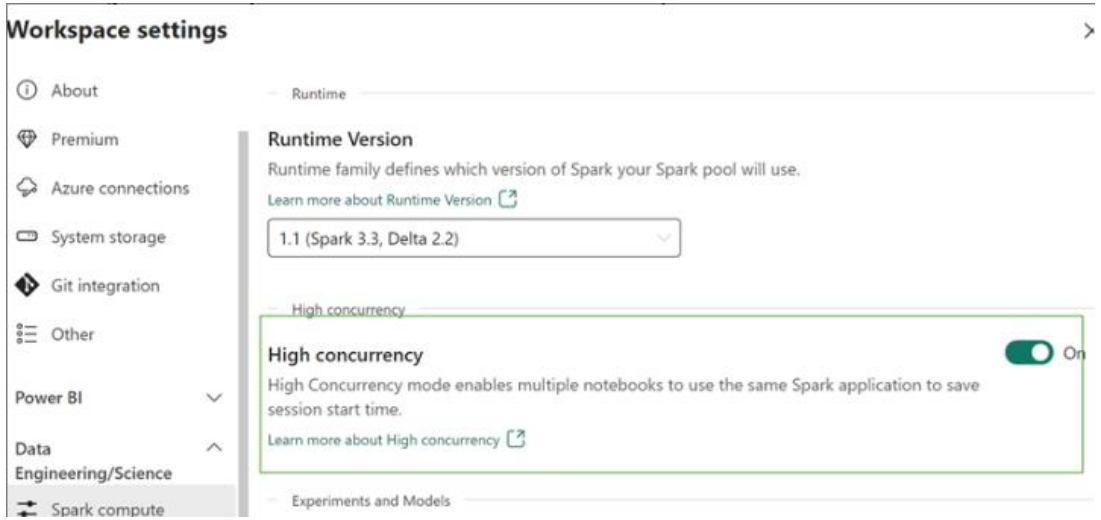
Notebook Resources

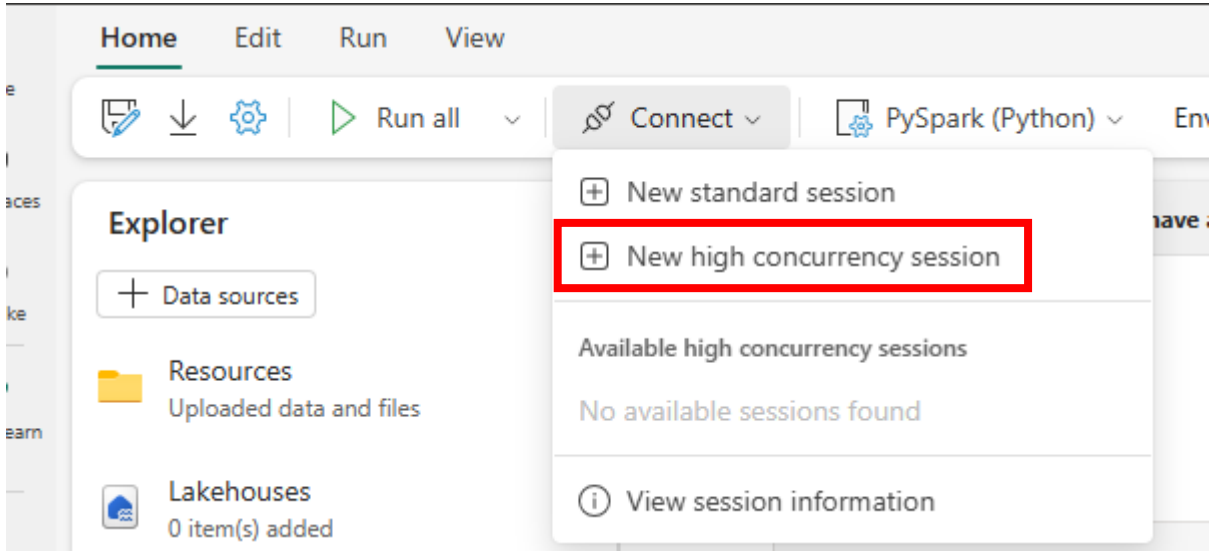
- File system space to store small-sized files (e.g., code modules, semantic models, images…).



High Concurrency

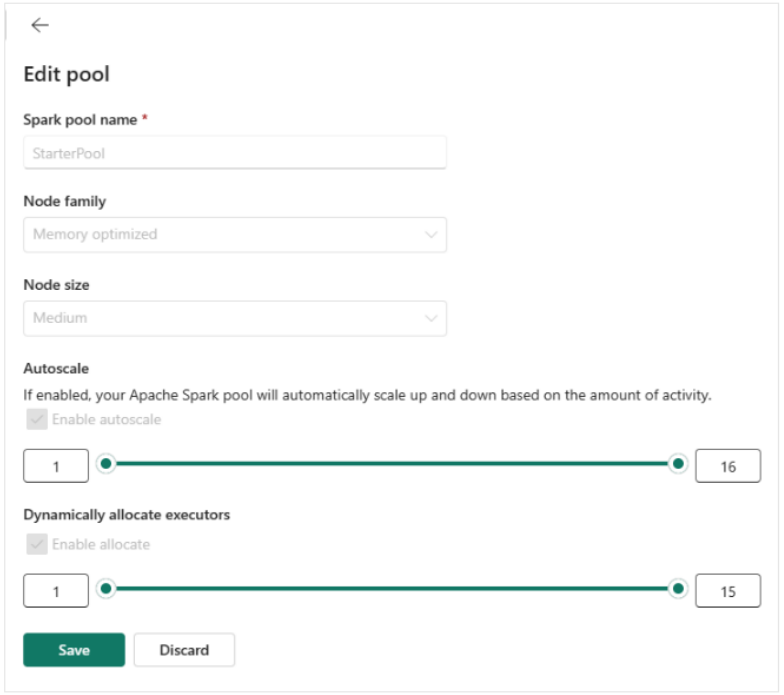
- Share Spark Sessions across multiple Notebooks.





Configure Spark Starter Pools

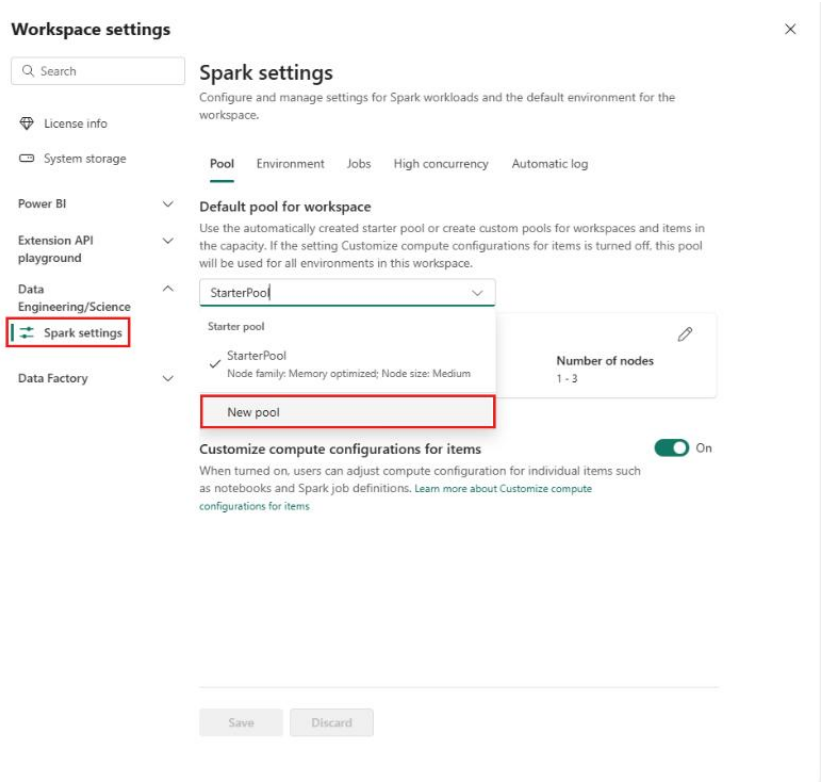
- Configure Starter Pool based on Capacity size and required resources.

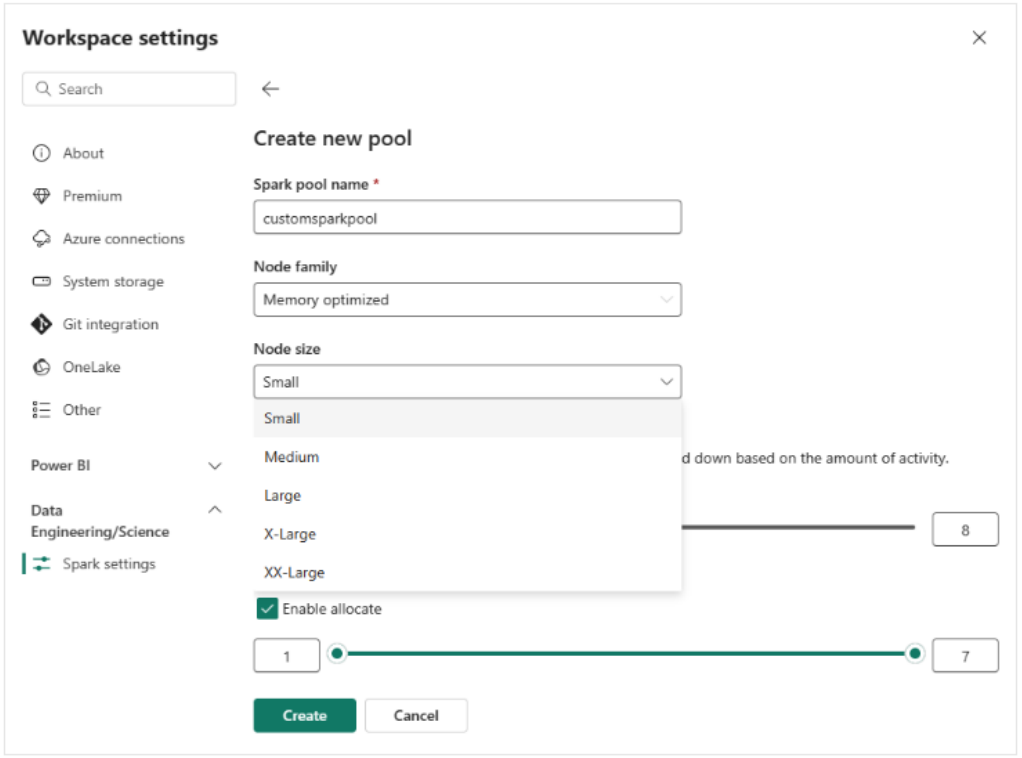


SKU name	Capacity units	Spark VCores	Node size	Default max nodes	Max number of nodes
F2	2	4	Medium	1	1
F4	4	8	Medium	1	1
F8	8	16	Medium	2	2
F16	16	32	Medium	3	4
F32	32	64	Medium	8	8
F64	64	128	Medium	10	16
(Trial Capacity)	64	128	Medium	10	16
F128	128	256	Medium	10	32
F256	256	512	Medium	10	64
F512	512	1024	Medium	10	128
F1024	1024	2048	Medium	10	200
F2048	2048	4096	Medium	10	200

Work with Custom Spark Pools

- Create and configure based on required workloads.





