

Fabric Workshop

2025-02-28 Part 1

The Team



Roberto



Diego



Horner



Jesus



Fernando



Arturo



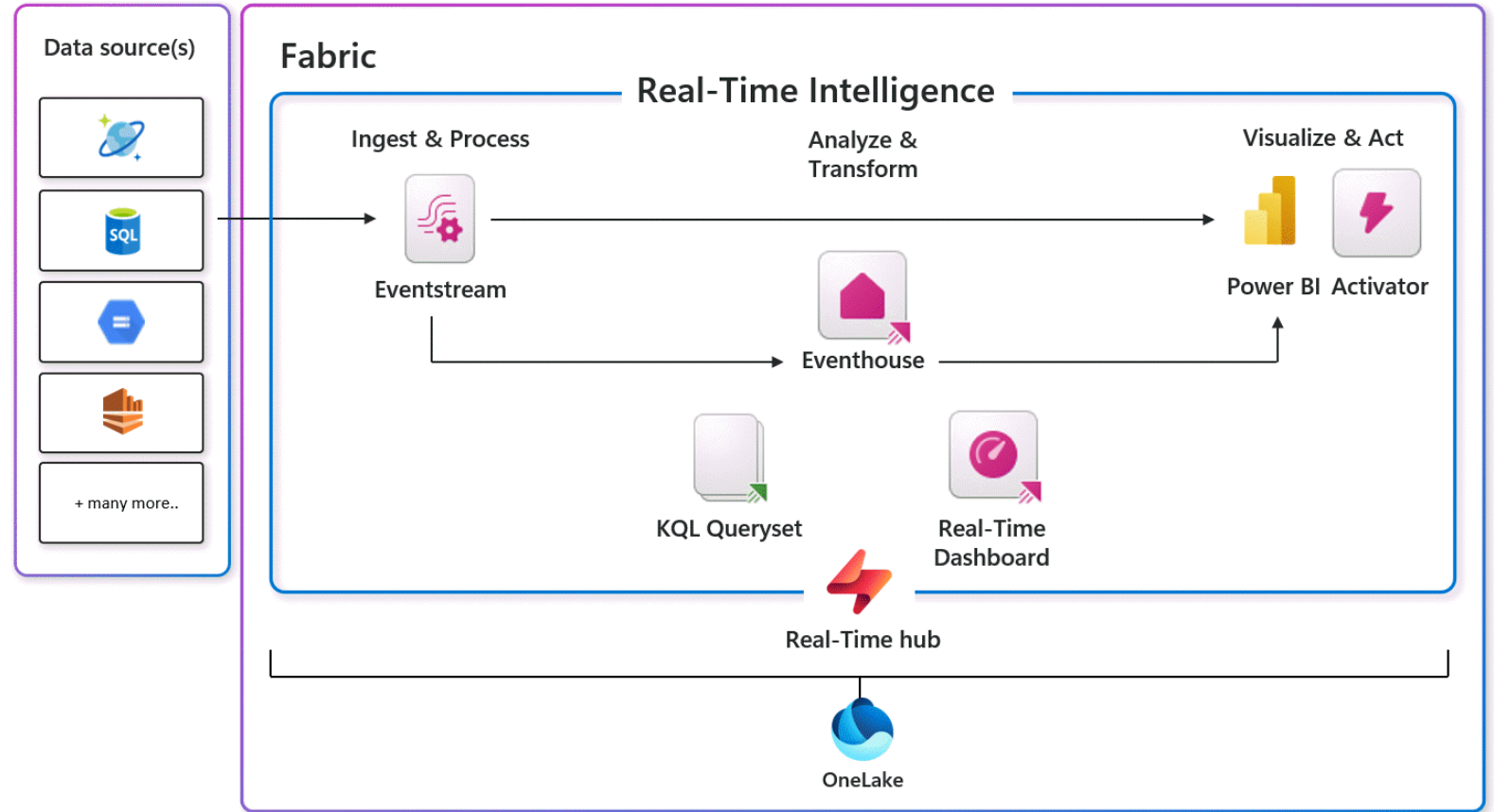
Michelle

Real-Time Intelligence

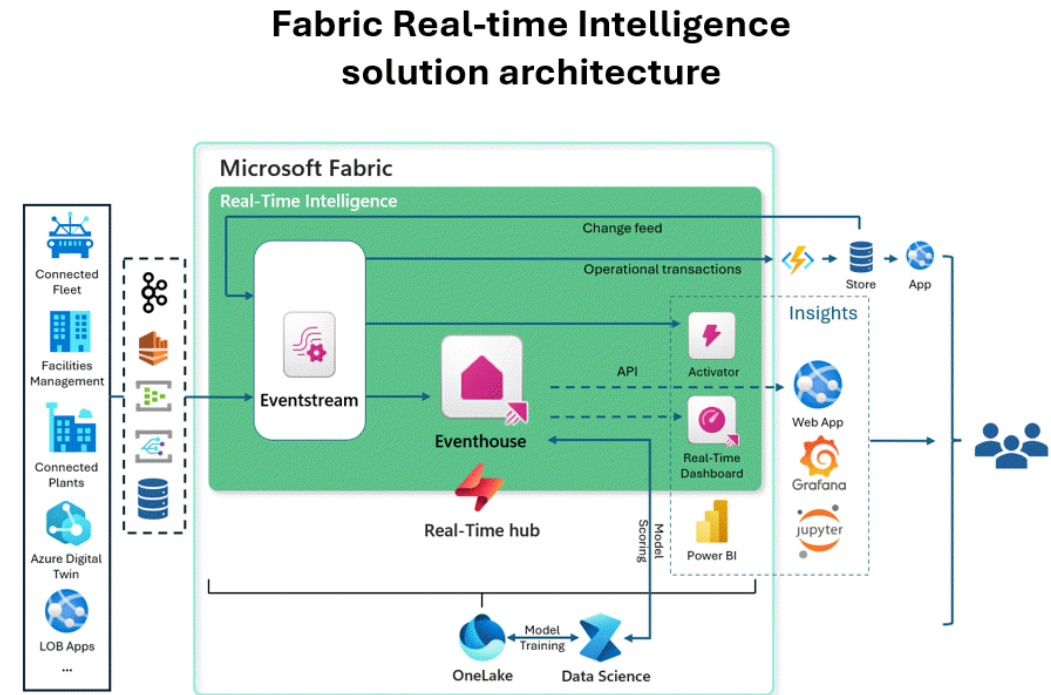
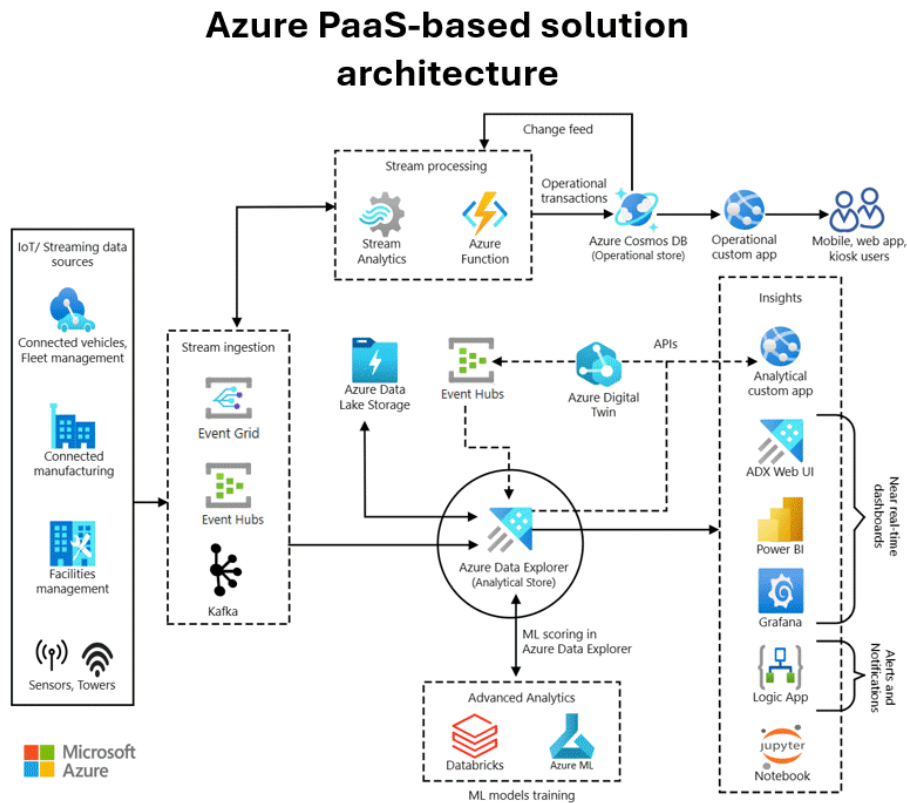
An abstract graphic on the left side of the slide, featuring several concentric, dark gray circular bands of varying thicknesses on a black background. The bands are centered vertically and horizontally within the left half of the image, creating a ripple effect.

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

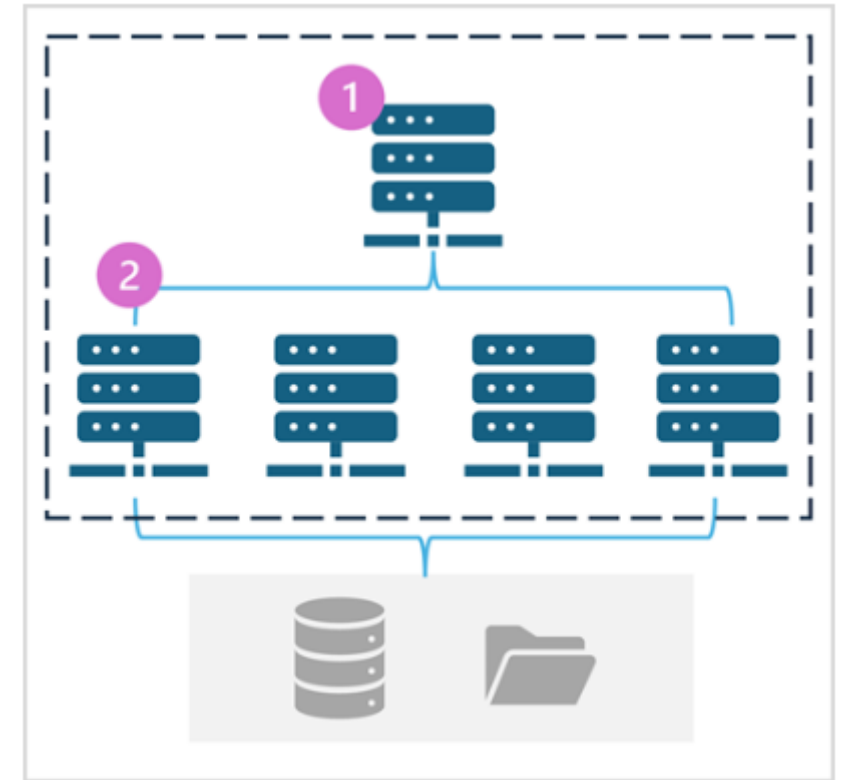
End-to-end Demo

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**.

