DATA+AI
SUMMIT 2022

Hassle-free data ingestion into the Lakehouse

Make data ingestion simpler

Burak Yavuz
Engineering Manager, Databricks

Benyue (Emma) Liu
Staff Product Manager, Databricks

ORGANIZED BY databricks

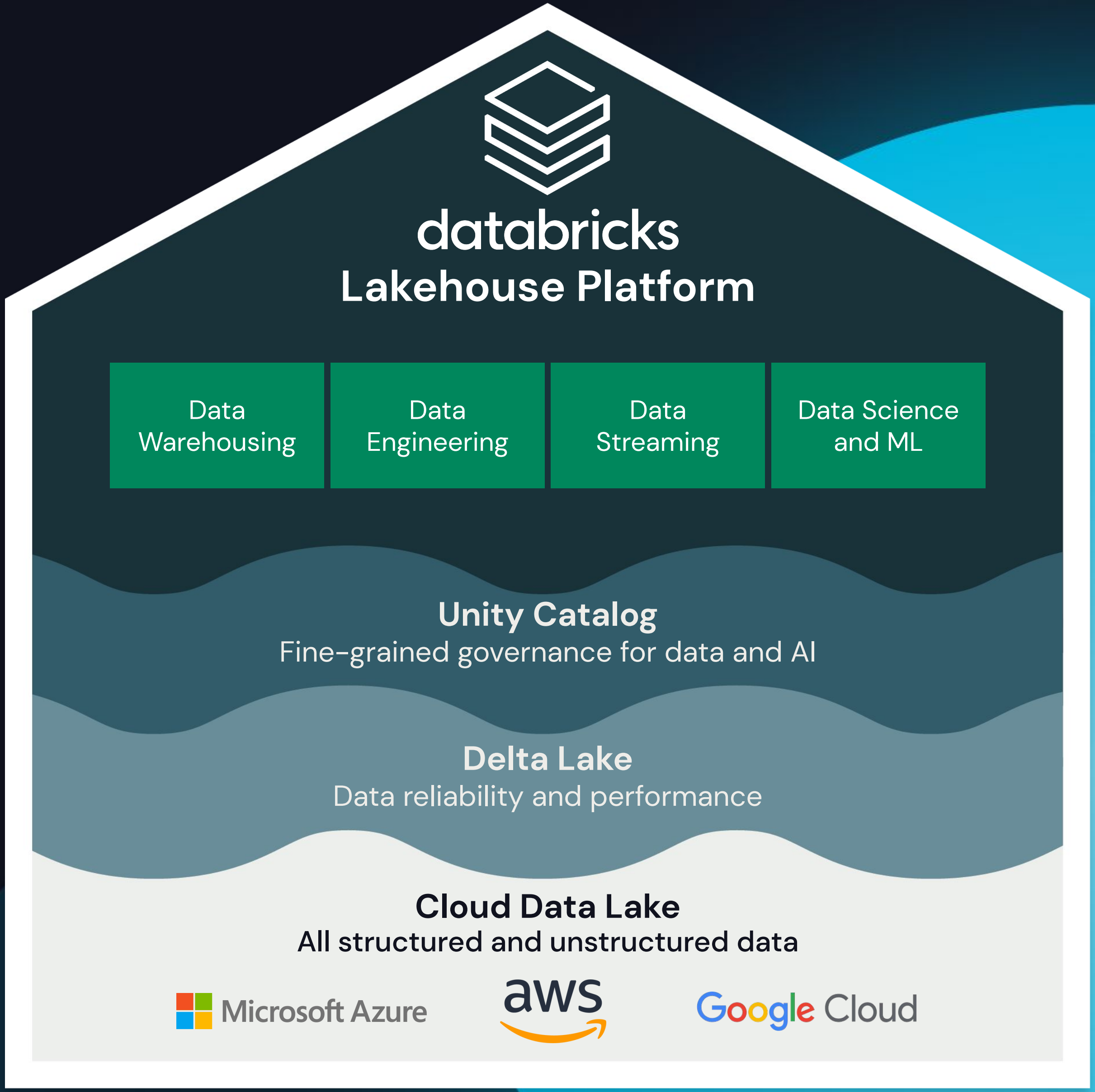
Product safe harbor statement

Agenda

1. Databricks Lakehouse overview
2. Ingestion challenges
3. Get started with ingestion in less than 10 minutes
4. Demo

Databricks Lakehouse overview

Your destination is the Lakehouse



Ingestion challenges

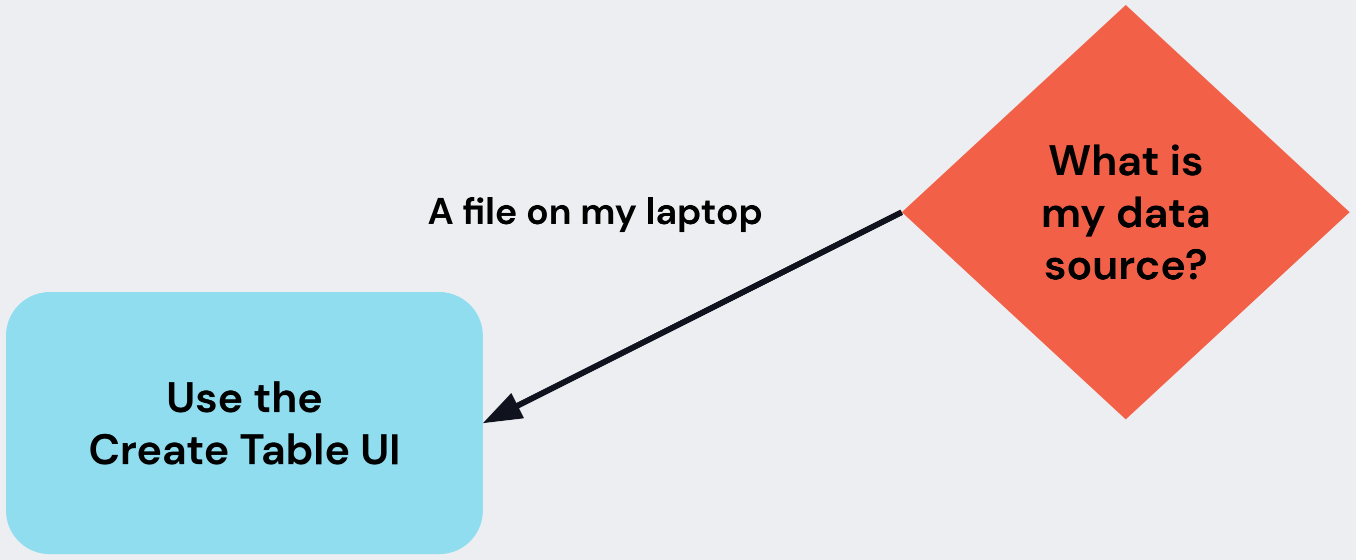
Data ingestion presents challenges

- Too many data sources
- Figuring out what files to process
- Schema changes over time
- Fixing issues with bad data
- Scalability
- Migrating existing tables
- Different requirements on data freshness
- Cloud configurations are complex
- Change-data-capture from OLTP databases
- Enable both ad hoc ingestion and production data pipeline

Cloud Data Lake

Structured, semi-structured, and unstructured data

Get started with Lakehouse ingestion in less than 10 minutes!


