

# Glossary

## Glossary of Terms associated with databricks

**ADLS2** - Azure Data Lake Storage Gen2; hierarchical namespace-enabled blob storage used for analytics.

**autoloader** - Databricks feature that incrementally ingests new files from a cloud storage path.

**autoscaling** - automatic resizing of cluster worker count based on workload demand.

**Azure Key Vault** - Azure-managed secret store used for storing credentials, keys, and SAS tokens for DBX access.

**Azure Managed Identity** - Identity assigned to a DBX resource to securely access Azure services without secrets.

**barcode** - internal Databricks deployment identifier for clusters, notebooks, and jobs.

**batch processing** - processing data in grouped chunks rather than continuous streams.

**blob storage** - Azure term for inexpensive storage of any type of file. Amazon term is S3

**bronze layer** - raw ingestion layer in the medallion architecture; contains minimally processed source data.

**catalog** - top-level container in Unity Catalog that organizes schemas and tables.

**checkpoint** - directory used by Structured Streaming and Autoloader to track state for incremental processing.

**cluster mode** - controls how local vs remote driver processes run (single-node, standard, high-concurrency).

**cluster policies** - governance rules that restrict cluster configurations that users are allowed to create.

**cloudFiles** - Autoloader's file discovery mechanism API for streaming ingestion.

**compute plane** - where the VM nodes actually run—notebooks, jobs, worker tasks, shuffle operations.

**concurrency** - number of simultaneous queries a cluster can process; high-concurrency clusters optimize this.

**control plane** - the "brains" of DBX. The DBX UI is in the control plane and issues commands to the node(s) in the data plane

**copy-on-write** - Delta Lake mechanism where updates create new versions of modified Parquet files.

**data lineage** - Unity Catalog-tracked history of data flows between tables, notebooks, and jobs.

**data plane** - location of the user's data - typically in blob storage container in an Azure storage account

**data skipping** - Delta optimization that uses statistics in data file metadata to prune unnecessary

files during reads.

**Databricks Connect** - tool allowing you to run local IDE code against a remote Databricks cluster.

**Databricks Runtime** - versioned environment that defines Spark version, Delta version, and libraries on a cluster.

**DBX** - abbreviation of databricks

**Delta Lake** - storage layer that adds ACID transactions, schema enforcement, and time travel to Parquet.

**Delta Live Tables** - declarative ETL framework in DBX for building pipelines with quality checks and event tracking.

**Delta Log** - `__delta_log` folder containing JSON transaction files tracking commits and table versions.

**driver node** - orchestrates Spark tasks, maintains metadata, and coordinates work among executors/workers.

**ETL** - extract, transform, load; standard data integration pipeline pattern.

**event grid** - Azure service that publishes notifications for blob storage events to trigger ingestion.

**executor node** - worker process in a cluster that performs the actual computation for tasks.

**Hive metastore** - legacy metadata catalog used prior to Unity Catalog, scoped per workspace.

**instance pool** - pre-warmed VMs used to speed up Databricks cluster startup times and reduce cost.

**job cluster** - ephemeral cluster automatically spun up for a job, then terminated after completion.

**lakehouse** - unified architecture combining data lake storage and data warehouse capabilities.

**manifest file** - JSON export of a Delta table for external tools not natively Delta-aware.

**materialized view** - table whose results are precomputed and refreshed to improve query performance.

**medallion architecture** - bronze → silver → gold tiered data modeling design for incrementally refined datasets.

**metastore** - metadata service containing catalogs, schemas, tables, permissions, and lineage.

**MLflow** - open-source model tracking and experiment management system native to Databricks.

**multi-cluster warehouse** - SQL warehouse that can automatically scale out with multiple clusters for concurrency.

**notebook** - interactive coding environment for Scala, SQL, Python, or R inside the Databricks workspace.

**optimize** - Delta table maintenance operation to coalesce small files and improve performance.

**parquet files** - columnar data file format that is compressed. Contains column headers, data types, and some metadata.

**photon** - next-generation Databricks execution engine written in C++ for faster SQL performance.

**pipelines** - automated workflows in DBX Jobs or Delta Live Tables for scheduled ETL processes.

**power BI connector** - direct connection option between Power BI and Databricks SQL warehouses.

**query profile** - graphical explanation of query stages, tasks, and performance characteristics.

**RBAC** - role-based access control; permissions model in Unity Catalog for fine-grained governance.

**schema** - logical grouping of tables within a catalog (similar to a database schema in SQL Server).

**schema enforcement** - Delta Lake's ability to prevent writes that violate expected column types or names.

**schema evolution** - Delta Lake's automated capability to add new columns when enabled.

**serverless SQL** - Databricks-managed SQL compute with instant start and no cluster management.

**shallow clone** - lightweight metadata-only clone of a Delta table referencing the same underlying

data files.

**silver layer** - cleaned and conformed data layer in the medallion architecture.

**spark** - distributed compute engine used under the hood by Databricks for parallel processing.

**SQL warehouse** - compute resource optimized for SQL workloads, formerly called SQL endpoints.

**table ACLs** - access controls that regulate who can query, modify, or manage tables.

**table history** - list of previous Delta table versions with timestamps and operations.

**task** - unit of work within a Spark stage executed on a worker.

**time travel** - Delta feature that lets you query older versions of a table via version number or timestamp.

**token** - workspace-specific personal access credential used to authenticate external tools to DBX.

**Unity Catalog** - centralized governance layer for permissions, metadata, auditing, and lineage.

**UDF** - user-defined function; custom Python/Scala logic registered for use in SQL.

**UDI (Update/Delete/Insert)** - mutation operations that modify Delta tables atomically.

**vacuum** - Delta operation that permanently deletes old versions and files older than a retention threshold.

**view** - saved SQL query definition that appears as a table but does not store its own data.

**volume** - UC-governed directory for unstructured files, supporting data and code assets.

**widget** - Notebook UI control (dropdowns, text boxes) enabling parameterization of jobs and dashboards.

**workflow** - job-based orchestrated set of tasks, dependencies, and triggers.

**worker node** - cluster node that executes Spark tasks and holds shuffled data.

**Z-order** - file-level clustering technique in Delta to co-locate related values for faster reads.

---

Revision #2

Created 1 December 2025 23:30:22 by Brian Dill

Updated 1 December 2025 23:51:13 by Brian Dill