Managing Libraries

- Available via Inline installation and Environment setup.

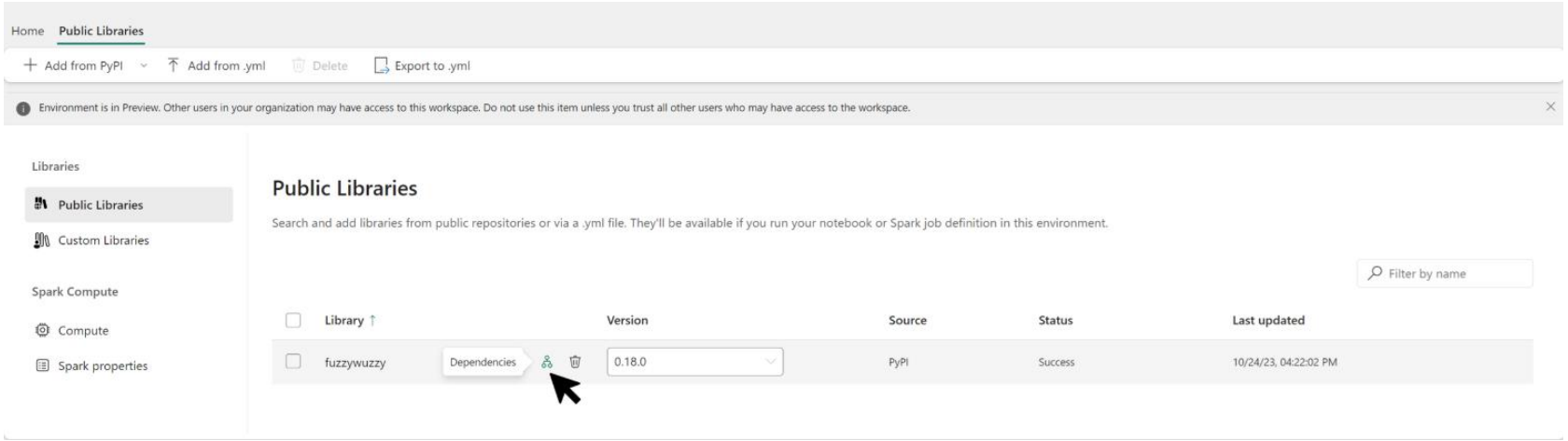
Python Copy

```
%conda install altair          # install latest version through conda command
%conda install vega_datasets   # install latest version through conda command
```

Library type	Environment library management	Inline installation
Python Public (PyPI & Conda)	Supported	Supported
Python Custom (.whl)	Supported	Supported
R Public (CRAN)	Not supported	Supported
R custom (.tar.gz)	Supported as custom library	Supported
Jar	Supported as custom library	Supported

Environments

- Configure Spark compute.
- Manage public and custom Libraries.
- Manage resources.



Manage Notebooks with APIs

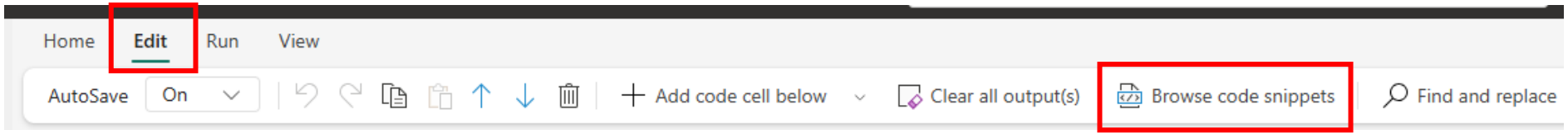
- Manage CRUD operations and Notebook Run instances.

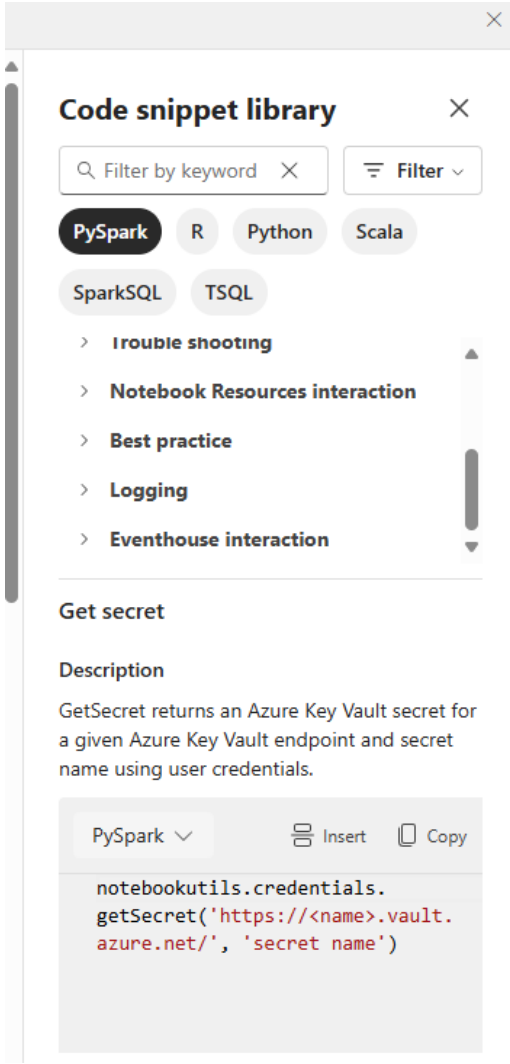
Action	Description
Create item	Creates a notebook inside a workspace.
Update item	Updates the metadata of a notebook.
Update item definition	Updates the content of a notebook.
Delete item	Deletes a notebook.
Get item	Gets the metadata of a notebook.
Get item definition	Gets the content of a notebook.
List item	List all items in a workspace.

Action	Description
Run on demand Item Job	Run notebook with parameterization.
Cancel Item Job Instance	Cancel notebook job run.
Get Item Job Instance	Get notebook run status.

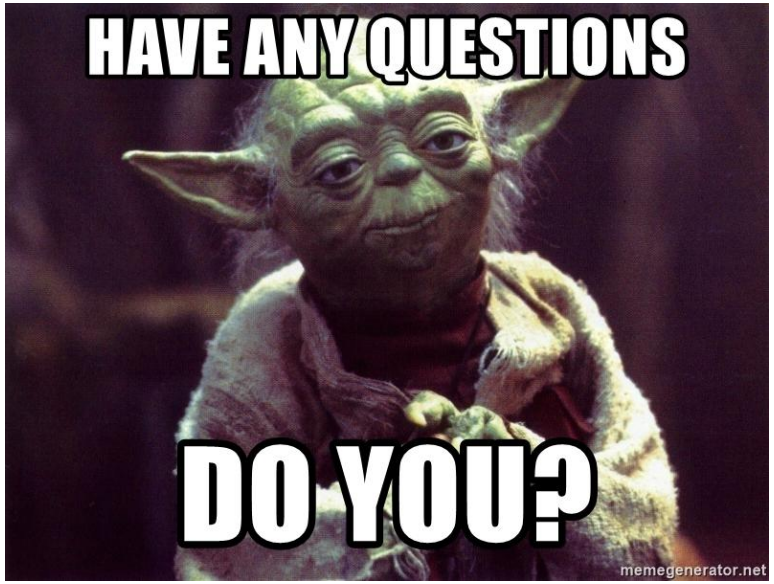
Code Snippets

- Get commonly used code patterns.





Questions?



Power BI Report with Copilot

Power BI

Business Intelligence tool:

- Provide data visibility
- Enhance data-driven decision making.

Formed by

- Semantic Model (Import Mode, Direct Query, Direct Lake)
- Reports

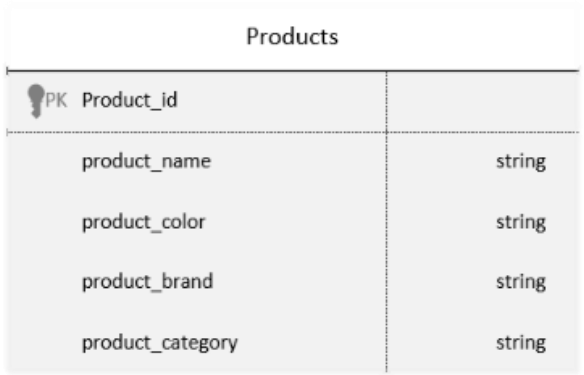
Not Dataset! It is a Semantic Model

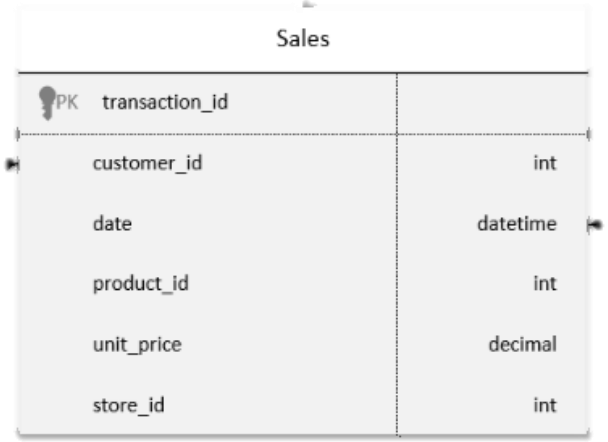
Give meaning to your data!

- Store Data.
- Defines relationships and connection among your tables.
- Define the measures/calculations your model will support.
- Define display formatting of your calculations or columns.

