

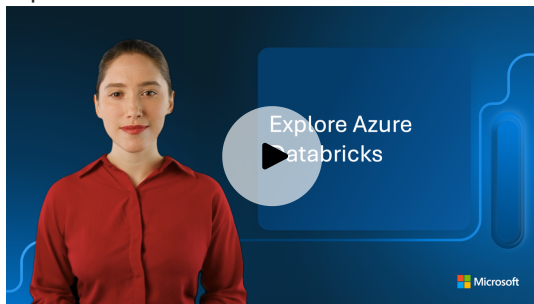
Explore Azure Databricks

1. Introduction

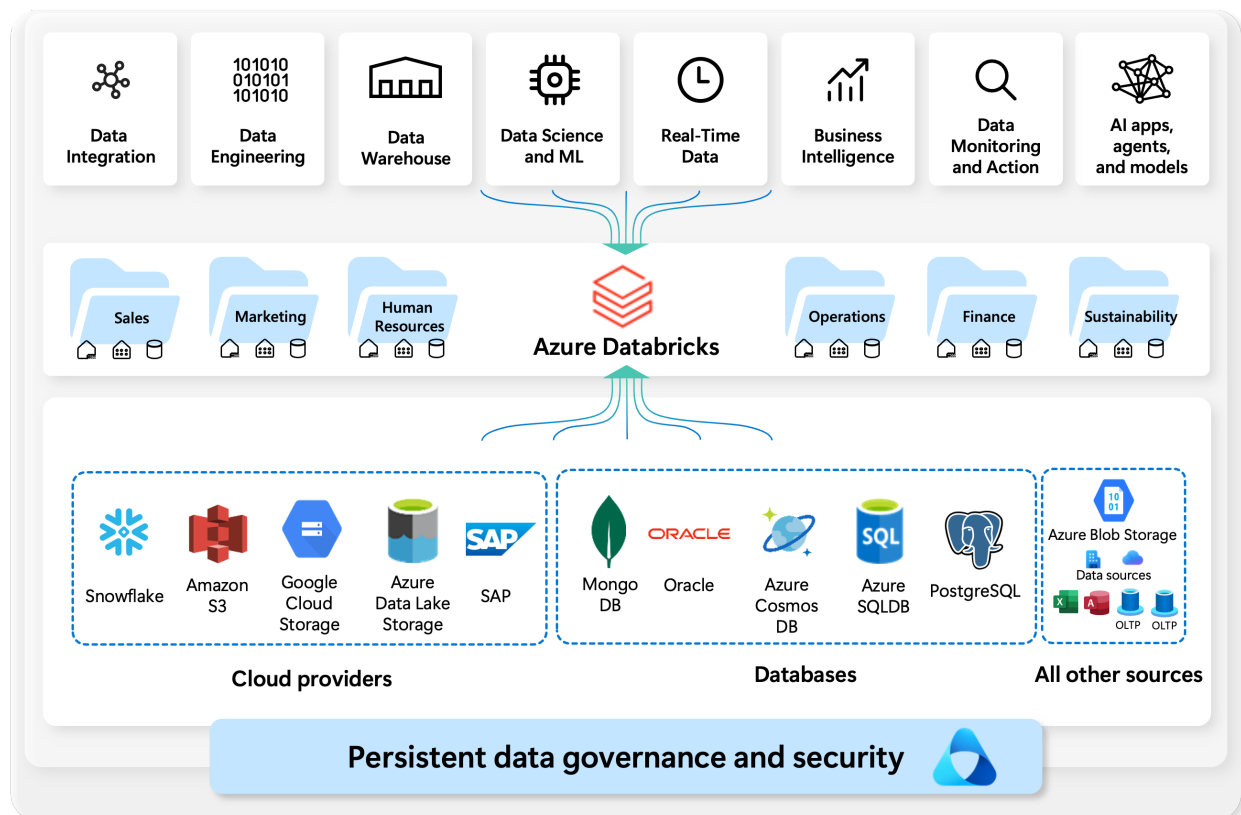
<https://learn.microsoft.com/en-us/training/modules/explore-azure-databricks/01-introduction>

Introduction

Completed



Azure Databricks is a cloud-based data platform that brings together the best of **data engineering, data science, and machine learning** in a single, unified workspace. Built on top of **Apache Spark**, it allows organizations to easily process, analyze, and visualize massive amounts of data in real time.



By connecting to a wide range of **data sources**—from cloud providers like Azure SQL Database, Amazon S3 and Google Cloud Storage, to enterprise systems such as SAP and Oracle—Azure Databricks makes it easy to integrate and transform data from anywhere.

Once data is ingested, teams across **sales, marketing, operations, finance, HR, and sustainability** can use Databricks for advanced analytics, machine learning, business intelligence, and AI-driven insights.

At its core, Azure Databricks helps organizations:

- **Integrate** data from multiple sources
- **Engineer** and transform raw data into usable formats
- **Store and manage** data efficiently with governance and security
- **Apply** real-time analytics, machine learning, and AI models
- **Drive** better business decisions and outcomes

Data Lakehouse

A **data lakehouse** is a data management approach that blends the strengths of both data lakes and data warehouses. It offers scalable storage and processing, allowing organizations to handle diverse workloads—such as machine learning and business intelligence—without relying on separate, disconnected systems. By centralizing data, a lakehouse supports a single source of truth, reduces duplicate costs, and ensures that information stays up to date.

Many lakehouses follow a layered design pattern where data is gradually improved, enriched, and refined as it moves through different stages of processing. This layered approach—commonly called the **medallion architecture**—organizes data into stages that build on one another, making it easier to manage and use effectively.

The Databricks lakehouse uses two key technologies:

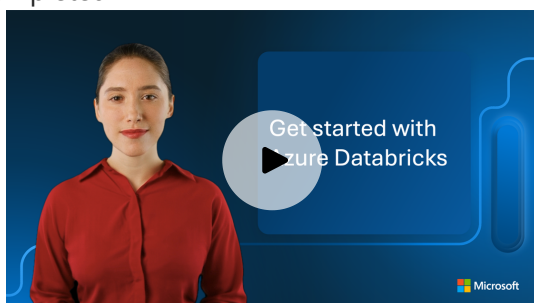
- **Delta Lake**: an optimized storage layer that supports ACID transactions and schema enforcement.
- **Unity Catalog**: a unified, fine-grained governance solution for data and AI.

2. Get started with Azure Databricks

<https://learn.microsoft.com/en-us/training/modules/explore-azure-databricks/02-azure-databricks>

Get started with Azure Databricks

Completed



To use Azure Databricks, you must create an Azure Databricks workspace in your Azure subscription. A workspace is an Azure Databricks deployment in a cloud service account. It provides a unified environment for working with Azure Databricks assets for a specified set of users.

You can create an Azure Databricks workspace by:

- Using the Azure portal user interface.
- Using an Azure Resource Manager (ARM), Bicep or Terraform template.
- Using the New-AzDatabricksWorkspace Azure PowerShell cmdlet.
- Using the az databricks workspace create Azure command line interface (CLI) command.

When you create a workspace, you must specify:

- A **workspace name**.
- Select an available **region**. For available regions, see [Azure services available by region](#).
- A **pricing tier**:
 - **Standard** - Core Apache Spark capabilities with Microsoft Entra ID integration.
 - **Premium** - Role-based access controls and other enterprise-level features.
 - **Trial** - A 14-day free trial of a premium-level workspace
- **Managed Resource Group name** (optional): an automatically created resource group where Azure provisions and manages the infrastructure resources needed for your Databricks workspace.

