databricks

Databricks Lakehouse Bootcamp

Your Hosts

Yasamin Mokri
Specialised Solutions Architect

Wilbur Tong
Senior Customer Success Engineer

Alistair Thrussell
Senior Solutions Architect

Agenda – 2 Days

Day 1

General

0900 – Welcome & Introduction
0910 – Databricks Overview
0930 – Lakehouse / Demo

Data Engineering

0950 – Data Engineering Overview
1020 – Lab Setup
1030 – Morning Tea
1045 – Data Engineering Lab
1300 – Lunch

Analytics/BI

1400 – Analytics/BI Overview
1420 – Databricks SQL Lab
1445 – Afternoon Tea
1500 – Databricks SQL Lab Cont'd
1530 – Databricks Roadmap

Day 2

Data Science

0950 – Databricks ML Overview
1020 – Lab Setup & Break
1030 – Databricks ML Lab

Wrap Up

1200 – Closing Comments
1210 – Q&A

Databricks Overview

databricks

Databricks

The Data + AI Company

Inventor and pioneer of the
data lakehouse

Gartner recognized leader in both
- Database Management Systems
- Data Science and Machine Learning Platforms

Creator of highly successful OSS data projects: Delta Lake, Apache Spark, and MLflow

Raised over $3B in investment

3000+ employees across the globe

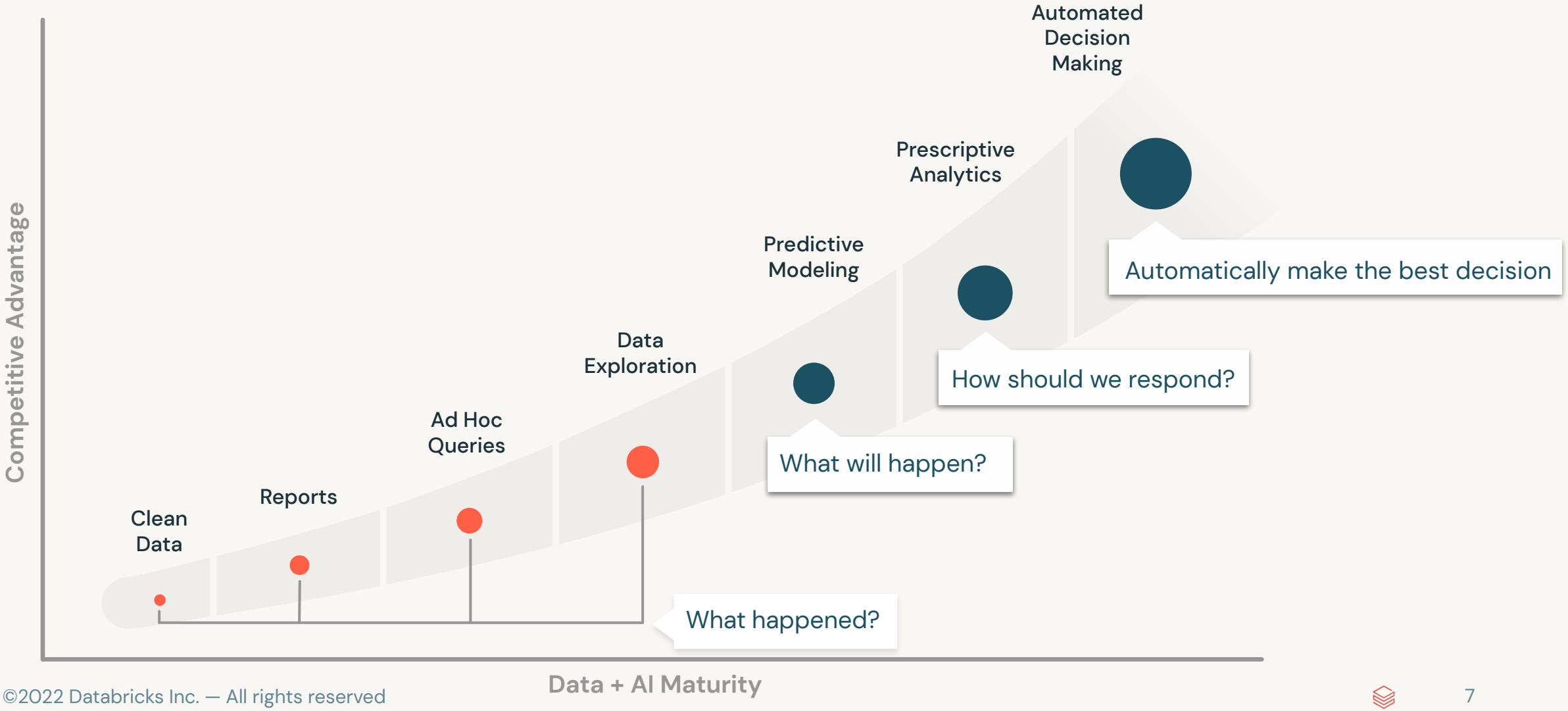
Global adoption

Over 7000 customers, from F500 to unicorns

Data, analytics, and AI enabled tech's leaders to disrupt industries

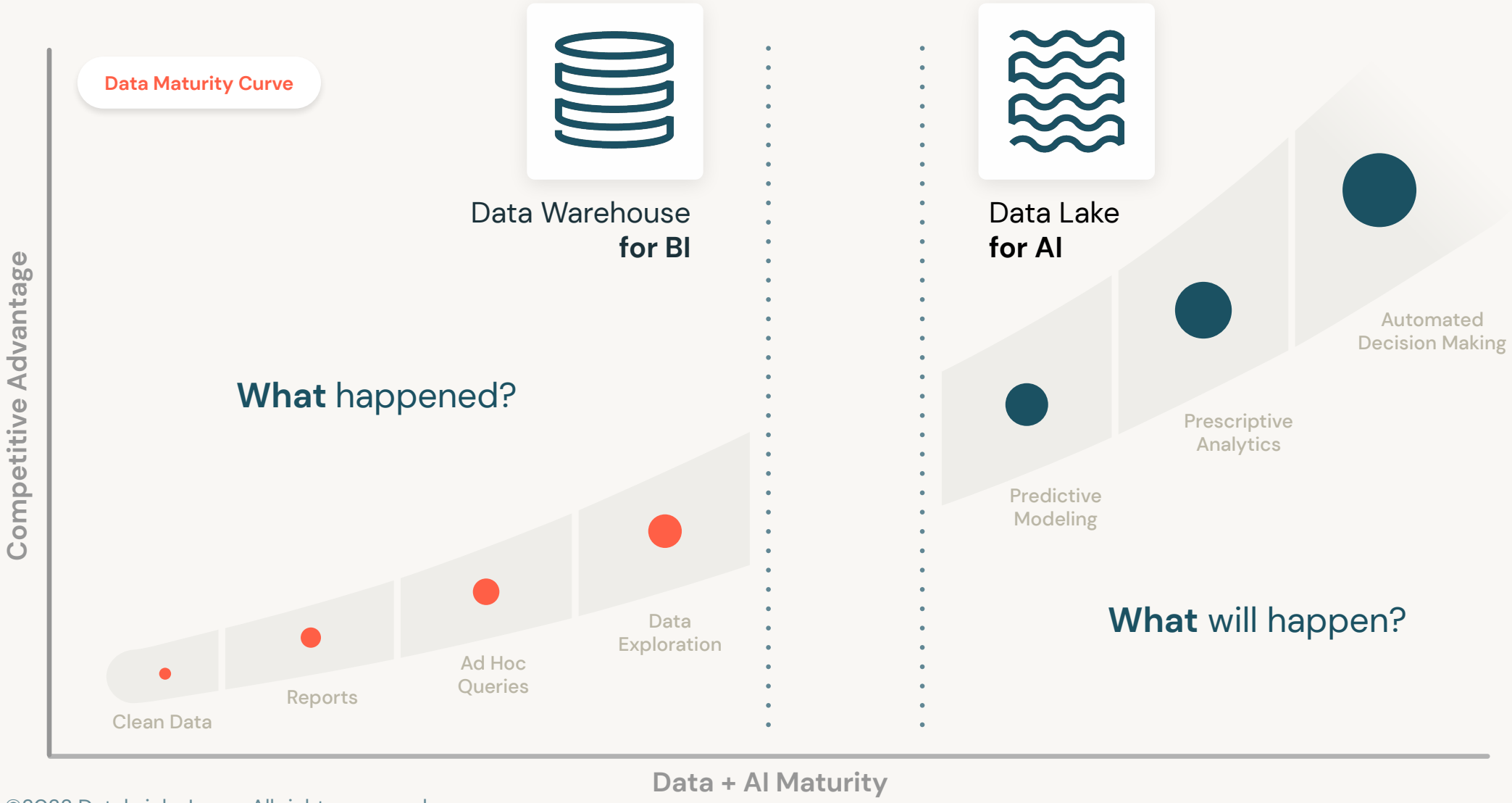
Tech leaders are to the right of the Data Maturity Curve

