

Databricks

Getting Started!



Rajesh Kumar

www.RajeshKumar.xyz

DevOps@RajeshKumar.xyz

Data

Data is any collection of facts, values, or measurements that can be recorded, stored, and processed by computers or humans.

- **Simple examples:** Numbers (42), words (“hello”), dates (2025-08-06), true/false (yes/no), files, images, videos, etc.
- **In technology:** Data is the raw material for software, analytics, machine learning, business intelligence, etc.

Types of Data

A. By Structure

Type	Description	Example
Structured	Organized in fixed fields/columns, like a table	Databases, Excel spreadsheets
Semi-Structured	Has some organization but not a strict schema	JSON, XML, CSV, log files
Unstructured	No pre-defined format or schema	Images, videos, emails, PDFs

Types of Data

B. By Nature/Content

Type	Description	Example
Numeric	Numbers, integers, decimals	100, 3.14, -7
Textual	Words, sentences, text blocks	"Customer name", "Review: great product"
Categorical	Labels or categories	Red/Green/Blue, Male/Female
Boolean	True/False, Yes/No, 0/1	true, false, 1, 0
Date/Time	Dates, timestamps	2025-08-06, 12:30 PM
Spatial	Locations, coordinates	GPS points, maps
Multimedia	Images, audio, video, graphics	profile_pic.jpg, song.mp3, video.mp4

Types of Data

C. By How It Arrives

Type	Description	Example
Batch	Collected and processed in chunks	Daily sales report, nightly backups
Streaming	Arrives and processed in real-time	Website clicks, sensor data, live chat

Sources of Data

Data can come from almost anywhere. Here are typical sources in tech and business:



Sources of Data

Data can come from almost anywhere. Here are typical sources in tech and business:

Source	Description	Example
Transactional	Systems that record daily business	Sales databases, banking systems
Operational	Logs/events from running systems	Web server logs, app logs, error logs
External APIs	Data from third-party services	Weather APIs, social media APIs, payment APIs
Manual Entry	Human input	Online forms, surveys, spreadsheets
IoT/Sensors	Physical devices measuring things	Thermometers, cameras, GPS trackers
Files/Blobs	Uploaded or shared files	CSVs, Excel, PDFs, videos
Web Scraping	Data extracted from websites	Product prices, news headlines
Public Data Sets	Open government or community data	Census data, COVID-19 stats, Wikipedia dumps
Streaming Services	Live feeds, events	Kafka, Kinesis, Event Hubs

Summary

Type	Examples
Structured	SQL databases, Excel tables
Semi-Structured	JSON, XML, CSV
Unstructured	Images, emails, audio, PDFs
Numeric	1, 2.5, -99
Textual	"Hello", "Feedback"
Categorical	"Red", "Male", "Success"
Boolean	true/false, 0/1
Date/Time	2024-05-23, 17:45
Batch	Daily ETL jobs
Streaming	Real-time clickstream, IoT
Sources	Databases, APIs, Logs, Sensors, Files, Scraping

Data Lake

What is it?

A big storage system (like S3, Azure Data Lake) for keeping raw or semi-processed data of any type (CSV, JSON, images, logs) before it's cleaned/processed.

When?

Usually the first landing place for all data from different sources, before ETL and warehousing.

ETL (Extract, Transform, Load)

What is it?

The process of taking raw data from somewhere (Extract), cleaning/restructuring it (Transform), and putting it into a storage system for later use (Load).

When do you do it?

Whenever you need to move data from source systems (like apps, logs, APIs, sensors) into a place where you can analyze or use it.

Data Pipeline

A set of processes that move data from one system to another, encompassing ETL, data movement, and integration

Data Warehouse

What is it?

A technology/approach: store processed, structured data in one big place (a “warehouse”) designed for fast analysis and business reporting.

When?

After ETL, when you want reliable, clean data ready for regular queries, dashboards, or reports.

Data Mart

A subject- or department-specific “mini-warehouse”
for focused analytics

Data Lakehouse

A modern architecture that combines the features of data lakes (flexibility, scale) and data warehouses (structure, performance) into a unified platform (sometimes called a "lakehouse").

Data Analytics

What is it?

The practice of inspecting, querying, visualizing, or exploring data to find insights, trends, or answers to business questions.

When?

After the data is available in a usable (structured/clean) form, usually from a warehouse or lake.

Business Intelligence (BI)

What is it?

The use of tools (like Power BI, Tableau, Databricks SQL) to create dashboards, KPIs, and visual reports from clean data so business users can make decisions.

When?

After data engineering and warehousing—BI is for consuming and visualizing data, not preparing it.

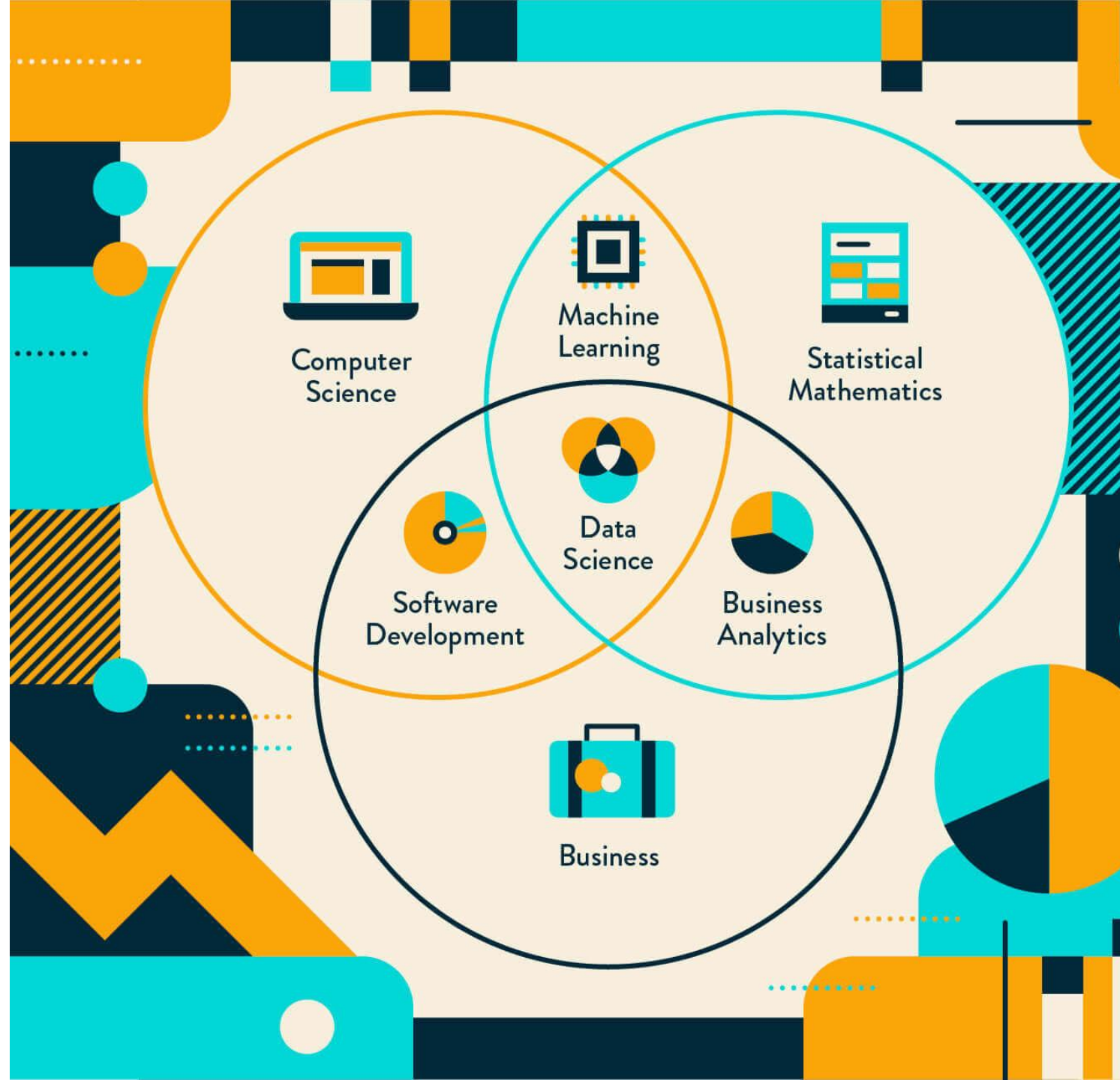
Data Science

What is it?

