databricks

The Databricks Platform Introduction

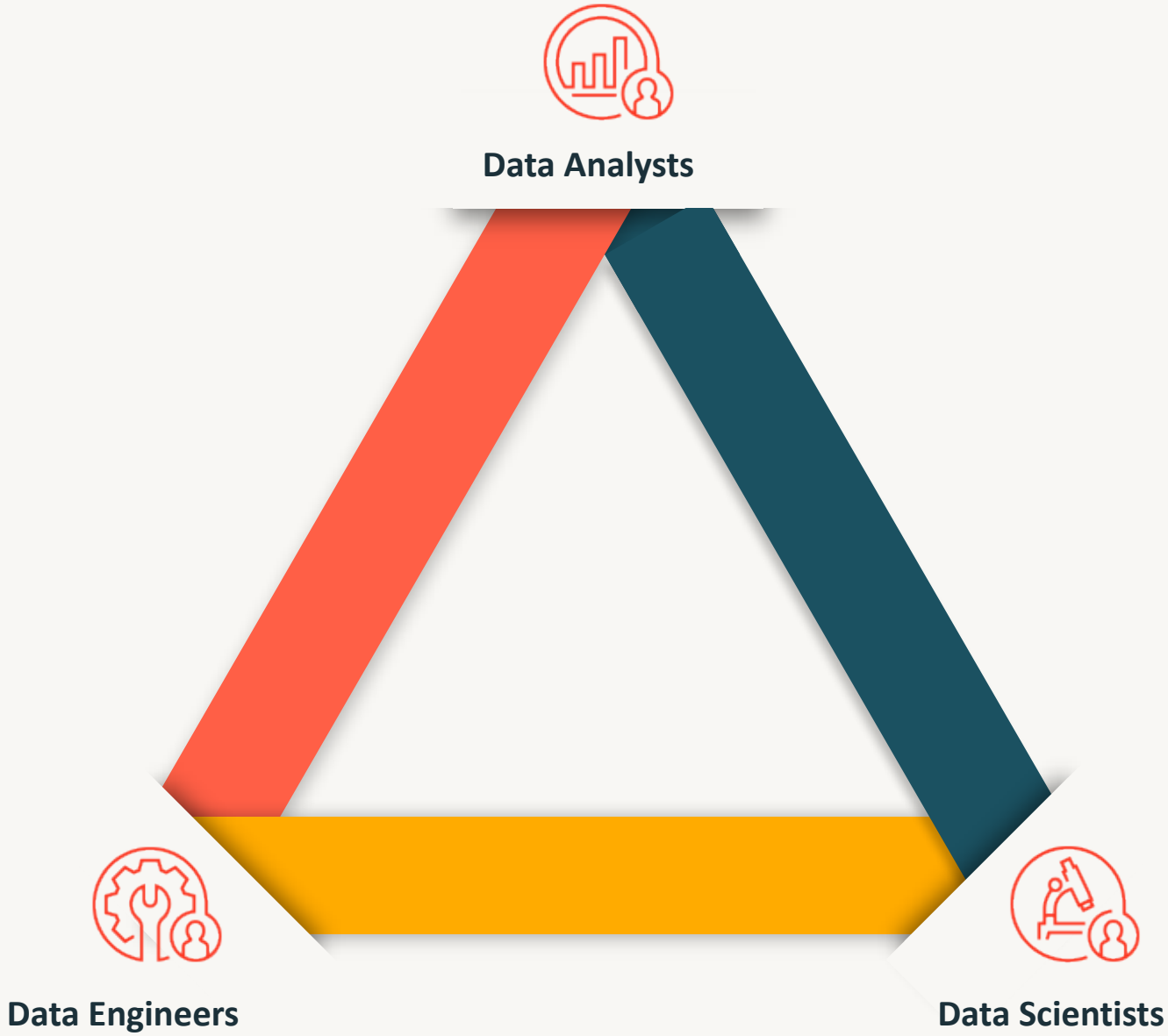
All your data, analytics and AI on one platform

March 2024

What is DataBricks ?

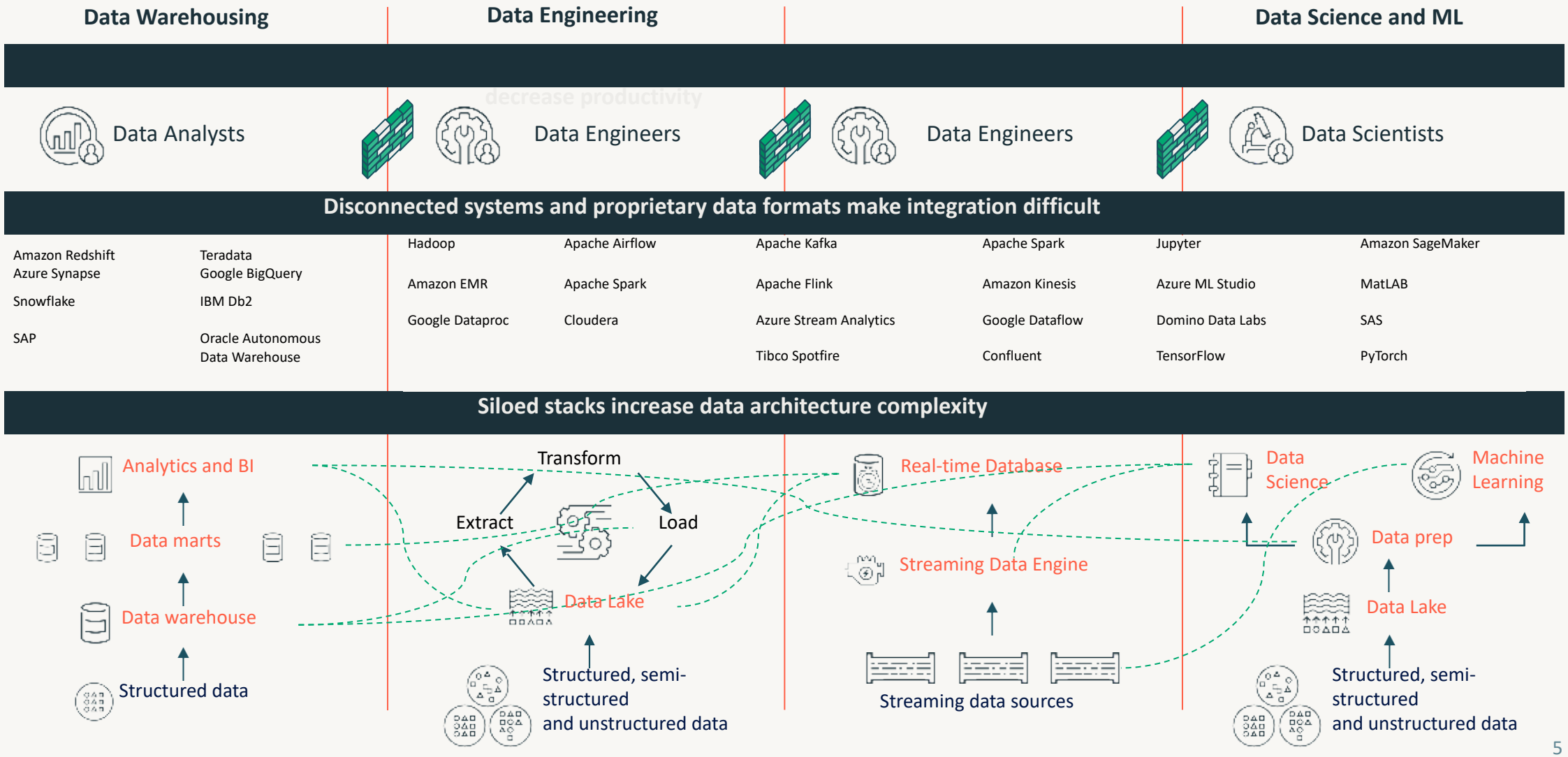
DataBricks is a unified & open Data and Analytics Platform

Modern Data Teams



How the data management looks like today ?

Data management complexity



VS.