What is my data source?
A file on my laptop
Use the Create Table UI

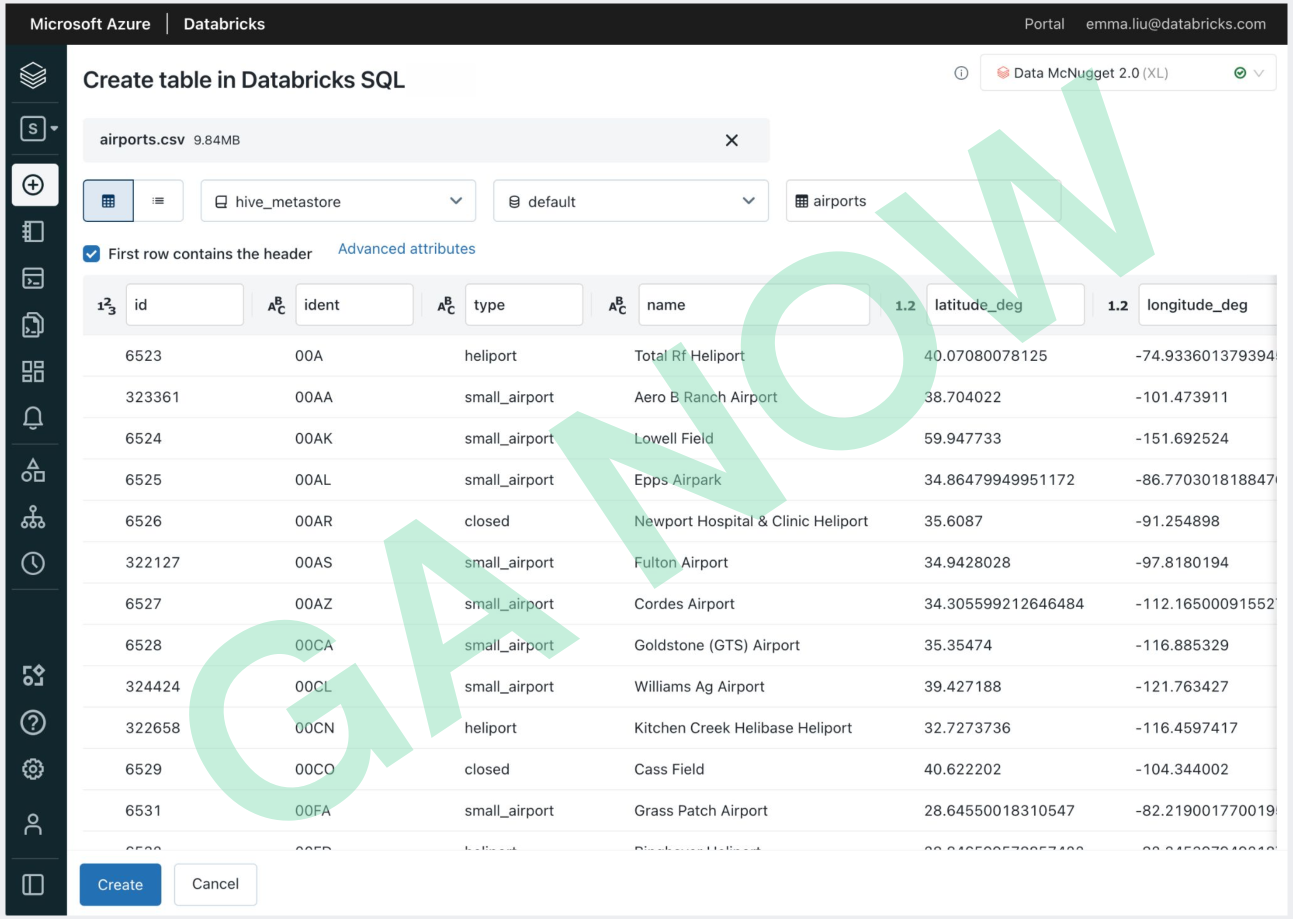
New ingestion features in Databricks SQL

Create table from local files or cloud object stores

Quickly upload a local CSV & create a Delta table from it from DBSQL UI

What's coming next?

- Support for additional file types (JSON)
- UI to ingest from cloud storage like S3
- Unified Data Source UI to the Lakehouse
- SQL API to ingest from cloud storage



