Ventagium
Data Consulting

Fabric Workshop

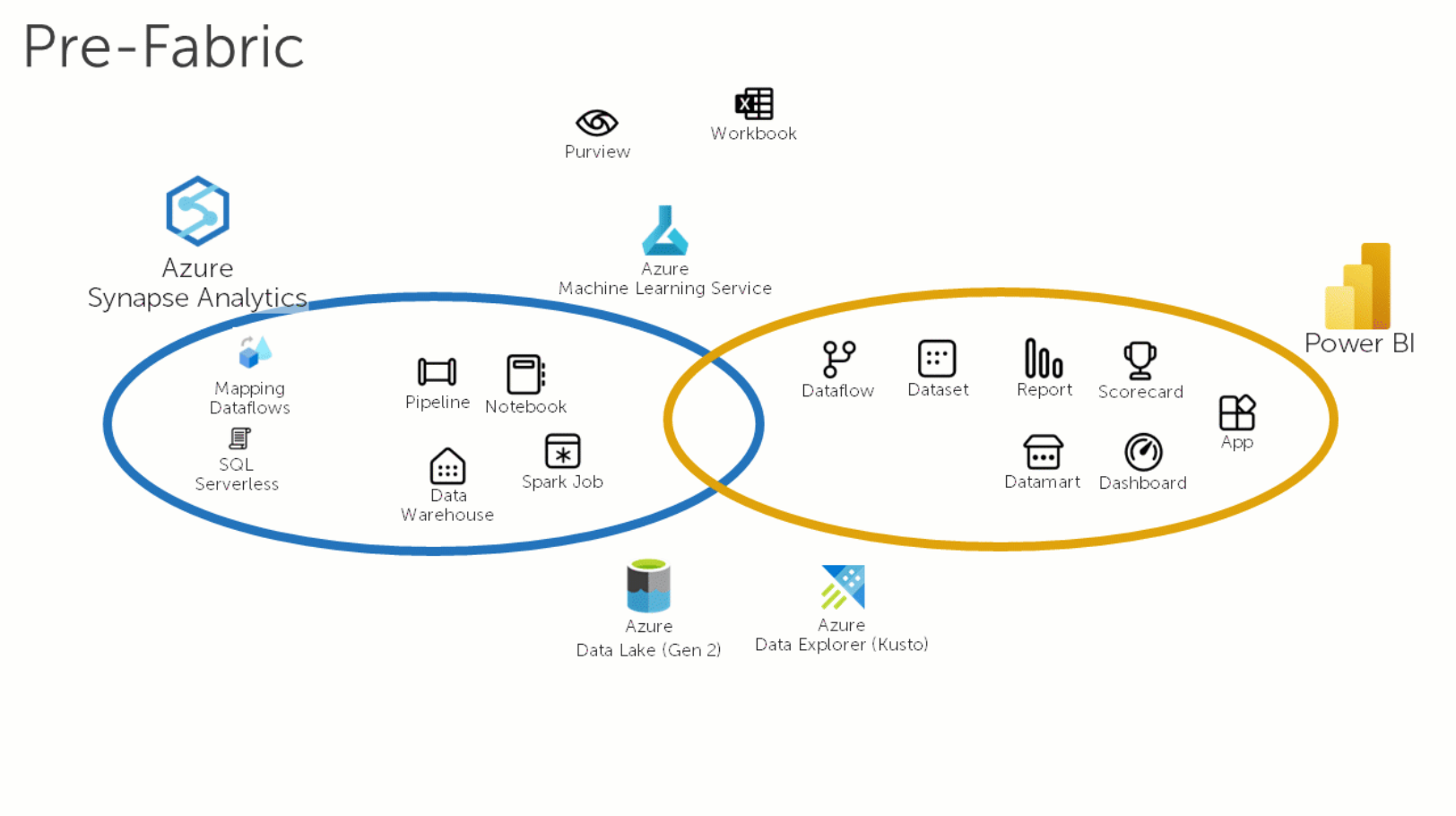
2025-02-27

Contents

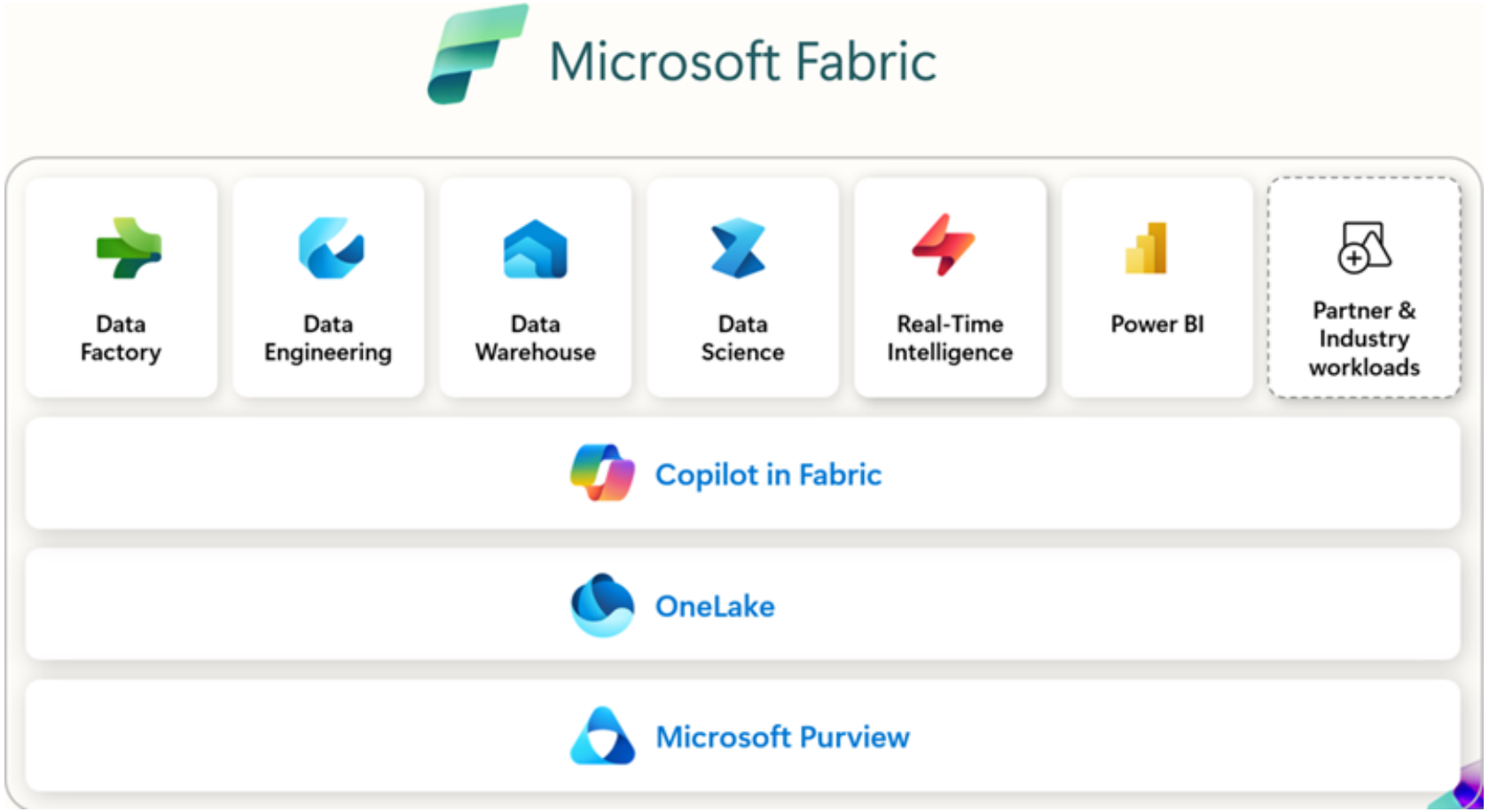
Contents

- Day 1
  - Ventagium + Leads Intro
  - Fabric Intro
  - Medallion Architecture
  - SQL Mirroring
  - Pipelines with Copy Job (Preview)
  - Shortcuts (External ADLS and Internal)
- Day 2 - Part 1
  - Real-time Intelligence (Theory)
  - Notebooks and Spark
  - CI Azure DevOps GIT
  - CD Deployment Pipelines
  - AI Skills
- Day 2 - Part 2
  - Fabric Databases
  - Creating a Power BI Report with the help of Copilot
  - Industry Solutions (Theory)
  - Data Science (only the theory on Experiments, ML Models)
  - OpenAI (Theory)

Before and After



The Toolbox


Microsoft Fabric
Data Factory
Data Engineering
Data Warehouse
Data Science
Real-Time Intelligence
Power BI
Partner & Industry workloads
Copilot in Fabric
OneLake
Microsoft Purview