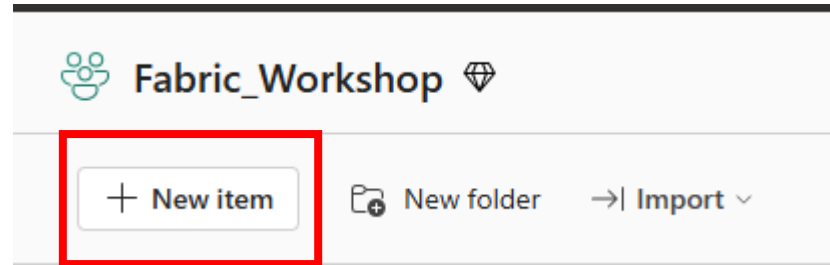


What are Fabric Notebooks?

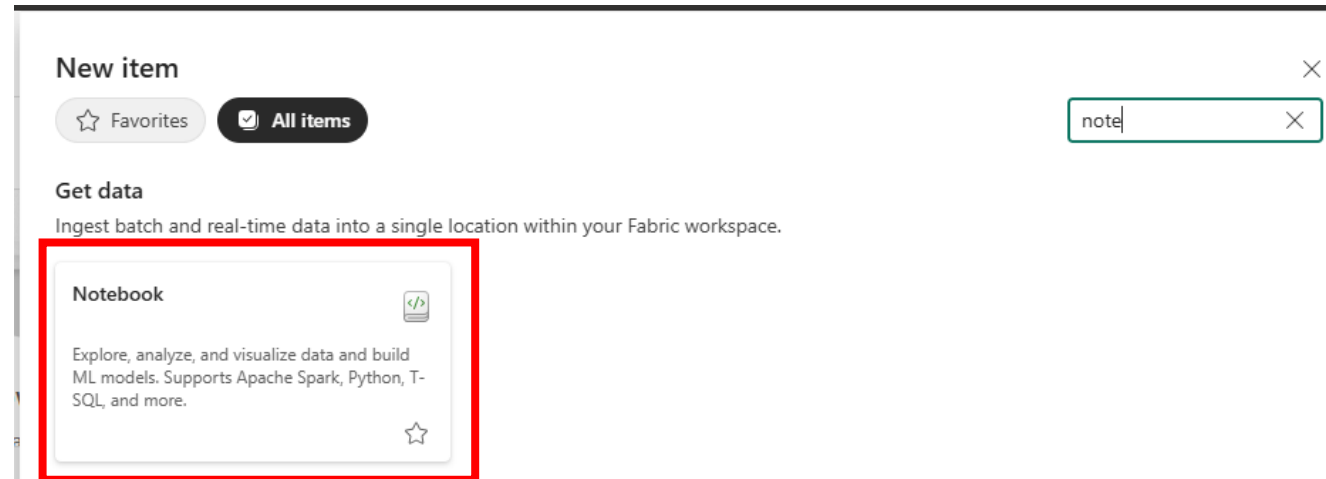
- Primary programming experience in Microsoft Fabric.
- Multiple programming languages supported.
 - Spark – Scala, PySpark, SparkR, Spark SQL
 - Python (Preview)
 - T-SQL
- Applications
 - ETL
 - EDA
 - Machine Learning

Getting started in 2 steps...

1. Click on the “+ **New item**” button.

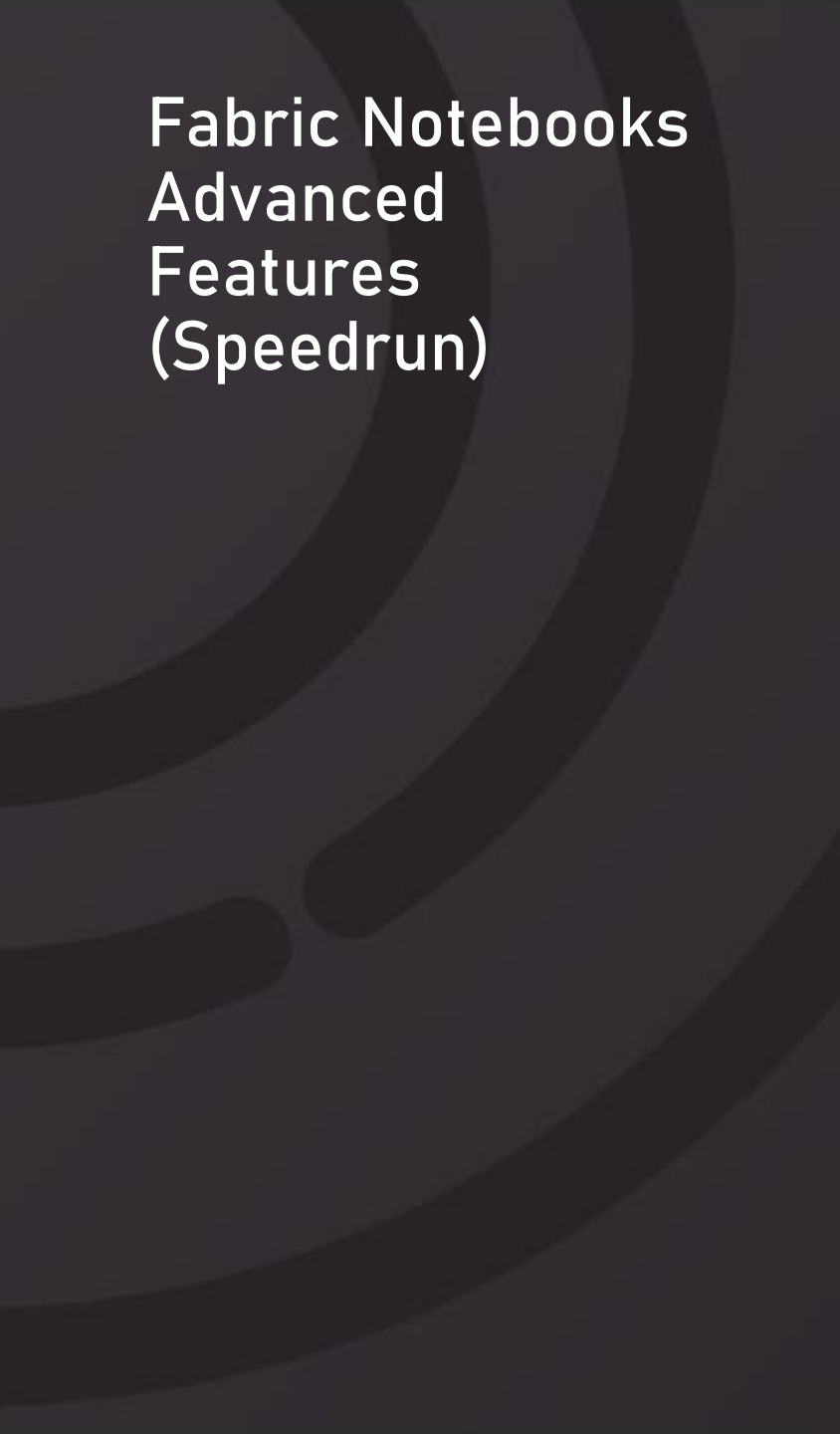


2. Click on **Notebook** item.



Exercise: Notebooks Demo and Tutorial



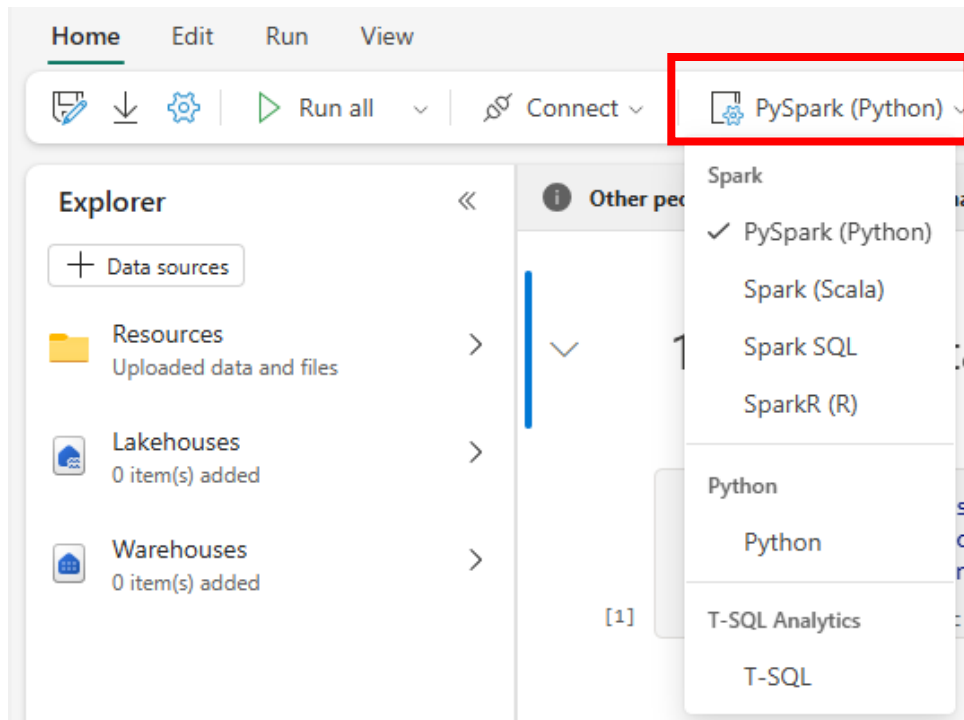


Fabric Notebooks Advanced Features (Speedrun)