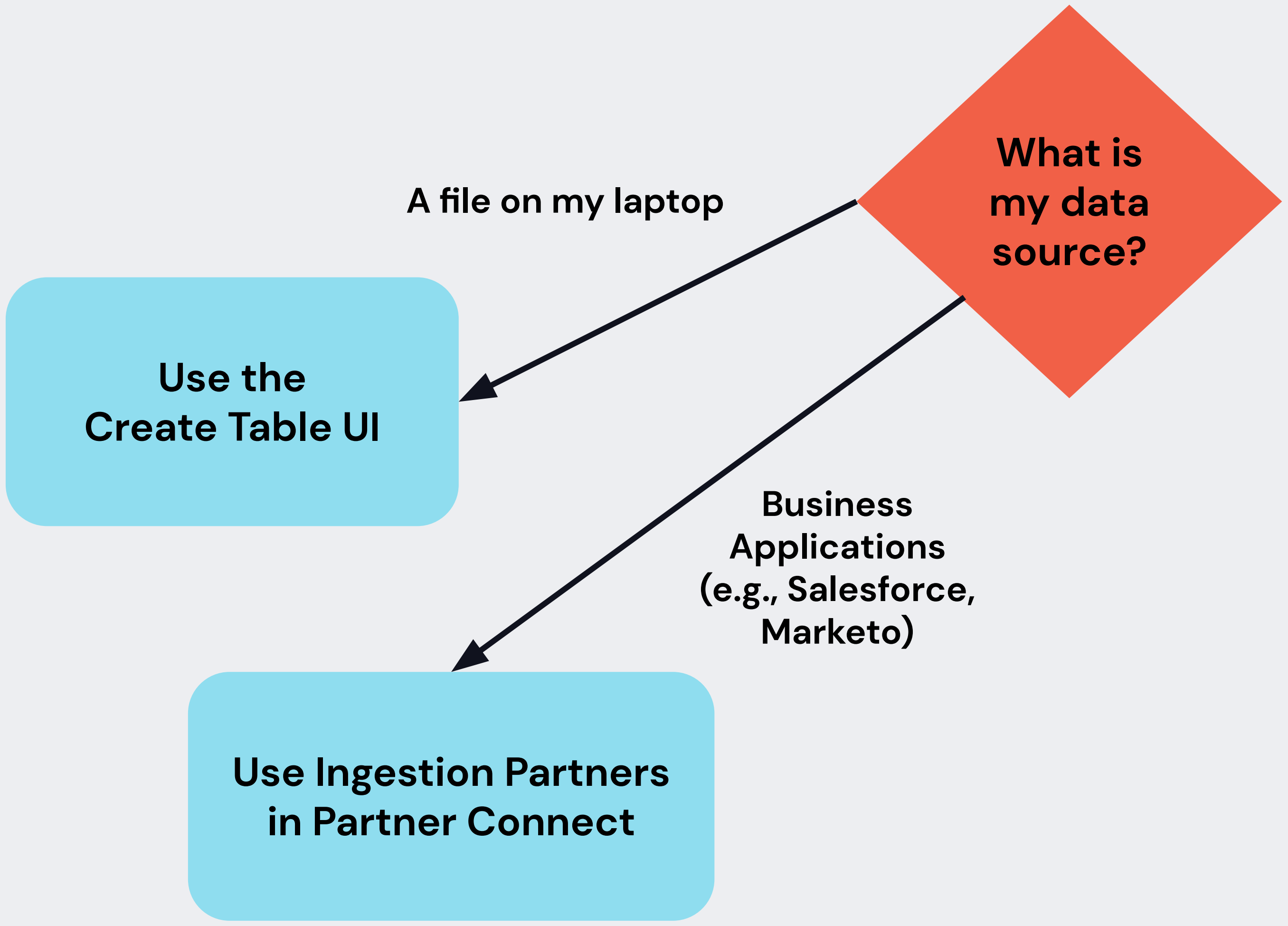
Create table in Databricks SQL

airports.csv 9.84MB

hive_metastore | default | airports

☑ First row contains the header   Advanced attributes

id	ident	type	name	latitude_deg	longitude_deg
6523	00A	heliport	Total Rf Heliport	40.07080078125	-74.933601379394
323361	00AA	small_airport	Aero B Ranch Airport	38.704022	-101.473911
6524	00AK	small_airport	Lowell Field	59.947733	-151.692524
6525	00AL	small_airport	Epps Airpark	34.86479949951172	-86.770301818847
6526	00AR	closed	Newport Hospital & Clinic Heliport	35.6087	-91.254898
322127	00AS	small_airport	Fulton Airport	34.9428028	-97.8180194
6527	00AZ	small_airport	Cordes Airport	34.305599212646484	-112.16500091552
6528	00CA	small_airport	Goldstone (GTS) Airport	35.35474	-116.885329
324424	00CL	small_airport	Williams Ag Airport	39.427188	-121.763427
322658	00CN	heliport	Kitchen Creek Helibase Heliport	32.7273736	-116.4597417
6529	00CO	closed	Cass Field	40.622202	-104.344002
6531	00FA	small_airport	Grass Patch Airport	28.64550018310547	-82.219001770019
[illegible]	[illegible]	heliport	[illegible]	[illegible]	[illegible]


What is my data source?
A file on my laptop
Use the Create Table UI
Business Applications (e.g., Salesforce, Marketo)
Use Ingestion Partners in Partner Connect

Databricks Partner Connect

Data Ingestion

Fivetran, Rivery, MATILLION, CONFLUENT, Airbyte, arcion, mongoDB.