From hindsight to foresight

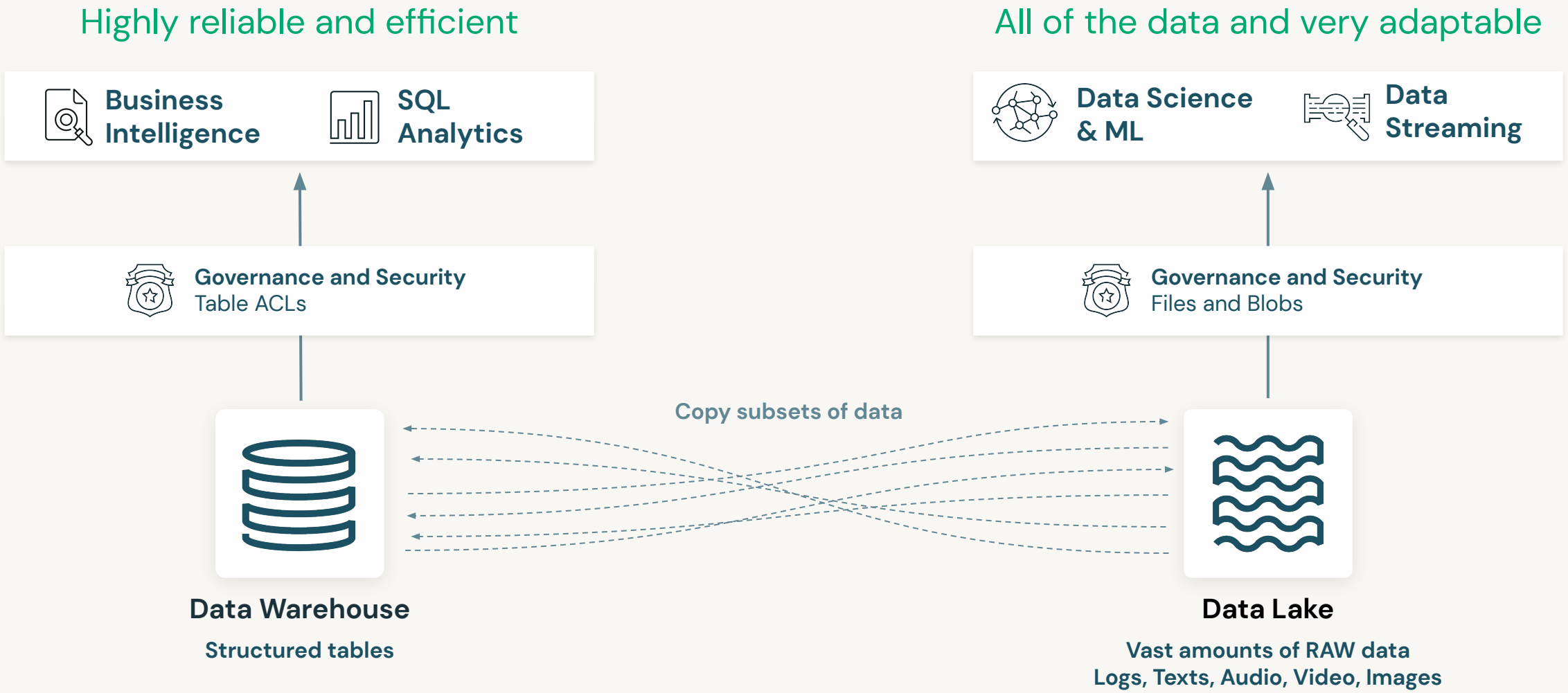


Most enterprises still struggle with data, analytics, and AI

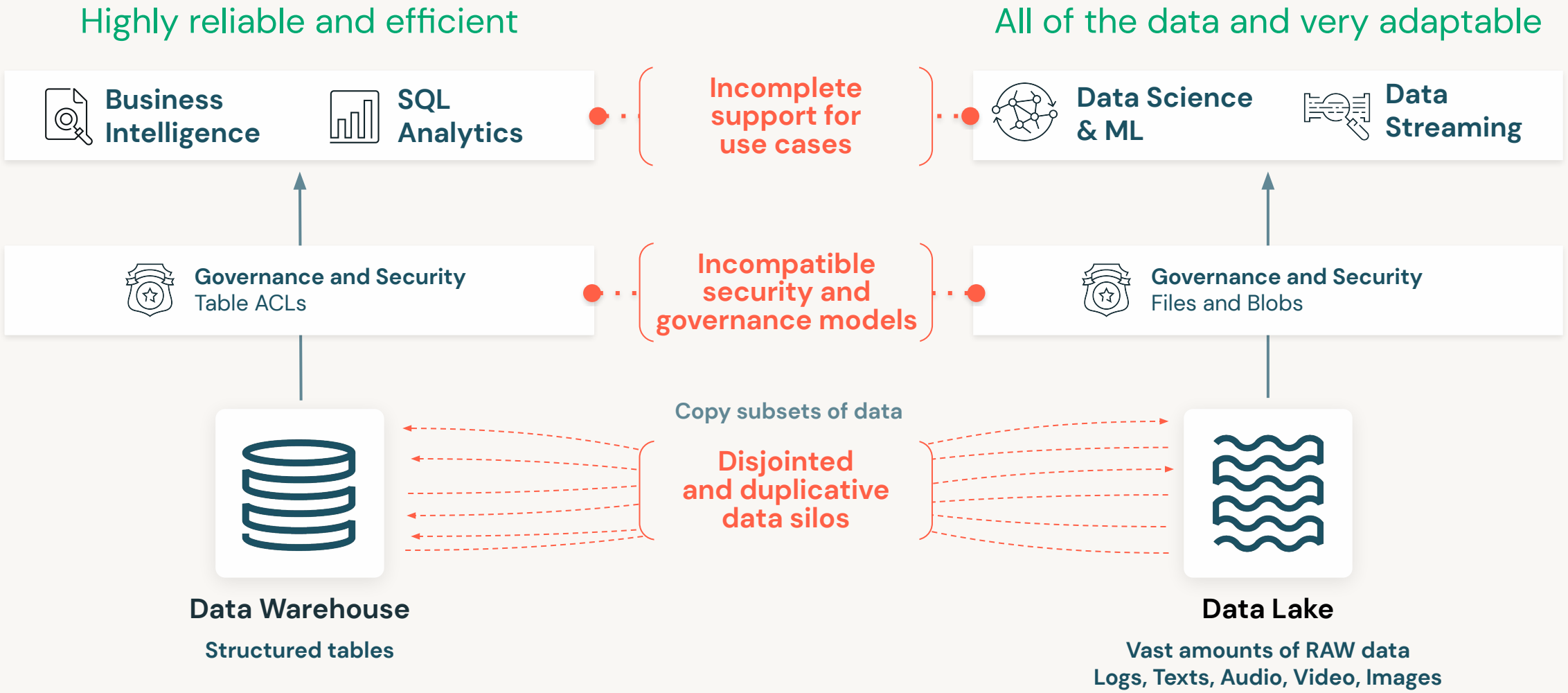
Realizing this requires two disparate, incompatible data platforms



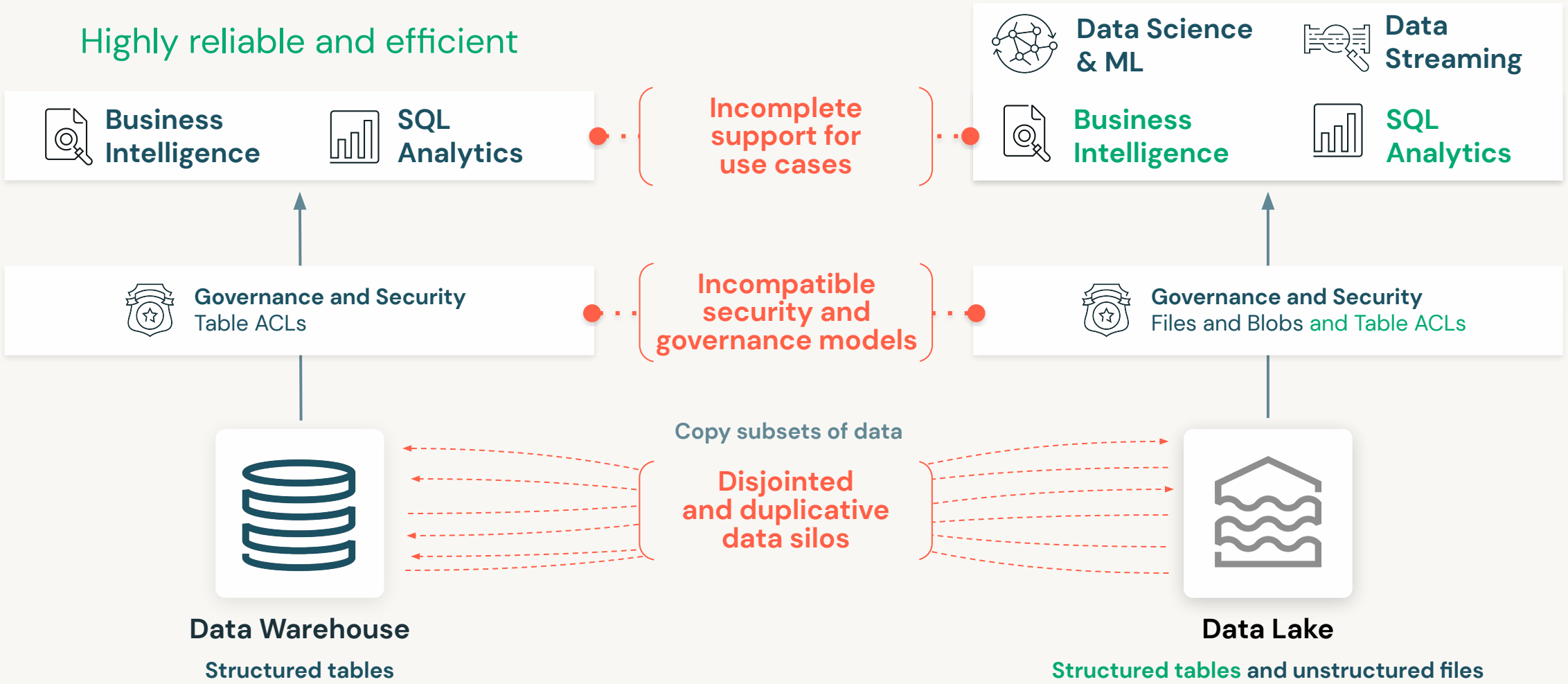
Realizing this required two disparate, incompatible data platforms



Realizing this required two disparate, incompatible data platforms



There is no need to have two disparate platforms

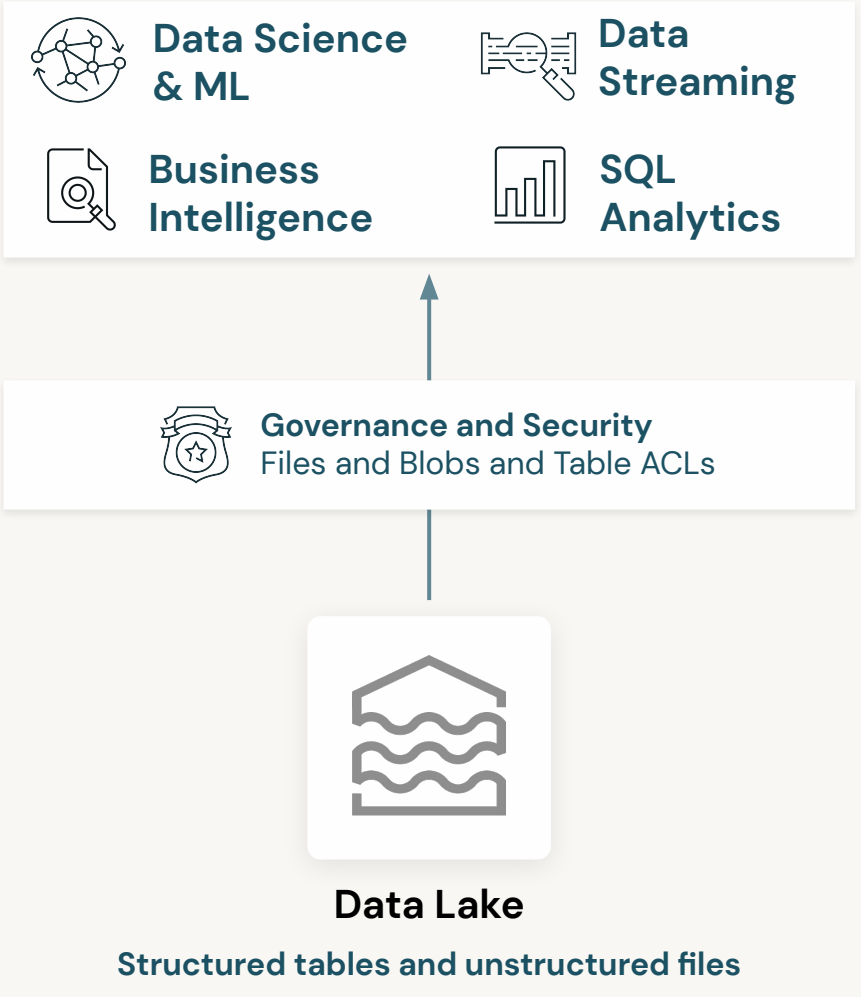


This is the lakehouse paradigm

All ML, SQL, BI,
and Streaming use cases

One **security** and **governance**
approach for all data assets
on all clouds

A **reliable**
data platform
to **efficiently** handle all data types

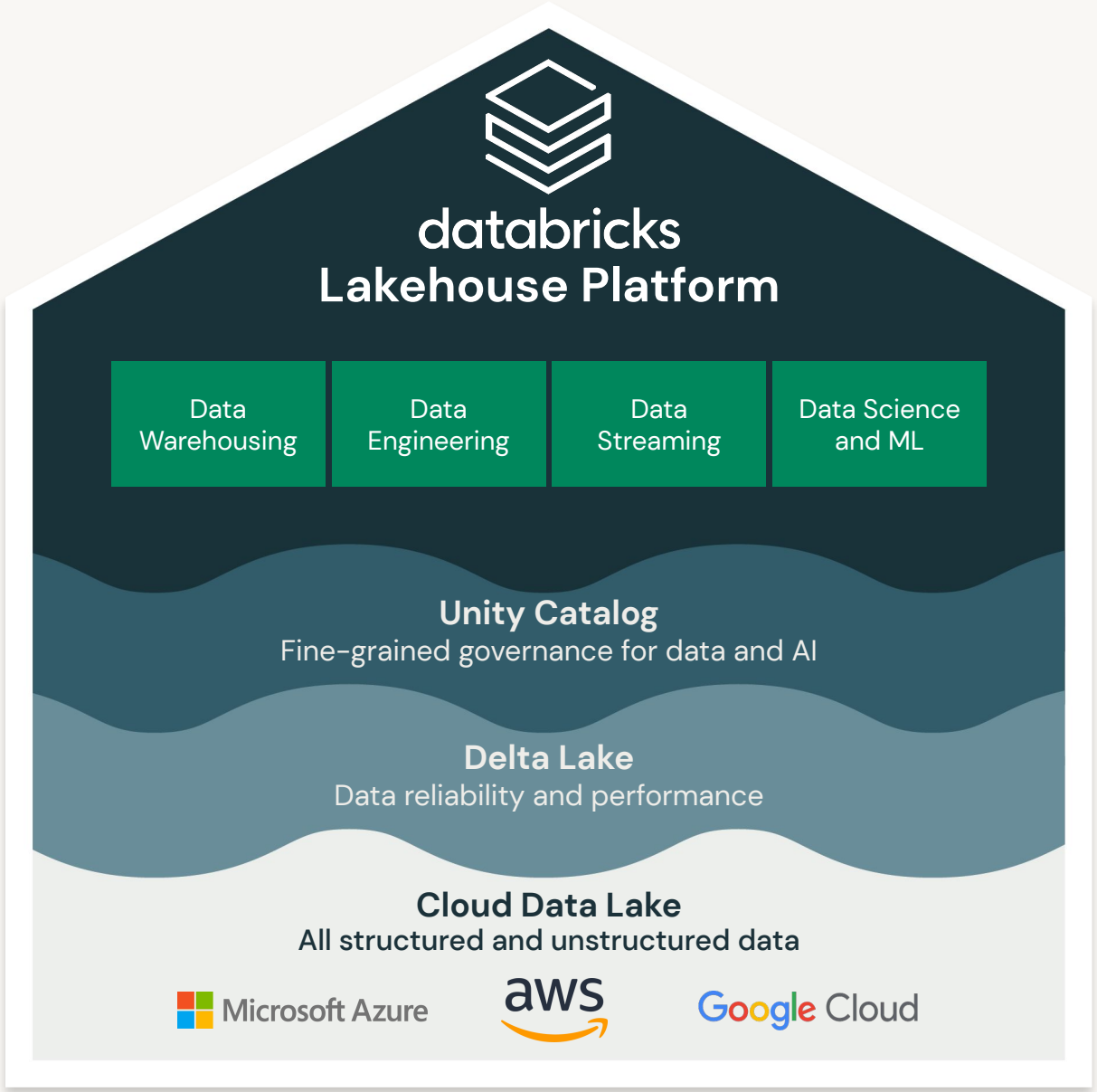


Technologies

Data Applications

Unity Catalog
Fine-grained governance
for data and AI

Delta Lake
Data reliability and performance



Databricks Lakehouse Platform

Simple

Unify your data warehousing and AI use cases on a single platform

Multicloud

One consistent data platform across clouds

Open

Built on open source and open standards

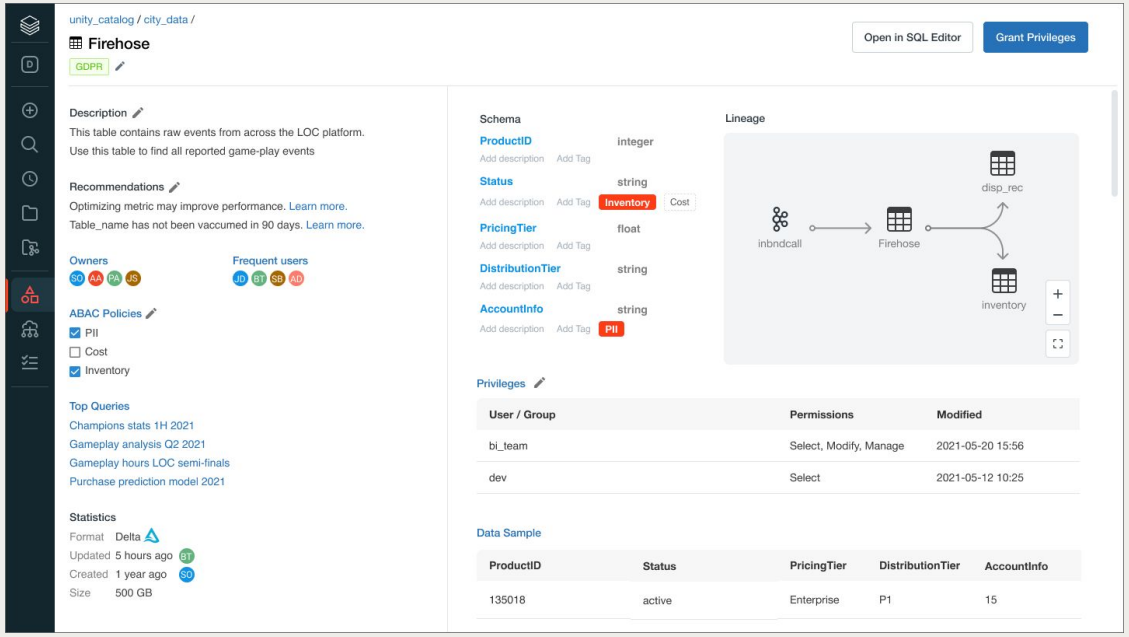
Unity Catalog for Lakehouse Governance

Govern and manage all data assets

- Warehouse, Tables, Columns
- Data Lake, Files
- Machine Learning Models
- Dashboards and Notebooks