The screenshot shows the 'Create an Azure Databricks workspace' page in the Azure portal. The page has a blue header with the Microsoft Azure logo and a search bar. Below the header, there's a breadcrumb 'Home >' and a title 'Create an Azure Databricks workspace'. The page is divided into tabs: 'Basics', 'Networking', 'Encryption', 'Security & compliance', 'Tags', and 'Review + create'. The 'Basics' tab is active. Under 'Project Details', there's a section for 'Subscription' and 'Resource group'. The 'Subscription' dropdown is set to 'Subscription' and the 'Resource group' dropdown is set to '(New) rg-databricks'. Below this, there's a section for 'Instance Details'. The 'Workspace name' field is set to 'databricks-ws' and has a green checkmark. The 'Region' dropdown is set to 'West Europe'. The 'Pricing Tier' dropdown is set to 'Premium (+ Role-based access controls)'. A notification banner below the pricing tier says: 'We selected the recommended pricing tier for your workspace. You can change the tier based on your needs.' The 'Managed Resource Group name' field is empty with the placeholder text 'Enter name for managed resource group'. At the bottom, there are three buttons: 'Review + create', '< Previous', and 'Next : Networking >'.

If you decide to create an Azure Databricks deployment using the Azure CLI, this would be the [az databricks workspace](#) command to remember:

```
az databricks workspace create
  --resource-group myresourcegroup \
  --name mydatabricksws \
```

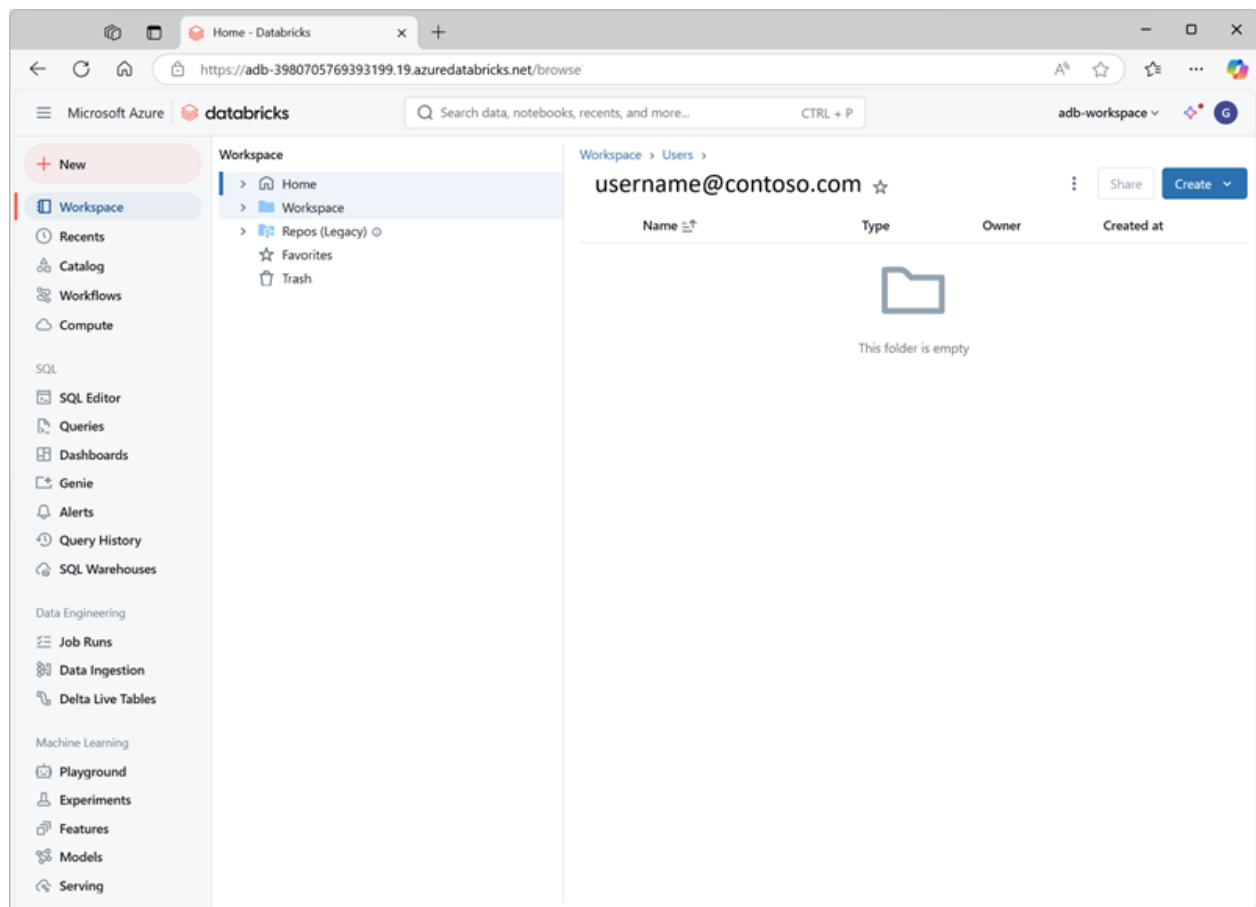
```
--location westus2 \  
--sku standard
```

The equivalent [New-AzDatabricksWorkspace](#) PowerShell cmdlet:

```
New-AzDatabricksWorkspace -Name mydatabricksws -ResourceGroupName myresourcegroup
```

Navigating the Azure Databricks Workspace UI

After you provision an Azure Databricks workspace, you can use the workspace UI to work with data and compute resources. The workspace UI is a web-based user interface where you can create and manage workspace resources, such as Spark clusters, and use notebooks and queries to work with data in files and tables.



The homepage provides shortcuts to common tasks and workspace objects to help you get started. You can import data, create a notebook, create a query and configure an AutoML experiment.

The sidebar shows common Databricks categories (Workspace, Recents, Catalog, Jobs & Pipelines, Compute, Marketplace). It then breaks out by product area:

- **SQL:** SQL Editor, Queries, Dashboards, Genie, Alerts, Query History, SQL Warehouses
- **Data Engineering:** Job Runs, Data Ingestion

- **Machine Learning:** Playground, Experiments, Features, Models, Serving

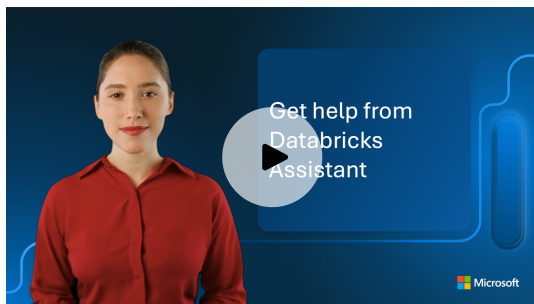
Select + **New** to:

- **Create workspace objects** such as notebooks, queries, repos, dashboards, alerts, jobs, pipelines, experiments, models, and serving endpoints.
- **Create compute resources** such as clusters, SQL warehouses, and ML endpoints.

