**What is Microsoft Fabric?**

**Microsoft Fabric** is an **end-to-end, SaaS-based data platform** that brings together all the essential capabilities needed for data engineering, data integration, data warehousing, data science, real-time analytics, and business intelligence — all in **one unified environment**. It is deeply integrated with **Power BI**, **Azure Data Factory**, and **Synapse Analytics**.

Launched in 2023, Microsoft Fabric aims to eliminate silos by providing a **unified compute and storage model** using **OneLake**, Microsoft's single logical data lake.

**🧩 Core Components of Microsoft Fabric**

Fabric is built around **seven core workloads**:

**1. Data Factory**

* Equivalent to Azure Data Factory (ADF).
* **Drag-and-drop interface** for building ETL/ELT pipelines.
* Supports **Dataflows Gen2**, mapping data flows with Power Query-like transformations.
* Built-in connectors to various sources: on-prem, SaaS apps, Azure services, and more.
* Scheduling, monitoring, and parameterization supported.

**2. Synapse Data Engineering**

* Spark-based data engineering environment.
* Use **Apache Spark notebooks** (PySpark, Scala, SQL, R).
* Real-time and batch processing using **Lakehouse** or **Delta Lake** architecture.
* Suitable for big data and machine learning pipelines.

**3. Synapse Data Warehousing**

* Serverless, scalable SQL engine over **OneLake Delta tables**.
* **T-SQL-based** queries on Lakehouse data.
* Combines the ease of a data warehouse with the flexibility of a data lake (i.e., **Lakehouse**).
* Supports **Warehouse-specific compute** (dedicated SQL pool-like behavior).

**4. Data Science**

* Build, train, and deploy ML models directly in Fabric.
* Integrated with **Azure Machine Learning** (in future roadmap).
* Supports **SparkML**, **MLlib**, and integration with open-source Python/R libraries.

**5. Real-Time Analytics**

* Built for ingesting and analyzing streaming data (e.g., IoT, logs).
* Combines **Eventstream**, **KQL-based queries**, and **Real-Time Hub**.
* Uses **Kusto Query Language (KQL)** similar to Azure Data Explorer.

**6. Power BI**

* Native integration — Fabric is **deeply connected** to Power BI.
* Reports and dashboards can be built directly on top of Lakehouse, Warehouse, or other Fabric data assets.
* Supports **DirectLake mode** — instant visualization without import or DirectQuery.

**7. Data Activator (New)**

* Low-code/no-code automation for data alerts and triggers.
* Works like **Power Automate** but focused on real-time data triggers (e.g., alert when sales drops below threshold).
* Useful for **action-oriented insights** without coding.

**🏗️ Foundational Concepts in Microsoft Fabric**

**🔹 OneLake — Unified Storage Layer**

* **OneLake (One Logical Lake)** is a single, SaaS-managed **data lake** that underpins the entire Fabric platform.
* Similar to how OneDrive works for Office 365 — OneLake abstracts physical storage.
* **Delta Lake format** used for interoperability and open standards.
* Organizes data into **workspaces** and **folders**.

**🔹 Lakehouse**

* Combines structured and unstructured data in an open format (Delta Lake).
* Data scientists, engineers, and analysts can all work on the **same dataset**.
* Can be queried using **Spark**, **SQL**, or **Power BI**.

**🔹 Warehouse**

* Traditional relational modeling with **SQL-first access**.
* Backed by **delta tables in OneLake**, allowing direct interop with Lakehouse/Spark.
* Familiar to data engineers from **Synapse dedicated SQL pools**.

**🔹 Shortcuts**

* OneLake allows you to **create shortcuts** to data in other storage accounts (e.g., ADLS Gen2, AWS S3).
* Works like a symbolic link — **no data movement**.
* Facilitates **data virtualization** and reuse.

**🔐 Security and Governance in Fabric**

* Integrated with **Microsoft Purview** for data cataloging, classification, and lineage.
* **Role-based access control (RBAC)** and **Microsoft Entra ID (Azure AD)** integration.
* **Row-level and column-level security** supported in Power BI and Warehouse.
* **Audit logs, sensitivity labels**, and **data masking** are enforced platform-wide.

