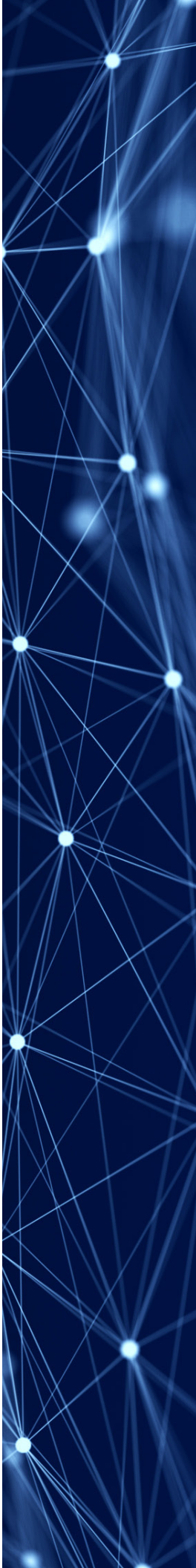
INVISO

Data engineering in Microsoft Fabric



Contents

Unlock the power of your data with data engineering in Microsoft Fabric 3

Lakehouse: A unified data platform for data engineering 5

Apache Spark job definition in the context of data engineering 7

Notebooks are a powerful tool for data engineering in Microsoft Fabric 9

Data pipelines in Fabric: A powerful tool for data engineering 12

Unlock the power of your data with data engineering in Microsoft Fabric

Data engineering is the process of building and maintaining the infrastructure and systems that enable organizations to collect, store, process, and analyze large volumes of data. In today's data-driven world, data engineering is essential for businesses of all sizes.

Microsoft Fabric is a unified platform for data engineering that provides a comprehensive set of tools and services for building and managing data pipelines, lakes, and warehouses. With Microsoft Fabric, data engineers can easily collect, store, process, and analyze data from a variety of sources, including on-premises systems, cloud services, and streaming data sources.

Microsoft Fabric also provides a variety of features that make it easy to collaborate with other data engineers and data scientists. For example, Microsoft Fabric provides a centralized repository for data assets, as well as tools for version control and collaboration.

If you're looking for a powerful and flexible data engineering platform, Microsoft Fabric is the perfect choice for you. With Microsoft Fabric, you can easily unlock the power of your data and gain insights that can help you make better business decisions.

Some of the key benefits of using Data Engineering in Microsoft Fabric are:

Increased agility: Microsoft Fabric makes it easy to build and deploy data pipelines quickly and easily. This means that you can respond to changing business needs more quickly and effectively.

Scalability: Microsoft Fabric can scale to meet the needs of even the most demanding organizations, so you can be confident that your data engineering platform will be able to handle growing data volumes.

Increased agility: Microsoft Fabric is a highly reliable platform that is backed by Microsoft's global support network — you can be confident that your data is always safe and accessible.

Increased agility: Security is a top priority for Microsoft Fabric. Fabric data is encrypted at rest and in transit. Users must authenticate with a strong password or MFA method and can be authorized to access only the resources they need. All user activity is logged for auditing purposes. Fabric is also compliant with a wide range of industry and regulatory standards.

If you're looking for a data engineering platform that can help you achieve your business goals, Microsoft Fabric is the perfect choice.

One of the advantages of using Fabric data engineering is that it provides many ways of implementing data engineering pipelines. To get started, first create a Fabric Lakehouse (if you haven't already). You could start with either a data pipeline or use an Azure Data Factory-like experience to create a pipeline. Another option is to use R or Python with a new notebook or import a notebook if one is already created. Spark jobs allow the creation of a job that can then automate a notebook to run on a schedule.