The field of building models, doing statistical analysis, and using machine learning on data to predict, classify, or discover hidden patterns.

When?

After you have a good supply of clean, organized data—data scientists need ready-to-use data from the engineering and warehousing layers.



Machine Learning

Machine learning is a branch of artificial intelligence (AI) that focuses on building algorithms and models enabling computers to learn from data and make predictions or decisions without being explicitly programmed for each task

Business Analytics

Purpose:

To help businesses make data-driven decisions, identify trends, solve problems, and improve outcomes (such as profits, efficiency, or customer satisfaction).

How it Works:

1. Collect business data (sales, customers, operations, marketing, etc.)
2. Analyze the data using statistical and analytical methods.
3. Visualize the results (charts, dashboards, reports).
4. Interpret insights to guide business actions or strategy.

Business Analytics Vs Business Intelligence Vs Data Analytics

Business Analytics is often used interchangeably with BI and Data Analytics, but:

- **Business Analytics** focuses more on exploring data, running analyses, and creating models to guide business choices.
- **Business Intelligence** focuses on reporting, dashboards, and visualizing current/historic performance.
- **Data Analytics** is the broader field that includes all kinds of data (not just business) and all levels (descriptive, predictive, ML, etc.).

Business Analytics Vs Business Intelligence Vs Data Analytics

Field	Main Focus	Typical Output
Business Analytics	Insights, predictions, recommendations	Reports, models, action plans
Business Intelligence	Dashboards, monitoring, reporting	Visualizations, reports
Data Analytics	All types of data, methods, and questions	Any (across industries)

Data Mesh

An architectural approach that decentralizes data ownership, letting domain teams treat their data as a product

Bigdata

Big Data refers to extremely large, complex, and rapidly growing sets of data that are difficult or impossible to process, store, or analyze using traditional (legacy) database and software tools.

In short:

Big Data = Large, fast, and diverse data that requires new tools and skills to make sense of it and unlock its value.

Bigdata

Key Characteristics of Big Data (The 5 V's):

1. Volume

1. Massive amounts of data (from gigabytes to exabytes and beyond).
2. Example: Billions of transactions, sensor readings, social media posts.

2. Velocity

1. The speed at which new data is generated and must be processed.
2. Example: Streaming live data from IoT devices or stock markets.

3. Variety

1. Different types and formats of data (structured, semi-structured, unstructured).
2. Example: Text, images, audio, video, logs, sensor data, databases.

4. Veracity

1. The quality, accuracy, and trustworthiness of the data.
2. Example: Handling missing, inconsistent, or “dirty” data.

5. Value

1. The usefulness of the data in generating business insights or innovation.
2. Example: Turning raw clickstream data into personalized recommendations.

Bigdata

Why Does Big Data Matter?

- Enables organizations to find hidden patterns, correlations, market trends, and customer preferences.
- Supports use cases like predictive analytics, real-time monitoring, AI/ML, fraud detection, healthcare analytics, etc.
- Drives better, faster, data-driven decisions.

Where Does Big Data Come From?

Social media, web and app logs, sensors (IoT), financial transactions, e-commerce, scientific research, images/videos, GPS/location, etc.

How Is Big Data Handled?

- Using distributed computing and storage (e.g., Hadoop, Apache Spark, Databricks, cloud platforms).
- Scalable NoSQL databases (Cassandra, MongoDB, etc.).
- Tools and frameworks for processing (Spark, Flink, Storm, Hive, etc.).

Data Engineering

What is it?

The profession/discipline of building systems and pipelines to move, clean, transform, and store data so it's reliable and usable for others (analysts, scientists, apps).

When?

Always at the foundation. No modern analytics or ML can happen without data engineers first organizing and preparing the data.

DataOps

The set of practices, processes, and technologies that automate data management, increase collaboration, and improve quality and cycle time in analytics and pipeline operations

AI (Artificial Intelligence)

AI (Artificial Intelligence) is a branch of computer science that focuses on creating systems or machines that can perform tasks that normally require human intelligence.

In summary:

AI = Technology that makes computers behave and “think” in ways that were once considered unique to humans.

AI (Artificial Intelligence)

Key Points about AI:

Definition:

AI enables computers to “think,” “learn,” or make decisions, often by mimicking human reasoning, problem-solving, or perception.

What Can AI Do?

- Recognize speech (like Siri or Alexa)
- Understand and process language (ChatGPT, Google Translate)
- Identify objects in images or videos (face recognition)
- Play games (chess, Go, video games)
- Make predictions (stock prices, weather)
- Control robots or self-driving cars

How Does AI Work?

- By using algorithms, data, and mathematical models to “learn” patterns or make choices.
- Often involves **machine learning (ML)**, where the system improves by analyzing more data.
- **Deep learning** is a subset of ML using neural networks (like the human brain).

Data science

What is it?

The field of building models, doing statistical analysis, and using machine learning on data to predict, classify, or discover hidden patterns.

When?

After you have a good supply of clean, organized data—data scientists need ready-to-use data from the engineering and warehousing layers.

Data governance

What is it?

The policies and controls for who can access data, data privacy, data quality, compliance, lineage, and data cataloging.

When?

Throughout the entire process! It starts as soon as you collect/store data, and is especially critical as you move, transform, and share data.

Real-time analytics

What is it?

The process of collecting, processing, and analyzing data instantly or within seconds of it being created or received.

Purpose:

To provide immediate insights, trigger alerts, or enable quick business decisions while the data is still fresh.

Examples:

- Fraud detection in banking—detect and block suspicious transactions as they happen.
- Real-time dashboards showing current website visitors or stock prices.
- Live monitoring of factory equipment for anomalies.

Streaming Analytics

What is it?

A type of real-time analytics specifically focused on continuous streams of data flowing in from sources like sensors, logs, apps, or IoT devices.

Purpose:

To analyze, filter, aggregate, or enrich data “on the fly” as it streams in, often at high volume and velocity.

Examples:

- Processing sensor data from hundreds of machines in a factory every second.
- Analyzing live social media feeds for trending topics.
- Monitoring ride-share app activity to optimize driver allocation instantly.

Batch vs. Real-Time / Streaming Analytics

	Batch Analytics	Real-Time/Streaming Analytics
When processed	After a delay (hours/days)	Instantly, as data arrives
Use case	Historical reporting, big ETL jobs	Live dashboards, fraud detection
Tools	SQL, Hadoop, nightly ETL	Spark Streaming, Kafka, Flink, Kinesis

Natural language

Natural Language refers to any language that humans use to communicate with each other—such as English, Hindi, Japanese, etc.—in speech or writing, as opposed to artificial or computer languages.

In Technology (AI/Data/Computing):

When people mention “natural language,” they usually mean natural language processing (NLP) or tasks where computers try to understand, interpret, or generate human language.

Key Points:

Natural Language (NL):

Everyday human languages that evolved naturally, not created by design.

Examples: English, Spanish, Hindi, Mandarin, etc.

Contrast:

Programming languages: Python, Java, SQL (created by humans for machines)

Markup languages: HTML, XML

Large language models

Large Language Models (LLMs) are advanced artificial intelligence models that are trained to understand, generate, and manipulate human (natural) language at scale.

In short:

Large Language Models are powerful AIs that “read” and “write” human language at scale, and are changing the way people and computers interact.

Large language models

Key Points about Large Language Models (LLMs):

- **What are they?**

LLMs are a type of AI that learns patterns in language by being trained on massive amounts of text data (books, websites, conversations, etc.).

- **How big?**

“Large” means they have billions (or even trillions) of parameters (internal settings) that help them predict and generate words and sentences.

- **What can they do?**

- Write human-like text (emails, stories, reports, code)
- Answer questions, summarize documents, translate languages
- Carry on conversations (like ChatGPT!)
- Assist with search, content creation, and more

- **How do they work?**

- Trained using deep learning (neural networks, especially “transformers”)
- Given a prompt, they predict the next word, then the next, building full sentences or paragraphs

- **Examples:**

- **OpenAI:** GPT-3, GPT-4, GPT-4o (what you’re using now!)
- **Google:** PaLM 2, Gemini
- **Meta:** Llama 2, Llama 3
- **Anthropic:** Claude

Generative AI

Generative AI is a branch of artificial intelligence that focuses on creating new content—such as text, images, audio, video, or code—that is original and resembles human-made material. It does this by learning from large datasets and then generating new data with similar patterns or styles.

