Prepared by Pavani Korada

Databricks

Databricks is a cloud platform used to work with big data and AI.

Get started with Data Engineering

About Databricks:

UseCases

Used to:

- Handle Big Data Easily
- Faster AI/ML Development
- Unified Platform
- Real-Time Analytics

In which domains Databricks Is used

- Healthcare
- Banking & Finance
- Telecom
- Marketing & Advertising
- Retail & E-commerce

Which type of companies use databricks??

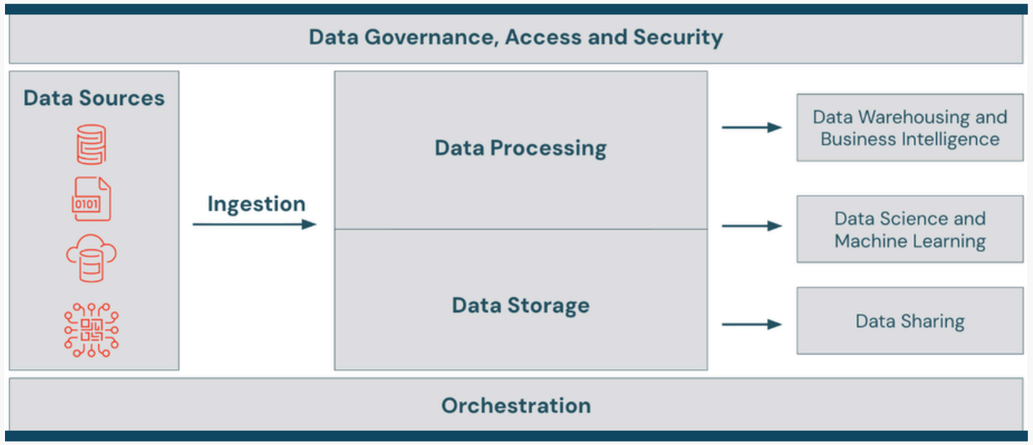
- Technology & Digital-Native Firms
- Retail & Consumer Goods
- Financial Services & Banking
- Manufacturing & Automotive
- Healthcare & Life Sciences

Data Engineering Basics

Transforming raw data to clean, reliable data	Ensuring the quality and integrity of data	Design, build, and maintain data pipelines

Data Engineer Responsibilities

- Transform raw data into clean, structured, reliable data
- Perform data extraction, cleansing, and transformation (ETL)
- Ensure data quality, accuracy, and integrity
- Design, build, automate, and maintain data pipelines



Data Engineering Architecture

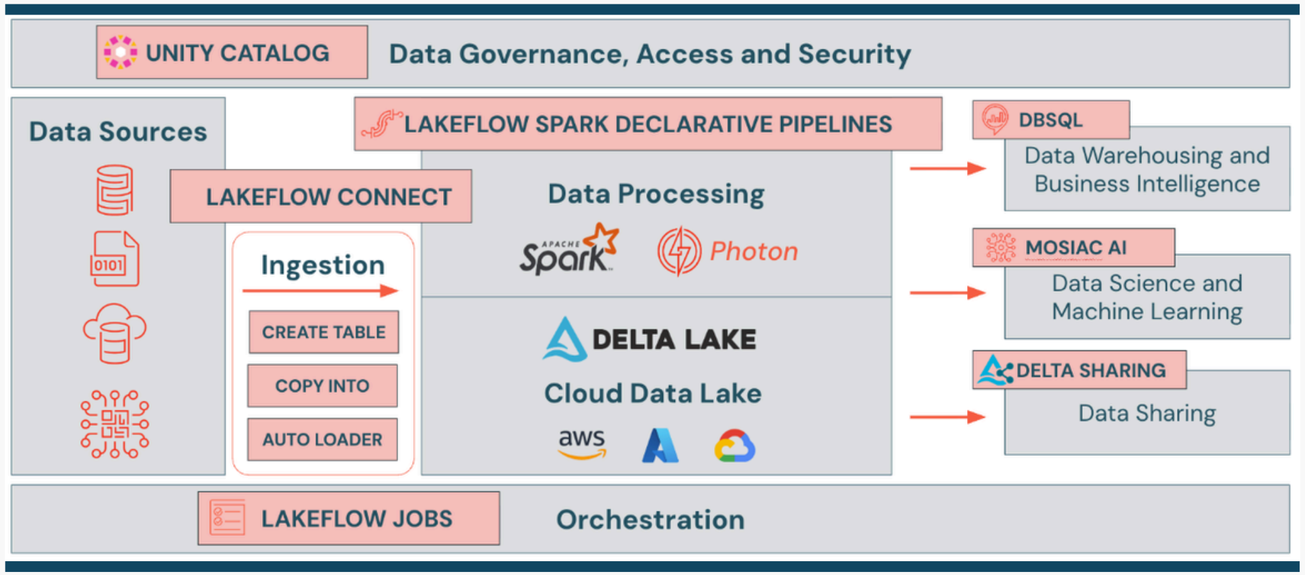
- Data Sources – Databases, cloud storage, logs, files
- Data Ingestion – Load data into data lakes/warehouses
- Data Processing – Clean and transform data
- Data Availability – Provide data for ML and analytics
- Orchestration & Governance – Automate workflows, manage security and access

