databricks

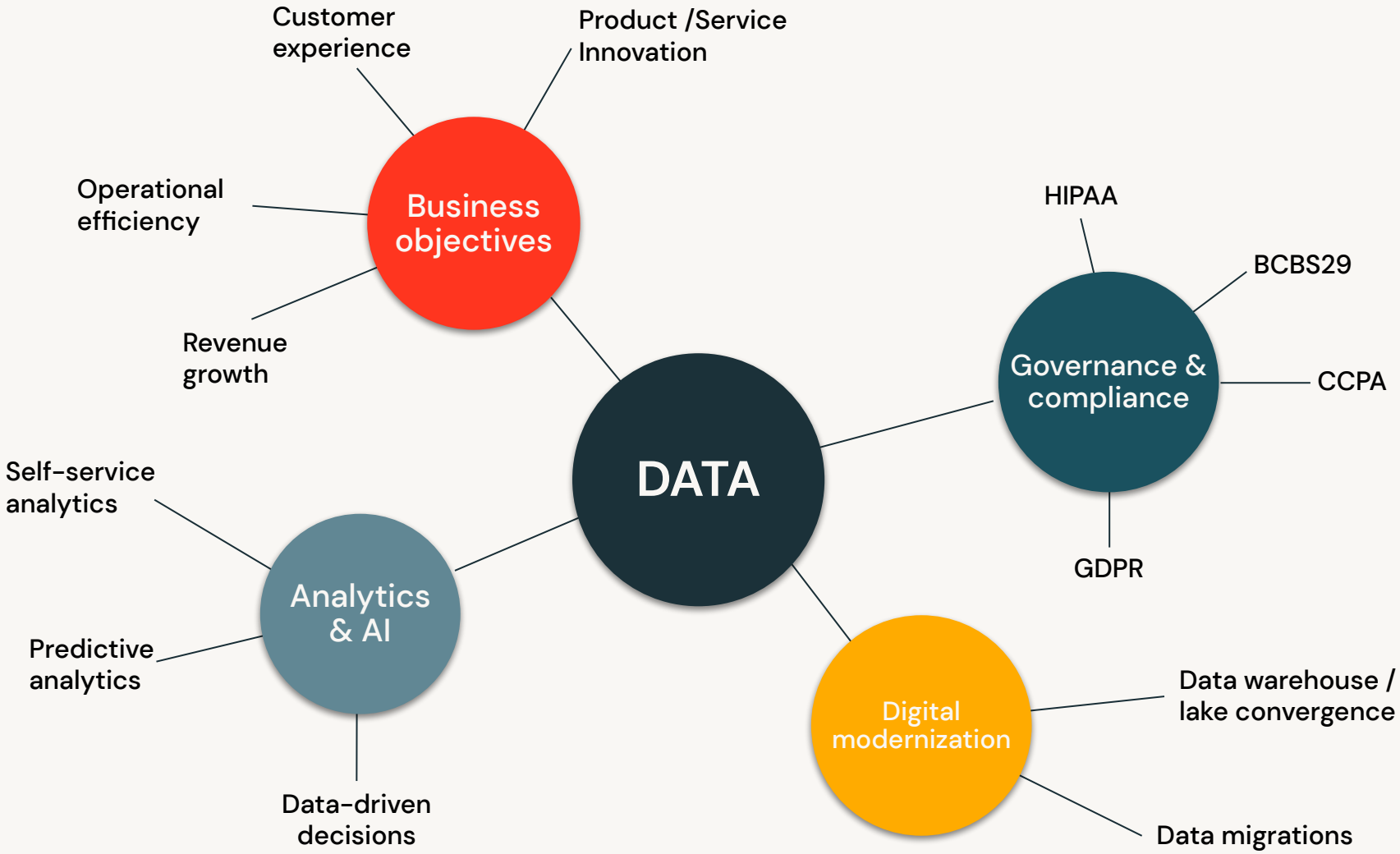
Simplify your Streaming: Delta Live Tables

Housekeeping

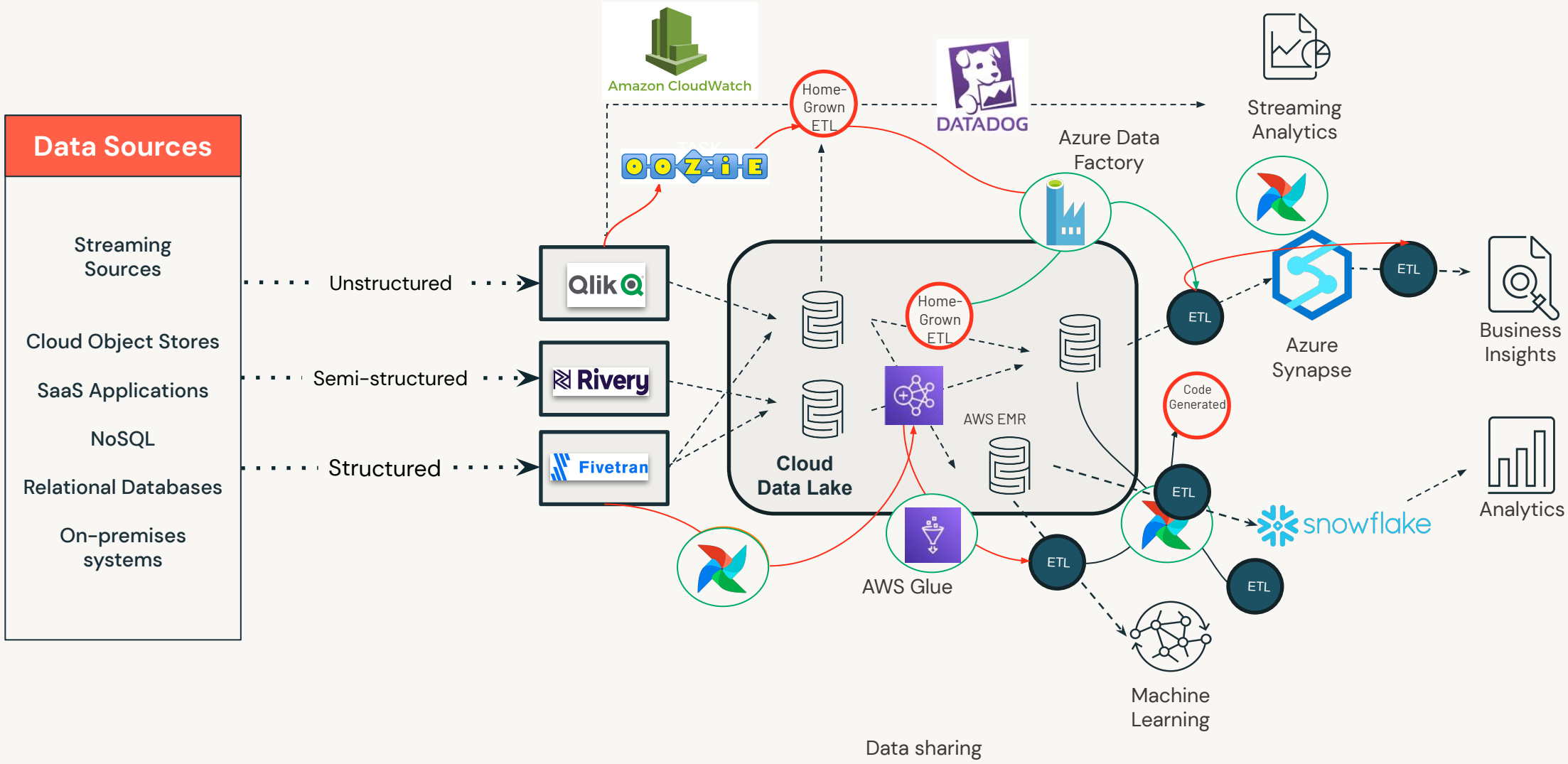
- Your connection will be muted
- We will share recording with all attendees after the session
- Submit questions in the Q&A panel
- If we do not answer you question during the event, we will follow-up with you to get you the information you need!