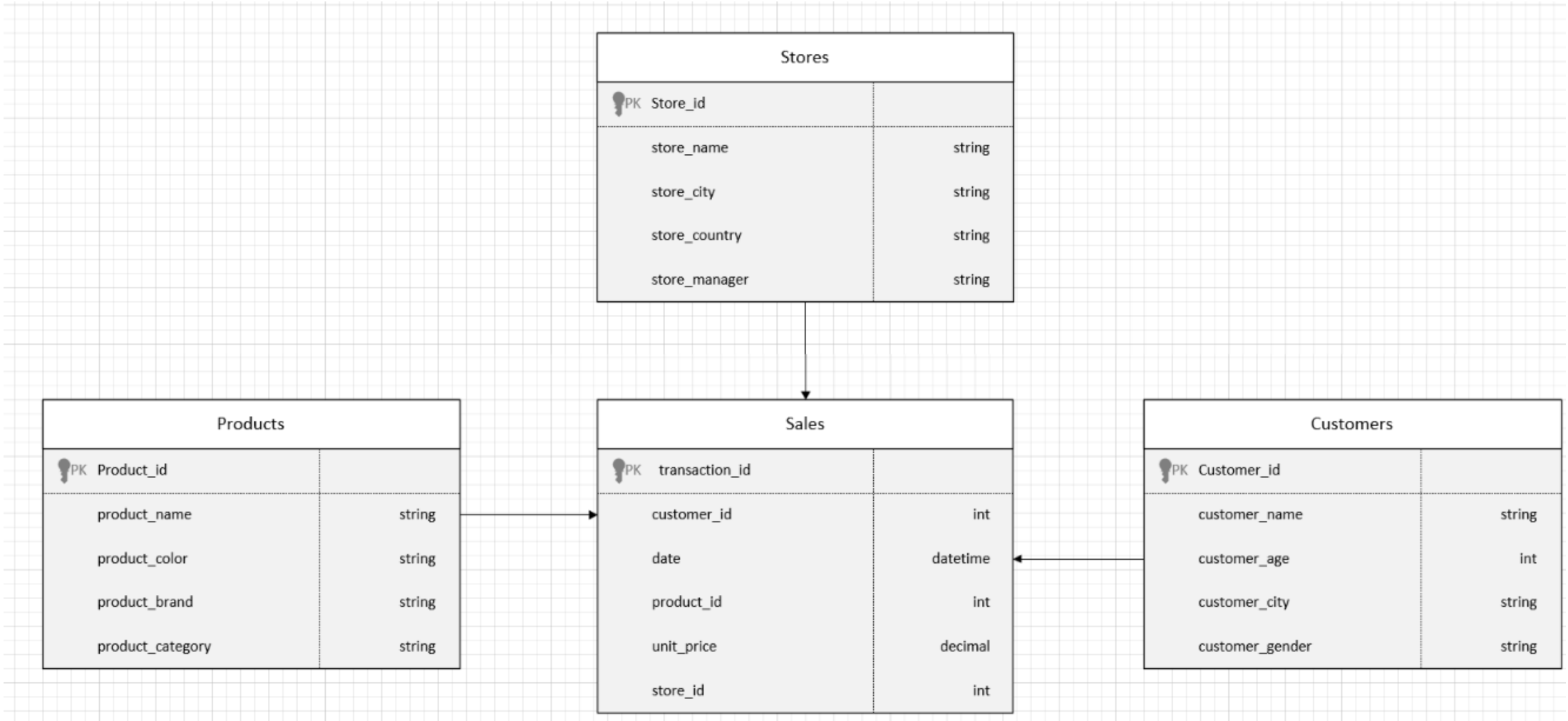
You define your business logic!


Stores
PK Store_id
store_name string
store_city string
store_country string
store_manager string


Products
PK Product_id
product_name string
product_color string
product_brand string
product_category string

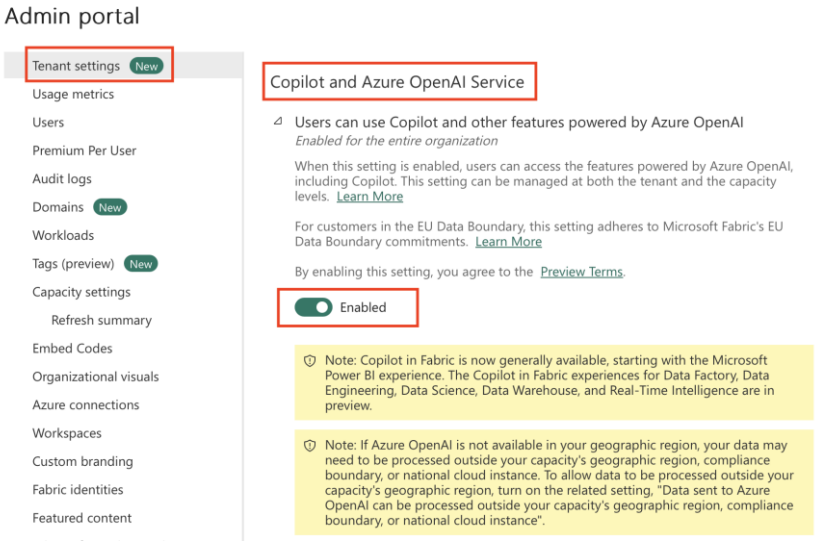

Sales
PK transaction_id
customer_id int
date datetime
product_id int
unit_price decimal
store_id int


Customers
PK Customer_id
customer_name string
customer_age int
customer_city string
customer_gender string


Stores
PK Store_id
store_name string
store_city string
store_country string
store_manager string
Products
PK Product_id
product_name string
product_color string
product_brand string
product_category string
Sales
PK transaction_id
customer_id int
date datetime
product_id int
unit_price decimal
store_id int
Customers
PK Customer_id
customer_name string
customer_age int
customer_city string
customer_gender string

Copilot Requirement

1. Enable Copilot in Microsoft Fabric.
2. F64 or higher FSKU
3. Copilot in Microsoft Fabric isn't supported on trial SKUs. Only paid SKUs (F64 or higher, or P1 or higher) are supported.



Exercise Direct Lake

1. Make sure the following tables are in your gold lakehouse
   - Customers
   - Products
   - SalesOrderHeader
   - SalesOrderDetail
2. Create a Semantic Model Item in Fabric
3. Open Semantic Model and Create Relationships
4. Use “Give Me Executive Summary”
5. Start with the following Prompt and improve it. Make 5 iterations to the prompt. Make Final adjustments

“Create a Sales Report”

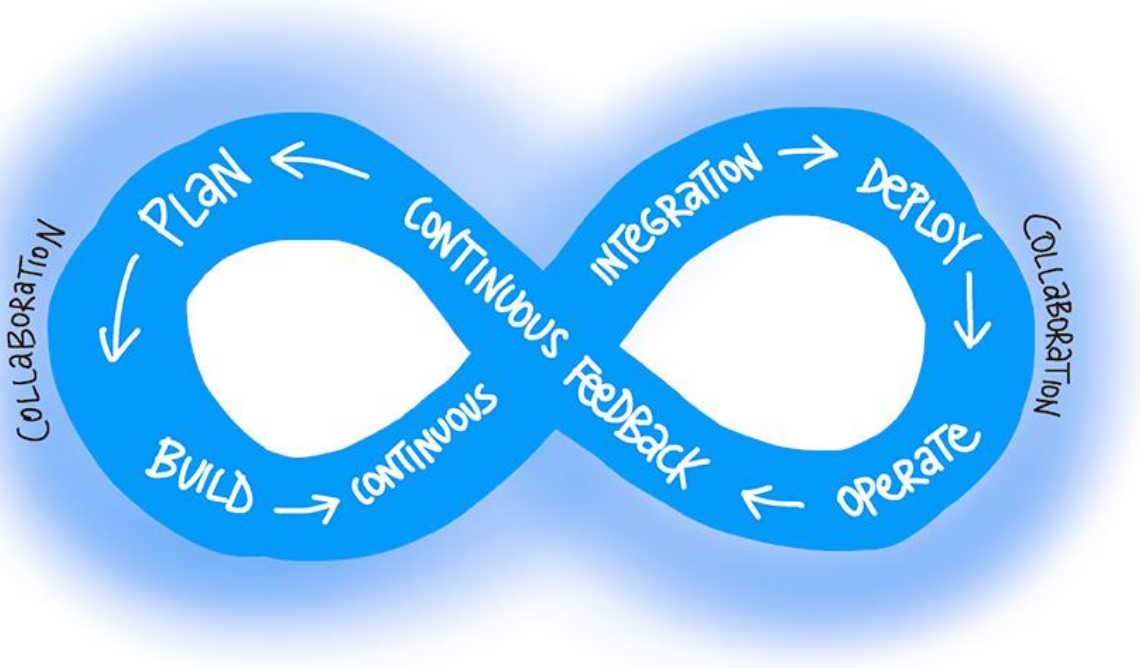
Exercise Import Mode

1. Make sure the following tables are in your gold lakehouse
   - Customers
   - Products
   - SalesOrderHeader
   - SalesOrderDetail
2. Open Power BI Desktop.
3. Make sure you are logged in on the correct account.
4. Import Data

CI with Azure DevOps and Git Integration

What is DevOps?