Common Challenges:

Complex data ingestion methods

Managing key data engineering principles

To address these challenges, many organizations are turning to unified platforms like Databricks.



Databricks Data Intelligence Platform (Solution)

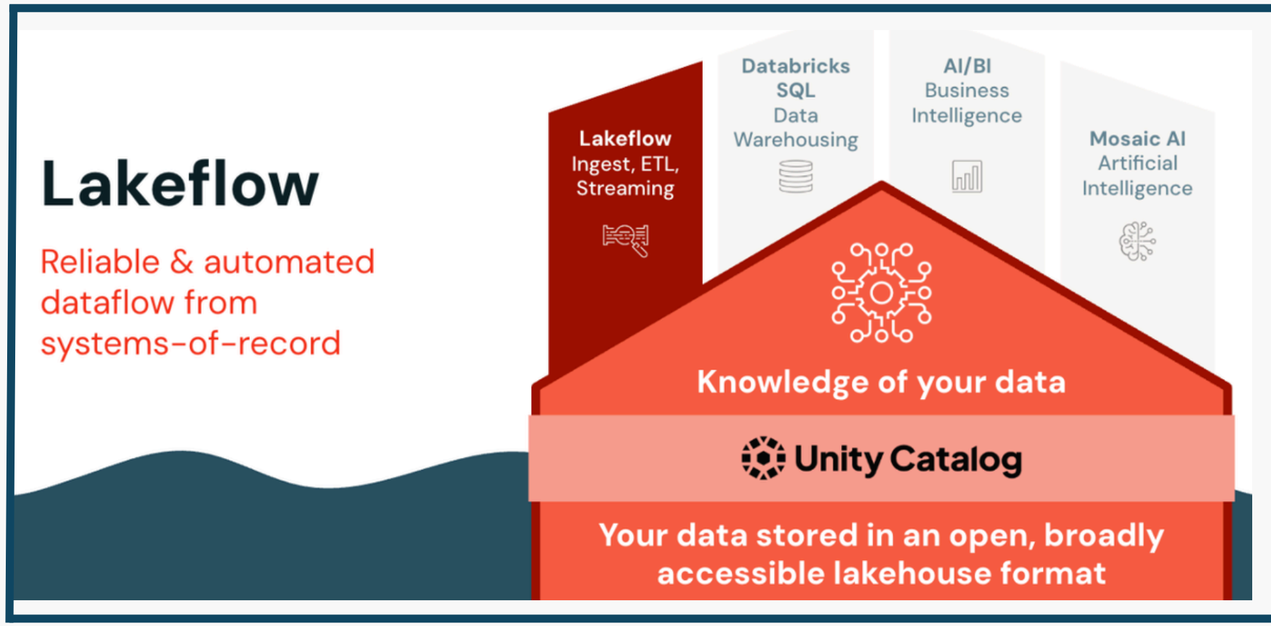
- Unified platform for ingestion, processing, storage & analytics
- Delta Lake for reliable lakehouse architecture
- Lakeflow for automated ETL & orchestration
- DBSQL (BI), Mosaic AI (ML), Delta Sharing (data sharing)
- Unity Catalog for governance, security & access control

