



# MicroHack: Microsoft Fabric Lakehouse End-to-End

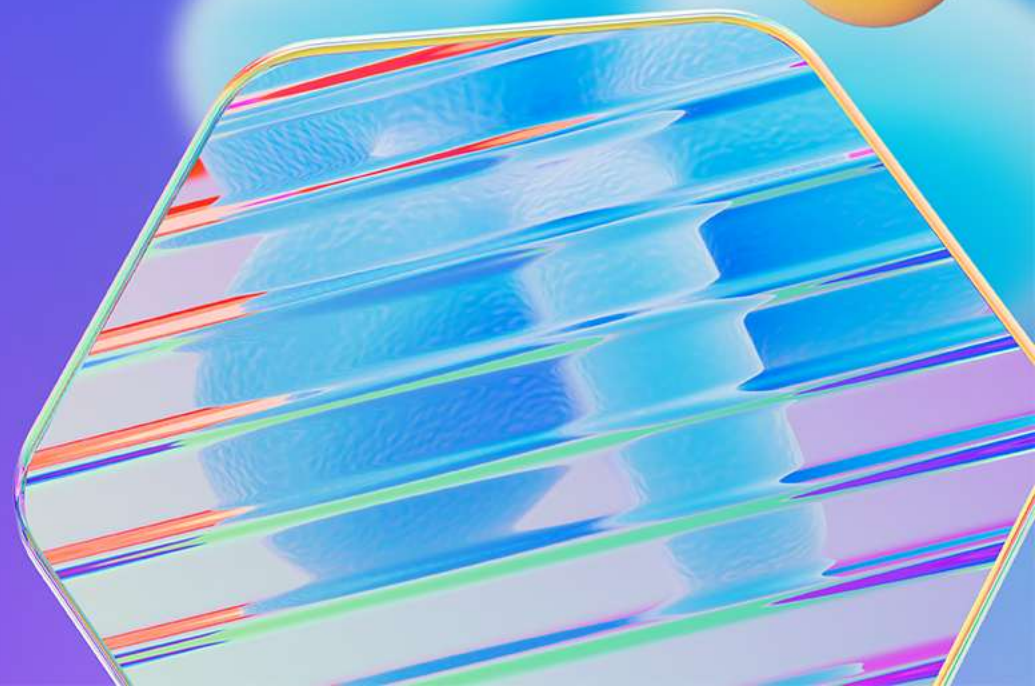
Erik Eckhardt | Filip Slánička



# Agenda

- Introduction to the MicroHack
- Introduction to the Microsoft Fabric Solution
- MicroHack: Microsoft Fabric Lakehouse End 2 End
- Let's Start Hacking

# Introducing to the MicroHack

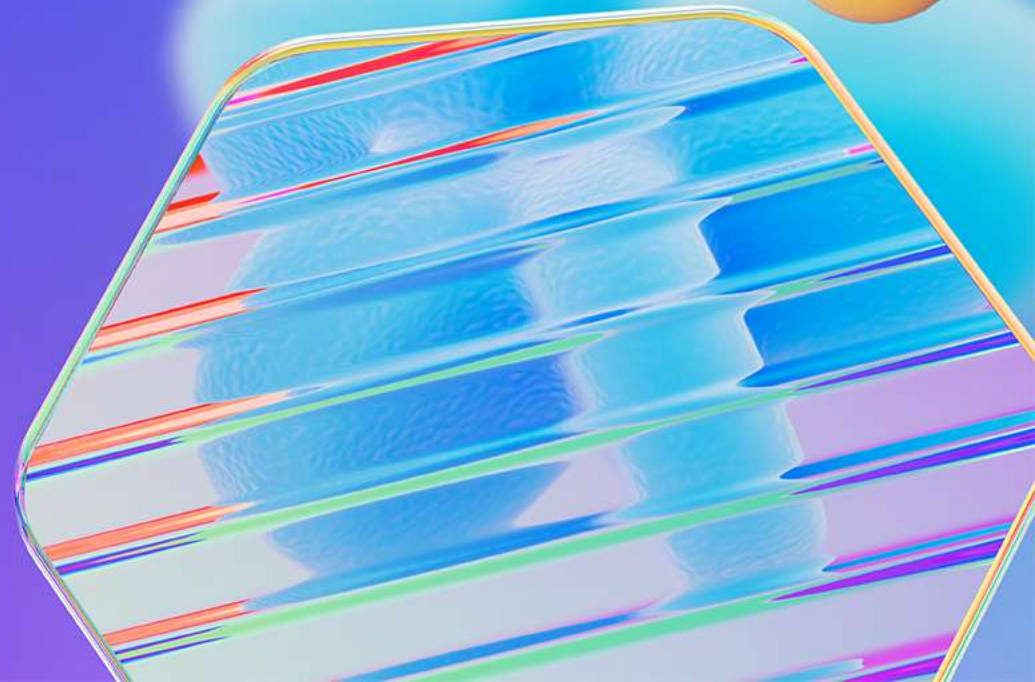


# Introducing to the MicroHack

## Concept

- Challenge-Based Tutorial Format
  - You'll be actively involved in coding, implementation, and problem-solving.
- What to Expect
  - Real-world scenarios and exercises with direct application of concepts in live environments.
- Hands-On Approach
  - Please read the detailed descriptions provided for each challenge.
  - Carefully review the goals, required actions, and success criteria.
- Support and Collaboration
  - Instructors are here to guide you. Feel free to ask questions and support your fellow participants.
  - A complete end-to-end solution guide with screenshots is also available.
- Outcome
  - By the end of this MicroHack, you'll gain practical experience and new skills that you can directly apply to your projects.

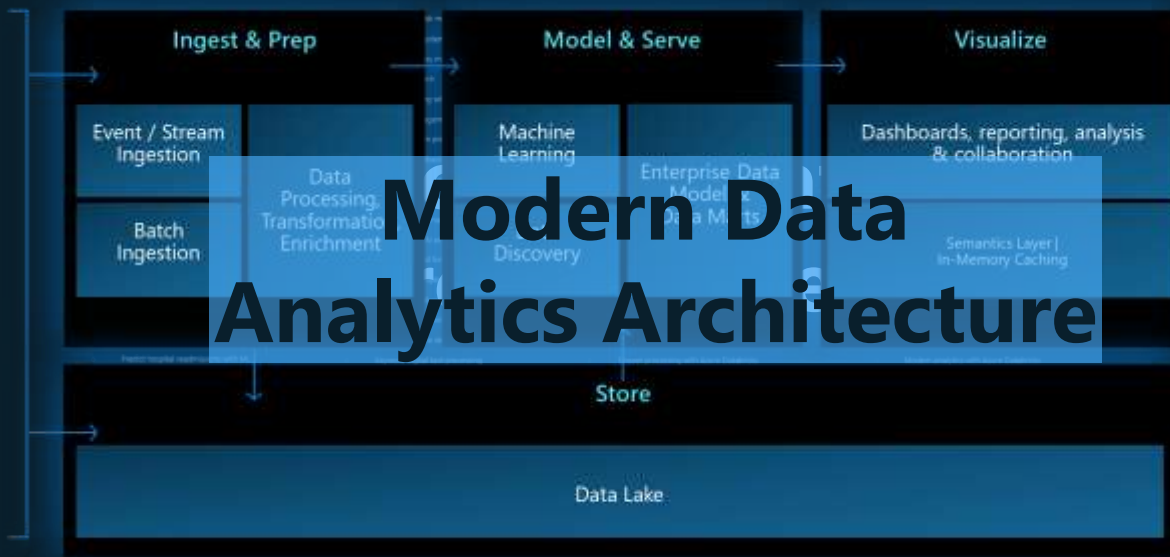
# Introducing the Microsoft Fabric solution