Data Warehouse

Data Lake

Warehouses and lakes create complexity

Two separate copies of the data

Warehouses	Lakes
Proprietary	Open

Incompatible interfaces

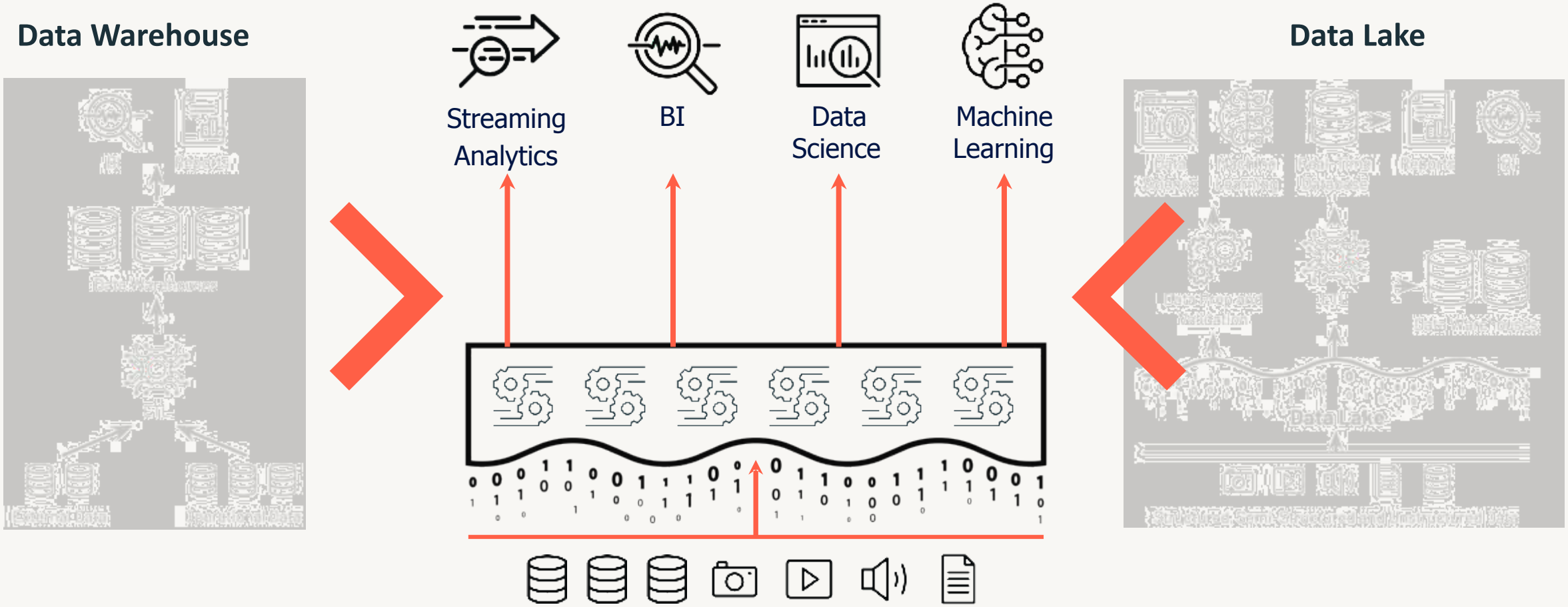
Warehouses	Lakes
SQL	Python

Incompatible security and governance models

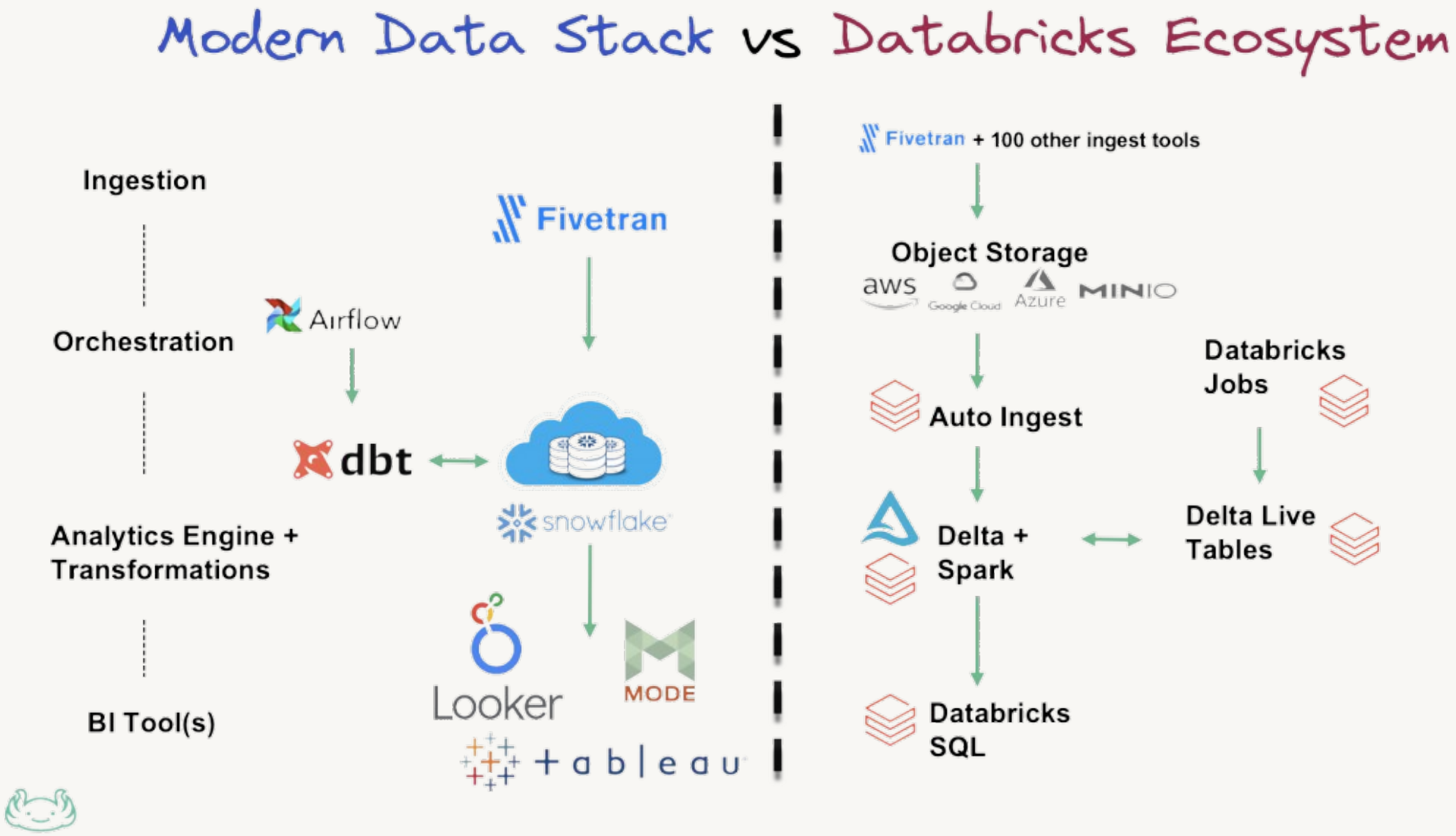
Warehouses	Lakes
Tables	Files

Data Lakehouse

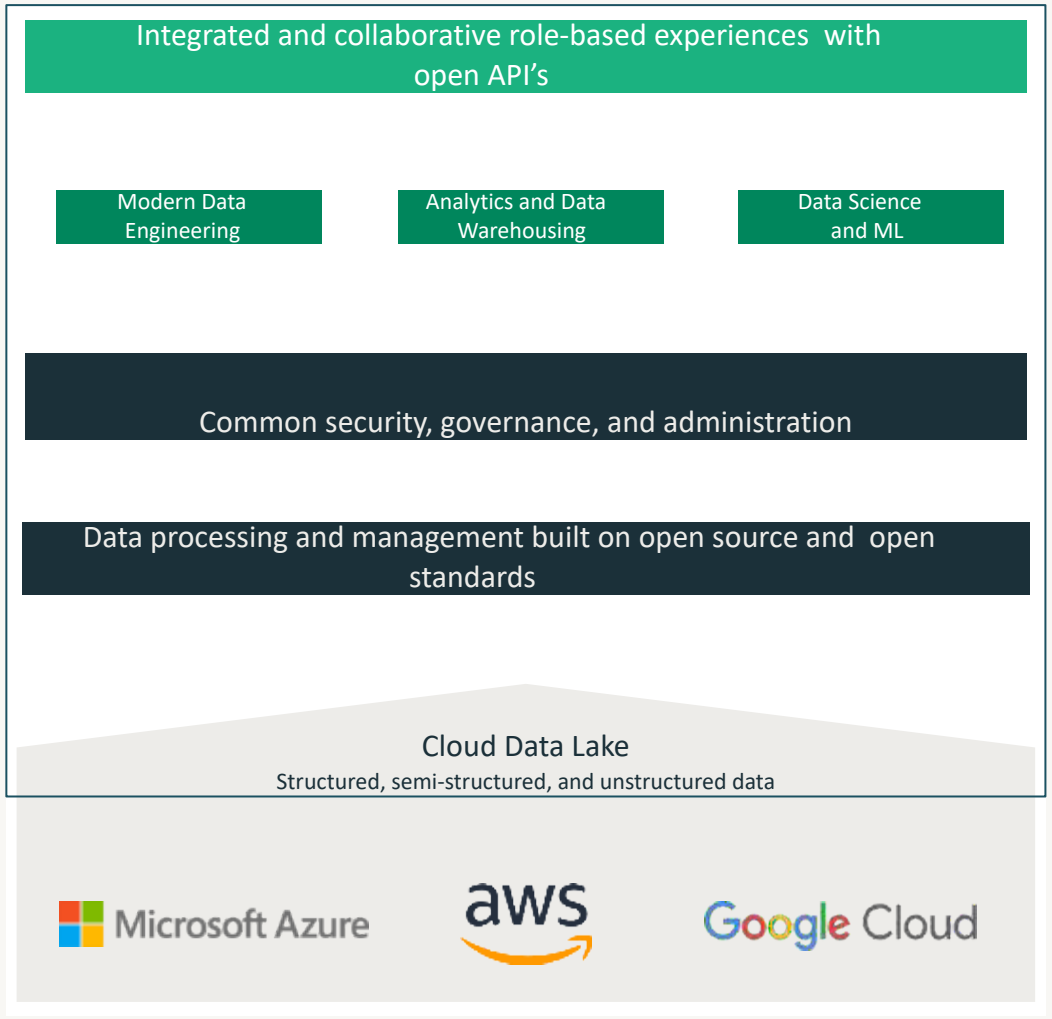
One platform to unify all of
your data, analytics, and AI workloads



Why choose Databricks ?



The data lakehouse offers a better path



Lake-first approach that builds upon where the freshest, most complete data resides

AI/ML from the ground up
High reliability and **performance** **Single approach** to managing data

Support for all use cases on a single platform:

- Data engineering
- Data warehousing
- Real time streaming
- Data science and ML

Built on **open source** and open standards

Multi-cloud, work with your cloud of choice

The Data Lakehouse Foundation

Data Lake

DELTA LAKE

An open approach to bringing data management and governance to data lakes

Better reliability with transactions

48x faster data processing with indexing

Data governance at scale with fine-grained access control lists

Data Warehouse

What is Delta Lake?

- A open source project that enables building a Lakehouse architecture on top of data lakes.
- An storage layer that brings scalable, ACID transactions to Apache Spark and other big-data engines.
- Delta Lake provides ACID transactions, scalable metadata handling, and unifies streaming and batch data processing on top of existing data lakes, such as S3, ADLS, GCS, and HDFS.

Key Features:

- ACID Transactions
- Scalable Metadata Handling
- Time Travel (data versioning)
- Open Format
- Delta Lake change data feed
- Unified Batch and Streaming Source and Sink
- Schema Enforcement
- Schema Evolution
- Audit History
- Updates and Delete
- 100% Compatible with Apache Spark API
- Data Clean-up

https://databricks.com/blog/2019/08/21/diving-into-delta-lake-unpacking-the-transaction-log.html

Delta Lake solves challenges with data lakes

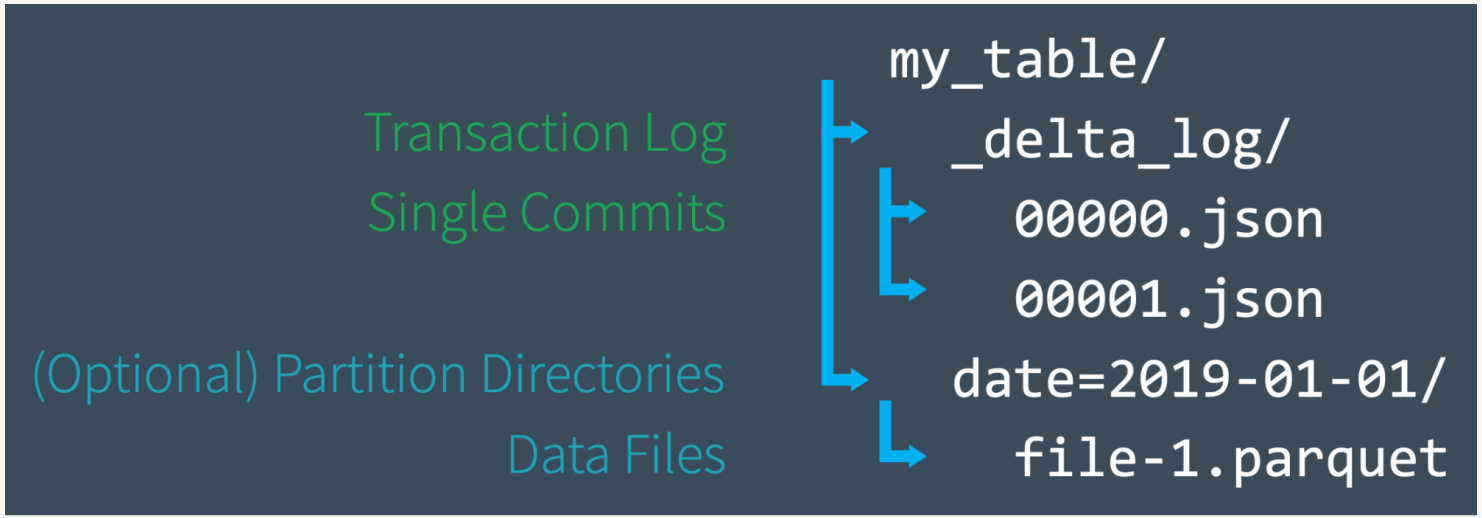
RELIABILITY & QUALITY → ACID transactions

PERFORMANCE & LATENCY → Advanced indexing & caching

GOVERNANCE → Governance with Data Catalogs

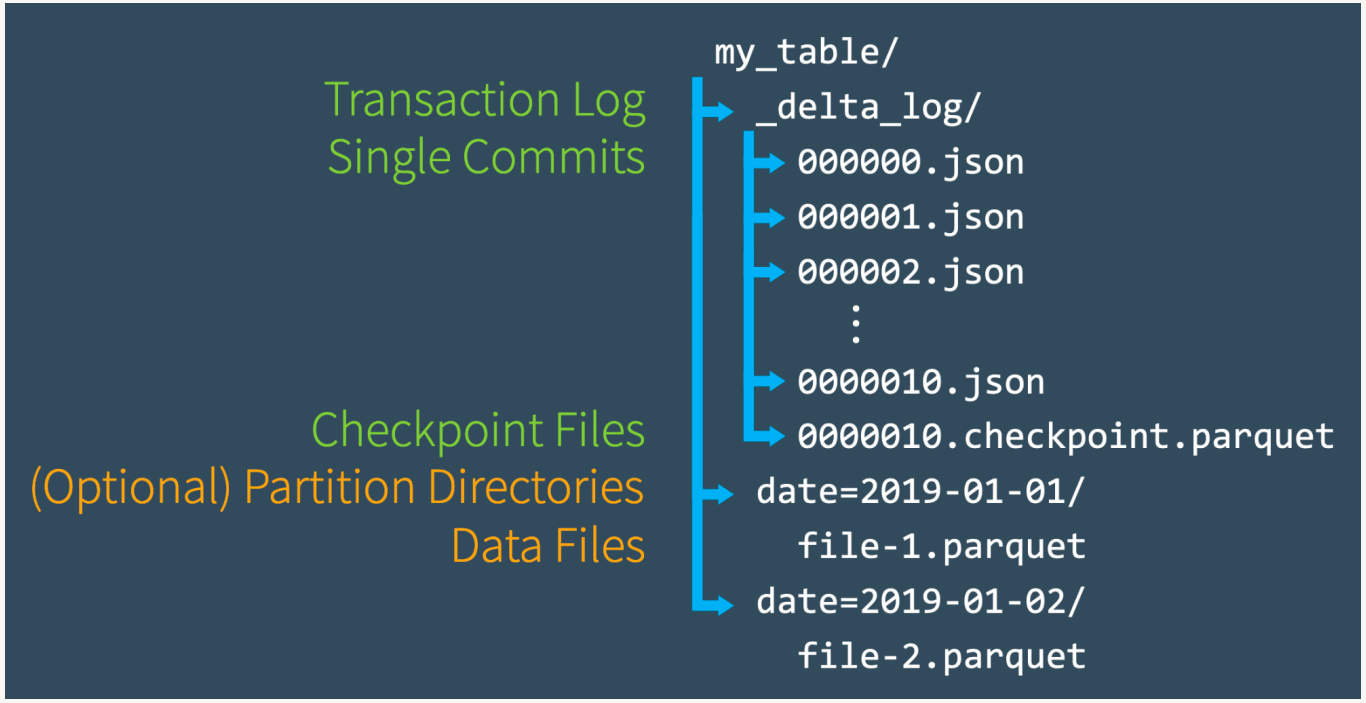
Delta Lake key feature - ACID transaction