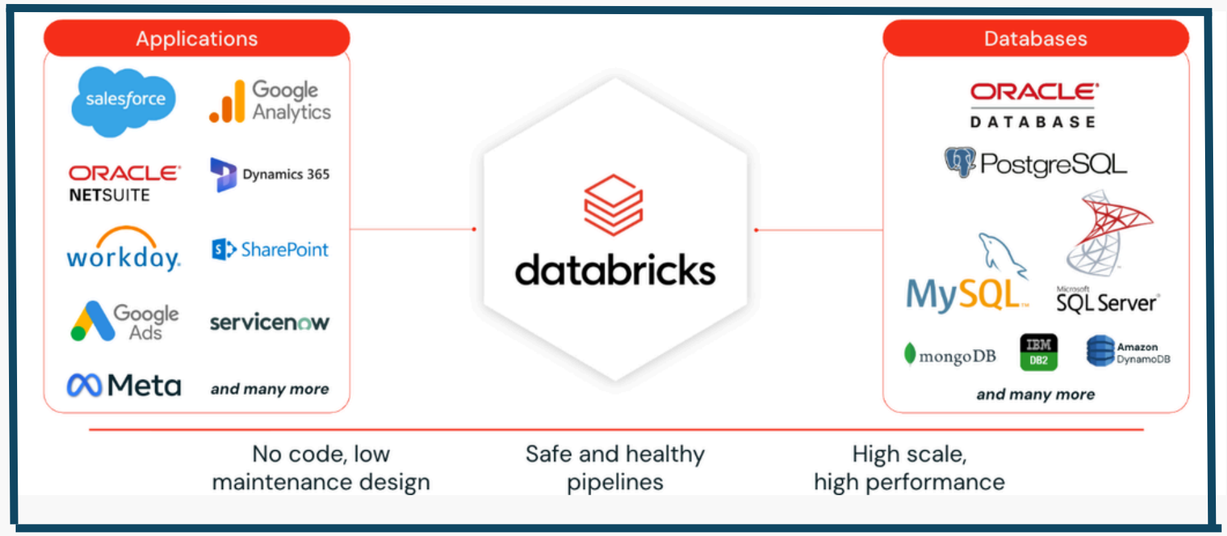
Intro to Lakeflow

Lakeflow is a unified set of tools in Databricks that manages data from ingestion to delivery in a reliable, automated way.

Lakeflow consists of three powerful components:

- Lakeflow Connect
- Lakeflow Spark
- Lakeflow Jobs





Lakeflow Connect (Ingestion)

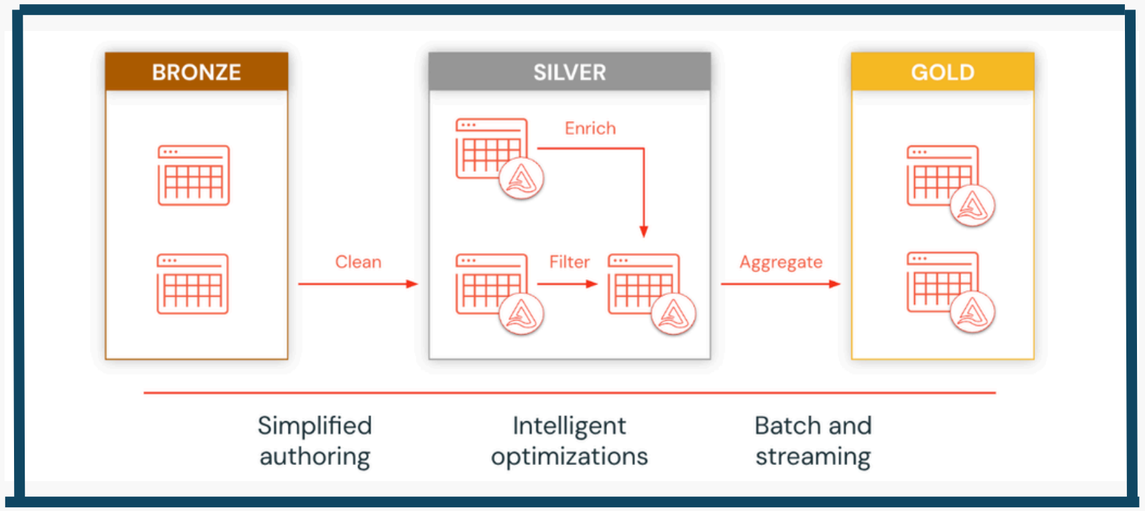
- No-code & low-code data ingestion
- Managed and standard connectors
- Ingest from applications, databases, and third-party systems
- Supports batch and streaming data