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

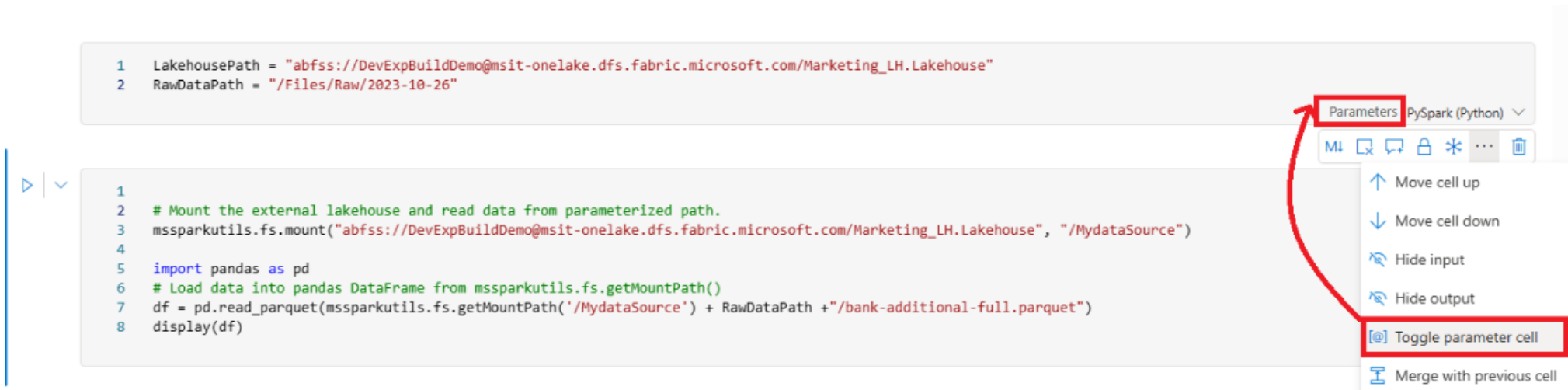
- Set for entire notebook for single cell.



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.



The screenshot shows a Jupyter Notebook interface with two code cells. The first cell contains two lines of Python code defining paths. The second cell contains a block of Python code for mounting a lakehouse, importing pandas, and reading a parquet file. A context menu is open on the right side of the second cell, with a red arrow pointing to the 'Parameters' tab and another red box highlighting the '@ Toggle parameter cell' option.

```
1 LakehousePath = "abfss://DevExpBuildDemo@msit-onelake.dfs.fabric.microsoft.com/Marketing_LH.Lakehouse"
2 RawDataPath = "/Files/Raw/2023-10-26"
```

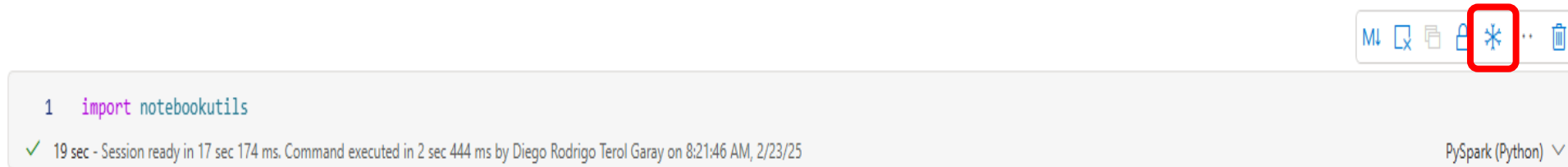
```
1
2 # Mount the external lakehouse and read data from parameterized path.
3 mssparkutils.fs.mount("abfss://DevExpBuildDemo@msit-onelake.dfs.fabric.microsoft.com/Marketing_LH.Lakehouse", "/MydataSource")
4
5 import pandas as pd
6 # Load data into pandas DataFrame from mssparkutils.fs.getMountPath()
7 df = pd.read_parquet(mssparkutils.fs.getMountPath('/MydataSource') + RawDataPath + "/bank-additional-full.parquet")
8 display(df)
```

Parameters PySpark (Python) ▾

- Move cell up
- Move cell down
- Hide input
- Hide output
- @ Toggle parameter cell**
- Merge with previous cell

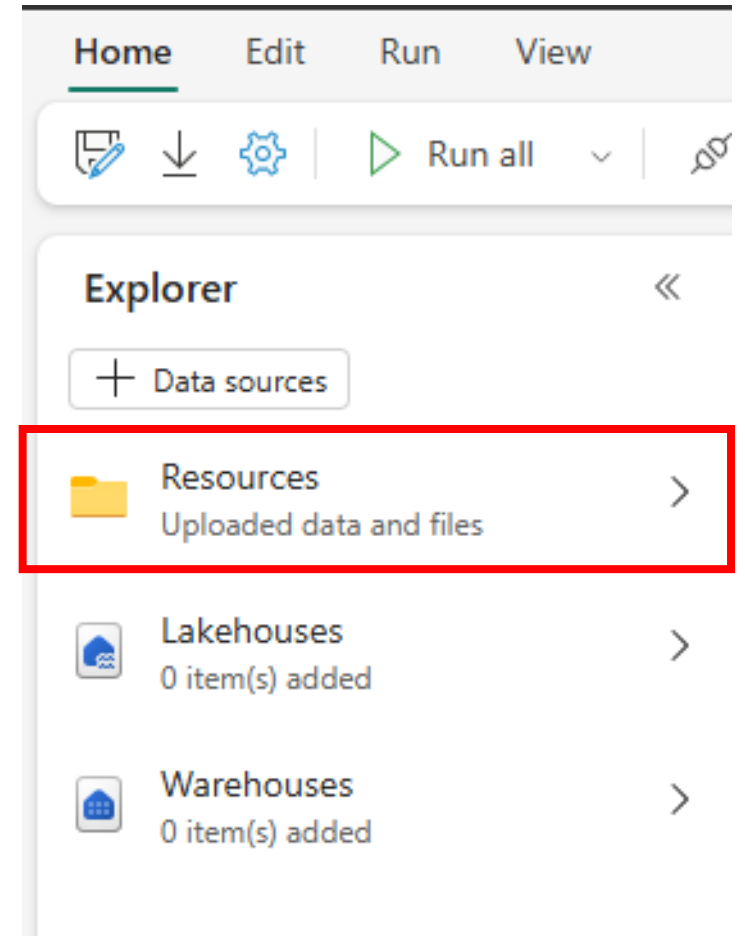
Frozen Cells

- 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.

The image displays two side-by-side screenshots from the Databricks workspace interface.

Left Screenshot: Workspace settings

- Runtime Version:** A dropdown menu is set to "1.1 (Spark 3.3, Delta 2.2)".
- High concurrency:** A toggle switch is turned "On". A green box highlights this section, which includes the text: "High Concurrency mode enables multiple notebooks to use the same Spark application to save session start time." and a link to "Learn more about High concurrency".

Right Screenshot: Workspace Explorer and Command Bar

- Command Bar:** Includes buttons for "Connect" and "PySpark (Python)".
- Explorer:** Shows sections for "Data sources", "Resources" (Uploaded data and files), and "Lakehouses" (0 item(s) added).
- Session Creation Menu:** A dropdown menu is open, showing two options: "New standard session" and "New high concurrency session". The "New high concurrency session" option is highlighted with a red rectangular border.
- Session Status:** Below the menu, it says "Available high concurrency sessions" and "No available sessions found".
- Footer:** A link to "View session information" is visible.

Configure Spark Starter Pools

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

Edit pool

Spark pool name *

StarterPool

Node family

Memory optimized

Node size

Medium

Autoscale

If enabled, your Apache Spark pool will automatically scale up and down based on the amount of activity.

☒ Enable autoscale

1

16

Dynamically allocate executors

☒ Enable allocate

1

15

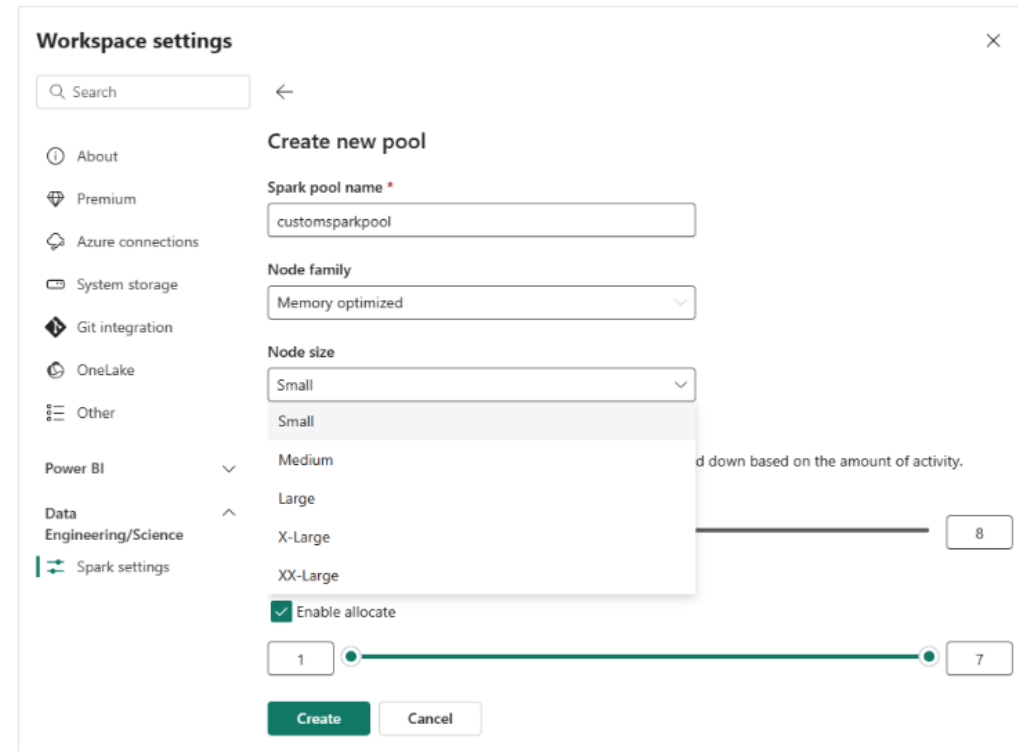
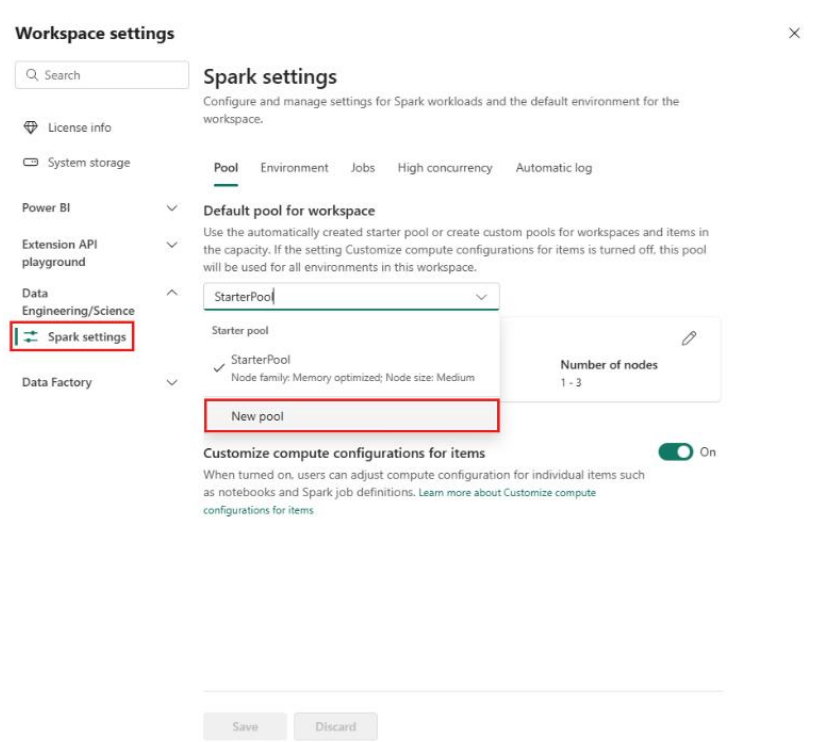
Save