# Microsoft Data Analytics Evolution

## Modern Data Analytics Architecture



## Infrastructure as a Service



## Platform as a Service



## Synapse Analytics as a Single Service



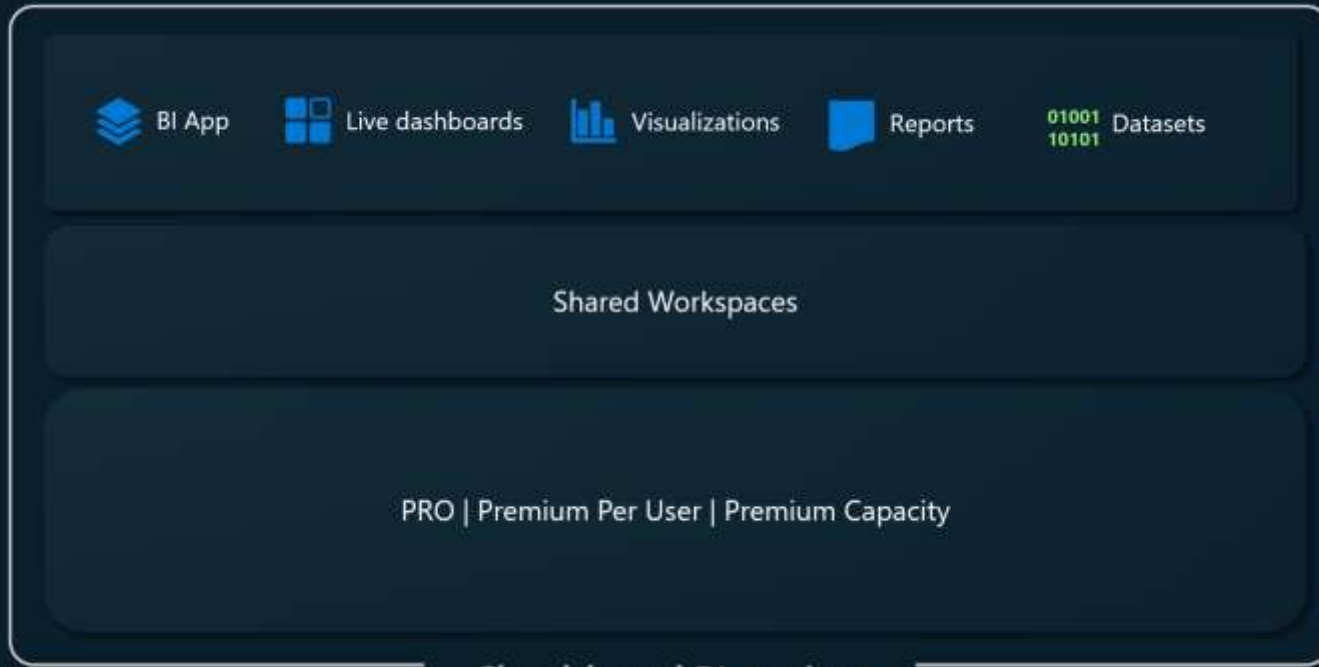


# Microsoft Power BI

+

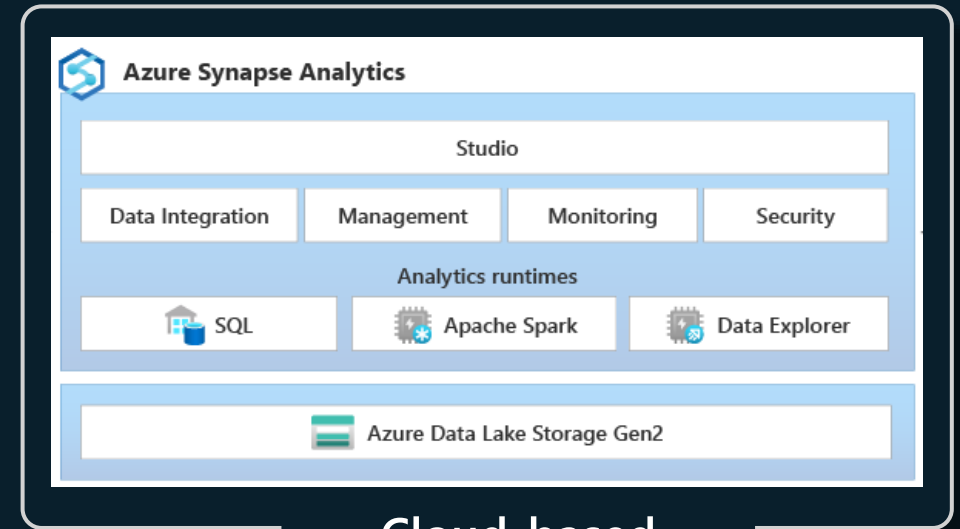


# Azure Synapse Analytics



Cloud-based BI services

+

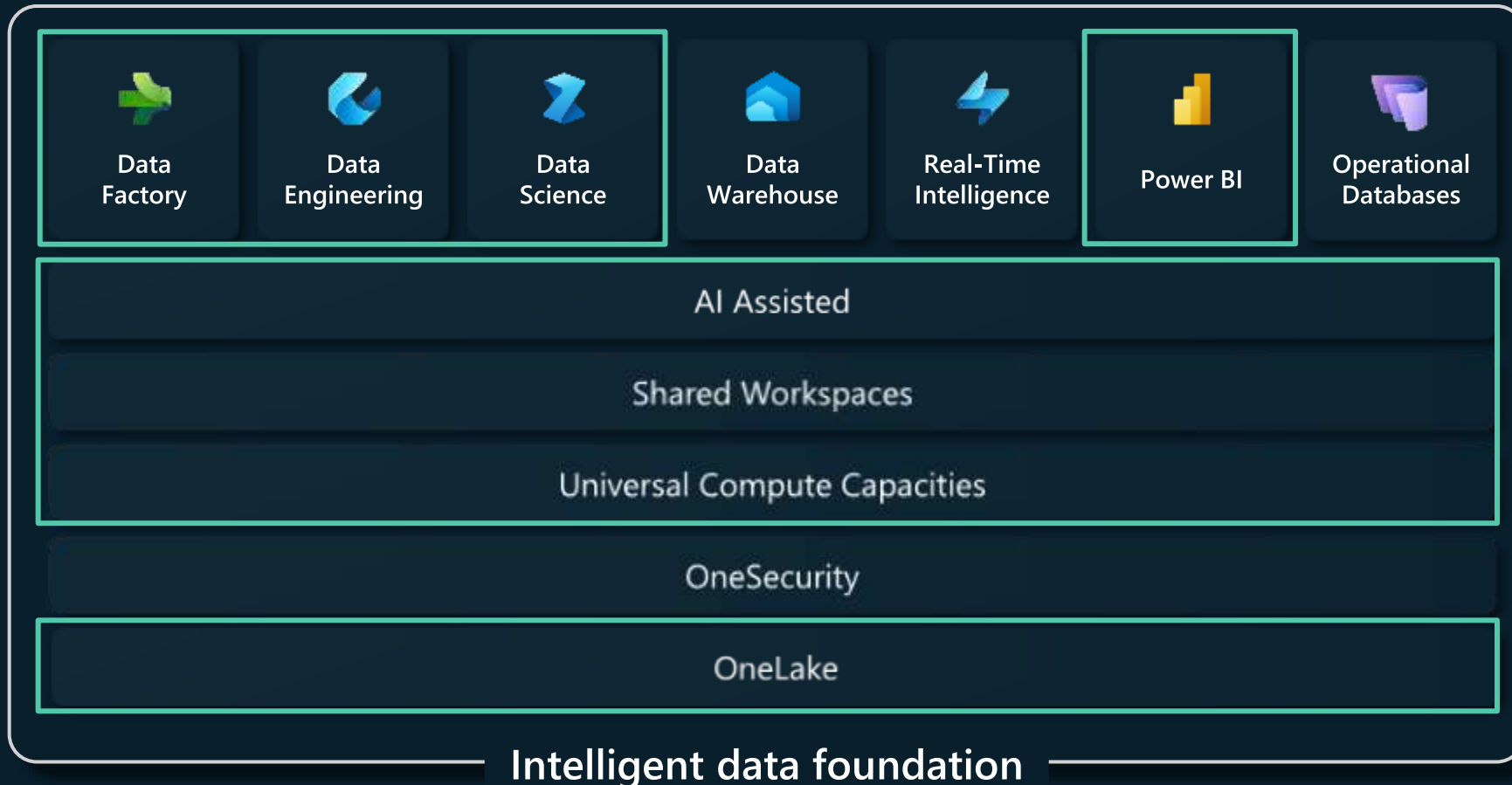


Cloud-based  
Analytics services



# Microsoft Fabric

The unified data platform for the era of AI



## SaaS with single...

- Onboarding and trials
- Sign-on
- Navigation model
- UX model
- Workspace organization
- Collaboration experience
- Data Lake
- Storage format
- Data copy for all engines
- Security model