- [What is DevOps? - Training | Microsoft Learn]



DevOps benefits for Fabric

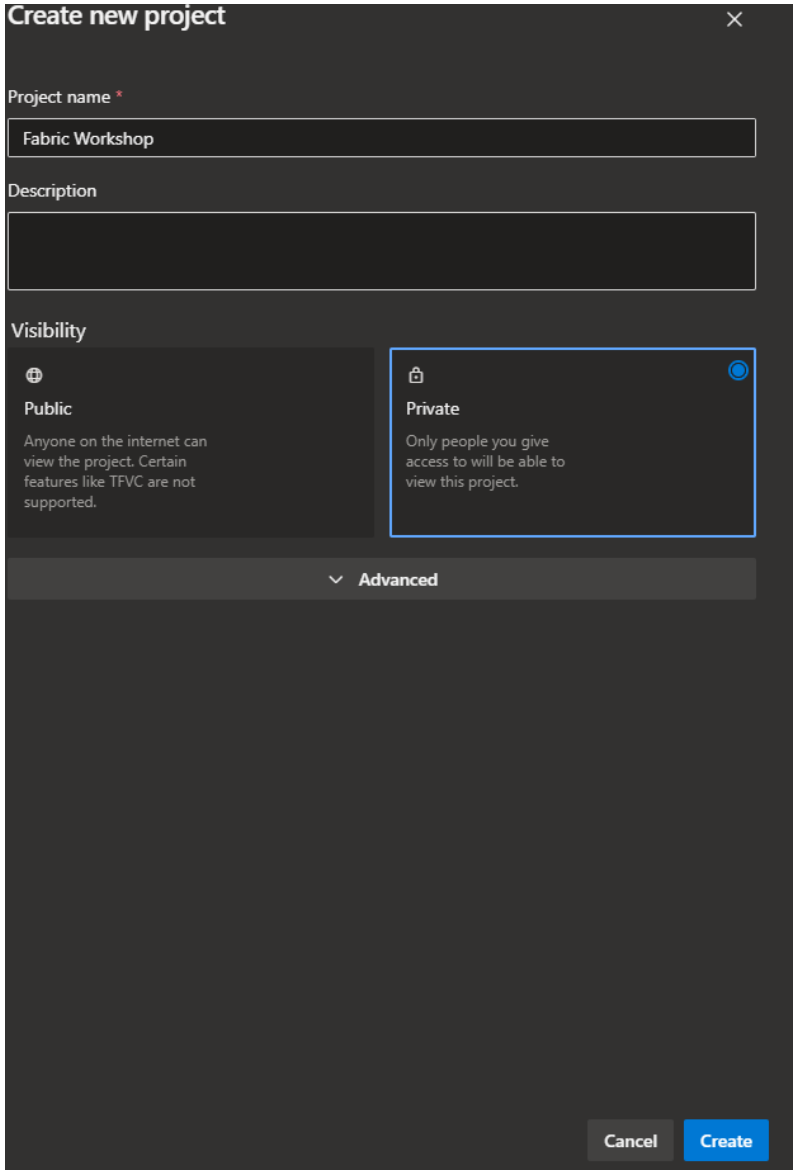
Enhance collaboration (CI)

Leverage your development (CD)