Key Points about Generative AI:

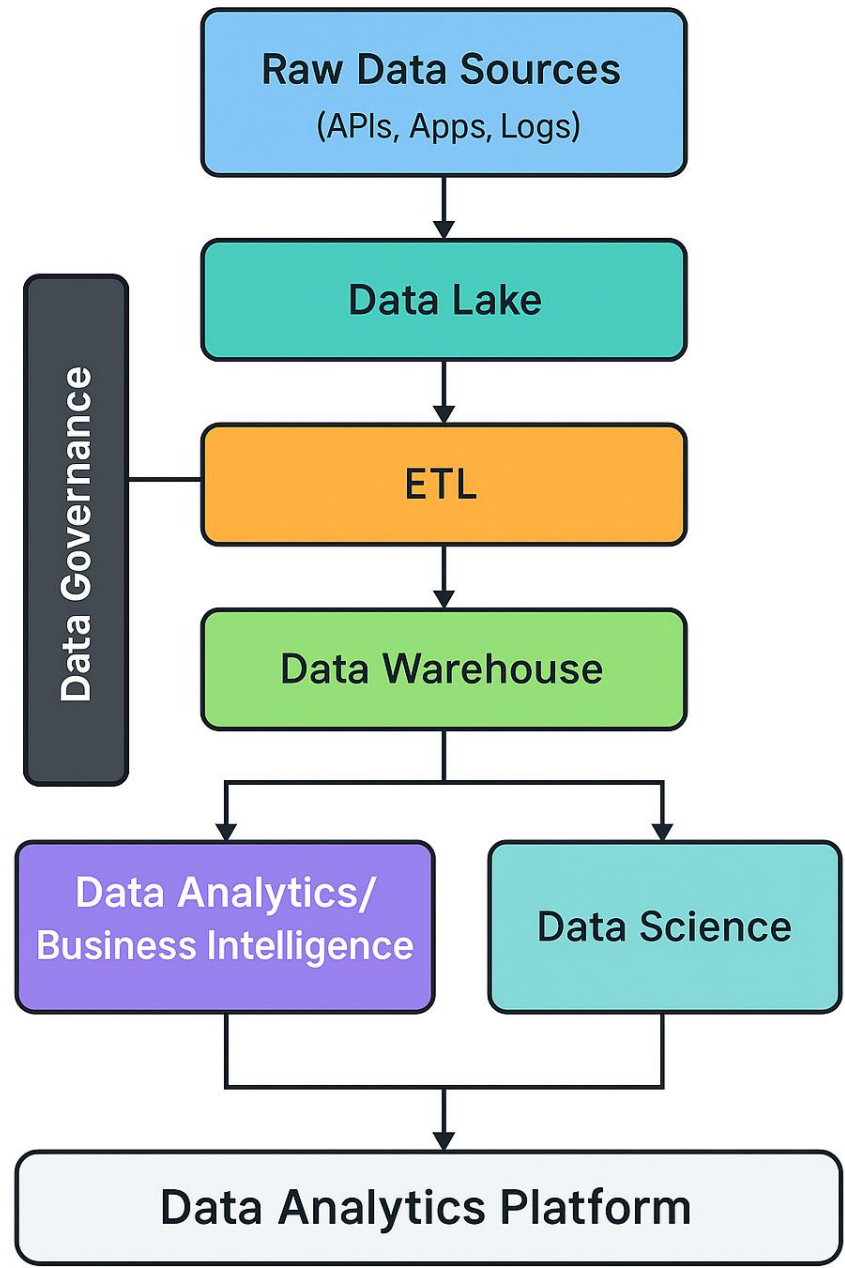
- **What does it do?**

- Creates content: writes stories, answers questions, generates images, composes music, produces videos, creates code, etc.
- Can be used for chatbots, creative design, synthetic data, code generation, and more.

- **How does it work?**

- Trained on massive datasets (e.g., billions of web pages, images, books).
- Uses advanced machine learning models, especially **deep learning** (neural networks).
- Most famous models are **Large Language Models (LLMs)** (like GPT-4, Gemini) and image generators (like DALL·E, Stable Diffusion).

Data Flow



Feature / Term	Databricks Paid Support?	Details / Notes
ETL	✅ Full Support	PySpark, SQL, Delta Live Tables, Jobs
Data Engineering	✅ Full Support	Core platform focus (pipelines, transformations, orchestration)
Data Warehousing	✅ Full Support	Databricks SQL, Delta Lake, Serverless SQL Warehouses
Data Analytics	✅ Full Support	Notebooks, SQL, dashboards, analytics workflows
Data BI	✅ Full Support	Built-in dashboards, BI tool connectors (Power BI, Tableau, Looker)
Data Science	✅ Full Support	ML Runtime, MLflow, notebooks, AutoML, Feature Store
Data Governance	✅ Full Support	Unity Catalog, audit logs, RBAC, data lineage, access controls
Data Lake	✅ Full Support	Native (Delta Lake, DBFS, cloud storage integration)
Data Analytics Platform	✅ Full Support	Databricks is a unified analytics/data platform
Data Pipeline	✅ Full Support	Jobs, Delta Live Tables, Workflows, DLT Pipelines
Data Lakehouse	✅ Full Support	Native Lakehouse engine (Delta Lake, SQL, ML, BI, governance)



Feature / Term	Databricks Paid Support?	Details / Notes
Data Mart	⚠️ Partial*	Can be built as schema/tables in Lakehouse; no dedicated “mart” product
Business Intelligence	✅ Full Support	Databricks SQL dashboards, native BI integrations
Machine Learning	✅ Full Support	MLflow, notebooks, AutoML, Model Serving
Business Analytics	✅ Full Support	BI dashboards, SQL, notebooks, ad-hoc analytics
Big Data	✅ Full Support	Built on Apache Spark, scales to petabytes, distributed compute
DataOps	✅ Full Support	Jobs API, Workflows, CI/CD, CLI, REST API, alerting, monitoring
AI (Artificial Intelligence)	✅ Full Support	ML, LLMs, GenAI (Mosaic AI), serving, experiment tracking
Real-time Analytics	✅ Full Support	Structured Streaming, Delta, DLT, live dashboards
Streaming Analytics	✅ Full Support	Spark Structured Streaming, Auto Loader, DLT
Natural Language	✅ Full Support	NLP with MLflow, Hugging Face, Python/R; LLM platform (Mosaic AI)
Large Language Models	✅ Full Support	Mosaic AI, LLM hosting, fine-tuning, inference
Data Mesh	⚠️ Partial*	Can implement with Unity Catalog, decentralization patterns



databricks

Use Databricks to accomplish tasks essential to processing, storing, and analyzing the data that drives critical business functions and decisions