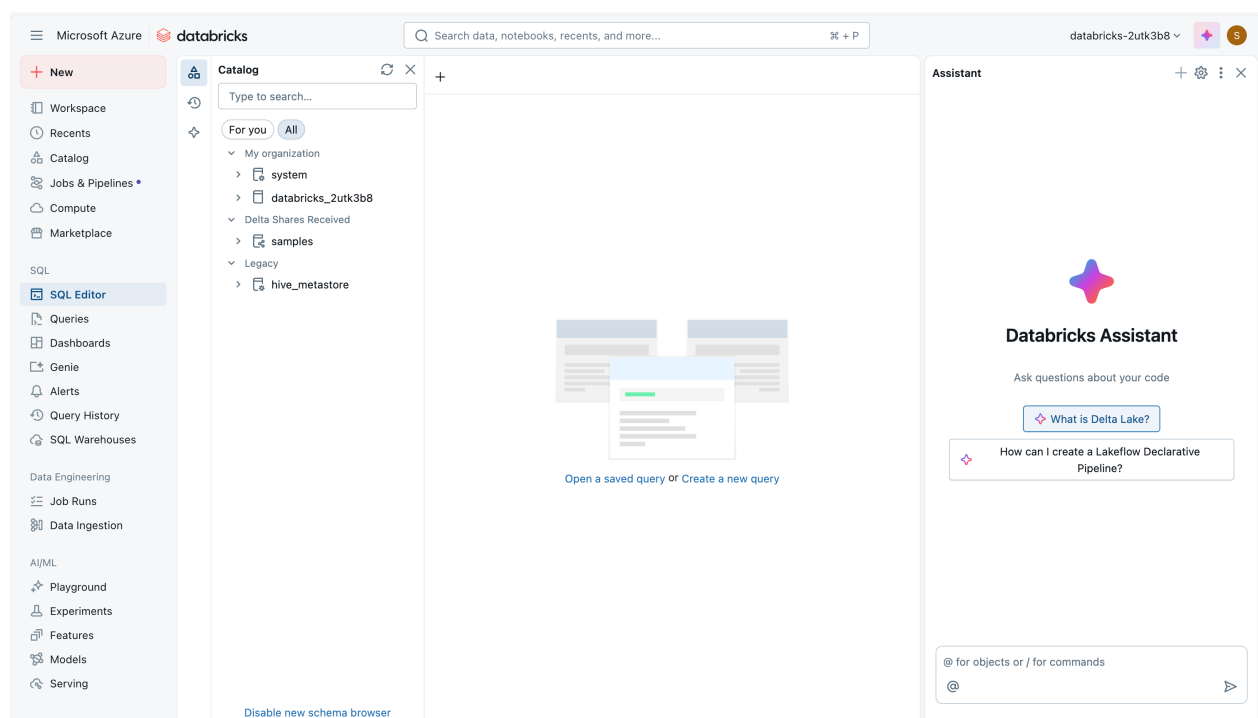
Use the top bar to **search** for workspace objects such as notebooks, queries, dashboards, alerts, files, folders, libraries, tables registered in Unity Catalog, jobs, and repos in a single place. You can also access recently viewed objects in the search bar.

The workspace is available in **multiple languages**. To change the workspace language, select your username in the top navigation bar, select **Settings** and go to the **Preferences** tab.

Get help from Databricks Assistant



Databricks Assistant is an AI-powered pair programmer and support tool that helps you work more efficiently in Databricks by generating, explaining, and fixing code or queries directly in notebooks, dashboards, and files.



It can assist with a wide range of tasks, including identifying and correcting errors, creating data visualizations, diagnosing job issues, and filtering or analyzing data using natural language prompts. The Assistant can surface relevant guidance from the Azure Databricks documentation.

By using Unity Catalog metadata, it personalizes its responses based on your organization's data assets—tables, columns, and descriptions—making it easier to explore and work with your data.

3. Identify Azure Databricks workloads

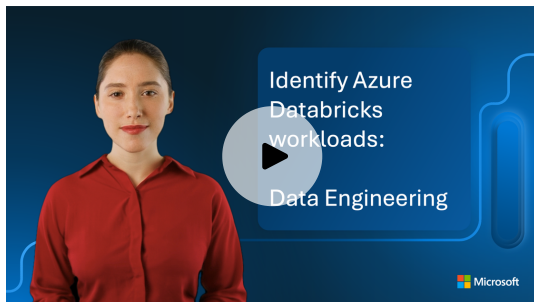
<https://learn.microsoft.com/en-us/training/modules/explore-azure-databricks/03-workloads>

Identify Azure Databricks workloads

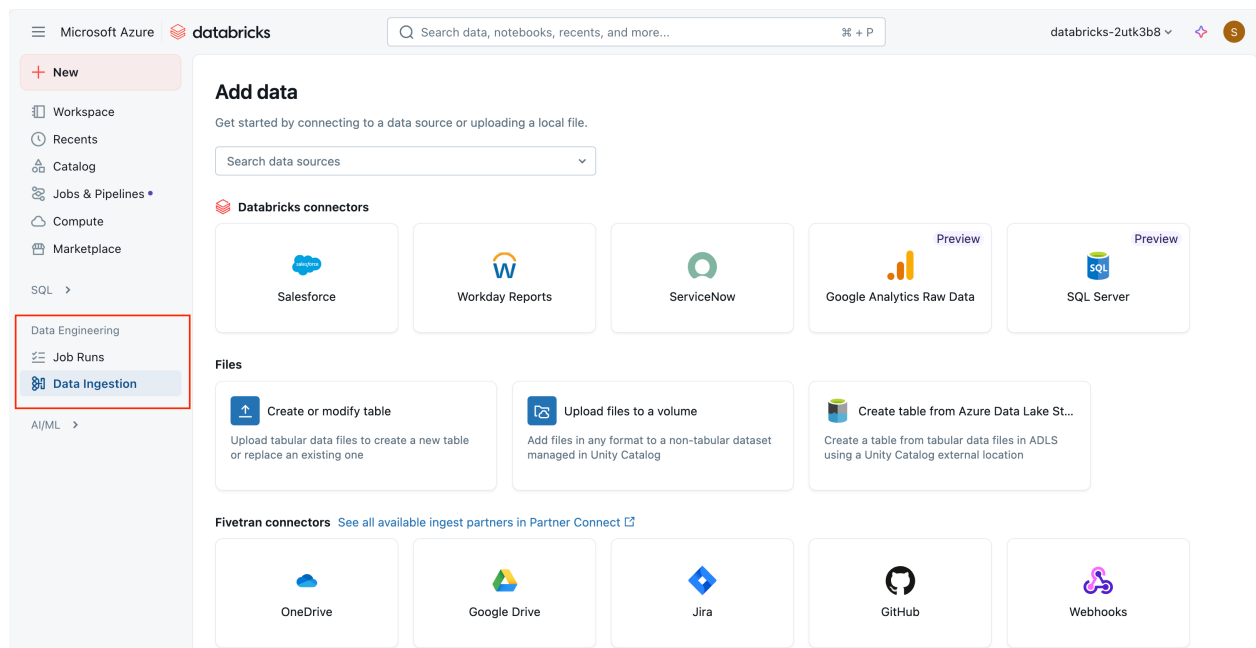
Completed

Azure Databricks offers capabilities for various workloads including Machine Learning and Large Language Models (LLM), Data Science, Data Engineering, BI and Data Warehousing, and Streaming Processing.