What's the problem with Data Engineering?

We know data is critical to business outcomes

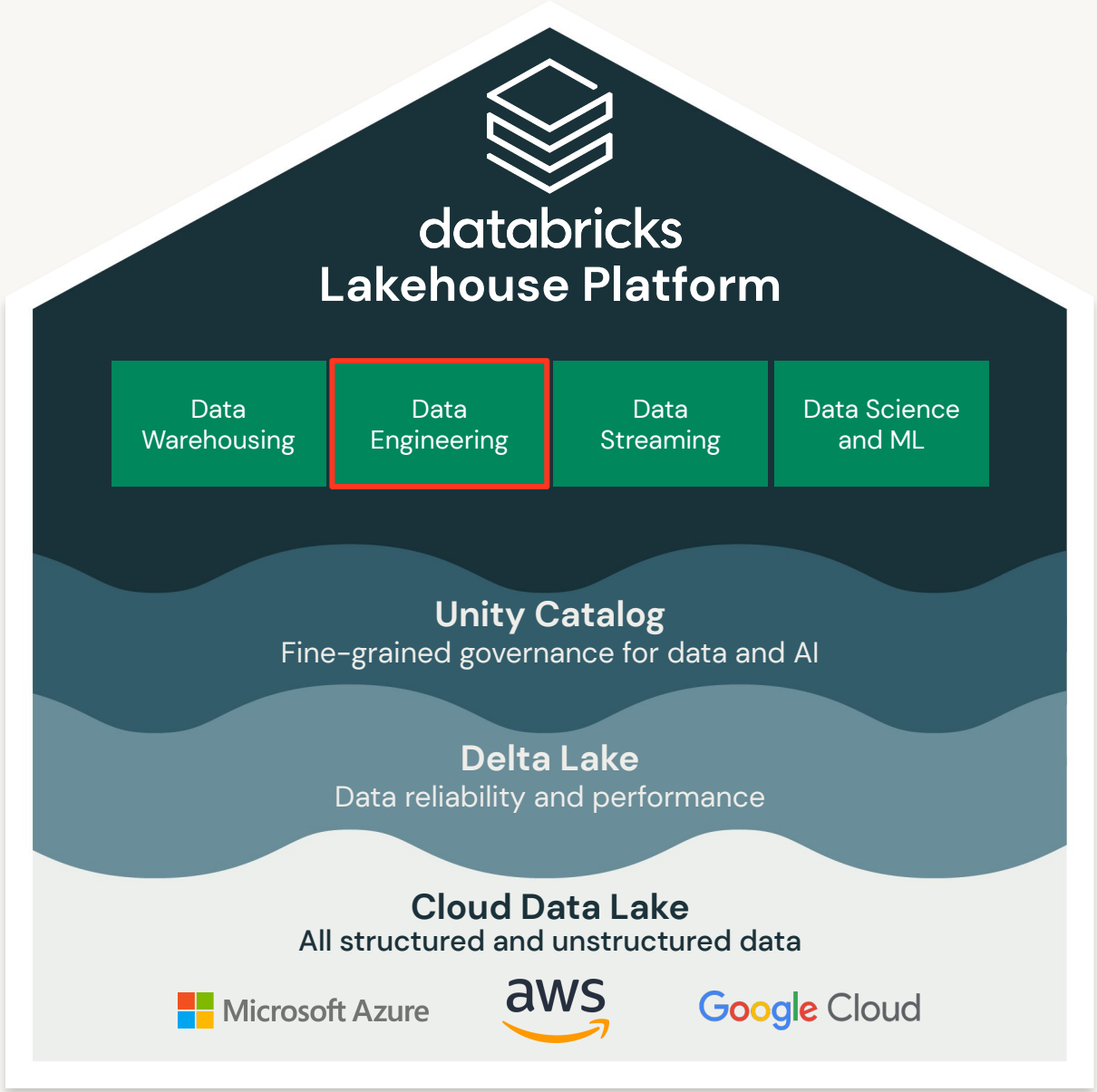


But there is complexity in the data delivery….



How does Databricks Help?

Databricks Lakehouse Platform is the foundation for **Data Engineering**



Delta Live Tables

The best way to do ETL on the lakehouse

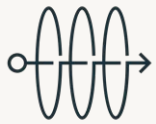
```
CREATE STREAMING TABLE raw_data
AS SELECT *
FROM cloud_files ("/raw_data", "json")
```

```
CREATE LIVE TABLE clean_data
AS SELECT ...
FROM LIVE.raw_data
```

Accelerate ETL development
Declare **SQL or Python** and DLT automatically orchestrates the DAG, handles retries, changing data

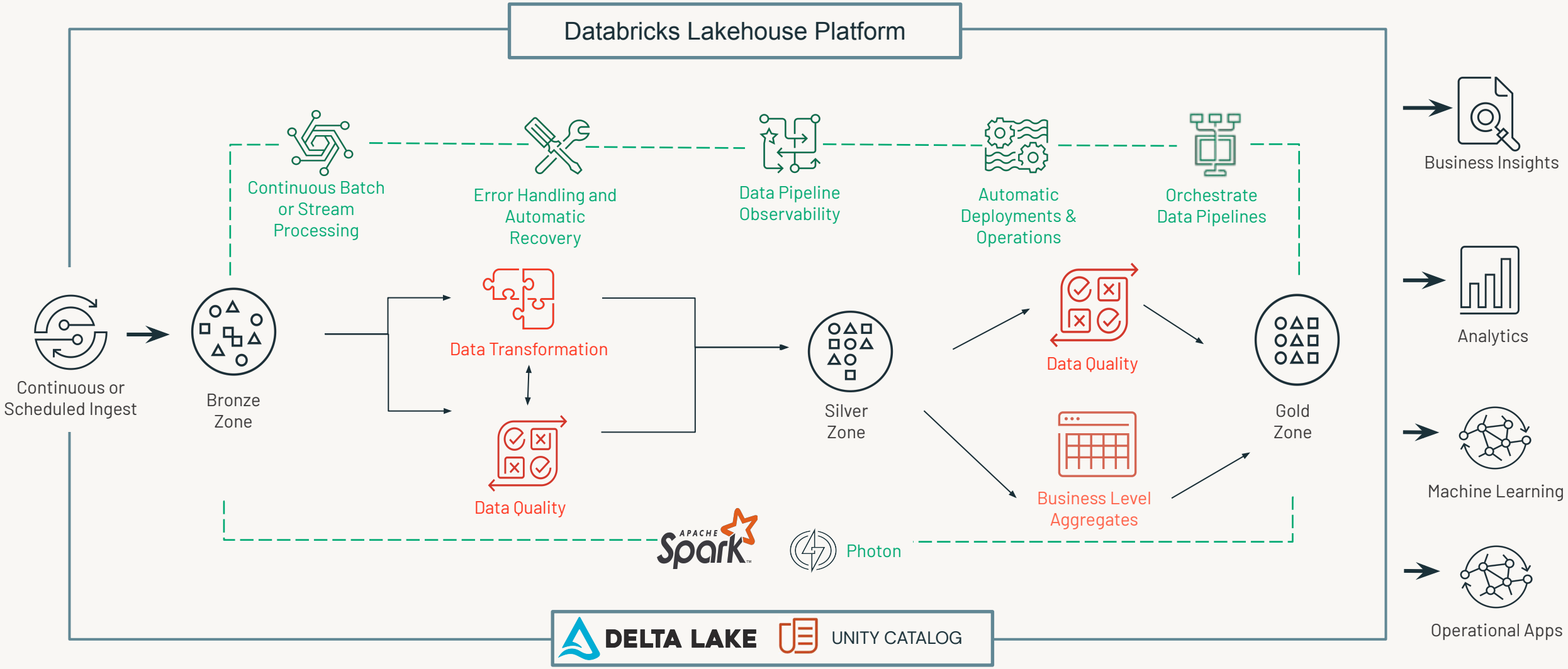
Automatically manage your infrastructure
Automates complex tedious activities like **recovery, auto-scaling, and performance optimization**