Data Pipelines

Prophecy, dbt, hightouch, MATILLION, Informatica

Data Governance

Collibra, IMMUTA, Anomalo, lightup, PRIVACERA, Alation

BI & Dashboards

Looker, tableau, ThoughtSpot, Power BI, preset, MODE

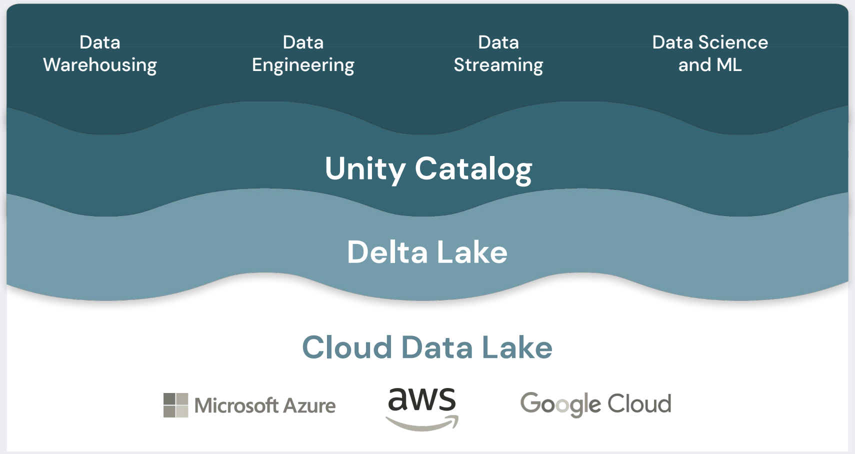
Data Science

PyCharm, jupyter, RStudio, HEX, plotly

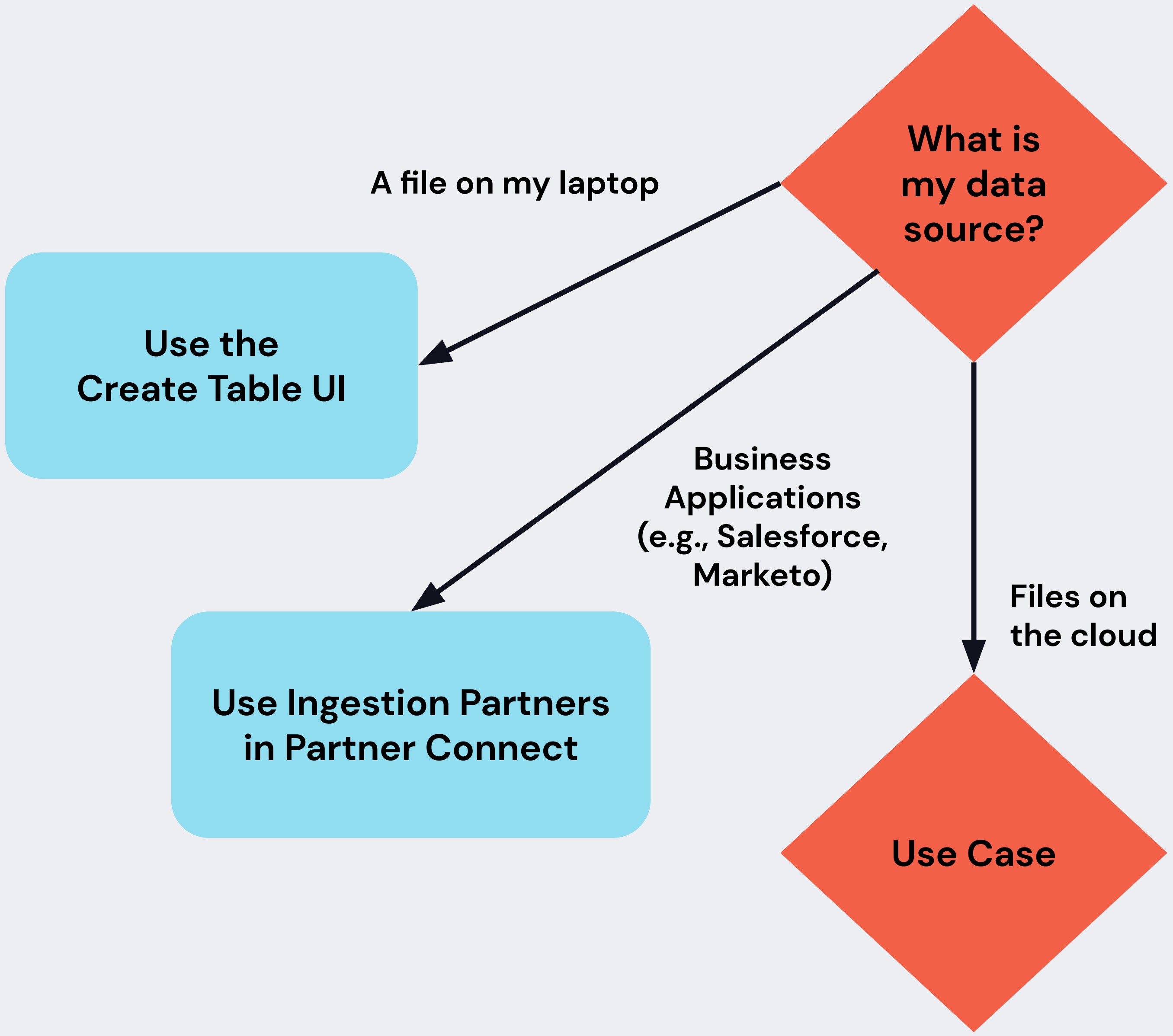
Machine Learning

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

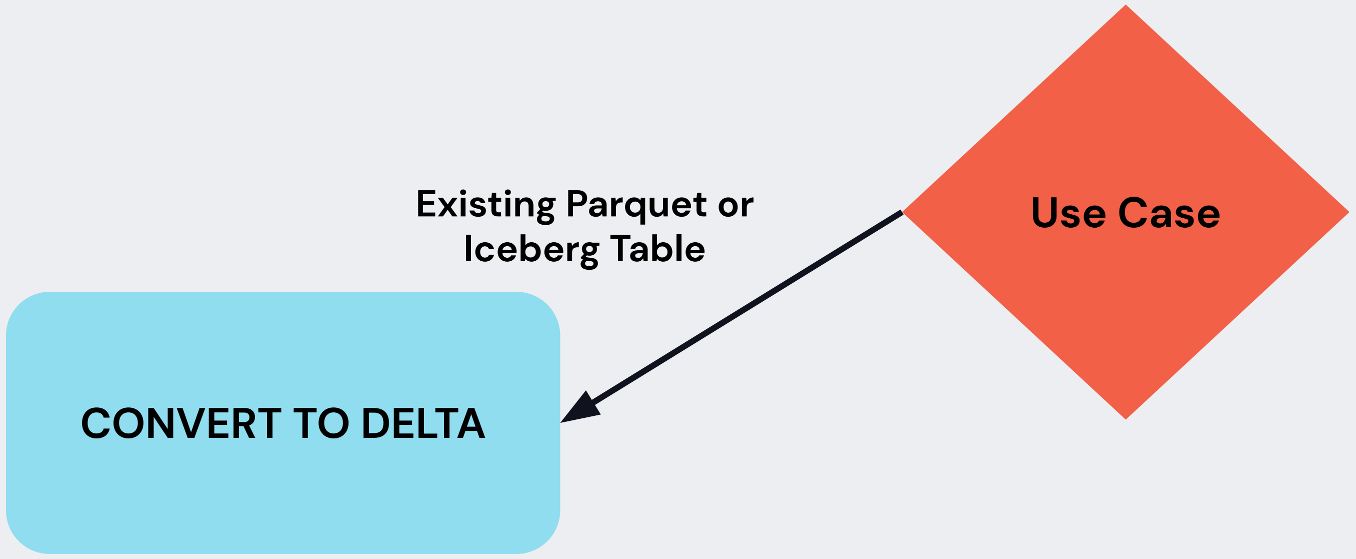
Partner Connect



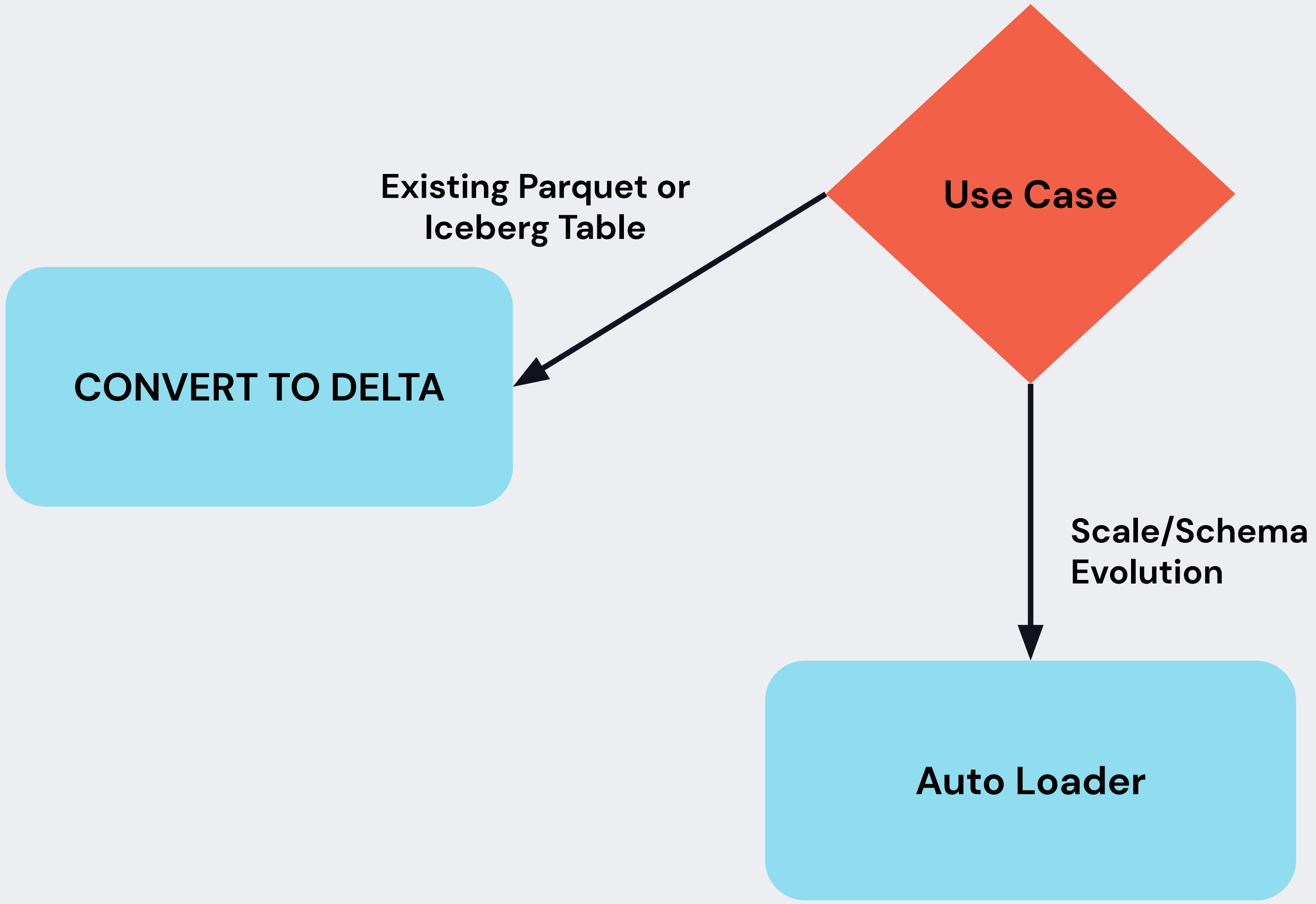
Easily discover and connect data, analytics, and AI tools to your lakehouse