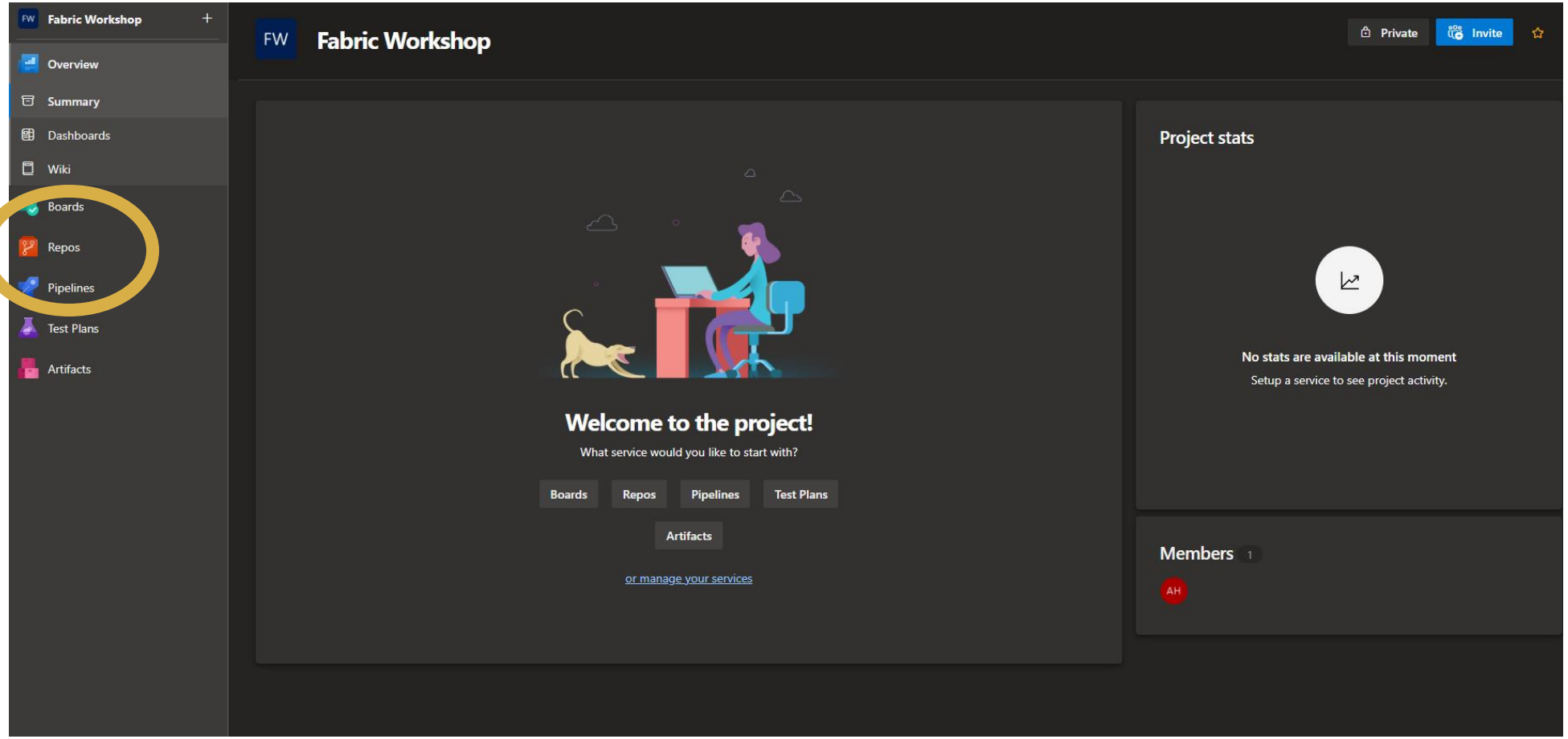
Version control

Create a DevOps Project

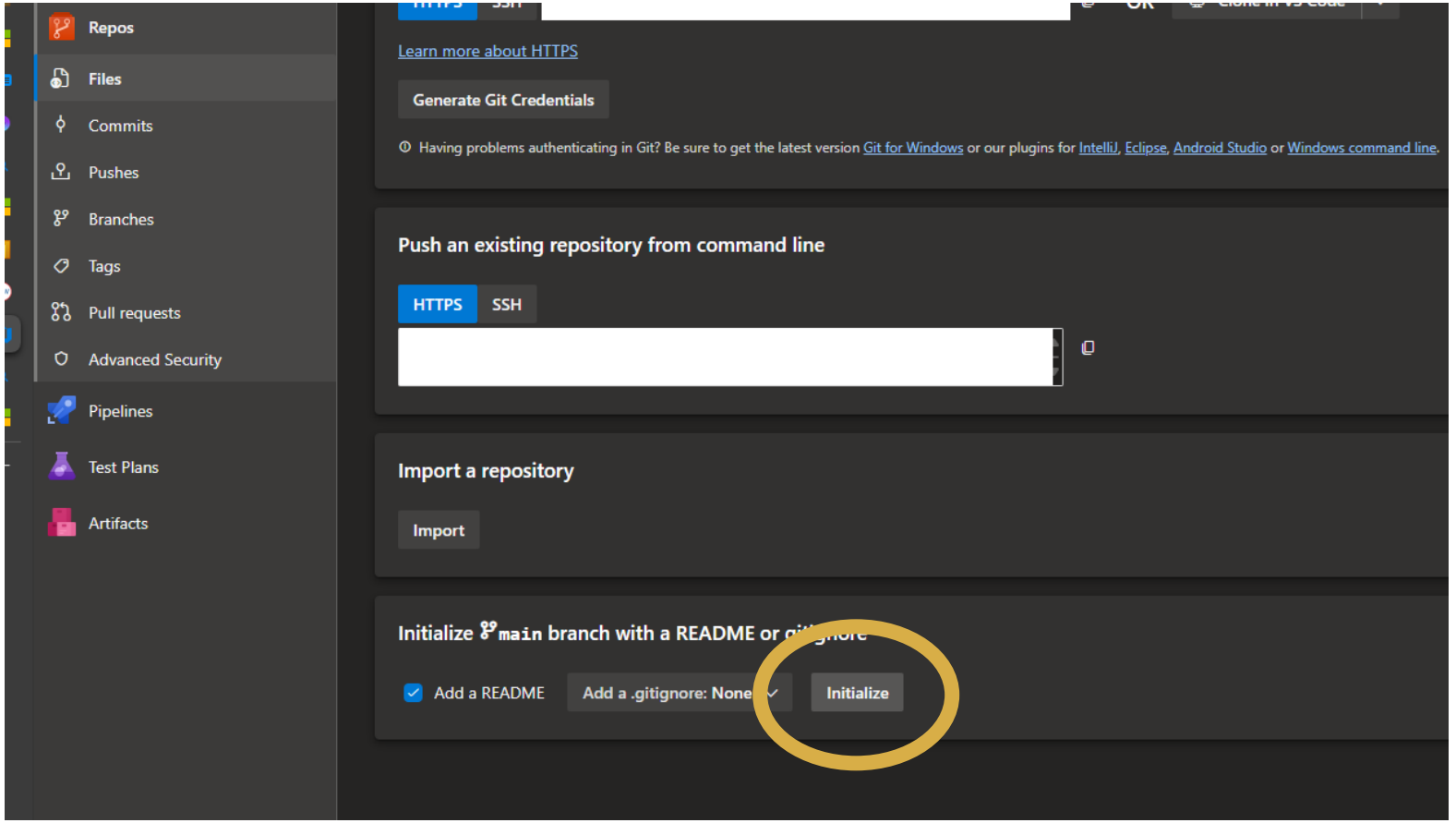
Create a DevOps Project



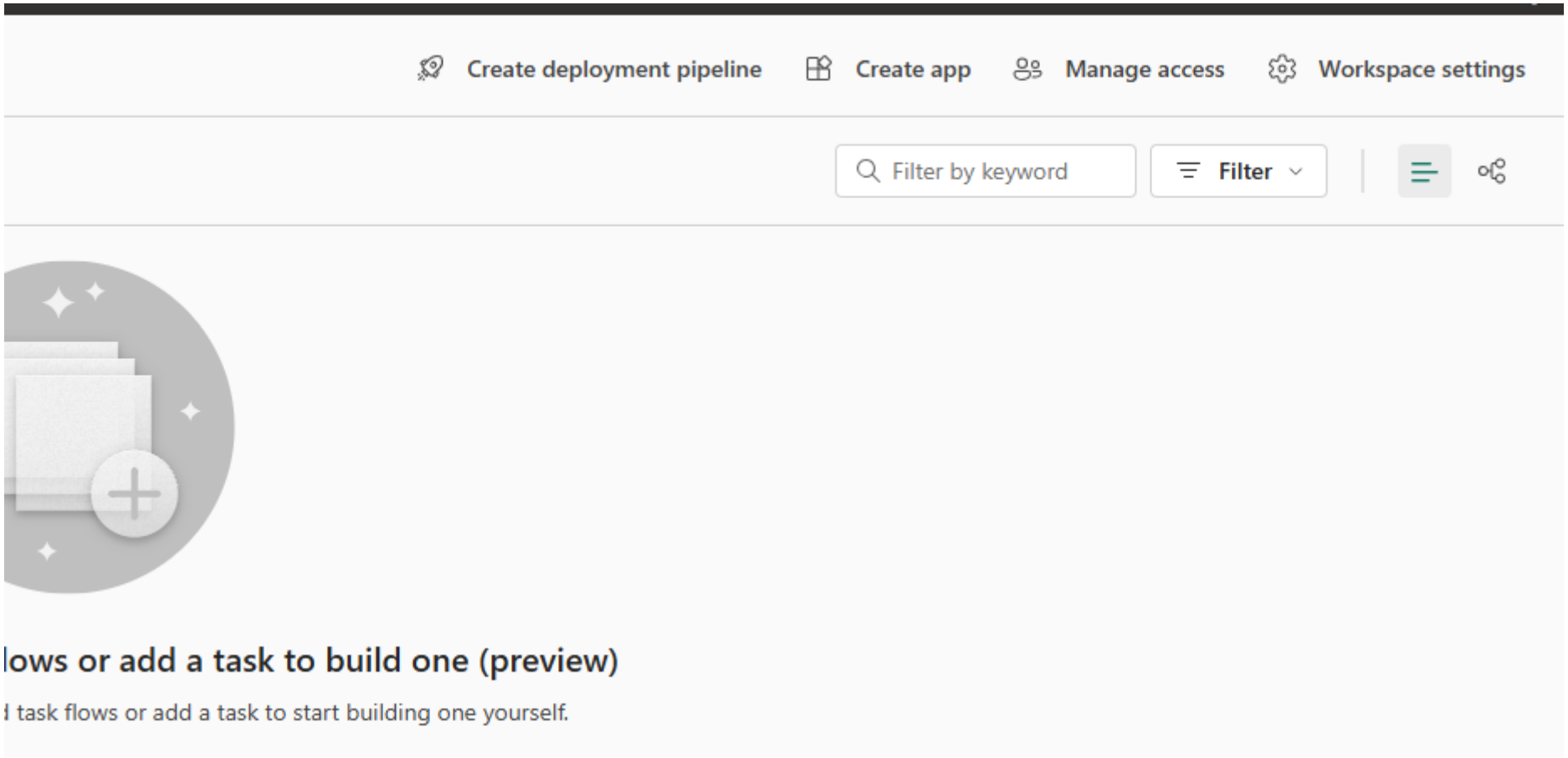
Initialize a Repo



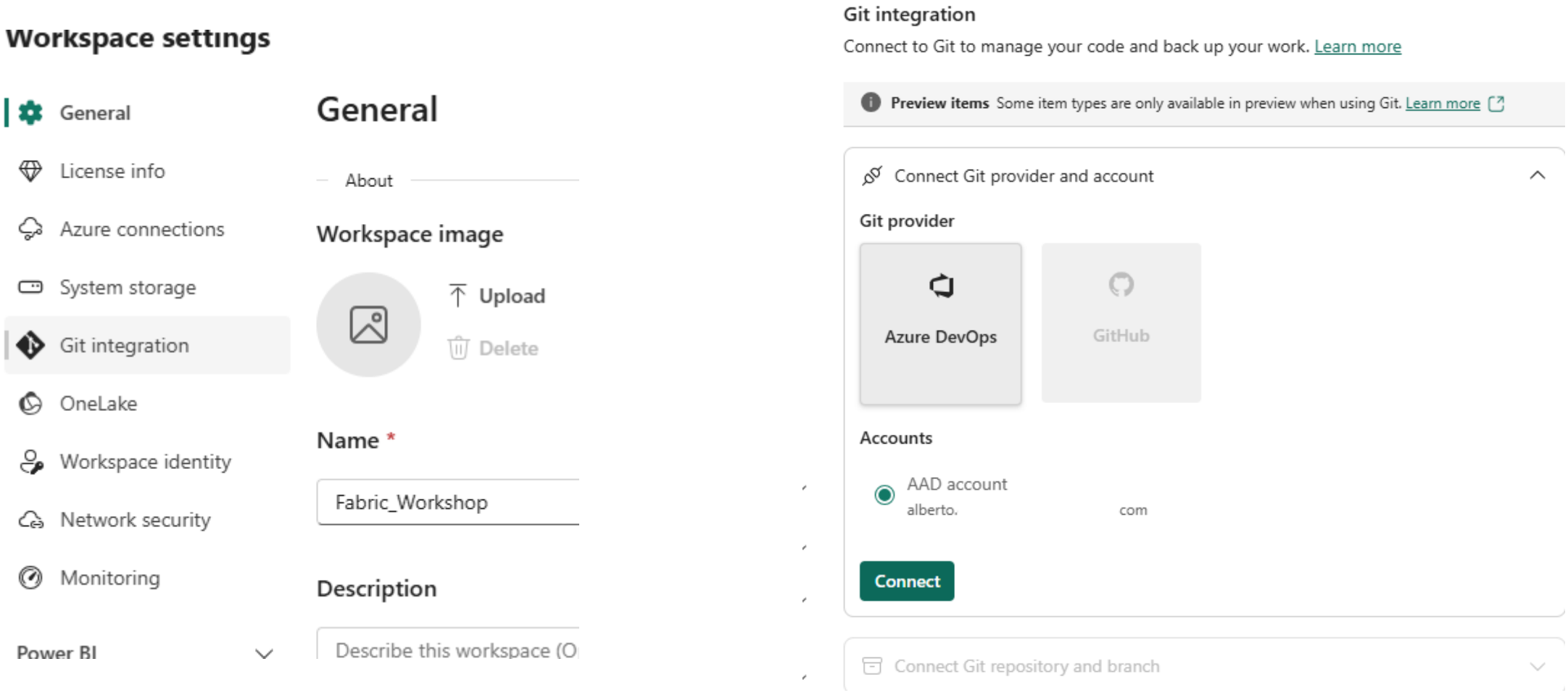
Initialize a Repo



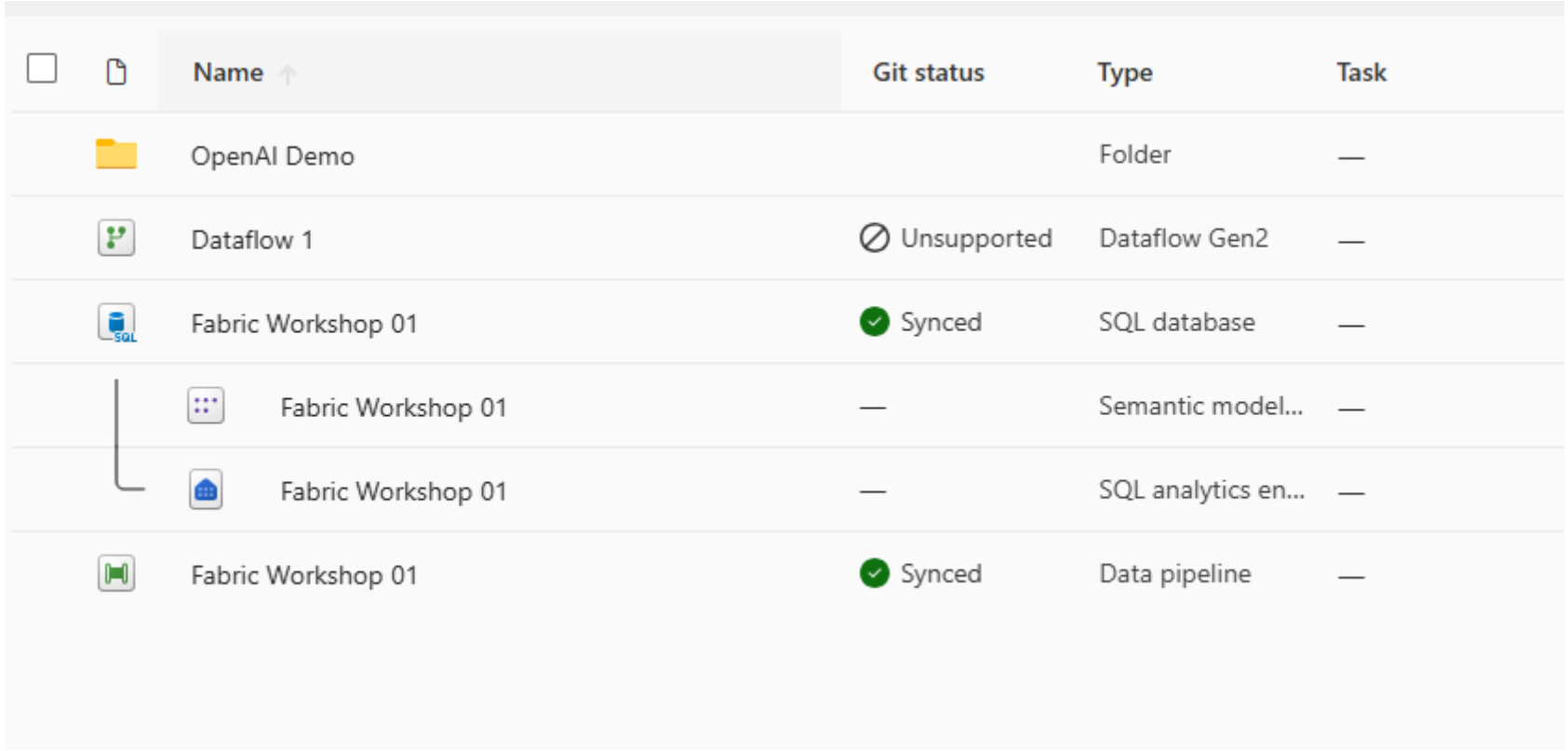
Enable Git Integration