Ensure high data quality
Deliver reliable data with built-in **quality controls, testing, monitoring, and enforcement**



Unify batch and streaming
Get the simplicity of SQL with freshness of streaming with one **unified API**

Build Production ETL Pipelines with DLT



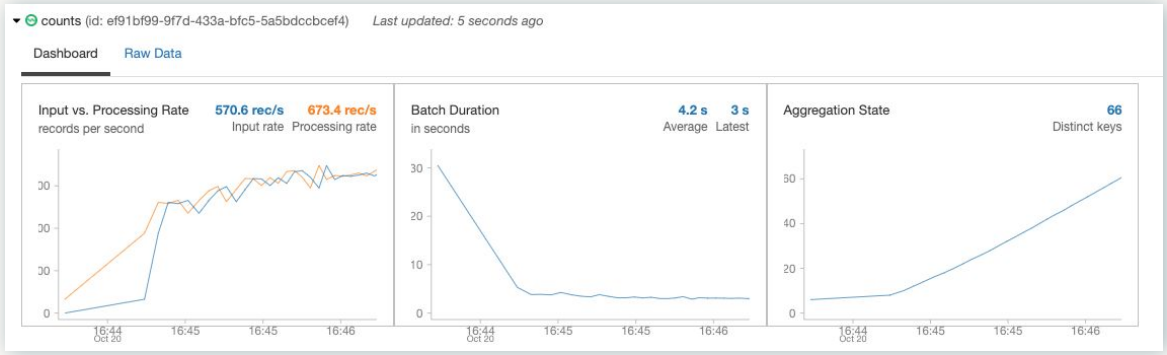
Key Differentiators

Continuous or scheduled data ingestion

Simple SQL syntax for streaming ingestion

Cmd 2

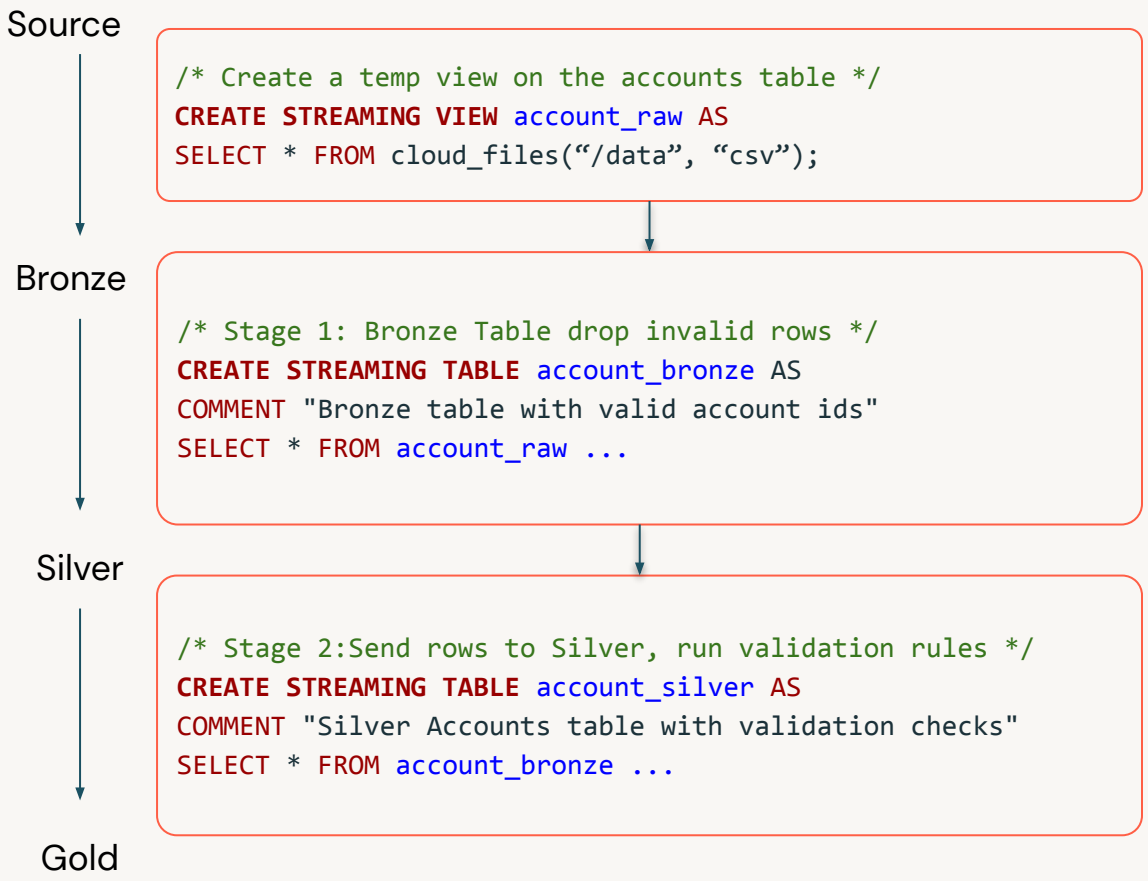
```
1  CREATE STREAMING LIVE TABLE sales_orders_raw
2  COMMENT "The raw sales orders, ingested from /databricks-datasets."
3  TBLPROPERTIES ("myCompanyPipeline.quality" = "bronze")
4  AS
5  SELECT * FROM cloud_files
6  ("/databricks-datasets/retail-org/sales_orders/",
7  "json", map("cloudFiles.inferColumnTypes", "true"))
8
```



- Incrementally and efficiently process new data files as they arrive in cloud storage using Auto Loader
- Automatically infer schema of incoming files or superimpose what you know with Schema Hints
- Automatic schema evolution
- Rescue data column – never lose data again