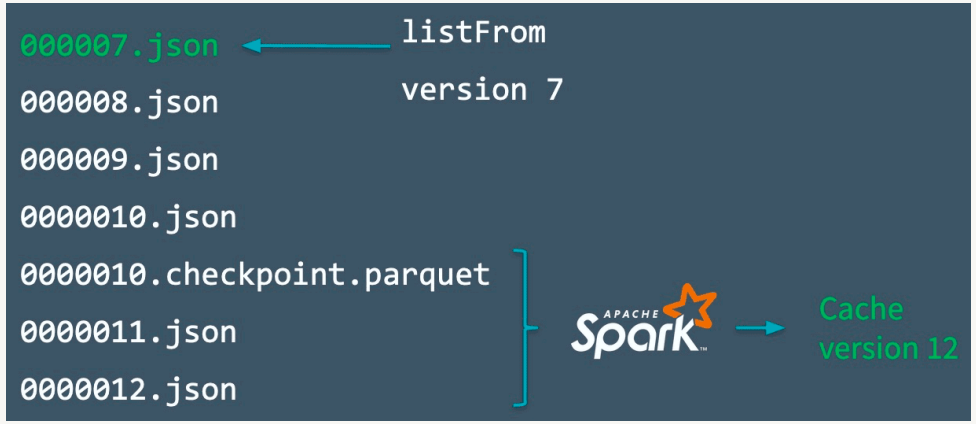
- Add File: It adds the data file
- Remove File: It removes the data file
- Update Metadata: It updates the table metadata.
- Set Transaction: It records that a structure streaming job created a micro-batch with ID
- Change Protocol: Makes more secure by transferring Delta Lakes to the latest securing protocol.
- Commit Info: It contains the information about the Commits.

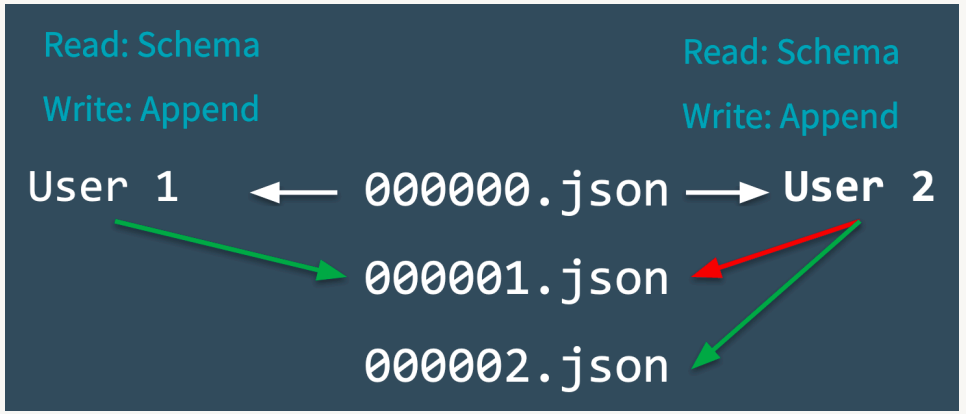


State Recomputing With Checkpoint Files

- Delta Lake automatically generates checkpoint files every 10 commits
- Delta Lake saves a checkpoint file in Parquet format in the same _delta_log subdirectory.





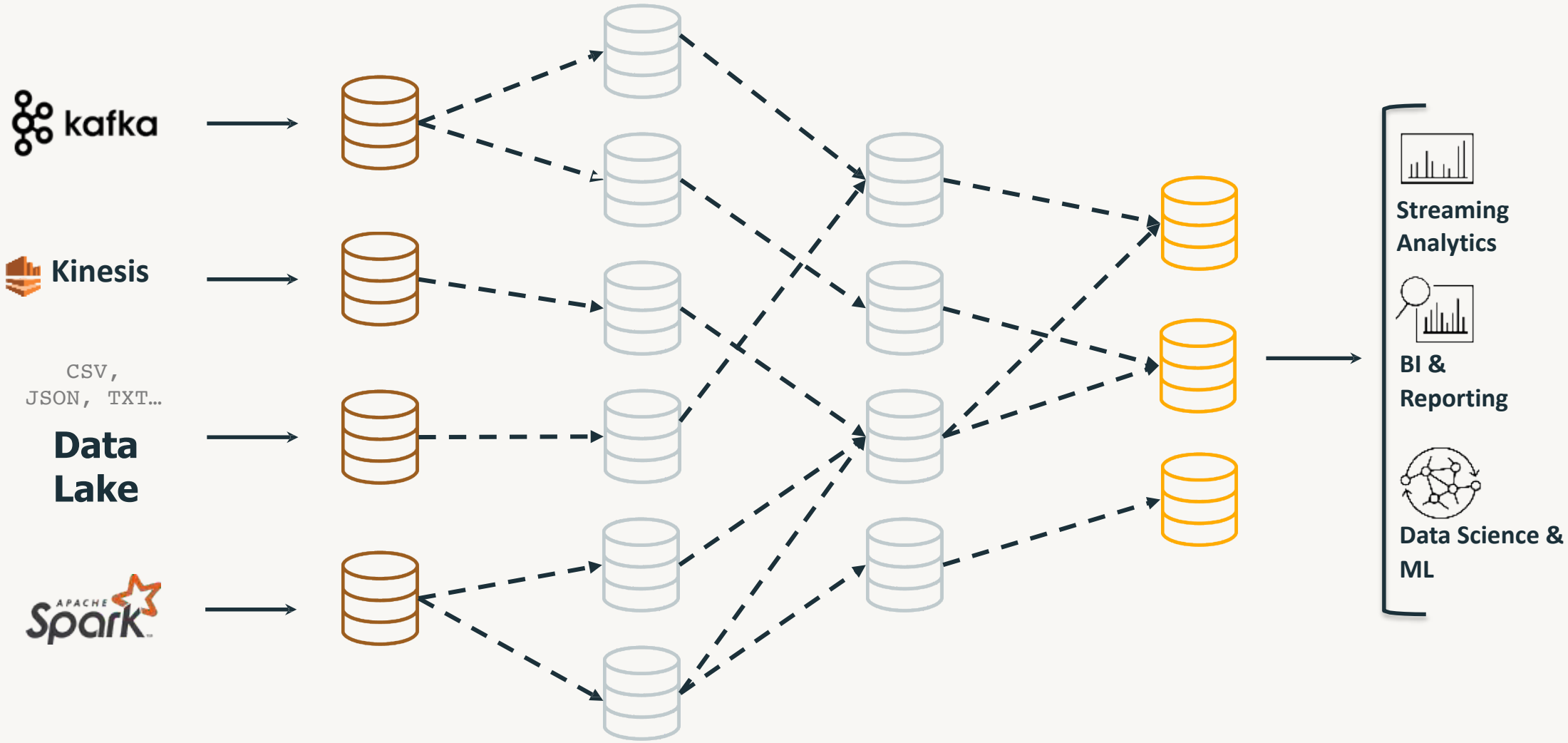


Building the foundation of a Lakehouse

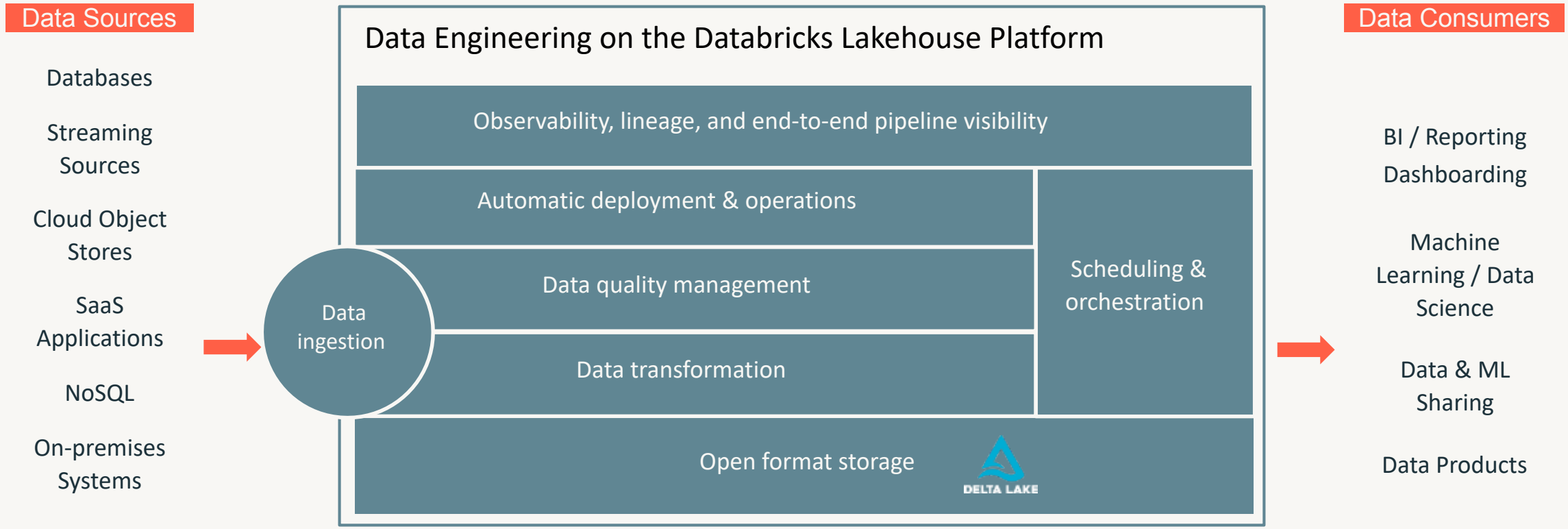
Greatly improve the quality of your data for end users

