DATA+AI SUMMIT BY databricks

INCREMENTAL INGESTION

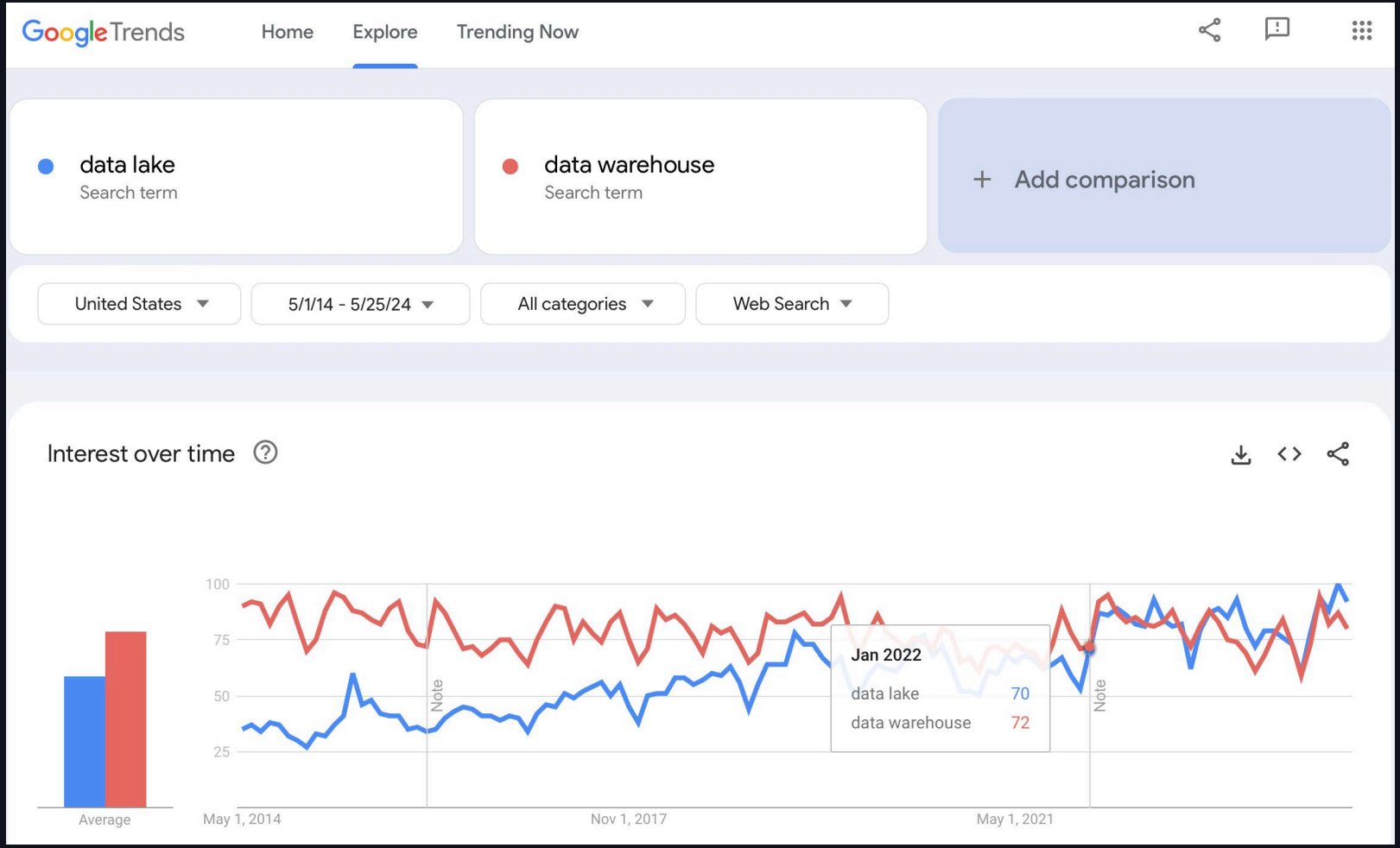
A Data Informed Journey

Christina Taylor
06/24

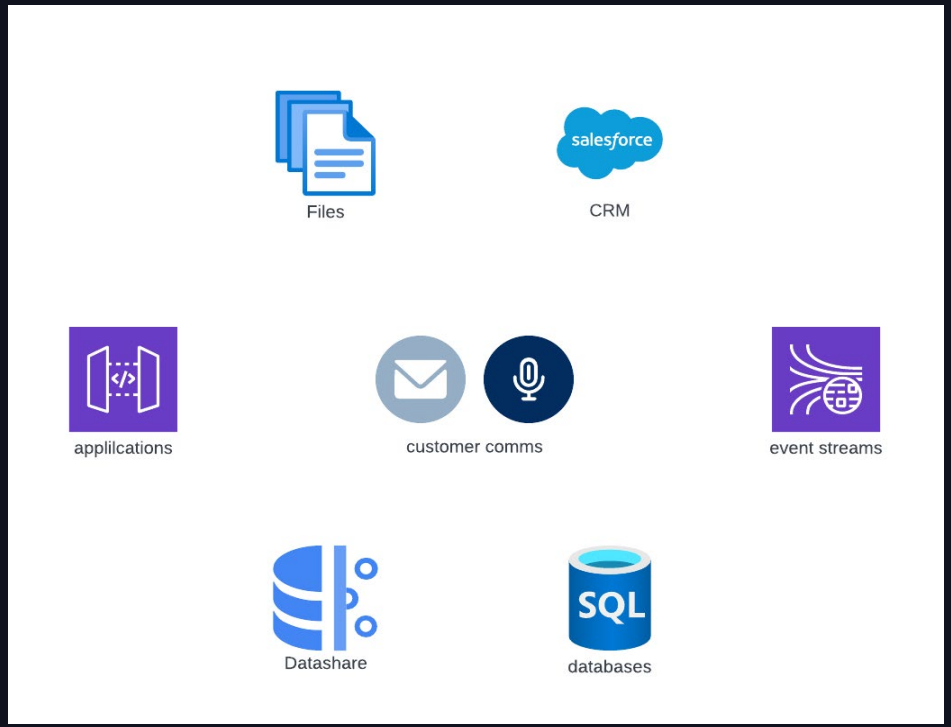
AGENDA

- Motivation
- CDC: The Necessary Evil?
- Alternatives
- Conclusion

MOTIVATION