PLATFORM

The Databricks Data Intelligence Platform

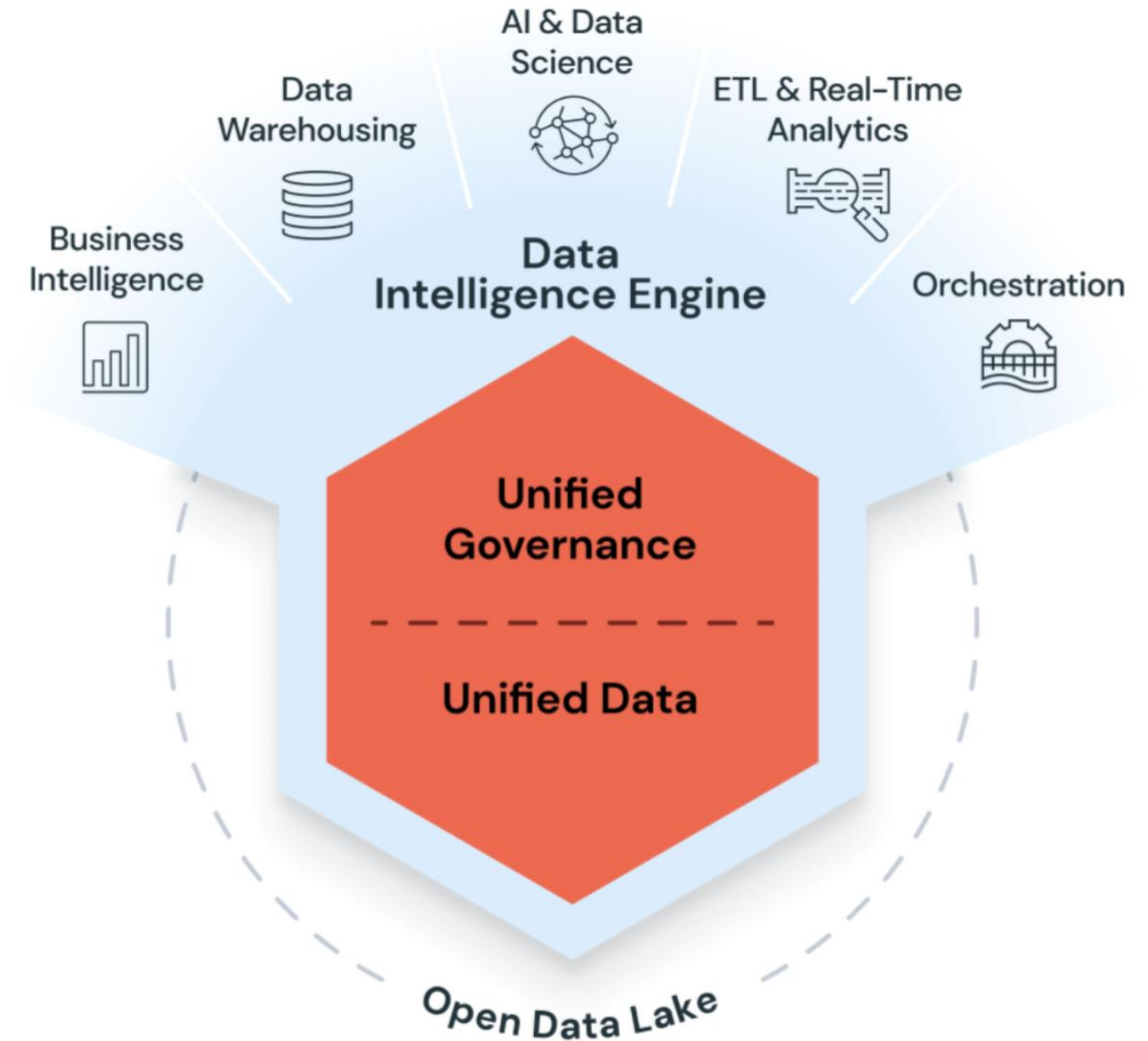
Databricks brings AI to your data to help you bring AI to the world.

What is Databricks?

Databricks is a unified, open **analytics platform** for building, deploying, sharing, and maintaining enterprise-grade data, analytics, and AI solutions at scale. The Databricks Data Intelligence Platform integrates with cloud storage and security in your cloud account, and manages and deploys cloud infrastructure for you.

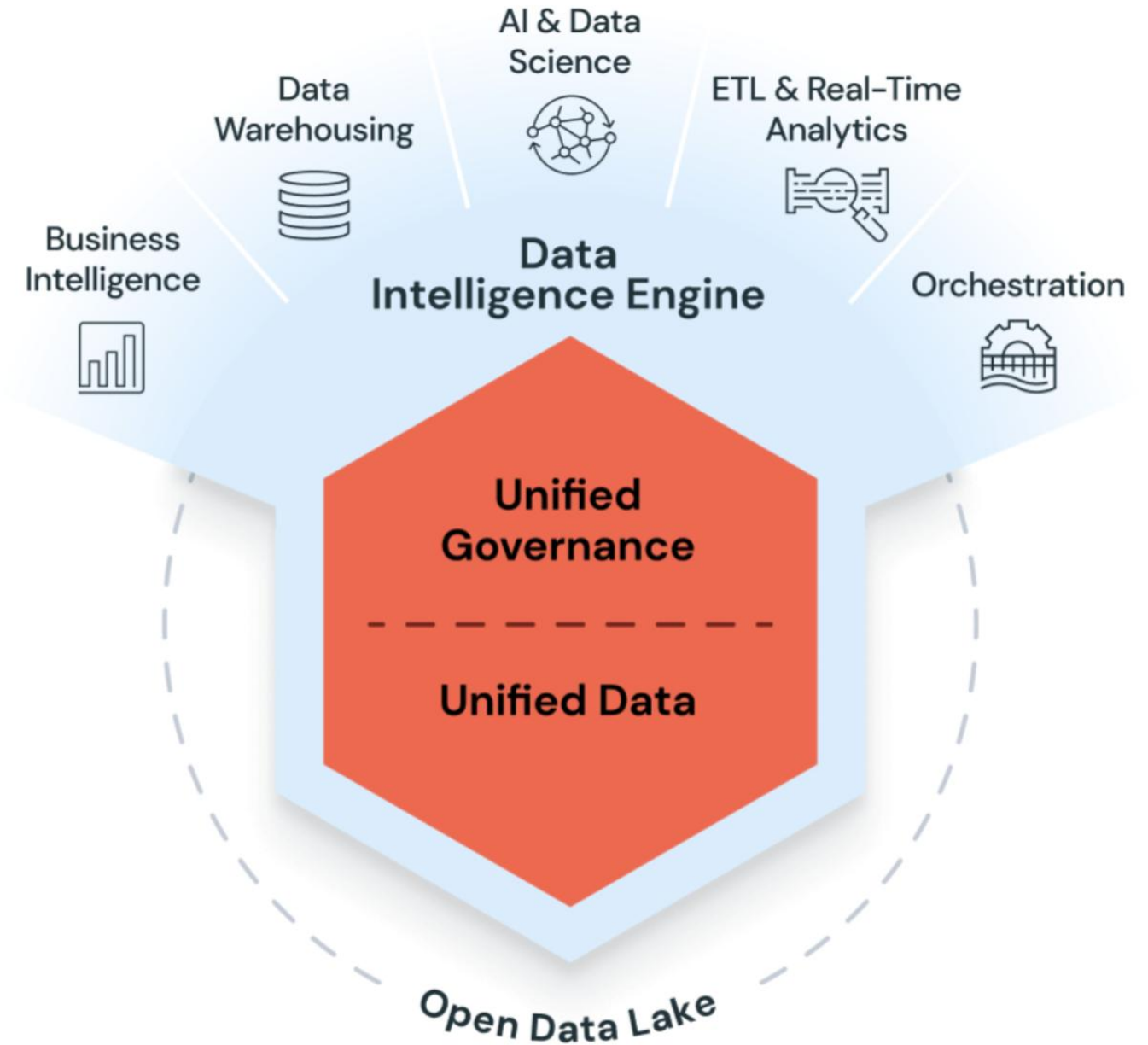
Databricks uses generative AI with the **data lakehouse** to understand the unique semantics of your data.

Then, it automatically optimizes performance and manages infrastructure to match your business needs.



Natural language processing learns your business's language, so you can search and discover data by asking a question in your own words.

Natural language assistance helps you write code, troubleshoot errors, and find answers in documentation.



Databricks Opensource Contribution

Databricks is committed to the open source community and manages updates of open source integrations with the Databricks Runtime releases. The following technologies are open source projects originally created by Databricks employees:

- Delta Lake and Delta Sharing
- MLflow
- Apache Spark and Structured Streaming
- Redash
- Unity Catalog

Databricks Use Cases

Build an enterprise data lakehouse

ETL and data engineering

Machine learning, AI, and data science

Data warehousing, analytics, and BI

Data governance and secure data sharing

DevOps, CI/CD, and task orchestration

Real-time and streaming analytics

Databricks Use Cases

Build an enterprise data lakehouse:

- Combines the strengths of enterprise data warehouses and data lakes into a unified platform.
- Accelerates, simplifies, and unifies enterprise data solutions.
- Serves as a **single source of truth** for the entire organization.
- Enables data engineers, data scientists, analysts, and production systems to access consistent, up-to-date data.
- Reduces the complexity of building, maintaining, and synchronizing multiple distributed data systems.
- Streamlines data access, improves collaboration, and supports advanced analytics and machine learning at scale.

Databricks Use Cases

ETL and Data Engineering

- Ensures data is **available, clean, and well-organized** for analytics and AI.
- Provides the backbone for all data-driven company operations.
- Databricks unifies Apache Spark, Delta Lake, and custom tools for a powerful ETL experience.
- Write ETL logic using **SQL, Python, or Scala**—and schedule jobs easily.
- **Lakeflow Declarative Pipelines:**
 - Automatically manage dependencies and deploy infrastructure for reliable, on-time data delivery.
- **Auto Loader:**
 - Quickly and efficiently ingests data from cloud storage or data lakes into the lakehouse.