The Problem

The Solution

Back to basics:

- Standards
- Governance
- Frameworks
- Architecture
- Best practices

Medallion Architecture

Medallion Architecture – Intro

Medallion Architecture – Changes

Raw Data to Bronze ETL:

- *It's the first medallion layer used for getting the data into the Bronze Lakehouse in Delta format.*
- Exception: reducing the number of columns brought from the source or changing data types to reduce load.

Bronze to Silver:

- *Second medallion layer used for cleansing and standardising data sourced from the Bronze Layer. The result is usually an 'enterprise view' of the data structured as tables or business-like entities such as, employee, transaction, customer, organisation or product. Here we start getting Dims and Facts.*
- Standardizing Column Names
- Enforcing schema
- Engineering Features (e.g., concatenating of columns into one, adding calculated columns, etc.)
- Filtering Rows
- Filtering Columns
- Deduplicating Data
- Merging/Appending*
- Enforcing Keys

Silver to Gold:

- *Third medallion layer, it usually contains denormalised, read-optimised data models (fact and dimension tables) conforming to a Kimball-style star schema design. It also serves as a presentation layer for Power BI Direct Lake connectivity to the data models.*
- Business Function Specific Features
- Creation of Views that Join Tables and the Columns that need data from multiple tables.
- SQL Type Constraints (currently only NOT NULLS)
- Features created by Machine Learning Models

Note: designing a warehouse is 80% science, 20% art.

Medallion Architecture – Tools of Choice

Raw to Bronze

1. *Shortcut* when source location available (Fabric, AWS, Azure Datalake, GCS)
2. *Copy Data when source location is available*
3. *Notebooks* when data engineering is required.
4. Dataflows