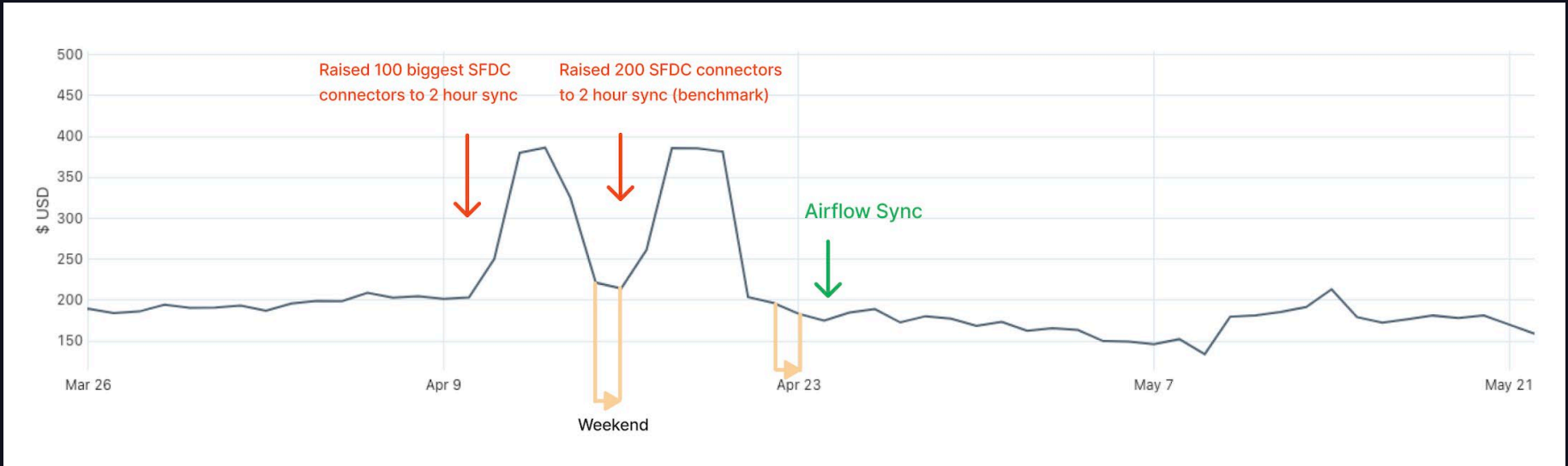


THE SINGLE SOURCE OF TRUTH



FALL IN LOVE WITH DATA

Insight from Overwatch



KNOWLEDGE IS POWER

All wisdom came from suffering

Be mindful of Default

- It is possible to be more efficient. Doing so often requires more infra setup.