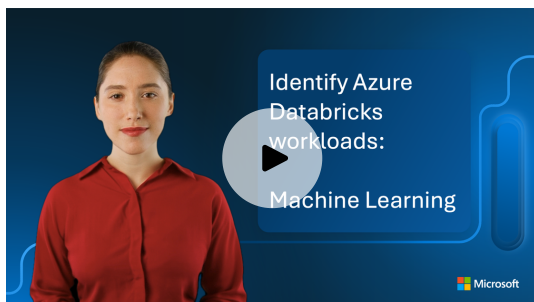
Data Engineering



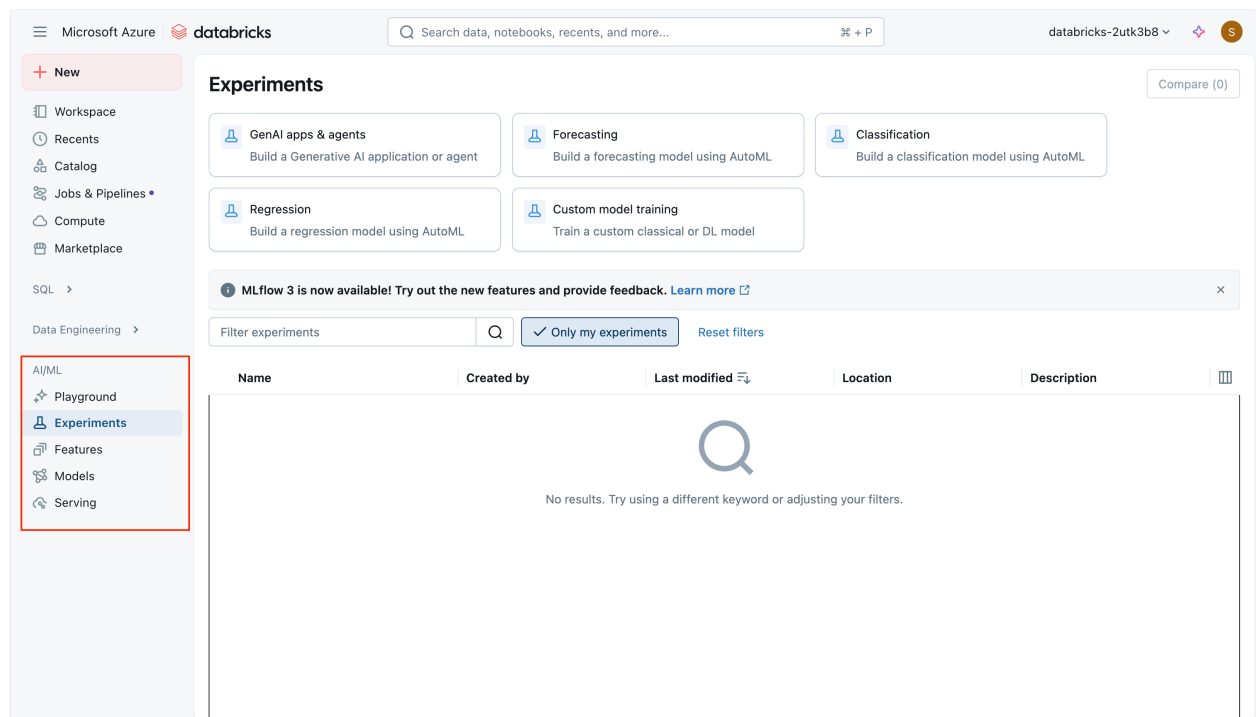
Azure Databricks provides capabilities for data scientists and engineers who need to collaborate on complex data processing tasks. It provides an integrated environment with Apache Spark for big data processing in a data lakehouse, and supports multiple languages including Python, R, Scala, and SQL. The platform facilitates data exploration, visualization, and the development of data pipelines.



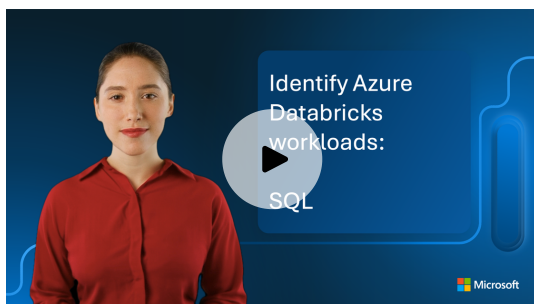
Machine Learning



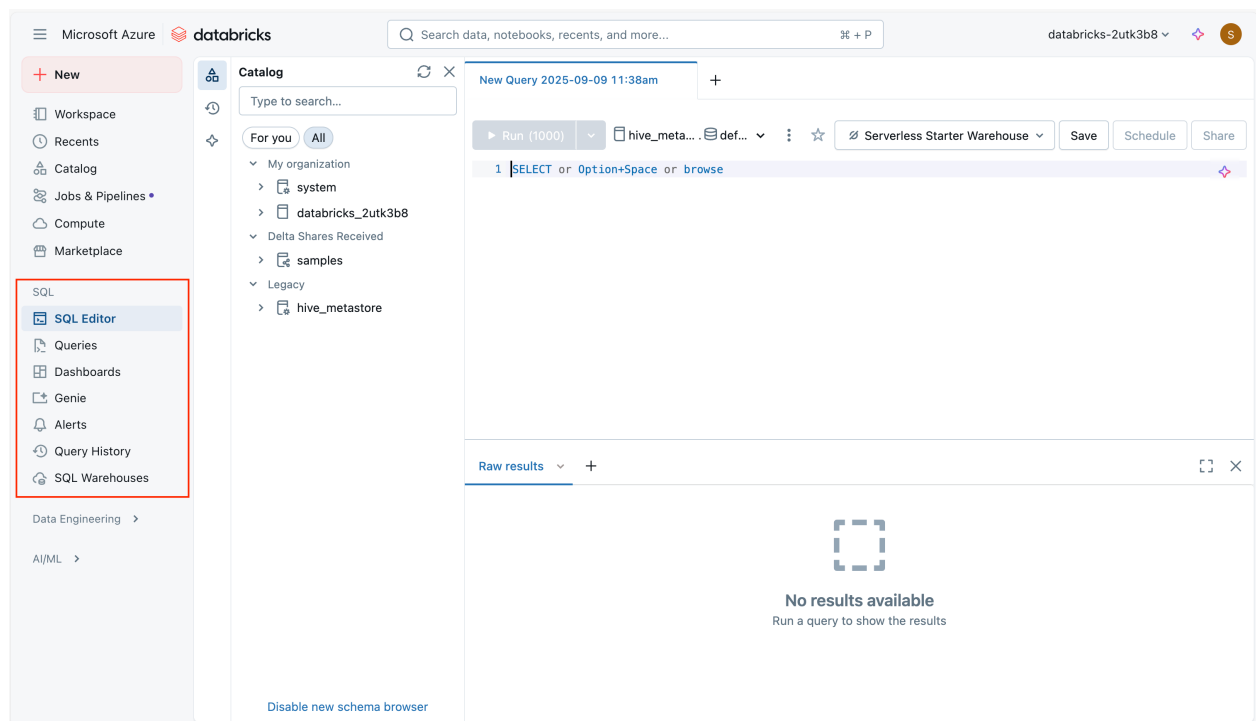
Azure Databricks supports building, training, and deploying machine learning models at scale. It includes MLflow, an open-source platform to manage the ML lifecycle, including experimentation, reproducibility, and deployment. It also supports various ML frameworks such as TensorFlow, PyTorch, and Scikit-learn, making it versatile for different ML tasks.



SQL



Data analysts who primarily interact with data through SQL can use SQL warehouses in Azure Databricks. The Azure Databricks Workspace UI provides a familiar SQL editor, dashboards, and automatic visualization tools to analyze and visualize data directly within Azure Databricks. This workload is ideal for running quick ad-hoc queries and creating reports from large datasets.



Note

SQL warehouses are included in the Premium (or higher) tier. Standard workspace doesn't provide SQL warehouses.