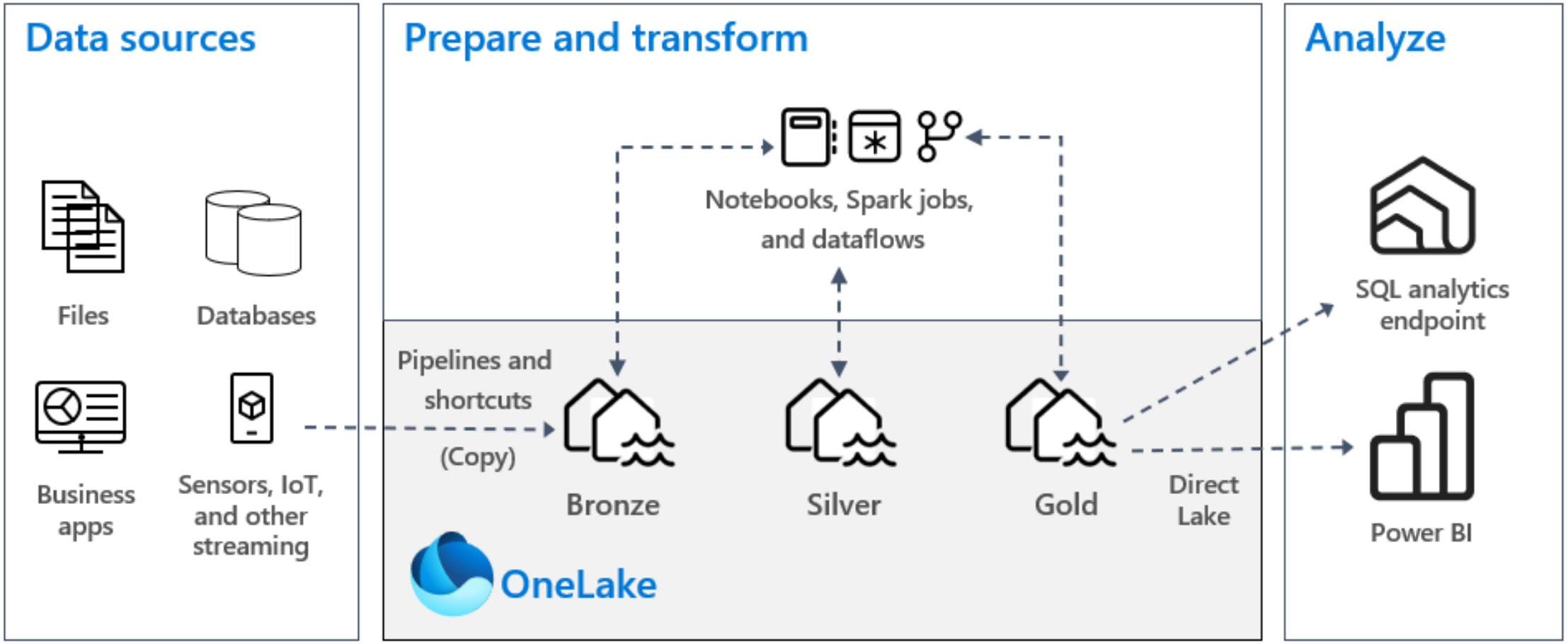
Bronze to Silver

1. *Shortcut* when no changes are needed
2. *Notebooks* when Feature Engineering is required, complex row cleansing, etc.
3. *Copy Data* when there is only need to choose columns or column names changes.
4. Dataflows but their performance is not optimized.

Silver to Gold

1. *Shortcuts*
2. *Notebooks*
3. Copy Data
4. Other options include Dataflows but their performance is not optimized.

Client Demo


Data sources
Files
Databases
Business apps
Sensors, IoT, and other streaming
Prepare and transform
Notebooks, Spark jobs, and dataflows
Pipelines and shortcuts (Copy)
Bronze
Silver
Gold
OneLake
Direct Lake
Analyze
SQL analytics endpoint
Power BI