Autoscaling vs Serverless

- Observe target warehouse scaling behavior. Compare with serverless offerings.

Invest in Observability

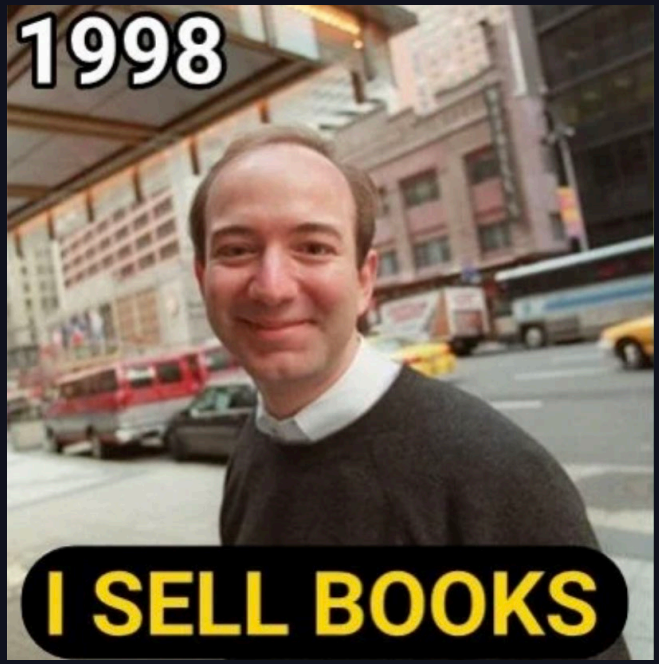
- Use system tables and billing/log delivery. This effort pays dividends long term.

NECESSARY EVIL?