4. Understand key concepts

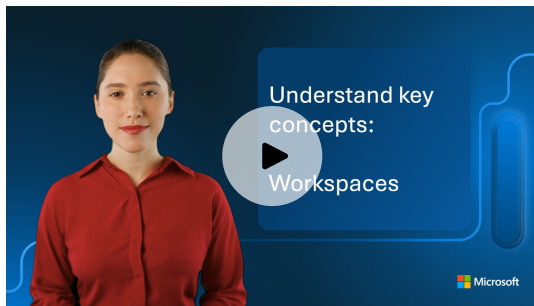
<https://learn.microsoft.com/en-us/training/modules/explore-azure-databricks/04-key-concepts>

Understand key concepts

Completed

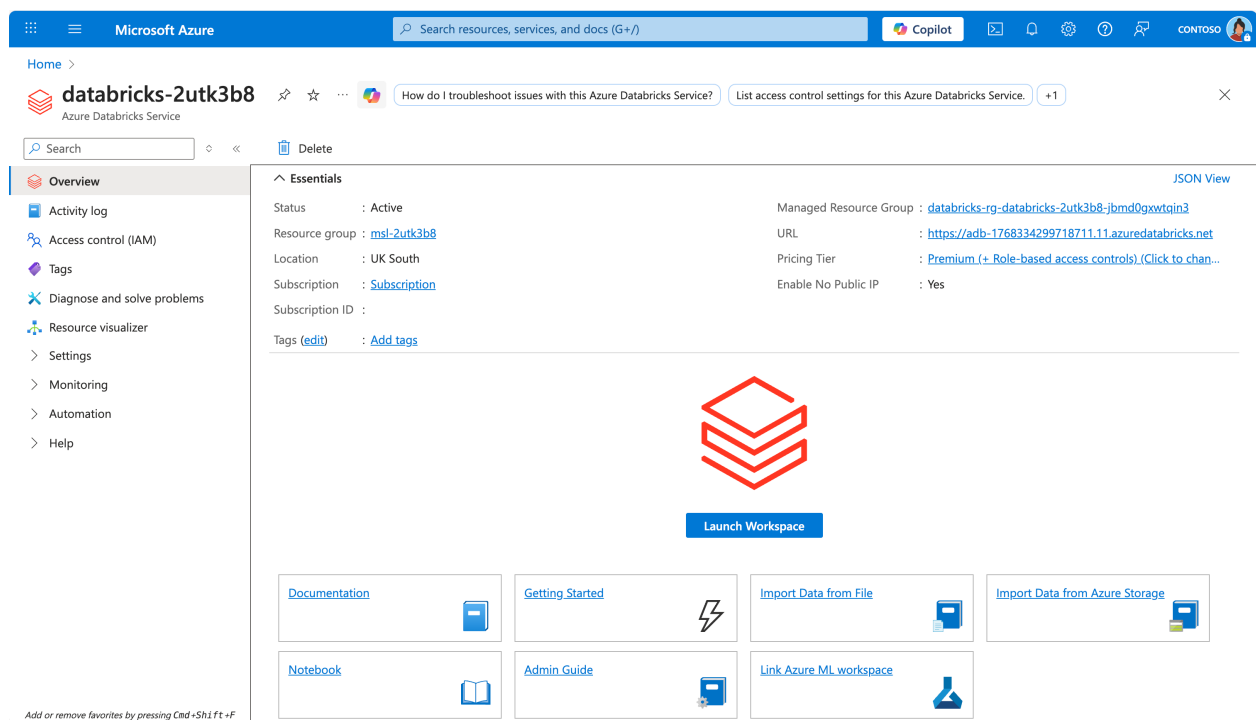
Azure Databricks is a single service platform with multiple technologies that enable working with data at scale. When using Azure Databricks, there are some key concepts to understand.

Workspaces



A **workspace** in Azure Databricks is a secure, collaborative environment where you can access and organize all Databricks assets, such as notebooks, clusters, jobs, libraries, dashboards, and experiments.

You can open an Azure Databricks Workspace from the Azure portal, by selecting **Launch Workspace**.



It provides a web-based **user interface (UI)** as well as REST APIs for managing resources and workflows. Workspaces can be structured into folders to organize projects, data pipelines, or team assets, and permissions can be applied at different levels to control access. They support **collaboration** by allowing multiple users—such as data engineers, analysts, and data scientists—to work together on shared notebooks, track experiments, and manage dependencies.

In addition, workspaces are tied to **Unity Catalog** (when enabled) for centralized data governance, ensuring secure access to data across the organization. Each workspace is also linked to an **underlying Azure resource group** (including a managed resource group) that holds the compute, networking, and storage resources Databricks uses behind the scenes.

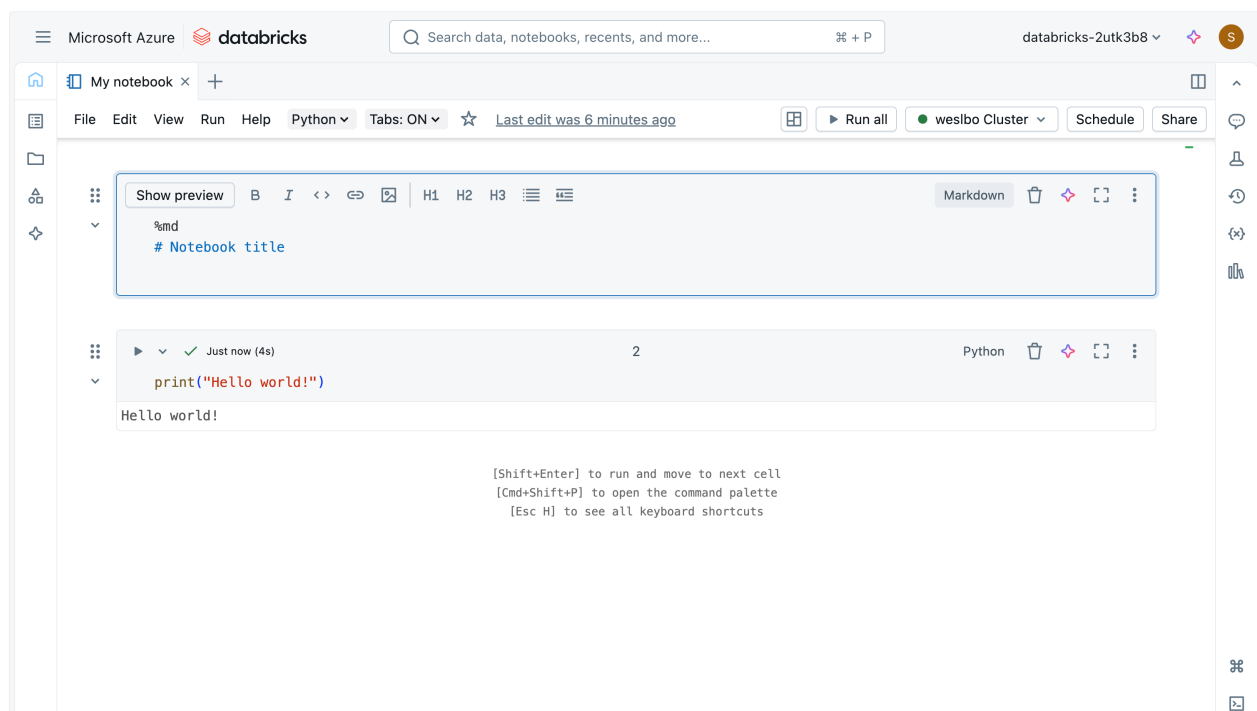
Notebooks



Databricks notebooks are interactive, web-based documents that combine **runnable code**, **visualizations**, and **narrative text** in a single environment. They support multiple languages—such as Python, R, Scala, and SQL—and allow users to switch between languages within the same notebook using *magic commands*. This flexibility makes notebooks well-suited for **exploratory data analysis**, **data visualization**, **machine learning experiments**, and **building complex data pipelines**.