- CI/CD
- Monitoring hub
- OneLake Catalog
- Governance & compliance
- Domains

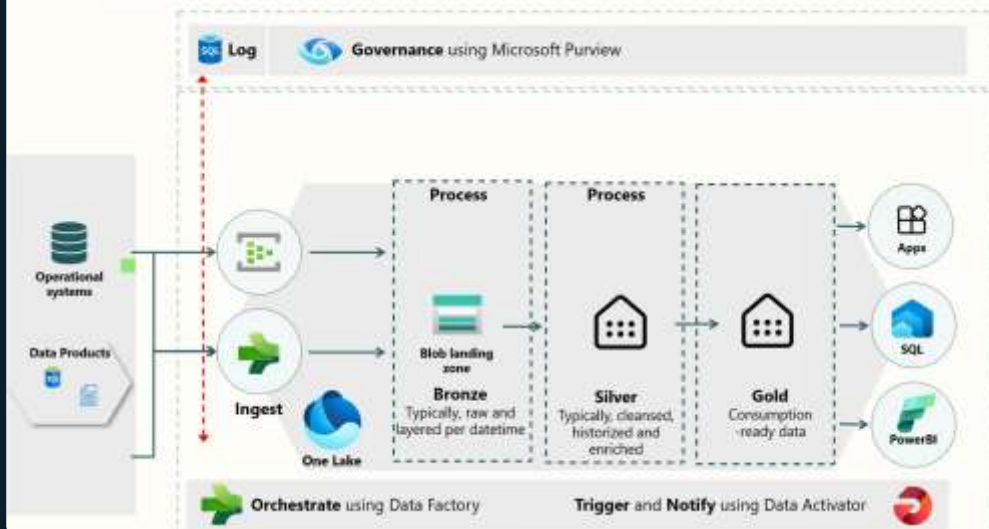




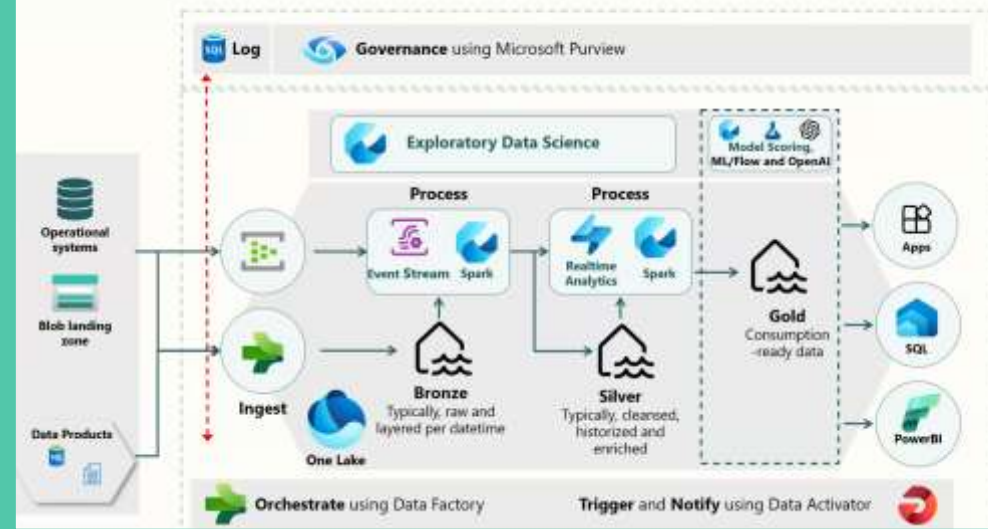
# Microsoft Fabric

## Data Architecture Patterns

The Warehouse Pattern



The Lakehouse Pattern





# Microsoft Fabric

## A lakehouse and a medallion architecture

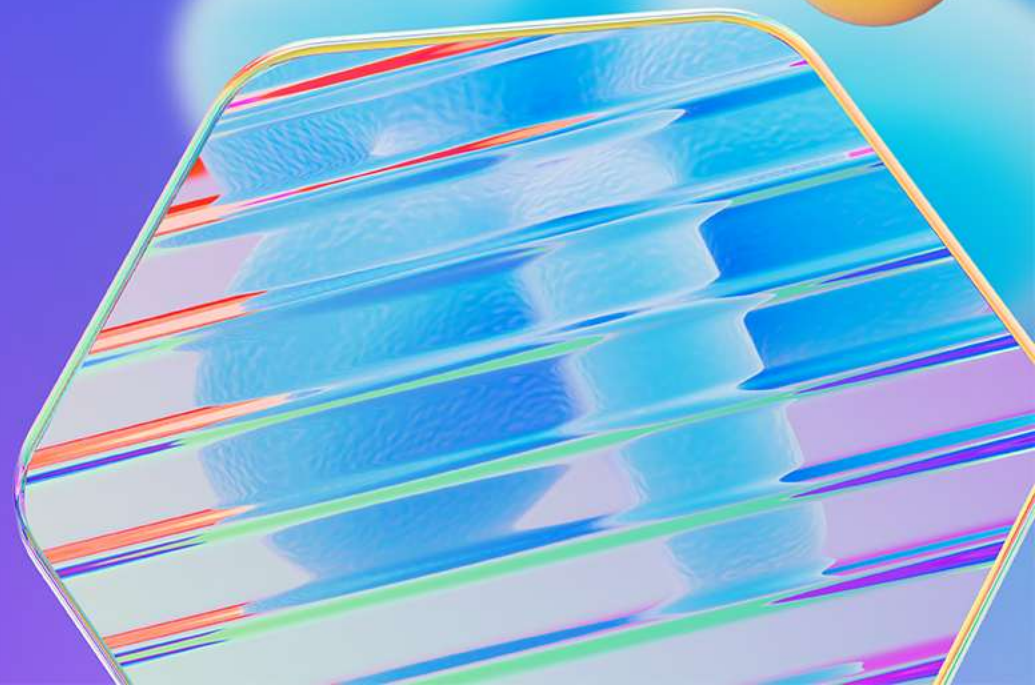


- A lakehouse is an open architecture that combines the best elements of data lakes and data warehouses.
- Lakehouses are enabled by a new system design: implementing similar data structures and data management features to those in a data warehouse directly on top of low-cost cloud storage in open formats with decoupled computes.