Capabilities

- Data lineage
- Attribute-based access control
- Security policies
- Table or column level tags
- Auditing
- Data sharing



ML & data science workloads on Databricks

Machine Learning

- Model registry, reproducibility, productionization
- Leverages Delta Lake for reproducibility
- AutoML for citizen data scientists

Data Science

- Collaborative notebooks and dashboards for interactive analysis
- Native support for Python, Java, R, Scala
- Delta Lake data natively supported

Data engineering workloads on Databricks

- Data orchestration through Databricks Workflows
- Delta Live Tables manage your full data pipelines
- Simplifies data engineering with a curated data lake approach through Delta Lake

SQL workloads on Databricks

- Great performance and concurrency for BI and SQL workloads on Delta Lake
- Native SQL interface for analysts
- Support for BI tools to directly query your most recent data in Delta Lake

Databricks thrives within your modern data stack

BI and Dashboards

Power BI, tableau, Looker, MicroStrategy, ThoughtSpot, Qlik

Machine Learning

MathWorks, Labelbox, John Snow LABS, Azure Machine Learning, H2O.ai, Amazon SageMaker

Data Science

PyCharm, jupyter, HEX, RStudio

Data Governance

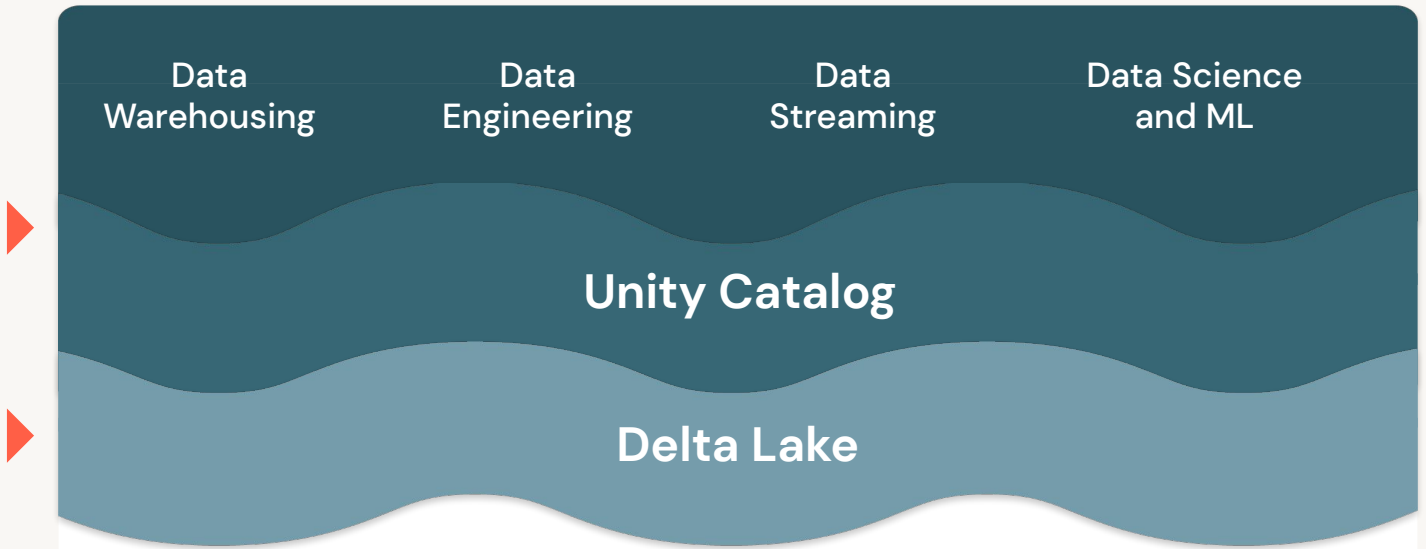
Collibra, IMMUTA, PRIVACERA, Quest, Alation, AZURE PURVIEW

Data Pipelines

dbt Labs, MATILLION, Azure Data Factory, Informatica, Prophecy

Data Ingestion

Fivetran, arcion, CONFLUENT, Rivery, Airbyte, Qlik



Consulting & SI Partners

accenture, avanade, Capgemini, Cognizant, Deloitte., slalom

Building your Lakehouse

Comprehensive investment into your success

Supported by 24/7/365 global, production operations at scale

Supporting enterprises in every industry

Healthcare & Life Sciences

gsk
Johnson & Johnson
Bayer
Merck
NHS
Amgen
Novartis
Sanofi
Regeneron
Biogen
Pfizer
Kaiser Permanente
AstraZeneca

Retail & CPG

Walmart
Nike
7-Eleven
Starbucks
Walgreens
H&M
CVS Health
Burberry
Coca-Cola
Best Buy
Equinox
McDonald's

Media & Entertainment

Condé Nast
Disney
Comcast
Publicis Groupe
Sony
NBCUniversal
Showtime
AT&T
ViacomCBS

Financial Services

HSBC
Santander
Goldman Sachs
Nationwide
AXA
GEICO
Nasdaq
S&P Global
Credit Suisse
Allianz

Public Sector

United States Postal Service
FDA
Navy
U.S. Citizenship and Immigration Services
U.S. Dept. of Defense
U.S. Army
U.S. Department of Energy
NIH
National Institutes of Health
JPL
Jet Propulsion Laboratory

Manufacturing & Logistics

Cummins
Rolls-Royce
Daimler
FedEx
GE
Schneider Electric
Airbus
BMW Group
thyssenkrupp

Energy & Utilities

Shell
TotalEnergies
Chevron
ExxonMobil
bp
eni
equinor
Schlumberger
Halliburton
Phillips 66

Digital Native

Robinhood
DoorDash
instacart
Rivian
Udemy
Grab
stripe
samsara
Square

Platform Demo

databricks

Agenda – 2 Days

Day 1

General

0900 – Welcome & Introduction
0910 – Databricks Overview
0930 – Lakehouse / Demo

Data Engineering

0950 – Data Engineering Overview
1020 – Lab Setup
1030 – Morning Tea
1045 – Data Engineering Lab
1300 – Lunch

Analytics/BI

1400 – Analytics/BI Overview
1420 – Databricks SQL Lab
1445 – Afternoon Tea
1500 – Databricks SQL Lab Cont'd
1530 – Databricks Roadmap

Day 2

Data Science

0950 – Databricks ML Overview
1020 – Lab Setup & Break
1030 – Databricks ML Lab

Wrap Up

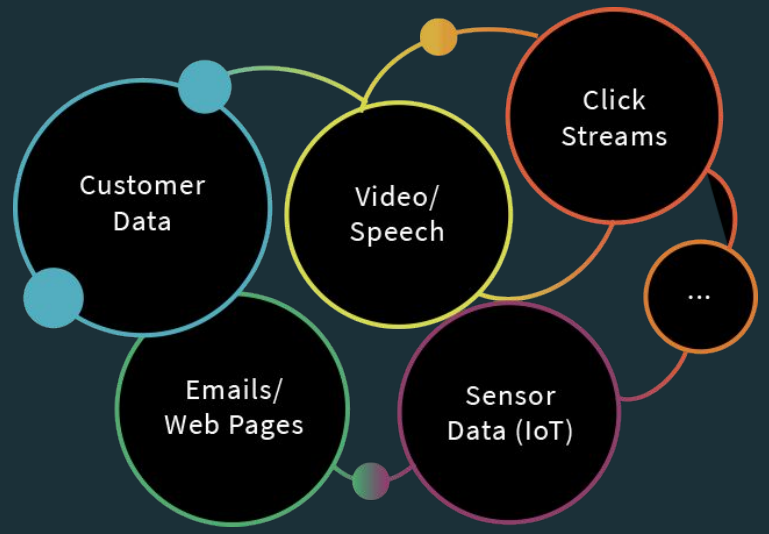
1200 – Closing Comments
1210 – Q&A

databricks

Delta Lake

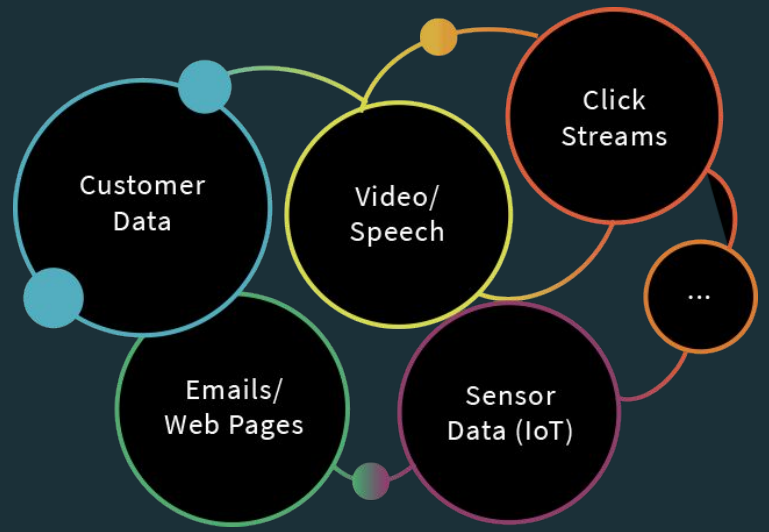
databricks

Enterprises have been spending millions of dollars getting data into data lakes with Apache Spark



Data Lake

The aspiration is to do data science and ML on all that data using Apache Spark!



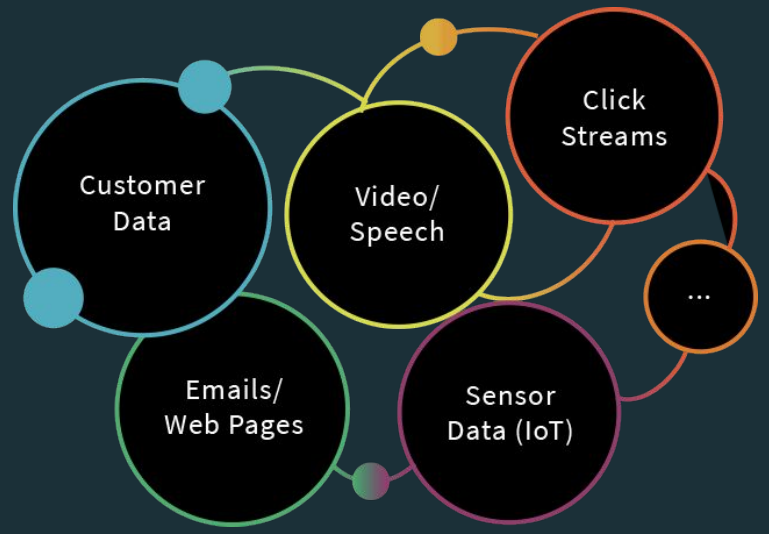
Data Lake

Data Science & ML

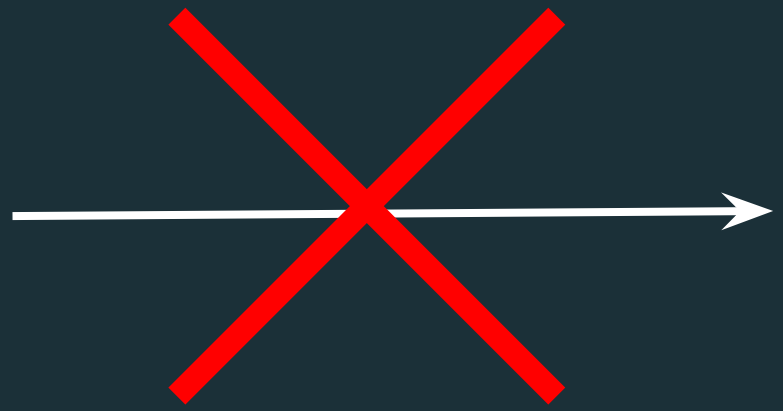
- Recommendation Engines
- Risk, Fraud Detection
- IoT & Predictive Maintenance
- Genomics & DNA Sequencing

But the data is not ready for data science & ML

The **majority** of these projects are failing due to **unreliable data and poor performance!**



Data Lake



Data Science & ML

- Recommendation Engines
- Risk, Fraud Detection
- IoT & Predictive Maintenance
- Genomics & DNA Sequencing

databricks

Why are these projects struggling with **reliability** and **performance**?