**⚙️ Development & Operations (DevOps)**

* **Git integration**: Native support for source control (Azure DevOps & GitHub).
* **CI/CD Pipelines**: Deployment of notebooks, dataflows, reports, and pipelines across environments.
* **Monitoring**: Built-in job tracking, pipeline runs, and cost insights.
* **Notebooks & Notepad**: Spark Notebooks for data engineering and data science; Data Notepad for ad hoc exploration.

**🔄 Comparison with Azure Synapse**

| **Feature** | **Microsoft Fabric** | **Azure Synapse** |
| --- | --- | --- |
| Storage | OneLake (Delta Lake) | ADLS Gen2 |
| Compute | SaaS-based, Unified | Separate SQL/DW/Spark |
| Integration | Native Power BI, One UI | Loose Power BI Integration |
| Setup | No infra management | Requires provisioning pools |
| Security | Built-in Purview, Entra ID | Needs integration |
| CI/CD | Git-native | Azure DevOps (manual integration) |
| Real-time | Built-in Eventstream | KQL & ADX needed |
| Pricing | Per capacity (SKU model) | Per compute/hour |

**Example Real-World Use Case in Fabric**

**Scenario: Sales Analytics Platform**

1. **Data Ingestion**:
   * Use **Data Factory pipelines** to bring CSV data from an FTP server daily.
   * Ingest customer and product data from SQL Server and Dynamics 365.
2. **Data Storage**:
   * Land raw data in **Lakehouse** in OneLake.
   * Cleaned and transformed tables saved as **Delta Tables**.
3. **Data Transformation**:
   * Use **Synapse Data Engineering** notebooks (PySpark) to clean and join datasets.
   * Implement Slowly Changing Dimensions (SCD) for the product dimension.
4. **Modeling**:
   * Load cleaned data into a **Warehouse** for structured reporting.
   * Create **semantic model** with relationships and measures.
5. **Visualization**:
   * Build reports using **Power BI** on top of the Warehouse and Lakehouse.
   * Dashboards track sales, revenue, churn, and product performance.
6. **Automation**:
   * **Data Activator** triggers email alerts when daily sales drop below threshold.
   * Real-time updates from POS systems handled through **Eventstreams**.

In **Microsoft Fabric**, models are a key part of organizing and analyzing data. The two main types of models you'll encounter are:

**1. Semantic Model (formerly known as a Dataset in Power BI)**

* **Purpose**: Delivers a **business-friendly view of data** with meaningful names, relationships, and calculated measures.
* **Interface**: Built with Power BI or directly in Fabric through the Lakehouse or Warehouse experiences.
* **Storage**: Stored in the **VertiPaq engine**, highly optimized for in-memory analytics.
* **Usage**:
  + Used in **Power BI reports** and dashboards.
  + Includes **measures (DAX)**, calculated columns, hierarchies, and relationships.
  + Enables **row-level security (RLS)**.
* **End-user friendly**: Designed for analysts and business users to easily drag and drop fields.

**2. Other Models in Fabric (Lakehouse, Warehouse, Direct Lake, etc.)**

**a. Lakehouse Model**

* **Purpose**: Combines structured and unstructured data using a **data lake + Delta tables**.
* **Interface**: SQL analytics with notebooks or Spark.
* **Storage**: Data stored in **OneLake using Delta Parquet format**.
* **Usage**:
  + Great for **big data processing**, ML, and ad hoc querying.
  + Can feed into a semantic model for reporting.

**b. Warehouse Model**

* **Purpose**: Traditional **relational data warehouse** model.
* **Interface**: T-SQL based, familiar for SQL professionals.
* **Storage**: Also uses **OneLake and Delta tables** but optimized for **relational workloads**.
* **Usage**:
  + Used for structured data storage, transformations, and BI preparation.