- A medallion architecture is a data design pattern used to logically organize data in a lakehouse.
- The goal is to incrementally and progressively improve the structure and quality of data as it flows through each layer of the architecture.
- From a data modeling perspective the Silver Layer has more 3NF write-performant data models, the Gold Layer is more de-normalized and read-optimized data models Kimball style star schema-based or Inmon style Data marts.

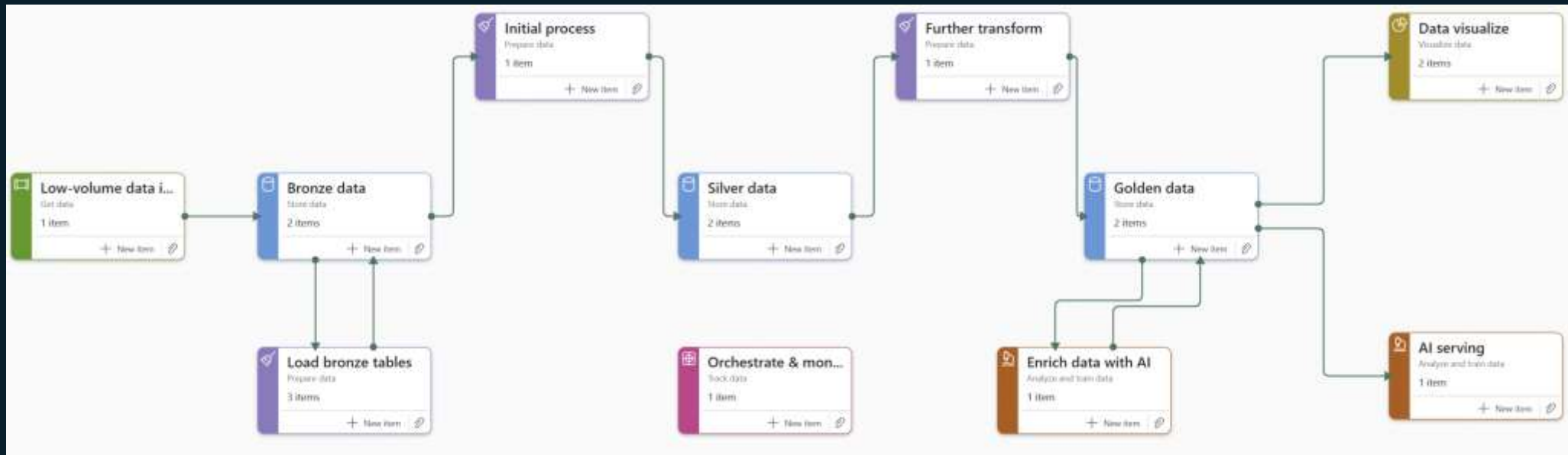
# MicroHack: Microsoft Fabric Lakehouse End-to-End



# MicroHack: Microsoft Fabric Lakehouse End-to-End

## Context & objectives

- Context
  - Learn how to integrate, transform, and manage data in a lakehouse using Fabric workloads such as Data Factory, Data Engineering, Data Science, and Power BI.
- Objectives
  - Develop a complete data workflow: ingestion, preparation, transformation and serving.



# MicroHack: Microsoft Fabric Lakehouse End-to-End Challenges

📌 Important

ch01: Prepare the infrastructure

[Challenge](#) | [Solution](#)

ch02: Ingest *raw* data

[Challenge](#) | [Solution](#)

ch03: Load *bronze* tables

[Challenge](#) | [Solution](#)

ch04: Transform data using notebooks and Spark clusters

[Challenge](#) | [Solution](#)

ch05: Build *gold* tables and a semantic layer for Power BI reporting

[Challenge](#) | [Solution](#)

ch06: Create a master pipeline for orchestration

[Challenge](#) | [Solution](#)

ch07: Process unstructured data with an LLM

[Challenge](#) | [Solution](#)

ch08: Data Agent: Talk with your data

[Challenge](#) | [Solution](#)

Extra exercises

[Solution](#)

- Focus on implementing **at least the first 6 challenges** today as a minimum out of the 8 major challenges.



# MicroHack: Microsoft Fabric Lakehouse End-to-End

## Pre-req. for challenges

### ⚠ Caution

Before starting the challenges, make sure to go through the step-by-step guide for [New Azure User Setup](#).

- You will receive “**XX**” now for your credentials. Please remember it!

### Credentials

Login	Password
UserXX@fabric.microhack.org	dummyHeslo1234

# MicroHack: Microsoft Fabric Lakehouse End-to-End

## Ch01: Prepare the infrastructure for Microsoft Fabric

### Goal

In this challenge, you will prepare the Microsoft Fabric environment for the upcoming challenges. Each participant should get their own Microsoft account for Azure and Office 365. Using your account, create a Fabric capacity and workspace, attach the capacity to your workspace, and create the first lakehouse for the [bronze layer](#) of the [medallion architecture](#).

### Actions

- Deploy MS Fabric Capacity into pre-defined Resource Group "rg-MicroHackFabric" and default Region (West US 3)
  - Name: `yoursurname0fabric` with size F4
  - e.g. `novak0fabric`
- Create Workspace in MS Fabric: YourSurname NY Taxi Analysis [DEV]
  - e.g. `Novak NY Taxi Analysis [DEV]`
- Add your Fabric Capacity to your Workspace.
- Change Spark settings: Use 1 node only
- Create the 1st lakehouse: `bronzerawdata`

### Success Criteria

- MS Fabric Capacity is running.
- Workspace is ready to solve your workloads.
- The 1st Lakehouse and SQL analytics endpoint is deployed and you can connect

The screenshot shows the Microsoft Fabric portal interface. At the top, there's a header with the Microsoft Fabric logo and a search bar. Below the header, there's a navigation pane on the left with icons for Home, Workspace, Capabilities, OneLake catalog, Workspace, Real-time, and Workspace. The main area displays the workspace 'NY Taxi Analysis [DEV]' with a list of resources. The resources are listed in a table with columns: Name, Type, and Task.

Name	Type	Task
bronzerawdata	Lakehouse	—
bronzerawdata	SQL analytics endpoint	—

The screenshot shows the Microsoft Fabric portal interface. At the top, there's a header with the Microsoft Fabric logo and a search bar. Below the header, there's a navigation pane on the left with icons for Home, Workspace, Capabilities, OneLake catalog, Workspace, Real-time, and Workspace. The main area displays a list of resources. The resources are listed in a table with columns: Name, Type, Resource Group, and Location.