Enable Git Integration



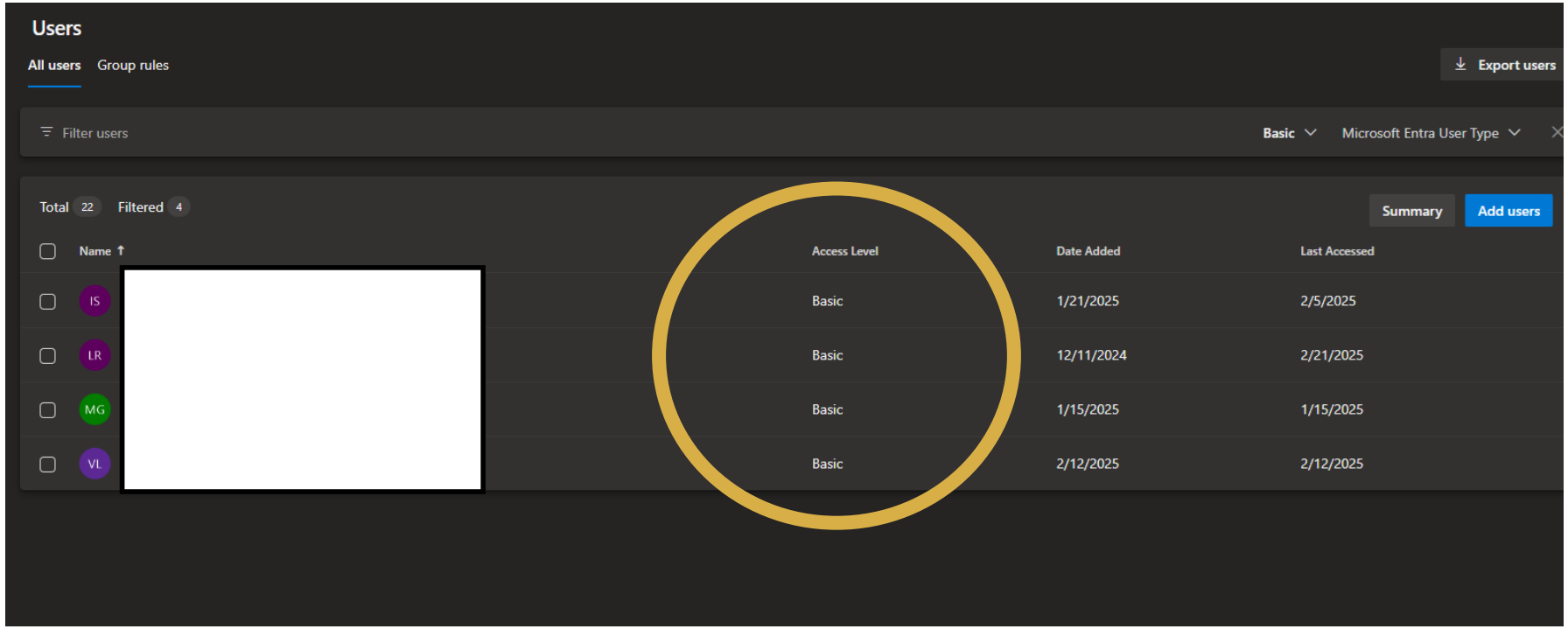
Enable Git Integration



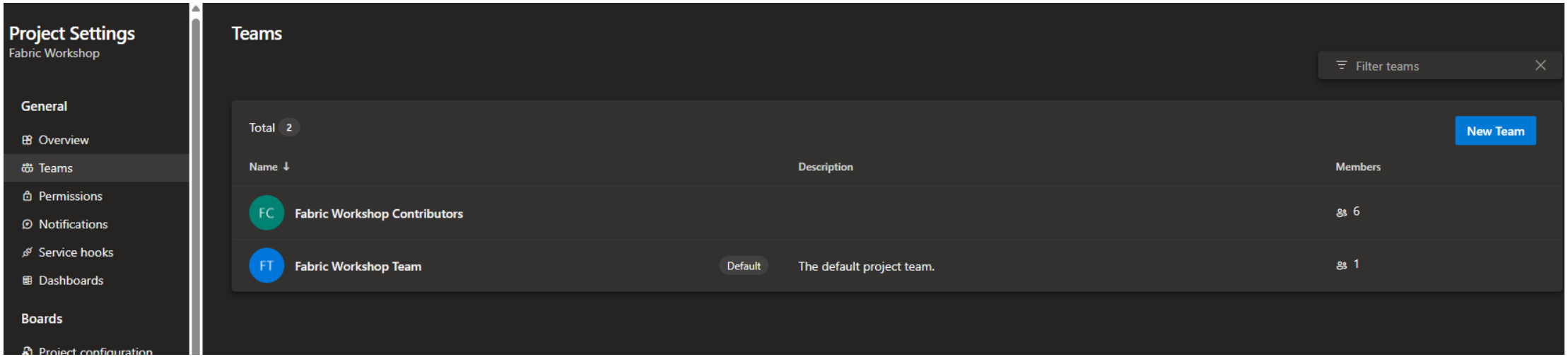
Name	Git status	Type	Task
OpenAI Demo		Folder	—
Dataflow 1	Unsupported	Dataflow Gen2	—
Fabric Workshop 01	Synced	SQL database	—
Fabric Workshop 01	—	Semantic model...	—
Fabric Workshop 01	—	SQL analytics en...	—