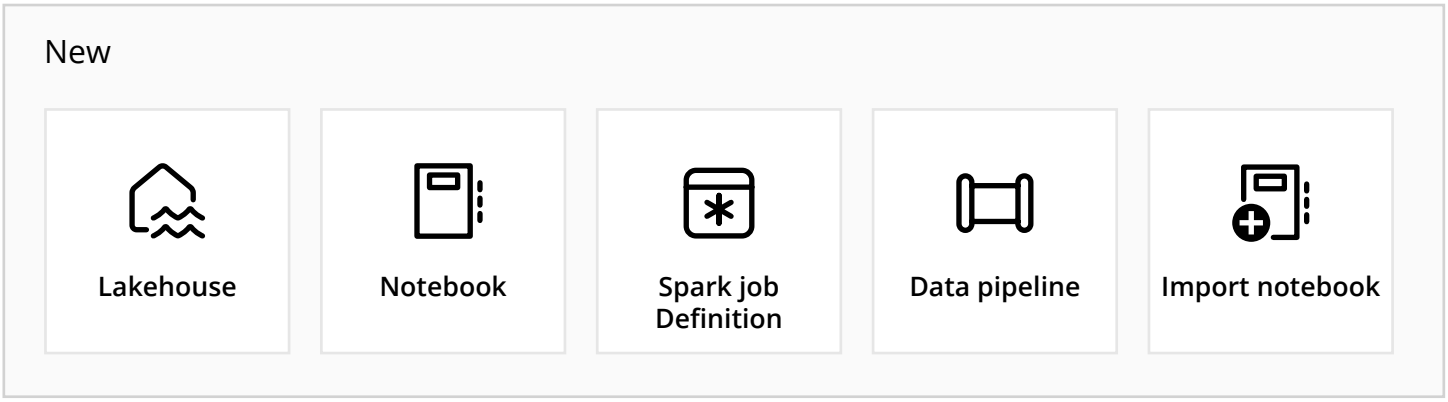
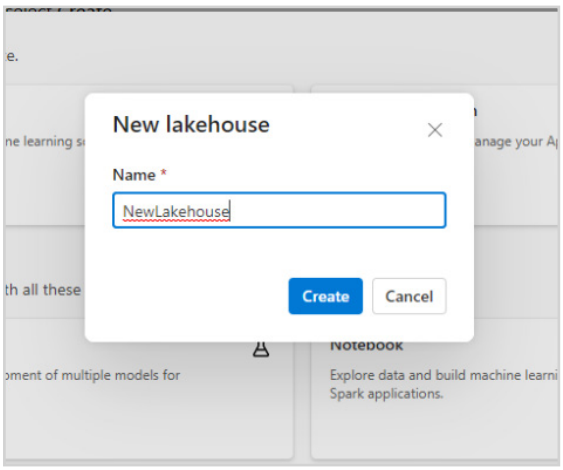


Image courtesy of Microsoft

Lakehouse: A unified data platform for data engineering

Lakehouse in Fabric provides a single location for storing, managing, and analyzing all of your data, regardless of whether it's structured or unstructured. This simplifies your data landscape and makes it easier to get insights from all of your data. Lakehouse plays a critical role in data engineering by providing a unified platform for data ingestion, processing, storage, and analytics.



Some of the key benefits of using Lakehouse for data engineering are:

Unified platform: Lakehouse provides a single location for storing and managing all types of data, including structured, unstructured, and semi-structured data. This makes it easy for data engineers to build and maintain adaptable data pipelines that can process all types of data.

Scalability and performance: Lakehouse in Fabric is optimized for speed and performance. It uses a variety of technologies, such as Apache Spark and Delta Lake, to accelerate data processing and querying.

Flexibility: Lakehouse is a flexible platform that can be used to support a wide variety of data engineering workloads, including batch processing, streaming data processing, machine learning, and data science.

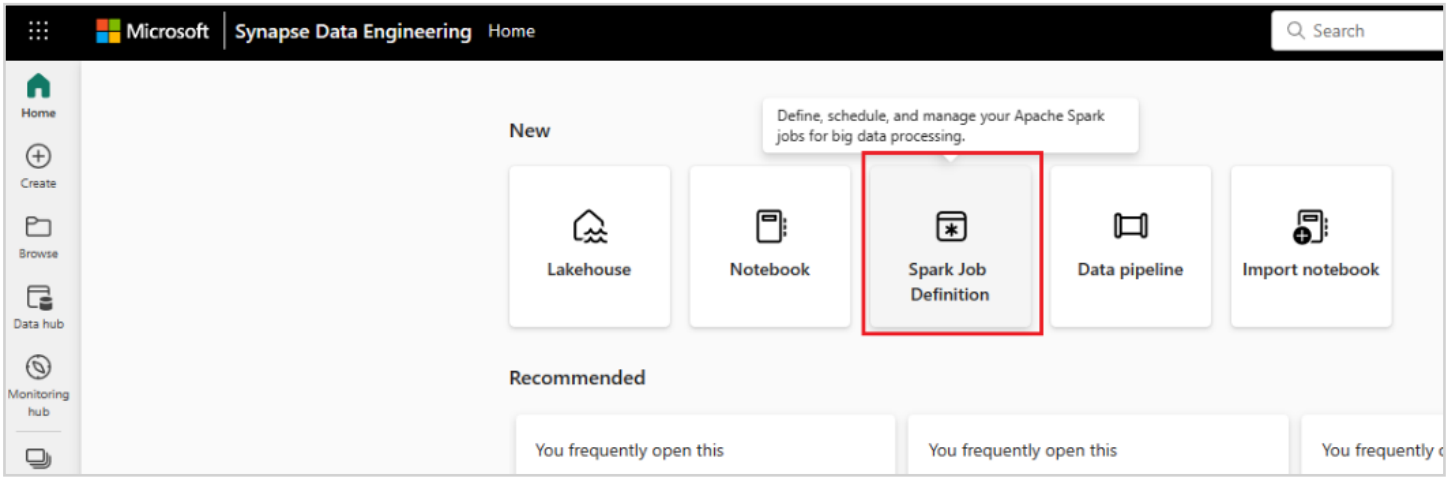
Open source: Lakehouse supports the Delta table format which is Open Source. Delta is free to use and there is a large community of developers who contribute to the platform. This makes it easy to find help and support when needed.