Discard

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.

The screenshot shows the 'Public Libraries' tab in the Databricks workspace. The interface includes a top navigation bar with 'Home' and 'Public Libraries' tabs. Below the navigation bar, there are buttons for '+ Add from PyPI', '+ Add from .yaml', 'Delete', and 'Export to .yaml'. A warning message states: 'Environment is in Preview. Other users in your organization may have access to this workspace. Do not use this item unless you trust all other users who may have access to the workspace.' The left sidebar contains a 'Libraries' section with 'Public Libraries' and 'Custom Libraries' options, and a 'Spark Compute' section with 'Compute' and 'Spark properties' options. The main content area is titled 'Public Libraries' and contains a search bar 'Filter by name'. Below the search bar is a table with columns: 'Library', 'Version', 'Source', 'Status', and 'Last updated'. The table has one row for the library 'fuzzywuzzy' with version '0.18.0', source 'PyPI', status 'Success', and last updated '10/24/23, 04:22:02 PM'. A 'Dependencies' button is located next to the library name, and a mouse cursor is pointing at it.

Home Public Libraries

+ Add from PyPI + Add from .yaml Delete Export to .yaml

Environment is in Preview. Other users in your organization may have access to this workspace. Do not use this item unless you trust all other users who may have access to the workspace.

Libraries

- Public Libraries
- Custom Libraries

Spark Compute

- Compute
- Spark properties

Public Libraries

Search and add libraries from public repositories or via a .yaml file. They'll be available if you run your notebook or Spark job definition in this environment.

Filter by name

Library	Version	Source	Status	Last updated
fuzzywuzzy	0.18.0	PyPI	Success	10/24/23, 04:22:02 PM

Manage Notebooks with APIs

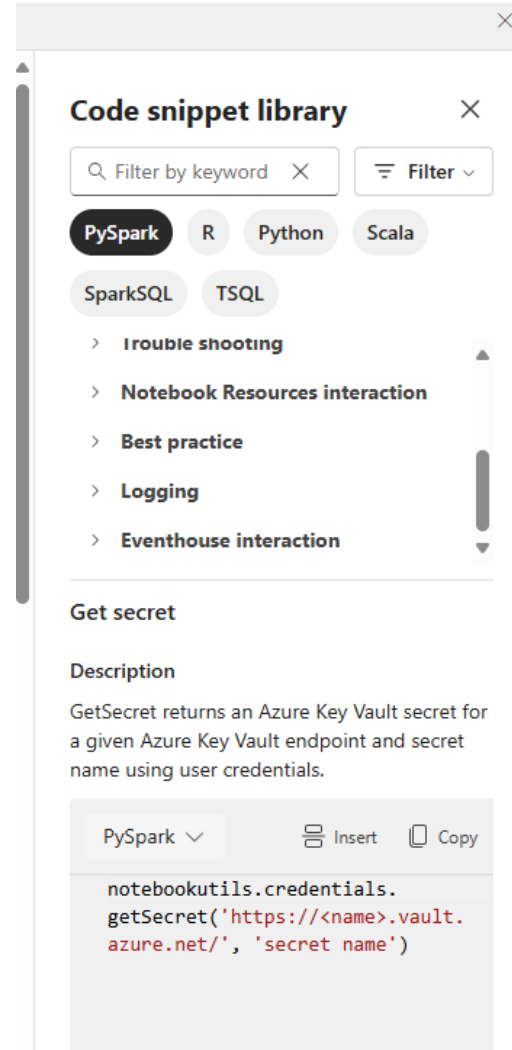
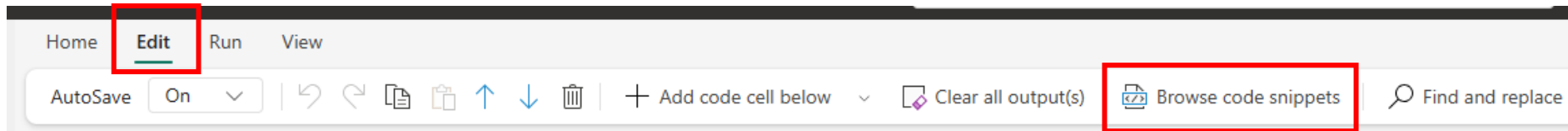
- Manage CRUD operations and Notebook Run instances.

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.

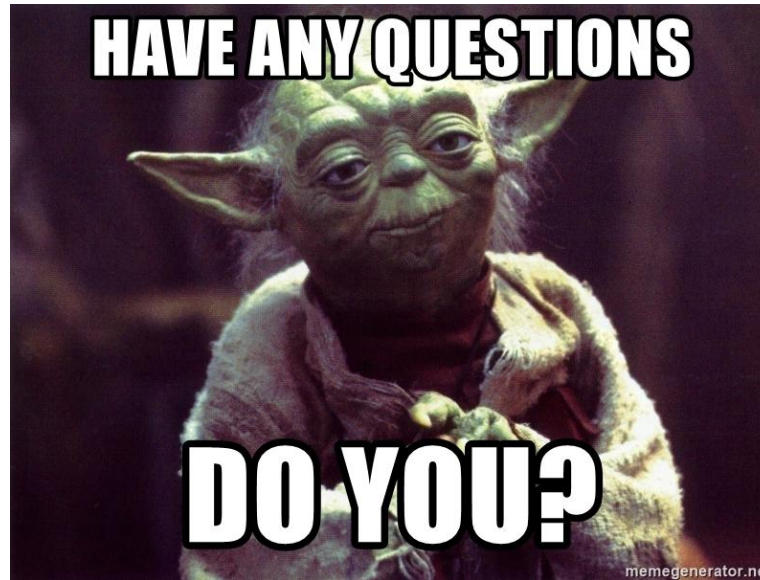
Action	Description
Run on demand Item Job	Run notebook with parameterization.
Cancel Item Job Instance	Cancel notebook job run.
Get Item Job Instance	Get notebook run status.

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.

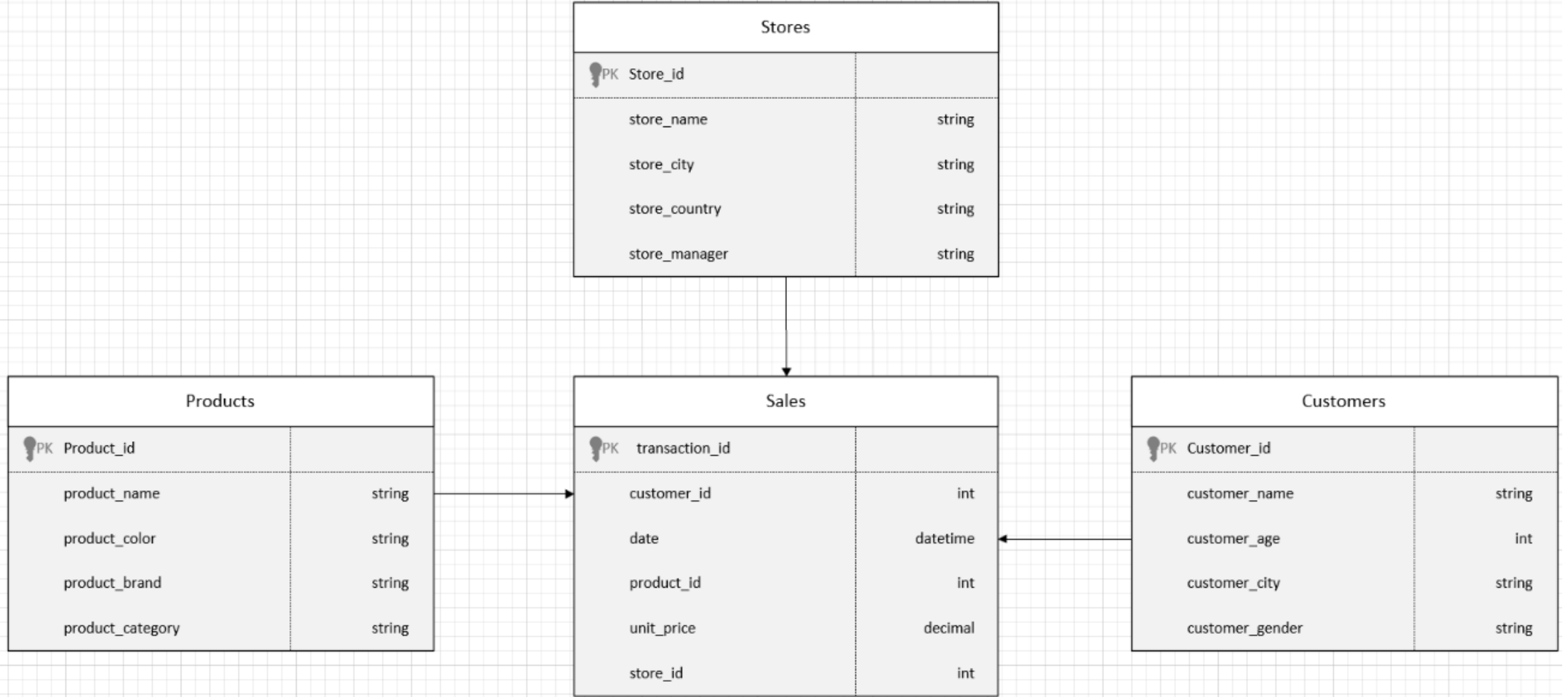
You define your business logic!