Name	Type	Resource Group	Location
eec0fabric	Fabric Capacity	rg-MicroHackFabric	West US 3

# MicroHack: Microsoft Fabric Lakehouse End-to-End

## Ch02: Ingest raw data with data pipelines and shortcuts

### Goal

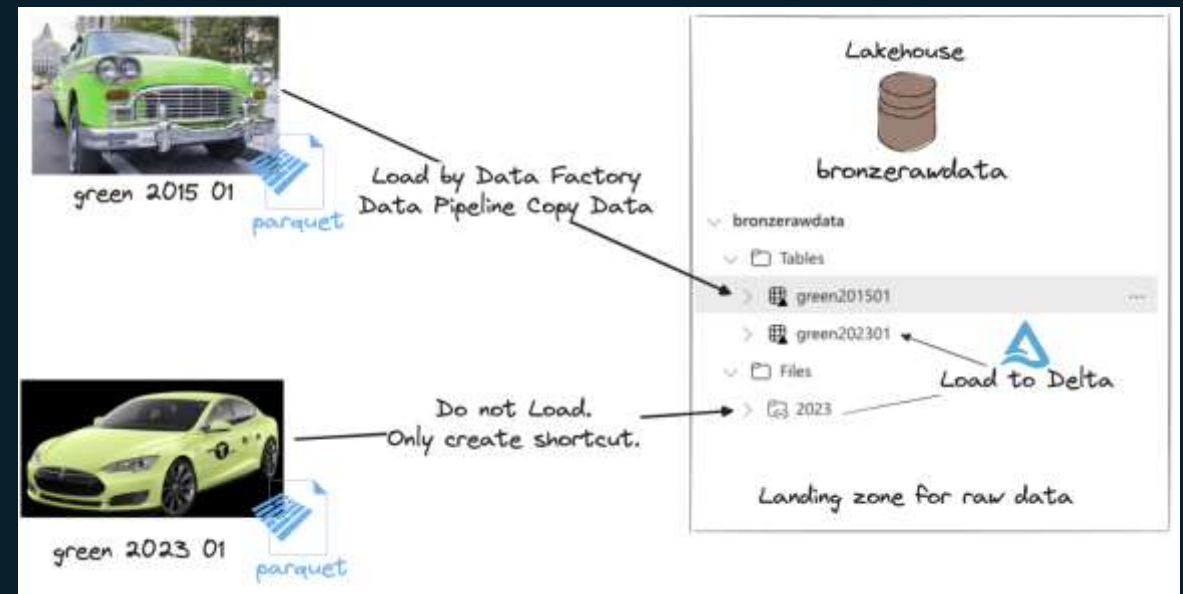
Focus on integrating two data sources from NYC taxi data: one from 2015 and the other from 2023. The table below presents the metaphors we will use to integrate the data:

### Actions

- Create Pipeline: Ingest Green Taxi 2015
  - With Copy Activity to Ingest Data
    - Source: Azure Blob Storage: <https://stfabricmicrohacksw.blob.core.windows.net/>
      - Authentication kind Shared Access Signature (SAS): sv=2024-11-04&ss=b&st=sco&sp=rft&se=2025-11-30T00:36:28Z&st=2025-10-06T15:21:28Z&spr=https&sig=AukZMGIOC7%2BdJC%2BAbRmaNpGcgBjagaYOWlmaSjtiTZw%3D
    - File path:
      - Container: `taxidata`
      - File: `green_tripdata_2015-01.parquet`
    - File format: Binary
  - Destination: Lakehouse `bronzerawdata`
    - File path directory: `Landing/GreenTaxi2015`
    - Format: Binary
- Create Shortcut in your Lakehouse into Files > Landing to 2023 Folder in Azure Data Lake Gen2:
  - <https://edstfabricmicrohacksw.dfs.core.windows.net/>
    - Authentication kind Shared Access Signature (SAS): sv=2024-11-04&ss=b&st=sco&sp=r&se=2025-11-30T00:38:39Z&st=2025-10-06T15:23:39Z&spr=https&sig=HCUs0mDys3JEU4MIOrcmdmc5KNYNNHhUS1Wvkr1Hhc%3D

### Success Criteria

- You have File "green\_tripdata\_2015-01.parquet" in your Lakehouse: Files > Landing > GreenTaxi2015
  - You can see the content of the file
- You have shortcut to Folder "2023" in your Lakehouse with files "NYC-Taxi-Discounts-Per-Day.csv" and "green\_tripdata\_2023-01.parquet"
  - You can see the content of both files



Home

Get data | New semantic model | Open notebook | Manage OneLake data access (preview) | Update all variables

Explore

Search tables

bronzerawdata

Tables

Files

Landing

2023

GreenTaxi2015

Files > Landing > GreenTaxi2015 > green\_tripdata\_2015-01.parquet (preview)

#	VendorID	pass_pickup...	pass_dropo...	trip_start...	trip_end...	trip_start...	trip_end...
1	2	2015-01-01T00...	2015-01-01T00...	N	1	2015-01-01T00...	2015-01-01T00...
2	2	2015-01-01T00...	2015-01-01T00...	N	1	2015-01-01T00...	2015-01-01T00...
3	2	2015-01-01T00...	2015-01-01T00...	N	1	2015-01-01T00...	2015-01-01T00...
4	2	2015-01-01T00...	2015-01-01T00...	N	1	2015-01-01T00...	2015-01-01T00...
5	2	2015-01-01T00...	2015-01-01T00...	N	1	2015-01-01T00...	2015-01-01T00...
6	2	2015-01-01T00...	2015-01-01T00...	N	1	2015-01-01T00...	2015-01-01T00...
7	2	2015-01-01T00...	2015-01-01T00...	N	1	2015-01-01T00...	2015-01-01T00...

# MicroHack: Microsoft Fabric Lakehouse End-to-End

## Ch03: Load bronze tables

**Goal**

In this challenge, you will load three files into three Delta tables in the [bronze layer](#) lakehouse. You will use three different load strategies:

1. ETL Developer: Data Factory Pipeline
2. Citizen Data Developer: Data Factory Dataflow Gen2
3. Data Engineer: Notebook

**Actions**

- ETL Developer: Data Factory Pipeline
  - Create a new pipeline: Load Green Taxi 2015
  - Source:
    - Connection: bronzeraawdata lakehouse
    - Root folder: Files
    - File path: Landing/GreenTaxi2015/green\_tripdata\_2015-01.parquet
    - File format: Parquet
  - Destination:
    - Connection: bronzeraawdata lakehouse
    - Root folder: Tables
    - Table: green201501 (new table)
    - Table action: Overwrite
- Citizen Data Developer: Data Factory Dataflow Gen2
  - Create a new Dataflow Gen2: Load Green Taxi Discounts
  - Source: bronzeraawdata lakehouse
    - File path: Files > Landing > 2023 > NYC-Taxi-Discounts-Per-Day.csv
  - Destination: bronzeraawdata lakehouse
  - Table: greenDiscountsPerDay
- Data Engineer: Notebook
  - Create a new Notebook: Load Green Taxi 2023 or import [challenges/ch03/Load Green Taxi 2023 - Ch03.ipynb](#)
  - Remove origin lakehouses from Notebook and Add/attach your bronzeraawdata (as default) lakehouse
  - Source: Files/Landing/2023/green\_tripdata\_2023-01.parquet
  - Target: Tables/green202301
    - Format: delta
    - Mode: overwrite

• Check # of rows per delta tables

**Success Criteria**

- Table green201501 exists with 1 506 493 rows
- Table green202301 exists with 68 231 rows
- Table greenDiscountsPerDay exists with 2 rows