But the reality is not so simple

Maintaining data quality and reliability at scale is complex and brittle



Modern data engineering on the lakehouse

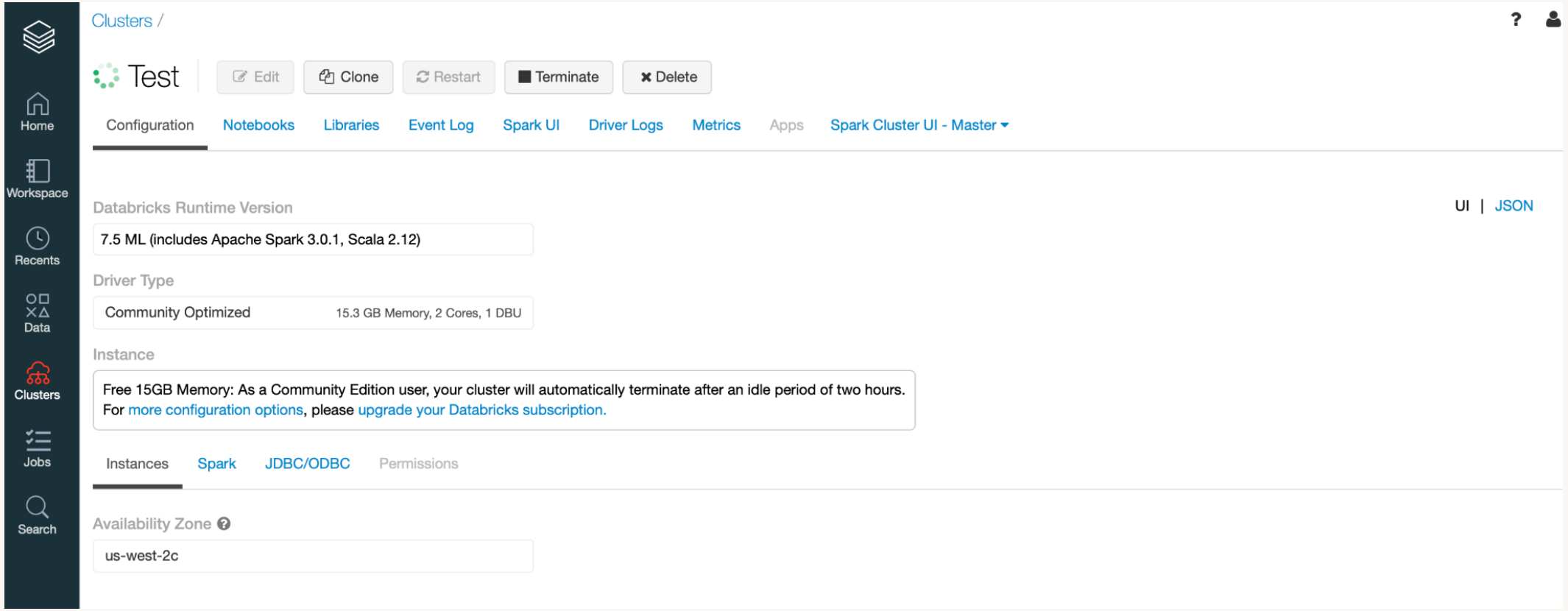


Data Science & Engineering Workspace

Databricks Workspaces: Clusters

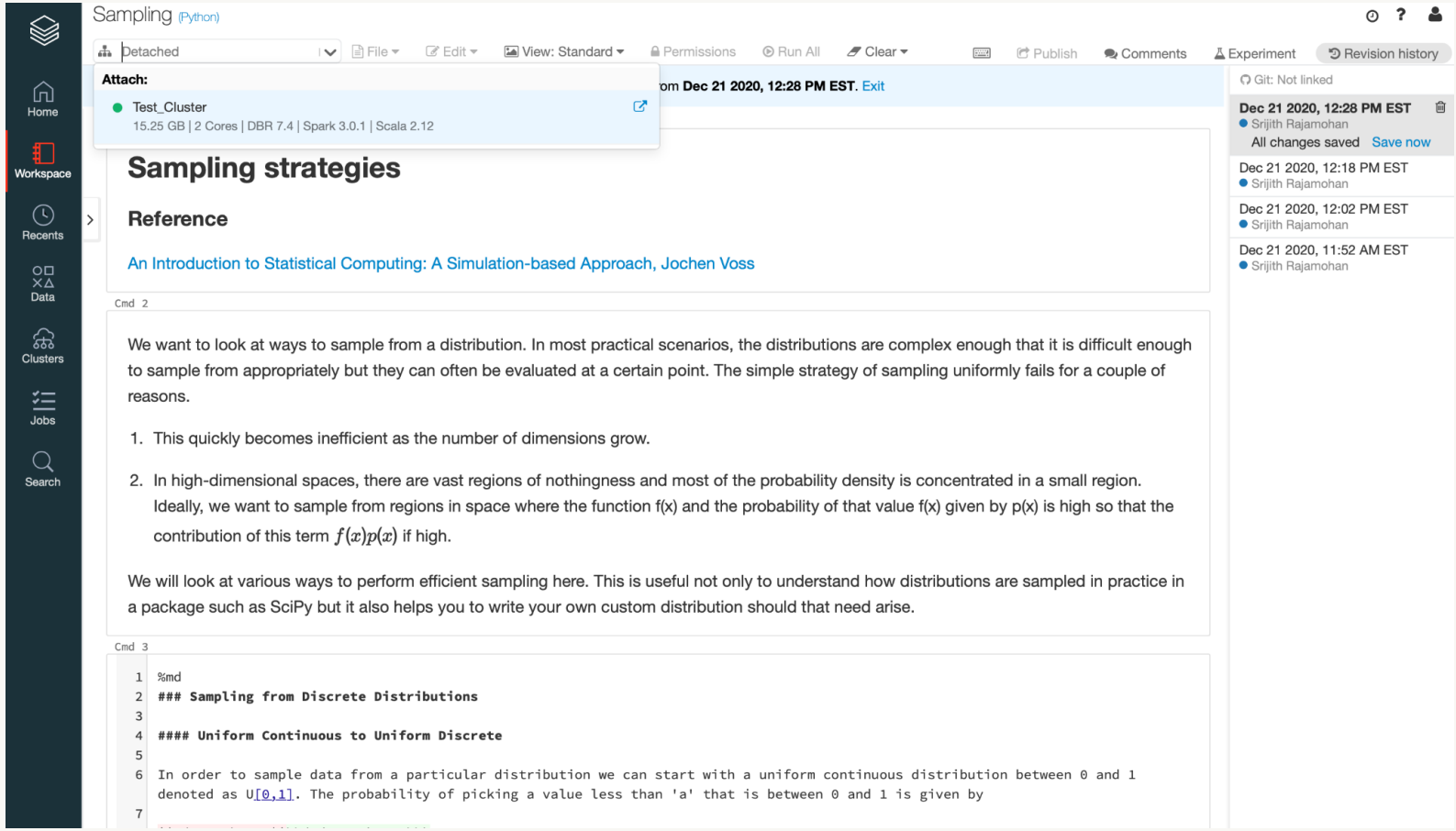
It is a set of computation resources where a developer can run Data Analytics, Data Science, or Data Engineering workloads.

The workloads can be executed in the form of a set of commands written in a notebook



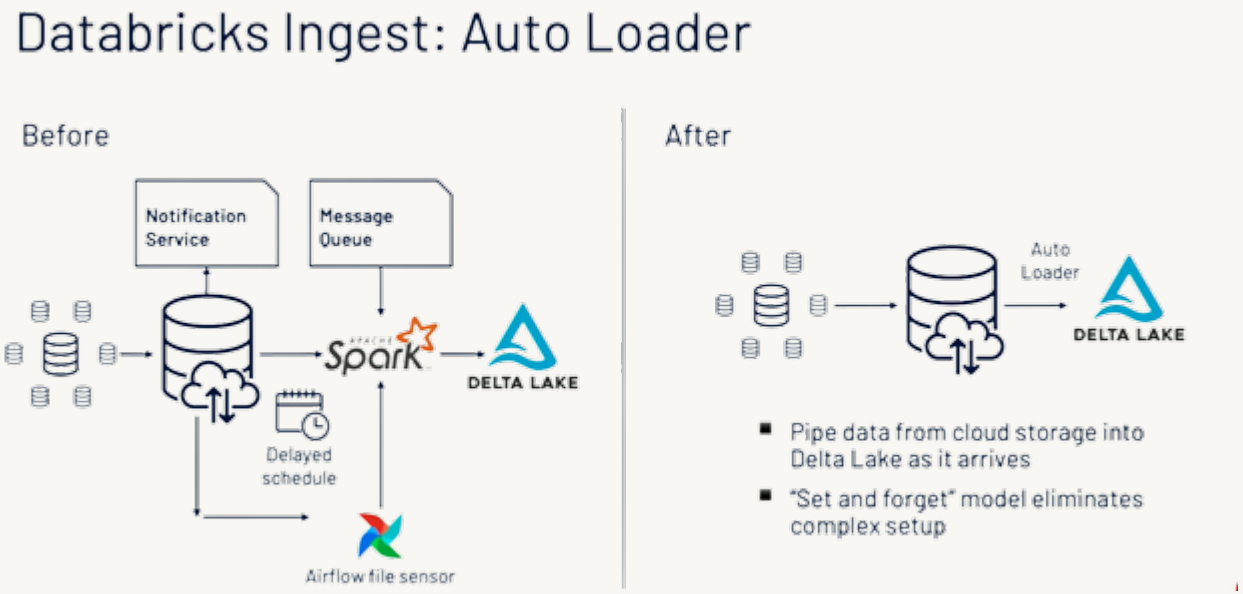
Databricks Workspaces: Notebooks

It is a Web Interface where a developer can write and execute codes. Notebook contains a sequence of runnable cells that helps a developer to work with files, manipulate tables, create visualizations, and add narrative texts



Databricks Workspaces: AutoLoader

Auto Loader incrementally and efficiently processes new data files as they arrive in cloud storage. Auto Loader can load data files from Google Cloud Storage (GCS, gs://) in addition to Databricks File System (DBFS, dbfs:/)



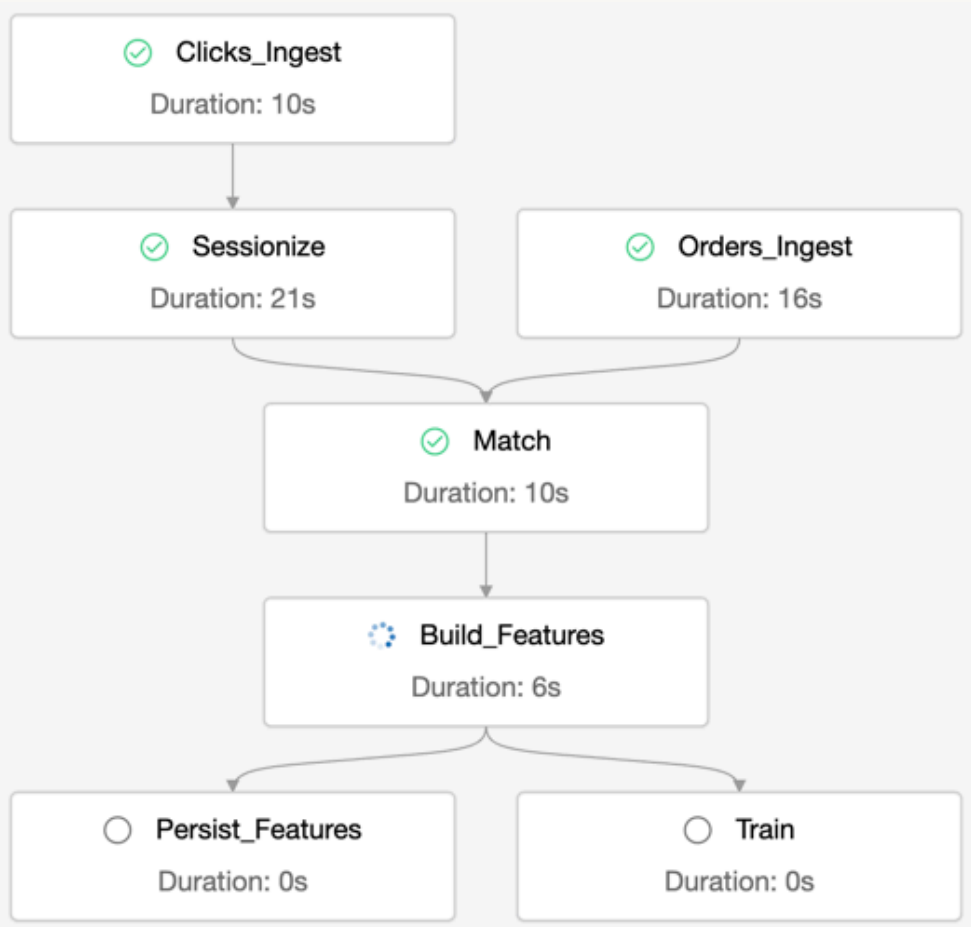

```scala
val checkpoint_path = "/tmp/delta/population_data/_checkpoints"
val write_path = "/tmp/delta/population_data"

// Set up the stream to begin reading incoming files from the
// upload_path location.
val df = spark.readStream.format("cloudFiles")
  .option("cloudFiles.format", "csv")
  .option("header", "true")
  .schema("city string, year int, population long")
  .load(upload_path)

// Start the stream.
// Use the checkpoint_path location to keep a record of all files that
// have already been uploaded to the upload_path location.
// For those that have been uploaded since the last check,
// write the newly-uploaded files' data to the write_path location.
df.writeStream.format("delta")
  .option("checkpointLocation", checkpoint_path)
   .start(write_path)
```