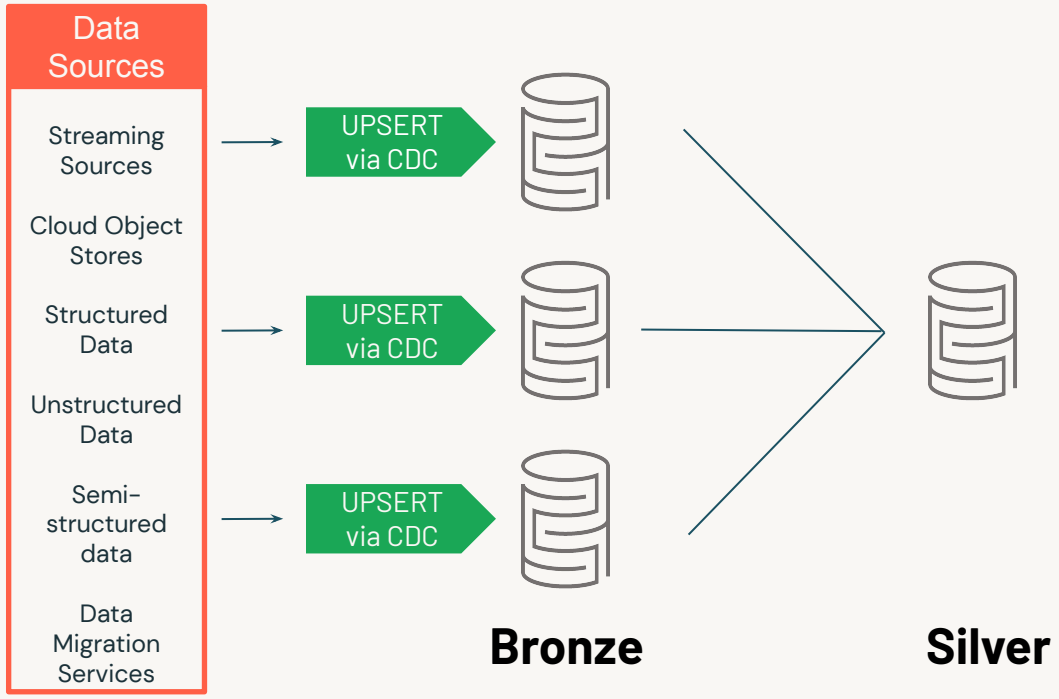
Schema Evolution	JSON	CSV	AVRO	PARQUET
	✅	✅	✅	✅

Declarative SQL & Python APIs



- Use intent-driven declarative development to abstract away the **"how"** and define **"what"** to solve
- Automatically generate **lineage** based on table dependencies across the data pipeline
- Automatically checks for errors, missing dependencies and syntax errors

Change data capture (CDC)

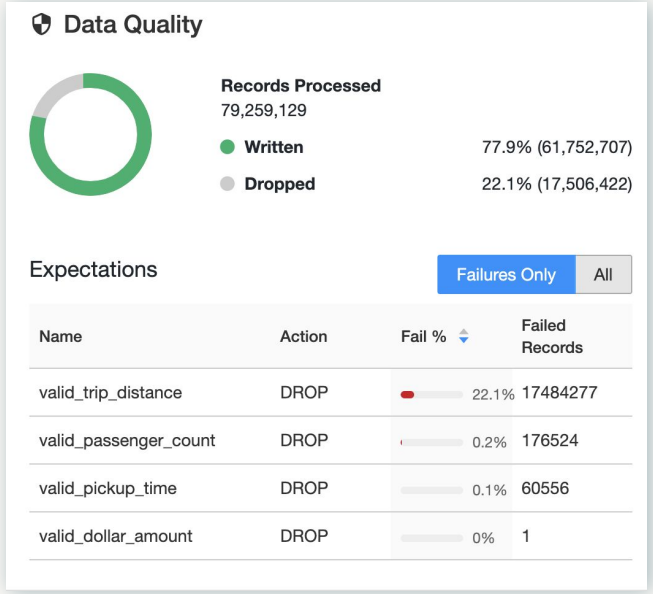


- Stream change records (inserts, updates, deletes) from any data source supported by DBR, cloud storage, or DBFS
- Simple, declarative "APPLY CHANGES INTO" API for SQL or Python
- Handles out-of-order events
- Schema evolution
- SCD2 support

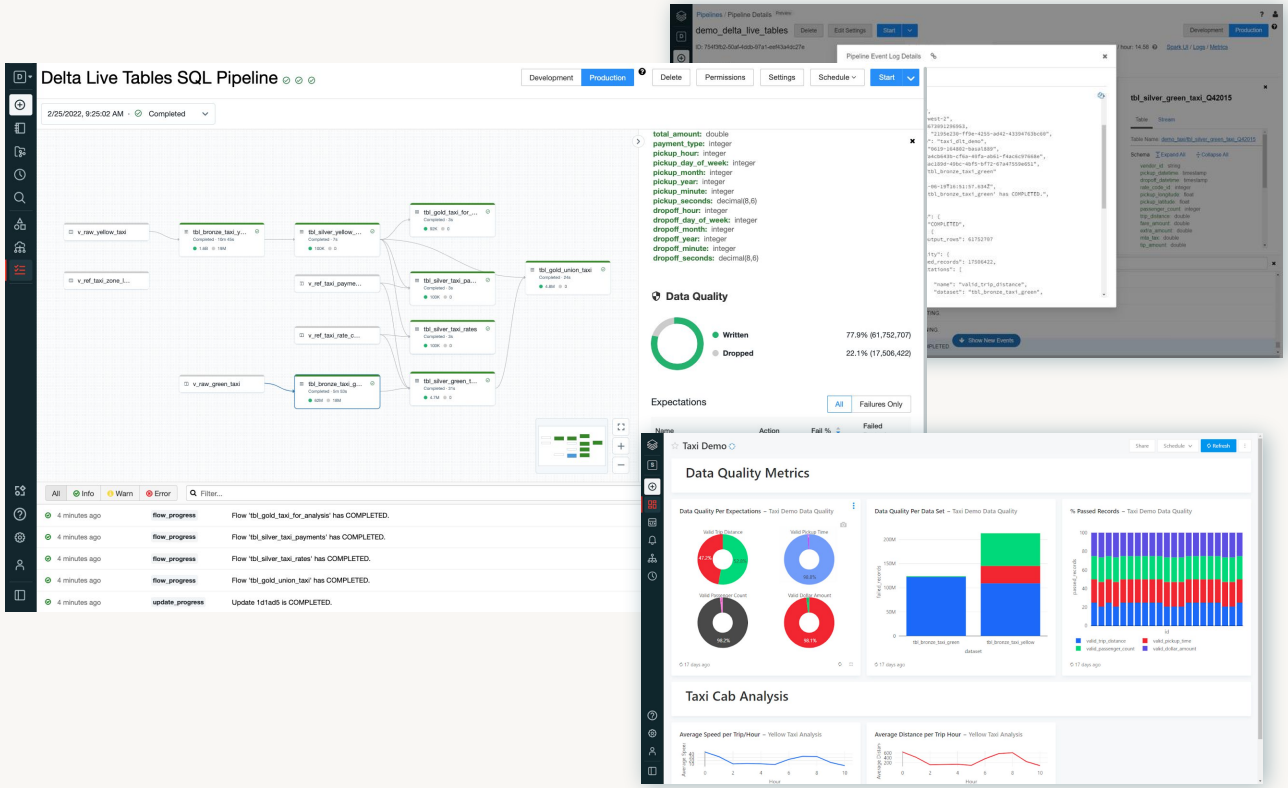
Data quality validation and monitoring

- Define data quality and integrity controls within the pipeline with data expectations
- Address data quality errors with flexible policies: fail, drop, alert, quarantine(future)
- All data pipeline runs and quality metrics are captured, tracked and reported

```
/* Stage 1: Bronze Table drop invalid rows */
CREATE STREAMING LIVE TABLE fire_account_bronze AS
( CONSTRAINT valid_account_open_dt EXPECT (acconut_dt is not
null and (account_close_dt > account_open_dt)) ON VIOLATION DROP
ROW
COMMENT "Bronze table with valid account ids"
SELECT * FROM fire_account_raw ...
```