Databricks Use Cases

Machine Learning, AI, and Data Science

- Databricks offers built-in tools for data scientists and ML engineers, including MLflow and Databricks Runtime for Machine Learning.
- Easily manage the entire machine learning lifecycle—experiment, track, and deploy models.
- **Large Language Models (LLMs) & Generative AI**
 - Built-in support for libraries like Hugging Face Transformers for using and fine-tuning pre-trained models.
 - MLflow integration makes it easy to track and manage transformer models and workflows.
 - Integrate with external models (e.g., OpenAI, John Snow Labs) directly into Databricks pipelines.
 - **Customize LLMs on your own data** using open-source tools like Hugging Face and DeepSpeed.
 - Data analysts can apply AI functions and use LLMs (including OpenAI models) directly from SQL and pipelines in Databricks.

Databricks Use Cases

Data Warehousing, Analytics, and BI

- Databricks offers an easy-to-use interface with scalable, cost-effective compute and storage.
- SQL warehouses allow users to run queries without cloud complexity.
- Analyze lakehouse data using the SQL editor or interactive notebooks.
- Notebooks support SQL, Python, R, and Scala.
- Create rich dashboards with visualizations, text, images, and commentary—all in one place.

Databricks Use Cases

Data Governance and Secure Data Sharing

- **Unity Catalog** delivers unified data governance for the lakehouse.
- Cloud admins set up initial access controls; Databricks admins manage team and user permissions.
- Privileges can be managed through easy UIs or SQL syntax—no need for complex cloud IAM.
- Ensures secure, compliant analytics and clear division of responsibilities.
- Sharing data internally is as easy as granting table or view access.
- Share data externally with managed **Delta Sharing** for secure, cross-organization collaboration.

Databricks Use Cases

DevOps, CI/CD, and Task Orchestration

- Centralizes all development—ETL, ML, analytics—on a single data source to reduce duplication and data drift.
- Provides tools for versioning, automation, scheduling, deployment, and production resource management.
- **Jobs** schedule notebooks, SQL queries, and arbitrary code execution.
- **Databricks Asset Bundles** enable programmatic definition, deployment, and running of resources (jobs, pipelines).
- **Git integration** allows syncing projects with popular Git providers for seamless CI/CD.
- Simplifies monitoring, orchestration, and operational overhead.

Databricks Use Cases

Real-time and Streaming Analytics

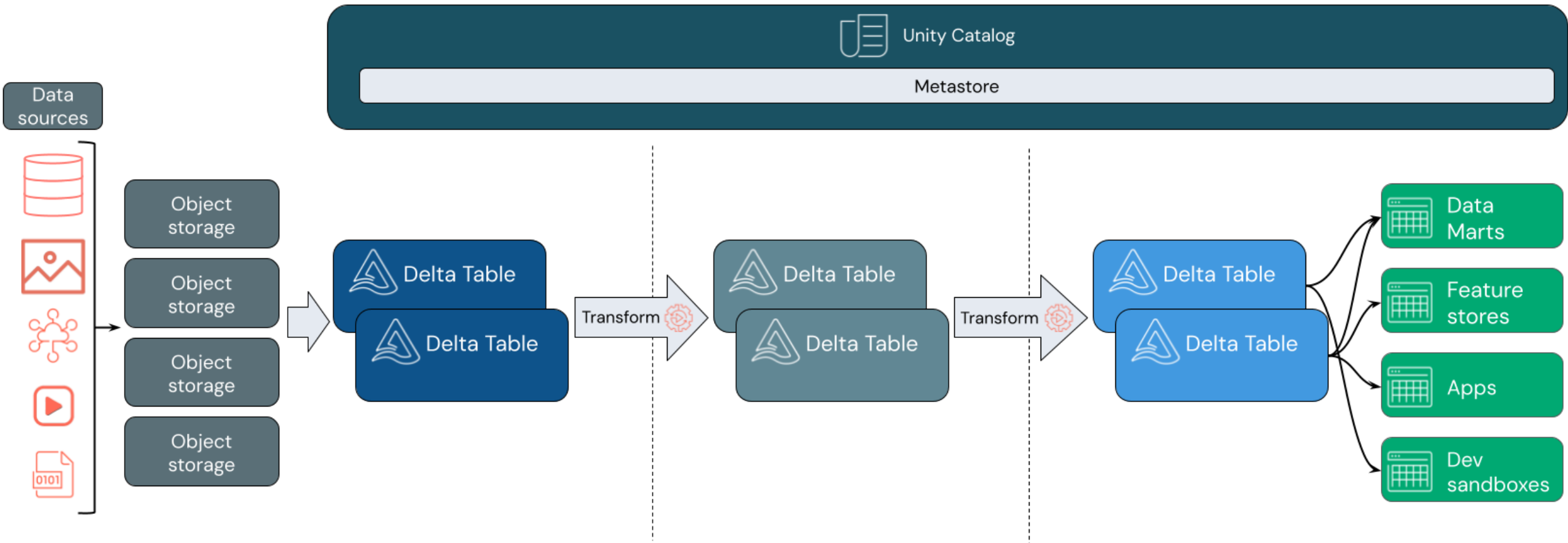
- Uses **Apache Spark Structured Streaming** to process live and incremental data.
- Deep integration with **Delta Lake** for reliability and performance.
- Powers **Lakeflow Declarative Pipelines** and **Auto Loader** for seamless, scalable streaming data pipelines.
- Enables fast, continuous analytics and real-time insights.

Data lakehouse

Data Lakehouse

A data lakehouse is a data management system that combines the benefits of data lakes and data warehouses. This article describes the lakehouse architectural pattern and what you can do with it on Databricks.

Data lakehouse



What is a Data Lakehouse Used For?

- Provides scalable storage and processing for all data workloads—analytics, machine learning, and business intelligence—on one platform.
- Eliminates isolated “data silos” and redundant systems.
- Establishes a single source of truth for the entire organization.
- Reduces costs and improves data freshness.
- Uses a layered design (medallion architecture) to incrementally refine and enrich data through multiple transformation stages.
- Supports both raw data ingestion and advanced analytics within a unified environment.

How does the Databricks lakehouse work?

Powered by Apache Spark:

Massively scalable processing engine, with compute and storage decoupled for flexibility and efficiency.

Delta Lake:

Optimized storage layer providing ACID transactions and schema enforcement for reliable, high-quality data.

Unity Catalog:

Unified governance for data and AI, with fine-grained access control, data lineage, and secure data isolation.

How does the Databricks lakehouse work?

Data Ingestion

- Handles batch and streaming data from various sources and formats.
- Raw data lands first; schema enforcement checks for quality using Delta Lake.
- Tables registered and governed via Unity Catalog; data lineage tracked from the start.

Data Processing, Curation, and Integration

- Data is curated, cleansed, combined, and enriched for analytics or ML.
- Schema-on-write and Delta schema evolution support easy changes without breaking downstream data flows.
- Data is integrated and structured for specific business needs.

Data Serving

- Delivers clean, enriched data to end users for all use cases: analytics, ML, reporting, and BI.
- Unified governance ensures secure access and clear lineage from source to consumption.
- Optimized data layouts for different workloads ensure performance and scalability.

Capabilities of a Databricks lakehouse

A lakehouse built on Databricks replaces the current dependency on data lakes and data warehouses for modern data companies. Some key tasks you can perform include:

- **Real-time data processing:** Process streaming data in real-time for immediate analysis and action.
- **Data integration:** Unify your data in a single system to enable collaboration and establish a single source of truth for your organization.
- **Schema evolution:** Modify data schema over time to adapt to changing business needs without disrupting existing data pipelines.
- **Data transformations:** Using Apache Spark and Delta Lake brings speed, scalability, and reliability to your data.
- **Data analysis and reporting:** Run complex analytical queries with an engine optimized for data warehousing workloads.
- **Machine learning and AI:** Apply advanced analytics techniques to all of your data. Use ML to enrich your data and support other workloads.
- **Data versioning and lineage:** Maintain version history for datasets and track lineage to ensure data provenance and traceability.
- **Data governance:** Use a single, unified system to control access to your data and perform audits.
- **Data sharing:** Facilitate collaboration by allowing the sharing of curated data sets, reports, and insights across teams.
- **Operational analytics:** Monitor data quality metrics, model quality metrics, and drift by applying machine learning to lakehouse monitoring data.