** Supports: JSON, CSV, PARQUET, AVRO, ORC, TEXT, and BINARYFILE file formats. **

https://docs.databricks.com/spark/latest/structured-streaming/auto-loader.html

Databricks Workspaces: Jobs

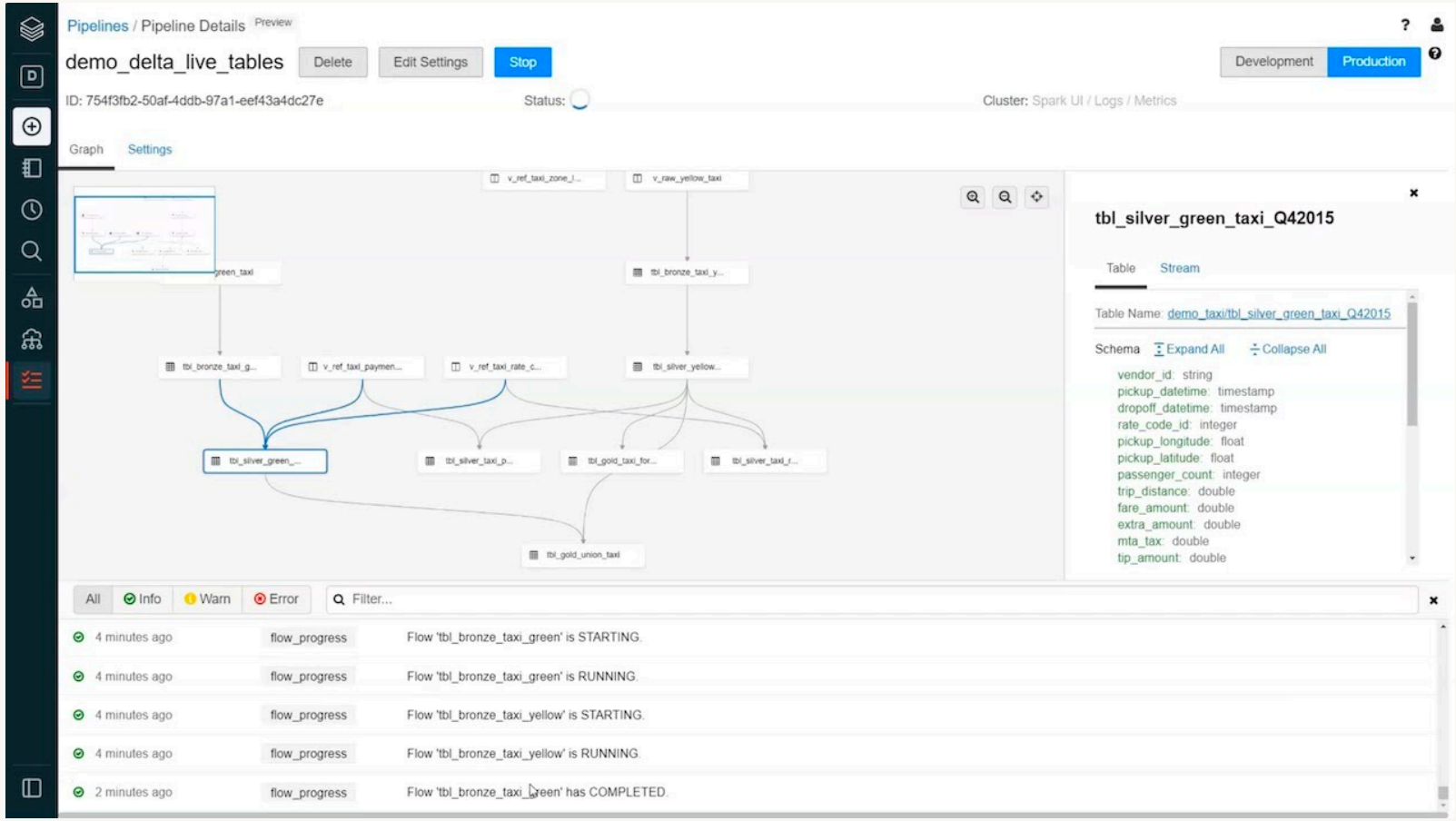
Jobs allow a user to run notebooks on a scheduled basis. It is a method of executing or automating specific tasks like ETL, Model Building, and more.



The pipeline of the ML workflow can be organized into jobs so that it sequentially runs the series of steps one after another

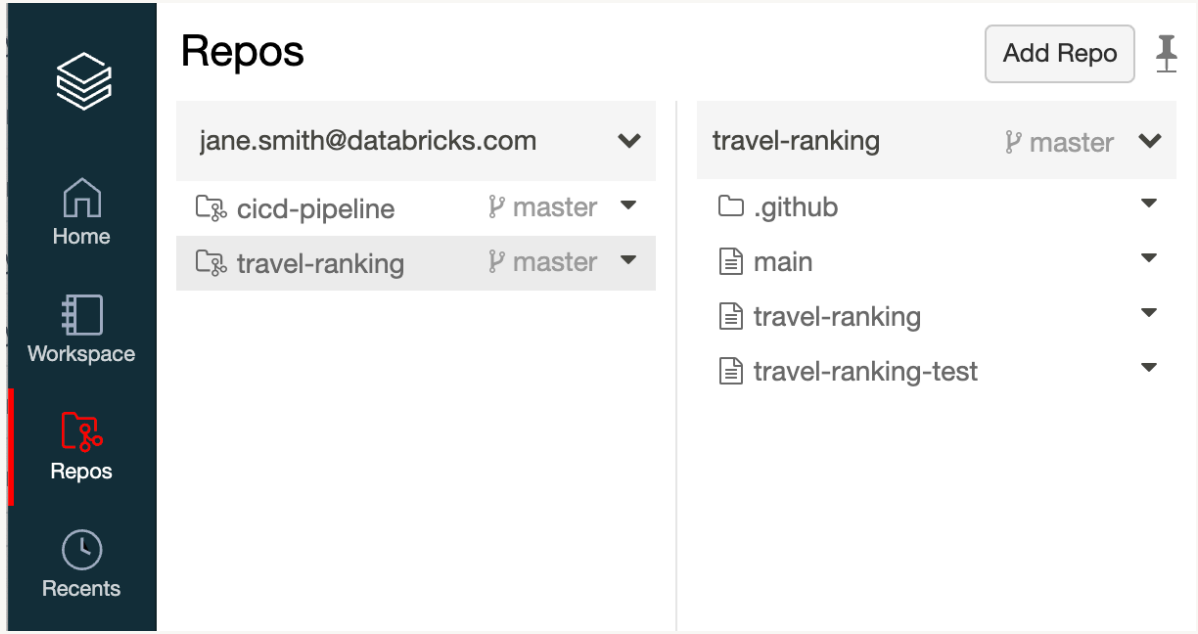
Databricks Workspaces:Delta Live Tables

Delta Live Tables is a framework designed to enable declaratively define, deploy, test & upgrade data pipelines and eliminate operational burdens associated with the management of such pipelines.



Databricks Workspaces: Repos

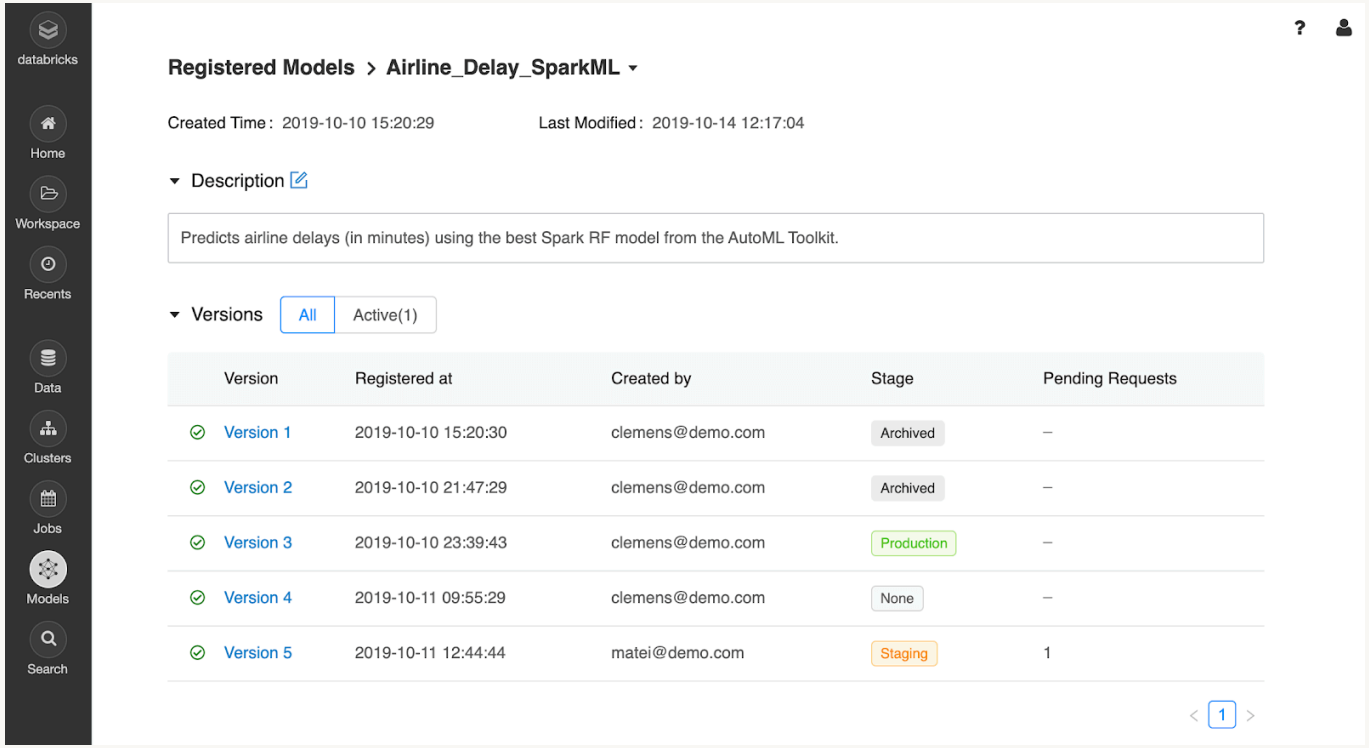
To empower the process of ML application development, repo's provide repository-level integration with Git-based hosting providers such as GitHub, GitLab, bitBucket, and Azure DevOps



Developers can write code in a Notebook and Sync it with the hosting provider, allowing developers to clone, manage branches, push changes, pull changes, etc.