**Summary Table**

| **Feature** | **Semantic Model** | **Lakehouse** | **Warehouse** |
| --- | --- | --- | --- |
| Built with | Power BI / Fabric UI | Notebooks / Spark SQL | T-SQL |
| Storage format | VertiPaq (in-memory) | Delta Parquet (OneLake) | Delta Parquet (OneLake) |
| Query language | DAX | SQL / PySpark | SQL |
| Use case | Reporting & analytics | Data science, ML, big data | Structured data processing |
| End user-friendly? | Yes | No | Moderate |
| Supports Measures/DAX? | Yes | No | No (not natively) |

Data pipelines are a common concept in data engineering and offer a wide variety of activities to orchestrate. Some common activities include:

* Copy data
* Incorporate Dataflow
* Add Notebook
* Get metadata
* Execute a script or stored procedure

Pipelines provide a visual way to complete activities in a specific order. You can use a dataflow for data ingestion, transformation, and landing into a Fabric data store. Then incorporate the dataflow into a pipeline to orchestrate extra activities, like execute scripts or stored procedures after the dataflow has completed.

Pipelines can also be scheduled or activated by a trigger to run your dataflow. By using a pipeline to run your dataflow, you can have the data refreshed when you need it instead of having to manually run the dataflow. When you're dealing with enterprise or frequently changing data, automation allows you to focus on other responsibilities.

*Dataflows* are a type of cloud-based ETL (*Extract, Transform, Load*) tool for building and executing scalable data transformation processes.

Dataflows Gen2 allow you to extract data from various sources, transform it using a wide range of transformation operations, and load it into a destination. Using Power Query Online also allows for a visual interface to perform these tasks.

**1. Power Query ribbon**

Dataflows Gen2 support a wide variety of data source connectors. Common sources include cloud and on-premises relational databases, Excel or flat files, SharePoint, SalesForce, Spark, and Fabric lakehouses. Then there are numerous data transformations possible, such as:

* Filter and Sort rows
* Pivot and Unpivot
* Merge and Append queries
* Split and Conditional split
* Replace values and Remove duplicates
* Add, Rename, Reorder, or Delete columns
* Rank and Percentage calculator
* Choose Top N and Bottom N

You can also create and manage data source connections, manage parameters, and configure the default data destination in this ribbon.

**2. Queries pane**

The Queries pane shows you the different data sources - now called *queries*. These queries are called tables when loaded to your data store. You can duplicate or reference a query if you need multiple copies of the same data, such as creating a star schema and splitting data into separate, smaller tables. You can also disable the load of a query, in case you only need the one-time import.

**3. Diagram view**

The Diagram View allows you to visually see how the data sources are connected and the different applied transformations. For example, your dataflow connects to a data source, duplicates the query, removes columns from the source query, then unpivots the duplicate query. Each query is represented as a shape with all of the applied transformations and connected by a line for the duplicate query. You can turn this view on or off.

**4. Data Preview pane**

The Data Preview pane only shows a subset of data to allow you to see which transformations you should make and how they affect the data. You can also interact with the preview pane by dragging and dropping columns to change order or right-clicking on columns to filter or make changes. The data preview shows all of your transformations for the selected query.

**5. Query Settings pane**

The Query Settings pane includes the **Applied Steps**. Each transformation is represented as a step, some of which are automatically applied when you connect the data source. Depending on the complexity of the transformations, you might have several applied steps for each query. Most steps have a gear icon that allows you to modify the step, otherwise you must delete and repeat the transformation.

Each step also has a contextual menu when you right-click so you can rename, reorder, or delete the steps. You can also view the data source query when connecting to a data source that supports query folding.

While this visual interface is helpful, you can also view the M code through **Advanced editor**.



In the Query settings pane, you can see a **Data Destination** option to land your data in one of the following locations in your Fabric environment:

* Lakehouse
* Warehouse
* SQL database

You can also load your dataflow to Azure SQL database, Azure Data Explorer, or Azure Synapse Analytics.

Dataflows Gen2 provide a low-to-no-code solution to ingest, transform, and load data into your Fabric data stores. Power BI developers are familiar and can quickly begin to perform transformations upstream to improve performance for their reports.