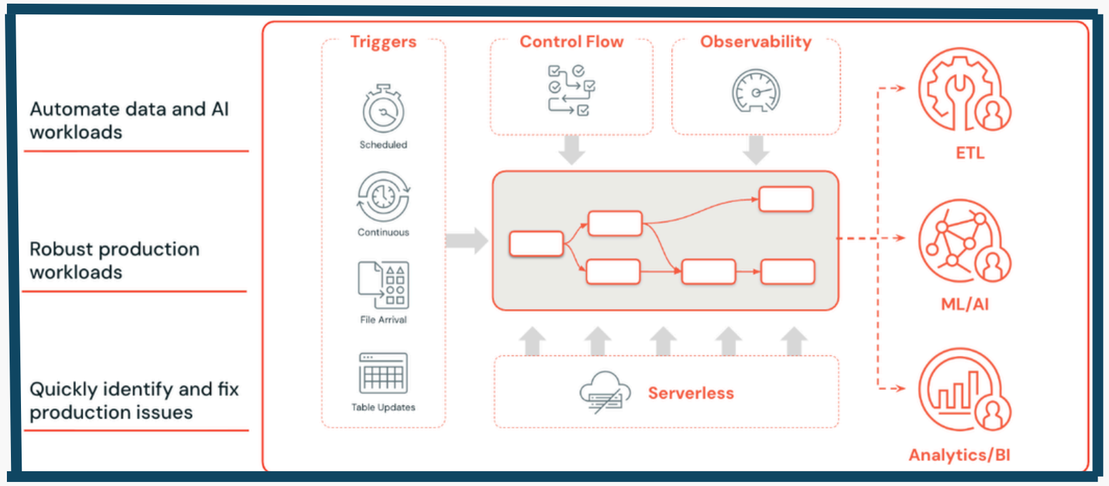
Lakeflow Spark Declarative Pipelines (ETL & Transformation):

- Build automated ETL pipelines
- Follows Medallion Architecture:

1. **Bronze** – Raw data
2. **Silver** – Cleaned & validated data
3. **Gold** – Aggregated, business-ready datasets

- Ensures data quality and reliable transformations





Lakeflow Jobs (Orchestration):

- Automates workflows with task dependencies
- Supports triggers (scheduled, continuous, file-based, etc.)
- Enables ETL, ML/AI, and Analytics workloads
- Serverless and production-ready

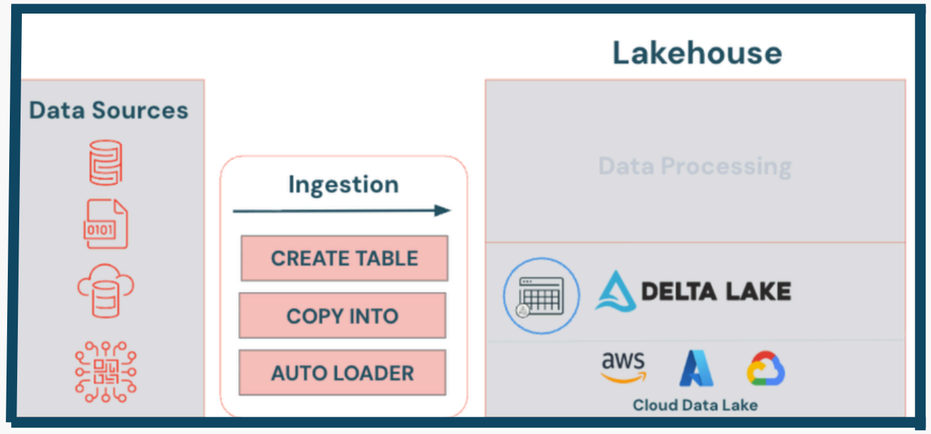
Ingestion with Lakeflow Connect

Upload via UI

- Upload CSV, JSON, Avro, Parquet, Text files directly
- Best for quick analysis or small datasets
- Not suitable for large-scale or streaming workloads

COPY INTO

- Loads files from cloud storage into Delta tables
- Idempotent – skips already ingested files
- Supports multiple formats and schema changes
- Good for regular incremental loads



syntax:

```
CREATE TABLE mydeltatable
USING DELTA    -- Optional
AS
your query
```

syntax:

```
COPY INTO mydeltatable
FROM 'your-path'
FILE_FORMAT = 'format'
FILE_OPTIONS = ('format-options')
```