Databricks Workspaces: Models

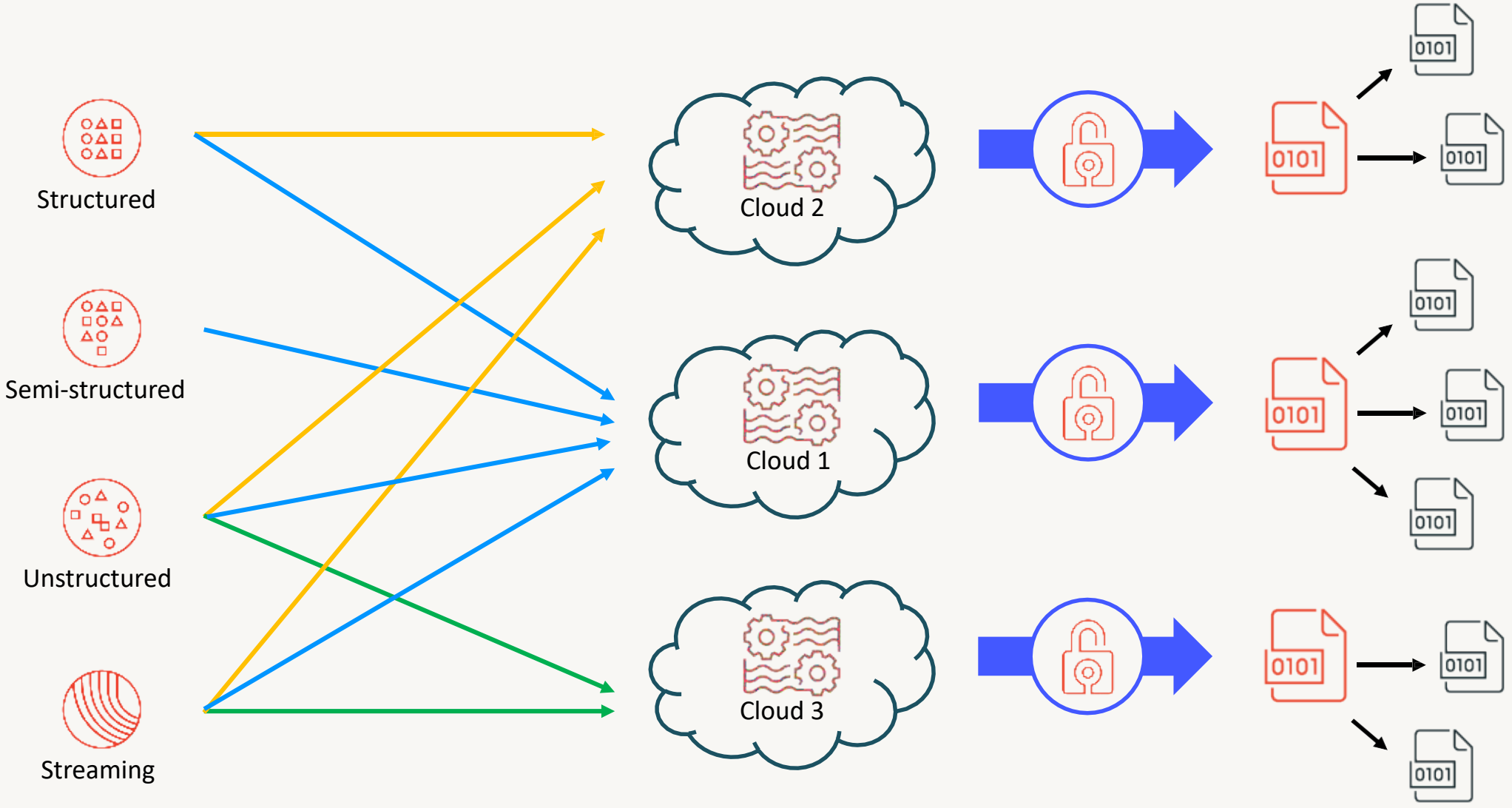
It refers to a Developer's ML Workflow Model registered in the MLflow Model Registry, a centralized model store that manages the entire life cycle of MLflow models.



MLflow Model Registry provides all the information about modern lineage, model versioning, present condition, workflow, and stage transition (whether promoted to production or archived).

Governance requirements for data are quickly evolving

Governance is hard to enforce on data lakes

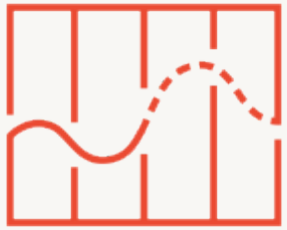


The problem is getting bigger

Enterprises need a way to share and govern a wide variety of data products

Files

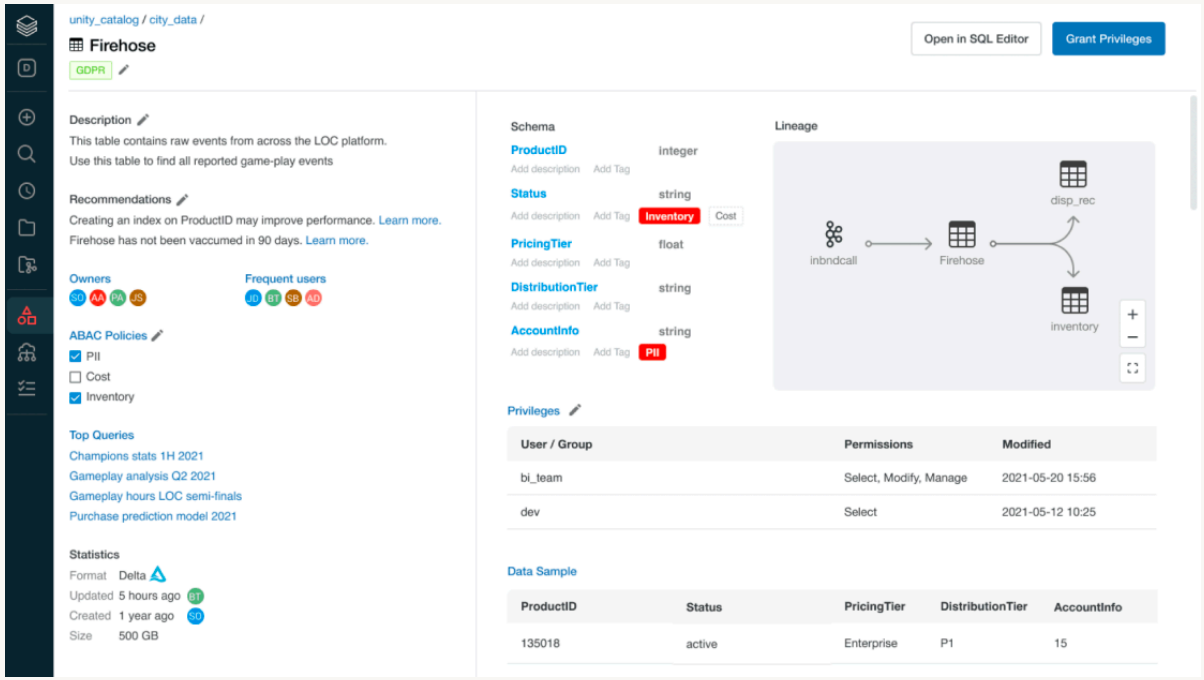
Dashboards



Models

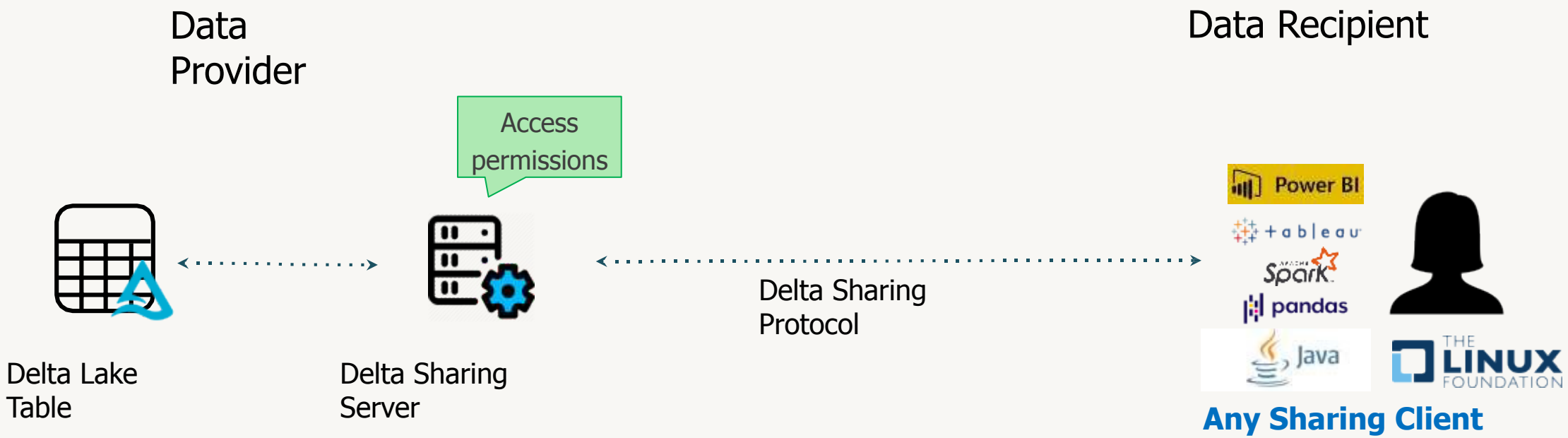
Tables

Unity Catalog for Lakehouse Governance



- **Centrally catalog, Search, and discover** data and AI assets
- Simplify governance with a **unified Cross- cloud governance model**
- **Easily integrate** with your existing Enterprise Data Catalogs
- **Securely share live data across platforms** with delta sharing