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



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.

Admin portal

The screenshot shows the Microsoft Fabric Admin portal interface. On the left, a sidebar menu lists various settings: Tenant settings (highlighted with a red box and a 'New' badge), Usage metrics, Users, Premium Per User, Audit logs, Domains (with a 'New' badge), Workloads, Tags (preview) (with a 'New' badge), Capacity settings, Refresh summary, Embed Codes, Organizational visuals, Azure connections, Workspaces, Custom branding, Fabric identities, and Featured content. The main content area is titled 'Copilot and Azure OpenAI Service' (highlighted with a red box). It contains a sub-header 'Users can use Copilot and other features powered by Azure OpenAI' followed by the text 'Enabled for the entire organization'. Below this, there is explanatory text about the setting's scope and a link to 'Learn More'. A toggle switch is shown in the 'Enabled' position (highlighted with a red box). At the bottom, there are two yellow informational notes: one stating that Copilot in Fabric is generally available starting with the Microsoft Power BI experience, and another note about data processing outside the geographic region if Azure OpenAI is not available.

Tenant settings New

Usage metrics

Users

Premium Per User

Audit logs

Domains New

Workloads

Tags (preview) New

Capacity settings

Refresh summary

Embed Codes

Organizational visuals

Azure connections

Workspaces

Custom branding

Fabric identities

Featured content

Copilot and Azure OpenAI Service

Users can use Copilot and other features powered by Azure OpenAI
Enabled for the entire organization

When this setting is enabled, users can access the features powered by Azure OpenAI, including Copilot. This setting can be managed at both the tenant and the capacity levels. [Learn More](#)

For customers in the EU Data Boundary, this setting adheres to Microsoft Fabric's EU Data Boundary commitments. [Learn More](#)

By enabling this setting, you agree to the [Preview Terms](#).

☒ Enabled

Note: Copilot in Fabric is now generally available, starting with the Microsoft Power BI experience. The Copilot in Fabric experiences for Data Factory, Data Engineering, Data Science, Data Warehouse, and Real-Time Intelligence are in preview.

Note: If Azure OpenAI is not available in your geographic region, your data may need to be processed outside your capacity's geographic region, compliance boundary, or national cloud instance. To allow data to be processed outside your capacity's geographic region, turn on the related setting, "Data sent to Azure OpenAI can be processed outside your capacity's geographic region, compliance boundary, or national cloud instance".

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"

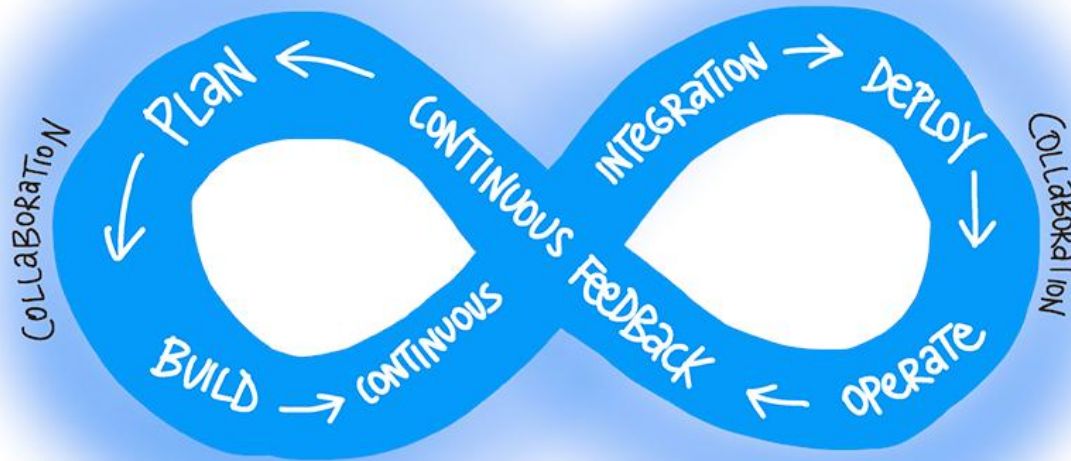
Exercise Import Mode

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)

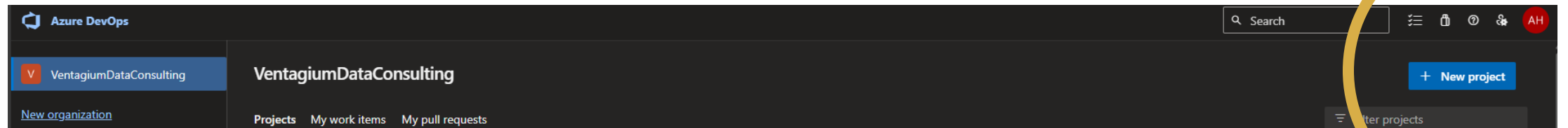


Leverage your development (CD)



Version control

Create a DevOps Project



Create a DevOps Project

Create new project

Project name *

Fabric Workshop

Description

Visibility

Public

Anyone on the internet can view the project. Certain features like TFVC are not supported.

Private

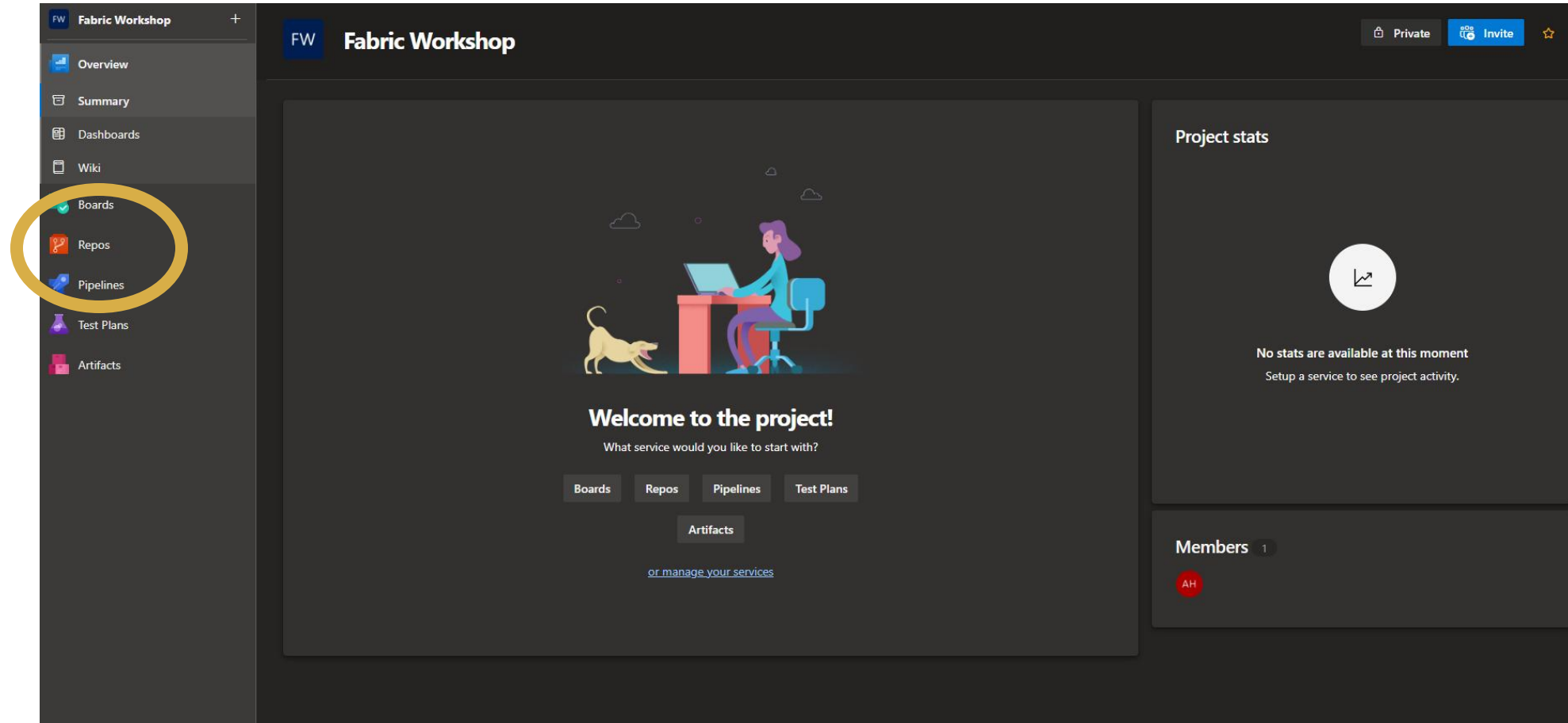
Only people you give access to will be able to view this project.