What is my data source?
A file on my laptop
Use the Create Table UI
Business Applications (e.g., Salesforce, Marketo)
Use Ingestion Partners in Partner Connect
Files on the cloud
Use Case

Ingesting files on the Cloud



Ingesting files on the Cloud



Auto Loader

- Available in Python & Scala (and SQL in Delta Live Tables!)
- Incremental loading
- Exactly once ingestion
- Scalable for large amounts of data
- Designed for structured, semi-structured and unstructured data

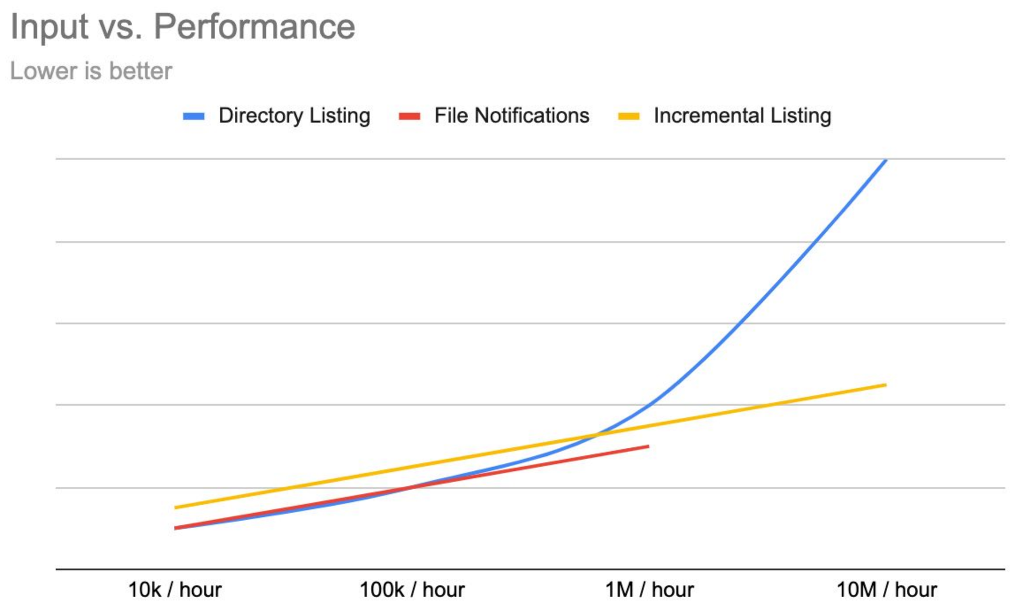
```
df = spark.readStream.format("cloudFiles")
  .option("cloudFiles.format", "json")
  .load("/path/to/table")
```

“We ingest more than **5 petabytes per day** with Auto Loader”

Auto Loader performance— File Notifications & Directory Listing

Comparison

Relative Performance



Why two modes?

- Directory Listing is **simple** to set up and works well for small throughputs
  - Optimized Cloud APIs reduce RPC costs
- *Incremental Listing* is a **special case** of Directory Listing where files are incrementally ordered. **Automagically** detected.
  - 2022-05-01-0001.csv
  - 2022-05-01-0002.csv
  - Etc.
- File Notifications allow **scaling** to millions of files/hour, but require permissions

Auto Loader schema management

Schema Location
(cloudFiles.schemaLocation)

- Stores changes to the inferred schema over time

Schema Inference
(cloudFiles.inferColumnTypes)

- Infer column data types or treat everything as a String

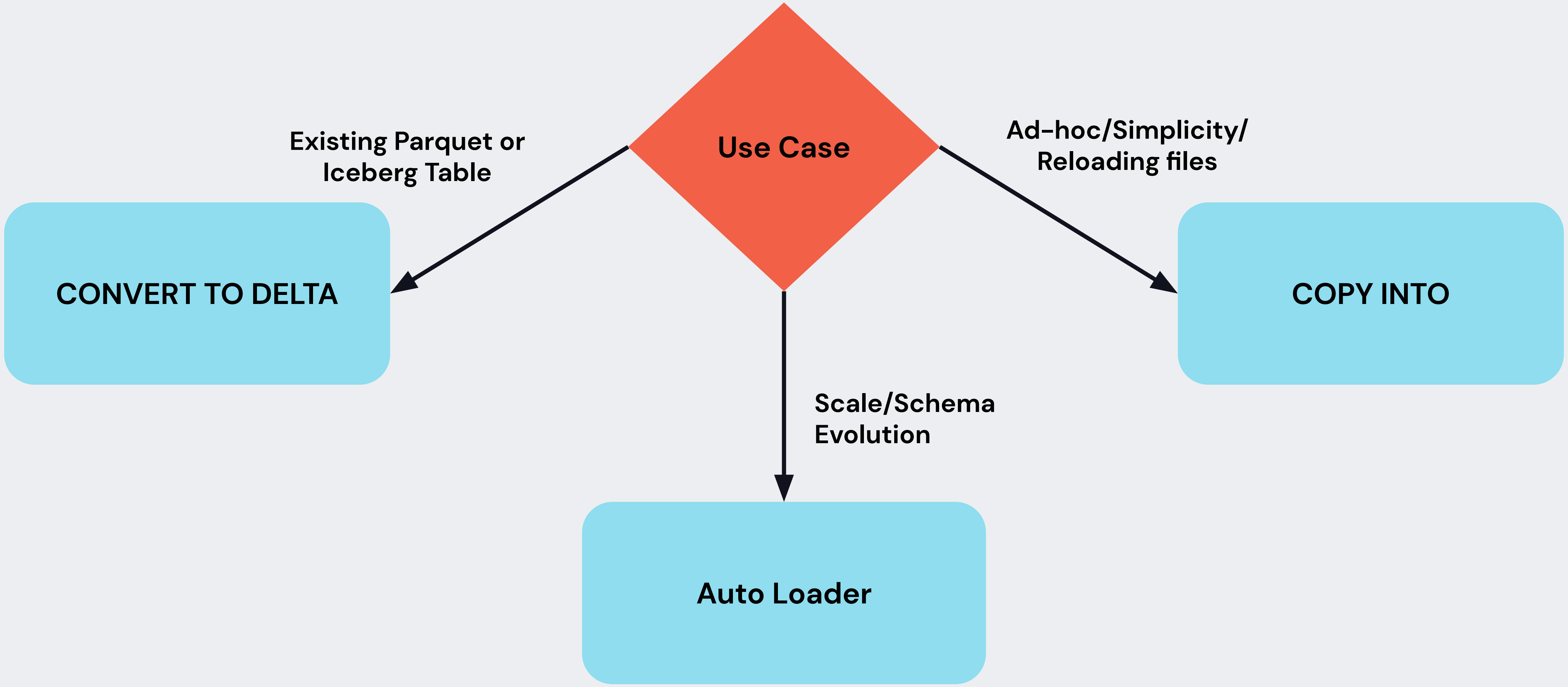
Data Rescue
(cloudFiles.rescuedDataColumn)