Exercise

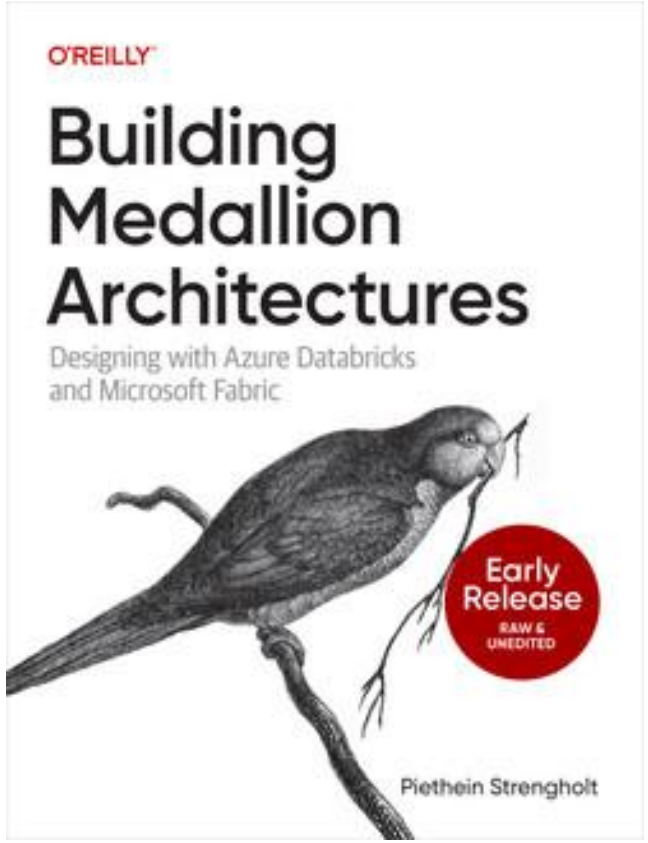
1. Create Workspaces (Bronze, Silver and Gold)
2. Create Pipeline
3. Create Copy Activities from Raw to Bronze
4. Create Copy Activity from Bronze to Silver
5. Create Copy Activity from Silver to Gold
6. Create Pipeline Orchestrator
7. Create Notifications

Best Practices

- Create a Pipeline per Schedule:
  - Pipeline – 5 am
  - Pipeline – Sunday 6 am
  - Pipeline – Every Working Hour
- Create a Pipeline Orchestrator to Handle Errors and Triggers
- Ensure Timeouts make sense (the default is 12 hours!)
- Create a Workspace per Medallion to handle Permissions

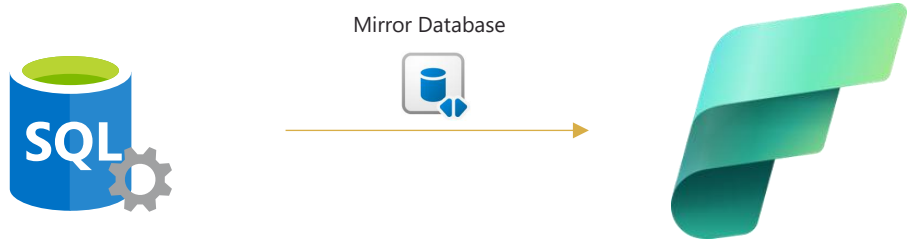
References

- [Implement medallion lakehouse architecture in Fabric - Microsoft Fabric | Microsoft Learn]
- [What is a Medallion Architecture?]
- Book: Building Medallion Architectures (Coming out in April 2025)



Mirroring

Integrate your Azure SQL Database to Fabric, avoiding ETL operations while keeping your data near real-time.



What is Mirroring?

Solution bring existing data from external sources into Fabric´s One Lake. It is characterized for offering near real-time data replication with no ETL steps.

As data is now in Fabric´s One Lake, it is available for other Fabric services such as notebooks or Power BI.

Shortly, you could say it is a synchronized copy of your data

Platform	Near real-time replication	Type of mirroring	End-to-end tutorial
Microsoft Fabric mirrored databases from Azure Cosmos DB (preview)	Yes	Database mirroring	Tutorial: Azure Cosmos DB
Microsoft Fabric mirrored databases from Azure Databricks (preview)	Yes	Metadata mirroring	Tutorial: Azure Databricks
Microsoft Fabric mirrored databases from Azure SQL Database	Yes	Database mirroring	Tutorial: Azure SQL Database
Microsoft Fabric mirrored databases from Azure SQL Managed Instance (preview)	Yes	Database mirroring	Tutorial: Azure SQL Managed Instance
Microsoft Fabric mirrored databases from Snowflake	Yes	Database mirroring	Tutorial: Snowflake
Open mirrored databases (preview)	Yes	Open mirroring	Tutorial: Open mirroring
Microsoft Fabric mirrored databases from Fabric SQL database (preview)	Yes	Database mirroring	Automatically configured

Azure SQL Database Mirroring

Continuous replication of an existing Azure SQL Database into Fabric´s One Lake. This is done through the conversion of data to Parquet.