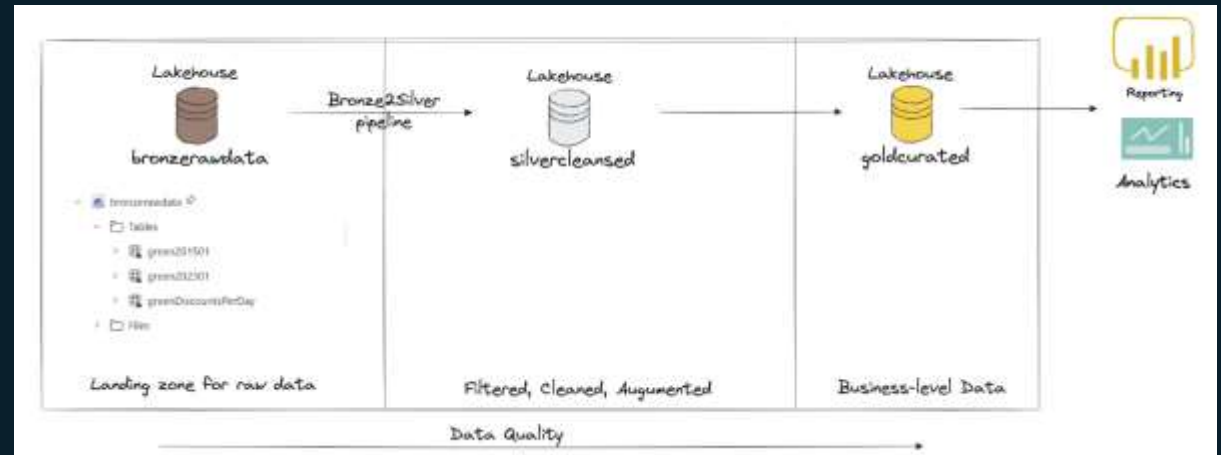
**Explorer**

2023

```
1 df_green2015 = spark.read.format("parquet").load("Tables/green201501")
2 df_green2023 = spark.read.format("parquet").load("Tables/green202301")
3 df_discount = spark.read.format("csv").load("Tables/greenDiscountsPerDay")
4
5 print(df_green2015.rdd.count()) # df_green2015.count()
6 print(df_green2023.rdd.count()) # df_green2023.count()
7 print(df_discount.rdd.count()) # df_discount.count()
```

11 | Spark job (0 of 0 succeeded) | [Refresh](#)

green2015 row count: 1506493  
green2023 row count: 68231  
greenDiscountsPerDay row count: 2



# MicroHack: Microsoft Fabric Lakehouse End-to-End

## Ch04: Transform data using notebooks and Spark (silver tables)

### Goal

In this challenge, you will explore data engineering tasks aimed at transforming raw data into a refined [silver layer](#).

By the end of the challenge, you will have completed the second step, the silver layer, of the [medallion architecture](#):

### Actions

- Create the 2nd lakehouse: silvercleansed
- Import [challenges/ch04/Transform bronze to silver - Ch04.ipynb](#) to your workspace and try to complete all Goals / Actions mentioned in notebook
  - Remove origin lakehouses from Notebook and Add/attach your `bronzerawdata` (as default) lakehouse and `silvercleansed` lakehouse

### Success Criteria

- All success criteria mentioned in the notebook are met
- 6 tables with data in silvercleansed lakehouse exist

Explore

Data Items | Permissions

+ Add data items

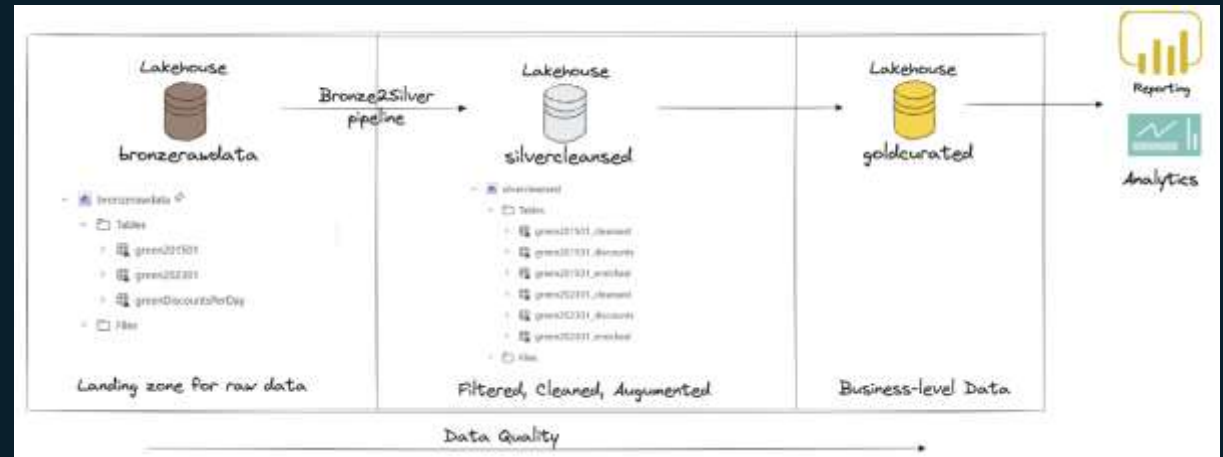
Items

- bronzerawdata
- silvercleansed
  - Tables
    - green201001\_cleaned
    - green201001\_electronics
    - green201001\_enriched
    - green202001\_cleaned
    - green202001\_fraudulent
    - green202001\_revised
  - Files

Table view

1478971 records

ID	VendorID	trip_start_datetime	trip_end_datetime	trip_distance	passenger_count	trip_duration	trip_rate
1	2	2015-01-01 00:01:10	2015-01-01 00:02:10	0.1	1	10	10.0
2	2	2015-01-01 00:02:10	2015-01-01 00:03:10	0.1	1	10	10.0
3	2	2015-01-01 00:03:10	2015-01-01 00:04:10	0.1	1	10	10.0
4	2	2015-01-01 00:04:10	2015-01-01 00:05:10	0.1	1	10	10.0
5	2	2015-01-01 00:05:10	2015-01-01 00:06:10	0.1	1	10	10.0
6	2	2015-01-01 00:06:10	2015-01-01 00:07:10	0.1	1	10	10.0
7	1	2015-01-01 00:07:10	2015-01-01 00:08:10	0.1	1	10	10.0
8	1	2015-01-01 00:08:10	2015-01-01 00:09:10	0.1	1	10	10.0
9	1	2015-01-01 00:09:10	2015-01-01 00:10:10	0.1	1	10	10.0
10	1	2015-01-01 00:10:10	2015-01-01 00:11:10	0.1	1	10	10.0
11	1	2015-01-01 00:11:10	2015-01-01 00:12:10	0.1	1	10	10.0
12	2	2015-01-01 00:12:10	2015-01-01 00:13:10	0.1	1	10	10.0
13	2	2015-01-01 00:13:10	2015-01-01 00:14:10	0.1	1	10	10.0
14	2	2015-01-01 00:14:10	2015-01-01 00:15:10	0.1	1	10	10.0





# MicroHack: Microsoft Fabric Lakehouse End-to-End

## Ch05: Build gold tables and a semantic layer for Power BI reporting

### Goal

In this challenge, you will aggregate augmented data from a [silver layer](#) into a curated [gold layer](#) for advanced business analysts. Next, you will define semantic layer for end users and create the first Power BI report.