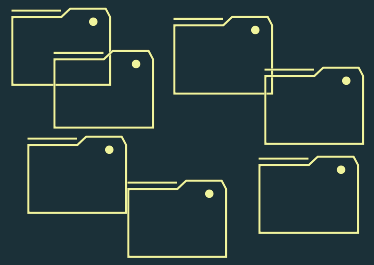
Data reliability challenges with data lakes

Failed production jobs leave data in corrupt state requiring tedious recovery

Lack of schema enforcement creates inconsistent and low quality data

Lack of consistency makes it almost impossible to mix appends and reads, batch and streaming

Performance challenges with data lakes



Too many small or very big files - more time opening & closing files rather than reading contents (worse with streaming).

Partitioning aka "poor man's indexing"– breaks down if you picked the wrong fields or when data has many dimensions, high cardinality columns.

No caching – cloud storage throughput is low (cloud object storage is 20–50MB/s/core vs 300MB/s/core for local SSDs).

A New Standard for Building Data Lakes

Open Format Based on Parquet

With Transactions

Apache Spark API's

Delta Lake: makes data ready for analytics

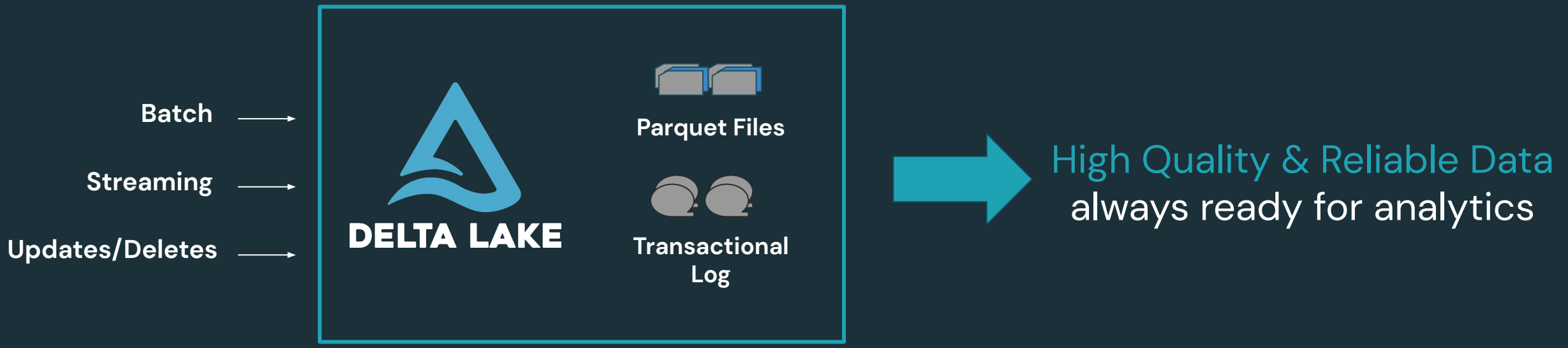
Reliability

Performance

Data Science & ML

- Recommendation Engines
- Risk, Fraud Detection
- IoT & Predictive Maintenance
- Genomics & DNA Sequencing

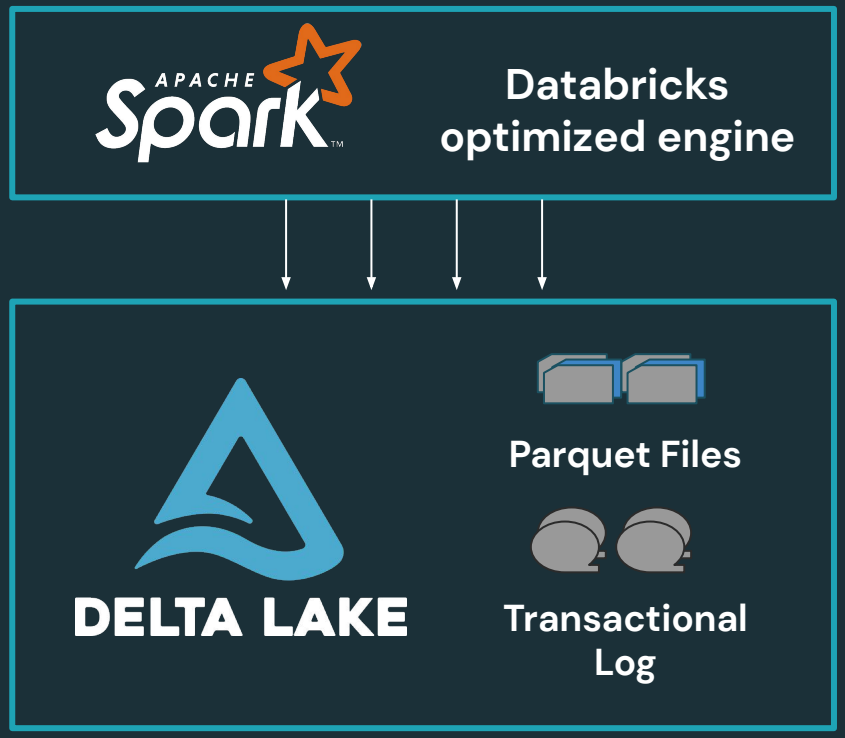
Delta Lake ensures data reliability



Key Features

- ACID Transactions
- Schema Enforcement
- Unified Batch & Streaming
- Time Travel/Data Snapshots

Delta Lake optimizes performance



Highly Performant queries at scale

Key Features

- Indexing
- Compaction
- Data skipping
- Caching

Get Started with Delta using Spark APIs

Instead of **parquet**...

```
CREATE TABLE ...
USING parquet
...

dataframe
    .write
    .format("parquet")
    .save("/data")
```

... simply say **delta**

```
CREATE TABLE ...
USING delta
...

dataframe
    .write
    .format("delta")
    .save("/data")
```

Using Delta with your Existing Parquet Tables

Step 1: Convert Parquet to Delta Tables

```
CONVERT TO DELTA parquet.`path/to/table` [NO STATISTICS]
[PARTITIONED BY (col_name1 col_type1, col_name2 col_type2, ...)]
```

Step 2: Optimize Layout for Fast Queries

```
OPTIMIZE events
WHERE date >= current_timestamp() - INTERVAL 1 day
ZORDER BY (eventType)
```

Upsert/Merge: Fine-grained Updates

```
%sql
MERGE INTO customers     -- Delta table
USING updates
ON customers.customerId = updates.customerId
WHEN MATCHED THEN
    UPDATE SET address = updates.address
WHEN NOT MATCHED
    THEN INSERT (customerId, address) VALUES (updates.customerId, updates.address)
```

Time Travel

Reproduce experiments & reports

```sql
%sql
-- Lookup data "at given time"
SELECT count(*) FROM events
TIMESTAMP AS OF timestamp

-- Lookup data "at given version"
SELECT count(*) FROM events
VERSION AS OF version
```

```python
%python
# Lookup data "at given time"
spark.read.format("delta") \
  .option("timestampAsOf", ts) \
  .load("/events/")

# Lookup data "at given version"
spark.read.format("delta") \
  .option("version", version) \
  .load("/events/")
```

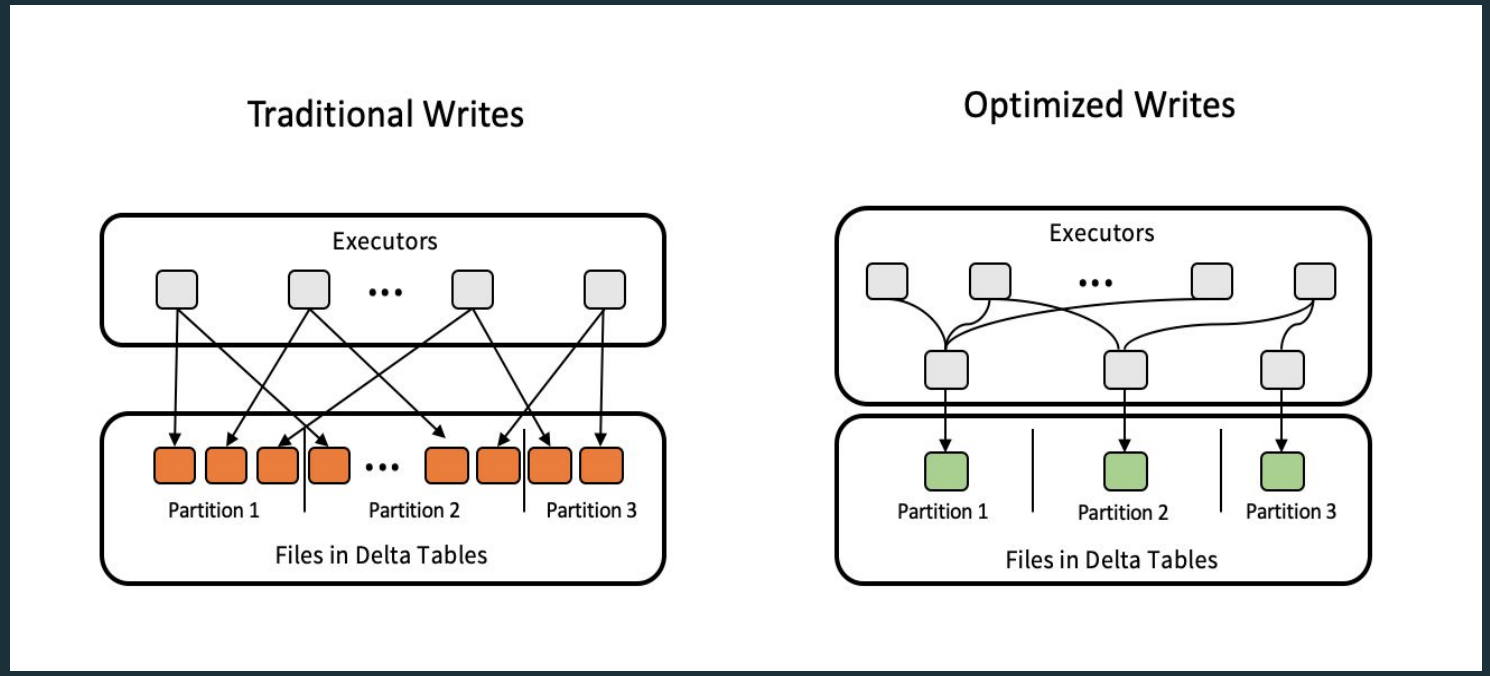
Time Travel

Rollback accidental bad writes

```
%sql
# Restore table to yesterday's state
RESTORE my_table TO
TIMESTAMP AS OF date_sub(current_date(), 1)
```

Auto Optimize & Auto Compact

```
set spark.databricks.delta.properties.defaults.autoOptimize.optimizeWrite = true;
set spark.databricks.delta.properties.defaults.autoOptimize.autoCompact = true;
```



databricks

Getting started with Delta Change Data Feed

Enable CDF for table

```
%sql
ALTER TABLE ...
SET TBLPROPERTIES (delta.enableChangeDataCapture = true);
```

Using Delta Change Data Feed

Query **changes…**

```
%sql
-- Select changes by version
SELECT … FROM
table_changes('tableName',
     10, -- Start version
     12) -- End version

-- Select changes by timestamp
SELECT … FROM
table_changes('tableName',
    start_ts,
     end_ts)
```

… and store **them**

```
%sql
-- Insert changes to destination table
INSERT INTO TABLE ...
USING DELTA ...
AS
SELECT … FROM table_changes(...)
```

in SQL, Python, or Scala

Getting started with Delta Change Data Feed

```
1  %sql
2  SELECT * FROM table_changes('silverTable', 2, 4) order by _commit_timestamp
```

▸ (1) Spark Jobs

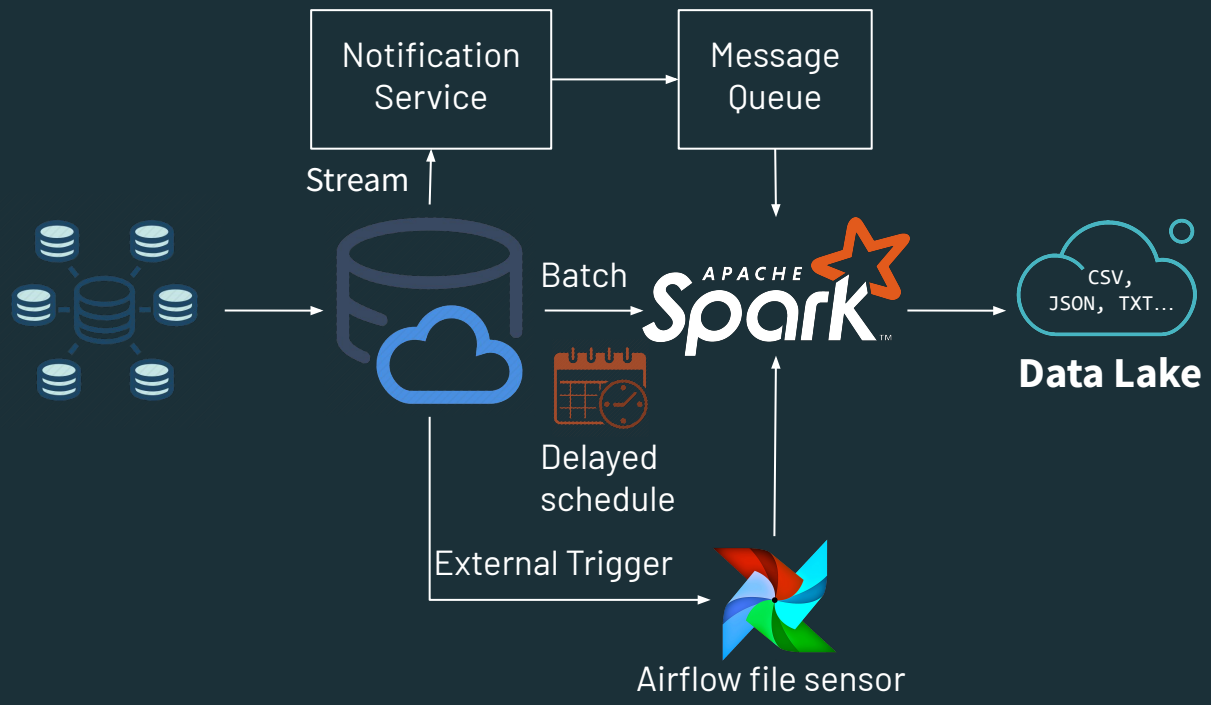
	Country	NumVaccinated	AvailableDoses	_change_type	_commit_version	_commit_timestamp
1	Australia	100	3000	insert	2	2021-04-12T20:48:05.000+0000
2	USA	10000	20000	update_preimage	3	2021-04-12T20:48:08.000+0000
3	USA	11000	20000	update_postimage	3	2021-04-12T20:48:08.000+0000
4	UK	7000	10000	delete	4	2021-04-12T20:48:11.000+0000

Showing all 4 rows.