Advanced

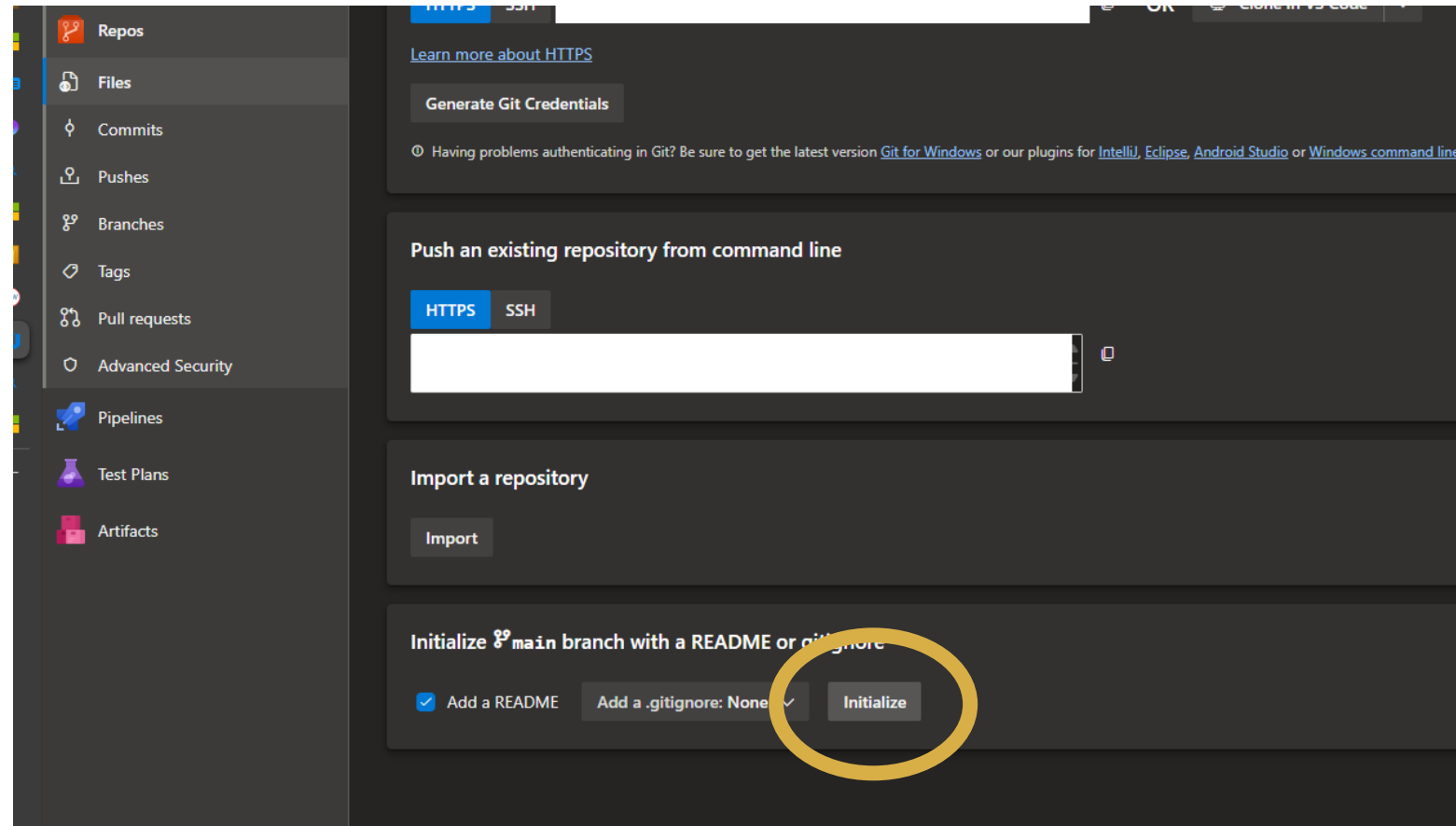
Cancel

Create

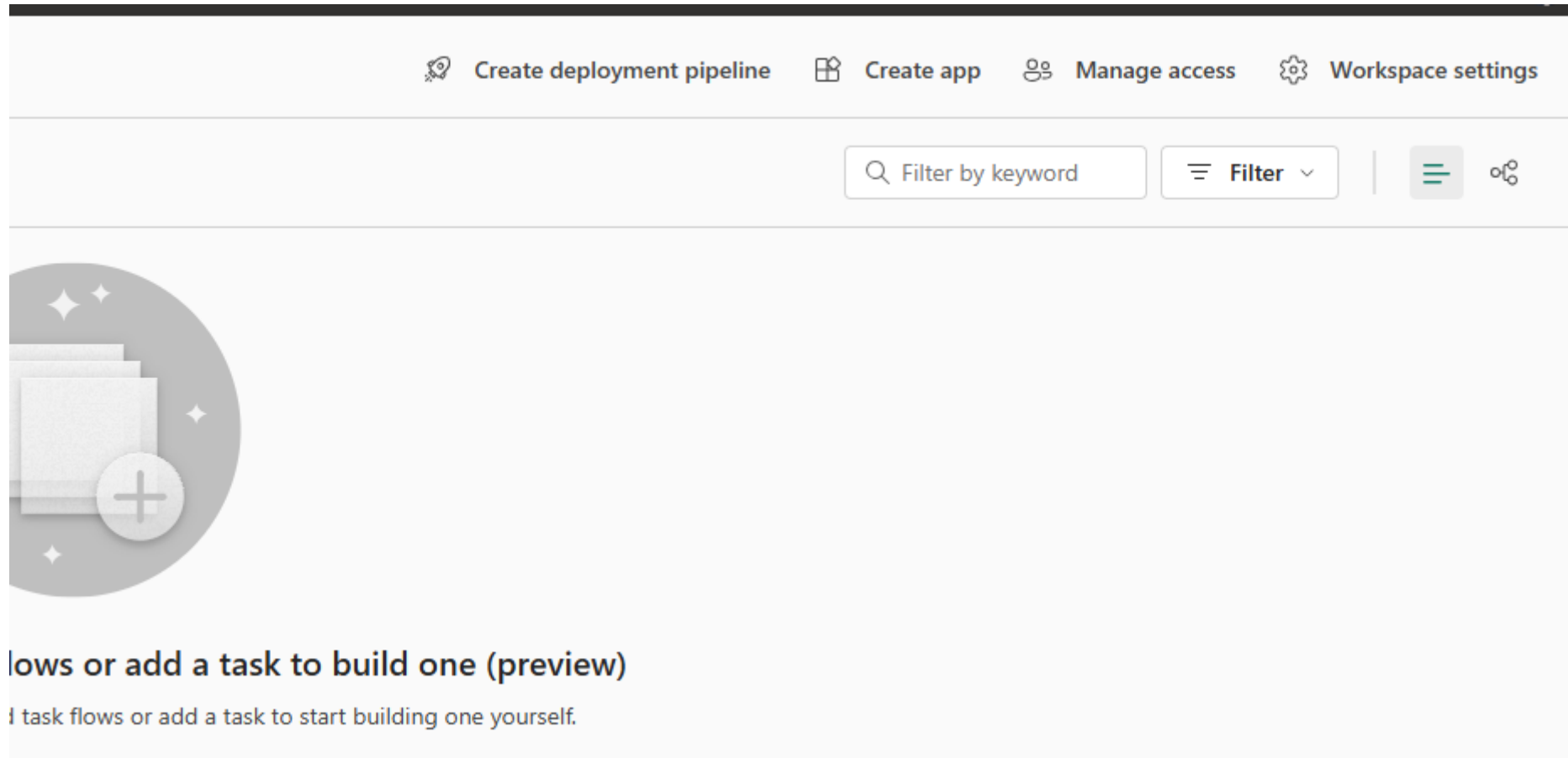
Initialize a Repo



Initialize a Repo



Enable Git Integration



Enable Git Integration

Workspace settings

General

- License info
- Azure connections
- System storage
- Git integration**
- OneLake
- Workspace identity
- Network security
- Monitoring

Power BI

General

About

Workspace image



Name *

Fabric_Workshop

Description

Describe this workspace (O

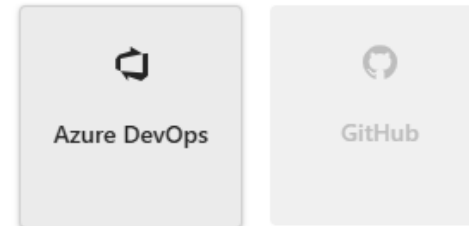
Git integration

Connect to Git to manage your code and back up your work. [Learn more](#)

Preview items Some item types are only available in preview when using Git. [Learn more](#)

Connect Git provider and account

Git provider














Accounts

☒ AAD account
alberto. com

Connect

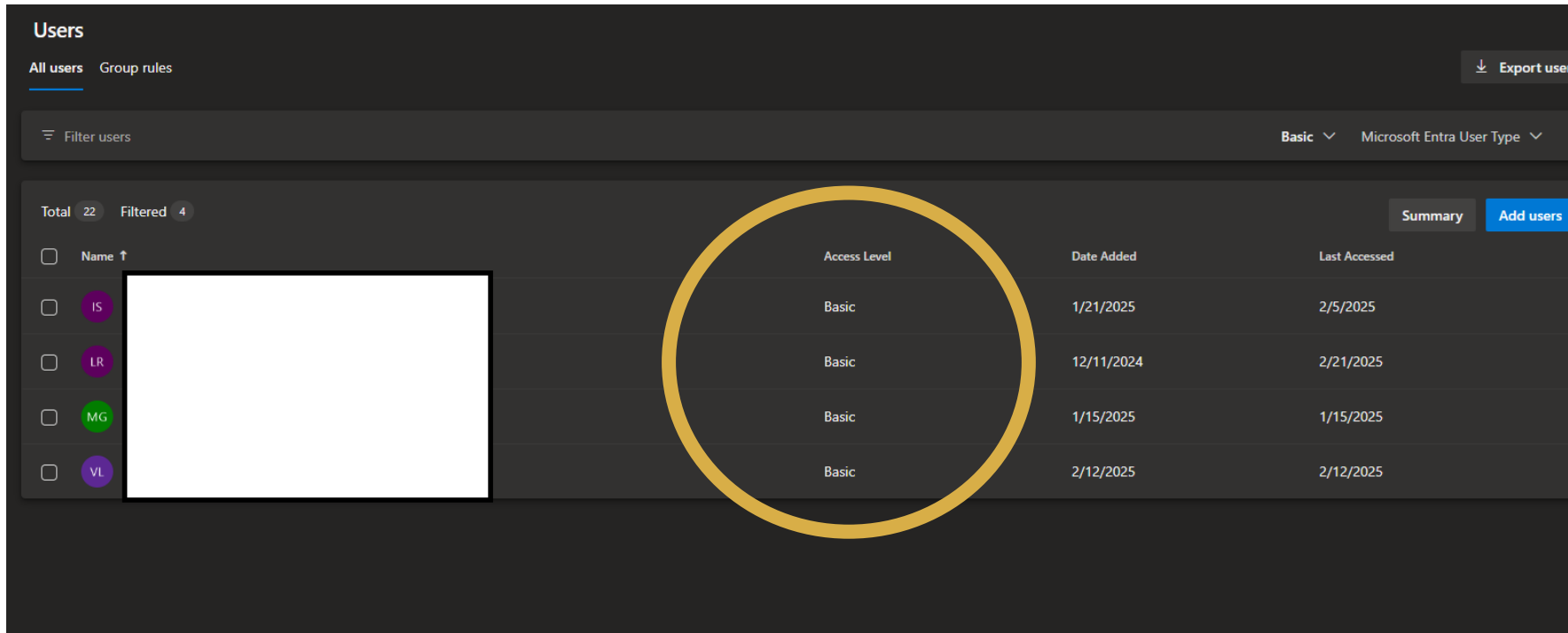
Connect Git repository and branch

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)



The screenshot displays the 'Users' management interface. At the top, there are tabs for 'All users' and 'Group rules', with 'All users' selected. An 'Export users' button is located in the top right corner. Below the tabs is a 'Filter users' section with a search bar. On the right side of the filter section, there are dropdown menus for 'Basic' and 'Microsoft Entra User Type'. Below the filter section, a summary bar shows 'Total 22' and 'Filtered 4'. On the left side of the user list, there is a sidebar with a search bar and a list of user initials (IS, LR, MG, VL) each with a checkbox. The main area displays a table of users. A yellow circle highlights the 'Access Level' column, which shows 'Basic' for all four users. The table also includes columns for 'Date Added' and 'Last Accessed'. Buttons for 'Summary' and 'Add users' are located at the top right of the table.