- Rescues data that does not match your schema expectation

Schema Evolution Mode
(cloudFiles.schemaEvolutionMode)

- Add new columns
- Fail on new columns
- Rescue new columns

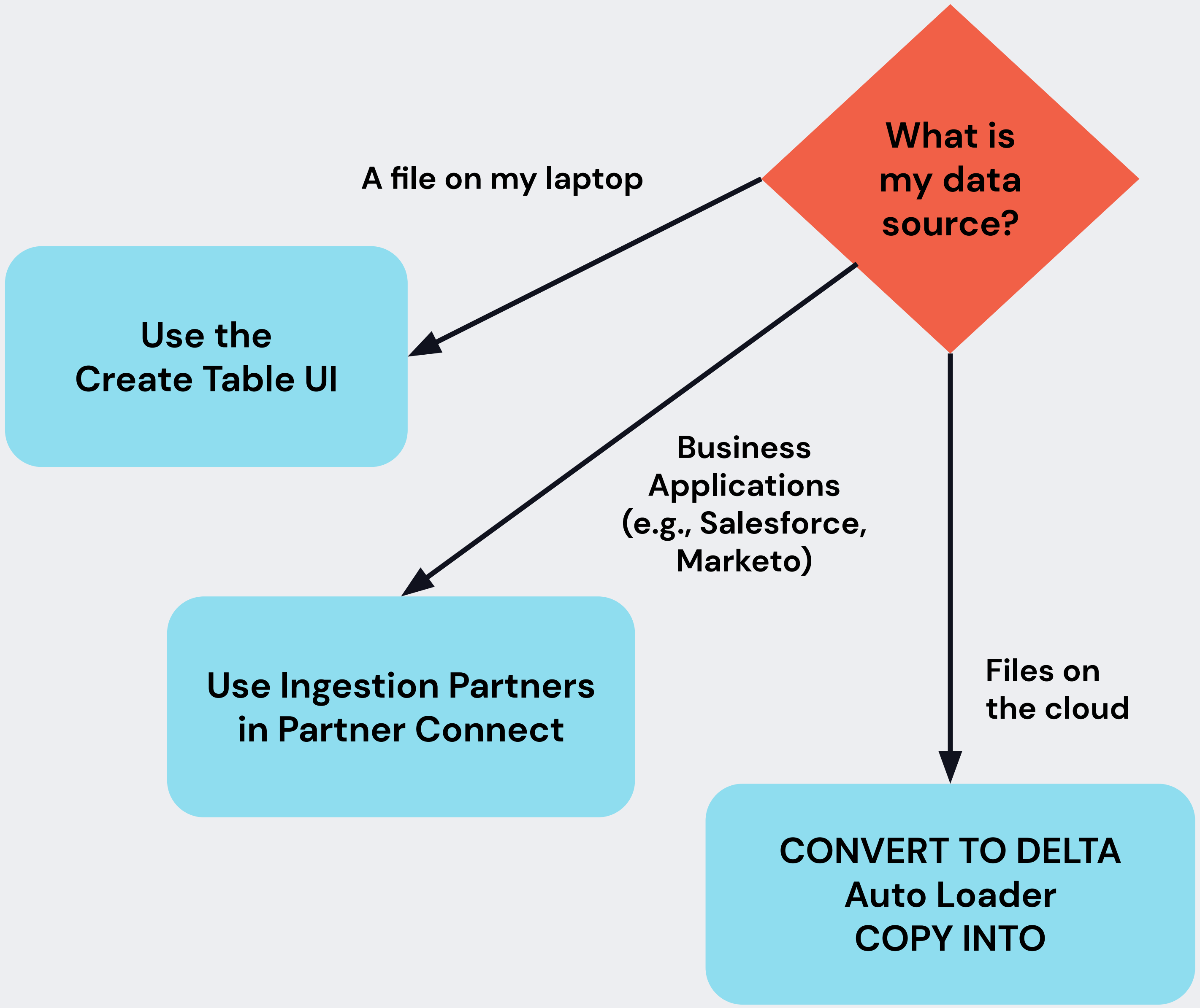
Ingesting files on the Cloud

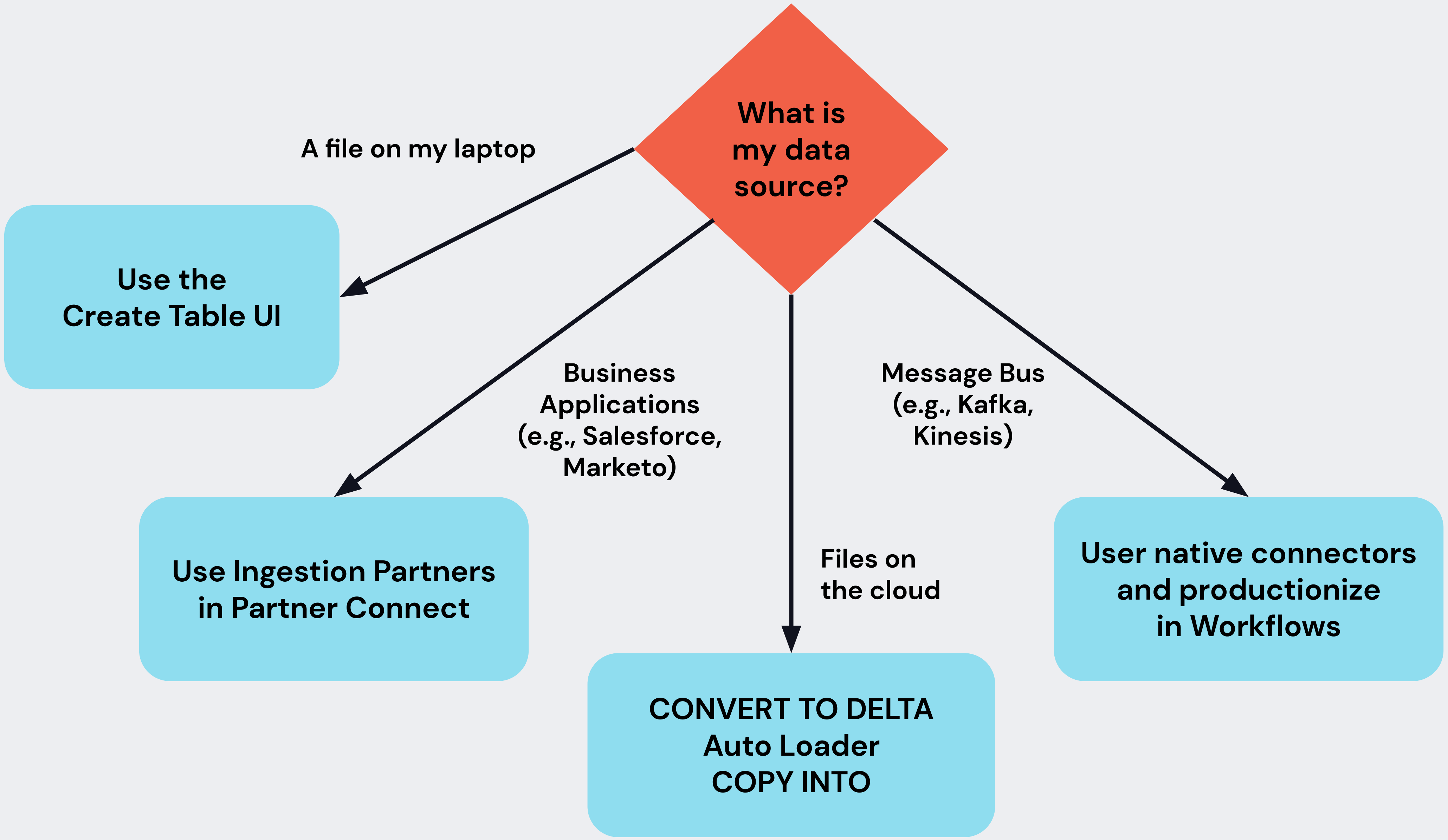

Use Case
Existing Parquet or Iceberg Table
CONVERT TO DELTA
Ad-hoc/Simplicity/ Reloading files