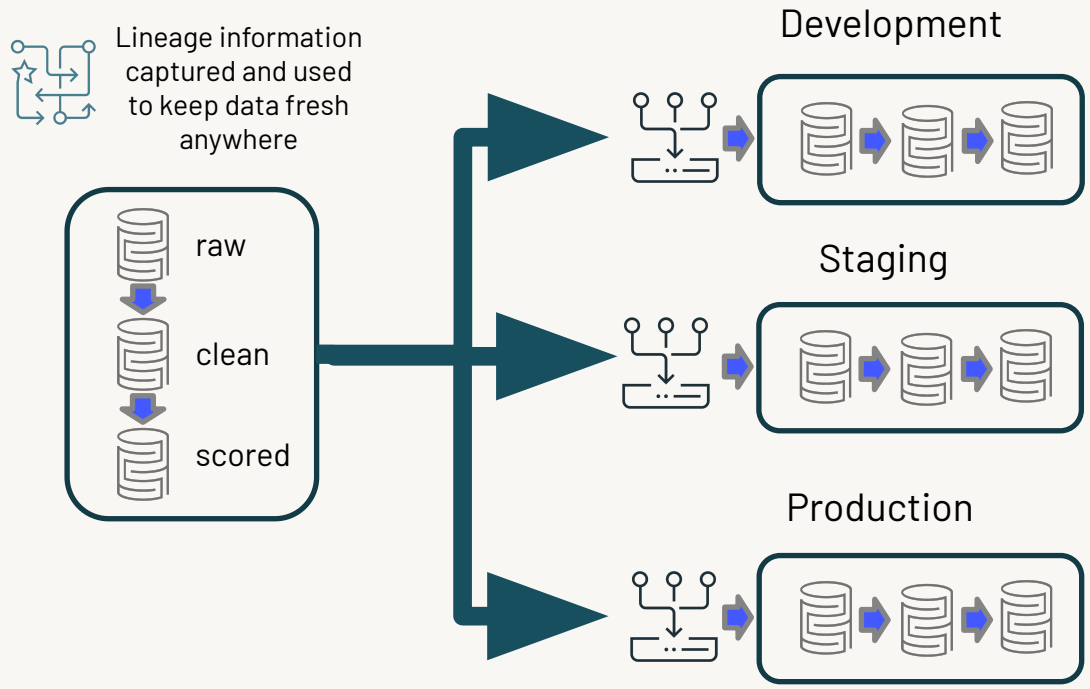
Data pipeline observability



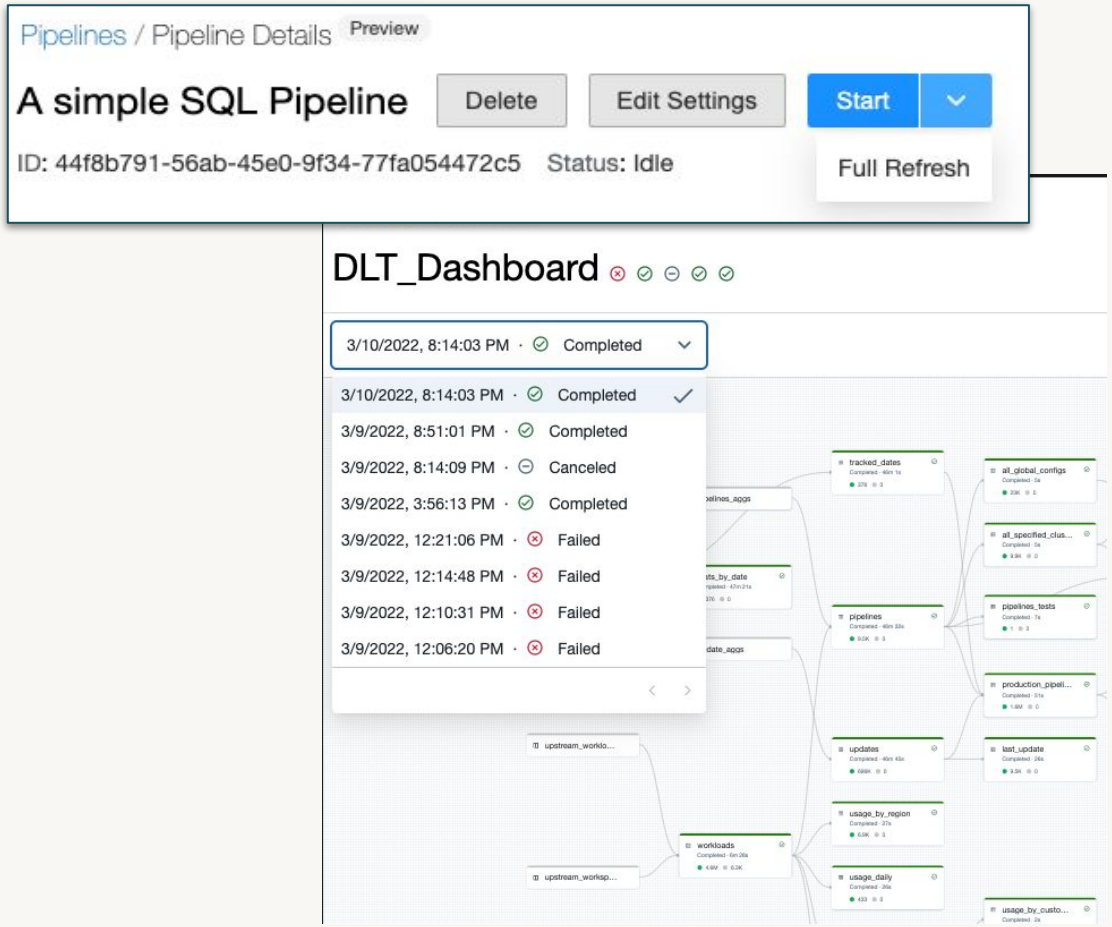
- High-quality, high-fidelity lineage diagram that provides visibility into how data flows for impact analysis
- Granular logging for operational, governance, quality and status of the data pipeline at a row level
- Continuously monitor data pipeline jobs to ensure continued operation
- Notifications using Databricks SQL

Automated ETL development lifecycle

- Develop in environment(s) separate from production with the ability to easily test it before deploying - entirely in SQL
- Deploy and manage environments using parameterization
- Unit testing and documentation
- Enables metadata-driven ability to programatically scale to 100s of tables/pipelines dynamically



Automated ETL operations



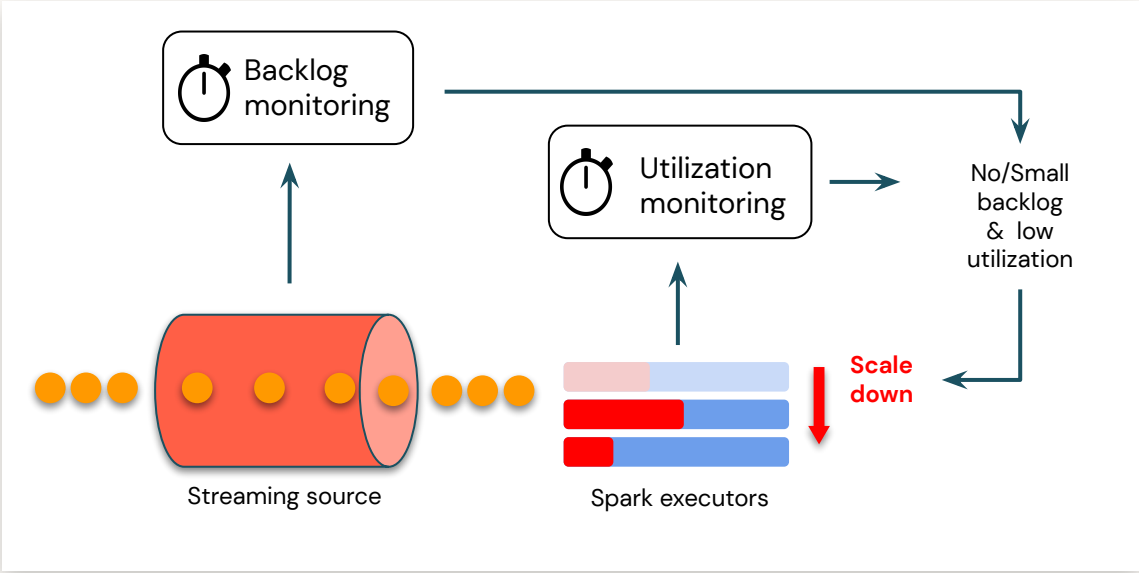
- Reduce down time with automatic error handling and easy replay
- Eliminate maintenance with automatic optimizations of all Delta Live Tables
- Auto-scaling adds more resources automatically when needed.