Enables a **read-only SQL Analytics Endpoint**. Therefore, changing tables in Fabric, will not alter the Azure SQL Database.

It is an **incremental replication**.

How Does it Work?

Mirroring reads data from the transaction log and write data to Fabric One Lake:

1. Change Data Capture (CDC): Mirroring in Fabric leverages SQL's Change Data Capture (CDC) stack. CDC captures changes made to the database and stores them locally in the transaction log.

2. Replication: The mirroring process reads the data from the transaction log and sends the change data to OneLake storage, ensuring that data is up to date.

3. Delta Tables: The change data from the transaction log is transformed into delta tables in OneLake. These delta tables are optimized for analytics and can be queried without impacting the performance of the primary database.

Azure SQL Database Requirements

Not supported

- For Azure SQL Databases behind an Azure Virtual Network or Private Networking (Gateway Access is not supported).

Server Requirements

- Firewall Rules must be in “Allow public network access” configuration.
- Enable “Allow Azure services” option for the desired Azure SQL Database.

Database Requirements

- Single Database or Elastic Pool Database.
- All Service Tiers in Vcore pricing model are allowed.
- For DTU pricing model, the database must be at least 100 DTUs.
- Views, transient and external tables not supported.
- System Assigned Managed Identity (SAMI) Enabled

Azure SQL Tables Requirements

Not supported

- Calculated Columns: They will not be replicated to Fabric
- For Primary Key Constraints, the following column types are not defined: computed types, user-defined types, geometry, geography, hierarchy ID, SQL variant, timestamp, datetime2(7), datetimeoffset(7) or time(7).

Fabric Requirements

Fabric Capacity must be active and running. Available for **every Fabric SKU.**

The following **tenant settings** must be enabled

- Service Principals can use Fabric APIs
- Users can access data stored in OneLake with apps external to Fabric

From the Azure SQL Database Perspective

To check if a Azure SQL Database is being mirrored by Fabric, run the following query:

```
SELECT * FROM sys.dm_change_feed_log_scan_sessions
```

To stop/disable Mirroring, execute the following stored procedure:

```
exec sp_change_feed_disable_db;
```

Pricing

- **Zero cost** for Mirroring into Fabric.
- **Storage** is free up to a **certain limit** based on your Fabric SKU capacity.
- If the Azure SQL database is in a **different region** from your Fabric capacity, data egress will be charged. What is data egress.

Exercise

1. Validate requirements:
   a) Firewall Rules: Public Network Access
   b) Allow Azure Services.
   c) Enable SAMI
   d) 100 DTUs or Vcore Pricing.
   e) No Azure Virtual Network or Private Networking
2. Create Credentials: SQL Authentication, Microsoft Entra ID
3. Fabric Portal: Create an Azure SQL Database Mirroring Item
4. Enter credentials for the connection
5. Import Sales Order Header and Sales Order Detail

Sources

- Microsoft Fabric Mirrored Databases From Azure SQL Database - Microsoft Fabric | Microsoft Learn
- Frequently asked questions for Mirroring Azure SQL Database in Microsoft Fabric - Microsoft Fabric | Microsoft Learn
- Tutorial: Configure Microsoft Fabric Mirrored Databases From Azure SQL Database - Microsoft Fabric | Microsoft Learn
- Mirroring - Microsoft Fabric | Microsoft Learn

Pipelines with Copy Job

Copy Job – Intro

- Another way to ETL/ELT Data!
- Promise:
  - Intuitive Experience
  - Efficiency
  - Flexibility
  - Robust performance
- **Two modes:**
  - Full Copy Mode
  - Incremental Copy Mode

- Few data sources (currently):
  - Azure SQL DB
  - On-premises SQL Server
  - Fabric Warehouse
  - Fabric Lakehouse
  - Amazon S3
  - Azure Data Lake Storage Gen2
  - Azure Blob Storage
  - Amazon RDS for SQL Server

Why do we need different modes?