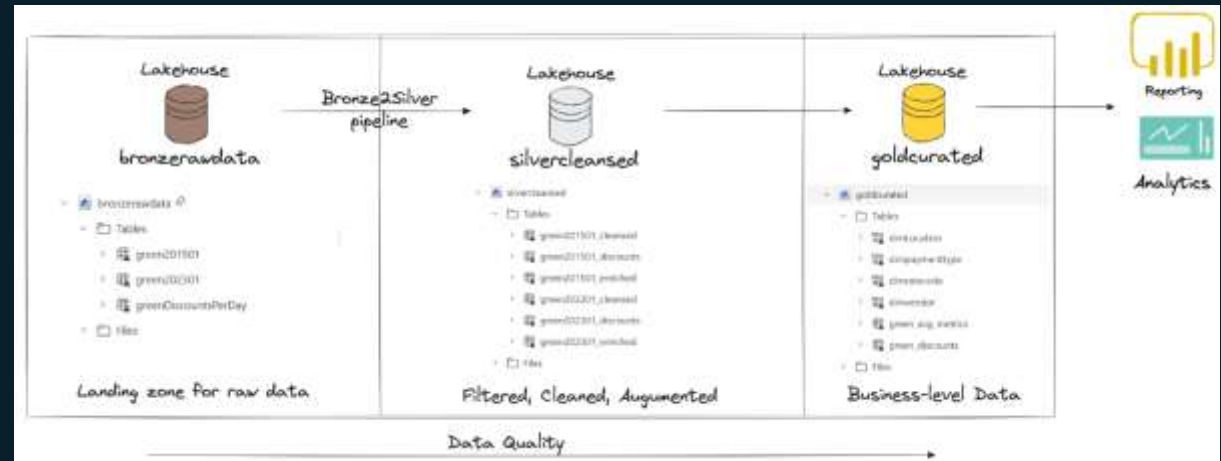
By the end of the challenge, you will have completed the implementation of the [medallion architecture](#) with semantics layer and reports:

### Actions

- Create the 3rd lakehouse: **goldcurated**
  - Create Shortcut in your Lakehouse into **Tables** to pre-created delta tables: **dimLocations**, **dimpaymenttype**, **dimratecode**, **dimvendor** in Azure Data Lake Gen2: <https://adfstabricmicrohackswdfs.core.windows.net/>
    - SAS: sv=2024-11-04&ss=b&sr=sco&sp=r&se=2025-11-30T00:38:39Z&st=2025-10-06T15:23:39Z&spr=https&sig=HCusDmQys3EU4MIOrcmdmc5KNYNNHxhUS1WwR1hRc%3D
  - Import [challenges/ch05/Transform silver to gold - Ch05.ipynb](#) to your workspace and try to complete all Goals / Actions mentioned in notebook
    - Remove origin lakehouses from Notebook and Add/attach your **silvercleansed** (as default) lakehouse and **goldcurated** lakehouse
  - Build Semantics model **GreenTaxi** based on 4 dimensions and 2 facts tables with Avg. function and relations
    - **dimLocations**.LocationID 1:N **green\_discount**.PULocationID
    - **dimpaymenttype**.payment\_type 1:N **green\_discount**.payment\_type
    - **dimratecode**.RatecodeID 1:N **green\_discount**.RateCodeID
    - **dimvendor**.VendorID 1:N **green\_discount**.VendorID
  - **dimLocations**.LocationID 1:N **green\_avg\_metrics**.PULocationID
  - **dimvendor**.VendorID 1:N **greengreen\_avg\_metrics**.VendorID
- Create sample report based on **GreenTaxi** Semantic model with
  - 1 Clustered bar chart, 1 Line chart, 1 Azure maps, 1 Narrative and 1 Slicer
- Create SQL View **vGetAverageFaresDiscounts** based on "Lakehouse SQL analytics endpoint" for business analysts
  - Get the distinct passenger count, total fares, average fares and discount grouped by the passenger count

### Success Criteria

- All success criteria mentioned in the notebook are met
- 6 tables with data in **goldcurated** lakehouse exist
- Semantics model **GreenTaxi** with 2 Fact tables and 4 Dimensions exists
- Report based on **GreenTaxi** Semantics model exist
- SQL View **vGetAverageFaresDiscounts** returns 11 records



# MicroHack: Microsoft Fabric Lakehouse End-to-End

## Ch06: Create a master pipeline for orchestration

### Goal

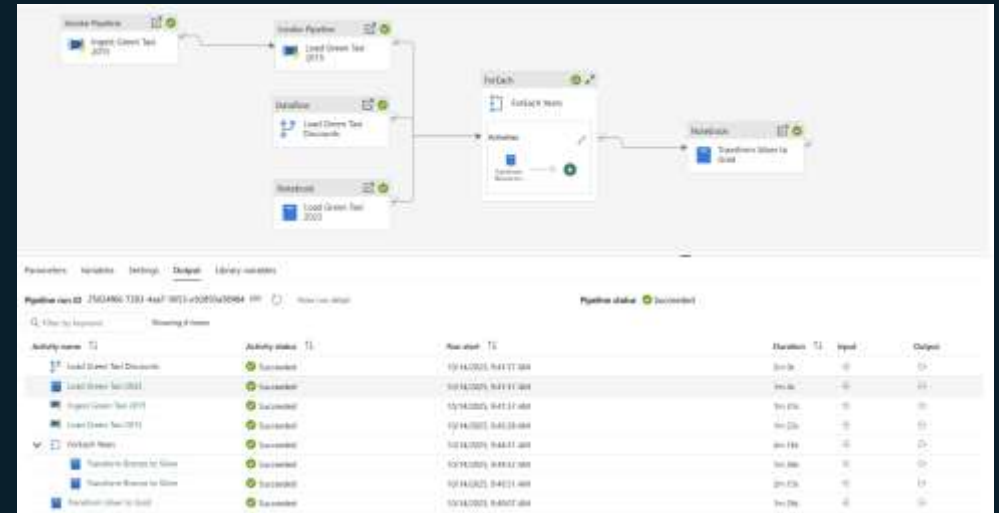
Now, let's focus on orchestration. Considering we have only two tables ingested by two pipelines and processed by three notebooks, imagine a scenario where you need to ingest and process tens or hundreds of tables, files, or data sources and manage dependencies between them. In such cases, the most efficient approach is to build and prioritize a data pipeline for the orchestration workflow.

### Actions

- Create master Orchestration Pipeline
  - Add pipeline parameter: Year (Array, Default value = ["2015", "2023"])
  - Build following workflow:
    - Run in parallel:
      - Invoke pipeline: Ingest Green Taxi 2015
        - On Success, run: Invoke Load Green Taxi 2015
      - Run Dataflow: Load Green Taxi Discount
      - Run Notebook: Load Green Taxi 2023
    - On Success, run: ForEach activity: Sequential | Parallel for @pipeline().parameters.Year
      - Run Notebook: Transform Bronze to Silver
        - Add Base parameter p\_year binded to ForEach items.
    - On Success: Run Notebook: Transform Silver to Gold
- Run pipeline

### Success Criteria

- Pipeline finishes successfully



# MicroHack: Microsoft Fabric Lakehouse End-to-End

## Ch07: Process unstructured data with an LLM

### Goal

In this challenge, you will focus on processing unstructured data from real user reviews of locations in New York.

With Microsoft Fabric, all business professionals—data engineers, data scientists, developers, and analysts—can gain more value from their data through Generative AI.

There are two options in Microsoft Fabric today:

1. Use [AI services](#) with prebuilt and customizable APIs and models, but you need to develop code to process the output into the final schema:
  - Language Models: gpt-5, gpt-4.1, gpt-4.1-mini, and gpt-4o-mini
  - APIs: Text Analytics, and Azure AI Translator
2. Use the new set of [AI functions](#) for data engineering, which is much easier. Fabric users can take advantage of industry-leading large language models (LLMs) to transform and enrich data with just a single line of code.

By the end of this challenge, you will have a new dataset with sentiment, classification, summarization, translation, key topics, and location extracted from real user reviews.