Notebooks are also designed for **collaboration**: multiple users can edit and run cells simultaneously, add comments, and share insights in real time. They integrate tightly with Databricks clusters, enabling users to process large datasets efficiently, and can connect to external data sources through **Unity Catalog** for governed data access. In addition, notebooks can be version-controlled, scheduled as jobs, or exported for sharing outside the platform, making them central to both **ad-hoc exploration** and **production-grade workflows**.

Notebooks contain a collection of two types of cells: **code cells** and **Markdown cells**. Code cells contain runnable code. Markdown cells contain Markdown code that renders as text and graphics. You can **run** a single cell, a group of cells, or the whole notebook.

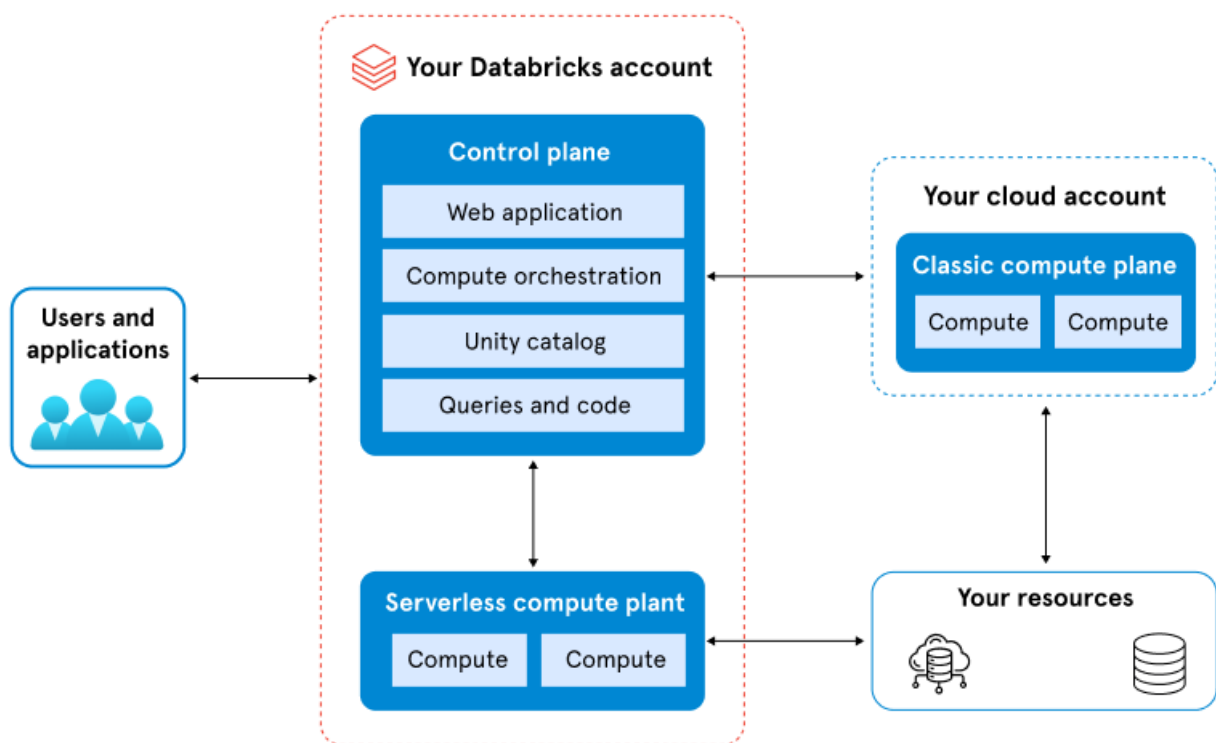


Clusters



Azure Databricks leverages a two-layer architecture:

- **Control Plane:** this internal layer, managed by Microsoft, handles backend services specific to your Azure Databricks account.
- **Compute plane:** this is the external layer that processes the data and lives in your Azure Subscription.



Clusters are the core computational engines in Azure Databricks. They provide the processing power required to run data engineering, data science, and analytics tasks. Each cluster consists of a **driver node**, which coordinates execution, and one or more **worker nodes**, which handle the distributed computations. Clusters can be created manually with fixed resources or set to **auto-scale**, allowing Databricks to add or remove worker nodes depending on workload demand. This flexibility ensures efficient resource use and cost control.

Azure Databricks Compute offers a broad set of compute options available for different workload types:

- **Serverless compute:** Fully managed, on-demand compute that automatically scales up or down to meet workload needs. Ideal for teams that want fast startup times, minimal

management overhead, and elastic scaling.

- **Classic compute:** User-provisioned and configured clusters that offer full control over compute settings, such as VM sizes, libraries, and runtime versions. Best for specialized workloads that require customization or consistent performance.
- **SQL warehouses:** Compute resources optimized for SQL-based analytics and BI queries. SQL warehouses can be provisioned as either **serverless** (elastic, managed) or **classic** (user-configured) depending on governance and performance requirements.

This allows you to tailor compute to specific needs—from exploratory analysis in notebooks, to large-scale ETL pipelines, to high-performance dashboards and reporting.

Databricks Runtime

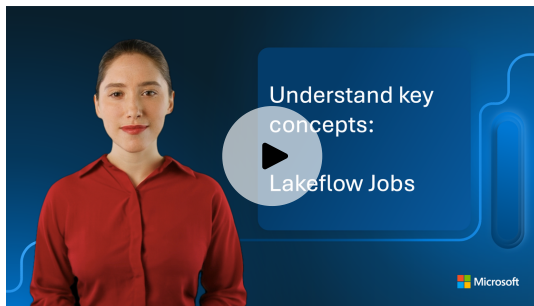
The **Databricks Runtime** is a set of customized builds of **Apache Spark** that include performance improvements and additional libraries. These runtimes make it easier to handle tasks such as **machine learning**, **graph processing**, and **genomics**, while still supporting general data processing and analytics.

Databricks provides multiple runtime versions, including **long-term support (LTS)** releases. Each release specifies the underlying Apache Spark version, its release date, and when support will end. Over time, older runtime versions follow a lifecycle:

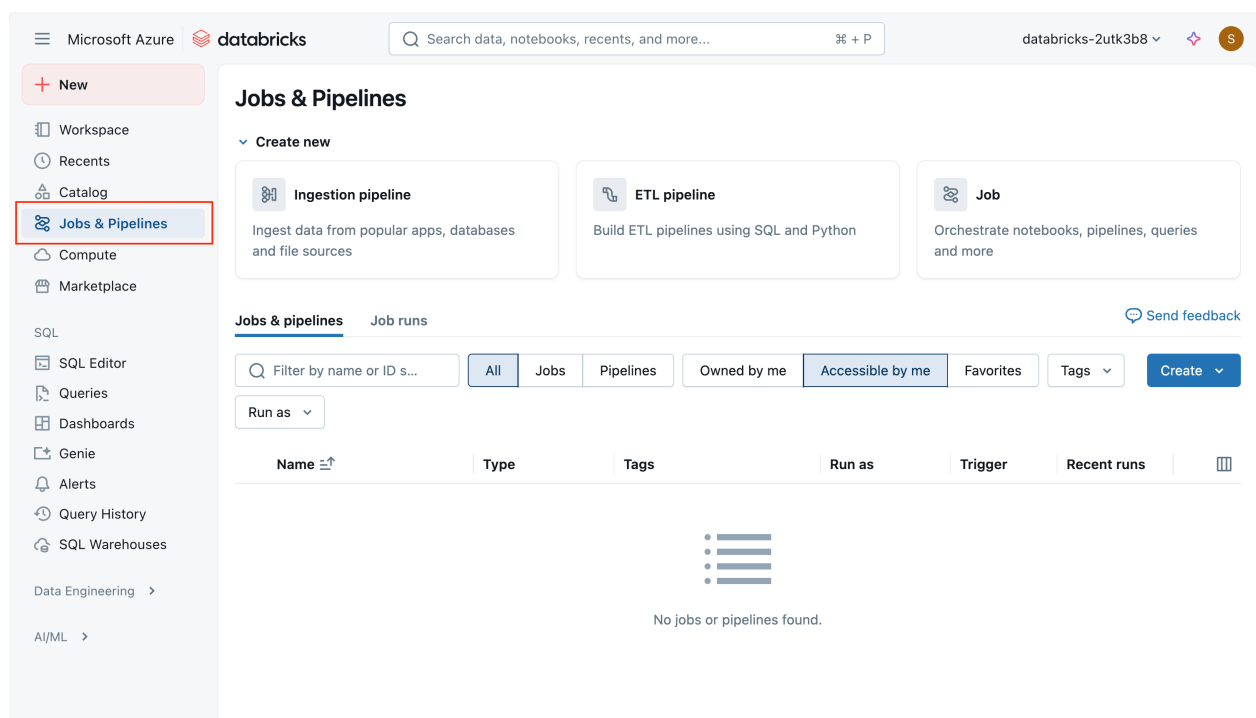
- **Legacy** – available but no longer recommended.
- **Deprecated** – marked for removal in a future release.
- **End of Support (EoS)** – no further patches or fixes are provided.
- **End of Life (EoL)** – retired and no longer available.

If a maintenance update is released for a runtime version you're using, you can apply it by **restarting your cluster**.

Lakeflow Jobs



Lakeflow Jobs provide workflow automation and orchestration in Azure Databricks, making it possible to reliably schedule, coordinate, and run data processing tasks. Instead of running code manually, you can use jobs to automate repetitive or production-grade workloads such as ETL pipelines, machine learning training, or dashboard refreshes.



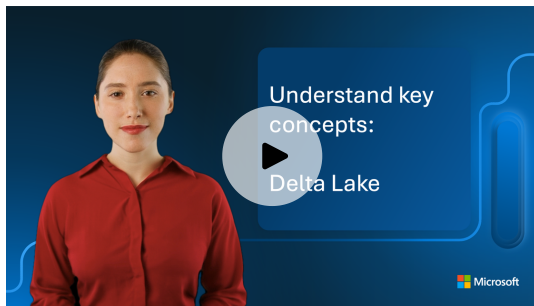
A job in Databricks is essentially a container for one or more **tasks**. Tasks define the work to be done—for example, running a notebook, executing a Spark job, calling external code, ...

Jobs can be triggered in different ways:

- On a schedule (for example, every night at midnight).
- In response to an event.
- Manually, when needed.

Because they're repeatable and managed, jobs are critical for **production workloads**. They ensure that data pipelines run consistently, ML models are trained and deployed in a controlled manner, and downstream systems receive updated, accurate data.

Delta Lake



Delta Lake is an open-source storage framework that improves the reliability and scalability of data lakes by adding transactional features on top of cloud object storage, such as **Azure Data Lake Storage**. Traditional data lakes can suffer from issues like inconsistent data, partial writes, or difficulties managing concurrent access. Delta Lake addresses these problems by supporting:

- **ACID transactions** (atomicity, consistency, isolation, durability) for reliable reads and writes.
- **Scalable metadata handling** so tables can grow to billions of files without performance loss.
- **Data versioning and rollback**, enabling time travel queries and recovery of previous states.
- **Unified batch and streaming processing**, so the same table can handle real-time ingestion and historical batch loads.

On top of this foundation, **Delta tables** provide a familiar table abstraction that makes it easier to work with structured data using **SQL queries** or the **DataFrame API**. Delta tables are the **default table format in Azure Databricks**, ensuring that new data is stored with transactional guarantees by default.

Databricks SQL

Databricks SQL brings **data warehousing capabilities** to the Databricks Lakehouse, allowing analysts and business users to query and visualize data stored in open formats directly in the data lake. It supports **ANSI SQL**, so anyone familiar with SQL can run queries, build reports, and create dashboards without needing to learn new languages or tools.

Databricks SQL is available only in the **Premium tier** of Azure Databricks. It includes:

- A **SQL editor** for writing and running queries.
- **Dashboards and visualization tools** for sharing insights.
- Integration with external BI and analytics tools.

SQL Warehouses



All Databricks SQL queries run on **SQL warehouses** (formerly called SQL endpoints), which are scalable compute resources decoupled from storage. Different warehouse types are available depending on performance, cost, and management needs:

- **Serverless SQL Warehouses**
 - **Instant and elastic compute** with fast startup and autoscaling.
 - **Low management overhead** since Databricks handles capacity, patching, and optimization.
 - **Cost efficiency** by scaling automatically and avoiding idle resource costs.
- **Pro SQL Warehouses**
 - More customizable but slower to start (≈4 minutes).
 - Less responsive autoscaling compared to serverless.
 - Useful when consistent, predictable workloads are required.
- **Classic SQL Warehouses**
 - Compute resources run in your **own Azure subscription**, not in Databricks.
 - Less flexible than serverless but may be preferred for specific governance or cost management requirements.

MLflow

MLflow is an open-source platform designed to manage the **end-to-end machine learning (ML) lifecycle**. It helps data scientists and ML engineers track experiments, manage models, and streamline the process of moving models from development to production. MLflow also supports generative AI workflows and includes tools for evaluating and improving AI agents.

5. Data governance using Unity Catalog and Microsoft Purview

Data governance using Unity Catalog and Microsoft Purview

Completed

Data governance is critical for ensuring that data within an organization is managed securely, efficiently, and in compliance with regulations.

In many organizations, data is distributed across databases, data warehouses, data lakes, and even multiple catalogs. It also exists in diverse formats like Parquet, CSV, and Delta Lake. Beyond structured data in tables, there's also unstructured data in files, along with other assets such as machine learning models, notebooks, and dashboards that require management and governance. This fragmentation creates silos across sources, formats, and asset types.