Lakehouse vs Data Lake vs Data Warehouse

Feature/Aspect	Data Lake	Data Warehouse	Lakehouse
Purpose	Store all raw/semi-structured data	Store clean, structured data for fast analytics	Combine the best of both: unified, flexible analytics platform
Data Types	Structured, semi-structured, unstructured	Structured (tables, columns)	All types (raw + structured)
Storage Cost	Low (object storage)	Higher (premium storage)	Low (object storage with added features)
Schema	Schema-on-read	Schema-on-write	Supports both (flexible + reliable)
Processing	Batch & streaming, but requires extra tools	Batch/real-time (highly optimized)	Batch, streaming, and advanced (unified engine)
Data Quality	Variable (raw, can be messy)	High (strict quality/enforced)	High (ACID with flexibility)
Governance	Basic	Strong (RBAC, auditing)	Enterprise-grade (fine-grained, lineage)
Analytics	Not optimized (needs extra layer)	Highly optimized (BI/SQL ready)	Optimized for BI, ML, SQL, streaming
Machine Learning	Needs integration	Possible, not native	Native ML/AI support
Typical Users	Data engineers, scientists	BI analysts, business users	All users (engineers, analysts, scientists)
Examples	AWS S3, Azure Data Lake	Snowflake, BigQuery, Redshift	Databricks Lakehouse, Delta Lake

Delta things in Databricks

What are all the *Delta* things in Databricks?

- Delta things refer to all the features and technologies built on Delta Lake in Databricks.
- Used for storing, managing, and processing data and tables in the Databricks lakehouse.
- Enable both real-time (streaming) and batch data processing with strong reliability.
- Provide ACID transactions, scalable metadata, data versioning, and efficient data updates by extending Parquet files with a transaction log.
- Support unified data management for analytics, BI, and machine learning—all in one platform.

Delta Lake

- Open-source storage layer that adds reliability and ACID transactions to cloud data lakes (S3, Azure, GCS).
- Enables data versioning and rollback for safer data management.
- Supports both batch and streaming data in a unified way.
- **Delta tables** provide an easy-to-use table interface for large-scale analytics with SQL or DataFrame APIs.

Delta Tables

- The default table format in Databricks, powered by Delta Lake.
- Ideal for data lakes with both streaming and batch data ingestion.
- Provides reliable, scalable storage and easy analytics using SQL or DataFrame APIs.

Lakeflow Declarative Pipelines: Data pipelines

- Manage and automate data flows between multiple Delta tables for ETL.
- Simplify pipeline development with a declarative, code-light approach.
- Support both batch and streaming operations—data is instantly queryable.
- Automatically handle task orchestration, cluster scaling, monitoring, and error handling.
- Enhance reliability and scalability for cloud-scale data engineering.

Delta tables vs. Lakeflow Declarative Pipelines

Delta Tables:

The core data storage format in Databricks, built for reliable, scalable table storage.

Lakeflow Declarative Pipelines:

- A pipeline framework to automate, orchestrate, and manage how data moves and transforms between Delta tables.
- Lets you define data flows declaratively; handles table creation, updates, and maintenance automatically.

In summary:

Delta Tables = How data is stored

Lakeflow Declarative Pipelines = How data moves and transforms between tables

Other *Delta* things on Databricks?

Delta Sharing

Open standard for secure data sharing—enables sharing data between organizations, regardless of platform.

Delta Engine

High-performance query optimizer for Spark SQL, Databricks SQL, and DataFrame operations—pushes computation to the data for faster results.

Delta Lake Transaction Log (DeltaLogs)

- Tracks all changes to Delta tables and guarantees atomic operations.
- Powers critical features like ACID transactions, scalable metadata, and time travel.

Databricks Components

Component: Accounts and workspaces

In Databricks, a workspace is a Databricks deployment in the cloud that functions as an environment for your team to access Databricks assets. Your organization can choose to have either multiple workspaces or just one, depending on its needs.

A Databricks account represents a single entity that can include multiple workspaces. Accounts enabled for Unity Catalog can be used to manage users and their access to data centrally across all of the workspaces in the account. Billing and support are also handled at the account level.

Component: Billing: Databricks units (DBUs)

Databricks bills based on Databricks units (DBUs), which are units of processing capability per hour based on VM instance type.

Component: Authentication and authorization

User:

An individual identity (email address) with access to Databricks.

Service Principal:

A service identity (app ID) for automating jobs, scripts, and integrations.

Group:

A collection of users and service principals for easier, centralized access management.

Access Control List (ACL):

A set of permissions defining who can access or modify Databricks assets (workspaces, clusters, jobs, tables).

Personal Access Token (PAT):

A secure token used for authenticating API calls and connecting third-party tools to Databricks.

Component: Databricks interfaces

UI:

Graphical web interface for managing workspaces, data, jobs, and clusters.

REST API:

Programmatic access to modify or retrieve information about Databricks accounts and workspace objects.

SQL REST API:

Automate and manage SQL-related tasks and queries via API.

CLI:

Command-line interface (built on the REST API) for scripting and automation, available on GitHub.

Component: Data management

Unity Catalog:

Unified governance for data and AI assets, providing centralized access control, lineage, auditing, and discovery.

Catalog:

Top-level container for organizing and isolating data across workspaces.

Schema:

Also known as databases; organizes tables, functions, models, and volumes within a catalog.

Table:

Structured data storage; queried via SQL and Spark APIs.

Delta Table: Default, ACID-compliant table format based on Delta Lake.

View:

Read-only object; saves and reuses queries on tables.

Component: Data management

Volume:

Logical storage for non-tabular (unstructured) data in cloud object storage.

Metastore:

Stores metadata about catalogs, schemas, tables, permissions, and AI assets.

Catalog Explorer:

UI for exploring, managing, and discovering data and AI assets, relationships, and permissions.

DBFS root:

Legacy storage location (now deprecated); use Unity Catalog for all new data management.

Component: Computation management

Cluster

- A set of computation resources for running notebooks and jobs.
- All-purpose clusters: Shared, interactive analysis; can be manually restarted.
- Job clusters: Created for scheduled jobs; auto-terminated after completion.

Pool

Pool of ready-to-use instances; speeds up cluster startup and scaling by reusing resources.

Databricks Runtime

- Core engine with Apache Spark plus enhanced usability, performance, and security.
- Databricks Runtime for Machine Learning: Prebuilt ML infrastructure with popular ML libraries.

Jobs & Pipelines UI

Visual interface for orchestrating and scheduling jobs, workflows, and data pipelines.

Component: Computation management

Jobs

Non-interactive, automated execution of notebooks, scripts, and tasks.

Pipelines

Lakeflow Declarative Pipelines for building reliable, testable data workflows.

Workload

Data engineering (job): Automated, runs on job clusters.

Data analytics (interactive): Manual, runs on all-purpose clusters.

Execution Context

REPL environment supporting Python, R, Scala, and SQL in notebooks.

Component: Data engineering

Data Engineering Tools

Enable collaboration across data scientists, engineers, analysts, and ML engineers.

Workspace

Central environment to organize and access all Databricks assets—folders, notebooks, dashboards, libraries, data, and compute resources.

Notebook

Web-based workspace for running code, creating visualizations, and documenting workflows.

Library

Reusable package of code for your notebooks or jobs; includes built-in and custom libraries.

Git Folder (Repos)

Sync project files with Git for source control, collaboration, and version management.

Component: AI and machine learning

AI & Machine Learning on Databricks

Integrated platform for developing, deploying, and managing ML and AI applications.

Mosaic AI

Databricks' brand for generative AI, LLMs, and cutting-edge AI features.

Machine Learning Runtime

Pre-built environment with common ML/DL libraries, GPU support, and easy compute setup.

Experiment

MLflow-based tracking of model training runs.

Feature Store

Central place to share, manage, and reuse ML features across projects.

Component: AI and machine learning

Generative AI Models

Develop and deploy foundation models, fine-tune/customize for your needs, or use third-party LLMs.

AI Playground: Chat-like environment to test and compare LLMs.

Model Registry

Central hub (via MLflow) for managing model lifecycle, access control, and discovery.

Model Serving

Deploy, govern, and query models (including LLMs) via unified REST APIs.

Component: Data warehousing

Data Warehousing

Collects and stores data from multiple sources for fast analytics and business reporting.

Databricks SQL provides high-performance data warehousing on your existing data lakes.

Query

SQL statements to interact with and analyze data, written in the SQL editor or via connectors/APIs.

SQL Warehouse

Compute resources for running SQL queries; available as Classic, Pro, and Serverless.