COPY INTO
Scale/Schema Evolution
Auto Loader

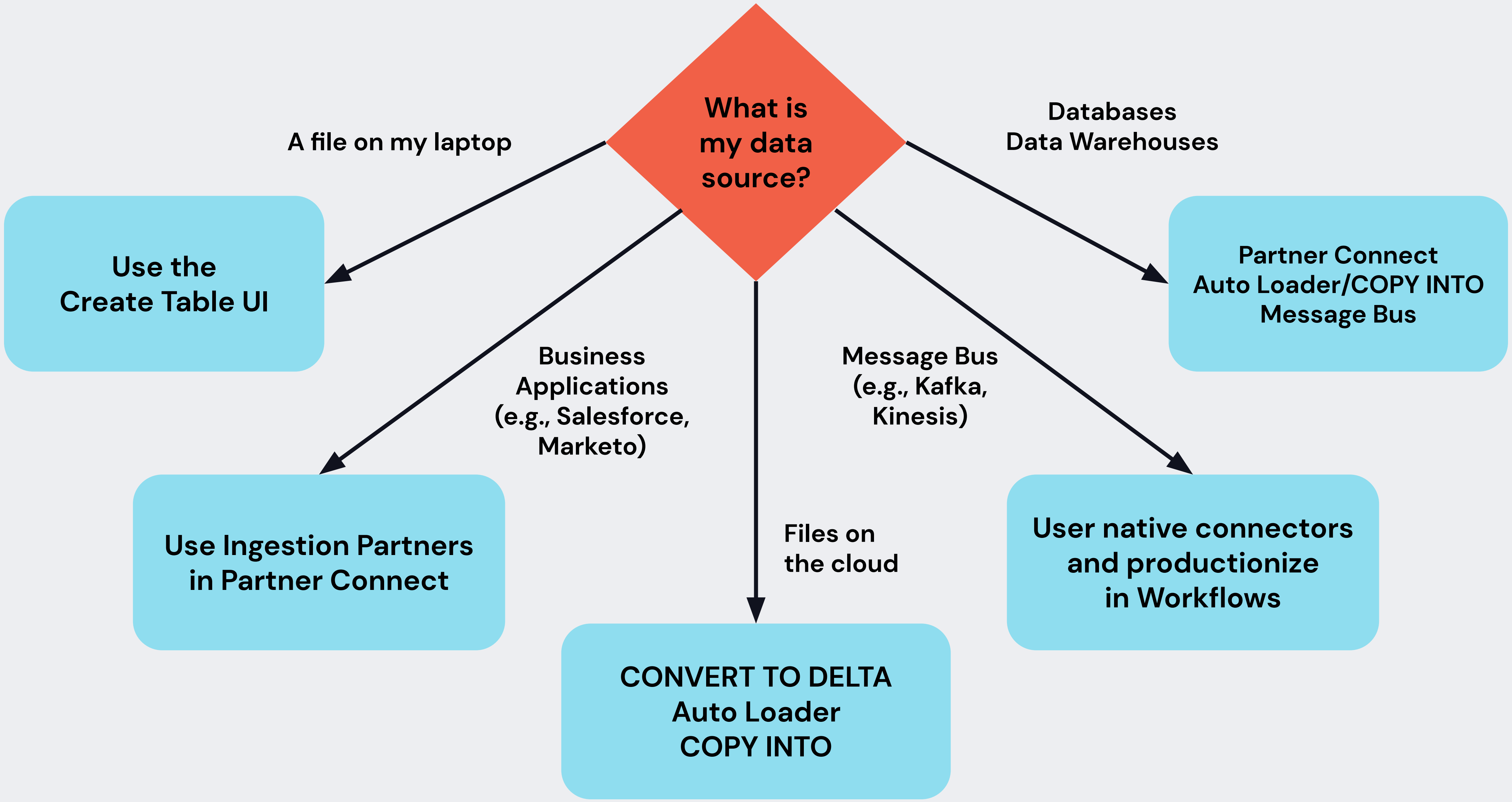
COPY INTO

- SQL command
- Idempotent and incremental
- Great when source directory contains ~ thousands of files
- Schema automatically inferred

```
COPY INTO my_delta_table
FROM 's3://my-bucket/path/to/csv_files'
FILEFORMAT = CSV
FORMAT_OPTIONS ('header'='true','inferSchema'='true')
```