Name ↑	Access Level	Date Added	Last Accessed
IS	Basic	1/21/2025	2/5/2025
LR	Basic	12/11/2024	2/21/2025
MG	Basic	1/15/2025	1/15/2025
VL	Basic	2/12/2025	2/12/2025

Create Contributors Team

Project Settings

Fabric Workshop

General

Overview

Teams

Permissions

Notifications

Service hooks

Dashboards

Boards

Project configuration

Teams

Filter teams

X

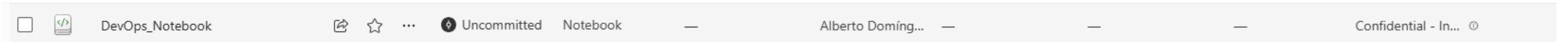
Total 2

New Team

Name ↓	Description	Members
FC Fabric Workshop Contributors		6
FT Fabric Workshop Team	Default The default project team.	1

Commit and Sync your Items

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



Commit and Sync your Items

create deployment pipeline Create app Manage access Workspace settings

Source control 2 Filter by keyword Filter

Next refresh	Endorsement
—	—
N/A	—
—	—
—	—
N/A	—
—	—
—	—

Source control

Current branch: main

Changes 2 Updates

feat: idle change to track it on devOps

	Item	Status
<input type="checkbox"/>	OpenAI_API_Notebook	+
<input checked="" type="checkbox"/>	DevOps_Notebook	+

Commit Undo

Create Branch policies

The screenshot shows the 'Branches' page in the Azure DevOps interface. The left sidebar contains navigation links: Overview, Boards, Repos, Files, Commits, Pushes, Branches (selected), Tags, Pull requests, Advanced Security, Pipelines, Test Plans, and Artifacts. The main area is titled 'Branches' and has tabs for 'Mine', 'All', and 'Stale'. A search bar for 'Search branch name' is at the top right. A table lists branches with columns: Branch, Commit, Author, Authored Date, Behind | Ahead, Status, and Pull Req... The 'main' branch is listed with commit '1e46bac8' by 'Alberto Domíng...' 10m ago. A context menu is open for the 'main' branch, showing options: New branch, New pull request, Delete branch, View files, View history, Compare branches, Set as compare branch, Set as default branch, Lock, Branch policies (highlighted with a yellow circle), and Branch security.

Branch	Commit	Author	Authored Date	Behind Ahead	Status	Pull Req...
main	1e46bac8	Alberto Domíng...	10m ago			

- + New branch
- New pull request
- Delete branch
- View files
- View history
- Compare branches
- Set as compare branch
- Set as default branch
- Lock
- Branch policies
- Branch security

Branch Policies

Fabric_Workshop

Create deployment pipeline>Create app

+ New item

New folder

→ Import

Source control 2

Enter by key

	Name	Git status	Type	Task	Owner	Refreshed	Next refresh	Endorsement	Se
	OpenAI Demo		Folder	—	—	—	—	—	—
	Dataflow 1	⊘ Unsupported	Dataflow Gen2	—	Roberto Hernán...	—	N/A	—	—
	DevOps_Notebook	⦿ Uncommitted	Notebook	—	Alberto Domíng...	—	—	—	Co
	Fabric Workshop 01	✓ Synced	SQL database	—	Roberto Hernán...	—	—	—	Co
	Fabric Workshop 01	—	Semantic model...	—	Fabric_Workshop	2/18/2025, 10:45:1...	N/A	—	Co
	Fabric Workshop 01	—	SQL application	—	Roberto Hernán...	—	—	—	Co

Notifications

Clear all

Earlier

✗ Unable to complete commit request

View details below to learn more.

View details

6 minutes ago

✓ Your selected changes were committed.

14 minutes ago

ⓘ Session stopped successfully.

19 minutes ago

Branch Policies

The screenshot displays a GitHub Pull Request (PR) titled "PR Example - DevOps Tutorial". The PR is in the "Active" state, with the number "11469" and the author "Alberto Domínguez Horner" (AH). The PR description is "Alberto Domínguez Horner proposes to merge development into main".

The "Overview" tab is selected, showing the following details:

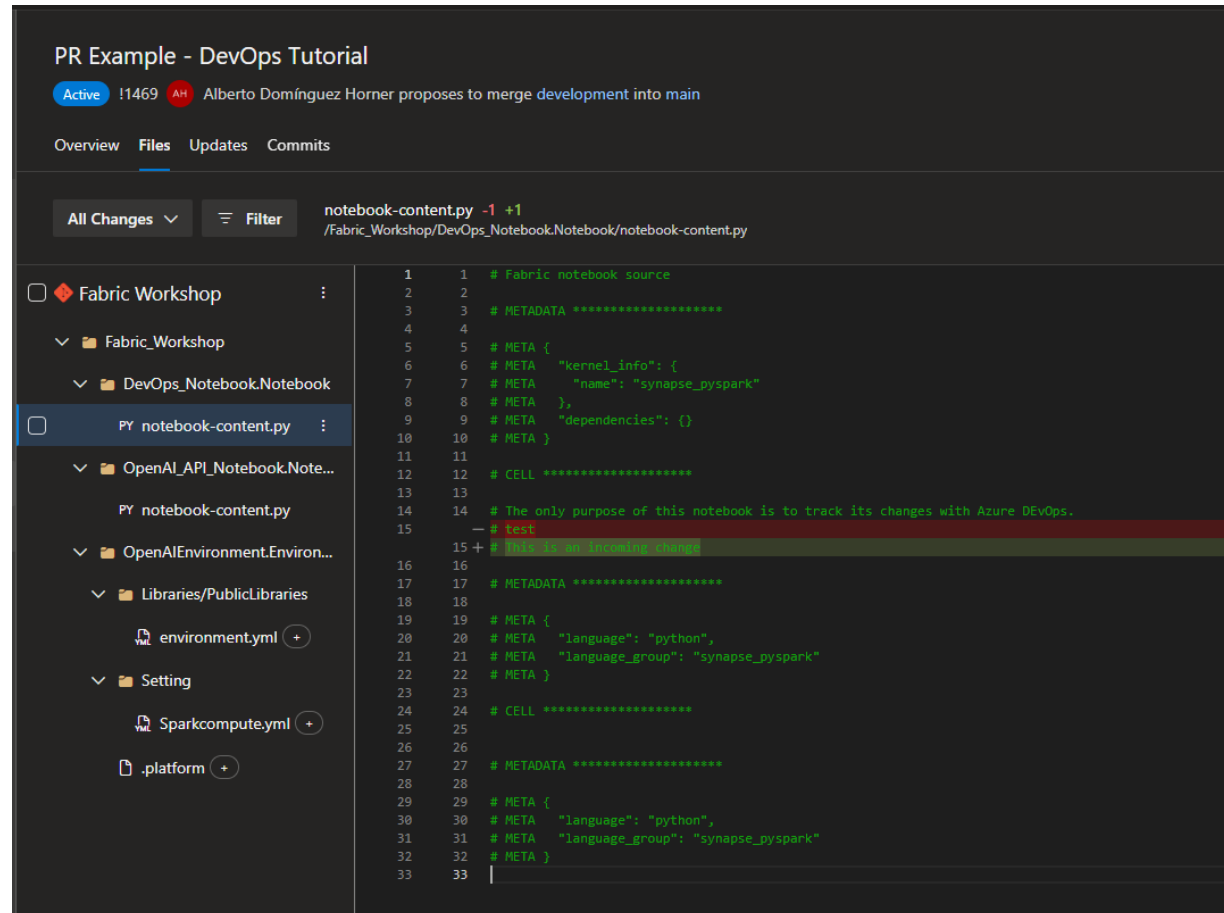
- Reviewers:** Alberto Domínguez Horner must approve.
- Checks:** No merge conflicts (Last checked: Just now).
- Description:** A text area for the PR description.
- Tags:** No tags.
- Work items:** No work items.

The "Commits" tab is also visible, showing a list of commits. The first commit is highlighted with a yellow oval:

- Alberto Domínguez Horner pushed 1 commit (Just now)
- test: incoming changes (Just now)
- Alberto Domínguez Horner was added as a required reviewer. (3m ago)
- Alberto Domínguez Horner created the pull request (3m ago)

The PR is currently approved, as indicated by the "Approve" button and the "Set auto-complete" dropdown menu.