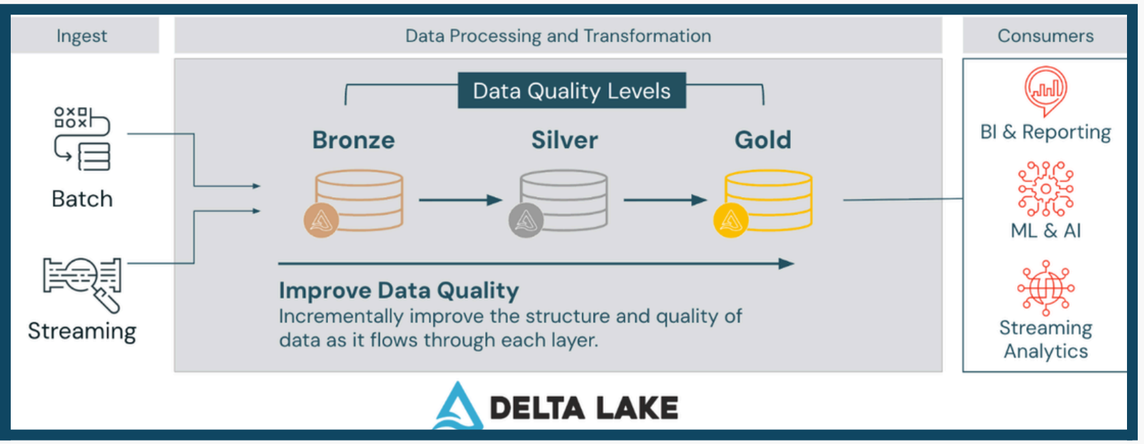
CREATE TABLE AS (CTAS)

- Creates a Delta table from a SELECT query
- Delta format is default (USING DELTA)
- Ideal for full batch loads (replacing table each time)

Auto Loader

- Automatically ingests new files as they arrive
- Highly scalable for large datasets
- Handles schema evolution
- “Rescues” unexpected or mismatched data

Data Transformation in Databricks – Medallion Architecture



Data Processing Engine

- Built on Apache Spark (open-source, in-memory, distributed engine)
- Supports batch & streaming, SQL, Python, Scala
- Photon (optional) → Faster query performance & optimized Delta processing

Delta Lake matters because it provides **ACID** transactions (atomic & reliable updates), versioning & time travel, and supports DELETE, UPDATE, MERGE with SQL, Python, Scala support.

Bronze (Raw Layer)

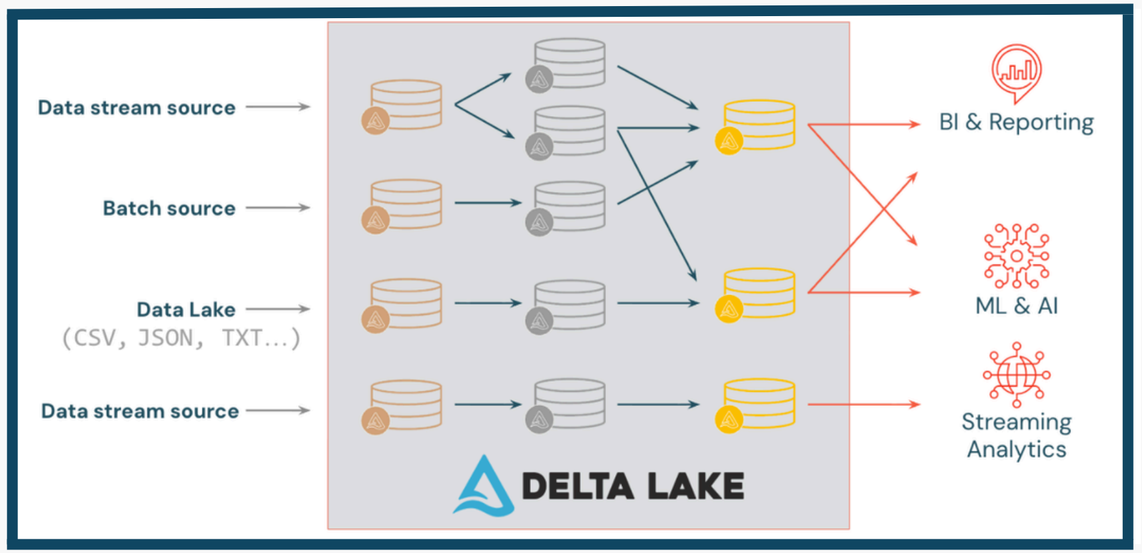
- Ingest raw data from batch, streaming, or data lakes
- Stored as-is
- Protects against transformation bugs
- Serves as historical record / source of truth

Silver (Cleaned Layer)

- Clean, filter, join, enrich data
- Apply schema enforcement & data quality rules
- Fix errors, apply business logic
- Becomes enterprise “single source of truth”