What is my data source?
A file on my laptop
Use the Create Table UI
Business Applications (e.g., Salesforce, Marketo)
Use Ingestion Partners in Partner Connect
Files on the cloud
CONVERT TO DELTA
Auto Loader
COPY INTO

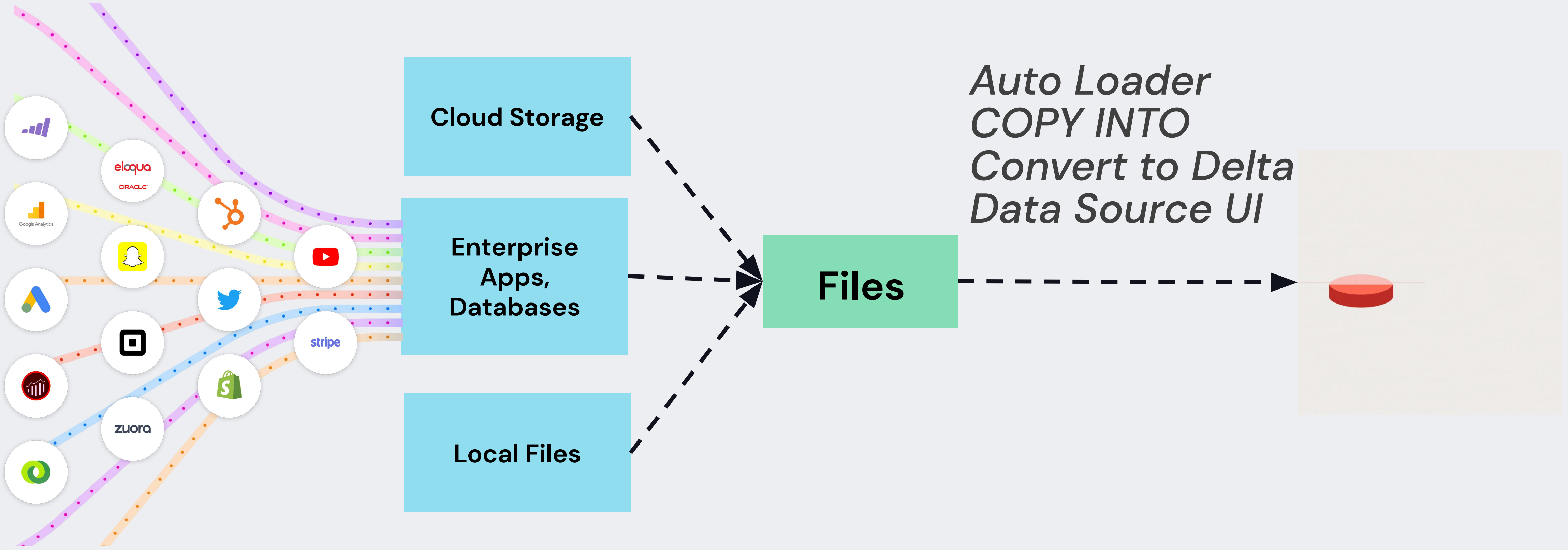

What is my data source?
A file on my laptop
Use the Create Table UI
Business Applications (e.g., Salesforce, Marketo)
Use Ingestion Partners in Partner Connect
Files on the cloud
CONVERT TO DELTA
Auto Loader
COPY INTO
Message Bus (e.g., Kafka, Kinesis)
User native connectors and productionize in Workflows



What is my data source?
A file on my laptop
Use the Create Table UI
Databases Data Warehouses
Partner Connect Auto Loader/COPY INTO Message Bus
Business Applications (e.g., Salesforce, Marketo)
Use Ingestion Partners in Partner Connect
Files on the cloud
CONVERT TO DELTA Auto Loader COPY INTO
Message Bus (e.g., Kafka, Kinesis)
User native connectors and productionize in Workflows

Demo

Common ingestion pattern



Auto Loader
COPY INTO
Convert to Delta
Data Source UI

Related ingestion sessions

DATA+AI SUMMIT 2022

Moving to the Lakehouse: Fast & Efficient Ingestion with Auto Loader

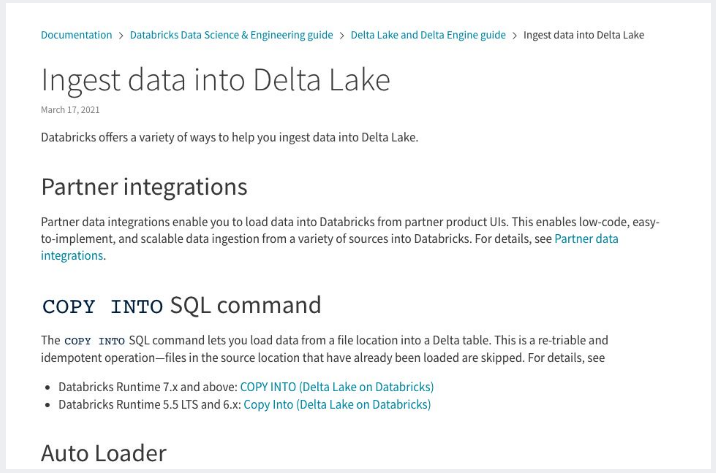
- Virtual session
- Speakers: Emma Liu & Eric Maynard

Ingest Data into Lakehouse with COPY INTO

- Virtual session
- Speakers: Yaohua Zhao & Jackie Zhang

Learn more

Read the documentation



Documentation > Databricks Data Science & Engineering guide > Delta Lake and Delta Engine guide > Ingest data into Delta Lake

Ingest data into Delta Lake

March 17, 2021

Databricks offers a variety of ways to help you ingest data into Delta Lake.

Partner integrations

Partner data integrations enable you to load data into Databricks from partner product UIs. This enables low-code, easy-to-implement, and scalable data ingestion from a variety of sources into Databricks. For details, see Partner data integrations.

`COPY INTO` SQL command

The `COPY INTO` SQL command lets you load data from a file location into a Delta table. This is a re-triable and idempotent operation—files in the source location that have already been loaded are skipped. For details, see

- Databricks Runtime 7.x and above: COPY INTO (Delta Lake on Databricks)
- Databricks Runtime 5.5 LTS and 6.x: Copy Into (Delta Lake on Databricks)

Auto Loader

Ingest data into Delta Lake

Read an ebook

All Roads Lead to the Lakehouse

A deep dive into data ingestion with the lakehouse

Watch Ingestion webinar series

Easily load data into Delta Lake to power analytics, data science and machine learning

DATA+AI SUMMIT 2022

Thank you!

Q&A