These governance challenges directly affect the value organizations can derive from data and AI:

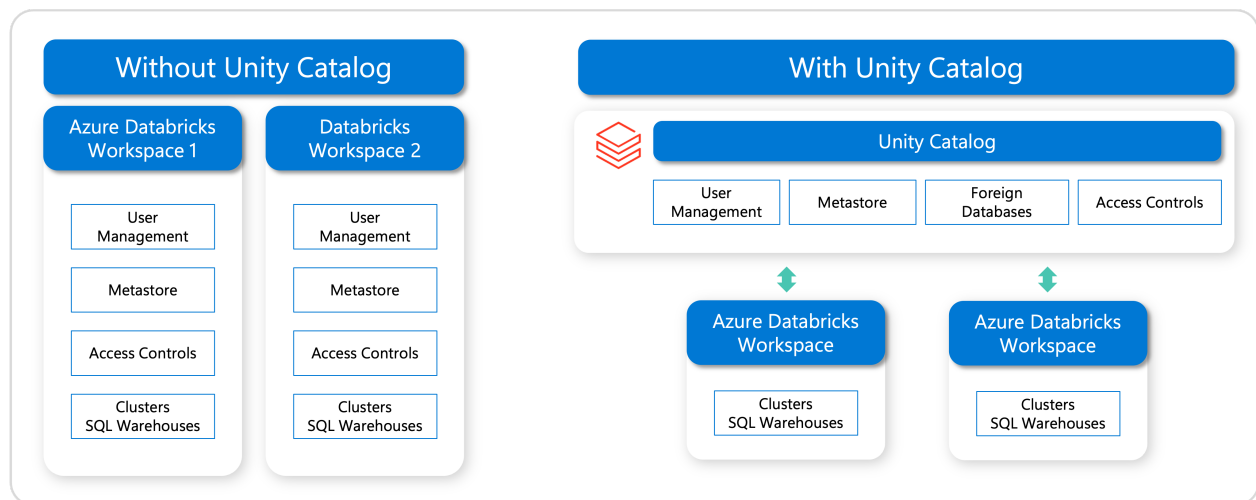
- Fragmented governance increases compliance, security, and data quality risks, while also creating operational inefficiencies as teams struggle to maintain a consistent view of their data and AI environments.
- Limited connectivity can result in vendor lock-in and make it harder to adopt new technologies as requirements change. Poor interoperability also complicates collaboration and scaling, often leading to higher costs from using multiple tools and duplicating data across systems.
- A lack of built-in intelligence restricts broader use of data and AI platforms, particularly for nontechnical users. This slows down innovation, delays decision-making, and prevents organizations from fully realizing the benefits of their data and AI investments.

Azure Databricks, combined with Unity Catalog and Microsoft Purview, provides a robust solution for managing and governing data effectively.

Unity Catalog



Unity Catalog provides a centralized way to manage access, discovery, lineage, audit logs, and quality monitoring across data and AI assets within Azure Databricks. It applies consistently across all workspaces in a region.



The **metastore** is the top-level metadata container; it holds information about data assets and the permissions that govern them. You typically have one metastore per region, and multiple workspaces can share that metastore.

Unity Catalog organizes data assets using a structured **three-level hierarchy**:

```
catalog.schema.table_or_other_object
```

- **Catalogs** group assets typically aligned to teams or environments.
- **Schemas** (also known as databases) are subdivisions within catalogs, organizing assets more granularly—for example by project or use case.
- **Objects** in schemas include tables (managed or external), views, volumes, functions, and models.

Tables can be either managed or external. With **managed tables**, Unity Catalog handles both governance and storage (always Delta Lake format). With **external tables**, Unity Catalog manages access from Databricks, but data lifecycle/storage is managed externally. This supports multiple formats (Delta, CSV, JSON, Parquet, etc.)

Unity Catalog implements **fine-grained access control** via ANSI SQL commands across multiple levels—metastore, catalog, schema, down to rows and columns. For example, the following

command gives the 'finance-team' user group the permission to create new tables in the 'myschema' within the 'mycatalog' database.

```
GRANT CREATE TABLE ON SCHEMA mycatalog.myschema TO `finance-team`;
```

Exploring data assets in Unity Catalog is straightforward. You can use the **Catalog Explorer** and a search interface to find what you need. To help you, assets have tags, comments, and even AI-generated descriptions. Once you find a data asset, you can use features like **lineage**, table insights, and Entity Relationship diagrams to get a better understanding of it.

Unity Catalog provides a complete picture of your data's history. It logs access, audit trails, and lineage—right down to the column level.

In most accounts, Unity Catalog is enabled by default when you create a workspace. You can get started using Unity Catalog with the default settings. There are optional configurations that you might want to enable, however.

Microsoft Purview



Microsoft Purview is a data governance service that lets you manage and oversee data across on-premises systems, multiple clouds, and SaaS platforms. It includes features such as data discovery, classification, lineage tracking, and access governance.

When integrated with Azure Databricks and Unity Catalog, Purview can discover Lakehouse data and ingest its metadata into the Data Map. This allows you to apply consistent governance across your entire data environment, while acting as a central catalog that brings together metadata from different sources.

With this integration, you can:

- **Scan** Azure Databricks in both public and private networks, powered by the fully managed Microsoft Purview integration runtime.
- Scan the entire **Unity Catalog metastore** or choose to scan only selective catalogs.
- Extract a comprehensive set of Unity Catalog metadata, including details of metastore, catalogs, schemas, tables/views, and columns, etc.
- Automatically **classify** the data based on built-in system classification rules or user-defined custom classification rules to identify sensitive data.

- Get detailed visibility into the **data lineage**, showing how data is transformed and moved across different systems and processes, including within Azure Databricks.
- Run the scan **on-demand** or on a daily/weekly/monthly recurring **schedule**.

SampleTable
Azure Databricks Table

☆ ☆ ☆ ☆ ☆ (0)

+ Add Tag

Edit Select for bulk edit Request access Refresh Delete Edit columns

Overview Properties **Schema** Lineage Contacts Related Updated on November 15, 2023 at 2:19 PM by automated scan ([DatabricksScan](#))

Filter by name

Showing 33 of 33 items

Column name	Classifications	Sensitivity label	Glossary terms	Data type	Column description
ID				BIGINT	ID
firstname	⚡ All Full Names +3 More			STRING	First name
State	GEN2CCTEST.STATE +1 More			STRING	
aba routing number				BIGINT	
age	⚡ Person's Age			BIGINT	
ssn	CBF8.PII +1 More			STRING	
credit_card_number	⚡ Credit Card Number +1 More			STRING	Credit card number

In addition, Microsoft Purview can scan the workspace-level **Hive metastore** in Azure Databricks.

6. Exercise - Explore Azure Databricks

<https://learn.microsoft.com/en-us/training/modules/explore-azure-databricks/06-exercise-explore-databricks>

Exercise - Explore Azure Databricks

Completed

Now it's your chance to explore Azure Databricks. In this lab, you explore the Azure Databricks workspace UI, upload a sample dataset to a Unity Catalog volume, and work with notebook features including Python, SQL magic commands, and Markdown. You use the Databricks Assistant throughout to generate and refine code in the context of CityMoves Transit, a fictional public transportation authority.

Note

To complete this lab, you need an [Azure subscription](#) in which you have administrative access.

Launch the exercise and follow the instructions.

Launch Exercise

7. Module assessment

<https://learn.microsoft.com/en-us/training/modules/explore-azure-databricks/07-knowledge-check>

Module assessment

Completed

8. Summary

<https://learn.microsoft.com/en-us/training/modules/explore-azure-databricks/08-summary>

Summary

Completed

Azure Databricks is a cloud-based data analytics platform that provides a unified environment for data engineering, machine learning, and analytics. It's built on top of Apache Spark, which is a powerful, open-source processing engine built around speed, ease of use and sophisticated analytics. Azure Databricks integrates with other services provided by Microsoft Azure, offering a seamless experience for data preparation, building machine learning models, and data analysis.

In this module, you learned how to:

- Provision an Azure Databricks workspace
- Identify core workloads for Azure Databricks
- Use Data Governance tools Unity Catalog and Microsoft Purview
- Describe key concepts of an Azure Databricks solution

Learn more

- [Azure Databricks Documentation](#)

- [Getting started with Azure Databricks](#)
- [Microsoft Purview Documentation](#)
- [Databricks Unity Catalog](#)