Component: Data warehousing

SQL Warehouse

Compute resources for running SQL queries; available as Classic, Pro, and Serverless.

Query History

Track and analyze previously executed queries for monitoring and optimization.

Visualization

Graphical representation of query results in notebooks and dashboards.

Dashboard

Combine multiple visualizations and commentary for reporting and sharing insights.

Databricks integrations

Databricks integrations: Partner Connect

Partner Connect is a user interface that allows validated solutions to integrate more quickly and easily with your Databricks clusters and SQL warehouses.

Databricks integrations: Data sources

Databricks can read data from and write data to a variety of data formats such as CSV, [Delta Lake](#), JSON, Parquet, XML, and other formats, as well as data storage providers such as Amazon S3, Google BigQuery and Cloud Storage, Snowflake, and other providers.

Databricks integrations: BI tools

In addition to access to all kinds of [data sources](#), Databricks provides integrations with ETL/ELT tools like dbt, Prophecy, and Azure Data Factory, as well as data pipeline orchestration tools like Airflow and SQL database tools like DataGrip, DBeaver, and SQL Workbench/J.

Databricks integrations: IDEs and other developer tools

Databricks supports developer tools such as DataGrip, IntelliJ, PyCharm, Visual Studio Code, and others, that allow you to programmatically access Databricks **compute**, including **SQL warehouses**.

Databricks integrations: Git

Databricks Git folders provide repository-level integration with your favorite Git providers, so you can develop code in a Databricks notebook and sync it with a remote Git repository. See [Git integration for Databricks Git folders](#).

Databricks Architecture

High-level architecture

Databricks operates out of a **control plane** and a **compute plane**.

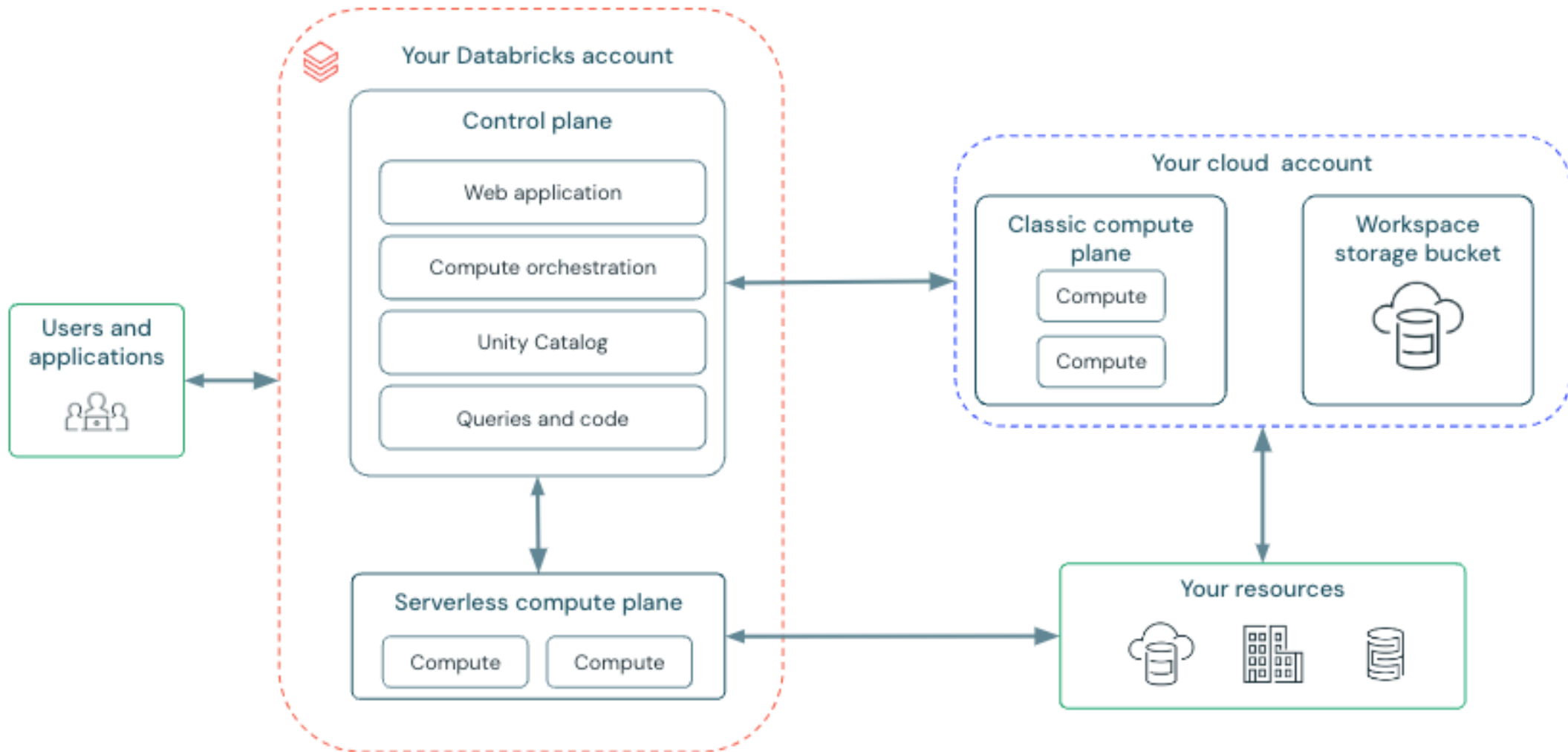
- The **control plane** includes the backend services that Databricks manages in your Databricks account. The web application is in the control plane.
- The **compute plane** is where your data is processed. There are two types of compute planes depending on the compute that you are using.
 - For serverless compute, the serverless compute resources run in a *serverless compute plane* in your Databricks account.
 - For classic Databricks compute, the compute resources are in your AWS account in what is called the *classic compute plane*. This refers to the network in your AWS account and its resources.

High-level architecture

Each Databricks workspace has an associated storage bucket known as the **workspace storage bucket**.

The workspace storage bucket is in your AWS account.

High-level architecture



Serverless compute plane

In the serverless compute plane, Databricks compute resources run in a compute layer within your Databricks account. Databricks creates a serverless compute plane in the same AWS region as your workspace's classic compute plane. You select this region when creating a workspace.

To protect customer data within the serverless compute plane, serverless compute runs within a network boundary for the workspace, with various layers of security to isolate different Databricks customer workspaces and additional network controls between clusters of the same customer.

Serverless compute plane

In the classic compute plane, Databricks compute resources run in your AWS account. New compute resources are created within each workspace's virtual network in the customer's AWS account.

A classic compute plane has natural isolation because it runs in each customer's own AWS account. To learn more about networking in the classic compute plane, see [Classic compute plane networking](#).

Workspace storage bucket

When you create a workspace, you provide an S3 bucket and prefix to use as the workspace storage bucket.

The workspace storage bucket contains:

Workspace system data: Workspace system data is generated as you use various Databricks features such as creating notebooks. This bucket includes notebook revisions, job run details, command results, and Spark logs

DBFS: DBFS (Databricks File System) is a distributed file system in Databricks environments accessible under the `dbfs:/` namespace. DBFS root and DBFS mounts are both in the `dbfs:/` namespace. Storing and accessing data using DBFS root or DBFS mounts is a deprecated pattern and not recommended by Databricks.

Unity Catalog workspace catalog: If your workspace was enabled for Unity Catalog automatically, the workspace storage bucket contains the default workspace catalog. All users in your workspace can create assets in the default schema in this catalog.

DATABRICKS PLATFORM

Platform Overview

A unified platform for data, analytics and AI

Sharing

An open, secure, zero-copy sharing for all data

Governance

Unified governance for all data, analytics and AI assets

Artificial Intelligence

Build and deploy ML and GenAI applications

Business Intelligence

Intelligent analytics for real-world data

Database

Postgres for data apps and AI agents

Data Management

Data reliability, security and performance

Data Warehousing

Serverless data warehouse for SQL analytics

Data Engineering

ETL and orchestration for batch and streaming data

Data Science

Collaborative data science at scale

Application Development

Quickly build secure data and AI apps

Databricks Pricing

What is DBU?

A Databricks Unit (DBU) is a normalized unit of processing power on the Databricks Lakehouse Platform used for measurement and pricing purposes. The number of DBUs a workload consumes is driven by processing metrics, which may include the compute resources used and the amount of data processed.