CHANGE
DATA
CAPTURE

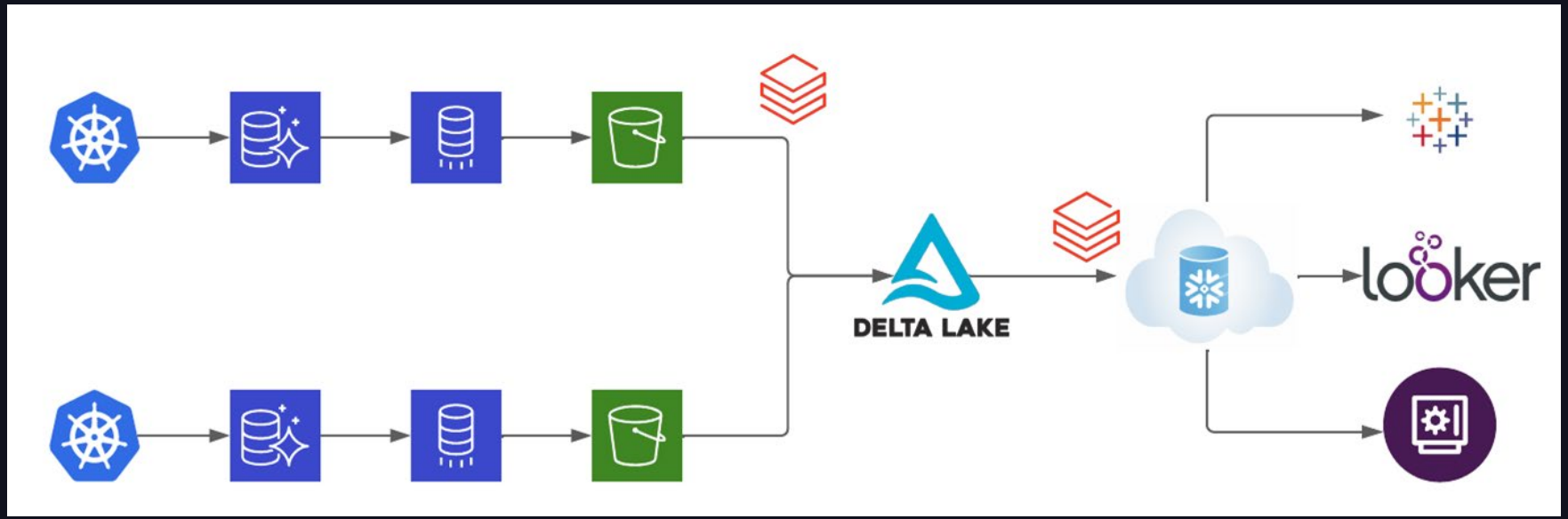
In the beginning,

I wanted to move mountains...