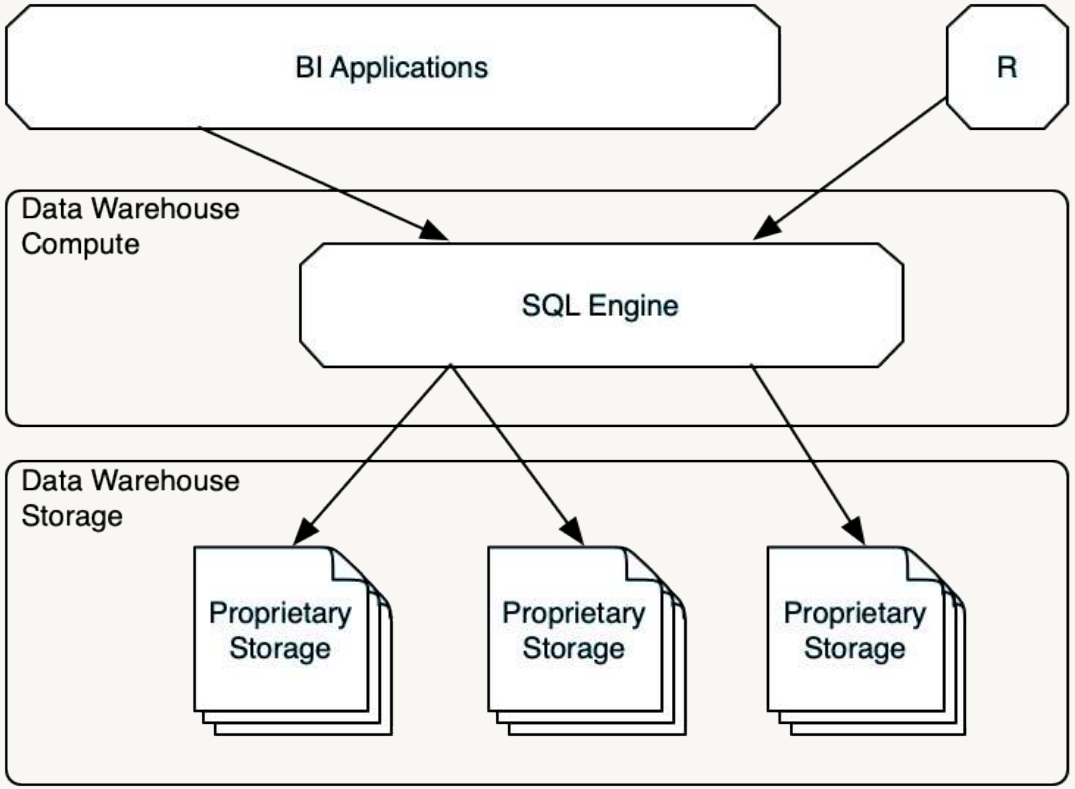
Delta Sharing on Databricks



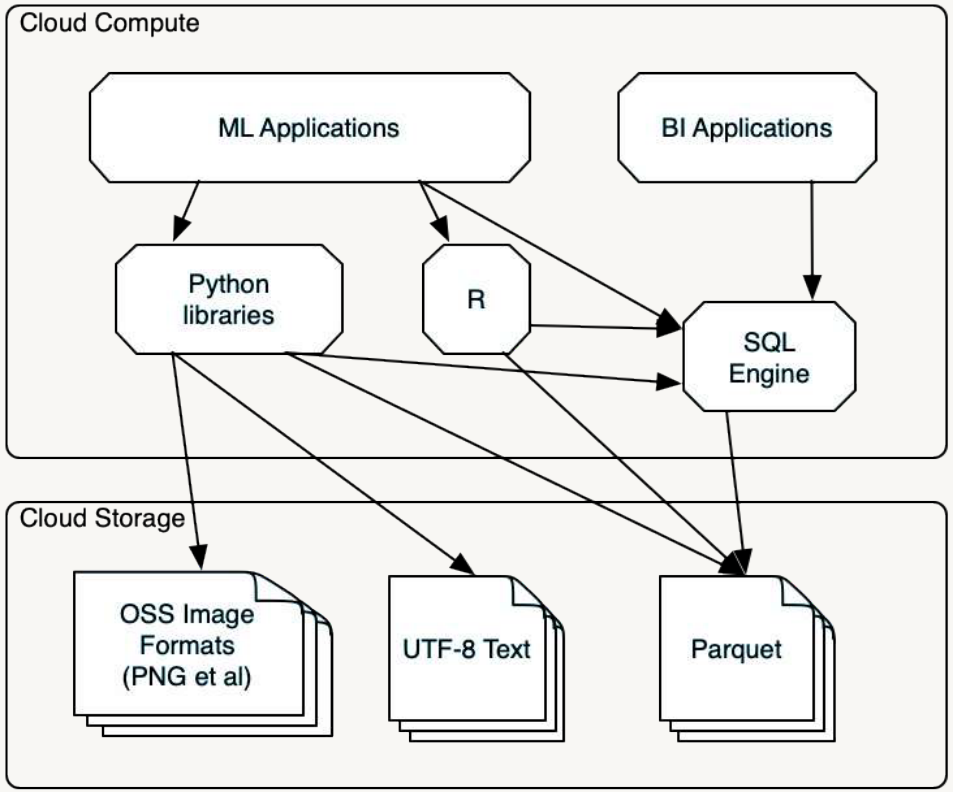
Machine Learning Workspace

ML Architecture: Data Warehouse VS Data Lakehouse

Data Warehouse

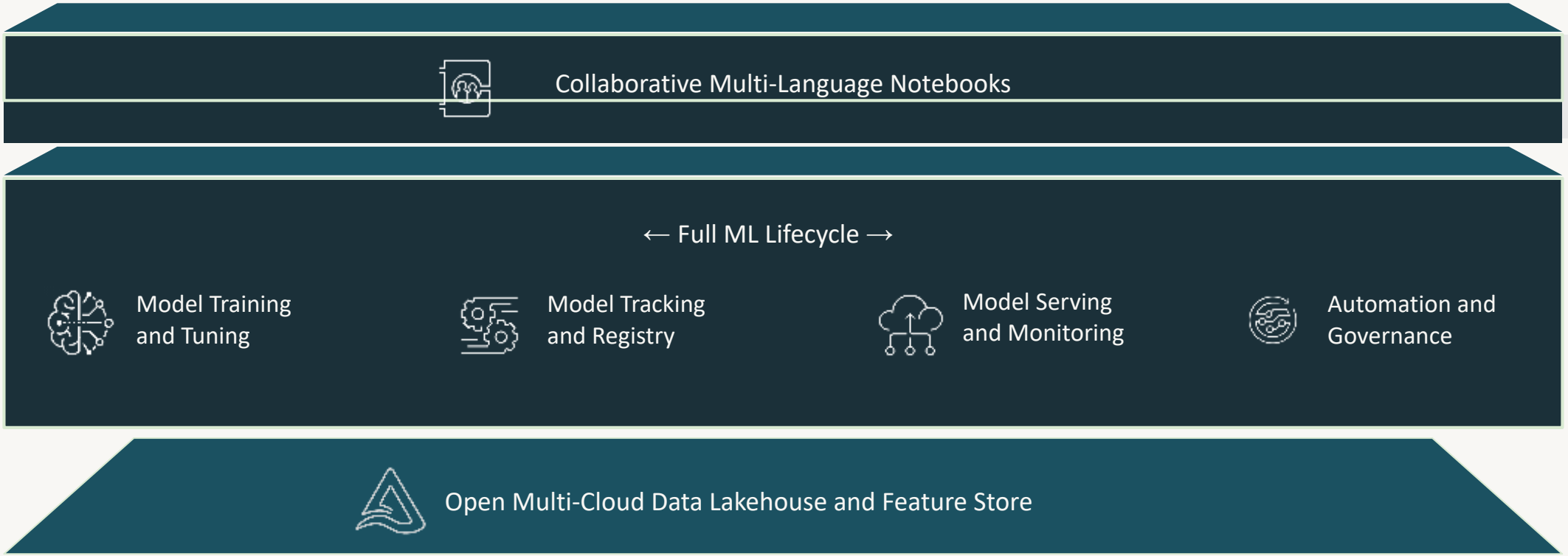


Data Lakehouse



Data Science and Machine Learning

A data-native and collaborative solution for the full ML lifecycle



What Does ML Need from a Lakehouse?

Access to Unstructured Data

- Images, text, audio, custom formats
- Libraries understand files, not tables
- Must *scale* to petabytes

Open Source Libraries

- OSS dominates ML tooling (Tensorflow, scikit- learn, xgboost, R, etc)
- Must be able to apply these in Python, R

Specialized Hardware, Distributed Compute

- Scalability of algorithms
- GPUs, for deep learning
- Cloud elasticity to manage that cost!

Model Lifecycle Management

- Outputs are model artifacts
- Artifact lineage
- Productionization of model

Three Data Users

Business Intelligence

- SQL and BI tools
- Prepare and run reports
- Summarize data
- Visualize data
- (Sometimes) Big Data
- Data Warehouse data store

Data Science

- R, SAS, some Python
- Statistical analysis
- Explain data
- Visualize data
- Often small data sets
- Database, data warehouse data store; local files

Machine Learning

- Python
- Deep learning and specialized GPU hardware
- Create predictive models
- Deploy models to prod
- Often big data sets
- Unstructured data in files

How Is ML Different?

- Operates on **unstructured** data like text and images
- Can require learning from **massive data** sets, not just analysis of a sample
- Uses **open source** tooling to manipulate data as “DataFrames” rather than with SQL
- Outputs are **models** rather than data or reports
- Sometimes needs **special hardware**

MLOps and the Lakehouse

- Applying open tools in-place to data in the lakehouse is a win for *training*
- Applying them for *operating* models is important too!
- "Models are data too"
- Need to apply models to *data*
- **MLFlow** for MLOps on the lakehouse
  - Track and manage model data, lineage, inputs
  - Deploy models as lakehouse "services"

Feature Stores for Model Inputs

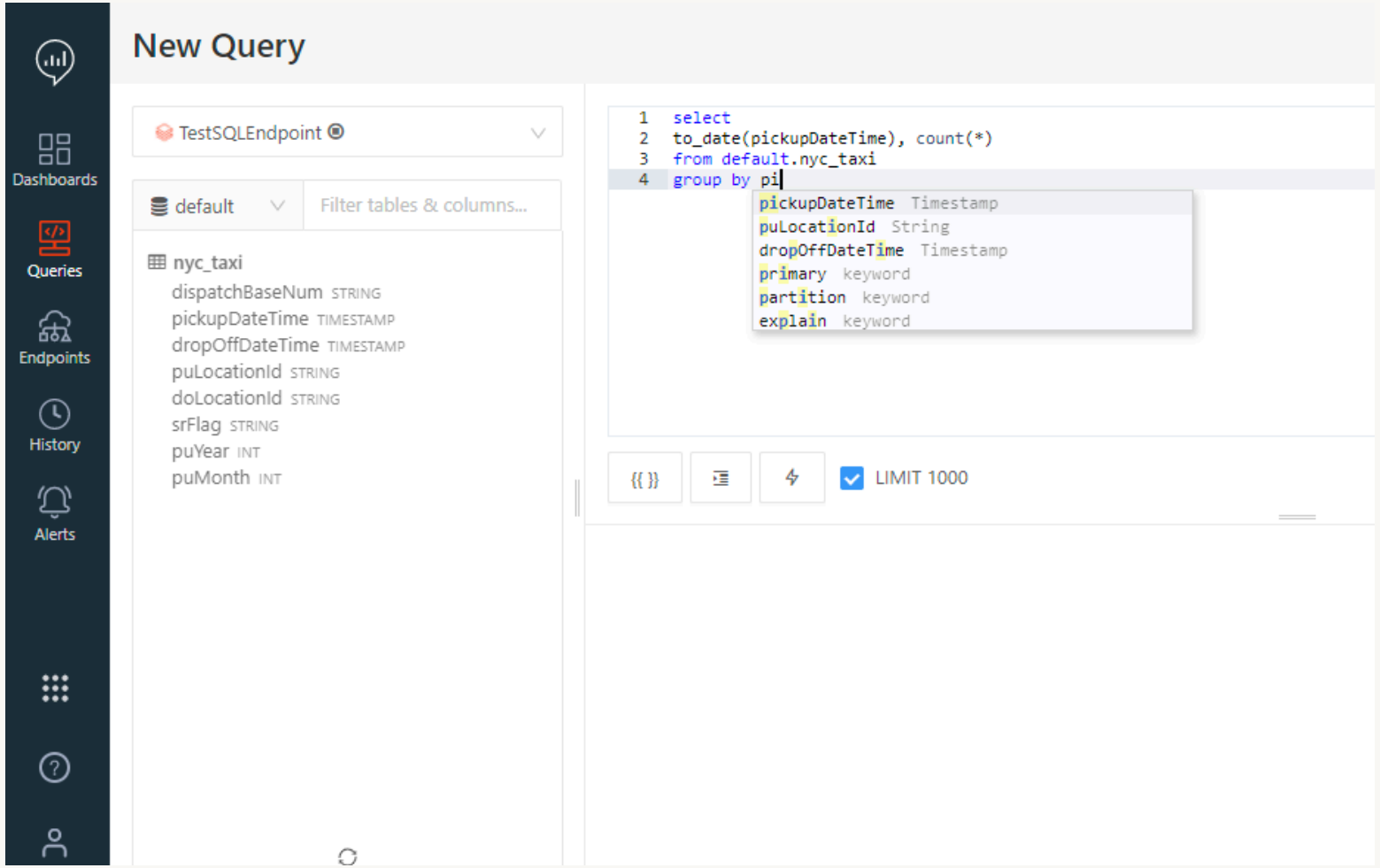
- Tables are OK for managing model input
  - Input often structured
  - Well understood, easy to access
- … but not quite enough
  - Upstream lineage: how were features computed?
  - Downstream lineage: where is the feature used?
  - Model caller has to read, feed inputs
  - How to do (also) access in real time?

SQL Analytics Workspace

Query data lake data using familiar **ANSI SQL**, and find and share new insights faster with the built-in SQL query editor, alerts, visualizations, and interactive dashboards.