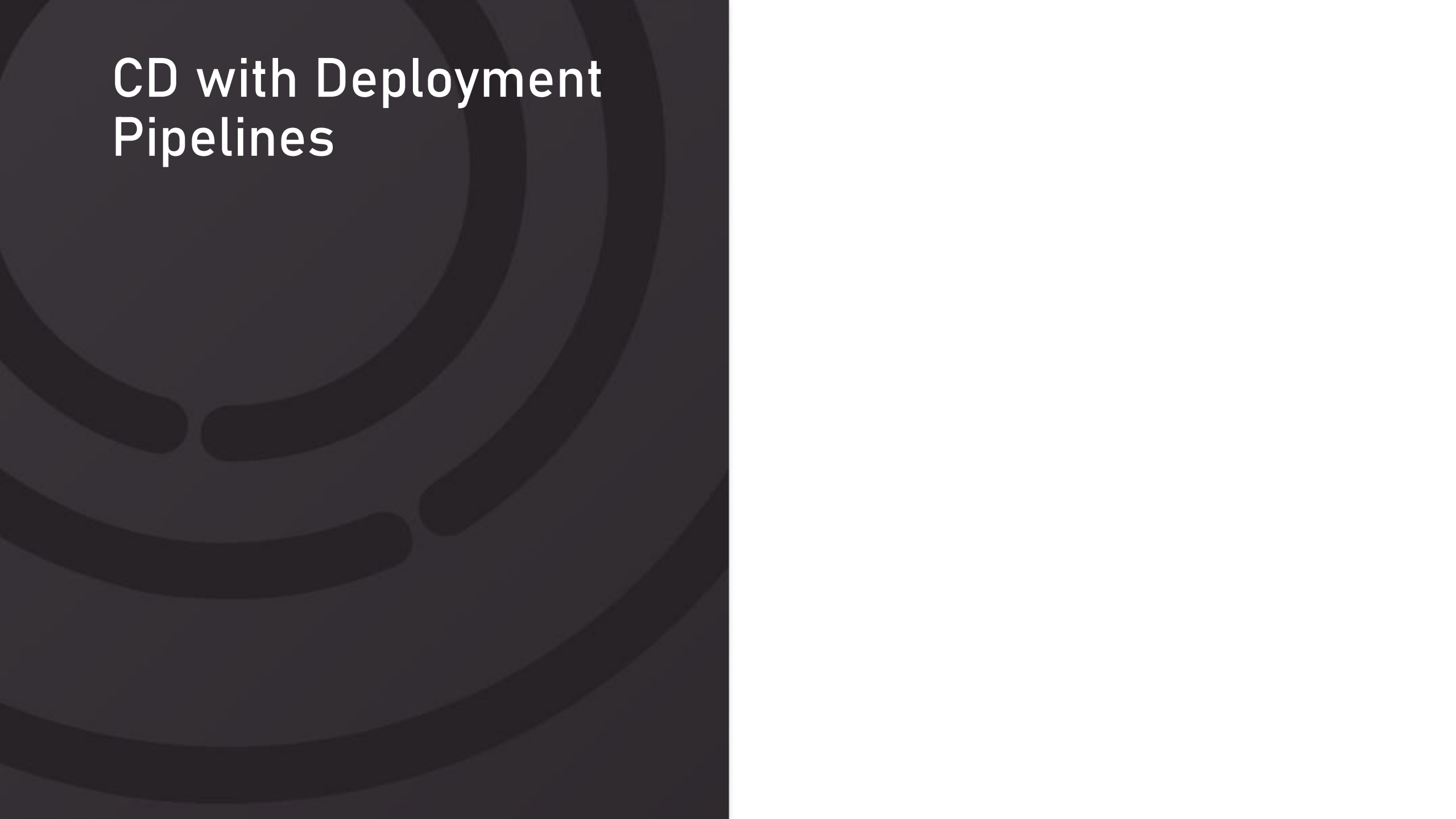
Continuous Integration by Pull Requests



The screenshot displays a GitHub Pull Request (PR) titled "PR Example - DevOps Tutorial". The PR is active and proposed by Alberto Domínguez Horner, aiming to merge development into main. The interface shows the "Files" tab, highlighting a change in the file `notebook-content.py` within the `Fabric_Workshop/DevOps_Notebook.Notebook` directory. The diff shows a single line added (line 15): `# This is an interesting change`. The left sidebar shows the repository structure, and the right pane displays the code with line numbers and syntax highlighting.

```
1 1 # Fabric notebook source
2 2
3 3 # METADATA *****
4 4
5 5 # META {
6 6 # META   "kernel_info": {
7 7 # META     "name": "synapse_pyspark"
8 8 # META   },
9 9 # META   "dependencies": {}
10 10 # META }
11 11
12 12 # CELL *****
13 13
14 14 # The only purpose of this notebook is to track its changes with Azure DEVOps.
15 15 - # test
16 16 + # This is an interesting change
17 17
18 18 # METADATA *****
19 19
20 20 # META {
21 21 # META   "language": "python",
22 22 # META   "language_group": "synapse_pyspark"
23 23 # META }
24 24
25 25 # CELL *****
26 26
27 27 # METADATA *****
28 28
29 29 # META {
30 30 # META   "language": "python",
31 31 # META   "language_group": "synapse_pyspark"
32 32 # META }
33 33 |
```

CD with Deployment Pipelines

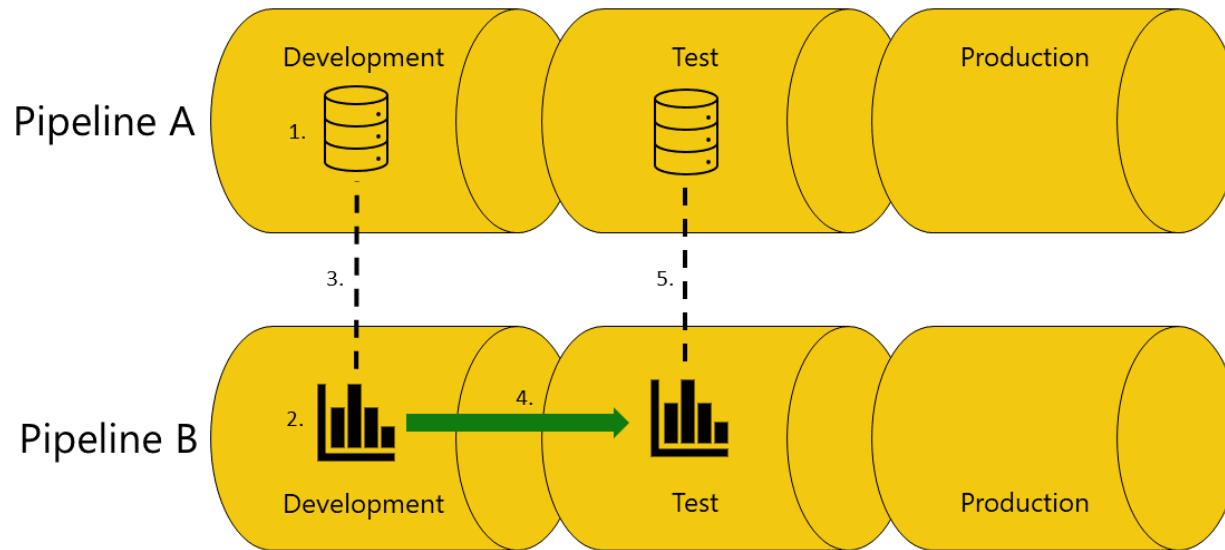
The background of the slide is split vertically. The left half is dark gray and features several concentric, slightly offset circles in varying shades of gray, creating a tunnel-like or ripple effect. The right half is a solid, bright white.

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


 01_Bronze_DEV
Fabric Workshop

+ New item

New folder

→| Import ▾




 01_Silver_DEV
Fabric Workshop

+ New item

New folder

→| Import ▾



 01_Gold_DEV
Fabric Workshop


+ New item

New folder

→| Import ▾






Three layers in one Workspace: 3 Workspaces need

 01_Supply_Chain_DEV
Fabric Workshop


+ New item

New folder

→| Import ▾

	Name	Type
	01_Bronze_Layer	Folder
	02_Gold_Layer	Folder
	03_Silver_Layer	Folder





 02_Supply_Chain_UAT
Fabric Workshop


+ New item

New folder

→| Import ▾

	Name	Type
	01_Bronze_Layer	Folder
<input type="checkbox"/>	02_Gold_Layer	Folder
	03_Silver_Layer	Folder






 03_Supply_Chain_PROD
Fabric Workshop

+ New item

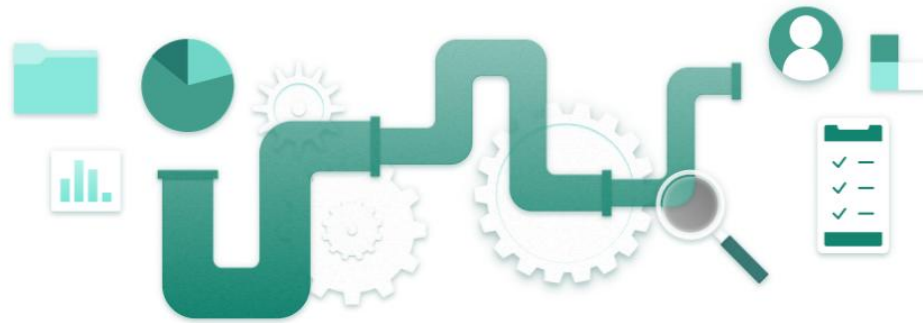
New folder

→| Import ▾

	Name	Type
	01_Bronze_Layer	Folder
	02_Gold_Layer	Folder
	03_Silver_Layer	Folder

The Deployment Pipeline

Microsoft Fabric's Tool for achieving the Continuous Deployment Process



Add a new deployment pipeline

Use a pipeline to manage your workspace content through the deployment stages, continuously delivering the latest content to your users.

[Learn more](#)

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

Change review

TestWS \ Retail Analysis Sample

To be modified [Test]

```
1 {
2   "annotations": [
3     {
4       "name": "PBIDesktopVersion",
5       "value": "2.91.884.0 (21.03)"
6     },
7     {
8       "name": "PBI_QueryOrder",
9       "value": "[\\"Store\\",\\"Sales\\",\\"Item\\",\\"Time\\",\\"District\\"]"
10    },
11    {
12      "name": "__PBI_TimeIntelligenceEnabled",
13      "value": "1"
14    }
15  ],
16  "culture": "en-US",
```

To be deployed [Development]

```
1 {
2   "annotations": [
3     {
4       "name": "PBIDesktopVersion",
5       "value": "2.115.6858.1"
6     },
7     {
8       "name": "PBI_QueryOrder",
9       "value": "[\\"Store\\",\\"Sales\\",\\"Item\\",\\"Time\\",\\"District\\"]"
10    },
11    {
12      "name": "__PBI_TimeIntelligenceEnabled",
13      "value": "1"
14    }
15  ],
16  "culture": "en-US",
```

Navigation controls: -4 +4, ↓ ↑, Diff icons

Line numbers: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16

Annotations: 1, 2, 3, 4