Data Engineering

Starting at \$0.15 / DBU

Orchestrate data processing, machine learning and analytics pipelines; Build streaming and batch pipelines; Ingest data from a wide variety of sources with in-built connectors

[Lakeflow Jobs](#)

[Lakeflow Declarative Pipelines](#)

[LakeFlow Connect](#)

[Learn more →](#)

Data Warehousing

Starting at \$0.22 / DBU

Run SQL queries for BI reporting, analytics and visualization to get timely insights from data lakes. Available in both Classic and Serverless (managed) Compute.

[Learn more →](#)

Interactive workloads

Starting at \$0.40 / DBU

Run interactive data science and machine learning workloads. Build and deploy custom applications with the full security and governance of the Data Intelligence Platform.

[Compute for Data Science](#)

[Databricks Apps](#)

[Learn more →](#)

Artificial Intelligence

Starting at \$0.07 / DBU

Build production-quality GenAI or ML apps across any use case

[Mosaic AI Gateway](#)

[Mosaic AI Model Serving](#)

[Mosaic AI Foundation Model Serving](#)

[Anthropic Foundation Model Serving](#)

[Shutterstock ImageAI](#)

[Mosaic AI Vector Search](#)

[Mosaic AI Agent Evaluation](#)

[Mosaic AI Model Training - fine-tuning](#)

[Mosaic AI Model Training - pre-training](#)

[Online Tables](#)

[Learn more →](#)

Operational Database

Starting at \$0.40 / DBU

Fully-managed Postgres transactional database for serving data and features to support applications built on Databricks.

[Learn more →](#)

Platform

Cross platform capabilities for governance, management and security. Managed services that automate the ongoing optimization and maintenance of your data lake

[Tiers and Add-ons](#)

[Managed Services](#)

[Data Transfer and Connectivity](#)

[Storage](#)

[Collaboration](#)

[Learn more →](#)

Databricks Opensource



Apache Spark™

Apache Spark is a unified engine for executing data engineering, data science and ML workloads.

[What is Apache Spark? →](#)

[Comparing Spark and Databricks →](#)

[Visit spark.apache.org →](#)



Delta Lake

Delta Lake lets you build a lakehouse architecture on top of storage systems such as AWS S3, ADLS, GCS and HDFS.

[Learn more about Delta Lake →](#)

[Visit delta.io →](#)

[Tech Talks: Getting Started With Delta Lake →](#)



Apache Iceberg™

Apache Iceberg lets you build a lakehouse architecture on top of storage systems such as AWS S3, ADLS, GCS and HDFS.

[Visit apache.iceberg.org →](#)



Unity Catalog

Unity Catalog is the industry's only universal catalog for data and AI.

[Learn more about Unity Catalog →](#)

[Visit unitycatalog.io →](#)



MLflow

MLflow manages the ML lifecycle, including experimentation, reproducibility, deployment and a central model registry.

[Managed MLflow on Databricks →](#)

[Visit mlflow.org →](#)

[Tech Talks: Managing the ML Lifecycle →](#)



Delta Sharing

Delta Sharing is the industry's first open protocol for secure data sharing, making it simple to share data with other organizations.

[Visit Delta Sharing →](#)



Redash

Redash enables anyone to leverage SQL to explore, query, visualize and share data from both big and small data sources.

Databricks supports these additional popular open source technologies



TensorFlow

Databricks supports TensorFlow, a library for deep learning and general computation on clusters.

[TensorFlow on Databricks →](#)



PyTorch™

Facebook, the creator of PyTorch, and Databricks have collaborated on integrations.

[PyTorch on Databricks →](#)



Keras™

Deep learning API written in Python, running on top of TensorFlow. Available in Databricks Runtime for ML.

[Keras on Databricks →](#)



RStudio

An open source suite of tools for collaborative data science using R.

[R programming on big data →](#)



scikit-learn

Widely used Python package for machine learning built on top of NumPy, SciPy and Matplotlib.

[Scikit-learn on Databricks →](#)



XGBoost

A distributed gradient boosting library that has bindings in languages such as Python, R and C++.

[XGBoost on Databricks](#)



Terraform

HashiCorp Terraform is a popular open source tool for creating safe and predictable cloud infrastructure across several cloud providers. Databricks Terraform provider allows customers to manage their entire Databricks workspaces along with the rest of their infrastructure using a flexible, powerful tool. Using Terraform also encourages customers to adopt best practices with infrastructure as code (IaC).

[Terraform on Databricks](#)

Databricks Paid Edition vs. Free Edition

Feature / Aspect	Free Edition	Paid Edition
Cost	Free	Billed (pay-as-you-go or subscription)
Cluster Size	Small, limited resources	Scalable clusters, large instance types, autoscaling
Session Limits	Limited (e.g., timeouts, max sessions)	Unlimited/longer session time
Users/Collaboration	Single user	Multi-user, team collaboration, role-based access
Data Storage	Limited storage (quota, file size)	Full cloud storage (S3, ADLS, GCS)
Compute Types	Limited compute options	All compute types (standard, high memory, GPU, etc.)
Workspaces	One per user	Multiple, with granular access control
SQL Warehouses	Limited or not available	Full SQL warehouses (Classic, Pro, Serverless)
Unity Catalog	Not available	Full data governance (Unity Catalog)
Delta Lake	Basic (for learning)	Full features, ACID, time travel, advanced options
Delta Live Tables	Not available	Yes, for production ETL pipelines
Mosaic AI/LLM	Not available or restricted	Yes, for GenAI and LLM workflows
Machine Learning	Basic support	Full ML/AI platform (MLflow, AutoML, Model Serving)
Jobs & Orchestration	Limited or not available	Full jobs, scheduling, orchestration (Workflows)
External Integrations	Minimal	Full (Power BI, Tableau, REST API, Git, etc.)
Data Sharing	Not available	Yes (Delta Sharing, cross-org/data mesh)
Security & Governance	Basic	Enterprise RBAC, audit, SSO, fine-grained control
Support	Community only	Professional support, SLAs
Production SLAs	No	Yes



Databricks Certification

Data Analyst


Data Engineer

ML Engineer

Generative AI Engineer

Additional Certifications

Data analysts transform data into insights by creating queries, data visualizations and dashboards using Databricks SQL and its capabilities.



Associate

Data Analyst

The Databricks Certified Data Analyst Associate certification exam assesses an individual's ability to use the Databricks SQL service to complete introductory data analysis tasks.

[Learn more →](#)

Data Analyst

Data Engineer

ML Engineer

Generative AI Engineer

Additional Certifications

Data engineers design, develop, test and maintain batch and streaming data pipelines using the Databricks Platform and its capabilities.



Associate

Data Engineer

The Databricks Certified Data Engineer Associate certification exam assesses an individual's ability to use the Databricks Data Intelligence Platform to complete introductory data engineering tasks.

[Learn more →](#)



Professional

Data Engineer

The Databricks Certified Data Engineer Professional certification exam assesses an individual's ability to use Databricks to perform advanced data engineering tasks.

[Learn more →](#)

[Data Analyst](#)[Data Engineer](#)[ML Engineer](#)[Generative AI Engineer](#)[Additional Certifications](#)

Machine learning engineers develop, deploy, test and maintain machine learning models and pipelines using Databricks Machine Learning and its capabilities.



Associate

ML Engineer

The Databricks Certified Machine Learning Associate certification exam assesses an individual's ability to use Databricks to perform basic machine learning tasks.

[Learn more →](#)



Professional

ML Engineer

The Databricks Certified Machine Learning Professional certification exam assesses an individual's ability to use Databricks Machine Learning and its capabilities to perform advanced machine learning in production tasks.

[Learn more →](#)

Data Analyst

Data Engineer

ML Engineer

Generative AI Engineer

Additional Certifications



Associate

Generative AI Engineer

The Databricks Certified Generative AI Engineer Associate certification exam assesses an individual's ability to design, build, and deploy Generative AI solutions with Databricks

[Learn more →](#)

Data Analyst

Data Engineer

ML Engineer

Generative AI Engineer

Additional Certifications



Associate

Apache Spark™ Developer

The Databricks Certified Associate Developer for Apache Spark certification exam assesses the understanding of the Spark DataFrame API.

[Learn more →](#)

Thank you!



- I hope you enjoyed this session.
 - I encourage you to leave feedback@ Trustpilot -
<https://www.trustpilot.com/review/devopsschool.com>
 - and ask questions
 - Good luck!
- Rajesh Kumar**
www.RajeshKumar.xyz
DevOps@RajeshKumar.xyz