Best practices for using Lakehouse for data engineering include:

- **Use a data governance framework:** Lakehouse is a powerful platform, but it is important to use a data governance framework to manage the data securely and effectively. A data governance framework will help you define who has access to the data, how the data can be used, and how the data will be protected.
- **Use a data quality framework**: It is also important to use a data quality framework to ensure that the data stored in Lakehouse is accurate and complete. A data quality framework will help you to identify and fix data errors and monitor the quality of your data over time.

Lakehouse is a powerful and flexible platform for data engineering that can help organizations achieve their business goals. By following the best practices outlined above, you can ensure that you are using Lakehouse effectively to build and maintain well-performing data pipelines.

Apache Spark job definition in the context of data engineering



- Apache Spark is a distributed computing framework built into Fabric that is widely used for data processing and analytics. It provides a variety of features that make it ideal for data engineering, including:

Scalability: Spark can scale to handle large volumes of data, making it ideal for data engineering tasks such as batch processing and streaming data processing.

Performance: Spark is a high-performance computing framework that can process data quickly and efficiently.

Flexibility: Spark can be used for a variety of data engineering tasks, including batch processing, streaming data processing, machine learning, and data science.

An Apache Spark job definition is a specification for a Spark job. It includes information such as the input and output data sources, the Spark application to run, and the configuration parameters for the job.

Spark job definitions are used in data engineering to build and maintain data pipelines. A data pipeline is a series of steps that are used to process and analyze data. Spark job definitions can be used to define each step in a data pipeline. For example, you can use them to define a job that:

- Ingests data from a variety of sources.
- Cleans and transforms data.
- Loads data into a data warehouse or data lake.
- Trains a machine learning model.
- Scores data with a machine learning model.

Spark job definitions are a powerful tool that can help organizations build and maintain data pipelines that can streamline the process of data engineering by providing a structured framework for executing complex data processing tasks.

Best practices for using Spark job definitions:

- **Use Spark job definitions with other data engineering tools**: Spark job definitions can be used with Fabric. This makes it easy to build and maintain data pipelines that can meet the specific needs of your organization.
- **Test and deploy Spark job definitions**: It is important to test and deploy Spark job definitions in a controlled environment before deploying them to production. This will help ensure your Spark job definitions are working as expected.
- **Monitor Spark job definitions**: Once your Spark job definitions are deployed to production, it's important to monitor their performance and health. This will help you to identify and fix any problems that may occur.

By following these best practices, you can ensure that you are using Spark job definitions effectively to build and maintain data pipelines that bolster the scalability and adaptability of data operations.

Notebooks are a powerful tool for data engineering in Microsoft Fabric

It is very helpful to provide a collaborative environment for data engineers to write and execute code, document their work, and visualize data. This helps enable teams to share and reuse code, thereby promoting consistency and reducing redundancy in data operations.

Notebooks in Fabric are based on Apache Spark, which is a distributed computing framework that is widely used for data processing and analytics. This makes notebooks ideal for a variety of data engineering tasks, such as batch processing, streaming data processing, machine learning, and data science.

Some of the key benefits of using notebooks for Data Engineering in Fabric:

Collaborative environment: Notebooks are a collaborative environment, which means multiple data engineers can work on the same notebook at the same time. This makes it easy to collaborate on data engineering projects.

Interactivity: Notebooks are interactive, which means that data engineers can execute code and see results immediately. This makes it easy to experiment with different code and to debug problems.

Documentation: Notebooks can be used to document the steps involved in a data engineering project so other data engineers can understand and maintain the project.

Visualization: Notebooks can be used to visualize data, making it easier to identify patterns and trends in the data.