Enhanced Autoscaling

Save infrastructure costs while maintaining end-to-end latency SLAs for streaming workloads

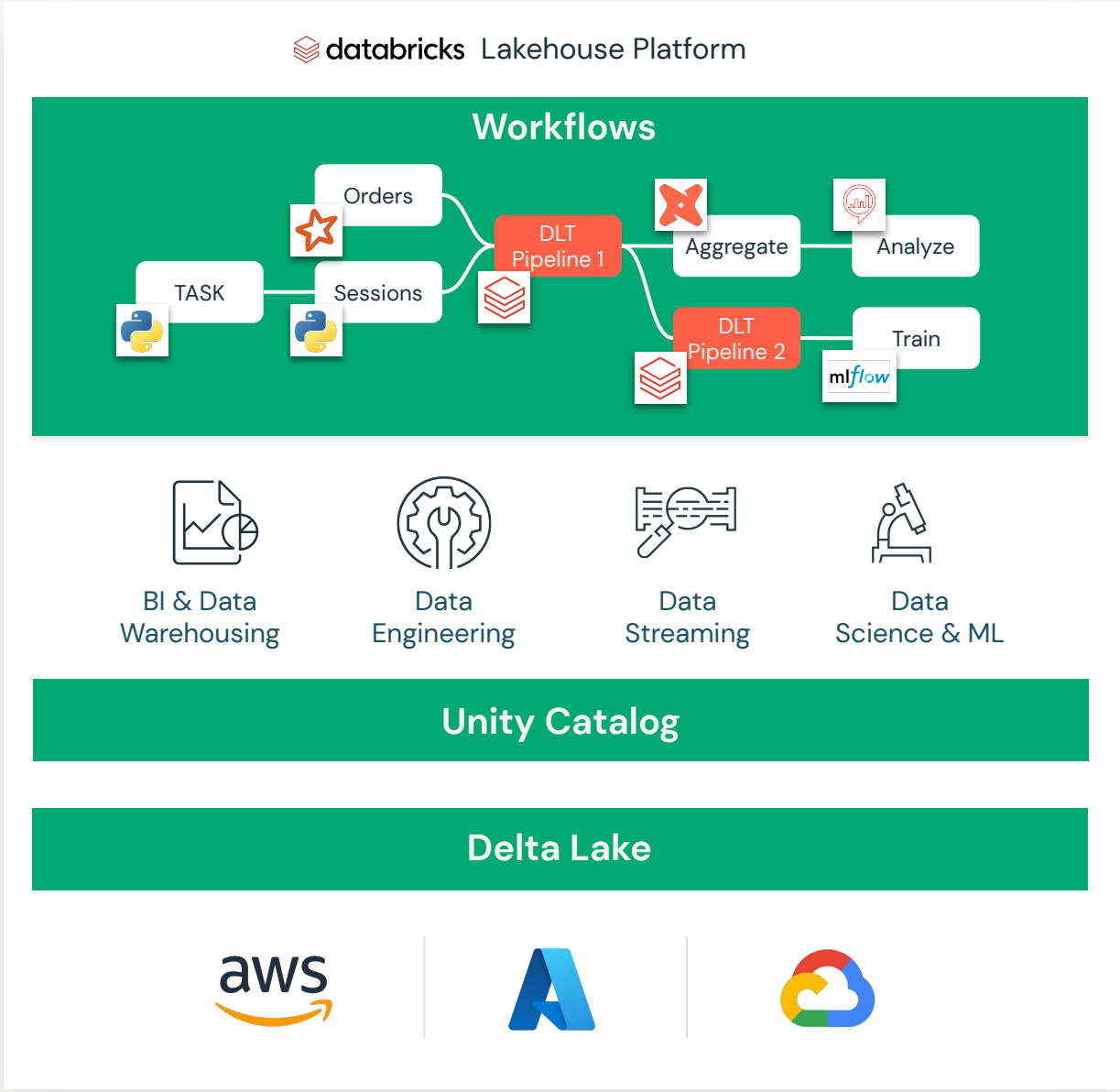
Problem

Optimize infrastructure spend when making scaling decisions for streaming workloads



- Built to handle streaming workloads which are spiky and unpredictable
- Shuts down nodes when utilization is low while guaranteeing task execution
- Only scales up to needed # of nodes

AWS	Azure	GCP
Generally Available	Generally Available	Public Preview GA Coming Soon



Databricks Workflows

Unified orchestration for Delta Live Tables pipelines and more

Simple workflow authoring for all data practitioners

Data practitioners can easily orchestrate DLT pipelines and other tasks from inside their Databricks workspace. Advanced users can use their favorite IDE's with full support for CI/CD.

Actionable insights from real-time monitoring

Full visibility into every task in every workflow. See the health of all your production workloads in real-time with detailed metrics and analytics to identify, troubleshoot, and fix issues fast.