Databricks Ingest: Auto Loader

Load new data easily and efficiently as it arrives in cloud storage

Before



Gets too complicated for multiple jobs

After

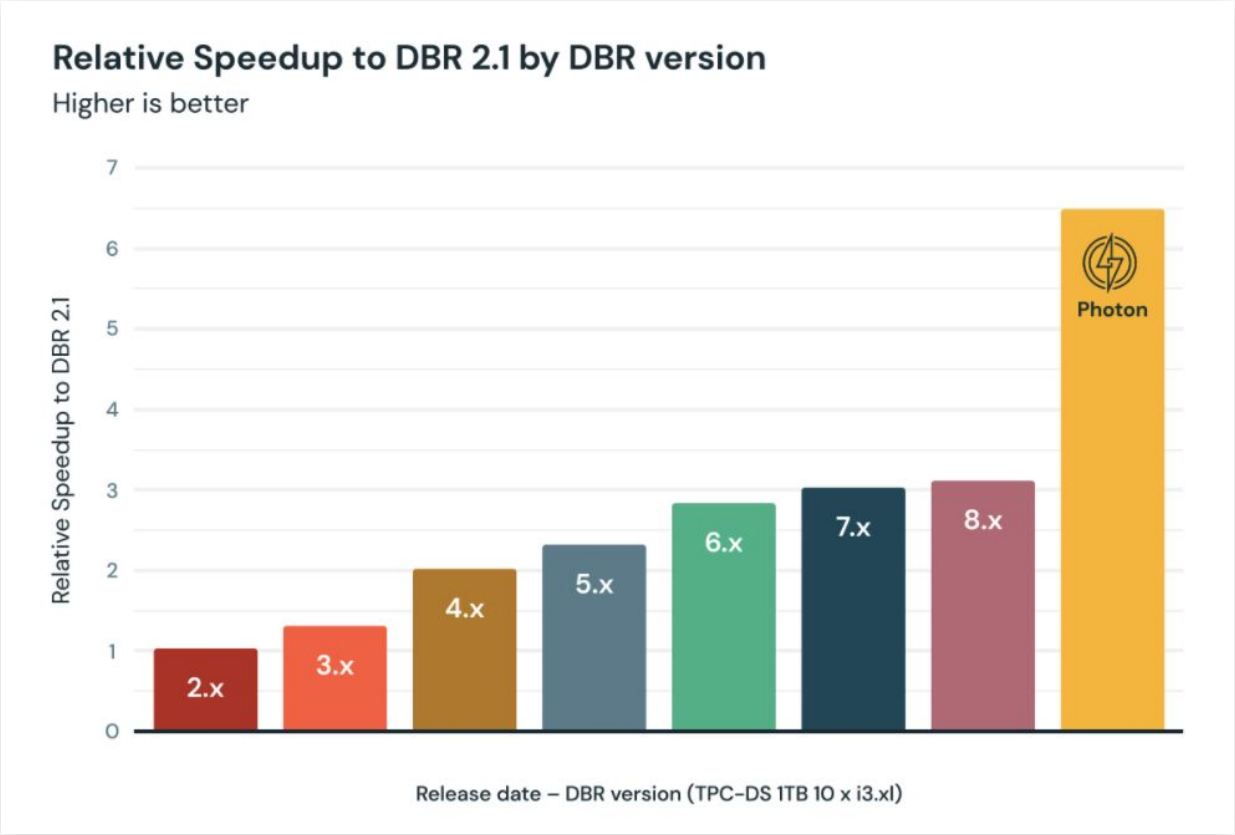
- Pipe data from cloud storage into your data lake with Delta Lake as it arrives
- "Set and forget" model eliminates complex setup

Launch Blog Post

Photon Performance

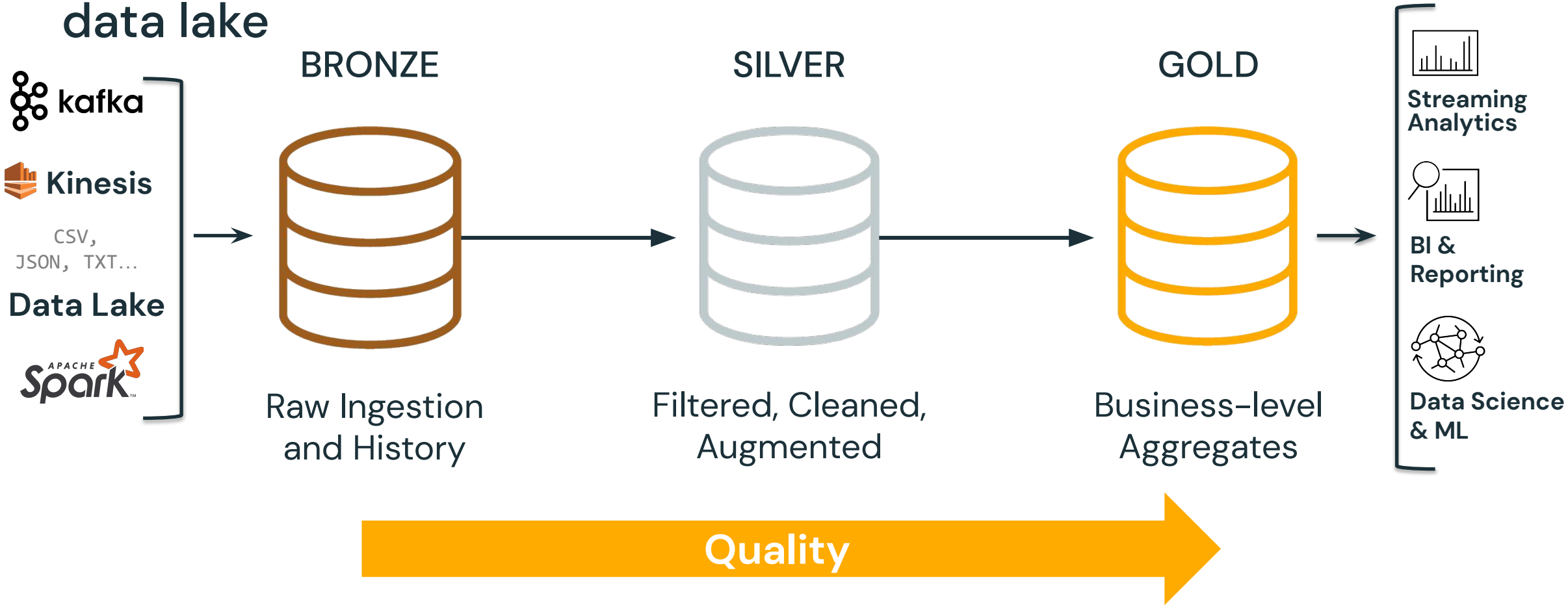
Supercharging Spark Performance – on Databricks

- Faster ETL and query performance
- Reduces cloud spend
- No code changes
- Broad language support



Building the foundation of the Lakehouse

Greatly improve data quality for end users – on your existing data lake



All your data together for all your data use cases

All your data

- CLICKSTREAM
- LOGS
- DATABASE
- SFDC
- IMAGES
- SENSORS
- STREAMING
- ...

Reliable data lake

DELTA LAKE™

Open format › Open APIs › Open source

Existing Data Lake

Azure Data Lake Storage

Amazon S3

hadoop

All your use cases

SQL Dashboards

redash

BI Reporting

looker

Data Science & ML

NOTEBOOK

jupyter

R

Data Pipelines

RUNTIME

APACHE Spark

Delta Lake Connectors

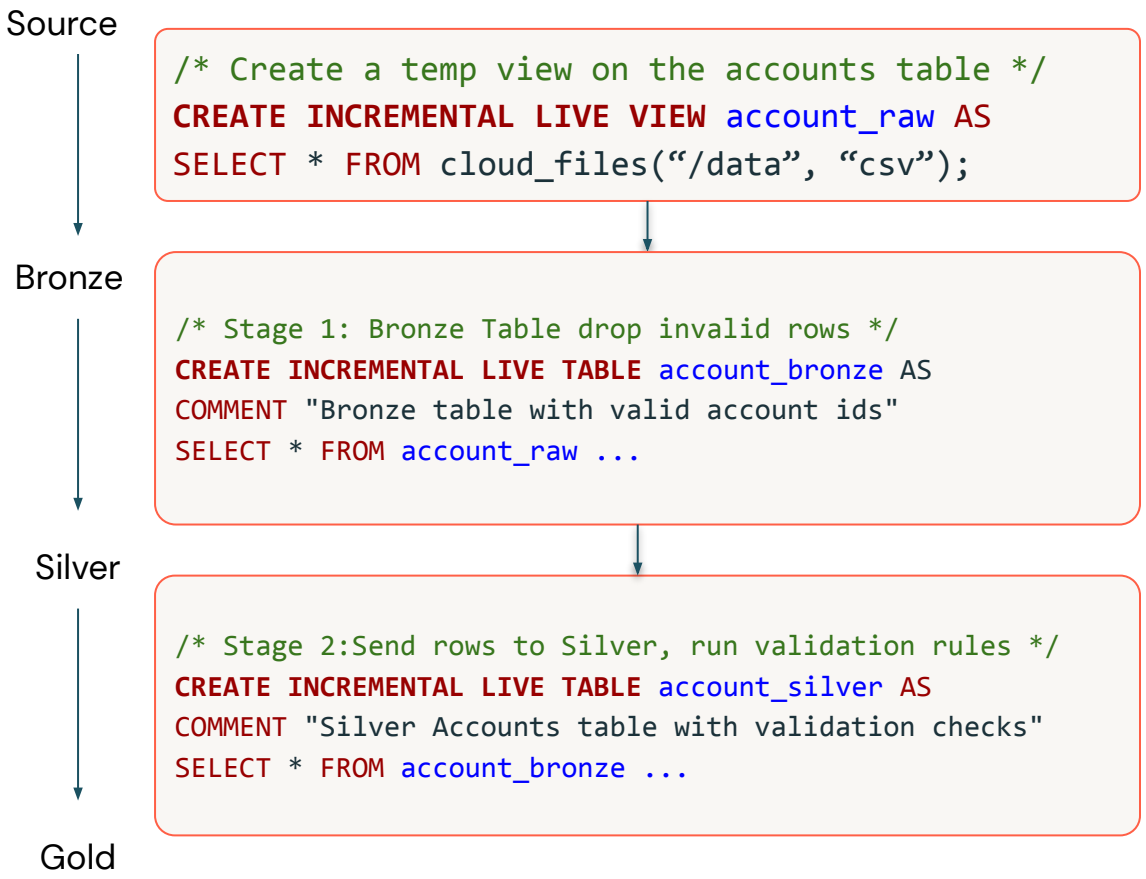
Standardize your big data storage with an open format accessible from various tools

databricks

Delta Live Tables

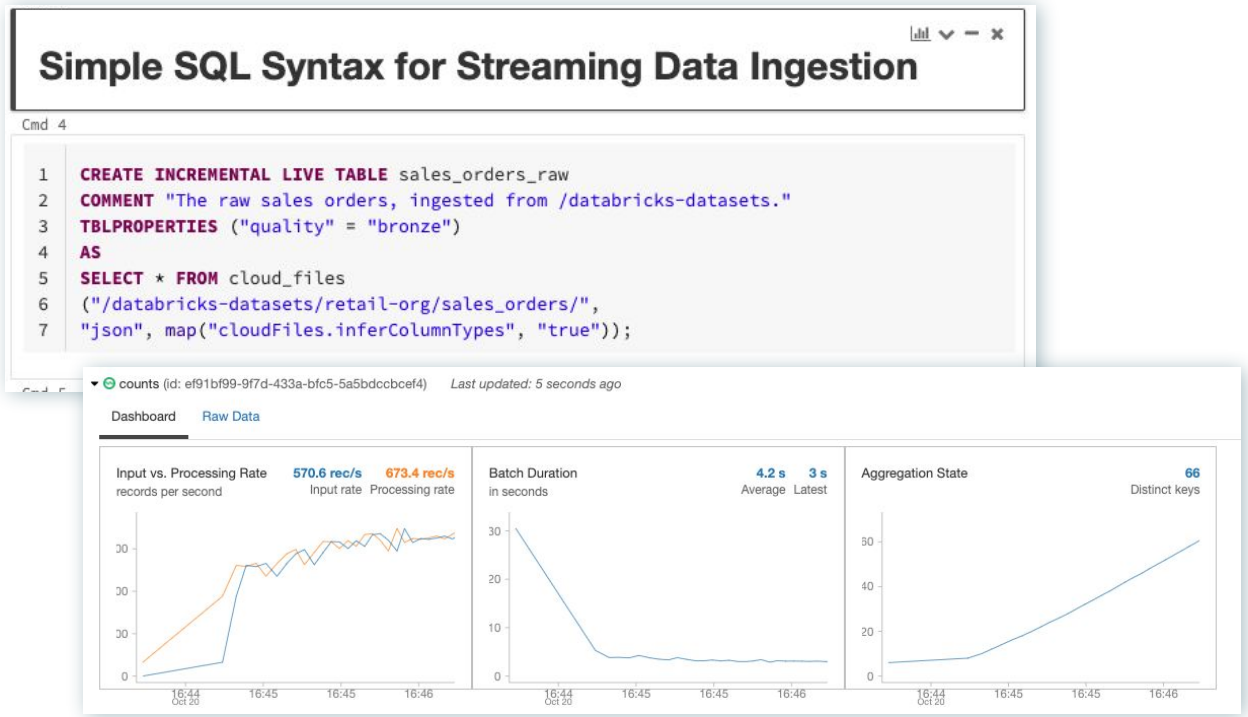
databricks

Declarative ETL pipelines with Delta Live Tables



- Use intent-driven declarative development to abstract away the **"how"** and define **"what"** to solve
- Automatically create high-quality lineage and manage table dependencies across the data pipeline
- Automatically checks for errors, missing dependencies and syntax errors, and manage pipeline recovery