Fabric Workshop 01	Synced	Data pipeline	—

Create a Contributors Team

(Users require Basic access at least)

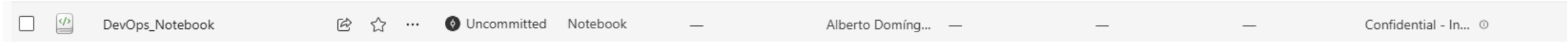


Create Contributors Team

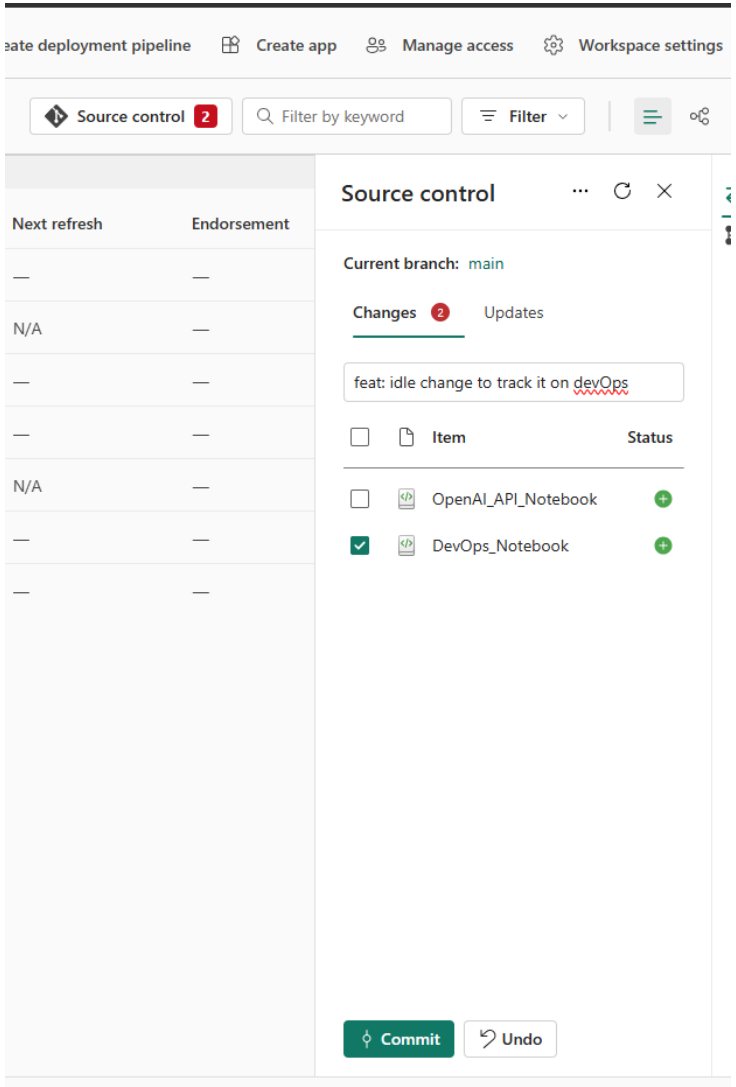


Commit and Sync your Items

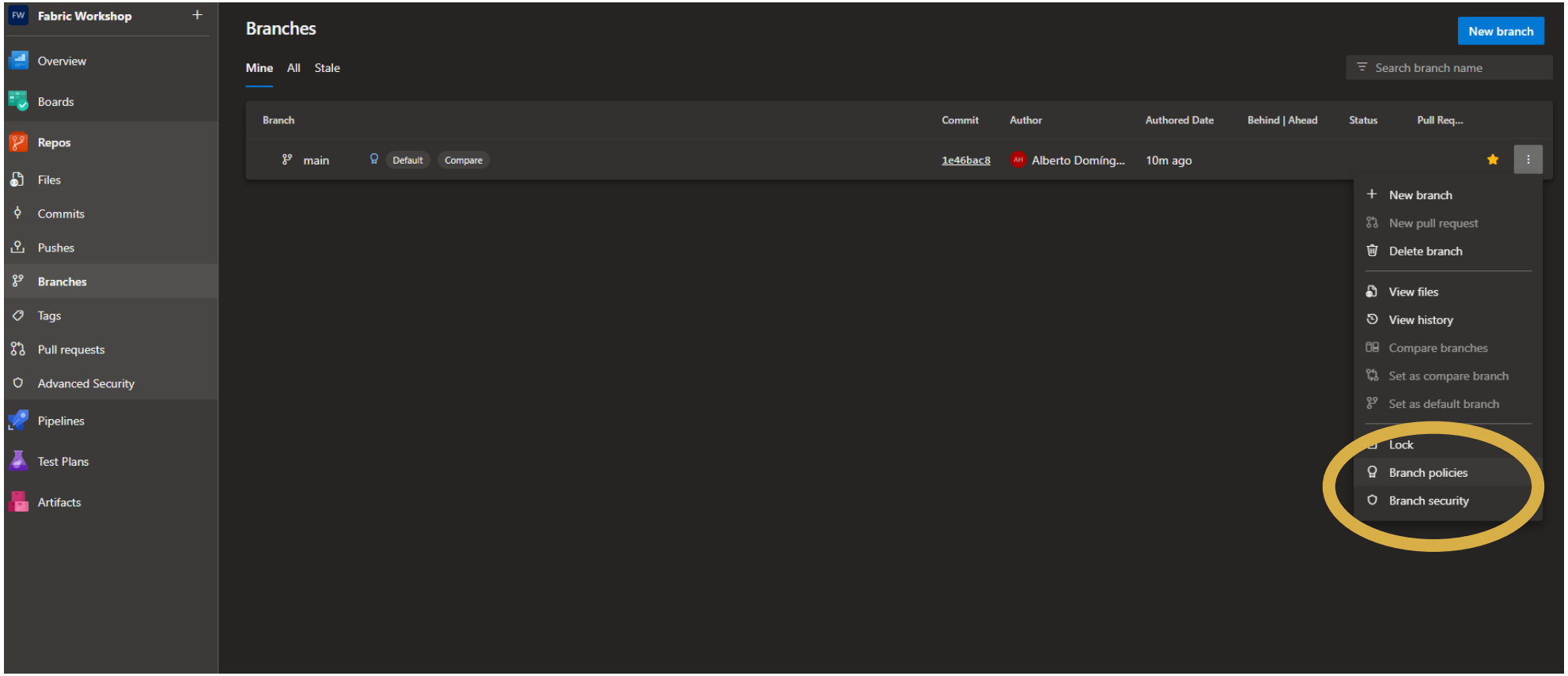
When someone edits an item, it will appear as uncommited:



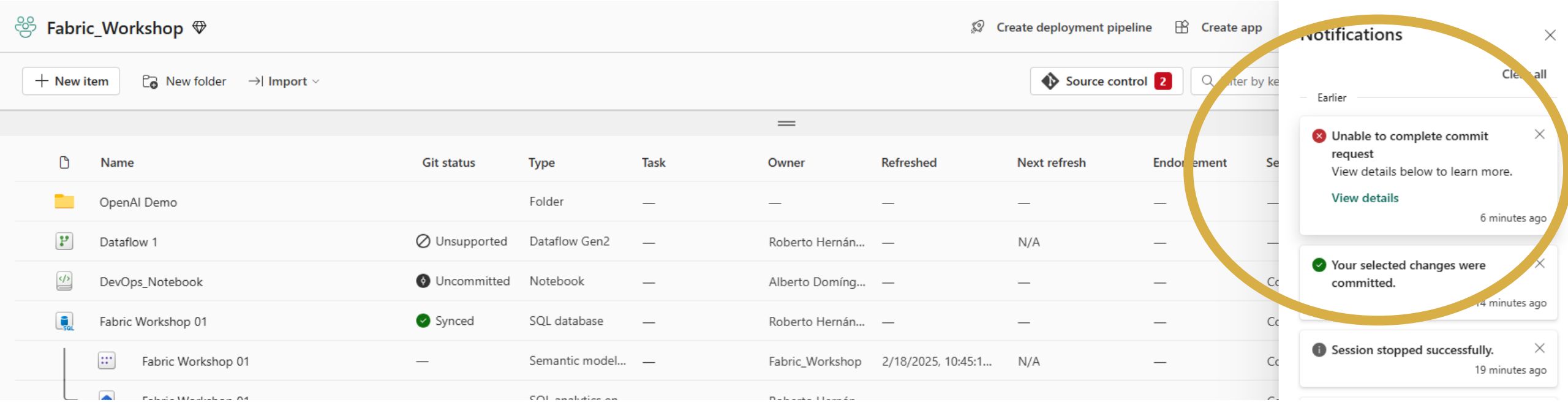
Commit and Sync your Items



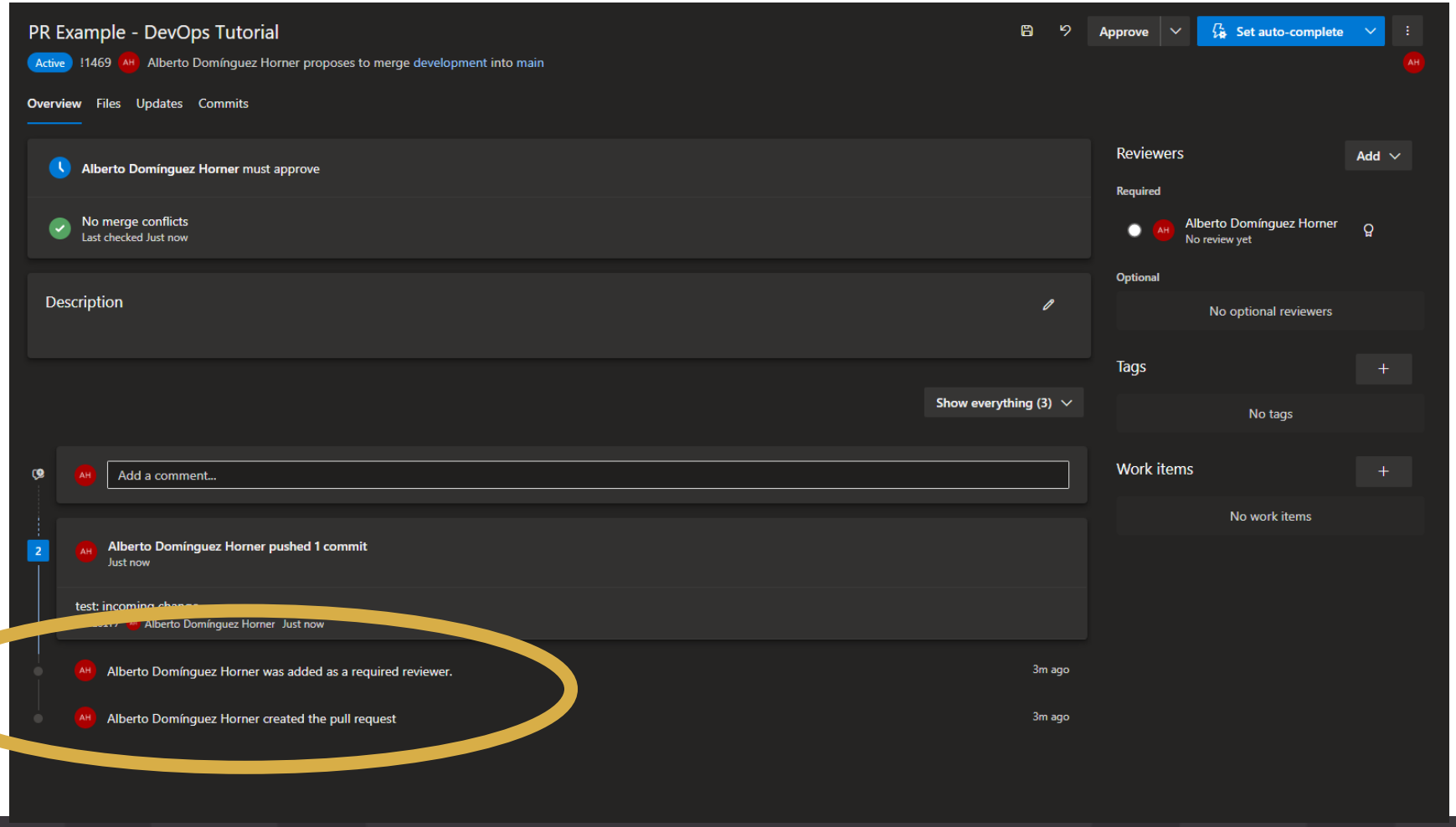
Create Branch policies



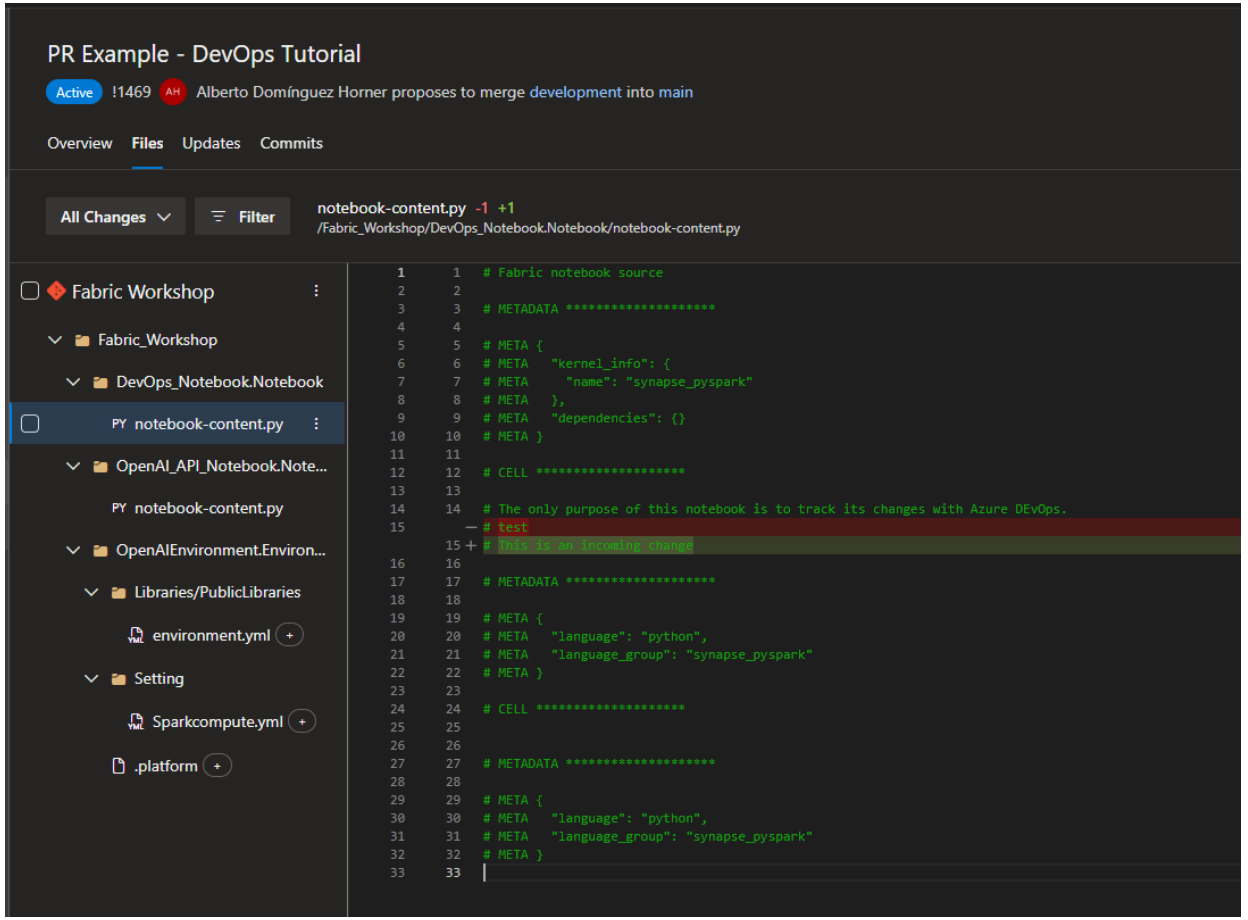
Branch Policies



Branch Policies



Continuous Integration by Pull Requests



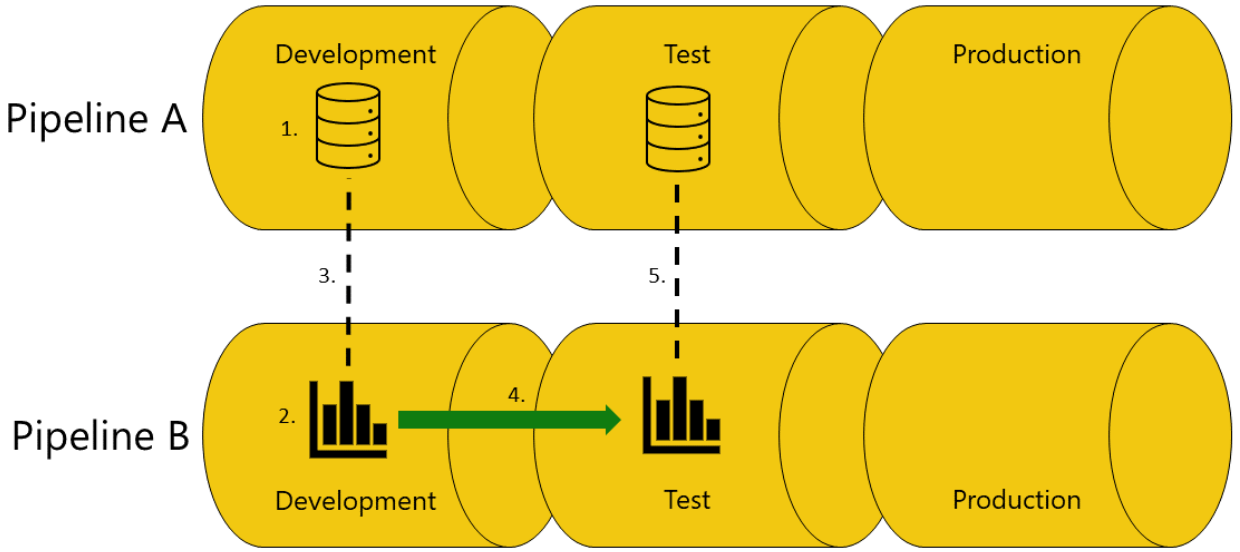
CD with Deployment Pipelines

Agenda

- Introduction to the CD Process
- Many different approaches to CD
  - The Tool vs the Process
  - Approach 1: One Workspace per Layer
  - Approach 2: One Workspace for the three layers
- The Deployment Pipeline
  - Supported Items
  - Item Pairing
  - Rules and Limitations

Introduction to the CD Process

Continuous Deployment (CD) is a practice that automates the release of code changes to testing and production environments, ensuring safe and reliable lifecycle management of organizational content.

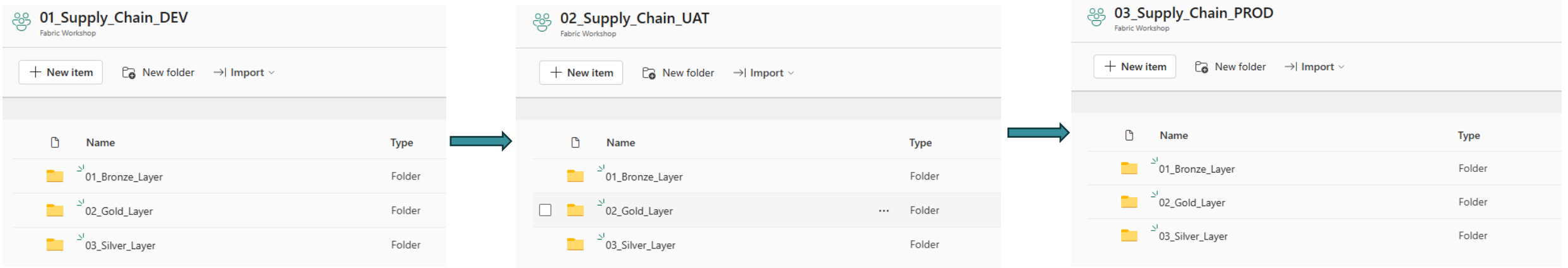


Enable creators to develop and test content in the service before it reaches the users.

Different Approaches to the CD Process within Fabric using the medallion architecture

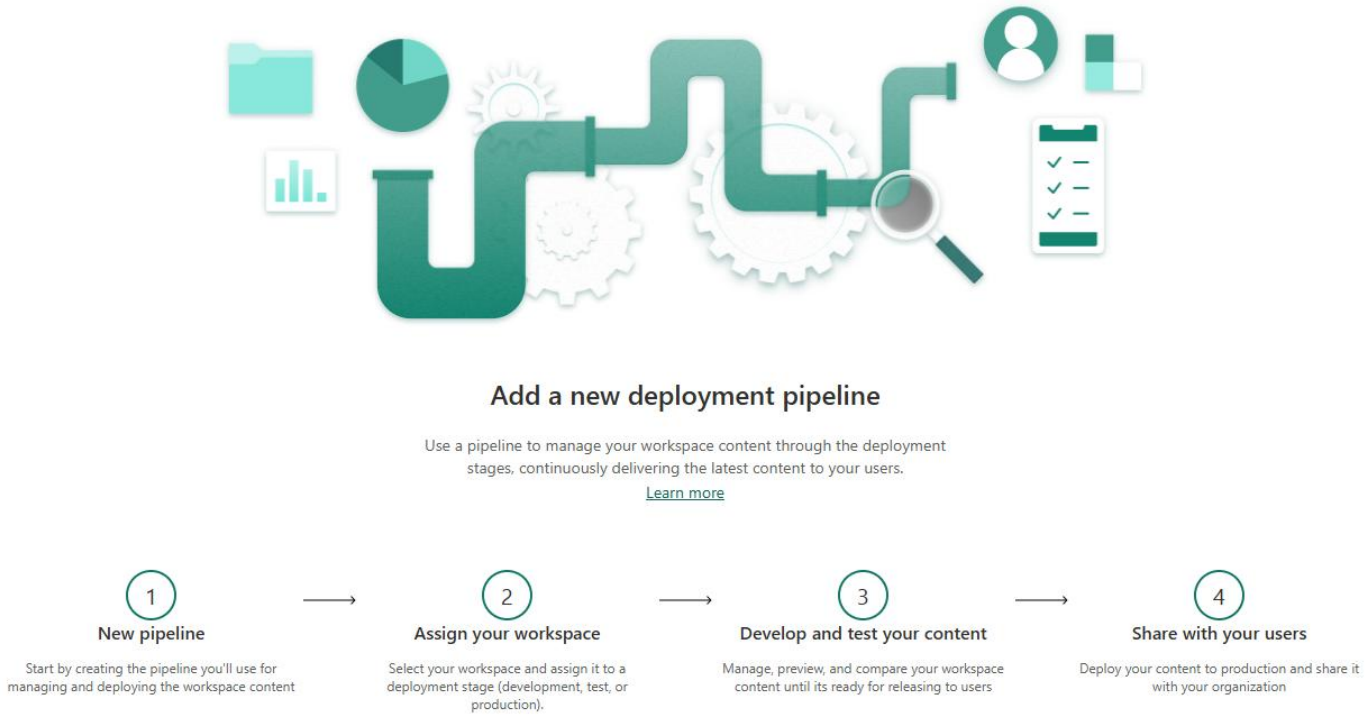
One Workspace per layer: 9 Workspaces needed

Three layers in one Workspace: 3 Workspaces need



The Deployment Pipeline

Microsoft Fabric's Tool for achieving the Continuous Deployment Process



Allows you to clone items across stages, define deployment rules and track changes across the deployment history

Supported items

When you deploy content from one pipeline stage to another, the copied content can contain the following items:

- Activator
- Dashboard
- Data pipeline *(preview)*
- Dataflows gen2 *(preview)*
- Datamart *(preview)*
- Environment *(preview)*
- Eventhouse and KQL database
- EventStream *(preview)*
- KQL Queryset
- Lakehouse *(preview)*
- Mirrored database *(preview)*
- Notebook
- Org app *(preview)*
- Paginated report
- Power BI Dataflow
- Real-time Dashboard
- Report (based on supported semantic models)
- Semantic model (that originates from a .pbix file and isn't a PUSH dataset)
- SQL database *(preview)*
- Warehouse *(preview)*

Item properties that are not copied

The following item properties aren't copied during deployment:

- Data - Data isn't copied. Only metadata is copied
- URL
- ID
- Permissions - For a workspace or a specific item
- Workspace settings - Each stage has its own workspace
- App content and settings - To update your apps, see Update content to Power BI apps
- Personal bookmarks

The following semantic model properties are also not copied during deployment:

- Role assignment
- Refresh schedule
- Data source credentials
- Query caching settings (can be inherited from the capacity)
- Endorsement settings

Deployment Rules

Item	Data source rule	Parameter rule	Default lakehouse rule	Details
Dataflow	✅	✅	❌	Use to determine the values of the data sources or parameters for a specific dataflow.
Semantic model	✅	✅	❌	Use to determine the values of the data sources or parameters for a specific semantic model.
Datamart	✅	✅	❌	Use to determine the values of the data sources or parameters for a specific datamart.
Paginated report	✅	❌	❌	Defined for the data sources of each paginated report. Use to determine the data sources of the paginated report.
Mirrored database	✅	❌	❌	Defined for the data sources of each mirrored database.
Notebook	❌	❌	✅	Use to determine the default lakehouse for a specific notebook.

The three types of deployment rules enable the automatic repointing of data sources across environments

Compare content in different stages