Gold (Consumer Layer)

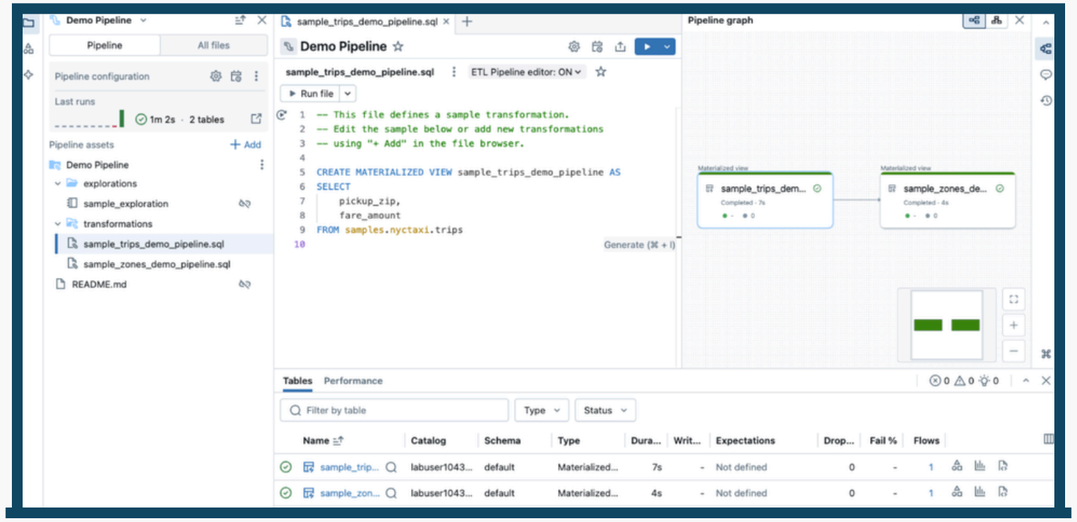
- Business-ready, aggregated datasets
- Optimized for BI reports, ML/AI, dashboards
- Designed for specific use cases & downstream consumption

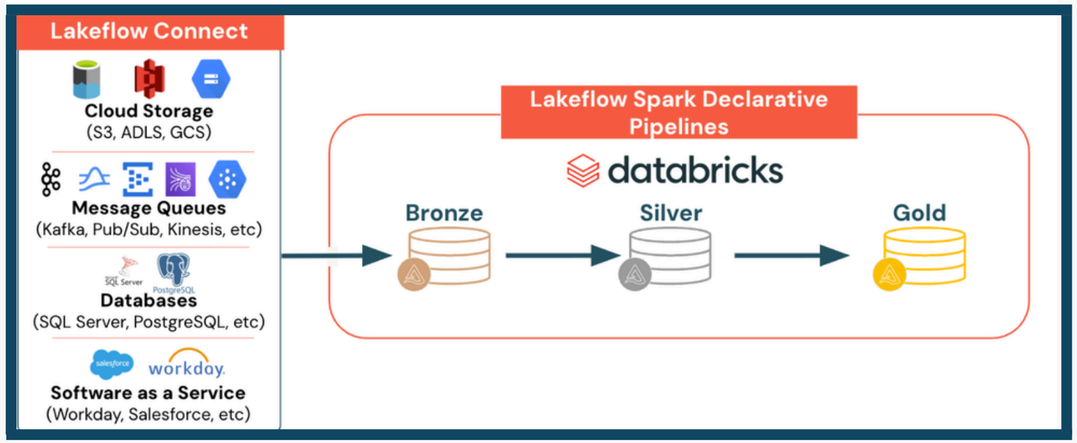


ETL with Lakeflow Spark Declarative Pipelines

It Provides:

- Declarative ETL using SQL or Python
- Automatic orchestration, dependency management & error handling
- Built-in observability and data quality tracking
- Automatic scaling & failure recovery
- Unified batch + streaming