Proven reliability for production workloads

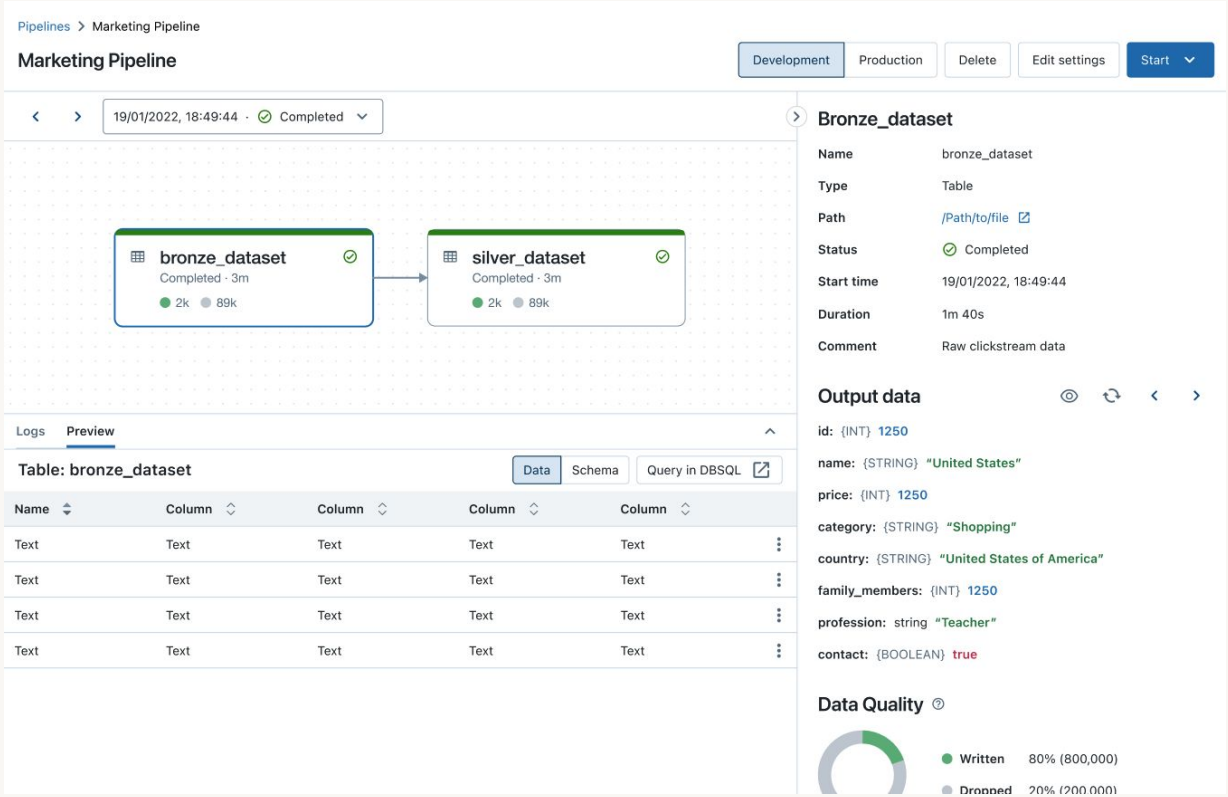
A fully managed orchestration service with serverless data processing and a history of 99.95% uptime. Trusted by thousands of Databricks customers running millions of production workloads.

Upcoming Features

Accelerate development

Improve time to insight with improved navigation and data management tools

- **Logical schema grouping**
  Group tables into different schemas depending on their contents.
- **Data sample previews**
  Easily preview data directly in the DLT console
- **Notebook integration**
  Users can run & see DLT pipelines update status from the Notebook



AWS	Azure	GCP
In development Preview Coming Soon	In development Preview Coming Soon	In development Preview Coming Soon

Change-data-capture (CDC) enhancements

Problem: It's hard to ingest & track changes with CDC from streaming or batch sources

city_updates

```
{"id": 1, "ts": 1, "city": "Bekerly, CA"}
{"id": 1, "ts": 2, "city": "Berkeley, CA"}
```

cities

id	city	__starts_at	__ends_at
1	Bekerly, CA	1	2
1	Berkeley, CA	2	null

```
APPLY CHANGES INTO LIVE.cities
FROM STREAM(LIVE.city_updates)
KEYS (id)
SEQUENCE BY ts
STORED AS SCD TYPE 2
```

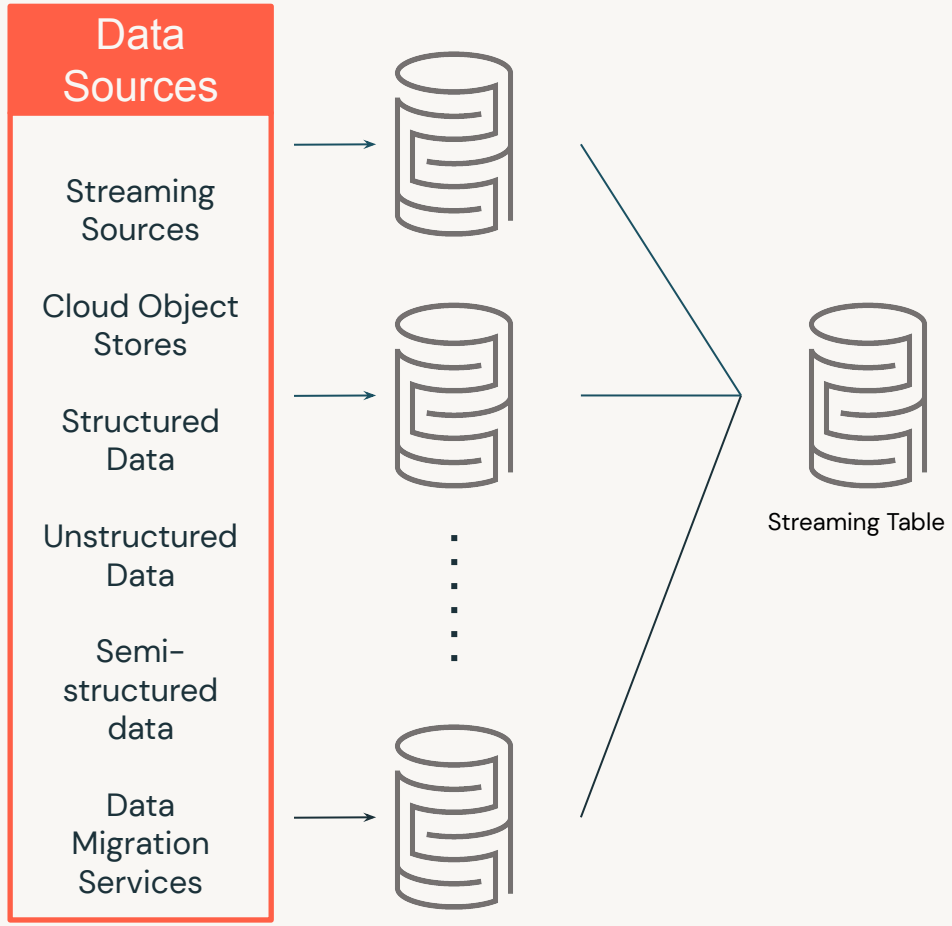
- **CDC from Full Snapshot**
  Perform change-data-capture from any source by providing full snapshots. Supports both SCD type 1 and type 2.

- **SCD type 2 GA**
  Track changes and retain full history of values. When the value of an attribute changes, the current record is closed, a new record is created with the changed data values.

	AWS	Azure	GCP
CDC from Full Snapshot	Preview GA Coming Soon	Preview GA Coming Soon	Preview GA Coming Soon
CDC SCD TYPE 2	Preview GA Coming Soon	Preview GA Coming Soon	Preview GA Coming Soon