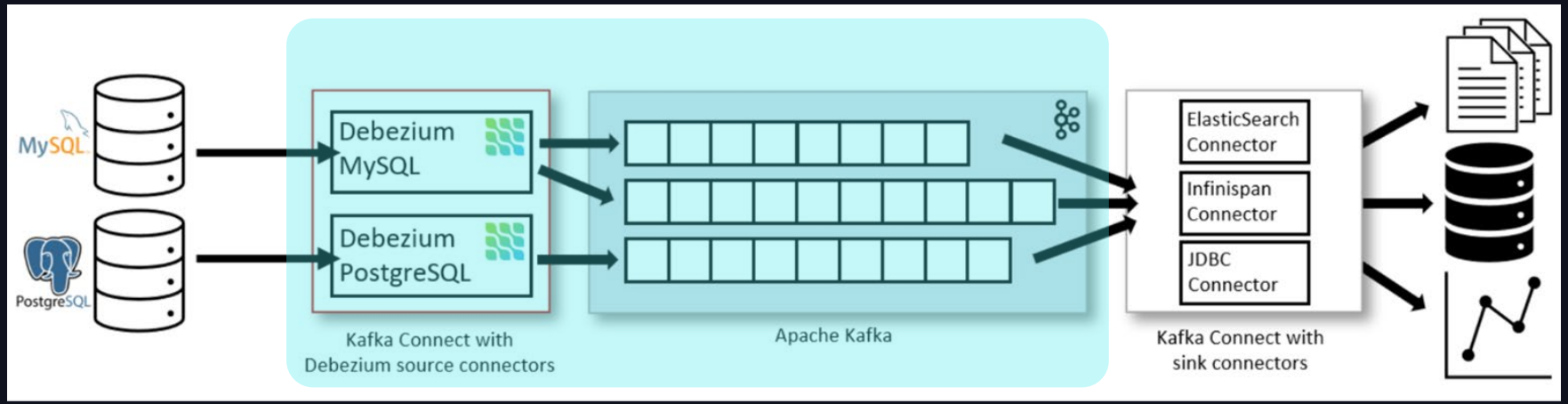
VERSION 0

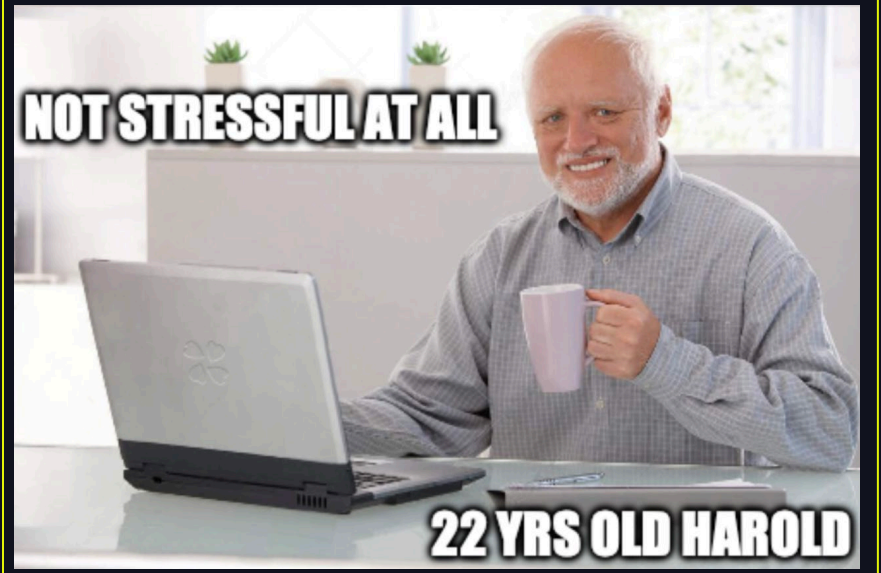
Database Migration Service + Autoloader



VERSION 1

Debezium + Delta Live Tables





You all suck.

I have spoken.

THE ESCAPADE?

CHRISTINA
DRINKS
CANS

STREAMING CDC ALTERNATIVES

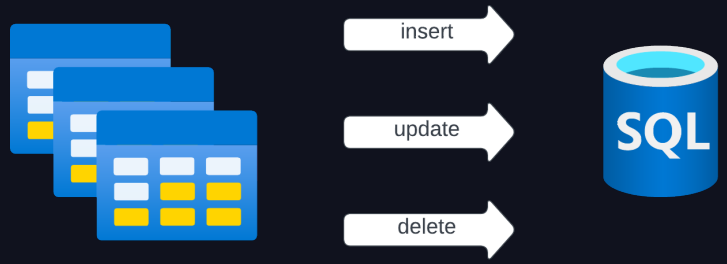
CDC != Log Based CDC

Query (Poll)

Use simple queries and a timestamp

Trigger (Push)

Execute triggers in response to DDL/DML



If the mountain won’t come to Mohammad,

Mohammad must go to the mountain.

VERSION 2

Spark Connector / Lakehouse Query Federation

JDBC Connector

```
# Control db connections and degrees of parallelism

spark.read
    .format("jdbc")
    .option("url", "jdbc:postgresql:dbserver")
    .option("fetchsize", 1000)
    .option("dbtable", f"""(select id, data, updated_at
             from public.my_table
             where updated_at >= '{start_dt}'
             and updated_at < '{end_dt}') as dbtable""")
    .options({
             "numPartitions": 16,
             "partitionColumn": "updated_at",
             "lowerBound": start_dt,
             "upperBound": end_dt,})
    .load()
```

Query Federation

```
# 1. Create a connection
# 2. Create a foreign catalog
# 3. Read data from Unity Catalog
# https://docs.databricks.com/en/query-federation/index.html

spark.table("foreign_pg_catalog.public.my_table")
```

SUPERCHARGE YOUR WORKERS

Liquid Clustering

	Count 1	Count 2
Partition A	□□□	□□□□□□
Partition B	□□□□□□□□□ □□□□□□□□□	□□
Partition C	□□	□□□□□□□□□ □□□□□□□□□ □□□□□□□□□

	Count 1	Count 2
Cluster A	□□□□□□	□□□□□□□□□ □□
Cluster B	□□□□□□	□□□□□□□□□ □□□
Cluster C	□□□□□	□□□□□□□□□ □□□

CLOSING REMARKS

Make everything as simple as possible, but not simpler.