V's	
Volume	Small datasets: full refresh Large datasets: incremental approach
Velocity	Daily Use Cases: nightly batch approach Minute by Minute Use Cases: CDC, event-driven pipelines, etc.
Variety	Hybrid architectures
Veracity	Full refreshes tend to be more accurate
Cost	Low budget: full refresh daily High budget: real-time data

Which modes exist in Fabric?

- **Change Data Capture (CDC):** row-level changes in databases and propagates them in real time.
- **Full Refresh:** ensures data consistency and simplicity but becomes inefficient with large datasets.
- **Incremental Refresh** through timestamps and high-water mark tracking.
  - New records
  - Modified records
- **Event-driven Pipelines:** process data asynchronously using messaging systems like Apache Kafka or Azure Event Hubs.
- **Hybrid**
  - Lambda architectures
  - Micro-batching

Shortcuts

Fabric objects which point to other storage location. The location could be internal or external

What is a Shortcut?

Object which points to other storage locations. The storage location pointed to is known as “target path”, the location where the shortcut is created, it is known as “shortcut path”.

Any service, Fabric or non-Fabric can access Shortcuts, as long as they have access to the OneLake.

Shortly, you could say they are just links to the data. Deleting a shortcut will not affect the target location.

Types of Shortcut

Internal One Lake: Reference data within other Fabric items. You can create shortcuts for items sharing a workspaces or for items in different workspaces.

External: You can create shortcuts to non-Fabric objects. Some examples are:

- ADLS
- Google Cloud Storage
- Amazon S3
- Amazon S3 Compatible Shortcut
- Dataverse
- Databricks
- Iceberg (Snowflake)

Where Can I Create Shortcuts in Fabric?

Lakehouse: You can create shortcuts in both, Tables and Files folder. For Table shortcuts, data must already be in a Delta format, therefore it is only used for internal shortcutting. External shortcuts are displayed in the Files folder.

KQL Database: Shortcuts will appear in a folder named as “Shortcuts” within the KQL Database. For querying data, the “external_table” function is used in KQL.

Limitations

- There is a limit of 100,000 shortcuts per Fabric item.
- Maximum number of shortcuts in a single OneLake path: 10
- Maximum number of direct shortcuts to shortcut links: 5
- The characters “%” or “+” are invalid for shortcut names.
- Shortcuts do not support non-Latin characters.

Shortcut vs Mirroring

	Shortcut	Mirroring
Storage	Not in fabric, data is stored in the original storage location.	Data is copied into OneLake. In consequence, it is in a Delta Format.
Performance	Depends on the original storage location speed.	It is optimized for analytical queries. It is Fabric-dependant
Updates	Brings the latest data	From time to time, it synchronizes with the external source by making incremental updates.
Storage Requirements	As no data is copied, there is no storage requirements	Does require storage space in OneLake

Exercise: ADLS Shortcut

ADLS shortcuts must point to the DFS endpoint for the storage account.

Requirements

Hierarchical Namespaces enabled in the ADLS: The files are now organized in directories and subdirectories. Like folders in a computer

The shortcut must be created in an existing Lakehouse.

Exercise

1. Validate requirements:
   a) Lakehouse created in Fabric
   b) Hierarchical Namespaces enabled
2. Copy the ".dfs" path of the storage account
3. Get Account Key
4. Enter credentials for the connection
5. Import Sales Order Header and Sales Order Detail

Sources

- Azure Data Lake Storage hierarchical namespace - Azure Storage | Microsoft Learn
- Create a storage account for Azure Data Lake Storage - Azure Storage | Microsoft Learn
- Create a OneLake shortcut - Microsoft Fabric | Microsoft Learn
- Create an Azure Data Lake Storage Gen2 shortcut - Microsoft Fabric | Microsoft Learn