Continuous or scheduled data ingestion with Auto Loader



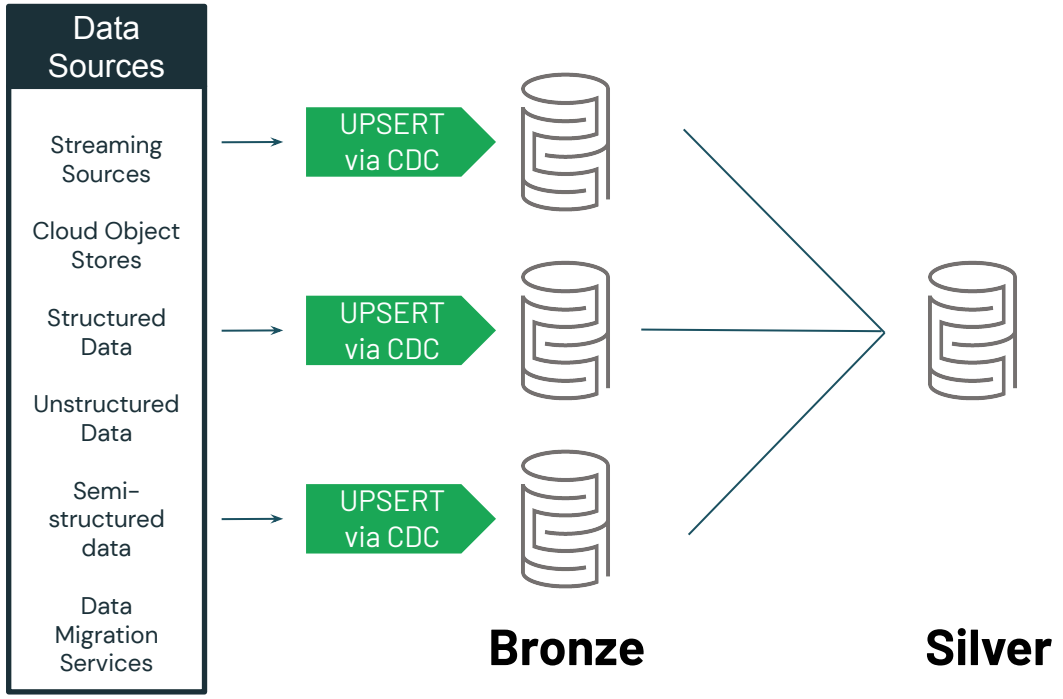
- **Incrementally and efficiently** process new data files as they arrive in cloud storage
- Automatically **infer schema** of incoming files or superimpose what you know with **Schema Hints**
- Automatic **schema evolution**
- **Rescue data column** - never lose data again

Schema Evolution

databricks

JSON	CSV	AVRO	PARQUET
✅	✅	Coming Soon	Coming Soon

Change data capture (CDC) with Delta Live Tables

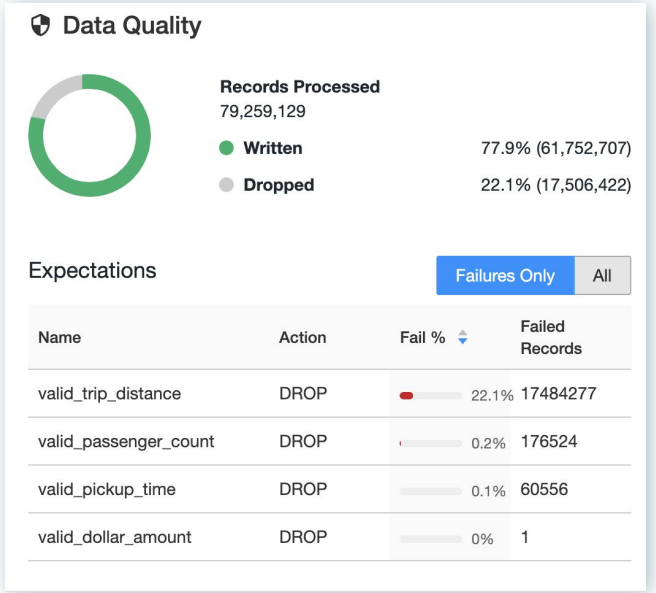


- Capture row-level changes from any data source supported by DBR, cloud storage, or DBFS
- Simpler architecture: build, simple incremental pipelines
- Handles out-of-order events
- Schema evolution
- Process change records (inserts, updates, deletes) incrementally using a simple, declarative "APPLY CHANGES INTO" SQL API

Data quality validation and monitoring with Delta Live Tables

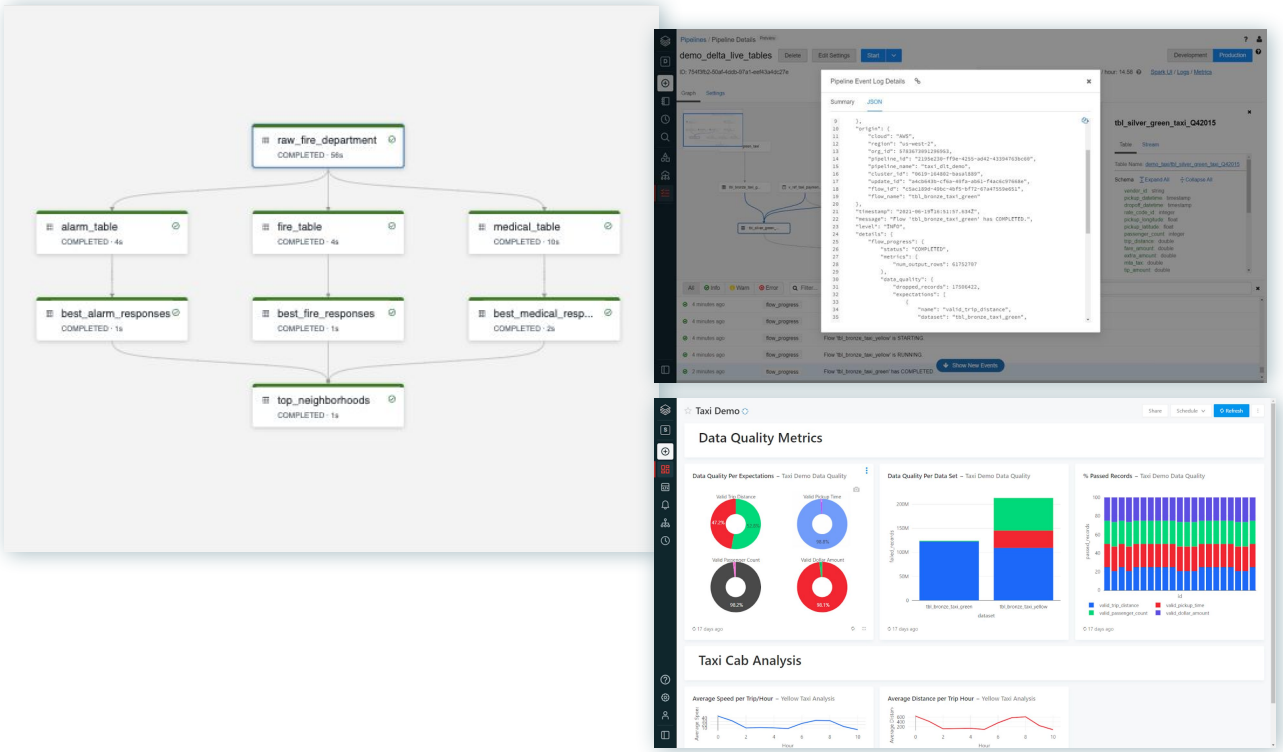
- Define data quality and integrity controls within the pipeline with data expectations
- Address data quality errors with flexible policies (fail, drop, alert, quarantine)
- All data pipeline runs and quality metrics are captured, tracked and reported

```
/* Stage 1: Bronze Table drop invalid rows */
CREATE INCREMENTAL 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 ...
```



databricks

Data pipeline observability in Delta Lives Table

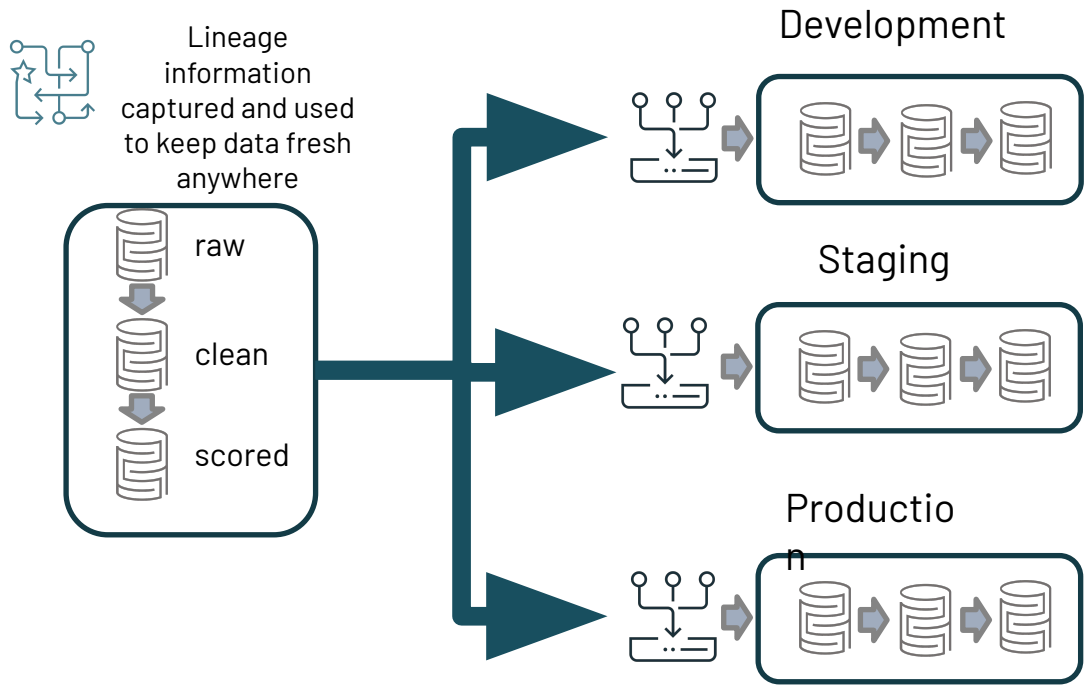


- High-quality, high-fidelity lineage diagram that provides visibility into how data flows for impact analysis
- Granular logging for **operational, governance, quality** and **status** of the data pipeline at a row level
- Continuously monitor data pipeline jobs to ensure continued operation
- Email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

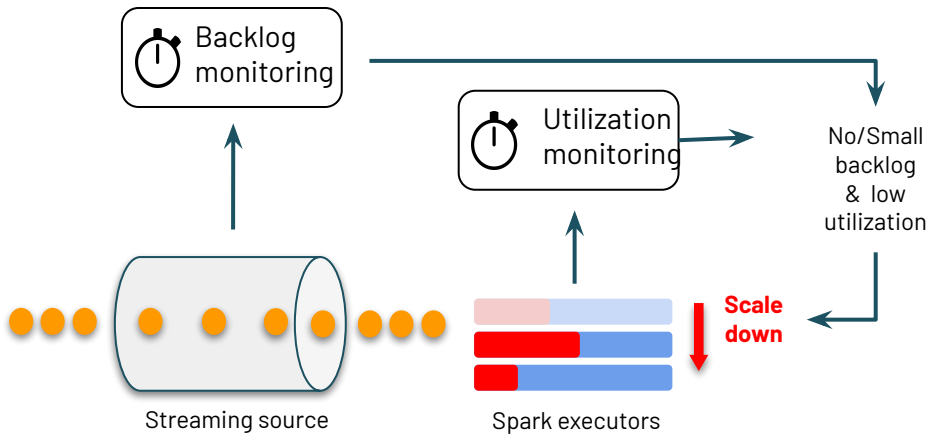
databricks

Automatic deployments and operations with Delta Lives Table

- Complete, parameterized and automated deployment for the continuous data delivery
- **Reuse ETL pipelines** across environments with config files and parameterization
- Orchestrates, tests, and monitor end-to-end the data pipeline