### Tip

It's up to you which option you choose, but this challenge is designed for **AI functions**.

### Actions

- You can create new lakehouse for Sandboxing or use **goldcurated**
- Upload file [NY\\_places\\_customer\\_reviews.csv](#) to your lakehouse (Files)
- Import [challenges/ch07/User reviews processing with LLM - AI Functions - Ch07.ipynb](#) to your workspace and try to complete all Goals / Actions mentioned in that notebook
  - Remove origin lakehouses from Notebook and Add/attach your **goldcurated** (as default) lakehouse

### Success Criteria

- All success criteria mentioned in the notebook are met
- Table **goldcurated.reviewLocations** exists with 265 rows and AI columns

```
1 # TRANSLATE input text to Czech language
2
3 dfp["AItranslationsReview"] = dfp["Review"].ai.translate("Czech")
4 dfp["AItranslationsSummarize"] = dfp["AIsummarize"].ai.translate("Czech")
5 display(dfp)
6
```

✓ - Command executed in 18 sec 892 ms by System Administrator on 2/20/25 PM 10/24/25

100% 265/265 [00:07:00:00, 36.36it/s]  
100% 265/265 [00:00:00:00, 39.93it/s]

Table + New chart

Table view

ID	LocationID	Place	Review	User	Sentiment	AIclassify	AIsummarize	AItranslationsReview	AItranslationsSummarize
1	1	New York ...	Tamable if yo...	Ecocodula	negative	historic building	User describes a ...	Prostě, pokud musíte m...	Uživatel popisuje velk...
2	2	New York ...	Even in centr...	HubertD	positive	accommodation	In central NYC, yo...	I v centru New Yorku st...	V centru New Yorku může...
3	3	New York ...	Pelham Gard...	Joe Cosenti...	positive	accommodation	Pelham Gardens L...	Pelham Gardens je zbyte...	Pelham Gardens je centrálně ...
4	4	New York ...	Alphabet Cit...	Lulu H	positive	bars	Alphabet City is a ...	Alphabet City působí jako...	Alphabet City je skvělá čt...
5	5	New York ...	Arden High...	Anonymous	positive	accommodation	Arden Heights is ...	Arden Heights je domov...	Arden Heights je předměst...
6	6	New York ...	Beautiful vie...	Michael S.	mixed	historic building	Beautiful views an...	Krásné výhledy a najítr...	Krásné výhledy a najítr...
7	7	New York ...	Astoria is a s...	Current Re...	positive	restaurants	Astoria is a safe ...	Astoria je velmi pěkná čt...	Astoria je bezpečná, dobře ...

```
3 df = spark.sql("SELECT * FROM goldcurated.reviewLocations LIMIT 1000")
4 display(df)
```

✓ - Command executed in 20 sec 857 ms by System Administrator on 2/12/25 PM 12/16/25

Table + New chart

Table view

ID	LocationID	Place	Review	User	Sentiment	AIclassify	AIsummarize	AItranslationsReview	AItranslationsSummarize	AItranslationsTopic	AItranslationsLocation
1	100	New York ...	Historic build...	Anonymous	neutral	historic building	Historic build in NYC...	Historic build in New York...	Historic build in New York...	Historic build, NYC, Queens	Queens, New York, NY...
2	100	New York ...	Beautiful view...	Anonymous	positive	historic building	Beautiful view of the...	Historic build in New York...	Historic build in New York...	Historic build, NYC, Queens	Queens, New York, NY...

# MicroHack: Microsoft Fabric Lakehouse End-to-End

## Ch08: Data Agent: Talk with your data

### Goal

In this challenge, you will focus on the Data Agent, which allows you to build your own conversational Q&A systems using generative AI. The Fabric Data Agent uses LLMs to help users interact with their data naturally. It leverages Azure OpenAI Assistant APIs and behaves like an agent: processing user questions, determining the most relevant data source (Lakehouse, Warehouse, Power BI dataset, KQL databases), and invoking the appropriate tool to generate, validate, and execute queries. This way, even people without technical expertise in AI or a deep understanding of the data structure can receive precise and context-rich answers.

During this challenge, you will define a new Data Agent and add specific instructions, examples, and guidance to fine-tune the Fabric Data Agent. This ensures that responses align with your needs and goals, allowing everyone to engage with data more effectively.

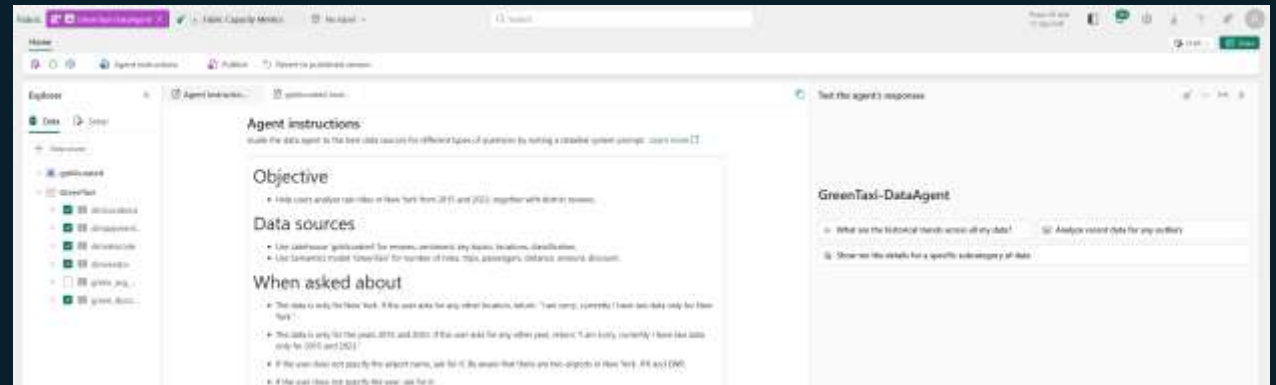
By the end of this challenge, the Data Agent will be ready to be consumed via the Python SDK, Azure AI Foundry, Microsoft Copilot Studio, and Copilot in Power BI.

### Actions

- Create new Data Agent.
- Add Data sources:
  - Semantic model: GreenTaxi
    - Tables:
      - dimLocations
      - dimpaymenttype
      - dimratecode
      - dimvendor
      - green\_discount
  - Lakehouse: goldcurated
    - Tables:
      - dimLocations
      - reviewLocations
- Test questions:
  - What were the most common pickup locations for green taxi rides in 2023?
  - On which weekday do people travel from the airport the most?
  - How many rides were in Boston?
  - How many rides were there in 2025?
  - Show me the top 3 districts in New York with the best accommodation.
  - Show me the districts where the most passengers moved from.
  - Show me the districts where most passengers moved from, and what reviews, sentiments, and key topics you know about these districts.
- Tune Data Agent via Agent instructions (System Prompt) and Data source instructions for above questions.
- Publish Data Agent.
- Consume Data Agent via Copilot (Home) in Power BI or Python SDK.

### Success Criteria

- Data Agent exists and is able to answer questions about data in Semantic model and Lakehouse.
- Data Agent is available via Python SDK or Copilot (Home) in Power BI.
- Data Agent can answer questions:
  - Question: What were the most common pickup locations for green taxi rides in 2023?
    - Answer: East Harlem North; East Harlem South; Forest Hills; Central Harlem; ...





# Let's Start Hacking



MicroHack-Fabric GitHub Repo:  
<https://github.com/CZSK-MicroHacks/MicroHack-Fabric>