Best practices for using notebooks for data engineering include:

- **Test and deploy notebooks**: It is important to test and deploy notebooks in a controlled environment before deploying them to production. This will help to ensure your notebooks are working as expected.
- **Monitor notebooks**: Once your notebooks are deployed to production, it's important to monitor their performance and health. This will help you identify and fix any problems that may occur.
- **Fabric language support**:
  - Python (PySpark)
    - Easy to learn and use.
    - Popular in the data science community.
    - Many libraries and tools are available.
    - Allows you to write Spark jobs in Python.
  - Scala
    - Highly scalable and performant language, making it ideal for working with large datasets.
    - Supports functional programming paradigms, which can be very useful for data science and machine learning tasks.
  - Spark SQL
    - High-performance SQL engine for Apache Spark.
    - Allows you to query and analyze data using SQL.
    - Standard query language that is widely used.
  - R (SparkR)
    - Popular language for statistical computing and data visualization.
    - Allows you to write Spark jobs in R.
  - HTML
    - Used to display formatted text and other HTML elements, such as images, tables, and lists. This can be useful for creating interactive dashboards and reports.

Specific examples of how a data engineer can use a notebook for data engineering in Fabric:

- Define a Spark job that ingests data from a variety of sources.
- Define a Spark job that cleans and transforms data.
- Define a Spark job that loads data into a data warehouse or data lake.
- Train a machine learning model.
- Score data with a machine learning model.
- Visualize data and identify patterns and trends.

Notebooks are a powerful tool Fabric that can help improve efficiency and productivity of data engineering in Fabric.

Data pipelines in Fabric: A powerful tool for data engineering

A data pipeline is a series of steps that are used to collect, process, and deliver data to a specific destination. Data pipelines are essential for data engineering, as they provide a way to automate the data processing process and make data available to users in a timely and efficient manner.

Microsoft Fabric provides a variety of features that make it easy to build and manage data pipelines. For example, Fabric includes a data pipeline designer that allows users to create data pipelines visually. Fabric also includes a variety of pre-built data connectors and data processors that can be used to build data pipelines quickly and easily.

Here are some of the benefits of using data pipelines in Fabric:

Increased efficiency: Data pipelines can automate the data processing process, which can free up data engineers to focus on other tasks.

Improved data quality: Data pipelines can be used to ensure that the data is processed correctly, and that data quality is maintained.

Reduced costs: Data pipelines can help to reduce the costs associated with data processing by automating the process and optimizing the use of resources.

Increased agility: Data pipelines can help organizations respond to changing business needs more quickly by making data available to users in a timely manner.

Best practices for using data piplelines in Fabric include:

- **Use data pipelines with other data engineering tools**: Data pipelines can be used with a variety of other data engineering technologies, such as Delta and Spark. This makes it easy to build and maintain data pipelines that can meet the specific needs of your organization.
- **Test and deploy data pipelines**: It is important to test and deploy data pipelines in a controlled environment before deploying them to production. This will help ensure that your data pipelines are working as expected.
- **Monitor data pipelines**: Once your data pipelines are deployed to production, it's important to monitor their performance and health so you can identify and fix any problems that may occur.
- **Design your data pipelines carefully**: Before you start building your data pipelines, it's important to design them carefully. This includes identifying the data sources, the data processors, and the data destinations. You'll also want to consider the performance and scalability requirements of your data pipelines.
- **Use modular design**: It's a good practice to design your data pipelines in a modular way so it's easier to maintain and update them in the future.
- **Use version control**: Using version control for your data pipelines makes it easier to track changes to your data pipelines so you can revert to previous versions if necessary.

By following these best practices, you can ensure that you are using data pipelines effectively in Fabric to build and maintain data pipelines that can meet the needs of your organization.

Need help?
You can rely on Inviso

If you're looking for help implementing data engineering in Fabric for your organization, we've got you covered.

Our team of experienced data engineers know how to get the most out of Microsoft Fabric. We can help you:

- Design, build, and deploy data pipelines that are reliable, efficient, and scalable.
- Choose the right data engineering tools and technologies for your organization.
- Train and support your team to help maximize your data engineering investment.

If you are interested in learning more about how Inviso can help you implement data engineering in Fabric, get in touch! We'd be happy to schedule a consultation to discuss your specific needs and requirements.

We look forward to hearing from you!

✉ info@invisocorp.com

🌐 www.invisocorp.com

NOTICE

The information contained in this document represents the current view of Inviso on the issues discussed as of the date of publication and is subject to change at any time without notice. This document and its contents are provided AS IS without warranty of any kind and should not be interpreted as an offer or commitment on the part of Inviso and Inviso cannot guarantee the accuracy of any information presented.

INVISO MAKES NO WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

All trademarks are the property of their respective companies.

All rights reserved.