Seamlessly add new streaming sources

Add new sources to a streaming table without a full refresh

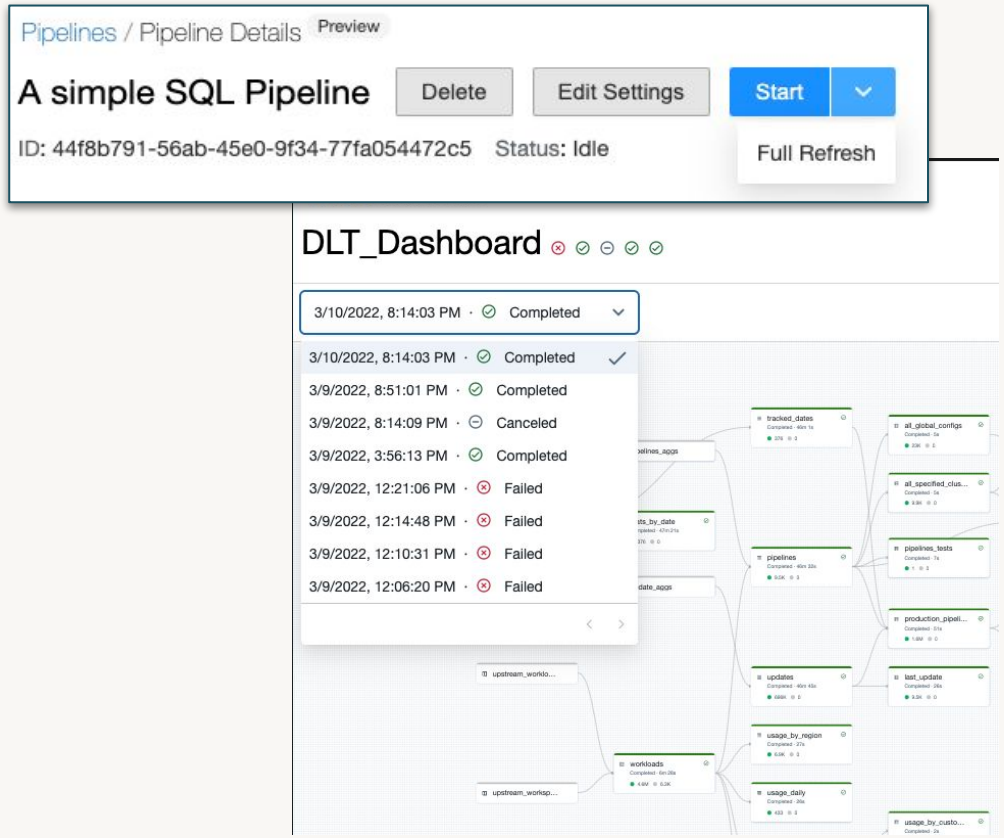


- Declarative API to define sources in DLT
- Adding a new streaming source without doing a full refresh
- Works with any streaming source supported by DLT

AWS	Azure	GCP
Preview GA Coming Soon	**Preview** GA Coming Soon	**Preview** GA Coming Soon

Simplify Operations

- **Email notifications**
  Configure detailed notifications at the individual table level for continuous and triggered pipelines.
- **Improved errors**
  Debug issues more quickly with detailed and accurate error reports in DLT.



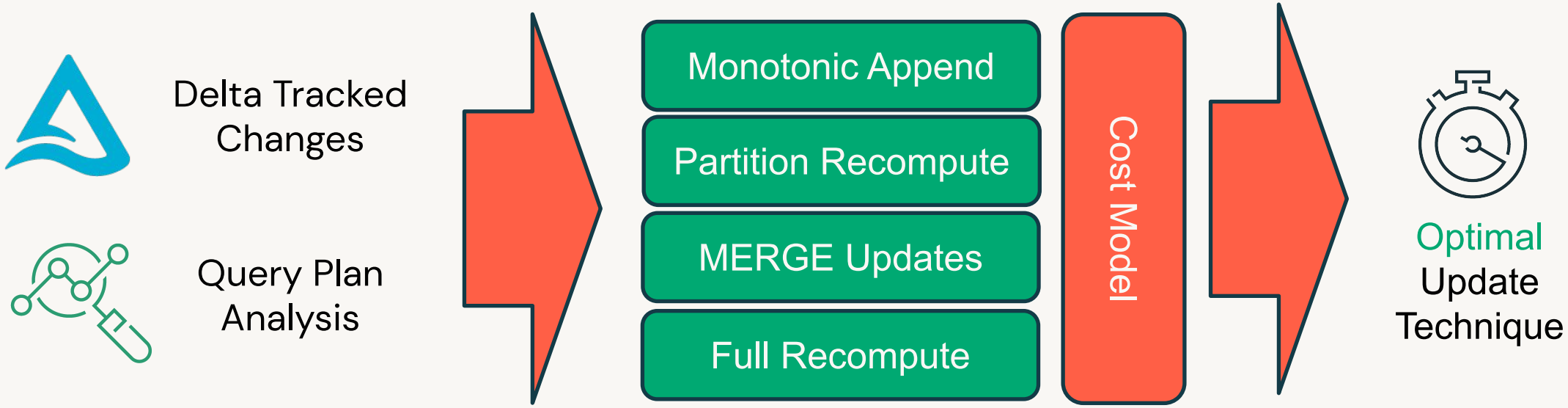
AWS	Azure	GCP
In Development Preview Coming Soon	In Development Preview Coming Soon	In Development Preview Coming Soon

Enzyme performance optimization for MVs

Improve end-to-end SLAs and reduce infrastructure costs by incrementally updating materialized views.

Problem

Achieving near real-time data freshness for streaming pipelines with large data volumes requires significant infrastructure spend or complex hand-coding.

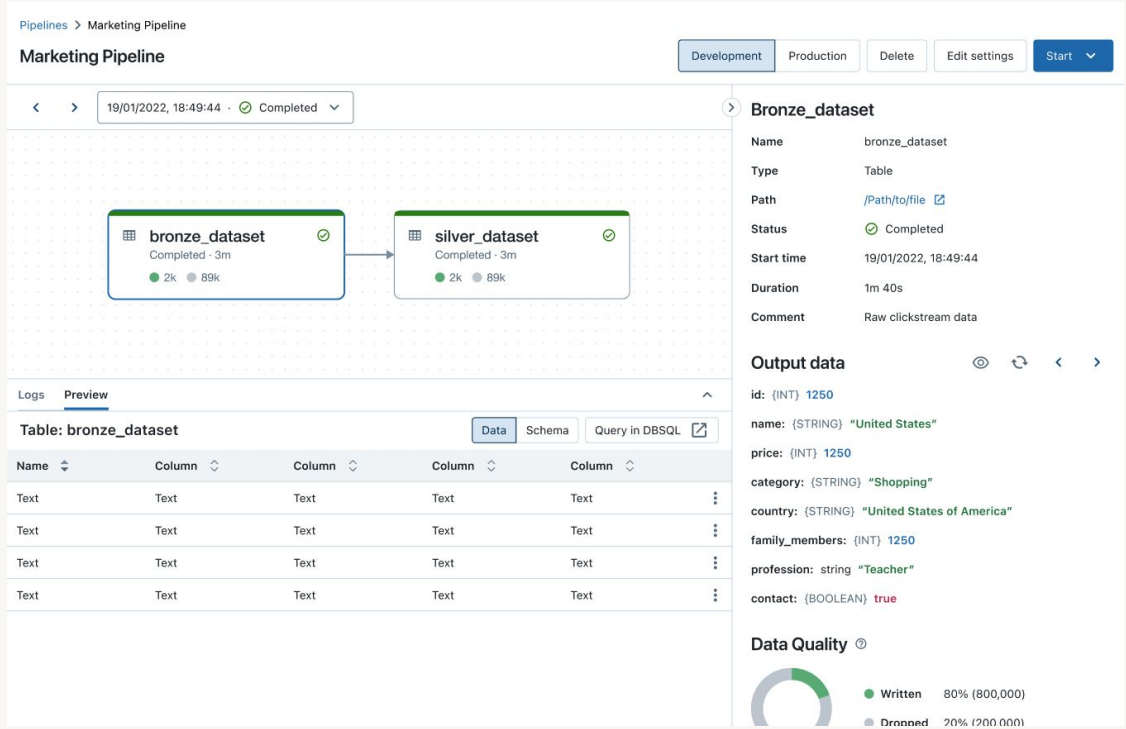


AWS	Azure	GCP
Private Preview	Private Preview	Private Preview

Data Samples

Easily preview data directly in the DLT console

- Single-click to view a selection of rows for any table in your pipeline
- Easily perform ad-hoc queries with Databricks SQL Warehouse query editor
- View detailed schema information



AWS	Azure	GCP
In Development Preview Coming Soon	In Development Preview Coming Soon	In Development Preview Coming Soon

Thank you