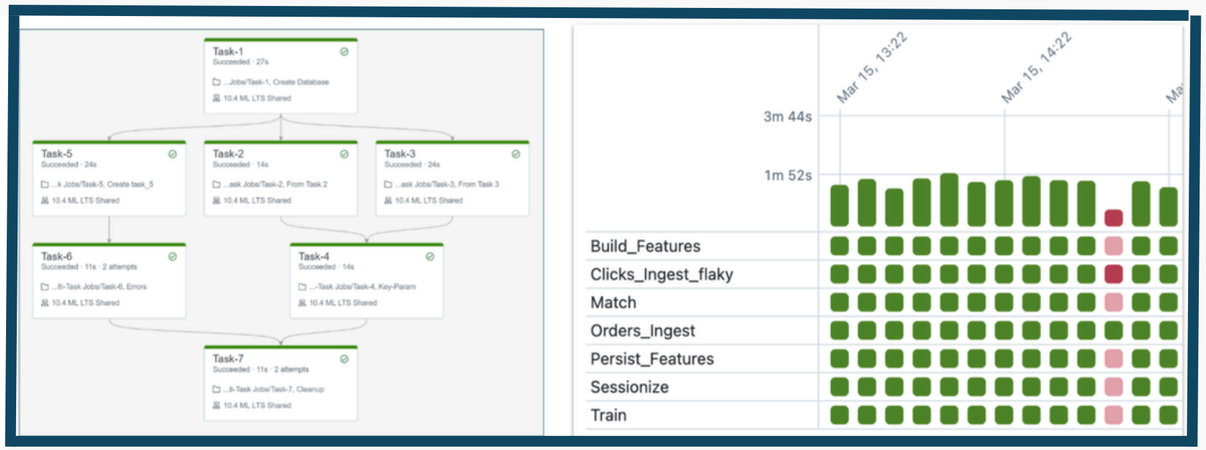


Data Ingestion (via Lakeflow Connect)

Supports ingestion from:

- Cloud storage (S3, ADLS, GCS)
- Message queues (Kafka, Pub/Sub, Kinesis)
- Databases (SQL Server, PostgreSQL, etc.)
- SaaS apps (Salesforce, Workday)

Orchestration with Lakeflow Jobs

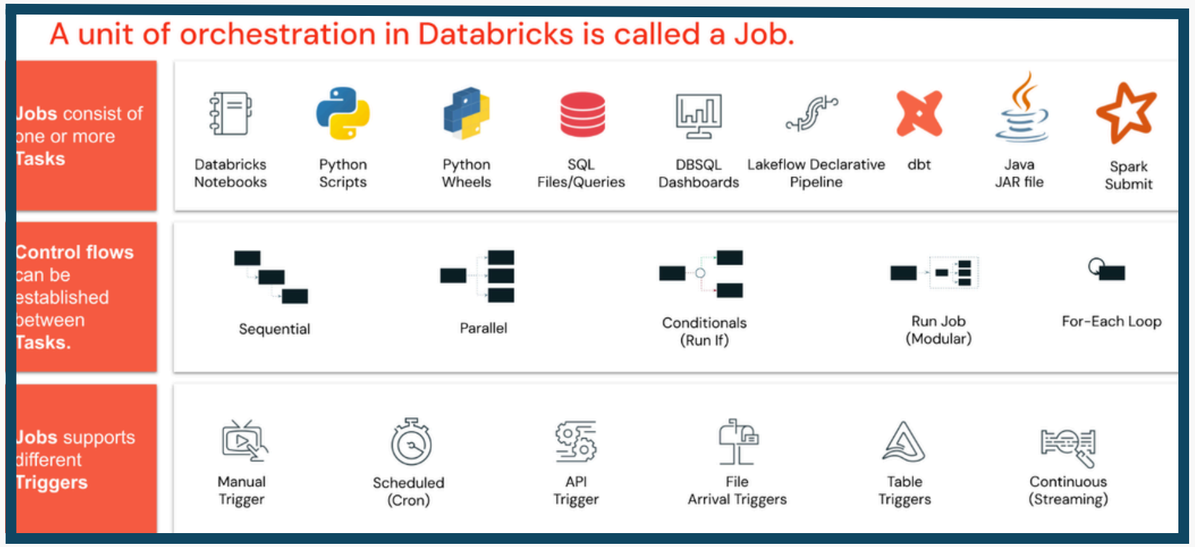


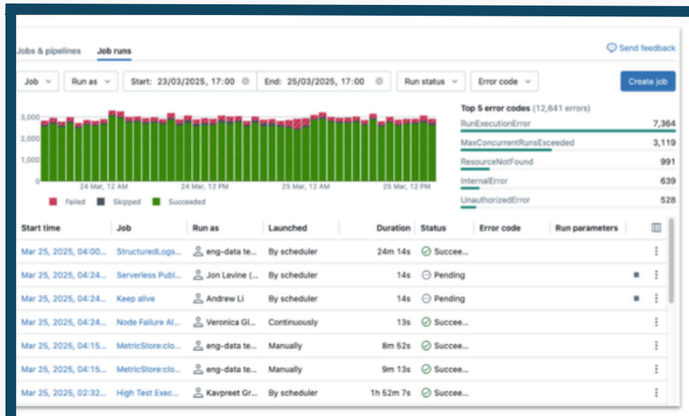
Lakeflow Jobs:

- Unit of orchestration in Databricks
- Automates execution of notebooks, scripts, SQL, pipelines & more
- Fully cloud-based – can automate anything with APIs
- Supports Data Engineering, ML, and GenAI workflows

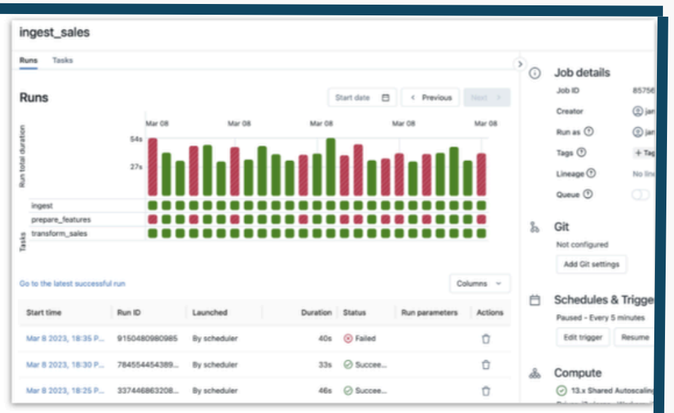
Building a Lakeflow Job begins with:

- Tasks – What to execute
- Control Flow – Sequential, Parallel, Conditional, For-Each, Run Job
- Triggers – Manual, Scheduled, API, File Arrival, Table, Continuous





A simple and intuitive monitoring UI provides real-time metrics and detailed analytics for every workflow run



Drill down to understand which tasks are failing and why. Troubleshoot issues before your customers are impacted

Real-Time Monitoring

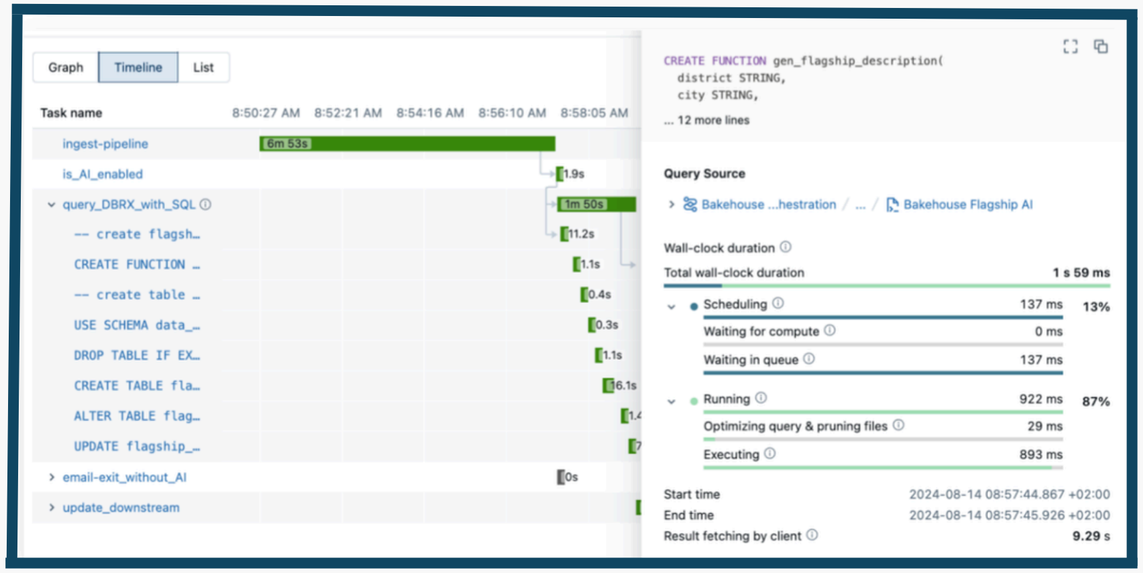
- Visual workflow graph of task dependencies
- View all job runs with status & duration
- Identify failures and retries instantly

Deep Observability

- Drill down into individual tasks
- Timeline (Gantt) view of job runs
- See dependencies clearly
- Identify slow-running tasks

Query Insights

- View executed queries per task
- Analyze scheduling vs execution time
- Optimize performance bottlenecks



Thank you