Automated scaling and fault tolerance with Delta Live Tables



- Meet streaming SLOs with backlog-aware scaling decisions - Monitor both, **backlog metrics and cluster utilization** to scale up or down
- **Reduce down time** with automatic error handling and easy replay
- **Eliminate maintenance** with automatic optimizations of all Delta Live Tables
- Execute data pipeline workload on **automatically provisioned** elastic Apache Spark™-based compute clusters that parallelize jobs as well as minimize data movement

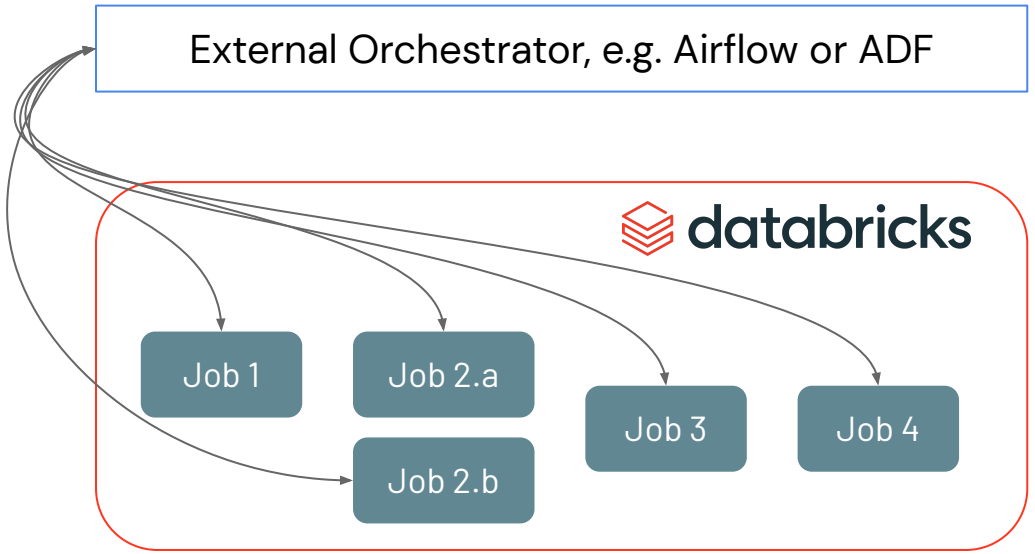
Multitask Jobs

databricks

Workflow Management on Databricks

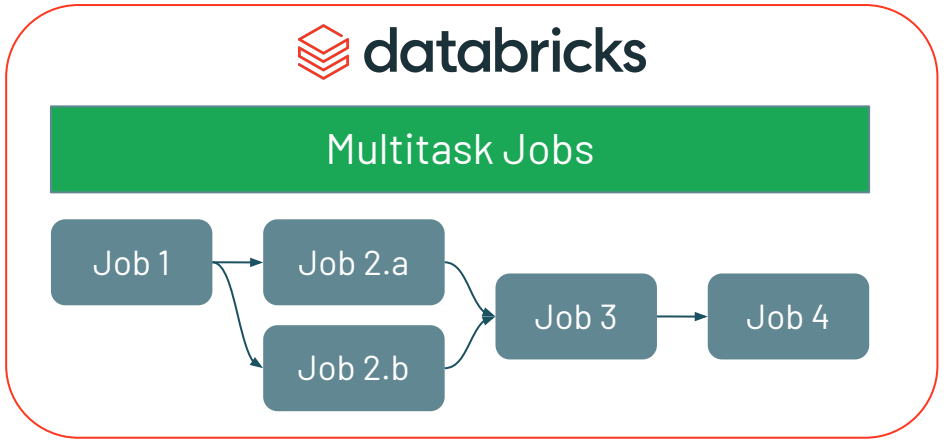
Simplify orchestration and management of multi-step workflows

Before



- Cost/complexity of maintaining external orchestrator
- Hard to monitor/debug

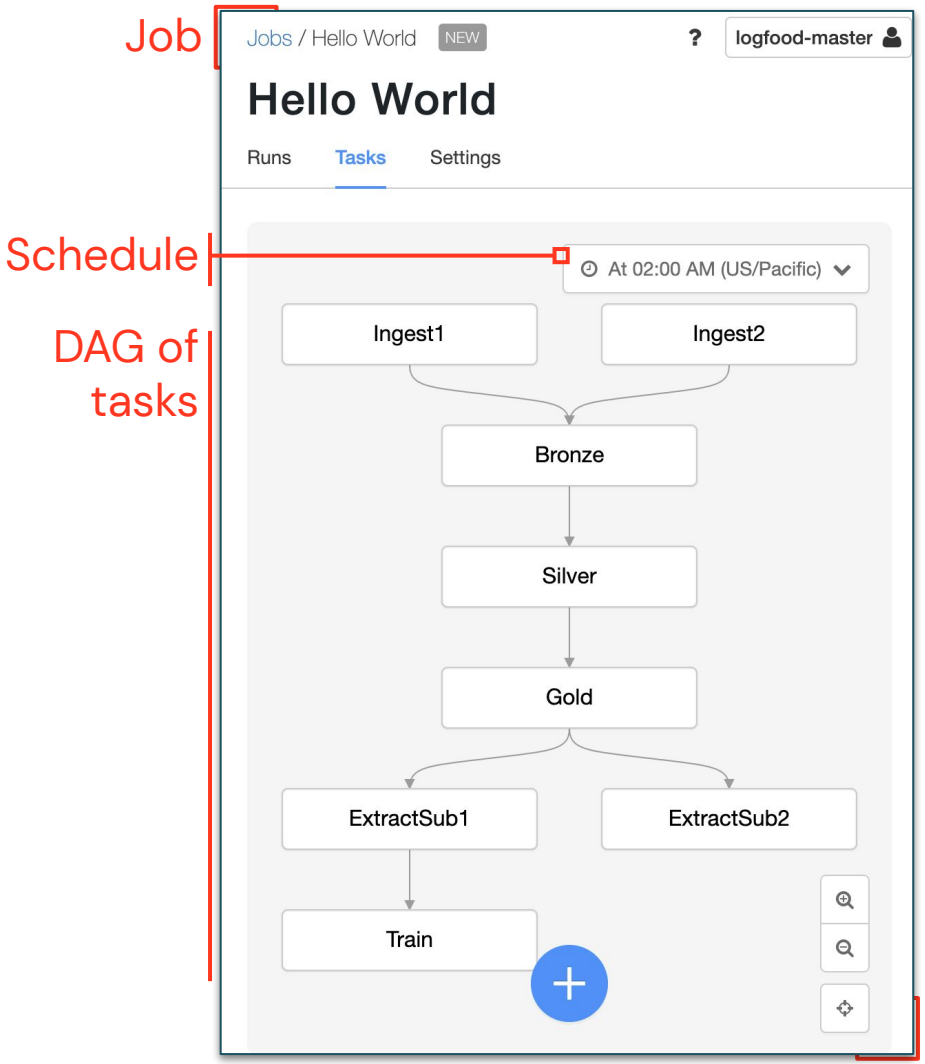
After



- Turnkey orchestration within Databricks
- Visibility into job dependencies, debugging, etc.
- Airflow and ADF integrations will continue to be supported

databricks

Easy orchestration for data and ML workloads



- What is it?
  Easy creation, scheduling, and orchestration of your code in DAGs (Directed Acyclic Graphs)
- Key features
  - **Simplicity:** No-code creation and monitoring in the UI
  - **Fully integrated** in Databricks platform, making inspecting results, debugging faster
  - **Orchestrate and manage workloads in multi-cloud environments**
  - **Reliability** of proven Databricks scheduler

databricks

Partner Connect

databricks

Databricks Partner Connect

Dedicated portal to easily integrate Databricks with Partner Solutions

Data Ingestion	Data Prep and Transformation	BI	Machine Learning
Fivetran	Prophecy	tableau	Labelbox
Rivery		Power BI	

Coming Soon

Azure Data Factory

dbt Labs

AIRBYTE

BLITZZ

databricks

Lab Environment Setup

databricks

Workspace Registration

Step 1: Register here using your work email address:
https://bit.ly/3A9DqgF

Step 2: Click the 'Submit' button (this will take 3-10 minutes to create)

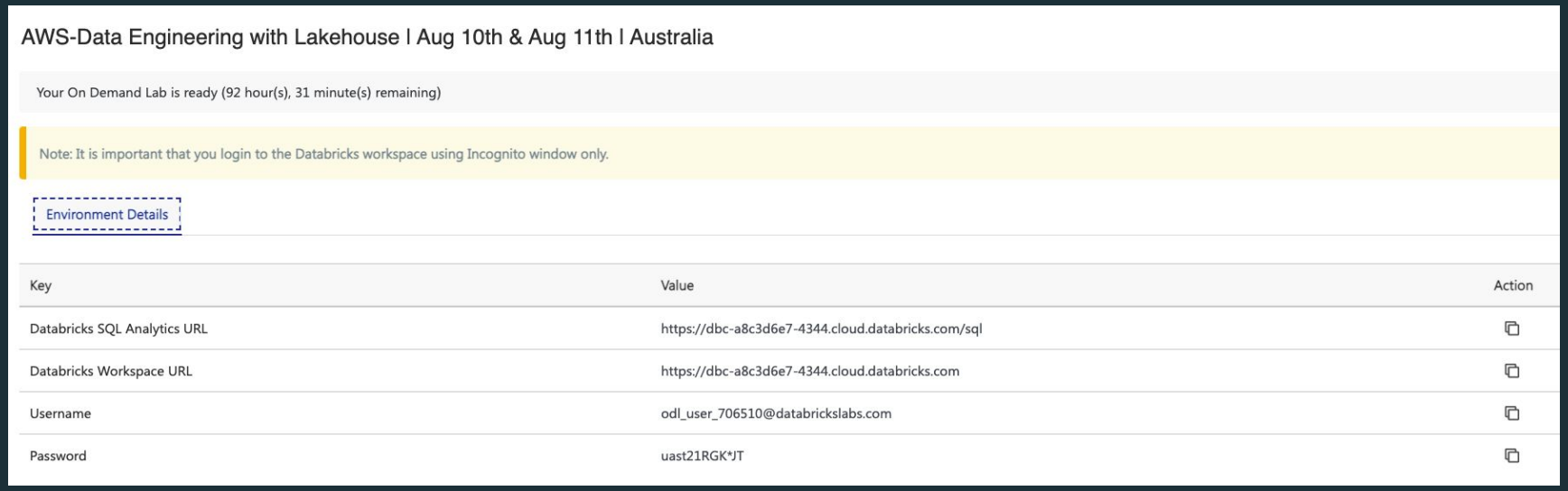
databricks

Workspace Registration

Step 1: Tap the 'Environment Details' tab

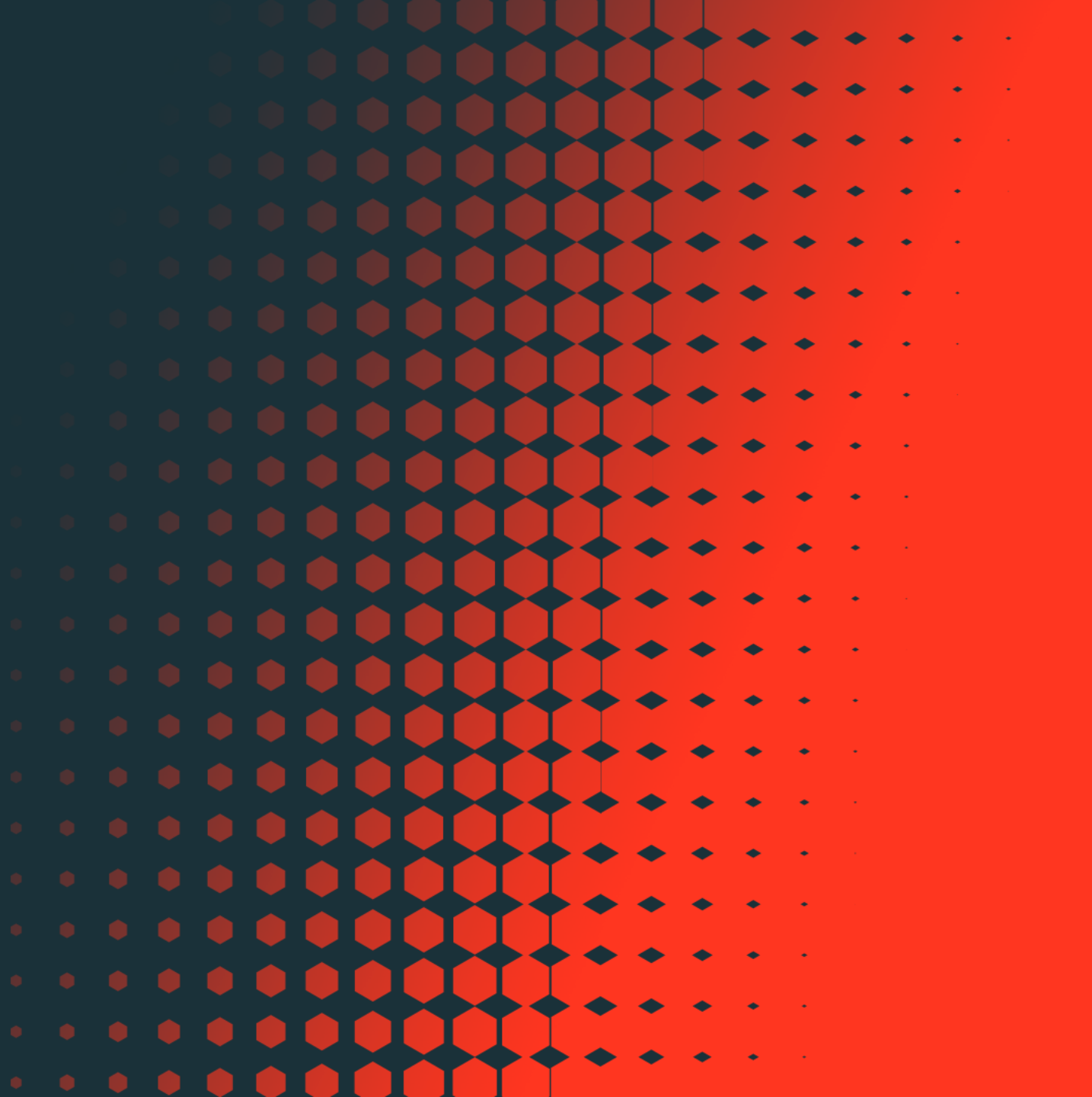
Step 2: Open a new Incognito window and paste the URL accordingly