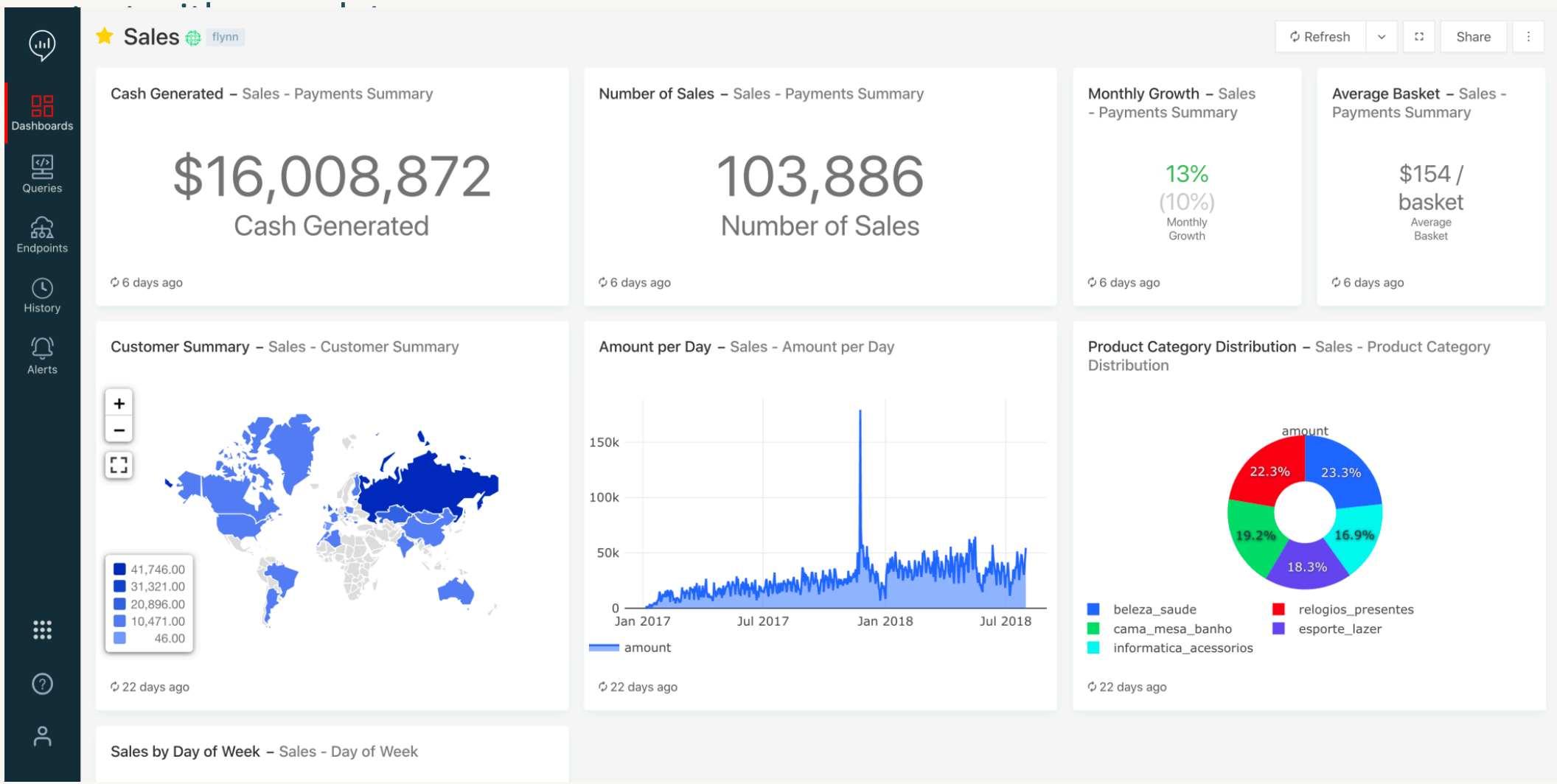
Databricks Workspaces: Queries

Provides a simplified control (which is SQL only) to query the data



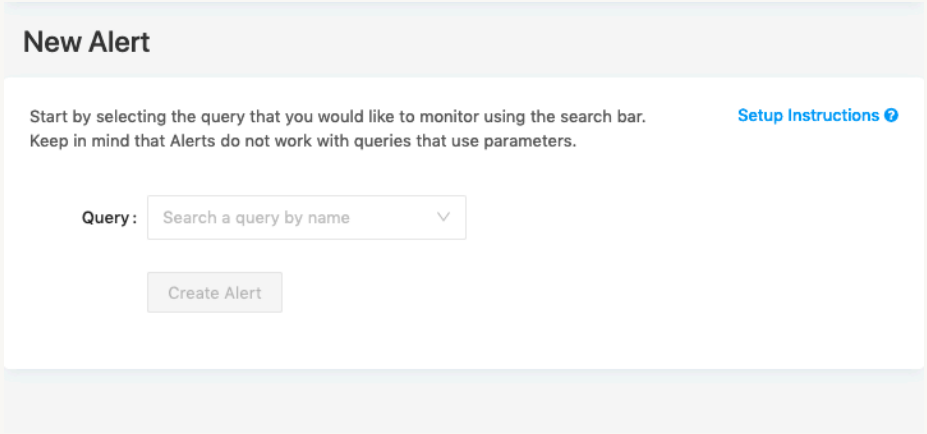
Databricks Workspaces: Dashboards

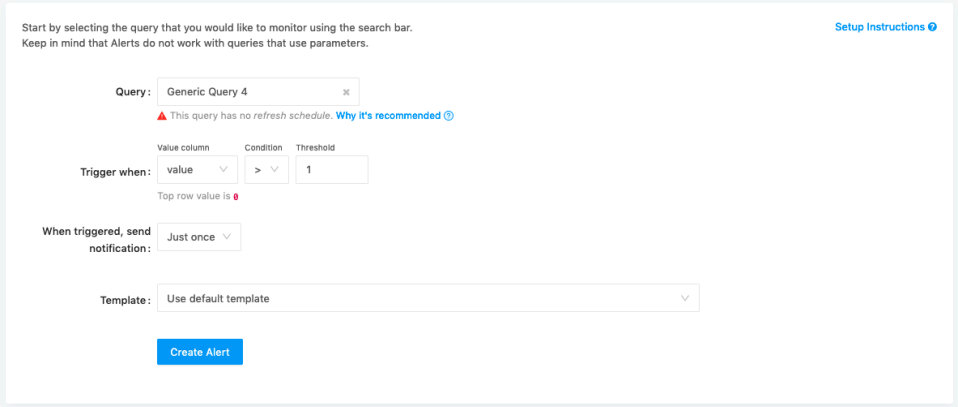
A Databricks SQL dashboard lets you combine visualizations and text boxes that provide

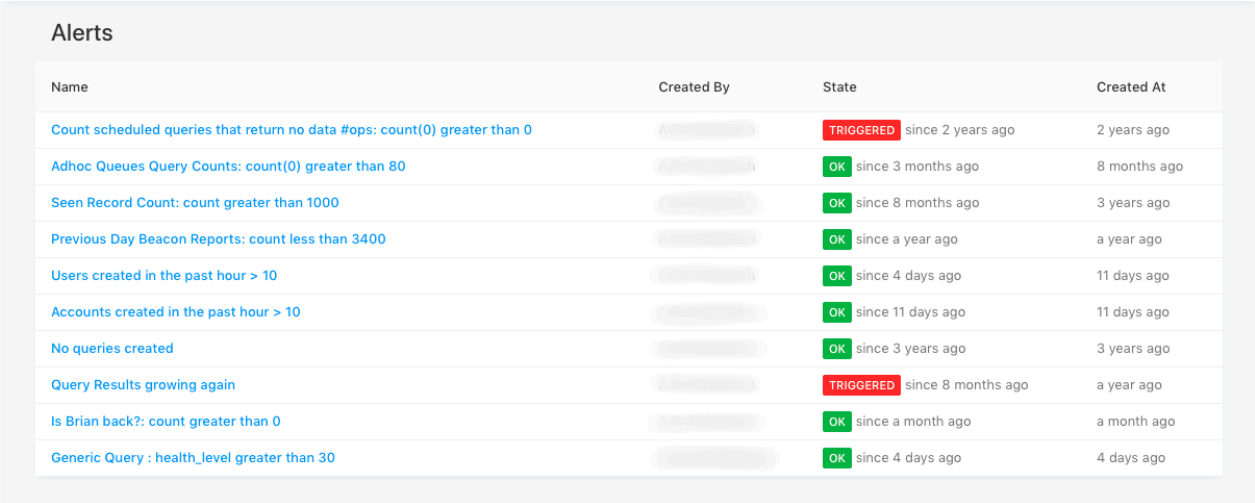


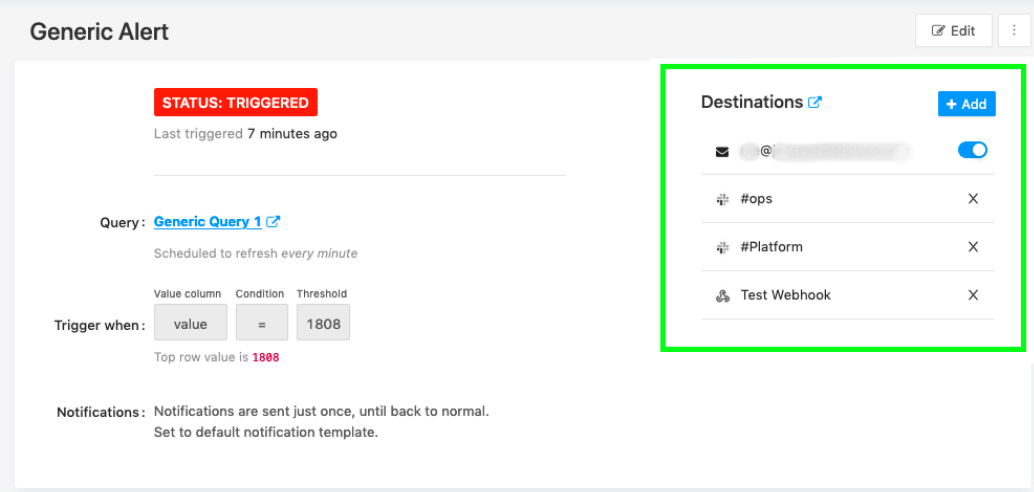
Databricks Workspaces: Alerts

Alerts notify you when a field returned by a scheduled query meets a threshold. Alerts complement scheduled queries, but their criteria are checked after every execution.



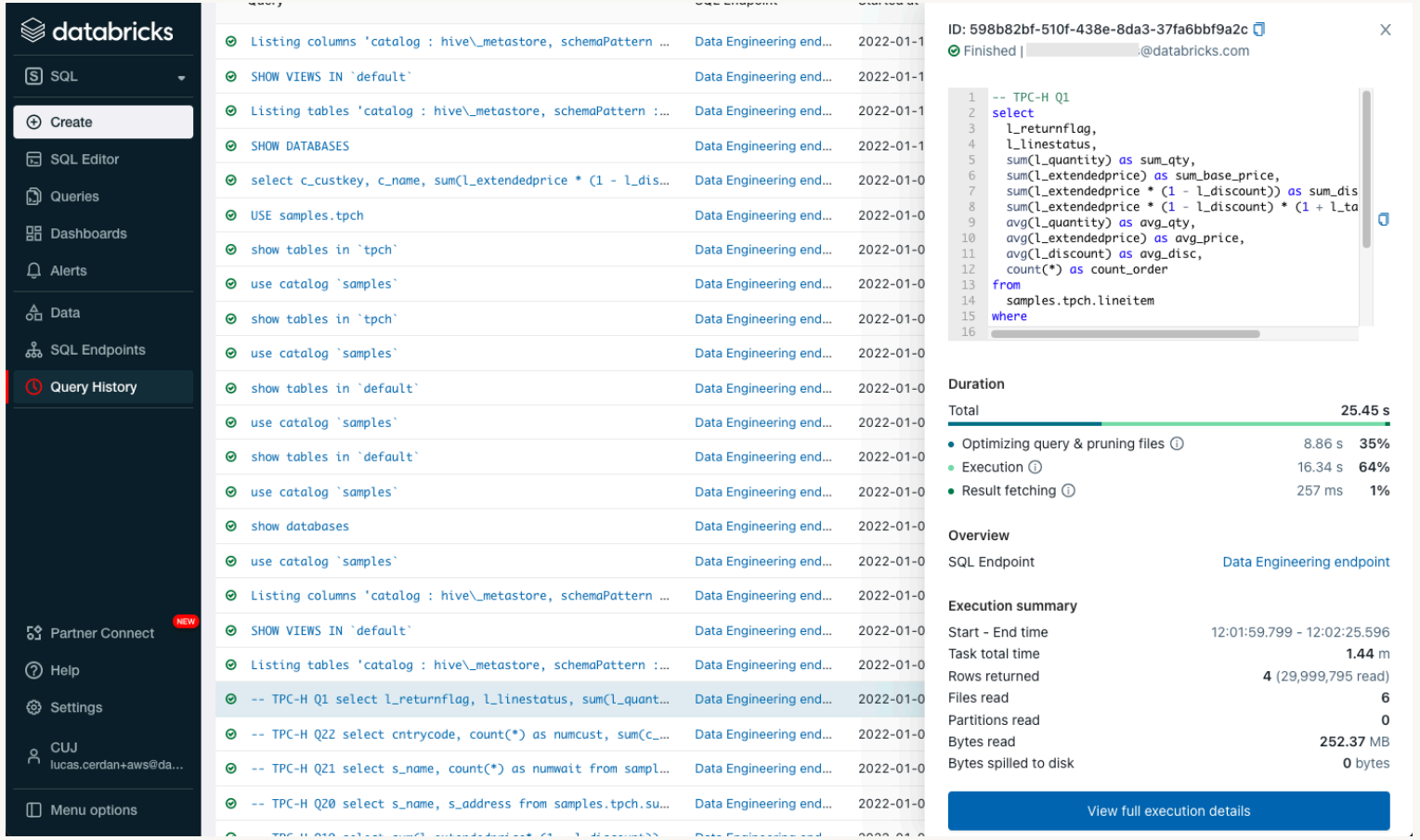






Databricks Workspaces: Query History

The query history shows SQL queries performed using SQL endpoints.



Thank you