Step 3: Copy and paste the credentials from the 'Environment Details' tab to log in to the Databricks workspace



https://bit.ly/3A9DqgF

Morning Tea



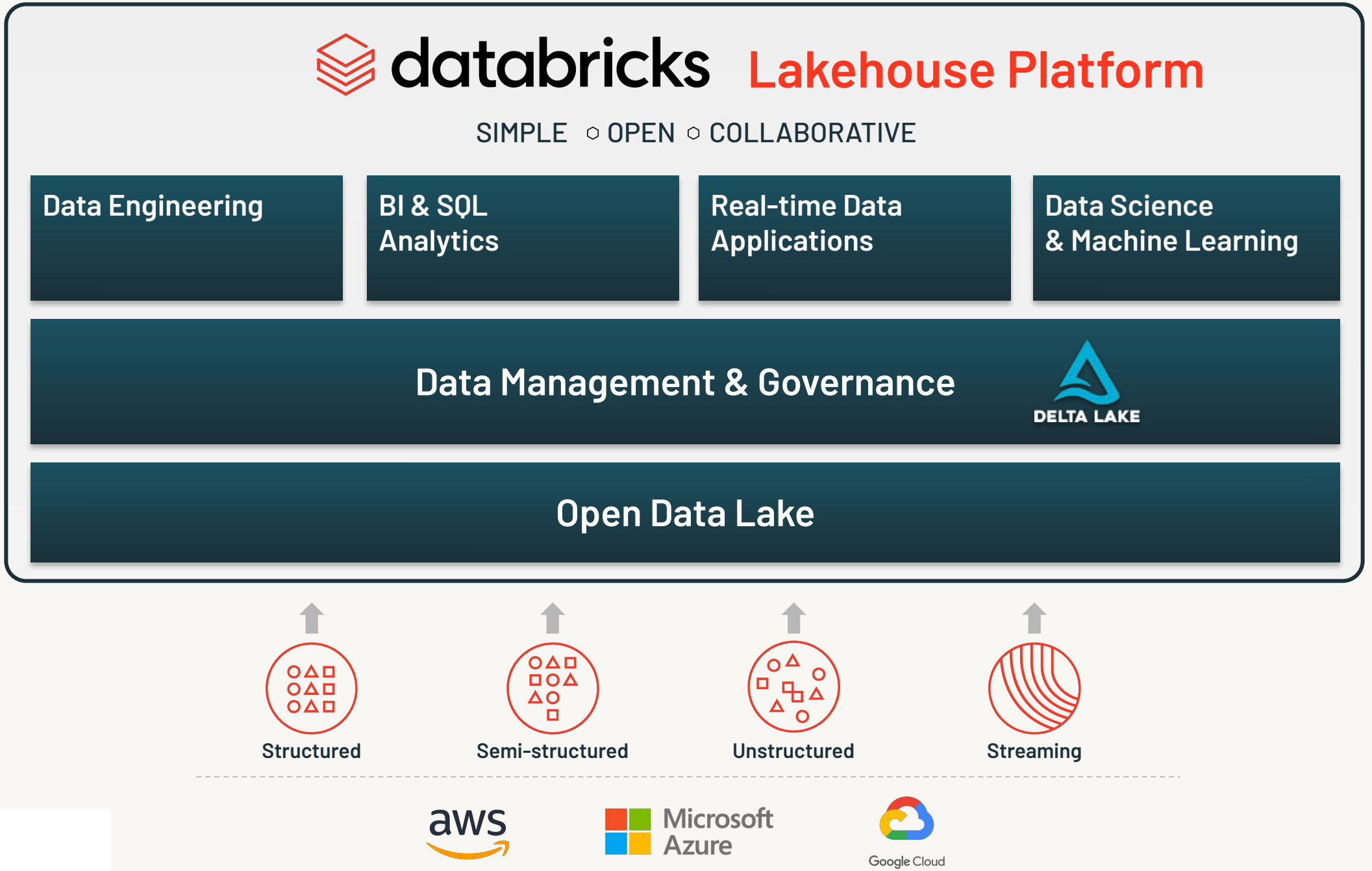
databricks

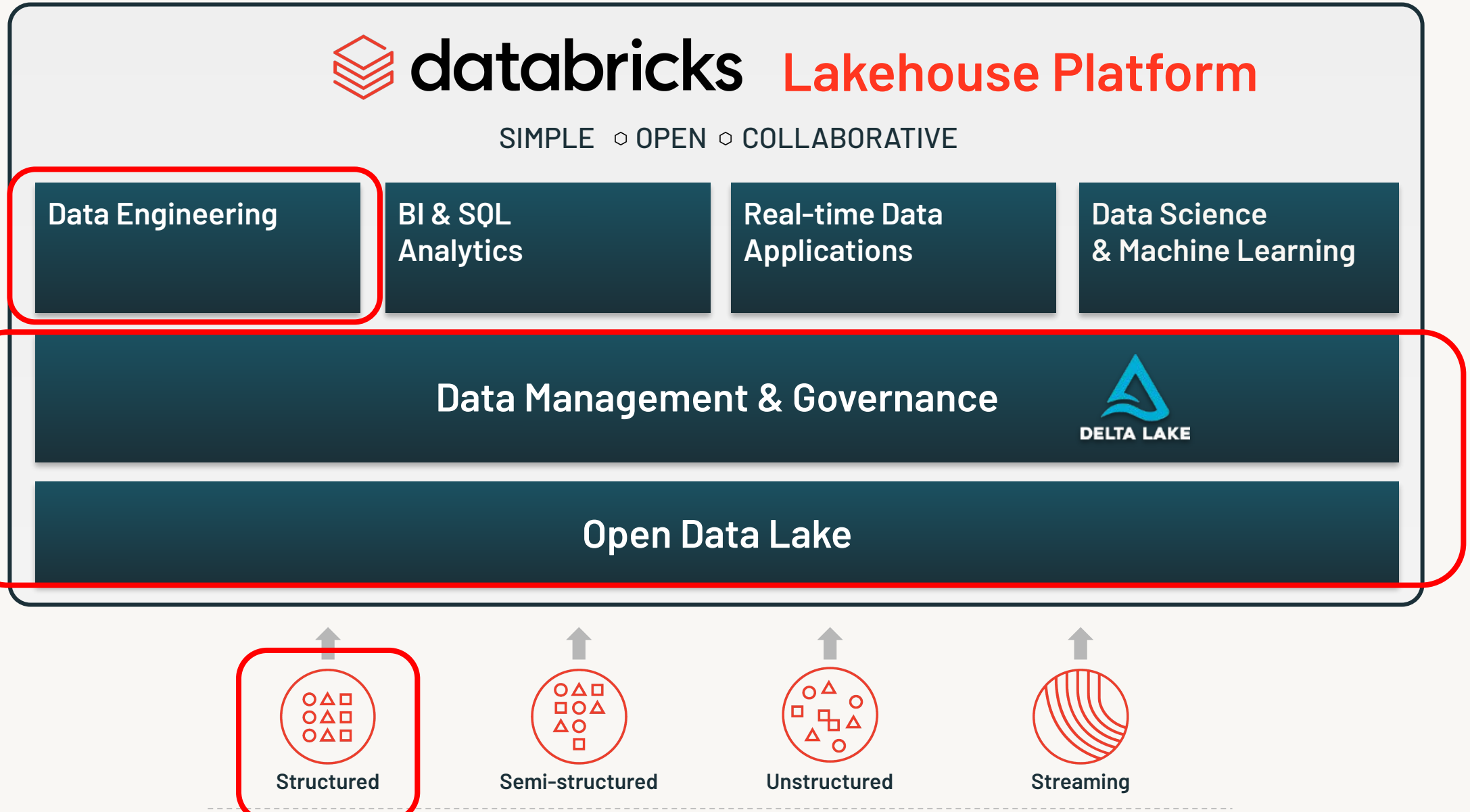
Data Engineering Lab Overview

databricks

Lab Context

- As part of the lab today, we will be working with APJuice data from different juice outlets across APJ
- Dataset consists of sales, stores, customer and product info
- The lab will focus on processing data using batch/ streaming in traditional and modern ways to help enable a Delta architecture and serve business as well as data analysts and data scientists in the organisation


databricks
Lakehouse Platform
SIMPLE ⬡ OPEN ⬡ COLLABORATIVE
Data Engineering
BI & SQL Analytics
Real-time Data Applications
Data Science & Machine Learning
Data Management & Governance
DELTA LAKE
Open Data Lake
Structured
Semi-structured
Unstructured
Streaming
aws
Microsoft Azure
Google Cloud


databricks
Lakehouse Platform
SIMPLE ⬡ OPEN ⬡ COLLABORATIVE
Data Engineering
BI & SQL Analytics
Real-time Data Applications
Data Science & Machine Learning
Data Management & Governance
DELTA LAKE
Open Data Lake
Structured
Semi-structured
Unstructured
Streaming
aws
Microsoft Azure
Google Cloud

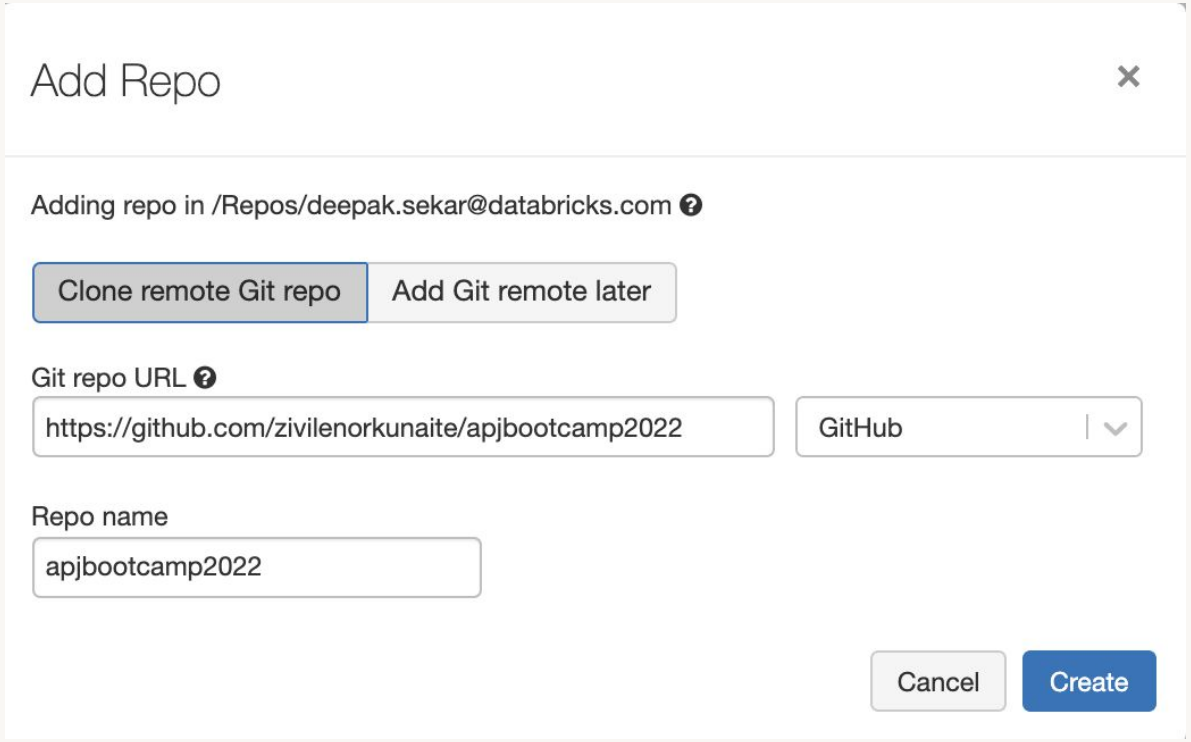
https://bit.ly/3NQiFLS

https://vnjv.co/boot22

Load The Notebooks

- Use the Repos feature to load the Databricks Notebooks for the labs today. Select Repos, Add Repo and enter the following as the *Git repo URL*: https://github.com/zivilenorkunaite/apjbootcamp2022

https://bit.ly/3NQiFLS



databricks

Data Engineering Lab

databricks

Lunch & Feedback

databricks